Multilinear algebra and chess endgames

Lewis Stiller*
Department of Computer Science
The Johns Hopkins University
Baltimore, MD 21218

July 13, 1995

1 Introduction

Parallel and vector architectures can achieve fairly high peak bandwidth, but it can be difficult for the programmer to design algorithms that exploit this bandwidth efficiently. Application performance can depend heavily on unique architecture features that complicate the design of portable code [160,163].

The work reported here is part of a project to explore the extent to which the techniques of multilinear algebra can be used to simplify the design of high-performance parallel and vector algorithms [91]. The approach advocated comprises the following:

- A set of fixed, structured matrices that encode architectural primitives of the machine, in the sense that left-multiplication of a vector by this matrix is efficient on the target architecture, is defined.
- The application problem is formulated as a matrix multiplication.
- The matrix corresponding to the application is factored in terms of the fixed matrices using addition, tensor product, and matrix multiplication as combining operators.

*Supported by U.S. Army Grant DAAL03-92-G-0345

- Code is generated from the matrix factorization.

This approach to the design of algorithms for vector and parallel architectures has been used by previous researchers in the domain of signal processing algorithms [72, 74, 89, 170, 171]. The success of that work motivates the attempt to generalize the domain of application of the multilinear-algebraic approach to parallel programming [157].

The author has utilized the methodology presented in this paper in several domains, including parallel N-body codes, Fortran 90 communication intrinsic functions, and statistical computations [68, 156, 159]. In each case, significant speedup over the best previous known algorithms was attained. On the other hand, it is clear that this methodology is intended to be applicable only to a narrow class of domains: those characterized by regular and oblivious memory access patterns. For example, parallel alpha-beta algorithms cannot be formulated within this paradigm.

This paper describes the application of the multilinear algebraic methodology to the domain of chess endgames. Dynamic programming was used to embed the state space in the archictecture. By successively unmoving pieces from the set of mating positions, the set of positions from which White could win can be generated.

This domain presents a particularly interesting challenge to the multilinear-algebraic parallel-program design methodology:

- The formalism for the existing multilinear algebra approach had been developed to exploit parallelization of linear transformations over a module. This formalism needed to be generalized so that it would work over Boolean algebras.

- The symmetry under a noncommutative crystallographic group had to be exploited without sacrificing parallelizability.

- The state-space size of 7.7 giganodes was near the maximum that the target architecture could store in RAM.

This paper describes the resolution of these problems. There are two main results reported here:

1. Table 1 gives equations defining the dynamic programming solution to chess endgames. Using the techniques described in this paper, the factorizations can be modified to produce efficient code for most current parallel and vector architectures.

2. Table 2 presents a statistical summary of the state space of several 6-piece chess endgames. This table could not have been generated in a practicable amount of time using previous techniques.

The organization of this paper will be as follows.

Section 2 provides the background of the chess endgame problem. A survey of some human analysis of chess endgames is given, followed by a survey of previous work in the area of computer endgame analysis.

Section 3 introduces some basic concepts of parallel processing.

Section 4 describes previous work in the area of tensor product formalism for signal-processing computations. First, the mathematical framework used is described; second, the relationship between this mathematical framework and executable code will be given; finally, the sequential, parallel, and vector FFT algorithms will be presented to illustrate the methodology.

Section 5 develops a generalized version of the formalism of Section 4, and describes the chess endgame algorithm in terms of this formalism.

Section 6 presents the main result, Table 1. Subsection 6.2 describes how the equations of Table 1 are modified to exploit symmetry. The derivation of crystallographic FFTs is used as a motivating example in the derivation of symmetry-invariant equations.

Section 7 discusses some implementation issues.

Section 8 presents some of the chess results discovered by the program. The appendix contains best play from a position requiring 243 moves for White to win.

2 Background

The next two subsections discuss the background of the human and computer analysis of endgames, particularly endgames containing at most 6

pieces. Due to space limitations, much significant work in this historical survey has been omitted.

2.1 Human analysis

Endgame analysis appears to date from at least the 9th century, with al-'Adlī's analysis of positions from ♔♖♔♘[1] [3, plate 105] and ♔♖♘♔♖ [3, plate 112]. However, the rules were slightly different in those days, as stalemate was not necessarily considered a draw. The oldest extant collection of compositions, including endgames, is the Alfonso manuscript, ca. 1250, which seems to indicate some interest during that time in endgame study [129, pp.111–112].

Modern chess is generally considered to have begun roughly with the publication, probably in 1497, of Luis Ramirez de Lucena's *Repetición de amores y arte de ajedrez* [53][2]. Ruy Lopez de Sigura's 1561 book briefly discusses endgame theory, but its main impact on this work would be the introduction of the controversial 50-move rule, under which a game that contains 50 consecutive moves for each side without the move of a pawn or a capture could be declared drawn [54, pp.55–56] [140].

Pietro Carrera's 1617 *Il gioco de gli scacchi* discussed a number of fundamental endgames such as ♔♕♔♗♗, and certain 6-piece endgames such as ♔♖♖♔♖♘ and ♔♖♖♔♖♗ [46, Book 3, p. 176–178]. A number of other authors of the time, such as Philip Stamma (1737), François-André D. Philidor (1749), and Gioacchino Greco (1624), began developing the modern theory of endgames [75,130,152]. Giovanni Lolli's monumental *Osservazioni teorico-pratiche sopra il giuoco degli scacchi* (1763) would be one of the most significant advances in endgame theory for the next 90 years [105,138]. Lolli analyzed the endgame ♔♕♔♗♗, and he agreed with the earlier conclusion of Salvio (1634) that the endgame was a general draw for White [142]. This assessment would stand substantially unchanged until Kenneth Thompson's computer analysis demonstrated the surprising 71 move win [167]. Notwith-

[1] In listing the pieces of an endgame, the order will be White King, other White pieces, Black King, other Black pieces. Thus, ♔♖♔♘ is the same as ♔♖♚♞, and comprises the endgame of White King and White Rook against Black King and Black Knight

[2] Ironically, this work does not contain the famous "Lucena position" from ♔♖♙♔♖, which seems to have been first published by Alessandro Salvio in 1634, who attributed it to Scipione Genovino.

standing this error, Lolli did manage to discover the unique ♔♕♔♗♗ position in which White to play draws but Black to play loses [105, pp.431–432].

Bernhard Horwitz and Josef Kling's 1851 *Chess Studies* contained a number of influential endgame studies, although their analysis of ♔♗♗♔♘ was questioned by A. John Roycroft (1972) [95, pp.62–66] [138, p. 207]. The Horwitz and Kling assessment was definitively shown to be incorrect by the independent 1983 computer analyses of Thompson and Ofer Comay [139, 166].

Alfred Crosskill (1864) [52] gave an analysis of ♔♖♗♔♖ in which he claimed a win in more than 50 moves was required; this was confirmed by computer analysis of Thompson. The Crosskill analysis was the culmination of a tradition of analysis of ♔♖♗♔♖ beginning at least from the time of Philidor [130, pp.165–169].

A generation later, Henri Rinck and Aleksei Troitzky were two of the most influential endgame composers of their time. Troitzky is well-known for his analysis of ♔♘♘♔♙—he demonstrated that > 50 move wins were at times required [173]. Rinck was a specialist in pawnless endgames, composing more than 500 such studies [134, 135], including some with 6 pieces. Troitzky summarized previous work in the area of ♔♘♘♔♙, beginning with a problem in ♔♘♔♙ from the 13th-century Latin manuscript *Bonus Socius* [55], and reserved particular praise for the systematic analysis of this endgame in an 18th-century manuscript by Chapais [48]. (An early version of the program reported in this paper resulted in the first published solution for the entire endgame [153]).

The 20th century saw the formal codification of endgame theory by scholars such as Johann Berger (1890) [38], André Chéron (1960) [49], Machgielis [Max] Euwe (1940) [62], Reuben Fine (1941) [65], Yuri L. Averbakh (1982) [27], and many others. Some work focusing particularly on pawnless 6-piece endings has also appeared, for example, [39, 98, 137].

Currently the Informator *Encyclopedia of Chess Endings* series [107], which now uses some of Thompson's computer analysis, is a standard reference. John Nunn has written several books based on that work [125, 126].

Additional historical information can be found in the references [71, 82, 122, 138].

2.2 Friedrich Amelung and Theodor Molien: A historical note

This subsection discusses the work of Friedrich Amelung and Theodor Molien as it pertains to pawnless chess endgame analysis.

Friedrich Ludwig Amelung (March 11, 1842–March 9, 1909) was a Latvian chess player and author who edited the chess column of the Riga newspaper *Düna-Zeitung*. He studied philosophy and chemistry at the University of Dorpat from 1862 to 1879, and later became a private teacher and director of a mirror factory [104, p.11] [22, 85]. He published a number of endgame studies and analyses of endgames, and began a systematic study of pawnless endgames. For example, he explored the endgame ♔♕♔♖♘ in detail [13, 14]; this endgame was shown to have unexpected depth, requiring up to 46 moves to win, in later work by the author [153]. He also published an article on ♔♗♘♔♘ and ♔♗♗♔♘ [5], which were not exhaustively analyzed until the 1980s [153, 167].

However, his main interest to our work actually inheres in two major projects: an analysis of the 4-piece endgame ♔♖♔♘, which appeared in 1900 [7–12], and his studies of certain pawnless 6-piece endgames [15–21].

Amelung's 1900 analysis of ♔♖♔♘ was significant because it contained the first histogram known to the author of a pawnless endgame or, for that matter, of any endgame [11, pp.265–266]. This table listed the approximate number of positions in ♔♖♔♘ from which White could win and draw in 2–5 moves, 5–10 moves, 10–20 moves, and 20–30 moves. Such tables have been a mainstay of computer-age endgame analysis, of course. The existence of this early analysis does not appear to have been known to contemporary workers, although it appeared in a widely read and influential publication, *Deutsche Schachzeitung*.

Even more intriguing, however, is Amelung's comment that an even earlier, exact numerical analysis, containing the number of win-in-k moves for each k of a four-piece chess endgame was known, and was due to "Dr. Th. Mollien, der Mathematiker von Fach ist"; that is, to the professor "Th. Mollien."

Theodor Molien(September 10, 1861–December 25, 1941)[3] was born in

[3] There are a number of variant English spellings of Molien's name: Molin [31], Mollin [12, p.5], Mollien [11, p.265], and Molien [4, 6]. His biography gives his name as Федор Эдуардович Молин (Fedor Eduardovich Molin) [92]. We will refer to him as Theodor

Riga.[4] His father, Eduard, was a philologist and teacher, and Theodor eventually became fluent in a number of languages, including Hebrew, Greek, Latin, French, Italian, Spanish, Portuguese, English, Dutch, Swedish, and Norwiegian, as well as German and Russian, of course. "If you read a hundred novels in a language," Molien liked to say, "you will know that language. [92, p.9]"[5] He studied celestial mechanics at Dorpat University (1880–1883) and also took courses from Felix Klein in Leipzig (1883–1885). His doctoral dissertation, which was published in *Mathematische Annalen* [113,114] proved a number of the fundamental structure theorems of group representation theory, including the decomposability of group algebras into direct sums of matrix algebras.

Molien's early papers on group representation theory [113–116,119], despite their importance, were obscure and difficult to understand. Indeed, his papers anticipated Frobenius' classic paper on the determinant of a group-circulant matrix [66], a fact which Frobenius readily admitted [78], although he had tremendous difficulty understanding Molien's work (letter to Alfred Knezer, May 6, 1898). In a letter to Dedekind, February 24, 1898, Frobenius wrote:

> You will have noticed that a young mathematician, Theodor Molien in Dorpat, has independently of me considered the group determinant. He has published, in volume 41 of of the *Mathematische Annalen* a very beautiful but difficult to read work "On systems of higher complex numbers [114]," in which he investigated non-commutative mulltiplication and obtained important general results of which the properties of the group determinant are special cases.[6]

Molien in conformity with his publications [113–116, 119].

[4] Molien's biographical information has been taken from Kanunov [92], which was translated for this project by Boris Statnikov.

[5] «Прочитайте сто романов на каком-либо языке,—любил говорить он позднее,—и Вы будете знать этот язык.»

[6] "Sie werden bemerkt haben, daßsich ein jungerer Mathematiker Theodor Molien in Dorpat unabhängig von mir mit der Gruppendeterminante beschäftigt hat. Er hat im 41. Bande der Mathematischen Annalen eine sehr schöne, aber schwer zu lesende Arbeit 'Ueber Systeme höherer complexer Zahlen' veröffentlicht, worin er die nicht commutative Multiplication untersucht hat und wichtige allgemeine Resultate erhalten hat, von denen die Eigenschaften der Gruppendeterminant specielle Fälle sind. [Excerpt from a transcription by Walter Kaufmann Bühler of a letter from Frobenius to Dedekind dated February 24, 1898. A copy of this transcription was kindly provided by Thomas Hawkins,

Despite these results, and despite Frobenius' support, Molien was rejected from a number of Russian academic positions, partly because of the Czarist politics of the time (according to Kanunov) and, at least in one case, because the committee considered his work too theoretical and without practical applications [92, pp.35–36]. After studying medieval mathematical manuscripts at the Vatican Library in 1899 [92, p.35], he accepted a post at the Tomsk Technological Institute in Siberia where he was cut off from the mathematical mainstream and became embroiled in obscure administrative struggles (he was, in fact, briefly fired). His remaining mathematical work had little influence and he spent most of his time teaching.

Thus, Molien's work was unknown or underestimated in the West for a long while, for example, Wussing's classic 1969 text barely mentions him [186]. With the publication of Thomas Hawkins series of articles on the history of group representation theory [76–78], the significance of Molien's contributions became better-known, and van der Waerden's 1985 history of algebra gives Molien due credit [182, pp.206–209,237–238].

Although it is not mentioned in Kanunov's biography, before Molien moved to Tomsk, he was one of the strongest players in Dorpat and was particularly known for his blindfold play (Ken Whyld, personal communication, 1995). He was president of the Dorpat chess club, and several of his games were published in a Latvian chess journal, *Baltische Schachblätter*, edited, for a time, by Amelung [4] [32, p.8]; one of his games (which he lost) won a "best-game" prize in the main tournament of the Jurjewer chess club in 1894 [120].

Molien's numerical studies of ♔♖♔♗ are alluded to several times in the chess journals of the time (about 1900) [6,117] [12, p.5] [11, p.265]. In 1898 he published four chess studies [118] based on his research into the endgame ♔♖♔♗ [12, p.5]. However, we have not been able to locate a publication of his complete results, despite the historical significance of such a document.

In any case, it seems to me to be an interesting coincidence that within a span of a few years Molien performed groundbreaking work in two apparently unrelated areas: group representation theory and quantitative chess endgame analysis, although his work in both areas was mostly ignored for a long time. There is, perhaps, some mathematical affinity between these areas as well,

---

Department of Mathematics, Boston University, and is excerpted here with the permission of Springer-Verlag.]"

since, as we shall see, the chess move operators can be encoded by a group-equivariant matrix; rapid multiplication of a group-equivariant matrix by a vector, in general, relies on the algebra-isomorphism between a group algebra and a direct sum of matrix algebras first noted by Molien [50,57,58,93]; massively parallel implementations are described by the author in [158, Chapter 7].

We now continue with our discussion of chess endgame history proper, particularly Amelung's work on pawnless endgames, of which his work on ♔♖♗♔♘♘ deserves special mention. Partly in response to the first edition of Johann Berger's influential 1890 manual of endings [38, 167–169], in 1902 Amelung published a three-part series in *Deutsche Schachzeitung*, perhaps the premier chess journal of its time, analyzing the endings of King, Rook and minor piece (♘ or ♗) against King and two minor pieces [15–17], and represented a continuation of Amelung's earlier work with Molien on the endgame ♔♖♔♘ [11]. Amelung indicated that the endgame ♔♖♗♔♘♘ was particularly interesting, and in 1908 he published a short article on the topic in *Für Haus und Familie*, a biweekly supplement to the Riga newspaper *Düna-Zeitung*, of which he was the chess editor [18]. Amelung's interest in this endgame was so great that he held a contest in *Düna-Zeitung* for the best solution to a particular example of this endgame [20]. A solution was published the next year [21], but Amelung died that year and was unable to continue or popularize his research. Consequently, succeeding commentators dismissed many of his more extreme claims, and his work seemed to pass into oblivion. It is discussed in the 1922 edition of Berger [39, p.223–233], but Amelung's work was criticized by the mathematician and chess champion Machgielis [Max] Euwe in his titanic 1940 study of pawnless endgames [63, pp.50–53].[7]

Indeed, *Düna-Zeitung* turned out to be an elusive newspaper; I was not able to locate any references to it in domestic catalogues and indices; the only copy I was able to find was archived at the National Library of Latvia. In addition to the remark about Molien, the research reported here argues for

[7] Euwe wrote "Dit eindspel [♔♖♗♔♘♘] biedt de sterkste partij zeer goede winstkansen. F. Amelung ging zelfs zoo ver, dat hij de verdediging als kansloos beschouwde, maar deze opvatting schijnt ojuist te zijn [63, p.50]", i.e., "This endgame [♔♖♗♔♘♘] offers the stronger side excellent winning chances. F. Amelung went so far as to say that the defense was hopeless, but this assessment seems to be untrue." (Translation from the Dutch is by Peter Jansen; translation into German is available in Euwe [64, Volume 5, Page 55].)

a renewed appreciation of the accuracy and importance of Amelung's work.

