Conjunctive Queries on Probabilistic Graphs: The Limits of Approximability

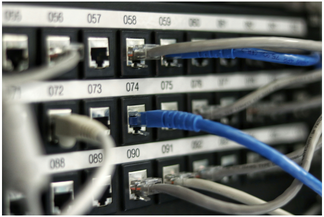
Antoine Amarilli[1] Timothy van Bremen[2] Kuldeep S. Meel[3]

[1] Télécom Paris

[2] National University of Singapore

[3] University of Toronto

Motivating Question 1: Operations Research

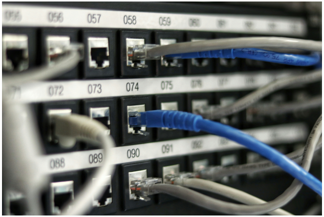


The **two-terminal network reliability problem** asks the following:

> *Given a directed graph with independent edge failure probabilities, and two vertices s and t, determine the probability that s and t are connected.*

Applications to verifying reliability of power transmission networks, computer networks, etc.

Motivating Question 1: Operations Research



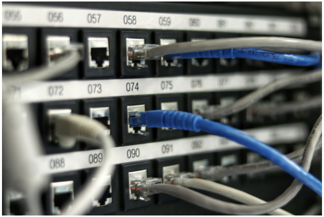
The **two-terminal network reliability problem** asks the following:

> *Given a directed graph with independent edge failure probabilities, and two vertices s and t, determine the probability that s and t are connected.*

Applications to verifying reliability of power transmission networks, computer networks, etc.

When can we get a **fully polynomial-time randomized approximation scheme (FPRAS)** for two-terminal network reliability?

Motivating Question 1: Operations Research



The **two-terminal network reliability problem** asks the following:

> *Given a directed graph with independent edge failure probabilities, and two vertices s and t, determine the probability that s and t are connected.*

Applications to verifying reliability of power transmission networks, computer networks, etc.

When can we get a **fully polynomial-time randomized approximation scheme (FPRAS)** for two-terminal network reliability?

Today: When the graph is a **DAG**, we can!

Motivating Question 2: Probabilistic Databases

Add **probability labelling** π to database D to get a **tuple-independent probabilistic database** (TID) $H = (D, \pi)$.

Classes

	Lec	Rm	Time
0.5	alice	02.10	10
0.1	bob	01.5	9
0.5	charlie	01.6	10

Mentors

	Lec	Student
0.2	alice	david
0.5	bob	emma

Query: is there someone who teaches a class at 10 and mentors David?

$$q = \exists x \exists y. \text{Classes}(x, y, 10) \wedge \text{Mentors}(x, \text{david})$$

Returns: ~~$q(D) = \texttt{true}$~~ $\Pr_H(q) =$ **?**

Motivating Question 2: Probabilistic Databases

Theorem [van Bremen and Meel, PODS 2023]

Let q be a Boolean conjunctive query that:

- has bounded hypertree width
- is self-join-free
  - e.g., $\exists xy.R(x) \wedge S(x, y)$ ✓ $\exists xy.R(x, y) \wedge R(y, z)$ ✗

Then q can be **tractably approximated (FPRAS) in combined complexity** on **any** TID.

Can we **relax** either of the two conditions on the query above and still always get an FPRAS?

Motivating Question 2: Probabilistic Databases

Theorem [van Bremen and Meel, PODS 2023]

Let q be a Boolean conjunctive query that:

- has bounded hypertree width
- is self-join-free
  - e.g., $\exists xy. R(x) \wedge S(x, y)$ ✓ $\exists xy. R(x, y) \wedge R(y, z)$ ✗

Then q can be **tractably approximated (FPRAS) in combined complexity** on **any** TID.

Can we **relax** either of the two conditions on the query above and still always get an FPRAS?

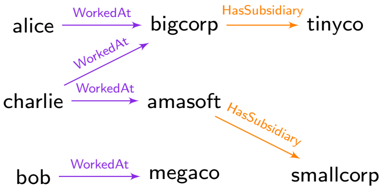
Today: **No!** (assuming RP $\neq$ NP)

Data as Graphs

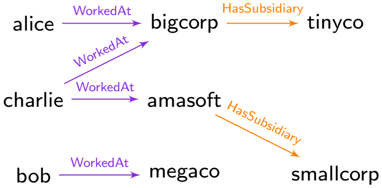
To answer these motivating questions (among others), we consider the restricted setting of **binary signatures**—i.e., data represented as a **labelled graph**.

