McNaughton games and extracting strategies for concurrent programs

Anil Nerode\textsuperscript{a,*,1}, Jeffrey B. Remmel\textsuperscript{b,2}, Alexander Yakhnis\textsuperscript{c,3}

\textsuperscript{a} Mathematical Sciences Institute, Cornell University, Ithaca, NY 14853, USA
\textsuperscript{b} Department of Mathematics and Computer Science, University of California at San Diego, San Diego, CA 92093, USA
\textsuperscript{c} Mathematical Sciences Institute, Cornell University, Ithaca, NY 14853, USA

Received 1 June 1995; revised 30 November 1995

1. Introduction

Nerode et al. [13] showed that a correct concurrent program can be viewed as a winning strategy in a suitably defined two player game played between the Programmer and the Computer in which the program specification is defined by the rules of the game together with the winning condition. This gives rise to the question as to whether there are useful algorithms to extract (provably) winning strategies in these games, which then yield (provably correct) concurrent programs.

Now these games can be described in Rabin's S2S, the monadic second-order theory of two successors. Decision procedures for the latter show that such algorithms exist. But past available decision methods were too cumbersome to use, even in simple cases. Successively simpler game-based decision procedures for S2S were provided by [5,19,20]. In 1993, based on these papers, McNaughton [8] introduced a class of two player infinite games which are played on a finite graph and have an especially lucid decision procedure for extraction of winning strategies.

The games considered in [13] can be viewed as a slight variant of Büchi–Landweber games [2]. We give clean algorithms for the equivalence of McNaughton games and Büchi–Landweber games. This allows the McNaughton algorithm to be used to extract (provably) winning strategies, and therefore (verifiably correct) concurrent programs via the Nerode–Yakbnis–Yakhnis paradigm.

\* Corresponding author. E-mail: anil@math.cornell.edu.
1 Supported in part by Army Research Office contract DAAL03-91-C-0027.
2 Supported in part by Army Research Office contract DAAL03-91-C-0027 and NSF grant DMS-9306427.
3 Supported by DARPA-US ARMY AMCCOM (Picatinny Arsenal, N. J.) contract DAAA21-92-C-0013 to ORA Corp.
But this is not the only potential use for clean, understandable, decision methods for S2S. Many problems in computer science have been shown to be decidable by reduction to S2S (see [14] and the references there), and decision methods in Discrete Event Systems and in Hybrid Systems use them as well. To locate the latter literature, see any recent Proceedings of the CDC.

The outline of this paper is as follows. In Section 2, we shall formally define Büchi–Landweber and McNaughton games and show that these two games are equivalent. In Section 3, we shall show how extraction of winning strategies in McNaughton games can be used to extract concurrent programs by showing how McNaughton games can be used to solve the so-called Producer–Consumer Problem. Then in Section 4, we shall give a detailed analysis of the complexity of extracting winning LVR strategies for McNaughton games. Finally in Section 5, we shall show that in a number of special cases the extraction of winning strategies for McNaughton games can be extremely efficient.

Briefly, a McNaughton game $G_M$ is a game played by two players Red and Black on a directed bipartite finite graph $G = (V, E)$. Here $E$ is the set of directed edges of $G$ and $V$ is the set of vertices of $G$. We assume that the set of vertices $V$ of $G$ is partitioned into two sets, $V_R$, the set of red nodes, and $V_B$, the set of black nodes, and all directed edges of $G$ either connect a red node to black node or a black node to a red node. Moreover, we assume that there is at least one directed edge out of each node. A play $p$ of the game consists of alternating moves of Red and Black where Red, given a red node $v_r \in V_R$, lists a black node $v_b$ which can be reached by following an edge out of $v_r$ and where Black, given a black node $v_b \in V_B$, lists a red node $v_r'$ which can be reached by following an edge out of $v_b$. A play $p$ starts with some node $v \in V$. Thus for example if $v$ is a black node, then Black will start the play and the play $p$ will produce a sequence of nodes $p = (q_b, q_r, q_b, q_r, \ldots)$ where $q_r$ are red nodes for all $i$, $q_b$ are black nodes for all $i$, and $(q_r, q_b)$ are edges of $E$ for all $i$.

There are two other components to the McNaughton game $G_M$. First we specify a set $S \subseteq V$ of significant nodes. We let $2^S$ denote the set of all subsets of $S$. This given, we define the perm set of a play of a play $p = (\eta_0, \eta_1, \eta_2, \ldots)$ by

$$\text{perm}(p) = \{ \eta \in S : \eta = \eta_i \text{ for infinitely many } i \}. \quad (1)$$

The final component of our McNaughton game $G_M$ is a set $\Omega \subseteq 2^S$ which we call the set of winning sets. We then say that for any play $p$, Black wins $p$ if $\text{perm}(p) \in \Omega$. (2)

Thus formally a McNaughton game $G_M$ is a triple $(G, S, \Omega)$, where $G = (V, E)$ is a directed bipartite finite graph as described above, $S \subseteq V$, and $\Omega \subseteq 2^S$.

Next we define strategies and winning strategies for McNaughton games. That is, assume Black always plays first. A strategy for Black is a function $f : (V_R \times V_R)^{<\omega} \times V_B \rightarrow V_R$ such that for all initial segments of a play of $(\eta_0, \eta_1, \ldots, \eta_{i+1}, \eta_{i+1})$ of $G_M$, $f((\eta_0, \eta_1, \ldots, \eta_i, \eta_{i+1}, \eta_{i+1})) = \eta_{i+1}$, where $\eta_{i+1} \in V_R$ and $(\eta_{i+1}, \eta_{i+1}) \in E$. Here for any set $C$, $C^{<\omega}$, denotes the set of all finite sequences from $C$. We say that $B$
follows the strategy $f$ during a play $p = (\eta_0, \eta_1, \eta_2, \ldots)$, if for all $k$, $\eta_{k+1} = f((\eta_0, \eta_1, \ldots, \eta_k))$. Then we say $f$ is a winning strategy for Black if Black wins all plays $p$ such that Black follows $f$ during the play $p$, no matter how Red plays. Strategies and winning strategies for Red are defined in a similar manner.

Note that when Black is following a strategy $f$ during a play $p$, Black’s next move at any given point in the play can, in principle, depend on the entire history of the play up to that point. McNaughton showed that there is a class of strategies called LVR strategies which depend only on a certain limited amount of information about the order in which the significant nodes were visited during the play which suffice to solve such games. That is, given a McNaughton game $G = (G, S, \Omega)$ with an underlying graph $G = (V, E)$, define $\text{Win}_B(G_M)$ and $\text{Win}_R(G_M)$ to be the set of all nodes $\eta$ in $V$ such that there is a LVR strategy $f_\eta$ for Black (Red) such that Black (Red) wins all plays $p$ such that $p$ starts at $\eta$ and Black (Red) follows $f_\eta$. For any set $A \subseteq V$, we say that $f$ is a winning LVR strategy from a set $A \subseteq V$ for a player $P$ if $f$ is a LVR strategy and in any play $p = (\eta_0, \eta_1, \ldots)$ where $P$ follows $f$ and $\eta_0 \in A$, $P$ wins. In [8], McNaughton proved the following.

**Theorem 1.1.** Let $G = (G, S, \Omega)$ where $G = (V, E)$ be a McNaughton game. Then $\text{Win}_B(G_M)$ and $\text{Win}_R(G_M)$ partition $V$. Moreover, there is an algorithm which runs in $O(c^{|S|}S! |V|)$ steps to find $\text{Win}_B(G_M)$ and $\text{Win}_R(G_M)$ for some constant $c$.

Moreover, implicit in McNaughton’s proof of Theorem 1.1 is a procedure to actually extract the winning LVR strategy for Black from $\text{Win}_B(G_M)$ and the winning LVR strategy for Red from $\text{Win}_R(G_M)$.

As mentioned above, our main interest in McNaughton’s games is in the algorithm to extract winning strategies since winning strategies can be viewed as a correct concurrent programs via the Nerode–Yakhnis–Yakhnis paradigm. The extraction of winning strategies also has applications to extracting control automaton for hybrid systems. McNaughton did not formally analyze the complexity of constructing the required LVR winning strategies for Black and Red. The main result of this paper is a careful analysis of the complexity of extracting such strategies. We show that with appropriate data structures, one can construct the sets $\text{Win}_B(G_M)$ and $\text{Win}_R(G_M)$ and the required LVR winning strategies in $O(2^{|S|}|S||E||S|)$ steps. Moreover, we shall show that in many special cases such as when the class of winning sets is an interval, i.e. when $\Omega = \{Z : Z_1 \subseteq Z \subseteq Z_2\}$ for some sets $Z_1 \subseteq Z_2 \subseteq S$, one can construct the sets $\text{Win}_B(G_M)$ and $\text{Win}_R(G_M)$ and the required LVR winning strategies in $O(|S||E|)$ steps. Thus in many special cases we can extract winning strategies in a very efficient manner.

2. Games: definitions and equivalences

In this section we shall define two classes of infinite games. We first introduce Büchi–Landweber games [2]. Büchi–Landweber games were introduced to solve certain
problems which arose in the study of the monadic second-order theory of arithmetic. Then we shall introduce McNaughton games which first appeared in McNaughton’s 1993 paper [8]. Our interest in McNaughton games is due to the fact that there is an efficient algorithm to extract winning strategies for such games. After giving the basic definitions of Büchi–Landweber and McNaughton games, we shall end this section by proving that these games are essentially equivalent.

