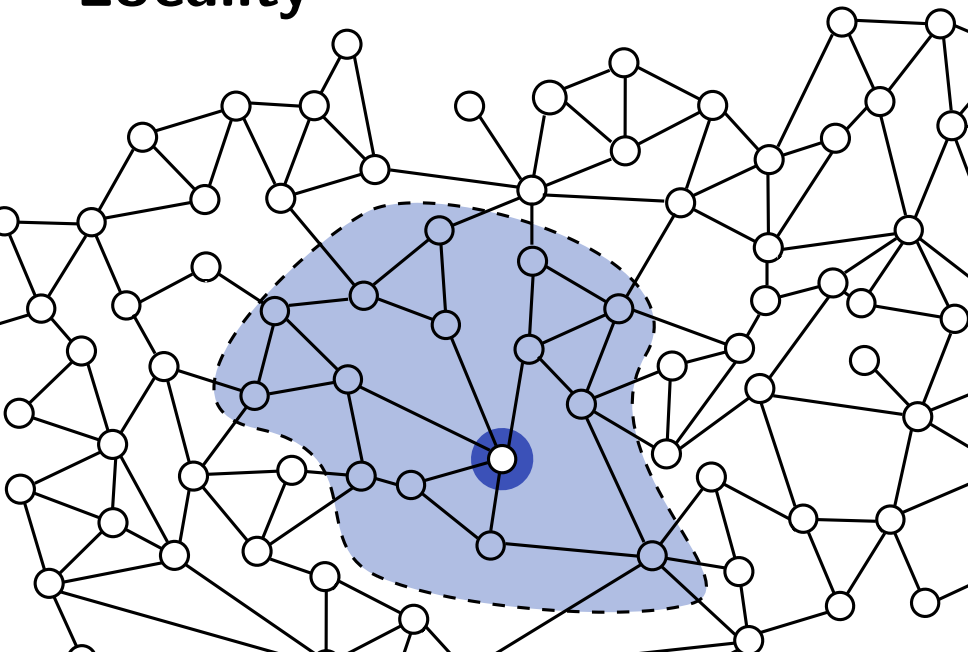


Error-sensitive proof-labeling schemes

Laurent Feuilloley
joint work with Pierre Fraigniaud
Université Paris Diderot

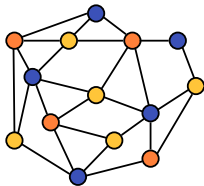
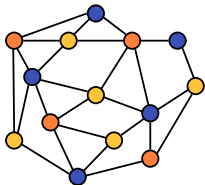
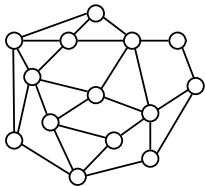
2017

Locality



Distributed decision

Building vs. deciding



Yes/No

Distributed languages

Context :

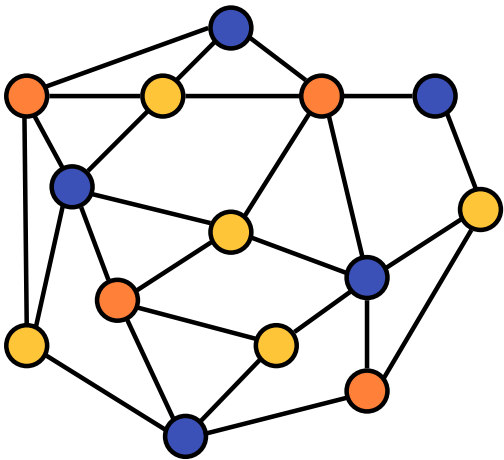
- ▶ Communication graph G
- ▶ Node inputs, $x : v \mapsto x(v)$

A language is

a set of configurations (G, x)

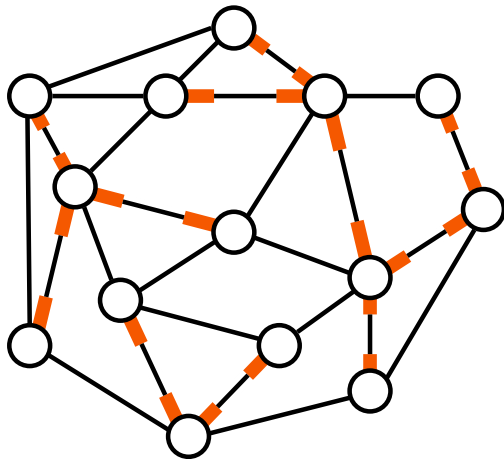
s.t. $\forall G, \exists x, (G, x) \in \mathcal{L}$

Properly-colored graphs



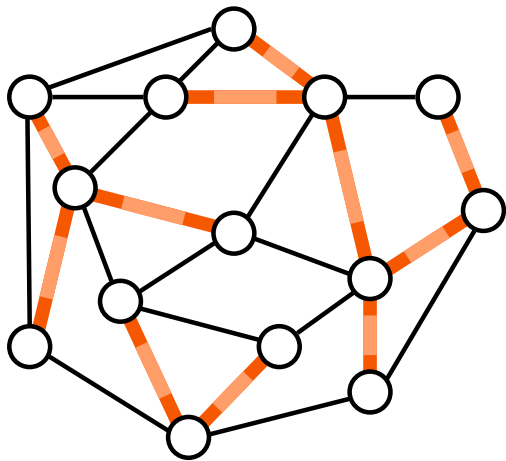
$$\mathcal{L} = \{(G, x) \text{ s.t. } x \text{ is a proper coloring of } G\}$$

Spanning forest



$$\mathcal{L} = \{(G, x) \text{ s.t. } x \text{ describes a spanning forest of } G\}$$

Spanning forest

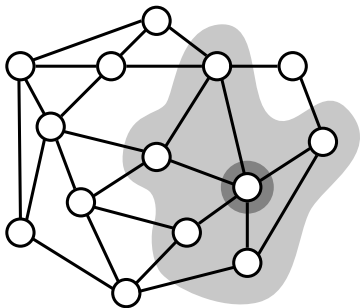


$$\mathcal{L} = \{(G, x) \text{ s.t. } x \text{ describes a spanning forest of } G\}$$

Decision mechanism

Every node :

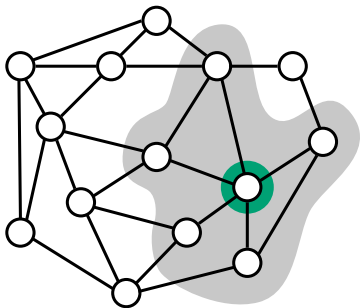
- ▶ gathers its 1-neighbourhood
- ▶ outputs a local decision **accept** or **reject**.



