

Ατομικά Αντικείμενα



ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΕΑΕΚ



ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ
ΣΥΓΧΡΗΜΑΤΟΔΟΤΗΣΗ
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Η ΠΑΙΔΕΙΑ ΣΤΗΝ ΚΟΡΥΦΗ
Επιχειρησιακό Πρόγραμμα
Εκπαίδευσης και Αρχικής
Επαγγελματικής Κατάρτισης

ΑΤΟΜΙΚΑ ΑΝΤΙΚΕΙΜΕΝΑ

Ένα ατομικό αντικείμενο κάποιου τύπου μοιάζει με μια κοινή μεταβλητή αυτού του τύπου.

Ένα ατομικό αντικείμενο βρίσκεται σε μια κατάσταση και υποστηρίζει ένα σύνολο από λειτουργίες μέσω των οποίων μπορεί να αλλάξει η κατάσταση του.

Τα ατομικά αντικείμενα έχουν προταθεί σαν δομικά μπλοκ για την κατασκευή πολυ-επεξεργαστικών συστημάτων:

✘ Εκκίνηση με απλά βασικά ατομικά αντικείμενα που παρέχονται από το υλικό.

✘ Διαδοχική κατασκευή όλο και πιο πολύπλοκων ατομικών αντικειμένων από πιο απλά αντικείμενα.

Το σύστημα που προκύπτει είναι απλό, καλά δομημένο, και είναι εύκολο να αποδειχθεί ότι είναι ορθό.

ΒΑΣΙΚΑ ΑΤΟΜΙΚΑ ΑΝΤΙΚΕΙΜΕΝΑ

Καταχωρητής Read/Write

Αποθηκεύει μια τιμή από κάποιο σύνολο και υποστηρίζει δύο λειτουργίες:

- ✘ $\text{read}(R)$: επιστρέφει την τιμή που είναι αποθηκευμένη στον R χωρίς να τον αλλάξει,
- ✘ $\text{write}(R,v)$: γράφει την τιμή v στον R και επιστρέφει ack

Καταχωρητής Test-And-Set

Αποθηκεύει μια τιμή από το σύνολο $\{0,1\}$ και υποστηρίζει δύο λειτουργίες:

- ✘ $\text{read}(R)$: επιστρέφει την τιμή που είναι αποθηκευμένη στον R χωρίς να τον αλλάξει,
- ✘ $\text{Test-And-Set}(R)$: αποθηκεύει την τιμή 1 στον R και επιστρέφει την παλιά τιμή του

ΒΑΣΙΚΑ ΑΤΟΜΙΚΑ ΑΝΤΙΚΕΙΜΕΝΑ

Καταχωρητής **Load-Link, Store-Conditional (LL/SC)**

Αποθηκεύει μια τιμή από κάποιο σύνολο και υποστηρίζει δύο λειτουργίες LL και SC, όπου κάθε LL συσχετίζεται μια κάποια SC.

LL(R): επιστρέφει την τιμή του R,

SC(R,v): αποθηκεύει v στον R αν δεν έχει γίνει καμία αλλαγή στον R από την χρονική στιγμή που εκτελέστηκε η LL η οποία έχει συσχετισθεί με την SC.

Καταχωρητής **Compare-And-Swap**

Αποθηκεύει μια τιμή από κάποιο σύνολο και υποστηρίζει δύο λειτουργίες:

read(R): επιστρέφει την τιμή του R,

CAS(R,u,v): Αν η κατάσταση του R είναι u αλλάζει σε v και επιστρέφεται TRUE. Διαφορετικά δεν αλλάζει ο R και επιστρέφεται FALSE.

ΒΑΣΙΚΑ ΑΤΟΜΙΚΑ ΑΝΤΙΚΕΙΜΕΝΑ

Καταχωρητής Πολλαπλής-Εγγραφής (MW)

Όλες οι διεργασίες επιτρέπεται να γράφουν στον καταχωρητή

Καταχωρητής Απλής-Εγγραφής (SW)

Μόνο μια από τις διεργασίες μπορεί να γράφει στον καταχωρητή.

Χωρητικότητα καταχωρητή

Ένας καταχωρητής είναι πεπερασμένης χωρητικότητας όταν το σύνολο τιμών που μπορεί να αποθηκεύει είναι πεπερασμένο. Στην αντίθετη περίπτωση είναι μη-πεπερασμένης χωρητικότητας.

- ◆ Το υλικό πολλών συστημάτων υποστηρίζει κάποια από τα παραπάνω αντικείμενα. Το υλικό τότε εγγυάται ότι κάθε μια λειτουργία εκτελείται ατομικά.

ΑΤΟΜΙΚΑ ΣΤΙΓΜΙΟΤΥΠΑ ΜΝΗΜΗΣ

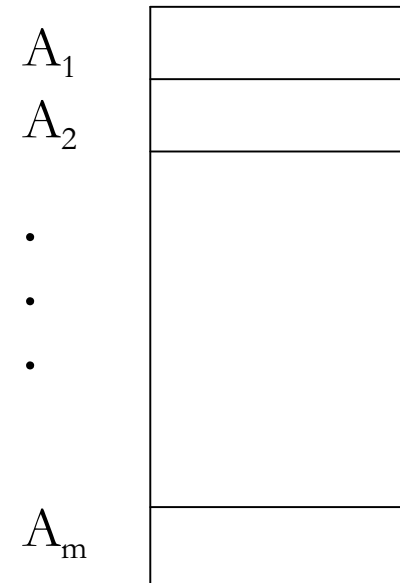
❖ Διαμοιραζόμενο αντικείμενο που αποτελείται από έναν πίνακα με m τμήματα που το καθένα αποθηκεύει μια τιμή.

