Fringe analysis of synchronized parallel insertion algorithms on 2–3 trees*

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

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

Abstract. The fringe analysis studies the distribution of bottom subtrees or fringe of trees under the assumption of random selection of keys, yielding an average case analysis of the fringe of trees.
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 MacroSplit algorithms, and our main theorem proves that two algorithms of this family, denoted MaxMacroSplit and MinMacroSplit, upper and lower bounds the fringe of the PVW algorithm.
Published papers deal with the fringe analysis of sequential algorithms and it was an open problem for parallel algorithms on search trees. We extend the fringe analysis to parallel algorithms and we get a rich mathematical structure giving new interpretations even in the sequential case. We prove that the random selection of keys generates a binomial distribution of them between leaves, that the synchronized insertions of keys can be modeled by a Markov chain, and that the coefficients of the transition matrix of the Markov Chain are related with 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.

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 are balanced search trees as 2–3 trees introduced by J. Hopcroft in the

* Partially supported by ACI-CONICYT through the Catalunya-Chile Cooperation Program (DOG 2320-30.1.1997), RITOS network (CYTED), ESPRIT LTR Project no. 20244–ALCOM-IT, DGICYT under grant PB95-0787 (project KOALA), CICYT Project TIC97-1475-CE, CIRIT 1997SGR-00366, and PR98-11 of the Universitat Politècnica de Catalunya.

seventies [AHJ74]. The exact analysis of the sequential case is still open, but good lower and upper bounds for several complexity measures have been obtained using a technique called *fringe analysis*. This analysis studies the distribution of bottom subtrees or fringe of trees under the assumption of random selection of 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.

We are interested on the fringe analysis of the synchronized parallel algorithm on 2–3 trees designed by Paul, Vishkin, and Wagener (PVW)[PVW83]. This kind of algorithms manage 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 these concurrent accesses. The first synchronized parallel algorithms on search trees was the PVW one. 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 $O(\log n)$. The analysis of this algorithm is still open and the main drawback is the reconstructing phase that is composed by waves of synchronized processors which modifies the tree bottom-up.

In this paper we introduce a new family of synchronized parallel algorithm, denoted MacroSplit, whose two extreme cases, denoted MaxMacroSplit and MinMacroSplit algorithms, bound the PVW one 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 meanwhile the PVW algorithm needs a pipeline of waves.

The fringe analysis of the MacroSplit algorithms is an extension of the fringe analysis of sequential case but with many significant improvements . As later on is shown, 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 random selection of keys generates a *binomial* distribution of them 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 the fringe analysis, but from practical considerations it is necessary to develop a formula to compute them. For this reason we present the power expansion of the transition matrix and we

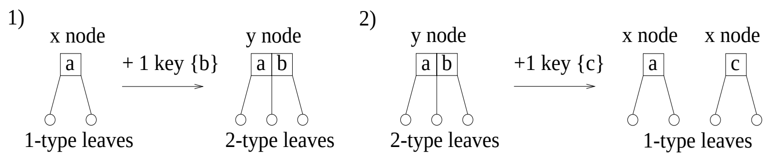

Fig. 1. The transformation of x and y bottom nodes after insertion of one key. In case (1) the key b hits a bottom node x and node x transforms into a node y. In case (2) the key c hits a bottom node y and node y splits into 2 nodes x.

calculate its coefficients for the two algorithms MaxMacroSplit and MinMacroSplit.

The rest of the paper is organized following the main facts pointed 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 greater values. 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. Section 7 shows that the fringe generated by these two algorithms bounds the fringe generated by the PVW. 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

The fringe of a tree is composed by the subtrees on the bottom part of the tree. Our fringe is composed by trees of height one. A bottom node with one key is called and x node, and a bottom node with two keys is called an y node. These nodes separate the leaves into 1-type leaves if their parents are x nodes and 2-type leaves if their parents are y nodes.

Let X_t and Y_t be the random variables associated to the number of 1-type leaves and 2-type leaves respectively at the step t. Notice that $X_t + Y_t = n + 1$ being n the number of keys of the tree (we assume also that it is not possible to insert a key greater than the key located at the right most leaf of the tree).

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

The probability that a key hits a bottom node x is $\frac{X_t}{n+1}$ and for a node y is $\frac{Y_t}{n+1}$. The conditional expectations verify

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

The expected number of leaves (conditioned to the random insertion of one key) at the step t can be modeled by the following definition

Definition 1. [Yao78,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 verifying:

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

As the conditional expectations verify

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

we get:

Theorem 2. *[EZG$^+$82,BY95] The 1-OneStep transition matrix is:*

$$T_{n,1} = \left(1 + \frac{1}{n+1}\right) I + \frac{1}{n+1} \begin{pmatrix} -3 & 4 \\ 3 & -4 \end{pmatrix} \qquad \textit{being} \qquad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

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

$$T_{n,1} = \frac{1}{n+1} \begin{pmatrix} n-1 & 4 \\ 3 & n-2 \end{pmatrix}$$

Later on we will give a direct proof of this compact expression.

3 Synchronized parallel insertion algorithms

In this section we recall the algorithm of Paul, Vishkin, and Wagener [PVW83], and we introduce our MacroSplit algorithm. It is assumed that an array of k sorted keys $a[1 \ldots k]$ is inserted into a 2-3 tree having n leaves. The algorithms first hang the keys from the leaves and later rebalance the tree. The PVW algorithms differs from the MacroSplit algorithms on the rebalancing phase.

