An optimal Gaifman normal form construction for structures of bounded degree

Lucas Heimberg[1] Dietrich Kuske[2] Nicole Schweikardt[1]

[1]Goethe-University Frankfurt am Main

[2]Ilmenau University of Technology

Highlights 2013

September 20

Gaifman's Theorem

Throughout this talk: signatures contain only relation symbols.

Theorem: *(Gaifman, 1981)*

Every FO-formula is equivalent to an FO-formula in Gaifman normal form.

And there is an algorithm which translates a given FO-formula into an equivalent formula in Gaifman normal form.

Reminder:

- Gaifman normal form:
  Boolean combination of local formulas
  and basic local sentences
- local formula:
  FO-formula $\varrho(\bar{y})$ in which every quantifier has the form
  $\exists x\, (\mathit{dist}(\bar{y}, x) \leqslant r \;\wedge\; \psi)$ or $\forall x\, (\mathit{dist}(\bar{y}, x) \leqslant r \;\rightarrow\; \psi)$.
- basic local sentence:

$$\exists x_1 \cdots \exists x_k \quad \left(\bigwedge_{1 \leqslant i < j \leqslant k} \mathit{dist}(x_i, x_j) > 2r \;\wedge\; \bigwedge_{i=1}^{k} \varrho(x_i) \right)$$

Gaifman's Theorem

Throughout this talk: signatures contain only relation symbols.

Theorem: *(Gaifman, 1981)*

Every FO-formula is equivalent to an FO-formula in Gaifman normal form.

And there is an algorithm which translates a given FO-formula into an equivalent formula in Gaifman normal form.

Reminder:

- Gaifman normal form:
  Boolean combination of local formulas
  and basic local sentences
- local formula:
  FO-formula $\varrho(\bar{y})$ in which every quantifier has the form
  $\exists x\, (\mathit{dist}(\bar{y}, x) \leqslant r \;\wedge\; \psi)$ or $\forall x\, (\mathit{dist}(\bar{y}, x) \leqslant r \;\rightarrow\; \psi)$.
- basic local sentence:

$$\exists x_1 \cdots \exists x_k \quad \left(\bigwedge_{1 \leqslant i < j \leqslant k} \mathit{dist}(x_i, x_j) > 2r \;\wedge\; \bigwedge_{i=1}^{k} \varrho(x_i) \right)$$

Gaifman's Theorem

Throughout this talk: signatures contain only relation symbols.

Theorem: *(Gaifman, 1981)*

Every FO-formula is equivalent to an FO-formula in Gaifman normal form.

And there is an algorithm which translates a given FO-formula into an equivalent formula in Gaifman normal form.

Reminder:

- Gaifman normal form:
  Boolean combination of local formulas
  and basic local sentences
- local formula:
  FO-formula $\varrho(\bar{y})$ in which every quantifier has the form
  $\exists x\,(dist(\bar{y},x) \leqslant r \;\wedge\; \psi)$ or $\forall x\,(dist(\bar{y},x) \leqslant r \;\rightarrow\; \psi)$.
- basic local sentence:

$$\exists x_1 \cdots \exists x_k \quad \left(\bigwedge_{1 \leqslant i < j \leqslant k} dist(x_i, x_j) > 2r \;\wedge\; \bigwedge_{i=1}^{k} \varrho(x_i) \right)$$

Gaifman's Theorem

Throughout this talk: signatures contain only relation symbols.

Theorem: *(Gaifman, 1981)*

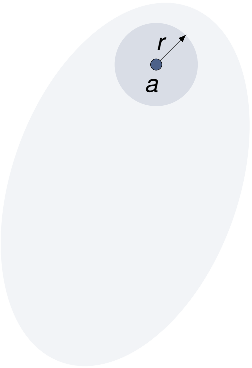
Every FO-formula is equivalent to an FO-formula in Gaifman normal form.

And there is an algorithm which translates a given FO-formula into an equivalent formula in Gaifman normal form.

Reminder:

- Gaifman normal form:
  Boolean combination of local formulas
  and basic local sentences
- local formula:
  FO-formula $\varrho(\bar{y})$ in which every quantifier has the form
  $\exists x\,(dist(\bar{y}, x) \leqslant r \;\wedge\; \psi)$ or $\forall x\,(dist(\bar{y}, x) \leqslant r \;\rightarrow\; \psi)$.
- basic local sentence:

$$\exists x_1 \cdots \exists x_k \quad \left(\bigwedge_{1 \leqslant i < j \leqslant k} dist(x_i, x_j) > 2r \;\wedge\; \bigwedge_{i=1}^{k} \varrho(x_i) \right)$$



Gaifman's Theorem

Throughout this talk: signatures contain only relation symbols.

Theorem: *(Gaifman, 1981)*

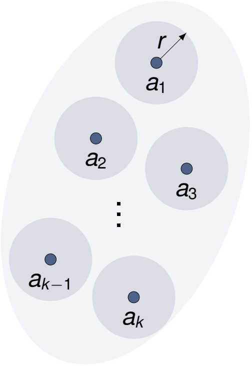
Every FO-formula is equivalent to an FO-formula in Gaifman normal form.

And there is an algorithm which translates a given FO-formula into an equivalent formula in Gaifman normal form.

Reminder:

- Gaifman normal form:
  Boolean combination of local formulas
  and basic local sentences
- local formula:
  FO-formula $\varrho(\bar{y})$ in which every quantifier has the form
  $\exists x\,(dist(\bar{y}, x) \leqslant r \ \wedge\ \psi)$ or $\forall x\,(dist(\bar{y}, x) \leqslant r \ \rightarrow\ \psi)$.
- basic local sentence:

$$\exists x_1 \cdots \exists x_k \ \left(\bigwedge_{1 \leqslant i < j \leqslant k} dist(x_i, x_j) > 2r \ \wedge\ \bigwedge_{i=1}^{k} \varrho(x_i) \right)$$



How difficult is it to transform Ψ into Gaifman normal form?

