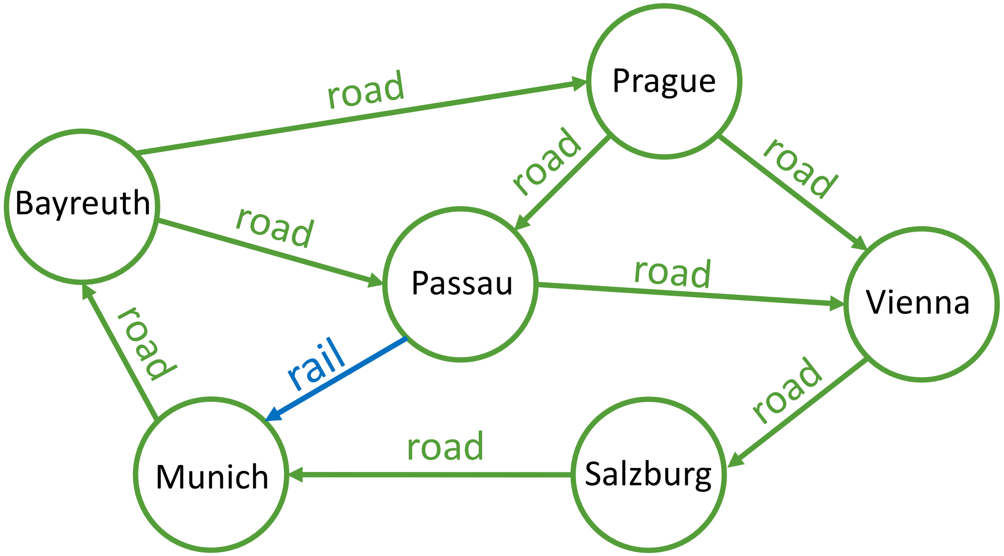
Evaluation and Enumeration Problems for Regular Path Queries

Wim Martens and Tina Trautner
University of Bayreuth

Practice

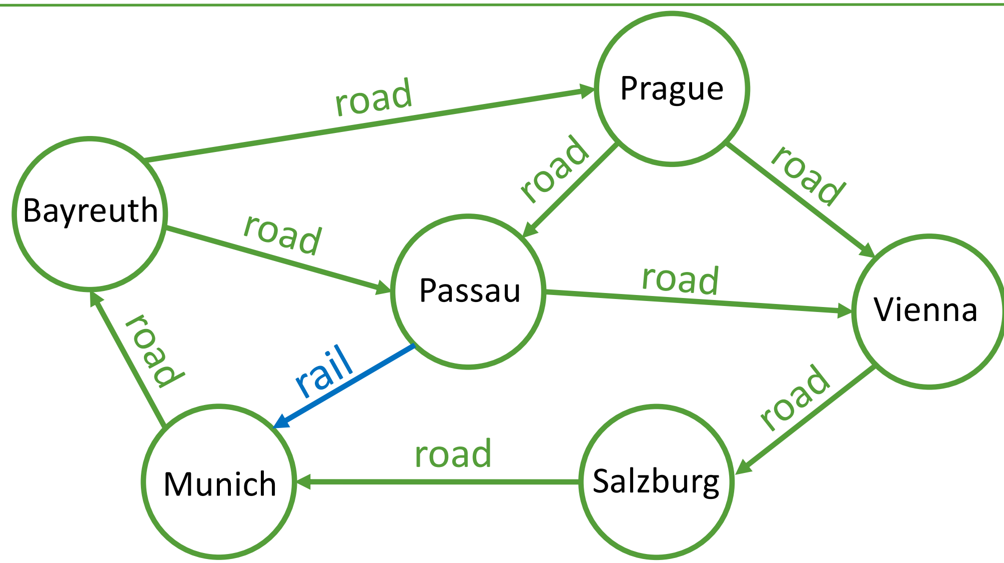
QUERYING PATHS IN GRAPH DATABASES

Graph Database



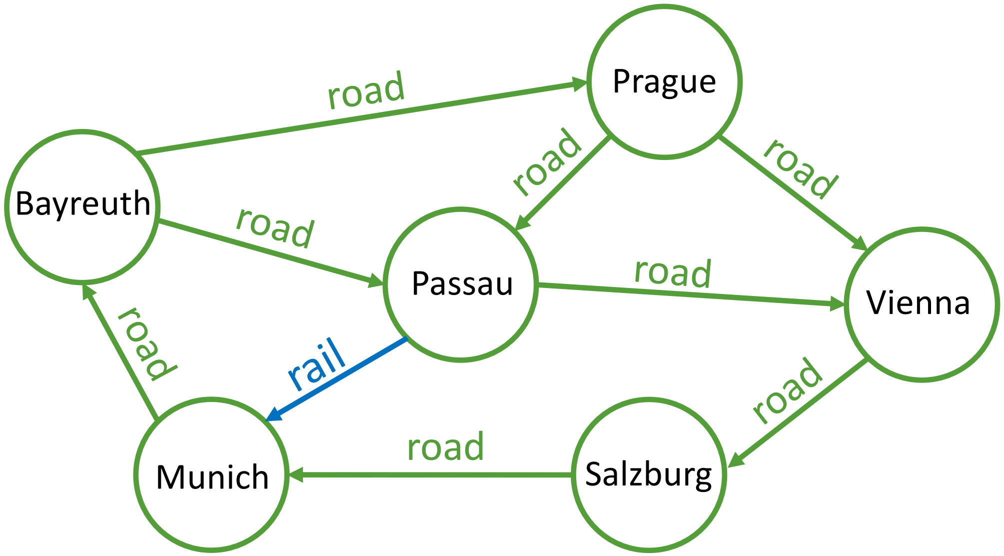
Node- and Edge-labeled directed graph

Warm Up Question



How many paths from Bayreuth to Vienna match the regular path query **(road)*** ?

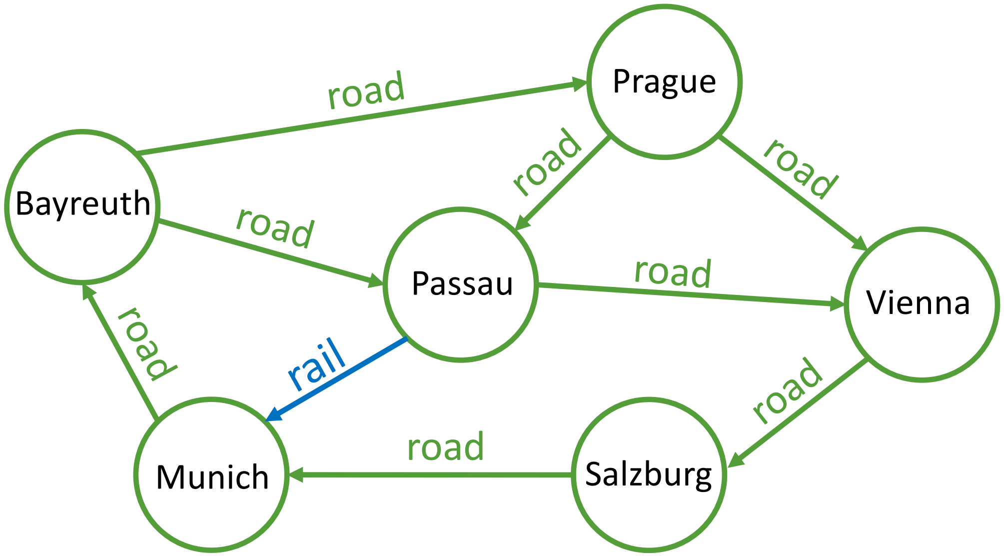
Warm Up Question



How many paths from Bayreuth to Vienna match the regular path query **(road)*** ?

(How many paths from Bayreuth to Vienna only use road-edges?)

Warm Up Question

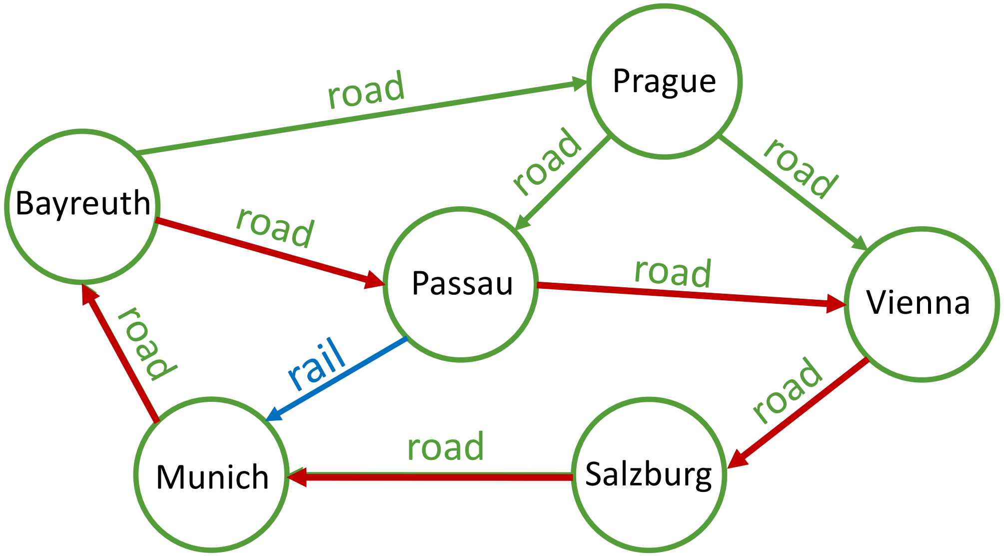


How many paths from Bayreuth to Vienna match the regular path query **(road)*** ?

(How many paths from Bayreuth to Vienna only use road-edges?)

[Theoreticians]:	∞
[SPARQL 2018]:	1
[SPARQL 2012]:	3
[Cypher]:	5

Warm Up Question

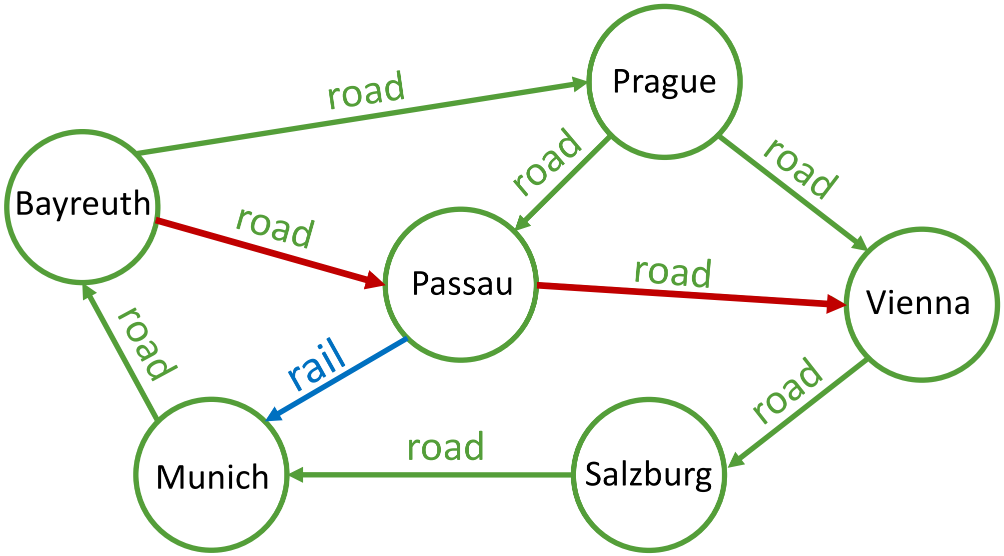


How many paths from Bayreuth to Vienna match the regular path query **(road)*** ?

