Computation and Information Theory of Chess Games

Chun-Kai Hwang, John Reuben Gilbert, Tsung-Ren Huang, Chen-An Tsai, and Yen-Jen Oyang

***Abstract*—We provide two methodologies in the area of computation theory to solve optimal strategies for games such as Watermelon chess and Go. From experimental results, we find relevance to graph theory, group representation, and mathematical consciousness. We prove that the decision strategy of movement for Watermelon chess and Chinese checker games belongs to a matrix that is a noncommutative ring or an abelian group over set Y={-1,0,1}. Additionally, the movement for any chess game with two players belongs to a noncommutative ring or an abelian group from Occam's razor principle. We derive the closed form of the transition matrix for any chess game with two players and discover that the element of the transition matrix belongs to a rational number. We propose a different methodology based on abstract algebra to analyze the complexity of chess games in their entirety, instead of being limited solely to endgame results. It is probable that similar decision processes of people may also belong to a noncommutative ring or an abelian group.**

***Index Terms*—relational game, group representation, mathematical consciousness.**

I. INTRODUCTION

PEBBLE games [1] [2] and relational games [3] are first order logic [1] [4]. In this paper, we use a traditional chess game, Watermelon chess, to discuss the phenomena about mathematics and computation theory. The playing rules of Watermelon chess are similar to the game Go, such that when an opponent's pieces are blocked, they become captured, and the endgame result has three possible states of a win, loss, or draw.

Go and Watermelon chess are tri-valued logic, differing from Chinese checkers which is fourth-valued logic. We proposed two methods that can used to solve games such as Watermelon chess and Go.

In addition, we demonstrate that the movement of pieces for any chess game with two players such as the Chinese chess, chess, or Go belongs to a noncommutative ring or an abelian group, in which the group representation matrix is only with different dimensions of complexity over $\mathbb{Q}$, the rational numbers. We apply group representation theory in abstract algebra, the noncommutative ring and abelian group, to analyze the complexity of the chess games as a novel methodology to analyze the complexity of whole chess games.

In graph theory, graphs can be categorized as being directed acyclic graphs (DAG) and cyclic graphs (CG). We found that the Watermelon chess can be represented as either a DAG or CG.

A graph G can be represented as a tuple of its set of vertices V and edges E. The simple graph topology model of the initial playing state of Watermelon chess in Figure 1 can be presented as $G = (V, E)$, where

$$V = \{x \mid 0 \leq x \leq 20, x \in N\}, \tag{1}$$

$$\begin{aligned} E = \{&(0,1),(0,2),(0,3),(0,4),(1,2),(1,0),(1,4),(1,5),(2,0),\\ &(2,1),(2,3),(2,6),(3,0),(3,7),(3,4),(4,0),(4,1),(4,3),\\ &(4,8),(5,1),(5,10),(5,9),(5,20),(6,2),(6,11),(6,12),\\ &(6,13),(7,3),(7,14),(7,15),(7,16),(8,4),(8,17),(8,18),\\ &(8,19),(9,5),(9,10),(9,20),(10,5),(10,9),(10,11),(11,6),\\ &(11,12),(11,10),(12,6),(12,13),(12,11),(13,6),(13,12),\\ &(13,14),(14,7),(14,13),(14,15),(15,7),(15,14),(15,16),\\ &(16,7),(16,15),(16,17),(17,8),(17,16),(17,18),(18,8),\\ &(18,17),(18,19),(19,8),(19,18),(19,20),(20,5),(20,9),\\ &(20,19)\}. \end{aligned} \tag{2}$$

Here, we consider the whole game as a higher ordered complex graph. If we view each snapshot of the chess board as a vector point, it can be represented as the vertex of a graph. Each snapshot of the chess board is $a \in A = X^{21}$, where $X = \{1, 2, 3\}$, in which the indices 1 and 2 represent the two possible colors of chess pieces, and 3 represents the blanket. Every chess movement is represented by an edge of the graph, thereby allowing the whole game to be composed of a graph. Each edge is a functional operator (a matrix) or a function mapping f, $f : A \mapsto A$, where $A = X^{21}, X = \{1, 2, 3\}$.

