SLR(1) Parsing

- 1): S→ SS+ 2): S→ SS* 3): S→ a

According to the grammar of 4.2.1, answer the following questions:

  - Augment the grammar and construct the DFA of LR(0) items for the augmented grammar.
  - Is this augmented grammar the LR(0) or SLR(1) grammar ? Give your reason.
  - Construct the SLR(1) parsing table.
  - Show the parsing stack and the action of the parser for the input token string “aa+a*$”.

SLR(1) Parsing

- LR(0) DFA
  - 扩充文法 (argument grammar)

G:
S→SS+|SS*|a

G’:
S’→S
S→SS+|SS*|a

SLR(1) Parsing

- LR(0) DFA

G’:
S’→S
S→SS+|SS*|a

S’→·S

SLR(1) Parsing

- LR(0) DFA

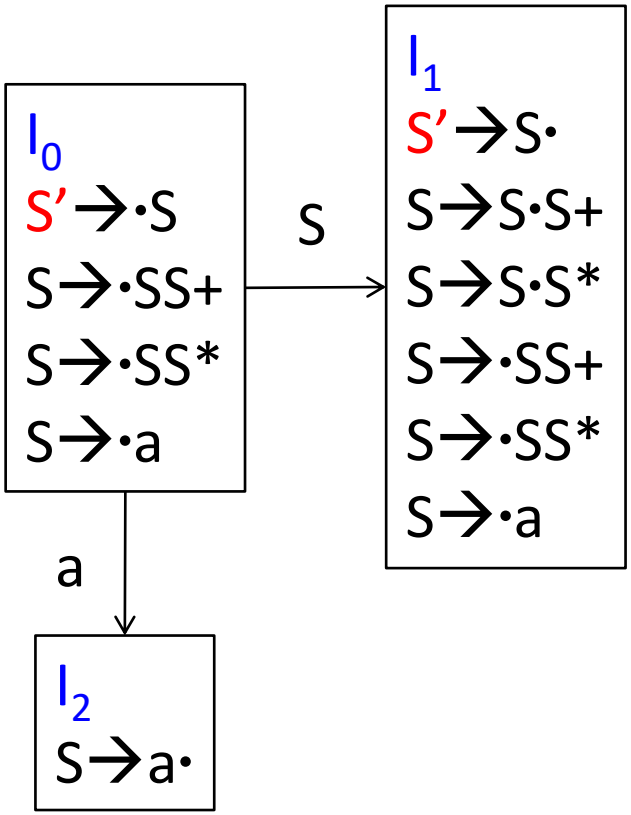
