

Πανεπιστήμιο Πατρών
Τμήμα Μηχ. Η/Υ & Πληροφορικής

Αρχές Γλωσσών Προγραμματισμού &
Μεταφραστών
Εργαστηριακή Άσκηση
2017-2018

ΟΝΟΜΑ: ΣΑΒΒΑΣ ΓΕΩΡΓΙΟΥ
ΑΜ : 1040803/235037
email : sgeorgiou@ceid.upatras.gr

Σε αυτή την εργαστηριακή άσκηση μας ζητήθηκε να υλοποιήσουμε λεκτικό και συντακτικό αναλυτή για ένα υποσύνολο του προτύπου XML.

1) i) Αρχικά φτιαχνουμε το υποσύνολο σε **Backus-Naur form** :

<xml> ::= <Workbook>

<Workbook> ::= "<ss:Workbook>" [<Styles>]* <Worksheet>+ "</ss:Workbook>"

<Styles> ::= "<ss:Styles>" [<Style>]* "</ss:Styles>"

<Style> ::= "<ss:Style ss:ID=" string "></ss:Style>"

<Worksheet> ::= "<ss:Worksheet ss:Name=" string ["Protected=" boolean] ">" [<Table>]* "</ss:Worksheet>"

<Table> ::= "<ss:Table " ["ss:ExpandedColumnCount=" int] ["ss:ExpandedRowCount=" int] ["ss:StyleID=" string] ">" [<Column>]* [<Row>]* "</ss:Table>"

<Column> ::= "<ss:Column " ["ss:StyleID=" string] ["ss:Hidden=" boolean] ["ss:Width=" int] ">"

<Row> ::= "<ss:Row " ["ss:StyleID=" string] ["ss:Hidden=" boolean] ["ss:Height=" int] ">" [<Cell>]* "</ss:Row>"

<Cell> ::= "<ss:Cell " ["ss:StyleID=" string] ["ss:MergeAcross=" int] ["ss:MergeDown=" int] ">" [<Data>]* "</ss:Cell>"

<Data> ::= "<ss:Data ss:Type=" [num | datetime | boolean | string] ">" [string | empty] "</ss:Data>"

<Comments> ::= "<!--" [.*?] "->"

1)ii) Στην συνέχεια με τα εργαλεια Flex και Bison δημιουργησα τον αντιστοιχο αναλυτη.

Στο Flex σαν εισοδο για να δημιουργηθει ο λεκτικος αναλυτης εδωσα το παρακατω .l αρχαιο :

```
%{
#include "y.tab.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
void lexerror(const char*);
//int getcurrentline() { return yylineno; }
int linenum = 1;
int tb = 0;
int tbflag=0;
int i=0;
void tbs(int tb,int tbflag){
    if(tbflag==0){
        for(i=0;i<tb;i++){
            printf("\t");
        }
    }
}
void newl(int tbflag){
```

```
        if(tbflag==0){
            printf("\n");
        }
    }
    %}
```

```
%option yylineno
%option noyywrap
```

```
dig [0-9]
num \"{dig}+\"
```

```
string \"[A-Za-z0-9]+\"
string2 [A-Za-z0-9]+
```

```
boolean \"[\"true\"|\"false\"|\"TRUE\"|\"FALSE\"|\"True\"|\"False\"]\"
```

```
year {dig}{4}
month {dig}{2}
day {dig}{2}
datetime \"[year \"-\" month \"-\" day]\"
```

```
whitespace [ \n\t\r]+
newline \n
```

```
str "String"
mystring \"{str}\"
```

```
nm "Number"
mynumber \"{nm}\"
```

```
dt "DateTime"
mydatetime \"{dt}\"
```

```
bl "Boolean"
myboolean \"{bl}\"
```

