Seminar paper, 5.Semester

Seminar Logic, Complexity, Games: Algorithmic Meta-Theorems and Parameterized Complexity

Hierarchies in parameterized complexity theory

Daniel Zilken
406837

Supervisor: Lovro Mrkonjic

Contents

1 Introduction

This seminar article is based on [1]. In the previous part, the class of Fixed-Parameter Tractable (FPT) problems has been introduced, as well as the concept of Fixed-Parameter Tractable Reductions. In this paper, we will get an overview of the hierarchies created by the various classes consisting of parameterized problems and expanding the FPT class. Most importantly, we will introduce the A-hierarchy and the W-hierarchy and see that the first levels of these hierarchies coincide.

2 Hierarchy of classes

2.1 para-NP and XP

The class FPT of parameterized problems shows similarities to the classical class P as the runtime is polynomial in the input size and both are pure deterministic. We want to find an analogue class to NP and therefore introduce the class para-NP resulting from allowing the algorithm in the definition of FPT to be nondeterministic.

Definition 1. *$(Q, \kappa) \in \mathsf{para\text{-}NP}$, if $N(x)$ decides $x \in Q$ in at most $f(\kappa(x)) \cdot p(|x|)$ steps for some nondeterministic algorithm N with given input $x \in \Sigma^*$, a computable function $f : \mathbb{N} \to \mathbb{N}$ and a polynomial $p \in \mathbb{N}_0[X]$.*

We introduce the p-Independent-Set problem. Clearly, this is an example of a problem in para-NP since we can nondeterministically guess an independent set and verify it in polynomial time.

p-Independent-Set
given: a graph G and a parameter $k \in \mathbb{N}$
decide: G having an independent set of cardinality k (pairwise non-adjacent vertices)

In addition to that, p-Independent-Set is also an example of a problem that is decidable in polynomial time, when fixing the parameter k. All parameterized problems with this property form the class XP.

Definition 2. *$(Q, \kappa) \in \mathsf{XP}$, if $A(x)$ decides $x \in Q$ in at most $p_{\kappa(x)}(|x|)$ steps for some algorithm A with given input $x \in \Sigma^*$ and computable function $p_{\kappa(x)}(|x|)$ that describes a polynomial $p_k(X)$ for every $k \geq 0$.*

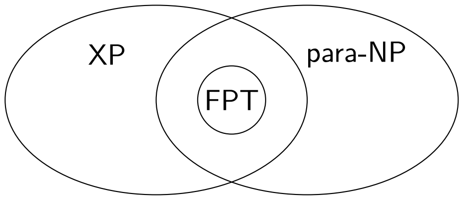


Fig. 1: The relations among the classes FPT, para-NP, and XP

2.2 W[P]

Another attempt for an analogue class of NP is to restrict the class para-NP in a way, that only a logarithmic amount of nondeterministic steps is allowed in the algorithm. We call a nondeterministic Turing machine M κ-restricted, if M performs at most $f(\kappa(x)) \cdot p(|x|)$ steps of which at most $f'(\kappa(x)) \cdot log(|x|)$ are nondeterministic for every input, for some computable functions $f, f' : \mathbb{N} \to \mathbb{N}$ and a polynomial $p \in \mathbb{N}_0[X]$.

Definition 3. $(Q, \kappa) \in \mathsf{W[P]}$*, if* $\mathsf{M}(x)$ *decides* $x \in Q$ *for some* κ*-restricted nondeterministic Turing machine* M *with given input* $x \in \Sigma^*$*.*

The main characteristic of each class is summarized in the following table:

	Runtime	Nondeterministic steps
FPT	$f(\kappa(x)) \cdot p(\|x\|)$	0
W[P]	$f(\kappa(x)) \cdot p(\|x\|)$	$f'(\kappa(x)) \cdot log(\|x\|)$
para-NP	$f(\kappa(x)) \cdot p(\|x\|)$	$f(\kappa(x)) \cdot p(\|x\|)$
XP	$p_{\kappa(x)}(\|x\|)$ (a possibly different polynomial for every $\kappa(x)$)	0

From here, the following inclusions are easy to see and justify the afterwards presented landscape.

1. FPT $\subseteq$ W[P] and W[P] $\subseteq$ para-NP.
   The required runtime for all three classes is the same, and the allowed amount of nondeterministic steps is just increasing from FPT over W[P] to para-NP.

2. W[P] $\subseteq$ XP.
   Note that $h(\kappa(x)) \cdot log(|x|)$ nondeterministic steps can be translated into $n^{O(h(\kappa(x)))}$ steps of a deterministic algorithm since the $log(|x|)$ in the exponent cancels out for any constant nondeterministic branching factor.

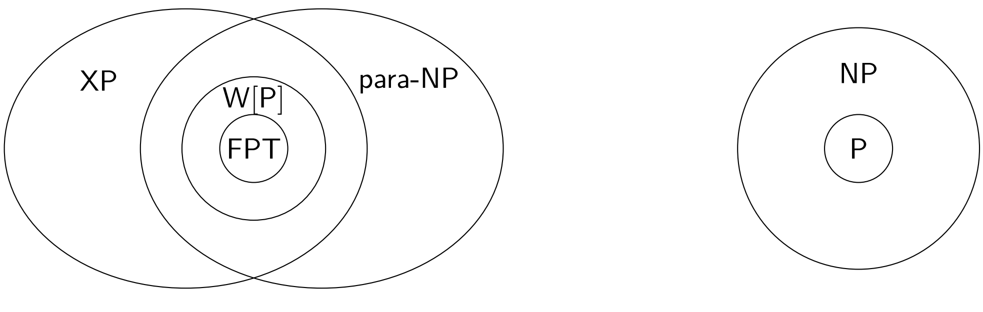


Fig. 2: Parameterized and classical hierarchy

2.3 Correlations to classical classes

In this section, we will see three propositions that connect the parameterized hierarchy to the classical complexity hierarchy. This will further express the analogies between the two hierarchies and relate multiple inclusions of the parameterized hierarchy to the $\mathsf{P} \neq \mathsf{NP}$ problem for classical classes.