G':
S'→S
S→SS+|SS*|a

I_0
S'→·S
S→·SS+
S→·SS*
S→·a

SLR(1) Parsing

- LR(0) DFA

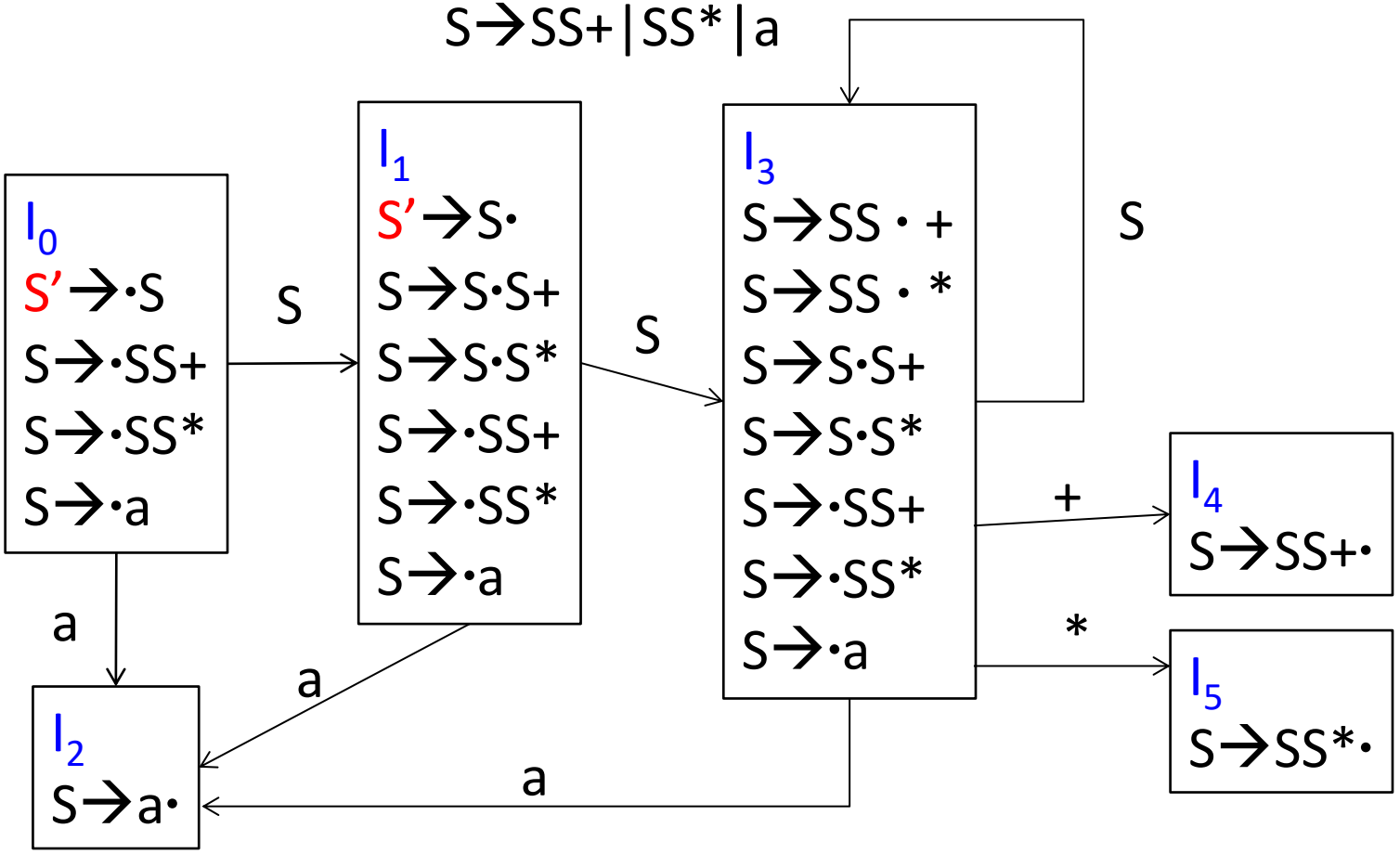
G':
S'→S
S→SS+|SS*|a



SLR(1) Parsing

- LR(0) DFA

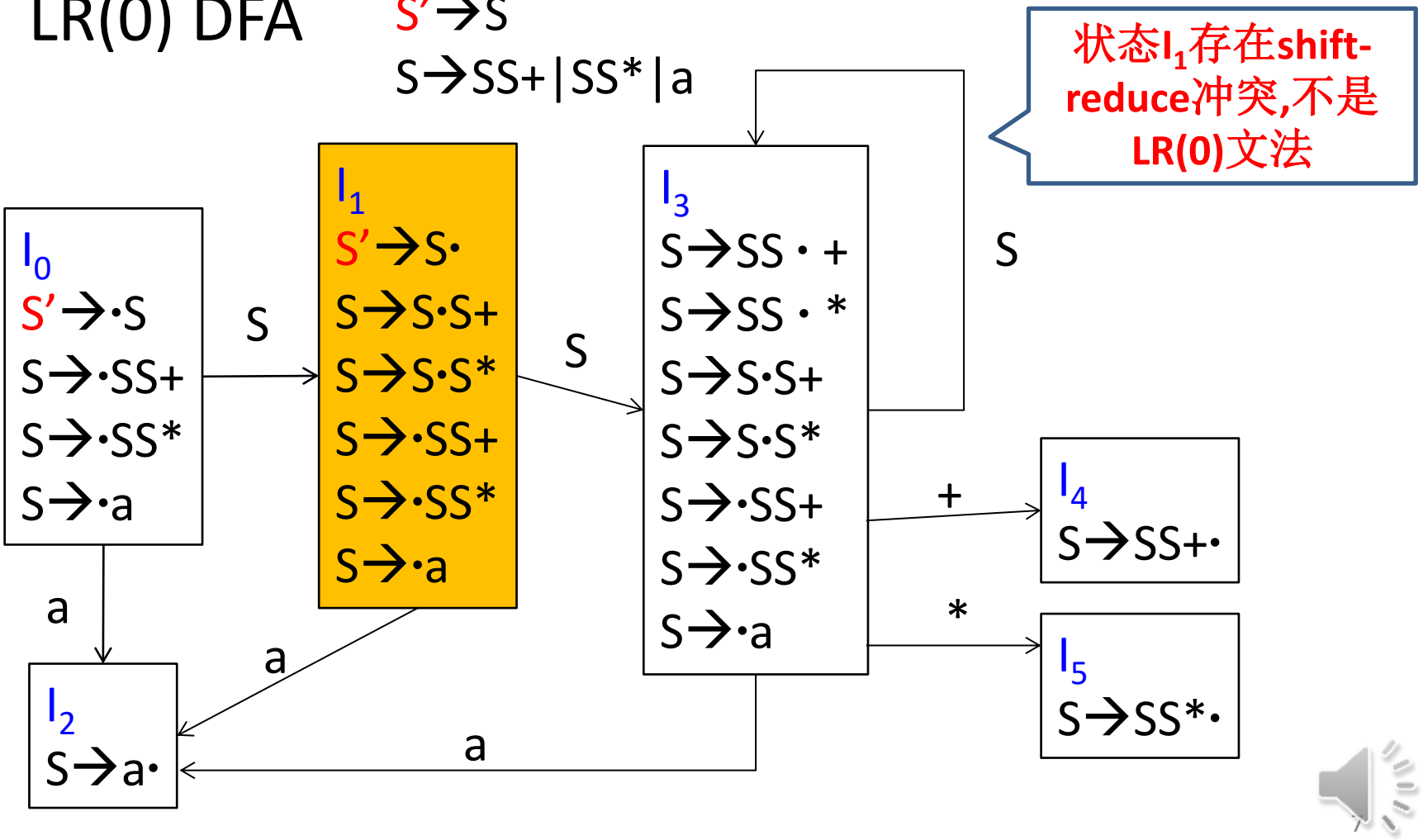
G':
S'→S
S→SS+|SS*|a



SLR(1) Parsing

- LR(0) DFA

G’:
S’→S
S→SS+|SS*|a



SLR(1) Parsing

- The Main Idea of SRL(1)
  - It consults the input token before a shift to make sure that an appropriate DFA transition exists
  - It uses the Follow set of a nonterminal to decide if a reduction should be preformed.
    - For item A→r•, reduction only takes place when the next token a ∈ FOLLOW(A)

