Settling the Relationship Between Wilber's Bounds for Dynamic Optimality

Victor Lecomte
Columbia University, New York, NY, USA
vl2414@columbia.edu

Omri Weinstein
Columbia University, New York, NY, USA
omri@cs.columbia.edu

Abstract

In FOCS 1986, Wilber proposed two combinatorial lower bounds on the operational cost of any binary search tree (BST) for a given access sequence $X \in [n]^m$. Both bounds play a central role in the ongoing pursuit of the *dynamic optimality conjecture* (Sleator and Tarjan, 1985), but their relationship remained unknown for more than three decades. We show that Wilber's *Funnel bound* dominates his *Alternation bound* for all X, and give a tight $\Theta(\lg \lg n)$ separation for some X, answering Wilber's conjecture and an open problem of Iacono, Demaine et. al. The main ingredient of the proof is a new *symmetric* characterization of Wilber's Funnel bound, which proves that it is invariant under *rotations* of X. We use this characterization to provide initial indication that the Funnel bound matches the Independent Rectangle bound (Demaine et al., 2009), by proving that when the Funnel bound is constant, $\mathsf{IRB}_{\boxslash}$ is linear. To the best of our knowledge, our results provide the first progress on Wilber's conjecture that the Funnel bound is dynamically optimal (1986).

2012 ACM Subject Classification Theory of computation → Data structures design and analysis

Keywords and phrases data structures, binary search trees, dynamic optimality, lower bounds

Digital Object Identifier 10.4230/LIPIcs.ESA.2020.68

Related Version A full version of the paper is available on arXiv [10] at https://arxiv.org/abs/1912.02858. This conference version doesn't contain the last section, which relates the Funnel bound to the Independent Rectangle bound. The versions are otherwise identical.

Funding *Victor Lecomte*: Research supported by a fellowship of the Belgian American Educational Foundation.
Omri Weinstein: Research supported by NSF CAREER award CCF-1844887.

Acknowledgements We want to thank the anonymous reviewers for their enthusiastic feedback and the numerous typos they spotted.

1 Introduction

The *dynamic optimality conjecture* of Sleator and Tarjan [14] postulates the existence of an *instance optimal* binary search tree algorithm (BST), namely, an online self-adjusting BST whose running time[1] matches the best possible running time *in hindsight* for any fixed sequence of queries. More formally, letting $\mathcal{T}(X)$ denote the operational time of a BST algorithm $\mathcal{T}$ on a sequence $X = (x_1, \ldots, x_m) \in [n]^m$ of keys to be searched, the conjecture says that there is an online BST $\mathcal{T}$ such that $\forall X$, $\mathcal{T}(X) \leq O(\mathsf{OPT}(X))$, where $\mathsf{OPT}(X) := \min_{\mathcal{T}'} \mathcal{T}'(X)$ denotes the optimal offline cost for X. Such instance optimal algorithms are generally impossible, as an offline algorithm that sees the input X in advance

[1] i.e. the number of pointer movements and tree rotations performed by the BST

28th Annual European Symposium on Algorithms (ESA 2020).
Editors: Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders; Article No. 68; pp. 68:1–68:21
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

can simply "store the answers" and output them in $O(1)$ per operation, which is why worst-case analysis is the typical benchmark for online algorithms. Nevertheless, in the BST model, where the competing class of algorithms are self-adjusting binary search trees, instance optimality is an intriguing possibility. After 35 years of active research, two BST algorithms are still conjectured to be constant-competitive: The first one is the celebrated *splay tree* of [14], the second one is the more recent *GreedyFuture* algorithm [12, 5, 13]. However, optimality of both splay trees and GreedyFuture was proven only in special cases, and they are not known to be $o(\lg n)$-competitive for general access sequences X (note that every balanced BST is trivially $O(\lg n)$-competitive). The best provable result to date on the algorithmic side is an $O(\lg\lg n)$-competitive BST, the *Tango Tree* ([5] and its subsequent variants [15, 2]).

The ongoing pursuit of dynamically-optimal BSTs motivated the development of lower bounds on the cost of the offline solution $\mathsf{OPT}(X)$, attempting to capture the "correct" complexity measure of a fixed access sequence X in the BST model, and thereby providing a concrete benchmark for competitive analysis. Indeed, one defining feature of the dynamic optimality problem (and the reason why it is a viable possibility) is the existence of nontrivial lower bounds on $\mathsf{OPT}(X)$ for individual *fixed* access sequences X, as opposed to distributional lower bounds. [2] These lower bounds are all derived from a natural geometric interpretation of the access sequence $X = x_1, \ldots, x_m$ as a point set on the plane, mapping the i^{th} access x_i to point (x_i, i) ([5, 8], see Figure 1). The earliest lower bounds on $\mathsf{OPT}(X)$ were proposed in an influential paper of Wilber [16], and are the main subject of this paper.

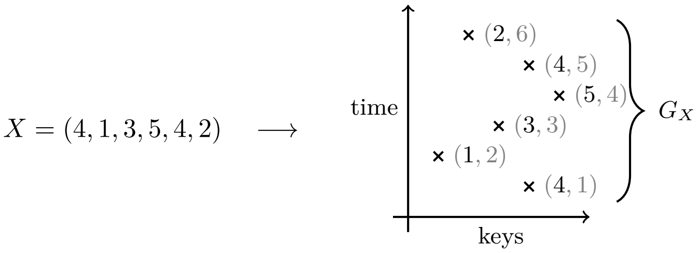


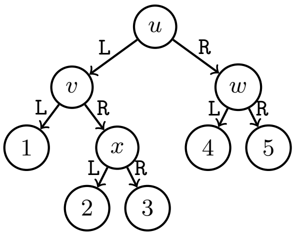
Figure 1 Transforming X into its geometric view G_X.

The Alternation bound

Wilber's first lower bound, the *Alternation bound* $\mathsf{Alt}_{\mathcal{T}}(X)$, counts the total number of left/right alternations obtained by searching the keys $X = (x_1, \ldots, x_m)$ on a *fixed* (static) binary search tree $\mathcal{T}$, where alternations are summed up over all nodes $v \in \mathcal{T}$ of the "reference tree" $\mathcal{T}$ (see Figure 2 and the formal definition in Section 2). Thus, the Alternation bound is actually a family of lower bounds, optimized by the choice of the reference tree $\mathcal{T}$, and we henceforth define $\mathsf{Alt}(X) := \max_{\mathcal{T}} \mathsf{Alt}_{\mathcal{T}}(X)$. This lower bound played a key role in the design and analysis of Tango trees and their variants [6, 15], whose operational cost is in fact shown to be $O(\lg\lg n) \cdot \mathsf{Alt}_{\mathcal{T}}(X) \leq O(\lg\lg n) \cdot \mathsf{OPT}(X)$ (when setting the reference tree $\mathcal{T}$ to be the canonical balanced BST on $[n]$). Unfortunately, this bound is not tight, as we show that there are access sequences $\tilde{X}$ for which $\mathsf{Alt}_{\mathcal{T}}(\tilde{X}) \leq O(\mathsf{OPT}(\tilde{X})/\lg\lg n)$ *simultaneously* for all

[2] For example, Wilber's Alternation bound can be used to show that the "bit-reversal" access sequence obtained by reversing the binary representation of the monotone sequence $\{1, 2, 3, \ldots, n\}$ has cost $\Omega(\lg n)$ per operation [16].

choices of reference trees $\mathcal{T}$ (previously, this was known only for any *fixed* $\mathcal{T}$ [8]), and hence the combined bound $\mathsf{Alt}(X)$ does not capture dynamic optimality in general. Nevertheless, the algorithmic interpretation of the Alternation bound is an interesting proof-of-concept of how lower bounds can lead to new and interesting online BST algorithms.



reference tree $\mathcal{T}$

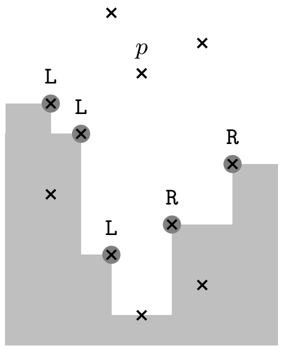
Node	Link used by each access	Group by letter	#
u	R, L, L, R, R, L	[R], [L, L], [R, R], [L]	4
v	L, R, R	[L], [R, R]	2
w	L, R, L	[L], [R], [L]	3
x	R, L	[R], [L]	2
Total			11

Figure 2 For access sequence $X = (4, 1, 3, 5, 4, 2)$ and reference tree $\mathcal{T}$, $\mathsf{Alt}_{\mathcal{T}}(X) = 11$.

The Funnel bound

The definition of Wilber's second bound, the *Funnel bound*, is less intuitive (and as such, was much less understood prior to this work). Let G_X be the set of m points in the plane given by the map $x_i \mapsto (x_i, i)$. The *funnel* of a point $p \in G_X$ is the set of "orthogonally visible" points below p, i.e. points q such that the axis-aligned rectangle with corners at p and q contains no other points (see Figure 3). For each p, look at the points in the funnel of p sorted by y coordinate, and count the number of alternations from the left to the right of P that occur. Call this $f(p)$; this is p's contribution to the lower bound. Summing this value for all $p \in G_X$ gives the lower bound $\mathsf{Funnel}(X) := \sum_{p \in G_X} f(p)$. An algorithmic view of this bound is as follows: consider the algorithm that simply brings each x_i to the root by a series of single rotations. Then $f(p)$ for $p = (x_i, i)$ is exactly the number of *turns* on the path from the root to x_i right before it is accessed [1, 8]. This view emphasizes the amortized nature of the funnel bound: at any point, there could be linearly many keys in the tree that are only *one* turn away from the root, so one can only hope to achieve this bound in some amortized fashion. This partially explains why Wilber's second bound has been so elusive to analyze (more on this interpretation can be found in the recent work of [11]).

Wilber conjectured that $\mathsf{Funnel}(X) \geq \Omega(\mathsf{Alt}(X))$ for every access sequence X, and that the Funnel bound is in fact *dynamically optimal*, i.e., that $\mathsf{Funnel}(X) = \Theta(\mathsf{OPT}(X))\ \forall X$. These conjectures were echoed multiple times in the long line of research spanning dynamic optimality (see e.g., [5, 8, 4, 9]). Very recently, Levy and Tarjan [11] gave a compelling intuitive explanation for why $\mathsf{Funnel}(X)$ is related to the amortized analysis of splay trees (see Section 4). Despite all this, the Funnel bound remained elusive and no progress was made on Wilber's conjectures for nearly 40 years (To the best of our knowledge, the only properties that were previously known about the Funnel bound is that it is optimal in the "key-independent" setting [7] and "approximately monotone" [11], both are prerequisites for dynamic optimality.)



the funnel of p has 5 points (highlighted)
Sorted by increasing y-coordinate: L, R, R, L, L.
This forms 3 groups [L], [R, R], [L, L], so $f(p) = 3$.

