ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός

Ενότητα # 3: Καθολικά κατηγορήματα

Διδάσκων: Γεώργιος Ξυλωμένος

Τμήμα: Πληροφορικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Οικονομικό Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

Σκοποί ενότητας

- Κατανόηση της έννοιας του καθολικού κατηγορήματος και των ιδιοτήτων του.
- Εισαγωγή στους αλγόριθμους εντοπισμού αδιεξόδων σε κατανεμημένους υπολογισμούς.
- Εξοικείωση με τους αλγορίθμους ανίχνευσης κατανεμημένου τερματισμού.

Περιεχόμενα ενότητας

- Ιδιότητες καθολικών κατηγορημάτων
- Αδιέξοδα
  - Συγκεντρωτική και ιεραρχική ανίχνευση
  - Κατανεμημένη ανίχνευση
- Κατανεμημένος τερματισμός

Ιδιότητες καθολικών κατηγορημάτων

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 3: Καθολικά κατηγορήματα

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Ιδιότητες κατηγορημάτων (1 από 6)

- Καθολικό κατηγόρημα
  - Λογική συνάρτηση των μεταβλητών υπολογισμού
  - Πολλά προβλήματα ανάγονται σε αυτό
    - Ανίχνευση αδιεξόδων και τερματισμού
    - Απώλεια σκυτάλης
    - Αποκομιδή απορριμμάτων
    - Καθορισμός σημείου ελέγχου και επανεκκίνηση
    - Παρακολούθηση και εκσφαλμάτωση
    - Αναδιάρθρωση συστήματος

Ιδιότητες κατηγορημάτων (2 από 6)

- Αποτίμηση καθολικού κατηγορήματος
  - Η κατάσταση ικανοποιεί το κατηγόρημα Φ;
    - Για παράδειγμα, έχει συμβεί αδιέξοδο;
  - Απαιτεί κατασκευή καθολικής κατάστασης
- Σταθερά κατηγορήματα
  - Όταν γίνουν αληθή, παραμένουν αληθή
    - Αδιέξοδο
    - Τερματισμός
    - Απώλεια σκυτάλης

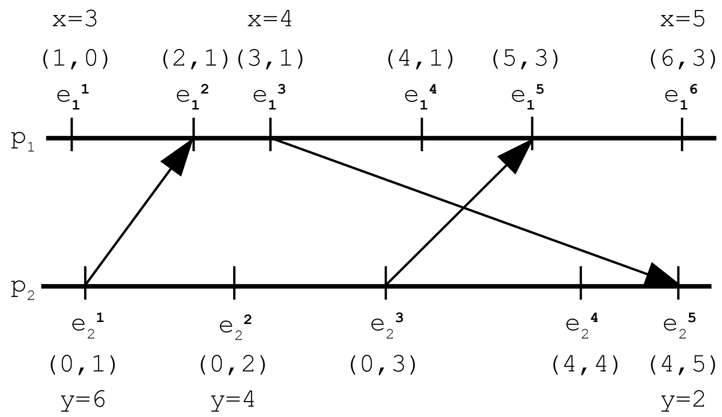
Ιδιότητες κατηγορημάτων (3 από 6)

- Σταθερά κατηγορήματα
  - Έστω η ακολουθία καταστάσεων $\Sigma^a \rightarrow \Sigma^s \rightarrow \Sigma^f$
    - Αρχική-τελική κατάσταση και ενδιάμεση παρατήρηση
  - Για ένα σταθερό κατηγόρημα έχουμε
    - $\Phi(\Sigma^s) = \text{true} \Rightarrow \Phi(\Sigma^f) = \text{true}$
  - Επιπλέον έχουμε
    - $\Phi(\Sigma^s) = \text{false} \Rightarrow \Phi(\Sigma^a) = \text{false}$
    - Δεν μπορεί να γίνει αληθές και μετά ψευδές

Ιδιότητες κατηγορημάτων (4 από 6)

- Ασταθή κατηγορήματα
  - Μπορεί να γίνουν αληθή και μετά ψευδή
    - Παράδειγμα: ισότητα δύο δυναμικών ουρών
- Δύο βασικά προβλήματα
  - Δεν είναι απαραίτητο να ανιχνεύσουμε το Φ
    - Μπορεί να γίνεται αληθές σε κάθε εκτέλεση
    - Αλλά να μην ισχύει μόνιμα
    - Άρα να μην το ανιχνεύσουμε

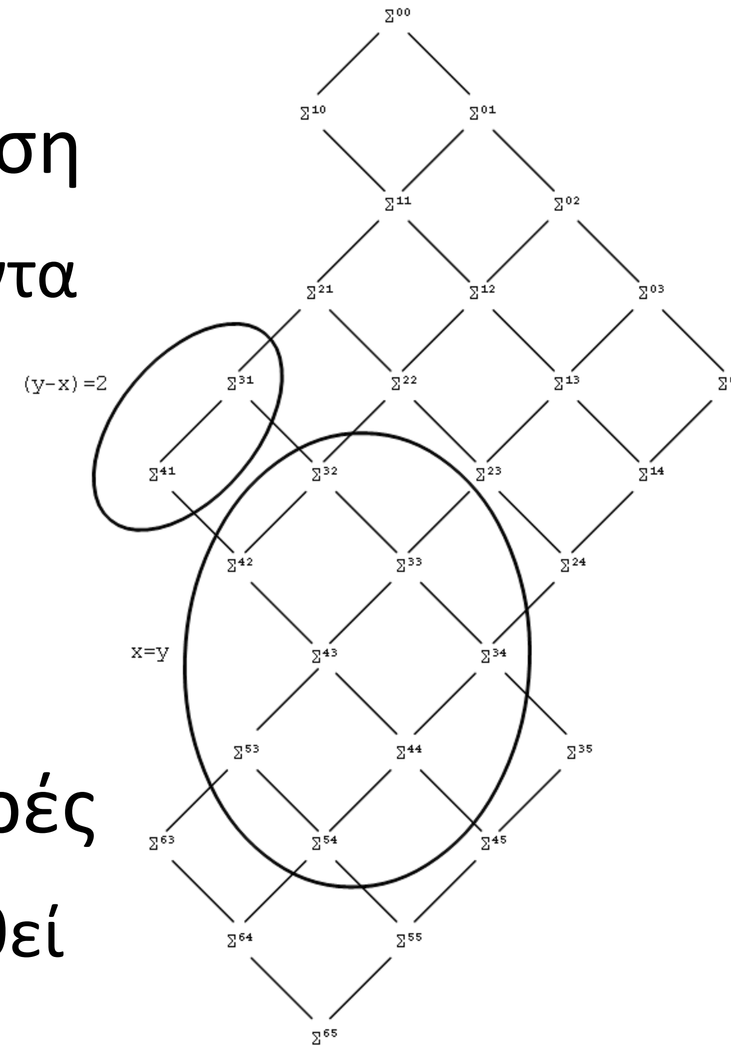
Ιδιότητες κατηγορημάτων (5 από 6)