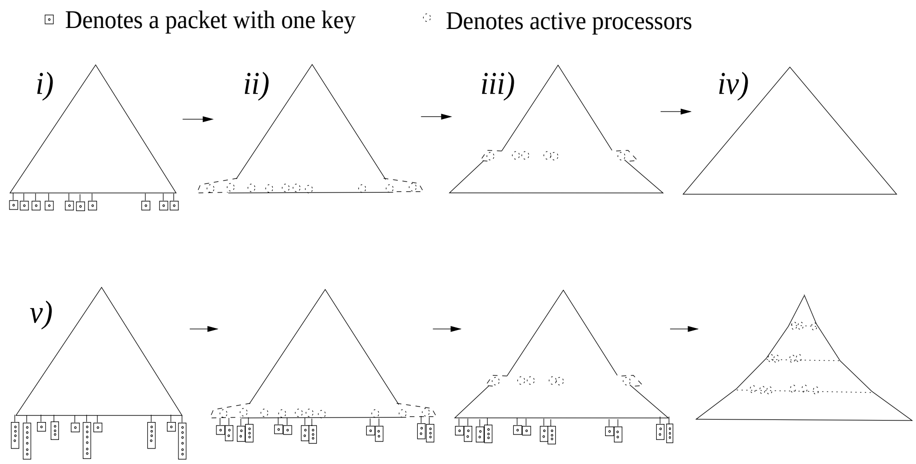

Fig. 2. Traveling waves for the PVW insertion algorithm on 2–3 trees

3.1 PVW algorithm

The tree is balanced using pipelines of processors. These pipelines can be envisaged intuitively in terms of traveling plane waves. Assume, for instance, the basic insertion case in which every leaf incorporates at most one new key (Figure 2.i). Something like a *wave* of processors is generated at the bottom of the tree, namely a *plane wave*, because all leaves of a 2–3 tree have the same depth (Figure 2.ii). This wave is sent up in further iterations (Figure 2.iii) until it disappears (Figure 2.iv). In the general insertion case (Figure 2.v), in which a packet of many new keys can hang from a single leaf, a pipeline of waves is generated to get something like periodic traveling waves. Each new wave is created as follows: some iterations after the last wave has been created, the packets are split, the middle key of each one is attached as a new leaf and the remaining left subpacket is hung from the new leaf, while the right subpacket is maintained in the same leaf. 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 $O(\log n + \log k)$.

3.2 MacroSplit algorithm

In the general insertion case (Figure 2.v) the MacroSplit algorithm 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 usage of rules so called MacroSplit rules (see Figure 3). These rules determine the transformation of wide nodes into

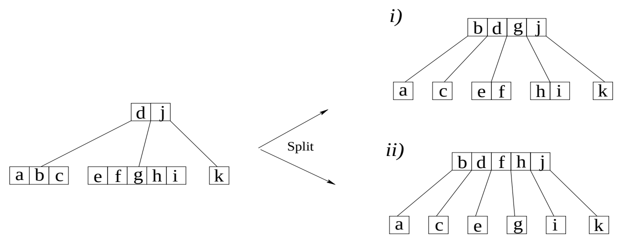

Fig. 3. Choices for MacroSplit rules. In (i) the rule creates a maximum number of splits. In (ii) the rule creates the minimum number. Intermediate strategies are allowed.

nodes x and y. For instance, the rule of case (i) of Figure 3 makes the maximum number of splits, and the rule of case (ii) makes the minimum number of splits. Intermediate strategies are allowed. Let us see several examples. At most, k keys can reach a node. If the node stores more than two keys, it must split using a MacroSplit rule. Table 1 show us several split possibilities for x and y bottom nodes. For instance, the first row show 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 show us how x and y nodes can be split when $k = 4$. In this case a bottom node x can be split into 3 nodes x or into 2 nodes y. Later on we will consider two extreme cases:

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

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

When $k = 1$ or 2 both algorithms coincides (see table 1).

The usage of MacroSplit rules increases the parallel time, but it allows 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 meanwhile the MacroSplit algorithm creates only one wave, but in the first case the time spent at each step is constant meanwhile the time spent in the second case is linear on k.

Let us introduce the analysis of the MacroSplit algorithm. Consider that at the $t + 1$ step k random keys (we asume a uniform distribution of them) fall in parallel into a fringe with X_t leaves of 1-type and Y_t leaves 2-type such that $X_t + Y_t = n + 1$. The expected values of X_{t+1} and Y_{t+1} after the insertions depends on two facts.

- The concrete form of the MacroSplit algorithm. This algorithm explicits how many leaves of 1-type and 2-type will be generated by bottom nodes when they receive some number of keys.
- The preceding values of X_t and Y_t.

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 once k keys are inserted.

