Algorithmic metatheorems: A survey

Abhisekh Sankaran
IMSc, Chennai

Formal Methods Update Meet

BITS Goa

July 21, 2018

The model checking problem

Question ($\mathcal{MC}(\mathcal{C}, \mathcal{L})$)

Let $\mathcal{C}$ be a class of finite structures and $\mathcal{L}$ a logic like FO, MSO, etc. Given a structure $\mathcal{A} \in \mathcal{C}$ and a sentence $\varphi \in \mathcal{L}$, check (algorithmically) if $\mathcal{A} \models \varphi$.

The model checking problem

Question ($\mathcal{MC}(\mathcal{C}, \mathcal{L})$)

Let $\mathcal{C}$ be a class of finite structures and $\mathcal{L}$ a logic like FO, MSO, etc. Given a structure $\mathcal{A} \in \mathcal{C}$ and a sentence $\varphi \in \mathcal{L}$, check (algorithmically) if $\mathcal{A} \models \varphi$.

- Decidable –
  - FO: $O(n^{|\varphi|})$ where $n = |\mathcal{A}|$
  - MSO: $O(2^{n \cdot |\varphi|})$

The model checking problem

Question ($\mathcal{MC}(\mathcal{C}, \mathcal{L})$)

Let $\mathcal{C}$ be a class of finite structures and $\mathcal{L}$ a logic like FO, MSO, etc. Given a structure $\mathcal{A} \in \mathcal{C}$ and a sentence $\varphi \in \mathcal{L}$, check (algorithmically) if $\mathcal{A} \models \varphi$.

- Decidable –
  - FO: $O(n^{|\varphi|})$ where $n = |\mathcal{A}|$
  - MSO: $O(2^{n \cdot |\varphi|})$
- The above is essentially the best we know so far: model checking FO/MSO over the set-structure $\mathcal{A} = (\{1, 2\})$ is already PSPACE-complete (reduction from QBF).

A parameterized view of $\mathcal{MC}(\mathcal{C}, \mathcal{L})$

- A fixed parameter tractable (FPT) algorithm for a problem having parameters $k_1, \ldots, k_r$ etc. is an algorithm solving the problem in time $f(k_1, \ldots, k_r) \cdot n^{O(1)}$ where n is the size of the input and $f : \mathbb{N}^r \to \mathbb{N}$ is a computable function.
- E.g. Minimum vertex cover is NP-complete but has an FPT algorithm.

Question ($\mathcal{MC}(\mathcal{C}, \mathcal{L})$)

Let $\mathcal{C}$ be a class of finite structures and $\mathcal{L}$ a logic like FO, MSO, etc. Given a structure $\mathcal{A} \in \mathcal{C}$ and a sentence $\varphi \in \mathcal{L}$, is there an FPT algorithm for $\mathcal{MC}(\mathcal{C}, \mathcal{L})$ where $|\varphi|$ is the parameter?

A parameterized view of $\mathcal{MC}(\mathcal{C}, \mathcal{L})$

- A fixed parameter tractable (FPT) algorithm for a problem having parameters $k_1, \ldots, k_r$ etc. is an algorithm solving the problem in time $f(k_1, \ldots, k_r) \cdot n^{O(1)}$ where n is the size of the input and $f : \mathbb{N}^r \to \mathbb{N}$ is a computable function.
- E.g. Minimum vertex cover is NP-complete but has an FPT algorithm.

Question ($\mathcal{MC}(\mathcal{C}, \mathcal{L})$)

Let $\mathcal{C}$ be a class of finite structures and $\mathcal{L}$ a logic like FO, MSO, etc. Given a structure $\mathcal{A} \in \mathcal{C}$ and a sentence $\varphi \in \mathcal{L}$, is there an FPT algorithm for $\mathcal{MC}(\mathcal{C}, \mathcal{L})$ where $|\varphi|$ and some structural aspects of the structures of $\mathcal{C}$, are parameters?

A parameterized view of $\mathcal{MC}(\mathcal{C}, \mathcal{L})$

- A fixed parameter tractable (FPT) algorithm for a problem having parameters $k_1, \ldots, k_r$ etc. is an algorithm solving the problem in time $f(k_1, \ldots, k_r) \cdot n^{O(1)}$ where n is the size of the input and $f : \mathbb{N}^r \to \mathbb{N}$ is a computable function.
- E.g. Minimum vertex cover is NP-complete but has an FPT algorithm.

Question (Algorithmic metatheorem for $\mathcal{MC}(\mathcal{C}, \mathcal{L})$)

Let $\mathcal{C}$ be a class of finite structures and $\mathcal{L}$ a logic like $\mathrm{FO}, \mathrm{MSO}$, etc. Given a structure $\mathcal{A} \in \mathcal{C}$ and a sentence $\varphi \in \mathcal{L}$, is there an FPT algorithm for $\mathcal{MC}(\mathcal{C}, \mathcal{L})$ where $|\varphi|$ and some structural parameters of the structures of $\mathcal{C}$, are parameters?

