



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών

Δομές Δεδομένων και Τεχνικές Προγραμματισμού

Ενότητα 2: ΑΤΔ Στοίβα

Ιωάννης Κοτρώνης
Σχολή Θετικών Επιστημών
Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Σκοποί ενότητας

- Ορίζει τον ΑΤΔ Στοίβα
- Σχεδιαστικές Επιλογές με Πίνακα
- Υλοποίηση με ενότητες στην C
 - Μερική Απόκρυψη
 - Ολική Απόκρυψη
- Αναπτύσσει Τυπικές Εφαρμογές



Περιεχόμενα ενότητας

- Ορισμός ΑΤΔ Στοίβα
- Χρήση Στοίβας για μετατροπή Δεκαδικού σε δυαδικού
- Σχεδιασμός με πίνακα
- Υλοποίηση με Μερική και Ολική Απόκρυψη
- Μεταδιατεταγμένες παραστάσεις (μετατροπή από ένδο και υπολογισμός με ΑΤΔ Στοίβα)



ΑΤΔ ΣΤΟΙΒΑ

Ορισμός-Σχεδιασμός-Υλοποίηση

ΣΤΟΙΒΑ

ΧΑΡΑΚΤΗΡΙΣΤΙΚΗ ΙΔΙΟΤΗΤΑ ΤΗΣ ΣΤΟΙΒΑΣ

Το τελευταίο στοιχείο που εισήχθη θα εξαχθεί πρώτο.

Άλλο όνομα L I F O (Last In First Out)



Χρήσεις Στοίβας

- Καθημερινή Ζωή (όχι πάρα πολλές)
 - Δίσκοι Τραπεζαρίας
 - Αυτοκίνητα σε αδιέξοδο δρόμο μιας λωρίδας
 - Κερματοδέκτης
- Πληροφορική (σημαντικότερες!!!)
 - Μεταγλώττιση (συντακτική αναγνώριση-π.χ. παρενθέσεις, μετατροπές παραστάσεων)
 - Εκτέλεση (Κλήσεις συναρτήσεων, αναδρομή, υπολογισμός παραστάσεων, κλπ)
 - Μεγάλος αριθμός αλγορίθμων



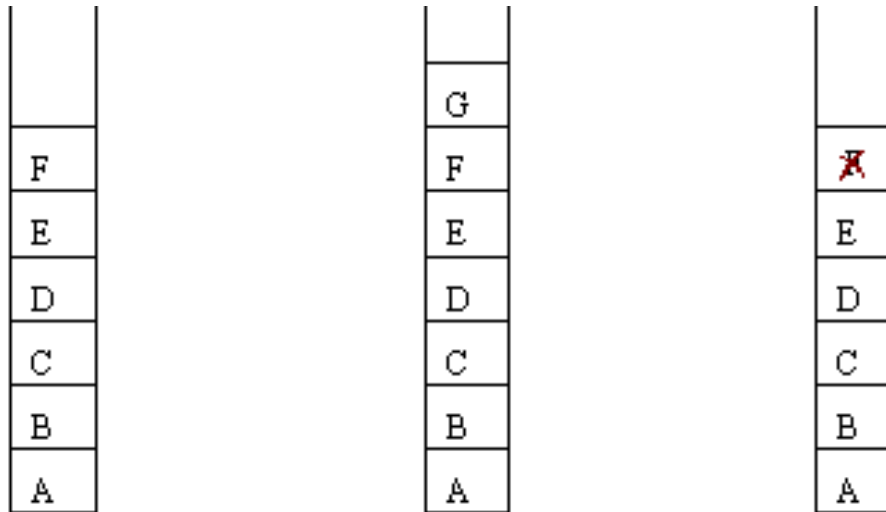
Ορισμός Στοίβας (stack)

- Μια στοίβα αντικειμένων τύπου T είναι μια ακολουθία αντικειμένων τύπου T στην οποία ορίζονται οι ακόλουθες πράξεις
 - Δημιουργία (ή αρχικοποίηση)
 - Διαπίστωση Κενής
 - Ώθηση αντικειμένου (push)
 - Εξαγωγή αντικειμένου (pop)
 - (Διαπίστωση Πλήρους)



Βασική Λειτουργικότητα

Η στοίβα είναι μια συλλογή δεδομένων με γραμμική διάταξη στην οποία όλες οι εισαγωγές και οι διαγραφές γίνονται στο ένα άκρο που λέγεται **κορυφή (top)** της στοίβας