2.3 Computer endgame analysis

Although some have dated computer chess from Charles Babbage's brief discussion of automated game-playing in 1864, his conclusion suggests that he did not appreciate the complexities involved:

> In consequence of this the whole question of making an automaton play any game depended upon the possibility of the machine being able to represent all the myriads of combinations relating to it. Allowing one hundred moves on each side for the longest game at chess, I found that the combinations involved in the Analytical Engine enormously surpassed any required, even by the game of chess. [29, p. 467]

Automated endgame play appears to date from the ♔♖♔ construction of Leonardo Torres-Quevedo. Although some sources give 1890 as the date in which the automaton was designed, it was exhibited at about 1915 [33, 146].[8] Quevedo's automaton, which, unlike most later work, could move its own pieces, used a rule-based approach [162, 172], like that of Barbara J. Huberman's 1968 thesis [83]. By contrast, we are concerned with exhaustive analysis of endgames, in which the value of each node of the state-space is computed by backing up the game-theoretic values of the leaves.

The mathematical justification for the retrograde analysis chess algorithm was already implicit in the 1912 paper of Ernst Zermelo [187]. Additional theoretical work was done by John von Neumann and Oskar Morgenstern (1944) [181, pp.124-125].

The contemporary dynamic programming methodology, which defines the field of retrograde endgame analysis, was discovered by Richard Bellman in 1965 [37].[9] Bellman's work was the culmination of his work reported as early as 1961:

[8] "Torres believes that the limit has by no means been reached of what automatic machinery can do, and in substantiation of his opinions presents his automatic chess-playing machine" [162, p. 298].

[9] Bellman's article, strangely enough, is not generally known to the computer game community, and it is not included in Herik's bibliography [179].

> *Checkers and Chess.* Interesting examples of processes in which the set of all possible states of the system is indescribably huge, but where the deviations are reasonably small in number, are checkers and chess. In checkers, the number of possible moves in any given situation is so small that we can confidently expect a complete digital computer solution to the problem of optimal play in this game. In chess, the general situation is still rather complex, but we can use the method described above to analyze completely all pawn-king endings, and probably all endings involving a minor piece and pawns. Whether or not this is desirable is another matter [36, p.3].

Bellman had considered game theory from a classical perspective as well [34, 35], but his work came to fruition in his 1965 paper [37], where he observed that the entire state-space could be stored and that dynamic programming techniques could then be used to compute whether either side could win any position. Bellman also sketched how a combination of forward search, dynamic programming, and heuristic evaluation could be used to solve much larger state spaces than could be tackled by either technique alone. Bellman predicted that checkers could be solved by his techniques, and the utility of his algorithms for solving very large state spaces has been validated by Jonathan Schaeffer et al. in the domain of checkers and Ralph Gasser in the domain of Nine Men's Morris [69, 100, 143]. On the other hand, $4 \times 4 \times 4$ tic-tac-toe has been solved by Patashnik (1980) using forward search and a variant of isomorph-rejection based on the automorphism group computation of Silver (1967) [127, 145].

E. A. Komissarchik and A. L. Futer (1974) studied certain special cases of ♔♕♙♔♕, although they were not able to solve the general instance of such endgames [96]. J. Ross Quinlan (1979) analyzed ♔♖♔♘ from the point of view of a machine learning testbed [131, 132]. Hans Berliner and Murray S. Campbell studied the Szén position of three connected passed pawns against three connected passed pawns by simplifying the promotion subgames [41]. Campbell has begun to extend this idea to wider classes of endgames [45] . Peter J. Jansen has studied endgame play when the opponent is presumed to be fallible [86–88]. H. Jaap van den Herik et al. have produced a number of retrograde analysis studies of various 4-piece endgames, or of endgames with more than 4 pieces whose special structure allows the state-space size to be reduced to about the size of the general 4-piece endgame [56, 177, 180].

Danny Kopec has written several papers in the area as well [97].

The first retrograde analysis of general 5-piece endgames with up to one pawn was due to Thompson (1986) [167]. The significance of this work was twofold. First, many more moves were required to win certain endgames than had previously been thought. Second, the Thompson work invalidated generally accepted theory concerning certain 5-piece endgames by demonstrating that certain classes of positions that had been thought to be drawn were, in fact, won. The winning procedure proved to be quite difficult for humans to understand [112]. The pawnless 5-piece work of Thompson was extended to all pawnless 5-piece endgames and many 5-piece endgames with one pawn by an early version of the program discussed in this paper.

3 Parallel processing

The motivation for using parallel processing is to achieve increased computation bandwidth by using large numbers of inexpensive processors [160, 176]. In particular, it is hoped that the so-called "von Neumann bottleneck" between the CPU and the memory of a standard serial computer could be alleviated by massive parallelism [84]. There are, of course, many tradeoffs that can be made in the architecture of a computer, such as the type of interconnection network, the granularity of the processors, and whether each processor executes the same instruction (SIMD) or different instructions (MIMD) [43, 79, 174, 175].

One common form of interconnection network is the hypercube [102]. Consider a parallel computer with 2^n processors, each with some local memory. Place each processor at a distinct vertex of the unit cube in Euclidean n-space, and imagine each edge of the cube as a wire that directly connects the processors at its endpoints. Two processors connected by an edge can communicate directly in a single timestep. Each processor in an n-dimensional hypercube can be viewed as having a length n binary address, with processors connected if and only if their addresses differ in exactly one bit location.

The hypercube can compute the effect of the *end-off shift* matrix E_8. Applied to an array, E_8 shifts each element of the array down, filling the initial element with 0.

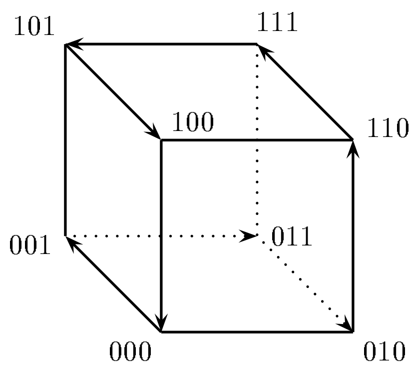


Figure 1: Left multiplication by an end-off shift matrix can be computed by Gray coding the coordinates of each element of the array. This figure illustrates a 3-bit Gray code, which can be thought of as an embedding of a cycle of length 8 into a 3-cube.

$$\mathsf{E}_8 \cdot \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_8 \end{pmatrix} = \begin{pmatrix} 0 \\ v_1 \\ \vdots \\ v_7 \end{pmatrix} \tag{1}$$

A hypercube can compute the effect of an end-off shift by Gray coding the coordinates of the elements of the array v so that v_i and v_{i+1} are physically adjacent in the hypercube [70] (see Figure 1). Higher dimensional shift patterns are also easy to compute on a hypercube [81].

It is also possible to perform arbitary permutations on a hypercube, although we shall not discuss the techniques required for that here. In practice general processor permutation is typically performed with hardware assistance [103, 123].

4 Parallel processing and tensor products

This section briefly summarizes some previous work on the application of tensor products to parallel processing, particularly to the parallel and vectorized computation of fast Fourier transform. The chess algorithm will be developed in the next section by generalizing this approach.

4.1 Mathematical preliminaries

Let $\mathfrak{V}_n$ be the space of length n vectors with entries in a field $\mathfrak{F}$. We let $\{\mathbf{e}_i^n\}_{i=1}^n$ be the "standard basis", that is, $\mathbf{e}_i^n$ is the vector whose ith component is 1 and whose other components are 0.

An element of $\mathfrak{V}_n$ may be thought of as a length n array whose elements are in $\mathfrak{F}$, or as an $n \times 1$ matrix over $\mathfrak{F}$ [106].

The mn basis elements of $\mathfrak{V}_n \otimes \mathfrak{V}_m$, $\{\mathbf{e}_i^n \otimes \mathbf{e}_j^m\}_{i=0,j=0}^{n-1,m-1}$, are ordered by

$$\mathbf{e}_i^n \otimes \mathbf{e}_j^m \mapsto \mathbf{e}_{mi+j}^{mn}.$$

In this manner an element of $\mathfrak{V}_n \otimes \mathfrak{V}_m$ may be considered to be a vector of length mn with elements drawn from $\mathfrak{F}$. Let $\mathfrak{M}_m^n$ be the space of $n \times m$ matrices over $\mathfrak{F}$. In the following, a linear transformation will be identified with its matrix representation in the standard basis. Let $\mathfrak{M}_n = \mathfrak{M}_n^n$. Let I_n be the $n \times n$ identity transformation of $\mathfrak{V}_n$.

Write $\mathrm{diag}(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{n-1}) \equiv \mathrm{diag}(\mathbf{v})$ for the diagonal matrix in $\mathfrak{M}_n$ whose diagonal elements are taken from the coordinates of $\mathbf{v}$.

If $\mathsf{A} \in \mathfrak{M}_m^n$ and $\mathsf{B} \in \mathfrak{M}_{m'}^{n'}$, the matrix of the tensor product $\mathsf{A} \otimes \mathsf{B} \in \mathfrak{M}_{mm'}^{nn'}$ is given by

$$\mathsf{A} \otimes \mathsf{B} = \begin{pmatrix} \mathsf{A}_{11}\mathsf{B} & \mathsf{A}_{12}\mathsf{B} & \cdots & \mathsf{A}_{1m}\mathsf{B} \\ \mathsf{A}_{21}\mathsf{B} & \mathsf{A}_{22}\mathsf{B} & \cdots & \mathsf{A}_{2m}\mathsf{B} \\ \vdots & \vdots & \cdots & \vdots \\ \mathsf{A}_{n1}\mathsf{B} & \mathsf{A}_{n2}\mathsf{B} & \cdots & \mathsf{A}_{nm}\mathsf{B} \end{pmatrix} \tag{2}$$

The importance of the tensor-product to our work in parallel processing

inheres in the following identity, for $\mathsf{B} \in \mathfrak{M}_m$: [91]

$$
(\mathsf{I}_n \otimes \mathsf{B}) \begin{pmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_{nm-1} \end{pmatrix} = \begin{pmatrix} \mathsf{B} \cdot \begin{pmatrix} \mathbf{v}_0 \\ \vdots \\ \mathbf{v}_{m-1} \end{pmatrix} \\ \mathsf{B} \cdot \begin{pmatrix} \mathbf{v}_m \\ \vdots \\ \mathbf{v}_{2m-1} \end{pmatrix} \\ \vdots \\ \mathsf{B} \cdot \begin{pmatrix} \mathbf{v}_{(n-1)m} \\ \vdots \\ \mathbf{v}_{nm-1} \end{pmatrix} \end{pmatrix} \tag{3}
$$

Suppose $n = ml$. The n-point stride l permutation matrix P^n_l is the $n \times n$ matrix defined by

$$
\mathsf{P}^n_l\,(\mathbf{v} \otimes \mathbf{w}) = \mathbf{w} \otimes \mathbf{v},
$$

where $\mathbf{v} \in \mathfrak{V}_m$ and $\mathbf{w} \in \mathfrak{V}_l$. The effect of P^n_l on a vector is to stride through the vector, taking m steps of size l. For example, taking $m = 3$, $l = 2$, and $n = 6$, we have:

$$
\mathsf{P}^6_2 \begin{pmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \mathbf{v}_4 \\ \mathbf{v}_5 \end{pmatrix} = \begin{pmatrix} \mathbf{v}_0 \\ \mathbf{v}_2 \\ \mathbf{v}_4 \\ \mathbf{v}_1 \\ \mathbf{v}_3 \\ \mathbf{v}_5 \end{pmatrix} \tag{4}
$$

Stride permutations are important due to the following *Commutation Theorem* [170]:

Theorem 1

$$
\mathsf{P}^n_l(\mathsf{A} \otimes \mathsf{B})\mathsf{P}^n_m = \mathsf{B} \otimes \mathsf{A}
$$

where $\mathsf{A} \in \mathfrak{M}_m$, $\mathsf{B} \in \mathfrak{M}_l$, *and* $n = ml$.

This theorem, which is easy to prove even when the entries are from a semiring, allows the order of evaluation in a tensor product to be varied. We shall see in the next subsection that some evaluation orders naturally

correspond to vectorization, and some to parallelizations; the Commutation Theorem will be the method by which one type of execution is traded off for another.

4.2 Code generation: Conversion from factorization to code

This subsection describes the relationship between the matrix representation of a formula and the denoted machine code. Because many of the algorithms to be presented will be presented in the tensorial manner, with the code-generation phase only represented implicitly, this subsection is fundamental to this dissertation.

The matrix notation we use is nothing more than an informal notation for describing algorithms. It differs from standard notations primarily in its explicit denotation of data distribution, communication, and operation scheduling. Whereas most high-level languages, and even special-purpose parallel languages, leave the distribution of data over the processors and the scheduling of operations within processors to the discretion of the compiler, the notation we use, at least potentially, encodes all such scheduling. This has both advantages and disadvantages: although it gives the programmer a finer level of control, which can be important for time-critical applications, it requires some conscious decision-making over data-distribution that is unnecessary in some other languages. On the other hand, the functional nature of the notation does make it potentially amenable to compiler reordering. The most serious disadvantage is its narrowness of application. Originally developed for signal processing codes, this work demonstrates its wider application, but there are many applications which would not easily fall under its rubric.

The target architecture of the language is a machine comprising m parallel processors, each with shared memory. However, it is easy to see that the results go through also, with an extra communication step or two, on local-memory machines. Each processor may also have vector capabilities, so that computations within the processors should be vectorized. We do not assume restrictions on the vector length capability of the processors.

User data is always stored conceptually in the form of a vector

$$\begin{pmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_{n-1} \end{pmatrix}.$$

Assuming that m divides n, elements $\mathbf{v}_0, \ldots, \mathbf{v}_{\frac{n}{m}-1}$ are stored in processor 0, elements $\mathbf{v}_{\frac{n}{m}}, \ldots, \mathbf{v}_{\frac{2n}{m}-1}$ are stored in processor 1, and so on. Matrices are stored in column-major order. It is assumed that certain general classes of specific matrices are already implemented on the architecture, in particular, the stride permutations and any specific permutations corresponding to the interconnection network.

Let $\mathsf{B} \in \mathfrak{M}_l$ and let $\mathbf{code}(\mathsf{B})$ be any sequence of machine instructions that computes the result of left-multiplication by B. That is, $\mathbf{code}(\mathsf{B})$ is a program that takes as input an array $\mathbf{v}$ of l elements of $\mathfrak{F}$, and returns as output the array $\mathsf{B} \cdot \mathbf{v}$ of l elements of $\mathfrak{F}$, where vectors are identified with their coordinates in the standard basis.

Given $\mathbf{code}(\mathsf{B})$ and $\mathbf{code}(\mathsf{B}')$ for two matrices B and B', it is easy to compute some $\mathbf{code}(\mathsf{B} + \mathsf{B}')$. Simply let $\mathbf{code}(\mathsf{B} + \mathsf{B}')$ be the program that, given its input array $\mathbf{v}$, first runs as a subroutine $\mathbf{code}(\mathsf{B})$ on $\mathbf{v}$ (saving the result), then runs $\mathbf{code}(\mathsf{B}')$ on $\mathbf{v}$, and then returns the coordinate-wise sum of the arrays that are returned by these two subroutine calls.

Similarly, given $\mathbf{code}(\mathsf{M})$ and $\mathbf{code}(\mathsf{M}')$, it is easy to find $\mathbf{code}(\mathsf{M} \cdot \mathsf{M}')$, assuming the dimensions of M and M' are compatible: run $\mathbf{code}(\mathsf{M})$ on the result of running $\mathbf{code}(\mathsf{M}')$ on the argument $\mathbf{v}$.

Of course, $\mathbf{code}(\mathsf{I}_l)$ is the code that returns its argument, an l-vector.

Consider a parallel processor with m processors, $p_1, \ldots, p_m$, each with some local memory. We make the convention that a length ml array will be stored with its first l elements in processor p_1, its second l elements in processor p_2, and so on.

