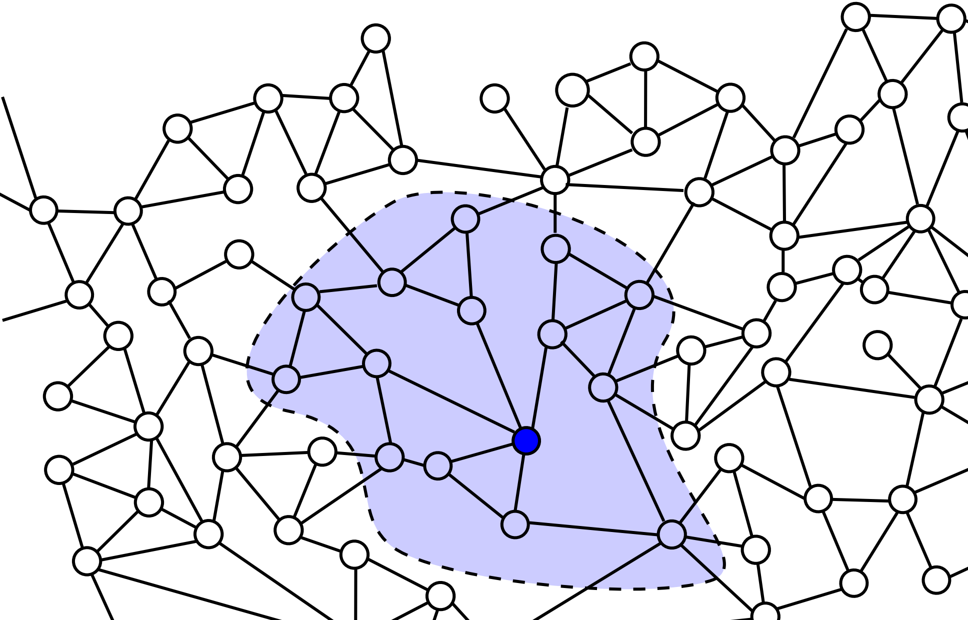
A Hierarchy of Local Decision

Laurent Feuilloley
Pierre Fraigniaud
Juho Hirvonen

ICALP · July 2016

Local computation



(Very) local computation

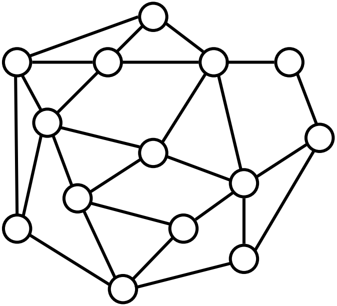
An undirected graph,

whose nodes have identifiers,

and run the same algorithm,

based on synchronous snapshots,

at constant distance.



(Very) local computation

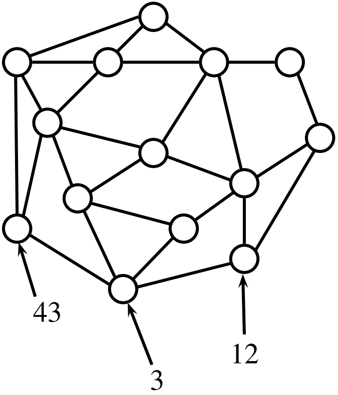
An undirected graph,

whose nodes have identifiers,

and run the same algorithm,

based on synchronous snapshots,

at constant distance.



(Very) local computation

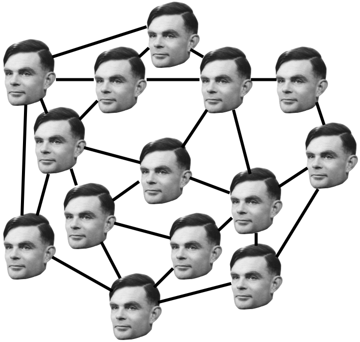
An undirected graph,

whose nodes have identifiers,

and run the same algorithm,

based on synchronous snapshots,

at constant distance.



