Διάλεξη 16: Πρόβλημα Συμφωνίας

ΕΠΛ 432: Κατανεμημένοι Αλγόριθμοι

Τι θα δούμε σήμερα

- Ορισμός του προβλήματος Συμφωνίας
- Αλγόριθμος Συμφωνίας με Σφάλματα Κατάρρευσης

Πρόβλημα Συμφωνίας

- Κάθε επεξεργαστής έχει μια είσοδο

- Ένας αλγόριθμος επιλύει το πρόβλημα της Συμφωνίας αν μπορεί να εγγυηθεί τις πιο κάτω συνθήκες
  - ***Συνθήκη Τερματισμού***: Κάθε μη εσφαλμένος επεξεργαστής πρέπει να αποφασίσει μια τιμή
    - Η απόφαση είναι μη αντιστρέψιμη

  - ***Συνθήκη Συμφωνίας***: Όλοι οι μη εσφαλμένοι επεξεργαστές αποφασίζουν την ίδια τιμή

  - ***Συνθήκη Εγκυρότητας***: Η κοινή απόφαση πρέπει να αποτελούσε τη είσοδο κάποιου επεξεργαστή.
    - Αν όλες οι είσοδοι είναι οι ίδιες, τότε κάθε μη εσφαλμένος επεξεργαστής πρέπει να αποφασίζει την κοινή είσοδο.

Παραδείγματα Συμφωνίας

- Δυαδικές είσοδοι:
  - Διάνυσμα εισόδου 1,1,1,1,1
    - Η απόφαση πρέπει να είναι 1
  - Διάνυσμα εισόδου 0,0,0,0,0
    - Η απόφαση πρέπει να είναι 0
  - Διάνυσμα εισόδου 1,0,0,1,0
    - Η απόφαση μπορεί να είναι είτε 0 ή 1

- Είσοδοι Πολλαπλών τιμών:
  - Διάνυσμα εισόδου 1,2,3,2,1
    - Η απόφαση μπορεί να είναι είτε 1 ή 2 ή 3

Ανασκόπηση Αποτελεσμάτων

- Σύγχρονο Μοντέλο
- Το πολύ f επεξεργαστές μπορούν να είναι εσφαλμένοι
- Στενά κάτω φράγματα για το μοντέλο ανταλλαγής μηνυμάτων

	Σφάλματα Κατάρρευσης	**Βυζαντινά Σφάλματα**
Αριθμός γύρων	$f + 1$	$f + 1$
Ολικός αριθμός επεξεργαστών	$f + 1$	$3f + 1$
Μέγεθος μηνύματων	Πολυωνυμικό	Πολυωνυμικό

Ανασκόπηση Αποτελεσμάτων

- ***Αδύνατη*** η επίλυση Συμφωνίας στο ασύγχρονο μοντέλο

- Ακόμα και στην παρουσία μόνο ενός σφάλματος κατάρρευσης

- Ισχύει για το μοντέλο ανταλλαγής μηνυμάτων και για το μοντέλο κοινόχρηστης μνήμης.

Μοντελοποίηση Σφαλμάτων Κατάρρευσης

- Αλλάζουμε τον ορισμό μιας **νόμιμης εκτέλεσης** χωρίς σφάλματα για να επιτρέπει σφάλματα κατάρρευσης

- Όλοι εκτός από το πολύ *f* επεξεργαστές (οι εσφαλμένοι) εκτελούν άπειρο αριθμό βημάτων.
  - *Σύγχρονο μοντέλο*: όταν ένας επεξεργαστής καταρρεύσει σε ένα γύρο δεν εκτελεί άλλα βήματα.

- Στο τελευταίο του βήμα ένας επεξεργαστής μπορεί να στείλει μηνύματα σε ένα τυχαίο υποσύνολο από επεξεργαστές προτού καταρρεύσει.

