Alcom-FT Technical Report Series
ALCOMFT-TR-02-119

Fringe Analysis of Synchronized Parallel Insertion Algorithms in 2–3 Trees ⋆

R. Baeza-Yates[1], J. Gabarró[2], and X. Messeguer[2]

[1] Departamento de Ciencias de la Computación, Universidad de Chile
[2] Dep. LSI, Universitat Politècnica de Catalunya, Barcelona

Abstract. Fringe analysis uses the distribution of bottom subtrees or fringe of search trees under the assumption of random insertion of keys, yielding an average case analysis of the fringe. The results in the fringe give upper and lower bounds for several measures for the whole tree.
We are interested in the fringe analysis of the synchronized parallel insertion algorithms of Paul, Vishkin, and Wagener (PVW) on 2–3 trees. This algorithm inserts k keys with k processors into a tree of size n with time $O(\log n + \log k)$. As the direct analysis of this algorithm is very difficult we tackle this problem by introducing a new family of algorithms, denoted by MacroSplit algorithms, and our main theorem proves that two algorithms of this family, denoted MaxMacroSplit and MinMacroSplit, bound the behavior of the fringe in the PVW algorithm.
Previous work deals with the fringe analysis of sequential algorithms, but this type of analysis was still an open problem for parallel algorithms on search trees. We extend fringe analysis to parallel algorithms and we get a rich mathematical structure giving new interpretations even in the sequential case. We prove that random insertion of keys generates a binomial distribution, that the synchronized insertion of keys can be modeled by a Markov chain, and that the coefficients of the transition matrix of the Markov chain are related to the expected local behavior of our algorithm. Finally, we show that the coefficients of the power expansion of this matrix over $(n+1)^{-1}$ are the binomial transform of the expected local behavior of the algorithm. We finally show that the fringe of the PVW algorithm asymptotically converges to the sequential case.

Keywords: *Fringe analysis, parallel algorithms, 2-3 trees, binomial transform.*

1 Introduction

One of the basic problems of managing information is the dictionary problem, where a set of keys has to be dynamically maintained. One solution to this problem is a balanced search tree like the 2-3 tree introduced by J. Hopcroft in the 1970s [AHU74]. A detailed analysis of this algorithm is still lacking, but good lower and upper bounds for several complexity measures have been obtained using a technique called *fringe analysis* [EZG+82,BYP95]. This analysis studies the distribution of bottom subtrees or fringe of search trees under the assumption of the insertion of random keys, and has been applied to most search trees [EZG+82,BY95]. Note that fringe analysis is the average case analysis of the fringe of the tree. The fringe analysis has also been modeled using generalized urn schemes [BP85,AFP88]. The results in the fringe allows to bound several measures on the whole tree.

We are interested in the fringe analysis of the synchronized parallel algorithm on 2–3 trees designed by Paul, Vishkin, and Wagener (PVW)[PVW83]. This kind of algorithms manages data types in a synchronized manner (PRAM algorithms [JáJ92]). They can be envisaged as many sequential algorithms running simultaneously and executing the same operation at the same time. Therefore, it may happen that several processes read or write on the same memory location at the same time. The goal is to avoid those concurrent accesses. The first synchronized parallel algorithm on search trees was the PVW algorithm. The time needed to search or update k elements with k processors on a tree with n keys is $O(\log n + \log k)$ which is very close to the optimal speedup of $O(\log n)$. A detailed analysis of this algorithm is still lacking and the main drawback is

⋆ Partially supported by: (i) ACI-CONICYT grant through the Catalunya-Chile Cooperation Program (DOG 2320-30.1.1997), (ii) AMYRI VII.13 project (CYTED), (iii) Fondecyt grant 199-0627 (CONICYT), (iv) the IST program of the EU under contract number IST-1999-14186 (ALCOM-FT), (v) Spanish CICYT under grant TIC1999-0754-C03-02 (Mallba), and (vi) Spanish CICYT under grant TIC-2000-1970.

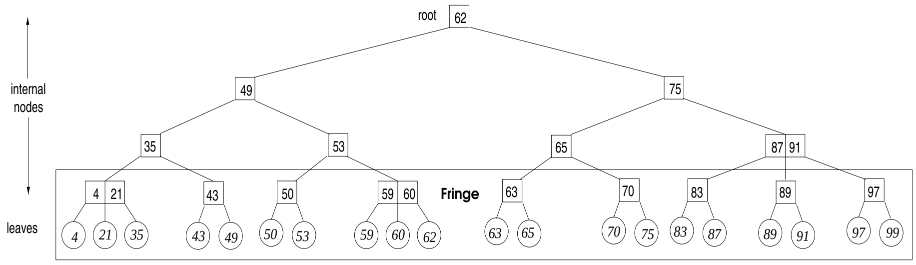


Fig. 1. An example of 2-3 tree. The fringe has been enclosed in a rectangle.

the reconstructing phase that is composed of waves of synchronized processors which modify the tree in a bottom-up fashion. A top-down approach has been developed in [BRM99].

In this paper we introduce a new family of synchronized parallel algorithms, denoted MacroSplit (the prefix "macro" means that nodes with a very large number of keys are accepted), whose two extreme cases, denoted MaxMacroSplit and MinMacroSplit algorithms, bound the PVW algorithm in the following sense: the expected values of the fringe derived from the PVW algorithm are upper and lower bounded by the expected values derived from these two extreme cases. The key idea is that the fringe analysis works for the MacroSplit algorithms because they reconstruct the tree with only one wave while the PVW algorithm needs a pipeline of waves.

The fringe analysis of the MacroSplit algorithms is an extension of the same type of analysis for the sequential case but with many significant improvements. As shown later on, the direct extensions of this technique for the parallel insertion of two and three keys suggest the inapplicability of this technique for the case of inserting more keys. We have overcome this limitation with two facts that allow us the analysis of the generic case (the insertion of k keys):

- The insertion of random keys generates a *binomial* distribution on each bottom node (nodes from which the leaves are attached). This fact allows us to analyze the local behavior for any bottom node.
- The global behavior or fringe evolution of all nodes can be analyzed because we prove that this binomial distribution can be assumed for all bottom nodes *simultaneously*. Then the global behavior is determined by the expected local behavior of the algorithm.

The relation between the global and the local behavior of the MacroSplit algorithms gives a new theoretical explanation to fringe analysis, and we develop both cases, obtaining the power expansion of the transition matrix and we calculate its coefficients for the two algorithms MaxMacroSplit and MinMacroSplit. Finally, we analyze both cases and we show that the parallel case converges asymptotically to the solution of the sequential case. Our results are the first on the expected behavior of parallel insertion algorithms in balanced trees using fringe analysis.

The rest of the paper is organized following the main facts pointed out in this introduction. In sections 2 and 3 we recall the fringe analysis of the sequential case and we introduce the PVW algorithm and the family of MacroSplit algorithms. Section 4 develops the direct extension of the sequential fringe analysis for the parallel insertion of two and three keys and discusses the inapplicability of this extension for more keys. Section 5 contains the analysis of the MacroSplit algorithms, relates their local and global behavior and develops the power expansion of the transition matrix. Section 6 contains the detailed results for the two concrete algorithms MaxMacroSplit and MinMacroSplit, and the analysis of the fringe. Section 7 shows that the fringe generated by these two algorithms bounds the fringe generated by the PVW algorithm. Finally, the last section contains the main conclusions and future work. A preliminary and partial version of this paper was presented in [BYGM98].

2 Fringe analysis for sequential insertions

Let us introduce 2–3 trees. They are balanced search trees whose nodes can be either *internal nodes* having two or three sons or *leaves* with no sons (see Figure 1).

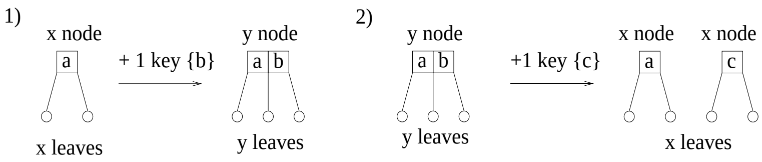


Fig. 2. The transformation of x and y bottom nodes after inserting one key. In case (1) the key b hits a bottom x node which is transformed into a y node. In case (2) the key c hits a bottom y node which is split into two x nodes.

Definition 1. [AHJ83] A 2–3 tree is a search tree with the following properties:

1. Each internal node has two or three sons.
2. Each path from the root to a leaf has the same length.

In general the fringe of a tree is composed of the subtrees on the bottom part of the tree; in this paper the fringe is given by subtrees of height one. A bottom node (or terminal node) with one key is called an x node, and a bottom node with two keys is called a y node. These nodes separate the leaves into x leaves if their parents are x nodes and y leaves if their parents are y nodes.

Let X_t and Y_t be the random number of x leaves and y leaves respectively at step t. Figure 1 gives us an example of the fringe at $t = 0$ with $X_0 = 14$ and $Y_0 = 6$. Notice that $X_t + Y_t = n + 1$, n being the number of keys of the tree.

When a new key falls into a bottom node this node is transformed according to the following rules (see Figure 2): if a key b hits a bottom x node that contains the key a then the x node is transformed into a y node having keys a and b (Case 1 of Figure 2). We have $X_{t+1} = X_t - 2$ and $Y_{t+1} = Y_t + 3$. If a key c hits a bottom y node containing a and b then the y node splits into two x nodes containing a and c respectively, while b is inserted in the parent node recursively (Case 2 of Figure 2). Now $X_{t+1} = X_t + 4$ and $Y_{t+1} = Y_t - 3$.

We need a probabilistic model of random 2–3 tree in order to study the expected behavior of the random variables. We recall the definition of a random insertion in a 2–3 tree from Yao's paper.

Definition 2. [Yao78a] A 2–3 tree with n keys determines $n + 1$ intervals. An insertion of a new key is *random* if it has equal probability of being inserted in any of the $n + 1$ intervals defined above.

According to Definition 2, when a new key is inserted, this key falls into anyone of the intervals with probability $\frac{1}{n+1}$. Uniformity of intervals means that we have the same probability to hit anyone of the $n + 1$ leaves, hence:

Lemma 3. *[EZG⁺82] Let us consider a fringe with n keys having X_t and Y_t leaves x and y respectively. The probability that, in a random insertion, a key hits an x node is $\frac{X_t}{n+1}$ and for an y node is $\frac{Y_t}{n+1}$.*

Hence, the conditional expectations when one random key is inserted satisfy

$$E_1(X_{t+1} \mid X_t, Y_t) = \frac{X_t}{n+1}(X_t - 2) + \frac{Y_t}{n+1}(X_t + 4) = \left(1 - \frac{2}{n+1}\right) X_t + \frac{4}{n+1} Y_t \ ,$$
$$E_1(Y_{t+1} \mid X_t, Y_t) = \frac{X_t}{n+1}(Y_t + 3) + \frac{Y_t}{n+1}(Y_t - 3) = \frac{3}{n+1} X_t + \left(1 - \frac{3}{n+1}\right) Y_t \ .$$

The expected number of leaves at step t can be modeled by the following definition:

Definition 4. [Yao78b,EZG⁺82,BY95] Given a fringe with $n + 1$ leaves and the sequential insertion algorithm, we define the 1-OneStep transition matrix $T_{n,1}$ as the matrix satisfying:

$$\begin{pmatrix} E_1(X_{t+1}) \\ E_1(Y_{t+1}) \end{pmatrix} = T_{n,1} \begin{pmatrix} E_1(X_t) \\ E_1(Y_t) \end{pmatrix} .$$

As the conditional expectations satisfy

$$\begin{aligned} E_1(X_{t+1}) &= E_1(E_1(X_{t+1} \mid X_t, Y_t)) \ , \\ E_1(Y_{t+1}) &= E_1(E_1(Y_{t+1} \mid X_t, Y_t)) \ . \end{aligned}$$

we recall the following lemma:

Lemma 5. *[EZG+82,BY95] The 1-OneStep transition matrix is:*

$$T^{Seq}_{n,1} = \left(1 + \frac{1}{n+1}\right) I + \frac{1}{n+1} \begin{pmatrix} -3 & 4 \\ 3 & -4 \end{pmatrix} \qquad \textit{where} \qquad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

In a more compact form, the 1-OneStep transition matrix can be rewritten as:

$$T^{\mathsf{Seq}}_{n,1} = \frac{1}{n+1} \begin{pmatrix} n-1 & 4 \\ 3 & n-2 \end{pmatrix} = \frac{n+2}{n+1} \left(I + \frac{1}{n+2} \begin{pmatrix} -3 & 4 \\ 3 & -4 \end{pmatrix} \right).$$

Later on we will give a direct proof of this Lemma. To consider the sequential insertion of k keys, we start with $k = 2$. In order to compare with the synchronized parallel insertion case we will develop expansions having powers of $1/(n+1)^j$.

Lemma 6. *The transition matrix of the sequential insertion of 2 keys is:*

$$\begin{aligned} T^{Seq}_{n,2} &= \left(1 + \frac{2}{n+1}\right) I + \left(\frac{6}{n+2} - \frac{4}{n+1}\right) \begin{pmatrix} -3 & 4 \\ 3 & -4 \end{pmatrix} \\ &= \left(1 + \frac{2}{n+1}\right) I + \frac{2}{n+1} \begin{pmatrix} -3 & 4 \\ 3 & -4 \end{pmatrix} - \frac{3}{2} \left(\frac{2}{n+1}\right)^2 \left(\sum_{j \geq 0} (-1)^j \frac{1}{(n+1)^j} \right) \begin{pmatrix} -3 & 4 \\ 3 & -4 \end{pmatrix}. \end{aligned}$$

Proof. The matrix $T^{\mathsf{Seq}}_{n,2}$ is the product of $T^{\mathsf{Seq}}_{n+1,1}$ and $T^{\mathsf{Seq}}_{n,1}$. The result follows from elementary calculation. □

The first terms are:

$$T^{\mathsf{Seq}}_{n,2} = \left(1 + \frac{2}{n+1}\right) I + \frac{2}{n+1} \begin{pmatrix} -3 & 4 \\ 3 & -4 \end{pmatrix} + \frac{3}{(n+1)^2} \begin{pmatrix} 6 & -8 \\ -6 & 8 \end{pmatrix} + \cdots$$

Finally, let us consider the matrix $T^{\mathsf{Seq}}_{n,k} = T_{n+k-1,1} \times \cdots \times T_{n,1}$ where $\times$ means matrix product, obtained iterating $k > 0$ times the one insertion case. Assuming fixed k, we obtain:

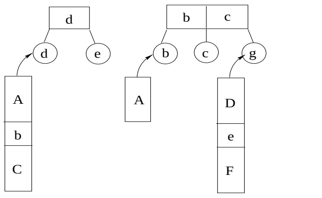
$$\begin{aligned} T^{\mathsf{Seq}}_{n,k} &= \left(1 + \frac{k}{n+1}\right) \left(I + \frac{1}{n+k+1} \begin{pmatrix} -3 & 4 \\ 3 & -4 \end{pmatrix} \right) \cdots \left(I + \frac{1}{n+2} \begin{pmatrix} -3 & 4 \\ 3 & -4 \end{pmatrix} \right) \\ &= \left(1 + \frac{k}{n+1}\right) I \\ &\quad + \left(1 + \frac{k}{n+1}\right) \left(\frac{1}{n+2} + \frac{1}{n+3} + \cdots + \frac{1}{n+k+1} \right) \begin{pmatrix} -3 & 4 \\ 3 & -4 \end{pmatrix} + O\left(\frac{1}{(n+1)^2}\right). \end{aligned}$$

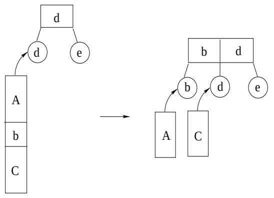
Since for fixed k and any $1 < i \leq k$

$$\frac{1}{n+i+1} = \frac{1}{n+1} + O\left(\frac{1}{(n+1)^2}\right)$$

we have

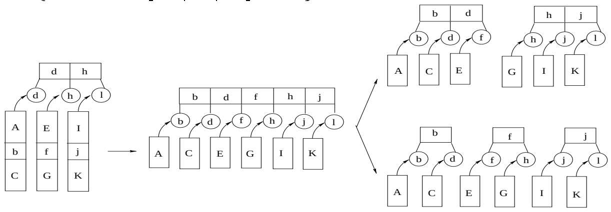
$$T^{\mathsf{Seq}}_{n,k} = \left(1 + \frac{k}{n+1}\right) I + \frac{k}{n+1} \begin{pmatrix} -3 & 4 \\ 3 & -4 \end{pmatrix} + O\left(\frac{1}{(n+1)^2}\right) \begin{pmatrix} -3 & 4 \\ 3 & -4 \end{pmatrix} \ .$$





(a) Left picture is $\{[A \mid b \mid C] \rightarrow d, e\}$ and the right one is $\{A \rightarrow b, c, [D \mid e \mid F] \rightarrow g\}$.

