

CSCI 3200

Final LL example



Today

↳ Grt is up
(see handout)

- Hw due Friday

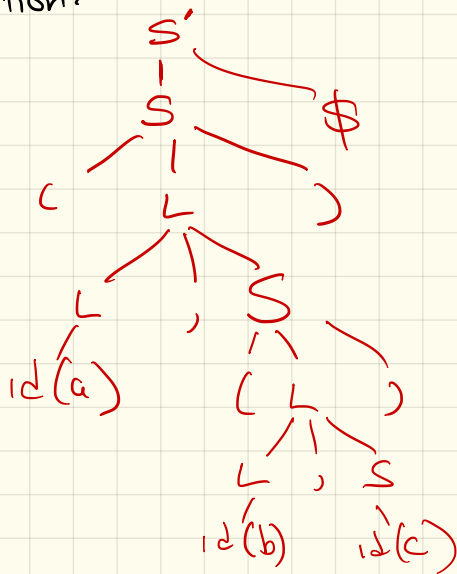
One more example:

A grammar for lists/tuples:

$$\begin{aligned} \text{start } S' &\rightarrow S \$ \\ S &\rightarrow (L) \mid \text{id} \\ \underline{L} &\rightarrow \underline{L}, S \mid \text{id} \end{aligned}$$

Ex: $(a, (b, c)) \$$

Derivation:



Problem:

↳ left recursive

LL version: same trick as before

$$S' \rightarrow S \underline{\$}$$

$$S \rightarrow (L) \mid id$$

$$L \rightarrow SL'$$

$$L' \rightarrow \underset{\uparrow}{,} SL' \mid \epsilon$$

FIRST & FOLLOW sets:

Table

S'

S

L

L'

FIRST

(, id

(, id

(, id

,, \epsilon

\uparrow
comma

FOLLOW

\$

{, \$ &)}

)

)

L \to S FOLLOW(S) =
L' \to S FOLLOW(L)
FOLLOW(L')

(Note: \epsilon can't be in follow sets.)

Table: To generate

① For each terminal in $FIRST(A)$, add $A \rightarrow \alpha$ to $M[A, a]$

② If $\epsilon \in FIRST(A)$, then for each b in $FOLLOW(A)$, add $A \rightarrow \alpha$ in $M[A, b]$

In ours, ϵ in $FIRST(L')$
Only thing in $FOLLOW(L')$ is)

③ Any blanks become errors.

Table is key! Tells it how to parse.

Our table:

Nonterminals	()	id	,	\$
S'	$S' \rightarrow S$		$S' \rightarrow S$		
S	$S' \rightarrow (L)$		$S \rightarrow id$		
L	$L \rightarrow SL'$		$L \rightarrow SL'$		
L'		$L' \rightarrow \epsilon$		$L' \rightarrow SL'$	

State (including non-terms)

which rule we apply, based on table

Matched

Stack

Input

Action

$S' \$$

$(a, (b, c)) \$$

use $S' \rightarrow S$

$S \$$

$(a, (b, c)) \$$

use $S \rightarrow (L)$

~~$(L) \$$~~

~~$(a, (b, c)) \$$~~

match! (

(

$L) \$$

$a, (b, c)) \$$

use $L \rightarrow SL'$

(

$SL') \$$

$a, (b, c)) \$$

use $S \rightarrow id$

(

$id(a) L') \$$

$a, (b, c)) \$$

match $id(a)$

(a

$L') \$$

, (b, c)) \\$

HW! ☺

Remember:

This whole approach is just
to "automate" parsing.

LL is a simple yet
powerful & fast class.