(Very) local computation

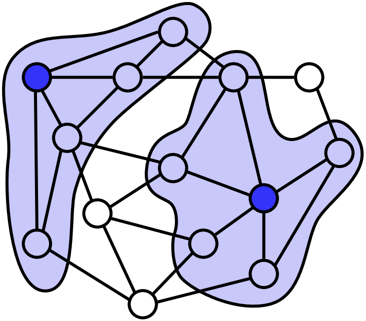
An undirected graph,

whose nodes have identifiers,

and run the same algorithm,

based on synchronous snapshots,

at constant distance.



(Very) local computation

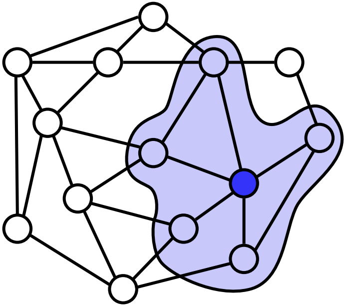
An undirected graph,

whose nodes have identifiers,

and run the same algorithm,

based on synchronous snapshots,

at constant distance.



(Very) local computation

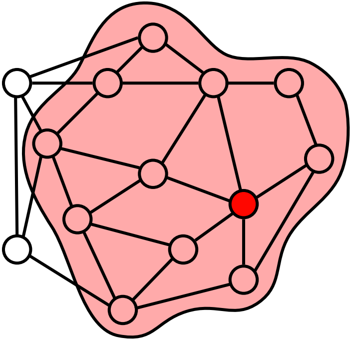
An undirected graph,

whose nodes have identifiers,

and run the same algorithm,

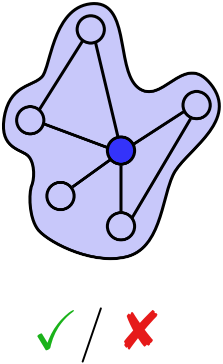
based on synchronous snapshots,

at constant distance.

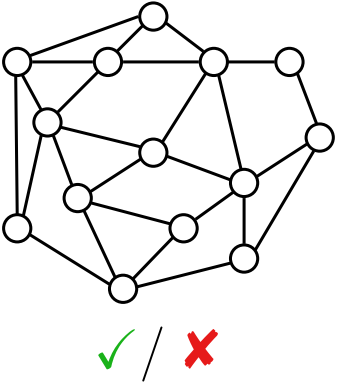


Distributed decision

Locally



Globally

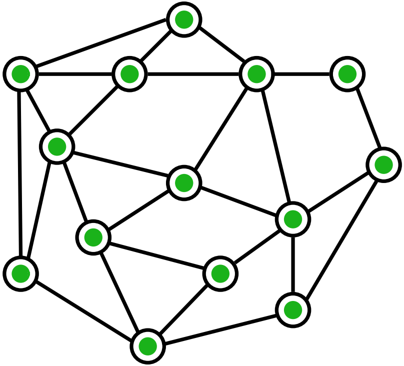


Distributed decision

The input graph is globally accepted if and only if it is locally accepted everywhere.

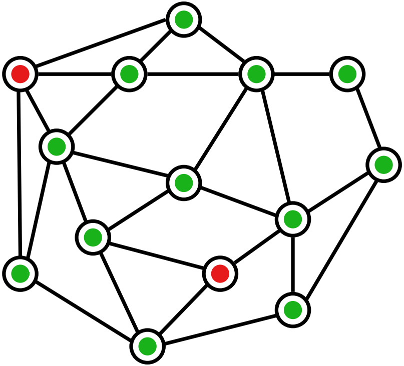
Distributed decision

The graph is either unanimously accepted...

