XML μεταγλωτιστής

Αρχές Γλωσσών Προγραμματισμού & Μεταφραστών

ΛΑΖΑΡΟΣ ΤΟΠΑΛΗΣ

Α.Μ.: 1088101

email: up1088101@upnet.gr

Μάιος 2023

Περιεχόμενα

1 Εισαγωγή

η εργασία υλοποιήθηκε κατά το εαρινό εξάμηνου του τρίτου έτους φοίτησης του προπτυχιακού πτυχίου του Τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής. Σκοπός της εργασίας είναι η κατασκευή ενός μέρους από έναν compiler μίας γλώσσας XML. Στην παρούσα εργασία υλοποιείται ο λεκτικός, ο συντακτικός και ο σημασιολογικός αναλυτής.

Ο κώδικας βρίσκεται αναρτημένος και στο github.

Οι παραδοχές που έχουν ληφθεί είναι οι εξής:

- Στο **BNF**, στις γραμματικές που παρουσιάζουν τα attributes των στοιχείων (linearAttr, relativeAttr, textViewAttr κλπ.), έχει θεωρηθεί ότι με οποιαδήποτε σειρά και αν εμφανιστούν τα στοχιεία, γίνονται αποδεκτά. Για παράδειγμα, αν σε ένα linearAttr δωθούν τα στοιχεία με την σειρά <width><height><comment>, αν και δεν έχει οριστεί στο BNF, θα γίνει αποδεκτό.

- Στο **BNF**, η εντολή %empty δηλώνει την κενή λέξη, δηλαδή μπορεί το συγκεκριμένο χαρακτηριστικό να είναι κενό.

- Στο **XML αρχείο**, τα σχόλια εκτείνονται σε μία γραμμή. Αν αποτελούν περισσότερες από μία γραμμές, η υλοποίηση δεν εγγυάται ότι θα λειτουργήσει με τον ανεμενόμενο τρόπο.

2 BNF

Το BNF είναι μία επίσημη μέθοδος για να περιγράψεις το συντακτικό μίας γλώσσας προγραμματισμού που διαγιγνώσκεται από Backus Naur Formas[1].

Για την επίτευξη αυτού, ορίστηκαν διάφορα σύμβολα, όπως γράμματα, αριθμοί και σύμβολα, με την βοήθεια των οποίων κατασκευάζονται διάφορες λέξεις. Όταν αυτές οι λέξεις και τα σύμβολα τοποθετηθούν με μία συγκεκριμένη σειρά παράγουν εκφράσεις που ικανοποιούν την γραμματική της γλώσσας μας. Με αυτόν τον τρόπο, κατασκευάστηκε μία απλή και αρκετά αφαιρετική μορφή συντακτικού δέντρου για την γλώσσα.

<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<alpha> ::= A | a | B | b | C | c | D | d | E | e | F | f |
G | g | H | h | I | i | J | j | K | k | L | l |
M | m | N | n | O | o | P | p | Q | q | R | r |
S | s | T | t | U | u | V | v | W | w | X | x |
Y | y | Z | z
<alphanumeric> ::= A | a | B | b | C | c | D | d | E | e | F | f |
G | g | H | h | I | i | J | j | K | k | L | l |
M | m | N | n | O | o | P | p | Q | q | R | r |
S | s | T | t | U | u | V | v | W | w | X | x |
Y | y | Z | z | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
8 | 9
<symbol> ::= , | . | ! | @ | # | $ | - | : | ; | ” ”
<underscore> ::= _
<colon> ::= :
<quotation> ::= ”
<startComment> ::= <!--
<endComment> ::= -->
<equalSign> ::= =
<whitespace> ::= ” ” | \t | \n
<startElem> ::= <
<endOneLineElem>::= />
<endMulLineElem>::= >

<relativeS> ::= "<RelativeLayout"
<relativeE> ::= "</RelativeLayout"
<linearS> ::= "<LinearLayout"
<linearE> ::= "</LinearLayout"
<radioGrS> ::= "<RadioGroup"
<radioGrE> ::= "</RadioGroup"
<progressBar> ::= "<ProgressBar"
<button> ::= "<Button"
<textview> ::= "<TextView"
<imageView> ::= "<ImageView"
<radioButton> ::= "<RadioButton"

<android> ::= "android"
<progress> ::= "progress"
<max> ::= "max"
<maxChildren> ::= "max_children"
<checkedbutton> ::= "checked_button"
<padding> ::= "padding"
<source> ::= "src"
<textColor> ::= "textColor"
<textToken> ::= "text"
<id> ::= "id"
<orientation> ::= "orientation"
<width> ::= "layout_width"
<height> ::= "layout_height"
<layout_value> ::= "match_parent" | "wrap_content" | <number>
<number> ::= <quotation><digit><quotation> | <quotation><number><digit><quotation>
<word> ::= <alphanumeric> | <alphanumeric><word>
<text> ::= <word> | <text><whitespace><text>

