An Algebraic Approach to the Prefix Model Analysis of Binary Trie Structures and Set Intersection Algorithms

EXTENDED ABSTRACT

Pilar de la Torre[1]
Department of Computer Science
University of New Hampshire

David T. Kao[2]
Department of Computer Science
University of New Hampshire

Abstract

The *trie*, or *digital tree*, is a standard data structure for representing sets of strings over a given finite alphabet. Since Knuth's original work [Knu73], this data structures have been extensively studied and analyzed. In this paper, we present an algebraic approach to the analysis of average storage and average time required by the retrieval algorithms of trie structures under the *prefix model*. This approach extends the work of Flajolet et al. for other models which, unlike the prefix model, assume that no key in a sample set is the prefix of another. As the main application, we analyze the average running time of two algorithms for computing set intersections.

Résumé

Le *trie*, ou *trie numérique*, est une structure de données standard pour représenter des ensembles de mots sur un alphabet fini donné. Depuis le travail original de Knuth [Knu73], ces structures de données ont été intensivement étudiées et analysées. Dans cet article, nous présentons une approche algébrique de l'analyse de l'espace moyen et du temps moyen requis par les algorithmes de recherche dans les structures de trie utilisant le *modèle préfixe.* Cette approche étend le travail de Flajolet et al. à d'autres modèles, qui, contrairement au modèle préfixe, supposent qu'aucune clef dans un échantillon n'est le préfixe d'une autre. Comme principale application, nous analysons le temps d'éxćution moyen de deux algorithmes pour le calcul d'intersections d'ensembles.

Address for correspondence: Pilar de la Torre, Department of Computer Science, University of New Hampshire, Durham, NH 03824, U.S.A.; **e-mail:** dltrr@cs.unh.edu; **phone:** (603) 862-2682, **FAX:** (603) 862-3493.

[1]The research of this author was supported in part by the National Science Foundation under Grant CCR-9010445, and Grant CCR-9410592.

[2]The research of this author was supported in part by the National Science Foundation under Grant IRI-9117153.

1 Introduction

Since Knuth's original analysis [Knu73], the average case performance of *trie* [Fre60], or *digital tree*, data structures has received a great deal of attention (see, for example, [GBY91, FS86]). In particular, systematic approaches to their analysis under several probability models have been developed (see, for example, [dlT87, Fra77, Fla83, FRS85, VF87]). All these models, however, preclude the possibility of sets in which the key of one element is a prefix of that of another.

The design of tries for storing sets of keys that may contain *prefixing keys* (that is, keys that are prefixes of other keys in the set), was taken up by Knott in [Kno86]. The first analysis of tries that store prefixing keys was done in [dlT87] under the *prefix model* which generalizes Trabb Pardo's model [Tra78] and is defined as follows.

Definition 1.1 PREFIX MODEL. The prefix model $\mathcal{P}(h, n, m)$ assumes as equally likely all sets of n strings with length at most h over an alphabet $\mathcal{A}$ of m characters. That is, all n-elements subsets of $\mathcal{A}^0 \cup \mathcal{A}^1 \cup \ldots \cup \mathcal{A}^h$.

In this paper, we present an algebraic approach to the analysis of trie structures for sets of binary strings under the prefix model, which extends the work done by Flajolet et al. in [FRS85] for other models. As the main application of this approach, we analyzed the average running time of two algorithms for computing intersections of sets of binary strings under the *set–intersection prefix model* (defined in §4) which generalizes Trabb Pardo's set–intersection model [Tra78].

Section §2 introduces the *root-function method* – a uniform approach to deriving the expectations of a wide class of random variables under the prefix model. Section §3 drives the generating function translation rules corresponding to the prefix model. Section §4 illustrates the use of these rules by applying them to compute the average space and time requirements of the retrieval algorithms of two trie varieties analyzed in [dlTK94b]: full prefixing-tries and compact prefixing-tries. Applying these rules, section §5 calculates the exact average running time of the algorithms for computing set intersections. To shorten the exposition, only the summary of our results are put into this extended abstract. All proofs are given in the full version of our paper [dlTK94a].

2 The Prefix Model

Let $\mathcal{A}$ be a totally ordered alphabet of m (≥ 2) symbols that we will identify with $\mathcal{A} = \{1, \ldots, m\}$, where $1 < 2 < \ldots < m$. Let $\mathcal{A}^{[h]} := \mathcal{A}^0 \cup \mathcal{A}^1 \cup \ldots \mathcal{A}^h$ be the set of all strings of length $\leq h$ composed from $\mathcal{A}$. The set of finite length strings composed from $\mathcal{A}$ will be denoted by $\mathcal{A}^*$, the set of infinitely long strings by $\mathcal{A}^\infty$, and $\mathcal{A}^{\circledast} := \mathcal{A}^* \cup \mathcal{A}^\infty$. For a finite set B, $\mathcal{R}_n(B)$ will denote the set of n-element subsets of B, and $\mathcal{R}(B) := \bigcup_{n \geq 0} \mathcal{R}_n(B)$.

