DECOMPOSABLE DISJOINT-UNION CIRCUITS FOR PRIMITIVE WORDS

WILL HOWE, GRANT HUFFMAN, JENSEN KOHLMEYER, AND LIAM KRUER

Abstract. For every fixed finite alphabet A with at least two letters, we construct smooth decomposable circuits representing the primitive words of each length n. The gates are literals, Cartesian products on disjoint position sets, and disjoint unions on equal position sets. The total number of gates and wires is $O_A(n^2 5^{\omega(n)}) = n^{2+o(1)}$, and the construction takes $n^{2+o(1)}$ bit operations. Prime periods are expressed as coordinate directions of independent rectangular grids. A recursion at the first nonconstant vertex computes the exact set of failed directions; sharing its continuations gives the size bound. This resolves the primitive-word instance of the circuit question posed by Amarilli, Monet, Raphaël, and Salvati. It does not determine whether primitive words form a context-free or a poly-slicewise-unambiguous language. The same circuit bound holds for any union of exact power orders at a fixed length.

hexagon:2611.00235v1

1. Statement and circuit model

A nonempty word is *primitive* if it is not a power u^d with $d \geq 2$. The empty word is excluded. Write $\mathrm{Prim}_A(n) \subseteq A^n$ for the primitive words of length n, and let $\omega(n)$ be the number of distinct prime divisors of the positive integer n, with $\omega(1) = 0$.

Amarilli, Monet, Raphaël, and Salvati introduced a multivalued circuit framework for probabilistic language membership [1, Section 5]. Their construction for primitive words uses complementation, and [1, Section 6] asks whether complementation can be avoided for this language. We use their circuit model, with the conventions made explicit below. Their quadratic arithmetic algorithm for probabilistic membership is already available; the result here concerns the form and size of a representation.

Fix a finite set U of positions. An assignment on $D \subseteq U$ is a function $D \to A$. A circuit is a finite directed acyclic graph, with a designated output, built from the following gates.

(i) A literal $i : a$ has domain $\{i\}$ and represents the one assignment sending i to a.
(ii) A product gate has inputs with pairwise disjoint domains. Its domain is their union, and its assignments are all unions of one assignment from each input.
(iii) A union gate has inputs with the same domain and pairwise disjoint represented sets. It represents their union, with that common domain.

These requirements are called *decomposability*, *smoothness*, and *disjointness*, respectively. We allow the nullary product, of domain $\varnothing$, representing the unique empty assignment, and the nullary union, of domain $\varnothing$, representing no assignment. The domain of every other gate is the union of its input domains. Thus the nullary union alone cannot represent the empty set on a prescribed nonempty domain. We denote the set represented by a gate c by $[\![c]\!]$. The size of a circuit is its number of gates plus its number of wires. Fan-out is unrestricted, so a constructed gate may be shared by several later gates. Positions may be relabeled by any bijection without changing these properties or the size.

Theorem 1.1. *Fix a finite alphabet A, with $|A| \geq 2$. For each integer $n \geq 1$ there is a smooth decomposable disjoint-union circuit with domain $\{0, \ldots, n-1\}$ representing exactly $\mathrm{Prim}_A(n)$ and*

Date: November 7, 2026.
Key words and phrases. Primitive words, decomposable circuits, disjoint unions, knowledge compilation.
Prepared with AI assistance from OpenAI Codex in mathematical exploration, checking, and manuscript preparation.

having size

$$O_A\Big(n^2 5^{\omega(n)}\Big).$$

One algorithm constructs these circuits in $n^{2+o(1)}$ bit operations as a function of n. The empty slice is represented by a nullary union.

Here and below A is fixed, so constants may depend on $|A|$. Construction time is measured in the represented length n, or equivalently with n given in unary. A stronger intermediate bound will be

$$O_A\left(n \prod_{p\mid n,\ p \text{ prime}} (1+4p)\right). \tag{1.1}$$

Products in this model may combine interlaced position sets. The theorem does not provide a context-free grammar or a family of polynomial-size unambiguous context-free grammars for the slices. These language-theoretic questions are separate from the circuit question; see also the discussion of primitive words in [2, Section 6].

