**Proceedings Paper:**
Thomassé, S, Trotignon, N and Vušković, K (2014) A polynomial turing-Kernel for weighted independent set in bull-free graphs. In: Kratsch, D and Todinca, I, (eds.) Graph-Theoretic Concepts in Computer Science, Lecture Notes in Computer Science. Springer Verlag , 408 - 419. ISBN 9783319123394

https://doi.org/10.1007/978-3-319-12340-0_34

# A polynomial Turing-kernel for weighted independent set in bull-free graphs

Stéphan Thomassé ⋆, Nicolas Trotignon⋆, and Kristina Vušković⋆⋆

CNRS, LIP, ENS de Lyon, INRIA, Université de Lyon,
School of Computing, University of Leeds, and Faculty of Computer Science (RAF),
Union University, Belgrade, Serbia

**Abstract.** The maximum stable set problem is NP-hard, even when restricted to triangle-free graphs. In particular, one cannot expect a polynomial time algorithm deciding if a bull-free graph has a stable set of size $k$, when $k$ is part of the instance. Our main result in this paper is to show the existence of an FPT algorithm when we parameterize the problem by the solution size $k$. A polynomial kernel is unlikely to exist for this problem. We show however that our problem has a polynomial size Turing-kernel. More precisely, the hard cases are instances of size $O(k^5)$. All our results rely on a decomposition theorem of bull-free graphs due to Chudnovsky which is modified here, allowing us to provide extreme decompositions, adapted to our computational purpose.

## 1 Introduction

In this paper all graphs are simple and finite. We say that a graph $G$ *contains* a graph $F$, if $F$ is isomorphic to an induced subgraph of $G$. We say that $G$ is $F$*-free* if $G$ does not contain $F$. For a class of graphs $\mathcal{F}$, the graph $G$ is $\mathcal{F}$*-free* if $G$ is $F$-free for every $F \in \mathcal{F}$. The *bull* is a graph with vertex set $\{x_1, x_2, x_3, y, z\}$ and edge set $\{x_1x_2, x_1x_3, x_2x_3, x_1y, x_2z\}$.

Chudnovsky in a series of papers [4–7] gives a complete structural characterisation of bull-free graphs (more precisely, bull-free trigraphs, where a trigraph is a graph with some adjacencies left undecided). Roughly speaking, this theorem asserts that every bull-free trigraph is either in a well-understood *basic* class, or admits a *decomposition* allowing to break the trigraph into smaller *blocks*. In Section 2, we extract what we need for the present work, from the very complex theorem of Chudnovsky. In Section 3, we prove that bull-free trigraphs admit *extreme* decompositions, that are decompositions such that one of the blocks is basic. In Section 4, we give polynomial time algorithms to actually compute the extreme decompositions whose existence is proved in the previous section. In

⋆ Partially supported by *Agence Nationale de la Recherche* under reference ANR 10 JCJC 0204 01 and ANR 14 BLAN STINT.

⋆⋆ Partially supported by EPSRC grant EP/K016423/1 and Serbian Ministry of Education and Science projects 174033 and III44006.

Section 5, we introduce the notion of weigthed trigraphs. In Section 6, we give an FPT-algorithm for the maximum stable set problem restricted to bull-free graphs. Let us explain this. The notion of fixed-parameter tractability (FPT) is a relaxation of classical polynomial time solvability. A parameterized problem is said to be *fixed-parameter tractable* if it can be solved in time $f(k)P(n)$ on instances of input size $n$, where $f$ is a computable function (so $f(k)$ depends only on the value of parameter $k$), and $P$ is a polynomial function independent of $k$. We give an FPT-algorithm for the maximum stable set problem restricted to bull-free graphs. This generalizes the result of Dabrowski, Lozin, Müller and Rautenbach [8] who give an FPT-algorithm for the same parameterized problem for $\{$bull, $\overline{P_5}\}$-free graphs, where $P_5$ is a path on 5 vertices and $\overline{P_5}$ is its complement. Recently, Lokshtanov, Vatshelle and Villanger [17] proved that maximum independent set in $P_5$-free graphs can be computed in polynomial time. Also, forbidding a bull and odd holes leads to polynomial algorithm for Maximum Weight Independent Sets, see Brandstäd and Mosca [3]. In a weighted graph the *weight* of a set is the sum of the weights of its elements, and with $\alpha_w(G)$ we denote the weight of a maximum weighted independent set of a graph $G$ with weight function $w$. We state below the problem that we solve more formally.

PARAMETERIZED WEIGHTED INDEPENDENT SET
*Instance:* A weighted graph $G$ with weight function $w : V(G) \longrightarrow \mathbb{N}$ and a positive integer $k$.
*Parameter: k*
*Problem:* Decide whether $G$ has an independent set of weight at least $k$. If no such set exists, find an independent set of weight $\alpha_w(G)$.

Observe that the problem above is $W[1]$-hard for general graphs [9].

In Section 7, we show that while a polynomial kernel is unlikely to exist since the problem is OR-compositional, we can prove nonetheless that the hardness of the problem can be reduced to polynomial size instances. Precisely we show that if it takes time $f(k)$ to decide if a stable set of size $k$ exists for bull-free graphs of size $O(k^5)$, then one can solve the problem on instances of size $n$ in time $f(k)P(n)$ for some polynomial $P$ in $n$. The fact that hard cases can be reduced to size polynomial in $k$ is not captured by the existence of a polynomial kernel, but by what is called a Turing-kernel (see Section 7 or Lokshtanov [16] for a definition of Turing-kernels). Even the existence of a Poly$(n)$ set of kernels of size Poly$(k)$ seems unclear for this problem. To our knowledge, stability in bull-free graphs is the first example of a problem admitting a polynomial Turing-kernel which is not known to have an independent set of polynomial kernels. An interesting question is to investigate which classical problems without polynomial kernels do have a polynomial Turing-kernel. This question is investigated by Hermelin et al. [15].

## 2 Decomposition of bull-free graphs

In the series of papers [4–7] Chudnovsky gives a complete structural characterisation of bull-free graphs which we first describe informally. Her construction of all bull-free graphs starts from three explicitly constructed classes of basic bull-free graphs: $\mathcal{T}_0, \mathcal{T}_1$ and $\mathcal{T}_2$. Class $\mathcal{T}_0$ consists of graphs whose size is bounded by some constant, the graphs in $\mathcal{T}_1$ are built from a triangle-free graph $F$ and a collection of disjoint cliques with prescribed attachments in $F$ (so triangle-free graphs are in this class), and $\mathcal{T}_2$ generalizes graphs $G$ that have a pair $uv$ of vertices, so that $uv$ is dominating both in $G$ and $\bar{G}$. Furthermore, each graph $G$ in $\mathcal{T}_1 \cup \mathcal{T}_2$ comes with a list $\mathcal{L}_G$ of "expandable edges". Chudnovsky shows that every bull-free graph that is not obtained by substitution from smaller ones, can be constructed from a basic bull-free graph by expanding the edges in $\mathcal{L}_G$ (where edge expansion is an operation corresponding to reversing the homogeneous pair decomposition). All these terms will be defined later in this section. To prove and use this result, it is convenient to work on trigraphs (a generalization of graphs where some edges are left "undecided"), and the first step is to obtain a decomposition theorem for bull-free trigraphs using homogeneous sets and homogeneous pairs. In this paper we need a simplified statement of this decomposition theorem, which we now describe formally.

### Trigraphs

For a set $X$, we denote by $\binom{X}{2}$ the set of all subsets of $X$ of size 2. For brevity of notation an element $\{u, v\}$ of $\binom{X}{2}$ is also denoted by $uv$ or $vu$. A *trigraph* $T$ consists of a finite set $V(T)$, called the *vertex set* of $T$, and a map $\theta : \binom{V(T)}{2} \longrightarrow \{-1, 0, 1\}$, called the *adjacency function*.

