Research Article

AND DATA SCIENCES
BULLETIN OF COMPUTER

From Parikh Vectors to Parikh Matrices: Structure, Ambiguity, and Complexity

Arto Salomaa

Turku Centre for Computer Science, University of Turku, Quantum Building 392, 20014 Turun yliopisto, Finland

Abstract

We study Parikh matrices as an order-sensitive refinement of Parikh vectors. We formalize the mapping PM_k and an indicator-based generalization PM_v, showing via a simple encoding that every PM_v reduces to an ordinary Parikh matrix. Structurally, we revisit M-equivalence and (un)ambiguity (including the complete binary characterization and finiteness for prints), and identify cyclic shifts and disjoint transpositions as Parikh-friendly permutations. Algorithmically, M-equivalence is decidable in polynomial time, while recognition and unambiguity admit natural exponential procedures; superdiagonals exhibit substantial independence. Finally, we construct binary families with exponentially large M-equivalence classes and relate their counts to Gaussian binomials, ruling out polynomial upper bounds on ambiguity degree.

Keywords: Parikh matrices, Parikh vectors, M-equivalence, M-ambiguity, subword indicators, scattered subwords, alphabet permutations, decision problems, combinatorics on words, Gaussian binomial coefficients

1. Introduction

Let w be a word over a finite alphabet $\mathcal{A}$ and write $|w|$ for its length. We use $\mathcal{A}^*$ for the free monoid of all (finite) words over $\mathcal{A}$, with identity element the empty word λ. The most immediate statistic of w is its length $|w|$, while its order–free letter frequencies are captured by the classical *Parikh vector* $\Psi(w) \in \mathbb{N}^{|\mathcal{A}|}$ [1]. Fixing an ordering $\mathcal{A} = \{a_1, \ldots, a_n\}$,

$$\Psi(w) = (\,|w|_{a_1}, \ldots, |w|_{a_n}\,),$$

where $|w|_{a_j}$ is the number of occurrences of a_j in w. By construction, $\Psi(w)$ forgets all information about order.

To retain order sensitivity, we count *(scattered) subwords*. If $u = x_1 x_2 \cdots x_k \in \mathcal{A}^*$ and $w = w_1 w_2 \cdots w_{|w|} \in \mathcal{A}^*$, define

$$|w|_u \;:=\; \#\Big\{(i_1, \ldots, i_k) : 1 \le i_1 < \cdots < i_k \le |w|, \quad w_{i_r} = x_r \text{ for } r = 1, \ldots, k\Big\}.$$

Thus $|w|_u$ is the number of embeddings of u into w as a not-necessarily-contiguous subword. By convention, $|w|_\lambda = 1$. We distinguish subwords from *factors* (contiguous subwords): u is a factor of w if $w = xuy$ for some $x, y \in \mathcal{A}^*$. When needed, we write $u \preccurlyeq w$ to mean that u is a subword of w.

Corresponding author: Arto Salomaa (asalomaa@utu.fi)

BCDS 2020, Vol. 1(1), 19-26. DOI: 10.71448/bcds2011-3

Even over the binary alphabet $\{a, b\}$, subword counts satisfy informative identities. A basic example is

$$|w|_a \, |w|_b \;=\; |w|_{ab} + |w|_{ba}\,, \tag{1}$$

see [2], every ordered pair consisting of an a-position and a b-position contributes exactly once to the right–hand side, according to whether the a precedes or follows the b. For repeated letters, scattered copies reduce to binomial coefficients:

$$|w|_{a^m} \;=\; \binom{|w|_a}{m} \qquad (m \geq 0), \tag{2}$$

because choosing an m-tuple of a-positions in increasing order is the same as choosing an m-subset of the $|w|_a$ many a's. Over a ternary alphabet with distinct letters a, b, c, summing over all relative orders yields

$$\sum_{\sigma \in S_3} |w|_{\sigma(abc)} \;=\; |w|_a \, |w|_b \, |w|_c\,, \tag{3}$$

since any triple of positions carrying one a, one b, and one c contributes to exactly one permutation on the left.

