



语法分析

LR分析器的简单模型

李诚

国家高性能计算中心(合肥)、信息与计算机国家级实验教学示范中心

计算机科学与技术学院

2024年09月18日



• 自顶向下 (Top-down)

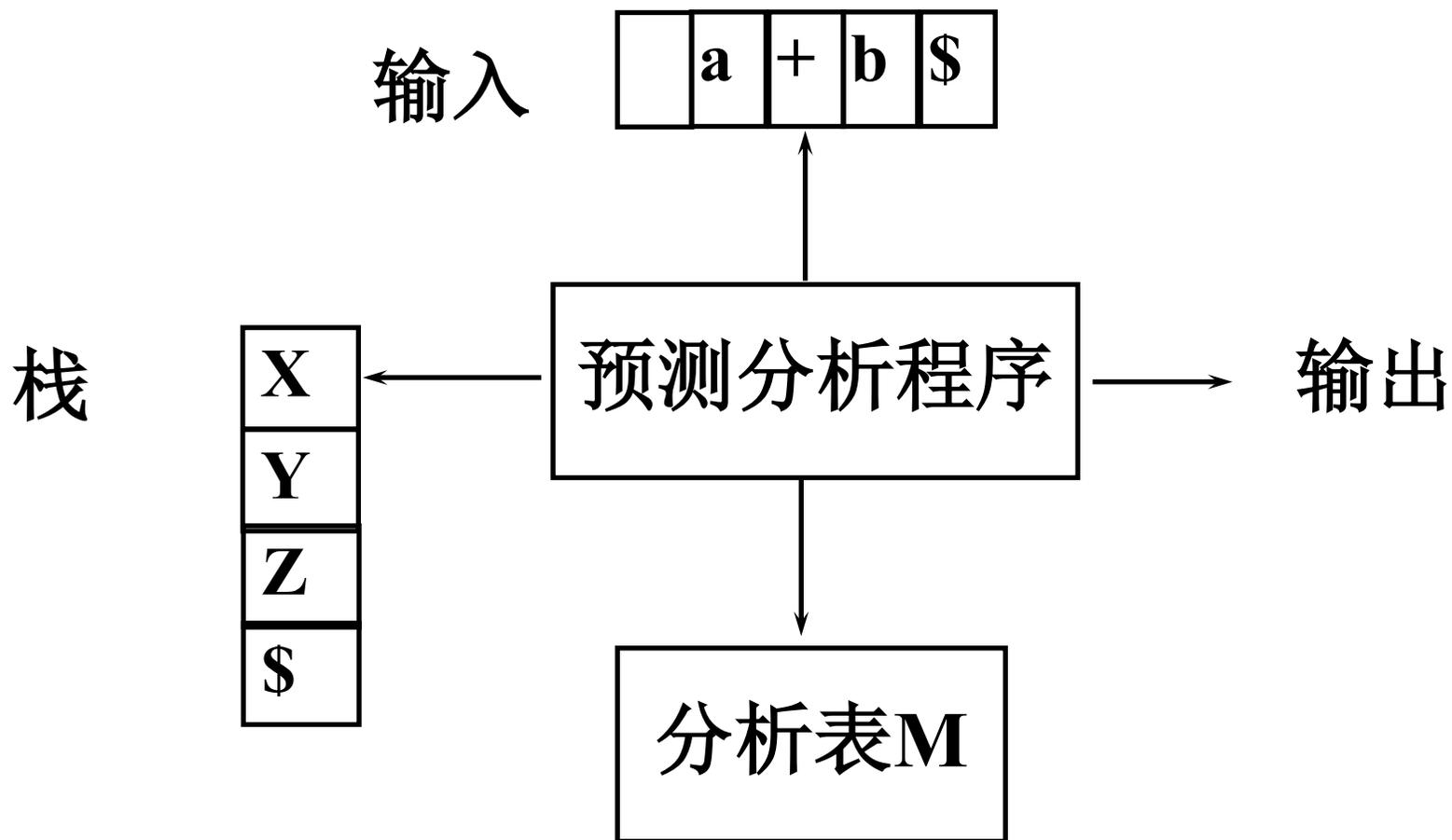
- 针对输入串，从文法的开始符号出发，尝试根据产生式规则推导 (derive) 出该输入串。
- LL(1)文法及非递归预测分析方法
- left-to-right scan + leftmost derivation

