Local certification in distributed computing

Error-sensitivity, uniformity, redundancy, and interactivity

Laurent Feuilloley

PhD Defense

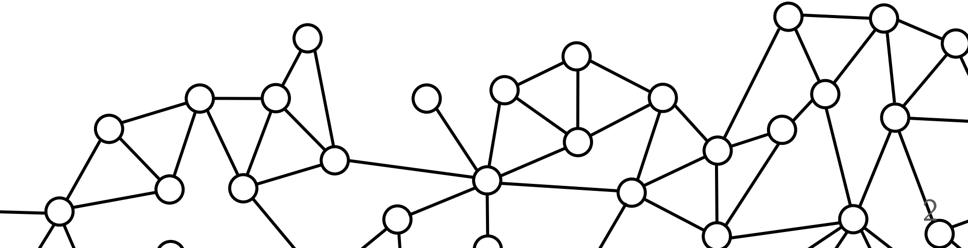
Supervised by Pierre Fraigniaud

Université Paris Diderot · 19th September 2018

Distributed network computing

Distributed network computing

- n machines, interacting during computation, no coordinator.
- Linked together by communication channels.
- Network represented by a graph.

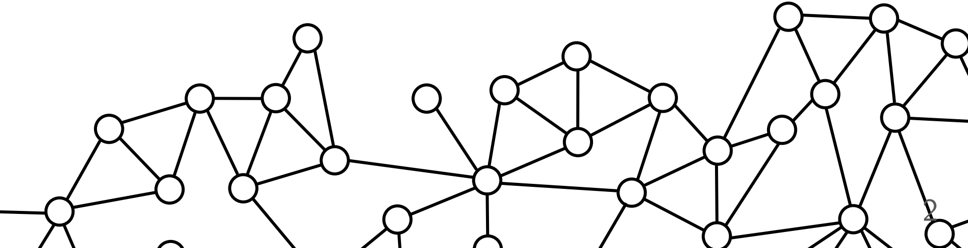


Distributed network computing

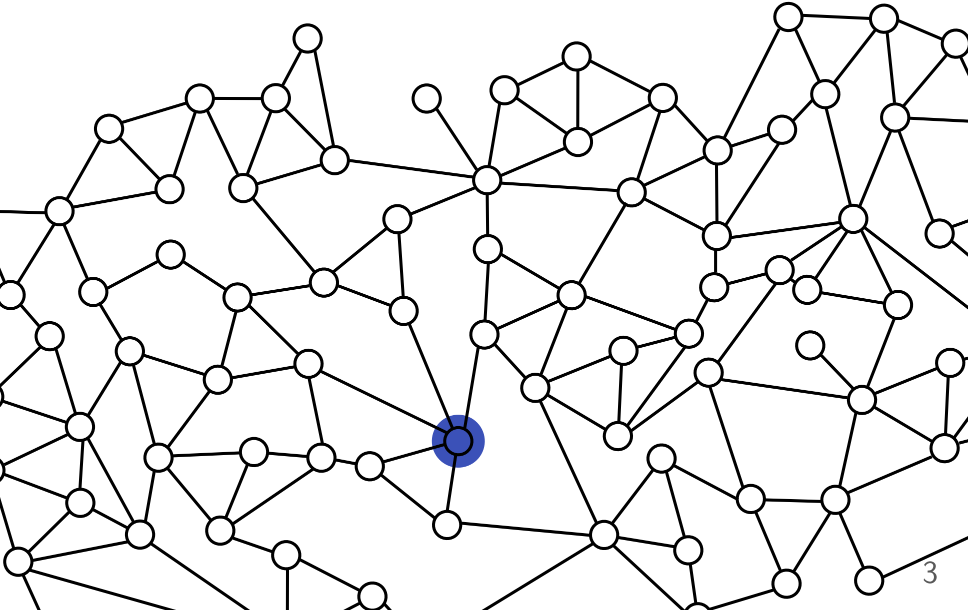
- n machines, interacting during computation, no coordinator.
- Linked together by communication channels.
- Network represented by a graph.