For the integer–valued parameters h, n, m, with $h, n \geq 0$ and $m \geq 2$, the probability space for the prefix model consists of the n-element subsets of $\mathcal{A}^{[h]}$, which are assumed to be equally probable. We have

$$m^{[h]} := |\mathcal{A}^{[h]}| = \frac{m^{h+1} - 1}{m - 1}, \qquad \text{and} \qquad |\mathcal{R}_n(\mathcal{A}^{[h]})| = \binom{m^{[h]}}{n}.$$

Throughout this section X will denote a real–valued function of finite subsets $\xi \subseteq \mathcal{A}^{[h]}$. The expected value of $X(\xi)$ over the n-element subsets $\xi \subseteq \mathcal{A}^{[h]}$ will be denoted by $E[X]$, and also by $E_{hn}[X]$ when we wish to emphasize its dependence on h and n. The sum

$$N_{hn}[X] := \sum_{\xi \in \mathcal{R}_n(\mathcal{A}^{[h]})} X(\xi)$$

is related to the expectation of X by $N_{hn}[X] = \binom{m^{[h]}}{n} E_{hn}[X]$, and will be called the *normalized expectation* of X.

2.1 Translation Rules

To each real–valued function X of subsets $\mathcal{A}^{[h]}$ we associate its *generating function of the normalized expectations* $X^{(h)}(x)$,

$$X^{(h)}(x) := \sum_{0 \le n \le m^{[h]}} N_{hn}[X]\, x^n = \sum_{\xi \in \mathcal{R}(\mathcal{A}_m^{[h]})} X(\xi)\, x^{|\xi|}.$$

Our intention is to establish rules that often help in translating a function X into its generating function $X^{(h)}(x)$. These translation rules will be formulated as properties of the operator $F_h[X] := X^{(h)}(x)$, which maps real–valued functions of subsets $\mathcal{A}_m^{[h]}$ to polynomials in x.

We introduce the family of auxiliary functions P_x, with $x \in \mathcal{A}^*$. The value of P_x on a subset $\xi \subset \mathcal{A}^{\circledast}$ is $P_x(\xi) := \xi_x$, where $\xi_x := \{y \mid xy \in \xi\}$ (i.e., ξ_x is the set of tails of the strings of ξ that begin with x). For each $c \in \mathcal{A}$, P_c maps $\mathcal{R}(\mathcal{A}^{[h]})$ onto $\mathcal{R}(\mathcal{A}^{[h-1]})$. We also define the function $P_\perp(\xi) := \xi \cap \{\varepsilon\}$, which maps $\mathcal{R}(\mathcal{A}^{[h]})$ onto $\mathcal{R}(\{\varepsilon\})$.

Lemma 2.1 ADDITIVE–MULTIPLICATIVE RULE. *Let X, Y, Y_0, Y_1,…,Y_m be real–valued functions of subsets of $\mathcal{A}^{[h]}$. Then,*

(i) $F_h[\lambda.X] = \lambda F_h[X]$;

(ii) $F_h[X+Y] = F_h[X] + F_h[Y]$;

(iii) For $h \ge 1$,

$$F_h[(Y_1 \circ P_1)\dots(Y_m \circ P_m)] = (1+x)\, F_{h-1}[Y_1]\dots F_{h-1}[Y_m];$$

(iv) For $h \ge 1$,

$$F_h[(Y_0 \circ P_\perp)(Y_1 \circ P_1)\dots(Y_m \circ P_m)] = F_0[Y_0]\, F_{h-1}[Y_1]\dots F_{h-1}[Y_m].$$

Lemma 2.2 INITIALIZATION RULE. *Let $I(\xi) := 1$, and $C(\xi) := |\xi|$. Then*

(i) $F_h[I] = (1+x)^{m^{[h]}}$;

(ii) $F_h[C] = m^{[h]}\, x\, (1+x)^{m^{[h]}-1}$;

(iii) If $X(\xi) = \delta_{|\xi|,p}$ *then* $F_h[X] = \binom{m^{[h]}}{p} x^p$.

Theorem 2.3 *Let X and Y be real-valued functions of subsets of $\mathcal{A}^{[h]}$.*

(i) If $X(\xi) = Y(\xi \cap \{\varepsilon\})$ then $X^{(h)}(x) = (1+x)^{m^{[h]}-1}\, Y^{(0)}(x)$;

(ii) If $X = Y \circ P_c$, with $c \in \mathcal{A}$, then $X^{(h)}(x) = (1+x)^{m^h} Y^{(h-1)}(x)$;

