Chess Is Primitive Recursive

Vladimir A. Kulyukin

1 Introduction

A deterministic two-player board game is played by two players on a finite board. The players take turns in choosing a move out of finitely many moves. Such a game has a unique starting board and end boards can be classified as a win for either player or a draw. Examples of deterministic two-player board games are Tic Tac Toe [1] and its many variants (e.g., Qubic [2]), chess, and checkers. Let an epoch be the set of all boards reachable from the starting board after a given number of moves. A deterministic two-player game is primitive recursive if there exists a primitive recursive (p.r.) function $G(p, i, j)$, where p is a player and i and j are epoch numbers such that $i < j$, that returns for p an optimal sequence of moves from a given board in epoch i to a board in epoch j if it is p's turn to play at epoch i.

In a previous paper [3], we showed Tic Tac Toe to be p.r. In this paper, a proof is presented to show that chess is a deterministic p.r. game. In Sect. 2, several operators on Gödel numbers are defined. Section 3 presents a proof that chess is p.r. Conclusions are presented in Sect. 4.

2 Gödel Number Operators

All variables such as x, y, z, a, b, i, j, k, and t refer to natural numbers (i.e., elements of $\mathbb{N}$). The Greek letters α, γ, and ω with appropriate subscripts refer to auxiliary

V. A. Kulyukin (✉)
Department of Computer Science, Utah State University, Logan, UT, USA
e-mail: vladimir.kulyukin@usu.edu

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_30

functions. All functions map $\mathbb{N}$ to $\mathbb{N}$. Let $\langle x, y \rangle = z$, where $z = 2^x(2y+1) \dot{-} 1$, $x = \max_d\{2^d | (z+1)\}$, and $y = ((z+1)/2^x - 1)/2$. The pairing functions $l(z)$ and $r(z)$ are defined in (1) and shown to be p.r. in [4], where `min` is the minimalization function that returns the smallest natural number for which the predicate being minimalized is true.

$$\begin{aligned} l(z) &= \min_{x \leq z}\{(\exists y)_{\leq z}\{z = \langle x, y \rangle\}\} \\ r(z) &= \min_{y \leq z}\{(\exists x)_{\leq z}\{z = \langle x, y \rangle\}\} \end{aligned} \tag{1}$$

Let $(a_1, \ldots, a_n)$ be a sequence of numbers. The *Gödel number* of this sequence is defined in (2).

$$[a_1, \ldots, a_n] = \prod_{i=1}^{n} p_i^{a_i} \tag{2}$$

The function $[a_1, \ldots, a_n]$ is p.r., because $x \cdot y$, x^y, and p_i are p.r., as shown in [4]. The Gödel number of () is 1. Let $x = [a_1, \ldots, a_n]$. Then the function $(x)_i = a_i$, $1 \leq i \leq n$ defined in (3) is shown to be p.r. in [4].

$$(x)_i = \min_{t \leq x}\{\neg(p_i^{t+1} | x)\}. \tag{3}$$

The length of x is the position of the last non-zero prime power in the Gödel representation of x defined by the p.r. function $Lt(x)$ in (4).

$$Lt(x) = \min_{i \leq x}\{(x)_i \neq 0 \wedge (\forall j)_{\leq x}\{j \leq i \vee (x)_j = 0\}\} \tag{4}$$

Let the p.r. function $\lfloor x/y \rfloor$ return the integer part of the quotient x/y [4]. Let $\gamma_1(i, b) \equiv i > Lt(b) \vee i < 1 \vee b < 1$. The p.r. function `set` in (5) sets the i-th element of b to x.

$$\texttt{set}(b, i, x) = \begin{cases} 0 & \text{if } \gamma_1(i, b), \\ \lfloor \frac{b}{p_i^{(b)_i}} \rfloor \cdot p_i^x & \text{otherwise.} \end{cases} \tag{5}$$

Let

$$\begin{aligned} \texttt{cntx}(x, y, 0) &= 0, \\ \texttt{cntx}(x, y, t+1) &= \gamma_2(x, y, t, \texttt{cntx}(x, y, t)), \end{aligned}$$

where $s(x) = x + 1$ and

$$\gamma_2(x, y, t, c) = \begin{cases} 1 + c & \text{if } (y)_{s(t)} = x, \\ c & \text{otherwise.} \end{cases}$$

The p.r. function count in (6) returns the count of occurrences of x in y. Let $\texttt{in}(x, y) \equiv \texttt{count}(x, y) \neq 0$. In the remainder of the paper, $\texttt{in}(x, y)$ and $x \in y$ are used interchangeably.

$$\texttt{count}(x, y) = \texttt{cntx}(x, y, Lt(y)) \tag{6}$$

The p.r. function rap in (7) appends its first argument x to the right of the second argument y.

$$\texttt{rap}(x, y) = \begin{cases} [x] & \text{if } y = 0 \vee y = 1, \\ y \cdot p^{x}_{Lt(y)+1} & \text{otherwise.} \end{cases} \tag{7}$$

