中山大學
SUN YAT-SEN UNIVERSITY

国家超级计算广州中心
NATIONAL SUPERCOMPUTER CENTER IN GUANGZHOU

DCS290

Compilation Principle

编 译 原 理

第四章 语法分析（2）

郑馥丹
zhengfd5@mail.sysu.edu.cn

CONTENTS

目 录

4. 确定的自顶向下语法分析[predictive]

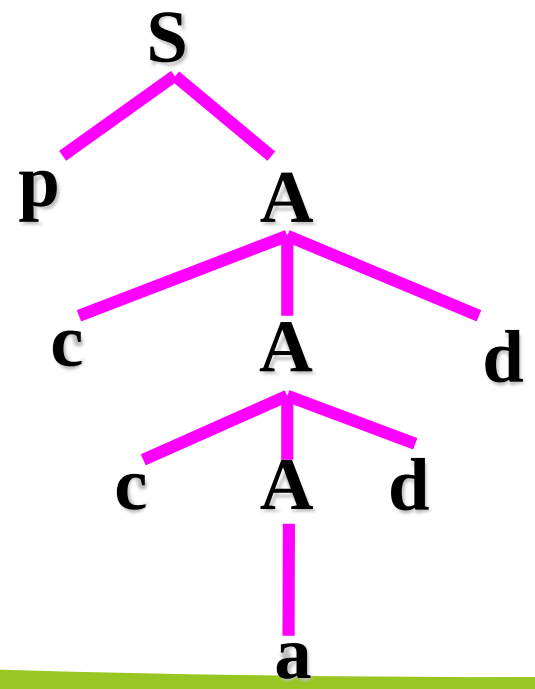
- 例：设有文法G[S]:

  S→pA|qB

  A→cAd|a

  B→dB|b

若输入串W=pccadd，自顶向下的推导过程为:



该推导过程是确定的！

原因:

(1) 每个产生式的右部由终结符开头

(2) 同一个非终结符的不同产生式的右部由不同的终结符开头。

因此，在推导过程中可以根据当前的输入符号**唯一确定选哪个产生式**往下推导，分析过程是确定的。

S =>pA =>pcAd =>pccAdd =>pccadd

4. 确定的自顶向下语法分析[predictive]

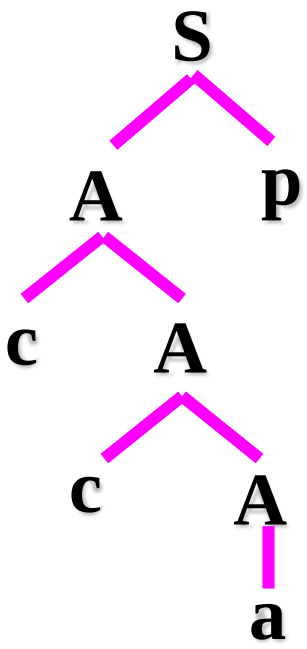
- 例：设有文法G[S]:

  S→Ap|Bq

  A→a|cA

  B→b|dB

  若输入串W=ccap，自顶向下的推导过程为：



S =>Ap =>cAp =>ccAp =>ccap

该推导过程可能是确定的

当：

(1) 产生式右部以终结符或非终结符开头（**无空产生式**）；

(2) **同一非终结符的不同产生式的右部由不同的符号开头**。

对于这种文法，在推导过程选用哪个产生式不直观，关键是判断产生式右部推出的**开始符号集——FIRST集**，分析过程可能是确定的。

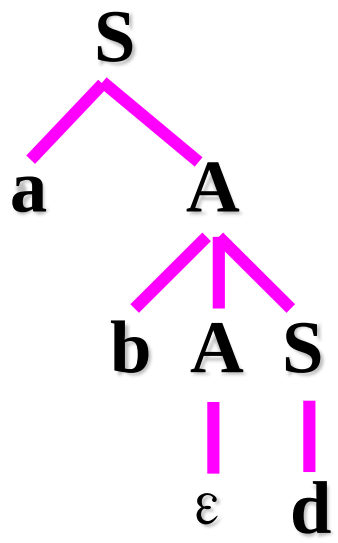
4. 确定的自顶向下语法分析[predictive]

- 例：设有文法G[S]

