

2. Συντακτικός Αναλυτής

2.1 Γενική Ιδέα

Στο κεφάλαιο αυτό παρουσιάζουμε τον συντακτικό αναλυτή. Ο συντακτικός αναλυτής ελέγχει την ορθότητα του προγράμματος βάσει μιας προκαθορισμένης γραμματικής και καθοδηγεί την παραγωγή κώδικα στις επόμενες φάσεις. Ο στόχος του είναι να διασφαλίσει ότι το πρόγραμμα που έχει γραφεί στη γλώσσα CPY είναι συντακτικά σωστό, ακολουθώντας τους κανόνες και τις δομές της γραμματικής της γλώσσας. Ο συντακτικός αναλυτής χρησιμοποιεί τα tokens που έχουν παραχθεί από τον λεξικό αναλυτή και τα οργανώνει σε μια δομή που είναι εύκολα διαχειρίσιμη για περαιτέρω επεξεργασία και εκτέλεση.

Ο συντακτικός αναλυτής ακολουθεί μια αναλυτική προσέγγιση για να επαληθεύσει την κάθε συντακτική μονάδα του προγράμματος, από τις δηλώσεις και τις εντολές, μέχρι τις εκφράσεις και τις παραμέτρους συναρτήσεων. Σε περίπτωση που εντοπιστεί κάποιο συντακτικό λάθος, Σταματάει την ανάλυση του προγράμματος και αναφέρει στον προγραμματιστή το λάθος και το σημείο που υπήρξε αυτό.

2.2 Περιγραφή των κλάσεων

Η κλάση που χρησιμοποιείται για την υλοποίηση του συντακτικού αναλυτή είναι η `SyntaxAnalyzer` και είναι υπεύθυνη για την λήψη και ανάλυση των tokens από τον λεκτικό αναλυτή καθώς και για την ενσωμάτωση του ενδιάμεσου κώδικα και του πίνακα συμβόλων στην λειτουργία της.

Μέθοδοι Κλάσης:

1. `__init__(self, tokens)`

- **Περιγραφή:** Αρχικοποιεί την κλάση με τη λίστα των tokens και τη συμβολοπίνακα.
- **Λεπτομέρειες:**
 - Αποθηκεύει τα tokens.
 - Θέτει τον δείκτη token στο 0.

- Ορίζει το τρέχον token.
- Δημιουργεί έναν νέο συμβολοπίνακα.

2. nextToken(self)

- Περιγραφή: Μεταβαίνει στο επόμενο token στην λίστα των tokens.
- Λεπτομέρειες:
 - Αυξάνει τα tokens κατά ένα.
 - Εάν δεν υπάρχουν άλλα tokens ορίζει το τρέχον στο κενό.

3. tokenCheck(self,expectedToken)

- Περιγραφή: Ελέγχει αν το τρέχον token είναι ίδιο με το αναμενόμενο.
- Λεπτομέρειες:
 - Αν το τρέχον token είναι το επιθυμητό τότε καλεί την nextToken().
 - Αν δεν ταιριάζει σταματάει το πρόγραμμα και ρίχνει εξαίρεση συντακτικού λάθους.

4. startRule(self):

- Περιγραφή: Αρχίζει την ανάλυση με τον αρχικό κανόνα της γραμματικής.
- Λεπτομέρειες:
 - Ελέγχει αν το τρέχον token είναι ανάμεσα στα token εκκίνησης ('#int', 'def', '#def',) και καλεί τις αντίστοιχες μεθόδους ('declarations', 'def_function', 'def_main').
 - Επαναλαμβάνει αναδρομικά ώσπου να φτάσει στο τέλος του αρχείου ή να βρεθεί συντακτικό λάθος.

5. def_main(self):

- Περιγραφή: Ανάλυση και επεξεργασία της δήλωσης της κύριας συνάρτησης main στο πρόγραμμα.
- Λεπτομέρειες:
 - Ξεκινάει μια καινούρια Procedure την οποία την τοποθετεί στον πίνακα συμβόλων.
 - Εκκινεί ένα scope που θα έχει όλες τις απαραίτητες μεταβλητές.
 - Ελέγχει ένα πρώτα ορίζεται κάποια συνάρτηση ή νέα μεταβλητή μέσα της για να καλέσει τις κατάλληλες διαδικασίες.
 - Παράγει τις κατάλληλες τετράδες για τον ενδιάμεσο κώδικα.
 - Ενημερώνει τον πίνακα συμβόλων.
 - Καλεί την code_block για να υλοποιήσει τις υπόλοιπες λειτουργίες που πιθανών να υπάρχουν μέσα στην main.

6. def_function(self):

- Περιγραφή: Αναγνωρίζει και επεξεργάζεται την δήλωση βοηθητικών συναρτήσεων.

- Λεπτομέρειες:
 - Ελέγχει συντακτικά εάν υπάρχουν τα κατάλληλα tokens για την ροή του προγράμματος σύμφωνα με τον πίνακα συμβόλων.
 - Ενημερώνει για νέες εκχωρήσεις στον πίνακα συμβόλων και ορατότητας καθώς και για την έξοδο από αυτά.
 - Ελέγχει και επεξεργάζεται τις παραμέτρους της συνάρτησης και τις εκχωρεί στον πίνακα συμβόλων.
 - Ελέγχει και επεξεργάζεται πιθανές ορίσεις νέων μεταβλητών καθώς και τυχόν καθολικές μεταβλητές.
 - Δημιουργεί ένα χρειάζεται εκφωλιασμένες συναρτήσεις αναδρομικά.
 - Δημιουργεί νέες τετράδες είτε για εκκίνηση του block είτε για τερματισμό
 - Καλεί την `code_block` για την υλοποίηση της λογικής της βοηθητικής συνάρτησης.