Quantum Chess: Developing a Mathematical Framework and Design Methodology for Creating Quantum Games

Christopher Cantwell
ccantwel@usc.edu

July 12, 2019

1 Introduction

Quantum phenomena have remained largely inaccessible to the general public. This can be attributed to the fact that we do not experience quantum mechanics on a tangible level in our daily lives. In order to explore how quantum systems work one generally needs access to a lab and must understand some mathematics that most people aren't ever exposed to. But to gain an intuitive understanding it might not be necessary to understand the math. Consider baseball as an analogy. One doesn't need to understand classical mechanics to play baseball, and playing baseball will not teach one classical mechanics. However, playing baseball may help to more intuitively understand how things move. Games can provide an environment in which people can experience the strange behavior of the quantum world in a fun and mentally engaging way.

With progress being made quantum computers one might pose the question, "Will there be quantum games played on quantum computers?" If one assumes the answer is yes, then the next question is, "What will those games be?" Games could offer an interesting test bed for near term quantum devices. Games can be tailored to support varying amounts of quantum behavior through simple rule changes, which can be useful when dealing with limited resources. AI algorithms that can play such games can be designed to tap into quantum computing resources to assist in evaluating next moves. We may see stronger AI players evolve as more advanced quantum hardware becomes available. Here we explore the design of Quantum Chess while keeping in mind the goal of allowing people the opportunity to experience quantum phenomena.

2 Overview of Previous Work

There has been some investigation into a formal quantum game theory [11, 12, 20, 24]. This paper will focus more on the concept of implementing playable quantum games. The body of previous work is somewhat expectedly small. There are a handful of games that have been proposed, but to our knowledge never implemented [15, 25, 26, 28]. The following games have indeed been implemented, and their properties will help direct the work to be done: Quantum Tic-Tic-Tac-Toe [13], Quantum Minesweeper [14], QCraft [5], and another version of Quantum Chess [7]. More recently there have been citizen science type games like Decodoku [32], Quantum Moves [29], and the Alice Challenge [18] that have been used to help solve quantum problems, but we will not discuss these here as their goal is different than ours and their design did not influence the design of Quantum Chess. In examining the games mentioned we can find some gaps to fill when designing Quantum Chess.

2.1 Quantum Tic-Tac-Toe

Quantum Tic-Tac-Toe is a variant of Tic-Tac-Toe that allows players to place marks in superposition of being in two places at once. This allows for the possibility of a square to be occupied by more than one mark. In this way squares can be seen to be entangled. Under the condition of cyclic entanglement collapse occurs. The player who did not create the cycle chooses the mark to collapse to, causing the entire cycle to collapse.

✓ This game does illustrate superposition and entanglement. However, it is deterministic in its measurement outcomes (collapse), where quantum mechanical measurements are specifically non-deterministic. There is also no representation of interference.

2.2 Quantum Minesweeper

Quantum Minesweeper is a variant of minesweeper that is played on a board that begins in a superposition of different mine configurations. The player then performs different types of measurements to try and guess the superposition. The measurements available are classical, interaction free, and entanglement measurements. In this way the player may gain some understanding of these quantum phenomena. However interference is not present, and the player is not actively generating quantum effects.

2.3 QCraft

QCraft is a mod that adds a quantum twist to the world of Minecraft. It adds certain quantum properties to Minecraft like observational dependency, teleportation, superposition, and entanglement. It is a great game for introducing people to select aspects of quantum mechanics in a fun macroscopic environment. However, qCraft's own makers state that isn't, nor is it meant to be, an accurate quantum simulation. It is just meant to be a fun introduction to quantum concepts.

2.4 Selim Akl's Quantum Chess

Quantum Chess by Selim Akl is a chess variant designed with the purpose of evening the playing field between computer and human players. Pieces begin the game in superposition of being a number of different types of pieces at once until the player chooses which one to move. At that point in time a measurement occurs and the piece is collapsed to a single type, after which it can be moved. The piece remains in a classical state until it lands on a black square, at which time it is placed back into a superposition of being multiple different types. The game illustrates superposition, but in the absence of other quantum effects it seems to only add an element of randomness to chess, at least as far as attempting to formulate a strategy for play goes.

3 Quantum Chess Design Goals

An examination of the previous work highlighted one major gap that could be filled in the design of a Quantum Chess: interference. None of these games exhibited quantum interference effects in any way. This realization, and other more subjective desires for a quantum game, led to the following list of design goals

1. The game would include a trifecta of quantum evolutionary phenomena: superposition, entanglement, and interference.

2. Said trifecta would be implemented in a way that could lead to the development of quantum strategies for play. In other words, players should be able to create and use superposition, entanglement, and interference effects.[1]

3. Measurements must be non-deterministic.

4. The game would not teach quantum mechanics. Instead it would provide an environment in which players could interact with quantum phenomena and, hopefully, develop a more intuitive understanding of how said phenomena work as a side effect of playing a fun game.

The last goal stems from a desire to make the game accessible to the largest audience possible. With these design goals in mind we lay down the following criteria for how we will build Quantum Chess:

1. Movement is accomplished through unitary evolution of a quantum state. [2]

2. Players have access to a set of moves through which they can create superposition.

4 Quantum Chess Algebraic Notation

Classical chess has an established algebraic notation [1, 3]. Here we define a simple Quantum Chess Algebraic Notation for describing different types of moves.

- Non-quantum Move: $(source)(target)$, e.g. $a1a2$.

- Quantum Split Move: $(source) \wedge (target1)(target2)$, e.g. $a1 \wedge a2a3$

- Quantum Merge Move: $(source1)(source2) \wedge (target)$, e.g. $a3a2 \wedge a1$

- Pawn Promotion: $(source)(target)(piece)$, e.g. $a7a8Q$

All non-quantum moves involve a single source and single target, even castling can be described completely by the king source and king target. Adding further characters to a move string to give a more detailed description, as is common in standard Chess, is a tricky prospect given the nature of superposition. Another consideration is that in Quantum Chess we will also have measurements (see section 7). We will expand the notation to describe if a move requires a measurement, and the outcome of said measurement. For the purposes of this paper we will design all of our measurements to have two outcomes[3], thus we can append the following measurement notation to a move string to achieve:

- Move with Measurement Outcome 0: $(movestring).(measurestring)0$, e.g. $a1a2.m0$

- Move with Measurement Outcome 1: $(movestring).(measurestring)1$, e.g. $a1a2.m1$

We can define a set of measurement strings to clarify the type of measurement being applied, but if we know the entire history it is enough to leave it generic. The type of measurement is defined by the type of move that corresponds to the move string acting on the current game state.

---

[1]See [16] for more information on the concepts of superposition, entanglement, and interference.

[2]Quantum state evolution is described by the application of unitary operators. See [16]

[3]It is possible to design more general measurements which could have three or more outcomes. See [16] [27]. This would require additional notation.

5 State Description

For a mathematical framework it will be useful to define a vector format that describes the state. We will use a hybrid classical / quantum state representation to describe the board.

Our quantum state will be composed of 64 qubits to describe the "occupancy" of the board[4].

$$|\psi_{\mathcal{B}}\rangle = \sum_i A_i \left| q_0^{(i)}, ... q_{63}^{(i)} \right\rangle, q_j \in \{0 = empty, 1 = occupied\} \tag{1}$$

This state representation is similar to the bitboard representation used in classical chess [3] where our state is analogous to a superposition of the "all pieces" bitboard.

On top of this superposition of occupancy bitboards we will store classical type information for each square. This information consists of a single 64-element vector that describes what piece, if any, is occupying an given square.

$$\vec{v} = \{v_0, ... v_{63}\}, v_i \in \{0, P, N, B, R, Q, K, p, n, b, r, q, k\} \tag{2}$$

The values v_i correspond to standard FEN values for chess pieces [4] with lower case letters representing black pieces, upper case white pieces, and 0 empty. We say a square is occupied, or partially occupied, by a piece if there is a non-zero probability of finding that piece in that square. Note, this representation does not account for the possibility of superpositions of piece type, but we exclude that possibility with the "No Double Occupancy" rule described in section 7.

Finally we will need a set of classical information encoding the color of the current player, as well as any special flags for moves such as castling and en passant (e.p.). We define the following:

$$\mathcal{F} = \{F_c, F_K, F_Q, F_k, F_q, F_{ep}\} \tag{3a}$$

$$F_c \in \{w, b\}, \text{ is the color of the current player} \tag{3b}$$

$$F_K = \text{True, if king side castling is legal for white player} \tag{3c}$$

$$F_Q = \text{True, if queen side castling is legal for white player} \tag{3d}$$

$$F_k = \text{True, if king side castling is legal for black player} \tag{3e}$$

$$F_q = \text{True, if queen side castling is legal for black player} \tag{3f}$$

$$F_{ep} = \begin{cases} file \in \{a, b, c, d, e, f, g, h\}, & \text{file of the capturable pawn if e.p. is legal} \\ -, & \text{otherwise} \end{cases} \tag{3g}$$

With our state representation complete we can then consider the action of a move. Each move will have the following parts:

1. A unitary that acts to update the occupancy superposition $|\psi_{\mathcal{B}}\rangle$.

2. An operation that updates the classical piece type information $\vec{v}$.

3. An operation that updates the extra flags $\mathcal{F}$.

6 Movement Unitary Design

In standard Chess a simple, non-capturing move from a source square to a target square can be accomplished by swapping the values of the pieces in the source square and target square.

---

[4]We use standard Dirac, or Bra-ket, notation to describe the quantum state. See [16] [27] [2]

This realization leads to a natural unitary to accomplish movement on a quantum board: the quantum Swap gate [27]. We can also base a move that allows for the creation of superposition on the $\sqrt{Swap}$ unitary. For arbitrary reasons we choose the iSwap instead of Swap, but this does not change the conceptual effect of the move. Our basic unitaries are thus:

$$U_{iSwap} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4a}$$

$$U_{\sqrt{iSwap}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}} & 0 \\ 0 & \frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4b}$$

However we can't just use these operators out of the box. We must also consider pathing and the piece types involved. Here we will detail the design of a set of movement unitaries, based on the idea of using Swap, that can be used to construct moves. The procedure for designing the unitary operators that act on the occupancy superposition will be as follows:

1. Express the desired action of the move, acting on the relevant subspace of otherwise empty board, in the form of an operator being applied to a Hilbert space.

2. Consider specific desired outcomes in the presence of superposition and modify the operator, keeping in mind unitarity, to reflect these outcomes. This may involve expanding the Hilbert space on which the operator acts.

3. Use the unitary requirement to complete the operator.

Once we have the unitary operators designed we can then use them to implement the various types of movement allowed in Quantum Chess

6.1   Jump Unitary

The Jump Unitary is the simplest of the movement unitaries. It will be useful for pieces that do not slide along some path, like the knight or pieces that take a single step. The operator acts on a subspace of the board involving only two squares: a source (s) and target (t). We can define the unitary operator as it acts on a reduced Hilbert space defined by the basis $|t, s\rangle$. This unitary then takes the simple form of the iSwap we introduced in equation 4a.

$$U_{jump} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5}$$

6.2   Slide Unitary

The Slide Unitary is the similar to the jump but we must add a control qubit. It will be useful for pieces that slide along some path, like the bishop, rook, and queen. We must expand the basis of the Hilbert space to be $|p, t, s\rangle$, where p is an ancilla qubit that we will set to be 1 if there is a piece occupying the path between s and t. We can set the state of p by applying a

sequence of CNOT operators commonly seen in quantum computing circuits [27]. This unitary then takes the simple form of a controlled-iSwap, where it acts as identity when the path qubit is set to 1.

$$U_{slide} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & i & 0 & 0 & 0 & 0 & 0 \\ 0 & i & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{6}$$

6.3 Split Jump Unitary

The Split Jump unitary is designed to allow a player to create superposition. This unitary will be based on the square root of an iSwap introduced in equation 4b. If $U_{\sqrt{iSwap}}$ is applied to the basis states of a two qubit Hilbert space we see

$$|00\rangle \to |00\rangle, |11\rangle \to |11\rangle$$
$$|01\rangle \to \frac{1}{\sqrt{2}}(|01\rangle + i|10\rangle)$$
$$|10\rangle \to \frac{1}{\sqrt{2}}(|10\rangle + i|01\rangle)$$

This can be interpreted as a piece both moving and not moving. Not moving a piece might not be very desirable for a player. One option to counteract this is to allow the move to have twice the range so that on average a player accomplishes the same amount of work. This was the solution in a previous variant of Quantum Chess [6,9] but we have chosen to drop it for a number of reasons. Instead we choose to allow the player to "split" the piece to exist in superposition on two different squares. For this we need two targets: t_1 and t_2. We can accomplish the Split Jump by first performing our $\sqrt{iSwap}$ between s and t_1, and then performing a full iSwap between s and t_2. In the $|t_2, t_1, s\rangle$ basis these unitaries take the form:

$$U_{\sqrt{iSwap}}(s, t_1) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad U_{iSwap}(s, t_2) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & i & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & i & 0 \\ 0 & i & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

which gives us a total Split Jump unitary:

$$U_{split} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & i & 0 & 0 & 0 \\ 0 & \frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-1}{\sqrt{2}} & \frac{i}{\sqrt{2}} & 0 \\ 0 & \frac{i}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{i}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{7}$$

A quick examination of the non-trivial action of U_{split} acting on the basis states of the $|t_2, t_1, s\rangle$ Hilbert space shows us that this unitary conserves piece number(occupancy number), which is an important consideration if we don't want pieces randomly appearing or disappearing in the game.

$$U_{split}|001\rangle = \frac{1}{\sqrt{2}}(i|010\rangle + i|100\rangle) \quad (8a) \qquad U_{split}|011\rangle = i|110\rangle \quad (8d)$$

$$U_{split}|010\rangle = \frac{1}{\sqrt{2}}(|010\rangle - |100\rangle) \quad (8b) \qquad U_{split}|101\rangle = \frac{1}{\sqrt{2}}(-|011\rangle + i|101\rangle) \quad (8e)$$

$$U_{split}|100\rangle = i|001\rangle \quad (8c) \qquad U_{split}|110\rangle = \frac{1}{\sqrt{2}}(i|011\rangle - |101\rangle) \quad (8f)$$

The zero and three piece subspaces are trivially conserved by identity.