Talk outline

We look at various classes of structures that admit algorithmic metatheorems for $\mathcal{MC}(\mathcal{C}, \mathcal{L})$ and give the intuitive ideas for the techniques used.

Structures	**Techniques**
Posets	Automata
Graphs	Feferman-Vaught composition Locality of FO

Structures: Posets

Technique: Automata

The case of words

Let Binary-words $=$ words over $\{0, 1\}$ and $\mathcal{L} = $ MSO.

$\mathcal{MC}$(Binary-words, MSO): For $w \in$ Binary-words and $\varphi \in \mathcal{L}$,
check if $w \models \varphi$.

The case of words

Let Binary-words $=$ words over $\{0, 1\}$ and $\mathcal{L} = $ MSO.
$\mathcal{MC}$(Binary-words, MSO): For $w \in$ Binary-words and $\varphi \in \mathcal{L}$, check if $w \models \varphi$.

- By the Büchi-Elgot-Trakhtenbrot theorem, there is an algorithm translating φ into a finite automaton $\mathcal{A}$ such that $L(\mathcal{A}) =$ word models of φ.
- Model checking algorithm: Convert φ into $\mathcal{A}$ and run $\mathcal{A}$ on w!
- Running time: $f(\varphi) + |w|$ where $f(\varphi)$ is the (computable) time taken in converting φ to $\mathcal{A}$.

The case of words

Let Binary-words $=$ words over $\{0, 1\}$ and $\mathcal{L} = $ MSO.
$\mathcal{MC}$(Binary-words, MSO): For $w \in$ Binary-words and $\varphi \in \mathcal{L}$, check if $w \models \varphi$.

Theorem. (Büchi-Elgot-Traktenbrot, 1960s [3])

The problem $\mathcal{MC}$(Binary-words, MSO) has a linear time FPT algorithm with the size of the MSO sentence as the parameter.

The case of words

Let Binary-words $=$ words over $\{0, 1\}$ and $\mathcal{L} = $ MSO.
$\mathcal{MC}$(Binary-words, MSO): For $w \in$ Binary-words and $\varphi \in \mathcal{L}$, check if $w \models \varphi$.

Theorem. (Büchi-Elgot-Traktenbrot, 1960s [3]; Frick-Grohe, 2004 [8])

The problem $\mathcal{MC}$(Binary-words, MSO) has a linear time FPT algorithm with the size of the MSO sentence as the parameter.
The dependence on the parameter is non-elementary.

Trees, nested words and traces

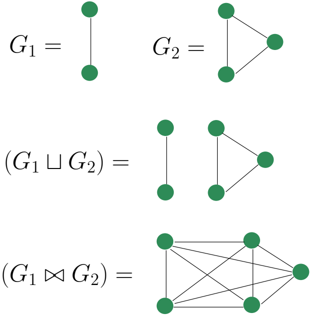
- Similar linear time algorithmic metatheorems for ordered/ranked trees, nested words and traces.
- This is because there exists a Büchi-Elgot-Traktenbrot-like theorem for
  - ordered/ranked trees: in terms of tree automata (Rabin, 1967 [12])
  - nested words: in terms of nested word automata (Alur-Madhusudan, 2009 [1])
  - traces: in terms of Zielonka automata (Zielonka, 1987 [14])
- The parameter dependence is again non-elementary as words are special cases of the above structures.

Structures: Graphs

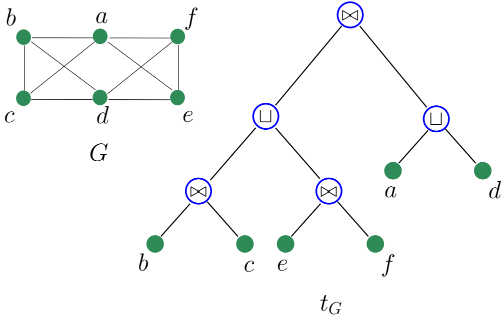
Technique: Feferman-Vaught composition

Cographs

Generated from point graphs using disjoint union and join.



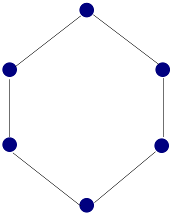
disjoint union $= \sqcup$; join $= \bowtie$



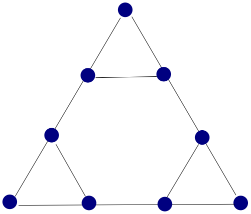
Cograph G and its cotree t_G

MSO$[m]$-similarity of graphs

- We say graphs G and H are MSO$[m]$-similar, denoted $G \equiv_m H$, if no MSO sentence having m quantifiers distinguishes G and H.



