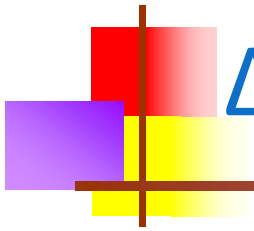


Διαχρονικές Δομές Δεδομένων

Χρονικές Βάσεις Δεδομένων

Βάσεις Δεδομένων Χρόνου Δοσοληψίας (Transaction Time)

Διατηρούν την ιστορία της δραστηριότητάς τους. Κάθε δοσοληψία με τη Βάση Δεδομένων προσδιορίζεται με μία χρονική ένδειξη. Είναι εφικτή η προσπέλαση διαφορετικών χρονικών στιγμιοτύπων της Βάσης Δεδομένων.

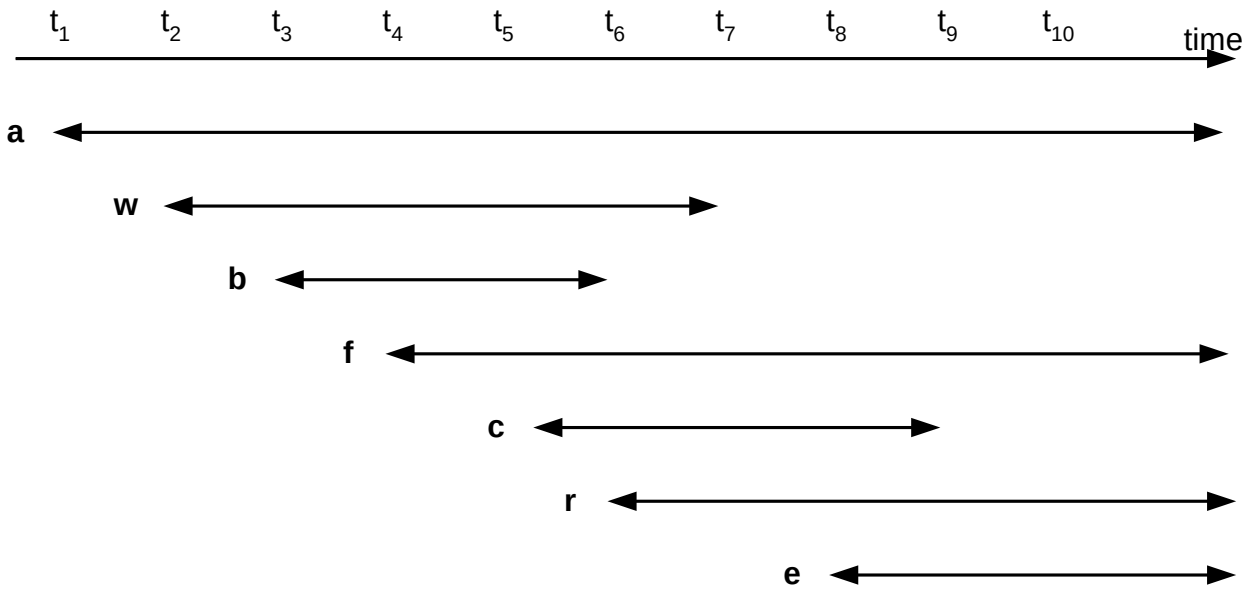
Βάσεις Δεδομένων Ισχύοντος Χρόνου (Valid Time)

Διατηρούν αντικείμενα που μία συνιστώσα τους υποδηλώνει χρονικά στιγμιότυπα. Το σύνολο των τιμών αυτών αποθηκεύουν τις τωρινές γνώσεις για το παρόν, το παρελθόν ή ακόμα και το μέλλον των αποθηκευμένων αντικειμένων.

Διχρονικές Βάσεις Δεδομένων (Bitemporal)

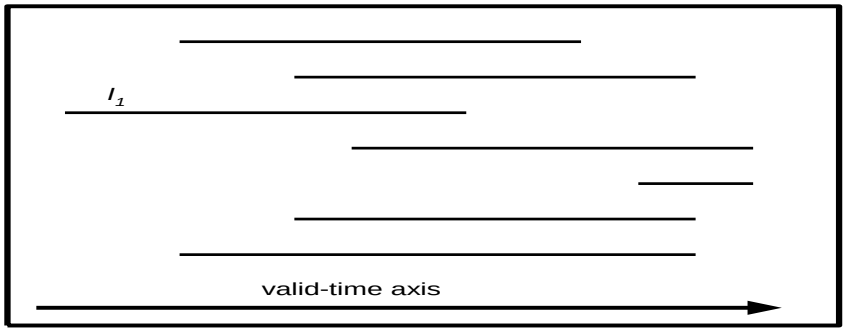
Αποτελούν συνδυασμό των δύο παραπάνω κατηγοριών. Αναπαριστούν την πραγματικότητα επιρέποντας είτε αναδρομικές είτε εκ των υστέρων αλλαγές.