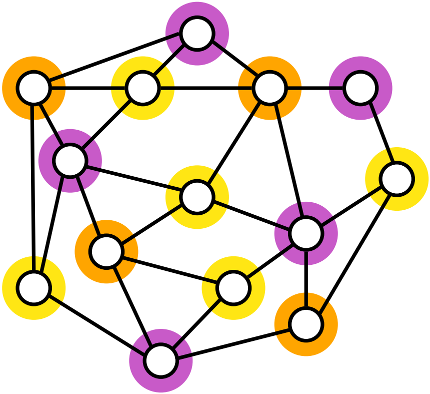


Distributed decision

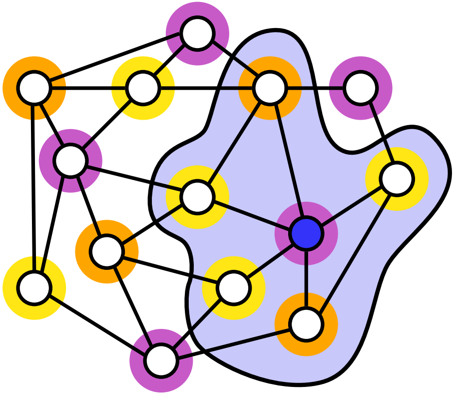
or rejected by veto.



