Bemerkung 4.16

Indem man zeigt, dass eine Anfrage nicht Gaifman-lokal ist, kann man (unter Verwendung von Satz 4.15) folgern, dass die Anfrage nicht FO-definierbar ist.

Beispiel 4.17

Die Erreichbarkeits-Anfrage E^*, die jedem endlichen Graphen $G = (V, E)$ die Relation

$$E^*(G) := \{ (v, w) \in V \times V : \text{es gibt in } G \text{ einen Weg von } v \text{ nach } w \}$$

zuordnet, ist nicht Gaifman-lokal auf der Klasse aller endlichen Graphen und daher laut Satz 4.15 auch nicht FO-definierbar auf der Klasse aller endlichen Graphen.

Beweis:

Um zu zeigen, dass die Erreichbarkeits-Anfrage nicht Gaifman-lokal ist, genügt es,

für alle Zahlen $r, m \in \mathbb{N}$ einen endlichen Graphen $G_{r,m}$ und Knoten $a_1, a_2 \in V$ und $b_1, b_2 \in V$ anzugeben, so dass gilt:

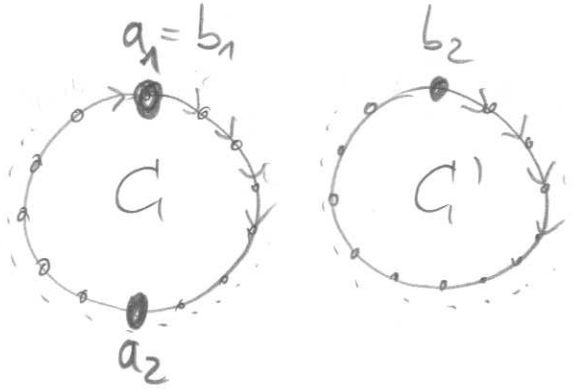
(1) $\mathcal{N}_r^{G_{r,m}}(a_1, a_2) \equiv_m \mathcal{N}_r^{G_{r,m}}(b_1, b_2)$,

(2) es gibt in $G_{r,m}$ einen Weg von a_1 nach a_2, und

(3) es gibt in $G_{r,m}$ keinen Weg von b_1 nach b_2.

Sei dazu $G_{r,m}$ der Graph, der aus 2 disjunkten gerichteten Kreisen C und C' auf je $2(2r+1)+2 = 4r+4$ Knoten besteht

Skizze: $G_{r,m}$



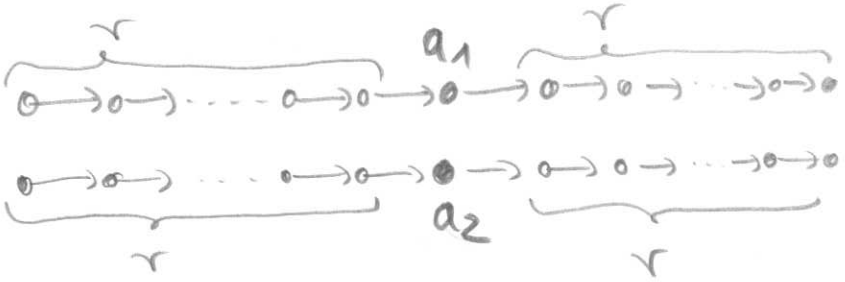
Ferner seien a_1 und a_2 zwei Knoten auf C vom Abstand $\mathrm{Dist}^G(a_1, a_2) > 2r+1$.
Sei $b_1 := a_1$ und sei b_2 ein beliebiger Knoten auf C'.

Damit gilt (2) und (3) (d.h. es gibt in $G_{r,m}$ einen Weg von a_1 nach a_2, aber es gibt keinen Weg von b_1 nach b_2).

Außerdem ist $W_r^{G_{r,m}}(a_1,a_2)$ die disjunkte Vereinigung von zwei gerichteten Pfaden der Länge $2r+1$, so dass a_1 und a_2 die Mittelpunkte der beiden Pfade sind

Skizze:

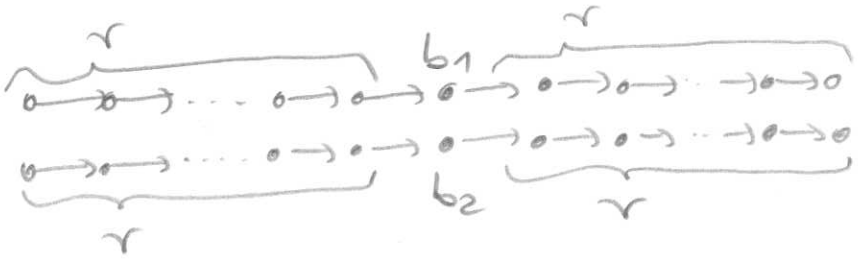
$W_r^{G_{r,m}}(a_1,a_2)$:



Außerdem ist $W_r^{G_{r,m}}(b_1,b_2)$ von der gleichen Form.