### Büchi–Landweber games

A Büchi–Landweber game is an infinite game played between two players, Player I and Player II. There are 3 components of a Büchi–Landweber game $G_{BL}$. First a play of the game consists of an infinite alternation of moves by Player I who lists an element from an alphabet $I$ and Player II who lists an element from an alphabet $J$. Thus a play $p$ of the game results in an infinite sequence $p = (i_1, j_1, i_2, j_2, \ldots)$ where $i_k \in I$ and $j_k \in J$ for all $k \geq 0$. An initial segment of a play $p$ of the form $(i_1, j_1, \ldots, i_n, j_n)$ or $(i_1, j_1, \ldots, i_n)$ will be called a partial play. The second component of a Büchi–Landweber game is a finite state automation $A = (Q, U, q_0, \delta)$ where $Q = I \times J$ is the input alphabet, $U$ is the set of states, $q_0$ is the initial state, and $\delta : \Sigma \times U \to U$ is the transition function. We use the finite automaton $A$ to associate to each play $p$ of our game an infinite sequence of states

$$(u_1(p), u_2(p), \ldots)$$

where $u_1(p) = \psi((i_1, j_1), u_0), u_2(p) = \psi((i_2, j_2), u_1(p)), u_3(p) = \psi((i_3, j_3), u_2(p)), \text{ etc.}$

We then define the permset of a play $p$, $\text{perm}(p)$, by

$$\text{perm}(p) = \{ u \in U : u = u_i(p) \text{ for infinitely many } i \}. \quad (3)$$

The third and final component of a Büchi–Landweber game is a set $\Omega \subseteq 2^U$ which we call the set of winning conditions. Here $2^U$ denotes the set of all subsets of $U$. $\Omega$ is used to define which player wins a play $p$ of the game by declaring that

Player I wins $p \iff \text{perm}(p) \in \Omega. \quad (4)$

Thus formally a Büchi–Landweber game $G_{BL}$ is a 4-tuple $(I, J, A, \Omega)$, where $I$ and $J$ are finite alphabets, $A = (I \times J, U, q_0, \delta)$ is a finite automaton, and $\Omega \subseteq 2^U$.

A strategy $\delta$ for Player I in a Büchi–Landweber game $G_{BL}$ is a map from $(I \times J)^{<\omega}$ into $I$. The idea is that a strategy $\delta$ gives the next move of Player I for any given partial play of even length. That is, Player I, given a partial play $(i_1, j_1, \ldots, i_k, j_k)$ where $r \in I$ and $j_r \in J$ for $r = 1, \ldots, k$, would next list $\delta((i_1, j_1), \ldots, (i_k, j_k))$ if he was following the strategy $\delta$. Similarly a strategy for Player II is map $\gamma : (I \times J)^{<\omega} \times I \to J$. We say $\delta$ is winning strategy for Player I if whenever Player I follows the strategy $\delta$, Player I wins no matter how Player II plays. A winning strategy $\gamma$ for Player II is defined similarly.

### McNaughton games

A McNaughton game $G_M$ is a game played by two players Red and Black on a directed bipartite finite graph $G = (V, V_B, V_R, E)$. Here $E$ is the set of directed edges
of $G$ and $V$ is the set of vertices of $G$. We assume that the set of vertices $V$ of $G$ is partitioned into two sets, $V_R$, the set of red nodes, and $V_B$, the set of black nodes, and all directed edges of $G$ are all of the form $(\eta_b, \eta_r)$ or $(\eta_r, \eta_b)$ where $\eta_r \in V_R$ and $\eta_b \in V_B$. Moreover, we assume that for each red node $\eta_r \in V_R$ there is at least one black node $\eta_b \in V_B$ such that $(\eta_r, \eta_b) \in E$ and vice versa for each black node $\eta_b \in V_B$, there is at least one red node $\eta_r \in V_R$ such that $(\eta_b, \eta_r) \in E$. A play $p$ of the game consists of alternating moves of Red and Black where Red, given a red node $\eta_r \in V_R$, lists a black node $\eta_b$ which can be reached by following an edge out of $\eta_r$ and where Black, given a black node $\eta_b \in V_B$, lists a red node $\eta_r$ which can be reached by following an edge out of $\eta_b$. A play $p$ starts with some node $\eta \in V$. If $\eta$ is a red node, then Red will start the play and the play $p$ will produce a sequence of nodes $p = (\eta = \eta_r, \eta_b, \eta_r, \eta_b, \ldots)$ where $\eta_r$ are red nodes for all $i$, $\eta_b$ are black nodes for all $i$, and $(\eta_r, \eta_b)$ and $(\eta_b, \eta_r)$ are edges of $E$ for all $i$. If $\eta$ is black node, then Black will start the play and the play $p$ will produce a sequence of nodes $p = (\eta = \eta_b, \eta_r, \eta_b, \eta_r, \ldots)$ where $\eta_b$ are black nodes for all $i$, $\eta_r$ are red nodes for all $i$, and $(\eta_b, \eta_r)$ and $(\eta_r, \eta_b)$ are edges of $E$ for all $i$. As with Büchi–Landweber games a partial play of $\mathcal{G}_M$ is defined to be an initial segment of some play of $\mathcal{G}_M$.

There are two other components to the McNaughton game $\mathcal{G}_M$. First we specify a set $S \subseteq V$ of significant nodes. This given, we define the perm set of play a play $p = (\eta_0, \eta_1, \eta_2, \ldots)$ by

$$\text{perm}(p) = \{ \eta \in S : \eta = \eta_i, \text{ for infinitely many } i \}.$$ (5)

The final component of our McNaughton game $\mathcal{G}_M$ is a set $\Omega \subseteq 2^S$ which we call the set of winning sets. We then say that for any play $p$,

$$\text{Black wins } p \iff \text{perm}(p) \in \Omega.$$ (6)

Thus formally a McNaughton game $\mathcal{G}_M$ is a triple $(G, S, \Omega)$, where $G = (V, V_B, V_R, E)$ is a directed bipartite finite graph as described above, $S \subseteq V$, and $\Omega \subseteq 2^S$.

Strategies for McNaughton games are defined in much the same manner as we defined strategies for Büchi–Landweber games. That is, assume Black always plays first. A strategy for Black is a function $f : (V_B \times V_R)^{<\omega} \times V_B \rightarrow V_R$ such that for all partial plays $(\eta_{b_1}, \eta_{r_1}, \ldots, \eta_{b_i}, \eta_{r_i}, \eta_{b_{i+1}})$ of $\mathcal{G}_M$, $f((\eta_{b_1}, \eta_{r_1}, \ldots, \eta_{b_i}, \eta_{r_i}, \eta_{b_{i+1}})) = \eta_{r_{i+1}}$, where $\eta_{r_{i+1}} \in V_R$ and $(\eta_{b_{i+1}}, \eta_{r_{i+1}}) \in E$. We say that $B$ follows the strategy $f$ during a play $p = (\eta_{b_1}, \eta_{r_1}, \eta_{b_2}, \eta_{r_2}, \ldots)$ if for all $k$, $\eta_{r_{k+1}} = f((\eta_{b_k}, \eta_{r_k}, \ldots, \eta_{b_{k+1}}))$. Then we say $f$ is a winning strategy for Black if Black wins all plays $p$ such that Black follows $f$ during the play $p$, no matter how Red plays.

It will be useful for our results in Sections 4 and 5 to define two subclasses of strategies for Black. First we say that $f$ is a no memory strategy for Black if the value of $f$ depends only on the last node in the sequence. That is, $f$ is a no memory strategy for Black if for all partial plays $p = (\eta_{b_1}, \eta_{r_1}, \ldots, \eta_{b_i}, \eta_{r_i}, \eta_{b_{i+1}})$ and $p' = (\eta_{b_1}', \eta_{r_1}', \ldots, \eta_{b_i}', \eta_{r_i}', \eta_{b_{i+1}}')$,

$$\eta_{b_{i+1}} = \eta_{b_{i+1}}' \implies f(p) = f(p').$$
Next we want to define what McNaughton terms \( LVR \) strategies, which are analogous to the latest appearance record of Gurevich and Harrington [5]. First we must define the last visitation record of any partial play \( (\eta_1, \ldots, \eta_t) \) of \( \mathcal{G}_M \). \( LVR(\eta_1, \ldots, \eta_t) \) will always be a sequence of distinct nodes from \( S \), the set of significant nodes. \( LVR(\eta_1, \ldots, \eta_t) \) is defined inductively as follows.

(a) \( LVR(\emptyset) = \emptyset \).

(b) \( LVR(\eta_1, \ldots, \eta_t) = LVR(\eta_1, \ldots, \eta_{t-1}) \) if \( \eta_t \notin S \).

(c) If \( \eta_t \in S \), there are two subcases. Namely

(i) \( LVR(\eta_1, \ldots, \eta_t) = LVR(\eta_1, \ldots, \eta_{t-1}) \sim \eta_t \)

if \( \eta_t \) does not appear in \( LVR(\eta_1, \ldots, \eta_{t-1}) \) and

(ii) \( LVR(\eta_1, \ldots, \eta_t) = (s_1, \ldots, s_k, s_{i_t+1}, \ldots, s_{i_t}, \eta_t) \)

if \( LVR(\eta_1, \ldots, \eta_{t-1}) = (s_1, \ldots, s_k, \eta_t, s_{i_t+1}, \ldots, s_{i_t}) \).

Here given a sequence \( s = (s_0, \ldots, s_m) \) and an element \( s_{m+1} \), we let \( s \sim s_{m+1} \) denote the sequence which results from concatenating \( s_{m+1} \) onto the end of \( s \), i.e. \( s \sim s_{m+1} = (s_0, \ldots, s_m, s_{m+1}) \).

Thus for example, if \( LVR(\eta_1, \ldots, \eta_t) = (s_1, \ldots, s_{i_t-2}, s_{i_{t-1}}, s_{i_t}) \), then \( s_{i_t} \) was the last node of \( S \) visited in the partial play \( (\eta_1, \ldots, \eta_t) \). \( s_{i_{t-1}} \) represents the last node other than \( s_{i_t} \) visited during the partial play. That is, if \( \eta_r \) is the last occurrence of \( s_{i_{t-1}} \) in \( (\eta_1, \ldots, \eta_t) \), then the only node of \( S \) which occurs in \( (\eta_1, \ldots, \eta_t) \) is \( s_{i_t} \). Similarly if \( \eta_p \) is the last occurrence of \( s_{i_{t-1}} \) in \( (\eta_1, \ldots, \eta_t) \), then \( p < r \) and the only elements of \( S \) which occur in \( (\eta_{r+1}, \ldots, \eta_t) \) are \( s_{i_{t-1}} \) and \( s_{i_t} \), etc.

We say that a strategy \( f \) for Black is an \( LVR \) strategy if the value of \( f \) at \( (\eta_1, \ldots, \eta_t) \) depends only on \( LVR(\eta_1, \ldots, \eta_t) \). That is, if \( p = (\eta_1, \ldots, \eta_t) \) and \( p' = (\eta_1, \ldots, \eta_t, \eta_{b_{i+1}}) \) are two partial plays, then

\[
LVR(p) = LVR(p') \implies f(p) = f(p').
\]

We will give some example of \( LVR \) strategies in Sections 4 and 5. Strategies, winning strategies, no memory strategies, and \( LVR \) strategies for Red are defined in an analogous manner.

