



Local Distributed Computing

Pierre Fraigniaud



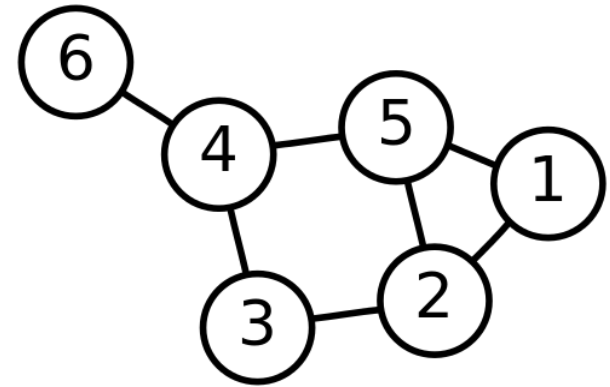
**INSTITUT
DE RECHERCHE
EN INFORMATIQUE
FONDAMENTALE**

École de Printemps en Informatique Théorique
Porquerolles 14-19 mai 2017

A 3D network diagram consisting of numerous nodes connected by black lines. The nodes are represented as spheres; some are blue and larger, while others are silver and smaller. The blue nodes are distributed across the network, with a few prominent ones in the foreground. The silver nodes are more numerous and form the bulk of the network structure. The entire network is set against a plain white background.

What can be computed locally?

LOCAL model



An abstract model capturing the **essence of locality**:

- Processors connected by a network $G=(V,E)$
- Each processor (i.e., each node) has an Identity
- Synchronous model (sequence of rounds)
- All processor start simultaneously
- No failures — all processors

Complexity as #rounds

At each round, each node:

- **Sends** messages to neighbors
- **Receives** messages from neighbors
- **Computes**

#rounds measures locality

t-round Algorithm **A**:

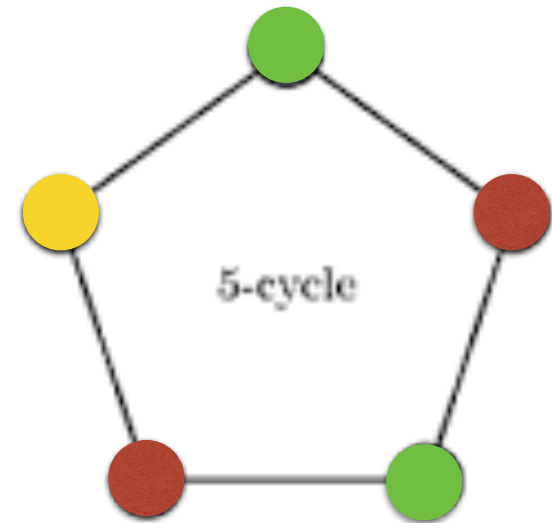
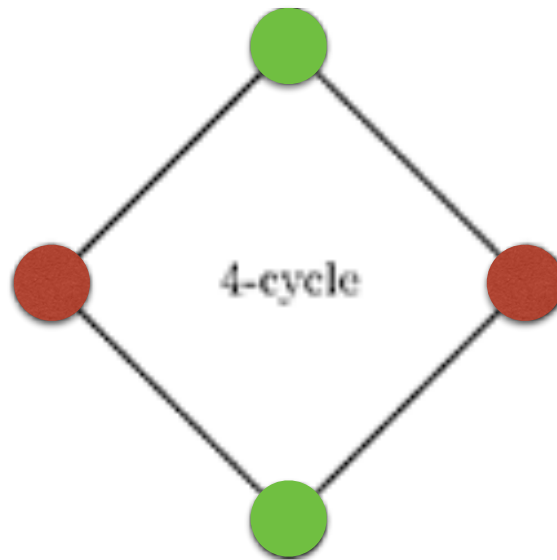
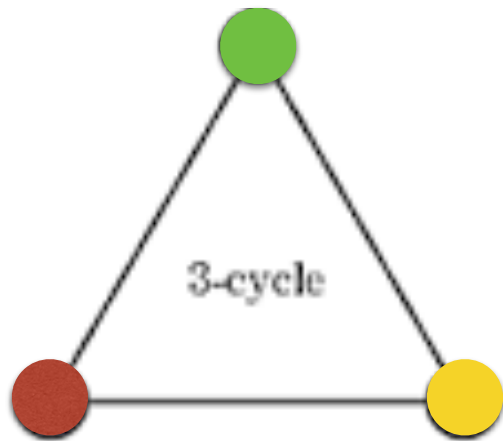
```
    byte[] data = new byte[1024]; string log;
    TcpClient server;
    try {
        server = new TcpClient("...", "8080");
    } catch (SocketException) {
        Console.WriteLine("Unable to connect");
        return;
    }
    NetworkStream ns = server.GetStream();
    int recv = ns.Read(data, 0, data.Length);
    stringData = Encoding.ASCII.GetString(data, 0, recv);
    Console.WriteLine(stringData);
    while (true) {
        input = Console.ReadLine();
        if (input == "exit") break;
        ProcessStartInfo newchild = new ProcessStartInfo("exit");
        Process newchild = Process.Start(newchild);
    }
}
```

Algorithm **B**:

1. Gather all data at distance at most **t** from me
2. Individually simulate the **t** rounds of **A**

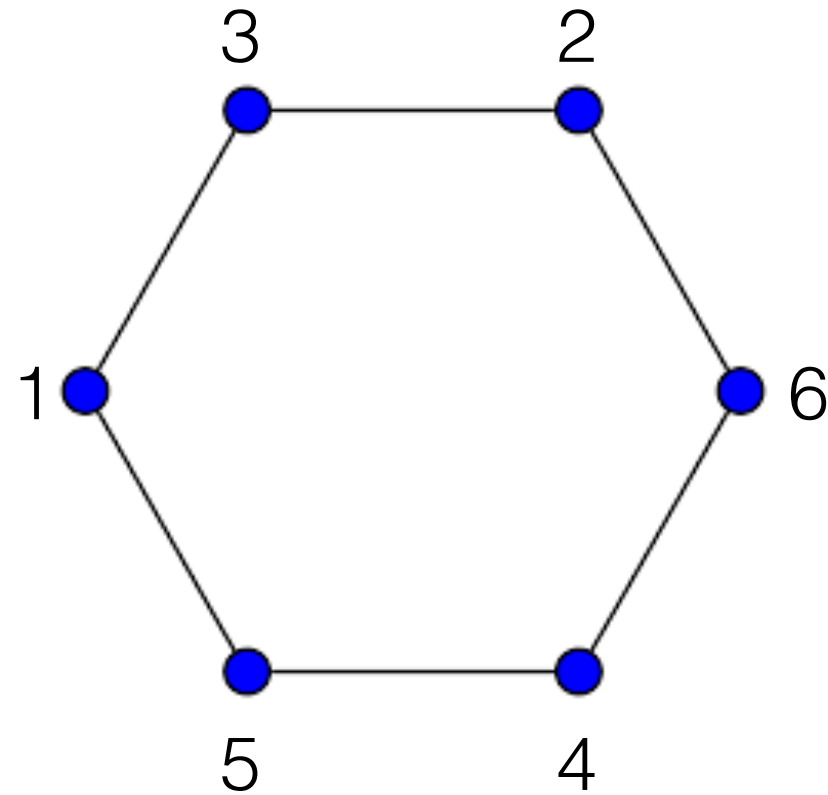
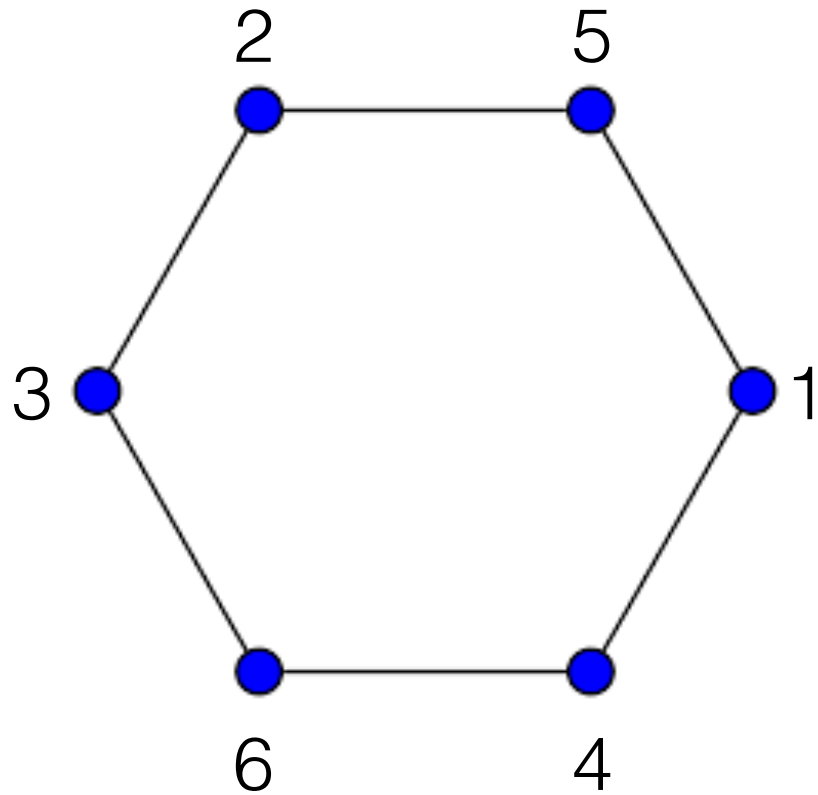
A Case Study: Distributed Coloring

