MOVEMENT PROB. ON GRAPH

ISTANZA

IN $G(V,E)$ t.c. $|V|=m$ OUT $M: P \to V$

P t.c. $|P|=k$

$\sigma: P \to V$

GOAL si $U := \{M(p)\}_{p \in P}$ t.c. $U \subseteq V$

CONNECTIVITY:

SOTTOGRAFO INDOTTO DA U CONNESSO

INDIPENDENCY:

U INDIPENDENTE E $|U|=k$

CLIQUE:

U CLIQUE DI G

MEASURE: $p \in P$ È MOSSO DA $\sigma(p)$ A $M(p)$ TRAMITE LO SHORTEST PATH SU G

OVERALL MOVEMENT

$$SUM(M) = \sum_{p \in P} d_G(\sigma(p), M(p))$$

MAXIMUM MOVEMENT

$$MAX(M) = \max_{p \in P} d_G(\sigma(p), M(p))$$

#PEBBLE MOVED

$$NUM(M) = \left| \{ p \in P \text{ t.c. } \sigma(p) \neq M(p) \} \right|$$

IND-MAX HARDNESS

IS $U \subseteq V$ di $G(V,E)$ t.c. $\forall u,v \in U \Rightarrow (u,v) \notin E$

max IS U^* t.c. $\forall U \in V \Rightarrow |U^*| > U$

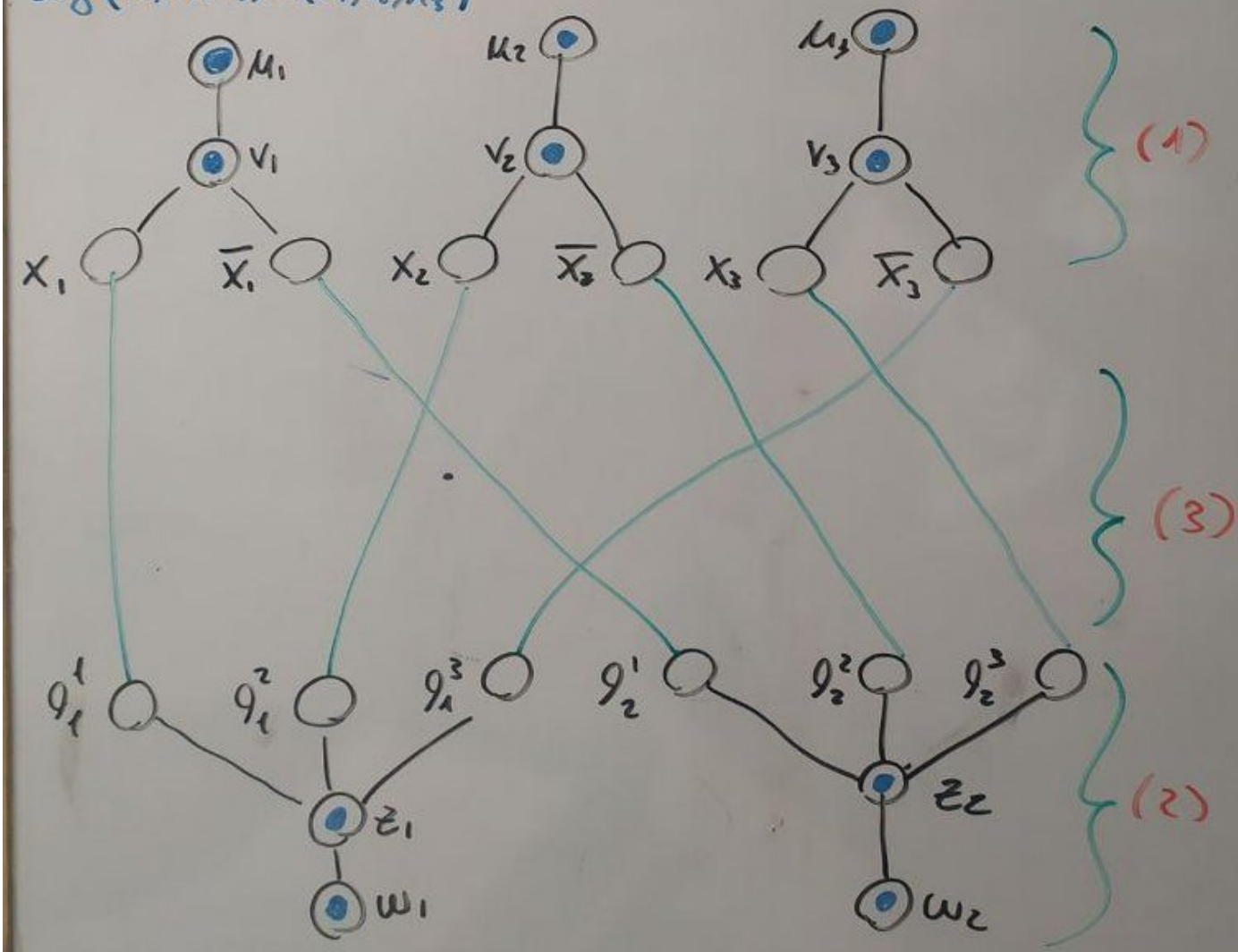
3-SAT → IND-MAX

(1) $\forall x_i \in \varphi$ VARIABLE GADGET

(2) $\forall C_j = (l_j^1, l_j^2, l_j^3)$ t.c. $C_j \in \varphi$ CLAUSE GADJET

(3) $\forall l_j^i \in C_j$ t.c. $i = \{1,2,3\}$ INTRO $\begin{cases} (l_j^1, x_i) \text{ se } l_j^1 = \overline{x_i} \\ (l_j^1, \overline{x_i} \text{ ALTRIMENTI} \end{cases}$

$\varphi = (\overline{x_1}, \overline{x_2}, x_3) \wedge (x_1, x_2, \overline{x_3})$



IND-MAX HARDNESS

CLAIM φ SODDISFACIBILE $\Leftrightarrow$ $\exists$ SOL PER IND-MAX DI COSTO 1

PROOF

($\Rightarrow$) · CONSIDERO T ASSEGNAMENTO DI VERITÀ PER φ

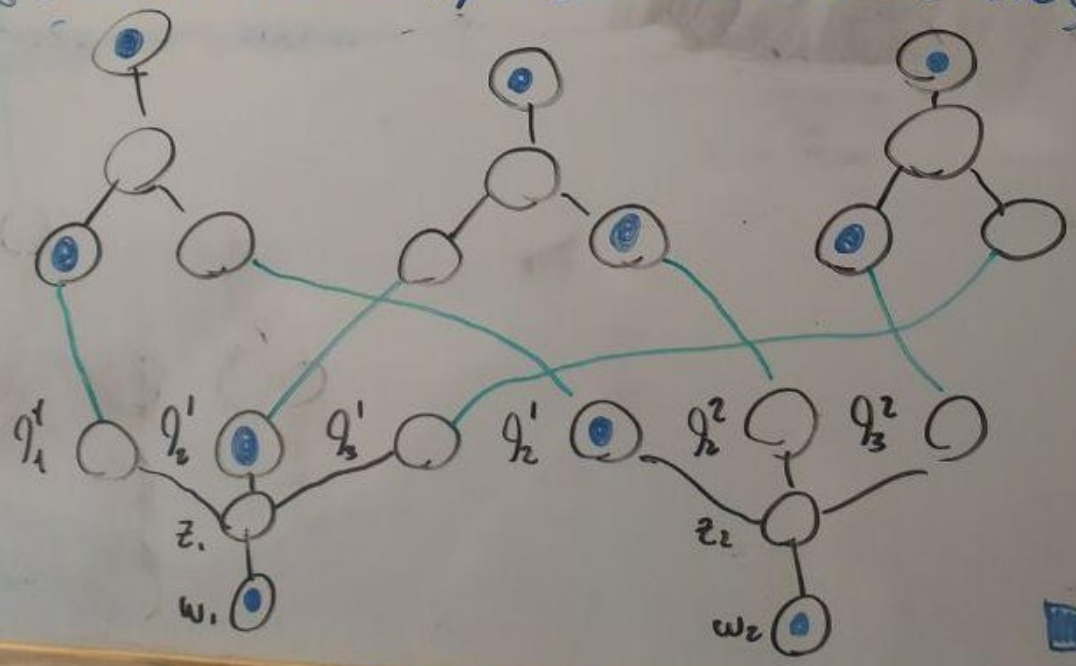
- $\forall x_i \in T$ T/C x_i = TRUE SPOSTO IL PEBBLE DA v_i IN x_i, IN $\bar{x}_i$ ALTR.
- $\forall q_j^k$ = TRUE T/C ADJ(q_j^k) VUOTO $\Rightarrow$ SPOSTO IL PEBBLE DA z_i A q_j^k
- $\forall C_j$ ALMENO UN q_j^k = TRUE T/C ADJ(q_j^k) VUOTO