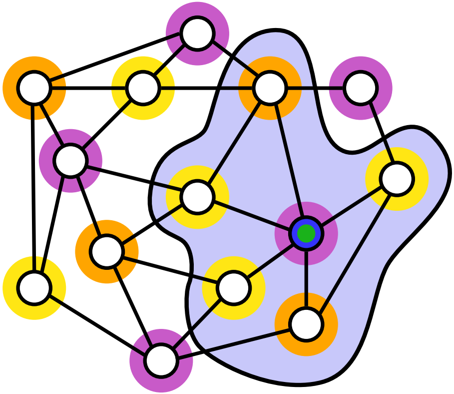
Example : coloured graphs



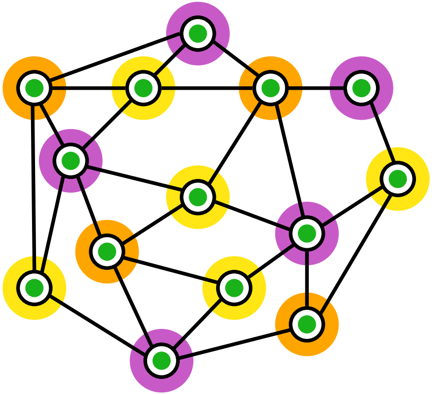
Example : coloured graphs



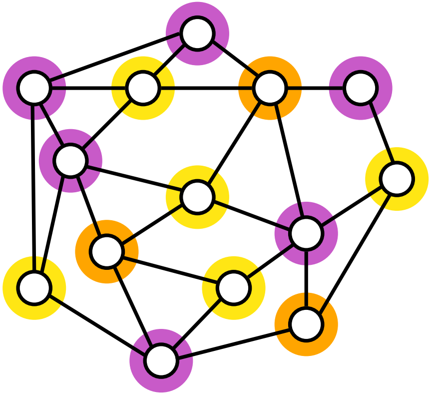
Example : coloured graphs



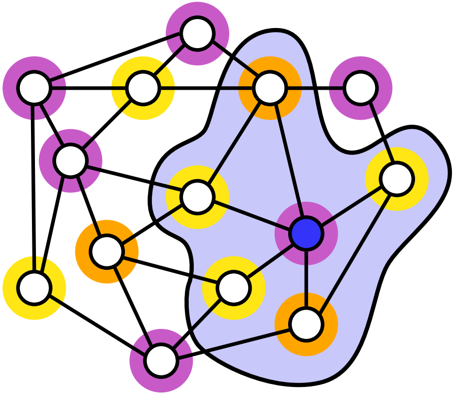
Example : coloured graphs



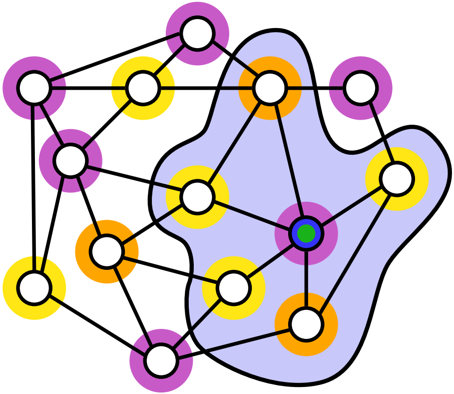
Example : coloured graphs



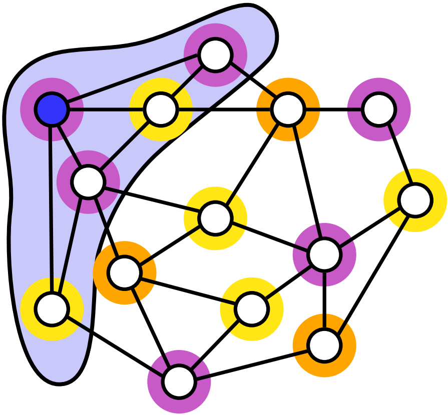
Example : coloured graphs



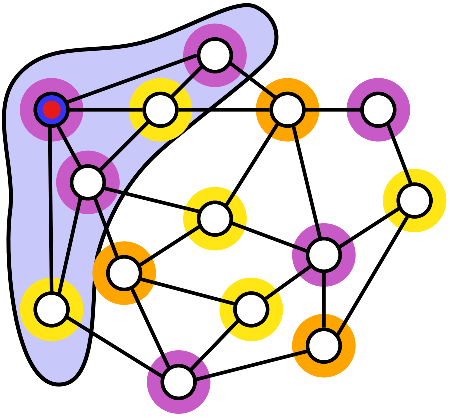
Example : coloured graphs



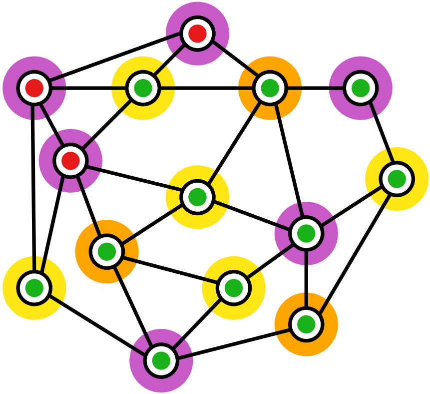
Example : coloured graphs



Example : coloured graphs



Example : coloured graphs



Master plan

Build a complexity theory
for this computational model.

Languages and classes

Language :

- Set of graphs with inputs ($\mathcal{L} = \{(G, x)\}$), Turing decidable
- Example : Well-coloured graphs

Class :

- Set of languages ($\mathcal{C} = \{\mathcal{L}\}$)
- Example : LD (Local Distributed)

Languages and classes

$\mathcal{L} \in P$ if and only if

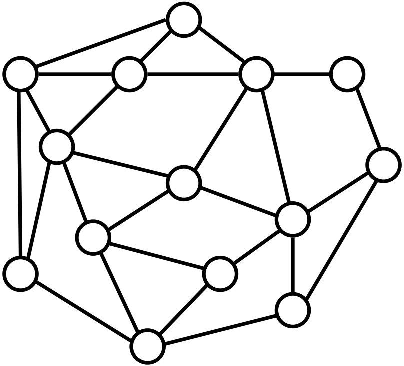
$\exists A \in$ Polytime such that $\forall x, x \in \mathcal{L} \Leftrightarrow A(x) = 1$

$\mathcal{L} \in LD$ if and only if

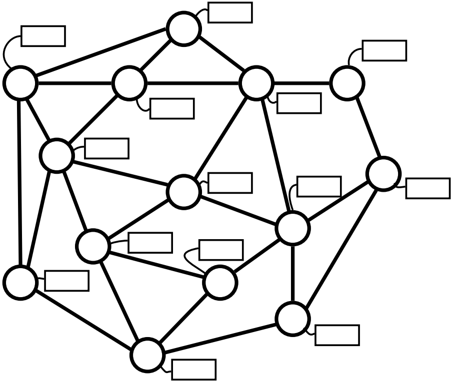
$\exists A \in$ Cst-dist s.t. $\forall G, x, (G, x) \in \mathcal{L} \Leftrightarrow A(G, x) = 1$

