

PASC400

Γενική Περιγραφή

Η γλώσσα PASC400 μοιάζει με τη γλώσσα υψηλού επιπέδου PASCAL, και ορίζεται στη συνέχεια. Η PASC400 είναι δομημένη με σύνθετες εντολές, οι οποίες είναι υποσύνολο των εντολών της κλασικής PASCAL. Η PASC400 υποστηρίζει βασικούς τύπους της PASCAL, ενώ από σύνθετους υποστηρίζει μόνο πίνακες και έναν τύπο ζεύγους. Η δομή των υποπρογραμμάτων διατηρείται όπως στην κλασική PASCAL, όπου όλα τα υποπρογράμματα ορίζονται μέσα από το πρόγραμμα, με τη δυνατότητα ορισμού φωλιασμένων υποπρογραμμάτων. Τέλος, η PASC400 δεν υποστηρίζει πυκνωμένες δομές, εκτός από τον τύπο ορμαθού χαρακτήρων (string).

Ειδική Περιγραφή

A. Λεκτικές Μονάδες

Οι λεκτικές μονάδες που αποτελούν και τα τερματικά σύμβολα της γραμματικής της PASC400 περιγράφονται στη συνέχεια. Σε παρένθεση – όπου χρειάζεται – δίνονται τα αντίστοιχα συμβολικά ονόματα που εμφανίζονται στη γραμματική της PASC400.

Μαζί με τις λεκτικές μονάδες δίνεται και η περιγραφή των σχολίων της PASC400, τα οποία όμως δεν εμφανίζονται στη γραμματική της γλώσσας.

Σημειώνεται ότι στη γλώσσα PASC400 δεν υπάρχει διάκριση μεταξύ κεφαλαίων και πεζών αλφαβητικών χαρακτήρων, εκτός αν αυτοί αποτελούν μέρος της λέξης μιας λεκτικής μονάδας CCONST ή SCONST.

Λέξεις-κλειδιά

Οι παρακάτω λέξεις που αποτελούν ανεξάρτητες λεκτικές μονάδες της PASC400:

PROGRAM CONST TYPE COUPLE ARRAY OF VAR FORWARD FUNCTION PROCEDURE
INTEGER REAL BOOLEAN CHAR STRING BEGIN END IF THEN ELSE CASE OTHERWISE
WHILE DO FOR DOWNTON TO READ WRITE LENGTH

Άλλες λέξεις-κλειδιά δίνονται πιο κάτω ως σταθερές ή τελεστές.

Αναγνωριστικά (ID)

Συμβολοσειρές που αρχίζουν με προαιρετικό χαρακτήρα ‘_’, ακολουθούμενο από αλφαβητικό χαρακτήρα, ακολουθούμενο από μηδέν ή περισσότερους αλφαριθμητικούς χαρακτήρες ή χαρακτήρες ‘_’, και δεν είναι λέξεις-κλειδιά. Ο χαρακτήρας ‘_’ δεν μπορεί να είναι ο τελευταίος χαρακτήρας του αναγνωριστικού.

Αποδεκτά παραδείγματα:

```
a100_version_2  
_a100_version2  
a100__version_2
```

Μη αποδεκτά παραδείγματα¹:

```
100__version_2  
a100__version2_  
_a100_version2_  
a100--version-2
```

Απλές σταθερές

¹ Σε όλα τα μη αποδεκτά παραδείγματα που δίνονται, η συμβολοσειρά δεν αναγνωρίζεται συνολικά, είναι όμως δυνατό να αναγνωρίζονται μέρη αυτής ως ανεξάρτητες λεκτικές μονάδες.

Μη προσημασμένες ακέραιες (ICONST):

Ο μοναδικός χαρακτήρας '0', που παριστάνει τη σταθερά με τιμή 0. Επίσης, ένας ή περισσότεροι αριθμητικοί χαρακτήρες, που ο πρώτος δεν είναι ο '0', οπότε η τιμή που παριστάνεται είναι ο αντίστοιχος αριθμός σε δεκαδική βάση. Επίσης, η συμβολοσειρά "0x" ακολουθούμενη από έναν ή περισσότερους αριθμητικούς χαρακτήρες που ο πρώτος δεν είναι ο '0', ή από έναν ή περισσότερους από τους αλφαβητικούς χαρακτήρες 'A', 'B', 'C', 'D', 'E' και 'F'. Στην περίπτωση αυτή, η τιμή που παριστάνεται είναι ο αντίστοιχος αριθμός – μετά το πρόθεμα "0x" – σε δεκαεξαδική βάση.

Αποδεκτά παραδείγματα:

0
180
0x9F0

Μη αποδεκτά παραδείγματα:

00
0180
xB7
0x0
00B10
0xG8A

Μη προσημασμένες πραγματικές (RCONST):

Μηδέν ή περισσότεροι αριθμητικοί χαρακτήρες που ακολουθούνται από το χαρακτήρα '.' και τουλάχιστον έναν αριθμητικό χαρακτήρα. Ακολουθεί προαιρετικά πεδίο εκθέτη, που ξεκινάει με τον χαρακτήρα 'e', ακολουθούμενο από προαιρετικό πρόσημο και τουλάχιστον έναν αριθμητικό χαρακτήρα. Εναλλακτικά, μη προσημασμένη ακέραια σταθερά που ακολουθείται υποχρεωτικά από πεδίο εκθέτη. Αν δεν υπάρχει πεδίο εκθέτη, ο αριθμός μπορεί να είναι σε δεκαεξαδική βάση, με πρόθεμα "0x". Το ακέραιο μέρος μιας πραγματικής σταθεράς, όπως και το αριθμητικό μέρος του εκθέτη, δε μπορεί να ξεκινά με '0' αν δεν είναι "0". Σε κάθε περίπτωση που υπάρχει κλασματικό μέρος, αυτό πρέπει να περιλαμβάνει τουλάχιστον ένα χαρακτήρα διαφορετικό από τον '0' και δε μπορεί να τελειώνει σε '0', εκτός αν είναι "0".

Αποδεκτά παραδείγματα:

180e-2
.5
180.1E+1
7.0
0xA.9
0X.00B9CF

Μη αποδεκτά παραδείγματα:

180E-2.2
.e-2
180.100
180E
.5G-2
5.
7.00
05.2E-05
0xBE-2
1100

Λογικές (BCONST):

Οι λέξεις-κλειδιά TRUE και FALSE.

Χαρακτήρες (CCONST):

Οποιοσδήποτε εκτυπώσιμος ASCII χαρακτήρας (κωδικοί 32-126) μεταξύ δύο εμφανίσεων του ειδικού χαρακτήρα ''. Επιπλέον, ειδικοί χαρακτήρες ASCII παριστάνονται με τη βοήθεια του χαρακτήρα '\'. Πιο συγκεκριμένα, ο χαρακτήρας LF (Line Feed) παριστάνεται ως '\n', ο

χαρακτήρας FF (Form Feed) ως '\f', ο χαρακτήρας HT (Horizontal Tab) ως '\t', ο χαρακτήρας CR (Carriage Return) ως '\r', ο χαρακτήρας BS (BackSpace) ως '\b' και ο χαρακτήρας VT (Vertical Tab) ως '\v'.

Αποδεκτά παραδείγματα:

```
'a'
'$'
' '
'''
'\n'
'\'
```

Μη αποδεκτά παραδείγματα:

```
'ac'
'\p'
'\\'
```

Ορμαθοί χαρακτήρων (SCONST):

Οποιαδήποτε συμβολοσειρά μεταξύ δύο εμφανίσεων του ειδικού χαρακτήρα '"', συμπεριλαμβανομένης της κενής συμβολοσειράς. Ο χαρακτήρας '"' και οι πιο πάνω ειδικοί χαρακτήρες ASCII παριστάνονται σε έναν ορμαθό χαρακτήρων με τη βοήθεια του χαρακτήρα '\'. Με κάθε άλλη χρήση του χαρακτήρα '\' παριστάνεται ο χαρακτήρας που ακολουθεί. Έτσι, ο χαρακτήρας '\' καθαυτός παριστάνεται ως \\. Ειδικά όταν ο χαρακτήρας '\' βρίσκεται στο τέλος της γραμμής, ο ορμαθός συνεχίζεται στην επόμενη γραμμή, χωρίς οι χαρακτήρες '\' και αλλαγής γραμμής να αποτελούν μέρος αυτού.

Αποδεκτά παραδείγματα:

```
"CHARACTER +"
""
"STRINGS START AND END WITH \""
"CHARACTER \\ AT THE END OF THE LINE \
EXTENDS STRING IN THE NEXT LINE\n"
```

Τελεστές

Τελεστές σύγκρισης (RELOP): > >= < <= <>

Προσθετικοί τελεστές (ADDOP): + -

Λογικό Η (OROP): OR

Πολλαπλασιαστικοί τελεστές (MULDIVANDOP): * / DIV MOD AND

Λογικό ΟΧΙ (NOTOP): NOT

Τελεστές στοιχείων ζεύγους (PAIROP): HEAD TAIL

Άλλες λεκτικές μονάδες

Άλλοι χαρακτήρες ή ακολουθίες χαρακτήρων που αποτελούν ανεξάρτητες λεκτικές μονάδες είναι:

'(' (LPAREN), ')' (RPAREN), ';' (SEMI), '.' (DOT), ',' (COMMA), '=' (EQU), ':' (COLON), '[' (LBRACK), ']' (RBRACK), "!=" (ASSIGN), ".." (DOTDOT), (EOF)

Σε ορισμένες περιπτώσεις κάποιες από τις παραπάνω λεκτικές μονάδες ενεργούν ως τελεστές, όπως αυτό θα φανεί στη γραμματική και σημασιολογία της PASC600.

Η λεκτική μονάδα EOF – που ως χαρακτήρας ορίζεται με ειδικό τρόπο από το κάθε σύστημα – δεν εμφανίζεται στη γραμματική της PASC600, αλλά πρέπει να παράγεται από το λεκτικό αναλυτή με τιμή 0 για τον τερματισμό της συντακτικής ανάλυσης.

Σχόλια

Τα σχόλια στην PASC600 είναι συμβολοσειρές που περικλείονται από το ζεύγος χαρακτήρων '{', '}'.