Figure 3 Computing $f(p)$ for $p = (4, 9)$ in the geometric view of $X = (4, 6, 3, 5, 1, 7, 2, 4, 6, 3)$. Notice how the funnel points form a staircase-like front on either side of p.

Our main contribution affirmatively answers Wilber's first question, and settles the relationship between the Alternation bound and the Funnel bound:

▶ **Theorem 1** (Funnel dominates Alt). *For every access sequence X without repeats*[3] *and for every tree $\mathcal{T}$,* $\mathsf{Alt}_{\mathcal{T}}(X) \leq O(\mathsf{Funnel}(X) + m)$.

▶ **Theorem 2** (Tight separation). *There is an access sequence $\tilde{X}$ for which* $\mathsf{Funnel}(\tilde{X}) \geq \Omega(\lg\lg n) \cdot (\mathsf{Alt}_{\mathcal{T}}(\tilde{X}) + m)$ *simultaneously for all trees $\mathcal{T}$.*

The latter separation is tight up to constant factors, since Tango trees imply that $\mathsf{OPT}(X) \leq O(\lg\lg n) \cdot \mathsf{Alt}(X)$. An interesting corollary of Theorem 2 is that the analysis of Tango trees cannot be improved by choosing *any* reference tree, answering an open question of Iacono [8]. (One attractive idea is to choose a *random* reference tree instead of the canonical balanced BST, but Theorem 2 shows that this will not help in general.)

A symmetric characterization of the Funnel bound

The geometric equivalence of dynamic optimality (through "arborally satisfied" rectangles [5]) makes it clear that $\mathsf{OPT}(X)$ is *invariant* under geometric transformations of the access sequence X. Indeed, a fundamental barrier in understanding the Funnel bound and its claim to optimality is that it was unclear whether Wilber's bounds were invariant under *rotations* of the access sequence X. Demaine et al. explicitly pointed out this challenge:

> "It is also unclear how [Wilber's] bounds are affected by 90-degree rotations of the point set representing the access sequence and, for the Funnel bound, by *flips*. Computer search reveals many examples where the bounds change slightly, and proving that they change by only a constant factor seems daunting." [5]

This shows that *exact* symmetry of $\mathsf{Funnel}(X)$ is hopeless, and can only hold in some "amortized" sense. Indeed, the heart of our paper, which is also a key ingredient in the proof of Theorem 1, is a new *symmetric* characterization of the Funnel bound, which proves

[3] As explained at the beginning of Section 2, it is fine for our purposes to focus on access sequences where each value appears only once.

that, up to a $\pm O(m)$ additive term, it is indeed invariant to rotations. More formally, we show that for any access sequence X, $\mathsf{Funnel}(X)$ is asymptotically equal to the number of occurrences in G_X of a configuration of 4 points that we call a **z-rectangle**[4] (see Figure 4).

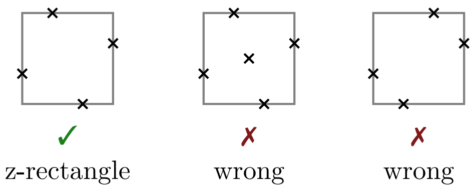


Figure 4 A z-rectangle is a configuration of 4 points. Its interior must be empty, and the relative order of the four points matters.

A crucial difference between z-rectangles and the notion of *independent rectangles* [5] is that the latter have to satisfy additional *independence* constaints across several rectangles, whereas z-rectangles have no “global” constraints whatsoever. In other words, z-rectangles are a *local* feature of the access sequence, in the sense that their existence and contribution to the lower bound are unaffected by other z-rectangles and by points outside of it. We believe this key property will make the analysis of online BST algorithms more tractable, as it gives a simpler competitive benchmark. We next describe an initial step in this direction.

Towards dynamic optimality of the Funnel Bound

One consequence of the simplicity of the z-rectangle characterization of the Funnel bound is that it makes it easier to compare it both to other BST lower bounds and to candidate algorithms for dynamic optimality. As a proof of concept, we show that when there is only a constant number of z-rectangle in G_X, then $\mathsf{IRB}_{\boxslash}(X)$ is linear, where $\mathsf{IRB}_{\boxslash}$ is one of the terms in the *Independent Rectangle* bound $\mathsf{IRB}(X) := \mathsf{IRB}_{\boxslash}(X) + \mathsf{IRB}_{\boxbslash}(X)$, which is known to dominate both of Wilber’s bounds [5] (we define $\mathsf{IRB}_{\boxslash}(X)$ in the last section of the full version [10]). More formally,

▶ **Theorem 3.** *If G_X contains $O(1)$ z-rectangles, then* $\mathsf{IRB}_{\boxslash}(X) \leq O(m)$.

We remark that the proof of this theorem already introduces a nontrivial charging argument that could (hopefully) be generalized to prove that Funnel matches IRB, as conjectured by previous works [8].

Techniques

At a very high level, the main ideas in Theorem 1 are to use the self-reducible structure of the Alternation bound, and to show that interleaving two access sequences $X_{\mathtt{L}}$ and $X_{\mathtt{R}}$ on *disjoint ranges* is a super-additive operation, i.e., it increases the overall value of $\mathsf{Funnel}(X)$ to more than the sum of its parts $\mathsf{Funnel}(X_{\mathtt{L}}) + \mathsf{Funnel}(X_{\mathtt{R}})$. This argument involves both X and its *reverse* (flip), hence our new symmetric characterization of the Funnel bound (through z-rectangles) is key to the proof. The main idea behind Theorem 2 is to form hard sequences over geometrically-spaced sets of keys $\{i+1, i+2, i+4, i+8, \ldots\}$, each of which can “force” $\mathsf{Alt}_{\mathcal{T}}$ to pick a very lopsided reference tree $\mathcal{T}$. Those sequences can then be

[4] We thank an anonymous reviewer for informing us that z-rectangles have been discussed in the past under the name “pinwheel configuration”, though (to the best of their knowledge) never in writing.

concatenated together so that the average value of $\mathsf{Alt}_{\mathcal{T}}$ is provably low whichever $\mathcal{T}$ was picked. Finally, the key idea in Theorem 3 is to study the consequences of the *absence* of z-rectangles on the combinatorial structure of point set G_X, and use this to bound the value of $\mathsf{IRB}_{\boxslash}(X)$ by a charging argument.

Remark on independent work

In a concurrent and independent work, Chalermsook, Chuzhoy and Saranurak [3] obtain a (weaker) $\Theta(\lg\lg n/\lg\lg\lg n)$ separation between Alt and Funnel, in the same spirit as the tight separation we give in Theorem 2. Our works are otherwise unrelated.

2 Preliminaries

To make our definitions and proofs easier, we will work directly in the geometric representation of access sequences as (finite) sets of points in the plane $\mathbb{R}^2$.

▶ **Definition 4** (geometric view). *Any access sequence $X = (x_1, \ldots, x_m) \in [n]^m$ can be represented as the set of points $G_X = \{(x_i, i) \mid i \in [n]\}$, where the x-axis represents the key and the y-axis represents time (see Figure 1).*

By construction, in G_X, no two points share the same y-coordinate. We will say such a set has "distinct y-coordinates". In addition, we note that it is fine to restrict our attention to sequences X without repeated values.[5] The geometric view G_X of such sequences also has no two points with the same x-coordinate. We will say that such a set has "distinct x- and y-coordinates".

▶ **Definition 5** (x- and y-coordinates). *For a point $p \in \mathbb{R}^2$, we will denote its x- and y-coordinates as $p.x$ and $p.y$. Similarly, we define $P.x = \{p.x \mid p \in P\}$ and $P.y = \{p.y \mid p \in P\}$.*

We start by defining the *mixing value* of two sets: a notion of how much two sets of numbers are interleaved. It will be useful in defining both the Alternation bound and the Funnel bound. We define it in a few steps.

▶ **Definition 6** (mixing string). *Given two disjoint finite sets of real numbers L, R, let $\mathrm{mix}(L, R)$ be the string in $\{\mathtt{L}, \mathtt{R}\}^*$ that is obtained by taking the union $L \cup R$ in increasing order and replacing each element from L by $\mathtt{L}$ and each element from R by $\mathtt{R}$. For example, $\mathrm{mix}(\{2,3,8\}, \{1,5\}) = \mathtt{RLLRL}$.*

▶ **Definition 7** (number of blocks). *Given a string $s \in \{\mathtt{L}, \mathtt{R}\}^*$, we define $\mathrm{blocks}(s)$ as the number of contiguous blocks of the same symbol in s. Formally,*

$$\mathrm{blocks}(s) := \begin{cases} 0 & \text{if } s \text{ is empty} \\ 1 + \#\{i \mid s_i \neq s_{i+1}\} & \text{otherwise.} \end{cases}$$

For example, $\mathrm{blocks}(\mathtt{LLLRLL}) = 3$. Note that if we insert characters into s, $\mathrm{blocks}(s)$ can only increase.

▶ **Definition 8** (mixing value). *Let $\mathrm{mixValue}(L, R) := \mathrm{blocks}(\mathrm{mix}(L, R))$ (see Figure 5).*

The mixing value has some convenient properties, which we will use later:

[5] Indeed, Appendix E in [4] gives a simple operation that transforms any sequence X into a sequence $\mathrm{split}(X)$ without repeats such that $\mathsf{OPT}(\mathrm{split}(X)) = \Theta(\mathsf{OPT}(X))$. Thus if we found a tight lower bound $L(X)$ for sequences without repeats, a tight lower bound for general X could be obtained as $L(\mathrm{split}(X))$.

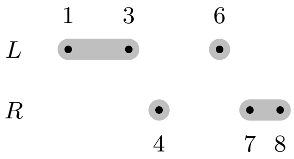


Figure 5 A visualization of $\text{mixValue}(\{1,3,6\},\{4,7,8\}) = 4$.

