

Στόχος

Στόχος του κεφαλαίου αυτού είναι να μάθετε το είδος και τη δομή των πληροφοριών που καταγράφονται σαν περιεχόμενα του πίνακα συμβόλων, τους διάφορους τρόπους οργάνωσης του πίνακα και τρόπους προσπέλασης των περιεχομένων του.

Προσδοκώμενα αποτελέσματα

Όταν θα έχετε μελετήσει το κεφάλαιο αυτό θα μπορείτε να

- Περιγράψετε το είδος των πληροφοριών που καταγράφονται στον πίνακα συμβόλων,
- Περιγράψετε τους διαφορετικούς τρόπους δόμησης του πίνακα συμβόλων,
- Εξηγήσετε τι είναι ο κερματισμός διευθύνσεων και πώς χρησιμοποιείται,
- Περιγράψετε διάφορες συναρτήσεις κερματισμού,
- Εξηγήσετε πώς υλοποιείται ο αλυσιδωτός κερματισμός διευθύνσεων,
- Περιγράψετε πώς υλοποιούνται οι δενδρικοί πίνακες,
- Περιγράψετε τους πίνακες συμβόλων για γλώσσες προγραμματισμού με δομές block,
- Γράψετε ρουτίνες δράσης οι οποίες υλοποιούν το άνοιγμα και το κλείσιμο των blocks,
- Γράψετε ρουτίνες για ανίχνευση και εισαγωγή ονομάτων στον πίνακα συμβόλων.

Έννοιες κλειδιά

Περιγραφέας, κερματισμός διευθύνσεων, γραμμικός κερματισμός, αλυσιδωτός κερματισμός, συναρτήσεις κερματισμού, block list.

Εισαγωγικές Παρατηρήσεις

Ο έλεγχος της σημασιολογικής ορθότητας και η δημιουργία κώδικα χρειάζεται να έχει γνώση των ιδιοτήτων (attributes) των ονομάτων (identifiers) τα οποία εμφανίζονται στο πρόγραμμα. Οι ιδιότητες αυτές βρίσκονται στις δηλωτικές εντολές και στον τρόπο με τον οποίο τα ονόματα χρησιμοποιούνται μέσα στο πρόγραμμα. Συλλέγονται δε από τον Μεταγλωττιστή και καταχωρούνται σε μια δομή η οποία ονομάζεται **πίνακας συμβόλων** (symbol table).

Ο πίνακας συμβόλων, όπως αναφέραμε και στο κεφάλαιο 1, είναι απαραίτητο εργαλείο για όλες τις φάσεις της διαδικασίας μεταγλώττισης. Πέρα όμως από αυτό είναι και απολύτως απαραίτητος για ορισμένες περιπτώσεις όπως στους διερμηνευτές και τους εκσφαλματωτές.

Στο κεφάλαιο αυτό θα εξετάσουμε τρόπους οργάνωσης και χρήσεις του πίνακα συμβόλων. Στην ενότητα 5.2 θα εξετάσουμε την μορφή και το είδος των περιεχομένων του πίνακα συμβόλων, ενώ στην 5.3 θα μελετήσουμε απλές μορφές οργάνωσης για ταξινομημένους και μη ταξινομημένους πίνακες. Στην 5.4 θα μιλήσουμε για κερματισμό διευθύνσεων, σαν ένα διαφορετικό τρόπο προσπέλασης των στοιχείων του πίνακα, και επανακερματισμό διευθύνσεων. Ακόμη θα μελετήσουμε την μέθοδο του αλυσιδωτού κερματισμού και διάφορες μορφές συναρτήσεων κερματισμού. Στην ενότητα 5.5 θα δούμε σύντομα τους δενδρικούς πίνακες. Τέλος στην ενότητα 5.6 θα μελετήσουμε την οργάνωση και προσπέλαση πινάκων για γλώσσες με δομές block.

ΕΝΟΤΗΤΑ 5.1 ΕΙΣΑΓΩΓΗ

Ο όρος "πίνακας" όπως αναφέρθηκε στις εισαγωγικές παρατηρήσεις αλλά και στο κεφάλαιο 1, δεν είναι απόλυτα σωστός γιατί όπως θα δούμε στο κεφάλαιο αυτό η πληροφορία που περιέχει μπορεί να αποθηκευθεί και σε μια δομή λίστας, ένα δένδρο, ή κάποια άλλη δομή δεδομένων.

Κάθε στοιχείο του πίνακα είναι ένα ζεύγος της μορφής <όνομα, περιγραφέας> (<name, descriptor>). Κάθε φορά που συναντάται κάποιο όνομα, ο πίνακας συμβόλων εξετάζεται για να διαπιστωθεί κατά πόσο το όνομα έχει ξανασυναντηθεί ή όχι. Εφόσον δεν έχει ξανασυναντηθεί εισάγεται στον πίνακα. Πληροφορίες για τις ιδιότητες του ονόματος εισάγονται στον πίνακα κατά την λεκτική και την συντακτική ανάλυση, και χρησιμοποιούνται σε διάφορα στάδια της διαδικασίας της μεταγλώττισης. Η πληροφορία (ιδιότητες) μέσα στον περιγραφέα (ή ακριβέστερα στην περιοχή που "δείχνεται" από τον περιγραφέα), χρησιμοποιείται, για παράδειγμα, κατά την δημιουργία του τελικού κώδικα για να μας πληροφορήσει για το πόσο και τι είδους χώρος χρειάζεται να δοθεί στο όνομα σαν χώρος εκτέλεσης (run-time storage).

Με δεδομένο ότι για κάθε εμφάνιση ενός ονόματος στο πρόγραμμα χρειάζεται και μια ανίχνευση του πίνακα, είναι φανερό ότι η ανίχνευση αυτή πρέπει να γίνεται όσο το δυνατόν πιο γρήγορα αλλιώς οι συνέπειες στην ταχύτητα

μετάφρασης του Μεταγλωττιστή είναι σημαντικές. Γι αυτό το λόγο θα εξετάσουμε στο κεφάλαιο αυτό τρόπους και μεθοδολογίες προσπέλασης του πίνακα. Από τους τρόπους προσπέλασης που θα δούμε βγαίνει και η συνταγή για την επιλογή. Αν ενδιαφερόμαστε μόνο για ταχύτητα χρησιμοποιούμε την μέθοδο του κερματισμού των διευθύνσεων. Αν μας ενδιαφέρει και η ευκολία κατασκευής του πίνακα τότε χρησιμοποιούμε την μέθοδο του κερματισμού με αλυσίδες. Αν μας ενδιαφέρει μόνο η εύκολη κατασκευή και καθόλου η ταχύτητα τότε μια γραμμική λίστα είναι η λύση. Δενδρικές μορφές του πίνακα αποτελούν μια ενδιάμεση λύση.

Βασικά στοιχεία στον σχεδιασμό του πίνακα συμβόλων είναι η μορφή των στοιχείων του και η μέθοδος προσπέλασής του. Ακόμα η δομή της γλώσσας την οποία μεταφράζει ο Μεταγλωττιστής-γλώσσα με blocks ή όχι- είναι στοιχείο που πρέπει να ληφθεί υπ' όψη στον σχεδιασμό. Γλώσσες με δομές block επιτρέπουν την χρήση του ίδιου ονόματος περισσότερες από μια φορές μέσα στον πίνακα. Σε Μεταγλωττιστές για τέτοιες γλώσσες το κάτω μέρος του πίνακα χρησιμοποιείται σαν στοίβα, όπως θα δούμε στην ενότητα 5.6.

ΕΝΟΤΗΤΑ 5.2 ΠΙΝΑΚΕΣ ΣΥΜΒΟΛΩΝ: ΠΕΡΙΕΧΟΜΕΝΑ

Είδαμε ότι τα στοιχεία του πίνακα αποτελούνται από ζεύγη <όνομα, περιγραφέας>. Το "όνομα" είναι το σύνολο των χαρακτήρων του ονόματος (lexeme) και χρησιμοποιείται από τον λεκτικό αναλυτή για να εντοπίζει τις εμφανίσεις του ονόματος μέσα στο πρόγραμμα. Ο περιγραφέας περιέχει πληροφορίες για το "αντικείμενο" για το οποίο έχει δηλωθεί το όνομα.