(iii) Let $r_X(\xi) := X(\xi) - X(\xi_1) - \cdots - X(\xi_m)$. Then,

$$X^{(h)}(x) \;=\; r_X^{(h)}(x) + m\,(1+x)^{m^h}\, X^{(h-1)}(x). \tag{1}$$

Lemma 2.4 (*Flajolet–Regnier–Sotteau*) Iteration Rule. *Let $A_1, \ldots, A_h$ and $B_0, \ldots, B_h$ be polynomials. The solution to the recurrence $z_0 = B_0$,*

$$z_h \;=\; A_h z_{h-1} + B_h \qquad (h > 0), \tag{2}$$

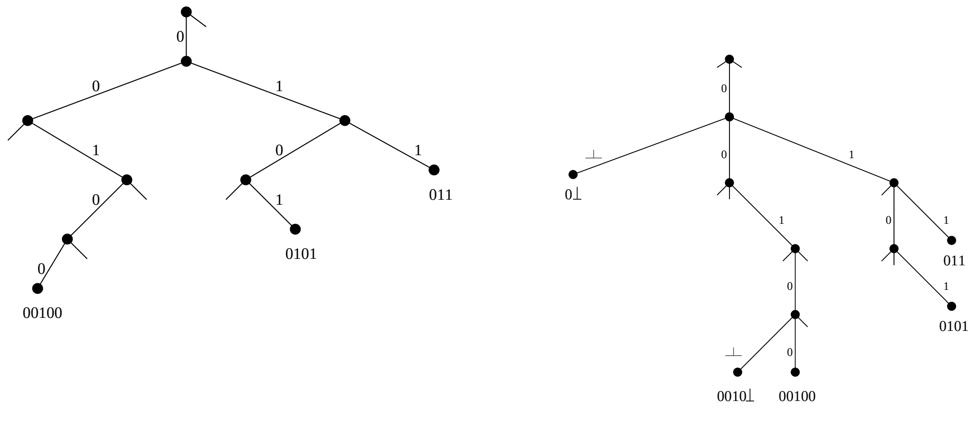
is $z_h = \sum_{0\le j\le h}[B_j \prod_{j+1\le k\le h} A_k]$.

Theorem 2.5 *Let X be a real-valued function of subsets of $\mathcal{A}^{[h]}$ and let $r_X(\xi) := X(\xi) - X(\xi_1) - \ldots - X(\xi_m)$. Then,*

$$X^{(h)}(x) \;=\; \sum_{0\le j\le h} m^{h-j}\,(1+x)^{m^{[h]}-m^{[j]}}\, r_X^{(j)}(x). \tag{3}$$

3 Analysis of Prefixing-Tries

This section presents the data structures used by the set intersection algorithms that will be presented later in §5. *Prefixing-tries*, which are natural adaptations of the original tries of Fredkin [Fre60] for the purpose of storing sets of keys that may contain prefixing keys, have been analyzed in [dlT87, dlTK94b]. Applying the generating function tools of §2, we shall now rederive the exact average space and time requirements of the retrieval algorithms of *full prefixing-tries* and *compact prefixing-tries*.



(a) (b)

Figure 1. (a) Prefix tree built from the set of keys $s = \{00100, 0101, 011\}$. (b) Prefixing-tree built from $s = \{00100, 0101, 011, 0010, 0\}$, which can not be represented by a prefix tree.

Tries are implementations of the *prefix tree* (see Figure 1(a)). A finite set of keys $\xi \subset \mathcal{A}^{\circledast}$, which may include prefixing keys, can be easily encoded to yield a suitable representation of ξ as a trie. This can be attained by attaching a symbol $\perp \notin \mathcal{A}$, the *endmarker*, to the end of the prefixing keys of ξ. In the resulting set of keys, $\xi[\perp] := (\xi - \mathit{prefixingkeys}(\xi)) \cup \{x\perp \mid x \in \mathit{prefixingkeys}(\xi)\}$, no key is a prefix of another. $\xi[\perp]$ can be represented as an *prefixing-tree* (see Figure 1(b)).

3.1 Full Prefixing-Tries

Definition 3.1 FULL PREFIXING-TRIE. *The full prefixing-trie built with a finite set of keys $\xi \subset \mathcal{A}^*$ is the $(m+1)$–ary tree, denoted by $t^{fe}(\xi)$, which is recursively defined as follows:*

(i) If ξ is empty, $t^{fe}(\xi)$ is the empty tree.

(ii) If $\xi = \{\varepsilon\}$, $t^{fe}(\xi)$ is the tree whose root is a leaf node (i.e., all its subtrees are empty).

(iii) Otherwise, $t^{fe}(\xi)$ is the $(m+1)$–ary tree having an 'internal' root node whose subtrees are $t^{fe}(\xi \cap \{\varepsilon\})$, $t^{fe}(\xi_1), \ldots, t^{fe}(\xi_m)$ in order.

See Figure 2(a) for an example of a full prefixing-trie. For a given full prefixing-trie, we use the notation S_f and T_f to represent the number of internal nodes and its total leaf node path length.