Let

$$\begin{aligned} \texttt{lcx}(x_1, x_2, 0) &= x_2, \\ \texttt{lcx}(x_1, x_2, t+1) &= \gamma_3(x_1, t, \texttt{lcx}(x, y, t)), \end{aligned}$$

where $\gamma_3(x, t, y) = \texttt{rap}((x)_{s(t)}, y)$. The p.r. function $\otimes_l$ in (8) places all elements of x_2, in order, to the left of the first element of x_1. Let $\otimes_l|_{i=1}^{k} x_i = x_1 \otimes_l x_2 \otimes_l \ldots \otimes_l x_k = (\ldots((x_1 \otimes_l x_2) \otimes_l \ldots \otimes_l x_k)\ldots) = (\ldots(x_1 \otimes_l (x_2 \otimes_l (\ldots \otimes_l (x_{k-1} \otimes_l x_k)\ldots)))\ldots)$.

$$x_1 \otimes_l x_2 = \texttt{lc}(x_1, x_2) = \texttt{lcx}(x_1, x_2, Lt(x_1)) \tag{8}$$

The p.r. function $\otimes_r$ in (9) places all elements of x_2, in order, to the right of the last element in x_1. Let $\otimes_r|_{i=1}^{k} x_i = x_1 \otimes_r x_2 \otimes_r \ldots \otimes_r x_k = (\ldots((x_1 \otimes_r x_2) \otimes_r \ldots \otimes_r x_k)\ldots) = (\ldots(x_1 \otimes_r (x_2 \otimes_r (\ldots \otimes_r (x_{k-1} \otimes_r x_k)\ldots)))\ldots)$. Note that $0 \otimes_l x = 1 \otimes_l x = 0 \otimes_r x = 1 \otimes_r x = x$.

$$x_1 \otimes_r x_2 = \texttt{lc}(x_2, x_1) \tag{9}$$

Let $\texttt{rmx}(x, y, 0) = [\,]$ and $\texttt{rmx}(x, y, t+1) = \gamma_4(x, y, \texttt{rmx}(x, y, t), s(t))$, where

$$\gamma_4(x, y, z, i) = \begin{cases} z & \text{if } (y)_i = x, \\ [(y)_i] \otimes_l z & \text{otherwise.} \end{cases}$$

The p.r. function $\texttt{rm}(x, y) = \texttt{rmx}(x, y, Lt(y))$ removes all occurrences of x from y.

Let $f(x)$ be a p.r. predicate and let $\texttt{mapx}_f(y, 0) = [\,]$ and $\texttt{mapx}_f(y, t+1) = \gamma_5(y, \texttt{mapx}_f(y, t), s(t))$, where

$$\gamma_5(y, z, i) = \begin{cases} z & \text{if } f((x)_i) = 0, \\ [(y)_i] \otimes_l z & \text{if } f((x)_i) = 1. \end{cases}$$

The p.r. function $\mathtt{map}_f(y) = \mathtt{mapx}_f(y, Lt(y))$ returns the list of occurrences of those elements x in y for which $f(x) = 1$.

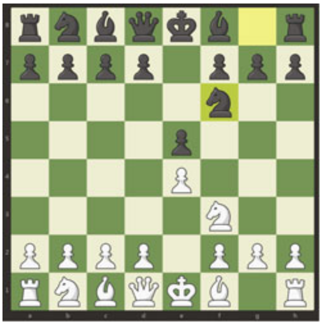
Let $\mathtt{pssx}(x, y, 0) = [\,]$ and $\mathtt{pssx}(x, y, t+1) = \gamma_6(x, y, \mathtt{pssx}(x, y, t), s(t))$, where

$$\gamma_6(x, y, z, i) = \begin{cases} [i] \otimes_l z & \text{if } (y)_i = x, \\ z & \text{otherwise.} \end{cases}$$

The p.r. function $\mathtt{pstn}(x, y) = \mathtt{pssx}(x, y, Lt(y))$ returns all positions of x in y.

3 Chess

We can encode a chess board as a Gödel number B with 64 elements (see Fig. 1). An empty cell is encoded as 1. A white pawn is encoded as $2 \leq n \leq 9$, the two white rooks are 10 and 17, the two white knights are 11 and 16, the two white bishops are 12 and 15, the white queen is 13, and the white king is 14. A black pawn is $18 \leq n \leq 25$, the two black rooks are 26 and 33, the two black knights are 27 and



26	27	28	29	30	31	1	33
18	19	20	21	1	23	24	25
1	1	1	1	1	32	1	1
1	1	1	1	22	1	1	1
1	1	1	1	6	1	1	1
1	1	1	1	1	16	1	1
2	3	4	5	1	7	8	9
10	11	12	13	14	15	1	17

Fig. 1 Chess board after 2 moves (above) and its Gödel number represented as a 2D matrix (below)

32, the two black bishops are 28 and 31, the black queen is 29, and the black king is 30. Let b_0 be the starting board. Then $(b_0)_j$, $1 \leq j \leq 16$, encode the black pieces, $(b_0)_j$, $49 \leq j \leq 64$, encode the white pieces, and $(b_0)_j$, $17 \leq j \leq 48$, encode the four empty rows in the middle of the board.