S→aA|d

A→bAS| ε

若输入串W=abd，自顶向下的推导过程为：



该推导过程可能是确定的

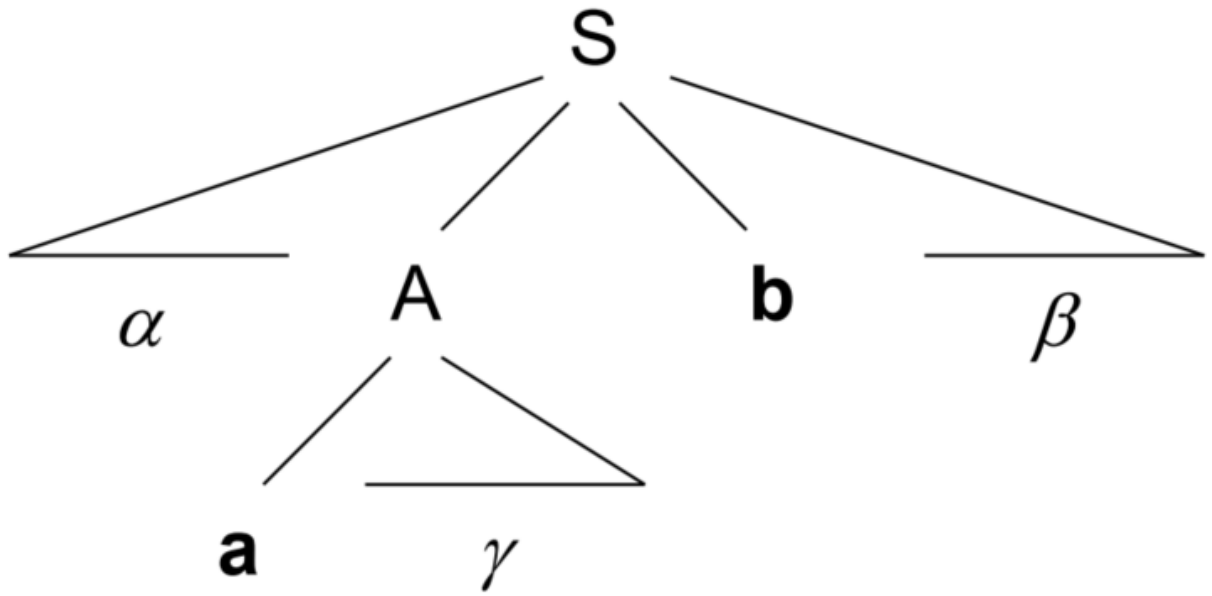
文法的特点是：**包含空产生式**。对于空产生式左部的非终结符，关键是判断该非终结符的**后跟符号集——FOLLOW集**，分析过程可能是确定的。

S =>aA =>abAS =>abS =>abd

4. 确定的自顶向下语法分析[predictive]

- **FIRST集**和**FOLLOW集**的直观理解

  a ∈ FIRST(A) and **b** ∈ FOLLOW(A)



- 开始符号集——**FIRST集**
  - 设G=(V_N, V_T, P, S)是上下文无关文法，$\beta \in (V_N \cup V_T)^*$ ，
    FIRST(β) = { **a** $\in V_T$ | $\beta \Rightarrow^*$ **a**……}
  - 若$\beta \Rightarrow^* \varepsilon$ 则规定$\varepsilon \in$FIRST(β)
  - 直观上说，文法符号串β的**开始符号集是由β推导出的开头的终结符（包括ε）组成的。**

4. 确定的自顶向下语法分析[predictive]

- 开始符号集——**FIRST集**

例：文法G[S]：

S→Ap	**FIRST(Ap)= {a,c}**
S→Bq	**FIRST(Bq)= {b,d}**
A→a	**FIRST(a)= {a}**
A→cA	**FIRST(cA)= {c}**
B→b	**FIRST(b)= {b}**
B→dB	**FIRST(dB)= {d}**

可进行确定的自顶向下分析

由于**同一非终结符的两个产生式**的右部推导出来的**开始符号集不相交**，因此可根据当前输入符属于哪个产生式右部的开始符号集而决定**唯一选哪个产生式进行推导**，可以进行确定的自顶向下分析。

4. 确定的自顶向下语法分析[predictive]

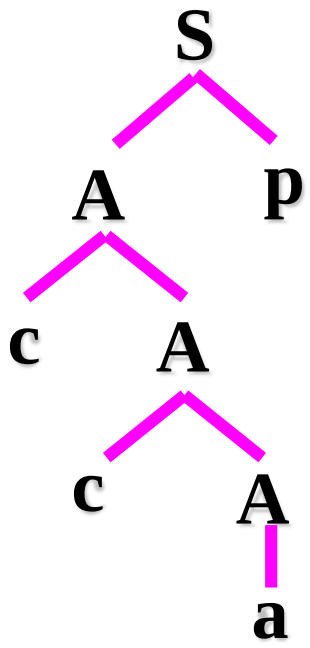
- 开始符号集——**FIRST集**

例：文法G[S]：

S→Ap|Bq

A→a|cA

B→b|dB



若输入串W=ccap,自顶向下的推导过程为：

S→Ap	FIRST(Ap)={a,c}
S→Bq	FIRST(Bq)={b,d}
A→a	FIRST(a)={a}
A→cA	FIRST(cA)={c}
B→b	FIRST(b)={b}
B→dB	FIRST(dB)={d}

S =>Ap =>cAp =>ccAp =>ccap

4. 确定的自顶向下语法分析[predictive]

- 开始符号集——**FIRST集**
  - 在求各个非终结符的FIRST集之前，先确定它们是否能⇒*ε
    - ✓ 第1次扫描——扫描文法中的产生式
      - 对能直接推出ε的产生式左部的终结符标“是”。
      - 删除所有右部含有终结符的产生式。若以某一非终结符为左部的所有产生式都被删除，则该非终结符不能推出ε，将其标为“否”。

例：G[S]

S→AB
~~S→bC~~
~~A→b~~
A→ε
~~B→aD~~

B→ε
C→AD
~~C→b~~
~~D→aS~~
~~D→c~~

非终结符	S	A	B	C	D
第1次扫描		是	是		否
第2次扫描					

4. 确定的自顶向下语法分析[predictive]