Skizze:

$W_r^{G_{r,m}}(b_1,b_2)$:



Somit gilt also: $W_r^{G_{r,m}}(a_1,a_2) \cong W_r^{G_{r,m}}(b_1,b_2)$, und insbes gilt auch f.a. $m \in \mathbb{N}$, dass

$$W_r^{G_{r,m}}(a_1,a_2) \equiv_m W_r^{G_{r,m}}(b_1,b_2).$$

Somit ist auch (1) erfüllt.

□

4.3 Der Satz von Seese über Klassen von Graphen von beschränktem Grad

Definition 4.18 (Klassen von Graphen von beschränktem Grad)

(a) Sei G ein Graph.
Der Grad eines Knotens v von G ist die Anzahl aller Kanten, die v als Ausgangspunkt oder Endpunkt haben.
Der Grad von G ist der maximale Grad eines Knotens von G.

(b) Sei C eine Klasse von Graphen und sei $d \in \mathbb{N}$.
C ist eine Klasse von Graphen vom Grad $\leq d$, falls jeder Graph $G \in C$ Grad $\leq d$ hat.

(c) C ist eine Klasse von Graphen von beschränktem Grad, falls es eine Zahl $d \in \mathbb{N}$ gibt, so dass C eine Klasse von Graphen vom Grad $\leq d$ ist.

Satz 4.19 (Seese, 1996)

Für jedes $d \in \mathbb{N}$ gibt es eine berechenbare Funktion $f_d : \mathbb{N} \to \mathbb{N}$, so dass das

> **Auswertungsproblem für FO auf der Klasse aller endlichen Graphen vom Grad $\leq d$**
>
> Eingabe: Ein endlicher Graph G vom Grad $\leq d$ und ein $FO[\{E\}]$-Satz φ
>
> Frage: Gilt $G \models \varphi$?

in Zeit $f_d(k) \cdot n$ gelöst werden kann, wobei k die Länge der Formel φ und n die Größe (einer geeigneten Kodierung) des Graphen G bezeichnet (etwa: $n = |V^G| + |E^G|$ für $G = (V^G, E^G)$).

Das heißt: Für jeden festen $FO[\{E\}]$-Satz φ ist das Problem, zu gegebenem Graphen G vom Grad $\leq d$ zu entscheiden, ob $G \models \varphi$, in Linearzeit lösbar.

Beweis:

Seeses Originalbeweis argumentiert unter Verwendung des Satzes von Hanf. Wir werden hier im Folgenden einen auf dem Satz von Gaifman basierenden Beweis geben, der (im Gegensatz zu dem Hanf-basierten Ansatz) den Vorteil hat, dass er auch auf andere Klassen von Strukturen übertragen werden kann (siehe den folgenden Satz 4.20).

Sei $d \in \mathbb{N}$ fest, sei φ der gegebene $FO[\{E\}]$-Satz und sei $G = (V^G, E^G)$ der gegebene Graph vom Grad $\leq d$. Sei $n := |V^G| + |E^G|$.

Zunächst benützen wir den Algorithmus aus dem Satz von Gaifman, um φ in einen äquivalenten Satz φ' in Gaifman-Normalform zu übersetzen. Dann testen wir für jeden einzelnen basis-lokalen Satz χ, der in φ' vorkommt, ob $G \models \chi$. Am Ende können wir dann die Resultate für die

einzelnen basis-lokalen Sätze χ einfach kombinieren, um zu ermitteln, ob $G \models \varphi$.

Sei im Folgenden χ ein basis-lokaler Satz der Form

$$\exists x_1 \dots \exists x_\ell \left(\left(\bigwedge_{1 \le i < j \le \ell} \mathrm{dist}(x_i, x_j) > 2r \right) \wedge \left(\bigwedge_{i=1}^{\ell} \psi(x_i) \right) \right)$$

wobei $\ell, r \in \mathbb{N}$ und $\psi(x)$ r-lokal um x ist.

Natürlich gilt $G \models \chi$ genau dann, wenn es mindestens ℓ Knoten vom paarweisen Abstand $> 2r$ in G gibt, so dass die r-Nachbarschaften dieser ℓ Knoten allesamt die r-lokale Formel ψ erfüllen.

