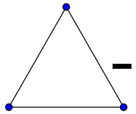
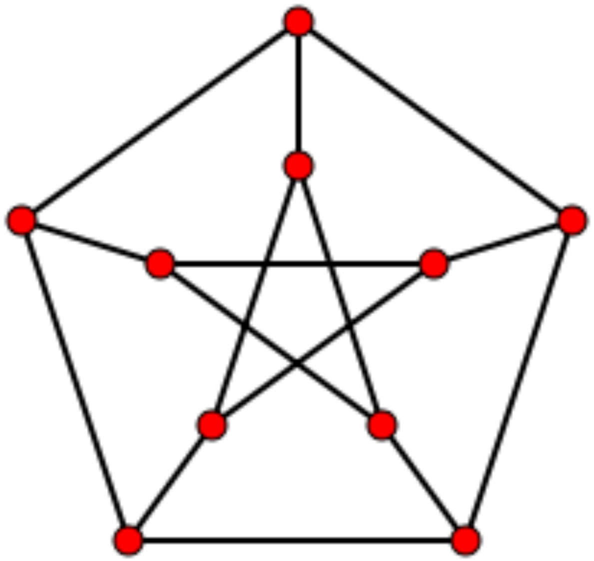
Exercices
(session 2)

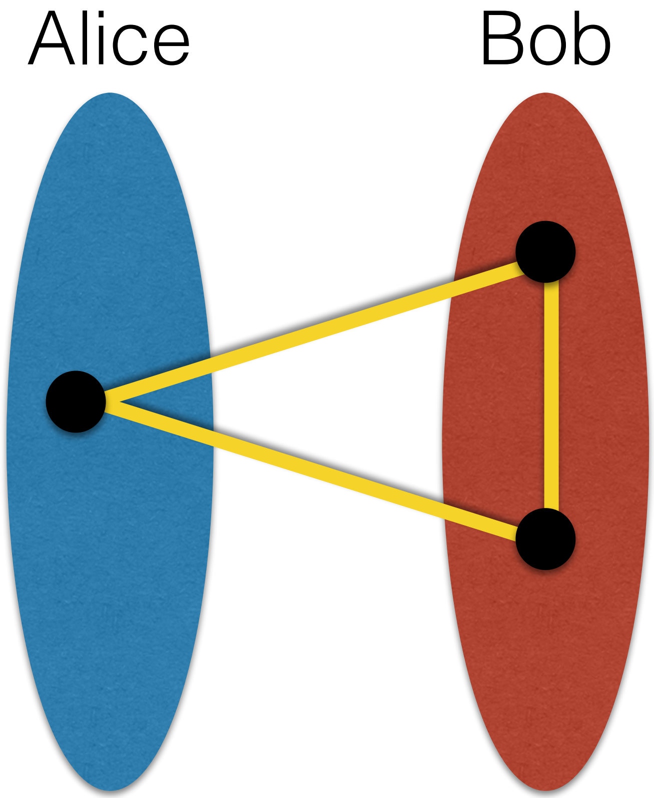
Exercice 1

C_3-freeness

deciding  -freeness



C_3-free graph



communication
complexity fails

Distributed Property Testing

- **Property testing:** checking correctness of large data structure, by performing small (sub-linear) amount of queries.

- Graph queries (with nodes labeled from 1 to n):
    - what is degree of node x?
    - what is the i^{th} neighbor of node x?

- Two relaxations:
    - G is ε-far from satisfying ϕ if removing/adding up to εm edges to/from G results in a graph which does not satisfy ϕ.
    - algorithm A tests ϕ if and only if:
        - $G \models \phi \Rightarrow$ Pr[all nodes output accept] $\geq 2/3$
        - $G \not\models \phi \Rightarrow$ Pr[at least one node outputs reject] $\geq 2/3$

Question 1. Design a randomized algorithm which detects any triangle with probability $\geq 1/n$.

Testing C_3-freeness

Algorithm of node u

Exchange IDs with neighbors
for every neighbor v do
    pick a received ID u.a.r.
    send that ID to v
if u receives ID(w) from $v \in N(u)$ with $w \in N(u)$ and $v \neq w$
then output reject
else output accept

Lemma 1 For any triangle Δ, $\Pr[\Delta \text{ is detected}] \geq 1/n$

Question 2 Show that if G is ε-far from being C_3-free, then G contains at least εm/3 edge-disjoint triangles.

Analysis

Lemma 2 If G is ε-far from being C_3-free, then G contains at least εm/3 edge-disjoint triangles.

Proof Let $S=\{e_1,e_2,\ldots,e_k\}$ be min #edges to remove for making G triangle-free ($k \geq \varepsilon m$).