2.1. Büchi–Landweber \( \equiv \) McNaughton

Next we shall prove that Büchi–Landweber games and McNaughton games are essentially equivalent. The only substantial differences between Büchi–Landweber games and McNaughton games will turn out to be the conventions for the start of the games. That is, in a Büchi–Landweber game, Player I always starts first. Now there is a basic symmetry between the Red and Black players in a McNaughton game so that there is no real loss in generality in assuming Black plays first. This given, we shall show that for any Büchi–Landweber game \( G_{BL} \), there is a McNaughton game \( M(G_{BL}) \) and a fixed starting black node \( \eta_0 \) such that there is an effective 1:1 correspondence \( \Gamma \) between plays \( p_{BL} \) of \( G_{BL} \) and plays \( p_M \) of \( M(G_{BL}) \) which start at \( \eta_0 \) such that Player I wins \( p_{BL} \) iff Black wins \( \Gamma(p_M) \). Moreover, our correspondence \( \Gamma \)
will naturally induce a 1:1 correspondence between strategies and winning strategies for Player I (Player II) and strategies and winning strategies for Black (Red).

Similarly given any McNaughton game \( G_M \), we shall show that there is a Büchi–Landweber game \( BL(G_M) \) such that there is an effective 1:1 correspondence \( \Theta \) between plays \( p_M \) of \( G_M \) where Black begins the play and a certain subset \( L \) of plays \( p_{BL} \) of \( BL(G_M) \) which includes all possible winning plays for Player I such that Black wins \( p_M \) iff Player I wins \( \Theta(p_M) \). Moreover, the correspondence will naturally induce a 1:1 correspondence between strategies and winning strategies for Black (Red) and strategies and winning strategies for Player I (Player II). In fact, we shall show that our set \( L \) contains in some sense all nontrivial plays of the game in the sense that any play \( p_{BL} \in L \) is winning for Player II and the fact that Player II wins is determined by some finite initial segment of the play. This fact will become clear when we formally define \( \Theta \).

The correspondence \( \Gamma \)

Suppose that we are given a Büchi–Landweber game \( G_{BL} = (I,J,A,\Omega) \), where \( A = (I \times J, U, u_0, \psi) \) is a finite automaton and \( \Omega \subseteq \sum^U \). Then our corresponding McNaughton game \( M(G_{BL}) = (G,S,\Omega') \) is defined as follows. \( G = (V, V_B, V_R, E) \) is the bipartite directed graph whose set of Black nodes \( V_B = \{u_0\} \cup (U \times J) \) and whose set of Red nodes \( V_R = U \times I \). The set of directed edges \( E \) is defined as follows. First \( (u_0, (u_0, i)) \in E, \) for all \( i \in I \). Next for each \( (u,j) \in V \), where \( j \in J \), \( ((u,j),(u,i)) \in E \) for all \( i \in I \). Finally, if \( i \in I \) and \( (u,i) \in V \), then \( (u,i),(u',j)) \in E \) if and only if \( \psi((i,j),u) = u' \).

The starting node for Black is \( u_0 \). The correspondence \( \Gamma \) then maps a play \( p_{BL} = (i_0,j_0,i_1,j_1,\ldots) \) of \( G_{BL} \) to the play \( \Gamma(p_{BL}) = (u_0, (u_0,i_0), (u_1,j_0), (u_1,i_1), (u_2,j_1), (u_2,i_2), \ldots) \) where \( u_{k+1} = \psi((i_k,j_k),u_k) \) for all \( k \). It is clear that \( \Gamma \) is a 1:1 correspondence between the plays \( p_{BL} \) of \( G_{BL} \) and plays \( p_M \) of \( M(G_{BL}) \) where Black starts at \( u_0 \).

We define the set \( S = U \times J \) to be the set of significant nodes of \( M(G_{BL}) \). Given a subset \( R \subseteq S \), we let \( \pi_1(R) = \{u \in U: \exists j \in J((u,j) \in R)\} \). Thus \( \pi_1(R) \) is the just the set of all first coordinates of elements of \( R \). We then let \( \Omega' = \{R \subseteq U \times J: \pi_1(R) \in \Omega\} \).

It is easy to see from our definition of \( \Gamma \) that if \( p_{BL} = (i_0,j_0,i_1,j_1,\ldots) \) is a play of \( G_{BL} \) and \( \Gamma(p_{BL}) = (u_0,(u_0,i_0),(u_1,j_0),(u_1,i_1),(u_2,j_1),\ldots) \), then

\[
\pi_1(\text{perm}(\Gamma(p_{BL}))) = \text{perm}(p_{BL}).
\]

Hence Player I wins \( p_{BL} \) iff \( \text{perm}(p_{BL}) \in \Omega \) iff \( \text{perm}(\Gamma(p_{BL})) \in \Omega' \) iff Black wins \( \Gamma(p_{BL}) \). Finally suppose \( f \) is strategy for Player I and \( g \) is a strategy for Player II in \( G_{BL} \). Then we define a strategy \( \Gamma(f) \) for Black in \( M(G_{BL}) \) by

\[
\Gamma(f)((u_0,(u_0,i_0),(u_1,j_0),\ldots,(u_{k+1},j_{k})) = (u_{k+1},i_{k+1})
\]

for a partial play \( (u_0,(u_0,i_0),(u_1,j_0),\ldots,(u_{k+1},j_{k})) \) of \( M(G_{BL}) \) if and only if \( f(i_0,j_0,\ldots,i_k,j_k) = i_{k+1} \). Similarly, we define a strategy \( \Gamma(g) \) for Red in \( M(G_{BL}) \) by

\[
\Gamma(g)((u_0,(u_0,i_0),(u_1,j_0),\ldots,(u_k,i_k)) = (u_{k+1},j_{k+1})
\]
for a partial play \((u_0, (u_0, i_0), (u_1, j_0), \ldots, (u_k, i_k))\) iff \(g(i_0, j_0, \ldots, i_k) = j_k\) and \(u_{k+1} = \psi((i_k, j_k), u_k)\). Again it is straightforward to see that the map \(f \mapsto \Gamma(f)\) is a 1:1 correspondence between strategies for Player I in \(GBL\) and strategies for Black in \(M(GBL)\) restricted to partial plays which start at \(\eta_0\) and moreover \(f\) is a winning strategy for Player I in \(GBL\) iff \(\Gamma(f)\) is a winning strategy for Black in \(M(GBL)\). Similarly the map \(g \mapsto \Gamma(g)\) is a 1:1 correspondence between strategies for Player II in \(GBL\) and strategies for Red in \(M(GBL)\) restricted to partial plays which start at \(\eta_0\) such that \(g\) is winning strategy for Player II iff \(\Gamma(g)\) is a winning strategy for Red.

Remark. If we are willing to relax the condition that \(\Gamma\) is a 1:1 correspondence, it is possible to get a more efficient representation of \(M(GBL)\). That is, suppose we let \(G^* = (V^*, V_B^*, V_R^*, E^*)\), where \(V_B^* = U\) and \(V_R^* = U \times I\). The edges of \(E^*\) are \((u, (u, i))\) for \(i \in I\) and \(\langle (u, i), u' \rangle\) where there is a \(j \in J\) such that \(\psi((i, j), u) = u'\). Thus we as collapsing all the edges \(\langle (u, i), (u', j) \rangle\) in \(G\) to a single edge \(\langle (u, i), u' \rangle\). Then we can simply let \(S = U\) and \(\Omega' = \Omega\). The effect of this is to map all plays of \(M(GBL)\), \(p_M = (u_0, (u_0, i_0), (u_1, j_0), (u_1, i_1), (u_2, j_1), \ldots)\) to a single play

\[p_M^* = (u_0, (u_0, i_0), u_1, (u_1, i_1), \ldots)\]

If we do this we get a much smaller graph and the induced McNaughton game \(M^*(GBL) = (G^*, U, \Omega)\) would be a covering game for \(GBL\) in the sense of [20]. In this situation the correspondence \(\Gamma^*\) which sends a play \(p_B^L\) to \(\Gamma(p_B^L)^*\) still give us a many-one correspondence between plays, partial plays, strategies, and winning strategies of \(GBL\) and play, partial plays, strategies, and winning strategies of \(M^*(GBL)\) as long as we restrict ourselves to plays which start at \(u_0\). The reason that a smaller graph \(G^* = (V^*, V_B^*, V_R^*, E^*)\) and smaller set of significant nodes \(S\) is important is due to the fact that the algorithm for constructing winning strategies in McNaughton games runs in time \(O(|S|^2 |S| |I| |E|)\).

The correspondence \(\Theta\)

Suppose we are given a McNaughton game \(G_M = (G, S, \Omega)\), where \(G = (V, E)\) is a directed bipartite graph with black nodes \(V_B\) and red nodes \(V_R\). We then define a Büchi–Landweber game \(BL(G_M) = (I, J, A, \Omega')\) as follows. We let \(I = V_B\) and \(J = V_R\). Now we finite automaton \(A = (I, J, U, \psi, \psi)\) is defined as follows. The set of states \(U = \{u_0, D\} \cup \{(i, j): (i, j) \in (I \times J) \cap E\}\). Here \(D\) is some default state and \(u_0\) is our start state. The transition function \(\psi\) is defined so that

1. If \((i, j) \notin E\), then \(\psi((i, j), u) = D\) for all \(u \in U\)
2. If \((i, j) \in E\), then
   1. \(\psi((i, j), D) = D\)
   2. \(\psi((i, j), u_0) = (i, j)\)
   3. \(\psi((i, j), (i_0, j_0)) = \begin{cases} D & \text{if } (j_0, i) \notin E, \\ (i, j) & \text{if } (j_0, i) \in E. \end{cases}\)

Our idea is to have the correspondence \(\Theta\) be trivial. That is, given a play \(p_M = (\eta_0, \eta_0, \eta_1, \eta_1, \ldots\) of \(G_M\) where \(\eta_i \in V_B\) and \(\eta_i \in V_R\) for all \(i\), we simply let
$\Theta(p_M) = p_M$. Now the only plays $p_{BL} = (i_0, j_0, i_1, j_1, \ldots)$ not in the range of $\Theta$ are those plays where either (i) there is some $k$ such that $(i_k, j_k) \notin E$ or (ii) there is some $l$ such that $(j_l, i_{l+1}) \notin E$. For such plays, consider the sequence of states induced by $\psi$, $u_0, u_1 = \psi((i_0, j_0), u_0), u_2 = \psi((i_1, j_1), u_1), \ldots$. Clearly in case (i), $u_{k+1} = \psi((i_k, j_k), u_k) = D$ by (1) and then by 2(a), we will have that $u_r = D$ for all $r \geq k+1$. Thus the perm set of $p_{BL}$ will be $\{D\}$. If we are not in case (i) so that $(i_k, j_k) \in E$ for each $k$, then it is easy to see that $u_k = (i_{k-1}, j_{k-1})$ for all $k$, until we reach the first $l$ such that $(j_l, i_{l+1}) \notin E$. Then by 2(c), $u_{l+2} = \psi((i_{l+1}, j_{l+1}), u_{l+1}) = \psi((i_{l+1}, j_{l+1}), (i_l, j_l)) = D$ and then by 2(a), we know $u_r = D$ for all $r \geq l + 2$. Thus again $\text{perm}(p_{BL}) = \{D\}$. In fact, it is easy to see our definition of $\psi$ ensures that $\text{perm}(p_{BL}) = \{D\}$ iff $p_{BL} \notin \text{range } \Theta$. Of course we shall not put $\{D\}$ into $\Gamma$ so that Red will automatically win all plays $p_{BL} \notin \text{range } \Theta$.

