Games of No Chance
MSRI Publications
Volume **29**, 1996

Multilinear Algebra and Chess Endgames

LEWIS STILLER

ABSTRACT. This article has three chief aims: (1) To show the wide utility of multilinear algebraic formalism for high-performance computing. (2) To describe an application of this formalism in the analysis of chess endgames, and results obtained thereby that would have been impossible to compute using earlier techniques, including a win requiring a record 243 moves. (3) To contribute to the study of the history of chess endgames, by focusing on the work of Friedrich Amelung (in particular his apparently lost analysis of certain six-piece endgames) and that of Theodor Molien, one of the founders of modern group representation theory and the first person to have systematically numerically analyzed a pawnless endgame.

1. Introduction

Parallel and vector architectures can achieve 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 [Szymanski et al. 1994; Stone 1993].

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 [Johnson et al. 1991]. The approach is this:

- Define 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.
- Formulate the application problem as a matrix multiplication.
- Factor the matrix corresponding to the application in terms of the fixed matrices using addition, tensor product, and matrix multiplication as combining operators.
- Generate code from the matrix factorization.

Supported by U.S. Army Grant DAAL03–92–G–0345.

This approach has been used in signal processing algorithms [Granata and Tolimieri 1991; Granata et al. 1991; Johnson et al. 1990; Tolimieri et al. 1989; Tolimieri et al. 1993]. The success of that work motivates the present attempt to generalize the domain of application of the multilinear-algebraic approach to parallel programming [Stiller 1992b].

I have used the methodology presented in this paper in several domains, including parallel N-body codes [Stiller 1994], Fortran 90 communication intrinsic functions [Stiller 1992a], and statistical computations [García and Stiller 1993]. 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 architecture. By successively unmoving pieces from the set of mating positions, the set of positions from which White could win can be generated. This application presents an 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, and had to be generalized to 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.

Two main results are reported here:

- Table 1 on page 168 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.
- Table 2 on page 175 presents a statistical summary of the state space of several six-piece chess endgames. This table could not have been generated in a practicable amount of time using previous techniques.

Section 2 below 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. Readers interested only in chess and not in the mathematical and programming aspects of this work can skip from this section to Section 8.

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.

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 equations defining the dynamic programming solution to chess endgames (Table 1). Section 6.2 describes how these equations 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 very briefly some possible refinements to the algorithm.

Section 8 presents some of the chess results discovered by the program, including the best play from a position that requires 243 moves to win.

Section 9 gives some implementation details and discusses run times.

2. Endgame Analysis by Humans and Computers

We start with a brief historical survey of the analysis of endgames, particularly those containing at most six pieces.

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.

2.1. Human analysis. Endgame analysis appears to date from at least the ninth century, with al-ʻAdlī's analysis of positions from ♔♖♔♘ and ♔♖♘♔♖ [ʻAdlī, plates 105 and 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 [Perez 1929, pp. 111–112].

Modern chess is generally considered to have begun roughly with the publication of Luis Ramirez de Lucena's *Repetición de amores y arte de ajedrez* [Lucena 1497?]. Ruy Lopez de Sigura's book [Lopez 1561] briefly discusses endgame theory, but its main impact on this work would be the introduction of the controversial fifty-move rule [pp. 55–56], under which a game that contains fifty consecutive moves for each side without the move of a pawn or a capture could be declared drawn [Roycroft 1984].

Pietro Carrera's *Il gioco de gli scacchi* [Carrera 1617] discussed a number of fundamental endgames such as ♔♕♔♗♗, and certain six-piece endgames such as ♔♖♖♔♖♘ and ♔♖♖♔♖♗ [Book 3, pp. 176–178]. A number of other authors began developing the modern theory of endgames [Greco 1624; Stamma 1737; Philidor 1749?]. Giovanni Lolli's monumental *Osservazioni teorico-pratiche sopra il giuoco degli scacchi* [Lolli 1763] would be one of the most significant advances in endgame theory for the next ninety years [Roycroft 1972]. Lolli analyzed the endgame ♔♕♔♗♗, and he agreed with the earlier conclusion of Salvio [1634] that the endgame was a general draw. This assessment would stand substantially unchanged until Kenneth Thompson's computer analysis demonstrated a surprising 71-move win [Thompson 1986]. Notwithstanding this error, Lolli did discover the unique ♔♕♔♗♗ position in which White to play draws but Black to play loses [pp. 431–432].

Bernhard Horwitz and Josef Kling's *Chess Studies* [Kling and Horwitz 1851, pp. 62–66] contained a number of influential endgame studies, although their analysis of ♔♗♗♔♘ was questioned in [Roycroft 1972, p. 207]. The Horwitz and Kling assessment was definitively shown to be incorrect by two independent computer analyses [Thompson 1983; Roycroft 1984].

Alfred Crosskill [1864] gave an analysis of ♔♖♗♔♖ in which he claimed that more than fifty moves were required for a win; 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 [Philidor 1749?, pp. 165–169].

Henri Rinck and Aleksei Troitzky were among the most influential endgame composers of the next two generations. Troitzky is well-known for his analysis of ♔♘♘♔♙, in which he demonstrated that more than fifty moves are at times required for a win [Troitskiĭ 1934]. Rinck was a specialist in pawnless endgames, composing more than 500 such studies [Rinck 1950; Rinck and Malpas 1947], including some with six pieces. Troitzky summarized previous work in the area of ♔♘♘♔♙, beginning with a problem in ♔♘♔♙ from the thirteenth-century Latin manuscript *Bonus Socius* [Anonymous], and reserved particular praise for the systematic analysis of this endgame in an eighteenthth-century manuscript [Chapais 1780]. An early version of the program reported in the present paper resulted in the first published solution for this entire endgame [Stiller 1989].

The twentieth century saw the formal codification of endgame theory in works such as [Berger 1890; Chéron 1952; Euwe 1940; Fine 1941; Averbakh 1982], and many others. Some work focusing particularly on pawnless six-piece endings has also appeared, for example, [Roycroft 1967; Kopnin 1983; Berger 1922].

Currently the Informator *Encyclopedia of Chess Endings* series [Matanović 1985], which now uses some of Thompson's computer analysis, is a standard reference. The books [Nunn 1992; Nunn 1994] are based on that work.

Additional historical information can be found in [Hooper and Whyld 1992; Golombek 1976; Murray 1913; Roycroft 1972].

2.2. Friedrich Amelung and Theodor Molien. Friedrich Ludwig Amelung (born March 11, 1842; died 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 [Lenz 1970, p. 11; Düna-Zeitung 1909a; 1909b]. 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 [Amelung 1901]; this endgame was shown to have unexpected depth, requiring up to 46 moves to win, in [Stiller 1989]. He also published an article [Amelung 1893] on ♔♗♘♔♘ and ♔♗♗♔♘, combinations that were not exhaustively analyzed until the 1980s [Thompson 1986; Stiller 1989].

However, his main interest to our work actually lies in two major projects: an analysis of the four-piece endgame ♔♖♔♘ [Amelung 1900], and his studies of certain pawnless six-piece endgames [Amelung 1902; 1908a; 1908b; 1908c].

Amelung's 1900 analysis of ♔♖♔♘ was significant because it contained the first histogram, to my knowledge, of a pawnless endgame or, for that matter, of any endgame [Amelung 1900, 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, the *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 was born on September 10, 1861 in Riga, and died on December 25, 1941. (The biographical information here has been taken from [Kanunov 1983], which was translated for me by Boris Statnikov; see also [Bashmakova 1991]. Kanunov gives his name Федор Эдуардович Молин, or Fyodor Eduardovich Molin; we write Theodor Molien, in conformity with his publications. Amelung variously used Mollin, Mollien and Molien.)

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 Norwegian, as well as German and Russian, of course. "Read a hundred novels in a language," Molien liked to say, "and you will know that language."[1] He studied celestial mechanics at Dorpat University (1880–1883) and also took courses from Felix Klein in Leipzig (1883–1885).

He studied celestial mechanics at Dorpat University (1880–1883) and took courses from Felix Klein in Leipzig (1883–1885). His doctoral dissertation, published in *Mathematische Annalen* [Molien 1893], proved a number of the fundamental structure theorems of group representation theory, including the decomposability of group algebras into direct sums of matrix algebras.

[1] «Прочитайте сто романов на каком-либо языке,—любил говорить он позднее,—и Вы будете знать этот язык.» [Kanunov 1983, p. 9].

Molien's early papers on group representation theory [1893; 1897a; 1897b; 1898], despite their importance, were obscure and difficult to understand. They anticipated Frobenius' classic paper on the determinant of a group-circulant matrix [Frobenius 1896], a fact that Frobenius readily admitted [Hawkins 1974], although he had tremendous difficulty understanding Molien's work (letter to Alfred Knezer, May 6, 1898). Referring to [Molien 1893], Frobenius wrote to Dedekind:

> You will have noticed that a young mathematician, Theodor Molien in Dorpat, has independently of me considered the group determinant. He has published ... a very beautiful but hard-to-read work "On systems of higher complex numbers", in which he investigated non-commutative multiplication and obtained important general results of which the properties of the group determinant are special cases.[2]

(This letter of February 24, 1898 was kindly supplied by Thomas Hawkins, in a transcription made by Walter Kaufmann-Bühler; it is quoted here with the permission of Springer-Verlag. I have benefited from Hawkins' translation in supplying mine.)

