

Lecture 21 - Nov. 29

Syntactic Analysis

Bottom-Up Parsing: Handles
Bottom-Up Parsing: Reverse RMD
LR(1) Items: Definition & Exercises

Bottom-Up Parsing: Handles

& Goto Tables
Syntax Error

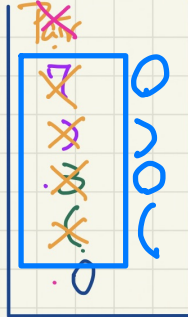
1	Goal	→ List
2	List	→ List Pair
3		Pair
4	Pair	→ (Pair)
5		()

Parse: ()

word: "~~(~~" "~~)~~" eof:
state: ~~0~~ ~~3~~ ~~7~~ ~~2~~ ~~7~~ ~~1~~

handle

List



trace

A **handle** denotes a parser's state that's ready for reduction.

β'' then

State	Action Table		Goto Table	
	eof	()	List	Pair
0	s 3		1	2
1	acc	s 3		4
2	r 3	r 3		
3	s 6	s 7		5
4	r 2	r 2		
5		s 8		
6		s 10		9
7	r 5	r 5		
8	r 4	r 4		
9		s 11		
10		r 5		
11		r 4		

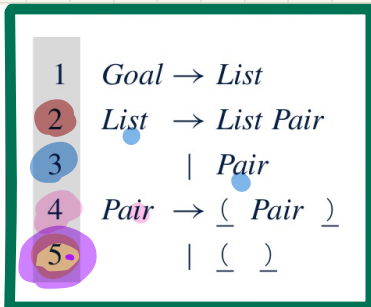
state ready for reduction

Iteration	State	word	Stack	Handle	Action
initial	—	(\$ 0	—	—
1	0	(\$ 0	—	shift 3
2	3)	\$ 0 (3	—	shift 7
3	7	eof	\$ 0 (3) 7	()	<u>reduce 5</u>
4	2	eof	\$ 0 Pair 2	Pair	reduce 3
5	1	eof	\$ 0 List 1	List	accept

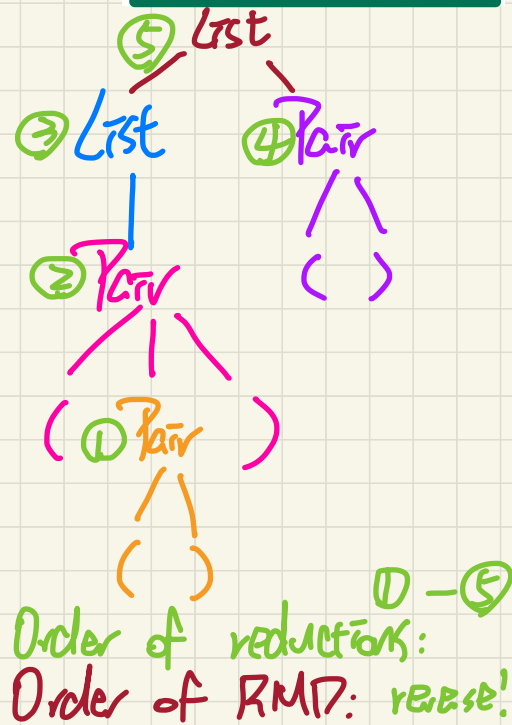
Bottom-Up Parsing: Right-Most Derivation

The **BUP** process corresponds to the reverse of a **RMD**.

