Module 3b: Compound Data

CPSC 110

Peyton Seigo

2018-09-21

Module 3b: Compound Data

- Be able to identify domain information that should be represented as compound data.
- Be able to read and write define-struct definitions.
- Be able to design functions that consume and/or produce compound data.
- Be able to design world programs that use compound world state.

Notes

- **`define-struct`**: defines four general definitions
  - Definitions:
    - the **struct** itself
    - **constructor**: `make-<struct-name>`
    - **selector(s)**: `<struct-name>-<field-name>`
      - A unique selector is created for each field name
    - **predicate**: `<struct-name>?`
  - `(define <struct-name> (x y))`
    - `x` and `y` have given this `struct` two field names
  - Define a `<struct-name>` struct using:
    - `(define S1 (<struct-name> x y))`
    - `x` and `y` set values for the field names

Data definition example:

```
(@HtDD Movie)
(define-struct movie (title budget year))
;; Movie is (make-movie String Natural Natural)
;; interp. metadata for a movie
;;         title is the movie's title
;;         budget is the movie's production budget
;;         year is the year the movie was released
(define M-TITANTIC (make-movie "Titantic" 200000000 1997))
(define M-ELEMENT (make-movie "The Fifth Element" 90000000 1997))
(define M-AVATAR (make-movie "Avatar" 237000000 2009))
(define M-AVENGERS (make-movie "The Avengers" 220000000 2012))

(@dd-template-rules compound)
(define (fn-for-movie m)
  (... (movie-title m)    ; String
       (movie-budget m)   ; Natural
```

```
17          (movie-year m)))    ; Natural
```

Terminology