Algorithmic meta-theorems:

Gaifman's normal form can often be used as a first step for finding "efficient" algorithms for computational problems defined by FO-formulas.

Example:

- FO-sentence: $\exists x\, \exists y\, \Big(\neg E(x,y) \,\wedge\, R(x) \,\wedge\, B(y)\Big)$
- equivalent sentence in Gaifman normal form:

$$\exists z\, \Big(\, \exists x\, \exists y \big(\, dist(x,z) \leqslant 2 \wedge dist(y,z) \leqslant 2 \wedge \neg E(x,y) \wedge R(x) \wedge B(y)\,\big)\,\Big)$$
$$\vee \Big(\, \exists x\, \exists y \big(\, dist(x,y) > 2 \wedge (R(x) \vee B(x)) \wedge (R(y) \vee B(y))\,\big)$$
$$\wedge\ \exists x\, R(x)\ \wedge\ \exists x\, B(x)\,\Big)$$

Answer: There is no elementary algorithm.

Even worse: Gaifman normal form may be non-elementarily larger.

How difficult is it to transform Ψ into Gaifman normal form?

Algorithmic meta-theorems:

Gaifman's normal form can often be used as a first step for finding "efficient" algorithms for computational problems defined by FO-formulas.

Example:

- FO-sentence: $\exists x\, \exists y\, \big(\neg E(x,y) \;\wedge\; R(x) \;\wedge\; B(y)\big)$
- equivalent sentence in Gaifman normal form:

$$\exists z\, \Big(\exists x\, \exists y\big(\mathit{dist}(x,z) \leqslant 2 \wedge \mathit{dist}(y,z) \leqslant 2 \wedge \neg E(x,y) \wedge R(x) \wedge B(y) \big) \Big)$$
$$\vee \Big(\exists x\, \exists y\big(\mathit{dist}(x,y) > 2 \wedge (R(x) \vee B(x)) \wedge (R(y) \vee B(y)) \big) \wedge \exists x\, R(x) \;\wedge\; \exists x\, B(x) \Big)$$

Answer: There is no elementary algorithm.

Even worse: Gaifman normal form may be non-elementarily larger.

How difficult is it to transform Ψ into Gaifman normal form?

Algorithmic meta-theorems:

Gaifman's normal form can often be used as a first step for finding "efficient" algorithms for computational problems defined by FO-formulas.

Example:

- FO-sentence: $\exists x\, \exists y\, \big(\neg E(x,y) \wedge R(x) \wedge B(y)\big)$
- equivalent sentence in Gaifman normal form:

$$\exists z\, \Big(\, \exists x\, \exists y\big(\, dist(x,z) \leqslant 2 \wedge dist(y,z) \leqslant 2 \wedge \neg E(x,y) \wedge R(x) \wedge B(y)\,\big)\,\Big)$$
$$\vee \Big(\, \exists x\, \exists y\big(\, dist(x,y) > 2 \wedge (R(x) \vee B(x)) \wedge (R(y) \vee B(y))\,\big)$$
$$\wedge\ \exists x\, R(x)\ \wedge\ \exists x\, B(x)\,\Big)$$

Answer: There is no elementary algorithm.

Even worse: Gaifman normal form may be non-elementarily larger.

How difficult is it to transform Ψ into Gaifman normal form?

Algorithmic meta-theorems:

Gaifman's normal form can often be used as a first step for finding "efficient" algorithms for computational problems defined by FO-formulas.

Example:

- FO-sentence: $\exists x\, \exists y\, \big(\neg E(x,y) \,\wedge\, R(x) \,\wedge\, B(y)\big)$
- equivalent sentence in Gaifman normal form:

$$\exists z\, \Big(\, \exists x\, \exists y \big(\, dist(x,z) \leqslant 2 \wedge dist(y,z) \leqslant 2 \wedge \neg E(x,y) \wedge R(x) \wedge B(y)\,\big)\,\Big)$$
$$\vee \Big(\, \exists x\, \exists y \big(\, dist(x,y) > 2 \wedge (R(x) \vee B(x)) \wedge (R(y) \vee B(y))\,\big)$$
$$\wedge\, \exists x\, R(x) \,\wedge\, \exists x\, B(x)\,\Big)$$

Answer: There is no elementary algorithm.

Even worse: Gaifman normal form may be non-elementarily larger.

How difficult is it to transform Ψ into Gaifman normal form?

A precise statement of the lower bound:

Theorem: *(Dawar, Grohe, Kreutzer, Schweikardt, ICALP'07 and H., Kuske, Schweikardt, LICS'13)*

There is an $\epsilon > 0$ *and a sequence of* FO(E)*-sentences* $(\Psi_n)_{n \geqslant 1}$ *of increasing size such that every* FO(E)*-sentence in Gaifman normal form that is equivalent to* Ψ_n *has size* $\geqslant$ *Tower*$(\epsilon \cdot ||\Psi_n||)$.

Proof: Use succinct encodings of large natural numbers by trees.

- A class $\mathfrak{C}$ of structures has bounded degree d, if there is a number $d \geqslant 1$ such that the Gaifman graph of every structure in $\mathfrak{C}$ has degree $\leqslant d$.
- $\mathfrak{C}$ has unbounded degree, if there is no such d.

Note: The proof doesn't work for classes of structures of bounded degree.

Question: Can we find a better algorithm on these classes of structures?

How difficult is it to transform Ψ into Gaifman normal form?