Repeat removing e from S, as well as all edges of a triangle Δ_e containing e ➡ at least k/3 steps.

All triangles Δ_e are edge-disjoint.

❏

Question 3 Let $\varepsilon \in]0,1[$. Show that if G is ε-far from being C_3-free, then a constant number of repetition of the algorithm detects a cycle with probability at least $1-(1/e)^{\varepsilon/3}$

Analysis (continued)

Theorem Let $\varepsilon \in]0,1[$. If G is ε-far from being C_3-free, then a constant number of repletion of the algorithm detects a cycle with probability $\geq 1-(1/e)^{\varepsilon/3}$

Proof (of theorem)

- $\Pr[\text{no } \Delta \text{ detected}] \leq (1-1/n)^{\varepsilon m/3} \leq (1-1/n)^{\varepsilon n/3}$
- $(1-1/n)^n \approx 1/e$
- $\Pr[\text{no } \Delta \text{ detected}] \leq (1/e)^{\varepsilon/3}$

Repeat k times with k such that $(1/e)^{\varepsilon k/3} \leq 1/3$

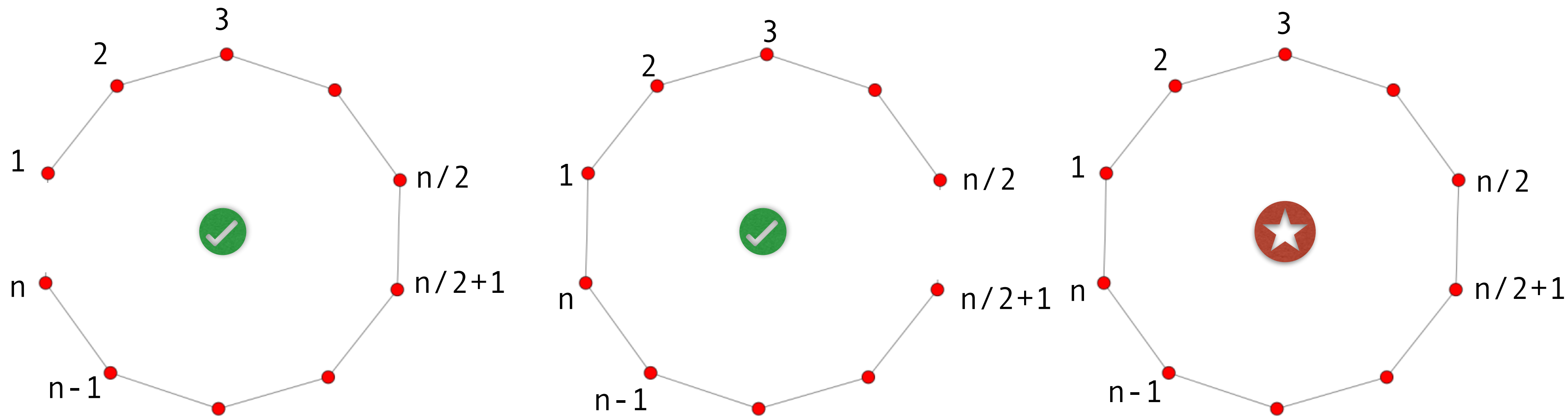
That is $k \geq 3 \ln(3) / \varepsilon$ ➡ #rounds = $O(1/\varepsilon)$. ❏

