

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ**



**Αρχές Γλωσσών Προγραμματισμού
και Μεταφραστών**

Project 2018-2019

Δεσύλλας Ανδρέας AM:6035

email: desyllas@ceid.upatras.gr

Κουμαδωράκης Δημήτρης AM: 6093

email: koumadorakis@ceid.upatras.gr

Σουκαράς Σωτήρης AM: 6205

email: soukaras@ceid.upatras.gr

Χρήστου Χρήστος AM: 6255

email: cchristou@ceid.upatras.gr

Ερώτημα 1

i) Συντακτικός ορισμός γραμματικής της γλώσσας **BNF**

<STARTOBJECT>::= OP_OBJECT EleCreatedat ElemIDstr ElemText ElemUserObject CL_OBJECT
| OP_OBJECT EleCreatedat ElemIDstr EleRetweetstatus CL_OBJECT
| OP_OBJECT EleCreatedat ElemIDstr EleTweet CL_OBJECT
| OP_OBJECT EleCreatedat ElemIDstr EleTweet COMMA EleExtendedTweet CL_OBJECT;

<EleExtendedTweet>::= /*e*/ | SINGLEQUOTE TRUNC SINGLEQUOTE EQ SINGLEQUOTE
TRUE SINGLEQUOTE COMMA SINGLEQUOTE EXTENDED SINGLEQUOTE EQ OP_OBJECT
SINGLEQUOTE FULLTEXT SINGLEQUOTE EQ SINGLEQUOTE SINGLEQUOTE CL_OBJECT
COMMA | SINGLEQUOTE TRUNC SINGLEQUOTE EQ SINGLEQUOTE FALSE SINGLEQUOTE
COMMA;

<EleCreatedat>::= SINGLEQUOTE CREATEDAT SINGLEQUOTE EQ SINGLEQUOTE DAYNAME
MONTHNAME DAYNUM TIMESTAMP ZONE YEAR SINGLEQUOTE COMMA;

<ElemIDstr>::= SINGLEQUOTE ID_STR SINGLEQUOTE EQ SINGLEQUOTE SINGLEQUOTE
COMMA;

<ElemText>::= SINGLEQUOTE TEXT SINGLEQUOTE EQ SINGLEQUOTE SINGLEQUOTE
COMMA | SINGLEQUOTE TEXT SINGLEQUOTE EQ SINGLEQUOTE RTTWEET
SINGLEQUOTE COMMA;

<ElemUserObject>::= SINGLEQUOTE USEROBJECT SINGLEQUOTE EQ OP_OBJECT
<ElemUserId> <ElemUserName> <ElemScreenname> <ElemLocation> <ElemUrl> <ElemDesc>
CL_OBJECT | SINGLEQUOTE USEROBJECT SINGLEQUOTE EQ OP_OBJECT <ElemScreenname>
CL_OBJECT;

<ElemUserId>::= SINGLEQUOTE USERID SINGLEQUOTE EQ INTEGER COMMA ;

<ElemUserName>::= SINGLEQUOTE USERNAME SINGLEQUOTE EQ SINGLEQUOTE
SINGLEQUOTE COMMA ;

<ElemScreenname>::= SINGLEQUOTE SCREENNAME SINGLEQUOTE EQ SINGLEQUOTE
SINGLEQUOTE COMMA;

<ElemLocation>::= SINGLEQUOTE LOCATION SINGLEQUOTE EQ SINGLEQUOTE
SINGLEQUOTE COMMA;

<ElemUrl>::= SINGLEQUOTE URL SINGLEQUOTE EQ SINGLEQUOTE SINGLEQUOTE
COMMA;

<ElemDesc>::= SINGLEQUOTE DESC SINGLEQUOTE EQ SINGLEQUOTE SINGLEQUOTE ;

```
<EleRetweetstatus>::= SINGLEQUOTE RETWEETED SINGLEQUOTE EQ OP_OBJECT <ElemText>  
<ElemUserObject> CL_OBJECT;  
  
<EleTweet>::= SINGLEQUOTE TWEET SINGLEQUOTE EQ OP_OBJECT <ElemText>  
<ElemUserObject> CL_OBJECT;
```