Observe that if $p_M = (\eta_0, \eta_1, \eta_2, \ldots)$ is a play of $G_M$ where Black starts, then the sequence of states

$$(u_0, u_1 - \psi((i_0, j_0), u_0), u_2 - \psi((i_1, j_1), u_1), \ldots)$$

induced by $\psi$ for the play $\Theta(p_M) = p_M$ of $BL(G_M)$ will simply have $u_k = (i_{k-1}, j_{k-1})$ for all $k \geq 1$. Then given a pair $(i, j) \in (I \times J) \cap E = U - \{u_0, D\}$, define $(i, j)^S = S \cap \{i, j\}$. The winning sets $\Omega'$ of the game $BL(G_M)$ consists of all sets $T \subseteq \{(i, j) \in U \cap E \}$ such that $T^S = \bigcup \{(i, j)^S : (i, j) \in T\} \in \Omega$. It is easy to see that for any play $p_M$ of $G_M$, the perm set of $p_M$ relative to $G_M$, $\text{perm}_M(p_M)$, and the perm set of $p_M$ relative to $BL(G_M)$, $\text{perm}_{BL}(p_M)$, are related by $\text{perm}_M(p_M) = \text{perm}_{BL}(p_M)^S$. It thus follows that Black wins $p_M$ in $G_M$ iff $\text{perm}_M(p_M) \in \Omega$ iff $\text{perm}_{BL}(p_M) \in \Omega'$ iff Player I wins $p_M$ relative to $BL(G_M)$.

There is also a simple correspondence between strategies in $G_M$ and $BL(G_M)$. Namely, we say that a strategy $f$ for Player I is edge preserving in $BL(G_M)$ iff for every partial play $(i_0, j_0, \ldots, i_k, j_k)$ such that $(i_s, j_s), (j_{s+1}, i_{s+1}) \in E$ for all $s \leq k$ and $(j_s, i_{s+1}) \in E$, we say that $f$ is a winning edge preserving strategy for Player I if $f$ is an edge preserving strategy such that wherever Player I follows $f$, Player I will win as long as Player II plays in an edge preserving way, i.e. as long as for any partial play $(i_0, j_0, \ldots, i_k, j_k, i_{k+1})$ such that $(i_s, j_s), (j_s, i_{s+1}) \in E$ for a partial play $(i_0, j_0, \ldots, i_k, j_k, i_{k+1})$ such that $(i_s, j_s) \in E$ for all $s \leq k$ and $(j_s, i_{s+1}) \in E$, then Player II next move is some $j_{k+1}$ such that $(i_{k+1}, j_{k+1}) \in E$. Edge preserving and winning edge preserving strategies for Player I are defined similarly. This given, it is easy to see that strategy for Black (Red) in $G_M$ is just the restriction of an edge preserving strategy for Player I (Player II) in $BL(G_M)$. Moreover, $f$ is an edge preserving winning strategy for Player I (Player II) in $BL(G_M)$ iff $f$ restricts to a winning strategy for Black (Red) in $G_M$.

3. A McNaughton game for the Consumer–Producer problem

As stated in the introduction, one of our main interests in McNaughton games is to use the algorithm to extract winning strategies in such games as part of a general
algorithm which can extract correct concurrent programs or extract a control automaton which can guarantee that a plant meets its performance specifications in the setting of hybrid systems. In this section, we describe a game which is a simplified version of games that might arise in such applications. The basic problem is the so-called Consumer–Producer problem in which we have a stack of fixed size \( n \), the Producer adds to the top of the stack, and the Consumer takes from the top of the stack. The constraints of the problem are that the Producer should never attempt to add to a full stack, the Consumer should eventually consume everything that the Producer puts on the stack. There are a number of places where this type of problem can arise, for example in message passing problems in certain computer architectures. Our aim is to extract strategies for both the Producer and the Consumer by finding winning strategies from a McNaughton game. One possible game is the following. The black nodes of our graph will consist of nodes of the following form:

(a) \( (P, i, \text{produce}) \) for \( 0 \leq i \leq n \),
(b) \( (C, i, \text{consume}) \) for \( 0 < i < n \), and
(c) \( \text{Fail}(C, B) \) and \( \text{Fail}(P, B) \).

Here \( (P, i, \text{produce}) \) is a node which is intended to indicate that the Producer is in control, the stack level is \( i \), and the Producer’s action is to add to the stack. Similarly \( (C, i, \text{consume}) \) is a node which is intended to indicate that the Consumer is in control, the stack level is \( i \), and the Consumer’s action is to take from the stack. \( \text{Fail}(P, B) \) and \( \text{Fail}(C, B) \) are default nodes which are reached only if, respectively, the Producer tries to add to a full stack or the Consumer tries to consume from an empty stack. The red nodes in our graph are intended to allow for decisions on whether the Producer wants to pass control to the Consumer or vice versa. The red nodes of our graph will be the following.

(a) \( (P, i, \text{pass}) \) and \( (P, i, \text{keep}) \) for \( 1 < i < n \),
(b) \( (C, i, \text{pass}) \) and \( (C, i, \text{keep}) \) for \( 0 < i < n - 1 \), and
(c) \( \text{Fail}(C, R) \) and \( \text{Fail}(P, R) \).

The edges of our graph are the following. First we consider edges which go from Red nodes to Black nodes. For any \( i \), \( (P, i, \text{pass}) \) is connected to \( (C, i, \text{consume}) \) to represent the fact that the Producer has passed control to the Consumer who will then proceed to take something off the stack. Similarly for any \( i \), \( (C, i, \text{pass}) \) is connected to \( (P, i, \text{produce}) \) to represent the fact that the Consumer has passed control to the Producer who will then proceed to add something to the stack. Also for any \( i \), \( (P, i, \text{keep}) \) is connected to \( (P, i, \text{produce}) \) to represent the fact that the Producer keeps control and the Producer will add again to the stack. Similarly for an \( i \), \( (C, i, \text{keep}) \) is connected to \( (C, i, \text{consume}) \) to represent that Consumer keeps control and the Consumer will again take from the stack. Finally, \( \text{Fail}(C, R) \) is connected to \( \text{Fail}(C, B) \) and \( \text{Fail}(P, R) \) is connected to \( \text{Fail}(P, B) \). Next we consider the edges from Black nodes to Red nodes. Any \( (P, i, \text{produce}) \) where \( 0 \leq i \leq n - 1 \) is connected to both \( (P, i + 1, \text{pass}) \) and \( (P, i + 1, \text{keep}) \). Here \( (P, i, \text{produce}) \) represents the action that Producer has added to
the stack so that the next stack size is \( i + 1 \) and Producer can either pass or keep control. Also \( (P, n, \text{produce}) \) is connected to \( \text{Fail}(P,R) \) to represent the fact that Producer has attempted to add to a full stack and hence we have an error situation caused by Producer. Any \( (C,i,\text{consume}) \) where \( 1 \leq i \leq n \) is connected to both \( (C,i - 1, \text{pass}) \) and \( (C,i - 1, \text{keep}) \). Here \( (C,i,\text{consume}) \) represents the action that Consumer has taken from the stack so that the new stack size is \( i - 1 \) and Consumer can either pass or keep control. Also \( (C,0,\text{consume}) \) is connected to \( \text{Fail}(C,R) \) to represent that Consumer has attempted to take from an empty stack and hence we have an error situation caused by Consumer. Finally, \( \text{Fail}(C,B) \) is connected to \( \text{Fail}(C,R) \) and \( \text{Fail}(P,B) \) is connected to \( \text{Fail}(P,R) \). The graph \( G \) when the limit on the stack size is 3 is pictured below in Fig. 1.

It is easy to see that any play of the game will represent an infinite series of additions and deletions from the stack unless we hit one the \( \text{Fail}(\_ , \_ ) \) nodes in which case we will loop among the \( \text{Fail}(\_ , \_ ) \) nodes forever. To complete the specification of

![Graph for the Producer-Consumer game.](image)

Fig. 1. Graph for the Producer–Consumer game.
the game we must specify the set of significant nodes $S$ and the set of set of winning
sets $\Omega \subseteq 2^S$. The choice of $S$ and $\Omega$ will depend on what behavior we would like
our winning strategies to have. For example, if we just want to ensure that in any
play which Black wins, eventually, Consumer consumes everything on the stack, then
all we have to ensure is that Consumer consumes from a stack of size 1 infinitely
often and avoids any of the failure nodes $\text{Fail}(\_, \_\_$. In this case, we might specify
$S = \{(C, 1, \text{consume})\}$ and set $\Omega = \{S\}$. Thus any winning play for Black will have to
visit $(C, 1, \text{consume})$ infinitely often. However, we can adjust $S$ to ask for more refined
behavior in the winning plays. For example, if we want to stack to be full infinitely
often during a winning play for Black, we could set $S = \{(C, 1, \text{consume}), (C, n,\n\text{consume})\}$ and $\Omega = \{S\}$. In this way, we can force a successful winning strategy
for Black to ensure that the stack reaches certain configurations repeatedly and this
is very similar to classical control problems where one wants to ensure that the plant
reaches a certain state repeatedly without violating any constraints.

