

## 引论

1. 简要解释编译器的作用以及编译过程的各个阶段。

编译器将程序员用高级编程语言编写的源代码转换成计算机可以理解和执行的机器代码或另一种编程语言。

编译阶段: 词法分析 → 语法分析 → 语义分析 → 中间代码生成 → 代码优化 → 目标代码生成

2. 解释编译器前端和后端的区别，并列举各自的主要任务

前端与源语言相关，负责词法分析、语法分析、语义分析、中间代码生成

后端与目标机器相关，负责代码优化、目标代码生成、存储分配、寄存器分配

3. 什么是语法分析？列举一些常见的语法分析方法。

语法分析是编译过程中的一个阶段，它接在词法分析之后，负责检查源代码中的语句是否符合给定的编程语言的语法规则，并构建出源代码的语法结构

常见分析方法：常见分析方法：自顶向下分析：LL(1)分析法、递归下降分析、预测分析  
自底向上分析：LR(0)分析法、SLR(1)分析法、LR(1)分析法、LALR(1) 分析法

4. 解释词法分析的作用，并说明词法分析器的输入和输出。

将源代码中的字符序列转换成词法单元序列。

输入：源代码文件 字符流

输出：词素(Token)序列，每个词素包含：类型（如标识符、关键字、运算符等）、属性值（如标识符的名字、常量的值等）

5. 什么是中间代码？列举几种常见的中间表示形式。

中间代码是编译器在源代码和目标代码之间生成的一种表示形式

作用：提供源程序的中间表示形式、便于代码优化、提高编译器的 可移植性

常见形式：三地址码、四元式、抽象语法树(AST)、静态单赋值形式(SSA)、P-代码

## 词法分析

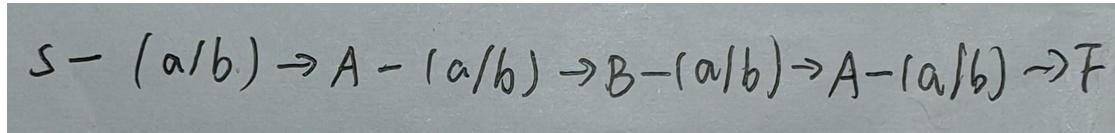
- ① 请写出一个正则表达式，匹配所有由 0 和 1 组成的偶数个字符的字符串。

00|11|01|10)\*(00|11|01|10)?

- ② 给定正则表达式(a|b)(aa|bb)\*(a|b)，请画出对应的非确定有限自动机 (NFA)，并

转换为确定有限自动机 (DFA)，并最小化 DFA；请编程实现从字符串中提取符合该正则表达式的 C 语言程序。

构建 NFA      起始状态：S      接受状态：F  
NFA 可以通过图示表示如下：



③ 给定一个确定有限自动机 (DFA)，如何将其转换为等价的正则表达式？

建立状态转移图      添加开始和结束状态      消除状态（重复直至只剩下开始状态与接受状态）  
                          消除状态

④ 给定一个确定有限自动机 (DFA)，如何判断它是否已经是最小化的状态

### 使用 Myhill-Nerode 等价关系

1. 定义一个关系 R 在状态集合上，使得  $q_i R q_j$  当且仅当对于所有输入符号  $a$   $\delta(q_i, a)$  和  $\delta(q_j, a)$  要么都是接受状态，要么都是非接受状态。
2. 检查 R 是否是等价关系。
3. 对于 RR 中的每个等价类，检查是否在 DFA 中存在一个对应的状态。

## 语法分析

1. 给定文法  $G: S \rightarrow aSb \mid \epsilon$  对于输入串 aaabbb，使用 LR(0) 分析法进行语法分析。

LR(0) 分析表		步骤	状态	导出栈	输入串	动作
1.	$G: S \rightarrow aSb / \epsilon$	1	$S'$	$\epsilon$	aaabbb\$	移入 转移到 2
	$S' \rightarrow S$	2	$S$	$a$	aabbb\$	移入 转移到 2
	$S \rightarrow aSb$	3	$aSb$	$a$	aabb\$	同上
	$GOTO(I_0, a) = \{S' \rightarrow S\}$	4	$S$	$a$	aaa	规约 $S \rightarrow \epsilon$
	$GOTO(I_0, \epsilon) = \{S' \rightarrow S\}$	5	$S$	$a$	aaaS	移入 转移到 5
	$GOTO(I_0, a) = \{S \rightarrow aSb\}$	6	$aSb$	$a$	bbb\$	规约 $S \rightarrow aSb$
	$GOTO(I_0, b) = \{S \rightarrow aSb\}$	7	$aSb$	$aaS$	bb\$	重叠 5.6.5
	$GOTO(I_0, a) = \{S \rightarrow aSb\}$	8	$aSb$	$aaS$	\$	规约 $S \rightarrow aSb$
		9	$S$	$aaS$	accept	

