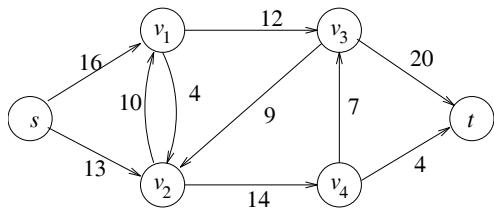


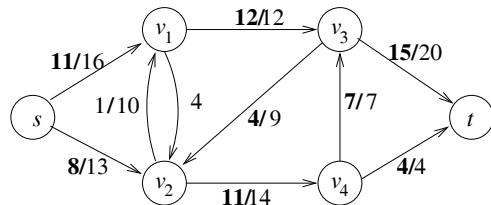
## Μεγιστη ροή (Maximum Flow)

- Δίνεται ένα δίκτυο ροής, μία κορυφή κόμβος -πηγή  $s$ , και μία κορυφή -προορισμός  $t$ . Θέλουμε να υπολογίσουμε την μέγιστη ποσότητα από κάποιο εμπόρευμα/αγαθό που μπορεί να ρεύσει από την πηγή  $s$  προς τον προορισμό  $t$  χωρίς να γίνεται υπέρβαση της χωρητικότητας -ροής (*flow capacity*) των ακμών του δικτύου.

### Παράδειγμα



Ένα δίκτυο ροής.



Μια ροή  $f$  με  $|f| = 19$ .

### Εφαρμογές

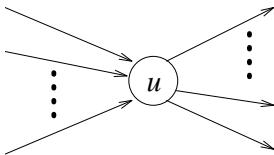
- Μπορούμε να μοντελοποιήσουμε:
  - Την ροή υγρών μέσα σε αγωγούς.
  - Την διαδρομή των συσκευών / ανταλλακτικών σε μία γραμμή παραγωγής (assembly line).
  - Δίκτυα ηλεκτρισμού.
  - Δίκτυα επικοινωνίας / μεταφορών.
- Πολλά άλλα προβλήματα τα οποία, με μία πρώτη μάτια, φαίνονται άσχετα με το πρόβλημα μέγιστης ροής μπορεί να αναχθούν σε αυτό (π.χ., μέγιστο ταίριασμα σε διμερές γράφημα (*maximal bipartite matching*)).

Δίκτυα ροής και ροές

- Ένα δίκτυο ροής  $G = (V, E)$  είναι ένα κατευθυνόμενο γράφημα στο οποίο η κάθε ακμή  $(u, v) \in E$  έχει μία μη-αρνητική χωρητικότητα (capacity)  $c(u, v) \geq 0$ . Εάν  $(u, v) \notin E$  τότε  $c(u, v) = 0$ .
- Υπάρχουν 2 διακεκριμένες κορυφές: Μία πηγή (source)  $s$  και ένας προορισμός (sink)  $t$ . Υποθέτουμε ότι κάθε κορυφή  $v \in V - \{s, t\}$  ανήκει σε ένα μονοπάτι από την  $s$  προς την  $t$ .
  $\implies |E| \geq |V| - 1$  (το γράφημα είναι συνδεδεμένο (connected)).
- Μία ροή στο  $G$  είναι μία συνάρτηση πραγματικών τιμών  $f : V \times V \longrightarrow \mathbb{R}$  η οποία ικανοποιεί τις παρακάτω συνθήκες:
  - Περιορισμός χωρητικότητας:  $\forall u, v \in V, f(u, v) \leq c(u, v)$ . (*Capacity constraint*)
  - Περιορισμός ασυμμετρίας:  $\forall u, v \in V, f(u, v) = -f(v, u)$ . (*Skew constraint*)
  - Διατήρηση ροής:  $\forall u \in V - \{s, t\}, \sum_{v \in V} f(u, v) = 0$ . (*Flow conservation*)

Διατήρηση της ροής

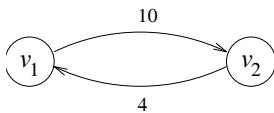
- Για κάθε  $u \in V - \{s, t\}$  απαιτούμε  $\sum_{v \in V} f(u, v) = 0$



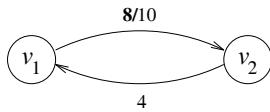
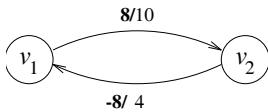
- Η ροή που εισέρχεται σε μία κορυφή είναι ίση με την ροή που εξέρχεται από αυτή.
- Η ροή για όλες τις εξερχόμενες ακμές είναι μη-αρνητική ενώ η ροή για όλες τις εισερχόμενες ακμές είναι μη-θετική.
- Η ποσότητα  $f(u, v)$  ονομάζεται ροή από την κορυφή  $u$  προς την κορυφή  $v$ .
- Η τιμή της ροής  $f$  ορίζεται ως:

$$|f| = \sum_{v \in V} f(s, v) \quad \left[ = \sum_{v \in V} f(v, t) \right]$$

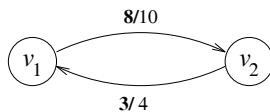
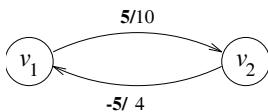
Πως να διαχειριστούμε τις ροές - «Ακύρωση»



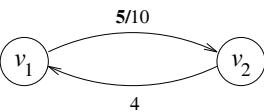
- Αποστολή 8 μονάδων από την  $v_1 \rightarrow v_2$ .



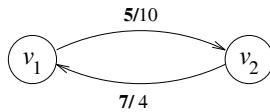
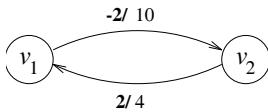
- Αποστολή 3 μονάδων από την  $v_2 \rightarrow v_1$ .



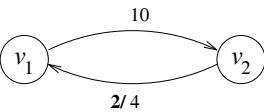
cancellation  $\Rightarrow$



- Αποστολή 7 μονάδων από την  $v_2 \rightarrow v_1$ .



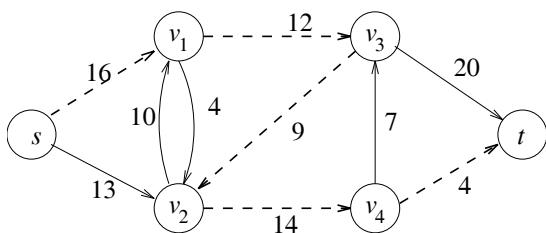
cancellation  $\Rightarrow$



**Η μέθοδος των Ford-Fulkerson**

**Ορισμός** Ένα επαυξάνων μονοπάτι (*augmenting path*) είναι ένα μονοπάτι από την πηγή  $s$  προς τον προορισμό  $t$  κατά μήκος του οποίου μπορούμε να προωθήσουμε επιπρόσθετη ροή, και επομένως, να αυξήσουμε την ροή κατά μήκος του μονοπατιού.

**Παράδειγμα**



*Ford\_Fulkerson\_Method( $G, s, t$ )*

1. Initialize flow  $f$  to 0
2. **while** there exists an augmenting path  $p$ 
  - do** augment flow  $f$  along  $p$
4. **return**  $f$

### Τυπολειμματικά δίκτυα (Residual networks)

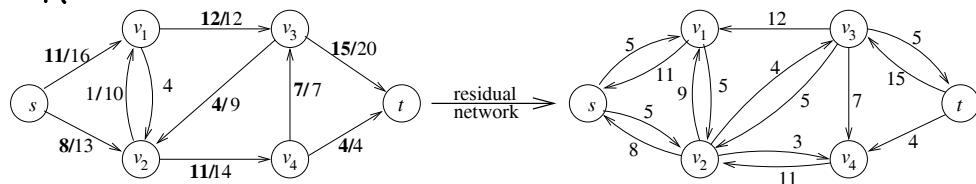
- Έστω  $f$  μία ροή στο  $G$  και ένα ζεύγος κορυφών  $u, v \in V$ . Η ποσότητα της επιπρόσθετης ροής που μπορεί να προωθηθεί από την  $u$  προς την  $v$  χωρίς να παραβιαστεί η χωρητικότητα  $c(u, v)$  ονομάζεται υπολείπουσα χωρητικότητα (residual capacity)  $c_f(u, v)$  της  $(u, v)$ .

$$c_f(u, v) = c(u, v) - f(u, v)$$

Σημείωση:  $c_f(u, v) \geq 0$  για κάθε  $u, v \in V$

- Με δεδομένο ένα δίκτυο ροής  $G = (V, E)$  και μία ροή  $f$ , το υπολειμματικό δίκτυο του  $G$  το οποίο προκύπτει από την  $f$  είναι το  $G_f = (V, E_f)$  όπου  $E_f = \{(u, v) : c_f(u, v) > 0\}$ .

