UMass Boston Computer Science

CS450 High Level Languages (section 2)

Intertwined Data

Wednesday, October 30, 2024

FROM NOW ON, EVERYONE WHO LIKES DAYLIGHT SAVING TIME SHOULD CHANGE THEIR CLOCKS, AND EVERYONE WHO DOESN'T, SHOULDN'T.

THE GOVERNMENT FINALLY DECIDES TO PUT AN END TO ALL THE ARGUMENTS.

Logistics

- HW 8 out
  - <u>due:</u> **Mon 11/4 12pm (noon) EST**

- (Daylight Savings ends 11/3)

- **no lecture:** Veteran's Day Mon 11/11

FROM NOW ON, EVERYONE WHO LIKES DAYLIGHT SAVING TIME SHOULD CHANGE THEIR CLOCKS, AND EVERYONE WHO DOESN'T, SHOULDN'T.

THE GOVERNMENT FINALLY DECIDES TO PUT AN END TO ALL THE ARGUMENTS.

Intertwined Data Definitions

- Come up with **a Data Definition** for **…**

- **… valid Racket Programs**

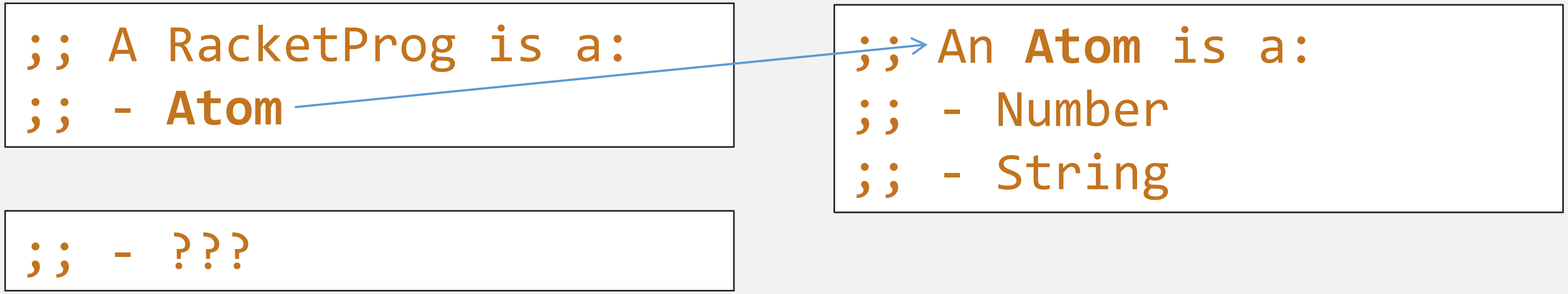
Valid Racket Programs

- 1
- “one”
- (+ 1 2)

```
;; A RacketProg is a:
;; - Number
;; - String
;; - ???
```

Valid Racket Programs

- 1
- “one”
- (+ 1 2)



```
;; A RacketProg is a:
;; - Atom
```

```
;; - ???
```

```
;; An Atom is a:
;; - Number
;; - String
```

Valid Racket Programs

- `(+ 1 2)`

List of … atoms?

“symbol”

```
;; A RacketProg is a:
;; - Atom
;; - List<Atom> ???
```

```
;; An Atom is a:
;; - Number
;; - String
;; - Symbol
```

Written with a single quote, e.g., ‘+

Valid Racket Programs

-
  ```
  (* (+ 1 2)
     (- 4 3))
  ```

-
  ```
  (* (+ 1 2)
     (- 4 3)
     (/ 10 5))
  ```

Tree?

Each tree “node” is a list, of … RacketProgs ??

But: how many values does each node have??

```
;; A RacketProg is a:
;; - Atom
;; - ~~List<???>~~
;; - Tree<???>
```

```
;; An Atom is a:
;; - Number
;; - String
;; - Symbol
```

Valid Racket Programs

-
  ```
  (* (+ 1 2)
     (- 4 3))
  ```

-
  ```
  (* (+ 1 2)
     (- 4 3)
     (/ 10 5))
  ```

Tree?