(b) Evolution rule when one packet is attached to an x node.



(c) Nondeterministic evolution rule when three packets are attached to a y node.

Fig. 3. Examples of evolution rules at the fringe level in the PVW algorithm.

3 Synchronized parallel insertion algorithms

It is assumed that an array of k sorted keys $a[1 \ldots k]$ is inserted into a 2-3 tree having n leaves. In order to define a random insertion of k keys we generalize Yao's sequential approach.

Definition 7. A 2–3 tree with n keys determines $n+1$ intervals. An insertion of a batch of k ordered keys $a[1 \ldots k]$ is *random* if each key $a[i]$, for $1 \leq i \leq k$, has equal probability of being inserted in any one of the $n+1$ intervals.

Notice that this random model is independent of particular insertion algorithms.

3.1 PVW algorithm

Now recall the algorithm of Paul, Vishkin, and Wagener [PVW83], and we introduce our MacroSplit algorithms. The algorithms first attach the keys to the leaves and later rebalance the tree. The PVW algorithm differs from the MacroSplit algorithms in the rebalancing phase.

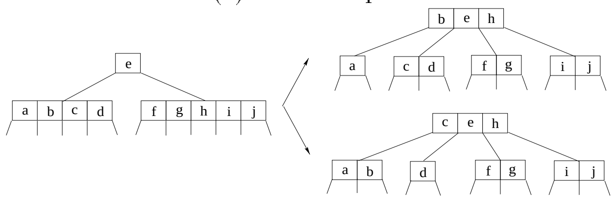
The tree is balanced in parallel using pipelines of processors. These pipelines can be envisaged intuitively in terms of traveling plane waves that are created at the bottom of the tree. Let's carefully explain the meaning of the waves.

Creation of waves: Waves are created by processes that synchronously apply the evolution rules at the fringe of the tree. These rules depend on the number of packets to be inserted and the type of nodes. In some cases there are different possibilities to split a node and a non-deterministic rule will be defined. In this case non-determinism means that any implementation of the PVW algorithm has to choose just one possibility.

We adopt some notations (see Figure 3(a)): $[a_1, \ldots, a_\ell]$ denotes a packet with keys $a_1, \ldots, a_\ell$ (note that usually $k \neq \ell$ where k is the total number of parallel inserted keys). If we need to distinguish the "middle key" we use: $[[a_1, \ldots, a_{\lfloor(\ell+1)/2\rfloor-1}] \mid a_{\lfloor(\ell+1)/2\rfloor} \mid [a_{\lfloor(\ell+1)/2\rfloor+1}, \ldots, a_\ell]]$. For instance, given a packet $[a, \ldots, g]$ we can write $[[a, b, c] \mid d \mid [e, f, g]]$. We can also write $[A \mid d \mid B]$ where the subpacket A contains the keys a, b, c and B contains e, f, g. The notation $[a_1, \ldots, a_\ell] \rightarrow h$ denotes that the packet is attached to leaf h. In the following we use lowercase letters for keys and uppercase letters for subpackets.

Let us consider separately each rule.

(a) The root splits.



(b) Two wide internal nodes split and keys go up.

Fig. 4. Examples of split rules at the internal nodes in the PVW algorithm.

- One packet is attached to an x node (see Figure 3(b)). This packet is split into two subpackets and a new key. This new key goes up and transforms the x node into a y node. This case generalizes to the first sequential transformation appearing in the Figure 2. We have the following evolution rule:

$$\begin{array}{ll} \text{Initial configuration: } \{[A \mid b \mid C] \to d, e\} \\ \text{Final configuration: } \{A \to b, C \to d, e\} \end{array}$$

  The same kind of rule can be written to deal with the initial configuration $\{a, [B \mid c \mid D] \to e\}$.
- Two packets are attached to an x node. Every subpacket is split, giving four subpackets and two new keys. The two new keys go up giving an unstable "wide node" with three keys. This wide node is "unstable" and splits into two x nodes. Each x leaf takes care of the corresponding subpacket:

$$\begin{array}{ll} \text{Initial configuration:} & \{[A \mid b \mid C] \to d, [E \mid f \mid G] \to h\} \\ \text{Unstable configuration:} & \{A \to b, C \to d, E \to f, G \to h\} \\ \text{Final configuration:} & \{A \to b, C \to d\}, \{E \to f, G \to h\} \end{array}$$

- One packet is attached to a y node. This packet is split into two subpackets and a new key. This new key goes up and transforms the y node into two x nodes. This case generalizes to the second sequential transformation appearing in Figure 2. Schematically we have:

$$\begin{array}{ll} \text{Initial configuration:} & \{[A \mid b \mid C] \to d, e, f\} \\ \text{Unstable configuration:} & \{A \to b, C \to d, e, f\} \\ \text{Final configuration:} & \{A \to b, C \to d\}, \{e, f\} \end{array}$$

  There are two similar cases $\{a, [B \mid c \mid D] \to e, f\}$, $\{a, b, [C \mid d \mid E] \to f\}$.
- Two packets are attached to a y node. Every subpacket is split giving four subpackets and two new keys. The two new keys go up, giving an unstable wide node having four keys. This wide node non-deterministically splits into one x node and one y node.

$$\begin{array}{ll} \text{Initial configuration:} & \{[A \mid b \mid C] \to d, [E \mid f \mid G] \to h, i\} \\ \text{Unstable configuration:} & \{A \to b, C \to d, E \to f, G \to h, i\} \\ \text{Final configuration:} & \{A \to b, C \to d\}, \{E \to f, G \to h, i\} \\ & \text{or } \{A \to b, C \to d, E \to f\}, \{G \to h, i\} \end{array}$$

  The following two cases are similar $\{[A \mid b \mid C] \to d, e, [F \mid g \mid H] \to i\}$, $\{a, [B \mid c \mid D] \to e, [F \mid g \mid H] \to i\}$.
- Three packets are attached to a y node (see Figure 3(c)). When packets split three new keys are generated. These keys go up, giving a wide node with five keys. This node splits non-deterministically into two y

nodes or into three x nodes. This case is schematized by:

$$\begin{array}{ll}
\text{Initial configuration:} & \{[A \mid b \mid C] \to d, [E \mid f \mid G] \to h, [I \mid j \mid K] \to \ell\} \\
\text{Unstable configuration:} & \{A \to b, C \to d, E \to f, G \to h, I \to j, K \to \ell\} \\
\text{Final configuration:} & \{A \to b, C \to d, E \to f\}, \{G \to h, I \to j, K \to \ell\} \\
 & \text{or } \{A \to b, C \to d\}, \{E \to f, G \to h\}, \{I \to j, K \to \ell\}
\end{array}$$

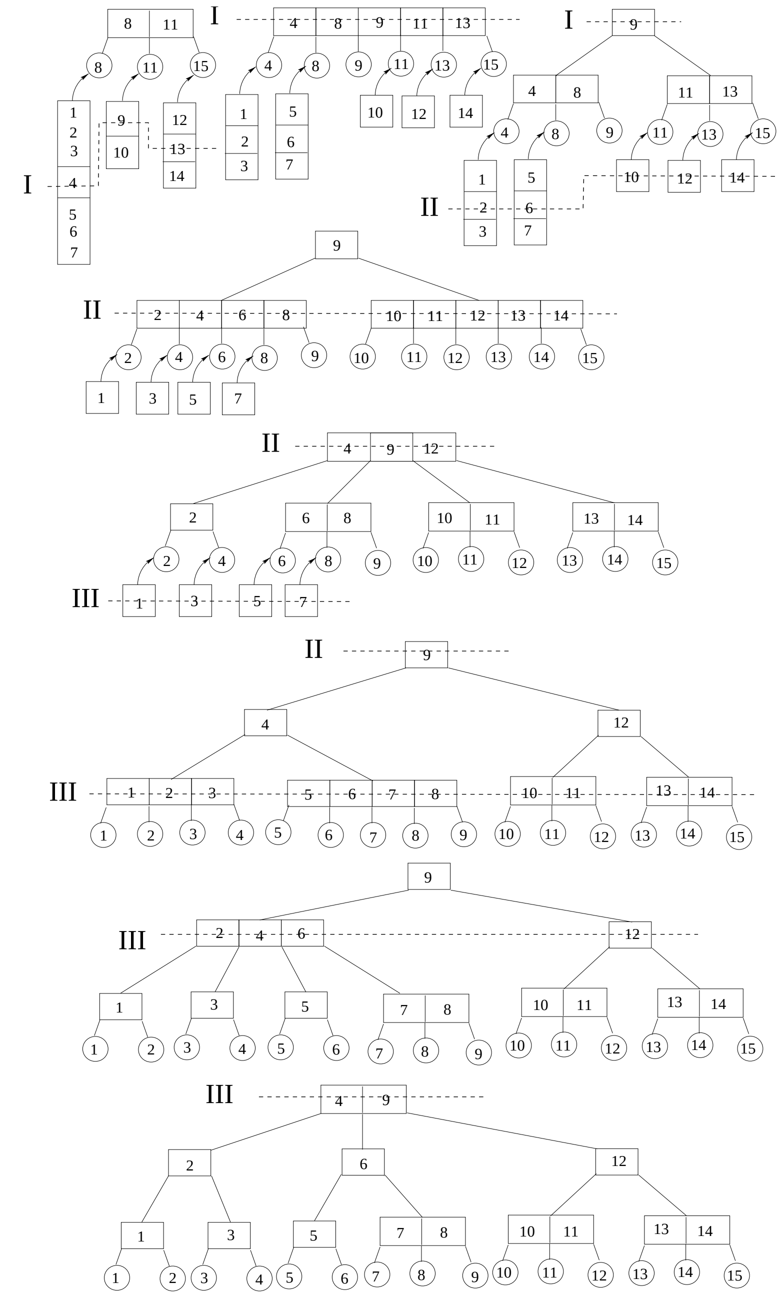


Fig. 5. An example in the PVW algorithm with three traveling waves. A wave is characterized by a "dotted line".

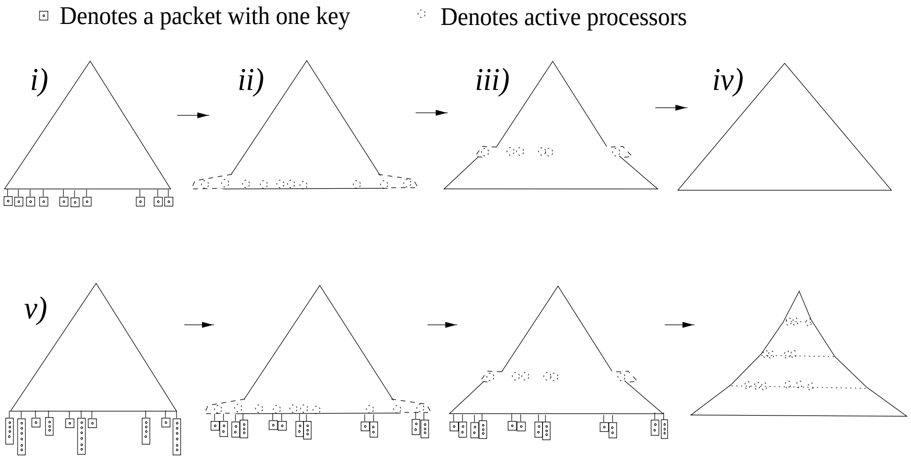


Fig. 6. Traveling waves for the PVW insertion algorithm on 2–3 trees.

Evolution of waves: Once the wave has been created it is sent up to the root. In this phase the processors apply the following evolution rules to wide internal nodes.

Let us consider the evolution of wide nodes while the perturbation goes up. There are two main cases:

- The root splits (Figure 4(a)). We have the following evolution rule:

$$\begin{array}{l} \text{Initial configuration: } \{[a \mid b \mid c \mid d \mid e]\} \\ \text{Final configuration: } \{[a \mid b] \rightarrow c, [d \mid e]\} \end{array}$$

  The same kind of rules can be written when the root contains 3 or 4 keys.
- Wide nodes splits (Figure 4(b)). We have the following evolution rule:

$$\begin{array}{ll} \text{Initial configuration:} & \{[a \mid b \mid c \mid d] \rightarrow e, [f \mid g \mid h \mid i \mid j]\} \\ \text{Final configuration:} & \{[a] \rightarrow b, [c \mid d] \rightarrow e, [f \mid g] \rightarrow h, [i \mid j]\} \\ & \text{or } \{[a \mid b] \rightarrow c, [d] \rightarrow e, [f \mid g] \rightarrow h, [i \mid j]\} \end{array}$$

  The same kind of rules can be written in all the other cases.