($\Leftarrow$) · CONSIDERO V SOL DELL'IND-MAX

- OGNI PEBBLE v_i DEVE ESSERE STATO SPOSTATO IN x_i O $\bar{x}_i$, SETTIAMO IN T x_i = TRUE $\oplus$ x_i = FALSE
- OGNI PEBBLE SU z_i DEVE ESSERE STATO SPOSTATO IN q_j^i, E ADJ(q_j^k) VUOTO
- q_j^k SODDISFATTO SODDISFA C_j

□

■ $\varphi = (\bar{x}_1, \bar{x}_2, x_3) \wedge (x_1 \vee x_2 \vee x_3)$, $T := \{x_1 = \text{TRUE}, x_2 = \text{FALSE}, x_3 = \text{TRUE}\}$



APPROXIMABILITY OF IND-MAX

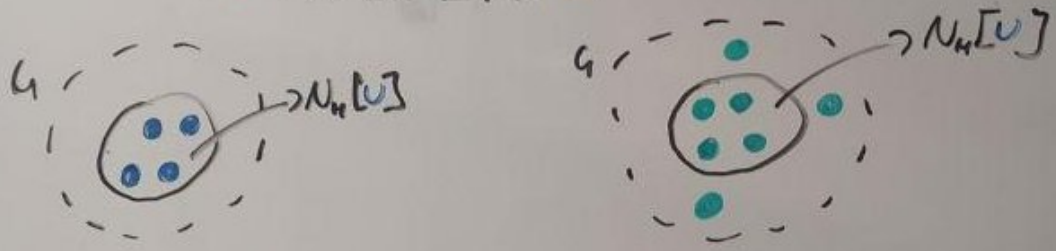
TH (HALL'S MATCHING)

- SIA $H=(V_1+V_2, E)$ GRAFO BIPARTITO, ALLORA
$\exists$ MATCHING M T/C $|M|=|V_1| \Leftrightarrow |A| \leq N_H(A), \forall A \subseteq V_1$

LEMMA

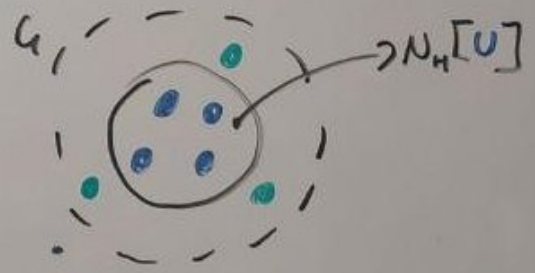
- SIA U^* UN max IS di $G \Rightarrow \forall U$ IS di G:

$$|U^* \cap N_H[U]| \geq U$$



PROOF. SUPPONIAMO $|U^* \cap N_H[U]| < U$

$\Rightarrow U' = (U^* \setminus N_H[U]) \cup U$ É UN IS



$\Rightarrow |U'| > |U^*|$ CONTRADDIZIONE

□

APPROXIMABILITY OF IND-MAX

LEMMA $\forall I \subseteq U$ di G $\exists g: U \to U^*$ INIETTIVA

T/C $d_G(u, g(u)) \leq 1$

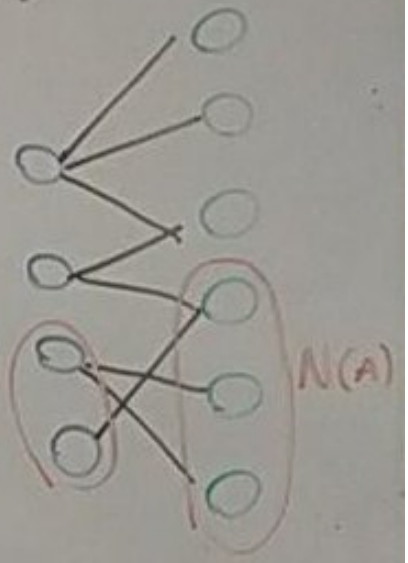
PROOF. COSTRUIAMO G BIPARTITO $H=(U \cdot U^*, E)$