(How many paths from Bayreuth to Vienna only use road-edges?)

[Theoreticians]:	∞	all paths
[SPARQL 2018]:	1	
[SPARQL 2012]:	3	
[Cypher]:	5	

Warm Up Question

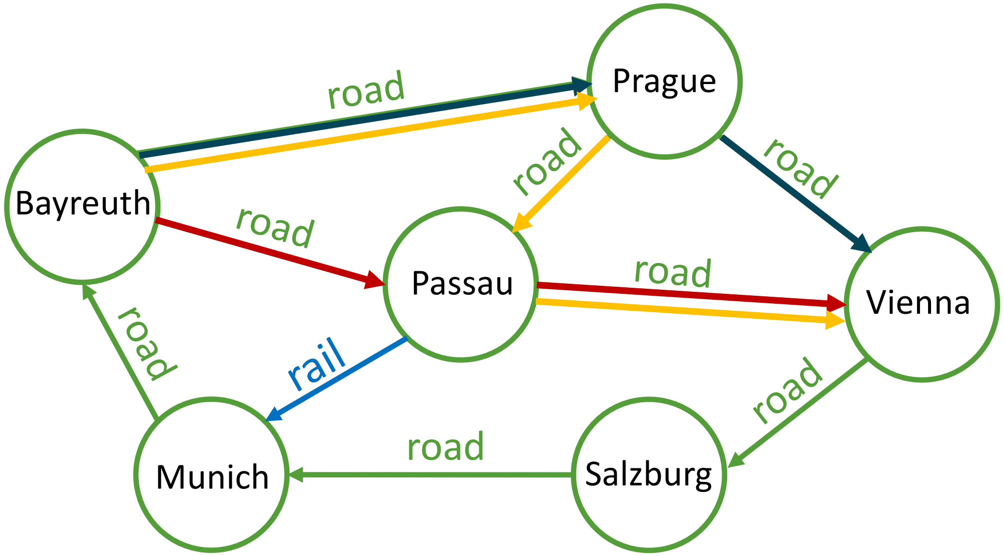


How many paths from Bayreuth to Vienna match the regular path query **(road)*** ?

(How many paths from Bayreuth to Vienna only use road-edges?)

[Theoreticians]:	∞	all paths
[SPARQL 2018]:	1	is there at least one path?
[SPARQL 2012]:	3	
[Cypher]:	5	

Warm Up Question

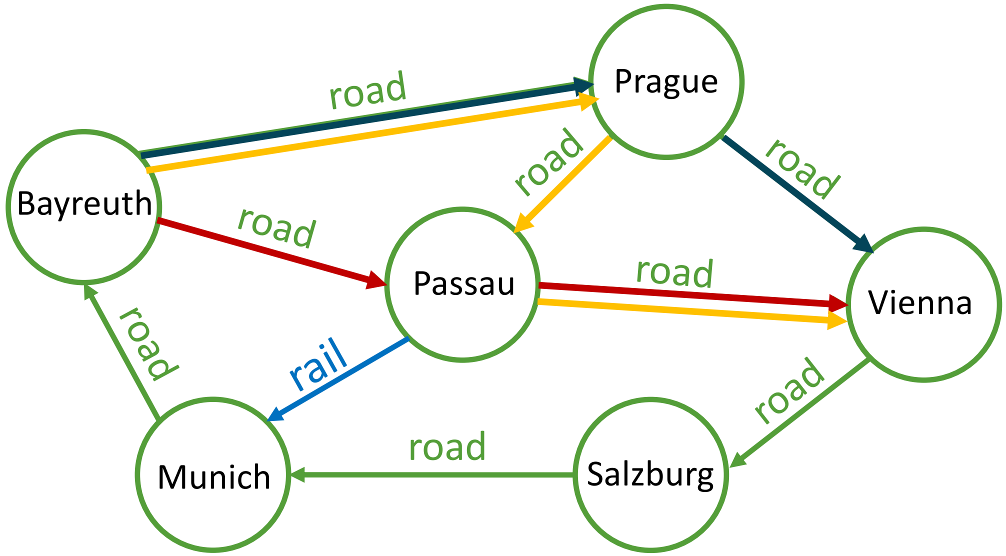


How many paths from Bayreuth to Vienna match the regular path query **(road)*** ?

(How many paths from Bayreuth to Vienna only use road-edges?)

[Theoreticians]:	∞	all paths
[SPARQL 2018]:	1	is there at least one path?
[SPARQL 2012]:	3	paths without node repetition
[Cypher]:	5	

Warm Up Question

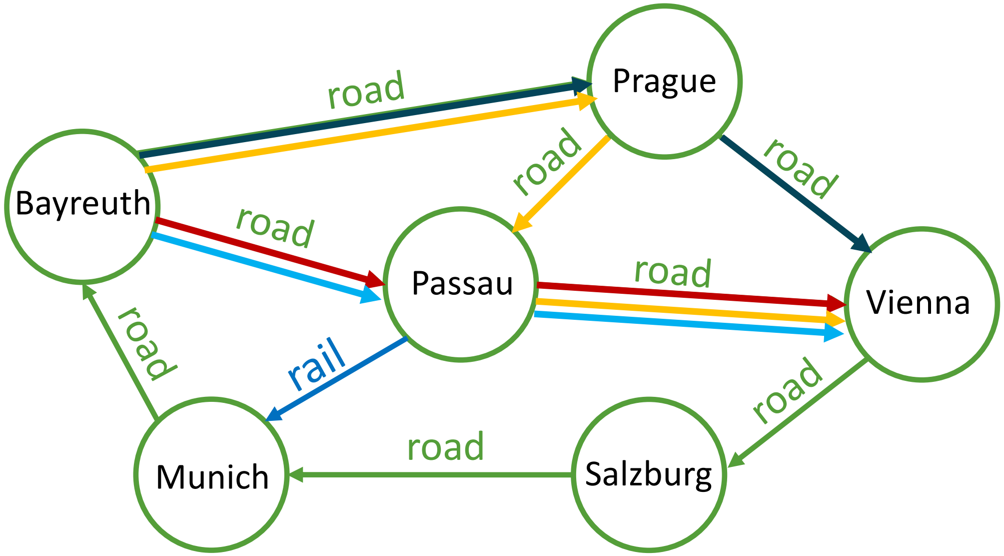


How many paths from Bayreuth to Vienna match the regular path query **(road)*** ?

(How many paths from Bayreuth to Vienna only use road-edges?)

[Theoreticians]:	∞	all paths
[SPARQL 2018]:	1	is there at least one path?
[SPARQL 2012]:	3	paths without node repetition
[Cypher]:	5	paths without edge repetition

Warm Up Question

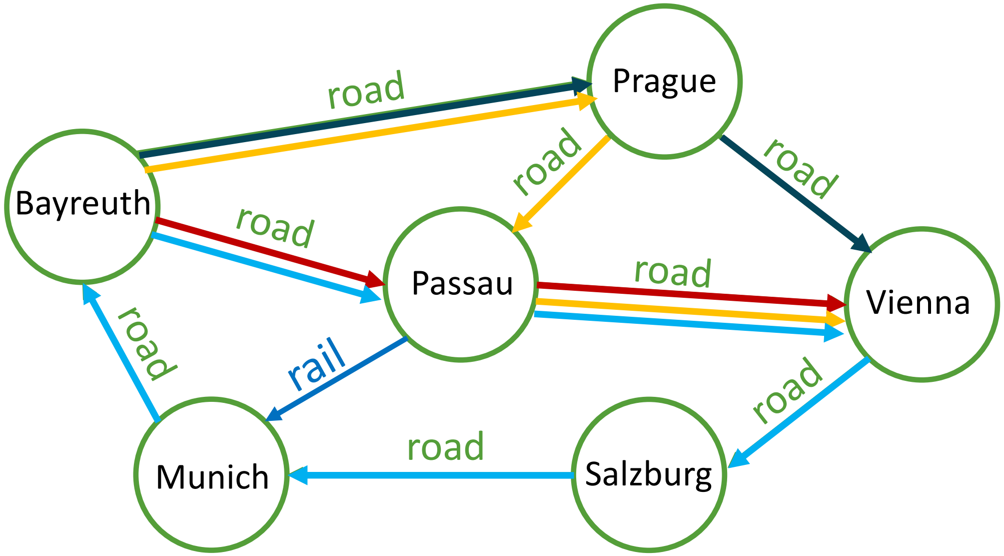


How many paths from Bayreuth to Vienna match the regular path query **(road)*** ?

(How many paths from Bayreuth to Vienna only use road-edges?)

[Theoreticians]:	∞	all paths
[SPARQL 2018]:	1	is there at least one path?
[SPARQL 2012]:	3	paths without node repetition
[Cypher]:	5	paths without edge repetition

Warm Up Question

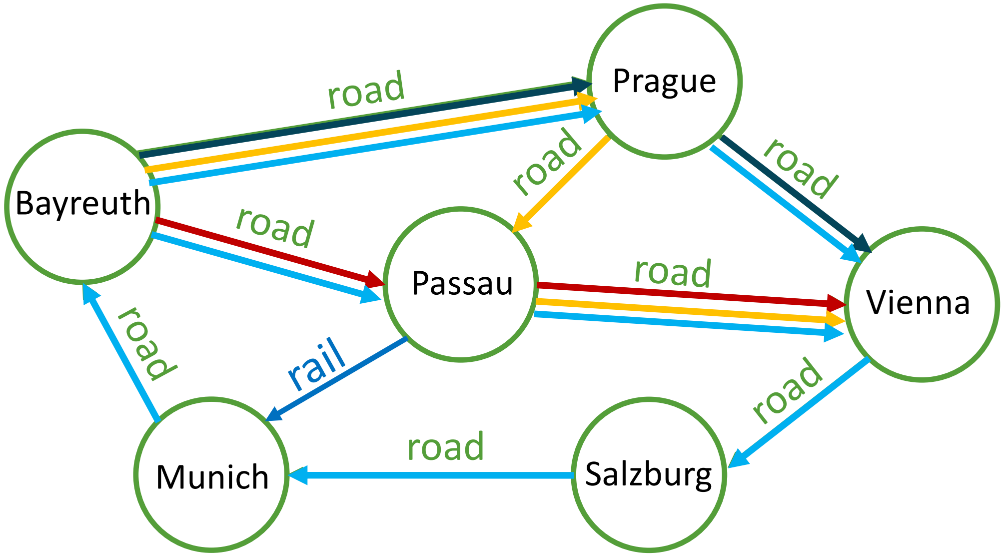


How many paths from Bayreuth to Vienna match the regular path query **(road)*** ?