❖ Υποστηρίζει δύο είδη ατομικών λειτουργιών:

✓ **UPDATE(i,v)**: εγγράφει στο τμήμα A_i του στιγμιότυπου την τιμή v .

✓ **SCAN**: επιστρέφει ένα διάνυσμα τιμών, μία για κάθε τμήμα και είναι εγγυημένο πως οι τιμές αυτές είναι συνεπείς.

🔗 Τα ατομικά στιγμιότυπα αποσκοπούν στην παροχή συνεπών όψεων.



Ατομικό Στιγμιότυπο A

ΑΤΟΜΙΚΑ ΣΤΙΓΜΙΟΤΥΠΑ

Ατομικό Στιγμιότυπο Πολλαπλής-Εγγραφής (MW Atomic Snapshot)

Κάθε διεργασία μπορεί να εκτελεί UPDATES σε κάθε τμήμα.

Στιγμιότυπο Απλής-Εγγραφής (SW Atomic Snapshot)

Μόνο η διεργασία i μπορεί να εκτελεί UPDATES στο τμήμα A_i .



Τα ατομικά στιγμιότυπα είναι πιο ισχυρά αντικείμενα από τους καταχωρητές και διευκολύνουν τη σχεδίαση κατανεμημένων αλγορίθμων!



Τα ατομικά στιγμιότυπα δεν παρέχονται από το υλικό.

ΕΠΑΛΗΘΕΥΣΗ ΚΑΘΟΛΙΚΩΝ ΚΑΤΗΓΟΡΗΜΑΤΩΝ (GLOBAL PREDICATE EVALUATION)

Σε πολλά προβλήματα των καταναμημένων συστημάτων μια ενέργεια πρέπει να συμβεί αν ένα κατηγορημα (predicate) που αφορά την καθολική κατάσταση του συστήματος είναι αληθές.

Παραδείγματα

- Ανίχνευση αδιεξόδων (deadlock detection).
- Ανίχνευση τερματισμού (termination detection).
- Ανίχνευση απώλειας διακριτικού (token loss detection).
- Διαχείριση μη-προσβάσιμης ή αχρησιμοποίητης μνήμης (garbage collection).
- Εισαγωγή σημείων ελέγχου και επανεκκίνηση (checkpointing & restarting).
- Παρακολούθηση και αποσφαλμάτωση (monitoring & debugging).

ΟΡΘΟΤΗΤΑ & ΑΠΟΣΦΑΛΜΑΤΩΣΗ

- ❌ Η μεγαλύτερη δυσκολία στο να αποδειχθεί η ορθότητα λειτουργίας ενός κατανεμημένου αλγορίθμου είναι η αναγκαιότητα στήριξης της επιχειρηματολογίας σε μη συνεπείς όψεις των διαμοιραζόμενων μεταβλητών.
- ✅ Ο υπολογισμός συνεπών όψεων διευκολύνει το έργο επαλήθευσης της ορθότητας των κατανεμημένων αλγορίθμων αλλά δεν είναι εύκολο πρόβλημα.

ΕΦΑΡΜΟΓΕΣ

- ✓ Αμοιβαίος αποκλεισμός [Katseff - STOC'78, Lamport – JACM'86, Dolev & Gafni & Shavit – STOC'88].
- ✓ Επίτευξη ομοφωνίας [Aspnes - J. of Alg.'93, Aspnes & Herlihy – J. of Alg '90] & Approximate Agreement [Attiya, Lynch & Shavit – JACM'94]
- ✓ Κατασκευή συστημάτων παράλληλης ανάθεσης χρονοσφραγίδων [Dolev & Shavit - STOC'88].
- ✓ Αξιόπιστη και αποδοτική υλοποίηση κατανεμημένων δομών δεδομένων [Aspnes & Herlihy - SPAA'90, Herlihy – PODC'91].
- ✓ Υλοποίηση χρήσιμων κατανεμημένων αντικειμένων [Vitanyi & Awerbuch - FOCS'86, Bloom – PODC'87, Peterson & Burns – FOCS'87].