2. 给定文法  $G$ :  $E \rightarrow E+E \mid E*E \mid (E) \mid id$  对于输入串  $id+id*id$ , 使用 LL(1) 分析法进行语法分析。

步骤	符号栈	读入	剩余
1.	\$E .	id	id + id*id \$
2.	\$E T .	id	id + id*id \$ E * T
3.	\$E' id .	id	id + id*id \$ T id
4.	\$E' .	+	id*id \$ 出栈
5.	\$E' T + .	+	id*id \$ E' + T
6.	\$E' T .	id	* id \$ 出栈
7.	\$E' id .	id	* id \$ T id
8.	\$E' .	*	id \$ 出栈
9.	\$E' T * .	*	id \$ E' * T
10.	\$E' T .	id	\$ 出栈
11.	\$E' id .	id	\$ T id
12.	\$E' .	\$	出栈
13.	\$ .	\$	<del>E' → ε</del>

3. 文法  $G$ :  $S \rightarrow aSb \mid T, T \rightarrow cTd \mid \epsilon$  对于输入串 acccdbd, 使用 SLR(1) 分析法 进行语法分析

3. $G: S \rightarrow aSb \mid T, T \rightarrow c \cdot Id \mid \epsilon$				
步骤	状态	待识别	剩余	操作
1.	0		acccdbd\$	移入栈2
2.	02	a	cccdbd\$	移入栈4
3.	024	ac	ccdbd\$	同上
4.	0244	acc	<del>cc</del> dbd\$	同上
5.	02444	acc	<del>cc</del> dbd\$	$T \rightarrow \epsilon$
6.	02447	accT	bd\$	移入栈9
7.	024474	accTd	bd\$	$T \rightarrow cTd$
8.	02447	acct	bd\$	error
		不等		

4. 给定文法  $G: E \rightarrow E + T \mid T, T \rightarrow T * F \mid F, F \rightarrow (E) \mid id$  对于输入串  $id^*$  ( $id+id$ ), 使用 LR(1) 分析法进行语法分析

4. G: E → E + T   T   T → T * F   F, F → (E)   id			状态	符号	剩余	动作
状态	id.	+	*	( ) \$	E T F	
0.	S <sub>5</sub> .	S <sub>4</sub>			, > 3	
1.	S <sub>6</sub> .		acc			
2.	R <sub>2</sub> .	S <sub>7</sub> .	R <sub>2</sub> .			
3.	R <sub>4</sub> .	R <sub>4</sub> .	R <sub>4</sub> .			
4.	S <sub>2</sub> .	S <sub>11</sub> .		8 9 10		
5.	R <sub>6</sub> .	R <sub>6</sub> .	R <sub>6</sub> .			
6.	S <sub>5</sub> .	S <sub>4</sub> .		13 3		
7.	S <sub>5</sub> .	S <sub>4</sub> .		14		
8.	S <sub>6</sub> .		S <sub>15</sub> .			
9.	R <sub>2</sub> .	S <sub>7</sub> .	R <sub>2</sub> .			
10.	R <sub>4</sub> .	R <sub>4</sub> .	R <sub>4</sub> .			
11.	S <sub>12</sub> .	S <sub>11</sub> .		18 9 10		
12.	R <sub>6</sub> .	R <sub>6</sub> .	R <sub>6</sub> .			
13.	R <sub>1</sub> .		R <sub>1</sub> .			
14.	R <sub>3</sub> .	R <sub>3</sub> .	R <sub>3</sub> .			
15.	R <sub>5</sub> .	R <sub>5</sub> .	R <sub>5</sub> .	19 10		
16.	S <sub>12</sub> .	S <sub>11</sub> .			20	
17.	S <sub>12</sub> .	S <sub>11</sub> .				
18.	S <sub>16</sub> .		S <sub>21</sub> .			
19.	R <sub>1</sub> .	S <sub>17</sub> .	R <sub>1</sub> .			
20.	R <sub>3</sub> .	R <sub>3</sub> .	R <sub>3</sub> .			
21.	R <sub>5</sub> .	R <sub>5</sub> .	R <sub>5</sub> .			
1.	0.			id*(id+id)\$	移入	
2.	05.	id.	*	(id+id)\$	T→id	
3.	03.	F.	*	(id+id)\$	T→F	
4.	02.	T.	*	(id+id)\$	移入	
5.	021.	T*	1	(id+id)\$	移入	
6.	0274.	T*	1	(id+id)\$	移入	
7.	027412.	T*	1	(id+id)\$	T→id	
8.	027410.	T*	1	(id+id)\$	T→F	
9.	02749.	T*	c7	+ id\$	E→T	
10.	02748.	T*	(E	+ id)	移入	
11.	0274816.	T*	(E	+ id)	移入	
12.	027481611.	T*	(E	+ id)	T→id	
13.	027481610.	T*	(E+F	1	T→F	
14.	027481619.	T*	(E+F	1	E→E+T	
15.	02748.	T*	(E	1	移入	
16.	0274815.	T*	(E)	\$	F→(E)	
17.	02714.	T*	F	\$	E→T*T	
18.	01.	E.	\$		accept	