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

Definition 3. Given a fringe with $n+1$ leaves and a MacroSplit algorithm, we define the k-OneStep transition matrix $T_{n,k}$ as the matrix verifying:

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

3.3 A first connection between both approaches

The MacroSplit algorithm can be seen as a "high level" description of the PVW algorithm. 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 meanwhile this same analysis for the MacroSplit algorithms take 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 4. *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 an algorithm A. The values of the fringe at the end of the first step depends on A, therefore we note X_1^{A}, Y_1^{A} these values. If algorithm A is MaxMacroSplit, PVW or MinMacroSplit algorithm, it holds*

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

The insertion of the first k keys generates three different trees depending on the algorithm. When the second set of k keys is inserted, it is possible that the PVW algorithm creates more x nodes than the MaxMacroSplit algorithm because the initial tree is different (even though the initial tree for the PVW algorithm has less x nodes than the initial tree for the MaxMacroSplit algorithm). Namely,

$(\cdot,\cdot)$	$P(\cdot,\cdot)$	$E(X_{t+1} \mid X_t, Y_t, 2, (.,.))$	$E(Y_{t+1} \mid X_t, Y_t, 2, (.,.))$
(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.

it is possible to find a fringe X_0, Y_0 and 2 batches of (non random) k keys such that the following inequalities,

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

which bound the values of the fringe and the end of the second step, are false: Then, the previous lemma holds only for one step. Later on, we prove that this lemma can be extended to consecutive insertions of k keys if we take into account the *expected* number of nodes.

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.

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 2-OneStep $T_{n,2}$ transition matrix:

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

We compute the probabilities of the different splits by an exhaustive case analysis (see Table 2). 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 $P(\cdot,\cdot)$, being $(.,.)$ the generic pair of nodes accessed.

As $E(X_{t+1} \mid 2) = E(E(X_{t+1} \mid X_t, Y_t, 2))$ we compute the expected number of 1-type leaves as

$$E(X_{t+1}|X_t,Y_t,2) = \sum_{(\cdot,\cdot)} P(\cdot,\cdot)\ E(X_{t+1}|X_t,Y_t,2,(.,.))$$

being $E(X_{t+1}|X_t,Y_t,2,(.,.))$ the expected number of 1-type leaves when 2 keys reach node $(\cdot,\cdot)$ conditioned to X_t and Y_t. For instance, if both keys reach different x nodes then it holds

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

and $E(X_{t+1}|X_t,Y_t,2,(x_1,x_2)) = X_t - 4$ (table 2 contains the other values).

Lemma 5. *The conditional expectations verify*

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

Proof. We compute the conditional expectation only for X_{t+1} (the Y_{t+1} term has a similar development). Then $E(X_{t+1}|X_t,Y_t,2)$ is:

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

As the conditional expectations are linear in X_t and Y_t and

$$\begin{aligned} E(X_{t+1} \mid 2) = E(E(X_{t+1} \mid X_t,Y_t,2)) \\ E(Y_{t+1} \mid 2) = E(E(Y_{t+1} \mid X_t,Y_t,2)) \end{aligned}$$

we have:

$(\cdot,\cdot)$	$P(\cdot,\cdot,\cdot)$	$E_{\ldots}(X_{n,3}\|X_n,Y_n)$	$E_{\ldots}(Y_{n,3}\|X_n,Y_n)$
(x,x,x)	$\frac{X_n}{n+1}\left(\frac{2}{n+1}\right)^2$	X_n	Y_n+3
(x_1,x_1,x_2)	$3\frac{X_n}{n+1}\frac{2}{n+1}\frac{X_n-2}{n+1}$	X_n	Y_n+3
(x_1,x_2,x_3)	$\frac{X_n}{n+1}\frac{X_n-2}{n+1}\frac{X_n-4}{n+1}$	X_n-6	Y_n+9
(x,x,y)	$3\frac{X_n}{n+1}\frac{2}{n+1}\frac{Y_n}{n+1}$	X_n+6	Y_n-3
(x,y,y)	$3\frac{X_n}{n+1}\frac{Y_n}{n+1}\frac{3}{n+1}$	X_n	Y_n+3
(x_1,x_2,y)	$3\frac{X_n}{n+1}\frac{X_n-2}{n+1}\frac{Y_n}{n+1}$	X_n	Y_n+3
(x,y_1,y_2)	$3\frac{X_n}{n+1}\frac{Y_n}{n+1}\frac{Y_n-3}{n+1}$	X_n+6	Y_n-3
(y_1,y_2,y_3)	$\frac{Y_n}{n+1}\frac{Y_n-3}{n+1}\frac{Y_n-6}{n+1}$	X_n+12	Y_n-9
(y_1,y_1,y_2)	$3\frac{Y_n}{n+1}\frac{3}{n+1}\frac{Y_n-3}{n+1}$	X_n+6	Y_n-3
(y,y,y)	$\frac{Y_n}{n+1}\left(\frac{3}{n+1}\right)^2$	X_n	Y_n+3

Table 3. Parallel insertion of three keys

Lemma 6. *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 $k=3$. Table 3 contains the exhaustive case analysis of the probabilities. Now there are two possibilities (third row of table 1). We have selected the second transformation that corresponds to the MinMacroSplit algorithm.

Lemma 7. *In the case of the MinMacroSplit algorithm, the 3-OneStep transition matrix $T_{n,3}$ 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 we can point several facts and questions:

1. The exhaustive case analysis (generalizing the sequential approach [EZG+82]) for larger $k=5,6,\ldots$, becomes intractable.

2. For $k = 1, 2, 3$ the expectations $E(X_{t+1} \mid X_t, Y_t, k)$ and $E(X_{t+1} \mid X_t, Y_t, k)$ are linear in X_t and Y_t. It is unclear why non-linear terms always disappears. Note that we assume this point of view in the equation k-OneStep transition matrix $T_{n,k}$.
3. The intuitive meaning of the coefficients appearing in the expectations is unclear. For instance, the term $1 - \frac{4}{n+1} + \frac{12}{(n+1)^2}$ appearing in $E(X_{t+1} \mid X_t, Y_t, 2)$ in lemma 5 does not have any direct explanation in terms of the MacroSplit algorithm.
4. By *local behavior* of the algorithm we mean what happens when i keys hit just *one* bottom node x or y (table 1) . By *global behavior* we mean the evolution of X_t and Y_t. The previous exhaustive analysis does not give a clear cut between the *local* and the *global behavior* of the MacroSplit algorithm.
5. Note that
   - Lemmas 6 and 7 can be envisaged as a *power expansion* over $(n+1)^{-1}$ of the transition matrix.
   - The matrices appearing when $k = 2$ also appears for $k = 3$ (see lemmas 6 and 7).