Αρχεία Περιγραφής της Γλώσσας

1) Flex→json_al.l

```
%{  
  
    #include "json_ab.tab.h"  
  
    #include <stdio.h>  
  
    #include <stdlib.h>  
  
    #include <string.h>  
  
    char *temp;  
  
    char *stemp;  
  
    int lineNumber = 1;  
  
}%  
  
%option noyywrap  
  
%option yylineno  
  
month_name ("Jan"|"Feb"|"Mar"|"Apr"|"May"|"Jun"|"Jul"|"Aug"|"Sep"|"Oct"|"Nov"|"Dec")  
day_name (Sun|Mon|Tue|Wen|Thu|Fri|Sat)  
digit [0-9]  
esc[\\]  
other [V.-: _@#^$<>!,?&*%+=]  
name [a-z,A-Z,0-9\\s]+  
string {name}({name}|{digit}|{other}|{esc})+  
text ({name}|{digit}|{other}|{esc}){1,140}
```

day_num [01-31]

boolean "true"|"false"

timestamp ([2][0-3][0-1][0-9][1-9]):[0-5][0-9]:([0-5][0-9][6][0])

descriptiontype [^\r\n]+((\r\n\r\n)[^\r\n]+)*

digit1to9 [1-9]

integer {digit1to9}{digit}*{digit}+

singlequote ["]

nline [\n]

plus ([+]"%2B")

mark (![!]"%21")

comma ([,]"%2C")

amp ([&]"%26")

ws ([\s\t]"%20")

%%

{timestamp} {printf("%s", yytext); return (TIMESTAMP);}

"created_at" {printf("%s", yytext); return (CREATEDAT);}

"truncated" {printf("%s", yytext); return (TRUNC);}

"retweeted_status" {printf("%s", yytext); return (RETWEETED);}

"tweet" {printf("%s", yytext); return (TWEET);}

"text" {printf("%s", yytext); return (TEXT);}

"extended_tweet" {printf("%s", yytext); return (EXTENDED);}

"full_text" {printf("%s", yytext); return (FULLTEXT);}

"name" {printf("%s", yytext); return (USERNAME);}

"screen_name" {printf("%s", yytext); return (SCREENNAME);}

"location" {printf("%s", yytext); return (LOCATION);}

"url" {printf("%s", yytext); return (URL);}

```

"description" {printf("%s", yytext); return (DESC);}

{day_num}    {printf("%s", yytext); return (DAYNUM);}

"id"         {printf("%s", yytext); return (USERID);}

"true"       {printf("%s", yytext); return (TRUE);}

>false"      {printf("%s", yytext); return (FALSE);}

"user"       {printf("%s", yytext); return (USEROBJECT);}

"id_str"     {printf("%s", yytext); return (ID_STR);}

{month_name} {printf("%s", yytext); return (MONTHNAME);}

{digit}{4}   {printf("%s", yytext); return (YEAR);}

{day_name}   {printf("%s", yytext); return (DAYNAME);}

{integer}    {printf("%s", yytext); return (INTEGER);}

{singlequote} {printf("%s", yytext); return (SINGLEQUOTE);}

"+"0000      {printf("%s", yytext); return (ZONE);}

{comma}      {printf("%s", yytext); return (COMMA);}

":"          {printf("%s", yytext); return (EQ);}

"RT@"{text}  {printf("%s", yytext); return (RTTWEET);}

{nline}     {printf("\n"); lineNumber++;}

"{"          {printf("%s", yytext); return (OP_OBJECT);}

"}"          {printf("%s", yytext); return (CL_OBJECT);}

"["          {printf("%s", yytext); return (OPARRAY);}

"]"          {printf("%s", yytext); return (CLARRAY);}

```

2) Bison → json_ab.y

```

%{

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

