THE AUSTRALIAN NATIONAL UNIVERSITY

TR-CS-97-03

Knight's Tours of an 8 × 8 Chessboard

Brendan D. McKay

February 1997

Joint Computer Science Technical Report Series

Department of Computer Science
Faculty of Engineering and Information Technology

Computer Sciences Laboratory
Research School of Information Sciences and Engineering

This technical report series is published jointly by the Department of Computer Science, Faculty of Engineering and Information Technology, and the Computer Sciences Laboratory, Research School of Information Sciences and Engineering, The Australian National University.

Please direct correspondence regarding this series to:

Technical Reports
Department of Computer Science
Faculty of Engineering and Information Technology
The Australian National University
Canberra ACT 0200
Australia

or send email to:

`Technical.Reports@cs.anu.edu.au`

A list of technical reports, including some abstracts and copies of some full reports may be found at:

`http://cs.anu.edu.au/techreports/`

Recent reports in this series:

TR-CS-97-02	Xun Qu and Jeffrey X. Yu. *Mobile file filtering*. February 1997.
TR-CS-97-01	Peter Arbenz and Markus Hegland. *The stable parallel solution of general narrow banded linear systems*. January 1997.
TR-CS-96-09	Ralph Back, Jim Grundy, and Joakim von Wright. *Structured calculational proof*. November 1996.
TR-CS-96-08	David Hawking and Paul Thistlewaite. *Relevance weighting using distance between term occurrences*. August 1996.
TR-CS-96-07	Andrew Tridgell, Paul Mackerras, David Sitsky, and David Walsh. *AP/Linux - initial implementation*. June 1996.
TR-CS-96-06	M. Hegland, M. H. Kahn, and M. R. Osborne. *A parallel algorithm for the reduction to tridiagonal form for eigendecomposition*. June 1996.

Knight's Tours of an 8×8 Chessboard

Brendan D. McKay
Computer Science Department, Australian National University,
Canberra, ACT 0200, Australia
bdm@cs.anu.edu.au

Abstract.

We describe a computation that determined the number of knight's tours of a standard chessboard. We also verify Knuth's count of tours with a symmetry. The total number of undirected tours is 13,267,364,410,532 and the number of equivalence classes under rotation and reflection of the board is 1,658,420,855,433.

1. Introduction.

A *knight's tour* is a hamiltonian cycle in the graph defined by legal knight's moves on a chessboard. We only consider tours that are cycles, and do not distinguish between a tour and its reverse. A recent paper [3] describes an elegant method that can solve the difficult problem of determining the total number of knight's tours. Unfortunately, the implementation of the algorithm was performed incorrectly, leading to the wrong answer. The long task of repeating the computation is underway at the time of writing. Nevertheless, the desirability of independent verification is clear, and that is the purpose of this note.

2. Counting all tours.

We will explain our method using the same tour as used in [3]. That tour is shown in Figure I on the left, and on the right it is broken into two pieces with the steps crossing the central line deleted. The subgraph of the tour induced by the squares in the lower half of the board consists of a set of paths, some of them possibly trivial. That set of paths is called the *lower half-tour* of the tour. The lower half-tour in Figure I consists of 6 disjoint paths, including the trivial paths in b4, f4 and h4.

More generally, we will use the expression *lower half-tour* to be a set of (possibly trivial) vertex-disjoint paths whose vertex-set is the lower half of the board, whose edges are legal knight's moves, and whose endpoints lie in ranks 3 and 4.

An *upper half-tour* is defined similarly. In Figure I, the lower and upper half-tours are shown on the right.

The *path structure* of a half-tour is the set of pairs of endpoints of the paths comprising it. For the lower half-tour in Figure I, the path structure is {a4-b3, b4-b4,

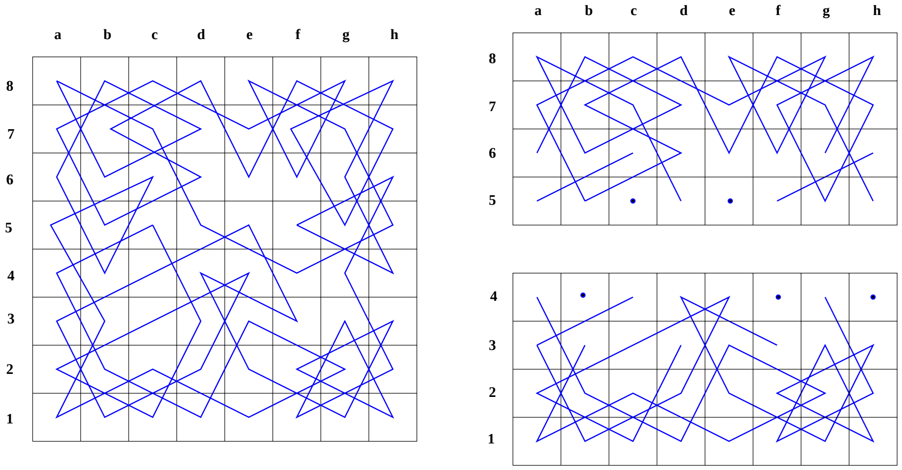

Figure I. A knight's tour and its two induced half-tours.

c4-d3, f4-f4, g4-f3, h4-h4}. The order in which the two endpoints of each path are written is immaterial.

