Project 2025

Εισαγωγή

Γραμματική BNF

Η γραμματική θα ξεκινάει από το xml και στην συνέχεια μέσω του **root** μπορεί να χρησιμοποιηθεί είτε το πεδίο **linearLayout** είτε το **relativelayout** το οποίο ξεχωριστά έχει τα δικά του **attributes** αλλά και κάποιο content.

Τα **attributes** μπορεί να είναι υποχρεωτικά ή να παραλείπονται (με τελεστή **OR** "|".)

Το **content** είναι τα διάφορα **elements** όπως **Textview**, **Imageview**, **Button** κα. τα οποία έχουν τα δικά τους **attributes**. Στο τέλος μπορεί να οριστεί και ο τρόπος που θα εμφανίζονται τα αλφαριθμητικά αλλά και οι αριθμοί.

Bison

Ο **Bison** είναι ο Συντακτικός Αναλυτής. Για τον **Bison** χρειάζεται να ορίσουμε ορισμένα tokens για χρήση στη συνέχεια στην γραμματική. Αυτά τα **tokens** θα συνδεθούν με τον Λεκτικό Αναλυτή **Flex** για να μπορέσει ο **parser** να αναγνωρίσει τα **tokens** από το αρχείο εισόδου.

Επίσης, στο **Bison** θα χρειαστεί να υλοποιηθεί η main η οποία θα κάνει **parse** το αρχείο μέσω της συνάρτησης **yyparse();.**

Κάθε token αναπαριστά το κάθε στοιχείο που χρειαζεται να αναγνωρίζεται από το αρχείο σε **xml**.

π.χ. **%token OPEN_RADIO_GROUP CLOSE_RADIO_GROUP**
%token OPEN_RADIO_BUTTON

π.χ. Τα **CLOSE** και **CLOSE2 συμβολίζουν τα “>” και “/>” στον κώδικα αντίστοιχα.**

%token CLOSE CLOSE2

Bison

Μετά τον προσδιορισμό των token ακολουθεί η γραμματική. Είναι πολύ σημαντικό η γραμματική να έχει περιέχει με σωστή σειρά τα **tokens** που θέλουμε να αναγνωρίζονται. Π.χ. για το στοιχείο **ImageView** θα έχει τα εξής χαρακτηριστικά:

```
image_view_attributes:
    ANDROID_LAYOUT_WIDTH ALPHANUMERIC
    ANDROID_LAYOUT_HEIGHT ALPHANUMERIC
    ANDROID_SRC ALPHANUMERIC
    android_id_padding_optional
    ;
```

- Θα ξεκινάει με τα **token ANDROID_LAYOUT_WIDTH** και **ALPHANUMERIC** που συσχετίζονται με το χαρακτηριστικό **android:layout_width** και ένα αλφαριθμητικό αντίστοιχα.
- Στην συνέχεια παρομοίως τα token **ANDROID_LAYOUT_HEIGHT**, **ALPHANUMERIC ANDROID_SRC** και **ALPHANUMERIC**
- Τέλος, θα μπορεί να χρησιμοποιεί έπειτα από τα παραπάνω **token** τον κανόνα **android_id_padding_optional**.

Bison

Επιπλέον, στο τελευταίο μέρος του Bison υλοποιούνται σε κώδικα C οι βασικές συναρτήσεις του Bison συμπεριλαμβανομένου και της **main**.

```
int has_error = 0;


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


   yyin = fopen(argv[1], "r");

   yyparse();


   if (has_error == 0) {

      printf("\n\nSuccessful
parsing\n");

   }

   return 0;

}
```

Στην main μέσω της **yyin** θα ορίσουμε το προκαθορισμένο αρχείο εισόδου το οποίο θα πρέπει να βρίσκεται στην θέση 1 της εκτέλεσης στην γραμμή εντολών. Π.χ. ***./run.exe input.txt*** όπου το ***input.txt*** αντιστοιχεί στο **argv[1]**.
Η **yyparse** κάθε φορά καλεί την **yylex** προς αναγνώριση κάποιας λεκτικής μονάδας. Στην ουσία αναγνωρίζει τις συμβολοσειρές που έχουμε ορίσει σαν εισόδο.
Επιπλέον, έχουμε ορίσει να εκτυπώνεται και ένα μήνυμα επιτυχής διαπέρασης του αρχείου. Για να το καταφέρουμε αυτό θα χρειαστεί να ορίσουμε μια μεταβλητή τύπου **int** την **has_error = 0** όπου θα γίνεται 1 μόνο αν καλεστεί η **yyeror** και έτσι το παραπάνω μήνυμα θα εκτυπώνεται όταν δεν υπάρχει κάποιο σφάλμα στην διαπέραση.

Bison

yyeror:
Καλείται αυτόματα όταν εντοπιστεί κάποιο συντακτικό σφάλμα.

```
void yyerror(char *s) {
fprintf(stderr, "Syntax error on line %d which is shown below:\n", yylineno);
has_error = 1;
        }
```

Flex

Flex είναι ο Λεκτικός Αναλυτής. Για τον **Flex** χρειάζεται να ορίσουμε κάποιους κανόνες ώστε να αναγνωρίζουν χαρακτήρες από το αρχείο και να επιστρέφουν το **token** στο οποίο αντιστοιχούν. Αυτά τα **tokens** θα συνδεθούν με τον **Bison** χρησιμοποιώντας την εντολή **#include** "**bison1.tab.h**".

Όταν αναγνωρίζεται το κάθε χαρακτηριστικό μαζί με το " = " θα επιστρέφεται το αντίστοιχο **token**.

```
"android:layout_width=" {flag=6;printf("%s",yytext);return ANDROID_LAYOUT_WIDTH;}

"android:layout_height=" {flag=6;printf("%s",yytext);return ANDROID_LAYOUT_HEIGHT;}
```