B. Συντακτικοί Κανόνες

Η γραμματική της PASC400 περιγράφεται από τους πιο κάτω κανόνες:

program \rightarrow header declarations subprograms comp_statement DOT

header \rightarrow PROGRAM ID SEMI

declarations \rightarrow constdefs typedefs vardefs

constdefs \rightarrow CONST constant_defs SEMI
| ϵ

constant_defs \rightarrow constant_defs SEMI ID EQU expression
| ID EQU expression

expression \rightarrow expression RELOP expression
| expression EQU expression
| expression OROP expression
| expression ADDOP expression
| expression MULDIVANDOP expression
| ADDOP expression
| NOTOP expression
| PAIROP expression
| variable
| ID LPAREN expressions RPAREN
| LENGTH LPAREN expression RPAREN
| constant
| LPAREN expression COLON expression RPAREN
| LPAREN expression RPAREN

variable \rightarrow ID
| variable LBRACK expressions RBRACK

expressions \rightarrow expressions COMMA expression
| expression

constant \rightarrow ICONST
| RCONST
| BCONST
| CCONST
| SCONST

typedefs \rightarrow TYPE type_defs SEMI
| ϵ

type_defs \rightarrow type_defs SEMI ID EQU type_def
| ID EQU type_def

type_def \rightarrow ARRAY LBRACK dims RBRACK OF typename
| COUPLE OF typename
| limit DOTDOT limit

dims \rightarrow dims COMMA limits
| limits

limits \rightarrow limit DOTDOT limit
| ID

```

limit → sign ICONST
      | CCONST
      | BCONST
      | ADDOP ID
      | ID

sign → ADDOP | ε

typename → standard_type
        | ID

standard_type → INTEGER | REAL | BOOLEAN | CHAR | STRING

vardefs → VAR variable_defs SEMI
        | ε

variable_defs → variable_defs SEMI identifiers COLON typename
              | identifiers COLON typename

identifiers → identifiers COMMA ID
            | ID

subprograms → subprograms subprogram SEMI
            | ε

subprogram → sub_header SEMI FORWARD
            | sub_header SEMI declarations subprograms comp_statement

sub_header → FUNCTION ID formal_parameters COLON typename
           | PROCEDURE ID formal_parameters
           | FUNCTION ID

formal_parameters → LPAREN parameter_list RPAREN
                 | ε

parameter_list → parameter_list SEMI pass identifiers COLON typename
              | pass identifiers COLON typename

pass → VAR | ε

comp_statement → BEGIN statements END

statements → statements SEMI statement
          | statement

statement → assignment
          | if_statement
          | while_statement
          | for_statement
          | subprogram_call
          | io_statement
          | comp_statement
          | ε

assignment → variable ASSIGN expression

if_statement → IF expression THEN statement if_tail

```

```

if_tail → ELSE statement
        | ε

while_statement → WHILE expression DO statement

for_statement → FOR ID ASSIGN iter_space DO statement

iter_space → expression TO expression
            | expression DOWNTO expression

subprogram_call → ID
                | ID LPAREN expressions RPAREN

io_statement → READ LPAREN read_list RPAREN
              | WRITE LPAREN write_list RPAREN

read_list → read_list COMMA read_item
           | read_item

read_item → variable

write_list → write_list COMMA write_item
           | write_item

write_item → expression

```

όπου το σύμβολο ‘|’ διαχωρίζει τα εναλλακτικά δεξιά μέλη των κανόνων και ε είναι η κενή συμβολοσειρά.

Οι παραπάνω κανόνες ορίζουν διφορούμενη γραμματική, που με κατάλληλους μετασχηματισμούς ή βοηθητικές περιγραφές προτεραιότητας και προσεταιριστικότητας τελεστών μπορεί να γίνει μη διφορούμενη.

Αρχικό σύμβολο της γραμματικής της PASC400 αποτελεί το “program”.

Γ. Σημασιολογία

Η σημασιολογία της PASC400 καθορίζεται από μια σειρά κανόνων που αφορούν τη δομή ενός *ορθού* προγράμματος. Κάποιοι από τους κανόνες αυτούς είναι πιθανό να καλύπτονται από τη σύνταξη της γλώσσας, ενώ κάποιοι άλλοι μπορούν να χρησιμοποιηθούν για τη μετατροπή της γραμματικής της PASC400 σε μη διφορούμενη. Όλοι οι υπόλοιποι κανόνες κωδικοποιούνται στο μεταγλωττιστή της PASC400 με τη βοήθεια σημασιολογικών ρουτινών που εκτελούνται κατά τη μετατροπή ενός αρχικού προγράμματος σε ενδιάμεσο κώδικα.

Τύποι δεδομένων

Η PASC400 υποστηρίζει τέσσερις βασικούς τύπους δεδομένων:

- integer : ο αριθμητικός τύπος των ακέραιων αριθμών,
- real : ο αριθμητικός τύπος των πραγματικών αριθμών,
- char : ο τύπος των ASCII χαρακτήρων, και
- boolean : ο τύπος των λογικών τιμών «Αληθής» και «Ψευδής».

Το μέγεθος και η αναπαράσταση των δεδομένων καθενός από τους βασικούς αριθμητικούς τύπους της PASC400 καθορίζονται από την τελική γλώσσα μεταγλώττισης. Αυτή επιτρέπει μεγέθη τύπων και αναπαραστάσεις ανάλογα με την υποστήριξη που δίνει η αρχιτεκτονική για τον καθένα. Η ευθυγράμμιση που απαιτείται για την προσπέλαση δεδομένων καθενός από τους τύπους αυτούς καθορίζεται επίσης από την τελική γλώσσα.

Τα δεδομένα του τύπου `char` καταλαμβάνουν τον ελάχιστο χώρο αποθήκευσης που επιτρέπει η τελική γλώσσα, συνήθως ένα `byte`. Η αναπαράστασή τους γίνεται με τον κώδικα ASCII. Όσο αφορά τον τύπο `boolean`, το μέγεθος των δεδομένων του είναι επίσης το ελάχιστο που επιτρέπει η τελική γλώσσα, ενώ η αναπαράστασή τους γίνεται με τη σταθερά 0 για την τιμή «Ψευδής» και τη σταθερά 1 για την τιμή «Αληθής».

Η κωδικοποίηση των δύο παραπάνω τύπων προσδίδει μια φυσική διάταξη στις τιμές τους, και γι' αυτό θεωρούνται *διατεταγμένοι* τύποι.

Εκτός από τους πιο πάνω βασικούς, η PASC400 υποστηρίζει και τέσσερεις ακόμα τύπους.

Ο τύπος υποπεριοχής είναι ένας βαθμωτός τύπος που ορίζεται ως υποσύνολο ενός από τους υπόλοιπους βαθμωτούς τύπους, εκτός από τύπο `real`.

Παραδείγματα υποπεριοχής είναι τα παρακάτω:

```
-128 .. 127
'A' .. 'F'
```

όπου η πρώτη υποπεριοχή είναι υποσύνολο του τύπου `integer`, ενώ η δεύτερη υποπεριοχή είναι υποσύνολο του τύπου `char`.

Οι τιμές μεταβλητών του τύπου υποπεριοχής λαμβάνονται μέσα από το αντίστοιχο υποσύνολο και κωδικοποιούνται όπως οι τιμές του τύπου από τον οποίο παράγονται.

Ο τύπος `string` είναι ο τύπος των ορμαθών χαρακτήρων. Συμμετέχει στη γραμματική της PASC400 ως βασικός, γι' αυτό και στη συνέχεια θα συμπεριλαμβάνεται σε αυτούς. Σε αντίθεση όμως με τους πιο πάνω τύπους, ο τύπος `string` παράγεται από τον τύπο `char`, καθώς οι τιμές του προκύπτουν σαν ένα διάνυσμα τιμών τύπου `char`, με την τελευταία να είναι πάντα 0. Ο τύπος `string` είναι διατεταγμένος με την λεξικογραφική σειρά των ορμαθών χαρακτήρων που παριστάνονται, με βάση τον κώδικα ASCII των επιμέρους χαρακτήρων. Η αποθήκευση των χαρακτήρων ενός ορμαθού γίνεται σε συνεχόμενες θέσεις μνήμης, μέσα σε χώρο μεγέθους 256 φορές το μέγεθος του τύπου `char`. Αυτό σημαίνει ότι το μήκος ενός ορμαθού – συμπεριλαμβανομένου του τερματικού 0 – δε μπορεί να ξεπερνάει το 256.

Η PASC400 υποστηρίζει και δύο σύνθετους τύπους, τον τύπο ζεύγους και τον τύπο πίνακα, με στοιχεία δεδομένα του ίδιου τύπου. Οι τύποι στοιχείων ζεύγους πρέπει να είναι βασικοί βαθμωτοί, ενώ οι τύποι στοιχείων πίνακα δεν είναι απαραίτητο να είναι βασικοί.

Αναφορά σε ζεύγος μπορεί να γίνει συνολικά ή και ξεχωριστά στα δύο στοιχεία του. Η συνολική αναφορά μπορεί να γίνει ως αναφορά σε μεταβλητή αυτού του τύπου, ή ως σύνθεση των δύο στοιχείων, με μια έκφραση παρενθέσεων, όπου τα στοιχεία δίνονται ως εκφράσεις που διαχωρίζονται με το σύμβολο `'.'`.

Παράδειγμα συνολικής αναφορά σε ζεύγος με σύνθεση στοιχείων είναι:

```
(5:3)
```

Όταν τα στοιχεία γράφονται ως εκφράσεις, ακολουθούνται οι κανόνες εκφράσεων που δίνονται πιο κάτω. Παράδειγμα τέτοιας αναφορά είναι το εξής:

```
(x-2:-1)
```

Όπου η μεταβλητή `x` πρέπει να είναι τύπου συμβατού με την αφαίρεση, και η έκφραση `x-2` να είναι συμβατού τύπου με την έκφραση `-1`.

Η ξεχωριστή αναφορά στα στοιχεία του ζεύγους γίνεται μέσω των ειδικών τελεστών `head` και `tail` που επεξηγούνται αργότερα.

Ένα στοιχείο πίνακα αναπαριστάται με το όνομα του πίνακα και μέσα σε αγκύλες μια λίστα από δείκτες, που αντιστοιχούν στις διαστάσεις του πίνακα. Δείκτες διαδοχικών διαστάσεων διαχωρίζονται με το σύμβολο `'.'`.

Παραδείγματα αναφοράς σε στοιχεία πίνακα είναι τα εξής:

```
C[103,5]
C[0,5,'k']
C[-1,false]
```

όπου false είναι σταθερά τύπου boolean. Ο τύπος του δείκτη πρέπει να είναι βαθμωτός εκτός από real, με όρια τιμών που ορίζονται κατά τη δήλωση του πίνακα. Έτσι, η παρακάτω αναφορά σε στοιχείο πίνακα δεν είναι αποδεκτή:

C[3.5]

Δεικτοδότηση με τη βοήθεια αναγνωριστικού επιτρέπεται μόνο αν ο τύπος δείκτη που προκύπτει είναι βαθμωτός εκτός από real. Έτσι, η αναφορά:

C[i, j+2]

δε θα είναι αποδεκτή, αν ο τύπος του αναγνωριστικού i και της έκφρασης “j+2” είναι μη βαθμωτός ή real.

