

4-1 自顶向下的分析

分析树的根结点, (开始符号)
最后推导

直到生成输入符号串或发生错误。

是一种试探的分析过程。

分为两类 ①回溯分析 ②预测分析程序。
①回溯分析 不断尝试, 不匹配就换, 退回到前一个非终结符

②预测分析程序。
预先读入一个符号串输入流来预测接下来是什么

两种预测方法 ①递归下降分析法 ②LL(1)分析

将每个非终结符的文法规则描述作为识别A的一个子过程

A的文法结构的右例描述为: 终结符与输入匹配的验证

非终结符对应相应子过程的调用。

示例

算术表达式文法:

```
exp → exp addop term | term
addop → + | -
term → term mulop factor | factor
mulop → *
factor → (exp) | number
```

procedure match(expectedToken);

```
begin
  if token = expectedToken then
    get token;
  else error;
  end if;
end match;
```

procedure factor;

```
begin
  case token of
    (: match ( ( );
    exp: match ( ) );
    number: match ( number );
    else error;
  end case;
end factor;
```

token里面存放的是下一个待分析记号。

EBNF来改写法。

$term \rightarrow factor A'$
 $A' \rightarrow (mulop factor) A' | \epsilon$

EBNF表示右文法(右线性), 靠程序。

```
function exp: integer;
var temp: integer;
begin
  temp := term;
  while token = + or token = - do
    case token of
      + : match ( + );
        temp := temp + term;
      - : match ( - );
        temp := temp - term;
    end case;
  end while;
  return temp;
end exp;
```

共同同步建语法树!

```
procedure term;
begin
  factor;
  while token = * do
    match ( * );
    factor;
  end while;
end term;
```

```
procedure exp;
begin
  term;
  while token = + or token = - do
    match ( + or - );
    term;
  end while;
end exp;
```

构建语法树的过程体现了一种分析的过程

function ifstatement: syntaxTree;

```
var temp: syntaxTree;
begin
  match ( if );
  match ( ( );
  temp := makeSintNode ( if );
  testChild ( temp ) := exp;
  match ( );
  thenChild ( temp ) := statement;
end ifstatement;
```

```
if
  0 other other
  if token = else then
    match ( else );
    elseChild ( temp ) := statement;
  else
    elseChild ( temp ) := nil;
  end if;
```

LL(1) 分析法, 自顶向下

从右至左推导
分析过程对最左推导。

方法, 显式栈。

两个动作: 匹配生成

栈顶非终结符, 两个可推导动作

Steps	Pushing stack	Input	Action
1	\$ S	() \$	S → (S) S
2	\$ S) S) \$	match.
3	\$ S) S) \$	S → ε
4	\$ S)) \$	match.
5	\$ S	\$	S → ε
6	\$	\$	accept

两个均能到达空状态, 只是可接受状态。

同一非终结符有多个产生式怎么选?

建立 LL(1) 分析表
对输入串中的符号进行预测。

分析表构造规则。

- ① 有 $A \rightarrow \alpha$ 且 α 以 a 开始, 则 $M[A, a] = A \rightarrow \alpha$
- ② 有 $A \rightarrow \alpha$ 且 $\alpha \Rightarrow \epsilon$, 且 $S \neq \alpha$ 非终结符, 则 $M[A, a] = A \rightarrow \alpha$

练习 $G(S): S \rightarrow +SA | *SA | n$
 $A \rightarrow (S) | n$

```
procedure A;
begin
  match ( ( );
  S;
  match ( ) );
begin
  case token of:
    C: match ( C );
      S;
      match ( ) );
    n: match ( n );
  else error;
  end case;
end A;
```

```
procedure S;
begin
  case token of
    +: match ( + );
      S;
      A;
    *: match ( * );
      S;
      A;
    n: match ( n );
  else error;
  end case;
end S;
```

```
procedure match;
begin
  if token = expectedToken then
    get token;
  else error;
  end if;
end match;
```

请给出右文法的 LL(1) 分析表。

M[N,T]	if	other	else	0	1	\$
statement	statement	statement	else-part			
	→ if-stmt	→ other	→ else-part			

因 $M[A, a] = A \rightarrow a$
 ② 有 $A \rightarrow a$ 且 $a \in \epsilon$, 且 $S \neq \epsilon$ 为 $\beta A \alpha$
 其中 a 是终结符. $M[A, a] = A \rightarrow a$

每个子项最多只有一个产生式

限制, 不能有左递归
左因子
二义性.