$\equiv_2$



$\mathcal{A}$ $\mathcal{B}$

MSO$[m]$-similarity of graphs

- We say graphs G and H are MSO$[m]$-similar, denoted $G \equiv_m H$, if no MSO sentence having m quantifiers distinguishes G and H.
- Observe that $\equiv_m$ is an equivalence relation. Each equivalence class can be represented as the set of MSO$[m]$ sentences true over the (structures of the) class.

Fact.

The set Δ_m of equivalence classes of the MSO$[m]$-similarity relation is finite. Further, there is a computable function $\Lambda : \mathbb{N} \to \mathbb{N}$ such that $|\Delta_m| \leq \Lambda(m)$.

A model-theoretic property of $\sqcup$ and $\bowtie$

Fact.

Each of $\sqcup$ and $\bowtie$ satisfies a Feferman-Vaught kind composition property.

$$\begin{tikzcd}
G_1 \arrow[d, leftrightarrow, "\equiv_m"] & H_1 \arrow[d, leftrightarrow, "\equiv_m"] \arrow[r] & G_1 \sqcup H_1 \arrow[d, leftrightarrow, "\equiv_m"] & G_1 \bowtie H_1 \arrow[d, leftrightarrow, "\equiv_m"] \\
G_2 & H_2 \arrow[r] & G_2 \sqcup H_2 & G_2 \bowtie H_2
\end{tikzcd}$$

A model-theoretic property of $\sqcup$ and $\bowtie$

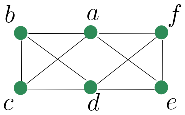
Fact.

Feferman-Vaught kind composition property of $\sqcup$ and $\bowtie$:
There exist composition functions $f_m, g_m : (\Delta_m \times \Delta_m) \rightarrow \Delta_m$ such that if $\delta_m(G)$ is the MSO$[m]$-similarity class of G, then

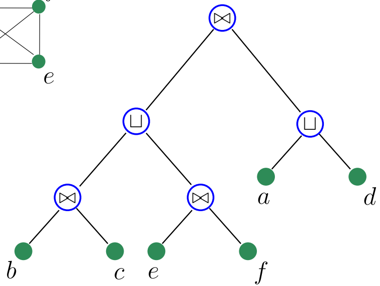
$$\begin{aligned} \delta_m(G_1 \sqcup G_2) &= f_m\big(\delta_m(G_1), \delta_m(G_2)\big) \\ \delta_m(G_1 \bowtie G_2) &= g_m\big(\delta_m(G_1), \delta_m(G_2)\big) \end{aligned}$$

Model checking MSO on cographs

Label bottom up in the cotree, each node z with the MSO$[m]$-similarity class of the graph represented by the tree rooted at z.



G

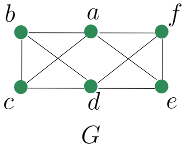


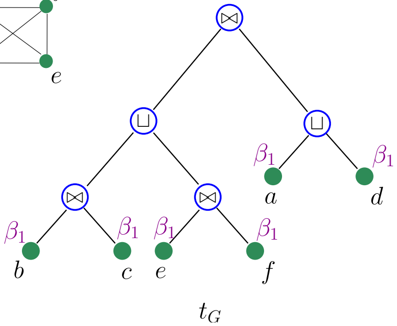
t_G

Model checking MSO on cographs

Label bottom up in the cotree, each node z with the MSO$[m]$-similarity class of the graph represented by the tree rooted at z.

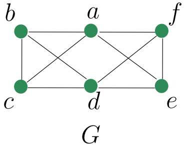
$\beta_1 = \delta_m(\bullet)$

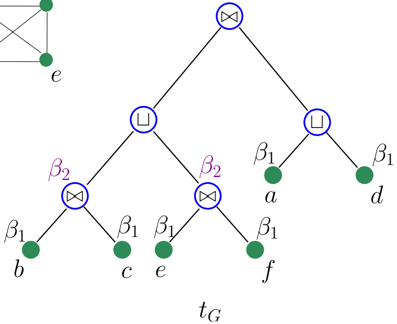




Model checking MSO on cographs

Label bottom up in the cotree, each node z with the MSO$[m]$-similarity class of the graph represented by the tree rooted at z.