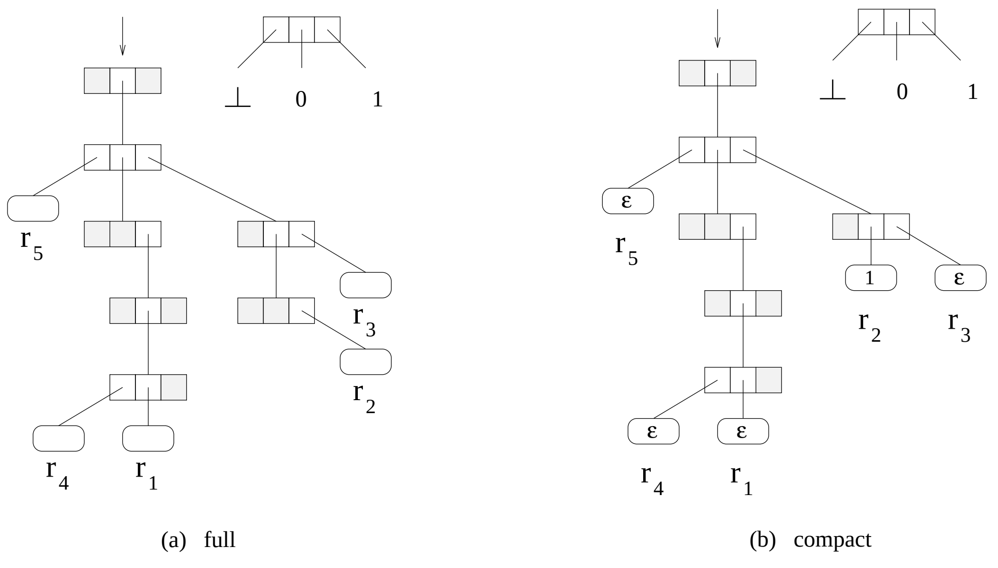


Figure 2. Prefixing-tries for the items $r_1, \ldots, r_5$ with respective keys $k_1 = 00100$, $k_2 = 0101$, $k_3 = 011$, $k_4 = 0010$, $k_5 = 0$. The alphabet is $\{\perp, 0, 1\}$, with $\perp < 0 < 1$.

Theorem 3.2 *The expectations of S_f and T_f over the n-element subsets of $\mathcal{A}^{[h]}$ are*

$$\begin{aligned} E[S_f] &= \sum_{1\le j\le h} m^{h-j}[1-\tau(m^{[h]},m^{[j]},n,0)-\tau(m^{[h]},m^{[j]},n,1)], \\ E[T_f] &= \sum_{1\le j\le h} m^{h-j}\Big[\frac{n}{m^{[h]}}m^{[j]}-\tau(m^{[h]},m^{[j]},n,1)\Big], \end{aligned}$$

where $\tau(a,b,c,d)=\frac{\binom{a-b}{c-d}}{\binom{a}{b}}$.

3.2 Compact Prefixing-Tries

See Figure 2(b) for an example of a compact prefixing-trie. For a given compact prefixing-trie, we use the notation S_c and T_c to represent the number of internal nodes and its total leaf node path length.

Theorem 3.3 *The expectations of S_c and T_c over the n-element subsets of $\mathcal{A}^{[h]}$ are*

$$\begin{aligned} E[S_c] &= \sum_{1\le j\le h} m^{h-j}[1-\tau(m^{[h]},m^{[j]},n,0)-m^{[j]}\tau(m^{[h]},m^{[j]},n,1)], \\ E[T_c] &= \sum_{1\le j\le h} m^{h-j}m^{[j]}\Big[\frac{n}{m^{[h]}}-\tau(m^{[h]},m^{[j]},n,1)\Big], \end{aligned}$$

where $\tau(a,b,c,d)=\frac{\binom{a-b}{c-d}}{\binom{a}{b}}$.

4 The Binary Set–Intersection Prefix Model

Definition 4.1 Binary Set–Intersection Prefix Model. *The sample space of the binary set–intersection prefix model consists of a class of ordered pairs (ξ,η) of sets of binary string keys. This class depends on four parameters: the size l of the first component ξ, the size n of the second component η, the size k of the intersection $\xi\cap\eta$, and the maximum length h of the of the binary string keys. For nonnegative integers h, l, n, and k, the probability space of the binary set–intersection prefix model is*

$$\mathcal{M}_{h,l,n,k} := \{\,(\xi,\eta)\mid \xi,\eta\subset\{0,1\}^{[h]},\, |\xi|=l,\ |\eta|=n,\ |\xi\cap\eta|=k\,\},$$

where $\{0,1\}^{[h]}=\{0,1\}^0\cup\{0,1\}^1\cup\ldots\{0,1\}^h$, and all set pairs (ξ,η) are assumed to be equally probable.

The expectation of a real–valued mapping $X(\xi,\eta)$ over the pairs $(\xi,\eta)\in\mathcal{M}_{h,l,n,k}$ will be denoted by $E[X]$. The sum

$$N_{h,l,n,k}[X] := \sum_{\substack{\xi,\eta\subset\{0,1\}^{[h]}\\ |\xi|=l,\ |\eta|=n,\ |\xi\cap\eta|=k}} X(\xi,\eta)$$

is related to the expectation of X by $N_{h,l,n,k}[X]=|\mathcal{M}_{h,l,n,k}|E[X]$ and will be called the *normalized expectation* of X.