Given this convention, one can interpret $\mathbf{code}(\mathsf{I}_m \otimes \mathsf{B})$ as code that runs on this m-processor architecture. To construct $\mathbf{code}(\mathsf{I}_m \otimes \mathsf{B})$, load $\mathbf{code}(\mathsf{B})$ in each p_i. When called on a length ml array $\mathbf{v}$, p_i runs $\mathbf{code}(\mathsf{B})$ on the l elements of $\mathbf{v}$ that are stored in its local memory, and outputs the result to its local memory. Equation 3 shows that this will compute the tensor

product. Similar rules can be derived when the number of processors is different from m.

The code corresponding to $\mathsf{A} \otimes \mathsf{I}_l$, for $\mathsf{A} \in \mathfrak{M}_m$, is a bit more subtle. The interpretation of $\mathbf{code}(\mathsf{A} \otimes \mathsf{I}_l)$ is as the code corresponding to A, except that it operates on l-vectors rather than on scalars. This code can be constructed (loosely speaking) from $\mathbf{code}(\mathsf{A})$ by interpreting the length ml argument array $\mathbf{v}$ as being an element of the m-module over the ring $\mathfrak{F}^l$. This corresponds closely to hardware primitives on certain vector architectures.

The relation

$$\mathsf{A} \otimes \mathsf{B} = (\mathsf{A} \otimes \mathsf{I}_l)(\mathsf{I}_m \otimes \mathsf{B}) \tag{5}$$

can be used to compute general tensor products.

By combining a fixed set of transformations reflecting the hardware primitives of the underlying architecture with combining rules like $+$, $\cdot$ and $\otimes$, and some simple tensor product identities, concise expressions that can be translated into efficient code for certain classes of functions can be defined [73].

4.3 Fast Fourier transforms

This subsection briefly discusses the formulation of the FFT in the tensor product framework presented above [171, pp.16-20]. The presentation is intended to illustrate the parallel code development methodology used to describe parallelization of the chess endgame algorithm, in subsection 6.1 and Table 1. The exposition of the chess material, however, does not depend on any of the results here.

Let F_n be the n-dimensional Fourier transform matrix $(\omega^{ij})_{i,j=0}^{n-1}$, ω a primitive nth root of unity.

Let $n = ml$. The Singleton (1967) [149] mixed-radix version of the Cooley-Tukey (1965) fast Fourier transform [51] can be expressed recursively,

$$\mathsf{F}_n = (\mathsf{F}_m \otimes \mathsf{I}_l)\mathsf{T}_l(\mathsf{I}_m \otimes \mathsf{F}_l)\mathsf{P}^n_m. \tag{6}$$

where T_l is a diagonal matrix encoding the twiddle factors:

$$\mathsf{T}_l = \bigoplus_{j=0}^{m-1} \left(\operatorname{diag}\left(1, \omega, \ldots, \omega^{l-1}\right)\right)^j.$$

This can be interpreted as a mixed parallel/vector algorithm. Given an input vector $\mathbf{v}$, $\mathsf{P}^n_m \mathbf{v}$ forms a list of m segments, each of length l. The $\mathsf{I}_m \otimes \mathsf{F}_l$ term performs m l-point FFTs in parallel on each segment. T_l just multiplies each element by a twiddle factor. Finally, the $\mathsf{F}_m \otimes \mathsf{I}_l$ term performs an m-point FFT on vectors of size l.

The commutation theorem can be used to derive a parallel form

$$\mathsf{F}_n \quad = \quad \mathsf{P}^n_m \left(\mathsf{I}_l \otimes \mathsf{F}_m\right) \mathsf{P}^n_l \mathsf{T}_l \left(\mathsf{I}_m \otimes \mathsf{F}_l\right) \mathsf{P}^n_m, \tag{7}$$

and a vector form

$$\mathsf{F}_n \quad = \quad \left(\mathsf{F}_m \otimes \mathsf{I}_l\right) \mathsf{T}_l \mathsf{P}^n_m \left(\mathsf{F}_l \otimes \mathsf{I}_m\right). \tag{8}$$

The parallel Pease (1968) FFT can be derived by unrolling the recursion in Equation 7, and the vectorized Korn-Lambiotte FFT (1979) can be derived by unrolling Equation 8 [99, 128].

By using the commutation theorem and varying the factorization, many different FFT algorithms have been derived, with different tradeoffs between parallelization and vectorization [26, 28, 47, 72, 90].

5 Application to chess

This section describes the chess endgame algorithm in a generalization of the tensor product formalism described in Section 4.

Let us imagine, for the moment, that a matrix over $\mathfrak{GF}_2$ is actually a Boolean matrix whose entries are taken from the Boolean algebra $\{0, 1, \vee, \wedge\}$. We write $+$ and $\cdot$ for $\vee$ and $\wedge$ respectively. The notion of linear transformations then changes, as does, therefore, $\mathfrak{M}^n_m$, in the natural way.

This generalization has been used for expressing graph algorithms [1, 30, 101, 164, 165]. The definitions of $\otimes$, matrix product, and matrix sum remain essentially unchanged.

In particular, the commutation theorem, the notion of **code**(M), and the relation between $\otimes$ and parallelization still holds.

These ideas could, of course, be presented categorically using the approach of Skillicorn and Bird-Meerstens [42, 150], or using the mathematics-of-arrays formalism of Mullin [121].

5.1 Definitions

For the sake of simplicity of exposition, captures, pawns, stalemates, castling, and the 50-move rule will be disregarded unless otherwise stated.

Let S be an ordered set of k chess pieces. For example, if $k = 6$ then one could choose $S = \langle$♚, ♔, ♕, ♖, ♛, ♜$\rangle$.

An *S-position* is a chess position that contains exactly the k pieces in S. We write $S = \langle S_1, S_2, \ldots, S_k \rangle$. An S-position can be viewed as an assignment of each piece $S_i \in S$ to a *distinct* square of the chessboard (note that captures are not allowed).

$\mathfrak{V}_n$ is the space of length n Boolean vectors. The space of 8×8 Boolean matrices is thus $\mathfrak{C} \equiv \mathfrak{V}_8 \otimes \mathfrak{V}_8$. Let $\{\mathbf{e}_i\}_{i=1}^{8}$ be the standard basis for $\mathfrak{V}_8$.

Let $\bigotimes^j \mathfrak{V}$ be the jth tensor power of $\mathfrak{V}$, *i.e.,* $\mathfrak{V} \otimes \cdots \otimes \mathfrak{V}$, with j factors.

Let $\mathfrak{B} \equiv \bigotimes^k \mathfrak{C}$. $\mathfrak{B}$ is called the *hyperboard* corresponding to S. It can be thought of as a cube of side-length 8 in $\mathbb{R}^{2k}$. Each of the 64^k basis elements corresponds to a point with integer coordinates between 1 and 8.

Each basis element of $\mathfrak{C}$ is of the form $\mathbf{e}_i \otimes \mathbf{e}_j$ for $1 \leq i, j \leq 8$. Any such basis element, therefore, denotes a unique square on the 8×8 chessboard. Any element of $\mathfrak{C}$ is a sum of distinct basis elements, and therefore corresponds to a set of squares [185].

Each basis element of $\mathfrak{B}$ is of the form $\mathbf{c}_1 \otimes \mathbf{c}_2 \otimes \cdots \otimes \mathbf{c}_k$, where each $\mathbf{c}_s$ is some basis element of $\mathfrak{C}$. Since each $\mathbf{c}_s$ is a square on the chessboard, each basis element of $\mathfrak{B}$ can be thought of as a sequence of k squares of the chessboard. Each position that is formed from the pieces of S is thereby associated with a unique basis element of $\mathfrak{B}$. Any set of positions, each of which is formed from pieces of S, is associated with a unique element of $\mathfrak{B}$: the sum of the basis elements corresponding to each of the positions from the set.

This correspondence between sets of chess positions and elements of $\mathfrak{B}$ forms the link between the chess algorithms and the tensor product formulation. In the following, the distinction between sets of chess positions formed from the pieces in S and elements of the hyperboard $\mathfrak{B}$ will be omitted when the context makes the meaning clear.

If $p \in \{$♔, ♕, ♖, ♗, ♘$\}$ is a piece, then the *unmove operator* $\text{Ш}_{\mathbf{p},s}$ is the

function that, given an S-position P returns the set of S-positions that could be formed by unmoving S_s in P as if S_s were a p.

$\text{Ш}_{\mathbf{p},s}$ can be extended to a linear[10] function from elements of $\mathfrak{B}$ to itself, and thereby becomes an element of $\mathfrak{M}_{64^k}$.

The core of the chess endgame algorithm is the efficient computation of the $\text{Ш}_{\mathbf{p},s}$. The following subsections describe a factorization of $\text{Ш}_{\mathbf{p},s}$ in terms of primitive operators. The ideas of subsection 4.2 may then be used to derive efficient parallel code from this factorization.

5.2 Group actions

This subsection introduces a few group actions [67]. We will use the group-theoretic terminology both to give concise descriptions of certain move operators and to describe the exploitation of symmetry. There is a close correspondence between multilinear algebra, combinatorial enumeration, and group actions which motivates much of this section [108–111].

The symmetric group on k elements $\mathfrak{S}_k$ acts on $\mathfrak{B}$ by permuting the order of the factors:

$$\mathfrak{s} \bigotimes_{s=1}^{k} \mathbf{c}_s = \bigotimes_{s=1}^{k} \mathbf{c}_{\mathfrak{s}s},$$

for $\mathfrak{s} \in \mathfrak{S}_k$ and $\mathbf{c}_s \in \mathfrak{C}$.

The dihedral group of order 8, $\mathfrak{D}_4$, is the group of symmetries of the square. It is generated by two elements $\mathfrak{r}$ and $\mathfrak{f}$ with relations $\mathfrak{r}^4 = \mathfrak{f}^2 = \mathfrak{e}$ and $\mathfrak{r}^3\mathfrak{f} = \mathfrak{f}\mathfrak{r}$. It acts on $\mathfrak{C}$ by

$$\mathfrak{r}(e_i \otimes e_j) = e_{8-j+1} \otimes e_i \tag{9}$$

$$\mathfrak{f}(e_i \otimes e_j) = e_i \otimes e_{8-j+1} \tag{10}$$

Thus, $\mathfrak{r}$ rotates the chessboard counterclockwise 90° and $\mathfrak{f}$ flips the chessboard about the horizontal bisector.

$\mathfrak{D}_4$ acts diagonally on $\mathfrak{B}$:

$$\mathfrak{d} \bigotimes_{s=1}^{k} c_s = \bigotimes_{s=1}^{k} \mathfrak{d} c_s$$

[10] Technically the unmove operators are only quasilinear, since the Boolean algebra is not a ring, and thus $\mathfrak{B}$ is not a module.

Let $\mathfrak{C}_4$ be the cyclic group generated by $\mathfrak{r}$.

A group $\mathfrak{G}$ acting on $\mathfrak{V}_n$ and $\mathfrak{V}_m$ acts on $\mathfrak{M}_n^m$ by conjugation: $(\mathfrak{g}M)\mathbf{v} = \mathfrak{g}(M\mathfrak{g}^{-1}(\mathbf{v}))$. We let

$$\int_{\mathfrak{G}} x = \sum_{\mathfrak{g}\in\mathfrak{G}} \mathfrak{g}x.$$

The notation $\int_{\mathfrak{G}} x$ is intended to represent the group average of x with respect to $\mathfrak{G}$ [67, p. 6]. It is a fixed point of the $\mathfrak{G}$ action: $\mathfrak{g}\int_{\mathfrak{G}} x = \int_{\mathfrak{G}} x$ for all $\mathfrak{g} \in \mathfrak{G}$.

6 Endgame algorithm

This section presents the endgame algorithm using the notation developed in Section 5. Subsection 6.1 gives the fundamental factorization. Subsection 6.2 describes the modification of the equations of Table 1 to exploit symmetry. Subsection 6.3 describes the control structure of the algorithm.

6.1 Factoring the unmove operator

Recall from Equation 1 that E_8 was defined to be the unit one-dimensional 8×8 end-off shift matrix. The unit multidimensional shift along dimension s is defined by

$$\mathsf{U}_s \in \mathfrak{M}_{64^k} \equiv \mathsf{I}_{64^{s-1}} \otimes (\mathsf{E}_8 \otimes \mathsf{I}_8) \otimes \mathsf{I}_{64^{k-s}}.$$

Such multidimensional shifts are commonly used in scientific computation.

Fix a basis $\{\mathbf{c}_i\}_{i=1}^{64}$ of $\mathfrak{C}$, and define

$$\mathsf{L} \in \mathfrak{M}_{64^k} \equiv \operatorname{diag}\left(\int_{\mathfrak{S}_k} \sum_{i_1<\cdots<i_k} \mathbf{c}_{i_1} \otimes \cdots \otimes \mathbf{c}_{i_k}\right) \tag{11}$$

Certain basis elements of $\mathfrak{B}$ do not correspond to legal S-positions. These "holes" are elements of the form $\bigotimes_{s=1}^{k} \mathbf{c}_s$ such that there exist distinct s, s' for which $\mathbf{c}_s = \mathbf{c}_{s'}$. If $\mathbf{v} \in \mathfrak{B}$ then $\mathsf{L}\mathbf{v}$ is the projection of $\mathbf{v}$ onto the subspace of $\mathfrak{B}$ generated by basis elements that are not holes.

Table 1 defines the piece-unmove operators.

$$
\begin{aligned}
\mathrm{III}_{\text{♖},s} &= \int_{\mathfrak{C}_4} \mathsf{LU}_s(\mathsf{I}_{64^k} + \mathsf{LU}_s)^6 && (12)\\
\mathrm{III}_{\text{♘},s} &= \mathsf{L}\int_{\mathfrak{D}_4} \mathsf{U}_s \cdot (\mathfrak{r}(\mathsf{U}_s^2)) && (13)\\
\mathrm{III}_{\text{♗},s} &= \int_{\mathfrak{D}_4} \mathsf{LU}_s(\mathsf{I}_{64^k} + \mathsf{LU}_s\mathfrak{r}\mathsf{U}_s)^6 && (14)\\
\mathrm{III}_{\text{♔},s} &= \mathsf{L}\int_{\mathfrak{C}_4} \mathsf{U}_s + \mathsf{U}_s\mathfrak{r}\mathsf{U}_s && (15)\\
\mathrm{III}_{\text{♕},s} &= \mathrm{III}_{\text{♖},s} + \mathrm{III}_{\text{♗},s} && (16)\\
& && (17)
\end{aligned}
$$

Table 1: These equations define the core of a portable endgame algorithm. By modifying the factorizations, code suitable for execution on a wide range of high-performance architectures can be derived.

Figure 2 illustrates the computation of the integrand in the expression for $\mathrm{III}_{\text{♖},1}$ in Table 1. This corresponds to moving the ♖ to the right. The average over $\mathfrak{C}_4$ means that the ♖ must be moved in 4 directions. For example, conjugation by $\mathfrak{r}$ of the operation of moving the ♖ right corresponds to moving the ♖ up: if one rotates the chessboard clockwise 90°, moves the ♖ right, and then rotates the chessboard counterclockwise 90°, the result will be the same as if the ♖ had been moved up to begin with.

As in the case of fast Fourier transforms (see Equations 7 and 8), by varying the factorization, code suitable for varying architectures can be derived. For example, if the interconnection architecture is a 2-dimensional grid, then only U_s for $s = 1$ can be directly computed. By using the relations $\mathsf{U}_s = (1\ s)\mathsf{U}_1$ and $\mathrm{III}_{\mathbf{p},s} = (1\ s)\mathrm{III}_{\mathbf{p},1}$, equations appropriate for a grid architecture can be derived. Here $(1\ s) \in \mathfrak{S}_k$ interchanges 1 and s.

These equations are vectorizable as well [151]. The vectorized implementation of Table 1 by Burton Wendroff et al. has supported this claim [183].

Other factorizations appropriate for combined vector and parallel architectures, such as a parallel network of vector processors, can also be derived [94].