6.4 Split Slide Unitary

The Split Slide unitary is a variation on the Split Jump unitary that takes into account the path a piece might take. When performing a Split Slide there are two paths to consider. We expand our basis to include two path control qubits: p_1 is the path from s to t_1, and p_2 is the path from s to t_2. We thus define the Split Slide unitary in the basis $|p_2, p_1, t_2, t_1, s\rangle$:

$$U_{split\_slide} = \begin{pmatrix} U_{split} & \underline{0} & \underline{0} & \underline{0} \\ \underline{0} & U_{iSwap}(s, t_2) & \underline{0} & \underline{0} \\ \underline{0} & \underline{0} & U_{iSwap}(s, t_1) & \underline{0} \\ \underline{0} & \underline{0} & \underline{0} & I_{8x8} \end{pmatrix}_{32x32} \tag{9}$$

We have chosen to have our move act as full iSwaps in the event that a single path is blocked.

6.5 Merge Jump Unitary

A natural extension of allowing players to split pieces is to allow them to merge pieces back together. With the Merge Jump unitary a player may undo a Split Jump. We define the Merge Jump operator to be the hermitian conjugate of the Split Jump operator. We must define our basis to be $|s_1, s_2, t\rangle$ if we want this unitary to act as the hermitian conjugate of a Split Jump

7

on the basis $|t_2, t_1, s\rangle$ described above. Thus our Merge Jump unitary is:

$$U_{merge} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-i}{\sqrt{2}} & 0 & \frac{-i}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{-1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -i & 0 \\ 0 & -i & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{-1}{\sqrt{2}} & 0 & \frac{-i}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & \frac{-i}{\sqrt{2}} & 0 & \frac{-1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{10}$$

And again we see that the non-trivial effect on the basis states of the $|s_1, s_2, t\rangle$ Hilbert space conserves piece number.

$$U_{merge}|001\rangle = -i|100\rangle \tag{11a}$$

$$U_{merge}|010\rangle = \frac{-1}{\sqrt{2}}(i|001\rangle - |010\rangle) \tag{11b}$$

$$U_{merge}|100\rangle = \frac{-1}{\sqrt{2}}(i|001\rangle + |010\rangle) \tag{11c}$$

$$U_{merge}|011\rangle = \frac{-1}{\sqrt{2}}(|101\rangle + i|110\rangle) \tag{11d}$$

$$U_{merge}|101\rangle = \frac{-1}{\sqrt{2}}(i|101\rangle + |110\rangle) \tag{11e}$$

$$U_{merge}|110\rangle = -i|011\rangle \tag{11f}$$

6.6 Merge Slide Unitary

As with the Split Jump, for some pieces we must consider the occupancy of the paths: p_1 from s_1 to t, and p_2 and from s_2 to t. In the basis $|p_2, p_1, s_1, s_2, t\rangle$ we define our Merge Slide Unitary:

$$U_{merge\_slide} = \begin{pmatrix} U_{mg} & \underline{0} & \underline{0} & \underline{0} \\ \underline{0} & U_{iSwap}^{\dagger}(s_2, t) & \underline{0} & \underline{0} \\ \underline{0} & \underline{0} & U_{iSwap}^{\dagger}(s_1, t) & \underline{0} \\ \underline{0} & \underline{0} & \underline{0} & I_{8x8} \end{pmatrix}_{32x32} \tag{12}$$

7   No Double Occupancy Rule

Before moving on to the design of the actual moves available to all of the pieces in the game, we need to discuss the No Double Occupancy Rule. Given that pieces can exist in superposition on a board it is possible that a move could result in a square being occupied by more than one type of piece.[5] This leads to a number of complications, both in execution of the game and in visual representation of the board for a player. For these, and various other, reasons we have chosen to add the "No Double Occupancy" rule.

Definition 1 *No Double Occupancy Rule: The no double occupancy rule states that at no point in the game do we allow a square to have a non-zero probability of being occupied by two or more pieces with different piece values.*

This language is chosen to allow for the interaction of like pieces. To enforce this rule we use a set of carefully designed projective measurements to ensure that the board is always in a state in which a move can't possibly lead to double occupancy. Whenever we perform a move

---

[5]More accurately, given our definition of state a move unitary could cause interference between different types of pieces, but then the classical piece type information is ambiguous. We may instead consider a Hilbert space that includes piece type, and then a move can be seen as placing two different pieces in the same square, i.e. Double Occupancy.

we first look at the classical piece type information to determine whether double occupancy is a possible outcome. If it is, we perform a measurement that projects the state into a subspace where double occupancy will not occur. The general procedure for constructing measurement operators will be as follows:

1. Consider the conditions that could lead to double occupancy after applying a move to a superposition of boards.

2. Construct two sets of mutually exclusive basis states, $\mathcal{M}_0$ and $\mathcal{M}_1$ for the Hilbert space on which the move unitaries act. These sets should be constructed such that double occupancy will never occur for a move acting on any state that has elements in only one subset or the other. And $\mathcal{M}_0$ should hold basis states for which the move must do nothing, while $\mathcal{M}_1$ holds basis states for which the move may have some non-trivial effect (i.e. the piece attempting the move does in fact move).

3. From $\mathcal{M}_0$ and $\mathcal{M}_1$ create two measurement operators, M_0 and M_1 that will be used to project the superposition into one of the two mutually exclusive subspaces.

4. There will be some basis states left over that can never conflict with the No Double Occupancy rule (e.g. $|000\rangle$ for moves that act on 3 qubits). Complete M_0 and M_1 with the goal of being able to describe what the measurement does to a player in a single sentence.[6]

8 Quantum Chess Moves

Now that our complete framework is in place we can detail the moves we allow in Quantum Chess. For each move (m) we will define its possibility.

$$\mathcal{P}_m(\vec{v}, \mathcal{F}) = \bigwedge_i C_i(m, \vec{v}, \mathcal{F}) \tag{13}$$

where $C_{m,i}(\vec{v}, \mathcal{F})$ are a set of constraints on the classical information of a given state for move m to be possible. We impose an additional condition to determine the *legality* of a move.

Definition 2 *Legal Move: A move (m) is said to be legal if it satisfies all of its possibility constraints, i.e. $\mathcal{P}_m(\vec{v}, \mathcal{F}) = true$, and it acts non-trivially on the occupancy superposition, i.e. $|\psi'_{\mathcal{B}}\rangle \neq |\psi_{\mathcal{B}}\rangle$ where $|\psi'_{\mathcal{B}}\rangle$ is the occupancy state after applying the move.*

We do not define a procedure for determining whether the superposition has changed. If executed on a quantum computer this would require some type of state tomography [22] [17] [23]. There are shortcuts one may take to make this determination when simulating on a classical computer.

We can now outline the measuring and non-measuring movement procedures all moves will follow using pseudocode.[7] Our non-measuring move procedure is:

Algorithm 1 *Execute Move: Consider a game with classical and quantum state $\{|\psi_{\mathcal{B}}\rangle, \vec{v}, \mathcal{F}\}$. Let $\underline{\psi_{\mathcal{B}}}$ be a quantum register initialized with state $|\psi_{\mathcal{B}}\rangle$. Let U_m be the unitary(s) associated with a move (m). To apply m to the game we use the following procedure.*

    procedure APPLY$(U_m, \underline{\psi_{\mathcal{B}}}, \vec{v}, \mathcal{F})$
        if $\mathcal{P}_m(\vec{v}, \mathcal{F}) = true$ **then**

---

[6]This is a somewhat arbitrary requirement for completing the operators. Any choice of design goal will be equally valid. For example one may choose to try and maintain as much superposition as possible during measurement, or reduce the total quantum circuit area of the combined measurement and move.

[7]We use a mix of classical pseudocode and quantum pseudocode introduced in [21].

$$\psi'_{\mathcal{B}} \leftarrow U_m \psi_{\mathcal{B}}$$
if $\psi'_{\mathcal{B}} \neq \psi_{\mathcal{B}}$ **then**
    Update $\vec{v}$ to reflect the correct pieces.
    ~~Update castling and e.p. flags in $\mathcal{F}$ appropriately~~
    ~~Update $\mathcal{F}_c$ to indicate change of player.~~
end if
    end if
  end procedure

This must be modified if we wish to include a measurement. The measurements presented here will always be two-outcome so it will be enough to supply M_1, as $M_0 = I - M_1$. Our algorithm thus becomes:

Algorithm 2 *Execute Measuring Move: Consider a game with state $\{|\psi_{\mathcal{B}}\rangle, \vec{v}, \mathcal{F}\}$. Let $\psi_{\mathcal{B}}$ be a quantum register initialized with state $|\psi_{\mathcal{B}}\rangle$. Let U_m and M_1 be the unitary(s) and measurement associated with a move (m). We define $Measure(M_1, \underline{a}, \psi_{\mathcal{B}})$ to be a subroutine that applies a quantum circuit to $\psi_{\mathcal{B}}$ to encode the measurement M_1 into ancilla qubit $\underline{a}$, measures $\underline{a}$, and returns the result (a) as well as the newly projected state $\psi'_{\mathcal{B}}$. To execute m on the game we apply the following procedure.*

  procedure APPLY$(U_m, M_1, \psi_{\mathcal{B}}, \vec{v}, \mathcal{F})$
    if $\mathcal{P}_m(\vec{v}, \mathcal{F}) = true$ **then**
      $\underline{a} \leftarrow \underline{0}$
      $(a, \psi'_{\mathcal{B}}) \leftarrow Measure(M_1, \underline{a}, \psi_{\mathcal{B}})$
      if $a = 1$ **then**
        $\psi'_{\mathcal{B}} \leftarrow U_m \psi'_{\mathcal{B}}$
      end if
      if $\psi'_{\mathcal{B}} \neq \psi_{\mathcal{B}}$ **then**
        Update $\vec{v}$ to reflect the correct pieces.
        Update castling and e.p. flags in $\mathcal{F}$ appropriately.
        Update $\mathcal{F}_c$ to indicate change of player.
      end if
    end if
  end procedure

The following sections will detail the types of moves available to all pieces in the game. All pieces in Quantum Chess follow the same movement patterns as in standard Chess. For a full description of the patterns see Article 2 of the FIDE Laws of Chess [3]. Unlike in standard Chess, we do not define a notion of "check" for Quantum Chess. Therefore any rules that standard Chess applies to the legality of a King's moves when under check do not apply to Quantum Chess. It is perfectly legal to move one's King into a position where it is being attacked by an opponent's piece, or to castle through a position that is being attacked.

Each section will define the possibility equation for the move being constructed from the perspective of the white player.[8] For this it will be helpful to define the notion of a valid move:

Definition 3 *Valid: A move is valid if $s - t$ is a valid movement pattern for piece v_s under the rules of standard Chess. In functional form:*
  $valid(t, s, v_s) = True$, if t is reachable from s by piece v_s under standard Chess rules.

After defining a move's possibility equation the measurement operators M_0 and M_1 will be built if necessary. Finally the unitary procedure needed to update the occupancy superposition will be described.

---

[8] For the black player simply substitute capital FEN letters for lower case ones.

8.1 Standard Jump

$$|c3, b1\rangle = a\,|01\rangle + b\,|11\rangle + c\,|10\rangle \qquad\qquad |c3, b1\rangle = ia\,|10\rangle + b\,|11\rangle + ic\,|01\rangle$$

$$\{v_t, v_s\} = \{N, N\} \qquad\qquad\qquad \{v_t, v_s\} = \{N, N\}$$
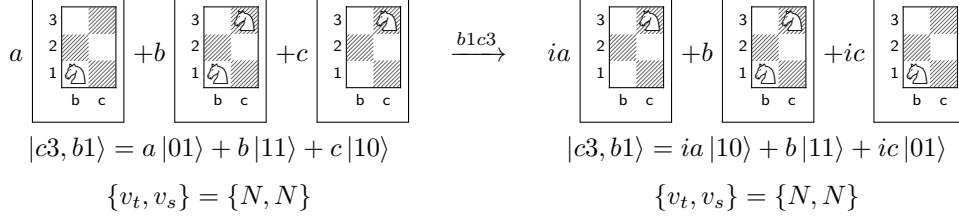
Figure 1: Example of Standard Jump $b1c3$ acting on a superposition. The classical piece information remains unchanged, but the amplitudes of the basis states have changed.

The Standard Jump is the Quantum Chess equivalent of the standard chess move for Knights and Kings (see figure 1).[9] These pieces do not care about a path so we will make use of the Jump unitary (eq. 5). This move attempts to fully swap a piece between the source(s) position and the target(t) position. We define the following possibility equation:

$$\mathcal{P}_{SJ} = (v_s \in \{N, K\}) \wedge valid(t, s, v_s) \wedge ((v_t = 0) \vee (v_t = v_s)) \tag{14}$$

This move will never result in Double Occupancy thus we can follow procedure 1 with $U_m = U_{jump}$. Figure 2 shows a simple quantum circuit that applies the Standard Jump to the appropriate qubits for the occupancy superposition.
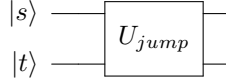

Figure 2: Quantum circuit diagram for applying the Standard Jump move to the $source(s)$ $target(t)$ qubits from the quantum state $|\psi_{\mathcal{B}}\rangle$.

