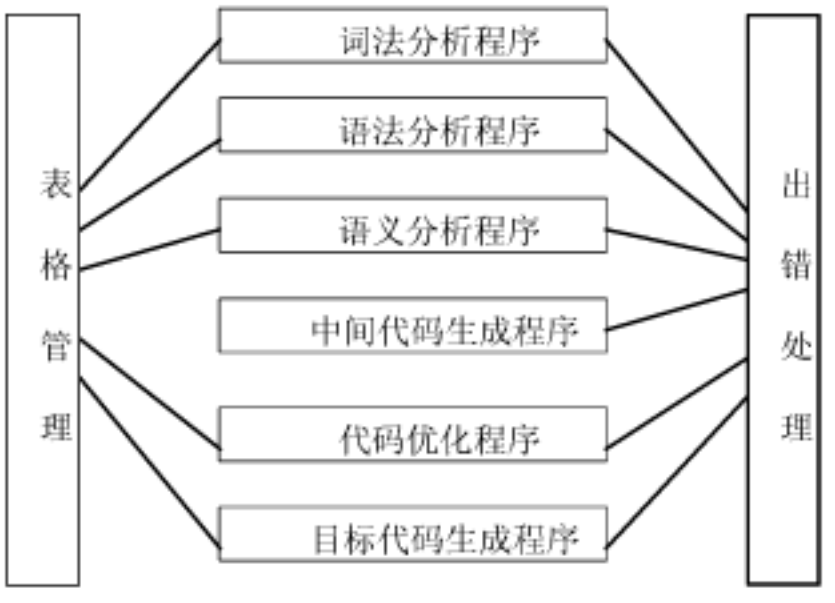
1．编译程序在逻辑上由哪几部分组成？请简述每部分的功能。编译程序和解释程序有哪些区别？



词法分析器，又称扫描器，输入源程序，进行词法分析，输出单词符号。
语法分析器，简称分析器，对单词符号串进行语法分析（根据语法规则进行推导或归约），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”。
语义分析和中间代码产生器，按照语义规则对于法分析器归约出（或推导出）的语法单位进行语义分析并把它们翻译成一定形式的中加代码。
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程序。
表格管理：编译程序在工作过程中需要保持一系列的表格，以登记源程序的各类信息和编译各结点的进展状况。
一个编译程序不仅应能对书写正确的程序进行翻译，而且应能对出现在源程序中的错误进行恢复。如果源程序有错误，编译程序应设法发现错误，把有关错误信息报告给用户，这就是由错误处理程序完成的。

编译程序和解释程序的区别：编译程序以一个可执行程序的描述作为输入，以另一个等价的可执行程序的描述作为输出。以某种语言（源语言程序）转变成为另外一种语言程序（目标语言程序），而后者与前者在逻辑上是等价的。

解释程序以一个可执行程序的描述作为输入，但是不产生目标程序，而是边解释边执行，以执行这一可执行程序描述的结果作为输出。

2．简述为什么自顶向下的语法分析技术不能处理具有左递归的文法。
答：在自顶向下的语法分析技术中，要解决的问题是根据当前输入符号判断将识别符号以及非终结符号替换成哪条规则的右部，若文法具有左递归，则在分析过程中，无法判断替换的规则，造成无穷递归求解过程。

3. 简要叙述语法分析的基本功能是什么？对于同一个文法，**LALR(1)**和 **SLR(1)** 的分析表状态个数相同，为什么前者的分析能力要比后者强？
答：语法分析的基本功能是：

a) 语法分析处于词法分析和语义分析之间，它的输入是词法分析的输出，它的输出是语义分析的输入。（**1** 分）
b) 词法分析对输入的字符串进行分析，判断是否一个合法的输入。其中合法是指输入的字符串是否符合程序设计语言的语法规定（或者文法的规定）。（**3** 分）
c) 对于不符合语法的字符串要设计错误处理机制。其分析方法包括自顶向下和自底向上分析两种。（**1** 分）

LALR(1)比 **SLR(1)**分析能力强的原因：

在构造分析表时，SLR(1)中的规约项填写的是全体 FOLLOW(A)集合中的符号，这样就增加了移动-规约冲突的可能性。（**2** 分）

而对于 LALR(1)，虽然分析表状态和 SLR(1)同样多，但是它采用了向前搜索符技术，使得规约项填写的只是 FOLLOW(A)集合的子集，而且大部分时间下是真子集，这就使得产生移动-规约冲突的可能性减少，因此更加精确，所以分析能力更强。（**3** 分）

4. 通过合并 **LR(1)**文法中的同心状态得到的 **LALR(1)**文法可能会产生哪些冲突？一定不会产生哪些冲突？为什么？

答：可能会产生归约-归约冲突，一定不会产生移进-归约冲突。

因为在对 LR(1)合并同心集合时，有可能将原本没有冲突的同心集的项目集合并后造成一些归约项目向前搜索符集合的交集不是空，产生归约-归约冲突。但是由于文法本身已经是 LR(1)文法，因此可知，在项目集中一定不存在移进-归约冲突，也就是移进项目要求输入的终结符和任意归约项目的向前搜索符集合的交集都是空集。这样，在将同心集合并之后，移进项目要求输入的终结符和归约项目的向前搜索符集合的交集也还是空集。