```

```

int errorCount = 0;

extern int lineNumber;

int yylex();

void yyerror(char *);

extern FILE *yyin;

extern FILE *yyout;

%}

%start STARTOBJECT

%token COMMA EQ CREATEDAT DAYNAME MONTHNAME DAYNUM TIMESTAMP ZONE
YEAR

%token USERID INTEGER SINGLEQUOTE USEROBJECT ID_STR USERNAME UNIQUENAME
TEXT RETWEETED NAME BOOL

%token SCREENNAME LOCATION URL DESC DESCRIPTIONTYPE USERNAMETYPE
NUMBER TWEET RTTWEET TRUNC TRUE FALSE EXTENDED FULLTEXT

%left CL_OBJECT CLARRAY

%right OP_OBJECT OPARRAY

%%

STARTOBJECT: OP_OBJECT EleCreatedat ElemIDstr ElemText ElemUserObject CL_OBJECT

| OP_OBJECT EleCreatedat ElemIDstr EleRetweetstatus CL_OBJECT

| OP_OBJECT EleCreatedat ElemIDstr EleTweet CL_OBJECT

| OP_OBJECT EleCreatedat ElemIDstr EleTweet COMMA EleExtendedTweet
CL_OBJECT;

EleExtendedTweet: /*e*/

| SINGLEQUOTE TRUNC SINGLEQUOTE EQ SINGLEQUOTE TRUE
SINGLEQUOTE COMMA SINGLEQUOTE EXTENDED SINGLEQUOTE EQ OP_OBJECT
SINGLEQUOTE FULLTEXT SINGLEQUOTE EQ SINGLEQUOTE SINGLEQUOTE CL_OBJECT
COMMA

| SINGLEQUOTE TRUNC SINGLEQUOTE EQ SINGLEQUOTE FALSE
SINGLEQUOTE COMMA;

```

EleCreatedat : SINGLEQUOTE CREATEDAT SINGLEQUOTE EQ SINGLEQUOTE DAYNAME
MONTHNAME DAYNUM TIMESTAMP ZONE YEAR SINGLEQUOTE COMMA;

ElemIDstr: SINGLEQUOTE ID_STR SINGLEQUOTE EQ SINGLEQUOTE SINGLEQUOTE
COMMA;

ElemText: SINGLEQUOTE TEXT SINGLEQUOTE EQ SINGLEQUOTE SINGLEQUOTE COMMA
|SINGLEQUOTE TEXT SINGLEQUOTE EQ SINGLEQUOTE RTTWEET
SINGLEQUOTE COMMA;

ElemUserObject: SINGLEQUOTE USEROBJECT SINGLEQUOTE EQ OP_OBJECT ElemUserId
ElemUserName ElemScreenname ElemLocation ElemUrl ElemDesc CL_OBJECT
|SINGLEQUOTE USEROBJECT SINGLEQUOTE EQ OP_OBJECT ElemScreenname
CL_OBJECT;

ElemUserId : SINGLEQUOTE USERID SINGLEQUOTE EQ INTEGER COMMA ;

ElemUserName : SINGLEQUOTE USERNAME SINGLEQUOTE EQ SINGLEQUOTE
SINGLEQUOTE COMMA ;

ElemScreenname: SINGLEQUOTE SCREENNAME SINGLEQUOTE EQ SINGLEQUOTE
SINGLEQUOTE COMMA;

ElemLocation: SINGLEQUOTE LOCATION SINGLEQUOTE EQ SINGLEQUOTE SINGLEQUOTE
COMMA;

ElemUrl: SINGLEQUOTE URL SINGLEQUOTE EQ SINGLEQUOTE SINGLEQUOTE COMMA;

ElemDesc: SINGLEQUOTE DESC SINGLEQUOTE EQ SINGLEQUOTE SINGLEQUOTE ;

EleRetweetstatus : SINGLEQUOTE RETWEETED SINGLEQUOTE EQ OP_OBJECT ElemText
ElemUserObject CL_OBJECT;

EleTweet: SINGLEQUOTE TWEET SINGLEQUOTE EQ OP_OBJECT ElemText ElemUserObject
CL_OBJECT;

