

Αρχές Γλωσσών Προγραμματισμού

Χρήστος Νομικός

Τμήμα Μηχανικών Η/Υ και Πληροφορικής
Πανεπιστήμιο Ιωαννίνων

2015

- 1 Τύποι δεδομένων
- 2 Μετατροπές τύπων
- 3 Ισοδυναμία τύπων

- 1 Τύποι δεδομένων
- 2 Μετατροπές τύπων
- 3 Ισοδυναμία τύπων

Μπορούμε να δούμε τους τύπους δεδομένων μίας γλώσσας από διάφορες οπτικές γωνίες

- δηλώτική: ο τύπος καθορίζεται από το πεδίο τιμών του
- κατασκευαστική: ο τύπος καθορίζεται από τον τρόπο κατασκευής του από τους πρωταρχικούς τύπους της γλώσσας
- αφαιρετική: ο τύπος καθορίζεται από ένα σύνολο πράξεων

Ένα σύστημα τύπων αποτελείται από:

- Έναν μηχανισμό ορισμού τύπων, μαζί με ένα σύνολο πρωταρχικών τύπων που υποστηρίζονται άμεσα από τη γλώσσα
- Ένα σύνολο κανόνων για ισοδυναμία, συμβατότητα και εξαγωγή τύπων

Οι τύποι χωρίζονται σε

- Απλούς ή βαθμωτούς: το πεδίο τιμών περιγράφεται από ένα σύνολο σταθερών τιμών
- Σύνθετους ή δομημένους: το πεδίο τιμών περιγράφεται ως συνάρτηση των πεδίων τιμών άλλων τύπων

Ο καθορισμός του τύπου μιας μεταβλητής μπορεί να γίνεται:

- στατικά, άμεσα (με δήλωση) ή έμμεσα
- δυναμικά (σε χρόνο εκτέλεσης)

Στη συνέχεια περιγράφουμε τους κυριότερους τύπους δεδομένων καθώς και ένα σύνολο βασικών λειτουργιών που συνήθως υποστηρίζουν.

Ακέραιοι

Πεδίο τιμών: ένα υποσύνολο του συνόλου των φυσικών αριθμών.
Συνήθως περιλαμβάνει όλες τις τιμές μεταξύ -2^{k-1} και $2^{k-1} - 1$, όπου k κάποιο πολλαπλάσιο του μήκους λέξης.

Προκαθορισμένες λειτουργίες: τουλάχιστον οι αριθμητικές πράξεις

- πρόσθεση
- αφαίρεση
- πολλαπλασιασμός
- ακέραια διαίρεση (αποκόπτει το δεκαδικό μέρος)
- υπόλοιπο διαίρεσης

Πολλές γλώσσες υποστηρίζουν περισσότερους από έναν τύπους ακεραίων, με διαφορετικά πεδία τιμών.

Η Java, η Haskell και η Python υποστηρίζουν μεταξύ άλλων τύπο ακέραιου οσοδήποτε μεγάλου μήκους.

Η Pascal διαθέτει έναν μόνο ακέραιο τύπο, τον integer.

Πραγματικοί

Πεδίο τιμών: ένα υποσύνολο του συνόλου των ρητών αριθμών, που θεωρείται ότι προσεγγίζουν πραγματικούς.

Προκαθορισμένες λειτουργίες: οι τέσσερις βασικές αριθμητικές πράξεις

- πρόσθεση
- αφαίρεση
- πολλαπλασιασμός
- διαίρεση

Πολλές γλώσσες υποστηρίζουν περισσότερους από έναν τύπους πραγματικών, που διαφέρουν στην ακρίβεια.

Επίσης οι γλώσσες συνήθως υποστηρίζουν δύο μορφές γραφής πραγματικών σταθερών: σταθερής υποδιαστολής ή κινητής υποδιαστολής.

Η Pascal διαθέτει έναν μόνο παραγματικό τύπο, τον real.

Η Fortran και η Python υποστηρίζουν και μιγαδικούς αριθμούς.

Συνήθως οι αριθμητικοί τύποι βασίζονται στην αριθμητική της μηχανής.

- πλεονέκτημα: οι πράξεις υλοποιούνται σε επίπεδο μηχανής, κέρδος σε ταχύτητα.
- μειονέκτημα: εξάρτηση από τη μηχανή.

Στη Java οι αριθμητικοί τύποι (π.χ. το πεδίο τιμών) είναι ανεξάρτητοι από τη μηχανή.