Decision mechanism

Every node :

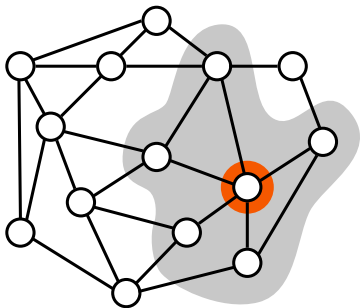
- ▶ gathers its 1-neighbourhood
- ▶ outputs a local decision **accept** or **reject**.



Decision mechanism

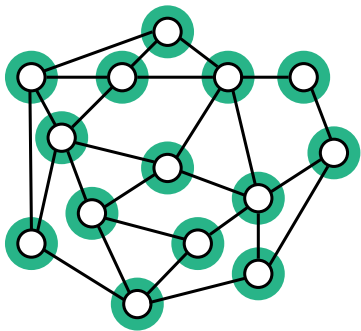
Every node :

- ▶ gathers its 1-neighbourhood
- ▶ outputs a local decision
accept or **reject**.



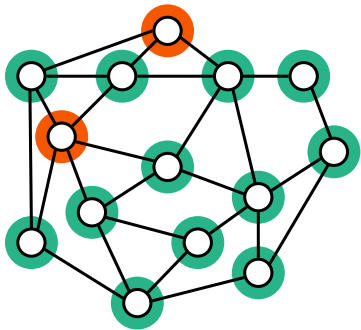
Decision mechanism

(G, x) is accepted
if all node accept.

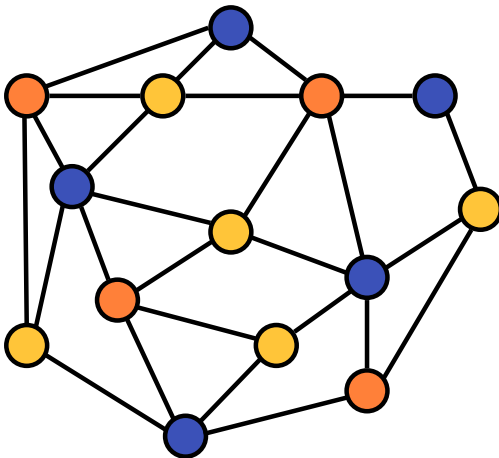


Decision mechanism

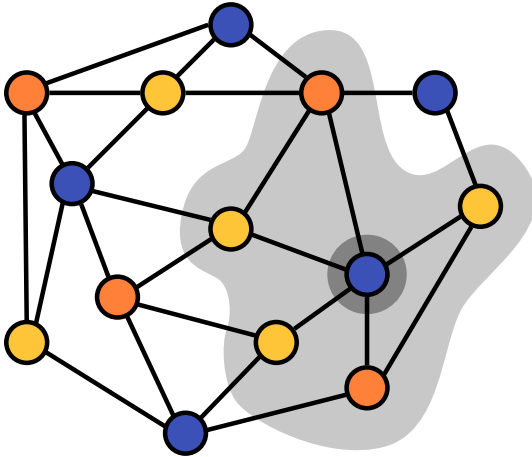
(G, x) is rejected
if at least one node
rejects.



