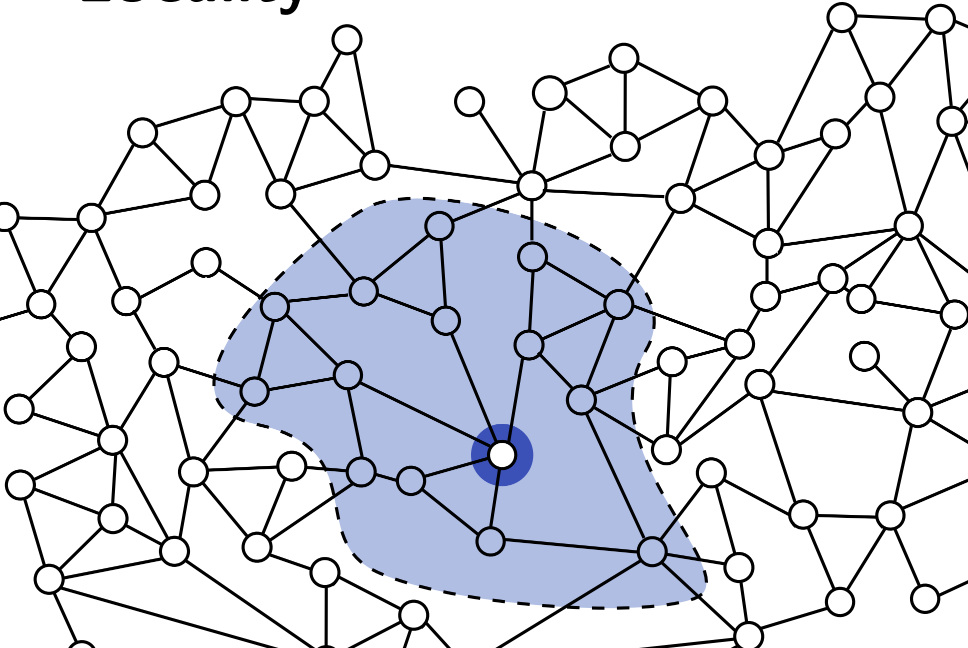
# Error-sensitive proof-labeling schemes

Laurent Feuilloley
joint work with Pierre Fraigniaud
Université Paris Diderot

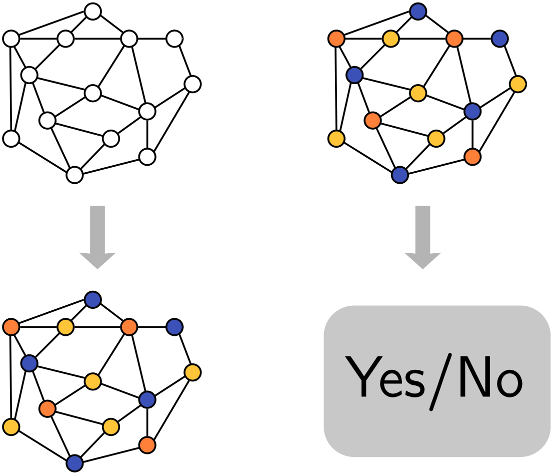
2017

# Locality



# Distributed decision

# Building vs. deciding



# Distributed languages

Context :

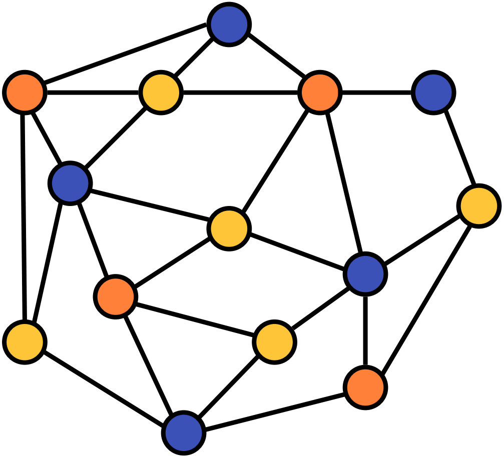
- Communication graph $G$
- Node inputs, $x : v \mapsto x(v)$

A language is

a set of configurations $(G, x)$

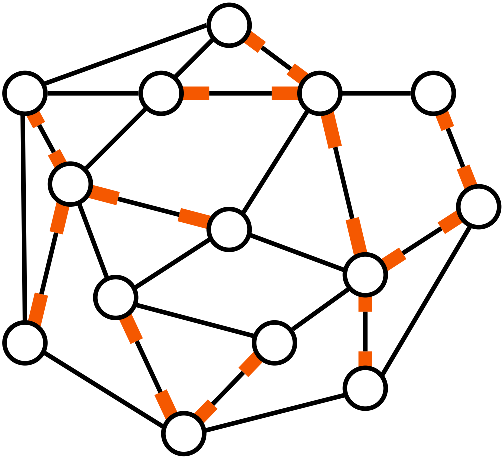
s.t. $\forall G, \exists x, (G, x) \in \mathcal{L}$

# Properly-colored graphs



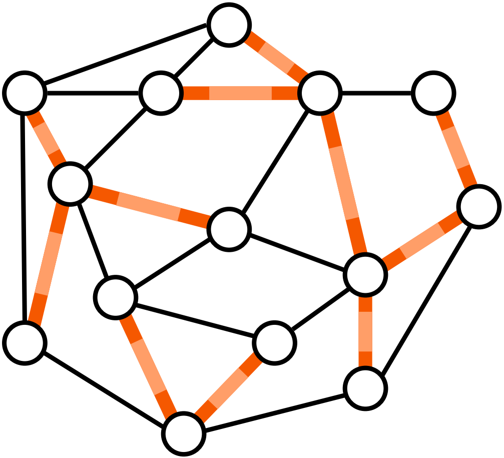
$\mathcal{L} = \{(G, x) \text{ s.t. } x \text{ is a proper coloring of } G\}$

# Spanning forest



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

# Spanning forest

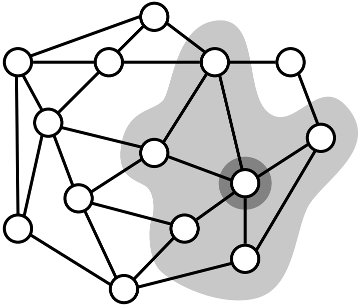


$\mathcal{L} = \{(G, x) \text{ s.t. } x \text{ describes a spanning forest of } G\}$

# Decision mechanism

Every node :

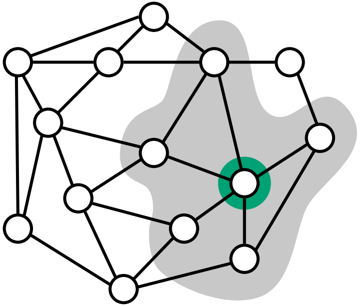
- gathers its 1-neighbourhood
- outputs a local decision accept or reject.



