

MOTIF COUNTING IN PREFERENTIAL ATTACHMENT GRAPHS

Jan Dreier, Peter Rossmanith

December 11, 2019

Theoretical Computer Science
RWTH Aachen University, Germany

FSTTCS 2019

Subgraph Counting

How often does H with k edges occur in G of size n ?

Subgraph Counting

How often does H with k edges occur in G of size n ?

- important in network analysis and biology

Subgraph Counting

How often does H with k edges occur in G of size n ?

- important in network analysis and biology
- many tools that work on real data for small k

How often does H with k edges occur in G of size n ?

- important in network analysis and biology
- many tools that work on real data for small k
- cannot be solved in time $f(k)n^{o(k)}$ under #ETH

Subgraph Counting

How often does H with k edges occur in G of size n ?

- important in network analysis and biology
- many tools that work on real data for small k
- cannot be solved in time $f(k)n^{o(k)}$ under #ETH
- *Curticapean, Dell, Marx*: in time $k^{O(k)}n^{0.174k+o(k)}$

Average Case

Analyze runtime on *typical* instances.

Average Case

Analyze runtime on *typical* instances.

theoretical computer science:

efficient algorithms

network science:

models for the real world

Average Case

Analyze runtime on *typical* instances.

theoretical computer science:

efficient algorithms

network science:

models for the real world



efficient algorithms for the real world?

Some properties of complex networks:

- **Degree distribution**

Most people have little but a few have a lot.

- **Small world**

Everyone is close to everyone.

- **Clustering**

If we have a common friend we are likely friends as well.

Some properties of complex networks:

- **Degree distribution** ✓

Most people have little but a few have a lot.

- **Small world** ✓

Everyone is close to everyone.

- **Clustering** ✗

If we have a common friend we are likely friends as well.

Preferential Attachment

- G_m^n graph with vertices v_1, \dots, v_n and m edges per vertex

- G_2^1 :



Preferential Attachment

○ G_m^n graph with vertices v_1, \dots, v_n and m edges per vertex

○ G_2^1 :



Preferential Attachment

- G_m^n graph with vertices v_1, \dots, v_n and m edges per vertex

- G_2^1 :



Preferential Attachment

- G_m^n graph with vertices v_1, \dots, v_n and m edges per vertex

- G_2^1 :



Preferential Attachment

- G_m^n graph with vertices v_1, \dots, v_n and m edges per vertex

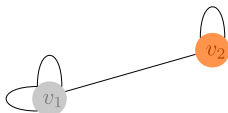
- G_2^1 :



Preferential Attachment

- G_m^n graph with vertices v_1, \dots, v_n and m edges per vertex

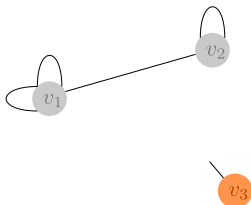
- G_2^2 :



Preferential Attachment

- G_m^n graph with vertices v_1, \dots, v_n and m edges per vertex

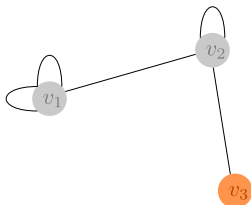
- G_2^2 :



Preferential Attachment

- G_m^n graph with vertices v_1, \dots, v_n and m edges per vertex

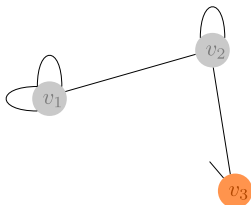
- G_2^2 :



Preferential Attachment

- G_m^n graph with vertices v_1, \dots, v_n and m edges per vertex

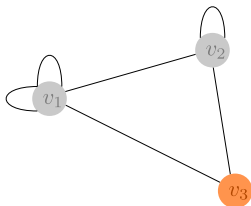
- G_2^2 :



Preferential Attachment

○ G_m^n graph with vertices v_1, \dots, v_n and m edges per vertex

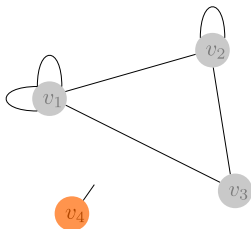
○ G_2^3 :



Preferential Attachment

○ G_m^n graph with vertices v_1, \dots, v_n and m edges per vertex

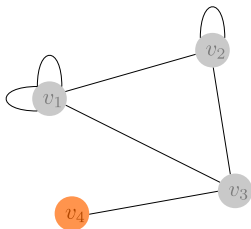
○ G_2^3 :



Preferential Attachment

○ G_m^n graph with vertices v_1, \dots, v_n and m edges per vertex

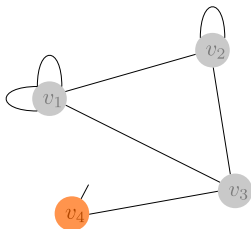
○ G_2^3 :



Preferential Attachment

○ G_m^n graph with vertices v_1, \dots, v_n and m edges per vertex

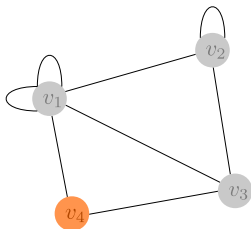
○ G_2^3 :



Preferential Attachment

○ G_m^n graph with vertices v_1, \dots, v_n and m edges per vertex

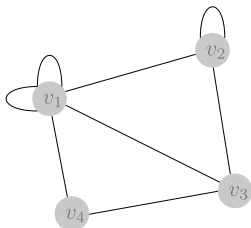
○ G_2^4 :



Preferential Attachment

○ G_m^n graph with vertices v_1, \dots, v_n and m edges per vertex

○ G_2^4 :



Theorem

One can compute how often H occurs in G_m^n in expected FPT time $f(|H|)m^{O(|H|^6)} \log(n)^{O(|H|^{12})}n$.

In particular, for fixed $m \in \mathbf{N}$ and H the run time is $\log(n)^{O(1)}n$.

Simple Algorithm



Bound on number of certain subgraphs



Bound on degrees of individual vertices

Counting Disconnected Subgraphs

It is sufficient to count connected subgraphs.