▶ **Fact 9** (properties of mixValue). *Function* $\text{mixValue}(L,R)$ *is:*

(a) *symmetric:* $\text{mixValue}(L,R) = \text{mixValue}(R,L)$*;*

(b) *monotone: if* $L_1 \subseteq L_2$ *and* $R_1 \subseteq R_2$*, then* $\text{mixValue}(L_1,R_1) \le \text{mixValue}(L_2,R_2)$*;*

(c) *subadditive under concatenation: if* $L_1, R_1 \subseteq (-\infty, x]$ *and* $L_2, R_2 \subseteq [x, +\infty)$*, then* $\text{mixValue}(L_1 \cup L_2, R_1 \cup R_2) \le \text{mixValue}(L_1,R_1) + \text{mixValue}(L_2,R_2)$*.*

Finally, $\text{mixValue}(L,R) \le 2 \cdot \min(|L|,|R|) + 1$*.*

We now give precise definitions of Wilber's two bounds.[6]

▶ **Definition 10** (Alternation bound). *Let P be a point set with distinct y-coordinates, and let $\mathcal{T}$ be a binary tree in which leaves are labeled with elements of $P.x$ in increasing order, and each non-leaf node has two children.*

We define $\mathsf{Alt}_{\mathcal{T}}(P)$ *using the recursive structure of $\mathcal{T}$. If $\mathcal{T}$ is a single node, let* $\mathsf{Alt}_{\mathcal{T}}(P) := 0$*. Otherwise, let $\mathcal{T}_{\mathtt{L}}$ and $\mathcal{T}_{\mathtt{R}}$ be the left and right subtrees at the root. Partition P into two sets* $P_{\mathtt{L}} := \{p \in P \mid p.x \in \mathcal{T}_{\mathtt{L}}\}$ *and* $P_{\mathtt{R}} := \{p \in P \mid p.x \in \mathcal{T}_{\mathtt{R}}\}$*. Define quantity*

$$a(P,\mathcal{T}) := \text{mixValue}(P_{\mathtt{L}}.y, P_{\mathtt{R}}.y),$$

which describes how much $P_{\mathtt{L}}$ and $P_{\mathtt{R}}$ are interleaved in time. Then

$$\mathsf{Alt}_{\mathcal{T}}(P) := a(P,\mathcal{T}) + \mathsf{Alt}_{\mathcal{T}_{\mathtt{L}}}(P_{\mathtt{L}}) + \mathsf{Alt}_{\mathcal{T}_{\mathtt{R}}}(P_{\mathtt{R}}). \tag{1}$$

In addition, for an access sequence X, let $\mathsf{Alt}_{\mathcal{T}}(X) := \mathsf{Alt}_{\mathcal{T}}(G_X)$*.*

▶ **Definition 11** (axis-aligned rectangle delimited two points). *Given two points p and q with distinct x- and y- coordinates, let $\square pq$ be the smallest axis-aligned rectangle that contains both p and q. Formally,*

$$\square pq := [\min(p.x,q.x), \max(p.x,q.x)] \times [\min(p.y,q.y), \max(p.y,q.y)].$$

▶ **Definition 12** (empty rectangles). *Let P be a point set. Given $p, q \in P$, we say $\square pq$ is empty*[7] *in P if $P \cap \square pq = \{p,q\}$ (see Figure 6).*

For the next definitions, it is helpful to refer back to Figure 3. In particular, $F_{\mathtt{L}}(P,p)$ and $F_{\mathtt{R}}(P,p)$ (the left and right funnel) correspond to the points marked with `L` and `R`.

▶ **Definition 13** (left and right funnel). *Let P be a point set. For each $p \in P$, we say that access $q \in P$ is in the left (resp. right) funnel of p within P if q is to the lower left (resp. lower right) of p and $\square pq$ is empty. Formally, let*

$$F_{\mathtt{L}}(P,p) := \{q \in P \mid q.y < p.y \wedge q.x < p.x \wedge P \cap \square pq = \{p,q\}\}$$

[6] These definitions may differ by a constant factor or an additive $\pm O(m)$ from the definitions the reader has seen before. We will ignore such differences, because the cost of a BST also varies by $\pm O(m)$ depending on the definition, and the interesting regime is when $\mathsf{OPT}(X) = \omega(m)$.

[7] This corresponds to the notion of "unsatisfied rectangle" in [5].

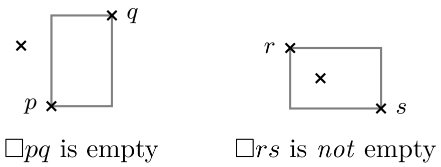


Figure 6 Some axis-aligned rectangles.

and

$$F_{\mathtt{R}}(P,p) := \{q \in P \mid q.y < p.y \wedge q.x > p.x \wedge P \cap \square pq = \{p,q\}\}.$$

We will collectively call $F_{\mathtt{L}}(P,p) \cup F_{\mathtt{R}}(P,p)$ the funnel of p within P.

▶ **Definition 14** (Funnel bound). *Let P be a point set with distinct y-coordinates. For each $p \in P$, define quantity*

$$f(P,p) := \mathrm{mixValue}(F_{\mathtt{L}}(P,p).y, F_{\mathtt{R}}(P,p).y),$$

which describes how much the left and right funnel of p are interleaved in time. Then

$$\mathsf{Funnel}(P) := \sum_{p \in P} f(P,p).$$

In addition, for an access sequence X, let $\mathsf{Funnel}(X) := \mathsf{Funnel}(G_X)$.

3 The Funnel bound dominates the Alternation bound

We prove that Funnel dominates Alt in two parts: in Section 3.1 we show that $\mathsf{Alt}(X)$ is dominated by the sum $\mathsf{Funnel}(X) + \mathsf{Funnel}(\overline{X})$, where $\overline{X}$ is the reverse of X, then in Section 3.2 we prove that $\mathsf{Funnel}(\overline{X}) \approx \mathsf{Funnel}(X)$ using our new characterization of Funnel by z-rectangles.

3.1 Upper-bounding the Alternation bound by a sum of two Funnel bounds

▶ **Definition 15** (time reversal). *The time reversal of a point $p \in \mathbb{R}^2$ is $\overline{p} := (p.x, -p.y)$.*[8] *The time reversal of a point set P is $\overline{P} := \{\overline{p} \mid p \in P\}$ (see Figure 7).*

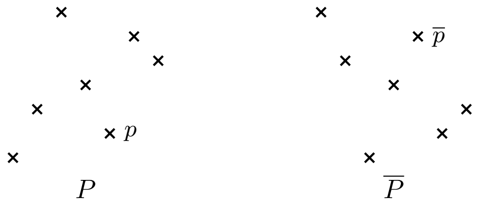


Figure 7 A point set and its time reversal.

We first prove the following lemma.

[8] The notation is inspired from the notion of complex conjugate, which is also a vertical flip.

▶ **Lemma 16.** *Let P be a point set with distinct y-coordinates, and let $\mathcal{T}$ be a tree that satisfies the conditions of Definition 10. Then* $\mathsf{Funnel}(P) + \mathsf{Funnel}(\overline{P}) \geq \mathsf{Alt}_{\mathcal{T}}(P)$.

Even though the formal proof of this lemma is a relatively involved case analysis, it is easy to understand geometrically. The key observation is the following. Consider two sequences $X_\mathtt{L}$ and $X_\mathtt{R}$ on disjoint ranges, and interleave to form a single sequence X. Then the more times we switch from elements of $X_\mathtt{L}$ to elements of $X_\mathtt{R}$, the bigger $\mathsf{Funnel}(X) + \mathsf{Funnel}(\overline{X})$ is going to be.

To see this, let's look at the geometric view of X (see Figure 8). Let p and q be two consecutive points on the $X_\mathtt{L}$ side that are separated by a streak of points from $X_\mathtt{R}$ (i.e. all accesses between p and q vertically are from $X_\mathtt{R}$). First, assume $p.x > q.x$. Then q is in the left funnel of p, and at least of the points on the $X_\mathtt{R}$ between p and q must be in the right funnel of p, which forms a completely new group of funnel points compared to what p had in $X_\mathtt{L}$. This means that the contribution of p to $\mathsf{Funnel}(X)$ is at least one higher than its contribution to $\mathsf{Funnel}(X_\mathtt{L})$.

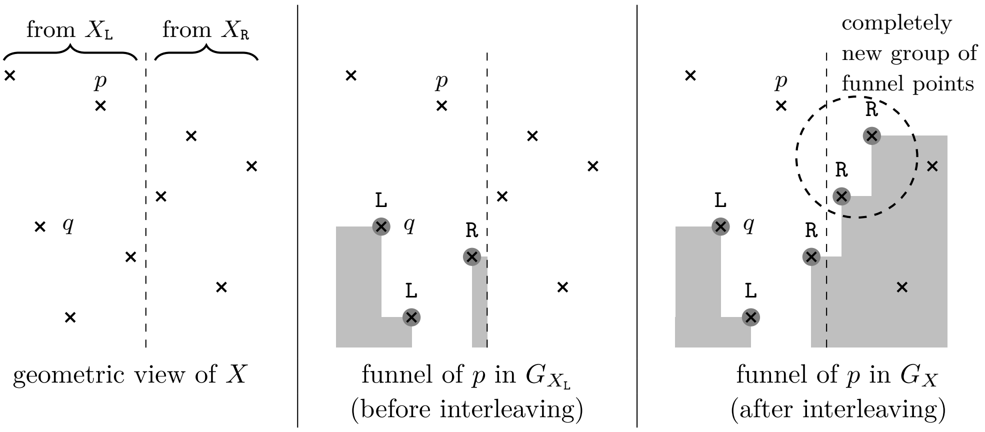


Figure 8 Interleaving sequences $X_\mathtt{L} = (3, 5, 2, 4, 1)$ and $X_\mathtt{R} = (8, 6, 9, 7)$ into $X = (3, 8, 5, 2, 6, 9, 7, 4, 1)$. The contribution of p to $\mathsf{Funnel}(X_\mathtt{L})$ is 3, while the contribution of p to $\mathsf{Funnel}(X)$ is 4.

What if $p.x < q.x$ instead? Then it turns out that an analogous argument can be made on q if we take the time reversal of X. That is, the contribution of $\overline{q}$ to $\mathsf{Funnel}(\overline{X})$ is at least one higher than its contribution to $\mathsf{Funnel}(\overline{X_\mathtt{L}})$. Indeed, if we flip the point set vertically, then p and q exchange roles, which means $p.x > q.x$ once again.

To conclude, it remains to observe that the $a(P, p)$ term in the recursive definition of $\mathsf{Alt}_{\mathcal{T}}(X)$ is precisely a measure of how much the subsequences $X_\mathtt{L}$ and $X_\mathtt{R}$ corresponding to the left and right subtree at the root of $\mathcal{T}$ are interleaved. So we can apply the argument above by induction to show that $\mathsf{Funnel}(X) + \mathsf{Funnel}(\overline{X}) \geq \mathsf{Alt}_{\mathcal{T}}(X)$. We now reluctantly move to the formal proof.