%%

void yyerror (char *s)

{

errorCount++;

fprintf(stderr,"%s on line %d \n",s,lineNumber); /*here I get the info about errors */

}

```

int main(int argc, const char **argv)
{
    if (argc > 1) {
        yyin = fopen(argv[1], "r");
    } else {
        yyin = stdin;
    }

    int result;

    if ((result = yyparse()) == 0) {
        printf("\n=====PARSING COMPLETED=====\\n");
        printf("=====YOUR PROGRAM IS VALID=====\\n");
    } else {
        printf("\\nFound %d syntax errors\\n",errorCount);
    }

    return 0;
}

```

Αρχεία Εισόδου για τα Flex και Bison

Φτιάξαμε κάποια χαρακτηριστικά αρχεία εισόδου που να ανταποκρίνονται στα ζητούμενα της εκφώνησης. Ενώ μερικά έγιναν parsing σωστά, μερικά matcharουν και δεν δίνουν το επιθυμητό αποτέλεσμα.

1) sample.txt

```
{
  "created_at": "Thu Jun 06 23:12:21 +0000 2017",
  "id_str": "",
  "text": "",
  "user": {
    "id": 51110213123123,
    "name": "Twitter Dev",
    "screen_name": "TwitterDev",
    "location": "Internet",
    "url": "https",
    "description": ""
  }
}
```

2) tweet.txt

```
{
  "created_at": "Thu Jun 06 23:12:21 +0000 2017",
  "id_str": "",
  "tweet": {
    "text": "RT@ Twitter Dev",
    "user": {
      "screen_name": "TwitterDev",
    }
  }
}
```

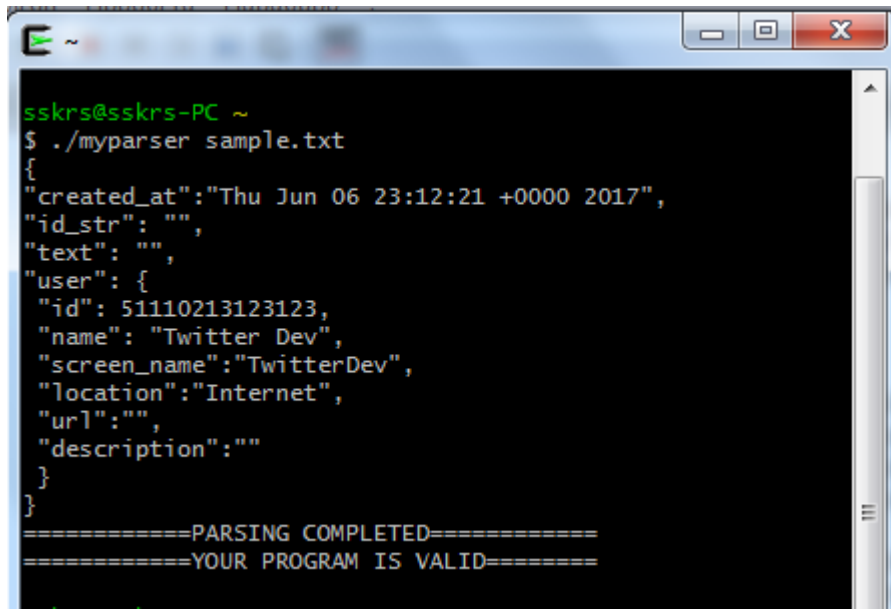
3) retweet.txt

```
{
  "created_at": "Thu Jun 06 23:12:21 +0000 2017",
  "id_str": "",
  "retweeted_status": {
    "text": "",
    "user": {
      "screen_name": "TwitterDev",
    }
  }
}
```

4) extended.txt

```
{
  "created_at": "Thu Jun 06 23:12:21 +0000 2017",
  "id_str": "",
  "tweet": {
    "text": "RT@ Twitter Dev",
    "user": {
      "screen_name": "TwitterDev",
    }
  },
  "truncated": "true",
  "extended_tweet": {
    "full_text": ""
  },
}
```

Screenshots



A screenshot of a terminal window titled "sskrs@sskrs-PC ~". The terminal shows the command `./myparser sample.txt` being executed. The output is a JSON object representing a tweet, followed by two lines of status messages: `=====PARSING COMPLETED=====` and `=====YOUR PROGRAM IS VALID=====`.