Exercice 2

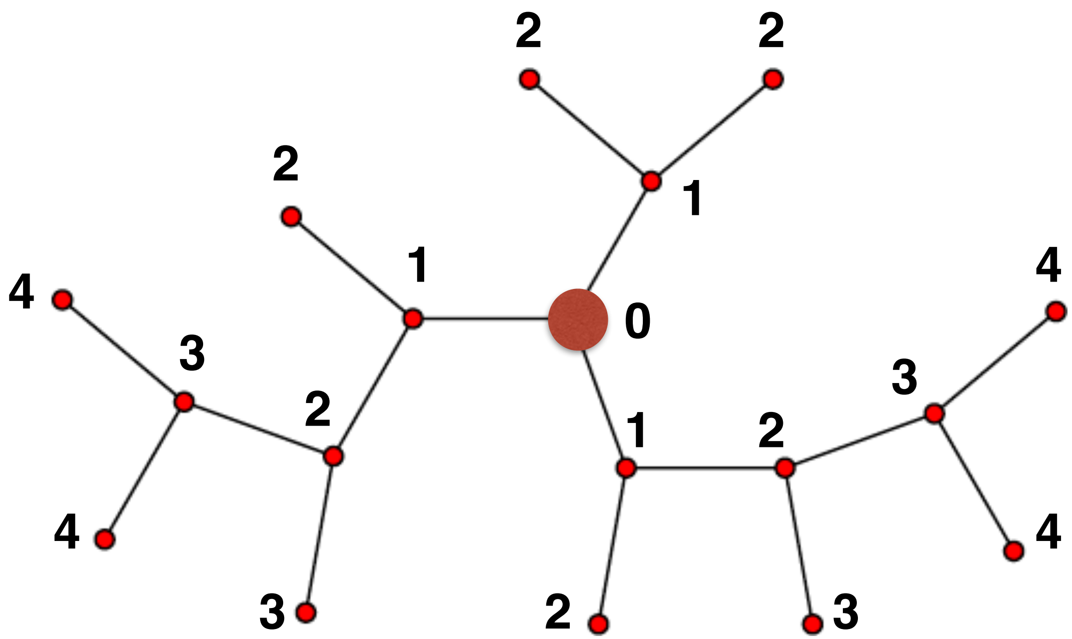
Cycle-freeness

Question 1. Show that cycle-freeness cannot be decided locally.

Cycle-freeness

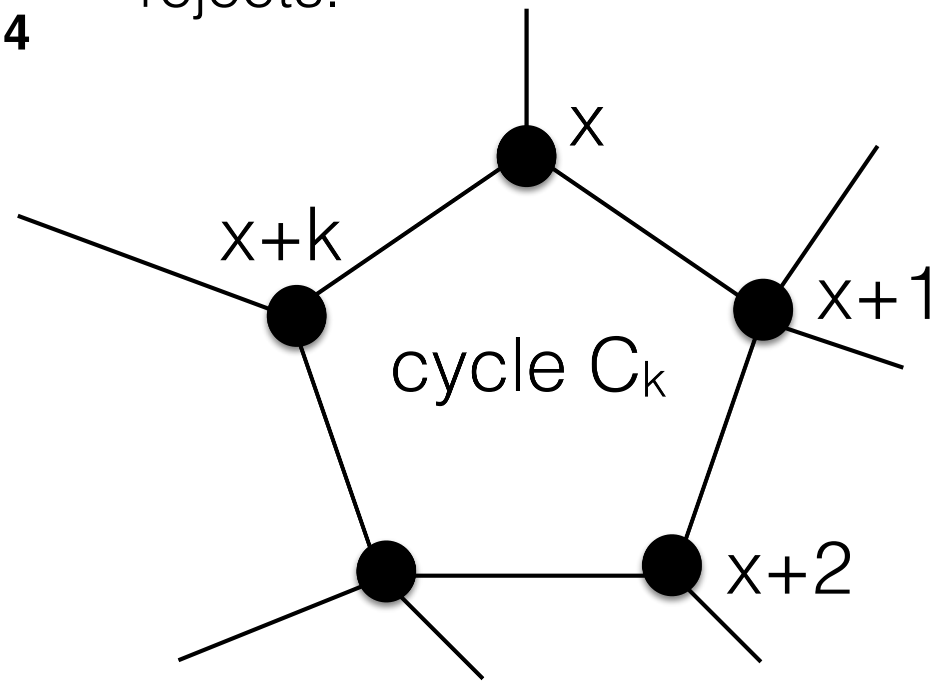


Certifying cycle-freeness



if G is acyclic, then there is an assignment of the counter resulting in all nodes accept.

if G is has a cycle, then for every assignment of the counters, at least one node rejects.



Algorithm of node u

exchange counters with neighbors
if ∃! v∈N(u) : cpt(v)=cpt(u)-1 and
∀ w∈N(u)∖{v}, cpt(w)=cpt(u)+1
then accept
else reject

Proof-Labeling Scheme

A distributed algorithm A *verifies* ϕ if and only if:

- $G \models \phi \;\Rightarrow\; \exists\, c: V(G) \to \{0,1\}^*$: all nodes accept (G,c)
- $G \not\models \phi \;\Rightarrow\; \forall\, c: V(G) \to \{0,1\}^*$ at least one node rejects (G,c)

The bit-string c(u) is called the *certificate* for u (cf. class NP)

Objective: Algorithms in O(1) rounds (ideally, just 1 round in LOCAL)

