Efficient FO-model checking on Barabasi-Albert Graphs

Jan Dreier, Philipp Kuinke, Peter Rossmanith

April 5, 2018

Theoretical Computer Science
RWTH Aachen University, Germany

Motivation

Real-World Networks

Sparsity

Logic

Random Graphs

Efficient Algorithms

FO-logic and graphs

FO-logic (first order logic) is a powerful language to express problems in.

Recursive definition:

- $\varphi = \forall x\, \psi(x)$
- $\varphi = \exists x\, \psi(x)$
- $\varphi = (u = v)$
- $\varphi = adj(u, v)$
- $\varphi = \varphi_1 \vee \varphi_2$
- $\varphi = \varphi_1 \wedge \varphi_2$
- $\varphi = \neg\psi$

$G \models \varphi$: Is G a model for φ?

Example: $\exists x \exists y \exists z (adj(x, y) \wedge adj(x, z) \wedge adj(y, z))$

Efficient FO model checking

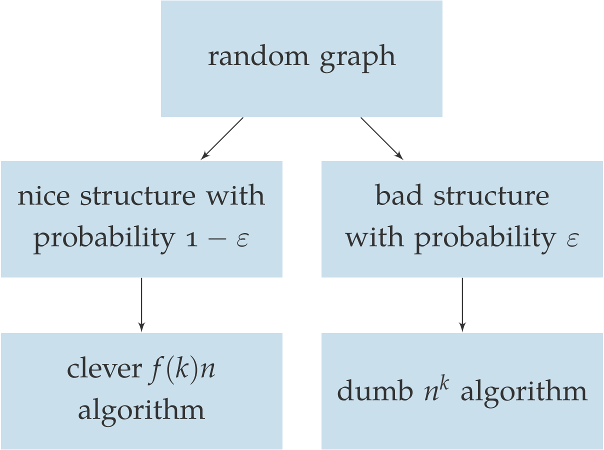
- If $\mathcal{G}$ has bounded treewidth, then we can decide $G \models \varphi$ in linear time if $G \in \mathcal{G}$ and φ is an MSO-formula. [Courcelle 1990]
- If $\mathcal{G}$ has bounded expansion, then we can decide $G \models \varphi$ in linear time if $G \in \mathcal{G}$ and φ is an FO-formula. [Dvořák, Kráľ, Thomas 2010]
- If $\mathcal{G}$ is nowhere dense, then we can decide $G \models \varphi$ in time $n^{1+\epsilon}$ if $G \in \mathcal{G}$ and φ is an FO-formula. [Kreutzer, Grohe, Siebertz 2011]

Algorithms on Random Graphs

When is an algorithm fast on random inputs?

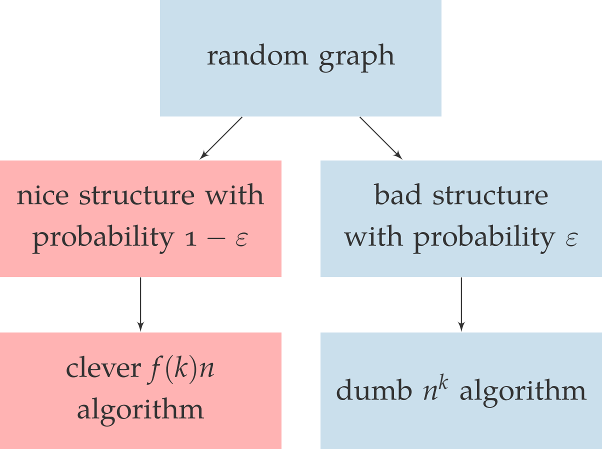
- worst case running time?
- fast on $1 - \varepsilon$ fraction of inputs for small ε?
- fast average running time
  $\sum_{G \in \mathcal{G}}$ running time on input $G \cdot$ probability of G.

Algorithms on Random Graphs



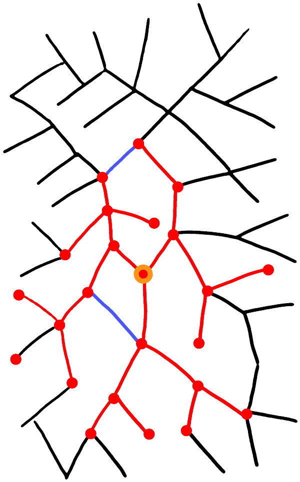
Expected running time: $(1 - \varepsilon)f(k)n + \varepsilon n^k$

Algorithms on Random Graphs



Expected running time: $(1 - \varepsilon)f(k)n + \varepsilon n^k$

Removing $\log(n)^{O(r)}$ nodes yields with high probability: Every r-neighborhood is only few edges away from a tree.