• 自底向上 (Bottom-up)

- 针对输入串，尝试根据产生式规则归约 (reduce) 到文法的开始符号。
- LR(k)文法及其分析器
- left-to-right scan + rightmost derivation



复习：LL(1)非递归分析



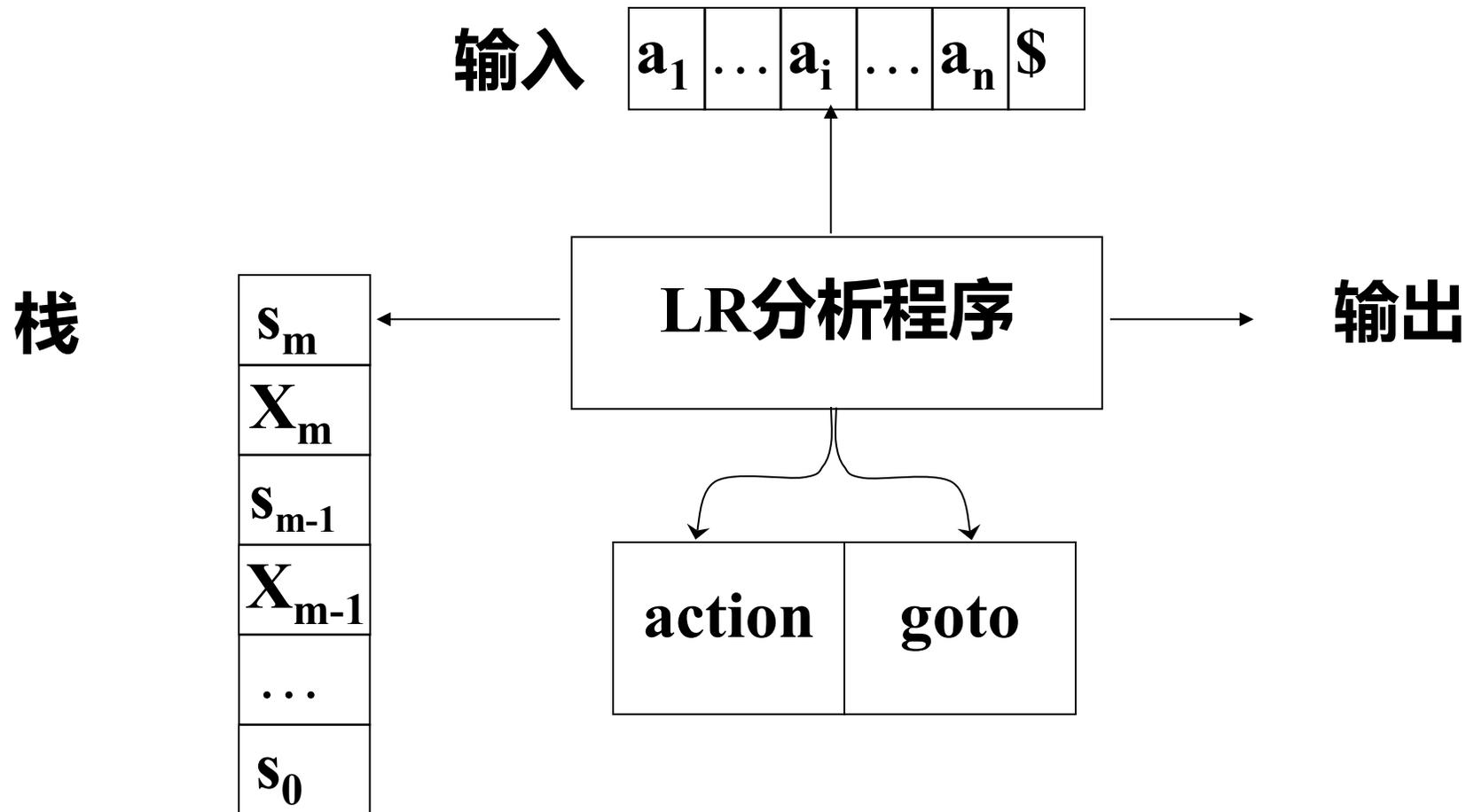


复习：LL(1)非递归分析



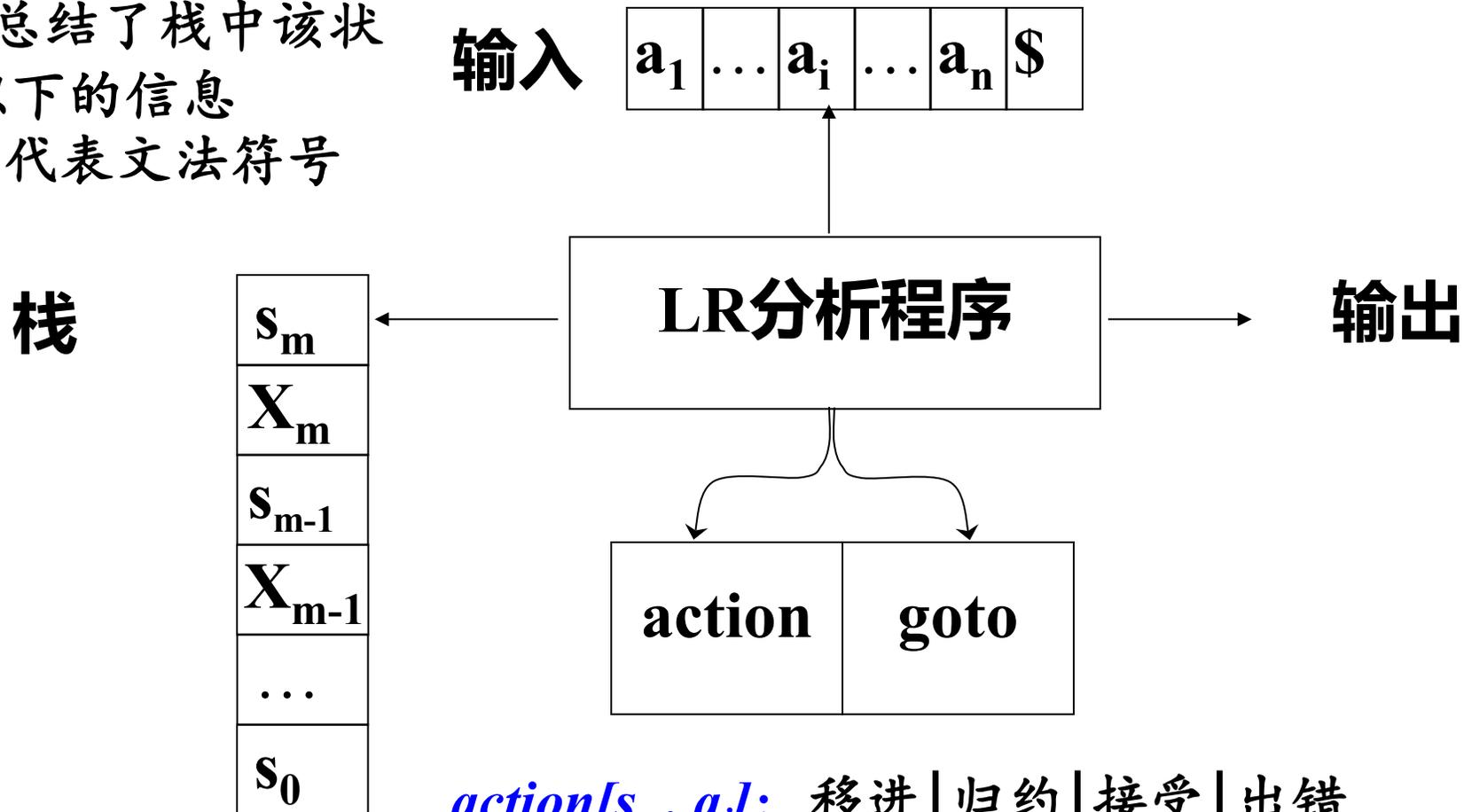
- 行：非终结符；列：终结符或\$；单元：产生式

非终结符	输入符号					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		





s_j : 总结了栈中该状态以下的信息
 X_i : 代表文法符号



$action[s_m, a_i]$: 移进 | 归约 | 接受 | 出错
 $goto[s_{m-r}, A]=s_j$: 移进 A 和 s_j (归约后使用)