Counting Disconnected Subgraphs

It is sufficient to count connected subgraphs.

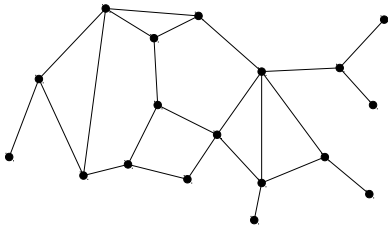
$$\# \square \triangleright = \# \square \times \# \triangleright$$

Counting Disconnected Subgraphs

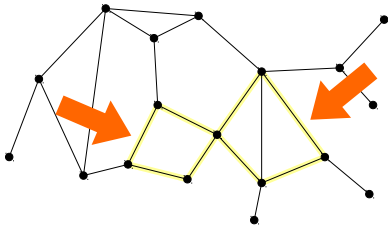
It is sufficient to count connected subgraphs.

$$\begin{aligned} & \# \square \triangleright \\ &= \# \square \times \# \triangleright \\ & - \# \square \triangleright \\ & - \# \square \triangleright \\ & - \# \square \end{aligned}$$

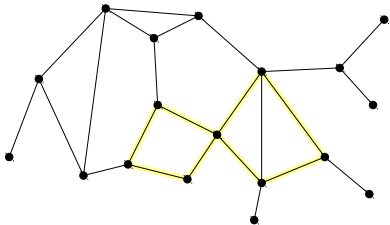
Example: Counting the number of squares



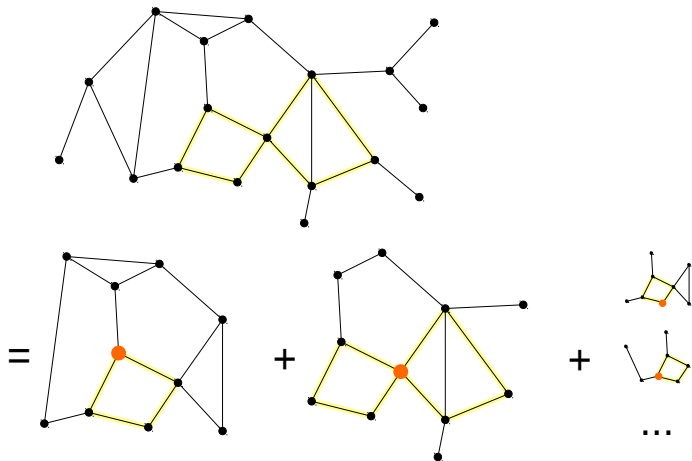