In computational spin networks, chapter 2 of [5], every eigenstate can be represented as a vector, and the matrix operator is the edge for dual Hilbert space [5] [6]. We apply this idea to the high ordered whole game graph. From Theorem 3, we found that the chess movement is a function mapping f that belongs to $M_n(\mathbb{Q})$, a $n \times n$ matrix over $\mathbb{Q}$, the rational

Manuscript received December 30, 2022; *(Tsung-Ren Huang, Chen-An Tsai, and Yen-Jen Oyang equally contributed to this work.) (Corresponding author:Yen-Jen Oyang.)*

C.-K. Hwang, and J.R. Gilbert are with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 106, R.O.C. e-mail:chunkai.hwang@gmail.com; johngilbert1182@gmail.com

T.-R. Huang is with the Department of Psychology, National Taiwan University, Taipei, Taiwan, 106, R.O.C. e-mail: trhuang@ntu.edu.tw

C.-A. Tsai is with the Department of Agronomy, National Taiwan University, Taipei, Taiwan, 106, R.O.C. e-mail: catsai@ntu.edu.tw

Y.-J. Oyang is with the Graduate Institute of Biomedical Electronics and Bioinformatics, National Taiwan University, Taipei, Taiwan, 106, R.O.C. He is also with the Department of Computer Science and Information Engineering, the Graduate Institute of Networking and Multimedia, and the Master Programs in Statistics of National Taiwan University. e-mail:yjoyang@csie.ntu.edu.tw

TABLE I
COMPARISON OF GAME STRATEGY ANALYSIS APPROACHES.

Presenter	Method	Concept	Game type
Ours	Algebra	Graph, Group representation	A whole game
Marton Morse and Gustav A. Hedlund [7]	Algebra	Semigroups	Chess endgames
Lewis Stiller [8]	Algebra	Multilinear algebra	Chess endgames
Julian Schrittwieser *et al.* [13]	AI	Reinforcement learning	Game strategy
Fenil Mehta1 *et al.* [14]	AI	Multilayer perceptron	Game strategy
David Noever1 *et al.* [15]	AI	Natural language transformer	Game strategy
Shengyu Zhang [16]	Game theory	Quantum game theory	Game strategy

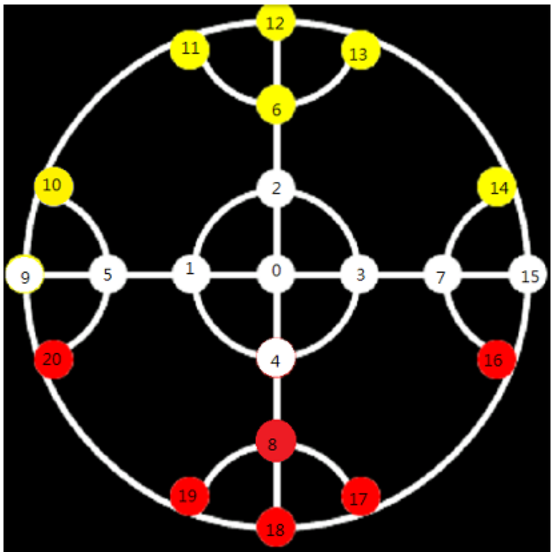

Fig. 1. The initial playing state of Watermelon chess.

numbers. Additionally, $M_n(\mathbb{Q})$ is a noncommutative ring or an abelian group under addition.

II. Related Works

Table I is a summary of different approaches to analyse chess games. Marton Morse and Gustav A. Hedlund [7] discussed the symbolic dynamics and the semigroups based on the unending chess. Lewis Stiller [8] applied multilinear algebra to construct the model to analyze chess endgames [9] [10] from humans and computers. Chess endgames can be analyzed with Kronecker tensor product and direct sums. They discussed the work of Friedrich Amelung and Theodor Molien, the founder of group representation, and the first person that analyzed a pawnless endgame [11] [12]. High-performance parallel computing can be solved from the endgame analysis.

Chess endgames have been studied for over a century. It has been applied to improve the prediction accuracy and efficiency of reinforcement learning, data compression [17], and two-level logic minimization [17]. The related algorithms for the logic minimization are MINI [18], ESPRESSO [19] [20], and Pupik [21]. It can also be applied to electronic design automation (EDA).