Two distinct vertices of $T$ are said to be *strongly adjacent* if $\theta(uv) = 1$, *strongly antiadjacent* if $\theta(uv) = -1$, and *semiadjacent* if $\theta(uv) = 0$. We say that $u$ and $v$ are *adjacent* if they are either strongly adjacent, or semiadjacent; and *antiadjacent* if they are either strongly antiadjacent, or semiadjacent. An *edge* (*antiedge*) is a pair of adjacent (antiadjacent) vertices. If $u$ and $v$ are adjacent (antiadjacent), we also say that $u$ is *adjacent (antiadjacent) to* $v$, or that $u$ is a *neighbor (antineighbor)* of $v$. Similarly, if $u$ and $v$ are strongly adjacent (strongly antiadjacent), then $u$ is a *strong neighbor (strong antineighbor)* of $v$.

Let $\eta(T)$ be the set of all strongly adjacent pairs of $T$, $\nu(T)$ the set of all strongly antiadjacent pairs of $T$, and $\sigma(T)$ the set of all semiadjacent pairs of $T$. Thus, a trigraph $T$ is a graph if $\sigma(T)$ is empty. A pair $\{u, v\} \subseteq V(T)$ of distinct vertices is a *switchable pair* if $\theta(uv) = 0$, a *strong edge* if $\theta(uv) = 1$ and a *strong antiedge* if $\theta(uv) = -1$. An edge $uv$ (antiedge, strong edge, strong antiedge, switchable pair) is *between* two sets $A \subseteq V(T)$ and $B \subseteq V(T)$ if $u \in A$ and $v \in B$, or if $u \in B$ and $v \in A$.

The *complement* $\overline{T}$ of $T$ is a trigraph with the same vertex set as $T$, and adjacency function $\overline{\theta} = -\theta$.

For $v \in V(T)$, $N(v)$ denotes the set of all vertices in $V(T) \setminus \{v\}$ that are adjacent to $v$; $\eta(v)$ denotes the set of all vertices in $V(T) \setminus \{v\}$ that are strongly adjacent to $v$; $\nu(v)$ denotes the set of all vertices in $V(T) \setminus \{v\}$ that are strongly antiadjacent to $v$; and $\sigma(v)$ denotes the set of all vertices in $V(T) \setminus \{v\}$ that are semiadjacent to $v$.

Let $A \subset V(T)$ and $b \in V(T) \setminus A$. We say that $b$ is *strongly complete* to $A$ if $b$ is strongly adjacent to every vertex of $A$; $b$ is *strongly anticomplete* to $A$ if $b$ is strongly antiadjacent to every vertex of $A$; $b$ is *complete* to $A$ if $b$ is adjacent to every vertex of $A$; and $b$ is *anticomplete* to $A$ if $b$ is antiadjacent to every vertex of $A$. For two disjoint subsets $A, B$ of $V(T)$, $B$ is *strongly complete (strongly anticomplete, complete, anticomplete)* to $A$ if every vertex of $B$ is strongly complete (strongly anticomplete, complete, anticomplete) to $A$. A set of vertices $X \subseteq V(T)$ *dominates (strongly dominates)* $T$ if for all $v \in V(T) \setminus X$, there exists $u \in X$ such that $v$ is adjacent (strongly adjacent) to $u$.

A *clique* in $T$ is a set of vertices all pairwise adjacent, and a *strong clique* is a set of vertices all pairwise strongly adjacent. A *stable set* is a set of vertices all pairwise antiadjacent, and a *strongly stable set* is a set of vertices all pairwise strongly antiadjacent. For $X \subset V(T)$ the trigraph *induced by $T$ on $X$* (denoted by $T[X]$) has vertex set $X$, and adjacency function that is the restriction of $\theta$ to $\binom{X}{2}$. Isomorphism between trigraphs is defined in the natural way, and for two trigraphs $T$ and $H$ we say that $H$ is an *induced subtrigraph* of $T$ (or $T$ *contains* $H$ *as an induced subtrigraph*) if $H$ is isomorphic to $T[X]$ for some $X \subseteq V(T)$. Since in this paper we are only concerned with the induced subtrigraph containment relation, we say that $T$ *contains* $H$ if $T$ contains $H$ as an induced subtrigraph. We denote by $T \setminus X$ the trigraph $T[V(T) \setminus X]$.

Let $T$ be a trigraph. A *path* $P$ of $T$ is a sequence of distinct vertices $p_1, \dots, p_k$ such that $k \geq 1$ and for $i, j \in \{1, \dots, k\}$, $p_i$ is adjacent to $p_j$ if $|i - j| = 1$ and $p_i$ is antiadjacent to $p_j$ if $|i - j| > 1$. Under these circumstances, $V(P) = \{p_1, \dots, p_k\}$ and we say that $P$ is a path *from* $p_1$ *to* $p_k$, its *interior* is the set $P^* = V(P) \setminus \{p_1, p_k\}$, and the *length* of $P$ is $k - 1$. We also say that $P$ is a $(k-1)$-*edge-path*. Sometimes, we denote $P$ by $p_1\text{-}\cdots\text{-}p_k$. Observe that, since a graph is also a trigraph, it follows that a path in a graph, the way we have defined it, is what is sometimes in literature called a chordless path.

A *semirealization* of a trigraph $T$ is any trigraph $T'$ with vertex set $V(T)$ that satisfies the following: for all $uv \in \binom{V(T)}{2}$, if $uv \in \eta(T)$ then $uv \in \eta(T')$, and if $uv \in \nu(T)$ then $uv \in \nu(T')$. Sometimes we will describe a semirealization of $T$ as an *assignment of values* to switchable pairs of $T$, with three possible values: "strong edge", "strong antiedge" and "switchable pair". A *realization* of $T$ is any graph that is semirealization of $T$ (so, any semirealization where all switchable pairs are assigned the value "strong edge" or "strong antiedge"). For $S \subseteq \sigma(T)$, we denote by $G^T_S$ the realization of $T$ with edge set $\eta(T) \cup S$, so in $G^T_S$ the switchable pairs in $S$ are assigned the value "edge", and those in $\sigma(T) \setminus S$ the value "antiedge". The realization $G^T_{\sigma(T)}$ is called the *full realization* of $T$.

A *bull* is a trigraph with vertex set $\{x_1, x_2, x_3, y, z\}$ such that $x_1, x_2, x_3$ are pairwise adjacent, $y$ is adjacent to $x_1$ and antiadjacent to $x_2, x_3, z$, and $z$ is adjacent to $x_2$ and antiadjacent to $x_1, x_3$. For a trigraph $T$, a subset $X$ of $V(T)$ is said to be a bull if $T[X]$ is a bull. A trigraph is *bull-free* if no induced subtrigraph of it is a bull, or equivalently, no subset of its vertex set is a bull.

Observe that we have two notions of bulls: bulls as graphs (defined in the introduction), and bulls as trigraphs. A bull as a graph can be seen as a bull as a trigraph. Also, a trigraph is a bull if and only if at least one of its realization is a bull (as a graph). Hence, a trigraph is bull-free if and only if all its realizations are bull-free graphs. The complement of a bull is a bull (with both notions), and therefore, if $T$ is bull-free trigraph (or graph), then so is $\overline{T}$.

### Decomposition theorem

A trigraph is called *monogamous* if every vertex of it belongs to at most one switchable pair (so the switchable pairs form a matching). We now state the decomposition theorem for bull-free monogamous trigraphs. We begin with the description of the cutsets.

Let $T$ be a trigraph. A set $X \subseteq V(T)$ is a *homogeneous set* in $T$ if $1 < |X| < |V(T)|$, and every vertex of $V(T) \setminus X$ is either strongly complete or strongly anticomplete to $X$. See Figure 1 (a line means all possible strong edges between two sets, nothing means all possible strong antiedges, and a dashed line means no restriction).