Οι τύποι πίνακα μπορούν να συνδυάζονται μεταξύ τους, έτσι ώστε να δημιουργούνται πίνακες πινάκων. Επίσης επιτρέπονται και πίνακες ζευγών. Συνολικά, φωλιάσματα τύπων επιτρέπονται μέχρι ένα μέγιστο βάθος 5 τύπων.

Η αποθήκευση των στοιχείων ενός πίνακα της PASC400 στη μνήμη γίνεται σε συνεχόμενες θέσεις, ανάλογα με την ευθυγράμμιση του τύπου αυτών, με αύξουσα σειρά μεταβολής των δεικτών από την τελευταία διάσταση προς την πρώτη. Για παράδειγμα, ένας τρισδιάστατος πίνακας A 3×3×2 στοιχείων με τιμές δεικτών για κάθε διάστασή του στις περιοχές ακεραίων από -1 έως 1, από 0 έως 2 και από 0 έως 1, αντίστοιχα, αποθηκεύει τα στοιχεία του με τη σειρά: A[-1,0,0], A[-1,0,1], A[-1,1,0], A[-1,1,1], A[-1,2,0], A[-1,2,1], A[0,0,0], A[0,0,1], A[0,1,0], A[0,1,1], A[0,2,0], A[0,2,1], A[1,0,0], A[1,0,1], A[1,1,0], A[1,1,1], A[1,2,0] και A[1,2,1]. Με τον τρόπο αυτό, ένας δισδιάστατος πίνακας αποθηκεύεται γραμμή-γραμμή. Ένας σύνθετος τύπος πίνακα καταλαμβάνει τόσες θέσεις αποθήκευσης, όσο είναι το πλήθος των στοιχείων του επί το μέγεθος του ευθυγραμμισμένου στοιχείου του.

Δομή ενός προγράμματος PASC400

Ένα πρόγραμμα σε γλώσσα PASC400 αποτελείται από την κύρια μονάδα και μονάδες υποπρογραμμάτων. Οι μονάδες υποπρογραμμάτων δηλώνονται μέσα στην κύρια μονάδα, ή φωλιασμένες σε άλλες μονάδες υποπρογραμμάτων. Οι μονάδες υποπρογραμμάτων έχουν την ίδια δομή με την κύρια μονάδα.

Κάθε μονάδα περιέχει:

- Επικεφαλίδα: Η επικεφαλίδα της μονάδας καθορίζει ανάμεσα στα άλλα τις παραμέτρους της, δηλαδή τον τρόπο επικοινωνίας αυτής με τις άλλες μονάδες του προγράμματος.
- Δηλώσεις σταθερών, τύπων και μεταβλητών (προαιρετικά): Οι δηλώσεις αυτές έχουν εμβέλεια τη μονάδα.
- Φωλιασμένες δηλώσεις υποπρογραμμάτων (προαιρετικά): Και αυτές οι δηλώσεις έχουν εμβέλεια τη μονάδα, αλλά επιπλέον ορίζουν εσωτερικές εμβέλειες σε αυτήν.
- Μια σύνθετη εντολή: Αυτή είναι μια σειρά απλών, δομημένων ή άλλων σύνθετων εντολών μεταξύ των λέξεων-κλειδιά “begin” και “end”. Η σειρά αυτή μπορεί να είναι κενή, όπως φαίνεται και στους συντακτικούς κανόνες της γλώσσας. Η σύνθετη εντολή της PASC400 δεν περιέχει δηλώσεις.

Η εκτέλεση του κώδικα μιας μονάδας ξεκινάει με την πρώτη και τερματίζεται με την τελευταία εντολή της σύνθετης εντολής της μονάδας.

Επικεφαλίδα

Η επικεφαλίδα της κύριας μονάδας περιλαμβάνει τη λέξη-κλειδί “program”, για να τη διαχωρίσει από τις επικεφαλίδες υποπρογραμμάτων, και το όνομά της, που είναι και το πρώτο αναγνωριστικό καθολικής εμβέλειας του προγράμματος.

Αν και στην κλασική PASCAL η κύρια μονάδα έχει παραμέτρους που καθορίζουν την επικοινωνία αυτής με το λειτουργικό σύστημα, στην PASC400 η κύρια μονάδα δεν έχει παραμέτρους.

Πληροφορίες για επικεφαλίδες υποπρογραμμάτων δίνονται σε επόμενη παράγραφο.

Δηλώσεις σταθερών, τύπων και μεταβλητών

Με μια τέτοια δήλωση αποδίδεται σε ένα αναγνωριστικό μια σταθερά, ένας τύπος ή μια μεταβλητή του προγράμματος. Κάθε τέτοια δήλωση εισάγει το αναγνωριστικό στον πίνακα συμβόλων, με εμβέλεια την τρέχουσα μονάδα. Ένα αναγνωριστικό μπορεί να δηλωθεί ξανά, πιθανά με άλλο τρόπο, σε εσωτερική εμβέλεια της μονάδας, οπότε η νέα δήλωση επισκιάζει την παλιά. Ένα αναγνωριστικό δε μπορεί να δηλωθεί για δεύτερη φορά στην ίδια εμβέλεια.

Μέσα στην ίδια μονάδα οι δηλώσεις σταθερών προηγούνται των υπολοίπων, ενώ οι δηλώσεις τύπων προηγούνται των δηλώσεων μεταβλητών. Οι δηλώσεις σταθερών, τύπων και μεταβλητών διαχωρίζονται μεταξύ τους, αλλά και με τη σύνθετη εντολή της μονάδας – αν δεν ακολουθούν δηλώσεις υποπρογραμμάτων, με το χαρακτήρα ‘;’.

Δηλώσεις σταθερών

Μια δήλωση σταθεράς στην PASC400 αποδίδει σε ένα αναγνωριστικό μια σταθερή τιμή, μια τιμή δηλαδή που δεν επιτρέπεται να αλλάξει με κάποια ανάθεση μέσα στην εμβέλεια στην οποία υπάρχει η δήλωση. Η σταθερή τιμή μπορεί να δίνεται είτε ως μία λεκτική μονάδα σταθεράς είτε ως έκφραση.

Ο τύπος της σταθεράς πρέπει να είναι βασικός ή ζεύγους, αλλά δεν ορίζεται ρητά μέσα από τη δήλωση. Έτσι, ο σημασιολογικός αναλυτής πρέπει να βρει και να αποδώσει τον κατάλληλο τύπο στο αναγνωριστικό της σταθεράς.

Παραδείγματα δηλώσεων σταθερών είναι τα παρακάτω:

```
const a = 10; b = -a * 3; s = "sTrInG";
      x = -3.0; y = b * x; z = true; w = ('0':'1');
```

Ο σημασιολογικός αναλυτής πρέπει να αποδώσει τον τύπο integer στα αναγνωριστικά a και b, τον τύπο real στα αναγνωριστικά x και y, τον τύπο boolean στο αναγνωριστικό z, τον τύπο couple of char στο αναγνωριστικό w και τον τύπο string στο αναγνωριστικό s.

Ας σημειωθεί ότι η λέξη-κλειδί “const” εμφανίζεται μόνο στην πρώτη από τις δηλώσεις σταθερών μιας μονάδας.

Σε περίπτωση που η σταθερά ορίζεται με τη βοήθεια έκφρασης, πρέπει να ακολουθούνται οι κανόνες που διέπουν τις εκφράσεις, όπως αυτοί περιγράφονται πιο κάτω. Επιπλέον, η έκφραση πρέπει να δίνει σταθερή τιμή, κάτι που θα συμβαίνει αν σε αυτή συμμετέχουν μόνο σταθερές, είτε ως λεκτικές μονάδες σταθερών, είτε ως άλλες δηλωμένες σταθερές, ορατές στο σημείο της παρούσας δήλωσης. Τη σταθερή τιμή της έκφρασης πρέπει να προσδιορίζει ο σημασιολογικός αναλυτής και να αποδίδει στο αναγνωριστικό της νέας σταθεράς.

Δηλώσεις τύπων

Μια δήλωση τύπου στην PASC400 αποδίδει σε ένα αναγνωριστικό τη δομή που περιγράφει έναν τύπο υποπεριοχής ή ένα σύνθετο τύπο.

Η PASC400 επιβάλλει στον προγραμματιστή να δηλώνει όλους τους μη βασικούς τύπους που χρησιμοποιεί. Μετά από μια δήλωση, ο τύπος θεωρείται γνωστός, κι επομένως το όνομά του μπορεί να χρησιμοποιηθεί σε επόμενη δήλωση. Επειδή ένα όνομα τύπου δε μπορεί να χρησιμοποιηθεί πριν τη δήλωσή του, δεν υλοποιούνται κυκλικές δομές δεδομένων.

Παραδείγματα δηλώσεων τύπων είναι τα παρακάτω:

```
type ww = -5..5;
      ar = array [-1..M, ww] of boolean;
      cl = couple of ww;
```

όπου M είναι σταθερά που έχει δηλωθεί νωρίτερα.

Όπως και με τις δηλώσεις σταθερών, η λέξη-κλειδί “type” εμφανίζεται μόνο στην πρώτη από τις δηλώσεις τύπων μιας μονάδας.

Πιο συγκεκριμένα, για δηλώσεις τύπων υποπεριοχής:

1. Τα όρια μιας υποπεριοχής ορίζονται με σταθερές ή αναγνωριστικά σταθερών που έχουν νωρίτερα δηλωθεί και είναι ορατά στο σημείο της δήλωσης της υποπεριοχής, και διαχωρίζονται μεταξύ τους με τη λεκτική μονάδα “..”. Το κάτω όριο τοποθετείται πριν, και το άνω όριο τοποθετείται μετά τη μονάδα διαχωρισμού.
2. Το κάτω όριο μιας υποπεριοχής πρέπει να έχει τιμή μικρότερη από ή ίση με την τιμή του πάνω ορίου.
3. Ένα όριο υποπεριοχής μπορεί να είναι προσημασμένο, αρκεί ο τύπος της σταθεράς ή του αναγνωριστικού που τον ορίζει να δέχεται πρόσημο, δηλαδή να μην είναι boolean, char ή απαρίθμησης.

Στις δηλώσεις τύπων ζεύγους αναγράφεται ο τύπος των στοιχείων, που πρέπει να είναι βασικός ή υποπεριοχής, μετά από τις λέξεις-κλειδιά “couple of”.