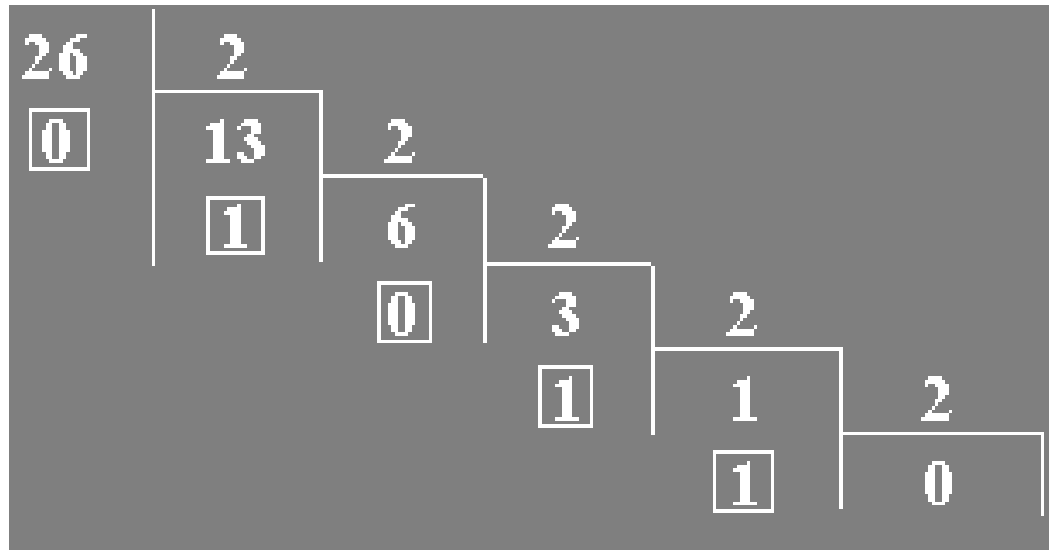
Σχήμα: Λειτουργία Στοίβας

εισαγωγή (b) και διαγραφή (c) από την ίδια αρχική στοίβα (a)



Παράδειγμα Χρήσης

Ο δεκαδικός αριθμός 26 στο δυαδικό σύστημα παριστάνεται ως 11010. Ο Υπολογισμός του όπως δείχνει το Σχήμα.



Παρατηρούμε ότι τα ψηφία του δυαδικού αριθμού δημιουργούνται με την αντίθετη σειρά (το υπόλοιπο της διαίρεσης του προηγούμενου πηλίκου δια 2).



Αλγόριθμος για τη μετατροπή ενός θετικού ακέραιου από το δεκαδικό στο δυαδικό σύστημα

Όσο ο αριθμός > 0 :

- Υπολογισμός του υπολοίπου που προκύπτει από τη διαίρεση του αριθμού με το 2.
- Τοποθέτηση του υπολοίπου στη στοίβα με τα υπόλοιπα.
- Αντικατάσταση του αριθμού με το ακέραιο πηλίκο του αριθμού δια του 2.

Όσο η στοίβα των υπολοίπων δεν είναι άδεια:

- Εξαγωγή του υπολοίπου από την στοίβα των υπολοίπων.
- Εμφάνιση του υπολοίπου.



Στο ακόλουθο σχήμα εμφανίζεται διεξοδικά η εφαρμογή του αλγορίθμου στον αριθμό 26.

Να σημειωθεί ότι για την εφαρμογή του παραπάνω αλγορίθμου χρησιμοποιούνται οι πράξεις του ΑΤΔ στοίβα.