LR分析算法：举例



- 例 (1) $E \rightarrow E + T$ (2) $E \rightarrow T$
 (3) $T \rightarrow T * F$ (4) $T \rightarrow F$
 (5) $F \rightarrow (E)$ (6) $F \rightarrow id$

si 移进当前输入符号和状态*i*
rj 按第*j*个产生式进行归约
acc 接受

状态	动作 action					转移 goto			
	id	+	*	()	\$	<i>E</i>	<i>T</i>	<i>F</i>
0	<i>s5</i>					<i>s4</i>	1	2	3
1		<i>s6</i>							
2		<i>r2</i>	<i>s7</i>			<i>r2</i>	<i>r2</i>		
3		<i>r4</i>	<i>r4</i>			<i>r4</i>	<i>r4</i>		
4	<i>s5</i>					<i>s4</i>	8	2	3
5		<i>r6</i>	<i>r6</i>			<i>r6</i>	<i>r6</i>		
6	<i>s5</i>					<i>s4</i>		9	3



LR分析算法：举例



- 例 (1) $E \rightarrow E + T$ (2) $E \rightarrow T$
 (3) $T \rightarrow T * F$ (4) $T \rightarrow F$
 (5) $F \rightarrow (E)$ (6) $F \rightarrow id$

si 移进当前输入符号和状态*i*
rj 按第*j*个产生式进行归约
acc 接受

状态	action					goto			
	id	+	*	()	\$	<i>E</i>	<i>T</i>	<i>F</i>
0	<i>s5</i>			<i>s4</i>			1	2	3
1		<i>s6</i>				<i>acc</i>			
2		<i>r2</i>	<i>s7</i>			<i>r2</i>	<i>r2</i>		
3		<i>r4</i>	<i>r4</i>			<i>r4</i>	<i>r4</i>		
4	<i>s5</i>			<i>s4</i>			8	2	3
5		<i>r6</i>	<i>r6</i>			<i>r6</i>	<i>r6</i>		
6	<i>s5</i>			<i>s4</i>				9	3



LR分析算法：举例



栈	输入	动作
0	id * id + id \$	

状态	action					goto			
	id	+	*	()	\$	E	T	F
0	s5			s4			1	2	3
1		s6				<i>acc</i>			
2		r2	s7		r2	r2			
3		r4	<u>r4</u>		<u>r4</u>	<u>r4</u>			
4	s5			s4			8	2	3
5		r6	<u>r6</u>		<u>r6</u>	<u>r6</u>			
6	s5			s4			9	3	

si 移进当前输入符号和状态 *i*
rj 按第 *j* 个产生式进行归约
acc 接受



LR分析算法：举例



栈	输入	动作
0	id * id + id \$	移进
0 id 5	* id + id \$	

状态	action					goto			
	id	+	*	()	\$	E	T	F
0	s5			s4			1	2	3
1		s6				<i>acc</i>			
2		r2	s7		r2	r2			
3		r4	<u>r4</u>		<u>r4</u>	<u>r4</u>			
4	s5			s4			8	2	3
5		r6	<u>r6</u>		<u>r6</u>	<u>r6</u>			
6	s5			s4			9	3	

si 移进当前输入符号和状态*i*
rj 按第*j*个产生式进行归约
acc 接受



LR分析算法：举例



栈	输入	动作
0	id * id + id \$	移进
0 id 5	* id + id \$	按 $F \rightarrow id$ 归约

1. 查 $action[5, *] \Rightarrow$ 归约
 2. 执行归约 ($F \rightarrow \alpha$):

- 从栈中弹出 $|\alpha|$ 个 \langle 状态, 符号 \rangle 对
- 查 $goto[0, F] \Rightarrow 3$
- 将 $(F, 3)$ 压入栈