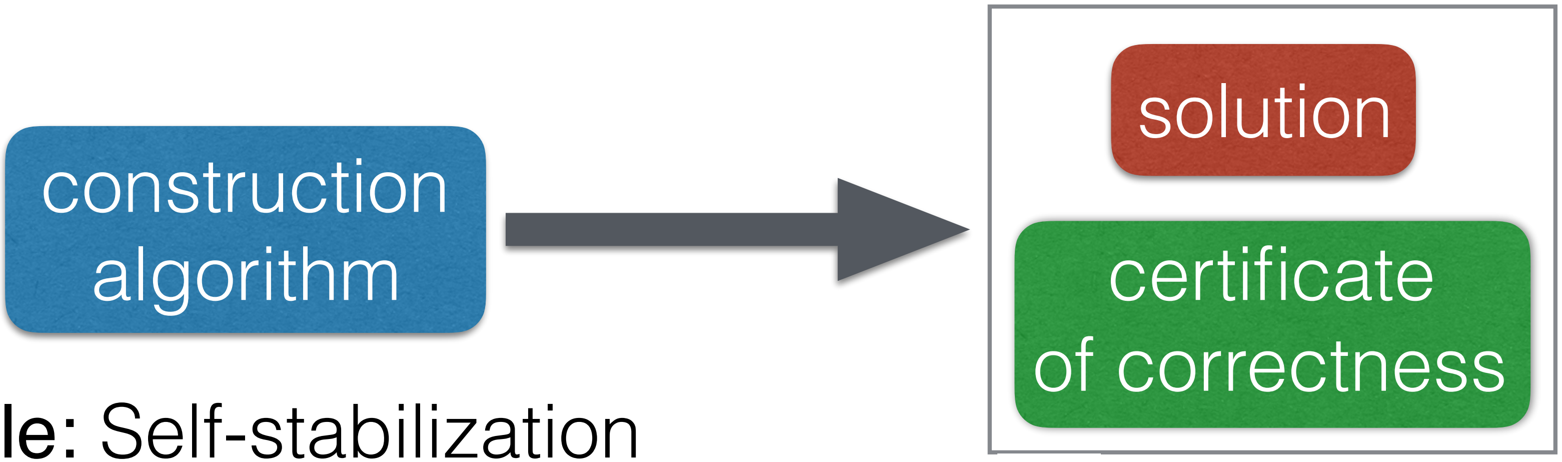
Examples:

O(log n) bits

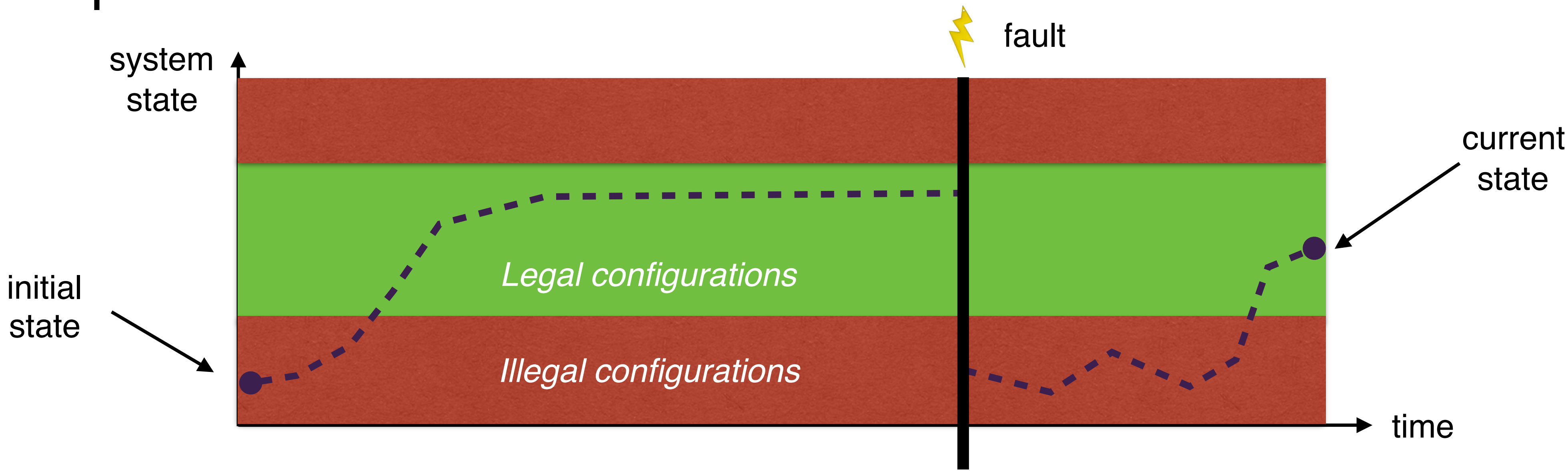
- Cycle-freeness: $c(u) = \mathrm{dist}_G(u,r)$
- Spanning tree: $c(u) = (\mathrm{dist}_G(u,r), \mathrm{ID}(r))$

Measure of complexity: $\max_{u \in V(G)} |c(u)|$

Application: Fault-Tolerance



Example: Self-stabilization



Universal PLS

Question 2. Show that, for any (decidable) graph property ϕ, there exists a PLS for ϕ, with certificates of size $O(n^2)$ bits in n-node graphs.

