



Lignages efficaces sur les instances quasi-arborescentes : Limites et extensions

Antoine Amarilli¹, Pierre Bourhis², Pierre Senellart^{1,3,4}

17 novembre 2016

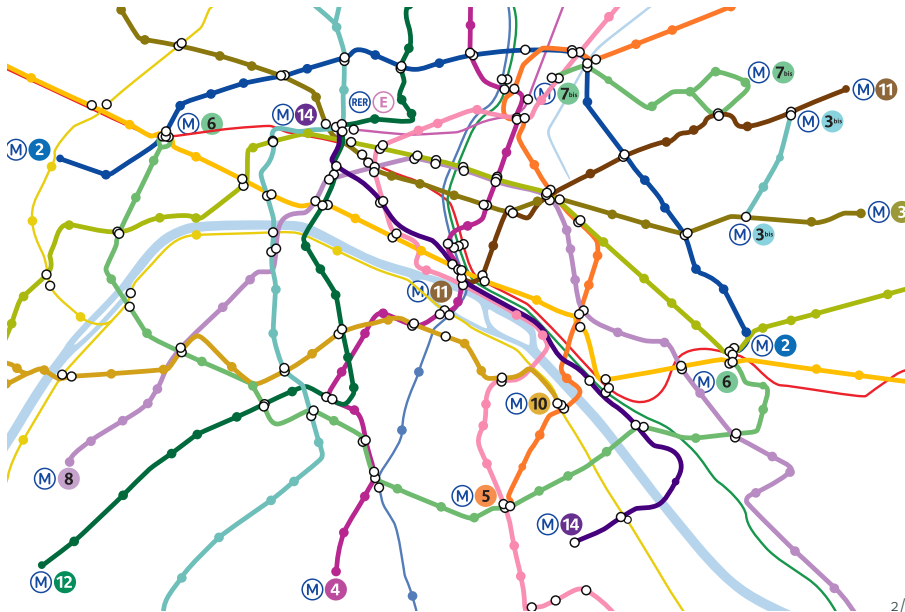
¹**Télécom ParisTech**

²CNRS CRISTAL

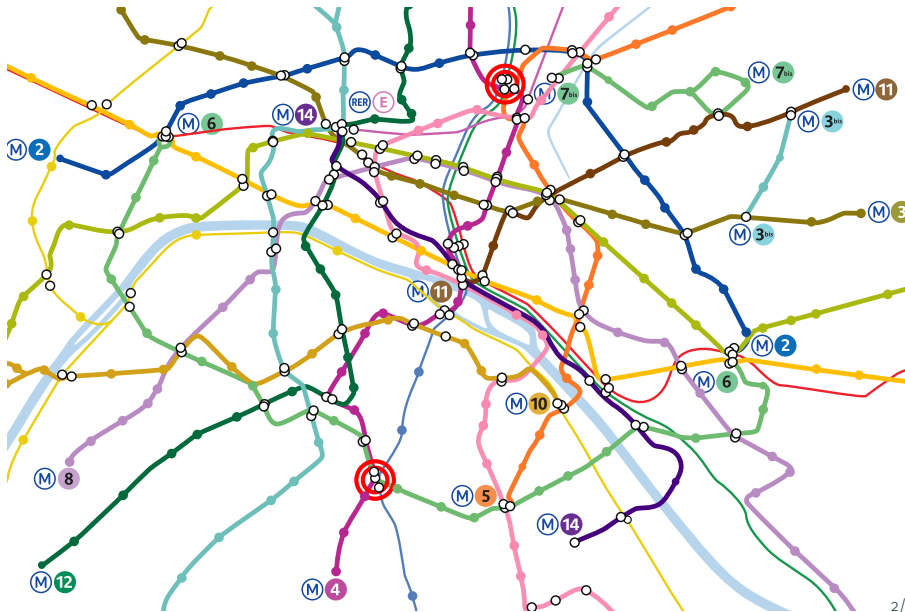
³National University of Singapore

⁴École normale supérieure de Paris

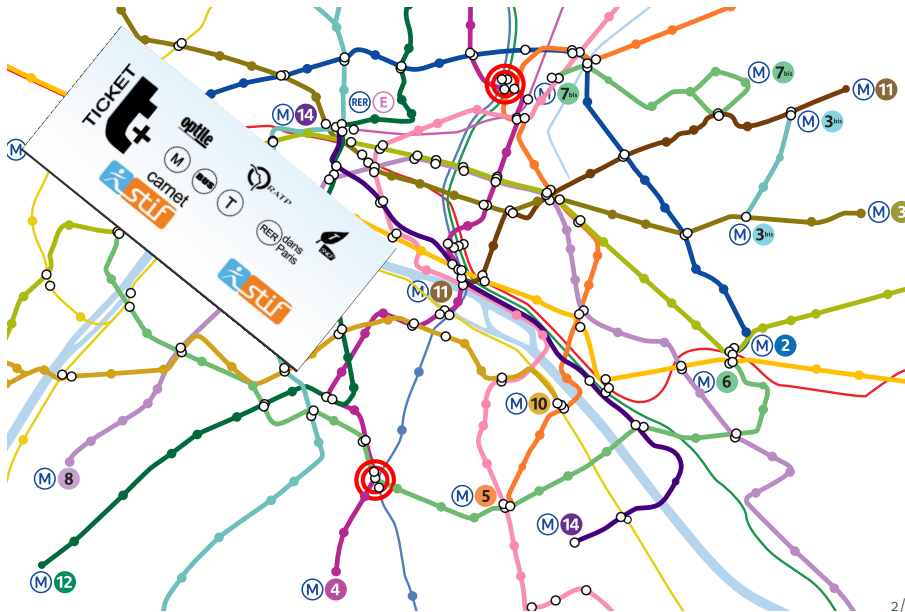
Comment aller de Porte d'Orléans à Gare du Nord ?



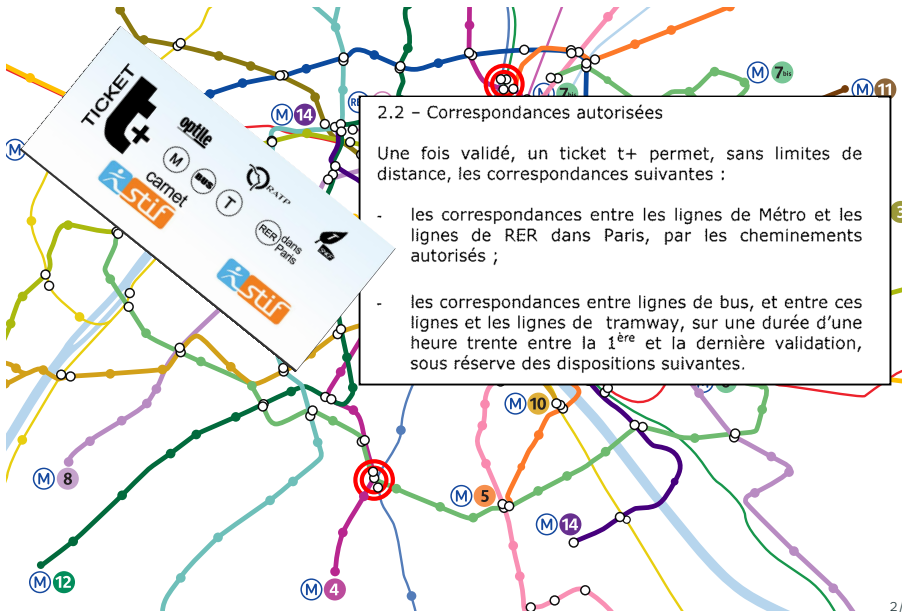
Comment aller de Porte d'Orléans à Gare du Nord ?



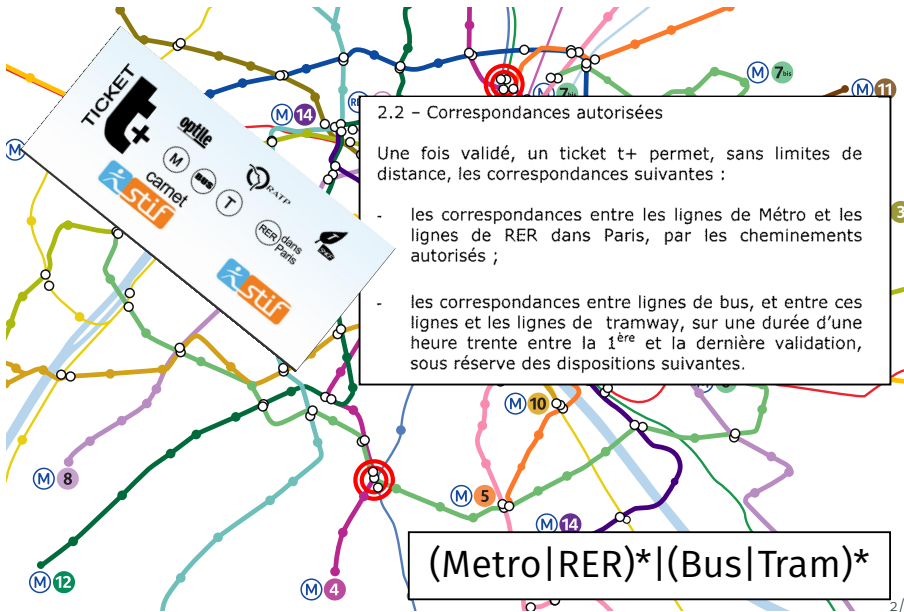
Comment aller de Porte d'Orléans à Gare du Nord ?