Julian Schrittwieser *et al.* [13] developed a reinforcement learning algorithm to solve for strategies for Atari, Go, chess and shogi games, while requiring approximately a million training steps. They compared different agents, or algorithms, including Ape-X, R2D2, MuZero, IMPALA, Rainbow, UNREAL, LASER, MuZero Reanalyze. Among these agents, MuZero Reanalyze achieved the best performance. Fenil Mehta1 *et al.* [14] predicted chess moevement by using the multilayer perceptron model. They used a chess board evaluation function that could be applied to evaluate the board without deep lookahead search algorithms. They developed a chess engine, thereby avoiding the use of state space search to find the next optimal movement. David Noever1 *et al.* [15] applied natural language transformers to support more generic strategic modeling, especially for text-archived games. Their approach focused on the breath search of millions of games that would allow a language model to define a game's rules and strategy by itself. Shengyu Zhang [16] presented a simple but complete model to extend the quantum strategic game theory.

III. Simulation Algorithms: Watermelon Chess

Watermelon chess is one kind of pebble game with two players. The initial state of the chess table is presented in Figure 1, with yellow and red representing the pieces of the two players.

A. Method 1: Game Tree

We constructed an objective function that optimized the degree of freedom (DoF), i.e., the number of chess pieces that are not captured by the opponent. The optimization objective value for a player is to maximize its DoF while minimizing its opponent's DoF. It is a general algorithm that can be applied with a min-max game tree [22], and it is applicable to both Watermelon chess and Go. In order to determine that the opponent's chess pieces are captured, we can derive a mutable tree presented in Figure 2. If the leaf nodes are comprised entirely of a single player's colors, then the opponent's chess pieces have all been catpured.

B. Method 2: Probabilistic NN Rule Models

We proposed PBCR1 and PBCR2 algorithms [23] to extract probabilistic Boolean rule models from Deep Neural Networks (DNN), which can also be applied to predict chess movement. It basically can be applied to auto-inference [23].

We recorded the 100 playing games of chess movements, and among these games, the win ratio of red color to yellow color is 50 to 50, and as a result we regarded it as a binary classification problem. There were a total of 4985 chess piece movements, with 2519 win (positive) cases and 2466 loss (negative) cases. The training features of this problem were

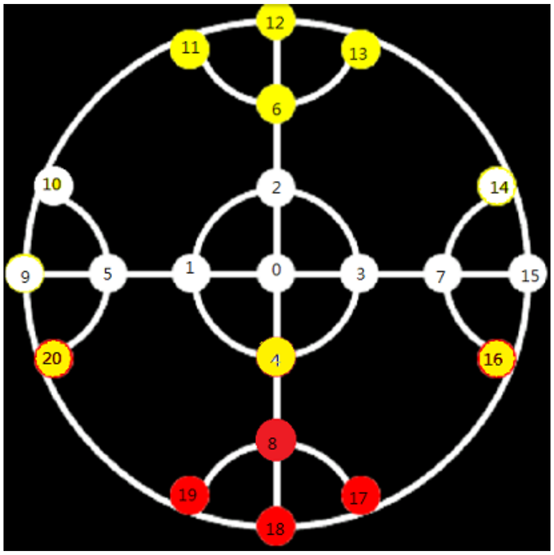

Fig. 2. The Graph of chess table during playing state.

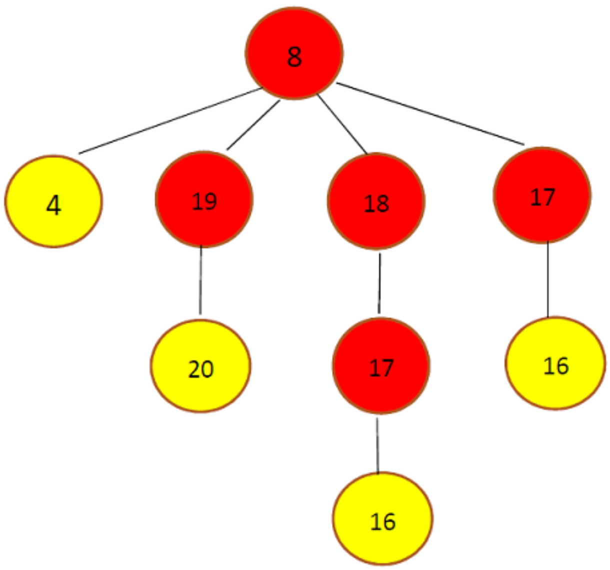

Fig. 3. The mutable tree for the chess pieces of red color blocked by the opponent's yellow color pieces that corresponds to Figure 2.