5. 给定文法  $G: S \rightarrow aS \mid bS \mid c$  对于输入串 ababcc, 使用 LALR(1)分析法进行语法分析。

步骤	状态	符号栈	输入串	动作
0	0	\$	ababccs	移入 a
1	移入 b	0 2	sababccs	
2	移入 a	0 2 3	sababccs	
3	移入 b	0 2 3 2	sababccs	
4	移入 C	0 2 3 2 3	sababccs	
5	归约 $S \rightarrow C$	0 2 3 2 3 4	sababc	cs
6	移入 C	0 2 3 2 3 1	\$ababs	cs
7	归约 $S \rightarrow C$	0 2 3 2 3 1 4	\$ababSc	\$
8	归约 $S \rightarrow bs$	0 2 3 2 3 1	\$ababs	\$
9	归约 $S \rightarrow aS$	0 2 3 2 1	sabas	\$
10	归约 $S \rightarrow bs$	0 2 3 1	sabs	\$
11	归约 $S \rightarrow aS$	0 2 1	sas	\$
12	接受	0 1	\$	

6. 给定文法  $G: S \rightarrow aSb \mid cSc \mid \epsilon$  对于输入串 acbcc, 使用 LL(1)分析法进行语法分析。

6.  $G: S \rightarrow aSb \mid aSc \cdot f_S$

$\text{First}(S) = \{a, c\}$ ,  $\text{Follow}(S) = \{\$, b, c\}$

$\text{First}(aSb)$ ,  $\text{First}(aSc)$ ,  $\text{First}(\epsilon)$  互不相交

$\epsilon \in \text{First}(S)$

$\text{First}(S) \cap \text{Follow}(S) = \{c\}$

即 PG 不得含 LL(1)

7. 给定文法  $G: E \rightarrow E+E \mid E*E \mid (E) \mid \text{id}$  对于输入串  $\text{id}^*(\text{id}+\text{id})$ , 使用 SLR(1) 分析法进行语法分析。

步骤	状态	符号栈	输入串	动作
0	0	\$	aabccs	移入 a
1	0 2	sa	abccs	移入 a
2	0 2 2	saa	bccs	移入 b
3	0 2 2 3	saab	ccs	移入 C
4	0 2 2 3 4	saabc	cs	移入 C
5	0 2 2 3 4 4	\$aabcc	\$	归约 $S \rightarrow cC$
6	0 2 2 3 1	saabs		归约 $S \rightarrow bS$
7	0 2 2 1	saas	\$	归约 $S \rightarrow aS$
8	0 2 1	\$as		归约 $S \rightarrow aS$
9	0 1	\$		接受