8.2 Blocked Jump

The Blocked variant of the Jump has the following possibility equation:

$$\mathcal{P}_{BJ} = (v_s \in \{N, K\}) \wedge valid(t, s, v_s) \wedge (v_t \neq v_s) \wedge (v_t \in \{P, N, B, R, Q, K\}) \tag{15}$$

If this move is legal it will result in double occupancy (see figure 3). If a controlled operation were used to ensure the move acts only on the knight then the target square would be occupied by both a white knight and a white bishop after execution. Our state notation does not even support such a case. The basis states $|01\rangle$ and $|10\rangle$ are mutually exclusive for this move, they cannot both exist in the state if we want our move to never produce double occupancy. We find the following complete subsets of mutually exclusive basis states for the $|t, s\rangle$ Hilbert space:

$$\mathcal{M}_0 = \{|11\rangle, |10\rangle\}, \mathcal{M}_1 = \{|01\rangle\}$$

---

[9]We may also use this option for any of the sliding pieces if they are moving to an adjacent square or if we know the path to be clear.

$$|p, c3, b1\rangle = a\,|001\rangle + b\,|110\rangle \qquad\qquad |p, c3, b1\rangle = ia\,|010\rangle + b\,|110\rangle$$

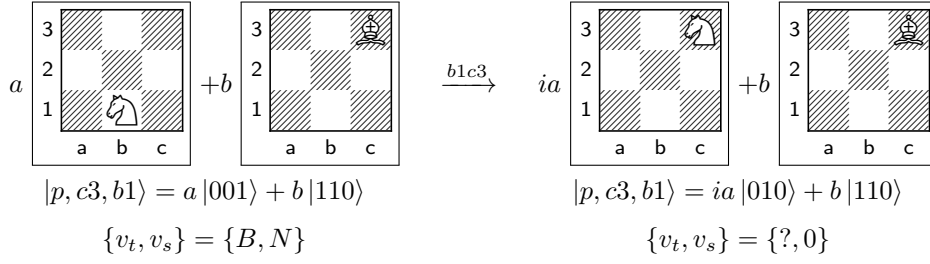$$\{v_t, v_s\} = \{B, N\} \qquad\qquad\qquad \{v_t, v_s\} = \{?, 0\}$$

Figure 3: Double Occupancy would occur if we attempted move b1c3 with the target (c3) occupancy encoded into the path (p) ancilla to act as a control for the swap. This must be prevented by applying a projective measurement before executing the move.

From these two subsets we can construct two measurement operators, M_0 and M_1, with $|00\rangle$ included in either. Here we choose the following construction:

$$M_0 = |10\rangle\,\langle 10| + |11\rangle\,\langle 11| \tag{16a}$$

$$M_1 = |00\rangle\,\langle 00| + |01\rangle\,\langle 01| \tag{16b}$$

This can be reduced to a simple measurement of the target qubit in the computational basis. We can thus apply procedure 2 with $U_m = U_{jump}$ (eq. 5). Figure 4 shows a simple quantum circuit for applying the measurement and conditional move operator.
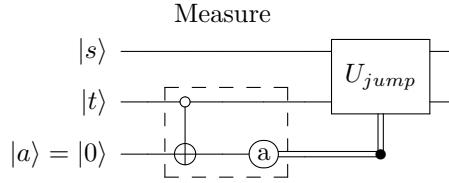

Figure 4: Quantum circuit diagram for applying the Blocked Jump move to the source (s) and target (t) qubits from the quantum register $\psi_{\mathcal{B}}$. First it encodes M_1 (eq. 16b) into measurement ancilla (a) and then conditionally applies U_{jump} (eq. 5) if the measurement outcome is 1.

8.3 Capture Jump

The Capture variant of the Jump has the following possibility equation:

$$\mathcal{P}_{CJ} = (v_s \in \{N, K\}) \wedge valid(t, s, v_s) \wedge (v_t \in \{p, n, b, r, q, k\}) \tag{17}$$

This move could result in double occupancy (see figure 5). Given this example we can deduce that the basis states $|11\rangle$ and $|10\rangle$ are mutually exclusive. The complete subsets of mutually exclusive basis states for the $|t, s\rangle$ Hilbert space are:

$$\mathcal{M}_0 = \{|10\rangle\}, \mathcal{M}_1 = \{|01\rangle, |11\rangle\}$$

We choose the following measurement operators:

$$M_0 = |10\rangle\,\langle 10| + |00\rangle\,\langle 00| \tag{18a}$$

$$M_1 = |01\rangle\,\langle 01| + |11\rangle\,\langle 11| \tag{18b}$$

$$|c, c3, b1\rangle = a\,|011\rangle + b\,|010\rangle \qquad\qquad |c, c3, b1\rangle = -a\,|110\rangle + b\,|010\rangle$$

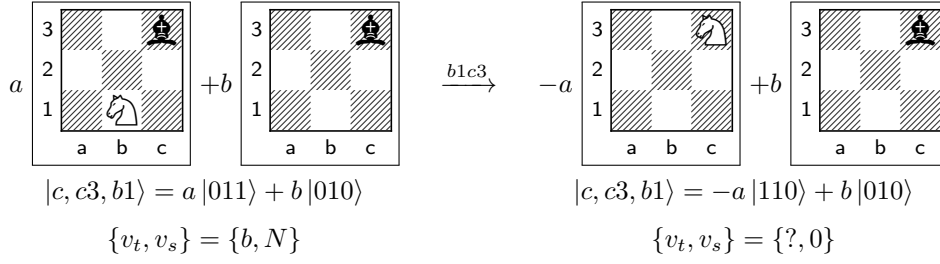$$\{v_t, v_s\} = \{b, N\} \qquad\qquad\qquad \{v_t, v_s\} = \{?, 0\}$$

Figure 5: Double Occupancy would occur if we attempted Capture Jump b1c3. This must be prevented by applying a projective measurement before executing the move. A "captured" ancilla (c) is added to hold the captured piece.

This can be reduced to a simple measurement of the source qubit in the computational basis. In order to perform capture in a unitary way we must expand the Hilbert space to include a "captured" ancilla ($\underline{c}$) initialized in state $|0\rangle$. We can then apply procedure 2 with $U_m = U_{jump}(s, t)U_{jump}(t, c)$. The circuit in figure 6 illustrates the move execution including the encoding of M_1 into an ancilla qubit $\underline{a}$ for the *Measure* subroutine.
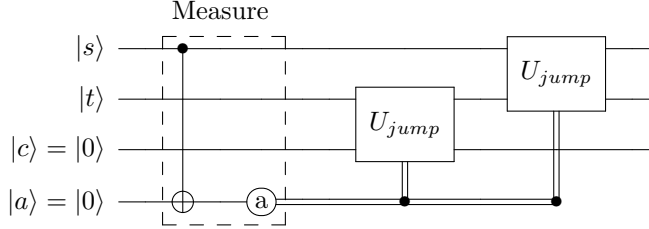

Figure 6: Quantum circuit diagram for applying the Capture Jump move to the source (s) and target (t) qubits from the quantum register $\underline{\psi_{\mathcal{B}}}$. A captured ancilla ($c$) is added to hold the captured piece. M_1 (eq. 18b) is encoded into measurement ancilla (a) and two U_{jump} (eq. 5) operations are conditionally applied based on the measurement outcome being 1.

8.4 Standard Slide

The Standard Slide is the Quantum Chess equivalent of the standard chess move for bishops, rooks, and queens (see figure 7). These pieces slide along a path so we must consider the occupancy of the squares between source and target. We introduce a "path" ancilla (p) and make use of the Slide unitary (eq. 6). The Standard Slide possibility equation takes almost the same form as that of the Standard Jump:

$$\mathcal{P}_{SS} = (v_s \in \{B, R, Q\}) \wedge valid(t, s, v_s) \wedge ((v_t = 0) \vee (v_t = v_s)) \tag{19}$$

This move will never result in Double Occupancy thus we can follow procedure 1 with $U_m = U_{slide}$. Figure 8a shows a quantum circuit that applies the Standard Slide where there are n squares in the path.

13

$$|p, a3, c1\rangle = a\,|001\rangle + b\,|101\rangle \qquad\qquad |p, a3, c1\rangle = ia\,|010\rangle + b\,|101\rangle$$

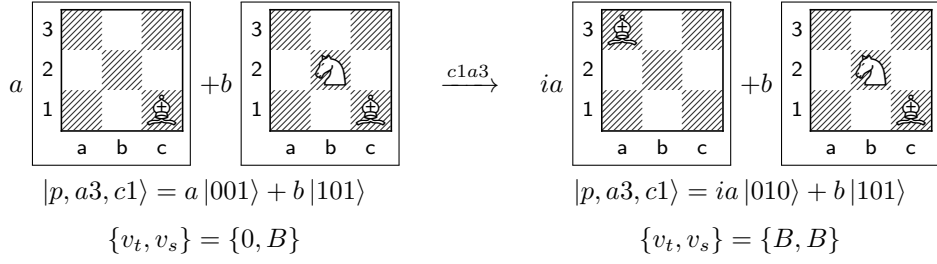$$\{v_t, v_s\} = \{0, B\} \qquad\qquad\qquad \{v_t, v_s\} = \{B, B\}$$

Figure 7: Example of Standard Slide $c1a3$ acting on a superposition. The move results in the bishop being entangled with the knight.

Figure 8: Quantum circuit diagrams for applying the Standard Slide move to the *source(s) target(t)* qubits with n squares in the path $\{p_1, ..., p_n\}$. In figure (a), a path ancilla will be in state $|0\rangle$ if no square in the path squares is occupied. This is in keeping with the definition of the Slide Unitary 6 acting on the $|p, t, s\rangle$ basis. Figure (b) shows a simplified circuit that has the same output with the slide operation converted to a zero-controlled U_{jump}.

8.5 Blocked Slide

Here we outline the Blocked variant of a Slide move. The possibility equation for the Blocked Slide takes almost the same form as that of the Blocked Jump:

$$\mathcal{P}_{BS} = (v_s \in \{B, R, Q\}) \wedge valid(t, s, v_s) \wedge (v_t \neq v_s) \wedge (v_t \in \{P, N, B, R, Q, K\}) \qquad (20)$$

This move will result in double occupancy (see figure 9). We find the following mutually exclusive subsets of basis states for the $|p, t, s\rangle$ Hilbert space:

$$\mathcal{M}_0 = \{|010\rangle, |011\rangle, |110\rangle, |111\rangle\}, \mathcal{M}_1 = \{|001\rangle\}$$

Given the freedom of choice for the other three basis states, $|000\rangle$, $|100\rangle$, and $|101\rangle$ in either measurement operator we construct the following measurement operators.

$$M_0 = |010\rangle \langle 010| + |011\rangle \langle 011| + |110\rangle \langle 110| + |111\rangle \langle 111| \qquad (21a)$$

$$M_1 = |000\rangle \langle 000| + |001\rangle \langle 001| + |100\rangle \langle 100| + |101\rangle \langle 101| \qquad (21b)$$

Result M_0 means no move occurs because the target square is occupied. This can be reduced to a simple measurement of the target qubit in the computational basis. We can thus apply procedure 2 with $U_m = U_{slide}$ (eq. 6). Figure 10 shows a quantum circuit for applying the measurement and conditional move operator.
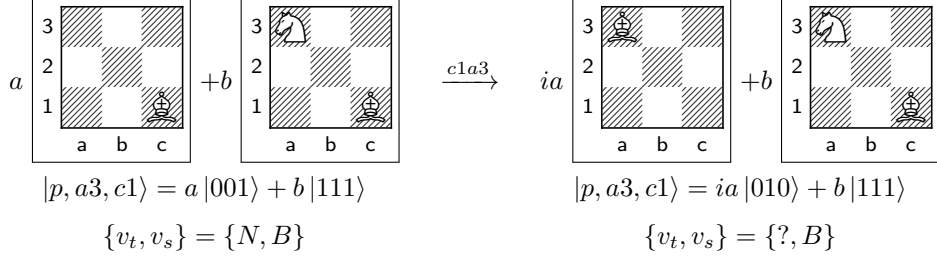
$$|p, a3, c1\rangle = a\,|001\rangle + b\,|111\rangle \qquad\qquad |p, a3, c1\rangle = ia\,|010\rangle + b\,|111\rangle$$

$$\{v_t, v_s\} = \{N, B\} \qquad\qquad\qquad \{v_t, v_s\} = \{?, B\}$$

Figure 9: Example of Blocked Slide $c1a3$ acting on a superposition. The occupancy of the target, a3, is encoded in the path control qubit. The move would result in Double Occupancy.
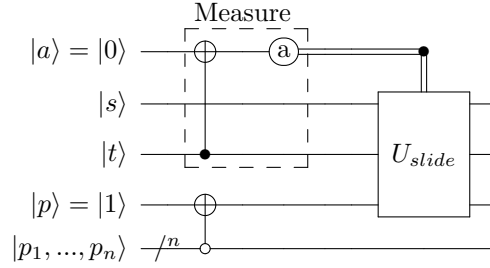

Figure 10: Quantum circuit diagram for applying the Blocked Slide move to the $source(s)$ $target(t)$. The line p_i represents the n qubits $\{p_1, ..., p_n\}$ in the path. The state of $|p\rangle$ will remain $|1\rangle$ if any square in the path is blocked.

8.6 Capture Slide

Here we outline the Capture variant of a Slide move. The possibility equation for the Capture Slide takes almost the same form as that of the Capture Jump:

$$\mathcal{P}_{CS} = (v_s \in \{B, R, Q\}) \wedge valid(t, s, v_s) \wedge (v_t \in \{p, n, b, r, q, k\}) \tag{22}$$

This move could result in double occupancy (see figure 11). Given this example we can deduce that the basis states $|011\rangle$ and $|111\rangle$ are mutually exclusive. We find the following complete subsets of mutually exclusive basis states for the $|p, t, s\rangle$ Hilbert space:

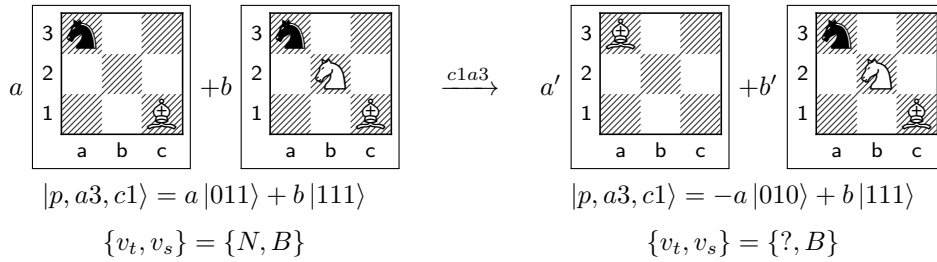$$\mathcal{M}_0 = \{|010\rangle, |110\rangle, |111\rangle\}, \mathcal{M}_1 = \{|001\rangle, |011\rangle\}$$

$$|p, a3, c1\rangle = a\,|011\rangle + b\,|111\rangle \qquad\qquad |p, a3, c1\rangle = -a\,|010\rangle + b\,|111\rangle$$

$$\{v_t, v_s\} = \{N, B\} \qquad\qquad\qquad \{v_t, v_s\} = \{?, B\}$$

Figure 11: Example of Capture Slide $c1a3$ acting on a superposition and resulting in Double Occupancy.