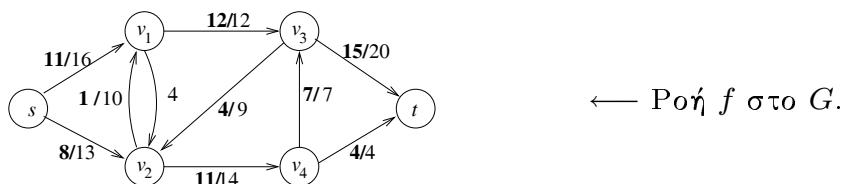
#### Παράδειγμα



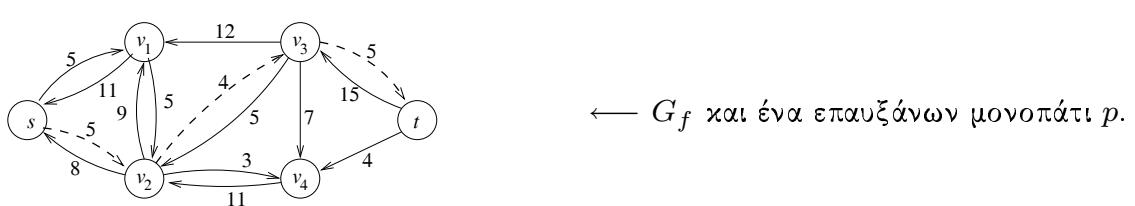
Παρατήρηση  $|E_f| \leq 2|E|$

Διότι η ακμή  $(u, v)$  ανήκει στο  $E_f$  εάν τουλάχιστον μία από τις  $(u, v)$  ή  $(v, u)$  ανήκει στο  $E$ .

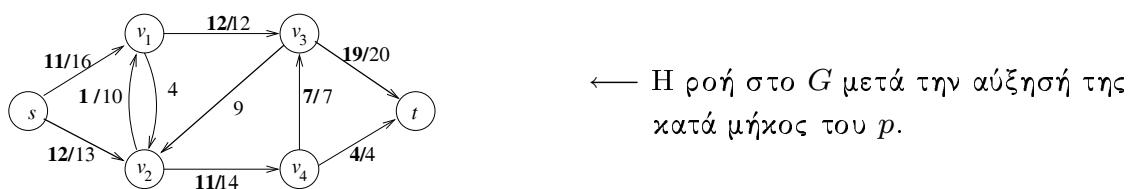
#### Παράδειγμα (Μία επανάληψη της Ford-Fulkerson Method)



← Ροή  $f$  στο  $G$ .



←  $G_f$  και ένα επαυξάνων μονοπάτι  $p$ .



← Η ροή στο  $G$  μετά την αύξησή της κατά μήκος του  $p$ .

- Έστω ένα επαυξάνων μονοπάτι  $p$ . Η μέγιστη ποσότητα ροής που μπορούμε να προωθήσουμε κατά μήκος των ακμών του  $p$  ονομάζεται *υπολείπουσα χωρητικότητα* (*residual capacity*) του  $p$ .

$$c_f(p) = \min\{c_f(u, v) : (u, v) \in p\}$$

*Ford-Fulkerson( $G, s, t$ )*

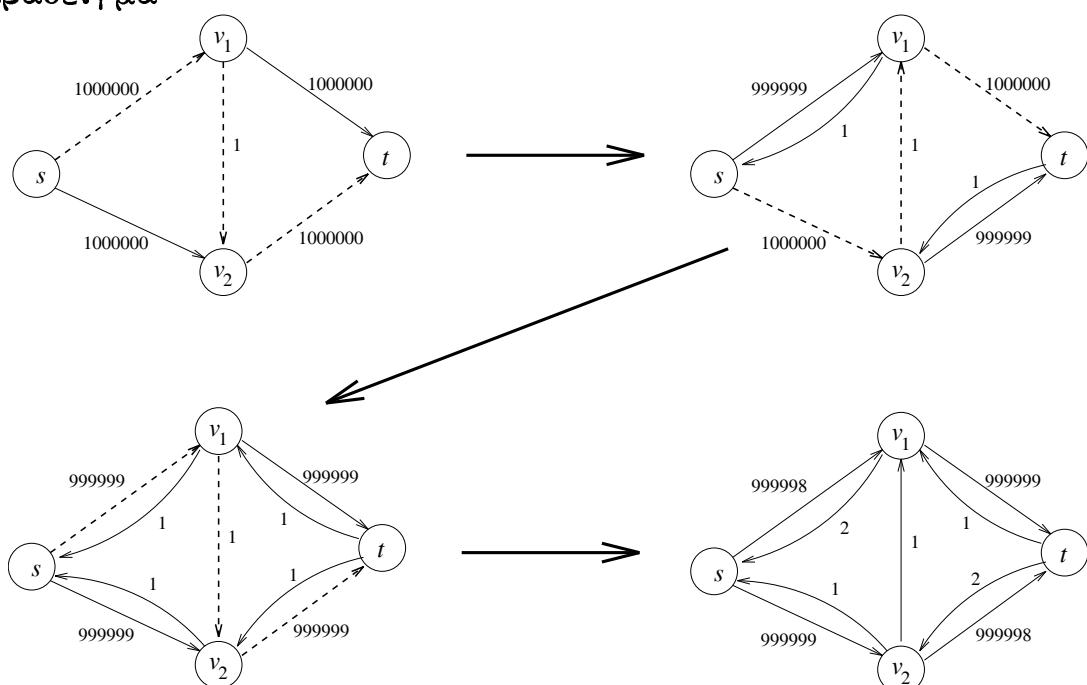
1. **for** each edge  $(u, v) \in E$ 
  2. **do**  $f[u, v] = 0$
  3.  $f[v, u] = 0$
4. **while** there exists a path  $p$  from  $s$  to  $t$  in  $G_f$ 
  5. **do**  $c_f(p) = \min\{c_f(u, v) : (u, v) \in p\}$
  6. **for** each edge  $(u, v)$  in  $p$ 
    7. **do**  $f[u, v] = f[u, v] + c_f(p)$
    8.  $f[v, u] = -f[u, v]$

### Χρονική πολυπλοκότητα

- Εξαρτάται από τον τρόπο με τον οποίο επιλέγουμε τα επαυξάνοντα μονοπάτια.
- Ο αλγόριθμος μπορεί να μην τερματίσει.
- Εάν οι χωρητικότητες είναι **ακέραιες** ο αλγόριθμος πάντοτε τερματίζει.

- Ένα άνω όριο για τον απαιτούμενο χρόνο εκτέλεσης της μεθόδου *Ford-Fulkerson()* για ακέραιες χωρητικότητας είναι  $O(E|f^*|)$  όπου  $f^*$  είναι η μέγιστη ροή που επιστρέφει ο αλγόριθμος.

### Παράδειγμα



- Ένα επαυξάνων μονοπάτι μπορεί να υπολογιστεί σε  $O(E)$  χρόνο είτε με κατά βάθος διαπέραση (*depth first search*) ή με κατά πλάτος διαπέραση (*breadth first search*).
- Η απόδειξη ορθότητας της μεθόδου Ford-Fulkerson προκύπτει από το θεώρημα:

**Θεώρημα** Εάν  $f$  είναι μία ροή στο δίκτυο ροής  $G = (V, E)$  με πηγή  $s$  και προορισμό  $t$ , τότε οι παρακάτω δύο προτάσεις είναι ισοδύναμες:

1. Η  $f$  είναι μία μέγιστη ροή στο  $G$ .
2. Το υπολειμματικό δίκτυο  $G_f$  δεν περιέχει επαυξάνοντα μονοπάτια.

### O Edmonds-Karp αλγόριθμος

*Edmonds-Karp( $G, s, t$ )*

- Compute the flow as in *Ford-Fulkerson()*.
- Use *breadth first search* to compute the augmenting paths.