Boolean

Πεδίο τιμών: {true, false }

Προκαθορισμένες λειτουργίες: οι λογικές πράξεις

- σύζευξη (και)
- διάζευξη (ή)
- άρνηση (όχι)

Η Pascal έχει τύπο `boolean`, καθώς και τους λογικούς τελεστές `and`, `or`, `not`.

Η C, η APL και η PL/1 εξομοιώνουν τον τύπο `boolean` χρησιμοποιώντας ακεραίους.

Σε ορισμένες γλώσσες τα `true`, `false` είναι δεσμευμένες λέξεις (Algol 60) ενώ σε άλλες προκαθορισμένες λέξεις (Pascal).

Οι αληθοτιμές στην Pascal είναι διατεταγμένες (`false < true`).

Χαρακτήρες

Πεδίο τιμών: οι χαρακτήρες του αλφαβήτου του υπολογιστή

Προκαθορισμένες λειτουργίες: σύγκριση (οι χαρακτήρες θεωρούνται διατεταγμένοι), προηγούμενος, επόμενος.

Η Pascal έχει τον τύπο char

Η C επιτρέπει αριθμητικές πράξεις ανάμεσα σε χαρακτήρες, ή χαρακτήρες και αριθμούς.

Η Pascal διαθέτει συναρτήσεις μετατροπής ενός χαρακτήρα στον αντίστοιχο ακέραιο και αντίστροφα.

Συμβολοσειρές

Πεδίο τιμών: οι ακολουθίες χαρακτήρων από ένα υποσύνολο του αλφαβήτου του υπολογιστή.

Προκαθορισμένες λειτουργίες: λεξικογραφική σύγκριση και παράθεση.

Σε πολλές γλώσσες δεν είναι πρωταρχικός τύπος αλλά ορίζεται ως πίνακας χαρακτήρων (Pascal, C, C++).

Στη Snobol, που είναι γλώσσα χειρισμού συμβολοακολουθιών, είναι πρωταρχικός τύπος και υπάρχουν πολλές άλλες προκαθορισμένες λειτουργίες.

Στη Java υπάρχει η κλάση String που δεν είναι ισοδύναμη με πίνακα χαρακτήρων.

Μία μεταβλητή τύπου string μπορεί να έχει προκαθορισμένο μήκος ή όχι.

Απαριθμητοί τύποι

Πεδίο τιμών: ένα πεπερασμένο σύνολο από συμβολικά ονόματα, που καθορίζονται από το χρήστη κατά τον ορισμό του τύπου.

Προκαθορισμένες λειτουργίες: σύγκριση για ισότητα. Αν θεωρηθεί ότι οι τιμές είναι διατεταγμένες, τότε ορίζονται $<$, $>$, επόμενος, προηγούμενος.

Η χρήση απαριθμητών τύπων, αντί για κωδικοποίηση με ακεραίους:

- αυξάνει την αναγνωσιμότητα του προγράμματος
- αυξάνει την αξιοπιστία του προγράμματος

Ερωτήματα:

- επιτρέπεται η χρήση του ίδιου αναγνωριστικού σε περισσότερους από έναν απαριθμητούς τύπους; (Στην Pascal όχι)
- υπάρχει διάταξη στις τιμές; (στην Pascal ναι)

Υλοποίηση: αντιστοιχούν σε διαδοχικές ακέραιες τιμές.

Παράδειγμα ορισμού απαριθμητών τύπων στην Pascal:

```
type  
  day = (sun, mon, tue, wed, thu, fri, sat);  
  color = (white, red, green, blue);
```

Τύποι υποδιαστήματος

Πεδίο τιμών: ένα διάστημα συνεχόμενων τιμών από έναν τύπο με διατεταγμένο πεδίο τιμών.

Προκαθορισμένες λειτουργίες: οι λειτουργίες που είναι διαθέσιμες για τον ευρύτερο τύπο.

Η χρήση υποδιαστημάτων, όπως και η χρήση απαριθμητών τύπων:

- αυξάνει την αναγνωσιμότητα του προγράμματος
- αυξάνει την αξιοπιστία του προγράμματος

Συνήθως είναι συμβατός με τον ευρύτερο τύπο του οποίου αποτελεί υποδιάστημα.

Υλοποίηση: όπως και ο ευρύτερος τύπος, με προσθήκη κώδικα κατά τη μετάφραση για έλεγχο τύπου.