. CONNETTIAMO $u \in U$ A $U^* \cap N[\{u\}]$

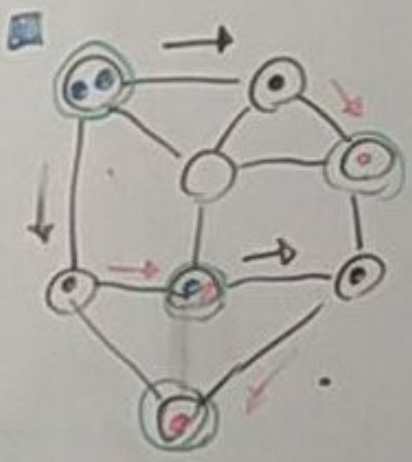
. DALL' HALL'S MATCHING

$$\forall A \subseteq U \Rightarrow N(A) = |U^* \cap N_G[A]| \geq A$$

□



CONSEGUENZA: PER IND-MAX $\exists$ SOL DI COSTO OPT+1 CHE MUOVE OGNI PeP SU VERTICI DI U^*



// COST = OPT = 1

// COST = OPT + 1 = 2

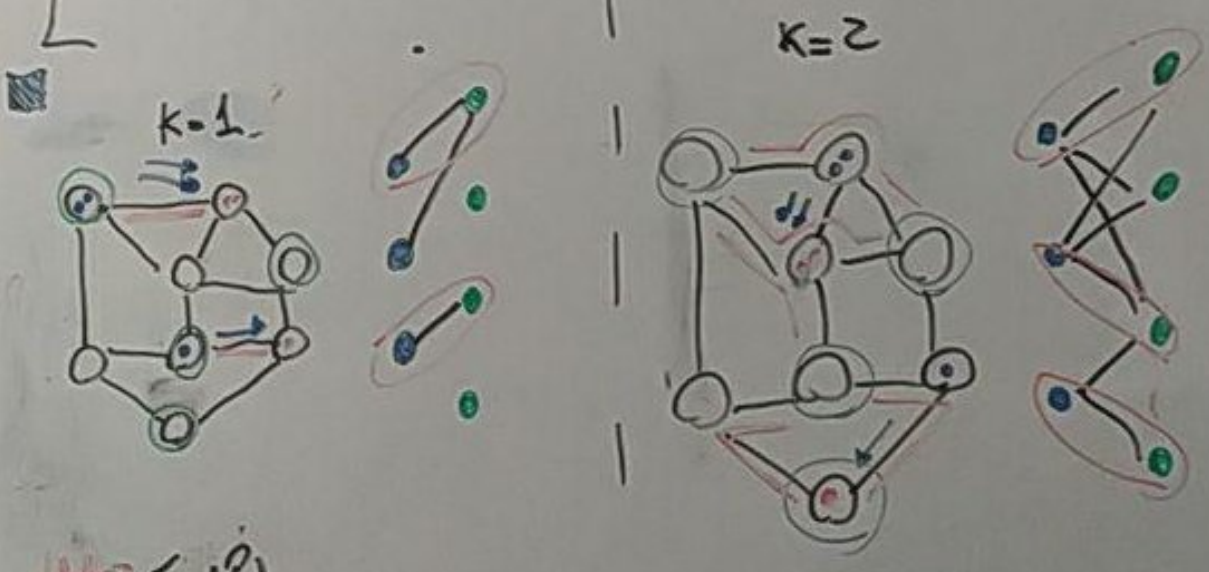
■

APPROXIMABILITY OF IND-MAX

TH SE UN max IS DI G TROVABILE IN POLY. T. (AD ES SU GRAFI BIPARTITI) ⇒ IND-MAX PUÒ ESSERE TROVATO CON ADDITIVE ERROR 1

ALGO
• Come trovo IND-MAX dato U^*?

$U^* \leftarrow$ max_IS(G)
if $|U^*| < |P|$
  RET NO_SOL
for k=0 to |V|-1
  $F = \{(p,u) \in P \times U^* \text{ t/c } d(\sigma(p),u) \leq k\}$
  $H = (P, U^*, F)$
  M = maxBipartiteMatching(H) // HALL'S TH
  if $|M| = |P|$
    RET M



$|M_{k=1}| < |P|$

$|M| = |P| = 3$

DISTRIBUTED VERTEX COLORING

- VERTEX COLORING $G(V,E)$ T/C $\forall v \in V$ COLORATO
- VALID VERTEX COLORING $\exists v_1, v_2 \in V$ ADIACENTI T/C $C_{v_1} = C_{v_2}$
- k-COLORING VERTEX-COLORING CON K-COLORI
- CHROMATIC NUMBER $\varphi(G) = \min K$ T/C $\exists$ K-COLORING (G)

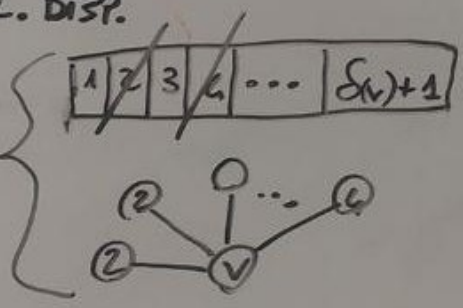
$\Delta+1$ COLORING

- $\Delta = \max_{v \in V} \delta(v) = \delta(G)$
- **CLAIM** OGNI G AMMETTE $(\Delta+1)$-COLORING

$$\varphi(G) \leq \Delta + 1, \forall G$$

↳ Proof

- $\forall v \in V \exists$ Palette(v) CON $\delta(v)+1$ COL. DISP.
- $\pi = \#N_{col}(v)$ $(\pi \leq \delta(v)$ OBV.)
- MARK NON-DISP i COL. DI Palette(v) GIÁ ASSEGNATI A $MEN_{col}(v)$
- # COL RIMASTI IN Palette(v):



$$\#COL\_RIMASTI \geq \delta(v) + 1 - \pi \geq \delta(v) + 1 - \delta(v) = 1$$

- RIMANE SEMPRE ALMENO UN COLORE, E PUÓ ESSERE USATO PER v ■

ALGO

> $\forall v \in V$ CREO Palette(v) CON $\delta(v)+1$ COL.
> WHILE $\exists v \in V$ NON COLORATO
>> COLORO v CON $c \in$ Palette(v) DISP
>> FORALL $u \in N(v)$
>>> MARK Palette(u) NON-DISP

DISTRIBUTED VERTEX COLORING

COLORING & IS: IN OGNI VALIDO VERTEX-COLORING(G), L'INSIEME DI NODI CON STESSO COLORE FORMANO UN IS

MIS-COLORING ALGO:

> $C = 1$
> WHILE $\exists$ $v \in V$ NON COLORATO
> > TROVA MIS I DEL SOTTO GRAFO DI G INDOTTO DAI $v \in V$ NON COLORATI
> > ASSEGNA C A OGNI $v \in I$
> > C++

↳ ANALYSIS

LEMMA MASSIMO $\Delta+1$ ITERAZIONI ($\Delta+1$ COLORI)

↳ Proof

- K : ITERAZIONE
- $\text{eff-d}(v) = |N_{\text{non-col}}(v)|$: EFFECTIVE DEGREE
- END OF K:
  - OGNI $v \in V$ È ADIACENTE A $u \in I$ (ALTR. $|I \cup \{v\}| > |I|$ (MIS) $\Rightarrow \perp$)
  - $\text{eff-d}(v) --$
- END OF $K = \Delta$:
  - $\text{eff-d}(v) = \phi$
- END OF $K = \Delta+1$:
  - OGNI v RIMASTO ENTRA IN I

COMPLEXITY

- $O(\Delta+1)\,O(\log \Delta \log n) = O(\Delta \log \Delta \log n)$ (HIGH PROBABILITY)

$O(\Delta+1)$: # ITERATIONS; $O(\log \Delta \log n)$: LUBY (HIGH PROB)

DISTRIBUTED VERTEX COLORING

2Δ-COLORING ALGO

- $|Palette(v)| = 2\delta(v)$
- k = FASE CORRENTE
- SE IN k, v NON È COLORATO
  - SCEGLI $C_v \in Palette(v)$ CANDIDATO (scelta uniforme)
  - SE $C_v \neq C_u \ \forall u \in N$, C_v DEFINITIVO
  - SENNÒ RIGETTA C_v
- I NODI CHE RIGETTANO PASSANO A $k+1$
- TERMINA SE OGNI v COLORATO

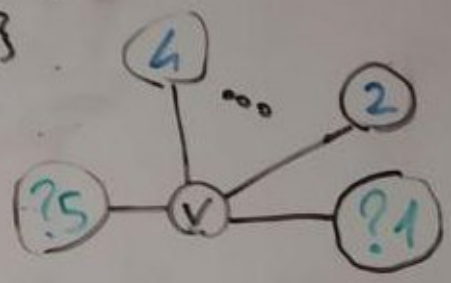
ALGO (per nodo v)

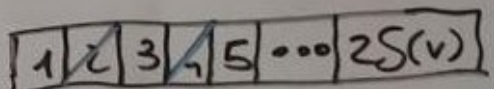
- WHILE v NON-COLORATO
  - SCEGLI $C_v \in$ Palette di v
  - SE $C_u = C_v$ PER QUALCHE $u \in N(v)$
    - RIGETTA C_v
  - SENNÒ
    - ACCETTA C_v COME DEFINITIVO
    - INFORMA OGNI $u \in N(v)$ DEL COLORE SCELTO
      (COSÌ u PUÒ RIMUOVERE C_v DA $Palette(u)$)