LOCAL model [Linial 92, Naor-Stockmeyer 93, Peleg 00]

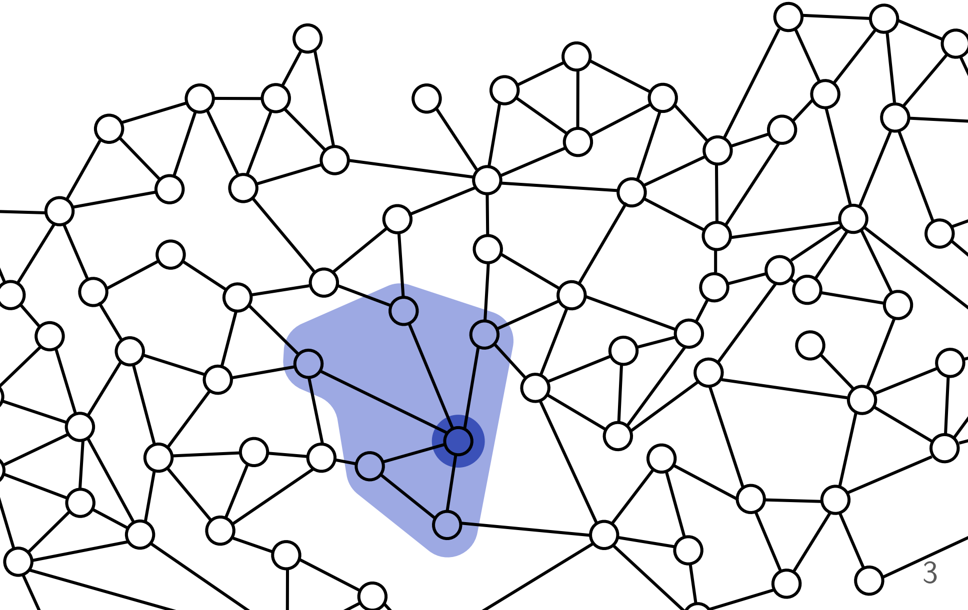
- Synchronous message-passing.
- No constraint on computational power and message size.
- Identifiers on $O(\log n)$ bits.
- Equivalent to a model with views.



