ΜΕΤΑΦΡΑΣΤΕΣ
Project εξαμήνου

Τσάλεσης Ευάγγελος, ΑΜ: 1779

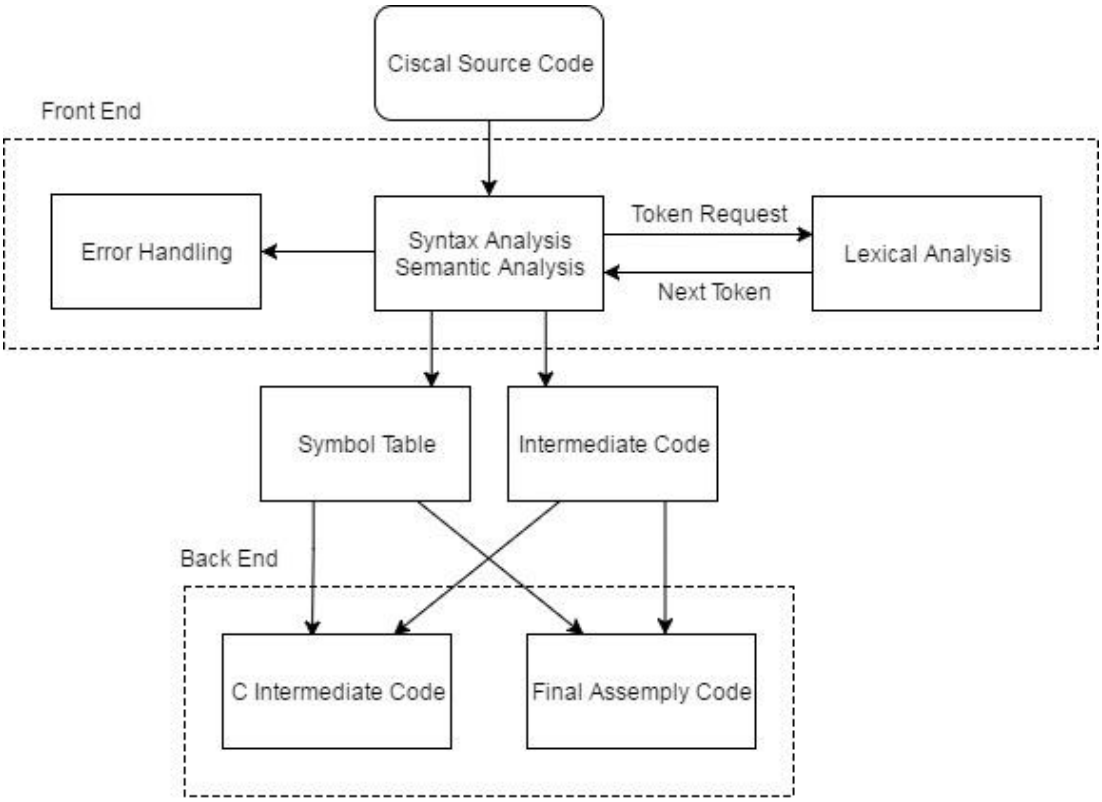
1. ΕΙΣΑΓΩΓΗ

Στόχος του project ήταν η υλοποίηση ενός ολοκληρωμένου μεταφραστή της προγραμματιστικής γλώσσας Ciscal.

Συγκεκριμένα, ο μεταφραστής δέχεται ένα αρχείο με τον πηγαίο κώδικα ενός προγράμματος ciscal και αφού εκτελέσει λεκτική, συντακτική και σημασιολογική ανάλυση παράγει σε πρώτο στάδιο τον ενδιάμεσο κώδικα (αρχείο εξόδου *.int) και τον πίνακα συμβόλων και σε δεύτερο στάδιο τον αντίστοιχο ενδιάμεσο κώδικα σε C (αρχείο εξόδου *.c) και τον τελικό κώδικα σε MIPS assembly (αρχείο εξόδου *.asm)

2. ΥΛΟΠΟΙΗΣΗ

2.1 Γενική Αρχιτεκτονική



2.2 Υλοποίηση Σταδίων

2.2.1 Λεκτική Ανάλυση

Ο λεκτικός αναλυτής υλοποιήθηκε ως ένα αυτόματο καταστάσεων. Ο συντακτικός αναλυτής ζητά από τον λεκτικό αναλυτή την επόμενη λεκτική μονάδα και ο λεκτικός αναλυτής επιστρέφει το token της αντίστοιχης μονάδας ή ένα error token αν οι τρέχοντες χαρακτήρες δεν αντιστοιχούν σε έγκυρη λεκτική μονάδα.

