Parameterized Complexity of Untangling Knots

Clément Legrand-Duchesne
Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400 Talence, France

Ashutosh Rai
Department of Mathematics, IIT Delhi, Hauz Khas, New Delhi, India

Martin Tancer
Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

Abstract

Deciding whether a diagram of a knot can be untangled with a given number of moves (as a part of the input) is known to be NP-complete. In this paper we determine the parameterized complexity of this problem with respect to a natural parameter called defect. Roughly speaking, it measures the efficiency of the moves used in the shortest untangling sequence of Reidemeister moves.

We show that the II^- moves in a shortest untangling sequence can be essentially performed greedily. Using that, we show that this problem belongs to W[P] when parameterized by the defect. We also show that this problem is W[P]-hard by a reduction from MINIMUM AXIOM SET.

2012 ACM Subject Classification Mathematics of computing → Geometric topology; Theory of computation → Problems, reductions and completeness

Keywords and phrases unknot recognition, parameterized complexity, Reidemeister moves, W[P]-complete

Digital Object Identifier 10.4230/LIPIcs.ICALP.2022.88

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2111.05001

Funding *Clément Legrand-Duchesne*: Part of this work has been done when C. L.-D. visited Charles University, whose stay was partially supported by the GAČR grant 19-04113Y.
Ashutosh Rai: Supported by SERB grant SRG/2021/002412.
Martin Tancer: Supported by the GAČR grant 19-04113Y.

1 Introduction

A classical and extensively studied question in algorithmic knot theory is to determine whether a given diagram of a knot is actually a diagram of an unknot. This question is known as the *unknot recognition problem.* The first algorithm for this problem was given by Haken [12]. Currently, it is known that the unknot recognition problem belongs to NP ∩ co-NP but no polynomial time algorithm is known. See [13] for the NP-membership and [19] for co-NP-membership (co-NP-membership modulo Generalized Riemann Hypothesis was previously established in [16]). In addition, a quasi-polynomial time algorithm for unknot recognition has been recently announced by Lackenby [20].

One possible path for attacking the unknot recognition problem is via Reidemeister moves (see Figure 2): if D is a diagram of an unknot, then D can be untangled to a diagram U with no crossing by a finite number of Reidemeister moves. In addition, Lackenby [17] provided a polynomial bound (in the number of crossings of D) on the required number of Reidemeister moves. This is an alternative way to show that the unknot recognition problem belongs to NP, because it is sufficient to guess the required Reidemeister moves for unknotting.

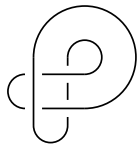

Figure 1 An example of a diagram.

However, if we slightly change our viewpoint, de Mesmay, Rieck, Sedgwick, and Tancer [23] showed that it is NP-hard to count the number of required Reidemeister moves exactly. (An anologous result for links has been shown to be NP-hard slightly earlier by Koenig and Tsvietkova [15].) More precisely, it is shown in [23] that given a digram D and a parameter k as input, it is NP-hard to decide whether D can be untangled using at most k Reidemeiser moves. For more background on unknotting and unlinking problems, we also refer to Lackenby's survey [18].

The main aim of this paper is to extend the line of research started in [23] by determining the parameterized complexity of untangling knots via Reidemeister moves. On the one hand, it is easy to see that if we consider parameterization by the number of moves, then the problem is in FPT (class of *fixed parameter tractable* problems). This happens due to a somewhat trivial reason: if a diagram D can be untangled with at most k moves, then D contains at most $2k$ crossings, thus we can assume that the size of D is (polynomially) bounded by k. In notions of parameterized complexity, this gives a kernel of size bounded by k which immediately gives the FPT membership. In the full version, we provide a bit more details; see Observation 1 in the full version.

On the other hand, we also consider parameterization with an arguably much more natural parameter called the defect (used in [23]). This parameterization is also relevant from the point of view of *above guarantee parameterization* introduced by Mahajan and Raman [21]. Here we show that the problem is W[P]-complete with respect to the defect. This is the core of the paper.

In order to state our main result more precisely, we need a few preliminaries on diagrams and Reidemeister moves. For purposes of this part of the introduction, we also assume that the reader is at least briefly familiar with complexity classes FPT and W[P]. Otherwise we refer to the end of the introduction, where we briefly overview these classes.

Diagrams and Reidemeister moves. A *diagram* of a knot is a piecewise linear map $D\colon S^1 \to \mathbb{R}^2$ in general position; for such a map, every point in $\mathbb{R}^2$ has at most two preimages, and there are finitely many points in $\mathbb{R}^2$ with exactly two preimages (called *crossings*). Locally at a crossing two arcs cross each other transversely, and the diagram contains the information of which arc passes "over" and which "under". This we usually depict by interrupting the arc that passes under. Diagrams are always considered up-to isotopy. The unique diagram without crossings is denoted U (untangled). See Figure 1 for an example of a diagram.

Let D be a diagram of a knot. *Reidemeister moves* are local modifications of the diagram depicted in Figure 2. We distinguish Reidemeister moves of types I, II, and III as depicted in the figure. In addition, for types I and II, we distinguish whether the moves remove crossings (types I^- and II^-) or whether they introduce new crossings (types I^+ and II^+).

A diagram D is a *diagram of an unknot* if it can be transformed to the untangled diagram U by a finite sequence of Reidemeister moves. (This is well known to be equivalent to stating that the lift of the diagram to $\mathbb{R}^3$, keeping the underpasses/overpasses, is ambient isotopic to

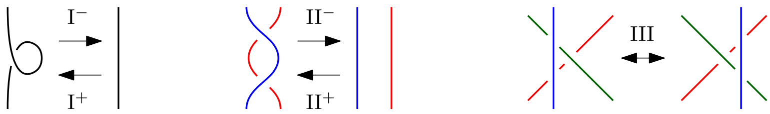

Figure 2 Reidemeister moves.

the unknot, that is standardly embedded S^1 in $\mathbb{R}^3$.) The diagram on Figure 1 is a diagram of an unknot. Diagrams may be encoded purely combinatorially as 4-regular plane graphs (with some additional combinatorial information) and the size of this encoding is comparable to the number of crossings. Similarly, the Reidemeister moves can be encoded by purely combinatorial data. For more details we refer to Sections 2 and 3 of the full version.

Parameterization via defect. As we discussed earlier, parameterization in the number of Reidemeister moves has the obvious disadvantage that once we fix k, the problem becomes trivial for arbitrarily large inputs (they are obviously a `NO` instance). We also see that if we have a diagram D with n crossings and want to minimize the number of Reidemeister moves to untangle D, presumably the most efficient way is to remove two crossings in each step, thus requiring at least $n/2$ steps. This motivates the following definition of the notion of defect.

Given a diagram D, by an *untangling* of D we mean a sequence $\mathcal{D} = (D_0, \dots, D_\ell)$ such that $D = D_0$; D_{i+1} is obtained from D_i by a Reidemeister move; and $D_\ell = U$ is the diagram with no crossings. Then we define the *defect* of an untangling $\mathcal{D}$ as above as

$$\operatorname{def}(\mathcal{D}) := 2\ell - n$$

where n is the number of crossings in D. Note that ℓ is just the number of Reidemeister moves in the untangling. It is easy to see that $\operatorname{def}(\mathcal{D}) \geq 0$ and $\operatorname{def}(\mathcal{D}) = 0$ if and only if all moves in the untangling are II^- moves. Therefore, $\operatorname{def}(\mathcal{D})$ in some sense measures the number of "extra" moves in the untangling beyond the trivial bound. (Perhaps, a more accurate expression for this interpretation would be $\ell - n/2 = \frac{1}{2}\operatorname{def}(\mathcal{D})$ but this is a minor detail and it is more convenient to work with integers.) In addition, it is possible to get diagrams with arbitrarily large number of crossings but with defect bounded by a constant (even for defect 0 this is possible). The defect also plays a key role in the reduction in [23] which suggests that the hardness of the untangling really depends on the defect.