Βάσεις Δεδομένων Χρόνου Δοσοληψίας

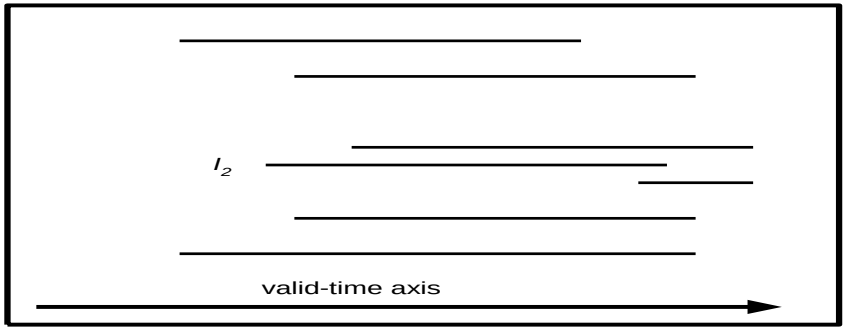


Χρονική Εξέλιξη

Βάσεις Δεδομένων Ισχύοντος Χρόνου



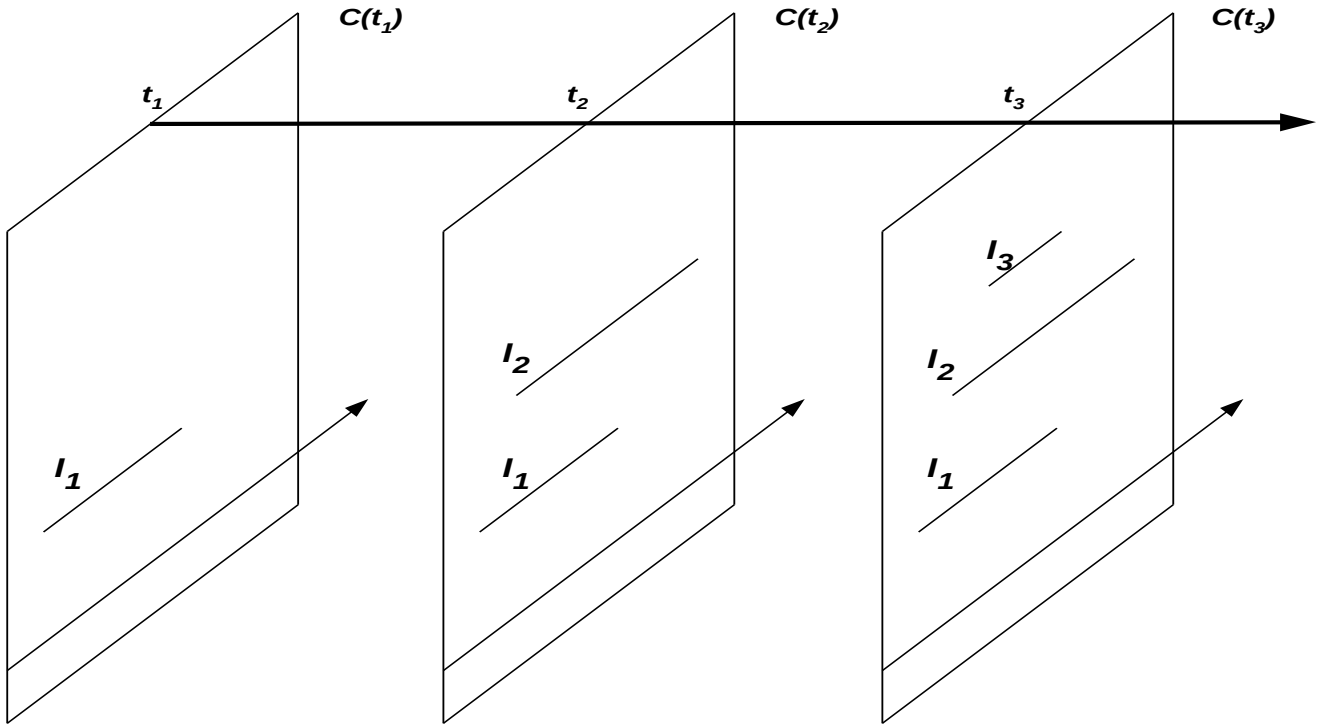
Old Collection



New Collection

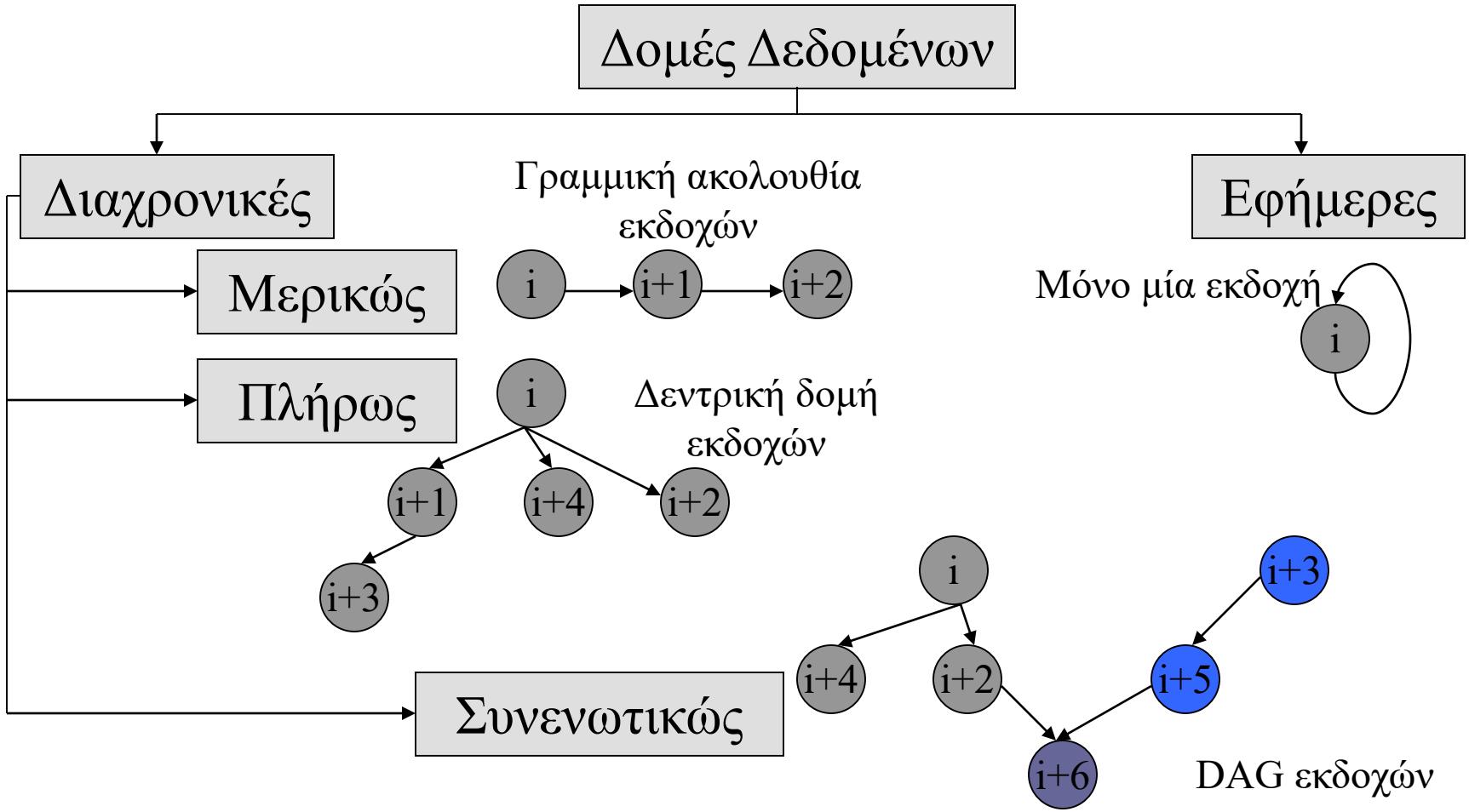
Χρονική Εξέλιξη

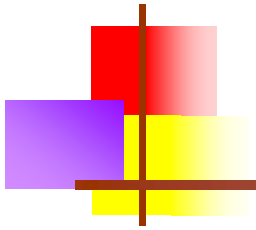
Διχρονικές Βάσεις Δεδομένων



Χρονική Εξέλιξη

Διαχρονικότητα-Έννοιες

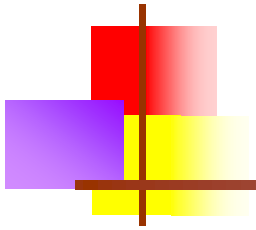




Εισαγωγή(1/3)

- **Εφήμερες δομές:**