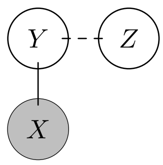


**Fig. 1.** A homogeneous set.

A *homogeneous pair* (see Figure 2) is a pair of disjoint nonempty subsets $(A, B)$ of $V(T)$, such that there are disjoint (possibly empty) subsets $C, D, E, F$ of $V(T)$ whose union is $V(T) \setminus (A \cup B)$, and the following hold:

- $A$ is strongly complete to $C \cup E$ and strongly anticomplete to $D \cup F$;
- $B$ is strongly complete to $D \cup E$ and strongly anticomplete to $C \cup F$;
- $A$ is not strongly complete and not strongly anticomplete to $B$;
- $|A \cup B| \geq 3$; and
- $|C \cup D \cup E \cup F| \geq 3$.

In these circumstances, we say that $(A, B, C, D, E, F)$ is a *split* for the homogeneous pair $(A, B)$. A homogeneous pair $(A, B)$ is *small* if $|A \cup B| \leq 6$. A homogeneous pair $(A, B)$ with split $(A, B, C, D, E, F)$ is *proper* if $C \neq \emptyset$ and $D \neq \emptyset$. Note that "$A$ is not strongly complete and not strongly anticomplete to $B$" does not imply that $|A \cup B| \geq 3$, because it could be that the unique vertex in $A$ is linked to the unique vertex in $B$ by a switchable pair.

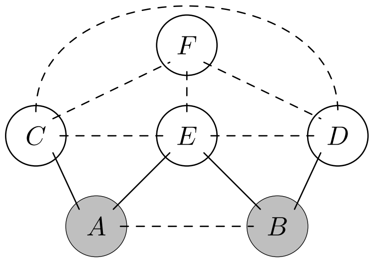


**Fig. 2.** A homogeneous pair.

We now describe the basic classes. A trigraph is a *triangle* if it has exactly three vertices, and these vertices are pairwise adjacent. Let $\mathcal{T}_0$ be the class of all monogamous trigraphs on at most 8 vertices. Let $\mathcal{T}_1$ be the class of monogamous trigraphs $T$ whose vertex set can be partitioned into (possibly empty) sets $X, K_1, \dots, K_t$ so that $T[X]$ is triangle-free, and $K_1, \dots, K_t$ are strong cliques that are pairwise strongly anticomplete. Furthermore, for every $v \in \cup_{i=1}^{t} K_i$, the set of neighbors of $v$ in $X$ partitions into strong stable sets $A$ and $B$ such that $A$ is strongly complete to $B$. Let $\overline{\mathcal{T}_1} = \{\overline{T} : T \in \mathcal{T}_1\}$. A trigraph is *basic* if it belongs to $\mathcal{T}_0 \cup \mathcal{T}_1 \cup \overline{\mathcal{T}_1}$. The following result is a direct consequence of the main result of Chudnovsky. Note that this is a simplification of a much more detailed characterization.

**Theorem 1 (Chudnovsky [4–7]).** *If $T$ is a bull-free monogamous trigraph, then one of the following holds:*

- *$T$ is basic;*
- *$T$ has a homogeneous set;*
- *$T$ has a small homogeneous pair; or*
- *$T$ has a proper homogeneous pair.*

We do not know whether the theorem above is algorithmic. Deciding whether a graph is bull-free can clearly be done in polynomial time. Also, detecting the decompositions is easy (see Section 4). The problem is with the basic classes. It follows directly from a theorem of Farrugia [10] that deciding whether a graph can be partitioned into a triangle-free part and a part that is disjoint union of cliques is NP-complete. This does not mean that recognizing $\mathcal{T}_1$ is NP-complete, because one could take advantage of several features, such as being bull-free or of the full definition of $\mathcal{T}_1$ in [5]. We leave the recognition of $\mathcal{T}_1$ as an open question.

## 3 Extreme decompositions

The way we use decompositions for computing stable sets requires building blocks of decomposition and asking at least two questions for at least one block.

When this process is recursively applied it potentially leads to an exponential blow-up even when the decomposition tree is linear in the size of the input trigraph. This problem is bypassed here by using what we call extreme decompositions, that are decompositions whose one block of decomposition is basic and therefore handled directly, without any recursive calls to the algorithm.

In this section, we prove that non-basic trigraphs in our class actually have extreme decompositions. We start by describing the blocks of decomposition for the cutsets used in Theorem 1.

We say that $(X, Y)$ is a *decomposition* of a trigraph $T$ if $(X, Y)$ is a partition of $V(T)$ and either $X$ is a homogeneous set of $T$, or $X = A \cup B$ where $(A, B)$ is a small homogeneous pair or a proper homogeneous pair of $T$. The *block of decomposition w.r.t.* $(X, Y)$ *that corresponds to* $X$, denoted by $T_X$, is defined as follows. If $X$ is a homogeneous set or a small homogeneous pair, then $T_X = T[X]$. Otherwise, $X = A \cup B$ where $(A, B)$ is a proper homogeneous pair, and $T_X$ consists of $T[X]$ together with *marker vertices* $c$ and $d$ such that $c$ is strongly complete to $A$, $d$ is strongly complete to $B$, $cd$ is a switchable pair, and there are no other edges between $\{c, d\}$ and $A \cup B$. The *block of decomposition w.r.t.* $(X, Y)$ *that corresponds to* $Y$, denoted by $T_Y$, is defined as follows. If $X$ is a homogeneous set, then let $x$ be any vertex of $X$ and let $T_Y = T[Y \cup \{x\}]$. In this case $x$ is called the *marker vertex* of $T_Y$. Otherwise, $X = A \cup B$ where $(A, B)$ is a homogeneous pair with split $(A, B, C, D, E, F)$. In this case $T_Y$ consists of $T[Y]$ together with two new *marker vertices* $a$ and $b$ such that $a$ is strongly complete to $C \cup E$, $b$ is strongly complete to $D \cup E$, $ab$ is a switchable pair, and there are no other edges between $\{a, b\}$ and $C \cup D \cup E \cup F$.

**Lemma 1.** *If* $(X, Y)$ *is a decomposition of a bull-free monogamous trigraph* $T$*, then the corresponding blocks* $T_X$ *and* $T_Y$ *are bull-free monogamous trigraphs.*

Let $(X, Y)$ be a decomposition of a trigraph $T$. We say that $(X, Y)$ is a *homogeneous cut* if $X$ is a homogeneous set or $X = A \cup B$ where $(A, B)$ is a proper homogeneous pair. A homogeneous cut $(X, Y)$ is *minimally-sided* if there is no homogeneous cut $(X', Y')$ with $X' \subsetneq X$.

**Lemma 2.** *If* $(X, Y)$ *is a minimally-sided homogeneous cut of a trigraph* $T$*, then the block of decomposition* $T_X$*, has no homogeneous cut.*

**Theorem 2.** *Let* $T$ *be a bull-free monogamous trigraph that has a decomposition. If* $T$ *has a small homogeneous pair* $(A, B)$*, then let* $X = A \cup B$ *and* $Y = V(T) \setminus X$*. Otherwise let* $(X, Y)$ *be minimally-sided homogeneous cut of* $T$*. Then the block of decomposition* $T_X$ *is basic.*

## 4 Algorithms for finding decompositions

The fastest known algorithm for finding a homogeneous set in a graph is linear time (see Habib and Paul [14]) and the fastest one for the homogeneous pair runs in time $O(n^2 m)$ (see Habib, Mamcarz, and de Montgolfier [13]). But we cannot use these algorithms safely here because we need minimally-sided decompositions with several technical requirements ("small", "proper") and we need our algorithms to work for trigraphs. However, it turns out that all classical ideas work well in our context.