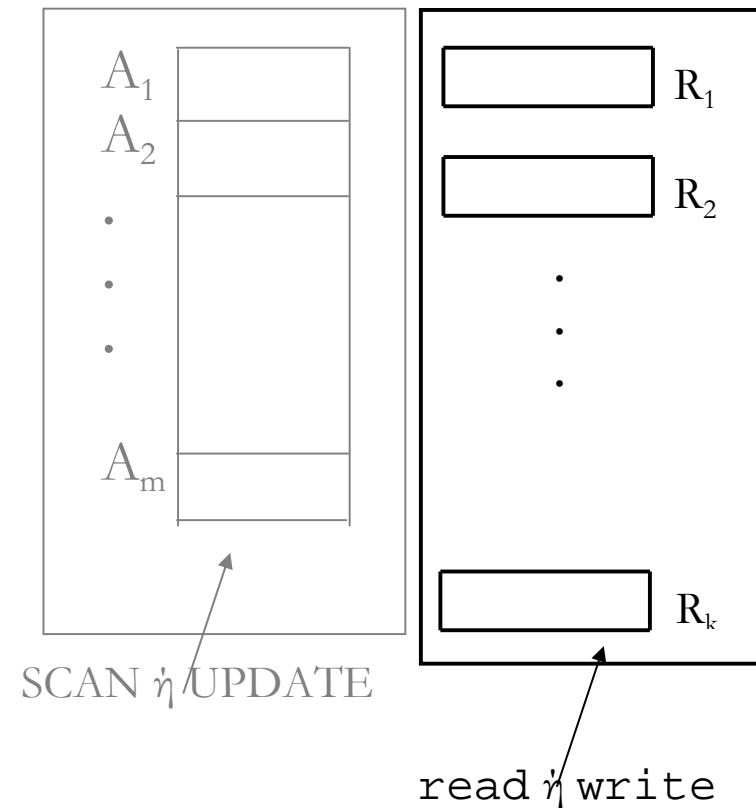
ΥΛΟΠΟΙΩΝΤΑΣ ΑΤΟΜΙΚΑ ΣΤΙΓΜΙΟΤΥΠΑ ΜΕ ΧΡΗΣΗ ΚΑΤΑΧΩΡΗΤΩΝ

? Παροχή αλγορίθμων για τις λειτουργίες
SCAN και UPDATE.

? Αποτελεσματικότητα Υλοποίησης

Χρονική πολυπλοκότητα μιας SCAN
ή μιας UPDATE λειτουργίας:
ο μέγιστος αριθμός βημάτων που
εκτελούνται από οποιαδήποτε διεργασία
σε οποιαδήποτε εκτέλεση προκειμένου
να υλοποιηθεί η λειτουργία

Χωρική Πολυπλοκότητα υλοποίησης:
καταχωρητών που χρησιμοποιούνται



Χρήσιμοι Ορισμοί

Διαφορετικά στιγμιότυπα των λειτουργιών του αντικειμένου μπορούν να εκτελούνται από διαφορετικές διεργασίες ταυτόχρονα.

Σε κάθε εκτέλεση, οι κλήσεις και οι αποκρίσεις των λειτουργιών του ατομικού αντικειμένου μπορεί να πραγματοποιούνται με οποιαδήποτε σειρά. Πολλές κλήσεις μπορούν π.χ. να συμβούν πριν να συμβεί οποιαδήποτε απόκριση.

Το διάστημα εκτέλεσης μιας λειτουργίας είναι το διάστημα από την κλήση της μέχρι την απόκρισή της. Τα διαστήματα εκτέλεσης των λειτουργιών ενός ατομικού στιγμιότυπου μπορεί να επικαλύπτονται (μερικώς ή ολικώς).

Μια ακολουθία από κλήσεις και αποκρίσεις είναι έγκυρη, αν κάθε κλήση μπορεί να συσχετισθεί με μια απόκριση.

ΤΕΡΜΑΤΙΣΜΟΣ & ΣΕΙΡΙΟΠΟΙΗΣΙΜΟΤΗΤΑ

Ιδιότητα Ελευθερίας Αναμονής (Wait-Freedom)

Κάθε μη-εσφαλμένη διεργασία εκτελεί έναν πεπερασμένο αριθμό βημάτων προκειμένου να διεκπεραιώσει τη λειτουργία της.

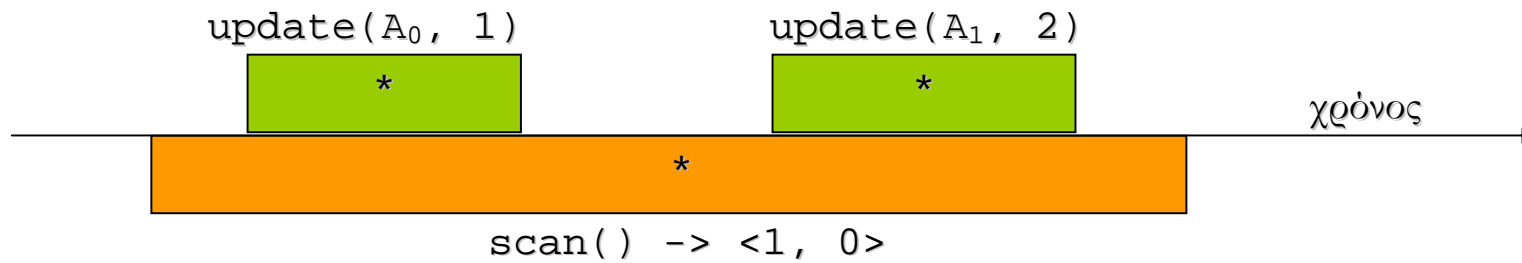
Σειριοποιησιμότητα (Linearizability)

Ακόμα και αν εκτελούνται παράλληλα λειτουργίες σε ένα διαμοιραζόμενο αντικείμενο, υπάρχει μία σειριακή εκτέλεση των λειτουργιών αυτών που έχει τα ίδια αποτελέσματα.