3-coloring cycles

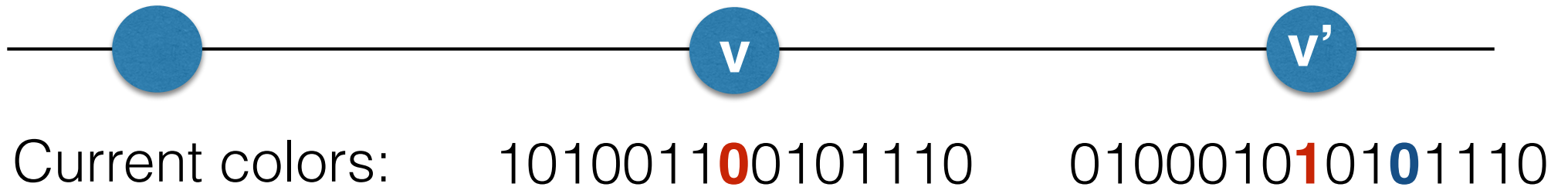


- Symmetry-breaking task
- Application to frequency assignment in radio networks

Instances: same graph, but different ID-assignments



Cole & Vishkin (1986)



b = bit-value

k = bit-position

$$\text{new color} = (k, b) = 2k + b \quad (k', b')$$

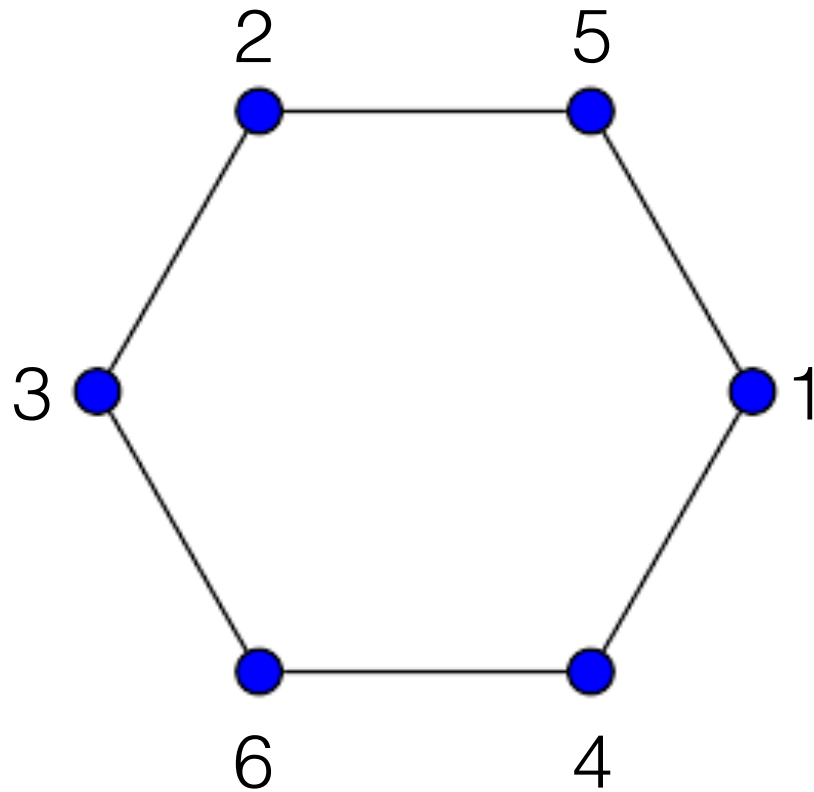
$$c(v) = c(v') \Rightarrow (k, b) = (k', b')$$

Complexity of Cole-Vishkin

- current colors on B bits
- new colors on $\lceil \log B \rceil + 1$ bits
- Iterated logarithms:
 - $\log^{(1)} x = \log x$
 - $\log^{(k+1)} x = \log \log^{(k)} x$
- $\log^* x = \min \{ k : \log^{(k)} x < 1 \}$

Cole-Vishkin: **$O(\log^* n)$** rounds

Linial Lower Bound (1992)



Distance-1 neighborhoods:
(2,5,1) (4,6,1) (5,1,4)

(2,5,1) consistent with (5,1,4)
(2,5,1) not consistent with (4,6,1)

Configuration graph $G_{n,1}$

- Nodes = distance-1 neighborhood
- Edges = between consistent neighborhoods

Configuration graph $G_{n,t}$

Definition

- node = $(x_0 \ x_1 \ \dots \ x_{t-1} \ \mathbf{x}_t \ x_{t+1} \ x_{t+2} \ \dots \ x_{2t})$
= a view of x_t at distance t in some cycle
- edge = $\{(x_0 \ \dots \ x_{t-1} \ \mathbf{x}_t \ x_{t+1} \ \dots \ x_{2t}), (x_1 \ \dots \ x_t \ \mathbf{x}_{t+1} \ x_{t+2} \ \dots \ x_{2t} \ y)\}$