WorkedAt	
Alice	BigCorp
Bob	MegaCo
Charlie	AmaSoft
Charlie	BigCorp

HasSubsidiary	
BigCorp	TinyCo
AmaSoft	SmallCorp

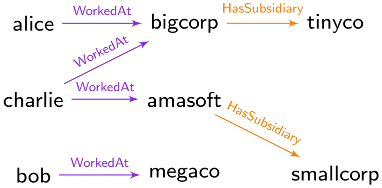


Querying Graphs



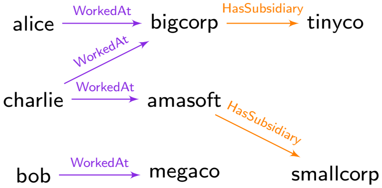
- Consider **Boolean** (yes/no) queries on graphs

Querying Graphs



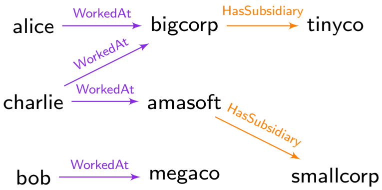
- Consider **Boolean** (yes/no) queries on graphs
- We can ask: is there a match of a pattern?
  - e.g., $x \xrightarrow{\text{WorkedAt}} y \xrightarrow{\text{HasSubsidiary}} z$
  - Yes
  - CQ: WorkedAt(x, y), HasSub(y, z)

Querying Graphs



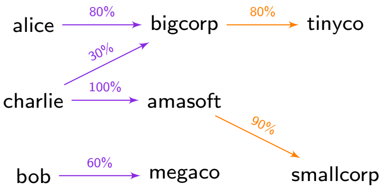
- Consider **Boolean** (yes/no) queries on graphs
- We can ask: is there a match of a pattern?
  - e.g., $x \xrightarrow{\text{WorkedAt}} y \xrightarrow{\text{HasSubsidiary}} z$
  - Yes
  - CQ: WorkedAt(x, y), HasSub(y, z)
- More formally: matches are **homomorphisms** from a query graph
  - these homomorphisms need not be injective!
  - e.g., $x \xrightarrow{\text{WorkedAt}} y \xleftarrow{\text{WorkedAt}} z$
  - Yes
  - CQ: WorkedAt(x, y), WorkedAt(z, y)

Querying Graphs



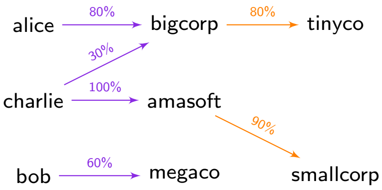
- Consider **Boolean** (yes/no) queries on graphs
- We can ask: is there a match of a pattern?
  - e.g., $x \xrightarrow{\text{WorkedAt}} y \xrightarrow{\text{HasSubsidiary}} z$
  - Yes
  - CQ: WorkedAt(x, y), HasSub(y, z)
- More formally: matches are **homomorphisms** from a query graph
  - these homomorphisms need not be injective!
  - e.g., $x \xrightarrow{\text{WorkedAt}} y \xleftarrow{\text{WorkedAt}} z$
  - Yes
  - CQ: WorkedAt(x, y), WorkedAt(z, y)
- Denote that G has a match in H by $G \rightsquigarrow H$

Uncertain Data



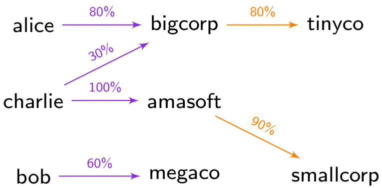
- **Probabilistic** labelled graphs

Uncertain Data



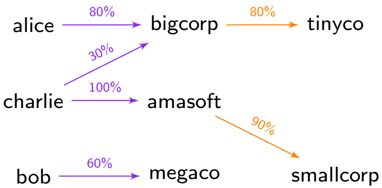
- **Probabilistic** labelled graphs
- Each edge carries an **independent** probability

Uncertain Data



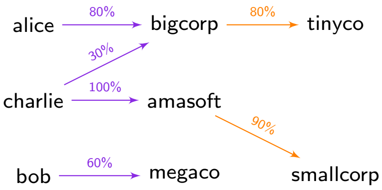
- **Probabilistic** labelled graphs
- Each edge carries an **independent** probability
- Each edge exists in the graph with its given probability