8. 给定文法  $G: S \rightarrow aS \mid bS \mid c$  对于输入串 ababcc, 使用 LR(1) 分析法进行语法分析。

6. $G: S \rightarrow aSb \mid aSc \cdot fS$				
$\text{First}(S) = \{a, \epsilon\}$ , $\text{Follow}(S) = \{\$, b, c\}$				
$\text{First}(aSb) = \text{First}(aSc) = \text{First}(\epsilon) \subseteq \text{Follow}(S)$				
$\epsilon \in \text{First}(S)$				
$\text{First}(S) \cap \text{Follow}(S) = \{c\}$				
即 G 不得合 LL(1)：				
步驟	狀態	符號	剩餘	動作
1.	0	ababa\$		移入
2.	02	a.	babcc\$	
3.	027	ab.	abcc\$	
4.	0276	aba.	bcc\$	
5.	02767	abab.	cc\$	
6.	027678	ababc.	c\$	$\text{移入 } S \rightarrow c$
7.	027679	abobs.	c\$	$\text{移入 } S \rightarrow bs$
$\text{Gr: } S \rightarrow aSb \mid aSc \mid \epsilon$ 10) $S' \rightarrow S$ . 11) $S \rightarrow aS$ . 12) $S \rightarrow bS$ . 13) $S \rightarrow c$ . $\text{First}(S) = \text{First}(S') = \{a, b, c\}$ . $\text{Follow}(S') = \text{Follow}(S) = \{\$\}$ . ACTION GoTo 狀態 $\alpha \quad b \quad c \quad \$$ $S$ 0. $S_2 \quad S_3 \quad S_4$ $\downarrow$ 1. $\downarrow$ 2. $S_6 \quad S_7 \quad S_8$ $\downarrow$ 3. $S_6 \quad b_7 \quad S_8$ $\downarrow$ 4. $\downarrow$ 5. $\downarrow$ 6. $S_6 \quad S_7$ $\downarrow$ 7. $S_6 \quad S_7$ $\downarrow$ 8. $r_3 \quad r_3 \quad r_3$ 9. $r_2 \quad r_2$ 10. $r_1 \quad r_1 \quad r_1$ 11. $r_2 \quad r_2 \quad r_2$				

9. 给定文法  $G: E \rightarrow E+T \mid T, T \rightarrow T^* F \mid F, F \rightarrow (E) \mid \text{id}$  对于输入串  $\text{id}^* (\text{id}+\text{id})$ ,

使

用 LALR(1)分析法进行语法分析。

步骤	状态	符号栈	输入串	动作
0	0	\$	abcccc\$	移入 a
1	02	\$a	bcccc\$	移入 b
2	023	\$ab	cccc\$	移入 c
3	0234	\$abc	cc\$	移入 c
4	02344	\$abcc	c\$	移入 c
5	023444	\$abccc		移入 c
6	0234444	\$abcccc	\$	
7	0231	\$abS	\$	
8	021	\$aS	\$	
9	01	\$S	\$	accept

10. 给定文法  $G: S \rightarrow aSb \mid T, T \rightarrow cTd \mid \epsilon$  对于输入串 acccdbd, 使用 LL(1)分析法进行语法分析。

10.  $\text{First}(S) = \{a, \epsilon\}$ ,  $\text{First}(T) = \{c, \epsilon\}$ .  
 $\text{Follow}(S) = \{b, \$ \}$ ,  $\text{Follow}(T) = \{b, d, \$ \}$ .  
 $\text{First}(a \notin b) \cap \text{First}(T) = \emptyset$ .  $\text{First}(cTd) \cap \text{First}(\epsilon) = \emptyset$ .  
 $\epsilon \in \text{First}(S)$ ,  $\epsilon \in \text{First}(T)$ ,  $\text{First}(S) \cap \text{Follow}(S) = \emptyset$ .  
 $\text{First}(T) \cap \text{Follow}(T) = \emptyset$ .  $G$  是 LL(1) 文法.  
 a      b,      c      d       $\epsilon - \$$ .

$S \xrightarrow{} S \rightarrow aSb$

$S \xrightarrow{} T$

$S \xrightarrow{} T$ .

$T \xrightarrow{} T \rightarrow \epsilon$   $T \xrightarrow{} aTd$ ,  $T \xrightarrow{} S$   $T \xrightarrow{} \epsilon$   $T \xrightarrow{} \epsilon$ .