Take $w = \texttt{ababa}$. Then $|w|_a = 3$, $|w|_b = 2$,

$$|w|_{ab} = 3, \quad |w|_{ba} = 3, \quad |w|_{aa} = \binom{3}{2} = 3, \quad |w|_{bb} = \binom{2}{2} = 1,$$

and

$$|w|_{aba} = 4, \qquad |w|_{aab} = 1, \qquad |w|_{abb} = 1.$$

Identity (1) holds as $3 \cdot 2 = 3 + 3$, and (2) gives the aa/bb counts directly.

While $\Psi(w)$ already provides a coarse, order–free fingerprint of w, the full family $(\,|w|_u\,)_{u \in \mathcal{A}^*}$ encodes much finer information about order. Between these two extremes lie the *Parikh matrices*, which stack carefully chosen subword counts along superdiagonals of an upper–triangular matrix; they refine $\Psi(w)$ while remaining far more compact than the entire subword profile. They are central to this work.

Beyond the elementary identities above, the landscape of relations that connect first–order counts $(\,|w|_u\,, |w|_v\,)$ with higher–order concatenations $(\,|w|_{uv}\,, |w|_{vu}\,)$ is subtle, and general explicit formulas are not known in full generality [3–12]. This paper develops a calibrated framework for those relations via Parikh matrices and their generalizations, studies ambiguity phenomena (when different words share the same matrix), and quantifies the size of the resulting equivalence classes. Along the way we highlight algorithmic questions (testing equivalence, recognizing valid matrices) and structural principles (how superdiagonals constrain one another) that shape the combinatorics of subword counts.

2. Parikh matrices and their generalization

2.1. The Parikh matrix mapping

Fix an ordered alphabet $\mathcal{A} = \{a_1, \ldots, a_k\}$. The *Parikh matrix mapping*

$$\mathrm{PM}_k : \; \mathcal{A}^* \longrightarrow \mathcal{M}_{k+1}$$

is the unique monoid morphism into the set of $(k{+}1)\times(k{+}1)$ upper–triangular integer matrices with ones on the diagonal (unitriangular matrices) determined by

$$\mathrm{PM}_k(\lambda) = \mathrm{I}_{k+1}, \qquad \mathrm{PM}_k(a_q) = \mathrm{I}_{k+1} + \mathrm{E}_{q,q+1} \quad (1 \le q \le k),$$

and extended multiplicatively by $\mathrm{PM}_k(uv) = \mathrm{PM}_k(u)\mathrm{PM}_k(v)$ for $u, v \in \mathcal{A}^*$. If $U = \mathrm{PM}_k(w)$, then for all $1 \le i \le j \le k$ the entry

$$U_{i,\,j+1} \;=\; |w|_{a_i a_{i+1}\cdots a_j}\,,$$

so the first superdiagonal ($i = j$) records single–letter counts (i.e., the Parikh vector), while higher superdiagonals encode ordered subword statistics arranged by increasing blocks of the alphabet order. This mapping depends on the chosen order of $\mathcal{A}$; permuting the alphabet corresponds to a simultaneous permutation of indices on rows and columns, and consequently reorders the superdiagonals.

For the binary alphabet $\mathcal{A} = \{a < b\}$ ($k = 2$),

$$\mathrm{PM}_2(a) = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \mathrm{PM}_2(b) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix},$$

and for any word w we have $(\mathrm{PM}_2(w))_{1,3} = |w|_{ab}$, $(\mathrm{PM}_2(w))_{1,2} = |w|_a$, $(\mathrm{PM}_2(w))_{2,3} = |w|_b$. Over a ternary alphabet $\{a < b < c\}$ ($k = 3$),

$$\mathrm{PM}_3(a) = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathrm{PM}_3(b) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathrm{PM}_3(c) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