Let b be a board. The p.r. predicate $\gamma_7(b) \equiv \texttt{count}(14, b) = \texttt{count}(30, b) = 1$ ensures that b has exactly one white king ($i = 14$) and exactly one black king ($i = 30$). The predicate $\gamma_8(b) \equiv (\forall i)_{\leq 64}\{\{i \leq 1 \vee i = 14 \vee i = 30\} \vee \{\texttt{count}(i, b) \leq 1\}\}$ ensures that, unless i encodes an empty square ($i = 1$), the white king ($i = 14$), or the black king ($i = 30$), its count on b is 0 or 1. The predicate $\gamma_9(b) \equiv (\forall i)_{\leq 64}\{\{i \neq 1\} \vee \{32 \leq \texttt{count}(i, b) \leq 62\}\}$ ensures that the count of the empty spaces on b is between 32 (in the starting chess board) and 62 (when only the two kings remain on the board). The predicate $\texttt{valid}(b)$ in (10) is true if b is a valid chess board.

$$\texttt{valid}(b) \equiv \texttt{Lt}(b) = 64 \wedge \gamma_7(b) \wedge \gamma_8(b) \wedge \gamma_9(b) \tag{10}$$

For each piece x in a specific position, there is a set of positions reachable for x from that position. Let $z = [1, 2, \ldots, 64]$ be the Gödel number encoding the board positions, where 1 encodes the top-left corner of the board and 64 encodes the bottom-right corner of the board, and let L_j^k be the Gödel number whose elements are the board positions where chess piece j can move from position k. For example, L_{15}^1 is the list of positions reachable by the light-colored bishop 15 from position 1. The lists of positions reachable by bishop 15 from the positions along the main diagonal are

$$\begin{aligned}
L_{15}^{1} &= [10, 19, 28, 37, 46, 55, 64];\\
L_{15}^{10} &= [1, 3, 17, 19, 28, 37, 46, 55, 64];\\
L_{15}^{19} &= [1, 10, 5, 12, 26, 33, 28, 37, 46, 55, 64];\\
L_{15}^{28} &= [1, 10, 19, 7, 14, 21, 35, 42, 49, 37, 46, 55, 64];\\
L_{15}^{37} &= [1, 10, 19, 28, 16, 23, 30, 44, 51, 58, 46, 55, 64];\\
L_{15}^{46} &= [1, 10, 19, 28, 37, 32, 39, 53, 60, 55, 64];\\
L_{15}^{55} &= [1, 10, 19, 28, 37, 46, 48, 62, 64];\\
L_{15}^{64} &= [1, 10, 19, 28, 37, 46, 55].
\end{aligned}$$

Let W be the Gödel number whose elements are the white-colored cells on the chess board. Then L_{15} in (11) defines the Gödel number whose elements are all possible board positions for bishop 15.

$$L_{15} = \otimes_r|_{i \in W} L_{15}^i. \tag{11}$$

Such lists (i.e., Gödel numbers) can be computed in a p.r. fashion for all pieces and all positions on the board. Let $L = \otimes_r|_{i=2}^{33}[<i, L_i>]$ be the list of pairs $<i, L_i>$, where i denotes a chess piece and L_i is the list of all possible positions where i can move. The p.r. function in (12) returns, for each piece x at position i on b, all potentially reachable positions where x can move from i.

$$\mathtt{prp}(x, i, b) = \begin{cases} r((L_x)_{\gamma_{10}(i, L_x)}) & \text{if } \mathtt{valid}(b) \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

where $\gamma_{10}(i, L) = \min_{k \leq Lt(L)} \{i = l((L)_k)\}$ and $L_x = r((L)_{\gamma_{10}(x,L)}))$.

The p.r. predicate $\mathtt{wp}(x) \equiv 2 \leq x \leq 17$ is true if x is a white piece and the p.r. predicate $\mathtt{bp}(x) \equiv 18 \leq x \leq 33$ is true if x is a black piece. Let the predicate $\mathtt{arp}(x, i, b)$ be true if and only if position i on b is actually reachable for piece x (i.e., $\mathtt{prp}(x, i, b) = 1$ and i is not blocked by another piece). Since $\mathtt{arp}(x, i, b)$ can be defined in this manner by cases, each of which is defined in terms of p.r. predicates such as bp, wp, valid, Lt, rap, etc., and combinations of compositions and primitive recursions thereof, arp is p.r.

Let $z = [1, 2, \ldots, 64]$ be the Gödel number encoding the board positions and b a chess board. Let $f(x) \equiv \mathtt{arp}(x, j, b)$, where $j \in z$ and $\mathtt{wp}(x) = 1$ or $\mathtt{bp}(x) = 1$. Then the p.r. function $\mathtt{alst}(x, b) = \mathtt{map}_f(x, z)$ returns the Gödel number of actually reachable positions for x on b.

Let $\mathtt{bkp}(b) = (\mathtt{pstn}(30, b))_1$ be a p.r. function that returns the position of the black king on b. The black king is checked when there is a white piece (other than the white king encoded as 14) for which the current position of the black king is actually reachable. Formally, $\mathtt{bchk}(b) \equiv (\exists x)_{<34}\{\mathtt{wp}(x) \wedge x \neq 14 \wedge \gamma_{11}(x, b)\}$, where $\gamma_{11}(x, b) \equiv (\exists j)_{<65}\{j > 0 \wedge \alpha(x, j, b)\}$ and $\alpha(x, j, b) \equiv (\mathtt{pstn}(x, b))_1 = j \wedge \mathtt{in}(\mathtt{bkp}(b), \mathtt{alst}(x, b))$. The black king is mated if, when checked, it cannot move to any cell that is not actually reachable by a white piece. The p.r. predicate bmtd in (13) defines this logic.

$$\mathtt{bmtd}(b) \equiv \mathtt{bchk}(b) \wedge \gamma_{12}(b), \tag{13}$$

where $\gamma_{12}(b) \equiv (\forall j)_{<65}\{\neg \mathtt{in}(j, \mathtt{alst}(30, b)) \vee \gamma_{13}(j, b)\}$ and $\gamma_{13}(j, b) \equiv (\exists x)_{<34}\{\mathtt{wp}(x) \wedge \mathtt{in}(j, \mathtt{alst}(x, b))\}$. The same logic can be used to define a p.r. predicate $\mathtt{wmtd}(b)$ to return 1 when the white king is mated and 0, otherwise.

