1 Definitionen zum Abschnitt über FPT

Definition 1.1 *Eine* Parametrisierung *von* Σ^* *ist eine in Polynomialzeit berechenbare Abbildung* $\kappa : \Sigma^* \to \mathbb{N}$.

Definition 1.2 *Ein* parametrisiertes Entscheidungsproblem *(über dem Grundalphabet* Σ*) ist gegeben durch eine formale Sprache* $L \subseteq \Sigma^*$ *und eine Parametrisierung* κ *von* Σ^*.

Beispiel 1.3 *Es bezeichne* $\langle G, k\rangle \in \Sigma^*$ *das Codewort (gemäß einer vorgegebenen natürlichen Kodierung) für ein Paar* (G, k) *mit einem Graphen* $G = (V, E)$ *und einer natürlichen Zahl* k. *Wir setzen dabei der Einfachheit halber voraus, dass jeder String aus* Σ^* *als Codewort für ein Paar* (G, k) *interpretiert wird. Es sei* L *die Menge aller Codewörter für Paare* (G, k) *mit: es gibt eine Auswahl* $C \subseteq V$ *von* k *Knoten aus* V, *so dass jede Kante von* G *mindestens einen Randknoten in* C *hat. Weiter sei* $\kappa(\langle G, k\rangle) = k$. *Dann ist* (L, κ) *ein parametrisiertes Entscheidungsproblem (und offensichtlich eine parametrisierte Variante von „Vertex Cover“).*

Definition 1.4 *Ein* fpt-Algorithmus *für ein parametrisiertes Entscheidungsproblem* (L, κ) *ist ein Algorithmus, der das Wortproblem „*$x \in L$*?“ korrekt entscheidet und dessen Laufzeit durch eine Funktion der Form* $p(|x|) \cdot f(\kappa(x))$, *mit einem Polynom* p *und einer berechenbaren Funktion* f, *beschränkt werden kann.*

Definition 1.5 *FPT ist definiert als die Klasse aller parametrisierter Entscheidungsprobleme, welche einen fpt-Algorithmus besitzen.*

In der Vorlesung wurde gezeigt:

Bemerkung 1.6 *Vertex Cover, mit der in Beispiel 1.3 beschriebenen Parametrisierung, liegt in FPT.*

Definition 1.7 *Es seien* (L, κ) *und* (L', κ') *zwei parametrisierte Entscheidungsprobleme. Wir sagen* (L, κ) *ist* FPT-reduzierbar *auf* (L', κ'), *notiert als* $(L, \kappa) \leq_{fpt} (L', \kappa')$, *falls eine Abbildung* $R : \Sigma^* \to \Sigma^*$, *berechenbare Funktionen* f, g *und ein Polynom* p *existieren mit:*

1. *Für alle* $x \in \Sigma^*$ *gilt:* $x \in L \Leftrightarrow R(x) \in L'$.
2. $R(x)$ *ist in Zeit* $p(|x|) \cdot f(\kappa(x))$ *berechenbar.*
3. $\kappa'(f(x)) \leq g(\kappa(x))$.

Bemerkung 1.8 *Aus* $(L, \kappa) \leq_{fpt} (L', \kappa')$ *und* $(L', \kappa') \in FPT$ *folgt* $(L, \kappa) \in FPT$.

Beweis Es sei A' ein fpt-Algorithmus mit Zeitschranke $p'(|x|) \cdot f'(\kappa'(x))$ für ein Polynom p' und eine berechenbare Funktion f'. Wir dürfen annehmen, dass p' von der Form $p'(n) = n^c$ für eine Konstante c und dass f' eine monoton wachsende Funktion ist. Wir erhalten einen fpt-Algorithmus A' für (L, κ) wie folgt:

1. Transformiere Eingabe x in $x' = R(x)$. Aufwand: $p(|x|) \cdot f(\kappa(x))$ Rechenschritte.
2. Wende A' auf Eingabe x' an. Aufwand: $p'(|x'|) \cdot f'(\kappa'(x'))$ Rechenschritte.
3. Akzeptiere x genau dann, wenn A' den String x' akzeptiert.

Offensichtlich gilt:

$$|x'| \le p(|x|) \cdot f(\kappa(x)) \ , \ p'(|x'|) \le p(|x|)^c \cdot (f(\kappa(x))^c \ , \ f'(\kappa'(x')) \le f'(g(\kappa(x))) \ .$$

Die aus dieser Diskussion resultierende Zeitschranke weist A' als fpt-Algorithmus aus. •