As we have seen above, asking the question whether a diagram can be untangled with defect at most k is same as asking if it can be untangled in $k/2$ moves above the trivial, but tight lower bound of $n/2$. This fits perfectly in the framework of above guarantee parameterization, which was introduced by Mahajan and Raman [21] for Max-Sat and Max-Cut problems. In this framework, when there is a trivial lower bound for the solution in terms of the size of the input, parameterizing by solution size trivially gives an FPT algorithm by either giving a trivial answer if the input is large, or bounding size of the input by a function of the solution size. Hence, for those problems, it makes more sense to parameterize above a tight lower bound. The paradigm of above guarantee parameterization has been very successful in the field of parameterized complexity and many results have been obtained [1, 4, 5, 9, 10, 11, 22].

For these reasons, we find the defect to be a more natural parameter than the number of Reidemeister moves. Therefore, we consider the following problem.

▶ **Problem** (Unknotting via defect).

Input	A diagram D of a knot.
Parameter	k.
Question	Can D be untangled with defect at most k?

▶ **Theorem 1.** *The problem* Unknotting via defect *is W[P]-complete.*

The proof of Theorem 1 consists of two main steps: W[P]-membership and W[P]-hardness. Both of them are non-trivial.

For W[P]-membership, roughly speaking, the idea is to guess a small enough set of *special* crossings on which we perform all possible Reidemeister moves, while we remove other crossings in a greedy fashion. In order to succeed with such an approach we will need some powerful and flexible enough lemmas on changing the ordering of Reidemeister moves in some untangling by swapping them. In Section 2 we provide an algorithm for W[P]-membership but we only sketch why it works correctly. For more details, we refer to Sections 4 and 5 of the full version.

For W[P]-hardness, we combine some techniques that were quite recently used in showing parameterized hardness of problems in computational topology [2, 3], along with the tools in [23] for lower bounding the defect. Namely, we use a reduction from the Minimum axiom set problem, which proved to be useful in [2, 3]. Roughly speaking, from an instance I of the Minimum axiom set problem, we need to build a diagram which has a small defect if and only if I admits a small set of axioms. For the "if" part, we use properties of Brunnian rings to achieve our goal. For the "only if" part, we need to lower bound the defect of our construction. We use the tools from [23] to show that the defect (of some subinstances) is at least 1. Then we use the very simple but powerful boosting lemma (Lemma 9) that shows that the defect is actually high. We describe the reduction in Section 4 and we sketch in a bit more detail why it may work. We refer to the Section 6 of the full version for a proof of correctness.

We conclude this part of introduction by proving a lemma on the properties of the defect which we will use soon after. Given a Reidemeister move m, let us define the *weight* of this move $w(m)$ via the following table:

Type of the move	II^-	I^-	III	I^+	II^+
$w(m)$	0	1	2	3	4

▶ **Lemma 2** (Lemma 3 in full version). *Let $\mathcal{D}$ be an untangling of a diagram D. Then* $\mathrm{def}(\mathcal{D})$ *equals to the sum of the weights of the Reidemeister moves in $\mathcal{D}$.*

The lemma is proved in the full version by a simple induction.

A brief overview of the parameterized complexity classes. Here we briefly overview the notions from parameterized complexity needed for this paper. For further background, we refer the reader to monographs [6, 7, 8]. A *parameterized problem* is a language $L \subseteq \Sigma^* \times \mathbb{N}$, where Σ^* is the set of strings over a finite alphabet Σ and the input strings are of the form (x, k). Here the integer k is called *the parameter*. In the rest of the paper, given an input for a parameterized problem, n denotes the size of the input and k denotes the value of the parameter on this input.

A parameterized problem belongs to the class FPT (fixed parameter tractable) if it can be solved by a deterministic Turing machine in time $f(k) \cdot n^c$ where $c > 0$ is some constant and $f(k)$ is some computable function of k. In other words, if we fix k, then the problem can be solved in polynomial time while the degree of the polynomial does not depend on k. This is, of course, sometimes not achievable and there is a wider class XP of problems, that can be solved in time $O(n^{f(k)})$ by a deterministic Turing machine. The problems in XP are still polynomial time solvable for fixed k, however, at the cost that the degree of the polynomial depends on k.

Somewhere in between FPT and XP there is an interesting class W[P]. A parameterized problem belongs to the class W[P] if it can be solved by a nondeterministic Turing machine in time $h(k) \cdot n^c$ provided that this machine makes only $O(f(k) \log n)$ non-deterministic choices where $f(k), h(k)$ are computable functions and $c > 0$ is some constant. Given an algorithm for some computational problem Π, we say that this algorithm is a *W[P]-algorithm* if it is represented by a Turing machine satisfying the conditions above.

Given two parameterized problems Π and Π', we say that Π reduces to Π' via an *FPT-reduction* if there exist computable functions $f\colon \Sigma^* \times \mathbb{N} \to \Sigma^*$ and $g\colon \Sigma^* \times \mathbb{N} \to \mathbb{N}$ such that (i) $(x, k) \in \Pi$ if and only if $(f(x, k), g(x, k)) \in \Pi'$ for every $(x, k) \in \Sigma^* \times \mathbb{N}$; (ii) $g(x, k) \leq g'(k)$ for some computable function g'; and (iii) there exists computable function h and a fixed constant $c > 0$ such that for all input string, $f(x, k)$ can be computed by a deterministic Turing-machine in $O(h(k)n^c)$ steps. Our definition of reduction is consistent with [8, Definition 2.1] or [6, Definition 13.1], though some authors, e.g., [7, Definition 20.2.1] require $g(x, k) = g'(k)$ in (ii).

The classes FPT, W[P] and XP are closed under FPT-reductions. A problem Π is said to be *C-hard* where C is a parameterized complexity class, if all problems in C can be FPT-reduced to Π. Moreover, if $\Pi \in$ C, we say that Π is *C-complete*.

2 An algorithm for W[P]-membership

In this section we provide the algorithm used to prove W[P]-membership in Theorem 1.

Brute force algorithm. First, let us however look at a brute force algorithm for UNKNOTTING VIA DEFECT which does not give W[P]-membership. Spelling it out will be useful for explaining the next steps. We will exhibit this algorithm as a non-deterministic algorithm, which will be useful for comparison later on. In the algorithm, D is a diagram, and k is an integer, not necessarily positive. Also, given a diagram D and a feasible Reidemeister move m, then $D(m)$ denotes the diagram obtained from D after performing m.

BruteForce(D, k):

1. If $k < 0$, then output No. If $D = U$ is a diagram without crossings and $k \geq 0$, then output Yes. In all other cases continue to the next step.
2. (Non-deterministic step.) Enumerate all possible Reidemeister moves $m_1, \ldots, m_t$ in D up to isotopy. Make a "guess" which move m_i is the first to perform. Then, for such m_i, run BruteForce$(D(m_i), k - w(m_i))$.

Therefore, altogether, the algorithm outputs Yes, if there is a sequence of guesses in step 2 which eventually yields Yes in step 1.

It can be easily shown that the algorithm terminates because whenever step 2 is performed, either $k - w(m_i) < k$, or m_i is a II^- move, $k - w(m_i) = k$, but $D(m_i)$ has fewer crossings than D.

It can be also easily shown by induction that this algorithm provides a correct answer due to Lemma 2. Indeed, step 1 clearly provides a correct answer, and regarding step 2, if $D(m_i)$ can be untangled with defect at most $k - w(m_i)$, then Lemma 2 shows that D

can be untangled with defect at most k. Because, this way we try all possible sequences of Reidemeister moves, the algorithm outputs Yes if and only if D can be untangled with defect at most k.