Example 8. Suppose that we insert into the tree $\{8, 11, 15\}$ the array

$$a[1..12] = [1, 2, 3, 4, 5, 6, 7, 9, 10, 12, 13, 14]$$

using the PVW algorithm. Initially, the keys are attached and we have the following initial configuration:

$$\{[1, 2, 3, 4, 5, 6, 7] \rightarrow 8, [9, 10] \rightarrow 11, [12, 13, 14] \rightarrow 15\}$$

The fringe evolves to:

$$\begin{array}{l} \{[1, 2, 3] \rightarrow, 4, [5, 6, 7] \rightarrow 8, 9\}, \{[10] \rightarrow 11, [12] \rightarrow 13, [14] \rightarrow 15\} \\ \{[1] \rightarrow 2, [3] \rightarrow 4\}, \{[5] \rightarrow 6, [7] \rightarrow 8, 9\}, \{10, 11, 12\}, \{13, 14, 15\} \\ \text{and } \ \{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8, 9\}, \{10, 11, 12\}, \{13, 14, 15\} \end{array}$$

which is the final configuration. □

Algorithm: Assume, for instance, the basic insertion case in which every leaf incorporates at most one new key (Figure 6.*i*). A wave of processors is generated at the bottom of the tree, namely a plane wave, because all leaves have the same depth (Figure 6.*ii*). This wave is sent up in further iterations (Figure 6.*iii*) until it

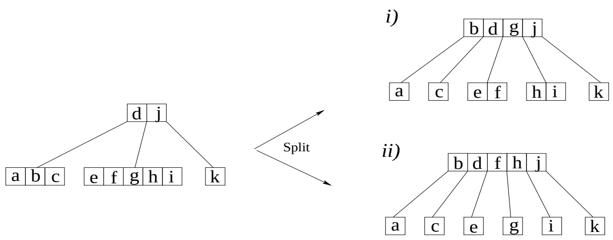


Fig. 7. Choices for MacroSplit rules. In (i) the rule makes the minimum number of splits. In (ii) the rule makes the maximum number of splits.

disappears (Figure 6.iv). In the general insertion case (Figure 6.v), in which a packet of many new keys can be attached to a single leaf, a pipeline of waves is generated to get periodic traveling waves. The creation of a new wave is delayed some steps in order to avoid concurrent accesses to nodes. Recall that the packets are split, the middle key of each one is attached as a new leaf and the remaining subpackets are attached to the leaves. This set of new leaves created by the middle keys constitute the new wave. Then, at most $O(\log k)$ waves are created and the time spent at each step is constant. So the parallel time to insert k keys becomes $O(\log n + \log k)$.

Figure 5 gives a graphical explanation of the rebalancing phase for Example 8. In this case three waves are created and sent up to the root.

3.2 MacroSplit algorithms

In the general insertion case (Figure 6.v) the MacroSplit algorithms incorporate simultaneously all the keys of packets at the bottom internal nodes of the tree creating only one wave. In successive steps the wave moves up until it reaches the root or disappears. Then the reconstruction is based in just *one unique* wave moving bottom up.

The evolution of this unique wave needs the use of the so called MacroSplit rules (see Figure 7). These rules determine the transformation of wide nodes into x and y nodes. For instance, the rule of case (i) of Figure 7 makes the minimum number of splits, and the rule of case (ii) makes the maximum number of splits. Intermediate strategies are possible. Let us see several examples. At most k keys can reach a node. If the node stores more than two keys, it must be split using a MacroSplit rule. Table 1 shows us several split possibilities for x and y bottom nodes. For instance, the first row shows us the splits of the x and y nodes when $k = 1$ (see Figure 2). In this case there is just one possibility. The fourth row shows us how x and y nodes can be split when $k = 4$. In this case a bottom x node can be split into three x nodes or into two y nodes. Later on we consider two extreme cases:

MaxMacroSplit algorithm: maximizes the number of splits at each step, then it maximizes also the number of x nodes created.

MinMacroSplit algorithm: minimizes the number of splits at each step, then it maximizes also the number of y nodes created.

When $k = 1$ or 2 both algorithms coincide (see Table 1).

The usage of MacroSplit rules increases the parallel running time, but it facilitates the fringe analysis of the MacroSplit algorithms. Suppose, for instance, that all the keys reach the same node, the PVW algorithm creates $\log k$ waves and the MacroSplit algorithm creates only one wave, but in the first case the time spent at each step is constant while the time spent in the second case is linear in k.

Let us introduce the analysis of the MacroSplit algorithm. Consider that at step $t+1$, k random insertions fall in parallel into a fringe with X_t leaves of x and Y_t leaves y such that $X_t + Y_t = n + 1$. Notice that $X_{t+1} + Y_{t+1} = n + k + 1$ and in general $X_{t+i} + Y_{t+i} = n + 1 + ki$. The expected values of X_{t+1} and Y_{t+1} after the insertion depend on two facts:

- The concrete form of the MacroSplit algorithm. This algorithm determines how many leaves of x and y will be generated at the bottom nodes when they receive some number of keys.

k	x node	y node
1	y	xx
2	xx	xy
3	xy	xxx or yy
4	xxx or yy	xxy
5	xxy	$xxxx$ or xyy
6	$xxxx$ or xyy	$xxxy$ or yyy

Table 1. MacroSplit possibilities for x and y bottom nodes when k keys are inserted.

– The current values of X_t and Y_t.

We deal with a Markov chain and the evolution can be analyzed through the so-called k-OneStep transition matrix $T_{n,k}$:

Definition 9. Given a fringe with $n+1$ leaves and a MacroSplit algorithm, we define the k-OneStep transition matrix $T_{n,k}$ as the matrix satisfying:

$$\begin{pmatrix} E_k(X_{t+1}) \\ E_k(Y_{t+1}) \end{pmatrix} = T_{n,k} \begin{pmatrix} E_k(X_t) \\ E_k(Y_t) \end{pmatrix} .$$

3.3 A first connection between both approaches

The MacroSplit algorithm can be seen as a "high level" description of the PVW algorithm. The PVW algorithm takes place by splitting a MacroSplit step into several more basic steps chained together in a pipeline. Then, the fringe analysis of the PVW algorithm must take into account all the waves of the pipeline while this same analysis for the MacroSplit algorithms takes into account only one wave.

The goal of this paper is to bound the evolution of the fringe of the PVW algorithm by the evolution of the MaxMacroSplit and MinMacroSplit algorithms. Consider the following lemma:

Lemma 10. *Let X_0, Y_0 be the initial values of the fringe. On the first step, k keys (not necessarily random), are inserted into this fringe using algorithm A. The values of the fringe at the end of the first step depend on A, therefore we denote these values by X_1^A, Y_1^A. If algorithm A is MaxMacroSplit, PVW or MinMacroSplit algorithm, the following holds*

$$X_1^{MaxMacroSplit} \geq X_1^{PVW} \geq X_1^{MinMacroSplit} ,$$
$$Y_1^{MaxMacroSplit} \leq Y_1^{PVW} \leq Y_1^{MinMacroSplit} .$$

We would like to generalize the preceding Lemma to more than one step, although a straightforward generalization is impossible as we see in the next example. It is possible to find a fringe X_0, Y_0 and 2 batches of (non random) k keys such that the generalization of Lemma 10 to two steps is false.

Example 11. Take as initial fringe $\{10, 20\}, \{30, 40\}, \{50, 60\}, \{70, 80\}$ and $X_0 = 8$ and $Y_0 = 0$. On this fringe we insert in parallel two batches of $k = 8$ keys using MaxMacroSplit, PVW and MinMacroSplit algorithms. The first batch is $12, 14, 16, 18, 22, 24, 26, 32$ and the second batch is $11, 15, 19, 23, 29, 39, 59, 79$. When the first search has been performed and packets have been attached to the leaves we have:

$$\{10, [12, 14, 16, 18] \to 20\}, \{[22, 24, 26] \to 30, [32] \to 40\}, \{50, 60\}, \{70, 80\}.$$

The MaxMacroSplit algorithm produces the fringe:

$$\{10, 12\}, \{14, 16\}, \{18, 20\}, \{22, 24\}, \{26, 30\}, \{32, 40\}, \{50, 60\}, \{70, 80\}$$

and $X_1^{\mathsf{MaxMacroSplit}} = 16$, $Y_1^{\mathsf{MaxMacroSplit}} = 0$. The PVW fringe evolves as:

$$\{10, [12] \to 14, [16, 18] \to 20\}, \{[22] \to 24, [26] \to 30\}, \{32, 40\}, \{50, 60\}$$
$$\{10, 12, 14\}, \{16, [18] \to 20\}, \{22, 24\}, \{26, 30\}, \{32, 40\}, \{50, 60\}$$
$$\{10, 12, 14\}, \{16, 18, 20\}, \{22, 24\}, \{26, 30\}, \{32, 40\}, \{50, 60\}, \{70, 80\},$$

and $X_1^{\mathsf{PVW}} = 10$, $Y_1^{\mathsf{PVW}} = 6$. The MinMacroSplit algorithm produces the fringe

$$\{10, 12, 14\}, \{16, 18, 20\}, \{22, 24, 26\}, \{30, 32, 40\}, \{50, 60\}, \{70, 80\}$$

and $X_1^{\mathsf{MinMacroSplit}} = 4$, $Y_1^{\mathsf{MinMacroSplit}} = 12$. Of course the inequalities given in Lemma 10 hold. Let us proceed with the second step. The set of keys to be inserted is $11, 15, 19, 23, 29, 39, 59, 79$. The MaxMacroSplit fringe evolves to:

$$\{10, [11] \to 12\}, \{14, [15] \to 16\}, \{18, [19] \to 20\}, \{22, [29] \to 24\},$$
$$\{26, [29] \to 30\}, \{32, [39] \to 40\}, \{50, [59] \to 60\}, \{70, 80\}$$
$$\{10, 11, 12\}, \{14, 15, 16\}, \{18, 19, 20\}, \{22, 23, 24\},$$
$$\{26, 29, 30\}, \{32, 39, 40\}, \{50, 59, 60\}, \{70, 79, 80\}$$

with $X_2^{\mathsf{MaxMacroSplit}} = 0$ and $Y_2^{\mathsf{MaxMacroSplit}} = 24$. The PVW fringe evolves non-deterministically to:

$$\{10, [11] \to 12, 14\}, \{[15] \to 16, [19] \to 18, 20\}, \{22, [23] \to 24\},$$
$$\{26, [29] \to 30\}, \{32, [39] \to 40\}, \{50, [59] \to 60\}, \{70, [79] \to 80\}$$
$$\{10, 11\}, \{12, 14\}, \{15, 16, 18\}, \{19, 20\}, \{22, 23, 24\},$$
$$\{26, 29, 30\}, \{32, 39, 40\}, \{50, 59, 60\}, \{70, 79, 80\}$$

with $X_2^{\mathsf{PVW}} = 6$ and $Y_2^{\mathsf{PVW}} = 18$. The MinMacroSplit fringe evolves:

$$\{10, [11] \to 12, 14\}, \{[15] \to 16, [19] \to 18, 20\}, \{22, [23] \to 24, 26\},$$
$$\{[29] \to 30, 32, [39] \to 40\}, \{50, [59] \to 60\}, \{70, [79] \to 80\}$$
$$\{10, 11\}, \{12, 14\}, \{15, 16, 18\}, \{19, 20\}, \{22, 23\}, \{24, 26\},$$
$$\{29, 30, 32\}, \{39, 40\}, \{50, 59, 60\}, \{70, 79, 80\}$$

with $X_2^{\mathsf{MinMacroSplit}} = 12$ and $Y_2^{\mathsf{MinMacroSplit}} = 12$.

We have found a fringe X_0, Y_0 and 2 batches of (non random) $k = 8$ keys such that the two following inequalities, are false:

$$X_2^{\mathsf{MaxMacroSplit}} \geq X_2^{\mathsf{PVW}} \geq X_2^{\mathsf{MinMacroSplit}},$$
$$Y_2^{\mathsf{MaxMacroSplit}} \leq Y_2^{\mathsf{PVW}} \leq Y_2^{\mathsf{MinMacroSplit}}.$$

□

Lemma 10 holds only for one step. Later on, we will extend this lemma to consecutive insertions of k keys taking into account the *expected* number of nodes. In fact, the main theorem of this paper (Theorem 42), extends Lemma 10 to any step by considering expected values.

4 Parallel insertion of 2 and 3 keys

In this section we compute $T_{n,2}$ and $T_{n,3}$ following directly the technique applied before to sequential insertions [EZG+82] and we discuss the viability of this approach.

$(\cdot,\cdot)$	$P(\cdot,\cdot)$	$E_2(X_{t+1}\vert X_t,Y_t,(\cdot,\cdot))$	$E_2(Y_{t+1}\vert X_t,Y_t,(\cdot,\cdot))$
(x,x)	$\frac{X_t}{n+1}\frac{2}{n+1}$	X_t+2	Y_t
(x_1,x_2)	$\frac{X_t}{n+1}\frac{X_t-2}{n+1}$	X_t-4	Y_t+6
(x,y)	$2\frac{X_t}{n+1}\frac{Y_t}{n+1}$	X_t+2	Y_t
(y,y)	$\frac{Y_t}{n+1}\frac{3}{n+1}$	X_t+2	Y_t
(y_1,y_2)	$\frac{Y_t}{n+1}\frac{Y_t-3}{n+1}$	X_t+8	Y_t-6

Table 2. Parallel insertion of two keys.

4.1 Direct extensions

First, let us consider the case $k=2$. We have only one MacroSplit algorithm (see Table 1). The expected number of leaves is characterized by the 2-OneStep $T_{n,2}$ transition matrix:

$$\begin{pmatrix} E_2(X_{t+1}) \\ E_2(Y_{t+1}) \end{pmatrix} = T_{n,2} \begin{pmatrix} E_2(X_t) \\ E_2(Y_t) \end{pmatrix}.$$

We compute the probabilities of the different splits by an exhaustive case analysis. As at most two keys can reach the same bottom node, the transformation of bottom nodes is unique (second row of Table 1). Both keys can be either at the same bottom node or at different bottom nodes, and in each case bottom nodes can be of type x or y. Let $P(x,x)$ be the probability that both keys reach the same x node, $P(x_1,x_2)$ the probability to reach different x nodes and so on for the remainder probabilities $P(x,y)$, $P(y,y)$ and $P(y_1,y_2)$. We denote the generic case as $P(\cdot,\cdot)$, with $(.,.)$ being the generic pair of nodes accessed.

Lemma 12. *Let us consider a fringe with n keys having X_t and Y_t leaves x and y respectively. The probability that, in a random insertion of a batch of 2 keys, these keys hit different x nodes, written $P(x_1,x_2)$ is:*

$$P(x_1,x_2) = \frac{X_t}{n+1}\frac{X_t-2}{n+1}$$

and the conditional expectation is $E_2(X_{t+1}|X_t,Y_t,(x_1,x_2)) = X_t - 4$. Table 2 contains the other values of probabilities and expectations.

The proof is direct from Definition 7. As $E_2(X_{t+1}) = E_2(E_2(X_{t+1} \mid X_t,Y_t))$ we compute the expected number of x leaves as

$$E_2(X_{t+1}|X_t,Y_t) = \sum_{(\cdot,\cdot)} P(\cdot,\cdot)\ E_2(X_{t+1}|X_t,Y_t,(.,.))\ ,$$

with $E_2(X_{t+1}|X_t,Y_t,(.,.))$ being the expected number of x leaves when 2 keys reach node $(\cdot,\cdot)$ conditioned on X_t and Y_t.

