种语言，而后者与前者在逻辑上是等价的。

编译过程：词法、语法、语义分析与中间代码、优化、目标代码生成

词法分析：输入源程序，对构成源程序的字符串进行扫描和分解，识别出一个个单词。

语法分析：在词法分析的基础上，根据语言的语法规则，把单词符号串分解成各类语法单位。

语义分析与中间代码产生：对语义分析所识别出的各类语法范畴，分析其含义并进行初步翻译（产生中间代码）；

优化：优化的任务在于对前段产生的中间代码进行加工变换，以期在最后阶段能产生出更为高效（省时间和空间)的目标代码。

目标代码生成：把中间代码（或经优化处理之后）变换成特定存储器上的低级语言代码。

编译程序结构：表格管理、出错处理

编译前端：由与源语言有关但与目标语言无关的那些部分组成，包括词法分析、语义分析、语义分析与中间代码产生。

后端：编译程序中与目标语言有关那些部分，优化与目标代码生成。后端不依赖于源语言而仅仅依赖于中间语言。

词法规则是指单词符号的形成规则。

语言的语法规则规定了如何从单词符号形成更大的结构（语法单位)。

所谓一个语言的语义是指这样的一组规则，使用它可以定义一个程序的意义，这些规则称为语义规则。**文法**是描述语言的语法结构的**形式规则**

上下文无关文法：是这样一种文法，它所定义的语法范畴是完全独立于这种范畴可能出现的环境。

上下文无关文法组成：一组终结符号一组非终结符号，一个开始符号以及一组产生式。

开始符号：是一个特殊的非终结符号，它代表所定义的语言中我们最终感兴趣的语法范畴，这个语法范畴通常称为“句子”

产生式：是定义语法范畴的一种书写规则。

二义性：如果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。

含有左递归的文法将使自上而下的分析过程陷入无限循环。

LL(1)分析条件 :当一个文法不含左递归，并且满足每个非终结符的所有候选首符集两两不相交的条件

LL(1)的含义:第一个 L 表示从左到右扫描输入串，第二个 L 表示最左推导，1 表示分析时每一步只需向前查看一个符号

自上而下分析的问题 :①文法含有左递归时，分析过程会陷入无限循环②回溯浪费分析时间③某一非终结符用某一候选式匹配成功时，可能是暂时的④分析不成功时，难以找到出错位置

自下而上分析的问题 :怎样判断栈顶的符号串的可归约性，以及如何归约。

一个句型的最左直接短语称为该句型的句柄。

在形式语言中最右推导常被称为 **规范推导**，由规范推导所得的句型称为规范句型，如果文法无二义的，那么规范推导（最右推导）的逆过程必是规范归约（最左归约)

属性分为两类：综合属性，继承属性，综合属性用于“自下而上”传递信息，继承属性用于“自上而下”传递信息。

在上下文无关文法的基础上，为每个文法符号（终结符或非终结符）配备若干相关的“值”（称为属性）

语义规则：文法每个产生式都配备了一组属性的计算规则。

语法制导翻译：由源程序的语法结构所驱动的处理办法。

输入串-----语法树-------依赖图--------语义规则计算次序

静态检查和中间代码产生的地位：

----语法分析器 -----静态检查器 -------中间代码产生器-------优化器-------

属性文法 :对于文法的每个产生式都配备了一组属性的计算规则，在上下文无关文法的基础上，为每个符号都配备了若干相关属性。

中间语言形式: 后缀式，三地址代码（包括三元式，四元式、间接三元式），DAG 图表示

后缀式表示法（逆法兰表示法）：把运算量（操作数）写在前面，把算符写在后面（后缀）

四元式:(OP Arg1 Arg2 Result)

三元式:(OP Arg1 Arg2)

E.T=merge(E_1.T,E_2.T)

E.F=E_2.F

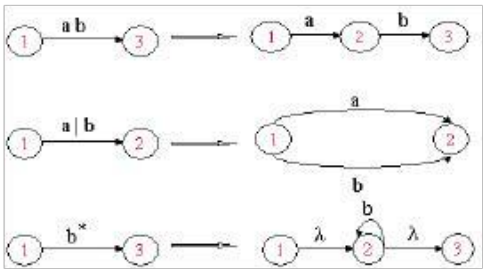
E→E_1 and M E_2: backpatch(E_1.T,M.quad)

E.T=E_2.T

E.F=merge(E_1.F,E_2.F)

最左规约=规范规约：A

最右推导=规范推导：B



短语:每棵子树对应一个短语

直接短语:只有两层的子树对应的短语

句柄:最左直接短语

E→TE’

```
Procedure E
Begin
    T;E’
End
```

E’→+TE’|

```
Procedure E’
If sym=’+’ then
    Begin
        Advance T;E’
    End
```

F→(E)|i

```
Procedure F
If sym=’i’ then advance
Else if sym=’(’ then
    Begin
        Advance E
        If sym=’)’ then advance
        Else error
    End
Else error
```