- 开始符号集——**FIRST集**
  - 在求各个非终结符的FIRST集之前，先确定它们是否能⇒*ε
    - ✓ 第2次扫描——扫描产生式右部的符号
      - 对每个产生式p: $A \rightarrow X_1 \ldots X_n$，如果$X_1,\ldots,X_n$都被标为“是”(即$X_1,\ldots,X_n$都能推出ε)，则A也能推出ε，将其标为“是”。
      - 如果$A \rightarrow Y_1 \ldots Y_n$中，$Y_1 \ldots Y_n$中任一个已被标为“否”，则删掉该产生式，若这么做使得A的所有产生式都被删去，则A不能推出ε，将其标为“否”。

S→AB　　B→ε
~~S→bC~~　　~~C→AD~~
~~A→b~~　　~~C→b~~
A→ε　　~~D→aS~~
~~B→aD~~　　~~D→c~~

非终结符	S	A	B	C	D
第1次扫描		是	是		否
第2次扫描	是			否	

4. 确定的自顶向下语法分析[predictive]

- 开始符号集——**FIRST集**
  - 对**每一文法符号X**($X \in V_T \cup V_N$)，求**FIRST(X)**的算法:
    - ✓ 对每个$a \in V_T$：FIRST(a)={a}
    - ✓ 对每个$A \in V_N$：若$A \Rightarrow^* \varepsilon$，则$\varepsilon \in$ FIRST(A)
    - ✓ 对每个$A \in V_N$：若$A \rightarrow a\cdots$，$a \in V_T$，则$a \in$FIRST(A)
    - ✓ 若$X, Y_1, Y_2, \ldots, Y_n$都$\in V_N$，有产生式$X \rightarrow Y_1 Y_2 \ldots Y_n$，当$Y_1, Y_2, \ldots, Y_{n-1}$都$\Rightarrow^* \varepsilon$时，$FIRST(Y_1)-\{\varepsilon\}, FIRST(Y_2)-\{\varepsilon\}, \ldots, FIRST(Y_{n-1})-\{\varepsilon\}, FIRST(Y_n)$都包含在FIRST(X)中
    - ✓ 当所有$Y_i \Rightarrow^* \varepsilon$，则$FIRST(X)= FIRST(Y_1) \cup FIRST(Y_2) \cup \ldots \cup FIRST(Y_n) \cup \{\varepsilon\}$

4. 确定的自顶向下语法分析[predictive]

- 开始符号集——**FIRST集**

例G[S]:

S→AB　S→ bC

A→b　　A→ ε

B→aD　B→ ε

C→AD　C→ b

D→aS　D→ c

已求出能推出ε的非终结符集为**{A,B,S}**

	First集(0)	First集(1)	First集(2)	First集(3)
S	ε	ε b	ε b a	ε b a
A	ε	ε b	ε b	ε b
B	ε	ε a	ε a	ε a
C		b	b a c	b a c
D		a c	a c	a c
a	a	a	a	a
b	b	b	b	b

4. 确定的自顶向下语法分析[predictive]

- 开始符号集——**FIRST集**
  - 利用求出每个文法符号的FIRST集求符号串的FIRST集，设$\alpha=X_1X_2...X_n$:
    - ✓ 当X_1不能=>* ε，则$FIRST(\alpha)=FIRST(X_1)$
    - ✓ 若对任何j($1\le j < n$）都有$\varepsilon\in FIRST(X_j)$，则$FIRST(\alpha)=(FIRST(X_1) - \{\varepsilon\}) \cup ... \cup (FIRST(X_j) - \{\varepsilon\}) \cup FIRST(X_{j+1})$
    - ✓ 若对所有i ($1\le i\le n$)，都有$\varepsilon\in FIRST(X_i)$，则 $FIRST(\alpha)=FIRST(X_1) \cup ... \cup FIRST(X_n) \cup \{\varepsilon\}$

4. 确定的自顶向下语法分析[predictive]

- 开始符号集——**FIRST集**

例：G[S]

S→AB|bC
A→b|ε
B→aD|ε
C→AD|b
D→aS|c

已求出非终结符的First集合如下:

First(S)={a,b, ε}　First(A)={b, ε}　First(B)={a, ε}
First(C)={a,b,c}　First(D)={a,c}