Είναι δυνατό μέσα σε ένα Μεταγλωττιστή να χρησιμοποιούνται διαφορετικοί πίνακες για διαφορετικά αντικείμενα όπως ονόματα μεταβλητών, επιγραφών (labels), procedures, σταθερές, πεδία δομών (records), και άλλους τύπους ονομάτων. Οι πληροφορίες του περιγραφέα είναι δυνατόν να διαφέρουν σε πλήθος κατά πολύ ανάλογα με το αντικείμενο, και αυτό να επιβάλλει χωριστούς πίνακες. Πάντως ένας πίνακας μπορεί να είναι αρκετός, ιδίως όταν η μορφή των στοιχείων του δεν ποικίλει. Αν η γλώσσα δεν αποκλείει την χρήση λέξεων κλειδιών όπως begin και goto, τότε είναι απαραίτητο, οι λέξεις κλειδιά να εισάγονται στον πίνακα από τον Μεταγλωττιστή εξ' αρχής, κάνοντας μάλιστα χρήση της ίδιας συνάρτησης κερματισμού που θα χρησιμοποιηθεί και για τα υπόλοιπα ονόματα. Επιπλέον πρέπει να συνοδεύονται και από σχετική

πληροφορία που αναφέρει ότι πρόκειται για λέξεις κλειδιά και όχι κοινά ονόματα μεταβλητών.

Όσον αφορά τη δομή του πίνακα πρέπει να είναι τέτοια που να μας επιτρέπει να προσδιορίσουμε κατά πόσο ένα όνομα βρίσκεται στον πίνακα, να εισάγουμε ένα καινούριο όνομα, να προσπελάσουμε μέσω του περιγραφέα τις ιδιότητες του ονόματος, να προσθέσουμε καινούριες ιδιότητες και να περιγράψουμε μια ιδιότητα, ένα όνομα ή μια ομάδα από ονόματα. Οι πληροφορίες που πρέπει να καταγράφονται μέσω του περιγραφέα για κάθε όνομα περιλαμβάνουν οντότητες όπως οι παρακάτω:

Παράδειγμα 1 / Κεφ. 5

-Για μεταβλητές:

- τύπος (real, boolean, label, κ.λ.π.)
- μορφή (απλή μεταβλητή, array, record, κ.λ.π.)
- ακρίβεια, μήκος
- διεύθυνση κατά τον χρόνο εκτέλεσης (run time address)
- αν πρόκειται για array, πόσες διαστάσεις, άνω και κάτω όριο διάστασης, τα όρια είναι σταθερές, εκφράσεις ή τυπικές παράμετροι μιας procedure (subroutine);
- είναι τυπική παράμετρος ή όχι; αν ναι τότε πρέπει να αναφέρεται σε αυτή, κάποια procedure.
- βρίσκεται σε εντολή COMMON ή EQUIVALENCE (FORTRAN); αν είναι, πρέπει να συνδέεται με τα αντίστοιχα ονόματα.
- έχει συναντηθεί η δηλωτική εντολή της μεταβλητής;

Παράδειγμα 2 / Κεφ. 5

-Για procedures:

- είναι external στο πρόγραμμα;
- είναι συναρτηση; και ποίος είναι ο τύπος της;
- έχει συναντηθεί η δηλωτική της εντολή;
- είναι αναδρομική;
- ποιες είναι οι τυπικές της παράμετροι; οι "περιγραφείς" των παραμέτρων αυτών πρέπει να συνδέονται με το όνομα της συνάρτησης για λόγους σύγκρισης με τις πραγματικές παραμέτρους.

Οι πληροφορίες αυτές εισάγονται στον (ή στους) πίνακες από τον Μεταγλωττιστή τόσο κατά την εξέταση των δηλωτικών εντολών όσο και κατά την εξέταση άλλων εντολών (π.χ. label: statement). Στο κεφάλαιο της λεκτικής ανάλυσης αναφέρθηκε ότι ο Λεκτικός Αναλυτής όταν αναγνωρίζει ένα token τύπου identifier, το εισάγει στον πίνακα και επιστρέφει στον συντακτικό Αναλυτή την ένδειξη id και την θέση του στον πίνακα συμβόλων (δείκτης). Σε γλώσσες που ένα και το αυτό όνομα χρησιμοποιείται σε διάφορα μέρη με διάφορους

τύπους (π.χ. real, logical), ο λεκτικός αναλυτής δεν μπορεί παρά να επιστρέφει μόνο το όνομα (τον ίδιο τον identifier) και όχι δείκτη στον πίνακα συμβόλων.

Άσκηση αυτοαξιολόγησης 1 / Κεφ. 5

Χωρίς να συμβουλευθείς το αμέσως προηγούμενο κείμενο,

1. γράψε τέσσερις τουλάχιστον διαφορετικές πληροφορίες που πρέπει να καταγράφονται μέσω του περιγραφέα για κάθε όνομα απλής μεταβλητής.
2. γράψε τέσσερις τουλάχιστον διαφορετικές πληροφορίες που πρέπει να καταγράφονται μέσω του περιγραφέα για κάθε όνομα procedure.

ΕΝΟΤΗΤΑ 5.3 ΠΙΝΑΚΕΣ ΣΥΜΒΟΛΩΝ: ΟΡΓΑΝΩΣΗ

Στην ενότητα αυτή συζητάμε τους τρόπους οργάνωσης κατασκευής και ανίχνευσης των πινάκων συμβόλων. Έχουμε ήδη αναφέρει ότι κάθε στοιχείο του πίνακα είναι ένα ζεύγος της μορφής <όνομα, περιγραφέας>. Μπορούμε επομένως να θεωρήσουμε ότι οι πίνακες συμβόλων έχουν την γενική μορφή του Σχήματος 5.1.

Το "όνομα" είναι η συμβολοσειρά των χαρακτήρων (lexeme) που απαρτίζουν ένα identifier και ο "Περιγραφέας" είναι οι ιδιότητες του ή ένας δείκτης σε μια άλλη δομή ή πίνακα που περιέχει τις ιδιότητές του.

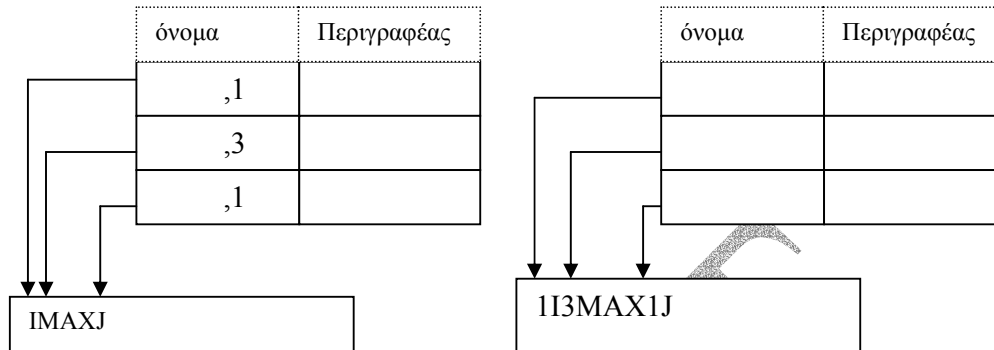
	Όνομα	Περιγραφέας
1ο στοιχείο		
2ο στοιχείο		
N στοιχείο		

Σχήμα 5.1 Μια γενική δομή του πίνακα συμβόλων

Μια και ο αριθμός των χαρακτήρων στους identifiers είναι μεταβλητός, συχνά αποθηκεύεται ένας δείκτης στην θέση του ονόματος. Με αυτό τον τρόπο το απαιτούμενο μέγεθος του ονόματος στον πίνακα παραμένει σταθερό. Ο αριθμός των χαρακτήρων σε κάθε identifier μπορεί να αποθηκεύεται σαν μέρος του ονόματος ή στην λίστα των identifiers μπροστά από τον identifier. Δείτε για παράδειγμα το σχήμα 5.2.

Όταν αρχίζει η μετάφραση ενός προγράμματός σας ο πίνακας συμβόλων είναι κενός ή πιθανόν να περιέχει μόνο τις αποκλειστικές λέξεις. Καθώς προχωράει η συμβολομετάφραση προστίθεται και ένα στοιχείο για κάθε καινούριο identifier, ο

πίνακας όμως ανιχνεύεται για κάθε εμφάνιση ενός identifier μέσα στο πρόγραμμα σας. Μια και η διαδικασία αυτή είναι χρονοβόρα ενδιαφερόμαστε για οργανώσεις πινάκων που διευκολύνουν την ανίχνευση, δηλαδή μειώνουν τον χρόνο ανίχνευσης.



Σχήμα 5.2 Δύο τρόποι αποθήκευσης των ονομάτων μεταβλητών.

Ο αναμενόμενος αριθμός συγκρίσεων E των ονομάτων, που χρειάζονται για να εντοπισθεί ένας identifier, εξαρτάται από τον **συντελεστή φόρτου** (load factor) If , του πίνακα σε κάποια στιγμή.

$$If = n/N$$

όπου n = το πλήθος των στοιχείων του πίνακα και N το μέγεθος του.

