Local Distributed Verification

Alkida Balliu
CNRS and University Paris Diderot
GSSI L’Aquila
(This is a joint work with G. D’Angelo, P. Fraigniaud, and D. Olivetti)

Goal and Motivation
• Classify problems according to their difficulty, i.e., build a complexity theory in the distributed setting.
• Build a hierarchy of complexity classes in the context of the LOCAL model.

LOCAL Model
• In the LOCAL model [1], each node:
  – has a unique identifier;
  – has a local input;
  – has a local view (up to constant distance $t$);
  – provides a local output.

Decision Problems
• Objective: decide whether a global input instance satisfies some specific property.
• Input instance: we consider as an input instance the pair $(G, x)$, where $x$ is a function that assigns to each node $v$ the local input $x(v)$.
• Distributed language: all the input instances that satisfy a specific property.
• Local decision: to decide whether an input instance $(G, x)$ satisfies a property, each node $v$ gathers its local information from its local view and outputs its local decision:
  – "accept" if $(G, x)$ satisfies the desired property;
  – "reject" if $(G, x)$ does not satisfy the property.

Verification Problems
• Objective: verify whether a global input instance satisfies some specific property.
• Certificate: information given by a third party that we don’t trust a priori. To preserve privacy, each certificate is independent from the id assignment.
• Prover: provides a certificate and tries to make nodes accept.
• Disprover: tries to make nodes reject the input instance by providing the first certificate.
• Prover: provides the second certificate and tries to make nodes accept.

Is the Graph Properly Colored?
• If yes, all nodes will locally accept.

Verification Problems
• Objective: verify whether a global input instance satisfies some specific property.

Distributed Complexity Classes
The classes LD and NLD are the bases of a local hierarchy that we define as follows.
• LD: $\Sigma^0_k \subset \Pi^0_k \subset \Delta^0_k$.
• NLD: $\Sigma^0_k \subset \Sigma^0_k$.

$\Sigma^0_k (k \geq 1)$ is the class of all distributed languages $\mathcal{L}$ for which there exists a local algorithm $A$ satisfying that, for every input instance $(G, x)$, $(G, x) \in \mathcal{L} \iff \exists c \text{ s.t. } A \text{ accepts } (G, x)$ with $c$.

LD is the class of all distributed languages $\mathcal{L}$ for which there exists a local algorithm $A$ such that:

$\forall (G, x) \in \mathcal{L} \iff A \text{ accepts } (G, x)$.

A Relation with the Polynomial Hierarchy.
$L \in \text{NP}$ if there is a polynomial time algorithm $A$ such that:

$x \in L \iff \exists c \text{ s.t. } A \text{ accepts } x$ with $c$.

• NLD is closed under lift [4]:
  - if $(G, x) \in \mathcal{L}$ and $(G', x')$ is a lift of $(G, x)$, then $(G', x') \in L$.

• NL is defined similarly but starting with a universal quantifier instead of an existential one.

References