the movement of chess pieces. The input was the current chess state, the next state, and the current playing color. The current state was composed of 21 nodes. Each chess piece used a triple value to denote it, red, yellow, and the blanket chess pieces respectively. We use 2 bits to store the triple value. The current and the next state needed 42 nodes, and they were composed of 42×2=84 bits, with 1 bit also used to denote the current playing color. As a result, there were a total of 85 inputs and 1 binary output (win or loss).

IV. Results

Table II presents the performance metrics of DNN, PBCR1, PBCR2 and decision tree (DT) methods. In general, the performance of the DNN on each metric was better than DT, PBCR1, and PBCR2. The comparison was based on the same sensitivity on training fold to be 80% statistically equivalent, such that the confidence intervals of the sensitivity with different algorithms were overlapped.

V. Discussion

We found some interesting phenomena during different kinds of chess games. If we regard each snapshot of the chess board as a vector point, it can be regarded as the vertex of a graph. Every chess movement is the edge of the graph, The edge is a function mapping or the transition matrix $f : A \mapsto A$, where $A = X^{21}, X = \{1, 2, 3\}$, in which 1 and 2 represent the different colors of the chess pieces, and 3 represents the blanket, such that the whole game composed a graph. The win or loss can be either a directed acyclic graph (DAG) or a cyclic graph (CG), such that if there were no duplicated vertices of the graph, it would be a DAG, and a CG otherwise. However, the drawn game is a CG. The cardinality of the vertex would be infinite but with a finite element of the vertex set. Therefore, the graph composed from a drawn game must contain duplicate vertices. As a result, it is a CG.

Theorem 1. Noncommutative Ring: If we view the snapshot of the chess board as a vertex of a graph, the edge of this graph represents a chess movement, and the edge is an operator (a matrix) or a function mapping f, $f : A \mapsto A$, where $A = X^{21}, X = \{1, 2, 3\}$, 1, and 2 are the different colors of the chess pieces, and 3 represents the blanket, then the following properties hold:

1) There exists a function mapping f belonging to $M_{21}(Y)$, a 21×21 matrix over Y, where $Y = \{-1, 0, 1\}$.
2) $M_{21}(Y)$ is a noncommutative ring in abstract algebra.
3) $M_{21}(Y)$ is an abelian group under addition.

Proof:

1) For a vertex $a = [a_i]$, where $i = 0, \dots, 20$, if we move the chess piece from p to q, then either $a_p = 1$ or $a_p = 2$. Furthermore, $a_q = 3$ as q is a blanket. The function mapping f is a 21×21 matrix M_{21}.
   First, consider the case that $a_p = 1$. Suppose that after the chess movement from p to q, we will get a vertex $b = [b_i] = M_{21}a$, with the properties that for chess pieces $b_p = 3$ and $b_q = 1$. Pieces with indices $k \in K$ are captured with this movement, resulting in $b_k = 3$. Similarly, if the movement does not cause the chess pieces to be captured, then $b_i = a_i$ for $\forall i \neq p, q$, and $i \notin K$.
   Next, we show how to construct the matrix $M_{21} = [\sigma_{ij}]$ with Occam's razor principle. Suppose that the chess piece s belongs to the opponent, i.e., $a_s = 2$, for the p-th row vector r_p of M_{21}, we can set $\sigma_{pp} = 1, \sigma_{ps} = 1$, and $\sigma_{pj} = 0, \forall j \neq p, s$. Thus, we can construct the transformation that $b_p = 3$.
   For the q-th row vector r_q of M_{21}, we can set $\sigma_{qq} = 1, \sigma_{qs} = -1$, and $\sigma_{qj} = 0, \forall q \neq p, s$. Hence, we can construct the transformation where $b_q = 1$.
   For the chess pieces $k \in K$ that are captured, which can be obtained from the tree from Method 1, the k-th row vector r_k of M_{21} is of the form $\sigma_{kk} = 1, \sigma_{kp} = 1$,

TABLE II
PERFORMANCE METRICS OF DNN, PBCR1, PBCR2, AND DT.