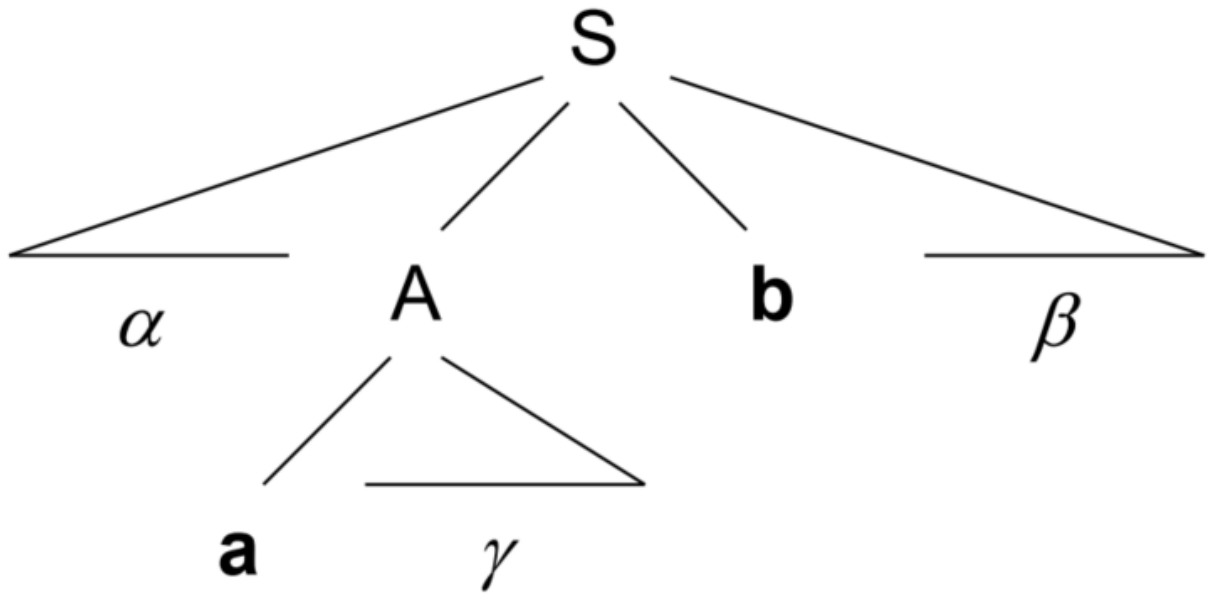
产生式右部符号串的开始符集合为:

S→AB　FIRST(AB)=FIRST(A)∪FIRST(B)∪{ε}={a,b,ε}
S→bC　FIRST(bC)={b}
A→ε　FIRST(ε)={ε}
A→b　FIRST(b)={b}
C→AD　FIRST(AD)=(FIRST(A)-{ε})∪FIRST(D) ={b,a,c}
C→b　FIRST(b)={b}
D→aS　FIRST(aS)={a}
D→c　FIRST(c)={c}

4. 确定的自顶向下语法分析[predictive]

- **FIRST集**和**FOLLOW集**的直观理解

a ∈ FIRST(A) and **b** ∈ FOLLOW(A)



- 后跟符号集——**FOLLOW集**
  - 设G=(V_T, V_N, P, **S**)是上下文无关文法，

  A∈V_N，FOLLOW(**A**)={**b**|**S**=>*…**A**b…，**b** ∈V_T}，
  - 若有**S**=>* …A，则规定 $ ∈FOLLOW(A)

    (注：‘**$**’**做为输入串的结束符**)
  - 直观上说，非终结符A的**后跟符号集是由句型中紧跟A后的那些终结符（包括$）组成。**

4. 确定的自顶向下语法分析[predictive]

- 后跟符号集——**FOLLOW集**

例：文法G[S]:

S→aA|d

A→bAS|ε

➢ 由**S=>* aA** 得 **$ ∈FOLLOW(A)**

由**S=>* abAS=>* abAaA** 得 **a ∈FOLLOW(A)**

… =>* abAd 得 **d ∈FOLLOW(A)**

FOLLOW(A)={$,a,d}

➢ 由 **S=>* S** 得 **$ ∈FOLLOW(S)**

由**S=>aA=>abAS=>abbASS=>abbASaA**

…=>abbASd

FOLLOW(S)={$,a,d}

4. 确定的自顶向下语法分析[predictive]

- 后跟符号集——**FOLLOW集**
  - 计算**每个非终结符A的FOLLOW(A)集**

(1) 对所有$A \in V_N$，令FOLLOW(A)={ }; 对开始符号S，令FOLLOW(S)={$}

因为S=>*S，显然$∈FOLLOW(S)

(2) 对每条产生式A→xBy，考察产生式右部的每一非终结符B，x,y∈V*：

- 如果y不能推出ε：FOLLOW(B)=FOLLOW(B)∪First(y)
- 否则，若y ⇒*ε：FOLLOW(B)=FOLLOW(B)∪(First(y)-{ε})∪ FOLLOW(A)

若a∈FOLLOW(A)，则表明S=>*...Aa...，
由于A→xBy，且y=>*ε，则有：S=>*...Aa...=>...xBya=>...xBa...，
即S=>*...xBa...，所以a∈FOLLOW(B)

注意：ε不存在于任何FOLLOW集中

(3) 重复(2)，直至对所有$A \in V_N$，FOLLOW(A)**收敛为止**。

4. 确定的自顶向下语法分析[predictive]

- 后跟符号集——**FOLLOW集**

例G[S]:

[1]S→AB

[2]S→bC

[3]A→b

[4]A→ε

[5]B→aD

[6]B→ε

[7]C→AD

[8]C→b

[9]D→aS

[10]D→c

已求出非终结符的First集合如下:

First(S)={a,b, ε}　First(A)={b, ε}

First(B)={a, ε}　First(C)={a,b,c}

First(D)={a,c}

已求出能推出ε的非终结符集为**{A,B,S}**

	Follow集(0)	Follow集(1)	Follow集(2)
S	$	$	$
A		a $ c	a $ c
B		$	$
C		$	$
D		$	$

随堂练习（5）

- 对文法G[E]：E→TE’，E’→+TE’|ε，T→FT’，T’→*FT’|ε，F→(E)|n，求**每个非终结符的FIRST集和FOLLOW集**

(1) 能产生ε的非终结符：E’，T’

(2)
FIRST(E)={(, n}
FIRST(E’)={+, ε}
FIRST(T)={(, n}
FIRST(T’)={*, ε}
FIRST(F)={(, n}

(3)
FOLLOW(E)={), $}
FOLLOW(E’)={), $}
FOLLOW(T)={+,), $}
FOLLOW(T’)={+,), $}
FOLLOW(F)={*, +,), $}

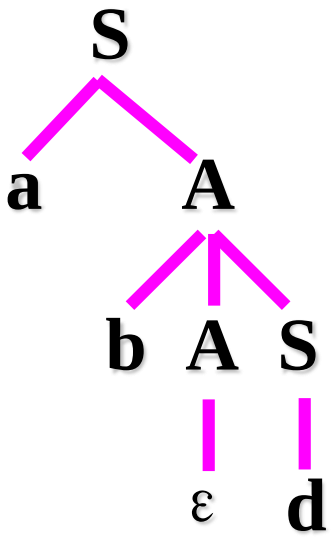
4. 确定的自顶向下语法分析[predictive]

- 例：文法G[S]:

  S→aA|d

  A→bAS|ε

  若输入串W=abd，自顶向下的推导过程为：



FIRST(bAS)={b}

FOLLOW(A)={#,a,d }

可进行确定的自顶向下分析

对于非终结符A的两个产生式A→bAS和A→ε:

当输入符号∈FIRST(bAS)={b}时，选A→bAS推导;

当输入符号∈FOLLOW(A)={#,a,d}时，选A→ε推导。

由于**FIRST(bAS)∩FOLLOW(A)=∅**，所以可进行**确定的**自顶向下分析。

S =>aA =>abAS =>abS =>abd

4. 确定的自顶向下语法分析[predictive]

- **LL(1)**文法
  - 含义
    - ✓ 第一个**L**表示**从左到右**扫描输入串
    - ✓ 第二个**L**表示分析过程用**最左推导**
    - ✓ **1**表明只需**向前看一个符号**便可以决定选哪个产生式进行推导，类似地LL(k)文法需要向前看K个符号才可以确定选用哪个产生式。
  - 定义
    - ✓ 一个上下文无关文法是LL(1)文法的充分必要条件是，若存在产生式A→α|β，则：
      - $FIRST(\alpha) \cap FIRST(\beta) = \Phi$
      - $\varepsilon \in FIRST(\beta) \Rightarrow FIRST(\alpha) \cap FOLLOW(A) = \Phi$

两个条件同时满足！

LL(1)文法就能在仅向前查看一个符号的情况下进行确定的自顶向下分析——LL(1)分析

4. 确定的自顶向下语法分析[predictive]

- 递归下降预测分析[Recursive Descent Predictive Parsing]
- LL(1)分析

5. 递归下降预测分析[Recursive Descent Predictive Parsing]

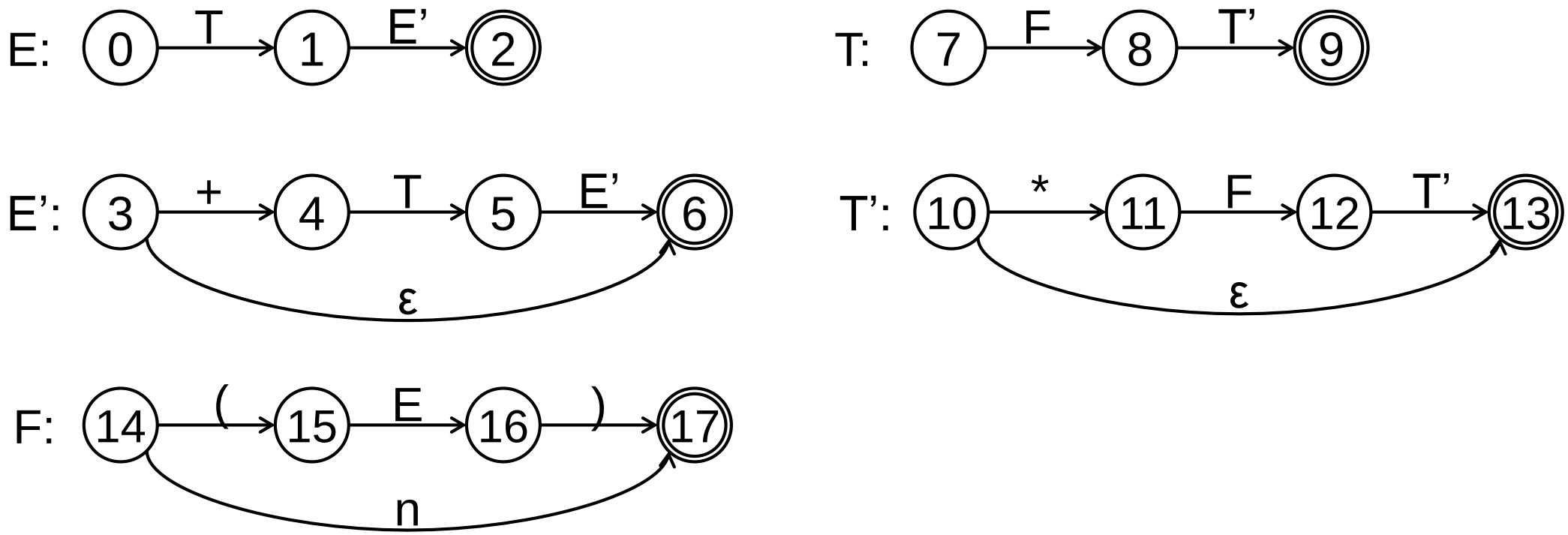
- 递归下降预测分析[Recursive Descent Predictive Parsing]
  - 基于**手写代码**实现，每个非终结符对应一个递归函数
  - 通过函数的递归调用模拟推导过程，**显式地编码产生式的选择**（通常通过if-else或switch分支实现）
  - 可以是LL(1)文法，也可以是更强的**LL(k)（通过向前看更多符号解决冲突）**
- 递归下降预测解析器的开发流程
  - 1. 消除语法中的二义性　　2. 消除语法中的左递归
  - 3. 提取左公因子　　4. 从语法构造转换图
  - 5. 简化转换图　　6. 使用转换图编写解析器作为蓝本