Chromatic number $X(G) =$ minimum #colors to proper color G

Lemma Algorithm in t -rounds for k -coloring $C_n \Rightarrow X(G_{n,t}) \leq k$

2-coloring C_{2k}

Theorem 2-coloring C_{2k} requires at least $k-1$ rounds

Proof If $t \leq k-2$ then there exists an odd-cycle in $G_{2k,t}$

- $(X_0 X_1 \dots X_{2k-4})$
 - $(X_1 \dots X_{2k-4} Y)$
 - $(X_2 \dots X_{2k-4} Y Z)$
 - $(X_3 \dots X_{2k-4} Y Z X_0)$
 - $(X_4 \dots X_{2k-4} Y Z X_0 X_1)$
 - ...
 - $(X_{2k-4} Y Z X_0 \dots X_{2k-7})$
 - $(Y Z X_0 \dots X_{2k-6})$
 - $(Z X_0 \dots X_{2k-5})$
- $(2k-1)$ -cycle



3-coloring C_n

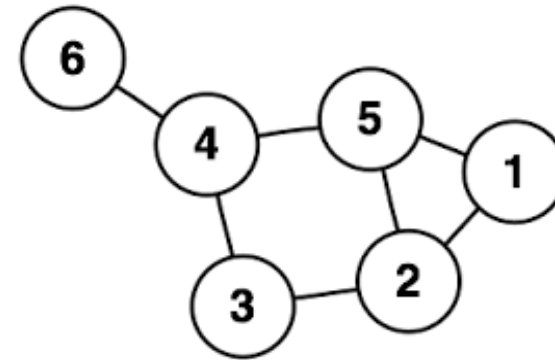
Theorem 3-coloring C_n requires $\Omega(\log^*n)$ rounds

Proof Show that if $t = o(\log^*n)$ then $X(G_{n,t}) = \omega(1)$ ■

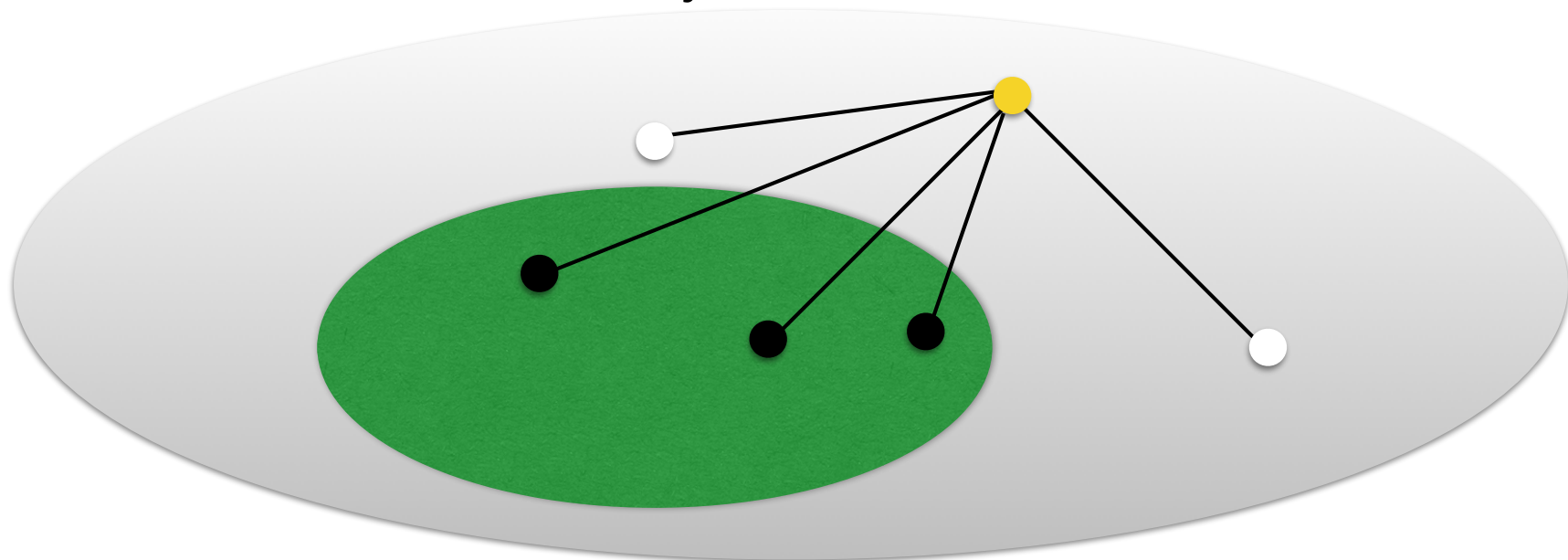
$(\Delta + 1)$ -coloring

Δ = maximum degree

For every graph G , $\chi(G) \leq \Delta + 1$



Greedily constructible



Complexity of $(\Delta+1)$ -coloring as a function of n

Theorem (Panconesi & Srinivasan, 1995)

$(\Delta+1)$ -coloring algorithm in $2^{O(\sqrt{\log n})}$ rounds

Theorem (Linial, 1992)

$(\Delta+1)$ -coloring requires $\Omega(\log^* n)$ rounds

Complexity of $(\Delta+1)$ -coloring as a function of n and Δ

| | |
|-----------------------------------------------------------------------|----------------------------------------------------------|
| Linial (1992) cf. also Goldberg, Plotkin and Shannon (1988) | $O(\log^*n + \Delta^2)$ |
| Szegedy & Vishwanathan (1993) | $\Omega(\Delta \log \Delta)$ for iterative algorithms |
| Kuhn & Wattenhofer (2006) | $O(\log^*n + \Delta \log \Delta)$ iterative |
| Barenboim & Elkin (2009) Kuhn (2009) | $O(\log^*n + \Delta)$ |
| Barenboim (2015) | $O(\log^*n + \Delta^{3/4})$ |
| F., Heinrich & Kosowski (2016) | $O(\log^*n + \sqrt{\Delta})$ |

Randomized algorithm for $(\Delta + 1)$ -coloring

Algorithme distribué de $(\Delta + 1)$ -coloration pour un sommet u :

début

$c(u) \leftarrow \perp$

$C(u) \leftarrow \emptyset$

tant que $c(u) = \perp$ **faire**

choisir une couleur $\ell(u) \in \{0, 1, \dots, \Delta + 1\} \setminus C(u)$ avec

$\Pr[\ell(u) = 0] = \frac{1}{2}$, et $\Pr[\ell(u) = \ell] = \frac{1}{2(\Delta + 1 - |C(u)|)}$ pour $\ell \in \{1, \dots, \Delta + 1\} \setminus C(u)$

envoyer $\ell(u)$ aux voisins et recevoir la couleur $\ell(v)$ de chaque voisin v

si $\ell(u) \neq 0$ et $\ell(v) \neq \ell(u)$ pour tout voisin v **alors** $c(u) \leftarrow \ell(u)$ **sinon** $c(u) \leftarrow \perp$

envoyer $c(u)$ aux voisins et recevoir la couleur $c(v)$ de chaque voisin v

ajouter à $C(u)$ les couleurs des voisins v tels que $c(v) \neq \perp$

fin.

