



## Περιεχόμενα Εργασίας

---

1. Εισαγωγή	2
2. BNF συντακτικού ορισμού γραμματικής	2
3. Κώδικας Flex	5
4. Κώδικας Bison	6
5. Βοηθητικές Συναρτήσεις	11
6. Αρχεία εισόδου	14
7. Το script make	17
8. Παραδείγματα εκτέλεσης	17

## 1. Εισαγωγή

---

Σε αυτήν την εργαστηριακή άσκηση κληθήκαμε να υλοποιήσουμε ένα λεξικό και έναν συντακτικό αναλυτή για το πρότυπο JSON. Το JSON αποτελεί ένα ανοικτό πρότυπο το οποίο χρησιμοποιεί κείμενο αναγνώσιμο από τον άνθρωπο με σκοπό τη μετάδοση πληροφοριακών αντικειμένων δεδομένων. Αυτό το πρότυπο χρησιμοποιείται και στο γνωστό *tweeter*. Για να υλοποιήσουμε το λεξικό αυτής της γλώσσας χρησιμοποιήσαμε το πρόγραμμα FLEX, ενώ για τον συντακτικό αναλυτή το πρόγραμμα Bison. Παρακάτω παρουσιάζουμε το BNF συντακτικού ορισμού γραμματικής της γλώσσας, τον κώδικα του FLEX και του BISON, καθώς σχόλια και screenshots από την εκτέλεσή του.

## 2. BNF συντακτικού ορισμού γραμματικής

---

Παρακάτω παραθέτουμε την περιγραφή της γραμματικής της γλώσσας που συντάξαμε σε BNF και για τα δύο ερωτήματα, δηλαδή για τα tweets, τα retweets και τα extended tweets:

```
//route GRAMMAR
jsonfile:      OBRACKETS ruleRESTART CBRACKETS;

//GRAMMAR
ruleRESTART:   ruleSTART
               |ruleRETWEET
               ;

ruleSTART:     ruleSTART COMMA ruleSTART
               |ruleCreate
               |ruleID_STR
               |ruleText
               |ruleDTR
               |ruleTRUNCA
               |ruleET
               |ruleUser
               |rulePlace
               |ruleEntities
               ;

//GRAMMAR for CREATED_AT
ruleCreate:    CREATED_AT COLON CREATED_AT_ATTRIBUTE;

//GRAMMAR for ID_STR
ruleID_STR:    ID_STR COLON ID_STR_ATTRIBUTE;

//GRAMMAR for TEXT
ruleText:      TEXT COLON STRING;

//GRAMMAR for USER
ruleUser:      USER COLON OBRACKETS rule3 CBRACKETS;
```

```

rule3:                rule3 COMMA rule3
                      |ID COLON INT
                      |NAME COLON STRING
                      |SCREEN_NAME COLON STRING
                      |LOCATION COLON STRING
                      |URL COLON STRING
                      |DESCRIPTION COLON STRING
                      |ID_STR COLON ID_STR_ATTRIBUTE
                      ;

//GRAMMAR for PLACE
rulePlace:            PLACE COLON OBRACKETS CBRACKETS;

//GRAMMAR for TRUNCATED
ruleTRUNCA:          TRUNCATED COLON BOOLEAN;

//GRAMMAR for DISPLAY_TEXT_RANGE
ruleDTR:             DISP_TEXT_RANGE COLON OBRACES INT COMMA INT CBRACES;

//GRAMMAR for EXTENDED_TWEET
ruleET:              EXTENDED_TWEET COLON OBRACKETS ruleET1 CBRACKETS;

ruleET1:             ruleET1 COMMA ruleET1
                      |FULL_TEXT COLON STRING
                      |DISP_TEXT_RANGE COLON OBRACES INT COMMA INT CBRACES
                      |ENTITIES COLON OBRACKETS ruleHA CBRACKETS
                      ;

//GRAMMAR for HASHTAGS (#)
ruleHA:              HASHTAGS COLON OBRACES ruleHA1 CBRACES;

ruleHA1:             ruleHA1 COMMA ruleHA1
                      |OBRACKETS TEXT COLON STRING COMMA INDICES COLON
                      OBRACES INT COMMA INT CBRACES CBRACKETS
                      ;

//GRAMMAR for ENTITIES
ruleEntities:        ENTITIES COLON OBRACKETS rule4 CBRACKETS;

rule4:              rule4 COMMA rule4
                    |HASHTAGS COLON OBRACES CBRACES
                    |URLS COLON OBRACES OBRACKETS rule5 CBRACKETS CBRACES
                    |USER_MENTIONS COLON OBRACES CBRACES
                    ;

```

rule5: URL COLON STRING COMMA UNWOUND COLON OBRACKETS rule6  
CBRACKETS;

rule6: URL COLON STRING COMMA TITLE COLON STRING;

//GRAMMAR for RETWEET

ruleRETWEET: ruleRETWEET COMMA ruleRETWEET  
|TWEET COLON OBRACKETS ruleTW CBRACKETS  
;

//GRAMMAR for TWEET

ruleTW: ruleTW COMMA ruleTW  
|TEXT COLON STRING  
|USER COLON OBRACKETS SCREEN\_NAME COLON STRING  
CBRACKETS  
|RETWEETED\_STATUS COLON OBRACKETS ruleRS  
CBRACKETS  
|ENTITIES COLON OBRACKETS CBRACKETS  
|EXTENDED\_ENTITIES COLON OBRACKETS CBRACKETS  
;

//GRAMMAR for RETWEETED\_STATUS

ruleRS: ruleRS COMMA ruleRS  
|TEXT COLON STRING  
|USER COLON OBRACKETS SCREEN\_NAME COLON STRING  
CBRACKETS  
|PLACE COLON OBRACKETS CBRACKETS  
|ENTITIES COLON OBRACKETS CBRACKETS  
|EXTENDED\_ENTITIES COLON OBRACKETS CBRACKETS  
;

