University of Glasgow

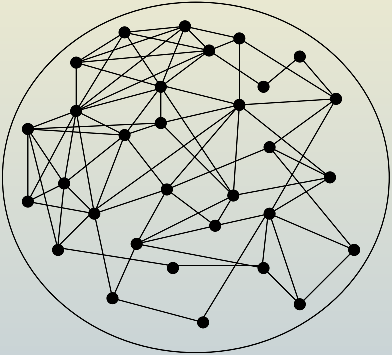
Subgraph counting problems

23rd March 2016

Kitty Meeks

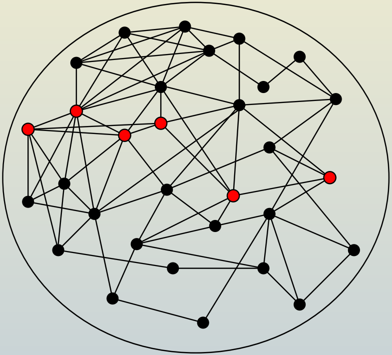
The problem

Given a graph on n vertices, we are interested in subgraphs with k vertices that have particular properties.



The problem

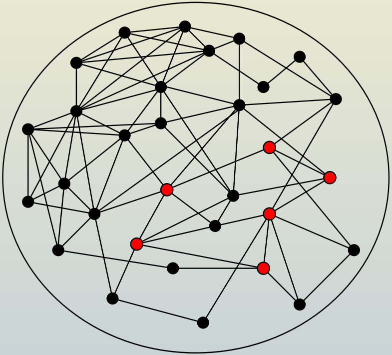
Given a graph on n vertices, we are interested in subgraphs with k vertices that have particular properties.



For example:

- Paths on k vertices

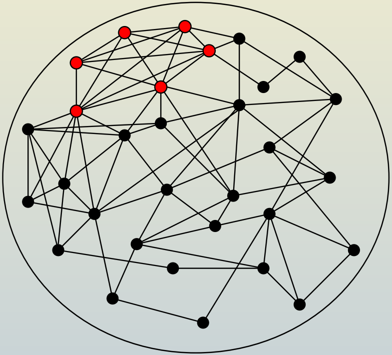
Given a graph on n vertices, we are interested in subgraphs with k vertices that have particular properties.



For example:

- Paths on k vertices
- Cycles on k vertices

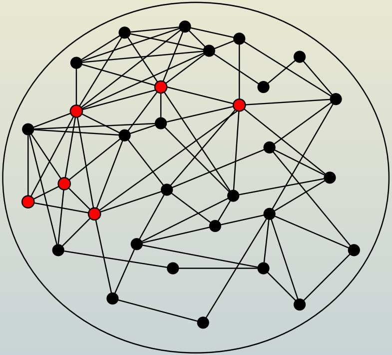
Given a graph on n vertices, we are interested in subgraphs with k vertices that have particular properties.



For example:

- Paths on k vertices
- Cycles on k vertices
- Cliques on k vertices

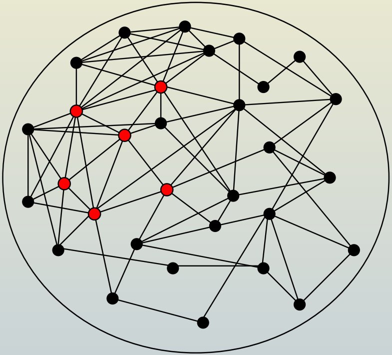
Given a graph on n vertices, we are interested in subgraphs with k vertices that have particular properties.



For example:

- Paths on k vertices
- Cycles on k vertices
- Cliques on k vertices
- Connected k-vertex induced subgraphs

Given a graph on n vertices, we are interested in subgraphs with k vertices that have particular properties.



For example:

- Paths on k vertices
- Cycles on k vertices
- Cliques on k vertices
- Connected k-vertex induced subgraphs
- k-vertex induced subgraphs with an even number of edges

DECISION

Is there a witness?

DECISION
Is there a witness?

APPROX COUNTING
Approximately how many witnesses?

DECISION
Is there a witness?

APPROX COUNTING
Approximately how many witnesses?

EXACT COUNTING
Exactly how many witnesses?

Deciding, counting and enumerating

DECISION
Is there a witness?

EXTRACTION
Identify a single witness

APPROX COUNTING
Approximately how many witnesses?

EXACT COUNTING
Exactly how many witnesses?