A precise statement of the lower bound:

Theorem: *(Dawar, Grohe, Kreutzer, Schweikardt, ICALP'07 and H., Kuske, Schweikardt, LICS'13)*

There is an $\epsilon > 0$ and a sequence of $\mathrm{FO}(E)$-sentences $(\Psi_n)_{n\geqslant 1}$ of increasing size such that every $\mathrm{FO}(E)$-sentence in Gaifman normal form that is equivalent to Ψ_n on the class of forests of unbounded degree has size $\geqslant$ $Tower(\epsilon \cdot ||\Psi_n||)$.

Proof: Use succinct encodings of large natural numbers by trees.

- A class $\mathfrak{C}$ of structures has bounded degree d, if there is a number $d \geqslant 1$ such that the Gaifman graph of every structure in $\mathfrak{C}$ has degree $\leqslant d$.
- $\mathfrak{C}$ has unbounded degree, if there is no such d.

Note: The proof doesn't work for classes of structures of bounded degree.

Question: Can we find a better algorithm on these classes of structures?

How difficult is it to transform Ψ into Gaifman normal form?

A precise statement of the lower bound:

Theorem: *(Dawar, Grohe, Kreutzer, Schweikardt, ICALP'07 and H., Kuske, Schweikardt, LICS'13)*

There is an $\epsilon > 0$ *and a sequence of* $\mathrm{FO}(E)$*-sentences* $(\Psi_n)_{n \geqslant 1}$ *of increasing size such that every* $\mathrm{FO}(E)$*-sentence in Gaifman normal form that is equivalent to* Ψ_n *on the class of forests of unbounded degree has size* $\geqslant$ *Tower*$(\epsilon \cdot ||\Psi_n||)$.

Proof: Use succinct encodings of large natural numbers by trees.

- A class $\mathfrak{C}$ of structures has bounded degree d, if there is a number $d \geqslant 1$ such that the Gaifman graph of every structure in $\mathfrak{C}$ has degree $\leqslant d$.
- $\mathfrak{C}$ has unbounded degree, if there is no such d.

Note: The proof doesn't work for classes of structures of bounded degree.

Question: Can we find a better algorithm on these classes of structures?

How difficult is it to transform Ψ into Gaifman normal form?

A precise statement of the lower bound:

Theorem: *(Dawar, Grohe, Kreutzer, Schweikardt, ICALP'07 and H., Kuske, Schweikardt, LICS'13)*

There is an $\epsilon > 0$ *and a sequence of* $\mathrm{FO}(E)$*-sentences* $(\Psi_n)_{n \geqslant 1}$ *of increasing size such that every* $\mathrm{FO}(E)$*-sentence in Gaifman normal form that is equivalent to* Ψ_n *on the class of forests of unbounded degree has size* $\geqslant$ *Tower*$(\epsilon \cdot ||\Psi_n||)$.

Proof: Use succinct encodings of large natural numbers by trees.

- A class $\mathfrak{C}$ of structures has bounded degree d, if there is a number $d \geqslant 1$ such that the Gaifman graph of every structure in $\mathfrak{C}$ has degree $\leqslant d$.
- $\mathfrak{C}$ has unbounded degree, if there is no such d.

Note: The proof doesn't work for classes of structures of bounded degree.

Question: Can we find a better algorithm on these classes of structures?

How difficult is it to transform Ψ into Gaifman normal form?

A precise statement of the lower bound:

Theorem: *(Dawar, Grohe, Kreutzer, Schweikardt, ICALP'07 and H., Kuske, Schweikardt, LICS'13)*

There is an $\epsilon > 0$ and a sequence of FO(E)-sentences $(\Psi_n)_{n\geqslant 1}$ of increasing size such that every FO(E)-sentence in Gaifman normal form that is equivalent to Ψ_n on the class of forests of unbounded degree has size $\geqslant$ Tower$(\epsilon \cdot ||\Psi_n||)$.

Proof: Use succinct encodings of large natural numbers by trees.

- A class $\mathfrak{C}$ of structures has bounded degree d, if there is a number $d \geqslant 1$ such that the Gaifman graph of every structure in $\mathfrak{C}$ has degree $\leqslant d$.
- $\mathfrak{C}$ has unbounded degree, if there is no such d.

Note: The proof doesn't work for classes of structures of bounded degree.

Question: Can we find a better algorithm on these classes of structures?

Main Result: A 3-fold exponential algorithm for structures of bounded degree

Let $d \geqslant 1$. Formulas Ψ and Φ are called *d*-equivalent if they are equivalent on all structures of degree $\leqslant d$.

Theorem: *(H., Kuske, Schweikardt, LICS'13)*

(a) *Let σ be a relational signature, and let $d \geqslant 1$.*
There is a 3-fold exponential algorithm which transforms an input $\mathrm{FO}(\sigma)$-formula Ψ in time

$$2^{d^{2^{O(||\Psi||)}}}$$

into a d-equivalent formula Ψ^G in Gaifman normal form.

(b) *This is optimal: There is an $\epsilon > 0$ and a sequence of $\mathrm{FO}(E)$-sentences $(\Psi_n)_{n\geqslant 1}$ of increasing size such that every $\mathrm{FO}(E)$-sentence in Gaifman normal form that is equivalent to Ψ_n on the class of binary forests has size at least*

$$2^{2^{2^{\epsilon||\Psi_n||}}}.$$

Proof: (a) In this talk. (b) Use succinct encodings of numbers by binary trees.

Main Result: A 3-fold exponential algorithm for structures of bounded degree

Let $d \geqslant 1$. Formulas Ψ and Φ are called d-equivalent if they are equivalent on all structures of degree $\leqslant d$.