Proposition 4. $\mathsf{FPT} = \mathsf{para\text{-}NP}$ *if and only if* $\mathsf{P} = \mathsf{NP}$.

Proof:
($\Rightarrow$): Assume $\mathsf{FPT} = \mathsf{para\text{-}NP}$.
Consider an arbitrary Problem $Q \subseteq \Sigma^*$ in NP. This means, a nondeterministic algorithm can decide $x \in Q$ in polynomial time $p(|x|)$. Note, this already is the definition of $\mathsf{para\text{-}NP}$ besides the prefactor $f(\kappa(x))$ which we can just add for the constant parameterization $\kappa(x) = 1$. Hence, $(Q, \kappa) \in \mathsf{para\text{-}NP}$ and together with our assumption also $(Q, \kappa) \in \mathsf{FPT}$. By that, a (deterministic) algorithm can decide $x \in Q$ in time $f(\kappa(x)) \cdot p(|x|) = f(1) \cdot p(|x|)$ which implies $Q \in \mathsf{P}$.

($\Leftarrow$): Assume $\mathsf{P} = \mathsf{NP}$.
For a parameterized problem $(Q, \kappa) \in \mathsf{para\text{-}NP}$ with alphabet Σ let A_Q be an algorithm deciding $x \in Q$ in $f(\kappa(x)) \cdot |x|^c$ nondeterministic steps with computable f and $c \in \mathbb{N}$. We will construct a related problem in NP. Note, that Q alone loses all information about the parameter and does not have to be in NP. Consider the following precomputation on the parameter to obtain a related problem in NP.
Let Π be the alphabet $\{1, \#\}$ and define $\pi : \mathbb{N} \to \Pi^*$ by $\pi(k) = k\#f(k)$ where k and $f(k)$ are written in unary. We obtain our NP-problem $X \subseteq \Sigma^* \times \Pi^*$ as the set of tuples accepted by the following algorithm A.
For an input $(x, y) \in \Sigma^* \times \Pi^*$, first A rejects all inputs of the wrong format, that is if the input does not fulfil $y = \kappa(x)\#u$ for some $u \in \{1\}^*$. If the format is correct, A simulates $|u| \cdot |x|^c$ steps of the computation of A_Q on input x. If A_Q and respectively A did not accept or reject in this time, A rejects. Notice, $|u| \leq |y|$ and by that A runs in nondeterministic polynomial time. The relation to our original parameterized problem is seen by the equivalence

$$x \in Q \textit{ iff } A \text{ accepts } (x, \kappa(x)\#f(\kappa(x))) \textit{ iff } (x, \pi(\kappa(x))) \in X.$$

By our assumption $P = NP$, there is also a polynomial time algorithm A' that decides X. Then, we can decide $x \in Q$ for our parameterized problem (Q, κ) by running A' on $(x, \pi(\kappa(x)))$ in deterministic time $g(\kappa(x)) \cdot |x|^c$. This shows $(Q, \kappa) \in \mathsf{FPT}$.

Proposition 5. *If* $\mathsf{P} \neq \mathsf{NP}$ *then* $\mathsf{para\text{-}NP} \not\subseteq \mathsf{XP}$.

The underlying idea of the proof is to conclude that an NP-complete problem is in P under the assumption $\mathsf{para\text{-}NP} \subseteq \mathsf{XP}$. A well known problem in NP is the colorability problem, that is, given a graph G and a $k \in \mathbb{N}$ decide whether G is k-colorable. This is the case if you can assign each node to one of the k colors such that in the end no adjacent nodes have the same color. Considering k as a parameter, we obtain the parameterized version of the problem para-colorable.

para-colorable
given: a graph G with parameter $k \in \mathbb{N}$
decide: G being k-colorable.

Proof: Assume $\mathsf{para\text{-}NP} \subseteq \mathsf{XP}$.
Clearly, para-colorable $\in \mathsf{para\text{-}NP}$ and with that para-colorable $\in \mathsf{XP}$. By definition of XP, we have that for fixed parameter $\kappa(x) = 3$, a polynomial algorithm can decide $x \in Q$. This means, the 3-colorability problem is in P. However, 3-colorability is well-known to be NP-complete.

3-colorability is just one example of many problems that are already NP-hard with a constant parameter. Another example is the satisfiability problem for propositional formulas in conjunctive normal form Sat(d-CNF) where the parameter d indicates the maximum number of literals in a clause of the formula.

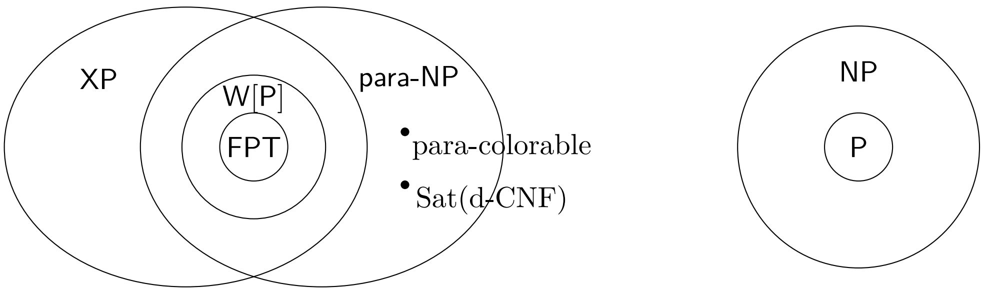


Fig. 3: para-colorable and Sat(d-CNF)

Proposition 6. *If* $\mathsf{FPT} \neq \mathsf{W[P]}$ *then* $\mathsf{P} \neq \mathsf{NP}$.

Proof: Assume $\mathsf{P} = \mathsf{NP}$.
By our proven proposition 2.1 we have $\mathsf{FPT} = \mathsf{para\text{-}NP}$ which leads to a collapse of our chain of inclusions $\mathsf{FPT} \subseteq \mathsf{W[P]} \subseteq \mathsf{para\text{-}NP}$ and thus $\mathsf{FPT} = \mathsf{W[P]}$.

