

语义分析器的构造——实验报告

19335074

黄玟瑜

实验要求

使用递归下降翻译法或LL(1)翻译法实现高级编程语言的语义分析，将其翻译为四元式格式的中间语言，至少支持算术表达式的语义分析。算术表达式至少支持加减乘除以及括号操作，即（+，-，*，/，（））。

提交内容

1. 实验报告，报告内容必须包括：
 - 翻译文法；
 - 若采用递归下降翻译法，须给出文法（至少实现算术表达式的文法）的子程序流程图，并在其上标注返回地址；
 - 给出一个算术表达式实例的分析表（表项内容参考实验三PPT P17）；
 - 运行结果展示；
 - 以及其他必要内容。
2. 语义分析源程序：source.c（源程序包）
3. 可执行文件
4. 程序测试文件：test.txt（实验输入，将测试案例写入程序的可没有此项）

实验报告

笔者在本次实验中采用LL(1)翻译法进行语义分析，程序的执行过程为：

1. 读入文法
2. 生成LL(1)分析表
3. 对输入串进行翻译
4. 输出结果

将执行该方法的功能封装为翻译器类 `Translator`。如下所示：

```
class Translator{
    vector<string> productionRules;
    vector<string> productionRulesWithActions;
    char startChar;
    vector<char> nonTerminators;
    vector<char> Terminators;
    vector<set<char>> pFirstSets;    // the same size of productionRules
    vector<set<char>> nFirstSets;    // the same size of nonTerminators
    vector<set<char>> nFollowSets;    // the same size of nonTerminators
    vector<vector<int>> analysisTable;

    int indexOf(const vector<char>& v, char x);    // return index of x in v

    void getFirstOfNonTerminator(char nonTerminator);    // get first set of
nonterminator
    void getFirstOfProductionRules();    // get first set of
terminator
```

```

        set<char> firstBate(const string& remain);

public:
    Translator();
    void printCharacters();
    void printFirst();
    void printFollow();
    void printTable();
    void printCurrentStatus(const string& s1, const string& s2, const string&
s3, const string& s4);
    void printQuat(const Quat& q);
    vector<Quat> AnalyseSentence(const string& sentence);
    void run(char* inputfile);        // interface for main function
};

```

其中的主要函数为：

1. `Translator();`

初始化函数，依次读入算术表达式的文法以及翻译文法，并根据文法得到LL(1)分析表。

2. `vector<Quat> AnalyseSentence(const string& sentence);`

执行翻译过程的函数，对输入串 `sentence` 进行翻译，返回翻译得到的四元式。

3. `void run(char* inputfile);`

提供给main函数调用的函数，该函数读入 `inputfile` 的语句并逐条分析，并输出结果。

此外其他函数都是供以上3个函数调用的工具函数，具体用法在源代码中有注释说明。

翻译文法

翻译文法设计如下：

```

E -> T E' (1)
E' -> + T{GEQ(+)}E' (2) | - T{GEQ(-)}E' (3) | ε (4)
T -> F T' (5)
T' -> * F{GEQ(*)}T' (6) | / F{GEQ(/)}T' (7) | ε (8)
F -> i{PUSH(i)} (9) | ( E ) (10)

```

算术表达式的文法存储在文件 `grammar.txt` 中，首先根据该文法得到LL(1)分析表，随后根据翻译文法进行翻译过程。

翻译文法保存在 `translation_grammar.txt` 中。

得到LL(1)分析表如下：

	+	-	*	/	i	()	\$
E					E->TA	E->TA		
A	A->+TA	A->-TA					A->@	A->@
T					T->FB	T->FB		
B	B->@	B->@	B->*FB	B->/FB			B->@	B->@
F					F->i	F->(E)		

实例分析

分析表达式 `a+b*c`，如下：

Analysing: a+b*c				
SYN Stack	Input	Operation	SEM stack	Quat
\$E	a+b*c\$	E->TA		
\$AT	a+b*c\$	T->FB		
\$ABF	a+b*c\$	F->iPi		
\$ABaPa	a+b*c\$	Match a		
\$ABaP	+b*c\$	Push a		
\$AB	+b*c\$		a	
\$AB	+b*c\$	B->@	a	
\$A	+b*c\$	A->+TG+A	a	
\$A+GT+	+b*c\$	Match +	a	
\$A+GT	b*c\$	T->FB	a	
\$A+GBF	b*c\$	F->iPi	a	
\$A+GBbPb	b*c\$	Match b	a	
\$A+GBbP	*c\$	Push b	a	
\$A+GB	*c\$		ab	
\$A+GB	*c\$	B->*FG*B	ab	
\$A+GB*GF*	*c\$	Match *	ab	
\$A+GB*GF	c\$	F->iPi	ab	
\$A+GB*GcPc	c\$	Match c	ab	
\$A+GB*GcP	\$	Push c	ab	
\$A+GB*G	\$		abc	
\$A+GB*G	\$	Geq *	abc	
\$A+GB	\$		at	(*, b, c, t1)
\$A+GB	\$	B->@	at	
\$A+G	\$	Geq +	at	
\$A	\$		t	(+, a, t1, t2)
\$A	\$	A->@	t	
\$	\$	Accept!	t	
Analyze successfully!				
Quats:				
(1) (*, b, c, t1)				
(2) (+, a, t1, t2)				