Αλγόριθμος Συμφωνίας με Σφάλματα Κατάρρευσης

Code for each processor:

v := my input
at each round 1 through *f+1:*
  if I have not yet sent *v* then send *v* to all
  wait to receive messages for this round
  v := minimum among all received values and
            current value of *v*
  if this is round *f+1* then decide on *v*

Εκτέλεση Αλγορίθμου

- round 1:
  - send my input
  - receive round 1 msgs
  - compute value for *v*
- round 2:
  - send *v* (if this is a new value)
  - receive round 2 msgs
  - compute value for *v*
- …
- round $f + 1$:
  - send *v* (if this is a new value)
  - receive round $f + 1$ msgs
  - compute value for *v*
  - decide *v*

Ορθότητα Αλγορίθμου

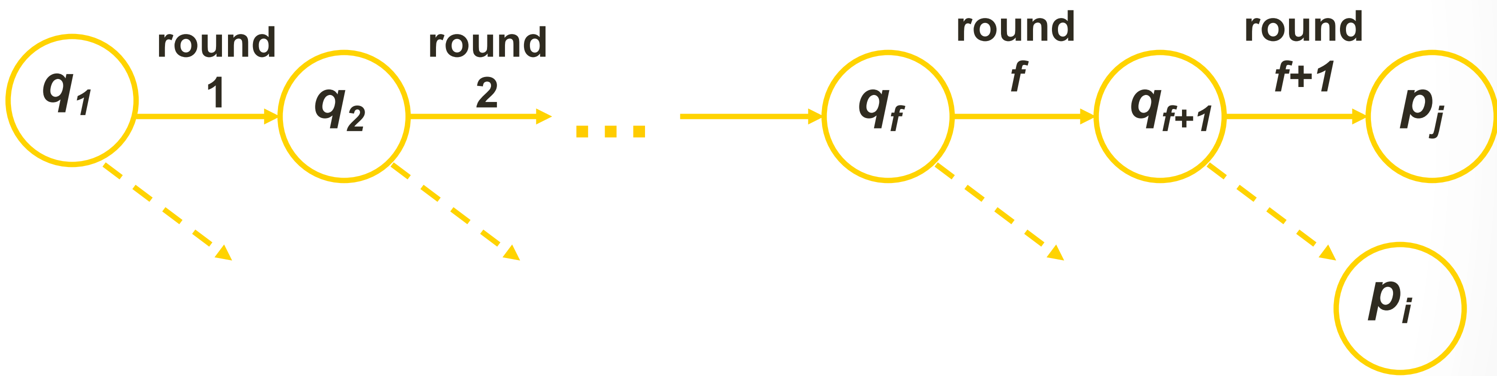
Συνθήκη Τερματισμού (Ζωτικότητα): Από τον κώδικα τελειώνουμε στον γύρο *f+1.*

Συνθήκη Εγκυρότητας (Ασφάλεια 1): Ισχύει από την στιγμή που οι επεξεργαστές δεν δημιουργούν τυχαίες τιμές: αν όλες οι είσοδοι είναι οι ίδιες τότε αυτή είναι η μόνη τιμή που ανταλλάσσεται στο σύστημα.

Ορθότητα Αλγορίθμου

Συνθήκη Συμφωνίας (Ασφάλεια 2):

- Υποθέτουμε με αντίφαση ότι ο p_j αποφασίζει μια πιο μικρή τιμή, *x*, από τον p_i
- Αυτό μπορεί να συμβεί εαν ο p_i δεν παραλάβει το *x* λόγω κάποιας αλυσίδας εσφαλμένων επεξεργαστών:



- Αφού όμως έχουμε *f* + *1* γύρους τότε πρέπει να υπάρχουν *f* + *1* εσφαλμένοι επεξεργαστές.
- Αυτό αντιφάσκει την αρχική μας υπόθεση ότι το πολύ *f* επεξεργαστές μπορεί να καταρρεύσουν.

