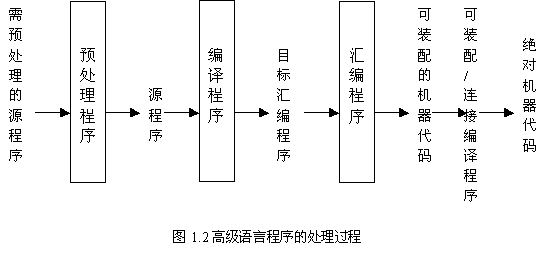
**第一章：引论**

1. **简述将高级语言源代码处理为绝对机器代码的处理过程。**

答：一个源程序有时可能分成几个模块存放在不同的文件里，将这些源程序汇集在一起的任务，由一个叫做预处理程序的程序完成，有些预处理程序也负责宏展开，像C语言和预处理程序要完成文件合并、宏展开等任务。下图中的编译程序生成的目标程序是汇编代码形式，需要经由汇编程序翻译成可在装配的机械代码，再经由装配/连接编辑程序与某些库程序连接成真正能在机器上运行的代码。



1. **编译过程可以划分为几个阶段？每个阶段的主要功能是什么？各完成什么任务？**

答：一个典型的编译程序通常包含 8 个组成部分，它们是词法分析程序、语法分析程序、语义分析程序、中间代码生成程序、中间代码优化程序、目标代码生成程序、表格管理程序和错误处理程序。

其各部分的主要功能及任务如下：

词法分析程序：输人源程序，拼单词、检查单词和分析单词，输出单词的机内表达形式。

语法分析程序：检查源程序中存在的形式语法错误，输出错误处理信息。

语义分析程序：进行语义检查和分析语义信息，并把分析的结果保存到各类语义信息表中。

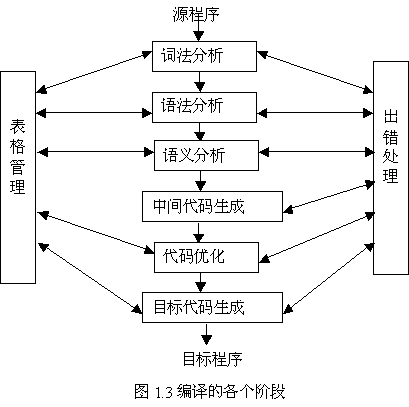
中间代码生成程序：按照语义规则，将语法分析程序分析出的语法单位转换成一定形式的中间语言代码，如三元式或四元式。

中间代码优化程序：为了产生高质量的目标代码，对中间代码进行等价变换处理。

目标代码生成程序：将优化后的中间代码程序转换成目标代码程序。

表格管理程序：负责建立、填写和查找等一系列表格工作。表格的作用是记录源程序的 各类信息和编译各阶段的进展情况，编译的每个阶段所需信息多数都从表格中读取，产生的中间结果都记录在相应的表格中。可以说整个编译过程就是造表、查表的工作过程。需要指出的是，这里的“表格管理程序”并不意味着它就是一个独立的表格管理模块，而是指编译程序具有的表格管理功能。

错误处理程序：处理和校正源程序中存在的词法、语法和语义错误。当编译程序发现源程序中的错误时，错误处理程序负责报告出错的位置和错误性质等信息，同时对发现的错误进行适当的校正（修复），目的是使编译程序能够继续向下进行分析和处理。



1. **什么是单词？单词可以分为哪几类？**
2. **什么是解释程序？它与编译程序的主要不同是什么？**

答：解释程序（Interpreter）是一种语言处理程序，它对源程序逐个语句地进行分析，根据每个语句的含义执行语句指定的功能。

广义上讲，编译程序和解释程序都属于翻译程序，但它们的翻译方式不同，解释程序是 边翻译（解释）边执行，不产生目标代码，输出源程序的运行结果。而编译程序只负责把源 程序翻译成目标程序，输出与源程序等价的目标程序，而目标程序的执行任务由操作系统来完成，即只翻译不执行。

1. **if(x>y)z=x+y;**

**else z=123.0; 上述C语言代码包括多少个单词？每个单词属于什么类别？**

答：if else 保留字

（） > = + 运算符

x y z 字母字符

123.0 数字字符

共包含有十五个单词。

**第三章：文法和语法**

1. **什么是最左推导、最右推导和二义文法？**
2. **给定文法G=（{E}，{+，\*，i，(，)}，P，E），其中P为**

E->E+E

E->E\*E

E->(E)