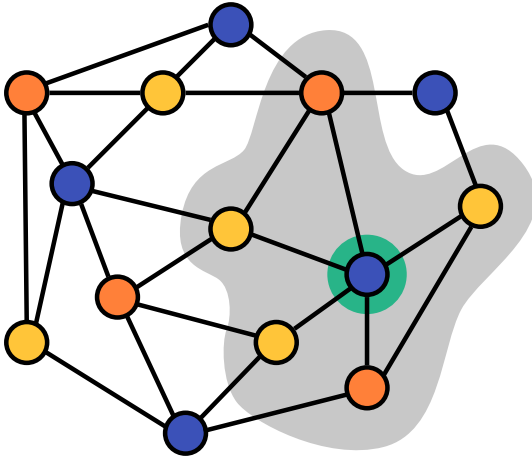
Properly-colored graphs



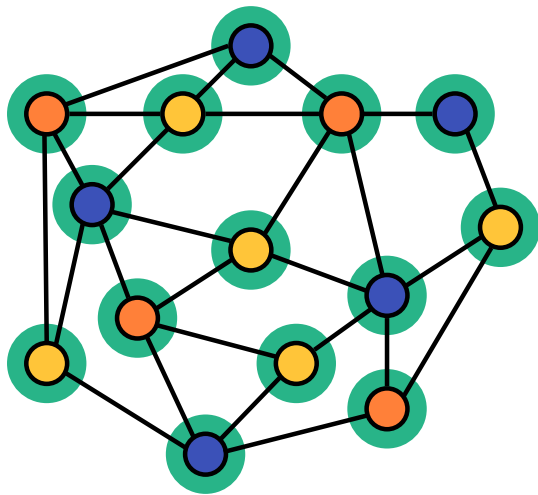
Properly-colored graphs



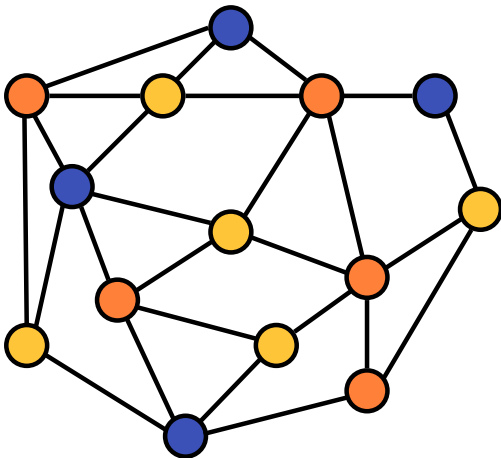
Properly-colored graphs



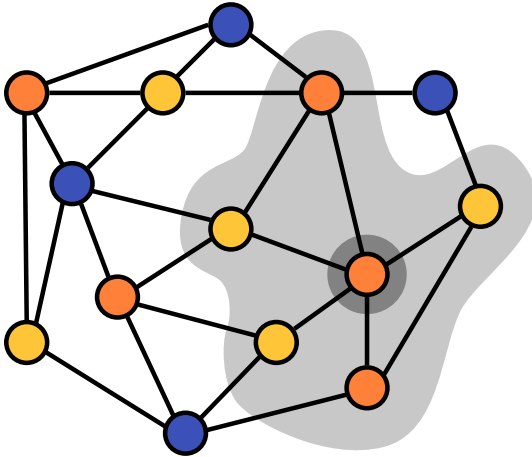
Properly-colored graphs



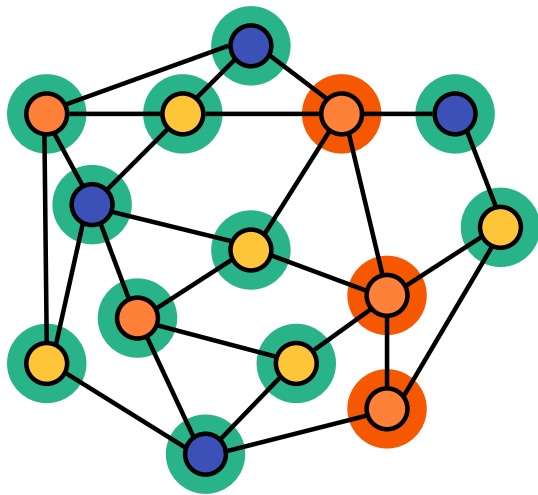
Properly-colored graphs



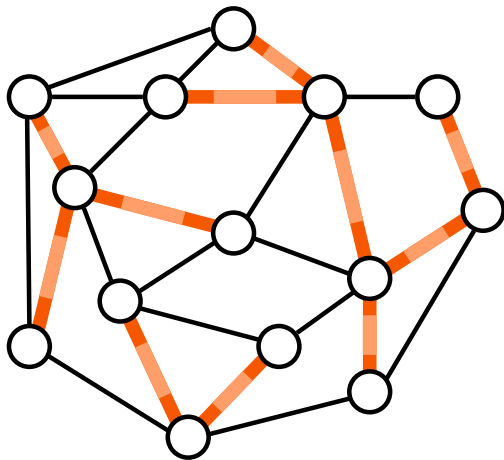
Properly-colored graphs



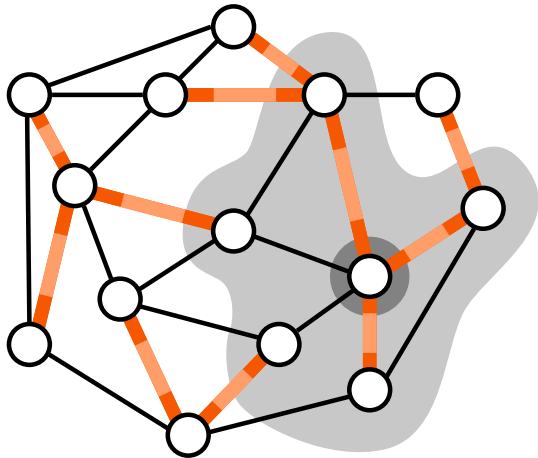
Properly-colored graphs



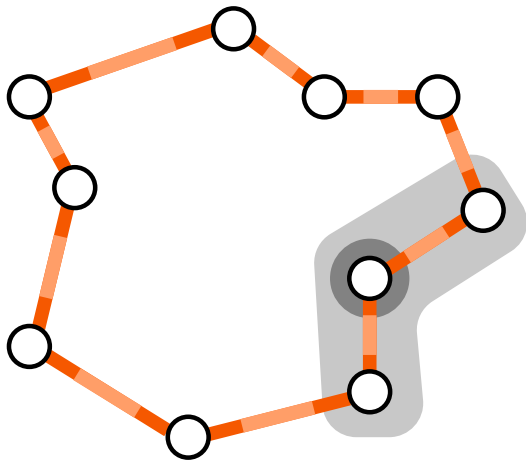
Spanning forest



Spanning forest



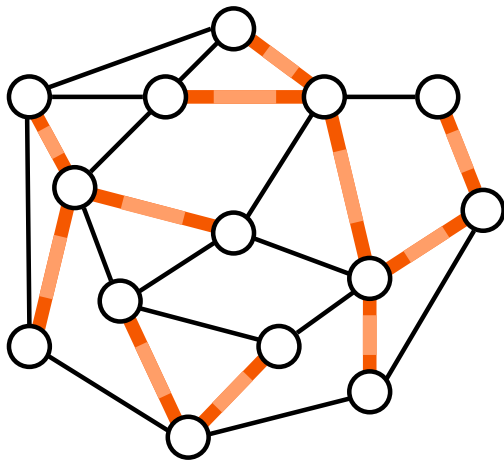
Spanning forest



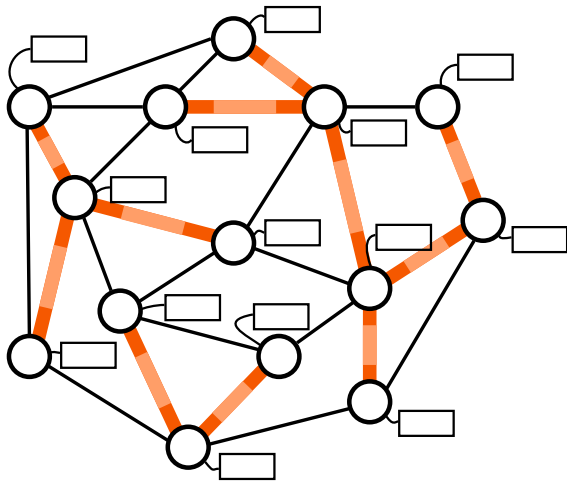
Proof-labeling schemes

**Distributed
non-determinism**

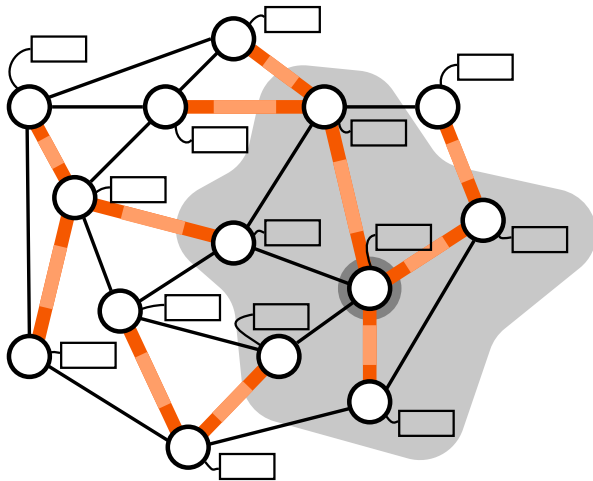
Proof-labeling schemes



Proof-labeling schemes



Proof-labeling schemes



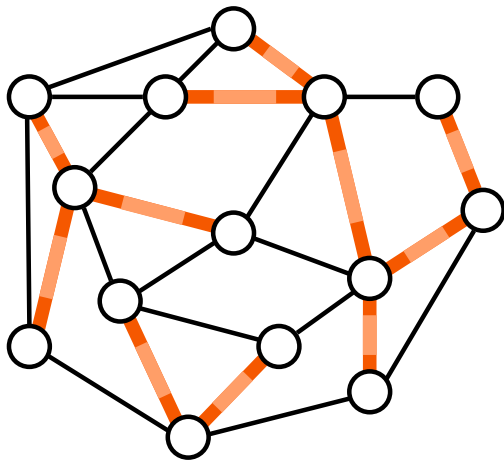
Proof-labeling schemes

Given a proof-labeling scheme for \mathcal{L} :

