

# Προηγμένες αρχιτεκτονικές υπολογιστών και προγραμματισμός παραλλήλων συστημάτων

## 11. Παράλληλοι βρόχοι

A series of horizontal lines of varying lengths and colors (teal, light blue, white) extending from the left edge of the slide towards the right, positioned below the section header.

# Εισαγωγή

- Ορισμός: Εμφωλευμένοι βρόχοι (nested loops) λέγονται οι βρόχοι που βρίσκεται ο ένας μέσα στον άλλο, πχ.

```
for (i=1; i<=1000; i++) {  
    for (j=1; j<10; j++) {  
        for (k=20; k>=0; k--) {  
            .....  
        }  
    }  
}
```

- Οι εμφωλευμένοι βρόχοι είναι στα περισσότερα προβλήματα η κύρια πηγή καθυστέρησης.
- Συνεπώς είναι πολύ σημαντική η καλύτερη δυνατή παραλληλοποίηση των βρόχων.

# Γενική μορφή απλού βρόχου

- Γενικά, ένας απλός (μη εμφωλευμένος βρόχος) έχει την εξής μορφή:

```
for (i = p; i <= q; i = i + s) {  
    /* κάποιες εντολές */  
    H(i);  
}
```

Αρχικό όριο

Τελικό όριο

Βήμα

Πυρήνας  
ή Σώμα  
του βρόχου

- Το  $i$  καλείται : **δείκτης**
  - Είναι συνήθως ακέραιος
  - Αλλά όχι απαραίτητα...

# Κανονικός απλός βρόχος

- Ορισμός: Κανονικός λέγεται ένας βρόχος που έχει:

- Αρχικό όριο  $(p) = 0$
- Τελικό όριο  $(q) \geq 0$
- Βήμα  $(s) = 1$

- Παράδειγμα:

```
for (i=0; i<=10; i++) {  
    /*κάποιες εντολές*/  
    H(i);  
}
```

# Μη κανονικός βρόχος

- Παράδειγμα: Ο αλγόριθμος Fibonacci:

```
int F[11];  
F[0] = 0;  
F[1] = 1;  
for (i=2; i<=10; i++) {  
    F[i] = F[i-1] + F[i-2];  
}
```

- Αρχικό όριο  $p=2$ . Τελικό όριο  $q=10$ . Βήμα  $s=1$ . Πυρήνας του βρόχου = « $F[i] = F[i-1] + F[i-2];$ ».

# Στιγμιότυπα και μετρητής επανάληψης

- Κάθε επανάληψη του βρόχου καλείται **στιγμιότυπο**.

| Επανάληψη | Δείκτης (i) | Πράξεις που εκτελούνται |
|-----------|-------------|-------------------------|
| i' = 0    | i=2         | F[2] = F[1] + F[0];     |
| i' = 1    | i=3         | F[3] = F[2] + F[1];     |
| i' = 2    | i=4         | F[4] = F[3] + F[2];     |
| i' = 3    | i=5         | F[5] = F[4] + F[3];     |
| ...       | ...         | .....                   |
| i' = 8    | i=10        | F[10] = F[9] + F[8];    |

- Το i' καλείται μετρητής επανάληψης και δεν ισούται με τον δείκτη i καθώς το i' ξεκινάει από το 0 ενώ το i ξεκινάει από το 2. Αν ο βρόχος ήταν κανονικός, τότε το i' και το i θα ταυτίζονταν.