(How many paths from Bayreuth to Vienna only use road-edges?)

[Theoreticians]:	∞	all paths
[SPARQL 2018]:	1	is there at least one path?
[SPARQL 2012]:	3	paths without node repetition
[Cypher]:	5	paths without edge repetition

Warm Up Question



How many paths from Bayreuth to Vienna match the regular path query **(road)*** ?

(How many paths from Bayreuth to Vienna only use road-edges?)

[Theoreticians]:	∞	all paths
[SPARQL 2018]:	1	is there at least one path?
[SPARQL 2012]:	3	paths without node repetition
[Cypher]:	5	paths without edge repetition

The Point

There are different ways of matching paths in graphs

and any of them can make sense

The Point

There are different ways of matching paths in graphs

and any of them can make sense

But which variant do you want to use in a system?

Theory

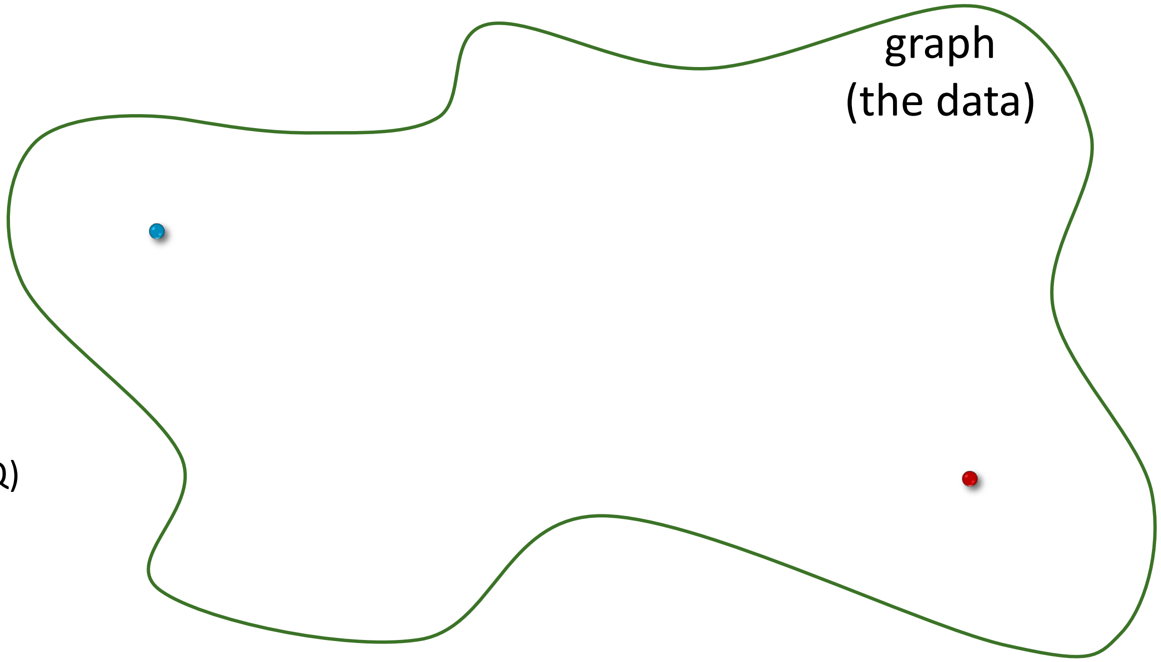
ON QUERYING PATHS IN GRAPH DATABASES

Computational Problems

Input

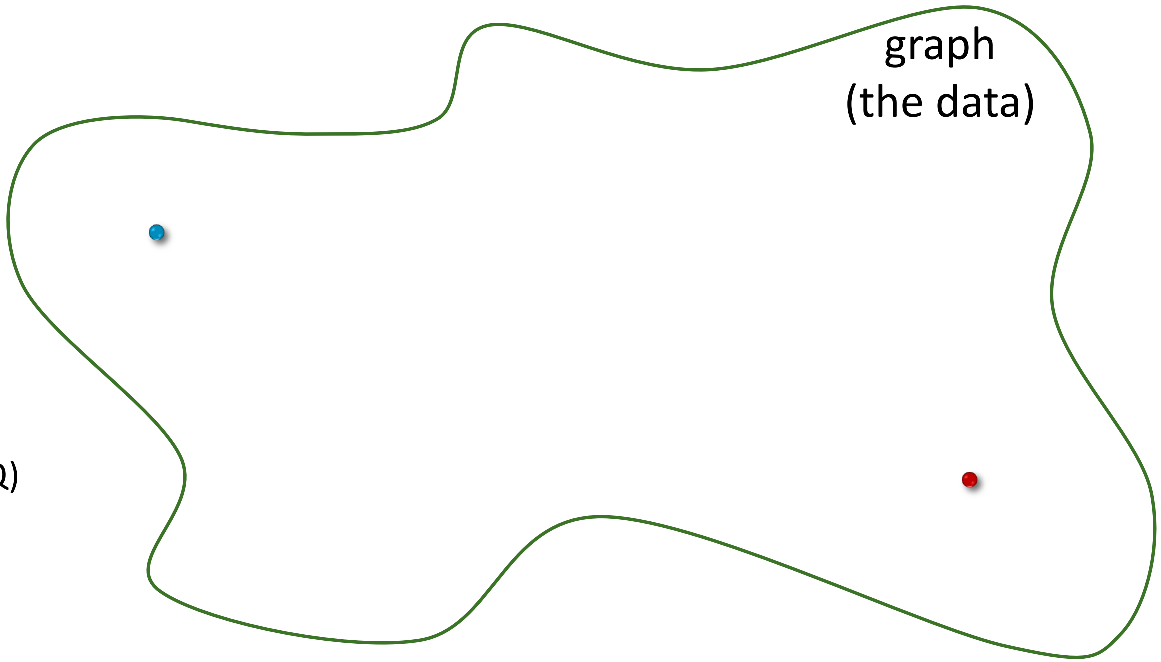
Regular expression r

called regular path query (RPQ)



Computational Problems

Input



Regular expression r

called regular path query (RPQ)

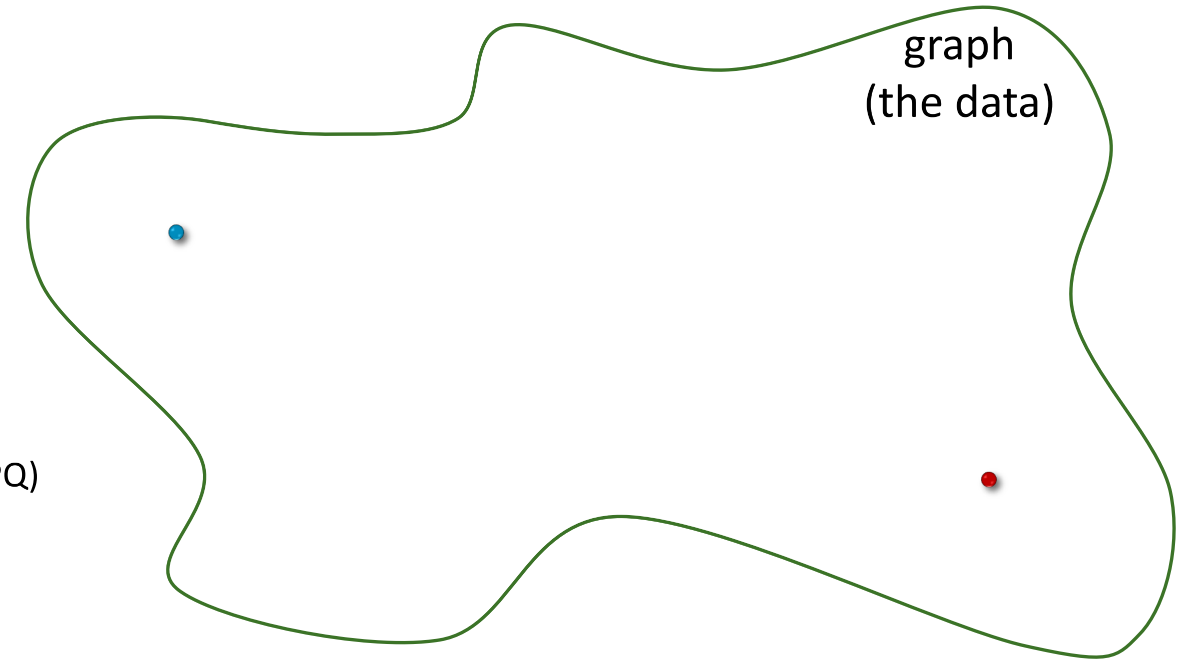
Problem

Path existence

Is there a path from [blue node] to [red node] that matches r?

Computational Problems

Input



Regular expression r

called regular path query (RPQ)

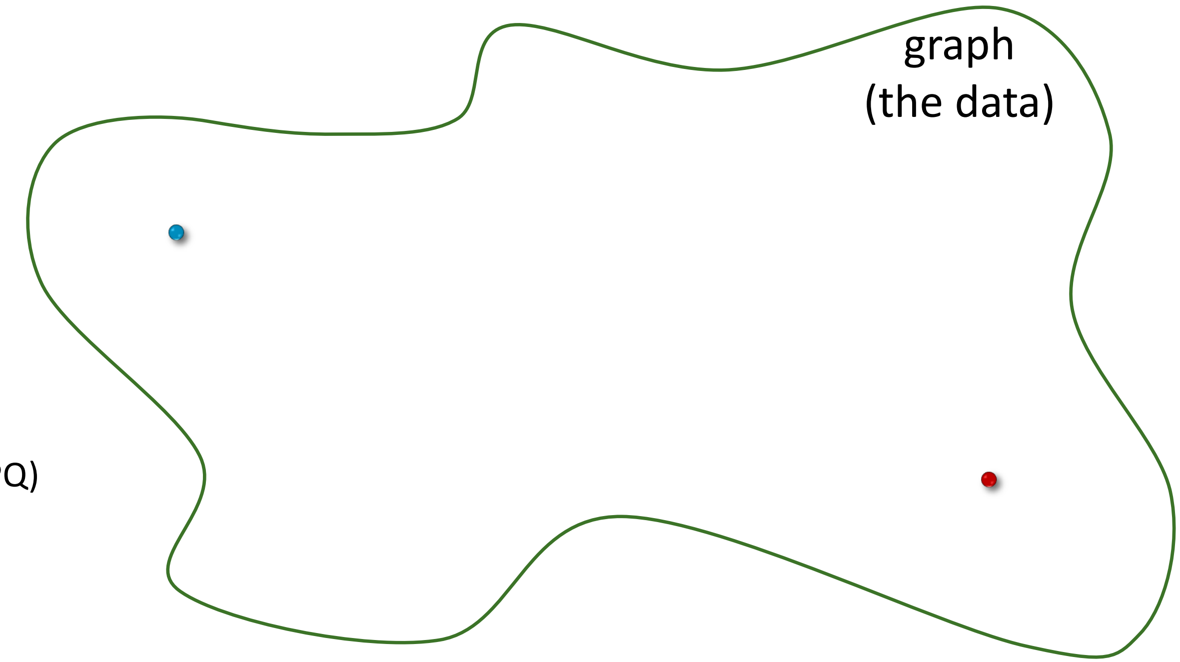
Problem

Path counting

How many paths from ● to ● match r?