5.3.1 ΤΑΞΙΝΟΜΗΜΕΝΟΙ ΚΑΙ ΜΗ ΤΑΞΙΝΟΜΗΜΕΝΟΙ ΠΙΝΑΚΕΣ

Ο ευκολότερος τρόπος οργάνωσης ενός πίνακα είναι η πρόσθεση στοιχείων με τη σειρά που καταφθάνουν χωρίς καμιά προσπάθεια ταξινόμησης. Μια ανίχνευση χρειάζεται μια σύγκριση με κάθε στοιχείο του πίνακα μέχρις ότου ευρεθεί ταυτότητα. Για ένα πίνακα n στοιχείων χρειάζονται κατά μέσο όρο $n/2$ συγκρίσεις. Για μεγάλο n ($n > 20$), η μέθοδος είναι ανεπαρκής. Η ανίχνευση προφανώς βελτιώνεται αν τα στοιχεία του πίνακα είναι ταξινομημένα.

Στην περίπτωση μας όπου τα στοιχεία είναι ακολουθίες χαρακτήρων ο πιο φυσικός τρόπος ταξινόμησης είναι συνάρτηση της εσωτερικής αναπαράστασής τους. Η ταξινόμηση αυτή είναι συνήθως ίδια με την αλφαβητική κατάταξη.

Μια ικανοποιητική μέθοδος ανίχνευσης μιας ταξινομημένης λίστας n στοιχείων είναι η δυαδική ή λογαριθμική ανίχνευση. Το σύμβολο (identifier) S προς ανίχνευση συγκρίνεται με το μεσαίο στοιχείο στην θέση $(n+1)/2$. Αν το

στοιχείο αυτό δεν είναι το S, πρέπει να ψάξουμε είτε στο σύνολο των στοιχείων που βρίσκονται στις θέσεις 1 έως $(n+1)/2-1$ είτε στα $(n+1)/2+1$ έως n, ανάλογα με το αν το S είναι μικρότερο ή μεγαλύτερο από τα συγκρινόμενα στοιχεία. Η διαδικασία επαναλαμβάνεται αναδρομικά. Μια και σε κάθε στάδιο ο αριθμός των στοιχείων που μπορεί να περιέχουν το S υποδιπλασιάζεται, ο μέγιστος αριθμός συγκρίσεων είναι:

$$1 + \lceil \log_2 n \rceil$$

Αν $n=2$ χρειαζόμαστε το πολύ 2 συγκρίσεις, αν $n=4$, 3 αν $n=8$, 4. Αν $n=128$ η μέθοδος χρειάζεται το πολύ 8 συγκρίσεις ενώ η προηγούμενη μέθοδος χρειάζεται κατά μέσο όρο $64 \lfloor n/2 \rfloor$ συγκρίσεις.

ΕΝΟΤΗΤΑ 5.4 ΚΕΡΜΑΤΙΣΜΟΣ ΔΙΕΥΘΥΝΣΕΩΝ

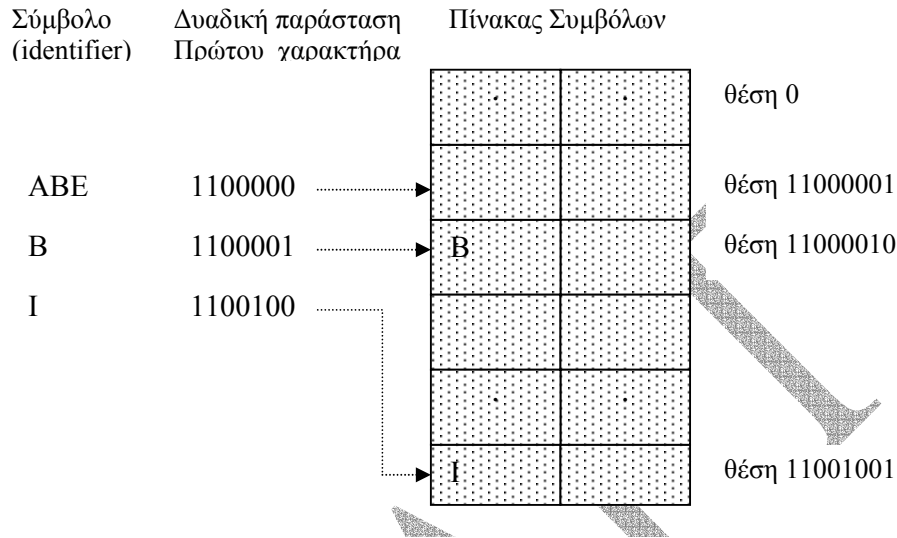
Ο κερματισμός διευθύνσεων είναι μια μέθοδος μετατροπής συμβόλων σε δείκτες στα στοιχεία του πίνακα (τα στοιχεία είναι αριθμημένα $0,1,2,\dots,N-1$). Με τον όρο σύμβολο εδώ εννοούμε την συμβολοσειρά των χαρακτήρων του ονόματος μιας μεταβλητής, μιας επιγραφής ή άλλης ονοματισμένης προγραμματιστικής οντότητας. Ο δείκτης προσδιορίζεται με "κερματισμό" (hashing) του συμβόλου κάνοντας κάποια απλή αριθμητική ή λογική πράξη με το σύμβολο και ενδεχομένως το μήκος του. Μια απλή συνάρτηση κερματισμού (hash function) είναι η εσωτερική αναπαράσταση του πρώτου χαρακτήρα του συμβόλου.

Παράδειγμα 3 / Κεφ. 5

Έτσι, αν η δυαδική αναπαράσταση του χαρακτήρα A είναι 11000001, το σύμβολο ABE κάνει κερματισμό στην διεύθυνση 11000001 (c1). Η πρώτη διεύθυνση του πίνακα που κοιτάζουμε κατά την ανίχνευση για το στοιχείο με περιεχόμενο ABE είναι η 11000001. Το Σχήμα 5.3 επεξηγεί αυτό για τα ονόματα (identifiers) ABE, B και I.

Εφ' όσον δύο ονόματα δεν κάνουν κερματισμό στην ίδια διεύθυνση (στον ίδιο δείκτη) το κόστος της ανίχνευσης είναι μόνο το κόστος της πράξης του κερματισμού. Προβλήματα δημιουργούνται όταν δύο σύμβολα κερματίζουν στον ίδιο δείκτη. Τότε έχουμε *σύγκρουση*. Προφανώς μόνο ένα σύμβολο μπορεί να τοποθετηθεί σε κάθε θέση του πίνακα, επομένως πρέπει να βρούμε κάποια άλλη θέση για το δεύτερο σύμβολο. Μια καλή συνάρτηση κερματισμού (hash function)

πρέπει να απλώνει ομοιόμορφα τις υπολογιζόμενες διευθύνσεις κατά μήκος του πίνακα ώστε οι συγκρούσεις να είναι σπάνιες. Η συνάρτηση κερματισμού που περιγράφηκε παραπάνω, είναι προφανώς κακή, διότι όλα τα σύμβολα που αρχίζουν με το ίδιο γράμμα κάνουν κερματισμό στην ίδια διεύθυνση.



Σχήμα 5.3 Κερματισμός διευθύνσεων με βάση τον πρώτο χαρακτήρα του ονόματος.

5.4.1 ΕΠΑΝΑΚΕΡΜΑΤΙΣΜΟΣ ΔΙΕΥΘΥΝΣΕΩΝ (Rehashing)

Ας υποθέσουμε τώρα ότι ένα σύμβολο S κερματίζεται στη θέση h . Αν στη θέση h υπάρχει άλλο σύμβολο, τότε έχουμε σύγκρουση και ψάχνουμε τη θέση $h+P_1 \pmod{N}$ για κάποιο ακέραιο P_1 .

Αν και πάλι υπάρχει σύγκρουση συγκρίνουμε με το $h+P_2 \pmod{N}$ για κάποιο ακέραιο P_2 . Αυτό συνεχίζεται μέχρι που κάποια θέση $h+P_i \pmod{N}$ είναι κενή, περιέχει το S , ή είναι και πάλι η θέση $h(P_i=0)$. Στην τελευταία περίπτωση σταματάμε την διαδικασία γιατί ο πίνακας είναι γεμάτος.

Έτσι, αν i συγκρούσεις έχουν συμβεί, η $i+1$ σύγκριση θα είναι στην θέση $h_i=h+P_i \pmod{N}$. Οι τιμές P_i θα πρέπει να διαλέγονται έτσι ώστε, ο αναμενόμενος αριθμός συγκρίσεων E (τον αναφέραμε στην ενότητα 5.3), είναι μικρός και να καλύπτονται όσον το δυνατόν περισσότερα στοιχεία του πίνακα. Στην ιδανική περίπτωση τα P_i θα πρέπει να καλύπτουν του ακέραιους $0,1,2,\dots, N-1$. Διάφορες μέθοδοι προσδιορισμού του P_i θα συζητηθούν παρακάτω.