- Δύο βασικά προβλήματα
  - Το Φ μπορεί να βρεθεί αληθές αλλά να μην ίσχυε
    - Μπορεί να είναι αληθές σε κάποια εκτέλεση
    - Αλλά όχι απαραίτητα στη συγκεκριμένη εκτέλεση
- Παράδειγμα: x=y και (y-x)=2

Ιδιότητες κατηγορημάτων (6 από 6)

- x=y: ισχύει σε κάθε εκτέλεση
  - Αλλά δεν παρατηρείται πάντα
- (y-x)=2: ισχύει μερικές φορές
  - Αλλά μπορεί να παρατηρηθεί



Possibly και Definitely (1 από 6)

- Επέκταση σε ολόκληρο τον υπολογισμό
  - Εξετάζουμε όλες τις συνεπείς παρατηρήσεις
- Possibly(Φ)
  - $\exists$ O : $\exists$ $\Sigma^l \in$ O τέτοιο ώστε $\Phi(\Sigma^l)$ = TRUE
  - Υπάρχει τέτοια κατάσταση σε κάποια παρατήρηση
- Definitely(Φ)
  - $\forall$ O : $\exists$ $\Sigma^l \in$ O τέτοιο ώστε $\Phi(\Sigma^l)$ = TRUE
  - Υπάρχει τέτοια κατάσταση σε κάθε παρατήρηση

Possibly και Definitely (2 από 6)

- Παράδειγμα
  - Possibly(y-x=2)
    - Υπάρχει μία εκτέλεση στην οποία θα ισχύσει
    - Αν είναι σφάλμα, τότε μπορεί να συμβεί
  - Definitely(x=y)
    - Ισχύει οπωσδήποτε σε όλες τις εκτελέσεις
    - Αν είναι σφάλμα, θα συμβεί σίγουρα
- Παρατηρούμε ότι
  - $\neg$Possibly(Φ) $\Rightarrow$ Definitely($\neg$Φ)

Possibly και Definitely (3 από 6)

- Αποτίμηση Possibly(Φ)
  - Αρκεί να ισχύει σε κάποια εκτέλεση
  - Δεν ξέρουμε όμως σε ποια!
  - Αναζήτηση σε όλες τις συνεπείς καταστάσεις
    - Ξεκινάμε από αρχική κατάσταση
    - Προχωράμε από επίπεδο σε επίπεδο στο πλέγμα
    - Αν ισχύει το Φ σε κάποια κατάσταση, Possibly (Φ)
    - Αν φτάσουμε στο τέλος, ¬Possibly(Φ)

Possibly και Definitely (4 από 6)

- states=$\{\Sigma^{0\ldots0}\}$, L=0, Possibly(Φ) = true
- Όσο $\neg\Phi(S)\ \forall\ S \in$ states και Possibly(Φ)
  - Αν states=={τελική καθολική κατάσταση} τότε Possibly(Φ) = false
  - L=L+1
  - states={όλες οι καταστάσεις επιπέδου L που είναι προσπελάσιμες από τις καταστάσεις του states}

Possibly και Definitely (5 από 6)

- Αποτίμηση Definitely(Φ)
  - Εντοπισμός συνόλου καταστάσεων όπου:
    - Σε όλες ισχύει το Φ
    - Κάθε υπολογισμός περνάει από μία τους
    - Ξεκινάμε από αρχική κατάσταση
    - Προχωράμε από επίπεδο σε επίπεδο στο πλέγμα
    - Κρατάμε μόνο καταστάσεις που δεν ικανοποιούν το Φ
    - Αν δεν μείνει καμία, τότε Definitely(Φ)

Possibly και Definitely (6 από 6)

- Αν $\Phi(\Sigma^{0\ldots0})$ states={} αλλιώς states ={$\Sigma^{0\ldots0}$};
- L=0, Definitely(Φ) = true
- Όσο states <> {} και Definitely(Φ)
  - L=L+1
  - reachable={όλες οι καταστάσεις επιπέδου L που είναι προσπελάσιμες από αυτές του states}
  - states={S $\in$ reachable : ¬Φ(S)}
  - Αν states=={τελική καθολική κατάσταση} τότε Definitely(Φ) = false

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Αδιέξοδα

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 3: Καθολικά κατηγορήματα

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

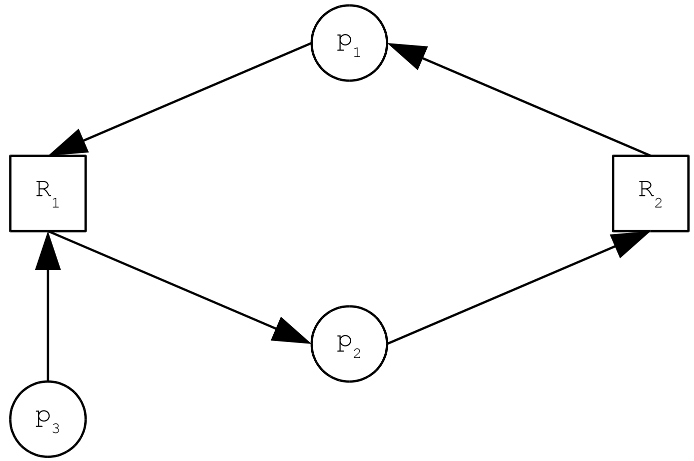
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Κατανεμημένα αδιέξοδα (1 από 4)