Universal PLS

Theorem For any (decidable) graph property ϕ, there exists a PLS for ϕ, with certificates of size $O(n^2)$ bits in n-node graphs.

Proof $c(u) = (M,x)$ where

- M = adjacency matrix of G
- x = table[1..n] with $x(i)$ = ID(node with index i)

Verification algorithm:

1. check local consistency of M using x
2. if no inconsistencies, check whether M satisfies ϕ

G satisfies $\iff$ both tests are passed ❏

Lower bound

Question 3. Show that there exists a graph property for which any PLS has certificates of size $\Omega(n^2)$ bits.

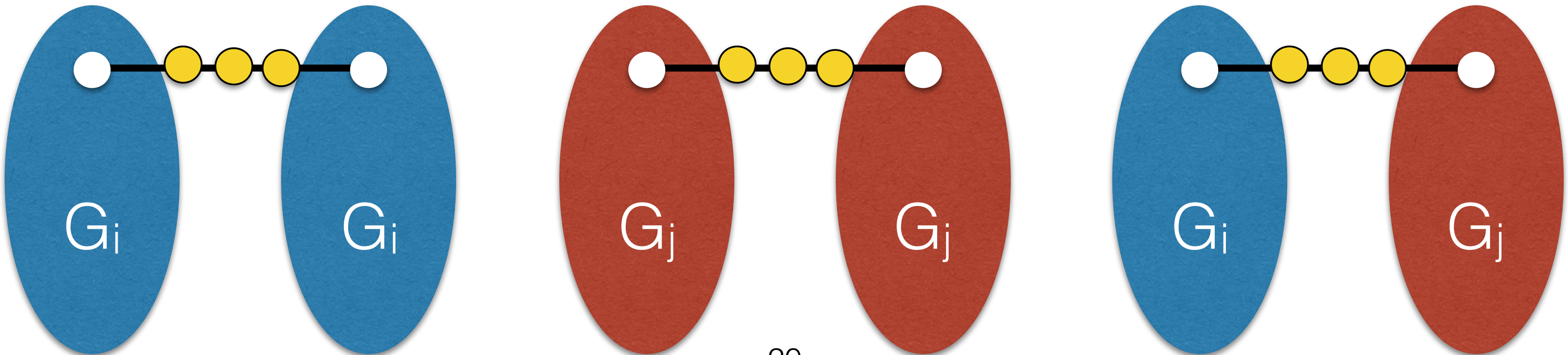
Lower bound

Theorem There exists a graph property for which any PLS has certificates of size $\Omega(n^2)$ bits.

Proof Graph automorphism = bijection $f:V(G)\rightarrow V(G)$ such that $\{u,v\} \in E(G) \iff \{f(u),f(v)\} \in E(G)$

Fact There are $\geq 2^{\varepsilon n^2}$ graphs with no non-trivial auto.

If certificates on $< \varepsilon n^2/3$ bits, then $\exists$ $i \neq j$ such that the three nodes ●●● have same certificates on G_i-G_i and G_i-G_i.



$\square$

Local hierarchy

- Equivalent of, e.g., polynomial hierarchy in complexity theory
- {locally decidable properties} = $\Sigma_0 = \Pi_0$
- {locally verifiable properties (with PLS)} = Σ_1

Deciding graph property ϕ is in Σ_1 if and only if:

- $G \models \phi \Rightarrow \exists\, c$ all nodes accept (G,c)
- $G \not\models \phi \Rightarrow \forall\, c$ at least one node rejects (G,c)

Deciding graph property ϕ is in Π_1 if and only if:

- $G \models \phi \Rightarrow \forall\, c$ all nodes accept (G,c)
- $G \not\models \phi \Rightarrow \exists\, c$ at least one node rejects (G,c)

The hierarchy $(\Sigma_k, \Pi_k)_{k \geq 0}$

Deciding graph property ϕ is in Σ_2 if and only if:

- $G \models \phi \ \Rightarrow \exists\, c_1\ \forall\, c_2$ all nodes accept (G, c_1, c_2)
- $G \not\models \phi \ \Rightarrow \forall\, c_1\ \exists\, c_2$ at least one node rejects (G, c_1, c_2)

Deciding graph property ϕ is in Π_2 if and only if:

- $G \models \phi \ \Rightarrow \forall\, c_1\ \exists\, c_2$ all nodes accept (G, c_1, c_2)
- $G \not\models \phi \ \Rightarrow \exists\, c_1\ \forall\, c_2$ at least one node rejects (G, c_1, c_2)