A draw by stalemate occurs when the player whose turn it is to move is not in check but has no legal move. The p.r. black stalemate predicate $\mathtt{bstlmt}(b) \equiv \neg \mathtt{bchk}(b) \wedge \gamma_{14}(b)$, where $\gamma_{14}(b) \equiv (\forall j)_{<65}[\neg \mathtt{bp}(j) \vee \mathtt{Lt}(\mathtt{alst}(j, b)) = 0]$, checks if the black king is not checked and no black piece has actually reachable positions. A white stalemate can be defined in the same fashion. All cases of the dead position rule can be defined as p.r. predicates with count, valid, in, =, and boolean combinations thereof.

A draw by repetition is achieved when the same position occurs three times in a row with the same player to move. The 50-move rule states that a game is a draw when the last 50 moves contain no capture or pawn move. The dead position rule applies when neither player can checkmate the opponent by any series of moves. The dead position rule applies to situations when there are only two kings left on the board, when one side has the king and a bishop and the other side has the king, when one side has the king and a knight and the other side has the king, when both sides have the king and a bishop and the bishops are both light-colored or dark-colored. The rules for the draw by repetition and the 50-move rule will be outlined below after we formalize the notion of the board history. Consider the p.r. function in (14).

$$\left(b\right)_j^x = \begin{cases} \mathtt{set}(b, j, x) & \text{if } \gamma_{15}(x, b) \\ 0 & \text{otherwise,} \end{cases} \tag{14}$$

where $\gamma_{15}(x, b) \equiv \mathtt{valid}(b) \wedge \{\mathtt{wp}(x) \vee \mathtt{bp}(x)\} \wedge \mathtt{in}(j, \mathtt{alst}(x, b))$. For example, the chess board in Fig. 1 is

$$((((b_0)_{37}^{6})_{29}^{22})_{46}^{16})_{22}^{32}.$$

Let b^1 be the list of all possible boards obtained from b by exactly one move of the white player, assuming that it is the white's turn to move.

$$b^1 = \otimes_r|_{x=2}^{17} \otimes_r |_{j=1}^{64}[(b)_j^x].$$

Let b^2 be the list of all possible moves obtained from b by exactly one move of the black player, assuming that it is the black's turn to move.

$$b^2 = \otimes_r|_{x=18}^{33} \otimes_r |_{j=1}^{64}[(b)_j^x].$$

Since both b^1 and b^2 are p.r., we can combine b^1 and b^2 into a single p.r. function $\mathtt{pm}(x, p)$ that maps the current board to the list of all possible boards obtained from it by exactly one move of either player.

$$\mathtt{pm}(b, p) = \begin{cases} b^1 & \text{if } p = 1 \wedge \mathtt{valid}(b) \\ b^2 & \text{if } p = 2 \wedge \mathtt{valid}(b) \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

Let $\mathtt{Z} = [b_{i_0}, b_{i_1}, \ldots, b_{i_k}]$ such that $\mathtt{valid}(b_{i_j}) = 1, 0 \leq j \leq k$. Let $p \in \{1, 2\}$. The p.r. function $\mathtt{scr}(\mathtt{Z}, p)$ in (16) takes a Gödel number that consists of valid boards and a player's number and returns another Gödel number that consists of successor boards such that each successor board is obtained from one of the boards in $\mathtt{Z}$ by exactly one move of p, assuming that it is p's turn to move.

$$\mathtt{scr}(\mathtt{Z}, p) = \mathtt{rm}(0, \otimes_r|_{i=1}^{\mathtt{Lt}(Z)} \mathtt{pm}((\mathtt{Z})_i, p)) \tag{16}$$

Let b_1 and b_2 be two boards and let $p \in \{1, 2\}$ be a player whose turn it is to play on b_1. The p.r. predicate $\mathtt{prn}(p, b_1, b_2) \equiv \mathtt{valid}(b_2) \wedge \mathtt{in}(b_2, \mathtt{pm}(p, b_1))$ is true when b_2 is in the Gödel number of the boards obtained from b_1 by exactly one move of p on it. In other words, b_1 is the parent of b_2. Let the p.r. function $\mathtt{B}(t)$, defined in (17), return the Gödel number that includes all boards, actually reachable from b_0 after t moves. We will refer to each $\mathtt{B}(t)$ as *epoch t*. Let $\mathtt{B}(0) = [\langle [b_0], 1\rangle]$.