15

With these two subsets we construct our measurement operators, M_0 and M_1.

$$M_0 = |000\rangle \langle 000| + |010\rangle \langle 010| + |110\rangle \langle 110| + |111\rangle \langle 111| \tag{23a}$$

$$M_1 = |001\rangle \langle 001| + |011\rangle \langle 011| + |100\rangle \langle 100| + |101\rangle \langle 101| \tag{23b}$$

Result M_1 means either the path is not blocked and the source is occupied, or the path is blocked but the target is unoccupied. Adding a "captured" ancilla ($\underline{c}$) initialized in state $|0\rangle$ let's us maintain unitarity. We can then apply procedure 2 with $U_m = U_{slide}(s,t)U_{slide}(t,c)$. The circuit in figure 12 illustrates the move execution including the encoding of M_1 into an ancilla qubit $\underline{a}$ for the $Measure$ subroutine. Note each U_{slide} has been decomposed into a zero-controlled U_{jump}.
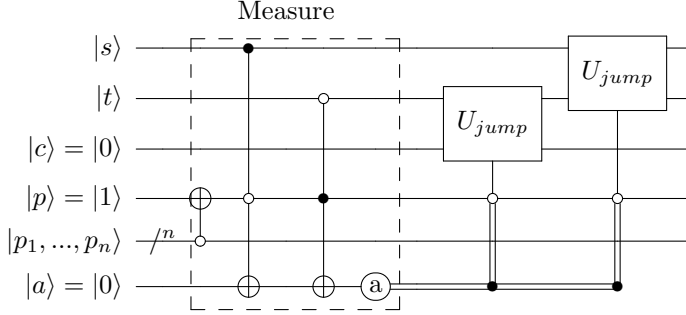

Figure 12: Quantum circuit diagram for applying the Capture Slide move to the $source(s)$ $target(t)$. The line p_i represents the n qubits $\{p_1, ..., p_n\}$ in the path. The state of $|p\rangle$ will remain $|1\rangle$ if any square in the path is blocked. Our measurement ancilla $|a\rangle$ will be in state $|1\rangle$ if $|p\rangle = |0\rangle$ and $|s\rangle = |1\rangle$, or if $|p\rangle = |1\rangle$ and $|t\rangle = |0\rangle$. Two zero-controlled U_{jump} are applied if we see measurement outcome 1.
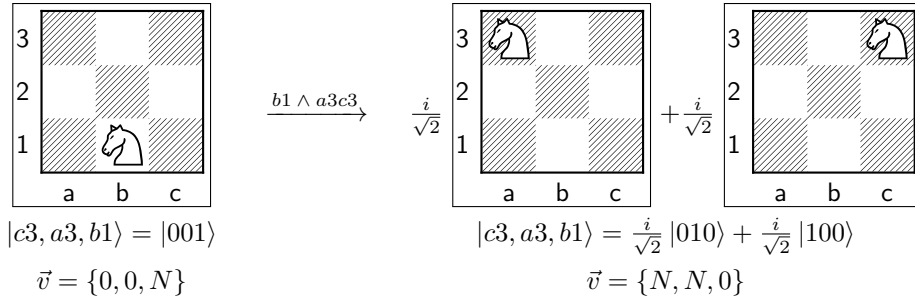
8.7 Split Jump

Figure 13: Split Jump $b1 \wedge a3c3$ acting on a knight on a single board.

The Split Jump is the move by which the player creates superposition with Knights and Kings (see figure 13). The move acts on three squares: source(s), target 1 (t1), and target 2 (t2). We define the following possibility equation:

$$\begin{aligned} \mathcal{P}_{SPJ} =& (v_s \in \{N, K\}) \wedge valid(t_1, s, v_s) \wedge valid(t_2, s, v_s) \wedge (t_1 \neq t_2) \\ & \wedge ((v_{t_1} = 0) \vee (v_{t_1} = v_s)) \wedge (v_{t_2} = 0 \vee v_{t_2} = v_s) \end{aligned} \tag{24}$$

16

This move will never result in double occupancy, thus we can apply procedure 1 with $U_m = U_{split\_jump}$ (eq. 7). Figure 14a shows a quantum circuit for the move.
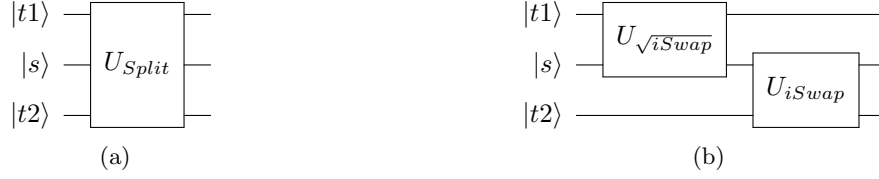

Figure 14: Quantum circuit diagrams for applying the Split Jump move to the source (s), target ($t1$), and target 2 ($t2$) qubits. In figure (a) The Split unitary is applied as a three-qubit gate. Figure (b) shows its decomposition into two gates, the $\sqrt{iSwap}$ from eq. 4b, and the $iSwap$ from eq. 4a.
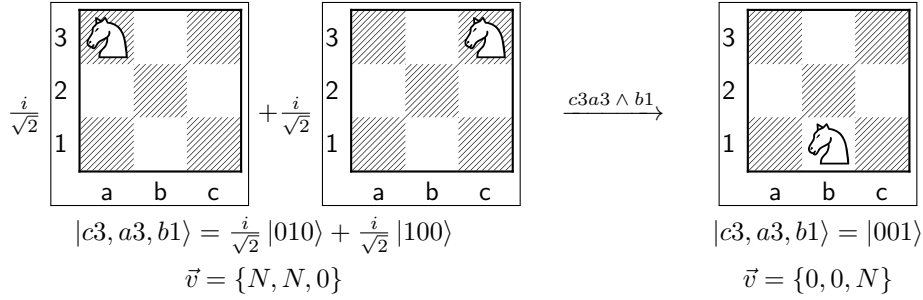
8.8 Merge Jump

Figure 15: Merge Jump $c3a3 \wedge b1$ is designed to *undo* the SplitJump $b1 \wedge a3c3$ on an otherwise empty board.

The Merge Jump is designed to undo the Split Jump (see figure 15). The move acts on three squares: source 1 (s1), source 2 (s2), and target (t). We define the following possibility equation:

$$
\begin{aligned}
\mathcal{P}_{MGJ} =&(v_{s_1} \in \{N, K\}) \wedge (v_{s_2} = v_{s_1}) \wedge valid(t, s_1, v_{s_1}) \wedge valid(t, s_2, v_{s_2}) \\
&\wedge (s_1 \neq s_2) \wedge (v_t = 0 \vee v_t = v_{s1})
\end{aligned}
\tag{25}
$$

This move will never result in double occupancy, thus we can apply procedure 1 with $U_m = U_{merge\_jump}$ (eq. 10). Figure 16a shows a quantum circuit for the move.

8.9 Split Slide

The Split Slide is the move by which the player creates superposition with bishops, rooks, and queens. The move acts on three squares: source(s), target 1 (t1), and target 2 (t2) (see figure 17). The possibility equation is similar to that of the Split Jump:

$$
\begin{aligned}
\mathcal{P}_{SPS} =&(v_s \in \{B, R, Q\}) \wedge valid(t_1, s, v_s) \wedge valid(t_2, s, v_s) \wedge (t_1 \neq t_2) \\
&\wedge ((v_{t_1} = 0) \vee (v_{t_1} = v_s)) \wedge (v_{t_2} = 0 \vee v_{t_2} = v_s)
\end{aligned}
\tag{26}
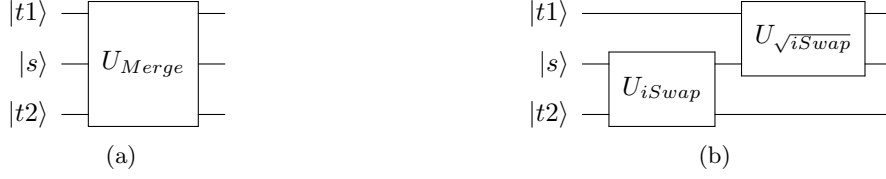$$

17

(a)   (b)

Figure 16: Quantum circuit diagrams for applying the Merge Jump move to the source 1 ($s1$), source 2 ($s2$), and target (t) qubits. In figure (a) The Merge unitary is applied as a three-qubit gate. Figure (b) shows its decomposition into two gates, the $\sqrt{iSwap}$ from eq. 4b, and the $iSwap$ from eq. 4a.

$|p2, p1, b1, a3, a1\rangle$

$= a\,|00001\rangle + b\,|01001\rangle$

$\{b1, a3, a1\} = \{0, 0, R\}$

$|p2, p1, b1, a3, a1\rangle$

$= \frac{ia}{\sqrt{2}}(|00010\rangle + |00100\rangle) + ib\,|01100\rangle$

$\{b1, a3, a1\} = \{R, R, 0\}$

Figure 17: Split Slide $a1 \wedge a3c1$ acting on a Rook in the presence of superposition. The move makes use of two path control qubits. Here we see the path (p_1) from s to t1 is partially blocked by a knight.

This move will never result in double occupancy, thus we can apply procedure 1 with $U_m = U_{split\_slide}$ (eq. 9). Figure 18 shows a quantum circuit for the move with U_{split_slide} decomposed to illustrate the path dependency.
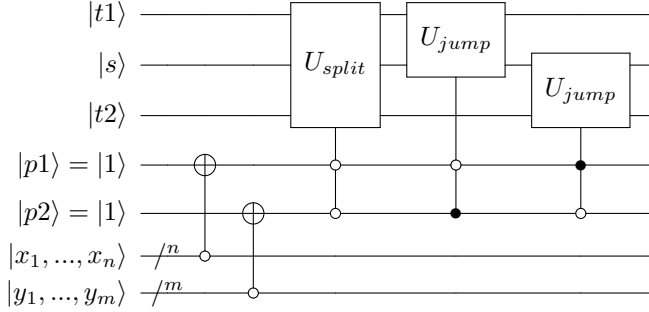

Figure 18: Quantum circuit for applying the Split Slide move to the source (s), target ($t1$), and target 2 ($t2$) qubits. The path from s to $t1$ is composed of squares $\{x_1, ..., x_n\}$ and the path from s to $t2$ is composed of squares $\{y_1, ..., y_m\}$. The controlled unitary operation reflects the blocked path effect expressed in eq. 9

$$|p2, p1, b1, a3, a1\rangle \qquad\qquad |p2, p1, b1, a3, a1\rangle$$

$$= \tfrac{a}{\sqrt{2}}(|00010\rangle + i\,|00100\rangle) + ib\,|01100\rangle \qquad = a\,|00001\rangle + b\,|01001\rangle$$

$$\{b1, a3, a1\} = \{R, R, 0\} \qquad\qquad \{b1, a3, a1\} = \{0, 0, R\}$$

Figure 19: Merge Slide $b1a3 \wedge a1$ acting on a Rook in the presence of superposition. The Merge undoes the Split that was illustrated in figure 17.

8.10 Merge Slide

The Merge Slide is designed to undo the Split Slide (see figure 19). Its possibility equation takes a form similar to that of the Merge Jump:

$$\mathcal{P}_{MGS} = (v_{s_1} \in \{B, R, Q\}) \wedge (v_{s_2} = v_{s_1}) \wedge valid(t, s_1, v_{s_1}) \wedge valid(t, s_2, v_{s_2}) \tag{27}$$
$$\wedge (s_1 \neq s_2) \wedge (v_t = 0 \vee v_t = v_{s_1})$$

This move will never result in Double Occupancy, thus we can use procedure 1 with $U_m = U_{merge\_slide}$ (eq. 12). Figure 20 shows a quantum circuit for the move with U_{merge_slide} decomposed to illustrate the path dependency.
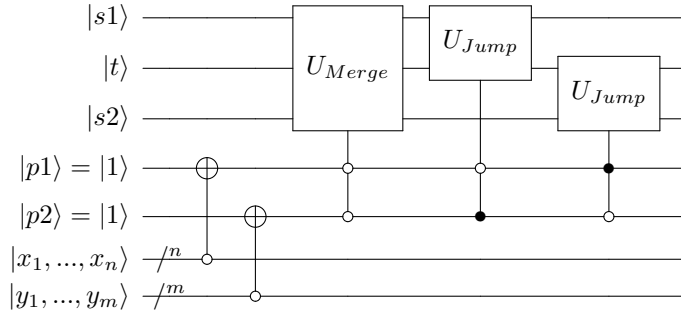
Figure 20: Quantum circuit for applying the Merge Slide move to the source 1 ($s1$), source 2 ($s2$), and target (t) qubits. The path from $s1$ to t is composed of squares $\{x_1, ..., x_n\}$ and the path from $s2$ to t is composed of squares $\{y_1, ..., y_m\}$. The controlled unitary operation reflects the blocked path effect illustrated in eq. 12

8.11 Pawn Movement

There are a number of special rules regarding pawn movement. We want pawns to behave similar to their counterpart in standard chess so here we detail the design of all of the ways in which they move, taking into account superposition and unitary movement restrictions. The following functions will be useful for defining the possibility of each type of pawn move.

- $step(t, s)$ = True, if t is one step in the forward direction from s.

- $two\_step(t, s) =$ True, if t is two steps in the forward direction from s, **and** s is on the first pawn rank for the current player (rank 2 for white and rank 7 for black).