### 3. Κώδικας Flex

```
1  %{
2  #include "myscanner.tab.h"
3  int line_num=1;
4  %}
5
6  |
7  day          [0][1-9]|[1-2][0-9]|[3][0-1]
8  days         ("Mon"|"Tue"|"Wed"|"Thu"|"Fri"|"Sat"|"Sun")
9  months       ("Jan"|"Feb"|"Mar"|"Apr"|"May"|"Jun"|"Jul"|"Aug"|"Sep"|"Oct"|"Nov"|"Dec")
10 timesta      [0-1][0-9]":"[0-5][0-9]":"[0-5][0-9]|[2][0-3]":"[0-5][0-9]":"[0-5][0-9]
11 year         [0-9]{4}
12 time         {days}{ws}{months}{ws}{day}{ws}{timesta}{ws}{year}
13 int          [1-9][0-9]*|0
14 boolean      ("true"|"false")
15 string       (\\.|[^\"])*
16 ws           [ \t\n\r]
17 %%
18
19 :             {return COLON;}
20 ,             {return COMMA;}
21 \{           {return OBRACKETS;}
22 \}           {return CBRACKETS;}
23 \[           {return OBRACES;}
24 \]           {return CBRACES;}
25 \"created_at\" {return CREATED_AT;}
26 \"{time}\"    {return CREATED_AT_ATTRIBUTE;}
27 \"{int}\"     {return ID_STR;}
28 \"{int}\"     {return ID_STR_ATTRIBUTE;}
29 \"text\"      {return TEXT;}
30 \"user\"      {return USER;}
31 \"id\"        {return ID;}
32 \"name\"      {return NAME;}
33 \"screen_name\" {return SCREEN_NAME;}
34 \"location\"  {return LOCATION;}
35 \"url\"       {return URL;}
36 \"description\" {return DESCRIPTION;}
37 \"place\"     {return PLACE;}
38 \"entities\"  {return ENTITIES;}
39 \"hashtags\"  {return HASHTAGS;}
40 \"urls\"      {return URLS;}
41 \"unwound\"   {return UNWOUND;}
42 \"title\"     {return TITLE;}
43 \"user_mentions\" {return USER_MENTIONS;}
44 \"retweeted_status\" {return RETWEETED_STATUS;}
45 \"extended_tweet\" {return EXTENDED_TWEET;}
46 \"truncated\"  {return TRUNCATED;}
47 \"display_text_range\" {return DISP_TEXT_RANGE;}
48 \"full_text\"  {return FULL_TEXT;}
49 \"indices\"    {return INDICES;}
50 \"tweet\"      {return TWEET;}
51 \"extended_entities\" {return EXTENDED_ENTITIES;}
52 {boolean}     {return BOOLEAN;}
53 \"{string}\"   {return STRING;}
54 {int}         {return INT;}
55 \n            {++line_num;}
56
57 {ws}          {;}
58
59 %%
60
61 int yywrap(void)
62 {
63     return 1;
64 }
```

## 4. Κώδικας Bison

---

```
1  %{
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <string.h>
5
6  extern FILE *yyin;
7  extern int yylex();
8  extern char* yytext;
9  extern int yyparse();
10 extern int line_num;
11 int errorline;
12 int range=0;
13 int strlength;
14 int errors=0;
15 int trun=0;
16 int id_table[1024];
17 int uniIDs=0;
18 int counter;
19 void checklength(char* s);
20 void checkifUnique (char* l);
21 void StartsWith(char* a);
22 void yyerror(const char* s);
23 %}
24
25 %error-verbose
26 %union {
27     int num;
28     char* string;}
29 %token COLON COMMA OBRACKETS OBRACES CBRACKETS CBRACES
30 %token CREATED_AT ID_STR
31 %token TEXT USER
32 %token ID
33 %token NAME URL
34 %token SCREEN_NAME LOCATION
35 %token DESCRIPTION PLACE
36 %token ENTITIES HASHTAGS
37 %token URLS UNWOUND
38 %token TITLE USER_MENTIONS
39 %token RETWEETED_STATUS EXTENDED_TWEET TRUNCATED DISP_TEXT_RANGE FULL_TEXT INDICES
40 TWEET
41 %token CREATED_AT_ATTRIBUTE
42 %token EXTENDED_ENTITIES
43 %token BOOLEAN
44 %token ID_STR_ATTRIBUTE
45 %token STRING
46 %token INT
47 %type <string> STRING
48 %type <num> INT
49 %start jsonfile
50 %%
51 //route GRAMMAR
52 jsonfile:      OBRACKETS {printf("{\n");} ruleRESTART CBRACKETS {printf("}\n");};
53
54 //GRAMMAR for 2.0
55 ruleRESTART:   ruleSTART
56               | ruleRETWEET
57               ;
```



```

60
61 ruleSTART:      ruleSTART COMMA {printf(",\n");} ruleSTART
62                | ruleCreate
63                | ruleID_STR
64                | ruleText
65                | ruleDTR
66                | ruleTRUNCA
67                | ruleET
68                | ruleUser
69                | rulePlace
70                | ruleEntities
71                ;
72
73
74
75 //GRAMMAR for CREATE
76 ruleCreate:      CREATED_AT {printf("\t\"created_at\"");} COLON {printf(":");}
77                 CREATED_AT_ATTRIBUTE {printf("\%s",yytext);};
78
79 //GRAMMAR for ID_STR
80 ruleID_STR:      ID_STR {printf("\t\"id_str\"");} COLON {printf(":");}
81                 ID_STR_ATTRIBUTE {checkifUnique(yytext); printf("\%s",yytext);};
82
83 //GRAMMAR for TEXT
84 ruleText:        TEXT {printf("\t\"text\"");} COLON {printf(":");} STRING {checklength
85                 (yytext); printf("\%s",yytext);};
86
87 //GRAMMAR for USER
88 ruleUser:        USER {printf("\t\"user\"");} COLON {printf(":");} OBRACKETS
89                 {printf("{\n");} rule3 CBRACKETS {printf("\n\t");};
90
91 rule3:           rule3 COMMA {printf(",\n");} rule3
92                 | ID {printf("\t\t\"id\"");} COLON {printf(":");} INT {checkifUnique(
93                 yytext); printf("\%s",yytext);}
94                 | NAME {printf("\t\t\"name\"");} COLON {printf(":");} STRING
95                 {printf("\%s",yytext);}
96                 | SCREEN_NAME {printf("\t\t\"screen_name\"");} COLON {printf(":");}
97                 STRING {printf("\%s",yytext);}
98                 | LOCATION {printf("\t\t\"location\"");} COLON {printf(":");} STRING
99                 {printf("\%s",yytext);}
100                | URL {printf("\t\t\"url\"");} COLON {printf(":");} STRING
101                {printf("\%s",yytext);}
102                | DESCRIPTION {printf("\t\t\"description\"");} COLON {printf(":");}
103                STRING {printf("\%s",yytext);}
104                | ID_STR {printf("\t\t\"id_str\"");} COLON {printf(":");}
105                ID_STR_ATTRIBUTE {printf("\%s",yytext);};
106                ;
107
108 rulePlace:        PLACE {printf("\t\"place\"");} COLON {printf(":");} OBRACKETS
109                 {printf("{\n");} CBRACKETS {printf("\t");};
110
111