Analysis

$$\begin{aligned}\Pr[u \text{ termine}] &= \Pr[\ell(u) \neq 0 \text{ et aucun } v \in N(u) \text{ satisfait } \ell(v) = \ell(u)] \\ &= \Pr[\forall v \in N(u), \ell(v) \neq \ell(u) \mid \ell(u) \neq 0] \cdot \Pr[\ell(u) \neq 0] \\ &= \frac{1}{2} \cdot \Pr[\forall v \in N(u), \ell(v) \neq \ell(u) \mid \ell(u) \neq 0]\end{aligned}$$

$$\begin{aligned}\Pr[\ell(v) = \ell(u) \mid \ell(u) \neq 0] &= \Pr[\ell(v) = \ell(u) \mid \ell(u) \neq 0 \wedge \ell(v) = 0] \Pr[\ell(v) = 0] \\ &\quad + \Pr[\ell(v) = \ell(u) \mid \ell(u) \neq 0 \wedge \ell(v) \neq 0] \Pr[\ell(v) \neq 0] \\ &= \Pr[\ell(v) = \ell(u) \mid \ell(u) \neq 0 \wedge \ell(v) \neq 0] \Pr[\ell(v) \neq 0] \\ &\leq \frac{1}{2} \Pr[\ell(v) = \ell(u) \mid \ell(u) \neq 0 \wedge \ell(v) \neq 0] \\ &= \frac{1}{2} \frac{1}{\Delta + 1 - |C(u)|}.\end{aligned}$$

$$\Pr[\exists v \in N(u) : \ell(v) = \ell(u) \mid \ell(u) \neq 0] \leq (\Delta - |C(u)|) \frac{1}{2(\Delta + 1 - |C(u)|)} < \frac{1}{2}$$

Analysis (continued)

Theorem (Barenboin & Elkin, 2013) The randomized algorithm performs $(\Delta+1)$ -coloring in $O(\log n)$ rounds, with high probability.

Proof $\Pr[u \text{ terminates at a given round}] > 1/4$

$\Pr[u \text{ has not terminated in } k \ln(n) \text{ rounds}] < (3/4)^{k \ln(n)}$

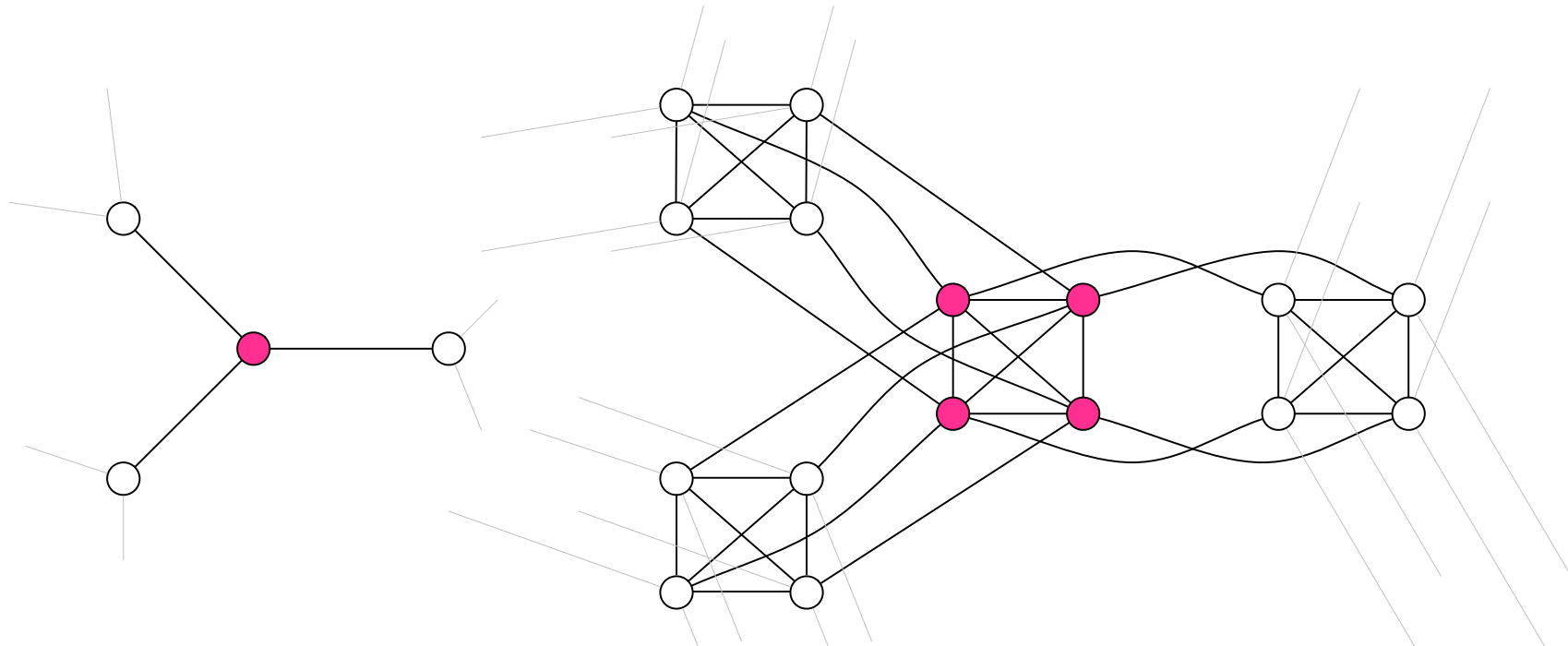
$\Pr[\text{some } u \text{ has not terminated in } k \ln(n) \text{ rounds}] < n (3/4)^{k \ln(n)}$

Pick $k = 2/\ln(4/3)$

$\Pr[\text{all nodes have terminated in } k \ln(n) \text{ rounds}] \geq 1 - 1/n$ 

Complexity of randomized $(\Delta + 1)$ -coloring

| | |
|------------------------------------------------------|-----------------------------------------------------|
| Alon, Babai & Itai (1986) Luby (1986) | $O(\log n)$ |
| Harris, Schneider & Su (2016) | $O(\sqrt{\log \Delta} + 2^{O(\sqrt{\log \log n})})$ |



Locally Checkable Labelings (LCL)

Distributed Languages

- *Configuration*: (G, λ) where $\lambda : V(G) \rightarrow \{0, 1\}^*$
- λ is called a *labeling*, and $\lambda(u)$ is the *label* of node u
- A *distributed language* is a collection of configurations
- Examples:
 - $L = \{(G, \lambda) : G \text{ is planar}\}$
 - $L = \{(G, \lambda) : \lambda \text{ is a proper coloring of } G\}$
 - $L = \{(G, \lambda) : \lambda \text{ encodes a spanning tree of } G\}$

Distributed decision

A distributed algorithm A *decides* L if and only if:

- $(G, \lambda) \in L \Rightarrow$ all nodes output *accept*
- $(G, \lambda) \notin L \Rightarrow$ at least one node output *reject*

The class LCL (locally checkable labelings)

Definition LCL is the class of distributed languages on graphs with

- bounded maximum degree $\Delta = O(1)$, and
- labels on bounded size $k = O(1)$

for which the membership to the language can be decided in $O(1)$ rounds.

LCL Construction Task

$L \in \text{LCL}$

Task: Given G , construct λ such that $(G, \lambda) \in L$

Example: Given C_n construct a 3-coloring of C_n

Theorem (Naor & Stockmeyer, 1995)

Constant #rounds construction is TM-undecidable
even for LCL

On the power of randomization

Theorem (Naor & Stockmeyer, 1995)

Let $L \in LCL$. If there exists a *randomized* Monte- Carlo construction algorithm for L running in $O(1)$ rounds, then there exists a *deterministic* construction algorithm for L running in $O(1)$ rounds.

Order-invariance: depend on the relative order of the IDs, not on their actual values.

Lemma If there exists a t -round construction algorithm for L , then there is t -round *order-invariant* construction algorithm for L .