- $diagonal(t, s) =$ True, if t is one step in either forward diagonal direction from s.

- $rank(x) =$ The rank of square x.

- $file(x) =$ The file of square x.

- $ep(t, s, \mathcal{F}) =$ True, if $file(t) = F_{ep} \in \{a, b, c, d, e, f, g, h\}$.

Pawn forward movement will be easy to handle with the tools we have already defined for Jump and Slide moves. However, Capture and En Passant will take some special care.

En passant (e.p.) is tricky in Quantum Chess. In standard Chess e.p. is possible if an opponent pawn begins its movement by advancing two steps and ends adjacent to one of our own pawns [3]. It has thus bypassed the square in which the player's pawn could have captured it. The player has have one turn in which we may chose to capture the opponent pawn by moving our pawn into the square it bypassed.

We will consider basis states of the form $|e.p.\ target, move\ target, source\rangle$, or $|ep, t, s\rangle$. We must determine what to do when the opponent Pawn being captured is in superposition. This can occur if it begins in superposition, and/or if it makes its two-step move through another piece that is in superposition. If there is another piece involved, and it belongs to the opponent, we would like to allow the capture of said piece. Thus pawn e.p. acts as a capturing move on both the target square (where attacking pawn ends up) as well as on the square occupied by the opponent's e.p. capturable pawn. There are three cases we must explore to fully determine how to implement e.p. in Quantum Chess:

1. Standard E.P.: The target square is either empty, or occupied in superposition by a pawn of the same color.

2. Blocked E.P.: The target square is occupied in superposition by a non-pawn belonging to the current player, and there exists a subspace where e.p. is valid.

3. Capture E.P.: The target square is occupied in superposition by an opponent piece and there exists a subspace where e.p. is valid.

In the following sections we fully detail all types of Pawn movement.

8.11.1 Pawn Step

The Pawn step is the simplest of pawn moves. In standard chess a pawn may always move one square forward, so long as the target square is not occupied by any other piece. The possibility equation for a Pawn Step is:

$$\mathcal{P}_{PS} = (v_s = P) \wedge (v_t = 0 \vee v_t = P) \wedge step(t, s) \tag{28}$$

This move will never result in Double Occupancy thus we can follow procedure 1 with $U_m = U_{jump}$. This is the same as what was defined for the Standard Jump. We can use the circuit shown in figure 2 to perform the move.

$$|b3, b2\rangle = a\,|01\rangle + b\,|10\rangle$$

$$\vec{v} = \{P, P\}$$

$$|b3, b2\rangle = i(a\,|10\rangle + b\,|01\rangle)$$

$$\vec{v} = \{P, P\}$$
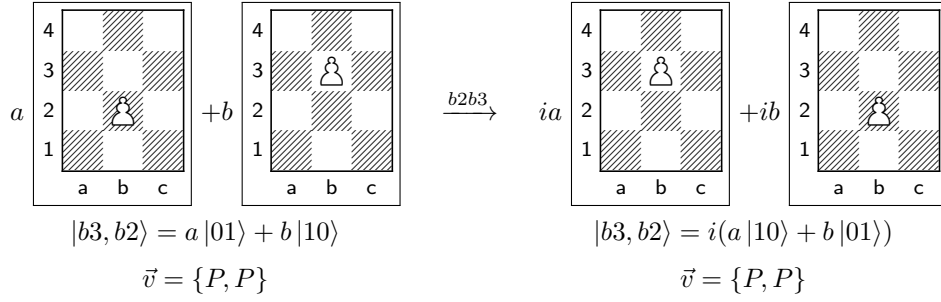
Figure 21: A pawn step is possible if the target is occupied by a pawn of the same color.

$$|b3, b2\rangle = a\,|01\rangle + b\,|10\rangle$$

$$\vec{v} = \{n, P\}$$

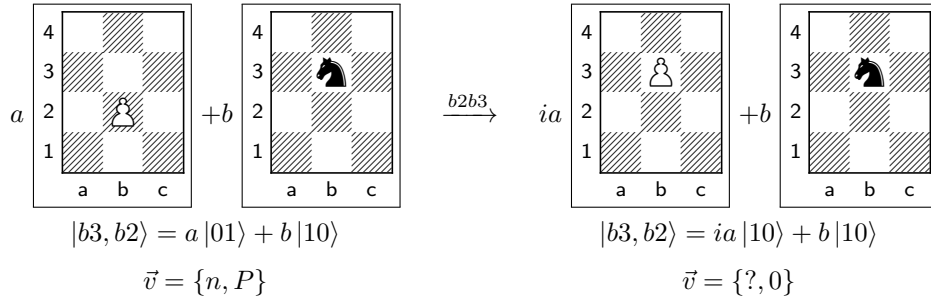$$|b3, b2\rangle = ia\,|10\rangle + b\,|10\rangle$$

$$\vec{v} = \{?, 0\}$$

Figure 22: A blocked pawn step will result in double occupancy. Here square b3 is occupied by both a black knight and a white pawn after the move.

8.11.2   Blocked Pawn Step

The Blocked Pawn Step is similar to the Blocked Jump with a different condition on the occupancy of the target square. Any piece blocks the move because pawns may not capture with forward movement. The possibility equation is:

$$\mathcal{P}_{BPS} = (v_s = P) \wedge (v_t \neq 0) \wedge (v_t \neq P) \wedge step(t, s) \tag{29}$$

The move will result in double occupancy (see figure 22). We use the same measurements defined for the Blocked Jump (eq. 16b), as well as the same procedure and circuit to perform the move (fig. 4).

8.11.3   Pawn Two Step

$$|b4, b3, b2\rangle = a\,|001\rangle + b\,|011\rangle \qquad\qquad |b4, b3, b2\rangle = ia\,|100\rangle + b\,|011\rangle$$
$$\vec{v} = \{0, N, P\} \qquad\qquad\qquad\qquad \vec{v} = \{P, N, P\}$$
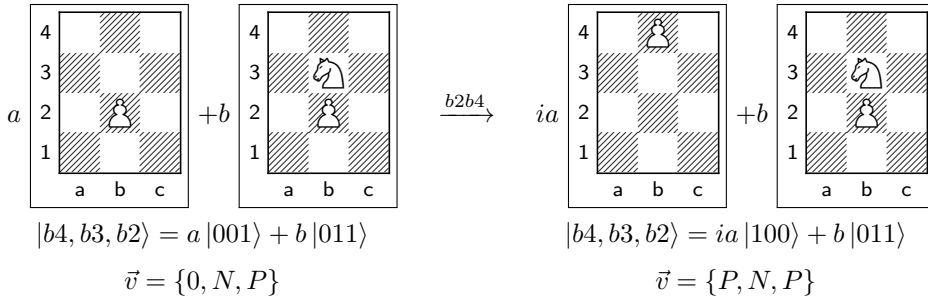
Figure 23: Pawn Two Step can entangle the pawn with another piece in superposition.

In standard chess the pawn two step is allowed on any given pawn's first move. If the path is not blocked then the pawn may slide two squares forward. Here we define the equivalent move in the presence of superposition. The possibility equation is

$$\mathcal{P}_{PTS} = (v_s = P) \wedge (v_t \neq 0) \wedge (v_t \neq P) \wedge two\_step(t, s) \tag{30}$$

This move acts similar to the Standard Slide (sec. 7). It wil never result in double occupancy. This move can entangle the pawn with any piece in superposition in the intermediate square (see figure 23), making it one of the means by which a pawn may exist in superposition. We can use the same procedure as that used for the Standard Slide, as well as the same circuit seen in figure 8. At the end of the move procedure we will set the classical e.p. information in $\mathcal{F}$ if move outcome is determined to be legal by definition 2.

8.11.4   Blocked Pawn Two Step

The Blocked Pawn Two Step is similar to the Blocked Slide introduced in section 8.5. The possibility equation for this move is:

$$\mathcal{P}_{BPTS} = (v_s = P) \wedge (v_t \neq 0) \wedge (v_t \neq P) \wedge two\_step(t, s) \tag{31}$$

This move will result in double occupancy (see figure 24). We can use the same measurements defined for the Blocked Slide (eq. 21b) as well as the same circuit (fig. 10). At the end of the move procedure we will set the classical e.p. information in $\mathcal{F}$ if our measurement result was 1 *and* the move outcome is determined to be legal by definition 2.

$$|b4, b3, b2\rangle = a\,|001\rangle + b\,|011\rangle$$

$$\vec{v} = \{N, 0, P\}$$

$$|b4, b3, b2\rangle = ia\,|100\rangle + b\,|011\rangle$$

$$\vec{v} = \{?, 0, P\}$$
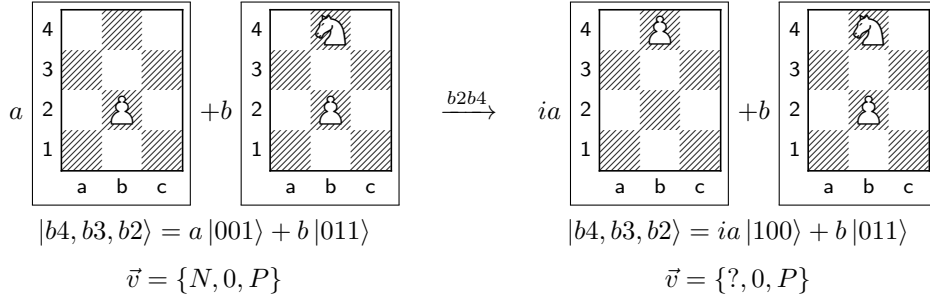
Figure 24: Blocked Pawn Two Step will result in double occupancy. Here square b4 would be occupied by both a white pawn and a white knight after the move.

8.11.5 Pawn Capture

$$|c3, b2\rangle = a\,|11\rangle + b\,|10\rangle$$

$$\vec{v} = \{n, P\}$$

$$|c3, b2\rangle = -a\,|10\rangle + b\,|10\rangle$$

$$\vec{v} = \{?, 0\}$$
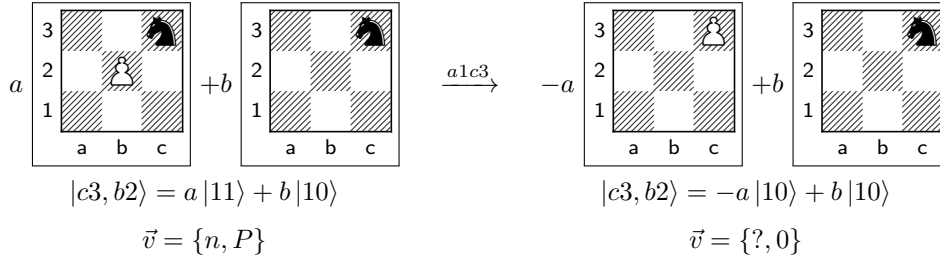
Figure 25: Double Occupancy would occur if we attempted the pawn capturing move b2c3.

In standard chess pawn capture is possible if the target square is one step in either diagonal direction, and it is occupied by an opponent piece. We present the following possibility equation:

$$\mathcal{P}_{PC} = (v_s = P) \wedge (v_t \in \{p, n, b, r, q, k\}) \wedge diagonal(t, s) \tag{32}$$

The move can result in double occupancy (see figure 25). We can use the measurements developed for the Capture Jump (eq. 18b). However, we can't use the same procedure or circuit. The Jump Capture allows for the piece to move and occupy the target even if the target is not present in a particular basis state. Pawn diagonal movement is only allowed in the case where it is capturing a piece. We must define a different U_m to use in procedure 2.

The move should only act non-trivially if both the moving pawn and the target are present so we will need to use controlled operations. We want our control ancilla to have state $|p\rangle = |0\rangle$ if both the source and target occupancy qubits are in state $|1\rangle$. Our controlled unitaries are then equivalent to U_{slide} (eq. 6) acting on the $|p, c, t\rangle$ and $|p, t, s\rangle$ subspaces. Thus we can apply procedure 2 with $U_m = U_{slide}(p, t, s)U_{slide}(p, c, t)$ as illustrated in the circuit in figure 26 where each U_{slide} has been converted into a zero-controlled U_{jump}.

8.11.6 Standard E.P.

In standard Chess e.p. is possible if the source square is occupied by a pawn belonging to the current player, and the *adjacent* square in the target direction is occupied by an opponent pawn
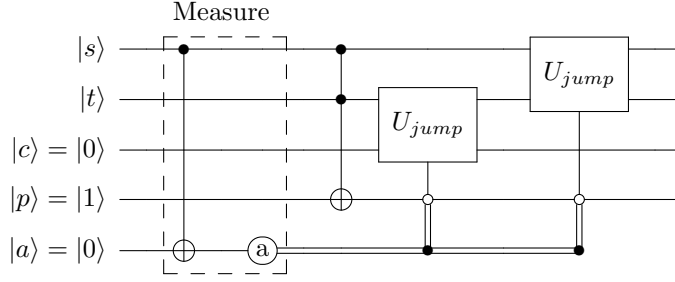
Figure 26: Quantum circuit diagram for applying the Pawn Capture move to the source (s) and target (t) qubits from the quantum register $\underline{\psi_{\mathcal{B}}}$. A captured ancilla ($c$) is added to hold the captured piece, and a "path" control ancilla (p) to encode the occupancy of s and t. M_1 (eq. 18b) is encoded into measurement ancilla (a) and two zero-controlled U_{jump} (eq. 5) operations are conditionally applied on with outcome 1 (source is occupied).
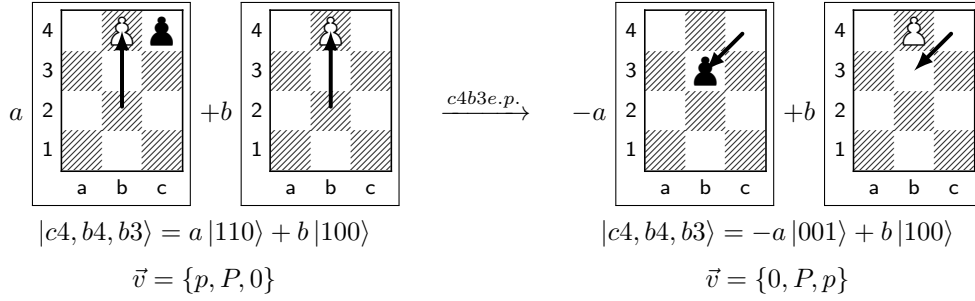
Figure 27: Standard E.P. can result in entanglement in the presence of superposition. Here the black pawn in b3 is entangled with the white pawn in b4.

that took two steps on the *previous turn*. The possibility equation for Standard E.P. acting on a source (s), target (t) and e.p. target (ep) is:

$$\mathcal{P}_{SEP} = (v_s = P) \wedge (v_t \in \{0, P\}) \wedge (v_{ep} = p) \wedge ep(t, s, \mathcal{F})^{10} \tag{33}$$

This move will never result in Double Occupancy. As with other capturing moves we must make use of the captured ancilla space, as well as a control ancilla that encodes the occupancy of both the source and the ep target. We must also introduce a control qubit (p) to ensure the move only acts non-trivially if both the source and e.p. target are occupied. With a design similar to Pawn Capture we can apply procedure 1 where $U_m = U_{slide}(p, t, s)U_{slide}(p, c, ep)$ as illustrated with the circuit in figure 28, where each U_{slide} is converted to a zero-controlled U_{jump}.

