## 1.3 Algorithmische Meta-Theoreme auf Grad-beschränkten Klassen von Strukturen

Ein Satz von Seese (1996) besagt, dass für jedes $d \in \mathbb{N}$ und jeden FO-Satz $\varphi$ das Auswertungsproblem für $\varphi$ auf Graphen vom Grad $\leq d$ in Linearzeit lösbar ist.

Unter Verwendung der schwachen Hanf-Normalform (Theorem 1.5) können wir dies auf die Logik $FO(\mathbb{P})$ verallgemeinern:

**Theorem 1.18**

Sei $\mathbb{P} \subseteq \mathcal{P}(\mathbb{Z})$ so dass für jedes $P \in \mathbb{P}$ gilt:
Bei Eingabe einer Zahl $n \in \mathbb{Z}$ kann in Zeit $O(|n|)$ entschieden werden ob $n \in P$ ist.

Sei $\sigma$ eine Signatur, sei $\varphi$ ein $FO(\mathbb{P})[\sigma]$-Satz und sei $d \in \mathbb{N}$.

Das Problem

$\boxed{EVAL_{\varphi,d}}$ (Auswertungsproblem für $\varphi$ auf Strukturen vom Grad $\leq d$)

**Eingabe:** Eine $\sigma$-Struktur $\mathcal{A}$ vom Grad $\leq d$

**Frage:** Gilt $\mathcal{A} \models \varphi$ ?

kann in Zeit $O(|\mathcal{A}|)$ gelöst werden.

Beweis:

Gemäß Theorem 1.5 gibt es einen zu $\varphi$ d-äquivalenten HNF-Satz $\psi$ für $FO(\mathbb{P})$ der Signatur $\sigma$. D.h.:

$\psi$ ist eine Boolesche Kombination von Hanf-Zählsätzen der Signatur $\sigma$ für $FO(\mathbb{P})$.

Um Theorem 1.18 zu beweisen genügt es also, einen beliebigen Hanf-Zählsatz $\chi$ der Signatur $\sigma$ für $FO(\mathbb{P})$ zu betrachten und zu zeigen, dass das Problem $EVAL_{\chi,d}$ in Zeit $O(\|\mathcal{A}\|)$ gelöst werden kann.

Sei im Folgenden $\chi$ von der Form

$$P\left(\sum_{\tau \in L} \#(y).\, sph_{\tau,r}(y) \;-\; m\right)$$

mit $r \in \mathbb{N}$, $m \in \mathbb{Z}$, $L \subseteq \mathcal{L}_r^{\sigma,d}(1)$, $P \in \mathbb{P} \cup \{\mathbb{N}_{\geq 1}\}$.

Sei $\mathcal{L}_r^{\sigma,d}(1) = \tau_1, \dots, \tau_\ell$. Sei $\{1 \leq j_1 < \dots < j_q \leq \ell\}$ s.d. $L = \{\tau_{j_1}, \dots, \tau_{j_q}\}$

Bei Eingabe einer $\sigma$-Struktur $\mathcal{A}$ vom Grad $\leq d$ geht der Algorithmus zum Lösen von $EVAL_{\chi,d}$ wie folgt vor:

1) f.a. $i \in \{1, \dots, \ell\}$ setze $anz_i := 0$

2) f.a. $a \in A$ tue Folgendes:

   2.1) Nutze Lemma 0.3 (b), um den r-Typ $\tau := (\mathcal{N}_r^{\mathcal{A}}(a), a)$ zu berechnen

   2.2) Nutze Lemma 0.4, um bei Eingabe von $\tau$ die Zahl $i \in [\ell]$ mit $\tau \cong \tau_i$ zu berechnen

   2.3) setze $anz_i := anz_i + 1$

3) Setze $n := \sum_{\nu=1}^{q} anz_{j_\nu} - m$

4) Entscheide, ob $n \in P$ ist;
falls ja, gib "$\mathcal{A} \models \chi$" aus
sonst gib "$\mathcal{A} \not\models \chi$" aus.

Man sieht leicht, dass der Algorithmus die korrekte Ausgabe liefert.
Zur Laufzeitanalyse beachte, dass $\ell, r, d, m, q$ Konstanten sind, die nicht von der Eingabe $\mathcal{A}$ abhängen (sondern nur vom Satz $\chi$).