Υπολογισμός

Στοιβά Υπολοίπων

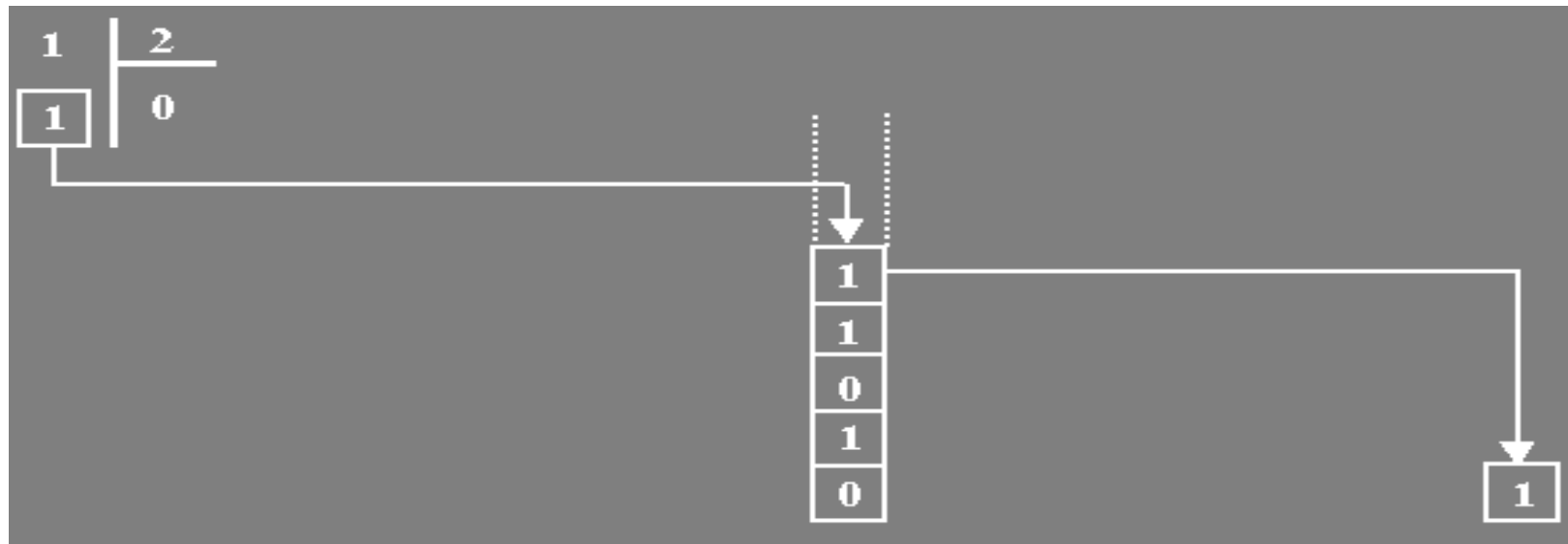
Αποτέλεσμα



Υπολογισμός

Στοίβα Υπολοίπων

Αποτέλεσμα



Υπολογισμός

Στοίβα Υπολοίπων

Αποτέλεσμα



Υπολογισμός

Στοίβα Υπολοίπων

Αποτέλεσμα



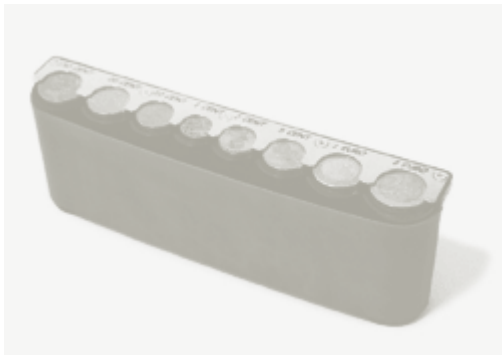
ΥΛΟΠΟΙΗΣΗ

- Ακολουθεί τον ορισμό
- Σχεδιασμός (επιλογές-προδιαγραφές)
 - Χρήση Πίνακα (με μετακινήσεις στοιχείων ή χωρίς)
 - Συνδεδεμένη λίστα
- Προγραμματισμός (από τις προδιαγραφές σε πρόγραμμα)
 - Ενότητες
 - Συναρτήσεις

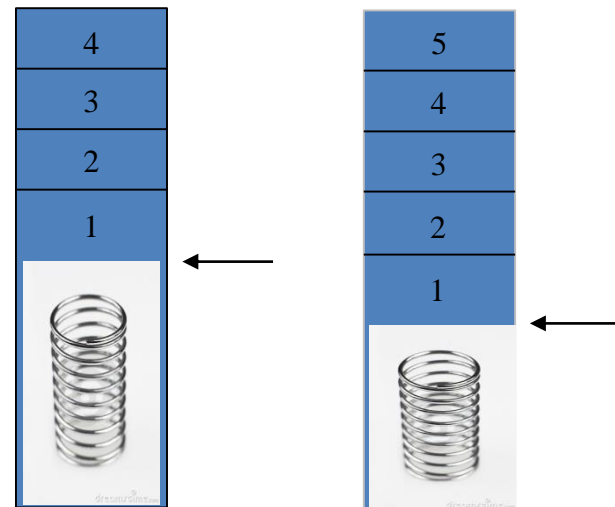


Σχεδιασμός (1^η επιλογή-κακή)

- Αναλογία
Κερματοδέκτη

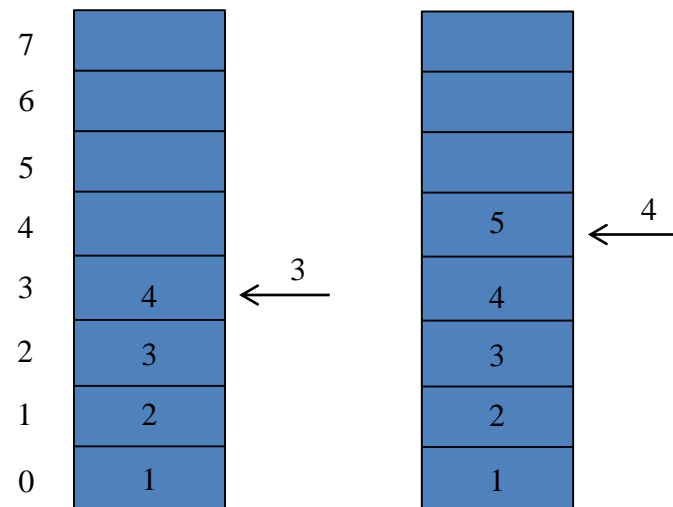


Πίνακας με Μετακινήσεις
Στοιχείων και τελευταίας
θέσης



Σχεδιασμός (2^η επιλογή-καλή)

- Πίνακας χωρίς μετακινήσεις στοιχείων
- Μόνο αλλαγή θέσης
- Σύμβαση: Θέση δηλώνει το πρώτο διαθέσιμο στοιχείο για εξαγωγή



Υλοποίηση του ΑΤΔ στοίβα με πίνακα στην C (δεδομένα):

```
#define plithos 100                //το μέγεθος της στοίβας
typedef ... typos_stoixeiou;
    //ο τύπος στοιχείων που διαχειρίζεται η στοίβα
    // αφήνεται αδιευκρίιστος, αλλά ονομάζεται

// Ο τύπος δεδομένων της στοίβας

typedef struct
{
    int korifi;
    typos_stoixeiou pinakas[plithos];
} typos_stoivas;

//Δήλωση τριών στοιβών

typos_stoivas stoiva1, stoiva2, stoiva3;
```



Στη συνέχεια θα υλοποιηθούν οι πράξεις στη στοίβα

Δημιουργία της στοίβας :

```
void dimiourgia(typos_stoivas * stoivaPtr)
/* Προ : Καμία.
   Μετά : Δημιουργία κενής στοίβας*/
{
    stoivaPtr->korifi=-1;
    //σύμβαση korifi πρώτο στοιχείο για εξαγωγή
}
```

Στην πραγματικότητα εδώ έχουμε αρχικοποίηση στοίβας και όχι δημιουργία



Έλεγχος αν μια στοίβα είναι κενή :

```
int keni(typos_stoivas stoiva)

/* Προ : Δημιουργία στοίβας.
   Μετά : Επιστρέφει 1 αν η στοίβα είναι κενή, διαφορετικά 0
*/
{
    return (stoiva.korifi==-1); //σύμφωνα με την σύμβαση
}
```