Each tree “node” is a list, of … RacketProgs ??

But: how many values does each node have??

```
;; A RacketProg is a:
;; - Atom
;; - ProgTree
```

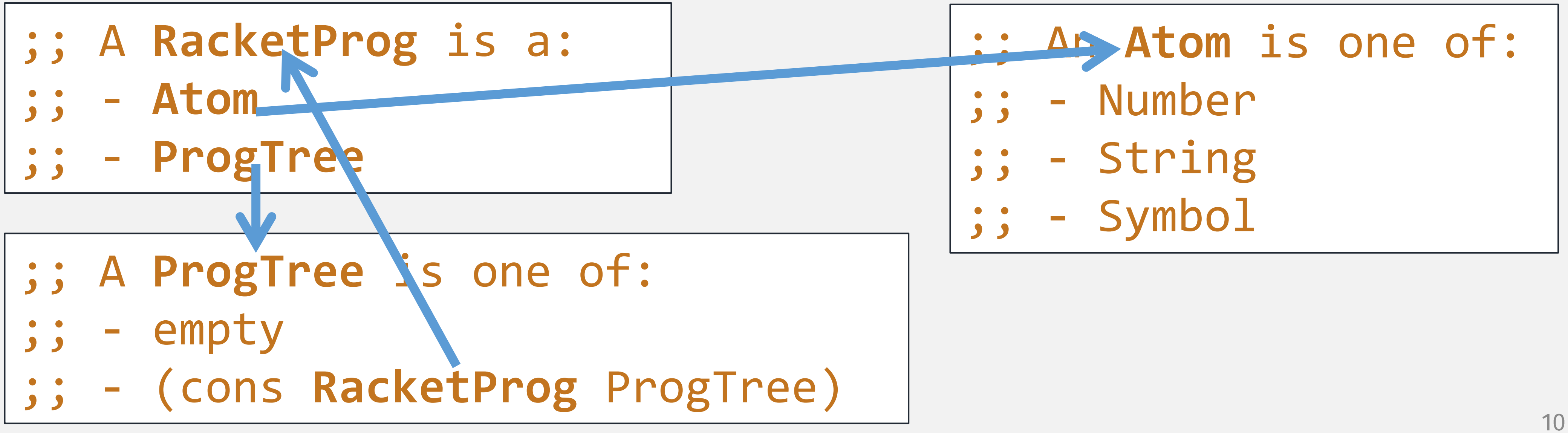
```
;; An Atom is a:
;; - Number
;; - String
;; - Symbol
```

```
;; A ProgTree is one of:
;; - empty
;; - (cons RacketProg ProgTree)
```

Recursive Data Def!

Valid Racket Programs

Also, **Intertwined** Data Defs!



