LR Parsing Example

CS 4447 / 9545 -- Stephen Watt

University of Western Ontario

Example

1. E → E “+” T
2. E → T
3. T → T “*” F
4. T → F
5. F → “(” E “)”
6. F → **id**

State	Action						Goto		
	Id	+	*	(	)	$	E	T	F
0	s5			s4			1	2	3
1		s6				acc	1		
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6			s11				
9		r1	s7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

Trace of LR Parsing

```
                                                a  *  b + c  * ( d  +  e )
Start
                          ( S0,                      id * id + id * ( id +
  id ) $ )
A[S0,id] = shift  S5
                          ( S0  id  S5,                * id + id * ( id +
  id ) $ )
A[S5,*]  = reduce F→id
G[S0,F]  = goto   S3
                          ( S0 F S3,                   * id + id * ( id +
  id ) $ )
A[S3,*]  = reduce T→F
G[S0,T]  = goto   S2
                          ( S0 T S2,                   * id + id * ( id +
  id ) $ )
A[S2,*]  = shift  S7
                          ( S0 T S2 * S7,                id + id * ( id +
  id ) $ )
A[S7,id] = shift  S5
                          ( S0 T S2 * S7  id  S5,              + id * ( id +
  id ) $ )
```

Trace of LR Parsing (contd)

```
                          ( S0 T S2 * S7 F S10,              + id * ( id +
  id ) $ )
A[S10,+] = reduce T→T*F
G[S0,T]  = goto   S2
                          ( S0 T S2,                        + id * ( id +
  id ) $ )
A[S2,+]  = reduce E→T
G[S0,E]  = goto   S1
                          ( S1 E S1,                        + id * ( id +
  id ) $ )
A[S1,+]  = shift  S6
                          ( S1 E S1 + S6,                     id * ( id +
  id ) $ )
A[S6,id] = shift  S5
                          ( S1 E S1 + S6  id  S5,                 * ( id +
  id ) $ )
A[S5,*]  = reduce F→id
G[S6,F]  = state  S3
                          ( S1 E S1 + S6 F S3,                    * ( id +
  id ) $ )
A[S3,*]  = reduce T→F
```

Trace of LR Parsing (contd)

```
                              ( S1 E S1 + S6 T S9,                    * ( id +
  id ) $ )
A[S9,*]  = shift  S7
                              ( S1 E S1 + S6 T S9 * S7,                 ( id +
  id ) $ )
A[S7,(]  = shift  S4
                              ( S1 E S1 + S6 T S9 * S7 ( S4,              id +
  id ) $ )
A[S4,id] = shift  S5
                              ( S1 E S1 + S6 T S9 * S7 ( S4  id  S5,         +
  id ) $ )
A[S5,+]  = reduce F→id
G[S4,F]  = goto   S3
                              ( S1 E S1 + S6 T S9 * S7 ( S4 F S3,            +
  id ) $ )
A[SS3,+] = reduce T→F
G[S4,T]  = goto   S2
                              ( S1 E S1 + S6 T S9 * S7 ( S4 T S2,            +
  id ) $ )
A[S2,+]  = reduce E→T
G[S4,E]  = goto   S8
```

Trace of LR Parsing (contd)

```
                              ( S1 E S1 + S6 T S9 * S7 ( S4 E S8,          +
  id ) $ )
A[S8,+]  = shift  S6
                              ( S1 E S1 + S6 T S9 * S7 ( S4 E S8 + S6,
  id ) $ )
A[S6,id] = shift  S5
                              ( S1 E S1 + S6 T S9 * S7 ( S4 E S8 + S6,
  id ) $ )
A[S6,id] = shift  S5
                              ( S1 E S1 + S6 T S9 * S7 ( S4 E S8 + S6 id S5,
  ) $ )
A[S5,)]  = reduce F→id
G[S6,F]  = goto   S3
                              ( S1 E S1 + S6 T S9 * S7 ( S4 E S8 + S6 F S3,
  ) $ )
A[S3,)]  = reduce T→F
G[S6,T]  = goto   S9
                              ( S1 E S1 + S6 T S9 * S7 ( S4 E S8 + S6 T S9,
  ) $ )
A[S9,)]  = reduce E→E+T
G[S4,E]  = goto   S8
```

Trace of LR Parsing (contd)

```
                          ( S1 E S1 + S6 T S9 * S7 ( S4 E S8,
  ) $ )
A[S8,)]  = shift S11
                          ( S1 E S1 + S6 T S9 * S7 ( S4 E S8 ) S11,
  $ )
A[S11,$] = reduce F→(E)
G[S7,F]  = goto S10
                          ( S1 E S1 + S6 T S9 * S7 F S10,
  $ )
A[S10,$] = reduce T→T*F
G[S6,T]  = goto S9
                          ( S1 E S1 + S6 T S9,
  $ )
A[S9,$]  = reduce E→E+T
G[S1,E]  = goto S1
                          ( S1 E S1,
  $ )
A[S1,$]  = accept
                          ( S1 E,
  )
```