A 4-tuple of vertices $(a, b, c, d)$ of a trigraph is *proper* if $ac$ and $bd$ are strong edges and $bc$ and $ad$ are strong antiedges. A proper 4-tuple $(a, b, c, d)$ is *compatible* with a homogeneous pair $(A, B)$ if $a \in A$, $b \in B$ and $c, d \notin A \cup B$ (note that $c, d$ must be respectively in the sets $C, D$ from the definition of a split of a homogeneous pair).

**Lemma 3.** *Let $T$ be a trigraph and $Z = (a, b, c, d)$ a proper 4-tuple of $T$. There is an $O(n^2)$ time algorithm that given a set $R_0 \subseteq V(T)$ of size at least 3 such that $Z \cap R_0 = \{a, b\}$, either outputs two sets $A$ and $B$ such that $(A, B)$ is a proper homogeneous pair of $T$ compatible with $Z$ and such that $R_0 \subseteq A \cup B$, or outputs the true statement "There exists no proper homogeneous pair $(A, B)$ in $T$ compatible with $Z$ and such that $R_0 \subseteq A \cup B$".*

*Moreover, when $(A, B)$ is output, $A \cup B$ is minimal with respect to these properties, meaning that $A \cup B \subseteq A' \cup B'$ for every homogeneous pair $(A', B')$ satisfying the properties.*

**Lemma 4.** *Let $T$ be a trigraph and $(a, b)$ a pair of vertices from $T$. There is an $O(n^2)$ time algorithm that given a set $R_0 \subseteq V(T)$ such that $a, b \in R_0$, either outputs a homogeneous set $X$ such that $R_0 \subseteq X$, or outputs the true statement "There exists no homogeneous set $X$ in $T$ such that $R_0 \subseteq X$".*

*Moreover, when $X$ is output, $X$ is minimal with respect to these properties, meaning that $X \subseteq X'$ for every homogeneous set $X'$ satisfying the properties.*

**Theorem 3.** *There exists an $O(n^8)$ time algorithm whose input is a trigraph $T$. The output is a small homogeneous pair of $T$ if some exists. Otherwise, if $G$ has a homogeneous cut, then the output is a minimally-sided homogeneous cut. Otherwise, the output is: "$T$ has no small homogeneous pair, no proper homogeneous pair and no homogeneous set".*

## 5 Weighted trigraphs

For the sake of induction, we need to work with weighted trigraphs. Here, a *weight* is a non-negative integer. By a *weighted trigraph with weight function $w$*, we mean a trigraph $T$ such that:

- every vertex $a$ has a weight $w(a)$;
- every switchable pair $ab$ of $T$ has a weight $w(ab)$;
- for every switchable pair $ab$, $\max\{w(a), w(b)\} \leq w(ab) \leq w(a) + w(b)$.

Let $S$ be a stable set of $T$. Recall that $\nu(T)$ denotes the set of all strongly antiadjacent pairs of $T$, and $\sigma(T)$ the set of all semiadjacent pairs of $T$. We set $c(S) = \{v \in S : \forall u \in S \setminus \{v\}, uv \in \nu(T)\}$. We set $\sigma(S) = \{uv \in \sigma(T) : u, v \in S\}$.

Observe that if $T$ is monogamous, then for every vertex $v$ of $S$, one and only one of the following outcomes is true: $v \in c(S)$ or for some unique $w \in S$, $vw \in \sigma(S)$. The *weight of a stable set* $S$ is the sum of the weights of the vertices in $c(S)$ and of the weights of the (switchable) pairs in $\sigma(S)$. From here on, $T$ is a weighted monogamous trigraph and $\alpha(T)$ denotes the maximum weight of a stable set of $T$.

When $(X, Y)$ is a decomposition of $T$, we already defined the block $T_Y$. We now explain how to give weights to the marker vertices and switchable pairs in $T_Y$. Every vertex and switchable pair in $T[Y]$ keeps its weight. If $X$ is a homogeneous set, then the marker vertex $x$ receives weight $\alpha(T[X])$. If $X = A \cup B$ where $(A, B)$ is a homogeneous pair, then we give weight $\alpha_A = \alpha(T[A])$ to marker vertex $a$, $\alpha_B = \alpha(T[B])$ to marker vertex $b$ and $\alpha_{AB} = \alpha(T[A \cup B])$ to the switchable pair $ab$. It is easy to check that the inequalities in the definition of a weighted trigraph are satisfied.

**Lemma 5.** $\alpha(T) = \alpha(T_Y)$.

Let $T$ be a weighted monogamous trigraph with weight function $w$ and a switchable pair $ab$. We now define four ways to get rid of the switchable pair $ab$ while keeping $\alpha$ the same. This is needed because sometimes we rely on algorithms for *graphs*. There are four ways because $a$ (resp. $b$) can be transformed into a strong edge or a strong antiedge. Only one way is needed in this section, but in Section 7, the four ways are needed.

The weighted monogamous trigraph $T_{a \to S}$ (resp. $T_{b \to S}$) is constructed as follows: replace switchable pair $ab$ with a strong edge $ab$; add a new vertex $a'$ (resp. $b'$) and make it strongly complete to $N_T(a) \setminus \{b\}$ (resp. $N_T(b) \setminus \{a\}$) and strongly anticomplete to the remaining vertices; keep the weights of vertices and switchable pairs of $T \setminus \{a\}$ (resp. $T \setminus \{b\}$) the same; assign the weight $w(a) + w(b) - w(ab)$ to $a$ (resp. $w(a) + w(b) - w(ab)$ to $b$) and the weight $w(ab) - w(b)$ to $a'$ (resp. $w(ab) - w(a)$ to $b'$).

The weighted monogamous trigraph $T_{a \to K}$ (resp. $T_{b \to K}$) is constructed as follows: replace switchable pair $ab$ with a strong edge $ab$; add a new vertex $a'$ (resp. $b'$) and make it strongly complete to $\{a\} \cup N_T(a) \setminus \{b\}$ (resp. $\{b\} \cup N_T(b) \setminus \{a\}$) and strongly anticomplete to the remaining vertices; keep the weights of vertices and switchable pairs of $T \setminus \{a\}$ (resp. $T \setminus \{b\}$) the same; assign the weight $w(a)$ to $a$ (resp. $w(b)$ to $b$) and the weight $w(ab) - w(b)$ to $a'$ (resp. $w(ab) - w(a)$ to $b'$).

Note that by the inequalities in the definition of a weighted trigraph, all weights of vertices in $T_{a \to S}$, $T_{b \to S}$, $T_{a \to K}$ and $T_{b \to K}$ are nonnegative.

**Lemma 6.** *If $T$ is a weighted monogamous trigraph and $ab$ is a switchable pair of $T$, then $\alpha(T_{a \to S}) = \alpha(T_{b \to S}) = \alpha(T_{a \to K}) = \alpha(T_{b \to K}) = \alpha(T)$.*

## 6 Computing $\alpha$ in bull-free graphs

In this section, we use positive weights (no vertex nor switchable pair in a trigraph has weight 0). Also, switchable pairs have weight at least 2.

**Lemma 7.** *If $T$ is a trigraph from $\overline{\mathcal{T}_1}$, then $T$ contains at most $|V(T)|^3$ maximal stable sets.*

We need the next classical algorithm that we use as a subroutine. For faster implementations (that we do not need here), see Makino and Uno [18].

**Theorem 4 (Tsukiyama, Ide, Ariyoshi, and Shirakawa [21]).** *There exists an algorithm for generating all maximal stable sets in a given graph $G$ that runs with $O(nm)$ time delay (i.e. the computation time between any consecutive output is bounded by $O(nm)$; and the first (resp. last) output occurs also in $O(nm)$ time after start (resp. before halt) of the algorithm).*

**Lemma 8.** *There exists an $O(n^4m)$ time algorithm whose input is any trigraph $T$ and whose output is a maximum weighted stable set of $T$, or a certificate that $T$ is not in $\overline{\mathcal{T}_1}$.*

Let $R(x, y)$ be the smallest integer $n$ such that every graph on at least $n$ vertices contains a clique of size $x$ or a stable set of size $y$. By a classical theorem of Ramsey, $R(3, x) \leq \binom{x+1}{2}$. We now define two functions $g$ and $f$ by $g(x) = \binom{x+1}{2} - 1$ and $f(x) = g(x) + (x-1)(\binom{g(x)}{2} + 2g(x) + 1)$. Note that $f(x) = O(x^5)$. The next lemma handles basic trigraphs.