If $w = \texttt{abca}$, then $|w|_a = 2$, $|w|_b = 1$, $|w|_c = 1$, $|w|_{ab} = 1$, $|w|_{bc} = 1$, and $|w|_{abc} = 1$, hence

$$\mathrm{PM}_3(w) = \begin{pmatrix} 1 & 2 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

The multiplicative rule $\mathrm{PM}_k(uv) = \mathrm{PM}_k(u)\mathrm{PM}_k(v)$ has a simple combinatorial interpretation: multiplying by $\mathrm{PM}_k(a_q) = \mathrm{I} + \mathrm{E}_{q,q+1}$ sends each existing contribution on the q-th column one step up to the $(q{+}1)$-st column, thereby incrementing precisely those subword counts whose rightmost letter is a_q. In particular, letter matrices with non-adjacent indices commute since $\mathrm{E}_{i,i+1}\mathrm{E}_{j,j+1} = 0$ for $|i - j| > 1$, while adjacent letters interact through $\mathrm{E}_{i,i+1}\mathrm{E}_{i+1,i+2} = \mathrm{E}_{i,i+2}$, which is responsible for the appearance of higher superdiagonals.

2.2. Generalized Parikh matrices via subword indicators

Let $v = c_1 \cdots c_t$ be an *indicator word* over $\mathcal{A}$ (letters may repeat). The generalized mapping

$$\mathrm{PM}_v : \; \mathcal{A}^* \longrightarrow \mathcal{M}_{t+1}$$

is the monoid morphism specified on letters by

$$\mathrm{PM}_v(a) \;=\; \mathrm{I}_{t+1} + \sum_{\substack{1 \le i \le t \\ c_i = a}} \mathrm{E}_{i,i+1}, \qquad a \in \mathcal{A},$$

so a letter contributes 1 just above the diagonal exactly at those positions where it appears in v. For $W = \mathrm{PM}_v(w)$ and $1 \le i \le j \le t$,

$$W_{i,\,j+1} \;=\; |w|_{c_i c_{i+1} \cdots c_j} \;,$$

hence $\mathrm{PM}_v(w)$ tabulates subword counts for all contiguous blocks of the indicator v. As an illustration, take $v = \mathtt{baba}$ over $\{a < b\}$ with $t = 4$. Then

$$\mathrm{PM}_v(a) = \mathrm{I}_5 + \mathrm{E}_{2,3} + \mathrm{E}_{4,5}, \qquad \mathrm{PM}_v(b) = \mathrm{I}_5 + \mathrm{E}_{1,2} + \mathrm{E}_{3,4},$$

and $(\mathrm{PM}_v(w))_{1,3} = |w|_{ba}$, $(\mathrm{PM}_v(w))_{2,5} = |w|_{aba}$, etc., aligned with the contiguous blocks of v.

There is an effective reduction from the generalized mapping to the ordinary one. Let $[t] := \{1 < 2 < \cdots < t\}$ be an ordered alphabet and write PM_t for the standard mapping on $[t]$ with $\mathrm{PM}_t(r) = \mathrm{I}_{t+1} + \mathrm{E}_{r,r+1}$. For each $a \in \mathcal{A}$, list the positions where a occurs in v as $P_a(v) = \{p_1 < p_2 < \cdots < p_m\} \subseteq [t]$, and define a coding morphism $h_v : \mathcal{A}^* \to [t]^*$ by

$$h_v(a) \;=\; p_m\, p_{m-1} \cdots p_1 \quad \text{(reverse order; empty word if } P_a(v) = \varnothing).$$

Then for every $w \in \mathcal{A}^*$ one has

$$\mathrm{PM}_v(w) \;=\; \mathrm{PM}_t(h_v(w)).$$

This identity is immediate on letters by construction of PM_v and PM_t, and extends to all words by multiplicativity; the reversal ensures that contributions accumulate on the correct superdiagonals so that blocks $c_i \cdots c_j$ in v are counted in the $(i, j{+}1)$ entry.

