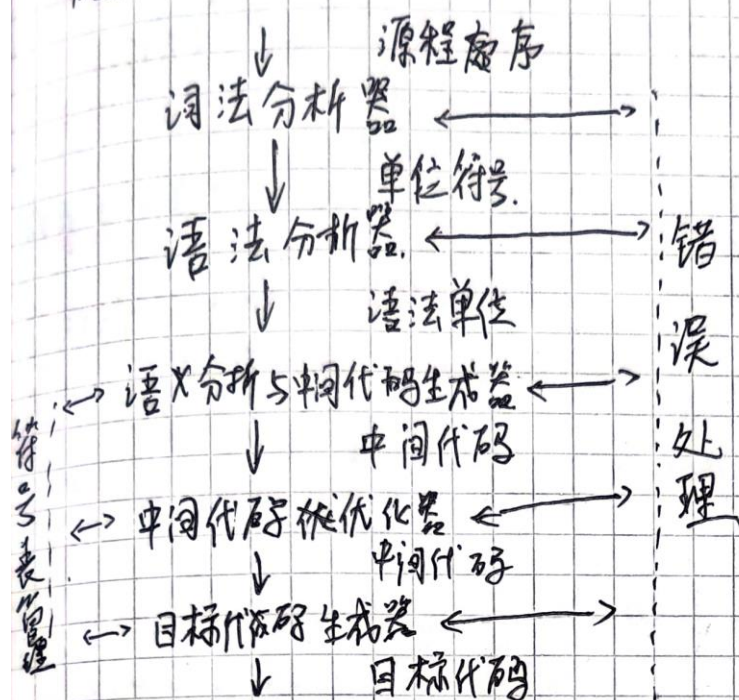


一、简答题

1. 画图表示编译的各个过程并说明各阶段的功能。



词法分析器：又称扫描器，将输入源程序进行词法分析，输出单词符号。

语法分析器：又称扫描器对单词符号进行语法分析，识别出各类语法单位，最终判断输入串是否构成语法上正确的程序。

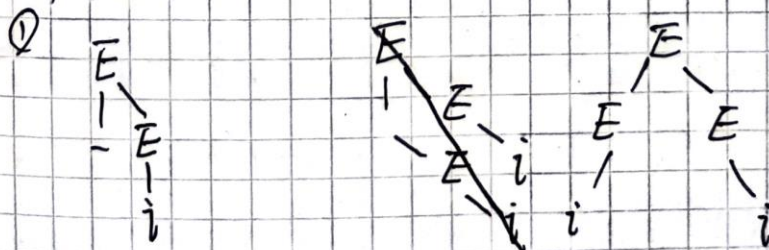
语义分析与中间代码生成器：语义按照语法规则对语法分析器识别(或推导出的)语法单位进行语义分析，并把它翻译成一定形式的中间代码。

中间代码优化器：对中间代码进行优化处理。

目标代码生成器：把中间代码翻译成目标代码。

2. 判断文法 $G[S]: E \rightarrow EE | -E | i$ 是否为二义文法，并说明理由。

是二义的，产生式： $E \rightarrow EE$ $E \rightarrow -E$ $E \rightarrow i$



存在两种不同的语法树，体现出文法二义性。

张翼龙 20210013 0226

3. 对于文法 $G[E]: E \rightarrow F > E | F, F \rightarrow x | y | z$, 写出句型 $x > y > z$ 的所有短语语法和句柄。

$G[E]: E \rightarrow F > E | F$
 $F \rightarrow x | y | z$

推导 $E \Rightarrow F > E \Rightarrow x > E \Rightarrow x > F > E \Rightarrow x > y > E \Rightarrow x > y > F \Rightarrow x > y > z$

短语: ~~$F \rightarrow x$~~ ~~$F \rightarrow y$~~ ~~$F \rightarrow z$~~

① x ② y ③ z ④ $x > y$ ⑤ $y > z$ ⑥ $x > y > z$

句柄 ① z ② $y > z$ ③ $x > y > z$

4. 消除文法 $G[S]$ 的左递归: $S \rightarrow Aa | b, A \rightarrow S | t$

首先消除 A 左递归: $A \rightarrow tA', A' \rightarrow SA' | \epsilon$

$\therefore S \rightarrow tA'a | b$

最终: $S \rightarrow tA'a | b, A' \rightarrow SA' | \epsilon$

5. 写下逆波兰式: $a^{\wedge} - (b - c^{\wedge}d)$ 其中 \wedge 为幂运算符。

① $c^{\wedge} \rightarrow cd^{\wedge}$ ② $b - c^{\wedge} \rightarrow bcd^{\wedge} -$