Metrics(mean±SD)	DNN	PBCR1	PBCR2	DT
AUC	0.971±0.003	0.814±0.018	0.798±0.020	0.855±0.017
Accuracy	88.435%±0.385%	75.486%±2.212%	75.236%±2.089%	76.239%±1.723%
Sensitivity	80.051%±0.046%	79.866%±0.315%	79.959%±0.368%	80.172%±0.239%
PPV	95.504%±0.910%	72.715%±3.057%	72.324%±2.856%	74.164%±2.341%
NPV	83.556%±0.107%	78.790%±1.010%	78.736%±0.957%	78.700%±0.846%
Specificity	96.409%±0.757%	71.321%±4.367%	70.744%±4.123%	72.361%±3.381%
Odds ratio	113.404±29.428	10.187±2.243	9.928±2.145	10.796±1.806
F1	0.871±0.004	0.761±0.016	0.759±0.016	0.770±0.013

and $\sigma_{kj} = 0, \forall j \neq k$. Therefore, we can construct the transformation in which $b_k = 3$.

For the chess pieces i that are not captured and do not equal p, q, and do not belong to K, the i-th row vector r_i of M_{21} is of the form $\sigma_{ii} = 1, \sigma_{ij} = 0, \forall j \neq i, \forall i \neq p, q$, and $i \notin K$.

Hence, we construct the transformation that $b_i = a_i$ for $\forall i \neq p, q$, and $i \notin K$. Similarly, we can construct the matrix M_{21} for the case that $a_p = 2$. Finally, we observed that the element of the matrix M_{21}, $\sigma_{ij} \in \{-1, 0, 1\}$.

2) Since $Y = \{-1, 0, 1\}$ is isomorphic to the set $\mathbb{Z}_3 = \{0, 1, 2\}$. The set $\mathbb{Z}_3$ is a ring, the operator f is equivalent to a 21×21 matrix defined on $\mathbb{Z}_3$. $M_{21}(\mathbb{Z}_3)$ of all 21×21 matrices over $\mathbb{Z}_3$ is a noncommutative ring with identity I_n from Theorem F.1 [24]. It is stated as follows, "If R is a ring with identity, then the set $M_n(R)$ of all $n \times n$ matrices over R is a noncommutative ring with identity I_n". Thus, $M_{21}(Y)$ is a noncommutative ring.
3) From Theorem 7.1 [24], every ring is an abelian group under addition and the statement of (2) that $M_{21}(Y)$ is a noncommutative ring, it is trivial that $M_{21}(Y)$ is an abelian group under addition.

Theorem 2. Noncommutative Ring for Chinese Checkers: If we view the snapshot of the chess board as a vertex of a graph, in which the edge of this graph is a chess movement, the edge is an operator (a matrix) or a function mapping f, $f : A \mapsto A$, where $A = X^n$, $X = \{1, 2, 3, 4\}$, and where 1,2,3 denote different categories of chess pieces, and 4 denotes the blanket, and n is the number of the chess board lattices, then the following properties hold:

1) There exists a function mapping f belonging to $M_n(Y)$, an $n \times n$ matrix over Y, where $Y = \{-1, 0, 1\}$.
2) $M_n(Y)$ is a noncommutative ring in abstract algebra.
3) $M_n(Y)$ is an abelian group under addition.

Proof: For a vertex $a = [a_i]$, where $i = 0, \ldots, n$. If we move the chess piece from p to q, then either $a_p = 1, 2, or 3$. Additionally, $a_q = 4$ since q is blanket. The function mapping f is an $n \times n$ matrix M_n.

First, we consider the case that $a_p = 1$. Suppose that after the chess movement from p to q, we get a vertex $b = [b_i] = M_n a$, with the properties that for chess pieces $b_p = 4$ and $b_q = 1$. For other lattices on the chess board, we would have $b_i = a_i$ for $\forall i \neq p, q$.

Now, we show how to construct the matrix $M_n = [\sigma_{ij}]$ with Occam's razor principle. Suppose that the chess piece s belongs to other opponents, i.e., $a_s = 3$, for the p-th row vector r_p of M_n, we can set $\sigma_{pp} = 1, \sigma_{ps} = 1$, and $\sigma_{pj} = 0, \forall j \neq p, s$. Hence, we construct the transformation that $b_p = 4$.