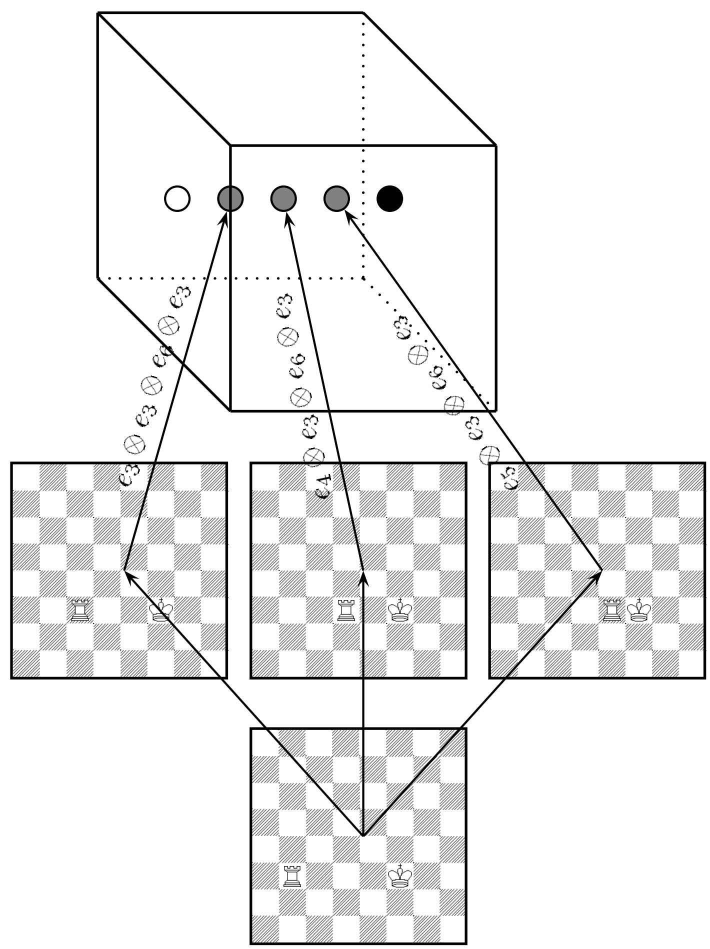


Figure 2: Unmoving the ♖ to the right from the position at bottom results in the three positions center. Here, $S = \langle$♖, ♔$\rangle$. Each position corresponds to a point in the hyperboard, top. The bottom position is $\mathbf{e}_2 \otimes \mathbf{e}_3 \otimes \mathbf{e}_6 \otimes \mathbf{e}_3$. The new positions are $\mathbf{e}_3 \otimes \mathbf{e}_3 \otimes \mathbf{e}_6 \otimes \mathbf{e}_3$, $\mathbf{e}_4 \otimes \mathbf{e}_3 \otimes \mathbf{e}_6 \otimes \mathbf{e}_3$, and $\mathbf{e}_5 \otimes \mathbf{e}_3 \otimes \mathbf{e}_6 \otimes \mathbf{e}_3$. $\mathbf{e}_6 \otimes \mathbf{e}_3 \otimes \mathbf{e}_6 \otimes \mathbf{e}_3$ is illegal and is zeroed out by L.

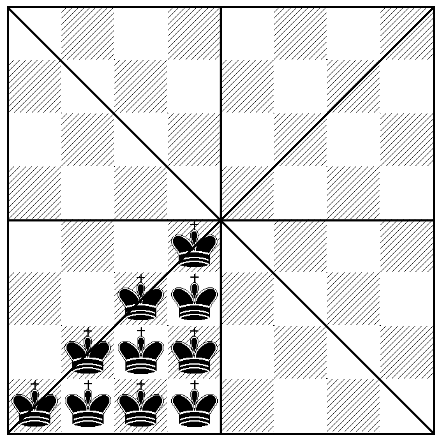


Figure 3: The chessboard may be rotated 90° or reflected about any of its bisectors without altering the value of a position without pawn. Therefore, the location of the ♚ may be restricted to the 10 squares shown, called a fundamental region.

6.2 Exploiting symmetry

The game-theoretic value of a chess position without pawns is invariant under rotation and reflection of the chessboard. Therefore, the class of positions considered can be restricted to those in which the ♚ is in the lower left-hand octant, or fundamental region, of the chessboard (Figure 3).

The chess positions with the ♚ in its fundamental region correspond to points in a triangular wedge in the hyperboard.

Algebraically, because each $\text{Ш}_{\mathbf{p},s}$ is a fixed point of the $\mathfrak{D}_4$ action, we need only consider the $10 \cdot 64^{k-1}$-space:

$$\mathfrak{B}' \equiv \mathfrak{C}/\mathfrak{D}_4 \otimes \bigotimes^{k-1} \mathfrak{C}$$

rather than the 64^k-space $\mathfrak{B}$. We suppose that the first piece of S, the piece corresponding to the first factor in the expression for $\mathfrak{B}'$, is the ♚.

When pieces other than the ♚ are moved, the induced motion in the hyperboard remains within the wedge. Thus, the induced functions $\text{Ш}'_{\mathbf{p},s}\colon \mathfrak{B}' \mapsto \mathfrak{B}'$

have the same form as Table 1 when $s \geq 1$.

However, when the ♚ is moved outside its fundamental region, the resulting position must be transformed so that the ♚ is in its fundamental region. This transformation of the chessboard induces a transformation on the hyperboard (Figure 4).

Algebraically,

$$\text{Ш}'_{\text{♚},1} = \sum_{\mathfrak{d}\in\mathfrak{D}_4} \text{Ш}'_{\text{♚},1_{\mathfrak{d}}} \otimes \bigotimes^{k-1} \mathfrak{d} \tag{18}$$

where $\text{Ш}'_{\text{♚},1_{\mathfrak{d}}} \in \mathfrak{M}_{10}$.

The sum over $\mathfrak{d} \in \mathfrak{D}_4$ corresponds to routing along the pattern of the Cayley graph of $\mathfrak{D}_4$ (see Figure 5).

This is a graph whose elements are the 8 transformations in $\mathfrak{D}_4$, and whose edges are labeled by one of the generators $\mathfrak{r}$ or $\mathfrak{f}$. An edge labeled $\mathfrak{h}$ connects node $\mathfrak{g}$ to node $\mathfrak{g}'$ if $\mathfrak{h}\mathfrak{g} = \mathfrak{g}'$. The communication complexity of the routing can be reduced by exploiting the Cayley graph structure [154]. The actual communication pattern used is that of a group action graph, which looks like a number of disjoint copies of the Cayley graph, together with some cycles [184].

The problem of parallel application of a structured matrix to a data set invariant under a permutation group has been studied in the context of finite-element methods by Danny Hillis and Washington Taylor as well. Although their terminology is different from our terminology, their general ideas are similar [80]. The method we use turns out to be similar to the orbital exchange method, which is used to compute the FFT of a data set invariant under a crystallographic group [23, 24, 169].

It is interesting to note that exploiting symmetry under interchange of identical pieces can be handled in this notation: j identical pieces correspond to a factor $\mathrm{Sym}^j\mathfrak{C}$ in the expression for $\mathfrak{C}$, where Sym^j is the jth symmetric power of $\mathfrak{C}$. [67, pp.472–475]

There are efficient algorithms, in general, for performing the purely algebraic operations required, as well as languages, such as GAP, MAGMA, and AXIOM, that are suitable for the denotation of the algebraic structures used [44, 147, 148]. The groups encountered here are so small, however, that computer-assisted group-theoretic computation was not required.

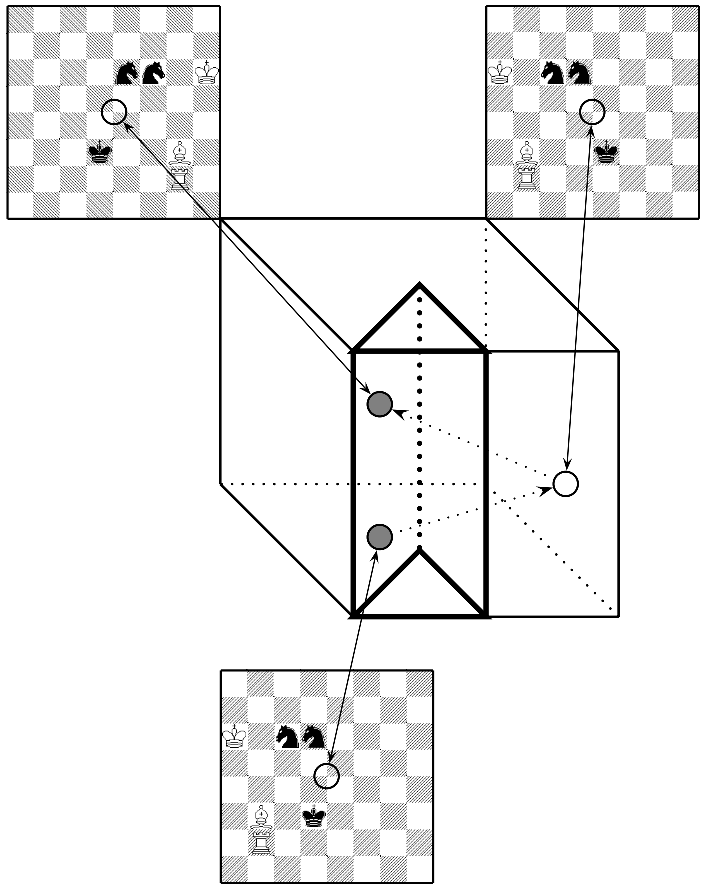


Figure 4: Only a wedge in the hyperboard is physically stored. To compute the effect of moving the ♚ outside the squares to which it is restricted, a communication pattern is induced in the hyperboard. In this example, the ♚ in the lower chessboard is moved, reaching the position at the upper right. This position must be reflected about the vertical bisector, yielding the position at upper left. These three positions correspond to three points in the hyperboard, only the first and third of which are physically stored. The Black-to-move position at bottom requires 222 moves against best play for White to win (see Table 2).

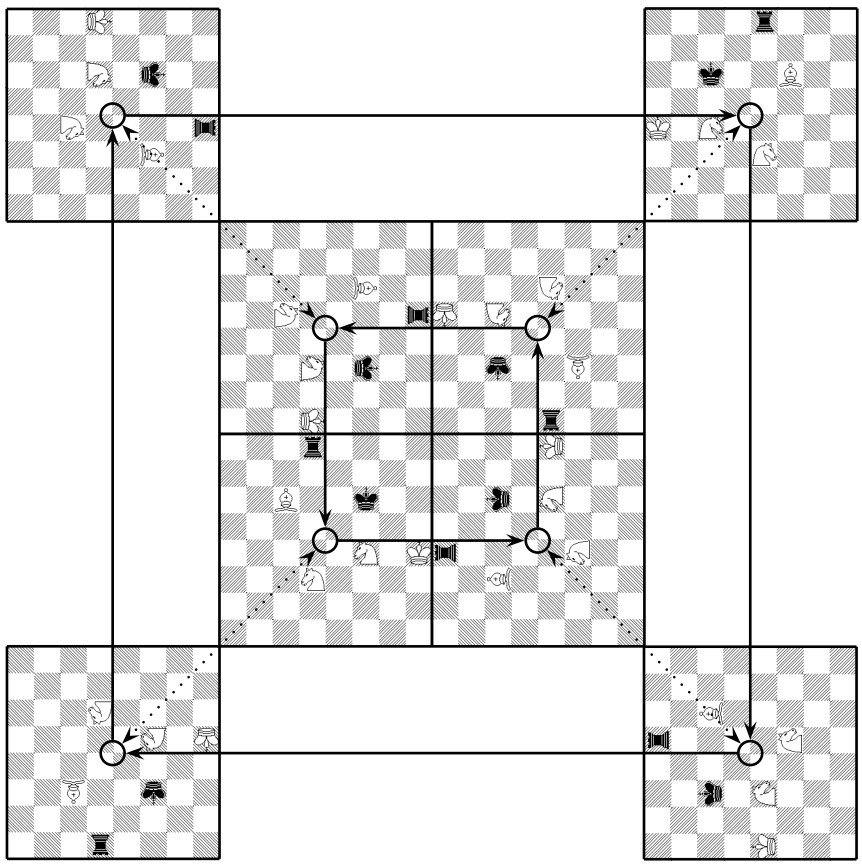


Figure 5: The Cayley graph for $\mathfrak{D}_4$. Each node is pictured by showing the effect of its corresponding transformation on a position in ♔♗♘♘♔♖; thus, the chess value of each of these nodes is the same. Solid lines correspond to $\mathfrak{r}$, and rotate the board counterclockwise 90°. Dotted lines correspond to $\mathfrak{f}$, and flip the board horizontally. The position shown arose during a game between Anatoly Karpov and Gary Kasparov in Tilburg, October 1991.

6.3 Control structure

For $i \geq 1$ we define $\mathbf{v}_i \in \mathfrak{B}$ to be the vector of positions from which White to move can checkmate Black within i moves (i.e., i White moves and $i-1$ Black moves). Thus, $\mathbf{v}_1$ is the vector of positions from which White can checkmate Black on the next move. $\mathbf{v}_2$ is the set of positions from which White can either checkmate Black in one move or can move to a position from which any Black reply allows a mate-in-one, and so on.

The overall structure of the algorithm is to iteratively compute the sets $\mathbf{v}_1, \mathbf{v}_2, \ldots$ until some i is reached for which $\mathbf{v}_i = \mathbf{v}_{i+1}$. Then $\mathbf{v} = \mathbf{v}_i$ is the set of positions from which White can win, and i is the *maximin* value of the set S: the maximum, over all positions from which White can win, of the number of moves required to win that position [161, 167].

The method for computing $\mathbf{v}_i$ from $\mathbf{v}_{i-1}$ is called the backup rule. Several backup rules have been used in various domains [100, 143]. They are all characterized by the use of an *unmove generator* to "unmove" pieces, or move them backward, possibly in conjunction with more traditional move generators. We let

$$\text{Ш}_{\text{White}} \equiv \sum_{\{s: S_s \text{ is White}\}} \text{Ш}_{S_s, s} \tag{19}$$

$$\text{Ш}_{\text{Black}} \equiv \sum_{\{s: S_s \text{ is Black}\}} \text{Ш}_{S_s, s} \tag{20}$$

The backup rule used is:

$$\mathbf{v}_{i+1} = \text{Ш}_{\text{White}}(\overline{\text{Ш}_{\text{Black}}(\overline{\mathbf{v}_i})}). \tag{21}$$

Here, $\overline{\mathbf{v}}$ denotes the complement of $\mathbf{v}$.

7 Implementation notes

7.1 Captures and pawns

The algorithms developed so far must be modified to account for captures and pawns.

Each subset of the original set of pieces S induces a *subgame*, and each subgame has its own hyperboard [37]. Without captures, moving and unmoving are the same, but when captures are considered they are slightly different. The equations for $\text{Ш}_{\mathbf{p},s}$ developed in the preceding section refer to *unmoving* pieces, not to moving them [167]. Unmoving pieces cannot capture, but they can uncapture, leaving a piece in their wake. This is simulated via interhyperboard communication.

The uncapture operation can be computed by using outer products, corresponding to the parallel broadcast, or `SPREAD` primitive [2, 156]. An uncapture is the product from left to right of an unmove operator in the parent game, a diagonal matrix, a sequence of stride matrices, and a broadcast. The broadcast is a tensor product of copies of an identity matrix with the 1×64 matrix of 1's.

Each pawn position induces a separate hyperboard. Pawn unpromotion induces communication between a quotient hyperboard and a full hyperboard, which is implemented again by multiplication by $\mathfrak{D}_4$.

7.2 Database

There are two values that can be associated with a position: *distance-to-mate* and *distance-to-win*.

The distance-to-mate is the number of moves by White required for White to checkmate Black, when White plays so as to checkmate Black as soon as possible, and Black tries to avoid checkmate as long as possible [187]. Although the distance-to-mate might seem like the natural metric to use, it can produce misleadingly high distance values because the number of moves to mate in trivial subgames, like ♔♖♔, would be included in the count of something like ♔♖♔♘. In fact, in ♔♖♔♘, it does not matter for most purposes how many moves are required to win the subgame ♔♖♔, once White captures the ♘, as long as the ♘ is captured safely [133].

The more usual distance-to-win metric is simply the number of moves required by White to force conversion into a winning subgame. In practice, this metric is more useful when the position has no pawns. It also is the metric of relevance to the 50-move rule. If a particular position has a distance-to-win of m,, then against perfect play the win value would be altered by an m' move rule for $m' > m$. Although our program has implemented distance-to-

mate metric for 5-piece endgames, the results presented here use the more conservative distance-to-win metric.

The *max-to-win* for a set of pieces (i.e., an endgame) is the maximum, over all positions using those pieces from which White can win, of the distance-to-win of that position.

The distance-to-win of each point in the hyperboard can be stored so that a 2-ply search permits optimum play.

By Gray coding this distance, the increment of the value can be done by modifying only one bit.

Curiously, the motif of embedding Cayley graphs into Cayley graphs arises several times in this work. Gray codes, which can be viewed as embedding the Cayley graph for $\mathbb{Z}_{2^n}$ into that of $\mathbb{Z}_2^n$, are used both for implementing U (and, therefore, $\text{Ш}_{\mathbf{p},s}$) and for maintaining the database. Embedding the Cayley graph for $\mathfrak{D}_4$ in that of $\mathbb{Z}_2^n$ arises during unpromotion and moving the ♔. Because many interconnection networks are Cayley graphs or group action graphs [25, 59, 60, 136], this motif will reappear on other implementations.

