Ομοφωνία σε σύστημα με αποτυχίες διεργασιών

ΕΛΛΑΔΑ
ΚΟΙΝΟΤΙΚΟ ΠΛΑΙΣΙΟ ΣΤΗΡΙΞΗΣ
2008
Ανάπτυξη παντού. Ανάπτυξη για όλους.

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

Το Πρόβλημα Ομοφωνίας – Σύγχρονα Συστήματα Μεταβίβασης Μηνύματος

Μοντέλο Κατάρρευσης (crash model)

Οι διεργασίες μπορούν να σταματούν να εκτελούνται σε αυθαίρετα σημεία της εκτέλεσης. Μια διεργασία που έχει καταρρεύσει λέγεται εσφαλμένη (fault).

Επιτρεπτές Εκτελέσεις

- Όλες οι μη-εσφαλμένες διεργασίες εκτελούν απεριόριστο αριθμό βημάτων.
- Αν μια διεργασία αποτύχει σε κάποιο βήμα, δεν εκτελεί ξανά βήματα.
- Στο τελευταίο βήμα μιας εσφαλμένης διεργασίας, αποστέλλεται μόνο ένα αυθαίρετο υποσύνολο των εξερχόμενων μηνυμάτων της διεργασίας.

Εφαρμογές

- Συστήματα ελέγχου πτήσεων.
- Οι διεργασίες ενός κατανεμημένου συστήματος χρειάζεται συχνά να συμφωνούν στο αν κάποιο μήνυμα έχει παραληφθεί.
- Διάγνωση αποτυχίας διεργασιών.

Το Πρόβλημα Ομοφωνίας – Σύγχρονα Συστήματα Μεταβίβασης Μηνύματος

Υποθέσεις

Ο αριθμός των αποτυχιών είναι το πολύ f, όπου f είναι κάποιος θετικός ακέραιος. Το f ονομάζεται βαθμός ανθεκτικότητας (resiliency) του συστήματος.

Ο γράφος είναι κλίκα με n κόμβους.

Οι σύνδεσμοι (κανάλια επικοινωνίας) είναι αξιόπιστοι.

Περιγραφή Προβλήματος

Κάθε διεργασία p_i έχει μια μεταβλητή εισόδου x_i και μια μεταβλητή εξόδου y_i.

Αρχικά, η x_i έχει μια τιμή από ένα σύνολο πιθανών εισόδων, ενώ η y_i δεν έχει αρχικοποιηθεί.

Η y_i θα πρέπει να εγγραφεί μία μόνο φορά (κάθε εγγραφή στην y_i είναι επομένως μη-αντιστρέψιμη).

Το Πρόβλημα Ομοφωνίας – Σύγχρονα Συστήματα Μεταβίβασης Μηνύματος

Ένας αλγόριθμος που επιλύει το πρόβλημα ομοφωνίας πρέπει να εγγυάται τα εξής:

Ομοφωνία (agreement)

Όλες οι μη-εσφαλμένες διεργασίες έχουν ως έξοδο την ίδια τιμή.

Εγκυρότητα (validity)

Αν όλες οι διεργασίες έχουν την ίδια τιμή εισόδου, η τιμή εξόδου πρέπει να είναι η τιμή αυτή.

Τερματισμός (termination)

Όλες οι μη-εσφαλμένες διεργασίες πρέπει να τερματίσουν.

Πρόβλημα Ομοφωνίας – Ένας απλός αλγόριθμος

code for processor p_i, $0 \le i \le n-1$.

Initially $V = \{x\}$ // V contains p_i's input

round k, $1 \le k \le f+1$:
1: send $\{v \in V : p_i$ has not already sent $v\}$ to all processors
2: receive S_j from p_j, $0 \le j \le n-1$, $j \ne i$
3: $V := V \cup \bigcup_{j=0}^{n-1} S_j$
4: if $k = f+1$ then $y := \min(V)$ // decide

Κάθε διεργασία p διατηρεί ένα σύνολο με τιμές που έχει δει μέχρι την τρέχουσα χρονική στιγμή, το οποίο αρχικά περιέχει μόνο την τιμή εισόδου της διεργασίας.

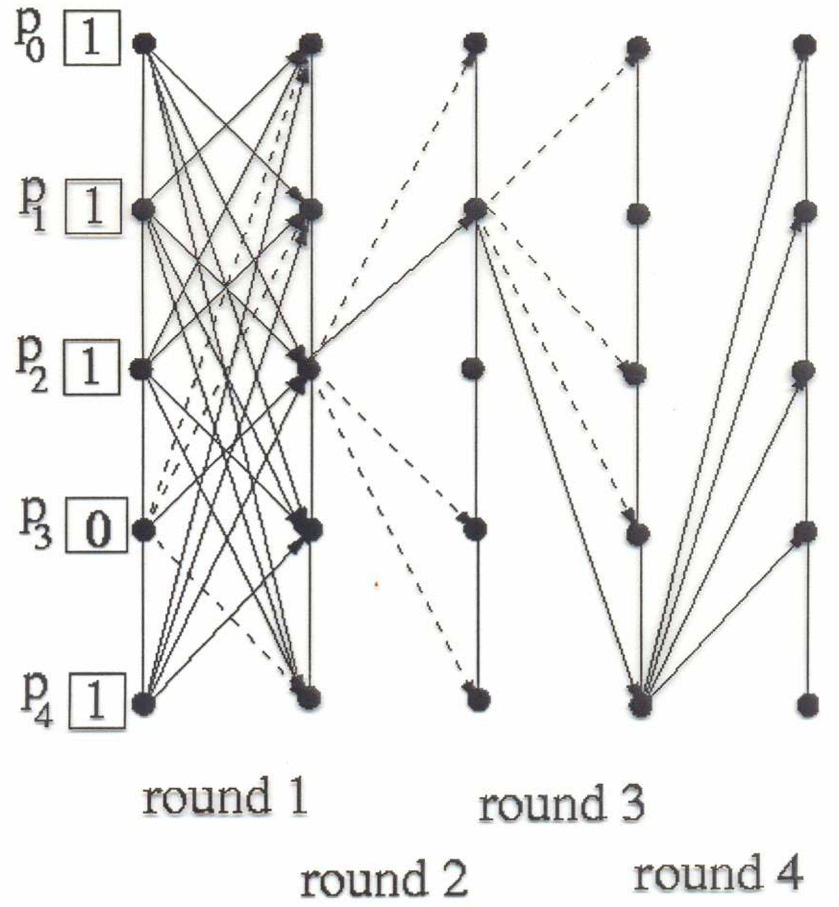
Για f+1 γύρους κάθε διεργασία:

- ενημερώνει το σύνολο της εκτελώντας την πράξη της ένωσης με τα σύνολα που λαμβάνει από άλλους καταχωρητές.
- εκπέμπει (broadcasts) οποιεσδήποτε προσθήκες έγιναν στο σύνολο σε όλες τις υπόλοιπες διεργασίες.

