Programming Languages and Techniques (CIS120)

Lecture 10

February 1st, 2013

Finite Maps

Announcements

- Homework 3 is due *MONDAY* at 11:59:59pm
- Read collaboration policy on syllabus
- Midterm 1
  - Scheduled *in class* on *Friday, February 15th*
  - More details to follow!

Motivating Scenario

- Suppose you were writing some course-management software and needed to lookup the lab section for a student given the student's PennKey?
  - Students might add/drop the course
  - Students might switch lab sections
  - Students should be in only one lab section

- How would you do it?

Finite Maps

- A *finite map, aka dictionary,* is a collection of *bindings* from distinct *keys* to *values*.
  - Operations to add & remove bindings, test for key membership, lookup a value by its key

- Example: an `(ID, int) map` might map a PennKey ID to the lab section.

- Like sets, such finite maps appear in many settings:
  - map domain names to IP addresses
  - map words to their definitions (a dictionary)
  - map user names to passwords
  - map game character unique identifiers to dialog trees
  - …

Finite Map Demo

Using module signatures to preserve data structure invariants

mymap.ml

test_map.ml

Finite Map Interface

```
type ('k,'v) map

val empty    : ('k,'v) map
val is_empty : ('k,'v) map -> bool
val mem      : 'k -> ('k,'v) map -> bool
val find     : 'k -> ('k,'v) map -> 'v
val add      : 'k -> 'v -> ('k,'v) map -> ('k,'v) map
val remove   : 'k -> ('k,'v) map -> ('k,'v) map

val from_list : (‘k * ‘v) list -> (‘k,’v) map
val bindings  : (‘k,’v) map -> (‘k * ‘v) list
```