- Παρόμοια με συγκεντρωτικά συστήματα
  - Εμποδισμός διεργασιών λόγω πόρων
  - Απλά οι πόροι είναι κατανεμημένοι
  - Η βασική συνθήκη είναι η κυκλική αναμονή
  - Πώς ξέρουμε αν έχει συμβεί;

Κατανεμημένα αδιέξοδα (2 από 4)

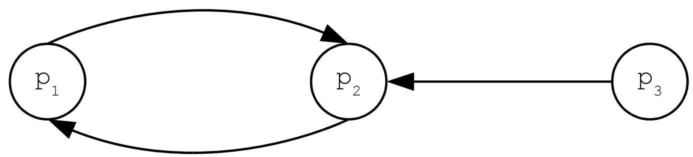


- Γράφος καταχώρισης πόρων
  - Δείχνει καταχώριση πόρων και ζήτηση πόρων
    - Τόξο από διεργασία: ζητάει πόρο
    - Τόξο προς διεργασία: έχει δεσμεύσει πόρο
  - Ενημερώνεται δυναμικά

Κατανεμημένα αδιέξοδα (3 από 4)

- Μία μονάδα ανά πόρο
  - Αρκεί να υπάρχει κύκλος στο γράφο
- Πολλές μονάδες ανά πόρο
  - Πρέπει να έχουμε δεσμό στο γράφο
  - Δεσμός είναι ένα μη κενό σύνολο κόμβων Κ
    - Όπου όλοι οι προσπελάσιμοι κόμβοι από το Κ…
    - …είναι και αυτοί μέλη του Κ
    - Περιλαμβάνει τουλάχιστον έναν κύκλο

Κατανεμημένα αδιέξοδα (4 από 4)



- Γράφος αναμονής
  - Απλούστερη μορφή γράφου καταχώρισης
  - Παραλείπει τους πόρους
    - Ισχύει όταν έχουμε μία μονάδα κάθε πόρου
  - Αν έχει κύκλο, έχουμε αδιέξοδο
    - Οι διεργασίες περιμένουν η μία την άλλη

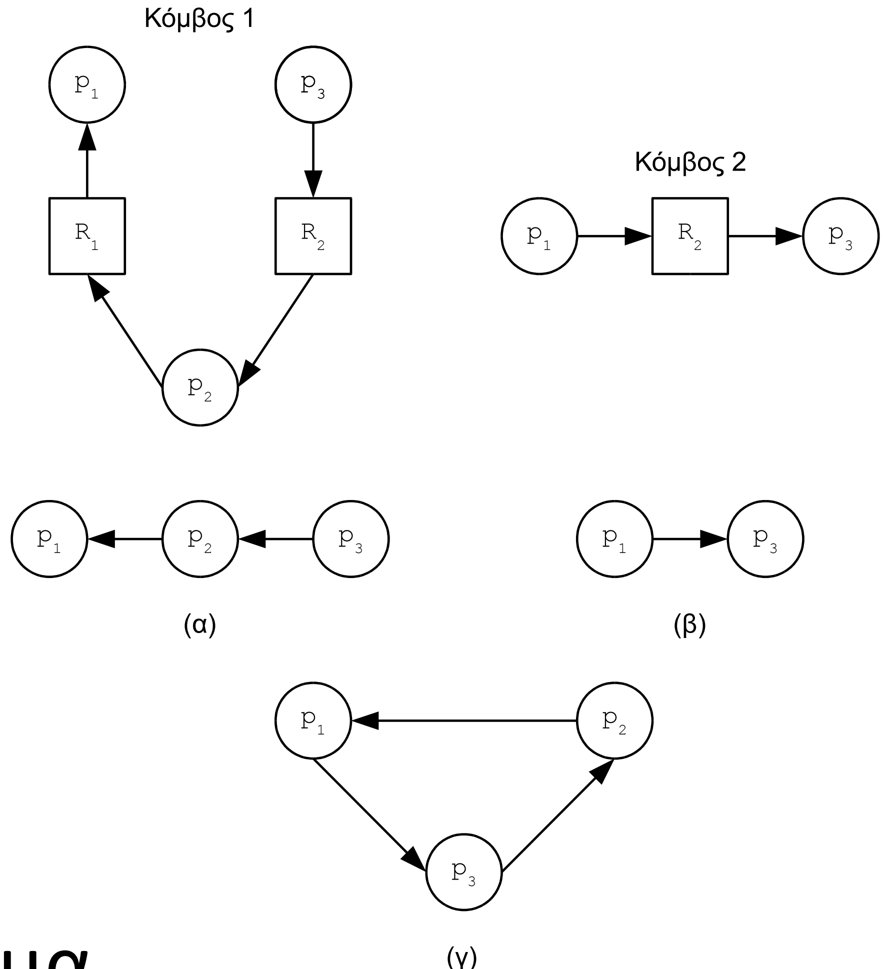
Χειρισμός αδιεξόδων

- Έγκριση
  - Μόνο σε συγκεκριμένες εφαρμογές
- Αποφυγή
  - Απαιτεί γνώση απαιτήσεων εκ των προτέρων
- Πρόληψη
  - Συνήθως αποφεύγουμε την κυκλική αναμονή
- Ανίχνευση και επανόρθωση
  - Η πιο συνηθισμένη στρατηγική

Ανίχνευση αδιεξόδων (1 από 4)

- Κατανεμημένη ανίχνευση αδιεξόδων
  - Πληροφορίες διασκορπισμένες σε κόμβους
- Κατασκευή καθολικού γράφου αναμονής
  - Κάθε κόμβος κατασκευάζει γράφο καταχώρισης
  - Από αυτόν κατασκευάζει γράφο αναμονής
  - Συνδυάζουμε για τον καθολικό γράφο αναμονής
  - Εντοπίζουμε αδιέξοδα (κύκλους)