3. Ambiguity and decision problems

3.1. M-equivalence and (un)ambiguity

Fix an ordered alphabet and write $\mathrm{PM}(w)$ for the Parikh matrix of a word w with respect to this order. Two words w_1, w_2 are *M-equivalent* if $\mathrm{PM}(w_1) = \mathrm{PM}(w_2)$. A word w is *M-unambiguous* if it is the unique word in its M-equivalence class; otherwise w is *M-ambiguous*. The notion depends on the chosen order of the alphabet, since reordering the alphabet permutes the superdiagonals of $\mathrm{PM}(\cdot)$.

Over the binary alphabet $\{a, b\}$ there is a complete description: a word is M-ambiguous if and only if it contains non-overlapping factors ab and ba. Equivalently, the M-unambiguous words are exactly those in the regular language

$$a^*b^* \;\cup\; b^*a^* \;\cup\; a^*ba^* \;\cup\; b^*ab^* \;\cup\; a^*bab^* \;\cup\; b^*aba^*.$$

For instance, $w = \mathtt{abba}$ is M-ambiguous since it has the disjoint factors $\mathtt{ab}$ and $\mathtt{ba}$; indeed $\mathrm{PM}(\mathtt{abba}) = \mathrm{PM}(\mathtt{baab})$ while $\mathtt{abba} \neq \mathtt{baab}$. In contrast, $\mathtt{aaabbb} \in a^*b^*$ is M-unambiguous. The language of M-unambiguous binary words is recognized by a deterministic automaton with five states.

3.2. Prints

The *print* of a word w is obtained by collapsing each maximal run a^i with $i > 1$ to a single a; we denote it by $\mathrm{print}(w)$. By definition, a printed word equals its own print, e.g., $\mathrm{print}(\mathtt{aaabbaaa}) = \mathtt{aba}$ and $\mathrm{print}(\mathtt{aba}) = \mathtt{aba}$. For any fixed alphabet, only finitely many printed words are M-unambiguous,

and there is an effective bound on their lengths. Moreover, for $k \geq 3$ there exists $N(k)$ such that every sufficiently long factor of the periodic word $(a_1 a_2 \cdots a_k)^\omega$ is M-ambiguous. The conjecture

$$\text{“}M\text{-unambiguous} \Rightarrow \text{print}(w) \text{ is } M\text{-unambiguous”}$$

holds for binary and ternary alphabets but fails for larger ones; nevertheless it supports the heuristic that, as the alphabet grows, sufficiently long prints inevitably force ambiguity.

3.3. Decision tasks

Core algorithmic questions include: recognizing which upper–triangular unitriangular matrices arise as Parikh matrices; deciding whether a given w is M-ambiguous; and testing M-equivalence of two words. The M-equivalence test reduces to computing and comparing two Parikh matrices and can be implemented in time $O(|w| \cdot |\mathcal{A}| + |\mathcal{A}|^2)$ and space $O(|\mathcal{A}|^2)$ by scanning w once and updating the relevant superdiagonals via the multiplicative rule $\mathrm{PM}(uv) = \mathrm{PM}(u)\mathrm{PM}(v)$. By contrast, recognition of Parikh matrices and M-unambiguity testing admit natural brute–force procedures (enumerating preimages or M-equivalent neighbors) that are exponential in the worst case.

Entries above the main diagonal exhibit substantial independence: fixing any selection of superdiagonals need not determine the remaining ones. For $\{a < b < c\}$, there exist distinct words with the same single–letter counts and the same second–superdiagonal counts $|w|_{ab}$ and $|w|_{bc}$ but different third–superdiagonal $|w|_{abc}$; for example,

$$\texttt{baacb} \text{ and } \texttt{abcba}$$

both satisfy $|\cdot|_a = 2$, $|\cdot|_b = 2$, $|\cdot|_c = 1$, $|\cdot|_{ab} = 2$, $|\cdot|_{bc} = 1$, yet $|\texttt{baacb}|_{abc} = 0$ while $|\texttt{abcba}|_{abc} = 1$. This illustrates why partial information across superdiagonals generally does not pin down the entire Parikh matrix.