where $A(G, x) = 1$ means $\forall v, A(G, x_v, v) = 1$

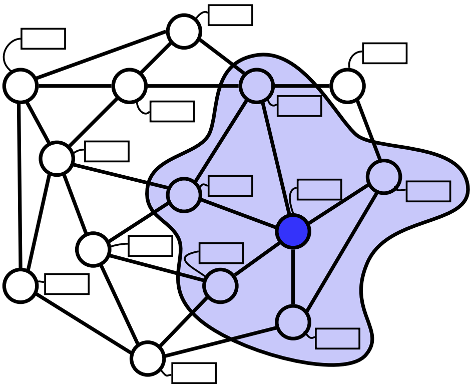
Distributed non-determinism



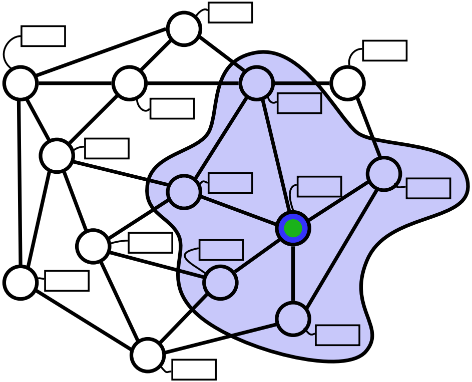
Distributed non-determinism



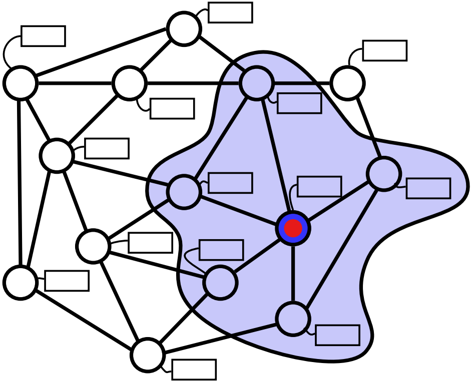
Distributed non-determinism



Distributed non-determinism



Distributed non-determinism



Distributed non-determinism

$\mathcal{L} \in NP$ if and only if

$\exists A \in$ Polytime such that for all x

$$x \in \mathcal{L} \Leftrightarrow \exists y, A(x, y) = 1$$

$\mathcal{L} \in$ "NLD" if and only if

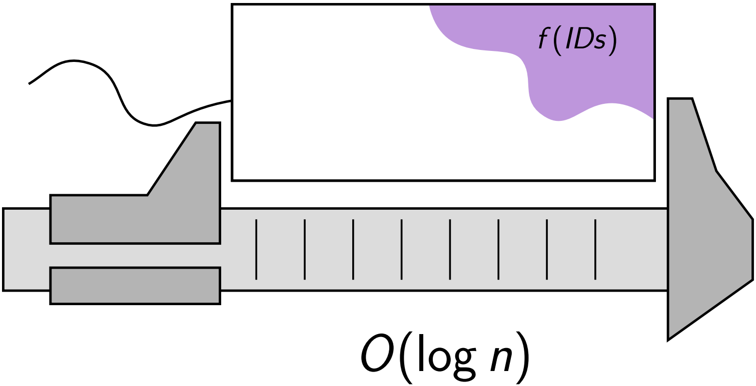
$\exists A \in$ Cst-dist such that for all G, x,

$$(G, x) \in \mathcal{L} \Leftrightarrow \exists y, A(G, x, y) = 1$$

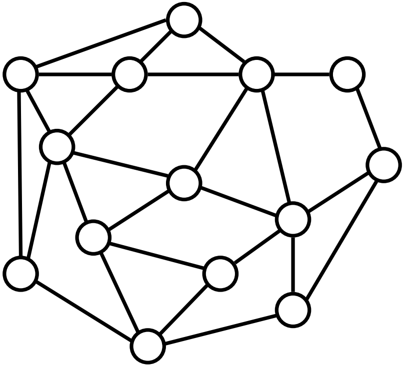
where $A(G, x, y) = 1$ means $\forall v, A(G, x_v, y_v, v) = 1$