Theorem: *(H., Kuske, Schweikardt, LICS'13)*

(a) *Let σ be a relational signature, and let $d \geqslant 1$.*
There is a 3-fold exponential algorithm which transforms an input $\mathrm{FO}(\sigma)$-formula Ψ in time

$$2^{d^{2^{O(||\Psi||)}}}$$

into a d-equivalent formula Ψ^G in Gaifman normal form.

(b) *This is optimal: There is an $\epsilon > 0$ and a sequence of $\mathrm{FO}(E)$-sentences $(\Psi_n)_{n \geqslant 1}$ of increasing size such that every $\mathrm{FO}(E)$-sentence in Gaifman normal form that is equivalent to Ψ_n on the class of binary forests has size at least*

$$2^{2^{2^{\epsilon||\Psi_n||}}}.$$

Proof: (a) In this talk. (b) Use succinct encodings of numbers by binary trees.

Main Result: A 3-fold exponential algorithm for structures of bounded degree

Let $d \geqslant 1$. Formulas Ψ and Φ are called d-equivalent if they are equivalent on all structures of degree $\leqslant d$.

Theorem: *(H., Kuske, Schweikardt, LICS'13)*

(a) *Let σ be a relational signature, and let $d \geqslant 1$.*
There is a 3-fold exponential algorithm which transforms an input $\mathrm{FO}(\sigma)$-formula Ψ in time

$$2^{d^{2^{O(||\Psi||)}}}$$

into a d-equivalent formula Ψ^G in Gaifman normal form.

(b) *This is optimal: There is an $\epsilon > 0$ and a sequence of $\mathrm{FO}(E)$-sentences $(\Psi_n)_{n\geqslant 1}$ of increasing size such that every $\mathrm{FO}(E)$-sentence in Gaifman normal form that is equivalent to Ψ_n*
on the class of binary forests has size at least

$$2^{2^{2^{\epsilon||\Psi_n||}}}.$$

Proof: (a) In this talk. (b) Use succinct encodings of numbers by binary trees.

Main Result: A 3-fold exponential algorithm for structures of bounded degree

Let $d \geqslant 1$. Formulas Ψ and Φ are called d-equivalent if they are equivalent on all structures of degree $\leqslant d$.

Theorem: *(H., Kuske, Schweikardt, LICS'13)*

(a) *Let σ be a relational signature, and let $d \geqslant 1$.*
There is a 3-fold exponential algorithm which transforms an input $\mathrm{FO}(\sigma)$-formula Ψ in time

$$2^{d^{2^{O(||\Psi||)}}}$$

into a d-equivalent formula Ψ^G in Gaifman normal form.

(b) *This is optimal: There is an $\epsilon > 0$ and a sequence of $\mathrm{FO}(E)$-sentences $(\Psi_n)_{n \geqslant 1}$ of increasing size such that every $\mathrm{FO}(E)$-sentence in Gaifman normal form that is equivalent to Ψ_n*
on the class of binary forests has size at least

$$2^{2^{2^{\epsilon ||\Psi_n||}}}.$$

Proof: (a) In this talk. (b) Use succinct encodings of numbers by binary trees.

Outline

Outline

The Algorithm

Input: An FO(σ)-sentence Ψ For simplicity: sentence!

Goal: Transform Ψ in time 3-exp($||\Psi||$) into
a d-equivalent sentence Ψ^G in Gaifman normal form

Step 1: Transform Ψ into a d-equivalent sentence Ψ^H in Hanf normal form

I.e., Ψ^H is a Boolean combination of Hanf-sentences, i.e., sentences of the form

$$\exists^{\geqslant k} x\, \varrho(x)\,, \qquad \text{where}$$

$k \geqslant 1$ and $\varrho(x)$ is r-local around x, for some r.
(More precisely, $\varrho(x)$ describes the isomorphism type of an r-neighbourhood.)

By a Theorem of Bollig and Kuske (2012), this can be done in time 3-exp($||\Psi||$).

The Algorithm

Input: An FO(σ)-sentence Ψ For simplicity: sentence!

Goal: Transform Ψ in time 3-exp($||\Psi||$) into a d-equivalent sentence Ψ^G in Gaifman normal form

Step 1: Transform Ψ into a d-equivalent sentence Ψ^H in Hanf normal form

I.e., Ψ^H is a Boolean combination of Hanf-sentences, i.e., sentences of the form

$$\exists^{\geqslant k} x \, \varrho(x) \,, \qquad \text{where}$$

$k \geqslant 1$ and $\varrho(x)$ is r-local around x, for some r.
(More precisely, $\varrho(x)$ describes the isomorphism type of an r-neighbourhood.)

By a Theorem of Bollig and Kuske (2012), this can be done in time 3-exp($||\Psi||$).

The Algorithm

Input: An FO(σ)-sentence Ψ For simplicity: sentence!

Goal: Transform Ψ in time 3-exp($||\Psi||$) into
a d-equivalent sentence Ψ^G in Gaifman normal form

Step 1: Transform Ψ into a d-equivalent sentence Ψ^H in Hanf normal form

I.e., Ψ^H is a Boolean combination of Hanf-sentences, i.e., sentences of the form

$$\exists^{\geqslant k} x\, \varrho(x)\,, \qquad \text{where}$$

$k \geqslant 1$ and $\varrho(x)$ is r-local around x, for some r.
(More precisely, $\varrho(x)$ describes the isomorphism type of an r-neighbourhood.)

By a Theorem of Bollig and Kuske (2012), this can be done in time 3-exp($||\Psi||$).

The Algorithm — Step 2

Step 1: Transform Ψ into a d-equivalent sentence Ψ^H in Hanf normal form ✔

Step 2: Transform each Hanf-sentence $\exists^{\geqslant k} x\, \varrho(x)$ of Ψ^H into an equivalent sentence in Gaifman normal form.

Technical Lemma:

This can be done in time $2^{O(k \log k)}(\log r + ||\varrho||)$.

