

# 实验十 LL (1) 分析表的构造

原创

计算机的小粽子 于 2017-12-13 19:55:39 发布 9772 收藏 37

分类专栏: [-----编译原理](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/tangyuanzong/article/details/78796078>

版权



[-----编译原理](#) 专栏收录该内容

7 篇文章 1 订阅

订阅专栏

## 一、实验目的

学习和掌握FIRST集合、FOLLOW集合的计算, LL (1) 分析表的构造方法。

## 二、实验任务

- (1) 存储文法;
- (2) 计算给定文法所有非终结符的FIRST集合;
- (3) 计算给定文法所有非终结符的FOLLOW集合;
- (4) 构造该文法的LL (1) 文法的分析表并按实验九的文法格式存储;
- (5) 结合实验九, 完成完整的LL (1) 分析过程。

## 三、实验内容

- (1) 确定文法的文件存储格式, 存储文法的非终结符集合、开始符号、终结符集合和产生式规则集合。要求为3个以上测试文法准备好相应的存储文件。
- (2) 计算给定文法所有非终结符的FIRST集合。
- (3) 计算给定文法所有非终结符的FOLLOW集合;
- (4) 构造该文法的LL (1) 文法的分析表并按实验九的文法格式存储;
- (5) 结合实验九, 完成完整的LL (1) 分析过程。

## 四、实验结果

文法一:

```
E -> TY
Y -> +TY | e
T -> FZ
Z -> *FZ | e
F -> (E) | i
```

测试结果:

FIRST集合为:

$\text{FIRST}(E) = \{ (, i \}$

$\text{FIRST}(F) = \{ (, i \}$

$\text{FIRST}(T) = \{ (, i \}$

$\text{FIRST}(Y) = \{ +, e \}$

$\text{FIRST}(Z) = \{ *, e \}$

FOLLOW集合为:

$\text{FOLLOW}(E) = \{ \$, ) \}$

$\text{FOLLOW}(F) = \{ \$, ), *, + \}$

$\text{FOLLOW}(T) = \{ \$, ), + \}$

$\text{FOLLOW}(Y) = \{ \$, ) \}$

$\text{FOLLOW}(Z) = \{ \$, ), + \}$

LL1分析表为:

	\$	(	)	*	+	i
E		$E \rightarrow TY$				$E \rightarrow TY$
F		$F \rightarrow (E)$				$F \rightarrow i$
T		$T \rightarrow FZ$				$T \rightarrow FZ$
Y	$Y \rightarrow e$		$Y \rightarrow e$		$Y \rightarrow +TY$	
Z	$Z \rightarrow e$		$Z \rightarrow e$	$Z \rightarrow *FZ$	$Z \rightarrow +e$	

文法二:

$S \rightarrow T$   
 $T \rightarrow (S)ST \mid e$

测试结果:

FIRST集合为:

FIRST(S) = { (, e }

FIRST(T) = { (, e }

FOLLOW集合为:

FOLLOW(E) = { \$, ) }

FOLLOW(F) = { \$ }

FOLLOW(T) = { \$ }

FOLLOW(Y) = { \$, ) }

FOLLOW(Z) = { \$ }

LL1分析表为:

	\$	(	)	*	+	i
E	E → TY	E → TY	E → TY			
F		F → (E)				F → i
T						
Y	Y → e		Y → e		Y → +TY	
Z	Z → e			Z → *FZ		<a href="http://blog.csdn.net/tangyuanzong">http://blog.csdn.net/tangyuanzong</a>

文法三:

```
S -> iEtST | a
T -> eS | e
E -> b
```

测试结果:

FIRST集合为:

FIRST(E)={ c }

FIRST(S)={ a , i }

FIRST(T)={ b , e }

FOLLOW集合为:

FOLLOW(E)={ \$ , t }

FOLLOW(S)={ \$ , b }

FOLLOW(T)={ \$ , b }

LL1分析表为:

	\$	a	b	c	i	t
E				E→c		
S		S→a			S→iEtST	
T	T→e		T→bS			

在第3行第3列处发生冲突: T→e 与 T→bS

该文法不是LL(1)文法!!!

<http://blog.csdn.net/tangyuanzong>

## 完整代码

```

#include<iostream>
#include<string>
#include<fstream>
#include<map>
#include<set>
#include<iomanip>
using namespace std;
ifstream input("in1.txt");
map<char, set<char>>first; //first集合
map<char, set<char>>follow; //foll集合
map<char, set<char>>first_temp;
map<char, set<char>>follow_temp;
map<char, set<string>>string_temp;
set<char>table; //终结符号
set<char>no_end; //非终结符号
string s[10][10]; //分析表
map<char, int>xlab;
map<char, int>ylab;
void First(){
    string temp;
    while(getline(input, temp)){
        int len = temp.length();
        if(first.find(temp[0])==first.end()){
            int l=5;
            while(temp[l]!='\0'&&l<len) l++;
            if(!(temp[l]>='A'&&temp[l]<='Z'))
                first[temp[0]].insert(temp[l]);
            else
                first_temp[temp[0]].insert(temp[l]);
            if(l<len&&temp[l]=='|'){