# Decision mechanism

Every node :

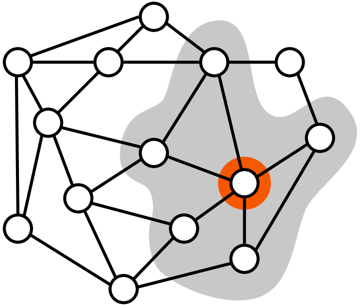
- gathers its 1-neighbourhood
- outputs a local decision accept or reject.



# Decision mechanism

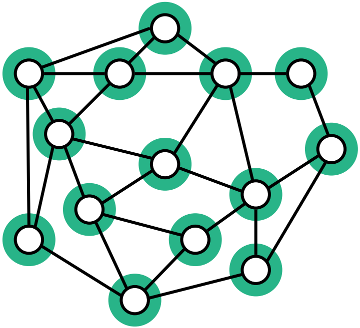
Every node :

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



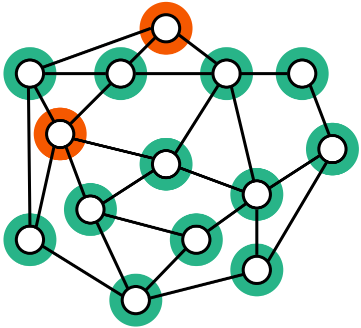
# Decision mechanism

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

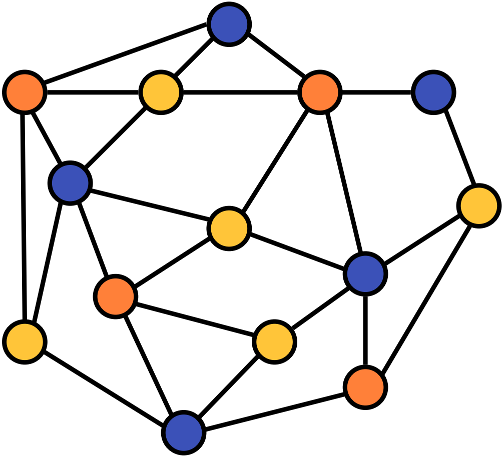


# Decision mechanism

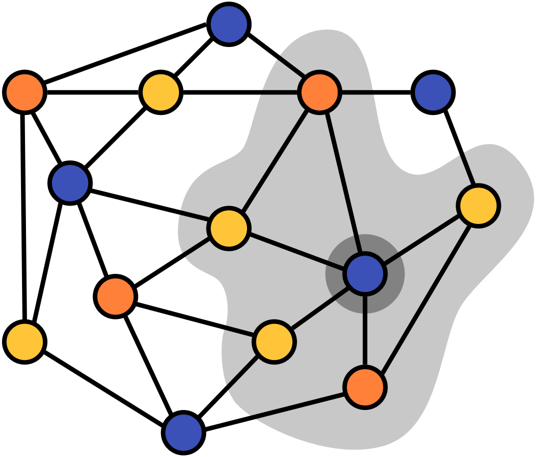
$(G, x)$ is rejected if at least one node rejects.