Ανίχνευση αδιεξόδων (2 από 4)



- Παράδειγμα

Ανίχνευση αδιεξόδων (3 από 4)

- Τρεις τρόποι κατασκευής
  - Συγκεντρωτικός
    - Ένας κόμβος συντονιστής
  - Ιεραρχικός
    - Ιεραρχία συντονιστών
    - Κατασκευή γράφου ιεραρχικά
  - Κατανεμημένος
    - Συμμετέχουν όλοι
    - Καθένας κατασκευάζει το γράφο

Ανίχνευση αδιεξόδων (4 από 4)

- Επιθυμητές ιδιότητες αλγορίθμων
  – Πρόοδος
    - Ανίχνευση σε πεπερασμένο χρόνο
  – Ασφάλεια
    - Ανίχνευση πραγματικών αδιεξόδων
- Αδιέξοδο-φάντασμα
  – Αδιέξοδο που δεν υπάρχει πραγματικά
  – Οφείλεται σε καθυστερήσεις μηνυμάτων

Αποκατάσταση αδιεξόδων

- Επέμβαση χειριστή: ποιού όμως;
  - Αν ξεκινήσουν όλοι την αντιμετώπιση;
  - Αν ένας κόμβος ευνοεί τις δικές του διεργασίες;
- Τερματισμός διεργασιών
  - Επιλέγει τις «καλύτερες» διεργασίες
  - Υψηλό κόστος λόγω σπατάλης πόρων
- Οπισθοδρόμηση διεργασιών
  - Στο προηγούμενο σημείο ελέγχου
  - Θα πρέπει όμως να διατηρούμε σημεία ελέγχου

Συγκεντρωτική και ιεραρχική ανίχνευση

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 3: Καθολικά κατηγορήματα

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Συγκεντρωτική ανίχνευση (1 από 2)

- Τοπικοί συντονιστές
  - Κατασκευή τοπικών γράφων αναμονής
  - Μπορεί να εντοπίσουν τοπικό αδιέξοδο
- Καθολικός συντονιστής
  - Συγκεντρωτικός ανιχνευτής αδιεξόδων
  - Κατασκευή καθολικού γράφου αναμονής
  - Μπορεί να εντοπίσει καθολικό αδιέξοδο

Συγκεντρωτική ανίχνευση (2 από 2)

- Αντιμετώπιση αδιεξόδου
  - Στο χαμηλότερο δυνατό επίπεδο (τοπικό)
- Πότε συλλέγονται πληροφορίες;
  - Συνεχής μετάδοση: σε κάθε αλλαγή
  - Περιοδική μετάδοση: μετά από n αλλαγές
  - Μετάδοση κατ’απαίτηση: το ζητά ο συντονιστής
  - Πάντα στέλνονται μόνο οι αλλαγές
    - Προσθαφαιρέσεις ακμών στο γράφο

Ιεραρχική ανίχνευση (1 από 3)

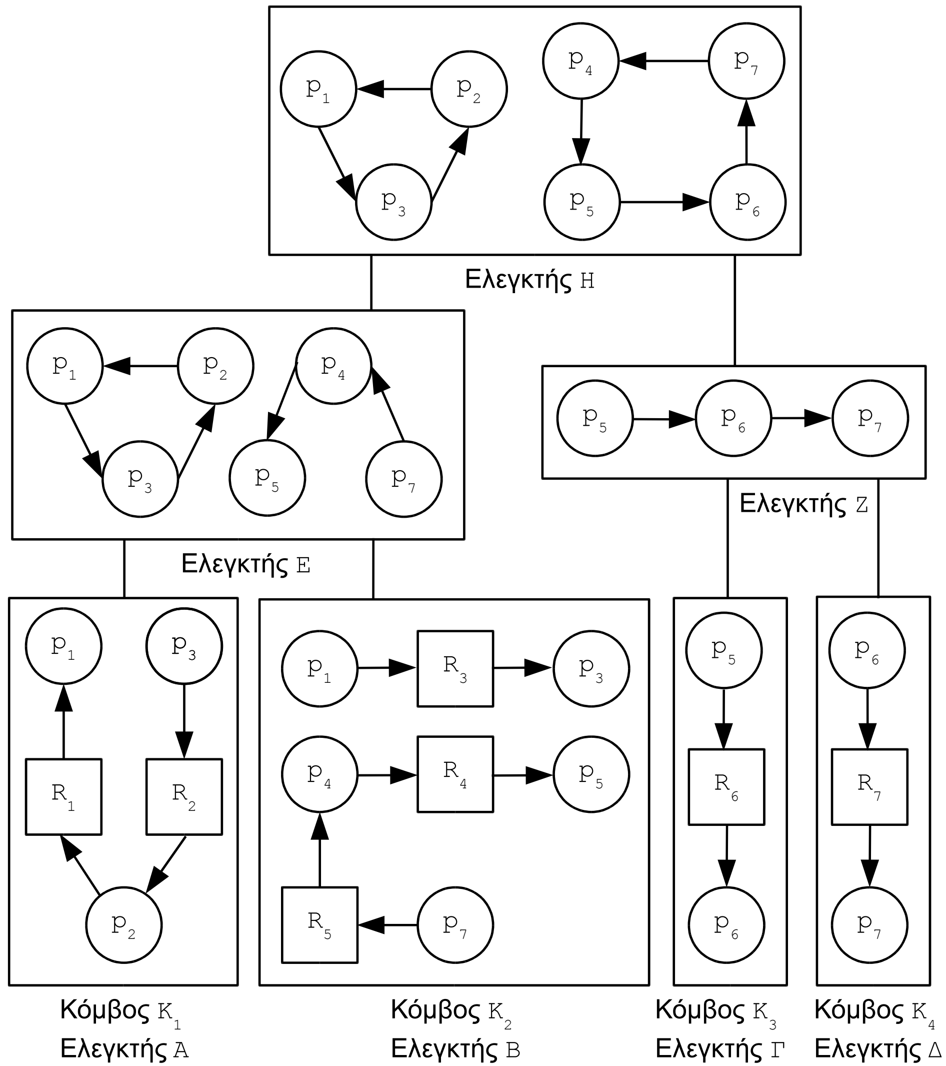
- Συνήθως οι κύκλοι είναι μικροί
  - Το 90% εμπλέκουν δύο διεργασίες
    - Δεν χρειάζεται καθολικός γράφος
  - Τοπική ανίχνευση και αντιμετώπιση
    - Ελαχιστοποίηση κόστους επικοινωνίας
- Ιεραρχική προσέγγιση
  - Λογικό δένδρο από ελεγκτές
  - Κάθε ελεγκτής είναι αρμόδιος για μια περιοχή