Deciding graph property ϕ is in Σ_k if and only if:

- $G \models \phi \ \Rightarrow \exists\, c_1\ \forall\, c_2\ \exists\, c_3 \ldots Q\, c_k$ all nodes accept $(G, c_1, \ldots, c_k)$
- $G \not\models \phi \ \Rightarrow \forall\, c_1\ \exists\, c_2\ \forall\, c_1 \ldots \neg Q\, c_k$ at least one node rejects $(G, c_1, \ldots, c_k)$

Deciding graph property ϕ is in Π_k if and only if:

- $G \models \phi \ \Rightarrow \forall\, c_1\ \exists\, c_2\ \forall\, c_3 \ldots Q\, c_k$ all nodes accept $(G, c_1, \ldots, c_k)$
- $G \not\models \phi \ \Rightarrow \exists\, c_1\ \forall\, c_2\ \exists\, c_3 \ldots \neg Q\, c_k$ at least one node rejects $(G, c_1, \ldots, c_k)$

Example: Minimum Dominating Set

Decision problem MinDS:

- input = dominating set $\mathcal{D}$ (i.e., $\mathcal{D}(u) \in \{0,1\}$)
- output = accept if $|\mathcal{D}| = \min_{\text{dom } D} |D|$

Question 4. Show that MinDS $\in \Pi_2$

Example: Minimum Dominating Set

Decision problem MinDS:

- input = dominating set $\mathcal{D}$ (i.e., $\mathcal{D}(u) \in \{0,1\}$)
- output = accept if $|\mathcal{D}| = \min_{\text{dom } D} |D|$

Theorem MinDS $\in \Pi_2$

Proof

c_1 encodes a dominating set, i.e., $c_1(u) \in \{0,1\}$

c_2 encodes:

- a spanning tree T_{err} pointing to node u with error in c_1 if any
- a spanning tree T_0 for counting $|\mathcal{D}|$ (w/ same root)
- a spanning tree T_1 for counting $|c_1|$ (w/ same root)

Algorithm:

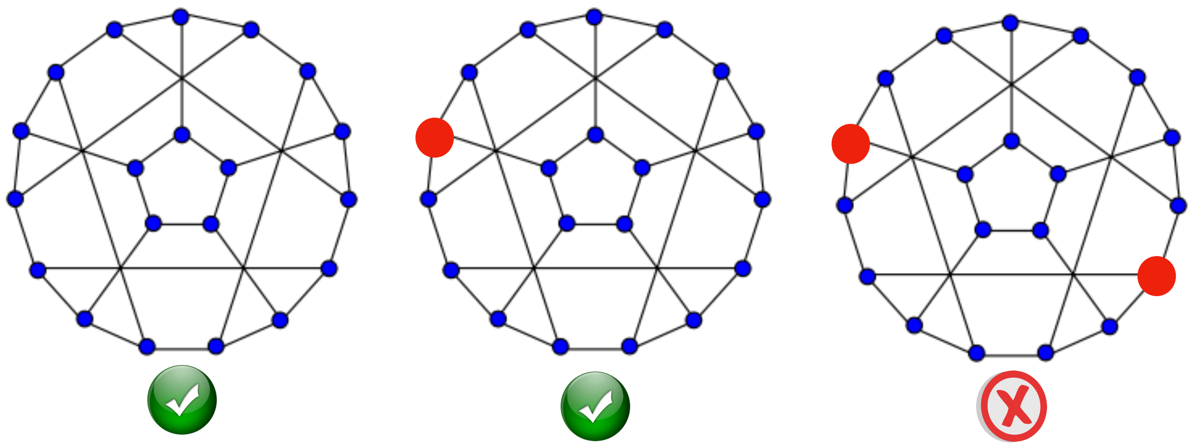
- If root u ses $|c_1| < |\mathcal{D}|$ with no error, it rejects, otherwise it accepts
- If any node detects inconsistencies in T_0, T_1 or T_{err} it rejects, otherwise it accepts. ❑

Exercice 3

Randomized Protocols

[FKP, 2013]

- At most one selected (AMOS)