5. 递归下降预测分析[Recursive Descent Predictive Parsing]

- 转换图——可视化预测分析器
  - 对每个非终结符A
    - ✓ 创建初始状态和结束状态
    - ✓ 对于每个$A \rightarrow X_1X_2...X_n$，创建一条从初始状态到最终状态的路径，边标记为$X_1$，$X_2$，…，$X_n$。
    - ✓ 如果$A \rightarrow \varepsilon$：表示路径为ε。

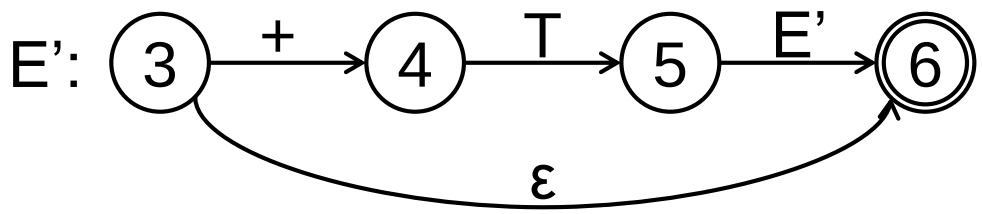
5. 递归下降预测分析[Recursive Descent Predictive Parsing]

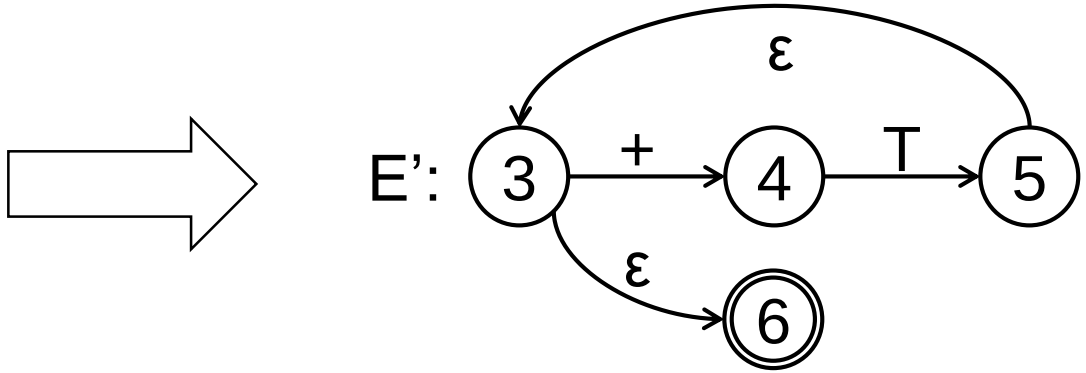
- 转换图——可视化预测分析器
  - 例：对文法G[E]：E→TE’，E’→+TE’|ε，T→FT’，T’→*FT’|ε，F→(E)|n构建转换图

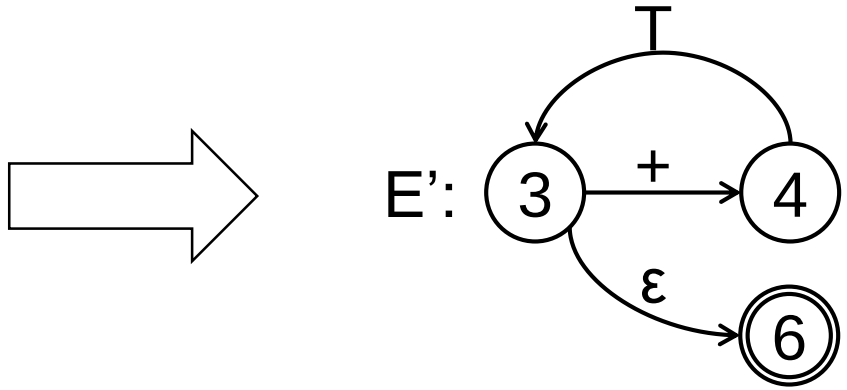


5. 递归下降预测分析[Recursive Descent Predictive Parsing]