**Θεώρημα** Ο *Edmonds-Karp* αλγόριθμος πραγματοποιεί  $O(VE)$  επαυξήσεις ροής.

**Χρονική πολυπλοκότητα:**  $O(VE^2)$   
(Συνολικά  $O(VE)$  κλήσεις για κατά πλάτος διαπέραση.)

**Σημείωση** Οι χωρητικότητες πρέπει να είναι ακέραιοι αριθμοί.

## ΥΠΟΛΟΓΙΣΙΜΟΤΗΤΑ (Computability)

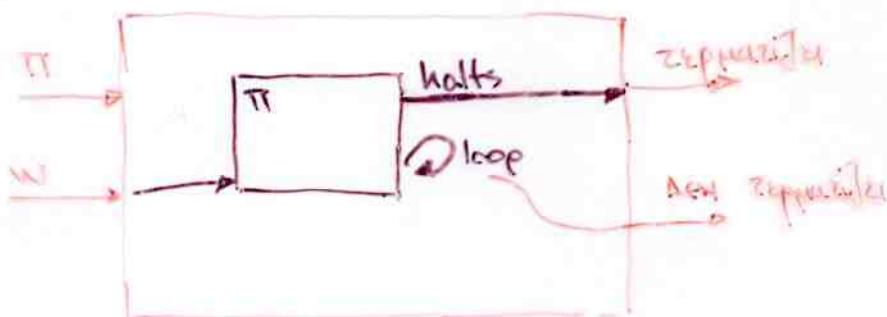
Αντικείμενο της θεωρίας υπολογιστών:

"Είναι μια πρόσβατη υπολογιστής ή όχι;"

"Είναι μια πρόσβατη αποφεύγουσα ή όχι;"

Παράδειγμα: To "halting problem" (HP)

Μπορούμε να φτιάξουμε ένα πρόγραμμα το οποίο, σας τας δίνουμε ως εισόδο μια πρόσβατη η οποία αποφεύγει την πρόσβαση  $w$  πας λίγη επί τη "πι ζερνάζα με εισόδο  $w$ ";



Επειδή: To halting problem είναι μια αποφεύγουσα

Παράδειγμα: Post-correspondence problem (PCP)

- Ντόμινο  $\begin{bmatrix} \text{string-1} \\ \text{string-2} \end{bmatrix}$   $\begin{bmatrix} a \\ ab \end{bmatrix}$
- Σύγχρονο οπιο Ντόμινο:
 
$$\left\{ \begin{bmatrix} b \\ ca \end{bmatrix}, \begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} ca \\ a \end{bmatrix}, \begin{bmatrix} abc \\ c \end{bmatrix} \right\}$$
- Αντορθία - ταίριασμα (match): δύο από ντόμινο τίτανα μέσω ο "αριθμητικής" ιδέας με τον "παραφερεντίου"
- PCP: Είναι μια διδούμενη σύγχρονη οπιο ντόμινο μια αντορθία - ταίριασμα?

ΘΕΩΡΗΜΑ: Το Post-correspondence problem είναι απομφίνο.

Αντικείμενο της Θεωρίας Πλούτοποντώσεως:

- "Εάν δοθεί ου μια συνάρτηση  $f$  εναντίου  $\pi_1$  (comparable), ποιό θα γίνεται η  $\pi_2$  αγάπη (resources) που χρειάζονται για την  $\pi_1$  συνάρτηση της;
- Μια συνάρτηση  $f$  δέχεται  $\pi_1$  (comparable) και  $\pi_2$  υπάρχει ένα πρόχρημα που αποδεικνύει την τιμή της για κάθε αριθμό.

**ΘΕΩΡΗΜΑ:** Υπάρχουν μια  $\pi_1$  αποδογετές συνάρτησης.

Αποδείξη: Από λήμμα 1 & λήμμα -2.

**ΛΗΜΜΑ -1:** Υπέρχου απόρα μεν, αλλά αριθμητικά  
Γιαφορετικά πρόγραμμα τα οποία (χρησιμο-  
ποίουσας μεδιατοποίηση) μπορεί να απαριθμηθούν  
μηχανικά (effectively enumerable).

Άποδημη:

- $\Sigma = \{q_1, q_2, \dots, q_m\}$  το αλφάριτμο των γλωσσών προγραμμάτων.
- $\Sigma_n$ : συνόλο συμβολοσειρών του  $\Sigma$  μήκους  $n$
- $\Sigma^* = \bigcup_{n=0}^{\infty} \Sigma_n$
- Κάθε πρόγραμμα είναι δυοιχίο του  $\Sigma^*$

$$\Sigma_0 = \{\epsilon\}$$

$$\Sigma_1 = \{q_1, q_2, \dots, q_m\}$$

$$\Sigma_2 = \{q_1 q_1, q_1 q_2, \dots, q_1 q_m, \dots, q_m q_m\}$$

⋮

⇒ Με χρησιμοποιώντας compiler για ελέγχο ορθότητας μπορούμε να παταχουν άσυρτες μηχανικά αριθμητικά των συντακτικά ορθών προγραμμάτων

□

ΛΗΜΜΑ - 2: Υπάρχουν μη-αριθμητικές οπωρες διαφορετικές συναρτήσεις.

Άσκηση:

- Εσω το σύνολο των ολικών συναρτήσεων  $\phi: \mathbb{N} \rightarrow \mathbb{N}$  (ολικές συναρτήσεις: ορίζονται για κάθε  $x \in \mathbb{N}$ )
- Εσω  $\phi_0, \phi_1, \dots$  μια αριθμοί τους
- Ορίζομε την  $f: f(x) = \phi_x(x) + 1$

$$f(0) = \phi_0(0) + 1$$

$$f(1) = \phi_1(1) + 1$$

⋮

→ Η  $f$  εναι ολική

→ Εσω οι  $n$   $f$  εχει διατη για συν αριθμού δι.   
 $f = \phi_y$

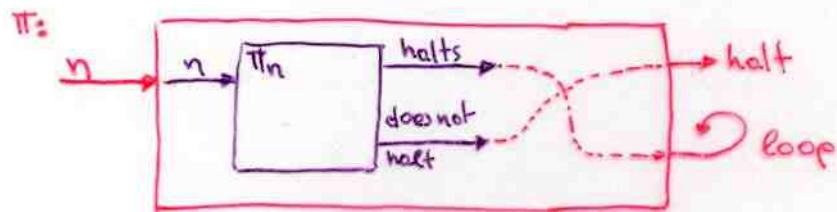
- $f = \phi_y \rightarrow \phi_y(y) = \phi_y(y) + 1 \rightarrow \text{ΑΤΩΤΟ}$

□

ΘΕΩΡΗΜΑ: To halting problem δεν είναι αποφεύγιμο

Άπορηση: Εσώρουχη είναι αποφεύγιμη:

- Εσώρουχη  $\Pi_0, \Pi_1, \dots$  μια μηχανήσαν αριθμούς όλων των προγραμμάτων.
- Φυλάχθη πρόγραμμα  $\Pi$ :



- Εσώρουχη  $\Pi \equiv \Pi_i$  (είναι αριθμούς)

$\Pi(n)$  σταματάει  $\Leftrightarrow \Pi_n(n)$  δεν σταματάει

$\Pi_i(n) \Rightarrow \Leftrightarrow \Pi_n(n) \Rightarrow$

$\Pi_i(i)$  σταματάει  $\Leftrightarrow \Pi_i(i)$  δεν σταματάει.  $\square$

ΟΡΙΣΜΟΣ: Ένα σύνολο  $S$  λέγεται απομρίσιμο εάν και μόνο εάν υπάρχει αλγορίθμος που σχηματίζει (ή μια υπολογιστική μηχανή που δίνει εξόδο "ναι" για κάθε εισόδο  $a \in S$  και "όχι" για κάθε εισόδο  $a \notin S$ )

ΟΡΙΣΜΟΣ: Ένα σύνολο  $S$  λέγεται λεπτομερέσιμο (listable, effectively generated, enumerable) εάν υπάρχει μια γεννιτρία διαδικασία ή μια μηχανή που λεπτομερά το  $S$

ΔΙΟΤΗΤΕΣ:

- ▷  $S$  απομρίσιμο  $\Rightarrow \bar{S}$  απομρίσιμο
- ▷  $S$  απομρίσιμο  $\Rightarrow S$  λεπτομερέσιμο
- ▷  $S, \bar{S}$  λεπτομερέσιμο  $\Rightarrow S$  απομρίσιμο
- ▷  $S$  λεπτομερέσιμο και  $\left. \begin{array}{c} \text{γνωστή} \\ \text{αύξοσα διαδικασία} \end{array} \right\} \Rightarrow S$  απομρίσιμο

## Θέση (thesis) των Church-Turing

[Όταν τα χωράντα ναι "όχυρα" μοντέλα των εννοιών υπολογιστών είναι μηχανικά ισοδύναμα

[Δοδέκατος ενός αλγορίθμου σε ένα μοντέλο για μια συναρτηση  $f$ , το οποίο μηχανικά μεταφέρει να πετασεμένων αλγορίθμων σε αλλο μοντέλο για την ίδια συναρτηση]

{Η εννοια των  
αλγορίθμων} αναι ισοδύναμη  
με {της εννοια των  
αλγορίθμου για  
μηχανές Turing}

Ti είναι είναι "Αλγορίθμος";

- Συνταξή
- Διαδικασία
- Ένα πρόγραμμα υπολογιστή
- ...