In this light, the results of Section 5 take on a much greater importance. That is, if
we can arrange it so that the translation of our problem into a McNaughton game $G_M$
results in a game whose sets of winning sets $\Omega$ a particularly nice form, then we can
guarantee that our extraction algorithm runs in a reasonable time. Thus it is desirable
to have more results like those of Theorem 5.5 and Theorem 5.6.

4. The complexity of finding winning strategies in McNaughton games

In [8], McNaughton showed that in any McNaughton game $G_M = (G, S, E)$ where
$G = (V, V_B, V_R, E)$, one can partition the set of vertices into two sets, $X$ and $Y$, where
Black has a LVR winning strategy to win any play which starts in $X$ and Red has LVR
winning strategy to win any play which starts in $Y$. McNaughton gave an algorithm to
find $X$ and $Y$ and showed that his algorithm runs in $O(c^{1|S| |V|^3})$ steps. McNaughton
did not explicitly construct the corresponding LVR strategies for Black and Red in his
algorithm. However an implicit description of such LVR strategies is contained in his
algorithm. We want to apply McNaughton’s algorithm or more efficient variants of his
algorithm to extract explicit strategies in certain applications. We shall give a careful
analysis of McNaughton algorithm, paying careful attention to how one can construct
the LVR strategies. We shall show that even when we explicitly construct the LVR
strategies in addition to constructing $X$ and $Y$, we can improve McNaughton bound
for the running time to $O(c^{1|S| |V|^3})$.

Given a McNaughton game $G_M = (G, S, \Omega)$, where $G = (V, V_R, V_B, E)$ define $Win_B$
($G_M$) ($Win_R(G_M$)) to be the set of all nodes $\eta$ in $V$ such that there is a LVR strategy
$f_\eta$ for Black (Red) such that Black (Red) wins all plays $p$ such that $p$ starts at $\eta$
and Black (Red) follows $f_\eta$. For any set $A \subseteq V$, we say that $f$ is a LVR (no memory)
strategy from a set $A \subseteq V$ for a player $P$ if $f$ is a LVR (no memory) strategy and in
any play $p = (\eta_0, \eta_1, \ldots)$ where $P$ follows $f$ and $\eta_0 \in A$, $P$ wins. In [8], McNaughton
proved the following.
Theorem 4.1 (McNaughton [8]). Let \( G = (V, E, \Omega) \) be a McNaughton game, where \( G = (V_R, V_B, E) \). Then \( \text{Win}_B(\mathcal{G}_M) \) and \( \text{Win}_R(\mathcal{G}_M) \) partition \( V \). Moreover, there is an algorithm which runs in \( O(c^{(|V|/2)}) \) steps to find \( \text{Win}_B(\mathcal{G}_M) \) and \( \text{Win}_R(\mathcal{G}_M) \) for some constant \( c \).

McNaughton also claims that "the strategies" (for \( \text{Win}_B(\mathcal{G}_M) \) and \( \text{Win}_R(\mathcal{G}_M) \), respectively) can be effectively determined from \( \mathcal{G}_M \). While it is clear from McNaughton's proof that such strategies exist, McNaughton does not construct such strategies explicitly in his algorithm. Moreover, he never makes it clear as to whether the strategies that he is talking about are individual \( LVR \) strategies \( f_\eta \) for each \( \eta \in V \) or are global strategies \( f \), where \( f_\eta \) is a \( LVR \) winning strategy for Red from \( \text{Win}_R(\mathcal{G}_M) \) and \( f \) is a \( LVR \) winning strategy for Black from \( \text{Win}_B(\mathcal{G}_M) \). McNaughton does prove the following lemma which makes it clear that the strategies \( f_\eta \) and \( f \) can be constructed from the individual strategies \( \{f_\eta : \eta \in V \} \). That is, McNaughton proved the following.

Theorem 4.2 ([8, Theorem 3.1]). If a player \( P \) in a McNaughton game \( \mathcal{G}_M \) has \( LVR \) (no memory) strategies \( f \) and \( g \) such that \( f \) is a winning \( LVR \) (no memory) strategy from \( A \) for \( P \) and \( g \) is a winning \( LVR \) (no memory) strategy from \( B \) for \( P \), then there is a winning \( LVR \) (no memory) strategy \( h \) from \( A \cup B \) for \( P \).

Proof. The desired strategy \( h \) is quite easy to construct. Let \( \mathcal{L} \) consist of all possible sequences from the set \( S \) of significant nodes of \( \mathcal{G}_M \). Note that \( |\mathcal{L}| = 1 + |S| + (|S| - 1)|S| + (|S| - 2)(|S| - 1)|S| + \cdots + |S|! \). It is easy to prove by induction on \( |S| \) that \( |\mathcal{L}| \leq 5/2|S|! \).

A \( LVR \) strategy is a map \( f : \mathcal{L} \times V \to V \) such that \((\eta, f((L, \eta))) \) is an edge for all \( \eta \in V \) and \( L \in \mathcal{L} \). Thus we could specify a \( LVR \) strategy by giving an array of size \( |\mathcal{L}| \times |V| \). A no memory strategy \( f \) is just a map \( f : V \to V \) such that \((\eta, f(\eta)) \) is an edge for all \( \eta \in V \) and hence a no memory strategy can be specified by an array of size \( |V| \). For a \( LVR \) strategy \( f \) for \( P \) from \( A \), we say that \((L, \eta) \in \mathcal{L} \times V \) is \( f \)-reachable from \( A \) if there is a partial play \( p = (\eta_0, \eta_1, \ldots, \eta_k) \) in which \( P \) follows \( f \) such that \( \eta_0 \in A \), \( \eta_k = \eta \), and \( LVR(\eta_0, \eta_1, \ldots, \eta_k) = L \). Similarly if \( f \) is a no memory strategy for \( P \) from \( A \), we say that \( \eta \) is \( f \)-reachable from \( A \) if there is a partial play \( p = (\eta_0, \ldots, \eta_k) \) in which \( P \) follows \( f \) such that \( \eta_0 \in A \) and \( \eta_k = \eta \).

Now suppose that \( f \) is a winning \( LVR \) strategy for \( P \) from \( A \) and \( g \) is a winning \( LVR \) strategy for \( P \) from \( B \). Then it is easy to check that \( h \) is a winning \( LVR \) strategy for \( P \) from \( A \cup B \) where for each \((L, \eta) \in \mathcal{L} \times V \),

\[
h((L, \eta)) = \begin{cases} 
    f((L, \eta)) & \text{if } (L, \eta) \text{ is } f \text{-reachable from } A \text{ for } P, \\
    g((L, \eta)) & \text{otherwise}. 
\end{cases}
\]

If \( f \) is a winning no memory strategy for \( P \) from \( A \) and \( g \) is a winning no memory strategy for \( P \) from \( B \), then we can define a winning no memory strategy for \( P \) from
A ⊃ B by setting

\[ L(\eta) = \begin{cases} 
  f(\eta) & \text{if } \eta \text{ is } f\text{-reachable from } A, \\
  g(\eta) & \text{otherwise.}
\end{cases} \]

From the proof of Theorem 4.2, it is clear that the complexity of combining strategies \( f \) and \( g \) for \( P \) from \( A \) and \( B \), respectively, into a single strategy for \( P \) from \( A \cup B \) depends on the complexity the \( f \)-reachability predicate.

**Theorem 4.3.** Let \( G = (G, S, \Omega) \) be a McNaughton game, where \( G = (V, V_B, V_R, E) \) and \( L \) be the set of all last visitation records for \( G \). Then

(a) if \( f \) is a \( LVR \) winning strategy for a player \( P \) from \( A \subseteq V \), then we can construct \( B_f,A = \{ (L, \gamma) \in L \times V : (L, \gamma) \text{ is } f\text{-reachable for } P \text{ from } A \} \) in \( O(|E| |S|!) \) steps and

(b) if \( f \) is a no memory winning strategy for a player \( P \) from \( A \subseteq V \), then we can construct \( C_f,A = \{ \eta \in V : \eta \text{ is } f\text{-reachable for } P \text{ from } A \} \) in \( O(|E|) \) steps.

**Proof.** (a) First define a function \( N : L \times V \rightarrow L \) by

\[ N(L, v) = \begin{cases} 
  L & \text{if } v \not\in S, \\
  L \setminus v & \text{if } v \in S \text{ and } v \text{ does not occur in } L, \\
  (L/v) \setminus v & \text{if } v \in S \text{ and } v \text{ does occur in } L,
\end{cases} \]

where \( L/v \) denotes the result of removing \( v \) from \( L \) if \( v \) occurs in \( L \). For example, \( (s_1, s_3, s_2, s_4)/s_2 = (s_1, s_3, s_4) \). Note that if \( LVR(\eta_0, \ldots, \eta_k) = L \), then our definition of \( N(L, v) \) is designed to ensure that \( LVR(\eta_0, \ldots, \eta_k, v) = N(L, v) \).

Now construct a directed graph \( G(f) \) as follows. The set of vertices of \( G(f) \) is \( L \times V \). Assume that the player \( P \) is Red. Then if \( v \in V_R \), \( ((L, v), (L', v')) \) is an edge of \( G(f) \) iff \( v' = f((L, v)) \) and \( L' = N(L, v) \). If \( v \in V_B \), then \( ((L, v), (L', v')) \) is an edge in \( G(f) \) iff \( v, v' \) is an edge in \( G \) and \( L' = N(L, v) \). It is then easy to see that there is a 1:1 correspondence \( I \) between partial plays \( p = (\eta_0, \eta_1, \ldots, \eta_k) \) in which Red follows \( f \) and paths in \( G(f) \) which begin at a node of the form \( (LVR(\eta_0), \eta_0) \) given by

\[ I(p) = ((LVR(\eta_0), \eta_0), (LVR(\eta_0, \eta_1), \eta_1), \ldots, (LVR(\eta_0, \ldots, \eta_k), \eta_k)). \]

It follows that \( (L, \gamma) \) is \( f\text{-reachable for Red from } A \) iff there is a path in \( G(f) \) which starts at \( (LVR(\eta_0), \eta_0) \) for some \( \eta_0 \in A \) which ends in \( (L, v) \). That is, \( (L, \gamma) \) is \( f\text{-reachable for Red from } A \) iff \( (L, \gamma) \) is in the union of the connected components of \( G(f) \) which contain \( \{ (LVR(\eta_0), \eta_0) : \eta_0 \in A \} \). This union of connected components can be easily found in \( O(|E(f)|) \) steps where \( E(f) \) is the set of edges of \( G(f) \). Note that each edge \( (v, w) \in E \) gives rise to at most \( |L| \) edges in \( G(f) \), namely \( \{(L, v), (N(L, w), w)\} \), in \( E(f) \) so that \( |E(f)| \leq |L| \times |E| \leq 5/2|S|! |E| \). Now \( |E| \geq |V| \) in all graphs \( G = (V, V_B, V_R, E) \) which underlie a McNaughton game. Thus it easily follows we can find the set of all \( f\text{-reachable pairs for Red from } A \) in \( O(|E| |S|!) \) steps given \( f \). A similar argument applies if the player \( P \) is Black.