Computational Problems

Input



Regular expression r
called regular path query (RPQ)

Problem

Path enumeration

Enumerate the paths from • to • that match r

Considering Different Paths

Arbitrary paths

Boolean paths

Paths without node repetitions

Paths without edge repetitions

Considering Different Paths

Arbitrary paths

Boolean paths

Paths without node repetitions

Paths without edge repetitions

Considering Different Paths

Arbitrary paths

Paths without node repetitions

Considering Different Paths

Arbitrary paths

Simple paths

Complexity of RPQ Evaluation

	Existence	Counting	Enumeration
Arbitrary paths			
Simple paths			

in P	**in FP**	**polynomial delay**
NP-hard	**#P-hard**	**too much delay**

Complexity of RPQ Evaluation

	Existence	Counting	Enumeration
Arbitrary paths			
Simple paths			

"user happy":	**in P**	**in FP**	**polynomial delay**
"user unhappy":	**NP-hard**	**#P-hard**	**too much delay**

Complexity of RPQ Evaluation

	Existence	Counting	Enumeration
Arbitrary paths	✓	✗	✓
Simple paths	✗	✗	✗

"user happy":	**in P**	**in FP**	**polynomial delay**
"user unhappy":	**NP-hard**	**#P-hard**	**too much delay**

Complexity of RPQ Evaluation

	Existence	Counting	Enumeration
Arbitrary paths	✓	✗	✓
Simple paths	✗	✗	✗

similar to
counting words in language of regular expression
#P-complete [Kannan et al., SODA 1995]

Complexity of RPQ Evaluation

	Existence	Counting	Enumeration
Arbitrary paths	✓	✗	✓
Simple paths	✗	✗	✗

Is there a simple path matching a^*ba^*?

NP-complete [Mendelzon, Wood, SICOMP 1995]

essentially because „simple path via a node" is NP-hard [Fortune et al., TCS 1980]

Is there a simple path matching $(aa)^*$?

NP-complete [Lapaugh, Papadimitriou, Networks 1984]

Complexity of RPQ Evaluation

	Existence	Counting	Enumeration
Arbitrary paths	✓	✗	✓
Simple paths	✗	✗	✗

Is there a simple path matching a^*ba^*?

NP-complete [Mendelzon, Wood, SICOMP 1995]

essentially because „simple path via a node“ is NP-hard [Fortune et al., TCS 1980]

Is there a simple path matching $(aa)^*$?

NP-complete [Lapaugh, Papadimitriou, Networks 1984]

[Bagan, Bonifati, Groz PODS 2013]

Dichotomy for which expressions

the data complexity of this problem is **in P** or **NP-complete**

Theory VS Systems

Theory:

Systems:

Theory VS Systems

Theory: „Simple paths are computationally difficult, even for very small RPQs“

Systems:

Theory VS Systems

Theory: „Simple paths are computationally difficult,
even for very small RPQs“

Systems: „But we use simple paths and we‘re fine“

What is going on with these simple paths?

RPQs in SPARQL Query Logs

[Bonifati, M., Timm, PVLDB 2017]

Extracted 247,404 RPQs from SPARQL query logs (2009 - 2017)
(from DBPedia, biological databases, British museum, Wikidata, …)

RPQs in SPARQL Query Logs

[Bonifati, M., Timm, PVLDB 2017]

Extracted 247,404 RPQs from SPARQL query logs (2009 - 2017)
(from DBPedia, biological databases, British museum, Wikidata, …)

→ **Only very few different kinds of RPQs (± 17)**

RPQs in SPARQL Query Logs

Expression Type	Relative	Expression Type	Relative
A^*	48.76%	$a^*b?$	<0.01%
A	32.10%	abc^*	<0.01%
$a_1 \cdots a_k$	8.66%	$A_1 \cdots A_k$	<0.01%
a^*b	7.73%	$(ab^*) + c$	<0.01%
A^+	1.54%	$a^* + b$	<0.01%
$a_1? \cdots a_k?$	1.15%	$a + b^+$	<0.01%
$aA?$	0.01%	$a^+ + b^+$	<0.01%
$a_1 a_2? \cdots a_k?$	0.01%	$(ab)^*$	<0.01%
$A?$	<0.01%		

$k \leq 6$

Disjunction of symbols: $A, A_1, \ldots$

Single symbols: $a, b, c, a_1, \ldots$

Data from [Bonifati et al., PVLDB 2017]

Simple Transitive Expressions

Atomic Expression

disjunction $(\boldsymbol{a_1} + \cdots + \boldsymbol{a_n})$ of symbols (denote this by $\boldsymbol{A}$, $\boldsymbol{A_i}$,...)

Simple Transitive Expressions

Atomic Expression

disjunction $(a_1 + \cdots + a_n)$ of symbols (denote this by A, A_i,...)

Local Expression

$A_1 \cdots A_k$ or $A_1? \cdots A_k?$

„follow a path of length k"

„follow a path of length at most k"

Simple Transitive Expressions

Atomic Expression

disjunction $(a_1 + \cdots + a_n)$ of symbols (denote this by A, A_i,...)

Local Expression

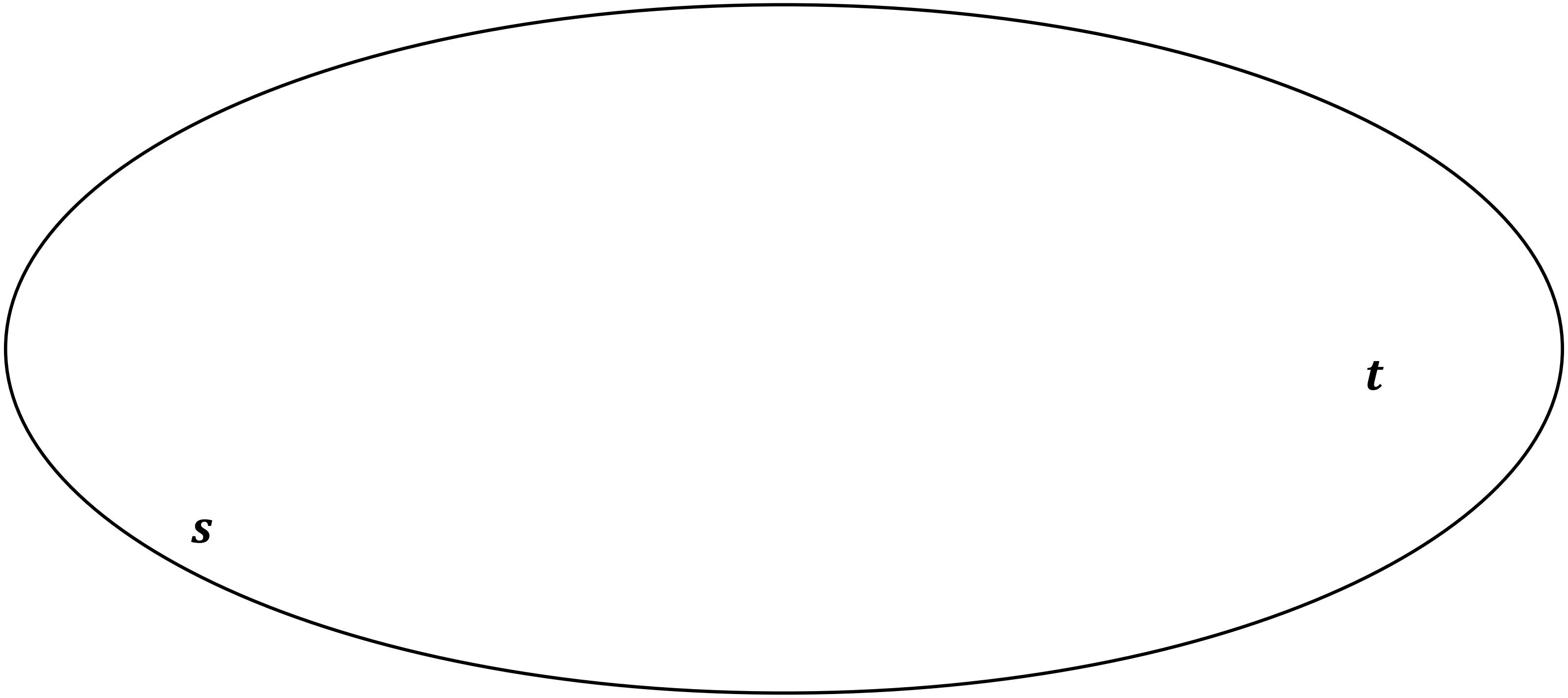
$A_1 \cdots A_k$ or $A_1? \cdots A_k?$

„follow a path of length k“ „follow a path of length at most k“

Simple Transitive Expression (STE)