# Κανονικοποίηση βρόχου

Όχι κανονικός βρόχος:

*/\* s θετικό \*/*

```
for (i=p; i<=q; i=i+s) {  
    H(i);  
}
```

Ή

*/\* s αρνητικό \*/*

```
for (i=p; i>=q; i=i+s) {  
    H(i);  
}
```

Ισοδύναμος κανονικός βρόχος:

```
for (I=0; I<=Q; I++) {  
    H(p+I*s);  
}
```

Όπου αντικαθιστούμε

$i$  με  $p+I*s$

και

$Q$  με  $(q-p)/s$

$I = (i-p)/s$

# Παράδειγμα: Κανονικοποίηση Fibonacci

Αρχικός βρόχος:

```
for (i=2; i<=10; i=i+1) {  
    F[i]=F[i-1]+F[i-2];  
}
```

- $p=2$
- $q=10$
- $s=1$

Κανονικοποιημένος βρόχος:

```
for (I=0; I<=8; I=I+1) {  
    F[I+2]=F[I+1]+F[I];  
}
```

- $Q=8$
- $i = I+2$

# Επαλήθευση: Κανονικοποίηση Fibonacci

```
for (i=2; i<=10; i=i+1) {  
    F[i] = F[i-1] + F[i-2];  
}
```

- i=2:  $F[2]=F[1]+F[0]$ ;
- i=3:  $F[3]=F[2]+F[1]$ ;
- i=4:  $F[4]=F[3]+F[2]$ ;
- i=5:  $F[5]=F[4]+F[3]$ ;
- i=6:  $F[6]=F[5]+F[4]$ ;
- ...

```
for (I=0; I<=8; I=I+1) {  
    F[I+2] = F[I+1] + F[I];  
}
```

- I=0:  $F[2]=F[1]+F[0]$ ;
- I=1:  $F[3]=F[2]+F[1]$ ;
- I=2:  $F[4]=F[3]+F[2]$ ;
- I=3:  $F[5]=F[4]+F[3]$ ;
- I=4:  $F[6]=F[5]+F[4]$ ;
- ...

# Παράδειγμα: Κανονικοποίηση βρόχου με βήμα > 1

- Αρχικός βρόχος:

```
for (i=-2; i<=10; i=i+2) {  
    x[i+2]=f(x[i+1],x[i]);  
}
```

- $p = -2$
- $q = 10$
- $s = 2$

- Κανονικός βρόχος:

```
for (I=0; I<=6; I++) {  
    x[2*I] = f(x[2*I-1], x[2*I-2]);  
}
```

- $Q = (10 - (-2)) / 2 = 6$
- $i = -2 + I * 2$

# Επαλήθευση

```
for (i=-2; i<=10; i=i+2) {  
    x[i+2]=f(x[i+1],x[i]);  
}
```

```
i=-2: x[0]=f(x[-1],x[-2]);  
i=0:  x[2]=f(x[1],x[0]);  
i=2:  x[4]=f(x[3],x[2]);  
i=4:  x[6]=f(x[5],x[4]);  
i=6:  x[8]=f(x[7],x[6]);  
...
```

```
for (I=0; I<=6; I++) {  
    x[2*I] = f(x[2*I-1], x[2*I-2]);  
}
```

```
I=0: x[0]=f(x[-1],x[-2]);  
I=1: x[2]=f(x[1],x[0]);  
I=2: x[4]=f(x[3],x[2]);  
I=3: x[6]=f(x[5],x[4]);  
I=4: x[8]=f(x[7],x[6]);  
...
```

# Παράδειγμα: Κανονικοποίηση βρόχου με αρνητικό βήμα

- Αρχικός βρόχος:

```
for (i=100; i>=2; i=i-3) {  
    A[i-2]=3*A[i]-B[i-1];  
}
```

- $p=100$
- $q=2$
- $s=-3$

- Κανονικός βρόχος:

```
for (I=0; I<=32; I++) {  
    A[98-3*I]=3*A[100-3*I]-B[99-3*I-1];  
}
```

- $Q=\text{floor}((2-100)/(-3))=32$
- $i=100-3*I$

# Επαλήθευση

```
for (i=100; i>=2; i=i-3) {  
    A[i-2]=3*A[i]-B[i-1];  
}
```

```
i=100: A[98]=3*A[100]-B[99];  
i=97:  A[95]=3*A[97]-B[96];  
i=94:  A[92]=3*A[94]-B[93];  
i=91:  A[89]=3*A[91]-B[90];  
...
```

```
for (I=0; I<=32; I++) {  
    A[98-3*I]=3*A[100-3*I]-B[99-3*I-1];  
}
```

```
I=0: A[98]=3 * A[100] - B[99];  
I=1: A[95]=3 * A[97]  - B[96];  
I=2: A[92]=3 * A[94]  - B[93];  
I=3: A[89]=3 * A[91]  - B[90];  
...
```

