

编译原理 期中考试

一、

1. 正则表达式在编译实现中有什么作用？

正则表达式可以用于定义token类，在词法分析中，让输入的字符流匹配正则表达式，以识别出不同类型的token。

2. 怎么区分NFA和DFA？

DFA

- 每个状态在每次输入单个字符时，只有一种转移方式。
- 一条边的标号不能是 ϵ 。
- 从初始状态到结束状态只有一条路径。

NFA

- 每个状态在每次输入单个字符时，有多种转移方式。
- 一条边的标号可以是 ϵ 。
- 从初始状态到结束状态有多条路径。

3. 什么是二义性文法，怎么消除？

二义性文法：一个文法可以为某个句子生成多颗语法分析树，即对同一个句子有多个最左推导或者最右推导的文法。

消除：指定优先级、指定结合性。

4. 为什么在语法解析中要使用 FIRST 和 FOLLOW 集？

FIRST集：可以通过下一个输入的语法符号属于哪一个FIRST集，来判断选择哪一个产生式。

FOLLOW集：如果一个产生式推导出 ϵ ，则通过FOLLOW集判断下一个输入的语法符号所对应的产生式。

5. 从分析能力上看，为什么说 LR 比 LL 要强？

因为LR可以分析左递归和左公共因子的语法。

6. LR(0)项目和 LR(1)项目的区别是什么？

LR(0)项是一个产生式，其中的RHS中有一个" \cdot "。

LR(1)项除了和LR(0)项一样的产生式之外，还有一个lookahead的终结符a，表示当下一个输入的符号是a时，根据该产生式进行归约。

7. 为什么 LALR 在实际中应用的比较广泛？

因为它形式上和LR(1)相同（较精确），但是状态数目和SLR(1)和LR(0)相同（节省空间）。

8. SDD 和 SDT 的联系和区别分别是什么？

SDD是语法制导定义，是上下文无关文法和属性及规则的结合，其中规则并没有关联到产生式中的特定位置。

SDT是语法制导翻译方案，是SDD的补充，具体翻译实施方案。其中的规则成为“动作”，插入在产生式RHS的特定位置中。

9. 怎么在语法解析过程中实现 L-SDD 文法的语义分析？

首先，将L-SDD转化为SDT：

- 将计算某个非终结符A的继承属性的动作插入到产生式右部中紧靠在A的本次出现之前的位置上。
- 将计算一个产生式左部符号的综合属性的动作放在这个产生式右部的末尾。

然后就可以用LL或者LR（需要引入Marker）分析。

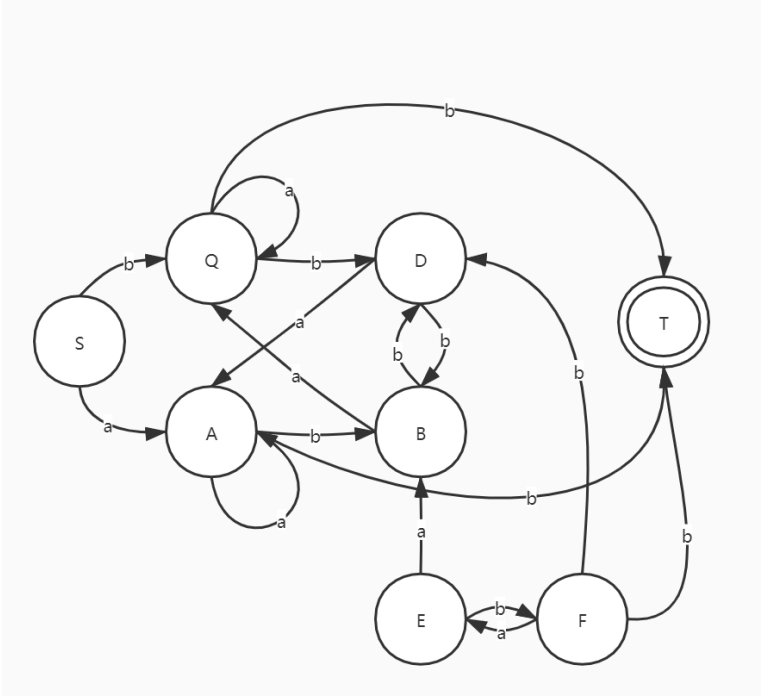
10. 符号表 (Symbol Table) 的主要作用是什么？

符号表用于记录每个符号（标识符）的信息，在代码生成阶段使用。

二、

1.