Example: Counting the number of squares



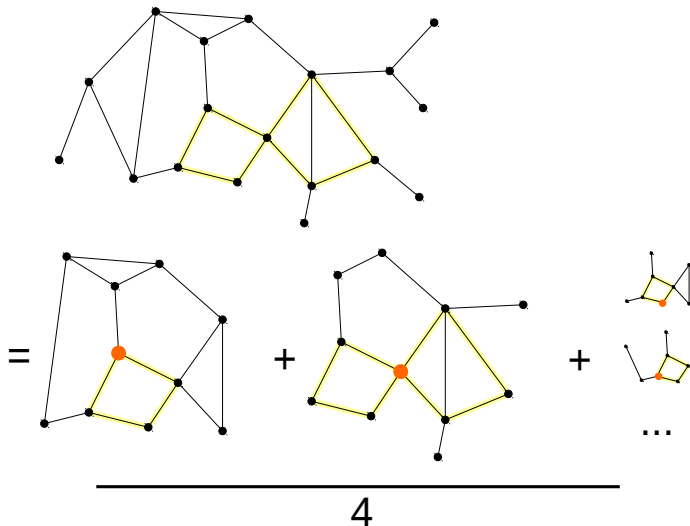
Example: Counting the number of squares



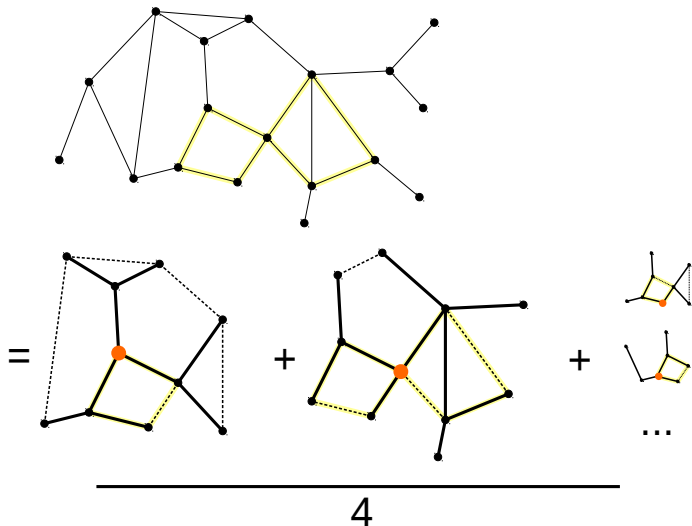
Example: Counting the number of squares



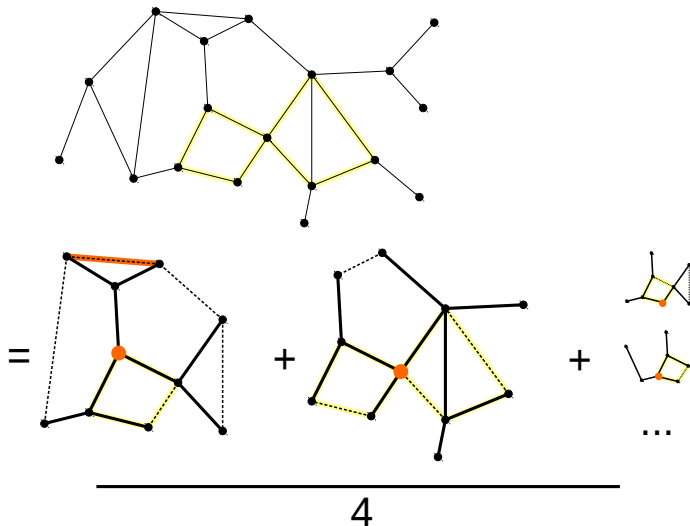
Example: Counting the number of squares



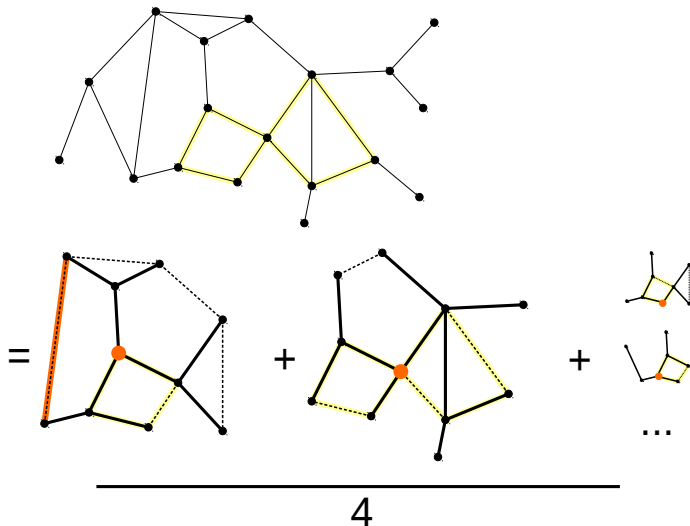
Example: Counting the number of squares



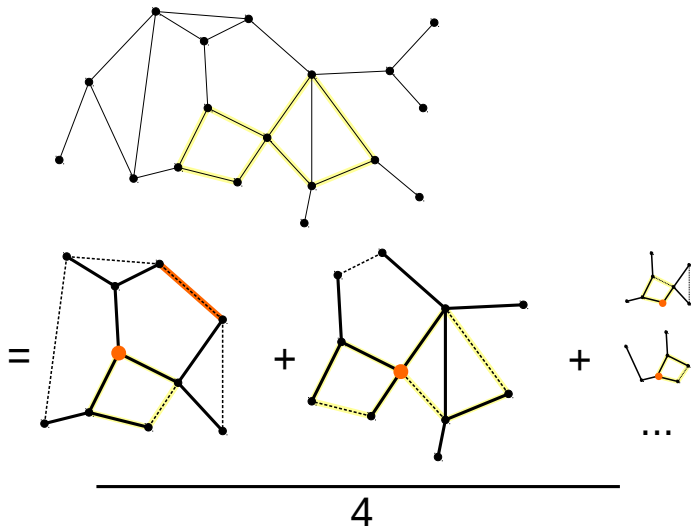
Example: Counting the number of squares



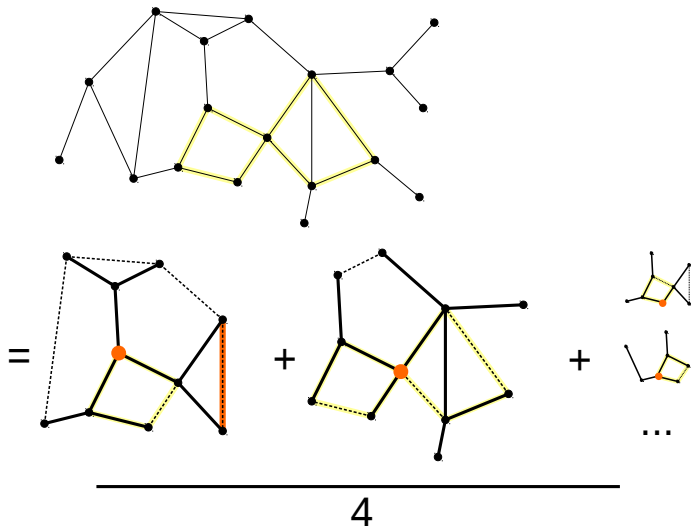
Example: Counting the number of squares



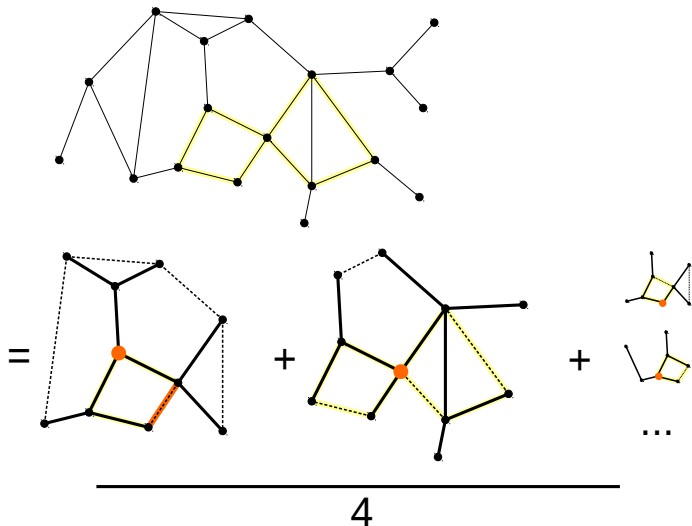
Example: Counting the number of squares