%%

```
"ss"      { printf("ss"); return ss; }
"<"      { tbs(tb,tbflag); printf("<"); return st ;}
">"      { printf(">"); newl(tbflag) ; return en;}
"/"      { printf("/"); return sl;}
"="      { printf("="); return eq;}
":"      { printf(":"); return cl;}
```

```
"Workbook" { printf("Workbook"); return Wbook;}
"Styles"   { printf("Styles"); return Styles;}
"Style"    { printf("Style"); return Style;}
"ID"       { printf("ID"); return Id;}
"StyleID"  { printf("StyleID"); return Styleid;}
"Worksheet" { printf("Worksheet"); return Wsheet;}
"Name"     { printf("Name"); return Name;}
"Table"    { printf("Table"); return Table;}
"ExpandedColumnCount" {printf("ExpandedColumnCount"); return Excol;}
"ExpandedRowCount" {printf("ExpandedRowCount"); return Exrow;}
"Protected" {printf("Protected"); return Protected;}
"Column"    {printf("Column"); return Col;}
"Row"       {printf("Row"); return Row;}
"Cell"      {printf("Cell"); return Cell;}
"Heigth"    {printf("Height"); return Height;}
"Width"     {printf("Width"); return Width;}
"Hidden"    {printf("Hidden"); return Hidden;}
"Data"      {printf("Data"); return Data;}
"MergeAcross" {printf("MergeAcross"); return Meracross;}
"MergeDown" {printf("MergeDown"); return Merdown;}
"Type"      {printf("Type"); return Type;}
{mystring} {printf("%s",yytext); return datastring;}
{mynumber} {printf("%s",yytext); return datanumber;}
{mydatetime} {printf("%s",yytext); return datadatetime;}
{myboolean} {printf("%s",yytext); return databoolean;}
```

```
{num}          {printf("%s",yytext); return Num;}
{string}       {printf("%s",yytext); return String;}
{string2}      {printf("%s",yytext); return String2;}
```

```

{boolean}      {printf("%s",yytext); return Boolean;}
{datetime}     {printf("%s",yytext); return Datetime;}

[\\t\\n | \\n]  { ++linenum;}
[ \\t\\n]+      {printf(" ");}
"<!--".*\\n*"-->" {printf("%s",yytext);}
.              ;
%%

```

Χρησιμοποίησα τις μεταβλητές `tb` ,`tbflag`,και τις συναρτήσεις `tbs` ,`newl` για να έχω ένα μετρητή για να τυπώνω τον αναλογό αριθμό `tabs`.Το `tbflag` το χρησιμοποίησα έτσι ώστε στα `Style` και `Data` όπου έχουμε `starting` και `closing tag` στην ίδια γραμμή,να μην εκτυπώνετε `tab`.

Επίσης χρησιμοποιήθηκαν τα `%option yylineno` και `%option nouywrap`,όπου το πρώτο είναι για να στείλει στο `bison` τον αριθμό της γραμμής για το σφάλμα και το δεύτερο για να του δηλώσω ότι θα κάνει `parse` ένα αρχείο την φορά.

Στην συνέχεια στο εργαλείο `Bison` ,σαν εισοδο έδωσα το παρακάτω `.y` αρχείο :

```

%{
#include <stdio.h>
#include <stdlib.h>

```

```

void yyerror(const char *);
extern FILE *yyin;
extern int linenum;
extern int tb;
extern int tbflag;
extern int yylineno;

```

```

%}

```

```

%error-verbose

```

%token Wbook
%token Styles
%token Style
%token Id
%token Styleid
%token Wsheet
%token Name
%token Table
%token Excol
%token Exrow
%token Protected
%token Col
%token Row
%token Cell
%token Height
%token Width
%token Hidden
%token Data
%token Meracross
%token Merdown
%token Type
%token datastring
%token datanumber
%token databoolean
%token datadatetime

