ΜΕΤΑΦΡΑΣΤΕΣ

ακαδ.έτος 2019-2020

Διδάσκων: Γεώργιος Μανής

Αθανάσιος Παπαναστασίου 3057
Φαίδων Χρηστάκης 3110

Αναφορά προγραμματιστικής άσκησης

Αντικείμενο της προγραμματιστικής άσκησης είναι η υλοποίηση ενός μεταφραστή για την γλώσσα Minimal++ και την παραγωγή τελικού κώδικα για την αρχιτεκτονική MIPS.

Λεκτικός Αναλυτής

Σκοπός του λεκτικού αναλυτή είναι η απομόνωση και η επιστροφή των λεκτικών μονάδων του δοθέντος προγράμματος. Ο λεκτικός αναλυτής (60-103) αρχικά ελέγχει αν είναι η πρώτη φορά που καλείται. Αν ναι διαβάζει τον πρώτο χαρακτήρα του προγράμματος, αλλιώς επαναφέρει τον τελευταίο που διαβάστηκε. Στη συνέχεια ξεχωρίζουμε τις εξής περιπτώσεις, ο χαρακτήρας:

- Είναι ο κενός. Οπότε διαβάζει όλα τα κενά και συνεχίζει.
- Είναι αριθμός. Άρα διαβάζουμε και τους υπόλοιπους αριθμούς.
- Είναι γράμμα. Εδώ μπορούμε να διαβάσουμε είτε αριθμούς είτε γράμματα καθώς μια μεταβλητή μπορεί να περιέχει και τα δύο με την προϋπόθεση ότι ξεκινάει με γράμμα. Στο τέλος ελέγχει αν είναι δεσμευμένη λέξη και επιστρέφει τους αντίστοιχους κωδικούς.
- Είναι σύμβολο. Και πάλι διαβάζει τον επόμενο χαρακτήρα για την περίπτωση που έχουμε διπλό σύμβολο. Τέλος στην περίπτωση που έχουμε άνοιγμα σχολίων διαβάζονται όλοι οι χαρακτήρες/λεκτικές μονάδες έως το τέλος τους, ανάλογα την περίπτωση.

Συντακτικός Αναλυτής

Ο συντακτικός αναλυτής χρησιμοποιώντας τον λεκτικό αναλυτή καλείται να επιβεβαιώσει την γραμματική της γλώσσας. Ουσιαστικά η γραμματική της γλώσσας αποτυπώνεται με ένα σύνολο if-else (107-532) εντολών για κάθε έναν κανόνα της γραμματικής. Αποτελεί το πιο σημαντικό σημείο του μεταφραστή απ’ όπου και επιτελούνται οι υπόλοιπες λειτουργίας παραγωγή ενδιάμεσου κώδικα, διαχείριση πίνακα συμβόλων, παραγωγή τελικού και έλεγχος λεκτικών μονάδων. Παράλληλα δημιουργεί χρήσιμα μηνύματα λάθους με την χρήση της **error**(18).

Ενδιάμεσος Κώδικας

Η ύπαρξη του ενδιάμεσου απλουστεύει την πολυπλοκότητα της μετάφρασης αλλά προσθέτει και την δυνατότητα παραγωγής πολλαπλών τελικών γλωσσών (εμπρόσθιο-οπίσθιο τμήμα). Συγκεκριμένα παράγεται κατά την συντακτική ανάλυση με την χρήση των ζητούμενων συναρτήσεων στις γραμμές 534-565. Υλοποιείται με χρήση τετράδων και οι βοηθητικές συναρτήσεις του έχουν σκοπό την σωστή μετάφραση των δομών επιλογής και επανάληψης.

Πίνακας Συμβόλων

Ο πίνακας συμβόλων δημιουργείται με σκοπό την διαχείριση της μνήμης κατά την εκτέλεση του τελικού προγράμματος. Οι ζητούμενες συναρτήσεις διαχείρισης του, βρίσκονται στις γραμμές 567-605. Καλείται από τον συντακτικό αναλυτή στην περίπτωση κλήσης ενός υποπρογράμματος και στην δήλωση μεταβλητών. Συγκεκριμένα, τα scope αποθηκεύονται στον πίνακα **table**(567) και μέσα σε αυτά τα επιμέρους entities μέσω της **insertEntity**(589) και **insertArgument**(594). Τέλος κατά την διαγραφή ενός scope, **deleteScope**(578), καλείται η συνάρτηση παραγωγής τελικού κώδικα.

Τελικός Κώδικας

Με τις πληροφορίες από τα προηγούμενα δύο στάδια παράγεται ο τελικός κώδικας. Η βασική συνάρτηση **gnfncode**(649) χρησιμοποιεί τις υπόλοιπες συναρτήσεις στις γραμμές 607-749 για την παραγωγή του τελικού κώδικα στο πίνακα **final**(608). Συγκεκριμένα καλείται η **genMIPS**(663) για κάθε γραμμή ενδιάμεσου κώδικα, όπου και παράγει τον τελικό με την βοήθεια των ζητούμενων συναρτήσεων (613-647). Τέλος ο ενδιάμεσος κώδικας, ο πίνακας συμβόλων και ο τελικός γράφονται στα αντίστοιχα αρχεία στον φάκελο του αρχικού προγράμματος με την χρήση της **writeTo**(37).

Δοκιμαστικά Αρχεία

Για την επιβεβαίωση της ορθής λειτουργίας του μεταφραστή χρησιμοποιήσαμε τα αρχεία *final_1*, *final_2*, *forc*, *simple* και *while_test*. Τα πρώτα δύο είναι παραδείγματα από τις διαφάνειες του μαθήματος και παράγουν σωστά αποτελέσματα. Το *forc* επιβεβαιώνει την λειτουργία της σύνθετης εντολής **forcase.** Το *simple* το πέρασμα παραμέτρων με αναφορά, την λειτουργία **if** και την κλήση συνάρτησης. Επίσης το αρχείο *while_test* επιβεβαιώνει την λειτουργία της **while** όπου επιστρέφει το δοθέν στοιχείο της ακολουθίας fibonacci. Τέλος σε κάποιες περιπτώσεις αναδρομής ο μεταφραστής αποτυγχάνει στην διαχείριση της τιμής επιστροφής.

Για την εκτέλεση των παραγόμενων προγραμμάτων χρησιμοποιήθηκε το πρόγραμμα Mars από το μάθημα της Αρχιτεκτονικής Υπολογιστών.