   This suggest us a power expansion of the k-OneStep of the form

$$T_{n,k} = \left(1 + \frac{k}{n+1}\right) I + \frac{\gamma_1(k)}{n+1} \begin{pmatrix} -3 & 4 \\ 3 & -4 \end{pmatrix} + \frac{\gamma_2(k)}{(n+1)^2} \begin{pmatrix} 12 & -18 \\ -12 & 18 \end{pmatrix} + \frac{\gamma_3(k)}{(n+1)^3} \begin{pmatrix} -48 & 54 \\ 48 & -54 \end{pmatrix} + \cdots$$

   Moreover, a little bit of thought suggest us $\gamma_i(k) = \binom{k}{i} \cdots$
6. The different coefficients appearing into the matrices reflect the behavior of the MacroSplit algorithm. We search for a precise meaning of this intuitive fact.

In the following we solve all these questions.

5 Behavior of the MacroSplit algorithms

In order to study the expected behavior of an x or y node belonging to a fringe of $n+1$ 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 1-type and 2-type leaves are generated when i keys fall at the same time into a unique node x or y. To deal with this fact we introduce the following definition.

Definition 8. At the bottom level, the local behavior of the MacroSplit algorithm is given by the following functions:

- The $\mathcal{X}_x(i)$ is the number of 1-type 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 1-type leaves after the insertion of i keys into an y node (for instance, $\mathcal{X}_y(0) = 0$, $\mathcal{X}_y(1) = 4, \ldots$).
- Dually, $\mathcal{Y}_x(i)$ is the number of 2-type leaves after the insertion of i keys into an x node (for instance, $\mathcal{Y}_x(0) = 0$, $\mathcal{Y}_x(1) = 3, \ldots$).
  Finally, $\mathcal{Y}_y(i)$ is the number of 2-type leaves after the insertion of i keys into an y node (for instance, $\mathcal{Y}_y(0) = 3$, $\mathcal{Y}_y(1) = 0, \ldots$).

These coefficients verify $\mathcal{X}_x(i) + \mathcal{Y}_x(i) = 2 + i$ and $\mathcal{X}_y(i) + \mathcal{Y}_y(i) = 3 + i$.

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

Lemma 9. *Let N_x and N_y be the random variables denoting the number of keys falling into a fixed bottom node x and y. Then, these variables follows a binomial distribution given by*

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

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

Recall that the expected value of the binomial distribution is kp.

The number of 1-type leaves generated by the keys falling into a unique node x is given by the random variable $X_x = \mathcal{X}_x(N_x)$ and the number of 2-type leaves generated by the keys falling into a unique node x is $Y_x = \mathcal{Y}_x(N_x)$ (similarly for X_y and Y_y).

Lemma 10. *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:*

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

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

Proof.

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

$\square$

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

Lemma 11. *The expected number of leaves generated by just one bottom node when k random keys are inserted in parallel into a fringe having $n+1$ is:*

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

5.2 Global behavior

Lemma 12. *Given an n-key random tree T with a fringe with X_t leaves of 1-*`type` *and Y_t leaves of 2-*`type`*, when k keys are inserted at random into T in one step we have*

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

Proof. Let us consider a fringe having X_t leaves of 1-`type` and Y_t leaves of 2-`type` and $X_t + Y_t = n+1$. Let us consider the set $\mathcal{S}$ of functions σ defined from $\{1, \ldots, k\}$ to $\{1, \ldots, n+1\}$. Note that each function σ determines the distribution of the k keys between the $n+1$ leaves. Then

$$E(X_{t+1}|X_t, Y_t, k) = \sum_{\sigma \in \mathcal{S}} P\{\sigma\} E(X_{t+1}|X_t, Y_t, k, \sigma).$$

Let $x_1, \ldots, x_{X_t/2}$ and $y_1, \ldots, y_{Y_t/3}$ be the x and y nodes, and let $X(x_m, \sigma)$ be the expected number of 1-`type` leaves created when k keys are inserted and some of them, determined by function σ, reaches node x_m. Then

$$E(X_{t+1}|X_t, Y_t, k, \sigma) = \sum_{m=1}^{X_t/2} X(x_m, \sigma) + \sum_{m=1}^{Y_t/3} X(y_m, \sigma),$$

and

$$\begin{aligned} E(X_{t+1}|X_t, Y_t, k) &= \sum_{\sigma \in \mathcal{S}} P\{\sigma\} \sum_{m=1}^{X_t/2} X(x_m, \sigma) + \sum_{\sigma \in \mathcal{S}} P\{\sigma\} \sum_{m=1}^{Y_t/3} X(y_m, \sigma) \\ &= \sum_{m=1}^{X_t/2} \sum_{\sigma \in \mathcal{S}} P\{\sigma\} X(x_m, \sigma) + \sum_{m=1}^{Y_t/3} \sum_{\sigma \in \mathcal{S}} P\{\sigma\} X(y_m, \sigma) \\ &= \frac{X_t}{2} \sum_{\sigma \in \mathcal{S}} P\{\sigma\} X(x, \sigma) + \frac{Y_t}{3} \sum_{\sigma \in \mathcal{S}} P\{\sigma\} X(y, \sigma) \end{aligned}$$