$L_1\ A^* L_2$ where L_1, L_2 are local expressions

Simple Transitive Expressions

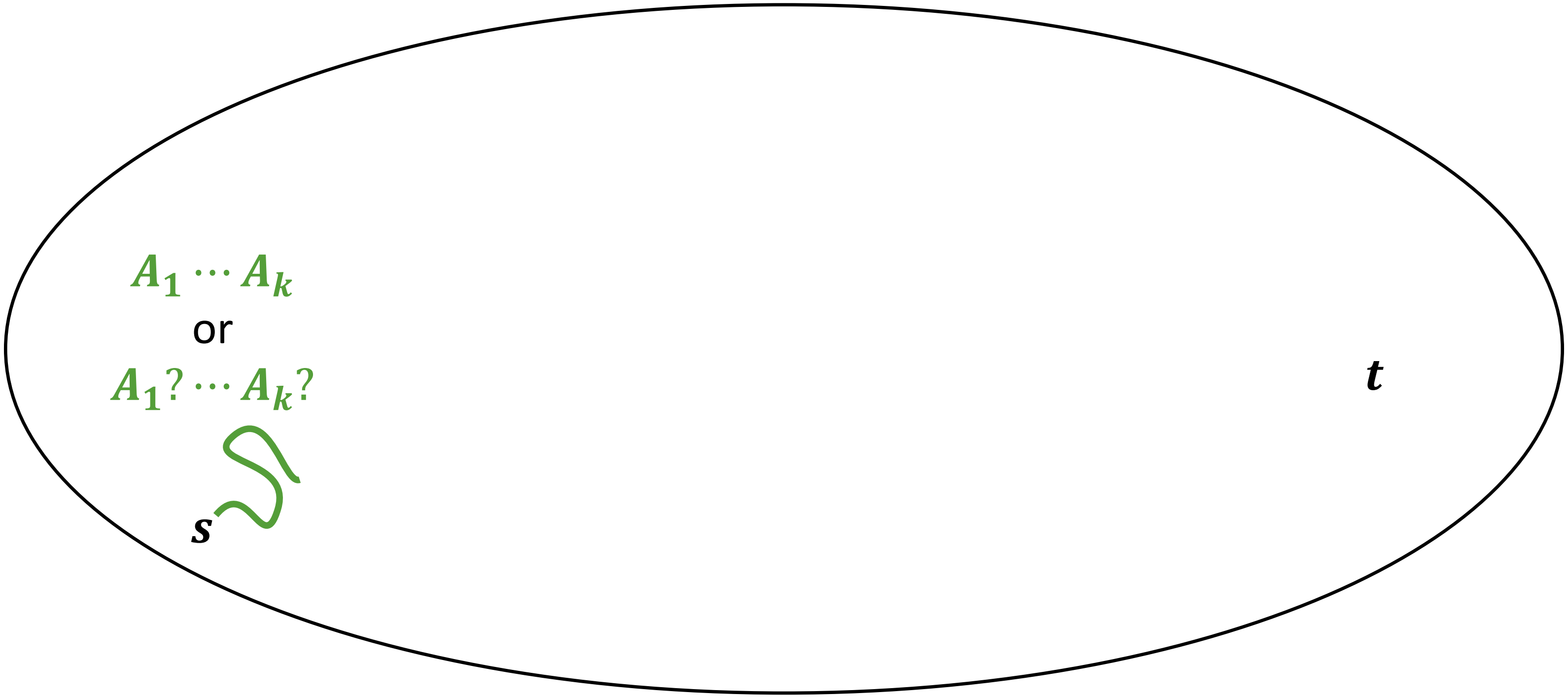


Simple Transitive Expression (STE)

$L_1\ A^* L_2$ where L_1, L_2 are local expressions

(we allow $A = \emptyset$)

Simple Transitive Expressions

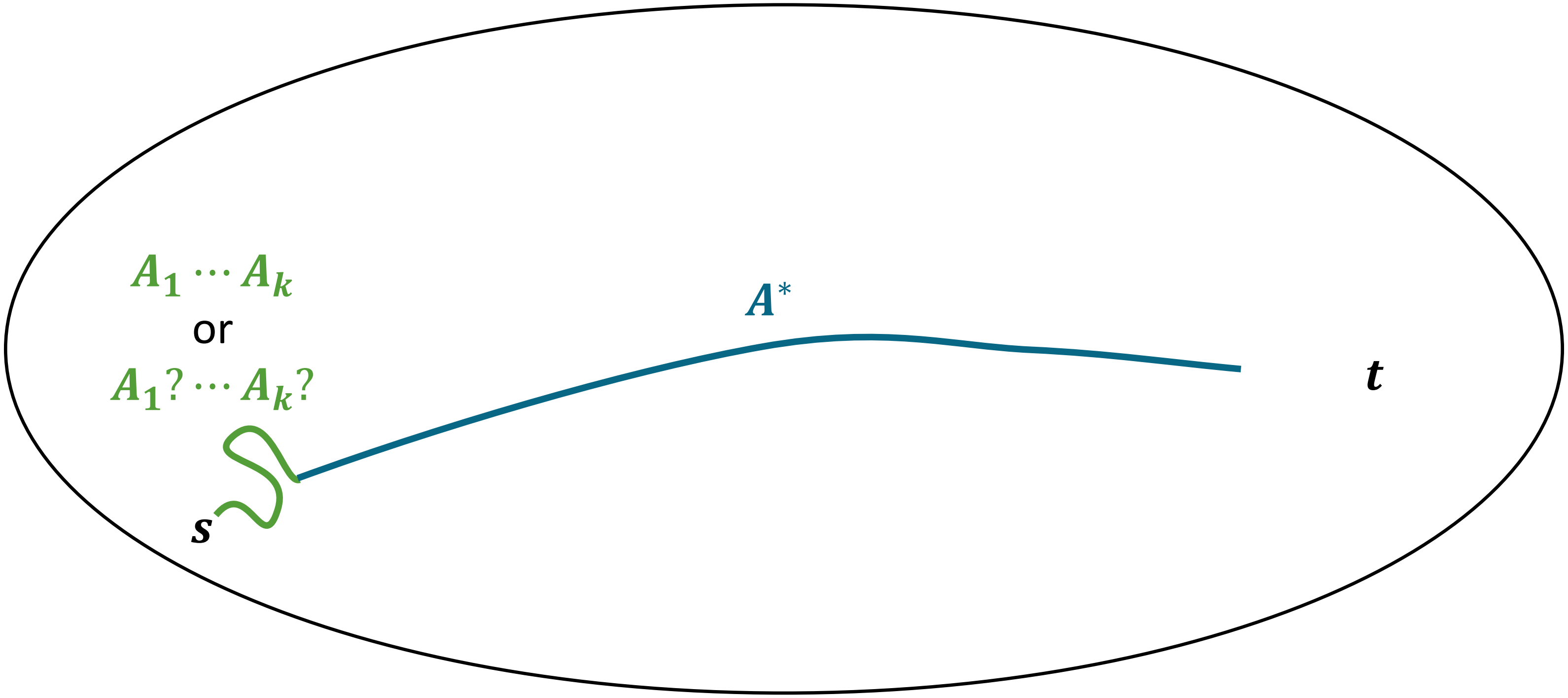


Simple Transitive Expression (STE)

$L_1\ A^* L_2$ where L_1, L_2 are local expressions

(we allow $A = \emptyset$)

Simple Transitive Expressions

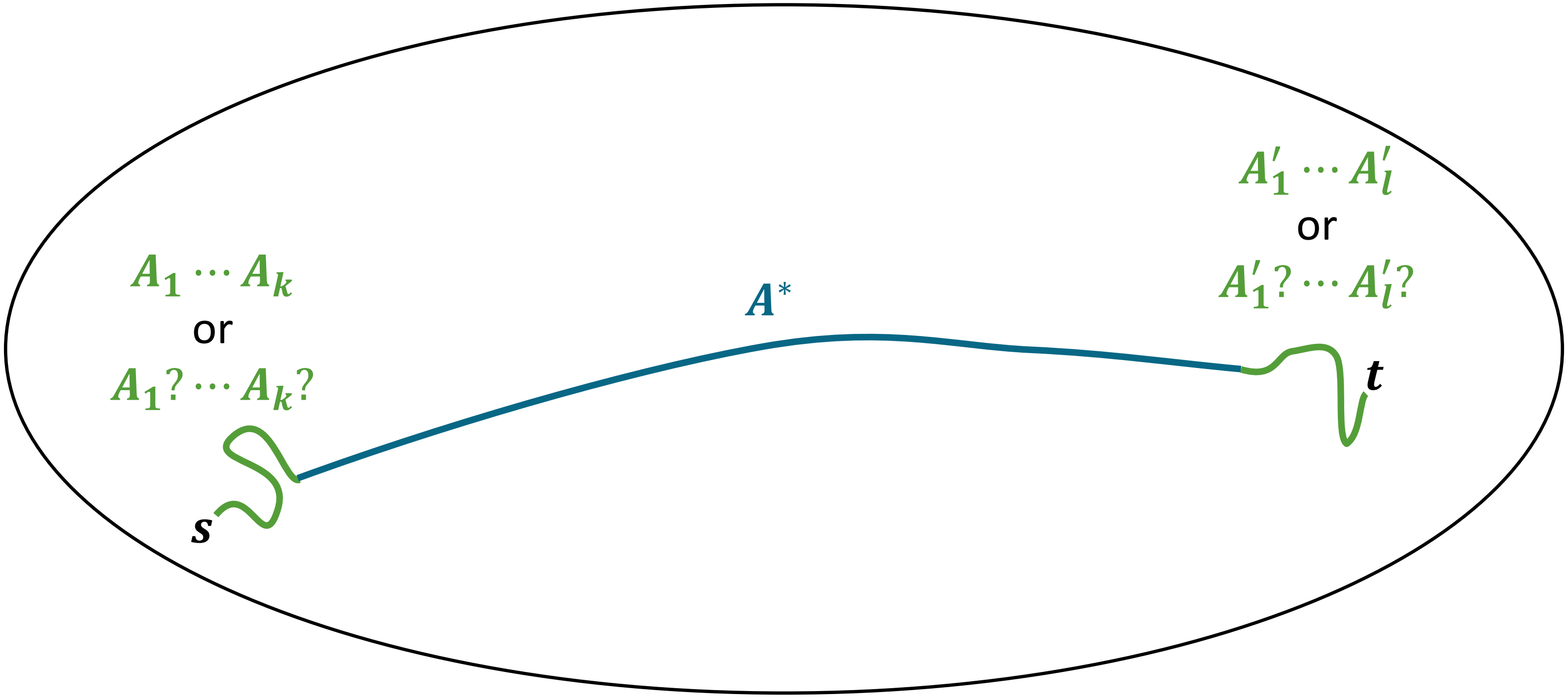


Simple Transitive Expression (STE)

$L_1\ A^* L_2$ where L_1, L_2 are local expressions

(we allow $A = \emptyset$)

Simple Transitive Expressions



Simple Transitive Expression (STE)

$L_1\ A^* L_2$ where L_1, L_2 are local expressions

(we allow $A = \emptyset$)

RPQs in SPARQL Query Logs

Expression Type	Relative	Expression Type	Relative
A^*	48.76%	$a^*b?$	<0.01%
A	32.10%	abc^*	<0.01%
$a_1 \cdots a_k$	8.66%	$A_1 \cdots A_k$	<0.01%
a^*b	7.73%	$(ab^*) + c$	<0.01%
A^+	1.54%	$a^* + b$	<0.01%
$a_1? \cdots a_k?$	1.15%	$a + b^+$	<0.01%
$aA?$	0.01%	$a^+ + b^+$	<0.01%
$a_1 a_2? \cdots a_k?$	0.01%	$(ab)^*$	<0.01%
$A?$	<0.01%		

$k \leq 6$

STE

Union of STEs

something else

Data from [Bonifati et al., PVLDB 2017]

RPQs in SPARQL Query Logs

Expression Type	Relative	Expression Type	Relative
A^*	48.76%	$a^*b?$	<0.01%
A	32.10%	abc^*	[illegible]
$a_1 \cdots a_k$	8.66%	[illegible]	[illegible]
a^*b	7.73%	[illegible]	[illegible]
A^+	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	<0.01%
[illegible]	[illegible]	$(ab)^*$	<0.01%
[illegible]	[illegible]		

$k \le$ [illegible]

[illegible]mething else

99.99% are STEs!

Data from [Bonifati et al., PVLDB 2017]

Main Theorem Warm-Up

Simple path existence

Given graph $\boldsymbol{G}$, nodes $\boldsymbol{s}$ and $\boldsymbol{t}$, and RPQ $\boldsymbol{r}$,
is there a simple path from $\boldsymbol{s}$ to $\boldsymbol{t}$ that matches $\boldsymbol{r}$?