  - Δεν έχουμε πρόσβαση στις προηγούμενες καταστάσεις (εκδοχές) της δομής.
  - Η παλιά έκδοση καταστρέφεται μετά από μία αλλαγή.

- **Διαχρονικές Δομές - Persistent:**
  - Επιτρέπει την πρόσβαση σε όλες τις εκδοχές της δομής.
  - Που χρειαζόμαστε τις διαχρονικές δομές;
    - Επεξεργασία κειμένου και αρχείων.
    - Υπολογιστική Γεωμετρία κ.α.



Εισαγωγή(2/3)

- **Μερική Διαχρονικότητα – Partial Persistent:**
  - Επιτρέπει την πρόσβαση σε όλες τις εκδοχές της δομής, αλλά είναι δυνατόν να τροποποιηθεί μόνο η τελευταία εκδοχή.

- **Πλήρης Διαχρονικότητα – Fully Persistent:**
  - Επιτρέπει τόσο την πρόσβαση όσο την τροποποίηση σε όλες τις εκδοχές της δομής.

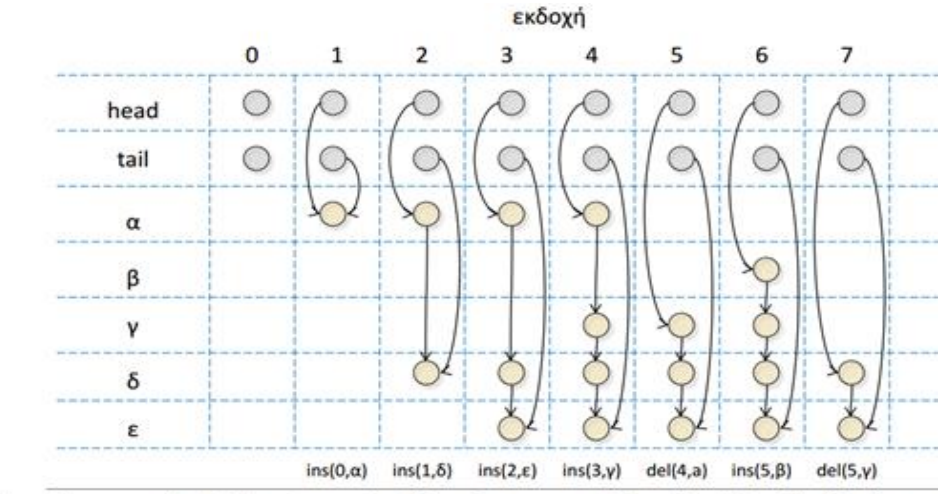
Εισαγωγή(3/3)

- Για να υποστηρίξουμε τη διαχρονικότητα της δομής εισάγουμε την ακόλουθη παράμετρο :
  - **Αριθμός εκδοχής:**
    - Η αρχική δομή αποτελεί την εκδοχή 0.
    - Η i-οστή πράξη τροποποίησης δημιουργεί την εκδοχή i.