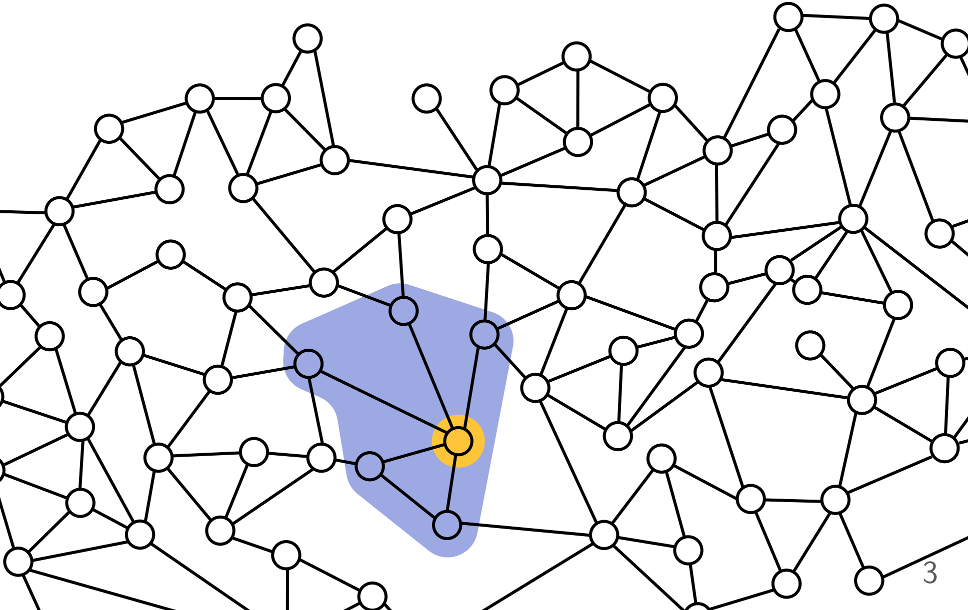
Distributed network computing



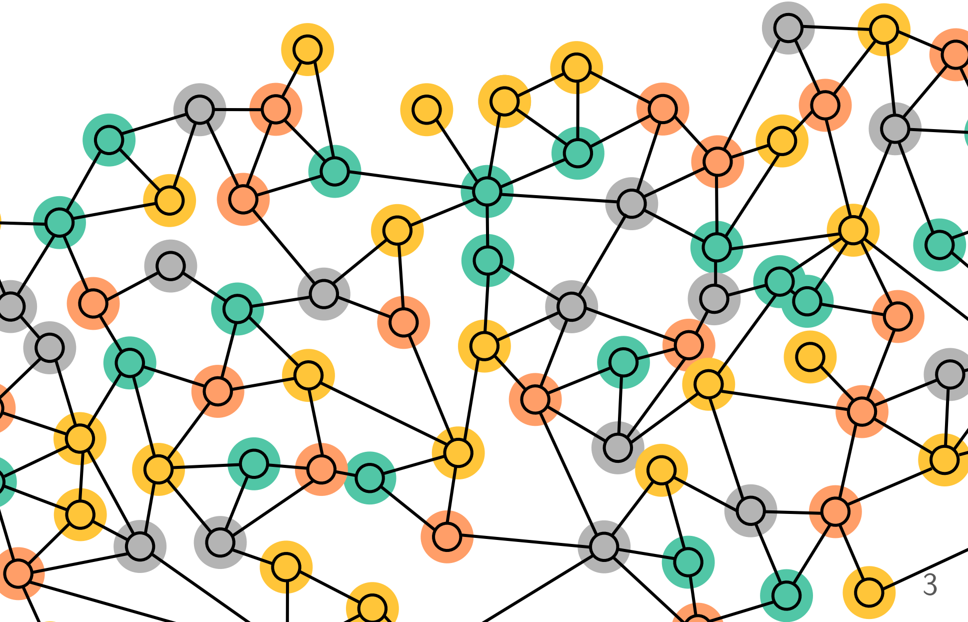
Distributed network computing



Distributed network computing



Distributed network computing



Local decision

Local decision : checking the status of the network. [Itkis-Levin 94, Awerbuch-Patt-Shamir-Varghese 91, Afek-Kutten-Yung 97].

- Motivated by fault-tolerance, in particular self-stabilizing algorithms [Dolev 00].
- A (more) universal framework [Fraigniaud, Korman, Peleg 11].
- Distributed complexity theory.

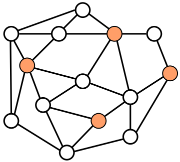
Formalization

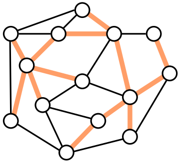
Definitions :

- A *configuration* is a pair (G, x), where G is the graph, and x an input assignment.
- A *language* is set of configurations.

The decision rule :

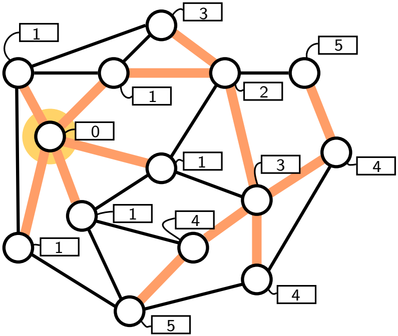
- Based on its 1-view, every node makes one (local) decision : accept or reject.
- The configuration is (globally) accepted if and only if it is (locally) accepted everywhere.





Local certification

Additional information at the nodes, certifying the configuration.
For spanning tree : distances and root-ID.



Formalization

Definition [Korman-Kutten-Peleg 05] : A *certificate* (or *proof*) assignment is a function $c : V \to \{0,1\}^*$, given by a *prover*.
A *certification scheme* is a couple (prover,verifier).

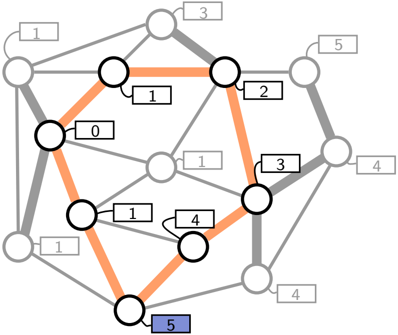
Correctness : A certification scheme is correct if, for all (G, x) :

$$(G, x) \in \mathcal{L} \Leftrightarrow \text{ there exists } c\text{, s.t. all nodes accept}$$

Like in NP.