Ιεραρχική ανίχνευση (2 από 3)

- Ιεραρχική προσέγγιση
  – Κάθε ελεγκτής έχει κάποια εμβέλεια
    - Τα φύλλα ελέγχουν τον τοπικό γράφο
    - Οι κόμβοι ελέγχουν την ένωση των παιδιών
  – Αντιμετώπιση αδιεξόδων
    - Από τον κατώτερο ιεραρχικά κόμβο
      – Όσο πιο τοπικά γίνεται
    - Αν δεν βρούμε κύκλους, πάμε παραπάνω

Ιεραρχική ανίχνευση (3 από 3)

- Παράδειγμα
  - 4 κόμβοι
  - 7 ελεγκτές



Κατανεμημένη ανίχνευση

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 3: Καθολικά κατηγορήματα

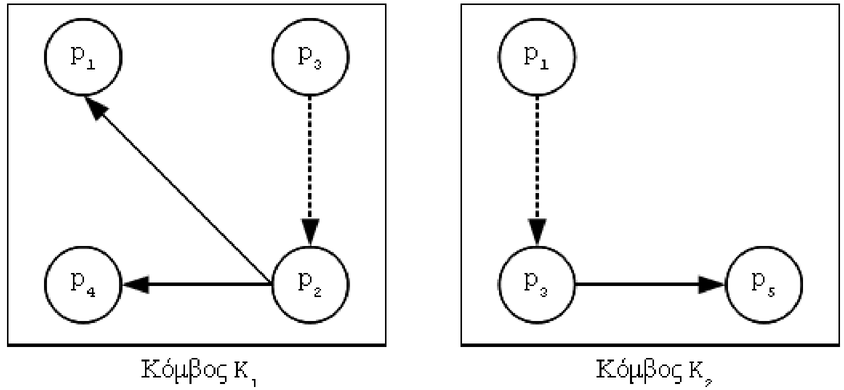
Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Αλγόριθμος προώθησης (1 από 8)

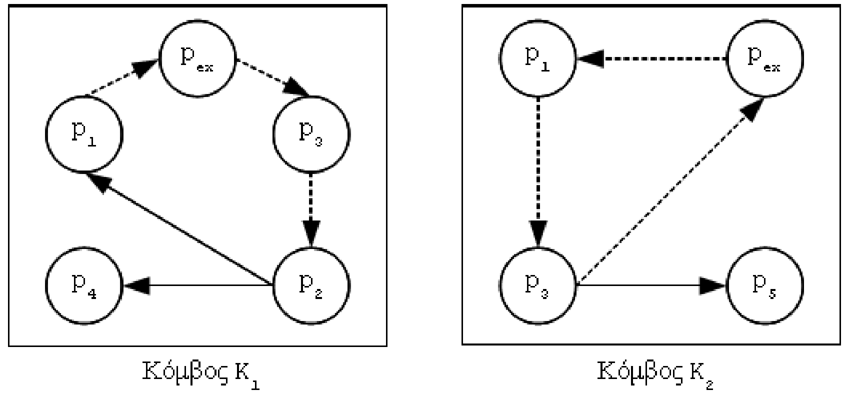
- Κάθε κόμβος υπεύθυνος για ανίχνευση
- Αλγόριθμος βασισμένος σε γράφο αναμονής
  - Ή αλγόριθμος προώθησης διαδρομών
  - Κάθε κόμβος διατηρεί τροποποιημένο γράφο
    - Περιέχει τις τοπικές ακμές και έναν κόμβο p_{ex}
    - Επιπλέον προστίθενται δύο είδη ακμών
    - (p_i, p_{ex}): εσωτερική p_i περιμένει εξωτερικό πόρο
    - (p_{ex}, p_j): εξωτερική p_j περιμένει τοπικό πόρο

Αλγόριθμος προώθησης (2 από 8)



- Παράδειγμα: έστω οι κόμβοι K_1 και K_2
  - Ο K_1 έχει τις διεργασίες p_1, p_2 και p_4
  - Ο K_2 έχει τις διεργασίες p_3 και p_5
  - Οι κόμβοι γνωρίζουν τις άμεσες εξωτερικές αιτήσεις
    - Ο K_1 γνωρίζει την αίτηση του p_3
    - Ο K_2 γνωρίζει την αίτηση του p_1

Αλγόριθμος προώθησης (3 από 8)



- Παράδειγμα: προσθήκη ακμών
  - K_1: (p_1,p_{ex}) αφού η p_1 ζητάει κάτι εκτός K_1
  - K_1: (p_{ex},p_3) αφού η p_3 ζητάει κάτι από τον K_1
  - K_2: (p_3,p_{ex}) αφού η p_3 ζητάει κάτι εκτός K_2
  - K_2: (p_{ex},p_1) αφού η p_1 ζητάει κάτι από τον K_2

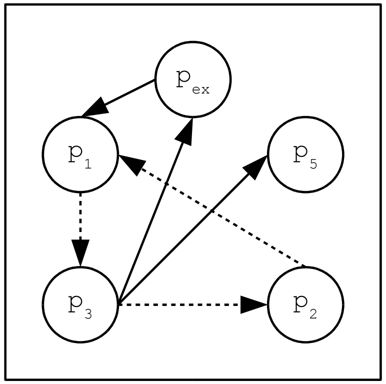
Αλγόριθμος προώθησης (4 από 8)

