Width-Indexed Positional Identifiers (WIPI)
A Graded Coordinate Algebra for Fixed-Width Strings

Hosameldeen Saleh
Root Zero Vault, Inc

Abstract

We define Width-Indexed Positional Identifiers (WIPI), a typed coordinate system for fixed-width positional identifiers in base $b \geq 2$. A WIPI element is a pair (v,w) where $w \geq 0$ is a declared width and v is an integer constrained by $0 \leq v < b^w$. Each width w induces a finite coordinate block I_w, and the full identifier space is the disjoint union $\bigsqcup_{w \geq 0} I_w$. We give explicit encoding/decoding maps between I_w and strings Σ^w over a chosen alphabet Σ with $|\Sigma|=b$, define constant-time navigation operators parent and extend_d that are compatible with a rooted b-ary tree structure, and present a global rank/unrank bijection that aligns with the shortlex (length-lexicographic) order on Σ^*. The contribution is a concise coordinate toolkit—typed width, canonical maps, and operator calculus—useful for any context requiring fixed-width identifiers with well-defined algebraic and ordering properties.

Keywords

fixed-width identifiers; graded sets; positional notation; tries/prefix trees; shortlex order; ranking/unranking

1. Introduction

Fixed-width identifiers are ubiquitous (e.g., database keys, filenames, versioned records). In many settings, width is not merely formatting: it determines the representation domain and enables length-aware ordering and indexing. This paper formalizes a width-indexed coordinate space in which the width component is explicit and semantically binding at the type level.

WIPI is intentionally conservative: it does not propose new arithmetic and does not require any governance, security, or policy interpretation. Instead, it packages standard primitives—positional valuation, length grading, and prefix-tree navigation—into a small, explicit algebra that is easy to implement and reason about.

1.2. Related work and positioning

The underlying objects are standard. The set Σ^* of all finite strings over an alphabet Σ forms the free monoid under concatenation; grading by length is the canonical decomposition $\Sigma^* = \bigsqcup_{w\geq 0} \Sigma^w$ [6], [10]. The prefix relation induces the familiar rooted prefix tree (trie) structure; tries were introduced by de la Briandais (1959) and popularized/termed by Fredkin (1960) [1], [2], with compressed variants such as PATRICIA (Morrison, 1968) [3].

Shortlex (length-lexicographic) order is a standard well-ordering on Σ^* when Σ is finite, used in formal language theory and related areas [6], [7], and it supports an order-preserving correspondence between Σ^* and the natural numbers via ranking/unranking (bijective numeration viewpoints) [7], [8].

What is new here is not the existence of these structures, but the explicit typed packaging for fixed-width identifiers: (i) a disjoint union of per-width integer blocks I_w with declared width, (ii) explicit encoding/decoding maps to Σ^w, (iii) a small operator calculus (parent, extend_d, promote) that is closed on the space, and (iv) a global rank/unrank mapping aligned with shortlex. This coordinate toolkit is intended to be cited and reused as a compact, formal substrate.

2. Preliminaries

2.1. Digits, alphabets, and valuation

Fix an integer base $b\geq 2$. Let Σ be a finite alphabet with $|\Sigma|=b$. Let $\mu: \Sigma \to \{0,1,\dots,b-1\}$ be a bijection interpreting symbols as digits. For $w\geq 0$, let Σ^w denote strings of length w over Σ, and let ε denote the empty string ($w=0$). For $s = s_1\dots s_w \in \Sigma^w$ define its positional valuation

$$\text{val}_b(s) = \sum_{i=1}^{w} \mu(s_i) \cdot b^{w-i}.$$

Remark (base vs. alphabet). All arithmetic in this paper is on integers (v,w) with digits $d\in\{0,\dots,b-1\}$. Alphabet symbols are merely encodings; for $a\in\Sigma$ we may write $\text{extend}_a := \text{extend}_{\mu(a)}$ to avoid conflating glyphs with digits.

2.2. Prefix order and rooted b-ary tree

Define the prefix relation $\preceq$ on Σ^* by $s \preceq t$ iff $t = su$ for some $u \in \Sigma^*$. The directed graph with vertices Σ^* and edges $s \to sa$ for $a \in \Sigma$ is the rooted b-ary trie with root ε [1], [2].

2.3. Shortlex order

Shortlex order (also called length-lexicographic order) compares strings primarily by length and breaks ties lexicographically among equal-length strings, assuming Σ is totally ordered [6], [7]. Over a finite Σ, shortlex is a well-ordering of Σ^* and is compatible with ranking/unranking constructions [7], [8].

3. The WIPI coordinate space

For each width $w \geq 0$ define the width-w block $I_w = \{0,1,\ldots,b^w - 1\} \times \{w\}$. Elements are written (v,w) with $0 \leq v < b^w$. The full WIPI space is the disjoint union

$$W = \bigsqcup_{w \geq 0} I_w.$$

The disjointness is essential: (v,w) and (v,w') are distinct when $w \neq w'$, even if their integer components match.