Παράδειγμα ορισμού τύπων διαστήματος στην Pascal:

```
type  
  smallint = -31 .. 31;  
  workingday = mon .. fri;  
  uppercase = 'A' .. 'Z';
```

Πίνακες

Πίνακας ονομάζεται μία συλλογή δεδομένων που όλα έχουν τον ίδιο τύπο και καθένα προσδιορίζεται από τη θέση του. Η θέση ενός στοιχείου καθορίζεται από έναν ή περισσότερους δείκτες. Το πλήθος των δεικτών καθορίζει τη διάσταση του πίνακα.

Πεδίο τιμών: D^* , όπου D το πεδίο τιμών ενός τύπου.

Προκαθορισμένες λειτουργίες: προσπέλαση ενός στοιχείου του πίνακα που βρίσκεται σε συγκεκριμένη θέση.

Αριθμός διαστάσεων:

- στις διάφορες εκδόσεις της Fortran είναι περιορισμένος (3 στη Fortran I και 7 στη Fortran 77).
- Οι γλώσσες C, C++ και Java υποστηρίζουν άμεσα μόνο μονοδιάστατους πίνακες, ενώ οι πολυδιάστατοι πίνακες υλοποιούνται έμμεσα ως πίνακες που τα στοιχεία τους επίσης πίνακες.
- Η Pascal υποστηρίζει πίνακες οποιασδήποτε διάστασης

Τύπος δεικτών:

- μπορεί να είναι οποιοσδήποτε ακέραιος τύπος
- δεν μπορεί να είναι πραγματικός αριθμός
- στην Pascal μπορεί να είναι χαρακτήρας, boolean ή απαριθμητός τύπος

Προσπέλαση στοιχείου: με το όνομα του πίνακα ακολουθούμενο από μία σειρά δεικτών μέσα σε () ή []:

- Fortran: $A(2,3)$
- C: $A[2][3]$
- Pascal: $A[2,3]$

Αρχικοποίηση:

- ορισμένες γλώσσες επιτρέπουν αρχικοποίηση των πινάκων (C, C++, Fortran, Ada)
- άλλες γλώσσες δεν επιτρέπουν αρχικοποίηση (Pascal, Modula-2)

Το μέγεθος ενός πίνακα μπορεί να καθορίζεται:

- στατικά (Fortran, Pascal, C)
- δυναμικά (σε χρόνο εκτέλεσης) (Algol, Java)

Ορισμένες γλώσσες δίνουν τη δυνατότητα περιγραφής υποπινάκων (τμημάτων του πίνακα).

Υλοποίηση:

- τα στοιχεία ενός πίνακα αποθηκεύονται σε συνεχόμενες θέσεις μνήμης
- οι πολυδιάστατοι πίνακες αποθηκεύονται είτε κατά γραμμές (αυξάνοντας ταχύτερα τον δεξιότερο δείκτη) ή κατά στήλες (αυξάνοντας ταχύτερα τον αριστερότερο δείκτη)
- η θέση ενός στοιχείου στη μνήμη, με βάση την ακολουθία δεικτών που το προσδιορίζει, υπολογίζεται με βάση μία απλή αριθμητική παράσταση.

Παράδειγμα ορισμού τύπων πίνακα στην Pascal:

type

```
vector = array[1..1000] of real;
```

```
string = array[1..MaxStringSize] of char;
```

```
matrix = array[1..9,'a'..'f',day] of integer;
```

Εγγραφές (δομές)

Εγγραφή ονομάζεται μία συλλογή δεδομένων που ενδεχόμενα ανήκουν σε διαφορετικούς τύπους και καθένα προσδιορίζεται από το όνομά του.

Πεδίο τιμών: $D_1 \times D_2 \times \dots \times D_n$, όπου D_i το πεδίο τιμών ενός τύπου.

Προκαθορισμένες λειτουργίες: προσπέλαση ενός πεδίου της εγγραφής, σύγκριση για ισότητα, αντιγραφή.

Αρχικοποίηση:

- η περισσότερες γλώσσες δεν επιτρέπουν αρχικοποίηση των δομών (Pascal, C)
- ορισμένες γλώσσες επιτρέπουν αρχικοποίηση των δομών (Ada)

Προσπέλαση πεδίου:

- Pascal,C,Ada: `x.age`
- Algol: `age(x)`
- Cobol: `age of x`
- ML: `#age x`

Υλοποίηση: τα πεδία μιας εγγραφής αποθηκεύονται σε συνεχόμενες θέσεις μνήμης.