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 [Kanunov 1983, pp. 35–36]. After studying medieval mathematical manuscripts at the Vatican Library in 1899, 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.

As a consequence, Molien's work was unknown or underestimated in the West for a long while; for example, the classic text [Wussing 1969] barely mentions him. With the publication of Hawkins' series of articles [1971; 1972; 1974] on the history of group representation theory, the significance of his work became better-known, and more recent historical appraisals gives him due credit [van der Waerden 1985, pp. 206–209, 237–238].

Not mentioned in Kanunov's biography is that, before Molien moved to Tomsk, he was one of the strongest chess players in Dorpat and was particularly known for his blindfold play (Ken Whyld, personal communication, 1995). He was

[2] "Sie werden bemerkt haben, daß sich 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."

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 [Balt. Schachbl. 1893; 1902, p. 8]; one of the games he lost fetched a "best-game" prize in the main tournament of the Jurjewer chess club in 1894 [Molien 1900].

In 1898 Molien published four chess studies [Molien 1898a] based on his research into the endgame ♔♖♔♗ [Amelung 1900, p. 5]. These numerical studies are alluded to several times in the chess journals of the time [Amelung 1898; 1909, pp. 5, 265; Molien 1901], but I have not been able to locate a publication of his complete results, despite the historical significance of such a document.

It is an interesting coincidence that within a span of a few years Molien performed groundbreaking work in the two apparently unrelated areas of 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, 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 [Clausen and Baum 1993; Diaconis 1988; Diaconis and Rockmore 1990; Karpovsky 1977]; massively parallel implementations are described in [Stiller 1995, 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 the influential manual of endings [Berger 1890, pp. 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 [Amelung 1902], and representing a continuation of Amelung's earlier work with Molien on the endgame ♔♖♔♘ [Amelung 1900]. Amelung indicated that the case ♔♖♗♔♘♘ 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 [Amelung 1908a].

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 [Amelung 1908c]. A solution was published the next year, 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 have been largely forgotten. It is discussed in [Berger 1922, p. 223–233], but it was criticized by the mathematician and chess champion Machgielis (Max) Euwe in his titanic study of pawnless endgames [Euwe 1940, Volume 5, pp. 50–53]:

> "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."[3]

The *Düna-Zeitung* has 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 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. [Babbage 1864, p. 467]

Automated endgame play appears to date from the construction by Leonardo Torres-Quevedo of an automaton to play ♔♖♔ endings. Although some sources give 1890 as the date in which this machine was designed, it was exhibited at about 1915 [Bell 1978, pp. 8–11; Simons 1986]. According to [Scientific American 1915, p. 298], "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."

Unlike most later work, Torres-Quevedo's automaton could move its own pieces. It used a rule-based approach [Scientific American 1915; Torres-Quevedo 1951], like that of [Huberman 1968]. 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 [Zermelo 1913]. Additional theoretical work was done by John von Neumann and Oskar Morgenstern [1944, pp. 124–125].

The contemporary dynamic programming methodology, which defines the field of retrograde endgame analysis, was discovered by Richard Bellman [1965]. (Strangely enough, this article is not generally known to the computer game community, and is not included in the comprehensive bibliography by van den

---

[3] "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." (Translation from the Dutch by Peter Jansen.)

Herik and I. S. Herschberg [1986]. Bellman's work was the culmination of work going back several years:

> *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 [Bellman 1961, p. 3].

Bellman [1954; 1957] had considered game theory from a classical perspective as well, but his work came to fruition with [Bellman 1965], 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. for checkers [Lake et al. 1994; Schaeffer et al. 1992; Schaeffer and Lake 1996] and Ralph Gasser for Nine Men's Morris [Gasser 1991; 1996]. 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].

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. J. Ross Quinlan [1979; 1983] analyzed ♔♖♔♘ from the point of view of a machine learning testbed. Hans Berliner and Murray S. Campbell [1984] studied the Szén position of three connected passed pawns against three connected passed pawns by simplifying the promotion subgames. Campbell [1988] has begun to extend this idea to wider classes of endgames. Jansen [1992a; 1992b; 1992c] has studied endgame play when the opponent is presumed to be fallible. H. Jaap van den Herik and coworkers have produced a number of retrograde analysis studies of various four-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 four-piece endgame [Dekker et al. 1987; van den Herik and Herschberg 1985a; 1988]. Danny Kopec has written several papers in the area as well [Kopec et al. 1988].

The first retrograde analysis of general five-piece endgames with up to one pawn was published in [Thompson 1986]. The significance of this work was twofold. First, many more moves were required to win certain endgames than had previously been thought. Second, Thompson's work invalidated generally accepted theory concerning certain five-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 [Michie and Bratko 1987]. The pawnless five-piece work of Thompson was extended to all pawnless five-piece endgames and many five-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 [van de Velde 1994; Stone 1993]. 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 [Hwang and Briggs 1985]. There are, of course, many trade-offs 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) [Hillis 1985; Valiant 1990a; 1990b; Blelloch 1990].

One common form of interconnection network is the hypercube [Leighton 1992]. 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:

$$E_8 \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_8 \end{pmatrix} = \begin{pmatrix} 0 \\ v_1 \\ \vdots \\ v_7 \end{pmatrix} \tag{3.1}$$

The computation is performed 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 [Gilbert 1958]; see Figure 1. Higher-dimensional shift patterns are also easy to compute on a hypercube [Ho and Johnsson 1990].

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

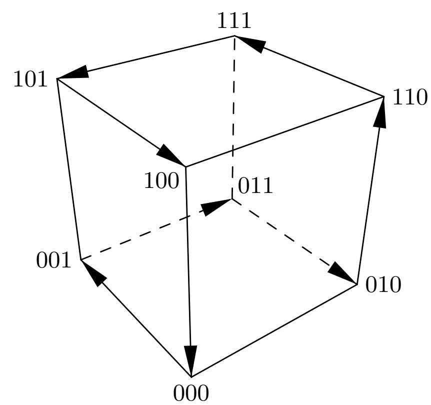


Figure 1. Left multiplication by an end-off shift matrix can be computed by Gray coding the coordinates of each element of the array. This figure illustrates a three-bit Gray code, which can be thought of as an embedding of a cycle of length eight into a three-cube.

processor permutation is typically performed with hardware assistance [Nassimi and Sahni 1982; Leiserson et al. 1992].

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 $\{\boldsymbol{e}_i^n\}_{i=1}^n$ be the standard basis consisting of vectors (having one component 1 and all others 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 $\mathfrak{F}$ [Marcus 1973].

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

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

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

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

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

The importance of the tensor-product to our work in parallel processing lies in the following identity, for $B \in \mathfrak{M}_m$ [Johnson et al. 1991]:

$$(I_n \otimes B) \begin{pmatrix} \boldsymbol{v}_0 \\ \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_{nm-1} \end{pmatrix} = \begin{pmatrix} B \cdot \begin{pmatrix} \boldsymbol{v}_0 \\ \vdots \\ \boldsymbol{v}_{m-1} \end{pmatrix} \\ B \cdot \begin{pmatrix} \boldsymbol{v}_m \\ \vdots \\ \boldsymbol{v}_{2m-1} \end{pmatrix} \\ \vdots \\ B \cdot \begin{pmatrix} \boldsymbol{v}_{(n-1)m} \\ \vdots \\ \boldsymbol{v}_{nm-1} \end{pmatrix} \end{pmatrix} \tag{4.1}$$

Suppose $n = ml$. The n-point stride l permutation matrix P_l^n is the $n \times n$ matrix defined by $P_l^n (\boldsymbol{v} \otimes \boldsymbol{w}) = \boldsymbol{w} \otimes \boldsymbol{v}$, where $\boldsymbol{v} \in \mathfrak{V}_m$ and $\boldsymbol{w} \in \mathfrak{V}_l$. The effect of P_l^n 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:

$$P_2^6 \begin{pmatrix} \boldsymbol{v}_0 \\ \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \boldsymbol{v}_3 \\ \boldsymbol{v}_4 \\ \boldsymbol{v}_5 \end{pmatrix} = \begin{pmatrix} \boldsymbol{v}_0 \\ \boldsymbol{v}_2 \\ \boldsymbol{v}_4 \\ \boldsymbol{v}_1 \\ \boldsymbol{v}_3 \\ \boldsymbol{v}_5 \end{pmatrix}$$

Stride permutations are important due to the following *Commutation Theorem* [Tolimieri et al. 1989]:

THEOREM 4.1. *If $A \in \mathfrak{M}_m$, $B \in \mathfrak{M}_l$, and $n = ml$, we have*

$$P_l^n (A \otimes B) P_m^n = B \otimes A.$$

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 Section 4.2 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. We now describe 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 section is fundamental to this work.