4. Subword indicators and alphabet permutations

Switching the indicator word may or may not preserve $\mathrm{PM}_v(w)$, depending on w. Over $\{a, b\}$, the equality $\mathrm{PM}_{ab}(w) = \mathrm{PM}_{ba}(w)$ forces

$$|w|_a = |w|_b \quad \text{and} \quad |w|_{ab} = |w|_{ba}\,.$$

Since $|w|_{ab} + |w|_{ba} = |w|_a\,|w|_b$, writing $|w|_a = |w|_b = m$ yields $2\,|w|_{ab} = m^2$, so m must be even. Such coincidences occur only for M-ambiguous w.

A permutation p of the alphabet is called *Parikh–friendly* if there exists a witness word w using all letters such that

$$\mathrm{PM}_{a_1 \cdots a_k}(w) = \mathrm{PM}_{p(a_1 \cdots a_k)}(w).$$

Two infinite families are known:

- **Cyclic shifts.** The k–cycle $p = (a_1 a_2 \cdots a_k)$ is Parikh–friendly. A witness is

  $$w = a_1 a_2 \cdots a_k\; a_k \cdots a_2 a_1,$$

  for which both indicator orders yield all superdiagonal entries equal to 2.

- **Disjoint transpositions.** Any product of disjoint swaps is Parikh–friendly: for each transposed pair $(x\,y)$ insert a short balancing block (e.g., $xyyx$) so that the contributions to all relevant subword counts match under the two orders, and concatenate these blocks across pairs. Fixed points may be included once or twice without effect.

A complete characterization of Parikh–friendly permutations remains open.

5. Quantifying M-ambiguity

Define the *degree of ambiguity* $d(M)$ of a Parikh matrix M as the number of words w with $\mathrm{PM}(w) = M$. For a word u, its degree is $d(\mathrm{PM}(u))$. Degrees are always finite: the first superdiagonal of $\mathrm{PM}(w)$ fixes all single–letter counts and hence $|w|$. Nonetheless, degrees can grow *exponentially* in the word length.

Binary example. Over $\{a, b\}$, consider

$$w(n) = a^n\, b^n\, b^n\, a^n = a^n\, b^{2n}\, a^n.$$

A direct computation gives

$$\mathrm{PM}(w(n)) \;=\; \begin{pmatrix} 1 & 2n & 2n^2 \\ 0 & 1 & 2n \\ 0 & 0 & 1 \end{pmatrix},$$

since $|w(n)|_a = 2n$, $|w(n)|_b = 2n$, and the ab–count is $|w(n)|_{ab} = n \cdot (2n) = 2n^2$ (every initial a sees all $2n$ b's to its right).

Local moves preserving the matrix. For binary words, $\mathrm{PM}(w)$ is determined by $(\,|w|_a\,, |w|_b\,, |w|_{ab}\,)$. An adjacent swap $ab \leftrightarrow ba$ changes $|w|_{ab}$ by ∓ 1. Hence any *synchronized* collection of such swaps whose net effect on $|w|_{ab}$ is zero preserves $\mathrm{PM}(\cdot)$. Starting from $w(n)$, many distinct words with the same triple $(2n, 2n, 2n^2)$ can be reached.

Counting via inversion profiles. Index the $2n$ occurrences of a from left to right and let r_i be the number of b's to the right of the i-th a. Then $r_1 \geq r_2 \geq \cdots \geq r_{2n}$, each $r_i \in \{0, 1, \ldots, 2n\}$, and

$$\sum_{i=1}^{2n} r_i \;=\; |w|_{ab}\,.$$

Thus binary words with $|w|_a = |w|_b = 2n$ and $|w|_{ab} = 2n^2$ are in bijection with partitions of $2n^2$ whose Ferrers diagram fits inside a $2n \times 2n$ rectangle. Equivalently,

$$d(\mathrm{PM}(w(n))) \;=\; [q^{\,2n^2}] \begin{bmatrix} 4n \\ 2n \end{bmatrix}_q,$$

the coefficient of q^{2n^2} in the Gaussian binomial $\begin{bmatrix} 4n \\ 2n \end{bmatrix}_q$.