Proof of Lemma 16. We prove this by induction on $\mathcal{T}$. The base case is $\mathcal{T}$ made of a single node. In this case, $\mathsf{Alt}_{\mathcal{T}}(P) = 0$ by definition, so the inequality trivially holds.

Now consider a general tree $\mathcal{T}$, and define $\mathcal{T}_\mathtt{L}$, $\mathcal{T}_\mathtt{R}$, $P_\mathtt{L}$ and $P_\mathtt{R}$ as in Definition 10. Note that each leaf of $\mathcal{T}$ has a label in $P.x$ and $\mathcal{T}_\mathtt{L}$ and $\mathcal{T}_\mathtt{R}$ must each have at least one leaf, so $P_\mathtt{L}$ and $P_\mathtt{R}$ are not empty. Let's apply the induction hypothesis on $(P_\mathtt{L}, \mathcal{T}_\mathtt{L})$ and $(P_\mathtt{R}, \mathcal{T}_\mathtt{R})$. This means that

$$\begin{aligned}&\mathsf{Funnel}(P_{\mathtt{L}}) + \mathsf{Funnel}(\overline{P_{\mathtt{L}}}) \geq \mathsf{Alt}_{\mathcal{T}_{\mathtt{L}}}(P_{\mathtt{L}})\\ &\mathsf{Funnel}(P_{\mathtt{R}}) + \mathsf{Funnel}(\overline{P_{\mathtt{R}}}) \geq \mathsf{Alt}_{\mathcal{T}_{\mathtt{R}}}(P_{\mathtt{R}}).\end{aligned}$$

Thus we find that

$$\begin{aligned}\mathsf{Alt}_{\mathcal{T}}(P) &= a(P,\mathcal{T}) + \mathsf{Alt}_{\mathcal{T}_{\mathtt{L}}}(P_{\mathtt{L}}) + \mathsf{Alt}_{\mathcal{T}_{\mathtt{R}}}(P_{\mathtt{R}}) && \text{(by definition)}\\ &\leq a(P,\mathcal{T}) + \mathsf{Funnel}(P_{\mathtt{L}}) + \mathsf{Funnel}(\overline{P_{\mathtt{L}}}) + \mathsf{Funnel}(P_{\mathtt{R}}) + \mathsf{Funnel}(\overline{P_{\mathtt{R}}}) && (2)\end{aligned}$$

▷ Claim 17. If $p \in P_{\mathtt{L}}$, then

$$f(P,p) \geq f(P_{\mathtt{L}},p) \quad \text{and} \quad f(\overline{P},\overline{p}) \geq f(\overline{P_{\mathtt{L}}},\overline{p});$$

and if $p \in P_{\mathtt{R}}$, then

$$f(P,p) \geq f(P_{\mathtt{R}},p) \quad \text{and} \quad f(\overline{P},\overline{p}) \geq f(\overline{P_{\mathtt{R}}},\overline{p}).$$

Proof. We will deal with the first case (the other three cases are symmetric). The key is that $P_{\mathtt{L}}$ and $P_{\mathtt{R}}$ operate on disjoint ranges of x-coordinates.

- The left funnel of p within $P_{\mathtt{L}}$ is identical to its left funnel within P, since all elements of $P_{\mathtt{R}}$ are to the right of p. Formally, $F_{\mathtt{L}}(P_{\mathtt{L}},p) = F_{\mathtt{L}}(P,p)$.
- All points q that were in the right funnel of p within $P_{\mathtt{L}}$ will still be part of the right funnel of p within P. Indeed, the only way for them to stop being funnel points would be to add accesses inside the rectangle delimited by p and q. This doesn't happen because all points in $P_{\mathtt{R}}$ are strictly to the right of all points in $P_{\mathtt{L}}$. Formally, $F_{\mathtt{R}}(P_{\mathtt{L}},p) \subseteq F_{\mathtt{R}}(P,p)$.

Therefore, $\mathrm{mix}(F_{\mathtt{L}}(P_{\mathtt{L}},p).y, F_{\mathtt{R}}(P_{\mathtt{L}},p).y)$ is a subsequence of $\mathrm{mix}(F_{\mathtt{L}}(P,p).y, F_{\mathtt{R}}(P,p).y)$, which means that

$$f(P_{\mathtt{L}},p) = \mathrm{blocks}(\mathrm{mix}(F_{\mathtt{L}}(P_{\mathtt{L}},p), F_{\mathtt{R}}(P_{\mathtt{L}},p))) \leq \mathrm{blocks}(\mathrm{mix}(F_{\mathtt{L}}(P,p), F_{\mathtt{R}}(P,p))) = f(P,p).$$

◁

Summing up $f(P,p)$ and $f(\overline{P},\overline{p})$ over all points $p \in P$, we obtain

$$\begin{aligned}\mathsf{Funnel}(P) &= \sum_{p\in P} f(P,p) \geq \sum_{p\in P_{\mathtt{L}}} f(P_{\mathtt{L}},p) + \sum_{p\in P_{\mathtt{R}}} f(P_{\mathtt{R}},p) = \mathsf{Funnel}(P_{\mathtt{L}}) + \mathsf{Funnel}(P_{\mathtt{R}})\\ \mathsf{Funnel}(\overline{P}) &= \sum_{p\in P} f(\overline{P},\overline{p}) \geq \sum_{p\in P_{\mathtt{L}}} f(\overline{P_{\mathtt{L}}},\overline{p}) + \sum_{p\in P_{\mathtt{R}}} f(\overline{P_{\mathtt{R}}},\overline{p}) = \mathsf{Funnel}(\overline{P_{\mathtt{L}}}) + \mathsf{Funnel}(\overline{P_{\mathtt{R}}}).\end{aligned} \qquad (3)$$

This, combined with (2), gives

$$\begin{aligned}\mathsf{Funnel}(P) + \mathsf{Funnel}(\overline{P}) &\geq \mathsf{Funnel}(P_{\mathtt{L}}) + \mathsf{Funnel}(P_{\mathtt{R}}) + \mathsf{Funnel}(\overline{P_{\mathtt{L}}}) + \mathsf{Funnel}(\overline{P_{\mathtt{R}}})\\ &\geq \mathsf{Alt}_{\mathcal{T}}(P) - a(P,\mathcal{T})\end{aligned}$$

This falls $a(P,\mathcal{T})$ short of our goal (which makes sense, since we haven't used the interleaving of $P_{\mathtt{L}}$ and $P_{\mathtt{R}}$ yet). To fix this, we will show the following claim.

▷ Claim 18. Consider the following properties defined over a point $p \in P$:

(a) $p \in \mathcal{T}_{\mathtt{L}}$ and $f(P,p) \geq f(P_{\mathtt{L}},p) + 1$;
(b) $p \in \mathcal{T}_{\mathtt{L}}$ and $f(\overline{P},\overline{p}) \geq f(\overline{P_{\mathtt{L}}},\overline{p}) + 1$;
(c) $p \in \mathcal{T}_{\mathtt{R}}$ and $f(P,p) \geq f(P_{\mathtt{R}},p) + 1$;
(d) $p \in \mathcal{T}_{\mathtt{R}}$ and $f(\overline{P},\overline{p}) \geq f(\overline{P_{\mathtt{R}}},\overline{p}) + 1$.

The sum of the number of points in P having each property (a)–(d) is at least $a(P, \mathcal{T})$.

Proof. Let's number the points of P by increasing y-coordinate (i.e. in chronological order) as $p_1, \ldots, p_m$. Recall that $a(P, \mathcal{T}) = \text{mixValue}(P_\mathtt{L}.y, P_\mathtt{R}.y)$. Also, $P_\mathtt{L}$ and $P_\mathtt{R}$ are non-empty, so $a(P, \mathcal{T}) \geq 2$. This means that as we go through the points $p_1, \ldots, p_m$, we switch $a(P, \mathcal{T}) - 1 \geq 1$ times between points of $P_\mathtt{L}$ and points of $P_\mathtt{R}$.

Therefore, there are exactly $a(P, \mathcal{T}) - 2$ pairs of indices (i, j) with $i + 1 < j$ such that

- case 1: $p_i, p_j \in P_\mathtt{L}$ but $p_{i+1}, \ldots, p_{j-1} \in P_\mathtt{R}$, or
- case 2: $p_i, p_j \in P_\mathtt{R}$ but $p_{i+1}, \ldots, p_{j-1} \in P_\mathtt{L}$,

which "straddle accesses of the opposite side". Also, there is an index $i^* > 1$ (the "first element of the side that starts appearing later") such that

- case 3: $p_{i^*} \in P_\mathtt{L}$ but $p_1, \ldots, p_{i^*-1} \in P_\mathtt{R}$, or
- case 4: $p_{i^*} \in P_\mathtt{R}$ but $p_1, \ldots, p_{i^*-1} \in P_\mathtt{L}$

and similarly, there is an index $j^* < m$ (the "last element of the side that finishes appearing earlier") such that

- case 5: $p_{j^*} \in P_\mathtt{L}$ but $p_{j^*+1}, \ldots, p_m \in P_\mathtt{R}$, or
- case 6: $p_{j^*} \in P_\mathtt{R}$ but $p_{j^*+1}, \ldots, p_m \in P_\mathtt{L}$.

This makes for a total of $a(P, \mathcal{T}) - 2 + 1 + 1 = a(P, \mathcal{T})$ occurrences of one of the six cases. We will show that each of them leads to a point p satisfying one of the properties (a)–(d). More precisely, we claim that:

- case 1 implies p_j has property (a) or p_i has property (b);
- case 2 implies p_j has property (c) or p_i has property (d);
- case 3 implies p_{i^*} has property (a);
- case 4 implies p_{i^*} has property (c);
- case 5 implies p_{j^*} has property (b);
- case 6 implies p_{j^*} has property (d).

We will show this for case 1 and case 3. The other four cases are analogous. To treat case 1, let's separate into more cases.[9]

- If $p_i.x < p_j.x$, then p_i is in the left funnel of p_j within both P and $P_\mathtt{L}$. But within P, p_{j-1} would be an additional right funnel point. Since it has a higher index than p_i, this would add at least 1 to $f(P, p_j)$ compared to $f(P_\mathtt{L}, p_j)$. In other words, $f(P, p_j) \geq f(P_\mathtt{L}, p_j) + 1$ (scenario (a)).
- If $p_i.x > p_j.x$, then we can use the same argument as above on $\overline{P}$ and $\overline{P_\mathtt{L}}$ by swapping i and j, obtaining $f(\overline{P}, \overline{p_i}) \geq f(P_\mathtt{L}, \overline{p_i}) + 1$ (scenario (b)).
- If $p_i.x = p_j.x$, then both funnels of p_j within $P_\mathtt{L}$ are completely empty, which means that $f(P_\mathtt{L}, p_j) = 0$, while the right funnel of p_j in P would contain at least p_{j-1}. Therefore, $f(P, p_j) = 1 \geq f(P_\mathtt{L}, p_j) + 1$ (scenario (a)).