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

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

Assuming that m divides n, elements $\boldsymbol{v}_0, \dots, \boldsymbol{v}_{(n/m)-1}$ are stored in processor 0, elements $\boldsymbol{v}_{n/m}, \dots, \boldsymbol{v}_{(2n/m)-1}$ 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.

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

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

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

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

Consider a parallel processor with m processors, $p_1, \dots, p_m$, 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 $\mathbf{code}(I_m \otimes B)$ as code that runs on this m-processor architecture. To construct $\mathbf{code}(I_m \otimes B)$, load $\mathbf{code}(B)$ in each p_i. When called on a length ml array $\boldsymbol{v}$, p_i runs $\mathbf{code}(B)$ on the l elements of $\boldsymbol{v}$ that are stored in its local memory, and outputs the result to its local memory. Equation (4.1) 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 $A \otimes I_l$, for $A \in \mathfrak{M}_m$, is a bit more subtle. The interpretation of $\mathbf{code}(A \otimes I_l)$ 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 $\mathbf{code}(A)$ by interpreting the length ml argument array $\boldsymbol{v}$ 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 $A \otimes B = (A \otimes I_l)\,(I_m \otimes B)$ 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, one can define concise expressions that can be translated into efficient code for certain classes of functions [Granata et al. 1992].

4.3. Fast Fourier transforms. We discuss briefly the formulation of the FFT in the tensor product framework presented above [Tolimieri et al. 1993, pp. 16–20]. The presentation is intended to illustrate the parallel code development methodology used to describe parallelization of the chess endgame algorithm, in Section 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 $(\omega^{ij})_{i,j=0}^{n-1}$, where ω is a primitive n-th root of unity.

Let $n = ml$. The Singleton [1967] mixed-radix version of the Cooley–Tukey fast Fourier transform [Cooley and Tukey 1965] can be expressed recursively by

$$F_n = (F_m \otimes I_l) T_l (I_m \otimes F_l) P^n_m,$$

where T_l is a diagonal matrix encoding the twiddle factors:

$$T_l = \bigoplus_{j=0}^{m-1} \bigl(\mathrm{diag}(1, \omega, \ldots, \omega^{l-1})\bigr)^j.$$

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

The commutation theorem can be used to derive a parallel form

$$F_n = P^n_m \left(I_l \otimes F_m\right) P^n_l T_l \left(I_m \otimes F_l\right) P^n_m, \tag{4.2}$$

and a vector form

$$F_n = \left(F_m \otimes I_l\right) T_l P^n_m \left(F_l \otimes I_m\right). \tag{4.3}$$

The parallel Pease FFT [Pease 1968] can be derived by unrolling the recursion in (4.2), and the vectorized Korn–Lambiotte FFT [Korn and Lambiotte 1979] can be derived by unrolling (4.3).

By using the commutation theorem and varying the factorization, many different FFT algorithms have been derived, with different tradeoffs between parallelization and vectorization [Chamberlain 1988; Averbuch et al. 1990; Johnson 1989; Granata et al. 1991; Auslander and Tolimieri 1985].

5. Application to Chess

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

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 $\wedge$. 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 [Backhouse and Carré 1975; Lehmann 1977; Abdali and Saunders 1985; Tarjan 1981a; 1981b]. 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 1993; Bird et al. 1989], or using the mathematics-of-arrays formalism of [Mullin 1988].

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

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

An *S-position* is a chess position that contains exactly the k pieces in S. We write $S = \langle S_1, S_2, \dots, S_k \rangle$. 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).

We denote by $\mathfrak{V}_n$ 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 $\{\boldsymbol{e}_i\}_{i=1}^{8}$ be the standard basis for $\mathfrak{V}_8$, and $\bigotimes^j \mathfrak{V}$ the j-th tensor power of $\mathfrak{V}$.

Let $\mathfrak{B} \equiv \bigotimes^k \mathfrak{C}$ be 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 $\boldsymbol{e}_i \otimes \boldsymbol{e}_j$ for $1 \le i, j \le 8$. 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 [White 1975].

Each basis element of $\mathfrak{B}$ is of the form $\boldsymbol{c}_1 \otimes \boldsymbol{c}_2 \otimes \cdots \otimes \boldsymbol{c}_k$, where each $\boldsymbol{c}_s$ is some basis element of $\mathfrak{C}$. Since each $\boldsymbol{c}_s$ 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, 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 function from elements of $\mathfrak{B}$ to itself, and thereby becomes an element of $\mathfrak{M}_{64^k}$. (Technically, the unmove operators are only quasilinear, since the Boolean algebra is not a ring, and thus $\mathfrak{B}$ is not a module. However, we need not worry about this distinction.)

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

5.2. Group actions. We introduce a few group actions [Fulton and Harris 1991]. 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 [Merris 1980; 1981; 1992; Merris and Watkins 1983].

The symmetric group on k elements $\mathfrak{S}_k$ acts on $\mathfrak{B}$ by permuting the order of the factors: $\mathfrak{s}\bigotimes_{s=1}^{k}\boldsymbol{c}_s = \bigotimes_{s=1}^{k}\boldsymbol{c}_{\mathfrak{s}s}$, for $\mathfrak{s}\in\mathfrak{S}_k$ and $\boldsymbol{c}_s\in\mathfrak{C}$.

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

$$\mathfrak{r}(e_i\otimes e_j) = e_{8-j+1}\otimes e_i,$$
$$\mathfrak{f}(e_i\otimes e_j) = e_i\otimes e_{8-j+1}.$$

Thus, $\mathfrak{r}$ rotates the chessboard counterclockwise 90° and $\mathfrak{f}$ flips the chessboard about the horizontal bisector. $\mathfrak{D}_4$ acts diagonally on $\mathfrak{B}$ by

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

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: $(\mathfrak{g}M)\boldsymbol{v} = \mathfrak{g}(M\mathfrak{g}^{-1}(\boldsymbol{v}))$. 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 $\mathfrak{G}$ [Fulton and Harris 1991, 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

We now present the endgame algorithm using the notation developed in Section 5. Section 6.1 gives the fundamental factorization. Section 6.2 describes the modification of the equations of Table 1 to exploit symmetry. Section 6.3 describes the control structure of the algorithm.

6.1. Factoring the unmove operator. Recall from (3.1) 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

$$U_s \in \mathfrak{M}_{64^k} \equiv I_{64^{s-1}} \otimes (E_8\otimes I_8) \otimes I_{64^{k-s}}.$$

Such multidimensional shifts are commonly used in scientific computation.

$$
\begin{aligned}
Ш_{♖,s} &= \int_{\mathfrak{C}_4} LU_s(I_{64^k} + LU_s)^6 \\
Ш_{♘,s} &= L\int_{\mathfrak{D}_4} U_s \cdot (\mathfrak{r}(U_s^2)) \\
Ш_{♗,s} &= \int_{\mathfrak{D}_4} LU_s(I_{64^k} + LU_s\mathfrak{r}U_s)^6 \\
Ш_{♔,s} &= L\int_{\mathfrak{C}_4} U_s + U_s\mathfrak{r}U_s \\
Ш_{♕,s} &= Ш_{♖,s} + Ш_{♗,s}
\end{aligned}
$$

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.

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

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

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

Table 1 defines the piece-unmove operators. Figure 2 illustrates the computation of the integrand in the expression for $Ш_{♖,1}$ 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 (4.2) and (4.3), by varying the factorization, one can derive code suitable for different architectures. For example, if the interconnection architecture is a two-dimensional grid, only U_s for $s = 1$ can be directly computed. By using the relations $U_s = (1\,s)U_1$ and $Ш_{\mathbf{p},s} = (1\,s)Ш_{\mathbf{p},1}$, where $(1\,s) \in \mathfrak{S}_k$ interchanges 1 and s, equations appropriate for a grid architecture can be derived.

These equations are vectorizable as well [Smitley 1991]. The vectorized implementation of Table 1 by Burton Wendroff et al. has supported this claim [Wendroff et al. 1993].

Other factorizations appropriate for combined vector and parallel architectures, such as a parallel network of vector processors, can also be derived [Kaushik et al. 1993].

