Parameter $k = \text{function: instance } \rightarrow \mathbb{N}$
- usually one of the numbers in instance
- sometimes hard to compute e.g. OPT

Parameterized problem = decision problem + parameter
- e.g. $(k)$-Vertex Cover: is there a vertex cover of \leq k?
  - $k$ is the natural parameter: comparing with OPT
- e.g. Vertex Cover with respect to OPT (Vertex Cover)
  - similar but $k$ not given
  - for $k = 0, 1, 2, \ldots$: run $k$-Vertex Cover
- e.g. Vertex Cover w.r.t. crossing number

$\text{XP} = \{\text{parameterized problems solvable in } n^{f(k)} \text{ time}\}$

Fixed-parameter tractable (FPT)
= \{\text{parameterized problems solvable in } f(k) \cdot n^{O(1)} \text{ time}\}
= \{\text{parameterized problems solvable in } f(k) + n^{O(1)} \text{ time}\}

- motivation: confine exponential to parameter $k$
  which may be $\ll$ problem size $n$

Example: $(k)$-Vertex Cover
- $\text{XP}$: guess $k$ vertices, test coverage $|V|^k \cdot |E|$
- $\text{FPT}$: take edge, guess endpoint, delete, repeat $2^k$ “bounded search tree technique” depth $\leq k$
EPTAS \in PTAS \text{ with running time } f(1/\varepsilon) \cdot n^{O(1)}
- \text{i.e. FPT w.r.t. } 1/\varepsilon
\Rightarrow \text{FPT w.r.t. natural parameter } k \ (\Rightarrow \text{w.r.t. OPT})
- \varepsilon FPT \Rightarrow \varepsilon \in \text{EPTAS}

\text{Parameterized reduction: } (A, k) \rightarrow (B, k')
- \text{instance } x \text{ of } A \Rightarrow \text{instance } x' = f(x) \text{ of } B
- f(k(x)) \cdot |x|^{O(1)} \text{ time } \Rightarrow |x'| \leq f(k(x)) \cdot |x|^{O(1)}
- \text{answer preserving: } x \text{ YES for } A \Leftrightarrow x' \text{ YES for } B
\quad \text{(just like NP/Karp reductions)}
- \text{parameter preserving: } k'(x') \leq g(k(x)) \quad \forall x
\quad \text{for some } g: \mathbb{N} \rightarrow \mathbb{N}
- B \in \text{FPT } \Rightarrow A \in \text{FPT}
\quad \overset{\text{parameter blowup}}{\uparrow}

\text{Nonexample: independent set } \rightarrow \text{vertex cover}
(G, k) \rightarrow (G, n-k)
- \text{preserves answer } \text{but not parameter}
- \text{indeed, vertex cover } \in \text{FPT}
\quad \text{but independent set is } W[1]-\text{hard}
\Rightarrow \varepsilon \text{ FPT unless } \text{FPT} = W[1]

\text{Example: independent set } \rightarrow \text{clique} \ (\text{or vice versa})
(G, k) \rightarrow (G', k)
Canonical hard problem for W[1]: (analogy to NP)
- k-step nondeterministic Turing machine
  - given nondeterministic Turing machine
    code, state, finger to k-cell memory?
  - O(n) lines, \(\Omega(n)\) options, \(\Omega(n)\) states
    (guess can have \(n\) choices/branches)
  - does some choice sequence finish in \(k\) steps?

Reduction to Independent Set:
- \(k^2\) cliques, \(k' = k^2 \Rightarrow 1\) node per clique
- clique \((i, j)\) represents memory cell \(i\) at time \(j\) (\(n\) choices) + state of machine
  (e.g. PC=which of \(n\) instructions next)
- add edges to forbid certain transitions
  \(j \rightarrow j'\): omit edges for allowed nondet. trans.