- 把文法转换为NFA，得到：



- 把NFA转换为DFA，得到：

NFA状态	DFA状态	a	b
{S}	S ₀	S ₁ = {A}	S ₂ = {Q}
{A}	S ₁	S ₁ = {A}	S ₃ = {B, T}
{Q}	S ₂	S ₂ = {Q}	S ₄ = {D, T}
{B, T}	S ₃	S ₂ = {Q}	S ₅ = {D}
{D, T}	S ₄	S ₁ = {A}	S ₆ = {B}
{D}	S ₅	S ₁ = {A}	S ₆ = {B}
{B}	S ₆	S ₂ = {Q}	S ₅ = {D}

- 简化
 - 把状态分为接受状态和非接受状态两组：
第一组：

DFA状态	a	b
S ₀	S ₁	S ₂
S ₁	S ₁	S ₃
S ₂	S ₂	S ₄
S ₅	S ₁	S ₆
S ₆	S ₂	S ₅

第二组：

DFA状态	a	b
S ₃	S ₂	S ₅
S ₄	S ₁	S ₆

- 第一组可以再分为：

第三组：

DFA状态	a	b
S ₀	S ₁	S ₂
S ₅	S ₁	S ₆
S ₆	S ₂	S ₅

第四组：

DFA状态	a	b
S ₁	S ₁	S ₃
S ₂	S ₂	S ₄

- 第三组可以再分为：

第五组：

DFA状态	a	b
S ₅	S ₁	S ₆
S ₆	S ₂	S ₅

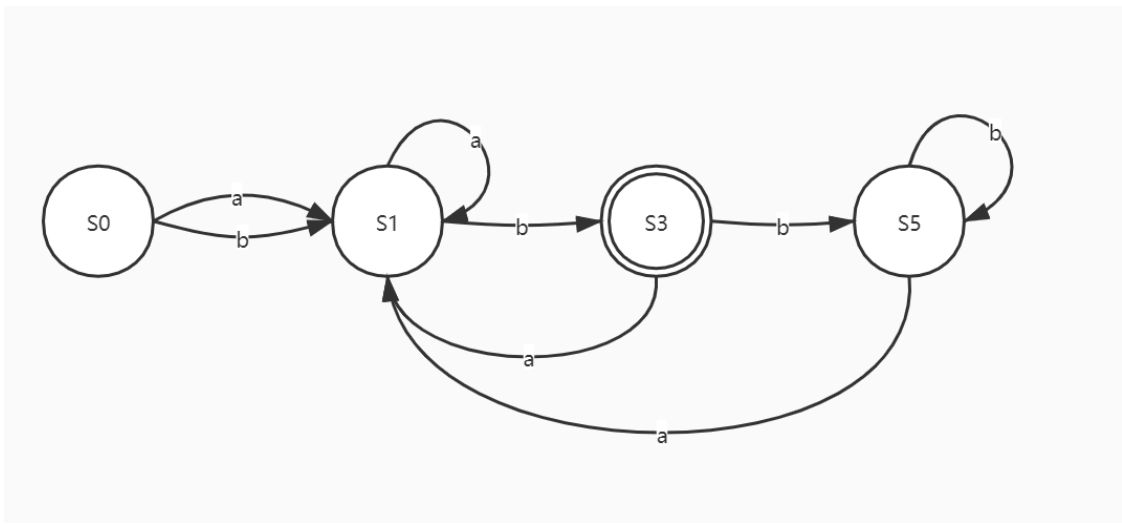
第六组：

DFA状态	a	b
S ₀	S ₁	S ₂

- 最后得到第二、四、五、六组，在每组中选取代表状态，最终得到：

DFA状态	a	b
S ₀	S ₁	S ₁
S ₁	S ₁	S ₃
S ₃	S ₁	S ₅
S ₅	S ₁	S ₅

- 图示如下：



2.