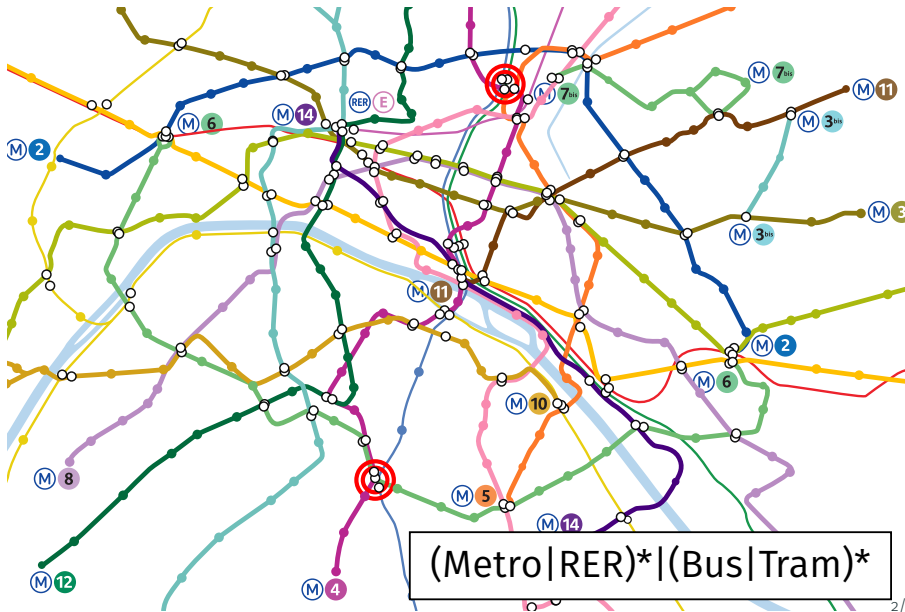
Comment aller de Porte d'Orléans à Gare du Nord ?



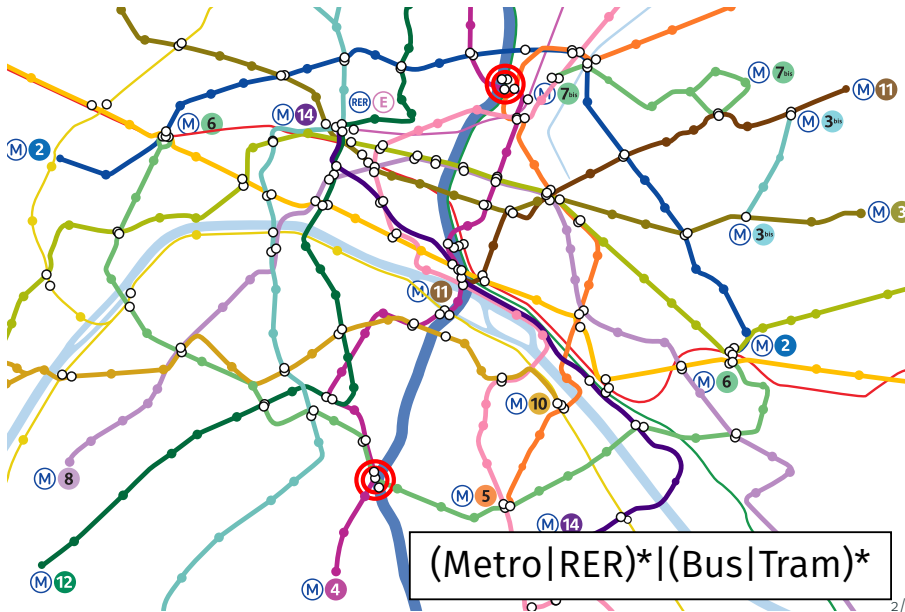
Comment aller de Porte d'Orléans à Gare du Nord ?



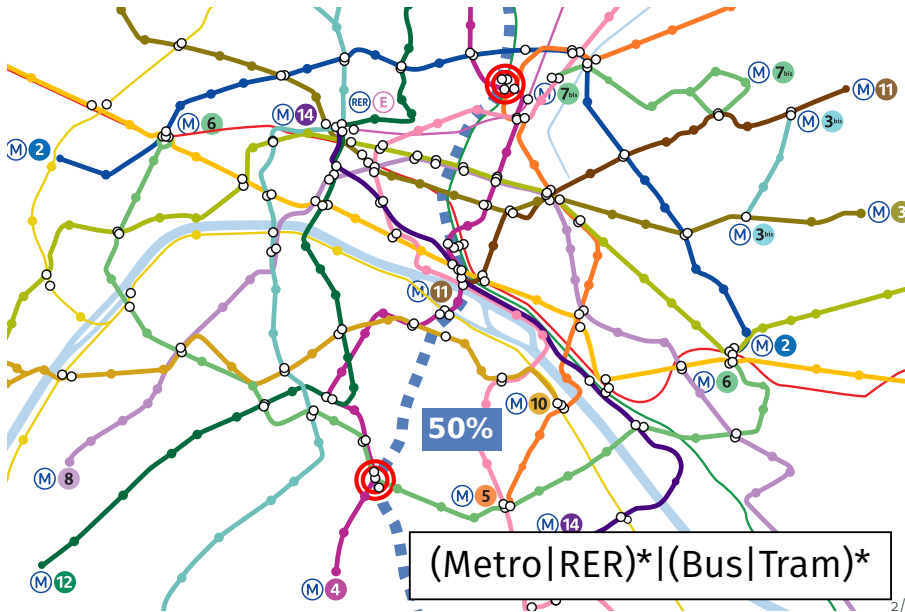
Comment aller de Porte d'Orléans à Gare du Nord ?



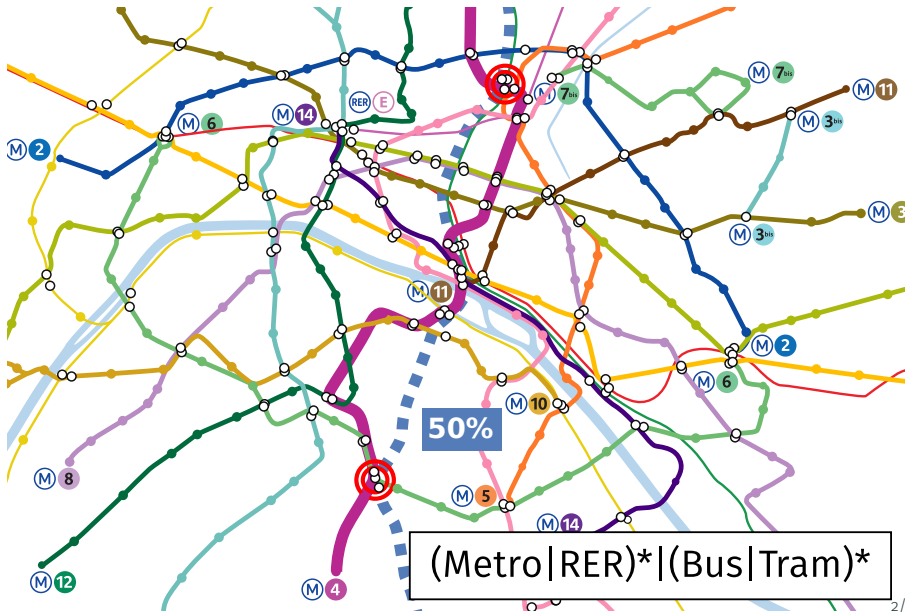
Comment aller de Porte d'Orléans à Gare du Nord ?



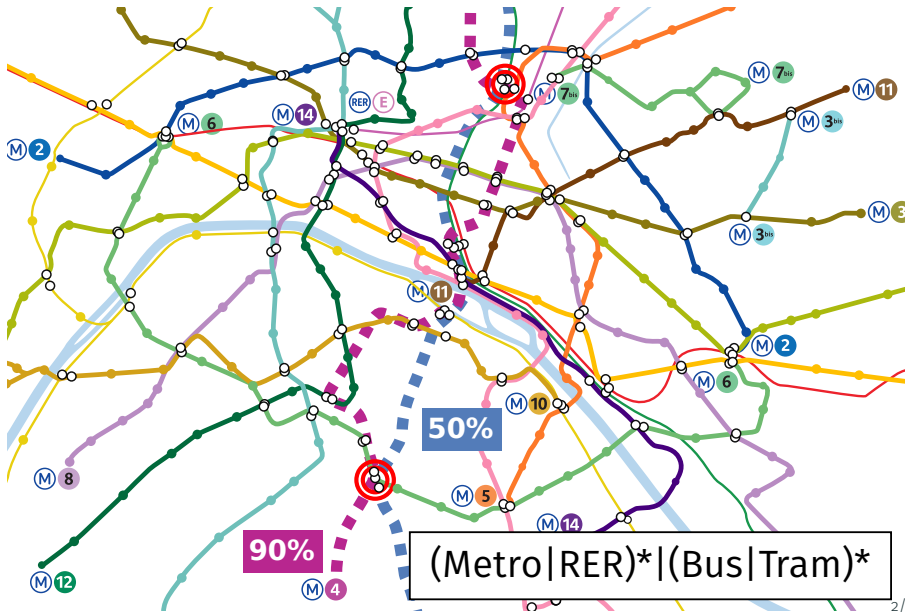
Comment aller de Porte d'Orléans à Gare du Nord ?



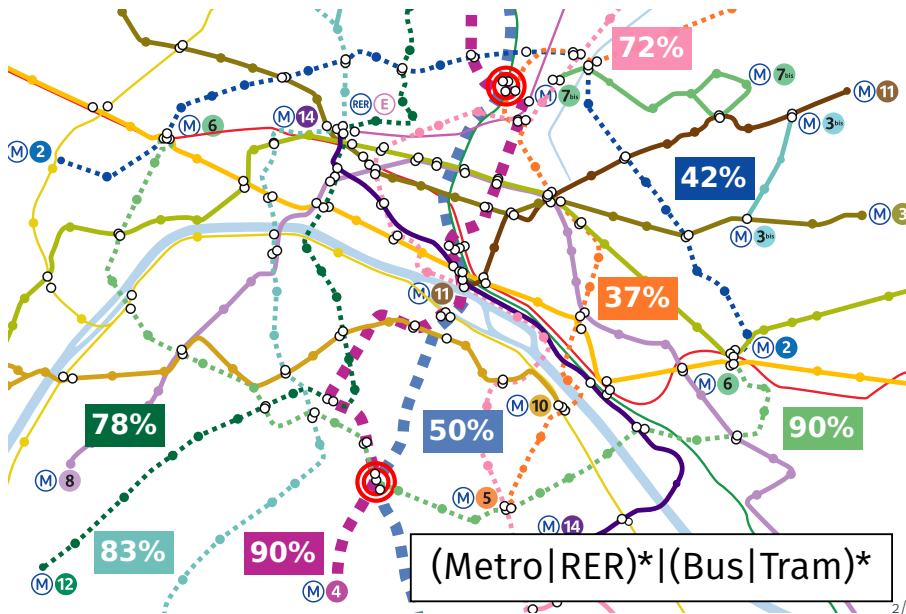
Comment aller de Porte d'Orléans à Gare du Nord ?