Για δηλώσεις τύπων πίνακα:

1. Οι δυνατές τιμές των δεικτών ενός πίνακα για κάθε διάστασή του δίνονται σε αγκύλες, μετά τη λέξη-κλειδί “array”. Οι τιμές δεικτών διαδοχικών διαστάσεων διαχωρίζονται μεταξύ τους με το χαρακτήρα ‘,’.
2. Οι τιμές που μπορεί να πάρει ο δείκτης μιας διάστασης του πίνακα δίνονται με μια περιγραφή ακέραιας υποπεριοχής, όπου ως ακέραια νοείται η υποπεριοχή τιμών τύπου integer, boolean ή char. Εναλλακτικά μπορούν να δοθούν με έναν τύπο ακέραιας υποπεριοχής που έχει δηλωθεί νωρίτερα.
3. Ο τύπος των στοιχείων του πίνακα δίνεται με όνομα τύπου, ο οποίος πρέπει να είναι βασικός ή να έχει οριστεί νωρίτερα, και αναγράφεται μετά τη λέξη-κλειδί “of”.

Δηλώσεις μεταβλητών

Μια δήλωση μεταβλητής αποδίδει σε ένα αναγνωριστικό μια μεταβλητή της μονάδας, ενός βασικού τύπου ή τύπου που έχει δηλωθεί νωρίτερα.

Παραδείγματα δηλώσεων μεταβλητών είναι τα παρακάτω:

```
var i,j,k: integer; s: string;
    l: boolean; dev: real; x: char; w: integer;
    c: complex; a: c_array;
```

όπου complex και c_array είναι τύποι που έχουν δηλωθεί νωρίτερα, και η ορατότητά τους περιλαμβάνει την παρούσα μονάδα. Όπως και με τις δηλώσεις σταθερών και τύπων, η λέξη-κλειδί “var” εμφανίζεται μόνο στην πρώτη από τις δηλώσεις μεταβλητών μιας μονάδας.

Κάθε δήλωση γίνεται για ένα μόνο τύπο. Περισσότερες δηλώσεις για τον ίδιο τύπο μπορούν να ακολουθούν πιο κάτω.

Όλες οι μεταβλητές της PASC400 – εκτός από τις μεταβλητές της κύριας μονάδας του προγράμματος – είναι μεταβλητές στοίβας. Οι μεταβλητές της κύριας μονάδας είναι στατικές καθολικής εμβέλειας, κι επομένως δεν τοποθετούνται στη στοίβα, αλλά έχουν σταθερές διευθύνσεις έξω από αυτή, έχουν δε διάρκεια ζωής όλη τη διάρκεια εκτέλεσης του προγράμματος.

Δηλώσεις υποπρογραμμάτων

Υποπρογράμματα στην PASC400 μπορούν να δηλώνονται μέσα σε οποιαδήποτε μονάδα του προγράμματος, αμέσως μετά τις δηλώσεις σταθερών, τύπων και μεταβλητών. Έχουν δε εμβέλεια παρόμοια με αυτή των υπόλοιπων δηλώσεων της μονάδας και ο έλεγχος της δήλωσης του ονόματος ενός υποπρογράμματος είναι παρόμοιος με τον έλεγχο δηλώσεων των σταθερών, τύπων και μεταβλητών της PASC400. Οι δηλώσεις υποπρογραμμάτων διαχωρίζονται μεταξύ τους, αλλά και με τη σύνθετη εντολή της μονάδας, με το χαρακτήρα ‘;’.

Τα υποπρογράμματα της PASC400 διακρίνονται σε διαδικασίες (procedures) και συναρτήσεις (functions). Οι διαδικασίες καλούνται μέσα από ειδική εντολή της PASC400, ενώ οι συναρτήσεις μέσα από εκφράσεις. Οι συναρτήσεις επιστρέφουν αποτελέσματα μέσα από τα ονόματά τους που για το σκοπό αυτό λειτουργούν και σαν τοπικές μεταβλητές τους.

Κάθε υποπρόγραμμα είναι μια μονάδα, και η δήλωσή του περιέχει ολόκληρη τη δομή του, σύμφωνα με τη δομή μονάδας που περιγράφηκε πιο πάνω. Ειδικά όσο αφορά την επικεφαλίδα ενός υποπρογράμματος της PASC400, αυτή περιέχει με τη σειρά:

- (α) το είδος του υποπρογράμματος,
- (β) το όνομά του, που πρέπει να είναι μοναδικό στην εμβέλεια στην οποία βρίσκεται η δήλωση,
- (γ) τις τυπικές παραμέτρους του και τον τρόπο περάσματος αυτών μέσα σε παρενθέσεις, και
- (δ) τον τύπο του αποτελέσματος, εάν το υποπρόγραμμα είναι συνάρτηση.

Παραδείγματα επικεφαλίδων υποπρογραμμάτων είναι τα εξής:

```
function LL (y: real; var w: c_array): real;
procedure A (m,n: integer; x: x_type; var z: char);
procedure S;
```

Ειδικότερα:

1. Η δήλωση μιας τυπικής παραμέτρου περιλαμβάνει το όνομα του τύπου της, ο οποίος μπορεί να είναι βασικός ή να έχει δηλωθεί νωρίτερα.
2. Το πέραςμα μιας παραμέτρου γίνεται κατ' αναφορά, εάν της δήλωσης της παραμέτρου προηγείται η λέξη-κλειδί "var", διαφορετικά γίνεται κατ' αξία.
3. Αν ο τύπος μιας τυπικής παραμέτρου είναι string ή σύνθετος τύπος πίνακα, τότε η παράμετρος πρέπει να μεταδίδεται κατ' αναφορά.
4. Κάθε τυπική παράμετρος που μεταδίδεται κατ' αξία αποτελεί τοπική μεταβλητή της μονάδας του υποπρογράμματος και αποθηκεύεται στη στοίβα.
5. Μια διαδικασία μπορεί να μην έχει παραμέτρους, μια συνάρτηση όμως πρέπει να έχει τουλάχιστον μία παράμετρο.
6. Ο τύπος του αποτελέσματος μιας συνάρτησης αναγράφεται μετά το χαρακτήρα ':' και πρέπει να είναι βασικός βαθμωτός ή τύπος ζεύγους.

Μια επικεφαλίδα μονάδας διαχωρίζεται από τις δηλώσεις ή τη σύνθετη εντολή της – αν δεν υπάρχουν δηλώσεις – με το χαρακτήρα ';'.

Σε αντίθεση με τις δηλώσεις τύπων, όπου ένα όνομα τύπου χρησιμοποιείται μετά τη δήλωσή του, σε μια δήλωση υποπρογράμματος μπορεί να χρησιμοποιηθεί το ίδιο το υποπρόγραμμα. Αυτό, σε συνδυασμό με το γεγονός ότι οι τοπικές μεταβλητές ενός υποπρογράμματος είναι μεταβλητές στοίβας, επιτρέπει στην PASC400 την υποστήριξη αναδρομής.

Για μεγαλύτερη ευελιξία και συγκεκριμένα για την υποστήριξη έμμεσης αναδρομής, η PASC400 – σύμφωνα με την κλασική PASCAL – επιτρέπει προαναγγελία σε μια δήλωση υποπρογράμματος, ως εξής:

1. Ένα υποπρόγραμμα – διαδικασία ή συνάρτηση – μπορεί να δηλωθεί με τη λέξη-κλειδί "forward", η οποία ακολουθεί την επικεφαλίδα και αναβάλλει την υπόλοιπη δήλωση γι' αργότερα.
2. Στο σημείο που επιθυμεί ο προγραμματιστής, το οποίο θα πρέπει να βρίσκεται στην ίδια μονάδα και να ακολουθεί την προηγούμενη δήλωση, συμπληρώνεται η δήλωση του υποπρογράμματος. Για το σκοπό αυτό, ξαναδίνεται μια σύντομη επικεφαλίδα, η οποία περιλαμβάνει μόνο το είδος και το όνομα του υποπρογράμματος.

Ένα παράδειγμα δήλωσης συνάρτησης με προαναγγελία είναι το πιο κάτω απόσπασμα από το χώρο δηλώσεων υποπρογραμμάτων μιας μονάδας:

```
function X (n:integer) : boolean; forward;
```

```

... { ένα ή περισσότερα άλλα υποπρογράμματα που πιθανά καλούν την X }
function X;
{ η υπόλοιπη δήλωση της X }

```

Ο σημασιολογικός αναλυτής πρέπει να εξετάζει αν οι προαναγγελίες δηλώσεων υποπρογραμμάτων γίνονται σύμφωνα με τα παραπάνω, αν δηλαδή (α) μετά από μια προαναγγελία υπάρχει όντως η συνέχεια της δήλωσης, και (β) αν μια δήλωση υποπρογράμματος με σύντομη επικεφαλίδα αντιστοιχεί σε προηγούμενη προαναγγελία.

Σύμφωνα με τη σύνταξη της PASC400 επιτρέπονται φωλιασμένες δηλώσεις υποπρογραμμάτων. Επειδή κάθε υποπρόγραμμα της PASC400 ορίζει μια εμβέλεια, κάθε αναφορά σε ένα αναγνωριστικό που δεν είναι τοπικό στην μονάδα στην οποία βρίσκεται η αναφορά, πρέπει να επιλύεται σε άλλη μονάδα. Καθώς το δέσιμο αναγνωριστικών στην PASC400 είναι στατικό, η επίλυση αναφορών γίνεται ακολουθώντας προς τα έξω τις φωλιασμένες μονάδες του προγράμματος.

Το μέγιστο βάθος φωλιάσματος υποπρογραμμάτων στην PASC400 είναι 10.

Εκφράσεις

Η PASC400 υποστηρίζει αριθμητικές και λογικές εκφράσεις. Οι τιμές των πρώτων μπορούν να είναι τύπου integer ή real, ενώ οι τιμές των δεύτερων μπορούν να είναι μόνο τύπου boolean. Οι εκφράσεις αποτιμώνται με βάση συγκεκριμένους κανόνες και με τη βοήθεια των τελεστών της PASC400. Οι αριθμητικές εκφράσεις μπορούν να περιέχονται σε λογικές εκφράσεις, αλλά όχι το αντίστροφο.

Εκτός από τις πιο πάνω, ορίζονται και απλές εκφράσεις τύπου char, string και υποπεριοχής, οι οποίες είτε εμφανίζονται μεμονωμένα σε κλήσεις υποπρογραμμάτων και ορισμένες εντολές της PASC400, είτε περιέχονται σε λογικές εκφράσεις. Ειδικότερα, εκφράσεις υποπεριοχής ακεραίων, και σε κάποιες περιπτώσεις και τύπου string, μπορούν να συμμετέχουν και σε σύνθετες εκφράσεις. Τέλος, ορίζονται και απλές εκφράσεις σύνθετου τύπου ζεύγους ή πίνακα, που εμφανίζονται μόνο σε κλήσεις υποπρογραμμάτων και αναθέσεις, όπως θα δούμε παρακάτω. Εκφράσεις μεταβλητών ζεύγους επιτρέπονται και σε εκφράσεις όπου ενεργούν συγκεκριμένοι τελεστές.