On the other hand, this algorithm (unsurprisingly) does not provide W[P]-membership as there are at least $n/2$ Reidemeister moves, which is unbounded in k. Thus not all moves can be guessed non-deterministically.

Naive greedy algorithm. In order to fix the problem with the previous algorithm, we want to reduce the number of non-deteministic steps. It turns out that the problematic non-deterministic steps in the previous algorithm are those where k does not decrease. (Because the other steps appear at most k times.) Therefore, we want to avoid non-deterministic steps where we perform a II^- move. The naive way is to perform such steps greedily and hope that if D untangles with defect at most k, there is also such a "greedy" untangling (and therefore, we do not have to search through all possible sequences of Reidemeister moves). This is close to be true but it does not really work in this naive way. Anyway, we spell this naive algorithm explicitly, so that we can easily upgrade it in the next step, though it does not always provide the correct answer to Unknotting via defect.

NaiveGreedy(D, k):

1. If $k < 0$, then output No. If $D = U$ is a diagram without crossings and $k \geq 0$, then output Yes. In all other cases continue to the next step.
2. If there is a feasible Reidemeister II^- move m, run NaiveGreedy$(D(m), k)$ otherwise continue to the next step.
3. (Non-deterministic step.) If there is no feasible Reidemeister II^- move, enumerate all possible Reidemeister moves $m_1, \ldots, m_t$ in D up to isotopy. Make a "guess" which m_i is the first move to perform and run NaiveGreedy$(D(m_i), k - w(m_i))$.

The algorithm must terminate for the same reason why BruteForce terminates. It can be shown that this is a W[P]-algorithm. However, we do not do this here in detail as this is not our final algorithm. The key is that the step 3 is performed at most $(k+1)$-times.

If the algorithm outputs Yes, then this is a correct answer from the same reason as in the case of BruteForce. However, as we hinted earlier, outputting No need not be a correct answer to Unknotting via defect. Indeed, there are known examples of diagrams of an unknot when untangling requires performing a II^+ move, see for example [14]. With NaiveGreedy, we would presumably undo such II^+ move immediately in the next step, thus we would not find any untangling using the II^+ move. We have to upgrade the algorithm a little bit to avoid this problem (and a few other similar problems).

Special greedy algorithm. The way to fix the problem above will be to guess in advance a certain subset S of so-called special crossings. This set will be updated in each non-deterministic step described above so that the newly introduced crossings will become special as well. Then, in the greedy steps we will allow to perform only those II^- moves which avoid S. (This also means that a II^+ move cannot be undone by a II^- move avoiding S in the next step.) It will turn out that we may also need to perform II^- moves on S but there will not be too many of them, thus such moves can be considered in the non-deterministic steps.

For the description of the algorithm, we introduce the following notation. Let D be a diagram of a knot and S be a subset of the crossings of D; we will refer to crossings in S as *special* crossings. Then we say that a feasible Reidemeister II^- move m is *greedy* (with respect to S), if it avoids S; that is, the crossings removed by m do not belong to S. On the other hand, a feasible Reidemeister move m is *special* (with respect to S) if it is

- a I^- or a II^- move removing only crossings in S; or
- a III move such that all three crossings affected by m are special; or
- a I^+ or a II^+ move performed on the edges with all their endpoints in S.

Given a move m in D, special or greedy with respect to S, by $S(m)$ we denote the following set of crossings in $D(m)$.

- If m is a greedy II^- move or if m is a III move, then $S(m) = S$ (under the convention that the three crossings affected by m persist in $D(m)$).
- If m is a special II^- move or a I^- move, then $S(m)$ is obtained from S by removing the crossings removed by m.
- If m is a I^+ or a II^+ move, then $S(m)$ is obtained from S by adding the crossings introduced by m.

Now, we can describe the algorithm.

SpecialGreedy(D, k):

0. (Non-deterministic step.) Guess a set S of at most $3k$ crossings in D. Then run SpecialGreedy(D, S, k).

SpecialGreedy(D, S, k):

1. If $k < 0$, then output No. If $D = U$ is a diagram without crossings and $k \geq 0$, then output Yes. In all other cases continue to the next step.

2. If there is a feasible greedy Reidemeister II^- move m with respect to S, run SpecialGreedy$(D(m), S(m), k)$ otherwise continue to the next step.

3. (Non-deterministic step.) If there is no feasible greedy Reidemeister II^- move with respect to S, enumerate all possible special Reidemeister moves $m_1, \dots, m_t$ in D with respect to S up to isotopy. If there is no such move, that is, if $t = 0$, then output No. Otherwise, make a guess which m_i is performed first and run SpecialGreedy$(D(m_i), S(m_i), k - w(m_i))$.

The bulk of the proof of W[P]-membership in Theorem 1 will be to show that the algorithm SpecialGreedy(D, k) provides a correct answer to Unknotting via defect. Of course, if the algorithm outpus Yes, then this is the correct answer by similar arguments for the previous two algorithms. Indeed, Yes answer corresponds to a sequence of Reidemeister moves performed in step 2 or guessed in step 3 (no matter how we guessed S in step 0 and the role of S in the intermediate steps of the run of the algorithm is not important if we arrived at Yes). The defect of the untangling given by this sequence of moves is at most k by Lemma 2. On the other hand, we also need to show that if D untanlges with defect at most k, then we can guess some such untangling while running the algorithm. This is based on the following two auxiliary results.

For stating the results, we inductively define the notation $D(m_1, \dots, m_{k-1}, m_k) := D(m_1, \dots, m_{k-1})(m_k)$ provided that m_k is a feasible Reidemeister move in $D(m_1, \dots, m_{k-1})$. We also say that a sequence $(m_1, \dots, m_k)$ is a *feasible* sequence of Reidemeister moves for D if m_i is a feasible Redemeister move in $D(m_1, \dots, m_{i-1})$ for every $i \in [k]$. Similarly, we inductively define $S(m_1, \dots, m_{k-1}, m_k) := S(m_1, \dots, m_{k-1})(m_k)$, provided that the move m_k is special or greedy in $D(m_1, \dots, m_{k-1})$ with respect to $S(m_1, \dots, m_{k-1})$. Then $(m_1, \dots, m_k)$ is a *feasible* sequence of Reidemeister moves for the pair (D, S) if $S(m_1, \dots, m_k)$ is well defined this way; that is, each move m_i in the sequence is special or greedy (with respect to the intermediate set $S(m_1, \dots, m_{i-1})$ of special crossings).

▶ **Lemma 3** (Lemma 8 in full version). *Let D be a diagram of a knot and let $(m_1, \dots, m_\ell)$ be a feasible sequence of Reidemeister moves in D. Then there is a set of crossings S in D such that $(m_1, \dots, m_\ell)$ is feasible for (D, S).*

In addition, if this sequence induces an untangling with defect at most k, then $|S| \leq 3k$.

▶ **Theorem 4** (Theorem 9 in full version). *Let D be a diagram of a knot and S be a set of crossings in D. Let $(m_1, \dots, m_\ell)$ be a feasible sequence of Reidemeister moves for (D, S), inducing an untangling of D with defect k. Assume that there is a greedy move $\widetilde{m}$ in D with respect to S. Then there is a feasible sequence of Reidemeister moves for (D, S) starting with $\widetilde{m}$ and inducing an untangling of D with defect k.*

Roughly speaking, the idea of the proof of Lemma 3 is to add into S all crossings that will eventually become affected by some of the moves $m_1, \dots, m_\ell$ which are not II^- moves. These moves must be special. However, if we have a II^- move removing a crossing that we have already added to S, then this move also has to be special and we add the other crossing to S as well. This does not propagate further, and we can bound the number of special II^- moves which also provides a bound on S. Details are given in the full version. Proof of Theorem 4 requires more work and we sketch it in Section 3.