4.1 Translation Rules

Throughout this section X will denote a real–valued mapping of ordered pairs (ξ, η) of sets $\xi, \eta \subseteq \{0,1\}^{[h]}$. To each such mapping X we associate the *generating function of normalized expectations* $X^{(h)}(x, y, t)$,

$$\begin{aligned} X^{(h)}(x, y, t) &:= \sum_{\xi,\eta \subset \{0,1\}^{[h]}} X(\xi, \eta)\, x^{|\xi|}\, y^{|\eta|}\, t^{|\xi \cap \eta|} \\ &= \sum_{l,n,k \geq 0} N_{h,l,n,k}[X]\, x^l\, y^n\, t^k. \end{aligned}$$

We shall now establish translation rules, between X and its generating function $X^{(h)}(x, y, t)$, similar to those derived for the prefix model in §2.1. As earlier, the translation rules will be formulated as properties of the operator F_h which maps a functions X into its generating function $F_h[X] := X^{(h)}(x, y, t)$. Some of these properties can be conveniently expressed in terms of the mappings $\overline{P}_c(\xi, \eta) := (\xi_c, \eta_c)$ (where $\xi_c = \{x \mid cx \in \xi\}$) for each $c \in \{0,1\}$, and also $\overline{P}_\perp(\xi, \eta) := (\xi \cap \{\varepsilon\}, \eta \cap \{\varepsilon\})$.

Lemma 4.2 ADDITIVE–MULTIPLICATIVE RULE. *Let X, Y, and Z be real–valued mappings of ordered pairs (ξ, η) of sets $\xi, \eta \subset \{0,1\}^{[h]}$.*

(i) $F_h[\lambda.X] = \lambda F_h[X]$;

(ii) $F_h[X + Y] = F_h[X] + F_h[Y]$;

(iii) $F_h[(Y \circ \overline{P}_0).(Z \circ \overline{P}_1)] = (1 + x + y + xyt)\, F_{h-1}[Y]\, F_{h-1}[Z]$, $h \geq 1$;

(iv) $F_h[(X \circ \overline{P}_\perp).(Y \circ \overline{P}_0).(Z \circ \overline{P}_1)] = F_0[X]\, F_{h-1}[Y]\, F_{h-1}[Z]$, $h \geq 1$.

Lemma 4.3 INITIALIZATION RULE. *If $I(\xi, \eta) := 1$ then*

$$I^{(h)}(x, y, t) = (1 + x + y + xyt)^{2^{[h]}}.$$

Theorem 4.4 *Let X and Y be real–valued functions of pairs (ξ, η) of subsets $\xi, \eta \subseteq \{0,1\}^{[h]}$, and let us assume that $h \geq 1$.*

(i) If $X(\xi, \eta) = Y(\xi \cap \{\varepsilon\}\, ,\, \eta \cap \{\varepsilon\})$ then $X^{(h)}(x, y, t) = (1 + x)^{2^{[h]}-1}\, Y^{(0)}(x, y, t)$.

(ii) If $X = Y \circ \overline{P}_c$, with $c \in \{0,1\}$, then

$$X^{(h)}(x, y, t) = (1 + x + y + xyt)^{2^h}\, Y^{(h-1)}(x, y, t).$$

(iii) If $r_X(\xi, \eta) := X(\xi, \eta) - X(\xi_0, \eta_0) - X(\xi_1, \eta_1))$ then

$$X^{(h)}(x) = r_X^{(h)}(x) + 2\,(1 + x + y + xyt)^{2^h}\, X^{(h-1)}(x). \tag{4}$$

Theorem 4.5 *If Let X be a real–valued function of pairs (ξ, η) of subsets $\xi, \eta \subseteq \{0,1\}^{[h]}$, and let $r_X(\xi,\eta) = X(\xi,\eta) - X(\xi_0,\eta_0) - X(\xi_1,\eta_1)$. Then,*

$$X^{(h)}(x,y,t) = \sum_{0\le j\le h} 2^{h-j}\,(1+x+y+xyt)^{2^{[h]}-2^{[j]}}\, r_X^{(j)}(x,y,t).$$

5 Analysis of Algorithms for Set Intersection

We now present two algorithms for computing the intersection of sets of binary string keys. For each of them we will compute the exact average running time with respect to the binary set–intersection prefix model.

5.1 Average Set–Intersection Time Using Full Prefixing-Tries

The set intersection $\textsc{intersectf}(\xi,\eta) := \xi \cap \eta$, with $\xi, \eta \subseteq \{0,1\}^{[h]}$, can be computed by the following algorithm:

[Set-Intersection Algorithm Using Full Prefixing-Tries]

1. If $|\xi| = 0$ or $|\eta| = 0$ then $\textsc{intersectf}(\xi,\eta) \leftarrow \emptyset$;
2. If $\xi = \{\varepsilon\}$ then $\textsc{intersectf}(\xi,\eta) \leftarrow \xi$;
3. If $\eta = \{\varepsilon\}$ then $\textsc{intersectf}(\xi,\eta) \leftarrow \eta$;
4. Otherwise,

$$\textsc{intersectf}(\xi,\eta) \leftarrow (\xi\cap\eta\cap\{\varepsilon\}) \cup 0\,\textsc{intersectf}(\xi_0,\eta_0) \cup 1\,\textsc{intersectf}(\xi_1,\eta_1).$$

Let $t^{fe}(\xi)$ and $t^{fe}(\eta)$ be the full prefixing-tries built from ξ and η respectively. The total time necessary to compute the intersection is thus proportional to the number, $F(\xi,\eta)$, of pairs of nodes that are simultaneously visited in $t^{fe}(\xi)$ and $t^{fe}(\eta)$ (i.e., $F(\xi,\eta)$ equals the total number of times that Step 4 is executed).