Μετά τους f+1 γύρους, η διεργασία αποφασίζει ως έξοδό της τη μικρότερη τιμή που περιέχεται στο σύνολό της.

Πρόβλημα Ομοφωνίας – Ένας απλός αλγόριθμος

Γιατί (f+1) γύροι;



f = 3, n = 4

Πρόβλημα Ομοφωνίας – Ένας απλός αλγόριθμος

Τερματισμός;

Εγκυρότητα;

Ομοφωνία:

- Ας υποθέσουμε ότι μια διεργασία p_i αποφασίζει μια μικρότερη τιμή, x, από κάποια άλλη p_j.
- Τότε, η x έχει παραμείνει «κρυφή» στην p_j για (f+1) γύρους.
- Έχουμε το πολύ f μη-εσφαλμένες διεργασίες. Άτοπο!!!

Αριθμός διεργασιών; $n > f$

Χρονική Πολυπλοκότητα; (f+1) γύρους

Πολυπλοκότητα Επικοινωνίας;

$n^2 * |V|$ μηνύματα, όπου V είναι το σύνολο τιμών εισόδου.

Κάτω Φράγμα στον Αριθμό των Γύρων – Η περίπτωση f = 1

Θεώρημα

Δεν υπάρχει αλγόριθμος που να επιλύει το πρόβλημα ομοφωνίας σε λιγότερο από δύο γύρους σε ένα σύστημα με $n \geq 3$ διεργασίες στο οποίο μια διεργασία μπορεί να αποτύχει.

Απόδειξη: Με εις άτοπο απαγωγή. Έστω ότι ο A είναι ένας τέτοιος αλγόριθμος.

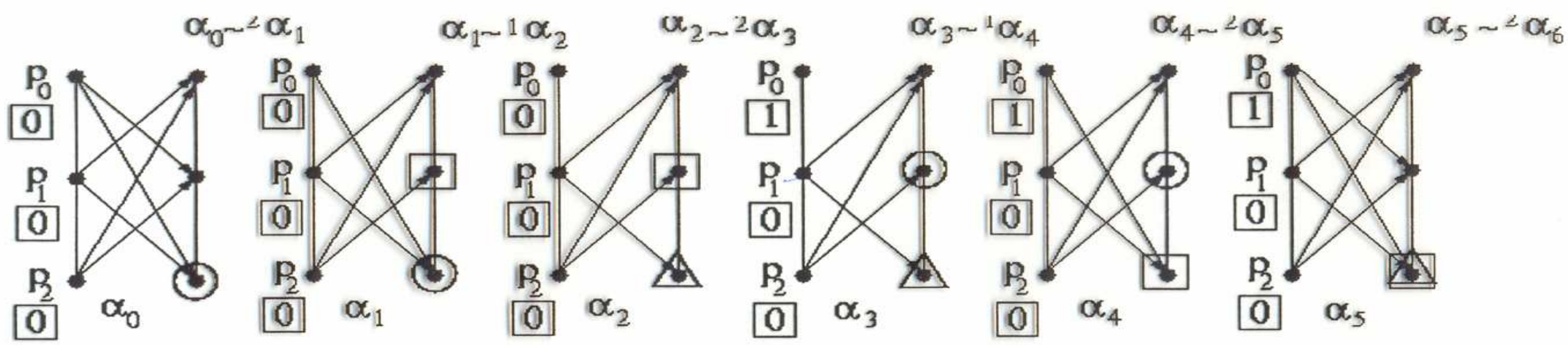
Κατασκευάζουμε μια αλυσίδα από εκτελέσεις:

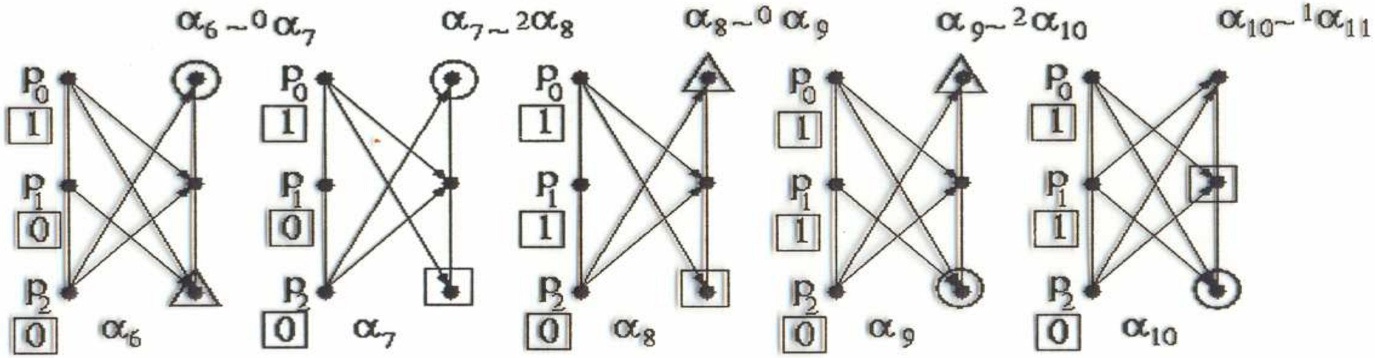
- σε κάθε εκτέλεση υπάρχει το πολύ μια εσφαλμένη διεργασία
- στην πρώτη όλες οι διεργασίες έχουν τιμή εισόδου 0
- στην τελευταία όλες οι διεργασίες έχουν τιμή εισόδου 1
- κάθε δυο διαδοχικές εκτελέσεις της αλυσίδας είναι παρόμοιες για κάποια μη-εσφαλμένη διεργασία