Παράδειγμα ορισμού τύπου εγγραφής στην Pascal:

```
type
  point = record
    x,y:real;
    c:color;
  end;
```


Ενώσεις

Ένωση είναι ο τύπος δεδομένων που επιτρέπει σε μία μεταβλητή να παίρνει τιμές από διαφορετικούς τύπους σε διαφορετικές χρονικές στιγμές. Συνήθως χρησιμοποιείται ένα διαφορετικό όνομα για την τιμή κάθε τύπου.

Πεδίο τιμών: $D_1 \cup D_2 \cup \dots \cup D_n$, όπου D_i το πεδίο τιμών ενός τύπου.

Σε πολλές περιπτώσεις παρέχονται σε συνδυασμό με τις εγγραφές (variant records της Pascal).

Δημιουργούν πρόβλημα με τον έλεγχο τύπου: ενδέχεται να διαβάσουμε μία μεταβλητή θεωρώντας διαφορετικό τύπο από τον αποθηκευμένο.

Σύνολα

Πεδίο τιμών: 2^D , όπου D το πεδίο τιμών ενός τύπου.

Προκαθορισμένες λειτουργίες: ένωση, τομή, διαφορά, σύγκριση (για ισότητα, υποσύνολο)

Υλοποίηση: αν το D είναι μικρού μεγέθους, τότε το σύνολο παριστάνεται με ακολουθία από bits.

Παράδειγμα ορισμού τύπου συνόλου στην Pascal:
`type schedule = set of days;`

Δείκτες

Πεδίο τιμών: οι διευθύνσεις μνήμης του υπολογιστή $\cup \{nil\}$

Προκαθορισμένες λειτουργίες: σύγκριση για ισότητα, προσπέλαση της τιμής στη θέση που δείχνει, δέσμευση-αποδέσμευση μνήμης.

Χρειαζόμαστε έναν τρόπο για να καθορίζεται το αν αναφερόμαστε στη τιμή της μεταβλητής τύπου δείκτη (που είναι η θέση μνήμης που δείχνει ο δείκτης ή nil) ή στην τιμή που είναι αποθηκευμένη στη θέση που δείχνει ο δείκτης.

- Στη Pascal και τη C με το όνομα της μεταβλητής τύπου δείκτη αναφερόμαστε στον ίδιο τον δείκτη. Για παράδειγμα η εντολή της Pascal $p:=q$ και η εντολή $p=q$ της C έχουν ως αποτέλεσμα ο δείκτης p να δείχνει εκεί που δείχνει και ο q .
- Για να αναφερθούμε στην τιμή που είναι αποθηκευμένη στη θέση που δείχνει ο δείκτης χρησιμοποιούμε τον τελεστή \uparrow στην Pascal και τον τελεστή $*$ στη C. Για παράδειγμα η εντολή της Pascal $p\uparrow:=q\uparrow$ και η εντολή $*p=*q$ της C έχουν ως αποτέλεσμα η θέση μνήμης που δείχνει ο p να αποκτήσει το ίδιο περιεχόμενο με τη θέση μνήμης που δείχνει ο q . Η θέση μνήμης που δείχνει ο p δεν αλλάζει.

Ενδέχεται περισσότεροι από ένας δείκτες να δείχνουν στην ίδια θέση μνήμης.

Σκουπίδια ονομάζονται οι θέσεις μνήμης που έχουν παραχωρηθεί δυναμικά στο πρόγραμμα και προς τις οποίες δεν υπάρχει πρόσβαση μέσω δεικτών από το πρόγραμμα.

Παράδειγμα δημιουργίας σκουπιδιών:

```
new(p);  
p:=q
```

Η πρώτη εντολή παραχωρεί στο πρόγραμμα χώρο μνήμης στον οποίο δείχνει ο p. Η δεύτερη εντολή αλλάζει την τιμή του p, χάνοντας την πρόσβαση προς τον παραχωρηθέντα χώρο μνήμης.

Η δημιουργία σκουπιδιών μειώνει τη διαθέσιμη μνήμη κατά την εκτέλεση του προγράμματος.

Συλλογή σκουπιδιών ονομάζεται η διαδικασία εύρεσης και αποδέσμευσης των θέσεων μνήμης που αποτελούν σκουπίδια. Η συλλογή σκουπιδιών γίνεται αυτόματα από τη γλώσσα.