Proof of the lemma (1/5)

Assumption IDs in \mathbb{N} (i.e., unbounded)

- Let X be a countably infinite set
- $X^{(r)}$ = set of all subsets of X with size exactly r
- Let $c : X^{(r)} \rightarrow \{1, \dots, s\}$ be a “coloring” of the sets in $X^{(r)}$.

Theorem (Ramsey) There exists an infinite set $Y \subseteq X$ such that all sets in $Y^{(r)}$ are colored the same by c .

Proof (2/5)

- \mathcal{B} = collection of all graphs isomorphic to *some* ball $B_G(v,t)$ of radius t , centered at *some* node v in *some* graph G with maximum degree Δ .
- β = #pairwise non-isomorphic balls in \mathcal{B} .
- Enumerate balls from 1 to β
- Let n_i = #vertices in the i^{th} ball.
- Vertices of the i^{th} ball can be ordered in $n_i!$ different manners.
- Let $N = \sum_{i=1, \dots, \beta} n_i!$ ordered balls
- Enumerate these ordered balls in arbitrary order: B_1, \dots, B_N

Proof (3/5)

Let $\mathbb{N} = X_0 \supseteq X_1 \supseteq \dots \supseteq X_j$ such that, for all $1 \leq i \leq j$, the output of **A** at the center of B_i is the same for all possible IDs in B_i with values in X_i respecting the ordering of the nodes in B_i .

Define the coloring $c : X^{(r)} \rightarrow \{0, 1\}^k$ where $r = |B_{j+1}|$, as follows

1. For $S \in X(r)$, assign r pairwise distinct identities to the nodes of B_{j+1} using the r values in S , and respecting the order in B_{j+1} .
2. Define $c(S)$ as the output of **A** at the center of B_{j+1} .

By [Ramsey's Theorem](#), there exists an infinite set $Y_j \subseteq X_j$ such that all r -element sets $S \in Y(r)$ are given the same color.

- Set $X_{j+1} = Y_j$.
- Exhaust all balls B_i , $i = 1, \dots, N$, and set $\mathbf{I} = X_N$.

Proof (4/5)

I satisfies that, for every ball B_i the output of **A** at the center of B_i is the same for all ID assignments to the nodes of B_i with IDs taken from **I** and assigned to the nodes in respecting the order of B_i .

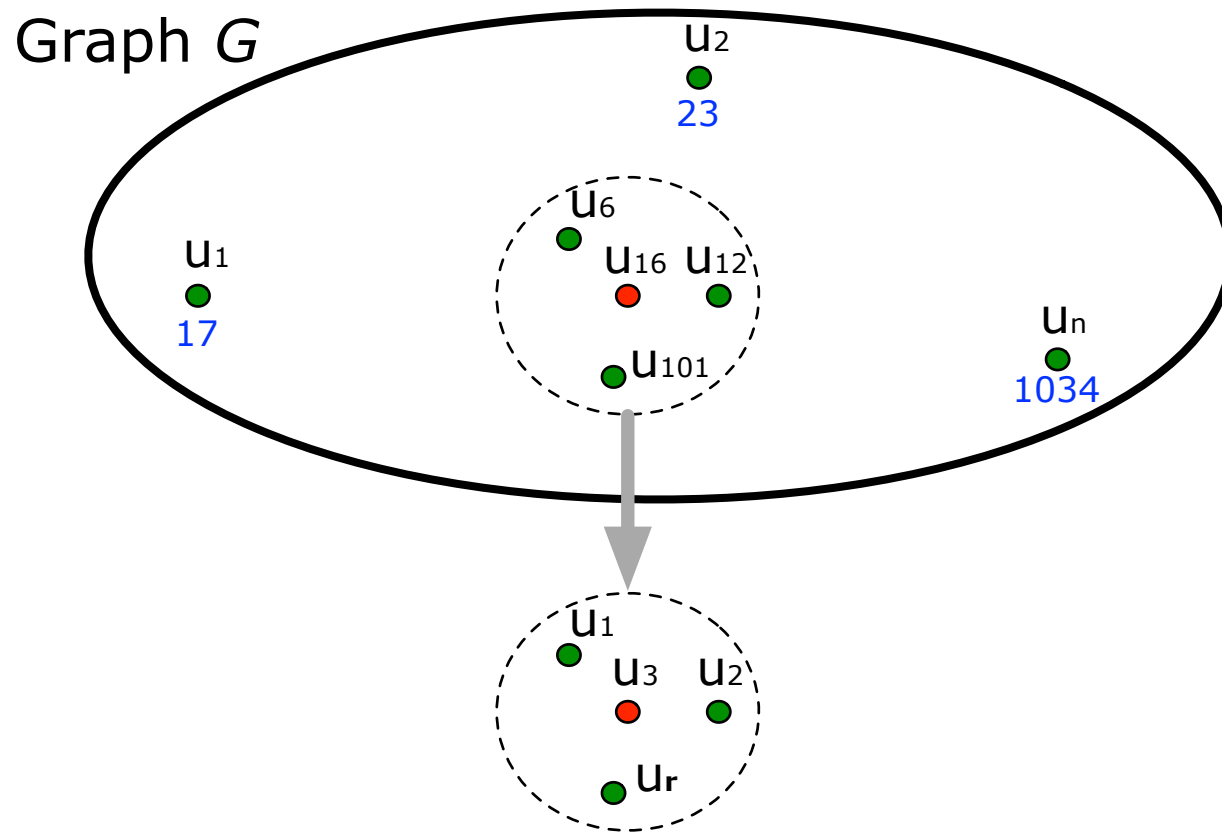
Order-invariant algorithm **A'**

1. Every v inspects its radius- t ball $B_G(v,t)$ in G . Let σ be the ordering of the nodes in $B_G(v,t)$ induced by their identities
2. Node v simulates **A** by reassigning identities to the nodes of $B_G(v,t)$ using the $r = |B_G(v,t)|$ smallest values in **I**, in order σ
3. Node v outputs what would have outputted **A** if nodes were given these identities.

Remark A' is well defined, and order-invariant.

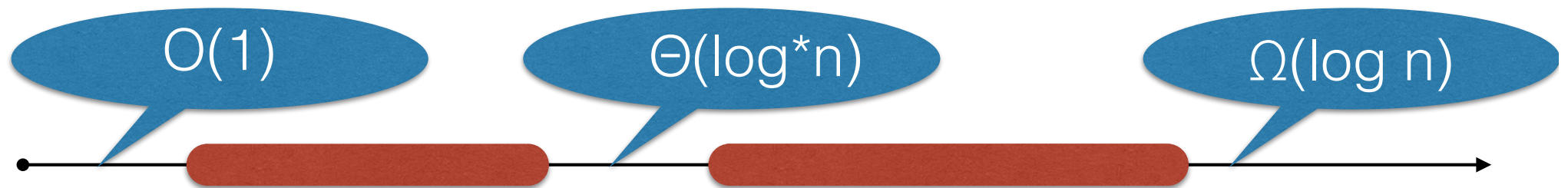
Proof (5/5)

A' is correct:



The three regimes for LCL construction tasks (in bounded-degree graphs)

Deterministic:



Randomized:



Local Decision

Decision classes

LD = class of distributed languages that can be decided in $O(1)$ rounds

PBLD (bounded probability local decision) = class of languages that can be probabilistically decided in $O(1)$ rounds:

- $(G, \lambda) \in L \Rightarrow \Pr[\text{all nodes output } \textit{accept}] \geq \frac{2}{3}$
- $(G, \lambda) \notin L \Rightarrow \Pr[\text{at least one node output } \textit{reject}] \geq \frac{2}{3}$

Generalization of Naor & Stockmeyer derandomization

Remark The previous proof for the order invariance lemma does not need $L \in LCL$

Theorem (Feuillley & F., 2015)

Let $L \in BPLD$. If there exists a *randomized* Monte-Carlo construction algorithm for L running in $O(1)$ rounds, then there exists a *deterministic* construction algorithm for L running in $O(1)$ rounds.