# Παράδειγμα: Κανονικοποίηση βρόχου με δεκαδικό βήμα

- Αρχικός βρόχος:

```
j = 0;  
for(i=-1; i<=1; i=i+0.01){  
    x[j] = f(i);  
    j++;  
}
```

- $p = -1$
- $q = 1$
- $s = 0.01$

- Κανονικός βρόχος:

```
j = 0;  
for (I=0; I<=200; I++) {  
    x[j] = f(-1+0.01*I);  
    j++;  
}
```

- $Q = (1 - (-1)) / 0.01 = 200$
- $i = -1 + 0.01 * I$

# Επαλήθευση

```
j = 0;
for (i=-1; i<=1; i=i+0.01) {
    x[j] = f(i);
    j++;
}
```

```
i=-1:    x[0]=f(-1); j=1;
i=-0.99: x[1]=f(-0.99); j=2;
i=-0.98: x[2]=f(-0.98); j=3;
i=-0.97: x[3]=f(-0.97); j=4;
...
```

```
j = 0;
for (I=0; I<=200; I++) {
    x[j] = f(-1 + 0.01 * I);
    j++;
}
```

```
I=0:    x[0]=f(-1); j=1;
I=1:    x[1]=f(-0.99); j=2;
I=2:    x[2]=f(-0.98); j=3;
I=3:    x[3]=f(-0.97); j=4;
• ...
```

# Εμφωλευμένοι Βρόχοι

- Όταν ο πυρήνας ενός βρόχου είναι επίσης βρόχος τότε λέμε ότι έχουμε ένα εμφωλευμένο βρόχο.
- Ορισμός: **Διάσταση** ή **βάθος** ενός εμφωλευμένου βρόχου είναι το πλήθος των βρόχων που βρίσκονται ο ένας μέσα στον άλλο.
- Παράδειγμα:

```
for (i1=p1; i1<=q1; i1+=s1) {  
    for (i2=p2; i2<=q2; i2+=s2) {  
        for (i3=p3; i3<=q3; i3+=s3) {  
            .....  
        }  
    }  
}
```

# Στιγμιότυπα, επαναλήψεις, όρια και σώμα εμφωλευμένου βρόχου (1)

- Όλοι οι παρακάτω ορισμοί αφορούν εμφωλευμένο βρόχο βάθους  $n$  με δείκτες  $i_1, i_2, \dots, i_n$  :

```
for (i1=p1; i1<=q1; i1++) {  
    for (i2=p2; i2<=q2; i2++) {  
        .....  
        for (in=pn; in<=qn; in++) {  
            .....  
        }  
    }  
}
```

- **Στιγμιότυπο:** το διάνυσμα των δεικτών  $\mathbf{i}=[i_1, i_2, \dots, i_n]$  για συγκεκριμένες τιμές των  $i_1, i_2, \dots, i_n$
- **Όρια:** τα διανύσματα  $\mathbf{P}=[p_1, p_2, \dots, p_n]$  και  $\mathbf{Q}=[q_1, q_2, \dots, q_n]$ .