(b) Again assume that the player \( P \) is Red. In this case, let the graph \( G(f) \) be the graph that results from \( G \) by removing all edges \( (v, w) \), where \( v \in V_R \) and \( w \neq f(v) \).
Again it is easy to see that $C_{f,A}$ is just the union of the connected components of $G(f)$ which contain $A$ and can be constructed in $O(|E(f)|)$ steps. Thus since $|E(f)| \leq |E|$, $C_{f,A}$ can be constructed in $O(|E|)$ steps. A similar argument holds if $P$ is the player Black. \hfill $\blacksquare$

We note that McNaughton never explicitly computed the complexity of combining strategies. Since we want to give an explicit bound for the complexity of computing the strategies which are implicit in McNaughton's algorithm, we shall need an explicit bound on the cost of combining strategies.

**Theorem 4.4.** Let $G_M = (G, S, \Omega)$ be a McNaughton game, where $G = (V, V_R, V_R, E)$ and $A = \bigcup_{i=1}^{k} A_i$.

(a) Suppose that we are given $f_1, \ldots, f_k$ which are $LVR$ winning strategies for a player $P$ from sets $A_1, \ldots, A_k$, respectively. Then we can construct a $LVR$ winning strategy $f$ for $P$ from $A$ in $O(k|E||S|!)$ steps.

(b) Suppose that we are given $f_1, \ldots, f_k$ which are no memory winning strategies for a player $P$ from sets $A_1, \ldots, A_k$, respectively. Then we can construct a no memory winning strategy $f$ for $P$ from $A$ in $O(k|E|)$ steps.

**Proof.** (a) Let $L$ be the set of last visitation records based on $S$. They by Theorem 4.3, for each $i$, it takes $O(|E||S|!)$ to compute the function $\varepsilon_i$ where for each $(L, \eta) \in L \times V$,

$$\varepsilon_i(L, \eta) = \begin{cases} 1 & \text{if $(L, \eta)$ is } f_i\text{-reachable for } P \text { from } A_i, \\ 0 & \text{otherwise.} \end{cases}$$

Then we claim our desired $f$ is given by setting for $(L, \eta) \in L \times V_P$,

$$f(L, \eta) = \begin{cases} f_i(L, \eta) & \text{where } i \text{ is the least } j < k \text{ such that } \\ \varepsilon_j(\eta) = 1 \text{ if there is such an } i, \\ f_k(L, \eta) & \text{otherwise}. \end{cases}$$

To see that $f$ is a $LVR$ winning strategy for $P$ from $A$, suppose that $P = \text{Red}$ and $r = (\eta_0, \eta_1, \eta_2, \ldots)$ is a play, where $\eta_0 \subseteq A$ and Red follows $f$. Suppose $\eta_0 \in V_R$ so that $\forall i (\eta_{2i+1} = f(LVR(\eta_0, \ldots, \eta_{2i}), \eta_{2i+1}))$. Let $h$ be the smallest $j$ such that there is an $i$ where $(LVR(\eta_0, \ldots, \eta_i), \eta_i)$ is $f_j$-reachable for Red from $A_i$. Thus there is a partial play $(\gamma_0, \ldots, \gamma_r)$ in which Red follows $f_h$ where $\gamma_0 \in A_h$, $\gamma_r = \eta_i$, and $LVR(\gamma_0, \ldots, \gamma_r) = LVR(\eta_0, \ldots, \eta_i)$. But then we claim that in the rest of the play $(\eta_i, \eta_{i+1}, \ldots)$, Red follows $f_h$. That is, suppose by induction on $m \geq i$ that in the partial play $(\eta_0, \ldots, \eta_m)$, Red is following the strategy $f_h$ from $\eta_i$ on. Then $(\gamma_0, \ldots, \gamma_r = \eta_i, \ldots, \eta_m)$ is a play starting from $A_h$ in which Red follows $f_h$. Moreover, since $LVR(\eta_0, \ldots, \gamma_r) = LVR(\eta_0, \ldots, \eta_i)$, it is easy to see that $LVR(\eta_0, \ldots, \gamma_r = \eta_i, \eta_{i+1}, \ldots, \eta_m) = LVR(\eta_0, \ldots, \eta_m)$ so that $LVR(\eta_0, \ldots, \eta_m)$ is $f_k$-reachable for Red from $A_h$. Moreover by our definition of $h$, $LVR(\eta_0, \ldots, \eta_m, \eta_m)$ is not $f_i$-reachable from $A_i$ for Red for any $i < h$. Thus if $\eta_m \in V_R$, then

$$\eta_{m+1} = f(LVR(\eta_0, \ldots, \eta_m), \eta_m) = f_h(LVR(\eta_0, \ldots, \eta_m), \eta_m)$$
so that \((\eta_0, \ldots, \eta_{m+1})\) is a partial play in which Red is following \(f_h\) from \(\eta_i\) on. If \(\eta_m \in V_B\), then automatically \((\eta_i, \ldots, \eta_{m+1})\) is a partial play in which Red is following \(f_h\). Thus Red follows \(f_h\) in the play \((\eta_0, \eta_1, \ldots, \eta_i, \eta_{i+1}, \ldots)\) from \(\eta_i\) on. Hence the play \((\gamma_0, \ldots, \gamma_r = \eta_i, \eta_{i+1}, \ldots)\) is a winning play for Red since \(\gamma_0 \in A_h\). But clearly \(\text{perm}(\eta_0, \eta_1, \ldots) = \text{perm}(\gamma_0, \ldots, \gamma_r = \eta_i, \eta_{i+1}, \ldots)\) so that \((\eta_0, \eta_1, \ldots)\) is also a winning play for Red. The same argument will show that if \(\eta_0 \in V_B\), the play \((\eta_0, \eta_1, \ldots)\) is also winning for Red. It follows that every play \((\eta_0, \eta_1, \ldots)\), which starts in \(A\) and in which Red is following \(f\), will be a win for Red. Thus \(f\) is an LVR winning for Red from \(A\). A similar argument applies if \(P = B\).

Note that to determine \(f(L, \eta)\) for any fixed \((L, \eta) \in \mathcal{L} \times V\), we need only find the first \(i\) among \(E_1(L, \eta), \ldots, E_k(L, \eta)\) which is equal to 1 or determine that there is no such \(i\). Thus it takes \(O(k)\) steps to determine \(f(L, \eta)\) for each \((L, \eta) \in \mathcal{L} \times V\) given \(f_1, \ldots, f_k\) and \(e_1, \ldots, e_k\). It takes \(O(k|E||S|!\) steps to compute \(e_1, \ldots, e_k\). Thus it takes at most \(k|V| \leq |E|\) additional steps to construct \(f\) given \(f_1, \ldots, f_k\) and \(e_1, \ldots, e_k\). Thus, \(f\) can be constructed in \(O(k|E|)|S|!\) steps.

(b) By a similar argument, one can show that the desired no memory winning strategy \(f\) for \(P\) from \(A\) is defined by setting for \(\eta \in V_P\),

\[
f(\eta) = \begin{cases} f_i(\eta) & \text{where } i \text{ is the least } j < k \text{ such that } e_j(\eta) = 1 \text{ if there is such an } i, \\ f_k(\eta) & \text{otherwise,} \end{cases}
\]

where \(e_i\) is the characteristic function of \(f_i\)-reachability predicate for \(P\) from \(A_i\) for \(i = 1, \ldots, k\). By Theorem 4.3, it takes \(O(k|E|)\) steps to compute \(e_1, \ldots, e_k\). Clearly it takes at most \(k|V| \leq k|E|\) additional steps to construct \(f\) given \(f_1, \ldots, f_k\) and \(e_1, \ldots, e_k\). Thus, \(f\) can be constructed in \(O(k|E|)|S|!\) steps. \(\square\)

Next we shall show that with appropriate data structures, we can find the sets \(\text{Win}_B(\mathcal{G}_M)\) and \(\text{Win}_R(\mathcal{G}_M)\) in \(O(|S||E|\text{odd!}(|S|))\), where

\[
\text{odd!}(k) = \prod_{i=1}^{k} (2i - 1) \quad \text{for } k \geq 1.
\]

Similarly, we shall show that we can find the sets \(\text{Win}_B(\mathcal{G}_M)\) and \(\text{Win}_R(\mathcal{G}_M)\) and the LVR strategies \(f_b\) and \(f_r\) in \(O(2|S||E|\text{odd!}(|S|))\). Thus, we improve on McNaughton’s complexity bound for finding \(\text{Win}_B(\mathcal{G}_M)\) and \(\text{Win}_R(\mathcal{G}_M)\) by replacing the factor \(|V|^3\) by \(|E|\) and by replacing the factor \(c^{|S|} \cdot |S|!\) by the more explicit expression \(|S|\text{odd!}(|S|)).\) Moreover, we also show that we can find \(\text{Win}_B(\mathcal{G}_M)\) and \(\text{Win}_R(\mathcal{G}_M)\) plus the LVR strategies \(f_b\) and \(f_r\) in a time bound which is still better than McNaughton’s original time bound for just finding \(\text{Win}_B(\mathcal{G}_M)\) and \(\text{Win}_R(\mathcal{G}_M)\).

**Theorem 4.5.** Let \(\mathcal{G}_M = (G, S, \Omega)\) be a McNaughton game, where \(G = (V, V_b, V_R, E)\).

(a) There is an algorithm which, given \(\mathcal{G}_M\), runs in \(O(|S||E|\text{odd!}(|S|))\) steps and produces \(\text{Win}_B(\mathcal{G}_M)\) and \(\text{Win}_R(\mathcal{G}_M)\).