```



```

100 //GRAMMAR for TRUNCATED
101 ruleTRUNCA:      TRUNCATED {printf("\t\t\"truncated\\");} COLON {printf(":");} BOOLEAN
{printf("%s",yytext);}
102         {
103             trun=1;
104             if(strcmp(yytext,"false")==0 && range!=0)
105                 {
106                     printf("\n Error in line %i ,the length of the text is
over 140 characters and truncated is false.\n",
line_num);
exit(EXIT_FAILURE);
}
107     };
108
109 //GRAMMAR for DTR
110
111 ruleDTR:          DISP_TEXT_RANGE {printf("\t\t\"display_text_range\\");} COLON
{printf(":");} OBRACES {printf("[");} INT {printf("%s",yytext);} COMMA
{printf(",");} INT {if(atoi(yytext)>139){range++;}printf("%s",yytext);} CBRACES
{printf("]");};
112
113 //GRAMMAR for ruleET
114
115 ruleET:           EXTENDED_TWEET {printf("\t\t\"extended_tweet\\");} COLON
{printf(": \n\t");} OBRACKETS {printf("{\n");} ruleET1 CBRACKETS {printf("\n\t");};
116
117 ruleET1:          ruleET1 COMMA {printf(", \n");} ruleET1
|FULL_TEXT {printf("\t\t\t\"full_text\\");} COLON {printf(":");}
STRING {printf("%s",yytext);}
118 |DISP_TEXT_RANGE {printf("\t\t\t\"display_text_range\\");} COLON
{printf(":");} OBRACES {printf("[");} INT {printf("%s",yytext);}
COMMA {printf(",");} INT {printf("%s",yytext);} CBRACES
{printf("]");};
119 |ENTITIES {printf("\t\t\t\"entities\\");} COLON {printf(":");}
OBRACKETS {printf("\n\t\t\t\t\n");} ruleHA CBRACKETS
{printf("\t\t\t\t");}
120 ;
121
122 ruleHA:            HASHTAGS {printf("\t\t\t\t\"hashtags\\");} COLON {printf(": \n");}
OBRACES {printf("\t\t\t\t[");} ruleHA1 CBRACES {printf("\n\t\t\t\t\t\n");};
123
124 ruleHA1:           ruleHA1 COMMA {printf(", \n");} ruleHA1
|OBRACKETS {printf("\n\t\t\t\t\t\n");} TEXT
{printf("\t\t\t\t\t\t\t\"text\\");} COLON {printf(":");} STRING
{printf("%s",yytext);} COMMA {printf(", \n");} INDICES
{printf("\t\t\t\t\t\t\t\"indices\\");} COLON {printf(":");} OBRACES
{printf("[");} INT {printf("%s",yytext);} COMMA {printf(",");} INT
{printf("%s",yytext);} CBRACES {printf("]");} CBRACKETS
{printf("\n\t\t\t\t\t\t\t");}
125 ;
126
127 //GRAMMAR for ENTITIES
128
129 ruleEntities:     ENTITIES {printf("\t\t\"entities\\");} COLON {printf(": \n\t");}
OBRACKETS {printf("{\n");} rule4 CBRACKETS {printf("\n\t");}

