Πανεπιστήμιο Ιωαννίνων

Πολυτεχνική Σχολή

Τμήμα Μηχανικών Η/Υ και Πληροφορικής

**Μεταφραστές**

Διδάσκων: Γ. Μανής

Ακαδημαϊκό έτος: 2018-2019

Καρέτση Φωτεινή Α.Μ.: 2990

Μπελλά Ελευθερία Α.Μ.: 3039

**1η φάση: Λεκτικός και Συντακτικός αναλυτής**

**Λεκτικός αναλυτής**

Ο λεκτικός αναλυτής της γλώσσας Starlet είναι μια συνάρτηση που διαβάζεις γράμμα-γράμμα το πηγαίο πρόγραμμα και κάθε φορά επιστρέφει την αμέσως επόμενη λεκτική μονάδα. Συγκεκριμένα, ο λεκτικός αναλυτής (lex) επιστρέφει δύο τιμές, το string token, που είναι η λεκτική μονάδα που έχει αναγνωρίσει, και το string tokenID, που αποτελεί τον τύπο της λεκτικής μονάδας που αναγνωρίστηκε. Πιο αναλυτικά, μπορεί να πάρει μία από τις ακόλουθες τιμές:

* Δεσμευμένες λέξεις, όπως έχουν οριστεί στη γλώσσα της Starlet (program, if, while, forcase, endforcase κτλ).
* Την τιμή id, που αντιστοιχεί σε αναγνωριστικό (μη δεσμευμένη λέξη). Το αναγνωριστικό αυτό μπορεί να αποτελεί όνομα μεταβλητής, συνάρτησης ή σχόλιο.
* Την τιμή constant, που αντιστοιχεί σε αριθμητική σταθερά.
* Έναν εκ των παρακάτω χαρακτήρων που επιτρέπονται στη γλώσσα, δηλαδή τα σύμβολα αριθμητικών πράξεων *(«+», «-», «\*», «/»),* τους τελεστές συσχέτισης *(«<», «>», «=», «<=», «>=», «<>»)*, το σύμβολο ανάθεσης *(«:=»)*, τους *διαχωριστές («;», «,», «:»)*, τα σύμβολα ομαδοποίησης *(«(»,«)»,«[»,«]»)* και διαχωρισμού σχολίων *(«/\*»,«\*/»,«//»)*. Για καθεμία από τις παραπάνω περιπτώσεις, η τιμή του token ταυτίζεται με την τιμή του tokenID.

Ο λεκτικός αναλυτής, όσον αφορά την εσωτερική λειτουργία του, λειτουργεί σαν ένα αυτόματο καταστάσεων, το οποίο ξεκινά από μια αρχική κατάσταση, ανάλογα με το χαρακτήρα που διαβάζει μεταβαίνει σε κάποια άλλη κατάσταση και τελικά καταλήγει σε μία τελική. Σε περίπτωση μη επιτρεπτής λειτουργίας, όπως αυτές έχουν οριστεί από τη γλώσσα, ο αναλυτής **εμφανίζει κατάλληλο μήνυμα λάθους** και τερματίζει τη λειτουργία του (υλοποίηση συνάρτησης displayError).

Επιπλέον, ο λεκτικός αναλυτής διατηρεί ενημερωμένη την τρέχουσα γραμμή του αρχείου που διαβάζει, κάθε φορά που συναντά τον χαρακτήρα «\n».

Οι καταστάσεις στις οποίες μπορεί να μεταβεί ο lex() είναι οι εξής:

1. Κατάσταση 0: Αρχική κατάσταση λεκτικού αναλυτή
2. Κατάσταση 1: Αναγνώριση λεκτικού (δεσμευμένης λέξης ή αναγνωριστικού id)
3. Κατάσταση 2: Αναγνώριση αριθμητικής σταθεράς
4. Κατάσταση 3: Αναγνώριση /
5. Κατάσταση 4: Αναγνώριση σχολίου μιας γραμμής
6. Κατάσταση 5: Αναγνώριση έναρξης σχολίων πολλαπλών γραμμών
7. Κατάσταση 6,11: Αναγνώριση εμφωλευμένων σχολίων
8. Κατάσταση 7: Αναγνώριση τερματισμού σχολίων πολλαπλών γραμμών
9. Κατάσταση 8: Αναγνώριση τελεστών που περιλαμβάνουν το χαρακτήρα <
10. Κατάσταση 9: Αναγνώριση τελεστών που περιλαμβάνουν το χαρακτήρα >
11. Κατάσταση 10: Αναγνώριση τελεστών που περιλαμβάνουν το χαρακτήρα :
12. Κατάσταση 100: Τελική κατάσταση

Στις περιπτώσεις που κάποιος χαρακτήρας αναγνωρίζεται απευθείας χωρίς να εξετάσουμε κάποιο άλλο ενδεχόμενο, μεταβαίνει αμέσως σε τελική κατάσταση.