Ο λεκτικός αναλυτής επίσης ενημερώνει και τις μεταβλητές με την τρέχουσα λέξη ("**word**" global variable) και την τρέχουσα γραμμή του πηγαίου κώδικα ("**line_number**" global variable), που απαιτούνται για την παραγωγή του ενδιάμεσου κώδικα, του πίνακα συμβόλων και της διαχείρισης των σφαλμάτων.

Υπορουτίνες

int lex()	Λεκτικός αναλυτής

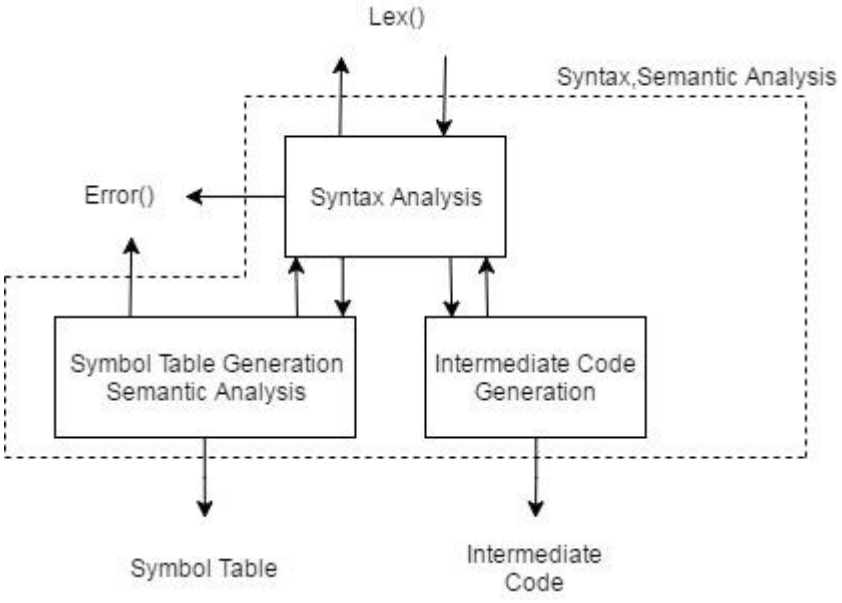
Κύριες δομές δεδομένων

stateMatrix	(2d πίνακας, int)	Κωδικοποιούνται οι μεταβάσεις καταστάσεων

2.2.2 Συντακτική και σημασιολογική ανάλυση

Η συντακτική και σημασιολογική ανάλυση, η παραγωγή του ενδιάμεσου κώδικα και του πίνακα συμβόλων πραγματοποιείται ταυτόχρονα σε ένα πέρασμα κατά την ανάγνωση του πηγαίου κώδικα ciscal.

Η ροή αυτού του σταδίου φαίνεται στο παρακάτω σχήμα.



2.2.2.1 Υπορουτίνες Συντακτικού Αναλυτή

Η δομή των υπορουτίνων του συντακτικού αναλυτή ακολουθεί την γραμματική της γλώσσας ciscal. Κατά την ανάγνωση του πηγαίου κώδικα καλούνται οι υπορουτίνες της παραγωγής του πίνακα συμβόλων (και του σημασιολογικού ελέγχου) και της παραγωγής ενδιάμεσου κώδικα. Σε περίπτωση λεκτικών, συντακτικών σφαλμάτων καλείται η υπορουτίνα διαχείρισης σφαλμάτων και ο μεταφραστής σταματά την εκτέλεση του.