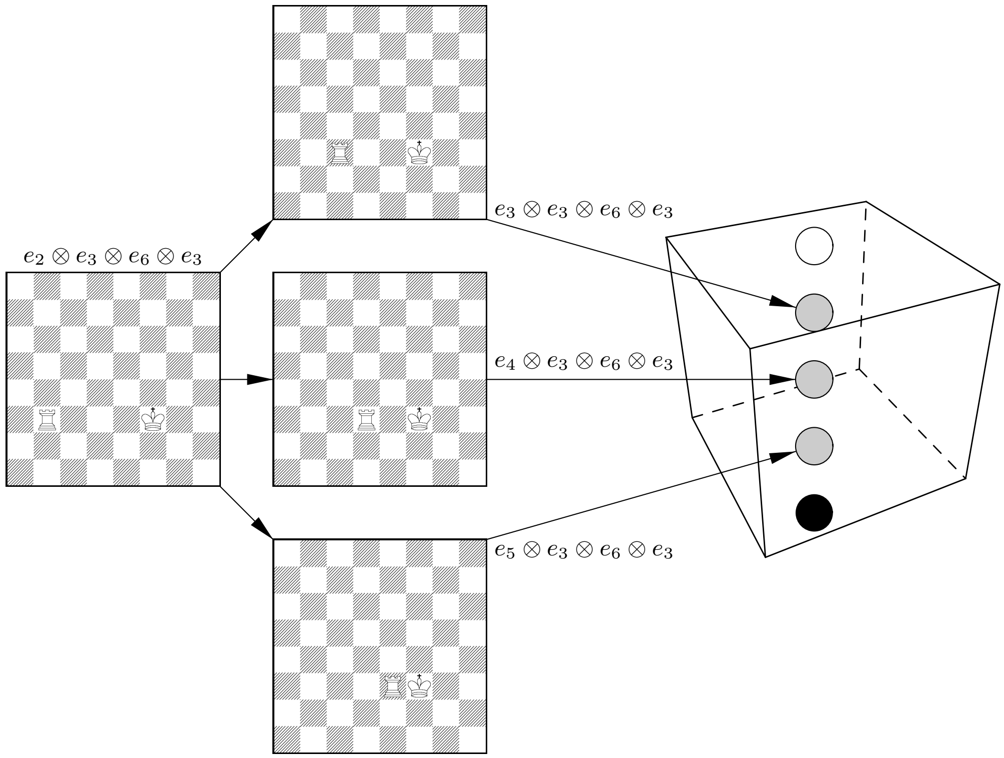
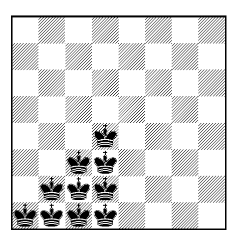


Figure 2. Unmoving the ♖ to the right from the position on the left results in the three positions shown in the center. Here, $S = \langle ♖, ♔ \rangle$. Each position corresponds to a point in the hyperboard, shown on the right. The position $\boldsymbol{e}_6 \otimes \boldsymbol{e}_3 \otimes \boldsymbol{e}_6 \otimes \boldsymbol{e}_3$ is illegal and is zeroed out by L.

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, as shown in Figure 3. These positions 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 bigger

Figure 3. The chessboard may be rotated 90° or reflected about any of its bisectors without altering the value of a pawnless position. Therefore, the ♚ may be restricted to lie in one of the ten squares shown.



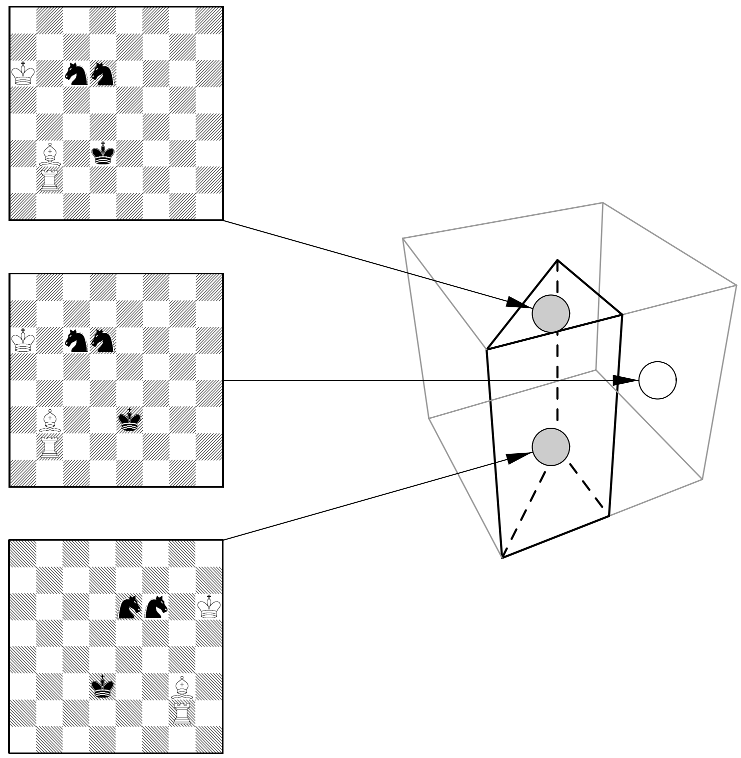


Figure 4. Only an eighth of the hyperboard is physically stored. When the ♚ is moved outside the squares marked in Figure 3, as in going from the top to the middle positions, we apply a symmetry transformation that puts the ♚ back in the allowed area; this is represented by the position at the bottom. 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 the top requires 222 moves against best play for White to win (see Table 2).

64^k-space $\mathfrak{B}$. By convention, the first piece of S, 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,

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

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

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 eight 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 [Stiller 1991a]. The actual

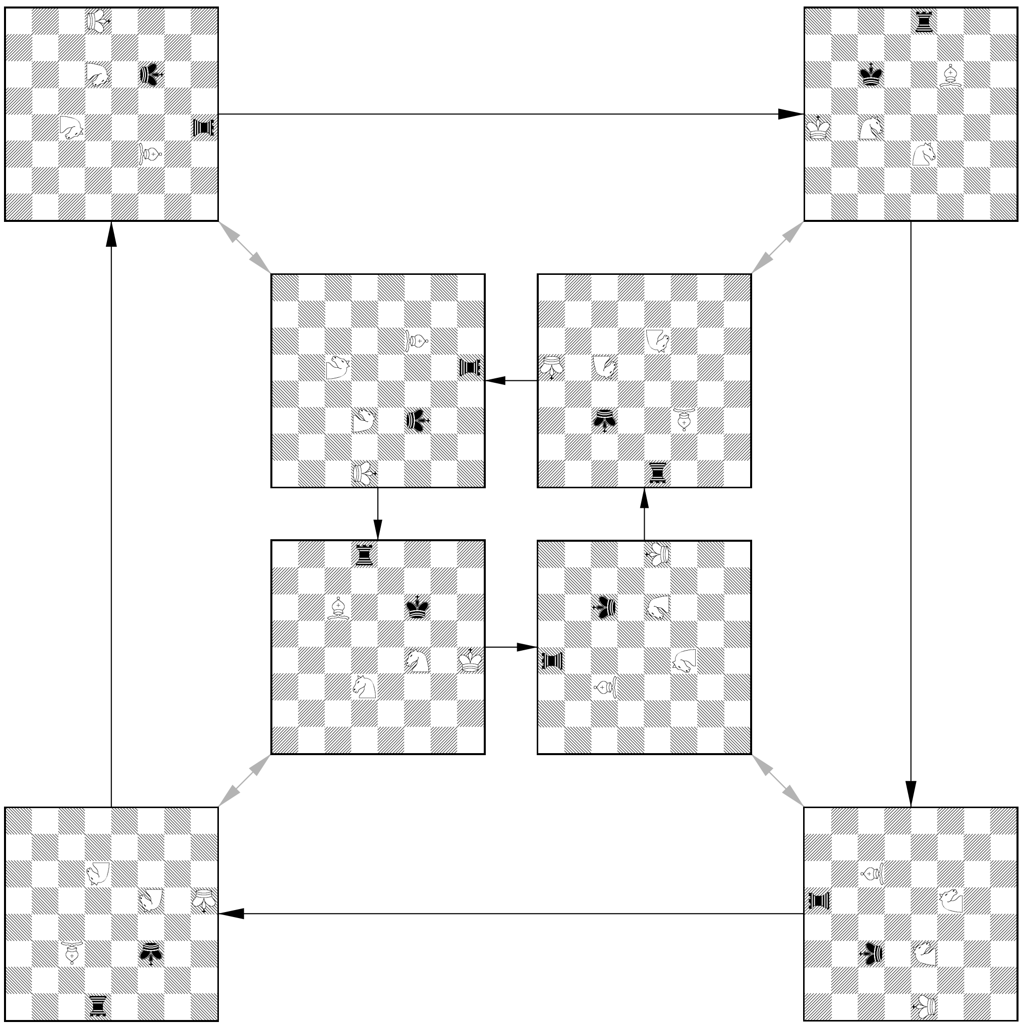


Figure 5. The Cayley graph for $\mathfrak{D}_4$. Each node is pictured by showing the effect of its corresponding transformation on a position in ♔♗♘♘♔♖; thus, the chess value of each of these nodes is the same. Black arrows correspond to $\mathfrak{r}$, and rotate the board counterclockwise 90°. Gray (diagonal) arrows 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.

communication pattern used is that of a group action graph, which looks like a number of disjoint copies of the Cayley graph, plus some cycles [White 1984].

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 [Hillis and Taylor 1990]. 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 [An et al. 1990; 1992; Tolimieri and An 1990].

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 j-th symmetric power of $\mathfrak{C}$ [Fulton and Harris 1991, 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 [Butler 1992; Sims 1971; Sims 1994]. The groups encountered here are so small, however, that computer-assisted group-theoretic computation was not required.

6.3. Control structure. For $i \geq 1$ we define $\boldsymbol{v}_i \in \mathfrak{B}$ 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, $\boldsymbol{v}_1$ is the vector of positions from which White can checkmate Black on the next move. $\boldsymbol{v}_2$ 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 $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, ... , until some i is reached for which $\boldsymbol{v}_i = \boldsymbol{v}_{i+1}$. Then $\boldsymbol{v} = \boldsymbol{v}_i$ 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 [Ströhlein 1970; Thompson 1986].