For the q-th row vector r_q of M_n, we can set $\sigma_{qq} = 1, \sigma_{qs} = -1$, and $\sigma_{qj} = 0, \forall q \neq p, s$. Thus, we construct the transformation that $b_q = 1$.

For other lattices i that does not equal p, q, the i-th row vector r_i of M_n is of the form $\sigma_{ii} = 1, \sigma_{ij} = 0, \forall j \neq i, \forall i \neq p, q$. Hence, we construct the transformation that $b_i = a_i$ for $\forall i \neq p, q$.

Similarly, we can construct the matrix M_n for the case that $a_p = 2, 3$. Finally, we observed that the element of the matrix $M_n, \sigma_{ij} \in \{-1, 0, 1\}$. (2) and (3) are proved in a similar procedure as Theorem 1.

Theorem 3. Noncommutative Ring for Any Chess Game with Two Players: If we view the snapshot of the chess board as a vertex of a graph, in which the edges of the graph are chess movements, the edge is an operator (a matrix) or a function mapping f, $f : A \mapsto A$, where $A = X^n, X = \{1, 2, 3, ..t, t+1, ..., d, d+1\}$, where $1, 2, 3, \ldots, t$, denotes t different categories of chess pieces for player Alice, $t+1, \ldots, d$ denotes $(d-t)$ different categories of chess pieces for player Bob, and $d + 1$ denotes the blanket, and n is the number of the chess board lattices, then the following properties hold:

1) There exists a function mapping f belongs to $M_n(\mathbb{Q})$, an $n \times n$ matrix over $\mathbb{Q}$, the rational numbers.
2) $M_n(\mathbb{Q})$ is a noncommutative ring in abstract algebra.
3) $M_n(\mathbb{Q})$ is an abelian group under addition.

Proof:

1) Similar to the proof of Theorem 1, since any $x \in X = \{1, 2, 3, ..., d+1\}$ can be composed from an inner product of row vector $y \in \mathbb{Q}^n$ and any vector$a \in X^n$, we can construct an $n \times n$ matrix over R, $M_n \in M_n(\mathbb{Q})$, such that $b = M_n a \in X^n$, i.e., M_n is composed of different row vector $y \in \mathbb{Q}^n$.

For a vertex $a = [a_i]$, where $i = 0, \ldots, n$. If we move the chess piece from p to q, then we have $a_p = 1, 2, \ldots, t, , t+1, \ldots, d$. In addition, $a_q = d+1$ as q is blanket. The function mapping f is an $n \times n$ matrix M_n.

First, we consider the case that $a_p = x \in X$. Suppose that after the chess movement from p to q, we will get a vertex $b = [b_i] = M_n a$, with the properties that for chess pieces $b_p = d+1$ and $b_q = x$. Pieces with indices $k \in K$ are captured with this movement, resulting in $b_k = d+1$. In addition, if the movement does not cause the chess pieces to be captured, then $b_i = a_i$ for $\forall i \neq p, q$, and $i \notin K$.

Here, we give an intuitive simple construction of $M_n = [\sigma_{ij}]$ with Occam's razor principle. Suppose that the opponent chess piece s with the properties, $a_s = u, t + 1 \leq u \leq d$. We want to derive the σ_{ps} such that $\sigma_{pp}x + \sigma_{ps}u = d + 1$. Therefore, if $\sigma_{pp} = 1$, we have $\sigma_{ps} = (1/u)(d + 1 - x) \in \mathbb{Q}$.

For the p-th row vector r_p of M_n, we can set $\sigma_{pp} = 1$, and $\sigma_{ps} = (1/u)(d+1-x) \in \mathbb{Q}$, and $\sigma_{pj} = 0, \forall j \neq p, s$. Hence, we construct the transformation that $b_p = d+1$.

For the q-th row vector r_q of M_n, we want to derive the σ_{ps} such that $\sigma_{qq}(d+1)+\sigma_{qs}u = x$. Therefore, if $\sigma_{qq} = 1$, we have $\sigma_{qs} = (1/u)(x - d - 1) \in \mathbb{Q}$, and $\sigma_{qj} = 0, \forall q \neq p, s$. Hence, we construct the transformation that $b_q = x$.