8 Results

8.1 Chess results

The combinatorially possible pawnless 5-piece games and many 5-piece games with a single pawn were solved using an early version of the current program. This work resulted in the first publication of the 77-move ♔♗♘♔♘ max-to-win, which at the time was the longest known pawnless max-to-win [153]. Some endgames were solved under the distance-to-mate metric as well. The distance-to-mate results were not particularly illuminating. The state space size is approximately $121 \cdot 10^6$ nodes for a single pawnless 5-piece endgame under the Thompson symmetries, in which one representative from the 462 orbits of the $\mathfrak{D}_4$ action on the nonadjacent positions of the two kings is stored.

Several pawnless 6-piece endgames were also solved. The state-space size per endgame was 6,185,385,360 nodes, although the size of each hyperboard is $462 \cdot 64^4$, or about $7.7 \cdot 10^9$.

Table 2 presents statistical information about some pawnless 6-piece endgames.

The percent-win can be misleading because of the advantage of the first move in a random position—White can often capture a piece in one move—and because it includes positions in which Black is in check.

The max-to-win values were significantly higher than previously known endgames. No 5-piece endgame had a max-to-win over 100, and most of the nontrivial ones had max-to-wins of approximately 50. ♔♖♘♔♘♘ has the longest known max-to-win of 243, although it is not a general win.

We remark that ♔♖♗♔♘♘ is a general win, with 223 moves required to win in the worst case. ♔♖♗♔♘♘ was called "known to be a draw" by Roycroft, a leading endgame expert, in 1972 (♔♗♗♔♘, which was considered a draw by most players, was only "controversial or unknown" according to the same source). Most of the standard works concurred with the opinion that ♔♖♗♔♘♘ was not a general win [63, pp.50–53], [49, p. 417], [38, pp.167–169], [65, p. 521]. Chéron, however, seems to reserve judgment.

The 50-move rule would affect the value of each endgame listed with max-to-win of 50 or more. The 92-move win in ♔♕♖♔♕♖ is somewhat surprising too.

A mutual zugzwang is closely related to a game whose Conway value is 0: it is a position in which White to move can only draw, but Black to move loses. Such positions seem amusing because, particularly when no pawns are involved, chess is a very "hot" game in the sense of *Winning Ways* [40].

Unlike the "maximin" positions (see appendix) whose analysis is fairly impenetrable, the mutual zugzwangs can sometimes be understood by humans.

An example may help clarify this concept. Figure 6 shows a mutual zugzwang discovered by the program, in ♔♗♘♘♔♖. The Black ♖ is trapped on **h8**,[11] since **g8** is guarded by the ♗ on **a2**, and the ♘'s guard each other. If the Black ♖ were to capture a ♘, then it would in turn be captured, and the resulting subgame of ♔♗♘♔ would be winning for White. The position seems to be a race between Kings to see who will reach

[11] The columns of a chessboard are conventionally lettered from left to right with letters going from **a** to **h**; the rows of the chessboard are numbered from **1** to **8** reading up the page. Thus, **h8** is the square on the upper right corner of the board.

Game	W	Wins	%W	Z	Game	W	Wins	%W	Z
♔♖♘♚♞♞	243	4821592102	78	18176	♔♕♚♝♝♞	63	5257968414	85	6670
♔♖♗♚♞♞	223	5948237948	96	456	♔♖♖♚♜♝	54	4529409548	73	1030
♔♖♘♚♝♞	190	4433968114	72	8030	♔♖♘♚♝♝⋆	52	1015903231	65	256
♔♕♘♚♜♜	153	5338803302	86	1858	♔♕♚♝♝♝	51	5058432960	82	2820
♔♖♘♚♝♝	140	4734636964	77	1634	♔♗♘♘♚♜	49	3302327120	53	1270
♔♖♖♘♚♛	101	5843483696	94	1520	♔♕♕♚♛♜	48	5689213742	92	32
♔♖♗♘♚♛	99	4242312073	69	1010	♔♕♗♚♛♝	46	4079610404	66	22
♔♖♗♚♝♞	98	5359504406	87	1478	♔♕♕♚♛♛	44	5122186896	83	32
♔♘♘♘♚♝	92	5324294232	86	6300	♔♖♗♗♚♛⋆	44	1185941301	75	396
♔♕♖♚♛♜	92	5125056553	83	243	♔♗♗♚♞♞⋆	38	981954704	63	1662
♔♘♘♘♚♞	86	5834381682	94	12918	♔♖♖♚♝♝⋆	37	1483910528	94	26
♔♕♗♚♜♜	85	5707052904	92	342	♔♕♗♚♛♞	36	4213729734	68	78
♔♖♖♗♚♛	82	5935067734	96	388	♔♕♚♞♞♞	35	4626525594	75	17688
♔♖♗♚♝♝⋆	75	1123471668	72	95	♔♕♗♚♛♜	32	3825698576	62	6
♔♕♖♚♛♝	73	5365200098	87	1410	♔♕♘♚♛♝	32	3789897570	61	35
♔♖♖♚♜♞	73	5023789102	81	1410	♔♘♗♗♚♞	31	6130532196	99	58
♔♕♘♘♚♛	72	5808880660	94	2228	♔♕♘♚♛♞	29	3920922433	63	152
♔♕♖♚♛♞	71	5553239408	90	1780	♔♕♘♚♛♜	27	3533291870	57	3
♔♗♗♗♚♜	69	4944693522	80	48	♔♖♖♚♜♜	18	4136045492	67	16
♔♗♗♘♚♜⋆	68	1497242834	95	83	♔♗♗♘♚♛⋆	12	970557572	62	18
♔♖♖♖♚♛	65	6054654948	98	6					

Table 2: Endgame description, maximin, number of wins, percent-win, and number of mutual zugzwangs for certain 6-piece endgames. The symmetries considered are the Thompson symmetries in which the one representative from each of the 462 orbits of the non-adjacent king-positions is stored, except for endgames with a ⋆, which indicates that the two bishops were constrained to lie on squares of opposite colors. A state-space size of 6,185,385,360 for normal endgames and 1,570,867,920 for endgames with a ⋆ was used. Thus, for example, there is really only a single mutual zugzwang in ♔♖♖♖♚♛, but it is counted 6 times.

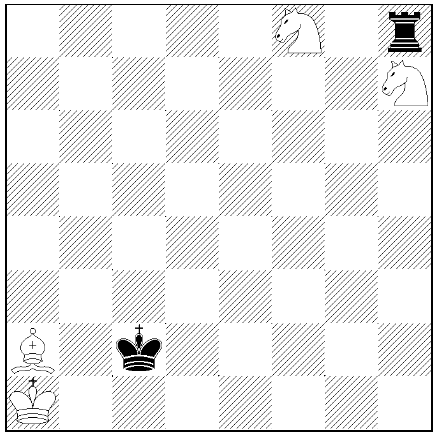


Figure 6: Mutual zugzwang: White to play draws, but Black to play loses

the upper right corner area first. If the Black ♔ reaches **g7** or **e8** first, the Black ♖ can sacrifice itself for a White ♘, and then the Black ♔ captures the other White ♘, leaving the drawn endgame ♔♗♔. On the other hand, if the White ♔ reaches **g7** first, it simply captures the Black ♖**h8**. Note also that neither ♘ can move, as the ♖ would immediately capture the other ♘.

It is not difficult to see that Black to play loses: White gets in first. For example, **1** ...♔**c3 2** ♔**b1** ♔**d4 3** ♔**c2** ♔**c5** (If 3...♔e5? 4 ♘g6+ wins the ♖) **4** ♔**d3** ♔**d6 5** ♔**e4** ♔**c7** (If ♔e7? 6 ♘g6+ wins) **6** ♔**f5** ♔**d8 7** ♔**g6** ♔**e8 8** ♔**g7** and White wins.

However, White to move from the position in Figure 6 must move the ♗. **1** ♗**b1+** ♔**c3** forces **2** ♗**a2** ♔**c2**, since other moves by White on the second move allow the ♖**h8** to escape via **g7**. Chess theory, confirmed by the program, shows that this general position in ♔♗♘♘♔♖ is drawn. Any other move of the ♗ on move **1** allows Black to win the race. For example, **1** ♗**f7** ♔**c3 2** ♔**b1** ♔**d4 3** ♔**c2** ♔**c5 4** ♔**d3** ♔**d6 5** ♔**e4** ♔**e7**! draws.

Figure 8 shows an endgame composed by Elkies based on the computer-discovered mutual zugzwang of Figure 7 [61] [141, Number 546]. Although non-chessplayers may have difficulty understanding the analysis of his position, it follows accepted aesthetic practice in the art of endgame composition by avoiding the use of promoted pieces in the original position and by striv-

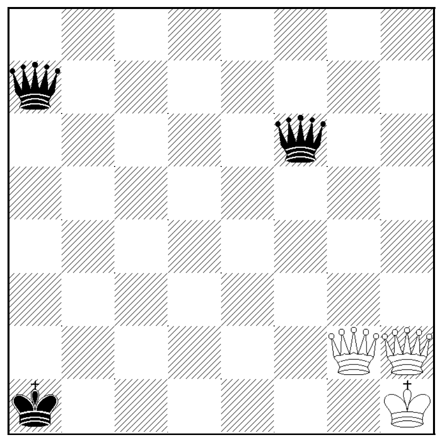


Figure 7: Mutual Zugzwang. If Black moves the ♕**a7** then ♕**hg1** or ♕**a2** mates. If the ♕**f6** moves then ♕**b2** or ♕**f1** will mate. If ♔**b1** then ♕**c2** mates. Thus, any Black move loses. On the other hand, if White moves first then Black can force the draw.

ing for a natural appearance.

The program was used to analyze a game between Anatoly Karpov and Gary Kasparov that occurred during an elite tournament in Tilburg. The players reached the position shown in Figure 5. After playing on for 50 moves a draw was reached, but an exhaustive analysis by the 6-piece program was necessary to prove that a win was not missed [155], since it was unclear whether a draw could have been obtained. In fact, however, pawnless 6-piece endgames almost never arise in tournament play.

8.2 Timing

The implementation was on a 64K processor CM-2/200 with 8 GBytes RAM. The processors were interconnected in a hypercube and clocked at 7MHz (10 MHz for the CM-200). The CM-2 6-piece code required approximately 1200 seconds for initialization and between 111 and 172 seconds to compute K_{i+1} from K_i. Exact timings depend on S (for instance, as is clear from Table 1, $\text{Ш}_{♕,s}$ is slower than either $\text{Ш}_{♖,s}$ or $\text{Ш}_{♗,s}$) as well as run-time settable

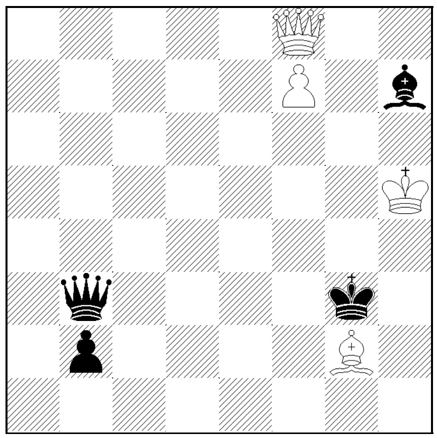


Figure 8: Elkies, *American Chess Journal* **1**(2) 1994. White to play and win. “**1 ♕g7+** Not 1 ♕d6+? ♔xg2 2 f8/♕ (interpolating further checks does not help) when 2…♕h3+ 3 ♔g5 ♕e3+ forces either perpetual check or a queen trade, drawing. **1…♔h2 2 f8/♕**. If 2 ♕e5+ ♔xg2 3 f8/♕ ♕h3+ 4 ♔g5 b1/♕ with ♔h1 and ♗e4 draws, but now 2…b1/♕ loses to 3 ♕f4+ ♔g1 4 ♗e4+ and mate. Thus, Black tries for perpetual check, and not with 2…♕d1+? 3 ♗f3. **2…♕b5+ 3 ♔h6 ♕b6+ 4 ♗c6!** Not yet 4 ♔xh7 b1/♕+ 5 ♔h8 ♕b8! drawing. Now Black must take the bishop because 4 …♕e3+ 5 ♕g5 ♕xg5+ 6 ♔xg5 b1/♕ 7 ♕f2+ mates. **4…♕xc6+ 5 ♔xh7 b1/♕+**. So Black does manage to give the first check in the four-queen endgame, but he is still in mortal danger. **6♔h8 ♔h1!** Black not only cannot continue checking, but must play this modest move to avoid being himself checked to death! For instance, 6 …♕g2 7 ♕c7+ ♔g1 8 ♕fc5+ ♔h1 9♕h5+ and the Black king soon perishes from exposure. But against the quiet **6…♔h1** White wins only with **7 ♕fg8!!**, a second quiet move in this most tactical of endgames, bringing about” the rotated version of the ♔♕♕♔♕♕ mutual zugzwang. (Quotation from Elkies’ analysis)

factorization choices and load on the front end.

Per-node time per endgame (time to solve the endgame divided by number of nodes in the state-space) is faster by a factor of approximately $6 \cdot 10^3$ than timings *of different endgames* reported using classical techniques [124, 167, 178, 180] based on the 5-piece timings of the code reported here.

In unpublished personal communication Thompson has indicated that the per-node time of the fastest serial endgame code is currently only a factor of approximately $7 \cdot 10^2$ times slower than that of the code reported in this paper (depending on the endgame) [168].

Unfortunately, direct comparison of 6-piece timing against other work is, of course, not currently possible since 6-piece endgames could not have been solved in a practicable amount of time using classical techniques on previous architectures. However, with larger and faster serial machines, and with enough spare cycles, 6-piece endgames are in fact coming within reach of classical solution techniques. This would permit a more informative timing comparison.

Thus, although per-node timing comparisons based on radically differently sized state-spaces are not very meaningful, the large per-node timing differential of the current program compared to classical programs does tend to support the hypothesis that the techniques reported here lend themselves to efficient parallel implementation.

The only program with per-node time of comparable speed to the author's CM-200 implementation is Burton Wendroff's et al. vectorized implementation of Table 1 [183], although this implementation currently solves only a single 4-piece endgame.

The CM-200 source code implementing Table 1 is currently available from `ftp.cs.jhu.edu:pub/stiller/snark`.

9 Future work

The main historical open question is to find out Molien's exact contribution to the history of numerical chess endgame analysis, and to locate and check his analysis of ♔♖♔♗. Kanunov [92, p.6] refers to private papers held by Molien's daughter; currently we are trying to locate these papers in the

hope that they might shed light on the questions raised in subsection 2.2. Amelung himself is also a figure about whom little is known, and the remarks here would seem to suggest that a detailed reassessment of his contribution to the endgame study would be desirable.

The question of Molien and Amelung's contributions to quantitative endgame analysis is part of the larger historical question of pre-digital precursors to computer chess algorithms. In addition to the work of Babbage, Molien, Amelung, Zermelo, and Quevedo, we remark that K. Schwarz, in a little-known 1925 article in *Deutsche Schachzeitung*, argued for a postional evaluation function similar to the squares-attacked heuristic used in some full-chess programs [144].

From a computational point of view, it might seem that the next logical step in the evolution of the current program should be the exhaustive solution of pawnless 7-piece endgames. In fact, in my opinion a more promising approach would be to follow up on the suggestions first made by Bellman [36, 37, 41] and solve endgames with multiple pawns and minor pieces. Such an approach would combine heuristic evaluation of node values corresponding to promotions with the exhaustive search techniques described here. Although the use of heuristics would introduce some errors, the results of such a search would, in my opinion, have considerable impact on the evaluation of many endgames arising in practical play.

Even more speculatively, it is also possible to search for certain classes of endgames considered artistic by endgame composers; such endgames typically depend on a key mutual-zugzwang or domination position some moves deep in the tree.

10 Acknowledgments

The author thanks his advisor, Simon Kasif, for his help in the formulation and presentation of these ideas.

Noam Elkies patiently provided invaluable chess advice and suggestions throughout the 6-piece project, including the mutual-zugzwangs algorithm. Burton Wendroff made possible much of the work and developed and implemented the vectorized Cray algorithm.

A. John Roycroft provided much of the historical information on human

endgame analysis.

Harold van der Heijden graciously provided all studies in his database of approximately 36000 endgame studies in which certain pawnless 6-piece endgames arose.

The author is also grateful to the following for useful conversations and suggestions: Alan Adler, Hans Berliner, Murray Campbell, Richard Draper, Charles Fiduccia, Ralph Gasser, Thomas Hawkins, Feng-Hsiung Hsu, Steven Salzberg, Jonathan Schaeffer, Ken Thompson, and David Waltz.