## Στιγμιότυπα, επαναλήψεις, όρια και σώμα εμφωλευμένου βρόχου (2)

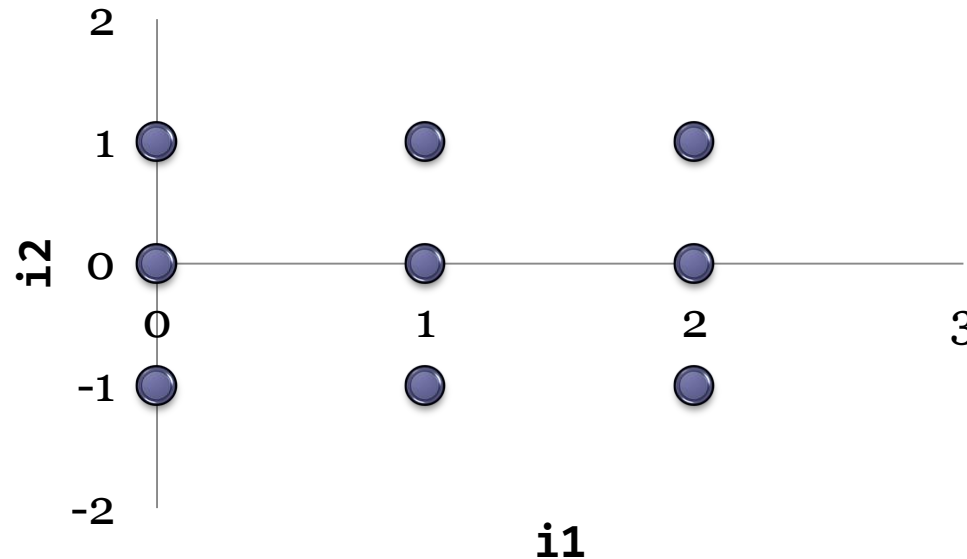
- **Χώρος στιγμιotypών:** το σύνολο όλων των τιμών που μπορεί να πάρει το διάνυσμα  $\mathbf{i}=[i_1, i_2, \dots, i_n]$  με βάση τα όρια  $\mathbf{P}=[p_1, p_2, \dots, p_n]$  και  $\mathbf{Q}=[q_1, q_2, \dots, q_n]$ .
- **Παράδειγμα:**

```
for (i1=0; i1<=2; i1++) {      /* p1=0; q1=2 */
    for (i2=-1; i2<=1; i2++) { /* p2=-1; q2=1 */
        .....
    }
}
```

Χώρος στιγμιotypών =  $\{[0, -1], [0, 0], [0, 1], [1, -1], [1, 0], [1, 1], [2, -1], [2, 0], [2, 1]\}$

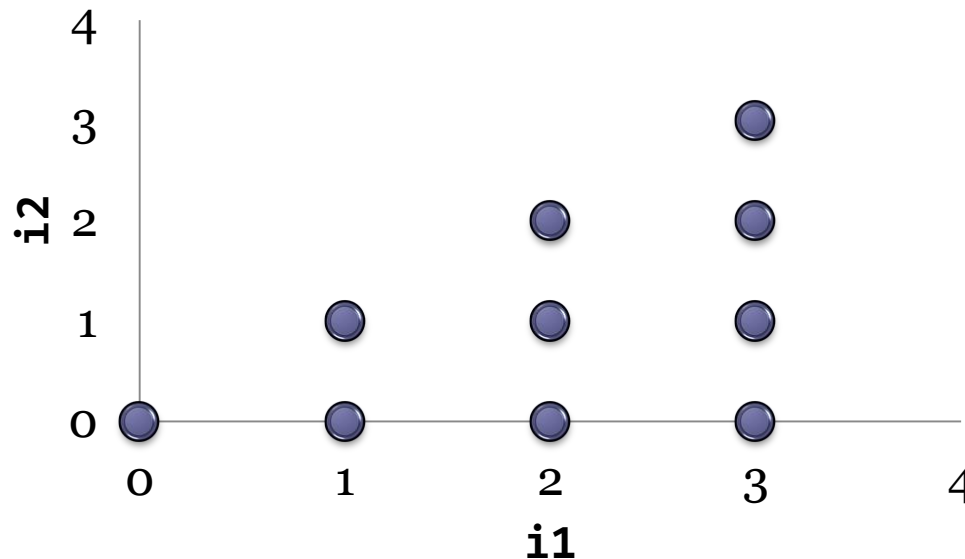
## Παράδειγμα 1: Χώρος Στιγμιοτύπων, Σταθερά όρια

```
for (i1=0; i1<=2; i1++) { /* p1=0; q1=2 */  
  for (i2=-1; i2<=1; i2++) { /* p2=-1; q2=1 */  
    .....  
  }  
}
```