3.1. Encoding and decoding

For each $w \geq 0$, define $enc_w: I_w \to \Sigma^w$ by taking the unique base-b expansion of v into w digits (allowing leading zeros) and mapping digits to Σ via μ^{-1}. Define $dec_w: \Sigma^w \to I_w$ by $dec_w(s) = (val_b(s), w)$.

Proposition 3.1 (Per-width bijection). For each $w \geq 0$, enc_w and dec_w are mutual inverses.

Proof sketch. Uniqueness of base-b expansion with fixed width w gives a unique digit vector; μ is bijective; val_b computes the same v. Conversely, encoding the digits of $val_b(s)$ reproduces s.

4. Operator calculus

4.1. Extension and parent

For $w \geq 0$ and digit $d \in \{0,\ldots,b-1\}$, define extension

$$extend_d(v,w) = (b \cdot v + d, w+1).$$

For $w \geq 1$ define the parent operator

$$parent(v,w) = (\lfloor v/b \rfloor, w-1).$$

Proposition 4.1 (Local inverses). For all $(v,w) \in W$ and $d \in \{0,\ldots,b-1\}$, $parent(extend_d(v,w)) = (v,w)$. Moreover, for $w \geq 1$, writing $v = b \cdot q + r$ with $0 \leq r < b$, we have $extend_r(parent(v,w)) = (v,w)$.

Proof. Immediate from integer division with remainder.

4.2. Promotion

For $k \geq 0$ define promotion (zero-padding) by

$$promote_k(v,w) = (v \cdot b^k, w+k).$$

Remark 4.2 (Promotion is not ancestry). Promotion is a grade-shift that preserves the represented string up to appending k zeros; it is not, in general, the same as repeated extension by arbitrary digits. Only the special case $promote\_k = extend\_0 \circ \dots \circ extend\_0$ (k times) coincides with a k-step path in the trie.

5. Tree structure and isomorphism

Define the directed graph G_W with vertex set W and an edge $(v,w) \to extend\_d(v,w)$ for each digit d. This is the canonical b-ary expansion graph induced by extension.

Define $\sigma: W \to \Sigma^*$ by $\sigma(v,w) = enc\_w(v,w)$ (viewed as a string in Σ^w).

Theorem 5.1 (Rooted tree isomorphism). The map σ is a graph isomorphism between (W, edges via extend_d) and (Σ^*, edges $s \to sa$). The unique root vertex is $(0,0) \in I\_0$, and $\sigma(0,0)=\varepsilon$.

Proof sketch. By Proposition 3.1, σ is bijective. Compatibility of extension with appending a digit follows from $val\_b(sa)=b\cdot val\_b(s)+\mu(a)$, so encoding after extension matches string extension. The width-0 block contains only (0,0), corresponding to ε, hence it is the unique root.

6. Shortlex-compatible ranking

Let $O(w) = \Sigma\_\{i=0\}^\{w-1\} b^i$, with $O(0)=0$ (and for $b>1$, $O(w)=(b^w-1)/(b-1)$). Define

$rank(v,w) = O(w) + v$.

Proposition 6.1 (Global bijection). rank: $W \to \mathbb{N}$ is a bijection with inverse $unrank(n) = (n-O(w), w)$, where w is the unique integer satisfying $O(w) \le n < O(w+1)$.

Proposition 6.2 (Shortlex alignment). If Σ is totally ordered and μ is order-preserving, then ranking by (w,v) with primary key w and secondary key v corresponds to the shortlex order on Σ^* under σ.

7. Representation independence

Changing the alphabet (glyphs) while preserving the base and digit order does not change the underlying coordinate space. If Σ and Σ' are alphabets of size b with bijections μ and μ', then the relabeling map ρ: $\Sigma^* \to \Sigma'^*$ defined by $\rho(s1\dots sw)=\mu'^\{-1\}(\mu(s1))\dots\mu'^\{-1\}(\mu(sw))$ commutes with encoding/decoding and preserves the induced structure.

8. Complexity notes

Navigation operators parent and extend_d are O(1) arithmetic operations on machine integers (ignoring bignum costs). Encoding/decoding for width w is O(w) in the number of digits/symbols. rank is O(1); unrank requires finding w, which is $O(\log_b n)$ via closed form or doubling/binary search.

9. Conclusion

WIPI formalizes fixed-width positional identifiers as a graded coordinate space of typed pairs (v,w) with explicit per-width domains, canonical maps to fixed-length strings, a small operator calculus compatible with a rooted b-ary tree, and a shortlex-aligned global ranking. The framework is deliberately minimal and mathematically conservative; its value is as a reusable coordinate toolkit that can be referenced independently of any higher-level application.

