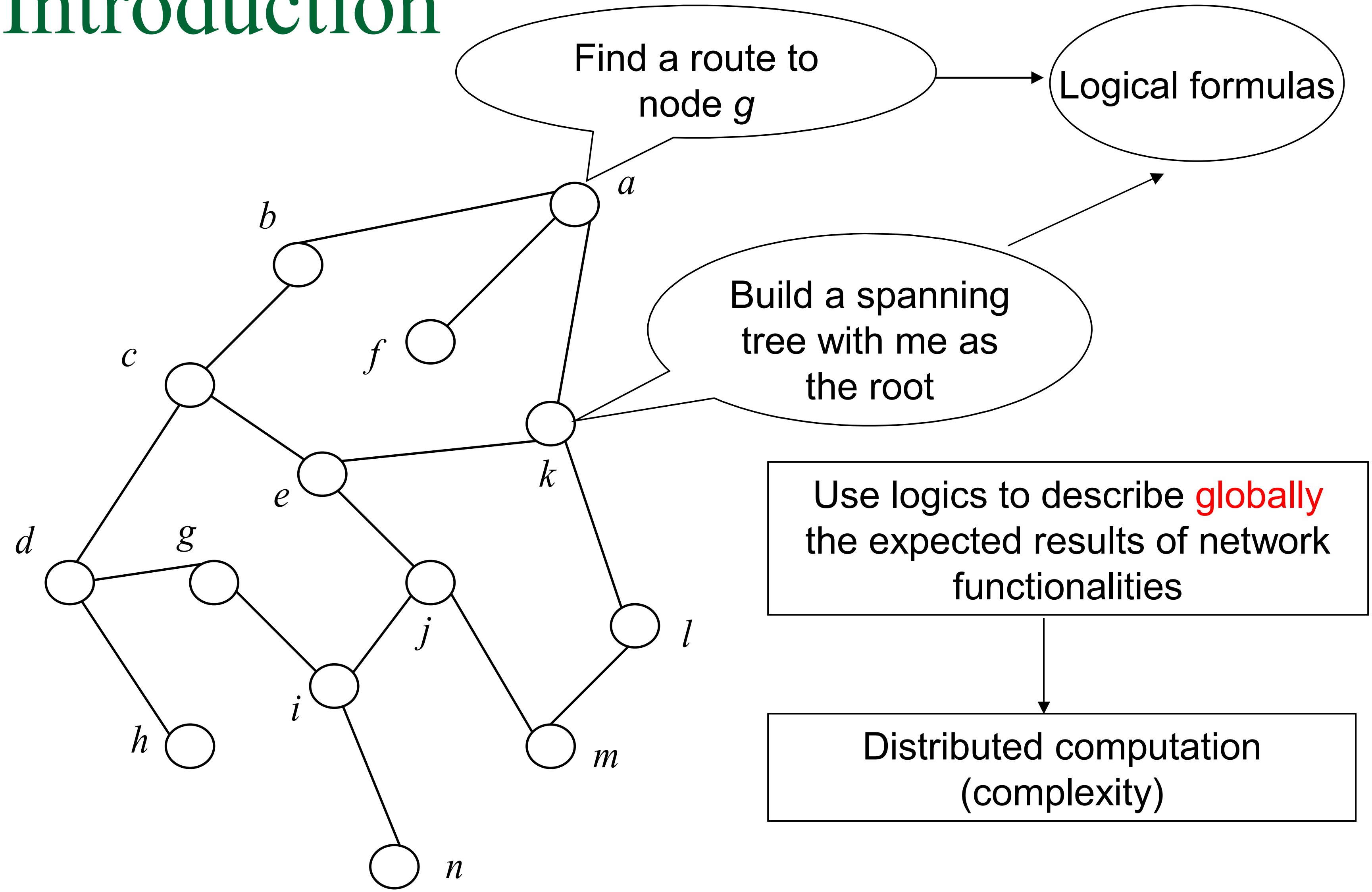
Logical locality entails frugal distributed computation over graphs

Stéphane Grumbach, Zhilin Wu
INRIA-LIAMA

Introduction



Classical logics

- First order logic (FO)
- Fixpoint logics
- Monadic second order logic (MSO)
- …

The starting point: FO

First Order Logic

- Foundation of mathematics and theoretical computer science
  - Used in databases: relational query languages
- Expressive power
  - Locality
    - Local areas of the graphs sufficient to evaluate FO formulas
  - Expressive power limited
    - Connectivity, parity (whether a set is of even size) are non-local properties, thus not expressible in FO

Complexity of FO

- Sequential complexity
  - Linear time over classes of locally tree decomposable graphs (including bounded degree graphs, planar graphs, etc)
- Parallel complexity
  - Evaluated in AC^0, constant time over Boolean circuits with unbounded fan-in
- Distributed complexity?