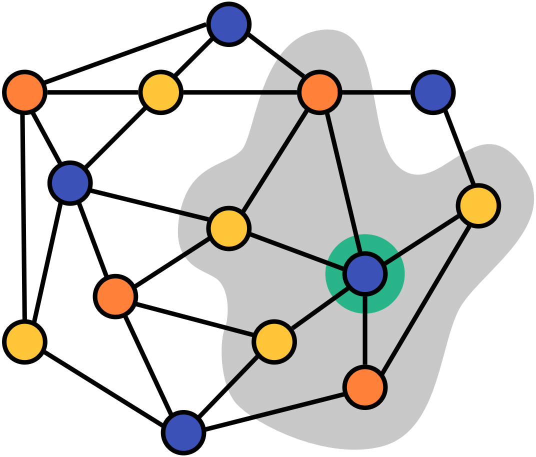
# Properly-colored graphs



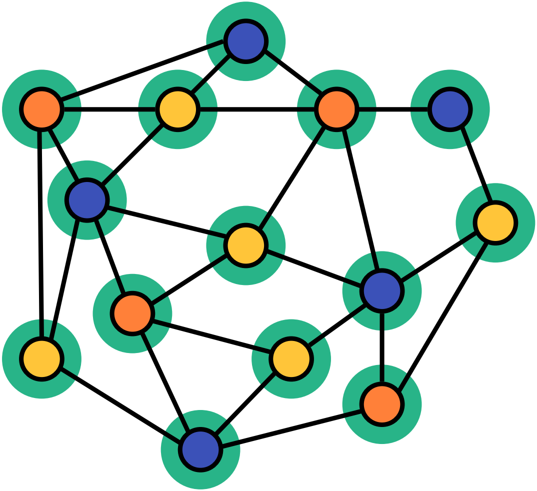
# Properly-colored graphs



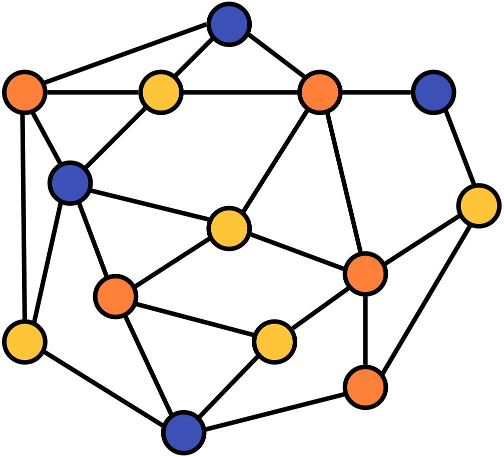
# Properly-colored graphs



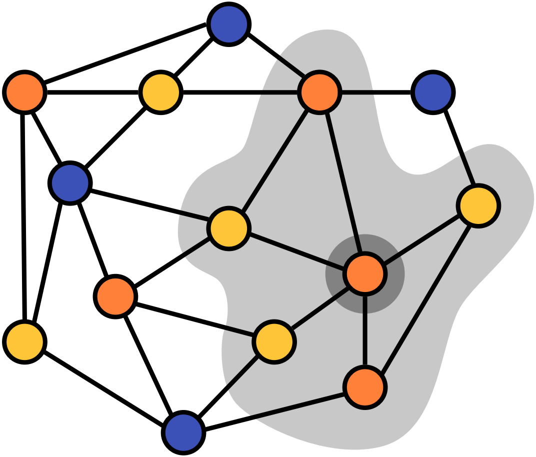
# Properly-colored graphs



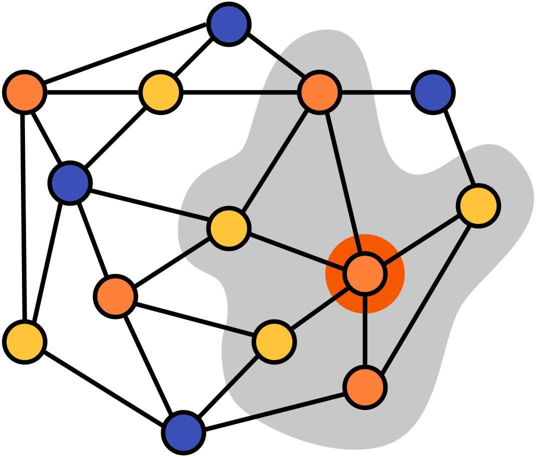
# Properly-colored graphs



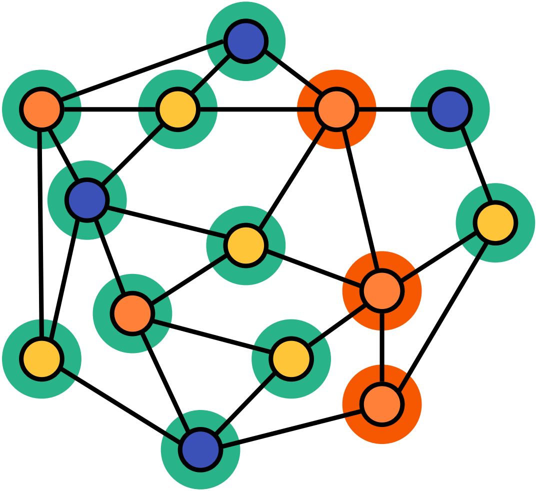
# Properly-colored graphs



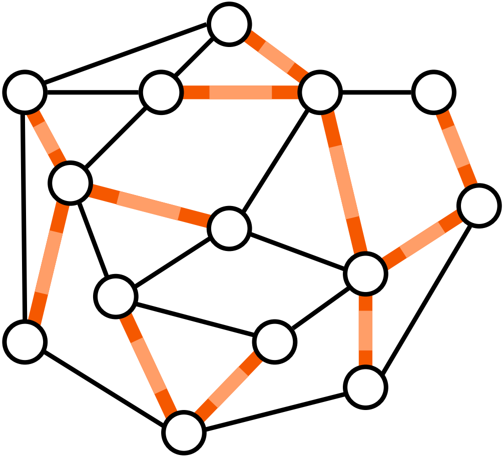
# Properly-colored graphs



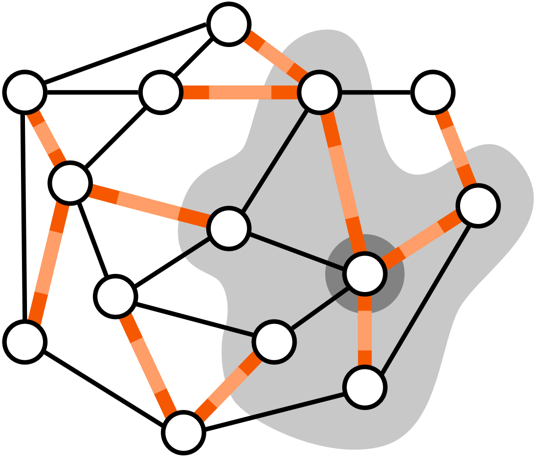
# Properly-colored graphs



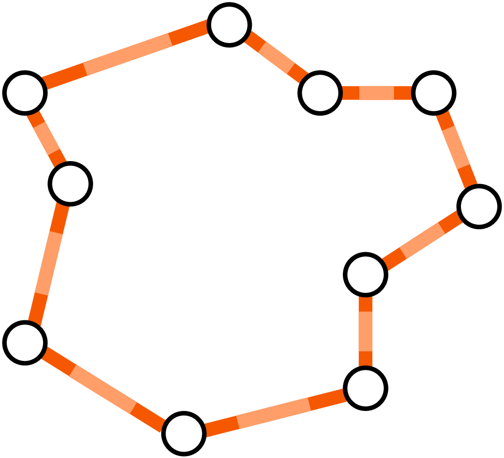