for any node x and y because nodes are not distinguishables. The set of functions σ assign $\binom{k}{i}$ times i keys with $0 \le i \le k$ to nodes x or y. Let us consider the case of a bottom node x. For each assignment there are 2^i possibilities to distribute i keys between the two leaves of this node. The other $k-i$ keys have to be assigned to the remaining $n-1$ leaves, so:

$$\sum_{\sigma\in\mathcal{S}} P\{\sigma\}X(x,\sigma) = \sum_{i=0}^{k} \frac{1}{(n+1)^k}\binom{k}{i}\ 2^i\ (n-1)^{k-i}\mathcal{X}_x(i)$$

$$= \sum_{i=0}^{k}\binom{k}{i}\left(1-\frac{2}{n+1}\right)^{k-i}\left(\frac{2}{n+1}\right)^{i}\mathcal{X}_x(i) = \sum_{i=0}^{k} b(i,k,\frac{2}{n+1})\mathcal{X}_x(i).$$

which is equal to $E(X_x|k)$ by lemma 10. In the case of a node y, there are 3^i possibilities to distribute i keys between the three leaves of such a node. The other $k-i$ have to be assigned to the other $n-2$ leaves and:

$$\sum_{\sigma\in\mathcal{S}} P\{\sigma\}X(y,\sigma) = \sum_{i=0}^{k} \frac{1}{(n+1)^k}\binom{k}{i}\ 3^i\ (n-2)^{k-i}\mathcal{X}_x(i)\sum_{i=0}^{k} b(i,k,\frac{3}{n+1})\mathcal{X}_y(i)$$

which is equal to $E(X_y|k)$ $\square$

Theorem 13. *Given a fringe with $n+1$ leaves and a* MacroSplit *algorithm, the* k-OneStep *transition matrix is:*

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

Proof. From the preceding lemma we have

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

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

$$E(X_t+1 \mid k) = \frac{1}{2}E(E(X_x \mid k)X_t \mid k) + \frac{1}{3}E(E(X_y \mid k)Y_t \mid k)$$

As X_x and X_t are independent $E(E(X_x \mid k)X_t \mid k) = E(X_x \mid k)E(X_t \mid k)$ and the proof is done. $\square$

Example 14. Let us recompute the 1-**OneStep** using theorem 13. Let us start with a bottom node x. As we have seen in the definition 8 we have $\mathcal{X}_x(0) = 2$, $\mathcal{X}_x(1) = 0$.

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

Using the property $E(X_x + Y_x \mid 1) = 2(1+\frac{1}{n+1})$ given in the lemma 11 we get

$$E(Y_x|1) = \frac{2}{n+1}\ 3$$

Let us consider a bottom node y.

$$E(X_y|1) = \left(1 - \frac{3}{n+1}\right)\mathcal{X}_y(0) + \left(\frac{3}{n+1}\right)\mathcal{X}_y(0) = \frac{3}{n+1}\,4$$

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

$$E(Y_y|1) = \frac{3}{n+1}(n-2)$$

Substituting we get

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

This concludes the example.

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, but previous published papers define the transition matrix by series (as we do in lemmas 6 and 7). In this section we show that these series are the power expansion over $(n+1)^{-1}$ of the k-OneStep transition matrix of theorem 13 as was suggested in note 5 of section 4.2.

Lemma 15. *Let I be the two dimensional identity matrix, the k-OneStep verifies:*

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

Proof. From lemma 11 we have: Substituting these values into the matrix expression $T_{n,k}$ given in the theorem 13 we get the result. □

In order to follow the power expansion, let us recall the *binomial transform* $\mathcal{B}$ recently developed by Poblete, Munro, and Papadakis [PMP95]. 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

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

This transformation verifies the following lemmas [PMP95]:

Lemma 16. *1. When $F_i = a$ we have $\hat{F}_0 = a$ and $\hat{F}_j = -1$ 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 17. *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 it holds:*

1. $F_i = \mathcal{B}_i \hat{F}_j$
2. $\mathcal{B}_j(a\,F_i + b\,G_i) = a\,\mathcal{B}_j F_i + b\,\mathcal{B}_j G_i$.
3. *For* $j > 0$ *we have* $\hat{F}_j = \mathcal{B}_{j-1}F_{i+1} - \hat{F}_{j-1}$.
4. $\hat{F}_j = \sum_{l=0}^{6}(-1)^l\binom{6}{l}\mathcal{B}_{j-6}F_{i+l}$ *for* $j \geq 6$
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* $\mathcal{B}_j b(i,\ell,p)F_i = p^j\hat{F}_j$.