Να θυμίσουμε εδώ ότι ο πίνακας αρχικά περιέχει τιμές οι οποίες δεν μπορούν να δηλώνουν κάποιο σύμβολο, ώστε οι κενές θέσεις να διακρίνονται από τις γεμάτες.

Ο γραμμικός επανακερματισμός (Linear Rehash) είναι η παλιότερη μέθοδος επανακερματισμού και έγκειται στο να πάρουμε $P_1=1, P_2=2, P_3=3$, κ.λ.π. Έτσι οι συγκρίσεις γίνονται με διαδοχικές θέσεις.

Δραστηριότητα 1 / Κεφ. 5

Ας υποθέσουμε ότι τα σύμβολα S1 και S2 κερματίζονται και εισάγονται στις θέσεις 2 και 4 αντίστοιχα. Έστω τώρα ότι το σύμβολο S3 επίσης κερματίζεται στη θέση 2. Λόγω της σύγκρουσης θα αποθηκευτεί στην θέση 3. Έστω τέλος ότι και το σύμβολο S4 κερματίζεται στην θέση 2. Τι παρατηρείτε; Σκεφθείτε λίγο πριν απαντήσετε και φυσικά πριν διαβάσετε την συνέχεια του κειμένου.

Παρατηρούμε ότι θα υπάρξουν 3 συγκρούσεις με τα S1, S2, S3 πριν το S4 τελικά αποθηκευτεί στην 5η θέση που είναι κενή.

Από το παράδειγμα φαίνεται το μειονέκτημα της μεθόδου μετά από μερικές συγκρούσεις. Οι θέσεις εισόδου τείνουν να ομαδοποιούνται σχηματίζοντας μακριές αλυσίδες από γεμάτες θέσεις.

Μια προσέγγιση του μέσου αριθμού συγκρίσεων για την εύρεση ενός στοιχείου, δίνεται από την σχέση:

$$E = (1-H/2) / (1-H)$$

Όπου **H** ο συντελεστής φόρτου. Έτσι για πίνακα γεμάτο κατά 10% αναμένουμε 1.06 συγκρίσεις. Για γεμάτο κατά 50%, 1.5 συγκρίσεις, και για γεμάτο κατά 90% 5.5 συγκρίσεις. Σημειώνεται ότι το E δεν εξαρτάται από το μέγεθος του πίνακα αλλά από το ποσο γεμάτος είναι.

Παρότι μπορούμε να βρούμε καλύτερες τιμές για τα P_1, P_2, P_3, \dots , η μέθοδος είναι καλύτερη από τη δυαδική ανίχνευση. Για ένα πίνακα 1024 θέσεων με 512 γεμάτες (50%) η δυαδική ανίχνευση χρειάζεται 9 με 10 συγκρίσεις ενώ ο γραμμικός επανακερματισμός μόνο 1.5.

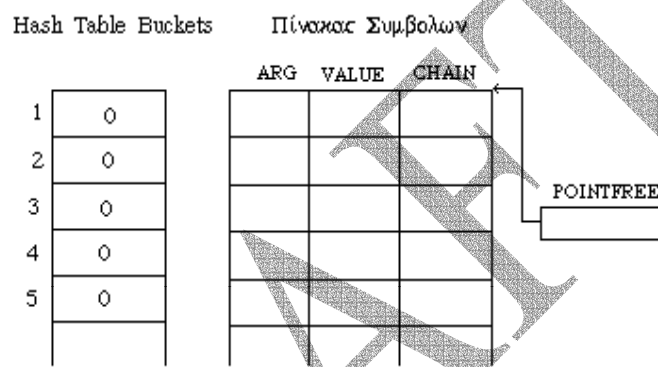
Εκτός από τον γραμμικό επανακερματισμό έχουν προταθεί και χρησιμοποιηθεί και άλλες μορφές επανακερματισμού όπως ο ψευδο-τυχαίος επανακερματισμός κ.α (μπορείς για παράδειγμα να συμβουλευθείς το [4]).

5.4.2 ΑΛΥΣΙΔΩΤΟΣ ΚΕΡΜΑΤΙΣΜΟΣ

Η μέθοδος του αλυσιδωτού κερματισμού χρησιμοποιεί τα εξής:

1. Ένα πίνακα κερματισμού (**hash table**) του οποίου τα στοιχεία καλούνται buckets και των οποίων η αρχική τιμή είναι 0.
2. Τον πίνακα συμβόλων που είναι αρχικά άδειος.
3. Ένα δείκτη POINTFREE που δείχνει την τρέχουσα τελευταία θέση του πίνακα συμβόλων και που αρχικά δείχνει στην θέση πριν από την αρχή του πίνακα.

Επί πλέον ο πίνακας συμβόλων έχει και μια ακόμη στήλη που καλείται chain (αλυσίδα) και της οποίας τα στοιχεία είναι 0 ή περιέχουν την διεύθυνση ενός άλλου στοιχείου του πίνακα συμβόλων. Έτσι η αρχική κατάσταση είναι όπως δείχνεται διαγραμματικά στο Σχήμα 5.4.



Σχήμα 5.4 Αρχική κατάσταση του πίνακα

Τα σύμβολα κερματίζονται σε buckets (κουβάδες - θέσεις) του hash Table. Κάθε bucket είναι 0 ή δείχνει το πρώτο στοιχείο του Πίνακα συμβόλων το οποίο κερματίζεται εκεί. Το πεδίο ARG περιέχει το όνομα, το πεδίο VALUE είναι ο περιγραφέας και το πεδίο CHAIN χρησιμοποιείται για να συνδέσει στοιχεία του πίνακα των οποίων τα σύμβολα κερματίζονται στην ίδια θέση του πίνακα κερματισμού.

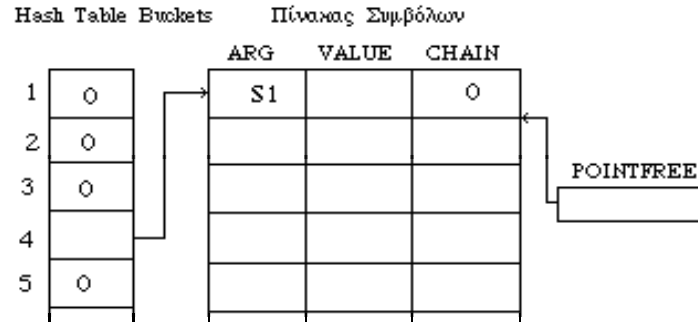
Παράδειγμα 4 / Κεφ. 5

Ας υποθέσουμε τώρα ότι έχουμε ένα κενό πίνακα συμβόλων και ότι το σύμβολο **S1** πρόκειται να εισαχθεί στον πίνακα. Έστω ότι η συνάρτηση κερματισμού (hash function) παράγει την διεύθυνση 4 του hash Table (**bucket 4**) η οποία είναι 0.

Κάνουμε τα εξής:

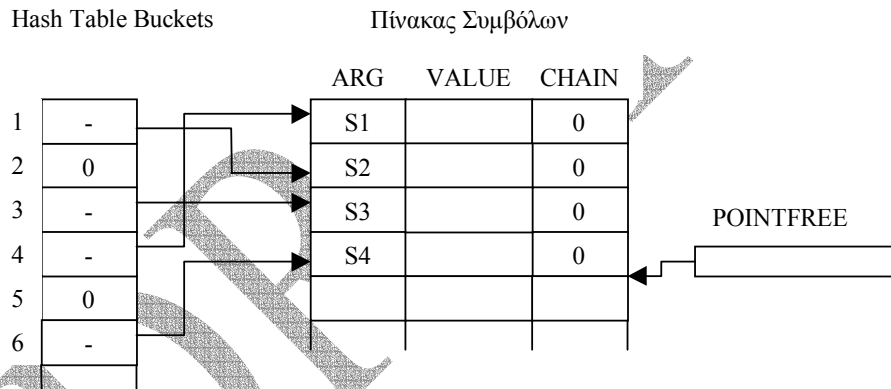
1. Προσθέτουμε 1 στο POINTFREE
2. Εισάγουμε (S1, τιμή, 0) στην θέση του πίνακα συμβόλων που δείχνει ο POINTFREE.
3. Βάζουμε την τιμή του POINTFREE στο bucket 4.