2. From periods to independent grids

We identify positions of a length-n word with $\mathbb{Z}/n\mathbb{Z}$. For a divisor $d \mid n$, a word is invariant under shift d if $w(j+d) = w(j)$ for all positions j.

Lemma 2.1. *Let $n \geq 2$, and let P be its set of distinct prime divisors. A word $w \in A^n$ is primitive if and only if it fails to be invariant under shift n/p for every $p \in P$.*

Proof. If $w = u^d$ with $d \geq 2$, then $d \mid n$. Choose a prime $p \mid d$ and write $w = (u^{d/p})^p$. It is invariant under shift n/p. Conversely, the orbits of shift n/p are the sets $\{i, i+n/p, \ldots, i+(p-1)n/p\}$ for $0 \leq i < n/p$. Invariance therefore says precisely that w is the pth power of its prefix of length n/p. □

Set

$$R = \prod_{p\in P} p, \qquad g = n/R, \qquad k = |P|.$$

Every shift n/p preserves the residue of a position modulo g. Within residue class r, $0 \leq r < g$, write a position uniquely as $r + gt$, where $t \in \mathbb{Z}/R\mathbb{Z}$. Use the Chinese remainder bijection

$$\mathbb{Z}/R\mathbb{Z} \longrightarrow \prod_{p\in P} \mathbb{Z}/p\mathbb{Z}, \qquad t \longmapsto (t \bmod p)_{p\in P}. \tag{2.1}$$

Shift n/p adds R/p to t. In these coordinates it fixes every coordinate other than p, and adds the nonzero residue $R/p \bmod p$ in coordinate p. Repeated addition visits all p values of that coordinate. Thus the g residue classes form g independent copies of a rectangular grid with side lengths $p \in P$.

For a labeled grid, call an axis *failed* if at least one line parallel to it is not constant. The set of failed axes is its *mask*. The preceding observations prove the following statement.

Lemma 2.2. *The word fails shift n/p exactly when axis p fails in at least one of the g grids. Its primitive-word condition is therefore that the union of the masks of the g grids is P.*

We next construct circuits for each exact grid mask. Exactness will make all unions in the construction disjoint.

3. The constant-prefix recursion

For this section take any ordered finite set I of axes, with integer side lengths $q_i \geq 2$. Write

$$X = \prod_{i \in I} \{0, \ldots, q_i - 1\}, \qquad V = |X| = \prod_{i \in I} q_i.$$

A *subgrid* fixes the coordinates outside some $T \subseteq I$ and varies all the coordinates in T. Its varying axes are T, its size is $v_T = \prod_{i \in T} q_i$, and its vertices inherit lexicographic order from the ordered axes. In particular, a subgrid with no varying axes has a single vertex.

Let $G = (x_0, \ldots, x_{v-1})$ be an ordered subgrid with varying axes T. For a labeling $f : G \to A$ define

$$\operatorname*{Fail}_G(f) = \left\{ i \in T : \begin{array}{l} \text{there exist } x, y \in G \text{ equal in every coordinate except } i, \\ \text{with } f(x) \neq f(y) \end{array} \right\}.$$

For $1 \leq t \leq v$, $b \in A$, and $M \subseteq T$, let $\mathcal{S}(G, t, b, M)$ be the assignments on $G_{\geq t} = \{x_t, \ldots, x_{v-1}\}$ whose extension by the value b on $x_0, \ldots, x_{t-1}$ has mask exactly M. Let $\mathcal{B}(G, M)$ be the assignments on all of G with mask exactly M. The constant prefix is a condition on the represented assignments, and its positions are outside their domain.