- 化简转换图

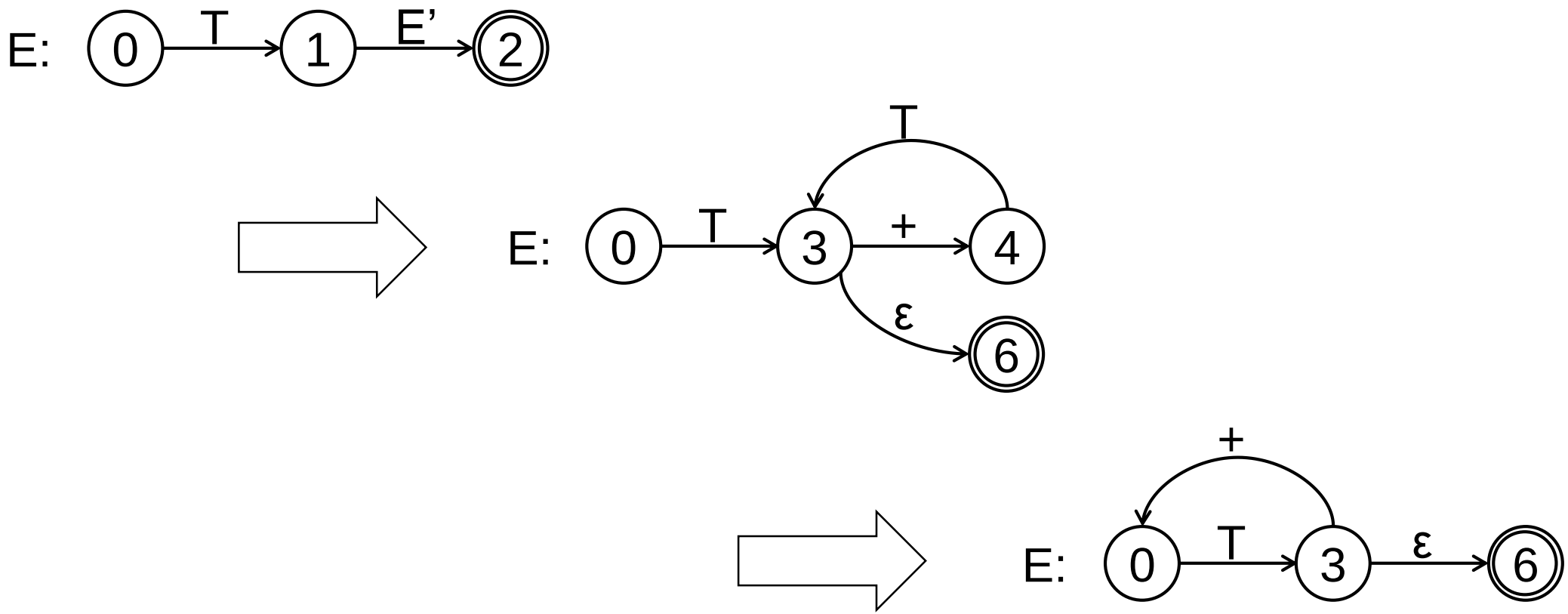






5. 递归下降预测分析[Recursive Descent Predictive Parsing]

- 化简转换图



5. 递归下降预测分析[Recursive Descent Predictive Parsing]

- 化简转换图

文法G[E]:

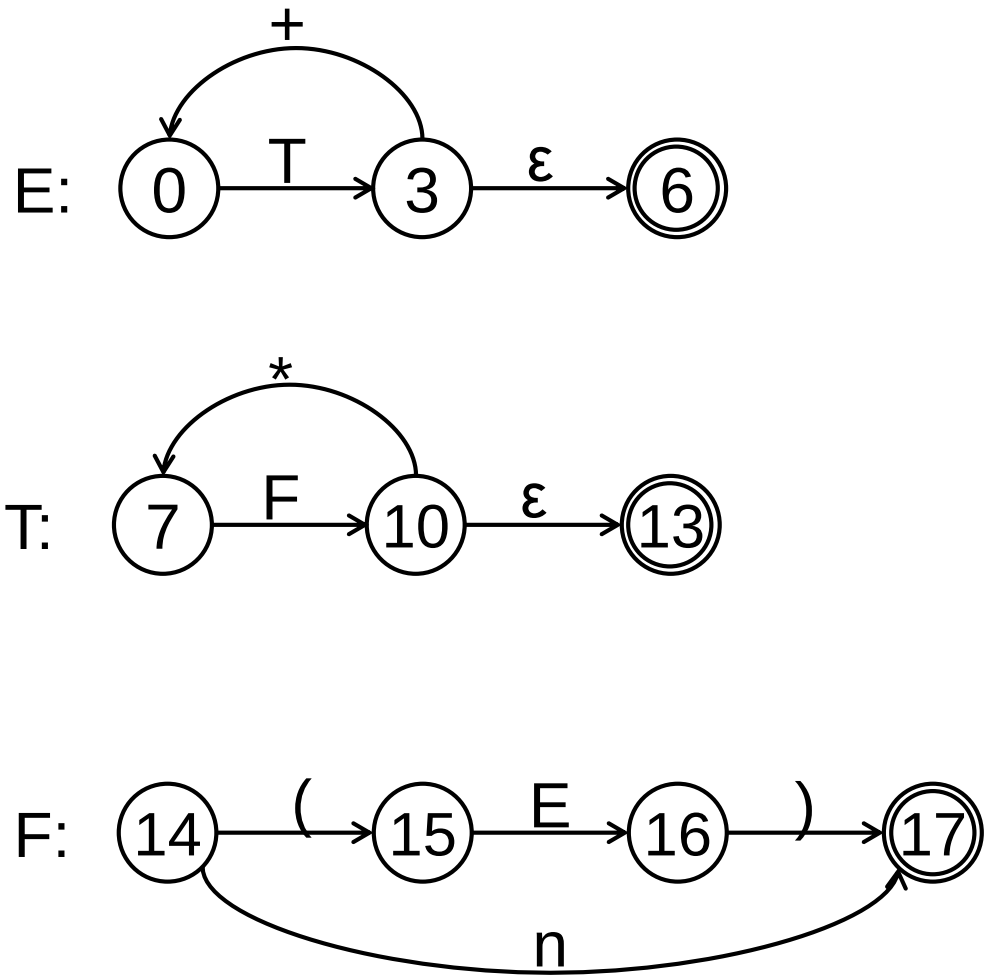
E→TE’