Η λειτουργία της εξαγωγής

Αν η στοίβα είναι κενή,
τότε λάθος και σταματά η εκτέλεση της
αλλιώς
Εξάγει το πρώτο στοιχείο,
μειώνει την korifi και
Επιστρέφει την τιμή του στοιχείου



```

void exagogi ( typos_stoivas *stoivaPtr,
               typos_stoixeiou *stoixeiοPtr)
/* Προ : Η στοίβα δεν είναι κενή.
Μετά : Επιστρέφει την τιμή του stoixeiο */
{
    if (keni (*stoivaPtr))
        printf(" Η στοίβα είναι κενή ");
    else
    {
        *stoixeiοPtr=stoivaPtr-> pinakas[stoivaPtr->korifi];
        stoivaPtr->korifi--;
    }
}

```



Μια καλύτερη μορφή της εξαγογι είναι η παρακάτω:
(όχι μήνυμα, αλλά ένδειξη-flag)

```
void εξαγογι( typos_stoivas *stoivaPtr,  
             typos_stoixeiou *stoixeiοPtr,  
             int *yποxeilisi)  
  
/* Προ : Μη κενή στοίβα.  
Μετά : Επιστρέφει την τιμή του stoixeiο και η τιμή  
της yποxeilisi είναι 0, διαφορετικά είναι 1. */  
  
{  
  if (keni(*stoivaPtr))  
    *yποxeilisi = 1;  
  else  
  {    *yποxeilisi = 0;  
    *stoixeiοPtr = stoivaPtr->pinakas[stoivaPtr->korifi];  
    stoivaPtr->korifi --;  
  }  
}
```

συνέχεια



Γιατί ο δεύτερος τρόπος είναι καλύτερος;

Αν η εξαγογι καλείται σε πολλά σημεία του προγράμματος μπορούμε να εντοπίσουμε το λάθος ευκολότερα και να κάνουμε διορθωτικές κινήσεις ανάλογα με την θέση κλήσης της συνάρτησης.

Όπου καλείται η εξαγωγή γράφουμε κώδικα ελέγχου

```
exagogi (&stoiva, &stoiceio, &yproxeilisi);  
  
if yproxeilisi          /* να γίνει διορθωτική ενέργεια */  
    printf("Position 1: yproxeilisi");  
else                    /* το στοιχείο που εξήχθη κανονικά */  
...  
;
```



Η υλοποίηση της ώθησης (push):

```
void othisi (typos_stoivas *stoivaPtr,
            typos_stoixeiou stoixeio,
            int *yperheilisi);

/*Αν η στοίβα είναι πλήρης yperheilisi είναι 1 αλλιώς 0 */

{if (stoivaPtr->korifi == plithos-1)
    *yperheilisi = 1
else {
    *yperheilisi = 0;
    stoivaPtr->korifi++;
    stoivaPtr->pinakas[stoivaPtr->korifi]=stoixeio;
}
}
```



Υλοποίηση του αλγορίθμου μετατροπής δεκαδικού αριθμού σε δυαδικό χρησιμοποιώντας τον ΑΤΔ στοίβα

```
dimiourgia (&stoiva);  
  
scanf ("%d", &arithmos);  
  
while (arithmos != 0)  
{  
    ypoloipo = arithmos % 2;  
    othisi (&stoiva, ypoloipo);  
    arithmos = arithmos / 2;  
}  
printf ("Η δυαδική παράσταση του %d είναι:", arithmos);  
  
while (!keni (&stoiva))  
{  
    exagogi (&stoiva, &ypoloipo);  
    putchar (ypoloipo);  
}
```



Το αρχείο stack.h (μερική απόκρυψη)

```
#ifndef __CH4_STACK__
#define __CH4_STACK__

#define PLITHOS 100 /* μέγιστο επιτρεπτό μέγεθος στοίβας */
#include "typos_stoixeiou.h" // ορίζεται ο τύπος της στοίβας
typedef struct {
    int korifi;
    typos_stoixeiou pinakas[PLITHOS];
} typos_stoivas;

void dimiourgia(typos_stoivas *stoivaPtr);
int keni(typos_stoivas *stoivaPtr);
void exagogi(typos_stoivas *stoivaPtr,
             typos_stoixeiou *stoixeioPtr, int *ypoxeilisi);
void othisi(typos_stoivas *stoivaPtr,
            typos_stoixeiou stoixeio, int *yperxeilisi);

#endif
```



Δημόσια (ανεπιθύμητη) πρόσβαση

Το πρόγραμμα πελάτης (main) έχει πρόσβαση στα μέλη της

```
typedef struct {  
    int korifi;  
    typos_stoixeiou pinakas[PLITHOS];  
} typos_stoivas;
```

Επομένως μπορεί να αλλάξει άμεσα τα μέλη της stoiva παρακάμπτοντας την (υποχρεωτική) χρήση των πράξεων, π.χ.

```
Typos_stoiva stoiva;  
Stoiva.korifi=100;  
Stoiva.pinaka[5]=99;
```



Άλλος πιο ασφαλής τρόπος (.h) (αδιαφανείς δείκτες)

```
#ifndef __CH4_STACK__
#define __CH4_STACK__

#include "typos_stoixeiou.h"
typedef struct StackStruct * typos_stoivas;
/* opaque (αδιαφανής) δείκτης */

typos_stoivas dimiourgia( );
/* real creation not initialisation */
int keni(const typos_stoivas stoiva);
void exagogi(const typos_stoivas stoiva,
             typos_stoixeiou * const stoixeioptr,
             int *ypoxeilisi);
void othisi(const typos_stoivas stoiva,
            typos_stoixeiou stoixeio, int *yperxeilisi);

#endif
```



