Beweis von Theorem 1.19:

Gemäß Theorem 1.5 gibt es eine zu $\varphi(\bar{x})$ d-äquivalente HNF-Formel $\psi(\bar{x})$ für $FO(\mathbb{P})$ der Signatur σ mit $frei(\psi) = \{x_1, \dots, x_n\}$, und für $r := \frac{3^{qr(\varphi)} - 1}{2}$ und $\mathcal{L}_r^{\sigma,d}(n) = \tau_1, \dots, \tau_\ell$ gilt:

ψ ist eine Boolesche Kombination von Hanf-Zählsätzen $\chi_1, \dots, \chi_s$ und von Sphärenformeln $sph_{\tau_i, r}(\bar{x})$ mit $i \in [\ell]$ (für eine geeignete Zahl $s \in \mathbb{N}$).

Bei Eingabe einer σ-Struktur $\mathcal{A}$ vom Grad $\leq d$ nutzen wir für jedes $j \in [s]$ den Algorithmus von Theorem 1.18 um in Zeit $O(|\mathcal{A}|)$ zu testen, ob $\mathcal{A} \models \chi_j$.

Insgesamt können wir so in Zeit $O(|\mathcal{A}|)$ die Menge

$$J := \{ j \in [s] : \mathcal{A} \models \chi_j \}$$

berechnen.

Sei $\psi_J(\bar{x})$ die Formel, die aus $\psi(\bar{x})$ entsteht, indem jedes Vorkommen eines Satzes χ_j ersetzt wird durch

$$\begin{cases} \text{true}, & \text{falls } j \in J \\ \text{false}, & \text{falls } j \notin J \end{cases}$$

Gemäß Lemma 1.7 können wir dann in Zeit $O(1)$ eine Menge $I \subseteq [\ell]$ berechnen, so dass gilt:

$$\psi_J(\bar{x}) \equiv_d \bigvee_{i \in I} sph_{\tau_i, r}(\bar{x}). \qquad (*)$$

Für die Eingabestruktur $\mathcal{A}$ gilt dann:

$$
\begin{aligned}
[\![\varphi(\bar{x})]\!]^{\mathcal{A}} &= [\![\psi(\bar{x})]\!]^{\mathcal{A}} && (\text{da } \psi \equiv_d \varphi) \\
&= [\![\psi_\gamma(\bar{x})]\!]^{\mathcal{A}} && (\text{gemäß Wahl von } \gamma) \\
&= [\![\bigvee_{i\in I} sph_{\tau_i,r}(\bar{x})]\!]^{\mathcal{A}} && (\text{wegen } (*)) \\
&= \dot{\bigcup_{i\in I}} [\![sph_{\tau_i,r}(\bar{x})]\!]^{\mathcal{A}}.
\end{aligned}
$$

Also ist

$$
\left| [\![\varphi(\bar{x})]\!]^{\mathcal{A}} \right| = \sum_{i\in I} \left| [\![sph_{\tau_i,r}(\bar{x})]\!]^{\mathcal{A}} \right| ;
$$

und um den Beweis von Theorem 1.19 (a) und (b) zu beenden genügt es, die Aussagen (a) und (b) für den Spezialfall zu beweisen, dass $\varphi(\bar{x})$ von der Form $sph_{\tau,r}(\bar{x})$ für $r \in \mathbb{N}$ und $\tau \in \mathcal{L}_r^{\sigma,d}(n)$ ist.

Diesen Spezialfall betrachten wir im Folgenden.

Um den Beweis von Theorem 1.19 abzuschließen genügt es also, im Folgenden nur noch den Spezialfall zu betrachten, in dem $\varphi(\bar{x})$ von der Form $\mathrm{sph}_{\tau,r}(\bar{x})$ ist.