- **Παράμετροι απόδοσης:**
  - n = αριθμός στοιχείων στην τρέχουσα εκδοχή
  - m = συνολικός αριθμός τροποποιήσεων

Απλοϊκές λύσεις

1. Κάθε εκδοχή δημιουργεί ένα π αντίγραφο.
   - Απαιτεί Ω(n) χρόνο και χώρο ανά

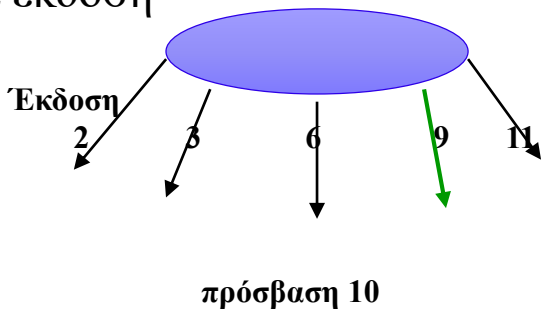


http://www.cs.uoi.gr/~loukas/courses/grad/Data_Structures_and_Algorithms/index.files/Persistence.

2. Δεν αποθηκεύουμε καμία εκδοχή, αλλά μόνο την ακολουθία τροποποιήσεων. Για πρόσβαση στην i-οστή εκδοχή κατασκευάζουμε τη δομή έως αυτή την εκδοχή από την αρχή.
   - Απαιτεί Ω(i) χρόνο για την κάθε πρόσβαση στην i-οστή εκδοχή.
3. Συνδυασμός των δύο παραπάνω

Μερική Διαχρονικότητα(1/2)

- **Μέθοδος παχιού κόμβου (Fat Node)**
  - Κάθε κόμβος αποθηκεύει αυθαίρετο αριθμό από τιμές (ετικέτες έκδοσης)
  - Έχει ετικέτα που υποδεικνύει την έκδοση που δημιουργήθηκε ο ίδιος
  - Εύρεση της i-οστής έκδοσης:
    - Ξεκινάμε από τον κατάλληλο δείκτη πρόσβασης της έκδοσης
    - Όταν βρισκόμαστε σε ένα διαχρονικό κόμβο P και θέλουμε να ανακτήσουμε μία τιμή ενός πεδίου αναζητούμε την τιμή του πεδίου με μέγιστη ετικέτα <=i.
      - O(log m) χρόνος για m εκδόσεις αν οι τιμές εκδόσεων είναι οργανωμένες σε δυαδικό δέντρο)
      - Χωρική Επιβάρυνση: O(1) ανά έκδοση



Μερική Διαχρονικότητα(2/2)

- **Μέθοδος αντιγραφής κόμβων (Node Copying)**
  - Καλύτερη χρονική επιβάρυνση κατά την διάρκεια της προσπέλασης.
  - Σε αυτή την τεχνική, ο κάθε κόμβος έχει σταθερό χώρο.
  - Όταν γίνεται μία αλλαγή έκδοσης
    - Καταγράφεται στον κόμβο αν έχει χώρο
    - Αλλιώς δημιουργείται ένας νέος κόμβος που περιέχει μόνο την τελευταία έκδοση του κόμβου.
  - Πλέον η ιστορικότητα καταγράφεται σε λίστα από κόμβους.
  - Δημιουργούνται δείκτες από τους προηγούμενους- προγόνους στον νέο κόμβο.
    - Αν δεν έχουν χώρο δημιουργούνται αντίγραφα.
  - Χρονικό κόστος:
    - Αναζήτηση και Ενημέρωση O(1) επιμερισμένη
    - Μετακίνηση στις εκδόσεις είναι O(1)
  - Χωρική Επιβάρυνση:
    - O(1)

Πλήρης Διαχρονικότητα

- **Χρήση μεθόδου fat node**
  - Χρονικό κόστος:
    - O(log m) επιβάρυνση για πρόσβαση
    - και O(1) για ενημέρωση
  - Χωρικό κόστος: O(1)
- **Χρήση μεθόδου διάσπασης κόμβου**
  - Χρονικό κόστος:
    - O(1) επιμερισμένη επιβάρυνση για αναζήτηση
    - και O(1) για ενημέρωση
  - Χωρικό κόστος: O(1)

Μετατροπή Εφήμερης Δομής σε Μερικώς Διαχρονική ([Driscoll et al. 89])

Δύο γενικές μέθοδοι για διασυνδεμένες δομές :

α) fat-node

Χώρος=O(1)/ενημερ.