**Lemma 9.** *There exists an $O(n^4m)$-time algorithm with the following specifications.*

**Input:** *A weighted monogamous basic trigraph $T$ on $n$ vertices, in which all vertices have weight at least 1 and all switchable pairs have weight at least 2, with no homogeneous set, and a positive integer $W$.*
**Output:** *One of the following true statements.*
1. *$n \leq f(W)$;*
2. *the number of maximal stable sets in $T$ is at most $n^3$;*
3. *$\alpha(T) \geq W$.*

**Theorem 5.** *There is an algorithm with the following specification.*

**Input:** *A weighted monogamous bull-free trigraph $T$ and a positive integer $W$.*
**Output:** *"YES" if $\alpha(T) \geq W$ and otherwise an independent set of maximum weight.*
**Running time:** $2^{O(W^5)}n^9$

## 7 A polynomial Turing-kernel

Once an FPT-algorithm is found, the natural question is to ask for a polynomial kernel for the problem. Precisely, is there a polynomial-time algorithm which takes as input a bull-free graph $G$ and a parameter $k$ and outputs a bull-free graph $H$ with at most $O(k^c)$ vertices and some integer $k'$ such that $G$ has a stable set of size $k$ if and only if $H$ has a stable set of size $k'$. Unfortunately, the problem is OR-compositional and thus we have the following:

**Theorem 6.** *Unless NP $\subseteq$ coNP/poly, there is no polynomial kernel for the problem $\alpha(G) \geq k$, where $G$ is a bull-free graph and $k$ is the parameter.*

Somewhat surprisingly, the non existence of a polynomial kernel is not related to the hard core of the algorithm (computing the leaves) but is related to the decomposition tree itself (since even complete sums cannot be handled). Indeed, our algorithm is a kind of kernelisation: the answer is obtained in polynomial time provided that we compute a stable set in a linear number of basic trigraphs of size at most $k^5$ (the leaves of our implicit decomposition tree). A similar behaviour was discovered by Fernau et al [12] in the case of finding a directed tree with at least $k$ leaves in a digraph (*Maximum Leaf Outbranching problem*): a polynomial kernel does not exist, but $n$ polynomial kernels can be found. In our case, the leaves of the decomposition tree are pairwise dependent, hence our method does not provide $O(n^c)$ independent kernels of size $O(k^5)$. It seems that the notion of kernel is not robust enough to capture this kind of behaviour in which the computationally hard cases of the problem admit polynomial kernels, but the (computationally easy) decomposition structure does not.

Let $f$ be a computable function. A parameterized problem has an $f$*-Turing-kernel* (see Lokshtanov [16]) if there exists a constant $c$ such that computing the solution of any instance $(X, k)$ can be done in $O(n^c)$ provided that we have unlimited access to an oracle which can decide any instance $(X', k')$ where $(X', k')$ has size at most $f(k)$.

**Theorem 7.** *Stability in bull-free weighted trigraphs (resp. graphs) has an $O(k^5)$-Turing-kernel. The unweighted versions of both problems also have an $O(k^5)$-Turing-kernel.*

## Acknowledgement

Thanks to Maria Chudnovsky for several suggestions. Thanks to Haiko Müller for pointing out to us [10]. Thanks to Sébastien Tavenas and the participants to GROW 2013 for useful discussions on Turing-kernels.

## References

1. N. Alon and J.H. Spencer. *The Probabilistic Method.* Wiley, 2008.
2. H. Bodlaender, R. Downey, M. Fellows, D. Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75 (8): 423-434, 2009.
3. A. Brandstädt and R. Mosca Maximum Weight Independent Sets in Odd-Hole-Free Graphs Without Dart or Without Bull. preprint
4. M. Chudnovsky. The structure of bull-free graphs I: Three-edge-paths with center and anticenters. *Journal of Combinatorial Theory B*, 102 (1): 233–251, 2012.
5. M. Chudnovsky. The structure of bull-free graphs II and III: A summary. *Journal of Combinatorial Theory B*, 102 (1): 252–282, 2012.
6. M. Chudnovsky, The structure of bull-free graphs II: Elementary trigraphs, manuscript.

7. M. Chudnovsky, The structure of bull-free graphs III: Global structure, manuscript.
8. K. Dabrowski, V. Lozin, H. Müller, and D. Rautenbach. Parameterized complexity of the weighted independent set problem beyond graphs of bounded clique number. *Journal of Discrete Algorithms*, 14:207–213, 2012.
9. R.G. Downey, and M.R. Fellows. Parameterized Complexity, *Monographs in Computer Science*, Springer, New York, 1999.
10. A. Farrugia. Vertex-partitioning into fixed additive induced-hereditary properties is NP-hard. *The Electronic Journal of Combinatorics*, 11(1), 2004.
11. D. Fernández-Baca, editor. *LATIN 2012: Theoretical Informatics - 10th Latin American Symposium, Arequipa, Peru, April 16–20, 2012. Proceedings*, volume 7256 of *Lecture Notes in Computer Science*. Springer, 2012.
12. H. Fernau, F. Fomin, D. Lokshtanov, D. Raible, S. Saurabh, and Y. Villanger. Kernels for Problems with No Kernel: On Out-Trees with Many Leaves. *STACS 2009*.
13. M. Habib, A. Mamcarz, and F. de Montgolfier. Algorithms for some $H$-join decompositions. In Fernández-Baca [11], pages 446–457.
14. M. Habib and C. Paul. A survey of the algorithmic aspects of modular decomposition. *Computer Science Review*, 4(1):41–59, 2010.
15. D. Hermelin, S. Kratsch, K. Soltys, M. Wahlström and X. Wu. *A Completeness Theory for Polynomial (Turing) Kernelization* IPEC, 2013.
16. D. Lokshtanov. *New Methods in Parameterized Algorithms and Complexity.* PhD thesis, University of Bergen, 2009.
17. D Lokshtanov, M. Vatshelle and Y. Villanger. Independent set in $p_5$-free graphs in polynomial time, 2014. SODA 2014.
18. K. Makino and T. Uno. New algorithms for enumerating all maximal cliques. SWAT 2004, 9th Scandinavian Workshop on Algorithm Theory, Humlebaek, Denmark, July 8-10, 2004, Proceedings, volume 3111 of *Lecture Notes in Computer Science*. Springer, 2004, pages 260–272.
19. N. Sauer. On the Density of Families of Sets. *J. Comb. Theory, Ser. A*, 13 (1): 145-147, 1972.
20. A. Schrijver. *Combinatorial Optimization, Polyhedra and Efficiency*, volume A, B and C. Springer, 2003.
21. S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa. A new algorithm for generating all the maximal independent sets. *SIAM Journal on Computing*, 6(3):505–517, 1977.

## Appendix. Some omitted proofs

**Proof of Lemma 1** Since all the edges in the blocks that go from marker vertices to the rest of the block are strong edges, it follows that $T_X$ and $T_Y$ are both monogamous trigraphs.