状态	action					goto			
	id	+	*	()	\$	E	T	F
0	s5			s4			1	2	3
1		s6				<i>acc</i>			
2		r2	s7			r2	r2		
3		r4	<u>r4</u>			<u>r4</u>	<u>r4</u>		
4	s5			s4			8	2	3
5		r6	<u>r6</u>			<u>r6</u>	<u>r6</u>		
6	s5			s4				9	3

si 移进当前输入符号和状态 *i*
rj 按第 *j* 个产生式进行归约
acc 接受



LR分析算法：举例



栈	输入	动作
0	id * id + id \$	移进
0 id 5	* id + id \$	按 $F \rightarrow id$ 归约
0 F 3	* id + id \$	

状态	action					goto			
	id	+	*	()	\$	E	T	F
0	s5			s4			1	2	3
1		s6				<i>acc</i>			
2		r2	s7			r2	r2		
3		r4	<u>r4</u>			<u>r4</u>	<u>r4</u>		
4	s5			s4			8	2	3
5		r6	<u>r6</u>			<u>r6</u>	<u>r6</u>		
6	s5			s4			9	3	

si 移进当前输入符号和状态 i
 rj 按第 j 个产生式进行归约
 acc 接受



LR分析算法：举例



栈	输入	动作
0	id * id + id \$	移进
0 id 5	* id + id \$	按 $F \rightarrow id$ 归约
0 F 3	* id + id \$	按 $T \rightarrow F$ 归约

状态	action					goto			
	id	+	*	()	\$	E	T	F
0	s5			s4			1	2	3
1		s6				<i>acc</i>			
2		r2	s7			r2	r2		
3		r4	<u>r4</u>			<u>r4</u>	<u>r4</u>		
4	s5			s4			8	2	3
5		r6	<u>r6</u>			<u>r6</u>	<u>r6</u>		
6	s5			s4			9	3	

si 移进当前输入符号和状态 i
 rj 按第 j 个产生式进行归约
 acc 接受



LR分析算法：举例



栈	输入	动作
0	id * id + id \$	移进
0 id 5	* id + id \$	按 $F \rightarrow id$ 归约
0 F 3	* id + id \$	按 $T \rightarrow F$ 归约
0 T 2	* id + id \$	

状态	action					goto			
	id	+	*	()	\$	E	T	F
0	s5			s4			1	2	3
1		s6				<i>acc</i>			
2		r2	s7			r2	r2		
3		r4	<u>r4</u>			<u>r4</u>	<u>r4</u>		
4	s5			s4			8	2	3
5		r6	<u>r6</u>			<u>r6</u>	<u>r6</u>		
6	s5			s4			9	3	

si 移进当前输入符号和状态 i
 rj 按第 j 个产生式进行归约
 acc 接受



LR分析算法：举例



栈	输入	动作
0	id * id + id \$	移进
0 id 5	* id + id \$	按 $F \rightarrow id$ 归约
0 F 3	* id + id \$	按 $T \rightarrow F$ 归约
0 T 2	* id + id \$	移进

状态	action					goto			
	id	+	*	()	\$	E	T	F
0	s5			s4			1	2	3
1		s6				<i>acc</i>			
2		r2	s7			r2	r2		
3		r4	<u>r4</u>			<u>r4</u>	<u>r4</u>		
4	s5			s4			8	2	3
5		r6	<u>r6</u>			<u>r6</u>	<u>r6</u>		
6	s5			s4			9	3	

si 移进当前输入符号和状态 i
 rj 按第 j 个产生式进行归约
 acc 接受



LR分析算法：举例



栈	输入	动作
0	id * id + id \$	移进
0 id 5	* id + id \$	按 $F \rightarrow id$ 归约
0 F 3	* id + id \$	按 $T \rightarrow F$ 归约
0 T 2	* id + id \$	移进
0 T 2 * 7	id + id \$	

状态	action					goto			
	id	+	*	()	\$	E	T	F
0	s5			s4			1	2	3
1		s6				<i>acc</i>			
2		r2	s7			r2	r2		
3		r4	<u>r4</u>			<u>r4</u>	<u>r4</u>		
4	s5			s4			8	2	3
5		r6	<u>r6</u>			<u>r6</u>	<u>r6</u>		
6	s5			s4			9	3	

si 移进当前输入符号和状态*i*
rj 按第*j*个产生式进行归约
acc 接受



LR分析算法：举例



栈	输入	动作
0	id * id + id \$	移进
0 id 5	* id + id \$	按 $F \rightarrow id$ 归约
0 F 3	* id + id \$	按 $T \rightarrow F$ 归约
0 T 2	* id + id \$	移进
0 T 2 * 7	id + id \$	移进