To treat case 3, it suffices to observe that both funnels of p_{i^*} within $P_\mathtt{L}$ would be completely empty (for lack of lower points), so $f(P_\mathtt{L}, p_{i^*}) = 0$, while in P the right funnel of x_{i^*} would contain at least p_{i^*-1}. Therefore, $f(P, p_{i^*}) \geq 1 = f(P_\mathtt{L}, p_{i^*}) + 1$ (scenario (a)). ◁

Now, if we sum up $f(P, p)$ and $f(\overline{P}, \overline{p})$ over all points p as we did in (3), but this time also apply Claim 18, we obtain that

$$\mathsf{Funnel}(P) + \mathsf{Funnel}(\overline{P}) \geq \mathsf{Funnel}(P_\mathtt{L}) + \mathsf{Funnel}(P_\mathtt{R}) + \mathsf{Funnel}(\overline{P_\mathtt{L}}) + \mathsf{Funnel}(\overline{P_\mathtt{R}}) + a(P, \mathcal{T}).$$

Combined with (2), this gives the desired result and concludes the inductive step. ◀

[9] We wish we were joking.

3.2 Characterizing the Funnel bound using z-rectangles

Lemma 16 asserts that all possible Alternation bounds for all choices of reference trees $\mathcal{T}$, are simultaneously upper-bounded by the sum of two specific Funnel bounds. While this is already a nontrivial bound, $\mathsf{Funnel}(P)$ and $\mathsf{Funnel}(\overline{P})$ could in principle be wildly different, and it is therefore more compelling to show that the *single* quantity $\mathsf{Funnel}(P)$ already provides an upper bound. (It is curious that the symmetry properties of the Funnel bound, which are a necessary precondition for dynamic optimality, already enter the picture in determining the relationship between Wilber's bounds.)

To achieve this, we need to think about how geometric transformations affect the value of the Funnel bound. It is clear from the definition that $\mathsf{Funnel}(P)$ is unaffected by a horizontal flip. Indeed, the left funnel would become the right funnel and vice versa, so this wouldn't affect the number of times we switch between the two: the quantity $f(P, p)$ would remain the same for each p (see Figure 9).

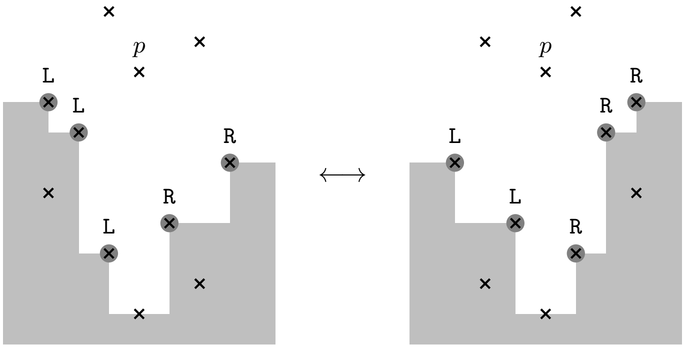


Figure 9 Flipping the geometric view horizontally conserves the contribution $f(P, p)$ of each point: the only change is that the labels of the funnel points flip between L and R.

On the other hand, it is far from obvious that the Funnel bound is unaffected by a vertical flip. Because of the time reversal, the notion of funnel changes completely. And indeed, the precise value will change, as is shown in Figure 10.

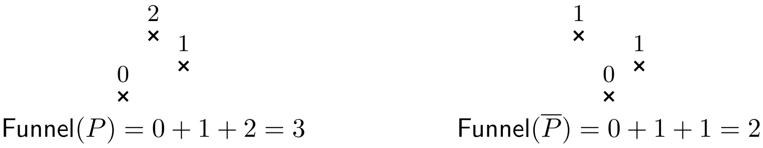


Figure 10 A minimal example such that $\mathsf{Funnel}(P) \neq \mathsf{Funnel}(\overline{P})$ is $P = \{(1,1),(3,2),(2,3)\}$. Each access p is labeled with its contribution $f(P, p)$ (left) or $f(\overline{P}, p)$ (right).

Nevertheless, we will show that for any point set P with distinct x- and y-coordinates, $\mathsf{Funnel}(P)$ and $\mathsf{Funnel}(\overline{P})$ are equal up to an additive $O(m)$. We do this by introducing a new characterization of the Funnel bound that is naturally invariant under *90° rotations* of the point set. This new characterization is the number of *z-rectangles*.

▶ **Definition 19** (z-rectangle). *Let P be a point set. We call tuple $(p, q, r, s) \in P^4$ a z-rectangle of P if the following conditions hold:*

(a) $q.x < p.x < r.x < s.x$;

(b) $r.y < q.y < s.y < p.y$;

(c) $P \cap [q.x, s.x] \times [r.y, p.y] = \{p, q, r, s\}$.

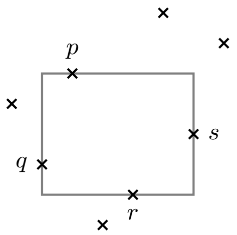


Figure 11 A z-rectangle. The relative order of points p, q, r, s horizontally and vertically matters.

In other words, a z-rectangle is a subsequence of 4 accesses with key values in relative order $3, 1, 4, 2$ and such that the axis-aligned rectangle that they span is empty (see Figure 11 for an example). We define the corresponding quantity, which we will prove is equivalent to the Funnel bound.

▶ **Definition 20** (z-rectangle bound). *For any point set P with distinct x- and y-coordinates,*[10] *let*

$$\mathsf{zRects}(P) := |\{(p, q, r, s) \mid (p, q, r, s) \textit{ is a z-rectangle of } P\}|.$$

First, we formally state the rotation-invariance of z-rectangles.

▶ **Definition 21** (counter-clockwise 90° rotation). *For a point $p \in \mathbb{R}^2$, let $p^\perp := (-p.y, p.x)$. Analogously, for a point set P, let $P^\perp := \{p^\perp \mid p \in P\}$.*

▶ **Lemma 22.** *For any point set P, $\mathsf{zRects}(P) = \mathsf{zRects}(P^\perp)$.*

Proof. Each z-rectangle of P induces a z-rectangle in $P^\perp$ and vice-versa: z-rectangle (p, q, r, s) in P becomes z-rectangle $(s^\perp, p^\perp, q^\perp, r^\perp)$ in $P^\perp$ (the reader is encouraged to physically rotate the page containing figure 11 in order to convince themselves of this fact). Therefore, P and $P^\perp$ have the same number of z-rectangles. ◀

The relation between $\mathsf{Funnel}(P)$ and $\mathsf{zRects}(P)$ is proved in the following two lemmas.

▶ **Lemma 23.** $\mathsf{zRects}(P) \geq \mathsf{Funnel}(P)/2 - O(m)$.

▶ **Lemma 24.** $\mathsf{Funnel}(P) \geq 2 \cdot \mathsf{zRects}(P)$.

We will use the fact that P has distinct x- and y- coordinates.

Proof of Lemma 23. We will show that for each $p \in P$, the funnel of p induces at least $\lfloor f(P, p)/2 \rfloor - 1$ different z-rectangles of the form $(p, \cdot, \cdot, \cdot)$. Summing this up for each p then completes the proof.

Let's assume $f(P, p) \geq 4$; otherwise the claim holds vacuously. Let's number the points in $F_{\mathtt{L}}(P, p) \cup F_{\mathtt{R}}(P, p)$ (the funnel of p) by increasing y-coordinate as $a_1, a_2, \ldots, a_l$. Note that l may be greater than $f(P, p)$, because a sequence of funnel points that are all on the same side of p counts only for 1 in $f(P, p)$.

[10] If the x- and y-coordinates are not distinct, $\mathsf{zRects}(P)$ may give absurd results. For example, if we start with any P and add a duplicate point $(x, y + \epsilon)$ for every point (x, y) of P (with ϵ small enough), then $\mathsf{zRects}(P)$ will drop to 0.

We will call $(i,j) \in [l]^2$ a *left-straddling pair* if $i+1<j$, $a_i.x > p.x$ and $a_j.x > p.x$, but for all $i<k<j$, $a_k.x < p.x$. That is, a_i and a_j are to the right of p but all funnel points between them in order of height are to the left of p. Because funnel points alternate $f(P,p)-1$ times between the left and the right of p, there must be at least $\lfloor f(P,p)/2 \rfloor - 1$ left-straddling pairs.

We claim that if (i,j) is a left-straddling pair, then (p, a_{i+1}, a_i, a_j) is a z-rectangle. Since all left-straddling pairs have distinct i, this produces $\lfloor f(P,p)/2 \rfloor - 1$ distinct z-rectangles.

First, we verify that p, a_{i+1}, a_i, a_j have the correct relative positions. The order in y-coordinate is correct by definition of the numbering $a_1, \dots, a_k$. For the order in x-coordinates, we know that a_{i+1} is to the left of p and a_i, a_j are to its right, so we only need to verify that $a_i.x < a_j.x$. This is true because a_i is in the funnel of p, so $\square pa_i$ must be empty. If $a_i.x > a_j.x$, then a_j would be in $\square pa_i$.

What we still need to prove is that rectangle $[a_{i+1}.x, a_j.x] \times [a_i.y, p.y]$ is empty (except for points p, a_{i+1}, a_i, a_j themselves). First, since a_i, a_{i+1} and a_j in the funnel of p, we know that $\square pa_i$, $\square pa_{i+1}$ and $\square pa_j$ are empty. This covers the zones pictured in Figure 12.

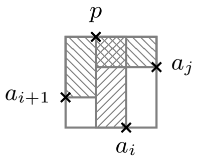


Figure 12 Proposed z-rectangle (p, a_{i+1}, a_i, a_j) with empty rectangles $\square pa_i$, $\square pa_{i+1}$ and $\square pa_j$ highlighted. If in addition we can prove that $\square a_i a_{i+1}$ and $\square a_i a_j$ are empty, then this is a valid z-rectangle.

Finally, we will prove that $\square a_i a_{i+1}$ and $\square a_i a_j$ are empty, which covers the missing parts.