Wir betrachten zunächst <u>den Fall, dass τ zusammenhängend ist.</u>

In diesem Fall können wir wie folgt vorgehen:

1) Setze $R := r + (n-1)\cdot(2r+1)$
2) Initialisiere $M := \emptyset$, count $:= 0$
3) Für jedes $a_1 \in A$ tue Folgendes:
   3.1) Berechne $N_R^{\mathcal{A}}(a_1)$
   3.2) Probiere alle Tupel $(a_2,\dots,a_n) \in \left(N_R^{\mathcal{A}}(a_1)\right)^{n-1}$ durch und entscheide (durch ein brute-force-Verfahren), ob
   $$\left(\mathcal{N}_r^{\mathcal{A}}(a_1,a_2,\dots,a_n),\ a_1,a_2,\dots,a_n\right) \cong \tau \text{ ist};$$
   und falls ja, füge das Tupel $(a_1,a_2,\dots,a_n)$ in M ein und erhöhe count um 1
4) Gib count aus und zähle M auf

Man sieht leicht (Details: Übung!), dass dieser Algorithmus nach Zeit $O(|A|)$ die Zahl $|[\![sph_{\tau,r}(\bar{x})]\!]^{\mathcal{A}}|$ ausgibt und mit konstanter Taktung die Menge $M = [\![sph_{\tau,r}(\bar{x})]\!]^{\mathcal{A}}$ ausgibt.

Im Folgenden betrachten wir den Fall, dass τ aus $c \geq 2$ Zusammenhangkomponenten besteht.

Sei $\tau = (\mathcal{T}, \bar{t})$ mit $\bar{t} = (t_1, \dots, t_n) \in T^n$

F.a. $i \in [c]$ sei C_i die Knotenmenge der i-ten Zusammenhangkomponente von $\mathcal{T}$ (d.h. $T = C_1 \dot\cup \dots \dot\cup C_c$) und sei $\tau_i = (\mathcal{T}[C_i], \bar{t}_i)$, wobei $\bar{t}_i$ dasjenige Tupel ist, das aus $\bar{t}$ entsteht, indem alle Einträge t_j gelöscht werden, die nicht zu C_i gehören.

OBdA betrachten wir den Fall, dass es Zahlen $v_1, \dots, v_c \geq 1$ gibt, s.d. $\bar{t}_1 = (t_1, \dots, t_{v_1})$, $\bar{t}_2 = (t_{v_1+1}, \dots, t_{v_1+v_2}), \dots, \bar{t}_i = (t_{v_1+\dots+v_{i-1}+1}, \dots, t_{v_1+\dots+v_{i-1}+v_i}), \dots, \bar{t}_c = (t_{v_1+\dots+v_{c-1}+1}, \dots, t_n)$ ist (insbes: $n = v_1 + \dots + v_c$ und $v_i = |C_i|$ f.a. $i \in [c]$).

Analog sei $\bar{x}_i := (x_{v_1+\dots+v_{i-1}+1}, \dots, x_{v_1+\dots+v_{i-1}+v_i})$ f.a. $i \in [c]$.

Notation: Für Tupel $\overline{y} = (y_1, \dots, y_k)$ und $\overline{z} = (z_1, \dots, z_\ell)$ schreiben wir $\overline{y}\,\overline{z}$ um das Tupel $(y_1, \dots, y_k, z_1, \dots, z_\ell)$ zu bezeichnen.

Man sieht leicht (Details: Übung), dass für $\varphi(\overline{x}) := sph_{\tau, r}(\overline{x})$ das Folgende gilt: (*)$_1$

$$\llbracket \varphi(\overline{x}) \rrbracket^{\mathcal{A}} =$$

$$\Big\{ \overline{a}_1 \cdots \overline{a}_c : \quad \begin{array}{l} \overline{a}_i \in \llbracket sph_{\tau_i, r}(\overline{x}_i) \rrbracket^{\mathcal{A}} \quad \text{f.a. } i \in [c], \\ \text{und f.a. } i, j \in [c] \text{ und alle Einträge} \\ a_{i'} \text{ in } \overline{a}_i \text{ und } a_{j'} \text{ in } \overline{a}_j \text{ gilt:} \\ \quad dist^{\mathcal{A}}(a_{i'}, a_{j'}) > 2r+1 \end{array} \Big\}$$

An Stelle der Formel $\varphi(\overline{x})$ betrachten wir im Folgenden die Formel

$$\varphi_c(z_1, \dots, z_c) := \bigwedge_{i=1}^{c} C_i(z_i) \wedge \bigwedge_{\substack{i,j \in [c], \\ i \neq j}} \neg E(z_i, z_j)$$

(Stichwort: "rainbow-colored independent set") über der Signatur $\sigma_c := \{E, C_1, \dots, C_c\}$, wobei E ein 2-stelliges und $C_1, \dots, C_c$ 1-stellige Relationssymbole sind.