Spanning tree

Theorem [Itkis-Levin 94] : The scheme with distances and root-ID is a correct certification scheme.



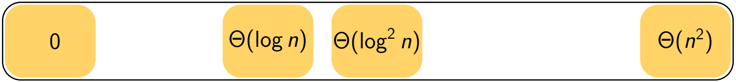
Certificate size

Definition : The *certificate size* of a language is the minimum certificate size of correct certification scheme.

$\hookrightarrow$ Certificate size is the cost of certification.

- Additional memory.
- Additional messages.
- More probability of corruption.

Certificate size



- [Naor-Stockmeyer 93] : LCL problems.
- [Korman, Kutten, Peleg 05] : formalization, $\Omega(\log n)$ for spanning tree, universal $O(n^2)$ scheme.
- [Korman, Kutten 06] $\Omega(\log^2 n)$ for minimum spanning tree.
- [Göös, Suomela 11] LogLCP, general model.

More previous works

- Impact of the identifier model [Fraigniaud, Hirvonen, Suomela 15]
- Randomization [Fraigniaud, Göös, Korman, Parter, Peleg 14], [Baruch, Fraigniaud, Patt-Shamir 15], [F., Fraigniaud 15]
- Message diversity [Patt-Shamir, Perry 17]
- Approximation [Censor-Hillel, Paz, Perry 17]
- Different decision mechanisms [Arfaoui, Fraigniaud, Pelc 13]
- Randomized interactivity [Kol, Oshman, Saxena 18]

$\hookrightarrow$ See *Survey of distributed decision* with P. Fraigniaud.

This thesis

Error-sensitivity
Uniformity
Redundancy
Interactivity

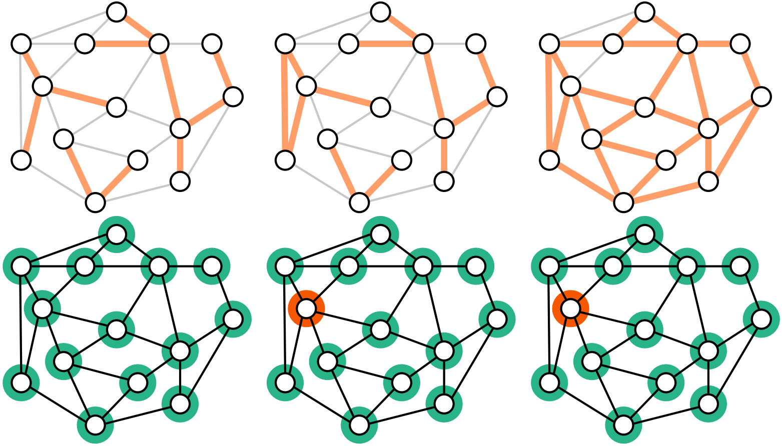
Part I

Error-sensitivity

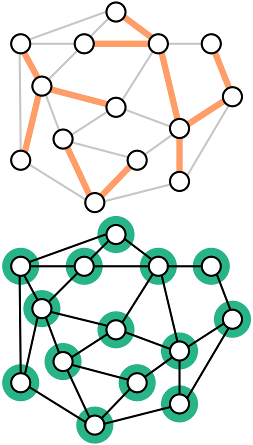
Error-sensitive proof-labeling scheme

with P. Fraigniaud. DISC 2017.

Motivation



Motivation



Formalization

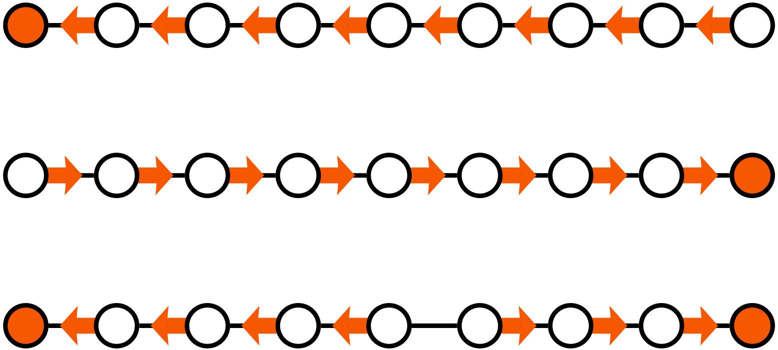
Definition Distance((G,x), $\mathcal{L}$) $=$ the minimum number of (node) inputs to change to get a configuration in $\mathcal{L}$.