<comment> ::= <startComment><text><endComment> | %empty

<linearAttr> ::= <height><width><comment>
| <height><width><orientation><comment>
| <height><width><id><comment>
| <height><width><id><orientation><comment>
<reltiveAttr> ::= <height><width><comment>
| <height><width><id><comment>
<textViewAttr> ::= <height><width><textEl><comment>
| <height><width><textEl><id><comment>
| <height><width><textEl><textColor><comment>
| <height><width><textEl><textColor><id><comment>
<imageViewAttr> ::= <height><width><source><comment>
| <height><width><source><id><comment>
| <height><width><source><padding><comment>
| <height><width><source><padding><id><comment>
<buttonAttr> ::= <height><width><textEl><comment>
| <height><width><textEl><id><comment><comment>
| <height><width><textEl><padding><comment>
| <height><width><textEl><id><padding><comment>
<RadioGroupAtt> ::= <height><width><comment>
| <height><width><id><comment>
| <height><width><checkedButton><comment>
| <height><width><checkedButton><id><comment>
<RadioBtnAtt> ::= <height><width><text><comment>
| <height><width><text><id><comment>
<progressBarAt> ::= <height><width><comment>
| <height><width><id><comment>
| <height><width><max><comment>
| <height><width><progress><comment>
| <height><width><max><id><comment>
| <height><width><max><progress><comment>
| <height><width><id><progress><comment>
| <height><width><max><id><progress><comment>

<layoutHeight> ::= <android><colon><height><equalSign><layout_value><comment>
<layoutWidth> ::= <android><colon><width><equalSign><layout_value><comment>
<orientation> ::= <android><colon><orientation><equalSign><word><comment>
<id> ::= <android><colon><id><equalSign><word><comment>
<textEl> ::= <android><colon><textToken><equalSign><text><comment>
| <android><colon><textToken><equalSign><word><comment>
| <android><colon><textToken><equalSign><number><comment>
<textColor> ::= <android><colon><textColor><equalSign><word><comment>
<source> ::= <android><colon><source><equalSign><word><comment>
<padding> ::= <android><colon><padding><equalSign><number><comment>
<checkedButton> ::= <android><colon><checkedbutton><equalSign><word><comment>
<max> ::= <android><colon><max><equalSign><number><comment>
<progress> ::= <android><colon><progress><equalSign><number><comment>
<maxChildren> ::= <android><colon><maxChildren><equalSign><number><comment>

<radioButtonEl> ::= <radioButton><RadioBtnAtt><endOneLineElem>
<radioButtonEl><comment>
<imageViewEl> ::= <imageView><imageViewAttr><endOneLineElem><comment>
<textView> ::= <textview><textViewAttr><endOneLineElem><comment>
<button> ::= <button><buttonAttr><endOneLineElem><comment>
<progressBar> ::= <progressBar><progressBarAt><endOneLineElem><comment>
<radioGroup> ::= <radioGrS><RadioGroupAtt><endMulLineElem>
<radioButtonEl><radioGrE><endMulLineElem><comment>
<linearLayout> ::= <linearS><linearAttr><endMulLineElem><body>
<linearE><endMulLineElem><comment>
<relativeLayout>::= <relativeS><reltiveAttr><endOneLineElem><comment>
| <relativeS><reltiveAttr><endMulLineElem><body> <relativeE><endMulLineElem><co

<body> ::= <linearLayout><body><comment>
| <relativeLayout><body><comment>
| <radioGroup><body><comment>
| <progressBar><body><comment>

| <button><body><comment>
| <textView><body><comment>
| <imageViewEl><body><comment>
| <radioButtonEl><body><comment>

3 Λεκτική Ανάλυση

Το πρώτο βήμα ενός μεταγλωττιστή είναι η λεκτική ανάλυση. Ο λεκτικός αναλυτής καθορίζει πως τα περιεχόμενα ενός αρχείου σπάνε σε tokens, διαβάζοντας την είσοδο, η οποία μπορεί να είναι είτε ένας χαρακτήρας, είτε μία ροή από bytes και την σπάει σε tokens, χρησιμοποιώντας κάποιους καθορισμένους κανόνες, την γραμματική. Σκοπός του λεκτικού αναλυτή είναι να εντοπίσει μέσα σε ένα αρχείο τι είδους λέξεις υπάρχουν.[2]

Για αρχή κρίνεται αναγκαίος ο εντοπισμός των διάφορων δεσμευμένων λέξεων. Όπως είναι φανερό, δεσμευμένες λέξεις στην συγκεκριμένη γλώσσα αποτελούν, αρχικά, τα διάφορα ονόματα στοιχείων (π.χ LinearLayout, TextView κλπ.) και τα ονόματα χαρακτηριστικών (π.χ. android, id, layout_width κλπ) καθώς και διάφορα σύμβολα όπως τα <, :, = κ.ο.κ. Πέρα από όλες τις δεσμευμένες λέξεις ο λεκτικός αναλυτής οφείλει να εντοπίζει και άλλα στοιχεία μέσα στο αρχείο που θ ατου δίνεται όπως για παράδειγμα συμβολοσειρές, whitespaces και αριθμούς.

Έχοντας όλα τα παραπάνω κατά νου κατασκευάζουμε, αρχικά, ένα αρχείο lexer.h, σκοπός του οποίου είναι να δώσει σε κάθε token μία προκαθορισμένη από εμάς τιμή. Ένα επιπλέον token, το οποίο υποχρεωτικά πρέπει να υπάρχει, είναι το EOF (End Of File) με τιμή 0. Η σημασία αυτού του token είναι η ανακοίνωση του τέλους του αρχείου στον αναλυτή.