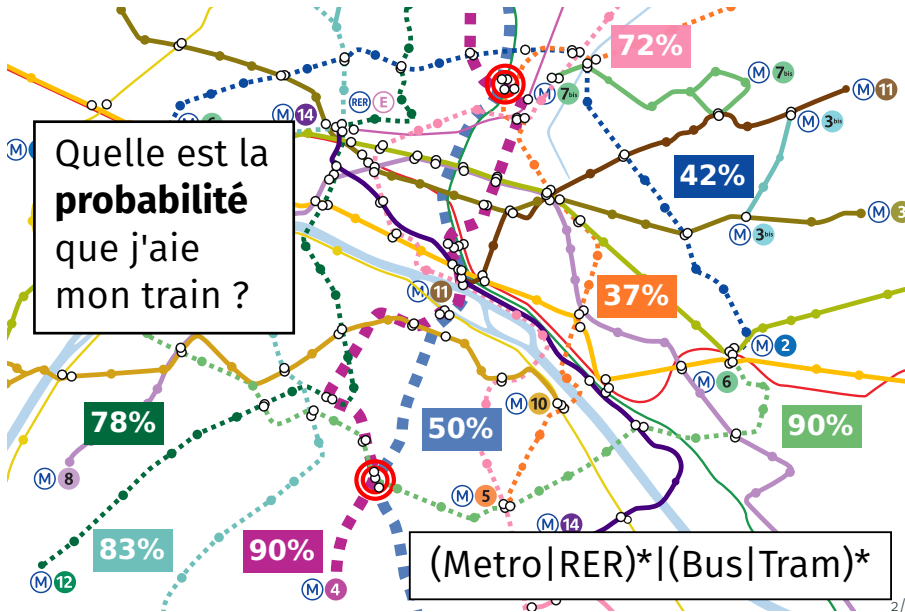
Comment aller de Porte d'Orléans à Gare du Nord ?



Comment aller de Porte d'Orléans à Gare du Nord ?



Comment aller de Porte d'Orléans à Gare du Nord ?



Définition du problème

Entrée :

 Requête Q

$(\text{Metro}|\text{RER})^*|(\text{Bus}|\text{Tram})^*$

Définition du problème

Entrée :



Requête Q

$(\text{Metro}|\text{RER})^*|(\text{Bus}|\text{Tram})^*$



Base de données D (ou graphe)



Définition du problème

Entrée :



Requête Q

$(\text{Metro}|\text{RER})^*|(\text{Bus}|\text{Tram})^*$



Base de données D (ou graphe)



Probabilités sur les faits (ou arêtes)

50%

Définition du problème

Entrée :



Requête Q

$(\text{Metro}|\text{RER})^*|(\text{Bus}|\text{Tram})^*$



Base de données D (ou graphe)



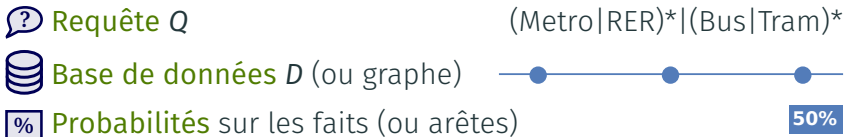
Probabilités sur les faits (ou arêtes)

50%

Sortie : La **probabilité** que la requête soit vraie sous cette distribution (en supposant l'indépendance de tous les événements)

Définition du problème

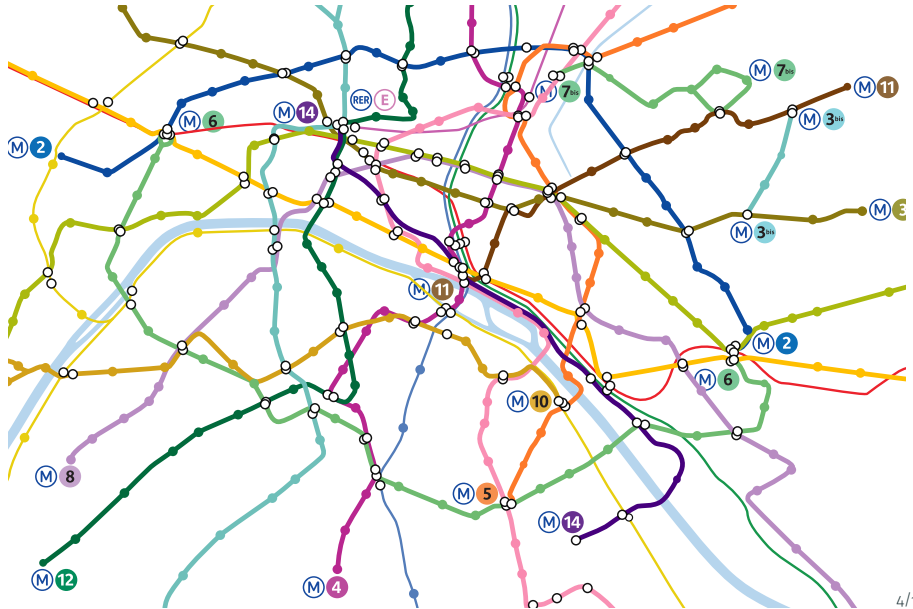
Entrée :



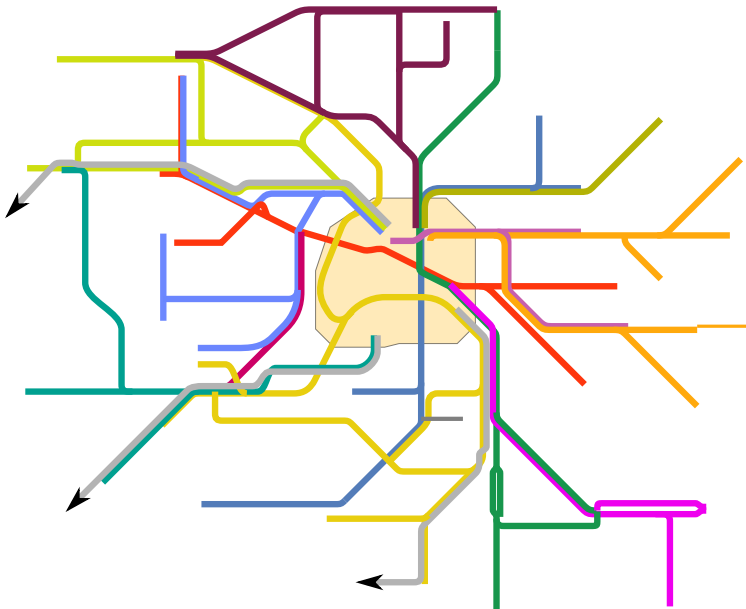
Sortie : La **probabilité** que la requête soit vraie sous cette distribution (en supposant l'indépendance de tous les événements)

Complexité : déjà **#P-difficile** en les données d'entrée !
(réduction depuis #MONOTONE-SAT)

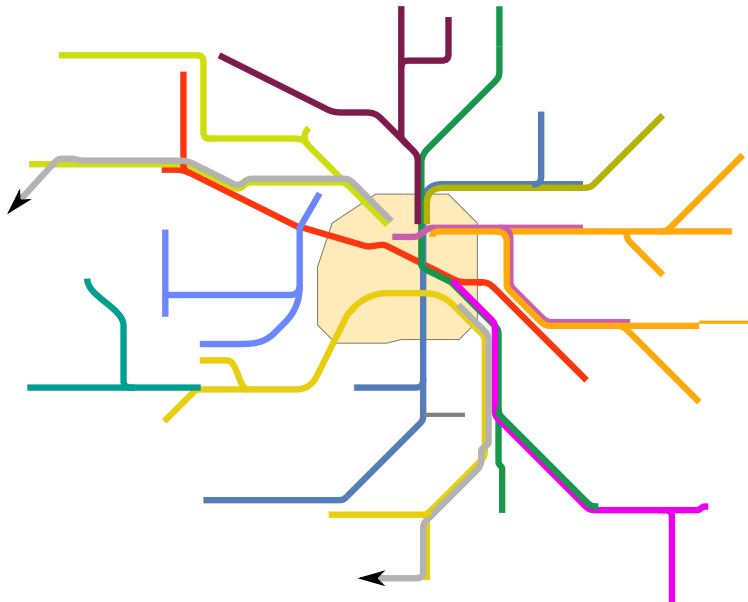
Rendre le problème faisable avec la largeur d'arbre



Rendre le problème faisable avec la largeur d'arbre

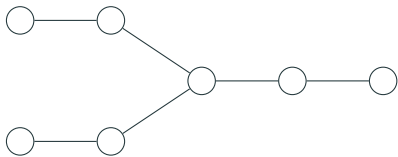


Rendre le problème faisable avec la largeur d'arbre



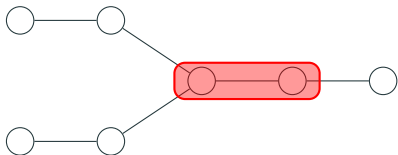
Rendre le problème faisable avec la largeur d'arbre

Exemple pour la largeur d'arbre :



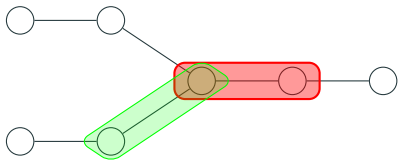
Rendre le problème faisable avec la largeur d'arbre

Exemple pour la largeur d'arbre :



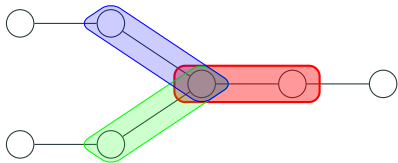
Rendre le problème faisable avec la largeur d'arbre