```

```

134 rule4:      rule4 COMMA {printf(",\n");} rule4
135      |HASHTAGS {printf("\t\t\t\"hashtags\\\"");} COLON {printf(":");} OBRACES
      {printf("[");} CBRACES {printf("]");}
136      |URLS {printf("\t\t\t\"urls\\\"");} COLON {printf(":");} OBRACES
      {printf("[\n");} OBRACKETS {printf("\t\t\t{\n");} rule5 CBRACKETS
      {printf("\t\t\t}\n");} CBRACES {printf("\t\t]");}
137      |USER_MENTIONS {printf("\t\t\t\"user_mentions\\\"");} COLON
      {printf(":");} OBRACES {printf("[");} CBRACES {printf("]");}
138      ;
139
140 rule5:      URL {printf("\t\t\t\t\"url\\\"");} COLON {printf(":");} STRING
      {printf("\%s",yytext);} COMMA {printf(",\n");} UNWOUND
      {printf("\t\t\t\t\"unwound\\\"");} COLON {printf(":");} OBRACKETS {printf("{\n");} rule6
      CBRACKETS {printf("\n\t\t\t\t}\n");}
141
142
143 rule6:      URL {printf("\t\t\t\t\t\"urls\\\"");} COLON {printf(":");} STRING
      {printf("\%s",yytext);} COMMA {printf(",\n");} TITLE {printf("\t\t\t\t\t\"title\\\"");}
      COLON {printf(":");} STRING {printf("\%s",yytext);}
144
145 ruleRETWEET: ruleRETWEET COMMA {printf(",\n");} ruleRETWEET
146      |TWEET {printf("\t\t\t\t\t\"tweet\\\"");} COLON {printf(": \n\t");} OBRACKETS
      {printf("{\n");} ruleTW CBRACKETS {printf("\t\t}\n");}
147
148      ;
149
150 ruleTW:      ruleTW COMMA {printf(",\n");} ruleTW
151      |TEXT {printf("\t\t\t\t\t\"text\\\"");} COLON {printf(":");} STRING
      {StartsWith(yytext); printf("\%s",yytext);}
152      |USER {printf("\t\t\t\t\t\"user\\\"");} COLON {printf(": \n\t\t\t");} OBRACKETS
      {printf("{\n");} SCREEN_NAME {printf("\t\t\t\t\t\"screen_name\\\"");}
      COLON {printf(":");} STRING {printf("\%s",yytext);} CBRACKETS
      {printf("\n\t\t\t");}
153      |RETWEETED_STATUS {printf("\t\t\t\t\t\"retweeted_status\\\"");} COLON
      {printf(": \n\t\t\t");} OBRACKETS {printf("{\n");} ruleRS CBRACKETS
      {printf("\t\t\t");}
154      |ENTITIES {printf("\t\t\t\t\t\"entities\\\"");} COLON {printf(": \n\t\t\t");}
      OBRACKETS {printf("{\n");} CBRACKETS {printf("\t\t\t");}
155      |EXTENDED_ENTITIES {printf("\t\t\t\t\t\"extended_entities\\\"");} COLON
      {printf(": \n\t\t\t");} OBRACKETS {printf("{\n");} CBRACKETS
      {printf("\t\t\t}\n");}
156      ;
157
158 ruleRS:      ruleRS COMMA {printf(",\n");} ruleRS
159      |TEXT {printf("\t\t\t\t\t\t\"text\\\"");} COLON {printf(":");} STRING
      {printf("\%s",yytext);}
160      |USER {printf("\t\t\t\t\t\t\"user\\\"");} COLON {printf(": \n\t\t\t\t");}
      OBRACKETS {printf("{\n");} SCREEN_NAME
      {printf("\t\t\t\t\t\t\t\"screen_name\\\"");} COLON {printf(":");} STRING
      {printf("\%s",yytext);} CBRACKETS {printf("\n\t\t\t\t\t");}
161      |PLACE {printf("\t\t\t\t\t\t\"place\\\"");} COLON {printf(": \n\t\t\t\t\t");}
      OBRACKETS {printf("{\n");} CBRACKETS {printf("\t\t\t\t\t");}
162      |ENTITIES {printf("\t\t\t\t\t\t\"entities\\\"");} COLON
      {printf(": \n\t\t\t\t\t");} OBRACKETS {printf("{\n");} CBRACKETS
      {printf("\t\t\t\t\t");}
163      |EXTENDED_ENTITIES {printf("\t\t\t\t\t\t\"extended_entities\\\"");} COLON
      {printf(": \n\t\t\t\t\t");} OBRACKETS {printf("{\n");} CBRACKETS
      {printf("\t\t\t\t\t}\n");}
164      ;

```



```

165
166
167 %%
168
169 void checklength (char* s)
170 {
171     if(strlen(s)>140)
172     {
173         errors++;
174         errorline=line_num;
175     }
176 }
177
178 void checkifUnique (char* l)
179 {
180     for(counter = 0; counter < uniIDs; counter++)
181     {
182         if(atoi(l) == id_table[counter])
183         {
184             errors++;
185             printf("\nError in line %i,the id is not unique\n", line_num);
186             exit(EXIT_FAILURE);
187         }
188     }
189     id_table[uniIDs] = atoi(l); //prosoxh 8esh Pinaka 0
190     uniIDs++;
191 }
192
193
194 void StartsWith(char* a)
195 {
196     char* b= "\"RT @";
197     if(strncmp(a, b, 5) != 0)
198     {
199         printf("\nError in line %i,the text does not have the right form (RT @).\n",
200             line_num);
201         exit(EXIT_FAILURE);
202     }
203 }
204
205 int main (int argc, char **argv) {
206     FILE *jfile = fopen(argv[1], "r");
207     yyin = jfile;
208     yyparse();
209     if(errors!=0 && trun ==0)
210     {
211         printf("\nError in line %i ,the length of the text is over 140
212             characters\n", errorline);
213         exit(EXIT_FAILURE);}
214     else
215     {
216         return 0;
217     }
218
219 void yyerror (const char *s) {fprintf (stderr, "%s\n", s);}
220

```

## 5. Βοηθητικές Συναρτήσεις

Χρησιμοποιούμε την συνάρτηση **void checklength (char\* s)**, η οποία παίρνει ως όρισμα το attribute του text (μέσω pointer χαρακτήρων) και ελέγχει αν είναι μεγαλύτερο των 140 χαρακτήρων. Αν είναι, αυξάνει την μεταβλητή errors και αποθηκεύει στην μεταβλητή errorline την γραμμή στην οποία παρατηρήθηκε αυτή η υπέρβαση χαρακτήρων. Ο λόγος που δεν κάνουμε exit(EXIT\_FAILURE) σε εκείνο το σημείο είναι για να γίνεται πρώτα έλεγχος αν υπάρχει extended tweet μέσω της truncated.

```
165 void checklength (char* s)
166 {
167     if(strlen(s)>140)
168     {
169         errors++;
170         errorline=line_num;
171     }
172 }
```