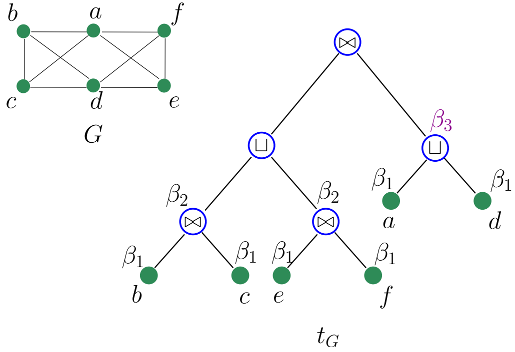


$\beta_1 = \delta_m(\bullet)$

$\beta_2 = g_m(\beta_1, \beta_1) = \delta_m(\bullet\!-\!\!-\!\bullet)$

Model checking MSO on cographs

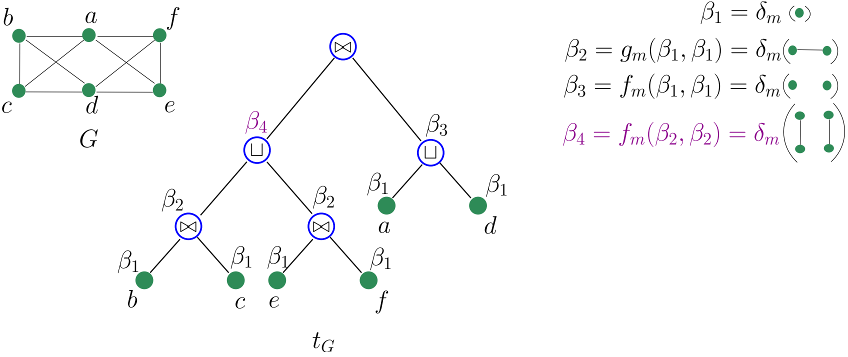
Label bottom up in the cotree, each node z with the MSO$[m]$-similarity class of the graph represented by the tree rooted at z.



$$\beta_1 = \delta_m(\bullet)$$
$$\beta_2 = g_m(\beta_1, \beta_1) = \delta_m(\bullet\!\!-\!\!\bullet)$$
$$\beta_3 = f_m(\beta_1, \beta_1) = \delta_m(\bullet \quad \bullet)$$

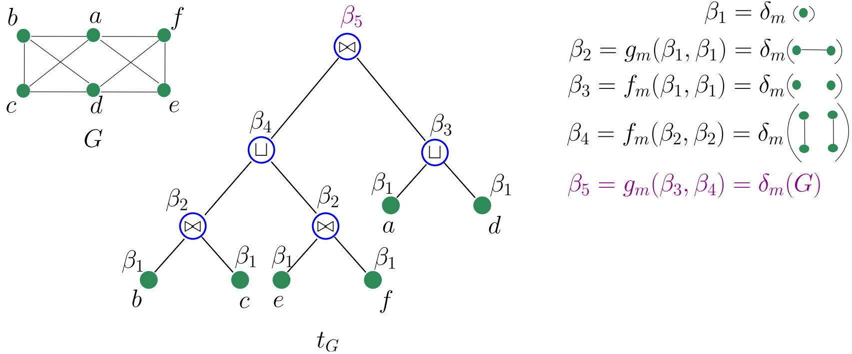
Model checking MSO on cographs

Label bottom up in the cotree, each node z with the MSO$[m]$-similarity class of the graph represented by the tree rooted at z.



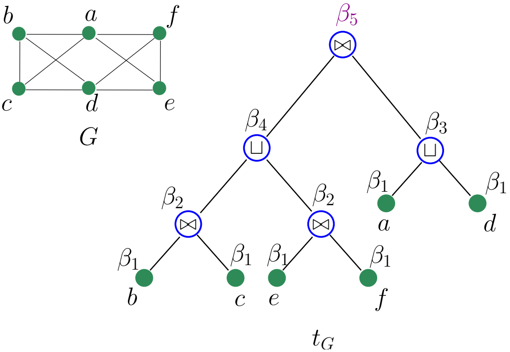
Model checking MSO on cographs

Label bottom up in the cotree, each node z with the MSO$[m]$-similarity class of the graph represented by the tree rooted at z.



Model checking MSO on cographs

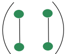
Label bottom up in the cotree, each node z with the MSO$[m]$-similarity class of the graph represented by the tree rooted at z.