- Σεν είχε ορίσει με αρίθμα πριν του 20 αιώνα.

To 10<sup>ο</sup> πρόβλημα του Hilbert

- To 1900, o David Hilbert σε μια διάσκηψη διάλεξε του
  - ▷ ήριξε 23 μαθηματικά προβλήματα
  - ▷ προηγήσας για του 20<sup>ο</sup> αιώνα
  - ▷ το 10<sup>ο</sup> πρόβλημα αφοράτο των αλγορίθμων.

Πολυωνυμα:

term: γινόμενο μεταβλητών και συνδρέσιν αντελεκτών  
 $6x^3yz^2$

polynomial: οδραιορά από terms.

$$6x^3yz^2 + 3xy^2 - x^3 - 10$$

ρίζα: τιμές των μεταβλητών οι οποίες η άριθμη του πολυωνυμού εναι λόγω με μηδείς (0)

$$x=5 \quad y=3 \quad z=0$$

To πρόβλημα: Να βρεθεί ενας αλγορίθμος που  
επέχει εάν ενα πολυώνυμο έχει  
κυρτήσεις στις πλευρές.

O Hilbert είπε:

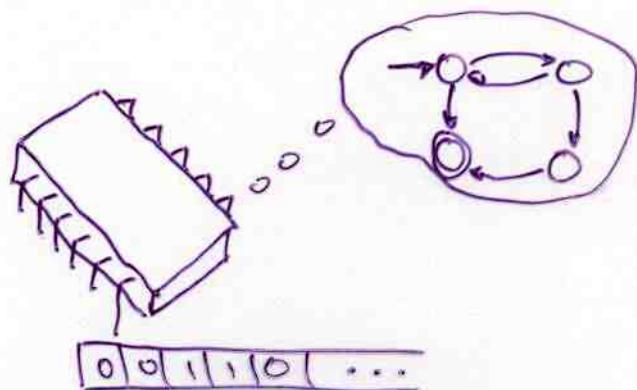
"...to device a process according to which,  
it can be determined by a finite number  
of operations."

Παραγμένος ου:

- O Hilbert ζητούν να μάζανεσει ενας  
αλγορίθμος
- Οποδέται ου ότι ο αλγορίθμος οπάρχει  
  
 $\Rightarrow$  Σημερα γνωρίζουμε ου δεν οπάρχει τέτοιος  
αλγορίθμος.

[Yuri Matijasevic , 1970]

## ΜΗΧΑΝΕΣ TURING



- Χρησιμοποιεί άπερι ταινία ως μνήμη
- Η ταινία αρχικά περιέχει τις συρβοδοσια - είναι οι αρχικούς ρυθμούς από περιορισμένη αριθμό
- Η ταινία χρησιμοποιείται ως "πρόχειρη μνήμη"
- Η μηχανή έχει accept και reject παραστάσεις
- Μπορεί να "πρέχει" για πάντα.

## ΠΑΡΑΔΕΙΓΜΑ

- Mia TM που ελέγχει εάν ο εισόδος ανήκει στη γλώσσα  $\{w\#w \mid w \in \{a,b\}^*\}$

aabbab#abbab      Xbab#abbab ...      Xbab#abbab

↑  
a

↓  
OK

- Η μετάληψη βρίσκεται στον αριστερότερο σύμβολο
- Εμφανίζεται το σύμβολο (#) από παρότρυνση και το διαχειρίζεται X
- Φάνηξε στα δεξιά: REJECT ταυτόχρονα με την πρώτη #
- Οταν βρει #, πηγαίνει μια διεύθυνση δεξιά
- εάν τα σύμβολα δεν ταυτίζονται reject

Xbab#Xbab      XXbab#Xbab      XXbab#Xbab

↑  
b

↓  
OK

- διαχειρίζεται το δύρτιο, γράψει X
- φάνηξε αριστερά, προσπέρασε το #, πηγαίνει εώς το X
- πηγαίνει μια διεύθυνση δεξιά
- διακατεί το σύμβολο (εις παρότρυνση), το ανικανάριστο με X
- φάνηξε δεξιά, μετά το # εώς το τελευταίο X ...

XXXXX#XXXXX      XXXXX#XXXXX

↑  
b

↓  
OK

- πηγαίνει μια διεύθυνση δεξιά
- εάν βρει a ή b, REJECT
- εάν βρει KENO, ACCEPT.

## ΟΡΙΣΜΟΣ (ΜΗΧΑΝΗ TURING)

- Βασίζεται στην επιφάνεια μεταβολών

$$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$$

$$\delta(q, a) = (r, b, L) \text{ σημαίνει:}$$

- Η TM στην κατάσταση  $q$  με την υπόθεση  $a$  αντικαθιστά το  $a$  με  $b$
- Η κατάσταση  $r$  γίνεται  $r$
- Η υπόθεση  $a$  αποτελεί προσέργα

Mia Μηχανή Turing (TM) θα μια 7-άδα  
 $(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$  έπου:

- $Q$  θα πεπερασμένο δύνοτο καταστάσεων
- $\Sigma$  η αλφάριτσα - ασύδου, δεν περιέχει το κενό αριθμό  $\sqcup$
- $\Gamma$  η αλφάριτσα - ταινίας, περιέχει τη  $\Sigma$  και το  $\sqcup$
- $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$  η συνάρτηση μεταβολών
- $q_0 \in Q$  η αρχική κατάσταση
- $q_a \in Q$  η accept-κατάσταση
- $q_r \in Q$  η reject-κατάσταση.

- Μια TM  $M$  αποδέχεται (accepts) την εισόδο  $x$  ανν σαφαρά σε παραγόμενη απόδοξή  $q_a$  για το  $x$ .
- Η γλώσσα  $L_M$  του αναγνωρίζει/αποδέχεται η TM  $M$  είναι:

$$L_M = \{ x \in \Sigma^* : M \text{ αποδέχεται } x \}$$

- Μια TM  $M$  αποφασίζει (decides) την εισόδο  $x$  ανν σαφαρά σε μια παραγόμενη απόδοξή  $q_d$  ή απόφερει  $q_r$ .
- Η γλώσσα  $L_M$  γιατιν οποια αποφέρεται/αποφασίζει (decides) η TM  $M$  είναι

$$L_M = \{ x \in \Sigma^* : M \text{ αποφασίζει } x \}$$

ΠΑΡΑΛΛΑΓΕΣ *zus Münchens Turing*

1. TM οπου η κεφαλή μπορεί να μείνει ανίχνη

$$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, S\}$$

→ Είναι ιδεογνωμόνια με την TM

Αντικατάσταση *zus*  $\delta(q, a) = (q', b, S)$  με

$$\delta(q, a) = (q_1, b, R) \quad q_1 \text{ είναι } \underline{\text{νέα}} \text{ σημαστόρευμα}$$

$$\delta(q_1, x) = (q', x, L) \quad \text{για κάθε } x \in \Gamma$$

Η έννοια *zus* προσομοιώσους είναι διμερής

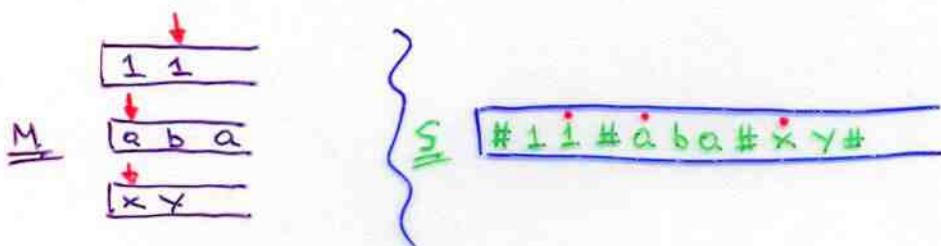
## 2. TM's με πολλαπλές ταινίες

- Εάντε ταινία έχει την δυνατότητα να κάνει πολλές πράξεις συγχρόνως.
- Αρχικά η εισόδος είναι συμπλήρωμα  $\#$  και οι απόδοσηις ταινίες είναι μεταξύ.