Lemma 1. *The number of tours which induce a particular pair of upper and lower half-tours depends only on the path structures of the half-tours.*

Proof. To complete a pair of half-tours into a tour, we must add knight steps across the middle line, joining endpoints of paths in the lower half-tour to endpoints of paths in the upper half-tour. The number of ways of doing this successfully depends only on the positions of the endpoints of the paths in the two half-tours, and on which of those endpoints belong to the same path. This information is precisely the path structure. ∎

The first step in our computation was to find all half-tour path structures, and the number of half-tours corresponding to each. This was achieved by a standard back-track algorithm in about 6 hours. There are altogether 70433448 half-tours, having 7934470 different path structures. There are between 1 and 5680 half-tours with each path structure.

In order to complete the computation, we need to consider each possible pair of path structures for the upper and lower half tours, and determine the number of tours that induce them. However, the number of such pairs is much too large as yet so we do a further grouping of cases.

The *type* of a lower half-tour is a pair $(\boldsymbol{a}, \boldsymbol{b})$, where $\boldsymbol{a} = a_1, a_2, \ldots, a_8$ and $\boldsymbol{b} = b_1, b_2, \ldots, b_8$. The values $a_1, a_2, \ldots, a_8$ are, reading across rank 4: 0 for the internal vertex of a path, 1 for the endpoint of a non-trivial path, and 2 for a trivial path. The values $b_1, b_2, \ldots, b_8$ contain the same information for rank 3. The lower half-tour of Figure I has type $((1,2,1,0,0,2,1,2),(0,1,0,1,0,1,0,0))$.

The type is clearly a function of the path structure. Altogether there are 453741 possible types for half-tours, with 6357 values of $\boldsymbol{a}$ and 1296 values of $\boldsymbol{b}$ occurring.

A particular tour has a pair of types $(\boldsymbol{a}_L, \boldsymbol{b}_L), (\boldsymbol{a}_U, \boldsymbol{b}_U)$ for its lower and upper half-tours. Whether an arbitrary such pair of types corresponds to any tours can be very often be determined in the negative by examining the pair $(\boldsymbol{a}_L, \boldsymbol{b}_U)$, as all of $\boldsymbol{b}_U$ must be accounted for by knight steps between rank 4 and rank 6 while obeying the limits imposed by $\boldsymbol{a}_L$. If there is at least one such set of knight steps, we call $\boldsymbol{a}_L$ and $\boldsymbol{b}_U$ *compatible*. Similarly for $(\boldsymbol{a}_U, \boldsymbol{b}_L)$.

We are now able to describe the remainder of the computation.

Consider each pair of types $(\boldsymbol{a}_L, \boldsymbol{b}_L), (\boldsymbol{a}_U, \boldsymbol{b}_U)$ such that $(\boldsymbol{a}_L, \boldsymbol{b}_U)$ and $(\boldsymbol{a}_U, \boldsymbol{b}_L)$ are compatible. Some of these are equivalent under horizontal or vertical flips of the board or their product; we selected the lexicographically least from each equivalence class and calculated a multiplicity (1, 2 or 4) to compensate.

For each of the remaining pairs of types $(\boldsymbol{a}_L, \boldsymbol{b}_L), (\boldsymbol{a}_U, \boldsymbol{b}_U)$, we determined all possible ways of choosing a set of knight steps between ranks 3 and 5, 4 and 5, or 4 and 6 to match these types. (The total number of such sets for the entire computation was about 361 million.) Finally, for each such set of steps, each path structure S_L with type $(\boldsymbol{a}_L, \boldsymbol{b}_L)$, and each path structure S_U with type $(\boldsymbol{a}_U, \boldsymbol{b}_U)$, we tested whether the union of the three consisted of a single circuit. If yes, we had identified a collection of tours with cardinality equal to the product of the number of half-tours with path structure S_U, the number of half-tours with path structure S_L, and the multiplicity defined in the previous paragraph.

This final part of the computation was quite expensive, taking 232 hours on a mixture of Sun workstations. However, this time is quite small compared to the time required for the method described in [3]. The result was as stated in the Abstract.

Equivalence classes.

It was pointed out by Knuth [2] that the only symmetry of an 8×8 board which may preserve a knight's tour is a rotation by 180 degrees. This is an easy but interesting exercise that we will leave for the reader. Knuth further reported a count of 608,233 equivalence classes of such symmetrical tours, each such class containing 4 tours. Since all other equivalence classes contain 8 tours, we can calculate the total number of equivalence classes to be the number stated in the Abstract.

Is it correct?

Every computer programmer knows that errors in programming or execution can escape the most rigorous checking. Although we feel confident our result is correct, independent verification is needed for practical certainty. Hopefully, that verification will be provided when the rerunning of [3] is complete. As a partial check of our result, we ran it all again in left-right mirror image, obtaining the same result. We also applied the method of Knuth [1] to estimate the total number of tours by performing random probes of an exhaustive search tree, obtaining an estimate close to our answer.

Finally, we verified Knuth's count of 608,233 classes of symmetric tours using a simple backtrack program that ran for five minutes.

References.

[1] D. E. Knuth, Estimating the efficiency of backtrack programs, *Math. Comput.*, **29** (1975) 121–136.

[2] D. E. Knuth, Private communication, 1996.

[3] M. Löbbing and Ingo Wegener, The number of knight's tours equals 33,439,123,484,294, *Electronic J. Combin.*, **3** (1996) R5.