Απόδοση Αλγορίθμου

- Αριθμός επεξεργαστών $n > f$
- $f+1$ γύρους
- Το πολύ $n^2 \bullet |V|$ μηνύματα όπου V είναι το σύνολο των εισόδων
- Κάθε μήνυμα έχει μέγεθος $log|V|$ bits

Κάτω Φράγμα στους Γύρους

Υποθέσεις:

- $n > f + 1$
- Κάθε επεξεργαστής στέλνει μήνυμα σε όλους τους άλλους επεξεργαστές σε κάθε γύρο.
- Δυαδική είσοδος {0,1}

Αραιά-Εσφαλμένες Εκτελέσεις

- Ένα κακό σενάριο για ένα αλγόριθμο ανεκτικό σε σφάλματα κατάρρευσης είναι όταν υπάρχει ένα σφάλμα σε κάθε γύρο.

- **Αραιά-Εσφαλμένες** εκτελέσεις ονομάζονται αυτές που έχουν το πολύ ένα σφάλμα σε κάθε γύρο

- Μας ενδιαφέρουν αυτές οι εκτελέσεις σε αυτή την απόδειξη

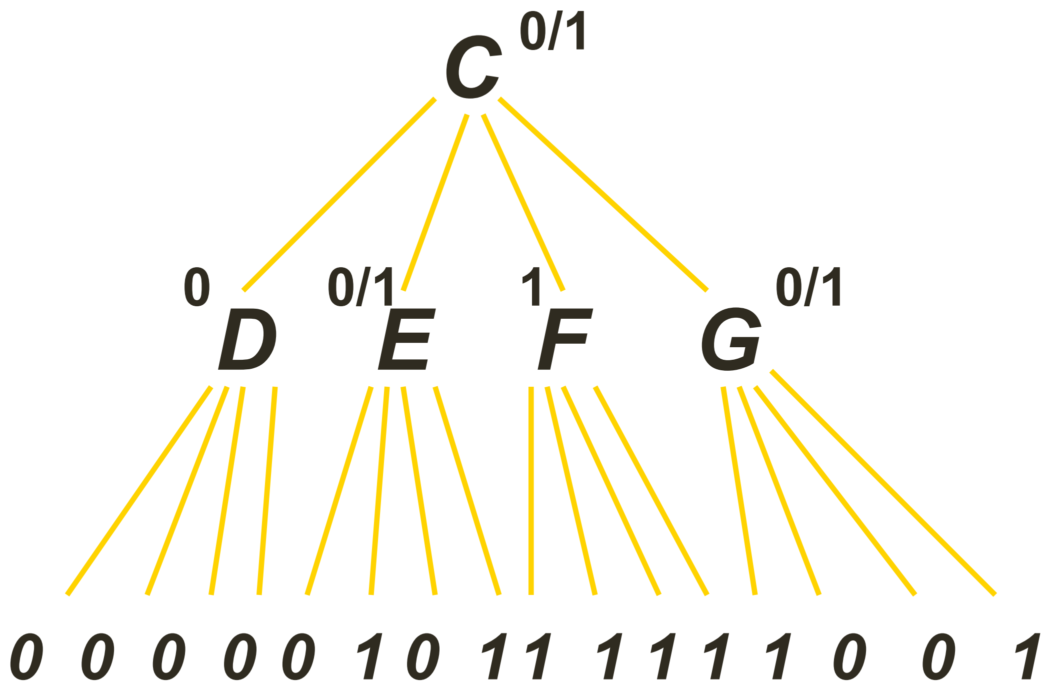
Σθένος Διατάξεων Τιμών

- Το **σθένος** μιας διάταξης τιμών C είναι το σύνολο όλων των τιμών που μπορούν να αποφασιστούν από ένα μη-εσφαλμένο επεξεργαστή σε μια διάταξη που μπορεί να προκύψει από το C, σε μια νόμιμη (αραιά-εσφαλμένη) εκτέλεση.