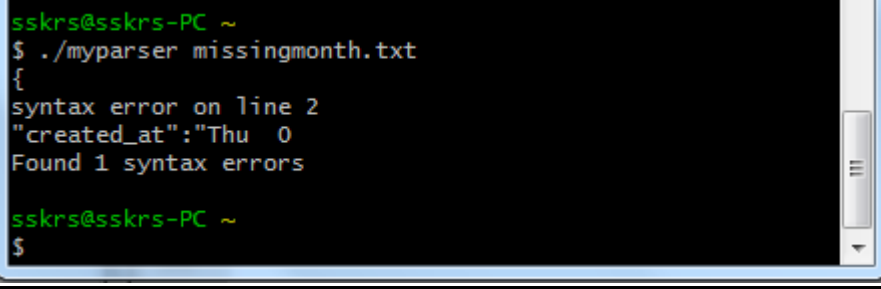
```
sskrs@sskrs-PC ~
$ ./myparser sample.txt
{
  "created_at": "Thu Jun 06 23:12:21 +0000 2017",
  "id_str": "",
  "text": "",
  "user": {
    "id": 51110213123123,
    "name": "Twitter Dev",
    "screen_name": "TwitterDev",
    "location": "Internet",
    "url": "",
    "description": ""
  }
}
=====PARSING COMPLETED=====
=====YOUR PROGRAM IS VALID=====
```

```
sskrs@sskrs-PC ~  
$ ./myparser tweet.txt  
{  
  "created_at": "Thu Jun 06 23:12:21 +0000 2017",  
  "id_str": "",  
  "tweet": {  
    "text": "RT@ Twitter Dev",  
    "user": {  
      "screen_name": "TwitterDev",  
    }  
  }  
}  
=====PARSING COMPLETED=====  
=====YOUR PROGRAM IS VALID=====
```

```
sskrs@sskrs-PC ~  
$ ./myparser retweet.txt  
{  
  "created_at": "Thu Jun 06 23:12:21 +0000 2017",  
  "id_str": "",  
  "retweeted_status": {  
    "text": "",  
    "user": {  
      "screen_name": "TwitterDev",  
    }  
  }  
}  
=====PARSING COMPLETED=====  
=====YOUR PROGRAM IS VALID=====
```

```
sskrs@sskrs-PC ~  
$ ./myparser extended.txt  
{  
  "created_at": "Thu Jun 06 23:12:21 +0000 2017",  
  "id_str": "",  
  "tweet": {  
    "text": "RT@ Twitter Dev",  
    "user": {  
      "screen_name": "TwitterDev",  
    }  
  },  
  "truncated": "true",  
  "extended_tweet": {  
    "full_text": ""  
  }  
}  
=====PARSING COMPLETED=====  
=====YOUR PROGRAM IS VALID=====
```

Φτιάξαμε ένα επιπλέον αρχείο που να δείχνει τι θα εμφανίσει το bison σε περίπτωση που το txt δεν έχει κάποιο token που έχουμε δηλώσει απαραίτητα να υπάρχει στο bison. Εδώ συγκεκριμένα λείπει το token MONTHNAME.

A terminal window with a black background and green text. The prompt is 'sskrs@sskrs-PC ~'. The user enters './myparser missingmonth.txt'. The output shows a syntax error on line 2, specifically with the string 'created_at': 'Thu 0', and states 'Found 1 syntax errors'. The prompt returns to 'sskrs@sskrs-PC ~' and '\$'.

```
sskrs@sskrs-PC ~  
$ ./myparser missingmonth.txt  
{  
syntax error on line 2  
"created_at": "Thu 0  
Found 1 syntax errors  
  
sskrs@sskrs-PC ~  
$
```