8.11.7 Blocked EP

The existence of superposition means it is possible for en passant to be a legal move where the pawn being captured is entangled with another piece of the same color as the pawn performing the move. The possibility equation for Blocked E.P. is:

$$\mathcal{P}_{BEP} = (v_s = P) \wedge (v_t \in \{N, B, R, Q, K\}) \wedge (v_{ep} = p) \wedge ep(t, s, \mathcal{F}) \tag{34}$$

---

[10]The constraint ($v_{ep} = p$) is redundant since $ep(t, s, \mathcal{F})$ should only return true if ep is a valid target, meaning it must be occupied by an opponent pawn that has just jumped forward two squares. It is explicitly included for clarity.
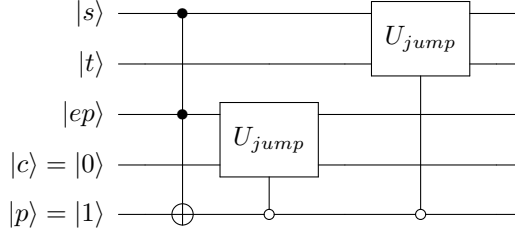
Figure 28: Quantum circuit diagram for applying the Standard E.P. move to the source (s), target (t), and e.p. target (ep) qubits from the quantum register $\psi_{\mathcal{B}}$. A captured ancilla (c) is added to hold the captured piece. Two zero-controlled $\overline{U_{jump}}$ (eq. 5) operations are applied with control $|p\rangle = |0\rangle$ if both the source and ep target are occupied.

$$|c4, b4, b3, b2\rangle = a\,|1011\rangle + b\,|1100\rangle \qquad\qquad |c4, b4, b3, b2\rangle = a\,|1011\rangle - b\,|0010\rangle$$

$$\vec{v} = \{p, P, n, P\} \qquad\qquad\qquad\qquad \vec{v} = \{p, 0, ?, P\}$$
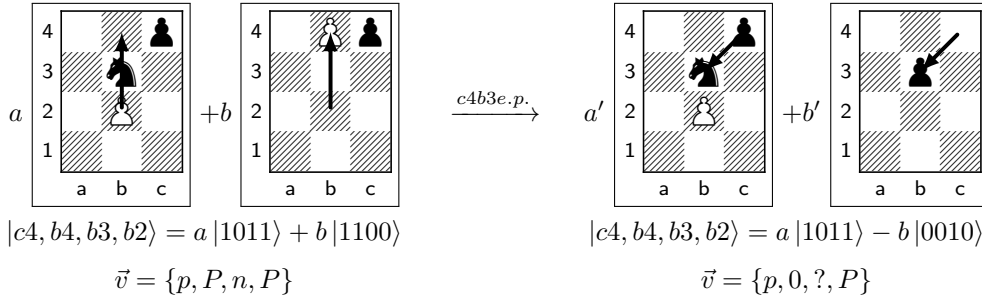
Figure 29: Blocked E.P. is possible when the pawn being captured is entangled with another piece of the opposite color. Double occupancy occurs in this example.

In this scenario the entangled piece will always be blocking the square the that the pawn is moving to. We can use the same measurements introduced for the Blocked Jump (eq. 16b). Thus we can apply procedure 2 with $U_m = U_{slide}(p, t, s)U_{slide}(p, c, ep)$ as illustrated with the circuit in figure 30, where each U_{slide} is converted to a zero-controlled U_{jump}.

8.11.8 Capture EP

Capture E.P. is a special case of en passant where the target square is occupied by an opponent piece that is entangled with the pawn being captured by e.p. (see figure 31). In this case the capture move is legal both under e.p. conditions, as well as conditions where normal pawn capture would be legal. This move can result in Double Occupancy in the presence of superposition. We can use the same measurement operator designed for the Capture Jump (eq. 18b).

To perform the move we make use of two captured ancilla squares and a single control (p). The control should be set to $|0\rangle$ if the source is occupied, and either the target is occupied or the e.p. target is occupied. We can then follow procedure 2 with $U_m = U_{slide}(p, t, s)U_{slide}(p, c_2, t)U_{slide}(p, c_1, ep)$ as illustrated with the circuit in figure 32, where each U_{slide} has been converted into zero-controlled U_{jump}.

8.12 Castling

We treat castling as a purely classical move, meaning we do not allow castling to be used in a Split or Merge. Similar to standard chess, the conditions under which castling is legal are

Figure 30: Quantum circuit diagram for applying the Blocked E.P. move to the source (s), target (t), and e.p. target (ep) qubits from the quantum register $\psi_{\mathcal{B}}$. A captured ancilla (c) is added to hold the captured piece, and a path ancilla $\overline{(p)}$ to encode the occupancy of s and t. Two zero-controlled U_{jump} (eq. 5) operations are conditionally applied if the measurement outcome is M_1 (target is not occupied).


$$|c4, b4, b3, b2\rangle = a\,|1011\rangle + b\,|1100\rangle + c\,|0011\rangle \qquad |c4, b4, b3, b2\rangle = -a\,|0011\rangle - b\,|0010\rangle + c\,|0011\rangle$$

$$\vec{v} = \{p, P, N, P\} \qquad\qquad \vec{v} = \{p, 0, ?, P\}$$

Figure 31: Capture E.P. is possible when the pawn being captured is entangled with another piece of the same color. Capture is legal on both boards in the superposition. In one case it is through e.p. In the other it is just standard pawn capture. Double Occupancy can occur in the presence of superposition of the moving pawn.

1. The king has not been involved in any previous move.

2. The rook which is participating in the castle has not been involved in any previous move.

There are some subtle differences with the classical rules worth noting. Superposition means we could find the game in a state where either piece has been involved in a move and yet has not actually moved. We do not make any distinction for this ruleset. We say ANY move that involves with the king or rook in any way invalidates castling, even if the result is a projection into a subspace where the king or rook did not actually change.[11] The other difference is that there is no concept of check, therefore any situation where check would invalidate castling in classical chess does not apply here.

8.12.1 King Side Castle

King side castling involves squares in columns $\{e, f, g, h\}$. The possibility equation for this move is:

$$\mathcal{P}_{KS} = (v_e = K) \wedge (v_h = R) \wedge (F_K = True) \tag{35}$$

---

[11]It is possible to allow castling to be both legal and not legal, it would just require a corresponding superposition of "legality" flags for the occupancy boards in the superposition.

Figure 32: Quantum circuit diagram for applying the Capture E.P. move to the source (s), target (t), and e.p. target (ep) qubits from the quantum register $\underline{\psi_{\mathcal{B}}}$. Two captured ancillas (c_1,c_2) are added to hold the captured pieces. Three zero-controlled U_{jump} (eq. 5) operations are conditionally applied if the measurement outcome is M_1 (source is occupied). Note: given the rules of the game and entanglement it is impossible for both the target and the e.p. target to be occupied in a single basis state if e.p. is valid.

$$|e, f, g, h\rangle = a\,|1001\rangle + b\,|1011\rangle \qquad\qquad |e, f, g, h\rangle = -a\,|0110\rangle + b\,|1011\rangle$$

$$\vec{v} = \{K, 0, B, R\} \qquad\qquad\qquad \vec{v} = \{K, R, ?, R\}$$

Figure 33: Double Occupancy would occur if we attempted castling move e1g1. This must be prevented by applying a projective measurement before executing the move.

This move can lead to double occupancy in cases of superposition of the target squares f and g (see figure 33). We find the following mutually exclusive subsets of basis states in the $|f, g\rangle$ basis:

$$\mathcal{M}_0 = \{|01\rangle, |10\rangle, |11\rangle, \mathcal{M}_1 = \{|00\rangle\}$$

From these two subsets we can begin constructing our measurement operators, M_0 and M_1 respectively.:

$$M_0 = |01\rangle\langle01| + |10\rangle\langle10| + |11\rangle\langle11| \tag{36a}$$

$$M_1 = |00\rangle\langle00| \tag{36b}$$

It is worth noting that if castling is determined to be possible, $\mathcal{F}_K = True$, it is not possible for squares e or h to be in superposition. We have defined castling to only be possible if neither the king or the rook has been involved in any previous move. If castling is possible, and we perform this measurement and find result 1, we can say with certainty that the state of squares e,f,g,h that we have projected into is $|1001\rangle$. We can thus apply procedure 2 with $U_m = U_{jump}(e, g)U_{jump}(f, h)$, as illustrated in the circuit in figure 34. These operators commute and can be performed simultaneously.
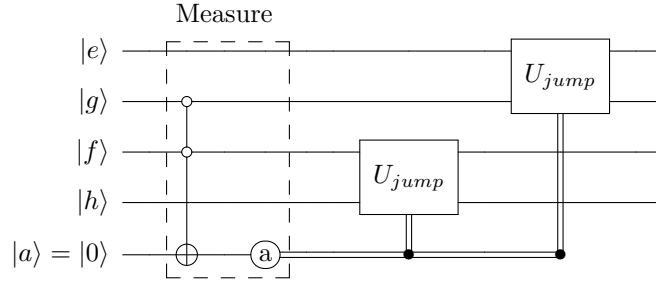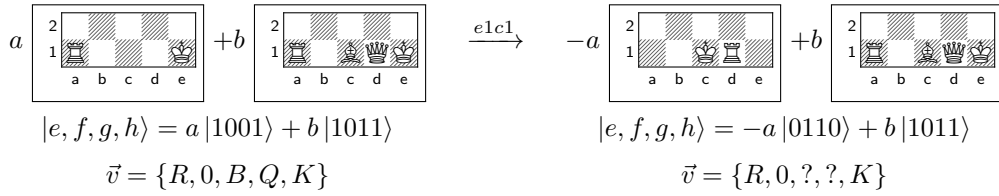
Figure 34: Quantum circuit diagram for applying the King Side Castle move to qubits representing the occupancy of squares e, f, g, h. M_1 (eq. 36b) is encoded into measurement ancilla (a) and two U_{jump} (eq. 5) operations are conditionally applied based on the measurement outcome.

8.12.2 Queen Side Castle

Figure 35: Double Occupancy would occur if we attempted castling move e1c1. This must be prevented by applying a projective measurement before executing the move.

Queen side castling involves squares in columns $\{a, b, c, d, e\}$. The possibility equation for this move is:

$$\mathcal{P}_{QS} = (v_e = K) \wedge (v_a = R) \wedge (F_Q = True) \tag{37}$$

This move can lead to double occupancy in cases of superposition of the target squares c and d (see figure 35). We can use the same measurement designed for King Side Castling (eq. 36b). As was the case with King Side Castling, if this move is determined to be possible, $F_Q = True$, it is not possible for squares a or e to be in superposition. If castling is possible, and we perform this measurement and find result 1, we can say with certainty that squares a and e are in state $|1\rangle$. The move must still be a controlled operation because square b may be in superposition. We can thus apply procedure 2 with $U_m = U_{slide}(b, a, d)U_{slide}(b, c, e)$, as illustrated in the circuit in figure 36 with the U_{slide} operators converted to zero-controlled U_{jump} operators.

9 Classical Simulation

When we consider the classical simulability of the game we must worry about the size ($\mathcal{S}$) of the superposition. As our state grows to include more possible board configurations we must contend with limits in both memory and execution time. If we wish for the game to be playable by the general public we need to keep the use of these resources at a reasonable level.

Modern CPUs have chips that run in the GHz range. If we imagine an algorithm for calculating the outcome of a move to be O(N) in the size of the superposition $\mathcal{S}$ then our run time
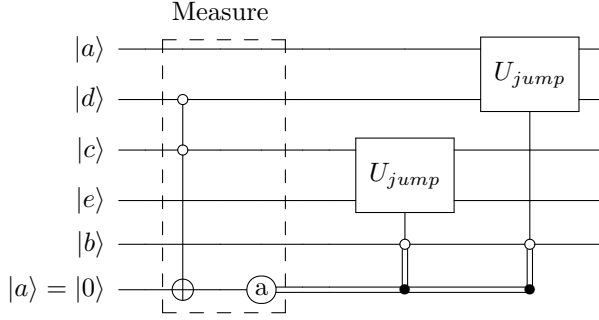
Figure 36: Quantum circuit diagram for applying the Queen Side Castle move to the qubits representing squares in files a, b, c, d, e. M_1 (eq. 18b) is encoded into measurement ancilla (a) and two zero-controlled U_{jump} (eq. 5) operations are conditionally applied if the measurement outcome is 1.

will be on the order of $\mathcal{S} \times 10^{-9}$ seconds. A reasonable maximum state size should likely not be much larger than 10^9 for single move calculation times on the order of one second. Considering our state representation this also gives us a memory size on the order of a gigabyte which is on the high end of reasonable assumption for modern personal machines. To account for the actual execution time of a move algorithm (more than one clock cycle), and in an attempt to decrease the worst case memory consumption we will shoot for a more modest maximum size of 10^6.