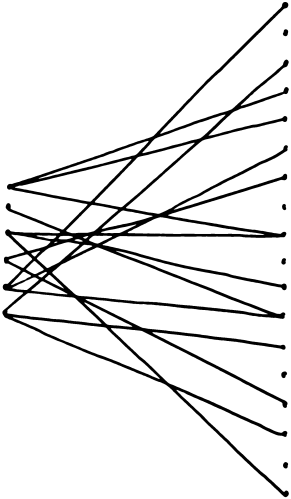
Gaifman: If we can do model-checking on r-neighborhoods we can do it for the complete graph.

Structure of Preferential Attachement Graphs

This is not enough.

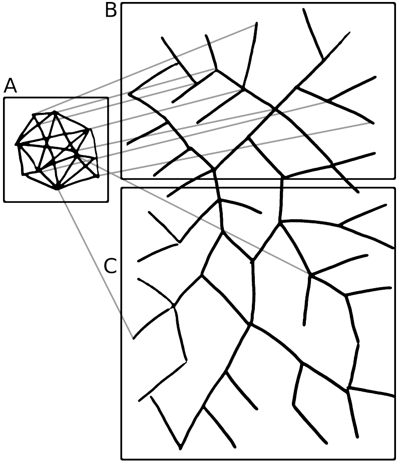
Model-checking on bipartite graphs with $\log(n)$ vertices on the left as hard as the general case.

Problem: Vertices on the right can have many neighbors on the left.



Structure of Preferential Attachement Graphs

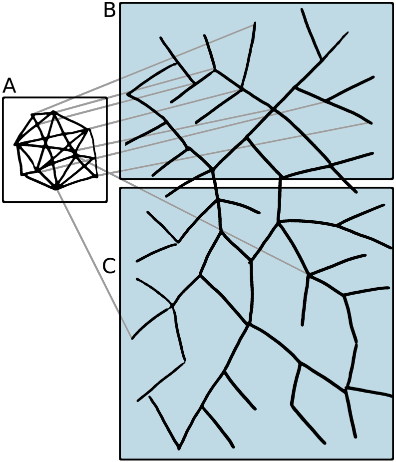
We decompose the vertices into sets A, B, C such that:



Structure of Preferential Attachement Graphs

We decompose the vertices into sets A, B, C such that:

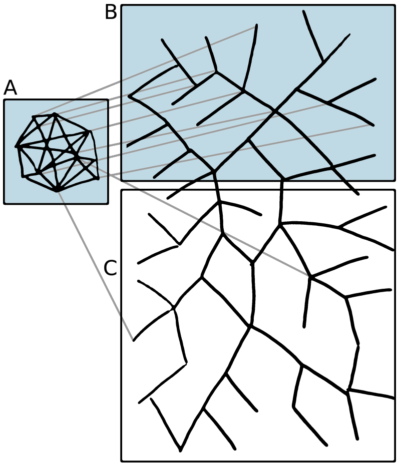
- $B \cup C$ is locally treelike



Structure of Preferential Attachement Graphs

We decompose the vertices into sets A, B, C such that:

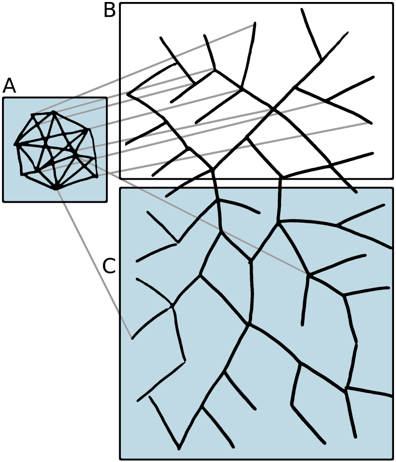
- $B \cup C$ is locally treelike
- A, B have size $\log(n)^{O(r)}$



Structure of Preferential Attachement Graphs

We decompose the vertices into sets A, B, C such that:

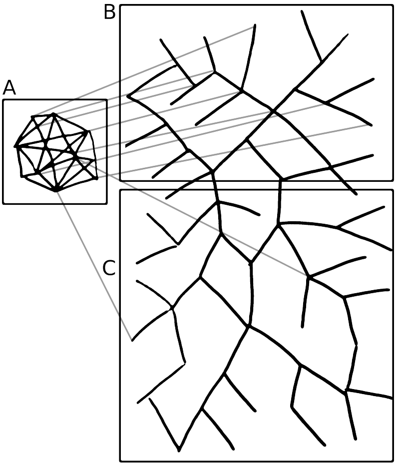
- $B \cup C$ is locally treelike
- A, B have size $\log(n)^{O(r)}$
- every r-neighborhood in C has only few edges to A



Structure of Preferential Attachement Graphs

We decompose the vertices into sets A, B, C such that:

- $B \cup C$ is locally treelike
- A, B have size $\log(n)^{O(r)}$
- every r-neighborhood in C has only few edges to A



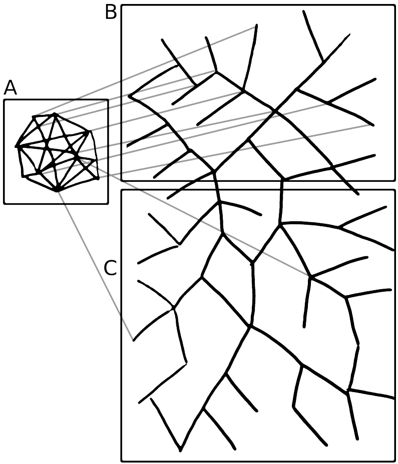
Structure of Preferential Attachement Graphs

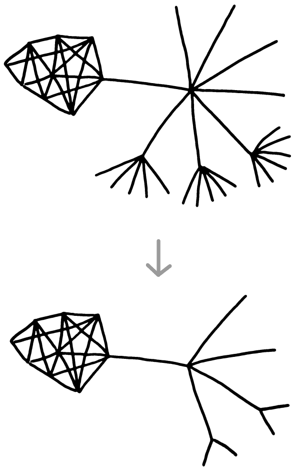
We decompose the vertices into sets A, B, C such that:

- $B \cup C$ is locally treelike
- A, B have size $\log(n)^{O(r)}$
- every r-neighborhood in C has only few edges to A

Then:

- use Gaifman
- Kernelize neighborhood to size $f(|\varphi|)\,\mathrm{polylog}(n)$
- Solve in time $\left(f(|\varphi|)\,\mathrm{polylog}(n)\right)^{|\varphi|} = O(n^{\varepsilon})$





Kernelization:

- Consider r-neighborhoods
- Prune redundant isomorphic subtrees in C
- Size: $f(|\varphi|)\,\mathrm{polylog}(n)$

Model-Checking Algorithm

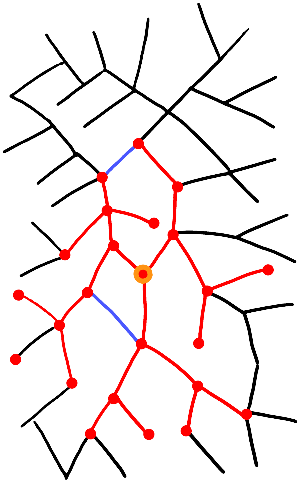
Input: Preferential attachement graph G and formula φ

- Check if G has A-B-C-structure.
- If not use $n^{|\varphi|}$-algorithm and exit. Otherwise proceed.
- Use Gaifman to restrict model-checking to neighborhoods.
- Build $f(|\varphi|)\,\mathrm{polylog}(n)$-kernel for neighborhood.
- Solve in time $\left(f(|\varphi|)\,\mathrm{polylog}(n)\right)^{|\varphi|}$ for each neighborhood.

Probability of not having A-B-C structure is less likely than $\frac{1}{n^{|\varphi|}}$.
Therefore, expected running time FPT.

Proving Structure

Removing $\log(n)^{O(r)}$ nodes yields with high probability: Every r-neighborhood is only few edges away from a tree.

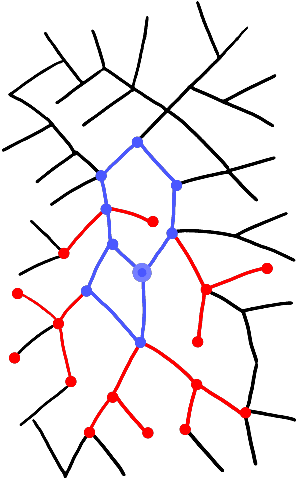


Proving Structure

Removing $\log(n)^{O(r)}$ nodes yields with high probability: Every r-neighborhood is only few edges away from a tree.

r-neighborhood with many edges implies small subgraph with many edges.

We show that small subgraphs with many edges are improbable.



Proving Structure

Probability that vertices $v_1, v_2, \ldots, v_k$ form dense subgraph is

$$s(v_1, v_2, \ldots, v_k) = P(adj(v_1, v_2), adj(v_5, v_3), \ldots)$$

One can show

$$P(adj(a, b)) \leq \frac{1}{\sqrt{ab}}$$

Use union-bound as upper bound

$$\sum_{v_1=l}^{n} \sum_{v_2=l}^{n} \cdots \sum_{v_k=l}^{n} s(v_1, v_2, \ldots, v_k)$$

This sum is small for $l = \log(n)^{O(1)}$.

Summary

We analyze fine structure of preferential attachment graphs to construct an efficient model-checking algorithm.

Possible future work: Generalize to other random graphs.