Distributed non-determinism

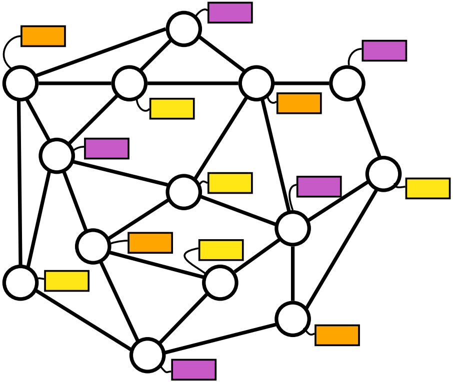
Distributed non-determinism



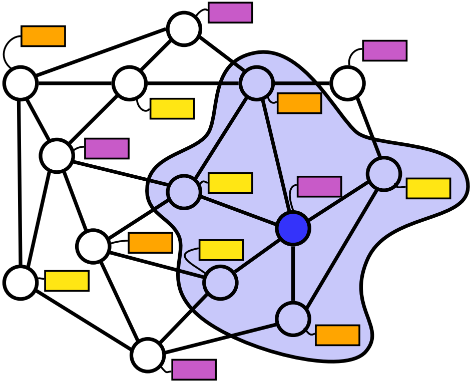
Is the graph 3-colourable ?



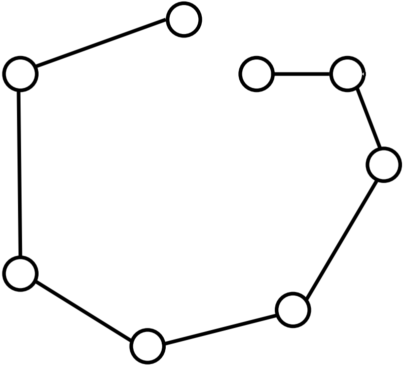
Is the graph 3-colourable ?



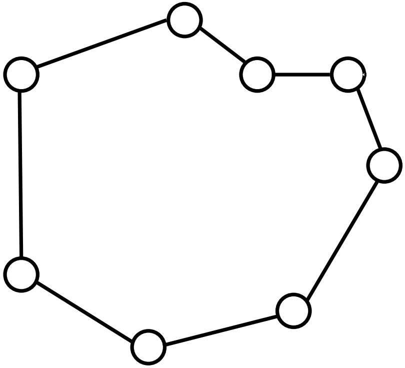
Is the graph 3-colourable ?



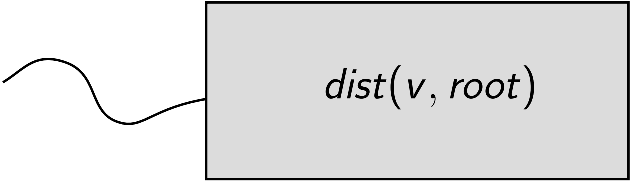
Is the graph a tree ?



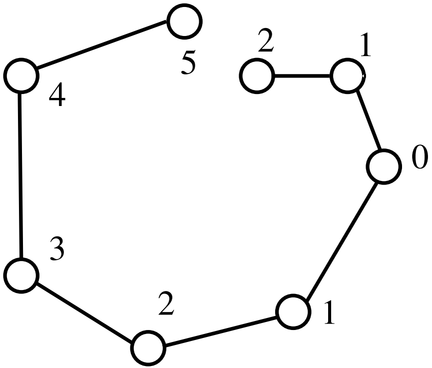
Is the graph a tree?



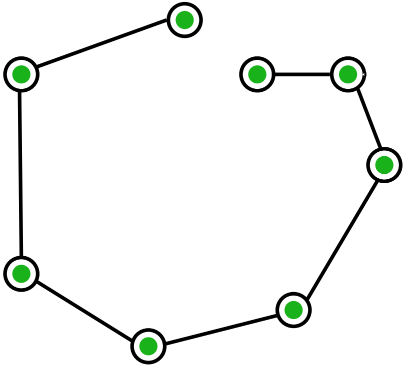
Is the graph a tree?



