编译原理第三次作业

任凯 2020141460080

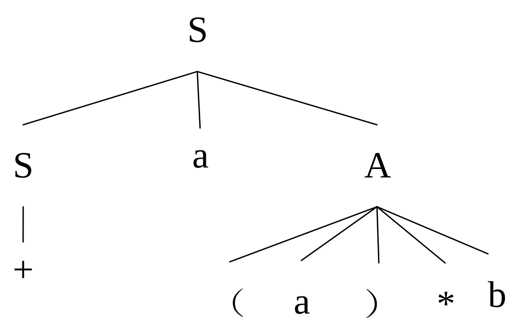
1. 已知文法 $G(S)\colon S \to SaA \mid +$
   $A \to (a) * b \mid (a)$

（1） （4 分）写出串 $+a(a)*b$ 的最右推导，并画出分析树

答：该串的最右推导如下：

$$
\begin{aligned}
S &\Rightarrow SaA \\
&\Rightarrow Sa(a)*b \\
&\Rightarrow +a(a)*b
\end{aligned}
$$

分析树如下：



（2） （6 分）改写文法为 EBNF，并构造递归下降程序伪代码识别该文法

答：该文法改写为 EBNF 为：$S \to +\{a(a)[*b]\}$。

递归下降程序伪代码如下：

```
procedure G(s);
begin
    match( + );
    while token = a do
        match( a );
        match( ( );
        match( a )
        match( ) );
    if token = * then
       match( * );
       match( b );
    end if;
    end while;
end G(s)
```

2. Consider the following grammar,

$$stmt \rightarrow assign-stmt \mid call-stmt \mid if-stmt \mid other$$
$$assign-stmt \rightarrow ID = \mathrm{exp}$$
$$call-stmt \rightarrow ID(\mathrm{exp}-list)$$
$$if-stmt \rightarrow if(\mathrm{exp})stmt \mid if(\mathrm{exp})stmt\ else\ stmt$$

（1） Rewrite the grammar using EBNF.

$$stmt \rightarrow assign-stmt \mid call-stmt \mid if-stmt \mid other$$
$$assign-stmt \rightarrow ID = \mathrm{exp} | ID(\mathrm{exp}-list)$$
$$if-stmt \rightarrow if(\mathrm{exp})stmt[else\ stmt]$$

（2） Write the pseudocode for the written grammar using recursive-descent parsing.

```
procedure stmt;
begin
     case token of
         ID:
             ID-stmt;
         if:
             if-stmt;
         else:
             other;
     end case;
end stmt;

precedure ID-stmt;
begin
    match( ID );
    case token of
        =:
            match( = );
            exp;
        (:
            match( ( );
            exp-list;
            match( ) );
        else error;
    end case;
end ID-stmt;

precedure if-stmt;
begin
    match( if );
    match( ( );
    exp;
    match( ) );
    stmt;
    if token = else then
```

```
        match(else);
        stmt;
    end if;
end if-stmt;
```