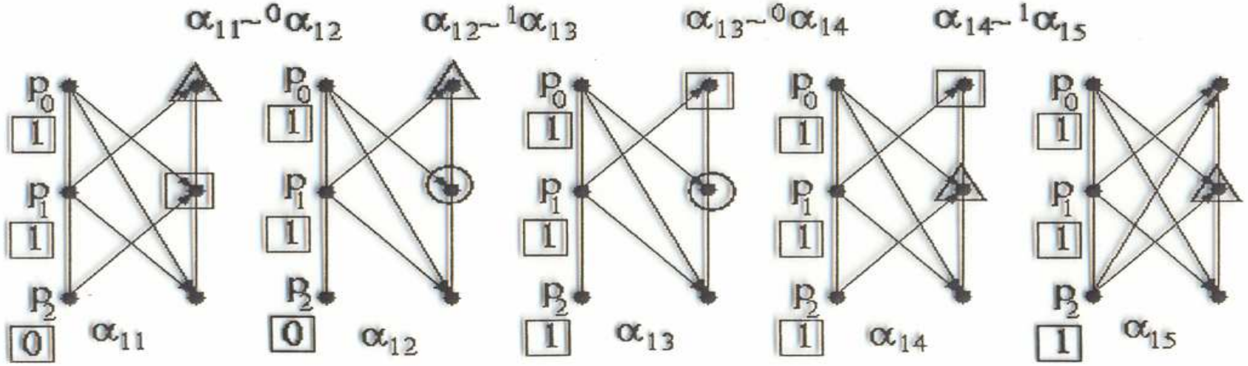
$\Rightarrow$ Σε όλες τις εκτελέσεις θα πρέπει οι διεργασίες να αποφασίζουν την ίδια τιμή. Άτοπο!!!!

Παράδειγμα

n = 3







Κάτω Φράγμα στον Αριθμό των Γύρων – Η περίπτωση f = 2

Θεώρημα

Δεν υπάρχει αλγόριθμος που να επιλύει το πρόβλημα ομοφωνίας σε λιγότερο από τρεις γύρους σε ένα σύστημα με $n \geq 4$ διεργασίες στο οποίο δύο διεργασίες μπορούν να αποτύχουν.

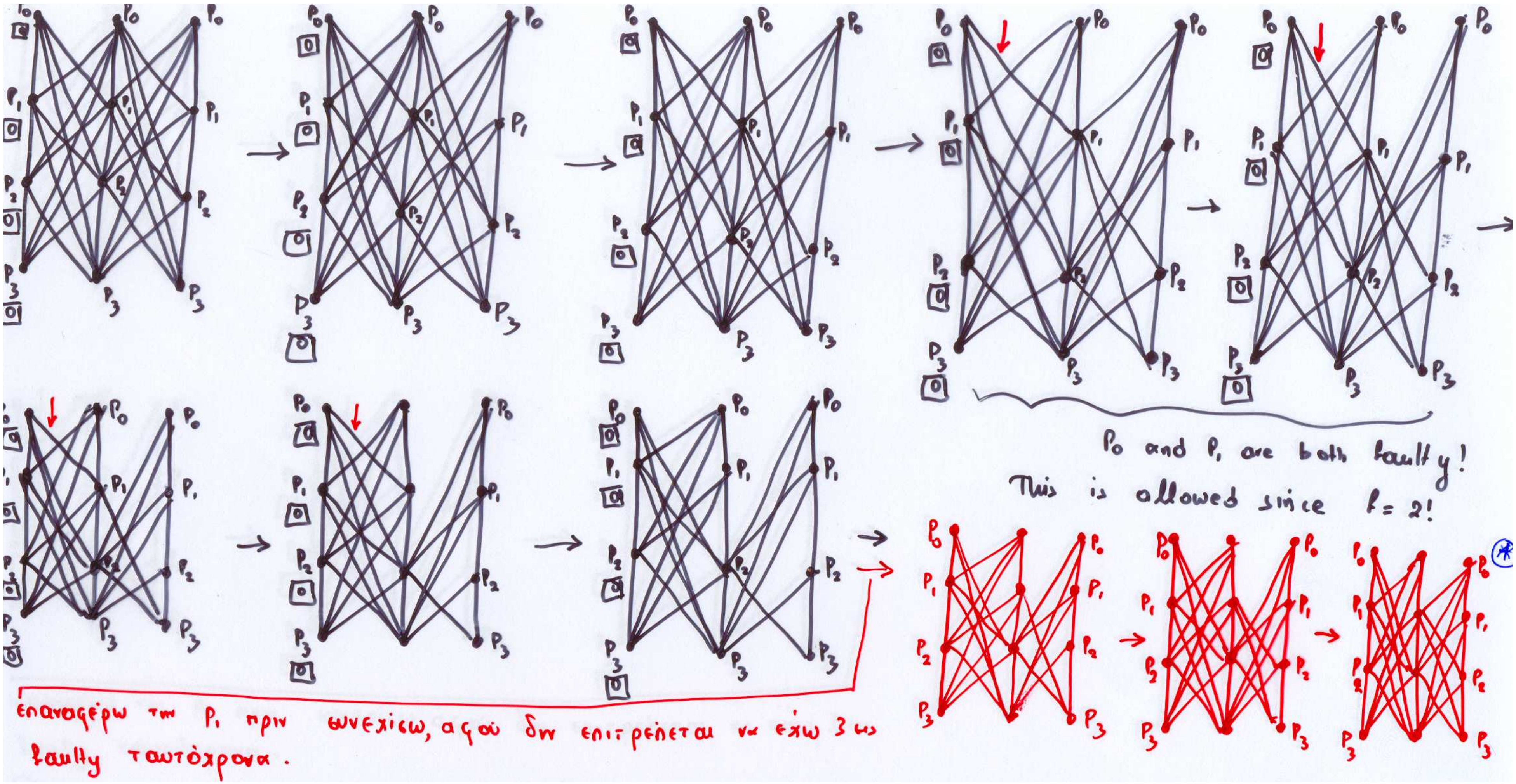
Απόδειξη: Με εις άτοπο απαγωγή. Έστω ότι ο A είναι ένας τέτοιος αλγόριθμος.