2.2.2.2 Υπορουτίνες παραγωγής ενδιάμεσου κώδικα

Υπορουτίνες

int nextquad()	Επιστρέφει τον επόμενο αριθμό τετράδας
genquad(op, x, y, z)	Παράγει μία τετράδα
backpatch(list, z)	Εισάγει στις τετράδες της λίστας τον αριθμο z (4^o στοιχειο)
string newtemp()	Επιστρέφει μια νέα προσωρινή μεταβλητή T_n
printquads()	Εκτυπώνει τις τετράδες (βοηθητική για αποσφαλμάτωση)
intCreator()	Αποθηκεύει τις τετράδες σε αρχείο *.int

Κύριες δομές δεδομένων

quadList	(array, string)	Αποθηκεύονται οι τετράδες

2.2.2.3 Υπορουτίνες παραγωγής πίνακα συμβόλων, σημασιολογικού ελέγχου

Υπορουτίνες

newScope()	Δημιουργεί νέο scope
deleteScope()	Διαγράφει το τελευταίο scope Το αποθηκεύει στον μόνιμο πίνακα σύμβολων
insertEntity(name, symType)	Εισάγει το σύμβολο name στον πίνακα συμβόλων. Ελέγχει αν έχει ξαναδηλωθεί
findEntity(name,typeList)	Αναζητεί το σύμβολο name στον πίνακα συμβόλων. Ελέγχει αν χρησιμοποιείται εσφαλμένος τύπος.
printTable()	Εκτυπώνει τον τρέχοντα πίνακα συμβόλων (αποσφαλμάτωση)

Κύριες δομές δεδομένων

symbolTable	list of dictionaries, κάθε scope μια εγγραφή της λίστας	Ο προσωρινός πίνακας συμβόλων. Χρησιμοποιείται για την σημασιολογική ανάλυση
savedSymbolTable	dictionary of dictionaries, κάθε dict περιέχει τα σύμβολα ενός block	Ο τελικός πίνακας συμβόλων. Χρησιμοποιείται για την παραγωγή τελικού κώδικα

2.2.3 Παραγωγή ενδιάμεσου κώδικα σε C

Υπορουτίνες

CintCreator()	Παραγωγή και αποθήκευση ενδιάμεσου κώδικα C σε αρχείο *.c

2.2.4 Παραγωγή τελικού κώδικα

Υπορουτίνες

gnvlcode(v)	Παράγει κώδικα assembly μεταφοράς της διεύθυνσης της μη τοπικής μεταβλητής v στον καταχωρητή $t0
loadvr(v, r)	Παράγει κώδικα assembly φόρτωσης της τιμής της μεταβλητής v από την μνήμη στον καταχωρητή $tr
storerv(r, v)	Παράγει κώδικα assembly αποθήκευσης της τιμής του καταχωρητή $tr στην μνήμη της μεταβλητής v
FinalCreator()	Κύρια υπορουτίνα παραγωγής και αποθήκευσης του τελικού κώδικα στο αρχείο *.asm

Κύριες δομές δεδομένων

finalCode	array, string	Αποθηκεύει τον τελικό κώδικα. Κάθε στοιχειό και μια γραμμή του κώδικα

2.2.5 Διαχείριση σφαλμάτων

Υπορουτίνες

error(error_str)	Εκτυπώνει το μήνυμα σφάλματος και την γραμμή στην οποία βρίσκεται και τερματίζει τον μεταφραστή

3. ΈΛΕΓΧΟΣ ΚΑΛΗΣ ΛΕΙΤΟΥΡΓΙΑΣ

Για τον έλεγχο της λειτουργίας του μεταφραστή χρησιμοποιήθηκαν 2 δοκιμαστικά προγράμματα ciscal:

test_form.ci: Περιέχει πολλές εντολές της γλώσσας ciscal, πολλες μεταβλητές, υπορουτίνες, τύπους σχολίων και συνδυασμούς τους ώστε να ελεγχθει η λειτουργία του λεκτικού, συντακτικού και σημασιολογικού αναλυτή. Δεν παράγει κάποιο χρήσιμο αποτέλεσμα.

test_function.ci: Περιέχει ένα απλό πρόγραμμα με πέρασμα μεταβλητών και των δύο τύπων ώστε να ελεγχθεί η λειτουργικότητα του τελικού κώδικα. Παράγει συγκεκριμένη ακολουθία αριθμών («*123421567891020*»)

Ο τελικός κώδικας ελέγχθηκε στο προσομοιωτή MIPS assembly MARS 4.5

Η εκτέλεση του μεταφραστή γίνεται με την εντολή:

python3 compiler.py <filename>