The results of the following lemma will be helpful in extracting coefficients from the generating functions that will emerge from our computations. The coefficient of the term $x^l y^n t^k$ in a polynomial $P(x,y,t)$ will be denoted by $[l,n,k]P(x,y,t)$.

Lemma 5.1 The coefficient

$$K_{l,n,k}[\alpha,\beta] := [l,n,k]\left\{[\,(1+x)^\alpha + (1+y)^\alpha - 1](1+x+y+xyt)^\beta\right\}$$

equals

$$K_{l,n,k}[\alpha,\beta] \;\; = I_{l,n,k}[\alpha,\beta] + I_{n,l,k}[\alpha,\beta] - I_{l,n,k}[0,\beta], \tag{5}$$

where $I_{l,n,k}[\alpha,\beta] := \binom{\beta}{k}\binom{\beta-k}{n-k}\binom{\beta+\alpha-n}{l-k}$. Also, $K_{l,n,k}[0,2^{[h]}] = I_{l,n,k}[0,2^{[h]}] = |\mathcal{M}_{h,l,n,k}|$.

Theorem 5.2 *The expected value of $F(\xi,\eta)$ over the pairs of sets $(\xi,\eta) \in \mathcal{M}_{h,l,n,k}$ is*

$$E[F] = (2^{[h]} - 1) - \frac{1}{|\mathcal{M}_{h,l,n,k}|} \sum_{1 \le j \le h} 2^{h-j} K_{l,n,k}[2^{[j]} - 1 \, , \, 2^{[h]} - 2^{[j]} + 1],$$

where $|\mathcal{M}_{h,l,n,k}| = K_{l,n,k}[0 \, , \, 2^{[h]}]$.

5.2 Average Set–Intersection Time Using Compact Prefixing-Tries

We shall now consider another algorithm for set intersection, which is based on compact prefixing-tries. Let $Part(\alpha,\beta)$ be the function of $\alpha,\beta \subseteq \{0,1\}^{[h]}$ that has the value α when $\alpha \subset \beta$, and the value $\emptyset$ otherwise. The set intersection $\text{INTERSECTC}(\xi,\eta) := \xi \cap \eta$, with $\xi,\eta \subseteq \{0,1\}^{[h]}$, can be computed by the following algorithm:

[Set-Intersection Algorithm Using Compact Prefixing-Tries]

1. If $|\xi| = 0$ or $|\eta| = 0$ then $\text{INTERSECTC}(\xi,\eta) \leftarrow \emptyset$;
2. If $|\xi| = 1$ then $\text{INTERSECTC}(\xi,\eta) \leftarrow Part(\xi,\eta)$;
3. If $|\eta| = 1$ then $\text{INTERSECTC}(\xi,\eta) \leftarrow Part(\eta,\xi)$;
4. Otherwise,

$$\text{INTERSECTC}(\xi,\eta) \leftarrow (\xi \cap \eta \cap \{\varepsilon\}) \cup 0\,\text{INTERSECTC}(\xi_0,\eta_0) \cup 1\,\text{INTERSECTC}(\xi_1,\eta_1).$$

Let $t^{ce}(\xi)$ and $t^{ce}(\eta)$ be the respective compact prefixing-tries of ξ and η, and let us assume that $|\xi|, |\eta| \ge 2$. Then, $\varepsilon \in \xi \cap \eta$ precisely when the first sons of $t^{ce}(\xi)$ and $t^{ce}(\eta)$ are nonempty. The sets ξ_0 and η_0 are represented by the respective second subtrees of $t^{ce}(\xi)$ and $t^{ce}(\eta)$; ξ_1 and η_1 are represented by the third subtrees of $t^{ce}(\xi)$ and $t^{ce}(\eta)$.

The algorithm $\text{INTERSECTC}(\xi,\eta)$ can thus be implemented by the simultaneous traversal of the compact prefixing-tries $t^{ce}(\xi)$ and $t^{ce}(\eta)$. We start at the root nodes of the tries, and implement Step 1 by testing whether one of the trees is empty. Step 2 (respectively Step 3) is realized by testing whether the root node of $t^{ce}(\xi)$ (respectively $t^{ce}(\eta)$) is a terminal node. If it is, i.e., $\xi = \{x\}$ (respectively $\eta = \{y\}$), $Part(\{x\},\eta)$ (respectively $Part(\xi,\{y\})$) is implemented by searching for the key x in $t^{ce}(\eta)$ (respectively searching for y in $t^{ce}(\xi)$). If this search is successful, we return the value $\{x\}$ (respectively $\{y\}$); otherwise, we return the value $\emptyset$. Since Step 4 is executed precisely when $|\xi|, |\eta| \ge 2$, we can then compute $\xi \cap \eta \cap \{\varepsilon\}$ by simply examining the first subtrees of $t^{ce}(\xi)$ and $t^{ce}(\eta)$ (these subtrees are terminal nodes precisely when $\varepsilon \in \xi \cap \eta$). The recursive call $\text{INTERSECTC}(\xi_0,\eta_0)$ (respectively $\text{INTERSECTC}(\xi_1,\eta_1)$) can be implemented by simultaneously visiting the second sons (respectively third sons) of $t^{ce}(\xi)$ and $t^{ce}(\eta)$, which are the root nodes of compact prefixing-tries representing the sets ξ_0 and η_0 (respectively ξ_1 and η_1).