③ $a - (b - c^{\wedge}d) \rightarrow abcd^{\wedge} - -^{\wedge}$

6. 文法 $G[S]: S \rightarrow (S) | i$, 试确定文法对应的语言

$S \rightarrow (S), S \rightarrow i$

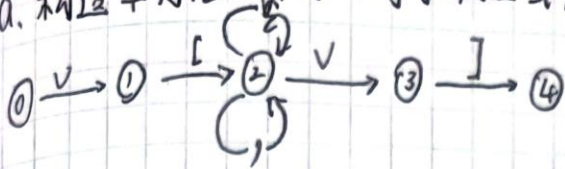
$S \rightarrow (S)$ 文法允许递归地在括号内嵌套别的结构

$S \rightarrow Si$ 表示 S 可直接替换为 i

$L(G) = \{(L(a))^n i (L(a))^n | n \geq 0\}$ 例 $i, (i), ((i)), \dots$ 页2

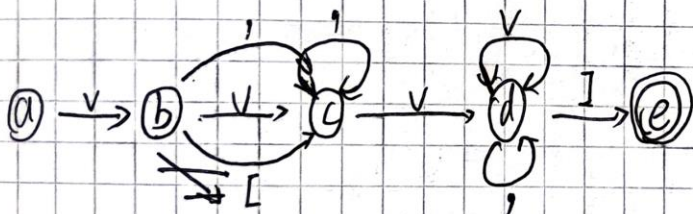
二 数组引用的正规式: $V[(V,)^*V]$, 其中 V 为变量名或数组名, 中括号是数组下标符号, 小括号是元语言符号, 不是文法符号

a. 构造单字符 NFA (每条弧上式为单个字符, 或为 ϵ)



b. 确定化

状态	输入 V	输入 [输入 ,	输入]
{0}	{1, 2}			
{1, 2}	{2, 3}	{2, 3}	{2, 3}	
{2, 3}	{2, 3, 4}		{2, 3}	
{2, 3, 4}	{2, 3, 4}		{2, 3, 4}	{4}
{4}				



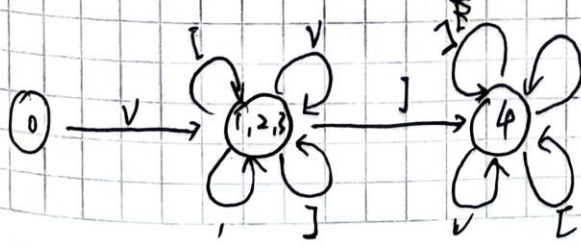
c. 最小化

~~{1, 2} 与 {2, 3} 等价~~

{1, 2} 与 {2, 3} 等价

{2, 3, 4} 与 {4} 有相似性, 可合并

	V	[,]
{0}	{1, 2, 3}			
{1, 2, 3}	{2, 3, 4}		{2, 3, 4}	
{2, 3, 4}	{4}		{4}	
{4}				



张疆龙

202100130226

Date

三. C风格声明语法G[D], 产生式如下, 其中id为变量名:

$D \rightarrow TidV$ $V \rightarrow ,idV$ $V \rightarrow \epsilon$ $T \rightarrow int$ $T \rightarrow double$

a. 求所有非终结符号的 First 集合.

非终结符: D, V, T

$First(D) = First(T) = \{int, double\}$

$First(V) = \{, , \epsilon\}$

$First(T) = \{int, double\}$

b. 求所有非终结符号的 Follow 集合 (紧跟 A 后面的终结符或 #)

D 为起始 \Rightarrow 则 D 中包含 #

$Follow(D) = \{ \# \}$

$Follow(V) = \{ \# , \epsilon \}$

$Follow(T) = \{ int, double \}$

c. int double , id #

D $D \rightarrow TidV$ $D \rightarrow TidV$

V $V \rightarrow ,idV$ $V \rightarrow \epsilon$ $V \rightarrow \epsilon$

T $T \rightarrow int$ $T \rightarrow double$

当输入为 int/double 选择 $D \rightarrow TidV$

由于 V 可推导出 ϵ 且属于 $Follow(V)$ \therefore 当输入 # 或 #id 时 选择 $V \rightarrow \epsilon$

题中规则在 V 处有 ", " 输入 可以选 $V \rightarrow ,idV$ 与 $V \rightarrow \epsilon$

d. 写出句子 int i, sum 的归约过程

张翼龙

20210013 0226

Date

三.d. 写出句子 $\text{int } i, \text{sum}$ 的识别过程;