Μια έκφραση της PASC400 περιλαμβάνει:

- Τιμές αριστερής προσπέλασης (L-values), δηλαδή διευθύνσεις του χώρου δεδομένων του προγράμματος που αντιστοιχούν σε μεταβλητές ή στοιχεία πινάκων.
- Σταθερές, είτε σαν τιμές που υπάρχουν στον αρχικό κώδικα της μονάδας, είτε σαν τιμές ονομάτων σταθερών, ορατών στη μονάδα.
- Τελεστές, που επιτρέπουν πράξεις μεταξύ υποεκφράσεων.
- Κλήσεις συναρτήσεων, που έχουν σαν αποτέλεσμα την αντικατάσταση της συνάρτησης στην έκφραση με την τιμή που αυτή επιστρέφει.

Τα τρία τελευταία αντικείμενα ορίζουν τιμές δεξιάς προσπέλασης (R-values).

Παραδείγματα εκφράσεων της PASC400 είναι τα εξής:

```

i
a+1
a[3,i+1,j-1]*wk*2 + (tu(b,z[2]) div (i-1))*i
(a+b[j-i][k-2] > 0) and not (x < "katerina")
z[i]+ head a(i)/tail b(i)

```

Τιμές αριστερής προσπέλασης

Κάθε διεύθυνση στο χώρο δεδομένων του προγράμματος ονομάζεται τιμή αριστερής προσπέλασης, επειδή μπορεί να βρεθεί στο αριστερό μέλος μιας ανάθεσης. Τέτοιες τιμές στην PASC400 αποτελούν οι μεταβλητές και τα στοιχεία πινάκων.

Όταν μια τιμή αριστερής προσπέλασης βρεθεί σε μια έκφραση, αποτιμάται, και η τιμή που αποδίδεται γι' αυτήν είναι η τιμή που περιέχεται στη διεύθυνση που αυτή παριστάνει, δηλαδή η τιμή της μεταβλητής ή του στοιχείου πίνακα. Απαραίτητη προϋπόθεση για αποτίμηση είναι η προηγούμενη δήλωση της αντίστοιχης μεταβλητής ή του αντίστοιχου πίνακα, στην ίδια εμβέλεια ή σε κάποια εξωτερική της. Σε περίπτωση απουσίας τέτοιας δήλωσης υπάρχει σημασιολογικό σφάλμα.

Αν η τιμή αριστερής προσπέλασης αντιστοιχεί σε σύνθετο τύπο πίνακα, είναι δυνατό να μην ακολουθείται από δεικτοδότηση, μόνο εάν αυτή αποτελεί πραγματική παράμετρο στην κλήση ενός υποπρογράμματος, ή αν συμμετέχει σε ανάθεση πινάκων.

Τέλος, μια μεταβλητή τύπου ζεύγους μπορεί να εμφανιστεί σε μια έκφραση χωρίς αναφορά στα στοιχεία της, μόνο αν αποτελεί πραγματική παράμετρο στην κλήση υποπρογράμματος ή αν αποτελεί τιμή ανάθεσης σε άλλη μεταβλητή ίδιου τύπου.

Η αποτίμηση μεταβλητής που έχει δηλωθεί κανονικά σε μια μονάδα γίνεται με ανάγνωση της τιμής της μεταβλητής από το χώρο δεδομένων της μονάδας.

Η αποτίμηση στοιχείου πίνακα που επίσης έχει δηλωθεί κανονικά μπορεί να γίνει, εφ' όσον οι τιμές των εκφράσεων που αποδίδονται στους δείκτες του στοιχείου αυτού είναι μέσα στα δηλωμένα όρια για τις αντίστοιχες διαστάσεις του πίνακα. Μ' άλλα λόγια, στην αποτίμηση στοιχείου πίνακα προηγείται η αποτίμηση των δεικτών του. Εάν οι τιμές των δεικτών είναι αποδεκτές, υπολογίζεται η διεύθυνση του ζητούμενου στοιχείου, λαμβάνοντας υπ' όψη την αρχική διεύθυνση του πίνακα στο χώρο δεδομένων της μονάδας, τον τρόπο αποθήκευσης των στοιχείων του πίνακα, τις διαστάσεις, τα όρια υποπεριοχής σε κάθε διάσταση, καθώς και το μέγεθος του στοιχείου του. Στη συνέχεια μπορεί να αποτιμηθεί το στοιχείο αυτό, με ανάγνωσή του από το χώρο δεδομένων.

Η PASC400 έχει ισχυρό σύστημα τύπων, κι επομένως ο πιο πάνω έλεγχος τιμών των δεικτών πρέπει να γίνεται από τον κώδικα του προγράμματος. Ο μεταγλωττιστής, δηλαδή, πρέπει για κάθε προσπέλαση πίνακα να παράγει κώδικα που να κάνει αυτόν τον έλεγχο.

Η αποτίμηση μεταβλητής που δεν είναι δηλωμένη στην παρούσα μονάδα, αλλά σε κάποια εξωτερική της, πρέπει να γίνεται στο χώρο δεδομένων εκείνης της μονάδας. Επειδή η PASC400 υποστηρίζει στατικό δέσιμο, η διεύθυνση της μεταβλητής στο χώρο δεδομένων της εξωτερικής μονάδας βρίσκεται με τη βοήθεια συνδέσμων προσπέλασης ή πινάκων δεικτών. Ο μεταγλωττιστής πρέπει να παράγει κώδικα, ο οποίος να υπολογίζει τη διεύθυνση αυτή, και στη συνέχεια να προχωράει στην αποτίμηση της μεταβλητής.

Οι τυπικές παράμετροι ενός υποπρογράμματος είναι τιμές αριστερής προσπέλασης στο δυναμικό χώρο δεδομένων του υποπρογράμματος στη στοίβα, αν μεταδίδονται κατ' αξία, και στον υπόλοιπο χώρο δεδομένων, αν μεταδίδονται κατ' αναφορά. Στη δεύτερη περίπτωση, και σε κάθε κλήση του υποπρογράμματος, οι αντίστοιχες πραγματικές παράμετροι πρέπει να είναι τιμές αριστερής προσπέλασης, που δεν αποτιμώνται, αλλά περνάνε στο υποπρόγραμμα με τη διεύθυνσή τους. Η αποτίμηση μιας τυπικής παραμέτρου που μεταδίδεται κατ' αναφορά γίνεται σε εκφράσεις που αυτή εμφανίζεται, με ανάγνωση από τη διεύθυνσή της. Ειδικά αν η τυπική παράμετρος είναι σύνθετου τύπου πίνακα, οπότε και μεταδίδεται κατ' αναφορά, η προσπέλαση κάποιου στοιχείου της γίνεται όπως αναφέρθηκε παραπάνω.

Σταθερές

Οι σταθερές της PASC400 είναι αυτές που αναγνωρίζονται άμεσα από το λεκτικό αναλυτή και περιγράφηκαν στην αντίστοιχη ενότητα, καθώς και σταθερές που έχουν δηλωθεί με αναγνωριστικά σε δήλωση σταθερών, και είναι ορατές στην παρούσα μονάδα.

Ειδικά όσο αφορά τις πρώτες:

Οι τιμές των αριθμητικών σταθερών προκύπτουν άμεσα με μετατροπή των αντίστοιχων λέξεων σε αριθμητικές τιμές. Οι τιμές των λογικών σταθερών, που συμβολίζονται με τις λέξεις “true” και “false” για «Αληθής» και «Ψευδής» αντίστοιχα, αποδίδονται με την κωδικοποίηση που αναφέρθηκε παραπάνω. Οι τιμές των σταθερών χαρακτήρων, καθώς και των ορμαθών χαρακτήρων, αποδίδονται με την κωδικοποίηση ASCII.

Οι αριθμητικές σταθερές δεν έχουν πρόσημο, και προσημασμένοι αριθμοί προκύπτουν από εκφράσεις με τη βοήθεια του τελεστή προσήμου ADDOP.

Τελεστές

Τελεστές είναι ειδικά σύμβολα, που εφαρμοζόμενα σε έναν αριθμό εκφράσεων – που ονομάζονται *τελούμενα εισόδου* ή *τελεστέοι*, παράγουν μια νέα έκφραση.

Οι τελεστές της PASC400 δίνονται στο σχετικό πίνακα και διακρίνονται σε τελεστές με ένα τελούμενο και τελεστές με δύο τελούμενα εισόδου. Στην πρώτη κατηγορία ανήκουν οι τελεστές του προσήμου, της λογικής άρνησης και της επιλογής στοιχείου ζεύγους. Οι τελεστές αυτής της κατηγορίας αναγράφονται πριν το τελούμενό τους. Στη δεύτερη κατηγορία ανήκουν οι υπόλοιποι τελεστές αριθμητικών και λογικών πράξεων. Οι τελεστές αυτής της κατηγορίας αναγράφονται ανάμεσα στα τελούμενά τους.

Τελεστές μπορούν να θεωρηθούν και τα διαχωριστικά σύμβολα ‘(’, ‘)’, ‘[’ και ‘]’, σε περιπτώσεις που θα αναλυθούν στη συνέχεια.

Πιο συγκεκριμένα:

- Στην απλούστερη περίπτωση, ο τελεστής αναφοράς σε στοιχείο πίνακα ‘[]’ έχει σύνταξη:
 <αναγνωριστικό> ‘[’ <λίστα εκφράσεων> ‘]’

Ο τελεστής αυτός έχει δύο τελούμενα εισόδου, από τα οποία το πρώτο είναι το όνομα του πίνακα, και το δεύτερο μια λίστα εκφράσεων, οι τιμές των οποίων αποδίδονται στους δείκτες του στοιχείου πίνακα. Η αποτίμηση των εκφράσεων στη λίστα γίνεται από τα αριστερά προς τα δεξιά, και η αναφορά στο στοιχείο πίνακα γίνεται όπως περιγράφεται παραπάνω.

Στην πιο γενική περίπτωση, ο πίνακας μπορεί να είναι στοιχείο άλλου πίνακα, αλλά και το στοιχείο μπορεί να είναι επίσης τύπου πίνακα. Τότε, για την αναφορά στο τελευταίο αναγραφόμενο στοιχείο πίνακα και τον υπολογισμό της διεύθυνσής του, πρέπει να προηγηθούν οι επιλύσεις αναφορών για τους προηγούμενους τύπους, που μπορεί να είναι άλλοι πίνακες. Ένα παράδειγμα αναφοράς σε στοιχείο πίνακα τέτοιας μορφής δίνεται πιο κάτω.