Local distributed computation

- Message passing model
- Constant distributed time
  - Compute with information only from a bounded neighborhood

Solve global problems

by local distributed computations around each node

Does logical locality imply local distributed computation?

- The answer is "NO"

Simple properties cannot be checked by local computations

"*There are at least two distinct triangles*"

Non-local communication is necessary to check the distinctness of the two triangles

Frugal distributed computations

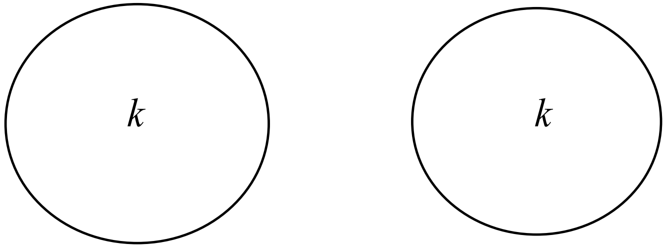
- A bounded number of messages
  of size $O(\log n)$
  sent over each link
- Frugal computations resemble
  local computations
  over bounded degree networks
  - Each node receives only
    a bounded number of messages
  - Though the messages can come from
    remote nodes

Outline

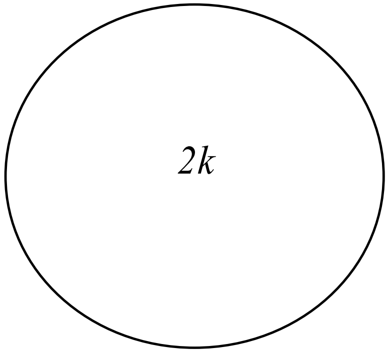
- Locality of first order logic (FO)
- Frugal distributed computation
- Frugality of FO over bounded degree networks
- Frugality of FO over planar networks
- Frugality beyond FO properties

Locality of FO: Intuition

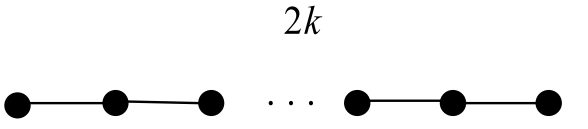
Connectivity : non-local



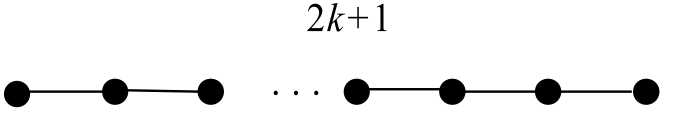
Versus



Parity : non-local



Versus



Neighborhood

- First order logic over graphs
  FO with vocabulary "E(x,y)"
- k-neighborhood of v ($N_k(v)$)
  Vertices with distance from v no more than k
- Let $\bar{v}$ be a list of vertices, then
  $$N_k(\bar{v}) := \bigcup_i N_k(v_i)$$
- "The distance between $\bar{x}$ and z is no more than k"
  Expressed by an FO formula $dist(\bar{x}, z) \leq k$

Local Formula

- Local formula $\varphi(\bar{y})^{(k)}(\bar{x})$

($\bar{x}$ is a list of variables not occurring in $\varphi(\bar{y})$)

The quantifiers of $\varphi(\bar{y})$ are bounded to the k-neighborhood of $\bar{x}$

- Example, if $\varphi(\bar{y}) := \exists z \psi(\bar{y}, z)$, then

$$\left(\varphi(\bar{y})\right)^{(k)}(\bar{x}) := \exists z \left(dist(\bar{x}, z) \leq k \wedge \left(\psi(\bar{y}, z)\right)^{(k)}(\bar{x}) \right)$$