- **Δισθενή Σύνολο (Bivalent):** το σύνολο που περιέχει 0 και 1.

- **Μονοσθενή Σύνολο (Univalent):** το σύνολο που περιέχει μια μόνο τιμή
  - 0-σθενή (0-valent) αν περιέχει μόνο το 0, ή
  - 1-σθενή (1-valent) αν περιέχει μόνο το 1

Υπολογισμός Σθένους



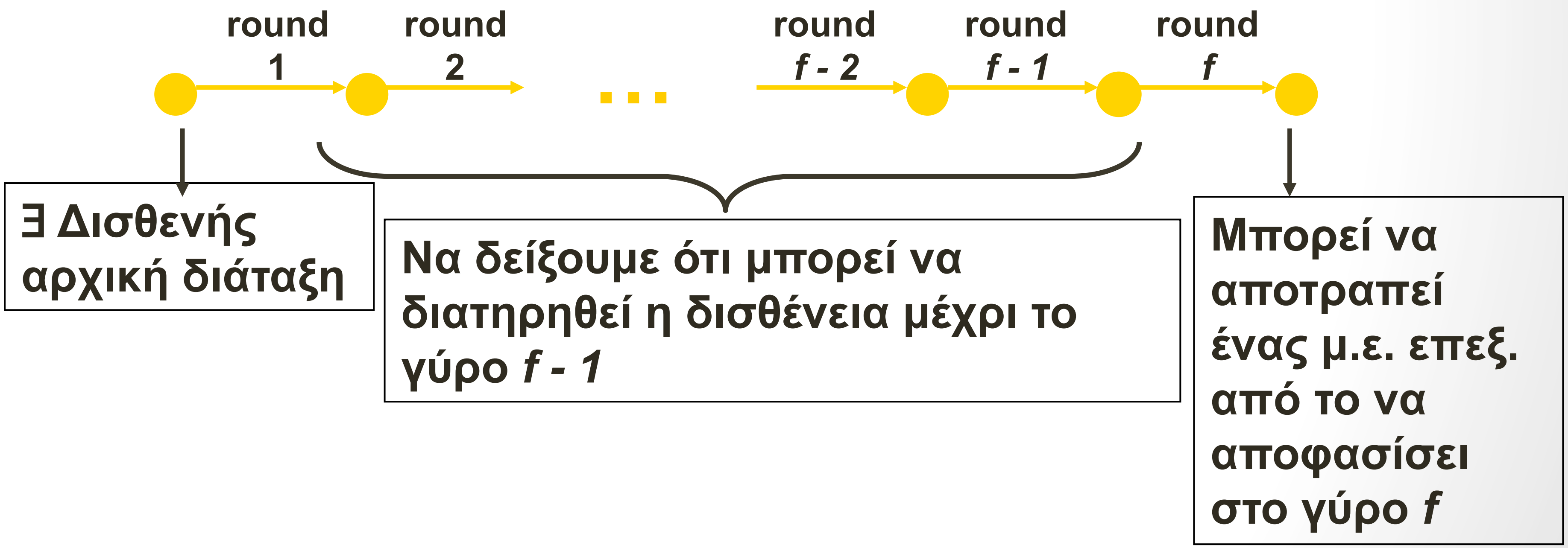
<= decisions

0/1 : bivalent
1 : 1-valent
0 : 0-valent

Κάτω Φράγμα

Θεώρημα: Κάθε αλγόριθμος συμφωνίας ανεκτικός σε *f* σφάλματα κατάρρευσης χρειάζεται τουλάχιστον *f*+1 γύρους στην χειρότερη περίπτωση.