Ενιαίρουν μετάβασης:  $Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R\}^k$

$$\delta(q_i, q_1, q_2, \dots, q_k) = (q_j, b_1, b_2, \dots, b_k, L, R, \dots, L)$$

### ΠΡΟΣΩΝΟΙΩΣΗ



- Η  $M$  έχει  $k$  ταινίες
- Προσδιορίζεται η  $k$  ταινίες σε μια ταινία στην αποδίνει τη διεύθυνση των δεξιοτήτων τους.
- Η  $\Sigma$  "συμπληρώνει" τη δύση των μεταβολών

Με εισόδο  $w = w_1 w_2 \dots w_n$ , η  $\Sigma$ :

→ έχει ταινία:  $\# w_1 w_2 \dots w_n \# \# \# \dots \# \#$

→ Φαχτεί από το  $i^{\text{th}}$  # εως το  $(k+1)^{\text{th}}$  # για να διαλέξει τα σύμβολα στις ταινίες

→ Γιασχήματα και δραστικές για τα σύμβολα

→ Εάν χρειάζεται  $\Sigma$  μετατοπίζει προς τη δεξιά τη περιεχουμένη των για να δικτυώνεται μετώπιστο

### 3. Μη-Νετερμηνούσιες ΤΜ (NTM)

Συνάριστη μετάβασης  $\delta: Q \times \Gamma \rightarrow P(Q \times \Gamma \times \{L, R\})$

- Ο "υπολογισμός" είναι ενα δένδρο
- "accepts" είναι ενα οποιοδήποτε "ηλαί" accepts

Βασική ιδία προσδοκίων:

→ Ηεγαροπι της NTM M σε νετερμηνούσιες ΤΜ D

- Η D δουμένει ότι τη πίστα υπολογισμός
- Εάν n D εντοπίσει accept-καταστάση, accept
- Εάν όταν τη ηλαίδη αδύνατη σε reject-καταστάση reject
- Εάν όταν τη ηλαίδη αδύνατη σε reject-καταστάση n loop, loop.

→ Ο υπολογισμός της N είναι ενα δένδρο.

→ καθη ηλαίδη αντιστοιχεί με ένα κάθισμα στην ηλαίδη

→ καθη πόρος είναι ενα "configuration" της N

→ n ηλαίδη είναι τη αρχικό configuration

→ χρησι BFS

## Προσομοιώση ...

### D έχει 3 ταινίες

ταινία εγκόδου

11010000 ...

ταινία προσομοιώσης

ab101cd ...

ταινία- διεύθυνσης

122314 ....

- Η ταινία - εγκόδου δεν μεταβιλεῖται ποτέ
- Η ταινία - προσομοιώσης είναι αναγράφω της ταινίας της  $N$
- Η ταινία- διεύθυνσεων αποδικεί της "δέση" της D στο "διάδρο υπολογισμού" της N

### H ταινία - διεύθυνσης ..

→ Κάθε κορίτσος του διαδρού εχει το τρόπο b παιδί

→ κάθε κορίτσος εχει μια διεύθυνση που είναι μια ζίζη με αριθμητικό το  $\Sigma_b = \{1, 2, \dots, b\}$

→ Τια να πάρει στούς κορίτσια 1-2-3

→ ξενιάσει από την είρη

→ αυτούσιει 1<sup>η</sup> επιλογή

→ >> 2<sup>η</sup> >>

→ >> 3<sup>η</sup> >>

→ Αγνώστης επιλογής του δως έχουν νόημα

## Η προσομοίωση ...

- Αρχικά, η ταινια-εισόδου περιέχει το **N**, αι άλλες είναι άδυτες.
- Αντιγράφει την ταινια εισόδου στην ταινια προσομοίωσης
- Στην ταινια-προσομοίωσης, προσομοίωνται τα **N** με εισόδο **w**, για να ιλαρίσει υπολογισμού. Για κάθε βίρια διαμορφώνεται την ταινια διεύθυνση.
- ACCEPT έχει φάσει σε "παραστασή" απόδοχης
- Τηγανει στο επόμενο βίρια έστιν
  - σύμβολα της ταινιας διεύθυνσης **Ελαζίλινης**
  - μη εγκριτική επίδραση
  - έχει φάσει σε "παραστασή" reject.
- Αντικαταστάτε ~~την~~<sup>την</sup> ταινια-διεύθυνσην με την δεξιοχεφιακή επόμενη της.

## ΘΕΩΡΙΑ ΠΟΛΥΠΛΟΚΟΤΗΤΑΣ

Η χρονική πολυπλοκότητα είναι αλγορίθμου που επιτελεί κάποια πρόβλημα είναι μια αύξουσα συναρτήση  $T(n)$  η οποία  $\nabla$  είναι τα μεγέθη των εισόδων.

$$T(n) = \max \{ \# \text{steps for input } x : |x| = n \}$$

- Το μεγέθη των εισόδων εξαρτάται από την αναπαρίσταση
- Δυαδικό / δυναδικό σύστημα. Οχι εναδικό.

Αποδοτικός αλγόριθμος (efficient)

Υπάρχει πολυώνυμο  $P$  που ισχύει για όλα:

$$\forall n \quad T(n) \leq P(n)$$

Δεν μας ευδιαφέρουν.

- το δυνατικότερο υπολογιστικό μοντέλο
- η ανδιαστρίβη
- ο βελτισμός του πολυώνυμου.

## Πρόβλημα Βελτιστοποίησης.

- Συγάριες λύση που βελτιστοποιεί μια αντικείμενη συναρτήση

$$\rightarrow \text{Συνάριστη εφιαλτική λύση}$$

$$\rightarrow \text{βελτιστη λύση}$$

## Πρόβλημα πλανόδιων ταξιδιών (Traveling Salesman problem)

- πόλεις  $C = \{c_1, c_2, \dots, c_n\}$
- αποστάσεις  $d(c_i, c_j) \in \mathbb{Z}^+ \wedge (c_i, c_j) \in C^2$

? διαδρομή που περιέχει όλα ταδε πόλη, επιστρέφει στην αρχική, έχει έλαχιστο μήκος

Μεταδεδούμενος  $\langle c_{\pi_1}, c_{\pi_2}, \dots, c_{\pi_n} \rangle$  του  $C$  :

$$\sum_{i=1}^{n-1} d(c_{\pi_i}, c_{\pi_{i+1}}) + d(c_{\pi_n}, c_{\pi_1}) \text{ είναι έλαχιστο.}$$

## Πρόβλημα απόφασις

- Επιδεχόνται απαντήσεων των μορφών  $\text{ΝΑΙ}$  ή  $\text{ΟΧΙ}$ .
- Εάν  $A$  το συνόλο των εισοδών  $\times$  για τις αποτελέσεις  
η απαντηση είναι  $\text{ΝΑΙ}$ , το πρόβλημα λογδυναμίζει  
και το  $"x \in A;"$

Πρόβλημα βελτιστοποίηση  $\rightarrow$  απόφασις.

"Για εισόδο  $X, n$  υπάρχει εφικτή λύση  $s$  για το  $x$  και  
 $c(s) > n ;"$

## ΜΕΓΙΣΤΗ ΚΛΙΚΑ (Clique)

- Γράφημα  $G(V, E)$
- Ποια η μεγιστηριακή;  $\leftarrow$  βελτιστοποίηση

- Γράφημα  $G(V, E)$
- αυθαίριστος  $n$
- Υπάρχει κλίκα μεγεθος  $\geq n \leftarrow$  απόφασις

- $\rightarrow$  Το πρόβλημα απόφασης ΔΕΝ ΕΙΝΑΙ ΤΙΠΟΔΙΚΟ ουδε το πρόβλημα βελτιστοποίησης
- $\rightarrow$  Το πρόβλημα βελτιστοποίησης είναι ζΩΝΑΧΙΣΤΟΝ το οποίο δεν  
και το πρόβλημα απόφασης.

O<sub>1</sub> αδιστις P ήτη NP

P (POLYNOMIAL): Η υλίση προβλημάτων που επιλύονται σε πολυαναρμό χρόνο από κάποιον μη-νημεροθυμητό άλγορίθμο.

(DTM: Deterministic Turing Machine)

NP (Non-deterministic Polynomial):

Η υλίση προβλημάτων που επιλύονται σε πολυαναρμό χρόνο από κάποιο μη-νημεροθυμητό άλγορίθμο (NDTM).