Suppose that $T_X$ or $T_Y$ contains a bull $H$. Since $H$ cannot be isomorphic to an induced subtrigraph of $T$, it follows that $X = A \cup B$ where $(A, B)$ is a homogeneous pair of $T$ and $H$ contains two marker vertices from the block. In a bull every pair of vertices has a common neighbor or a common antineighbor. Since $c$ and $d$ do not have a common neighbor nor a common antineighbor in $T_X$, it follows that $H$ is a bull of $T_Y$ and $H$ contains $a$ and $b$. But then, since $A$ is not strongly complete nor strongly anticomplete to $B$, for some $a' \in A$ and $b' \in B$, $(V(H) \setminus \{a, b\}) \cup \{a', b'\}$ induces a bull in $T$, a contradiction. ∎

**Proof of Lemma 2** Assume not and let $(X', Y')$ be a homogeneous cut of $T_X$. We now consider the following two cases.

*Case 1:* $X$ is a homogeneous set of $T$.
Since every vertex of $V(T) \setminus X$ is either strongly complete or strongly anticomplete to $X$, it follows that $(X', V(T) \setminus X')$ is a homogeneous cut of $T$, contradicting our choice of $(X, Y)$ since $X' \subsetneq X$.

*Case 2:* $X = A \cup B$ where $(A, B)$ is a proper homogeneous pair of $T$ with split $(A, B, C, D, E, F)$.
Since $cd$ is a switchable pair of $T_X$, $\{c, d\} \subseteq X'$ or $\{c, d\} \subseteq Y'$.

Suppose that $X'$ is a homogeneous set of $T_X$. Since $c$ and $d$ do not have a common strong neighbour nor a common strong antineighbor, it follows that $\{c, d\} \subseteq Y'$. Since $c$ is strongly complete to $A$ and strongly anticomplete to $B$, $X' \subseteq A$ or $X' \subseteq B$. But then $X'$ is a homogeneous set of $T$, contradicting our choice of $(X, Y)$.

Therefore, $X' = A' \cup B'$ where $(A', B')$ is a proper homogeneous pair of $T_X$ with split $(A', B', C', D', E', F')$. First assume that $\{c, d\} \subseteq Y'$. Since $c$ is strongly complete to $A$ and strongly anticomplete to $B$, it follows that $A' \subseteq A$ or $A' \subseteq B$, and $B' \subseteq A$ or $B' \subseteq B$. Hence $(A', B')$ is a homogeneous pair of $T$. We now obtain a contradiction to the choice of $(X, Y)$ by showing that $(A', B')$ is in fact a proper homogeneous pair of $T$. If $A' \cup B' \subseteq A$, then $c \in E'$, $d \in F'$ (i.e. $(C' \cup D') \cap \{c, d\} = \emptyset$) and hence, since $C'$ and $D'$ are nonempty, $(A', B')$ is a proper homogenous pair of $T$. So by symmetry we may assume that $A' \subseteq A$ and $B' \subseteq B$. But then since $C$ and $D$ are nonempty, and $C$ (resp. $D$) is strongly complete to $A$ (resp. $B$) and strongly anticomplete to $B$ (resp. $A$), it follows that $(A', B')$ is a proper homogeneous pair of $T$.

Now assume that $\{c, d\} \subseteq X'$. Since $C'$ and $D'$ are nonempty, and no vertex of $T_X$ is strongly complete nor strongly anticomplete to $\{c, d\}$, we may assume w.l.o.g. that $c \in A'$ and $d \in B'$. Hence $E' = F' = \emptyset$, $C' \subseteq A$ and $D' \subseteq B$. If $C'$ is strongly complete or strongly anticomplete to $D'$, then since $|C' \cup D'| \geq 3$, $C'$ or $D'$ is a homogeneous set of $T_X$ and we obtain a contradiction as above. So we may assume that $C'$ is not strongly complete nor strongly anticomplete to $D'$. But then, since $C$ and $D$ are nonempty, $(C', D')$ is a proper homogeneous pair of $T$, contradicting our choice of $(X, Y)$. ∎

**Proof of Theorem 2** If $X = A \cup B$ where $(A, B)$ is a small homogenous pair then clearly $T_X \in \mathcal{T}_0$, so assume that $T$ has no small homogeneous pair and that $(X, Y)$ is a minimally-sided homogeneous cut of $T$. By Lemma 2, $T_X$ has no homogeneous cut. If $T_X$ has no small homogeneous pair, then by Theorem 1 and Lemma 1, $T_X \in \mathcal{T}_0 \cup \mathcal{T}_1 \cup \overline{\mathcal{T}_1}$. So assume that $T_X$ has a small homogeneous pair with split $(A', B', C', D', E', F')$. Set $X' = A' \cup B'$ and $Y' = V(T_X) \setminus X'$. We now consider the following two cases.

*Case 1:* $X$ is a homogeneous set of $T$.
Since every vertex of $V(T) \setminus X$ is either strongly complete or strongly anticomplete to $X$, it follows that $(X', V(T) \setminus X')$ is a small homogeneous pair of $T$,

contradicting the assumption that $T$ has no small homogeneous pair.

*Case 2:* $X = A \cup B$ where $(A, B)$ is a proper homogeneous pair of $T$ with split $(A, B, C, D, E, F)$.
Since $cd$ is a switchable pair of $T_X$, $\{c, d\} \subseteq X'$ or $\{c, d\} \subseteq Y'$.

First assume that $\{c, d\} \subseteq Y'$. Since $c$ is strongly complete to $A$ and strongly anticomplete to $B$, it follows that $A' \subseteq A$ or $A' \subseteq B$, and $B' \subseteq A$ or $B' \subseteq B$. Hence $(A', B')$ is a small homogeneous pair of $T$, contradicting the assumption that $T$ has no small homogeneous pair.

Now assume that $\{c, d\} \subseteq X'$. Since no vertex of $T_X$ is strongly complete nor strongly anticomplete to $\{c, d\}$ we may assume w.l.o.g. that $c \in A'$ and $d \in B'$. Hence $E' = F' = \emptyset$, $C' \subseteq A$ and $D' \subseteq B$. But then, since $C$ and $D$ are nonempty, either $(C', D')$ is a proper homogeneous pair of $T$ or a subset of $C' \cup D'$ is a homogeneous set of $T$ (if $C'$ is either strongly complete or strongly anticomplete to $D'$), contradicting the minimality of $(X, Y)$. ∎

**Proof of Lemma 3** We set $R = R_0$ and $S = V(T) \setminus R$, and we implement several forcing rules, stating that some sets of vertices must be moved from $S$ to $R$.

We give mark $\alpha$ to all vertices of $V(T)$ that are strongly adjacent to $c$ and strongly antiadjacent to $d$. We give mark $\beta$ to all vertices of $V(T)$ that are strongly adjacent to $d$ and strongly antiadjacent to $c$. We give mark $\varepsilon$ to all vertices of $V(T)$ not marked so far. Observe that $a$, $b$, $c$ and $d$ receive marks $\alpha$, $\beta$, $\varepsilon$ and $\varepsilon$ respectively.

Vertices of $R$ should be thought of as "vertices that must be in $A \cup B$". Vertices with mark $\alpha$ should be thought of as "vertices that are in $A$ if they are in $R$"; vertices with mark $\beta$ should be thought of as "vertices that are in $B$ if they are in $R$"; and vertices with mark $\varepsilon$ should be thought of as "vertices that should not be in $R$" . Note that the adjacency to $c$ and $d$ is enough to distinguish the three cases, and this is why the marks are not changed during the process.

Here are the rules. While there exists a vertex $x \in R$ that is marked, we apply them to $x$, and we unmark $x$.

- If $x$ has mark $\varepsilon$, then stop and output "There exists no homogeneous pair $(A, B)$ in $T$ compatible with $Z$ and such that $R_0 \subseteq A \cup B$".
- If $x$ has mark $\alpha$, then move the following sets from $S$ to $R$: $\sigma(x) \cap S$, $(\eta(a) \cap S) \setminus \eta(x)$ and $(\eta(x) \cap S) \setminus \eta(a)$.
- If $x$ has mark $\beta$, then move the following sets from $S$ to $R$: $\sigma(x) \cap S$, $(\eta(b) \cap S) \setminus \eta(x)$ and $(\eta(x) \cap S) \setminus \eta(b)$.