An Stelle der σ-Struktur $\mathcal{A}$ (vom Grad $\leq d$) betrachten wir die σ_c-Struktur $\mathcal{G} = (V, E^{\mathcal{G}}, C_1^{\mathcal{G}}, \dots, C_c^{\mathcal{G}})$, die wie folgt definiert ist:

- Für jedes $i \in [c]$ und jedes $\bar{a}_i \in [\![sph_{\tau_i, r}(\bar{x}_i)]\!]^{\mathcal{A}}$ sei $v_{\bar{a}_i}^i$ ein neuer Knoten.

- Für jedes $i \in [c]$ sei
$$C_i^{\mathcal{G}} := \{ v_{\bar{a}_i}^i : \bar{a}_i \in [\![sph_{\tau_i, r}(\bar{x}_i)]\!]^{\mathcal{A}} \}$$

- Sei $V := C_1^{\mathcal{G}} \dot\cup \cdots \dot\cup C_c^{\mathcal{G}}$ und
$$E^{\mathcal{G}} := \{ (v_{\bar{a}_i}^i, v_{\bar{a}_j}^j) : i, j \in [c],\ i \neq j,\ \text{ex } a_i' \text{ in } \bar{a}_i \text{ und } a_j' \text{ in } \bar{a}_j \text{ s.d. } dist^{\mathcal{A}}(a_i', a_j') \leq 2r+1 \}$$

Behauptung 1: Die Abbildung $\pi : C_1^{\mathcal{G}} \times \cdots \times C_c^{\mathcal{G}} \to A^n$ mit $\pi(v_{\bar{a}_1}^1, \dots, v_{\bar{a}_c}^c) := \bar{a}_1 \cdots \bar{a}_c$ f.a. $v_{\bar{a}_1}^1 \in C_1^{\mathcal{G}}, \dots, v_{\bar{a}_c}^c \in C_c^{\mathcal{G}}$ ist injektiv; und π eingeschränkt auf $[\![\psi_c(z_1, \dots, z_c)]\!]^{\mathcal{G}}$ ist eine Bijektion zwischen $[\![\psi_c(z_1, \dots, z_c)]\!]^{\mathcal{G}}$ und $[\![\psi(\bar{x})]\!]^{\mathcal{A}}$.

Beweis: Übung.

Behauptung 2: Bei Eingabe von $\mathcal{A}$ kann $\mathcal{G}$ in Zeit $O(|A|)$ erzeugt werden; und der Grad d' von $\mathcal{G}$ hängt von d, τ, r, n aber *nicht* von $|A|$ ab.

Beweisidee: Für jedes $i \in [c]$ berechne

$[\![sph_{\tau_i, r}(\bar{x}_i)]\!]^{\mathcal{A}}$ (das geht in Zeit $O(|A|)$, da τ_i zusammenhängend ist).

Für jedes $\bar{a}_i \in [\![sph_{\tau_i, r}(\bar{x}_i)]\!]^{\mathcal{A}}$ erzeuge einen neuen Knoten $v^i_{\bar{a}_i}$ und füge ihn in $C_i^{\mathcal{G}}$ ein.

So können wir $C_1^{\mathcal{G}}, \dots, C_c^{\mathcal{G}}$ und $V := C_1^{\mathcal{G}} \cup \dots \cup C_c^{\mathcal{G}}$ in Zeit $O(|A|)$ erzeugen.

Um $E^{\mathcal{G}}$ in Zeit $O(|A|)$ zu erzeugen, nutze, dass für alle $i, j \in [c]$ mit $i \neq j$ und alle $\bar{a}_i \in [\![sph_{\tau_i, r}(\bar{x}_i)]\!]^{\mathcal{A}}$ und $\bar{a}_j \in [\![sph_{\tau_j, r}(\bar{x}_j)]\!]^{\mathcal{A}}$ gilt:

- jeder Eintrag $a_{i'}$ in $\bar{a}_i$ hat in $\mathcal{A}$ den Abstand $\leq (2r+1) \cdot \nu_i$ zum ersten Eintrag $a_{i,1}$ in $\bar{a}_i$ (wobei ν_i die Länge des Tupels $\bar{a}_i$ ist) (und analog für $a_{j'}, \bar{a}_j, \nu_j, a_{j,1}$)

- Falls $(v^{i}_{\bar{a}_i}, v^{j}_{\bar{a}_j}) \in E^{\mathcal{G}}$, dann gilt insbes.:

  Es gibt ein $a_{i,l}$ in $\bar{a}_i$ und ein $a_{j,l'}$ in $\bar{a}_j$ s.d. $dist^{\mathcal{A}}(a_{i,l}, a_{j,l'}) \leq 2r+1$ ist —

  insbes ist $dist^{\mathcal{A}}(a_{j,l'}, a_{i,1}) \leq 2r+1 + (2r+1)\cdot v_i$,

  und jeder Eintrag in $\bar{a}_j$ ist in

  $N_R^{\mathcal{A}}(a_{i,1})$ für $R := (2r+1) + (2r+1)\cdot v_i + (2r+1)\cdot v_j$.

Nutze dies, um in Zeit $O(|A|)$ die Kantenmenge $E^{\mathcal{G}}$ zu erzeugen; und zeige, dass der Grad d' von $\mathcal{G}$ nur von d, τ, r, n abhängt, aber nicht von $|A|$. Details: Übung.

$\Box_{\text{Beh. 2}}$

Aus Behauptung 1 und Behauptung 2 ergibt sich, dass wir, um den Beweis von Theorem 1.19 abzuschließen, nur noch den folgenden Fall betrachten müssen:

Sei $d' \in \mathbb{N}$, $c \in \mathbb{N}_{\geq 1}$, $\sigma_c := \{E, C_1, \dots, C_c\}$;

$\overline{z} = (z_1, \dots, z_c)$ seien c verschiedene Variablen,

$$\varphi_c(\overline{z}) := \bigwedge_{i=1}^{c} C_i(z_i) \wedge \bigwedge_{\substack{i,j \in [c], \\ i \neq j}} \neg E(z_i, z_j)$$

("rainbow-colored independent set").

<u>Zu zeigen</u>: Bei Eingabe einer σ_c-Struktur $\mathcal{G} = (V, E^{\mathcal{G}}, C_1^{\mathcal{G}}, \dots, C_c^{\mathcal{G}})$ vom Grad $\leq d'$ kann $\text{COUNT}_{\varphi_c(\overline{z}), d'}$ in Zeit $O(|V|)$ gelöst werden, und $\text{ENUM}_{\varphi_c(\overline{z}), d'}$ kann nach $O(|V|)$ Vorverarbeitungszeit mit konstanter Taktung gelöst werden.

<u>Wir stellen nun die Lösung für $\boxed{\text{COUNT}_{\varphi_c(\overline{z}), d'}}$ vor:</u>

Für jede Eingabe $\mathcal{G}$ sieht $[\![\varphi_c(\overline{z})]\!]^{\mathcal{G}}$ wie folgt aus, wobei $\vartheta_{j,j'}(\overline{z}) := \left(\bigwedge_{i=1}^{c} C_i(z_i)\right) \wedge E(z_j, z_{j'})$ sei:

$$[\![\varphi_c(\bar{z})]\!]^{\mathcal{G}} = \left(C_1^{\mathcal{G}} \times \cdots \times C_c^{\mathcal{G}}\right) \setminus \left(\bigcup_{\substack{j,j' \in [c], \\ j \neq j'}} [\![\theta_{j,j'}(\bar{z})]\!]^{\mathcal{G}} \right),$$

und es gilt:

$$\left| [\![\varphi_c(\bar{z})]\!]^{\mathcal{G}} \right| = \underbrace{\left(\prod_{i=1}^{c} \left| C_i^{\mathcal{G}} \right| \right)}_{=: N_1} - \underbrace{\left| \bigcup_{j \neq j'} [\![\theta_{j,j'}(\bar{z})]\!]^{\mathcal{G}} \right|}_{=: N_2}$$

<u>Klar:</u> N_1 kann in Zeit $O(|V|)$ berechnet werden.

Um N_2 zu berechnen, nutzen wir das Prinzip der Inklusion und Exklusion (kurz: P.I.E.), das für Mengen $M_1, \ldots, M_n$ besagt:

$$|M_1 \cup \cdots \cup M_n| = \sum_{\emptyset \neq K \subseteq \{1,\ldots,n\}} (-1)^{|K|+1} \cdot \left| \bigcap_{k \in K} M_k \right|$$

(Übung: Zeigen Sie, dass dies für alle $n \geq 1$ und alle endlichen Mengen $M_1, \ldots, M_n$ korrekt ist).