Την παραπάνω βιβλιοθήκη την συμπεριλαμβάνουμε, μαζί με μερικές επιπλέον, όπως stdio.h, stdlib.h και string.h στο lexer.l, στο λεκτικό αναλυτή δηλαδή. Ο λεκτικός αναλυτής αποτελείται από τρία μέρη, τα οποία χωρίζονται με τον διαχωριστή %%. Το πρώτο μέρος, γραμμένο σε C, αποτελεί το μέρος με τις δηλώσεις, όπου υπάρχουν οι βιβλιοθήκες, διάφορες ρυθμίσεις, δηλώσεις συναρτήσεων κλπ. Στο δεύτερο μέρος τοποθετούνται οι κανόνες, δηλαδή τα regular expressions, επιστρεφόμενες τιμές, ρουτίνες για την εύρεση και διόρθωση λαθών κ.α. Στο τρίτο κομμάτι, γραμμένο επίσης σε C, υπάρχουν οι συναρτήσεις του χρήστη, όπως η συνάρτηση main, συναρτήσεις εκτύπωσης μηνυμάτων κ.α.

Έτσι, στο πρώτο μέρος έχουν τοποθετησθεί οι βιβλιοθήκες, διάφορες global μεταβλητές που απαιτούνται για την εκτύπωση μηνυμάτων προς τον χρήστη, ρυθμίσεις και μερικά βασικά RegEx για την διευκόλυνση κατασκευής των κανόνων. Η ρύθμιση noyywrap, χρησιμοποιείται έτσι ώστε ο λεκτικός αναλυτής να διαβάσει μόνο ένα αρ-

χείο, ενώ η ρύθμιση yylineno είναι υπεύθυνη για τον υπολογισμό της γραμμής στην οποία βρισκόμαστε. Τέλος, κατασκευάστηκαν βασικές κανονικές εκφράσεις, όπως για παράδειγμα, το DIGIT, που είναι ένα ψηφίο στο διάστημα [0,9], το NUMBER, όπου αναπαριστά ένα αυστηρά θετικό ακέραιο αριθμό ή το μηδέν, καθώς και εκφράσεις για τον εντοπισμό αλφαριθμητικών καθώς και κειμένου. Κρίθηκε αναγκαία η υλοποίηση ενός ξεχωριστού κανόνα για τον εντοπισμό ενός αριθμού, καθώς αυτός δεν θα θέλαμ ενα ξεκινάει με μηδενικά. Για παράδειγμα ο αριθμός 0000012, είναι όντως το 12 αλλά δεν θα θέλαμε να βρίσκεται στην πρώτη μορφή μέσα στον κώδικα, όπως και ο αριθμός μηδέν, αρκεί να γραφτεί ως 0 και όχι σε οποιαδήποτε άλλη μόρφή, π.χ. 00000. Σημαντικό σε αυτό το σημείο είναι να τονισθεί, ότι ο κώδικας σε C πρέπει να καταγράφεται μέσα σε ένα πλαίσιο που αναπαρίσταται με την μορφή %{ %} ενώ έξω από αυτό γράφουμε κώδικα flex.

Στο δεύτερο μέρος του λεκτικού αναλυτή, καταγράφονται οι κανόνες για τον εντοπισμό των διάφορων tokens της γλώσσας, τυπώνεται ένα μήνυμα εύρεσης και επιστρέφεται ο κωδικός του token όπως έχει δηλωθεί στο αρχείο lexer.h. Όταν εντοπιστεί τέλος αρχείου επιστρέφεται ο κωδικός του αντίστοιχου token, που είναι το 0, τα whitespaces σε όλο το αρχείο κώδικα αγνοούνται ενώ να βρεθεί κάποιο άλλο σύμβολο το οποίο δεν έχει συμπεριληφθεί στην γραμματική, εμφανίζεται αντίστοιχο μήνυμα σφάλματος λέγοντας πως δεν το αναγνωρίζει.

Πέρα από τα διάφορα attributes που περιέχει η γραμματική και είναι εύκολο σχετικά να εντοπιστούν από μερικούς κανόνες, παρατηρούμε, ότι υπάρχουν και διάφορες δομές με συγκεκριμένη μορφοποίηση, όπως τα σχόλια που πρέπει να συμπεριλαμβάνονται ανάμεσα σε <!– –>, τα διάφορα στοιχεία που πρέπει να είναι της μορφής <A-SPECIFIC-STRING> ή </A-SPECIFIC-STRING> καθώς και οι τιμές των χαρακτηριστικών, που πρέπει να είναι υποχρεωτικά ανάμεσα σε παρενθέσεις (”). Για τον σκοπό αυτό, κατασκευάστηκαν υπο-λεκτικοί αναλυτές για τον εντοπισμό τέτοιων δομών, η δήλωση των οποίων έγινε και στο πρώτο μέρος του αρχείου lexer με την ρύθμιση %x.

Όταν εντοπίζεται το token <!– ξεκινάει ο υπο-λεκτικός αναλυτής COMMENT, ο οποίος δοαβάζει το περιεχόμενο του σχολίου, ελέγχοντας ότι δεν υπάρχει το σύμβολο – και με το που εντοπίσει το σύμβολο –>, δηλαδή το τέλος του σχολίου, καλεί τον αρχικό λεκτικό αναλυτή για να συνεχίσει την ανάγνωση του υπόλοιπου αρχείου. Το περιεχόμενο του σχολίου το αγνοούμε, καθώς κατά την μετάφραση του κώδικα δεν

χρειάζεται. Το μόνο που ελέγχουμε είναι η ύπαρξη του συμβόλου –, που προκαλεί συντακτικό λάθος, ενώ οποιαδήποτε άλλη μορφή είναι αποδεκτή. Οι άλλοι δύο υπολεκτικοί αναλυτές κατασκευάστηκαν με το ίδιο σκεπτικό, μόνο που σε αυτές τις περιπτώσεις είναι σημαντικό και το περιεχόμενο τους.