Gaifman theorem

- Each FO formula $\varphi(\overline{u})$ ($\overline{u} = u_1 \cdots u_p$)
  can be written into a Boolean combination
  of sentences of the form
  $$\exists x_1 \cdots \exists x_s \left(\bigwedge_{1 \le i < j \le s} d(x_i, x_j) > 2r \wedge \bigwedge_i \psi^{(r)}(x_i) \right)$$
  and formulae of the form $\psi^{(t)}(\overline{y})$,
  where $\overline{y} = y_1 \cdots y_q$ and $y_i \in \{u_1, \cdots, u_p\}$

Outline

- Locality of first order logic (FO)
- Frugal distributed computation
- Frugality of FO over bounded degree networks
- Frugality of FO over planar networks
- Frugality beyond FO properties

Distributed computation model

- Message passing model
- Asynchronous system
- Network topology modeled by a graph $G=(V,E)$
- Unique identifier for each node
- A breath-first-search (BFS) tree rooted at a distinguished node (requesting node) distributively stored

Frugal distributed algorithms

Only a constant number of messages of size $O(\log n)$ sent over each link during its computation

Frugal computation of FO sentences

- Let ***C*** be a class of graphs, and φ be an FO sentence. φ can be frugally computed over ***C*** if
  $\exists$ a frugal distributed algorithm such that
  $\forall$ *G* $\in$ ***C*** and any requesting node in *G*,
  after the algorithm terminates, the requesting node is in the accepting state iff $G \models \varphi$

Outline

- Locality of first order logic (FO)
- Frugal distributed computation
- Frugality of FO over bounded degree networks
- Frugality of FO over planar networks
- Frugality beyond FO properties

Locality of FO over bounded degree networks

- r-type of v

  Isomorphic type of $(\langle N_r(v)\rangle^G, v)$

- G_1 and G_2 are (r,m)-equivalent

  For every r-type τ,
  either G_1 and G_2 have the same number of vertices with r-type τ,
  or both have at least m vertices with r-type τ.

Locality of FO over bounded degree networks (continued)

- Theorem. Fagin-Stockmeyer-Vardi

  $\forall k,d$, $\exists r=f(k)$, $m=g(k,d)$, s.t.

  For all G_1 and G_2 with degree at most d,

  if G_1 and G_2 are (r,m)-equivalent, then

  G_1 and G_2 satisfy the same

  FO sentences with quantifier rank at most k.

Frugal algorithm for FO over bounded degree networks

- The requesting node ask each node to collect the topology information of its *r*-neighborhood.
- The r-types of the nodes are aggregated through the BFS tree to the requesting node up to the threshold *m* for each r-type.
- Finally the requesting node decides whether $G \models \varphi$ by using the information about the r-types.

Complexity

- The total number of distinct r-types of graphs with degree bounded by d depends only on r and d
- Each r-type is counted up to the threshold m
- Over each link, only $O(1)$ messages are sent during the computation

Outline

- Locality of first order logic (FO)
- Frugal distributed computation
- Frugality of FO over bounded degree networks
- Frugality of FO over planar networks
- Frugality beyond FO properties

Frugality of FO over planar networks

- Idea
  Transform
  the linear time centralized evaluation algorithm
  into
  the frugal distributed evaluation algorithm

Linear time centralized computation of FO over planar graphs

- Sufficient to consider FO sentences of the form

$$\exists x_1 \cdots \exists x_s \left(\bigwedge_{1 \le i < j \le s} d(x_i, x_j) > 2r \wedge \bigwedge_i \psi^{(r)}(x_i) \right)$$

- Centralized algorithm
  - Select a distinguished vertex v_0, decompose $G=(V,E)$ into subgraphs $G[i,i+2r]:=\{v \mid i \le dist(v_0,v) \le i+2r\}$.
  - Evaluate $\psi^{(r)}(x)$ over $G[i,i+2r]$
    - Each $G[i,i+2r]$ is a graph with bounded tree width
    - Construct from $\psi^{(r)}(x)$ an automaton ***A*** and run it over a tree decomposition obtained from $G[i,i+2r]$
  - Let *P* be the set of vertices v such that $\exists i$, $N_r(v) \subseteq G[i,i+2r]$ and $G[i,i+2r] \models \psi^{(r)}(v)$.