Parse: (()) ()

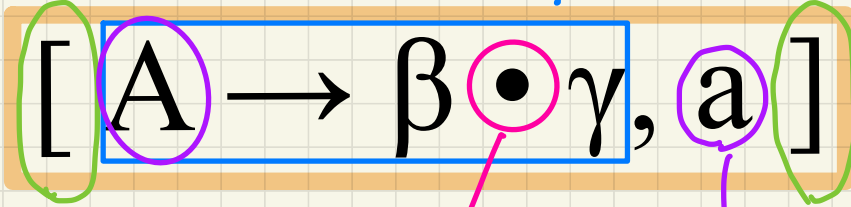


Iteration	State	word	Stack	Handle	Action
initial	—	(\$ 0	— none —	—
1	0	(\$ 0	— none —	shift 3
2	3	(\$ 0 (3	— none —	shift 6
3	6)	\$ 0 (3 (6	— none —	shift 10
4	10)	\$ 0 (3 (6) 10	()	reduce 5
5	5)	\$ 0 (3 Pair 5	— none —	shift 8
6	8	(\$ 0 (3 Pair 5) 8	(Pair)	reduce 4
7	2	(\$ 0 Pair 2	Pair	reduce 3
8	1	(\$ 0 List 1	— none —	shift 3
9	3)	\$ 0 List 1 (3	— none —	shift 7
10	7	eof	\$ 0 List 1 (3) 7	()	reduce 5
11	4	eof	\$ 0 List 1 Pair 4	List Pair	reduce 2
12	1	eof	\$ 0 List 1	List	accept



LR(1) Items: Definition

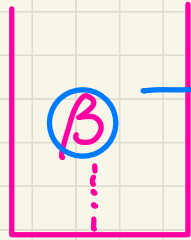
production rule $A \rightarrow \beta\gamma$



possible states of parser

look-ahead $\in \text{Follow}(A)$

current top of stack



- we have already recognize β
- once we recognize γ
↳ in a handle ready for reducing into A

LR(1) Items: Scenarios

Possibility: $[A \rightarrow \cdot \beta \gamma, a]$

↳ initial state of parsing towards reduction to A

Partial Completion: $[A \rightarrow \beta \cdot \gamma, a]$

$\left[\begin{array}{c} \beta \\ \vdots \end{array} \right]$

↳ already recognized β
still expecting to recognize γ

Completion: $[A \rightarrow \beta \gamma \cdot, a]$

$\left[\begin{array}{c} \gamma \\ \beta \\ \vdots \end{array} \right]$

↓ Follow(A)
if word matches a , reduce to A

LR(1) Items: Exercise (1.1a)

- 1 $Goal \rightarrow List$
- 2 $List \rightarrow List Pair$
- 3 $\quad \quad | Pair$
- 4 $Pair \rightarrow (Pair)$
- 5 $\quad \quad | (\underline{\quad})$

Q. **LR(1) item** denoting the **initial** state of parsing?

$[\underline{Goal} \rightarrow \bullet List, \boxed{eof}]$

$\leftarrow Follow(Goal)$
 $\{eof\}$

Q. **LR(1) item** denoting the desired **final** state of parsing?

not necessarily
the final state

$[Pair \rightarrow () \bullet \Rightarrow] [Goal \rightarrow List \bullet, eof]$

LR(1) Items: Exercise (1.1b)

Q. Derive all LR(1) items for the production rule $A \rightarrow \beta\gamma$

- union
- set comprehension
- floating "point"

\mathcal{I}_1 : floating positions of \cdot

$\rightarrow A \rightarrow \cdot \beta \gamma$

$A \rightarrow \beta \cdot \gamma$

$A \rightarrow \beta \gamma \cdot$

\mathcal{I}_2 : Follow(A)

$\{ [A \rightarrow \cdot \beta \gamma, a] \mid a \in \text{Follow}(A) \}$

\cup

$\{ [A \rightarrow \beta \cdot \gamma, a] \mid a \in \text{Follow}(A) \}$

\cup

$\{ [A \rightarrow \beta \gamma \cdot, a] \mid a \in \text{Follow}(A) \}$

LR(1) Items: Exercise (1.2)

1	Goal \rightarrow List
2	List \rightarrow List Pair
3	Pair
4	Pair \rightarrow (Pair)
5	()

How many LR(1) items?
 - possible floating
 - cardinality of Follow(Pair) = 4
 Follow(Pair) = {eof, (,)}
 | Follow(Pair) | = 3
 12