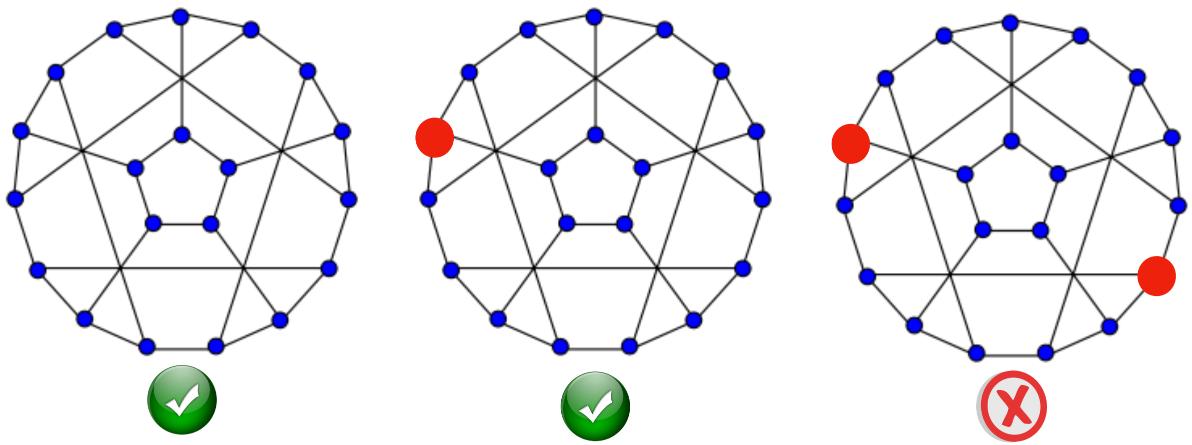


- **Question 1.** Show that there exists a randomized algorithm performing in a constant number of rounds for deciding AMOS.

Randomized Protocols

[FKP, 2013]

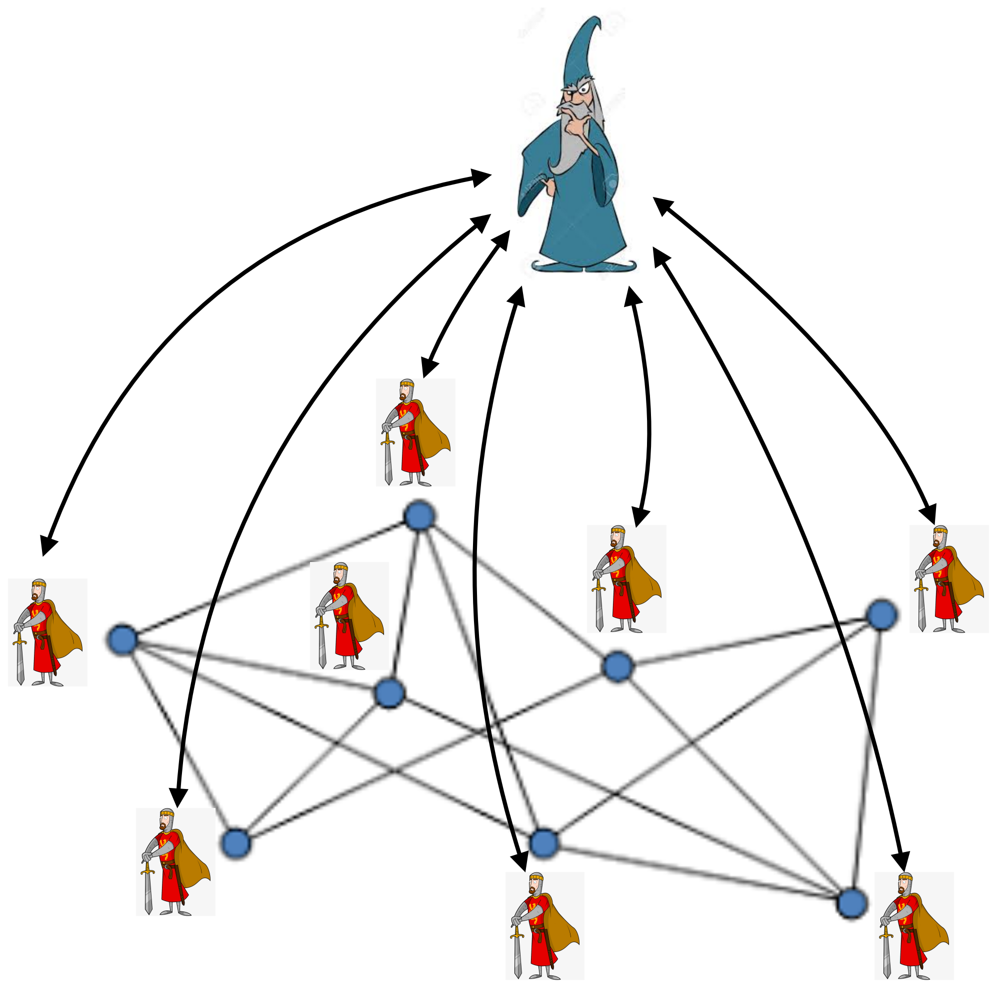
- At most one selected (AMOS)