Σε περίπτωση **extended tweet**, στο πεδίο display\_text\_range ελέγχουμε αν το range είναι μεγαλύτερο από 140. Αν είναι μεγαλύτερο αυξάνει την μεταβλητή range και περιμένει να διαβάσει το attribute του truncated.

```
108 //GRAMMAR for DTR
109 ruleDTR:    DISP_TEXT_RANGE {printf("\t\"display_text_range\"");} COLON
{printf(":"");} OBRACES {printf("[");} INT {printf("%s",yytext);} COMMA {printf(",");}
INT {if(atoi(yytext)>139){range++;}printf("%s",yytext);} CBRACES {printf("]");};
```

Όταν διαβάσει το truncated δίνει στην μεταβλητή trun (που έχει αρχικοποιηθεί με 0) την τιμή 1. Αν η τιμή του attribute του truncated είναι false και η τιμή της μεταβλητής range δεν είναι 0 (δηλαδή το display range είναι μεγαλύτερο από 140) το πρόγραμμα τερματίζει με μήνυμα σφάλματος.

```
97 //GRAMMAR for TRUNCATED
98 ruleTRUNCA:    TRUNCATED {printf("\t\"truncated\"");} COLON {printf(":"");} BOOLEAN
{printf("%s",yytext);}
99 {
100     trun=1;
101     if(strcmp(yytext,"false")==0 && range!=0)
102     {
103         printf("\n Error in line %i ,the length of the text is
over 140 characters and truncated is false.\n", line_num);
104         exit(EXIT_FAILURE);
105     }
106 };
```

Για να εξασφαλίσουμε την μοναδικότητα των id (id, id\_str) χρησιμοποιούμε την συνάρτηση checkifUnique( char\* l). Δημιουργήσαμε τον πίνακα id\_table, στον οποίο μέσω της συνάρτησης αποθηκεύονται τα id. Πριν την αποθήκευση η συνάρτηση διατρέχει τον πίνακα και ελέγχει εάν το id υπάρχει ήδη στον πίνακα. Εάν υπάρχει, τερματίζει με μήνυμα σφάλματος, διαφορετικά προχωράει στην αποθήκευση. Επίσης, υποθέτουμε ότι τα id και id\_str δεν είναι ίδια για αυτό τα αποθηκεύουμε στον ίδιο πίνακα.

```
174 void checkifUnique (char* l)
175 {
176     for(counter = 0; counter < uniIDs; counter++)
177     {
178         if(atoi(l) == id_table[counter])
179         {
180             errors++;
181             printf("Error in line %i,the id is not unique\n", line_num);
182             exit(EXIT_FAILURE);
183         }
184     }
185     id_table[uniIDs] = atoi(l); //prosoxh 8esh Pinaka 0
186     uniIDs++;
187 }
188 }
```

Δημιουργήσαμε την συνάρτηση StartsWith (char\* a), ώστε στα retweet να ελέγχουμε αν η μορφή του text είναι έγκυρη, δηλαδή αν αρχίζει με «RT @». Αυτό το πετύχαμε με την χρήση της συνάρτησης strncmp(), η οποία συγκρίνει τους 'n' πρώτους χαρακτήρες του text (χρησιμοποιήσαμε n=5 – θεωρούμε ως πρώτο χαρακτήρα το ") με το RT @ και αν δεν είναι ίδιοι τερματίζει με το αντίστοιχο μήνυμα σφάλματος.

```
190 void StartsWith(char* a)
191 {
192     char* b= "\"RT @\"";
193     if(strncmp(a, b, 5) != 0)
194     {
195         printf("Error in line %i,the text does not have the right form (RT @).\n",
196             line_num);
197         exit(EXIT_FAILURE);
198     }
199 }
```

Στην `main()` χειριζόμαστε την περίπτωση που το `text` υπερβαίνει τους 140 χαρακτήρες και δεν υπάρχει `extended tweet`. Αν η μεταβλητή `errors` δεν είναι 0, δηλαδή το `text` είναι πάνω από 140 χαρακτήρες και επιπλέον η μεταβλητή `trun` είναι 0, δηλαδή δεν υπάρχει το `truncated` ( και συνεπώς `extended tweet` ), τότε το πρόγραμμα τερματίζει με το κατάλληλο μήνυμα σφάλματος.

```
201 int main (void) {
202     yyparse();
203     if(errors!=0 && trun ==0)
204     {
205         printf("Error in line %i ,the length of the text is over 140 characters\n",
                errorline);
206         exit(EXIT_FAILURE);}
207     else
208     {
209         return 0;
210     }
211 }
```

Στο αρχείο `flex` χρησιμοποιούμε την μεταβλητή `line_num`, η οποία αυξάνεται κατά ένα όταν βρίσκει το σύμβολο `'\n'`, δηλαδή σε αλλαγή γραμμής. Στο `bison` αρχείο δηλώνεται ως `extern`, δηλαδή ως εξωτερική μεταβλητή και χρησιμοποιείται για να τυπώνεται σε περίπτωση οποιουδήποτε σφάλματος η γραμμή που αυτό παρουσιάστηκε.

Η συνάρτηση `void yyerror (const char* s)` χρησιμοποιείται από το `bison` για να τυπώνονται τυχόν `errors` στο `parsing`.

Το `%error-verbose` είναι οδηγία του `bison` που εμφανίζει πιο συγκεκριμένες πληροφορίες σε περίπτωση εμφάνισης σφάλματος.

Η μεταβλητή `yytext` περιέχει το κομμάτι του αρχείου που γίνεται `parse` την δεδομένη στιγμή. Την χρησιμοποιούμε για μερικά `print` αλλά και στις βοηθητικές συναρτήσεις για τους ελέγχους που κάνουμε.