状态	action					goto			
	id	+	*	()	\$	E	T	F
0	s5			s4			1	2	3
1		s6				<i>acc</i>			
2		r2	s7			r2	r2		
3		r4	<u>r4</u>			<u>r4</u>	<u>r4</u>		
4	s5			s4			8	2	3
5		r6	<u>r6</u>			<u>r6</u>	<u>r6</u>		
6	s5			s4			9	3	

si 移进当前输入符号和状态 i
 rj 按第 j 个产生式进行归约
 acc 接受



LR分析算法：举例



栈	输入	动作
0	id * id + id \$	移进
0 id 5	* id + id \$	按 $F \rightarrow id$ 归约
0 F 3	* id + id \$	按 $T \rightarrow F$ 归约
0 T 2	* id + id \$	移进
0 T 2 * 7	id + id \$	移进
0 T 2 * 7 id 5	+ id \$	

状态	action					goto			
	id	+	*	()	\$	E	T	F
0	s5			s4			1	2	3
1		s6				<i>acc</i>			
2		r2	s7			r2	r2		
3		r4	<u>r4</u>			<u>r4</u>	<u>r4</u>		
4	s5			s4			8	2	3
5		r6	<u>r6</u>			<u>r6</u>	<u>r6</u>		
6	s5			s4			9	3	

si 移进当前输入符号和状态 i
 rj 按第 j 个产生式进行归约
 acc 接受



LR分析算法：举例



栈	输入	动作
0	id * id + id \$	移进
0 id 5	* id + id \$	按 $F \rightarrow id$ 归约
0 F 3	* id + id \$	按 $T \rightarrow F$ 归约
0 T 2	* id + id \$	移进
0 T 2 * 7	id + id \$	移进
0 T 2 * 7 id 5	+ id \$	按 $F \rightarrow id$ 归约

状态	action					goto			
	id	+	*	()	\$	E	T	F
0	s5			s4			1	2	3
1		s6				<i>acc</i>			
2		r2	s7			r2	r2		
3		r4	<u>r4</u>			<u>r4</u>	<u>r4</u>		
4	s5			s4			8	2	3
5		r6	<u>r6</u>			<u>r6</u>	<u>r6</u>		
6	s5			s4			9	3	

si 移进当前输入符号和状态 i
 rj 按第 j 个产生式进行归约
 acc 接受



LR分析算法：举例



栈	输入	动作
0	id * id + id \$	移进
0 id 5	* id + id \$	按 $F \rightarrow id$ 归约
0 F 3	* id + id \$	按 $T \rightarrow F$ 归约
0 T 2	* id + id \$	移进
0 T 2 * 7	id + id \$	移进
0 T 2 * 7 id 5	+ id \$	按 $F \rightarrow id$ 归约
0 T 2 * 7 F 10	+ id \$	

状态	action					goto			
	id	+	*	()	\$	E	T	F
0	s5			s4			1	2	3
1		s6				<i>acc</i>			
2		r2	s7			r2	r2		
3		r4	<u>r4</u>			<u>r4</u>	<u>r4</u>		
4	s5			s4			8	2	3
5		r6	<u>r6</u>			<u>r6</u>	<u>r6</u>		
6	s5			s4			9	3	

si 移进当前输入符号和状态*i*
rj 按第*j*个产生式进行归约
acc 接受



LR分析算法：举例



栈	输入	动作
0	id * id + id \$	移进
0 id 5	* id + id \$	按 $F \rightarrow id$ 归约
0 F 3	* id + id \$	按 $T \rightarrow F$ 归约
0 T 2	* id + id \$	移进
0 T 2 * 7	id + id \$	移进
0 T 2 * 7 id 5	+ id \$	按 $F \rightarrow id$ 归约
0 T 2 * 7 F 10	+ id \$	按 $T \rightarrow T * F$ 归约