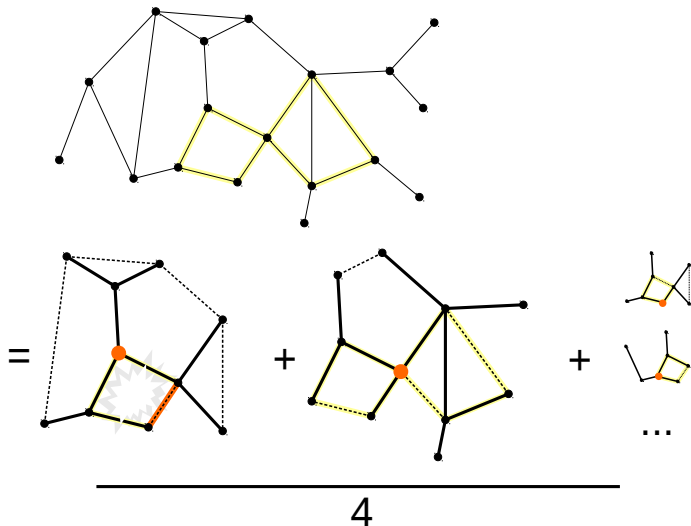
Example: Counting the number of squares



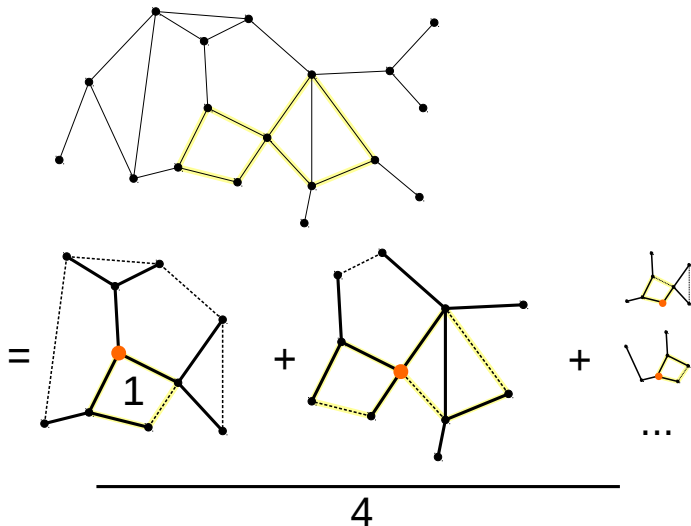
Example: Counting the number of squares



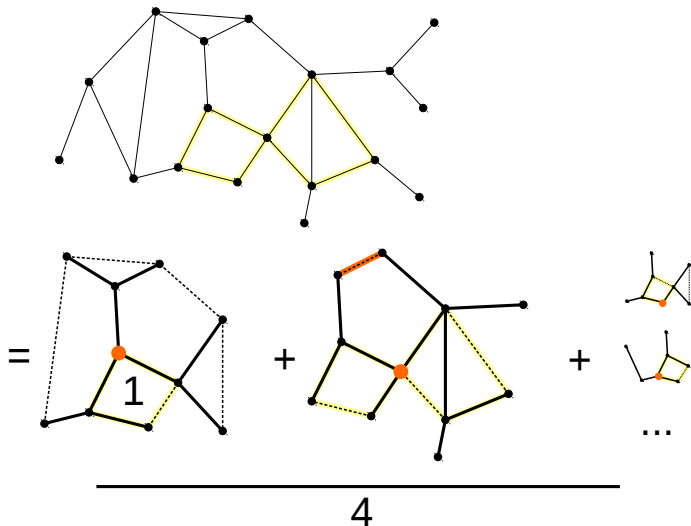
Example: Counting the number of squares



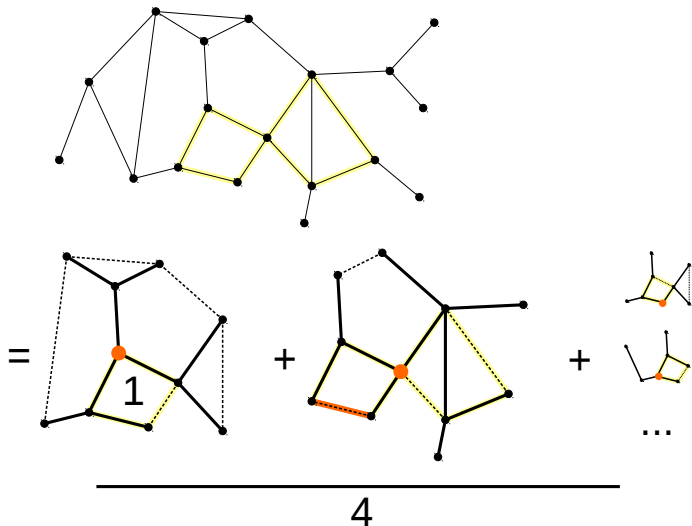
Example: Counting the number of squares



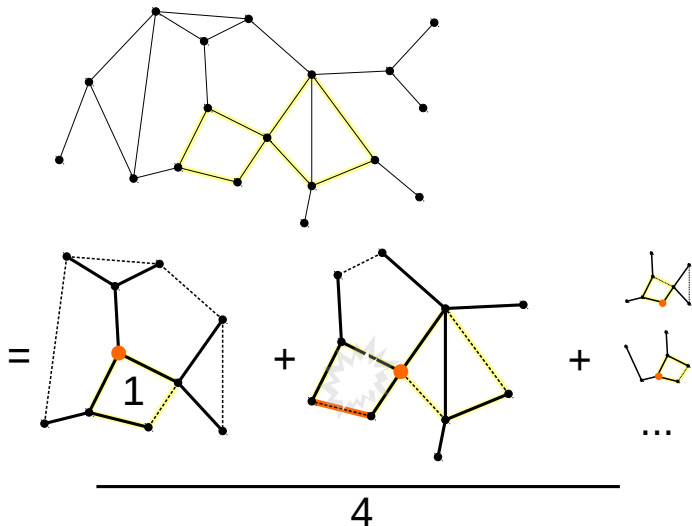
Example: Counting the number of squares



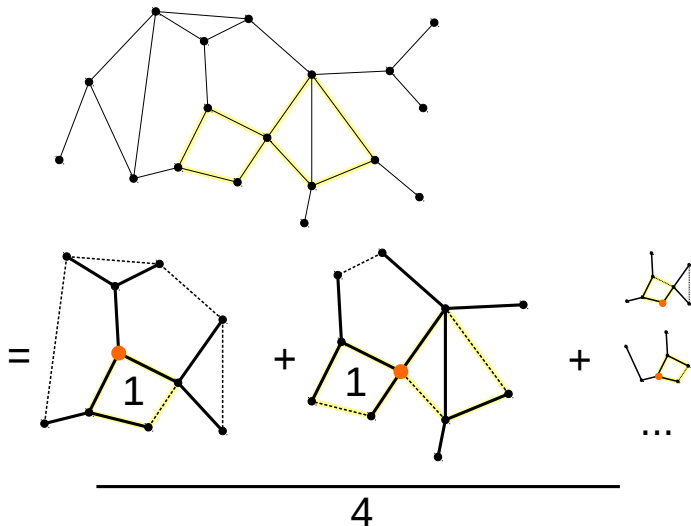
Example: Counting the number of squares



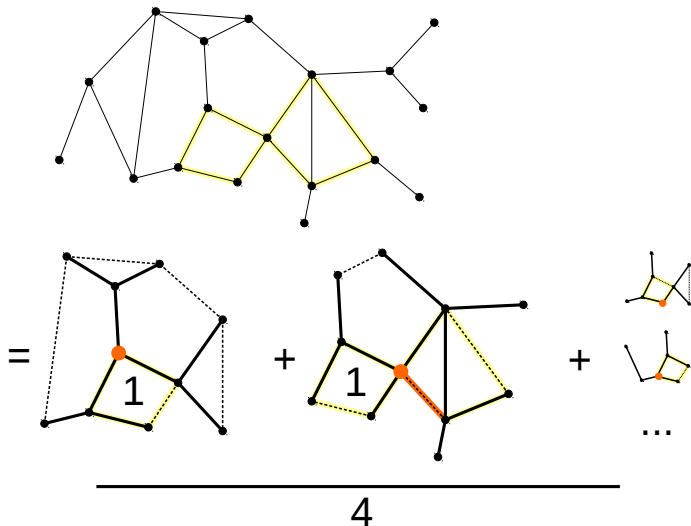
Example: Counting the number of squares



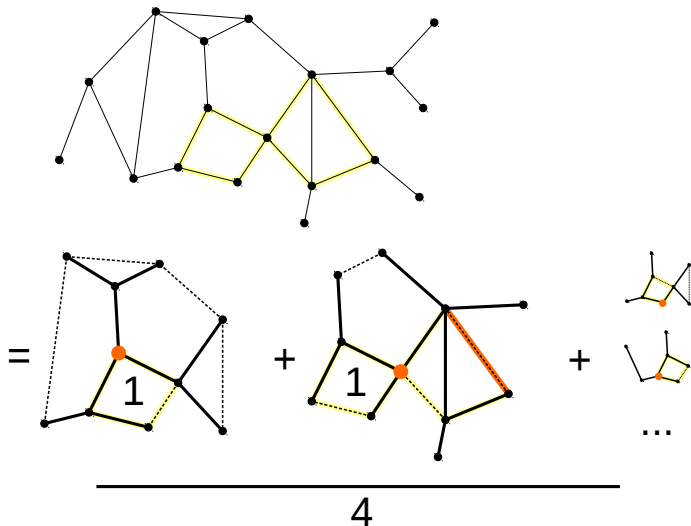