- Assume $\square a_i a_{i+1}$ is not empty, and let b be the highest point of P in it (except for a_{i+1}). We have already shown that $\square pa_i$ and $\square pa_{i+1}$ are empty, so $\square pb$ must be empty. This means that b must be in the funnel of p. But $a_i.y < b.y < a_{i+1}.y$, so this contradicts the numbering by increasing y-coordinate.
- Assume $\square a_i a_j$ is not empty, and let b be the highest point of P in it (except for a_j). We have already shown that $\square pa_i$ and $\square pa_j$ are empty, so $\square pb$ must be empty. This means that b must be in the (right) funnel of p. But this contradicts our assumption that all funnel points between a_i and a_j in y-coordinate must be to the left of p.

Since points p, a_{i+1}, a_i, a_j and $[a_{i+1}.x, a_j.x] \times [a_i.y, p.y]$ is empty, (p, a_{i+1}, a_i, a_j) is a z-rectangle. This completes the proof of Lemma 23. ◀

Proof of Lemma 24. Essentially, the reason why this is true is because all z-rectangles must be exactly of the form described in the previous proof. We will prove something slightly weaker which still reaches the desired result. We will group the z-rectangles by their top point and show that if P has k rectangles of the form $(p, \cdot, \cdot, \cdot)$, then $f(P,p) \geq 2k$.

Fix p, and sort the k z-rectangles by the increasing y-coordinate of their bottom point r. Name their points (p, q_1, r_1, s_1) to (p, q_k, r_k, s_k). First, we will show that there can be no ties. Indeed, if $r_i.y = r_j.y$ then $r_i = r_j$. Also, when the p and r (top and bottom) points of a z-rectangle are fixed, then the other two points q and s are uniquely determined as the rightmost point in $(-\infty, p.x] \times [r.x, p.x]$ and the leftmost point in $[r.x, \infty) \times [r.x, p.x]$, respectively.

We will now prove that

$$q_1.y < s_1.y < q_2.y < s_2.y < \cdots < q_k.y < s_k.y. \tag{4}$$

The $q_i.y < s_i.y$ inequalities are true by the definition of a z-rectangle, so we only need to prove $s_i.y < q_{i+1}.y$. To do this, consider two consecutive z-rectangles (p, q_i, r_i, s_i) and $(p, q_{i+1}, r_{i+1}, s_{i+1})$ (see Figure 13). Since $r_i.y < r_{i+1}.y$, s_i can't be strictly to the right of r_{i+1}, because otherwise r_{i+1} would be inside z-rectangle (p, q_i, r_i, s_i). In turn, this means that s_i can't be strictly higher than r_{i+1} because otherwise it would be inside $\square pr_{i+1}$. Therefore, we have $s_i.y \leq r_{i+1}.y < q_{i+1}.y$.

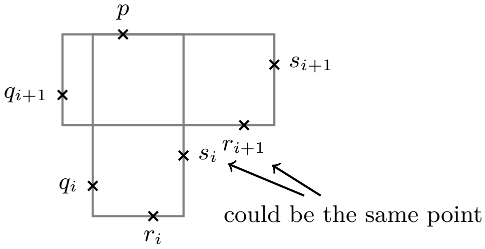


Figure 13 The only possible relative position of two z-rectangle with the same top point p.

Points $q_1, s_1, \ldots, q_k, s_k$ are all in the funnel of p by the definition of z-rectangle. Therefore, Equation (4) reveals $2k$ funnel points that alternate from the left to the right side of p with increasing y-coordinates. Thus $\text{mix}(F_{\mathtt{L}}(P,p).y, F_{\mathtt{R}}(P,p).y)$ contains a subsequence $\mathtt{LRLR}\cdots\mathtt{LR}$ of length $2k$, and

$$f(P,p) = \text{blocks}(\text{mix}(F_{\mathtt{L}}(P,p).y, F_{\mathtt{R}}(P,p).y)) \geq \text{blocks}(\underbrace{\mathtt{LRLR}\cdots\mathtt{LR}}_{\text{length } 2k}) = 2k.$$

Summing this up for each p completes the proof. ◀

▶ **Corollary 25.** $\mathsf{Funnel}(P) \geq \mathsf{Funnel}(\overline{P}) - O(m)$.

Proof. By the left-right symmetry of $\mathsf{Funnel}(\cdot)$, we know that $\mathsf{Funnel}(\overline{P}) = \mathsf{Funnel}(P^{\perp\perp})$, where $P^{\perp\perp}$ is P rotated by 180°. Therefore,

$$\begin{aligned}
\mathsf{Funnel}(P) &\geq 2 \cdot \mathsf{zRects}(P) && \text{(Lemma 24)}\\
&= 2 \cdot \mathsf{zRects}(P^{\perp\perp}) && \text{(Lemma 22)}\\
&\geq \mathsf{Funnel}(P^{\perp\perp}) - O(m) && \text{(Lemma 23)}\\
&= \mathsf{Funnel}(\overline{P}) - O(m).
\end{aligned}$$
◀

We can now finally prove Theorem 1.

Proof of Theorem 1. By Lemma 16, $\mathsf{Alt}_{\mathcal{T}}(P) \leq \mathsf{Funnel}(P) + \mathsf{Funnel}(\overline{P})$. Combining this with Corollary 25, we obtain $\mathsf{Alt}_{\mathcal{T}}(P) \leq \mathsf{Funnel}(P) + (\mathsf{Funnel}(P) + O(m)) \leq O(\mathsf{Funnel}(P) + m)$. ◀

4 Separation between the Alternation bound and the Funnel bound

We will now define an access sequence $\tilde{X}$ such that the Alternation bound is too low for all reference trees $\mathcal{T}$ simultaneously. More precisely, we will define an access sequence $\tilde{X} \in [n]^m$ such that $\mathsf{Alt}_{\mathcal{T}}(\tilde{X}) = O(m)$ for all trees $\mathcal{T}$ while on the other hand $\mathsf{OPT}(\tilde{X})$ and $\mathsf{Funnel}(\tilde{X})$ are $\Theta(m \lg\lg n)$. This $\lg\lg n$ factor is the biggest possible separation: indeed, Tango trees show that for a balanced tree $\mathcal{T}$, $\mathsf{Alt}_{\mathcal{T}}(X)$ is always within $O(\lg\lg n)$ of $\mathsf{OPT}(X)$.

To define $\tilde{X}$, we will need the notion of a *bit-reversal* sequence. This is a permutation that in a sense looks "maximally shuffled" to a binary search tree.

▶ **Definition 26.** *Let k be a positive integer and let $K = 2^k$. Then let* $\text{bitReversal}^k \in \{0, \cdots, K-1\}^K$ *be the sequence where* bitReversal^k_i *is the number obtained by taking the binary representation of $i-1$, padding it with leading zeroes to reach length k, flipping it, then converting this back to a number.*

It is easiest to understand through an example. Take $k = 2$, then bitReversal^2 is obtained this way:

$$(0,1,2,3) \xrightarrow{\text{to binary}} (00,01,10,11) \xrightarrow{\text{flip}} (00,10,01,11) \xrightarrow{\text{from binary}} (0,2,1,3).$$

The reason why we use this sequence is the following well-known fact.

▶ **Fact 27.** *Let $\mathcal{T}$ be the complete binary tree of height k which has K leaves labeled 0 through $K-1$. Then $\mathsf{Alt}_{\mathcal{T}}(\text{bitReversal}^k) = kK = K \lg K$.*

Proof. Because of the way bitReversal^k is defined, for each node $u \in \mathcal{T}$, the keys that are accessed below u as the sequence is processed constantly alternate from u's left subtree to u's right subtree. So the contribution of u is exactly the number of keys of its subtree. This way, every key is counted once at each of the $k = \lg K$ levels, so the total is $K \lg K$. ◀

We can now define our access sequence as follows. Let $n \coloneqq 2^K = 2^{2^k}$, and let

$$S_i \coloneqq (i + 2^{\text{bitReversal}^k_1}, i + 2^{\text{bitReversal}^k_2}, \ldots, i + 2^{\text{bitReversal}^k_K}).$$

Then, denoting concatenation by $\circ$, we define

$$\tilde{X} \coloneqq \underbrace{S_0 \circ \cdots \circ S_0}_{n \text{ times}} \circ \underbrace{S_1 \circ \cdots \circ S_1}_{n \text{ times}} \circ \cdots \circ \underbrace{S_{n/2} \circ \cdots \circ S_{n/2}}_{n \text{ times}}.$$

The range of $\tilde{X}$ is $[n]$ and its length is $m = (\frac{n}{2}+1) \cdot n \cdot K = \Theta(n^2 \lg n)$. See Figure 14 for an example with $k = 2$. We will prove that for all $\mathcal{T}$, $\mathsf{Alt}_{\mathcal{T}}(\tilde{X}) \le O(m)$ while on the other hand $\mathsf{Funnel}(\tilde{X}) \ge \Omega(m \lg\lg n)$.

▶ **Lemma 28.** *For any $\mathcal{T}$, $\mathsf{Alt}_{\mathcal{T}}(\tilde{X}) \le O(m)$.*

▶ **Lemma 29.** *$\mathsf{Funnel}(\tilde{X}) \ge \Omega(m \lg\lg n)$.*

The combination of Lemma 28 and Lemma 29 shows the separation claimed in Theorem 2. Before we move to the proofs of those lemmas, let's go over some intuition for the proof of Lemma 28, which is the more complicated one.

First, note that the only reason we use bitReversal^k in $\tilde{X}$ is to make $\mathsf{Funnel}(\tilde{X})$ large. Replacing bitReversal^k by any other permutation of $\{0, \ldots, K-1\}$ would not affect the proof of Lemma 28 in any way because that proof only looks at the *set* of keys that are hit by each of the parts $S_0, \ldots, S_{n/2}$.

The general intuition of the proof of Lemma 28 is that while one tree could give a high lower bound for *one* of the sequences S_i, no tree can give a high lower bound *on average* over all S_i. The reason is that, given the geometric spacing of each S_i, any way to split an interval of keys into two will typically (on average over i) leave almost all the keys of S_i in either the left or the right part (Claim 31). Therefore, it is impossible to split the keys into subtrees in a way that would ensure a high number of alternations.

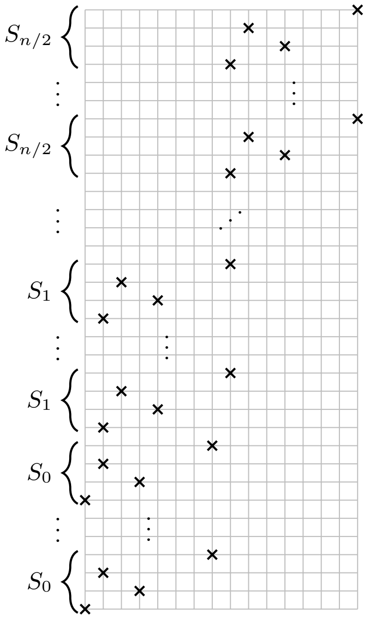


Figure 14 A schematic view of sequence $\tilde{X}$ for $k = 2$. Each part S_i is made of $K = 2^k = 4$ accesses. There are $n = 2^K = 16$ distinct keys and the length of $\tilde{X}$ is $m = (16/2 + 1)nK = 576$.