Ο πίνακας γίνεται ως εξής:



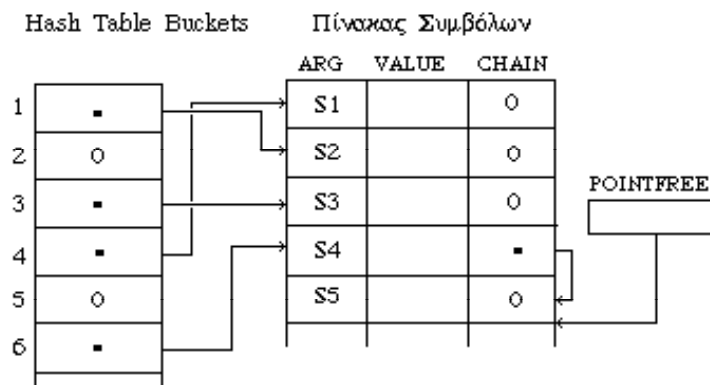
Σχήμα 5.5 Η εικόνα του πίνακα μετά την εισαγωγή του S1

Εφ' όσον σύμβολα κερματίζονται σε διαφορετικές διευθύνσεις, εισάγονται όπως παραπάνω. Έτσι, αν εισάγουμε τα σύμβολα S2, S3 και S4 τα οποία κερματίζονται στις διευθύνσεις 1,3 και 6 αντίστοιχα ο πίνακας θα εμφανίζεται όπως στο Σχήμα 5.6.



Σχήμα 5.6 Η εικόνα του πίνακα μετά την εισαγωγή των S2, S3, S4

Κατόπιν το σύμβολο S5 θα εισαχθεί και το οποίο κερματίζεται στη θέση 6 η οποία ήδη χρησιμοποιήθηκε. Τώρα γίνεται χρήση του δείκτη CHAIN όπως δείχνεται στο Σχήμα 5.7.



Σχήμα 5.7 Η εικόνα του πίνακα μετά την εισαγωγή του S5

Άσκηση αυτοαξιολόγησης 2 / Κεφ. 5

Συνεχίζοντας την προηγούμενη διαδικασία ας υποθέσουμε ότι ακολουθούν τα σύμβολα S6, S7, S8 τα οποία έστω ότι κερματίζονται στις θέσεις 4, 3 και 3 αντίστοιχα. Μπορείτε να δώσετε την εικόνα του πίνακα όπως θα διαμορφωθεί μετά την εισαγωγή αυτών των συμβόλων;

Παράδειγμα 5 / Κεφ. 5

Ο παρακάτω ψευδοκώδικας περιγράφει μια ρουτίνα η οποία ανιχνεύει τον πίνακα για κάποιο σύμβολο S και το εισάγει στον πίνακα εφόσον δεν υπάρχει ήδη. P1 και P2 είναι προσωρινές μεταβλητές δείκτη. HASH είναι μια συνάρτηση κερματισμού. Η ρουτίνα επιστρέφει στην P2 την διεύθυνση του S.

```

START: P1:=HASH(S) + ADDRESS (HASH-TABLE);
      P2:=CONTENTS (P1);
LOOP: IF P2=∅
      THEN
      BEGIN;POINTFREE:=POINTFREE+1;
      P2:=POINTFREE;
      P2.ARG:=S;
      P2.CHAIN:=∅;
      CONTENTS(P1):=POINTFREE;
      RETURN;
      END
      ELSE IF S=P2.ARG
      THEN RETURN
      ELSE BEGIN;
      P1:=ADDRESS (P2.CHAIN);
      P2:=CONTENTS (P1)
      GO TO LOOP
      END
  
```

Αν η γλώσσα προγραμματισμού (υλοποίησης) επιτρέπει δυναμική εκχώρηση χώρου, τότε σε περίπτωση που γεμίσει ο πίνακας συμβόλων μπορεί να αυξηθεί ο χώρος του.

5.4.3 ΣΥΝΑΡΤΗΣΕΙΣ ΚΕΡΜΑΤΙΣΜΟΥ (HASH FUNCTIONS)

Αν η παράμετρος S που πρόκειται να κερματιστεί έχει μήκος περισσότερο από μια λέξη υπολογιστή, το πρώτο βήμα της hash function είναι η δημιουργία ενός S' μιας λέξης από το S. Το S' συνήθως υπολογίζεται είτε προσθέτοντας όλες τις

λέξεις του S είτε χρησιμοποιώντας το αποκλειστικό (Exclusive) OR. Το δεύτερο βήμα είναι ο υπολογισμός του τελικού δείκτη από το S'. Υπάρχουν διάφοροι τρόποι για να γίνει αυτό:

1. Πολλαπλασίασε το S' με τον εαυτό του και χρησιμοποίησε τα μεσαία n bits σαν κερματισμό (αν ο πίνακας έχει 2^n στοιχεία). Τα μεσαία n bits εξαρτώνται από όλα τα bits του S'.
2. Χρησιμοποίησε κάποια λογική πράξη, όπως XOR, σε ορισμένα κομμάτια του S'.
3. Αν ο πίνακας έχει μήκος 2^n , χώρισε το S' σε κομμάτια των n bits και πρόσθεσέ τα. Χρησιμοποίησε τα δεξιότερα n bits του αθροίσματος.
4. Διείρεσε το S' με το μέγεθος του πίνακα και χρησιμοποίησε το υπόλοιπο σαν τον δείκτη κερματισμού ($h=S'(\text{mod } 2^n)$).

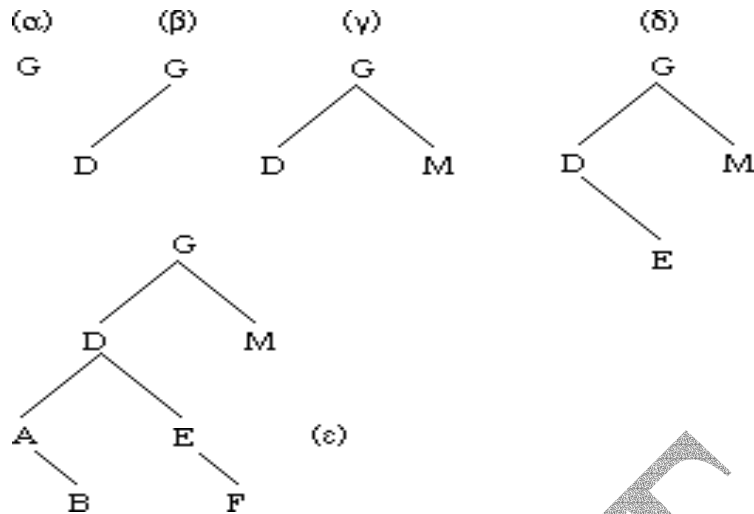
Όλες οι παραπάνω μέθοδοι έχουν χρησιμοποιηθεί με ικανοποιητικά αποτελέσματα. Άλλες μέθοδοι μπορούν επίσης να αναπτυχθούν. Προσοχή χρειάζεται ώστε η hash function να παράγει διευθύνσεις που είναι αρκετά τυχαίες για το είδος των παραμέτρων S που πρόκειται να χρησιμοποιηθούν.

Σημειώνεται ότι στον κώδικα EBCDIC τα δυο αριστερότερα bits για όλα τα κεφαλαία γράμματα και τα ψηφία είναι ίδια (11), ενώ στον ASCII τα δύο αριστερότερα bits των κεφαλαίων γραμμάτων είναι 10.

ΕΝΟΤΗΤΑ 5.5 ΔΕΝΔΡΙΚΑ ΔΟΜΗΜΕΝΟΙ ΠΙΝΑΚΕΣ

Η μέθοδος χρησιμοποιεί ένα δένδρο (δυναδικό). Κάθε κόμβος του δένδρου παριστάνει ένα γεμάτο στοιχείο του πίνακα και η ρίζα του δένδρου είναι το πρώτο στοιχείο.

Το παρακάτω σχήμα (α) δείχνει τον πίνακα με ένα στοιχείο για το σύμβολο (identifier) G. Έστω τώρα ότι το σύμβολο D πρόκειται να προστεθεί. Μια και $D < G$ το σχήμα (β) δείχνει τον πίνακα. Αν κατόπιν εισαχθεί το σύμβολο M ο πίνακας γίνεται όπως στο σχήμα (γ). Τέλος, έστω ότι εισάγονται τα σύμβολα E, A, B, και F. Ο πίνακας θα διαμορφωθεί όπως φαίνεται στα σχήματα (δ) και (ε).



Την δομή αυτή μπορεί να την υλοποιήσει κανείς κάνοντας χρήση δύο pointers με κάθε στοιχείο.