Deciding, counting and enumerating

DECISION
Is there a witness?

EXTRACTION
Identify a single witness

APPROX COUNTING
Approximately how many witnesses?

UNIFORM SAMPLING
Pick a single witness uniformly at random

EXACT COUNTING
Exactly how many witnesses?

Deciding, counting and enumerating

DECISION Is there a witness?	**EXTRACTION** Identify a single witness
APPROX COUNTING Approximately how many witnesses?	**UNIFORM SAMPLING** Pick a single witness uniformly at random
EXACT COUNTING Exactly how many witnesses?	**ENUMERATION** List all witnesses

The values of n and k

Consider the k-cycle problem:

- if $k = 3$ then we are interested in triangles
- if $k = n$ then we are interested in Hamilton Cycles

The values of n and k

Consider the k-cycle problem:

- if $k = 3$ then we are interested in triangles
- if $k = n$ then we are interested in Hamilton Cycles

We are interested in what happens as n and k both tend to infinity, independently, with $k << n$.

The values of n and k

Consider the k-cycle problem:

- if $k = 3$ then we are interested in triangles
- if $k = n$ then we are interested in Hamilton Cycles

We are interested in what happens as n and k both tend to infinity, independently, with $k << n$.

- We can consider all possible k-vertex subgraphs in time $O(n^k)$.

The values of n and k

Consider the k-cycle problem:

- if $k = 3$ then we are interested in triangles
- if $k = n$ then we are interested in Hamilton Cycles

We are interested in what happens as n and k both tend to infinity, independently, with $k << n$.

- We can consider all possible k-vertex subgraphs in time $O(n^k)$.
- We would like to be able to answer questions about k-vertex subgraphs in time $f(k) \cdot n^{O(1)}$; in this case the problem belongs to the parameterised complexity class FPT.

Theorem (Based on Chen, Chor, Fellows, Huang, Juedes, Kanj and Xia, 2005)

Assuming the Exponential Time Hypothesis, there is no algorithm to determine whether an n-vertex graph contains a clique on k vertices in time $f(k) \cdot n^{o(1)}$, for any function f.

Deciding, counting and enumerating

DECISION
Is there a witness?

EXTRACTION
Identify a single witness

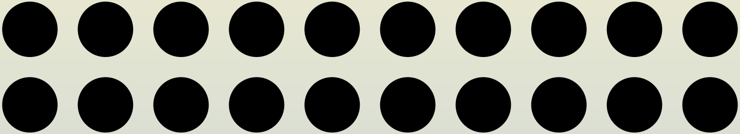
APPROX COUNTING
Approximately how many witnesses?

UNIFORM SAMPLING
Pick a single witness uniformly at random

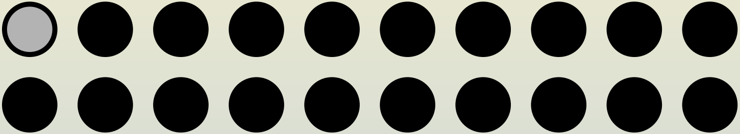
EXACT COUNTING
Exactly how many witnesses?