# Spanning forest



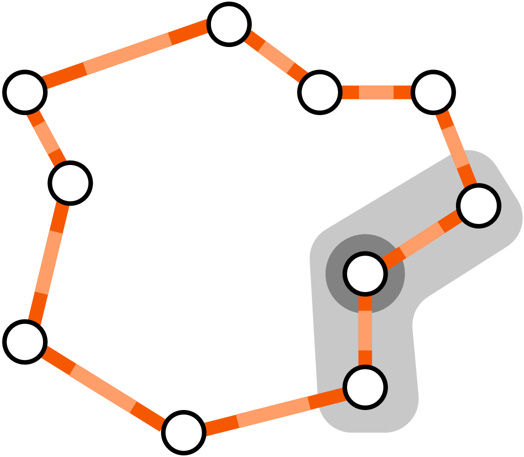
# Spanning forest



# Spanning forest



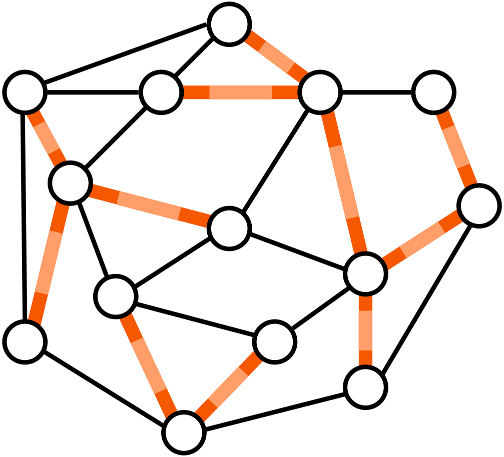
# Spanning forest



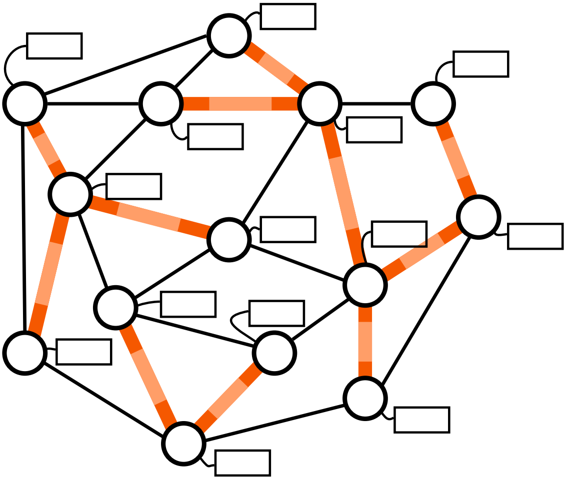
# Proof-labeling schemes

## Distributed non-determinism

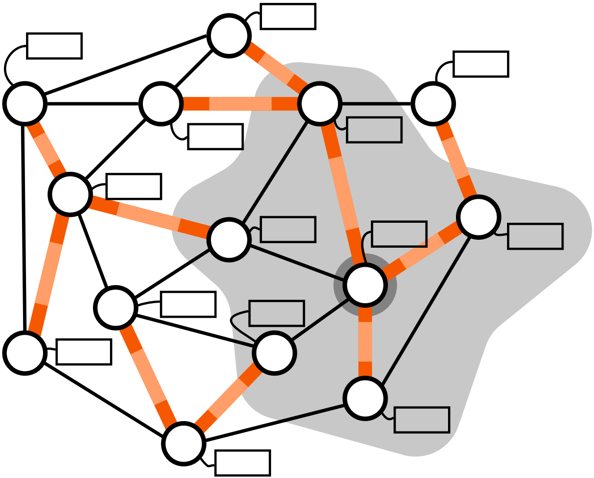
# Proof-labeling schemes



# Proof-labeling schemes



# Proof-labeling schemes



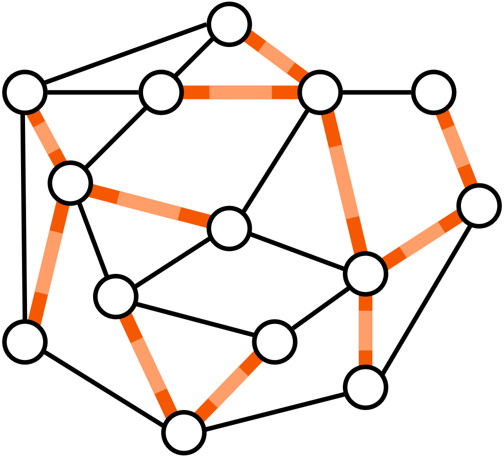
# Proof-labeling schemes

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

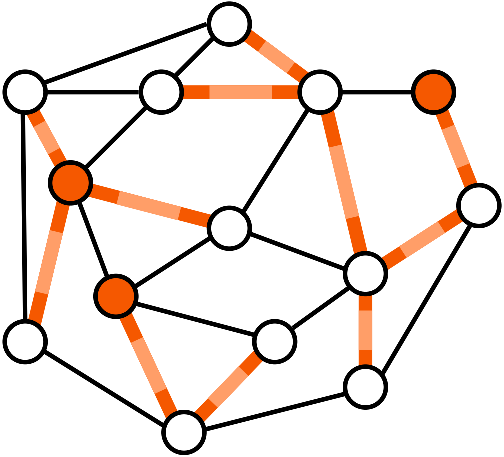
For all $(G, x)$ :

- 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.