Ο αριθμός των συγκρίσεων που χρειάζονται εξαρτάται κατά πολύ από την σειρά με την οποία καταφθάνουν οι identifiers. Για παράδειγμα, αν φθάσουν με την σειρά A, B, C, D, E, F για να βρεθεί το F χρειάζονται 6 συγκρίσεις.

ΕΝΟΤΗΤΑ 5.6 ΠΙΝΑΚΕΣ ΣΥΜΒΟΛΩΝ ΓΙΑ ΓΛΩΣΣΕΣ ΜΕ ΔΟΜΕΣ BLOCK

Σε γλώσσες τύπου Algol (π.χ. Pascal) ο ίδιος identifier μπορεί να δηλώνεται πολλές φορές μέσα σε διαφορετικά blocks και procedures. Κάθε τέτοιος identifier πρέπει να αντιστοιχεί σε ένα μοναδικό στοιχείο του πίνακα συμβόλων. Αρχίζουμε με μια αρίθμηση των blocks όπως ανοίγουν (η σειρά με την οποία συναντώνται τα begins). Ο αλγόριθμος για την εύρεση της σωστής δήλωσης που αντιστοιχεί στην χρήση ενός identifier συνίσταται στο να ψάξουμε πρώτα στο τρέχον block (όπου ο identifier χρησιμοποιείται), κατόπιν στο περιβάλλον block κ.λ.π. μέχρις ότου ευρεθεί. Μια τέτοια ανίχνευση μπορεί να υλοποιηθεί αν κρατάμε όλα τα στοιχεία του πίνακα που ευρίσκονται στο ίδιο block σε συνεχόμενη περιοχή, χρησιμοποιώντας μια **block list**.

Προς στιγμή μπορούμε να υποθέσουμε ότι τα στοιχεία του πίνακα για κάθε block δεν είναι ταξινομημένα. Το στοιχείο i της block list περιέχει τον αριθμό του block που περιβάλλει το block i (SURRENO), το πλήθος των στοιχείων (τοπικών ονομάτων του block i) του πίνακα για το block i (NOENT), και ένα δείκτη

(POINT) στη θέση των στοιχείων αυτών στον πίνακα συμβόλων. Έτσι κάθε στοιχείο του block list είναι μια τριάδα (SURRNO, NOENT, POINT).

Παράδειγμα 6 / Κεφ. 5

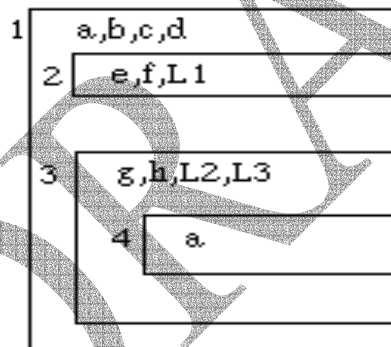
Ας πάρουμε για παράδειγμα το παρακάτω κομμάτι κώδικα

```

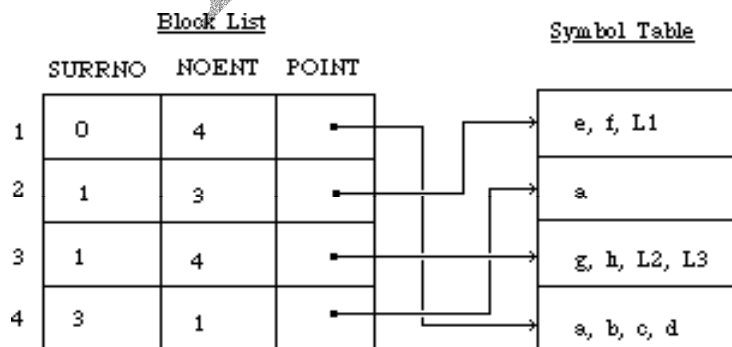
BEGIN REAL a,b,c,d;
  BEGIN REAL e,f;
    L1:
  END;
  BEGIN REAL g,h;
    L2:BEGIN REAL a;
      ...
    END;
  L3:
  END;
END;

```

Στο Σχήμα 5.8 δείχνεται η δομή των block και η εμφάνιση των ονομάτων, ενώ στο Σχήμα 5.9 δείχνονται τα περιεχόμενα της block list και του πίνακα συμβόλων μετά το κλείσιμο όλων των blocks.



Σχήμα 5.8 Η δομή των blocks και η εμφάνιση των ονομάτων



Σχήμα 5.9 Εικόνα της block list και του πίνακα συμβόλων μετά το κλείσιμο όλων των blocks.

5.6.1 ΑΝΟΙΓΜΑ ΚΑΙ ΚΛΕΙΣΙΜΟ ΤΩΝ BLOCKS

Τα blocks του πίνακα συμβόλων στο Σχήμα 5.9 εμφανίζονται με την σειρά 2, 4, 3, 1. Αυτή είναι η σειρά κατά την οποία έκλεισαν τα blocks (η σειρά εμφάνισης των ENDS στο πρόγραμμα). Αυτό υπαγορεύεται από την ανάγκη τα στοιχεία του κάθε block να είναι συνεχόμενα.

Για να υλοποιήσουμε το χτίσιμο των πινάκων χρησιμοποιούμε τον χώρο στο τέλος του πίνακα συμβόλων σαν στοίβα. Η στοίβα αυτή $S(N)$, $S(N-1)$, ... θα περιέχει όλα τα στοιχεία για κάθε block του οποίου το "begin" έχει συναντηθεί κατά την διάρκεια της συντακτικής ανάλυσης αλλά του οποίου το "end" δεν έχει συναντηθεί ακόμη. Όταν ολόκληρο το block έχει ανιχνευθεί τα στοιχεία του μετακινούνται στην αρχή του πίνακα.

Ας θεωρήσουμε ότι η σύνταξη ενός block είναι η:

- (1) `<block>` → `<blockbegin><declist>;<statlist> END`.
 (2) `<blockbegin>` → BEGIN

τότε θα ανοίξουμε ένα block όταν εκτελέσουμε την δράση (ρουτίνα) για τον κανόνα (2) και θα κλείσουμε ένα block όταν εκτελέσουμε την δράση για τον κανόνα (1). Λέγοντας δράση εννοούμε κομμάτι κώδικα – ρουτίνα – που εκτελείται όταν ο συντακτικός αναλυτής αναγνωρίσει μια σωστή συντακτική δομή. Οι δράσεις αυτές είναι αντικείμενο του επόμενου κεφαλαίου.

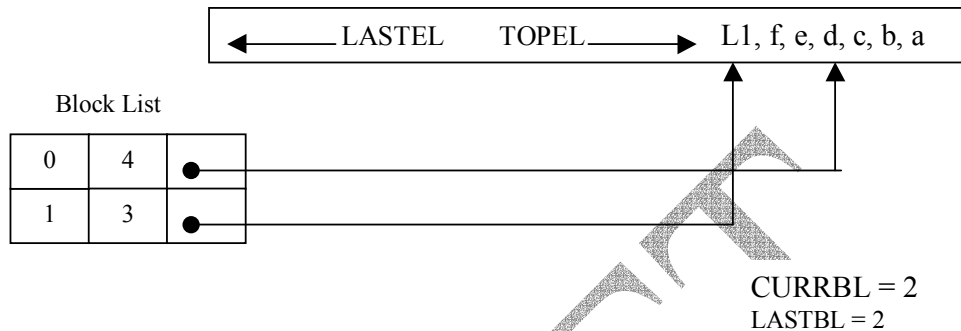
Οι μεταβλητές (δομές) που χρησιμοποιούνται στη συνέχεια είναι οι εξής:

- $S(1:N)$ Ο πίνακας συμβόλων με N στοιχεία του οποίου το κάτω μέρος $S(N)$, $S(N-1)$, ... χρησιμοποιείται σαν στοίβα. Τα στοιχεία τελικά μετακινούνται στις θέσεις $S(1)$, $S(2)$, ...
- $B(1:M)$ Η λίστα των blocks της οποίας κάθε στοιχείο έχει τρία πεδία: SURRNO, NOENT και POINT.
- CURRBL Ο τρέχων αριθμός του block, αρχικά 0.
- LASTBL Ο μεγαλύτερος αριθμός block, αρχικά 0.
- TOPEL Δείκτης στο κορυφαίο στοιχείο της στοίβας, αρχικά $N+1$.
- LASTEL Δείκτης του τελευταίου στοιχείου του Πίνακα συμβόλων (από την αρχή του πίνακα), αρχικά 0.