The method for computing $\boldsymbol{v}_i$ from $\boldsymbol{v}_{i-1}$ is called the backup rule. Several backup rules have been used in various domains [Schaeffer et al. 1992; Lake et al. 1994]. 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

$$\text{Ш}_{\text{White}} = \sum_{\{s\,:\,S_s \text{ is White}\}} \text{Ш}_{S_s,s},$$
$$\text{Ш}_{\text{Black}} = \sum_{\{s\,:\,S_s \text{ is Black}\}} \text{Ш}_{S_s,s}.$$

The backup rule used is $\boldsymbol{v}_{i+1} = \text{Ш}_{\text{White}}(\overline{\text{Ш}_{\text{Black}}(\overline{\boldsymbol{v}_i})})$, where a vertical bar denotes complement.

7. Refinements

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 [Bellman 1965]. 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 [Thompson 1986]. 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 [Adams et al. 1992; Stiller 1992a]. 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, again implemented 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 [Zermelo 1913]. 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 [Rasmussen 1988].

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 fifty-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 five-piece endgames, the results presented here use the more conservative distance-to-win metric.

The *max-to-win* for a set of pieces 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 two-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 [Annexstein 1990; Rosenberg 1988; Draper 1990; 1991] this motif will reappear on other implementations.

8. Chess Results

The combinatorially possible pawnless five-piece games and many five-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 [Stiller 1989]. Some endgames were solved under the distance-to-mate metric as well. The distance-to-mate results were not very illuminating.

Later, several pawnless six-piece endgames were also solved. Table 2 presents statistical information about these six-piece endgames. For most endgames, we considered $6,185,385,360 = 462 \cdot 62 \cdot 61 \cdot 60 \cdot 59$ positions, a number explained as follows: there are 462 arrangements of two nonadjacent kings, modulo the dihedral symmetry; for each such arrangement, the remaining four pieces are placed in any available square. Thus the state space per endgame has about $6.2 \cdot 10^9$ nodes, although the size of each hyperboard is $462 \cdot 64^4 \approx 7.7 \cdot 10^9$ nodes.

Note that this inflates the statistics for wins, because of the advantage of the first move in a random position: White is already won if the ♚ is in check, or it may be able to capture a piece in one move. When there are repeated pieces, the aggregate statistics (though not the percentages) are also inflated, because games arising from permutations of the identical pieces are counted as different. Thus, for example, there is really only a single mutual zugzwang in ♔♖♖♖♚♛, but it is counted six times.

The max-to-win values were significantly higher than previously known endgames. No five-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.

(The phrase "general win" is not susceptible to precise definition. It applies to any class of games, like ♔♕♘♔♘, where all "normal" starting configurations lead to a win, although some special configurations may not be wins for obvious reasons, such as the immediate loss of a piece by White due to a fork.)

We remark that ♔♖♗♔♘♘ is a general win, with 223 moves required to win in the worst case. Roycroft, a leading endgame expert, said in 1972 that this

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

Table 2. Statistics gathered for six-piece endgames: description of endgame, number and percentage of positions that are wins for White, maximum D of distance-to-win over all positions that are wins for White, and number Z of mutual zugzwangs. Bishops linked with a bar are constrained to lie on squares of opposite colors; this reduces the state space roughly by a factor of four, from 6.19×10^9 to 1.67×10^9 nodes.

configuration was "known to be a draw", while ♔♗♗♔♘, which was considered a draw by most players, he called only "controversial or unknown". Most of the standard works concurred with the opinion that ♔♖♗♔♘♘ was not a general win [Euwe 1940, Vol. 5, pp. 50–53; Chéron 1952, p. 417; Berger 1890, pp. 167–169; Fine 1941, p. 521]. Chéron, however, seems to reserve judgment.

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

A mutual zugzwang is closely related to a game whose Conway value is zero: 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* [Berlekamp et al. 1982].

Unlike the "maximin" positions such as the one in Table 3, whose analysis is fairly impenetrable, mutual zugzwangs can sometimes be understood by humans.

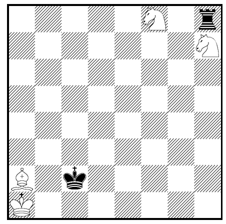


Figure 6. Mutual zugzwang: White to play draws, but Black to play loses.

Figure 6 shows an example, discovered by the program. The ♖ is trapped on h8, since g8 is guarded by the ♗ on a2, and the ♘'s guard each other. If the Black ♖ were to capture a ♘, it would 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 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.

Now consider Figure 7, left. 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. This mutual zugzwang, discovered by the computer, is somewhat inelegant in that includes promoted pieces. Noam Elkies used it as the basis for a much more elegant composition, one that follows accepted aesthetic practice by avoiding the use of promoted pieces in the original position and by appearing more natural. This composition [Elkies 1993; Rusinek 1994, #546] is shown in Figure 7, right. We quote from Elkies' analysis: "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+

Figure 7. Left: mutual zugzwang found by the program. Right: Position constructed by Noam Elkies [1993], leading to a version of the position on the left. White to play and win.

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 [a rotated version of the ♔♕♕♔♕♕ mutual zugzwang]."

Other analyses of mutual zugzwang can be found in [Elkies 1996] in this book.

Pawnless six-piece endgames are extremely rare in tournament play, but they do arise sometimes. During an elite tournament in Tilburg, a game between Anatoly Karpov and Gary Kasparov reached the position shown in Figure 5. After another fifty moves a draw was reached, but it was far from clear that a win could not be achieved given unlimited play. An exhaustive analysis by the six-piece program [Stiller 1991b] proved that it couldn't.

We conclude this section displaying the best play line for the ♔♖♘♔♘♘ position with maximal distance-to-win, namely 243. Recall that this means it takes 243 moves against optimal Black play in order for White to be able to safely capture a piece, thereby ensuring an easy win. The initial position is shown in Figure 8, and the line of play in Table 3.

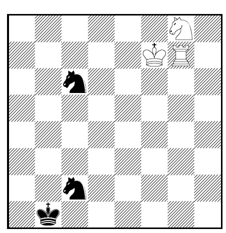


Figure 8. Starting point for the longest ♔♖♘♔♘♘ fight. See Table 3.