Reduction from Independent Set: \(k' = \Theta(k^2)\)
- guess \(k\) vertices
- for each pair of these vertices:
  - check no edge (lookup table in code)

\(\Rightarrow\) both W[1]-complete
Clique in regular graphs: reduction from Clique
- $\Delta = \max$ degree
- $\Delta$ copies of graph
- vertex $v$ of degree $d \Rightarrow v_1, v_2, \ldots, v_\Delta$ copies
  - add $\Delta - d$ vertices
  - biclique between $v$ &
  $\Rightarrow \Delta$-regular
- add no cliques ($\geq 3$):
  - new vertices in no $\Delta$

Independent set in regular graphs — just take complement

Partial vertex cover:
- are there $k$ vertices that cover $l$ edges?
- $\text{FPT w.r.t. } l$
- $\text{W}[1]$-complete w.r.t. $k$

Reduction from Independent set in regular graphs:
- $k' = \Delta k$

(based on upcoming book by
Cygan, Fomin, Kowalik, Lokshtanov, Marx, Pilipczuk, Pilipczuk,
Saurabh 2015: Parameterized Algorithms)
Multicolored clique: - like (Numerical) 3DM
- given graph & vertex k-coloring
- find k vertices, one of each color, that form a k-clique
- \(W[1]\)-complete \cite{Pietrzak-JCSS2003, Fellows-Hermelin-Rosamond-Vialette-TCS2009}

Reduction from Clique:
- vertex \(v \rightarrow k\) copies \(v_1, v_2, \ldots, v_k\)
- colors: \(1, 2, \ldots, k\)
- edge \((v, w) \rightarrow \text{edges } (v_i, w_j) \forall i \neq j\)
- \(k' = k\)
- \(k\)-clique \(\iff\) \(k\)-colored \(k\)-clique

Reduction to Clique:
- nothing: coloring \(\Rightarrow\) all cliques are multicolored

Multicolored independent set: - just take complement
Shortest common supersequence:
- given $k$ strings over alphabet $\Sigma_i$ & number $l$
- is there a common supersequence of length $l$
- $\mathcal{W}[1]$-hard w.r.t. $k$ for $|\Sigma| = 2$ [Pietrzak-JCSS2003]
- reduction from Multicolored Clique

Reduces to restricted form where input strings never repeat character twice in a row parameterized by $k$ & $\Sigma_i$
- add new symbol $s_i$ after every character in string $i$ $\Rightarrow$ no repeats
- $k' = k$
- $|\Sigma_i'| = |\Sigma_i| + k$
- $l' = l + \text{total length of input strings}$

Reduces to Flood-It on trees w.r.t. # colors ($|\Sigma_i|$) & # leaves ($k$)
Dominating set: (based on Cygan et al. book 2015)

Reduction from Multicolored independent set:
- vertex \(\rightarrow\) vertex
- connect each color class in clique
- also add 2 dummy vertices to each clique
- \(k'=k\) \(\Rightarrow\) dominating set chooses one vertex from each clique, representing one vertex of each color in ind. set
- for each edge \((v,w)\):
  - add vertex connected to all vertices in color classes of \(v\) \& \(w\), except \(v\) \& \(w\)
  \(\Rightarrow\) dominated \(\iff\) \(v\) \& \(w\) not both chosen (i.e. independent set)

\[ \Rightarrow W[1]-\text{hard} \]
- \(W[2]\)-complete in fact
\[ \Downarrow \text{\#FPT unless FPT }= W[2]\] (weaker assumption)
\[ \Downarrow \text{reverse reduction impossible unless } W[1]=W[2]\]

Reduction to Set Cover: same as \(L_{11}\)
- vertex \(v \rightarrow\) set \(N(v) \cup \exists v^3\) \(\quad - k'=k\)
**Weighted Circuit SAT** (Circuit k-Ones)
- given acyclic Boolean circuit & parameter k
- can we set k inputs to 1 to get output = 1?

\[ W[\mathcal{P}] = \{ \text{parameterized problems reducible to Weighted Circuit SAT} \} \]

- depth = longest input→output path
- weft = max # big gates on input→output path
  (not O(1) inputs: e.g. \( \geq 3 \) inputs)

\[ W[t] = \{ \text{parameterized problems reducible to O(1)-depth weft-t Weighted Circuit SAT} \} \]

\[ = \{ \text{parameterized problems reducible to depth-t output=AND Weighted Circuit SAT} \} \]

[Buss & Islam - TCS 2006]

\[ W[*] = W[O(1)] \]

**W[1]-complete:**
- weighted O(1)-SAT 
  (big AND of small ORs)

**W[2]-complete:**
- weighted CNF-SAT 
- k-step 2-finger nondeterministic Turing machine 
  = 2-tape 

\[ W[\text{SAT}] = \text{reducible to SAT} \]
- SAT → CNF-SAT reduction adds extra vars, so weighted problems not the same