$$\mathtt{B}(t+1) = [\langle \mathtt{scr}(l((\mathtt{B}(t))_1), \gamma_{16}(s(t)))\rangle], \tag{17}$$

where $\gamma_{16}(x) = 1$ if $\neg(2|x)$ and 2 if $2|x$. Let $G_0 = [b_0]$ and G_i, for $i > 0$, be the Gödel number encoding all boards actually reachable from the boards in G_{i-1} in 1 move by the appropriate player. Let $b \in G_i$, $G_{i-1} = l((\mathtt{B}(i-1))_1)$, and $p = r((\mathtt{B}(i-1))_1)$. The p.r. function $\mathtt{ipb}$ in (18) returns the index of the parent of b in G_{i-1} for $i > 0$.

$$\mathtt{ipb}(b, i) = \min_{t \leq Lt(G_{(i-1)})} \{\mathtt{prn}(p, (G_{i-1})_t, b)\} \tag{18}$$

The p.r. function $\mathtt{prb}$ in (19) returns the parent of b.

$$\mathtt{prb}(b, i) = (l((\mathtt{B}(i-1))_1))_{\mathtt{ipb}(b,i)} \tag{19}$$

The p.r. function $\mathtt{prbs}$ in (20) computes the Gödel number whose last element is b and whose previous elements are its predecessors. In other words, element 8 is the parent board of element 9, element 7 is the parent board of element 8, etc.

$$\begin{aligned} prbs(b,i) = [&prb(\ldots(prb(b,i),7),\ldots,1),\\ &\ldots,\\ &prb(prb(prb(b,i),7),6),\\ &prb(prb(b,i),7), prb(b,i), b] \end{aligned} \tag{20}$$

Let $X = \mathtt{prbs}(b, i)$ such that $i > 7$ and $\mathtt{valid}(b) = 1$. The p.r. predicate $\mathtt{drw3r}(b, i) \equiv \gamma_{17}(X)$, where $\gamma_{17}(X) \equiv \{(X)_1 = (X)_5 = (X)_9\} \wedge \{(X)_2 = (X)_6\} \wedge \{(X)_3 = (X)_7\} \wedge \{(X)_4 = (X)_8\}$ is true if b is a threefold repetition board. To put it differently, in the list of b's predecessors, elements 1 and 5 must be the same as b (i.e., element 9), element 2 must be the same as element 6, element 3 is the same as element 7, and element 4 is the same as element 8.

It is straightforward to extend the definition of $\mathtt{prbs}(b, i)$ to a p.r. predicate $\mathtt{drw50}(b, i)$ that computes 49 predecessors of a valid board b in epoch $\mathtt{B}(i)$ and checks if each board in the Gödel number of the predecessors and b itself contains no capture, which can be done by comparing the number of pieces on a given board and its immediate predecessor (i.e., its parent), or a pawn move, which can be done by comparing the pawn positions of all the predecessor boards of b and b itself. All these functions are p.r., because they manipulate Gödel numbers.

We can similarly express, in a p.r. fashion, each case of the dead position rule. For example, checking if a given board b contains only two kings or whether the white has the king and a knight and the black has only the king is p.r., because it requires checking p.r. properties of a given Gödel number. Consequently, we may assume that there is a p.r. predicate $\texttt{drw}(b, i)$ that returns 1 if a valid board b in epoch $i > 0$ is a draw and 0 otherwise.

Let $t \in \mathbb{N}$. We define the chess game history in (21) as the Gödel number encoding the boards at each epoch and the player whose turn it is to play at the next epoch.

$$\texttt{H}(t) = \otimes_r|_{i=0}^{t}\texttt{B}(i) \tag{21}$$

For example $\texttt{H}(3) = [\texttt{B}(0), \texttt{B}(1), \texttt{B}(2), \texttt{B}(3)] = [\langle G_0, 1\rangle, \langle G_1, 2\rangle, \langle G_2, 1\rangle, \langle G_3, 2\rangle]$. Let $t, i, j \in \mathbb{N}$. Let $G_i^t = l((\texttt{H}(t))_i)$, $b_{i,j}^t = (G_i^t)_j$, $L_t = \texttt{Lt}(\texttt{H}(t))$, and $L_i^t = \texttt{Lt}(G_i^t)$. The p.r. predicate $\texttt{ww}(t)$, $t \geq 0$, in (22) returns true if there is a board in epoch i, $0 \leq i \leq t$, where the white checkmates its opponent.

$$\texttt{ww}(t) \equiv (\exists i)_{\leq L_t}\{(\exists j)_{\leq L_i^t}\{\texttt{bmtd}(b_{i,j}^t)\}\} \tag{22}$$

The p.r. predicate $\texttt{bw}(t)$ in (23) is true if there is a board in epoch i, $0 \leq i \leq t$, where the black checkmates its opponent.

$$\texttt{bw}(t) \equiv (\exists i)_{\leq L_t}\{(\exists j)_{\leq L_i^t}\{\texttt{wmtd}(b_{i,j}^t)\}\} \tag{23}$$

If $\texttt{H}(t)$ is the history of the game, then the white can win only in the even-numbered epochs and the black can win only in the odd-numbered epochs. We can define two predicates $\texttt{W}_w(m)$ and $\texttt{W}_b(m)$ that are true if the white or black, respectively, wins within t moves. Specifically, $\texttt{W}_w(m) \equiv (\exists t)_{\leq m}\{\texttt{ww}(t)\}$ and $\texttt{W}_b(m) \equiv (\exists t)_{\leq m}\{\texttt{bw}(t)\}$. The p.r. predicate in (24) combines both predicates into one.

