Πανεπιστημιο Ιωαννινων
Τμήμα Πληροφορικής

Μεταφραστές
Ακαδημαϊκό Έτος 1999-2000

Φάση 3: Παραγωγή Ενδιάμεσου Κώδικα

Ημερομηνία Παράδοσης: 28 Μαίου
Ποσοστό: 60%

Επεκτείνετε το συντακτικό σας αναλυτή ώστε να πραγματοποιεί σημασιολογικό έλεγχο και να παράγει ενδιάμεσο κώδικα. Η έξοδος του προγράμματός σας στην περίπτωση που το πρόγραμμα εισόδου είναι συντακτικά και σημασιολογικά σωστό πρέπει να είναι ένα πρόγραμμα (λίστα από εντολές) τριών διευθύνσεων. Στην περίπτωση που το πρόγραμμα εισόδου δεν είναι σωστό, η έξοδος του προγράμματός σας πρέπει να αναφέρει ποια λάθη εντοπίστηκαν και σε ποιές γραμμές του προγράμματος εισόδου.

Απλοποιήσεις. Για ευκολία, δε θα υλοποιήστε συναρτήσεις/διαδικασίες. Θεωρείστε ότι υπάρχει μόνο μία διαδικασία, η main.

Πίνακας Συμβόλων. Σε αυτή τη φάση, θα χρειαστεί να χρησιμοποιείστε έναν πίνακα συμβόλων. Η πληροφορία που συνήθως αποθηκεύεται για τις μεταβλητές απλού τύπου στον πίνακα συμβόλων περιλαμβάνει το όνομα, τον τύπο και τη θέση μνήμης. Στην περίπτωση πινάκων (array), η πληροφορία περιλαμβάνει επιπρόσθετα στοιχεία όπως τον αριθμό των στοιχείων ανά διάσταση. Θεωρείστε ότι οι ακέραιες μεταβλητές καταλαμβάνουν $w = 4$ θέσεις μνήμης, και οι μεταβλητές τύπου χαρακτήρα $w = 2$ θέσεις μνήμης.

Πίνακες. Θεωρείστε ότι οι πίνακες αποθηκεύονται ανά στήλη (όπως στην Fortran). Προσοχή, οι τύποι στις σημειώσεις αναφέρονται σε αποθήκευση ανά γραμμή.

Σημασιολογικός Έλεγχος. Παραδείγματα σημασιολογικών λαθών που μπορούν να εντοπιστούν σε στατικό χρόνο (κατά τη μετάφραση) είναι:

- συμβατότητα τελεστών και τελεσταίων στις πράξεις,
- συμβατότητα τύπων του αριστερού και δεξιού μέρους στις εντολές ανάθεσης,
- ορισμός των μεταβλητών πριν τη χρήση τους.

Ενδιάμεση Αναπαράσταση. Η ενδιάμεση αναπαράσταση που θα χρησιμοποιείστε είναι εντολές 3-διευθύνσεων με μορφή τετράδων.

Οι εντολές (op) που υποστηρίζονται και χρησιμοποιούν και τα τρία πεδία είναι:

- (πρόσθεση): ADD arg_1 arg_2 arg_3

- (αφαίρεση): SUB arg_1 arg_2 arg_3
- (πολλαπλασιασμός): MULT arg_1 arg_2 arg_3
- (διαίρεση): DIV arg_1 arg_2 arg_3
- (ακέραιο υπόλοιπο): MOD arg_1 arg_2 arg_3

και η σημασία τους είναι $arg_3 := arg_1\ op\ arg_2$

Επίσης, υποστηρίζονται και οι παρακάτων εντολές:

- UMINUS arg_1 arg_2 _ (που σημαίνει arg_1 := - arg_2)
- ASSIGN arg_1 arg_2 _ (που σημαίνει $arg_1 := arg_2$)
- GOTO _ arg_2 _ (unconditional jump στην τετράδα με ετικέττα arg_2)
- PGOTO arg_1 arg_2 _ (Jump (με συνθήκη) στην τετράδα με ετικέττα arg_2 αν $arg_1 > 0$)
- NGOTO arg_1 arg_2 _ (Jump (με συνθήκη) στην τετράδα με ετικέττα arg_2 αν $arg_1 \leq 0$)
- INDEX1 arg_1 arg_2 arg_3 (αναθέτει στο arg_1 την τιμή (περιεχόμενο) της θέσης μνήμης που βρίσκεται arg_3 θέσεις μνήμης από το arg_2)
- INDEX2 arg_1 arg_2 arg_3 (θέτει το περιεχόμενο της θέσης μνήμης που βρίσκεται arg_2 θέσεις μνήμης μετά από το arg_1 ίσο με arg_3)
- PRGEND (τέλος του προγράμματος)

Έξοδος του Προγράμματος. Η έξοδος του προγράμματος σας πρέπει να είναι μία ακολουθία από εντολές τριών διευθύνσεων (μία ανά γραμμή). Οι εντολές θα τυπώνονται ως εξής:

Αύξοντας_Αριθμός_Εντολής: Κωδικός_Εντολής arg_1 arg_2 arg_3

Αν κάποιο από τα arg_i δεν ορίζεται για κάποια εντολή θα αφήνετε το αντίστοιχο πεδίο κενό ή θα γράφεται _.

Στην περίπτωση που το arg_i είναι:

- μεταβλητή: θα τυπώνετε ως arg_i ένα ζεύγος (όνομα, θέση μνήμης) όπου η θέση μνήμης είναι η απόσταση (offset) από την αρχική θέση αποθήκευσης των μεταβλητών,
- σταθερά: θα τυπώνετε την τιμή της,
- ετικέττα: θα τυπώνετε τον αύξοντα αριθμό της εντολής.