The time required to compute $\xi \cap \eta$ by the above algorithm is proportional to $C(\xi,\eta)$, which is defined as the number of pairs of internal nodes simultaneously visited in tries $t^{ce}(\xi)$ and $t^{ce}(\eta)$ (i.e., the number of times that Step 4 is executed) plus the number of internal nodes visited in only one of the tries after a terminal node has been reached in the other (i.e., the number of nodes visited while executing the calls to $Part$).

We shall calculate the expectation of C in two ways. Our first calculation makes use of the relation between full and compact prefixing-tries. That is, the compact prefixing-trie $t^{ce}(\xi)$ results from the full prefixing-trie $t^{fe}(\xi)$ by pruning every internal node that has only one terminal node among its descendants. Hence, $M(\xi,\eta) := F(\xi,\eta) - C(\xi,\eta)$ is equal to the number of pairs of internal nodes of $t^{fe}(\xi)$ and $t^{fe}(\eta)$ simultaneously visited, in the implementation of INTERSECTF(ξ,η) given in §5.1, such that each internal node in the pair has only one terminal among its descendants. Thus the function $r_M(\xi,\eta) := M(\xi,\eta) - M(\xi_0,\eta_0) - M(\xi_1,\eta_1)$ has the expression $r_M(\xi,\eta) = \delta_{(|\xi|=1)\, and\, (\xi\neq\{\varepsilon\})}\, \delta_{(|\eta|=1)\, and\, (\eta\neq\{\varepsilon\})}$.

Theorem 5.3 The expectation of $M(\xi,\eta)$ over the pairs $(\xi,\eta)\in\mathcal{M}_{h,l,n,k}$ is

$$E[M] = \tfrac{1}{|\mathcal{M}_{h,l,n,k}|}\textstyle\sum_{1\le j\le h} 2^{h-j}\,(2^{[j]}-1)\quad \Big\{K_{l-1,n-1,k-1}[0\,,\,2^{[h]}-2^{[j]}] + (2^{[j]}-2)\,K_{l-1,n-1,k}[0\,,\,2^{[h]}-2^{[j]}]\Big\},$$

where $|\mathcal{M}_{h,l,n,k}| = K_{l,n,k}[0\,,\,2^{[h]}]$.

Theorem 5.4 *The average total time $E[C]$ required to compute the intersection using compact prefixing-tries is*

$$E[C] = (2^{[h]}-1) - \frac{1}{|\mathcal{M}_{h,l,n,k}|}\Bigg\{\sum_{1\le j\le h} 2^{h-j}K_{l,n,k}[2^{[j]}-1\,,\,2^{[h]}-2^{[j]}+1] + \sum_{1\le j\le h} 2^{h-j}(2^{[j]}-1)\Big[K_{l-1,n-1,k-1}[0\,,\,2^{[h]}-2^{[j]}] + (2^{[j]}-2)K_{l-1,n-1,k}[0\,,\,2^{[h]}-2^{[j]}]\Big]\Bigg\},$$

where $|\mathcal{M}_{h,l,n,k}| = K_{l,n,k}[0\,,\,2^{[h]}]$.

The following alternative way of computing $E[C]$ yields additional information of interest to the cost analysis. We break up the values of the function C into two components,

$$C(\xi,\eta) = A(\xi,\eta) + B(\xi,\eta). \tag{6}$$

The first component, $A(\xi,\eta)$, is the number of pairs internal nodes of $t^{ce}(\xi)$ and $t^{ce}(\eta)$ that are simultaneously visited in the implementation of of the above algorithm for INTERSECTC(ξ,η) (i.e., the number of times Step 4 is is executed). This quantity is of interest in its own right since, as remarked by Trabb Pardo in [Tra78], $A(\xi,\eta)$ measures the risk of computing the intersection $\xi\cap\eta$ to find that it is empty. The second component, $B(\xi,\eta)$, is the number of internal nodes visited in only one of the tries after an internal node has been encountered in the other (i.e., the number of nodes visited in the execution of the calls to $Part(\xi,\eta)$).

Since Step 4 is executed precisely when $|\xi|,|\eta|\ge 2$, $r_A(\xi,\eta) := A(\xi,\eta) - A(\xi_0,\eta_0) - A(\xi_1,\eta_1)$ can be written as

$$r_A(\xi,\eta) \;=\; 1 - Z(\xi,\eta), \tag{7}$$

with $Z(\xi,\eta)=\delta_{(|\xi|\le 1)\, or\, (|\eta|\le 1)}$. We further observe that an internal node v of $t^{ce}(\xi)$ (respectively $t^{ce}(\eta)$) is visited in the process of executing the function $Part(\xi,\eta)$ (respectively $Part(\eta,\xi)$) precisely when the string x, corresponding to the path that connects the root and v, satisfies $|\xi_x|\ge 2$ and $|\eta_x|=1$ (respectively $|\eta_x|\ge 2$ and $|\xi_x|=1$). Thus, $r_B(\xi,\eta):=B(\xi,\eta)-B(\xi_0,\eta_0)-B(\xi_1,\eta_1)$ has the expression

$$r_B(\xi,\eta) \quad = \quad \delta_{|\xi|=1}\,\delta_{\xi\ne\{\varepsilon\}}\,\delta_{|\eta|\ge 2}+\delta_{|\eta|=1}\,\delta_{\eta\ne\{\varepsilon\}}\,\delta_{|\xi|\ge 2}. \tag{8}$$