Ακολουθούμε αντίστοιχα επιχειρήματα με την περίπτωση f=1. Ωστόσο:

Τώρα έχουμε δύο γύρους!!! Τι πρόβλημα δημιουργεί αυτό;

Πως θα σκοτώσουμε ένα μήνυμα από την p_1 στην p_j;

- θα αφαιρέσουμε τα μηνύματα $2^{ου}$ γύρου της pj ένα-προς-ένα
- στη συνέχεια θα αφαιρέσουμε το μήνυμα από την p1 στην pj
- θα επαναφέρουμε τα μηνύματα $2^{ου}$ γύρου της pj ένα-προς-ένα



Στο σημείο αυτό μόνο η p0 είναι εσφαλμένη.

Κάτω Φράγμα στον Αριθμό των Γύρων – Η περίπτωση f = 2

- Επαναλαμβάνω με την p2, προκειμένου να αφαιρέσω το μήνυμα από την p0 στην p2. Στη συνέχεια, επαναλαμβάνω άλλη μια φορά με την p2.

- Στη συνέχεια:
  - Αλλάζω την τιμή εισόδου της p0 σε 1.
  - Επιτελώ την αντίστροφη από την πιο πάνω διαδικασία για να επαναφέρω τα 2 μηνύματα της p0 στο 1ο γύρο

- Επαναλαμβάνω όλα τα παραπάνω για p1, p2 και p3 (στο ρόλο της p0)!

Κάτω Φράγμα στον Αριθμό των Γύρων – Η γενική περίπτωση

Θεώρημα

Δεν υπάρχει αλγόριθμος που να επιλύει το πρόβλημα ομοφωνίας σε λιγότερο από f+1 γύρους σε ένα σύστημα με $n \geq f+2$ διεργασίες στο οποίο f διεργασίες μπορούν να αποτύχουν.

Σκιαγράφηση Απόδειξης

Οι κύριες ιδέες έχουν ήδη παρουσιαστεί!

Η αλυσίδα εκτελέσεων που δημιουργείται είναι πολύ μακρύτερη και χρησιμοποιεί f αποτυχίες διεργασιών.

Επίτευξη Ομοφωνίας σε σύστημα με Βυζαντινές Αποτυχίες Διεργασιών

Αλγόριθμοι Συλλογής Εκθετικά Μεγάλης Πληροφορίας (Exponential Information Gathering Algorithms)

Βασική Ιδέα

◊ Οι διεργασίες στέλνουν και αναμεταδίδουν αρχικές τιμές για αρκετούς γύρους, καταγράφοντας τις τιμές που λαμβάνουν κατά μήκος διαφορετικών «μονοπατιών» επικοινωνίας (communication paths) σε μια δενδρική δομή δεδομένων που ονομάζεται δένδρο EIG (Exponential Information Gathering).

◊ Τελικά, ακολουθούν έναν καθορισμένο (ίδιο για όλες) κανόνα για να αποφασίσουν ποια από τις τιμές που καταγράφονται στο δένδρο τους θα επιλέξουν ως έξοδο.

Δομή Δεδομένων

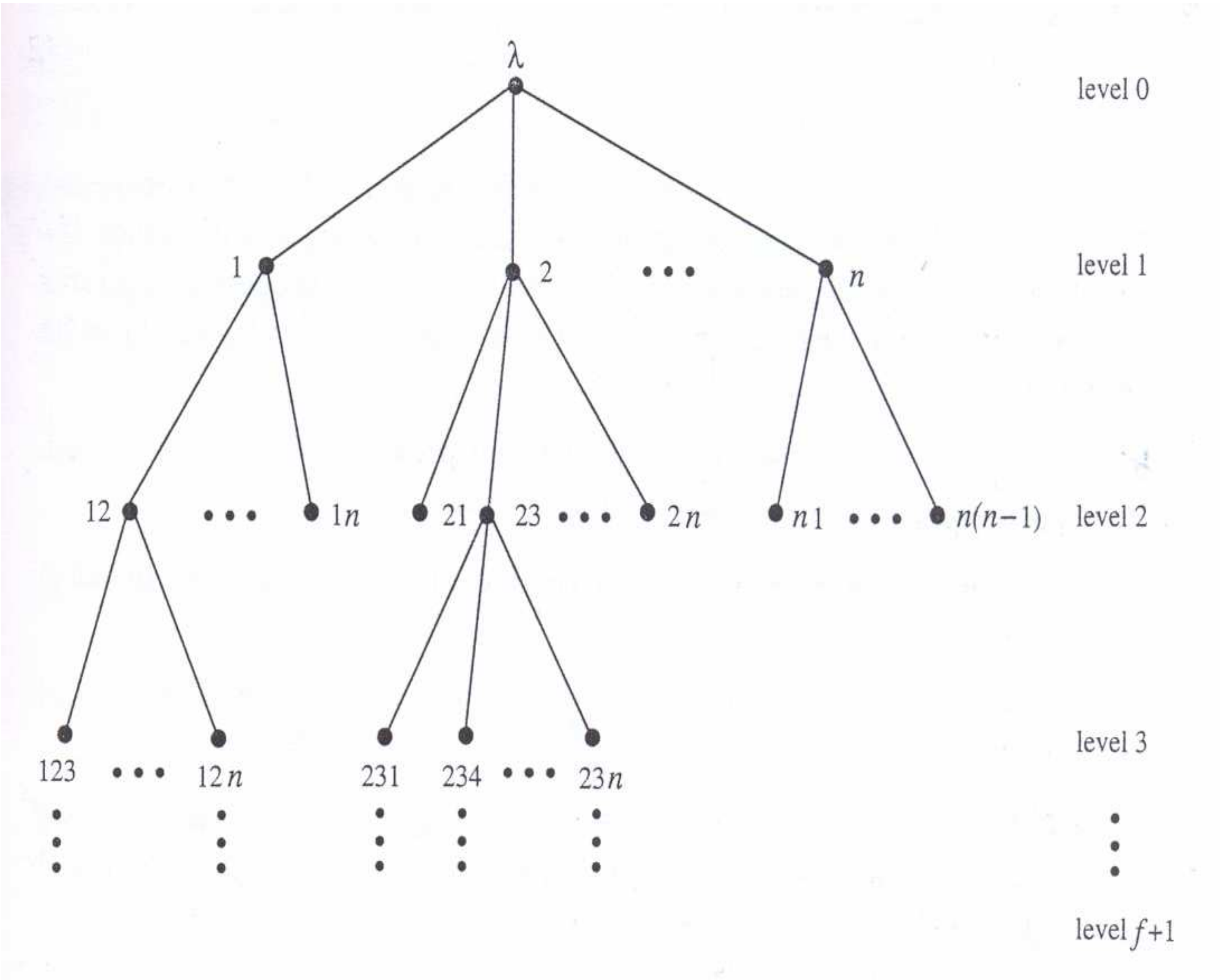
Κάθε διεργασία χειρίζεται ένα δένδρο EIG ($T = T_{n,f}$), κάθε κόμβος του οποίου έχει μια ετικέτα.