$\beta_1 = \delta_m$ ()

$\beta_2 = g_m(\beta_1, \beta_1) = \delta_m$ ()

$\beta_3 = f_m(\beta_1, \beta_1) = \delta_m$ ()

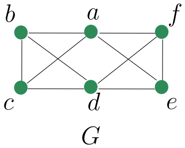
$\beta_4 = f_m(\beta_2, \beta_2) = \delta_m$ ()

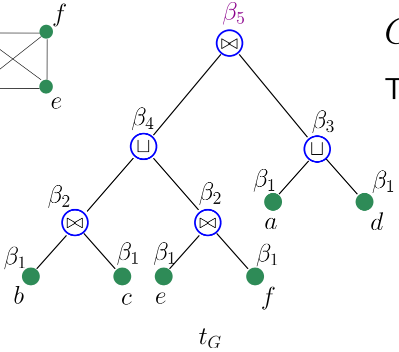
$\beta_5 = g_m(\beta_3, \beta_4) = \delta_m(G)$

$$G \models \varphi \Leftrightarrow \varphi \in \delta_m(G)$$

Model checking MSO on cographs

Label bottom up in the cotree, each node z with the MSO$[m]$-similarity class of the graph represented by the tree rooted at z.





$$G \models \varphi \Leftrightarrow \varphi \in \delta_m(G)$$

Total time taken $=$

time to compute composition functions

$+$

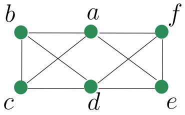
time for labeling t_G

$+$

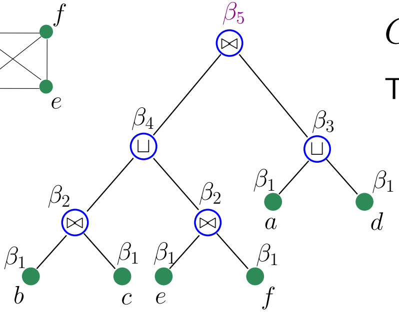
time to check $\varphi \in \delta_m(G)$

Model checking MSO on cographs

Label bottom up in the cotree, each node z with the MSO$[m]$-similarity class of the graph represented by the tree rooted at z.



G



t_G

$$G \models \varphi \Leftrightarrow \varphi \in \delta_m(G)$$

Total time taken =

time to compute composition functions

$+$

time for labeling t_G

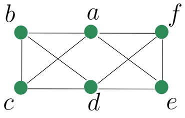
$+$

time to check $\varphi \in \delta_m(G)$

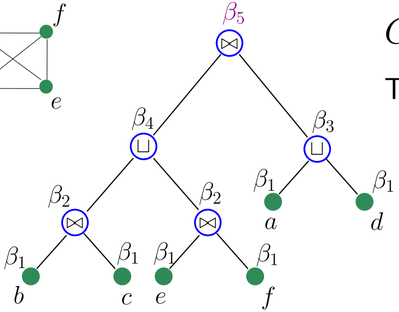
$$\leq f(\varphi) + g(\varphi) \cdot |t_G|$$

Model checking MSO on cographs

Label bottom up in the cotree, each node z with the MSO$[m]$-similarity class of the graph represented by the tree rooted at z.



G



t_G

$$G \models \varphi \Leftrightarrow \varphi \in \delta_m(G)$$

Total time taken $=$

time to compute composition functions

$+$

time for labeling t_G

$+$

time to check $\varphi \in \delta_m(G)$

$$\leq f(\varphi) + g(\varphi) \cdot |G|$$

Graphs of bounded tree-width or clique-width

- These graphs admit tree representations in terms of operations that satisfy the Feferman-Vaught composition property.
- The tree representation of such G is obtainable in polytime:
  - Tree-width $\leq k$: tree-decomposition of width k in time $O(2^k \cdot n)$ where $n = |G|$ (Bodlaender, 1996 [2])
  - Clique-width $\leq k$: A $(2^{3k+2} - 1)$-expression in time $O(f(k) \cdot n^9 \log n)$ (Oum-Seymour, 2006 [11])
- By exactly the same reasoning as for cographs, we obtain the following algorithmic metatheorems for bounded tree-width/ clique-width graphs.

Graphs of bounded tree-width or clique-width

Theorem. (Courcelle, 1990 [4]; Frick-Grohe, 2004 [8])

Let $\mathcal{T}_k$ be the class of all graphs of tree-width $\leq k$. Then $\mathcal{MC}(\mathcal{T}_k, \mathsf{MSO})$ has a linear time FPT algorithm with the size of the MSO sentence as the parameter. The parameter dependence is non-elementary.

Theorem. (Courcelle-Makowsky-Rotics, 2000 [5]; Frick-Grohe, 2004 [8]; Oum-Seymour, 2006 [11])

Let $\mathcal{C}_k$ be the class of all graphs of clique-width $\leq k$. Then $\mathcal{MC}(\mathcal{C}_k, \mathsf{MSO})$ has an FPT algorithm with the size of the MSO sentence as the parameter. The parameter dependence is non-elementary.