Für jedes feste $a \in A$ ist die für 2.1) und 2.2) verwendete Laufzeit gemäß Lemma 0.3(b) und Lemma 0.4

$$v_d(r)^{O(\|\sigma\|)} \quad \text{und} \quad 2^{v_d(r)^{O(\|\sigma\|)}},$$

wobei $v_d(r) = 1 + d \cdot \sum_{i=0}^{r-1} (d-1)^i$ ist.
Somit wird für jedes feste $a \in A$ in den Zeilen 2.1), 2.2) und 2.3) nur konstant viel Zeit aufgewendet.
Für 2) wird insgesamt also Zeit $O(|A|)$ verwendet.
Die in Zeile 3) berechnete Zahl $n$ ist von der Größe $O(|A|)$, und gemäß unserer Voraussetzung an $P$ kann Zeile 4) daher in Zeit $O(|A|)$ gelöst werden. Insgesamt benötigt der Algorithmus in den Zeilen 1)–4) also Zeit $O(|A|)$.

□ Theorem 1.18

Unser nächstes Ziel ist, Theorem 1.18 auf geeignete Weise auf Formeln zu erweitern, die freie Variablen besitzen.

Sei $\bar{x} = (x_1, \dots, x_n)$ ein Tupel aus $n \geq 1$ verschiedenen Variablen und sei $\varphi$ eine $FO(\mathbb{P})[\sigma]$-Formel mit $frei(\varphi) \subseteq \{x_1, \dots, x_n\}$. Wir betrachten folgende Probleme:

$\boxed{COUNT_{\varphi(\bar{x}), d}}$ (Zählproblem für $\varphi(\bar{x})$ auf Strukturen vom Grad $\leq d$)

<u>Eingabe:</u> Eine $\sigma$-Struktur $\mathcal{A}$ vom Grad $\leq d$

<u>Ziel:</u> Berechne die Anzahl der Tupel $\bar{a} \in A^n$, für die gilt: $\mathcal{A} \models \varphi[\bar{a}]$.

$\boxed{ENUM_{\varphi(\bar{x}), d}}$ (Aufzählungsproblem für $\varphi(\bar{x})$ auf Strukturen vom Grad $\leq d$)

<u>Eingabe:</u> Eine $\sigma$-Struktur $\mathcal{A}$ vom Grad $\leq d$

<u>Ziel:</u> Gib nacheinander (ohne Duplikate) alle Tupel $\bar{a} \in A^n$ mit $\mathcal{A} \models \varphi[\bar{a}]$ aus.

Wir sagen:

„ $ENUM_{\varphi(\bar{x}), d}$ kann nach linearer Vorverarbeitung mit konstanter Taktung gelöst werden"

um auszudrücken, dass es einen Algorithmus gibt, der bei Eingabe einer $\sigma$-Struktur $\mathcal{A}$ vom Grad $\leq d$ in Zeit $O(|A|)$ eine Datenstruktur aufbaut, die es ermöglicht, nacheinander (ohne Duplikate) alle Tupel in $[\![\varphi(\bar{x})]\!]^{\mathcal{A}} := \{\bar{a} \in A^n : \mathcal{A} \models \varphi[\bar{a}]\}$, gefolgt von der Nachricht "EOE" ("end-of-enumeration") auszugeben, so dass die Wartezeit bis zur ersten Ausgabe und die Wartezeit zwischen zwei aufeinander folgenden Ausgaben $O(1)$ ist.

Unter Verwendung der schwachen Hanf-Normalform können wir Folgendes beweisen:

__Theorem 1.19__

Sei $\mathcal{P} \subseteq \mathcal{P}(\mathbb{Z})$ so dass für jedes $P \in \mathcal{P}$ gilt: Bei Eingabe einer Zahl $n \in \mathbb{Z}$ kann in Zeit $O(|n|)$ entschieden werden, ob $n \in P$ ist.
Sei $\sigma$ eine Signatur, sei $n \in \mathbb{N}_{\geq 1}$, sei $\bar{x} = (x_1, \ldots, x_n)$ ein Tupel von $n$ verschiedenen Variablen, sei $\varphi$ eine $FO(\mathcal{P})[\sigma]$-Formel mit $frei(\varphi) \subseteq \{x_1, \ldots, x_n\}$. Sei $d \in \mathbb{N}$.

__(a)__ $COUNT_{\varphi(\bar{x}),d}$ kann in Zeit $O(|A|)$ gelöst werden

__(b)__ $ENUM_{\varphi(\bar{x}),d}$ kann nach linearer Vorverarbeitung mit konstanter Taktung gelöst werden.

__Bemerkung:__ Aussage (b) für FO statt FO($\mathcal{P}$) wurde von Durand und Grandjean (2007) bewiesen. Die Aussagen (a) und (b) für FO+MOD statt FO($\mathcal{P}$) wurden von Berkholz, Keppeler, Schweikardt (2017) erzielt. Die Verallgemeinerung von Theorem 1.19 für FOC($\mathcal{P}$) statt FO($\mathcal{P}$) wurde von Kuske und Schweikardt (2017) erzielt.