	moves 1–40		moves 41–80		moves 81–120		moves 121–160	
1	♔f7-e6	♘c6-b4	♔e5-d4	♘b4-a6	♔d6-e5	♘f2-g4	♘d6-e4	♘e7-g6
2	♔e6-e5	♘b4-d3	♖c3-c2	♔c6-d7	♔e5-d4	♘h3-f4	♔e5-f5	♘g6-f8
3	♔e5-e4	♘d3-f2	♘c8-b6	♔d7-d6	♘c5-e4	♔g5-g6	♖h7-h6	♔c6-c7
4	♔e4-f3	♘f2-d3	♘b6-c4	♔d6-c6	♖a8-a6	♔g6-f5	♖h6-h1	♘f8-d7
5	♔f3-e2	♘c2-b4	♘c4-e3	♔c6-d6	♖a6-a5	♔f5-e6	♖h1-b1	♘d7-b8
6	♔e2-e3	♔b1-b2	♘e3-f5	♔d6-e6	♘e4-c5	♔e6-e7	♔f5-e5	♘d5-e3
7	♔e3-d4	♘d3-f4	♘f5-g7	♔e6-f7	♖a5-a7	♔e7-f6	♔e5-d4	♘e3-f5
8	♔d4-c4	♘b4-d5	♘g7-h5	♘c5-e6	♔d4-e4	♔f6-g5	♔d4-d5	♘f5-e3
9	♖g7-h7	♘d5-e3	♔d4-e5	♘a6-b4	♖a7-a5	♘f4-h5	♔d5-c5	♘b8-d7
10	♔c4-d4	♘e3-c2	♖c2-e2	♘b4-d3	♘c5-e6	♔g5-g6	♔c5-d4	♘e3-g4
11	♔d4-e4	♘f4-e6	♔e5-e4	♘d3-b4	♖a5-b5	♔g6-f7	♖b1-c1	♔c7-d8
12	♔e4-e5	♘e6-g5	♖e2-b2	♔f7-g6	♘e6-c5	♔f7-e7	♖c1-e1	♘g4-f6
13	♖h7-h5	♘c2-e1	♘h5-g3	♘e6-g5	♖b5-b2	♔e7-d6	♘e4-g5	♔d8-c7
14	♔e5-f5	♘g5-f3	♔e4-d4	♘g5-e6	♘c5-b7	♔d6-e7	♘g5-f7	♘d7-f8
15	♔f5-e4	♘f3-d2	♔d4-c4	♘b4-a6	♖b2-a2	♘h5-g7	♖e1-f1	♘f6-g4
16	♔e4-e3	♘d2-b3	♖b2-f2	♘e6-g5	♖a2-e2	♔e7-d7	♖f1-g1	♘g4-f6
17	♖h5-h1	♘e1-c2	♖f2-f1	♘a6-c7	♖e2-g2	♘g7-e8	♖g1-e1	♔c7-d7
18	♔e3-d3	♘b3-c1	♘g3-e2	♘g5-f7	♔e4-f4	♘g4-f6	♔d4-e5	♘f6-e8
19	♔d3-e4	♘c1-b3	♘e2-f4	♔g6-g5	♔f4-e5	♔d7-e7	♘f7-h8	♔d7-e7
20	♖h1-h3	♘b3-c5	♔c4-d4	♘c7-b5	♖g2-e2	♔e7-d7	♔e5-d5	♔e7-d7
21	♔e4-e5	♘c2-e1	♔d4-c5	♘b5-d6	♘b7-a5	♘f6-g4	♖e1-f1	♘e8-c7
22	♘g8-f6	♘e1-d3	♘f4-e6	♔g5-g6	♔e5-f5	♘g4-h6	♔d5-e5	♘f8-e6
23	♔e5-d6	♘c5-b7	♘e6-f8	♔g6-g5	♔f5-g6	♘h6-g8	♘h8-g6	♘e6-c5
24	♔d6-c7	♘b7-c5	♔c5-d5	♘d6-f5	♘a5-c4	♘e8-c7	♖f1-b1	♔d7-c6
25	♔c7-c6	♔b2-c2	♖f1-b1	♘f5-g3	♔g6-f7	♘g8-h6	♘g6-e7	♔c6-d7
26	♖h3-h2	♔c2-b3	♖b1-b7	♘f7-h6	♔f7-f6	♘h6-g8	♘e7-f5	♔d7-c6
27	♔c6-d5	♔b3-b4	♖b7-g7	♔g5-f4	♔f6-e5	♘g8-e7	♘f5-d4	♔c6-d7
28	♔d5-d4	♘d3-f4	♘f8-e6	♔f4-f3	♖e2-d2	♔d7-c6	♖b1-d1	♘c7-a6
29	♖h2-h4	♔b4-b5	♖g7-b7	♘g3-h5	♖d2-c2	♘c7-a6	♘d4-f5	♔d7-c6
30	♘f6-e8	♘c5-b3	♖b7-b4	♘h5-f6	♘c4-e3	♔c6-d7	♖d1-h1	♘a6-b4
31	♔d4-e4	♘f4-g6	♔d5-d4	♘f6-h5	♖c2-d2	♔d7-c6	♖h1-h6	♔c6-d7
32	♖h4-h7	♘b3-c5	♔d4-d3	♘h6-g4	♖d2-d6	♔c6-b5	♔e5-d4	♘c5-e6
33	♔e4-d4	♘g6-f4	♘e6-g5	♔f3-g3	♖d6-h6	♘e7-c8	♔d4-c4	♘b4-a6
34	♘e8-d6	♔b5-c6	♘g5-e4	♔g3-h4	♔e5-d4	♘a6-b4	♖h6-h7	♔d7-c6
35	♖h7-h6	♘c5-b3	♖b4-a4	♘h5-f4	♖h6-h5	♔b5-c6	♖h7-h1	♘a6-c7
36	♔d4-e4	♘f4-e6	♔d3-d4	♘f4-e6	♘e3-c4	♘c8-e7	♖h1-d1	♘c7-e8
37	♔e4-e5	♘e6-d4	♔d4-d5	♘e6-f4	♖h5-h6	♔c6-c7	♘f5-e7	♔c6-c7
38	♖h6-h3	♘b3-c5	♔d5-d6	♘f4-h3	♖h6-h7	♔c7-d7	♔c4-d5	♘e6-f8
39	♘d6-c8	♘d4-c2	♖a4-a8	♘g4-f2	♔d4-e5	♘b4-d5	♘e7-g8	♔c7-d7
40	♖h3-c3	♘c2-b4	♘e4-c5	♔h4-g5	♘c4-d6	♔d7-c6	♔d5-c5	♔d7-e6

Table 3. An optimal line of play for the position of Figure 8. Occasional local variations are possible (for instance, 15. f5-f4), but are not shown.

	moves 161–180		moves 181–200		moves 201–220		moves 221–243	
161	♖d1-e1	♔e6-d7	♖d5-a5	♘h6-g4	♘h6-f5	♔f8-f7	♔d5-d6	♘c7-e8
162	♖e1-e7	♔d7-d8	♔e5-d4	♔f8-f7	♖e1-e2	♘d5-b6	♔d6-e7	♘e8-c7
163	♖e7-a7	♘f8-d7	♖a5-a7	♔f7-f6	♖e2-e7	♔f7-f8	♖h7-h6	♘a6-b8
164	♔c5-c6	♘d7-e5	♔d4-e4	♘g7-e8	♖e7-e1	♘b6-d5	♘a4-b6	♔c8-b7
165	♔c6-d5	♘e5-g6	♖a7-a6	♔f6-g7	♖e1-e5	♘d5-b6	♘b6-c4	♘b8-c6
166	♖a7-h7	♘e8-c7	♖a6-b6	♘g4-f6	♔g5-g6	♘e8-c7	♔e7-d6	♘c6-b4
167	♔d5-c6	♘g6-e5	♔e4-f5	♘f6-d7	♘f5-d6	♘b6-d5	♖h6-h8	♘b4-a6
168	♔c6-d6	♘e5-c4	♘f4-e6	♔g7-f7	♖e5-e1	♘c7-e6	♖h8-h7	♔b7-c8
169	♔d6-c5	♘c4-e5	♘e6-g5	♔f7-f8	♔g6-f5	♘e6-c7	♘c4-a5	♔c8-d8
170	♖h7-h5	♘e5-f7	♖b6-a6	♘e8-g7	♔f5-e5	♘d5-b4	♘a5-c6	♔d8-c8
171	♔c5-c6	♘c7-e6	♔f5-g6	♘d7-e5	♖e1-f1	♔f8-e7	♘c6-e7	♔c8-d8
172	♖h5-a5	♔d8-e8	♔g6-h7	♘g7-e8	♖f1-f7	♔e7-d8	♘e7-d5	♘c7-e8
173	♘g8-f6	♔e8-e7	♖a6-e6	♘e5-f7	♘d6-b7	♔d8-c8	♔d6-c6	♘a6-b8
174	♘f6-d5	♔e7-f8	♘g5-f3	♘f7-d6	♘b7-c5	♘c7-b5	♔c6-b5	♘e8-d6
175	♔c6-d7	♘e6-d4	♔h7-g6	♘d6-f5	♖f7-g7	♔c8-d8	♔b5-c5	♘d6-c8
176	♘d5-f4	♘f7-h6	♖e6-e1	♘f5-e7	♖g7-b7	♘b4-c6	♖h7-h8	♔d8-d7
177	♖a5-d5	♘d4-f5	♔g6-g5	♔f8-f7	♔e5-e6	♔d8-c8	♘d5-f6	♔d7-c7
178	♔d7-e6	♘f5-g7	♘f3-e5	♔f7-g7	♖b7-h7	♘c6-b4	♖h8-h7	♔c7-d8
179	♔e6-f6	♘h6-g8	♘e5-g4	♔g7-f8	♘c5-a4	♘b4-a6	♖h7-b7	♘b8-a6
180	♔f6-e5	♘g8-h6	♘g4-h6	♘e7-d5	♔e6-d5	♘b5-c7	♔c5-c6	♘c8-e7
							♔c6-b6	♘a6-b4
							♖b7-d7	♔d8-c8
							♖d7×♘e7	

Table 3 (continuation)

9. Implementation Notes and Run Times

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 six-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 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 6000 than timings *of different endgames* reported using classical techniques [van den Herik and Herschberg 1985b; Thompson 1986; Nefkens 1985; van den Herik et al. 1988] based on the five-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 700 times slower than that of the code reported in this paper (depending on the endgame) [Thompson 1990].

Unfortunately, direct comparison of six-piece timing against other work is, of course, not currently possible since six-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, six-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 the vectorized implementation of Table 1 by Burton Wendroff et al. [1993], although this implementation currently solves only a single four-piece endgame.

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

10. 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 [1983, 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 Section 2.1. 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 [Schwarz 1925].

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 seven-piece endgames. In fact, in my opinion a more promising approach would be to follow up on the suggestions first made by Bellman [Bellman 1961; Bellman 1965; Berliner and Campbell 1984] 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.

Acknowledgments

Thanks go to my advisor, Simon Kasif, for his help in the formulation and presentation of these ideas.

Throughout the six-piece project, Noam Elkies patiently provided invaluable chess advice and suggestions, including the mutual zugzwang recognition 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 six-piece endgames arose.

I am also grateful to the following people 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.

Silvio Levy skillfully and scrupulously edited the manuscript.

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 my 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. Thanks to 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 them 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.

References

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

[Adams et al. 1992] 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, New York, 1992.

[‘Adlī] اَلعَدلِي, كِتَاب الشَطرَنج [Al-‘Adlī, *Book of Chess*], ninth century, photographic copy of Arabic manuscript (available at the John G. White Collection; see Acknowledgements).

[Amelung 1893] Friedrich Ludwig Amelung, "Das Endspiel von zwei Offizieren gegen einen Springer" [The endgame of two pieces against a rook], *Baltische Schachblätter* **4** (1893), 290–297.