Χρόνος=O(logm)/ενημερ.ή προσπέλαση

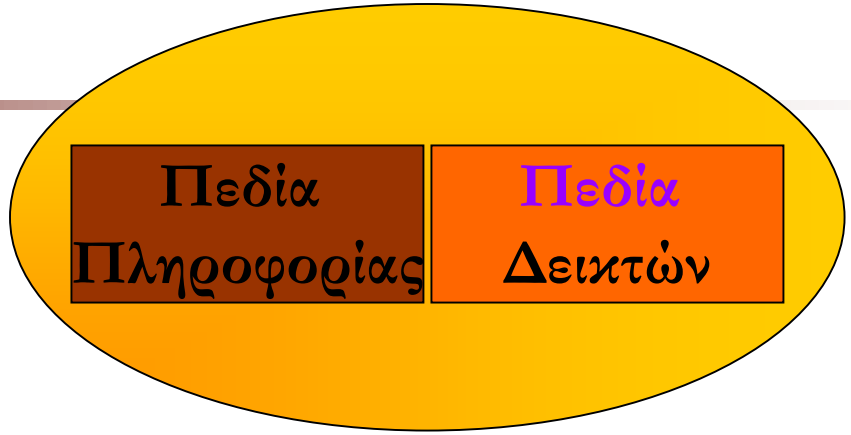
β) node-copying

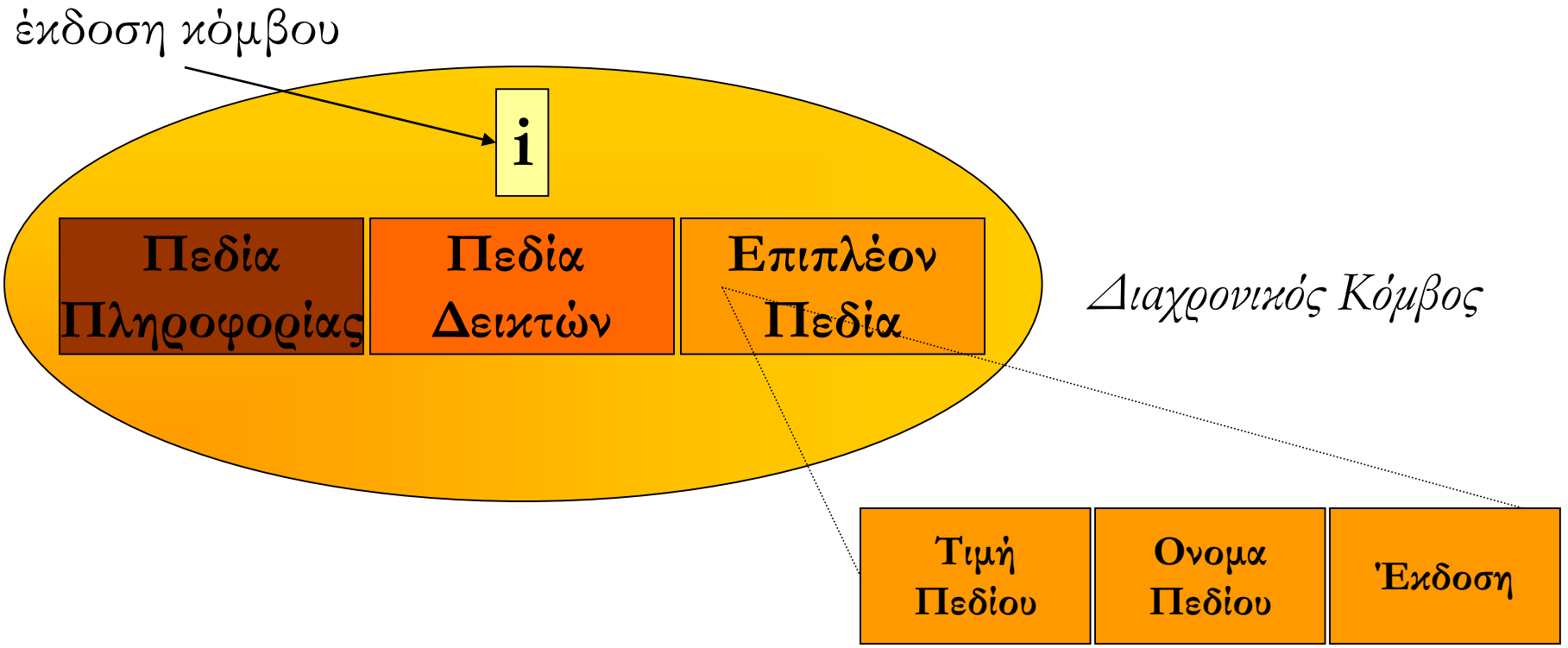
Επιμερισμένος Χώρος=O(1)/ενημέρωση

Επιμερισμένος Χώρος=O(1)/ενημέρωση

- πεπερασμένο σύνολο κόμβων με σταθερό αριθμό πεδίων (δεδομένα και δείκτες)
- σταθερό αριθμό κόμβων εισόδου

