CSE 341: Programming Languages

Dan Grossman

Spring 2008

Lecture 3— Lack of Mutation, let bindings, options

List Review

- Build lists: `[]`, `::`, and shorthand `[e1,e2,...,en]`
- Use lists: `null`, `hd`, `tl`
- Types: Each list has elements of the same type. Examples:

```
int list

(int*int) list

((int*int) list) list
```

- So what are the typing rules for `[]`, `::`, `null`, `hd`, and `tl`?
- Functions that build or use lists are usually recursive
  - And/or use other recursive functions
  - Elegant algorithms by "thinking high-level" (e.g., append)

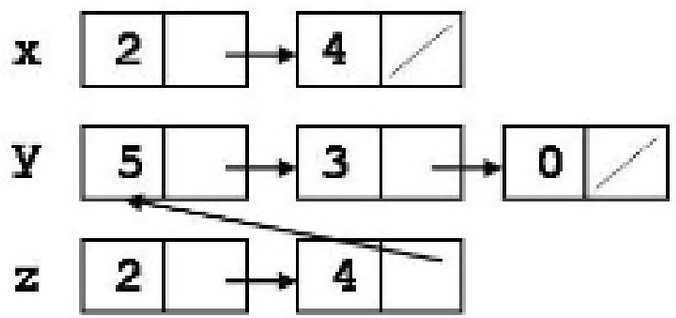
Sharing

Recall `append([2,4],[5,3,0])` evaluates to `[2,4,5,3,0]`.

Similarly, `tl [9,7,4,2]` evaluates to `[7,4,2]`.

Do the results *share*, i.e., *alias* the arguments?

Example: `val x=[2,4]; val y=[5,3,0]; val z=append(x,y)`



versus