Structures: Graphs

Technique: FO Locality

Locality of FO

Define a local FO sentence as one that asserts for some $r, d \in \mathbb{N}$ and some FO sentence φ that
"There exist r vertices that are pairwise $> 2d$ distance apart and whose d-neighborhoods satisfy φ"

Locality of FO

Define a local FO sentence as one that asserts for some $r, d \in \mathbb{N}$ and some FO sentence φ that

"There is a d-scattered set consisting of r vertices each of whose d-neighborhoods satisfies φ"

Locality of FO

Define a local FO sentence as one that asserts for some $r, d \in \mathbb{N}$ and some FO sentence φ that
"There is a d-scattered set consisting of r vertices each of whose d-neighborhoods satisfies φ"

Theorem. (Gaifman, 1982 [9])

Every FO sentence is equivalent to a Boolean combination of local sentences.

- The "Gaifman normal form" is computable.
- The size of the GNF of φ is a non-elementary function of $|\varphi|$ (Dawar-Grohe-Kreutzer-Schweikardt, 1997 [6]).

The case of bounded degree graphs

Evaluating local sentences

Let G have degree $\leq k$ and ψ be the local sentence asserting

"There exists a d-scattered set of r vertices each of whose d-neighborhoods satisfy φ"

Evaluating local sentences

Let G have degree $\leq k$ and ψ be the local sentence asserting "There exists a d-scattered set of r vertices amenable for φ"

Evaluating local sentences

Let G have degree $\leq k$ and ψ be the local sentence asserting
"There exists a d-scattered set of r vertices amenable for φ"

Labeling G using φ:

Evaluating local sentences

Let G have degree $\leq k$ and ψ be the local sentence asserting "There exists a d-scattered set of r vertices amenable for φ"

Labeling G using φ:

- Label a vertex x of G with 1 if its d-neighborhood $N_d(x)$ models φ, else label with 0.

Evaluating local sentences

Let G have degree $\leq k$ and ψ be the local sentence asserting
"There exists a d-scattered set of r vertices amenable for φ"

Labeling G using φ:

- Label a vertex x of G with 1 if its d-neighborhood $N_d(x)$ models φ, else label with 0.
- deg$(G) \leq k \Rightarrow |N_d(x)| \leq k^{d+1}$ = constant, say c

Evaluating local sentences

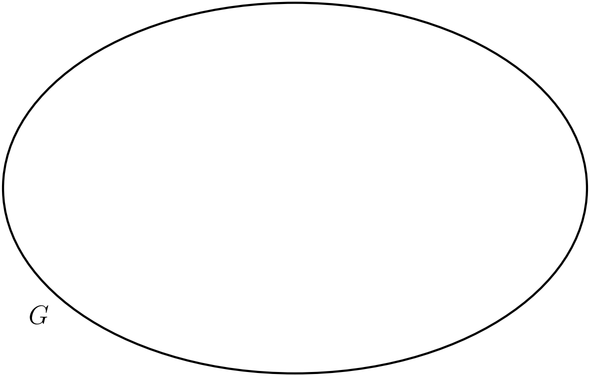
Let G have degree $\leq k$ and ψ be the local sentence asserting "There exists a d-scattered set of r vertices amenable for φ"

Labeling G using φ:

- Label a vertex x of G with 1 if its d-neighborhood $N_d(x)$ models φ, else label with 0.
- $\deg(G) \leq k \Rightarrow |N_d(x)| \leq k^{d+1} =$ constant, say c
- Time taken to label $x = c^{|\varphi|}$. Then time to label all vertices of $G = c^{|\varphi|} \times n$ where $n = |G|$.

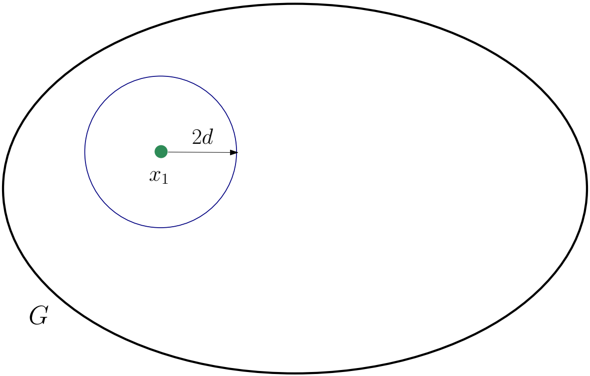
Evaluating local sentences (Contd.)

Finding a d-scattered set of r vertices amenable for φ:



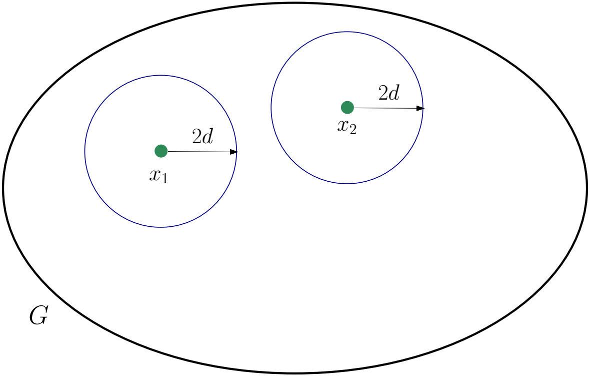
Evaluating local sentences (Contd.)