Step 1 + Step 2: Total running time: 3-exp($||\Psi||$).

Proof of the technical lemma:

- Instead of σ-structures $\mathcal{A}$, consider their Gaifman graphs $G_{\mathcal{A}}$.
- Each node v of $G_{\mathcal{A}}$ is
  - colored red, if $\mathcal{A} \models \varrho(v)$,
  - colored blue, if $\mathcal{A} \not\models \varrho(v)$.
- $\mathcal{A} \models \exists^{\geqslant k} x\, \varrho(x) \iff$ There are at least k red nodes in $G_{\mathcal{A}}$.
- Thus, consider red-blue-colored graphs G, and investigate the distribution of red nodes in G.

The Algorithm — Step 2

Step 1: Transform Ψ into a d-equivalent sentence Ψ^H in Hanf normal form ✔

Step 2: Transform each Hanf-sentence $\exists^{\geqslant k} x\, \varrho(x)$ of Ψ^H into an equivalent sentence in Gaifman normal form.

Technical Lemma:

This can be done in time $2^{O(k \log k)}(\log r + ||\varrho||)$.

Step 1 + Step 2: Total running time: 3-exp($||\Psi||$).

Proof of the technical lemma:

- Instead of σ-structures $\mathcal{A}$, consider their Gaifman graphs $G_{\mathcal{A}}$.
- Each node v of $G_{\mathcal{A}}$ is
  - colored red, if $\mathcal{A} \models \varrho(v)$,
  - colored blue, if $\mathcal{A} \not\models \varrho(v)$.
- $\mathcal{A} \models \exists^{\geqslant k} x\, \varrho(x) \iff$ There are at least k red nodes in $G_{\mathcal{A}}$.
- Thus, consider red-blue-colored graphs G, and investigate the distribution of red nodes in G.

The Algorithm — Step 2

Step 1: Transform Ψ into a d-equivalent sentence Ψ^H in Hanf normal form ✔

Step 2: Transform each Hanf-sentence $\exists^{\geqslant k} x\, \varrho(x)$ of Ψ^H into an equivalent sentence in Gaifman normal form.

Technical Lemma: *No restriction to structures of bounded degree!*

This can be done in time $2^{O(k \log k)}(\log r + ||\varrho||)$.

Step 1 + Step 2: Total running time: 3-exp($||\Psi||$).

Proof of the technical lemma:

- Instead of σ-structures $\mathcal{A}$, consider their Gaifman graphs $G_{\mathcal{A}}$.
- Each node v of $G_{\mathcal{A}}$ is
  - colored red, if $\mathcal{A} \models \varrho(v)$,
  - colored blue, if $\mathcal{A} \not\models \varrho(v)$.
- $\mathcal{A} \models \exists^{\geqslant k} x\, \varrho(x) \iff$ There are at least k red nodes in $G_{\mathcal{A}}$.
- Thus, consider red-blue-colored graphs G, and investigate the distribution of red nodes in G.

The Algorithm — Step 2

Step 1: Transform Ψ into a d-equivalent sentence Ψ^H in Hanf normal form ✔

Step 2: Transform each Hanf-sentence $\exists^{\geqslant k} x\, \varrho(x)$ of Ψ^H into an equivalent sentence in Gaifman normal form.

Technical Lemma: *No restriction to structures of bounded degree!*

This can be done in time $2^{O(k \log k)}(\log r + ||\varrho||)$.

Step 1 + Step 2: Total running time: 3-exp($||\Psi||$).

Proof of the technical lemma:

- Instead of σ-structures $\mathcal{A}$, consider their Gaifman graphs $G_{\mathcal{A}}$.
- Each node v of $G_{\mathcal{A}}$ is
  - colored red, if $\mathcal{A} \models \varrho(v)$,
  - colored blue, if $\mathcal{A} \not\models \varrho(v)$.
- $\mathcal{A} \models \exists^{\geqslant k} x\, \varrho(x) \iff$ There are at least k red nodes in $G_{\mathcal{A}}$.
- Thus, consider red-blue-colored graphs G, and investigate the distribution of red nodes in G.

The Algorithm — Step 2

Step 1: Transform Ψ into a d-equivalent sentence Ψ^H in Hanf normal form ✔

Step 2: Transform each Hanf-sentence $\exists^{\geqslant k} x\, \varrho(x)$ of Ψ^H into an equivalent sentence in Gaifman normal form.

Technical Lemma: *No restriction to structures of bounded degree!*

This can be done in time $2^{O(k \log k)}(\log r + ||\varrho||)$.

Step 1 + Step 2: Total running time: 3-exp($||\Psi||$).

Proof of the technical lemma:

- Instead of σ-structures $\mathcal{A}$, consider their Gaifman graphs $G_{\mathcal{A}}$.
- Each node v of $G_{\mathcal{A}}$ is
  - colored red, if $\mathcal{A} \models \varrho(v)$,
  - colored blue, if $\mathcal{A} \not\models \varrho(v)$.
- $\mathcal{A} \models \exists^{\geqslant k} x\, \varrho(x) \iff$ There are at least k red nodes in $G_{\mathcal{A}}$.
- Thus, consider red-blue-colored graphs G, and investigate the distribution of red nodes in G.

The Algorithm — Step 2

Step 1: Transform Ψ into a d-equivalent sentence Ψ^H in Hanf normal form ✔

Step 2: Transform each Hanf-sentence $\exists^{\geqslant k} x\, \varrho(x)$ of Ψ^H into an equivalent sentence in Gaifman normal form.

Technical Lemma: *No restriction to structures of bounded degree!*

This can be done in time $2^{O(k \log k)}(\log r + ||\varrho||)$.

Step 1 + Step 2: Total running time: 3-exp($||\Psi||$).

