concat strings on M5

CS480
Translators
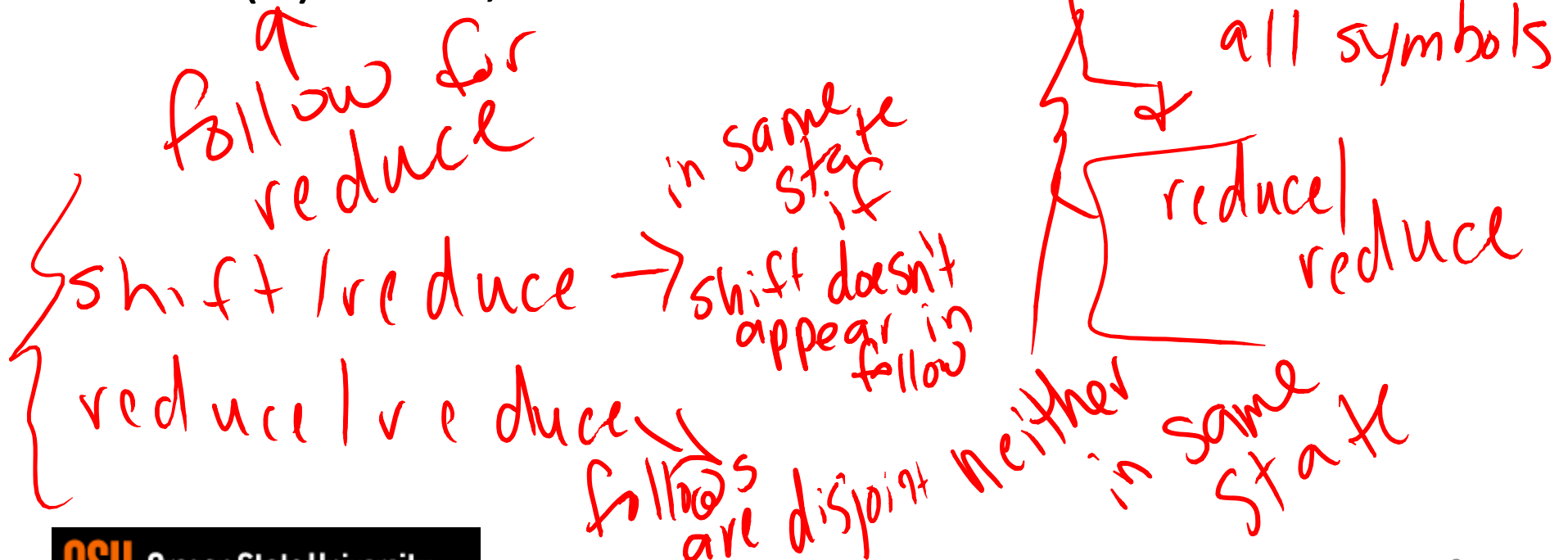
SLR(1) vs. LR(1) Parsing

Chap. 4

Quiz #7 Recap...

- LR(<u>0</u>) Parse, when does it fail?

- SLR(1) Parse, when does it fail?

Quiz #7

Follow(S) =
{$}

- Determine if the grammar is LR(0) or SLR(1)

  S' —> S

  ① S->**real** IDLIST

  ② IDLIST->IDLIST, ID

  ③ IDLIST->ID

  ④ ID->**A|B|C|D**

- Construct the corresponding parse table for the grammar.

- Show how you would parse **real A, B, C**

I_0 S'→ .S
S→ .real IDLIST I_1 I_2

I_1 S'→ S.   Accept

I_2 S→real .IDLIST I_3
IDLIST→ .IDLIST, ID I_3
→ .ID I_4
ID→ .A|B|C|D

ID→ I_5

I_3 S→real IDLIST .R1
IDLIST→ IDLIST .,ID I_6

Oregon State University

Example SLR(1)

1. S->**real** *IDLIST*
2. *IDLIST->IDLIST*, *ID*
3. *IDLIST->ID*
4. *ID->***A** ~~IDLIST->ID~~
5. *ID->***B**
6. *ID->***C**
7. *ID->***D**

