****

**同济大学编译原理课程设计**

**——基于Xml的简单数据库**

**姓 名：傅 聪**

**学 号： 1252458**

**专 业：计算机科学**

**年 级：大 四**

**目录**

[**一、系统方案设计说明 3**](#_Toc429922150)

[**1.1、项目背景 3**](#_Toc429922151)

[**1.2、项目设计 3**](#_Toc429922152)

[**二、程序功能描述 4**](#_Toc429922153)

[**2.1、通用功能 4**](#_Toc429922154)

[**2.2、主要功能 4**](#_Toc429922155)

[**三、程序具体实现 4**](#_Toc429922156)

[**3.1、主要数据结构 4**](#_Toc429922157)

[**3.2、部分关键算法 7**](#_Toc429922158)

[**3.2.1、FIRST/FOLLOW集 7**](#_Toc429922159)

[**3.2.2、项目集规范族算法 8**](#_Toc429922160)

[**3.2.3、预测分析表及冲突解决方法 8**](#_Toc429922161)

[**3.2.4、算符优先分析方法 9**](#_Toc429922162)

[**四、执行界面和运行结果 10**](#_Toc429922163)

[**4.1、文法分析结果 10**](#_Toc429922164)

[**4.2、词法、语法分析结果 11**](#_Toc429922165)

[**4.3、命令，算符文法解析结果 11**](#_Toc429922166)

[**五、设计体会 12**](#_Toc429922167)

# 一、系统方案设计说明

## 1.1、项目背景

XML数据库是一种支持对[XML](http://baike.baidu.com/view/63.htm)（[标准通用标记语言](http://baike.baidu.com/view/5286041.htm)下的一个应用）格式文档进行存储和查询等操作的数据管理系统。在系统中，开发人员可以对数据库中的XML文档进行查询、导出和指定格式的序列化。

## 1.2、项目设计

项目主要包含以下几部分工作：

(1)、设计编写一个SLR文法分析器，按要求输入文法后该程序可以生成一张预测分析表。

(2)、设计编写计算first/follow集的程序。

(3)、设计编写一个词法分析子程序，能识别xml文档中的每个单词，供语法分析使用。

(4)、设计一个语法分析程序，输入文法分析的结果（预测分析表），每次从词法分析器中获取下一个单词，做相应的移进/规约动作。规约的同时生成相应的数据表，存储在程序中供使用。

(5)、从linux源码中挑出shell命令处理部分的程序，放在本系统中用于解析接受的命令。

(6)、解析简单sql语言，对于条件表达式采用算符文法的分析方法来判断是否满足要求。

# 二、程序功能描述

## 2.1、通用功能

根据输入文法，计算FIRST/FOLLOW集，生成预测分析表，同时可判断该文法是否为SLR文法。

## 2.2、主要功能

识别特定标签组成的xml文档，解析成一个数据库，并且能通过sql对其进行简单的增、删、查的操作。

# 三、程序具体实现

## 3.1、主要数据结构

记录文法中的每个表达式（包含分析过程中的偏1移量，序号）：

class Experssion

{

public:

char expr[128];

int offset;

int no;

public:

Experssion ();

bool operator==(Experssion t);

void Show();

};

记录分析过程中的每个项目集：

class ItemSet

{

public:

int no;

int num;

Experssion \*expr;

public:

ItemSet ();

void AddExpr(Experssion new\_expr);

bool operator==(ItemSet t);

//testshow

void Show();

};

文法分析过程中用到的图结构（邻接链表）：

边节点：

class ArcNode

{

public:

char simbol;

ArcNode \* next;

int index;

public:

ArcNode ();

};

点节点：

class VNode

{

public:

ItemSet \*itemset;

ArcNode \* firstarc;

public:

VNode ();

void AddArcNode(char c,int in);

};

图结构：

class Graphis

{

public:

//边数，节点数

int vexnum;

VNode \* vertices;

public:

Graphis ();

VNode\* AddVNode ();

void Show();

};

词法分析结果

class Word{

public:

int size;

char \*word;

int type; //种类，t为标签名字，a为"=<>\#中的一种，c为内容,f为非终结符

public:

Word ();

void Insert(char c);

bool IS\_TAG(); //比较该单词是否是标签

};

栈节点

class StackNode{

public:

Word c;

int state;

StackNode \* last;

public:

StackNode();

};

分析用栈结构

class Stack{

public:

int size;

StackNode \* top;

public:

Stack();

Word Pop();

void Push(Word c,int s);

void Show(bool tag);

bool IsEmpty();

};

数据表：

class Table{

public:

char \*name;

char \*\*pro; //属性名字

int \*type; //属性种类，0:int,1:char,>1,char[]

char \*\*record; //具体记录

int record\_num; //记录数量

int pro\_num; //属性数量

Table\* next\_table; //该数据库中的下一张表

Database \*db;

public:

Table();

void Show(); //显示Table

void ShowWithCondition(int loc\_of\_from); //select语句

void Insert();

void Delete();

char \*WhereAnaylize(int loc\_of\_where); //分析where后的语句，返回逆波兰式

bool Match(char\* rcd,char\* polan,int loc\_of\_where); //根据记录计算波兰式

};

数据库

class Database{

public:

char \* name;

ifstream file;

Table \*t;

public:

Database();

bool Open(string filename);

Word Get\_next\_word();

bool Grammar\_Anaylize();

Table\* InsertTable();

void AddRecord(Table \* t,Word record);

void Save();

};

## 3.2、部分关键算法

### 3.2.1、FIRST/FOLLOW集

对于符号串α= X1X2… Xn，构造 FIRST (α)

（1） 置 FIRST(α) = FIRST (X1) - {ε};

（2） 若对所有的 Xj ,1<=j<= i -1, ε∈FIRST (Xj), 则把FIRST(Xi) -{ε}加到FIRST(α)中；

（3） 若对所有的 Xj ,1<= j <=n, ε∈FIRST (Xj), 则把

ε加到FIRST(α)中。

FOR 每条产生式P→X1X2…Xn DO

FOR i:=1 TO n-1 DO

BEGIN

IF Xi和Xi+1均为终结符 THEN 置Xi = Xi+1

IF i≤n-2且Xi和Xi+2都为终结符

但Xi+1为非终结符 THEN 置Xi = Xi+2；

IF Xi为终结符而Xi+1为非终结符 THEN

FOR FIRSTVT(Xi+1)中的每个a DO

置 Xi < a；

IF Xi为非终结符而Xi+1为终结符 THEN

FOR LASTVT(Xi)中的每个a DO

置 a > Xi+1

END

FOLLOW集同上

### 3.2.2、项目集规范族算法

PROCEDURE ITEMSETS(G′)；

BEGIN

C:={CLOSURE({S′→·S})}；

REPEAT

FOR C中每个项目集I和G′的每个符号X DO

IF GO(I，X)非空且不属于C THEN

把GO(I，X)放入C族中;

UNTIL C 不再增大

END

转换函数GO把项目集连接成一个DFA转换图。

### 3.2.3、预测分析表及冲突解决方法

分析表的ACTION和GOTO子表构造方法：

1. 若项目A→α·aβ属于Ik且GO(Ik, a)＝Ij，a为终结符，则置ACTION[k,a] 为“sj”。

2. 若项目A→α·属于Ik，那么，对任何终结符a(或结束符#)，置ACTION[k,a]为 “rj”(假定产生式A→α是文法G′的第j个产生式)。

3. 若项目S′→S·属于Ik，则置ACTION[k,#]为 “acc”。

4. 若GO(Ik,A)＝Ij，A为非终结符，则置GOTO[k,A]=j。

5. 分析表中凡不能用规则1至4填入信息的空白格均置上“报错标志”。

冲突解决方法：

假定LR(0)规范族的一个项目集I={A1→α·a1β1，A2→α·a2β2，…，Am→α·amβm，B1→α·，B2→α·，…，Bn→α·} 如果集合{a1，…，am}，FOLLOW(B1)，…，FOLLOW(Bn)两两不相交(包括不得有两个FOLLOW集合有#)，则：

1. 若a是某个ai，i=1,2,…,m，则移进；

2. 若a∈FOLLOW(Bi)，i=1,2,…,n，则用产生式Bi→α进行归约；

3. 此外，报错。

### 3.2.4、算符优先分析方法

优先关系：

设 G为算符文法且不含ε- 产生式， a, b∈ VT, 算符间的优先关系定义为：

a =b 当且仅当G含有产生式 P →… ab… 或

P →… aQb…

a < b 当且仅当G含有产生式 P →… aR … 且

R ⇒ b… 或 R ⇒ Qb…

a > b 当且仅当G含有产生式 P →… Rb… 且

R ⇒ … a 或R ⇒ … aQ

FIRST(P)，LASTVT(P)