Στις παρακάτω εκτελέσεις των αρχείων εμφανίζονται κάποια `warnings`, τα οποία, όμως, δεν εμποδίζουν την ομαλή λειτουργία του συντακτικού αναλυτή.

Τα `shift/reduce`, `reduce/reduce` `warnings` εμφανίζονται διότι σε μερικές περιπτώσεις περισσότεροι από έναν κανόνες μπορεί να ικανοποιούνται, όμως στην υλοποίησή μας ο `BISON` επιλέγει τον «κατάλληλο».

## 6. Αρχεία Εισόδου

---

- Το αρχείο εισόδου που χρησιμοποιούμε για το πρώτο ερώτημα είναι το παρακάτω (config.txt):

```
{
  "created_at": "Thu Apr 06 15:24:15 2017",
  "id_str": "850006245121695744",
  "text": "1\ Today we\u2019re shard our vision for the future
of the Twitter API platform!\nhttps://t.co/XweGngmxlP",
  "user": {
    "id": 2244994945,
    "name": "Twitter Dev",
    "screen_name": "TwitterDev",
    "location": "Internet",
    "url": "https://dev.twitter.com/",
    "description": "Your official source for Twitter Platform
news, updates & events. Need technical help? Visit
https://twittercommunity.com/\u2328\ufe0f #TapIntoTwitter"
  },
  "place": {
  },
  "entities": {
    "hashtags": [
    ],
    "urls": [
      {
        "url": "https://t.co/XweGngmxlP",
        "unwound": {
          "url":
            "https://cards.twitter.com/cards/18ce53wgo4h/3xo1c",
          "title": "Building the Future of the Twitter API
Platform"
        }
      }
    ],
    "user_mentions": [
    ]
  }
}
```



- Το αρχείο εισόδου που χρησιμοποιούμε για τα retweets είναι το παρακάτω (config2.txt):

```
{
  "tweet":
  {
    "text": "RT @author original message",
    "user":
    {
      "screen_name": "Retweeter"
    },
    "retweeted_status":
    {
      "text": "original message",
      "user":
      {
        "screen_name": "OriginalTweeter"
      },
      "place":
      {
      },
      "entities":
      {
      },
      "extended_entities":
      {
      }
    },
    "entities": {
    },
    "extended_entities": {
    }
  }
}
```

- Το αρχείο εισόδου που χρησιμοποιούμε για τα extended tweets είναι το παρακάτω (config3.txt):

```

"created_at": "Thu May 10 17:41:57 2018",
"id_str": "994633657141813248",
"text": "Just another Extended Tweet with more than 140 characters, generated as a
documentation example, showing that [\"tru... https://t.co/U7Se4NM7Eu" ,
"display_text_range": [0, 140],
"truncated": true,
"user":
{
    "id_str": "944480690",
    "screen_name": "FloodSocial"
},
"extended_tweet":
{
    "full_text": "Just another Extended Tweet with more than 140 characters,
generated as a documentation example, showing that [\"truncated\": true] and the
presence of an \"extended_tweet\" object with complete text and \"entities\"
#documentation #parsingJSON #GeoTagged https://t.co/e9yhQTJ5IA" ,
    "display_text_range": [0, 249],
    "entities": {
        "hashtags":
        [
            {
                "text": "documentation",
                "indices": [211, 225]
            },
            {
                "text": "parsingJSON",
                "indices": [226, 238]
            },
            {
                "text": "GeoTagged",
                "indices": [239, 249]
            }
        ]
    },
    "entities":
    {
        "hashtags": []
    }
}

```

## 7. To script make

Το script make χρησιμοποιήθηκε για την αυτοματοποίηση της διαδικασίας δημιουργίας και εκτέλεσης του λεκτικού και συντακτικού μας αναλυτή (Το επισυνάπτουμε μαζί με την εργασία μας).

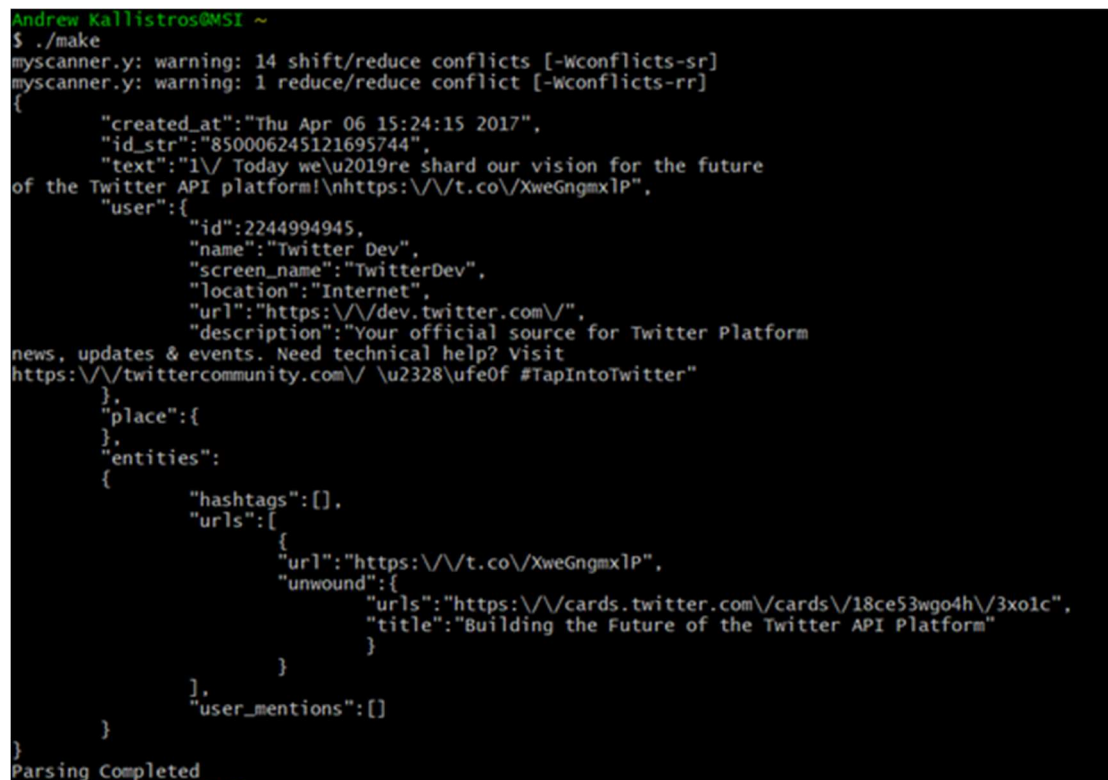
Παράδειγμα χρήσης της make και για τα 3 παραπάνω αρχεία εισόδου:

```
bison -d myscanner.y
flex myscanner.l
gcc myscanner.tab.c lex.yy.c -lfl
./a.exe <config.txt
./a.exe <config2.txt
./a.exe <config3.txt
```