Parse **real A, B, C**

Stack	real	,	A	B	C	D	$	S	IDLIST	ID
0	s2							1		
1							A			
2			s5	s6	s7	s8			3	4
3		s9					r1			
4		r3					r3			
5		r4					r4			
6		r5					r5			
7		r6					r6			
8		r7					r7			
9			s5	s6	s7	s8				10
10		r2					r2			

Example SLR(1)

1. *S->**real** IDLIST*
2. *IDLIST->IDLIST**,** ID*
3. *IDLIST->ID*
4. *ID->**A|B|C|D***

Parse **real A, B, C**

Stack	real	,	[ABCD]	$	S	IDLIST	ID
0	s2				1		
1				A			
2			s5				3	4
3		s6		r1			
4		r3		r3			
5		r4		r4			
6			s5					7
7		r2		r2			

When does SLR(1) fails?

S' –> S

1. S –> L = R
2. S –> R
3. L –> *R
4. L –> id
5. R –> L

Follow(S) = ?

Follow(R) = Follow(L) = ?

What happens w/ **id=id**?

Configurating set	Successor
I0: S' –> •S	I1
S –> •L = R	I2
S –> •R	I3
L –> •*R	I4
L –> •id	I5
R –> •L	I2
I1: S' –> S•	Accept
I2: S –> L• = R	**I6**
R –> L•	**Reduce 5**
I3: S –> R•	Reduce 2
I4: L –> *•R	I7
R –> •L	I2
L –> •*R	I4
L –> •id	I5
I5: L –> id•	Reduce 4
I6: S –> L =•R	I8
R –> •L	I9
L –> •*R	I4
L –> •id	I5
I7: L –> *R•	Reduce 3
I8: S –> L = R•	Reduce 1
I9: R –> L•	Reduce 5

6

Let's keep more information

- Repeat the following until no more configurations can be added to state I: *(carry symbol)*
  - For each configuration [A –> u•Bv, a] in I, *(closure)*
    - For each production B –> w in G', and
      - For each terminal b in First(va) such that [B –> •w, b] is not in I: add [B –> •w, b] to I.

LR(1) Collection of Cofigurating Sets

S' –> S

1. S –> L = R
2. S –> R
3. L –> *R
4. L –> id
5. R –> L

What happens w/ **id=id**?

Configurating set	Successor
I0: S' –> •S, $	I1
S –> •L = R, $	I2
S –> •R, $	I3
L –> •*R, =	I4
L –> •id, =	I5
R –> •L, $	I2
L –> •*R, $	??
L –> •id, $	??
I1: S' –> S•	Accept
I2: S –> L• = R, $	**I6**
R –> L• $	**Reduce 5**
I3: S –> R•, $	Reduce 2
I4: L –> *•R, =/$	I7
R –> •L , =/$	I8
L –> •*R , =/$	I4
L –> •id , =/$	I5
I5: L –> id• , =/$	Reduce 4
I6: S –> L =•R, $	I9
R –> •L , $	I2
L –> •*R , $	I10
L –> •id , $	I11

Configurating set	Successor
I7: L –> *R•, =/$	Reduce 3
I8: R –> L•, =/$	Reduce 5
I9: S –> L = R•, $	Reduce 1
I10: L –> *•R, $	I12
R –> •L , $	I2
L –> •*R , $	I10
L –> •id , $	I11
I11: L –> id•, $	Reduce 4
I12: L –> *R•, $	Reduce 3

8

Example LR(1)

1. S –> L = R
2. S –> R
3. L –> *R
4. L –> id
5. R –> L

Parse id=id

Stack	=	id	*	$	S	L	R
0		s5	s4		1	2	3
1				A			
2	s6			r5			
3				r2			
4		s5	s4			8	7
5	r4			r4			
6		s11	s10			2	9
7	r3			r3			
8	r5			r5			
9				r1			
10		s11	s10			2	12
11				r4			
12				r3			

Let's Do an Example...

$S' \rightarrow S$

1. S-> XX
2. X-> aX
3. X-> b

$LR(1)$ — we need carries

$First(\alpha) = \{a, b\}$

$Follow(x) = \{\$, a, b\}$

- First and Follow?
- Table?

I_0 $S' \rightarrow \cdot S, \$$
$S \rightarrow \cdot X X, \$$ (4a)
$X \rightarrow \cdot aX, a/b$
$X \rightarrow \cdot b, a/b$

I_1
I_2
I_3
I_4