Example: Counting the number of squares



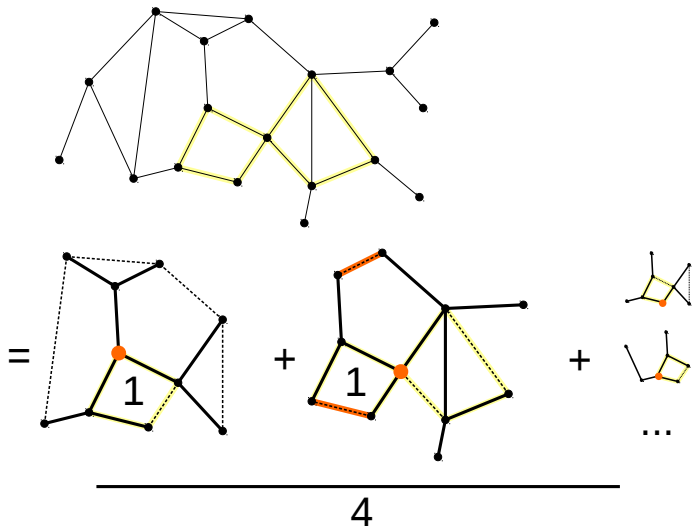
Example: Counting the number of squares



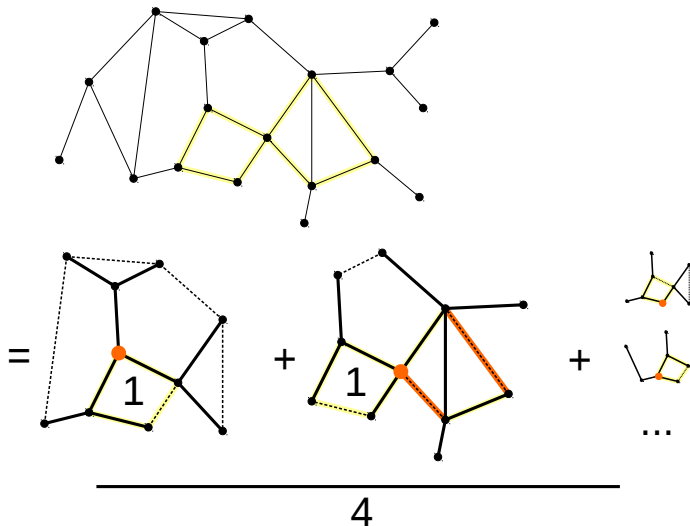
Example: Counting the number of squares



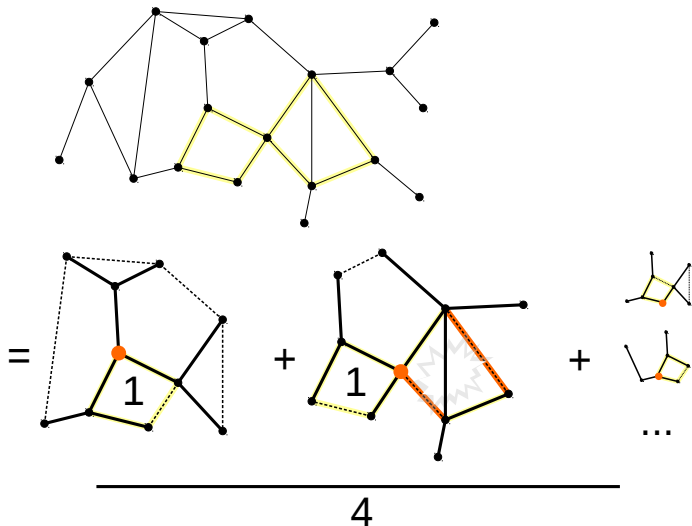
Example: Counting the number of squares



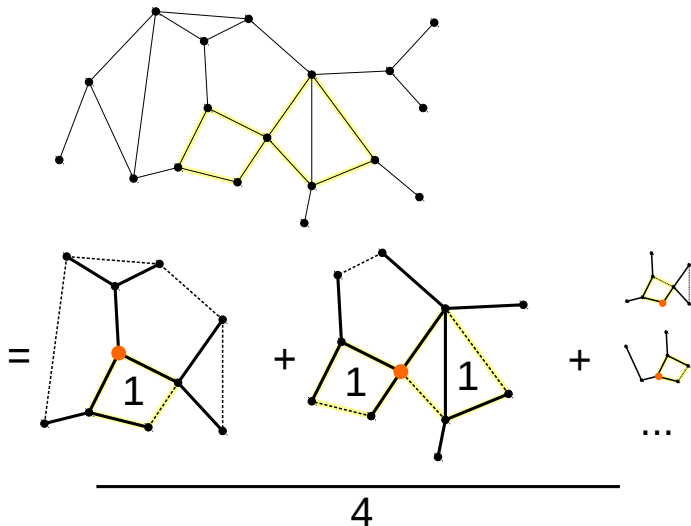
Example: Counting the number of squares



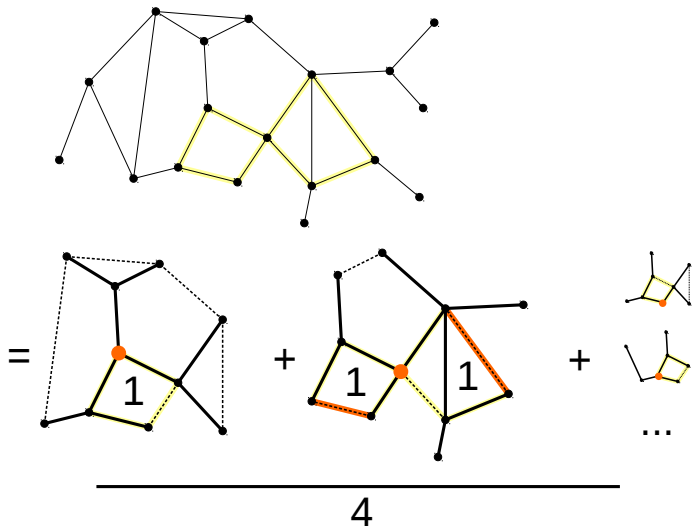
Example: Counting the number of squares



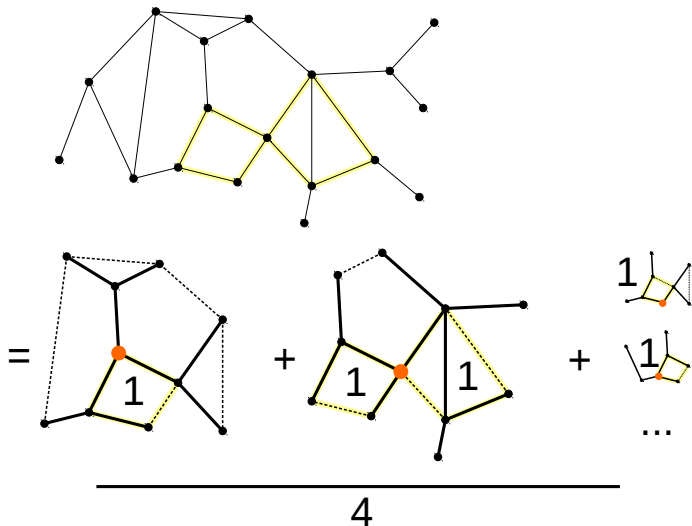
Example: Counting the number of squares



Example: Counting the number of squares

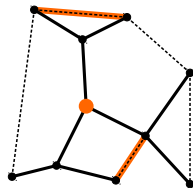