Exemple pour la largeur d'arbre :



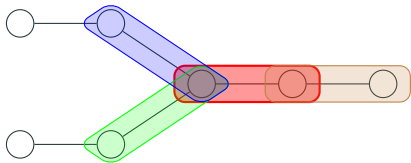
Rendre le problème faisable avec la largeur d'arbre

Exemple pour la largeur d'arbre :



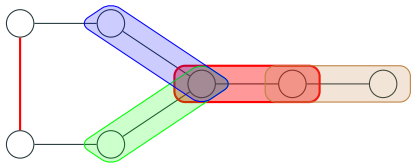
Rendre le problème faisable avec la largeur d'arbre

Exemple pour la largeur d'arbre :



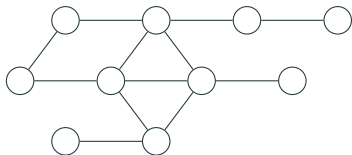
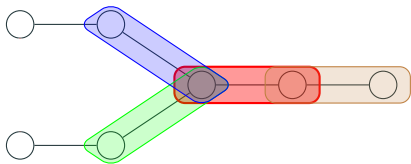
Rendre le problème faisable avec la largeur d'arbre

Exemple pour la largeur d'arbre :



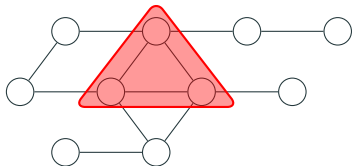
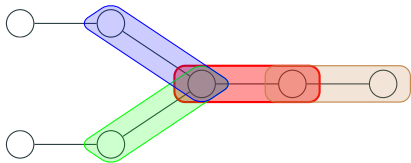
Rendre le problème faisable avec la largeur d'arbre

Exemple pour la largeur d'arbre :



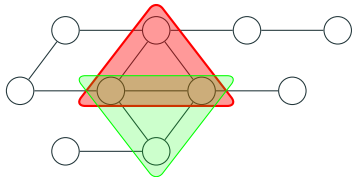
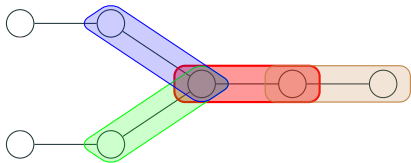
Rendre le problème faisable avec la largeur d'arbre

Exemple pour la largeur d'arbre :



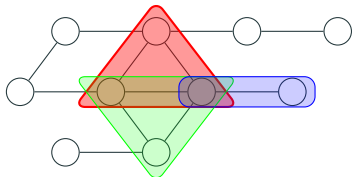
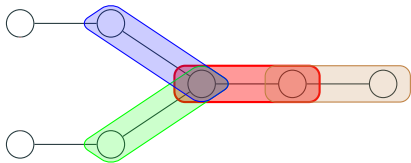
Rendre le problème faisable avec la largeur d'arbre

Exemple pour la largeur d'arbre :



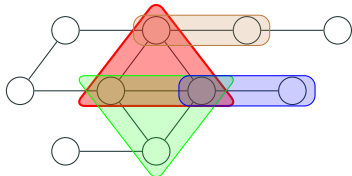
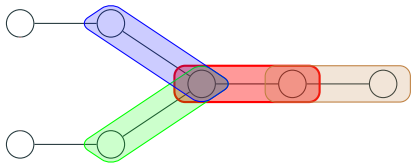
Rendre le problème faisable avec la largeur d'arbre

Exemple pour la largeur d'arbre :



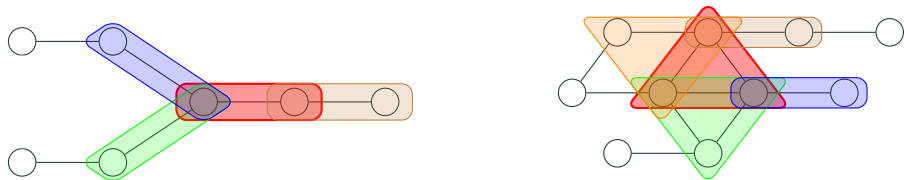
Rendre le problème faisable avec la largeur d'arbre

Exemple pour la largeur d'arbre :



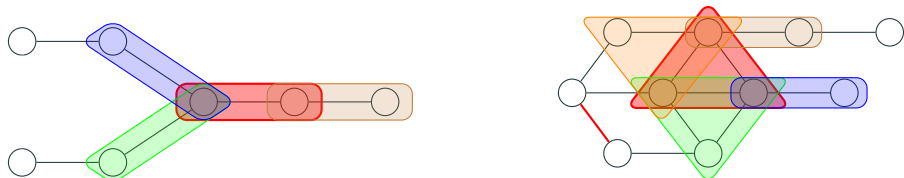
Rendre le problème faisable avec la largeur d'arbre

Exemple pour la largeur d'arbre :



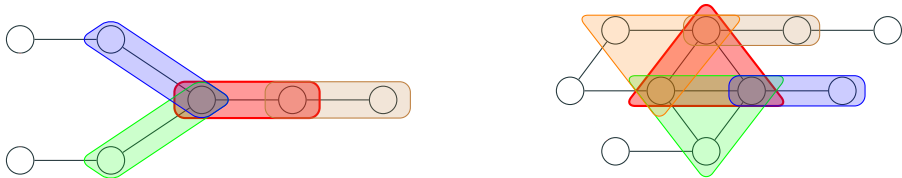
Rendre le problème faisable avec la largeur d'arbre

Exemple pour la largeur d'arbre :



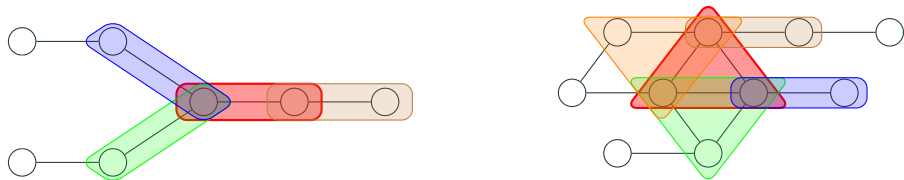
Rendre le problème faisable avec la largeur d'arbre

Exemple pour la largeur d'arbre :



Rendre le problème faisable avec la largeur d'arbre

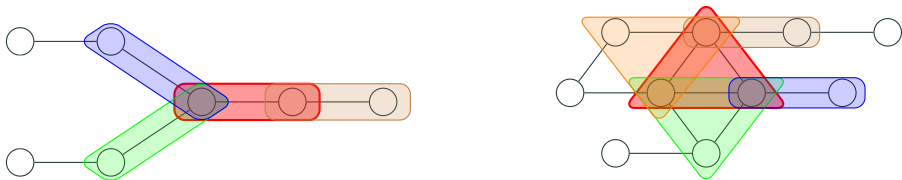
Exemple pour la largeur d'arbre :



- Les **arbres** ont une largeur de **1**
- Les **cycles** ont une largeur de **2**
- Les **k -cliques** et **$(k - 1)$ -grilles** ont une largeur de **$k - 1$**

Rendre le problème faisable avec la largeur d'arbre

Exemple pour la largeur d'arbre :



- Les **arbres** ont une largeur de **1**
- Les **cycles** ont une largeur de **2**
- Les **k -cliques** et **$(k - 1)$ -grilles** ont une largeur de **$k - 1$**