Lemma 13. *The conditional expectations satisfy*

$$E_2(X_{t+1} \mid X_t,Y_t) = \left(1 - \frac{4}{n+1} + \frac{12}{(n+1)^2}\right) X_t + \left(\frac{8}{n+1} - \frac{18}{(n+1)^2}\right) Y_t,$$
$$E_2(Y_{t+1} \mid X_t,Y_t) = \left(\frac{6}{n+1} - \frac{12}{(n+1)^2}\right) X_t + \left(1 - \frac{6}{n+1} + \frac{18}{(n+1)^2}\right) Y_t.$$

$(\cdot,\cdot)$	$P(\cdot,\cdot,\cdot)$	$E_3(X_{t+1}\vert X_t,Y_t,(\cdot,\cdot,\cdot))$	$E_3(Y_{t+1}\vert X_t,Y_t,(\cdot,\cdot,\cdot))$
(x,x,x)	$\frac{X_t}{n+1}\left(\frac{2}{n+1}\right)^2$	X_t	Y_t+3
(x_1,x_1,x_2)	$3\frac{X_t}{n+1}\frac{2}{n+1}\frac{X_t-2}{n+1}$	X_t	Y_t+3
(x_1,x_2,x_3)	$\frac{X_t}{n+1}\frac{X_t-2}{n+1}\frac{X_t-4}{t+1}$	X_t-6	Y_t+9
(x,x,y)	$3\frac{X_t}{n+1}\frac{2}{n+1}\frac{Y_t}{n+1}$	X_t+6	Y_t-3
(x,y,y)	$3\frac{X_t}{n+1}\frac{Y_t}{n+1}\frac{3}{n+1}$	X_t	Y_t+3
(x_1,x_2,y)	$3\frac{X_t}{n+1}\frac{X_t-2}{n+1}\frac{Y_t}{n+1}$	X_t	Y_t+3
(x,y_1,y_2)	$3\frac{X_t}{n+1}\frac{Y_t}{n+1}\frac{Y_t-3}{n+1}$	X_t+6	Y_t-3
(y_1,y_2,y_3)	$\frac{Y_t}{n+1}\frac{Y_t-3}{n+1}\frac{Y_t-6}{n+1}$	X_t+12	Y_t-9
(y_1,y_1,y_2)	$3\frac{Y_t}{n+1}\frac{3}{n+1}\frac{Y_t-3}{n+1}$	X_t+6	Y_t-3
(y,y,y)	$\frac{Y_t}{n+1}\left(\frac{3}{n+1}\right)^2$	X_t	Y_t+3

Table 3. Parallel insertion of three keys in the MinMacroSplit algorithm.

Proof. We compute the conditional expectation only for X_{t+1} (the Y_{t+1} term has a similar development). Then $E_2(X_{t+1}|X_t,Y_t)$ is:

$$\begin{aligned}
&\sum_{(\cdot,\cdot)} P(\cdot,\cdot)\ E_2(X_{t+1}|X_t,Y_t,(.\ ,.))\\
=&\frac{1}{(n+1)^2}\Big(2X_t(X_t+2)+X_t(X_t-2)(X_t-4)+2X_tY_t(X_t+2)\\
&\qquad\qquad +3Y_t(X_t+2)+Y_t(Y_t-3)(X_t+8)\Big)\\
=&X_t+\frac{1}{(n+1)^2}\Big(12X_t-4X_t^2+4X_tY_t+8Y_t^2-18Y_t)\Big)\\
=&\left(1-\frac{4}{n+1}+\frac{12}{(n+1)^2}\right)X_t+\left(\frac{8}{n+1}-\frac{18}{(n+1)^2}\right)Y_t\ .
\end{aligned}$$

□

As the conditional expectations are linear in X_t and Y_t and

$$E_2(X_{t+1}) = E_2(E_2(X_{t+1} \mid X_t,Y_t))\ ;$$
$$E_2(Y_{t+1}) = E_2(E_2(Y_{t+1} \mid X_t,Y_t))\ ,$$

we have:

Lemma 14. *The 2-OneStep transition matrix is:*

$$T_{n,2}=\left(1+\frac{2}{n+1}\right)I+\frac{2}{n+1}\begin{pmatrix}-3 & 4\\ 3 & -4\end{pmatrix}+\frac{1}{(n+1)^2}\begin{pmatrix}12 & -18\\ -12 & 18\end{pmatrix}\ .$$

Consider briefly the case $k = 3$. Now there are two possibilities (third row of Table 1). Table 3 contains the exhaustive case analysis of the probabilities; we have selected the second transformation that corresponds to the MinMacroSplit algorithm. The MaxMacroSplit case has a similar analysis but the final result is different.

Lemma 15. *When $k = 3$ there are two cases for the 3-OneStep transition matrix $T_{n,3}$.*

- *In the MaxMacroSplit algorithm, the matrix is:*

$$\left(1 + \frac{3}{n+1}\right) I + \frac{3}{n+1}\begin{pmatrix} -3 & 4 \\ 3 & -4 \end{pmatrix} + \frac{3}{(n+1)^2}\begin{pmatrix} 12 & -18 \\ -12 & 18 \end{pmatrix} + \frac{3}{(n+1)^3}\begin{pmatrix} -16 & 36 \\ 16 & -36 \end{pmatrix} .$$

- *In the MinMacroSplit algorithm the matrix is:*

$$\left(1 + \frac{3}{n+1}\right) I + \frac{3}{n+1}\begin{pmatrix} -3 & 4 \\ 3 & -4 \end{pmatrix} + \frac{3}{(n+1)^2}\begin{pmatrix} 12 & -18 \\ -12 & 18 \end{pmatrix} + \frac{1}{(n+1)^3}\begin{pmatrix} -48 & 54 \\ 48 & -54 \end{pmatrix} .$$

4.2 Discussion of the cases 2 and 3

Based on the preceding cases, it is clear that an exhaustive case analysis (generalizing the sequential approach [EZG+82]) for larger $k = 4, 5, 6, \ldots$, becomes lengthly and tedious. Notice also that the transition matrix for the parallel case for $k = 2$ and $k = 3$ only differ from the iterated sequential case shown at the end of Section 2 only in the $O(1/(n+1)^2)$.

In the sequel of the paper, we use *local behavior* of the algorithm to denote what happens when i keys hit just *one* bottom x or y node (Table 1). Similarly, we use *global behavior* to denote the evolution of X_t and Y_t. The previous exhaustive analysis does not distinguish between the *local* and the *global behavior* of the MacroSplit algorithm. Next, we take a different approach to solve the generic case for constant k.

5 Behavior of the MacroSplit algorithms

In order to study the expected behavior of an x or y node belonging to a fringe with $n+1$ leaves when k keys are inserted at a given step, we need to know the characteristics of the MacroSplit algorithm we are using.

5.1 Local behavior

We would like to know how many x and y leaves are generated when i keys fall at the same time into an x node or a y node. To deal with this question we introduce the following definition.

Definition 16. At the bottom level, the local behavior of the MacroSplit algorithm is given by the following functions:

- The function $\mathcal{X}_x(i)$ is the number of x leaves after the insertion of i keys into a unique x node (for instance, $\mathcal{X}_x(0) = 2$, $\mathcal{X}_x(1) = 0$, $\ldots$). In the same way, $\mathcal{X}_y(i)$ is the number of x leaves after the insertion of i keys into a y node (for instance, $\mathcal{X}_y(0) = 0$, $\mathcal{X}_y(1) = 4$, $\ldots$).
- Analogously, $\mathcal{Y}_x(i)$ is the number of y leaves after the insertion of i keys into an x node (for instance, $\mathcal{Y}_x(0) = 0$, $\mathcal{Y}_x(1) = 3$, $\ldots$). Similarly, $\mathcal{Y}_y(i)$ is the number of y leaves after the insertion of i keys into a y node (for instance, $\mathcal{Y}_y(0) = 3$, $\mathcal{Y}_y(1) = 0$, $\ldots$).

Assume that k random keys fall (in parallel) into a fringe having $n+1$ leaves. First of all, let us isolate just one bottom x node and one key to insert. Then, the new key can be inserted into the x node in two different positions. Therefore, according to Definition 7, just one key hits an x node with probability $\frac{2}{n+1}$. By a similar reasoning one key hits a y node with probability $\frac{3}{n+1}$. Now we consider what happens with an x or y node when k randomly selected keys are inserted.

Lemma 17. *Let N_x and N_y be the random variables denoting the number of keys falling into fixed bottom x and y nodes when k keys are inserted into the tree. Then, these variables follow a binomial distribution given by*

$$P\{N_x = i\} = b\Big(i, k, \frac{2}{n+1}\Big) \qquad \textit{and} \qquad P\{N_y = i\} = b\Big(i, k, \frac{3}{n+1}\Big),$$

such that $b\,(i, k, p) = \binom{k}{i} p^i \,(1-p)^{k-i}$.

Proof. Suppose that we have a fringe with $n+1$ keys having x and y nodes. On this fringe the k keys are inserted at random. Suppose that we "mark" or fix an x node belonging to the fringe. As each inserted key has the same probability to hit every leaf in the fringe the probability that i keys hit the x node is $b\big(i, k, \frac{2}{n+1}\big)$. A similar analysis can be done with a fixed y node. □

Note that N_x and N_y are not independent random variables as we see in the following example. Take a small fringe with $n+1 = 5$ having one x node and one y node. When k keys are inserted we have $N_x + N_y = k$. However these random variables help us to give precise meaning to the idea of the "expected behavior of just one node" as we will see in Lemma 18. Recall that the expected value of the binomial distribution is kp.

The number of x leaves generated by the keys falling into a unique x node is given by the random variable $X_x = \mathcal{X}_x(N_x)$ and the number of y leaves generated by the keys falling into a unique x node is $Y_x = \mathcal{Y}_x(N_x)$ (similarly for X_y and Y_y).

Lemma 18. *The expected number of leaves generated by one bottom node when a batch of k keys is inserted into a fringe having $n+1$ leaves is:*

$$E_k(X_x) = \sum_{i=0}^{k} b\Big(i, k, \frac{2}{n+1}\Big)\mathcal{X}_x(i)\ , \qquad E_k(Y_x) = \sum_{i=0}^{k} b\Big(i, k, \frac{2}{n+1}\Big)\mathcal{Y}_x(i)\ ,$$

$$E_k(X_y) = \sum_{i=0}^{k} b\Big(i, k, \frac{3}{n+1}\Big)\mathcal{X}_y(i)\ , \qquad E_k(Y_y) = \sum_{i=0}^{k} b\Big(i, k, \frac{3}{n+1}\Big)\mathcal{Y}_y(i)\ .$$

Proof.

$$E_k(X_x) = \sum_{i=0}^{k} P\{N_x = i\}\mathcal{X}_x(i) = \sum_{i=0}^{k} b\Big(i, k, \frac{2}{n+1}\Big)\mathcal{X}_x(i).$$

The other cases are argued similarly. □

Note that these expected values depend on the concrete local behavior of the algorithm.

Lemma 19. *The expected number of leaves generated by just one bottom node when k random keys are inserted in parallel into a fringe having $n+1$ leaves is:*

$$E_k(X_x + Y_x) = 2\Big(1 + \frac{k}{n+1}\Big) \qquad \textit{and} \qquad E_k(X_y + Y_y) = 3\Big(1 + \frac{k}{n+1}\Big).$$

Proof. The functions $\mathcal{X}_x(i)$, $\mathcal{Y}_x(i)$, $\mathcal{X}_y(i)$ and $\mathcal{Y}_y(i)$ satisfy $\mathcal{X}_x(i) + \mathcal{Y}_x(i) = 2 + i$ and $\mathcal{X}_y(i) + \mathcal{Y}_y(i) = 3 + i$, and adding the expected values given in Lemma 18 we get:

$$E_k(X_x + Y_x) = \sum_{i=0}^{k} b\Big(i, k, \frac{2}{n+1}\Big)(\mathcal{X}_x(i) + \mathcal{Y}_x(i)) = \sum_{i=0}^{k} b\Big(i, k, \frac{2}{n+1}\Big)(2 + i).$$

Similarly for the other case. □

5.2 Global behavior

We characterize the evolution of the fringe in terms of the preceding "local expectations" given in Lemma 18.

Lemma 20. *Given an n-key random tree T with a fringe having X_t leaves of x and Y_t leaves of y, when k keys are inserted at random into T in one step we have*

$$E_k(X_{t+1} \mid X_t, Y_t) = E_k(X_x)\frac{X_t}{2} + E_k(X_y)\frac{Y_t}{3} \,,$$
$$E_k(Y_{t+1} \mid X_t, Y_t) = E_k(Y_x)\frac{X_t}{2} + E_k(Y_y)\frac{Y_t}{3} \,.$$

Proof. Let us consider a fringe having X_t leaves of x and Y_t leaves of y and $X_t+Y_t = n+1$. On this fringe the number of x nodes is $X_t/2$ and the number of y nodes is $Y_t/3$, the total number of nodes is $X_t/2+Y_t/3 = \ell$. Let $(z_1, \dots, z_i, \dots, z_\ell)$ be the ordered enumeration of fringe's nodes (each z_i is either an x node or and y node). The probability for just one key to hit the node z_i is:

$$P\{z_i\} = \begin{cases} \frac{2}{n+1} & \text{the node } z_i \text{ is an } x \text{ node} \\ \frac{3}{n+1} & \text{the node } z_i \text{ is a } y \text{ node.} \end{cases}$$

Note that:

$$P\{z_1\} + P\{z_2\} + \cdots + P\{z_\ell\} = \frac{2}{n+1}\frac{X_t}{2} + \frac{3}{n+1}\frac{Y_t}{3} = 1.$$

The expected number of leaves generated by z_i when a batch of k keys are inserted is:

$$\mathcal{X}_{z_i}(j) = \begin{cases} \mathcal{X}_x(j) & \text{the node } z_i \text{ is an } x \text{ node,} \\ \mathcal{X}_y(j) & \text{the node } z_i \text{ is a } y \text{ node.} \end{cases}$$

When k keys are inserted in parallel, the number of keys falling into z_i called N_{z_i} follows:

$$P\{N_{z_i} = j\} = \binom{k}{j} P\{z_i\}^j (1 - P\{z_i\})^{k-j} = b(j, k, P\{z_i\}).$$

and the local expectation of $X_{z_i} = \mathcal{X}_{z_i}(N_{z_i})$ is given by:

$$E_k(X_{z_i}) = \sum_j P\{N_{Z_i} = j\}\mathcal{X}_{z_i}(j).$$

We will compute $E_k(X_{t+1}|X_t, Y_t)$ when k keys are inserted in parallel at random in the fringe $(z_1, \dots, z_i, \dots, z_\ell)$. The different possibilities can be partitioned in the set of different "macro-states" $(k_1, \dots, k_i, \dots, k_\ell)$ with $k_1 + \cdots + k_i + \cdots + k_\ell = k$. In this macro-state k_i is the number of keys falling into the node z_i and:

$$P\{N_{z_1} = k_1, \dots, N_{z_\ell} = k_\ell\} = \binom{k}{k_1, \dots, k_\ell} P\{z_1\}^{k_1} \cdots P\{z_\ell\}^{k_\ell}.$$