(b) There is an algorithm which, given \(\mathcal{G}_M\) runs in \(O(2|S||E|\text{odd!}(|S|))\) steps and produces \(\text{Win}_B(\mathcal{G}_M)\), \(\text{Win}_R(\mathcal{G}_M)\), a LVR winning strategy \(f_b\) for Black from \(\text{Win}_B(\mathcal{G}_M)\), and a LVR winning strategy \(f_r\) for Red from \(\text{Win}_R(\mathcal{G}_M)\).
Proof. We shall basically follow the algorithm that was used by McNaughton to prove Theorem 4.1. Before we can proceed with our description of the algorithm, we need to state some basic definitions and prove a few key lemmas.

Our first lemma is how to compute last visitation records which result from concatenating two partial plays. Suppose \( L_1 \) and \( L_2 \) are two elements of \( \mathcal{L} \), the set of all last visitation records in \( \mathcal{G}_M \). Let \( L_1 / L_2 \) denote the last visitation record that results when we delete from \( L_1 \) all occurrences of elements of \( L_2 \). For example, if \( L_1 = (s_2, s_3, s_4) \) and \( L_2 = (s_4, s_1) \), then \( L_1 / L_2 = (s_2, s_3) \). Then the following is easily proved by induction on \(|L_2|\).

**Lemma 4.6.** Let \((\eta_0, \ldots, \eta_k)\) and \((\eta_{k+1}, \ldots, \eta_r)\) be partial plays of a McNaughton game \( \mathcal{G}_M = (G, S, \Omega) \) and let \( L_1 = \text{LVR}(\eta_0, \ldots, \eta_k) \) and \( L_2 = \text{LVR}(\eta_{k+1}, \ldots, \eta_r) \). Then \( \text{LVR}(\eta_0, \ldots, \eta_k, \eta_{k+1}, \ldots, \eta_r) = (L_1 / L_2)L_2 \).

Our next lemma states that there is an obvious closure condition that is satisfied by the set of points \( v \in V \) from which there is LVR winning strategy for \( P \) from \( \{v\} \).

**Lemma 4.7 ([8, Theorem 3.2]).** If a player \( P \) in a McNaughton game \( \mathcal{G}_M \) has a LVR (no memory) winning strategy \( f \) from \( \{v\} \subseteq V \) and \( w \) is visited in some play \( p = (\eta_0 = v, \eta_1, \eta_2, \ldots) \) in which \( P \) follows \( f \), then \( P \) has a LVR (no memory) winning strategy from \( \{w\} \).

One of the key ingredients which will be used repeatedly in our algorithm is our ability to solve the following type of problem. Suppose \( D \) and \( H \) are contained in \( V \) and \( D \cap H = \emptyset \). For a player \( P \), we want to construct a set of nodes \( N \subseteq V \) such that \( P \) has a no-memory strategy \( f \) to force any play which starts at some \( \eta \in N \) to visit \( D \) in 0 or more moves before it visits \( H \). McNaughton proved that one could construct \( N \) in \( O(|V|^3) \) steps. We shall actually show that we can construct \( N \) in \( O(|E|) \) steps.

**Lemma 4.8.** Let \( \mathcal{G}_M = (G, S, \Omega) \) be a McNaughton game, where \( G = (V, V_B, V_R, E) \). Let \( D, H \subseteq V, D \cap H = \emptyset \), and \( X_{D,H,P} \) be the set of nodes in \( V - H \) from which a player \( P \) has a no-memory strategy \( f_{D,H,P} \) to force any play starting at \( \eta \) to visit \( D \) in 0 or more moves before it can visit \( H \). Then given \( D \) and \( H \), we can construct \( f_{D,H,P} \) and \( X_{D,H,P} \) in \( O(|E|) \) steps.

**Proof.** We set up a data structure so that each \( \eta \in V \) has two lists associated to it, namely, \( \text{In}(\eta) \) consisting all nodes \( \gamma \) such that \( (\gamma, \eta) \in E \) and \( \text{Out}(\eta) \) consisting of all nodes \( \beta \) such that \( (\eta, \beta) \in E \). Moreover, we assume that we have a set of pointers for each \( \eta \) which point to any occurrence of \( \eta \) on any list \( \text{In}(\beta) \) or \( \text{Out}(\beta) \). Assume that the player, \( P \) is Red. It will be clear that our argument is symmetric with respect to the roles of Red and Black so that there is no loss in generality in making this assumption. Now let \( \gamma_H(D) \) consist of the set of \( \eta \in V - H \) such that either \( \eta \in D \) or Red can force a play to visit \( D \) in one step. Then clearly a red node \( r \in \gamma_H(D) \) if
either \( r \in D \) or \( r \in V - (H \cup D) \) and \( \text{Out}(r) \cap D \neq \emptyset \). A black node \( b \in \gamma_H(D) \) if either \( b \in D \) or \( b \in V - (D \cup H) \) and \( \text{Out}(b) \subseteq D \). These two possibilities are pictured in Fig. 2.

Now clearly we can construct \( N = X_{D,H,R} \) by iterating the \( \gamma_H \) operator. That is, \( N = \gamma_H^{(r)}(D) \), where \( r \) is the least \( k \) such that \( \gamma_H^{(k)}(D) = \gamma_H^{(k+1)}(D) \) where we define \( \gamma_H^{(r)}(D) \) by induction by setting \( \gamma_H^{(0)}(D) = D \) and \( \gamma_H^{(r+1)}(D) = \gamma_H(\gamma_H^{(r)}(D)) \).

We can also construct \( N \) using our set of pointers as follows. In the first round, we mark all the elements of \( D \) and use the pointers to mark each occurrence of an element of \( D \) in a list \( \text{Out}(\beta) \). In the second round, we mark each \( \beta \) which is not already marked and where either \( \beta \in V_R - H \) and some element \( \gamma \in \text{Out}(\beta) \) is marked or \( \beta \in V_B - H \) and all \( \gamma \in \text{Out}(\beta) \) are marked. Again we use the pointers to mark all occurrences of a newly marked element \( \beta \) in some list \( \text{Out}(\gamma) \). In the third round and later rounds, we repeat the process of the second round. That is, we mark any previously unmarked \( \beta \in V_R - H \) such that some element of \( \text{Out}(\beta) \) is marked and any previously unmarked \( \beta \in V_B - H \) such that all elements of \( \text{Out}(\beta) \) are marked. We continue this way until we reach a round where no new marked elements are produced.

To help us determine, when an element \( \beta \) should be marked, we associate a counter to each \( \beta \). If \( \beta \in V_R - H - D \), then this counter is initially set equal to 0 and is incremented by 1 whenever an element in \( \text{Out}(\beta) \) is marked. \( \beta \) is then marked whenever the counter exceeds 0. If \( \beta \in V_B - H - D \), then the counter is initially set equal to \( |\text{Out}(\beta)| \) and is decremented by 1 each time a \( \gamma \in \text{Out}(\beta) \) is marked. Such a \( \beta \) becomes marked if the counter becomes 0. It now follows that to determine whether \( \beta \) is marked costs at most \( c_1 \) steps for some fixed constant \( c_1 \), for each \( \gamma \in \text{Out}(\beta) \) or equivalently \( c_1 \) steps for each edge out of \( \beta \). To update the marking of \( \beta \) on the lists \( \text{Out}(\delta) \) once \( \beta \) has been marked costs us \( c_2 \) steps, for some fixed constant \( c_2 \), for each \( \gamma \) such that

Fig. 2. Nodes in \( \gamma_H(D) \).
\( \beta \in \text{Out}(\gamma) \) or equivalently \( c_2 \) steps for each edge into \( \beta \). Thus if \( c_3 = \max(c_1, c_2) \), we require at most \( c_3 \) steps for each edge in or out of \( \beta \) for each \( \beta \in V \). Thus the number of steps required to find \( N = X_{D,H,R} \) is \( \leq c_3 \sum_{\beta \in V} |\text{Out}(\beta)| + |\text{In}(\beta)| = 2c_3|E| \).

It should be clear that in \( n \)th round of our marking procedure, the newly marked elements are just the element of \( \gamma^{(n-1)}_H(D) - \gamma^{(n-2)}_H(D) \). When a \( v \in V_R \) is marked at round \( n \), it is because there is some element \( w \in V_B \cap \text{Out}(v) \) has been marked at round \( n-1 \). Thus \( \gamma \in \gamma^{(n-2)}_H(D) \). Our strategy \( f \) is to define \( f(v) = w \), where \( w \) is the least element on \( \text{Out}(v) \) that was marked at stage \( n - 1 \). It then easily follows that to determine \( f = f_{D,H,R} \) on \( N \cap V_R \) requires at most \( c_4|E| \) steps for each \( \beta \in N \cap V_R \).

Now if \( v \in V_R - N \), then we can define \( f(v) \) arbitrarily to be the least \( w \) in \( \text{Out}(v) \). Thus it requires at most \( c_5V_c\sum_{\beta \in V}|\text{Out}(\beta)| = c_4|E| \) steps to construct the no memory strategy \( f \). Thus \( f \) and \( N \) can be constructed in \( c_5|E| \) steps where \( c = 2 \max(c_3, c_4) \).

We pause that this point to make an important remark about the points in \( V - X_{D,H,P} \) based on the construction of Lemma 4.8. Let \( \overline{P} \) denote the opposite player to \( P \) and suppose that \( v \in V - X_{D,H,P} \). Let \( V_P = V_R \) if \( P \) is Red and \( V_P = V_B \) if \( P \) is Black and let \( V_{\overline{P}} = V - V_P \). Then

(a) if \( v \in V_{\overline{P}} - H \), then there is no edge from \( v \) into \( X_{D,H,P} \)

(b) if \( v \in V_{\overline{P}} - H \), there is at least one edge from \( v \) into \( V - X_{D,H,P} \), and

(c) if \( v \in H \), then we will have no condition on edges out of \( v \) and thus a node \( v \in H \cap V_{\overline{P}} \) may have an edge from \( v \) into \( X_{D,H,P} \) and a node \( v \in H \cap V_{\overline{P}} \) may have \( \text{Out}(v) \subseteq X_{D,H,P} \).