$$\texttt{W}(p, m) = \begin{cases} \texttt{W}_w(m) & \text{if } p = 1 \\ \texttt{W}_b(m) & \text{if } p = 2 \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

In a similar fashion, we can define the p.r. predicate in (25) that is true if a draw is achieved for player p within m moves.

$$\texttt{D}(p, m) \equiv (\exists i)_{\leq L_t}\{(\exists j)_{\leq L_i^t}\{\texttt{dwr}(b_{i,j}^t, i)\}. \tag{25}$$

Let $p \in \{1, 2\}$, $t \in \mathbb{N}$. The p.r. functions $W_{\leq}(p, t)$ in (26) and $D_{\leq}(p, t)$ in (27) return the lists of all win and draw boards for p, respectively, within t moves.

$$\texttt{W}_{\leq}(p, t) = \otimes_r|_{i=1}^{t}\texttt{W}(p, i) \tag{26}$$

$$\texttt{D}_{\leq}(p, t) = \otimes_r|_{i=1}^{t}\texttt{D}(p, i) \tag{27}$$

Let $b_x \in l((\mathtt{B}(i))_1)$ and $b_y \in l((\mathtt{B}(j))_1)$, where $i < j$. Let

$$\mathtt{ptx}(b_x, b_y, 0) = \gamma_{18}(b_x, b_y),$$
$$\mathtt{ptx}(b_x, b_y, t+1) = [b_y] \otimes_l \mathtt{ptx}(b_x, \mathtt{prb}(b_y, s(t)), t),$$

where

$$\gamma_{18}(x, y) = \begin{cases} [x] & \text{if } x = y, \\ 0 & \text{otherwise.} \end{cases}$$

The p.r. function path in (28) gives the list of boards, possibly empty, from the board $b_x \in \mathtt{B}(i)$ to the board $b_y \in \mathtt{B}(j)$.

$$\mathtt{path}(b_x, b_y, i, j) = \mathtt{ptx}(b_x, b_y, j - i) \tag{28}$$

If $\mathtt{path}(b_x, b_y, i, j) \neq 0$, $b_y \in \mathtt{B}(j)$ is *reachable* from $b_x \in \mathtt{B}(i)$. If $\mathtt{path}(b_x, b_y, 1, 3) = 0$, then b_y is *unreachable* from b_x. Let $b \in l((\mathtt{B}(i))_1)$, $k \geq 0$, $i < j$, and let $Z \in \mathbb{N}$ be a list of boards. Let

$$\mathtt{ppx}(b, Z, i, j, 0) = [\,],$$
$$\mathtt{ppx}(b, Z, i, j, t+1) = \gamma_{19}(b, Z, t, i, j) \otimes_l \mathtt{ppx}(b, Z, i, j, t),$$

where $\gamma_{19}(b, Z, t, i, j) = [\mathtt{path}(b, (Z)_{s(t)}, i, j)]$. The p.r. function ppx returns a list of paths from a given board b to each board in Z. Let $\mathtt{ppxx}(b, Z, i, j) = \mathtt{ppx}(b, Z, i, j, Lt(Z))$. The p.r. function in (29) returns the list of all paths, possibly empty, from $b \in l((\mathtt{B}(i))_1)$ to a win board $b' \in l((\mathtt{B}(j))_1)$.

$$\mathtt{wpss}(b, i, j) = \otimes_r|_{k=i+1}^{j} \gamma_{20}(b, k, i, j), \tag{29}$$

where $\gamma_{22}(b, k, i, j) = \mathtt{ppxx}(b, \mathtt{W}(p, k), i, j)$, where $p = 1$ if $2|i$ and $p = 2$, otherwise. The p.r. function in (30) removes all empty paths from the list returned by wpss.

$$\mathtt{wps}(b, i, j) = \mathtt{rm}(0, \mathtt{wpss}(b, i, j)) \tag{30}$$

The p.r. function in (31) returns the list of all paths, possibly empty, that start at $b \in l((\mathtt{B}(i))_1)$ and end with a draw board $b' \in l((\mathtt{B}(j))_1)$.

$$\mathtt{dpss}(b, i, j) = \otimes_r|_{k=i+1}^{j} \gamma_{21}(b, k, i, j), \tag{31}$$

where $\gamma_{21}(b, k, i, j) = \mathtt{ppxx}(b, \mathtt{D}(p, k), i, j)$, where $p = 1$ if $2|i$ and $p = 2$, otherwise. The p.r. function in (32) removes all empty paths from the list returned by dpss.

$$\mathtt{dps}(b, i, j) = \mathtt{rm}(0, \mathtt{dpss}(b, i, j)) \tag{32}$$

Let $0 \leq k < t$, $k < j \leq t$, $p \in \{1, 2\}$, and $b \in l((\mathtt{B}(k))_1)$. If $p = 1$ (i.e., p plays white), then p is the max player whose objective is to maximize the utility score of b. Let the highest utility score that can be assigned to b be 3 if there is at least one win board $b' \in \mathtt{B}(j)$ reachable from b. If there are no reachable win boards, let the utility of b be 2 so long as there is at least one draw board $b' \in \mathtt{B}(j)$ reachable from b. Let b have the lowest utility score of 1 when there is no win or draw board $b' \in \mathtt{B}(j)$ reachable from b. Let the utility score of 0 be assigned to invalid boards. The p.r. function in (33) returns the utility score of b for $p = 1$, where $\gamma_{22}(b, k, t) \equiv Lt(\mathtt{dps}(b, k, t)) = Lt(\mathtt{wps}(b, k, t)) = 0$.