Assistance in translating many of the non-English sources was provided by Peter Jansen, Michael Klepikov, George Krotkoff, A. John Roycroft, Claudia Salzberg, Boris Statnikov, and the staff of the Cleveland Public Library.

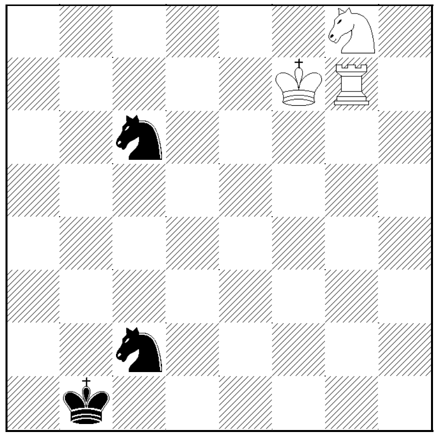
Access to most of the manuscripts, rare books and chess journals cited in the paper was obtained during the author's visit to the John Griswold White Collection of Chess, Orientalia and Fine Arts, located at the Cleveland Public Library, 325 Superior Avenue, Cleveland, OH 44414. The author thanks Alice N. Loranth and Motoko B. Reece of the Cleveland Public Library for their assistance during his visits to the collection.

The National Library of Latvia in Riga (Latvijas Nacionālā Bibliotēka) assisted in providing copies of a number of articles by Friedrich Amelung; Kenneth Whyld helped narrow down the search for these to the city of Riga.

Computing facilities were graciously provided by the Advanced Computing Laboratory of the Los Alamos National Laboratory, Los Alamos, NM 87545.

11 Appendix: A best play line

Distance-to-win (conversion) metric is used. Equioptimal moves are parenthesized. For technical reasons the last move in the line is omitted.



1♔f7-e6 ♘c6-b4 2♔e6-e5 ♘b4-d3 3♔e5-e4 ♘d3-f2 4♔e4-f3 ♘f2-d3 5♔f3-e2 ♘c2-b4 6♔e2-e3 ♔b1-b2 7♔e3-d4 ♘d3-f4 8♔d4-c4 ♘b4-d5 9♖g7-h7 ♘d5-e3 10♔c4-d4 ♘e3-c2 11♔d4-e4 ♘f4-e6 12♔e4-e5 ♘e6-g5 13♖h7-h5 ♘c2-e1 14♔e5-f5 ♘g5-f3 15♔f5-e4 (♔f5-f4) **♘f3-d2 16♔e4-e3 ♘d2-b3 17♖h5-h1 ♘e1-c2 18♔e3-d3 ♘b3-c1 19♔d3-e4 ♘c1-b3 20♖h1-h3 ♘b3-c5 21♔e4-e5 ♘c2-e1 22♘g8-f6 ♘e1-d3 23♔e5-d6 ♘c5-b7 24♔d6-c7 ♘b7-c5 25♔c7-c6 ♔b2-c2 26♖h3-h2 ♔c2-b3** (♔c2-c3) **27♔c6-d5 ♔b3-b4 28♔d5-d4** (♖h2-h4) **♘d3-f4 29♖h2-h4 ♔b4-b5 30♘f6-e8 ♘c5-b3 31♔d4-e4 ♘f4-g6 32♖h4-h7 ♘b3-c5 33♔e4-d4 ♘g6-f4 34♘e8-d6 ♔b5-c6 35♖h7-h6 ♘c5-b3 36♔d4-e4 ♘f4-e6 37♔e4-e5 ♘e6-d4 38♖h6-h3 ♘b3-c5 39♘d6-c8 ♘d4-c2 40♖h3-c3 ♘c2-b4 41♔e5-d4 ♘b4-a6 42♖c3-c2 ♔c6-d7 43♘c8-b6 ♔d7-d6 44♘b6-c4 ♔d6-c6 45♘c4-e3 ♔c6-d6 46♘e3-f5 ♔d6-e6 47♘f5-g7 ♔e6-f7 48♘g7-h5 ♘c5-e6 49♔d4-e5 ♘a6-b4 50♖c2-e2 ♘b4-d3 51♔e5-e4 ♘d3-b4 52♖e2-b2 ♔f7-g6 53♘h5-g3 ♘e6-g5 54♔e4-d4 ♘g5-e6 55♔d4-c4 ♘b4-a6 56♖b2-f2 ♘e6-g5 57♖f2-f1 ♘a6-c7 58♘g3-e2 ♘g5-f7 59♘e2-f4 ♔g6-g5 60♔c4-d4 ♘c7-b5 61♔d4-c5 ♘b5-d6 62♘f4-e6 ♔g5-g6 63♘e6-f8 ♔g6-g5 64♔c5-d5 ♘d6-f5 65♖f1-b1** (♖f1-a1) **♘f5-g3 66♖b1-b7 ♘f7-h6 67♖b7-g7 ♔g5-f4 68♘f8-e6 ♔f4-f3 69♖g7-b7 ♘g3-h5 70♖b7-b4 ♘h5-f6 71♔d5-d4 ♘f6-h5 72♔d4-d3 ♘h6-g4 73♘e6-g5 ♔f3-g3 74♘g5-e4 ♔g3-h4 75♖b4-a4 ♘h5-f4 76♔d3-d4 ♘f4-e6** (♘f4-e2) **77♔d4-d5 ♘e6-f4 78♔d5-d6 ♘f4-h3 79♖a4-**

a8 ♘g4-f2 80♘e4-c5 ♔h4-g5 81♔d6-e5 ♘f2-g4 82♔e5-d4 ♘h3-f4 83♘c5-e4 ♔g5-g6 84♖a8-a6 ♔g6-f5 85♖a6-a5 ♔f5-e6 86♘e4-c5 ♔e6-e7 87♖a5-a7 ♔e7-f6 88♔d4-e4 ♔f6-g5 89♖a7-a5 ♘f4-h5 90♘c5-e6 ♔g5-g6 91♖a5-b5 ♔g6-f7 92♘e6-c5 ♔f7-e7 93♖b5-b2 ♔e7-d6 94♘c5-b7 ♔d6-e7 95♖b2-a2 ♘h5-g7 96♖a2-e2 ♔e7-d7 97♖e2-g2 ♘g7-e8 98♔e4-f4 ♘g4-f6 99♔f4-e5 ♔d7-e7 100♖g2-e2 ♔e7-d7 101♘b7-a5 ♘f6-g4 102♔e5-f5 ♘g4-h6 103♔f5-g6 ♘h6-g8 104♘a5-c4 ♘e8-c7 105♔g6-f7 ♘g8-h6 106♔f7-f6 ♘h6-g8 107♔f6-e5 ♘g8-e7 108♖e2-d2 ♔d7-c6 109♖d2-c2 ♘c7-a6 (♘e7-g6) 110♘c4-e3 ♔c6-d7 111♖c2-d2 ♔d7-c6 112♖d2-d6 ♔c6-b5 113♖d6-h6 ♘e7-c8 114♔e5-d4 (♖h6-h5) ♘a6-b4 115♖h6-h5 ♔b5-c6 116♘e3-c4 ♘c8-e7 117♖h5-h6 ♔c6-c7 118♖h6-h7 ♔c7-d7 119♔d4-e5 ♘b4-d5 120♘c4-d6 (♘c4-d2) ♔d7-c6 121♘d6-e4 ♘e7-g6 122♔e5-f5 ♘g6-f8 123♖h7-h6 ♔c6-c7 124♖h6-h1 ♘f8-d7 125♖h1-b1 ♘d7-b8 126♔f5-e5 ♘d5-e3 127♔e5-d4 ♘e3-f5 128♔d4-d5 ♘f5-e3 129♔d5-c5 ♘b8-d7 130♔c5-d4 ♘e3-g4 131♖b1-c1 ♔c7-d8 132♖c1-e1 ♘g4-f6 133♘e4-g5 (♘e4-d6) ♔d8-c7 134♘g5-f7 ♘d7-f8 135♖e1-f1 ♘f6-g4 136♖f1-g1 ♘g4-f6 137♖g1-e1 ♔c7-d7 138♔d4-e5 ♘f6-e8 139♘f7-h8 ♔d7-e7 140♔e5-d5 ♔e7-d7 141♖e1-f1 ♘e8-c7 142♔d5-e5 ♘f8-e6 143♘h8-g6 ♘e6-c5 144♖f1-b1 ♔d7-c6 145♘g6-e7 ♔c6-d7 146♘e7-f5 ♔d7-c6 147♘f5-d4 ♔c6-d7 148♖b1-d1 ♘c7-a6 149♘d4-f5 ♔d7-c6 150♖d1-h1 ♘a6-b4 151♖h1-h6 ♔c6-d7 152♔e5-d4 ♘c5-e6 153♔d4-c4 ♘b4-a6 154♖h6-h7 ♔d7-c6 155♖h7-h1 ♘a6-c7 156♖h1-d1 ♘c7-e8 157♘f5-e7 ♔c6-c7 158♔c4-d5 ♘e6-f8 159♘e7-g8 ♔c7-d7 160♔d5-c5 ♔d7-e6 (♔d7-c7) 161♖d1-e1 ♔e6-d7 162♖e1-e7 ♔d7-d8 163♖e7-a7 ♘f8-d7 164♔c5-c6 ♘d7-e5 165♔c6-d5 ♘e5-g6 166♖a7-h7 ♘e8-c7 167♔d5-c6 ♘g6-e5 168♔c6-d6 ♘e5-c4 169♔d6-c5 ♘c4-e5 170♖h7-h5 ♘e5-f7 171♔c5-c6 ♘c7-e6 172♖h5-a5 ♔d8-e8 173♘g8-f6 ♔e8-e7 174♘f6-d5 ♔e7-f8 175♔c6-d7 ♘e6-d4 176♘d5-f4 ♘f7-h6 177♖a5-d5 ♘d4-f5 178♔d7-e6 ♘f5-g7 179♔e6-f6 ♘h6-g8 180♔f6-e5 ♘g8-h6 181♖d5-a5 ♘h6-g4 182♔e5-d4 (♔e5-d5) ♔f8-f7 183♖a5-a7 ♔f7-f6 184♔d4-e4 ♘g7-e8 185♖a7-a6 ♔f6-g7 186♖a6-b6 (♘f4-g2) ♘g4-f6 187♔e4-f5 ♘f6-d7 188♘f4-e6 ♔g7-f7 189♘e6-g5 ♔f7-f8 190♖b6-a6 ♘e8-g7 191♔f5-g6 ♘d7-e5 192♔g6-h7 ♘g7-e8 193♖a6-e6 ♘e5-f7 194♘g5-f3 ♘f7-d6 195♔h7-g6 ♘d6-f5

(♘d6-c8) **196♖e6-e1 ♘f5-e7 197♔g6-g5 ♔f8-f7 198♘f3-e5 ♔f7-g7 199♘e5-g4 ♔g7-f8 200♘g4-h6 ♘e7-d5 201♘h6-f5 ♔f8-f7 202♖e1-e2** (♖e1-e4♖e1-e5) **♘d5-b6 203♖e2-e7 ♔f7-f8 204♖e7-e1 ♘b6-d5 205♖e1-e5 ♘d5-b6** (♘e8-c7) **206♔g5-g6 ♘e8-c7 207♘f5-d6 ♘b6-d5 208♖e5-e1 ♘c7-e6** (♘d5-f4♘d5-e7♘d5-b4) **209♔g6-f5 ♘e6-c7 210♔f5-e5 ♘d5-b4** (♔f8-e7) **211♖e1-f1 ♔f8-e7 212♖f1-f7 ♔e7-d8 213♘d6-b7 ♔d8-c8 214♘b7-c5 ♘c7-b5 215♖f7-g7** (♖f7-h7) **♔c8-d8 216♖g7-b7 ♘b4-c6 217♔e5-e6 ♔d8-c8 218♖b7-h7 ♘c6-b4 219♘c5-a4 ♘b4-a6 220♔e6-d5 ♘b5-c7 221♔d5-d6 ♘c7-e8 222♔d6-e7 ♘e8-c7 223♖h7-h6 ♘a6-b8 224♘a4-b6 ♔c8-b7 225♘b6-c4 ♘b8-c6 226♔e7-d6** (♔e7-d7) **♘c6-b4 227♖h6-h8 ♘b4-a6 228♖h8-h7 ♔b7-c8 229♘c4-a5 ♔c8-d8 230♘a5-c6 ♔d8-c8 231♘c6-e7 ♔c8-d8 232♘e7-d5 ♘c7-e8 233♔d6-c6 ♘a6-b8 234♔c6-b5 ♘e8-d6 235♔b5-c5 ♘d6-c8 236♖h7-h8 ♔d8-d7 237♘d5-f6 ♔d7-c7 238♖h8-h7 ♔c7-d8 239♖h7-b7 ♘b8-a6 240♔c5-c6 ♘c8-e7 241♔c6-b6 ♘a6-b4 242♖b7-d7**

References

[1] Syed Kamal Abdali and Benjamin David Saunders. Transitive closure and related semiring properties via elimination. *Theoretical Computer Science*, **40**(2–3):257–274, 1985.

[2] Jeanne C. Adams, Walter S. Brainerd, Jeanne T. Martin, Brian T. Smith, and Jerrold L. Wagener. *Fortran 90 Handbook: Complete ANSI/ISO Reference*. McGraw-Hill Book Co., New York, 1992.

[3] [اَلعَدلِي]. كِتَاب الشَطرَنج, n.d. (al-'Adlī, Book of Chess. Photographic copy of Arabic manusicript).

[4] [Friedrich Ludwig Amelung?]. Baltische Schachpartien aus den Jahren 1858 bis 1892, Partie 114 [Baltic chess games from the years 1892 to 1858, Game 114]. *Baltische Schachblätter*, **4**:266–267, 1893. Score of consultation Theodor Molien and A. Hasselblatt v. Friedrich Amelung.

[5] FRIEDRICH LUDWIG AMELUNG. Das Endspiel von zwei Offizieren gegen einen Springer [The endgame of two pieces against a rook]. *Baltische Schachblätter*, **4**:290–297, 1893.

[6] FRIEDRICH LUDWIG AMELUNG. Auszüge aus den Briefen von A. Ascharin an F. Amelung [Excerpts from the letters of A. Ascharin to F. Amelung]. *Baltische Schacblätter*, **5**:12–38, 1898.

[7] FRIEDRICH LUDWIG AMELUNG. Das Endspiel von Thurm gegen Springer (Fortsetzung) [The endgame of rook against knight (continuation)]. *Deutsche Schachzeitung*, **55**(2):37–41, February 1900.

[8] FRIEDRICH LUDWIG AMELUNG. Das Endspiel von Thurm gegen Springer (Fortsetzung) [The endgame of rook against knight (continuation)]. *Deutsche Schachzeitung*, **55**(4):101–105, April 1900.

[9] FRIEDRICH LUDWIG AMELUNG. Das Endspiel von Thurm gegen Springer (Fortsetzung) [The endgame of rook against knight (continuation)]. *Deutsche Schachzeitung*, **55**(5):134–138, May 1900.

[10] FRIEDRICH LUDWIG AMELUNG. Das Endspiel von Thurm gegen Springer (Fortsetzung) [The endgame of rook against knight (continuation)]. *Deutsche Schachzeitung*, **55**(7):198–202, July 1900.

[11] FRIEDRICH LUDWIG AMELUNG. Das Endspiel von Thurm gegen Springer (Schluss) [The endgame of rook against knight (conclusion)]. *Deutsche Schachzeitung*, **55**(9):261–266, September 1900.

[12] FRIEDRICH LUDWIG AMELUNG. Das Endspiel von Thurm gegen Springer [The endgame of rook against knight]. *Deutsche Schachzeitung*, **55**(1), January 1900.

[13] FRIEDRICH LUDWIG AMELUNG. Das Endspiel von Thurm und Springer gegen die Dame [The endgame of rook and knight against queen]. *Deutsche Schachzeitung*, **56**(7):193–197, July 1901.

[14] FRIEDRICH LUDWIG AMELUNG. Das Endspiel von Thurm und Springer gegen die Dame [The endgame of rook and knight against queen]. *Deutsche Schachzeitung*, **56**(8):225–229, August 1901.

[15] FRIEDRICH LUDWIG AMELUNG. Die Endspiele mit Qualitätsvortheil, insbesondere das Endspiel von Thurm und Läufer gegen Läufer und

Springer [The endgames with exchange advantage, especially the endgame of rook and bishop against bishop and knight]. *Deutsche Schachzeitung*, **57**(9):265–268, September 1902.

[16] Friedrich Ludwig Amelung. Die Endspiele mit Qualitätsvortheil, insbesondere das Endspiel von Thurm und Läufer gegen Läufer und Springer (Fortsetzung) [The endgames with exchange advantage, especially the endgame of rook and bishop against bishop and knight (continuation)]. *Deutsche Schachzeitung*, **57**(10):297–300, October 1902.