ENUMERATION
List all witnesses

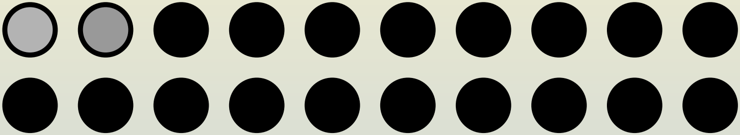
If we can decide, we can find a witness



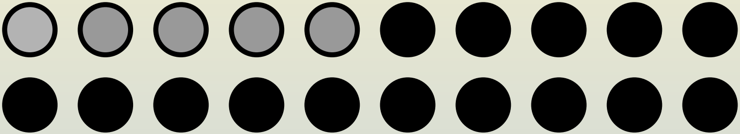
If we can decide, we can find a witness



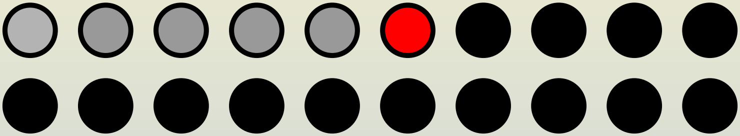
If we can decide, we can find a witness



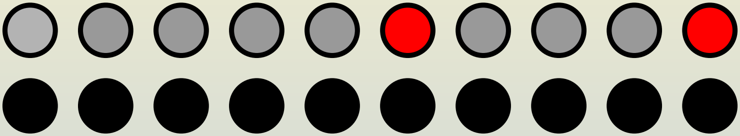
If we can decide, we can find a witness



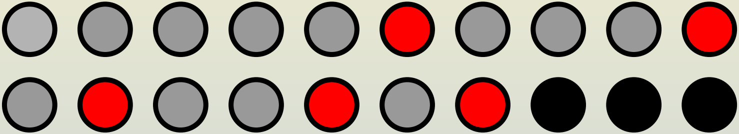
If we can decide, we can find a witness



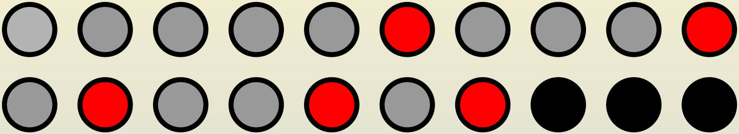
If we can decide, we can find a witness



If we can decide, we can find a witness



If we can decide, we can find a witness



Theorem (Björklund, Kaski and Kowalik, 2014)

There exists an algorithm that extracts a witness using at most

$$2k\left(\log_2 \frac{n}{k} + 2\right)$$

queries to a deterministic decision algorithm.

Deciding, counting and enumerating

DECISION Is there a witness?	**EXTRACTION** Identify a single witness
APPROX COUNTING Approximately how many witnesses?	**UNIFORM SAMPLING** Pick a single witness uniformly at random
EXACT COUNTING Exactly how many witnesses?	**ENUMERATION** List all witnesses

Exact counting is harder than decision

Question: Does G contain a connected induced subgraph on k vertices?

We can find the maximum component size, and hence answer this question, in $O(|V| + |E|)$ using a breadth first search.

Exact counting is harder than decision

Question: Does G contain a connected induced subgraph on k vertices?

We can find the maximum component size, and hence answer this question, in $O(|V| + |E|)$ using a breadth first search.

Theorem (Jerrum & M., 2015)

It is #W[1]-complete to count exactly the number of k-vertex connected induced subgraphs in a given input graph.

Exact counting is harder than decision

Question: Does G contain a connected induced subgraph on k vertices?

We can find the maximum component size, and hence answer this question, in $O(|V| + |E|)$ using a breadth first search.

Theorem (Jerrum & M., 2015)

It is #W[1]-complete to count exactly the number of k-vertex connected induced subgraphs in a given input graph.

Theorem (Jerrum & M., 2015)

There is an efficient algorithm to count approximately the number of k-vertex connected induced subgraphs in a given input graph.

Exact counting is harder than decision

Question: Does G contain a connected induced subgraph on k vertices?

We can find the maximum component size, and hence answer this question, in $O(|V| + |E|)$ using a breadth first search.

Theorem (Jerrum & M., 2015)

It is #W[1]-complete to count exactly the number of k-vertex connected induced subgraphs in a given input graph.

Theorem (Jerrum & M., 2015)

There is an efficient algorithm to count approximately the number of k-vertex connected induced subgraphs in a given input graph.

Many other examples: k-vertex paths, k-vertex cycles, k-edge matchings, k-vertex regular induced subgraphs...

Deciding, counting and enumerating

DECISION
Is there a witness?

EXTRACTION
Identify a single witness

APPROX COUNTING
Approximately how many witnesses?

UNIFORM SAMPLING
Pick a single witness uniformly at random

EXACT COUNTING
Exactly how many witnesses?

ENUMERATION
List all witnesses

If one witness implies many witnesses...

Proposition

Suppose that, for each k and any graph G on n vertices, the number of k-vertex subgraphs of G that have our property is either

1. *zero, or*
2. *at least*

$$\frac{1}{g(k)p(n)}\binom{n}{k}.$$

Then there is an efficient algorithm to count witnesses approximately.

If one witness implies many witnesses...

Proposition

Suppose that, for each k and any graph G on n vertices, the number of k-vertex subgraphs of G that have our property is either

1. *zero, or*
2. *at least*

$$\frac{1}{g(k)p(n)}\binom{n}{k}.$$

Then there is an efficient algorithm to count witnesses approximately.

Examples: k-vertex regular induced subgraphs; k-vertex induced subgraphs with an even number of edges.

If one witness implies many witnesses...