In the following we will use a weighted form of the binomial transforms of $\langle \mathcal{Y}_x(i)\rangle_{i\geq 0}$ and $\langle \mathcal{X}_y(i)\rangle_{i\geq 0}$:

Definition 18. Let α_j and β_j be the coefficients $\alpha_j = -2^{j-1}\hat{\mathcal{Y}}_x(j)$ 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:

Lemma 19.

$$E(Y_x \mid k) = -2\sum_{j=0}^{k}\frac{(-1)^j}{(n+1)^j}\binom{k}{j}\alpha_j \qquad E(X_y \mid k) = -3\sum_{j=0}^{k}\frac{(-1)^j}{(n+1)^j}\binom{k}{j}\beta_j$$

Proof. Recall that

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

Consider the sequence $\langle E(Y_x \mid k)\rangle_{k\geq 0}$, by property 1.5:

$$\hat{E}(Y_x \mid j) = \mathcal{B}_j E(Y_x \mid k) = \Big(\frac{2}{n+1}\Big)^j\hat{\mathcal{Y}}_x(j)$$

Now we apply the property 1.1 of the binomial transform ($F_k = \mathcal{B}_k\hat{F}_j$),

$$E(Y_x \mid k) = \mathcal{B}_k\hat{E}(Y_x \mid j) = \mathcal{B}_k\Big(\Big(\frac{2}{n+1}\Big)^j\hat{\mathcal{Y}}_x(j)\Big)$$

Using linearity (property 1.2) and $\alpha_j = -2^{j-1}\hat{\mathcal{Y}}_x(j)$ we have

$$E(Y_x \mid k) = 2\mathcal{B}_k\Big(\Big(\frac{1}{n+1}\Big)^j 2^{j-1}\hat{\mathcal{Y}}_x(i)\Big) = -2\sum_{j=0}^{k}(-1)^j\binom{k}{j}\frac{\alpha_j}{(n+1)^j}$$

The case $E(x_y \mid k)$ is quite similar. □

From lemmas 15 and 19 we get the following expansion

Theorem 20. *The k-OneStep transition matrix can be rewritten as*

$$T_{n,k} = \Big(1+\frac{k}{n+1}\Big)I + \sum_{j=0}^{k}\frac{(-1)^j}{(n+1)^j}\binom{k}{j}\begin{pmatrix}\alpha_j & -\beta_j\\ -\alpha_j & \beta_j\end{pmatrix},$$

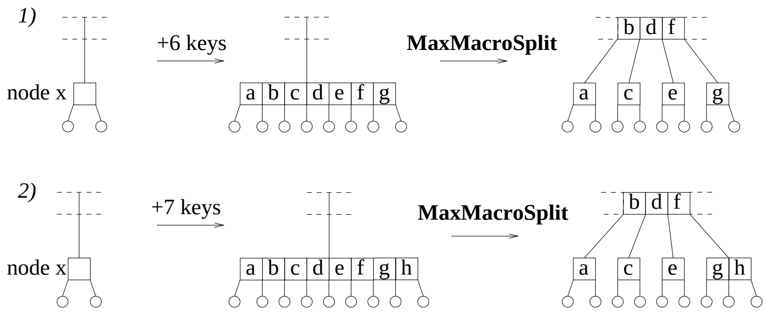

Fig. 4. Application of MaxMacroSplit algorithm on a node x

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 algorithms 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 a node x ($i = 6$ in the case 1 of the figure 4). This wide node splits by yielding $i + 2$ 1-type leaves (8 in the preceding case) and 0 2-type leaves. Then $\mathcal{X}_x(i) = i + 2$ and $\mathcal{Y}_x(i) = 0$. On the other hand, an odd number i of keys are attached ($i = 7$ in case 2 of the figure 4). In this case the split only creates one node y, 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 21. *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 of the MaxMacroSplit algorithm.

Lemma 22. *The* expected local behavior *is*

$$E(X_x \mid k) = kp + \frac{1}{2} + \frac{3}{2}(q-p)^k \quad E(Y_x \mid k) = \frac{3}{2} - \frac{3}{2}(q-p)^k \quad \textit{for} \quad p = \frac{2}{n+1}$$
$$E(X_y \mid k) = kp + \frac{3}{2} - \frac{3}{2}(q-p)^k \quad E(Y_y \mid k) = \frac{3}{2} + \frac{3}{2}(q-p)^k \quad \textit{for} \quad p = \frac{3}{n+1}$$

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

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

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

$$F_0(k,p) = \sum_{i=0,2,4,\ldots} b(i,k,p) \quad \text{and} \quad F_1(k,p) = \sum_{i=1,3,5,\ldots} b(i,k,p)$$

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

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

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 $F_1(k,p)$ acts as probabilities and we deal with a Markov chain having a 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 diagonalizes 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, 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 $E(X_x \mid k)$. As $p = \frac{2}{n+1}$, the expected value of Y_x can be computed directly from lemma 11:

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

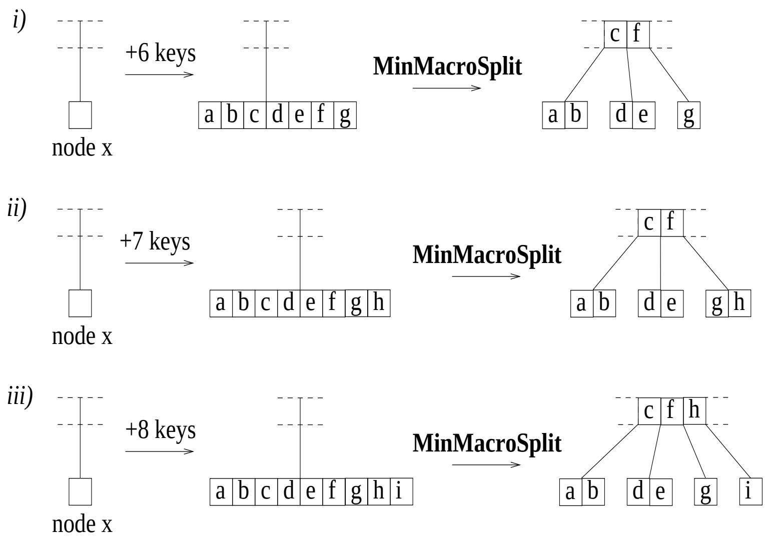

Fig. 5. Application of MinMacroSplit algorithm

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

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

we compute the expected value of Y_y. □

Lemma 23. *The coefficients of the* power expansion *verifies* $\alpha_0 = \beta_0 = 0$, $\beta_1 = 4$. *For* $j > 0$ *we have* $\alpha_j = -3\,4^{j-1}$ *and for* $j > 1$ *we have* $\beta_j = -3\,6^{j-1}$.