Main Theorem Warm-Up

Simple path existence for $\boldsymbol{R}$

Given graph $\boldsymbol{G}$, nodes $\boldsymbol{s}$ and $\boldsymbol{t}$, and RPQ $\boldsymbol{r} \in \boldsymbol{R}$,
is there a simple path from $\boldsymbol{s}$ to $\boldsymbol{t}$ that matches $\boldsymbol{r}$?

Example classes $\boldsymbol{R}$:

Main Theorem Warm-Up

Simple path existence for $\boldsymbol{R}$

Given graph $\boldsymbol{G}$, nodes $\boldsymbol{s}$ and $\boldsymbol{t}$, and RPQ $\boldsymbol{r} \in \boldsymbol{R}$,
is there a simple path from $\boldsymbol{s}$ to $\boldsymbol{t}$ that matches $\boldsymbol{r}$?

Example classes $\boldsymbol{R}$: $\underbrace{\boldsymbol{aa} \dots \boldsymbol{a}}_{\boldsymbol{k}}$ for $\boldsymbol{k} \in \mathbb{N}$ denote this by $\{\boldsymbol{a}^{\boldsymbol{k}} \mid \boldsymbol{k} \in \mathbb{N}\}$

Main Theorem Warm-Up

Simple path existence for $\boldsymbol{R}$

Given graph $\boldsymbol{G}$, nodes $\boldsymbol{s}$ and $\boldsymbol{t}$, and RPQ $\boldsymbol{r} \in \boldsymbol{R}$,
is there a simple path from $\boldsymbol{s}$ to $\boldsymbol{t}$ that matches $\boldsymbol{r}$?

Example classes $\boldsymbol{R}$:

$\underbrace{\boldsymbol{aa \ldots a}}_{\boldsymbol{k}}$ for $\boldsymbol{k} \in \mathbb{N}$ denote this by $\{\boldsymbol{a^k} \mid \boldsymbol{k} \in \mathbb{N}\}$

$\overbrace{\boldsymbol{aa \ldots a}}^{\boldsymbol{k}}\boldsymbol{a}^*$ for $\boldsymbol{k} \in \mathbb{N}$ denote this by $\{\boldsymbol{a^k a^*} \mid \boldsymbol{k} \in \mathbb{N}\}$

Main Theorem Warm-Up

Simple path existence for $\boldsymbol{R}$

Given graph $\boldsymbol{G}$, nodes $\boldsymbol{s}$ and $\boldsymbol{t}$, and RPQ $\boldsymbol{r} \in \boldsymbol{R}$,
is there a simple path from $\boldsymbol{s}$ to $\boldsymbol{t}$ that matches $\boldsymbol{r}$?

Example classes $\boldsymbol{R}$:

$\underbrace{\boldsymbol{aa \dots a}}_{\boldsymbol{k}}$ for $\boldsymbol{k} \in \mathbb{N}$ denote this by $\{\boldsymbol{a^k} \mid \boldsymbol{k} \in \mathbb{N}\}$

$\overbrace{\boldsymbol{aa \dots a}}^{\boldsymbol{k}}\boldsymbol{a}^*$ for $\boldsymbol{k} \in \mathbb{N}$ denote this by $\{\boldsymbol{a^k a^*} \mid \boldsymbol{k} \in \mathbb{N}\}$

These are non-trivial problems!

Main Theorem Warm-Up

Simple path existence for $\boldsymbol{R}$

Given graph $\boldsymbol{G}$, nodes $\boldsymbol{s}$ and $\boldsymbol{t}$, and RPQ $\boldsymbol{r} \in \boldsymbol{R}$,
is there a simple path from $\boldsymbol{s}$ to $\boldsymbol{t}$ that matches $\boldsymbol{r}$?

Example classes $\boldsymbol{R}$:

$\underbrace{\boldsymbol{aa \ldots a}}_{\boldsymbol{k}}$ for $\boldsymbol{k} \in \mathbb{N}$ denote this by $\{\boldsymbol{a^k} \mid \boldsymbol{k} \in \mathbb{N}\}$

$\overbrace{\boldsymbol{aa \ldots a}}^{\boldsymbol{k}}\boldsymbol{a}^*$ for $\boldsymbol{k} \in \mathbb{N}$ denote this by $\{\boldsymbol{a^k a^*} \mid \boldsymbol{k} \in \mathbb{N}\}$

Theorem [Alon, Yuster, Zwick, JACM 1995]

“Simple path existence for $\{\boldsymbol{a^k} \mid \boldsymbol{k} \in \mathbb{N}\}$ is in FPT“

Color coding technique

Main Theorem Warm-Up

Simple path existence for $\boldsymbol{R}$

Given graph $\boldsymbol{G}$, nodes $\boldsymbol{s}$ and $\boldsymbol{t}$, and RPQ $\boldsymbol{r} \in \boldsymbol{R}$, is there a simple path from $\boldsymbol{s}$ to $\boldsymbol{t}$ that matches $\boldsymbol{r}$?

Example classes $\boldsymbol{R}$:

$\underbrace{aa \ldots a}_{k}$ for $k \in \mathbb{N}$ — denote this by $\{a^k \mid k \in \mathbb{N}\}$

$\overbrace{aa \ldots a}^{k}a^*$ for $k \in \mathbb{N}$ — denote this by $\{a^k a^* \mid k \in \mathbb{N}\}$

Theorem [Alon, Yuster, Zwick, JACM 1995]

"Simple path existence for $\{a^k \mid k \in \mathbb{N}\}$ is in FPT"

Color coding technique

Theorem [Technique from Fomin et al., JACM 2016 communicated to us by Holger Dell]

"Simple path existence for $\{a^k a^* \mid k \in \mathbb{N}\}$ is in FPT"

Representative sets technique

Main Theorem

Simple path existence for $\boldsymbol{R}$

Given graph $\boldsymbol{G}$, nodes $\boldsymbol{s}$ and $\boldsymbol{t}$, and RPQ $\boldsymbol{r} \in \boldsymbol{R}$,
is there a simple path from $\boldsymbol{s}$ to $\boldsymbol{t}$ that matches $\boldsymbol{r}$?

Main Theorem

Let $\boldsymbol{R}$ be a class(*) of STEs:
if $\boldsymbol{R}$ is ***cuttable***, then simple path existence for $\boldsymbol{R}$ is in **FPT**
otherwise, simple path existence for $\boldsymbol{R}$ is **W[1]-hard**.

(*) satisfying a mild condition, needed for W[1] hardness

Main Theorem

Simple path existence for $\boldsymbol{R}$

Given graph $\boldsymbol{G}$, nodes $\boldsymbol{s}$ and $\boldsymbol{t}$, and RPQ $\boldsymbol{r} \in \boldsymbol{R}$,
is there a simple path from $\boldsymbol{s}$ to $\boldsymbol{t}$ that matches $\boldsymbol{r}$?

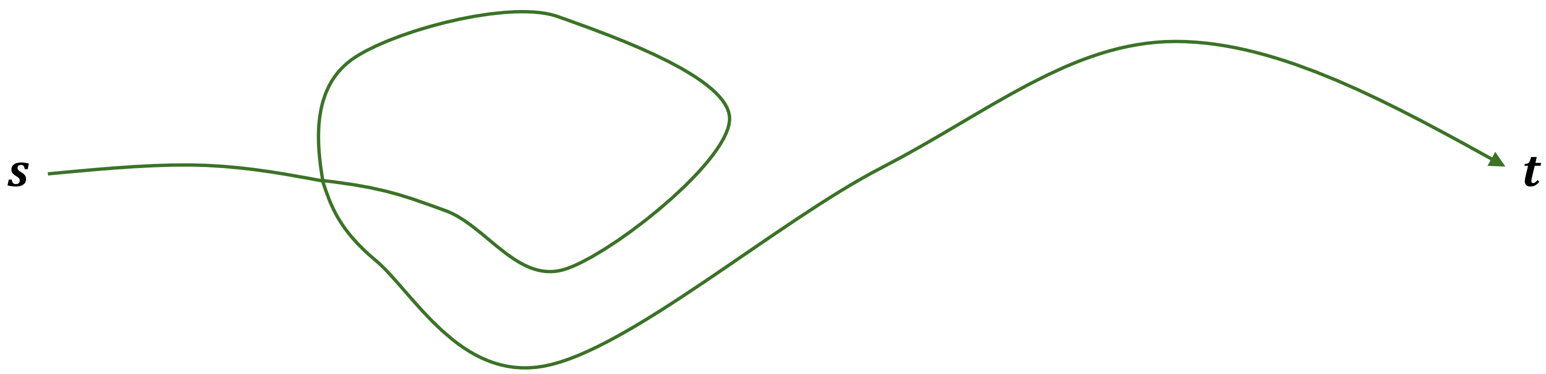
parameter: size of RPQ

Main Theorem

Let $\boldsymbol{R}$ be a class(*) of STEs:
if $\boldsymbol{R}$ is ***cuttable***, then simple path existence for $\boldsymbol{R}$ is in **FPT**
otherwise, simple path existence for $\boldsymbol{R}$ is **W[1]-hard**.

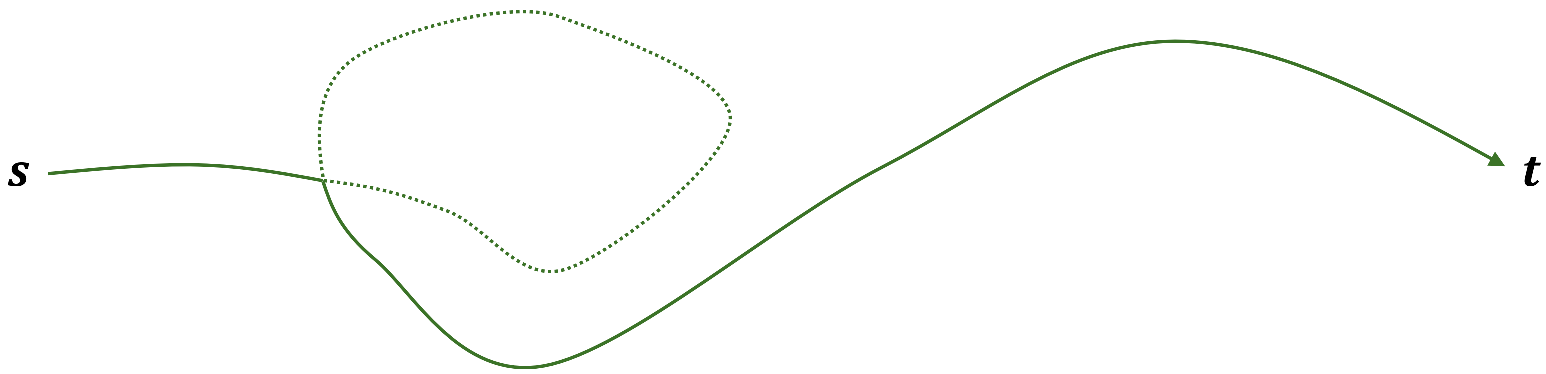
(*) satisfying a mild condition, needed for W[1] hardness