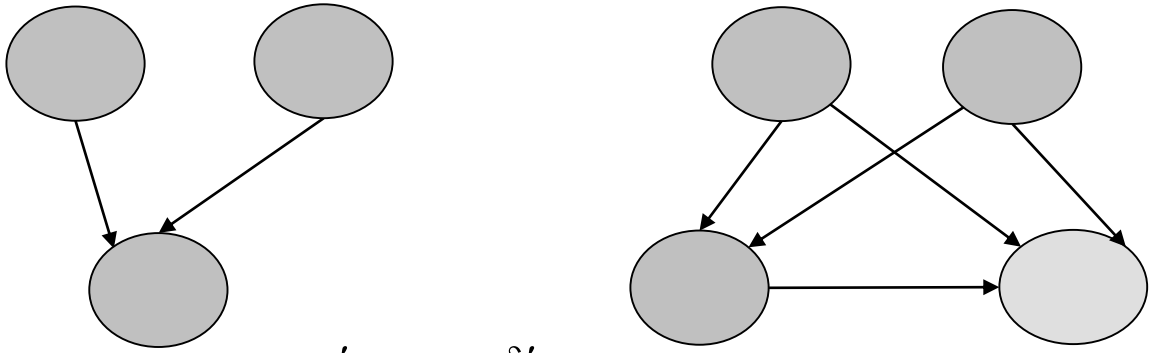
Fat Node Μέθοδος





Node-Copying Μέθοδος

Κόμβοι περιορισμένης χωρητικότητας: όταν ο κόμβος γεμίσει δημιουργούμε νέο αντίγραφο του κόμβου



ενημέρωση πεδίου

Επειδή πρέπει να είναι εφικτή η προσπέλαση ενός νέου κόμβου πρέπει να αντιγράψουμε τους άμεσους προγόνους του $\Rightarrow$ **η μέθοδος δίνει ένα σταθερό επιμερισμένο κόστος όταν ο βαθμός εισόδου κάθε κόμβου είναι φραγμένος, με την υπόθεση ότι ο επιπλέον αριθμός των προστιθεμένων φύλλων είναι αρκετά μεγάλος**

Ιστορία του Προβλήματος

- Στην εργασία *[Driscoll et al.89]* παρουσιάστηκε μία γενική μεθοδολογία για μετατροπή κάθε διασυνδεδεμένης φραγμένου βαθμού εφήμερης δομής σε πλήρως διαχρονική με *O(1)* επιμερισμένο κόστος ανά ενημέρωση και *O(1)* κόστος χειρότερης περίπτωσης ανά προσπέλαση.