Proof. First we prove the value of α_j by using lemmas 3 and 2

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

By lemma 21, $\mathcal{Y}_x(i+1) - \mathcal{Y}_x(i) = 3$ then $\hat{\mathcal{Y}}_x(j) = 32^{j-1}$. As $\alpha_j = -3^{j-1}\hat{\mathcal{Y}}_x(j)$ then $\alpha_j = -3\,4^{j-1}$. The value of β_j can be proved in a similar manner by applying $\mathcal{X}_y(i) = i + \mathcal{Y}_x(i)$. □

6.2 The MinMacroSplit algorithm

The Figure 5 pictures the split of an x node when six, seven and eight keys are attached. The MinMacroSplit algorithm has the following characterization:

Lemma 24. *The* local behavior *of the MinMacroSplit algorithm is given by:*

– *For* $i \mod 3 = 0$ *we have* $\mathcal{X}_x(i) = 2$, $\mathcal{Y}_x(i) = i$, $\mathcal{X}_y(i) = 0$, $\mathcal{Y}_y(i) = i+3$.
– *For* $i \mod 3 = 1$ *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* $i \mod 3 = 2$ *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 25. *The expected local behavior is determined by:*

$$\begin{aligned} E(X_x \mid k) &= 2 - \frac{4}{3}\varphi & E(Y_x \mid k) &= pk + \frac{4}{3}\varphi & \textit{for} \quad p &= \frac{2}{n+1} \\ E(X_y \mid k) &= 2 - 2\phi + \frac{2}{3}\varphi & E(Y_y \mid k) &= pk + 1 + 2\phi - \frac{2}{3}\varphi & \textit{for} \quad p &= \frac{3}{n+1}. \end{aligned}$$

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

$$F_0(k,p) = \sum_{i=0,3,6,\dots} b(i,k,p) \qquad F_1(k,p) = \sum_{i=1,4,7\dots} b(i,k,p)$$
$$F_2(k,p) = \sum_{i=2,5,8,\dots} b(i,k,p)$$

The expected values can be rewritten using these functions as:

$$\begin{aligned} E(X_x \mid k) &= 2F_0(k,p) + 4F_2(k,p) & E(X_y \mid k) &= 4F_1(k,p) + 2F_2(k,p) \\ E(Y_x \mid k) &= 2F_1(k,p) - 2F_2(k,p) + kp & E(Y_y \mid k) &= 3F_0(k,p) - F_1(k,p) + kp \end{aligned}$$

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

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

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

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

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

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

This concludes the proof. □

Example 26. 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 $T_{n,3}$.

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

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

$$\begin{aligned} E(X_x|3) &= 2 - \frac{12}{(n+1)} + \frac{36}{(n+1)^2} - \frac{48}{(n+1)^3} \\ E(X_y|3) &= \frac{36}{(n+1)} + \frac{162}{(n+1)^2} - \frac{162}{(n+1)^3}. \end{aligned}$$

From lemma 12, $E(X_{t+1} \mid X_t, Y_t, k) = E(X_x \mid k)\frac{X_t}{2} + E(X_y \mid k)\frac{Y_t}{3}$ then

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

Using these expected values we recover the 3-OneStep transition matrix given in 7.

Lemma 27. *The coefficients of the power expansion of the* MinMacroSplit *algorithm for $j > 2$ verify $\alpha_{j+6} = 12^3\alpha_j$ and $\beta_{j+6} = 12^3\beta_j$*

Proof. We prove the relation for α_j (β_j can be proved in a similar maner). $\alpha_{j+6} = 12^3\alpha_j$ if $\mathcal{B}_{j+6}\mathcal{Y}_x(i) = 3^3\mathcal{B}_j\mathcal{Y}_x(i)$ for $j > 2$. We prove this relation by induction on j. It holds for $j = 3, 4, \ldots, 8$. We assume for $2 < k < j$ that $\mathcal{B}_{k+6}\mathcal{Y}_x(i) = 3^3\mathcal{B}_k\mathcal{Y}_x(i)$ and we should demonstrate that $\mathcal{B}_{j+6}\mathcal{Y}_x(i) = 3^3\mathcal{B}_j\mathcal{Y}_x(i)$. By applying lemma 4 this relation holds if for $\ell = 0, \ldots, 6$

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

- $\ell = 0$: is verified by induction.
- $\ell = 1$: By lemma 3

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

  But $\mathcal{B}_{j+1}\mathcal{Y}_x(i) = 3^3\mathcal{B}_{j-5}\mathcal{Y}_x(i)$ and $\mathcal{B}_j\mathcal{Y}_x(i) = 3^3\mathcal{B}_{j-6}\mathcal{Y}_x(i)$.
- $\ell = 2, 3$: It can be proved in a similar maner by applying lemma 3.
- $\ell = 4, 5, 6$: These cases can be reduced to previous ones because $\mathcal{Y}_x(i+\ell) = \mathcal{Y}_x(i+\ell-3) + 3$.