Ακόμα, ως πίνακας μπορεί να είναι και κάποια έκφραση αριστερής προσπέλασης τύπου string, οπότε επιτρέπεται μόνο μία έκφραση δείκτη που αναφέρεται σε ένα συγκεκριμένο χαρακτήρα του ορμαθού. Η έκφραση δείκτη πρέπει τότε να είναι ακέραιου τύπου. Επειδή το μήκος του ορμαθού δεν είναι προκαθορισμένο, ο έλεγχος ορίων δείκτη πρέπει να γίνεται με βάση το τρέχον μήκος του ορμαθού. Η αρίθμηση των χαρακτήρων ενός ορμαθού γίνεται ξεκινώντας με 1 από την αρχή του ορμαθού.

- Ο τελεστής κλήσης συνάρτησης ‘()’ έχει σύνταξη:
 <αναγνωριστικό> ‘(’ <λίστα εκφράσεων> ‘)’

Ο τελεστής αυτός έχει δύο τελούμενα εισόδου, από τα οποία το πρώτο είναι το όνομα της συνάρτησης, και το δεύτερο μια λίστα εκφράσεων, οι τιμές των οποίων αποδίδονται στις πραγματικές παραμέτρους της κλήσης. Η αποτίμηση των εκφράσεων στη λίστα γίνεται από τα αριστερά προς τα δεξιά. Η κλήση συνάρτησης θα επεξηγηθεί παρακάτω.

Ο τελεστής δημιουργίας ζεύγους ‘()’ έχει σύνταξη:

‘(’ <έκφραση> ‘:’ <έκφραση> ‘)’

όπου οι δύο εκφράσεις πρέπει να είναι του ίδιου βασικού τύπου, ή συμβατών αριθμητικών τύπων ή τύπων υποπεριοχής. Ο τελεστής αυτός χρησιμοποιείται μόνο σε αναθέσεις και περάσματα παραμέτρων.

- Οι τελεστές επιλογής στοιχείου ζεύγους PAIROP, με σύνταξη:
 PAIROP <μεταβλητή>

Τελεστής	Περιγραφή	Αριθμός τελούμενων	Προσεταιριστικότητα
'[]', '()'	Αναφορά σε στοιχείο πίνακα, κλήση συνάρτησης/ δημιουργία ζεύγους	2	-
PAIROP	Αναφορά σε στοιχείο ζεύγους	1	-
NOTOP	Λογική άρνηση	1	-
MULDIVANDOP	Πολλαπλασιασμός, ακέραια / πραγματική διαίρεση, λογικό γινόμενο	2	αριστερή
ADDOP, OROP	Πρόσθεση, πρόσθεση, αφαίρεση, ένωση ορμαθών, λογικό άθροισμα	1, 2	αριστερή
RELOP, '='	Σχεσιακοί τελεστές	2	καμία

Τελεστές της PASC400 σε φθίνουσα σειρά προτεραιότητας

όπου <μεταβλητή> μπορεί να είναι αναγνωριστικό ή στοιχείο πίνακα. Το τελούμενο εισό-
δου του τελεστή αυτού πρέπει να είναι τύπου ζεύγους, ενώ το αποτέλεσμα της εφαρμογής
είναι του τύπου στοιχείου του ζεύγους.

- Ο τελεστής λογικής άρνησης NOTOP. Ο μοναδικός τελεστής του τελεστή αυτού πρέπει να είναι τύπου boolean και το αποτέλεσμα είναι του ίδιου τύπου.
- Οι τελεστές προσήμου ADDOP. Ο μοναδικός τελεστής των τελεστών αυτών πρέπει να είναι αριθμητικού τύπου και το αποτέλεσμα της εφαρμογής τους είναι του ίδιου τύπου.
- Οι αριθμητικοί τελεστές MULDIVANDOP '*', '/', 'div' (ακέραιο πηλίκο) και 'mod' (α-
κέραιο υπόλοιπο), και οι τελεστές ADDOP. Όταν οι τελεστές αυτοί εφαρμόζονται σε αριθ-
μητικούς τύπους, η συμβατότητα τύπων των τελεστών αυτών και ο τύπος του αποτελέ-
σματος της εφαρμογής τους καθορίζονται στον επόμενο πίνακα:

A \ B	integer	real
integer	integer, real με '/'	real, όχι 'div', 'mod'
real	real, όχι 'div', 'mod'	real, όχι 'div', 'mod'

Τύπος αποτελέσματος αριθμητικής έκφρασης A op B

Όπως δείχνει ο πίνακας, η μόνη περίπτωση μη συμβατότητας τύπων αφορά τους τελεστές
MULDIVANDOP 'div' και 'mod', οι οποίοι μπορούν να έχουν τελούμενα εισόδου μόνο
τύπου integer. Ακόμα, ο τελεστής MULDIVANDOP '/' δίνει πάντα αποτέλεσμα τύπου
real. Σε οποιαδήποτε περίπτωση τα τελούμενα εισόδου είναι διαφορετικού τύπου, είναι
απαραίτητο ο κώδικας να περιέχει μετατροπή του ενός τύπου στον τύπο του αποτελέσμα-
τος, η δε πράξη πρέπει να γίνεται για τον τύπο του αποτελέσματος. Ειδικά για τον τελεστή
'/', ακόμα κι όταν και τα δύο τελούμενά του είναι τύπου integer, πρέπει αυτά να μετατρέ-
πονται σε τύπο real.

Ο τελεστής ADDOP '+' εφαρμόζεται και σε τύπους ορμαθών χαρακτήρων, οπότε υλοποιεί
την ένωση δύο ορμαθών χαρακτήρων. Η ένωση είναι η μόνο πράξη που επιτρέπεται σε
τέτοιους τύπους, και το αποτέλεσμα είναι ένας νέος ορμαθός που περιέχει τους χαρακτήρες
του πρώτου χωρίς το τερματικό 0, ακολουθούμενους από τους χαρακτήρες του δεύτερου
ορμαθού.

- Οι τελεστές λογικών πράξεων MULDIVANDOP 'and' και OROP. Τα τελούμενα εισόδου
αυτών πρέπει να είναι τύπου boolean. Το αποτέλεσμα της εφαρμογής τους είναι επίσης
τύπου boolean.

- Οι σχεσιακοί τελεστές σύγκρισης RELOP και ισότητας '='. Τα τελούμενα εισόδου των τελεστών αυτών μπορούν να είναι του ίδιου ή διαφορετικού αριθμητικού τύπου. Στην τελευταία περίπτωση, ο κώδικας πρέπει να μετατρέπει το τελούμενο τύπου integer σε real. Μετά από πιθανή μετατροπή, η σύγκριση πρέπει να γίνεται για τον τύπο των τελούμενων. Οι τελεστές αυτοί δέχονται και τελούμενα μη αριθμητικών διατεταγμένων τύπων, δηλαδή των βασικών τύπων char, boolean και string, όπως και υποπεριοχών αυτών. Το αποτέλεσμα της εφαρμογής των σχεσιακών τελεστών είναι τύπου boolean.

Παρά την αναπαράσταση των τύπων boolean, char και απαρίθμησης με ακέραιες τιμές, στη PASC400 δεν υπάρχει συμβατότητα μεταξύ αριθμητικών και μη αριθμητικών τιμών. Έτσι, η συμμετοχή αριθμητικών εκφράσεων σε λογικές που αναφέρθηκε νωρίτερα μπορεί να γίνει μόνο μέσω των πιο πάνω σχεσιακών τελεστών. Αυτοί είναι οι μόνοι τελεστές που κατασκευάζουν λογικές εκφράσεις από αριθμητικές, καθώς δέχονται αριθμητικά τελούμενα και παράγουν αποτέλεσμα τύπου boolean. Το αντίστροφο δεν είναι εφικτό, γι' αυτό και δε μπορεί μια λογική έκφραση να συμμετέχει σε μια αριθμητική. Παρόμοια και για τον τύπο char – όπως και υποπεριοχών αυτού, μια έκφραση αυτού του τύπου μπορεί να συμμετάσχει σε κάποια λογική έκφραση μόνο μέσω των σχεσιακών τελεστών που αναφέρθηκαν, ενώ δε μπορεί να συμβεί το αντίστροφο, αφού δεν υπάρχει τελεστής που να δίνει αποτέλεσμα τύπου char. Επίσης δεν υπάρχει καμία δυνατότητα συμμετοχής τύπου char ή υποπεριοχών αυτού σε αριθμητική έκφραση.

Εξαίρεση στα παραπάνω αποτελούν οι τελεστές '[' και '(' που από τη μία προσφέρουν τη δυνατότητα *έμμεσης* συμμετοχής μιας έκφρασης σε μια μεγαλύτερη, αλλά και από την άλλη μπορούν να δώσουν αποτέλεσμα τύπου που δε μπορούν να δώσουν άλλοι τελεστές.

Σε κάθε εφαρμογή τελεστή, και όταν συμμετέχουν περισσότερα από ένα τελούμενα εισόδου, η αποτίμηση αυτών γίνεται από αριστερά προς τα δεξιά. Σε κάθε περίπτωση απαιτείται αποτίμηση όλων των τελούμενων, μια που η PASC400 δεν υποστηρίζει βραχυκύκλωση.

Μέσα σε μια έκφραση μπορούν να υπάρχουν πολλοί τελεστές. Η σειρά εφαρμογής αυτών για την αποτίμηση της έκφρασης καθορίζεται από παρενθέσεις ή από κανόνες προτεραιότητας και προσεταιριστικότητας. Η σειρά προτεραιότητας των τελεστών καθορίζεται από τη σειρά που αυτοί αναγράφονται στον πρώτο από τους δύο παραπάνω πίνακες, από μεγαλύτερη προς μικρότερη προτεραιότητα. Η προσεταιριστικότητα των τελεστών, δίνεται στον ίδιο πίνακα. Αν δεν έχει νόημα, η προσεταιριστικότητα δεν είναι ορισμένη και αναγράφεται ως '-'. Αξίζει να σημειωθεί ότι οι τελεστές RELOP δε μπορούν να έχουν καμία προσεταιριστικότητα, εφ' όσον η εφαρμογή τους αλλάζει τον τύπο της έκφρασης. Έτσι, η έκφραση:

$$x+2 > y > 0$$

είναι λανθασμένη. Εφαρμογή οποιουδήποτε από τους δύο τελεστές RELOP '>' δίνει αποτέλεσμα τύπου boolean, που δεν επιτρέπει την εφαρμογή του άλλου.