[Amelung 1898] Friedrich Ludwig Amelung, "Auszüge aus den Briefen von A. Ascharin an F. Amelung" [Excerpts from the letters of A. Ascharin to F. Amelung], *Baltische Schachblätter* **5** (1898), 12–38.

[Amelung 1900] Friedrich Ludwig Amelung, "Das Endspiel von Thurm gegen Springer" [The endgame of rook against knight], *Deutsche Schachzeitung*, **55** (1900) 1–5 (January), 37–41 (February), 101–105 (April), 134–138 (May), 198–202 (July), 261–266 (September).

[Amelung 1901] Friedrich Ludwig Amelung, "Das Endspiel von Thurm und Springer gegen die Dame" [The endgame of rook and knight against queen], *Deutsche Schachzeitung*, **56** (1901) 193–197 (July), 225–229 (August).

[Amelung 1902] 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** (1902) 265–268 (September), 297–300 (October), 330–332 (November).

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

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

[Amelung 1908c] Friedrich Ludwig Amelung, Lösungspreis-endspiel Nr. 178 [Prize-endgame 178], *Für Haus und Familie* (supplement to *Düna-Zeitung*) **63** (15–28 March 1908), 87. Solution in **13** (17–30 January 1909), 20–21.

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

[An et al. 1992] 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* A**48** (1992), 346–349.

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

[Anonymous] [Nicholas de St Nicholai?], "*Bonus Socius* [Good companion], 13th century.

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

[Averbakh 1982] Yuri Lvovich Averbakh, Шахматные окончания; ферзевые [*Chess endings: queen*], Fizkul'tura i sport, Moscow (second edition), 1982.

[Averbuch et al. 1990] Amir Averbuch, Eran Gabber, Boaz Gordissky, and Yoav Medan, "A parallel FFT on a MIMD machine", *Parallel Computing*, **15** (1990), 61–74.

[Babbage 1864] Charles Babbage, "Games of skill", pp. 465–471 in *Passages From the Life of a Philosopher*, Longman and Green, London, 1864.

[Backhouse and Carré 1975] 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) (1975), 161–186.

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

[Balt. Schachbl. 1902] [Carl Behting and Paul Kerkovius?] "Das zweite Baltische Schachturnier" [The second Baltic chess tournament], *Baltische Schachblätter* **9** (1902), 1–24.

[Bashmakova 1991] J. G. Bashmakova, "Fedor Eduardovich Molin", pp. 1739–1740 in *Biographical Dictionary of Mathematicians*, Vol. 3, Scribner's, New York, 1991.

[Bell 1978] Alex G. Bell, *The Machine Plays Chess?*, Pergamon Chess Series, Pergamon Press, Oxford, 1978.

[Bellman 1954] 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.

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

[Bellman 1961] 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.

[Bellman 1965] 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** (1965), 244–246.

[Berger 1890] Johann Berger, *Theorie und Praxis der Endspiele: Ein Handbuch für Schachfreunde* [*Theory and practice of the endgame: a handbook for chessplayers*], Veit, Leipzig, 1890.

[Berger 1922] Johann Berger, *Theorie und Praxis der Endspiele: Ein Handbuch für Schachfreunde* [*Theory and practice of the endgame: a handbook for chessplayers*], de Gruyter, Berlin and Leipzig (second edition), 1922.

[Berlekamp et al. 1982] Elwyn Ralph Berlekamp, John Horton Conway, and Richard K. Guy, *Winning Ways for Your Mathematical Plays* I, Academic Press, London, 1982.

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

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

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

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

[Campbell 1988] Murray S. Campbell, "Chunking as an abstraction mechanism", Technical Report CMU-CS-88-116, Carnegie-Mellon University, 22 February 1988.

[Carrera 1617] Pietro Carrera, *Del gioco de gli scacchi* [*T he game of chess*], volume 3, de Rossi[?], Trento[?], 1617.

[Chamberlain 1988] R. M. Chamberlain, "Gray codes, fast Fourier transforms, and hypercubes", *Parallel Computing* **6** (1988), 225–233.

[Chapais 1780] Chapais, *Essais analytiques sur les échecs, avec figures* [*Analytical essay on chess, with illustrations*], [ca. 1780].

[Chéron 1952] André Chéron, *Nouveau traité complet d'échecs: La fin de partie* [*New complete treatise of chess: The endgame*], Yves Demailly, Lille, France, 1952.

[Clausen and Baum 1993] Michael Clausen and Ulrich Baum, *Fast Fourier Transforms*, "Bibliographisches Institut Wissenschaftsverlag, Mannheim, 1993.

[Cooley and Tukey 1965] James William Cooley and John Wilder Tukey, "An algorithm for the machine calculation of complex Fourier series", *Mathematics of Computation* **19** (1965), 297–301.

[Crosskill 1864] [Alfred Crosskill], "The rook and bishop against rook", *The Chess-Player's Magazine* **2** (1864), 305–311.

[Dekker et al. 1987] Sito T. Dekker, H. Jaap van den Herik, and I. S. Herschberg, "Complexity starts at five", *International Computer Chess Association Journal* **10**(3) (September 1987), 125–138.

[Diaconis 1988] Persi Diaconis, *Group Representations in Probability and Statistics*, Lecture Notes—Monograph Series **11**, Institute of Mathematical Statistics, Hayward, CA, 1988.

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

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

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

[Düna-Zeitung 1909a] Announcement of Amelung's death in *Für Haus und Familie* (supplement to *Düna-Zeitung*) (10–23 March 1909), 52.

[Düna-Zeitung 1909b] "Friedrich Ludwig Amelung", *Für Haus und Familie* (supplement to *Düna-Zeitung*) **70** (28 March–10 April 1909), 76–78.

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

[Elkies 1996] Noam David Elkies, "On numbers and endgames: Combinatorial game theory in chess endgames", pp. 135–150 in this volume.

[Euwe 1940] Machgielis Euwe, *Het Eindspel* [*The endgame*], G. B. van Goor Zonen, 's-Gravenhage, 1940. German translation: *Das Endspiel*, Das Schach-Archiv, F. L. Rattman, Hamburg, 1957.

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

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

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

[García and Stiller 1993] 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** (1993), 1396–1406.

[Gasser 1991] Ralph Gasser, "Applying retrograde analysis to Nine Men's Morris", pp. 161–173 in *Heuristic Programming in Artificial Intelligence: The Second Computer Olympiad* (edited by D. N. L. Levy and D. F. Beal), Ellis Horwood, Chichester, England, 1991.

[Gasser 1996] Ralph Gasser, "Solving Nine Men's Morris", pp. 101–113 in this volume.

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

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

[Granata and Tolimieri 1991] 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** (1991), 289–90.

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

[Granata et al. 1992] 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** (1992), 40–48.

[Greco 1624] 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.

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

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

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

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

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

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

[van den Herik et al. 1988] 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) (March 1988), 26–31.

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

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

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

[Hooper and Whyld 1992] D. Hooper, K. Whyld, *The Oxford Companion to Chess* (second edition), Oxford Univ. Press, 1992.

[Huberman 1968] 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.

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

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

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

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

[Johnson et al. 1990] 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** (1990), 449–500.

[Johnson 1989] Robert W. Johnson, "Automatic implementation of tensor products", manuscript, 24 April 1989.

[Johnson et al. 1991] Robert W. Johnson, Chua-Huang Huang, and Jeremey R. Johnson, "Multilinear algebra and parallel programming", *Journal of Supercomputing* **5** (1991), 189–217.

[Kanunov 1983] Nikolai Fedorovich Kanunov, Федор Эдуардович Молин 1861–1941 [*Fyodor Eduardovich Molin* 1861–1941], Nauka, Moscow, 1983.

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

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

[Kling and Horwitz 1851] 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.

[Komissarchik and Futer 1974] E. A. Komissarchik and Aaron L. Futer, "Об анализе ферзевого эндшпиля при помощи ЭВМ [Analysis of a queen endgame using an IBM computer], Problemy Kibernetiki **29** (1974), 211–220.

[Kopec et al. 1988] Danny Kopec, Brent Libby, and Chris Cook, "The endgame king, rook and bishop vs. king and rook (KRBKR)", pp. 60–61 in *Proceedings: 1988 Spring Symposium Series: Computer Game Playing* (Stanford, 1988), American Association for Artificial Intelligence.

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

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

[Lake et al. 1994] Robert Lake, Jonathan Schaeffer, and Paul Lu, "Solving large retrograde analysis problems using a network of workstations", pp. 135–162 in *Advances in Computer Chess* 7 (edited by H. J. van den Herik et al.), University of Limburg, Maastricht, Netherlands, 1994.

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

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

[Leiserson et al. 1992] Charles Eric Leiserson et al., "The network architecture of the CM-5", pp. 272–285 in *Proceedings of the 4th Annual ACM Symposium on Parallel Algorithms and Architectures* (San Diego, 1992), ACM Press, New York, 1992.

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

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

[Lopez 1561] 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.

[Lucena 1497?] 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, 1953.