① 输入串: $\text{int } i, \text{sum} \#$

栈: $D \#$

② 输入串: $\text{int } i, \text{sum} \#$

栈: $T \text{ id } V$

③ 输入: $\text{int } i, \text{sum} \#$

栈: $\text{int id } V \#$

④ 输入 $i, \text{sum} \#$

栈: $\text{id } V \#$

⑤ 输入: $, \text{sum} \#$

栈: $V \#$

⑥ 输入: $, \text{sum} \#$

栈: $, \text{id } V \#$

⑦ 输入 $\text{sum} \#$

栈: $\text{id } V \#$

⑧ 输入: $\#$

栈: $V \#$

⑨ 利用 $V \rightarrow \epsilon$

\therefore 输入串入栈为 $\#$ 的弹出

$\hookrightarrow D \rightarrow T \text{ id } V \rightarrow \text{int int id } V \rightarrow \text{int id}, \text{id } V \rightarrow \text{int id}, \text{id } \epsilon$
 $\text{int id}, \text{id}$

张疆龙

202100130226

四. C风格声明语法 $G[D]$, 产生式如下, 其中 id 为变量名:

$D \rightarrow T id$ $D \rightarrow D id$ $T \rightarrow int$ $T \rightarrow int real$

a. 扩充文法

$\exists \wedge S$ $G'[S]: S \rightarrow D$ $D \rightarrow T id$ $D \rightarrow D, id$ $T \rightarrow int$
 $T \rightarrow real$

b. 求扩充文法的 LR(1) 项目规范族。

$I_0 = \{S' \rightarrow \cdot D, D \rightarrow \cdot T id, D \rightarrow \cdot D, id, T \rightarrow \cdot int, T \rightarrow \cdot real\}$

$I_1 = GOTO(I_0, D) = \{S' \rightarrow D \cdot\}$

$I_2 = GOTO(I_0, T) = \{D \rightarrow T \cdot id\}$

$I_3 = GOTO(I_0, int) = \{T \rightarrow int \cdot\}$

$I_4 = GOTO(I_0, real) = \{T \rightarrow real \cdot\}$

~~$I_5 = \{D \rightarrow T id\}$~~

$I_5 = GOTO(I_2, id) = \{D \rightarrow T id \cdot\}$

$I_6 = GOTO(I_0, D) = \{D \rightarrow D \cdot id\}$

$I_7 = GOTO(I_6, id) = \{D \rightarrow D, id \cdot\}$

$I_8 = GOTO(I_7, id) = \{D \rightarrow D, id \cdot\}$

项目规范族 $\{I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8\}$

C. 构造 LR(1) 分析表

状态	int	real	id	,	#	GoTo
I ₀	S3	S4				D 1, T 2
I ₁					acc	
I ₂			S5			
I ₃	R ₀ (T → int)	R ₄ (T → int)	R ₄ (T → int)	R ₄ (T → int)	R ₄ (T → int)	
I ₄	R ₅ (T → real)	R ₅ (T → real)	R ₅ (T → real)	R ₅ (T → real)	R ₅ (T → real)	
I ₅	R ₂ (D → T id)	R ₂ (D → T id)	R ₂ (D → T id)	R ₂ (D → T id)	R ₂ (D → T id)	6
I ₆				S7		7
I ₇			S8			
I ₈	R ₃ (D → D, id)	R ₃ (D → D, id)	R ₃ (D → D, id)	R ₃ (D → D, id)	R ₃ (D → D, id)	

S_x 指将符号压入栈并转入状态 x
R 指用规则 R 使用 1 推导产生式
空白为语法集错误

d. 给出句子 int i, sum 的分析过程

- ① 栈 [0], 输入 int i, sum #
- ② 移进 栈 [0, [-3]], 输入 i, sum #
- ③ 归约 D → T id, 栈变 [0, D], 输入: , sum #
- ④ 移进, 栈变 [0, D, ,] 输入: sum #
- ⑤ 移进, 栈: [0, D, sum] 输入: #
- ⑥ 归约, ~~栈变 [0, D, sum]~~ D → D, id 栈: [0, D], 输入 #
- ⑦ accept.

