CSE341: Programming Languages

Lecture 3
Local bindings, Options, Benefits of No Mutation

Dan Grossman

Fall 2011

Review

Huge progress in 2 lectures on the core pieces of SML:

- Types: **`int bool unit  t1*…*tn  t list  t1*…*tn->t`**
  - Types "nest" (each **`t`** above can be itself a compound type)
- Variables and environments
- Functions
  - Build: **`fun x0 (x1:t1, …, xn:tn) = e`**
  - Use: **`e0 (e1, …, en)`**
- Tuples
  - Build: **`(e1, …, en)`**
  - Use: **`#1 e, #2 e, …`**
- Lists
  - Build: **`[]  e1::e2`**
  - Use: **`null e  hd e  tl e`**

Today

- The big thing we need: local bindings
  - For style and convenience
  - For efficiency (***not*** "just a little faster")
  - A big but natural idea: nested function bindings

- One last feature for last problem of homework 1: options

- Why not having mutation (assignment statements) is a valuable language feature
  - No need for you to keep track of sharing/aliasing, which Java programmers must obsess about