Το αρχείο stack.c (1)

```
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include "ch4_Stack.h"

#define PLITHOS 100 /*μέγιστο επιτρεπτό μέγεθος στοίβας*/
typedef struct StackStruct {
    int korifi;
    typos_stoixeiou pinakas[PLITHOS];
}StackStruct;

typos_stoivas dimiourgia( )
{ typos_stoivas ThisStoiva= malloc(sizeof(StackStruct));
  ThisStoiva->korifi = -1;
  return ThisStoiva;
}
```



```

int keni(const typos_stoivas stoiva)
{
    return (stoiva->korifi == -1);
}

void exagogi(const typos_stoivas stoiva,
             typos_stoixeioy * const stoixeioPtr, int *ypoxeilisi)
{
    if (keni(stoiva))
        *ypoxeilisi = 1;
    else
    {
        *ypoxeilisi = 0;
        *stoixeioPtr = stoiva->pinakas[stoiva->korifi];
        stoiva->korifi--;
    }
}

void othisi( const typos_stoivas stoiva,
             typos_stoixeioy stoixeio, int *yperxeilisi)
{
    if (stoiva->korifi == (PLITHOS -1))
        *yperxeilisi = 1;
    else
    {
        *yperxeilisi = 0;
        stoiva->korifi++;
        stoiva->pinakas[stoiva->korifi]=stoixeio;
    }
}

```



Ποια μορφή χρησιμοποιώ;

- Στον κώδικα των σημειώσεων και στις περισσότερες διαφάνειες παρουσιάζεται η πρώτη μορφή (με την ανεπιθύμητη δημόσια πρόσβαση)
- Όμως συνιστάται θερμά η εξοικείωση και η χρήση της 2^{ης} (πλήρους απόκρυψης με αδιαφανείς δείκτες) μορφής που παρέχουν περισσότερη προστασία.
- Στις 1^η εργασία αποδεκτές όλες οι μορφές. Στις 2 και 3 η 2^η μορφή.



Εφαρμογές με τη χρήση στοίβας

- Υπολογισμός Αριθμητικών Παραστάσεων - Πολωνικός Συμβολισμός
- Αλγόριθμος για τον υπολογισμό της μεταθεματικής μορφής
- Μετατροπή Ενδοθεματικής στη Μεταθεματική μορφή



Υπολογισμός Αριθμητικών Παραστάσεων - Πολωνικός Συμβολισμός

$$\begin{array}{l} A + (B * C) \quad \text{Ενδοθεματική μορφή} \\ A + (BC *) \\ A(BC *) + \\ ABC * + \quad \text{Μεταθεματική μορφή} \end{array}$$

Οι κανόνες που διέπουν τη μετατροπή μιας αριθμητικής παράστασης από την ενδοθεματική στη μεταθεματική μορφή είναι :

- 1) Οι πράξεις που έχουν τη μεγαλύτερη προτεραιότητα πρέπει να μετατραπούν πρώτες
- 2) Μόλις μετατραπεί ένα τμήμα της παράστασης σε μεταθεματική μορφή θα πρέπει το τμήμα αυτό να εκλαμβάνεται ως ένας όρος.



Περιττές οι παρενθέσεις στη Μεταθεματική μορφή

Ας υποθέσουμε ότι έχουμε την ενδοθεματική μορφή: $(1 + 5) * (8 - (4 - 1))$
η οποία έχει τη μεταθεματική μορφή: 1 5 + 8 4 1 - - *

Αντικαθιστώντας το 1 5 + με την τιμή του αποτελέσματος 1 + 5, δηλαδή με το 6
έχουμε την

$$6 8 4 1 - - *$$

Αντικαθιστώντας το 4 1 - με την τιμή της 4 - 1 έχουμε

$$6 8 3 - *$$

Συνεχίζοντας λαμβάνουμε διαδοχικά:

$$6 \underline{83} - *$$

$$\underline{65} *$$

$$30$$

η τιμή της παράστασης.



Αλγόριθμος για τον υπολογισμό της μεταθεματικής μορφής

1. Να δημιουργηθεί μια στοίβα.
2. Όσο υπάρχουν χαρακτήρες στο αρχείο εισόδου να εκτελούνται οι παρακάτω εργασίες:
 - a. Πάρε τον επόμενο χαρακτήρα (ψηφίο ή αριθμητικός τελεστής) της μεταθεματικής μορφής.
 - b. Αν ο χαρακτήρας είναι όρος, τότε να εισαχθεί στη στοίβα
Αν ο χαρακτήρας είναι τελεστής
 - (i) Βγάλε πρώτα δύο στοιχεία- όρους - της στοίβας (αν η στοίβα δεν περιέχει δύο στοιχεία -όρους- , τότε η μεταθεματική παράσταση είναι λάθος και ο υπολογισμός της σταματά).
 - (ii) Εφάρμοσε τον τελεστή για τους δύο όρους.
 - (iii) Τοποθέτησε το αποτέλεσμα στη στοίβα.
3. Το αποτέλεσμα του υπολογισμού της μεταθεματικής μορφής βρίσκεται στην κορυφή της στοίβας (το μοναδικό στοιχείο, αν όχι τότε λάθος).



