Proj #1: March 6
Midterm March 13
#2: Mar 27
#3: Apr 17
#4: May 8

dangling else ambiguity
C: if(…)
if(…)
y = 7;
else(…)

error recovery
LL(1) C num
E
T
T'
F
add semi-column and jump forward

Bottom-up parsing

E->E+T | T
T->T*F | F
F->(e) | id

a+b*c
ID + ID * ID <= F+ id*id <= T + id*id <= E + id*id <= E + F*id <= E+T*id <= E+T*F <= E+T <=E
a rightmost derivation, always expand on the rightmost
handle

Operator precedence argument / parser
E->E op E

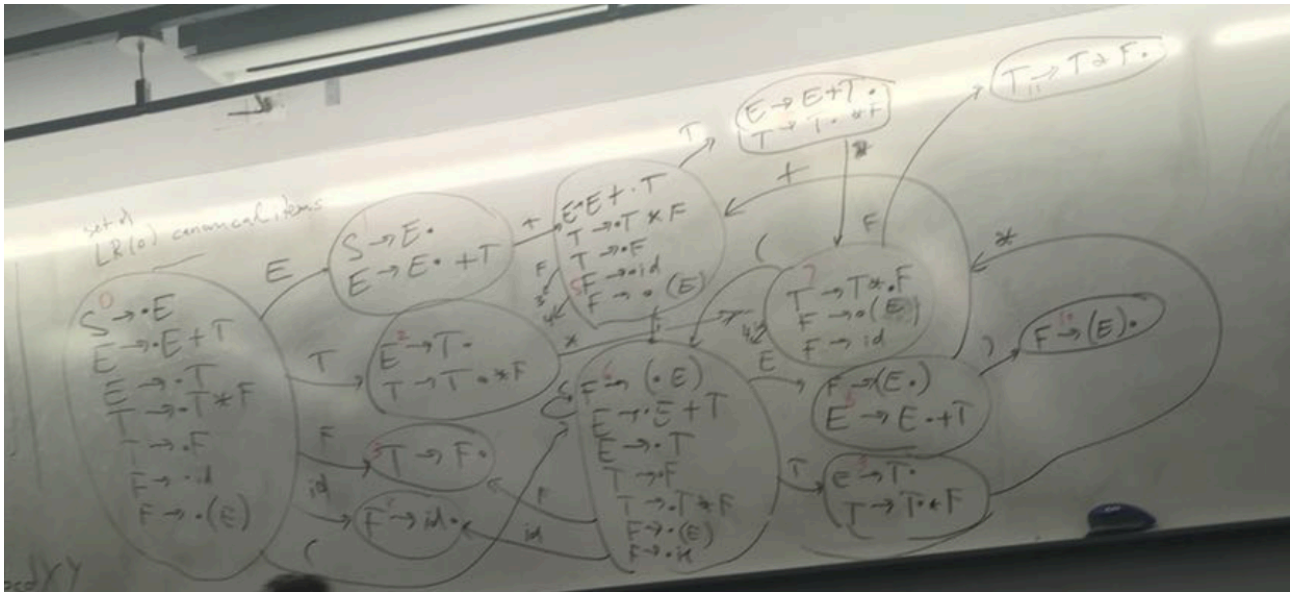
shift-reduce parsers
shift means move token from right to left
Parsing stack | unprocessed input
$ id + id * id$
$id + id * id$
$F + …
$T + …

Reduce: replace Right hand side of a token by a non-terminated
E = E+ T
<— — —
$ a + b * c$
$a + b * c$
$E + b * c$
$E+ b * c$
$E+ b * c$
$E+ E * c$
$E+ E*E
$reduce -> E

set of LR(0) canonical items
S->E
E->E+T | T
T->T*F | F
F->(e) | id

S -> @E -> E (see a E): S-> E@, E-> E@ + T
E -> @E+T
E -> @T -> T (see a T) : E->T@,…
T -> @T*F -> T->F@
F-> @id
T -> @F
F -> @(e) -> F->(@e) E-> @E+T E->@T

LR(0) automaton



stack symbol input
0 $ id*id$
shift #1
04 $id *id$ (4 is the state num in the above automaton)
no outgoing arrow , so reduce, F->id
03 $F
so reduce, T->F
02 $T *id$
shift
027 $T* id$
shift
024 $T*id $
02711 $T*F $
02 $T
$E

$see S->E@ pick

store
next token
state ( )

reduce reduce ambiguity / reduce shift amibuity