## Παράδειγμα 2: Χώρος Στιγμιοτύπων, Όρια = συναρτήσεις των δεικτών

```
for (i1=0; i1<=3; i1++) { /* p1=0; q1=3 */  
  for (i2=0; i2<=i1; i2++) { /* p2=0; q2=i1 */  
    .....  
  }  
}
```



## Στιγμιότυπα, επαναλήψεις, όρια και σώμα εμφωλευμένου βρόχου (3)

- Όπως και στην μονοδιάστατη περίπτωση, για κάθε δείκτη  $i1, i2, \dots$ , ορίζουμε την επανάληψη  $I1, I2, \dots$ , κλπ.
- Όπως και στην μονοδιάστατη περίπτωση, ισχύει:
$$\begin{aligned}i1 &= p1 + I1 * s1 \\i2 &= p2 + I2 * s2 \\&\dots\dots\dots \\in &= pn + In * sn\end{aligned}$$
- **Χώρος επαναλήψεων:** το σύνολο όλων των τιμών που μπορεί να πάρει το διάνυσμα  $\mathbf{I} = [I1, I2, \dots, In]$ .
- **Πυρήνας:** το περιεχόμενο του πιο εσωτερικού βρόχου.

# Λεξικογραφική διάταξη στιγμιοτύπων

- Ένα στιγμιότυπο **i** **προηγείται** ενός στιγμιοτύπου **j** αν κατά τη σειριακή εκτέλεση του βρόχου το **i** εκτελείται **πριν** από το **j**. Γράφουμε:  
**i** prec **j**
- Ένα στιγμιότυπο **i** **έπεται** ενός στιγμιοτύπου **j** αν κατά τη σειριακή εκτέλεση του βρόχου το **i** εκτελείται **μετά** από το **j**. Γράφουμε:  
**i** succ **j**
- Για να βρούμε αν ένα διάνυσμα-στιγμιότυπο **i** προηγείται ή έπεται ενός διανύσματος-στιγμιοτύπου **j** κάνουμε λεξικογραφική σύγκριση.

# Λεξικογραφική σύγκριση διανυσμάτων

- Το διάνυσμα  $[i_1, i_2, \dots, i_n]$  προηγείται λεξικογραφικά του διανύσματος  $[j_1, j_2, \dots, j_n]$ .
  - $\text{Av} ( i_1 < j_1 )$
  - $\text{ή} ( i_1 = j_1 \text{ και } i_2 < j_2 )$
  - $\text{ή} ( i_1 = j_1 \text{ και } i_2 = j_2 \text{ και } i_3 < j_3 )$
  - .....
  - $\text{ή} ( i_1 = j_1 \text{ και } i_2 = j_2 \text{ και } \dots \text{ και } i_{n-1} = j_{n-1} \text{ και } i_n < j_n )$
- Ακριβώς όπως στο λεξικό, όπου
  - η λέξη «**Αυγή**» προηγείται του «**Αυγό**» και.
  - η λέξη «**Μ**έρα» προηγείται της «**N**ύχτα».

# Λεξικογραφική σύγκριση διανυσμάτων

- Το διάνυσμα  $[i_1, i_2, \dots, i_n]$  προηγείται λεξικογραφικά του διανύσματος  $[j_1, j_2, \dots, j_n]$ .
  - $\text{Av} ( i_1 < j_1 )$
  - $\dot{\vee} ( i_1 = j_1 \text{ και } i_2 < j_2 )$
  - $\dot{\vee} ( i_1 = j_1 \text{ και } i_2 = j_2 \text{ και } i_3 < j_3 )$
  - .....
  - $\dot{\vee} ( i_1 = j_1 \text{ και } i_2 = j_2 \text{ και } \dots \text{ και } i_{n-1} = j_{n-1} \text{ και } i_n < j_n )$