Κάθε μονοπάτι του δένδρου αναπαριστά μια αλυσίδα από διεργασίες κατά μήκος των οποίων κάποια αρχική τιμή έχει μεταδοθεί.

Το δένδρο έχει f+2 επίπεδα, 0, ..., f+1.

Κάθε κόμβος επιπέδου k έχει ακριβώς n-k παιδιά, όπου $0 \leq k \leq f$.

Αλγόριθμοι Συλλογής Εκθετικά Μεγάλης Πληροφορίας



Ονομασία Κόμβων

Χρησιμοποιούνται αλφαριθμητικά που αποτελούν ακολουθίες από δείκτες διεργασιών. Η ετικέτα της ρίζας είναι το κενό αλφαριθμητικό λ. Κάθε κόμβος με ετικέτα $i_1...i_k$ έχει ακριβώς n-k παιδιά με ετικέτες $i_1...i_kj$, όπου $j \in \{1, ..., n\} - \{i_1, ..., i_k\}$.

Αλγόριθμοι Συλλογής Εκθετικά Μεγάλης Πληροφορίας

Ο υπολογισμός διαρκεί για f+1 γύρους ακριβώς. Κατά τη διάρκεια του, κάθε διεργασία διανθίζει τους κόμβους του δένδρου της με τιμές στο V ή με το null. Όλοι οι κόμβοι σε επίπεδο k, έχουν διανθιστεί μέχρι και το τέλος του k-οστού γύρου.

Τρόπος Διάνθισης του δένδρου μιας διεργασίας pi

Η ρίζα του δένδρου της διεργασίας p_i διανθίζεται με την τιμή εισόδου της.

Σε κάθε γύρο, αν ο κόμβος με ετικέτα $i_1 \ldots i_k$, $1 \leq k \leq f+1$, διανθιστεί με κάποια τιμή $v \in V$, $\Rightarrow$ η i_k είπε στην i στο γύρο k ότι η i_{k-1} είπε στην i_k στο γύρο k-1 ότι ... ότι η i_1 είπε στην i_2 στον γύρο 1 ότι η αρχική τιμή της i_1 είναι v.

Αν ο κόμβος με ετικέτα $i_1 \ldots i_k$, διανθιστεί με null $\Rightarrow$ η αλυσίδα επικοινωνίας $i_1, \ldots, i_k$, i έχει «σπάσει» λόγω κάποιας αποτυχίας.

Υπόθεση

Κάθε διεργασία μπορεί να στέλνει μηνύματα στον εαυτό της.

Περιγραφή του Αλγορίθμου EIGStop – διεργασία p_i

Για κάθε κόμβο με ετικέτα x του δένδρου της, η p_i διατηρεί μια μεταβλητή val(x) όπου αποθηκεύεται η τιμή με την οποία διανθίζεται ο κόμβος. Αρχικά, val(λ) = αρχική τιμή p_i.

Γύρος 1: Η p_i εκπέμπει την val(λ) σε όλες τις διεργασίες (καθώς και στον εαυτό της).

Στη συνέχεια, η p_i καταγράφει τις εισερχόμενες πληροφορίες:

1. Αν ένα μήνυμα με τιμή v φθάσει στην p_i από την $p_j \Rightarrow$ val(j) = v.
2. Αν η p_i δεν λάβει μήνυμα από την $p_j \Rightarrow$ val(j) = null.

Γύρος k, $2 \leq k \leq f+1$: Η p_i εκπέμπει τα ζευγάρια (x,val(x)), για κάθε ετικέτα x ενός κόμβου στο επίπεδο k-1 του T, τ.ω. η ακολουθία που αναπαριστά η x δεν περιέχει τον δείκτη i.

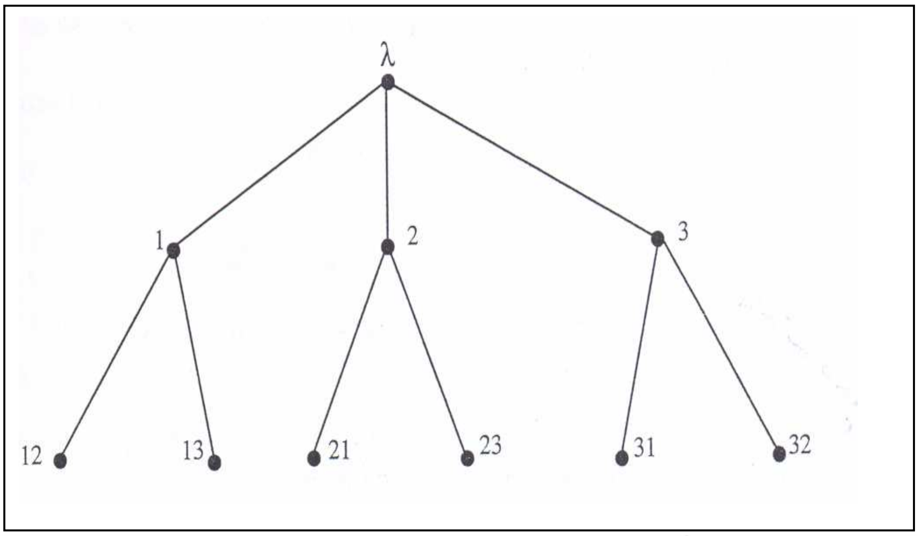
Στη συνέχεια, η p_i καταγράφει τις εισερχόμενες πληροφορίες:

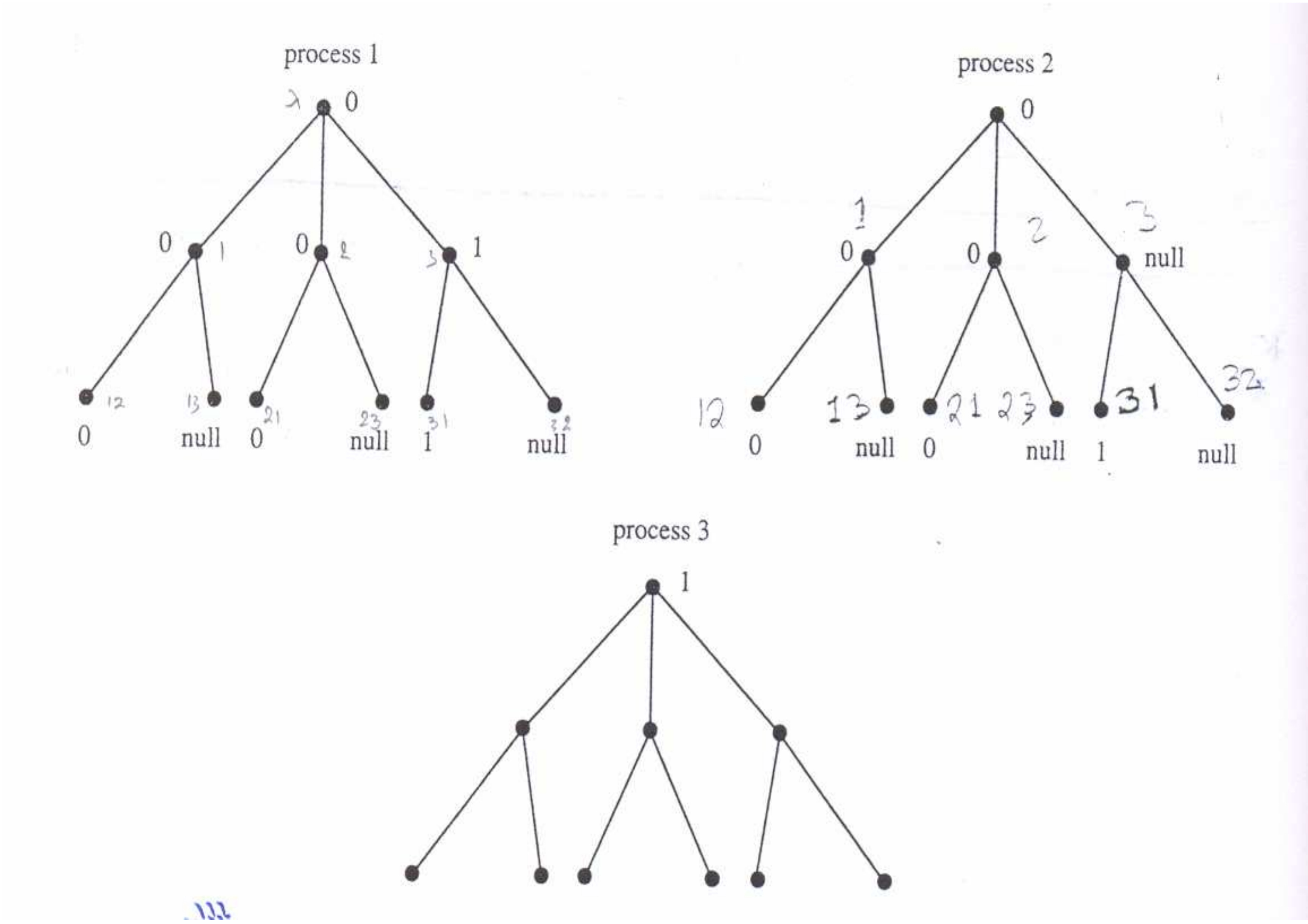
1. Αν xj είναι ετικέτα ενός κόμβου επιπέδου k στο T, όπου x είναι ένα αλφαριθμητικό που αναπαριστά μια ακολουθία από δείκτες διεργασιών και j είναι ένας ακόμη δείκτης διεργασίας, και η p_i λάβει μήνυμα με τιμή val(x) = v από την p_j, τότε η p_i θέτει την val(xj) = v.
2. Αν η p_i δεν λάβει τέτοιο μήνυμα, τότε val(xj) = null.

Μετά από f+1 γύρους έστω W_i το σύνολο των τιμών που έχουν διανθίσει το δένδρο της p_i. Η p_i αποφασίζει την έξοδό της να είναι π.χ., το μικρότερο στοιχείο του W_i.

Περιγραφή του Αλγορίθμου EIGStop – διεργασία p_i

Παράδειγμα





n=3, f = 1, 2 γύροι, 3 επίπεδα στο δένδρο

αρχικές τιμές των p_1, p_2, p_3, 0, 0, και 1, αντίστοιχα.

Αλγόριθμος EIGStop – Ορθότητα

Λήμμα 1: Μετά από f+1 γύρους εκτέλεσης του EIGStop ισχύουν τα ακόλουθα:

1. $val(\lambda)_i$ = αρχική τιμή της p_i
2. Αν xj είναι ετικέτα ενός κόμβου και $val(xj)_i = v$, τότε $val(x)_j = v$.
3. Αν xj είναι η ετικέτα ενός κόμβου και $val(xj)_i = null$, τότε είτε $val(x)_j = null$ ή η p_j αποτυγχάνει να στείλει μήνυμα στην p_i στο γύρο $|x|+1$.

Λήμμα 2: Μετά από f+1 γύρους εκτέλεσης του EIGStop ισχύουν τα ακόλουθα:

1. Αν y είναι η ετικέτα ενός κόμβου, $val(y)_i = v$ και xj είναι πρόθεμα της y, τότε $val(x)_j = v$.
2. Αν η τιμή v εμφανίζεται στο W οποιασδήποτε διεργασίας, τότε $v = val(\lambda)_i$, για κάποιο δείκτη i.
3. Αν η τιμή v εμφανίζεται στο W_i, τότε υπάρχει κάποια ετικέτα y που δεν περιέχει το i τ.ω. $v = val(y)_i$.