Correctness of the algorithm. Here we explain that the algorithm $\textsc{SpecialGreedy}(D, k)$ provides the answer $\textsc{Yes}$ whenever there is an untangling with defect at most k modulo Lemma 3 and Theorem 4. We also need to know that this is a W[P]-algorithm (mainly) by bounding the number of non-deterministic steps. This part is relatively straightforward and we refer to Section 4 of full version for details.

Given an input (D, k) such that D admits an untangling with defect k, by Lemma 3, there is a set of crossings S in D of size at most $3k$ such that there is a sequence $\mathcal{M}$ of Reidemeister moves feasible for (D, S) inducing an untangling of D with defect at most k. Thus it is sufficient to show that $\textsc{SpecialGreedy}(D, S, k)$ outputs $\textsc{Yes}$ if such a sequence for a given S exists. (In fact this is if and only if but we only need the if case.) We will show this by a double induction on k and $\mathrm{cr}(D)$, where $\mathrm{cr}(D)$ is the number of crossings in D. The outer induction is on k, the inner one is on $\mathrm{cr}(D)$. It would be sufficient to start our induction with the pair $(k, \mathrm{cr}(D)) = (0, 0)$; however, whenever $\mathrm{cr}(D) = 0$, then the algorithm outputs $\textsc{Yes}$ in step 1. Thus we may assume that $\mathrm{cr}(D) > 0$.

If (D, S) admits any greedy move, then we are in step 2. For every greedy move m there is a sequence of Reidemeister moves starting with m feasible for (D, S) inducing an untangling of D with defect at most k by Theorem 4. This move has weight 0. Thus by Lemma 2 there is a sequence of Reidemeister moves feasible for $(D(m), S(m))$ inducing an untangling of $D(m)$ with defect at most k. In addition $\mathrm{cr}(D(m)) < \mathrm{cr}(D)$, thus $\textsc{SpecialGreedy}(D(m), S(m), k)$ outputs $\textsc{Yes}$ by induction.

If (D, S) does not admit any greedy move, then we are in step 3. We in particular know that the first move in the sequence $\mathcal{M}$ must be special. We guess this move as we are in a non-deterministic step and denote it m_i (in consistence with the notation in step 3). Now, if we remove m_i from $\mathcal{M}$, this is a feasible sequence for $(D(m_i), S(m_i))$ inducing an untangling of $D(m_i)$ with defect at most $k - w(m_i)$ by Lemma 2. Thus $\textsc{SpecialGreedy}(D(m_i), S(m_i), k - w(m_i))$ outputs $\textsc{Yes}$ by induction, as we need.

3 Sketch for a proof of Theorem 4

Here we sketch a proof of Theorem 4. In general, the Reidemeister moves are not commutative: if (m, m') is a feasible sequence of Reidemeister moves in a diagram D, we cannot even expect that m' is feasible in D. A fortiori we cannot expect that (m', m) is feasible in D and that the resulting diagram would be the same when applying (m, m'). For the proof of Theorem 4, we need commutativity of the greedy moves under certain circumstances. In particular, we need to be able to swap the greedy moves with the neighbors so that we can shift them both forward and backward through the sequence. Hint that this might be sometimes possible is the following: if m and m' can be performed inside disjoint balls, then they commute. In this section, we only state the lemmas on rearranging Reidemeister moves. The proofs of these lemmas (and their corollaries) are in the full version. Then we deduce Theorem 4 from the lemmas up to minor details explained in the full version.

First, we extend our earlier notation: Let D be a diagram of a knot, S be a set of crossings and $(m_1, \dots, m_k)$ be a sequence of Reidemeister moves feasible for a pair (D, S). Then by $(D, S)(m_1, \dots, m_k)$ we denote the pair (D', S') obtained by applying moves $(m_1, \dots, m_k)$ to (D, S). In our earlier notation, this means that $D' = D(m_1, \dots, m_k)$ and $S' = S(m_1, \dots, m_k)$. We also denote a II^- move removing crossings x and y by $\{x, y\}_{\mathrm{II}^-}$.

The first lemma allows to perform a greedy move m earlier in a sequence (m', m).

▶ **Lemma 5** (Lemma 10 in full version). *Let D be a diagram of a knot and S be a set of crossings in D. Let $m = \{x, y\}_{\mathrm{II}^-}$ be a II^- move in D, greedy with respect to S. Let m' be a feasible move for the pair (D, S) which avoids $\{x, y\}$. Then there is a move $\widehat{m}'$ feasible in $D(m)$ of same type as m' and the following conditions hold:*

(i) *m is greedy in $D(m')$ with respect to $S(m')$, in particular m is feasible in $(D, S)(m')$*

(ii) *$\widehat{m}'$ is feasible in $(D, S)(m)$; and*

(iii) *$(D, S)(m, \widehat{m}') = (D, S)(m', m)$.*

Lemma 5 is proved by a careful local analysis depending on the type of m'. By a suitable induction, Lemma 5 implies the following corollary for longer sequences of moves.

▶ **Corollary 6** (Corollary 11 in full version). *Let D be a diagram, S be a set of crossings in D, $\ell \geq 2$, and $(m_1, \dots, m_\ell)$ be a feasible sequence of Reidemeister moves for the pair (D, S). Assume that m_ℓ is a II^- move which is also feasible and greedy in D with respect to S. Then there is a sequence of Reidemester moves $(m_\ell, \widehat{m}_1, \dots, \widehat{m}_{\ell-1})$ feasible for (D, S) such that*

(i) *$(D, S)(m_1, \dots, m_\ell) = (D, S)(m_\ell, \widehat{m}_1, \dots, \widehat{m}_{\ell-1})$; and*

(ii) *$w(m_i) = w(\widehat{m}_i)$ for $i \in [\ell - 1]$.*

We will also need a variant of Corollary 6 which allows to postpone a greedy move. The following corollary allows us to do that. (this follows from a lemma analogous to Lemma 5 which we skip here).

▶ **Corollary 7** (Corollary 13 in full version). *Let D be a diagram, S be a set of crossings in D, $\ell \geq 2$ and $(m_1, \dots, m_\ell)$ be a feasible sequence of moves for (D, S). Assume that $m_1 = \{x_1, y_1\}_{\mathrm{II}^-}$ and $m_\ell = \{x_\ell, y_\ell\}_{\mathrm{II}^-}$ are II^- moves where $x_1, y_1, x_\ell, y_\ell \notin S$ (in particular m_1 is greedy in D with respect to S). Finally, assume also that $\{x_1, x_\ell\}_{\mathrm{II}^-}$ is a feasible Reidemeister move for (D, S) (again, it must be greedy). Then, there is a feasible sequence $(\widehat{m}_2, \widehat{m}_3, \dots, \widehat{m}_{\ell-1}, m_1)$ of moves for (D, S) such that*

(i) *$(D, S)(m_1, \dots, m_{\ell-1}) = (D, S)(\widehat{m}_2, \widehat{m}_3, \dots, \widehat{m}_{\ell-1}, m_1)$; and*

(ii) *$w(m_i) = w(\widehat{m}_i)$ for $i \in \{2, \dots, \ell - 1\}$.*

Last, but not the least, both the corollaries above essentially only allow swapping the moves. In the corollaries above, m_i is essentially the same move as $\hat{m}_i$ up to a combinatorial description (not discussed in this extended abstract) which is a reason why we use a different notation. Such corollaries cannot be sufficient for a proof of Theorem 4 in the case that $\tilde{m}$ (from the statement of the theorem) is not in the sequence $(m_1, \dots, m_\ell)$. We also need to be able to replace some greedy moves with different ones which is the content of the following lemma.