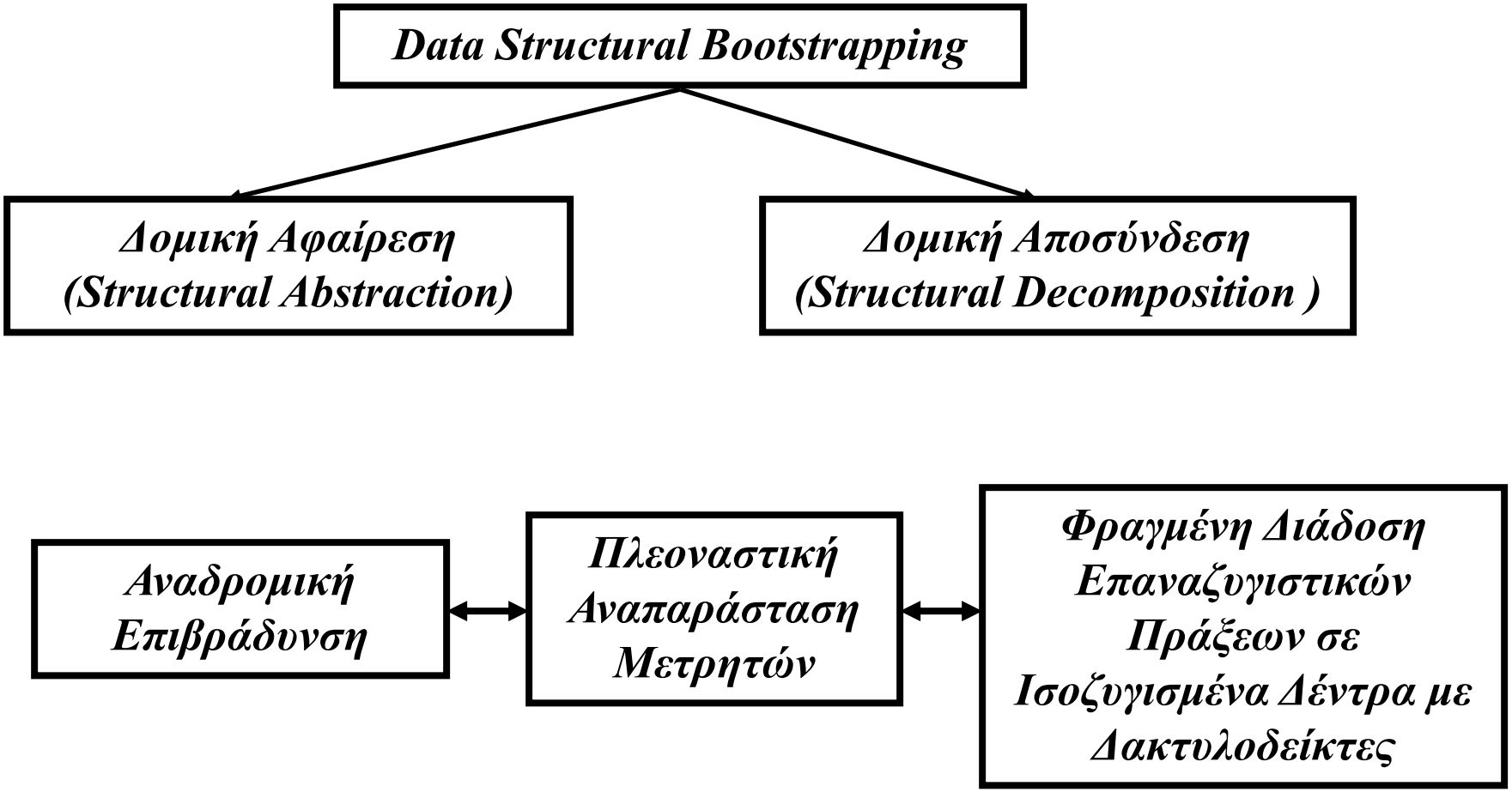
- Στην εργασία *[Dietz 89]* παρουσιάστηκε μία γενική μεθοδολογία για μετατροπή κάθε εφήμερης δομής, στο RAM μοντέλο υπολογισμού σε πλήρως διαχρονική με *O(loglogm)* επιμερισμένο κόστος ανά πράξη.

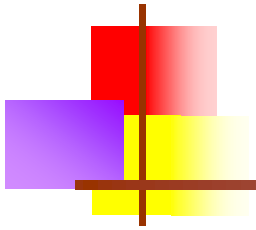
- Στην εργασία *[Driscoll et al.94]* παρουσιάστηκε ένας αλγόριθμος για τη δημιουργία πλήρως διαχρονικών *steques* με *O(1)* κόστος για κάθε πράξη εκτός από την *catenate (O(loglogm)*) επιμερισμένο κόστος ανά πράξη.

- Στην εργασία *[Buchsbaum &Tarjan 95]* παρουσιάστηκε ένας αλγόριθμος για τη δημιουργία συνενωτικώς διαχρονικών *deques* με *O(1)* κόστος για κάθε πράξη, εκτός από τις *pop*,*eject* (*O(log*m))*

- Στην εργασία *[Kaplan &Tarjan95]* παρουσιάστηκε μία μέθοδος για τη δημιουργία συνενωτικώς διαχρονικών *deques* με *O(1)* κόστος ανά πράξη .

Νέες Τεχνικές





Διαχρονικές Δομές Δεδομένων

Ερωτήσεις?