- Έλεγχος για κύκλους
  - Αν δεν περιέχει τον p_{ex}, τοπικό αδιέξοδο
  - Αν περιέχει τον p_{ex}, ίσως να έχουμε αδιέξοδο
    - Πρέπει να συμβουλευτούμε και άλλους κόμβους
  - Έστω ο K_i έχει βρει κύκλο (p_{ex},p_i,p_j,…,p_k,p_{ex})
  - Έστω ότι επόμενος στον κύκλο είναι ο K_j
    - Η διεργασία p_k εκτελείται στον K_i
    - Η p_k περιμένει πόρο από τον κόμβο K_j

Αλγόριθμος προώθησης (5 από 8)

- Αλγόριθμος εντοπισμού κύκλου (path pushing)
  - Ο K_i στέλνει στον K_j τον κύκλο
  - Ο K_j ενημερώνει τον γράφο του με τις ακμές
  - Αν ο νέος γράφος δεν περιέχει κύκλο, τερματισμός
  - Αν ο νέος γράφος περιέχει κύκλο χωρίς τον p_{ex}
    - Εντοπίστηκε κύκλος με τις εμπλεκόμενες διεργασίες
  - Αν ο νέος γράφος περιέχει κύκλο με τον p_{ex}
    - Ο K_j στέλνει τον νέο κύκλο στον κόμβο K_k
    - Αναδρομική εκτέλεση αλγορίθμου

Αλγόριθμος προώθησης (6 από 8)

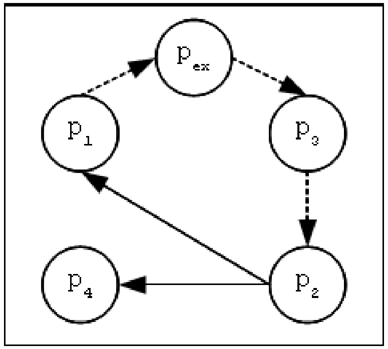


- Έστω το προηγούμενο παράδειγμα
  - Ο K_1 εντοπίζει τον κύκλο $(p_{ex},p_3,p_2,p_1,p_{ex})$
  - Η p_1 περιμένει πόρο από τον K_2
  - Ο κύκλος στέλνεται στον K_2
  - Ο K_2 προσθέτει τις ακμές που δεν περιέχουν τον P_{ex}
  - Ο K_2 εντοπίζει τον κύκλο (p_3,p_2,p_1)

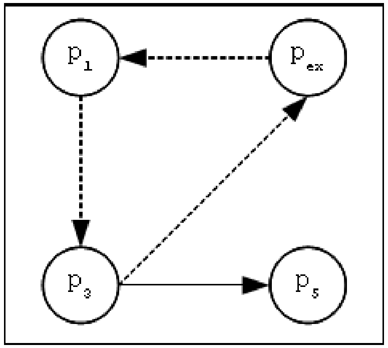
Αλγόριθμος προώθησης (7 από 8)

- Αν ξεκινήσουν ταυτόχρονα δύο κόμβοι;
  - Θα εντοπίσουν και οι δύο τον κύκλο
  - Σπατάλη μηνυμάτων
  - Πιθανόν περιττοί τερματισμοί διεργασιών
- Χρήση μοναδικού ID διεργασίας
  - Όταν εντοπίζεται κύκλος (p_{ex},p_i,p_j,…,p_k,p_{ex})
    - Αν $ID(p_k) < ID(p_i)$ ο κόμβος εκτελεί τον αλγόριθμο
    - Αλλιώς αφήνει την εκτέλεση σε άλλους

Αλγόριθμος προώθησης (8 από 8)



Κόμβος K_1



Κόμβος K_2

- Συνέχεια προηγούμενου παραδείγματος
  - Έστω $ID(p_1)<ID(p_2)<ID(p_3)<ID(p_4)<ID(p_5)$
  - Ο K_1 εντοπίζει τον κύκλο $(p_{ex},p_3,p_2,p_1,p_{ex})$
  - Αφού $ID(p_1)<ID(p_3)$, ο K_1 ξεκινά τον αλγόριθμο
  - Ο K_2 εντοπίζει τον κύκλο (p_{ex},p_1,p_3,p_{ex})
  - Αφού $ID(p_3)>ID(p_1)$, ο K_2 δεν ξεκινά τον αλγόριθμο

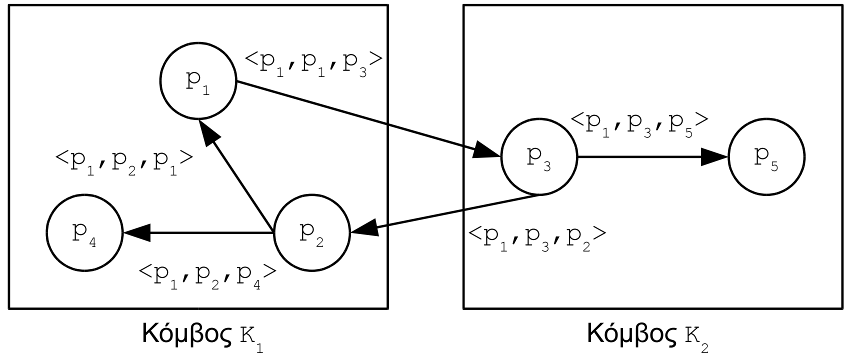
Αλγόριθμος ιχνηλασίας (1 από 4)

- Αλγόριθμος βασισμένος στη διερεύνηση
  - Ή αλγόριθμος ιχνηλασίας ακμών
  - Δεν κατασκευάζει καθολικό γράφο
  - Ακολουθεί τις διαδρομές με μηνύματα
  - Κάθε διεργασία p_i έχει ένα μοναδικό ID
  - Αν η αίτηση της p_i στην p_j καθυστερεί πολύ
    - Η p_i στέλνει μήνυμα διερεύνησης στην p_j
    - Πεδία: εμποδιζόμενη, αποστολέας, παραλήπτης

Αλγόριθμος ιχνηλασίας (2 από 4)