## Παραδείγματα:

- $[2, 8, 12] \text{ prec } [3, 0, 0]$
- $[2, 8, 12] \text{ prec } [2, 9, 0]$
- $[2, 8, 12] \text{ prec } [2, 8, 15]$
- $[2, 8, 12] \text{ succ } [2, 8, 0]$
- $[2, 8, 12] \text{ succ } [2, 7, 20]$
- $[2, 8, 12] \text{ succ } [1, 9, 0]$

# Σχεδιασμός Γράφου Εξάρτησης

- **Ορισμός:** Γράφος Εξάρτησης (ΓΕ) [Dependence Graph (DG)] είναι ο γράφος με:
  - κόμβους τις πράξεις που πρέπει να εκτελεστούν και
  - ακμές που αναπαριστούν τις εξαρτήσεις μεταξύ των κόμβων.
- Υπάρχουν 3 πιθανά είδη εξαρτήσεων μεταξύ δύο κόμβων A και B:
  - **Εξαρτήσεις ροής** που προκύπτουν από κινδύνους read-after-write (RAW).
  - **Αντιεξαρτήσεις**, που προκύπτουν από κινδύνους write-after-read (WAR).
  - **Εξαρτήσεις εξόδου**, που προκύπτουν από κινδύνους write-after-write (WAW).

# Εξαρτήσεις Ροής

- Έχουμε **εξάρτηση ροής** μεταξύ δύο στιγμιοτύπων **i, j** όταν:
  - Το **i** γράφει σε μια θέση μνήμης **X** απ' όπου το **j** διαβάζει,
  - Το στιγμιότυπο **i** προηγείται του **j**.
- Κίνδυνος Read-After-Write: Αν το **j** εκτελεστεί πριν από το **i** τότε το **j** θα διαβάσει εσφαλμένη τιμή του **X**
- Ορισμός. Εφόσον δύο στιγμιότυπα **i, j** συνδέονται με εξάρτηση ροής τότε ορίζουμε ως **διάνυσμα εξάρτησης** τη διαφορά:

$$\mathbf{d} = \mathbf{j} - \mathbf{i}$$

# Παράδειγμα: Εξαρτήσεις Ροής

```
for (i=0; i<=5; i++) {  
    A[i+2] = 2 * A[i] - 1;  
}
```

Εξάρτηση ροής **d=2**

Εξάρτηση ροής **d=2**

Εξάρτηση ροής **d=2**

Εξάρτηση ροής **d=2**

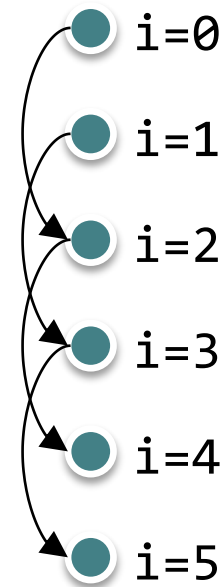
Εετύλιγμα βρόχου:

| Στιγ. | Πράξεις (εργασίες) |
|-------|--------------------|
| i=0   | A[2]=2*A[0]-1;     |
| i=1   | A[3]=2*A[1]-1;     |
| i=2   | A[4]=2*A[2]-1;     |
| i=3   | A[5]=2*A[3]-1;     |
| i=4   | A[6]=2*A[4]-1;     |
| i=5   | A[7]=2*A[5]-1;     |

# Παράδειγμα: Εξαρτήσεις Ροής

```
for (i=0; i<=5; i++) {  
    A[i+2] = 2 * A[i] - 1;  
}
```

Γράφος εξάρτησης



# Αντιεξαρτήσεις

- Έχουμε **αντιεξάρτηση** μεταξύ δύο στιγμιοτύπων **i**, **j** όταν:
  - Το **i** γράφει σε μια θέση μνήμης **X** απ' όπου το **j** διαβάζει και
  - Το στιγμιότυπο **i** έπεται του **j**.
- Κίνδυνος **Write-After-Read**: Αν το **i** εκτελεστεί πριν από το **j** τότε το **j** θα διαβάσει εσφαλμένη τιμή του **X**.
- **Ορισμός**: Εφόσον δύο στιγμιότυπα **i**, **j** συνδέονται με εξάρτηση ροής τότε ορίζουμε ως **διάνυσμα απόστασης της αντι-εξάρτησης** τη διαφορά:
$$d^* = i - j$$