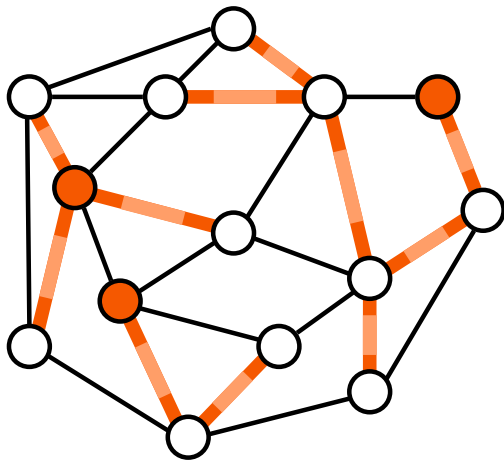
For all (G, x) :

- ▶ If $(G, x) \in \mathcal{L}$:
 $\exists c$ s.t. (G, x, c) is **accepted**.
- ▶ If $(G, x) \notin \mathcal{L}$:
 $\forall c$, (G, x, c) is **rejected**.

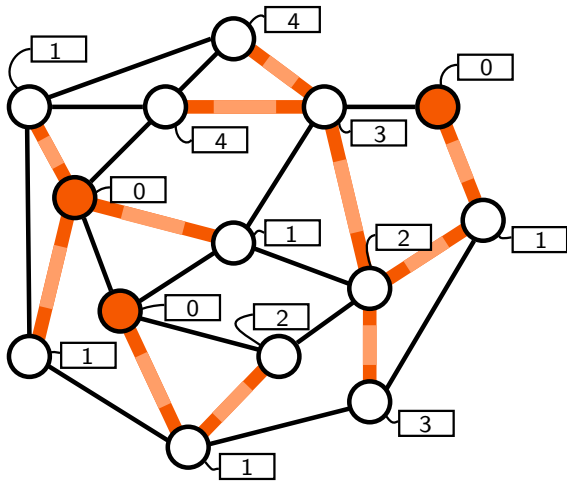
Proof-labeling schemes



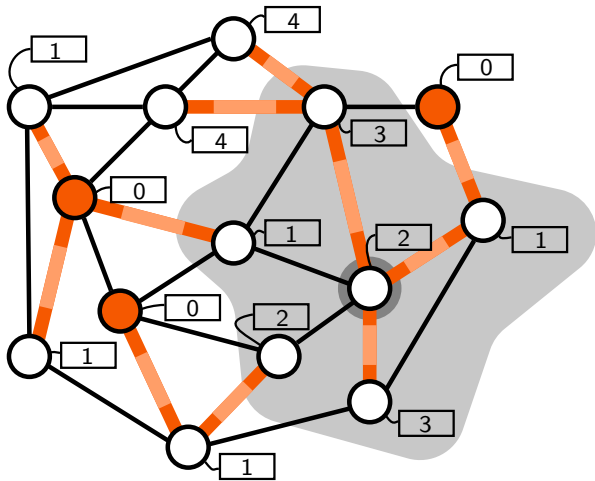
Proof-labeling schemes



Proof-labeling schemes

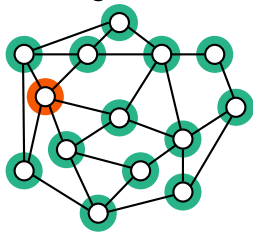
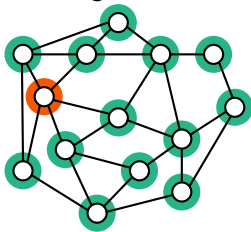
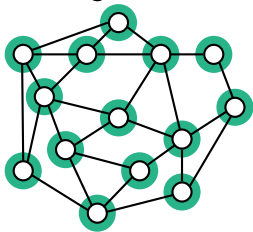
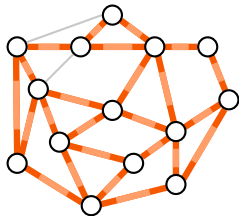
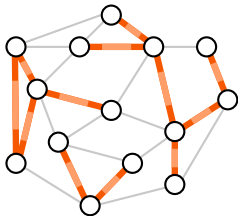
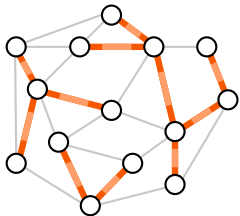


Proof-labeling schemes

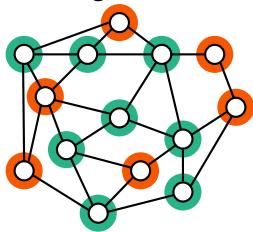
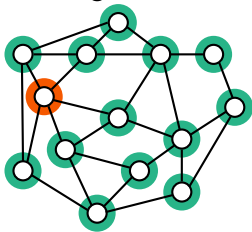
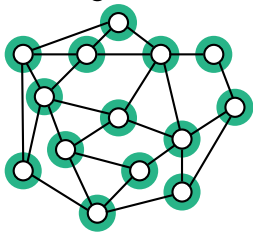
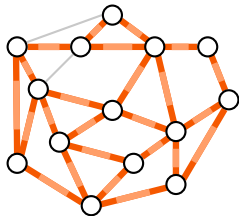
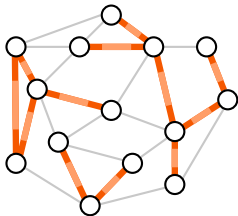
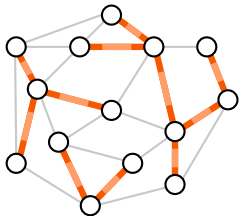


Error-sensitivity of proof-labeling schemes

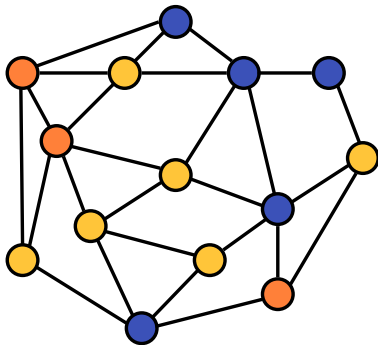
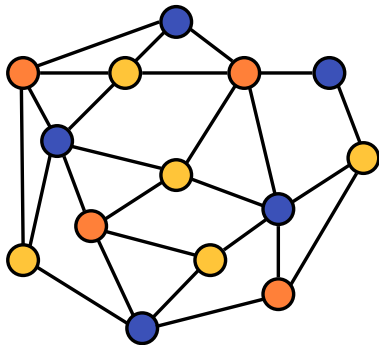
One node to reject