消除左递归.
提取左因子

$A \rightarrow Aa|B$

新的左递归.

$A \rightarrow BA'$

$A' \rightarrow aA'| \epsilon$

$G(s):$
 $S \rightarrow BAB$
 $A \rightarrow Aa|B$
 $B \rightarrow ab|ac$

练习: $S \rightarrow S^*T | S/T|T|(T)$
 $S \rightarrow TA' | (T)A'$
 $A' \rightarrow *TA' | /TA' | \epsilon$

$\$ S S \rightarrow C \rightarrow \phi$ match
 $\$ S S \rightarrow C \rightarrow \phi$ $S \rightarrow S$
 $\$ S \rightarrow C \rightarrow \phi$ match.
 $\$ S \rightarrow C \rightarrow \phi$ $S \rightarrow CS \rightarrow S$
 $\$ S \rightarrow S \rightarrow \phi$ match.
 $\$ S \rightarrow \phi$ $S \rightarrow \epsilon$
 $\$ S \rightarrow \phi$ match
 $\$ S \rightarrow \phi$ $S \rightarrow \epsilon$
 $\$ \rightarrow \phi$ accept.

练习: $G(s)$
 $S \rightarrow aaAB$
 $A \rightarrow bAa|Ba$
 $B \rightarrow cb| \epsilon$

	a	b	c	ϕ
S	$S \rightarrow aaAB$			
A	$A \rightarrow bAa Ba$	$A \rightarrow Ba$		
B	$B \rightarrow \epsilon$	$B \rightarrow cb$	$B \rightarrow \epsilon$	

step

stack	input	ACTION
$\$ S$	aaaa ϕ	$S \rightarrow aaAB$
$\phi BaAaa$	aaaa ϕ	match
$\phi BaAa$	aaa ϕ	match
ϕBaA	aa ϕ	$A \rightarrow Ba$
$\phi Baab$	a ϕ	$B \rightarrow \epsilon$
ϕBa	a ϕ	match
ϕBa	a ϕ	match
ϕB	ϕ	$B \rightarrow \epsilon$
ϕ	ϕ	accept.

LL(1)分析表.

else-part \rightarrow else statement | ϵ
 $\exp \rightarrow 0 | 1$

M[N,T]	if	other	else	0	1	$\$$
statement	statement \rightarrow if-stmt	statement \rightarrow other				
if-stmt	if-stmt \rightarrow if (exp) statement else-part					
else-part			else-part \rightarrow else statement else-part $\rightarrow \epsilon$			else-part $\rightarrow \epsilon$
exp				exp $\rightarrow 0$	exp $\rightarrow 1$	

$G(s) \Rightarrow S \rightarrow AB$
 $A \rightarrow aAb| \epsilon$
 $B \rightarrow ba| \epsilon$

$S \xrightarrow{*} ba.$
 $S \xrightarrow{*} \beta A \alpha$
 $M[A, a] = A \rightarrow a$

$\downarrow abb$

ϕS	abb ϕ	$S \rightarrow AB$
ϕBA	abb ϕ	$A \rightarrow aAb$
$\phi BbAa$	abb ϕ	match
ϕBbA	bb ϕ	$A \rightarrow \epsilon$
ϕBb	bb ϕ	match
ϕB	b ϕ	$B \rightarrow ba$
ϕab	b ϕ	match
ϕa	ϕ	err 无法消