Proof of the technical lemma:

- Instead of σ-structures $\mathcal{A}$, consider their Gaifman graphs $G_{\mathcal{A}}$.
- Each node v of $G_{\mathcal{A}}$ is
  - colored red, if $\mathcal{A} \models \varrho(v)$,
  - colored blue, if $\mathcal{A} \not\models \varrho(v)$.
- $\mathcal{A} \models \exists^{\geqslant k} x\, \varrho(x) \iff$ There are at least k red nodes in $G_{\mathcal{A}}$.
- Thus, consider red-blue-colored graphs G, and investigate the distribution of red nodes in G.

The Algorithm — Step 2

Step 1: Transform Ψ into a d-equivalent sentence Ψ^H in Hanf normal form ✔

Step 2: Transform each Hanf-sentence $\exists^{\geqslant k} x\, \varrho(x)$ of Ψ^H into an equivalent sentence in Gaifman normal form.

Technical Lemma: *No restriction to structures of bounded degree!*

This can be done in time $2^{O(k \log k)}(\log r + ||\varrho||)$.

Step 1 + Step 2: Total running time: 3-exp($||\Psi||$).

Proof of the technical lemma:

- Instead of σ-structures $\mathcal{A}$, consider their Gaifman graphs $G_{\mathcal{A}}$.
- Each node v of $G_{\mathcal{A}}$ is
  - colored red, if $\mathcal{A} \models \varrho(v)$,
  - colored blue, if $\mathcal{A} \not\models \varrho(v)$.
- $\mathcal{A} \models \exists^{\geqslant k} x\, \varrho(x) \iff$ There are at least k red nodes in $G_{\mathcal{A}}$.
- Thus, consider red-blue-colored graphs G, and investigate the distribution of red nodes in G.

The Algorithm — Step 2

Step 1: Transform Ψ into a d-equivalent sentence Ψ^H in Hanf normal form ✔

Step 2: Transform each Hanf-sentence $\exists^{\geqslant k} x\, \varrho(x)$ of Ψ^H into an equivalent sentence in Gaifman normal form.

Technical Lemma: *No restriction to structures of bounded degree!*

This can be done in time $2^{O(k \log k)}(\log r + ||\varrho||)$.

Step 1 + Step 2: Total running time: 3-exp($||\Psi||$).

Proof of the technical lemma:

- Instead of σ-structures $\mathcal{A}$, consider their Gaifman graphs $G_{\mathcal{A}}$.
- Each node v of $G_{\mathcal{A}}$ is
  - colored red, if $\mathcal{A} \models \varrho(v)$,
  - colored blue, if $\mathcal{A} \not\models \varrho(v)$.
- $\mathcal{A} \models \exists^{\geqslant k} x\, \varrho(x) \iff$ There are at least k red nodes in $G_{\mathcal{A}}$.
- Thus, consider red-blue-colored graphs G, and investigate the distribution of red nodes in G.

Where are the red nodes?

An r-scattered red set is a non-empty set of red nodes of pairwise distance $> r$.

Combinatorial Lemma:

Let $k, r, c \geqslant 1$, and let G be a red-blue-colored graph. Then, one of the following statements is true:

(a) *There is no red node.*

(b) *There is a cr-scattered red set of size $\geqslant k$.*

(c) *There is a number s with $0 \leqslant s \leqslant k-2$ such that for $R_s := (c+1)^s cr$, there is a cR_s-scattered red set W_s of size $< k$, such that every red node of G belongs to the R_s-neighbourhood of W_s.*

Proof: W.l.o.g., neither (a) nor (b) is true. Construct a sequence $W_0 \supset W_1 \supset \ldots \supset W_s$ of sets of red nodes such that, for every j, $N_{R_j}(W_j)$ contains all red nodes of G:

W_0: a cr-scattered red set of maximum size.
Clearly, $|W_0| < k$ and $N_{cr}(W_0) = N_{R_0}(W_0)$ contains all red nodes of G.

W_{j+1}: if W_j contains two nodes a, b with $dist(a, b) \leqslant cR_j$, then $W_{j+1} := W_j \setminus \{b\}$.
Clearly, $N_{R_j + cR_j}(W_{j+1}) = N_{R_{j+1}}(W_{j+1})$ contains all red nodes of G. $\square$

Where are the red nodes?

An r-scattered red set is a non-empty set of red nodes of pairwise distance $> r$.

Combinatorial Lemma:

Let $k, r, c \geqslant 1$, and let G be a red-blue-colored graph.
Then, one of the following statements is true:

(a) *There is no red node.*

(b) *There is a cr-scattered red set of size $\geqslant k$.*

(c) *There is a number s with $0 \leqslant s \leqslant k-2$ such that for $R_s := (c+1)^s cr$, there is a cR_s-scattered red set W_s of size $< k$, such that every red node of G belongs to the R_s-neighbourhood of W_s.*

Proof: W.l.o.g., neither (a) nor (b) is true. Construct a sequence $W_0 \supset W_1 \supset \ldots \supset W_s$ of sets of red nodes such that, for every j, $N_{R_j}(W_j)$ contains all red nodes of G:

W_0: a cr-scattered red set of maximum size.
Clearly, $|W_0| < k$ and $N_{cr}(W_0) = N_{R_0}(W_0)$ contains all red nodes of G.

W_{j+1}: if W_j contains two nodes a, b with $dist(a,b) \leqslant cR_j$, then $W_{j+1} := W_j \setminus \{b\}$.
Clearly, $N_{R_j + cR_j}(W_{j+1}) = N_{R_{j+1}}(W_{j+1})$ contains all red nodes of G. □

Where are the red nodes?

An r-scattered red set is a non-empty set of red nodes of pairwise distance $> r$.