G':

(0) S'→S

(1) S→SS+

(2) S→ SS*

(3) S→a

	FOLLOW
S'	$
S	$, +, *, a

For state I_1

•If the next token is '$', then reduce

•If the next token is 'a' , then shift

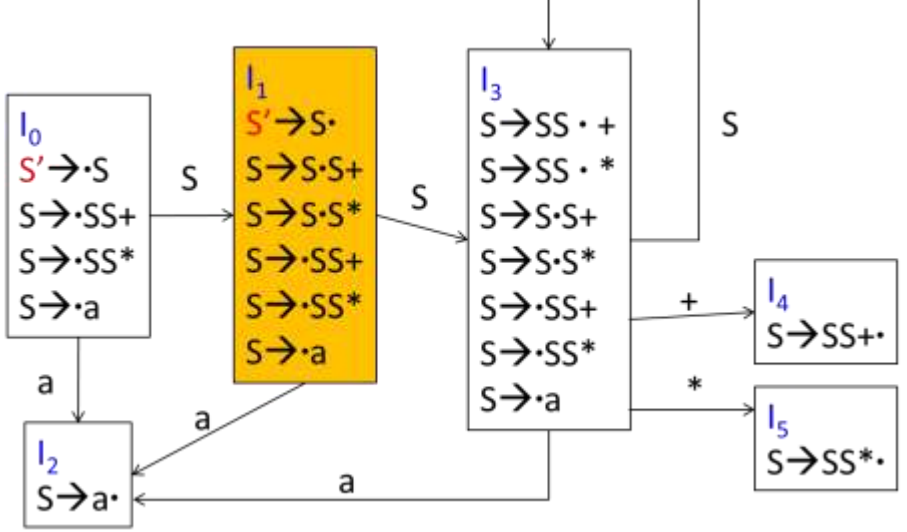
Conflict can be solved. So it is SLR(1) grammar.

Construction of SLR(1) Parse Table

Given a grammar G, we augment G to produce G'

1. Construct DFA of sets of LR(0) items
2. The ACTION section for state K is determined as follows:
   a) If $A \rightarrow a \bullet a\beta \in K$, $a \in V_T$, and goto(K,a)=J, then set ACTION[K,a]='S_J'
   b) If $A \rightarrow a \bullet \in K$, and the number of $A \rightarrow a$ is j, then set ACTION[K,b]=R_j for each $b \in$ Follow(A)
   c) If $S' \rightarrow S \bullet \in K$, then set ACTION[K,$]='acc'
3. The GOTO section for state K is constructed for all nonterminals using the rule: If $A \rightarrow a \bullet B\beta \in K$, $B \in V_N$, and goto(K,B)=J, then set GOTO[K,B]='J'

- SLR(1) Parsing table



G':
(0) S'→S
(1) S→SS+
(2) S→ SS*
(3) S→a

	FOLLOW
S'	$
S	$, +, *, a

	Action				Goto
	a	+	*	$	S
0	s2				1
1	s2			acc	3
2	r3	r3	r3	r3	
3	s2	s4	s5		3
4	r1	r1	r1	r1	
5	r2	r2	r2	r2	

	Action				Goto
	a	+	*	$	S
0	s2				1
1	s2			acc	3
2	r3	r3	r3	r3	
3	s2	s4	s5		3
4	r1	r1	r1	r1	
5	r2	r2	r2	r2	

SLR(1) Parsing

G’:
(0) S’→S
(1) S→SS+
(2) S→ SS*
(3) S→a

- SLR(1) 分析过程模拟

Step	Stack	Input	Action	Goto
1	$0	aa+a* $	s2	
2	$0a2	a+a* $	r3	1
3	$0S1	a+a* $	s2	
4	$0S1a2	+a* $	r3	3
5	$0S1S3	+a* $	s4	
6	$0S1S3+4	a* $	r1	1
7	$0S1	a* $	s2	

SLR(1) Parsing

	Action				Goto
	a	+	*	$	S
0	s2				1
1	s2			acc	3
2	r3	r3	r3	r3	
3	s2	s4	s5		3
4	r1	r1	r1	r1	
5	r2	r2	r2	r2	

- SLR(1) 分析过程模拟

Step	Stack	Input	Action	Goto
8	$0S1a2	* $	r3	3
9	$0S1S3	* $	s5	
10	$0S1S3*5	$	r2	1
11	$0S1	$	acc	