

CSC151.02 2014S, Class 37: Trees

Overview

- Preliminaries.
 - Upcoming work.
 - Admin.
 - Questions.
- Thinking about trees in terms of cons.
- Thinking about trees recursively.
- Writing recursive tree procedures.

Preliminaries

Upcoming work.

- Readings for Tuesday: Elements of Two-Dimensional Design and Project Guidelines.
- Homework for Tuesday: HW 8: Producing Playful Polygons.
- No writeup today!

Admin

- Welcome to any prospective students!
- New partner assignments: Try to find someone you have not worked with before. If that fails, try to find someone you've worked with infrequently.

Extra Credit

- "Pioneer Weekend is a three day event, sponsored by the Wilson Program and Grinnell AppDev, that is being organized to take place this month from April 18th - 20th. The objective of this event is to bring together student innovators from different backgrounds, to work together in teams of 3-6 people and complete a prototype of an idea that they come up with at the event."
<http://www.strikingly.com/pioneerweekend>
- CS extra Thursday: Software to enhance wellness: The DavisJan team.
- CS table Friday: TBD.
- CS extra next Monday: Walker and Liberto on bluetooth.

Questions on HW

Questions on the Reading

Thinking about trees, starting with lists

- Look, it's '(1 2 3)
- Which is really built by calls to cons, as in (cons 1 (cons 2 (cons 3 null)))
- Behind the scenes, implemented by collected cons cells

What about structures that aren't lists that are built with cons cells

- Sometimes build purposefully
- Sometimes inadvertently
- Why build them? Sometimes useful to represent certain things
 - Useful for images, for example
- Today: Thinking about these structures, which we call trees

Thinking about trees recursively

First, whole numbers

- 0 is a whole number
- If i is a whole number, i+1 is whole number

Lists,

- null is a list
- if l is a list (cons val l) is a list
- Is (cons 2 '(1)) a list?
 - '(1) is a short hand for (cons 1 null)
 - null is a list
 - (cons 1 null) is a list
 - So '(1) is a list
 - So (cons 2 '(1)) is a list
- Is (cons 2 1) a list?
 - Only if 1 is a list
 - Is 1 a list? It's not null
 - 1 is not a list, so (cons 2 1) is not a list

Detour: What is a cons cell?

- Every time we call `(cons x y)`, DrRacket builds a new structure in memory containing two references, one to `x` and one to `y`.
- The diagrams are abstractions of what actually happens in memory.

Defining trees recursively.

A tree of integers

- an integer is a tree of integers
- If `t1` and `t2` are trees of integers, so is `(cons t1 t2)`

Writing recursive tree procedures

Standard recursive technique:

```
(define proc
  (lambda (input)
    (if (simple? input)
        (base-case input)
        (combine (computation input) (proc (simplify input))))))

(define proc
  (lambda (lst)
    (if (null? lst)
        (base-case lst)
        (combine (? (car lst) (proc (cdr lst))))))

(define length
  (lambda (lst)
    (if (null? lst)
        0
        (+ 1 (length (cdr lst))))))
```

Trees have a more complex recursive pattern. You need to recurse on both the `car` and the `cdr`.

```
(define proc
  (lambda (tree)
    (if (pair? tree)
        (combine (proc (car tree)) (proc (cdr tree)))
        (base-case)))
```

Example: Adding the numbers in the tree

```
(define tree-sum
  (lambda (tree)
    (if (pair? tree)
        (+ (tree-sum (car tree)) (tree-sum (cdr tree)))
        0)))
```

Samuel A. Rebelsky, rebelsky@grinnell.edu

Copyright (c) 2007-2014 Janet Davis, Samuel A. Rebelsky, and Jerod Weinman. (Selected materials are copyright by John David Stone or Henry Walker and are used with permission.)



This work is licensed under a Creative Commons Attribution 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.