A basic approach to calculating the upper-bound on the size of the superposition would be to consider that we are simply storing the occupancy in superposition. For a given number of pieces (k), the maximum size of the superposition would simply be the number of ways of arranging k 1's on the board. Superposition allows for varying numbers of pieces to exist when some pieces have been partially captured. A naive upper bound would then be the sum of the maximum size for a given k over the possible values of k. The maximum number of pieces on a board is 32 so we get:

$$\mathcal{S} = \sum_{k=0}^{32} \binom{64}{k} \approx 1.0 \times 10^{19} \tag{38}$$

This is still quite large. We can improve on this by considering the effect of the No Double Occupancy rule. Each piece of a different value must be confined to a separate subset of squares. Any overlap will result in a collapse. Thus our formula becomes:

$$S_v = \sum_{m=0}^{m_v} \binom{s_v}{m} \tag{39a}$$

$$\mathcal{S} = \prod_{v_w \in V_w} S_{v_w} \prod_{v_b \in V_b} S_{v_b} \tag{39b}$$

where $V_w = \{P, N, B, R, Q, K\}$ and $V_b = \{p, n, b, r, q, k\}$ are the sets of distinct piece values for the white and black players respectively, m_v is the maximum multiplicity of the pieces of value v, and s_v is the number of squares occupied by the pieces of value v with the following restrictions:
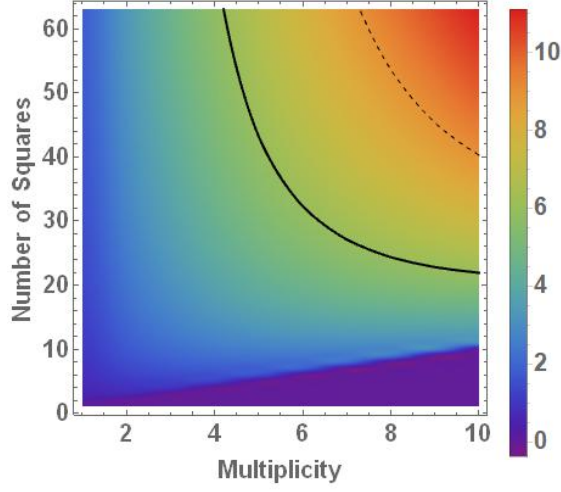
Figure 37: Heat map of the Log_{10} of the upper bound on the size of the superposition as we vary the multiplicity of a single piece in superposition and the number of squares that piece occupies. The solid black contour line shows the 10^6 cutoff, where the maximum size of the superposition is about 1 million basis states. The dashed line is the 10^9 contour.

$$\sum_{v \in V_w \cup V_b} s_v \leq 64 \tag{40a}$$

$$m_k = m_K = 1 \tag{40b}$$

$$m_{v_w} \leq (8 + m_{v_w,0}) - \sum_{u \in V_w \backslash P,K} (m_{u_w} - m_{u_w,0}) \tag{40c}$$

$$m_{v_b} \leq (8 + m_{v_b,0}) - \sum_{u \in V_b \backslash p,k} (m_{u_b} - m_{u_b,0}) \tag{40d}$$

where $m_{v,0}$ is the number of pieces of value v that start on the board. The restrictions on $m_{v_{w(b)}}$ come from the promotion rules of chess. Pieces that begin the game with multiplicity 2 can possibly achieve multiplicity 10 by promoting all 8 pawns. Let $\zeta_v = \{s_v, m_v\}$ denote the number of occupied squares and multiplicity for a piece value v. Solving equation 39b numerically we find that the new upper bound is about 7.9×10^{17} and occurs when $\zeta_p = \zeta_P = \{0, 0\}$, $\zeta_k = \zeta_K = \zeta_q = \zeta_Q = \{1, 1\}$, $\zeta_n = \zeta_N = \zeta_b = \zeta_B = \{3, 2\}$, and $\zeta_r = \zeta_R = \{24, 10\}^{12}$. This upper bound is dominated by the single piece with maximum multiplicity subset. We can see how the upper bound behaves as a player spreads a single piece in superposition over the board. Figure 37 shows that the simulation should remain tractable until we begin seeing a piece with $m_v > 5$ and $s_v > 40$.

10 Rules

Now that we have completed the design of the tools we will need to make Quantum Chess we can lay out the rules.

---

[12] Swapping rooks with knights will give the same upper bound

1. A 64x64 board is set up exactly as in standard Chess.

2. No square can ever be in a state of being occupied by more than one piece. A square is considered to be occupied by a piece if there is a non-zero probability of finding the piece in that square.

3. Players alternate turns with *white* moving first.

4. Pieces follow the same movement patterns as in standard Chess.

5. There is no concept of check or checkmate. Kings are captured like any other piece.

6. All pieces have access to a "standard equivalent" move.

7. Non-pawns may access Split and Merge moves.

8. A player may choose from any of the moves outlined in section 8. The player indicates the move to be done as follows:

   - Two Square Move: For any move involving a single source and single target for the moving piece. The player specifies a single source square occupied by a piece belonging to the player, and a target square that defines a valid possible move as determined by the possibility equations. This applies to all standard Chess equivalent moves, including e.p. and castling for which the other squares involved can be extrpolated by the indicated source and target.
   - Split Move: The player specifies a single source occupied by a piece belonging to the player, and two target squares that define a valid possible move as determined by the Split Jump (8.7) and Split Slide (8.9) possibility equations.
   - Merge Move: The player specifies two source squares each occupied by the same piece belonging to the player, and a single target square that defines a valid possible move as determined by the Merge Jump (8.8) and Merge Slide (8.10) possibility equations.

9. If the move has no effect the player repeats their turn with a different move. See 2

10. If a move results in a state where one player has zero probability of having a king on the board then the opposing player wins.

11. If a move results in a state where both players have zero probability of having a king on the board then the game ends in a draw.

11   Quantum Effects In Game

Now that we have finished the construction of Quantum Chess it would be helpful to determine whether it can actually result in non-trivial quantum effects. More specifically, does the construction achieve the goal of giving players access to superposition, entanglement, and interference? Superposition is trivially accessible through simple use of Split moves. Here we will show how to generate both entanglement and interference in game.
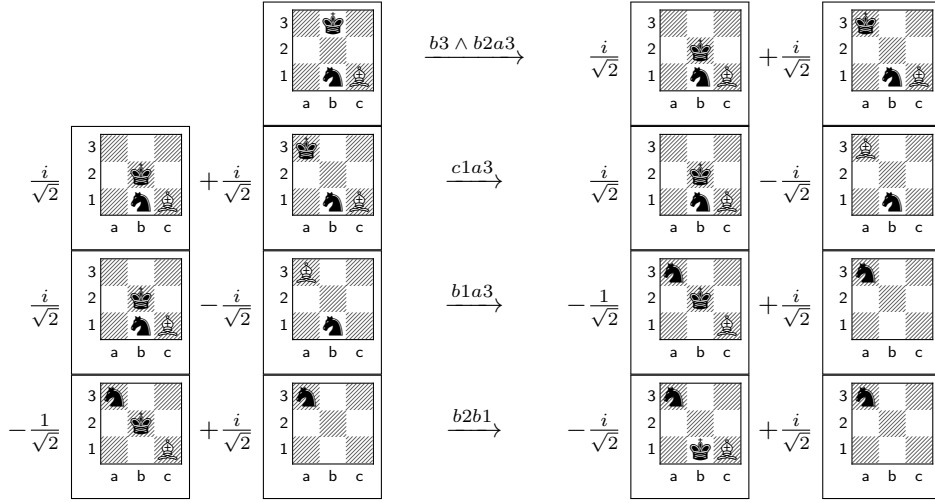
Figure 38: A Split Jump ($b3 \wedge b2a3$) followed by a Capture Slide ($c1a3$), a Capture Jump ($b1a3$), and a Standard Jump ($b2b1$) help create one of the Bell States. The occupancy of the final state of squares c1, b1, and a3 is $|a3, b1, c1\rangle = \frac{i}{\sqrt{2}}(-|111\rangle + |100\rangle)$

11.1 Entanglement: Quantum Chess Bell States

The prototypical entangled states are the Bell States [27]:

$$\left|\Phi^+\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{41}$$

$$\left|\Phi^-\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \tag{42}$$

$$\left|\Psi^+\right\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \tag{43}$$

$$\left|\Psi^-\right\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \tag{44}$$

These states are useful in a number of areas of quantum information including superdense coding [8] [27], and are used to rule out local hidden variable theories of quantum mechanics via the Bell-CHSH inequality [10]. Here we show how to generate the Bell States in Quantum Chess by applying various sequences of moves to a simple starting state.

Section 8.7 shows us that the Split Jump, which applies a simple Split unitary to a three-qubit state, gets us close to our first Bell state $|\Psi^-\rangle$. Consider applying the moves $a1 \wedge a2b1$, $b1a1$, and $b1a2$ to a board with a single king in position a1. Since there is only a single piece we will ignore the classical piece type information. The game begins in state $|a2, b1, a1\rangle = |001\rangle$. Applying the move unitaries we have

$$U_{split}(a1, a2, b1)|001\rangle = \frac{i}{\sqrt{2}}(|100\rangle + |010\rangle) \tag{45a}$$

$$U_{jump}(b1, a1)\frac{i}{\sqrt{2}}(|100\rangle + |010\rangle) = \frac{1}{\sqrt{2}}(i|100\rangle - |001\rangle) \tag{45b}$$

$$U_{jump}(b1, a2)\frac{1}{\sqrt{2}}(i|100\rangle - |001\rangle) = \frac{1}{\sqrt{2}}(-|010\rangle - |001\rangle) \tag{45c}$$

We can rewrite the final state in eq. 45c to get $|\Psi^+\rangle$:

$$|a2\rangle \otimes |b1, a1\rangle = -|0\rangle \otimes [\frac{1}{\sqrt{2}}(|10\rangle + |01\rangle)] = |0\rangle \otimes |\Psi^+\rangle \qquad (46)$$

We can phase rotate the $|01\rangle$ state by applying two more jump moves, $a1a2, a2a1$. In the basis $|a2, b1, a1\rangle$ we have:

$$U_{jump}(a1, a2)\frac{1}{\sqrt{2}}(-|010\rangle - |001\rangle) = \frac{1}{\sqrt{2}}(-|010\rangle - i|100\rangle) \qquad (47a)$$

$$U_{jump}(a2, a1)\frac{1}{\sqrt{2}}(-|010\rangle - i|100\rangle) = \frac{1}{\sqrt{2}}(-|010\rangle + |001\rangle) \qquad (47b)$$

which can be rewritten to get $|\Psi^-\rangle$:

$$|a2\rangle \otimes |b1, a1\rangle = -|0\rangle \otimes [\frac{1}{\sqrt{2}}(|10\rangle - |01\rangle)] = |0\rangle \otimes |\Psi^-\rangle \qquad (48)$$

To get $|\Phi^+\rangle$ and $|\Phi^-\rangle$ we will need to add more pieces. Consider the state and sequence of moves shown in figure 38. These moves act on squares c1, b1, b2, b3, and a3, and two captured ancillas (x_1, x_2) must be introduced which will be thrown away at the end. The evolution of the occupancy state in the basis $|x_2, x_1\rangle \otimes |a3, b3, b2, b1, c1\rangle$ is as follows:

$$U_{split}(a3, b2, b3)|00\rangle|01011\rangle = \frac{1}{\sqrt{2}}(i|00\rangle|00111\rangle + i|00\rangle|10011\rangle) \qquad (49a)$$

$$U_{slide}(b2, a3, c1)U_{slide}(b2, x_1, a3)\frac{1}{\sqrt{2}}(i|00\rangle|00111\rangle + i|00\rangle|10011\rangle)$$
$$= \frac{1}{\sqrt{2}}(i|00\rangle|00111\rangle - i|01\rangle|10010\rangle) \qquad (49b)$$

$$U_{jump}(b1, a3)U_{jump}(x_2, a3)\frac{1}{\sqrt{2}}(i|00\rangle|00111\rangle - i|01\rangle|10010\rangle)$$
$$= \frac{1}{\sqrt{2}}(-|00\rangle|10101\rangle + i|11\rangle|10000\rangle) \qquad (49c)$$

$$U_{jump}(b2, b1)\frac{1}{\sqrt{2}}(-|00\rangle|10101\rangle + i|11\rangle|10000\rangle)$$
$$= \frac{1}{\sqrt{2}}(-i|00\rangle|10011\rangle + i|11\rangle|10000\rangle) \qquad (49d)$$

If we trace out the ancillas and the qubits for empty squares we can rewrite the state in the basis $|a3, b1, c1\rangle$ to get $|\Phi^-\rangle$:

$$|a3\rangle \otimes |b1, c1\rangle = |1\rangle \otimes [\frac{i}{\sqrt{2}}(|00\rangle - |11\rangle)] = i|1\rangle \otimes |\Phi^-\rangle \qquad (50)$$