- Αλγόριθμος βασισμένος στη διερεύνηση
  - Αν η p_j δεν είναι εμποδισμένη
    - Δεν υπάρχει αδιέξοδο, τέλος προώθησης
  - Αν η p_j είναι εμποδισμένη προωθεί το μήνυμα
    - Σε όλες τις p_k τις οποίες περιμένει
    - Δεύτερο πεδίο: δικό της αναγνωριστικό (p_j)
    - Τρίτο πεδίο: αναγνωριστικό του παραλήπτη (p_k)
  - Αν η p_j δει το δικό της ID στο πρώτο πεδίο
    - Έχει συμβεί αδιέξοδο

Αλγόριθμος ιχνηλασίας (3 από 4)



- Έστω το προηγούμενο παράδειγμα
  - Η p_1 στέλνει μήνυμα $<p_1,p_1,p_3>$ στην p_3
  - Η p_3 στέλνει το μήνυμα στις p_2 και p_5
  - Στην p_5 το μήνυμα σταματάει
  - Στην p_2 το μήνυμα προωθείται σε p_1 και p_4
  - Στην p_1 εντοπίζεται το αδιέξοδο

Αλγόριθμος ιχνηλασίας (4 από 4)

- Συνήθως χρήση δεύτερου αλγόριθμου
  - Εύκολη υλοποίηση
  - Μηνύματα σταθερού μεγέθους
  - Το πολύ ένα μήνυμα ανά ακμή
  - Δεν χρειάζεται καθολικούς γράφους
  - Δεν ανιχνεύει ανύπαρκτα αδιέξοδα
  - Δεν απαιτεί διάταξη των διεργασιών
  - Όλα αυτά δεν ισχύουν στον πρώτο αλγόριθμο!

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Κατανεμημένος τερματισμός

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 3: Καθολικά κατηγορήματα

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Έννοια του τερματισμού

- Πότε τερματίζει ένα παράλληλο πρόγραμμα;
  - Οι διεργασίες περιμένουν εθελοντικά μηνύματα
    - Έτσι καλύπτουμε και τις άπειρες ανακυκλώσεις
  - Μη εθελοντική αναμονή = αδιέξοδο
  - Πώς όμως μπορούμε να το διαπιστώσουμε;
- Ανίχνευση κατανεμημένου τερματισμού
  - Υποθέτουμε αξιόπιστα κανάλια επικοινωνίας
  - Τα μηνύματα φτάνουν μία φορά και σωστά

Αλγόριθμος Dijkstra (1 από 5)

- Αλγόριθμος του Dijkstra
  - Έστω n κόμβοι σε ενεργή ή παθητική κατάσταση
  - Μόνο οι ενεργοί κόμβοι στέλνουν μηνύματα
  - Ο παθητικός γίνεται ενεργός όταν λάβει μήνυμα
  - Ο ενεργός γίνεται παθητικός όταν δεν περιμένει κάτι
- Κατάσταση τερματισμού
  - Όλοι οι κόμβοι είναι παθητικοί
  - Δεν υπάρχει κανένα μήνυμα καθοδόν

Αλγόριθμος Dijkstra (2 από 5)

- Δομή αλγορίθμου
  - Οι κόμβοι διατάσσονται σε δακτύλιο
    - Μπορεί να είναι λογική διάταξη
  - Κάθε κόμβος i διατηρεί έναν μετρητή c_i
    - Αρχικά $c_i=0$
    - Αυξάνεται σε κάθε αποστολή μηνύματος
    - Μειώνεται σε κάθε λήψη μηνύματος
  - Σc_i: πόσα μηνύματα εκκρεμούν
  - Σκυτάλη με άθροισμα μετρητών

Αλγόριθμος Dijkstra (3 από 5)

- Δομή αλγορίθμου
  - Όλοι οι κόμβοι και η σκυτάλη έχουν χρώμα
  - Αρχικά όλα είναι λευκά
  - Ο κόμβος γίνεται μαύρος όταν λάβει μήνυμα
    - Εξαιρείται το μήνυμα με τη σκυτάλη
  - Γίνεται λευκός όταν προωθεί τη σκυτάλη
  - Αν η σκυτάλη βρει μαύρο κόμβο, γίνεται μαύρη
    - Η σκυτάλη μετά παραμένει μαύρη

Αλγόριθμος Dijkstra (4 από 5)

- Λειτουργία αλγορίθμου
  - Όταν ο κόμβος i=0 γίνει παθητικός
    - Στέλνει λευκή σκυτάλη με τιμή 0 στον n-1
  - Όταν ο κόμβος i<>0 λάβει τη σκυτάλη
    - Κρατά τη σκυτάλη μέχρι να γίνει παθητικός
    - Προωθεί στον i-1 αυξάνοντας την τιμή κατά c_i
    - Αν ο κόμβος είναι μαύρος
      - Η σκυτάλη γίνεται μαύρη
    - Ο κόμβος γίνεται λευκός

Αλγόριθμος Dijkstra (5 από 5)

- Λειτουργία αλγορίθμου
  - Όταν ένας κόμβος λάβει μήνυμα (όχι σκυτάλη)
    - Γίνεται μαύρος
  - Όταν ο κόμβος 0 λάβει τη σκυτάλη
    - Αν είναι παθητικός και λευκός
    - και η σκυτάλη είναι λευκή
    - και τιμή σκυτάλης + c_0 είναι 0
    - Έχει συμβεί τερματισμός

Ανάκτηση πίστωσης (1 από 7)

- Τι σημαίνει τερματισμός υπολογισμού;
  - Ένα έργο j αποτελείται από καθήκοντα
    - Κάθε καθήκον εκτελείται σε έναν μόνο κόμβο
  - Αρχικά κάθε έργο περιέχει ένα καθήκον
  - Ένα καθήκον μπορεί να δημιουργήσει νέα
    - Στον ίδιο ή σε άλλους κόμβους
    - Άρα μπορούμε να έχουμε καθήκοντα καθοδόν

Ανάκτηση πίστωσης (2 από 7)