We will represent every nonempty one of these sets by a gate. If a state is empty, we record that fact and create no gate for it. A proposed product involving an empty state is omitted, and unions range only over nonempty inputs. This convention respects the structural definition of domain above. Feasibility is determined during the same recursion, without examining all assignments.

Lemma 3.1 (first differing vertex)**.** *Let $1 \leq t \leq v$, and complete a prefix of t copies of b to a nonconstant labeling of G. Let x_s be its first vertex whose label a differs from b. Put*

$$F = F(x_s) = \{i \in T : (x_s)_i \neq 0\}, \qquad U = T \setminus F.$$

Then $s \geq t$, F is nonempty, and every axis in F fails. Partition G into subgrids H by fixing all coordinates in F, so that each H has varying axes U. The following hold.

(i) In the subgrid H_* containing x_s, that vertex is first, and no earlier vertex of G lies in H_*.
(ii) In every other H, the vertices preceding x_s in G form an initial segment of H, all labeled b.
(iii) For $i \in U$, axis i fails in G if and only if it fails in at least one of these subgrids.

Consequently, for every such completed labeling f,

$$\operatorname*{Fail}_G(f) = F \cup \bigcup_H \operatorname*{Fail}_H(f|_H). \tag{3.1}$$

Proof. The first vertex x_0 has all varying coordinates zero. Since $s \geq t \geq 1$, we have $x_s \neq x_0$ and hence $F \neq \varnothing$. For each $i \in F$, reset coordinate i of x_s to zero, keeping all others fixed. The resulting vertex precedes x_s in lexicographic order, so its label is b, whereas the label of x_s is a. The two vertices share an i-axis line, and that line is nonconstant.

Every coordinate of x_s in U is zero. After the F coordinates are fixed to those of x_s, it is therefore the first vertex of H_*. Lexicographic order on any other H is the restriction of the order on G. Intersecting an initial segment of G with H gives an initial segment of H, proving (ii). Finally, any line in an axis $i \in U$ has fixed F coordinates. It is contained in one H, and the i-lines of all the H are exactly the i-lines of G. This proves (iii) and (3.1). □

We spell out the use of exact masks in combining subgrids.

Lemma 3.2 (combining exact masks)**.** *Suppose $D_1, \ldots, D_h$ are pairwise disjoint position sets. For each j suppose gates on domain D_j represent pairwise disjoint families $\mathcal{L}_{j,N}$ indexed by $N \subseteq U$, with empty families recorded as absent. There is a smooth decomposable disjoint-union construction representing, for every $M \subseteq U$,*

$$\bigcup_{N_1 \cup \cdots \cup N_h = M} \mathcal{L}_{1,N_1} \times \cdots \times \mathcal{L}_{h,N_h} \tag{3.2}$$

using $O(h4^{|U|})$ new gates and wires. All nonempty resulting states have domain $D_1 \cup \cdots \cup D_h$.

Proof. Start with the nullary product at mask $\varnothing$ and no state at any other mask. After processing j domains, keep one state $E_{j,M}$ for each nonempty family with accumulated mask M. For every nonempty pair $E_{j-1,B}$ and $\mathcal{L}_{j,N}$, form their product and attach it to the union for $E_{j,B\cup N}$. These products are decomposable. For a fixed resulting mask, all products have domain $D_1 \cup \cdots \cup D_j$. An assignment on this domain determines its restrictions to the first $j-1$ domains and to D_j. Pairwise disjointness of the state families then determines both B and N uniquely. Thus the union is disjoint. Induction proves (3.2) and pairwise disjointness across its different masks. At each step there are at most $4^{|U|}$ mask pairs, so the claimed bound counts both gates and wires. Empty states are recognized by the absence of a nonempty pair. □

Proposition 3.3. *For every subgrid G and every admissible t, b, M, the nonempty sets $\mathcal{S}(G, t, b, M)$ and $\mathcal{B}(G, M)$ admit circuits of the specified type. All these circuits can be constructed simultaneously by induction on the number of varying axes.*

Proof. Suppose the construction is available on subgrids with fewer than $|T|$ varying axes. For each b the constant completion contributes the single all-b assignment on $G_{\geq t}$ to $\mathcal{S}(G, t, b, \varnothing)$. It is represented by a product of literals; when the domain is empty this is the nullary product.