DISTRIBUTED VERTEX COLORING

ANALYSIS

- CONSIDERO v ALLA FASE k
- $A(v) := \{ c_v \text{ DISPONIBILE} \mid c_v \in Palette_k(v) \}$
- $U(v) := \{ c_u \text{ CANDIDATO} \mid u \in N^{(v)}_{NON\text{-}COL} \}$
- $A'(v) = A(v) \setminus U(v)$
- $\delta = |N^{(v)}_{NON\text{-}COL}|$





$Palette_k(v)$

$A(v) = \{1, 3, 5, \_, 25(v)\}$

$U(v) = \{5, \dots, 1\}$

$$|A'| = |A(v) \setminus U(v)| \geq |A(v)| - |U(v)| \geq$$

$$\geq \underbrace{(25(v)}_{\max |Palette(v)|} - \underbrace{(\delta(v) - \delta)}_{N^{(v)}_{col}} - \underbrace{\delta}_{|U(v)| \leq \delta} = \delta(v)$$

- $Prob_k(v \text{ ACCETTI } c_v) = \dfrac{|A'(v)|}{|A(v)|} \geq \dfrac{\delta(v)}{25(v)} = \dfrac{1}{2}$

- $Prob\ (v \text{ RIGETTA } c_v) \leq \left(1 - \frac{1}{2}\right)^{2\log m} = \dfrac{1}{2^{2\log m}} = \dfrac{1}{m^2}$
  $*\ k=1\_2\log m$

- $Prob\ (\text{ALMENO UN } v \text{ RIGETTA } c_v) \leq m \cdot \dfrac{1}{m^2} = \dfrac{1}{m}$
  $*\ k=1\_2\log m$

- $Prob\ (\text{OGNI } v \text{ ACCETTA } c_v) \geq 1 - \dfrac{1}{m}$
  $*\ k=1\_2\log m$

$k=1\_2\log m$ $*$ $Prob(x)$ IN $2\log m$ FASI

- 2Δ-COLORAZIONE ALLO INTERNO, CON $Prob \geq 1 - \frac{1}{m}$; $2\log m$ FASI, OGNI FASE RICHIEDE $O(\Delta)$ STEPS.

COMPLESSITÀ $O(\log m)$ (HIGH PROB)

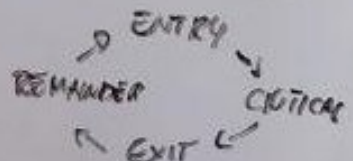
SHARED MEMORY SYSTEMS (SMS)

- NO COMMUNICATION CHANNEL
- NOTIFICA STATO VIA SHARED VARS
- SHARED VARIABLE TYPE DEFINES ATOMIC OPERATIONS PERMITTED

MUTEX PROBLEM

- ... TO COORDINATE ACCESS
- ASSUME: N IN ANONYMOUS, NON UNIFORM, ASYNC.
- MUTEX ALGOS: SPECIFICA ENTRY/EXIT PER GARANTIRE:
  - MUTUAL EXCLUSION
  - LIVENESS CONDITION
    - (a) NO DEADLOCK
    - (c) NO LOCKOUT
    - (s) BOUNDED WAITING
- COMPLEXITY: SHARED SPACE (DIPENDE DA TYPE & LIVENESS COND)

// Process Code



BAKERY ALGO

- GARANTISCE MUTUAL EXCLUSION E BOUNDED WAITING
- n SHARED VARS (R/W)
- (ALGO)

```
. CHOOSING_i = TRUE
. NUMBER_i = max{NUMBER_0 ... NUMBER_{n-1}} + 1
. CHOOSING_i = FALSE
. FOR j=0 ... n-1 t.c. j≠i
      WAIT UNTILL CHOOSING_j = FALSE
      WAIT UNTILL NUMBER_j = 0 OR NUMBER_j > NUMBER_i
```

} ENTRY

```
. NUMBER_i = 0
```

} EXIT

BAKERY ALGO

- BAKERY ALGO MUTUAL EX.

LEMMA 1: SE P_i IN CS $\Rightarrow NUM_i > 0$

LEMMA 2: SE P_i IN CS E $NUM_k \neq 0, k \neq i \Rightarrow NUM_k > NUM_i$

PROOF

- (CASO 1) P_k SCEGLIE DOPO P_i, DATO CHE NUM_i È STATO PRESO, P_k SCEGLIE $NUM_k > NUM_i$
- (CASO 2) P_k SCEGLIE PRIMA DI P_i, DATO CHE NUM_k CHE È STATO PRESO, P_i SCEGLIE $NUM_i > NUM_k$

□

$\Rightarrow$ SUPPONIAMO P_i E P_k IN CS

- DA LEMMA 1: $NUM_i, NUM_k > 0$
- DA LEMMA 2:
  - $NUM_k > NUM_i$
  - $NUM_i > NUM_k$

// CONTRADDIZIONE

- B. ALGO NO LOCKOUT: P_i PUÒ BLOCCARSI AL PIÙ AL SECONDO WAIT, OGNI ALTRO PROCESSO O ENTRA E PRENDE UN TICKET MAGGIORE O ESCE E PRENDE TICKET $\emptyset \Rightarrow P_i$ ENTRA PER FORZA.
- B. ALGO BOUNDED WAITING: P_i NELLA ENTRY, PUÒ ESSERE SORPASSATO AL PIÙ UNA VOLTA DA OGNI PROCESSORE (AL PIÙ n-1 VOLTE)

BOUNDED-SPACE 2-PROCESSOR MUTEX ALGO

- 2 BINARY SHARED VARS W[0], W[1]
- ASYMMETRIC CODE: P_0 HA SEMPRE PRIORITÀ

P_0 CODE

```
1.
2.
3. W[0] = 1
4.
5.
6. WAIT UNTIL(W[1]=0)      } ENTRY

7.
8. W[0] = 0                } EXIT
```

P_1 CODE

```
1. W[1] = 0
2. WAIT UNTIL(W[0]=0)      } ENTRY
3. W[1] = 1
4.
5. IF(W[0]=1) GOTO (1)
6.

7.
8. W[1] = 0                } EXIT
```

- MUTUAL EXCLUSION: P_i ENTRA SOLO SE $W_i = 1$ E $W_{i-1} = 0$
  - SE P_1 IN CS E P_0 A (5) W[0] = W[1] = 1
  - SE P_0 IN CS E P_1 A (2) W[0] = 1, W[1] = 0
- NO DEADLOCK: SE P_0 SET W[0] = 1 ⇒ P_1 FORZATO A (5) A SETTARE W[1] = 0
- NO LOCKOUT: SE P_0 SET W[0] = 1 MENTRE P_1 TRA (3) E (5), E LO FA CONTINUAMENTE ⇒ LOCKOUT DI P_0

↳ INTRODUCIAMO LA VAR PRIORITY (ACQUISIAMO LA PRIORITÀ)

P_i CODE

```
1. W[i] = 0
2. WAIT UNTILL (W[1-i]=0 o PRIORITY != i)   } ENTRY
3. W[i] = 1
4. IF (PRIORITY = i-1)
5.     IF (W[i-1] = 1) GOTO (2)
6. ELSE
7.     WAIT UNTIL (W[i-1] = 0)

8. PRIORITY = i-1          } EXIT
9. W[i] = 0
```

ANALISI

- MUTUA ESCLUSIONE: SUPPONIAMO ENTRAMBI IN CS, ALLORA w[0], w[1] SONO TRUE; QUINDI w[0], w[1] SETTATE A (3) UNO DEI DUE DEVE BLOCCARSI A (5) o (6) → CONTRADDIZIONE