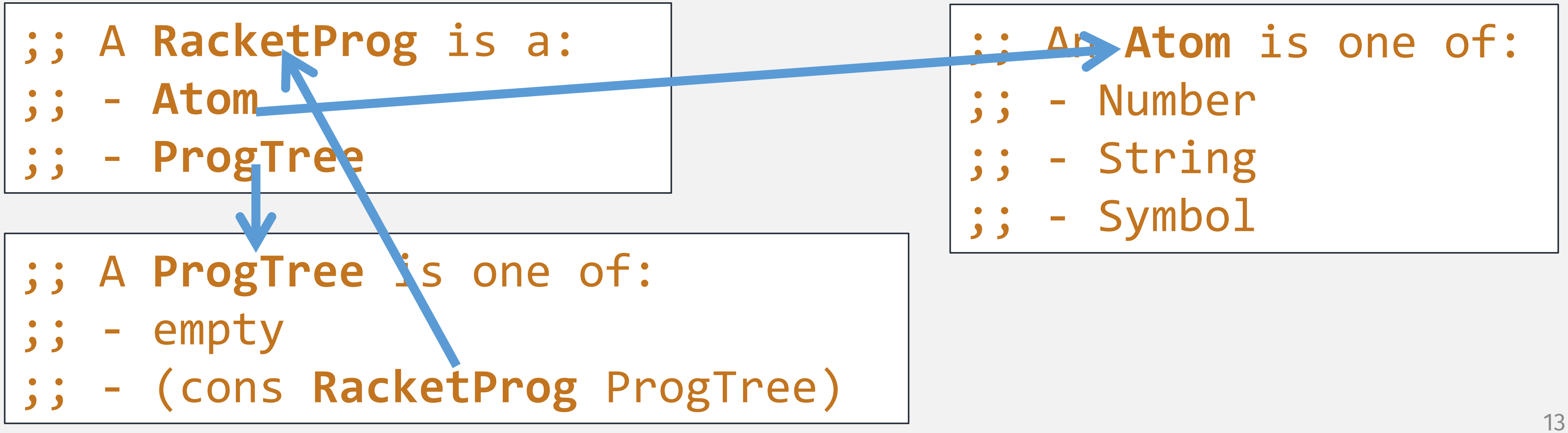
```
;; A RacketProg is a:
;; - Atom
;; - ProgTree
```

```
;; A ProgTree is one of:
;; - empty
;; - (cons RacketProg ProgTree)
```

```
;; An Atom is one of:
;; - Number
;; - String
;; - Symbol
```

Intertwined Data

- A set of Data Definitions that reference each other
- Templates should be defined together …



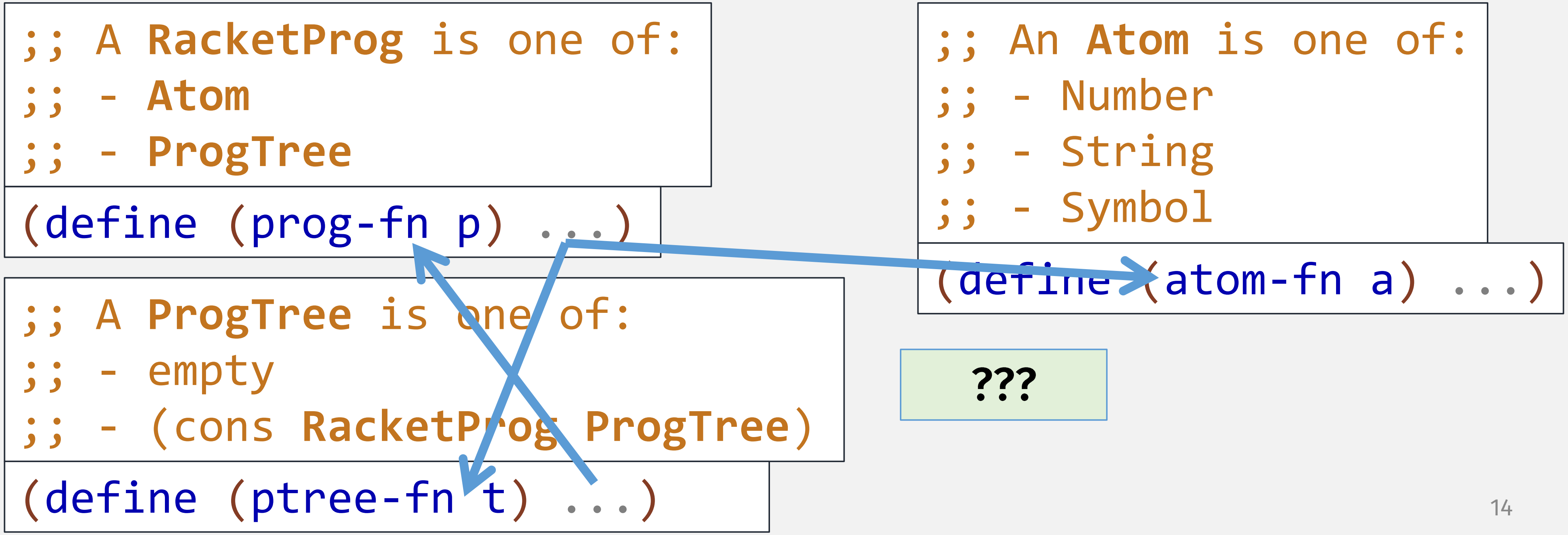
```
;; A RacketProg is a:
;; - Atom
;; - ProgTree
```

```
;; A ProgTree is one of:
;; - empty
;; - (cons RacketProg ProgTree)
```

```
;; An Atom is one of:
;; - Number
;; - String
;; - Symbol
```