Κάθε φορά που καλείται ο lex() ξεκινά από την **κατάσταση 0.** Εάν διαβάσει κάποιο λευκό χαρακτήρα παραμένει σε αυτή την κατάσταση, ενώ αν διαβάσει κάποιο κεφαλαίο ή μικρό γράμμα μεταβαίνει στην κατάσταση 1. Όμοια, αν διαβάσει ψηφίο μεταβαίνει στην κατάσταση 2 και με «/» στην κατάσταση 3. Εάν διαβάσει «<», «>», «:» πηγαίνει στις καταστάσεις 8,9 και 10 αντίστοιχα, ενώ αν συναντήσει κάποιον από τους χαρακτήρες «+»,«-»,«\*», «,», «;», «=», «(», «)», «[», «]» ή τέλος αρχείου (EOF), μεταβαίνει στην κατάσταση 100 και επιστρέφει τα token και tokenID. Εάν συναντήσει χαρακτήρα που δεν ανήκει στο λεξιλόγιο της Starlet εμφανίζει κατάλληλο μήνυμα λάθους.

Στην **κατάσταση 1** παραμένει όσο διαβάζει γράμματα ή ψηφία. Διαφορετικά μεταβαίνει στην κατάσταση 100. Επίσης, έχει χρησιμοποιηθεί ένας counter που μετρά το πλήθος των χαρακτήρων του αλφαριθμητικού και επιστρέφει τους 30 πρώτους εάν το πλήθος αυτό υπερβεί την τιμή 30.

Στην **κατάσταση 2** παραμένει όσο διαβάζει ψηφία. Εάν διαβάσει γράμμα επιστρέφει μήνυμα λάθους, καθώς δεν είναι αποδεκτά ονόματα μεταβλητών που ξεκινούν από αριθμό. Επιπλέον, γίνεται έλεγχος ώστε η απόλυτη τιμή της σταθεράς να μην ξεπερνά το 32767, διαφορετικά εμφανίζεται κατάλληλο μήνυμα λάθους.

Από την **κατάσταση 3** και διαβάζοντας τον χαρακτήρα «\*» μεταβαίνει στην κατάσταση 5 και με τον χαρακτήρα «/» στην κατάσταση 4. Διαφορετικά μεταβαίνει σε τελική κατάσταση και επιστρέφει το χαρακτήρα «/».

Από την **κατάσταση 4** επιστρέφει στην αρχική όταν συναντήσει το χαρακτήρα αλλαγής γραμμής και τα σχόλια που είχαν αναγνωριστεί αγνοούνται. Εάν διαβαστεί το κενό (τέλος αρχείου) o lex() τερματίζει πρόωρα τη λειτουργία του. Διαβάζοντας το χαρακτήρα «/» μεταβαίνει στην κατάσταση 11 και από εκεί, εάν αναγνωριστεί εμφωλευμένο σχόλιο τυπώνεται μήνυμα λάθους και τερματίζει, αλλιώς επιστρέφει στην κατάσταση 4.

Από την **κατάσταση 5** μεταβαίνει στην κατάσταση 6 διαβάζοντας το χαρακτήρα «/», στην 7 αν διαβάσει «\*» ή τερματίζει πρόωρα αν διαβάσει το τέλος αρχείου. Σε οποιαδήποτε άλλη περίπτωση παραμένει στην ίδια κατάσταση.

Από την **κατάσταση 6** επιστρέφει στην κατάσταση 5 σε κάθε περίπτωση πλην του τέλους αρχείου και της περίπτωσης αναγνώρισης εμφωλευμένων σχολίων (διαβάζοντας τους χαρακτήρες «/» ή «\*»). Σε αυτές τις περιπτώσεις εμφανίζει κατάλληλο μήνυμα λάθους.

