Constant delay enumeration for FO queries and nowhere dense graphs

PART 2

Alexandre Vigny[1]

Join work with: Nicole Schweikardt[2] and Luc Segoufin[3]

[1]Université Paris Diderot, Paris

[2]Humboldt-Universität zu Berlin

[3]ENS Ulm, Paris

March 15, 2018

Outline

1. Part 1: On nowhere dense graphs
2. Part 2: Sketch of proof

Definitions nowhere dense

There are many equivalent definitions:
Winning strategies, asymptotic ratio edges/vertices, good ordering...

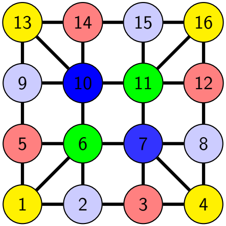
Definition : nowhere dense

$\mathscr{C}$ is *nowhere dense* if and only if for all $\epsilon > 0$, there is a $N_\epsilon \in \mathbb{N}$, such that for all $G \in \mathscr{C}$, G with $|G| > N_\epsilon$ admit a *k-tree width colouring* using $|G|^\epsilon$ colors.

tree-width colouring

Definition : k-tree width colouring with M colors

G is k-tree width coloured if and only if it is coloured (with less than M colors) and for all $H \subseteq G$, if H use less than k colours, then tree-width$(H) \leq k$

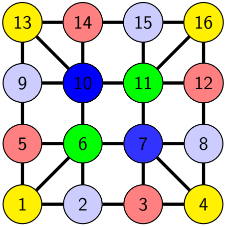


4-tree widh colouring with 5 colors

tree-width colouring

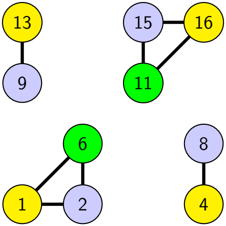
Definition : k-tree width colouring with M colors

G is k-tree width coloured if and only if it is coloured (with less than M colors) and for all $H \subseteq G$, if H use less than k colours, then tree-width$(H) \leq k$



4-tree widh colouring with 5 colors

$\Longrightarrow$



choosing 3 colours

Neighborhood cover

A neighborhood cover is a set of “representative” neighborhoods.

$\mathscr{X} := X_1, \ldots, X_n$ is a $(r, 2r)$ neighborhood cover if it has the following properties:

- $\forall a \in G, \quad \exists X \in \mathscr{X}, \quad N_r(a) \subseteq X$
- $\forall X \in \mathscr{X}, \quad \exists a \in G, \quad X \subseteq N_{2r}(a)$
- $\forall a \in G, \quad |\{i \mid a \in X_i\}|$ is pseudo-constant (smaller than $|G|^\epsilon$)

Examples

Over trees, $(r,2r)$-neighborhood cover with constant degree can easily be computed.

Over graphs with bounded degree, $(r,2r)$-neighborhood cover with constant degree can easily be computed.

The game characterization

Definition : (ℓ, r)-Splitter game

A graph G and two players, Splitter and Connector.
Each turn:

- Connector picks a node c
- Splitter picks a node s
- $G' = N_r^G(c) \setminus s$

If in less than ℓ rounds the graph is empty, Splitter wins.

The game characterization

Definition : (ℓ, r)-Splitter game

A graph G and two players, Splitter and Connector.
Each turn:

- Connector picks a node c
- Splitter picks a node s
- $G' = N_r^G(c) \setminus s$

If in less than ℓ rounds the graph is empty, Splitter wins.

Theorem

$\mathscr{C}$ is nowhere dense if and only if there is a function $f_{\mathscr{C}}$ such that for every $G \in \mathscr{C}$ and every $r \in \mathbb{N}$:

Splitter has a wining strategy for the $(f_{\mathscr{C}}(r), r)$-splitter game on G.

How to play the (ℓ, r)-Splitter game on a graph G ?

- If G is a star, Splitter wins in 2 rounds.
- If G is a path, Splitter wins in $\log(r)$ rounds.
- If G is a tree, Splitter wins in r rounds.
- If G has degree d, splitter wins in d^r rounds.
- If G is a clique of size $> \ell$, Splitter looses the (ℓ, r)-splitter game.

Outline

The examples queries

- $q_1(x, y) := \exists z \; E(x, z) \wedge E(z, y)$

  (The distance two query)

- $q_2(x, y) := \neg q_1(x, y)$

  (Nodes that are far apart)

How to use the game 1/2

G is now fixed

Goal : Given a node a we want to enumerate all b such that $q_1(a, b)$.
(Here $r = 2$)

- Base case: If Splitter wins the $(1, r)$-Splitter game on G.

  Then G is edgeless and there is no solution !

- By induction: assume that there is an algorithm for every G' such that Splitter wins the (ℓ, r)-Splitter game on G'.

How to use the game 2/2

Here, Splitter wins the $(\ell + 1, r)$-game on G.

Idea :

1. Compute some new graphs on which Splitter wins the (ℓ, r) game.
2. Apply the induce algorithm for a particular query.
3. Enumerate those solutions.

How to use the game 2/2

Here, Splitter wins the $(\ell+1, r)$-game on G.