Finding a d-scattered set of r vertices amenable for φ:



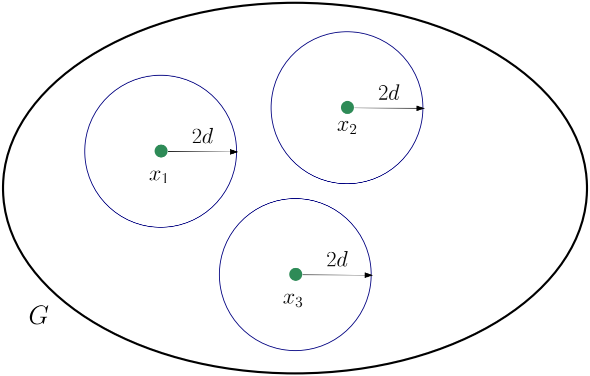
Evaluating local sentences (Contd.)

Finding a d-scattered set of r vertices amenable for φ:



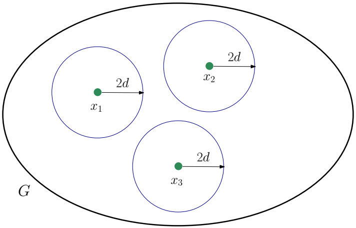
Evaluating local sentences (Contd.)

Finding a d-scattered set of r vertices amenable for φ:



Evaluating local sentences (Contd.)

Finding a d-scattered set of r vertices amenable for φ:



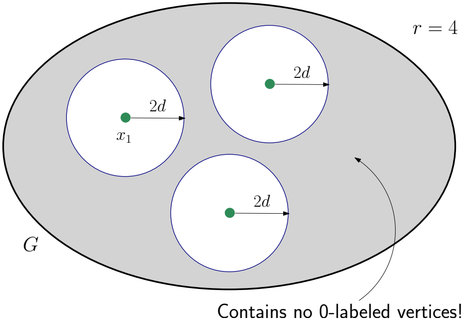
- $\mathcal{S} = \emptyset$
- While (G has a 1-labeled vertex) {
    - $x :=$ some 1-labeled vertex
    - $\mathcal{S} := \mathcal{S} \cup \{x\}$
    - $G := G \setminus N_{2d}(x)$

  }

Evaluating local sentences (Contd.)

Finding a d-scattered set of r vertices amenable for φ:

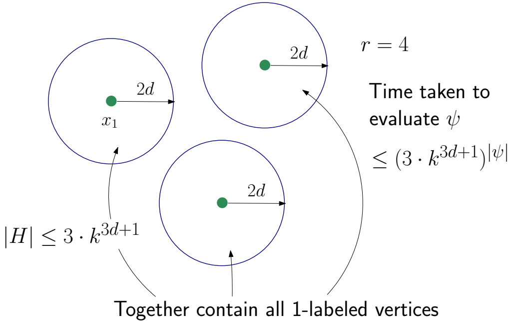
- If $|\mathcal{S}| \geq r$, return True.
- Else check (brute force) if $H = \bigcup_{x \in \mathcal{S}} \bigcup_{y \in \mathcal{Z}(x)} N_d(y)$ models the local sentence ψ where $\mathcal{Z}(x) =$ set of 1-labeled vertices in $N_{2d}(x)$, and return the answer.



Evaluating local sentences (Contd.)

Finding a d-scattered set of r vertices amenable for φ:

- If $|\mathcal{S}| \geq r$, return True.
- Else check (brute force) if $H = \bigcup_{x \in \mathcal{S}} \bigcup_{y \in \mathcal{Z}(x)} N_d(y)$ models the local sentence ψ where $\mathcal{Z}(x)$ = set of 1-labeled vertices in $N_{2d}(x)$, and return the answer.



Model checking FO on bounded degree graphs

- Recall: Converting an FO sentence into its Gaifman normal form is computable.
- Recall: Each local sentence can be evaluated in FPT linear time on a bounded degree graph.
- Evaluating a Boolean combination of truth values is linear in the number of truth values.

Theorem. (Seese, 1996 [13]; Frick-Grohe, 2004 [8])

Let $\mathcal{Z}_k$ be the class of graphs of degree $\leq k$. Then $\mathcal{MC}(\mathcal{Z}_k, \mathsf{FO})$ has a linear time FPT algorithm with the size of the FO sentence as the parameter. The parameter dependence is triply exponential.