Ξεκρέμαστοι δείκτες ονομάζονται δείκτες οι οποίες δείχνουν σε μη έγκυρες θέσεις μνήμης, για παράδειγμα σε θέσεις που έχουν επιστραφεί από το πρόγραμμα στο σύστημα.

Παράδειγμα δημιουργίας ξεκρέμαστου δείκτη:

```
p:=q  
dispose(q);
```

Η πρώτη εντολή έχει ως αποτέλεσμα ο p να δείχνει στη θέση μνήμης που δείχνει και ο q. Η δεύτερη εντολή επιστρέφει τη θέση μνήμης όπου δείχνει ο q στο σύστημα.

Η ύπαρξη ξεκρέμαστων δεικτών είναι καταστροφική για την εκτέλεση του προγράμματος.

Η Java δεν περιέχει τύπο δείκτη, όμως η πρόσβαση προς τα αντικείμενα γίνεται με τα handles που έχουν ανάλογες ιδιότητες με τους δείκτες.

Παράδειγμα ορισμού τύπου δείκτη στην Pascal:

```
type
  intptr = ↑integer;
  list = ↑node;
  node = record
    value: integer;
    next: list
  end
```

Συναρτήσεις

Στις συναρτησιακές γλώσσες οι συναρτήσεις αποτελούν τύπο δεδομένων, καθώς μπορούν να περαστούν ως παράμετροι ή να επιστραφούν από άλλες συναρτήσεις (οι οποίες ονομάζονται συναρτήσεις υψηλότερης τάξης).

Ορισμένες γλώσσες μεταξύ των οποίων και η Pascal επιτρέπουν στα υποπρογράμματά τους να δέχονται ως παραμέτρους άλλα υποπρογράμματα.

Παράδειγμα υποπρογράμματος ως παραμέτρου στην Pascal:

```
type  
  function integral(function f(x:real):real; a,b:real);
```

- 1 Τύποι δεδομένων
- 2 Μετατροπές τύπων
- 3 Ισοδυναμία τύπων

Σε πολλές περιπτώσεις είναι χρήσιμο να μετατρέπουμε τιμές ενός τύπου σε αντίστοιχες τιμές άλλου τύπου, π.χ. μετατροπή ακεραίου σε πραγματικό.

Η μετατροπή τυπου μπορεί να γίνεται:

- με ρητό τρόπο (casting), π.χ.
`c = (char) n;`
- αυτόματα αν είναι αναγκαίο, π.χ. στη Java:
`s = "n=" + n;`

- ονομάζουμε διεύρυνση τη μετατροπή μίας τιμής από έναν τύπο με στενότερο πεδίο τιμών σε έναν τύπο με ευρύτερο πεδίο τιμών, π.χ ακεραίου σε πραγματικό, πραγματικού απλής ακρίβειας σε διπλής ακρίβειας κ.λ.π. Σε μία τέτοια μετατροπή δεν έχουμε απώλεια πληροφορίας (η τιμή στον αρχικού τύπου καθορίζεται μονοσήμαντα από την τιμή στον τελικού τύπου).
- ονομάζουμε περιορισμό τη μετατροπή μίας τιμής από έναν τύπο με ευρύτερο πεδίο τιμών σε έναν τύπο με στενότερο πεδίο τιμών, π.χ πραγματικού σε ακέραιο, πραγματικού διπλής ακρίβειας σε απλής ακρίβειας κ.λ.π. Σε μία τέτοια μετατροπή ενδέχεται να έχουμε απώλεια πληροφορίας, καθώς η τιμή στον τελικό τύπο αποτελεί προσέγγιση της τιμής στον αρχικό τύπο.
- δεν είναι πάντα δυνατή η μετατροπή ανάμεσα σε δύο τύπους

Έλεγχος τύπου ονομάζεται η διαδικασία επιβεβαίωσης ότι ο τύπος των ορίσματος ενός τελεστή (ή των πραγματικών παραμέτρων κατά την κλήση μιας συνάρτησης) είναι συμβατός με τον τελεστή (ή τις τυπικές παραμέτρους της συνάρτησης)

Ένας τύπος είναι συμβατός με έναν τελεστή αν είναι ένας από τους τύπους για τους οποίους είναι ορισμένος ο τελεστής, είτε αν μπορεί να μετατραπεί σε έναν τέτοιο τύπο.