分析过程执行正确，和样例一致。

分析一个较为复杂的算术表达式 `a+b*(c/d-e)`，如下：

Analysing: a+b*(c/d-e)				
SYN Stack	Input	Operation	SEM stack	Quat
\$E	a+b*(c/d-e)\$	E->TA		
\$AT	a+b*(c/d-e)\$	T->FB		
\$ABF	a+b*(c/d-e)\$	F->iPi		
\$ABaPa	a+b*(c/d-e)\$	Match a		
\$ABaP	+b*(c/d-e)\$	Push a		
\$AB	+b*(c/d-e)\$		a	
\$AB	+b*(c/d-e)\$	B->@	a	
\$A	+b*(c/d-e)\$	A->+TG+A	a	
\$A+GT+	+b*(c/d-e)\$	Match +	a	

\$A+GT	b*(c/d-e)\$	T->FB	a	
\$A+GBF	b*(c/d-e)\$	F->iPi	a	
\$A+GBbPb	b*(c/d-e)\$	Match b	a	
\$A+GBbP	*(c/d-e)\$	Push b	a	
\$A+GB	*(c/d-e)\$		ab	
\$A+GB	*(c/d-e)\$	B->*FG*B	ab	
\$A+GB*GF*	*(c/d-e)\$	Match *	ab	
\$A+GB*GF	(c/d-e)\$	F->(E)	ab	
\$A+GB*G)E((c/d-e)\$	Match (ab	
\$A+GB*G)E	c/d-e)\$	E->TA	ab	
\$A+GB*G)AT	c/d-e)\$	T->FB	ab	
\$A+GB*G)ABF	c/d-e)\$	F->iPi	ab	
\$A+GB*G)ABcPc	c/d-e)\$	Match c	ab	
\$A+GB*G)ABcP	/d-e)\$	Push c	ab	
\$A+GB*G)AB	/d-e)\$		abc	
\$A+GB*G)AB	/d-e)\$	B->/FG/B	abc	
\$A+GB*G)AB/GF/	/d-e)\$	Match /	abc	
\$A+GB*G)AB/GF	d-e)\$	F->iPi	abc	
\$A+GB*G)AB/GdPdd-e)\$		Match d	abc	
\$A+GB*G)AB/GdP	-e)\$	Push d	abc	
\$A+GB*G)AB/G	-e)\$		abcd	
\$A+GB*G)AB/G	-e)\$	Geq /	abcd	
\$A+GB*G)AB	-e)\$		abt	(/, c, d, t1)
\$A+GB*G)AB	-e)\$	B->@	abt	
\$A+GB*G)A	-e)\$	A->-TG-A	abt	
\$A+GB*G)A-GT-	-e)\$	Match -	abt	
\$A+GB*G)A-GT	e)\$	T->FB	abt	
\$A+GB*G)A-GBF	e)\$	F->iPi	abt	
\$A+GB*G)A-GBePee)\$		Match e	abt	
\$A+GB*G)A-GBeP)\$	Push e	abt	
\$A+GB*G)A-GB)\$		abte	
\$A+GB*G)A-GB)\$	B->@	abte	
\$A+GB*G)A-G)\$	Geq -	abte	
\$A+GB*G)A)\$		abt	(-, t1, e, t2)
\$A+GB*G)A)\$	A->@	abt	
\$A+GB*G))\$	Match)	abt	
\$A+GB*G	\$	Geq *	abt	
\$A+GB	\$		at	(*, b, t2, t3)
\$A+GB	\$	B->@	at	
\$A+G	\$	Geq +	at	
\$A	\$		t	(+, a, t3, t4)
\$A	\$	A->@	t	
\$	\$	Accept!	t	

Analyze successfully!

Quats:

- (1) (/ , c, d, t1)
- (2) (- , t1, e, t2)
- (3) (*, b, t2, t3)
- (4) (+, a, t3, t4)

分析过程执行正确。

运行结果

运行结果在文件 `cmd_output.txt` 中，该文件保存了在终端运行程序的输出结果。

经验证，分析过程执行正确，结果符合预期。