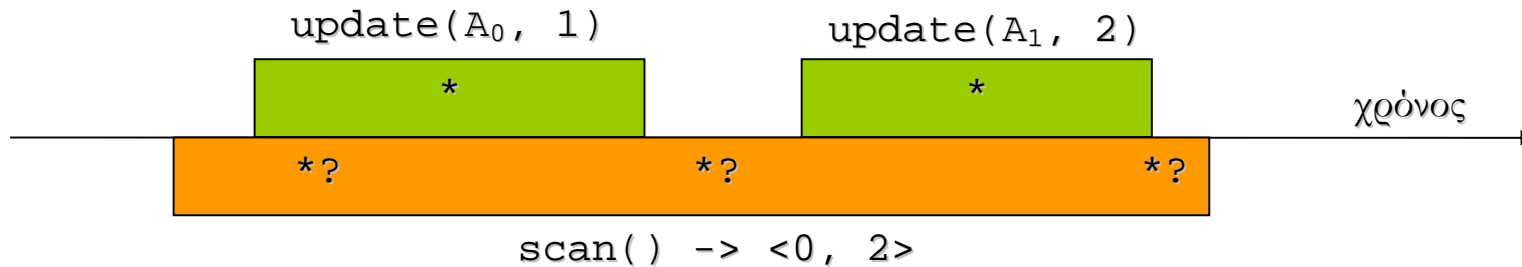
✓ Σε κάθε εκτέλεση, κάθε SCAN και UPDATE λειτουργία που εμπεριέχεται στην εκτέλεση επιστρέφει το ίδιο αποτέλεσμα σαν να εκτελέστηκε σειριακά σε κάποιο σημείο στο χρονικό διάστημα της εκτέλεσης της. Το σημείο αυτό λέγεται σημείο σειριοποίησης της λειτουργίας (linearization point).

ΣΕΙΡΙΟΠΟΙΗΣΙΜΟΤΗΤΑ

Παράδειγμα σειριοποιήσιμης εκτέλεσης



Παράδειγμα μη-σειριοποιήσιμης εκτέλεσης



Σειριοποιησιμότητα - Φορμαλιστικά

Έστω μια εκτέλεση β . Λέμε ότι η β είναι σειριοποιήσιμη αν τα αιόλουθα είναι δυνατά:

1. Για κάθε προσομοιούμενη λειτουργία π που έχει περατωθεί στη β , μπορούμε να επιλέξουμε ένα σημείο σειριοποίησης $*\pi$ κάπου μεταξύ της κλήσης της π και της απόκρισής της.
2. Μπορούμε να επιλέξουμε ένα υποσύνολο Φ των λειτουργιών των οποίων η εκτέλεση δεν έχει περατωθεί στη β τ.ω. για κάθε λειτουργία π στο Φ :
 - ο μπορούμε να εισάγουμε μια απόκριση, και
 - ο μπορούμε να επιλέξουμε ένα σημείο σειριοποίησης $*\pi$, το οποίο να βρίσκεται σε κάποιο σημείο της εκτέλεσης που έπεται της κλήσης της π .

Διαισθητικά:

«Κάθε λειτουργία μοιάζει σαν να εκτελέστηκε στιγμιαία κάποια χρονική στιγμή μέσα στο διάστημα εκτέλεσής της.»

«Αν και η εκτέλεση κάποιων λειτουργιών μπορεί να συμβαίνει ταυτόχρονα, η υλοποίηση πρέπει να εγγυάται ότι οι λειτουργίες λαμβάνουν χώρα σειριακά, μια κάθε φορά, και η σειριακή αυτή διάταξη θα πρέπει να σέβεται τις κλήσεις και αποκρίσεις των λειτουργιών.»

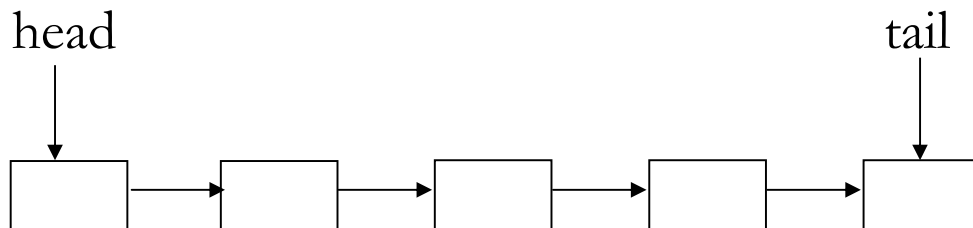
Σειριοποιησιμότητα – Περισσότερα Παραδείγματα

Μια ουρά FIFO Q υποστηρίζει δύο λειτουργίες:

$enq(Q,v)$: προσθέτει το στοιχείο v στο τέλος της ουράς Q

$deq(Q)$: αφαιρεί και επιστρέφει το πρώτο στοιχείο της ουράς Q

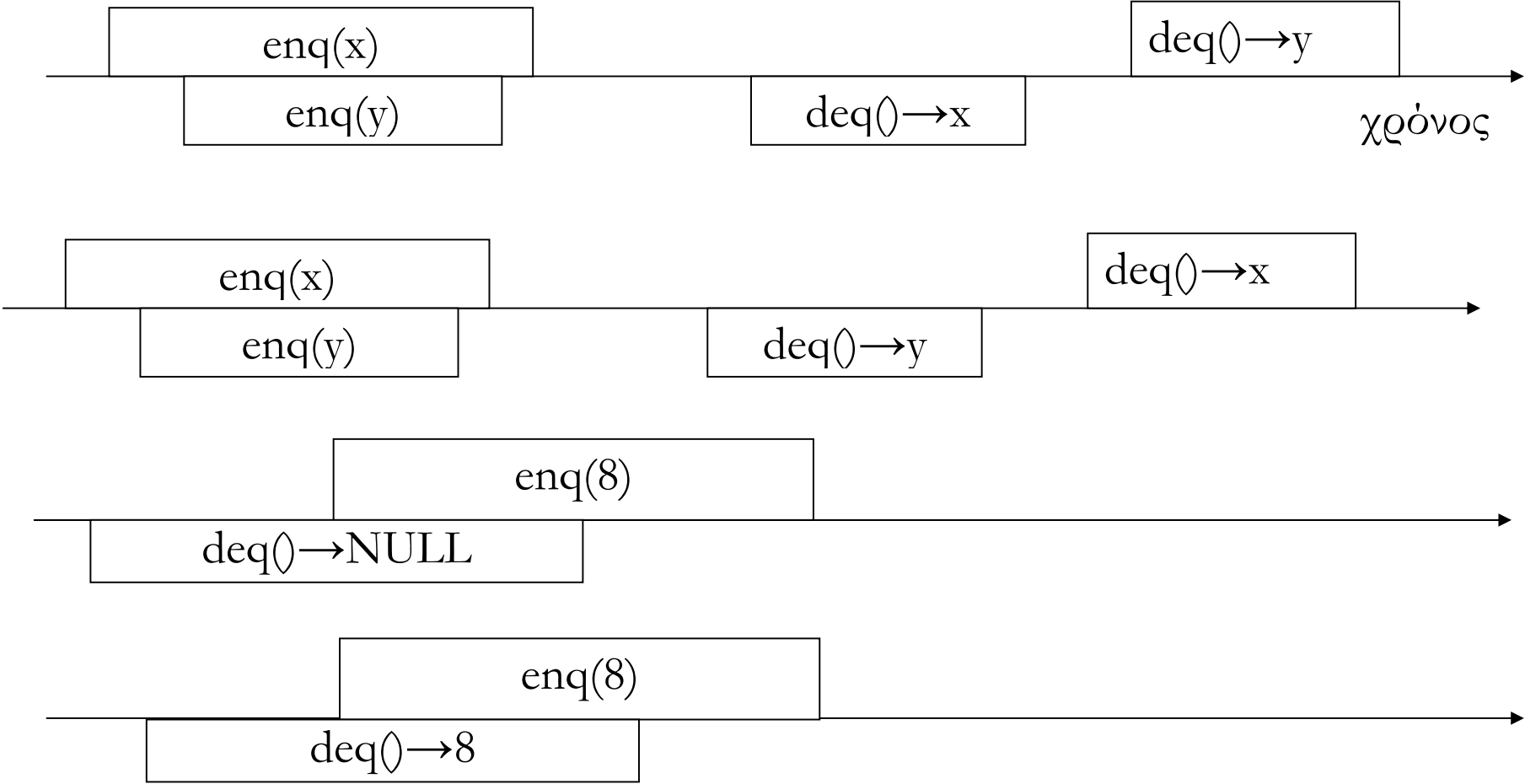
Σε μια παράλληλη ουρά πολλές διεργασίες μπορεί να προσπαθούν να εισάγουν (enq) και να εξάγουν (deq) στοιχεία ταυτόχρονα.



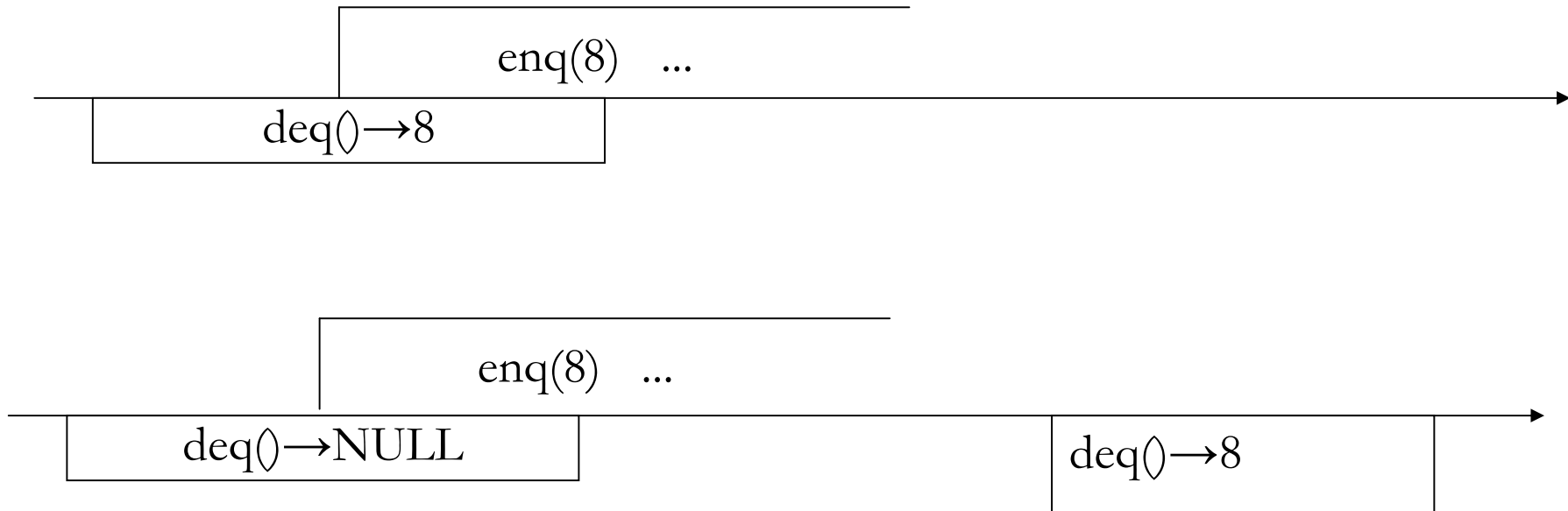
Ουρές μπορούν να υλοποιηθούν από απλούστερα αντικείμενα (π.χ., Test-And-Set και LL/SC registers).