Στο τελευταίο κομμάτι του lexer, κατασκευάστηκαν συναρτήσεις για την εκτύπωση των διάφορων token κατά τη διάρκεια του debugging καθώς και για την εκτύπωση λαθών. Σε αυτό το σημείο δημιουργήθηκε και η συνάρτηση main, σκοπός της οποίας είναι να διαβάζει το αρχείο, έως ότου εντοπίσει το EOF.

Σημαντικό είναι να σημειωθεί, ότι ο λεκτικός αναλυτής ελέγχει μόνο αν τα tokens, οι λέξεις και τα σύμβολα δηλαδή, είναι επιτρεπτά στοιχεία της γλώσσας, ενώ αδιαφορεί για την σύνταξη, την οποία την ελέγχει ο συντακτικός αναλυτής, όπως παρουσιάζεται στο κεφάλαιο 4. Για παράδειγμα, ο λεκτικός αναλυτής θα αποδεχόταν χωρίς την εμφάνιση κάποιου error τον παρακάτω - συνατκτικά λάθος - κώδικα σε C.

```
x int ; main
( 0 { return
} 5 = ; )
```

4 Συντακτική Ανάλυση

Εφόσον η λεκτική ανάλυση δεν εντοπίσει λάθος, ακολουθεί η συντακτική ανάλυση, σκοπός της οποίας είναι να ελέγξει ότι τα tokens που εντοπίστηκαν βρίσκονται σε σωστή σειρά, όπως απαιτείται από την γλώσσα. Αυτό επιτυγχάνεται με την δημιουργία ενός parse tree, το οποίο κατασκευάζεται από την προκαθορισμένη γραμματική της γλώσσα. Εφόσον η είσοδος που δίνεται, μπορεί να παραχθεί από το δέντρο, τότε το συντακτικό είναι σωστό, ενώ σε αντίθετη περίπτωση εμφανίζεται συντακτικό σφάλμα.

Και σε αυτήν την περίπτωση, ο κώδικας χωρίζεται σε τρία μέρη χωριζόμενα από το %%, όπως ακριβώς και στον lexer, μόνο που τώρα, στο δεύτερο μέρος τοποθετούνται οι συντακτικοί κανόνες.

Στο πρώτο μέρος τοποθετούνται διάφορες βιβλιοθήκες της C που απαιτούνται καθώς και συναρτήσεις που έχουν οριστεί σε αρχεία που ενσωματώνονται στο syntax file, γι' αυτό και ενσωματώνονται με την λέξη-κλειδί extern. Στο ίδιο μέρος, τοποθετούνται και διάφορες ρυθμίσεις του bison, όπως το parse.error verbose, σκοοπός του οποίου είναι να εμφανίζει πιο κατανοητά προς εμάς μηνύματα λάθους, όταν εντοπίζεται κάποιο συντακτικό λάθος, καθώς και από ποιον κανόνα ξεκινάει η συντακτική ανάλυση, που στην συγκεκριμένη περίπτωση είναι η program, όπως θα αναλυθεί στην συνέχεια. Στο σημείο αυτό, πρέπει να οριστούν και τα tokens που θα αναγνωριστούν από τον lexer. Αυτό επιτυγχάνεται μέσω της ρύθμισης %token, η δομή της οποίας είναι: %token <TYPE> <TOKEN_NAME> <VALUE> <COMMENT>. Τα <TYPE>, <VALUE> και <COMMENT> είναι προαιρετικά. Σε όλα τα tokens, με εξαίρεση το EOF που παίρνει τιμή 0, δεν αναθέτουμε κάποια προκαθορισμένη τιμή αλλά αφήνουμε το πρόγραμμα να αναθέσει μόνο του. Τις τιμές που έχουν ανατεθεί, μπορούμε να τις εντοπίσουμε στο αρχείο syntax.output. Το πεδίο <COMMENT> παίρνει μία συμβολοσειρά με κάποια περιγραφή του συγκεκριμένου tokens και εμφανίζεται όταν προκληθεί κάποιο error, το οποίο οφείλεται σε έλλειψη του εν λόγω στοιχείου, ενώ το <TYPE> δέχεται τον τύπο των τιμών που μπορεί να λάβει το συγκεκριμένο token. Για να επιτευχθεί αυτό κατασκευάζεται ένα union το οποίο περιέχει τους διαφορετικούς τύπους μεταβλητών που απαιτούνται μαζί με ένα χαρακτηριστικό όνομα, στην συγκεκριμένη περίπτωση έχουμε τους τύπους int με όνομα intValue και char * με όνομα strValue.