Definition : A certification scheme for a language $\mathcal{L}$ is *error-sensitive* if for any configuration, for any certificate assignment :

$$\#(\text{Rejecting nodes}) \geq \mathit{Distance}((G,x),\mathcal{L})$$

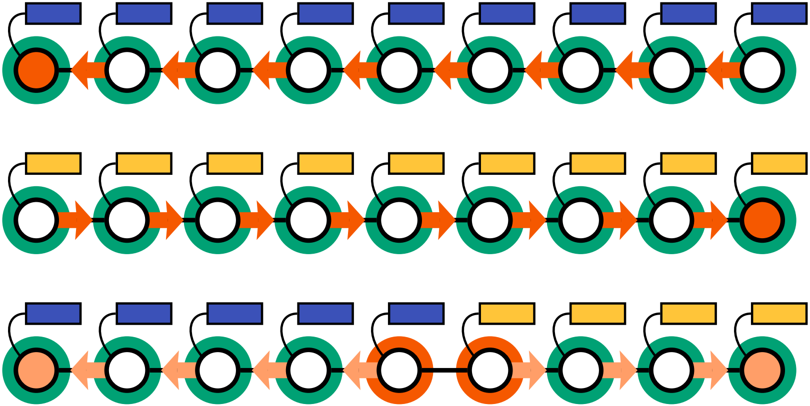
Not every language is sensitive

Theorem : The language of oriented paths is not error-sensitive.



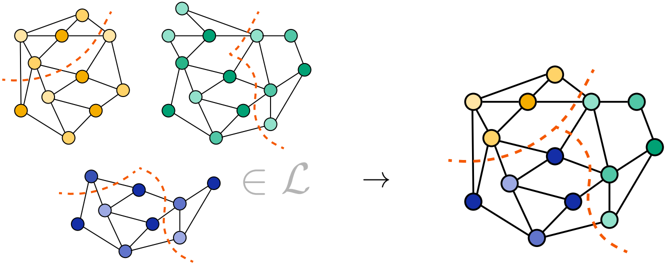
Not every language is sensitive

Theorem : The language of oriented paths is not error-sensitive.



Characterization

Definition : Hybrid.



Definition : $\mathcal{L}$ is locally stable if for any hybrid (G, h) :

$$d((G, h), \mathcal{L}) \leq \#\{\text{Border nodes}\}$$

Theorem : A language is error-sensitive iff it is locally stable.

Corollaries and certificate size

Corollary : The language of oriented paths is not error-sensitive.

Corollary : Spanning tree and minimum spanning tree are error-sensitive.

Theorem :
Spanning tree and minimum spanning tree have a error-sensitive schemes with certificate size $O(\log n)$ and $O(\log^2 n)$.

Open question : Can error-sensitivity require larger certificates (for locally stable languages) ?

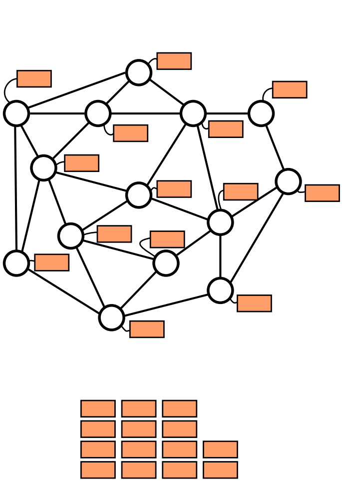
Part II

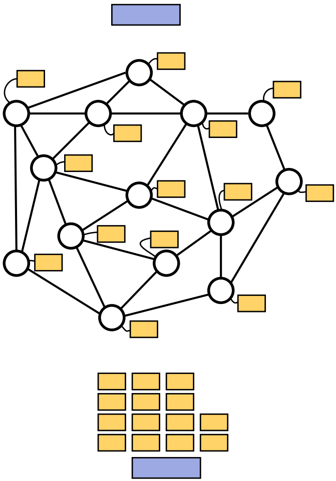
Uniformity

Local verification of global proofs

with J. Hirvonen. DISC 2018.