Στρατηγική Απόδειξης:



Ύπαρξη Δισθενής Αρχικής Διάταξης

- Ας υποθέσουμε χάριν της αντίφασης ότι όλες οι αρχικές διατάξεις είναι μονοσθενείς
- Επομένως:

	inputs	valency
	000…00	0
	000…01	?
	000…11	?
	…	
I_0	001…11	? 0
I_1	011…11	? 1
	111…11	1

Από την συνθήκη εγκυρότητας

Πρέπει να υπάρχουν δύο κοντινές διατάξεις I_0, I_1 τ.ω. η πρώτη είναι 0-σθενής και η δεύτερη 1-σθενής

Ύπαρξη Δισθενής Αρχικής Διάταξης

- Έστω
  - I_0 είναι 0-σθενή αρχική διάταξη
  - I_1 είναι 1-σθενή αρχική διάταξη
  - τ.ω. Διαφέρουν στην είσοδο του p_i
  - σ είναι μια ακολουθία γεγονότων

σ

I_0 **p_i καταρρέει από την αρχή, και δεν έχουμε άλλα σφάλματα. Από την συνθήκη τερματισμού όλοι οι υπόλοιποι αποφασίζουν.**

Όλοι εκτός του p_i επιστρέφουν 0

σ

I_1 ***! "#$#"% & ' &(&) *"+, -% &. , /% " ' / ,0# p_i 1&# μ ' 0)&2 #" (&34)2+&5 "6, - ,7# &. ,$8&+7 " ' / ,7# ') 9 ,7.***

Όλοι εκτός του p_i επιστρέφουν 0

Contradiction!

Διατήρηση Δισθένειας

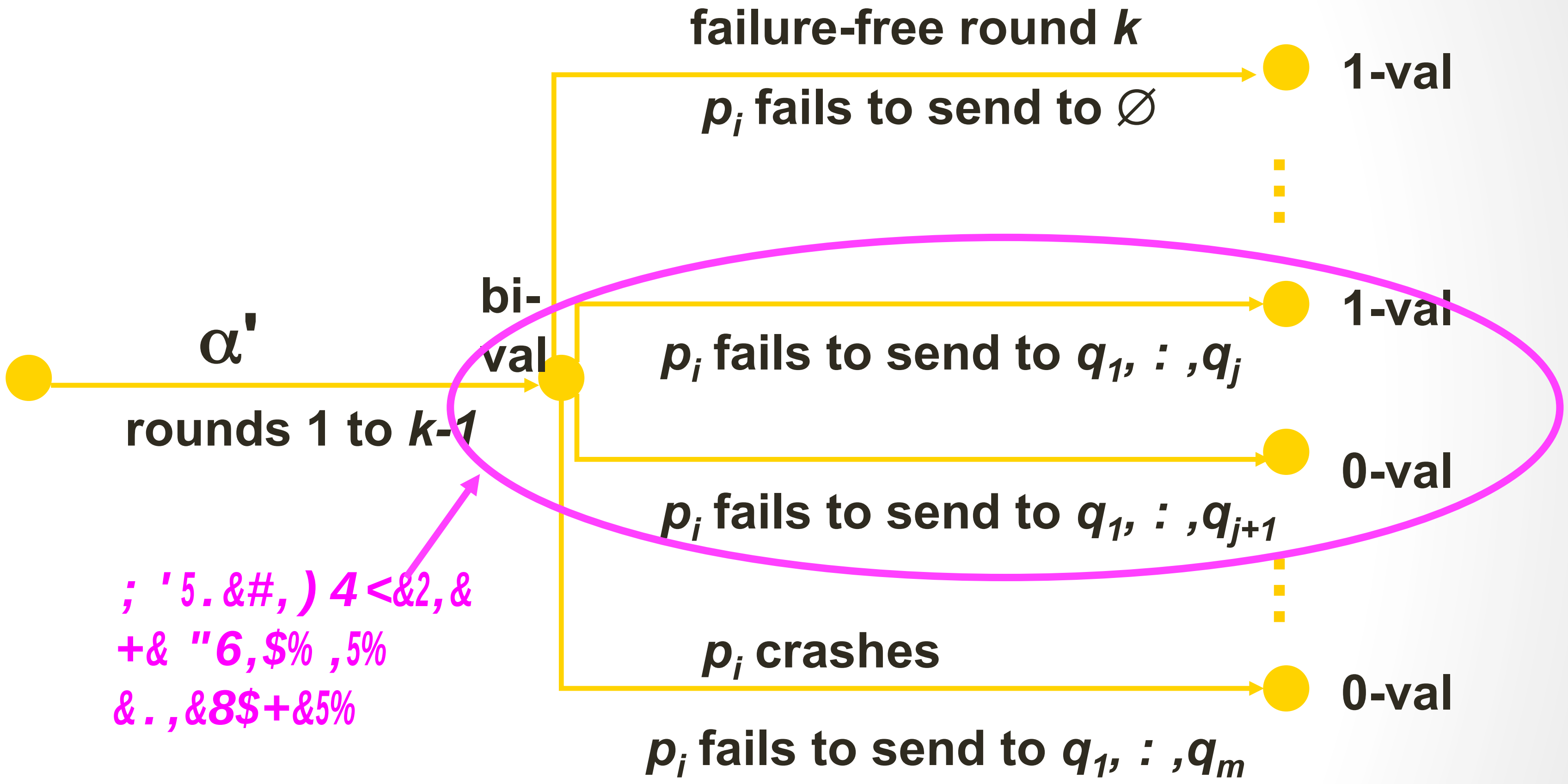
- Έστω α' είναι μια (αραιά-εσφαλμένη) εκτέλεση *k-1* γύρων που καταλήγει σε μια δισθενή διάταξη.
  - για $k - 1 < f - 1$