Idea :

1. Compute some new graphs on which Splitter wins the (ℓ, r) game.
2. Apply the induce algorithm for a particular query.
3. Enumerate those solutions.

For every bags X of the (2,4)-neighborhood cover, $X' := X \setminus \{s\}$.

For every $(a,b) \in G^2$ we have:

$$G \models q_1(a,b) \iff \bigvee_{X \in \mathscr{X}} X \models q_1(a,b) \iff \mathscr{X}(a) \models q_1(a,b)$$

How to use the game 2/2

Here, Splitter wins the $(\ell+1, r)$-game on G.

Idea :

1. Compute some new graphs on which Splitter wins the (ℓ, r) game.
2. Apply the induce algorithm for a particular query.
3. Enumerate those solutions.

For every bags X of the $(2,4)$-neighborhood cover, $X' := X \setminus \{s\}$.

For every $(a,b) \in G^2$ we have:

$$G \models q_1(a,b) \iff \bigvee_{X \in \mathscr{X}} X \models q_1(a,b) \iff \mathscr{X}(a) \models q_1(a,b)$$

$X = \mathscr{X}(a) \models q_1(a,b)$ iff:

- $X' \models q_1(a,b)$
- $E(a,s) \wedge E(s,b)$
- $b = s$ and $X \models q_1(a,s)$
- $a = s$ and $X \models q_1(s,b)$

The second query

$q_2(x,y) := \mathrm{dist}(x,y) > 2$

Two kinds of solutions:

- $b \in \mathcal{X}(a)$ (similar to the previous example)
- $b \notin \mathcal{X}(a)$ We need something else !

The second query

$q_2(x,y) := \mathrm{dist}(x,y) > 2$

Two kinds of solutions:

- $b \in \mathscr{X}(a)$ (similar to the previous example)
- $b \notin \mathscr{X}(a)$ We need something else !

Goal: given a bag X, enumerate all $b \notin X$

The shortcut pointers

Given X we want to enumerate all b such that $b \not\in X$.

The shortcut pointers

Given X we want to enumerate all b such that $b \notin X$.

$$\underset{b \in X}{NEXT(b, X)} := \min\{b' \in G \mid b' \geq b \ \wedge \ b' \notin X\}$$

The shortcut pointers

Given X we want to enumerate all b such that $b \notin X$.

$$\underset{b \in X}{NEXT(b,X)} := \min\{b' \in G \mid b' \geq b \ \wedge \ b' \notin X\}$$

For all $X \in \mathscr{X}$ with $b_{max} \in X$, we have $NEXT(b_{max}, X) \ = \ NULL$

The shortcut pointers

Given X we want to enumerate all b such that $b \notin X$.

$$\underset{b \in X}{NEXT(b,X)} := \min\{b' \in G \mid b' \geq b \ \wedge \ b' \notin X\}$$

For all $X \in \mathscr{X}$ with $b_{max} \in X$, we have $NEXT(b_{max}, X) \ = \ NULL$

$$NEXT(b,X) \in \{b+1 \ , \ NEXT(b+1,X)\}$$

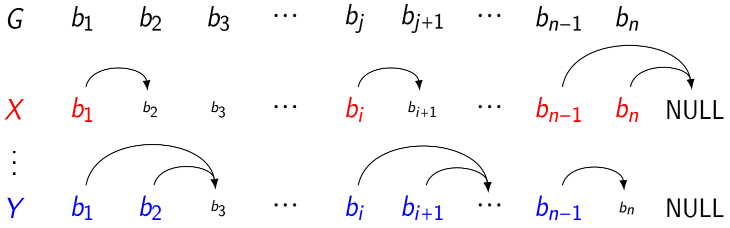
The shortcut pointers

Given X we want to enumerate all b such that $b \notin X$.

$$\underset{b \in X}{NEXT(b,X)} := \min\{b' \in G \mid b' \geq b \ \wedge \ b' \notin X\}$$

For all $X \in \mathscr{X}$ with $b_{max} \in X$, we have $NEXT(b_{max}, X) = NULL$

$$NEXT(b,X) \in \{b+1 \ , \ NEXT(b+1,X)\}$$



Recap

We use :

- A new "Hanf like" normal form for FO queries.[1]
- The algorithm for the model checking.[2]
- Neighbourhood cover.[2]
- Game characterization of Nowhere-Dense classes.[2]
- Short-cut pointers dedicated to the enumeration.[3]

We can :

- Enumerate with constant delay after pseudo-linear preprocessing.
- Test in constant time after pseudo-linear preprocessing.

---

[1] Grohe, Schweikardt '17
[2] Grohe, Kreutzer, Siebertz '14
[3] Segoufin, V. '17

Future work

- Classes of graphs that are not closed under subgraphs
- Enumeration with update:
  What happens if a small change occurs after the preprocessing ?

  Existing results for: words, graphs with bounded tree-width or bounded degree.

Future work

- Classes of graphs that are not closed under subgraphs
- Enumeration with update:
  What happens if a small change occurs after the preprocessing ?

  Existing results for: words, graphs with bounded tree-width or bounded degree.

Thank you !

Questions ?