Proposition

Suppose that, for each k and any graph G on n vertices, the number of k-vertex subgraphs of G that have our property is either

1. *zero, or*
2. *at least*
$$\frac{1}{g(k)p(n)}\binom{n}{k}.$$

Then there is an efficient algorithm to count witnesses approximately.

Examples: k-vertex regular induced subgraphs; k-vertex induced subgraphs with an even number of edges.

These problems are still hard for exact counting.

If there are only $f(k) \cdot n^{O(1)}$ witnesses...

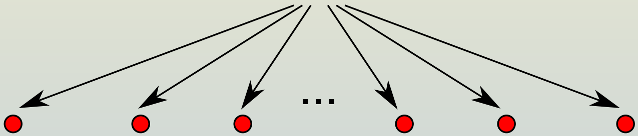
Theorem

*Suppose that we have an $f(k) \cdot n^{O(1)}$ decision algorithm for the **multicolour** version of the problem. Then we can enumerate (and hence count) all witnesses in time $g(k) \cdot n^{O(1)} \cdot N$, where N is the total number of witnesses.*

If there are only $f(k) \cdot n^{O(1)}$ witnesses...

Theorem

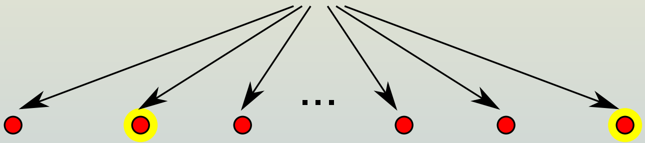
Suppose that we have an $f(k) \cdot n^{O(1)}$ decision algorithm for the ***multicolour*** *version of the problem. Then we can enumerate (and hence count) all witnesses in time $g(k) \cdot n^{O(1)} \cdot N$, where N is the total number of witnesses.*



If there are only $f(k) \cdot n^{O(1)}$ witnesses...

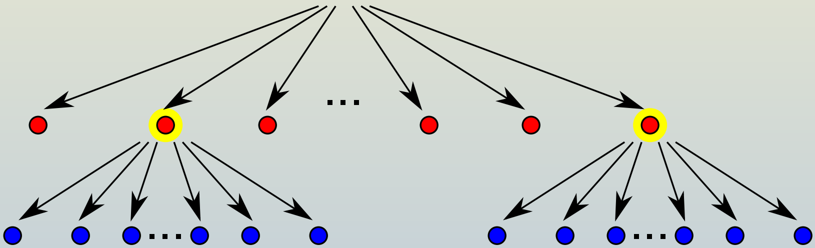
Theorem

*Suppose that we have an $f(k) \cdot n^{O(1)}$ decision algorithm for the **multicolour** version of the problem. Then we can enumerate (and hence count) all witnesses in time $g(k) \cdot n^{O(1)} \cdot N$, where N is the total number of witnesses.*



Theorem

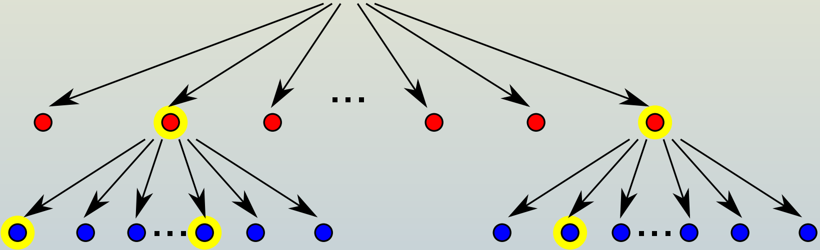
*Suppose that we have an $f(k) \cdot n^{O(1)}$ decision algorithm for the **multicolour** version of the problem. Then we can enumerate (and hence count) all witnesses in time $g(k) \cdot n^{O(1)} \cdot N$, where N is the total number of witnesses.*



If there are only $f(k) \cdot n^{O(1)}$ witnesses...

Theorem

Suppose that we have an $f(k) \cdot n^{O(1)}$ decision algorithm for the ***multicolour*** *version of the problem. Then we can enumerate (and hence count) all witnesses in time $g(k) \cdot n^{O(1)} \cdot N$, where N is the total number of witnesses.*



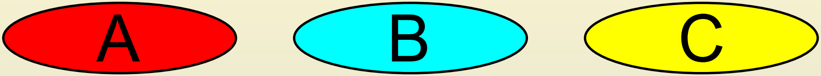
A randomised approach for the general case

- IDEA: create many coloured instances, and enumerate the colourful copies in each (omitting duplicates)

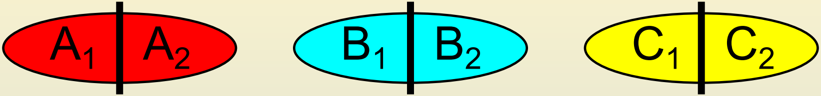
A randomised approach for the general case

- IDEA: create many coloured instances, and enumerate the colourful copies in each (omitting duplicates)
- PROBLEM: although we're now looking for colourful witnesses, we still only have a decision algorithm for the uncoloured version...