Combinatorial Lemma:

Let $k, r, c \geqslant 1$, and let G be a red-blue-colored graph. Then, one of the following statements is true:

(a) *There is no red node.*

(b) *There is a cr-scattered red set of size $\geqslant k$.*

(c) *There is a number s with $0 \leqslant s \leqslant k-2$ such that for $R_s := (c+1)^s cr$, there is a cR_s-scattered red set W_s of size $< k$, such that every red node of G belongs to the R_s-neighbourhood of W_s.*

Proof: W.l.o.g., neither (a) nor (b) is true. Construct a sequence $W_0 \supset W_1 \supset \ldots \supset W_s$ of sets of red nodes such that, for every j, $N_{R_j}(W_j)$ contains all red nodes of G:

W_0: a cr-scattered red set of maximum size.
Clearly, $|W_0| < k$ and $N_{cr}(W_0) = N_{R_0}(W_0)$ contains all red nodes of G.

W_{j+1}: if W_j contains two nodes a, b with $dist(a, b) \leqslant cR_j$, then $W_{j+1} := W_j \setminus \{b\}$.
Clearly, $N_{R_j + cR_j}(W_{j+1}) = N_{R_{j+1}}(W_{j+1})$ contains all red nodes of G. $\square$

Where are the red nodes?

An r-scattered red set is a non-empty set of red nodes of pairwise distance $> r$.

Combinatorial Lemma:

Let $k, r, c \geqslant 1$, and let G be a red-blue-colored graph.
Then, one of the following statements is true:

(a) *There is no red node.*

(b) *There is a cr-scattered red set of size $\geqslant k$.*

(c) *There is a number s with $0 \leqslant s \leqslant k-2$ such that for $R_s := (c+1)^s cr$, there is a cR_s-scattered red set W_s of size $< k$, such that every red node of G belongs to the R_s-neighbourhood of W_s.*

Proof: W.l.o.g., neither (a) nor (b) is true. Construct a sequence $W_0 \supset W_1 \supset \ldots \supset W_s$ of sets of red nodes such that, for every j, $N_{R_j}(W_j)$ contains all red nodes of G:

W_0: a cr-scattered red set of maximum size.
Clearly, $|W_0| < k$ and $N_{cr}(W_0) = N_{R_0}(W_0)$ contains all red nodes of G.

W_{j+1}: if W_j contains two nodes a, b with $dist(a, b) \leqslant cR_j$, then $W_{j+1} := W_j \setminus \{b\}$.
Clearly, $N_{R_j + cR_j}(W_{j+1}) = N_{R_{j+1}}(W_{j+1})$ contains all red nodes of G. □

Where are the red nodes?

An r-scattered red set is a non-empty set of red nodes of pairwise distance $> r$.

Combinatorial Lemma:

Let $k, r, c \geqslant 1$, and let G be a red-blue-colored graph.
Then, one of the following statements is true:

(a) *There is no red node.*

(b) *There is a cr-scattered red set of size $\geqslant k$.*

(c) *There is a number s with $0 \leqslant s \leqslant k-2$ such that for $R_s := (c+1)^s cr$, there is a cR_s-scattered red set W_s of size $< k$, such that every red node of G belongs to the R_s-neighbourhood of W_s.*

Proof: W.l.o.g., neither (a) nor (b) is true. Construct a sequence $W_0 \supset W_1 \supset \ldots \supset W_s$ of sets of red nodes such that, for every j, $N_{R_j}(W_j)$ contains all red nodes of G:

W_0: a cr-scattered red set of maximum size.
Clearly, $|W_0| < k$ and $N_{cr}(W_0) = N_{R_0}(W_0)$ contains all red nodes of G.

W_{j+1}: if W_j contains two nodes a, b with $dist(a, b) \leqslant cR_j$, then $W_{j+1} := W_j \setminus \{b\}$.
Clearly, $N_{R_j+cR_j}(W_{j+1}) = N_{R_{j+1}}(W_{j+1})$ contains all red nodes of G. $\square$

Where are the red nodes?

An r-scattered red set is a non-empty set of red nodes of pairwise distance $> r$.

Combinatorial Lemma:

Let $k, r, c \geqslant 1$, and let G be a red-blue-colored graph. Then, one of the following statements is true:

(a) *There is no red node.*

(b) *There is a cr-scattered red set of size $\geqslant k$.*

(c) *There is a number s with $0 \leqslant s \leqslant k-2$ such that for $R_s := (c+1)^s cr$, there is a cR_s-scattered red set W_s of size $< k$, such that every red node of G belongs to the R_s-neighbourhood of W_s.*

Proof: W.l.o.g., neither (a) nor (b) is true. Construct a sequence $W_0 \supset W_1 \supset \ldots \supset W_s$ of sets of red nodes such that, for every j, $N_{R_j}(W_j)$ contains all red nodes of G:

W_0: a cr-scattered red set of maximum size.
Clearly, $|W_0| < k$ and $N_{cr}(W_0) = N_{R_0}(W_0)$ contains all red nodes of G.

W_{j+1}: if W_j contains two nodes a, b with $dist(a, b) \leqslant cR_j$, then $W_{j+1} := W_j \backslash \{b\}$.
Clearly, $N_{R_j + cR_j}(W_{j+1}) = N_{R_{j+1}}(W_{j+1})$ contains all red nodes of G. □

Where are the red nodes?

An r-scattered red set is a non-empty set of red nodes of pairwise distance $> r$.