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

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

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

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

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

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

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

[Molien 1893] Theodor Molien, "Ueber Systeme höherer complexer Zahlen" [On systems of higher complex numbers], *Mathematische Annalen*, **41** (1893), 83–156; errata in **42** (1893), 308–312.

[Molien 1897a] Theodor Molien, "Eine Bemerkung zur Theorie der homogenen Substitutionsgruppen" [A remark on the theory of homogeneous substitution groups], *Sitzungberichte Naturforscher-Gesellschaft Dorpat* (Yurev, Estonia) **11** (1897), 259–274.

[Molien 1897b] Theodor Molien, "Über die Anzahl der Variabeln einer irreductibelen Substitutionsgruppen" [On the number of variables of irreducible substitution groups], *Sitzungsberichte der ber Naturforscher-Gesellschaft Dorpat* (Yurev, Estonia) **11** (1897), 277–288.

[Molien 1898a] Theodor Molien, pp. 208–209 in "Sammlung baltischer Schachprobleme und Endspiele aus den Jahren 1890 bis 1897", *Baltische Schachblätter* **6** (1898), 179–212.

[Molien 1898b] Theodor Molien, "Über die Invarianten der linearen Substitutionsgruppen" [On the invariants of linear substitution groups], *Sitzungsberichte Akademie der Wissenschaft. Berlin* (1898), 1152–1156.

[Molien 1900] Theodor Molien, Partie 73 [score of game against W. Sohn], "*Baltische Schachblätter* **7** (1900), 346.

[Molien 1901] 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 Schachblätter* **8** (1901), page 72.

[Mullin 1988] Lenore M. Restifo Mullin, "A mathematics of arrays", Technical Report 8814, CASE Center, Syracuse University, December 1988.

[Murray 1913] Harold James Ruthven Murray, *A History of Chess*, Oxford University Press, London, 1913.

[Nassimi and Sahni 1982] 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** (1982), 642–667.

[Nefkens 1985] Harry J. Nefkens, "Constructing data bases to fit a microcomputer", *International Computer Chess Association Journal* **8**(4) (December 1985), 217–224.

[von Neumann and Morgenstern 1944] John von Neumann and Oskar Morgenstern, *Theory of Games and Economic Behavior*, Wiley, New York, 1944.

[Nunn 1992] John Nunn, *Secrets of Rook Endings*, Batsford Chess Library, H. Holt, New York, 1992.

[Nunn 1994] John Nunn, *Secrets of Pawnless Endings*, Batsford Chess Library, H. Holt, New York, 1994.

[Patashnik 1980] Oren Patashnik, "Qubic: $4 \times 4 \times 4$ tic-tac-toe", *Math. Mag.* **53**(4) (September 1980), 202–216.

[Pease 1968] Marshall Carleton Pease, "An adaptation of the fast Fourier transform for parallel processing", *Journal of the Association for Computing Machinery* **15** (1968), 252–264.

[Perez 1929] Juan Bautista Sanchez Perez, *El Ajedrez de D. Alfonso el Sabio* [*The chess of D. Alfonso the Wise*], Franco-Española, Madrid, 1929.

[Philidor 1749?] 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 perfecting oneself in a short time in this noble game*], London, [1749?].

[Quinlan 1979] John Ross Quinlan, "Discovering rules by induction from large collections of examples", pp. 168–201 in *Expert systems in the micro-electronic age* (edited by Donald Michie), Edinburgh University Press, 1979.

[Quinlan 1983] John Ross Quinlan, "Learning efficient classification procedures and their application to chess end games", pp. 463–482 in *Machine Learning: An Artificial Intelligence Approach* (edited by Ryszard Stanislaw Michalski et al.), Tioga, Palo Alto, CA, 1983.

[Rasmussen 1988] Lars Rasmussen, "Ultimates in KQKR and KQKN", *International Computer Chess Association Journal* **11**(1) (March 1988), 21–25.

[Rinck 1950] Henri Rinck, *1414 Fins de partie* [*1414 endgames*], La Academica, Barcelona, 1950.

[Rinck and Malpas 1947] Henri Rinck and Louis Malpas, *Dame contre tour et cavalier* [*Queen against rook and knight*], L'Echiquier, Bruxelles, 1947.

[Rosenberg 1988] Arnold Leonard Rosenberg, "Shuffle-oriented interconnection networks", Technical Report COINS 88-84, Computer and Information Science Department, University of Massachusetts at Amherst, 11 October 1988.

[Roycroft 1967] Arthur John Roycroft, "A note on 2S's v R + B", *EG* (ARVES, Amsterdam) **1** (April 1967), 197–198.

[Roycroft 1972] Arthur John Roycroft, *Test Tube Chess: A Comprehensive Introduction to the Chess Endgame Study*, Faber and Faber, London, 1972.

[Roycroft 1984] Arthur John Roycroft, “Two bishops against knight”, *EG* (ARVES, Amsterdam) **5** (#75, April 1984), 249.

[Roycroft 1984] 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) (September 1984), 164–170.

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

[Salvio 1634] 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.

[Schaeffer et al. 1992] Jonathan Schaeffer, Joseph Culberson, Norman Treloar, Brent Knight, Paul Lu, and Duane Szafron, “A World Championship caliber checkers program”, *Artificial Intelligence* **53** (1992), 273–290.

[Schaeffer and Lake 1996] Jonathan Schaeffer and Robert Lake, “Solving the Game of Checkers”, pp. 119–133 in this volume.

[Schwarz 1925] K. H. Schwarz, “Versuch eines mathematischen Schachprinzips” [An attempt toward a mathematical principle for chess], *Deutsche Schachzeitung* **53**(11) (November 1925), 321–324.

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

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

[Sims 1971] Charles Coffin Sims, “Computations with permutation groups”, pp. 23–28 in *Proceedings of the Second Symposium on Symbolic and Algebraic Manipulation*, Los Angeles, 1971 (edited by Stanley Roy Petrick), ACM Press, New York 1971.

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

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

[Skillicorn 1993] David Benson Skillicorn, “Structuring data parallelism using categorical data types”, pp. 110–115 in *Proceedings of the 1993 Workshop on Programming Models for Massively Parallel Computers*, Berlin, 1993, IEEE Computer Society Press, Los Alamitos, CA, 1993.

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

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

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

[Stiller 1991a] Lewis Benjamin Stiller, “Group graphs and computational symmetry on massively parallel architecture”, *Journal of Supercomputing* **5** (1991), 99–117.

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

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

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

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

[Stiller 1995] Lewis Benjamin Stiller, "Exploiting Symmetry on Parallel Architectures", Ph.D. thesis, Department of Computer Science, The Johns Hopkins University, Baltimore, 1995.

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

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

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

[Szymanski et al. 1994] 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?" pp. 299–304 in *Proceedings of the 1994 International Conference on Computer Languages*, Toulouse, IEEE Computer Society Press, Los Alamitos, CA, 1994.

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

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

[Thompson 1983] [Kenneth Lane Thompson], *EG* (ARVES, Amsterdam) **5**(74) (November 1983).

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

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

[Tolimieri and An 1990] Richard Tolimieri and Myoung An, "Computations in X-ray crystallography", pp. 237–250 of *Recent advances in Fourier analysis and its applications* (edited by J. S. Byrnes and Jennifer L. Byrnes) NATO ASI Series C: Mathematical and Physical Sciences **315**, Kluwer, Dordrecht and Boston, 1990.

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

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

[Torres-Quevedo 1951] Gonzalo Torres-Quevedo, "Présentation des appareils de Leonardo Torres-Quevedo" [Presentation of the machines of Leonardo Torres-Quevedo], pp. 383–406 in *Les machines à calculer et la pensée humaine*, Paris, 1951, Colloques internationaux du Centre National de la Recherche Scientifique **37**, Publications du CNRS, Paris, 1953.

[Troitskiĭ 1934] Alekseii Alekseevich Troitskiĭ, "Два коня против пешек (теоретический очерк)" [Two knights against pawn (theoretical essay)], pp. 248–288 in Сборник шахматных этюдов [*Collection of chess studies*], Fizkul'tura i turizm, Moscow, 1934.

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

[Valiant 1990b] Leslie G. Valiant, "General purpose parallel architectures", pp. 945–971 in *Handbook of Theoretical Computer Science: Algorithms and Complexity*, volume A (edited by Jan van Leeuwen), Elsevier, Amsterdam, 1990.

[van, von ...] See under last name.

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

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

[Wendroff et al. 1993] 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** (1993), 285–295.

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

[White 1975] Dennis Edward White, "Multilinear enumerative techniques, "*Linear and Multilinear Algebra* **2** (1975), 341–352.

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

[Zermelo 1913] Ernst Zermelo, "Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels" [On an application of set theory to the theory of playing chess], pp. 501–504 in *Proceedings of the Fifth International Congress of Mathematicians*, Cambridge, 1912, vol. 2 (edited by Ernest William Hobson and Augustus Edward Hough Love), Cambridge University Press, England, 1913.

Lewis Stiller
Department of Computer Science
The Johns Hopkins University
Baltimore, MD 21218
stiller@symmetry.xo.com