## 8. Παραδείγματα εκτέλεσης

Παρακάτω παραθέτουμε παραδείγματα από την εκτέλεση του λεκτικού και συντακτικού αναλυτή για τις διάφορες περιπτώσεις που εξετάσαμε:

- Εκτέλεση του αρχείου config.txt



```
Andrew Kallistros@MSI ~
$ ./make
myscanner.y: warning: 14 shift/reduce conflicts [-wconflicts-sr]
myscanner.y: warning: 1 reduce/reduce conflict [-wconflicts-rr]
{
  "created_at": "Thu Apr 06 15:24:15 2017",
  "id_str": "850006245121695744",
  "text": "1\ Today we\u2019re shard our vision for the future
of the Twitter API platform!\nhhttps://t.co/XweGngmx1P",
  "user": {
    "id": 2244994945,
    "name": "Twitter Dev",
    "screen_name": "TwitterDev",
    "location": "Internet",
    "url": "https://dev.twitter.com/",
    "description": "Your official source for Twitter Platform
news, updates & events. Need technical help? Visit
https://twittercommunity.com/ \u2328\u201c\u201c #TapIntoTwitter"
  },
  "place": {
  },
  "entities": {
    "hashtags": [],
    "urls": [
      {
        "url": "https://t.co/XweGngmx1P",
        "unwound": {
          "urls": "https://cards.twitter.com/cards/18ce53wgo4h/3xo1c",
          "title": "Building the Future of the Twitter API Platform"
        }
      }
    ],
    "user_mentions": []
  }
}
Parsing Completed
```

- Εκτέλεση του αρχείου config2.txt

```
Andrew Kallistros@MSI ~
$ ./make
myscanner.y: warning: 14 shift/reduce conflicts [-Wconflicts-sr]
myscanner.y: warning: 1 reduce/reduce conflict [-Wconflicts-rr]
{
    "tweet":
    {
        "text": "RT @author original message"
        "user":
        {
            "screen_name": "Retweeter"
        },
        "retweeted_status":
        {
            "text": "original message"
            "user":
            {
                "screen_name": "OriginalTweeter"
            },
            "place":
            {
            },
            "entities":
            {
            },
            "extended_entities":
            {
            }
        },
        "entities": {
        },
        "extended_entities": {
        }
    }
}
Parsing Completed
```

- Εκτέλεση του αρχείου config3.txt

```
Andrew Kallistros@MSI ~
$ ./make
myscanner.y: warning: 14 shift/reduce conflicts [-Wconflicts-sr]
myscanner.y: warning: 1 reduce/reduce conflict [-Wconflicts-rr]
{
    "created_at": "Thu May 10 17:41:57 2018",
    "id_str": "994633657141813248",
    "text": "Just another Extended Tweet with more than 140 characters, generated as a documentation example, showing that [\\\"tru. https://t.co/e9yhQTJ5IA\\\" with complete text and \\\"entities\\\" #documentation #parsingJSON #GeoTagged https://t.co/e9yhQTJ5IA",
    "display_text_range": [0,140],
    "truncated": true,
    "user": {
        "id_str": "944480690",
        "screen_name": "FloodSocial"
    },
    "extended_tweet":
    {
        "full_text": "Just another Extended Tweet with more than 140 characters, generated as a documentation example, showing that [\\\"tru. https://t.co/e9yhQTJ5IA\\\" with complete text and \\\"entities\\\" #documentation #parsingJSON #GeoTagged https://t.co/e9yhQTJ5IA",
        "display_text_range": [0,249],
        "entities":
        {
            "hashtags":
            [
                {
                    "text": "documentation",
                    "indices": [211,225]
                },
                {
                    "text": "parsingJSON",
                    "indices": [226,238]
                },
                {
                    "text": "GeoTagged",
                    "indices": [239,249]
                }
            ]
        },
        "hashtags": []
    }
}
Parsing Completed
```