Intuition behind Cuttability



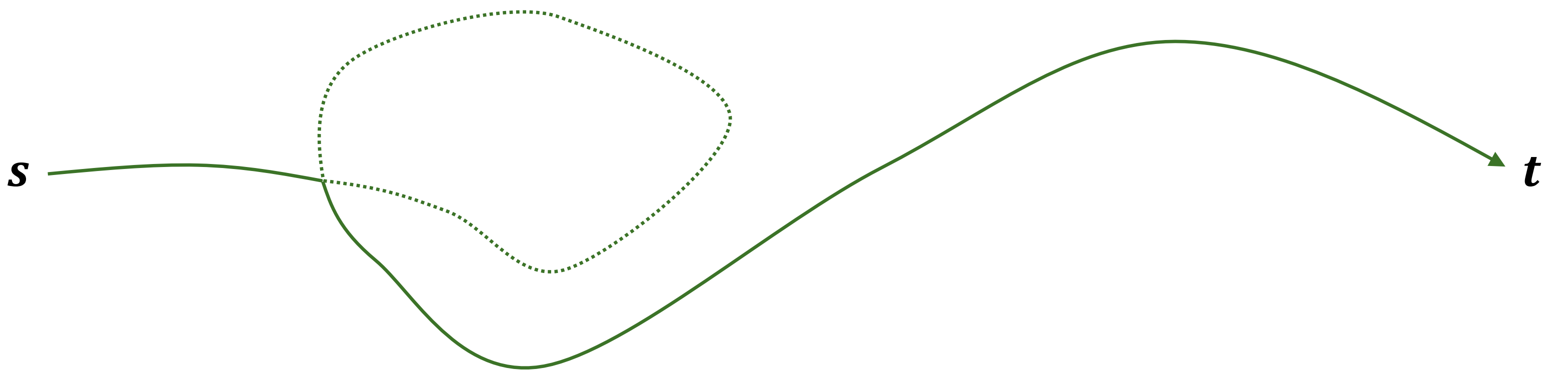
Path that matches $\boldsymbol{r}$

Intuition behind Cuttability



Simple Path that matches $\boldsymbol{r}$?

Intuition behind Cuttability

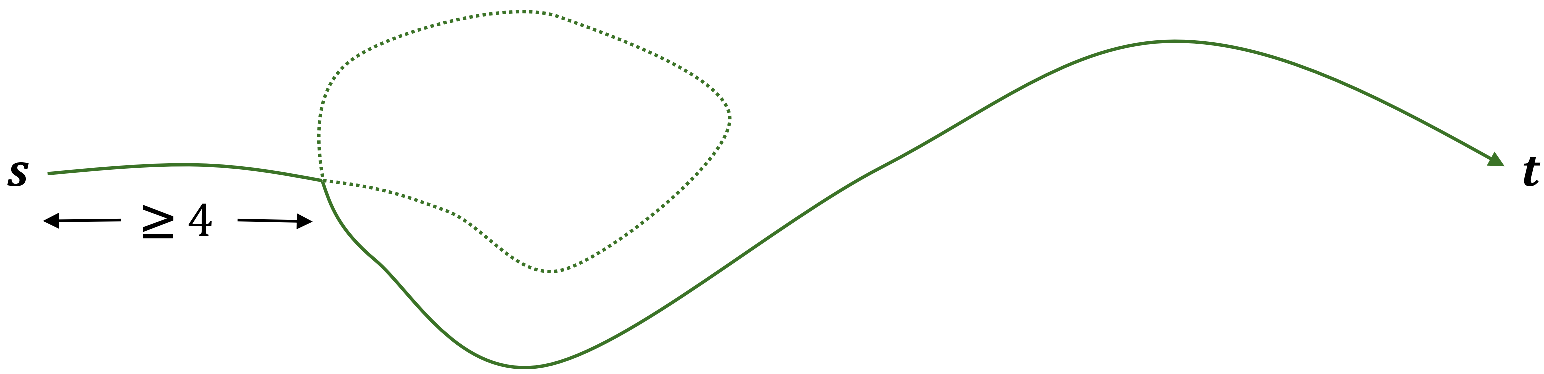


Simple Path that matches $\boldsymbol{r}$?

Does the simple path still match $\boldsymbol{r}$?

- "Easy" to check for $\boldsymbol{aaaaa}^*$ (check length)
- "Hard" to check for $\boldsymbol{bbbba}^*$ (check length+label)

Intuition behind Cuttability

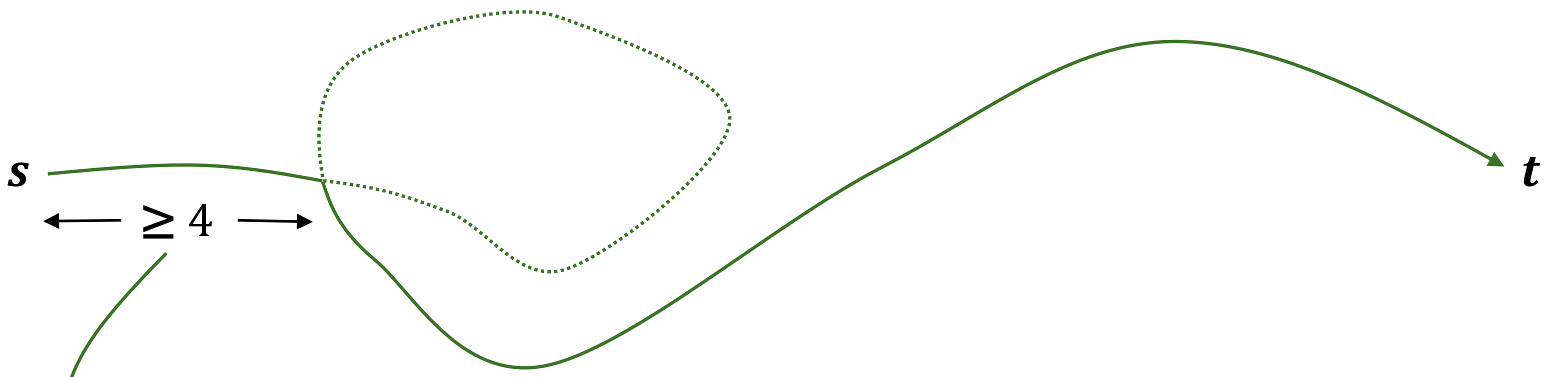


Simple Path that matches $\boldsymbol{r}$?

Does the simple path still match $\boldsymbol{r}$?

- "Easy" to check for $\boldsymbol{aaaaa}^*$ (check length)
- "Hard" to check for $\boldsymbol{bbbba}^*$ (check length+label)

Intuition behind Cuttability



Simple Path that matches $\boldsymbol{r}$?

Does the simple path still match $\boldsymbol{r}$?

- "Easy" to check for $\boldsymbol{aaaaa}^*$ (check length)
- "Hard" to check for $\boldsymbol{bbbba}^*$ (check length+label)

cut border for $\boldsymbol{bbbba}^*$

Formalizing this Idea

Formalizing this Idea

Consider STE $\boldsymbol{r} = \boldsymbol{A_1} \cdots \boldsymbol{A_k}\ \boldsymbol{A}^*$

Its **cut border** ℓ is the largest number such that $\boldsymbol{A} \not\subseteq \boldsymbol{A}_\ell$

(and $\ell = 0$ if no such $\boldsymbol{A}_\ell$ exists)

Formalizing this Idea

Consider STE $\boldsymbol{r = A_1 \cdots A_k \ A^*}$

Its **cut border** ℓ is the largest number such that $\boldsymbol{A} \not\subseteq \boldsymbol{A_\ell}$
(and $\ell = 0$ if no such $\boldsymbol{A_\ell}$ exists)

Examples

- $\boldsymbol{aaaa^*}$ $\ell = 0$ because $\{\boldsymbol{a}\} \subseteq \{\boldsymbol{a}\}$
- $\boldsymbol{aaba^*}$ $\ell = 3$ because $\{\boldsymbol{a}\} \not\subseteq \{\boldsymbol{b}\}$
- $(\boldsymbol{a} + \boldsymbol{c})\boldsymbol{ab}(\boldsymbol{a} + \boldsymbol{b})^*$ $\ell = 3$ because $\{\boldsymbol{a}, \boldsymbol{b}\} \not\subseteq \{\boldsymbol{b}\}$

Formalizing this Idea

Consider STE $r = A_1 \cdots A_k \ A^*$

Its **cut border** ℓ is the largest number such that $A \not\subseteq A_\ell$
(and $\ell = 0$ if no such A_ℓ exists)

Examples

- $aaaa^*$ — $\ell = 0$ — because $\{a\} \subseteq \{a\}$
- $aaba^*$ — $\ell = 3$ — because $\{a\} \not\subseteq \{b\}$
- $(a+c)ab(a+b)^*$ — $\ell = 3$ — because $\{a, b\} \not\subseteq \{b\}$

Formalizing this Idea

Consider STE $r = A_1 \cdots A_k \ A^*$

Its **cut border** ℓ is the largest number such that $A \not\subseteq A_\ell$
(and $\ell = 0$ if no such A_ℓ exists)

Examples

- $aaaa^*$ — $\ell = 0$ — because $\{a\} \subseteq \{a\}$
- $aaba^*$ — $\ell = 3$ — because $\{a\} \not\subseteq \{b\}$
- $(a+c)ab(a+b)^*$ — $\ell = 3$ — because $\{a,b\} \not\subseteq \{b\}$

Definition

A class R of STEs is ***cuttable***, if
there is a constant c such that all its expressions have cut border $\leq c$

Formalizing this Idea

Definition

A class $\boldsymbol{R}$ of STEs is ***cuttable***, if
there is a constant $\boldsymbol{c}$ such that all its expressions have cut border $\leq \boldsymbol{c}$

parameter: size of RPQ

Main Theorem

Let $\boldsymbol{R}$ be a class$^{(*)}$ of STEs:
if $\boldsymbol{R}$ is ***cuttable***, then simple path existence for $\boldsymbol{R}$ is in **FPT**
otherwise, simple path existence for $\boldsymbol{R}$ is **W[1]-hard**.