Εφαρμογή του Αλγορίθμου

Ενδοθεματική : $2 * 4 - (9 + 5)$

Μεταθεματική : $2 4 * 9 5 + -$

Παράσταση

$2 \ 4 * 9 5 + -$

$4 \ * 9 5 + -$

$* \ 9 5 + -$

$9 \ 5 + -$

Στοιβά

$2 \leftarrow$

$4 \leftarrow$
 2

$8 \leftarrow$

$9 \leftarrow$
 8

Σχόλια

Τοποθέτηση του 2 στη στοιβά

Τοποθέτηση του 4 στη στοιβά

Εξαγωγή των 4 και 2, υπολογισμός της $2*4=8$ και τοποθέτηση του 8 στη στοιβά

Τοποθέτηση του 9 στη στοιβά



Παράσταση

5 + -

+ -

-

τέλος

Στοιίβα

5	←
9	
8	

14	←
8	

-6	←
----	---

-6	←
----	---

Σχόλια

Τοποθέτηση του 5 στη στοιίβα

Εξαγωγή των 5 και 9, υπολογισμός της $9+5=14$ και τοποθέτηση του 14 στη στοιίβα

Εξαγωγή των 14 και 8, υπολογισμός της $8-14=-6$ και τοποθέτηση του -6 στη στοιίβα

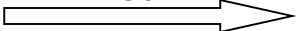
Η τιμή της παράστασης βρίσκεται στη κορυφή της στοιίβας



Πρόγραμμα για τον υπολογισμό της τιμής μιας μεταθεματικής παράστασης (υποθέτουμε ότι κάθε γραμμή του αρχείου εισόδου περιέχει μια σειρά από θετικούς μονοψήφιους ακέραιους και αριθμητικούς τελεστές μόνο)

```
#include <stdio.h>
#include <stdlib.h>

#include "stoiva.h"
```

συνέχεια 



Συνάρτηση που ορίζει πράξεις τελεστών

```
float telestis (char s, float oros1, float oros2, int *error )
{
    switch (s)
    {
        case '+': return (oros1+oros2); break;
        case '-': return (oros1-oros2); break;
        case '*': return (oros1*oros2); break;
        case '/': return (oros1/oros2); break;
        default : *error=0;           // flag!!!
                return (1/0);
    }
}
```



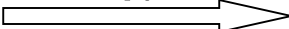
```
int main( )
{
    typos_stoivas stoiva;

    float oros1,oros2, apotelesma;

    char symbolo , apantisi , *metathematiki;

    int i;

    int ypo, yper, err; // για ενδείξεις λάθων
```

συνέχεια 



```

dimiourgia (&stoiva);

printf("Δώσε τη παράσταση:");
while ( (symbolo=getchar( )) != '\n' ) {
    if (symbolo!=' ' ) /*αγνοεί τα κενά*/

    if (symbolo>='0' && symbolo<='9') // μονοψήφιοι
    {
        apotelesma=(float)symbolo-'0';
        othisi( &stoiva, apotelesma, &yper);
    }
    else /* είναι τελεστής */{
        exagogi (&stoiva, &oros2, &ypo);
        exagogi (&stoiva, &oros1, &ypo);
        apotelesma=telestis (symbolo,oros1,oros2, &err);
        othisi (&stoiva,apotelesma);
    }
}/* while ..!= '\n' */

exagogi (&stoiva, &apotelesma, &ypo);
printf("%5.2f\n", apotelesma);
}

```

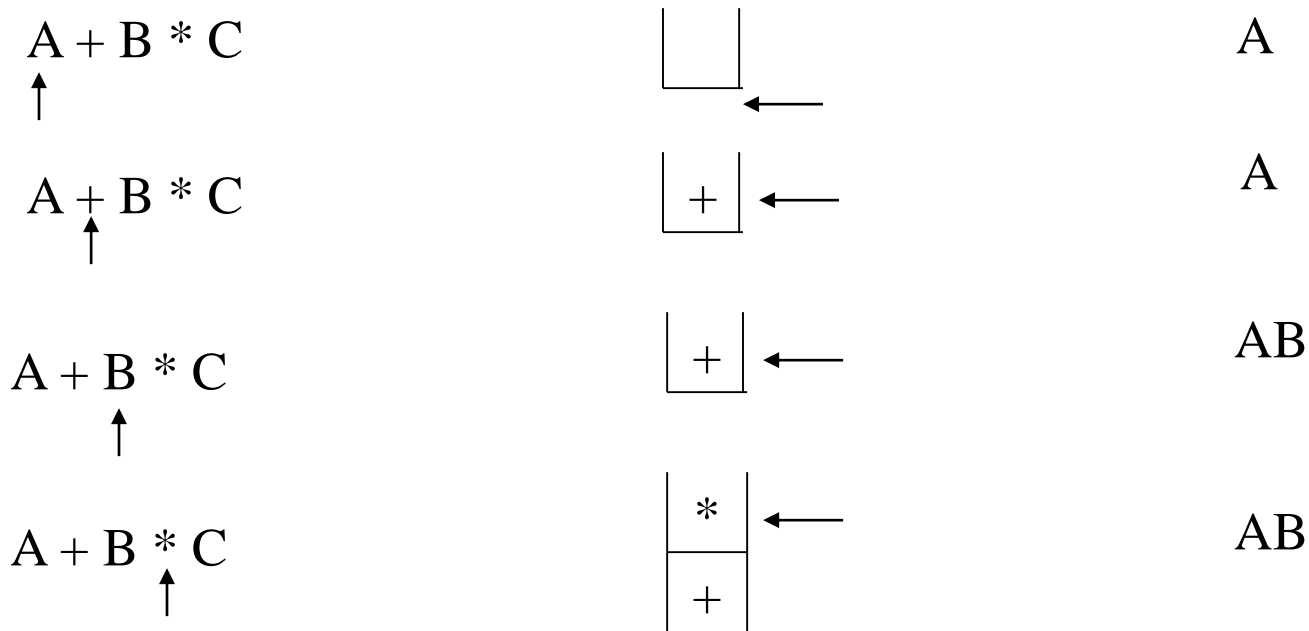


Μετατροπή Ενδοθεματικής στη Μεταθεματική μορφή

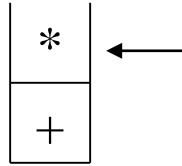
$$A + B * C \longrightarrow A B C * +$$

Οι όροι τυπώνονται απευθείας στην έξοδο

Οι τελεστές αποθηκεύονται στην στοίβα (ίσως πρέπει να βγουν άλλοι πριν)