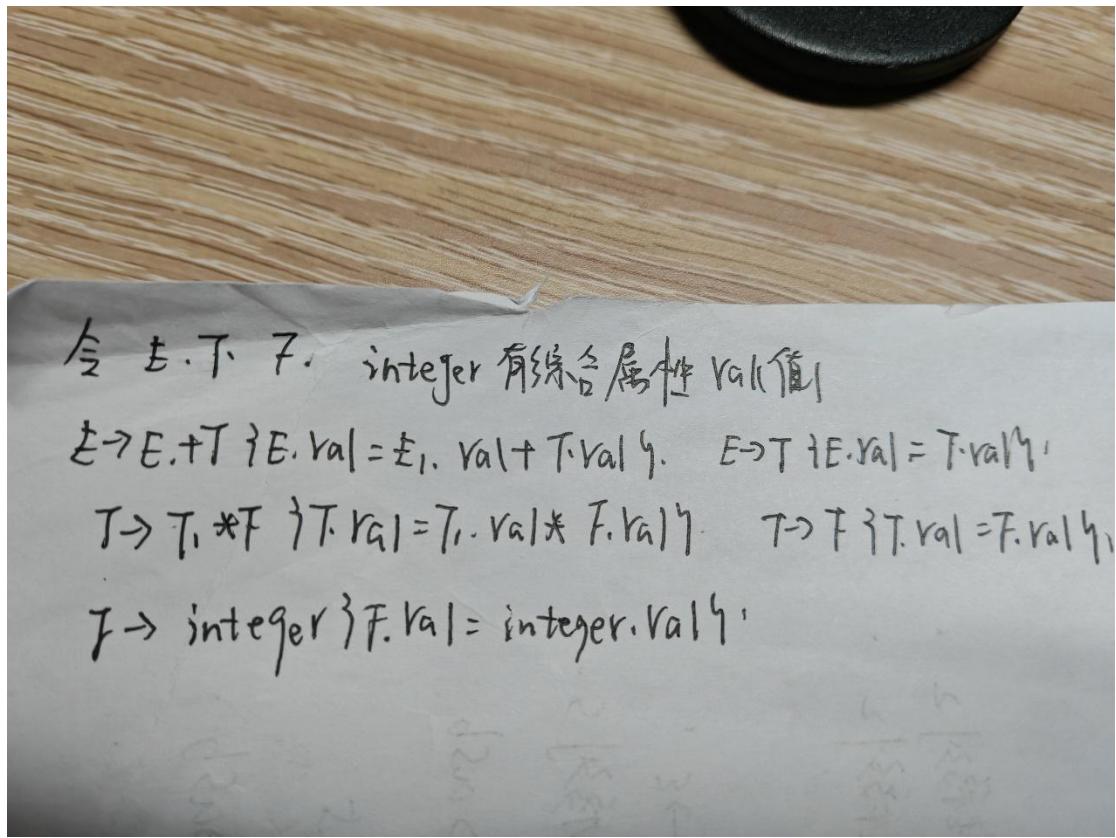
符号.      读入.      判断.      规则.

1. $\$S$	a.	cccdbd\$	
2. $\$bSa$	a	cccdbd\$	$S \rightarrow aSb$
3. $\$bs$	c.	cccdbd\$	出栈
4. $\$bT$	c.	cccdbd\$	$S \rightarrow T$
5. $\$bbTc$	c.	cccdbd\$	$T \xrightarrow{} cTd$
6. $\$bdT$	c.	cccdbd\$	出栈
7. $bddTc$	c.	cccdbd\$	$T \xrightarrow{} cTd$
8. $\$bddT$	c.	cccdbd\$	出栈
9. $\$bdddTc$	c.	cccdbd\$	$T \xrightarrow{} cTd$
10. $\$bdddT$	d.	cccdbd\$	出栈
11. $\$ bdd$	d.	cccdbd\$	$T \xrightarrow{} \epsilon$
12. $\$ bdd$	b.	cccdbd\$	出栈

不存在于栈底.      不得文法  $G$ .

## 语义分析

习题 1: 给定文法:  $E \rightarrow E + TE \rightarrow TT \rightarrow T * FT \rightarrow FF \rightarrow \text{integer}$  设计一个语法制导翻译方案, 计算表达式的值。



习题 2: 给定文法:  $\text{stmt} \rightarrow \text{id} = E$

$E \rightarrow \text{id}$

$E \rightarrow \text{int}$

设计一个语法制导翻译方案, 检查赋值语句中的类型匹配

DV.V

stmt 有综合属性 val, 其值 true 表示匹配成功, false 表示失败.

E.id 有综合属性 type 表示数据属性.

stmt  $\rightarrow$  id = E {if. id.type = E.type then stmt.val = true,  
else stmt.val = false}.

$E \rightarrow id \quad \{E.type = id.type\}$ .

$E \rightarrow int \quad \{E.type = int\}$ .