Στην συνέχεια, ορίζονται οι κανόνες που απαρτίζουν την γραμματική. Για να πραγματοποιηθεί αυτό, ξεκινάει η υλοποίηση από μέσα προς τα έξω, δηλαδή πρώτα υλοποιούνται τα διάφορα attributes και έχοντας αυτά δημιουργείται το σώμα των στοιχείων. Έτσι, ξεκινώντας από την δομή των tokens, δημιουργούνται κανόνες που αφορούν μεμονωμένα attributes. Το σώμα κάθε στοιχείου αποτελείται από τον συνδυασμό των attributes που κατασκευάστηκαν προηγουμένως. Η σειρά με την οποία θα εμφανίζονται αυτά δεν παίζει κάποιον ρόλο και γι' αυτό κατασκευάστηκε πρόγραμμα σε python που παράγει όλα τα permutations που απαιτούνται. Ο κώδικας δίνεται στον φάκελο pythonProject μαζί με το απαιτούμενο virtual environment (**για Linux**). Έπειτα, με την βοήθεια αυτών των σωμάτων ορίζονται και ολόκληρα τα στοιχεία της XML γλώσσας και κατά συνέπεια η δομή ολόκληρου του προγράμματος.

Υπάρχει περίπτωση, όμως, κάποιο από τα υποχρεωτικά ορίσματα να λείπει από κάποιον κώδικα και σε αυτή την περίπτωση πρέπει να εμφανίζεται συντακτικό σφάλμα στον χρήστη. Αυτό επιτυγχάνεται με χρήση της δεσμευμένης λέξεις error, η οποία υπονοεί ότι στην συγκεκριμένη θέση υπάρχει κάποιο λάθος, σε αυτή την περίπτωση λείπει κάποιο χαρακτηριστικό. Επειδή και τώρα δεν έχει σημασία η σειρά εμφάνισης τους, παίρνουμε όλα τα πιθανά permutations που μπορούν να δημιουργηθούν. Όταν εντοπίζεται κάποιο τέτοιο συντακτικό λάθος εμφανίζεται μήνυμα σφάλματος, με χρήση της συνάρτησης yyerror() ενώ η εντολή yyerrok χρησιμοποιείται ώστε ακόμα και αν υπάρχει λάθος να αγνοηθεί και να συνεχίσει την εκτέλεση ο συντακτικός αναλυτής αντί να σταματήσει. Με αυτόν τον τρόπο εμφανίζονται όλα τα λάθη που υπάρχουν στον κώδικα και δεν σταματάμε στο πρώτο. Σημαντικό είναι να τονισθεί, ότι ο δοσμένος αναλυτής δεν καλύπτει όλες τις περιπτώσεις καθώς είναι πιθανόν να λείπουν περισσότερα του ενός attributes, ενώ ο συγκεκριμένος ανιχνεύει μόνο τις περιπτώσεις που λείπει μόνο ένα.

Τέλος, κατασκευάζεται η main(), η οποία ανοίγει για ανάγνωση το αρχείο και εκτελεί τον συντακτικό αναλυτή. Εφόσον δεν εντοπιστεί κάποιο λάθος - είτε συντακτικό είτε κατά την διάρκεια του λεκτικού αναλυτή - εμφανίζει ολόκληρο τον δοθέντα κώδικα. Η συνάρτηση main() του λεκτικού αναλυτή έχει αφαιρεθεί.

5 Σημασιολογική Ανάλυση

Στο τελευταίο στάδιο είναι η σημασιολογική ανάλυση, σκοπός του οποίου είναι να εξασφαλίσει, ότι η δηλώσεις των μεταβλητών και οι εκφράσεις του προγράμματος έχουν σημασιολογική έννοια, δηλαδή ακολουθούν ορισμένους κανόνες και περιορισμούς [3].

Αρχικά, πρέπει να οριστούν τα scopes. Κάθε μεταβλητή έχει το δικό της εύρος και είναι προσβάσιμη μόνο σε αυτό, ενώ εκτός αυτού όχι. Το πρώτο scope είναι το 0 και όσο προχωράμε προς τα μέσα αυξάνεται η τιμή. Μεταβλητές που βρίσκονται σε μικρότερο scope είναι ορατές και στα εσωτερικά (με μεγαλύτερη τιμή) scope.

Στη συγκεκριμένη γλώσσα, μπορούμε εύκολα να παρατηρήσουμε ότι ένα scope ξεκινάει μόλις εμφανιστεί η κάποιο open tag ενώ σταματάει στο αντίστοιχο closing tag.

Για την αποθήκευση των πληροφοριών αυτών, υλοποιήθηκε ένα hash table (βρίσκεται στον φάκελο extras) που κρατάει τις σημαντικές πληροφορίες, δηλαδή το όνομα της μεταβλητής, την τιμή της καθώς και το scope στο οποίο βρίσκεται. Στην συνέχεια, ορίστηκαν και συναρτήσεις, σκοπός των οποίων είναι να βοηθήσουν στην εισαγωγή και έλεγχο ορισμένων τιμών. Ενδεικτικά, η συνάρτηση add_id ελέγχει για αρχή, αν η τιμή του id που πρόκειται να προστεθεί (κατ' επέκταση η τιμή του id που έχει δηλωθεί) έχει ξαναχρησιμοποιηθεί. Εφόσον αυτό δεν ισχύει, προστίθεται η τιμή στο hash table. Κατ' αναλογία, έχει κατασκευαστεί και η συνάρτηση check_maxChildren_radioGroup, η οποία ελέγχει αν το πλήθος των radioButton μέσα σρο redioGroup είναι ίσος με την τιμή που έχει οριστεί στο attribute android:max. Αυτό επιτυγχάνεται, με την σύγκριση της τιμής του android:max και της μεταβλητής childrenCounter, σκοπός της οποίας είναι να δείχνει πόσα radioButtons υπάρχουν στο συγκεκριμένο radioGroup.