(1)

$$\begin{aligned}
 S &\Rightarrow (T) \\
 &\Rightarrow (T, S) \\
 &\Rightarrow (S, S) \\
 &\Rightarrow (a, S) \\
 &\Rightarrow (a, (T)) \\
 &\Rightarrow (a, (T, S)) \\
 &\Rightarrow (a, (S, S)) \\
 &\Rightarrow (a, (a, S)) \\
 &\Rightarrow (a, (a, a))
 \end{aligned}$$

$$\begin{aligned}
 S &\Rightarrow (T) \\
 &\Rightarrow (T, S) \\
 &\Rightarrow (S, S) \\
 &\Rightarrow ((T), S) \\
 &\Rightarrow ((T, S), S) \\
 &\Rightarrow ((T, S, S), S) \\
 &\Rightarrow ((S, S, S), S) \\
 &\Rightarrow (((T), S, S), S) \\
 &\Rightarrow (((T, S), S, S), S) \\
 &\Rightarrow (((S, S), S, S), S) \\
 &\Rightarrow (((a, S), S, S), S) \\
 &\Rightarrow (((a, a), S, S), S) \\
 &\Rightarrow (((a, a), ^, S), S) \\
 &\Rightarrow (((a, a), ^, (T)), S) \\
 &\Rightarrow (((a, a), ^, (S)), S) \\
 &\Rightarrow (((a, a), ^, (a)), S) \\
 &\Rightarrow (((a, a), ^, (a)), a)
 \end{aligned}$$

(2)

- 修改文法

对 $T \rightarrow T, S \mid S$, 消除左递归得:

$T \rightarrow ST'$

$T' \rightarrow, ST' \mid \epsilon$

故LL(1)文法为:

1. $S \rightarrow a$
2. $S \rightarrow ^$
3. $S \rightarrow (T)$
4. $T \rightarrow ST'$
5. $T' \rightarrow, ST'$
6. $T' \rightarrow \epsilon$

- 构建预测分析表

首先计算FIRST集和FOLLOW集：

$$FIRST(S) = \{a, ^, (\}$$

$$FIRST(T) = \{a, ^, (\}$$

$$FIRST(T') = \{, , \epsilon\}$$

$$FOLLOW(S) = \{\$, , ,)\}$$

$$FOLLOW(T) = \{)\}$$

$$FOLLOW(T') = \{)\}$$

则有：

$$FIRST(a) = \{a\}$$

$$FIRST(^) = \{^ \}$$

$$FIRST((T)) = \{(\}$$

$$FIRST(ST') = \{a, ^, (\}$$

$$FIRST(, ST') = \{, \}$$

$$FIRST(\epsilon) = \{\epsilon\}$$

构建预测分析表如下：

- 先不考虑 ϵ ，得到：

	a	\wedge	$($	$)$	$,$	$\$$
S	$S \rightarrow a$	$S \rightarrow \wedge$	$S \rightarrow (T)$			
T	$T \rightarrow ST'$	$T \rightarrow ST'$	$T \rightarrow ST'$			
T'					$T' \rightarrow , ST'$	

- 再考虑 ϵ ，得到：

	a	\wedge	$($	$)$	$,$	$\$$
S	$S \rightarrow a$	$S \rightarrow \wedge$	$S \rightarrow (T)$			
T	$T \rightarrow ST'$	$T \rightarrow ST'$	$T \rightarrow ST'$			
T'				$T' \rightarrow \epsilon$	$T' \rightarrow , ST'$	