▶ **Lemma 8** (Lemma 14 in full version). *Let D be a diagram of a knot and let $m = \{x, y\}_{\text{II}^-}$, $\widetilde{m} = \{x, z\}_{\text{II}^-}$ be two feasible Reidemeister moves in D where $y \neq z$. Assume also that $m' = \{w, z\}_{\text{II}^-}$ is feasible in $D(m)$ with x, y, z, w mutually distinct. Then*
- **(i)** *$\widetilde{m}' = \{w, y\}_{\text{II}^-}$ is feasible in $D(\widetilde{m})$; and*
- **(ii)** *$D(m, m') = D(\widetilde{m}, \widetilde{m}')$.*

Lemma 8 is again proved by local analysis in the full version.

Proof of Theorem 4 (sketch). First, let us assume that $\widetilde{m} = m_j$ for some $j \in [\ell]$. If $j = 1$, there is nothing to prove, thus we may assume $j \geq 2$. Then, by using Corollary 6 on the sequence $(m_1, \ldots, m_j)$, we get a sequence of moves $(m_j, \widehat{m}_1, \ldots, \widehat{m}_{j-1})$ feasible for (D, S). By item (i) of Corollary 6, the sequence $(m_j, \widehat{m}_1, \ldots, \widehat{m}_{j-1}, m_{j+1}, \ldots, m_\ell)$ is also feasible for (D, S) and induces an untangling of D. By item (ii) of Corollary 6 and Lemma 2, the defect of this sequence equals k, and that is what we need.

Thus it remains to consider the case where $\widetilde{m} \neq m_j$ for every $j \in [\ell]$. Because $\widetilde{m}$ is greedy, it is a II^- move; say $\widetilde{m} = \{x, z\}_{\text{II}^-}$. Because the final diagram $D(m_1, \ldots, m_\ell)$ has no crossings, the crossings x and z have to be removed by some moves in the sequence $(m_1, \ldots, m_\ell)$. Say that a move m_i removes x and m_j removes z. Since $\widetilde{m}$ is greedy with respect to S, we get $x, z \notin S$. This also implies that $x \notin S(m_1, \ldots, m_{i-1})$ (see the full version for details). Thus m_i has to be greedy move in $D(m_1, \ldots, m_{i-1})$ with respect to $S(m_1, \ldots, m_{i-1})$. Similarly, m_j is greedy in $D(m_1, \ldots, m_{j-1})$ with respect to $S(m_1, \ldots, m_{j-1})$. Because we assume that $\widetilde{m}$ is not in the sequence $(m_1, \ldots, m_\ell)$, we get $i \neq j$; without loss of generality $i < j$. Let y and w be such that $m_i = \{x, y\}_{\text{II}^-}$ and $m_j = \{w, z\}_{\text{II}^-}$. As these moves are greedy, we get $y, w \notin S$ (we again refer to the full version for more detail).

Let $D' := D(m_1, \ldots, m_{i-1})$ and $S' := S(m_1, \ldots, m_{i-1})$. By Corollary 7, applied to D', S' and the sequence $(m_i, \ldots, m_j)$ feasible for (D', S'), we get another sequence $(\widehat{m}_{i+1}, \ldots, \widehat{m}_{j-1}, m_i)$ feasible for (D', S'). (For verifying the assumptions of the corollary note that $\widetilde{m}$ is feasible for (D', S') as all the moves $m_1, \ldots, m_{i-1}$ are special or greedy, thus they cannot remove x or z nor affect the arcs connecting them.) Then we get that $(\widehat{m}_{i+1}, \ldots, \widehat{m}_{j-1}, m_i, m_j, \ldots, m_\ell)$ is also feasible for (D', S') by item (i) of Corollary 7.

Next, let $D'' := D'(\widehat{m}_{i+1}, \ldots, \widehat{m}_{j-1})$ and $S'' := S'(\widehat{m}_{i+1}, \ldots, \widehat{m}_{j-1})$. By a similar argument as above, we get that $\widetilde{m}$ is feasible in (D'', S''). (Note that the moves m_i, m_j removing x and z have not been performed yet in order to get D''.) By Lemma 8 used in D'', we get that $\widetilde{m}' := \{w, y\}_{\text{II}^-}$ is feasible in $(D'', S'')(\widetilde{m})$ and $(D'', S'')(m_i, m_j) = (D'', S'')(\widetilde{m}, \widetilde{m}')$. Altogether, by expanding D'' and D'

$$(m_1, \ldots, m_{i-1}, \widehat{m}_{i+1}, \ldots, \widehat{m}_{j-1}, \widetilde{m}, \widetilde{m}', m_{j+1}, \ldots, m_\ell) \tag{1}$$

is an untangling of D, feasible for (D, S). We also get that the defect of this untangling is equal to k by Lemma 2 and item (ii) of Corollary 7, when we used it. Note that all the moves m_i, m_j, $\widetilde{m}$ and $\widetilde{m}'$ are II^- moves, thus they do not contribute to the weight.

The sequence (1) is not the desired sequence yet, because it does not start with $\widetilde{m}$. However, it contains $\widetilde{m}$, thus we can further modify this sequence to the desired sequence starting with $\widetilde{m}$ as in the first paragraph of this proof. ◀

4 W[P]-hardness

In this section we sketch a proof of W[P]-hardness part of Theorem 1. This is done by an FPT-reduction from MINIMUM AXIOM SET defined below.

4.1 Minimum axiom set

It is well known that the Minimum axiom set problem is W[P]-hard; see [8, Exercise 3.20] or [7, Lemma 25.1.3] (however, let us recall that our definition of FPT-reduction is consistent with [8]).

▶ **Problem** (Minimum axiom set).

Input	A finite set S, and a finite set $\mathcal{R}$ which consists of pairs of form (T, t) where $T \subseteq S$ and $t \in S$.
Parameter	k.
Question	Does there exist a subset $S_0 \subseteq S$ of size k such that if we define inductively S_i to be the union of S_{i-1} and all $t \in S$ such that there is $T \subseteq S_{i-1}$ with $(T, t) \in \mathcal{R}$, then $\bigcup_{i=1}^{\infty} S_i = S$?

The problem above deserves a brief explanation. The elements of S are called *sentences* and the elements of $\mathcal{R}$ are relations. A relation (T, t) with $T = \{t_1, \dots, t_m\}$ should be understood as an implication

$$t_1 \wedge t_2 \wedge \cdots \wedge t_m \Rightarrow t.$$

Given a set $S_0 \subseteq S$, let us define the *consequences* of S_0 as $c(S_0) := \bigcup_{i=1}^{\infty} S_i$ where S_i is defined as in the statement of the problem. Intuitively, $c(S_0)$ consists of all sentences that can be deduced from S_0 via the *relations* (implications) in $\mathcal{R}$. As we work with finite sets, $c(S_0) = S_i$ for some high enough i. A set A is a set of *axioms* if $c(A) = S$. Therefore, the goal of the minimum axiom set problem is to determine whether there is a set of axioms of size k. Note that the axiom sets are upward-closed: If A is an axiom set and $A \subseteq A' \subseteq S$, then A' is an axiom set as well.

The following boosting lemma is very useful in our reduction.

▶ **Lemma 9** (Boosting lemma; Lemma 15 in full version). *Let $(S, \mathcal{R})$ be an input of the minimum axiom set problem (ignoring the parameter for now). Let $\mu \colon S \to \mathbb{Z}$ be a non-negative function. Given $U \subseteq S$, let $\mu(U) = \sum_{s \in U} \mu(s)$. Assume that $\mu(U) \geq 1$ for all U such that $S \setminus U$ is not an axiom set. (Equivalently, U meets every axiom set.) Then $\mu(S) \geq k^*$ where k^* is the size of a minimum axiom set.*

Proof. Let $Z := \{s \in S \colon \mu(s) = 0\}$ be the zero set of μ. Then $\mu(Z) = 0$, thus $S \setminus Z$ is a set of axioms by the assumptions. This gives $|S \setminus Z| \geq k^*$ and, in addition, $\mu(S) = \mu(S \setminus Z) \geq |S \setminus Z| \geq k^*$ because $\mu(s) \geq 1$ for every $s \in S \setminus Z$. ◀

4.2 Construction of the reduction

Our aim is to show that there is an FPT-reduction from the Minimum axiom set to Unknotting via defect. It is not hard to see (see the full version) that we can assume that the input $(S, \mathcal{R}, k)$ is *preprocessed*; i.e., (i) $\mathcal{R}$ does not contain relations of the form $(\emptyset, t)$; (ii) for every $s \in S$ there is a relation of a form (T, s) in $\mathcal{R}$; and (iii) $t \notin T$ for every $(T, t) \in \mathcal{R}$.