Taking the sum over all macro-states $(k_1, \dots, k_i, \dots k_\ell)$, the expectation is:

$$E_k(X_{t+1}|X_t, Y_t) = \sum P\{N_{z_1} = k_1, \dots, N_{z_\ell} = k_\ell\}(\mathcal{X}_{z_1}(k_1) + \cdots + \mathcal{X}_{z_\ell}(k_\ell)).$$

The first summand (corresponding to z_1) can be factorized as:

$$\sum P\{N_{z_1} = k_1, \dots, N_{z_\ell} = k_\ell\}\mathcal{X}_{z_1}(k_1) = \sum_{k_1} \binom{k}{k_1} P\{z_1\}^{k_1}\mathcal{X}_{z_1}(k_1) \sum_{k_2,\dots,k_\ell} \binom{k-k_1}{k_2, \dots, k_\ell} P\{z_2\}^{k_2} \cdots P\{z_\ell\}^{k_\ell}.$$

as $P\{z_2\} + \cdots + P\{z_\ell\} = 1 - P\{z_1\}$ we have:

$$\sum_{k_2,\ldots,k_\ell} \binom{k-k_1}{k_2,\ldots,k_\ell} P\{z_2\}^{k_2} \cdots P\{z_\ell\}^{k_\ell} = (1 - P\{z_1\})^{k-k_1}$$

as $P\{N_{z_i} = k_i\} = b(k_i, k, P\{z_i\})$ we can rewrite:

$$\begin{aligned}
\sum P\{N_{z_1} = k_1, \ldots, N_{z_\ell} = k_\ell\} \mathcal{X}_{z_1}(k_1) &= \sum_{k_1} \binom{k}{k_1} P\{z_1\}^{k_1} (1 - P\{z_1\})^{k-k_1} \mathcal{X}_{z_1}(k_1) \\
&= \sum_{k_1} P\{N_{z_1} = k_1\} \mathcal{X}_{z_1}(k_1) \\
&= E_k(X_{z_1}) ,
\end{aligned}$$

the preceding computation can be reproduced for any z_i obtaining:

$$\sum P\{N_{z_1} = k_1, \ldots, N_{z_\ell} = k_\ell\} \mathcal{X}_{z_i}(k_i) = \sum_{k_1} P\{N_{z_i} = k_i\} \mathcal{X}_{z_i}(k_i) = E_k(X_{z_i}).$$

Note that $z_i = x$ implies $E_k(X_{z_i}) = E_k(X_x)$ and $z_i = y$ implies $E_k(X_{z_i}) = E_k(X_y)$. As the nodes $(z_1, \ldots, z_\ell)$ are partitioned into $\frac{X_t}{2}$ x nodes and $\frac{Y_t}{3}$ y nodes the conditional expectation can be rewritten as:

$$E_k(X_{t+1}|X_t, Y_t) = E_k(X_{z_1}) + \cdots + E_k(X_{z_i}) + \cdots + E_k(X_{z_\ell}) = E_k(X_x)\frac{X_t}{2} + E_k(X_y)\frac{Y_t}{3}.$$

The computation for $E_k(Y_{t+1}|X_t, Y_t)$ is similar. □

Note that $P\{N_{z_1} = k_1, \ldots N_{z_\ell} = k_\ell\} \neq P\{N_{z_1} = k_1\} \times \cdots \times P\{N_{z_\ell} = k_\ell\}$ because $N_{z_1}, \ldots, N_{z_\ell}$ are not independent, but:

$$\sum_{k_1+\cdots+k_\ell=k} P\{N_{z_1} = k_1, \ldots, N_{z_\ell} = k_\ell\}(\mathcal{X}_{z_1}(k_1) + \cdots + \mathcal{X}_{z_\ell}(k_\ell))$$

coincides with:

$$\sum_{k_1=0}^{k} \cdots \sum_{k_\ell=0}^{k} P\{N_{z_1} = k_1\} \times \cdots \times P\{N_{z_\ell} = k_\ell\}(\mathcal{X}_{z_1}(k_1) + \cdots + \mathcal{X}_{z_\ell}(k_\ell)),$$

and the result of Lemma 20 "looks" like was obtained assuming independent random variables $N_{z_1}, \ldots, N_{z_\ell}$. However, this lemma does not use independence between $N_{z_1}, \ldots, N_{z_\ell}$ but the properties of the multinomial of $P\{N_{z_1} = k_1, \ldots, N_{z_\ell} = k_\ell\}$.

Lemma 21. *Given a fringe with $n+1$ leaves and a* MacroSplit *algorithm, the* k-OneStep *transition matrix is:*

$$T_{n,k} = \begin{pmatrix} \frac{1}{2}E_k(X_x) & \frac{1}{3}E_k(X_y) \\ \frac{1}{2}E_k(Y_x) & \frac{1}{3}E_k(Y_y) \end{pmatrix} .$$

Proof. From the preceding lemma we have

$$E_k(X_{t+1} \mid X_t, Y_t) = E_k(X_x)\frac{X_t}{2} + E_k(X_y)\frac{Y_t}{3}.$$

As $E_k(X_{t+1}) = E_k(E_k(X_{t+1} \mid X_t, Y_t))$ we have

$$E_k(X_{t+1}) = \frac{1}{2}E_k(E_k(X_x)X_t) + \frac{1}{3}E_k(E_k(X_y)Y_t).$$

As the expectation is a linear operator, $E_k(E_k(X_x)X_t) = E_k(X_x)E_k(X_t)$. □

Example 22. Let us recompute the 1-OneStep transition matrix using Lemma 21. Let us start with a bottom x node. As we have seen in the definition 16 we have $\mathcal{X}_x(0) = 2$ and $\mathcal{X}_x(1) = 0$.

$$E_1(X_x) = \left(1 - \frac{2}{n+1}\right) \mathcal{X}_x(0) + \left(\frac{2}{n+1}\right) \mathcal{X}_x(1) = \frac{2}{n+1}(n-1).$$

Using the property $E_1(X_x + Y_x) = 2(1 + \frac{1}{n+1})$ given in the Lemma 19 we get

$$E_1(Y_x) = \frac{2}{n+1}\, 3.$$

Let us consider a bottom y node, with $\mathcal{X}_y(0) = 0$ and $\mathcal{X}_y(1) = 4$.

$$E_1(X_y) = \left(1 - \frac{3}{n+1}\right) \mathcal{X}_y(0) + \left(\frac{3}{n+1}\right) \mathcal{X}_y(1) = \frac{12}{n+1}\ .$$

Using $E_1(X_y + Y_y) = 3(1 + \frac{1}{n+1})$ we get

$$E_1(Y_y) = \frac{3}{n+1}(n-2).$$

Substituting we get

$$T_{n,1} = \begin{pmatrix} \frac{1}{2}E_1(X_x) & \frac{1}{3}E_1(X_y) \\ \frac{1}{2}E_1(Y_x) & \frac{1}{3}E_1(Y_y) \end{pmatrix} = \frac{1}{n+1}\begin{pmatrix} n-1 & 4 \\ 3 & n-2 \end{pmatrix}\ ,$$

which is the same result of the sequential case (see Lemma 3).

5.3 Power expansion of the transition matrix

In the last section we have proved that the transition matrix is determined by the expected local behavior of the MacroSplit algorithms (Lemma 21). Now we give a series expansion of the k-OneStep transition matrix ($T_{n,k}$, similar to the series expansions of the transition matrices $T_{n,2}$ and $T_{n,3}$ given Lemmas 14 and 15.

Lemma 23. *Let I be the two dimensional identity matrix. The k-OneStep matrix satisfies:*

$$T_{n,k} = \left(1 + \frac{k}{n+1}\right) I + \begin{pmatrix} -\frac{1}{2}E_k(Y_x) & \frac{1}{3}E_k(X_y) \\ \frac{1}{2}E_k(Y_x) & -\frac{1}{3}E_k(X_y) \end{pmatrix}.$$

Proof. Lemma 19 relates the expected values of the matrix expression $T_{n,k}$ given in Lemma 21. □

In order to get the power expansion, let us recall the *binomial transform* ("Trick 3: Inversion", section 5.2 of Graham, Knuth and Patashnik [GKP94]). The binomial transform has been applied by Poblete, Munro, and Papadakis [PMP95] to the analysis of skip lists, we adapt the notations of that paper. Let $\langle F_i \rangle_{i \geq 0}$ be a sequence of real numbers, the binomial transform is the sequence $\langle \hat{F}_j \rangle_{j \geq 0}$ defined as

$$\hat{F}_j = \mathcal{B}_j \langle F_i \rangle = \sum_{i=0}^{j} (-1)^i \binom{j}{i} F_i.$$

This transform is related to well-known Poissonization techniques (for instance, Rice's method). If we consider the exponential generating functions $F(z)$ and $\hat{F}(z)$ of the sequences $\langle F_i \rangle$ and $\langle \hat{F}_j \rangle$, then $\hat{F}(z) = e^z F(z)$ and the information on $\hat{F}(z)$ can be translated back to $F(z)$ and its coefficients F_i. This transformation satisfies the following lemmas:

Lemma 24. *1. When $F_i = 1$ we have $\hat{F}_0 = 1$ and $\hat{F}_j = 0$ otherwise.*

2. *When* $F_i = (-1)^i$ *we have* $\hat{F}_j = 2^j$.
3. *When* $F_i = i$ *we have* $\hat{F}_1 = -1$ *and* $\hat{F}_j = 0$ *otherwise.*

Lemma 25. *Let* $\langle F_i\rangle_{i\geq 0}$ *and* $\langle G_i\rangle_{i\geq 0}$ *be sequences of real numbers and* a, b *real numbers, then the following holds:*

1. $F_i = \mathcal{B}_i\langle \hat{F}_j\rangle$.
2. $\mathcal{B}_j\langle a\,F_i + b\,G_i\rangle = a\,\mathcal{B}_j\langle F_i\rangle + b\,\mathcal{B}_j\langle G_i\rangle$.
3. *For* $j > 0$ *we have* $-\hat{F}_j = \mathcal{B}_{j-1}\langle F_{i+1}\rangle - \hat{F}_{j-1}$.
4. $\hat{F}_j = \sum_{\ell=0}^{k}(-1)^\ell\binom{k}{\ell}\mathcal{B}_{j-k}\langle F_{i+\ell}\rangle$ *for* $j \geq k$.
5. *Given* $p+q = 1$ *and* $\langle F_i\rangle_{i\geq 0}$ *we can define* $\sum_i \binom{\ell}{i}p^i q^{\ell-i}F_i = b(i,\ell,p)F_i$ *and get the sequence* $\langle b(i,\ell,p)F_i\rangle_{\ell\geq 0}$. *Then* $\mathcal{B}_j\langle b(i,\ell,p)F_i\rangle = p^j\,\hat{F}_j$.

Proof. We give a proof of item 4, other items are well known [GKP94,PMP95]. In the definition of $\hat{F}_j$ we replace: $\binom{j}{i} = \sum_{\ell=0}^{k}\binom{k}{\ell}\binom{j-k}{i-\ell}$ and:

$$\hat{F}_j = \sum_{i=0}^{j}(-1)^i\sum_{\ell=0}^{k}\binom{k}{\ell}\binom{j-k}{i-\ell}F_i.$$

Changing the order of the two summands and using that $\binom{j-k}{i-\ell} > 0$ only if $i \geq \ell$, we obtain:

$$\hat{F}_j = \sum_{\ell=0}^{k}\binom{k}{\ell}\sum_{i=\ell}^{j}(-1)^i\binom{j-k}{i-\ell}F_i.$$

Changing the scale from $\binom{j-k}{i-\ell}$ to $\binom{j-k}{i}$ we get:

$$\hat{F}_j = \sum_{\ell=0}^{k}\binom{k}{\ell}\sum_{i=0}^{j-\ell}(-1)^{i+\ell}\binom{j-k}{i}F_{i+\ell}.$$

As $\binom{j-k}{i} > 0$ only if $i \leq j-k$ we obtain:

$$\hat{F}_j = \sum_{\ell=0}^{k}\binom{k}{\ell}\sum_{i=0}^{j-k}(-1)^{i+\ell}\binom{j-k}{i}F_{i+\ell}.$$

□

In the following we use a weighted form of the binomial transforms of $\langle\mathcal{Y}_x(i)\rangle_{i\geq 0}$ and $\langle\mathcal{X}_y(i)\rangle_{i\geq 0}$ defining $\alpha_j = -2^{j-1}\hat{\mathcal{Y}}_x(j)$ and $\beta_j = -3^{j-1}\hat{\mathcal{X}}_y(j)$. Let us develop the relationship of the preceding coefficients with the local expected values of the k-OneStep transitions:

Lemma 26.

$$E_k(Y_x) = -2\sum_{j=0}^{k}\frac{(-1)^j}{(n+1)^j}\binom{k}{j}\alpha_j, \quad E_k(X_y) = -3\sum_{j=0}^{k}\frac{(-1)^j}{(n+1)^j}\binom{k}{j}\beta_j.$$

Proof. Recall that

$$E_k(Y_x) = \sum_{i=0}^{k}P\{X_x = i\}\mathcal{Y}_x(i) = \sum_{i=0}^{k}\binom{k}{i}\Big(\frac{2}{n+1}\Big)^i\Big(1-\frac{2}{n+1}\Big)^{k-i}\mathcal{Y}_x(i).$$

Consider the sequence $\langle E_k(Y_x)\rangle_{k\geq 0}$, by Lemma 25.5:

$$\hat{E}_j(Y_x) = \mathcal{B}_j\langle E_k(Y_x)\rangle = \Big(\frac{2}{n+1}\Big)^j\hat{\mathcal{Y}}_x(j)\ .$$

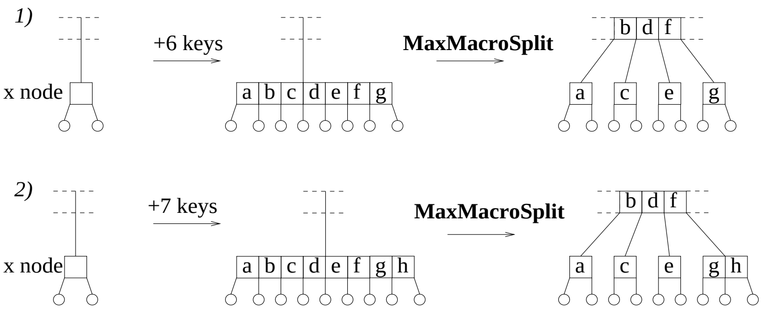


Fig. 8. Application of the MaxMacroSplit algorithm on an x node.