3 W[t] and A[t]

For the well known First-Order Logic, we construct the following sets of formulas with quantifier alternation. Σ_0 and Π_0 denote the class of quantifier-free formulas. Inductively,
$\Sigma_{t+1} = \{\exists x_1 \dots \exists x_k \varphi \mid \varphi \in \Pi_t\}$ and $\Pi_{t+1} = \{\forall x_1 \dots \forall x_k \varphi \mid \varphi \in \Sigma_t\}$. Second-Order Logic adds quantification over subsets and relation of the universe, conversely denoted with large letters X. We construct similar sets of formulas. Σ_0^1 and Π_0^1 denote the class of all second-order formulas without any quantification over relation variables. Inductively, $\Sigma_{t+1}^1 = \{\exists X_1 \dots \exists X_k \varphi \mid \varphi \in \Pi_t^1\}$ and $\Pi_{t+1}^1 = \{\forall X_1 \dots \forall X_k \varphi \mid \varphi \in \Sigma_t^1\}$.
This in combination with the following Fagin-defined problem as well as the parameterized model-checking problem will lead us to the W- and A-hierarchy.

p-WD$_\varphi$

given: structure $\mathfrak{A}$ with parameter $k \in \mathbb{N}$

decide: $\mathfrak{A} \models \varphi(S)$ for some relation $S \subseteq A^s$ of cardinality $|S| = k$.

Furthermore, p-WD-Φ for a set Φ of first-order formulas denotes all parameterized problems p-WD$_\varphi$, where $\varphi \in \Phi$.

Example 7. $\varphi = \forall x \forall y (Sx \wedge Sy \wedge x \neq y \rightarrow Exy)$

Regarding this example, p-WD$_\varphi$ is the k-Clique problem. That is, given a graph $\mathfrak{A} = (V, E)$ decide whether it contains k nodes which are pairwise connected by an edge.

p-MC(Φ)

given: structure $\mathfrak{A}$ and formula $\varphi(\overline{x}) \in \Phi$ with parameter $|\varphi|$

decide: $\mathfrak{A} \models \varphi(\overline{a})$ for some $\overline{a} \subseteq A$

Definition 8
$\mathsf{W[t]} := [\mathsf{p\text{-}WD\text{-}}\Pi_t]^{\mathsf{fpt}} = \{P \mid P \leq_{\mathsf{FPT}} P' \text{ for some } P' \in \mathsf{p\text{-}WD\text{-}}\Pi_t\}$, the W-hierarchy for $t \geq 1$.
$\mathsf{A[t]} := [\mathsf{p\text{-}MC}(\Sigma_t)]^{\mathsf{fpt}} = \{P \mid P \leq_{\mathsf{FPT}} P' \text{ for some } P' \in \mathsf{p\text{-}MC}(\Sigma_t)\}$, the A-hierarchy for $t \geq 1$.

3.1 Inside the W-hierarchy

By definition, we have $\mathsf{W[t]} \subseteq \mathsf{W[t+1]}$ for all $t \geq 1$ since $\Pi_t \subseteq \Pi_{t+1}$. The following proposition shows that the class W[P] can be placed on top of the W-hierarchy.

Proposition 9. $\mathsf{W[t]} \subseteq \mathsf{W[P]}$ *for every* $t \geq 1$*.*

We prove this by showing that the whole set $[\mathsf{p\text{-}WD\text{-}FO}]^{\mathsf{fpt}}$ is contained in W[P]. Therefore, we start with an FO-formula $\varphi(X)$ where X is s-ary and construct an algorithm solving $\mathsf{p\text{-}WD}_\varphi$ with the required bound of nondeterministic steps. The algorithm is given a structure $\mathfrak{A}$ and a $k \in \mathbb{N}$ and proceeds as follows. Deterministically, check $\mathfrak{A} \models \varphi(S)$ for $S = \{a_1, \ldots, a_k\}$ for a nondeterministic choice of $a_1, \ldots, a_k \in A^s$. For the time complexity: Checking $\mathfrak{A} \models \varphi(S)$ can be done in polynomial time and for the nondeterministic choosing we need $\log(|A|)$ nondeterministic bits. Thus, $\mathsf{p\text{-}WD}_\varphi \in \mathsf{W[P]}$ which leads to our assumption since φ was arbitrary.