Θεωρούμε ότι έχει μη-νημεροθυμητός άλγορίθμος αποτελείται από 2 στάδια

- παρένει πια τους  $S$  για το σημερινό Ι
- επανδέιει ότι  $n$   $S$  είναι περάματι του.

ΕΥΡΕΣΗ ΣΤΟΙΧΕΙΟΥ:

choose  $a[i]$

verify: if  $a[i] = x$  then found  $\Theta(1)$

ΤΡΙΝΟΜΗΣ ΔΙΑΝΥΣΜΑΤΟΣ.

choose a permutation

verify:  $a[i] < a[i+1]$  for all  $i$   $\Theta(n)$

## SAT (SATisfiability)

- $x_i$  : προσωνικη μεταβλητη
- literals :  $x_i, \bar{x}_i$
- clauses : Συγκριτικοι αποτελεσματικοι λεγοντες αποτελεσματικη μεταβλητη  $lit_1 \vee lit_2 \vee \dots \vee lit_m$
- CNF (Conjunctive Normal Form) συγκριτικοι αποτελεσματικοι λεγοντες αποτελεσματικη μεταβλητη  $cl_1 \wedge cl_2 \wedge \dots \wedge cl_n$

SAT - προβλημα:

Εισοδος : μια Boolean εικόνα σε CNF

Εξοδος : Υπάρχει ανάλογη γραμμή που μακριποτεί την εικόνα;

choose : για αποφοιτησιανη εικόνα

verify : ελεγχει εάν η αποφοιτησιανη εικόνα εικόνα.

$NP = \{ L \mid \exists \text{ NDTM πρόγραμμα } M \text{ το οποίο σε καλυπτώμενο χρόνο αποδίχιται ότι } \gamma \text{ λέγεται } L \}$

Σύγχρονη θεωρία Π και NP

- $P \subseteq NP$
- $P = NP$  ???
- $NP \subseteq DEXPTIME$

## Αναγωγή ( $\leq$ , $\alpha$ )

$A \leq_m^P B$  : Α αναγράφει πολυωνυμικό μέτρο Karp σε ενα πρόβλημα B

$P_f$  : Ευναρπίστες του υπολογιζόνται σε πολυωνυμικό χρόνο

$A \leq_m^P B$  :  $\exists f \in P_f, \forall x (x \in A \Leftrightarrow f(x) \in B)$

### ΔΙΟΤΗΤΕΣ

1. Αναδιστροφή  $A \leq_m^P A$

2. Μικραρισμός

$$A \leq_m^P B \wedge B \leq_m^P C \Rightarrow A \leq_m^P C$$

$$3 \quad A \leq_m^P B \wedge B \leq_m^P A \Rightarrow A \equiv_m^P B$$

$$4. \quad A \leq_m^P B \wedge B \in P \Rightarrow A \in P$$

## Hardness - Completeness

- $C$ : ολα τα προβλήματα
  - $\forall B \in C : B \leq A$
- } To  $A$  ονομάζεται  $C$ -δύσκολο  
( $C$ -hard) ως τρόπος  $\leq$
- 
- $A \in C$
  - $A : C\text{-hard}$
- }  $A$  ονομάζεται  $C$ -πλήρες  
( $C$  complete)

Ενα πρόβλημα  $L$  είναι NP-complete ως τρόποι  $\leq_m^P$  αν:

$$(L \in NP) \wedge (\forall L' \in NP : L' \leq_m^P L)$$

An είναι NP-complete πρόβλημα αποδειχθεί ου κανείς στο P, τότε ολα τα προβλήματα του NP θα αποδειχθεί στο P

ΑΙΤΗΣΙΑ: An  $L_1 \leq_m^P L_2$

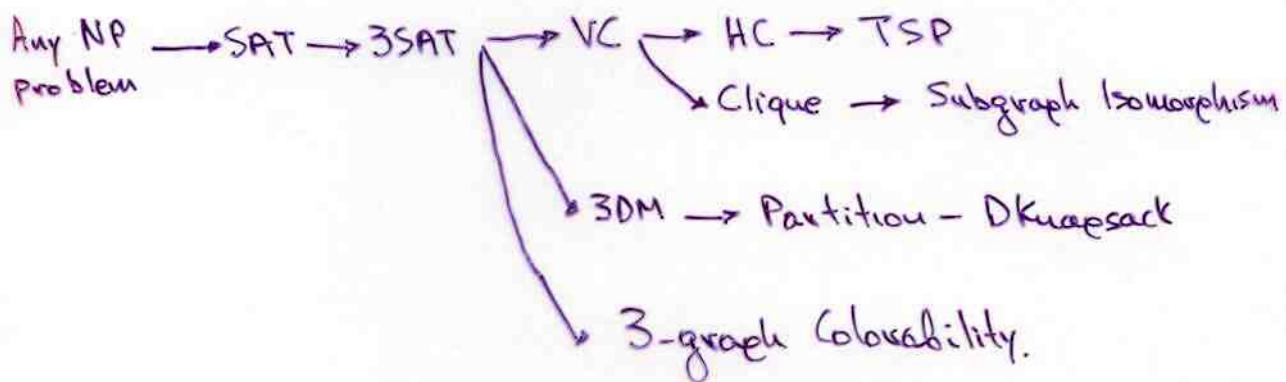
$L_2 : \text{NP-complete}$

$L_2 \in \text{NP}$

}  $L_2 \in \text{NP-complete}$

Θεωρήμα: Το πρόβλημα SAT είναι NP-complete.  
(Cook)

Αναγρέψτε:



Τια ως διδώ σαν ενο-πρόβλημα ή είναι NP-complete

1.  $\Pi \in \text{NP}$

2. Εστω  $\Pi' \in \text{NP-complete}$ .

Κατασκευής, ευάριστης f του μετασχηματισμού από  $\Pi'$  στο  $\Pi$ .

3. Ο μετασχηματισμός f γίνεται σε πολυώνυμο χρόνο

4.  $x \in \Pi' \Leftrightarrow f(x) \in \Pi$

## ΠΡΟΒΛΗΜΑΤΑ ΑΠΟΦΑΣΗΣ

### SAT: (Satisfiability)

**Δεδομένα:** Λογική επίφραση σε CNF μορφή

**Ερώτηση:** Είναι η λογική επίφραση πληντούμενη;

Υπάρχει αποτελεσματικός τρόπος για να διαβάσουμε την επίφραση ώστε η επίφραση να αποτελέσει True;

### 3-SAT

**Δεδομένα:** Boolean formula σε CNF; καθε clause έχει 3 literals

**Ερώτηση:** Είναι η Boolean formula πληντούμενη;

### VC (Vertex Cover)

**Δεδομένα:** Γράφημα  $G = (V, E)$

Θετικός αριθμός  $K \leq |V|$

**Ερώτηση:** Υπάρχει ένα vertex cover στο γράφημα του  $E$  μεγέθους  $\leq K$ ?

Διαδικ. υπορρέει  $V' \subseteq V$ :  $|V'| \leq K$  και

$\forall (u, v) \in E$ :  $u \in V'$  &  $v \in V'$  ?

## 3DM (3-dimensional Matching)

**Δεδομένα:** Σύνολο  $M \subseteq W \times X \times Y$  οπου  
 $|W| = |X| = |Y| = q$  και  $W, X, Y$  δίνε περιήγη τους

**Ερώτηση:** Περιέχει το  $M$  ενα ταυτόσημα (matching);  
 Δηλ., υπάρχει σύνολο  $M' \subseteq M$ :  $|M'| = q$  ιστοι  
 ώστε 2 συσταθμώσεις συσχετίζει του  $M'$  δω εξουν  
 μέρη που διανυσσαχθήσουν;

## Graph 3-Colorability

**Δεδομένα:** Γράφημα  $G = (V, E)$

**Ερώτηση:** Μπορεί να χρησιμοποιηθεί τας μόρφωσ μία  
 3 χρώματα επειδή ώστε 2 σπασμένη πολύ γεγονούσια  
 μόρφωσ να έχουν διαφορετικό χρώμα;  
 Γιατί:  $\exists f: V \rightarrow \{1, 2, 3\} : \forall (u, v) \in E: f(u) \neq f(v)$ .