Um zu entscheiden, ob $G \models \chi$, gehen wir in 2 Schritten vor:

Schritt 1: Bestimme für jeden Knoten $v \in V^G$, ob $N_r^G(v) \models \psi[v]$, und markiere diejenigen Knoten v, für die $N_r^G(v) \models \psi[v]$ gilt, als rot.

Schritt 2: Entscheide, ob es in G ℓ rote Knoten vom paarweisen Abstand $> 2r$ gibt.

Zum Lösen von Schritt 1 beachte, dass für jeden Knoten v in einem Graphen G vom Grad $\leq d$ gilt:

$$|N_r^G(v)| \leq \underbrace{1}_{\text{Abstand 0}} + \underbrace{d}_{\text{Abstand 1}} + \underbrace{d^2}_{\text{Abstand 2}} + \cdots + \underbrace{d^r}_{\text{Abstand } r} \leq d^{r+1}.$$

D.h.: Die Größe von $N_r^G(v)$ hängt nicht von der Größe von G ab, sondern nur von den Zahlen d und r.

Schritt 1 kann daher folgendermaßen gelöst werden:

Algorithmus 1:

1: for $v \in V^G$ do
2: berechne $N_r^G(v)$
3: teste, ob $N_r^G(v) \models \psi[v]$
4: if $N_r^G(v) \models \psi[v]$ then markiere v als <u>rot</u>
5: endfor

Jede der Zeilen 2-4 wird $|V^G|$-mal durchlaufen.

Für jeden einzelnen Durchlauf durch Zeile 2 wird Zeit $\leq f_d^{(1)}(r)$ gebraucht (für eine geeignete Funktion $f_d^{(1)}: \mathbb{N} \to \mathbb{N}$), da $|N_r^G(v)| \leq d^{r+1}$ ist.

Für jeden einzelnen Durchlauf durch Zeile 3 wird Zeit $\leq f_d^{(2)}(r, \underbrace{\|\psi\|}_{\text{die Länge von } \psi})$ gebraucht

(für eine geeignete Funktion $f_d^{(2)}: \mathbb{N} \times \mathbb{N} \to \mathbb{N}$), da $|N_r^G(v)| \leq d^{r+1}$ ist.

Für jeden einzelnen Durchlauf durch Zeile 4 wird Zeit $O(1)$ gebraucht.

Insgesamt benötigt Algorithmus 1 also bei Eingabe eines Graphen G vom Grad $\leq d$ und einer r-lokalen Formel $\psi(x)$ also Zeit $\leq f_d^{(3)}(r, \|\psi\|) \cdot n$, wobei $f_d^{(3)}: \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ eine geeignete Funktion ist.

Zum Lösen von Schritt 2 müssen wir nun noch ℓ rote Knoten vom paarweisen Abstand $> 2r$ finden. Dies wird durch den folgenden Algorithmus gewährleistet:

Algorithmus 2:

1: initialisiere R als die Menge aller roten Knoten in V^G

2: initialisiere $X := \emptyset$

% X ist die bisher gefundene Menge
% roter Knoten vom Abstand $> 2r$

3: while $R \neq \emptyset$ do

4: wähle ein beliebiges $v \in R$

5: $X := X \cup \{v\}$

6: $R := R \setminus N^G_{2r}(v)$

7: endwhile

8: if $|X| \geq \ell$ then akzeptiere G

9: else

10: $N^G_{4r}(X) := \bigcup_{x \in X} N^G_{4r}(x)$

11: nütze einen naiven Algorithmus zum Testen, ob es in $N^G_{4r}(X)$ ℓ rote Knoten vom paarweisen Abstand $> 2r$ gibt

12: endif

Dieser Algorithmus verwendet zunächst eine Greedy-Strategie, um aus den roten Knoten in G eine Menge X von roten Knoten mit paarweisem Abstand $> 2r$ auszuwählen.

Dazu wird in der Schleife in Zeilen 3–7 jeweils ein beliebiger Knoten v aus R ausgewählt, und danach werden alle Knoten mit Abstand $\leq 2r$ von v aus R gelöscht.

Würde auf diese Weise eine Menge X mit mindestens ℓ roten Knoten vom paarweisen Abstand $> 2r$ gefunden, so können wir anhalten und akzeptieren.