More nodes to reject



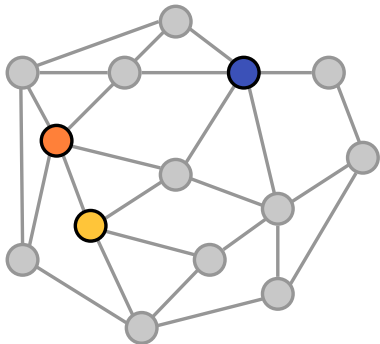
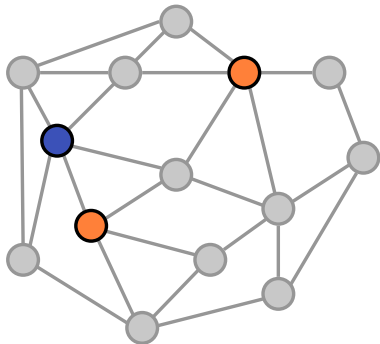
Distance



$$d((G, x_1), (G, x_2)) = \#\{v : x_1(v) \neq x_2(v)\}$$

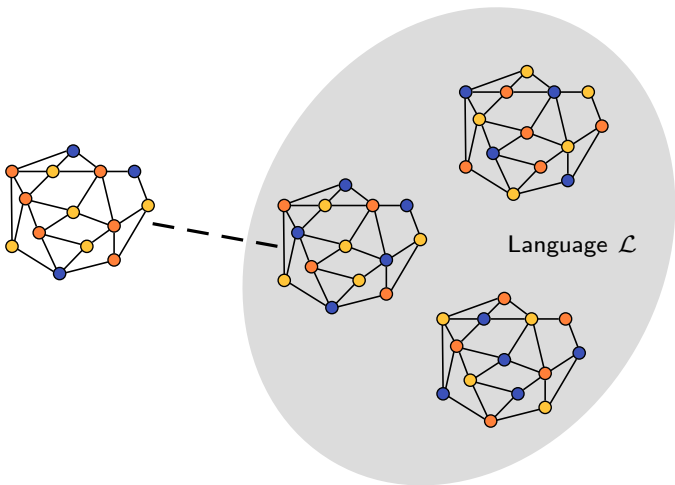
Distance

Distance=3



$$d((G, x_1), (G, x_2)) = \#\{v : x_1(v) \neq x_2(v)\}$$

Distance



$$d((G, x), \mathcal{L}) = \min_{(G', x') \in \mathcal{L}} d((G, x), (G', x'))$$

Error-sensitivity

in words

A PLS is **error-sensitive** if the number of rejecting nodes grows linearly with the distance.

Error-sensitivity

with a formula

A PLS is **error-sensitive** if

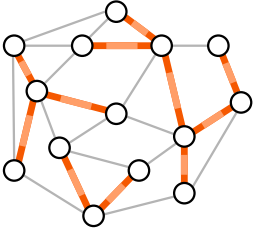
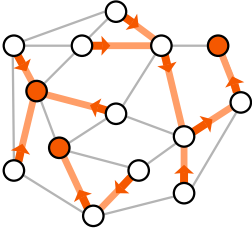
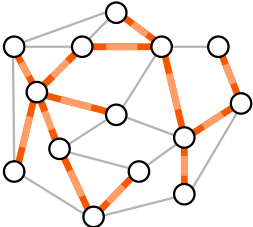
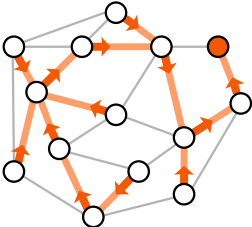
there exists $\alpha > 0$ s.t.,

for all (G, x) , for all certificate :

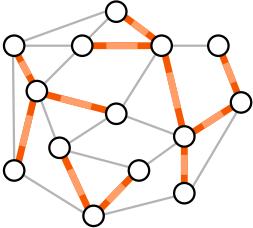
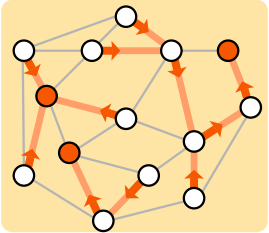
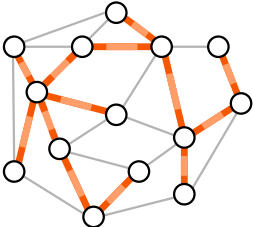
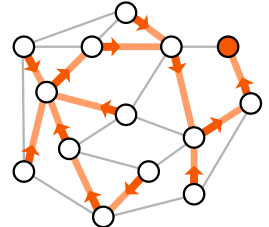
$$\#\{\text{Rejecting nodes}\} \geq \alpha \cdot d((G, x), \mathcal{L})$$

Examples

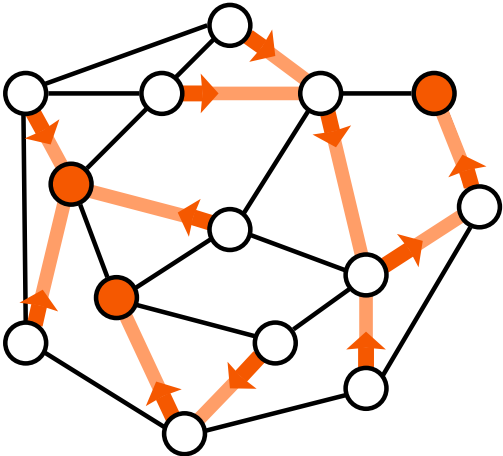
Acyclicity problems

	Adjacency lists	Pointers
Spanning forest	 A graph with 12 nodes and 18 edges. A spanning forest is highlighted in orange, consisting of 10 edges that connect all nodes without forming any cycles.	 The same graph as the adjacency lists version. The spanning forest edges are highlighted in orange and have arrows pointing from parent nodes to child nodes, indicating a directed forest structure.
Spanning tree	 The same graph as the spanning forest version. The spanning tree is highlighted in orange, consisting of 11 edges that connect all nodes without forming any cycles.	 The same graph as the spanning tree version. The spanning tree edges are highlighted in orange and have arrows pointing from parent nodes to child nodes, indicating a directed tree structure.