Doubling the instance. Now let $(S, \mathcal{R}, k)$ be a preprocessed instance of the Minimum axiom set. We will need the following doubled instance: Let $\widehat{S} := \{\widehat{s} \colon s \in S\}$ be an auxiliary copy of S. Given $T = \{t_1, \dots, t_m\} \subseteq S$, let $\widehat{T} := \{\widehat{t}_1, \dots, \widehat{t}_m\}$. Then we define $\widehat{\mathcal{R}} := \{(\widehat{T}, \widehat{t}) : (T, t) \in \mathcal{R}\}$. Then $(S \cup \widehat{S}, \mathcal{R} \cup \widehat{\mathcal{R}}, 2k)$ is a *double* of the instance $(S, \mathcal{R}, k)$. The proof of the following observation is straightforward and it is given in the full version; see Observation 16 there.

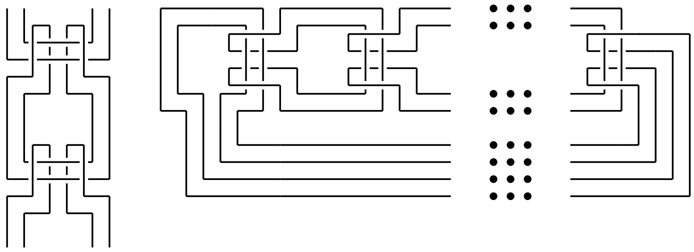

Figure 3 A Brunnian link.

▶ **Observation 10.** *The pair* $(S, \mathcal{R})$ *admits an axiom set of size* k *if and only if its double* $(S \cup \widehat{S}, \mathcal{R} \cup \widehat{\mathcal{R}})$ *admits an axiom set of size* $2k$*.*

Brunnians. A *Brunnian link* is a nontrivial link that becomes trivial whenever one of the link components is removed. We will use the following well known construction of a Brunnian link with $\ell \geq 2$ components. We take an untangled unknot and we interlace it with two "neighboring" unknots as in Figure 3, left. We repeat this ℓ-times and we get a Brunnian link with ℓ components as in Figure 3, right.

Gadgets. From now on let $(S, \mathcal{R}, k)$ be a preprocessed instance of the MINIMUM AXIOM SET and $(S \cup \widehat{S}, \mathcal{R} \cup \widehat{\mathcal{R}}, 2k)$ be its double. Our aim is to build a diagram $D(S, \mathcal{R})$ such that $D(S, \mathcal{R})$ untangles with defect $2k$ if and only if $(S, \mathcal{R})$ has an axiom set of size k. We will build $D(S, \mathcal{R})$ using several gadgets. Formally speaking, gadgets will be maps of a form $G\colon I_1 \sqcup \cdots \sqcup I_h \to \mathbb{R}^2$ where $I_1 \sqcup \cdots \sqcup I_h$ stands for a disjoint union of intervals $[0, 1]$. We will work with them in the same way as with diagrams. In particular, we assume the same transversality assumptions on crossings as for diagrams and we mark the underpasses and overpasses. We also extend the notion of arc to this setting: It is a set $G(A)$, where A is a closed subinterval of one of the intervals in $I_1 \sqcup \cdots \sqcup I_h$.

Sentence gadget. For each sentence $s \in S$ we define a sentence gadget $G(s)$ as follows. (We will also create an analogous gadget $G(\widehat{s})$ for $\widehat{s} \in \widehat{S}$ which we specify after describing $G(s)$.) We consider all relations $R \in \mathcal{R}$ of the form (T, s). Let $\ell = \ell(s)$ be the number of such relations and we order these relations as $R_1(s), \ldots, R_\ell(s)$, or simply as $R_1, \ldots, R_\ell$ if s is clear from the context (which is the case now). Note that $\ell \geq 1$ due to preprocessing.

Now we take our Brunnian link with $\ell+1$ components, which we denote by $C_0(s), \ldots, C_\ell(s)$, in the order along the Brunnians. We disconnect each component of this Brunnian link in an arc touching the outer face and we double each such disconnected component. See Figure 4, left. For the further description of the construction, we assume that our construction is rotated exactly as in the figure. (If $\ell \neq 2$, then we just modify the length of $C_0(s)$ and insert more or fewer components $C_i(s)$ in the same way as $C_1(s)$ and $C_2(s)$ are inserted for $\ell = 2$.) Now we essentially have a collection of interlacing arcs where each former component of Brunnians yields a pair of parallel arcs with four loose ends.

We merge each pair of arcs, coming from $C_i(s)$ into a single arc $\gamma_i(s)$ as follows; see Figure 4, right. For the pair coming from $C_i(s)$ for $i \neq 0$, we connect the bottom loose ends by a straight segment up to isotopy. In Figure 4, right, we have isotoped the figure a bit which will be useful in further steps of the construction, and we call this subarc the *head* of $\gamma_i(s)$. For the pair coming from $C_0(s)$, we first cross the arcs next to the top loose ends as well as next to the bottom loose ends as in the figure. Then we connect the bottom loose

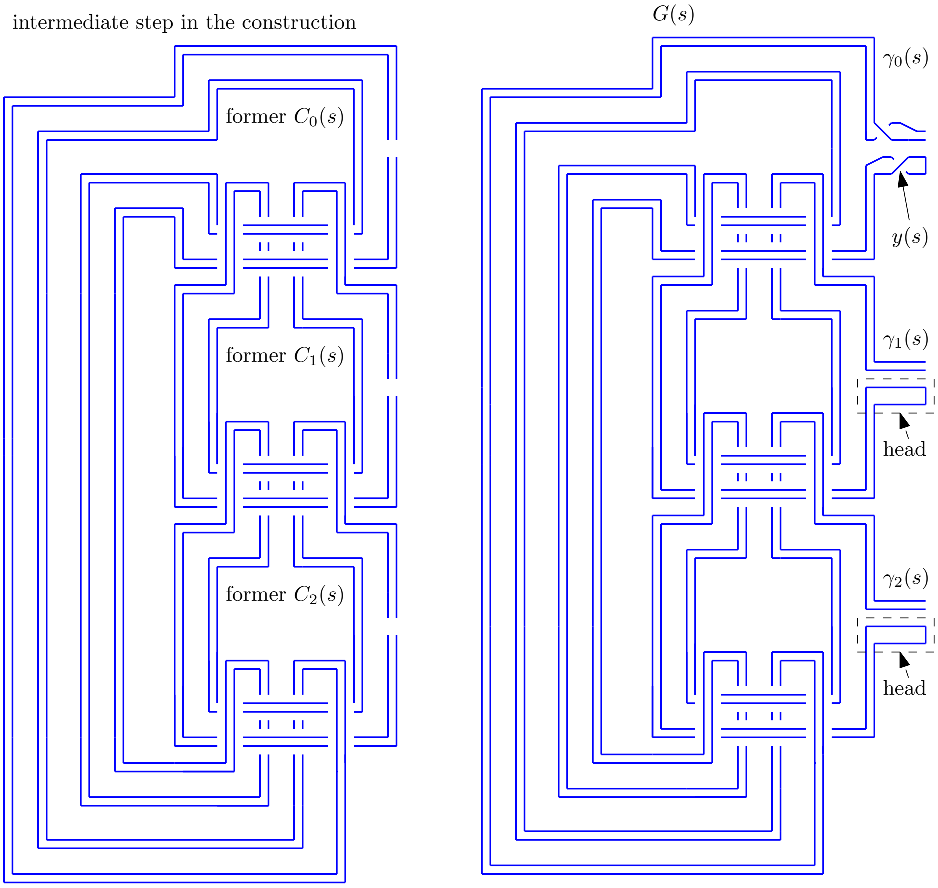

Figure 4 The sentence gadget with $\ell = 2$.

ends. Note that if we remove $\gamma_1(s), \dots, \gamma_l(s)$ from the figure, then the crossings on $\gamma_0(s)$ can be removed by a II^- move. This finishes the construction of $G(s)$. The gadget $G(\widehat{s})$ is a mirror image of $G(s)$ along the vertical line (y-axis).