E->i

给出句型i\*i+i的最左推导和最右推导过程，并判断该文法是否二义文法。

答：（1）对于题目中给出的文法G，句型i\*i+i有两个不同的最左推导1和2，它们对应的语法树分别如下图：

E

E

E

E

E

+

+

i

i

i

E 

E

E

E

+

+

i

i

i

E

推导1语法树 推导2语法树

推导1 E→E+E→E\*E+E

→i\*E+E→i\*i+E→i\*i+i

推导2 E→E\*E→i\*E

→i\*E+E→i\*i+E→i\*i+i

（2）该文法是二义的。

**3．已知文法G[S]:**

**S→(L) | aS | a**

**L→L， S | S**

**求出句型（S，（a））的所有短语、直接短语和句柄。**

答：（1）句型(S,(a))的语法树如图：

S

S

）

（

L

，

L

S i

L

）

（

S i

a i

（2）从语法树中我们可以找到：

短语：a, S, (a), S,(a) , (S,(a))

直接短语：a, S

句柄：S

**4．现有文法G[E]：**

**E->E+T|E-T|T**

**T->T\*F|T/F|F**

**F->(E)|i**

**画出句型(F+i)-T\*(E-T)的语法树，求出该句型的所有短语、直接短语和句柄。**

答：所以短语有：

E

E

-

T

T

F

E

(

)

+

E

T

T

F

F

i

\*

T

F

E

(

)

-

E

T

F，i，F+i，(F+i)，E-T，(E-T)

T\*(E-T)，(F+i)-T\*(E-T)

直接短语有：

F，i ，E-T

句柄为：F

**5．****简述自上而下和自下而上语法分析方法基本思想。两种方法在分析过程中各自会遇到什么困难？**

答：自上而下分析法

基本思想：它从文法的开始符号出发，反复使用各种产生式，寻找"匹配"的推导。

优点：直观、简单和宜于手工实现；

缺点：若发现推导过程有误，需要回溯，代价极高，效率很低。

自下而上分析法

基本思想：从输入串开始，逐步进行“归约”，直到文法的开始符号。即从树末端开始，构造语法树。

优点：适合分析表达式；

缺点：需要精确定义“可归约串”。

**第四章：词法分析**

1. **词法分析器的主要任务是什么？单词常分为哪几类？识别出的单词在编译程序中如何表示？**
2. **给出下列文法所对应的正规式，并指出该文法的语言是什么。**

**S->0A|1B**

**A->1S|1**

**B->0S|0**

答：解方程组S的解：

S=0A|1B

A=1S|1

B=0S|0

将A、B产生式的右部代入S中

S=01S|01|10S|10=（01|10）S|（01|10）

所以：S= (01|10)\*(01|10)

1. **构造与正规表达式 1(0|1)\*101相应的NFA，再将NFA转换成DFA。**

答：先构造该正规式的转换系统：

Z

1(0|1)\*101

Z

1

1

0

1

(0|1)\*

Z

1

1

0

1

ε

ε

0

1

由上述转换系统可得状态转换集K={S, 1, 2, 3, 4, 5, Z}，状态子集转换矩阵如下表所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| I | I0 | I1 | S | 0 1 |
| {S} |  | {1, 2, 3} | 0 | 1 |
| {1, 2, 3} | {2, 3} | {2, 3, 4} | 1 | 2 3 |
| {2, 3} | {2, 3} | {2, 3, 4} | 2 | 2 3 |
| {2, 3, 4} | {2, 3, 5} | {2, 3, 4} | 3 | 4 3 |
| {2, 3, 5} | {2, 3} | {2, 3, 4, Z} | 4 | 2 5 |
| {2, 3, 4, Z} | {2, 3, 5} | {2, 3, 4} | 5 | 4 3 |

其对应的DFA状态转换图为：

5

1

1

0

0

1

1

0

0

1

1

0

现在对该DFA进行化简，最终得到下列化简后的状态转换图（先将其分成两组——终态组{5}和非终态组{0, 1, 2, 3, 4}，再根据是否可继续划分来确定最后的组数）：