If a vertex with mark $\varepsilon$ is in $R$, then no homogeneous pair compatible with $(a, b, c, d)$ contains all vertices of $R$; this explains the first rule. If a vertex $x$ is in $R$, then all switchable pairs with end $x$ must be entirely in $R$; this explains why we move $\sigma(x) \cap S$ to $R$. If a vertex $x$ in $R$ has mark $\alpha$, it must share the same neighborhood in $S$ as $a$; this explains the second rule. The third rule is explained similarly for vertices marked $\beta$.

The following properties are easily checked to be invariant during all the execution of the procedure. This means that they are true before we start applying the rules, and they remain true after applying the rules to each vertex.

- $R$ and $S$ form a partition of $V(T)$ and $R_0 \subseteq R$.
- For all unmarked $v \in R$, and all $u \in S$, $uv$ is not a switchable pair.
- All unmarked vertices belonging to $R \cap \eta(c)$ have the same neighborhood in $S$, namely $S \cap \eta(a)$ (and it is a strong neighborhood).
- All unmarked vertices belonging to $R \cap \eta(d)$ have the same neighborhood in $S$, namely $S \cap \eta(b)$ (and it is a strong neighborhood).
- For every homogenous pair $(A, B)$ compatible with $(a, b, c, d)$ such that $R_0 \subseteq A \cup B$, we have $R \subseteq A \cup B$ and $V(T) \setminus (A \cup B) \subseteq S$.

By the last item all moves from $S$ to $R$ are necessary. This is why the algorithm reports a failure if some vertex of $R$ has mark $\varepsilon$. If the process does not stop for that particular reason, then all vertices of $R$ have been explored and are unmarked. Note that $|R| \geq 3$ since $R_0 \subseteq R$. So, if $|S| \geq 3$ at the end, we set $A = R \cap \eta(c)$, $B = R \cap \eta(d)$, and we observe that $(A, B)$ is a proper homogeneous pair.

Since all moves from $S$ to $R$ are necessary, the homogeneous pair is minimal as claimed. This also implies that if $|S| < 3$, then no proper homogeneous pair exists and we output this. ∎

**Proof of Lemma 4** The proof is similar to the previous one, so we just give a sketch. We mark all vertices except $a$ and we move $\sigma(a)$ to $R$. While there exists a marked vertex $x$ in $R$, we move $\sigma(x)$, $\eta(x) \setminus \eta(a)$ and $\eta(a) \setminus \eta(x)$ to $R$, and we unmark $x$. ∎

**Proof of Theorem 3** We search for a small homogeneous pair by enumerating all sets of vertices of size at most 6. This can be done in time $O(n^8)$ ($n^6$ for the enumeration, and $n^2$ to check wether a given small set is a homogeneous pair). If no small homogeneous pair is detected, we first run the algorithm from Lemma 4 for all pairs of vertices. We then run the algorithm from Lemma 3 for all proper 4-tuples $(a, b, c, d)$ of $T$ and vertex $e$ with $R_0 = \{a, b, e\}$. Among the (possibly) outputted homogeneous sets and pairs, we choose one of minimum cardinality. This forms a minimally-sided cut. ∎

**Proof of Lemma 5** If $X$ is a homogeneous set, then this is clearly true since if a maximum weight stable set $S$ of $T$ contains a vertex of $X$, then $S \cap X$ is a maximum weight stable set of $T[X]$.

Suppose that $X = A \cup B$ where $(A, B)$ is a homogeneous pair with split $(A, B, C, D, E, F)$. Let $S$ be a maximum weighted stable set of $T$. If $S \cap (A \cup B) = \emptyset$, then $S$ is a stable set of $T_Y$. If $\emptyset \subsetneq S \cap (A \cup B) \subseteq A$, then $S \cap A$ is a stable set of $T$ of weight $\alpha_A$, and hence $(S \setminus A) \cup \{a\}$ is a stable set of $T_Y$ of the same weight as $S$. If $\emptyset \subsetneq S \cap (A \cup B) \subseteq B$, then $S \cap B$ is a stable set of $T$ of weight $\alpha_B$, and hence $(S \setminus B) \cup \{b\}$ is a stable set of $T_Y$ of the same weight as $S$. If $S \cap A \neq \emptyset$ and $S \cap B \neq \emptyset$, then $S \cap (A \cup B)$ is a stable set of $T$ of weight $\alpha_{AB}$, and hence $(S \setminus (A \cup B)) \cup \{a, b\}$ is a stable set of $T_Y$ of the same weight as $S$.

Therefore $\alpha(T) \leq \alpha(T_Y)$. The reverse inequalities can be shown similarly, and hence the result holds. ∎

**Proof of Lemma 6** First let $S$ be a maximum weighted stable set of $T$. If $S \cap \{a,b\} = \{a\}$ then let $S' = S \cup \{a'\}$, if $S \cap \{a,b\} = \{a,b\}$ then let $S' = (S \setminus \{a\}) \cup \{a'\}$, and otherwise let $S' = S$. Then $S'$ is a stable set of $T'$ of the same weight as the weight of $S$ in $T$, and hence $\alpha(T) \leq \alpha(T')$. Now let $S$ be a maximum weighted stable set of $T'$. Note that we may assume w.l.o.g. that $S \cap \{a,a',b\} = \emptyset, \{a,a'\}, \{b\}$ or $\{a',b\}$. If $S \cap \{a,a',b\} = \{a,a'\}$ then let $S' = S \setminus \{a'\}$, if $S \cap \{a,a',b\} = \{a',b\}$ then let $S' = (S \setminus \{a'\}) \cup \{a\}$, and otherwise let $S' = S$. Then $S'$ is a stable set of $T$ of the same weight as the weight of $S$ in $T'$, and hence $\alpha(T') \leq \alpha(T)$, completing the proof. ∎

**Proof of Lemma 7** Consider sets $X, K_1, \dots, K_t$ that partition $V(\overline{T})$ as in the definition of $\mathcal{T}_1$. A maximal stable set in $T$ is formed by a subset $S$ of size at most 2 of $X$ together with all the non-neighbors of $S$ in some $K_i$. Therefore, there are at most $n^3$ maximal stable sets in $T$. ∎

**Proof of Lemma 8** Let $G$ be the realization of $T$ obtained by transforming every switchable pair of $T$ by a non-edge. Note that a subset of $V(T) = V(G)$ is a stable set in $G$ if and only if it is a stable set in $T$. So, the problem of enumerating all maximal stable sets of $G$ is equivalent to the problem of enumerating all maximal stable sets of $T$. Note also that if $S$ is a stable set of $T$ and $S' \subseteq S$, then $w(S') \leq w(S)$.

The algorithm uses Theorem 4 to enumerates all maximal stable sets of $T$ (but stops if more than $n^3$ sets are found). Lemma 7 certifies that if more than $n^3$ sets are found, then $T$ is not in $\overline{\mathcal{T}_1}$. Otherwise, among all enumerated stable sets, the algorithm outputs one of maximum weight. ∎

**Proof of Lemma 9** Let $G$ be the realization of $T$ in which all switchable pairs are assigned value "strong antiedge". Note that $G$ is a graph. We claim that testing whether O-output $i$ is true or not can be done in polynomial time for $i = 1, 2$. For $i = 1$, this is trivial and for $i = 2$, it follows from Theorem 4 applied to $G$. The algorithm does these two tests, stops if one of them is a success, and if each attempt fails, it gives the answer 3. The running time is clearly $O(n^4 m)$. It remains to check that when output 3 is the answer it is a true statement. So suppose for a contradiction that $\alpha(T) < W$. In particular, $W \geq 2$.