- NO DEADLOCK: SUPPONIAMO IL DEADLOCK E PRIORITY=0
  - P_0, P_1 BLOCCATI IN ENTRY (2), w[0]·w[1] = 0
  - P_0 HA LA PRIORITY, OLTREPASSA (2) E VA A (6) CON QUINDI w[0]=1
  - P_0 OLTREPASSA (6) PERCHÉ w[1]=0
  - P_0 ENTRA IN CS, E USCENDO PRIORITY=1 E w[0]=0
  - P_1 OLTREPASSA (2)

  → CONTRADDIZIONE

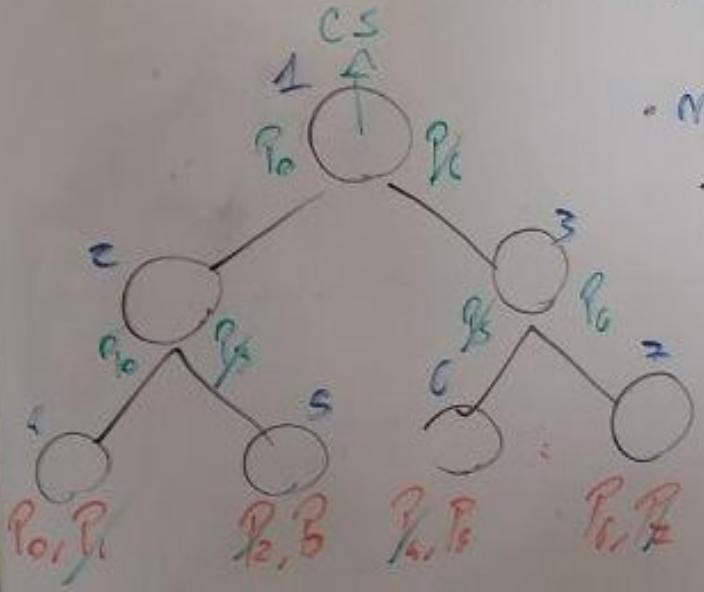
- NO LOCKOUT: SUPPONIAMO P_0 BLOCCATO A (2)
  - P_1 VA AVANTI, NELL'EXIT SECTION SETTA PRIORITY = 0
  - P_0 VA AVANTI

  → CONTRADDIZIONE

BOUNDED-WAITING: (NO) P_1 POTREBBE ENTRARE ALL'INFINITO MENTRE È TRA (2) E (3)

BOUNDED SPACE N-PROCESSOR MUTEX ALGO

- ASSUMIAMO #PROCESSORI $n = 2^k$, $k > 1$
- COSTRUIAMO TOURNAMENT TREE ($n-1$ nodi)
- IL 2-PROCESSORE ALGO ASSOCIATO A OGNI NODO



- $n = 2^3 = 8$ #PROCESSORI
- ASSOCIA P_i, P_{i+1} ALLE FOGLIE DA SX A DX
- SE MANCANO i PROCESSI ($n \neq 2^k$) COMPLETIAMO L'ALBERO CON DUMMY LEAVES
- 2-PROCESS. ALGO A OGNI FOGLIA P_i: LA LOCAZIONE

ANALYSIS: COME 2-PROC ALGO

- COMPLEXITY = $3(n-1)$ BOOL R/W VARS

↳ OGNI 2-PROC USA 3 VARS, I NODI SONO $n-1$

CONSENSUS PROBLEM

FALURES { LINK-FAIL; PROCESSOR CRASH-FAIL; BYZANTIN-FAIL }

ALGO PROPERTIES {
- TERMINATION: $\forall p$ NON-FAULTY SCELGE $y \in Y$
- AGREEMENT: $\forall p$ NON-FAULTY SCELGONO STESSO $y \in Y$
- VALIDITY: $\forall x \in X$ SETUP UGUALI $y \in Y$ T.C. $y = x$ $\forall p$ NON FAULTY
}

LINK-FAIL

- $\exists$ ISTANZE DI INPUT T.C. NON RAGGIUNGIBILE CONSENSO IN CASO DI LINK-FAIL

NEGATIVE RES.: 2 GENERALI

- $\Rightarrow$ SIA $\mathcal{A}$ SHORTEST-PROTOCOL PER RAGGIUNGERE CONSENSO
- $\Rightarrow$ SE m MSG VIENE PERSO E IL CONSENSO RAGGIUNTO COMUNQUE
  $\Rightarrow$ $\mathcal{A}$ NON È SHORTEST $\perp$

PROC-FAIL

NEGATIVE RES.: ASYNCHRONO $\rightarrow$ IMPOSSIBILE CONSENSO ANCHE PER UN SOLO CRASH-FAIL

POSITIVE RES: SYNCRONO E CLIQUE TOPOLOGY

- COMPLETO (SO NON UNIFORME)
- SYNC-START
- ROUND. LEGGE MSGS $\rightarrow$ INVIA MSGS $\rightarrow$ LEGGE MSGS

• SIMPLE ALGO (FAULT-FREE)

- $\forall p$:
  - > BROADCAST VAL;
  - > LEGGE MSGS IN INPUT
  - > SCEGLIE IL MINIMO VAL

$\rightarrow$ 1 ROUND (COMPLETO)
$\rightarrow$ NON WORKA CON CRASH-FAIL

- F-RESILIENT TO CRASH-FAIL ALGO

- $\forall P_i$
  - > ROUND 1: BROADCAST ~~V_i~~ VAL_i
  - > ROUND 2 TO F+1: BROADCAST OGNI VALORE RICEVUTO (UN MSG A VALORE)
  - > END F+1. SCEGLIE IL MINIMO VAL.

- **LEMMA 1** END F+1 $\Rightarrow$ OGNI P_i CONOSCE STESSI VALS

PROOF

- SIA LEMMA 1 FALSO $\Rightarrow$ $\exists$ VAL CONOSCIUTO DA $P' \in P$ A END F+1
- $P' \in P'$ NON POTREBBE CONOSCERE x AL ROUND F, LO AVREBBE BROADCASTATO
- P' RICEVE x AL ROUND F+1, MA ESSENDO F+1 IL ROUND SENZA FAILURE x VIENE BROADCASTATO $\Rightarrow P' = P$ $\perp$ □

. CORRETTEZZA

- AGREEMENT: SYNCH-START $\Rightarrow$ SAME-KNOWLEDGE DEL ROUND F+1 NONCHALANCE
- VALIDITY: IL VAL SCELTO FA PARTE DEI VALORI IN INPUT

. PERFORMANCE

$$O(m \cdot m \cdot k) = O(m^3)$$

- m: #PROCs
- m: P_i MANDA 1 MSG PER OGNI PROCESSORE
- k: K DIVERSI IN INPUT, $k = O(m)$

LOW-BOUND (CRASH-FAIL)

TH OGNI F-RESILIENT TO CRASH-FAIL CONSENSUS ALGO RICHIEDE ALMENO $f+1$ ROUND

PROOF SE BASTASSERO MENO DI $f+1$ ROUND, TUTTI DOVREBBERO AVERE LA STESSA KNOWLEDGE

. WORST-CASE

- UN PROC A ROUND FALLISCE, MANDANDO IL SUO VAL SOLO A UN ALTRO
- IL PROCESSORE CHE HA RICEVUTO IL VAL. SARÀ IL PROSSIMO A FALLIRE
- AL ROUND $f+1$, P_i HA RICEVUTO VAL, MA SE LO SCEGLIE, SARÀ L'UNICO A FARLO, POICHÉ È L'UNICO (NON CRASHATO) AD AVERLO RICEVUTO

$\Rightarrow$ f ROUND NON SONO ABBASTANZA! □

BYZANTINE-FAIL

LOW-BOUND (BYZ-FAIL)