Σε μια σύνθετη έκφραση δεν επιτρέπεται διαδοχική εφαρμογή τελεστών προσήμου ADDOP, όπως για παράδειγμα στην έκφραση:

$$x > +3$$

που δεν είναι αποδεκτή.

Ας θεωρήσουμε για παράδειγμα την έκφραση:

$$(i*x-y-j*z < -a \text{ div } k \text{ div } 2) \text{ and not } (\text{head } b[k][i+2,m] <> 'a')$$

στην οποία κατ' αρχήν υποθέτουμε ότι οι τύποι των τιμών που συμμετέχουν είναι οι προβλεπόμενοι.

Στην έκφραση αυτή ο τελεστής MULDIVANDOP 'and' έχει τη μικρότερη προτεραιότητα και δε μπορεί να εφαρμοστεί πριν την αποτίμηση και των δύο τελεστών του. Αυτό συμβαίνει λόγω της χρήσης παρενθέσεων, μια που χωρίς αυτές, ο τελεστής MULDIVANDOP θα εφαρμοζόταν πριν τους τελεστές RELOP και ADDOP. Επίσης ο σχεσιακός τελεστής RELOP '<' στο αριστερό τελούμενο του προηγούμενου – το οποίο και αποτιμάται πριν από το δεξί – εφαρμόζεται

μόνο αφού έχουν αποτιμηθεί και τα δύο τελούμενά του. Στο αριστερό τελούμενο του τελευταίου, οι δύο διαδοχικές αφαιρέσεις που ορίζονται από τον τελεστή ADDOP ‘-’ εφαρμόζονται από αριστερά προς τα δεξιά, λόγω αριστερής προσηταιριστικότητας του τελεστή αυτού.

Ο πρώτος τελεστής που θα εφαρμοστεί στην πιο πάνω έκφραση θα είναι ο πρώτος από αριστερά τελεστής MULDIVANDOP ‘*’ που έχει μεγαλύτερη προτεραιότητα από τον ADDOP. Στη συνέχεια θα εφαρμοστεί ο πρώτος από αριστερά τελεστής ADDOP ‘-’, ενώ πριν εφαρμοστεί ο δεύτερος από τους δύο διαδοχικούς ADDOP, θα πρέπει να εφαρμοστεί ο δεύτερος τελεστής MULDIVANDOP ‘*’ που υπάρχει στο δεξί τελούμενο του προηγούμενου. Η εφαρμογή του πρώτου από αριστερά τελεστή MULDIVANDOP ‘div’ προηγείται του δεύτερου, ενώ ο τελεστής προσήμου ADDOP ‘-’ εφαρμόζεται μετά τους δύο διαδοχικούς MULDIVANDOP ‘div’, επιτρέποντας στη συνέχεια την εφαρμογή του τελεστή RELOP ‘<’.

Έτσι ολοκληρώνεται η αποτίμηση του αριστερού τελεστέου του τελεστή MULDIVANDOP ‘and’ και μπορούμε να συνεχίσουμε με την αποτίμηση του δεξιού.

Στο τελούμενο του τελεστή NOTOP ‘not’ – που λόγω των παρενθέσεων εφαρμόζεται μετά την εφαρμογή του RELOP ‘<’ – αποτιμάται πρώτα το αριστερό τελούμενο του τελευταίου, το οποίο είναι στοιχείο ζεύγους, το οποίο είναι στοιχείο πίνακα, ο οποίος είναι επίσης στοιχείο πίνακα.

Έτσι, λόγω της προτεραιότητας του τελεστή ‘[]’ έναντι του τελεστή PAIROP ‘head’, προηγείται η επίλυση αναφοράς στον πίνακα “b”, η οποία μπορεί να γίνει, αφού πρώτα αποτιμηθεί ο δείκτης “k” και ελεγχθεί η ορθότητα της τιμής του. Η επίλυση αναφοράς στον “b” θα μας δώσει την αρχική διεύθυνση του πίνακα “b[k]”. Στη συνέχεια και με παρόμοιο τρόπο, αφού αποτιμηθούν οι δείκτες “i+2” και “m”, και γίνει έλεγχος ορθότητας των τιμών τους, υπολογίζεται η αρχική διεύθυνση του στοιχείου “b[k][i+2,m]”, το οποίο είναι ζεύγος και από το οποίο επιλέγεται και αποτιμάται το πρώτο στοιχείο για να μας δώσει τη ζητούμενη τιμή, που πρέπει να είναι τύπου char.

Η αποτίμηση του δεύτερου τελούμενου του RELOP ‘<’ δίνει την τιμή της σταθεράς ‘a’ τύπου char.

Τελειώνοντας, εφαρμόζεται ο παραπάνω τελεστής RELOP ‘<’, ακολουθούμενος από τον NOTOP. Η εφαρμογή του MULDIVANDOP ‘and’ μπορεί τώρα να πραγματοποιηθεί.

Όπως φάνηκε στο παραπάνω παράδειγμα, πρέπει να χρησιμοποιούμε παρενθέσεις σε κάθε περίπτωση που επιθυμούμε παρέμβαση στους πιο πάνω κανόνες, όπως προβλέπει η σύνταξη των εκφράσεων της PASC400. Για παράδειγμα, η έκφραση:

$$(-i+1) * (a - (b-c))$$

(α) επιβάλλει την αποτίμηση του δεξιού τελούμενου του ADDOP ‘-’ πριν από το αριστερό, και (β) επιβάλλει την εφαρμογή των τελεστών ADDOP ‘+’ και ‘-’ πριν από την εφαρμογή του MULDIVANDOP ‘*’.

Πιο γενικά, παρενθέσεις μπορούν να χρησιμοποιηθούν και για βελτίωση της εμφάνισης μιας έκφρασης, χωρίς αναγκαστικά να επηρεάζουν τους κανόνες εφαρμογής των τελεστών που συμμετέχουν σε αυτήν.

Κλήση συναρτήσεων

Αν ‘func’ είναι το όνομα μιας συνάρτησης με αποτέλεσμα τύπου ‘type’, τότε η έκφραση

$$\text{func}(e_1, e_2, \dots, e_n)$$

είναι μια τιμή δεξιάς προσπέλασης τύπου ‘type’. Η αποτίμηση αυτής γίνεται με την εκτέλεση του κώδικα της μονάδας της συνάρτησης, με τις εξής προϋποθέσεις:

- Ο αριθμός n των εκφράσεων στις παρενθέσεις – οι πραγματικές παράμετροι – πρέπει να είναι ίσος με τον αριθμό των τυπικών παραμέτρων.
- Ο τύπος κάθε πραγματικής παραμέτρου με πέρασμα κατ’ αξία πρέπει να είναι συμβατός με τον τύπο της αντίστοιχης τυπικής παραμέτρου, σύμφωνα με τους κανόνες συμβατότητας για ανάθεση που περιγράφονται πιο κάτω.
- Ο τύπος κάθε πραγματικής παραμέτρου με πέρασμα κατ’ αναφορά πρέπει να ταυτίζεται με τον τύπο της αντίστοιχης τυπικής παραμέτρου.

Κατά την κλήση μιας συνάρτησης οι πραγματικές παράμετροι αποτιμώνται από αριστερά προς τα δεξιά και τοποθετούνται στη στοίβα, στο χώρο που δεσμεύεται ο χώρος δεδομένων της συνάρτησης, απ' όπου θα μπορεί να τους χρησιμοποιήσει η μονάδα αυτής. Για μια παράμετρο κατ' αναφορά, στη στοίβα τοποθετείται η αντίστοιχη διεύθυνση. Ειδικά για παράμετρο τύπου string, αν η πραγματική παράμετρος είναι ένας σταθερός ορμαθός χαρακτήρων και όχι τιμή αριστερής προσπέλασης, ο μεταγλωττιστής δεσμεύει χώρο για μια προσωρινή τοπική μεταβλητή τύπου string στο καλούν περιβάλλον, παράγει κώδικα που να αντιγράφει στο χώρο αυτό το περιεχόμενο του ορμαθού μαζί με το τερματικό 0, και περνάει πια ως πραγματική παράμετρο τη διεύθυνση αυτής της μεταβλητής. Οι διαδοχικοί χαρακτήρες του ορμαθού αποθηκεύονται σε διαδοχικές θέσεις του χώρου 256 θέσεων που δεσμεύονται, ξεκινώντας από την πρώτη θέση. Ο ορμαθός που παρέχεται δεν πρέπει να έχει μήκος μεγαλύτερο από 255 χαρακτήρες.

Με την έξοδο από τη συνάρτηση, θα πρέπει η τιμή του αποτελέσματος να βρίσκεται σε προκαθορισμένο σημείο αποθήκευσης, επίσης στη στοίβα. Με κάθε κλήση μιας συνάρτησης, η στοίβα μεταβάλλεται, κι έτσι ο χώρος δεδομένων της συνάρτησης, δηλαδή το *εγγραφήμα δραστηριοποίησης* της συνάρτησης, είναι δυναμικός.

Η προκαθορισμένη συνάρτηση LENGTH() δέχεται ως παράμετρο έναν ορμαθό χαρακτήρων, και επιστρέφει το πλήθος των χαρακτήρων του ορμαθού – χωρίς το τερματικό 0.

Εντολές

Η PASC400 υποστηρίζει απλές και δομημένες εντολές, καθώς και τη σύνθετη εντολή. Μια δομημένη εντολή της PASC400 περιέχει μία ή περισσότερες απλές, δομημένες ή σύνθετες εντολές. Υπενθυμίζεται ότι μια σύνθετη εντολή μπορεί να είναι κενή, αλλά περιέχει οπωσδήποτε τις λέξεις-κλειδιά “begin” και “end”, ενώ δε μπορεί να περιέχει δηλώσεις.

Οι απλές εντολές της PASC400 είναι:

- Η εντολή ανάθεσης
- Η εντολή κλήσης διαδικασίας
- Οι εντολές εισόδου/εξόδου

Οι δομημένες εντολές της PASC400 είναι:

- Η εντολή διακλάδωσης
- Η εντολή βρόχου
- Η εντολή επανάληψης

Μεταξύ οποιωνδήποτε διαδοχικών εντολών της PASC400 πρέπει να υπάρχει ο χαρακτήρας ‘;’.

Η εντολή ανάθεσης

Η εντολή αυτή αποδίδει την τιμή μιας έκφρασης σε μια τιμή αριστερής προσπέλασης, δηλαδή σε μια μεταβλητή ή ένα στοιχείο πίνακα του χώρου δεδομένων του προγράμματος. Στη δεύτερη περίπτωση, της ανάθεσης πρέπει να προηγηθεί η αποτίμηση των εκφράσεων των δεικτών και ο υπολογισμός της τιμής αριστερής προσπέλασης του στοιχείου.