Το ίδιο ισχύει και για άλλα αντικείμενα, π.χ., στοίβες, λίστες, λίστες παράλειψης, γράφους, κλπ.

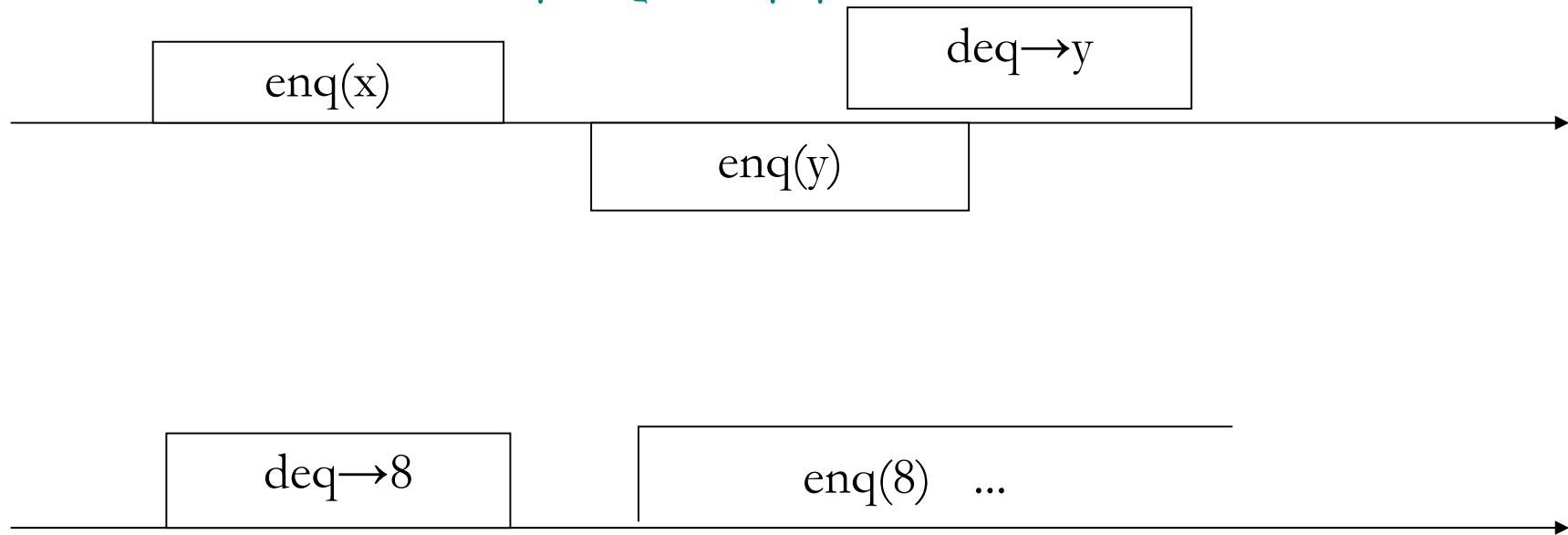
Σειριοποιήσιμες Εκτελέσεις



Σειριοποιήσιμες Εκτελέσεις



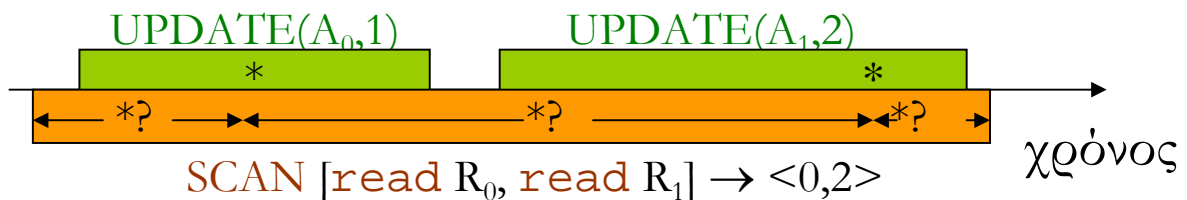
Μη-Σειριοποιήσιμες Εκτελέσεις



Ατομικά Στιγμιότυπα – Προφανής Λύση

- ✓ Είναι δύσκολο για μια διεργασία να καταφέρει να υπολογίσει μια συνεπή όψη όλων των καταχωρητών.

Λειτουργίες Διεργασιών		Τιμές Καταχ/τών	
P	Q	R ₀	R ₁
read R ₀ → 0	write(R ₀ ,1) write(R ₁ ,2)	0	0
read R ₁ → 2		1	0
		1	2



- ☹ Σε καμία χρονική στιγμή δεν συνυπάρχουν οι τιμές 0 και 2 στους καταχωρητές.

- 👉 Τα ατομικά στιγμιότυπα έχουν σειριοποιήσιμες, ελεύθερες-αναμονής υλοποιήσεις χρησιμοποιώντας καταχωρητές. [Afeke et. al-JACM'93, Anderson-DistComp'93, Aspnes&Herliby-SPAA'90]

ΥΛΟΠΟΙΗΣΕΙΣ ΑΤΟΜΙΚΩΝ ΣΤΙΓΜΙΟΤΥΠΩΝ

- ✘ Anderson – PODC'90, PODC'93, DISC'91
- ✘ Afek, Attiya, Dolev, Gafni, Merritt and Shavit – PODC'90 + JACM'93
- ✘ Attiya, Herlihy & Rachman - DISC'92
- ✘ Attiya & Rachman – PODC'93 + SICOMP'98
- ✘ Israeli & Shaham – PODC'92
- ✘ Israeli, Shaham & Shirazi – DISC'93
- ✘ Inoue, Chen, Masuzawa & Tokura – DISC'94
- ✘ Afek, Stupp & Touitou – FOCS'99
- ✘ Afek, Attiya, Fouren, Stupp & Touitou – PODC'99
- ✘ Fatourou, Fich & Ruppert – PODC'02, STOC'03, STOC'06
- ✘ Fatourou & Kallimanis – PODC'06