6 Παραδείγματα

6.1 Λεκτική ανάλυση

6.1.1 example1.xml

Το πρώτο παράδειγμα, είναι ένας κώδικας XML που ακολουθεί και την σωστό συντακτικό και τη σωστή γραμματική. Ο lexer εντοπίζει κανονικά τα tokens και τα εμφανίζει στην οθόνη μαζί με τον κωδικό που τους έχει ανατεθεί στο αρχείο lexer.h.

```
<LinearLayout
   android:layout_width=”match_parent”
   android:layout_height=”match_parent”
   android:orientation=”vertical”>

   <TextView
      android:layout_width=”20”
      android:id=”TV1”
      android:text=”Dummy text”/>

   <RadioGroup
      android:layout_width=”wrap_content”
      android:layout_height=”wrap_content”>

      <RadioButton
         android:layout_width=”wrap_content”
         android:layout_height=”wrap_content”
         android:id=”RB1”
         android:text=”Option 1”/>

      <RadioButton
         android:layout_width=”wrap_content”
         android:layout_height=”wrap_content”
         android:id=”RB2”
```

```
        android:text="Option 2"/>

    </RadioGroup>
<!-- This is a comment, ignore me! -->
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="image_name"
            android:padding="50"/>

    </RelativeLayout>

</LinearLayout>
```

Και η έξοδος του προγράμματος.

```
Found token 'LinearLayout' (16) at line 2
Found token 'android' (30) at line 3
Found token ':' (3) at line 3
Found token 'layout_width' (19) at line 3
Found token '=' (1) at line 3
Found token 'match_parent"' (29) at line 3
Found token 'android' (30) at line 4
Found token ':' (3) at line 4
Found token 'layout_height' (18) at line 4
Found token '=' (1) at line 4
Found token 'match_parent"' (29) at line 4
Found token 'android' (30) at line 5
Found token ':' (3) at line 5
Found token 'orientation' (20) at line 5
```

Found token '=' (1) at line 5
Found token 'vertical"' (33) at line 5
Found token '>' (5) at line 5
Found token 'TextView' (7) at line 7
Found token 'android' (30) at line 8
Found token ':' (3) at line 8
Found token 'layout_width' (19) at line 8
Found token '=' (1) at line 8
Found token 'android' (30) at line 9
Found token ':' (3) at line 9
Found token 'id' (21) at line 9
Found token '=' (1) at line 9
Found token 'TV1"' (33) at line 9
Found token 'android' (30) at line 10
Found token ':' (3) at line 10
Found token 'text' (22) at line 10
Found token '=' (1) at line 10
Found token 'Dummy text"' (31) at line 10
Found token '/>' (4) at line 10
Found token 'RadioGroup' (15) at line 12
Found token 'android' (30) at line 13
Found token ':' (3) at line 13
Found token 'layout_width' (19) at line 13
Found token '=' (1) at line 13
Found token 'wrap_content"' (29) at line 13
Found token 'android' (30) at line 14
Found token ':' (3) at line 14
Found token 'layout_height' (18) at line 14
Found token '=' (1) at line 14
Found token 'wrap_content"' (29) at line 14
Found token '>' (5) at line 14
Found token 'RadioButton' (8) at line 16
Found token 'android' (30) at line 17

Found token ':' (3) at line 17
Found token 'layout_width' (19) at line 17
Found token '=' (1) at line 17
Found token 'wrap_content"' (29) at line 17
Found token 'android' (30) at line 18
Found token ':' (3) at line 18
Found token 'layout_height' (18) at line 18
Found token '=' (1) at line 18
Found token 'wrap_content"' (29) at line 18
Found token 'android' (30) at line 19
Found token ':' (3) at line 19
Found token 'id' (21) at line 19
Found token '=' (1) at line 19
Found token 'RB1"' (33) at line 19
Found token 'android' (30) at line 20
Found token ':' (3) at line 20
Found token 'text' (22) at line 20
Found token '=' (1) at line 20
Found token 'Option 1"' (31) at line 20
Found token '/>' (4) at line 20
Found token 'RadioButton' (8) at line 22
Found token 'android' (30) at line 23
Found token ':' (3) at line 23
Found token 'layout_width' (19) at line 23
Found token '=' (1) at line 23
Found token 'wrap_content"' (29) at line 23
Found token 'android' (30) at line 24
Found token ':' (3) at line 24
Found token 'layout_height' (18) at line 24
Found token '=' (1) at line 24
Found token 'wrap_content"' (29) at line 24
Found token 'android' (30) at line 25
Found token ':' (3) at line 25

Found token 'id' (21) at line 25
Found token '=' (1) at line 25
Found token 'RB2"' (33) at line 25
Found token 'android' (30) at line 26
Found token ':' (3) at line 26
Found token 'text' (22) at line 26
Found token '=' (1) at line 26
Found token 'Option 2"' (31) at line 26
Found token '/>' (4) at line 26
Found token '/RadioGroup' (14) at line 28
Found token '>' (5) at line 28
Found token 'RelativeLayout' (9) at line 30
Found token 'android' (30) at line 31
Found token ':' (3) at line 31
Found token 'layout_width' (19) at line 31
Found token '=' (1) at line 31
Found token 'match_parent"' (29) at line 31
Found token 'android' (30) at line 32
Found token ':' (3) at line 32
Found token 'layout_height' (18) at line 32
Found token '=' (1) at line 32
Found token 'wrap_content"' (29) at line 32
Found token '>' (5) at line 32
Found token 'ImageView' (10) at line 34
Found token 'android' (30) at line 35
Found token ':' (3) at line 35
Found token 'layout_width' (19) at line 35
Found token '=' (1) at line 35
Found token 'wrap_content"' (29) at line 35
Found token 'android' (30) at line 36
Found token ':' (3) at line 36
Found token 'layout_height' (18) at line 36
Found token '=' (1) at line 36