Deciding the presence of subgraphs

H is a *subgraph* of $G \iff V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$

G is H -free $\iff H$ is not a subgraph of G

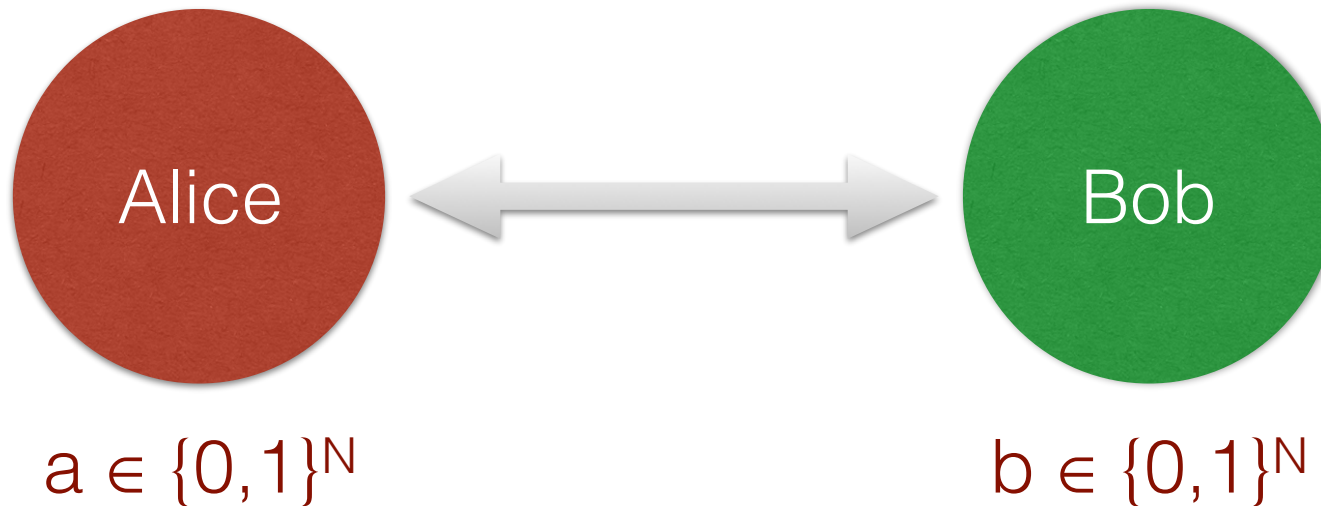
Remark Deciding H -freeness can be done in $diam(H)$ rounds

What about the message length?

Theorem (Drucker, Kuhn & Oshman, 2014) Deciding C_4 -freeness required sending $\Omega(\sqrt{n})$ bits between some neighbors

Communication complexity

$$f : \{0,1\}^N \times \{0,1\}^N \rightarrow \{0,1\}$$



Alice & Bob must compute $f(a,b)$

How many bits need to be exchanged between them?

Set-disjointness

- Ground set S of size N
- Alice gets $A \subseteq S$, and Bob gets $B \subseteq S$

$$f(A,B) = 1 \iff A \cap B = \emptyset$$

Theorem $CC(f) = \Omega(N)$, even using randomization.

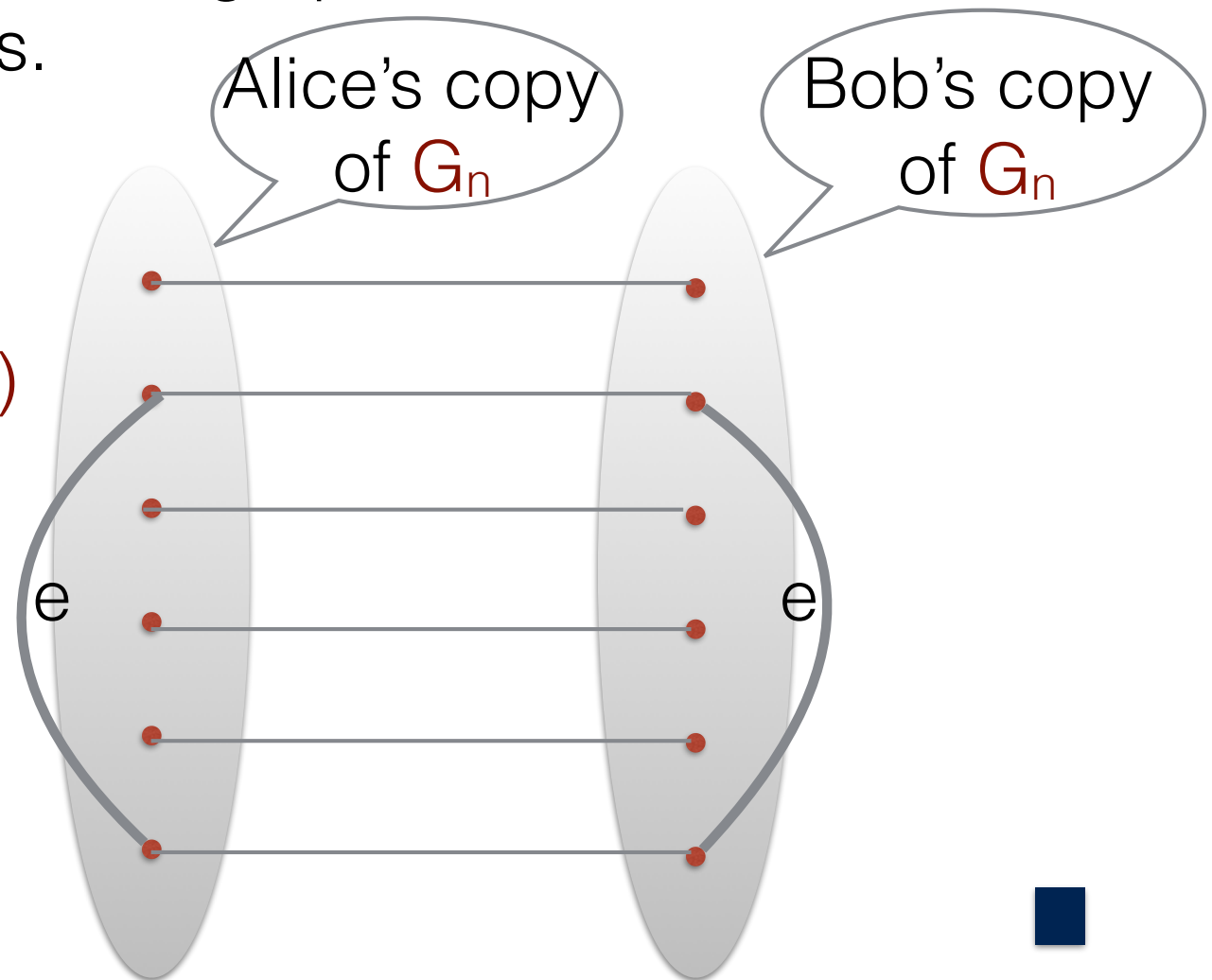
Reduction from Set-Disjointness

Lemma There are C_4 -free graphs G_n with n nodes and $m = \Omega(n^{3/2})$ edges.