For a nonconstant completion, fix its first differing index $s \geq t$ and its letter $a \neq b$. Use F, U, and the subgrids H of Lemma 3.1. They have fewer varying axes because F is nonempty. For $H \neq H_*$ let

$$r_H = |H \cap \{x_0, \ldots, x_{s-1}\}|.$$

If $r_H = 0$, use the states $\mathcal{B}(H, N)$ on all of H. If $r_H \geq 1$, use $\mathcal{S}(H, r_H, b, N)$ on the remaining suffix of H. For H_* use $\mathcal{S}(H_*, 1, a, N)$. In every case the prescribed prefix lies outside the state's domain. These remaining domains partition $G_{>s} = \{x_{s+1}, \ldots, x_{v-1}\}$.

Apply Lemma 3.2 to these states. Denote the gate of accumulated mask $N \subseteq U$, if nonempty, by $K_{G,s,b,a,N}$. It has domain $G_{>s}$. To obtain a branch for the original suffix, multiply it by the assignment b on $\{x_t, \ldots, x_{s-1}\}$ and by the literal $x_s : a$. The three domains are disjoint, and their union is $G_{\geq t}$. By (3.1), this branch contributes precisely to

$$\mathcal{S}(G, t, b, F \cup N). \tag{3.3}$$

For each required mask take the union of these branches and the constant branch, if appropriate. Their domains agree. Two nonconstant branches are disjoint because their completed labelings have different first differing positions or different letters at that position. For fixed s, a, masks N are exact and pairwise disjoint; moreover $F \cup N = M$ determines $N = M \setminus F$. The constant branch is disjoint from every nonconstant branch. Lemma 3.1 shows that every completion occurs in the prescribed branch, so these gates represent exactly the claimed sets.

Finally, form $\mathcal{B}(G, M)$ by the disjoint union, over $b \in A$, of the literal $x_0 : b$ times the state $\mathcal{S}(G, 1, b, M)$, omitting absent states. The two factors have disjoint domains. The resulting branches all have domain G, and their first letters make them pairwise disjoint.

When $T = \varnothing$, G is a singleton. The only possible mask is $\varnothing$; its states at $t = 1$ are nullary products, and $\mathcal{B}(G, \varnothing)$ is the disjoint union of the literals at its one position. No nonconstant branch occurs. This supplies the base case, including all empty-domain states used above. □

4. Sharing and the size bound

The induction alone gives a circuit construction. The following count depends on sharing a continuation across all compatible prefix lengths.

Lemma 4.1. *Let G have v vertices and m varying axes. Besides gates already constructed on its proper subgrids, all its states in Proposition 3.3 require $O_A(v^2 4^m)$ gates and wires.*

Proof. We first build the constant assignment of letter b on every interval $\{x_i, \dots, x_{j-1}\}$, $0 \le i \le j \le v$. For fixed i, b, start with the nullary product and append one literal at a time. Each extension uses a binary decomposable product. This costs $O_A(v^2)$ gates and wires in total. Literals can be shared wherever they recur.

For fixed $s \ge 1$, b, and $a \ne b$, the partition into H, all numbers r_H, and the distinguished state with letter a depend only on G, s, b, a. They do not depend on $t \le s$. The core gates $K_{G,s,b,a,N}$ are therefore built once and shared for every such t. There are at most v subgrids in this partition, and at most m remaining axes. Lemma 3.2 bounds the cost of one core by $O(v4^m)$. There are $O_A(v)$ choices of s, b, a, so all cores cost $O_A(v^2 4^m)$.