```

```

        if(!(temp[l+2]>='A'&&temp[l+2]<='Z'))
            first[temp[0]].insert(temp[l+2]);
        else first_temp[temp[0]].insert(temp[l+2]);
    }
}
}
map<char,set<char>>::iterator it=first_temp.begin();
while(first_temp.size()!=0){
    set<char>::iterator ij=it->second.begin();
    while(ij!=it->second.end()){
        if(first.find(*ij)!=first.end()){
            set<char>::iterator ik=first[*ij].begin();
            while(ik!=first[*ij].end())
                first[it->first].insert(*ik),ik++;
            it->second.erase(ij++);
        }
        else ij++;
    }
    if(it->second.size()==0)
        first_temp.erase(it++);
    else
        it++;
    if(it==first_temp.end())
        it=first_temp.begin();
}
input.close();
}

void Follow(){
    input.open("in.txt");
    string temp;
    table.insert('$');
    while(getline(input,temp)){
        int l=5,len=temp.length();
        no_end.insert(temp[0]);
        while(l<len-1&&temp[l+1]!=' '){
            if(!(temp[l]>='A'&&temp[l]<='Z')&&temp[l]!='e')
                table.insert(temp[l]);
            if(temp[l]>='A'&&temp[l]<='Z') {
                follow[temp[l]].insert('$');
                if(!(temp[l+1]>='A'&&temp[l+1]<='Z'))
                    follow[temp[l]].insert(temp[l+1]);
            }
            else{
                int ok=1;
                set<char>::iterator it=first[temp[l+1]].begin();
                while(it!=first[temp[l+1]].end()){
                    if(*it!='e')
                        follow[temp[l]].insert(*it);
                    else ok=0;
                    it++;
                }
                if(!ok)
                    follow_temp[temp[l]].insert(temp[0]);
            }
        }
        l++;
    }
    int k=0;
    while(temp[k+5]!='\0'&&temp[k+5]!='!')

```

```

        k++;
    if(temp[k+5]=='\0')
        string_temp[temp[0]].insert(temp.substr(5,k));
    else
        string_temp[temp[0]].insert(temp.substr(5,k-1));
    if(temp[k+5]=='|'){
        int y=0;
        while(temp[k+5+y]!='\0')
            y++;
        string_temp[temp[0]].insert(temp.substr(k+7,y));
    }
    if(1<len&&temp[1]>='A'&&temp[1]<='Z')
        follow_temp[temp[1]].insert(temp[0]);
    while(1<len) {
        if(!(temp[1]>='A'&&temp[1]<='Z')&&temp[1]!=' '&&temp[1]!='|'&&temp[1]!='e')
            table.insert(temp[1]);
        l++;
    }
}
map<char,set<char>>::iterator it=follow_temp.begin();
while(follow_temp.size()!=0){
    set<char>::iterator ij=it->second.begin();
    while(ij!=it->second.end()){
        if(follow.find(*ij)!=follow.end()){
            set<char>::iterator ik=follow[*ij].begin();
            while(ik!=follow[*ij].end())
                follow[it->first].insert(*ik),ik++;
            it->second.erase(ij++);
        }
        else
            ij++;
    }
    if(it->second.size()==0)
        follow_temp.erase(it++);
    else
        it++;
    if(it==follow_temp.end())
        it=follow_temp.begin();
}
input.close();
}

```

```

void show_first(){ //打印FIRST集合
    cout<<"FIRST集合为: "<<endl;
    map<char,set<char>>::iterator it=first.begin();
    while(it!=first.end()){
        cout<<"FIRST"<<"("<<it->first<<")"<<"={ ";
        set<char>::iterator ij=it->second.begin();
        int l=0,len=it->second.size();
        while(ij!=it->second.end()){
            cout<<*ij;ij++;
            if(l<=len-2)
                cout<<" , ";
            l++;
        }
        it++;
        cout<<" }"<<endl;
    }
}

```

```

}

void show_follow(){ //打印FOLLOW集合
    cout<<"FOLLOW集合为: "<<endl;
    map<char,set<char>>>::iterator it=follow.begin();
    while(it!=follow.end()){
        cout<<"FOLLOW"<<"("<<it->first<<")"<<"={ ";
        set<char>::iterator ij=it->second.begin();
        int l=0,len=it->second.size();
        while(ij!=it->second.end()){
            cout<<*ij;ij++;
            if(l<=len-2)
                cout<<" , ";
            l++;
        }
        it++;
        cout<<" }"<<endl;
    }
}

void show_table(){ //打印并存储LL1分析表
    cout<<"LL1分析表为: "<<endl;
    cout<<"-----"<<endl;
    cout<<" ";
    set<char>::iterator it=table.begin();
    int k=0;
    while(it!=table.end()){
        ylab[*it]=k++;
        cout<<*it<<" ";it++;
    }
    cout<<endl;
    it = no_end.begin();
    int l=0;
    while(it!=no_end.end()){
        cout<<*it<<" ";
        set<char>::iterator ij=table.begin();
        int j=0;
        while(ij!=table.end()){
            set<string>::iterator ik=string_temp[*it].begin();
            int ok=0;
            while(ik!=string_temp[*it].end()){
                string temp=*ik;
                if(first[temp[0]].count(*ij)!=0||temp[0]==*ij){
                    ok=1;cout<<*it<<"->"<<temp; s[1][j]=temp;
                    for(int x=temp.length()+3;x<10;x++) cout<<" ";
                }
                if(ok==0 &&(first[temp[0]].count('e')!=0||(*ik)[0]=='e')){
                    if(follow[*it].count(*ij)!=0){
                        ok=1;cout<<*it<<"->"<<temp; s[1][j]=temp;
                        for(int x=temp.length()+3;x<10;x++)
                            cout<<" ";
                    }
                }
                ik++;
            }
            if(!ok)
                cout<<" ",s[1][j]=" ";
            ij++;
            j++;
        }
    }
}

```

```
        cout<<endl;
        xlab[*it]=l;
        it++;
        l++;
    }
}

int main(){
    First(); //求first集合
    show_first(); //打印first集合
    Follow(); //求follow集合
    show_follow(); //打印follow集合
    show_table(); //打印L1分析表
}
```