%token Num
%token String
%token String2
%token Boolean
%token Datetime

%token ss
%token st
%token en
%token sl
%token eq
%token cl

%start program

%%

```
program : workbook {printf("\n\nThe syntax of the XML file is correct \n");}
```

```
;
```

```
workbook : stss cl Wbook en worksheet2 stsl ss cl Wbook en
```

```
|
```

```
stss cl Wbook en styles2 worksheet2 stsl ss cl Wbook en
```

```
;
```

```
styles2 : styles2 styles | styles ;
```

```
styles : stss cl Styles en stsl ss cl Styles en
```

```
|
```

```
stss cl Styles en style2 stsl ss cl Styles en
```

```
;
```

```
worksheet2 : worksheet2 worksheet {--tb;}| worksheet;
```

```
worksheet : stss cl Wsheet ss cl Name eq String en stsl ss cl Wsheet en
```

```
|
```

```
stss cl Wsheet ss cl Name eq String en table2 stsl ss cl Wsheet en
```

```
|
```

```
stss cl Wsheet ss cl Name eq String ss cl Protected eq Boolean en stsl ss cl Wsheet en
```

```
|
```

```
stss cl Wsheet ss cl Name eq String ss cl Protected eq Boolean en table2 stsl ss cl Wsheet en
```

```
;
```

```
style2: style2 style | style ;
```

```
style : stssclStyle ss cl Id eq String en stsl ss cl Style en
```

```
;
```

```
table2 : table2 table {--tb;}| table;
```

```
table : stss cl Table en stsl ss cl Table en
```

```
|
```

```
stss cl Table en column2 stsl ss cl Table en
```

```
|
```

```
stss cl Table en row2 stsl ss cl Table en
```

```
|
```



```

stss cl Table en column2 row2 stsl ss cl Table en
|
stss cl Table tableop2 en stsl ss cl Table en
|
stss cl Table tableop2 en row2 stsl ss cl Table en
|
stss cl Table tableop2 en column2 stsl ss cl Table en
|
stss cl Table tableop2 en column2 row2 stsl ss cl Table en
;

```

```

tableop2 : tableop2 tableop | tableop ;

```

```

tableop : ss cl Excol eq Num
|
ss cl Exrow eq Num
|
ss cl Styleid eq String
;

```

```

column2 : column2 column {--tb;} | column ;

```

```

column: stss cl Col slen
|
stss cl Col colop2 slen
;

```

```

colop2 : colop2 colop | colop;

```

```

colop : ss cl Styleid eq String
|
ss cl Hidden eq Boolean
|
ss cl Width eq Num
;

```

```

row2 : row2 row {--tb;} | row;

```

```

row : stss cl Row en stsl ss cl Row en
|
stss cl Row en cell2 stsl ss cl Row en
|
stss cl Row rowop2 en stsl ss cl Row en
|
stss cl Row rowop2 en cell2 stsl ss cl Row en

```

```

;

rowop2 : rowop2 rowop | rowop;

rowop : ss cl Styleid eq String
|
ss cl Hidden eq Boolean
|
ss cl Height eq Num
;

cell2 : cell2 cell {--tb;}| cell;

cell : stss cl Cell en stsl ss cl Cell en
|
stss cl Cell en data2 stsl ss cl Cell en
|
stss cl Cell cellop2 en stsl ss cl Cell en
|
stss cl Cell cellop2 en data2 stsl ss cl Cell en
;

cellop2 : cellop2 cellop | cellop ;

cellop : ss cl Styleid eq String
|
ss cl Meracross eq Num
|
ss cl Merdown eq Num
;

data2 : data2 data {--tb;}| data ;

data : stssclData ss cl Type eq var en stsl ss cl Data en
|
stssclData ss cl Type eq var en String2 stsl ss cl Data en
;

var : datanumber | datadatetime | databoolean | datastring ;

stss : st ss {++tb;}
;

stsl : st sl {--tb; tbflag=0;}
;

```

```

slen : sl en {--tb;}
;

stssclStyle : stss cl Style {tbflag=1;}
;

stssclData : stss cl Data {tbflag=1;}
;

%%

void yyerror(const char *s) {

    fprintf(stderr, " %s at line number : %d \n ---> ",s,yylineno);

}

int main (int argc , char **argv){

    yyin = fopen(argv[1] , "r");
    if(!yyin){printf("cant open file"); return 1;}
    do{
        yyparse();
    }while(!feof(yyin));
    printf("\n");
    return 0;
}

```

Οι αυξομειώσεις τις μεταβλητής tb και η “ενεργοποίηση” του tbflag στα σημεία που έχω ενώσει τα <ss , </ , </> , <ss:Style και <ss:Data είναι για να ενημερωθούν καταλλήλα οι μεταβλητές του Flex και να εκτυπωθεί το αρχείο με την σωστή στοιχηση.