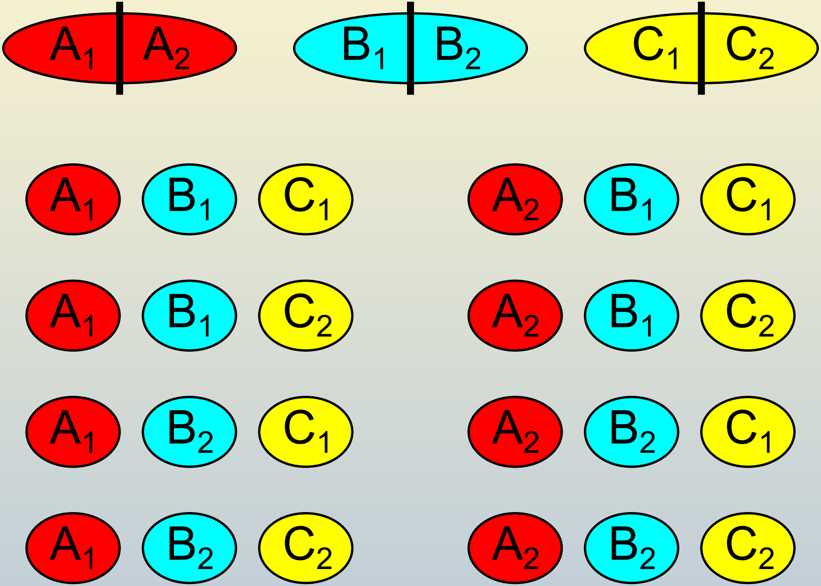
A randomised approach for the general case



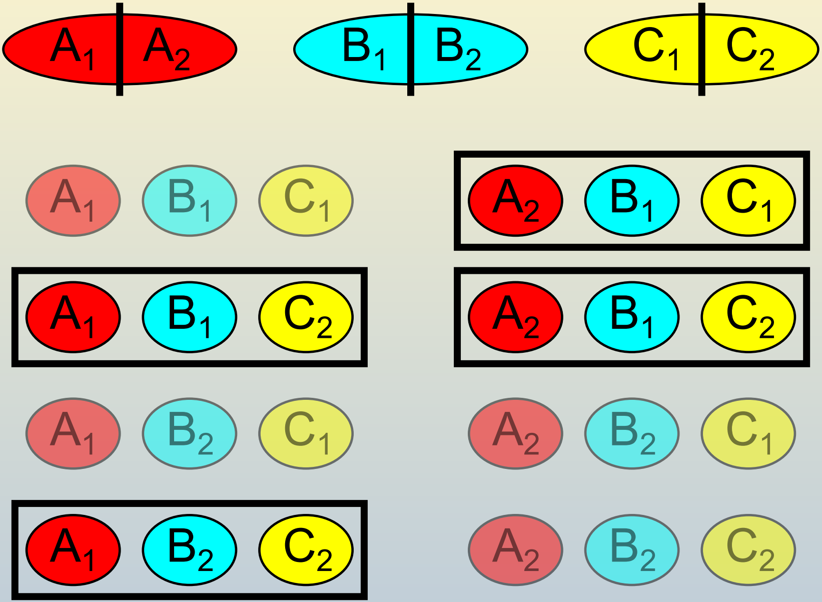
A randomised approach for the general case



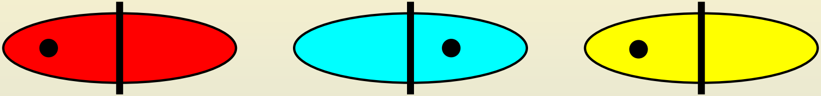
A randomised approach for the general case



A randomised approach for the general case



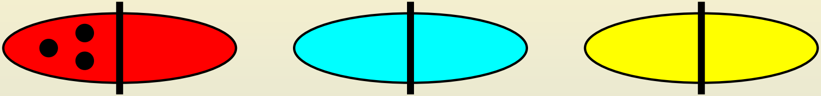
A randomised approach



If a witness is colourful:

- It will always survive in exactly one combination

A randomised approach



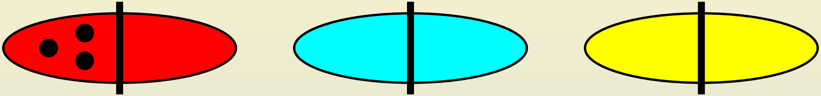
If a witness is colourful:

- It will always survive in exactly one combination

If a witness contains vertices of only $\ell < k$ colours:

- the probability it survives in at least one combination is at most $2^{-(k-\ell)}$
- if it survives in any combination, it will survive in exactly $2^{k-\ell}$ combinations

A randomised approach



If a witness is colourful:

- It will always survive in exactly one combination

If a witness contains vertices of only $\ell < k$ colours:

- the probability it survives in at least one combination is at most $2^{-(k-\ell)}$
- if it survives in any combination, it will survive in exactly $2^{k-\ell}$ combinations

It can then be shown that, for **any** witness, the **expected** number of combinations in which it survives at each level is at most one.

Theorem

Suppose that we can decide if there is at least one witness in time $f(k) \cdot n^{O(1)}$. Then there is a randomised algorithm which enumerates all witnesses in expected time $f(k) \cdot n^{O(1)} \cdot N$, where N is the total number of witnesses in the instance.

Theorem

Suppose that we can decide if there is at least one witness in time $f(k) \cdot n^{O(1)}$. Then there is a randomised algorithm which enumerates all witnesses in expected time $f(k) \cdot n^{O(1)} \cdot N$, where N is the total number of witnesses in the instance.

Corollary

Suppose that we can decide if there is at least one witness in time $f(k) \cdot n^{O(1)}$ and that, for each k and any graph G on n vertices, the total number of witnesses is at most $f(k)n^{O(1)}$. Then there is an efficient algorithm to count witnesses approximately.

Open problems

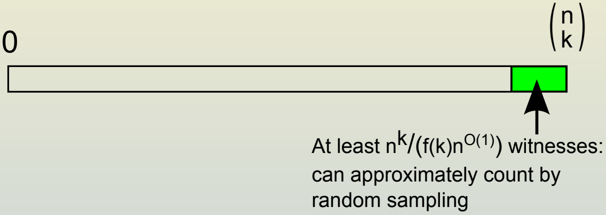
- Can the randomised enumeration process be derandomised?

- Can the randomised enumeration process be derandomised?
- Can we close the gap?

0 $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\binom{n}{k}$

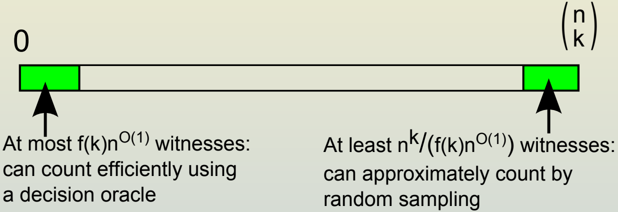
Open problems

- Can the randomised enumeration process be derandomised?
- Can we close the gap?



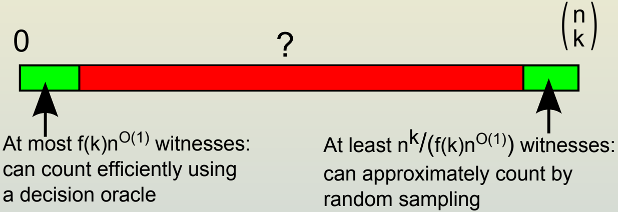
Open problems

- Can the randomised enumeration process be derandomised?
- Can we close the gap?



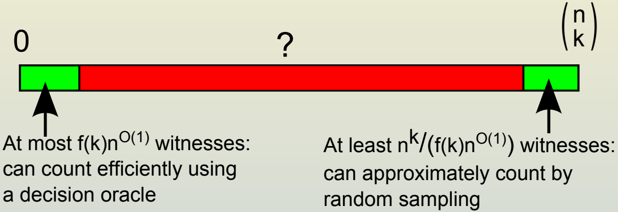
Open problems

- Can the randomised enumeration process be derandomised?
- Can we close the gap?



Open problems

- Can the randomised enumeration process be derandomised?
- Can we close the gap?



Thank you