Now we apply Lemma 25.1 of the binomial transform,

$$E_k(Y_x) = \mathcal{B}_k\langle \hat{E}_j(Y_x)\rangle = \mathcal{B}_k\left\langle \left(\frac{2}{n+1}\right)^j \hat{\mathcal{Y}}_x(j)\right\rangle \ .$$

Using linearity (Lemma 25.2) we have

$$E_k(Y_x) = 2\mathcal{B}_k\left\langle \left(\frac{1}{n+1}\right)^j 2^{j-1}\hat{\mathcal{Y}}_x(i)\right\rangle = -2\sum_{j=0}^{k}(-1)^j\binom{k}{j}\frac{\alpha_j}{(n+1)^j} \ .$$

The case $E_k(X_y)$ is similar. □

By direct computation $\alpha_0 = \beta_0 = 0$, and from Lemmas 23 and 26, we get the following expansion:

Theorem 27. *The k-OneStep transition matrix can be rewritten as*

$$T_{n,k} = \left(1+\frac{k}{n+1}\right) I + \sum_{j=1}^{k}\frac{(-1)^j}{(n+1)^j}\binom{k}{j}\begin{pmatrix}\alpha_j & -\beta_j\\ -\alpha_j & \beta_j\end{pmatrix}.$$

6 Two extreme MacroSplit algorithms

We have shown that the k-OneStep transition matrix depends on the concrete MacroSplit algorithm. In this section we develop two extreme cases of this algorithm: one denoted MaxMacroSplit algorithm that makes the maximum number of splits and creates the maximum number of x nodes, and another denoted MinMacroSplit algorithm that makes the minimum number of splits and creates the maximum number of y nodes. These two extreme cases bound the behavior of the PVW algorithm.

6.1 The MaxMacroSplit algorithm

Assume that an even i number of keys are attached to an x node ($i = 6$ in case 1 of Figure 8). This wide node splits yielding $i+2$ x leaves (8 in the preceding case) and no y leaves. Then $\mathcal{X}_x(i) = i+2$ and $\mathcal{Y}_x(i) = 0$. On the other hand, when an odd number i of keys are attached ($i = 7$ in case 2 of Figure 8), the split only creates one y node. Then $\mathcal{Y}_x(i) = 3$ and $\mathcal{X}_x(i) = i-1$ (3 and 6 respectively in the figure). Note that $\mathcal{X}_x(i) + \mathcal{Y}_x(i) = i+2$. We summarize the previous paragraph into the following lemma.

Lemma 28. *The* local behavior *of the MaxMacroSplit algorithm is given by:*

- *For even i we have $\mathcal{X}_x(i) = i+2$, $\mathcal{Y}_x(i) = 0$, $\mathcal{X}_y(i) = i$, $\mathcal{Y}_y(i) = 3$.*
- *For odd i we have $\mathcal{X}_x(i) = i-1$, $\mathcal{Y}_x(i) = 3$, $\mathcal{X}_y(i) = i+3$, $\mathcal{Y}_y(i) = 0$.*

The following lemma summarizes the expected local behavior.

Lemma 29. *The* expected local behavior *of the MaxMacroSplit algorithm is*

$$\begin{aligned} E_k(X_x) &= kp + \frac{1}{2} + \frac{3}{2}(q-p)^k, \quad E_k(Y_x) = \frac{3}{2} - \frac{3}{2}(q-p)^k, \quad \textit{with} \quad p = \frac{2}{n+1}, \\ E_k(X_y) &= kp + \frac{3}{2} - \frac{3}{2}(q-p)^k, \quad E_k(Y_y) = \frac{3}{2} + \frac{3}{2}(q-p)^k, \quad \textit{with} \quad p = \frac{3}{n+1}, \end{aligned}$$

q being $1-p$ in both cases.

Proof. First, let us consider the case $p = \frac{2}{n+1}$. The expected local behavior of X_x is given by

$$E_k(X_x) = \sum_{i=0}^{k} b(i,k,p)\mathcal{X}_x(i).$$

As $\mathcal{X}_x(i)$ depends on the parity of i (previous lemma), we define the following two functions

$$F_0(k,p) = \sum_{i \geq 0} b(2i,k,p) \quad \text{and} \quad F_1(k,p) = \sum_{i \geq 0} b(2i+1,k,p) \ .$$

The expected value of X_x becomes $E_k(X_x) = 2F_0(k,p) - F_1(k,p) + kp$. As $\binom{k}{i} = \binom{k-1}{i-1} + \binom{k-1}{i}$, writing $q = 1-p$, the functions F_0 and F_1 satisfy:

$$\begin{aligned} F_0(k,p) &= qF_0(k-1,p) + pF_1(k-1,p), \\ F_1(k,p) &= pF_0(k-1,p) + qF_1(k-1,p). \end{aligned}$$

with $F_0(0,p) = 1$ and $F_1(0,p) = 0$. Note that $F_0(k,p) + F_1(k,p) = 1$, therefore $F_0(k,p)$ and $F_1(k,p)$ act as probabilities and we deal with a Markov chain having transition matrix $P = \begin{pmatrix} q & p \\ p & q \end{pmatrix}$ such that:

$$\begin{pmatrix} F_0(k,p) \\ F_1(k,p) \end{pmatrix} = P^k \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

In order to compute P^k we diagonalize. The matrix P has eigenvalues 1 and $q-p$ and eigenvectors $(1,1)$ and $(-1,1)$ respectively. Let M be the matrix having as rows the eigenvectors

$$M = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}.$$

The matrix P is diagonalized as

$$P = M^{-1} \begin{pmatrix} 1 & 0 \\ 0 & q-p \end{pmatrix} M$$

and

$$\begin{pmatrix} F_0(k,p) \\ F_1(k,p) \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & q-p \end{pmatrix}^k \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Finally (example (a), section 4, chapter XI of Feller [Fel68]), for $k \geq 0$ we have

$$F_0(k,p) = \frac{1}{2}\left(1 + (q-p)^k\right) \quad \text{and} \quad F_1(k,p) = \frac{1}{2}\left(1 - (q-p)^k\right).$$

From the values of F_0 and F_1 we compute $E_k(X_x)$. As $p = \frac{2}{n+1}$, the expected value of Y_x can be computed directly from Lemma 19:

$$E_k(Y_x) = 2\Big(1 + \frac{k}{n+1}\Big) - E_k(X_x).$$

Let us consider the computation of the expected values of X_y and Y_y. The first expected value satisfies $E_k(X_y) = 3F_1(k,p) + kp$ therefore substituting the value of F_1 we get the result. Using $p = \frac{3}{n+1}$ and the equality

$$E_k(Y_y) = 3\Big(1 + \frac{k}{n+1}\Big) - E_k(X_y)$$

we compute the expected value of Y_y. □

Lemma 30. *The coefficients of the* power expansion *of the* MaxMacroSplit *algorithm satisfies:*

- *For $j > 0$ we have $\alpha_j = 3 \cdot 4^{j-1}$.*
- *The β_js are $\beta_1 = 4$ and $\beta_j = 3 \cdot 6^{j-1}$ for $j > 1$.*

Proof. First we prove the values for α_j. Using Lemmas 2 and 3 we have

$$-\hat{\mathcal{Y}}_x(j) = -\mathcal{B}_j\langle \mathcal{Y}_x(i)\rangle = \mathcal{B}_{j-1}\langle \mathcal{Y}_x(i+1)\rangle - \hat{\mathcal{Y}}_x(j-1) = \mathcal{B}_{j-1}\langle \mathcal{Y}_x(i+1) - \mathcal{Y}_x(i)\rangle.$$

By Lemma 28, $\mathcal{Y}_x(i+1) - \mathcal{Y}_x(i) = (-1)^i \cdot 3$ then $\hat{\mathcal{Y}}_x(j) = -3 \cdot 2^{j-1}$. As $\alpha_j = -2^{j-1}\hat{\mathcal{Y}}_x(j)$ then $\alpha_j = 3 \cdot 4^{j-1}$. Consider now the computation of β_j. Transforming $\mathcal{X}_y(i) = i + \mathcal{Y}_x(i)$ we obtain:

$$\hat{\mathcal{X}}_y(j) = \mathcal{B}_j\langle i\rangle + \hat{\mathcal{Y}}_x(j).$$

As $\mathcal{B}_1\langle i\rangle = -1$ substituting $\hat{\mathcal{Y}}_x(1)$ we get $\hat{\mathcal{X}}_y(1) = -4$ and $\beta_1 = 4$. For $j > 1$, as $\mathcal{B}_j\langle i\rangle = 0$ we obtain $\hat{\mathcal{X}}_y(j) = -3 \cdot 2^{j-1}$. As $\beta_j = -3^{j-1}\hat{\mathcal{X}}_y(j)$, we obtain the results. □

Lemma 31. *The* k-OneStep *transition matrix of* MaxMacroSplit *algorithm is:*

$$T_{n,k} = \Big(1 + \frac{k}{n+1}\Big) I + \frac{k}{n+1}\begin{pmatrix} -3 & 4 \\ 3 & -4 \end{pmatrix} + \frac{3}{2}\sum_{j>1}\Big(-\frac{2}{n+1}\Big)^j \binom{k}{j} \begin{pmatrix} 2^{j-1} & -3^{j-1} \\ -2^{j-1} & 3^{j-1} \end{pmatrix}.$$

Proof. Substituting into Theorem 27 the values of α_j and β_j given in Lemma 30 we obtain the result. □

For $k = 1, 2$ we obtain the Lemmas 5 and 14 respectively. For $k = 3$ we obtain the first part of Lemma 15.

6.2 The MinMacroSplit algorithm

Figure 9 pictures the split of an x node when six, seven and eight keys are attached. The MinMacroSplit algorithm has the following characterization:

Lemma 32. *The* local behavior *of the* MinMacroSplit *algorithm is given by:*

- *For $i \bmod 3 = 0$ we have $\mathcal{X}_x(i) = 2$, $\mathcal{Y}_x(i) = i$, $\mathcal{X}_y(i) = 0$, $\mathcal{Y}_y(i) = i + 3$.*
- *For $i \bmod 3 = 1$ we have $\mathcal{X}_x(i) = 0$, $\mathcal{Y}_x(i) = i + 2$, $\mathcal{X}_y(i) = 4$, $\mathcal{Y}_y(i) = i - 1$.*
- *For $i \bmod 3 = 2$ we have $\mathcal{X}_x(i) = 4$, $\mathcal{Y}_x(i) = i - 2$, $\mathcal{X}_y(i) = 2$, $\mathcal{Y}_y(i) = i + 1$.*

In the following, we use the next two functions:

$$\phi = \Re\left(\frac{2 - 3p + p\sqrt{3}\mathbf{i}}{2}\right)^k \qquad \text{and} \qquad \varphi = \sqrt{3}\,\Im\left(\frac{2 - 3p + p\sqrt{3}\mathbf{i}}{2}\right)^k.$$

Lemma 33. *The expected local behavior of the* MinMacroSplit *is determined by:*

$$\begin{aligned} E_k(X_x) &= 2 - \frac{4}{3}\varphi\,, & E_k(Y_x) &= pk + \frac{4}{3}\varphi\,, & &\text{for } p = \frac{2}{n+1}\,, \text{ and} \\ E_k(X_y) &= 2 - 2\phi + \frac{2}{3}\varphi\,, & E_k(Y_y) &= pk + 1 + 2\phi - \frac{2}{3}\varphi\,, & &\text{for } p = \frac{3}{n+1}\,. \end{aligned}$$

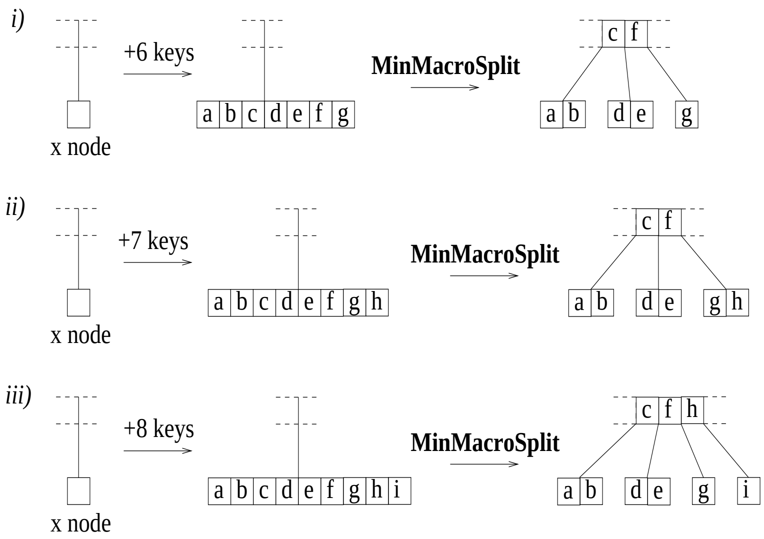


Fig. 9. Application of the MinMacroSplit algorithm.