## HC (Hamilton Circuit)

**Δεδομένα:** Γράφημα  $G = (V, E)$

**Ερώτηση:** Υπάρχει σου  $G$  ενας μέντος Hamilton? Δηλ.  
 υπάρχει μία διόρθωση των μόρφωσ του  $G$   
 $\langle v_1, v_2, \dots, v_n \rangle$ ,  $n = |V|$  η οποία ωστε

$(v_i, v_{i+1}) \in E$ ,  $1 \leq i \leq n-1$ , και  $(v_n, v_1) \in E$ . ?

## TSP (Traveling Salesman Problem)

Δεδομένα: Γράφημα  $G = (V, E)$  με θετικά βάρη.  
Θετικός αριθμός  $B$ .

Ερώτηση: Υπάρχει μήδουσα διαδρομή που περνά από όλους τους κατόπιν τους του  $G$

$$\langle V_{\pi(1)}, V_{\pi(2)}, \dots, V_{\pi(n)} \rangle \text{ είναι } w\sigma\sigma$$

$$\sum_{i=1}^{n-1} w(V_{\pi(i)}, V_{\pi(i+1)}) + w(V_{\pi(n)}, V_{\pi(1)}) \leq B ?$$

## CLIQUE

Δεδομένα: Γράφημα  $G = (V, E)$   
Θετικός αριθμός  $j \leq |V|$

Ερώτηση: Περιέχει ο  $G$  μήδια μεγέθους  $\geq j$ ;

δηλαδή, υπάρχει  $V' \subseteq V$ :  $|V'| \geq j$  και

$$\forall u, v \in V', (u, v) \in E ;$$

### SUBGRAPH - ISOMORPHISM

Δεδομένα: Γραφήματα  $G(V_1, E_1)$ ,  $H(V_2, E_2)$

Ερώτηση: Εξει α το  $G$  υπογράφημα ισομόρφως με τον  $H$  διη.

$\exists V \subseteq V_1, E \subseteq E_1 : |V|=|V_2|, |E|=|E_2|$  και συναρτηση  $f: V_2 \rightarrow V$  οπ. η οποία πρέπει να ισχύει  $(u, v) \in E_2 \Leftrightarrow (f(u), f(v)) \in E$ ;

### Partition

Δεδομένα Πεπερασμένο σύνολο  $A$  με βάρος  $w(a) \in \mathbb{Z}^+$ ,  $\forall a \in A$ .

Ερώτηση: Μπορεί το  $A$  να μοιραστεί σε 2 ίσοβαρι υποσύνολα;

$$\exists A' \subseteq A : \sum_{a \in A'} w(a) = \sum_{a \in (A-A')} w(a);$$

### D KNAKPSACK (Discrete Knapsack)

Δεδομένα: Πεπερασμένο σύνολο  $U$ ,

- διαφορετικού βάρους  $w(u) \in \mathbb{Z}^+$ ,  $\forall u \in U$
- συναρτησης ποσού  $p(u) \in \mathbb{Z}^+$ ,  $\forall u \in U$
- θετικοί αριθμοί  $W, P$

Ερώτηση: Μπορεί να διαλέξουμε μέρια αντικείμενα του  $U$  με συνολικό βάρος  $\leq W$  και αξία  $\geq P$ ;

$$\exists U' \subseteq U : \sum_{u \in U'} w(u) \leq W \quad \text{και} \quad \sum_{u \in U'} p(u) \geq P$$

Θεώρημα: To 3-SAT είναι NP-complete

Άποδειξη:

- 3-SAT  $\in$  NP ✓

- SAT  $\leq_m^*$  3-SAT

$$\left. \begin{array}{l} C = \{c_1, c_2, \dots, c_m\} \\ U = \{z_1, z_2, \dots, z_n\} \end{array} \right\} \Leftrightarrow \left. \begin{array}{l} C', V' : |c| = 3 \wedge c \in C' \\ 3 \text{-SAT} \end{array} \right.$$

a)  $c \in C : |c|=1$ . Εφών  $c=z$ .

φυλακή με clauses.

$$(z + y_1 + y_2) \cdot (z + \bar{y}_1 + y_2) \cdot (z + y_1 + \bar{y}_2) \cdot (z + \bar{y}_1 + \bar{y}_2)$$

b)  $c \in C : |c|=2$ . Εφών  $c=z_1 + z_2$

φυλακή 2 clauses.

$$(z_1 + z_2 + y_1) \cdot (z_1 + z_2 + \bar{y}_1)$$

c)  $c \in C : |c|=3$ . Τριάδες με 3 χρονι.

d)  $c \in C : |c| > 3$ . Εφών  $c = (z_1 + z_2 + \dots + z_k)$

φυλακή με clauses

$$(z_1 + z_2 + y_1) \cdot (\bar{y}_1 + z_3 + y_2) \cdot (\bar{y}_2 + z_4 + y_3) \cdot \dots \cdot$$

$$(\bar{y}_{k-4} + z_{k-2} + y_{k-3}) \cdot (\bar{y}_{k-3} + z_{k-1} + z_k)$$

☒

ΑΛΓΟΡΙΘΜΟΣ: To 2-SAT  $\in P$

Өмүрбек : To Vertex Cover  $\in$  NP-complete.

Аттасыз:

- $VC \in NP$
- $3SAT \leq_m^P VC$

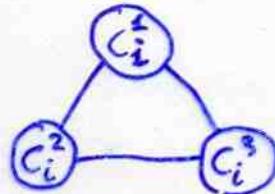
$$\left. \begin{array}{l} C = \{C_1, C_2, \dots, C_m\} \\ U = \{u_1, \dots, u_n\} \end{array} \right\} \Rightarrow G = (V, E), \quad k \in \mathbb{Z}^+, \quad k \leq |V| :$$

$G$  екин  $VC$  жүйесінде  $\leq k \Leftrightarrow C_1 \cdot C_2 \cdot \dots \cdot C_m$  ерталық шарттарынан

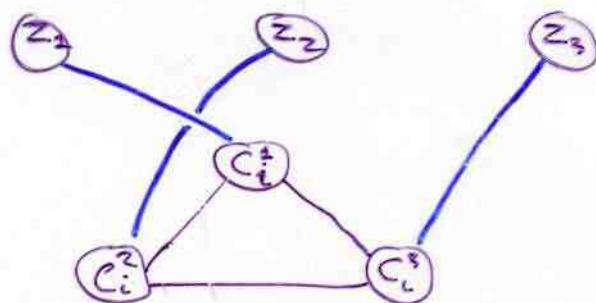
a)  $\forall u_i \in U : \text{есең } G \text{ жүйесінде } u_i \text{---} \bar{u}_i$



b)  $\forall c_i \in C : \text{есең } G \text{ жүйесінде } c_i$



c)  $\forall c_i = (z_1 + z_2 + z_3) : \text{есең } G \text{ жүйесінде}$



d)  $k = n + 2m$

$$G : |V| = 2m + 3n$$

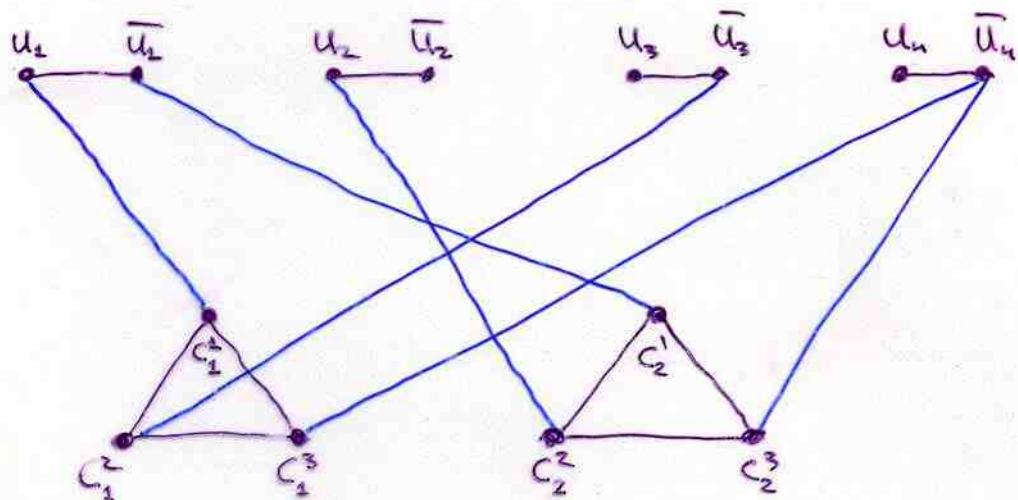
$$|E| = m + 6n$$

Паріважура:

$$\Phi: (U_1 + \bar{U}_3 + \bar{U}_4) (\bar{U}_1 + U_2 + \bar{U}_4)$$



G:



$$K = n + 2m = 4 + 2 \cdot 2 = 8$$

ΕΞΙΔΩΣΗ: Το πρόβλημα CLIQUE είναι NP-complete.