Merging gadget. We will also need a merging gadget depicted on Figure 5. We order all sentences as $s_1, \dots, s_m$. The merging gadget consists of subgadgets $M(s_1), M(\widehat{s}_1), M(s_2), \dots,$ $M(s_m), M(\widehat{s}_m)$ separated by the dotted lines in the figure. Each subgadget has several loose ends. Two or four of them serve for connecting it to other subgadgets. The remaining ones come in pairs and the number of pairs equals to $\ell(s) + 1$. (Recall that this is the number of components of the sentence gadget $G(s)$.) These pairs of loose ends will be eventually connected to the loose ends of $\gamma_0(s), \dots, \gamma_{\ell(s)}(s)$ in the top-down order.

Interconnecting the gadgets. Now, we describe how to interconnect the gadgets.

We place the sentence gadgets $G(s_1), \dots, G(s_m)$ to the left of the merging gadget M in a top-down ordering. Similarly, we place $G(\widehat{s}_1), \dots, G(\widehat{s}_m)$ to the right, again in a top-down ordering. First, for any $\bar{s} \in S \cup \widehat{S}$ and $i \in \{0, 2, 3, \dots, \ell(\bar{s})\}$ (that is, $i \neq 1$), we connect $\gamma_i(\bar{s})$ with the $(i+1)$th pair of loose ends of $M(\bar{s})$ by a pair of parallel arcs as directly as possible without introducing new crossings; see Figure 7 (left).

Now we want to connect $\gamma_1(\bar{s})$ to the second pair of loose ends of $M(\bar{s})$. For simplicity, we describe this in the case $\bar{s} = s \in S$. The case $\bar{s} \in \widehat{S}$ is mirror symmetric. We pull a pair of parallel arcs from $\gamma_1(s)$ with the aim to reach $M(s)$ while obeying the following rules:

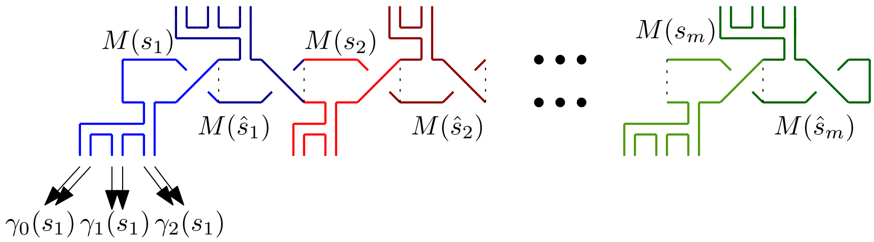

Figure 5 The merging gadget. The gadget is rotated. See also Figure 7 for correct orientation of the gadget when connecting it to other gadgets.

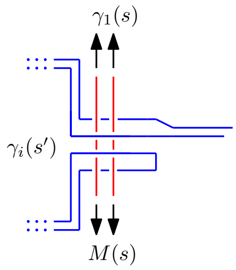

Figure 6 Interlacing the parallel arcs pulled out of $\gamma_1(s)$ with $\gamma_j(s')$.

- **(R1)** We are not allowed to cross the merging gadget or the sentence gadgets (except the case described in the third rule below). We keep the newly introduced arcs on the left side from the merging gadget.
- **(R2)** We are allowed to cross other pairs of parallel arcs introduced previously (when connecting $\gamma_i(s')$ to $M(s')$ for some i and s'). However, if we cross such a pair, we require that all four newly introduced crossings are resolved simultaneously (for example the new pair of parallel arcs is always above the older one). We even allow a self crossing of the newly introduced parallel arcs (but we again require that four newly introduced crossings are resolved simultaneously).
- **(R3)** For every relation $R = (T, s')$ where $s \in T$, let $R = R_i(s')$. We interlace the newly introduced parallel arcs with $\gamma_i(s')$ as in Figure 6.
- **(R4)** The total number of crossing is of polynomial size in the size of our instance $(S, \mathcal{R})$.

As we have quite some freedom how to perform the construction obeying the rules above, the resulting construction is not unique. An example how to get this construction systematically is sketched on Figure 7(right): The newly introduced pair of arcs is pulled little bit to the right, then we continue down towards the level of $G(s_m)$, then up towards the level of $G(s_1)$, and then back down to the original position. On this way we make a detour towards $\gamma_j(s')$ whenever we need to apply the rule (R3). Finally, we connect the parallel arcs to $M(s)$ without any further detour. This finishes the construction of $D(S, \mathcal{R})$.

A small set of axioms implies small defect. Now we very briefly sketch the easier implication that if $(S, \mathcal{R})$ admits an axiom set of size k, then $D(S, \mathcal{R})$ can be untangled with defect $2k$. For every sentence s in some fixed minimum axiom set, we unscrew the loop next to $y(s)$ (compare with Figure 4, right) and we also unscrew the loops next to $y(\widehat{s})$. This is altogether $2k$ I^- moves. If we show that the remaining crossings can be removed by II^- moves only, then we are done via Lemma 2. It is not hard to see that for s in the minimum axiom set, the initial unscrewing allows to simplify the gadget $G(s)$ essentially to $M(s)$ via II^- moves. This