Found token 'wrap_content"' (29) at line 36
Found token 'android' (30) at line 37
Found token ':' (3) at line 37
Found token 'src' (23) at line 37
Found token '=' (1) at line 37
Found token 'image_name"' (32) at line 37
Found token 'android' (30) at line 38
Found token ':' (3) at line 38
Found token 'padding' (24) at line 38
Found token '=' (1) at line 38
Found token '/>' (4) at line 38
Found token '/RelativeLayout' (13) at line 40
Found token '>' (5) at line 40
Found token '/LinearLayout' (12) at line 42
Found token '>' (5) at line 42

6.1.2 example2.xml

Στο δεύτερο παράδειγμα, κατασκευάστηκε ένας κώδικας XML, ο οποίος αν και αποτελείται από tokens τα οποία αναγνωρίζονται από την γλώσσα, δεν ακολουθεί το επιθυμητό συνατκτικό. Παρ' όλα αυτά βλέπουμε ότι τα αναγνωρίζει κανονικά ενώ δεν εμφανίζει κάποιο λάθος.

```
android:layout_height="84"
<RadioGroup>

<!--
   This
is
-->

<RadioButton
src
android:layout_width="78"
```

android

:

=

/>

”file/path”

Και η έξοδος είναι η παρακάτω

Found token ’android’ (30) at line 2
Found token ’:’ (3) at line 2
Found token ’layout_height’ (18) at line 2
Found token ’=’ (1) at line 2
Found token ’RadioGroup’ (15) at line 3
Found token ’>’ (5) at line 3
Found token ’RadioButton’ (8) at line 10
Found token ’src’ (23) at line 11
Found token ’android’ (30) at line 12
Found token ’:’ (3) at line 12
Found token ’layout_width’ (19) at line 12
Found token ’=’ (1) at line 12
Found token ’android’ (30) at line 16
Found token ’:’ (3) at line 17
Found token ’=’ (1) at line 18
Found token ’/>’ (4) at line 20
Found token ’file/path”’ (31) at line 22

6.1.3 example3.xml

Στο τρίτο παράδειγμα, δημιουργήθηκε κώδικας XML, ο οποίος όμως περιέχει λεκτικά λάθη, όπως αρνητικές τιμές σε αριθμούς και σύμβολα, τα οποία δεν είναι

μέρος της γραμματικής του. Όλα αυτά τα αναγνωρίζει και τυπώνει το λάθος μαζί με την γραμμή που εμφανίζεται. Ο αριθμός της γραμμής είναι ο επόμενος από εκείνον, όπου εμφανίζεται το λάθος.

src
:
=
padding
”-65”

android

<!-- : -- android:
id =”bye” --- -->

src

= android

file/path

:

Και η έξοδος του προγράμματος.

Found token ’src’ (23) at line 2
Found token ’:’ (3) at line 3
Found token ’=’ (1) at line 4
Found token ’padding’ (24) at line 5
ERROR at line 6: Negative Number
Found token ’android’ (30) at line 8
ERROR at line 10: Comments should not contain --
Found token ’src’ (23) at line 13
Found token ’=’ (1) at line 15

Found token 'android' (30) at line 15
ERROR at line 17: Unrecognised token 'file'
ERROR at line 17: Unrecognised token '/'
ERROR at line 17: Unrecognised token 'path'
Found token ':' (3) at line 19

6.2 Συντακτική ανάλυση

6.2.1 example1.xml

Σε αυτό το παράδειγμα, δημιουργήθηκε κώδικας σε XML που ακολουθεί τους συντακτικούς κανόνες. Ο compiler δεν εντοπίζει κάποιο λάθος, οπότε τυπώνει στην οθόνη τον κώδικα που δόθηκε.

```
<LinearLayout
   android:layout_width=”match_parent”
   android:layout_height=”match_parent”
   android:orientation=”vertical”>

   <TextView
      android:layout_width=”20”
      android:id=”TV1”
      android:text=”Dummy text”
      android:layout_height=”65”/>

   <RadioGroup
      android:layout_width=”wrap_content”
      android:layout_height=”wrap_content”
      android:max_children=”2”
      android:checked_button=”RB1”>

      <RadioButton
         android:layout_width=”wrap_content”
         android:layout_height=”wrap_content”
         android:id=”RB1”
         android:text=”Option 1”/>

      <RadioButton
         android:layout_width=”wrap_content”
         android:layout_height=”wrap_content”
         android:id=”RB2”
```

```
        android:text="Option 2"/>

    </RadioGroup>
<!-- This is a comment, ignore me! -->
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="image_name"
            android:padding="50"/>

    </RelativeLayout>

</LinearLayout>
```