→ **Quasi-arborescent** : la **largeur** est bornée par une **constante**

Faisabilité sur les instances quasi-arborescentes

Données de
largeur bornée



Requête **MSO**

$(\text{RER}|\text{metro})^*$
 $|(\text{bus}|\text{tram})^*$

Faisabilité sur les instances quasi-arborescentes

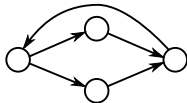
Données de
largeur bornée



Requête **MSO**

Automate d'arbres

$(\text{RER}|\text{metro})^*$
 $|(\text{bus}|\text{tram})^*$

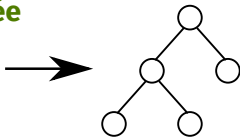


Faisabilité sur les instances quasi-arborescentes

Données de
largeur bornée



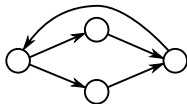
Encodage en arbre



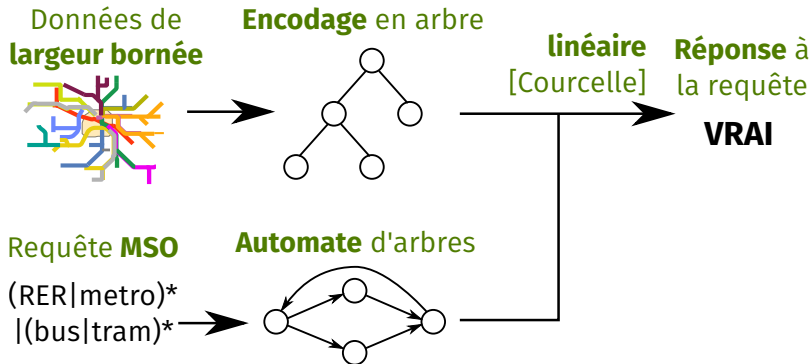
Requête **MSO**

$(\text{RER}|\text{metro})^*$
 $|(\text{bus}|\text{tram})^*$

Automate d'arbres



Faisabilité sur les instances quasi-arborescentes

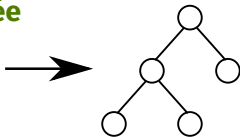


Faisabilité sur les instances quasi-arborescentes

Données de
largeur bornée



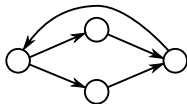
Encodage en arbre



Requête **MSO**

$(\text{RER}|\text{metro})^*$
 $|(\text{bus}|\text{tram})^*$

Automate d'arbres

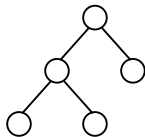


Faisabilité sur les instances quasi-arborescentes

Données de
largeur bornée

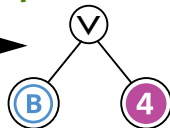


Encodage en arbre



linéaire

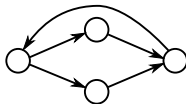
Circuit de
provenance



Requête **MSO**

$(\text{RER}|\text{metro})^*$
 $|(\text{bus}|\text{tram})^*$

Automate d'arbres

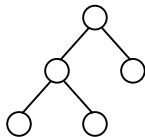


Faisabilité sur les instances quasi-arborescentes

Données de
largeur bornée

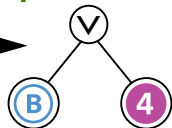


Encodage en arbre



linéaire

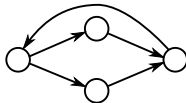
**Circuit de
provenance**



Requête **MSO**

$(\text{RER}|\text{metro})^*$
 $|(\text{bus}|\text{tram})^*$

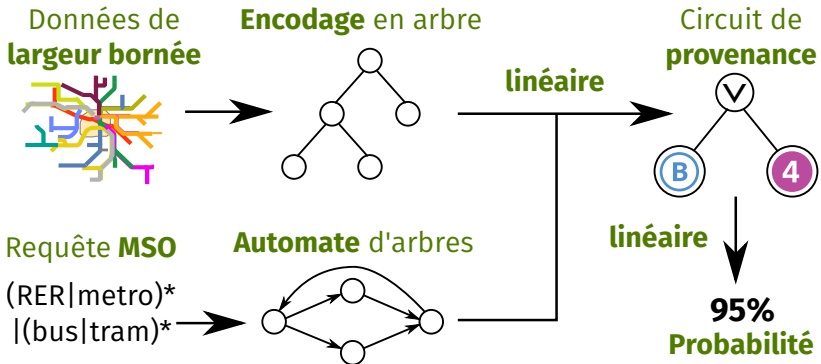
Automate d'arbres



linéaire

**95%
Probabilité**

Faisabilité sur les instances quasi-arborescentes



Théorème

Pour toute requête MSO booléenne *fixée* q et $k \in \mathbb{N}$, pour une base de données d'entrée D de *largeur* $\leq k$ avec des annotations probabilistes *indépendantes*, on peut calculer en *temps linéaire* la probabilité que D satisfasse q

Autres résultats de faisabilité

- Explication par la **compilation de lignages faisables**
 - Le circuit de provenance est une **d-DNNF**
 - On peut calculer des **OBDD** en PTIME

Autres résultats de faisabilité

- Explication par la **compilation de lignages faisables**
 - Le circuit de provenance est une **d-DNNF**
 - On peut calculer des **OBDD** en PTIME
- Extension à des événements incertains **corrélés**
 - Faisable si les corrélations sont de **largeur bornée**
 - Par ex., si elles **suivent la structure** (arbres probabilistes)

Autres résultats de faisabilité

- Explication par la **compilation de lignages faisables**
 - Le circuit de provenance est une **d-DNNF**
 - On peut calculer des **OBDD** en PTIME
- Extension à des événements incertains **corrélés**
 - Faisable si les corrélations sont de **largeur bornée**
 - Par ex., si elles **suivent la structure** (arbres probabilistes)
- Extension à des **semianneaux de provenance** généraux ($\mathbb{N}[X]$)

Autres résultats de faisabilité

- Explication par la **compilation de lignages faisables**
 - Le circuit de provenance est une **d-DNNF**
 - On peut calculer des **OBDD** en PTIME
- Extension à des événements incertains **corrélés**
 - Faisable si les corrélations sont de **largeur bornée**
 - Par ex., si elles **suivent la structure** (arbres probabilistes)
- Extension à des **semianneaux de provenance** généraux ($\mathbb{N}[X]$)
- Liens avec les **requêtes prudentes** de [Dalvi et Suciu, 2012]
 - Les requêtes **sans inversion** se traitent par nos méthodes

Que faire si la largeur n'est pas bornée ?

Borne inférieure

Que faire si la largeur n'est pas bornée ? ... pas grand-chose.

Théorème

Pour toute signature *de graphe* σ ,
il y a une requête du *premier ordre* q telle que
pour toute famille constructible $\mathcal{I}$ de *largeur non bornée*,
l'évaluation probabiliste de q sur $\mathcal{I}$ est *#P-dure* sous réductions RP

Théorème

Pour toute signature *de graphe* σ ,
il y a une requête du *premier ordre* q telle que
pour toute famille constructible $\mathcal{I}$ de *largeur non bornée*,
l'évaluation probabiliste de q sur $\mathcal{I}$ est *#P-dure* sous réductions RP

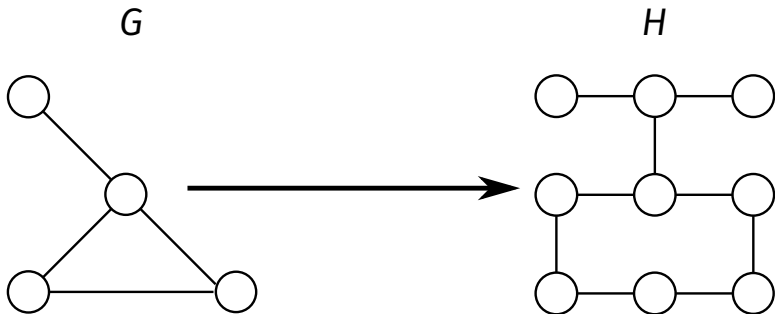
- *Largeur non bornée* : pour tout $k \in \mathbb{N}$, il y a $I_k \in \mathcal{I}$ de largeur $\geq k$
- *Constructible* : étant donné k , on peut calculer une telle I_k en PTIME
- *#P-dure sous réductions RP* : réduire en PTIME avec haute proba depuis le problème de compter les chemins acceptants d'une machine de Turing nondéterministe PTIME

Idée : mineurs topologiques

- G est un mineur topologique de H si :

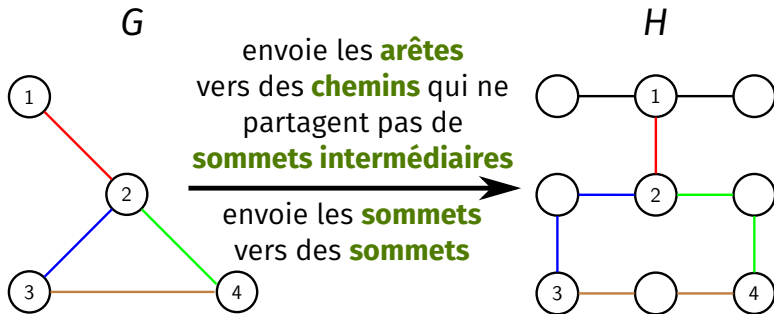
Idée : mineurs topologiques

- G est un mineur topologique de H si :



Idée : mineurs topologiques

- G est un mineur topologique de H si :



Résultats sur l'extraction de mineurs topologiques

Théorème [Robertson et Seymour, 1986]

*Pour tout graphe planaire G de degré ≤ 3 ,
pour tout graphe H de **largeur suffisamment grande**,
 G est un mineur topologique de H .*

Résultats sur l'extraction de mineurs topologiques

Théorème [Robertson et Seymour, 1986]

Pour tout graphe planaire G de degré ≤ 3 ,
pour tout graphe H de *largeur suffisamment grande*,
 G est un mineur topologique de H .

Plus récemment :

Théorème [Chekuri et Chuzhoy, 2014]

Il y a une certaine constante $c \in \mathbb{N}$ telle que
pour tout graphe planaire G de degré ≤ 3 et H de *largeur $\geq |G|^c$,*
on peut extraire G comme mineur topologique de H
en PTIME avec haute proba

Ébauche de la preuve du résultat

- Choisir le **problème** à partir duquel on réduit :
 - Doit être **#P-difficile** sur les graphes planaires de degré 3
 - Doit s'encoder comme une **requête FO q**
- Choix : problème de comptage des **couplages de graphe**

Ébauche de la preuve du résultat

- Choisir le **problème** à partir duquel on réduit :
 - Doit être **#P-difficile** sur les graphes planaires de degré 3
 - Doit s'encoder comme une **requête FO** q
 - Choix : problème de comptage des **couplages de graphe**
- Étant donné un **graphe d'entrée** G , calculer $k := |G|^c$

Ébauche de la preuve du résultat

- Choisir le **problème** à partir duquel on réduit :
 - Doit être **#P-difficile** sur les graphes planaires de degré 3
 - Doit s'encoder comme une **requête FO** q

→ Choix : problème de comptage des **couplages de graphe**
- Étant donné un **graphe d'entrée** G , calculer $k := |G|^c$
- Calculer en PTIME une **instance** I_k de $\mathcal{I}$ de largeur $\geq k$

Ébauche de la preuve du résultat

- Choisir le **problème** à partir duquel on réduit :
 - Doit être **#P-difficile** sur les graphes planaires de degré 3
 - Doit s'encoder comme une **requête FO** q

→ Choix : problème de comptage des **couplages de graphe**
- Étant donné un **graphe d'entrée** G , calculer $k := |G|^c$
- Calculer en PTIME une **instance** I_k de $\mathcal{I}$ de largeur $\geq k$
- Extraire en temps RP G comme **mineur topologique** de I_k

Ébauche de la preuve du résultat

- Choisir le **problème** à partir duquel on réduit :
 - Doit être **#P-difficile** sur les graphes planaires de degré 3
 - Doit s'encoder comme une **requête FO** q
 - Choix : problème de comptage des **couplages de graphe**
- Étant donné un **graphe d'entrée** G , calculer $k := |G|^c$
- Calculer en PTIME une **instance** I_k de $\mathcal{I}$ de largeur $\geq k$
- Extraire en temps RP G comme **mineur topologique** de I_k
- Construire une **annotation probabiliste** de I_k telle que :
 - Les arêtes inutiles de I_k sont éliminées
 - L'évaluation pour q **donne la réponse** du problème difficile
 - Facile pour MSO mais **subtil** pour FO