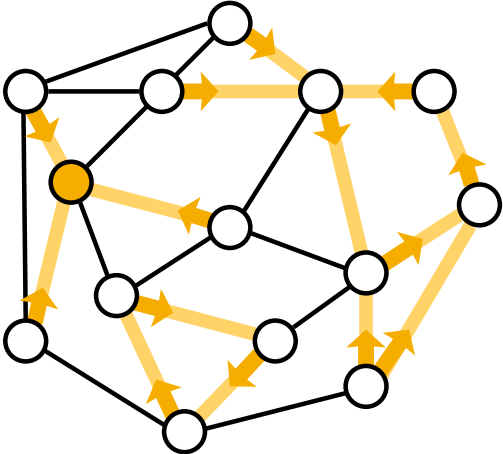
Acyclicity problems

	Adjacency lists	Pointers
Spanning forest	 A graph with 12 nodes and 18 edges. A spanning forest is highlighted in orange, consisting of three separate connected components: a path of 4 nodes, a path of 3 nodes, and a path of 5 nodes. No cycles are present.	 The same graph as the adjacency lists representation, but with directed edges (pointers) and a yellow background. The spanning forest is highlighted in orange. The nodes are colored white or orange. The graph contains cycles, such as a cycle of 4 nodes.
Spanning tree	 The same graph as the spanning forest, but with a single spanning tree highlighted in orange. This tree connects all 12 nodes and contains no cycles.	 The same graph as the spanning tree representation, but with directed edges (pointers) and a yellow background. The spanning tree is highlighted in orange. The nodes are colored white or orange. The graph contains cycles, such as a cycle of 4 nodes.

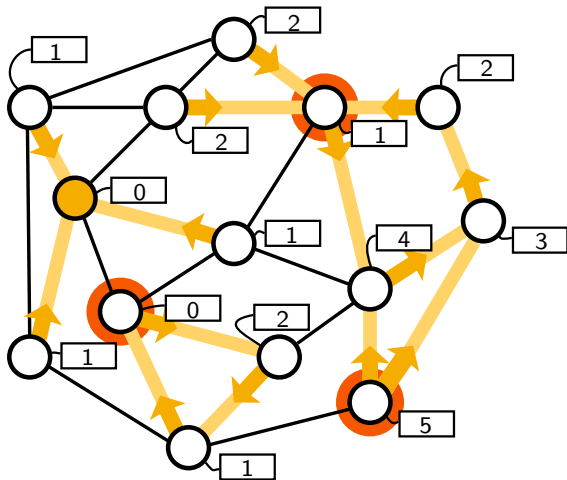
Spanning forest with pointers



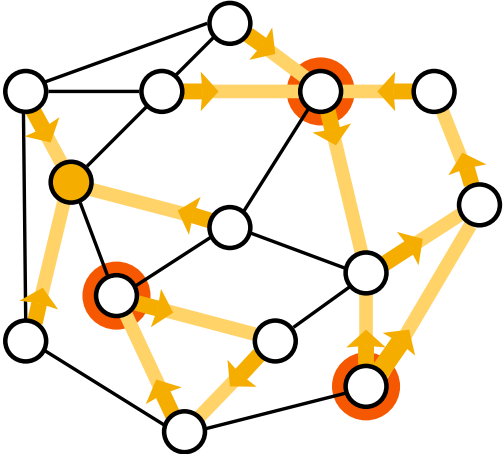
Spanning forest with pointers



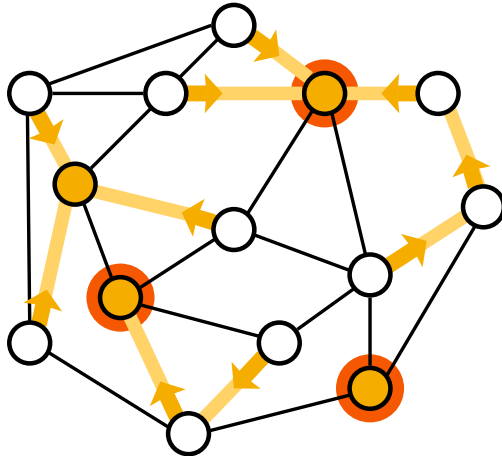
Spanning forest with pointers



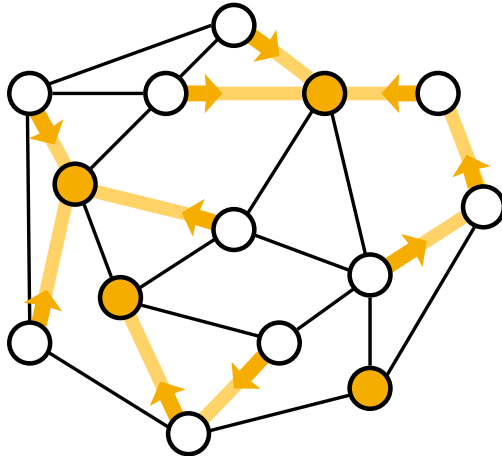
Spanning forest with pointers



Spanning forest with pointers



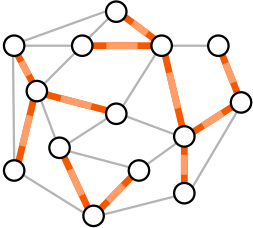
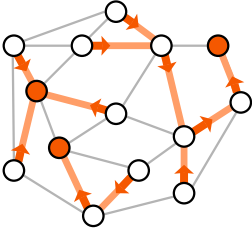
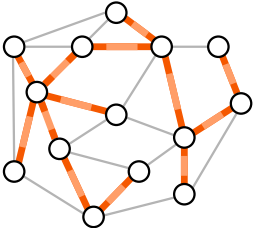
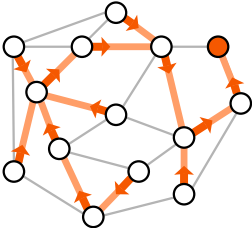
Spanning forest with pointers



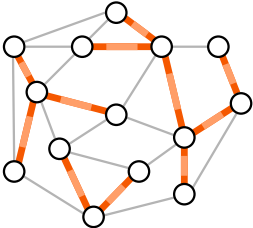
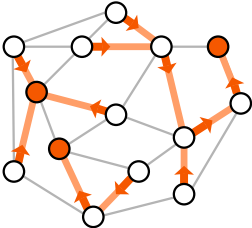
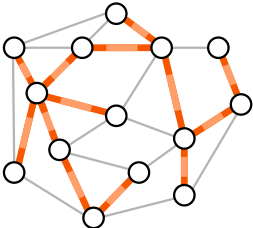
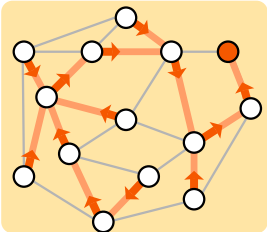
Spanning forest with pointers

Spanning forest with pointers
has an error-sensitive PLS.