- CLIQUE  $\in$  NP

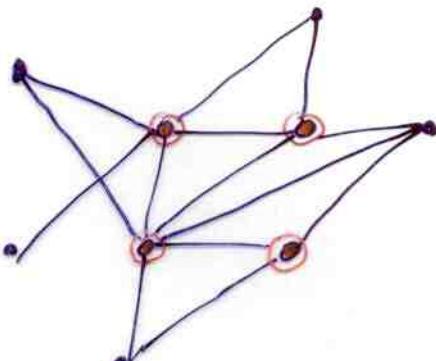
- $\text{VC} \leq_m^p \text{Clique}$

$$\left\{ \begin{array}{l} \text{VC} \\ G = (V, E) \\ k \in \mathbb{Z}^+ \\ k \leq |V| \end{array} \right\} \quad \left\{ \begin{array}{l} \text{Clique} \\ G' = (V', E') \\ j \in \mathbb{Z}^+ \\ j \leq |V'| \end{array} \right\}$$

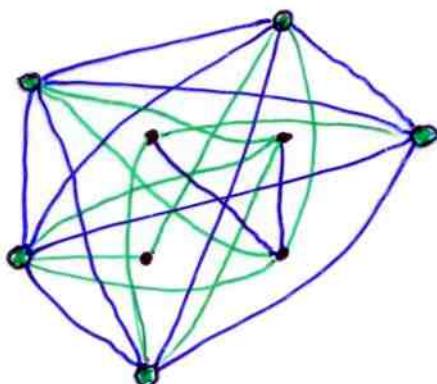
$G$  έχει VC  
μηδενος  $\leq k$        $\Leftrightarrow$   $G'$  μηδενος  $\geq j$

- $G' = \bar{G}$  :  $V' = V$ ,  $E' = \{(u, v) : u, v \in V \text{ and } (u, v) \notin E\}$
- $j = |V| - k$

Λαζαρίδης



$$k=4$$



$$j = 9 - 4 = 5$$

Θεώρημα: To Traveling Salesman Problem είναι NP-complete

- TSP  $\in \text{NP}$
- Hamilton Circuit  $\leq_m^P \text{TSP}$

$$\left\{ G = (V, E) \begin{array}{l} \\ \text{HC} \end{array} \right\} \quad \left\{ \begin{array}{l} \cdot G' = (V', E'), \text{πάτωμα, με βαρούς} \\ \cdot B \in \mathbb{Z}^+ \end{array} \right.$$

$G$  έχει μοντο Hamilton  $\Leftrightarrow G'$  έχει tour με βαρούς  $\leq B$

$$G' = (V, E') : E' = \{(u, v) \mid u, v \in V\}$$

$$w(u, v) = \begin{cases} 1 & (u, v) \in E \\ 2 & (u, v) \notin E \end{cases}$$

$$B = |V|$$

ΦΕΥΔΟΠΟΛΥΟΝΥΜΙΚΗ ΝΖΗ ΤΟΥ PARTITION

PARTITION:

Δεδομένα :  $A = \{a_1, a_2, \dots, a_n\}$

size  $s(a) \in \mathbb{Z}^+ \quad \forall a \in A$

Έρωτης : Υπάρχει  $A' \subset A$  :

$$\sum_{a \in A'} s(a) = \sum_{a \in A - A'} s(a) \quad ?$$

Παραδείγμα:

$$A = \{a_1, a_2, a_3, a_4, a_5\}$$

$$s(a_1) = 1$$

$$s(a_2) = 9$$

$$s(a_3) = 5$$

$$s(a_4) = 3$$

$$s(a_5) = 8$$

$$\text{Έστω } A' = \{a_3, a_5\}$$

$$\Rightarrow \sum_{a \in A'} s(a) = s(a_3) + s(a_5) = 5 + 8 = 13$$

$$\sum_{a \in A - A'} s(a) = s(a_1) + s(a_2) + s(a_4) = 1 + 9 + 3 = 13$$

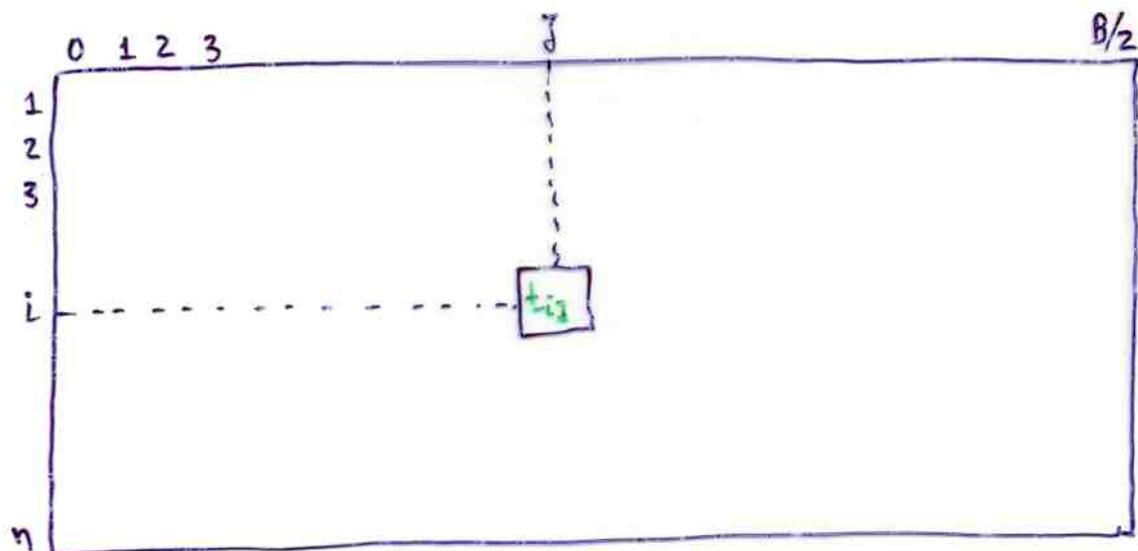
$$\cdot \text{Εστω } B = \sum_{a \in A} s(a)$$

- Εάν το  $B$  είναι περιζός  $\Rightarrow$   $\nexists A'$  του μακοπεύσι το πρόβλημα
- Για  $1 \leq i \leq n$ ,  $0 \leq j \leq \frac{B}{2}$  ορίζω την πρόσθια:

"Υπάρχει υποσύνολο του  $\{a_1, a_2, \dots, a_i\}$  με αδρούμε μεγέθυνση των διαφορών του 100 με  $j$ "

$t(i,j)$  η (boolean) τιμή της πρόσθιας

- Οι σημείοι  $t(i,j)$  μπορεύ να διαταχθούν σε δισταστατο πινακα διαστάσεων  $n \times (\frac{B}{2} + 1)$



(επονέμεται,  $n=5$ ,  $B=13$ )

- Ο Τίτλος συμβιβάεται ως εξής:

Γραμμή-1:  $t(1, j) = T \Leftrightarrow j=0 \text{ or } j=s(a_1)$

Γραμμή- $i$ :  $t(i, j) = T \Leftrightarrow t(i-1, j) = T \text{ or}$   
 $(s(a_i) < j \text{ and } t(i-1, j-s(a_i)) = T)$

Παράδειγμα:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	T	T												
2	T	T							T	T				
3	T	T			T	T			T	T				
4	T	T	T	T	T	T		T	T	T	T	T	T	T
5	T	T	T	T	T	T	-	T	T	T	T	T	T	T

$$s(a_1)=1 \quad s(a_2)=9 \quad s(a_3)=5 \quad s(a_4)=3 \quad s(a_5)=8$$

- Πολυπλοκότητα:  $O(nB)$

Αποδείξαρε ότι  $P=NP$  ??

ΟΧΙ

• Καθε αυριανός  $s(a_i)$  αναπαριστάται με  $O(\log s(a_i))$  bit

⇒ Συνολικό μέγεθος είδοσου:

$$\text{length}(I) = \sum_{i=1}^n \log s(a_i) \leq \sum_{i=1}^n \log B = O(n \log B)$$

\*\*\*  $nB$  δεν φράσσεται από πολυωνυμικό συνδρομ του  $n \log B$