Autres bornes inférieures dans l'article

- Évaluation non-probabiliste sur familles closes par sous-graphe
 - Hypothèse de constructibilité plus faible (*densely unbounded poly-logarithmically*)
 - Liens avec [Ganian et al., 2014]

Autres bornes inférieures dans l'article

- Évaluation non-probabiliste sur familles closes par sous-graphe
 - Hypothèse de constructibilité plus faible (*densely unbounded poly-logarithmically*)
 - Liens avec [Ganian et al., 2014]
- Comptage de résultats de requêtes
 - Compter le nombre de X tels que I satisfasse $q(X)$

Autres bornes inférieures dans l'article

- **Évaluation non-probabiliste** sur familles **closes par sous-graphe**
 - Hypothèse de constructibilité **plus faible** (*densely unbounded poly-logarithmically*)
 - Liens avec [Ganian et al., 2014]
- **Comptage de résultats** de requêtes
 - Compter le **nombre** de X tels que I satisfasse $q(X)$
- Taille des représentations de **lignages**
 - Il y a une **union de requêtes conjonctives avec inégalités** ($UCQ^{\neq}$) qui n'a d'OBDD de taille polynomiale sur aucune famille de graphes de largeur non bornée constructible (au sens faible)
 - **Méta-dichotomie** sur les $UCQ^{\neq}$ avec cette propriété

Travaux en cours et ultérieurs

- Améliorer la **borne inférieure**
 - **Étendre** : aller des **graphes** à des instances d'**arité arbitraire**
 - **Restreindre** : aller de **FO** à des **UCQ $\neq$**

Travaux en cours et ultérieurs

- Améliorer la **borne inférieure**
 - **Étendre** : aller des **graphes** à des instances d'**arité arbitraire**
 - **Restreindre** : aller de **FO** à des **UCQ $\neq$**
- Complexité **combinée** : actuellement $\Omega\left(2^{2^{\dots^{2^{|Q|}}}} \times |D|\right)$
 - Quelles requêtes peut-on compiler **efficacement** en automates ?
 - Quand peut-on avoir une **complexité combinée** faisable ?

Travaux en cours et ultérieurs

- Améliorer la **borne inférieure**
 - **Étendre** : aller des **graphes** à des instances d'**arité arbitraire**
 - **Restreindre** : aller de **FO** à des **UCQ $\neq$**
- Complexité **combinée** : actuellement $\Omega\left(2^{2^{\dots 2^{|Q|}}} \times |D|\right)$
 - Quelles requêtes peut-on compiler **efficacement** en automates ?
 - Quand peut-on avoir une **complexité combinée** faisable ?
- Pour les requêtes **non-booléennes**, peut-on énumérer **efficacement** les résultats ?
 - Énumération **à délai constant** connue pour MSO sur données quasi-arborescentes [Bagan, 2006, Kazana et Segoufin, 2013]
 - Peut-on le **reprover** plus simplement sur les d-DNNF ?

Travaux en cours et ultérieurs

- Améliorer la **borne inférieure**
 - **Étendre** : aller des **graphes** à des instances d'**arité arbitraire**
 - **Restreindre** : aller de **FO** à des **UCQ $\neq$**
- Complexité **combinée** : actuellement $\Omega\left(2^{2^{\dots 2^{|Q|}}} \times |D|\right)$
 - Quelles requêtes peut-on compiler **efficacement** en automates ?
 - Quand peut-on avoir une **complexité combinée** faisable ?
- Pour les requêtes **non-booléennes**, peut-on énumérer **efficacement** les résultats ?
 - Énumération **à délai constant** connue pour MSO sur données quasi-arborescentes [Bagan, 2006, Kazana et Segoufin, 2013]
 - Peut-on le **reprover** plus simplement sur les d-DNNF ?

Merci pour votre attention !

Références I



Bagan, G. (2006).

Mso queries on tree decomposable structures are computable with linear delay.

In *CSL*.



Chaudhuri, S. et Vardi, M. Y. (1992).

On the equivalence of recursive and nonrecursive Datalog programs.

In *PODS*.



Chekuri, C. et Chuzhoy, J. (2014).

Polynomial bounds for the grid-minor theorem.

In *STOC*.

Références II



Courcelle, B. (1990).

The monadic second-order logic of graphs. I. Recognizable sets of finite graphs.

Inf. Comput., 85(1).



Dalvi, N. et Suciu, D. (2012).

The dichotomy of probabilistic inference for unions of conjunctive queries.

JACM, 59(6):30.



Ganian, R., Hliněný, P., Langer, A., Obdržálek, J., Rossmanith, P., et Sikdar, S. (2014).

Lower bounds on the complexity of MSO₁ model-checking.

JCSS, 1(80).

Références III



Green, T. J., Karvounarakis, G., et Tannen, V. (2007).

Provenance semirings.

In *PODS*.



Kazana, W. et Segoufin, L. (2013).

Enumeration of first-order queries on classes of structures with bounded expansion.

In *PODS*.



Robertson, N. et Seymour, P. D. (1986).

Graph minors. V. Excluding a planar graph.

J. Comb. Theory, Ser. B, 41(1).

Crédits pour les illustrations

- Transparent 2:
 - https://commons.wikimedia.org/wiki/File:Paris_Metro_map.svg (recadrée), utilisateur Umx sur Wikimedia Commons, domaine public
 - <http://www.parisvoyage.com/images/cartoon18.jpg>, ParisVoyage, en vertu du droit de citation
 - http://www.vianavigo.com/fileadmin/galerie/pdf/CGU_t_.pdf (recadrée), RATP, en vertu du droit de citation
- Transparents 4 and 5:
https://commons.wikimedia.org/wiki/File:Carte_Transilien_RER_sch%C3%A9matique.svg (modifiée), utilisateur Benjamin Smith sur Wikimedia Commons, licence CC BY-SA 4.0 international.

Non-probabilistic evaluation

Apply [Chekuri et Chuzhoy, 2014] to method of [Ganian et al., 2014] for MSO **non-probabilistic** query evaluation (QE):

Théorème

*For any arity-2 signature σ and level Σ_i^P of the polynomial hierarchy, there is a **MSO** query q_i such that, for any constructible, **subinstance-closed, unbounded-tw** class $\mathcal{I}$, the QE problem for $\mathcal{Q} = \{q_i\}$ and $\mathcal{I}$ is **Σ_i^P -hard** under RP reductions*

Non-probabilistic evaluation

Apply [Chekuri et Chuzhoy, 2014] to method of [Ganian et al., 2014] for MSO **non-probabilistic** query evaluation (QE):

Théorème

*For any arity-2 signature σ and level Σ_i^P of the polynomial hierarchy, there is a **MSO** query q_i such that, for any constructible, **subinstance-closed, unbounded-tw** class $\mathcal{I}$, the QE problem for $\mathcal{Q} = \{q_i\}$ and $\mathcal{I}$ is **Σ_i^P -hard** under RP reductions*

Variant: we also show a result like in [Ganian et al., 2014], with:

- weaker **constructibility** requirement on $\mathcal{I}$:
 - $\mathcal{I}$ is **densely unbounded poly-logarithmically**
- stronger **complexity hypothesis**:
 - PH does not admit $2^{o(n)}$ -sized circuits

Lower bound on OBDD representations

OBDD for a query q on instance I :

ordered decision diagram on the facts of I to decide whether q holds

Lower bound on OBDD representations

OBDD for a query q on instance I :

ordered decision diagram on the facts of I to decide whether q holds

$$q : \exists x y R(x) \wedge S(x, y) \wedge T(y)$$

R	S	T
$a \quad r_1$	$a \quad a \quad s_1$	$v \quad t_1$
$b \quad r_2$	$b \quad v \quad s_2$	$w \quad t_2$
$c \quad r_3$	$b \quad w \quad s_3$	$b \quad t_3$

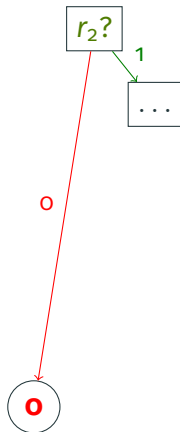
Lower bound on OBDD representations

OBDD for a query q on instance I :

ordered decision diagram on the facts of I to decide whether q holds

$$q : \exists x y R(x) \wedge S(x, y) \wedge T(y)$$

R		S			T	
a	r_1	a	a	s_1	v	t_1
b	r_2	b	v	s_2	w	t_2
c	r_3	b	w	s_3	b	t_3



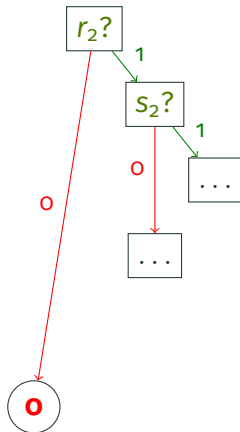
Lower bound on OBDD representations

OBDD for a query q on instance I :

ordered decision diagram on the facts of I to decide whether q holds

$$q : \exists x y R(x) \wedge S(x, y) \wedge T(y)$$

R		S			T	
a	r_1	a	a	s_1	v	t_1
b	r_2	b	v	s_2	w	t_2
c	r_3	b	w	s_3	b	t_3



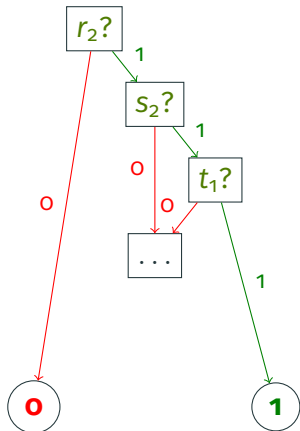
Lower bound on OBDD representations

OBDD for a query q on instance I :

ordered decision diagram on the facts of I to decide whether q holds

$$q : \exists x y R(x) \wedge S(x, y) \wedge T(y)$$

R		S			T	
a	r_1	a	a	s_1	v	t_1
b	r_2	b	v	s_2	w	t_2
c	r_3	b	w	s_3	b	t_3



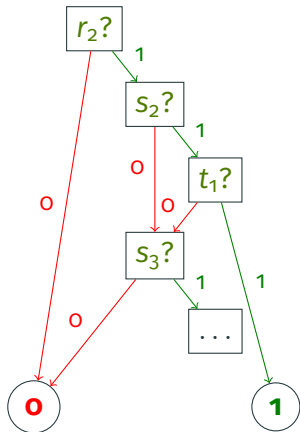
Lower bound on OBDD representations

OBDD for a query q on instance I :

ordered decision diagram on the facts of I to decide whether q holds

$$q : \exists x y R(x) \wedge S(x, y) \wedge T(y)$$

R		S			T	
a	r_1	a	a	s_1	v	t_1
b	r_2	b	v	s_2	w	t_2
c	r_3	b	w	s_3	b	t_3



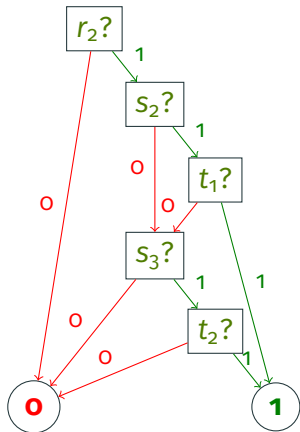
Lower bound on OBDD representations

OBDD for a query q on instance I :

ordered decision diagram on the facts of I to decide whether q holds

$$q : \exists x y R(x) \wedge S(x, y) \wedge T(y)$$

R		S			T	
a	r_1	a	a	s_1	v	t_1
b	r_2	b	v	s_2	w	t_2
c	r_3	b	w	s_3	b	t_3



Lower bound on OBDD representations (result)