- Θέλουμε να δείξουμε ότι υπάρχει επέκταση της α' με ένα γύρο (έστω α) η οποία επίσης καταλήγει σε δισθενή διάταξη
  - επομένως α έχει $k < f$ γύρους

- Για να φτάσουμε στο επιθυμητό αποτέλεσμα υποθέτουμε με αντίφαση ότι κάθε επέκταση της α' με ένα γύρο καταλήγει σε μονοσθενή διάταξη

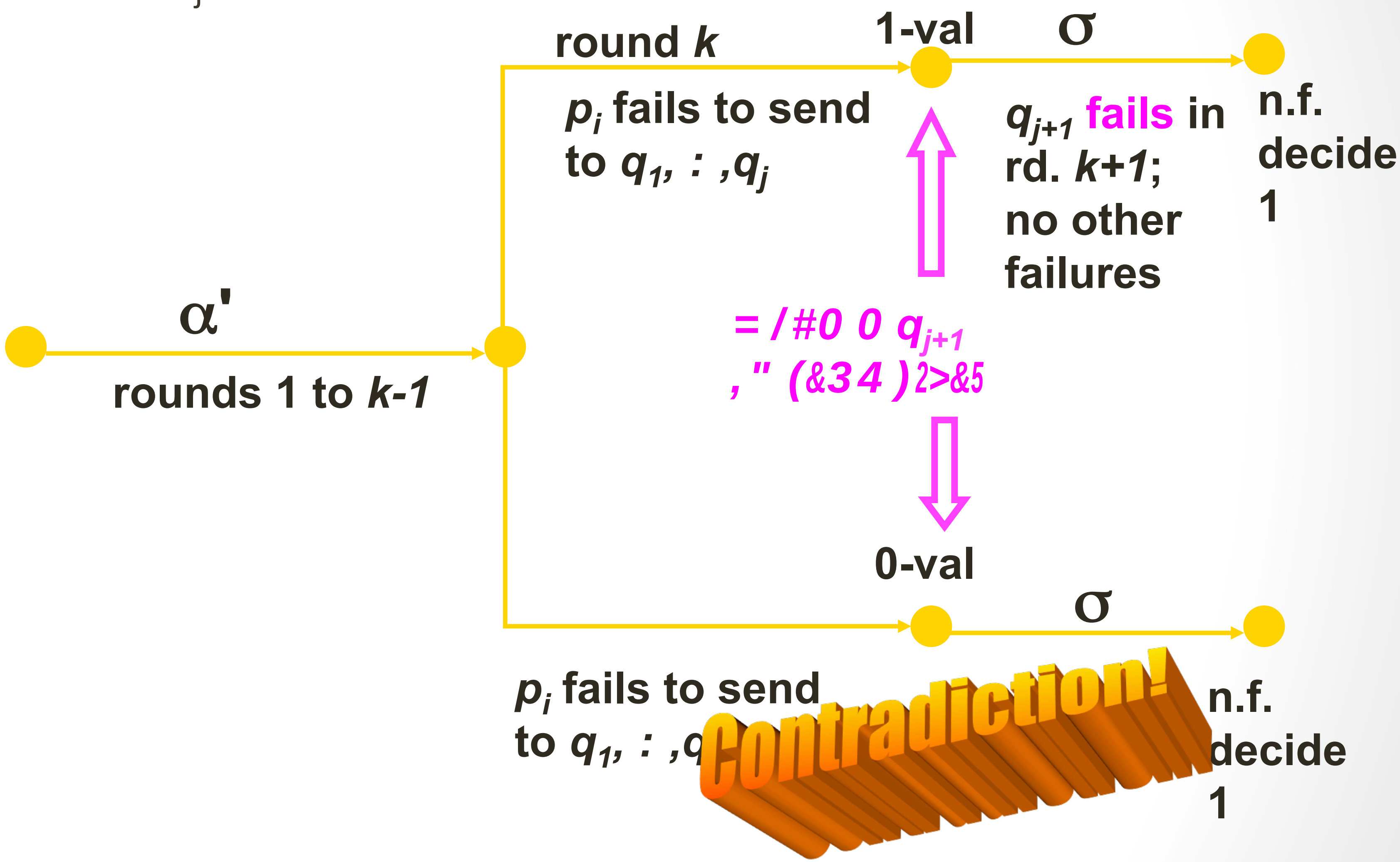
Διατήρηση Δισθένειας

- Πιθανές εκτελέσεις α:



Διατήρηση Διαθένειας

- Μόνο ο q_j μπορεί να αποκαλύψει την διαφορά των εκτελέσεων!



Δεν μπορούμε να αποφασίσουμε στο γύρο *f*

- Δείξαμε ότι μπορούμε να κατασκευάσουμε μια αραιά-εσφαλμένη εκτέλεση α, *f - 1* γύρων, που καταλήγει σε μια δισθενή διάταξη.

- Τι θα συμβεί αν επεκτήνουμε την α κατά ένα ακόμα γύρο (δηλαδή να έχει συνολικά *f* γύρους);
  - Μπορεί να χαθεί η δισθένεια
  - Αλλά μπορούμε να εμποδίσουμε ένα επεξεργαστή από το να αποφασίσει στον γύρο *f* και έτσι θα χρειαστούμε ακόμα ένα γύρο (*συνολικά f+1 γύρους)*

