# 20. Parameterized Complexity

**Goal:** Refine the complexity analysis of computational problems to find tractability within NP and beyond.

**Observation:**
- So far, we measure the complexity of a problem in terms of the size of the input.
- In real life, we also have information about the structure of the input:
  graphs are planar, at most $k$ threads, ...

**Idea:**
- Understand which parameter of the structure is to blame for the combinatorial explosion.
- If this parameter is small for common inputs (or even constant) we have practical tractability:

$$\underbrace{f(k)}_{k \text{ the parameter}} \cdot \underbrace{\text{poly}(n)}_{\text{polynomial in the size } n \text{ of the input.}}$$

$f(k)$ may be exponential

- So we allow for exponential aspects in the running times.

**Definition:**
A parameterized language is of the form $L \subseteq \Sigma^* \times \Sigma^*$.
If $(x,k) \in L$, we call $k$ the parameter.
Usually, the parameter will be from $\mathbb{N}$,
but it can also be a graph or an algebraic structure.
We will identify the domain of the parameter as $\mathbb{N}$ where appropriate and consider $L \subseteq \Sigma^* \times \mathbb{N}$.
For a fixed $k$, we call $L_k := L \cap (\Sigma^* \times \{k\})$ the $k$-th slice of $L$.

Note: · Classically, a decision problem took the form:

Given: The input.

Question: The condition to be checked.

· Now the problem specification has three parts:

Given: The input.

Parameter: The aspects of the input / the problem that constitute the parameter.

Question: The condition to be checked.

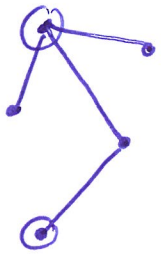
· Note that there may be different parameterized versions of the same decision problem.

Example:

· A vertex cover of a graph $G=(V,E)$ is a subset $V' \subseteq V$ of the vertices so that

for all $\{v_1,v_2\} \in E$: $\{v_1,v_2\} \cap V' \neq \emptyset$

(the vertices cover the edges).



· Phrased as a parameterized language, we obtain the problem VERTEXCOVER:

Given: A graph $G=(V,E)$.

Parameter: $k \in \mathbb{N}$.

Question: Does $G$ have a vertex cover of size $\leq k$?

Note that this is only one of the possible parameterizations.

· We will see that VERTEXCOVER, parameterized by $k$, can be solved in time $2^k \cdot |G|$.

· The idea of practical tractability, as proposed by parameterized complexity, is to find languages that are tractable by the size.

- Being tractable by the slice means
  there is a constant $c$ independent of $k$
  so that for all $k$
  membership in $L_k$ can be determined in $O(|x|^c)$.

Definition:

- A parameterized language $L \subseteq \Sigma^* \times \Sigma^*$ is fixed-parameter tractable (FPT),
  if there is a DTM $M$
  so that for all $(x,k) \in \Sigma^* \times \Sigma^*$:
  $(x,k) \in L$ iff $M((x,k)) = 1$.
  Moreover, $M$ runs in time at most $f(|k|) \cdot |x|^c$,
  for some $c \in \mathbb{N}$ and $f$ a computable function.
  $M$ is called a parameterized algorithm.
- If $k \in \mathbb{N}$, we do not use $|k|$ and no binary encoding.
  If $k \in \mathbb{N}$, it formally is $1^k$.
- Yes, $f(k) = \underbrace{2^{2^{\cdot^{\cdot^{2}}}}}_{k\text{-times}}$ is allowed.

Definition:

If a parameterized algorithm has running time $f(|k|) \cdot |x|^c$,
we write $O^*(f(|k|))$.
So we ignore the polynomial part and focus on the exponential part.