Από την **κατάσταση 7** επιστρέφει στην αρχική όταν συναντήσει το χαρακτήρα «/» και τα σχόλια που είχαν αναγνωριστεί αγνοούνται ή στην κατάσταση 5 για οποιοδήποτε άλλο χαρακτήρα.

Από **την κατάσταση 8** θα αναγνωριστούν τα εξής σύμβολα:

* «<>» , με τον χαρακτήρα «>»
* «<=», με τον χαρακτήρα «=»
* «<», ειδάλλως

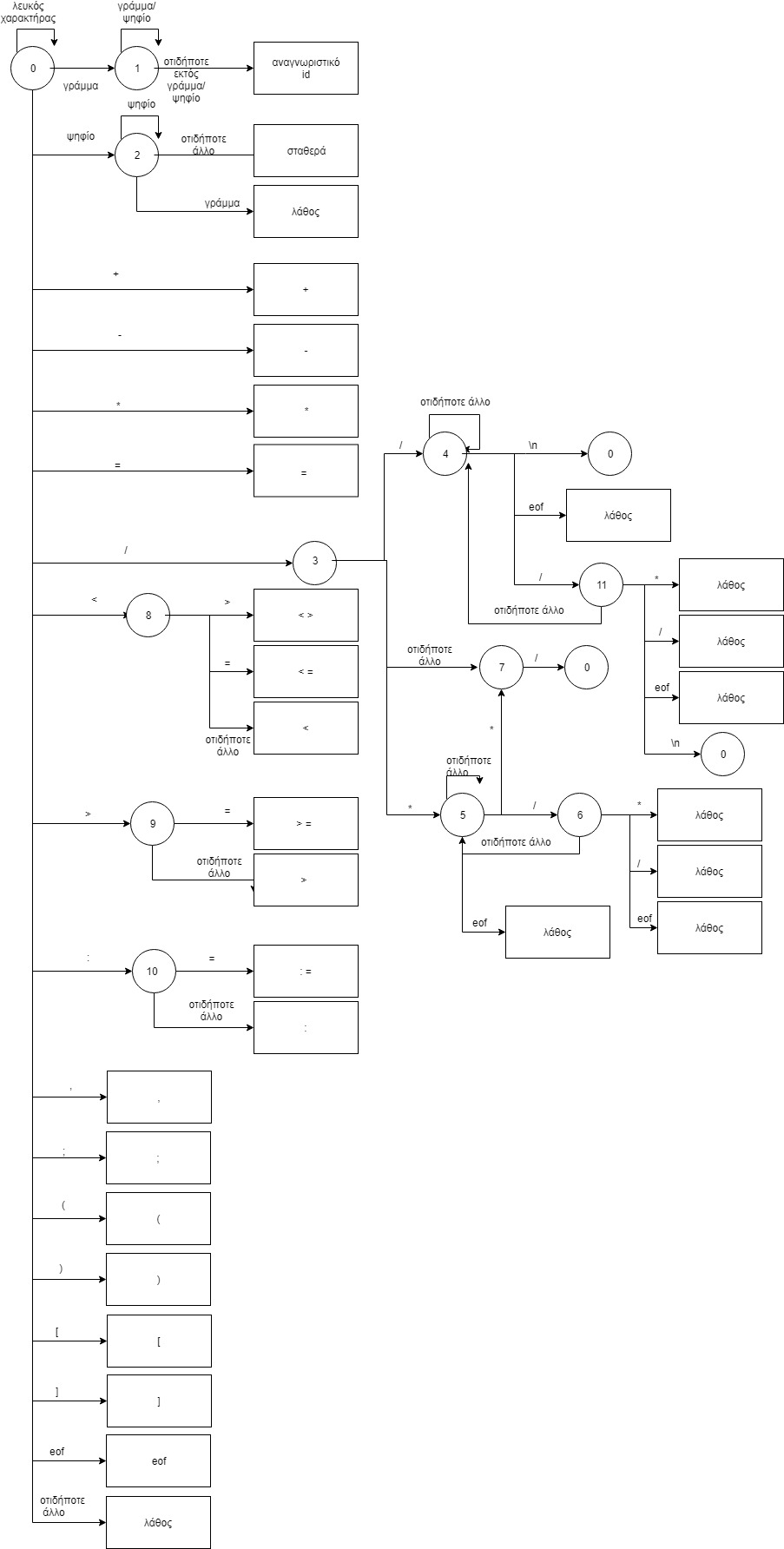
Από **την κατάσταση 9** θα αναγνωριστούν τα εξής σύμβολα:

* «>=», με τον χαρακτήρα «=»
* «>», ειδάλλως

Από **την κατάσταση 10** θα αναγνωριστούν τα εξής σύμβολα:

* «:=», με τον χαρακτήρα «=»
* «:», ειδάλλως

Για να γίνει σωστά η πρόβλεψη της λεκτικής μονάδας που θα επιστραφεί, στις καταστάσεις 1,2,3,8,9 και 10 αναιρούμε την ανάγνωση του τελευταίου χαρακτήρα που έχει διαβαστεί από τον λεκτικό αναλυτή στις περιπτώσεις που αυτός δεν εμπίπτει σε κάποια από τις προκαθορισμένες κατηγορίες που έχουμε εξετάσει παραπάνω. Εάν δεν ληφθεί υπόψη η παραπάνω περίπτωση, καταναλώνεται άσκοπα χαρακτήρας που μπορεί να είναι σημαντικός για τη μετέπειτα λειτουργία του lex. Παρακάτω παρατίθεται το αυτόματο καταστάσεων του λεκτικού αναλυτή σχηματικά.



**Συντακτικός αναλυτής**

Για κάθε κανόνα της γραμματικής έχει υλοποιηθεί η αντίστοιχη συνάρτηση. Σε κάθε συνάρτηση γίνεται έλεγχος για δεσμευμένες λέξεις / χαρακτήρες και σε περίπτωση μη τερματικών συμβόλων καλείται η αντίστοιχη συνάρτηση. Όταν συναντήσουμε τερματικό σύμβολο, τότε εάν ο λεκτικός αναλυτής επιστρέψει λεκτική μονάδα που αντιστοιχεί στο τερματικό σύμβολο το οποίο ελέγχουμε, συνεχίζει τη λειτουργία του. Στην αντίθετη περίπτωση, εμφανίζεται το κατάλληλο μήνυμα λάθους. Κάθε λεκτική μονάδα που εμφανίζεται στο συντακτικό αναλυτή, προκύπτει από την κλήση του lex().

Για κάθε κανόνα της γραμματικής ακολουθούνται οι εξής κανόνες :

* Το **αριστερό μέρος** κάθε κανόνα αποτελεί το όνομα του αντίστοιχου υποπρογράμματος.
* Γίνεται σύγκριση κάθε tokenID που επιστρέφει ο lex() με τη δεσμευμένη λέξη ή χαρακτήρα που αναμένει ο κανόνας.
* **Κλήση της κατάλληλης συνάρτησης** στις περιπτώσεις μη τερματικών συμβόλων.
* Στους κανόνες που εμφανίζεται *()\** είναι πιθανό το περιεχόμενο της παρένθεσης να καλείται από καμία έως πολλαπλές φορές. Αυτό υλοποιείται με τη χρήση της δομής while σε αυτές τις συναρτήσεις.
* Στις παρακάτω συναρτήσεις γίνεται «παραβίαση» της LL1 γραμματικής που υλοποιούμε : *optional-sign, idtail, term, expression, statement, subprograms*. Με τον όρο παραβίαση εννοούμε τον έλεγχο της συγκεκριμένης λεκτικής μονάδας που αναμένει ο κανόνας. Σε περίπτωση αποτυχίας ταυτοποίησης εμφανίζεται μήνυμα λάθους.
* Στους κανόνες που επιτρέπεται η εμφάνιση του κενού χαρακτήρα *ε*, δεν εμφανίζεται κάποιο μήνυμα λάθους.

Υποσημείωση: Στην περίπτωση πρόβλεψης της επόμενης συνάρτησης που αναμένεται να εκτελεστεί έχουν παραλειφθεί πλεονάζοντες έλεγχοι ορισμένων λεκτικών μονάδων, π.χ. η συνάρτηση statement κάνει έλεγχο για την ύπαρξη της μονάδας if και καλεί την αντίστοιχη συνάρτηση if\_stat. Στην if\_stat δεν πραγματοποιείται έλεγχος για την εμφάνιση του if, ούτε ορίζεται αντίστοιχο λάθος καθώς, για να κληθεί η if\_stat, σημαίνει ότι η λέξη if θα υπάρχει πάντα.