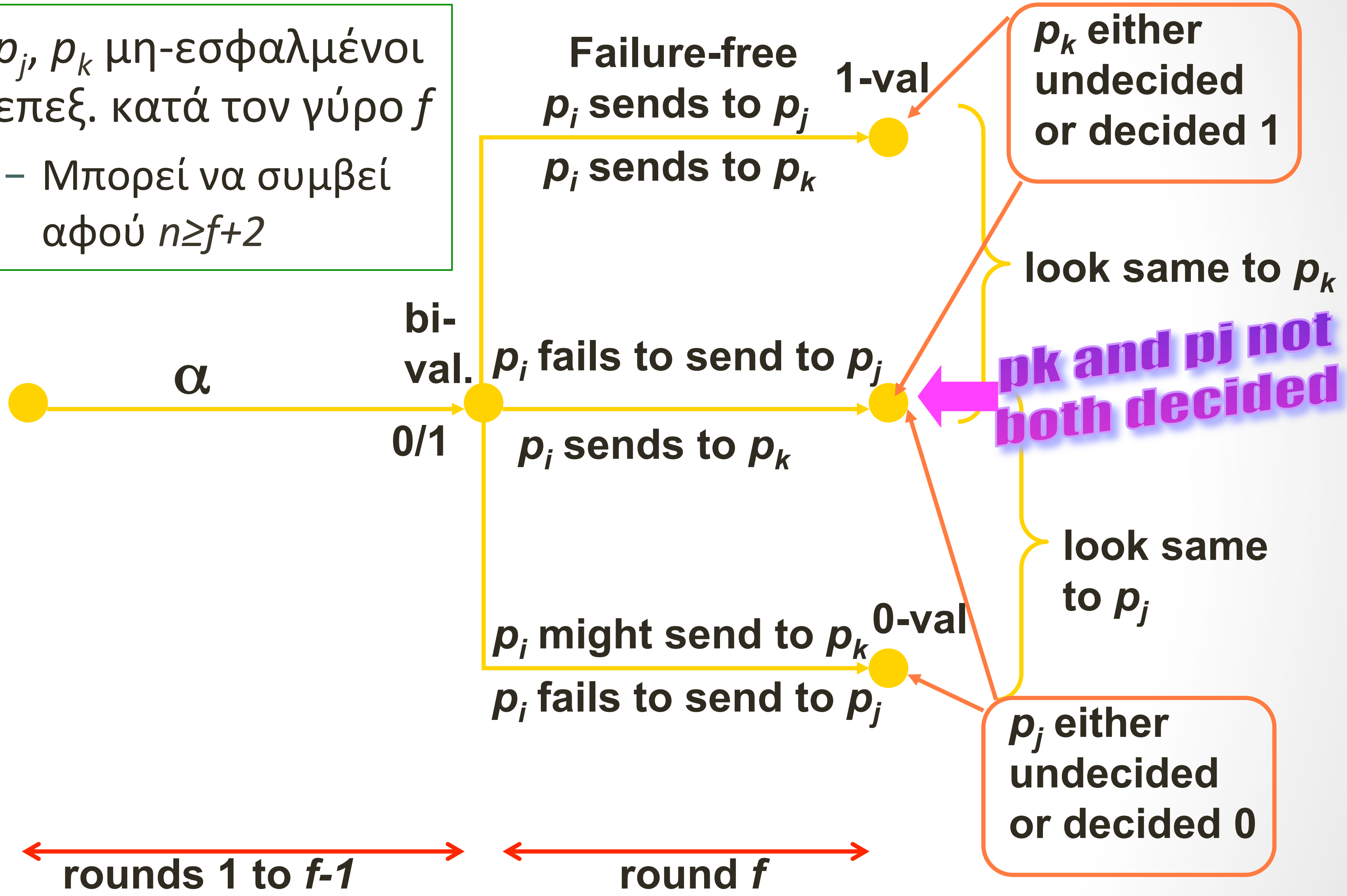
Δεν μπορούμε να αποφασίσουμε στο γύρο f

- **Περίπτωση 1:** Μετά από κάποιο επιπρόσθετο γύρο φθάνουμε σε δισθενή διάταξη
  - Τότε δεν μπορεί να παρθεί απόφαση

- **Περίπτωση 2:** Όλες οι επεκτάσεις ενός γύρου του α μας οδηγούν σε μονοσθενή διάταξη.

Δεν μπορούμε να αποφασίσουμε στο γύρο f

- p_j, p_k μη-εσφαλμένοι επεξ. κατά τον γύρο f
  - Μπορεί να συμβεί αφού $n \geq f+2$



Ερωτήσεις;