CS443: Compiler Construction

Lecture 4: LR parsing

Shift-reduce parsers work for LR(k) grammars

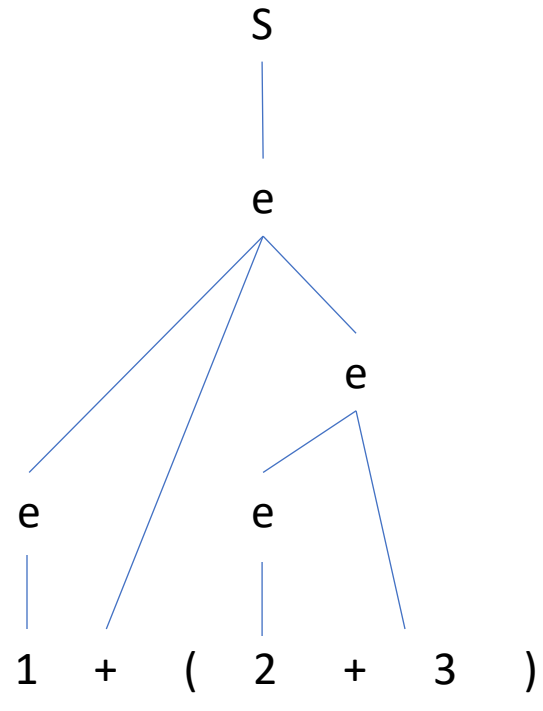
State machine with a stack and 2 actions:

- *Shift* a token onto the stack
- *Reduce* the top of the stack to a nonterminal by a production

Shift-reduce parsers work for LR(k) grammars

S ::= e *e* ::= n | e + n | *e* + (*e*)

Stack	Input	
	1 + (2 + 3)$	
1	+ (2 + 3)$	Shift
e	+ (2 + 3)$	Reduce n -> e
e +	(2 + 3)$	Shift
e + (	2 + 3)$	Shift
e + (2	+ 3)$	Shift
e + (e	+ 3)$	Reduce n -> e
e + (e +	3)$	Shift
e + (e + 3	)$	Shift
e + (e	)$	Reduce e + n -> e
e + (e)	$	Shift
e	$	Reduce e + (e) -> e
S	$	Reduce e -> S, accept



Shift-reduce parsers make decisions based on DFAs

- Edges: terminals + nonterminals on stack
- Just treat the stack as a stack of states (can reconstruct orig. stack)
- Transition table has two parts:
  - ACTION(state, terminal)
    - s*n* – shift state *n* onto stack
    - r*n* – reduce using rule *n*
    - a – accept
    - error (leave the table blank)
  - GOTO(state, nonterminal)
    - next state

Building the DFA for LR(0) parsing (see book for details)

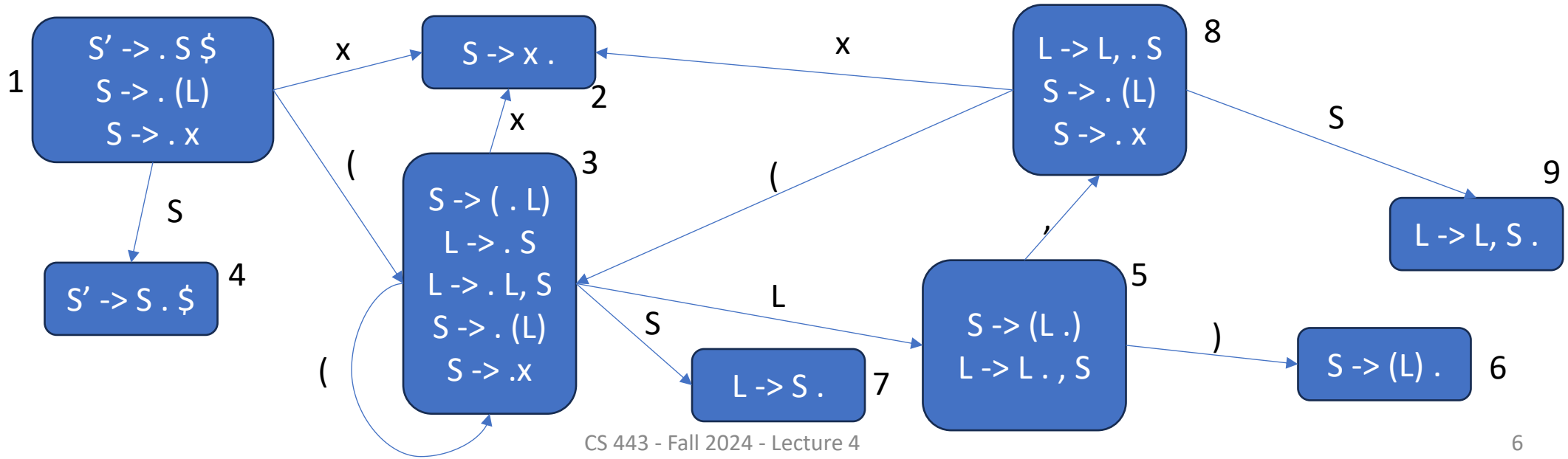
- Items: Productions with a . indicating where we are
  - e.g. e -> e + . n
- DFA states = sets of items

1 S’ -> S$
2 S -> (L)
3 S -> x
4 L -> S
5 L -> L, S

S’ -> . S$
S’ -> S . $
S -> . (L)
S -> (. L)
S -> (L .)
S -> (L) .
S -> . x
S -> x .
L -> . S
L -> S .
L -> . L, S
L -> L ., S
L -> L, . S
L -> L, S .

1 S’ -> S$
2 S -> (L)
3 S -> x
4 L -> S
5 L -> L, S

	(	)	x	,	$	S	L
1	s3		s2			4	
2	r3	r3	r3	r3	r3		
3	s3		s2			7	5
4					a		
5		s6		s8			
6	r2	r2	r2	r2	r2		
7	r4	r4	r4	r4	r4		
8	s3		s2			9	
9	r5	r5	r5	r5	r5		



	(	)	x	,	$	S	L
1	s3		s2			4	
2	r3	r3	r3	r3	r3		
3	s3		s2			7	5
4					a		
5		s6		s8			
6	r2	r2	r2	r2	r2		
7	r4	r4	r4	r4	r4		
8	s3		s2			9	
9	r5	r5	r5	r5	r5		

1 S’ -> S$
2 S -> (L)
3 S -> x
4 L -> S
5 L -> L, S

Stack of states	Stack	Input
1		(x, (y))$
13	(	x, (y))$
132	(x	, (y))$
13	(S	, (y))$
137	(S	, (y))$
13	(L	, (y))$
135	(L	, (y))$
1358	(L,	(y))$
13583	(L, (	y))$
135832	(L, (y	))$
13583	(L, (S	))$
135837	(L, (S	))$
13583	(L, (L	))$
135835	(L, (L	))$
1358356	(L, (L)	)$
1358	(L, S	)$
13589	(L, S	)$
13	(L	)$
135	(L	)$
1356	(L)	$
1	S	$
14	S	$