    Decide whether $\psi' := \exists x_1 \cdots \exists x_s \left(\bigwedge_{1 \le i < j \le s} d(x_i, x_j) > 2r \wedge \bigwedge_i P(x_i) \right)$ holds over G or not

Distributed computation of FO over planar networks

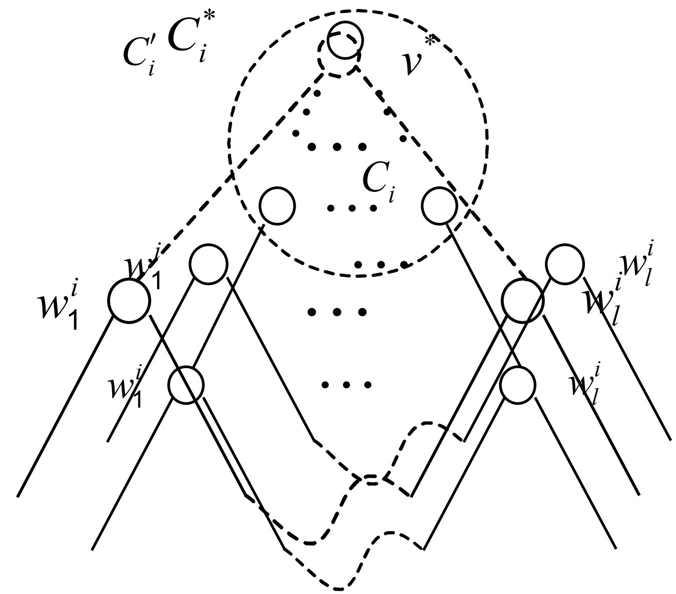
- Transform the centralized algorithm
  into a distributed one
  while ensuring frugality
- The most difficult part is
  to evaluate distributively $\psi^{(r)}(x)$ over $G[i,i+2r]$

Distributively Evaluate $\psi^{(r)}(x)$ over $G[i,i+2r]$

- $\psi^{(r)}(x)$ is a local formula, so $\psi^{(r)}(x)$ can be evaluated over each connected component of $G[i,i+2r]$
- Still use automata-theoretic technique
  - Construct an automaton ***A*** from $\psi^{(r)}(x)$
  - Construct and store distributively a tree decomposition for each connected component of $G[i,i+2r]$
  - Evaluate $\psi^{(r)}(x)$ by distributively running ***A*** over the tree decomposition.

Construct a tree decomposition for C_i

- Let C_i be a connected component of $G[i,i+2r]$ and $w_1^i, \cdots, w_l^i$ be the nodes of C_i with distance i from the requesting node.
  - Let C_i' be the graph obtained from C_i by including all ancestors of $w_1^i, \cdots, w_l^i$ in the BFS tree.
  - Let C_i^* be the graph obtained from C_i by contracting all the ancestors of $w_1^i, \cdots, w_l^i$ into one vertex v^*



C_i^* is a planar graph with bounded diameter

Construct a tree decomposition for C_i (continued)

- A tree decomposition of
  planar networks with bounded diameter
  can be computed frugally
  [Grumbach & Wu, manuscript]
- Construct a tree decomposition for C_i

  Construct distributively a tree decomposition for C_i^*,
  while doing special treatments for the virtual vertex,
  and get a tree decomposition for C_i.

Outline

- Locality of first order logic (FO)
- Frugal distributed computation
- Frugality of FO over bounded degree networks
- Frugality of FO over planar networks
- Frugality beyond FO properties

Frugality beyond FO properties

- FO unary queries: FO formulas $\varphi(x)$ with one free variable x
  - Example: "Vertices of degree d"
    $\xi_d(x) := \exists y_1 \ldots \exists y_d \left(\wedge_{i \neq j} y_i \neq y_j \wedge \wedge_i E(x, y_i) \wedge \forall z \left(E(x,z) \rightarrow \vee_i z = y_i\right)\right)$
  - Frugally computed over
    bounded degree and planar networks
- FO(#): Extension of FO with unary counting
  - Example: "the same number of vertices of degree 2 and 3"
    $$\#x.\xi_2(x) = \#x.\xi_3(x)$$
  - Frugally computed over
    bounded degree and planar networks
- Fixpoint, MSO, …

Conclusion

- Logical locality implies low distributed complexity
- BFS tree assumption
  - Can be replaced by
    a pre-computed tree 2-spanner,
    independently of the choice of the requesting node
  - Best distributed message complexity:
    $O(\Delta)$ messages sent over each link
  - Open question:
    Does a BFS tree can be frugally computed?
- Open question: Does NP-complete problems like Hamiltonicity, can be frugally computed?

Thanks!