Für $J := \{ (j,j') : j,j' \in [c], \ j \neq j' \}$

gilt:

$$\left| \bigcup_{(j,j')\in J} [\![\theta_{j,j'}(\bar{z})]\!]^{\mathcal{G}} \right| \overset{\text{P.I.E.}}{=}$$

$$\sum_{\emptyset \neq K \subseteq J} (-1)^{|K|+1} \cdot \left| \bigcap_{(j,j')\in K} [\![\theta_{j,j'}(\bar{z})]\!]^{\mathcal{G}} \right| =$$

$$\sum_{\emptyset \neq K \subseteq J} (-1)^{|K|+1} \cdot \left| [\![\alpha_K(\bar{z})]\!]^{\mathcal{G}} \right| ,$$ wobei

$$\alpha_K(\bar{z}) := \left(\bigwedge_{i=1}^{c} C_i(z_i) \right) \wedge \left(\bigwedge_{(j,j')\in K} E(z_j, z_{j'}) \right)$$ sei.

<u>Beachte:</u> $|J|$ ist konstant $(< c^2)$. Somit genügt es zu zeigen, dass für jedes $K \subseteq J$ mit $K \neq \emptyset$ die Zahl $|[\![\alpha_K(\bar{z})]\!]^{\mathcal{G}}|$ in Zeit $O(|V|)$ berechnet werden kann.

Betrachte im Folgenden also ein beliebiges $K \subseteq J$ mit $K \neq \emptyset$.

Sei $H = (W, K)$ der gerichtete Graph mit Knotenmenge $W := [c]$ und Kantenmenge K.

Sei s die Anzahl der Zusammenhangskomponenten des Gaifman-Graphen von H. Für jedes $i \in [s]$ sei

W_i die Knotenmenge der i-ten Zusammenhangskomponente, sei $H_i := H[W_i]$, und schreibe K_i, um die Kantenmenge von H_i zu bezeichnen.

Für jedes $i \in [s]$ sei $v_i := |W_i|$.

OBdA sei $W_1 = \{1, \dots, v_1\}$, $W_2 = \{v_1+1, \dots, v_2\}$, ..., $W_\ell = \{v_1 + \dots + v_{\ell-1}+1, \dots, v_1 + \dots + v_{\ell-1} + v_\ell\}$ f.a. $\ell \in [s]$.

Analog sei $\overline{z_1} := (z_1, \dots, z_{v_1})$, $\overline{z_2} := (z_{v_1+1}, \dots, z_{v_2})$ und allgemein $\overline{z_\ell} := (z_{v_1+\dots+v_{\ell-1}+1}, \dots, z_{v_1+\dots+v_{\ell-1}+v_\ell})$ f.a. $\ell \in [s]$.

Dann ist $\overline{z} = (z_1, \dots, z_c) = \overline{z_1} \cdots \overline{z_s}$, und es gilt

$$[\![\alpha_K(\overline{z})]\!]^g \underset{\text{Warum? – Übung!}}{=} [\![\alpha'_{K_1}(\overline{z_1})]\!]^g \times \dots \times [\![\alpha'_{K_s}(\overline{z_s})]\!]^g,$$

wobei f.a. $\ell \in [s]$

$$\alpha'_{K_\ell}(\overline{z_\ell}) := \Big(\bigwedge_{i \in W_\ell} C_i(z_i) \Big) \wedge \Big(\bigwedge_{(j,j') \in K_\ell} E(z_j, z_{j'}) \Big).$$

Es bleibt also nur noch zu zeigen, dass für jedes $\ell \in [s]$ die Zahl $|[\![\alpha'_{K_\ell}(\overline{z_\ell})]\!]^g|$ in Zeit $O(|V|)$ berechnet werden kann.

Betrachte dazu ein beliebiges $\ell \in [s]$.

Beachte, dass $H_\ell = (W_\ell, K_\ell)$ zusammenhängend ist.

Daher gilt für jedes Tupel

$$(a_1, \ldots, a_{r_\ell}) = \bar{a}_\ell \in [\![\alpha'_{K_\ell}(\bar{z}_\ell)]\!]^{\mathcal{G}},$$

dass $a_2, \ldots, a_{r_\ell} \in N_R^{\mathcal{G}}(a_1)$ für $R := r_\ell$ ist

(Begründung: Übung!).