- Decision algorithm (2-sided):
  - let $p = (\sqrt{5}-1)/2 = 0.61\ldots$
  - If not selected then accept
  - If selected then accept w/ prob p, and reject w/ prob 1-p
- Issue with boosting! — But OK for 1-sided error

Distributed Interactive Protocols

[KOS, 2018]

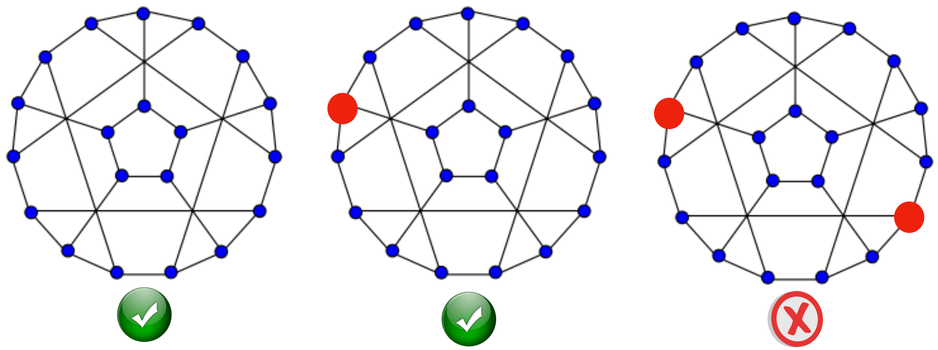


- Arthur-Merlin Phase *(no communication, only interactions)*
- Verification Phase *(only communications)*

- Merlin has infinite communication power
- Arthur is randomized

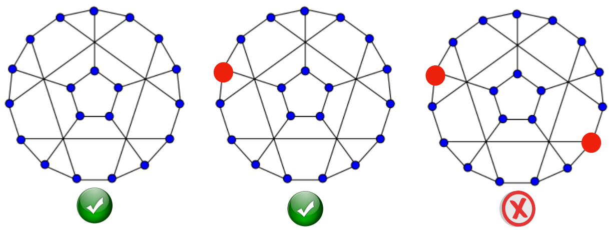
- k = #interactions
- dAM[k] or dMA[k]

Example: AMOS



- In BPLD with success prob $(\sqrt{5}-1)/2 = 0.61\ldots$
- In Σ_1LD($O(\log n)$) — Not in Σ_1LD($o(\log n)$)
- Not in dMA($o(\log n)$) for success prob > 4/5
- **Question 2.** Show that AMOS is in dAM(k) with k random bits, and success prob $1-1/2^k$

Example: AMOS



- In BPLD with success prob $(\sqrt{5}-1)/2 = 0.61\ldots$
- In Σ_1LD(O(log n)) — Not in Σ_1LD(o(log n))
- Not in dMA(o(log n)) for success prob > 4/5
- In dAM(k) with k random bits, and success prob $1-1/2^k$
  - Arthur independently picks a k-bit index at each node u.a.r.
  - Merlin answer $\perp$ if no nodes selected, or the index of the selected node

Sequential setting

- For every $k \geq 2$, $AM[k] = AM$
- $MA \subseteq AM$ because $MA \subseteq MAM = AM[3] = AM$
- $MA \in \Sigma_2 P \cap \Pi_2 P$
- $AM \in \Pi_2 P$
- $AM[\mathit{poly}(n)] = IP = PSPACE$

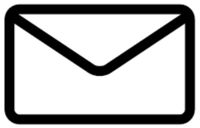
Known results

[KOS 2018, NPY 2018]

- Sym ∈ dAM($n \log n$)
- Sym ∈ dMAM($\log n$)
- Any dAM protocol for Sym requires $\Omega(\log\log n)$-bit certificates
- ¬Sym ∈ dAMAM($\log n$)
- Other results on graph non-isomorphism

Parameters

- Number of interactions between  and 
- Size of 

- Size of 
- Number of random 
- Shared vs distributed 

Tradeoffs

[CFP, 2019]

- **Theorem 1** For every c, there exists a Merlin-Arthur (dMA) protocol for *triangle-freeness*, using $O(\log n)$ bits of shared randomness, with $\tilde{O}(n/c)$-bit certificates and $\tilde{O}(c)$-bit messages between nodes.

- **Theorem 2** There exists a graph property admitting a proof-labeling scheme with certificates and messages on $O(n)$ bits, that cannot be solved by an Arthur-Merlin (dAM) protocol with certificates on $o(n)$ bits, for any fixed number $k \geq 0$ of interactions between Arthur and Merlin, even using shared randomness, and even with messages of unbounded size.