COUNTING PALINDROMIC HAMILTONIAN WORDS IN THE SYMMETRIC GROUP ON FOUR LETTERS

COLE DEXTER AND GPT 6.2 ASTRA

ABSTRACT. There are exactly 764,135,616 palindromic words of length 23 in S_4 whose successive partial products, including the identity, exhaust the group. They form 31,838,984 orbits under group automorphisms. These counts answer two questions of Peter Kagey. We derive a subset recurrence over any finite group, evaluate it for S_4, and reproduce the result by an independent search that grows the walk from its middle edge. A Lean formalization proves the recurrence and its specialization to the 23-step problem. Two C++ programs supply the numerical evaluation.

1. THE COUNT

For a word $w = (g_1, \dots, g_\ell)$ over a group G, put $p_0 = 1$ and $p_i = p_{i-1}g_i$. The word is *Hamiltonian* if $p_0, \dots, p_\ell$ list every element of G exactly once. It is *palindromic* if $g_i = g_{\ell+1-i}$ for every i. Kagey asked for the number of such words in S_4, following a count of 12 in S_3. In a later comment, he asked how many remain after identifying words under the coordinatewise action of group automorphisms [1]. The starting partial product is fixed at the identity throughout.

Theorem 1.1. *The number of palindromic Hamiltonian words in S_4 is* 764,135,616. *The number up to automorphisms of S_4 is* 31,838,984. *For a prescribed nonidentity terminal permutation the counts are given in Table 1.*

Terminal cycle type	Class size	Per terminal	All terminals in class
Transposition	6	44,589,360	267,536,160
Double transposition	3	31,469,376	94,408,128
3-cycle	8	18,574,068	148,592,544
4-cycle	6	42,266,464	253,598,784

TABLE 1. Counts for the final partial product. The identity cannot be a terminal because it occurs first.

2020 *Mathematics Subject Classification.* Primary 05A15; Secondary 05C45, 20B30.
Key words and phrases. palindrome, partial products, Hamiltonian path, finite group.
Cole Dexter and GPT 6.2 Astra are co-authors and owners of this research.

2. Peeling equal outside letters

Fix a finite group G. For $S \subseteq G$ and $x, y \in G$, let $C_k(S; x, y)$ count palindromic words of length $2k+1$ whose walk from x ends at y and visits exactly S without repetition. If $x = y$ or either endpoint lies outside S, put $C_k(S; x, y) = 0$.

Proposition 2.1. *For distinct $x, y \in S$ one has*

$$C_0(S; x, y) = \mathbf{1}_{S=\{x,y\}}, \tag{1}$$

$$C_{k+1}(S; x, y) = \sum_{g \in G} C_k(S \setminus \{x, y\}; xg, yg^{-1}). \tag{2}$$

If $|G| = 2k+2$, the total number of palindromic Hamiltonian words is $\sum_{y \in G} C_k(G; 1, y)$.

Proof. In length one the unique possible letter is $x^{-1}y$. A palindrome of length at least three has the unique form gug, with u palindromic. Removing its first and last vertices gives the walk of u from xg to yg^{-1}. All its vertices are in $S \setminus \{x, y\}$, and it must visit each exactly once. Conversely, attach g at both ends of any such interior word. This restores the two deleted vertices without repetition, and recovers the original endpoints. Summing over the disjoint choices of g proves (2). For a full walk starting at the identity the terminal vertex is uniquely determined, so the final sum is disjoint as well. □

Equations (1)–(2) apply directly to S_n for every $n \geq 2$. The factor controlling the subset search is the order of the group, which is already 120 for S_5.

3. Evaluating the recurrence in the symmetric group on four letters

Left translation preserves step labels. Hence we may normalize the first endpoint to 1. Write p for the terminal element and $I = S \setminus \{1, p\}$. For a choice $g \in I$ in (2), the last interior vertex is pg^{-1}, which must also belong to I and differ from g. Translating that interior by g^{-1} yields a normalized state with set $g^{-1}I$ and terminal $g^{-1}pg^{-1}$. Simultaneous conjugation of every vertex and every letter gives a bijection of walks. Thus we may conjugate the whole interior so that its terminal is a fixed representative of its conjugacy class. There are four possible nonidentity classes.

The two programs use opposite conventions for composing permutations. Inversion is an isomorphism from a group to its opposite and preserves cycle type in S_4, so the conventions give identical counts in each class.