Uncertain Data



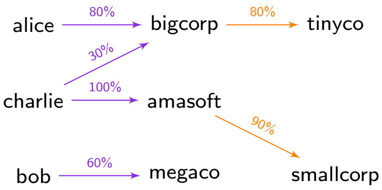
- **Probabilistic** labelled graphs
- Each edge carries an **independent** probability
- Each edge exists in the graph with its given probability
- Vertices always stay fixed

Uncertain Data



- **Probabilistic** labelled graphs
- Each edge carries an **independent** probability
- Each edge exists in the graph with its given probability
- Vertices always stay fixed
- Probability distribution on $2^{|H|}$ **subgraphs**

Uncertain Data



- **Probabilistic** labelled graphs
- Each edge carries an **independent** probability
- Each edge exists in the graph with its given probability
- Vertices always stay fixed
- Probability distribution on $2^{|H|}$ **subgraphs**
- Special case when all probabilities are 50% $\rightarrow$ every subgraph is equally likely

Probabilistic Graph Homomorphism

$\mathsf{PHom}_{\mathsf{L}}(\mathcal{G}, \mathcal{H})$

Given:

- **labelled** (non-probabilistic) "query" graph $G \in \mathcal{G}$
- probabilistic **labelled** "instance" graph $H \in \mathcal{H}$

Compute: probability that a randomly sampled subgraph $H' \subseteq H$ admits a homomorphism from G:

$$\Pr(G \rightsquigarrow H) = \sum_{H' \subseteq H \text{ s.t. } G \rightsquigarrow H'} \prod_{e \in H'} \Pr(e) \prod_{e \in H \setminus H'} (1 - \Pr(e))$$

Observe that the problem is stated in terms of *combined complexity* (both query and instance as input).

Probabilistic Graph Homomorphism

$\mathrm{PHom}_{\not L}(\mathcal{G}, \mathcal{H})$

Given:

- **unlabelled** (non-probabilistic) "query" graph $G \in \mathcal{G}$
- probabilistic **unlabelled** "instance" graph $H \in \mathcal{H}$

Compute: probability that a randomly sampled subgraph $H' \subseteq H$ admits a homomorphism from G:

$$\Pr(G \rightsquigarrow H) = \sum_{H' \subseteq H \text{ s.t. } G \rightsquigarrow H'} \prod_{e \in H'} \Pr(e) \prod_{e \in H \setminus H'} (1 - \Pr(e))$$

Observe that the problem is stated in terms of *combined complexity* (both query and instance as input).

Graph Classes

Many possible choices for graph classes $\mathcal{G}$ and $\mathcal{H}$:

- The class 1WP of **one-way paths**:

$$a_1 \xrightarrow{R_1} \dots \xrightarrow{R_{m-1}} a_m$$

- The class of **two-way paths** (2WP) of the form:

$$a_1 \; - \; \dots \; - \; a_m$$

  with each $-$ being $\xrightarrow{R_i}$ or $\xleftarrow{R_i}$

- ...

Previous Work

The complexity of probabilistic graph homomorphism has been studied before for various combinations of graph classes $\mathcal{G}$ (query) and $\mathcal{H}$ (instance).

[Amarilli, Monet, and Senellart, PODS 2017]

Existing results imply the tables below.

Table: Complexity of $\mathrm{PHom}_{\mathrm{L}}(\mathcal{G}, \mathcal{H})$.

$\mathcal{G} \downarrow$ \ $\mathcal{H} \rightarrow$	1WP	2WP	DWT	PT	DAG	All
1WP				■	■	■
2WP			■	■	■	■
DWT			■	■	■	■
PT			■	■	■	■

Table: Complexity of $\mathrm{PHom}_{\not\mathrm{L}}(\mathcal{G}, \mathcal{H})$.

$\mathcal{G} \downarrow$ \ $\mathcal{H} \rightarrow$	1WP	2WP	DWT	PT	DAG	All
1WP					■	■
2WP				■	■	■
DWT					■	■
PT				■	■	■

- white ( ) means that the problem lies in P
- dark grey (■) means #P-hardness

Previous Work

The complexity of probabilistic graph homomorphism has been studied before for various combinations of graph classes $\mathcal{G}$ (query) and $\mathcal{H}$ (instance).

[Amarilli, Monet, and Senellart, PODS 2017]

Existing results imply the tables below.

Table: Complexity of $\mathrm{PHom}_{\mathrm{L}}(\mathcal{G}, \mathcal{H})$.