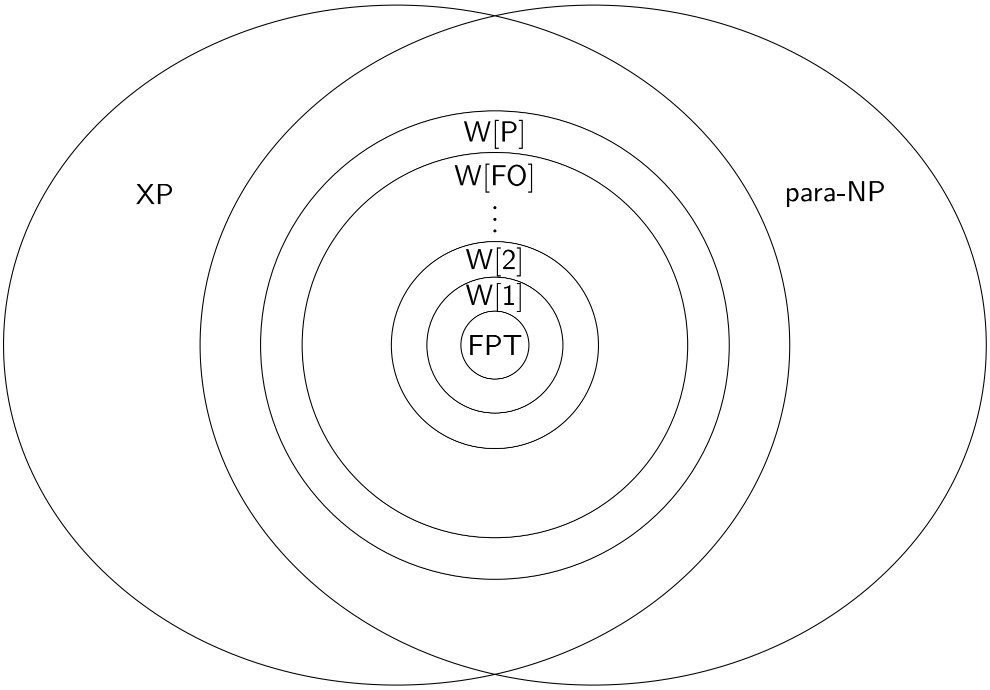


Fig. 4: W[P] and W[FO] placed on top of the W-hierarchy

3.2 Inside the A-hierarchy

First, note that, similar to the W-hierarchy, $\mathsf{A[t]} \subseteq \mathsf{A[t+1]}$ since $\Sigma_t \subseteq \Sigma_{t+1}$.

Proposition 10. $\mathsf{A[t]} \subseteq \mathsf{XP}$ *for every* $t \geq 1$*.*

We prove this by showing that the whole set $[\mathsf{p\text{-}MC(FO)}]^{\mathsf{fpt}}$ is an element of XP which is a corollary of the following theorem. The non parameterized Model Checking problem MC is the same as the p-MC just with omitted parameter.

Theorem 11. MC(FO) *can be solved in time* $O(|\varphi| \cdot |A|^w \cdot (w + |\varphi|))$ *where* w *denotes the width of* φ*, that is the maximum number of free variables of a subformula.*

Proof: Consider the model checking game $MC(\mathfrak{A}, \varphi)$. The positions of the game are pairs (ψ, β) for subformulas ψ of φ and β, a mapping from the free variables of ψ in the universe A. Since the size of the formula $|\varphi|$ can not be exceeded by the number of subformulas in φ and there are up to $|A|^w$ different possible mappings, the construction takes time $O(|\varphi| \cdot |A|^w)$. Furthermore, computing the winning regions of the game can be done by a variant of depth first search in linear time of the nodes and edges $O(|V| + |E|)$. The number of nodes is by construction also in $O(|\varphi| \cdot |A|^w)$ while the amount of edges is in $O(|\varphi| \cdot |A|^w \cdot (w + |\varphi|))$. For this, note that the amount of edges is equal to the amount of predecessors of all nodes. A node with subformula $\psi(x_1, \dots, x_k)$ and mapping $\beta : \{x_1, \dots, x_k\} \to A$ can only originate from two types of nodes:

1. Nodes with subformula $Qx_j\psi(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_k)$ and the restriction of β to $\{x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_k\}$ as mapping for $Q \in \{\exists, \forall\}$ and $j \in \{1, \dots, k\}$ which leads to a maximum of $2 \cdot k = O(w)$ predecessors.

2. Nodes with subformula $\delta(x_1, \dots, x_k) \circ \psi(x_1, \dots, x_k)$ and the unchanged mapping β for $\circ \in \{\wedge, \vee\}$ which leads to a maximum of $2 \cdot |\varphi| \in O(|\varphi|)$ predecessors since there can only be so many nodes with the fixed mapping β.

3.3 Relations between W and A

Proposition 12. $\mathsf{W[t]} \subseteq \mathsf{A[t+1]}$ *for every* $t \geq 1$.

Given a Π_t-formula $\varphi(X)$, we want to construct a Σ_{t+1}-formula φ_k such that $\mathfrak{A} \models \varphi(S)$ for some relation $S \subseteq A^s$ with $|S| = k$ if and only if $\mathfrak{A} \models \varphi_k$. Before proving the general case, let us go through the construction steps for the k-clique example.
Recall, $\varphi = \forall x \forall y (Sx \wedge Sy \wedge x \neq y \to Exy)$ and note that since this is a Π_1 formula, the k-clique problem is in $\mathsf{W[1]}$. We equivalently transform φ to

$$\varphi'_k = \forall x \forall y (\bigvee_{i \in \{1,\dots,k\}} x_i = x \wedge \bigvee_{i \in \{1,\dots,k\}} x_i = y \wedge x \neq y \to Exy).$$

Our final Σ_{t+1}-sentence, which shows that the k-clique problem is also in $\mathsf{A[2]}$ is

$$\varphi_k = \exists x_1 \dots \exists x_k (\bigwedge_{1 \leq i < j \leq k} x_i \neq x_j \wedge \forall x \forall y (\bigvee_{i \in \{1,\dots,k\}} x_i = x \wedge \bigvee_{i \in \{1,\dots,k\}} x_i = y \wedge x \neq y \to Exy)).$$

Proof: Let $\varphi(X)$ be given. To replace the free relation variable X, we introduce s-tuples $\overline{x}_1, \dots, \overline{x}_k$ as new variables with the intended meaning that these are precisely all the members of the (old) relation X. Therefore, we construct an equivalent formula φ' by replacing each occurrence of $X\overline{y}$ in φ by expressing, that at least one of the tuples $\overline{x}_1, \dots, \overline{x}_k$ is equal to $\overline{y}$:

$$\varphi' = \varphi[X\overline{y} / \bigvee_{i \in \{1,\dots,k\}} \overline{x}_i = \overline{y}].$$

It remains to quantify over these k tuples creating a sentence and expressing that they have to be pairwise different. This leads to our final Σ_{t+1}-sentence, φ_k, for $k \geq 1$:

$$\varphi_k = \exists \overline{x}_1 \dots \exists \overline{x}_k (\bigwedge_{1 \leq i < j \leq k} \overline{x}_i \neq \overline{x}_j \wedge \varphi').$$

Note, that the inequality $\overline{x}_i \neq \overline{x}_j$ for tuples is defined as usual by saying that at least one index has to differ and can be written out to $\bigvee_{l \in [s]} x_{il} \neq x_{jl}$ (when the tuples are denoted by $\overline{x}_i = (x_{i1}, \dots, x_{is})$).