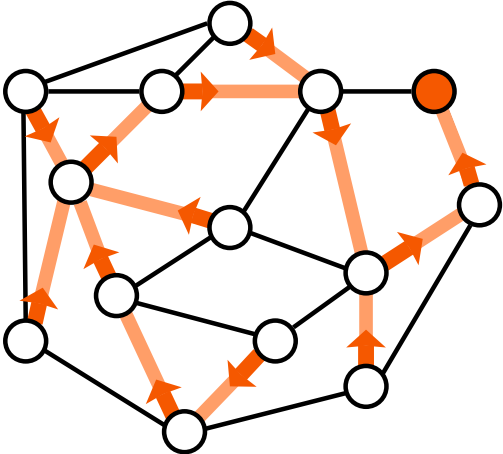
Acyclicity problems

	Adjacency lists	Pointers
Spanning forest	 A graph with 12 nodes and 18 edges. A spanning forest is highlighted in orange, consisting of three separate connected components: a path of 4 nodes, a path of 3 nodes, and a path of 5 nodes. No cycles are present.	 The same graph as the adjacency lists version. The spanning forest is highlighted in orange with arrows indicating direction. The nodes in the forest are white, while the nodes in the two cycles (one of length 3 and one of length 4) are filled orange. The arrows in the cycles point in a clockwise direction.
Spanning tree	 The same graph as the spanning forest version. A spanning tree is highlighted in orange, which is a single connected component containing all 12 nodes and 11 edges. No cycles are present.	 The same graph as the spanning tree version. The spanning tree is highlighted in orange with arrows indicating direction. The nodes in the tree are white, while the nodes in the two cycles (one of length 3 and one of length 4) are filled orange. The arrows in the cycles point in a clockwise direction.

Acyclicity problems

	Adjacency lists	Pointers
Spanning forest	 A graph with 12 nodes and 18 edges. A spanning forest is highlighted in orange, consisting of 5 trees with 10 nodes and 9 edges. The remaining 8 nodes and 9 edges are not part of the forest.	 The same graph as the adjacency lists view. The spanning forest is highlighted in orange. The nodes in the forest are white circles, and the nodes not in the forest are orange circles. Arrows on the orange edges indicate a direction, showing a forest of rooted trees.
Spanning tree	 The same graph as the spanning forest view. A spanning tree is highlighted in orange, consisting of 12 nodes and 11 edges. The remaining 7 edges are not part of the tree.	 The same graph as the spanning tree view. The spanning tree is highlighted in orange. The nodes in the tree are white circles, and the nodes not in the tree are orange circles. Arrows on the orange edges indicate a direction, showing a rooted tree. The entire graph is highlighted with a yellow background.

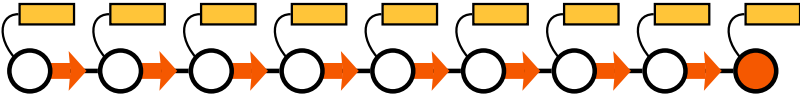
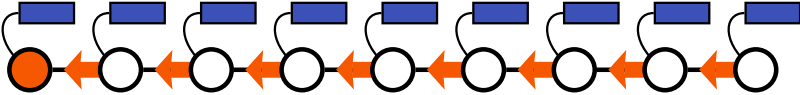
Spanning tree with pointers



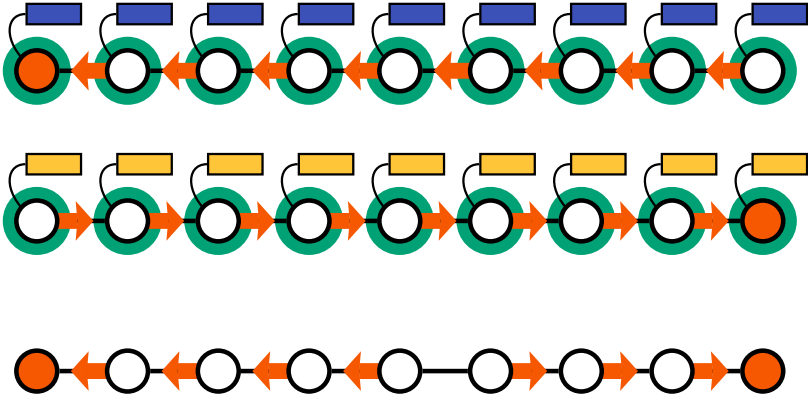
Spanning tree with pointers



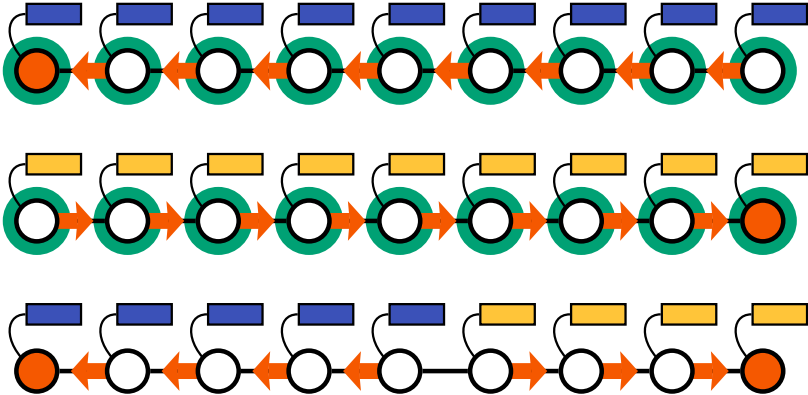
Spanning tree with pointers



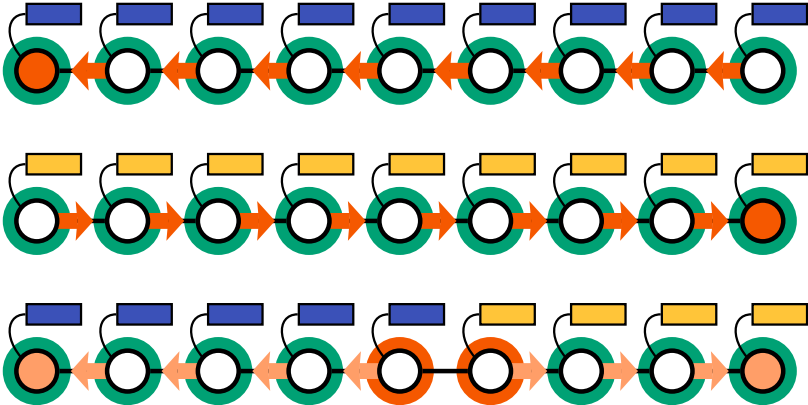
Spanning tree with pointers



Spanning tree with pointers



Spanning tree with pointers



Spanning tree with pointers

Spanning tree with pointers
has no error-sensitive PLS
(for any certificate size).