5

1

0

1

1

1

1

0

0

0

1. **将下图所示的非确定的有穷自动机（NFA）转换成与之等价的确定的有穷自动机（DFA），并最小化。**

**0**

**ε**

**ε**

**0**

**1**

1. **请构造与正规式R=(a\*|b\*)b(ba)\*等价的状态最少的DFA。**

答：

ε

ε

ε

ε

ε

ε

b

a

b

a

b

a

b

**b**

a

b

a

b

b

b

b

b

|  |  |  |
| --- | --- | --- |
|  | a | b |
| [S，A，B，C] 1 | [B] 2 | [C，D] 3 |
| [B] 2 | [B] 2 | [D] 4 |
| [C，D] 3 | Φ | [C，D，F] 5 |
| [D] 4 | Φ | [F] 6 |
| [C，D，F] 5 | [E] 7 | [C，D，F] 5 |
| [F] 6 | [E] 7 | Φ |
| [E] 7 | Φ | [F] 6 |

a

b

b

a

a

a

b

b

P0={{3，4，5，7}，{1，2，6}}

P1={{3}，{4，7}，{5}，{1，2}，{6}}

P2={{3}，{4，7}，{5}，{1}，{2}，{6}}

P3=P2

a

a

b

b

b

b

a

b

a

(a\*|b\*)b(ba)\*的最小DFA

**第五章：自顶向下语法分析方法**

1. **简述自顶向下语法分析法的基本思想？**

答：从左到右分析输入序列ω，经过一系列的步骤，最终将ω归约为文法的开始符号，或者发现一个语法错误。

归约是推导的逆过程，是一个反复用产生式的左部替换产生式的右部、谋求对输入序列进行匹配的过程。

1. **什么文法是LL（1）文法？**

**3．求文法G[E]中每个非终结符的First集和Follow集：**

**E → TE’ E’ → +TE’ |ε**

**T → FT’ T’ → \*FT’ |ε F → (E) | i**

答：

①非终结符的First集:

FIRST(F)={（，id }

FIRST(T')={\*，ε} FIRST(T)=FIRST(F)={（，id }

FIRST(E')={+，ε}

FIRST(E)=FIRST(T)={（，id }

②非终结符的Follow集:

FOLLOW( E ) = { )， # }

FOLLOW( E') = FOLLOW( E ) = { )， # }

FOLLOW( T ) = { +， )， # }

FOLLOW( T') = FOLLOW( T ) = { +， )， # }

FOLLOW( F ) = { \*， +， )， # }

**4．已知文法G：**

**E → E + T | T**

**T → T \* F | F**

**F → ( E ) | i**

**请给出字符串 i + i \* i 的最右推导过程和语法树**

答：最右推导过程：

T

E

E

F

T

+

\*

i

F

T

F

E→E+T

→T+T\*F

→T+F\*i

→F+F\*i

→i+i\*i

**5．已知文法G[E]，请消除文法中的直接左递归**

**E → E + T | T**

**T → T \* F | F**

**F → ( E ) | I**

答：按转换规则，可得:

E→TE'

E'→+TE'|ε

T→FT '

T'→\*FT'|ε

F→( E ) | I

**6．对下面文法先消除左递归，并提取左公共因子，再判断其是否是LL（1）文法。**

S->Aa|b

A->SB

B->ab

答：用A的产生式右部代替S的产生式右部的A得：

S→SBa|b

B→ab

消除左递归后文法变为：

0) S→b N

1) N→B a N

2) N→ε

3) B→a b

|  |  |  |
| --- | --- | --- |
| 非终结符 | FIRST集 | FOLLOW集 |
| S | {b} | {#} |
| B | {a} | {a} |
| N | {ε，a} | {#} |

对相同左部的产生式可知：

SELECT(N→B a N)∩SELECT(N→ε) ={ a }∩ {# }= Φ

所以文法是LL(1)的。

预测分析表（Predicting Analysis Table）

|  |  |  |  |
| --- | --- | --- | --- |
|  | a | b | # |
| S |  | →b N |  |
| B | →a b |  |  |
| N | →B a N |  | →ε |

也可由预测分析表中无多重入口判定文法是LL(1)的。

**7．递归下降法(递归子程序法)的主要思想是什么？如何用递归下降法实现语法分析？**