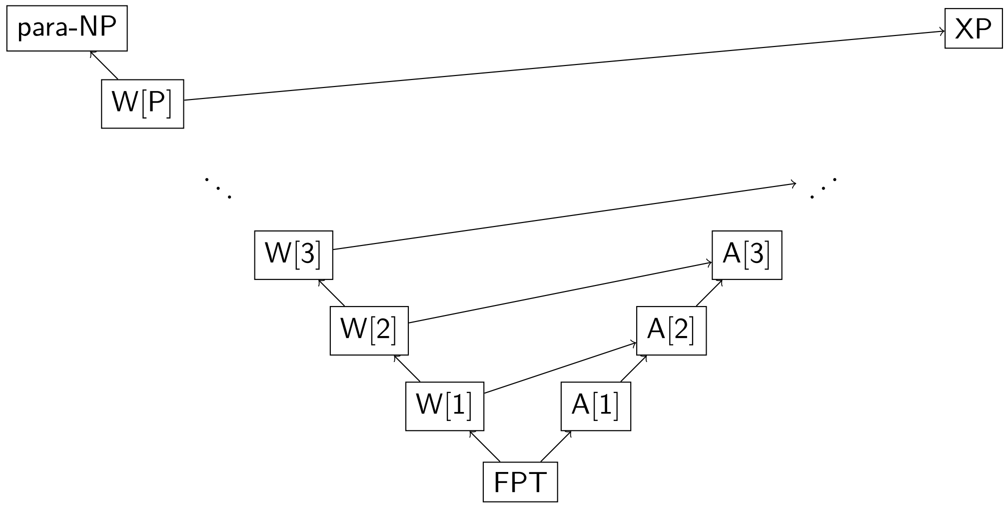


Fig. 5: Arrows indicate containment between classes

4 First level of W and A

4.1 Independent-Set and WSat

In this section, we prove that p-Independent-Set is A[1]-complete. Furthermore, we define the p-WSat(A) problem for a set of propositional formula A and prove that p-WSat(d-CNF) is a member of A[1].

Lemma 13. *p-Independent-Set is* A[1]*-complete.*

Proof: We have p-Independent-Set $\in$ A[1] since it is easy to see that p-Independent-Set with parameter k can be reduced to the parameterized model checking p-MC(Σ_1) via the Σ_1-sentence

$$\varphi_{IS} = \exists x_1 \dots \exists x_k (\bigwedge_{1 \leq i < j \leq k} x_i \neq x_j \wedge \bigwedge_{1 \leq i < j \leq k} \neg E x_i x_j).$$

For the A[1]-hardness, we show the following chain of reductions. Σ_1^+ denotes the class of Σ_1-formulas without negation symbols and $\Sigma_1[2]$ the class of Σ_1-formulas whose vocabulary τ is at most binary (no relations with arity ≥ 3).

$$\text{p-MC}(\Sigma_1) \overset{(1)}{\leq_{\mathsf{FPT}}} \text{p-MC}(\Sigma_1^+) \overset{(2)}{\leq_{\mathsf{FPT}}} \text{p-MC}(\Sigma_1[2]) \overset{(3)}{\leq_{\mathsf{FPT}}} \text{p-Independent-Set}$$

1. p-MC(Σ_1) $\leq_{\mathsf{FPT}}$ p-MC(Σ_1^+).
   Let $(\mathfrak{A}, \varphi)$ be an instance of p-MC(Σ_1). For the reduction, we need to construct a pair $(\mathfrak{A}', \varphi')$ such that $\mathfrak{A} \models \varphi$ *iff* $\mathfrak{A}' \models \varphi'$, where φ' is a Σ_1^+-formula. First, we know that φ is equivalent to a formula in negation normal form and therefore we only have to deal with subformulas of the form $\neg x = y$ and $\neg R x_1 \dots x_r$. We construct $\mathfrak{A}'$ as an expansion of $\mathfrak{A}$. We add an arbitrary linear order $<^{\mathfrak{A}'}$ of the universe A as well as for every r-ary relation R, two r-ary relations R_f, R_l and one 2r-ary relation R_s. In $R_f^{\mathfrak{A}'}$ ($R_l^{\mathfrak{A}'}$) only a single tuple is contained, the lexicographically first (last) element of $R^{\mathfrak{A}}$ with respect to $<^{\mathfrak{A}'}$. In $R_f^{\mathfrak{A}'}$ all consecutive tuples (direct successors) are contained, again based on the lexicographical order. The lexicographical order lifts our linear order $<^{\mathfrak{A}'}$ from elements to tuples and can be defined by

   $$<_{lex} (\overline{y}, \overline{z}) := \bigvee_{i \in \{1, \dots, r\}} (y_i < z_i \wedge \bigwedge_{j \in \{1, \dots, i-1\}} y_j = z_j).$$

   Now, we can see that our desired Σ_1^+-formula φ' can be obtained from φ by replacing subformulas of the form $\neg x = y$ by $(x < y \vee y < x)$ and subformulas of the form $\neg R x_1 \dots x_r$ by

   $$\exists y_1 \dots y_r \exists z_1 \dots \exists z_r ((R_f \overline{y} \wedge \overline{x} <_{lex} \overline{y}) \vee (R_s \overline{y}\overline{z} \wedge \overline{y} <_{lex} \overline{x} \wedge \overline{x} <_{lex} \overline{z}) \vee (R_l \overline{z} \wedge \overline{z} <_{lex} \overline{x})).$$