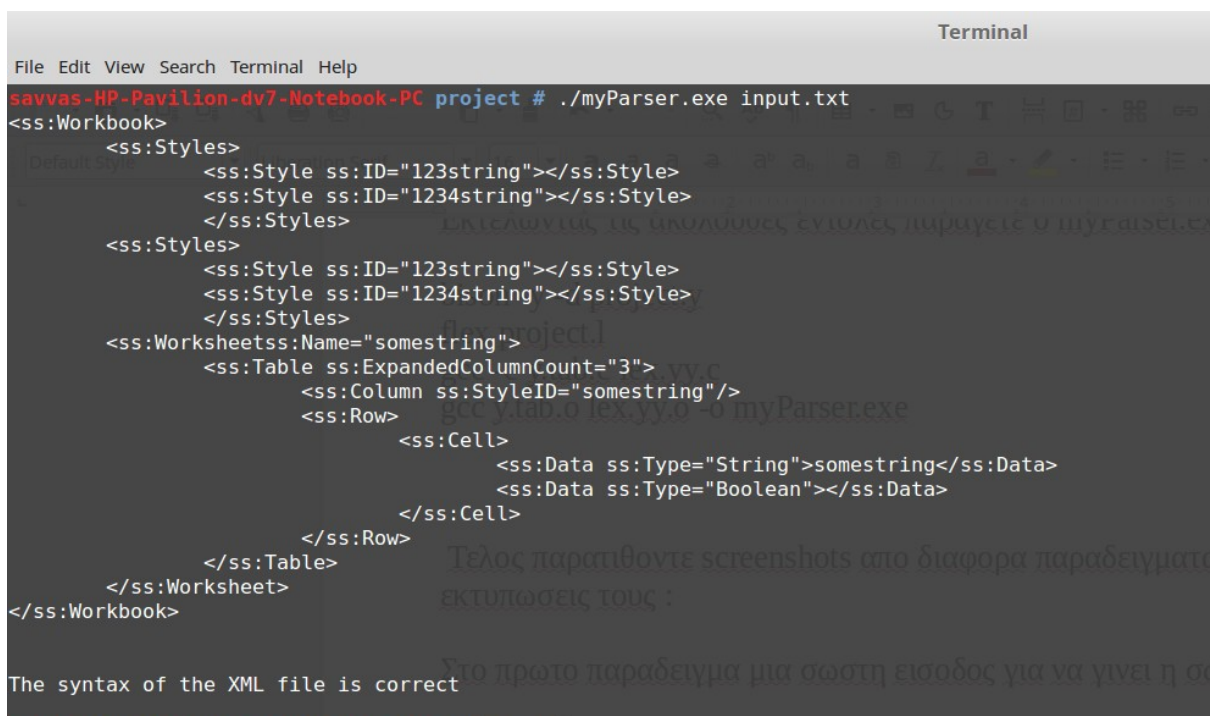
Για την εκτύπωση του συντακτικού σφάλματος προσθέσα την επιλογή %error-verbose έτσι ώστε το σφάλμα να είναι πιο κατατοπιστικό.Επίσης για να είναι πιο ευκολο να εντοπιστεί ένα σφάλμα προσθέσα στο printf του yyerror το yylineno όπου είναι μια μεταβλητή του Flex που κρατά την γραμμή του token που σταλθηκε έτσι όταν υπάρξει error θα είναι πιο ευκολο να το βρούμε.

Εκτελώντας τις ακόλουθες εντολές παραγάγετε ο myParser.exe

```
bison -y -d project.y
flex project.l
gcc -c y.tab.c lex.yy.c
gcc y.tab.o lex.yy.o -o myParser.exe
```

Τέλος παρατιθόντε screenshots απο διαφορα παραδειγματα εισοδου και οι εκτυπώσεις τους :

Στο πρωτο παραδειγμα μια σωστη εισοδος για να γίνει η σωστη στοιχηση:



```
Terminal
File Edit View Search Terminal Help
savvas-HP-Pavilion-dv7-Notebook-PC project # ./myParser.exe input.txt
<ss:Workbook>
  <ss:Styles>
    <ss:Style ss:ID="123string"></ss:Style>
    <ss:Style ss:ID="1234string"></ss:Style>
  </ss:Styles>
  <ss:Styles>
    <ss:Style ss:ID="123string"></ss:Style>
    <ss:Style ss:ID="1234string"></ss:Style>
  </ss:Styles>
  <ss:Worksheetss:Name="somestring">
    <ss:Table ss:ExpandedColumnCount="3">
      <ss:Column ss:StyleID="somestring"/>
      <ss:Row>
        <ss:Cell>
          <ss:Data ss:Type="String">somestring</ss:Data>
          <ss:Data ss:Type="Boolean"></ss:Data>
        </ss:Cell>
      </ss:Row>
    </ss:Table>
  </ss:Worksheet>
</ss:Workbook>

The syntax of the XML file is correct
```