Η ανάθεση γίνεται με αποθήκευση της τιμής της έκφρασης στη διεύθυνση που παριστάνει η τιμή αριστερής προσπέλασης.

Παραδείγματα αναθέσεων είναι τα εξής:

```
a := w > 0.0
ch := 'c'
c[i-1, mm[j]] := x*x+1
l := (i+5:j*m)
```

όπου c και mm είναι πίνακες ακεραίων, ενώ l είναι ζεύγος ακεραίων.

Για να μπορεί να γίνει ανάθεση, πρέπει η τιμή της έκφρασης να είναι συμβατού τύπου με τον τύπο της αριστερής προσπέλασης.

Δύο βασικοί τύποι είναι συμβατοί για ανάθεση, εάν:

- (α) ταυτίζονται, ή αλλιώς
- (β) είναι αριθμητικοί, οπότε συμβαίνει μετατροπή σε πραγματικό για ανάθεση από τύπο integer σε τύπο real, ή αποκοπή του κλασματικού μέρους για ανάθεση από τύπο real σε τύπο integer.

Δύο τύποι απαρίθμησης είναι συμβατοί για ανάθεση, εάν ταυτίζονται.

Ο τύπος υποπεριοχής θεωρείται ως του τύπου από τον οποίο παράγεται, όσο αφορά συμβατότητα για ανάθεση. Όμως, και αν ο τύπος αριστερής προσπέλασης είναι τύπος υποπεριοχής, ο κώδικας θα πρέπει πριν την ανάθεση να ελέγχει αν η τιμή της έκφρασης είναι μέσα στα όρια της υποπεριοχής.

Σε μια ανάθεση ενός ορμαθού χαρακτήρων σε μια μεταβλητή τύπου string, οι διαδοχικοί χαρακτήρες του ορμαθού αντιγράφονται σε διαδοχικές θέσεις του χώρου 256 θέσεων που έχουν δεσμευτεί για τη μεταβλητή, ξεκινώντας από την πρώτη θέση. Μετά τους χαρακτήρες του ορμαθού, αποθηκεύεται η τιμή 0, ενώ οι υπόλοιπες θέσεις του χώρου μέχρι τις 256 διατηρούν την όποια προηγούμενη τιμή τους. Ο ορμαθός που παρέχεται ως τιμή δεξιάς προσπέλασης στο δεξί μέλος της ανάθεσης δεν πρέπει να έχει μήκος μεγαλύτερο από 255 χαρακτήρες.

Ανάθεση μεταξύ σύνθετων τύπων επιτρέπεται, όταν αυτοί έχουν την ίδια δομή και οι βασικοί τύποι των στοιχείων τους είναι συμβατοί. Τότε, ο μεταγλωττιστής πρέπει να κατασκευάσει τον κώδικα που θα αντιγράψει πιθανά πολλαπλές τιμές για την εκτέλεση της ανάθεσης. Αν για παράδειγμα x και y είναι πίνακες συμβατών τύπων, η ανάθεση

$x := y$

γίνεται με αντιγραφή όλων των στοιχείων του y στο αντίστοιχο στοιχείο του x.

Μια ειδική μορφή ανάθεσης της PASC400 είναι η ανάθεση στη μεταβλητή που παριστάνεται από το όνομα μιας συνάρτησης. Η τιμή αριστερής προσπέλασης αυτής είναι διεύθυνση στη στοίβα. Η ανάθεση αυτή κατά τα άλλα ακολουθεί τους πιο πάνω κανόνες. Σε μια συνάρτηση πρέπει να υπάρχει τουλάχιστον μία ανάθεση στο όνομά της.

Η εντολή κλήσης διαδικασίας

Η εντολή αυτή μεταφέρει τη ροή του κώδικα σε κάποια διαδικασία. Παραδείγματα κλήσης διαδικασίας είναι τα παρακάτω:

```
proc_A(x, n+2, matrix, s[i] and t)
proc_B
```

Στην εντολή κλήσης διαδικασίας οι περιορισμοί στις πραγματικές παραμέτρους είναι ταυτόσημοι με τους αντίστοιχους περιορισμούς στην κλήση συνάρτησης που περιγράφηκαν νωρίτερα. Ο μηχανισμός κλήσης είναι επίσης ο ίδιος με το μηχανισμό κλήσης μιας συνάρτησης.

Οι εντολές εισόδου/εξόδου

Αυτές είναι οι εντολές ανάγνωσης (read) και εγγραφής (write) δεδομένων. Συντάσσονται σαν κλήσεις διαδικασιών – όπως και είναι στην πραγματικότητα – με το όνομα της εντολής και μια λίστα στοιχείων σε παρενθέσεις, που στην περίπτωση ανάγνωσης είναι τιμές αριστερής προσπέλασης βασικού τύπου, ενώ στην περίπτωση εγγραφής είναι εκφράσεις.

Παραδείγματα εντολών εισόδου/εξόδου είναι:

```
read (n,x)
write ("Temperature: ",f,"F, or ",5/9*(f-32),"C.")
```

Η είσοδος και η έξοδος γίνονται στα συνήθη αρχεία εισόδου/εξόδου χωρίς προδιαγραφές, δηλαδή προηγούμενο καθορισμό του τύπου και της μορφής των στοιχείων που διαβάζονται ή γράφονται. Με κάθε εντολή εξόδου γράφεται μια γραμμή κειμένου στην έξοδο του προγράμματος.

Η εντολή διακλάδωσης

Η εντολή διακλάδωσης `if` είναι μια δομημένη εντολή με παραμέτρους μια λογική έκφραση και μία ή δύο άλλες εντολές.

Η εκτέλεση της εντολής διακλάδωσης ξεκινά με την αποτίμηση της λογικής έκφρασης. Αν αυτή έχει τιμή «Αληθής», εκτελείται η εντολή που ακολουθεί τη λέξη-κλειδί “`then`”, ενώ αν υπάρχει η λέξη-κλειδί “`else`”, τότε η εντολή που ακολουθεί αυτή τη λέξη δε θα εκτελεστεί. Εάν η λογική έκφραση έχει τιμή «Ψευδής» και υπάρχει η αντίστοιχη λέξη-κλειδί “`else`”, τότε εκτελείται μόνο η εντολή που ακολουθεί αυτή τη λέξη, ενώ αν δε υπάρχει η λέξη-κλειδί “`else`”, δεν εκτελείται καμία εντολή.

Ένα παράδειγμα εντολής διακλάδωσης είναι το παρακάτω:

```
if ((a[i+1] > 13.5) and z[j]) then
  begin
    a[i-1] := f(z[j+1], i-1)*2;
    if x_play = 0 then b_over := a[i]
  end
else
  a[i-1] := f(z[n-i], i-1)*k_z[i]+1
```

Η εντολή βρόχου

Η εντολή βρόχου `while` είναι μια δομημένη εντολή με παραμέτρους μια λογική έκφραση και μια εντολή.

Η εκτέλεση της εντολής `while` ξεκινά με αποτίμηση της έκφρασης. Αν η έκφραση έχει τιμή «Αληθής», εκτελείται η εντολή που ακολουθεί τη λέξη-κλειδί “`do`”. Στη συνέχεια η διαδικασία επαναλαμβάνεται με νέα αποτίμηση της έκφρασης, και η εντολή ξαναεκτελείται όσο η τιμή της έκφρασης που υπολογίζεται είναι «Αληθής».

Μόλις η αποτίμηση της έκφρασης δώσει τιμή «Ψευδής», η εκτέλεση του βρόχου τερματίζεται.

Ένα παράδειγμα εντολής βρόχου δίνεται παρακάτω:

```
while i > 0 do
  begin
    a[i*m] := y/2;
    k := f(n*m, a, x);
    while j < 10 do
      b[j, i+k] := g(j, m-k)*1.0;
    i := i-1
  end
```

Η εντολή επανάληψης

Η εντολή επανάληψης `for` είναι μια δομημένη εντολή της PASC400 με παραμέτρους ένα πεδίο επανάληψης και μια εντολή.

Το πεδίο επανάληψης περιέχει το όνομα μιας μεταβλητής, που ονομάζεται μεταβλητή ελέγχου του βρόχου επανάληψης, και δύο αριθμητικές εκφράσεις, που αποτιμώμενες δίνουν: η πρώτη την αρχική τιμή και η δεύτερη την τελική τιμή της μεταβλητής ελέγχου. Το βήμα αύξησης της μεταβλητής ελέγχου μεταξύ διαδοχικών επαναλήψεων του βρόχου είναι 1, αν μεταξύ των δύο εκφράσεων υπάρχει η λέξη-κλειδί “`to`”, και -1, αν μεταξύ των δύο εκφράσεων υπάρχει η λέξη-κλειδί “`downto`”.

Η εκτέλεση της εντολής `for` ξεκινά με αποτίμηση των εκφράσεων του πεδίου επανάληψης, και ανάθεση της αρχικής τιμής στη μεταβλητή ελέγχου. Στη συνέχεια η τιμή αυτή συγκρίνεται με την τελική, και εάν είναι μέσα στα όρια μεταβολής της, εκτελείται η εντολή που ακολουθεί τη

λέξη-κλειδί “do”. Έπειτα αυξάνεται κατάλληλα η τιμή της μεταβλητής ελέγχου του βρόχου, και η διαδικασία επαναλαμβάνεται με τη σύγκριση της τιμής αυτής με την τελική. Όταν η τιμή της μεταβλητής ελέγχου βρεθεί εκτός ορίων μεταβολής της, η εκτέλεση του βρόχου τερματίζεται.

Ας σημειωθεί ότι οι εκφράσεις του πεδίου επανάληψης αποτιμώνται *μόνο* μία φορά, στην αρχή της εκτέλεσης της εντολής for.

Ένα παράδειγμα εντολής επανάληψης δίνεται παρακάτω:

```
for i := 0 to n-1 do
  begin
    a[i*m] := y/2;
    k := s*f(n*m, a, x[i]);
    for j := m downto m-i do
      b[j, i+k] := (m-k)*1.0
    end
```

Η σημασιολογία της εντολής επανάληψης συμπληρώνεται από τους ακόλουθους κανόνες:

1. Η μεταβλητή ελέγχου του βρόχου πρέπει να είναι τύπου integer.
2. Ο τύπος των εκφράσεων του πεδίου επανάληψης πρέπει να είναι integer.
3. Η μεταβλητή ελέγχου δεν επιτρέπεται να λαμβάνει τιμή μέσα στην εσωτερική εντολή του βρόχου – άρα ούτε να μεταδίδεται κατ’ αναφορά σε κάποιο υποπρόγραμμα που καλείται μέσα από την εντολή αυτή.