- Εκτέλεση του αρχείου config.txt με κείμενο πάνω από 140 χαρακτήρες

```
Andrew Kallistros@MSI ~
$ ./make
myscanner.y: warning: 14 shift/reduce conflicts [-Wconflicts-sr]
myscanner.y: warning: 1 reduce/reduce conflict [-Wconflicts-rr]
{
  "created_at": "Thu Apr 06 15:24:15 2017",
  "id_str": "850006245121695744",
  "text": "OVER 140 CHARACTERS 123njfnrvjnevfwdefefefe1/ Today we\u2019re shard our vision for the future
of the Twitter API platform!\nhttps://t.co/XweGngmx1P",
  "user": {
    "id": 2244994945,
    "name": "Twitter Dev",
    "screen_name": "TwitterDev",
    "location": "Internet",
    "url": "https://dev.twitter.com/",
    "description": "Your official source for Twitter Platform
news, updates & events. Need technical help? Visit
https://twittercommunity.com/ \u2328\u201c\u201c #TapIntoTwitter"
  },
  "place": {
  },
  "entities": {
    "hashtags": [],
    "urls": [
      {
        "url": "https://t.co/XweGngmx1P",
        "unwound": {
          "urls": "https://cards.twitter.com/cards/18ce53wgo4h/3xo1c",
          "title": "Building the Future of the Twitter API Platform"
        }
      }
    ],
    "user_mentions": []
  }
}
Error in line 4 ,the length of the text is over 140 characters
```

- Εκτέλεση του αρχείου config.txt με λανθασμένη ημερομηνία

```
Andrew Kallistros@MSI ~
$ ./make
myscanner.y: warning: 14 shift/reduce conflicts [-Wconflicts-sr]
myscanner.y: warning: 1 reduce/reduce conflict [-Wconflicts-rr]
{
  "created_at":
syntax error, unexpected STRING, expecting CREATED_AT_ATTRIBUTE
```

- Εκτέλεση παραλλαγής του αρχείου config.txt έχοντας 2 users με το ίδιο ID

```
Andrew Kallistros@MSI ~
$ ./make
myscanner.y: warning: 14 shift/reduce conflicts [-Wconflicts-sr]
myscanner.y: warning: 1 reduce/reduce conflict [-Wconflicts-rr]
{
  "created_at": "Thu Apr 06 15:24:15 2017",
  "id_str": "850006245121695744",
  "text": "1/ Today we\u2019re shard our vision for the future
of the Twitter API platform!\nhttps://t.co/XweGngmx1P",
  "user": {
    "id": 2244994945,
    "name": "Twitter Dev",
    "screen_name": "TwitterDev",
    "location": "Internet",
    "url": "https://dev.twitter.com/",
    "description": "Your official source for Twitter Platform
news, updates & events. Need technical help? Visit
https://twittercommunity.com/ \u2328\u201c\u201c #TapIntoTwitter"
  },
  "user": {
    "id":
Error in line 14,the id is not unique
```

- Εκτέλεση παραλλαγής του αρχείου config.txt έχοντας 2 tweets με ίδιο id\_str

```
Andrew Kallistros@MSI ~
$ ./make
myscanner.y: warning: 14 shift/reduce conflicts [-Wconflicts-sr]
myscanner.y: warning: 1 reduce/reduce conflict [-Wconflicts-rr]
{
    "created_at": "Thu Apr 06 15:24:15 2017",
    "id_str": "850006245121695744",
    "id_str":
Error in line 4, the id is not unique
```

- Εκτέλεση παραλλαγής του αρχείου config.txt (διαφορετική διάταξη στοιχείων)

```
Andrew Kallistros@MSI ~
$ ./make
myscanner.y: warning: 14 shift/reduce conflicts [-Wconflicts-sr]
myscanner.y: warning: 1 reduce/reduce conflict [-Wconflicts-rr]
{
    "id_str": "850006245121695744",
    "user": {
        "id": 2244994945,
        "name": "Twitter Dev",
        "screen_name": "TwitterDev",
        "location": "Internet",
        "url": "https://dev.twitter.com/",
        "description": "Your official source for Twitter Platform
news, updates & events. Need technical help? Visit
https://twittercommunity.com/ \u2328\u201c\u201c #TapIntoTwitter"
    },
    "created_at": "Thu Apr 06 15:24:15 2017",
    "text": "1\ Today we\u2019re shard our vision for the future
of the Twitter API platform!\nhttps://t.co/XweGngmx1P",
    "place": {
    },
    "entities": {
        "hashtags": [],
        "urls": [
            {
                "url": "https://t.co/XweGngmx1P",
                "unwound": {
                    "urls": "https://cards.twitter.com/cards/18ce53wgo4h/3xo1c",
                    "title": "Building the Future of the Twitter API Platform"
                }
            }
        ],
        "user_mentions": []
    }
}
```

- Εκτέλεση παραλλαγής του αρχείου config2.txt (retweet) με λανθασμένη είσοδο στο "text" (που δεν αρχίζει με "RT @ ")

```
Andrew Kallistros@MSI ~
$ ./make
myscanner.y: warning: 14 shift/reduce conflicts [-Wconflicts-sr]
myscanner.y: warning: 1 reduce/reduce conflict [-Wconflicts-rr]
{
    "tweet": {
        "text":
Error in line 4, the text does not have the right form (RT @).
```

- Εκτέλεση παραλλαγής του αρχείου config3.txt (extended tweet) με display text range μεγαλύτερο του 140 και truncated = false

```
Andrew Kallistros@MSI ~  
$ ./make  
myscanner.y: warning: 14 shift/reduce conflicts [-wconflicts-sr]  
myscanner.y: warning: 1 reduce/reduce conflict [-wconflicts-rr]  
{  
  "created_at": "Thu May 10 17:41:57 2018",  
  "id_str": "994633657141813248",  
  "text": "Just another Extended Tweet with more than 140 characters, generated as a documentation example, showing that [\"tru... https://t.co/U7Se4NW7Eu",  
  "display_text_range": [0,140],  
  "truncated": false  
Error in line 6 ,the length of the text is over 140 characters and truncated is false.
```

- Περίπτωση μη εύρεσης του αρχείου

```
Andrew Kallistros@MSI ~  
$ ./make  
myscanner.y: warning: 14 shift/reduce conflicts [-wconflicts-sr]  
myscanner.y: warning: 1 reduce/reduce conflict [-wconflicts-rr]  
./make: line 4: confi.txt: No such file or directory
```