As we did in equations 47a and 47b, we can pick up a phase rotation on the $|11\rangle$ state by applying two more king moves, $b1b2$ and $b2b1$, to get $|\Phi^+\rangle$:

$$|a3\rangle \otimes |b1, c1\rangle = |1\rangle \otimes [\frac{i}{\sqrt{2}}(|00\rangle + |11\rangle)] = i|1\rangle \otimes |\Phi^+\rangle \qquad (51)$$
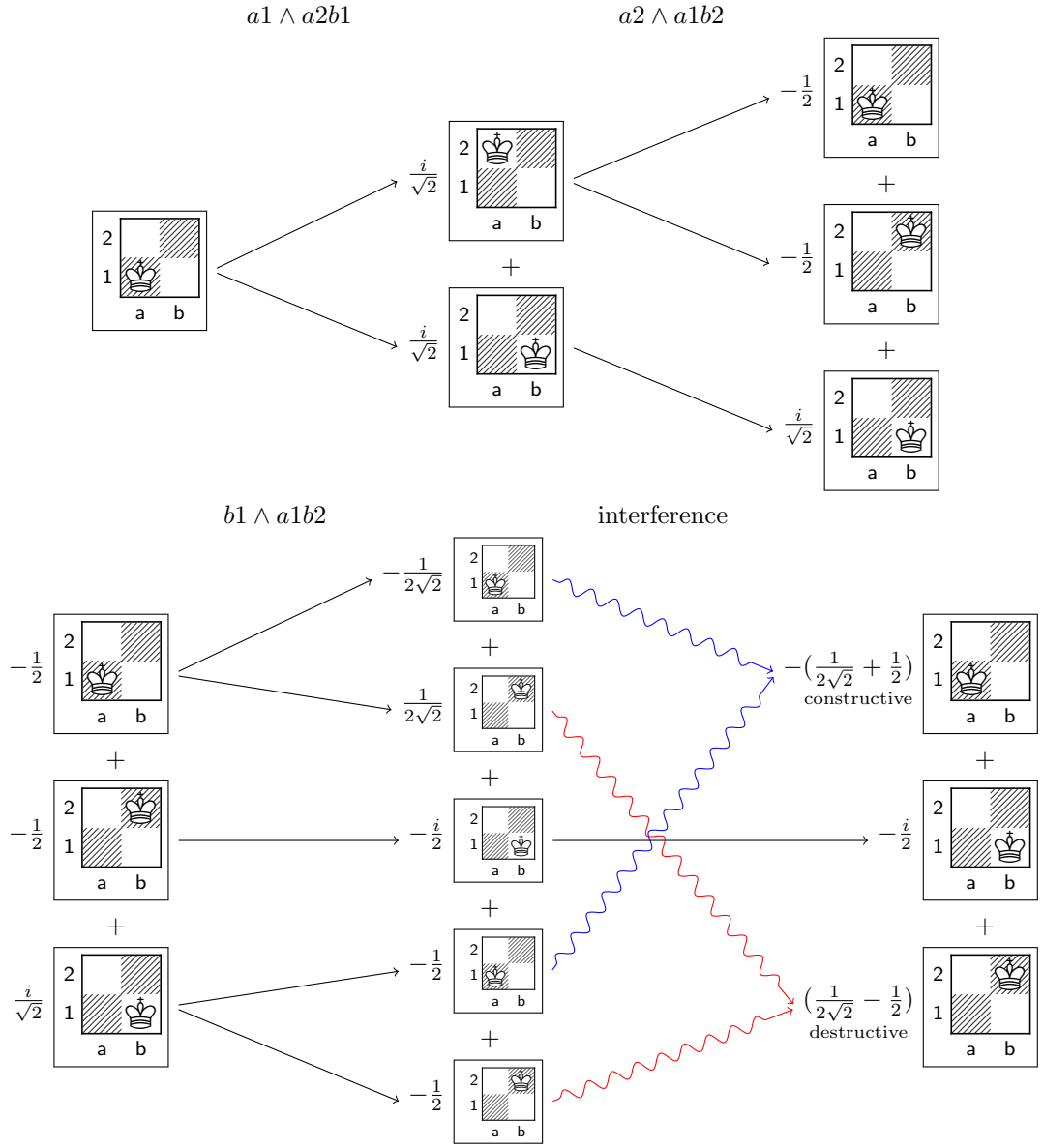
Figure 39: A series of three Split Jumps results in constructive and destructive interference.

11.2   Interference

Interference can be seen when like pieces interact. Consider a single king in square a1, and the sequence of moves $a1 \wedge a2b1$, $a2 \wedge a1b2$, and $b1 \wedge a1b2$. By applying the U_{split} unitary 7 to the appropriately ordered subspace[13] of the occupancy state in the basis $|b2, b1, a2, a1\rangle$ we have the following evolution:

$$|0001\rangle \xrightarrow{a1 \wedge a2b1} \frac{i}{\sqrt{2}}(|0010\rangle + |0100\rangle)$$

$$\frac{i}{\sqrt{2}}|0010\rangle + \frac{i}{\sqrt{2}}|0100\rangle \xrightarrow{a2 \wedge a1b2} -\frac{1}{2}(|0001\rangle + |1000\rangle) + \frac{i}{\sqrt{2}}|0100\rangle$$

$$-\frac{1}{2}|0001\rangle - \frac{1}{2}|1000\rangle + \frac{i}{\sqrt{2}}|0100\rangle \xrightarrow{b1 \wedge a1b2} -\frac{1}{2\sqrt{2}}(|0001\rangle - |1000\rangle) - \frac{i}{2}|0100\rangle$$

$$-\frac{1}{2}(|0001\rangle + |1000\rangle)$$

$$= -(\underbrace{\frac{1}{2\sqrt{2}} + \frac{1}{2}}_{\text{constructive}})|0001\rangle + (\underbrace{\frac{1}{2\sqrt{2}} - \frac{1}{2}}_{\text{destructive}})|1000\rangle - \frac{i}{2}|0100\rangle$$

If we square the amplitudes we get a probability distribution of $p(a1 = 1) \approx 0.73$, $p(a2 = 1) \approx 0.02$, and $p(b1 = 1) = 0.25$. Figure 39 shows how the splits act on each board in the superposition resulting in the final state. Note that the outcome depends on the order of the squares chosen during the move. If we follow the same procedure as above, but for the last split we instead do $b1 \wedge b2a1$ the probability distribution changes to $p(a1 = 1) \approx 0.02$, $p(a2 = 1) \approx 0.73$, $p(b1 = 1) = 0.25$. This difference may allow players to come up with interesting strategies involving interference.

12   Conclusion

Here we have detailed the process of designing Quantum Chess and discussed some of the considerations that went into certain design decisions. We see that it is possible to design a game using fully unitary dynamics and non-deterministic measurements while allowing the player to use quantum effects as a resource in game play. Through careful crafting of the rules of play we can see that it is possible to limit the size of the superposition so the game remains simulable and at the same time allow for significant quantum effects to be present (i.e. not just random). There is much future work that can be done with both Quantum Chess, and its future variants. There are many other classic games that we would also like to see receive a similar quantum treatment, such as Checkers and Reversi. We can also investigate implementation of these games using quantum resources. Recent work in the area of implementing games for quantum computers can be seen in *The History of Games for Quantum Computers* [31]. AI for these types of games is another area of research to be explored. Standard Chess has served as a test bed for AI development for decades, perhaps Quantum Chess can do the same for quantum AI development. The future is quantum, and quantum games can be fun.

---

[13]Correct ordering of the qubits is imperative when applying the unitary operations. For example, to apply $U_{split}(b1, a1, b2)$ to a state in the basis $|b2, b1, a2, a1\rangle$ one must first map the occupancy values to the basis $|b2, a1, b1\rangle$, apply the unitary, then map back.

13 Acknowledgements

This project had very humble beginnings. Initially it was not even related to Chess. It was a game where each player had a single piece type on a 5x5 board and the pieces moved using the square root of a swap. It accomplished the goals of superposition, entanglement, and interference, but it was decidedly not fun. The story likely would have ended there if not for my advisor, Todd Brun, who recognized its potential and put me in contact with someone he knew who "would be interested in this sort of thing". The introduction to Spiros Michalakis led to an amazing friendship and a project that has been more fulfilling than anything else I have worked on to date.

It is safe to say Quantum Chess would not be what is today without Spiros' enthusiasm and support. Our frequent meetings at Caltech's Institute for Quantum Information and Matter to discuss gameplay mechanics and how to create a fun experience helped shape what it would become. And then IQIM's video *Anyone Can Quantum* [19], written by Spiros and Jose Gonzalez, and subsequent outreach funding from the Institute gave Quantum Chess a life of its own which led to a number of incredible opportunities that ultimately pushed me to finish this long overdue paper.

A.I. for the game has been a challenge, and work on that front is still far from complete. To that end Spiros introduced me to Evert van Nieuwenburg who has been pivotal in its progress. We've had a number of discussions about A.I. for Quantum Chess and for his new version of Quantum Tic Tac Toe [30], as well as other quantum games, and I look forward to working on future projects with him.

I must also mention the Quantum Chess community, and particularly Elmar van Kordenoordt and M. Solé. Frequent discussions in the Quantum Chess Discord channel led to the development of Quantum Chess Algebraic Notation, and community input on gameplay mechanics and knowledge of Chess strategy has helped shape the current ruleset.

Finally I want to acknowledge my wife Laurie who pushed me to step outside my comfort zone, and the rest of my family and friends who have been incredibly supportive and enthusiastic about the entire project.

To all of you, and the many I have not mentioned by name but who have acted as sounding boards for my ideas and as moral support, I thank you.

References

[1] Algebraic notation (chess) - wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Algebraic_notation_(chess)`.

[2] Bra-ket notation. `https://en.wikipedia.org/wiki/Bra%E2%80%93ket_notation`.

[3] Fide laws of chess. `https://www.fide.com/FIDE/handbook/LawsOfChess.pdf`.

[4] Fosyth-edwards notation - wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Forsyth-Edwards_Notation`.

[5] qcraft wiki. `https://sites.google.com/a/elinemedia.com/qcraft/`.

[6] Quantum chess. `http://www.quantumrealmgames.com/`.

[7] S. G. Akl. On the importance of being quantum. *Parallel Processing Letters, Special Issue on Advances in Quantum Computation*, 20(3):275–286, 2010.

[8] Charles H. Bennett and Stephen J. Wiesner. Communication via one- and two-particle operators on einstein-podolsky-rosen states. *Phys. Rev. Lett.*, 69:2881–2884, Nov 1992.

[9] Christopher Cantwell. Quantum chess: Making quantum phenomena accessible. `https://www.youtube.com/watch?v=N3zFEpEPVF8`, Oct 2016.

[10] John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, 23:880–884, Oct 1969.

[11] Paul Dorbec and Mehdi Mhalla. Toward Quantum Combinatorial Games. *arXiv e-prints*, page arXiv:1701.02193, Jan 2017, 1701.02193.

[12] Jens Eisert, Martin Wilkens, and Maciej Lewenstein. Quantum games and quantum strategies. *Phys. Rev. Lett.*, 83:3077–3080, Oct 1999.

[13] A. Goff. Quantum tic-tac-toe: A teaching metaphor for superposition in quantum mechanics. *American Journal of Physics*, 74:962–973, nov 2006.

[14] Michal Gordon and Goren Gordon. Quantum computer games: quantum minesweeper. *Physics Education*, 45(4):372, 2010.

[15] Michal Gordon and Goren Gordon. Quantum computer games: Schrodinger cat and hounds. *Physics Education*, 47(3):346, 2012.

[16] David J. Griffiths. *Introduction to Quantum Mechanics (Second ed.)*. Cambridge University Press, New York, NY, USA, 10th edition, 2005.

[17] David Gross, Yi-Kai Liu, Steven T. Flammia, Stephen Becker, and Jens Eisert. Quantum state tomography via compressed sensing. *Phys. Rev. Lett.*, 105:150401, Oct 2010.

[18] Robert Heck, Oana Vuculescu, Jens Jakob Sørensen, Jonathan Zoller, Morten G. Andreasen, Mark G. Bason, Poul Ejlertsen, Ottó Elíasson, Pinja Haikka, Jens S. Laustsen, Lærke L. Nielsen, Andrew Mao, Romain Müller, Mario Napolitano, Mads K. Pedersen, Aske R. Thorsen, Carsten Bergenholtz, Tommaso Calarco, Simone Montangero, and Jacob F. Sherson. Remote optimization of an ultracold atoms experiment by experts and citizen scientists. *Proceedings of the National Academy of Sciences*, 115(48):E11231–E11237, 2018, https://www.pnas.org/content/115/48/E11231.full.pdf.

[19] IQIM. Anyone can quantum. `https://www.youtube.com/watch?v=Hi0BzqV_b44`, Jan 2016.

[20] Faisal Shah Khan, Neal Solmeyer, Radhakrishnan Balu, and Travis S. Humble. Quantum games: a review of the history, current state, and interpretation. *Quantum Information Processing*, 17(11):309, Oct 2018.

[21] E. Knill. Conventions for quantum pseudocode. `https://pdfs.semanticscholar.org/60d1/e63ca31555ec7013c5eb9a8a63788398fd14.pdf`, June 1996.

[22] Ulf Leonhardt. Quantum-state tomography and discrete wigner function. *Phys. Rev. Lett.*, 74:4101–4105, May 1995.

[23] A. I. Lvovsky and M. G. Raymer. Continuous-variable optical quantum-state tomography. *Rev. Mod. Phys.*, 81:299–332, Mar 2009.

[24] David A. Meyer. Quantum games and quantum algorithms. *arXiv e-prints*, pages quant–ph/0004092, Apr 2000, quant-ph/0004092.

[25] M. Nagy and N. Nagy. Quantum tic-tac-toe: A genuine probabilistic approach. *Applied Mathematics*, 3:1779–1786, 2012.

[26] João Pedro Neto. Quantum chess. `http://www.di.fc.ul.pt/~jpn/cv/heisenberg.htm`, Jan 2000.

[27] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, New York, NY, USA, 10th edition, 2011.

[28] André Ranchin. Quantum go. http://arxiv.org/abs/1603.04751v2.

[29] Jens Jakob W. H. Sørensen, Mads Kock Pedersen, Michael Munch, Pinja Haikka, Jesper Halkjær Jensen, Tilo Planke, Morten Ginnerup Andreasen, Miroslav Gajdacz, Klaus Mølmer, Andreas Lieberoth, and Jacob F. Sherson. Exploring the quantum speed limit with computer games. *Nature*, 532:210 EP –, Apr 2016.

[30] Evert van Nieuwenburg. Quantum tic tac toe. `http://quantumtictactoe.com`, 2019.

[31] James Wooten. The history of games for quantum computers. `https://medium.com/@decodoku/the-history-of-games-for-quantum-computers-a1de98859b5a`, Mar 2018.

[32] James R. Wootton. Getting the public involved in Quantum Error Correction. Proceedings of GSGS'17, 2017, 2017, 1712.09649. arXiv:1712.09649v1.