Let A and B as in set-disjointness ($N=m$)

- Alice keeps $e \in E(G_n)$ iff $e \in A$
- Bob keeps $e \in E(G_n)$ iff $e \in B$

$$\Omega(n^{3/2})/n = \Omega(\sqrt{n})$$



The bound is tight

Algorithm 3 C_4 -detection executed by node u .

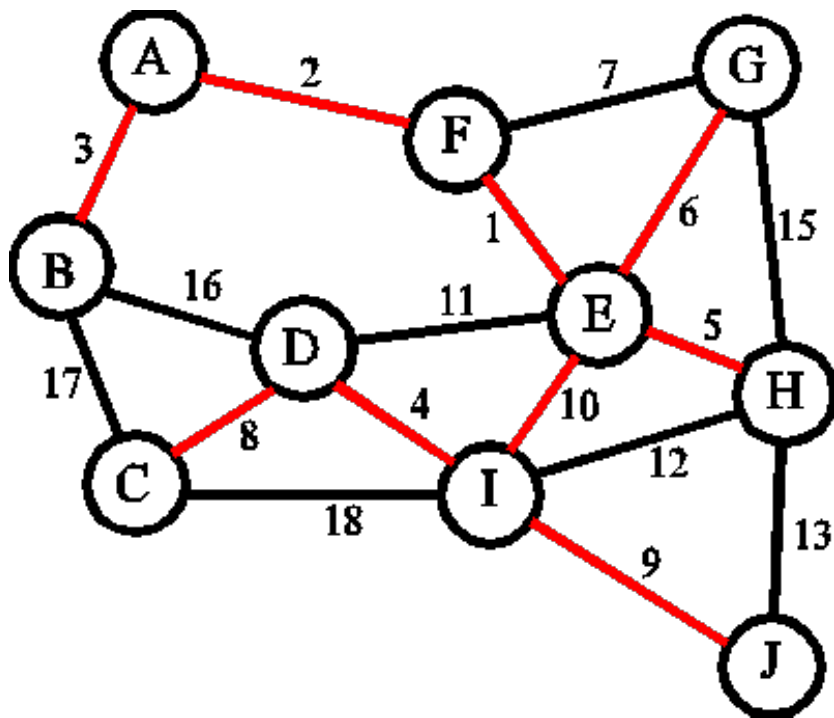
```
1: send  $ID(u)$  to all neighbors, and receive  $ID(v)$  from every neighbor  $v$ 
2: send  $\deg(u)$  to all neighbors, and receive  $\deg(v)$  from every neighbor  $v$ 
3:  $S(u) \leftarrow \{\text{IDs of the } \min\{\sqrt{2n}, \deg(u)\} \text{ neighbors with largest degrees}\}$ 
4: send  $S(u)$  to all neighbors, and receive  $S(v)$  from every neighbor  $v$ 
5: if  $\sum_{v \in N(u)} \deg(v) \geq 2n + 1$  then
6:   output reject
7: else
8:   if  $\exists v_1, v_2 \in N(u), \exists w \in S(v_1) \cap S(v_2) : w \neq u$  and  $v_1 \neq v_2$  then
9:     output reject
10:  else
11:    output accept
12:  end if
13: end if
```

Local Verification and Beyond

Deciding Spanning Trees

$ST = \{(G, \lambda) : \lambda \text{ encodes a spanning tree of } G\}$

$$\lambda(u) = \text{ID}(\text{parent}(u))$$



- $ST \notin LD$
- $ST \notin PBLD$

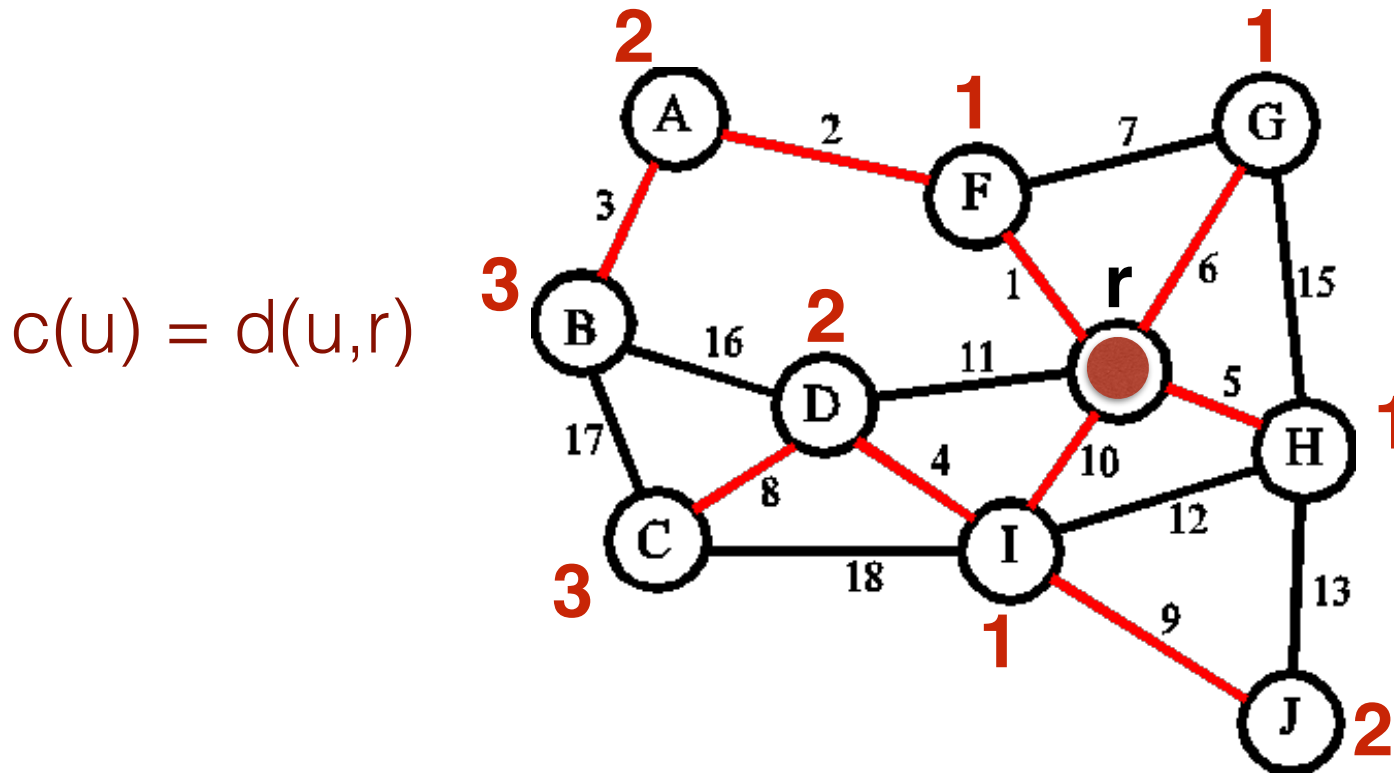
Non-deterministic Local Decision (NLD)

$L \in \text{NLD}$ iff there exists a distributed algorithm taking a pair label-certificate $(\lambda(u), c(u))$ at every node u such that:

- $(G, \lambda) \in L \Rightarrow \exists c : V(G) \rightarrow \{0, 1\}^*$ for which all nodes output *accept*
- $(G, \lambda) \notin L \Rightarrow \forall c : V(G) \rightarrow \{0, 1\}^*$ at least one node outputs *reject*

Applications: Fault-tolerance, self-stabilization, etc.