- Τι σημαίνει τερματισμός υπολογισμού;
  - terminated(j)=true: τα καθήκοντα τελείωσαν
    - Σταθερό καθολικό κατηγόρημα
  - $idle_S(j)$=true: ο S δεν έχει ενεργά καθήκοντα
    - Η αδράνεια είναι ασταθές τοπικό κατηγόρημα
    - Μπορεί να φτάσουν σε λίγο κι άλλα καθήκοντα
  - Η αδράνεια των κόμβων δεν υπονοεί τερματισμό
    - Μπορεί να έχουμε καθήκοντα καθοδόν

Ανάκτηση πίστωσης (3 από 7)

- Αλγόριθμος ανάκτησης πίστωσης
  - Κάθε έργο έχει μία αρχική πίστωση
    - Κάθε καθήκον παίρνει μέρος της πίστωσης
    - Όταν τελειώσει το καθήκον η πίστωση επιστρέφεται
  - Τερματισμός έργου: ανάκτηση όλης της πίστωσης
  - Για κάθε καθήκον t έχουμε
    - job(t) είναι το έργο που ανήκει
    - credit(t) είναι η πίστωσή του, της μορφής 2^{-i}

Ανάκτηση πίστωσης (4 από 7)

- Αλγόριθμος ανάκτησης πίστωσης
  - Για κάθε έργο j έχουμε
    - home(j) είναι ο κόμβος που ξεκίνησε
    - recovered(j) είναι η ανακτηθείσα πίστωση
  - Για κάθε κόμβο S έχουμε
    - done_S(j) είναι το άθροισμα ανακτηθεισών πιστώσεων
    - Μόνο για τα καθήκοντα που ολοκληρώθηκαν

Ανάκτηση πίστωσης (5 από 7)

- Λειτουργία αλγορίθμου
  - Δημιουργία έργου j με καθήκον t στον H
    - home(j)=H
    - job(t)=j
    - credit(t)=1
    - recovered(j)=0
    - $done_H(j)=0$

Ανάκτηση πίστωσης (6 από 7)

- Λειτουργία αλγορίθμου
  - Το t δημιουργεί νέο καθήκον t’
    - credit(t)=credit(t’)=credit(t)/2
  - Ολοκλήρωση του t στον κόμβο S
    - $done_S(j)=done_S(j)+credit(t)$
  - Ολοκλήρωση όλων των καθηκόντων στον S
    - Ο S στέλνει το $done_S(j)$ στον home(j)
    - Ο H θέτει $recovered(j)=recovered(j)+done_S(j)$
    - Αν recovered(j)=1, έχουμε τερματισμό

Ανάκτηση πίστωσης (7 από 7)

- Ιδιότητες αλγορίθμου
  - Απλός και κατανοητός αλγόριθμος
    - Εύκολη απόδειξη ορθότητας
  - Ένα μήνυμα ανά κατάσταση αδράνειας κόμβου
    - Μπορούμε να περιμένουμε πριν στείλουμε
    - Μήπως εμφανιστούν νέα μηνύματα
  - Δυαδικοί αριθμοί αυθαίρετης ακρίβειας
    - Αλλιώς περιορίζεται η δημιουργία καθηκόντων

Dijkstra-Scholten (1 από 3)

- Αλγόριθμος των Dijkstra και Scholten
  - Οι κόμβοι είναι ενεργοί ή παθητικοί
    - Όπως στον αλγόριθμο του Dijkstra
  - Αρχικά όλοι οι κόμβοι είναι παθητικοί
  - Ο συντονιστής S είναι ενεργητικός
    - Οι υπολογισμοί ξεκινούν από τον S

Dijkstra-Scholten (2 από 3)

- Αλγόριθμος των Dijkstra και Scholten
  - Κατασκευή αντεστραμμένου δένδρου
    - Ρίζα ο συντονιστής S
    - Κάθε κλαδί δείχνει εκκρεμή υπολογισμό
  - Ο p δείχνει στον πατέρα του (parent(p))
    - Ένας κόμβος χωρίς πατέρα είναι ελεύθερος
  - Ο p μετράει τα παιδιά του (children(p))
    - Ένας κόμβος χωρίς παιδιά είναι φύλλο

Dijkstra-Scholten (3 από 3)

– Όταν ο p<>S με parent(p)=null λάβει μήνυμα από p’
  - Εισάγεται στο δένδρο η ακμή (p,p’)
  - Ο p ενημερώνει τον p’
  - children(p’)++
– Όταν ο p γίνει παθητικός και children(p)=0
  - Ο p ενημερώνει τον p’=parent(p)
  - children(p’)--
  - Αφαιρούνται όλες οι εξερχόμενες ακμές του p
– Όταν ο S γίνει παθητικό φύλλο έχουμε τερματισμό

Αλγόριθμος στιγμιοτύπων (1 από 2)

- Ανίχνευση τερματισμού με στιγμιότυπα
  - Προσαρμογή αλγορίθμου στιγμιοτύπων
  - Έστω ότι η p λαμβάνει μήνυμα στιγμιοτύπου
  - Αν το έλαβε από την p’
    - Λέμε ότι η p’ είναι ο προκάτοχος της p
    - Λέμε ότι η p είναι ο διάδοχος της p’

Αλγόριθμος στιγμιοτύπων (2 από 2)

- Όταν η p ολοκληρώσει τη δουλειά της
  - Αν έχει λάβει <done> από τους διαδόχους
  - και δεν έχει λάβει μήνυμα όσο κατέγραφε κανάλια
    - Στέλνει <done> στην προκάτοχό της
  - Διαφορετικά
    - Στέλνει <continue> στην προκάτοχό της
- Αν ο συντονιστής λάβει <done> από προκατόχους
  - Ο υπολογισμός έχει τελειώσει
- Αν ο συντονιστής λάβει κάποιο <continue>
  - Ξεκινάει άλλη λήψη στιγμιοτύπου

Τέλος Ενότητας # 3

Μάθημα: Κατανεμημένα Συστήματα: Θεωρία και Προγραμματισμός,
Ενότητα # 3: Καθολικά κατηγορήματα

Διδάσκων: Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