For the chess pieces $k \in K$ that are captured, as determined by the rules of chess, let the captured chess piece be denoted with value v. For the k-th row vector r_k of M_n, we want to derive the σ_{kp} such that $\sigma_{kk}v + \sigma_{kp}x = d + 1$. We can construct M_n of the form $\sigma_{kk} = 1, \sigma_{kp} = (1/x)(d + 1 - v) \in \mathbb{Q}$, and $\sigma_{kj} = 0, \forall j \neq k$. Therefore, we construct the transformation that $b_k = d + 1$.

For the chess pieces i that are not captured, do not equal p, q, and do not belong to K, the i-th row vector r_i of M_n is of the form $\sigma_{ii} = 1, \sigma_{ij} = 0, \forall j \neq i, \forall i \neq p, q$, and $i \notin K$. Hence, we construct the transformation that $b_i = a_i$ for $\forall i \neq p, q$, and $i \notin K$. Consequently, we observed that the element of the matrix belongs to rational numbers, that is, $M_n = [\sigma_{ij}], \sigma_{ij} \in \mathbb{Q}$.

2) Since $\mathbb{Q}$ is a ring with identity, from Theorem F.1 [24]. $M_n(\mathbb{Q})$ is a noncommutative ring.
3) It is trivial and similar to the proof (3) of Theorem 1.

VI. Conclusions and Future Works

We can discuss the complexity of different chess games from Theorem 1, 2, and 3. From Theorem 1, we found that the movements in Watermelon chess belong to $M_{21}(Y)$, a 21×21 matrix over Y, where $Y = \{-1, 0, 1\}$. From Theorem 2, we found that the chess movements of Chinese checker belong to $M_n(Y)$, a $n \times n$ matrix over Y, where $Y = \{-1, 0, 1\}$ and n is the number of the chess board lattices. From Theorem 3, we found that the movements of any chess game with two players belong to $M_n(\mathbb{Q})$, a $n \times n$ matrix over $\mathbb{Q}$, where $\mathbb{Q}$ represents the rational numbers and n is the numbers of the chess board lattices.

In Theorem 1, we presented a proof that the set of the function mapping f is a noncommutative ring or an abelian group. It is an interesting topic about the algebraic structure of the operator f, for example, whether it is a group similar to $SU(2)$ [5] [25]. We found the group representation matrix is composed of the element form set $Y = \{-1, 0, 1\}$, and it is a sparse matrix. Chess games might be related to reductive groups [26], compact groups [27], Lie groups [28] [29] and random walks on groups and random transformations [30] in abstract algebra.

In quantum theory, we know that the quantum is in essence the tensor operation on the dual space Figure 2.1 [5]. We may develop a quantum game theory method [16] [31] [32] [33] via an approach that is similar to computational spin networks [5]. Every state corresponds to a vector. It is probable that the human decision-making process for other strategies not limited to playing chess can be represented as a matrix operator that belongs to a noncommutative ring or an abelian group, $M_n(\mathbb{R})$, or $M_n(\mathbb{C})$, where $\mathbb{R}$ represents the real numbers, and $\mathbb{C}$ represents the complex numbers.