状态	action					goto			
	id	+	*	()	\$	E	T	F
0	s5			s4			1	2	3
1		s6				<i>acc</i>			
2		r2	s7			r2	r2		
3		r4	<u>r4</u>			<u>r4</u>	<u>r4</u>		
4	s5			s4			8	2	3
5		r6	<u>r6</u>			<u>r6</u>	<u>r6</u>		
6	s5			s4				9	3

si 移进当前输入符号和状态*i*
rj 按第*j*个产生式进行归约
acc 接受



LR分析算法：举例



栈	输入	动作
0	id * id + id \$	移进
0 id 5	* id + id \$	按 $F \rightarrow id$ 归约
0 F 3	* id + id \$	按 $T \rightarrow F$ 归约
0 T 2	* id + id \$	移进
0 T 2 * 7	id + id \$	移进
0 T 2 * 7 id 5	+ id \$	按 $F \rightarrow id$ 归约
0 T 2 * 7 F 10	+ id \$	按 $T \rightarrow T * F$ 归约
...

状态	action					goto			
	id	+	*	()	\$	E	T	F
0	s5			s4			1	2	3
1		s6				<i>acc</i>			
2		r2	s7			r2	r2		
3		r4	<u>r4</u>			<u>r4</u>	<u>r4</u>		
4	s5			s4			8	2	3
5		r6	<u>r6</u>			<u>r6</u>	<u>r6</u>		
6	s5			s4				9	3

si 移进当前输入符号和状态 i
 rj 按第 j 个产生式进行归约
 acc 接受



LR分析算法：举例



栈	输入	动作
0	id * id + id \$	移进
0 id 5	* id + id \$	按 $F \rightarrow id$ 归约
0 F 3	* id + id \$	按 $T \rightarrow F$ 归约
0 T 2	* id + id \$	移进
0 T 2 * 7	id + id \$	移进
0 T 2 * 7 id 5	+ id \$	按 $F \rightarrow id$ 归约
0 T 2 * 7 F 10	+ id \$	按 $T \rightarrow T * F$ 归约
...
0 E 1	\$	

状态	action					goto			
	id	+	*	()	\$	E	T	F
0	s5			s4			1	2	3
1		s6				<i>acc</i>			
2		r2	s7			r2	r2		
3		r4	<u>r4</u>			<u>r4</u>	<u>r4</u>		
4	s5			s4			8	2	3
5		r6	<u>r6</u>			<u>r6</u>	<u>r6</u>		
6	s5			s4				9	3

si 移进当前输入符号和状态 i
 rj 按第 j 个产生式进行归约
 acc 接受



LR分析算法：举例



栈	输入	动作
0	id * id + id \$	移进
0 id 5	* id + id \$	按 $F \rightarrow id$ 归约
0 F 3	* id + id \$	按 $T \rightarrow F$ 归约
0 T 2	* id + id \$	移进
0 T 2 * 7	id + id \$	移进
0 T 2 * 7 id 5	+ id \$	按 $F \rightarrow id$ 归约
0 T 2 * 7 F 10	+ id \$	按 $T \rightarrow T * F$ 归约
...
0 E 1	\$	接受

状态	action					goto			
	id	+	*	()	\$	E	T	F
0	s5			s4			1	2	3
1		s6				<i>acc</i>			
2		r2	s7			r2	r2		
3		r4	<u>r4</u>			<u>r4</u>	<u>r4</u>		
4	s5			s4			8	2	3
5		r6	<u>r6</u>			<u>r6</u>	<u>r6</u>		
6	s5			s4				9	3

si 移进当前输入符号和状态 i
 rj 按第 j 个产生式进行归约
 acc 接受



• 关键在于构造LR分析表

- 计算所有可能的状态
 - 每一个状态描述了语法分析过程中所处的位置
 - 可确定正在分析的产生式集合
 - 可确定句柄形成的中间步骤
- 明确状态之前的**跳转关系**
- 明确状态与输入之间对应的**移进或者归约**操作



一起努力 打造国产基础软硬件体系!

李诚

国家高性能计算中心(合肥)、信息与计算机国家级实验教学示范中心

计算机科学与技术学院

2024年09月18日