Motivation





Uniformity

Definition : The *uniformity* of a language is the ratio :

$$\frac{\sum_v |c(v)| \text{ (Classic scheme)}}{\sum_v |c_{Loc}(v)| + |c_{Glob}| \text{ (Mixed scheme)}}$$

Theorem : The uniformity is between 1 and n.
Definition : Two languages :

- AMOS : configurations where *at most* one node is selected.
- ALOS : configurations where *at least* one node is selected.

Theorem :

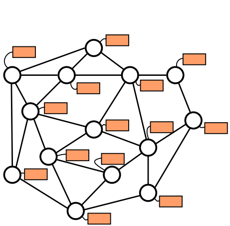
- AMOS has uniformity n.
- ALOS has uniformity $\Theta(1)$.

More results

Corollaries : The uniformity of spanning trees, non-bipartiteness, leader election is $\Theta(1)$.

Theorem : Minimum spanning tree has uniformity $\Theta(\log n)$.

Open question : Can purely global proofs be less efficient than purely local proofs ?



Part III

Redundancy

Redundancy in distributed proofs

with P. Fraigniaud, J. Hirvonen, A. Paz and M. Perry. DISC 2018.

Distance-r certification

- Trade-off between certificate size and radius.
- [Korman, Kutten, Masuzawa 11]
  $(\log n, \log n)$-scheme for minimum spanning tree.
- [Ostrovsky, Perry, Rosenbaum 17]
  Linear scaling for universal scheme and spanning trees.

Scaling

Definition : The *scaling* of a language is a function $f(r)$ s.t. :

$$\text{proof-size}(r) = \frac{\text{proof-size}(r = 1)}{f(r)}$$

We witness two main scenarios :

- Linear scaling : $f(r)$ is $\Theta(r)$.
- Maximum scaling : $f(r)$ is $\Theta(b(r))$,
  $b(r)$= minimum number of nodes in a ball of radius r.

Theorems

Theorem :

- Optimal uniform schemes imply a maximum scaling.
- Minimum spanning tree has a linear scaling.
- In paths, cycles, grids, torii, *any language* has a linear scaling.

Open question : does every language scales linearly ?

Part IV

Interactivity

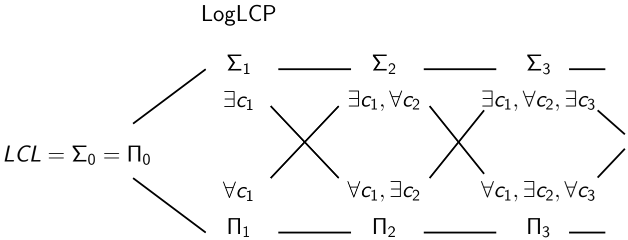
A hierarchy of local decision

with P. Fraigniaud and J. Hirvonen. ICALP 2016.

Local hierarchy

- LCL [Naor-Stockmeyer 93] :
  $\mathcal{L} \in P$: $\exists$ local algorithm $\mathcal{A}$:
  $x \in \mathcal{L} \leftrightarrow \mathcal{A}(x) =$ accept.
- LogLCP [Göös-Suomela 11] :
  $\mathcal{L} \in P$: $\exists$ local algorithm $\mathcal{A}$:
  $x \in \mathcal{L} \leftrightarrow \exists c,$ log-size $, \mathcal{A}(x, c) =$ accept.
- LH :
  $\mathcal{L} \in P$: $\exists$ local algorithm $\mathcal{A}$:
  $x \in \mathcal{L} \leftrightarrow \exists c_1, \forall c_2, \exists c_3 ...$ log-size $\mathcal{A}(x, c_1, c_2, c_3, ...) =$ accept.

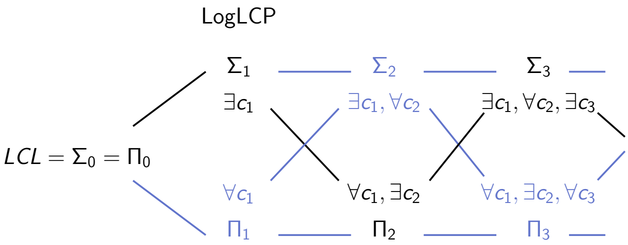
Local hierarchy structure



- Collapse of classes.
- But complement classes.
- MST$\in$ co-$\Sigma_1 \subseteq \Pi_2$ and ISO$\in$ co-$\Pi_2 \subseteq \Sigma_3$.
- No lower bound technique for higher levels.