Intertwined Data

- A set of Data Definitions that reference each other
- Templates should be defined together …
    - … and should reference each other’s templates (when needed)



```
;; A RacketProg is one of:
;; - Atom
;; - ProgTree
(define (prog-fn p) ...)
```

```
;; A ProgTree is one of:
;; - empty
;; - (cons RacketProg ProgTree)
(define (ptree-fn t) ...)
```

```
;; An Atom is one of:
;; - Number
;; - String
;; - Symbol
(define (atom-fn a) ...)
```

???

- File: **lecture16**.rkt
- Submit: … to GradeScope later

In-class Coding 10/30 #1: Intertwined Templates

- Templates should be defined together …
  - … and should reference each other’s templates (when needed)

```
;; A RacketProg is one of:
;; - Atom
;; - ProgTree
```

```
(define (prog-fn p) ...)
```

```
;; A ProgTree is one of:
;; - empty
;; - (cons RacketPRog ProgTree)
```

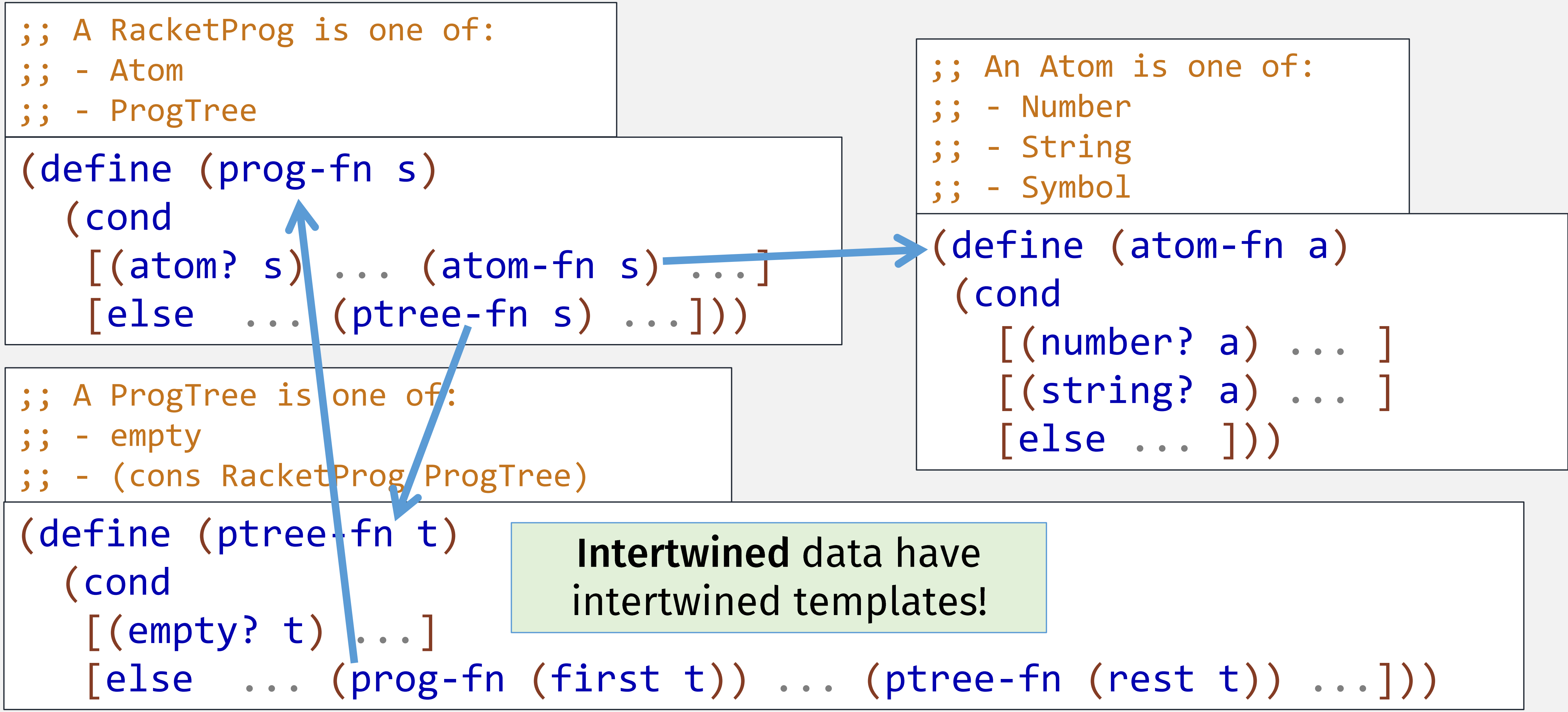
```
(define (ptree-fn t) ...)
```

```
;; An Atom is one of:
;; - Number
;; - String
;; - Symbol
```

```
(define (atom-fn a) ...)
```