# Παράδειγμα: Αντιεξαρτήσεις

```
for (i=0; i<=5; i++) {  
    A[i] = A[i + 2] + 1;  
}
```

Αντιεξάρτηση  $d^*=2$

Αντιεξάρτηση  $d^*=2$

Αντιεξάρτηση  $d^*=2$

Αντιεξάρτηση  $d^*=2$

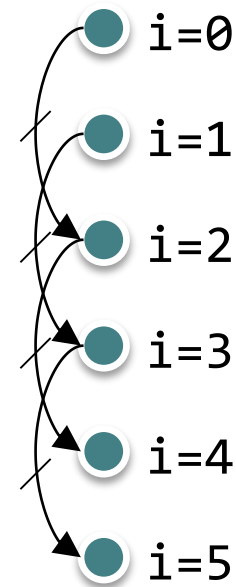
Εετύλιγμα βρόχου:

| Στιγ. | Πράξεις (εργασίες) |
|-------|--------------------|
| i=0   | A[0]=A[2]+1;       |
| i=1   | A[1]=A[3]+1;       |
| i=2   | A[2]=A[4]+1;       |
| i=3   | A[3]=A[5]+1;       |
| i=4   | A[4]=A[6]+1;       |
| i=5   | A[5]=A[7]+1;       |

# Παράδειγμα: Αντιεξαρτήσεις

```
for (i=0; i<=5; i++) {  
    A[i] = A[i + 2] + 1;  
}
```

Γράφος εξάρτησης



# Εξαρτήσεις ροής και αντιεξαρτήσεις

- Και στις δύο περιπτώσεις το στιγμιότυπο **i** γράφει ενώ το στιγμιότυπο **j** διαβάζει.
- Και στις δύο περιπτώσεις για να υπάρχει εξάρτηση (είτε εξάρτηση ροής είτε αντιεξάρτηση) πρέπει το **i** να γράφει στην ίδια θέση απ' όπου διαβάζει το **j**.
- **Η μόνη διαφορά είναι ότι:**
  - Αν το **i** προηγείται του **j** τότε μιλάμε για εξάρτηση ροής
  - Αν το **j** προηγείται του **i** τότε μιλάμε για αντιεξάρτηση

# Εξαρτήσεις εξόδου

- Έχουμε **εξάρτηση εξόδου** μεταξύ δύο στιγμιοτύπων **i, j** όταν:
  - Το **i** και το **j** γράφουν στην ίδια θέση μνήμης **X** και
  - Το στιγμιότυπο **i** προηγείται του **j**
- Κίνδυνος **Write-After-Write**: Αν το **j** εκτελεστεί πριν από το **i** τότε το **i** θα εγγράψει τελευταίο και στη μνήμη θα παραμείνει τελικά η εσφαλμένη τιμή του **X**.
- Σπάνια περίπτωση καθώς διαδοχικές εγγραφές στην ίδια τιμή δεν συνηθίζονται (στους περισσότερους αλγορίθμους).