E’→+TE’|ε

T→FT’

T’→*FT’|ε

F→(E)|n

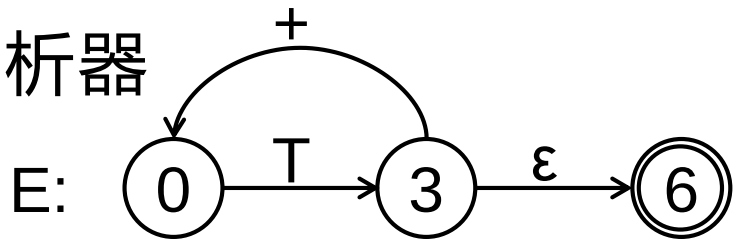


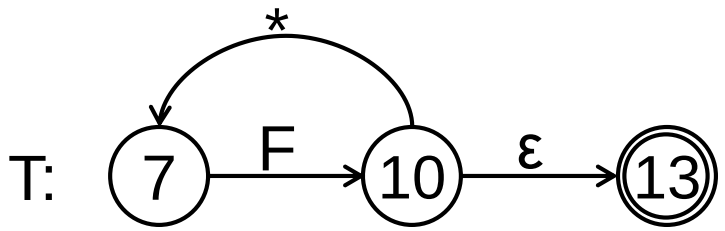
5. 递归下降预测分析[Recursive Descent Predictive Parsing]

- 编写解析器
  - 为每个图编写递归过程，**开始符号S**对应的转换图是程序的主要**main入口**
  - 设计程序的控制流，模仿图中的路径，考虑到前瞻
    - ✓ 如果路径中边的符号等于终结符，则执行**匹配[match]**动作
    - ✓ 如果路径中边的符号是非终结符，则执行**派生[derive]**操作，即调用非终结符的过程（递归）

5. 递归下降预测分析[Recursive Descent Predictive Parsing]

- 编写解析器



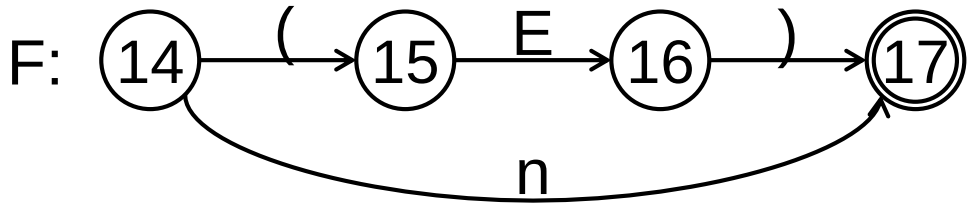



```
void E(){
   if(lookahead in FIRST(T)){
      T();
   }else error();
   while(lookahead=='+'){
      match('+');
      if(lookahead in FIRST(T)){
         T();
      }else error();
   }
   if(lookahead in FOLLOW(E)){
     // do nothing
   }else error();
}
```

```
void T(){
   if(lookahead in FIRST(F)){
      F();
   }else error();
   while(lookahead=='*'){
      match('*');
      if(lookahead in FIRST(F)){
         F();
      }else error();
   }
   if(lookahead in FOLLOW(T)){
     // do nothing
   }else error();
}
```

5. 递归下降预测分析[Recursive Descent Predictive Parsing]

- 编写解析器




```
void F(){
   if(lookahead=='('){
      match('(');
      if(lookahead in FIRST(E)){
         E();
      }else error();
      if(lookahead==')'){
         match(')');
      }else error();
   }else if(lookahead=='n'){
      match('n');
   }else error();
}
```

```
void match(Token tok){
   if(lookahead==tok){
      lookahead=scanner.getNextToken();
   }else error();
}
```