Απόδειξη: Το 1 συνεπάγεται από Λήμμα 1 (μέρος 2, επαναληπτική εφαρμογή). Το 2 συνεπάγεται από 1. Για το 3, έστω ότι η v εμφανίζεται ως η val μόνο σε κόμβους με ετικέτες που περιέχουν το i. Έστω y η μικρότερη ετικέτα με $v = val(y)_i$. Τότε η y έχει πρόθεμα της μορφής xi. Από 1 $\Rightarrow$ $val(x)_i = v$, που αντιτίθεται στην επιλογή του y.

Αλγόριθμος EIGStop – Ορθότητα

Λήμμα 3: Αν οι p_i, p_j είναι μη-εσφαλμένες διεργασίες, τότε $W_i = W_j$.

1. **Απόδειξη:** Έστω ότι $i \neq j$. Αποδεικνύουμε ότι $W_i \subseteq W_j$ *και* $W_j \subseteq W_i$.
2. $W_i \subseteq W_j$

   Έστω $v \in W_i$. Τότε το Λήμμα 2 συνεπάγεται ότι $v = val(x)i$, *για κάποιο* x που δεν περιέχει το i.

   i. $|x| < f+1 \Rightarrow |xi| \leq f+1$. Αφού η x δεν περιέχει το i, η (μη-εσφαλμένη) p_i θα στείλει την τιμή v στην p_j στο γύρο $|xi| \Rightarrow val(xi)_j = v \Rightarrow v \in W_j$.

   ii. $|x| = f+1$. Αφού υπάρχουν το πολύ f εσφαλμένες διεργασίες *και* όλοι οι δείκτες στην x είναι μοναδικοί, υπάρχει *κάποια* μη-εσφαλμένη διεργασία p_l τ.ω. το l εμφανίζεται στην x

   $\Rightarrow$ η x έχει πρόθεμα της μορφής yl. Το Λήμμα 2 συνεπάγεται ότι $val(y)_l = v$. Αφού η l είναι μη εσφαλμένη, μεταδίδει την v στην p_j στο γύρο $|yl|$. $\Rightarrow val(yl)_j = v \Rightarrow v \in W_j$.
3. $W_j \subseteq W_i$. Η απόδειξη είναι συμμετρική.

Αλγόριθμος EIGStop – Ορθότητα

Θεώρημα

Ο αλγόριθμος EIGStop επιλύει το πρόβλημα ομοφωνίας σε σύστημα στο οποίο διεργασίες μπορούν να καταρρέουν.

Απόδειξη

Ο τερματισμός είναι προφανής. Επίσης, αν όλες οι αρχικές τιμές είναι v τότε κάθε W_i περιέχει ακριβώς την τιμή v. Έτσι όλες έχουν έξοδο την v.

Έστω ότι p_i,p_j είναι δύο διεργασίες που τερματίζουν $\Rightarrow$ οι p_i, p_j είναι μη-εσφαλμένες. Το Λήμμα 3 συνεπάγεται ότι $W_i = W_j$. Άρα οι p_i, p_j αποφασίζουν την ίδια τιμή εξόδου.

Πολυπλοκότητες

Χρονική πολυπλοκότητα;

Πολυπλοκότητα επικοινωνίας;

Αλγόριθμοι για Βυζαντινές Αποτυχίες

Ο Αλγόριθμος EIGByz – Κώδικας για διεργασία p_i

Οι διεργασίες (υποθέτουμε n > 3f) μεταδίδουν τιμές για f+1 γύρους με τον ίδιο τρόπο όπως στον αλγόριθμο EIGStop με τις εξής διαφορές:

- ◊ Αν η p_i λάβει μήνυμα από την p_j που δεν έχει τη σωστή μορφή, το αγνοεί.
- ◊ Στο τέλος του γύρου f+1, η p_i, ξεκινώντας από τα φύλλα, διανθίζει όλους τους κόμβους του δένδρου της με μια ακόμη μεταβλητή newval, την οποία ενημερώνει ως εξής:
  - o Για κάθε κόμβο-**φύλλο** με ετικέτα x, newval(x) = val(x).
  - o Για κάθε **εσωτερικό** κόμβο με ετικέτα x, η newval(x) παίρνει την τιμή που έχει η μεταβλητή newval σε μια αυστηρή πλειοψηφία των παιδιών της x (δηλαδή παίνρει μια τιμή v, τ.ω., newval(xj) = v για την πλειοψηφία των κόμβων με ετικέτες της μορφής xj, αν φυσικά υπάρχει τέτοια πλειοψηφία). Αν τέτοια πλειοψηφία δεν υπάρχει, newval(x) = null.
- ◊ Η τιμή εξόδου της p_i είναι η newval(λ).

Ο Αλγόριθμος EIGByz –Ορθότητα

Λήμμα 1: Μετά από f+1 γύρους εκτέλεσης του αλγορίθμου EIGByz ισχύει το εξής. Αν p_i, p_j και p_k είναι μη-εσφαλμένες διεργασίες, με $i \neq j$, τότε $val(x)_i = val(x)_j = val(y)_k$ για κάθε ετικέτα $x = yk$.

Απόδειξη: Αφού η k είναι μη-εσφαλμένη στέλνει το ίδιο μήνυμα με την τιμή $val(y)_k$ και στην p_i και στην p_j στο γύρο $|x|$.

Λήμμα 2: Μετά από f+1 γύρους εκτέλεσης του αλγορίθμου EIGByz ισχύει το εξής. Έστω ότι $x = yk$ είναι μια ετικέτα τ.ω. η p_k είναι μη-εσφαλμένη διεργασία. Τότε, $newval(x)_i = val(x)_i = val(y)_k$ για όλες τις μη-εσφαλμένες διεργασίες i.

Απόδειξη: Με επαγωγή στο μήκος των ετικετών του δένδρου, ξεκινώντας από τα φύλλα.

Βάση Επαγωγής: Έστω x η ετικέτα ενός κόμβου-φύλλου ($|x| = f+1$).

Λόγω του τρόπου ανάθεσης τιμών στις newval κόμβων-φύλλων $\Rightarrow$ $newval(x)_i = val(x)_i$

Λήμμα 1 $\Rightarrow$ όλες οι μη-εσφαλμένες διεργασίες p_i έχουν την ίδια $val(x)_i = val(y)_k$.

