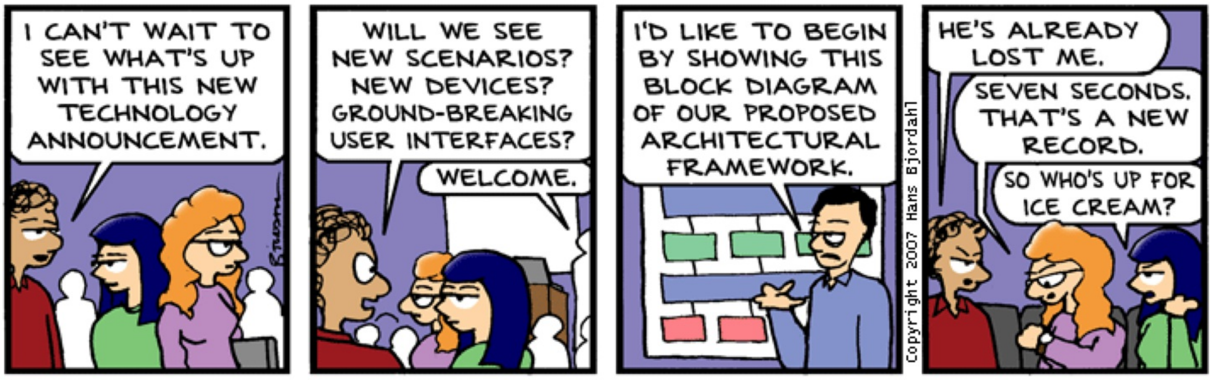
Meeting 17: Functions


I CAN'T WAIT TO SEE WHAT'S UP WITH THIS NEW TECHNOLOGY ANNOUNCEMENT.
WILL WE SEE NEW SCENARIOS? NEW DEVICES? GROUND-BREAKING USER INTERFACES?
WELCOME.
I'D LIKE TO BEGIN BY SHOWING THIS BLOCK DIAGRAM OF OUR PROPOSED ARCHITECTURAL FRAMEWORK.
Copyright 2007 Hans Bjordahl
HE'S ALREADY LOST ME.
SEVEN SECONDS. THAT'S A NEW RECORD.
SO WHO'S UP FOR ICE CREAM?

Bug Bash by Hans Bjordahl http://www.bugbash.net/

Announcements

- HW5 due next Friday 11/3 -- start now!
- Project Pre-Proposal due this Friday
- Midterm will be returned on Thursday
- HW4 COG re-opened until at Thursday 11:55pm
  - Note that COG points are not so significant.
  - Lots of folks working on improving HW4 after the deadline to get ready for the interviews and HW5.
  - But don't ignore HW5. HW5 has started!
  - Do you have lessons learned about how to approach the homeworks from HW4?

Today

- Preview HW5 Functions
- Explicate from HW4?
- Your Questions

Questions

① Flattening lambdas and functions

② Closures (and closure conversions) partial

③ Subscripts — reading semantics
— explications HW4

④ Laying out assembly of functions ✓

⑤ Preview HW5 ✓

⑥ How functions affect liveness

```
def main():                                         main:
    x=3
    def f():
        return [x]
    f()
```

```
def main():                                         fcode
    x=3
    y=10
    f_fv_env=[x]          <- caveat wrt heapify
    f=[fcode, f_fv_env]   <- function value = closure
         code     env
    f[0](f[1])
def fcode(fv_env):                       x ~ fv_env[0]
    return fv_env[0]
```

```
a=2     c=3
b=1
def g():
    a=0          a
    return a+b
```

find the free variables
"used here and defined outside"

P2 "growing the language"

1) closed functions (w/o free fv)

2) add open functions (w/ free fv) (closure conversion
but only "immutable" vars
assign once

3) adding mutable variables (heapify)