$\mathcal{G} \downarrow$ \ $\mathcal{H} \rightarrow$	1WP	2WP	DWT	PT	DAG	All
1WP				■	■	■
2WP			■	■	■	■
DWT			■	■	■	■
PT			■	■	■	■

Table: Complexity of $\mathrm{PHom}_{\not\mathrm{L}}(\mathcal{G}, \mathcal{H})$.

$\mathcal{G} \downarrow$ \ $\mathcal{H} \rightarrow$	1WP	2WP	DWT	PT	DAG	All
1WP					■	■
2WP				■	■	■
DWT					■	■
PT				■	■	■

- white ( ) means that the problem lies in P
- dark grey (■) means #P-hardness

What about for **approximations**?

FPRAS for Probabilistic Graph Homomorphism

FPRAS for $\mathrm{PHom}_{\mathrm{L}}(\mathcal{G}, \mathcal{H})/\mathrm{PHom}_{\not\mathrm{L}}(\mathcal{G}, \mathcal{H})$

Given:

- (non-probabilistic) "query" graph $G \in \mathcal{G}$
- probabilistic "instance" graph $H \in \mathcal{H}$
- ϵ, $\delta > 0$

Compute: a quantity t such that

$$\Pr\left[(1-\epsilon)\Pr(G \rightsquigarrow H) \leq t \leq (1+\epsilon)\Pr(G \rightsquigarrow H)\right] \geq 1-\delta$$

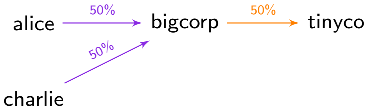
in time polynomial in $|G|$, $|H|$, ϵ^{-1}, and δ^{-1}

FPRAS for Probabilistic Graph Homomorphism

Query G**:**

$$x \xrightarrow{\text{WorkedAt}} y \xrightarrow{\text{HasSubsidiary}} z$$

Instance H**:**



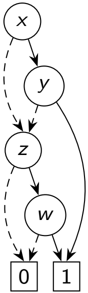
- Transform the instance graph to one in which all probabilities are 50%—the problem now is equivalent to **counting subgraphs** that admit a homomorphism from G
- **The key idea**: **intensional query evaluation**. We build a *non-deterministic ordered binary decision diagram* (nOBDD) Δ that represents the **Boolean provenance** of G on H. Satisfying assignments of Δ are in **bijection** with the subgraphs of H admitting a homomorphism from G
- We can then **apply an off-the-shelf FPRAS** for counting the satisfying assignments of Δ

[Arenas, Croquevielle, Jayaram, and Riveros, J. ACM 2021]

Crash course: (n)OBDDs

Ordered binary decision diagrams (OBDDs): compact representations of Boolean functions.

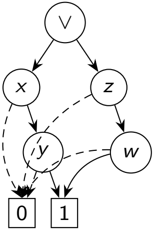
$(x \wedge y) \vee (z \wedge w)$



Crash course: (n)OBDDs

Non-deterministic ordered binary decision diagrams (nOBDDs): **even more** compact representations of Boolean functions.

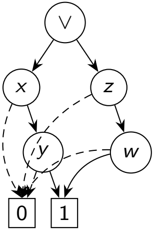
$(x \land y) \lor (z \land w)$



Crash course: (n)OBDDs

Non-deterministic ordered binary decision diagrams (nOBDDs): **even more** compact representations of Boolean functions.

$(x \wedge y) \vee (z \wedge w)$



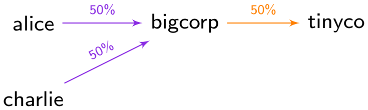
Theorem [Arenas, Croquevielle, Jayaram, and Riveros, J. ACM 2021]

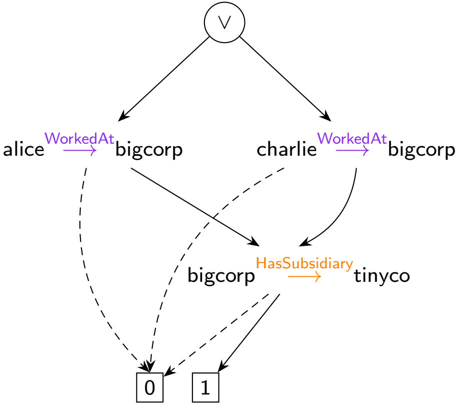
Every nOBDD admits an FPRAS for counting its satisfying assignments.