The program enumerates permutations in lexicographic one-line order, choosing the least permutation in each class. It encodes S by a 24-bit mask. For each terminal class and each first letter it precomputes the permutation of bits induced by the translation and conjugation just described. A table caches each reachable count by the pair consisting of the terminal class and the mask with its constant identity bit omitted. At a two-vertex state the

return value is one by (1). Each recursion removes two vertices. With the four terminal representatives as roots the calculation visits 1,755,798 distinct states of size greater than two and gives Table 1. The first twelve letters determine a length-23 palindrome, and a Hamiltonian word contains no identity letter. Thus $23^{12} < 2^{64}$ bounds the total, all subproblem counts and the partial sums in the calculation.

4. Independent enumeration from the middle

For an independent check, fix the middle letter h. Translate the two central vertices to $1, h$, and grow the word outward. If the current left and right vertices are a, b, choosing an unused new left vertex v forces the common new letter to be $v^{-1}a$. Consequently the new right vertex is $z = bv^{-1}a$. Make this extension exactly when z is unused and $v \neq z$. After eleven extensions the walk covers S_4. If its final boundary vertices before translation are a', b', its rooted terminal product is $(a')^{-1}b'$.

Every rooted palindrome determines exactly one normalized central edge and one sequence of choices of the new left vertices. Conversely translating a completed inside-out walk by the inverse of its first vertex yields one rooted palindrome. Conjugation allows one search per middle-letter class. This search uses direct depth-first enumeration and produces the bivariate counts in Table 2. Multiplying each row by the size of its middle-letter class and summing each column gives the four class totals in Table 1.

	Terminal cycle type			
Fixed middle letter	(2)	(2, 2)	(3)	(4)
Transposition	25,492,960	0	0	19,096,400
Double transposition	0	14,215,808	17,253,568	0
3-cycle	0	6,470,088	12,103,980	0
4-cycle	19,096,400	0	0	23,170,064

Table 2. Counts for one specified middle letter and a terminal ranging over a class. One-cycles are suppressed in the column headings.

The zero entries follow from sign, since the sign of every paired letter occurs twice. For a second check, write the palindrome as uhu^R, where u^R is the reversal of u, and append the letter c inverse to the terminal product. This closes the Hamiltonian path into a cycle. Deleting the edge labelled h at the middle position and starting immediately after it gives the palindrome $u^R cu$. Left translation restores the initial vertex to 1 and preserves the step labels. The new terminal product is h^{-1}, so the same operation recovers uhu^R. Inversion preserves cycle type in S_4, so the table becomes symmetric after each row is multiplied by its class size. The middle-edge program also reproduces the counts 1 for S_2 and 12 for S_3.

5. Automorphism orbits

An automorphism fixing a word fixes its partial products. For a Hamiltonian word these run through the whole group, so its stabilizer is trivial. Every orbit therefore has size $|\operatorname{Aut}(G)|$.

For completeness, automorphisms of S_4 preserve the transpositions, since among nonidentity involutions only transpositions have centralizer order four. Regard the transpositions as edges of the complete graph on four points. Two such edges meet if and only if their distinct transpositions fail to commute. Three pairwise intersecting edges form either a star or a triangle. The three transpositions of a star generate S_4, whereas a triangle generates a subgroup fixing the fourth point. Thus every automorphism permutes the four stars. Each transposition is determined as the common edge of two stars, so this action is faithful. There are at most 24 automorphisms, and conjugation supplies 24 because the center is trivial. Dividing the total by 24 gives the second assertion of Theorem 1.1.

6. Verification and reproduction

The archive [2] contains both C++20 programs, their full output, and the Lean 4 sources. The Lean development characterizes the words of Proposition 2.1 by their step lists and visited vertices, proves the recurrence and its specialization to S_4, and proves the translation and automorphism identities used in the subset algorithm. It also proves the trivial-stabilizer assertion and the order of $\operatorname{Aut}(S_4)$. The orbit quotient is expressed in Lean as the same finite sum divided by 24. The evaluation yielding the two tables was performed by the two C++ implementations outside the Lean kernel. The archive includes exact build and replay commands.

References

[1] P. Kagey, *Counting Palindromic Hamiltonian sequences on S_4*, Mathematics Stack Exchange, June 13, 2020. `https://math.stackexchange.com/q/3717562`. See also the author's comment of April 22, 2022.

[2] C. Dexter and GPT 6.2 Astra, *Palindromic Hamiltonian words in the symmetric group on four letters*, computational and formal proof archive. Versioned sources, outputs, and reproduction instructions.