TH OGNI F-RESILIENT TO BYZ+FAIL ~~ALGO~~ CONSENSUS ALGO RICHIEDE ALMENO $f+1$ ROUND

PROOF SEGUE DAL LOW-BOUND (CRASH-FAIL)

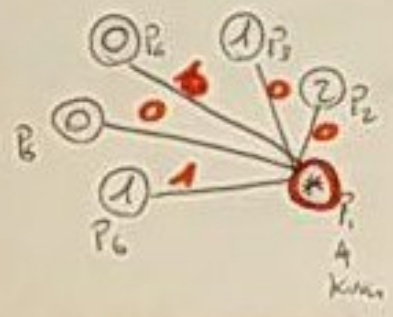
F-RESILIENT TO BYZ-FAIL ALGO (KING ALGO)

- RISOLVE IN $2(f+1)$ ROUNDS PER n PROCS DI CUI $n/4$ BYZ.
- $f \leq n/4 \Rightarrow n \geq 4f+1$
- ASSUNTO $\Rightarrow$ NON-UNIFORM / DISTINT. IDs (NON-ANONIM)
- P_i: PROC DI ID i
- $f+1$ FASI DA 2 ROUND
- UN KING DIVERSO A FASE $\Rightarrow$ ∃ UN KING NON FAULTY

> FASE $k = 1, \dots, f+1$
>
> ROUND 1:
>
> BROADCAST V_i
>
> SIA a IL V_i PIÙ FREQUENTE RICEVUTO (MAJORITY VAL) E #OCCORRENZE (MAJORITY)
>
> SIA $1 \leq m_i \leq n$
>
> $\Rightarrow V_i = a$
>
> ROUND 2:
>
> KING P_k BROADCAST V_k
>
> OGNI P_i:
>
> SE IN ROUND 1 HA SCELTO V_i T/C $m_i < n/2 + 1 + f$ (WEAK MAJORITY)
>
> $\Rightarrow V_i = V_k$
>
> SENNÒ:
>
> MANTIENE $V_i = \sim$
>
> FINE $f+1 \Rightarrow P_i$ SCEGLIE IL SUO V_i

6 PROCS, 1 FAULT => 2 FASI

-> ROUND 1, FASE 1



	SCELGONO
P_2 0 2 1 0 0 1 =>	∅
P_3 0 2 1 0 0 1 =>	∅
P_4 1 2 1 0 0 1 =>	1
P_5 0 2 1 0 0 1 =>	∅
P_6 1 2 1 0 0 1 =>	1

=> TUTTE LE SCELTE HANNO WEAK-MAJORITY $3 < n/2 + f + 1 = 5$

L> AL ROUND 2 SCELGONO IL KING VALUE

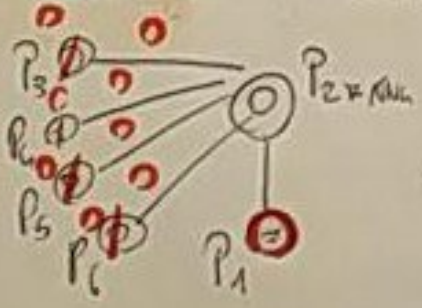
L> ROUND 2, FASE 1

-> ROUND 1, FASE 2

	SCELTA
~~P0 3 0 0 0 0 1~~	
P_2 0 3 1 0 0 2 =>	∅
P_3 0 " =>	∅
P_4 1 " =>	1
P_5 0 " =>	∅
P_6 1 " =>	1

=> TUTTE WEAK-MAJORITY $3 < n/2 + f + 1 = 5$

L> AL ROUND 2 TUTTI SCELGONO IL KING VALUE



// NON FAULTY KING BROADCASTA ∅

• CORRETTEZZA

LEMMA 1: END OF K DOVE KING NON-FAULTY ⇒ OGNI P_i NON-FAULTY SCEGLIE IL KING-VALUE V_K

PROOF

CASO 1: SE OGNI P_i NON-FAULTY HA WEAK MAJ ALLA FINE DI ROUND 1
↳ TUTTI SCEGLIONO V_K AL ROUND 2

CASO 2: SE UN P_i HA STRONG MAJ $V_i = a$
↳ AL MENO $n/2+1$ BROADCASTATO a; ALLORA TUTTI I P_i-NON-FAULTY HANNO RICEVUTO a CON STRONG MAJORITY (KING COMPLESSO)
⇒ TUTTI SCEGLIONO a (O IL PROPRIO O QUELLO DEL KING) □

LEMMA 2: SI $a = V_i$ $\forall P_i$ ALLA FASE $k \Rightarrow$ I V_i NON CAMBIANO PIÙ

PROOF

- AL PIÙ f BYZ-PROC. ⇒ $n-f$ NON-FAULTY
- $f < n/4 \Rightarrow n-f > n/2+f$
- DOPO k a ANCA SEMPRE STRONG MAJORITY ($> n/2+f$) E FINO A $f+1$ OGNI P_i NON-FAULTY SCEGLIERÀ SEMPRE a □

- **AGREEMENT**: C'È SEMPRE UNA FASE SENZA BYZ-FAIL
  → LEMMA 1: IN QUESTA TUTTI SCEGLIONO STESSO a
  → LEMMA 2: DOPO a NON CAMBIA

- **VALIDITY**: SE P_i NON-FAULTY HANNO a IN INPUT ~~AL ROUND 0, AL E~~ ALLA FASE 1, OGNI P_i RICEVE $n-f$ VOLTE a (STRONG MAJ) AL ROUND 1.
  AL ROUND 2 a È IL COMMON-V_i E DA LEMMA 2, QUESTO NON CAMBIA

• PERFORMANCE - COMPLEXITY

- $n > 4f$ #PROCS
- $2(f+1)$ #ROUNDS
- $\Theta(n^2 + f) = O(n^3)$ #MSGS

NOTA BYZ-PROC possono mandare UNBOUNDED MSGS.

n MSGS IN ROUND 1 DA P_i-NON FAULTY
$n-1$ MSGS IN ROUND 2 DA NON-FAULTY KING
$f+1$ ROUNDS #FAULTY-PROC

RANDOMIZED BYZ CONSENSUS

- C'È UN PROCESSORE q AFFIDABILE CHE AD OGNI ROUND LANCIA UNA MONETA
- P_i HA PREFERRED VALUE V_i
- $f < n/8$
- THRESHOLDS $\begin{cases} L = 5n/8 \\ H = 6n/8 \\ G = 7n/8 \end{cases}$

(ALGO)

- P_i AD OGNI ROUND:

```
1  > BCV_i
2  > RCV V_S ∀P_S
3  > SIA MAJ_i (MAJORITY VAL) E TALLY_i (FREQUENZA)
4  > RCV COIN OUTCOME DA q
5  > SE Testa:
6  >     THRESHOLD ← L
7  > SENNO
8  >     THRESHOLD ← H
9  > SE TALLY_i ≥ THRESHOLD:
10 >     V_i ← maj_i
11 > SENNO
12 >     V_i ← ϕ
13 > SE TALLY_i > G
14 > STOP
```

- CASI A FINE ROUND:
  - (1) TERMINATION CASE
  - (2) OTHER CASES:
    - (2.1) P_i, P_k TC $maj_i \neq maj_k$
    - (2.2) $maj_i \ \forall P_i$

⇒ TUTTI I CASI PORTANO CONSENSUS

ANALYSIS

(1) TERMINATION CASE: P_i VEDE $Tally_i > G$ PER maj_i

→ DATO CHE $f < n/8$, I VOTI PER maj_i DA ALMENO TRE SONO $(Tally_i - f) > H$

→ OGNI P_k NON FAULTY È TC E $V_k = maj_k = maj_i$ E $Tally_k > H$, QUINDI TUTTI I NON FAULTY RAGGIUNTO CONSENSO