Example: Counting the number of squares



Bounding the Run Time

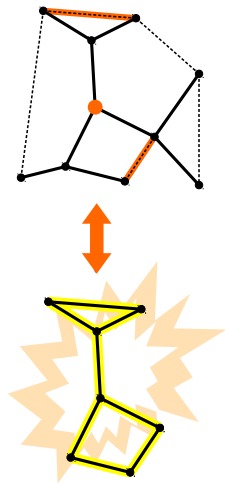
We need to bound the number of subroutine calls per neighborhood.



Bounding the Run Time

We need to bound the number of subroutine calls per neighborhood.

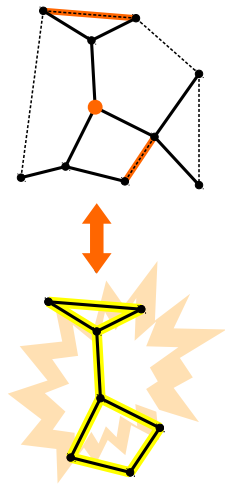
- For every subroutine call there is exactly one small leaf-free subgraph.



Bounding the Run Time

We need to bound the number of subroutine calls per neighborhood.

- For every subroutine call there is exactly one small leaf-free subgraph.
- Number of subroutine calls bounded by number of small leaf-free subgraphs.