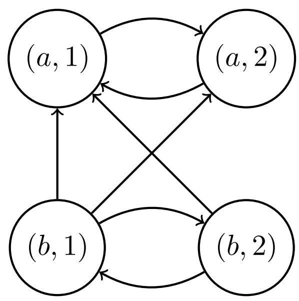
2. p-MC(Σ_1^+) $\leq_{\mathsf{FPT}}$ p-MC($\Sigma_1^+[2]$).
Let $(\mathfrak{A}, \varphi)$ be an instance of p-MC(Σ_1^+). For the reduction, we construct a pair $(\mathfrak{A}', \varphi')$ such that $\mathfrak{A} \models \varphi$ *iff* $\mathfrak{A}' \models \varphi'$, where φ' is a $\Sigma_1[2]$-formula. To deal with more than 2-ary relations, φ' is over a different vocabulary, τ'. For this, we replace every relation R by a unary relation P_R and binary relations $E_1, \dots, E_s$, where s is the arity of τ. The construction of $\mathfrak{A}'$ is done by the following three steps.
   - The universe expands A by new elements $b_{R,\overline{a}}$ for all relations R and $\overline{a} \in R^{\mathfrak{A}}$.
   - The relation $E_i^{\mathfrak{A}'}$ consists of all pairs $(a_i, b_{R,\overline{a}}$, where R has arity $r \geq i$ and $\overline{a} = (a_1, \dots, a_r) \in R^{\mathfrak{A}}$.
   - The relation $P_R^{\mathfrak{A}'}$ consists of all elements $b_{R,\overline{a}}$, where $\overline{a} \in R^{\mathfrak{A}}$.

   Now, we can see that our desired $\Sigma_1[2]$-formula φ' can be obtained from φ by replacing every atomic formula $Rx_1 \dots x_r$ by
$$\exists y (P_R y \wedge E_1 x_1 y \wedge \dots \wedge E_r x_r y).$$

3. p-MC($\Sigma_1^+[2]$) $\leq_{\mathsf{FPT}}$ p-Independent-Set.
Let $(\mathfrak{A}, \varphi)$ be an instance of p-MC($\Sigma_1[2]$). If φ is of the form $\exists x_1 \dots \exists x_k \bigwedge_{i \in I} t_i$ where the t_i are (negated) atomic formulas, we construct a graph $G(\mathfrak{A}, \varphi)$ depending on the input such that
$$\mathfrak{A} \models \varphi \textit{ iff } G(\mathfrak{A}, \varphi) \text{ contains an independent set of } k \text{ elements.}$$
As vertices, we take $V := A \times \{1, \dots, k\}$. For the edges, we start with all possible edges $E := V \times V$, but then omit an edge between two nodes (a, r) and (b, s) (for $a, b \in A$ and $1 \leq r < s \leq k$) if for all atomic formula $t_i = t_i(x_r, x_s)$ where only x_r and x_s occur $\mathfrak{A} \models t_i(a, b)$ holds. Ensuring $k \geq 2$ by possibly adding a dummy node, we have for all $a_1, \dots, a_k \in A$ that
$$\mathfrak{A} \models \bigwedge_{i \in I} t_i(a_1, \dots, a_k) \textit{ iff } \{(a_1, 1), \dots, (a_k, k)\} \text{ is an independent set in } G.$$
Since any independent set of G of cardinality k must contain an element (a, j) for every $j \in \{1, \dots, k\}$, this yields the desired equivalence.
If φ is a different formed $\Sigma_1[2]$-sentence, we first transform φ into an equivalent sentence φ' whose quantifier-free part is in disjunctive normal form, say $\varphi' = \exists x_1 \dots \exists x_k \bigvee_{j \in J} \bigwedge_{i \in I} t_{ji}$. But this is clearly equivalent to $\bigvee_{j \in J} \exists x_1 \dots \exists x_k \bigwedge_{i \in I} t_{ji}$ which for every $j \in J$ gives rise to a formula of our known form from the previous case. We combine all the graphs $G(\mathfrak{A}, \varphi_j)$ for $\varphi_j = \exists x_1 \dots \exists x_k \bigwedge_{i \in I} t_{ji}$ by adding edges between all pairs of nodes of different graphs to ensure that an independent set does not mix nodes from multiple graphs. This leads to our final Graph G. By the shown properties of the seperate graphs $G(\mathfrak{A}, \varphi_j)$ it is easy to see that G has an independent set of cardinality k *iff* $\mathfrak{A} \models \varphi$.

We give an example of the graph construction for the following input $(\mathfrak{A}, \varphi)$. In our example, $\mathfrak{A} \models \varphi$ holds and $G(\mathfrak{A}, \varphi)$ has an independent set of cardinality 2.

$$\mathfrak{A} := (\{a, b\}, R^{\mathfrak{A}} := \{a\})$$
$$\varphi := \exists x_1 \exists x_2 (x_1 \neq x_2 \wedge R x_1) \quad \Longrightarrow$$



The A[1]-completeness of p-Independent-Set as well as the now proven membership of p-WSat(d-CNF) in A[1] will be essential for the proof that the first levels of the W-hierarchy and the A-hierarchy coincide. The parameterized weighted satisfiability problem p-WSat(A) for a set of propositional formula A is defined as follow.

p-WSat(A)
given: $\alpha \in A$ and a parameter $k \in \mathbb{N}$
decide: α being k-satisfiable (there is a model with k variables set to true)

Lemma 14. *p-WSat(d-CNF)* $\in \mathsf{A}[1]$.

Proof: To prove the membership in $\mathsf{A}[1]$, we reduce p-WSat(d-CNF) $\leq_{\mathsf{FPT}}$ $\mathsf{p\text{-}MC}(\Sigma_1)$. Let (α, k) be an instance of p-WSat(d-CNF) with $\text{var}(\alpha) = \{X_1, \ldots, X_n\}$. For the reduction, we construct a pair $(\mathfrak{A}, \varphi)$ with a Σ_1-sentence φ such that α is k-satisfiable *iff* $\mathfrak{A} \models \varphi$. Assume our d-CNF formula is of the form $\alpha(X_1, \ldots, X_n) = \bigwedge_{i \in I} \delta_i$, where each δ_i is the disjunction of $\leq d$ literals. We construct the structure $\mathfrak{A}$ with universe $A = \{1, \ldots, n\}$ and for every $r \in \{1, \ldots, d\}$ the r-ary relations

- $R_r^{\mathfrak{A}} := \{(i_1, \ldots, i_r) \mid \neg X_{i_1} \vee \cdots \vee \neg X_{i_r}$ is a clause of $\alpha\}$;
- $S_r^{\mathfrak{A}} := \{(i_1, \ldots, i_r) \mid \neg X_{i_1} \vee \cdots \vee \neg X_{i_r} \vee X_{j_1} \vee \cdots \vee X_{j_s}$ is a clause of α for $s > 0$ $\}$.