Theorem 5.5 *The expectation of $A(\xi,\eta)$ over the pairs $(\xi,\eta)\in\mathcal{M}_{h,l,n,k}$ is*

$$\begin{aligned} E[A] \quad &= \quad (2^{[h]}-1)-\frac{1}{|\mathcal{M}_{h,l,n,k}|} \\ &\Big\{\sum_{1\le j\le h} 2^{h-j}2^{[j]}\Big[K_{l,n,k}[2^{[j]}-1\,,\,2^{[h]}-2^{[j]}+1]-(2^{[j]}-1)\,K_{l-1,n-1,k}[0\,,\,2^{[h]}-2^{[j]}]\Big] \\ &-\sum_{1\le j\le h} 2^{h-j}(2^{[j]}-1)\,K_{l,n,k}[2^{[j]}\,,\,2^{[h]}-2^{[j]}]\Big\}, \end{aligned}$$

with $|\mathcal{M}_{h,l,n,k}|=K_{l,n,k}[0\,,\,2^{[h]}]$.

Theorem 5.6 *The expectation of $B(\xi,\eta)$ over the pairs $(\xi,\eta)\in\mathcal{M}_{h,l,n,k}$ is*

$$\begin{aligned} E[B]=\tfrac{1}{|\mathcal{M}_{h,l,n,k}|}\Big\{\textstyle\sum_{1\le j\le h}\quad & 2^{h-j}(2^{[j]}-1)\Big[K_{l,n,k}[2^{[j]}-1\,,\,2^{[h]}-2^{[j]}+1]-K_{l,n,k}[2^{[j]}\,,\,2^{[h]}-2^{[j]}] \\ & -K_{l-1,n-1,k-1}[0\,,\,2^{[h]}-2^{[j]}]-2\,(2^{[j]}-1)K_{l-1,n-1,k}[0\,,\,2^{[h]}-2^{[j]}]\Big]\Big\}, \end{aligned}$$

where $|\mathcal{M}_{h,l,n,k}|=K_{l,n,k}[0\,,\,2^{[h]}]$.

References

[dlT87] P. de la Torre. Analysis of tries. Ph.D. Thesis CS–TR–1890, Department of Computer Science, University of Maryland, 1987.

[dlTK94a] P. de la Torre and D. T. Kao. An algebraic approach to the prefix model analysis of binary trie structures and set intersection algorithms. Technical Report 94–18, Department of Computer Science, University of New Hampshire, 1994. Submitted to *7th International Conference on Formal Power Series and Algebraic Combinatorics.*

[dlTK94b] P. de la Torre and D. T. Kao. A uniform approach to the analysis of trie structures that store prefixing keys. Technical Report 94–17, Department of Computer Science, University of New Hampshire, 1994. Submitted to *Journal of Algorithms.*

[Fla83] Ph. Flajolet. Methods in the analysis of algorithms: Evaluation of a recursive partitioning process. In M. Karpinski, editor, *Proceedings of the 1983 International FCT–Conference*, number 158 in Lecture Notes in Computer Science, pages 141–158, Borgholm, Sweden, 1983. Springer-Verlag.

[Fra77] J. Françon. On the analysis of algorithms for trees. *Theoretical Computer Science*, 4:155–169, 1977.

[Fre60] E. Fredkin. Trie memory. *CACM*, 3(9):490–499, 1960.

[FRS85] Ph. Flajolet, M. Regnier, and D. Sotteau. Algebraic methods for trie statistics. *Annals of Discrete Mathematics*, 25:145–188, 1985.

[FS86] Ph. Flajolet and R. Sedgewick. Digital search trees revisited. *SIAM J. Comput.*, 15(3):748–767, 1986.

[GBY91] G. H. Gonnet and R. Baeza-Yates. *Handbook of Algorithms and Data Structures: In Pascal and C*. Addison–Wesley, Reading, Massachusetts, 2 edition, 1991.

[Kno86] G. D. Knott. Including prefixes in doubly–chained tries. Technical Report CAR–TR–236, Computer Science Department, University of Maryland, 1986.

[Knu73] D. E. Knuth. *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison–Wesley, Reading, Massachusetts, 1973.

[Tra78] L. I. Trabb Pardo. Set representation and set intersection. Ph.D. Thesis STAN–CS–78–681, Department of Computer Science, Stanford University, 1978.

[VF87] J. S. Vitter and P. Flajolet. Average-case analysis of algorithms and data structures. Technical Report CS–87–20, Department of Computer Science, Brown University, Providence, Rhode Island, 1987. Revised April 1989.