- 证明是LL(1)

- 对于任意两个不同的产生式 $A \rightarrow \alpha | \beta$ ，均满足：

- $FIRST(\alpha) \cap FIRST(\beta) = \phi$
- 若 ϵ 在 $FIRST(\beta)$ 中，则 $FIRST(\alpha) \cap FOLLOW(A) = \phi$

(3)

已匹配	栈	输入	动作
	$S\$$	$(a, a)\$$	
	$(T)\$$	$(a, a)\$$	输出 $S \rightarrow (T)$
($T)\$$	$a, a)\$$	匹配(
($ST')\$$	$a, a)\$$	输出 $T \rightarrow ST'$
($aT')\$$	$a, a)\$$	输出 $S \rightarrow a$
(a	$T')\$$	$, a)\$$	匹配a
(a	$, ST')\$$	$, a)\$$	输出 $T' \rightarrow, ST'$
(a,	$ST')\$$	$a)\$$	匹配,
(a,	$aT')\$$	$a)\$$	输出 $S \rightarrow a$
(a, a	$T')\$$	$)\$$	匹配a
(a, a	$)\$$	$)\$$	输出 $T' \rightarrow \epsilon$
(a, a)	$\$$	$\$$	匹配)

分析得到 (a, a) 符合新的文法。

3.

- 首先引入增广文法，得到：

- $A' \rightarrow A$
- $A \rightarrow aAd$
- $A \rightarrow aAb$
- $A \rightarrow \epsilon$

- 对于上述文法，计算LR(0)项集组。

$$I_0 = CLOSURE(\{A' \rightarrow \cdot A\}) = \{A' \rightarrow \cdot A, A \rightarrow \cdot aAd, A \rightarrow \cdot aAb, A \rightarrow \cdot\}$$

$$I_1 = GOTO(I_0, A) = CLOSURE(\{A' \rightarrow A \cdot\}) = \{A' \rightarrow A \cdot\}$$

$$I_2 = GOTO(I_0, a) = CLOSURE(\{A \rightarrow a \cdot Ad, A \rightarrow a \cdot Ab\}) = \{A \rightarrow a \cdot Ad, A \rightarrow a \cdot Ab, A \rightarrow \cdot aAd, A \rightarrow \cdot aAb, A \rightarrow \cdot\}$$

$$I_3 = GOTO(I_2, A) = CLOSURE(\{A \rightarrow aA \cdot d, A \rightarrow aA \cdot b\}) = \{A \rightarrow aA \cdot d, A \rightarrow aA \cdot b\}$$

$$GOTO(I_2, a) = CLOSURE(\{A \rightarrow a \cdot Ad, A \rightarrow a \cdot Ab\}) = I_2$$

$$I_4 = GOTO(I_3, d) = CLOSURE(\{A \rightarrow aAd \cdot\}) = \{A \rightarrow aAd \cdot\}$$

$$I_5 = GOTO(I_3, b) = CLOSURE(\{A \rightarrow aAb \cdot\}) = \{A \rightarrow aAb \cdot\}$$

- 计算FOLLOW集：

$$FOLLOW(A) = \{\$, d, b\}$$

- 构造分析表

	ACTION				GOTO
状态	a	b	d	\$	A
0	s2	r4	r4	r4	1
1				acc	
2	s2	r4	r4	r4	3
3		s5	s4		
4		r2	r2	r2	
5		r3	r3	r3	

构造过程中没有产生任何冲突动作，因此这个文法是SLR(1)的。

- 分析 $ab\#$

栈	符号	输入	动作
0		$ab\#$	移入
0 2	a	$b\#$	根据 $A \rightarrow \epsilon$ 归约
0 2 3	aA	$b\#$	移入
0 2 3 5	aAb	$\#$	根据 $A \rightarrow aAb$ 归约
0 1	A'	$\#$	acc