For the **FPT** upper bound,
the complexity in the parameter is single exponential

Upper Bound Idea

Upper Bound Idea

Theorem [Alon, Yuster, Zwick, JACM 1995]

Finding **simple paths** of length **exactly k** is in FPT

Color coding technique

Theorem [Fomin et al., JACM 2016]

Finding **simple cycles** of length **at least k** is in FPT

Representative sets technique

Theorem [Technique from Fomin et al., JACM 2016 communicated to us by Holger Dell]

Finding **simple paths** of length **at least k** is in FPT

Representative sets technique

Upper Bound Idea

Find a simple path matching $\boldsymbol{A_1} \cdots \boldsymbol{A_k}\, \boldsymbol{A}^*$

$\boldsymbol{s}$ $\boldsymbol{t}$

Upper Bound Idea

Find a simple path matching $A_1 \cdots A_k \, A^*$

s $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ t

$A_1 \cdots A_c$

Brute force

Upper Bound Idea

Find a simple path matching $\boldsymbol{A_1 \cdots A_k\, A^*}$

$\boldsymbol{s}$ $\boldsymbol{t}$

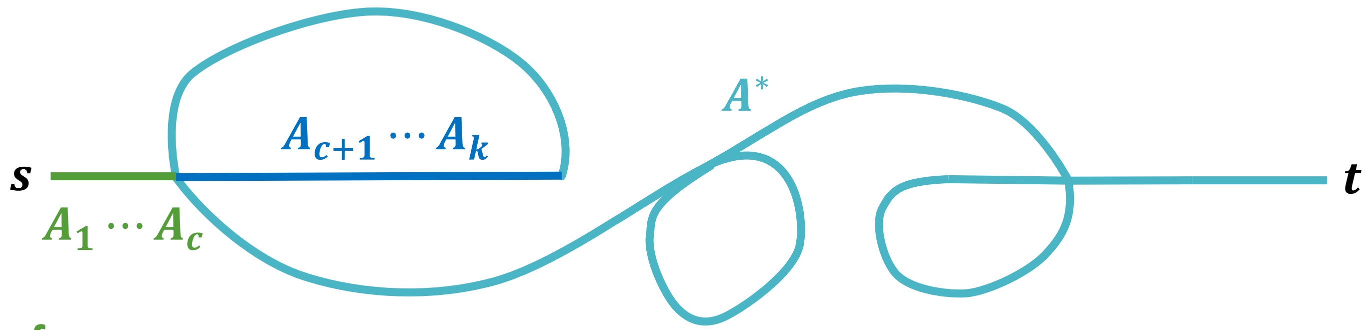
$\boldsymbol{A_1 \cdots A_c}$

Brute force

Simple Path matching $\boldsymbol{A_{c+1} \cdots A_k\, A^*}$
and avoiding the brute force part

Upper Bound Idea

Find a simple path matching $\boldsymbol{A_1 \cdots A_k\ A^*}$



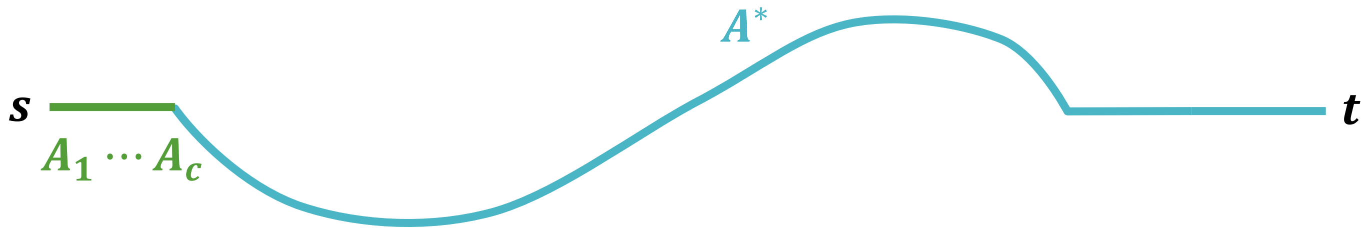
Brute force

Simple Path matching $\boldsymbol{A_{c+1} \cdots A_k}\ A^*$

and avoiding the brute force part

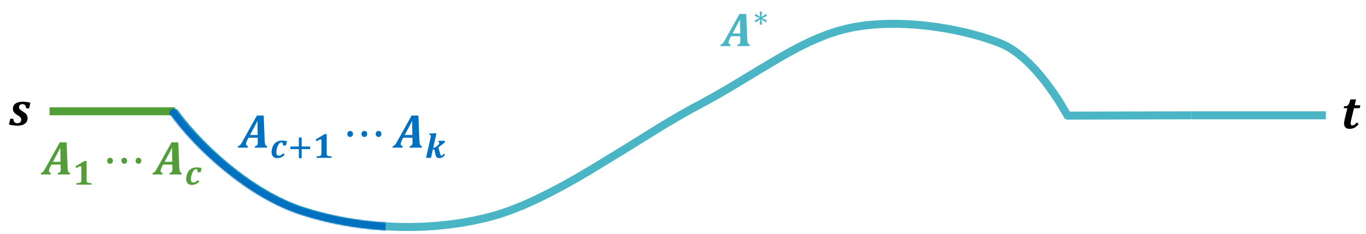
Upper Bound Idea

Find a simple path matching $A_1 \cdots A_k \, A^*$



Upper Bound Idea

Find a simple path matching $\boldsymbol{A_1 \cdots A_k\, A^*}$



Since $A \subseteq A_i$

Lower Bound Idea

Parameterized Two Disjoint Paths

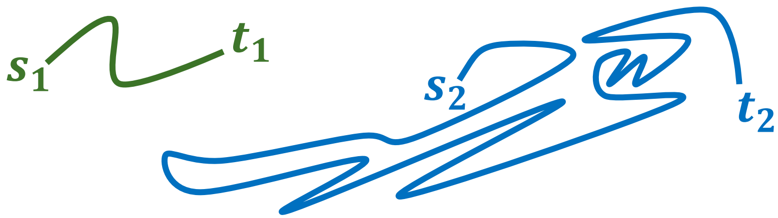
s_1 t_1 s_2 t_2

Lower Bound Idea

Parameterized Two Disjoint Paths

Given graph G,
nodes s_1 and t_1 and s_2 and t_2
and a parameter **k**

Are there node-disjoint paths
from s_1 to t_1
from s_2 to t_2



Lower Bound Idea

Parameterized Two Disjoint Paths

Given graph G,

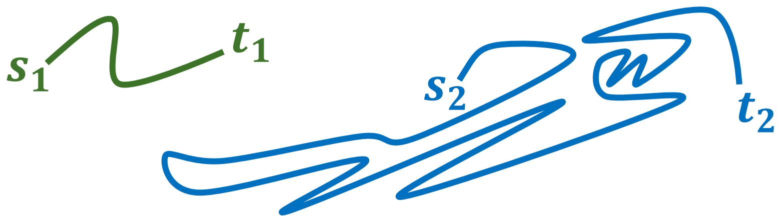
nodes s_1 and t_1 and s_2 and t_2

and a parameter **k**

Are there node-disjoint paths

from s_1 to t_1 of length at most **k**

from s_2 to t_2



Lower Bound Idea

Parameterized Two Disjoint Paths

Given graph G,

nodes s_1 and t_1 and s_2 and t_2

and a parameter **k**

Are there node-disjoint paths

from s_1 to t_1 of length at most **k**

from s_2 to t_2

Theorem (Main Technical Result)

Parameterized Two Disjoint Paths is **W[1]-hard**

Building on proofs from [Slivkins, SIDMA 10; Grohe&Grüber ICALP 07]

Lower Bound Idea

Theorem (Main Technical Result)

Parameterized Two Disjoint Paths is **W[1]-hard**

Lemma

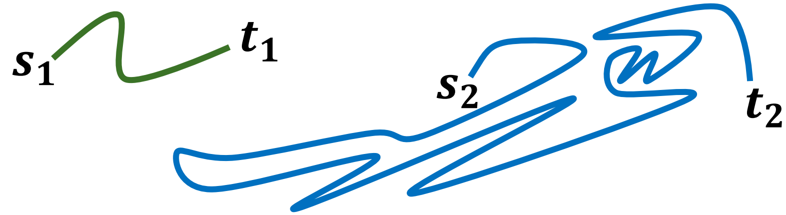
Let $\boldsymbol{R}$ be a class(*) of STEs:

if $\boldsymbol{R}$ is ***not cuttable***, then simple path existence for $\boldsymbol{R}$ is **W[1]-hard**.

Lower Bound Idea

Theorem (Main Technical Result)

Parameterized Two Disjoint Paths is **W[1]-hard**



Lemma

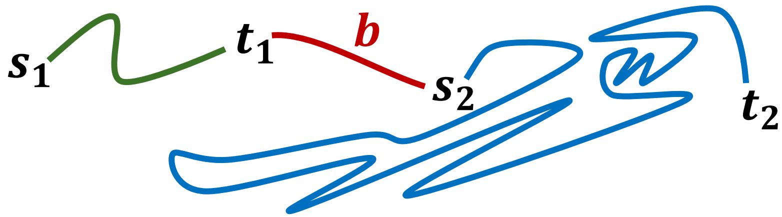
Let $\boldsymbol{R}$ be a class(*) of STEs:

if $\boldsymbol{R}$ is ***not cuttable***, then simple path existence for $\boldsymbol{R}$ is **W[1]-hard**.

Lower Bound Idea

Theorem (Main Technical Result)

Parameterized Two Disjoint Paths is **W[1]-hard**



Warning: drastic oversimplification

Lemma

Let $\boldsymbol{R}$ be a class(*) of STEs:

if $\boldsymbol{R}$ is ***not cuttable***, then simple path existence for $\boldsymbol{R}$ is **W[1]-hard**.

Extensions

Extensions

Main Theorem

Let $\boldsymbol{R}$ be a class(*) of STEs:

if $\boldsymbol{R}$ is ***cuttable***, then simple path existence for $\boldsymbol{R}$ is in **FPT**
otherwise, simple path existence for $\boldsymbol{R}$ is **W[1]-hard**.

The main result extends to:

- Enumeration problems
  FPT time becomes FPT delay — using [Yen 1971]
- Edge-disjoint paths
  But the dichotomy slightly changes — [ArXiv 2017]

Taking a Step Back

WHAT DID WE LEARN HERE?

So what does all this mean?

Expression Type	Relative	Expression Type	Relative
A^*	48.76%	$a^*b?$	<0.01%
A	32.10%	abc^*	<0.01%
$a_1 \cdots a_k$	8.66%	$A_1 \cdots A_k$	<0.01%
a^*b	7.73%	$(ab^*) + c$	<0.01%
A^+	1.54%	$a^* + b$	<0.01%
$a_1? \cdots a_k?$	1.15%	$a + b^+$	<0.01%
$aA?$	0.01%	$a^+ + b^+$	<0.01%
$a_1 a_2? \cdots a_k?$	0.01%	$(ab)^*$	<0.01%
$A?$	<0.01%		

$k \leq 6$

Cuttable STEs ($\ell \leq 2$) Thus in FPT

Union of STEs

something else

- These expressions have **cut border** $\leq$ **2**
- The FPT algorithms have parameter $\boldsymbol{k} \leq$ **6**

- **But even naive algorithms are expected to work reasonably well (brute-force checks for paths of lengh 2 and simple paths of length 6)**

Take Home Messages

- Looking in query logs can pay off and inspire new research questions!
- 99.99% of RPQs found in a practical study are
  Simple Transitive Expressions (STEs)
- Dichotomy for simple path evaluation of STEs
  - Another one for no-repeated-edge semantics is similar
- If “cut borders are bounded”, evaluation of STEs is FPT
  - Cut borders in the real data are at most 2
  - “FPT parameters” in the real data are 6 (for exact length) and 2 (for minimum length)

Thank you!