Ο Αλγόριθμος EIGByz –Ορθότητα

Επαγωγική Υπόθεση: Έστω ότι $1 \leq r \leq f$ και έστω ότι ο ισχυρισμός ισχύει για όλες τις ετικέτες $x' = y'k'$ τ.ω. $|x'| = r+1$ (και η $p_{k'}$ είναι μη εσφαλμένη διεργασία).

Επαγωγικό Βήμα: Θα αποδειχθεί ότι ο ισχυρισμός ισχύει για όλες τις ετικέτες $x = yk$ με $|x|=r$ (για τις οποίες ισχύει ότι η p_k είναι μη-εσφαλμένη διεργασία).

Λήμμα 1 $\Rightarrow$ όλες οι μη-εσφαλμένες διεργασίες p_i έχουν $val(x)_i = val(y)_k = v \Rightarrow$

Κάθε μη-εσφαλμένη διεργασία j εκπέμπει την ίδια τιμή v για την ετικέτα x στον γύρο $r+1 \Rightarrow$ $val(xj)_i = v$ για όλες τις μη-εσφαλμένες διεργασίες p_i και p_j.

Από επαγωγική υπόθεση $\Rightarrow newval(xj)_i = val(xj)_i = v$, για όλες τις μη-εσφαλμένες p_i και p_j.

Η πλειοψηφία των ετικετών των παιδιών του κόμβου x τελειώνει σε δείκτες μη-εσφαλμένων διεργασιών:

παιδιών $x = n-r \geq n-f > 3f - f = 2f \Rightarrow$ η πλειοψηφία είναι εγγυημένη από το ότι το πολύ f από τα παιδιά έχουν ετικέτες των οποίων οι τελευταίοι δείκτες αντιστοιχούν σε εσφαλμένες διεργασίες.

$\Rightarrow$ για κάθε μη-εσφαλμένη διεργασία p_i, $newval(xj)_i = v$ για την πλειοψηφία των παιδιών xj του x.

$\Rightarrow newval(x)_i = v$.

Ο Αλγόριθμος EIGByz –Ορθότητα

Λήμμα 3: Αν όλες οι μη-εσφαλμένες διεργασίες έχουν την ίδια τιμή εισόδου v, τότε η v είναι η μοναδική τιμή εξόδου για κάθε μη-εσφαλμένη διεργασία.

Απόδειξη: Όλες οι μη-εσφαλμένες διεργασίες εκπέμπουν v στον 1^o γύρο $\Rightarrow$ $val(j)_i = v$ για όλες τις μη-εσφαλμένες διεργασίες p_i και p_j.

Λήμμα 2 $\Rightarrow$ $newval(j)_i = val(j)_i = v$.

Από τον κανόνα πλειοψηφίας: $newval(\lambda)_i = v$.

Ορισμοί

◊ Ένας κόμβος με ετικέτα x ονομάζεται **κοινός** αν για κάθε μη-εσφαλμένη διεργασία p_i, το $newval(x)_i$ έχει την ίδια τιμή.

◊ Ένα υποδένδρο λέμε ότι έχει ένα **κοινό μέτωπο** αν υπάρχει ένας κοινός κόμβος σε κάθε μονοπάτι από τη ρίζα του υποδένδρου προς τα φύλλα.

Ο Αλγόριθμος EIGByz –Ορθότητα

Λήμμα 4: Έστω ένας οποιοσδήποτε κόμβος με ετικέτα x. Αν υπάρχει ένα κοινό μέτωπο για το υποδένδρο με ρίζα τον x, τότε ο x είναι κοινός.

Απόδειξη: Με επαγωγή στο μήκος των ετικετών του δένδρου, ξεκινώντας από τα φύλλα.

Βάση Επαγωγής: Ο x είναι φύλλο. Αν υπάρχει κοινό μέτωπο για το υποδένδρο με ρίζα τον x, αυτό θα αποτελείται μόνο από τον x. Έτσι, ο x είναι κοινός όπως απαιτείται.

Επαγωγική Υπόθεση: Έστω ότι $1 \leq r \leq f$ και έστω ότι ο ισχυρισμός ισχύει για κάθε κόμβο x' τ.ω. $|x'| = r+1$.

Επαγωγικό Βήμα: Θα αποδειχθεί ότι ο ισχυρισμός ισχύει για όλες τις ετικέτες x τ.ω. $|x|=r$.

Έστω ότι υπάρχει ένα κοινό μέτωπο του υποδένδρου με ρίζα τον x. Αν ο x ανήκει σε αυτό, τότε είναι κοινός, όπως απαιτείται. Ας υποθέσουμε ότι αυτό δεν ισχύει.

Τότε, για κάθε παιδί xl του x, υπάρχει κοινό μέτωπο για το υποδένδρο με ρίζα τον xl. Από επαγωγική υπόθεση $\Rightarrow$ ο xl είναι κοινός. Άρα, όλες οι διεργασίες έχουν το ίδιο newval(xl). Αυτό ισχύει για κάθε ένα από τα παιδιά του x. Άρα, το newval(x) είναι το ίδιο για όλες τις διεργασίες και επομένως ο x είναι κοινός.

Ο Αλγόριθμος EIGByz –Ορθότητα

Λήμμα 5: Μετά από f+1 γύρους εκτέλεσης του αλγορίθμου EIGByz, ο κόμβος ρίζα λ του δένδρου κάθε μη-εσφαλμένης διεργασίας είναι κοινός.

Απόδειξη: Η ετικέτα οποιουδήποτε φύλλου αντιστοιχεί σε μια ακολουθία από f+1 διαφορετικούς δείκτες διεργασιών.

Μόνο f από αυτούς μπορούν να αντιστοιχούν σε εσφαλμένες διεργασίες.

Η ετικέτα ενός από τους κόμβους του μονοπατιού από τη ρίζα στο φύλλο τελειώνει με τον δείκτη αυτής της μη-εσφαλμένης διεργασίας. Ο κόμβος αυτός είναι κοινός.

Άρα, όλο το δένδρο έχει ένα κοινό μέτωπο.

Από Λήμμα 4, η ρίζα είναι επίσης κοινός κόμβος.

Θεώρημα: Ο αλγόριθμος EIGByz επιλύει το πρόβλημα της Βυζαντινής ομοφωνίας για n διεργασίες με f αποτυχίες αν n > 3f.