A + B * C
↑



ABC

A + B * C
↑



ABC*+



Αν προτ(τελεστή) > προτ(τελεστή στοίβας) τότε ώθηση στη στοίβα

$A * B + C$ \longrightarrow $AB * C +$

$A * B + C$ \leftarrow A
 \uparrow

$A * B + C$ \leftarrow A
 \uparrow $*$

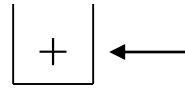
$A * B + C$ \leftarrow AB
 \uparrow $*$

$A * B + C$ \leftarrow $AB *$
 \uparrow



$$A * B + C$$

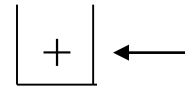
↑



$$AB^*$$

$$A * B + C$$

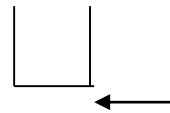
↑



$$AB^*C$$

$$A * B + C$$

↑



$$AB^*C+$$



Ίδια προτεραιότητα, τότε κριτήριο προσεταιριστικότητας

$A + B - C$ ↑	$\boxed{}$ ←	A
$A + B - C$ ↑	$\boxed{+}$ ←	A
$A + B - C$ ↑	$\boxed{+}$ ←	AB
$A + B - C$ ↑	$\boxed{-}$ ←	AB+
$A + B - C$ ↑	$\boxed{-}$ ←	AB+C
$A + B - C$ ↑	$\boxed{}$ ←	AB+C-



Μετατροπή ενδοθεματικής παράστασης σε μεταθεματική

- Δημιουργία μιας κενής στοίβας τελεστών.
- Όσο υπάρχουν σύμβολα στην ενδοθεματική μορφή και δεν υπάρχει λάθος, πάρε το επόμενο σύμβολο και εξέτασε 4 περιπτώσεις:
 - 1) Αριστερή παρένθεση '(' : τοποθετείται στη στοίβα.
 - 2) Δεξιά παρένθεση ')' : εξάγονται από τη στοίβα και προστίθενται στη μεταθεματική σειρά (εκτυπώνονται) όλοι οι τελεστές μέχρις ότου βρεθεί αριστερή παρένθεση, η οποία δεν τυπώνεται. Αν δεν βρεθεί '(', η ενδοθεματική μορφή είναι λανθασμένη.
 - 3) Αριθμητικός Τελεστής (Εισόδου-ΤΕ):
 - 1) Όσο (α και β και (γ ή δ))
 - α) η στοίβα δεν είναι κενή και
 - β) ο τελεστής στην κορυφή της στοίβας (ΤΣ) δεν είναι '(' και
 - γ) προτεραιότητά του ΤΕ < προτεραιότητα του ΤΣ, ή
 - δ) προτεραιότητά του ΤΕ = προτεραιότητα του ΤΣ και προσηταιριστικότητα ΤΣ αριστερήΤότε εξάγεται ο ΤΣ και προστίθενται στη μεταθεματική σειρά
 - 2) Τοποθετείται στη στοίβα ο ΤΕ.
 - 4) όρος: προστίθεται απευθείας στη μεταθεματική σειρά (εκτυπώνεται).



Η φιλοσοφία του παραπάνω αλγορίθμου βασίζεται στην επιθυμία της εμφάνισης των τελεστών με τη σειρά με την οποία θα εκτελεστούν οι πράξεις. Στο ακόλουθο σχήμα εμφανίζεται η εφαρμογή του αλγορίθμου για την έκφραση $7 * 8 - (2 + 3)$

Παράσταση	Στοιβά	Εκτύπωση	Σχόλια
$7 * 8 - (2 + 3)$	κενή	7	Εκτύπωση του 7
$* 8 - (2 + 3)$	* ←	7	ώθηση του *
$8 - (2 + 3)$	* ←	7 8	Εκτύπωση του 8
$- (2 + 3)$		7 8*	Εξαγωγή και εκτύπωση του *
$(2 + 3)$	- ←	7 8*	ώθηση του -



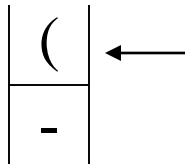
Παράσταση

Στοιβά

Εκτύπωση

Σχόλια

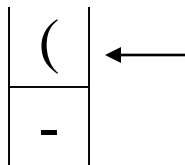
(2+3)



7 8*

ώθηση της (

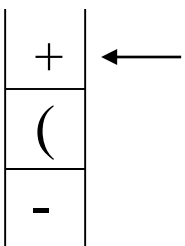
2 +3)



7 8*2

Εκτύπωση του 2

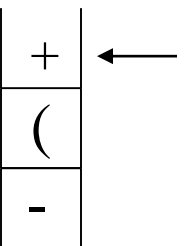
+ 3)



7 8*2

ώθηση του +

3)



7 8*2 3

Εκτύπωση του 3



Παράσταση

)

Στοιβά

(←
-

Εκτύπωση

78*23+

Σχόλια

Εξαγωγή και εκτύπωση
του +

- ←

78*23+

Εξαγωγή της (

τέλος της
παράστασης

78*23+-

Εξαγωγή και εμφάνιση
του -

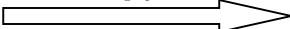


Στη συνέχεια ακολουθεί το πρόγραμμα για τη μετατροπή μιας ενδοθεματικής παράστασης στην αντίστοιχη μεταθεματική.

```
#include <stdio.h>
#include <stdlib.h>
#include "stoiva.h"
#define mikos 80

typedef char typos_parastasis[mikos];

int protereotita(char telestis);
void metatropi (typos_parastasis endo_parastasi);
```