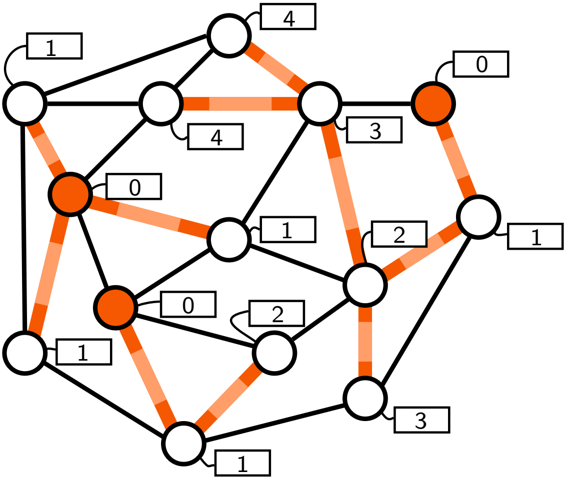
# Proof-labeling schemes



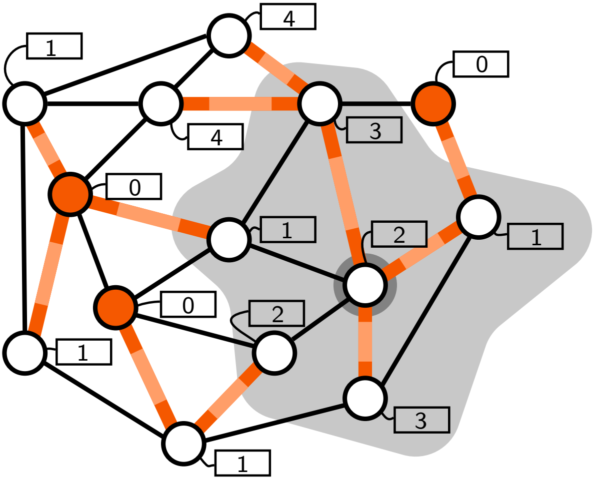
# Proof-labeling schemes



# Proof-labeling schemes

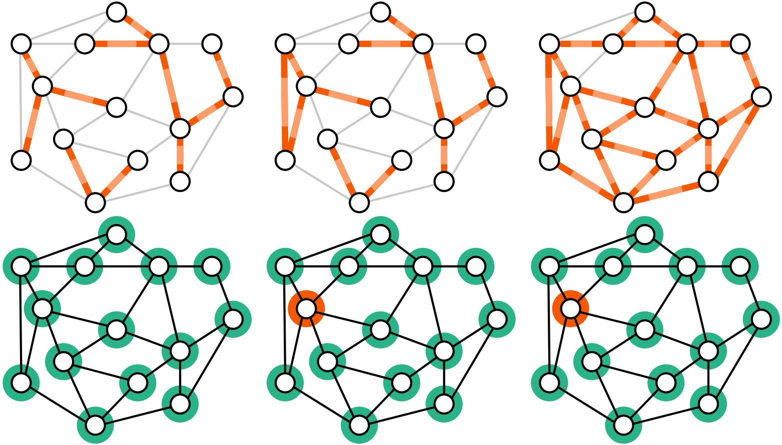


# Proof-labeling schemes

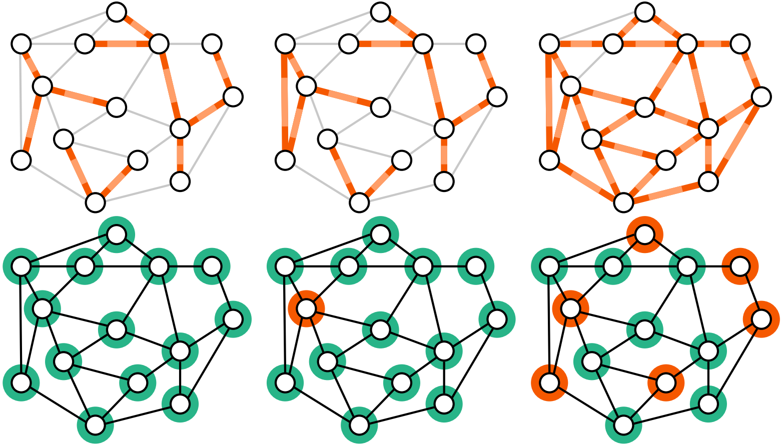


# Error-sensitivity
## of proof-labeling schemes

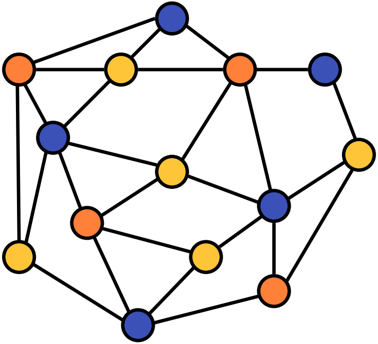
# One node to reject



# More nodes to reject

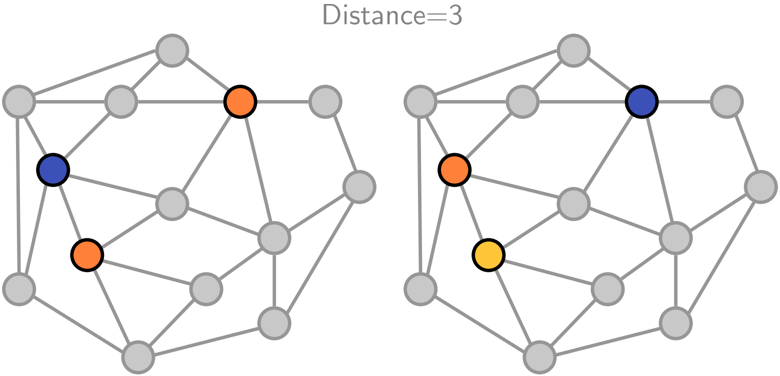


# Distance



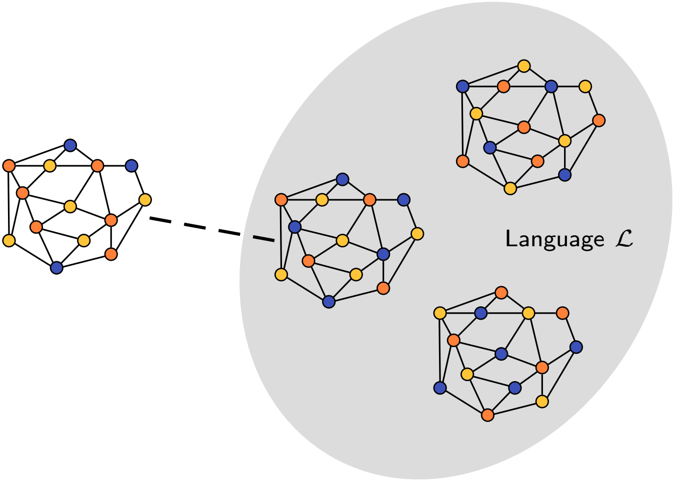
$$d((G, x_1), (G, x_2)) = \#\{v : x_1(v) \neq x_2(v)\}$$

# Distance




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

# 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.

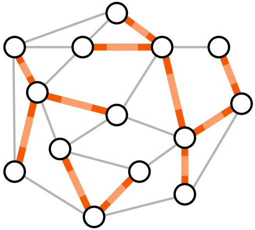
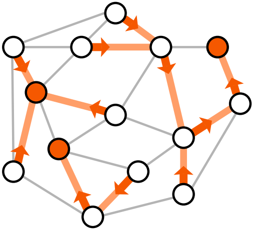
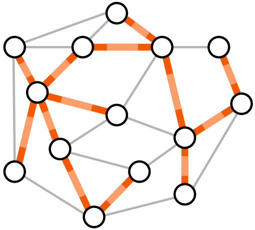
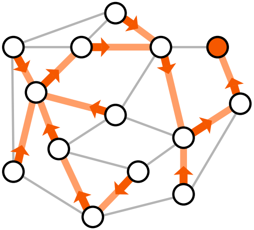
# Error-sensitivity
## with a formula

A PLS is error-sensitive if
there exists $\alpha > 0$ s.t.,
for all $(G, x)$, for all certificate :

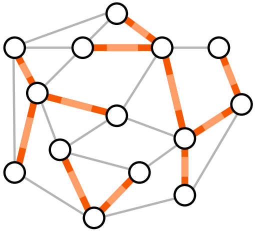
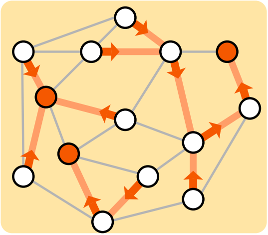
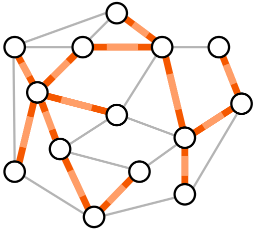
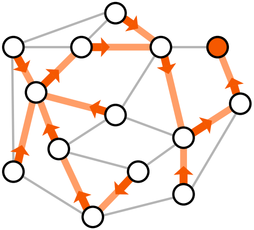
$$\#\{\text{Rejecting nodes}\} \geq \alpha \cdot d((G, x), \mathcal{L})$$

# Examples

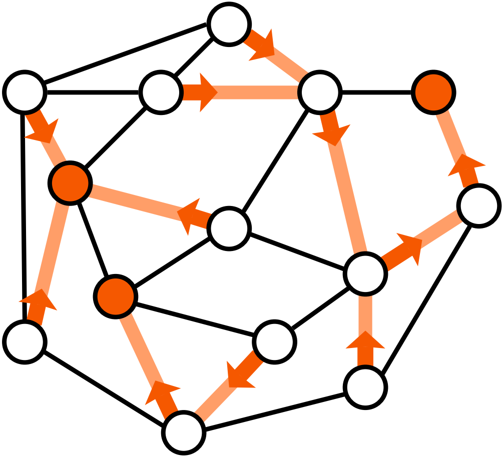
# Acyclicity problems

| | Adjacency lists | Pointers |
|---|---|---|
| Spanning forest |  |  |
| Spanning tree |  |  |

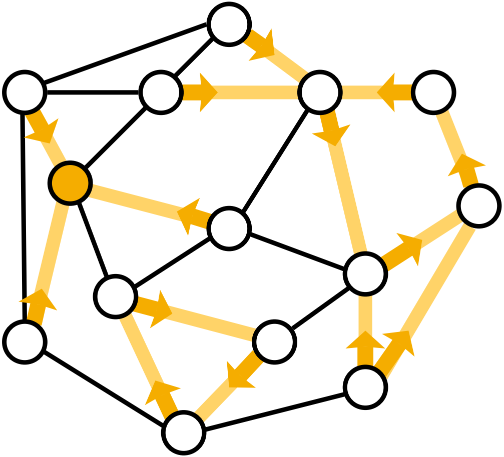
# Acyclicity problems

| | Adjacency lists | Pointers |
|---|---|---|
| Spanning forest |  |  |
| Spanning tree |  |  |