Proof. As $\mathcal{X}_x(i)$ depends on the value of $i \mod 3$ we define the functions

$$F_0(k,p) = \sum_{i\geq 0} b(3i,k,p) \ , \qquad F_1(k,p) = \sum_{i\geq 0} b(3i+1,k,p) \ , \text{ and}$$
$$F_2(k,p) = \sum_{i\geq 0} b(3i+2,k,p) \ .$$

The expected values can be rewritten using these functions as:

$$E_k(X_x) = 2F_0(k,p) + 4F_2(k,p) \ , \qquad E_k(X_y) = 4F_1(k,p) + 2F_2(k,p) \ ,$$
$$E_k(Y_x) = 2F_1(k,p) - 2F_2(k,p) + kp \ , \qquad E_k(Y_y) = 3F_0(k,p) - F_1(k,p) + kp \ .$$

Now we compute the value of these functions. As $\binom{k}{i} = \binom{k-1}{i-1} + \binom{k-1}{i}$ then

$$\begin{aligned} F_0(k,p) =& qF_0(k-1,p) + pF_1(k-1,p) \ , \\ F_1(k,p) =& pF_0(k-1,p) + qF_1(k-1,p) \ , \\ F_2(k,p) =& pF_1(k-1,p) + qF_2(k-1,p) \ . \end{aligned}$$

with $F_0(0,p) = 1$ and $F_1(0,p) = F_2(0,p) = 0$. We deal with a Markov chain whose transition matrix is

$$P = \begin{pmatrix} q & 0 & p \\ p & q & 0 \\ 0 & p & q \end{pmatrix}$$

with eigenvalues

$$1, \ 1 - \frac{1}{2}p(3-\sqrt{3}\mathbf{i}), \ 1 - \frac{1}{2}p(3+\sqrt{3}\mathbf{i}),$$

and eigenvectors

$$(1,1,1), \ \Big(\ \frac{1}{2}(-1-\sqrt{3}\mathbf{i}) \ , \ \frac{1}{2}(-1+\sqrt{3}\mathbf{i}) \ , \ 1 \ \Big), \ \Big(\ \frac{1}{2}(-1-\sqrt{3}\mathbf{i}) \ , \ \frac{1}{2}(-1+\sqrt{3}\mathbf{i}) \ , \ 1 \ \Big).$$

Let M be the matrix of the eigenvectors, that is,

$$M = \begin{pmatrix} 1 & 1 & 1 \\ \frac{1}{2}(-1-\sqrt{3}\mathbf{i}) & \frac{1}{2}(-1+\sqrt{3}\mathbf{i}) & 1 \\ \frac{1}{2}(-1+\sqrt{3}\mathbf{i}) & \frac{1}{2}(-1-\sqrt{3}\mathbf{i}) & 1 \end{pmatrix}$$

and $\overline{M}$ the complex conjugate matrix of M. As

$$P^k = \overline{M}^{\,-1} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1-\frac{1}{2}p(3-\sqrt{3}\mathbf{i}) & 0 \\ 0 & 0 & 1-\frac{1}{2}p(3+\sqrt{3}\mathbf{i}) \end{pmatrix}^k \overline{M},$$

and after a tedious computation, taking ϕ and φ as defined in the lemma, we obtain for $k \geq 0$:

$$F_0(k,p) = \frac{1}{3} + \frac{2}{3}\phi\ , \qquad F_1(k,p) = \frac{1}{3} - \frac{1}{3}(\phi - \varphi)\ , \qquad F_2(k,p) = \frac{1}{3} - \frac{1}{3}(\phi + \varphi)\ .$$

□

Example 34. Let us compute more precise expressions for the expected behavior of the MinMacroSplit algorithms when $k = 3$. This will allow us to recover the expression for $T_{n,3}$.

$$\phi = \Re\left(\frac{2-3p+p\sqrt{3}\mathbf{i}}{2}\right)^3 = \frac{1}{2}\left(2 - 9\,p + 9\,p^2\right)\ ,$$

$$\varphi = \sqrt{3}\,\Im\left(\frac{2-3p+p\sqrt{3}\mathbf{i}}{2}\right)^3 = \frac{9}{2}\,p\left(1 - 3\,p + 2\,p^2\right)\ .$$

As $E_3(X_x) = 2 - \frac{4}{3}\varphi$ with $p = \frac{2}{n+1}$ and $E_3(X_y) = 2 - 2\phi + \frac{2}{3}\varphi$ with $p = \frac{3}{n+1}$, we have

$$E_3(X_x) = 2 - \frac{12}{(n+1)} + \frac{36}{(n+1)^2} - \frac{48}{(n+1)^3}\ , \quad \text{and}$$
$$E_3(X_y) = \frac{36}{(n+1)} + \frac{162}{(n+1)^2} - \frac{162}{(n+1)^3}\ .$$

From Lemma 20, $E_k(X_{t+1} \mid X_t, Y_t) = E_k(X_x)\frac{X_t}{2} + E_k(X_y)\frac{Y_t}{3}$ then:

$$E_3(X_{t+1} \mid X_t, Y_t) = \left(2 - \frac{12}{(n+1)} + \frac{72}{(n+1)^2} - \frac{96}{(n+1)^3}\right)\frac{X_t}{2} + \left(\frac{36}{(n+1)} + \frac{162}{(n+1)^2} - \frac{162}{(n+1)^3}\right)\frac{Y_t}{3}\ .$$

Using these expected values we recover the 3-OneStep transition matrix given in Lemma 15.

Lemma 35. *The coefficients of the power expansion of the MinMacroSplit algorithm satisfy:*

- *The initial values are $\alpha_1 = 3$ and $\beta_1 = 4$.*
- *For $1 < j < 8$ the values are computed directly (see Table 4).*
- *For $j > 1$ we have $\alpha_{j+6} = -2^6 3^3 \alpha_j$ and $\beta_{j+6} = -3^9 \beta_j$.*

Proof. For $j > 1$ we prove by induction on j the relation

$$\mathcal{B}_{j+6}\langle \mathcal{Y}_x(i)\rangle = -3^3 \mathcal{B}_j \langle \mathcal{Y}_x(i)\rangle.$$

j	$\mathcal{B}_j\langle\mathcal{Y}_x(i)\rangle$	α_j	$\mathcal{B}_j\langle\mathcal{X}_y(i)\rangle$	β_j
2	-6	12	-6	18
3	-12	48	-6	54
4	-18	144	0	0
5	-18	288	18	-1458
6	0	0	54	-13122
7	54	-3456	108	-78732
8	162	-20736	162	-354294
9	324	-82944	162	-1062882
10	486	-248832	0	0
11	486	-497664	-486	28697814
12	0	0	-1458	258280326
13	-1458	5971968	-2916	1549681956

Table 4. Coefficients of the power expansion of the MinMacroSplit algorithm. For $1 < j < 8$ we have the relations $\mathcal{B}_{j+6}\langle\mathcal{Y}_x(i)\rangle = -3^3\mathcal{B}_j\langle\mathcal{Y}_x(i)\rangle$, $\alpha_{j+6} = -2^6 3^3\alpha_j$, $\mathcal{B}_{j+6}\langle\mathcal{X}_y(i)\rangle = -3^3\mathcal{B}_j\langle\mathcal{X}_y(i)\rangle$ and $\beta_{j+6} = -3^9\beta_j$.

First, we verify that it holds for $j = 2, 3, \ldots, 7$. Substituting the values of $\mathcal{Y}_x(i)$ given in Lemma 32 and using the properties of the binomial transform given in Lemma 24 (item 3) we have:

$$\begin{aligned}\mathcal{B}_j\langle\mathcal{Y}_x(i)\rangle &= \sum_{i=0}^{j}(-1)^i\binom{j}{i}i + 2\sum_{i=0}^{j}(-1)^{3i+1}\binom{j}{3i+1} - 2\sum_{i=0}^{j}(-1)^{3i+2}\binom{j}{3i+2}\\ &= -2\left(\sum_{i=0}^{j}(-1)^i\binom{j}{3i+1} + \sum_{i=0}^{j}(-1)^i\binom{j}{3i+2}\right).\end{aligned}$$

Using Maple [HHR96] we compute the preceding expression and get the values of $\mathcal{B}_j\langle\mathcal{Y}_x(i)\rangle$ for $j = 2, \ldots, 13$ and we check the start of the induction (see Table 4). By induction, we assume for $1 < k < j$ the equality $\mathcal{B}_{k+6}\langle\mathcal{Y}_x(i)\rangle = -3^3\mathcal{B}_k\langle\mathcal{Y}_x(i)\rangle$ and we have to prove $\mathcal{B}_{j+6}\langle\mathcal{Y}_x(i)\rangle = -3^3\mathcal{B}_j\langle\mathcal{Y}_x(i)\rangle$. We can assume that $j > 7$, and by applying Lemma 25 (item 4) to the preceding relation we get

$$\sum_{\ell=0}^{6}(-1)^\ell\binom{6}{\ell}\mathcal{B}_j\langle\mathcal{Y}_x(i+\ell)\rangle = 3^3\sum_{\ell=0}^{6}(-1)^\ell\binom{6}{\ell}\mathcal{B}_{j-6}\langle\mathcal{Y}_x(i+\ell)\rangle.$$

Therefore, the induction holds if, for $\ell = 0, \ldots, 6$ we have:

$$\mathcal{B}_j\langle\mathcal{Y}_x(i+\ell)\rangle = -3^3\mathcal{B}_{j-6}\langle\mathcal{Y}_x(i+\ell)\rangle.$$

Let us consider the different values of ℓ. The case $\ell = 0$ is verified by induction. When $\ell = 1$, by Lemma 25 (item 3) and induction we have

$$\begin{aligned}\mathcal{B}_j\langle\mathcal{Y}_x(i+1)\rangle &= \mathcal{B}_j\langle\mathcal{Y}_x(i)\rangle - \mathcal{B}_{j+1}\langle\mathcal{Y}_x(i)\rangle\\ &= -3^3\mathcal{B}_{j-6}\langle\mathcal{Y}_x(i)\rangle + 3^3\mathcal{B}_{j-5}\langle\mathcal{Y}_x(i)\rangle\\ &= -3^3\mathcal{B}_{j-6}\langle\mathcal{Y}_x(i+1)\rangle.\end{aligned}$$

The case $\ell = 2$ can be proved in a similar way using the case $\ell = 1$.

$$\mathcal{B}_j\langle\mathcal{Y}_x(i+2)\rangle = \mathcal{B}_j\langle\mathcal{Y}_x(i+1)\rangle - \mathcal{B}_{j+1}\langle\mathcal{Y}_x(i+1)\rangle = -3^3\mathcal{B}_{j-6}\langle\mathcal{Y}_x(i+2)\rangle.$$

The case $\ell = 3$ is similar. The cases $\ell = 4, 5, 6$ can be reduced to the previous ones because

$$\mathcal{Y}_x(i+\ell) = \mathcal{Y}_x(i+\ell-3) + 3.$$

As for $j > 1$ we have $\mathcal{B}_{j+6}\langle \mathcal{Y}_x(i)\rangle = -3^3\mathcal{B}_j\langle \mathcal{Y}_x(i)\rangle$, then for $j > 1$ we have that

$$\alpha_{j+6} = -2^{j+6-1}\mathcal{B}_{j+6}\langle \mathcal{Y}_x(i)\rangle = -2^6 3^3(-2^{j-1}\mathcal{B}_j\langle \mathcal{Y}_x(i)\rangle) = -2^6 3^3\alpha_j.$$

The first α_j values are computed directly using the definition (see Table 4).
The β_j values can be found similarly. For $j > 1$ we have $\mathcal{B}_{j+6}\langle \mathcal{X}_y(i)\rangle = -3^3\mathcal{B}_j\langle \mathcal{X}_y(i)\rangle$. To start the induction we use the expression:

$$\mathcal{B}_j\langle \mathcal{X}_y(i)\rangle = -4\sum_{i=0}^{j}(-1)^i\binom{j}{3i+1} + 2\sum_{i=0}^{j}(-1)^i\binom{j}{3i+2}.$$

The first values of $\mathcal{B}_j\langle \mathcal{X}_y(i)\rangle$ and β_j have been computed directly (see Table 4). Moreover, for $j > 1$ we have that $\beta_{j+6} = -3^9\beta_j$. $\square$

Lemma 36. *Define the functions $g(j) = \lfloor (j-2)/6 \rfloor$ and $f(j) = j - 6g(j)$. For $j \geq 8$ the coefficients of the MinMacroSplit algorithm satisfy:*

$$\alpha_j = (-2^6 3^3)^{g(j)}\alpha_{f(j)}, \quad \beta_j = (-3^9)^{g(j)}\beta_{f(j)}.$$

The k-OneStep transition matrix is then:

$$\begin{aligned} T_{n,k} = {} & \left(1 + \frac{k}{n+1}\right) I + \frac{k}{n+1}\begin{pmatrix} -3 & 4 \\ 3 & -4 \end{pmatrix} \\ & + \sum_{1<j<8} \frac{(-1)^j}{(n+1)^j}\binom{k}{j}\begin{pmatrix} \alpha_j & -\beta_j \\ -\alpha_j & \beta_j \end{pmatrix} \\ & + \sum_{j \geq 8} \frac{(-1)^j}{(n+1)^j}\binom{k}{j}(-3)^{3g(j)}\begin{pmatrix} 2^{6g(j)}\alpha_{f(j)} & -3^{6g(j)}\beta_{f(j)} \\ -2^{6g(j)}\alpha_{f(j)} & 3^{6g(j)}\beta_{f(j)} \end{pmatrix} \end{aligned}$$

Proof. For $j \geq 8$, every α_j can be rewritten (see Lemma 35), for instance:

$$\alpha_{25} = -2^6 3^3\alpha_{19} = (-2^6 3^3)^2\alpha_{13} = (-2^6 3^3)^3\alpha_7.$$

Let us introduce $f(j)$ and $g(j)$ such that $\alpha_j = (-2^6 3^3)^{g(j)}\alpha_{f(j)}$ and $f(j) \in \{2, \ldots, 7\}$. For instance, when $j = 25$, we have $f(25) = 7$ and $g(25) = 3$. In general:

$$f(j) = \begin{cases} j \bmod 6 & \text{if } j \bmod 6 \in \{2, \ldots, 5\} \\ 6 + (j \bmod 6) & \text{if } j \bmod 6 \in \{0, 1\} \end{cases}$$
$$g(j) = \begin{cases} \lfloor j/6 \rfloor & \text{if } j \bmod 6 \in \{2, \ldots, 5\} \\ \lfloor j/6 \rfloor - 1 & \text{if } j \bmod 6 \in \{0, 1\}. \end{cases}$$

These functions can be rewritten as given in the statement of the lemma. $\square$

For $k = 1, 2$ we obtain the Lemmas 5 and 14 respectively. For $k = 3$ we obtain the second part of Lemma 15.

6.3 Analysis

Now we analyze the two algorithms presented. In fact, as they both have a similar structure, we do an analysis that applies to both algorithms which shows that they converge to the sequential case.

Definition 37. The expected probability to hit one x leaf in an insertion is $px_t = E_k(X_t)/(n+1)$. Similarly $py_t = E_k(Y_t)/(n+1)$ for a y leaf.

Because $E_k(X_t) + E_k(Y_t) = n+1$ we have that $py_t = 1 - px_t$. When $k = 1$ these expected probabilities have been studied in [EZG$^+$82] and px_t converges to $4/7$. We will show next that px_t also converges to $4/7$ from below when $k > 1$.