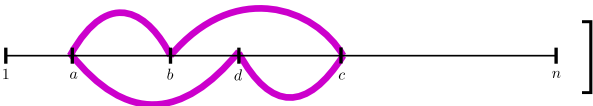


Bounding Number of Leaf-Free Subgraphs

expectation of $\# \square$ in G_m^n

Bounding Number of Leaf-Free Subgraphs

expectation of $\# \square$ in G_m^n

$$= \Pr \left[\begin{array}{c} \text{---} | \text{---} | \text{---} | \text{---} | \text{---} \\ 1 \quad a \quad b \quad d \quad c \quad n \end{array} \right]$$


Bounding Number of Leaf-Free Subgraphs

expectation of $\# \square$ in G_m^n

$$= \Pr \left[\begin{array}{c} \text{Diagram 1} \\ \text{Diagram 2} \end{array} \right] + \Pr \left[\begin{array}{c} \text{Diagram 3} \\ \text{Diagram 4} \end{array} \right]$$

Bounding Number of Leaf-Free Subgraphs

expectation of $\# \square$ in G_m^n

$$\begin{aligned} &= \Pr \left[\begin{array}{c} \text{Diagram 1: Path } 1 \text{ to } n \text{ with vertices } a, b, d, c \text{ and edges } (a,b), (b,d), (d,c). \end{array} \right] \\ &+ \Pr \left[\begin{array}{c} \text{Diagram 2: Path } 1 \text{ to } n \text{ with vertices } d, b, a, c \text{ and edges } (d,b), (b,a), (a,c). \end{array} \right] \\ &+ \Pr \left[\begin{array}{c} \text{Diagram 3: Path } 1 \text{ to } n \text{ with vertices } a, d, b, c \text{ and edges } (a,d), (d,b), (b,c). \end{array} \right] \end{aligned}$$

Bounding Number of Leaf-Free Subgraphs

expectation of $\# \square$ in G_m^n

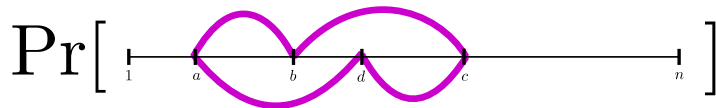
$$\begin{aligned} &= \Pr \left[\begin{array}{c} \text{Diagram 1: A horizontal line with vertices } 1, a, b, d, c, n. \text{ Edges connect } (1,a), (a,b), (b,d), (d,c), (c,n). \text{ Two magenta loops are drawn: one between } a \text{ and } b, \text{ and another between } d \text{ and } c. \end{array} \right] \\ &+ \Pr \left[\begin{array}{c} \text{Diagram 2: A horizontal line with vertices } 1, d, b, a, c, n. \text{ Edges connect } (1,d), (d,b), (b,a), (a,c), (c,n). \text{ Two magenta loops are drawn: one between } d \text{ and } b, \text{ and another between } a \text{ and } c. \end{array} \right] \\ &+ \Pr \left[\begin{array}{c} \text{Diagram 3: A horizontal line with vertices } 1, a, d, b, c, n. \text{ Edges connect } (1,a), (a,d), (d,b), (b,c), (c,n). \text{ Two magenta loops are drawn: one between } a \text{ and } d, \text{ and another between } b \text{ and } c. \end{array} \right] \\ &+ \Pr \left[\begin{array}{c} \text{Diagram 4: A horizontal line with vertices } 1, a, d, c, b, n. \text{ Edges connect } (1,a), (a,d), (d,c), (c,b), (b,n). \text{ Two magenta loops are drawn: one between } a \text{ and } d, \text{ and another between } c \text{ and } b. \end{array} \right] \end{aligned}$$

Bounding Number of Leaf-Free Subgraphs

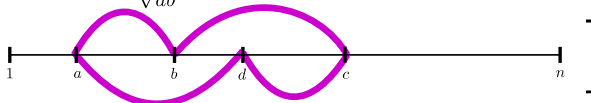
expectation of $\# \square$ in G_m^n

$$\begin{aligned} &= \Pr \left[\begin{array}{c} \text{Diagram 1: } 1 \text{---} a \text{---} b \text{---} d \text{---} c \text{---} n \\ \text{with arcs } (a,b), (b,d), (d,c) \end{array} \right] \\ &+ \Pr \left[\begin{array}{c} \text{Diagram 2: } 1 \text{---} d \text{---} b \text{---} a \text{---} c \text{---} n \\ \text{with arcs } (d,b), (b,a), (a,c) \end{array} \right] \\ &+ \Pr \left[\begin{array}{c} \text{Diagram 3: } 1 \text{---} a \text{---} d \text{---} b \text{---} c \text{---} n \\ \text{with arcs } (a,d), (d,b), (b,c) \end{array} \right] \\ &+ \Pr \left[\begin{array}{c} \text{Diagram 4: } 1 \text{---} a \text{---} d \text{---} c \text{---} b \text{---} n \\ \text{with arcs } (a,d), (d,c), (c,b) \end{array} \right] \\ &+ \dots \end{aligned}$$

Bounding Number of Leaf-Free Subgraphs



Bounding Number of Leaf-Free Subgraphs

$$\Pr\left[\begin{array}{c} \text{Pr}[ab] = \frac{1}{\sqrt{ab}} \\ \text{---} \end{array} \right]$$


The diagram shows a horizontal line representing a path graph with nodes labeled 1, a, b, d, c, and n. A thick magenta cycle is drawn between nodes a, b, d, and c, consisting of two arcs: one above the line and one below. The expression $\Pr[ab] = \frac{1}{\sqrt{ab}}$ is written above the line between nodes a and b.

Bounding Number of Leaf-Free Subgraphs

$$\Pr \left[\begin{array}{c} \text{---} | \text{---} | \text{---} | \text{---} | \text{---} \\ 1 \quad a \quad b \quad d \quad c \quad n \\ \text{---} \end{array} \right]$$

$\Pr[ab] = \frac{1}{\sqrt{ab}}$ $\Pr[bc] = \frac{1}{\sqrt{bc}}$

Bounding Number of Leaf-Free Subgraphs

$$\Pr \left[\begin{array}{c} \text{---} 1 \text{---} a \text{---} b \text{---} d \text{---} c \text{---} n \text{---} \\ \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \\ \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \\ \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \end{array} \right]$$

$\Pr[ab] = \frac{1}{\sqrt{ab}}$ $\Pr[bc] = \frac{1}{\sqrt{bc}}$

$\Pr[da] = \frac{1}{\sqrt{da}}$

Bounding Number of Leaf-Free Subgraphs

$$\Pr \left[\begin{array}{c} \text{---} 1 \text{---} a \text{---} b \text{---} d \text{---} c \text{---} n \text{---} \\ \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \\ \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \end{array} \right]$$

$\Pr[ab] = \frac{1}{\sqrt{ab}}$ $\Pr[bc] = \frac{1}{\sqrt{bc}}$

$\Pr[da] = \frac{1}{\sqrt{da}}$ $\Pr[cd] = \frac{1}{\sqrt{cd}}$

The diagram illustrates a path on a number line starting at 1 and ending at n. Points a, b, d, and c are marked on the line in increasing order. Purple arcs connect a to b, b to d, d to c, and c to a, forming a closed loop. The probability of each arc is given by the formula $\frac{1}{\sqrt{\text{product of endpoints}}}$.