设 P∈VN ，定义 ：

FIRSTVT (P) =

{ a│P⇒a … 或 P⇒Qa … , a ∈VT , Q ∈VN }

LASTVT (P) =

{ a│P⇒… a 或 P ⇒… aQ , a ∈VT , Q∈VN }

构造算法：

把所有初值为真的数组元素F[P, a]的符号对(P, a)全都放在STACK之中。

如果栈STACK不空，就将栈顶逐出，记此项为(Q, a)。对于每个形如

P→Q…

的产生式，若F[P, a]为假，则变其值为真，且将(P, a)推进STACK栈。

上述过程必须一直重复，直至栈STACK拆空为止。

构造优先关系表算法：

FOR 每条产生式P→X1X2…Xn DO

FOR i:=1 TO n-1 DO

BEGIN

IF Xi和Xi+1均为终结符 THEN 置Xi = Xi+1

IF i≤n-2且Xi和Xi+2都为终结符

但Xi+1为非终结符 THEN 置Xi = Xi+2；

IF Xi为终结符而Xi+1为非终结符 THEN

FOR FIRSTVT(Xi+1)中的每个a DO

置 Xi < a；

IF Xi为非终结符而Xi+1为终结符 THEN

FOR LASTVT(Xi)中的每个a DO

置 a > Xi+1

END

# 四、执行界面和运行结果

## 4.1、文法分析结果

拿课上的文法做试验，结果如下：

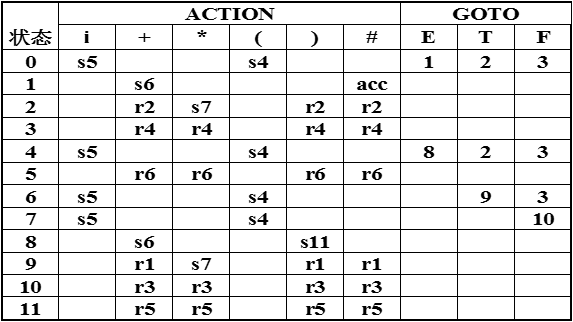
文法：G(E)

(1) E→E＋T (2) E→T

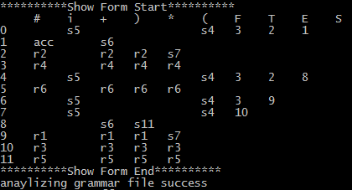
(3) T→T\*F (4) T→F

(5) F→(E) (6) F→i

正确结果：

****

分析结果：

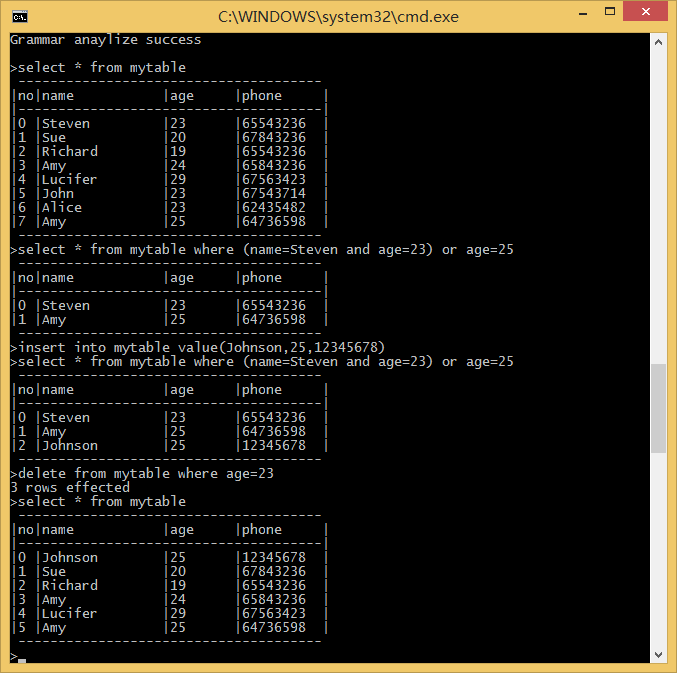
****

由于程序识别S为文法开始符号，所以添加表达式S->E，比较两图，虽然位置稍有差别，但结果一致。

## 4.2、词法、语法分析结果

这两个步骤没有直观的表现，略过。

## 4.3、命令，算符文法解析结果

****

# 五、设计体会

本次课设，先花了一段时间完成SLR文法的分析器，能够通过输入的文法计算FIRST/FOLLOW集，并且生成相应的预测分析表，为接下来的工作打下基础。

随后挑选主题，PL0与类C语言文法结构太大，输入文法分析器后不断显示有冲突，但是找不到冲突缘由。并且感觉对语法分析后的工作该如何做完全没有头绪，于是选择完成xml的解析。

xml文档的语法就比较简单，文法分析器可以识别并生成预测分析表。通过完成这个文法分析器，对项目集规范族、FIRST/FOLLOW集合这部分知识有了深刻认识。

分析完成后从shell中挑出命令解析的部分做为界面，这部分工作同时也是一个简单的词法分析。

然后对sql语句的简单解析，主要工作在于对where后的表达式的解析。没有写自动计算FIRSTVT/LASTVT的程序，手算后输入程序，用于分析。这部分工作完成后对算符文法的分析有了了解。

觉得xml这个主题完成的较好，但是没有做类c语言的稍感遗憾。