Open problems : Is the hierarchy infinite ?

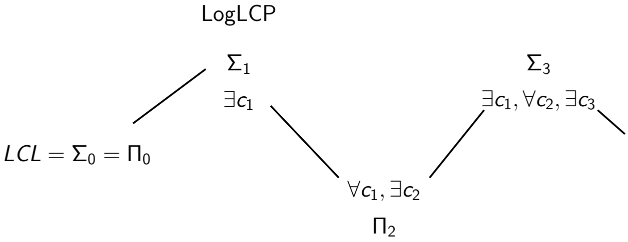
Local hierarchy structure



- Collapse of classes.
- But complement classes.
- MST$\in$ co-$\Sigma_1 \subseteq \Pi_2$ and ISO$\in$ co-$\Pi_2 \subseteq \Sigma_3$.
- No lower bound technique for higher levels.

Open problems : Is the hierarchy infinite ?

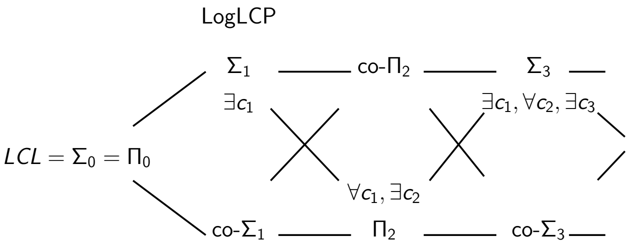
Local hierarchy structure



- Collapse of classes.
- But complement classes.
- MST$\in$ co-$\Sigma_1 \subseteq \Pi_2$ and ISO$\in$ co-$\Pi_2 \subseteq \Sigma_3$.
- No lower bound technique for higher levels.

Open problems : Is the hierarchy infinite ?

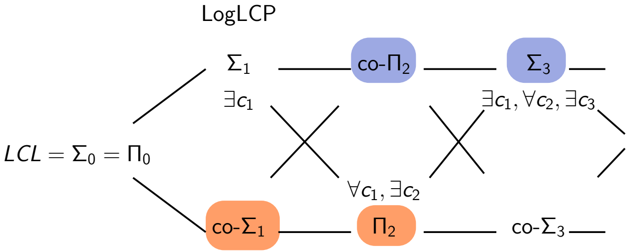
Local hierarchy structure



- Collapse of classes.
- But complement classes.
- MST$\in$ co-$\Sigma_1 \subseteq \Pi_2$ and ISO$\in$ co-$\Pi_2 \subseteq \Sigma_3$.
- No lower bound technique for higher levels.

Open problems : Is the hierarchy infinite ?

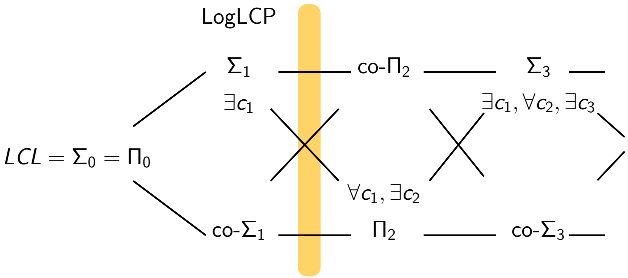
Local hierarchy structure



- Collapse of classes.
- But complement classes.
- MST$\in$ co-$\Sigma_1 \subseteq \Pi_2$ and ISO$\in$ co-$\Pi_2 \subseteq \Sigma_3$.
- No lower bound technique for higher levels.

Open problems : Is the hierarchy infinite ?

Local hierarchy structure



- Collapse of classes.
- But complement classes.
- MST$\in$ co-$\Sigma_1 \subseteq \Pi_2$ and ISO$\in$ co-$\Pi_2 \subseteq \Sigma_3$.
- No lower bound technique for higher levels.

Open problems : Is the hierarchy infinite ?

Perspectives

- Solve the open problems on the specific topics
- Applications
  - Message complexity
  - Fault-tolerance
  - Dynamic setting
- A decomposition theorem ?
- Use in other domains :
  - Graph theory
  - Property testing.