συνέχεια 



```
main( ){
    typos_parastasis endo_parastasi;
    int i;

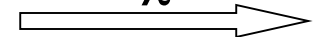
    printf("Δώσε την παράσταση\n");
    i=-1;
    do
    {
        i++;
        endo_parastasi[i] = getchar( );
    } while (endo_parastasi[i]!=telos);
    i++;

    endo_parastasi[i]='\0';

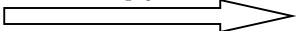
    printf ("Μετατροπή της παράστασης:");

    metatropi(endo_parastasi);
    printf ("\n");
}
```

συνέχεια



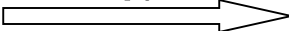
```
int protereotita(char telestis)
/*Επιστρέφει την προτεραιότητα του τελεστή telestis */
{
    switch (telestis)
    {
        case '+': case '-': return 1;
        case '*': case '/': return 2;
    }
}
```

συνέχεια




```
void metatropi (typos_parastasis endo_parastasi)
/* Αυτή η υπορουτίνα μετατρέπει την ενδοθεματική μορφή
   μιας παράστασης στην μεταθεματική της μορφή*/
{
    typos_stoivas stoiva;
    int i;
    char symbolo, symbolo_korifi ;
    int lathos, telos_exagogis;

    dimiourgia (&stoiva);
    lathos = 0;
    /* αρχή της μετατροπής */
    i = 0;
    symbolo = endo_parastasi[0]; // πρώτο σύμβολο
```

συνέχεια 




```

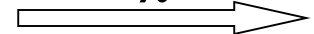
while ((symbolo!=telos) && ( ! lathos ))
{
    while ( endo_parastasi[i] == ' ' ) /*Αγνόηση κενών*/
        i++;
    symbolo = endo_parastasi[i];

    /* Για κάθε σύμβολο στη θέση i εξετάζουμε 4 περιπτώσεις:
    1η περίπτωση '(' */

    if ( symbolo == '(' ) /* αριστερή παρένθεση */
        othisi(&stoiva,symbolo);
    else // άλλες περιπτώσεις

```

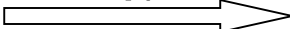
συνέχεια



```

    if ( symbolo == ')' ) /* 2η περίπτωση: δεξιά
παρένθεση*/
    {
        telos_exagogis = 0;
        do {
            if (keni(stoiva))
                lathos = 1;
            else {
                exagogi( &stoiva , &symbolo_korifi );
                if ( symbolo_korifi != '(' )
                    printf("%2c", symbolo_korifi);
                else
                    telos_exagogis = 1;
            }
        } while ( ( ! telos_exagogis ) && ( ! lathos ) );
    }
else // άλλες περιπτώσεις

```

συνέχεια 

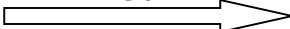


```

if ( strchr(synolo_teleston, symbolo )!= NULL )
/* 3η τελεστής */
{
    telos_exagogis = 0;
    while ( (! keni(stoiva)) && (! telos_exagogis) )
    {
        exagogi( &stoiva , &symbolo_korifi );
        if ((protereotita(symbolo))
            <=(protereotita(symbolo_korifi)))
            printf("%2c", symbolo_korifi);
        else
        {
            othisi( &stoiva, symbolo_korifi );
            telos_exagogis = 1;}
        }
        othisi(&stoiva, symbolo);
    }
else
/* 4η όρος */
if (symbolo != telos)
    printf("%2c",symbolo);

i++; // επόμενο σύμβολο στην μεταθετική
} /* while 4 περιπτώσεις */

```

συνέχεια 



```

/* Εξαγωγή και εκτύπωση των στοιχείων της στοίβας. */
while ( ( ! keni(stoiva) ) && ( ! lathos ) )
{
    exagogi(&stoiva, &symbolo_korifi);

    if (symbolo_korifi != '(')
        printf("%2c", symbolo_korifi);
    else lathos = 1;
}
if ( lathos )
    printf («Λάθος στην ενδοθεματική παράσταση.\n");
else
    printf ("\n");
}

```



Παράδειγμα

Να μετατραπεί η παράσταση:

$$A \wedge B * C - D + E / F / (G + H)$$

$$A \wedge B * C - D + E / F / (GH +)$$

$$(AB \wedge) * C - D + E / F / (GH +)$$

$$(AB \wedge C *) - D + E / F / (GH +)$$

$$(AB \wedge C *) - D + (EF /) / (GH +)$$

$$(AB \wedge C *) - D + (EF / GH + /)$$

$$(AB \wedge C * D -) + (EF / GH + /)$$

$$AB \wedge C * D - EF / GH + / +$$

μεταθεματική μορφή



Τέλος Ενότητας

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Σημειώματα

Σημείωμα Αναφοράς

Copyright Εθνικών και Καποδιστριακών Πανεπιστημίων Αθηνών, Κοτρώνης Ιωάννης. «Δομές Δεδομένων και Τεχνικές Προγραμματισμού. Ενότητα 2: ΑΤΔ Στοίβα». Έκδοση: 1.01. Αθήνα 2015.

Διαθέσιμο από τη δικτυακή διεύθυνση:
<http://opencourses.uoa.gr/courses/DI105/>.



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.



Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