ΑΠΛΗ ΥΛ/ΣΗ SW Snapshot ΜΕ ΧΡΗΣΗ n SW ΚΑΤΑΧΩΡΗΤΩΝ

Υπάρχουν n τμήματα, A_1, \dots, A_n , ένα για κάθε διεργασία. Στο τμήμα A_i μπορεί να εκτελεί UPDATES μόνο η p_i .

Κάθε καταχωρητής R_i μπορεί να εγγραφεί μόνο από την p_i αλλά να διαβαστεί από όλες τις διεργασίες.

Το τμήμα A_i έχει συσχετισθεί με τον καταχωρητή R_i . UPDATES στο A_i γράφουν μόνο στον R_i .

Κάθε καταχωρητής R_i αποτελείται από τα πεδία:

◇ val_i : τιμή του τμήματος με το οποίο έχει συσχετισθεί

◇ tag_i : χρονοσφραγίδα την οποία χρησιμοποιεί η p_i για να ξεχωρίζει τις UPDATES της

◇ $view_i$: διάνυσμα n τιμών, μια για κάθε τμήμα.

ΑΠΛΗ ΥΛ/ΣΗ SW Snapshot ΜΕ ΧΡΗΣΗ n SW ΚΑΤΑΧΩΡΗΤΩΝ

UPDATE στο τμήμα A_i με τιμή v :

view := SCAN;

increment p 's tag;

write($R_i, \langle v, \text{tag}, \text{view} \rangle$);

SCAN από τη διεργασία p :

repeatedly read R_1, \dots, R_m until see

1. either the same vector twice (then, return the vector), or
2. two new UPDATES by some process (return the view written 2nd time);

☞ Χρονική Πολυπλοκότητα = $O(mn)$ για τη SCAN και την UPDATE.

☞ Απαιτούνται m μη-πεπερασμένης χωρητικότητας MW καταχωρητές.

ΑΠΛΗ ΥΛ/ΣΗ SW Snapshot ΜΕ ΧΡΗΣΗ η SW ΚΑΤΑΧΩΡΗΤΩΝ

Θεώρημα: Ο παραπάνω αλγόριθμος είναι μια σωστή υλοποίηση ενός ατομικού στιγμιότυπου μνήμης και επιτυγχάνει τερματισμό ελεύθερο-αναμονής.

Ορισμός: Μια SCAN επιστρέφει ένα **συνεπές διάνυσμα** τιμών αν, για κάθε τμήμα του στιγμιότυπου, η τιμή που επιστρέφει η SCAN για το τμήμα είναι η τιμή που χρησιμοποίησε (ως παράμετρο) η τελευταία UPDATE στο τμήμα που σειριοποιείται πριν τη SCAN.

Για να αποδείξουμε το θεώρημα θα πρέπει:

Να αποδώσουμε σημεία σειριοποίησης.

Να δείξουμε ότι για κάθε λειτουργία (SCAN ή UPDATE), το σημείο σειριοποίησης της βρίσκεται εντός του διαστήματος εκτέλεσής της.

Να δείξουμε ότι οι SCANS επιστρέφουν συνεπή διανύσματα.

ΑΠΛΗ ΥΛ/ΣΗ SW Snapshot ΜΕ ΧΡΗΣΗ n SW ΚΑΤΑΧΩΡΗΤΩΝ

Απόδοση Σημείων Σειριοποίησης

Κάθε UPDATE γραμμικοποιείται στο σημείο που εκτελεί τη write εντολή της.

Για ευκολία αποδίδουμε σημεία σειριοποίησης όχι μόνο στις SCANS αλλά και στις embedded SCANS (δηλαδή σε αυτές που καλούνται από τις UPDATES). Στη συνέχεια θα τις ονομάζουμε όλες SCAN.

Χωρίζουμε τις SCAN σε δύο κατηγορίες:

Εύκολες: Αυτές που τερματίζουν λόγω της συνθήκης 1 στον ψευδο-κώδικα.

Δύσκολες: Αυτές που τερματίζουν λόγω της συνθήκης 2.

Γραμμικοποιούμε κάθε εύκολη SCAN ανάμεσα στις δύο ομάδες αναγνώσεων (collect) που επιστρέφουν το ίδιο διάλυσμα.

Γραμμικοποιούμε τις δύσκολες SCAN με επαγωγή ως προς τα σημεία απόκρισης τους.

ΑΠΛΗ ΥΛ/ΣΗ SW Snapshot ΜΕ ΧΡΗΣΗ n SW ΚΑΤΑΧΩΡΗΤΩΝ

Βάση Επαγωγής: Απόδοση σημείο σειριοποίησης στη δύσκολη SCAN (έστω S) της οποίας το σημείο απόκρισης είναι 1° .

Η S επιστρέφει ένα διάνυσμα τιμών το οποίο έχει εγγραφεί από μια UPDATE U (δηλαδή το διάνυσμα αυτό υπολογίστηκε από την embedded SCAN S' της UPDATE):

Η S' είναι εύκολη SCAN και έτσι της έχει ήδη αποδοθεί σημείο σειριοποίησης.