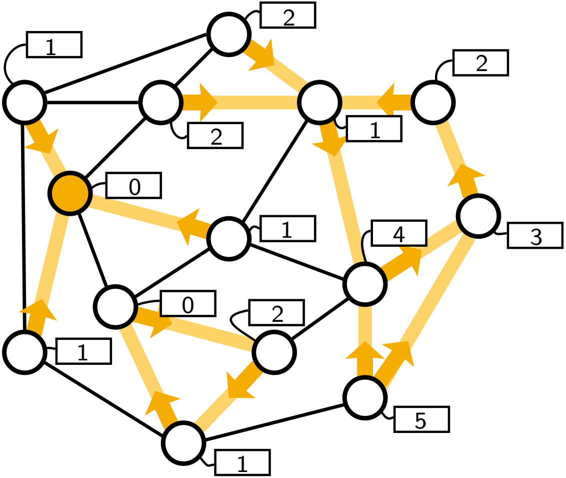
# Spanning forest with pointers



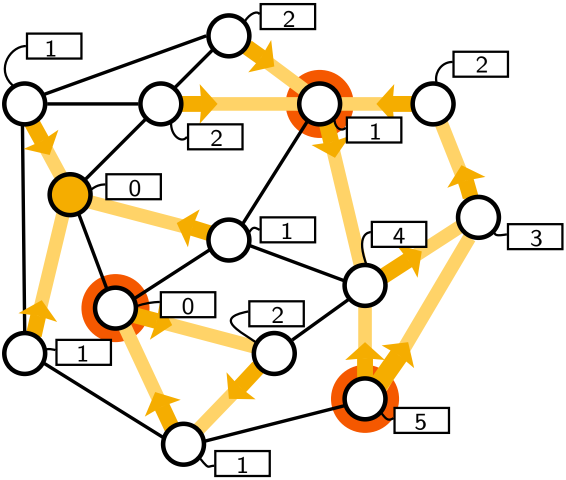
# Spanning forest with pointers



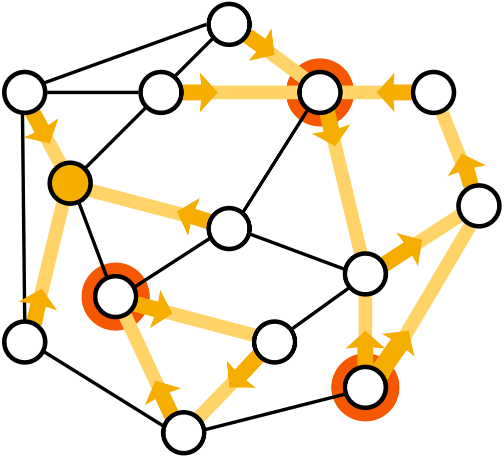
# Spanning forest with pointers



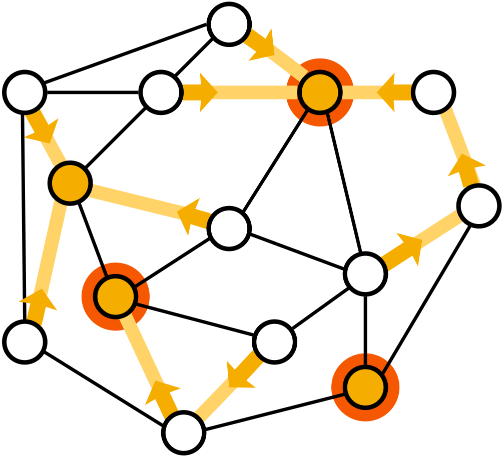
# Spanning forest with pointers



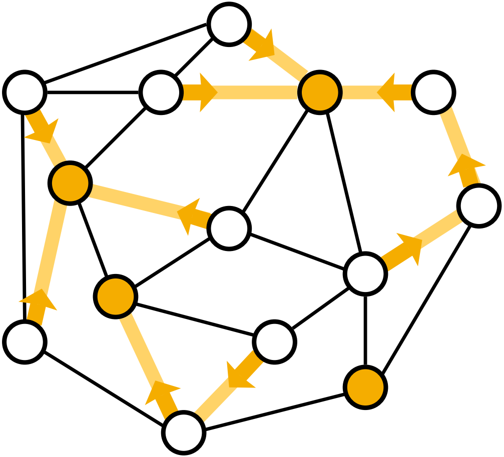
# Spanning forest with pointers



# Spanning forest with pointers



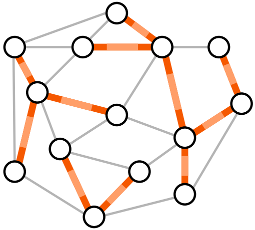
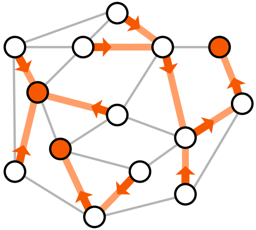
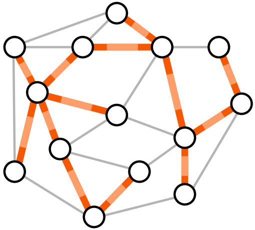
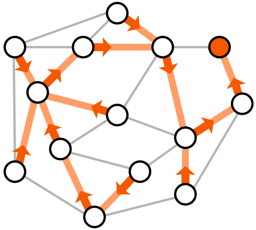
# Spanning forest with pointers



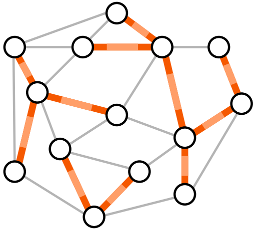
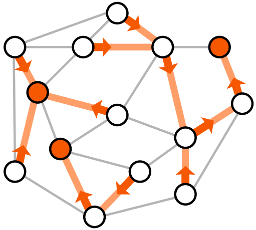
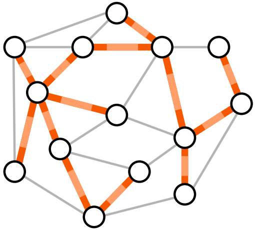
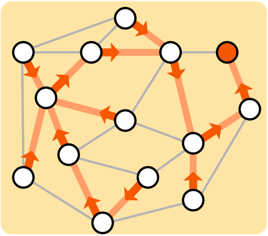
# Spanning forest with pointers

Spanning forest with pointers
has an error-sensitive PLS.