Daher können wir zur Berechnung von $[\![\alpha'_{K_\ell}(\bar{z}_\ell)]\!]^{\mathcal{G}}$ wie folgt vorgehen:

1.) Für jedes $a_1 \in V$ berechne $N_R^{\mathcal{G}}(a_1)$ (diese Menge enthält höchstens $(d')^{R+1}$ Knoten, da $\mathcal{G}$ Grad $\leq d'$ hat)

und berechne die Menge

$$M_{a_1} := \{ (a_2, \ldots, a_{r_\ell}) \in \left(N_R^{\mathcal{G}}(a_1)\right)^{r_\ell - 1} :$$

$$\mathcal{G} \models \alpha'_{K_\ell}[a_1, a_2, \ldots, a_{r_\ell}] \}$$

(für jedes feste $a_1 \in V$ geht das in konstanter Zeit)

und berechne die Zahl $m_{a_1} := |M_{a_1}|$.

2) Setze $m_{K_\ell} := \sum_{a_1 \in V} m_{a_1}$

All dies geht in Zeit $O(|V|)$, und es gilt:

$M := \bigcup_{a_1 \in V} M_{a_1} = [\![\alpha'_{K_\ell}(\bar{z}_\ell)]\!]^{\mathcal{G}}$ und $m_{K_\ell} = |M| = |[\![\alpha'_{K_\ell}(\bar{z}_\ell)]\!]^{\mathcal{G}}|$

(Begründung: Übung!)

Insgesamt erhalten wir dadurch einen Algorithmus, der $\text{COUNT}_{\varphi_c(\bar{z}),d'}$ in Zeit $O(|V|)$ löst.

Um den Beweis von Theorem 1.19 abzuschließen, müssen wir nur noch einen Algorithmus finden, der $\boxed{\text{ENUM}_{\varphi_c(\bar{z}),d'}}$ nach $O(|V|)$ Vorverarbeitungszeit mit konstanter Taktung löst.

Zur Erinnerung: $\varphi_c(\bar{z}) = \bigwedge_{i=1}^{c} C_i(z_i) \wedge \bigwedge_{j \neq j'} \neg E(z_j, z_{j'})$.

Als Eingabe erhalten wir eine σ_c-Struktur
$$\mathcal{G} = (V, E^{\mathcal{G}}, C_1^{\mathcal{G}}, \ldots, C_c^{\mathcal{G}})$$ vom Grad $\leq d'$.

1\. Idee zur Aufzählung von $[\![\varphi_c(\bar{z})]\!]^{\mathcal{G}}$:

```
for all a_1 ∈ C_1^G do
   for all a_2 ∈ C_2^G do
      if (a_1,a_2) ∉ E^G and (a_2,a_1) ∉ E^G
      then for all a_3 ∈ C_3^G do
         if (a_i,a_3) ∉ E^G and (a_3,a_i) ∉ E^G f.a. i∈{1,2}
         then for all a_4 ∈ C_4^G do
            ⋱
                  for all a_c ∈ C_c^G do
                  if (a_i,a_c) ∉ E^G and (a_c,a_i) ∉ E^G f.a. i∈[c-1]
                  then output (a_1,a_2,...,a_c).
```

Gut: Dieser Algorithmus gibt genau diejenigen Tupel aus, die zu $[\![\varphi_c(\bar{z})]\!]^{\mathcal{G}}$ gehören — und zwar jedes davon genau einmal.

Schlecht: Wir haben keine Garantie darüber, dass zwischen der Ausgabe von je 2 Tupeln nur konstant viel Zeit (unabhängig von $|V|$) vergeht (Warum? — Übung!)

Betrachte z.B. den Fall $d' = 3$ und die Eingabe $\mathcal{G} = (V, E^{\mathcal{G}}, C_1^{\mathcal{G}}, C_2^{\mathcal{G}}, C_3^{\mathcal{G}}, C_4^{\mathcal{G}})$ (also $c = 4$) mit $C_1^{\mathcal{G}} = \{0, 1\}$, $C_2^{\mathcal{G}} = \{2, 3, 4, \dots, n+1\}$, $C_3^{\mathcal{G}} = \{n+2, n+3, \dots, 2n+1\}$, $C_5^{\mathcal{G}} = \{2n+2, 2n+3\}$ und $E^{\mathcal{G}} = \{(0, 2n+2), (0, 2n+3)\}$.