Proof of Lemma 28. The first step of the proof is to decompose $\tilde{X}$ into substrings $S_0 \circ \cdots \circ S_0$ through $S_{n/2} \circ \cdots \circ S_{n/2}$, and then bound the sum of their Alternation bounds. Let's denote those substrings as $S_0 * n$, $S_1 * n$, $\dots$, $S_{n/2} * n$. Because of the subadditivity of mixValue under concatenation (Fact 9), we have

$$\mathsf{Alt}_{\mathcal{T}}(\tilde{X}) \leq \sum_{i=0}^{n/2} \mathsf{Alt}_{\mathcal{T}}(S_i * n). \tag{5}$$

Note that we don't want to decompose $\tilde{X}$ down to the S_i's themselves: every time we split it, our analysis loses up to an additive $O(n)$ in precision. Intuitively, this $O(n)$ is due to a "warmup" cost which we might or might not incur at the beginning of each substring, depending on which parts of the tree were last visited. With our decomposition into n substrings, that's an extra $O(n^2)$ cost, which is okay since it is small compared to the total length of the sequence $\Theta(n^2 \log n)$. In fact, this is precisely why we repeated each S_i several times: if we had defined $\tilde{X}$ as $S_0 \circ S_1 \circ \cdots \circ S_{n/2}$ instead, this $O(n^2)$ would have been large compared to the length of the sequence $\Theta(n \log n)$.[11]

We will upper-bound the sum $\sum_i \mathsf{Alt}_{\mathcal{T}}(S_i * n)$ by induction on the recursive definition of $\mathsf{Alt}_{\mathcal{T}}(\cdot)$. Concretely, let $\mathcal{T}^*$ be a subtree of $\mathcal{T}$, and let ${\mathcal{T}_{\mathtt{L}}}^*, {\mathcal{T}_{\mathtt{R}}}^*$ be the left and right subtrees of $\mathcal{T}^*$. Let s, $s_{\mathtt{L}}$ and $s_{\mathtt{R}}$ be the number of keys in $\mathcal{T}^*$, ${\mathcal{T}_{\mathtt{L}}}^*$ and ${\mathcal{T}_{\mathtt{R}}}^*$ (note that $s = s_{\mathtt{L}} + s_{\mathtt{R}}$). For each i, let P_i^* be the subset of $P(S_i * n)$ corresponding to keys in $\mathcal{T}^*$, and let $P_{i,\mathtt{L}}^*, P_{i,\mathtt{R}}^*$ be the same for ${\mathcal{T}_{\mathtt{L}}}^*$ and ${\mathcal{T}_{\mathtt{R}}}^*$. We will prove the following claim by induction:

[11] The astute reader will notice that we could have repeated each S_i only $\Theta(n/\log n)$ times instead of n times. But we are not limited in terms of the length of $\tilde{X}$, so it was (notationally) simpler to repeat them n times.

▷ Claim 30. For some constant $C > 0$,

$$\sum_{i=0}^{n/2} \mathsf{Alt}_{\mathcal{T}^*}(P_i^*) \le (s-1)(n/2+1) + 2Cns\lg s.$$

The base case is when $\mathcal{T}^*$ is a single node. Then $\mathsf{Alt}_{\mathcal{T}^*}(S_i * n) = 0$ for all i, while $s = 1$, so the result holds. To deal with the inductive step, we will need make a few tools first. By definition of the Alternation bound (Definition 10), for each i we have

$$\mathsf{Alt}_{\mathcal{T}^*}(P_i^*) = a(P_i^*, \mathcal{T}^*) + \mathsf{Alt}_{\mathcal{T}_{\mathtt{L}}{}^*}(P_{i,\mathtt{L}}^*) + \mathsf{Alt}_{\mathcal{T}_{\mathtt{R}}{}^*}(P_{i,\mathtt{R}}^*). \tag{6}$$

The challenging part is how to deal with $a(P_i^*, \mathcal{T}^*)$. By Fact 9, we have

$$a(P_i^*, \mathcal{T}^*) = \text{mixValue}(P_{i,\mathtt{L}}^*.y, P_{i,\mathtt{R}}^*.y) \le 2 \cdot \min(|P_{i,\mathtt{L}}^*|, |P_{i,\mathtt{R}}^*|) + 1.$$

Summing this up over all i, we get

$$\sum_{i=0}^{n/2} a(P_i^*, \mathcal{T}^*) \le \sum_{i=0}^{n/2} (2 \cdot \min(|P_{i,\mathtt{L}}^*|, |P_{i,\mathtt{R}}^*|) + 1) = (n/2+1) + 2 \cdot \sum_{i=0}^{n/2} \min(|P_{i,\mathtt{L}}^*|, |P_{i,\mathtt{R}}^*|). \tag{7}$$

▷ Claim 31. For some constant $C > 0$,

$$\sum_{i=0}^{n/2} \min(|P_{i,\mathtt{L}}^*|, |P_{i,\mathtt{R}}^*|) \le Cn \cdot \begin{cases} s_{\mathtt{L}} \lg \frac{s}{s_{\mathtt{L}}} & \text{if } s_{\mathtt{L}} \le s_{\mathtt{R}}, \text{ and} \\ s_{\mathtt{R}} \lg \frac{s}{s_{\mathtt{R}}} & \text{if } s_{\mathtt{R}} \le s_{\mathtt{L}}. \end{cases}$$

This left-right symmetry is very surprising given that the sequences S_i themselves are *not* left-right symmetric. But it will be very convenient.

Proof. To simplify the notation, let's say that the keys in $\mathcal{T}_{\mathtt{L}}{}^*$ are in range $[a, b]$ and the keys in $\mathcal{T}_{\mathtt{R}}{}^*$ are in range $[b, c]$, for some real numbers a, b, c with $b - a = s_{\mathtt{L}}$ and $c - b = s_{\mathtt{R}}$.[12]

For each i, let $V_i = \{i + 2^0, \dots, i + 2^{K-1}\}$ be the set of values that are hit by sequence S_i. Then $|P_{i,\mathtt{L}}^*|$ (resp. $|P_{i,\mathtt{L}}^*|$) is exactly n times the number of elements of V_i that are in $[a, b]$ (resp. $[b, c]$). Let's name this number of keys l_i (resp. r_i). We will instead prove that

$$\sum_{i=0}^{n/2} \min(l_i, r_i) \le O\left(s_{\mathtt{L}} \lg \frac{s}{s_{\mathtt{L}}}\right) \text{ if } s_{\mathtt{L}} \le s_{\mathtt{R}}, \text{ and} \tag{8}$$

$$\sum_{i=0}^{n/2} \min(l_i, r_i) \le O\left(s_{\mathtt{R}} \lg \frac{s}{s_{\mathtt{R}}}\right) \text{ if } s_{\mathtt{R}} \le s_{\mathtt{L}}. \tag{9}$$

Once this is proved, C can be set to the maximum of the two constants hidden inside the $O(\cdot)$s. Those constants might be different since the reasonings leading to (8) and (9) are completely different.

We first make a general observation. Look at set $V_i = \{i+2^0, \dots, i+2^j, \dots, \}$ in increasing order. Note that after $i + 2^j$, all further elements are spaced by at least 2^j. In order for $\min(l_i, r_i)$ to be non-zero, we need to have at least two elements of S_i in $[a, c]$: specifically, one in $[a, b]$ and one in $[b, c]$. But this means that $i + 2^{j+1} \in [a, c]$ isn't acceptable for $j > \lg s$:

[12] We can for example fix a to the first key of $\mathcal{T}_{\mathtt{L}}{}^*$ minus $\frac{1}{2}$, b to the last key of $\mathcal{T}_{\mathtt{L}}{}^*$ plus $\frac{1}{2}$, and c to the last key of $\mathcal{T}_{\mathtt{R}}{}^*$ plus $\frac{1}{2}$.

indeed, the closest other point in S_i is more than s away, so it must be outside of $[a, c]$. Therefore, in bounding $\sum \min(l_i, r_i)$, it is fine to imagine that the elements $i + 2^{j+1}$ for $j > \lg s$ simply do not exist.

Let us now prove (8). Assume $s_\mathrm{L} \leq s_\mathrm{R}$. We split into two cases:

- "Far" case: $i < a - s_\mathrm{L}$. Since i is further from $[a, b]$ than its size s_L, this means that $[a, b]$ can only contain at most one point from S_i. So $l_i \leq 1$. Besides, that (potential) single point must have $j \leq 1 + \lg s$ (see above) and $j \geq \lg s_\mathrm{L}$ (because we have $i + 2^j \geq a$). And of course, we have in addition that $i + 2^j \in [a, b]$. Therefore, this limits the number of possible values of i to at most $s_\mathrm{L}(2 + \lg s - \lg s_\mathrm{L})$, and since $l_i \leq 1$, this also limits the total contribution to $\sum \min(l_i, r_i)$.
- "Close to right" case: $i \geq a - s_\mathrm{L}$. Then we also have $i \geq b - 2s_\mathrm{L}$. Since we need $l_i \neq 0$ to have some contribution, we must have $i < b$, so the total number of possible values of i is limited to $2s_\mathrm{L}$. Let's consider the values of j such that $i + 2^j$ can lie in $[b, c]$, the right part. We already know that $j \leq 1 + \lg s$, but we have no lower limit, as i could be very close to b. However, values of j much smaller than $\lg s_\mathrm{L}$ will be only for the few values of i close enough to b.

  More precisely, we study the contribution of each j to $\sum r_i$ into two groups:
  - $j \geq \lg s_\mathrm{L}$: there are $2 + \lg s - \lg s_\mathrm{L}$ such values j, and there are $2s_\mathrm{L}$ possible values of i, so the total contribution is at most $2s_\mathrm{L}(2 + \lg s - \lg s_\mathrm{L})$.
  - $j < \lg s_\mathrm{L}$: as j decreases, the number of acceptable values of i decreases exponentially. The number of values of i for which $i + 2^j \in [b, c]$ for $j \leq \lg s_\mathrm{L} - l$ is at most $s_\mathrm{L}/2^l$. Therefore, the overal contribution is at most $s_\mathrm{L} + s_\mathrm{L}/2 + \cdots \leq 2s_\mathrm{L}$.