Στο δευτερο παραδειγμα μπερδεψα την πρωτη εισοδο ενωνοντας αρκετες γραμμες,προσθετωντας σχολια για να φανει η σωστη λειτουργια και στοιχηση της εξοδου απο τον parser:

```
Terminal
File Edit View Search Terminal Help
savvas-HP-Pavilion-dv7-Notebook-PC project # ./myParser.exe input2.txt
<ss:Workbook>
  <ss:Styles>
    <ss:Style ss:ID="123string"></ss:Style>
    <ss:Style ss:ID="1234string"></ss:Style>
  </ss:Styles>
  <ss:Styles>
    <ss:Style ss:ID="123string"></ss:Style<!-- comments--> >
    <ss:Style ss:ID="1234string"></ss:Style>
  </ss:Styles>
  <ss:Worksheetss:Name="somestring">
    <ss:Table <!-- comments-->ss:ExpandedColumnCount="3">
      <ss:Column ss:StyleID="somestring"/>
      <ss:Row>
        <ss:Cell>
          <ss:Data ss:Type="String">somestring</ss:Data>
          <ss:Data ss:Type="Boolean"></ss:<!-- comments-->Data>
        </ss:Cell>
      </ss:Row>
    </ss:Table>
  </ss:Worksheet>
</ss:Workbook>

The syntax of the XML file is correct
savvas-HP-Pavilion-dv7-Notebook-PC project #
```

Στο 3ο παραδειγμα προσθεσα ενα συντακτικο λαθος (“>”)για να εκτυπωθει το αντιστοιχο μηνυμα σφαλματος:

```
Terminal
File Edit View Search Terminal Help
savvas-HP-Pavilion-dv7-Notebook-PC project # ./myParser.exe input3.txt
<ss:Workbook>
  <ss:Styles>
    <ss:Style ss:ID="123string"></ss:Style>
    <ss:Style ss:ID="1234string"></ss:Style>
  </ss:Styles>
  <ss:Styles>
    syntax error, unexpected en, expecting st at line number : 6
    --->
    <ss:Style ss:ID="123string">>
savvas-HP-Pavilion-dv7-Notebook-PC project #
```

Στο 4ο παραδειγμα εδωσα μια λαθος εισοδο στο Type του Data :

```
Terminal
File Edit View Search Terminal Help
savvas-HP-Pavilion-dv7-Notebook-PC project # ./myParser.exe input4.txt
<ss:Workbook>
  <ss:Styles>
    <ss:Style ss:ID="123string"></ss:Style>
    <ss:Style ss:ID="1234string"></ss:Style>
  </ss:Styles>
  <ss:Styles>
    <ss:Style ss:ID="123string"></ss:Style>
    <ss:Style ss:ID="1234string"></ss:Style>
  </ss:Styles>
  <ss:Worksheetss:Name="somestring">
    <ss:Table ss:ExpandedColumnCount="3">
      <ss:Column ss:StyleID="somestring"/>
      <ss:Row>
        <ss:Cell>
          <ss:Data ss:Type="String">somestring</ss:Data>
syntax error, unexpected String2, expecting datastring or datanumber or databoolean or datadatetime at line number : 15
--->
          <ss:Data ss:Type=incorrect
savvas-HP-Pavilion-dv7-Notebook-PC project #
```

Και τελος στην εισοδο εγραψα πρωτα το Row και μετα το Column :

```
Terminal
File Edit View Search Terminal Help
savvas-HP-Pavilion-dv7-Notebook-PC project # ./myParser.exe input5.txt
<ss:Workbook>
  <ss:Styles>
    <ss:Style ss:ID="123string"></ss:Style>
    <ss:Style ss:ID="1234string"></ss:Style>
  </ss:Styles>
  <ss:Styles>
    <ss:Style ss:ID="123string"></ss:Style>
    <ss:Style ss:ID="1234string"></ss:Style>
  </ss:Styles>
  <ss:Worksheetss:Name="somestring">
    <ss:Table ss:ExpandedColumnCount="3">
      <ss:Row>
        <ss:Cell>
          <ss:Data ss:Type="String">somestring</ss:Data>
          <ss:Data ss:Type="Boolean"></ss:Data>
        </ss:Cell>
      </ss:Row>
syntax error, unexpected Col, expecting Row at line number : 17
--->
      <ss:Column
savvas-HP-Pavilion-dv7-Notebook-PC project #
```