References

[1] A. Kfoury and A. P. Stolboushkin, "An infinite pebble game and applications," *information and computation*, vol. 136, no. 1, pp. 53–66, 1997.
[2] A. Lee and I. Streinu, "Pebble game algorithms and sparse graphs," *Discrete Mathematics*, vol. 308, no. 8, pp. 1425–1437, 2008.
[3] Y. Venema, *Relational games*. Instituut voor Taal, Logica en Informatie, 1990.
[4] J. Y. Halpern and V. Weissman, "Using first-order logic to reason about policies," *ACM Transactions on Information and System Security (TISSEC)*, vol. 11, no. 4, pp. 1–41, 2008.
[5] J. Moffat, *A Mathematically Coherent Quantum Gravity*. IOP Publishing, 2020.
[6] A. J. Sieradski, *An introduction to topology and homotopy*. Wadsworth Publishing Company, 1992.
[7] M. Morse and G. A. Hedlund, "Unending chess, symbolic dynamics and a problem in semigroups," *Duke Mathematical Journal*, vol. 11, no. 1, pp. 1–7, 1944.
[8] L. Stiller, "Multilinear algebra and chess endgames," *Games of no chance*, vol. 29, pp. 151–192, 1996.
[9] M. S. Bourzutschky, J. A. Tamplin, and G. M. Haworth, "Chess endgames: 6-man data and strategy," *Theoretical Computer Science*, vol. 349, no. 2, pp. 140–157, 2005.
[10] E. V. Nalimov, G. Haworth, and E. A. Heinz, "Space-efficient indexing of chess endgame tables," *ICGA Journal*, vol. 23, no. 3, pp. 148–162, 2000.
[11] L. B. Stiller, *Exploiting symmetry on parallel architectures*. The Johns Hopkins University, 1995.
[12] I. Good, "P. hĭjek and t. havrĭnek, mechanizing hypothesis formation. mathematical foundations for a general theory," *Bulletin (New Series) of the American Mathematical Society*, vol. 1, no. 4, pp. 650–654, 1979.
[13] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel *et al.*, "Mastering atari, go, chess and shogi by planning with a learned model," *Nature*, vol. 588, no. 7839, pp. 604–609, 2020.
[14] F. Mehta, H. Raipure, S. Shirsat, S. Bhatnagar, and B. Bhovi, "Predicting chess moves with multilayer perceptron and limited lookahead," *Journal of Engineering Research and Applications*, vol. 10, no. 4, pp. 05–08, 2020.
[15] D. Noever, M. Ciolino, and J. Kalin, "The chess transformer: Mastering play using generative language models," *arXiv preprint arXiv:2008.04057*, 2020.
[16] S. Zhang, "Quantum strategic game theory," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, 2012, pp. 39–59.
[17] D. Gomboc and C. R. Shelton, "Chess endgame compression via logic minimization," in *Advances in Computer Games*. Springer, 2022, pp. 153–162.
[18] S. J. Hong, R. G. Cain, and D. L. Ostapko, "Mini: A heuristic approach for logic minimization," *IBM journal of Research and Development*, vol. 18, no. 5, pp. 443–458, 1974.
[19] H. Kanakia, M. Nazemi, A. Fayyazi, and M. Pedram, "Espresso-gpu: blazingly fast two-level logic minimization," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2021, pp. 1038–1043.
[20] O. Coudert and T. Sasao, "Two-level logic minimization," in *Logic Synthesis and Verification*. Springer, 2002, pp. 1–27.

[21] P. Fiser, P. Rucky, and I. Vanova, "Fast boolean minimizer for completely specified functions," in *2008 11th IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems*. IEEE, 2008, pp. 1–6.
[22] R. L. Rivest, "Game tree searching by min/max approximation," *Artificial Intelligence*, vol. 34, no. 1, pp. 77–96, 1987.
[23] C. K. Hwang, "Extracting classification boolean rules with probabilities from neural networks," Master's thesis, National Taiwan University, Jan 2020.
[24] T. W. Hungerford, *Abstract algebra: an introduction*. Cengage Learning, 2012.
[25] C. Rovelli and F. Vidotto, *Covariant loop quantum gravity: an elementary introduction to quantum gravity and spinfoam theory*. Cambridge University Press, 2015.
[26] A. Minchenko and A. Ovchinnikov, "Zariski closures of reductive linear differential algebraic groups," *Advances in Mathematics*, vol. 227, no. 3, pp. 1195–1224, 2011.
[27] P. P. Varjú, "Random walks in compact groups," *arXiv preprint arXiv:1209.1745*, 2012.
[28] E. Breuillard, "Random walks on lie groups," *Survey, http://www. math. u-psud. fr/breuilla/part0gb. pdf*, 2004.
[29] R. Versendaal, "Large deviations for random walks on lie groups," *arXiv preprint arXiv:1909.05065*, 2019.
[30] A. Furman, "Random walks on groups and random transformations," in *Handbook of dynamical systems*. Elsevier, 2002, vol. 1, pp. 931–1014.
[31] I. Ghosh, "Quantum game theory—i," *Resonance*, vol. 26, no. 5, pp. 671–684, 2021.
[32] ——, "Quantum game theory—ii: A comprehensive study." *Resonance: Journal of Science Education*, vol. 26, no. 6, 2021.
[33] ——, "Quantum game theory—iii," *Resonance*, vol. 26, no. 7, pp. 939–951, 2021.