All those quantities are upper bounded by $O(s_\mathrm{L}(1 + \lg(s/s_\mathrm{L})))$, which under the assumption $s_\mathrm{L} \leq s_\mathrm{R}$, is also bounded by $O(s_\mathrm{L} \lg(s/s_\mathrm{L}))$.

We now prove (9) in a very similar way. Assume $s_\mathrm{L} \leq s_\mathrm{R}$.

- "Far" case: $i < b - s_\mathrm{R}$. The argument is analogous to the "far" case for (8), but considering r_i this time. We obtain a contribution of at most $s_\mathrm{R}(2 + \lg s - \lg s_\mathrm{R})$.
- "Close to right" case: $i \geq b - s_\mathrm{R}$. The argument is analogous to the "close to right" case for (8), but with a distance of s_R instead of $2s_\mathrm{L}$ this time. We obtain contributions of at most $s_\mathrm{R}(2 + \lg s - \lg s_\mathrm{R})$ and $2s_\mathrm{R}$ for the two subcases.

All those quantities are upper bounded by $O(s_\mathrm{R}(1 + \lg(s/s_\mathrm{R})))$, which under the assumption $s_\mathrm{R} \leq s_\mathrm{L}$, is also bounded by $O(s_\mathrm{R} \lg(s/s_\mathrm{R}))$. ◁

We are now ready to finish the induction step.

Proof of Claim 30. We define C to be the same as in Claim 31. We have

$$\begin{aligned}
\sum_{i=0}^{n/2} \mathsf{Alt}_{\mathcal{T}^*}(P_i^*) &= \sum_{i=0}^{n/2} \left(a(P_i^*, \mathcal{T}^*) + \mathsf{Alt}_{\mathcal{T}_\mathrm{L}^*}(P_{i,\mathrm{L}}^*) + \mathsf{Alt}_{\mathcal{T}_\mathrm{R}^*}(P_{i,\mathrm{R}}^*) \right) && \text{(by (6))} \\
&\leq \left(\sum_{i=0}^{n/2} a(P_i^*, \mathcal{T}^*) \right) + (s_\mathrm{L} - 1)(n/2 + 1) + 2Cns_\mathrm{L} \lg s_\mathrm{L} + (s_\mathrm{R} - 1)(n/2 + 1) + 2Cns_\mathrm{R} \lg s_\mathrm{R} \\
& && \text{(inductive hypothesis)} \\
&\leq (n/2 + 1) + 2 \cdot \sum_{i=0}^{n/2} \min(|P_{i,\mathrm{L}}^*|, |P_{i,\mathrm{R}}^*|) \\
&\qquad + (s_\mathrm{L} - 1)(n/2 + 1) + 2Cns_\mathrm{L} \lg s_\mathrm{L} + (s_\mathrm{R} - 1)(n/2 + 1) + 2Cns_\mathrm{R} \lg s_\mathrm{R} && \text{(by (7))} \\
&\leq (s - 1)(n/2 + 1) + 2Cn(s_\mathrm{L} \lg s_\mathrm{L} + s_\mathrm{R} \lg s_\mathrm{R}) + 2 \cdot \sum_{i=0}^{n/2} \min(|P_{i,\mathrm{L}}^*|, |P_{i,\mathrm{R}}^*|) && (s = s_\mathrm{L} + s_\mathrm{R})
\end{aligned}$$

All we need to show is that

$$Cn(s_{\mathtt{L}} \lg s_{\mathtt{L}} + s_{\mathtt{R}} \lg s_{\mathtt{R}}) + \sum_{i=0}^{n/2} \min(|P^*_{i,\mathtt{L}}|, |P^*_{i,\mathtt{R}}|) \le Cn(s \lg s).$$

Let's assume that $s_{\mathtt{L}} \le s_{\mathtt{R}}$ (the other case is identical). Then by Claim 31,

$$\begin{aligned} Cn(s_{\mathtt{L}} \lg s_{\mathtt{L}} + s_{\mathtt{R}} \lg s_{\mathtt{R}}) + \sum_{i=0}^{n/2} \min(|P^*_{i,\mathtt{L}}|, |P^*_{i,\mathtt{R}}|) &\le Cn(s_{\mathtt{L}} \lg s_{\mathtt{L}} + s_{\mathtt{R}} \lg s_{\mathtt{R}}) + Cns_{\mathtt{L}} \lg \frac{s}{s_{\mathtt{L}}} \\ &\le Cn(s_{\mathtt{L}} \lg s + s_{\mathtt{R}} \lg s_{\mathtt{R}}) \\ &\le Cn(s_{\mathtt{L}} \lg s + s_{\mathtt{R}} \lg s) \\ &= Cns \lg s. \end{aligned}$$

This completes the proof of Claim 30. ◁

Applying Claim 30 to the full tree $\mathcal{T}$, which has n keys, we get

$$\begin{aligned} \mathsf{Alt}_{\mathcal{T}}(\tilde{X}) &\le \sum_{i=0}^{n/2} \mathsf{Alt}_{\mathcal{T}^*}(S_i * n) && \text{(by (5))} \\ &\le (n-1)(n/2+1) + 2Cn^2 \lg n && \text{(Claim 30)} \\ &\le O(n^2 \lg n) \\ &= O(m). \end{aligned}$$ ◀

We now move to the proof of Lemma 29, which is much simpler.

Proof of Lemma 29. From the definition of $\mathsf{Funnel}(\cdot)$ (Definition 14), it is easy to see that for any two sequences S and T, $\mathsf{Funnel}(S \circ T) \ge \mathsf{Funnel}(S) + \mathsf{Funnel}(T)$. Indeed concatenating S and T does not affect the funnel of each point in S, and can only add points to the funnel of each point in T. Therefore,

$$\mathsf{Funnel}(\tilde{X}) \ge n \sum_{i=0}^{n/2} \mathsf{Funnel}(S_i). \tag{10}$$

Since $\mathsf{Funnel}(\cdot)$ only depends on the relative order of the keys in the access sequence, not on their exact value, we have $\mathsf{Funnel}(S_i) = \mathsf{Funnel}(\text{bitReversal}^k)$ for each i. Besides, defining $\mathcal{T}$ to be the complete binary search tree of height k as in Fact 27, we have

$$\begin{aligned} \mathsf{Funnel}(\text{bitReversal}^k) &\ge \Omega(\mathsf{Alt}_{\mathcal{T}}(\text{bitReversal}^k)) - K && \text{(by Theorem 1)} \\ &\ge \Omega(K \lg K) - K && \text{(by Fact 27)} \\ &\ge \Omega(K \lg K). \end{aligned}$$

Combined with (10), this gives $\mathsf{Funnel}(\tilde{X}) \ge n \cdot (n/2+1) \cdot \Omega(K \lg K) \ge \Omega(m \lg K) = \Omega(m \lg \lg n)$. ◀

Due to space constraints, the last (short) section, which relates the Funnel bound to the Independent Rectangle bound, is deferred to the full version.

References

1 Brian Allen and Ian Munro. Self-organizing binary search trees. *J. ACM*, 25(4):526–535, October 1978. doi:10.1145/322092.322094.

2 Prosenjit Bose, Karim Douïeb, Vida Dujmovic, and Rolf Fagerberg. An $O(\log \log n)$-competitive binary search tree with optimal worst-case access times. In *Algorithm Theory - SWAT 2010, 12th Scandinavian Symposium and Workshops on Algorithm Theory, Bergen, Norway, June 21-23, 2010. Proceedings*, pages 38–49, 2010. doi:10.1007/978-3-642-13731-0_5.

3 Parinya Chalermsook, Julia Chuzhoy, and Thatchaphol Saranurak. Pinning down the strong wilber 1 bound for binary search trees. *CoRR*, abs/1912.02900, 2019. arXiv:1912.02900.

4 Parinya Chalermsook, Mayank Goswami, László Kozma, Kurt Mehlhorn, and Thatchaphol Saranurak. Pattern-avoiding access in binary search trees. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 410–423. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.32.

5 Erik D. Demaine, Dion Harmon, John Iacono, Daniel M. Kane, and Mihai Patrascu. The geometry of binary search trees. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, New York, NY, USA, January 4-6, 2009*, pages 496–505, 2009. URL: http://dl.acm.org/citation.cfm?id=1496770.1496825.

6 Erik D. Demaine, Dion Harmon, John Iacono, and Mihai Patrascu. Dynamic optimality - almost. *SIAM J. Comput.*, 37(1):240–251, 2007. doi:10.1137/S0097539705447347.

7 John Iacono. Key-independent optimality. *Algorithmica*, 42(1):3–10, 2005. doi:10.1007/s00453-004-1136-8.

8 John Iacono. In pursuit of the dynamic optimality conjecture. In *Space-Efficient Data Structures, Streams, and Algorithms - Papers in Honor of J. Ian Munro on the Occasion of His 66th Birthday*, pages 236–250, 2013. doi:10.1007/978-3-642-40273-9_16.

9 László Kozma and Thatchaphol Saranurak. Smooth heaps and a dual view of self-adjusting data structures. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 801–814, 2018. doi:10.1145/3188745.3188864.

10 Victor Lecomte and Omri Weinstein. Settling the relationship between wilber's bounds for dynamic optimality. *CoRR*, abs/1912.02858, 2019. arXiv:1912.02858.

11 Caleb C. Levy and Robert E. Tarjan. A new path from splay to dynamic optimality. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1311–1330. SIAM, 2019. doi:10.1137/1.9781611975482.80.

12 Joan Marie Lucas. *Canonical forms for competitive binary search tree algorithms*. Rutgers University, Department of Computer Science, Laboratory for Computer Science Research, 1988.

13 J. Ian Munro. On the competitiveness of linear search. In Mike Paterson, editor, *Algorithms - ESA 2000, 8th Annual European Symposium, Saarbrücken, Germany, September 5-8, 2000, Proceedings*, volume 1879 of *Lecture Notes in Computer Science*, pages 338–345. Springer, 2000. doi:10.1007/3-540-45253-2_31.

14 Daniel Dominic Sleator and Robert Endre Tarjan. Self-adjusting binary search trees. *J. ACM*, 32(3):652–686, July 1985. doi:10.1145/3828.3835.

15 Chengwen Chris Wang, Jonathan Derryberry, and Daniel Dominic Sleator. O(log log n)-competitive dynamic binary search trees. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm*, SODA '06, pages 374–383, Philadelphia, PA, USA, 2006. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=1109557.1109600.

16 R. Wilber. Lower bounds for accessing binary search trees with rotations. *SIAM J. Comput.*, 18(1):56–67, February 1989. doi:10.1137/0218004.