- αν ο καθορισμός τύπου είναι στατικός τότε και οι περισσότεροι έλεγχοι τύπου μπορούν να γίνουν στατικά.
- αν ο καθορισμός τύπου είναι δυναμικός (γίνεται σε χρόνο εκτέλεσης) τότε και οι έλεγχοι τύπου πρέπει να γίνουν δυναμικά.

Υπερφόρτωση ενός τελεστή (ή ονόματος συνάρτησης) ονομάζουμε την ανάθεση σε αυτόν διαφορετικών λειτουργιών για διαφορετικούς τύπους ορισμάτων. Παραδείγματα:

- οι αριθμητικοί τελεστές δέχονται ορίσματα οποιουδήποτε αριθμητικού τύπου.
- ο τελεστής * στη C χρησιμοποιείται για πολλαπλασιασμό, αλλά και ως εναδικός τελεστής με όρισμα τύπου δείκτη.
- στη Fortran το = είναι τελεστής ανάθεσης αλλά και σύγκρισης
- στη Java το + χρησιμοποιείται και για παράθεση αλφαριθμητικών.

Σε πολλές περιπτώσεις αριθμητικές παραστάσεις περιέχουν σταθερές ή μεταβλητές διαφορετικών αριθμητικών τύπων.

Σε αυτή την περίπτωση γίνεται αυτόματη μετατροπή (διεύρυνση ή περιορισμός) ώστε και τα δύο ορίσματα κάθε τελεστή να έχουν τον ίδιο τύπο.

Υπάρχουν δύο προσεγγίσεις:

- πρὶν εφαρμοστεί ἕνας αριθμητικός τελεστής, γίνεται διεύρυνση του ορίσματος που ανήκει σε στενότερο τύπο, στον τύπο του άλλου ορίσματος. Αν η παράσταση πρέπει να δώσει τιμή συγκεκριμένου τύπου, τότε μετά την αποτίμησή της γίνεται η απαραίτητη μετατροπή.

Π.χ η δήλωση

```
float r = 5 / 2 + 2.0;
```

αρχικοποιεί την r με την τιμή 4.0 (η πρώτη διαίρεση είναι ακέραια και προκαλεί αποκοπή)

- όλες η τιμές που εμπλέκονται στην παράσταση μετατρέπονται στον τύπο που πρέπει να έχει η τιμή της παράστασης, πριν εφαρμοστεί οποιοσδήποτε τελεστής

Π.χ η δήλωση

```
float r = 5 / 2 + 2.0;
```

αρχικοποιεί την r με την τιμή 4.5 (οι τιμές 5 και 2 μετατρέπονται σε 5.0 και 2.0 πριν γίνει οποιαδήποτε πράξη, οπότε η πρώτη διαίρεση είναι πραγματική)

- 1 Τύποι δεδομένων
- 2 Μετατροπές τύπων
- 3 Ισοδυναμία τύπων

Ερώτημα: πότε δύο μεταβλητές έχουν ισοδύναμους τύπους;

- δύο μεταβλητές έχουν τύπους ισοδύναμους ως προς το όνομα αν έχουν δηλωθεί είτε στην ίδια δήλωση είτε σε διαφορετικές δηλώσεις που χρησιμοποιούν το ίδιο όνομα τύπου.
- δύο μεταβλητές έχουν τύπους ισοδύναμους ως προς τη δομή αν οι τύποι τους έχουν ακριβώς την ίδια δομή.

Παράδειγμα: Έστω οι παρακάτω δηλώσεις:

type

```
vector = array[1..100] of integer;
```

```
list = vector;
```

var

```
x,y:array[1..100] of integer;
```

```
z:array[1..100] of integer;
```

```
w:vector;
```

```
r:vector;
```

```
L:list;
```


- όλες οι μεταβλητές στην παραπάνω δήλωση έχουν ισοδύναμους τύπους ως προς τη δομή.
- υπάρχουν δύο ζεύγη μεταβλητών με ισοδύναμους τύπους ως προς όνομα:
 - x, y που δηλώνονται στη ίδια δήλωση
 - w, r που δηλώνονται με το ίδιο αναγνωριστικό τύπου
- η μεταβλητή L είναι ισοδύναμη με τις w, r , σύμφωνα με μία πιο ασθενή ισοδυναμία ως προς όνομα, η οποία θεωρεί τον τύπο `list` ως ψευδώνυμο του `vector` και όχι ως διαφορετικό τύπο

Συνήθως οι γλώσσες προγραμματισμού χρησιμοποιούν μία ενδιάμεση μορφή ισοδυναμίας τύπων.