5. 何谓二义性文法？试举一例说明。

答：若文法 G 的一个句子对应有两棵或两棵以上不同的推导树，则称该句子是二义性的。产生二义性句子的文法称为二义性文法，否则该文法是无二义性的。

例子：E---i|E+E|E*E　　i*i+i 的推导

6. 设 **G=**（$\mathbf{V_N}$，$\mathbf{V_T}$，**P**，**<S>**）是上下文无关文法，产生式集合 **P** 中任意一个产生式应具有什么样的形式？若 **G** 是正则文法呢？

答：上下文无关文法的产生式形式为：

$A\rightarrow\alpha$ ，其中，$A\in V_N$，$\alpha\in(V_N\cup V_T)^*$

正则文法产生式形式为：

$A\rightarrow aB$，或 $A\rightarrow a$（右线性文法）其中，$A,B\in V_N$，$a\in V_T$

$A\rightarrow Ba$，或 $A\rightarrow a$（左线性文法）其中，$A,B\in V_N$，$a\in V_T$

7. 试简单描述 **LL(1)**，**LR(1)**，**SLR(1)**和 **LALR(1)**的定义。并说明它们是怎么解决分析表中的冲突的。

LL(1)定义：一个文法 G 是 LL（1）的，当且仅当对于 G 的每一个非终结符 A 的任何两个不同产生式 $A\rightarrow\alpha|\beta$，下面的条件成立：

SELECT($A\rightarrow\alpha$)∩SELECT($A\rightarrow\beta$)= ,其中，

α|β 不能同时 ε。

SLR(1)定义：满足下面两个条件的文法是 SLR(1)文法

a.对于在 s 中的任何项目 $A\rightarrow\alpha.X\beta$,当 X 是一个终结符，且 X 在 Follow(B)中时，s 中没有完整的项目 $B\rightarrow r.$

b.对于在 s 中的任何两个完整项目 $A\rightarrow\alpha.$和 $B\rightarrow\beta.$,Follow(A)∩Follow(B)为空。

LL(1)就是向前只搜索 1 个符号，即与 FIRST()匹配，如果 FIRST 为空则还要考虑 Follow。

LR 需要构造一张 LR 分析表，此表用于当面临输入字符时，将它移进，规约（即自下而上分析思想），接受还是出错。

LR(0)找出句柄前缀，构造分析表，然后根据输入符号进行规约。不考虑先行，只要出现终结符就移进，只要出现归约状态，就无条件归约,这样子可能出现归约－移进，归约－归约冲突。

SLR(1)使用 LR(0)时若有归约－归约冲突，归约－移进冲突，所以需要看先行，则只把有问题的地方向前搜索一次。

8. 简要叙述词法分析的基本功能是什么？如果让你实现一个词法分析程序（独立的词法分析程序，不是语法分析的子程序），你将如何考虑？请简述你的设计方案（简述要点即可）

1、识别出源程序中的各个单词符号，并转换成内部编码形式

2、删除无用的空白字符回车字符以及其他非实质性字符

3、删除注释

4、进行词法检查，报告所发现的错误。

9. 何谓源程序、目标程序、翻译程序、编译程序和解释程序**?**它们之间可能有何种关系**?**

解：源程序是指以某种程序设计语言所编写的程序。目标程序是指编译程序（或解释程序）将源程序处理加工而得的另一种语言（目标语言）的程序。翻译程序是将某种语言翻译成另一种语言的程序的统称。编译程序与解释程序均为翻译程序，但二者工作方法不同。解释程序的特点是并不先将高级语言程序全部翻译成机器代码，而是每读入一条高级语言程序语句，就用解释程序将其翻译成一段机器指令并执行之，然后再读入下一条语句继续进行解释、执行，如此反复。即边解释边执行，翻译所得的指令序列并不保存。编译程序的特点是先将高级语言程序翻译成机器语言程序，将其保存到指定的空间中，在用户需要时再执行之。即先翻译、后执行。

1．遍 答：源程序或目标程序的中间结果从头至尾扫描一遍，并进行相关的加工处理，产生一种新的中间结果保存形式或者目标程序。

2、二义性文法 答：若一个文法存在某个句子对应两棵不同的语法树，则称这个文法是二义的。

或者说，若一个文法存在某个句子有两个不同的最左（右）推导，则称这个文法是二义的。

3．源程序和目标程序 答：源程序是高级程序设计语言书写的程序，目标程序是低级程序设计语言（目标语言）书写的程序

4．规范推导 答：在推导的任何一步 $\alpha\Rightarrow\beta$，其中 α、β 是句型，都是对 α 中的最右非终结符进行替换，这种推导叫做最右推导，也叫做规范推导。

5. 编译程序和解释程序

答：编译程序以一个可执行程序的描述作为输入，以另一个等价的可执行程序的描述作为输出。解释程序以一个可执行程序的描述作为输入，以执行这一可执行程序描述的结果作为输出。

6.句柄

文法 G[S]，α β δ 是文法 G 的一个句型，S=>*α Aδ 且 A=>+β 则称β 是句型 α β δ 相对于非终结符 A 的短语。若有 A　β 则称β 是句型α β δ 相对于该规则 A→β 的直接短语。一个句型的最左直接短语称为该句型的句柄。

7.活前缀

活前缀：指规范句型的一个前缀，这种前缀不含句柄之后任何符号。之所以称为活前缀，是因为在右边添加一些符号之首，就可以使它称为一个规范句型。