$$\mathtt{U}_{max}(b, k, t) = \begin{cases} 3 & \text{if } Lt(\mathtt{wps}(b, k, t)) > 0, \\ 2 & \text{if } Lt(\mathtt{dps}(b, k, t)) > 0, \\ 1 & \text{if } \gamma_{22}(b, k, t), \\ 0 & \text{if } \neg\mathtt{valid}(b). \end{cases} \tag{33}$$

If $p = 2$ (i.e., p plays black), then p is the min player whose objective is to minimize the utility score of b. Let the utility score of b be 1 if there is at least one win board $b' \in \mathtt{B}(j)$ reachable from b. If there are no win boards in $\mathtt{B}(j)$ reachable from b, let the utility score of b be 2 so long as there is at least one draw board in $b' \in \mathtt{B}(j)$ reachable from b. Let the highest utility score of 3 indicate that there is no win or draw board $b' \in \mathtt{B}(j)$ reachable from b. Again, let the utility score of 0 be assigned to invalid boards. The p.r. function in (34) returns the utility score for b for player 0.

$$\mathtt{U}_{min}(b, k, t) = \begin{cases} 1 & \text{if } Lt(\mathtt{wps}(b, k, t)) > 0, \\ 2 & \text{if } Lt(\mathtt{dps}(b, k, t)) > 0, \\ 3 & \text{if } \gamma_{22}(b, k, t), \\ 0 & \text{if } \neg\mathtt{valid}(b). \end{cases} \tag{34}$$

Let $b \in l((\mathtt{B}(i))_1)$, $0 \leq i < j$, The p.r. function $\mathtt{U}$ in (35) returns the utility score of $b \in \mathtt{B}(i)$ for player $p \in \{1, 2\}$ when the game continues from epoch $\mathtt{B}(i)$ to epoch $\mathtt{B}(j)$.

$$\mathtt{U}(b, p, i, j) = \begin{cases} \mathtt{U}_{max}(b, i, j) & \text{if } 2|i, \\ \mathtt{U}_{min}(b, i, j) & \text{if } \neg(2|i), \\ 0 & \text{otherwise}. \end{cases} \tag{35}$$

Let $0 \le i < j$. The p.r. function $\mathtt{fb}$ in (36) returns a list of boards with a given utility score $x \in \{1, 2, 3\}$ for player p from epoch i to epoch j.

$$\mathtt{fb}(p, i, j, x) = \mathtt{fbx}(p, i, j, Z, L, x), \tag{36}$$

where $Z = l((\mathtt{B}(i))_1)$, $L = Lt(l((\mathtt{B}(j))_1))$, $\mathtt{fbx}(p, i, j, Z, 0, x) = [\,]$, $\mathtt{fbx}(p, i, j, Z, t+1, x) = \gamma_{23}(p, i, j, (Z)_{s(t)}, \mathtt{fbx}(p, i, j, Z, t, x), x)$, and

$$\gamma_{23}(p, i, j, b, Z, x) = \begin{cases} [b] \otimes_l Z & \text{if } \mathtt{U}(b, p, i, j) = x, \\ b & \text{if } \mathtt{U}(b, p, i, j) \neq x. \end{cases}$$

Let $0 \le i < j$ and $\alpha_3(p, i, j) \equiv \mathtt{fb}(p, i, j, 3) \neq 0$, $\alpha_2(p, i, j) \equiv \mathtt{fb}(p, i, j, 2) \neq 0 \wedge \mathtt{fb}(p, i, j, 3) = 0$, and $\alpha_1(p, i, j) \equiv \mathtt{fb}(p, i, j, 1) \neq 0 \wedge \mathtt{fb}(p, i, j, 3) = \mathtt{fb}(p, i, j, 2) = 0$, The p.r. function $\mathtt{fb}_{max}$ returns a list of boards for player $p = 1$ from epoch i to epoch j.

$$\mathtt{fb}_{max}(p, i, j) = \begin{cases} \mathtt{fb}(p, i, j, 3) & \text{if } \alpha_3(p, i, j), \\ \mathtt{fb}(p, i, j, 2) & \text{if } \alpha_2(p, i, j), \\ \mathtt{fb}(p, i, j, 1) & \text{if } \alpha_1(p, i, j), \\ 0 & \text{otherwise.} \end{cases}$$

Let $\beta_1(p, i, j) \equiv \mathtt{fb}(p, i, j, 1) \neq 0$, $\beta_2(p, i, j) \equiv \mathtt{fb}(p, i, j, 2) \neq 0 \wedge \mathtt{fb}(p, i, j, 1) = 0$, and $\beta_3(p, i, j) \equiv \mathtt{fb}(p, i, j, 3) \neq 0 \wedge \mathtt{fb}(p, i, j, 1) = \mathtt{fb}(p, i, j, 2) = 0$. The p.r. function $\mathtt{fb}_{min}$ returns a list of boards for player $p = 2$ from epoch i to epoch j.