2. bessere Idee zur Aufzählung von $[\![\varphi_c(\bar{z})]\!]^{\mathcal{G}}$:

Berechne $I := \{ i \in [c] : |C_i^{\mathcal{G}}| \leq c \cdot d' \}$.

Die $C_i^{\mathcal{G}}$ mit $i \in I$ nennen wir "die kleinen Farben", während wir die $C_i^{\mathcal{G}}$ mit $i \in [c] \setminus I$ "die großen Farben" nennen.

Sei $k := |I|$ die Anzahl der kleinen Farben in $\mathcal{G}$.

Wir betrachten im Folgenden o.B.d.A den Fall, dass $I = \{ i \in [c] : i \leq k \}$ ist

(d.h.: $C_1^g, \ldots, C_k^g$ sind die kleinen Farben, und $C_{k+1}^g, \ldots, C_c^g$ sind die großen Farben;

beachte: Es könnte $k=0$ (also $I = \emptyset$) oder auch $k=c$ (also $I = [c]$) sein).

Der Begriff "große Farbe" ist so gewählt, dass Folgendes gilt:

Behauptung (*): F.a. $i \in [c-1]$ mit $i \geq k$ und alle Tupel $(a_1, \ldots, a_i) \in C_1^g \times \cdots \times C_i^g$ gilt:

es gibt (mind.) ein $a_{i+1} \in C_{i+1}^g$ s.d. f.a. $j \in [i]$ gilt: $(a_i, a_j) \notin E^g$ und $(a_j, a_i) \notin E^g$.

Beweis: Wir wissen, dass (der Gaifman-Graph von) $\mathcal{G}$ den Grad $\leq d'$ hat. Daher gibt es für jedes $j \in [i]$ höchstens d' verschiedene Knoten b mit $(a_j, b) \in E^g$ oder $(b, a_j) \in E^g$.

Wegen $i \geq k$ ist C_{i+1}^g eine große Farbe, d.h. $|C_{i+1}^g| \geq c \cdot d' + 1 \geq i \cdot d' + 1$. Somit gibt es in C_{i+1}^g mind. einen Knoten, der zu keinem der Knoten $a_1, \ldots, a_i$ benachbart ist

$\square$ Beh(*).

In der Vorverarbeitungsphase berechnen wir brute-force die Menge

$$M := \{ (a_1,\dots,a_k) \in C_1^{\mathcal{G}} \times \dots \times C_k^{\mathcal{G}} :$$
$$\text{f.a. } j,j' \in [k] \text{ mit } j \neq j' \text{ gilt } (a_j,a_{j'}) \notin E^{\mathcal{G}} \}.$$

Da $C_1^{\mathcal{G}}, \dots, C_k^{\mathcal{G}}$ kleine ~~Farben~~ sind, geht dies in Zeit, die nur von c und d', nicht aber von $|V|$ abhängt.

Zum Aufzählen aller Tupel in $[\![\varphi_c(\bar{z})]\!]^{\mathcal{G}}$ nutzen wir nun den folgenden Algorithmus:

```
for all (a_1,...,a_k) ∈ M do
  for all a_{k+1} ∈ C^G_{k+1} do
    if (a_i,a_{k+1}) ∉ E^G and (a_{k+1},a_i) ∉ E^G f.a. i∈[k]
    then for all a_{k+2} ∈ C^G_{k+2} do
           if (a_i,a_{k+2}) ∉ E^G and (a_{k+2},a_i) ∉ E^G f.a. i∈[k+1]
           then for all a_{k+3} ∈ C^G_{k+3} do
                  ...
                    for all a_c ∈ C^G_c do
                      if (a_i,a_c) ∉ E^G and (a_c,a_i) ∉ E^G f.a. i∈[c-1]
                      then output (a_1,a_2,...,a_c)
```

Deutlich hübscher lässt sich dieser Algorithmus rekursiv formulieren, nämlich wie folgt:

```
for all (a_1,...,a_k) ∈ M do
    ENUM(a_1,...,a_k)
```

mit der rekursiv definierten Funktion

```
function ENUM(a_1,...,a_i)

if i=c
then output (a_1,...,a_c)
else for all a_{i+1} ∈ G^g_{i+1} do
        if (a_{i+1},a_j) ∉ E^g and (a_j,a_{i+1}) ∉ E^g f.a. j∈[i]
        then ENUM(a_1,...,a_i,a_{i+1})
```