LEMMA SE ALL'INIZIO DI UN ROUND V_k STESSO $\forall P_k$:
- L'ALGO TERMINA
- L'ALGO RISOLVE CONSENSUS

PROOF

- α SONO ALMENO $7/3m$ PROCESSORI GOOD — NOTA: IMPLICA VALIDITY
- STESSO PREFERRED VAL
- STESSO MAJORITY VAL CON TALLY $> G$
- TERMINATION COND. = TRUE $\square$

COROLLARY SE TERMINATION = TRUE PER P_i, LO SARÀ PER TUTTI AL ROUND SUCCESSIVO

PROOF

- SE P_i VEDE m_i CON $TALLY_i > G$
- OGNI GOOD P_j NEL PROSSIMO ROUND VEDRÀ m_i CON TALLY: $(tally_j - f) > H$
- QUINDI OGNI NON-FAULTY HA STESSO PREFERRED VAL
- DA LEMMA PRECEDENTE CLAIM FOLLOW $\square$

(2) OTHER CASES: NO P_i vede m_i CON $TALLY_i > G$

(2.1) DIFFERENT MAJs PER SOME PROCs

- $\exists P_i, P_k$ T/C $MAJ_j \neq MAJ_k$ PER SOME $i \neq k$

LEMMA SE NO P_i VEDE m_i CON $TALLY_i > G$, E $\exists P_i, P_k$ T/C $maj_i \neq maj_k$ ALLORA:

$$TALLY_i < L \quad , \quad TALLY_k < L$$

PROOF

- SUPPONIAMO $tally_j \geq L$
- ALLORA $TALLY_i - f > \frac{6m}{\delta} - \frac{1m}{\delta} > \frac{4m}{\delta} = m/2$

$\Rightarrow m/2$ PROCESSORI HANNO VOTATO maj_i

- QUINDI OGNI P_j VEDE LO STESSO MAJORITY $\Rightarrow m_i = m_k \; \forall P_i, P_k$ $\perp$ $\square$

(2.2) SAME MAJs $\forall$ PROCs

LEMMA ASSUMIAMO CHE IN UN ROUND NO P_i VEDE m_i CON $tally_j > G$, E CHE TUTTI P_j HANNO STESSO maj_j; ALLORA $\forall P_i, P_k$ T/C $i \neq k$:

$$|tally_j - tally_k| \leq f$$

Proof

- ASSUMIAMO $|tally_i - tally_k| > f$ PER QUALCHE $i \neq k$
- SUPPONIAMO f', f'' IL NUMERO DI CORRUPTED VALUES $(f', f'' \leq f)$
- SIANO $tally_i^+, tally_k^+$ IL NUMERO DI TRUSTED VALUES $(tally_i - f = tally_i^+)$

$$|tally_i - tally_k| = |tally_i^+ + f' - tally_k^+ - f''| = |f' - f''|$$

- DA HYPOTESI $|tally_i - tally_k| > f \Rightarrow |f' - f''| > f$ ⊥
- $f' \leq f$, $f'' \leq f$ NON possono FARE più di f ~~SOTTRATTI~~ SOTTRATTI □

-> SOTTOCASI DI 22:

$f \leq P$
- (1) $TALLY_{min} < L$ e THRESHOLD $= H$
- (2) " $\geq L$ " $= L$

} GOOD

$f \leq P$
- (3) $< L$ " $= L$
- (4) $\geq L$ " $= H$

} BAD

---

(1) $Tally_k \leq tally_i + f < L + f \leq 6n/8 = H \Rightarrow$ $V_i = V_k = 0$ quindi Termina al PROX ROUND

(2) $tally_k \geq tally_{min} \geq L \Rightarrow V_k = V_{min} = min_{min} \Rightarrow$ TUTTI LIVELLI quindi terminano al PROX ROUND

(3,4) NON TERMINA AL PROX ROUND

PERFORMANCE

- $O(\log n)$ ROUNDS : $1/2$ DI PROB. DI PROCEDERE AL ROUND SUCCESSIVO

-> Prob(Termina entro $\log n$) $\Rightarrow (1/2)^{\log n} = 1/n$

-> Prob terminazione = $1 - 1/n$

- $O(n^2)$ MSGS PER ROUND

-> $O(n^2 \log n)$ MSGS TOT.

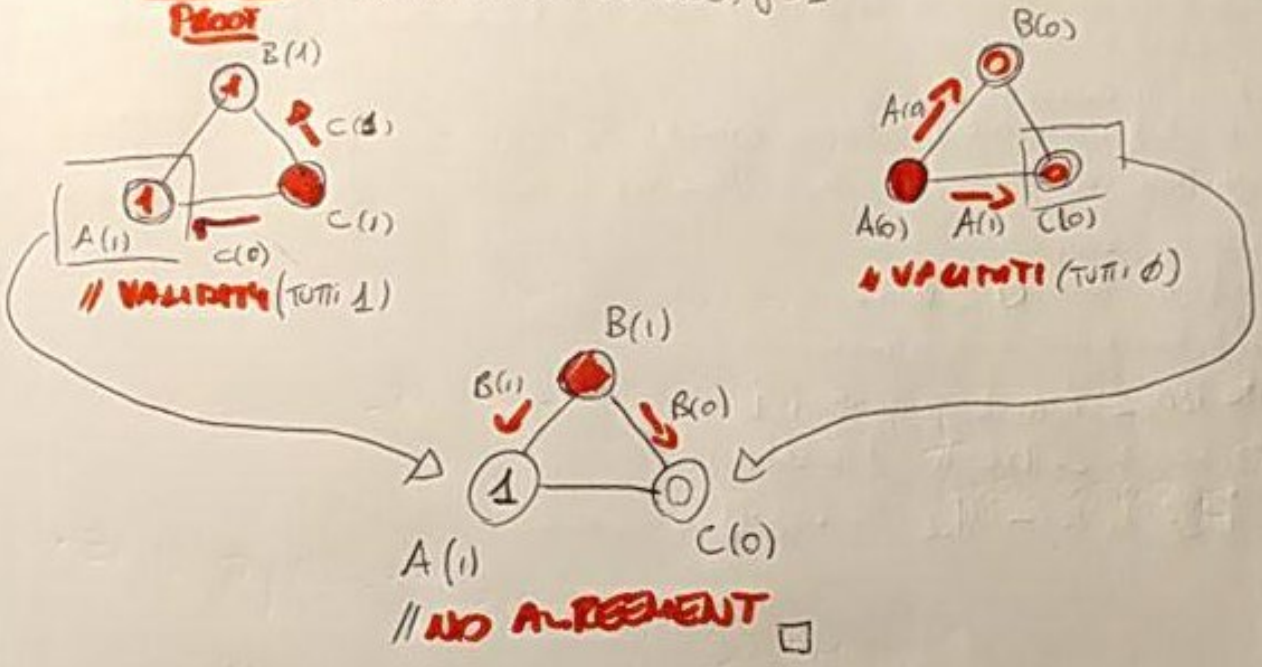
IMPOSSIBILITY RESULT

TH NON ESISTE f-RESILIENT BYE-FAIL ALGO PER n PROCS SE $f \geq n/3$

PROOF

LEMMA 1 L'ALGO NON ESISTE CON $n=3$, $f=1$

PROOF



□

- ASSUMIAMO ∃ ALGO A PER $f = n/3$ E $n > 3$
- USIAMO A PER COSTRUIRE UN ISTANZA CON $n=3$, $f=1$
- SIA $n = 3f$
- SIA $P = \langle P_0, P_1, \ldots, P_n \rangle$
- SIA $Q = \langle q_0, q_1, q_2 \rangle$ TC $q_i = \langle P_0, \ldots, P_{A/3} \rangle$ ...
- USIAMO A SU SOTTOINSIEME q_i
- ASSUMIAMO q_2 FALLISCE, PERCHÉ $P_i \in q_2$ BYE-FAIL
- SUPPONIAMO q_1, q_3 DANNO K

⇒ CONSENSO RAGGIUNTO CON $n=3$, $f=1$ ⊥ LEMMA 1

□

EXPONENTIAL TREE ALGO

- $n = 2f+1$ - $f+1$ #ROUNDS - EXP. # MSGS

- OGNI P_i HA UN TREE ASSOCIATO, IN OGNI NODO RAPPRESENTA P_j E CUI È ASSOCIATA UNA STR.