Probabilistic Graph Homomorphism via nOBDDs

Query:

Instance:



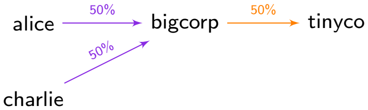


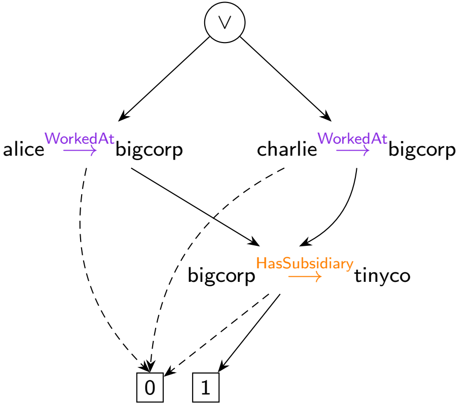
Probabilistic Graph Homomorphism via nOBDDs

Query:

$$x \xrightarrow{\text{WorkedAt}} y \xrightarrow{\text{HasSubsidiary}} z$$

Instance:





Theorem

$\mathrm{PHom}_{\mathrm{L}}(\mathrm{1WP}, \mathrm{DAG})$ admits an FPRAS.

Refined Perspective

We can also show a number of **inapproximability** results (not discussed today), conditional on RP $\neq$ NP.

Taken together with our approximability results, we may **refine the table earlier**:

Table: Complexity of $\mathrm{PHom}_{\mathrm{L}}(\mathcal{G}, \mathcal{H})$.

$\mathcal{G} \downarrow$	$\mathcal{H} \rightarrow$					
	1WP	2WP	DWT	PT	DAG	All
1WP						
2WP						
DWT						
PT						

Table: Complexity of $\mathrm{PHom}_{\not\mathrm{L}}(\mathcal{G}, \mathcal{H})$.

$\mathcal{G} \downarrow$	$\mathcal{H} \rightarrow$					
	1WP	2WP	DWT	PT	DAG	All
1WP						?
2WP						
DWT						?
PT						

- white ( ) means that the problem lies in P
- light grey (■) means #P-hardness but existence of an FPRAS
- dark grey (■) means #P-hardness and non-existence of an FPRAS assuming RP $\neq$ NP.

Refined Perspective

We can also show a number of **inapproximability** results (not discussed today), conditional on RP $\neq$ NP.

Taken together with our approximability results, we may **refine the table earlier**:

Table: Complexity of $\mathrm{PHom}_{\mathrm{L}}(\mathcal{G}, \mathcal{H})$.

$\mathcal{G} \downarrow$	$\mathcal{H} \rightarrow$					
	1WP	2WP	DWT	PT	DAG	All
1WP						
2WP						
DWT						
PT						

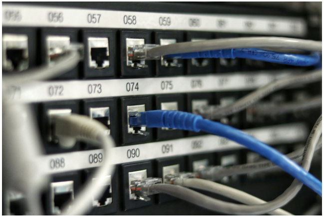
Table: Complexity of $\mathrm{PHom}_{\not\mathrm{L}}(\mathcal{G}, \mathcal{H})$.

$\mathcal{G} \downarrow$	$\mathcal{H} \rightarrow$					
	1WP	2WP	DWT	PT	DAG	All
1WP						?
2WP						
DWT						?
PT						

- white ( ) means that the problem lies in P
- light grey (■) means #P-hardness but existence of an FPRAS
- dark grey (■) means #P-hardness and non-existence of an FPRAS assuming RP $\neq$ NP.

We also get **unconditional** circuit lower bounds on the size of Boolean provenance representations in a mildly tractable form (DNNF), for all of the inapproximable pairs.

Application to Operations Research

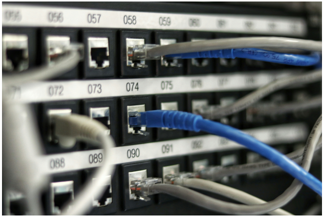


The **two-terminal network reliability problem** asks the following:

> *Given a directed graph with independent edge failure probabilities, and two vertices s and t, determine the probability that s and t are connected.*

Applications to verifying reliability of power transmission networks, computer networks, etc.

Application to Operations Research



The **two-terminal network reliability problem** asks the following:

> *Given a directed graph with independent edge failure probabilities, and two vertices s and t, determine the probability that s and t are connected.*

Applications to verifying reliability of power transmission networks, computer networks, etc.

Theorem

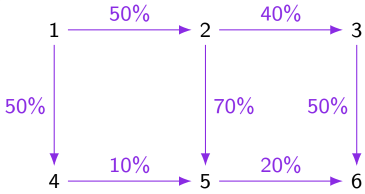
The two-terminal network reliability problem on DAGs admits an FPRAS.

