CS 61A Week 4

Topic: Data abstraction

Reading: Abelson & Sussman, Sections 2.1 and 2.2.1 (pages 79–106)

Homework:

Abelson & Sussman, exercises 2.7, 2.8, 2.10, 2.12, 2.17, 2.20, 2.22, 2.23

(Note: "Spans zero" means that one bound is $\leq$ zero and the other is $\geq$ zero!)

• Write a procedure `substitute` that takes three arguments: a list, an *old* word, and a *new* word. It should return a copy of the list, but with every occurrence of the old word replaced by the new word, even in sublists. For example:

```
> (substitute '((lead guitar) (bass guitar) (rhythm guitar) drums)
              'guitar 'axe)
((lead axe) (bass axe) (rhythm axe) drums)
```

• Now write `substitute2` that takes a list, a *list* of old words, and a *list* of new words; the last two lists should be the same length. It should return a copy of the first argument, but with each word that occurs in the second argument replaced by the corresponding word of the third argument:

```
> (substitute2 '((4 calling birds) (3 french hens) (2 turtle doves))
               '(1 2 3 4) '(one two three four))
((four calling birds) (three french hens) (two turtle doves))
```

Note: The first midterm is next week.

Extra for experts:

Write the procedure `cxr-function` that takes as its argument a word starting with `c`, ending with `r`, and having a string of letters `a` and/or `d` in between, such as `cdddadaadar`. It should return the corresponding function.

Abelson & Sussman, exercise 2.6. Besides addition, invent multiplication and exponentiation of nonnegative integers. If you're *really* enthusiastic, see if you can invent subtraction. (Remember, the rule of this game is that you have only `lambda` as a starting point.) Read `~cs61a/lib/church-hint` for some suggestions.

Unix feature of the week: `rm`, `mv`, `cp`, `rmdir`, `ln -s`

Emacs feature of the week: `M-%` (find and replace text)