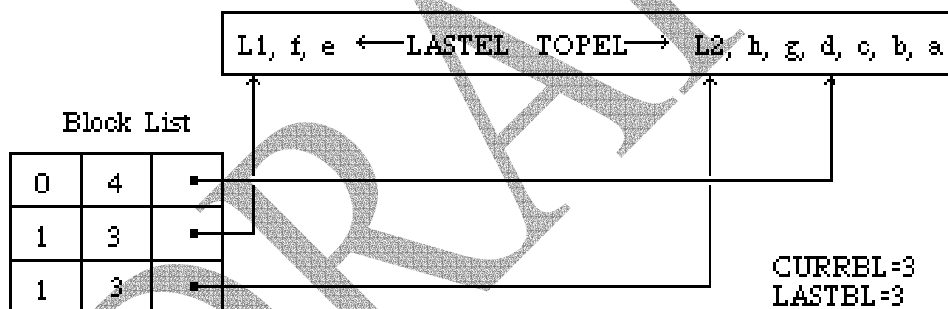
Παράδειγμα 7 / Κεφ. 5

Παρακάτω δείχνουμε "στιγμιότυπα" σε δυο διαφορετικά σημεία από την ανίχνευση του προγράμματος του παραδείγματος 6 στα σημεία των επιγραφών L1 και L2.

Πίνακας στο L1



Πίνακας στο L2



Άσκηση αυτοαξιολόγησης 3 / Κεφ. 5

Σε συνέχεια των προηγούμενων "στιγμιότυπων", μπορείτε να δώσετε ένα τρίτο στιγμιότυπο στο σημείο της επιγραφής L3 του προγράμματος;

Δραστηριότητα 2 / Κεφ.5

Γνωρίζετε ότι η σύνταξη ενός block είναι αυτή που περιγράφεται από τους κανόνες (1) και (2). Επιπλέον σας έχει δοθεί η δομή και οι μεταβλητές του πίνακα συμβόλων (όπως περιγράφηκαν παραπάνω). Μπορείτε να γράψετε ένα κομμάτι ψευδοκώδικα (δράση) μέσω του οποίου να υλοποιείται η διαδικασία ανοίγματος ενός νέου block στην block list; Προσπαθήστε χωρίς να κοιτάξετε την απάντηση που δίνεται στη συνέχεια.

Ο κώδικας ο οποίος θα υλοποιεί το άνοιγμα, δηλαδή την δημιουργία, ενός νέου block θα εκτελεσθεί από ένα bottom up συντακτικό αναλυτή όταν αυτός αναγνωρίσει το BEGIN στην κορυφή της στοίβας και το “μειώσει” στο <blockbegin>. Στο συντακτικό αναλυτή έχει δοθεί ο κανόνας και η δράση ως εξής:

<blockbegin> → BEGIN	{ κώδικας της δράσης }
----------------------	------------------------

Εμείς γνωρίζουμε ότι το αποτέλεσμα της δράσης πρέπει να είναι να προσθέσει ένα στοιχείο στο block list και να κάνει το στοιχείο αυτό να αναφέρεται σαν το τρέχον block. Έτσι λαμβάνοντας υπόψη την δομή του πίνακα και τις μεταβλητές που μας δόθηκαν γράφουμε την παρακάτω δράση.

<pre> { /* αύξησε κατά ένα τον μέγιστο αριθμό block LASTBL */ LASTBL := LASTBL+1; /* βάλε στη θέση LASTBL του block list μια τριάδα τιμών που αντιστοιχούν στο περιβάλλον block (SURRNO), στο πλήθος των τοπικών μεταβλητών (NOENT) και τη θέση του πίνακα που θα βρίσκονται οι μεταβλητές αυτές (POINT) */ B(LASTBL) := (CURRBL, 0, TOPEL); /* κάνε το block αυτό τρέχον */ CURRBL := LASTBL; } </pre>

Το κλείσιμο ενός block θα υλοποιηθεί, σε αντιστοιχία με το άνοιγμα, όταν ο συντακτικός αναλυτής, βάσει του κανόνα,

<block> → <blockbegin><declist>;<statlist> END	{δράση κλεισίματος}
--	---------------------

αναγνωρίσει στην κορυφή της στοίβας το,

<blockbegin><declist>;<statlist> END

το οποίο και θα μειώσει σε <block>. Τότε θα εκτελέσει και τον κώδικα {δράση κλεισίματος}.

Άσκηση αυτοαξιολόγησης 4 / Κεφ. 5

Προσπάθησε να γράψεις τον ψευδοκώδικα που αντιστοιχεί στη {δράση κλεισίματος} ενός block. Σου θυμίζω ότι πρέπει να μετακινήσεις τις μεταβλητές του τρέχοντος (CURRBL) block από το τέλος της στοίβας (δείχνεται από τον δείκτη TOPEL), στις μόνιμες θέσεις τους στην αρχή της στοίβας (δείχνεται από τον δείκτη LASTEL). Μην ξεχάσεις στο τέλος να κάνεις το περιβάλλον block, του block που έκλεισε, τρέχον block.

5.6.2 ΕΙΣΑΓΩΓΗ ΚΑΙ ΑΝΙΧΝΕΥΣΗ

Η εισαγωγή ενός identifier γίνεται μόνο στο τρέχον block και συνίσταται στην πρόσθεσή του στο κορυφαίο block του πίνακα συμβόλων (εφόσον δεν υπάρχει ήδη εκεί) και την αύξηση κατά ένα του αριθμού των μεταβλητών του block αυτού.

Άσκηση αυτοαξιολόγησης 5 / Κεφ. 5

Γράψε ψευδοκώδικα για την υλοποίηση της ρουτίνας εισαγωγής ενός identifier στο τρέχον block του πίνακα συμβόλων.

Για την ανίχνευση (look up) του πίνακα συμβόλων χρειάζονται δυο ρουτίνες. Μια για να ψάξει στο τρέχον block μόνο και μια για να ψάξει στο τρέχον αλλά και στα περιβάλλοντα blocks. Η πρώτη ρουτίνα καλείται από τον συντακτικό αναλυτή στα πλαίσια της ανίχνευσης μέσα στις δηλωτικές εντολές μόνο, ενώ δεύτερη καλείται όταν ο αναλυτής βρίσκεται μέσα σε εκτελέσιμες εντολές.

5.6.3 ΚΕΡΜΑΤΙΣΜΟΣ ΔΙΕΥΘΥΝΣΕΩΝ ΚΑΙ ΔΟΜΗ BLOCK

Όταν χρησιμοποιείται κερματισμός διευθύνσεων δεν είναι εύκολη η ταξινόμηση του πίνακα συμβόλων κατά blocks. Μια μέθοδος για να λυθεί το πρόβλημα των blocks είναι να συνδυαστεί ο αριθμός του block με τον identifier οπότε έχουμε μοναδικό σύμβολο. Για να βρεθεί ένας identifier B στην περίπτωση αυτή, θα πρέπει κανείς να ψάξει πρώτα για το B συνδυασμένο με τον τρέχοντα αριθμό block, μετά για το B συνδυασμένο με τον περιβάλλοντα αριθμό block κ.λ.π.

ΣΥΝΟΨΗ ΚΕΦΑΛΑΙΟΥ

Στο κεφάλαιο αυτό μελετήσαμε τα περιεχόμενα και τρόπους οργάνωσης του πίνακα συμβόλων. Μελετήσαμε ταξινομημένους και μη ταξινομημένους πίνακες, τον κερματισμό διευθύνσεων, σαν ένα διαφορετικό τρόπο προσπέλασης των στοιχείων του πίνακα. Ακόμη μελετήσαμε την μέθοδο του αλυσιδωτού κερματισμού και διάφορες μορφές συναρτήσεων κερματισμού και είδαμε σύντομα την δενδρική μορφή των πινάκων. Επίσης μελετήσαμε την οργάνωση και προσπέλαση πινάκων για γλώσσες με δομές block, είδαμε πώς μπορούμε να ανοίξουμε και να κλείσουμε ένα block, πώς να εισάγουμε ένα όνομα στον πίνακα συμβόλων. Τέλος σημειώσαμε ότι πρέπει να συνδυάσουμε τον αριθμό του

περιβάλλοντος block με το όνομα που θέλουμε να εισάγουμε στον πίνακα όταν χρησιμοποιούμε κερματισμό διευθύνσεων.