In order to construct φ, we construct subformulas that deal with clauses of different amounts of negative and positive literals. Therefore, we define $\psi_\neg$ such that $\mathfrak{A} \models \psi_\neg(m_1, \ldots, m_k)$ *iff* $\{X_{m_1}, \ldots, X_{m_k}\}$ satisfies every clause of α with only negative literals. Furthermore, we define for all $r \in \{0, \ldots, d\}$, $\psi_{r,+}$ such that $\mathfrak{A} \models \psi_{r,+}(m_1, \ldots, m_k)$ *iff* $\{X_{m_1}, \ldots, X_{m_k}\}$ satisfies every clause with r negative literals and at least one positive literal. The first desired formula $\psi_\neg$ can immediately be constructed

$$\psi_\neg(x_1, \ldots, x_k) := \bigwedge_{r \in \{1, \ldots, d\}} \bigwedge_{1 \leq i_1, \ldots, i_r \leq k} \neg R_r x_{i_1} \ldots x_{i_r}.$$

For fixed $(i_1, \ldots, i_r) \in S_r^{\mathfrak{A}}$, we let $F = F(i_1, \ldots, i_r)$ be the following collection of subsets of A:

$$F(i_1, \ldots, i_r) := \{\{j_1, \ldots, j_s\} | \neg X_{i_1} \vee \cdots \vee \neg X_{i_r} \vee X_{j_1} \vee \cdots \vee X_{j_s} \text{ is a clause of } \alpha \text{ for } s > 0 \ \}.$$

Then, for $1 \leq m_1, \ldots, m_k \leq n$, the following statements are equivalent:

1. The assignment $\{X_{m_1}, \ldots, X_{m_k}\}$ (setting exactly $X_{m_1}, \ldots, X_{m_k}$ to true) satisfies all clauses in α of the form $\neg X_{i_1} \vee \cdots \vee \neg X_{i_r} \vee X_{j_1} \vee \cdots \vee X_{j_s}$ with $s > 0$.
2. Either the assignment $\{X_{m_1}, \ldots, X_{m_k}\}$ satisfies $\neg X_{i_1} \vee \cdots \vee \neg X_{i_r}$, or $\{m_1, \ldots, m_k\}$ is a hitting set of F, that is $\{m_1, \ldots, m_k\} \subseteq \bigcap_{f \in F} f$.

Let $H_1, \ldots, H_{d^k}$ be an enumeration of the minimal hitting sets of F (an algorithm therefore can e.g. be found in [1, chapter 1]). For $u = 1, \ldots, d^k$ and $l = 1, \ldots, k$ we add to $\mathfrak{A}$ the $(r+1)$-ary relations

$$L_{r,u,l}^{\mathfrak{A}} := \{(i_1, \ldots, i_r, m) \mid \text{m is the } l\text{th element of the } u\text{th hitting set } H_u \text{ of } F(i_1, \ldots, i_r)\}.$$

Now, we can define our desired formulas $\psi_{r,+}$. For $r = 0$, we have

$$\psi_{0,+}(x_1, \ldots, x_k) := \bigvee_{u \in \{1, \ldots, d^k\}} \bigwedge_{l \in \{1, \ldots, k\}} \bigwedge_{j \in \{1, \ldots, k\}} L_{0,u,l} x_j.$$

and for $r > 0$

$$\psi_{r,+}(x_1, \ldots, x_k) := \bigwedge_{1 \leq i_1, \ldots, i_r \leq k} (S_r x_{i_1} \ldots x_{i_r} \rightarrow \bigvee_{u \in \{1, \ldots, d^k\}} \bigwedge_{l \in \{1, \ldots, k\}} \bigwedge_{j \in \{1, \ldots, k\}} L_{r,u,l} x_{i_1} \ldots x_{i_r} x_j.$$

Finally, we obtain our desired formula as

$$\varphi = \exists x_1 \ldots \exists x_k (\bigwedge_{1 \leq i < j \leq k} x_i \neq x_j \wedge \psi_\neg \wedge \bigwedge_{r \in \{0, \ldots, d\}} \psi_r).$$

4.2 A[1] = W[1]

Proposition 15. $\mathsf{A}[1] \subseteq \mathsf{W}[1]$.

For this, we prove p-Independent-Set $\in \mathsf{W}[1]$ which immediately implies the lemma, since then every problem in $\mathsf{A}[1]$ can be reduced to a problem in $\mathsf{W}[1]$ due to the $\mathsf{A}[1]$-hardness of p-Independent-Set.

Proof: Consider the following problem in $\mathsf{W}[1]$: $\mathsf{p\text{-}WD}_{is}$ for the following Π_1-formula $is(X)$ that expresses the non-adjacent ness of all pairs of nodes in X

$$\text{is}(X) = \forall y \forall z((Xy \wedge Xz) \rightarrow \neg Eyz).$$

Clearly, solving $\mathsf{p\text{-}WD}_{is}$, i.e. deciding if there is a set X of cardinality k such that the graph structure is a model of $is(X)$, is equivalent to solving p-Independent-Set. This yields p-Independent-Set $\leq_{\mathsf{FPT}}$ $\mathsf{p\text{-}WD}_{is}$ and by that p-Independent-Set $\in \mathsf{W}[1]$.

Proposition 16. $\mathsf{W}[1] \subseteq \mathsf{A}[1]$.