???

Intertwined Templates



```
;; A RacketProg is one of:
;; - Atom
;; - ProgTree
```

```
(define (prog-fn s)
  (cond
   [(atom? s) ... (atom-fn s) ...]
   [else   ... (ptree-fn s) ...]))
```

```
;; An Atom is one of:
;; - Number
;; - String
;; - Symbol
```

```
(define (atom-fn a)
 (cond
   [(number? a) ... ]
   [(string? a) ... ]
   [else ... ]))
```

```
;; A ProgTree is one of:
;; - empty
;; - (cons RacketProg ProgTree)
```

```
(define (ptree-fn t)
  (cond
   [(empty? t) ...]
   [else   ... (prog-fn (first t)) ... (ptree-fn (rest t)) ...]))
```

Intertwined data have intertwined templates!

A “Racket Prog” = S-expression!

```
;; A ~~RacketProg~~ Sexpr is one of:
;; - Atom
;; - ProgTree
```

```
(define (sexpr-fn s)
  (cond
   [(atom? s) ... (atom-fn s) ...]
   [else  ... (ptree-fn s) ...]))
```

```
;; An Atom is one of:
;; - Number
;; - String
;; - Symbol
```

```
(define (atom-fn a)
 (cond
   [(number? a) ... ]
   [(string? a) ... ]
   [else ... ]))
```

```
;; A ProgTree is one of:
;; - empty
;; - (cons ~~RacketProg~~ Sexpr ProgTree)
```

```
(define (ptree-fn t)
  (cond
   [(empty? t) ...]
   [else  ... (sexpr-fn (first t)) ... (ptree-fn (rest t)) ...]))
```

- File: **lecture16**.rkt
- Submit: GradeScope “**in-class 10/30**”

In-class Coding 10/30 #2: Counting Symbols

```
;; A Sexpr is one of:
;; - Atom
;; - ProgTree
```

```
;; An Atom is one of:
;; - Number
;; - String
;; - Symbol
```

```
;; count : Symbol Sexpr -> Nat
;; Computes the number of times the given
;; symbol appears in the given s-expression
```

```
;; A ProgTree is one of:
;; - empty
;; - (cons Sexpr ProgTree)
```

```
;; count-atom : Symbol Atom -> Nat
;; ???
```

```
;; count-ptree : Symbol ProgTree -> Nat
;; ???
```