6.2.2 example2.xlm

Με αυτό το παράδειγμα, παρατηρούμε, ότι, όταν ο κώδικας δεν ξεκινάει με RelativeLayout ή LinearLayout εμφανίζεται συντακτικό λάθος.

```
    <RadioGroup
    android:layout_width="58"
    android:layout_height="wrap_content"
    android:max_children="2"
    android:checked_button="first">

    <RadioButton
        android:layout_width="65"
        android:layout_height="wrap_content"
        android:id="first"
        android:text="Delete"/>
```

<RadioButton
android:layout_width=”wrap_content”
android:layout_height=”35”
android:id=”second”
android:text=”Click me”/>

</RadioGroup>

ERROR at line 20: Should start with RelativeLayout or LinearLayout

6.2.3 example3.xml

Στο τρίτο παράδειγμα βλέπουμε πως αν κάποιο υποχρεωτικά χαρακτηριστικά παραληφθεί τότε εμφανίζεται συντακτικό σφάλμα όπου ενημερώνει για το ποιο χαρακτηριστικό απουσιάζει.

<RelativeLayout
android:layout_height=”84”
android:layout_width=”98”>

<RadioGroup
android:layout_height=”54”
android:layout_width=”match_parent”
android:id=”radioGroup”
android:checked_button=”RB1”>

<RadioButton
android:layout_height=”54”
android:id=”RB1”
android:layout_width=”wrap_content”
android:text=”This is a text. Ignore it!”/>

<RadioButton
android:layout_height=”54”

```
    android:id="RB2"
    android:text="This is another text. Ignore It!"/>

</RadioGroup>

<RelativeLayout
    android:layout_height="54"
    android:layout_width="78">

    <ImageView
        android:layout_width="98"
        android:layout_height="105"
        android:src="test"/>
</RelativeLayout>

</RelativeLayout>
```

Ο compiler επιστρέφει το εξής

```
ERROR at line 10: syntax error, unexpected >, expecting android
ERROR at line 10: android:maxChildren is mandatory!
ERROR at line 21: syntax error, unexpected />, expecting android
ERROR at line 21: android:layoutWidth is mandatory!
```

6.3 Σημασιολογική ανάλυση

6.3.1 example1.xml

Στο πρώτο παράδειγμα βλέπουμε ότι, όταν κάποια τιμή στο χαρακτηριστικό id δεν είναι μοναδική εμφανίζεται σφάλμα από τον compiler, όπως ακριβώς και όταν η τιμή του layout_width ή και του padding δεν είναι ένας θετικός ακέραιος.

```
<LinearLayout
   android:layout_height=”wrap_content”
   android:layout_width=”match_parent”
   android:id=”ID1”>

   <ImageView
      android:layout_height=”52”
      android:layout_width=”-9”
      android:src=”foo”
      android:padding=”5.23”
      android:id=”ID1”/>

</LinearLayout>
```

```
ERROR at line 9: Negative Number is not allowed
ERROR at line 11: Only integer numbers allowed
ERROR at line 12: Values of ‘id‘ should be unique
```

6.3.2 example2.xml

Στο δεύτερο παράδειγμα γίνεται επίδειξη πιο περίπλοκων σημασιολογικών ελέγχων, όπως η τιμή του progress ενός ProgressBar στοιχείου, η οποία πρέπει να είναι στο διάστημα 0..max. Οποιαδήποτε άλλη τιμή εμφανίζει σφάλμα. Επίσης, παρατηρούμε ότι πρέπει να βρίσκονται μέσα σε ένα RadioGroup τόσα στοιχεία όσα δηλώνει και η τιμή του χαρακτηριστικού max_children ενώ η τιμή του checked_button πρέπει να είναι το id κάποιου εμφωλευμένου στοιχείου. Σε αντίθετη περίπτωση εμφανίζεται σημασιολογικό λάθος.

```
<RelativeLayout
    android:layout_height=”match_parent”
    android:layout_width=”wrap_content”
    android:id=”ID1”>

    <ProgressBar
        android:layout_height=”50”
        android:layout_width=”50”
        android:max=”10”
        android:progress=”50”/>

    <RadioGroup
        android:layout_height=”59”
        android:layout_width=”62”
        android:checked_button=”unknown”
        android:max_children=”50”>

        <RadioButton
            android:layout_height=”match_parent”
            android:layout_width=”match_parent”
            android:id=”RB1”
            android:text=”Hey”/>

        <RadioButton
            android:layout_height=”match_parent”
            android:layout_width=”match_parent”
            android:text=”Click me!”
            android:id=”RB2”/>

    </RadioGroup>

    <ProgressBar
        android:layout_height=”50”
```

android:layout_width=”50”
android:max=”10”
android:progress=”-2”/>

</RelativeLayout>

ERROR at line 11: Value of android:progress should be less or equal to max_value (10)
ERROR at line 31: The value of checked_button should exists in radioButton inside radioGroup
ERROR at line 31: Should be 50 radioButton inside RadioGroup
ERROR at line 37: Negative Number is not allowed

Αναφορές

[1] Bnf notation in compiler design. https://www.geeksforgeeks.org/bnf-notation-in-compiler-design/. Accessed: 2023-06-14.

[2] What is a lexer ? known also as tokenizer or scanner - lexical analysis. https://datacadamia.com/code/compiler/lexer. Accessed: 2023-05-16.

[3] Semantic analysis in compiler design. https://www.geeksforgeeks.org/semantic-analysis-in-compiler-design/. Accessed: 2023-06-11.