Behauptung (**): F.a. $i \in \{k, k+1, \ldots, c\}$ und alle Tupel $(a_1,\ldots,a_i) \in G_1^g \times \cdots \times G_i^g$ mit $(a_j,a_{j'}) \notin E^g$ f.a. $j,j' \in [i]$ mit $j \neq j'$ gilt: Beim Aufruf von ENUM$(a_1,\ldots,a_i)$ werden genau diejenigen Tupel $(b_1,\ldots,b_c)$ ausgegeben, für die gilt: $(b_1,\ldots,b_i) = (a_1,\ldots,a_i)$ und $(b_1,\ldots,b_c) \in [\![\varphi_c(\bar{z})]\!]^g$. Jedes dieser Tupel wird nur einmal ausgegeben. Vor der Ausgabe des ersten Tupels, zwischen der Ausgabe von zwei aufeinander folgenden Tupeln und nach der Ausgabe des letzten Tupels vergehen jeweils höchstens $O((c-i)\cdot(cd'+1))$ Berechnungsschritte.

<u>Beweis</u>: Per Induktion nach i.

<u>Induktionsanfang</u>: $i = c$ ✓

<u>Induktionsschritt</u>: $i+1 \to i$

Sei $(a_1, \ldots, a_i) \in C_1^{\mathcal{G}} \times \cdots \times C_i^{\mathcal{G}}$ mit $(a_j, a_{j'}) \notin E^{\mathcal{G}}$ f.a. $j, j' \in [i]$ mit $j \neq j'$.

Beim Aufruf von $\text{ENUM}(a_1, \ldots, a_i)$ wird nach höchstens $i \cdot d' + 1$ Schritten ein $a_{i+1} \in C_{i+1}^{\mathcal{G}}$ gefunden, s.d. $(a_{i+1}, a_j) \notin E^{\mathcal{G}}$ und $(a_j, a_{i+1}) \notin E^{\mathcal{G}}$ f.a. $j \in [i]$ gilt (denn: $\mathcal{G}$ hat Grad $\leq d'$, und $C_{i+1}^{\mathcal{G}}$ ist eine große Farbe, dh $|C_{i+1}^{\mathcal{G}}| \geq cd'+1 \geq id'+1$).

D.h: nach höchstens $id'+1 \leq c \cdot d' + 1$ Schritten wird ein rekursiver Aufruf $\text{ENUM}(a_1, \ldots, a_i, a_{i+1})$ gestartet. Gemäß Induktionsannahme zählt dieser mit Taktung $O\big((c-(i+1))(cd'+1)\big)$ alle Tupel $(b_1, \ldots, b_c) \in [\![\psi_c(\bar{z})]\!]^{\mathcal{G}}$ auf, für die gilt: $(b_1, \ldots, b_i, b_{i+1}) = (a_1, \ldots, a_i, a_{i+1})$.

Danach wird bei der weiteren Berechnung von $\text{ENUM}(a_1, \ldots, a_i)$ wieder nach max $id'+1 \leq cd'+1$ Schritten das nächste passende $a_{i+1}' \in C_{i+1}^{\mathcal{G}}$ gefunden, für das wiederum ein rekursiver Aufruf $\text{ENUM}(a_1, \ldots, a_i, a_{i+1}')$ gestartet wird usw., bis alle Elemente aus $C_{i+1}^{\mathcal{G}}$ abgearbeitet wurden.

Insgesamt erhält man, dass ENUM$(a_1, \dots, a_i)$ mit Taktung $O\left(cd'+1 + (c-(i+1))(cd'+1)\right)$

$= O\left((c-i)(cd'+1)\right)$ genau diejenigen Tupel $(b_1, \dots, b_c) \in [\![\varphi_c(\bar{z})]\!]^{\mathcal{G}}$ ausgibt, für die gilt: $(b_1, \dots, b_i) = (a_1, \dots, a_i)$.

$\square$ Beh $(**)$.

Die Aussage von Beh $(**)$ in Kombination mit der Wahl der Menge M liefert, dass unser Algorithmus mit Taktung $O(c^2 \cdot d')$ arbeitet und die Menge $[\![\varphi_c(\bar{z})]\!]^{\mathcal{G}}$ ausgibt.

Dies beendet den Beweis von Theorem 1.19

$\square_{\text{Thm } 1.19}$