COMP 302 Lecture 5 14 September 2016

see 5c302

Let-abstraction
Rather than writing nested functions, it would be nice if syntax in the language did that for us.
Comes from LISP.

see 5c302-1

Variations on let. By default, let assumes that var1-exp1 and var2-exp2, etc, are independent.
Cannot refer to var1 in exp2.

- let* allows the above
- letrec allows recursive definitions

In JS, we do have var (which we will use). We can form let from basic principles.
JS has var and let. The difference is scoping.

see 5c302-2

LISP was well known for not having much of a data structure.

- Define lists, ex (a b c d e).
- Or rather, defines pairs, recursing.
    - (a (b (c (d e))))
    - Or more accurately (a (b (c (d (e ()))))))

To do this, we have a few operators

- cons(a, b) → (a b)
- car(pair) → returns the 1st of the pair
    - car(cons(a, b)) → a
- cdr(pair) → returns the 2nd of the pair
    - cdr(cons(a, b)) → b

see 5c302-3

In JS
see 5c302-4