For each t, s, b, a, N the branch in (3.3) is one product with at most three inputs. The number of choices is $O_A(v^2 2^m)$. Attach each to the union for its unique output mask $F(x_s) \cup N$. There are at most $O_A(v2^m)$ suffix states and $O_A(2^m)$ full-grid states, and the latter have at most $|A|$ inputs each. Counting their wires as well as their gates adds $O_A(v^2 2^m)$. Absent states and branches decrease these counts. Together with the constant intervals this proves the bound. □

Proposition 4.2. *All exact-mask circuits for the full grid X can be constructed simultaneously with size*

$$O_A\left(V \prod_{i \in I} (1 + 4q_i)\right). \tag{4.1}$$

The number of bit operations is this bound times a polynomial in $|I| + \log(V + 1)$.

Proof. For fixed varying axes T, there are V/v_T subgrids, one for each choice of their fixed coordinates. Summing Lemma 4.1 over every subgrid gives

$$\begin{aligned} \sum_{T \subseteq I} \frac{V}{v_T} O_A(v_T^2 4^{|T|}) &= O_A\left(V \sum_{T \subseteq I} \prod_{i \in T} (4q_i)\right) \\ &= O_A\left(V \prod_{i \in I} (1 + 4q_i)\right). \end{aligned}$$

Build subgrids in increasing order of $|T|$, then cores, suffix states, and full-grid states, as in the proofs. This is a topological construction. Represent a subgrid by its varying-axis set and fixed coordinates, a vertex by its coordinate tuple, and a mask by a bit vector. All indices have polynomial length in $|I| + \log(V + 1)$. For each possible differing vertex one may scan all v_T vertices to obtain the partition and the prefix lengths r_H; over all vertices this takes $O(v_T^2)$ coordinate operations. Every remaining loop has one iteration per state, mask pair, or wire counted above. Feasibility flags are propagated by conjunction at products and disjunction at unions. Thus all indexing, arithmetic, and feasibility work fit within the stated polynomial factor. □

Proof of Theorem 1.1. For $n \ge 2$ apply Proposition 4.2 to each of the g grids from (2.1), with $q_p = p$. Their total size is

$$O_A\left(gR \prod_{p \in P} (1 + 4p)\right) = O_A\left(n \prod_{p \in P} (1 + 4p)\right).$$

Combine their exact masks by Lemma 3.2, at additional cost $O(g4^k)$, and choose the output with mask P. The grid position sets are disjoint, so all products remain decomposable. By Lemmas 2.1 and 2.2, the chosen output represents precisely $\mathrm{Prim}_A(n)$. This output is nonempty: for distinct $a, b \in A$, the word $a^{n-1}b$ is primitive, because the number of occurrences of b in a dth power would be divisible by d. The additional cost is absorbed by the displayed bound. Indeed, $g4^k \le gR \prod_{p \in P}(1 + 4p)$. As $1 + 4p \le 5p$ and $R \le n$, this proves both (1.1) and $O_A(n^2 5^k)$.

For completeness, $5^{\omega(n)} = n^{o(1)}$ follows without a prime number estimate. If the k distinct primes are increasing, the jth is at least $j+1$, so $(k+1)! \le n$. For $k \ge 4$, at least $k/2$ factors in this factorial are at least $k/2$, whence

$$\log n \ge \frac{k}{2}\log(k/2).$$

It follows that $k = o(\log n)$: when k is large the ratio $k/\log n$ is at most $2/\log(k/2)$, and bounded values of k also give a vanishing ratio. Thus $5^k = n^{o(1)}$. Also $k \le \log_2 n$, so any fixed polynomial in $k + \log n$ is $n^{o(1)}$.

Trial division finds the distinct primes of n within the stated time. The position relabeling can be tabulated by enumerating $0 \le r < g$ and $0 \le t < R$ and computing the residues in (2.1); no search over assignments is involved. Together with Proposition 4.2, these facts give construction time $n^{2+o(1)}$ in the bit model. For $n = 1$, use the disjoint union of the $|A|$ literals at its sole position. For $n = 0$, use the nullary union. □

5. Exact power orders

We finish with a consequence for the exact order of a word. The usual uniqueness of a primitive root can be proved directly in the cyclic notation used here.

