

编译原理第二次实验

编译原理第二次实验

介绍

思路

输入

输出

介绍

本次实验采用自上而下的语法分析技术，对于给定的若干个上下文无关语法规则，构造其LL(1)预测分析表。再对于某个特定的表达式，确定其是否是该语法的某个句子。

测验用的文法：

(0) $E \rightarrow TG$

(1) $G \rightarrow \varepsilon$ | (2) $G \rightarrow +TG$

(3) $T \rightarrow FH$

(4) $H \rightarrow \varepsilon$ | (5) $H \rightarrow *FH$

(6) $F \rightarrow (E)$ | (7) $F \rightarrow i$

实验采用C#语言

	i	+	*	()	\$
E	0			0		
G		2			1	1
T	3			3		
H		4	5		4	4
F	7			6		

思路

消除二义性和左递归

```
public void change(List l);
```

检查文法是否是LL(1)的

```
public bool check(List l);
```

构造终结符非终结符表

```
public void createbase();
```

构造每个产生式右部的First集合

```
public void createRightFirst();
```

构造每个产生式左部的Follow集合

```
public void createLeftFollow();
```

构建预测分析表

```
public void createLL1();
```

分析输入串

```
public void analyse();
```

理论上来讲可以通过改变文法定义来重新生成分析表进行分析，不过有一些情况会出现错误，目前还没有想到解决办法

输入

string input="i+i*i";

文法也作为输入串

输出

LL1分析表

	+	*	()	i	\$
E:			E: TG		E: TG	
G:	G: +TG		G: ϵ		G: ϵ	
T:		T: FH		T: FH		
H:	H: ϵ	H: *FH	H: ϵ		H: ϵ	
F:		F: (E)		F: i		

分析过程

	E\$	i+i*i\$		
	TG\$	i+i*i\$	输出	E: TG
	FHG\$	i+i*i\$	输出	T: FH
	iHG\$	i+i*i\$	输出	F: i
i	HG\$	+i*i\$	匹配	i
i	G\$	+i*i\$	输出	H: ϵ
i	+TG\$	+i*i\$	输出	G: +TG
i+	TG\$	i*i\$	匹配	+
i+	FHG\$	i*i\$	输出	T: FH
i+	iHG\$	i*i\$	输出	F: i
i+i	HG\$	*i\$	匹配	i
i+i	*FHG\$	*i\$	输出	H: *FH
i+i*	FHG\$	i\$	匹配	*
i+i*	iHG\$	i\$	输出	F: i
i+i*i	HG\$	\$	匹配	i
i+i*i	G\$	\$	输出	H: ϵ
i+i*i	\$	\$	输出	G: ϵ