Structures: Graphs

Technique: Combining FV-Composition and FO Locality

A closer look at an earlier argument

Let G have degree $\leq k$ and ψ be the local sentence asserting

"There exists a d-scattered set of r vertices amenable for φ"

Labeling G using φ:

- Label a vertex x of G with 1 if its d-neighborhood $N_d(x)$ models φ, else label with 0.
- $\deg(G) \leq k \Rightarrow |N_d(x)| \leq k^{d+1} =$ constant, c
- Time taken to label $x = c^{|\varphi|}$. Then time to label all vertices of $G = c^{\varphi|} \times n$ where $n = |G|$.

A closer look at an earlier argument

Let G have degree $\leq k$ and ψ be the local sentence asserting "There exists a d-scattered set of r vertices amenable for φ"

Labeling G using φ:

- Label a vertex x of G with 1 if its d-neighborhood $N_d(x)$ models φ, else label with 0.
- $\deg(G) \leq k \Rightarrow |N_d(x)| \leq k^{d+1} =$ constant, c
- Time taken to label $x = c^{|\varphi|}$. Then time to label all vertices of $G = c^{\varphi|} \times n$ where $n = |G|$.

Graph classes of bounded local tree-width

- These graph classes are locally tree-like: there is $f : \mathbb{N} \to \mathbb{N}$ such that for every d, every graph G in the class and every vertex x of G, tree-width of $N_d(x)$ is $\leq f(d)$.
- E.g. planar graphs: $f(d) \leq 3d$ (Robertson-Seymour, 1984); graphs embeddable on a surface of genus k: $f(d) \leq c \cdot k \cdot d$ (Eppstein, 1999).
- By combining the ideas of locality and FV-composition (that gave linear time FPT algorithms), we get the following.

Theorem. (Frick-Grohe, 2001 [7], 2004 [8])

Let $\mathcal{LT}$ be a class of bounded local tree-width. Then $\mathcal{MC}(\mathcal{LT}, \text{FO})$ has a linear time FPT algorithm with the parameter being the FO sentence size. The parameter dependence is non-elementary.

Algorithmic metatheorems for sparse graphs [10]

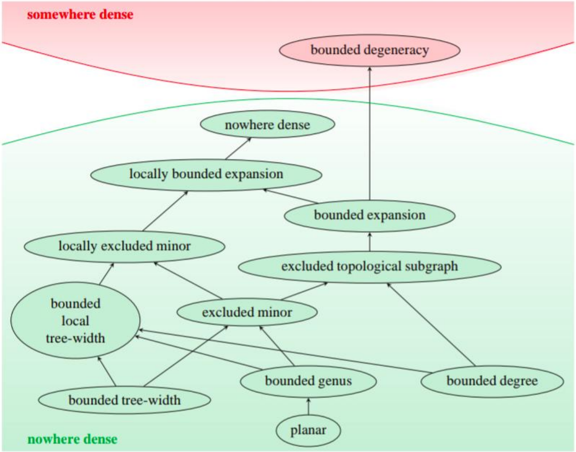


Figure 1: Sparse graph classes

Thank you!

References I

- Rajeev Alur and Parthasarathy Madhusudan. Adding nesting structure to words. *J. ACM*, 56(3), 2009.
- Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on computing*, 25(6):1305–1317, 1996.
- Julius R. Büchi and Calvin C. Elgot. Decision problems of weak second order arithmetics and finite automata, Part I. *Notices of the American Mathematical Society*, 5:834, 1958.
- Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990.
- Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150, 2000.

References II

Anuj Dawar, Martin Grohe, Stephan Kreutzer, and Nicole Schweikardt. Model theory makes formulas large. In *Proceedings of the International Colloquium on Automata, Languages and Programming, ICALP 2007, Wroclaw, Poland, July 9 – 13, 2007*, pages 913–924, 2007.

Markus Frick and Martin Grohe. Deciding first-order properties of locally tree-decomposable structures. *Journal of the ACM (JACM)*, 48(6):1184–1206, 2001.

Markus Frick and Martin Grohe. The complexity of first-order and monadic second-order logic revisited. *Ann. Pure Appl. Logic*, 130(1-3):3–31, 2004.

References III

Haim Gaifman. On local and non-local properties. In J. Stern, editor, *Proceedings of the Herbrand Symposium*, volume 107 of *Studies in Logic and the Foundations of Mathematics*, pages 105 – 135. Elsevier, 1982.

Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. *J. ACM*, 64(3):17:1–17:32, 2017.

Sang-il Oum and Paul Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Series B*, 96(4):514–528, 2006.

Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969.

References IV

- Detlef Seese. Linear time computable problems and first-order descriptions. *Math. Struct. Comp. Sci.*, 6:505–526, 1996.
- Wieslaw Zielonka. Notes on finite asynchronous automata. *RAIRO-Theoretical Informatics and Applications*, 21(2):99–135, 1987.