Combinatorial Lemma:

Let $k, r, c \geqslant 1$, and let G be a red-blue-colored graph. Then, one of the following statements is true:

(a) *There is no red node.*

(b) *There is a cr-scattered red set of size $\geqslant k$.*

(c) *There is a number s with $0 \leqslant s \leqslant k-2$ such that for $R_s := (c+1)^s cr$, there is a cR_s-scattered red set W_s of size $< k$, such that every red node of G belongs to the R_s-neighbourhood of W_s.*

With an additional argument, this leads to a set of conditions which can be directly translated into a Boolean combination Φ^G of basic-local sentences, such that

$$G_{\mathcal{A}} \text{ has } \geqslant k \text{ red nodes} \quad \Longleftrightarrow \quad \mathcal{A} \models \Phi^G.$$

Outline

Final Remarks

Main result presented here:

- For every $d \geqslant 1$, there is a 3-fold exponential algorithm which transforms an input FO-formula Ψ into a formula in Gaifman normal form that is equivalent to Ψ on the class of all structures of degree at most d.

  And this is optimal:
  For binary forests, we show a 3-fold exponential lower bound on the formula size.

Variations:

- For structures of degree 2: Our algorithm is 2-fold exponential; and we can prove a matching lower bound.
- For structures where the size of r-neighborhoods is $\leqslant p(r)$, for a polynomial p: Our algorithm is 2-fold exponential.

Applications:

- Simplified proofs for Linear-time FO-model checking (Frick, Grohe, LICS'02) and Constant-delay FO-query enumeration (Kazana, Segoufin, LMCS'11).

Ongoing work:

- Construction of Feferman-Vaught-decompositions on structures of bounded degree.

Final Remarks

Main result presented here:

- For every $d \geqslant 1$, there is a 3-fold exponential algorithm which transforms an input FO-formula Ψ into a formula in Gaifman normal form that is equivalent to Ψ on the class of all structures of degree at most d.

  And this is optimal:
  For binary forests, we show a 3-fold exponential lower bound on the formula size.

Variations:

- For structures of degree 2: Our algorithm is 2-fold exponential; and we can prove a matching lower bound.
- For structures where the size of r-neighborhoods is $\leqslant p(r)$, for a polynomial p: Our algorithm is 2-fold exponential.

Applications:

- Simplified proofs for Linear-time FO-model checking (Frick, Grohe, LICS'02) and Constant-delay FO-query enumeration (Kazana, Segoufin, LMCS'11).

Ongoing work:

- Construction of Feferman-Vaught-decompositions on structures of bounded degree.

Final Remarks

Main result presented here:

- For every $d \geqslant 1$, there is a 3-fold exponential algorithm which transforms an input FO-formula Ψ into a formula in Gaifman normal form that is equivalent to Ψ on the class of all structures of degree at most d.

  And this is optimal:
  For binary forests, we show a 3-fold exponential lower bound on the formula size.

Variations:

- For structures of degree 2: Our algorithm is 2-fold exponential; and we can prove a matching lower bound.
- For structures where the size of r-neighborhoods is $\leqslant p(r)$, for a polynomial p: Our algorithm is 2-fold exponential.

Applications:

- Simplified proofs for Linear-time FO-model checking (Frick, Grohe, LICS'02) and Constant-delay FO-query enumeration (Kazana, Segoufin, LMCS'11).

Ongoing work:

- Construction of Feferman-Vaught-decompositions on structures of bounded degree.

Final Remarks

Main result presented here:

- For every $d \geqslant 1$, there is a 3-fold exponential algorithm which transforms an input FO-formula Ψ into a formula in Gaifman normal form that is equivalent to Ψ on the class of all structures of degree at most d.

  And this is optimal:
  For binary forests, we show a 3-fold exponential lower bound on the formula size.

Variations:

- For structures of degree 2: Our algorithm is 2-fold exponential; and we can prove a matching lower bound.
- For structures where the size of r-neighborhoods is $\leqslant p(r)$, for a polynomial p: Our algorithm is 2-fold exponential.

Applications:

- Simplified proofs for Linear-time FO-model checking (Frick, Grohe, LICS'02) and Constant-delay FO-query enumeration (Kazana, Segoufin, LMCS'11).

Ongoing work:

- Construction of Feferman-Vaught-decompositions on structures of bounded degree.

Final Remarks

Main result presented here:

- For every $d \geqslant 1$, there is a 3-fold exponential algorithm which transforms an input FO-formula Ψ into a formula in Gaifman normal form that is equivalent to Ψ on the class of all structures of degree at most d.

  And this is optimal:
  For binary forests, we show a 3-fold exponential lower bound on the formula size.

Variations:

- For structures of degree 2: Our algorithm is 2-fold exponential;
  and we can prove a matching lower bound.
- For structures where the size of r-neighborhoods is $\leqslant p(r)$, for a polynomial p:
  Our algorithm is 2-fold exponential.

Applications:

- Simplified proofs for Linear-time FO-model checking (Frick, Grohe, LICS'02) and Constant-delay FO-query enumeration (Kazana, Segoufin, LMCS'11).

Ongoing work:

- Construction of Feferman-Vaught-decompositions on structures of bounded degree.

Final Remarks

Main result presented here:

- For every $d \geqslant 1$, there is a 3-fold exponential algorithm which transforms an input FO-formula Ψ into a formula in Gaifman normal form that is equivalent to Ψ on the class of all structures of degree at most d.

  And this is optimal:
  For binary forests, we show a 3-fold exponential lower bound on the formula size.

Variations:

- For structures of degree 2: Our algorithm is 2-fold exponential; and we can prove a matching lower bound.
- For structures where the size of r-neighborhoods is $\leqslant p(r)$, for a polynomial p: Our algorithm is 2-fold exponential.

Applications:

- Simplified proofs for Linear-time FO-model checking (Frick, Grohe, LICS'02) and Constant-delay FO-query enumeration (Kazana, Segoufin, LMCS'11).

Ongoing work:

- Construction of Feferman-Vaught-decompositions on structures of bounded degree.