种语言，而后者与前者在逻辑上是等价的。

编译过程：词法、语法、语义分析与中间代码、优化、目标代码生成

词法分析：输入源程序，对构成源程序的字符串进行扫描和分解，识别出一个个单词。

语法分析：在词法分析的基础上，根据语言的语法规则，把单词符号串分解成各类语法单位。

语义分析与中间代码产生：对语义分析所识别出的各类语法范畴，分析其含义并进行初步翻译（产生中间代码）；

优化：优化的任务在于对前段产生的中间代码进行加工变换，以期在最后阶段能产生出更为高效（省时间和空间)的目标代码。

目标代码生成：把中间代码（或经优化处理之后）变换成特定存储器上的低级语言代码。

编译程序结构：表格管理、出错处理

编译前端：由与源语言有关但与目标语言无关的那些部分组成，包括词法分析、语义分析、语义分析与中间代码产生。

后端：编译程序中与目标语言有关那些部分，优化与目标代码生成。后端不依赖于源语言而仅仅依赖于中间语言。

词法规则是指单词符号的形成规则。

语言的语法规则规定了如何从单词符号形成更大的结构（语法单位)。

所谓一个语言的语义是指这样的一组规则，使用它可以定义一个程序的意义，这些规则称为语义规则。**文法**是描述语言的语法结构的**形式规则**

上下文无关文法：是这样一种文法，它所定义的语法范畴是完全独立于这种范畴可能出现的环境。

上下文无关文法组成：一组终结符号一组非终结符号，一个开始符号以及一组产生式。

开始符号：是一个特殊的非终结符号，它代表所定义的语言中我们最终感兴趣的语法范畴，这个语法范畴通常称为“句子”

产生式：是定义语法范畴的一种书写规则。

二义性：如果一个文法存在某个句子对应两棵不同的语法树，则称这个文法是关键字、标识符、常数、运算符、界符含有左递归的文法将使自上而下的分析过程陷入无限循环。

LL(1)分析条件 :当一个文法不含左递归，并且满足每个非终结符的所有候选首符集两两不相交的条件

LL(1)的含义:第一个 L 表示从左到右扫描输入串，第二个 L 表示最左推导，1 表示分析时每一步只需向前查看一个符号

自上而下分析的问题 :①文法含有左递归时，分析过程会陷入无限循环②回溯浪费分析时间③某一非终结符用某一候选式匹配成功时，可能是暂时的④分析不成功时，难以找到出错位置

自下而上分析的问题 :怎样判断栈顶的符号串的可归约性，以及如何归约。

一个句型的最左直接短语称为该句型的句柄。

在形式语言中最右推导常被称为 **规范推导**，由规范推导所得的句型称为规范句型，如果文法无二义的，那么规范推导（最右推导）的逆过程必是规范归约（最左归约)

属性分为两类：综合属性，继承属性，综合属性用于“自下而上”传递信息，继承属性用于“自上而下”传递信息。

在上下文无关文法的基础上，为每个文法符号（终结符或非终结符）配备若干相关的“值”（称为属性）

语义规则：文法每个产生式都配备了一组属性的计算规则。

语法制导翻译：由源程序的语法结构所驱动的处理办法。

输入串-----语法树-------依赖图--------语义规则计算次序

静态检查和中间代码产生的地位：

----语法分析器 -----静态检查器 -------中间代码产生器-------优化器-------

属性文法 :对于文法的每个产生式都配备了一组属性的计算规则，在上下文无关文法的基础上，为每个符号都配备了若干相关属性。

中间语言形式: 后缀式，三地址代码（包括三元式，四元式、间接三元式），DAG 图表示

后缀式表示法（逆法兰表示法）：把运算量（操作数）写在前面，把算符写在后面（后缀）

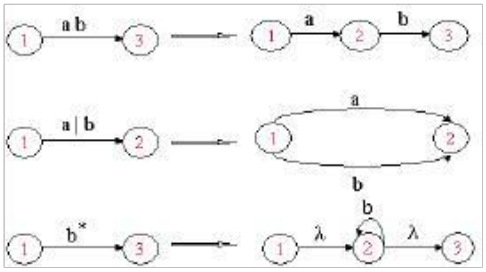
四元式:(OP Arg1 Arg2 Result)

$E \rightarrow E_1$ or M E_2: backpatch(E_1.F,M.quad);
E.T=merge(E_1.T,E_2.T)
E.F=E_2.F

$E \rightarrow E_1$ and M E_2: backpatch(E_1.T,M.quad)
E.T=E_2.T
E.F=merge(E_1.F,E_2.F)

最左规约=规范规约：A

最右推导=规范推导：B



短语:每棵子树对应一个短语

直接短语:只有两层的子树对应的短语

句柄:最左直接短语

E→TE’

```
Procedure E
Begin
    T;E’
End
```

E’→+TE’|

```
Procedure E’
If sym=’+’ then
    Begin
        Advance T;E’
    End
```

F→(E)|i

```
Procedure F
If sym=’i’ then advance
Else if sym=’(’ then
    Begin
        Advance E
        If sym=’)’ then advance
        Else error
    End
Else error
```