- If we find an OBDD of q on I in PTIME, then PQE of q on I also is
- Show inexistence of poly-size OBDDs (rather than PQE hardness)

Lower bound on OBDD representations (result)

- If we find an OBDD of q on I in PTIME, then PQE of q on I also is
- Show inexistence of poly-size OBDDs (rather than PQE hardness)

Théorème

For any arity-2 signature σ , there is a UCQ with inequalities q s.t.
for any $\mathcal{I}$ of treewidth *densely unbounded poly-logarithmically*,
 q has *no OBDDs of polynomial size* on instances of $\mathcal{I}$

Lower bound on OBDD representations (result)

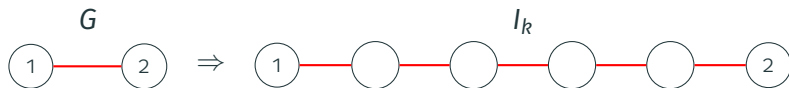
- If we find an OBDD of q on I in PTIME, then PQE of q on I also is
- Show **inexistence of poly-size OBDDs** (rather than PQE hardness)

Théorème

For any arity-2 signature σ , there is a UCQ with inequalities q s.t.
for any $\mathcal{I}$ of treewidth **densely unbounded poly-logarithmically**,
 q has **no OBDDs of polynomial size** on instances of $\mathcal{I}$

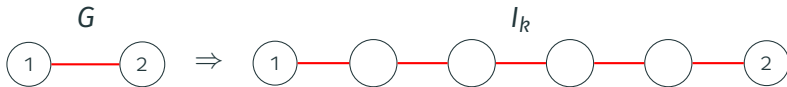
- **Meta-dichotomy** on the connected UCQ $^\neq$ for which this holds

Defining the query



- In the embedding, edges of G can become **long paths** in I_k
- q must answer the hard problem on G **despite subdivisions**
 - Easy in MSO but tricky in **FO**!

Defining the query



- In the embedding, edges of G can become **long paths** in I_k
 - q must answer the hard problem on G **despite subdivisions**
 - Easy in MSO but tricky in FO!
- Our q restricts to a **subset of the worlds** of known weight and gives the right answer **up to renormalization**

Encoding treelike instances [Chaudhuri et Vardi, 1992]

Instance:

N

a *b*

b *c*

c *d*

d *e*

e *f*

S

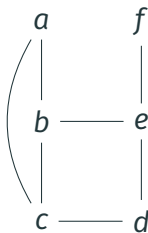
a *c*

b *e*

Encoding treelike instances [Chaudhuri et Vardi, 1992]

Instance: Gaifman graph:

N	
<i>a</i>	<i>b</i>
<i>b</i>	<i>c</i>
<i>c</i>	<i>d</i>
<i>d</i>	<i>e</i>
<i>e</i>	<i>f</i>

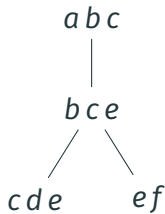
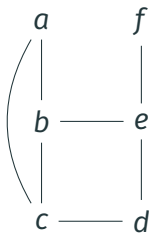


S	
<i>a</i>	<i>c</i>
<i>b</i>	<i>e</i>

Encoding treelike instances [Chaudhuri et Vardi, 1992]

Instance: Gaifman graph: Tree decomp.:

N	
<i>a</i>	<i>b</i>
<i>b</i>	<i>c</i>
<i>c</i>	<i>d</i>
<i>d</i>	<i>e</i>
<i>e</i>	<i>f</i>



S	
<i>a</i>	<i>c</i>
<i>b</i>	<i>e</i>

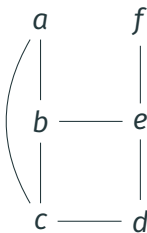
Encoding treelike instances [Chaudhuri et Vardi, 1992]

Instance:

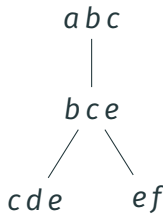
N	
a	b
b	c
c	d
d	e
e	f

S	
a	c
b	e

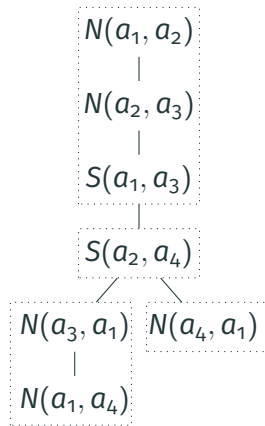
Gaifman graph:



Tree decomp.:



Tree encoding:



Provenance semirings

- **Semiring** of positive Boolean functions ($\text{PosBool}[X], \vee, \wedge, f, t$)

Provenance semirings

- **Semiring** of positive Boolean functions ($\text{PosBool}[X], \vee, \wedge, f, t$)
- **Provenance semirings:** [Green et al., 2007]
 - Provenance generalized to **arbitrary (commutative) semirings**
 - For queries in the **positive relational algebra** and Datalog

Provenance semirings

- **Semiring** of positive Boolean functions ($\text{PosBool}[X]$, $\vee$, $\wedge$, f , t)
- **Provenance semirings**: [Green et al., 2007]
 - Provenance generalized to **arbitrary (commutative) semirings**
 - For queries in the **positive relational algebra** and Datalog

→ Our circuits capture $\text{PosBool}[X]$ -**provenance** in this sense

Provenance semirings

- **Semiring** of positive Boolean functions ($\text{PosBool}[X], \vee, \wedge, f, t$)
 - **Provenance semirings**: [Green et al., 2007]
 - Provenance generalized to **arbitrary (commutative) semirings**
 - For queries in the **positive relational algebra** and Datalog
- Our circuits capture $\text{PosBool}[X]$ -**provenance** in this sense
- The **definitions** match: all subinstances that satisfy the query

Provenance semirings

- **Semiring** of positive Boolean functions ($\text{PosBool}[X]$, $\vee$, $\wedge$, f , t)
 - **Provenance semirings**: [Green et al., 2007]
 - Provenance generalized to **arbitrary (commutative) semirings**
 - For queries in the **positive relational algebra** and Datalog
- Our circuits capture **$\text{PosBool}[X]$ -provenance** in this sense
- The **definitions** match: all subinstances that satisfy the query
 - For **monotone** queries, we can construct **positive** circuits

Universal provenance

- **Universal** semiring of polynomials $(\mathbb{N}[X], +, \times, 0, 1)$
 - The provenance for $\mathbb{N}[X]$ can be **specialized** to any $K[X]$

Universal provenance

- **Universal** semiring of polynomials $(\mathbb{N}[X], +, \times, 0, 1)$
 - The provenance for $\mathbb{N}[X]$ can be **specialized** to any $K[X]$
- Captures many **useful semirings**:
 - counting the number of **matches** of a query
 - computing the **security level** of a query result
 - computing the **cost** of a query result

$\mathbb{N}[X]$ -provenance example

R		
a	b	x_1
b	c	x_2
d	e	x_3
e	d	x_4
f	f	x_5

$\mathbb{N}[X]$ -provenance example

R		
a	b	x_1
b	c	x_2
d	e	x_3
e	d	x_4
f	f	x_5

$$\exists x y z R(x, y) \wedge R(y, z)$$

→ PosBool[X]-provenance:

→ $\mathbb{N}[X]$ -provenance:

- **Definition** of provenance for conjunctive queries:
 - **Sum** over query matches
 - **Multiply** over matched facts

$\mathbb{N}[X]$ -provenance example

R		
a	b	x_1
b	c	x_2
d	e	x_3
e	d	x_4
f	f	x_5

$$\exists x y z R(x, y) \wedge R(y, z)$$

→ PosBool[X]-provenance:

→ $\mathbb{N}[X]$ -provenance:

- **Definition** of provenance for conjunctive queries:
 - **Sum** over query matches
 - **Multiply** over matched facts

$\mathbb{N}[X]$ -provenance example

R		
a	b	x_1
b	c	x_2
d	e	x_3
e	d	x_4
f	f	x_5

$$\exists x y z R(x, y) \wedge R(y, z)$$

→ PosBool[X]-provenance:
 $(x_1 \wedge x_2)$

→ $\mathbb{N}[X]$ -provenance:
 $(x_1 \times x_2)$

- **Definition** of provenance for conjunctive queries:
 - **Sum** over query matches
 - **Multiply** over matched facts

$\mathbb{N}[X]$ -provenance example

R		
a	b	x_1
b	c	x_2
d	e	x_3
e	d	x_4
f	f	x_5

$$\exists x y z R(x, y) \wedge R(y, z)$$

→ PosBool[X]-provenance:
 $(x_1 \wedge x_2)$

→ $\mathbb{N}[X]$ -provenance:
 $(x_1 \times x_2)$

- **Definition** of provenance for conjunctive queries:
 - **Sum** over query matches
 - **Multiply** over matched facts

