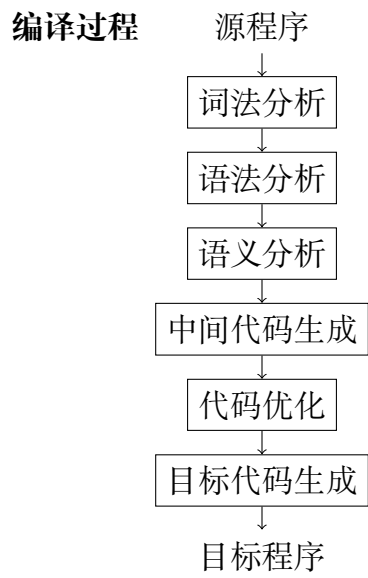


# 第一章 引论





## 第二章 语法分析

### 2.1 文法

终结字符, 非终结字符

规则, 重写规则, 产生式, 生成式  $\alpha \rightarrow \beta, \alpha ::= \beta$

文法  $G = (V_N, V_T, P, S)$ ,  $V_N$ : 非终结符,  $V_T$ : 终结符,  $P$ : 规则集合,  $S$ : 识别符, 开始符, 一种文法的开始

推导, 归约 对于  $a \rightarrow b \rightarrow c$ ,  $a$  直接推导  $b$ ,  $b$  直接归约到  $a$  ( $a \Rightarrow b$ ),  $a$  推导  $c$ ,  $c$  归约到  $a$  ( $a \xRightarrow{*/+} b$ )

句型, 句子  $S \xRightarrow{*} x$ ,  $x$  为句型, 若  $x$  为终结符组成称为句子

例 2.1.1 已知语言  $L(G) = a^n b^m c^m | n \geq 1, m \geq 0$ , 求  $G[S]$

解  $S \rightarrow AB$

$$A \rightarrow aA | a$$

$$B \rightarrow bBc | \epsilon$$

注意:  $S$  写最前面

#### 2.1.1 文法类型

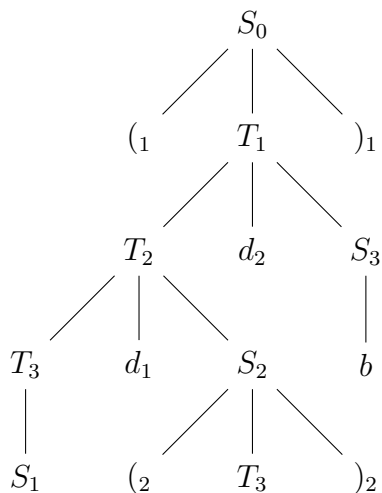
- 0 型文法: 全都是
- 1 型 (上下文有关) 文法:  $aBc \rightarrow aDcc, |aBc| \leq |aDcc|$  or  $A \rightarrow \epsilon$ ,  
( $|a| \rightarrow \text{length } a$ )

- 2 型 (上下文无关) 文法:  $a \rightarrow bCd$ , 其中  $a$  为单个终结符
- 3 型 (正规) 文法:  $A \rightarrow \epsilon|a|aB$

### 2.1.2 句型分析

**最右推导 (规范推导)** 优先展开最右非终结符, 推出的句型称为**右 (规范) 句型**

对于右句型  $(Sd(T)db)$ ,



**短语**  $S_1, ({}_2T_3)_2, S_1d_1({}_2T_3)_2, b, S_1d_1({}_2T_3)_2d_2b, ({}_1S_1d_1({}_2T_3)_2d_2b)_1$

**直接短语, 简单短语**  $S_1, ({}_2T_3)_2, b$

**句柄 (最左直接短语, 仅在右句型中适用)**  $S_1$

## 2.2 自顶向下

**FIRST** 产生的句子的可能的第一个符号

- $FIRST(\epsilon) = \{\epsilon\}$
- $FIRST(a \dots) = \{a\}$
- $FIRST(A \dots) = FIRST(A), A \rightarrow \epsilon$