Bounding Number of Leaf-Free Subgraphs

$$\Pr \left[\begin{array}{c} \text{---} 1 \text{---} a \text{---} b \text{---} d \text{---} c \text{---} n \text{---} \\ \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} \end{array} \right]$$

$\Pr[ab] = \frac{1}{\sqrt{ab}}$ $\Pr[bc] = \frac{1}{\sqrt{bc}}$

$\Pr[da] = \frac{1}{\sqrt{da}}$ $\Pr[cd] = \frac{1}{\sqrt{cd}}$

$$= \Pr[ab] \cdot \Pr[bc] \cdot \Pr[cd] \cdot \Pr[da]$$
$$= \frac{1}{abcd}$$

Bounding Number of Leaf-Free Subgraphs

$$\Pr \left[\begin{array}{c} \text{---} 1 \text{---} a \text{---} b \text{---} d \text{---} c \text{---} n \text{---} \\ \text{---} \text{---} \end{array} \right]$$

$\Pr[ab] = \frac{1}{\sqrt{ab}} \qquad \Pr[bc] = \frac{1}{\sqrt{bc}}$
 $\Pr[da] = \frac{1}{\sqrt{da}} \qquad \Pr[cd] = \frac{1}{\sqrt{cd}}$

$$= \log(n)^{O(1)} \cdot \Pr[ab] \cdot \Pr[bc] \cdot \Pr[cd] \cdot \Pr[da]$$

$$= \frac{\log(n)^{O(1)}}{abcd}$$

Bounding Number of Leaf-Free Subgraphs

expectation of $\# \square$ in G_m^n

Bounding Number of Leaf-Free Subgraphs

expectation of $\# \square$ in G_m^n

$$= \sum_{a=1}^n$$

Bounding Number of Leaf-Free Subgraphs

expectation of $\# \square$ in G_m^n

$$= \sum_{a=1}^n \sum_{b=1}^n$$

Bounding Number of Leaf-Free Subgraphs

expectation of $\# \square$ in G_m^n

$$= \sum_{a=1}^n \sum_{b=1}^n \sum_{c=1}^n$$

Bounding Number of Leaf-Free Subgraphs

expectation of $\# \square$ in G_m^n

$$= \sum_{a=1}^n \sum_{b=1}^n \sum_{c=1}^n \sum_{d=1}^n$$

Bounding Number of Leaf-Free Subgraphs

expectation of $\# \square$ in G_m^n

$$= \sum_{a=1}^n \sum_{b=1}^n \sum_{c=1}^n \sum_{d=1}^n \frac{\log(n)^{O(1)}}{abcd}$$

Bounding Number of Leaf-Free Subgraphs

expectation of $\# \square$ in G_m^n

$$\begin{aligned} &= \sum_{a=1}^n \sum_{b=1}^n \sum_{c=1}^n \sum_{d=1}^n \frac{\log(n)^{O(1)}}{abcd} \\ &= \log(n)^{O(1)} \sum_{a=1}^n \frac{1}{a} \sum_{b=1}^n \frac{1}{b} \sum_{c=1}^n \frac{1}{c} \sum_{d=1}^n \frac{1}{d} \end{aligned}$$

Bounding Number of Leaf-Free Subgraphs

expectation of $\# \square$ in G_m^n

$$\begin{aligned} &= \sum_{a=1}^n \sum_{b=1}^n \sum_{c=1}^n \sum_{d=1}^n \frac{\log(n)^{O(1)}}{abcd} \\ &= \log(n)^{O(1)} \sum_{a=1}^n \frac{1}{a} \sum_{b=1}^n \frac{1}{b} \sum_{c=1}^n \frac{1}{c} \sum_{d=1}^n \frac{1}{d} \\ &= \log(n)^{O(1)} \end{aligned}$$

(similarly for every leaf-free graph)

Analysis of run time:

Analysis of run time:

- Number of vertices: $O(n)$

Analysis of run time:

- Number of vertices: $O(n)$
- Subroutine calls per vertex: $\log(n)^{O(1)}$

Analysis of run time:

- Number of vertices: $O(n)$
- Subroutine calls per vertex: $\log(n)^{O(1)}$
- Time for each call: $O(n)$

Analysis of run time:

- Number of vertices: $O(n)$
- Subroutine calls per vertex: $\log(n)^{O(1)}$
- Time for each call: $O(n)$

Total run time: $\log(n)^{O(1)}n^2$

Analysis of run time:

- Number of vertices: $O(n)$
- Subroutine calls per vertex: $\log(n)^{O(1)}$
- Time for each call: $O(n)$

Total run time: $\log(n)^{O(1)}n^2$

With a more careful analysis: $\log(n)^{O(1)}n$

First-Order Model Checking

- \exists and \forall quantification over vertices, equality, adjacency
- cannot be solved in time $f(|\varphi|)n^{o(|\varphi|)}$ under ETH
- *Grohe, Kreutzer, Siebertz*: for fixed formula decidable in time $O(n^{1+\varepsilon})$ on nowhere-dense graph classes

Theorem

Let $\varepsilon > 0$, $m \in \mathbf{N}$ be fixed.

One can decide every property expressible in FO logic on G_m^n in expected time $O(n^{1+\varepsilon})$.

Thank you!