Appendix A. Alignment with Root Zero Conversion Rules (Informative)

A.1. Zero-free digit instantiation (cp+1, base+1)

Root Zero's latest conversion rules define a single-integer coordinate for a variable-length identifier string by shifting each digit by +1 and using base (B+1), where B is the size of the underlying symbol domain. This appendix records the exact specialization under which WIPI aligns with that rule while remaining base-agnostic.

Let U be the domain of admissible symbols (e.g., Unicode scalar values (scalar values exclude surrogate code points) [12] after NFC normalization) [11]. Fix $B := |U|$ and set the numeric base $b := B+1$. Define an augmented alphabet $\Sigma := U \cup \{\perp\}$ of size $|\Sigma|=b$, where $\perp$ is a distinguished reserved symbol that never appears in admissible identifiers. Define the digit map $\mu: \Sigma \to \{0,1,\ldots,b-1\}$ by $\mu(\perp)=0$ and $\mu(u)=\iota(u)+1$ for $u \in U$, where $\iota: U \to \{0,\ldots,B-1\}$ is any fixed, published bijection. For Unicode scalar values (scalar values exclude surrogate code points) [12], one convenient choice is $\iota(cp)=cp$ for $cp<0xD800$ and $\iota(cp)=cp-0x800$ for $cp \geq 0xE000$, which skips the surrogate range U+D800..U+DFFF.

Then any admissible identifier string $s=u1\ldots un \in U^n$ is encoded as the WIPI element (v,n) with $v = val_b(s) = \Sigma_{i=1..n} \mu(ui)\cdot b^{n-i}$. Since $\mu(ui) \in \{1,\ldots,b-1\}$ for all $ui \in U$, the base-b expansion of v has no leading zero digits; consequently, the width n is recoverable from v alone as the number of digits in its base-b expansion. Decoding is performed by repeated div/mod by b to recover digits, subtracting 1, and applying $rank^{-1}$ to recover symbols.

A.2. Local subset coordinates

For a tenant-declared alphabet Σ_T of size k with mapping $\mu_T:\Sigma_T\to\{0,\dots,k-1\}$, Root Zero's local coordinate rule uses digits (μ_T+1) and base $(k+1)$. This is captured identically by the construction above with $U:=\Sigma_T$, $B:=k$, $b:=k+1$ and a reserved $\perp$ symbol mapped to digit 0.

Remark (relationship to fixed-width WIPI blocks).

WIPI's core model remains fixed-width: per-width blocks $I_w = \{0,\dots,b^w-1\}\times\{w\}$ and the per-width bijection $I_w \leftrightarrow \Sigma^w$. The Root Zero conversion is an informative specialization that restricts admissible strings to U^w (excluding $\perp$) so that width may be inferred from the integer coordinate when desired. When width is carried explicitly (as in WIPI), no such restriction is required.

References

[1] R. de la Briandais, “File searching using variable length keys,” in Papers presented at the March 3–5, 1959, Western Joint Computer Conference (IRE-AIEE-ACM ’59), AFIPS Conf. Proc., vol. 15, pp. 295–298, 1959.

[2] E. Fredkin, “Trie memory,” Communications of the ACM, vol. 3, no. 9, pp. 490–499, Sept. 1960, doi: 10.1145/367390.367400.

[3] D. R. Morrison, “PATRICIA—Practical Algorithm To Retrieve Information Coded in Alphanumeric,” Journal of the ACM, vol. 15, no. 4, pp. 514–534, Oct. 1968, doi: 10.1145/321479.321481.

[4] P. E. Black, “Trie,” Dictionary of Algorithms and Data Structures, National Institute of Standards and Technology (NIST).

[5] P. E. Black, “Patricia tree,” Dictionary of Algorithms and Data Structures, National Institute of Standards and Technology (NIST).

[6] M. Sipser, Introduction to the Theory of Computation, 3rd ed. Boston, MA, USA: Cengage Learning, 2013.

[7] R. M. Smullyan, Theory of Formal Systems, Annals of Mathematics Studies, no. 47. Princeton, NJ, USA: Princeton University Press, 1961.

[8] D. B. A. Epstein, J. W. Cannon, D. F. Holt, S. V. F. Levy, M. Paterson, and W. P. Thurston, Word Processing in Groups. Boston, MA, USA: Jones and Bartlett Publishers, 1992.

[9] S. Mac Lane, Categories for the Working Mathematician, 2nd ed. New York, NY, USA: Springer, 1998.

[10] M. Lothaire, Combinatorics on Words. Reading, MA, USA: Addison-Wesley, 1983.

[11] The Unicode Consortium, “Unicode Normalization Forms (UAX #15),” Unicode Standard Annex #15, latest version. Available: https://www.unicode.org/reports/tr15/.

[12] The Unicode Consortium, “Glossary,” The Unicode Standard. See entries related to Unicode scalar values and Unicode encoding forms. Available: https://unicode.org/glossary/.