Lemma 5.1. *Every nonempty word w has a unique expression $w = u^d$ with u primitive and $d \ge 1$.*

Proof. For a word of length n, let

$$H = \{h \in \mathbb{Z} : w(j+h) = w(j) \text{ for every } j \in \mathbb{Z}/n\mathbb{Z}\}.$$

This is an additive subgroup of $\mathbb{Z}$ containing $n\mathbb{Z}$, so $H = m\mathbb{Z}$ for a positive divisor m of n. The word is the (n/m)th power of its prefix u of length m. If u were a proper power, its shorter block length would belong to H, contrary to minimality of m.

If also $w = v^e$ with v primitive, then $|v| \in H$ and hence $m \mid |v|$. Invariance under shift m says that the prefix v is itself a power of the prefix of length m. Primitivity forces $|v| = m$, so $v = u$ and $e = n/m$. □

Call d in Lemma 5.1 the *order* of w.

Corollary 5.2. *Fix A as in Theorem 1.1. For any $n \ge 1$ and any set D of positive divisors of n, the length-n words whose orders belong to D have a smooth decomposable disjoint-union circuit of size $O_A(n^2 5^{\omega(n)})$, constructible in $n^{2+o(1)}$ bit operations. This includes a circuit for each individual exact order.*

Proof. Fix $d \mid n$ and put $m = n/d$. A word has order d exactly when it is u^d for $u \in \mathrm{Prim}_A(m)$. Take the circuit of Theorem 1.1 for length m. For every literal $i : a$, replace it by the product of the d literals

$$i : a,\ (i+m) : a,\ \ldots,\ (i+(d-1)m) : a.$$

For distinct source positions i, the corresponding position sets are disjoint. Consequently every original product stays decomposable, and equal original domains give equal expanded domains. The expansion of an assignment is injective: restriction to the first m positions recovers it. Disjoint unions therefore remain disjoint. The output represents exactly the words of order d.

Share equal source literals before replacing them. There are at most $|A|m$ of them, so the replacements add $O_A(n)$ gates and wires. The circuit for this order has size

$$O_A\Big((n/d)^2 5^{\omega(n/d)} + n\Big).$$

Construct such circuits separately for $d \in D$ and take the union of their outputs. All domains are the same, and the outputs are pairwise disjoint by Lemma 5.1. Finally,

$$\sum_{d|n} (n/d)^2 5^{\omega(n/d)} \leq n^2 5^{\omega(n)} \sum_{d=1}^{\infty} d^{-2} \leq 2n^2 5^{\omega(n)},$$

where the last inequality follows by comparing the terms from $d = 2$ onwards with $\int_1^\infty x^{-2}\,dx$. Also n times the number of divisors of n is at most n^2. These estimates give the size bound. The same summation, with the polynomial indexing factors already bounded uniformly in $\log n$, proves the construction-time bound. If D is empty, a constant assignment on the full domain multiplied by a nullary union gives the required empty output with size $O(n)$. □

References

[1] Antoine Amarilli, Mikaël Monet, Paul Raphaël, and Sylvain Salvati, *On the complexity of language membership for probabilistic words*, in *43rd International Symposium on Theoretical Aspects of Computer Science (STACS 2026)*, Leibniz International Proceedings in Informatics **364** (2026), article 5, 5:1–5:21. doi:10.4230/LIPIcs.STACS.2026.5.

[2] Antoine Amarilli, Mikaël Monet, Paul Raphaël, and Sylvain Salvati, *On the complexity of language membership for probabilistic words*, full version, arXiv:2510.08127v2, 22 June 2026. `https://arxiv.org/abs/2510.08127v2`.

Email address: `howew@purdue.edu`

Email address: `huffma97@purdue.edu`

Email address: `jenwin@purdue.edu`

Email address: `kruerl@purdue.edu`