  - $h = f+1$ $\longrightarrow 2$
  - #children(ROOT) n $\longrightarrow 0, \dots, n-1$
  - #children(child(ROOT)) $n-1$ $\longrightarrow i:0, \dots, i:n-1$ TRANNE $i:i$
  - ⋮
  - #children(NODO AL LV d) $n-d$ $\longrightarrow i_1:i_2:\dots:i_d:0, \dots, i_1:i_2:\dots:i_d:i_{n-1}$
    $\hookrightarrow i_1:i_2:\dots:i_d$

  - AL LV $f+1$ $\longrightarrow i_1:i_2:\dots:i_{f+1}$
    $\hookrightarrow h = \phi$

- AL ROUND i RICEVUTO LV_i
- AL ROUND $f+1$ CALCOLATA DECISIONE (ATM-UP)

ROUND 1

- INPUT ALLA ROOT di P_j
- BROADCAST STATO A TUTTI I PROCESSI
- LV_1 STORE X INVIATO DA P_j NEL NODO j DI P_i

ROUND 2

P_j BROADCAST LV_2

- SIA $\{X_0, \dots, X_{n-1}\}$ INVIATO DA P_j

P_i SCARTA X_j E SALVA GLI ALTRI X_k NEL LV_2 NEL NODO $k:j$

ROUND $d \geq 2$

P_j BROADCAST LV_{d-1} $(n(n-1)\dots(n(d-2))$ MSGS)

- SIA X INVIATO DA P_k, P_i STORES X NEL LV_d NEL NODO $i_1:i_2:\dots:i_{d-1}:j$

ROUND $f+1$

- CALCOLA RICORSIVAMENTE $resolve(\tilde{n})$ PER OGNI NODO FINO ALLA RADICE $\tilde{n} = \lambda$

$$resolve(\tilde{n}) \begin{cases} \tilde{n} \text{ se } \tilde{n} \text{ foglia} \\ \text{MAGGIORE } resolve(\tilde{n}') \ \forall \ \tilde{n}' \text{ figlio } \tilde{n} \end{cases}$$

CONSISTENZA dei resolve (π)

LEMMA 1 Se $n \geq 3f$ e P_i, P_j NON-FAULTY:

- resolve($\pi_i = \pi_{ij}$) = resolve(π_j)

PROOF INDUZIONE SU h

(ϕ) π_{ij} FOGLIA, $h=0$

$\therefore$ LO P_i STORE IN $\pi_{ij} = \pi_{ji}$ QUELLO CHE P_j MANDATO AL ROUND $f+1$ CIOÈ π_{ji}

(IND) π_{ij} NON FOGLIA, $h>0$

LO π_{ij} HA $n-f$ FIGLI

$n > 3f \Rightarrow n-f > 2f \Rightarrow \pi_{ij}$ ha ℓ_h MAGGIORANZA DI NON-FAULTY CHILDREN

ULTIMO DIGIT NELLA LABEL È L'INDICE DI UN PROC NON-FAULTY

LO SIA $\pi_{ijk} = \pi_{jk}$ CHILD DI π_{ij} DI $h = h-1$

T/C P_k NON FAULTY ALLORA P_j (NON-FAULTY) MANDA CORRETTAMENTE IL VALORE v E P_k LO STORA CORRETTAMENTE IN $\pi_{ijk} = \pi_{jk}$

$\therefore$ PER INDUZIONE

P_i's resolve(π_{ij}) = v = resolve($\pi_{ijk} = \pi_{jk}$)

$\Rightarrow$ OGNI NON-FAULTY CHILD DI π risolve v, QUINDI π RISOLVE v

□

VALIDITY

- SUPPONIAMO OGNI INPUT DEI NON-FAULTY SIA V
- DAL LEMMA 1 SE T_j NON FAULTY $\Rightarrow$ T_i STESSA CHIAVE NE NODO J
- ESSENDO LA MAGGIORANZA (2k) NON-FAULTY: SE GLI INPUT SONO TUTTI UGUALI T_i's ROOT uguale V

AGREEMENT

DEF $\tilde{T}$ COMMON SE OGNI NON-FAULTY CONTIENE STESSO $result(\tilde{T})$

- LA ROOT È COMMON?

LEMMA 1 NON COPRIVA TUTTI I CASI, SOLO QUELLI CON ULTIMO NODO ROOT LABEL UGUALE A NON-FAULTY

DEF $\tilde{T}$ È COMMON-FRONT SE OGNI PATH DA $\tilde{T}$ NON FOGLIA HA UN COMMON NODO

LEMMA 2 SE $\tilde{T}$ HA COMM-FRONT È COMMON

PROOF INDUZIONE SU h

($\emptyset$) $\tilde{T}$ FOGLIA, $h=\emptyset$

↳ NEL PATH C'È SOLO $\tilde{T}$, DEVE ESSERE COMMON

(IND.) $\tilde{T}$ NON FOGLIA, $h>0$

↳ ASSUMIAMO ABBIA COMM-FRONT MA NON SIA COMMON

→ OGNI $\tilde{T}'$ FIGLIO DI $\tilde{T}$ HA COMM-FRONT

→ $\tilde{T}'$ HA $h=h-1$

⇒ PER INDUZIONE $\tilde{T}'$ COMMON

→ $\tilde{T}$ ALLORA SCEGLIE IL VAL. DEI FIGLI ⇒ COMMON ANCHE $\tilde{T}$

□

AGREEMENT

- ci sono $f+2$ nodi in un PATH ROOT-LEAF
- L'ultimo livello di ogni nodo nel path DISTINTO

  ⇒ ALMENO UN NODO NON_FAULTY

  ⇒ da LEMMA 1 COMMON

- SEGUE CHE ROOT HA COMMON FRONTIER QUINDI da LEMMA 2 COMMON

TERMINATION

- $f+1$ ROUND

COMPLESSITÀ

- ESPONENZIALE IN n

→ ROUND 1 : $O(n^2)$ // $O(n)\,O(n)$ (NON FAULTY PROCS, #msg)

→ ROUND $2 \le d \le f+1$: $O(n^{d+1})$ // $O(n \cdot n \cdot n(n-1)\dots(n-(d-2)))$

(n: NON FAULTY PROCS; n: PROCS A CUI BROADCASTARE; $n(n-1)\dots(n-(d-2))$: NODI AL LV $d-1$)

⇒ TOT $O(n^2) + O(n^3) + \dots + O(n^{f+2}) = O(n^{f+2})$

→ DATO CHE $f = O(n) \Rightarrow O(n^{n+2})$ EXP. IN n