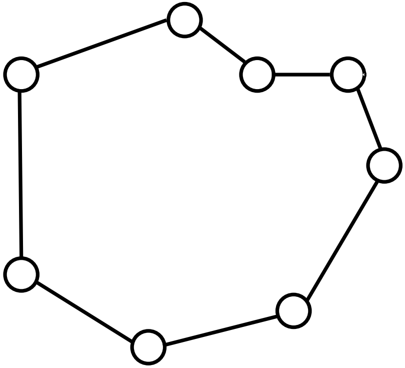
Is the graph a tree ?



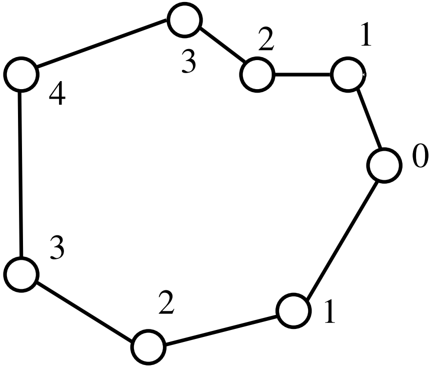
Is the graph a tree ?



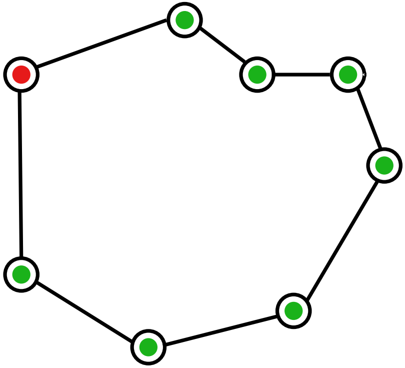
Is the graph a tree ?



Is the graph a tree?



Is the graph a tree ?



Spanning tree

- Language :

$\mathcal{L} = \{(G, x) : x \text{ encodes a spanning tree of } G\}$

- Certificate :

Distance(u,root), and the ID of the root

- Consequence :

$\{(G, x) : \exists u \in V(G) \text{ with local property } P\} \in NLD$

- In particular : co-LD$\in$ NLD.

Hierarchy

$\mathcal{L} \in \Sigma_p$ if and only if

$\exists A \in$ Polytime such that for all x

$$x \in \mathcal{L} \Leftrightarrow \exists y_1, \forall y_2, \ldots, Q_p y_p, yA(x, y_1, \ldots, y_p) = 1$$

$\mathcal{L} \in \Sigma_p^L$ if and only if

$\exists A \in$ Cst-dist such that for all G, x,

$$(G, x) \in \mathcal{L} \Leftrightarrow \exists y_1, \forall y_2, \ldots, Q_p y_p, A(G, x, y) = 1$$

Hierarchy

$\mathcal{L} \in \Pi_p$ if and only if

$\exists A \in$ Polytime such that for all x

$$x \in \mathcal{L} \Leftrightarrow \forall y_1, \exists y_2, \ldots, Q_p y_p, yA(x, y_1, \ldots, y_p) = 1$$

$\mathcal{L} \in \Pi_p^L$ if and only if

$\exists A \in$ Cst-dist such that for all G, x,

$$(G, x) \in \mathcal{L} \Leftrightarrow \forall y_1, \exists y_2, \ldots, Q_p y_p, A(G, x, y) = 1$$

A connection to logic

By restricting the model, the hierarchy coincides with the properties expressible in MSO.