Enthält die Menge X jedoch weniger als ℓ Knoten, so können wir daraus noch nicht schließen, dass wir G verwerfen können, denn wir könnten ja in der Schleife die Knoten v für X einfach ungeschickt ausgewählt haben. Was wir jedoch auf jeden Fall wissen ist, dass

jeder rote Knoten von G in der $2r$-Nachbarschaft eines Elements aus X liegt

Daraus folgt auch, dass die $2r$-Nachbarschaft jedes roten Knotens in der $4r$-Nachbarschaft eines Elements aus X, also in der in Zeile 10 definierten Menge $N_{4r}^G(X)$ liegt

Da $|X| \leq \ell - 1$ ist und da G ein Graph vom Grad $\leq d$ ist, wissen wir, dass

$$|N_{4r}^G(X)| \leq (\ell-1) \cdot d^{4r+1}.$$

Daher hängt die Zeit, die für Zeile 11 des Algorithmus benötigt wird, also nicht von der Größe von G ab, sondern nur von einer Zahl $f_d^{(4)}(\ell, r)$ (für eine geeignete Funktion $f_d^{(4)} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$).

Insgesamt löst Algorithmus 2 damit die in Schritt 2 gestellte Aufgabe und benötigt dabei für die Zeilen 1–7 eine Laufzeit $\leq f_d^{(5)}(r) \cdot n$.

Für die Zeilen 8-12 wird eine Laufzeit benötigt, die nur von d, ℓ und r, nicht aber von der Größe von G abhängt.

Insgesamt liefert die Hintereinander-Ausführung der Algorithmen 1 und 2 einen Algorithmus, der bei Eingabe eines basis-lokalen Satzes χ der Größe k und eines Graphen G vom Grad $\leq d$ in Zeit $f_d(k) \cdot n$ testet, ob $G \models \chi$ (wobei $n = |V^G| + |E^G|$ und $f_d : \mathbb{N} \to \mathbb{N}$ eine geeignete Funktion ist). □

Ein weiteres Beispiel für eine Klasse von Strukturen, auf der das Auswertungsproblem für jeden festen FO-Satz in Linearzeit gelöst werden kann, ist:

Satz 4.20 (Frick, Grohe, 2001)

Es gibt eine berechenbare Funktion f, so dass das

> **Auswertungsproblem für FO auf planaren Graphen**
>
> Eingabe: Ein endlicher planarer Graph G und ein $FO[\{E\}]$-Satz φ
>
> Frage: Gilt $G \models \varphi$?

in Zeit $f(k) \cdot n$ gelöst werden kann, wobei k die Länge der Formel φ und n die Größe (einer geeigneten Kodierung) des Graphen G ist (etwa: $n = |V^G| + |E^G|$ für $G = (V^G, E^G)$).

Das heißt: Für jeden festen $FO[\{E\}]$-Satz φ ist das Problem, für gegebenen planaren Graphen G zu entscheiden, ob $G \models \varphi$, in Linearzeit lösbar.

Der Beweis von Satz 4.20 basiert
(a) auf dem Satz von Gallai und
(b) dem Konzept einer Baumzerlegung
eines Graphen, das in dieser Vorlesung
allerdings nicht genauer betrachtet wird.
Wir werden hier daher keinen Beweis von
Satz 4.20 angeben.

Um die Aussage von Satz 4.19 und Satz 4.20
zu verdeutlichen, betrachten wir kurz ein
konkretes Beispiel.
Für jede feste Zahl $k \in \mathbb{N}$ kann das Problem

k-Independent-Set

Eingabe: Ein endlicher Graph G

Frage: Gibt es in G eine unabhängige Menge der Größe k, d.h. eine Menge von k Knoten, zwischen denen es keine Kanten gibt?

durch einen Algorithmus gelöst werden, der Zeit $\Theta(n^k)$ benötigt (der Algorithmus testet einfach jede k-elementige Knotenmenge darauf, ob sie unabhängig ist).

Andererseits ist zunächst völlig unklar, ob es einen Algorithmus geben kann, der das Problem in Linearzeit, also in Zeit $O(n)$ löst.

Da das k-Independent-Set-Problem jedoch durch den $FO[\{E\}]$-Satz

$$\varphi := \exists x_1 \cdots \exists x_k \bigwedge_{1 \le i < j \le k} \left(\neg x_i = x_j \wedge \neg E x_i x_j \wedge \neg E x_j x_i \right)$$

definiert wird (in dem Sinn, dass für jeden Graphen G gilt: $G \models \varphi \Leftrightarrow G$ besitzt eine unabhängige Menge der Größe k), folgt aus Satz 4.19 und Satz 4.20, dass es Algorithmen gibt, die das k-Independent-Set-Problem in Zeit $O(n)$ lösen, wenn man als Eingabe-Graphen ausschließlich planare Graphen oder Graphen von beschränktem Grad zulässt.