Η U (και άρα και η S') εμπεριέχονται στην S .

Το σημείο σειριοποίησης της S τοποθετείται στο ίδιο σημείο με εκείνο της S' .

Επαγωγική Υπόθεση: Έστω S η δύσκολη SCAN της οποίας το σημείο απόκρισης είναι το k -οστό. Ας υποθέσουμε ότι έχουμε αποδώσει σημεία σειριοποίησης σε όλες τις δύσκολες SCAN των οποίων το σημείο απόκρισης προηγείται εκείνου της S .

ΑΠΛΗ ΥΛ/ΣΗ SW Snapshot ΜΕ ΧΡΗΣΗ n SW ΚΑΤΑΧΩΡΗΤΩΝ

Επαγωγικό Βήμα: Αποδίδουμε σημείο σειριοποίησης στην S .

Η S επιστρέφει ένα διάνυσμα τιμών το οποίο έχει εγγραφεί από μια UPDATE U (δηλαδή το διάνυσμα αυτό υπολογίστηκε από την embedded SCAN S' της UPDATE):

Η U (και άρα και η S') εμπεριέχονται στην S .

Αν η S' είναι εύκολη SCAN, της έχει ήδη αποδοθεί σημείο σειριοποίησης.

Αν η S' είναι δύσκολη SCAN, έχει τερματίσει πριν την $S \Rightarrow$ της έχει αποδοθεί σημείο σειριοποίησης (από επαγωγική υπόθεση).

Το σημείο σειριοποίησης της S τοποθετείται στο ίδιο σημείο με εκείνο της S' .

Λήμμα: Τα σημεία σειριοποίησης κάθε SCAN ή UPDATE βρίσκεται μέσα στο διάστημα εκτέλεσης της.

Απόδειξη: Για τις UPDATES και τις εύκολες SCAN αυτό είναι προφανές.

Για τις δύσκολες SCANS αποδεικνύεται επαγωγικά ως προς τα σημεία απόκρισής τους.

ΑΠΛΗ ΥΛ/ΣΗ SW Snapshot ΜΕ ΧΡΗΣΗ n SW ΚΑΤΑΧΩΡΗΤΩΝ

Στη βάση της επαγωγής (θυμηθείτε ότι) η SCAN S της οποίας το σημείο απόκρισης είναι 1^ο σειριοποιείται στο ίδιο σημείο με την εύκολη embedded SCAN S' της οποίας το διάνυσμα επιστρέφει. Επίσης, η S' εμπεριέχεται στην S.

Το σημείο σειριοποίησης της S' βρίσκεται στο διάστημα εκτέλεσής της \Rightarrow το σημείο σειριοποίησης της S εμπεριέχεται στο διάστημα εκτέλεσής της.

Επαγωγικό Βήμα: Αποδεικνύουμε τον ισχυρισμό για τη δύσκολη SCAN S της οποίας το σημείο σειριοποίησης είναι το k-οστό.

Η S επιστρέφει ένα διάνυσμα τιμών το οποίο έχει εγγραφεί από μια UPDATE U (δηλαδή το διάνυσμα αυτό υπολογίστηκε από την embedded SCAN S' της UPDATE):

Η U (και άρα και η S') εμπεριέχονται στην S.

Αν η S' είναι εύκολη το σημείο σειριοποίησης είναι μέσα στο διάστημα εκτέλεσής της.

Το ίδιο ισχύει αν η S' είναι δύσκολη (επαγωγική υπόθεση: η S' τερματίζει πριν την S).

Το σημείο σειριοποίησης της S τοποθετείται στο ίδιο σημείο με εκείνο της S'.

\Rightarrow το σημείο σειριοποίησης της S βρίσκεται μέσα στο διάστημα εκτέλεσής της.

ΑΠΛΗ ΥΛ/ΣΗ SW Snapshot ΜΕ ΧΡΗΣΗ n SW ΚΑΤΑΧΩΡΗΤΩΝ

Λήμμα: Σε κάθε εκτέλεση του αλγορίθμου, οι SCAN λειτουργίες επιστρέφουν συνεπή διανύσματα τιμών.

Απόδειξη: Ο ισχυρισμός ισχύει προφανώς για τις εύκολες SCAN.

Για δύσκολες SCAN θα αποδειχθεί με επαγωγή ως προς τα σημεία απόκρισης.

Στη βάση της επαγωγής (θυμηθείτε ότι) η SCAN S , της οποίας το σημείο απόκρισης είναι 1° , σειριοποιείται στο ίδιο σημείο με την εύκολη embedded SCAN S' της οποίας το διάνυσμα επιστρέφει. Αφού η S' επιστρέφει συνεπές διάνυσμα, το ίδιο και η S .

Επαγωγικό Βήμα: Η Αποδεικνύουμε τον ισχυρισμό για τη δύσκολη SCAN S της οποίας το σημείο σειριοποίησης είναι το k -οστό.

Η S επιστρέφει το ίδιο διάνυσμα τιμών που επιστρέφει μια embedded SCAN S' και σειριοποιείται στο ίδιο σημείο.

Αν η S' είναι εύκολη, επιστρέφει προφανώς συνεπές διάνυσμα. Αν η S' είναι δύσκολη επιστρέφει συνεπές διάνυσμα από επαγωγική υπόθεση (η S' τερματίζει πριν την S).

Άρα και η S επιστρέφει συνεπές διάνυσμα.