If $T$ is a trigraph in $\mathcal{T}_0$, then $n \leq 8 = f(2) \leq f(W)$, so the algorithm should have stopped to give outcome 1, a contradiction. If $T$ is a trigraph in $\overline{\mathcal{T}_1}$, then by Lemma 7, the number of maximal stable sets in $T$ is at most $n^3$. So, the algorithm should have stopped to give outcome 2, a contradiction.

So, suppose that $T$ is a trigraph in $\mathcal{T}_1$, and consider the sets $X, K_1, \dots, K_t$ as in the definition of $\mathcal{T}_1$. If $|X| \geq \binom{W+1}{2}$, then by Ramsey Theorem, $G$ contains a stable set of size at least $W$, and therefore $T$ contains a stable set of weight at least $W$ (since weights of vertices are at least 1 and weights of switchable pairs are at least 2), a contradiction. So, $|X| \leq g(W)$. If $t \geq W$, then by taking a vertex in each $K_i$, $i = 1, \dots, t$, we obtain a stable set of size at least $W$, a contradiction. So $t \leq W - 1$.

If for some $i \in \{1, \ldots, t\}$ we have $|K_i| \geq \binom{g(W)}{2} + 2g(W) + 2$, then since $T$ is monogamous and $|X| \leq g(W)$, at least $\binom{g(W)}{2} + g(W) + 2$ vertices in $K_i$ are not adjacent to any switchable pair and we call $K'_i$ the set formed by these vertices (so, $|K'_i| \geq \binom{g(W)}{2} + g(W) + 2$). Consider the hypergraph $N$ with vertex set $X$ and hyperedge set $\{N(v) \cap X | v \in K'_i\}$ and observe that $N$ has Vapnik-Cervonenkis dimension bounded by 2 (for an introduction to Vapnik-Cervonenkis dimension, see [1]). Indeed, assume for contradiction that $S = \{x_1, x_2, x_3\}$ is a shattered subset of (three) vertices of $N$, i.e. for every subset $Y$ of $S$ there exists a hyperedge $e$ of $N$ such that $S \cap e = Y$. This would imply the existence of three vertices $y_1, y_2, y_3$ in $K'_i$ such that $y_i$ is joined only to $x_i$ in $S$, for $i = 1, 2, 3$. Since $X$ is triangle-free, there exists an antiedge in $S$, say $x_1x_2$. But then a contradiction appears since $\{y_1, y_2, y_3, x_1, x_2\}$ induces a bull. Since the VC-dimension is at most 2, by Sauer's Lemma [19], the number of distinct hyperedges of $N$ is at most $\binom{|X|}{2} + |X| + 1$, so at most $\binom{g(W)}{2} + g(W) + 1$. But since two distinct vertices of $K'_i$ have distinct neighborhoods to avoid homogeneous sets, it follows that $K'_i$ has size bounded by $\binom{g(W)}{2} + g(W) + 1$, a contradiction. So, $|K_i| \leq \binom{g(W)}{2} + 2g(W) + 1$.

We proved that $|X| \leq g(W)$, $t \leq W - 1$ and for $i \in \{1, \ldots, t\}$, $|K_i| \leq \binom{g(W)}{2} + 2g(W) + 1$. It follows that

$$n \leq g(W) + (W - 1)\left(\binom{g(W)}{2} + 2g(W) + 1\right) = f(W).$$

So, the algorithm should have stopped to give outcome 1, a contradiction. ∎

**Proof of Theorem 5** First, we delete all vertices of weight 0, and for all switchable pairs of weight 1, we replace the switchable pair by a strong edge. It is easy to check that this does not change $\alpha$. Now, all vertices have weight at least 1, and all switchable pairs have weight at least 2. Apply the algorithm from Theorem 3.

Suppose that no decomposition is found. In particular, $T$ has no homogeneous set. Also by Theorem 1, $T$ is basic. Run the algorithm from Lemma 9. If outcome 1 is the answer, we compute by brute force a maximum weighted stable set in time $2^{O(W^5)}$. If outcome 2 is the answer, we compute a maximum weighted stable set in polynomial time by Theorem 4 applied to the realization of $T$ in which all switchable pairs are assigned value "strong antiedge". In both cases, we know the answer. Finally, if outcome 3 is the answer, then we have that $\alpha(T) \geq W$ and we output "yes".

Suppose that a decomposition $(X, Y)$ is found. By Theorem 2, $T_X$ is basic. We run the algorithm from Lemma 9 for $T_X$. If outcome 3 is the answer, output $\alpha(T) \geq W$, which is the right answer since $\alpha(T[X]) \leq \alpha(T)$. If outcome 1 or 2 is the answer, then compute a maximum weighted stable set in $T[X]$ as above (if $X = A \cup B$ where $(A, B)$ is a homogeneous pair, then we also compute a maximum weighted stable set in $T[A]$ and $T[B]$ that are basic). We now have the weights needed to construct the block $T_Y$. Run the algorithm recursively for

$T_Y$ (this is correct by Lemma 5). Since $T_Y$ has fewer vertices than $T$, the number of recursive calls is bounded by $n$. ∎

**Proof of Theorem 6** This simply follows from the facts that the unparameterized version of $\alpha(G) \geq k$ is NP-hard for bull-free graphs, and that the problem is OR-compositional (see [2]). Indeed, if we are given a family $G_1, \dots, G_\ell$ of bull-free graphs and some integer $k$, one can form the complete sum $G$ of these graphs by taking disjoint copies of them and joining them pairwise by complete bipartite graphs (i.e. for all $i \neq j$, put all edges between $G_i$ and $G_j$). We then have that $G$ is bull-free, and moreover $\alpha(G) \geq k$ if and only if there exists some $i$ for which $\alpha(G_i) \geq k$ (this is the definition of an OR-compositional problem). By a result of Bodlaender et al. [2], unless NP $\subseteq$ coNP/poly, no NP-hard OR-compositional problem can admit a polynomial kernel. ∎

**Proof of Theorem 7** The proof is done already for weighted trigraphs. For weighted graphs, there is a problem: with the present proof, we reduce graphs to trigraphs, so we need to interpret a trigraph as a graph. It is not the case that every (integer) weigthed bull-free trigraph can be interpreted as an unweighted bull-free graph with the same $\alpha$. Indeed, it is false in general that for every switchable pair $ab$ of a bull-free trigraph, at least one of the trigraph $T_{a \to S}$, $T_{b \to S}$, $T_{a \to K}$ or $T_{b \to K}$ is bull-free. In Fig. 3, we show an example of a bull-free trigraph with a switchable pair represented by a dashed line, where all the four obtained graphs contain a bull. However, if we start with a bull-free *graph* and compute leaves of the decomposition tree, every switchable pair in them is obtained at some point by shrinking a homogeneous pair $(A, B)$ of a trigraph $T$ into a switchable pair $ab$ of a trigraph $T'$. Because of the requirement that $A$ is not strongly complete and not strongly anticomplete to $B$, we see that at least one of $T'_{a \to S}$, $T'_{b \to S}$, $T'_{a \to K}$ or $T'_{b \to K}$ is in fact an induced subtrigraph of some semirealization of $T$ (and recall that a trigraph is bull-free if and only if all its semirealizations are bull-free). By Lemma 6, this allows us to represent the weighted bull-free trigraphs generated by our Turing-kernel as bull-free graphs (with the same $\alpha$).

To prove the unweighted versions, just note that we can get rid of weights by substituting a (strong) stable set on $w$ vertices for every vertex of weight $w$. ∎

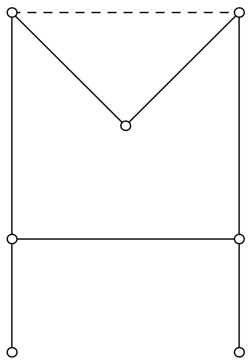

**Fig. 3.** A bull-free trigraph where all ways to expand a switchable pair creates a bull.