ΑΠΑΝΤΗΣΕΙΣ ΑΣΚΗΣΕΩΝ ΑΥΤΟΑΞΙΟΛΟΓΗΣΗΣ

Απάντηση άσκησης 1 / Κεφ. 5

1. Για απλές μεταβλητές ο πίνακας συμβόλων πρέπει να περιλαμβάνει τουλάχιστον τις εξής πληροφορίες

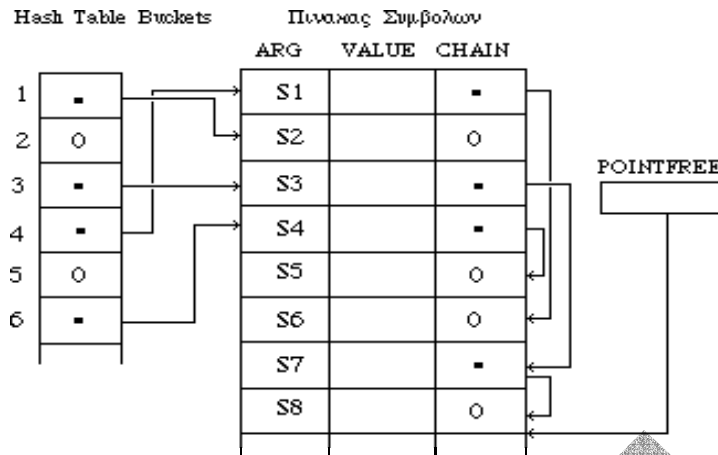
- τύπος της μεταβλητής (real, boolean, label, κ.λ.π.)
- μορφή (απλή μεταβλητή, array, record, κ.λ.π.)
- ακρίβεια, μήκος
- διεύθυνση κατά τον χρόνο εκτέλεσης (run time address)

2. Για procedures ο πίνακας συμβόλων πρέπει να περιλαμβάνει τουλάχιστον τις εξής πληροφορίες

- είναι external στο πρόγραμμα;
- έχει συναντηθεί η δηλωτική της εντολή;
- είναι recursive;
- ποιες είναι οι τυπικές της παράμετροι; οι "περιγραφείς" των παραμέτρων αυτών πρέπει να συνδέονται με το όνομα της συνάρτησης για λόγους σύγκρισης με τις πραγματικές παραμέτρους κατά τον χρόνο εκτέλεσης.

Απάντηση άσκησης αυτοαξιολόγησης 2 / Κεφ. 5

Η εικόνα του πίνακα μετά την εισαγωγή των συμβόλων S6, S7, S8 τα οποία ότε κερματίζονται στις θέσεις 4, 3 και 3 αντίστοιχα είναι ως εξής



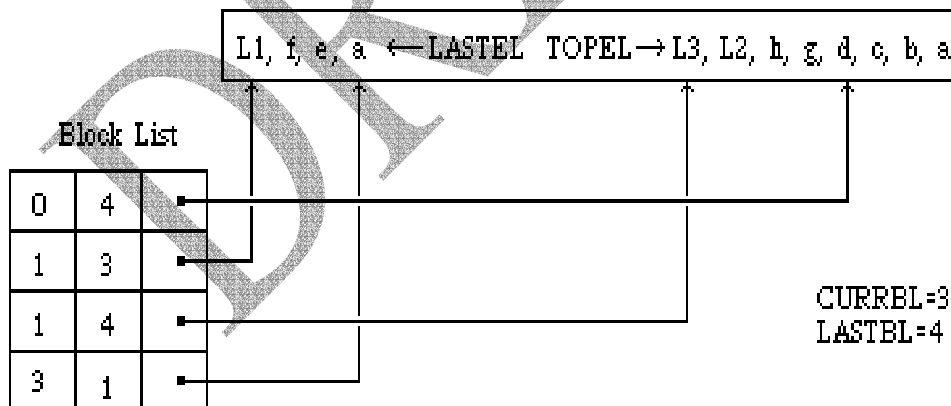
Αν τα κατάφερες μπράβο σου. Προχωράς πολύ καλά.

Αν δεν κατάφερες να φτιάξεις την παραπάνω μορφή του πίνακα, δεν πειράζει αρκεί να δοκιμάσεις πάλι αφού ξαναμελετήσεις κυρίως την χρήση της CHAIN γιατί μάλλον εκεί θα βρίσκεται το πρόβλημά σου.

Απάντηση άσκησης αυτοαξιολόγησης 3 / Κεφ. 5

Ένα τρίτο στιγμιότυπο στο σημείο της επιγραφής L3 του προγράμματος είναι το εξής

Πίνακας στο L3



Βλέπουμε ότι τα blocks 2 και 4 έχουν ήδη κλείσει και οι αντίστοιχες μεταβλητές έχουν μεταφερθεί από την κορυφή της στοίβας στις τελικές τους θέσεις στο αριστερό μέρος του πίνακα συμβόλων. Αντίθετα τα blocks 1 και 3 παραμένουν ανοικτά.

Αν η απάντησή σου δεν είναι ίδια με την παραπάνω πρέπει να ξαναδιαβάσεις την αντίστοιχη ενότητα και να ξαναπροσπαθήσεις. Μην επιχειρήσεις την επόμενη άσκηση αν δεν λύσεις πρώτα αυτή.

Απάντηση άσκησης αυτοαξιολόγησης 4 / Κεφ. 5

Ο ψευδοκώδικας που αντιστοιχεί στη {δράση κλεισίματος} ενός block είναι ως εξής.

```
/* βάλε τον δείκτη να δείχνει στην αρχή της τελικής θέσης των ονομάτων */
B(CURRBL).POINT:= LASTEL+1;
/* μετακίνησε τα ονόματα στην τελική τους θέση*/
/* μετακινείς τόσα → B(CURRBL).NOENT ονόματα αυτού του block */
FOR i :=1 STEP 1 UNTIL B(CURRBL).NOENT DO
BEGIN   LASTEL := LASTEL+1;
        S(LASTE) := S(TOPEL);
        TOPEL := TOPEL+1
END; /* μετακίνηση ολοκληρώθηκε */
/* Κάνε τώρα το περιβάλλον block τρέχον */
CURRBL := B(CURRBL).SURRNO
```

Αν η απάντησή σου είναι ίδια ή παραπλήσια με την παραπάνω τότε συγχαρητήρια γιατί όχι μόνο έχεις διαβάσει και κατανοήσει την αντίστοιχη ενότητα πολύ καλά, αλλά ξέρεις και να κάνεις καλή χρήση των σχολίων μέσα στον κώδικα.

Αν η απάντησή σου απέχει από την παραπάνω, μην το βάζεις κάτω γιατί δεν ήτανε και εύκολη η άσκηση. Μελέτησε προσεκτικά την απάντηση και διάβασε πάλι την αντίστοιχη ενότητα του κεφαλαίου.

Απάντηση άσκησης αυτοαξιολόγησης 5 / Κεφ. 5

Γράψε ψευδοκώδικα για την υλοποίηση της ρουτίνας εισαγωγής ενός identifier στο τρέχον block του πίνακα συμβόλων.

```
/* Αρχή ρουτίνας εισαγωγής ονόματος στον πίνακα συμβόλων */
/* υποθέτω ότι το όνομα που θα εισαχθεί στον πίνακα συμβόλων βρίσκεται στην
σφαιρική μεταβλητή NAME*/
/* βάζω το NAME στην κορυφή της στοίβας στην θέση TOPEL-1 */
/* κανονικά πρέπει εδώ να κάνω και έλεγχο υπερχείλισης της στοίβας. Το αφήνω
όμως να το κάνεις εσύ */
TOPEL := TOPEL - 1;
S(TOPEL) := NAME;
/* αυξάνω κατά 1 το πλήθος των ονομάτων του block */
B(CURRBL).NOENT := NOENT +1;
```

```
/* κάνω τον δείκτη POINT να δείχνει στην κορυφή της στοίβας */  
B(CURRBL).POINT := TOPEL;  
/* τέλος ρουτίνας εισαγωγής ονόματος στον πίνακα συμβόλων */
```

Αν ο κώδικάς σου είναι σαν της απάντησης τότε μπράβο σου, προχωράς πολύ καλά. Αν έχεις κάνει και έλεγχο υπερχειλίσης, τότε η απάντησή σου εκτός από σωστή είναι και πλήρης. Αν όχι, τότε κάνε και έλεγχο υπερχειλίσης, είναι απλός.

Αν ο κώδικάς σου δεν είναι όπως στην απάντηση, τότε, μελέτησε την απάντηση και ξαναδοκίμασε.

Παρατήρησε ότι στην παραπάνω ρουτίνα λείπει κάτι βασικό: δεν κάνω έλεγχο μήπως το όνομα υπάρχει ήδη στον πίνακα συμβόλων. Αν το όνομα υπάρχει ήδη στον πίνακα πρέπει η ρουτίνα μου να εντοπίζει το γεγονός αυτό και να ενημερώνει κατάλληλα τον συντακτικό αναλυτή ο οποίος πρέπει να αναφέρει πιθανό συντακτικό λάθος (πολλαπλή δήλωση ονόματος). Προσπάθησε να ξαναγράψεις την ρουτίνα και συζήτησε τη λύση σου με τον καθηγητή σου.