For this, we prove that for every Π_1-formula $\varphi(X)$ there is a $d \geq 1$ such that $\mathsf{p\text{-}WD}_{\varphi} \leq_{\mathsf{FPT}}$ p-WSat(d-CNF) which immediately implies the lemma since then every problem in $\mathsf{W}[1]$ can be reduced to a problem in $\mathsf{A}[1]$ (due to p-WSat(d-CNF) being in $\mathsf{A}[1]$).

Proof: We assume that the formula $\varphi(X)$ is in conjunctive normal form, besides possible universal quantifiers at the front

$$\varphi(X) = \forall x_1 \ldots \forall x_r \bigwedge_{i \in I} \bigvee_{j \in J_i} \lambda_{ij}$$

Recall, that the input of $\mathsf{p\text{-}WD}_{\varphi}$ is a structure together with a parameter $(\mathfrak{A}, k)$. Now, we have to construct a propositional formula α in d-CNF out of this, such that $(\mathfrak{A}, k) \in \mathsf{p\text{-}WD}_{\varphi}$ if and only if α is k-satisfiable.
We will see that choosing $d = \max\{2, |J|\}$ suffices and for the construction of α we will start with $\varphi(X)$ and convert it into desired the d-CNF formula α in two essential steps. The first one will deal with the universal quantifiers our Π_1-formula does have and the second one with the free relation variable X. In both steps, the key of the construction is based on that the structure $\mathfrak{A}$ is fixed. We denote the universe of the structure with A.

1. To deal with the universal quantifiers, the idea is to obtain an equivalent formula by replacing all the $\forall x_i$ with $\bigwedge_{a_i \in A}$. Doing this for every quantifier and combining all the big conjunctions yields

   $$\varphi(X)' = \bigwedge_{a_1, \ldots, a_r \in A, i \in I} \bigvee_{j \in J_i} \lambda_{ij}$$

   Before going to step two, we will have a closer look at disjunctive clauses that have a literal λ_{ij} where the relation variable X is not contained. In case $\mathfrak{A} \not\models \lambda_{ij}(a_1, \ldots, a_r)$ the literal can be omitted since then the evaluation of $\mathfrak{A}$ on that clause only depends on the remaining disjuncts of the clause. In case $\mathfrak{A} \models \lambda_{ij}(a_1, \ldots, a_r)$ the whole clause can be omitted, since then $\mathfrak{A}$ is always a model of the clause and by that the evaluation of $\mathfrak{A}$ on the outer conjunction only depends on the remaining clauses.

2. To deal with the free relation variable X, we replace literals containing it or its negation, $(\neg)X_{x_{l_1} \ldots x_{l_s}}$ by $(\neg)Y_{a_{l_1} \ldots a_{l_s}}$. Here, $Y_{\overline{a}}$ is a (new) propositional variable expressing $\overline{a}$ is in X for $\overline{a} \in A^s$, where s is the arity of X.

So far, we have achieved a d-CNF formula that we denote α' for which by construction for all $S \subseteq A^s$

$$\mathfrak{A} \models \varphi(S) \text{ if and only if } \{Y_{\overline{a}} \mid \overline{a} \in S\} \text{ satisfies } \alpha'.$$

Finally, to ensure that the variable $Y_{\overline{a}}$ occurs for all $\overline{a} \in A^s$ and thus always has to be mapped by an interpretation either to 0 or 1 (similar to deciding $\overline{a} \in X$), we combine it with some tautology that contains all these variables. This leads to our final desired d-CNF formula

$$\alpha = \alpha' \wedge \bigwedge_{\overline{a} \in A^s} (Y_{\overline{a}} \vee \neg Y_{\overline{a}}).$$

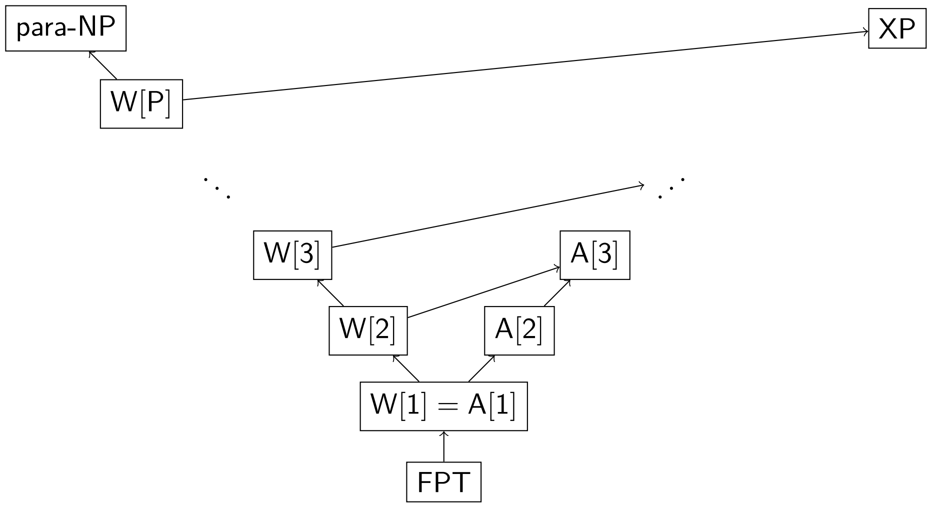


Fig. 6: Final landscape: A[1] and W[1] coincide

5 Conclusion

The most important result is the expansion of FPT to the W-hierarchy and the A-hierarchy, which relate parametrized complexity to logic. The resulting landscape is illustrated in figure 2 and shows that the first levels of both hierarchies coincide. The paper by Flum and Grohe continues by adding more classes to the hierarchy and showing that our inclusion result $\mathsf{A}[\mathsf{t}] \subseteq \mathsf{W}[\mathsf{t}+1]$ can even be strengthened to $\mathsf{A}[\mathsf{t}] \subseteq \mathsf{W}[\mathsf{t}]$.

References

[1] J. Flum and M. Grohe. Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series). Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[2] Prof. Dr. Erich Grädel and Dr. Wied Pakusa. Algorithmic Model Theory. Mathematische Grundlagen der Informatik, RWTH Aachen, 2019