$$\mathtt{fb}_{min}(p, i, j) = \begin{cases} \mathtt{fb}(p, i, j, 1) & \text{if } \beta_1(p, i, j), \\ \mathtt{fb}(p, i, j, 2) & \text{if } \beta_2(p, i, j), \\ \mathtt{fb}(p, i, j, 3) & \text{if } \beta_3(p, i, j), \\ 0 & \text{otherwise.} \end{cases}$$

The p.r. function $\mathtt{fb}_{mnx}$ in (37) returns the list of optimal boards for player $p \in \{1, 2\}$ from epoch i to epoch j.

$$\mathtt{fb}_{mnx}(p, i, j) = \begin{cases} \mathtt{fb}_{max}(p, i, j) & \text{if } p = 1, \\ \mathtt{fb}_{min}(p, i, j) & \text{if } p = 2, \\ 0 & \text{otherwise.} \end{cases} \tag{37}$$

The p.r. function $\mathtt{et}(k)$ in (38) determines whose turn it is to play at epoch $\mathtt{B}(k)$.

$$\texttt{et}(k) = \begin{cases} 1 & \text{if } 2|k, \\ 2 & \text{if } \neg(2|k). \end{cases} \tag{38}$$

Let $0 \leq k < t$. Let

$$\omega_1(k, t) = \otimes_r|_{j=k}^{t-1}[(\texttt{fb}_{mnx}(\texttt{et}(j), j, t))_1].$$

The p.r. ω_1 function chooses the first board returned from the function $\texttt{fb}_{mnx}$. Other p.r. functions can also be defined to inspect the first n boards returned by $\texttt{fb}_{mnx}$. The p.r. function $\texttt{bseq}$ in (39) returns a sequence of optimal boards for player p whose turn it is to play at k.

$$\texttt{bseq}(p, k, t) = \begin{cases} \omega(k, t) & \text{if } \texttt{et}(k) = p, \\ [\,] & \text{otherwise.} \end{cases} \tag{39}$$

Let $0 \leq k < t$ and let b_1 and b_2 be two boards such that $\phi(b_1, b_2, k) \equiv \texttt{in}(b_1, l((\texttt{B}(k))_1)) \wedge \texttt{in}(b_2, l((\texttt{B}(k+1))_1))$ and $\texttt{prb}(b_2, k+1) = b_1$. Let

$$\omega_2(b_1, b_2) = \min_{i \leq 64}\{(b_1)_i = 1 \wedge (b_2)_i \neq 1\}.$$

The p.r. function $\omega_3(b_1, b_2)$ extracts a move $\langle p, j\rangle$, where $1 \leq j \leq 64$, that changes b_1 to b_2.

$$\omega_3(b_1, b_2, k) = \begin{cases} \langle \texttt{et}(k), \omega_2(b_1, b_2)\rangle & \text{if } \phi(b_1, b_2, k), \\ 0 & \text{otherwise.} \end{cases} \tag{40}$$

Let $Z = \texttt{bseq}(p, k, t) = [b_1, \ldots, b_{t-k+1}]$ and let

$$\texttt{mseq}(p, k, t) = \otimes_r|_{i=1}^{Lt(Z)-1}[\omega_3((Z)_i, (Z)_{i+1}, k - 1 + i)]$$

Let $\gamma_{24}(p, i) \equiv \{2|i \wedge p = 1\} \vee \{\neg(2|i) \wedge p = 2\}$. The p.r. function in (41) defines a game of chess for player p and epoch i by returning a sequence of optimal moves for p beginning at epoch $\texttt{B}(i)$ and ending at epoch $\texttt{B}(j)$.

$$G(p, i, j) = \begin{cases} \texttt{mseq}(p, i, j) & \text{if } \gamma_{24}(p, i), \\ 0 & \text{otherwise.} \end{cases} \tag{41}$$

Since $G(i, j)$ is p.r., we have the following theorem.

Theorem *Chess is a deterministic two-player p.r. game.*

4 Conclusion

A proof is presented to show that chess is a deterministic two-player primitive recursive game. To the extent that the proof holds, chess can be characterized in terms of primitive recursive functions. If this is the case, some deterministic two-player games and processes that can be formalized as such are likely to have algorithmic solutions that outperform human players. The techniques developed in this paper may lead to proofs that other deterministic two-player board games are primitive recursive and contribute to the theory of primitive recursive functions [5].

References

1. T. Bolon, *How to Never Lose at Tic Tac Toe* (Book Country, New York, NY, USA, 2013)
2. W. Daly, Jr., Computer Strategies for the Game of Qubic, M. Eng. thesis, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, Feb. 1961
3. V. Kulyukin, On primitive recursiveness of Tic Tac Toe, in *Proceedings of the International Conference on Foundations of Computer Science (FCS'19)*, pp. 9–15, Las Vegas, NV, USA, Jul. 29–Aug. 01, 2019
4. M. Davis, R. Sigal, E. Weyuker, *Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science*, 2nd edn. (Harcourt, Brace & Company, Boston, MA, USA, 1994)
5. H. Rogers, Jr., *Theory of Recursive Functions and Effective Computability* (The MIT Press, Cambridge, MA, USA, 1988)