□

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 prove that the coefficients $b(i, k, p)$ of the binomial distribution decrease quickly because $p << 1$:

Lemma 28. *If $p^{-1} \geq 1 + ck$ then $b(i, k, p) \geq cb(i+1, k, p)$ 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 $p^{-1} \geq 1 + ck$ the lemma holds. □

The following lemma recalls that the MaxMacroSplit and MinMacroSplit algorithms generates more and less x-nodes that the PVW algorithm:

Lemma 29. *Let A be a random tree, then*

$$E(X|A, k, \textit{MaxSplit}) \geq E(X \mid A, k, \textit{PVW})$$
$$E(X|A, k, \textit{MinSplit}) \leq E(X \mid A, k, \textit{PVW})$$

Lemma 30. *Let A and B be two random trees with $n+1$ leaves with X_A and X_B leaves of 1-type such that $E(X_A) \geq E(X_B)$, then after inserting k new random keys with the MaxMacroSplit or MinMacroSplit algorithm it holds $E(X|A,k) \geq E(X|B,k)$.*

Proof. Recall that

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

Then $E(X|A,k) - E(X|B,k) \geq 0$ if $\frac{3}{2}E(X_x|k) \geq E(X_y|k)$. We verify this last inequality for both algorithms.

MaxMacroSplit algorithm: Recall the functions F_0 and F_1 from lemma 22. By lemma 21 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 if greater than one and the right term is less than one. But by lemma 28 for $p^{-1} \geq 1 + 2k$ it holds $b(i,k,p) \geq 2b(i+1,k,p)$ and then $F_0(k,p) \geq 2F_1(k,p)$. As $p = \frac{2}{n+1}$ then at least $n \geq 4k+1$.

MinMacroSplit algorithm: Recall the functions F_0, F_1 and F_2 from lemma 25. By lemma 24 the inequality become

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

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 28 this inequality holds if at least $n \geq 6k+2$.

□

Let X_t^{PVW}, Y_t^{PVW} be the fringe distribution before the algorithm starts and let X_{t+1}^{PVW}, Y_{t+1}^{PVW} be the fringe once the algorithm has finished. A bound is given in the following theorem.

Theorem 31. *Let X_t^{MaxSplit}, Y_t^{MaxSplit} be the fringe in the MaxMacroSplit algorithm and X_t^{MinSplit}, Y_t^{MinSplit} be the fringe in the MinMacroSplit algorithm. Let X_t^{PVW}, Y_t^{PVW} be the fringe in the PVW algorithm, we have:*

$$E(X_t^{\mathsf{MinSplit}} \mid k) \leq E(X_t^{\mathsf{PVW}} \mid k) \leq E(X_t^{\mathsf{MaxSplit}} \mid k)$$
$$E(Y_t^{\mathsf{MaxSplit}} \mid k) \leq E(Y_t^{\mathsf{PVW}} \mid k) \leq E(Y_t^{\mathsf{MinSplit}} \mid k)$$

Proof. We prove the inequalities by induction on t. Recall that $E(X|A, k, \mathsf{MaxSplit})$ means the expected value of X when k keys have been inserted into a random tree A with the MaxSplit algorithm.

For $t = 1$, let A be a random tree, then by lemma 29

$$E(X_1^{\mathsf{PVW}} \mid k) = E(X|A, k, \mathsf{PVW}) \leq E(X|A, k, \mathsf{MaxSplit}) = E(X_1^{\mathsf{MaxSplit}} \mid k)$$

For $t > 1$ it holds by induction that $E(X_{t-1}^{\mathsf{PVW}} \mid k) \leq E(X_{t-1}^{\mathsf{MaxSplit}} \mid k)$ and we should demonstrate that $E(X_t^{\mathsf{PVW}} \mid k) \leq E(X_t^{\mathsf{MaxSplit}} \mid k)$. Let B_{t-1} and C_{t-1} be the random trees generated after inserting k keys $t-1$ times with the PVW and MaxSplit algorithm. Then by lemma 29

$$E(X_t^{\mathsf{PVW}} \mid k) = E(X|k, B_{t-1}, \mathsf{PVW}) \leq E(X|k, B_{t-1}, \mathsf{MaxSplit}).$$

By lemma 30 and the hypothesis of induction

$$E(X|k, B_{t-1}, \mathsf{MaxSplit}) \leq E(X|k, C_{t-1}, \mathsf{MaxSplit}) = E(X_t^{\mathsf{MaxSplit}} \mid k).$$

□

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 31) and we have developed the power expansion of these last two algorithms (lemmas 21 to 27). Then, for any number of keys its is possible to bound the expected number of leaves of two types generated by the PVW algorithm.

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 then 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.

References

[AHJ74] A.V. Aho, J.E. Hopcroft, and Ullman J.D. *The design and analysis of computer algorithms.* Addison-Wesley, 1974.

[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.

[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.

[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.

[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 International Colloquium on Automata, Languages and Programming, LNCS 154*, pages 597–609. Springer-Verlag, 1983. Also appeared as "Parallel computation on 2–3 trees" in RAIRO Informatique Théorique, pages 397-404, 1983.

[Yao78] A.C-C. Yao. On random 2-3 trees. *Acta Informatica*, 9(2):159–170, 1978.