Q. Derive all LR(1) items for the production rule **Pair \rightarrow (Pair)**

FOLLOW(List) = {eof, (} **FOLLOW(Pair) = {eof, (,)}**

- [Pair \rightarrow • (Pair), eof]
- [Pair \rightarrow • (Pair), (]
- [Pair \rightarrow • (Pair),)]
- [Pair \rightarrow (•Pair), eof]
- [Pair \rightarrow (•Pair), (]
- [Pair \rightarrow (•Pair),)]
- [Pair \rightarrow (Pair)•, eof]
- [Pair \rightarrow (Pair)•, (]
- [Pair \rightarrow (Pair)•,)]

LR(1) Items: Exercise (1.3)

- 1 $Goal \rightarrow List$
- 2 $List \rightarrow List Pair$
- 3 $\quad \quad | Pair$
- 4 $Pair \rightarrow (Pair)$
- 5 $\quad \quad | ()$

$$FOLLOW(List) = \{eof, (\}$$

$$FOLLOW(Pair) = \{eof, (,)\}$$

$[Goal \rightarrow \bullet List, eof]$

$[Goal \rightarrow List \bullet, eof]$

$[List \rightarrow \bullet List Pair, eof]$ $[List \rightarrow \bullet List Pair, (]$

$[List \rightarrow List \bullet Pair, eof]$ $[List \rightarrow List \bullet Pair, (]$

$[List \rightarrow List Pair \bullet, eof]$ $[List \rightarrow List Pair \bullet, (]$

$[List \rightarrow \bullet Pair, eof]$ $[List \rightarrow \bullet Pair, (]$

$[List \rightarrow Pair \bullet, eof]$ $[List \rightarrow Pair \bullet, (]$

$[Pair \rightarrow \bullet (Pair), eof]$ $[Pair \rightarrow \bullet (Pair),)]$ $[Pair \rightarrow \bullet (Pair), (]$

$[Pair \rightarrow (\bullet Pair), eof]$ $[Pair \rightarrow (\bullet Pair),)]$ $[Pair \rightarrow (\bullet Pair), (]$

$[Pair \rightarrow (Pair \bullet), eof]$ $[Pair \rightarrow (Pair \bullet),)]$ $[Pair \rightarrow (Pair \bullet), (]$

$[Pair \rightarrow (Pair) \bullet, eof]$ $[Pair \rightarrow (Pair) \bullet,)]$ $[Pair \rightarrow (Pair) \bullet, (]$

$[Pair \rightarrow \bullet (), eof]$ $[Pair \rightarrow \bullet (), (]$ $[Pair \rightarrow \bullet (),)]$

$[Pair \rightarrow (\bullet), eof]$ $[Pair \rightarrow (\bullet), (]$ $[Pair \rightarrow (\bullet),)]$

$[Pair \rightarrow () \bullet, eof]$ $[Pair \rightarrow () \bullet, (]$ $[Pair \rightarrow () \bullet,)]$

LR(1) Items: Exercise (2)

0	$Goal \rightarrow Expr$	6	$Term' \rightarrow \times Factor Term'$
1	$Expr \rightarrow Term Expr'$	7	$ \div Factor Term'$
2	$Expr' \rightarrow + Term Expr'$	8	$ \epsilon$
3	$ - Term Expr'$	9	$Factor \rightarrow (Expr)$
4	$ \epsilon$	10	$ num$
5	$Term \rightarrow Factor Term'$	11	$ name$

Q. Derive all LR(1) items for the the above grammar.

FOLLOW Set

	<i>Expr</i>	<i>Expr'</i>	<i>Term</i>	<i>Term'</i>	<i>Factor</i>
FOLLOW	eof, $_$	eof, $_$	eof, +, -, $_$	eof, +, -, $_$	eof, +, -, x, \div , $_$