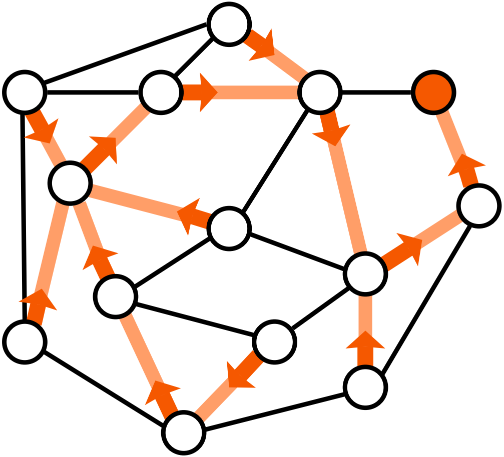
# Acyclicity problems

| | Adjacency lists | Pointers |
|---|---|---|
| Spanning forest |  |  |
| Spanning tree |  |  |

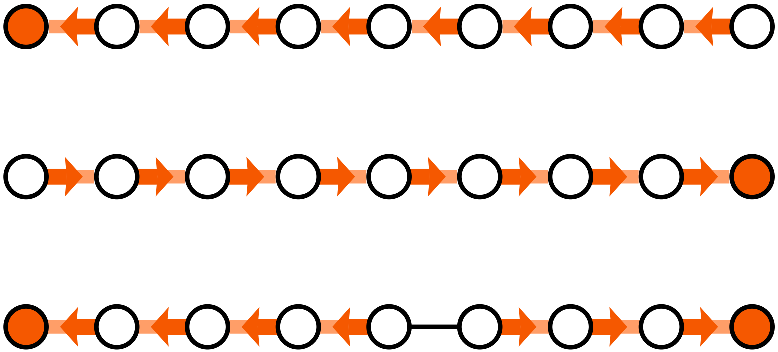
# Acyclicity problems

| | Adjacency lists | Pointers |
|---|---|---|
| Spanning forest |  |  |
| Spanning tree |  |  |

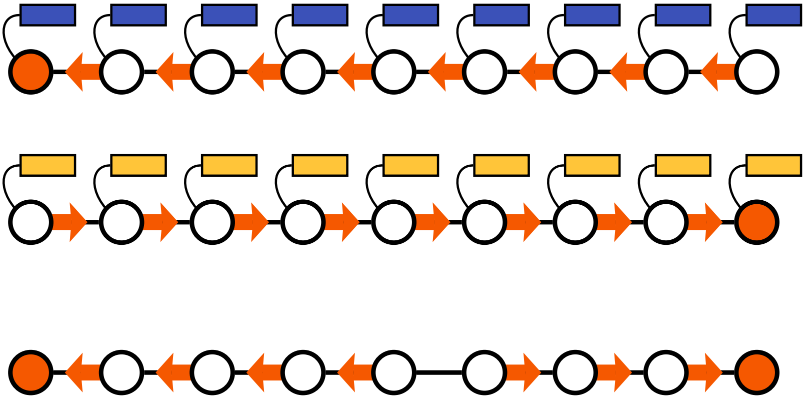
# Spanning tree with pointers



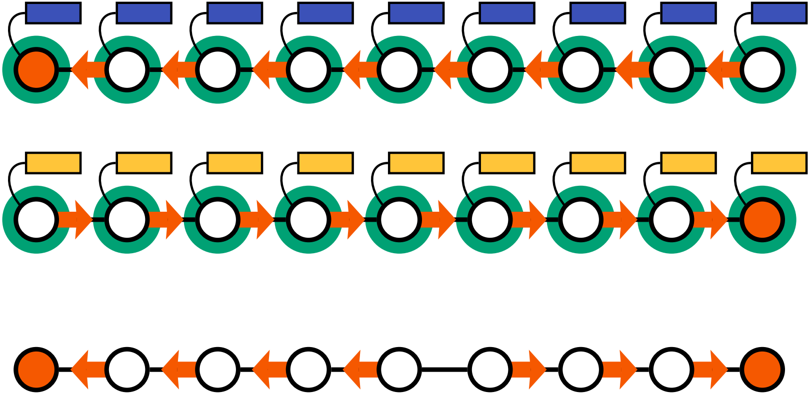
# Spanning tree with pointers



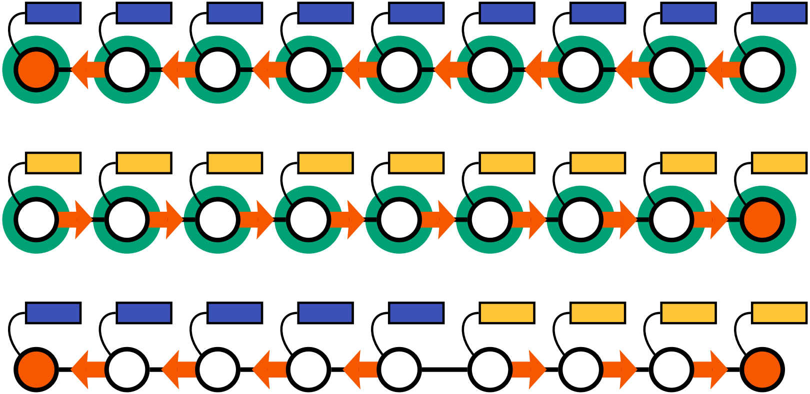
# Spanning tree with pointers



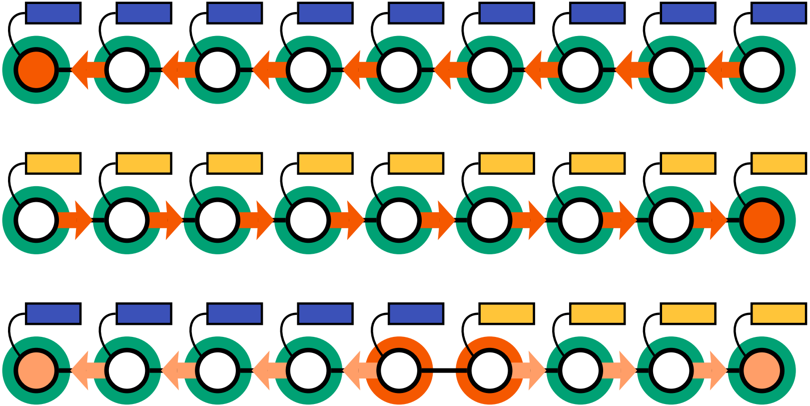
# Spanning tree with pointers



# 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

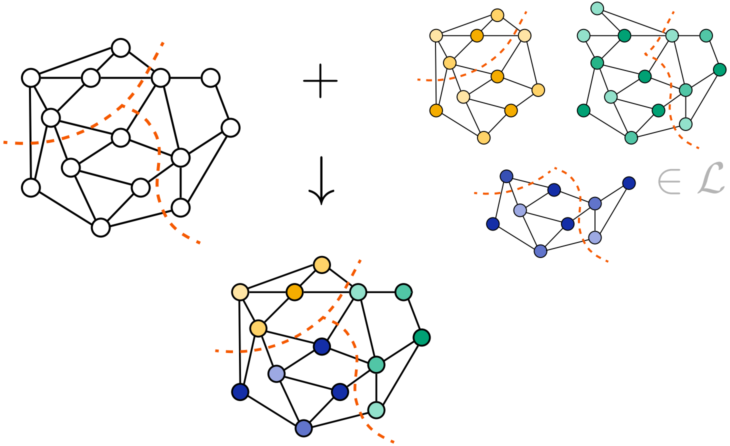
A language $\mathcal{L}$ admits
an error-sensitive PLS

$$\Leftrightarrow$$

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

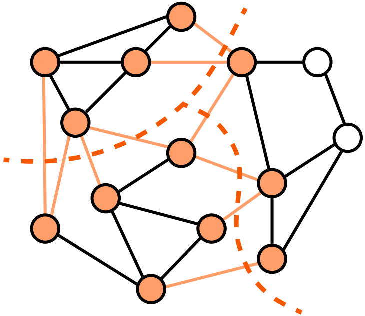
# Local stability

## Hybridization



# Local stability

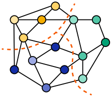
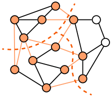
## Boundary nodes