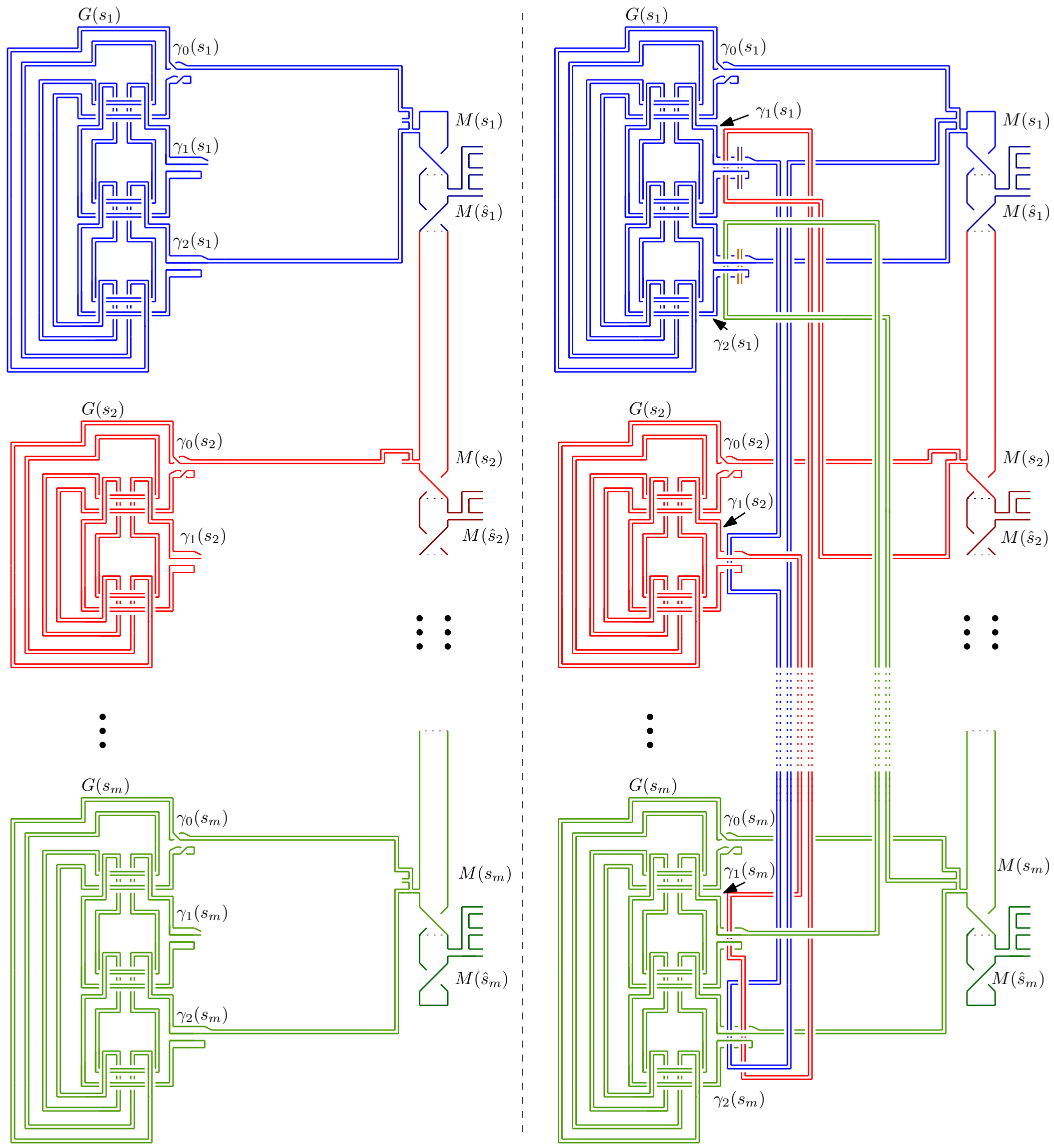

Figure 7 Left: The first step of interconnecting the gadgets – connecting $M(s)$ to the arcs $\gamma_i(s)$ for $i \neq 1$. The mirror symmetric part for $\widehat{s}$ is not displayed. Right: The second step of interconnecting the gadgets – connecting $M(s)$ to the arc $\gamma_1(s)$.

in particular releases at least one of the heads (see Figure 4, right) of the gadgets implied by the sentences in the minimum axiom set. Such a gadget can be then simplified again by II^- moves. By repeating this procedure we simplify all gadgets $G(s_1), \ldots, G(s_m)$ and analogously $G(\widehat{s}_1), \ldots, G(\widehat{s}_m)$. Then we remove remaining crossings by m more II^- moves.

Big minimum axiom sets implies big defect. It turns out that the core of the second implication in the reduction is to show the following: If the minimum axiom set for $(S, \mathcal{R})$ has size at least k, then any untangling of $D(S, \mathcal{R})$ has defect at least $2k$. For a proof of this claim we use the approach from [23] that allows to show by a local analysis that for some diagrams the defect of any untangling is strictly positive. On the other hand, this local analysis does not show much more than that the defect is at least 1 but we need $2k$. In order to circumvent this problem we use the boosting lemma (Lemma 9) and we verify, roughly speaking, that the defect is at least 1 also for certain subdiagrams corresponding to sets of sentences U such that $S \setminus U$ is not an axiom set (as in the assumptions of Lemma 9). Then Lemma 9 implies that the defect is as high as we need.

References

1 N. Alon, G. Z. Gutin, E. J. Kim, S. Szeider, and A. Yeo. Solving MAX-r-sat above a tight lower bound. *Algorithmica*, 61(3):638–655, 2011.

2 U. Bauer, A. Rathod, and J. Spreer. Parametrized Complexity of Expansion Height. In Michael A. Bender, Ola Svensson, and Grzegorz Herman, editors, *27th Annual European Symposium on Algorithms (ESA 2019)*, volume 144 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 13:1–13:15, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

3 B. A. Burton, T. Lewiner, J. Paixão, and J. Spreer. Parameterized complexity of discrete Morse theory. *ACM Trans. Math. Software*, 42(1):Art. 6, 24, 2016.

4 R. Crowston, M. R. Fellows, G. Z. Gutin, M. Jones, E. J. Kim, F. Rosamond, I. Z. Ruzsa, S. Thomassé, and A. Yeo. Satisfying more than half of a system of linear equations over GF(2): A multivariate approach. *J. Comput. Syst. Sci.*, 80(4):687–696, 2014.

5 R. Crowston, M. Jones, G. Muciaccia, G. Philip, A. Rai, and S. Saurabh. Polynomial kernels for lambda-extendible properties parameterized above the Poljak-Turzik bound. In A. Seth and N. K. Vishnoi, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2013, December 12-14, 2013, Guwahati, India*, volume 24 of *LIPIcs*, pages 43–54. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2013.

6 M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, Ma. Pilipczuk, Mi. Pilipczuk, and S. Saurabh. *Parameterized algorithms*. Springer, Cham, 2015.

7 R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013.

8 J. Flum and M. Grohe. *Parameterized complexity theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2006.

9 G. Z. Gutin, L. van Iersel, M. Mnich, and A. Yeo. Every ternary permutation constraint satisfaction problem parameterized above average has a kernel with a quadratic number of variables. *J. Comput. Syst. Sci.*, 78(1):151–163, 2012.

10 G. Z. Gutin, E. J. Kim, M. Lampis, and V. Mitsou. Vertex cover problem parameterized above and below tight bounds. *Theory Comput. Syst.*, 48(2):402–410, 2011.

11 G. Z. Gutin and V. Patel. Parameterized traveling salesman problem: Beating the average. *SIAM J. Discret. Math.*, 30(1):220–238, 2016.

12 W. Haken. Theorie der Normalflächen. *Acta Math.*, 105:245–375, 1961.

13 J. Hass, J. C. Lagarias, and N. Pippenger. The computational complexity of knot and link problems. *J. ACM*, 46(2):185–211, 1999.

14 L. H. Kauffman and S. Lambropoulou. Hard unknots and collapsing tangles. In *Introductory lectures on knot theory*, volume 46 of *Ser. Knots Everything*, pages 187–247. World Sci. Publ., Hackensack, NJ, 2012.

15 D. Koenig and A. Tsvietkova. NP–hard problems naturally arising in knot theory. *Trans. Amer. Math. Soc. Ser. B*, 8:420–441, 2021.

16 G. Kuperberg. Knottedness is in **NP**, modulo GRH. *Adv. Math.*, 256:493–506, 2014.

17 M. Lackenby. A polynomial upper bound on Reidemeister moves. *Ann. of Math. (2)*, 182(2):491–564, 2015.

18 M. Lackenby. Elementary knot theory. In *Lectures on geometry*, Clay Lect. Notes, pages 29–64. Oxford Univ. Press, Oxford, 2017.

19 M. Lackenby. The efficient certification of knottedness and Thurston norm. *Adv. Math.*, 387, 2021.

20 M. Lackenby. Marc Lackenby announces a new unknot recognition algorithm that runs in quasi-polynomial time, 2021. URL: `https://www.maths.ox.ac.uk/node/38304`.

21 M. Mahajan and V. Raman. Parameterizing above guaranteed values: MaxSat and MaxCut. *J. Algorithms*, 31(2):335–354, 1999.

22 M. Mahajan, V. Raman, and S. Sikdar. Parameterizing above or below guaranteed values. *J. Comput. Syst. Sci.*, 75(2):137–153, 2009.

23 A. de Mesmay, Y. Rieck, E. Sedgwick, and M. Tancer. The unbearable hardness of unknotting. *Adv. Math.*, 381:107648, 36, 2021.