Example: (Spanning) Tree



certificates may depend on IDs

- Tree \in NLD
- Spanning tree \notin NLD but has a *proof-labeling scheme*

Beyond NLD

NLD: $(G, \lambda) \in L \iff \exists c : V(G) \rightarrow \{0, 1\}^* : \mathbf{A}$ accepts

NLD = Σ_1

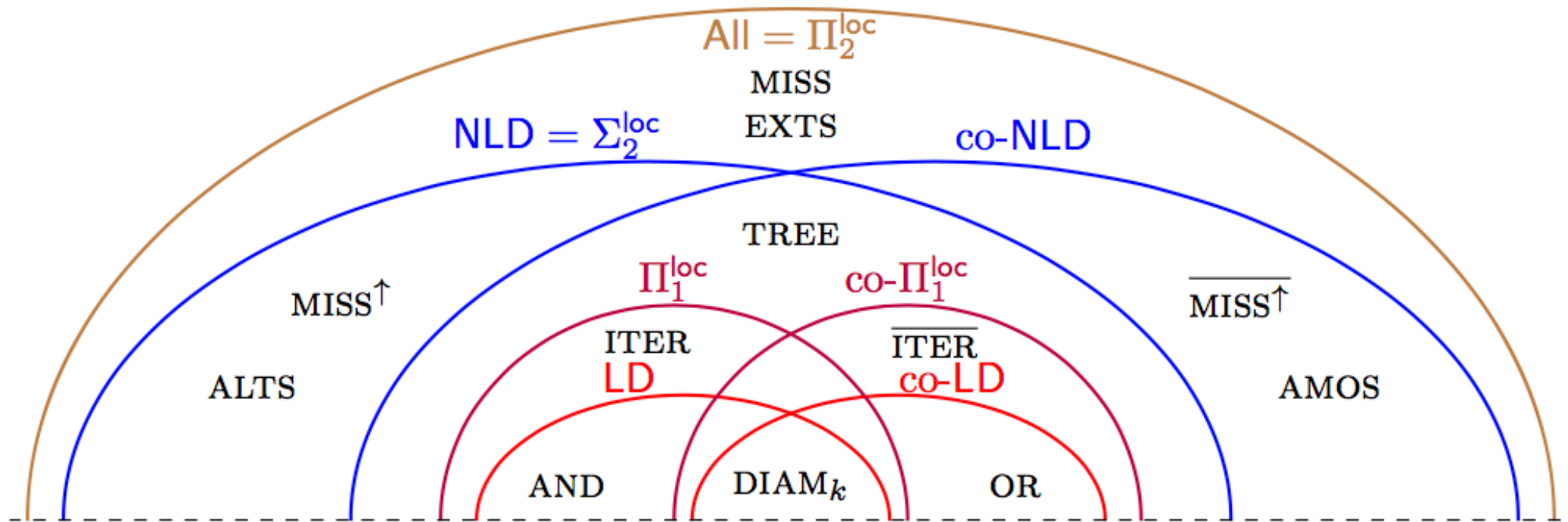
Π_1 : $(G, \lambda) \in L \iff \forall c : V(G) \rightarrow \{0, 1\}^* : \mathbf{A}$ accepts

Σ_2 : $(G, \lambda) \in L \iff \exists c \forall c' : \mathbf{A}$ accepts

Π_2 : $(G, \lambda) \in L \iff \forall c \exists c' : \mathbf{A}$ accepts

Local hierarchy: (Σ_k, Π_k) for $k \geq 0$ with $\Sigma_0 = \Pi_0 = \text{LD}$

Landscape of distributed decision



From Balliu, D'Angelo, F., Olivetti (2016)

Certificate size (upper bound)

Theorem (Korman, Kutten & Peleg) Every (TM-decidable) language with k -bit labels has a *proof-labeling scheme* (Σ_1) with certificates of size $\tilde{O}(n^2+nk)$ bits

- Certificate(u) = (M, Λ, \mathbf{I})
- Verification algorithm checks consistency of certificates

certificates may depend on IDs

Certificate size (Lower bound)

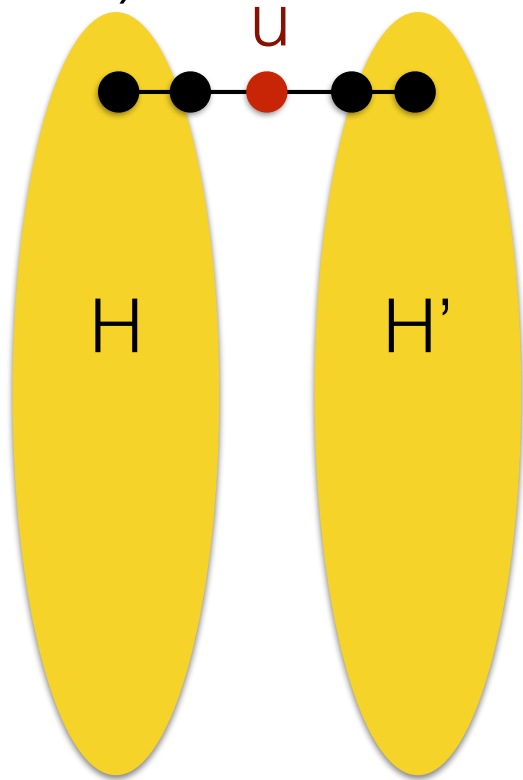
Theorem (Göös & Suomela) There exists a language with k -bit labels for which any proof-labeling scheme requires certificates of size $\Omega(n^2+nk)$ bits

Automorphism is a one-to-one label-preserving mapping $f : V(G) \rightarrow V(G)$ such that:
 $\{u,v\} \in E(G) \iff \{f(u),f(v)\} \in E(G)$

$L = \{(G,\lambda) : (G,\lambda) \text{ has a non-trivial automorphism}\}$

Non-trivial automorphism requires large certificates

$G=(H,H')$



There are $\sim 2^{n^2}$
 n -node graphs with no
non-trivial automorphisms

if $o(n^2)$ -bit certificates then consider
 (H_1, H'_1) and (H_2, H'_2) with the same
certificate at u

Consider (H_1, H'_2) : no nodes see any difference! ■

$O(\log n)$ -bit certificates

[Feuilleley, F., Hirvonen]

There are languages outside the local hierarchy $(\Sigma_k, \Pi_k)_{k \geq 0}$

‘Last for-all’ quantifier is of no help:

$$\Sigma_{2k} = \Sigma_{2k-1} \text{ and } \Pi_{2k+1} = \Pi_{2k}$$

Hierarchy: $\Lambda_{2k} = \Pi_{2k}$ and $\Lambda_{2k+1} = \Sigma_{2k+1}$

- Separation: $\Lambda_1 \neq \Lambda_0$; $\Lambda_2 \neq \Lambda_1$; $\Lambda_3 ? \Lambda_2$
- Collapsing: if $\Lambda_{k+1} = \Lambda_k$ then hierarchy collapses at Λ_k

Conclusion

Research directions

- Characterizing locality
- Interplay between decision and construction
- Incorporating errors, selfishness, and misbehaviors
- Many core-problems, like $(\Delta+1)$ -coloring, MIS, etc. are still open
- Incorporating the access to non-classical resources, e.g., entangled particles

Thank you!