五 程序语句如下, 假设 $n \times q = 100$ 。

```
int i, sum; real x, y;
```

```
sum = 0; i = 1;
```

```
while (i <= n) { sum = sum + 1; i = i + 1; } y = x;
```

a. 填写符号表, 包括名字, 数据类型, 偏移量三项

名字	数据类型	偏移量
i	int	0
sum	int	4
x	real	8
y	real	12

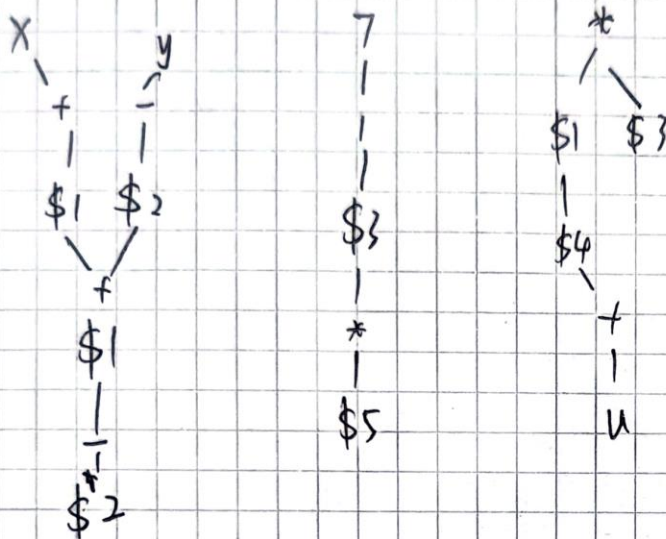
b. 写出四元式形式的中间代码

sum = 0;	$\Rightarrow (=, 0, -, \text{sum})$	(=, 0, -, sum)
i = 1;	$\Rightarrow (=, 1, -, i)$	(=, 1, -, i)
i <= n;	$\Rightarrow (<=, i, n, t_1)$ (此 t_1 为临时变量)	Loop start: (<=, i, n, t1)
sum = sum + 1;	$\Rightarrow (+, \text{sum}, i, t_2)$ ($=, t_2, -, \text{sum}$)	(IF, t1, -, LoopBody)
i = i + 1	$\Rightarrow (+, i, 1, t_3)$ ($=, t_3, -, i$)	(GOTO, -, -, Loop_End)
跳转: 若 i <= n	(IF, t1, -, [起点])	Loop-Body:
	反之 (GOTO, -, -, [循环结束点])	(+, sum, i, t2)
y = x	$\Rightarrow (=, x, -, y)$	(=, t2, -, sum)
		(+, i, 1, t3)
		(=, t3, -, i)
		(Goto, -, -, Loop_Start)
		Loop_End: (=, x, -, y)

六、有如下基本块代码：① $(+, x, y, \$1)$ ② $(-, x, y, \$2)$ ③ $(+, \$1, \$2, \$1)$
 ④ $(-, \$1, \$2, \$2)$ ⑤ $(=, 7, -, \$3)$ ⑥ $(*, \$1, \$3, \$4)$
 ⑦ $(*, \$2, 2, \$5)$ ⑧ $(+, \$4, \$5, u)$

a. 构造 DAG

翻译语句：① $\$1 = x + y$ ② $\$2 = x - y$ ③ $\$1 = \$1 + \$2$ ④ $\$2 = \$1 - \$2$
 ⑤ $\$3 = 7$ ⑥ $\$4 = \$1 * \$3$ ⑦ $\$5 = \$3 * 2$ ⑧ $u = \$4 + \5



b. 写出 DAG 优化后的代码

$\$1 = x + y$ $\$2 = \$1 - y$ $\$3 = 7$ $\$4 = \$1 * \$3$ $\$5 = \$3 * 2$
 $u = \$4 + \5

c. 写出 DAG 优化后的对应的目标代码。

ADD R1, x, y

MUL R5, R3, 2

SUB R2, R1, y

ADD u, R4, R5

MOV R3, 7

MUL R4, R1, R3