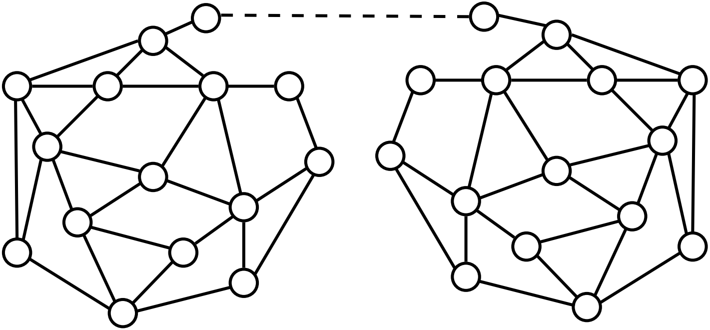
A bit of structure

$$\Lambda_i = \begin{cases} \Sigma_i^L & \text{if } i \text{ is odd,} \\ \Pi_i^L & \text{if } i \text{ is even.} \end{cases}$$

Basically : a node can simulate the last $\forall$ quantifier.

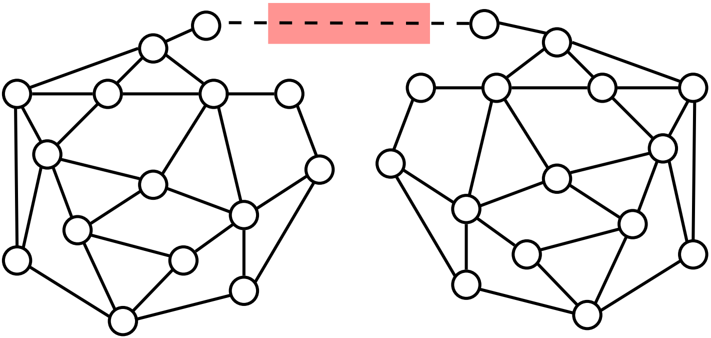
A bit of algorithms

$$\mathcal{L} = \{\text{Symmetric graphs}\}$$



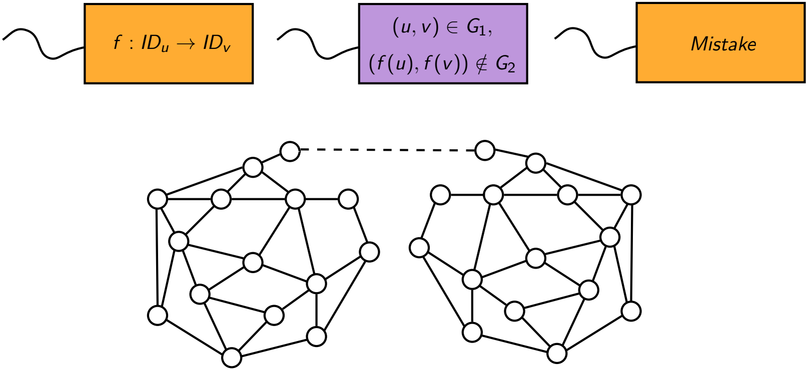
A bit of algorithms

$$\mathcal{L} = \{\text{Symmetric graphs}\} \notin \Lambda_1$$

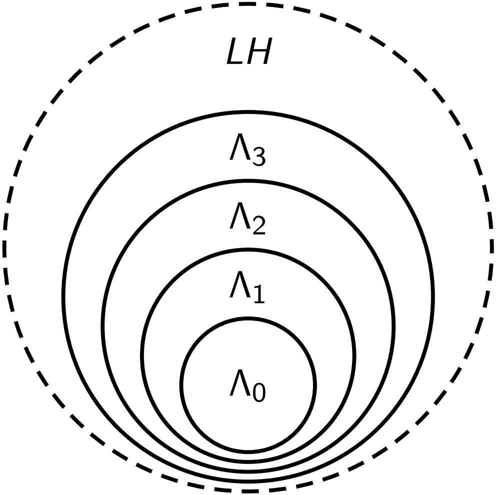


A bit of algorithms

$$\mathcal{L} = \{\text{Symmetric graphs}\} \in \Lambda_3$$



The picture



More results

- The hierarchy does not capture all the languages
- More on co-classes
- Π_1 is special.
- Weak connection with the polynomial hierarchy
- Λ_2 contains the classic optimization problems.

Conclusion

- More on distributed decision :

"Survey of Distributed Decision" in the BEATCS

- More research :

Open problem : separation of the levels 2 and 3.