[17] Friedrich Ludwig Amelung. Die Endspiele mit Qualitätsvortheil, insbesondere das Endspiel von Thurm und Läufer gegen Läufer und Springer (Schluss) [The endgames with exchange advantage, especially the endgame of rook and bishop against bishop and knight (conclusion)]. *Deutsche Schachzeitung*, **57**(11):330–332, November 1902.

[18] Friedrich Ludwig Amelung. Das Endspiel von Turm und Läufer gegen zwei Springer [The endgame of rook and bishop against two knights]. *Düna-Zeitung (Feuilleton-Beilage: Für Haus und Familie)*, **40**:52–53, 16–29 February 1908.

[19] Friedrich Ludwig Amelung. Endspiel 1028. *Deutsches Wochenschach*, **24**(14):130, 5 April 1908.

[20] Friedrich Ludwig Amelung. Lösungspreis-endspiel Nr. 178 [Solution prize-endgame No. 178]. *Düna-Zeitung (Feuilleton-Beilage: Für Haus und Familie)*, **63**:87, 15–28 March 1908.

[21] Friedrich Ludwig Amelung. Lösung des Preisendspiels Nr. 178 [Solution of prize endgame No. 178]. *Düna-Zeitung (Feuilleton-Beilage: Für Haus und Familie)*, **13**:20–21, 17–30 January 1909.

[22] [Düna-Zeitung]. Friedrich Ludwg Amelung. *Düna-Zeitung (Feuilleton-Beilage: Für Haus und Familie)*, **70**:76–78, 28 March–10 April 1909.

[23] Myoung An, James William Cooley, and Richard Tolimieri. Factorization method for crystallographic Fourier transforms. *Advances in Applied Mathematics*, **11**(3):358–371, September 1990.

[24] Myoung An, Chao Lu, E. Prince, and Richard Tolimieri. Fast Fourier transforms for space groups containing rotation axes of order

three and higher. *Acta Crystallographica*, **A48**(Part 3):346–349, May 1992.

[25] Fred S. Annexstein, Marc Baumslag, and Arnold Leonard Rosenberg. Group action graphs and parallel architectures. *SIAM Journal on Computing*, **19**(3):544–569, June 1990.

[26] Louis Auslander and Richard Tolimieri. Ring structure and the Fourier transform. *Mathematical Intelligencer*, **7**(3):49–52, 54, 1985.

[27] Юрий Львович Авербах [Yuri Lvovich Averbakh]. Шахматные Окончания; Ферзевые [Chess endings: queen]. Физкультура и спорт, Москва, 2nd edition, 1982.

[28] Amir Averbuch, Eran Gabber, Boaz Gordissky, and Yoav Medan. A parallel FFT on a MIMD machine. *Parallel Computing*, **15**(1–3):61–74, September 1990.

[29] Charles Babbage. Games of skill. In *Passages From the Life of a Philosopher*, pages 465–471. Longman, Green, Longman, Roberts, and Green, London, 1864.

[30] Roland C. Backhouse and B. A. Carré. Regular algebra applied to path-finding problems. *Journal of the Institute of Mathematics and its Applications*, **15**(2):161–186, April 1975.

[31] J.G. Bashmakova. *Molin, Fedor Eduardovich*, volume 3, pages 1739–1740. Charles Scribner's Sons, New York, 1991.

[32] [Carl Behting and Paul Kerkovius?]. Das zweite Baltische Schachturnier [The second Baltic chess tournament]. *Baltische Schachblätter*, **9**:1–24, 1902.

[33] Alex G. Bell. Torres y Quevedo. In *The Machine Plays Chess?*, Pergamon Chess Series, chapter 2, pages 8–11. Pergamon Press, Oxford, 1978.

[34] Richard Ernest Bellman. On a new iterative algorithm for finding the solutions of games and linear programming problems. Technical Report P-473, The RAND Corporation, U.S. Air Force Project RAND, Santa Monica, CA, 1 June 1954.

[35] Richard Ernest Bellman. The theory of games. Technical Report P-1062, The RAND Corporation, Santa Monica, CA, 15 April 1957.

[36] Richard Ernest Bellman. On the reduction of dimensionality for classes of dynamic programming processes. Technical report, The RAND Corporation, Santa Monica, CA, 7 March 1961.

[37] Richard Ernest Bellman. On the application of dynamic programming to the determination of optimal play in chess and checkers. *Proceedings of the National Academy of Sciences of the United States of America*, **53**(2):244–246, February 1965.

[38] Johann Berger. Der Turm und ein leichter Offizier gegen zwei leichte Offiziere [Rook and minor piece against two minor pieces]. In *Theorie und Praxis der Endspiele: Ein Handbuch für Schachfreunde*, pages 167–169. Veit and comp., Leipzig, 1890.

[39] Johann Berger. *Theorie und Praxis der Endspiele: Ein Handbuch für Schachfreunde* [Theory and practice of the endgame: a handbook for chessplayers]. Vereinigung Wissenschaftlicher Verleger Walter de Gruyter & Co., Berlin, Leipzig, 2nd edition, 1922.

[40] Elwyn Ralph Berlekamp, John Horton Conway, and Richard K. Guy. *Winning Ways For Your Mathematical Plays.* Volume 1: *Games in General.* Academic Press, London and New York, 1982.

[41] Hans Jack Berliner and Murray S. Campbell. Using chunking to solve chess pawn endgames. *Artificial Intelligence*, **23**(1):97–120, May 1984.

[42] Richard S. Bird, Jeremy Gibbons, and Geraint Jones. Formal derivation of a pattern matching algorithm. *Science of Computer Programming*, **12**(2):93–104, July 1989.

[43] Guy E. Blelloch. *Vector Models for Data-Parallel Computing.* Artificial Intelligence. MIT Press, Cambridge, MA, 1990.

[44] Gregory Butler. *Fundamental Algorithms For Permutation Groups*, volume 559 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin/New York, 1992.

[45] MURRAY S. CAMPBELL. Chunking as an abstraction mechanism. Technical Report CMU-CS-88-116, Carnegie-Mellon University, 22 February 1988.

[46] PIETRO CARRERA. *Del gioco de gli scacchi* [The game of chess], volume 3. [Gionanni de Rossi?], [Trento?], 1617.

[47] R.M. CHAMBERLAIN. Gray codes, fast Fourier transforms, and hypercubes. *Parallel Computing*, **6**(2):225–233, February 1988.

[48] CHAPAIS. *Essais analytiques sur les échecs, avec figures* [analytical essay on chess, with illustrations], [ca. 1780].

[49] ANDRÉ CHÉRON. *Nouveau traité complet d'échecs: La fin de partie* [New complete treatise of chess]. Yves Demailly, Lille, France, 1952.

[50] MICHAEL CLAUSEN AND ULRICH BAUM. *Fast Fourier Transforms.* Bibliographisches Institut Wissenschaftsverlag, Mannheim, 1993.

[51] JAMES WILLIAM COOLEY AND JOHN WILDER TUKEY. An algorithm for the machine calculation of complex Fourier series,. *Mathematics of Computation*, **19**(90):297–301, April 1965.

[52] [ALFRED CROSSKILL]. The rook and bishop against rook. *The Chess-Player's Magazine*, **2**:305–311, 1864.

[53] LUIS RAMIREZ DE LUCENA. *Repetición de Amores y Arte de Ajedrez* [treatise on love and the game of chess]. Salamanca, ca. 1497. Facsimile reprint Colección Joyas Bibliográficas:Madrid, Spain. 1953.

[54] RUY LOPEZ DE SIGURA. *Libro de la invencion liberal y arte del juego del axedrez* [The book of liberal invention and the art of the game of chess]. Alcala, 1561.

[55] [NICHOLAS DE ST NICHOLAI?]. *Bonus Socius* [good companion], 13th century.

[56] SITO T. DEKKER, H. JAAP VAN DEN HERIK, AND I.S. HERSCHBERG. Complexity starts at five. *International Computer Chess Association Journal*, **10**(3):125–138, September 1987.

[57] PERSI DIACONIS. *Group Representations in Probability and Statistics*, volume 11 of *Lecture Notes—Monograph Series.* Institute of Mathematical Statistics, Hayward, CA, 1988.

[58] Persi Diaconis and Daniel Nahum Rockmore. Efficient computation of the Fourier transform on finite groups. *Journal of the American Mathematical Society*, **3**(2):297–332, April 1990.

[59] Richard Noel Draper. A fast distributed routing algorithm for supertoroidal networks. Technical report, Supercomputing Research Center, July 1990.

[60] Richard Noel Draper. An overview of supertoroidal networks. Technical Report SRC-TR-91-035, Supercomputing Research Center, Bowie, Maryland, 17 January 1991.

[61] Noam David Elkies. Chess art in the computer age. *American Chess Journal*, **1**(2):48–52, September 1993.

[62] Machgielis Euwe. *Het Eindspel* [The endgame]. G.B. van Goor Zonen's Uitgeversmaatschappij, 's-Gravenhage;Batavia, 1940.

[63] Machgielis Euwe. *Stukken Tegen Stukken II* [Pieces against pieces II], volume 5 of *Het Eindspel*. G.B. van Goor Zonen's Uitgeversmaatschappij, 's-Gravenhage;Batavia, 1940.

[64] Machgielis Euwe. *Das Endspiel* [The endgame]. Das Schach-Archiv, F.L. Rattman, Hamburg, 1957.

[65] Reuben Fine. *Basic Chess Endings*. David McKay Company, New York, 1941.

[66] Ferdinand Georg Frobenius. Über die Primfactoren der Gruppendeterminante [On prime factors of group determinants]. *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin*, pages 1343–1382, 1896.

[67] William Fulton and Joe Harris. *Representation Theory: A First Course*, volume 129 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1991.

[68] Angel E. García and Lewis Benjamin Stiller. Computation of the mean residence time of water in the hydration shells of biomolecules. *Journal of Computational Chemistry*, **14**(11):1396–1406, November 1993.

[69] Ralph Gasser. Applying retrograde analysis to Nine Men's Morris. In David N. L. Levy and Don F. Beal, editors, *Heuristic Programming in Artificial Intelligence: the Second Computer Olympiad*, Ellis Horwood Series in Artificial Intelligence, pages 161–173. Ellis Horwood Limited, Chichester, England, 1991.

[70] E.N. Gilbert. Gray codes and paths on the n-cube. *Bell Systems Technical Journal*, **37**:815–826, May 1958.

[71] Harry Golombek. *Chess: A History*. G. P. Putnam's Sons, New York, 1976.

[72] John A. Granata, Michael Conner, and Richard Tolimieri. A tensor product factorization of the linear convolution matrix. *IEEE Transactions on Circuits and Systems*, **38**(11):1364–1366, November 1991.

[73] John A. Granata, Michael Conner, and Richard Tolimieri. The tensor product: a mathematical programming language for FFTs and other fast DSP operations. *IEEE Signal Processing Magazine*, **9**(1):40–48, January 1992.

[74] John A. Granata and Richard Tolimieri. Matrix representations of the multidimensional overlap and add technique. *IEEE Transactions on Circuits and Systems for Video Technology*, **1**(3):289–90, September 1991.

[75] Gioacchino Greco. *Trattato del Nobilissimo et Militare Essercitio de Scacchi nel Qvale si Contengono Molti Bellissimi Tratti et la Vera Scienza di Esso Gioco* [treatise on the very noble and military exercise of chess: the science of that game], 1624.

[76] Thomas William Hawkins. The origins of the theory of group characters. *Archive for History of Exact Sciences*, **7**(2):142–170, 3 March 1971.

[77] Thomas William Hawkins. Hypercomplex numbers, Lie groups, and the creation of group representation theory. *Archive for History of Exact Sciences*, **8**(4):243–287, 25 April 1972.

[78] Thomas William Hawkins. New light on Frobenius' creation of the theory of group characters. *Archive for History of Exact Sciences*, **12**(3):217–143, 14 August 1974.

[79] William Daniel Hillis. *The Connection Machine.* ACM Distinguished Dissertations. MIT Press, Cambridge, MA, 1985.

[80] William Daniel Hillis and Washington Taylor IV. Exploiting symmetry in high-dimensional finite-difference calculations. *Journal of Parallel and Distributed Computing*, **8**(1):77–79, January 1990.

[81] Ching-Tien Ho and S. Lennart Johnsson. Embedding meshes in Boolean cubes by graph decomposition. *Journal of Parallel and Distributed Computing*, **8**(4):325–339, April 1990.

[82] David Hooper and Kenneth Whyld. *The Oxford Companion to Chess.* Oxford University Press, Oxford, 2nd edition, 1992.

[83] Barbara Jane Huberman. A program to play chess end games. Technical Report CS 106, Stanford Artificial Intelligence Project Memo AI-65, Stanford University Department of Computer Science, 1968.

[84] Kai Hwang and Faye Alaye Briggs. *Computer Architecture and Parallel Processing.* McGraw-Hill Series in Computer Organization and Architecture. McGraw-Hill, New York, 1985.

[85] [Amelung family]. [Announcement]. *Düna-Zeitung*, page 52, 10–23 March 1909.

[86] Peter Jozef Jansen. KQKR: Assessing the utility of heuristics. *International Computer Chess Association Journal*, **15**(4):179–191, December 1992.

[87] Peter Jozef Jansen. KQKR: Awareness of a fallible opponent. *International Computer Chess Association Journal*, **15**(3):111–131, September 1992.

[88] Peter Jozef Jansen. *Using Knowledge about the Opponent in Game-Tree Search.* PhD thesis, Carnegie Mellon University Department of Computer Science, Pittsburgh, PA, September 1992. Also published as Tech. Report CMU-CS-92-192.

[89] Jeremy R. Johnson, Robert W. Johnson, Domingo Rodriguez, and Richard Tolimieri. A methodology for designing, modifying and implementing Fourier transform algorithms on various

architectures. *Circuits, Systems, and Signal Processing*, **9**(4):449–500, 1990.

[90] Robert W. Johnson. Automatic implementation of tensor products. Manuscript, 24 April 1989.

[91] Robert W. Johnson, Chua-Huang Huang, and Jeremey R. Johnson. Multilinear algebra and parallel programming. *Journal of Supercomputing*, **5**(2–3):189–217, October 1991.

[92] Николай Федорович Канунов [Nikolai Fedorovich Kanunov]. Федор Эдуардович Молин *1861–1941* [Fedor Eduardovich Molin 1861–1941]. Издательство «Наука», Москва, 1983.

[93] Mark Girshevich Karpovsky. Fast Fourier transforms on finite non-Abelian groups. *IEEE Transactions on Computers*, **C-26**(10):1028–1030, October 1977.

[94] Shivnandan D. Kaushik, Sanjay Sharma, and Chua-Huang Huang. An algebraic theory for modeling multistage interconnection networks. *Journal of Information Science and Engineering*, **9**(1):1–26, March 1993.

[95] Josef Kling and Bernhard Horwitz. *Chess Studies, or Endings of Games. - Containing Upwards of Two Hundred Scientific Examples of Chess Strategy.* C. J. Skeet, Charing Cross, England, 1851.

[96] Э. А. Комиссарчик, Арон Л. Футер [E. A. Komissarchik, and Aaron L. Futer]. Об анализе ферзевого эндшпиля при помощи ЭВМ [Analysis of a queen endgame on an IBM computer]. Проблемы Кибернетики *[Problemy Kibernetiki]*, **29**:211–220, 1974.

[97] Danny Kopec, Brent Libby, and Chris Cook. The endgame king, rook and bishop vs. king and rook (KRBKR). In *Proceedings: 1988 Spring Symposium Series: Computer Game Playing*, pages 60–61, Stanford University, Stanford, CA, 22–24 March 1988. American Association for Artificial Intelligence.

[98] Aleksey Grigoryevich Kopnin. Some special features of the endgame struggle Rook and Knight against 2 Knights (GBR class 0107). *EG*, **5**(70):89–92, January 1983.

[99] David G. Korn and Jules J. Lambiotte, Jr. Computing the fast Fourier transform on a vector computer. *Mathematics of Computation*, **33**(147):977–992, July 1979.

[100] Robert Lake, Jonathan Schaeffer, and Paul Lu. Solving large retrograde-analysis problems using a network of workstations. In H. Jaap van den Herik, I.S. Herschberg, and Jos W.H.M. Uiterwijk, editors, *Advances in Computer Chess 7*, pages 135–162. University of Limburg, Maastricht, Netherlands, 1994.

[101] Daniel J. Lehmann. Algebraic structures for transitive closure. *Theoretical Computer Science*, **4**(1):59–76, February 1977.

[102] Frank Thomson Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes.* M. Kaufmann Publishers, San Mateo, CA, 1992.