Theorem 38. *Let $px_t = 4/7 - \delta_n$. For constant k, when $n \to \infty$, δ_n converges to 0.*

Proof. From the recurrence equation of Theorem 27 (following [BY95]) we obtain a recurrence equation in only one variable:

$$px_{t+1} = px_t + \frac{k}{n+k+1}(4 - 7px_t) + \frac{1}{n+k+1}\sum_{j=2}^{k}\left(\frac{-1}{n+1}\right)^{j-1}\binom{k}{j}(\beta_j - (\alpha_j + \beta_j)px_t) \ ,$$

where we have used the fact that α_1 and β_1 are equal for both algorithms. Using the definition of px_t and replacing in the previous equation we obtain:

$$\delta_{n+k} = \delta_n - \frac{7k}{n+k+1}\delta_n - \frac{1}{n+k+1}\sum_{j=2}^{k}\left(\frac{-1}{n+1}\right)^{j-1}\binom{k}{j}((3\beta_j - 4\alpha_j)/7 + (\alpha_j + \beta_j)\delta_n) \ .$$

Let:

$$R_k(n) = -\sum_{j=2}^{k}\left(\frac{-1}{n+1}\right)^{j-1}\binom{k}{j}(3\beta_j - 4\alpha_j)/7 \ , \text{ and}$$

$$S_k(n) = -\sum_{j=2}^{k}\left(\frac{-1}{n+1}\right)^{j-1}\binom{k}{j}(\alpha_j + \beta_j).$$

Using these functions we have

$$\delta_{n+k} = \delta_n\left(1 - \frac{7k}{n+k+1} + \frac{S_k(n)}{n+k+1}\right) + \frac{R_k(n)}{n+k+1} \ .$$

Next we show that there exist n_0 such that for $n > n_0$, $\delta_{n+k} < \epsilon(n)\delta_n$ where $0 < \epsilon(n) < 1$ (by definition recall that $0 < \delta_n < 1$). Let $\epsilon(n)$ be $(n - 5k + 1)/(n + k + 1) < 1$. This implies $n_0 > 5k$. Then, we can rewrite the recurrence as

$$\delta_{n+k} = \epsilon(n)\delta_n + \frac{(S_k(n) - k)\delta_n + R_k(n)}{n+k+1} \ .$$

Now we show that the second term is negative. We can express $R_k(n)$ as $\sum_{j=1}^{k-1}\frac{r_j}{(n+1)^j} < \frac{r_2}{n+1} + O\left((n+1)^{-2}\right)$. Similarly, we have $S_k(n) = \sum_{j=1}^{k-1}\frac{s_j}{(n+1)^j} < \frac{s_2}{n+1} + O\left((n+1)^{-2}\right)$. Then we need to show that

$$\delta_n s_2 + r_2 + O\left(\frac{1}{n+1}\right) < k\delta_n(n+1) \ .$$

We can strengthen the inequality using $\delta_n = 1$, obtaining

$$n + 1 > \frac{s_2 + r_2}{k} + O\left(\frac{1}{k(n+1)}\right) \ .$$

Notice that the constant in the $1/(n+1)$ order term is less than $\max_{j=3}^{k-1}(|r_j + s_j|)$. The above inequality is satisfied for any $n > n_0$ by choosing the smallest n_0 that fulfills it. This is always possible because as the left side increases $(n_0 + 1)$, the right-hand side decreases. For example, for $k = 2$ we have that $\alpha_2 = 12$ and $\beta_2 = 18$, so we must have $n_0 > 92(k-1)/7 \approx 15k$. For $k = 3$ we obtain $n_0 > 306(k-1)/7 \approx 44k$. Hence, applying i times the inequality $\delta_{n+k} < \epsilon(n)\delta_n$ starting at any $n > n_0$ we have $\delta_{n+ik} < \prod_{j=0}^{i-1}\epsilon(n + jk)\delta_n$. Therefore, $\lim_{\to\infty}\delta_n = 0$. $\square$

7 Bounding the PVW algorithm

In this section we prove that the MaxMacroSplit and MinMacroSplit algorithms bound the expected behavior of the PVW algorithm.

The following lemma proves that the coefficients $b(i,k,p)$ of the binomial probabilities decreases quickly because $p << 1$:

Lemma 39. *If $p^{-1} \geq 1 + c \cdot k$ then $b(i,k,p) \geq c \cdot b(i+1,k,p)$ for $i = 0, \ldots, k-1$.*

Proof.

$$\frac{b(i,k,p)}{b(i+1,k,p)} = \frac{(i+1)(1-p)}{p(k-i)} \geq \frac{1-p}{pk} = \frac{1/p-1}{k}$$

By appling $p^{-1} \geq 1 + c \cdot k$ the inequality is satisfied. □

Taking expected values into the Lemma 10, the MaxMacroSplit and MinMacroSplit algorithms generate more and less x nodes than the PVW algorithm:

Lemma 40. *Let A be a random tree, assume that k random keys are inserted in parallel into this tree using each one of the MaxMacroSplit, PVW and MinMacroSplit algorithms. Let X be the number of x leaves after the insertion of these keys in A. This number depends of the algorithm used to perform the insertion. We note the expected number of x leaves after the insertion in each case as $E_k^{\textsf{MaxMacroSplit}}(X \mid A)$, $E_k^{\textsf{PVW}}(X \mid A)$ and $E_k^{\textsf{MinMacroSplit}}(X \mid A)$. It holds:*

$$E_k^{\textsf{MaxMacroSplit}}(X \mid A) \geq E_k^{\textsf{PVW}}(X \mid A) \geq E_k^{\textsf{MinMacroSplit}}(X \mid A).$$

Lemma 41. *Let A and B be two random trees with $n+1$ leaves with X_A and X_B leaves of x such that $E(X_A) \geq E(X_B)$, then insert k new random keys with the MaxMacroSplit or MinMacroSplit algorithm. Let X be the number of x leaves after the insertion of these keys. This number depends both, of the algorithm used and of the starting tree. We denote $E_k(X \mid A)$ as the expected number of x leaves after the insertion into A, and similarly for $E_k(X \mid B)$. It holds that $E_k(X \mid A) \geq E_k(X \mid B)$.*

Proof. Recall from Lemma 20 that

$$E_k(X \mid A) = \frac{1}{2}E(X_x)E(X_A) + \frac{1}{3}E(X_y)E(Y_A)\ ,$$
$$E_k(X \mid B) = \frac{1}{2}E(X_x)E(X_B) + \frac{1}{3}E(X_y)E(Y_B)\ .$$

Then $E_k(X \mid A) - E_k(X \mid B) \geq 0$ if $\frac{3}{2}E_k(X_x) \geq E_k(X_y)$. We verify this last inequality for both algorithms. First let us consider the MaxMacroSplit algorithm. Recall the functions F_0 and F_1 from Lemma 29. By Lemma 28 the inequality becomes

$$\frac{3}{2}(2F_0(k,p) - F_1(k,p)) \geq 3F_1(k,p).$$

Note that if $F_0(k,p) \geq 2F_1(k,p)$ the left term is greater than one and the right term is less than one. But by Lemma 39 for $p^{-1} \geq 1 + 2k$ it holds that $b(i,k,p) \geq 2b(i+1,k,p)$ and then $F_0(k,p) \geq 2F_1(k,p)$. As $p = \frac{2}{n+1}$ then at least $n \geq 4k+1$. Now we consider the MinMacroSplit algorithm. Recall the functions F_0, F_1 and F_2 from Lemma 33. By Lemma 32 the inequality becomes

$$\frac{3}{2}(F_0(k,p) + 2F_2(k,p)) \geq 2F_1(k,p) + F_2(k,p).$$

If $F_0(k,p) \geq 2F_1(k,p)$ the left term is greater than one and the right term is less than one. By applying Lemma 39 this inequality holds if at least $n \geq 6k+2$. □

As we have seen previously, Lemma 10 holds just for one step. Now we extend this lemma to t consecutive insertions of k keys.

Theorem 42. *Let $X_t^{MaxMacroSplit}$, $Y_t^{MaxMacroSplit}$ be the fringe in the MaxMacroSplit algorithm and $X_t^{MinMacroSplit}$, $Y_t^{MinMacroSplit}$ be the fringe in the MinMacroSplit algorithm. Let X_t^{PVW}, Y_t^{PVW} be the fringe in the PVW algorithm, then we have:*

$$E_k(X_t^{MinMacroSplit}) \leq E_k(X_t^{PVW}) \leq E_k(X_t^{MaxMacroSplit}) ,$$
$$E_k(Y_t^{MaxMacroSplit}) \leq E_k(Y_t^{PVW}) \leq E_k(Y_t^{MinMacroSplit}) .$$

Proof. We prove the inequalities by induction on t. Recall that $E_k^{\mathsf{MaxMacroSplit}}(X \mid A)$ means the expected value of X when k keys have been inserted into a random tree A with the MaxMacroSplit algorithm. For $t = 1$, let A be a random tree, then by Lemma 40

$$E_k(X_1^{\mathsf{PVW}}) = E_k^{\mathsf{PVW}}(X \mid A) \leq E_k^{\mathsf{MaxMacroSplit}}(X \mid A) = E_k(X_1^{\mathsf{MaxMacroSplit}}) .$$

For $t > 1$ it holds by induction that $E_k(X_{t-1}^{\mathsf{PVW}}) \leq E_k(X_{t-1}^{\mathsf{MaxMacroSplit}})$ and we should demonstrate that $E_k(X_t^{\mathsf{PVW}}) \leq E_k(X_t^{\mathsf{MaxMacroSplit}})$. Let B_{t-1} and C_{t-1} be the random trees generated after inserting k keys $t-1$ times with the PVW and MaxMacroSplit algorithm. Then by Lemma 40

$$E_k(X_t^{\mathsf{PVW}}) = E_k^{\mathsf{PVW}}(X \mid B_{t-1}) \leq E_k^{\mathsf{MaxMacroSplit}}(X \mid B_{t-1}).$$

By Lemma 41 and the hypothesis of the induction

$$E_k^{\mathsf{MaxMacroSplit}}(X \mid B_{t-1}) \leq E_k^{\mathsf{MaxMacroSplit}}(X \mid C_{t-1}) = E_k(X_t^{\mathsf{MaxMacroSplit}}).$$

□

A corollary to the last theorem using the analysis of the fringes of the MaxMacroSplit and MinMacroSplit algorithms at the end of the previous section, is that the fringe of the PVW algorithm converge to the fringe of the sequential case. Hence, asymptotically, all the bounds for normal 2-3 trees with a fringe of height one in [Yao78b,EZG+82] apply also to 2-3 trees updated in parallel by the PVW algorithm. For example, the average space for keys used in the fringe is asymptotically $E_k(X_t + 2Y_t) = 4(n+1)/7$ and as the average number of keys outside the fringe is $3(n+1)/7$, we have that the total average space used in the tree for large n is bounded by $(n+1)$ and $10(n+1)/7$. In the transient phase (small n), the parallel case is slightly better than the sequential case, and the convergence speed depends on k.

8 Conclusions

We have explained the evolution or global behavior of the fringe with a Markov chain whose matrix coefficients are determined by the local behavior of the MacroSplit rule and the binomial distribution of keys that can reach any node.

We have proved that the expected evolution of the fringe generated by the PVW algorithm is bounded by the expected evolutions of the MinMacroSplit and MaxMacroSplit algorithms (Theorem 42) and we have developed the power expansion of these last two algorithms (Lemmas 28 to 35). Then, for any number of keys it is possible to bound the expected number of leaves of two types generated by the PVW algorithm. Asymptotically these bounds converge to the sequential case, which means that the parallel algorithm does not worsen the behavior of the tree. This is true for constant k, and the case of variable $k = f(n)$ is left open, but the results should be similar if $f(n) = o(n)$.

The analysis presented is not only valid for the PVW insertion algorithm, as the MacroSplit algorithms also bound any insertion algorithm that do not use overflow or underflow techniques that involve sibling nodes. There are synchronized parallel algorithms for other search structures as B-trees [HS94], Skip lists [GMM96], AVL trees [GM98,MD98], and Red-black trees [MV98]. Our analysis is generic and can be extended to B-trees, AVL trees, and Red-black trees. The exact average case analysis of balanced search trees remains open for both the sequential and the parallel case.

Acknowledgments. We thank M.S. Busquier for early joint work on this topic. We thank the anonymous referees for the detailed comments that improved the presentation of the paper.

References

[AFP88] D. Aldous, B. Flannery, and J.L. Palacios. Two applications of urn processes: The fringe analysis of search trees and the simulation of quasi-stationary distributions of markov chains. *Probab. Eng. Inf. Sci.*, 2:293–307, 1988.

[AHJ83] A.V. Aho, J.E. Hopcroft, and Ullman J.D. *Data structures and algorithms.* Addison-Wesley, 1983.

[AHU74] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The design and analysis of computer algorithms.* Addison-Wesley, 1974.

[BP85] A. Bagchi and A. K. Pal. Asymptotic normality in the generalized Polya-Eggenberger urn model, with an application to computer data structures. *Siam J. Alg. Disc. Meth.*, 6(3):394–405, 1985.

[BRM99] G.H Blelloch and M. Reid-Miller. Pipelining with futures. *Theory of Comp. Sys.*, 32:213–239, 1999.

[BY95] R.A. Baeza-Yates. Fringe analysis revisited. *ACM Computing Surveys*, 27(1):109–119, 1995.

[BYGM98] R. Baeza-Yates, J. Gabarró, and X. Messeguer. Fringe analysis of synchronized parallel algorithms on 2–3 trees. In *Randomization and Approximation Techniques in Computer Science (RANDOM'98)*, Published by Springer-Verlag in LNCS 1518, pages 131–144, 1998.

[BYP95] R.A. Baeza-Yates and Poblete P.V. Higher-order analysis of 2–3 trees. *International Journal of Foundations of Computer Science*, 6(1):1–10, 1995.

[EZG⁺82] B. Eisenbarth, N. Ziviani, G.H. Gonnet, K. Mehlhorn, and D. Wood. The theory of fringe analysis and its application to 2–3 trees and B-trees. *Information and Control*, 55(1-3):125–174, 1982.

[Fel68] W. Feller. *An introduction to probability theory and its applications*, volume first. John Wiley, third edition, 1968.

[GKP94] R.L. Graham, D.E. Knuth, and O. Patashnik. *Concrete mathematics.* Addison-Wesley, second edition, 1994.

[GM98] J. Gabarró and X. Messeguer. Parallel dictionaries with local rules on AVL and Brother trees. *Information Processing Letters*, 68(2):79–85, 1998.

[GMM96] J. Gabarró, C. Martínez, and X. Messeguer. A design of a parallel dictionary using skip lists. *Theoretical Computer Science*, 158:1–33, 1996.

[HHR96] K.H. Heal, M.L. Hansen, and K.L. Rickard. *Maple V learning guide.* Springer, 1996.

[HS94] L. Higham and E. Schenks. Maintaining B-trees on an EREW PRAM. *J. of Parallel and Dist. Comp.*, 22:329–335, 1994.

[JáJ92] J. JáJá. *An Introduction to Parallel Algorithms.* Addison-Wesley, 1992.

[MD98] M. Medidi and N. Deo. Parallel dictionaries using AVL trees. *J. of Parallel and Distributed Computing*, 49(1):146–155, 1998.

[MV98] X. Messeguer and B. Valles. Synchronized parallel algorithms on red-black trees. In Universidade do Porto, editor, *3th. International Meeting on Vector and Parallel Processing (VECPAR98)*, pages 699–704, 1998.

[PMP95] P.V. Poblete, J.I. Munro, and T. Papadakis. The binomial transform and its application to the analysis of skip lists. In *ESA 95*, pages 1–10. Springer-Verlag, 1995.

[PVW83] W. Paul, U. Vishkin, and H. Wagener. Parallel dictionaries on 2–3 trees. In J. Díaz, editor, *Proc. 10th ICALP, LNCS 154*, pages 597–609. Springer-Verlag, 1983.

[Yao78a] A. Yao. On random 2–3 trees. *Acta Informatica*, 9(159):159–170, 1978.

[Yao78b] A.C-C. Yao. On random 2-3 trees. *Acta Informatica*, 9(2):159–170, 1978.