Was an **open problem** specifically posed for DAGs. [Zenklusen and Laumanns, Networks 2010]

Network Reliability: Example 1

Consider **computing the probability that nodes 1 and 6 are connected**, where each link fails independently with a given probability.

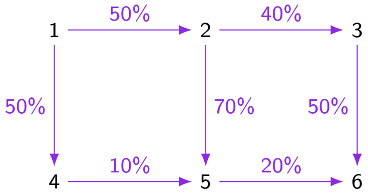
Instance $H =$



Network Reliability: Example 1

Consider **computing the probability that nodes 1 and 6 are connected**, where each link fails independently with a given probability.

Instance $H =$

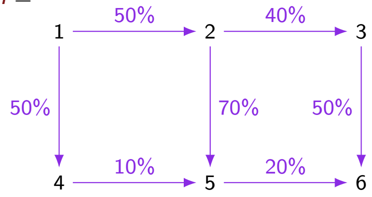


Query $G = \longrightarrow\longrightarrow\longrightarrow$

Network Reliability: Example 1

Consider **computing the probability that nodes 1 and 6 are connected**, where each link fails independently with a given probability.

Instance H =



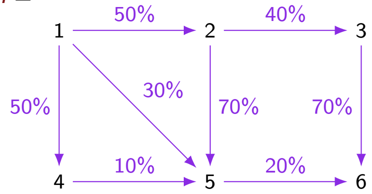
Query $G = \longrightarrow\longrightarrow\longrightarrow$

$$\Pr(\text{node 1 and 6 connected}) = \Pr(G \rightsquigarrow H)$$

Network Reliability: Example 2

Consider **computing the probability that nodes 1 and 6 are connected**, where each link fails independently with a given probability.

Instance $H =$



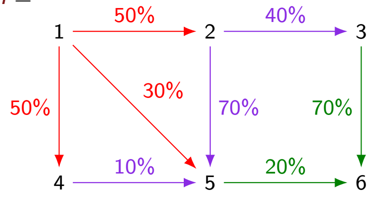
Query $G = \longrightarrow\longrightarrow\longrightarrow$

$$\Pr(\text{node 1 and 6 connected}) \neq \Pr(G \rightsquigarrow H)$$

Network Reliability: Example 2

Consider **computing the probability that nodes 1 and 6 are connected**, where each link fails independently with a given probability.

Instance H =



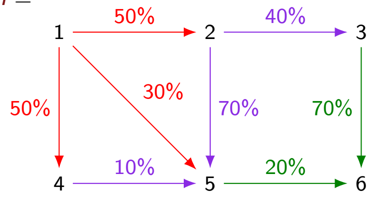
Queries $G_1 = \longrightarrow\longrightarrow\longrightarrow$ and $G_2 = \longrightarrow\longrightarrow$

Pr(node 1 and 6 connected) = Pr(subgraph of H admits a homomorphism from G_1 or G_2)

Network Reliability: Example 2

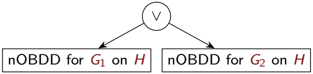
Consider **computing the probability that nodes 1 and 6 are connected**, where each link fails independently with a given probability.

Instance $H =$



Queries $G_1 = \longrightarrow\longrightarrow\longrightarrow$ and $G_2 = \longrightarrow\longrightarrow$

Pr(node 1 and 6 connected) = Pr(subgraph of H admits a homomorphism from G_1 or G_2)



Conclusion and Future Work

Recap

- Studied the **(in)approximability of probabilistic graph homomorphism in combined complexity**, and also showed lower bounds on tractable (DNNF) provenance circuit sizes
- Results show that #P-**hardness usually implies hardness of approximation**, with **important exception** of one-way path queries on DAGs

Conclusion and Future Work

Recap

- Studied the **(in)approximability of probabilistic graph homomorphism in combined complexity**, and also showed lower bounds on tractable (DNNF) provenance circuit sizes
- Results show that #P-**hardness usually implies hardness of approximation**, with **important exception** of one-way path queries on DAGs

Future work

- Figuring out **missing gaps** (approximability status of $\mathrm{PHom}_{\not L}(\mathrm{1WP}, \mathrm{All})$ and $\mathrm{PHom}_{\not L}(\mathrm{DWT}, \mathrm{All})$)
- Extensions to **richer queries and graph classes** (e.g., bounded DAG-width, disconnected queries, recursion)
- Lifting to general prob. database setting, i.e., signatures of **arbitrary arity**

Thank you! Questions?