Exponential growth. By standard partition asymptotics, the number of partitions of $2n^2$ with at most $2n$ parts and largest part at most $2n$ grows as $\exp(\Theta(n))$. Consequently, there exists a constant $c > 0$ and n_0 such that for all $n \geq n_0$,

$$d(\mathrm{PM}(w(n))) \;\geq\; e^{cn} \;=\; 2^{\Omega(n)} \;=\; 2^{\Omega(|w(n)|)}.$$

In particular, there is no polynomial (in the word length) upper bound on the ambiguity degree even for the fixed binary alphabet.

6. Conclusion and outlook

We revisited Parikh matrices as a bridge between order-free Parikh vectors and fully order-sensitive subword statistics. After fixing notation, we formalized the Parikh matrix mapping PM_k and its indicator-based generalization PM_v, showing that the latter reduces effectively to the former via a simple encoding that reverses letter-occurrence positions. On the structural side, we reviewed M-equivalence and M-unambiguity, including the complete binary characterization, finiteness phenomena for prints, and the dependence of M-equivalence on alphabet order. Algorithmically, we remarked that testing M-equivalence is polynomial (via matrix comparison), whereas recognition and unambiguity testing admit natural but exponential search procedures and exhibit substantial independence across superdi

References

[1] Parikh, R. J. (1966). On context-free languages. *Journal of the ACM (JACM), 13*(4), 570-581.

[2] Mateescu, A. (2004). Algebraic aspects of Parikh matrices. In *Theory Is Forever: Essays Dedicated to Arto Salomaa on the Occasion of His 70th Birthday* (pp. 170-180). Berlin, Heidelberg: Springer Berlin Heidelberg.

[3] Salomaa, A. (2005). Connections between subwords and certain matrix mappings. *Theoretical Computer Science, 340*(2), 188-203.

[4] Serbanuta, V. N. (2009). On Parikh Matrices, Ambiguity, and PRINTS. *International Journal of Foundations of Computer Science*, 20(1), 151-165.

[5] Atanasiu, A., Atanasiu, R., & Petre, I. (2008). Parikh matrices and amiable words. *Theoretical Computer Science, 390*(1), 102-109.

[6] Claesson, A. (2015). Subword counting and the incidence algebra. arXiv preprint arXiv:1502.03065.

[7] Subramanian, K. G., Huey, A. M., & Nagar, A. K. (2009). On Parikh matrices. *International Journal of Foundations of Computer Science, 20*(02), 211-219.

[8] Atanasiu, A., & Atanasiu, R. F. (2013). Enriching Parikh matrix mappings. *International Journal of Computer Mathematics, 90*(3), 511-521.

[9] Atanasiu, R. F. (2010). *Languages attached to Parikh matrices*. Technical Report, University" Alexandru Ioan Cuza" of Iasi Faculty of Computer Science.

[10] Bhattacharjee, A., & Purkayastha, B. S. (2014). Some alternative ways to find M-ambiguous binary words corresponding to a Parikh matrix. *International Journal of Computer Applications, 4*(1), 53-64.

[11] Chern, Z. J., & Teh, W. C. (2019). Extension of Parikh matrices to terms and its injectivity problem. *Malaysian Journal of Mathematical Sciences, 13*, 147-156.

[12] Lavado, G. J. (2015). *Descriptional Complexity and Parikh Equivalence.*

How to cite this article: Arto Salomaa (2020). From Parikh Vectors to Parikh Matrices: Structure, Ambiguity, and Complexity. *Bulletin of Computer and Data Sciences*, 1(1), 19-26. DOI: 10.71448/bcds2011-3

Received: 02/6/2020 **Revised:** 28/8/2020 **Accepted:** 30/10/2020 **Publish:** 30/12/2020