# Local stability

$\mathcal{L}$ is locally stable if :

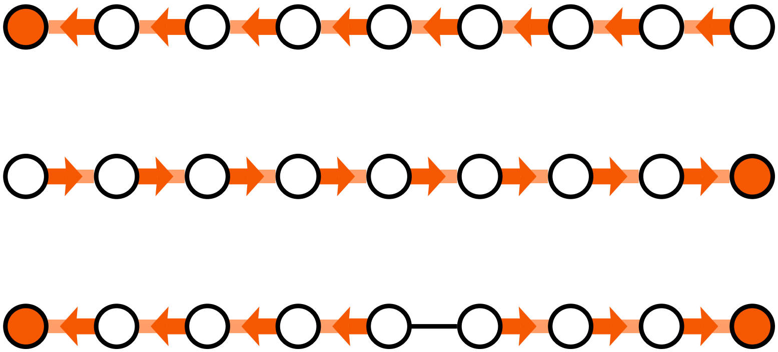
$\exists \beta, \forall G, \forall$ hybridization,

$d($  $, \mathcal{L}) \leq \beta \cdot \#\{$  $\}$

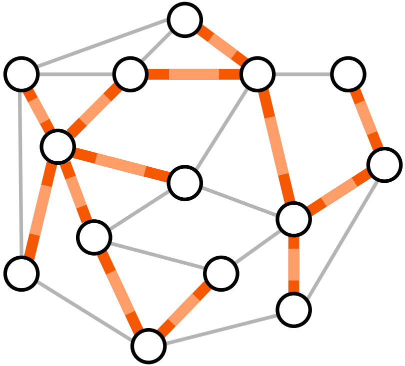
$d(\text{hybrid}, \mathcal{L}) \leq \beta \cdot \#\{\text{Boundary nodes}\}$

# Spanning tree with pointers

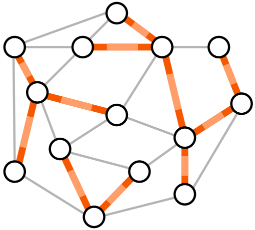
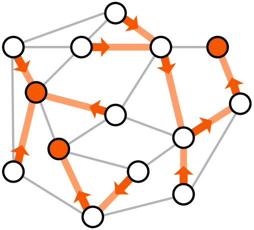
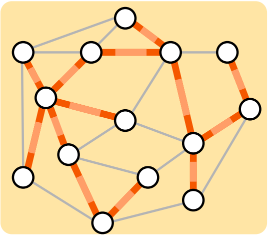
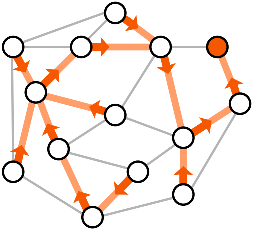
**is not locally stable**



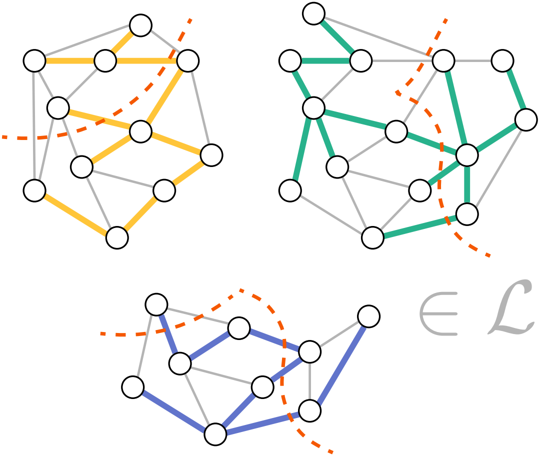
# Spanning tree with adjacency lists is locally stable



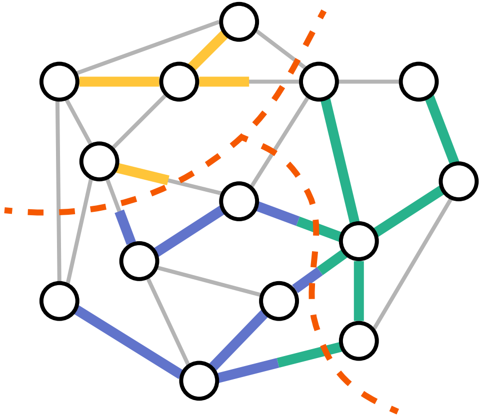
# Acyclicity problems

| | Adjacency lists | Pointers |
|---|---|---|
| Spanning forest |  |  |
| Spanning tree |  |  |

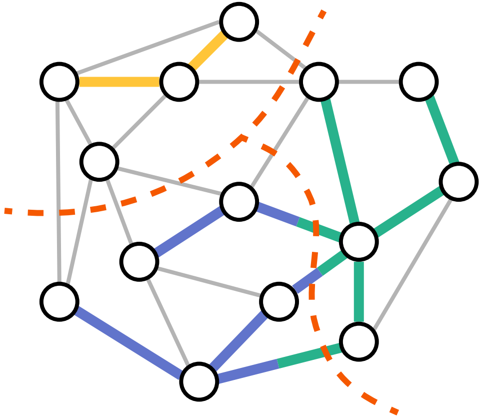
# Spanning tree with adjacency lists



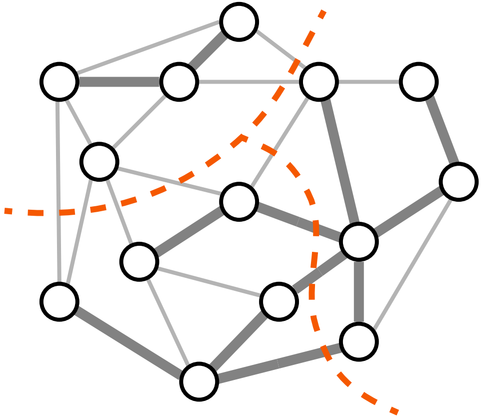
# Spanning tree with adjacency lists



# 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) ?