[103] Charles Eric Leiserson, Zahi S. Abuhamdeh, David C. Douglas, Carl R. Feynman, Mahesh N. Ganmukhi, Jeffrey V. Hill, William Daniel Hillis, Bradley C. Kuszmaul, Margaret A. St. Pierre, David S. Wells, Monica C. Wong, Shaw-Wen Yang, and Robert Zak. The network architecture of the CM-5. In *Proceedings of the 4th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 272–285, San Diego, CA, 29 June–1 July 1992. ACM, EATCS, ACM Press, New York, 1992.

[104] Wilhem Lenz, editor. *Deutschbaltisches biographisches Lexikon 1710–1960* [German-Baltic biographical dictionary 1710–1960]. Böhlau Verlag, Köln, 1970.

[105] Giovanni Battista Lolli. *Osservazione teorico-pratiche sopra il giuoco degli scacchi* [theoretical-practical observations on the game of chess]. Bologna, 1763.

[106] Marvin Marcus. *Finite Dimensional Multilinear Algebra: Part 1.* Marcel Dekker, Inc., New York, 1973.

[107] Aleksandar Matanović. *Enciklopedija Šahovskih Završnica* [Encyclopedia of Chess Endings]. Šahovski informator, Beograd, 1985.

[108] Russell Merris. Pattern inventories associated with symmetry classes of tensors. *Linear Algebra and Applications*, **29**:225–230, February 1980.

[109] Russell Merris. Pólya's counting theorem via tensors. *American Mathematical Monthly*, **88**(3):179–185, March 1981.

[110] Russell Merris. Applications of multilinear algebra. *Linear and Multilinear Algebra*, **32**(3–4):211–224, 1992.

[111] Russell Merris and William Watkins. Tensors and graphs. *SIAM Journal on Algebraic and Discrete Methods*, **4**(4):534–547, December 1983.

[112] Donald Michie and Ivan Bratko. Ideas on knowledge synthesis stemming from the KBBKN endgame. *International Computer Chess Association Journal*, **10**(1):3–10, March 1987.

[113] Федор Эдуардович Молин [Theodor Molien]. Berichtigung zu dem Aufsatze 'Ueber Systeme höherer complexer Zahlen' [Correction to the article 'on systems of higher complex numbers']. *Mathematische Annalen*, **42**:308–312, 1893.

[114] Федор Эдуардович Молин [Theodor Molien]. Ueber Systeme höherer complexer Zahlen [On systems of higher complex numbers]. *Mathematische Annalen*, **41**:83–156, 1893.

[115] Федор Эдуардович Молин [Theodor Molien]. Eine Bemerkung zur Theorie der homogenen Substitutionsgruppen [A remark on the theory of homogeneous substitution groups]. *Sitzungberichte Naturforscher-Gesellschaft Dorpat (Yurev in Estonia)*, **11**:259–274, 24 April 1897.

[116] Федор Эдуардович Молин [Theodor Molien]. Ueber di Anzahl der Variabeln einer irreductibelen Substitutionsgruppen [On the number of variables of irreducible substitution groups]. *Sitzungsberichte der ber Naturforscher-Gesellschaft Dorpat*, **11**:277–288, 1897.

[117] Федор Эдуардович Молин [Theodor Molien]. Ziffermässig genaue Ausrechnung aller 12 millionen Gewinne und Remisen im Endspiel „Thurm gegen Läufer" [Numerical exact computation of all 12 million wins and draws in the endgame "rook against bishop"], April 1897. Cited in the index of *Baltische Schacblätter* **8**, 1901, page 72.

[118] Федор Эдуардович Молин [Theodor Molien]. [Four endgame studies]. *Baltische Schachblätter*, **6**:208–209, 1898.

[119] Федор Эдуардович Молин [THEODOR MOLIEN]. Ueber die Invarianten der linearen Substitutionsgruppen [On the invariants of linear substitution groups]. *Sitzungsberichte Akademie der Wissenschaft. Berlin*, pages 1152–1156, 1898.

[120] [Федор Эдуардович Молин [THEODOR MOLIEN]]. [Partie 73]. *Baltische Schachblätter*, **7**:346, 1900. Score of game against W. Sohn.

[121] LENORE M. RESTIFO MULLIN. A mathematics of arrays. Technical Report 8814, CASE Center, Syracuse University, Syracuse, NY 13244, December 1988.

[122] HAROLD JAMES RUTHVEN MURRAY. *A History of Chess*. Oxford University Press, London, 1913.

[123] DAVID NASSIMI AND SARTAJ KUMAR SAHNI. Parallel permutation and sorting algorithms and a new generalized connection network. *Journal of the Association for Computing Machinery*, **29**(3):642–667, July 1982.

[124] HARRY J. NEFKENS. Constructing data bases to fit a microcomputer. *International Computer Chess Association Journal*, **8**(4):217–224, December 1985.

[125] JOHN NUNN. *Secrets of Rook Endings*. Batsford Chess Library. H. Holt, New York, 1992.

[126] JOHN NUNN. *Secrets of Pawnless Endings*. Batsford Chess Library. H. Holt, New York, 1994.

[127] OREN PATASHNIK. Qubic: $4 \times 4 \times 4$ tic-tac-toe. *Mathematics Magzine*, **53**(4):203–216, September 1980.

[128] MARSHALL CARLETON PEASE. An adaptation of the fast Fourier transform for parallel processing. *Journal of the Association for Computing Machinery*, **15**(2):252–264, April 1968.

[129] JUAN BAUTISTA SANCHEZ PEREZ. *El Ajedrez de D. Alfonso el Sabio* [The chess of D. Alfonso the Wise]. Tip. La Franco Española, Madrid, 1929.

[130] FRANÇOIS-ANDRÉ DANICAN PHILIDOR. *L'analyze des échecs: contenant une nouvelle methode pour apprendre en peu temps à se perfectionner dans ce noble jeu* [the analysis of chess: containing a new method for learning in a short time and towards perfection in the noble game]. London, [1749?].

[131] JOHN ROSS QUINLAN. Discovering rules by induction from large collections of examples. In Donald Michie, editor, *Expert systems in the micro-electronic ages*, pages 168–201. Edinburgh University Press, 1979.

[132] JOHN ROSS QUINLAN. Learning efficient classification procedures and their application to chess end games. In Ryszard Stanislaw Michalski, Jaime Guillermo Carbonell, and Tom Michael Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, pages 463–482. Tioga Publishing Company, Palo Alto, CA, 1983.

[133] LARS RASMUSSEN. Ultimates in KQKR and KQKN. *International Computer Chess Association Journal*, **11**(1):21–25, March 1988.

[134] HENRI RINCK. *1414 Fins de partie [1414 endgames].* Tipografia la Academica, Barcelona, 1950. 5'th edition of *150 Fins de Partie*, 1909.

[135] HENRI RINCK AND LOUIS MALPAS. *Dame contre tour et cavalier* [Queen against rook and knight]. L'Echiquier, Bruxelles, 1947.

[136] ARNOLD LEONARD ROSENBERG. Shuffle-oriented interconnection networks. Technical Report COINS 88-84, University of Massachusetts at Amherst Computer and Information Science, 11 October 1988.

[137] ARTHUR JOHN ROYCROFT. A note on 2S's v R + B. *EG*, **1**:197–198, April 1967.

[138] ARTHUR JOHN ROYCROFT. *Test Tube Chess: A Comprehensive Introduction to the Chess Endgame Study.* Faber and Faber Ltd., London, England, 1972.

[139] ARTHUR JOHN ROYCROFT. Two bishops against knight. *EG*, **5**(75), April 1983.

[140] ARTHUR JOHN ROYCROFT. A proposed revision of the '50-move rule': Article 12.4 of the Laws of Chess. *International Computer Chess Association Journal*, **7**(3):164–170, September 1984.

[141] Jan Rusinek. *Almost Miniatures: 555 Studies With Eight Chessmen*, volume 3 of *Selected Endgame Studies*. University of Limburg, 1994.

[142] Alessandro Salvio. *Trattato dell'inventione et art liberale del gioco di scacchi* [treatise of the invention and liberal art of the game of chess]. Napoli, 1634.

[143] Jonathan Schaeffer, Joseph Culberson, Norman Treloar, Brent Knight, Paul Lu, and Duane Szafron. A world championship caliber checkers program. *Artificial Intelligence*, **53**(2–3):273–289, February 1992.

[144] K. H. Schwarz. Versuch eines mathematischen Schachprinzips [Essay on mathematical chess principles]. *Deutsche Schachzeitung*, **53**(11):321–324, November 1925.

[145] Roland Silver. The group of automorphisms of the game of 3-dimensional Ticktacktoe. *American Mathematical Monthly*, **74**(3):247–254, March 1967.

[146] Geoff Leslie Simons. *Is Man a Robot?* John Wiley & Sons, Chichester, England, 1986.

[147] Charles Coffin Sims. Computations with permutation groups. In Stanley Roy Petrick, editor, *Proceedings of the Second Symposium on Symbolic and Algebraic Manipulation*, pages 23–28, Los Angeles, 23–25 March 1971. ACM Press, New York 1971.

[148] Charles Coffin Sims. *Computation With Finitely Presented Groups*, volume 48 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, England, 1994.

[149] Richard C. Singleton. On computing the fast Fourier transform. *Communications of the ACM*, **10**(10):647–654, October 1967.

[150] David Benson Skillicorn. Structuring data parallelism using categorical data types. In *Proceedings of the 1993 Workshop on Programming Models for Massively Parallel Computers*, pages 110–115, Berlin, Germany, 20–23 September 1993. IEEE Computer Society Press, Los Alamitos, CA, 1993.

[151] David Smitley and Kent Iobst. Bit-serial SIMD on the CM-2 and the Cray-2. *Journal of Parallel and Distributed Computing*, **11**(2):135–145, February 1991.

[152] Philip Stamma. *Essai sur le jeu des échecs* [essay on the game of chess]. Paris, 1737.

[153] Lewis Benjamin Stiller. Parallel analysis of certain endgames. *International Computer Chess Association Journal*, **12**(2):55–64, June 1989.

[154] Lewis Benjamin Stiller. Group graphs and computational symmetry on massively parallel architecture. *Journal of Supercomputing*, **5**(2/3):99–117, November 1991.

[155] Lewis Benjamin Stiller. Karpov and Kasparov: the end is perfection. *International Computer Chess Association Journal*, **14**(4):198–201, December 1991.

[156] Lewis Benjamin Stiller. An algebraic foundation for Fortran 90 communication intrinsics. Technical Report LA-UR-92-5211, Los Alamos National Laboratory, Los Alamos, NM 87545, August 1992.

[157] Lewis Benjamin Stiller. An algebraic paradigm for the design of efficient parallel programs. Technical Report JHU-92/26, Deparment of Computer Science, The Johns Hopkins University, Baltimore, MD 21218, November 1992.

[158] Lewis Benjamin Stiller. *Exploiting Symmetry on Parallel Architectures*. PhD thesis, Department of Computer Science, The Johns Hopkins University, Baltimore, MD, 1995.

[159] Lewis Benjamin Stiller, Luke L. Daemen, and James E. Gubernatis. n-body simulations on massively parallel architectures. *Journal of Computational Physics*, **115**(2):550–552, December 1994.

[160] Harold Stuart Stone. *High-Performance Computer Architecture*. Addison-Wesley, Reading, MA, 1993.

[161] Thomas Ströhlein. *Untersuchungen über Kombinatorische Spiele* [Investigations of combinatorial games]. PhD thesis, Fakultät für Allgemeine Wissenschaften der Technischen Hochshule München, February 1970.

[162] [Scientific American Supplement]. Torres and his remarkable automatic devices. *Scientific American Supplement*, **53**(2079):296–298, November 1915.

[163] Boleslaw K. Szymanski, James Hicks, R. Jagannathan, Vivek Sarkar, David Benson Skillicorn, and Robert K. Yates. Is there a future for functional languages in parallel programming? In *Proceedings of the 1994 International Conference on Computer Languages*, pages 299–304, Toulouse, France, 16–19 May 1994. IEEE Computer Society Press, Los Alamitos, CA, 1994.

[164] Robert Endre Tarjan. Fast algorithms for solving path problems. *Journal of the Association for Computing Machinery*, **28**(3):594–614, July 1981.

[165] Robert Endre Tarjan. A unified approach to path problems. *Journal of the Association for Computing Machinery*, **28**(3):577–593, July 1981.

[166] [Kenneth Lane Thompson]. Chess games. *EG*, **5**(74), November 1983.

[167] Kenneth Lane Thompson. Retrograde analysis of certain endgames. *International Computer Chess Association Journal*, **9**(3):131–139, September 1986.

[168] Kenneth Lane Thompson. personal communication, April 1990.

[169] Richard Tolimieri and Myoung An. *Computations in X-ray crystallography*, volume 315 of *NATO ASI Series C: Mathematical and Physical Sciences*, pages 237–250. Kluwer Academic Publishers, Dordrecth/Boston, 1990.

[170] Richard Tolimieri, Myoung An, and Chao Lu. *Algorithms for Discrete Fourier Transform and Convolution.* Springer-Verlag, New York, 1989.

[171] Richard Tolimieri, Myoung An, and Chao Lu. *Mathematics of Multidimensional Fourier Transform Algorithms.* Springer-Verlag, New York, 1993.

[172] Gonzalo Torres-Quevedo. Présentation des appareils de Leonardo Torres-Quevedo [Presentation of the apparatus of Leonardo Torres-Quevedo]. In *Les machines à calculer et la pensée humaine*, volume 37 of *Colloques internationaux du Centre National de la Recherche Scientifique*, pages 383–406. Service des Publications du Centre National de la Recherche Scientifique, Paris, 1953, 9–13 January 1951.

[173] Алексей Алексеевич Троицкий [Alekseii Alekseevich Troitzky]. Два коня против пешек (теоретический очерк). In *Сборник Шахматных Этюдов, с Приложением Краткой Теории Эндшпиля: Два Коня Против Пешек* [Collection of chess studies, with a theoretical supplement on the the endgame of two knights against pawn], pages 248–288. ОГИЗ, Физкультура и туризм, Москва, 1934.

[174] Leslie G. Valiant. A bridging model for parallel computation. *Communications of the ACM*, **33**(8):103–111, August 1990.

[175] Leslie G. Valiant. General purpose parallel architectures. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science: Algorithms and Complexity*, volume A, chapter 18, pages 945–971. Elsevier Science Publishers, Amsterdam, 1990.

[176] Eric F. van de Velde. *Concurrent Scientific Computing*, volume 16 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 1994.

[177] H. Jaap van den Herik and I. S. Herschberg. The construction of an omniscient endgame database. *International Computer Chess Association Journal*, **8**(2):66–87, June 1985.

[178] H. Jaap van den Herik and I. S. Herschberg. Elementary theory improved, a conjecture refuted. *International Computer Chess Association Journal*, **8**(3):141–149, September 1985.

[179] H. Jaap van den Herik and I.S. Herschberg. A data base on data bases. *International Computer Chess Association Journal*, **9**(1):29–34, March 1986.

[180] H. Jaap van den Herik, I.S. Herschberg, T.R. Hendriks, and J.P. Wit. Computer checks on human analyses of the KRKB endgame. *International Computer Chess Association Journal*, **11**(1):26–31, March 1988.

[181] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, Princeton, NJ, 1944.

[182] Bartel Leendert Waerden. *A History of Algebra: From Al-Khwārizmī to Emmy Noether.* Springer-Verlag, Berlin/New York, 1985.

[183] Burton Wendroff, Tony Warnock, Lewis Benjamin Stiller, Dean Mayer, and Ralph Brickner. Bits and pieces: constructing chess endgame databases on parallel and vector architectures. *Applied Numerical Mathematics*, **12**(1–3):285–95, May 1993.

[184] Arthur Thomas White. *Graphs, Groups, and Surfaces.* North-Holland/Elsevier, New York, 1984.

[185] Dennis Edward White. Multilinear enumerative techniques. *Linear and Multilinear Algebra*, **2**:341–352, 1975.

[186] Hans Wussing. *The Genesis of the Abstract Group Concept: A Contribution to the History of the Origin of Abstract Group Theory.* The MIT Press, Cambridge, MA, 1984. Translation by Abe Shenitzer of Hans Wussing's *Die Genesis des abstrakten Gruppenbegriffes: ein Beitrag zur Entstehungsgeschichte der abstrakten Gruppentheorie*, Deutscher Verlag der Wissenschaften, Berlin, 1969.

[187] Ernst Zermelo. Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels [On an application of set theory to the theory of playing chess]. In Ernest William Hobson and Augustus Edward Hough Love, editors, *Proceedings of the Fifth International Congress of Mathematicians*, pages 501–504, Vol. 2, Cambridge, England, 22–28 August 1912. Cambridge University Press, Cambridge, England, 1913.