# Παράδειγμα: Εξαρτήσεις εξόδου

```
for (i=0; i<=5; i++) {  
    A[i]=B[i]+1;  
    A[i+1]=g(i);  
}
```

Εξ. εξόδου  $d=1$

Εξ. εξόδου  $d=1$

Εξ. εξόδου  $d=1$

Εξ. εξόδου  $d=1$

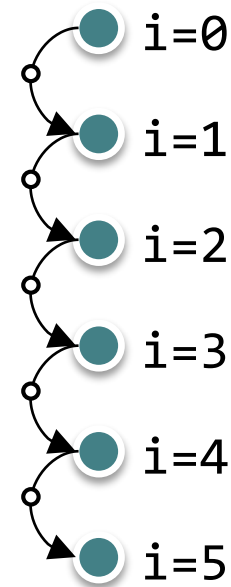
Εξ. εξόδου  $d=1$

| Στιγ. | Πράξεις (εργασίες)         |
|-------|----------------------------|
| i=0   | A[0]=B[0]+1;<br>A[1]=g(0); |
| i=1   | A[1]=B[1]+1;<br>A[2]=g(1); |
| i=2   | A[2]=B[2]+1;<br>A[3]=g(2); |
| i=3   | A[3]=B[3]+1;<br>A[4]=g(3); |
| i=4   | A[4]=B[4]+1;<br>A[5]=g(4); |
| i=5   | A[5]=B[5]+1;<br>A[6]=g(5); |

# Παράδειγμα: Εξαρτήσεις εξόδου

```
for (i=0; i<=5; i++) {  
    A[i]=B[i]+1;  
    A[i+1]=g(i);  
}
```

Γράφος εξάρτησης



## Παράδειγμα 1: Εξαρτήσεις σε εμφωλευμένους βρόχους

```
for (i1=0; i1<=5; i1++) {  
    for (i2=0; i2<4; i2++) {  
        A[i1+1,i2]=2*A[i1,i2];  
    }  
}
```

- Στιγμιότυπο  $\mathbf{k}=[k1,k2]$ : Εγγραφή  
 $A[k1+1,k2]=2*A[k1,k2];$
- Στιγμιότυπο  $\mathbf{j}=[j1,j2]$ : Ανάγνωση  
 $A[j1+1,j2]=2*A[j1,j2];$
- Για να υπάρχει εξάρτηση (είτε εξάρτηση ροής είτε αντεξάρτηση) πρέπει:  
 $[k1+1,k2]=[j1,j2]$

## Παράδειγμα 1: Εξαρτήσεις σε εμφωλευμένους βρόχους

- Στη συγκεκριμένη περίπτωση έχουμε
$$[k1+1, k2] = [j1, j2]$$

- δηλαδή

$$k1+1=j1, \quad k2=j2$$

$$k1=j1-1, \quad k2=j2$$

- οπότε

$$\mathbf{k1} < \mathbf{j1}, \quad k2=j2$$

- Συνεπώς το στιγμιότυπο  $\mathbf{k}$  προηγείται του  $\mathbf{j}$  και άρα έχουμε μια εξάρτηση ροής:

$$\mathbf{d} = \mathbf{j} - \mathbf{k} = [j1 - k1, j2 - k2] = [1, 0]$$

## Παράδειγμα 2: Εξαρτήσεις σε εμφωλευμένους βρόχους

```
for (i1=0; i1<=10; i1++) {  
    for (i2=0; i2<=12; i2++) {  
        A[i1,i2+1]=2*A[i1+1,i2];  
    }  
}
```

- Στιγμιότυπο  $\mathbf{i}=[i1,i2]$ : Εγγραφή  
 $A[i1,i2+1]=2*A[i1+1,i2];$
- Στιγμιότυπο  $\mathbf{j}=[j1,j2]$ : Ανάγνωση  
 $A[j1,j2+1]=2*A[j1+1,j2];$
- Για να υπάρχει εξάρτηση (είτε εξάρτηση ροής είτε αντεξάρτηση) πρέπει:  
 $[i1,i2+1]=[j1+1,j2]$

## Παράδειγμα 2: Εξαρτήσεις σε εμφωλευμένους βρόχους

- Στη συγκεκριμένη περίπτωση έχουμε
$$[i1, i2+1] = [j1+1, j2]$$

- δηλαδή

$$i1 = j1+1, \quad i2+1 = j2$$

$$i1 = j1+1, \quad i2 = j2-1$$

- οπότε

$$i1 > j1, \quad i2 < j2$$

- Συνεπώς το στιγμιότυπο  $\mathbf{i}$  έπεται του  $\mathbf{j}$  και άρα έχουμε μια αντιεξάρτηση:

$$\mathbf{d}^* = \mathbf{i} - \mathbf{j} = [i1 - j1, i2 - j2] = [1, -1]$$