**FOLLOW** 符号的下一个可能符号, 末尾为 #

- $FOLLOW(S) = \# \cup \dots$

**SELECT** 读取下一个符号可推导

**LL(1)** 从左到右 (Left), 最左推导 (Left derivation), 向后读取一个符号

**LL(1) 判别方式**  $SELECT(A \rightarrow X) \cap SELECT(A \rightarrow Y) = \emptyset$

### 2.2.1 常见非 LL(1)

1. 左公因子  $A \rightarrow aX|aY$  转化为  $A \rightarrow aA', A' \rightarrow X|Y$
2. 左递归  $A \rightarrow Aa|b|c$  转化为右递归  $A \rightarrow bA'|cA', A' \rightarrow aA$

### 2.2.2 例题

P93 4.5.2

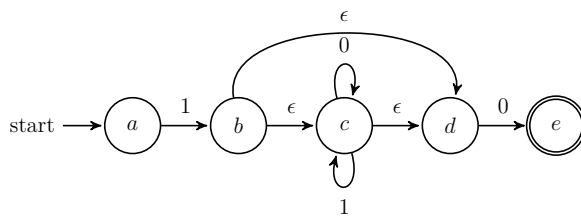
## 2.3 自底向上



## 第三章 词法分析

**正规式**  $1(0|1)^*0$

**NFA** 不确定的有穷自动机



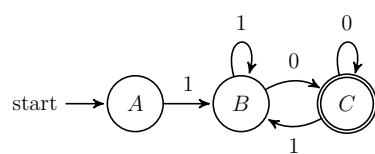
**NFA 转 DFA**

状态/输入	0	1
A {a}	-	B {b, c, d}
B {b, c, d}	$\epsilon\text{-closure}(c, e) = \mathbf{C} \{c, d, e\}$	D {c, d}
<b>C</b> {c, d, e}	<b>C</b>	D
D {c, d}	<b>C</b>	D

**DFA** 确定的有穷自动机

**DFA 化简 (确定法, 最小化)**  $\{\{A, B, D\}, \{C\}\}$ , 若状态任意输入后下一个状态在其他组则独立出来, 最后同组内合并

状态/输入	0	1
A	-	B
B	<b>C</b>	B
<b>C</b>	<b>C</b>	B





## 第四章 语义分析

**综合属性** 自底向上传播

**继承属性** 自顶向下传播



## 第五章 中间代码生成

- AST: 抽象语法树
- TAC: 三地址码, 四元式

### 5.1 考试用汇编

**JEZ** *c \_ addr* 如果 *c* 为假跳转到 *addr*, 逆波兰:  $c(addr)(jez)$

**JMP** *\_ \_ addr* 跳转到 *addr*, 逆波兰:  $(addr)(jez)$

```
if c:           # 0x01: JEZ c _ 0x05
    x           # 0x02 - 0x03: x
else           # 0x04: JMP _ _ 0x09
    y           # 0x05 - 0x08: y
               # 0x09:
```

```
while c:        # 0x01: JEZ c _ 0x05
    x           # 0x02 - 0x03: x
               # 0x04: JMP _ _ 0x01
               # 0x05
```

**例 5.1.1** 完整例题

**解**

```
while x + y > 3:
```

```
a = c + 3 * b
b = a + e - f * e
```

**逆波兰**  $xy+3>(26)(jez)ac3b+=bae+fe-=1(jmp)$

```
01: (+, x, y, T1)
    (>, T1, 3, T2)
    (jez, T2, _, 12)
    (*, 3, b, T3)
05: (+, c, T3, T4)
    (=, T4, _, a)
    (+, a, e, T5)
    (*, f, e, T6)
    (-, T5, T6, T7)
10: (=, T7, _, b)
11: (jmp, _, _, 1)
12:
```

## 第六章 中间代码优化

TAC 转 DAG P273



# 索引

产生式, 3

词法分析, 1

代码优化, 1

非终结字符, 3

归约, 3

规则, 3

句型, 3

句子, 3

目标代码生成, 1

生成式, 3

推导, 3

文法, 3

语法分析, 1

语义分析, 1

中间代码生成, 1

终结字符, 3

重写规则, 3