Structural characterization

Theorem

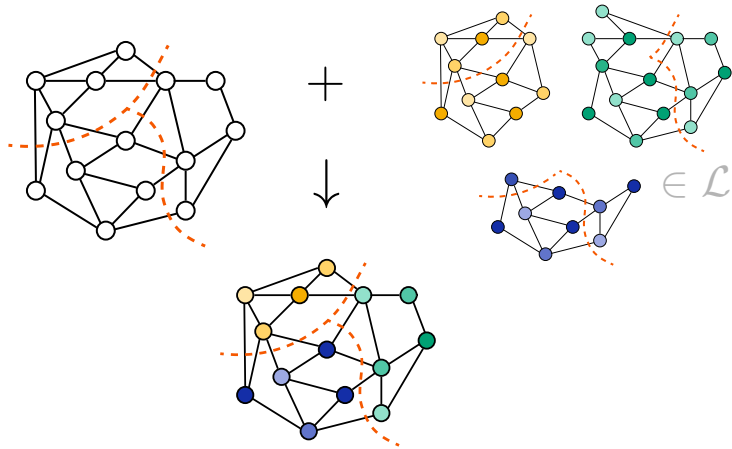
A language \mathcal{L} admits
an error-sensitive PLS



\mathcal{L} is **locally stable**

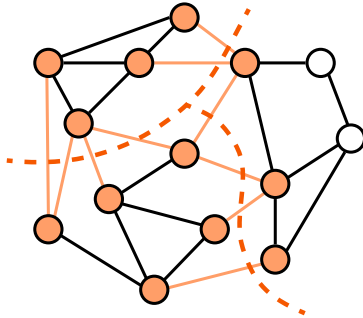
Local stability

Hybridization



Local stability

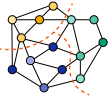
Boundary nodes



Local stability

\mathcal{L} is locally stable if :

$\exists \beta, \forall G, \forall$ hybridization,

$$d(\text{graph}, \mathcal{L}) \leq \beta \cdot \#\{\text{graph}\}$$
A diagram of a graph with 10 nodes. The nodes are colored in a pattern: blue, green, yellow, and orange. Dashed orange lines connect some of the nodes, representing a specific hybridization or a set of edges.

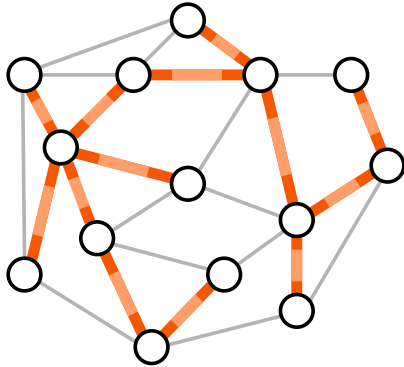
$$d(\text{hybrid}, \mathcal{L}) \leq \beta \cdot \#\{\text{Boundary nodes}\}$$
A diagram of a graph with 10 nodes, all colored orange. Dashed orange lines connect some of the nodes, representing a specific hybridization or a set of edges.

Spanning tree with pointers

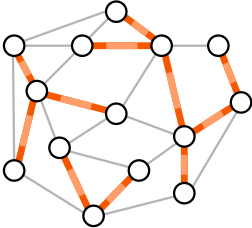
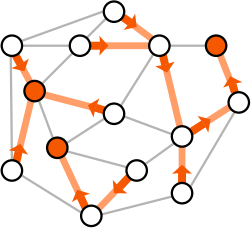
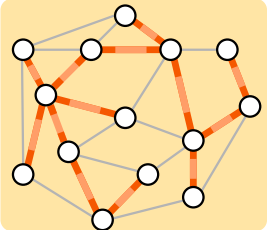
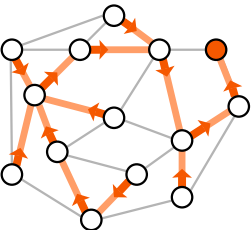
is not locally stable



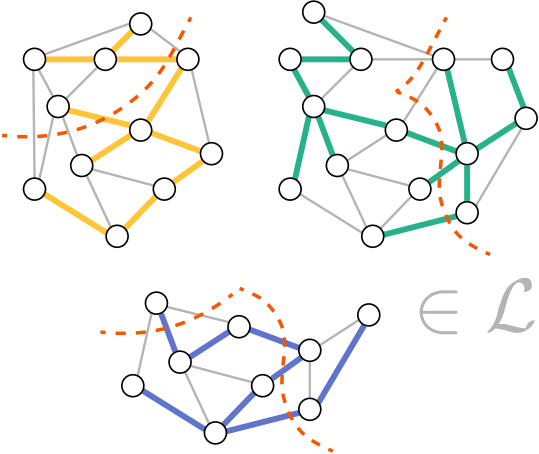
Spanning tree with adjacency lists is locally stable



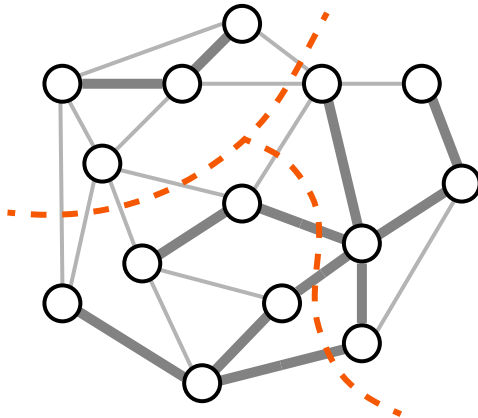
Acyclicity problems

	Adjacency lists	Pointers
Spanning forest	 A graph with 12 nodes and 18 edges. A spanning forest is highlighted in orange, consisting of 5 trees with 10 nodes and 9 edges. The remaining 8 nodes and 9 edges are not part of the forest.	 The same graph as the adjacency lists version. The spanning forest is highlighted in orange. The nodes in the forest are white, while the nodes not in the forest are colored orange. Arrows on the orange edges indicate a direction, likely representing a DFS tree.
Spanning tree	 The same graph as the spanning forest version. The entire graph is highlighted in orange, representing a spanning tree. The background of this cell is yellow.	 The same graph as the spanning forest version. The entire graph is highlighted in orange. The nodes in the tree are white, while the nodes not in the tree are colored orange. Arrows on the orange edges indicate a direction, likely representing a DFS tree.

Spanning tree with adjacency lists



Spanning tree with adjacency lists



With adjacency lists

Thm : With **adjacency lists**,
spanning tree and minimum
spanning tree, are locally stable.

\Rightarrow they have error-sensitive PLS.

Compact schemes

Compact PLS

Theorem (Korman et al.) :

- ▶ ST has a $O(\log n)$ -PLS;
- ▶ MST has a $O(\log^2 n)$ -PLS.

Compact PLS

New Theorem :

- ▶ ST has a $O(\log n)$ -ESPLS ;
- ▶ MST has a $O(\log^2 n)$ -ESPLS.

Open problem

Does error-sensitivity always come for free
(when achievable)?