$\mathbb{N}[X]$ -provenance example

R		
a	b	x_1
b	c	x_2
d	e	x_3
e	d	x_4
f	f	x_5

$$\exists x y z R(x, y) \wedge R(y, z)$$

$$\rightarrow \text{PosBool}[X]\text{-provenance:}$$
$$(x_1 \wedge x_2) \vee (x_3 \wedge x_4)$$

$$\rightarrow \mathbb{N}[X]\text{-provenance:}$$

$$(x_1 \times x_2) + (x_3 \times x_4)$$

- **Definition** of provenance for conjunctive queries:
 - **Sum** over query matches
 - **Multiply** over matched facts

$\mathbb{N}[X]$ -provenance example

R		
a	b	x_1
b	c	x_2
d	e	x_3
e	d	x_4
f	f	x_5

$$\exists x y z R(x, y) \wedge R(y, z)$$

$$\rightarrow \text{PosBool}[X]\text{-provenance:}$$
$$(x_1 \wedge x_2) \vee (x_3 \wedge x_4)$$

$$\rightarrow \mathbb{N}[X]\text{-provenance:}$$

$$(x_1 \times x_2) + (x_3 \times x_4)$$

- **Definition** of provenance for conjunctive queries:
 - **Sum** over query matches
 - **Multiply** over matched facts

$\mathbb{N}[X]$ -provenance example

R		
a	b	x_1
b	c	x_2
d	e	x_3
e	d	x_4
f	f	x_5

$$\exists x y z R(x, y) \wedge R(y, z)$$

$$\rightarrow \text{PosBool}[X]\text{-provenance:}$$
$$(x_1 \wedge x_2) \vee (x_3 \wedge x_4)$$

$$\rightarrow \mathbb{N}[X]\text{-provenance:}$$

$$(x_1 \times x_2) + (x_3 \times x_4) + (x_4 \times x_3)$$

- **Definition** of provenance for conjunctive queries:
 - **Sum** over query matches
 - **Multiply** over matched facts

$\mathbb{N}[X]$ -provenance example

R		
a	b	x_1
b	c	x_2
d	e	x_3
e	d	x_4
f	f	x_5

$$\exists x y z R(x, y) \wedge R(y, z)$$

$$\rightarrow \text{PosBool}[X]\text{-provenance:}$$
$$(x_1 \wedge x_2) \vee (x_3 \wedge x_4)$$

$$\rightarrow \mathbb{N}[X]\text{-provenance:}$$

$$(x_1 \times x_2) + (x_3 \times x_4) + (x_4 \times x_3)$$

- **Definition** of provenance for conjunctive queries:
 - **Sum** over query matches
 - **Multiply** over matched facts

$\mathbb{N}[X]$ -provenance example

R		
a	b	x_1
b	c	x_2
d	e	x_3
e	d	x_4
f	f	x_5

$$\exists x y z R(x, y) \wedge R(y, z)$$

→ PosBool[X]-provenance:

$$(x_1 \wedge x_2) \vee (x_3 \wedge x_4) \quad \vee \quad x_5$$

→ $\mathbb{N}[X]$ -provenance:

$$(x_1 \times x_2) + (x_3 \times x_4) + (x_4 \times x_3) + (x_5 \times x_5)$$

- **Definition** of provenance for conjunctive queries:
 - **Sum** over query matches
 - **Multiply** over matched facts

$\mathbb{N}[X]$ -provenance example

R		
a	b	x_1
b	c	x_2
d	e	x_3
e	d	x_4
f	f	x_5

$$\exists x y z R(x, y) \wedge R(y, z)$$

→ PosBool[X]-provenance:

$$(x_1 \wedge x_2) \vee (x_3 \wedge x_4) \quad \vee \quad x_5$$

→ $\mathbb{N}[X]$ -provenance:

$$(x_1 \times x_2) + (x_3 \times x_4) + (x_4 \times x_3) + (x_5 \times x_5)$$

- **Definition** of provenance for conjunctive queries:
 - **Sum** over query matches
 - **Multiply** over matched facts

$\mathbb{N}[X]$ -provenance example

R			$\exists x y z R(x, y) \wedge R(y, z)$	
a	b	x_1	$\rightarrow$ PosBool[X]-provenance:	
b	c	x_2	$(x_1 \wedge x_2) \vee (x_3 \wedge x_4)$	$\vee x_5$
d	e	x_3	$\rightarrow$ $\mathbb{N}[X]$ -provenance:	
e	d	x_4	$(x_1 \times x_2) + (x_3 \times x_4) + (x_4 \times x_3) + (x_5 \times x_5)$	
f	f	x_5	$= x_1 x_2 + 2x_3 x_4 + x_5^2$	

- **Definition** of provenance for conjunctive queries:
 - **Sum** over query matches
 - **Multiply** over matched facts

$\mathbb{N}[X]$ -provenance example

R			$\exists x y z R(x, y) \wedge R(y, z)$	
a	b	x_1	$\rightarrow$ PosBool[X]-provenance:	
b	c	x_2	$(x_1 \wedge x_2) \vee (x_3 \wedge x_4)$	$\vee x_5$
d	e	x_3	$\rightarrow$ $\mathbb{N}[X]$ -provenance:	
e	d	x_4	$(x_1 \times x_2) + (x_3 \times x_4) + (x_4 \times x_3) + (x_5 \times x_5)$	
f	f	x_5	$= x_1 x_2 + 2 x_3 x_4 + x_5^2$	

- **Definition** of provenance for conjunctive queries:
 - **Sum** over query matches
 - **Multiply** over matched facts

How is $\mathbb{N}[X]$ **more expressive** than PosBool[X]?

- $\rightarrow$ **Coefficients**: counting multiple matches
- $\rightarrow$ **Exponents**: using facts multiple times

Capturing $\mathbb{N}[X]$ -provenance

Our construction can be extended to $\mathbb{N}[X]$ -provenance
for conjunctive queries and **unions of conjunctive queries** (UCQ):

Capturing $\mathbb{N}[X]$ -provenance

Our construction can be extended to $\mathbb{N}[X]$ -provenance for conjunctive queries and unions of conjunctive queries (UCQ):

Théorème

For any fixed UCQ q and $k \in \mathbb{N}$,
for any input instance I of $\text{treewidth} \leq k$,
we can build in linear time a $\mathbb{N}[X]$ -provenance circuit of q on I .

Capturing $\mathbb{N}[X]$ -provenance

Our construction can be extended to $\mathbb{N}[X]$ -provenance for conjunctive queries and unions of conjunctive queries (UCQ):

Théorème

For any fixed UCQ q and $k \in \mathbb{N}$,
for any input instance I of $\text{treewidth} \leq k$,
we can build in linear time a $\mathbb{N}[X]$ -provenance circuit of q on I .

→ What fails for MSO and Datalog?

- Unbounded maximal multiplicity of fact uses

Correlations

- Our **probabilistic instances** assume **independence** on all facts
 - Not very **expressive**!

Correlations

- Our **probabilistic instances** assume **independence** on all facts
→ Not very **expressive**!

More expressive formalism: **Block-Independent Disjoint** instances:

<u>name</u>	city	iso	p
pods	san francisco	us	0.8
pods	los angeles	us	0.2
icalp	rome	it	0.1
icalp	florence	it	0.9

pc-tables

More generally, **pc-tables** to represent **arbitrary correlations**

pc-tables

More generally, **pc-tables** to represent **arbitrary correlations**

date	teacher	room	
04	John	C42	$\neg x_1$
04	Jane	C42	x_1
11	John	C017	$x_2 \wedge \neg x_1$
11	Jane	C017	$x_2 \wedge x_1$
11	John	C47	$\neg x_2 \wedge \neg x_1$
11	Jane	C47	$\neg x_2 \wedge x_1$

pc-tables

More generally, **pc-tables** to represent **arbitrary correlations**

date	teacher	room	
04	John	C42	$\neg x_1$
04	Jane	C42	x_1
11	John	C017	$x_2 \wedge \neg x_1$
11	Jane	C017	$x_2 \wedge x_1$
11	John	C47	$\neg x_2 \wedge \neg x_1$
11	Jane	C47	$\neg x_2 \wedge x_1$

x_1 John gets sick

→ **Probability** 0.1

x_2 Room C017 is available

→ **Probability** 0.2

Our results on correlations

Probabilistic query evaluation on instances with correlations is tractable if the **instance and correlations** are bounded-tw:

Our results on correlations

Probabilistic query evaluation on instances with correlations is tractable if the **instance and correlations** are bounded-tw:

Théorème

*Probabilistic query evaluation of MSO queries on treelike **BID** is in linear time up to arithmetic operations.*

Our results on correlations

Probabilistic query evaluation on instances with correlations is tractable if the **instance and correlations** are bounded-tw:

Théorème

*Probabilistic query evaluation of MSO queries on treelike **BID** is in linear time up to arithmetic operations.*

“Tree-like” just means the **underlying instance** (easy correlations)

Our results on correlations

Probabilistic query evaluation on instances with correlations is tractable if the **instance and correlations** are bounded-tw:

Théorème

*Probabilistic query evaluation of MSO queries on treelike **BID** is in linear time up to arithmetic operations.*

“Tree-like” just means the **underlying instance** (easy correlations)

Théorème

Probabilistic query evaluation of MSO queries on treelike pc-tables is in linear time up to arithmetic operations.

Our results on correlations

Probabilistic query evaluation on instances with correlations is tractable if the **instance and correlations** are bounded-tw:

Théorème

*Probabilistic query evaluation of MSO queries on treelike **BID** is in linear time up to arithmetic operations.*

“Tree-like” just means the **underlying instance** (easy correlations)

Théorème

Probabilistic query evaluation of MSO queries on treelike pc-tables is in linear time up to arithmetic operations.

“Tree-like” refers to the **underlying instance**, adding facts to represent variable **occurrences** and **co-occurrences**