Consider the no memory strategy \( g_{D,H,P} \) for \( \overline{P} \) where for \( \eta \in V_P \),

\[
g_{D,H,P}(\eta) = \begin{cases} 
\beta & \text{where } \beta \text{ is the first element of } \text{Out}(\eta) \\
\gamma & \text{where } \gamma \text{ is the first element of } \text{Out}(\eta) \\
\text{such that } \beta \in V - X_{D,H,P} \text{ if } \eta \notin (X_{D,H,P} \cup H), \\
\text{if } \eta \in X_{D,H,P} \cup H.
\end{cases}
\]

First it is easy to see that given \( D, H, \) and \( X_{D,H,P} \), there is a fixed constant \( c \) such that it requires at most \( c|\text{Out}(\eta)| \) steps to compute \( g_{D,H,P}(\eta) \) for each \( \eta \in V_P \). Thus it takes at most \( \sum_{\beta \in V_P}|\text{Out}(\beta)| \leq c|E| \) steps to construct \( g_{D,H,P} \). Moreover suppose that \( p = (\eta_0, \eta_1, \ldots) \) is a play in which \( \overline{P} \) is following \( g_{D,H,P} \) such that \( \eta_0 \notin X_{D,H,P} \). Then either the play will continue forever in \( V - X_{D,H,P} \) or there must be some \( \eta_i \) such that \( \eta_i \in H \) and \( \eta_j \notin X_{D,H,P} \) for all \( j < i \). That is, by following \( g_{D,H,P} \), \( \overline{P} \) forces the play to visit \( H \) before it can visit \( X_{D,H,P} \). Note that in particular if \( H = \emptyset \), then by following \( g_{D,H,P} \), \( \overline{P} \) can force the play to stay entirely within \( V - X_{D,H,P} \).

Next we need to define the concept of a subgame of a McNaughton game \( G_M \) determined by a subset \( A \subseteq V \) where \( G = (V, V_B, V_R, E) \). For \( A \subseteq V \), let \( G_A = (A, V_B(A), V_R(A), E_A) \) be the directed bipartite graph obtained by restricting \( G \) to \( A \), i.e. \( V_B(A) = V_B \cap A \), \( V_R(A) = V_R \cap A \) and \( E_A = \{(x,y) \in E : x,y \in A \} \).

**Definition 4.9.** If \( A \subseteq V \) and \( G_A \) has the property that for each \( \eta \in V_B(A) \), there is a \( \gamma \in V_R(A) \) such that \( (\eta, \gamma) \in E_A \), and for each \( \gamma \in V_R(A) \), there is a \( \delta \in V_B(A) \)
such that \((\gamma, \delta) \in E_A\), then we say \(A\) determines the subgame \(G_M(A)\) of \(G_M\) where 
\[ G_M(A) = (G_A, S_A, \Omega_A), \quad S_A = S \cap A, \quad \text{and} \quad \Omega_A = \{X \in \Omega : X \subseteq S_A\}. \]

**Example 4.1.** Again consider the set \(X_{LD,0,P}\) constructed in Lemma 4.8. It is easy to see that the conditions (a) and (b) for \(X_{LD,0,P}\) ensure that \(V - X_{LD,0,P}\) determines a subgame of \(G_M\). Because \(P\) has no move out of \(V - X_{LD,0,P}\), the subgame \(G_M(V - X_{LD,0,P})\) has the following property.

\((*)\) Suppose \(f\) is a \(LVR\) winning strategy for \(\overline{P}\) from \(A \subseteq V - X_{LD,0,P}\) in the subgame \(G_M(V - X_{LD,0,P})\) and \(\overline{f}\) is any \(LVR\) strategy for \(\overline{P}\) in \(G_M\) which agrees with \(f\) on all positions \((L, \eta)\) which occur in \(G_M(V - X_{LD,0,P})\). Then \(\overline{f}\) is also a \(LVR\) winning strategy for \(\overline{P}\) from \(A\) in \(G_M\).

That is, since player \(P\) has no move to force the play out of \(V - X_{LD,0,P}\) from a node in \(V - X_{LD,0,P}\), it is easy to prove by induction on \(i\) that any partial play \(p' = (\eta_0, \eta_1, \ldots, \eta_i)\) such that \(\eta_0 \in A\) and \(\overline{P}\) follows \(\overline{f}\) in \(p'\) will lie entirely in \(V - X_{LD,0,P}\). Hence \(p'\) will be a play in \(G_M(V - X_{LD,0,P})\) in which \(\overline{P}\) is following \(f\). Thus if \(p' = (\eta_0, \eta_1, \ldots)\) is a play in which \(\overline{P}\) is following \(\overline{f}\) and which starts in \(A\), then \(p'\) is just a play in \(G_M(V - X_{LD,0,P})\) in which \(\overline{P}\) is following \(f\). Thus for such a play \(p'\), if \(P = \text{Black}\), the \(\text{perm}(p') \in \Omega(V - X_{LD,0,P}) \subseteq \Omega\) and if \(\overline{P} = \text{Red}\), then \(\text{perm}(p') \notin \Omega(V - X_{LD,0,P})\) and hence \(\text{perm}(p') \notin \Omega\). Thus in either case, \(\overline{P}\) wins \(p'\).

A similar remark applies to no memory strategies \(f\) for \(\overline{P}\) in \(G_M(V - X_{LD,0,P})\). That is, we have the following property.

\((**)\) Suppose \(f\) is a no memory winning strategy for \(\overline{P}\) from \(A \subseteq V - X_{LD,0,P}\) in the subgame \(G_M(V - X_{LD,0,P})\) and \(\overline{f}\) is any no memory strategy for \(\overline{P}\) in \(G_M\) which agrees with \(f\) for all \(\eta \in V - X_{LD,0,P}\). Then \(\overline{f}\) is a winning no memory strategy for \(\overline{P}\) from \(A\) in \(G_M\).

As stated at the beginning of our proof of Theorem 4.5, we shall basically follow the outline of McNaughton's algorithm to prove Theorem 4.1. Our contribution is to add in the extra steps needed to actually compute the \(LVR\) strategies \(f_b\) and \(f_r\) described above and to tighten some of the complexity bounds in certain steps in McNaughton's algorithm.

Our algorithm proceeds by recursion first on \(|S|\) and then on \(|E|\). McNaughton proves Theorem 4.1 directly for any game \(G_M = (G, S, \Omega)\), where \(|S| = 1\). This case will also form the base case for our recursion.

**Theorem 4.10.** Let \(G_M = (G, S, \Omega)\) be a McNaughton game, where \(G = (V, V_B, V_R, E)\) and \(S = \{s\}\) is a singleton. Then given \(G_M\), we can find \(\text{Win}_B(G_M), \text{Win}_R(G_M), \) a no memory winning strategy \(f_b\) for \(\text{Black}\) from \(\text{Win}_B(G_M)\), and a no memory winning strategy \(f_r\) for \(\text{Red}\) from \(\text{Win}_R(G_M)\) in \(O(|E|)\) steps.

**Proof.** There are 4 cases depending on \(\Omega\).
Case 1: \( \Omega = \{ \emptyset, \{s\} \} \).

In this case, Black wins every possible play. Thus \( \text{Win}_B(\mathcal{G}_M) = V \) and \( \text{Win}_R(\mathcal{G}_M) = \emptyset \). Moreover, the no memory strategies \( f_b \) and \( f_r \) can be anything. Thus we define for \( V \in \mathcal{V}_B \), \( f_b(v) = w \) where \( w \) is the first vertex on \( \text{Out}(v) \) and for each \( x \in \mathcal{V}_R \), \( f_r(x) = y \) where \( y \) is the first vertex on \( \text{Out}(x) \).

Case 2: \( \Omega = \{ \{s\} \} \).

Apply Lemma 4.8 and construct \( X_B = X_{\{s\},0,B} \), its corresponding no memory strategy \( f = f_{\{s\},0,B} \) for Black, and its corresponding no memory strategy \( g = g_{\{s\},0,B} \) for Red. By Lemma 4.8 and the remarks following Lemma 4.8, we can construct \( X_B, f, \) and \( g \) in \( \mathcal{O}(|E|) \) steps. There are 4 subcases.

Subcase 2.1: \( s \in \mathcal{V}_B \) and there exists an edge \( (s, \gamma) \in E \) with \( \gamma \in X_B \).

In this case, we claim \( \text{Win}_B(\mathcal{G}_M) = X_B \) and \( \text{Win}_R(\mathcal{G}_M) = V - X_B \). That is, consider the no memory strategy \( f_b \) where for \( \eta \in \mathcal{V}_B \)

\[
f_b(\eta) = \begin{cases} f(\eta) & \text{if } \eta \neq s, \\ \gamma_0 & \text{if } \gamma_0 \text{ is the least } \gamma \in \text{Out}(s) \text{ such that } \gamma \in X_B, \text{ if } \eta = s. \end{cases}
\]

Then clearly any play \( p = (\eta_0, \eta_1, \ldots) \) which starts in \( X_B \) will visit \( s \) infinitely many times. That is, when Black follows \( f_b \), he will force a visit to \( s \) since \( f_b(\eta) = f_{\{s\},0,B}(\eta) \) for \( \eta \neq s \). Once we visit \( s \), our next move will be to visit \( \gamma_0 \) and once again Black can force another visit to \( s \) by following \( f_b \). Thus \( X_B \subseteq \text{Win}_B(\mathcal{G}_M) \).

By our remarks following Lemma 4.8, \( g = g_{\{s\},0,B} \) is a no memory strategy for Red such that by following \( g \), Red forces any play \( p \) which starts in \( V - X_B \) to stay in \( V - X_B \) and hence Red forces \( \text{perm}(p) = \emptyset \). Thus, \( f_r = g \) is a no memory winning strategy for Red from \( V - X_B \). Thus \( V - X_B \subseteq \text{Win}_R(\mathcal{G}_M) \).

It then follows that since \( \text{Win}_B(\mathcal{G}_M) \cap \text{Win}_R(\mathcal{G}_M) = \emptyset \), \( \text{Win}_B(\mathcal{G}_M) = X_B \) and \( \text{Win}_R(\mathcal{G}_M) = V - X_B \).

Subcase 2.2: \( s \in \mathcal{V}_R \) and \( \text{Out}(s) \subseteq X_B \).

By essentially the same analysis as in subcase 2.1, we can show that \( \text{Win}_B(\mathcal{G}_M) = X_B \) with \( f_b = f_{\{s\},0,B} \) and \( \text{Win}_R(\mathcal{G}_M) = V - X_B \) with \( f_r = g \).

Subcase 2.3: \( s \in \mathcal{V}_B \) and \( \text{Out}(s) \cap X_B = \emptyset \).

Let \( f_r \) be defined in subcase 2.1. Then it is easy to see that any play \( p \) in which Red follows \( f_r \) will either never visit \( s \) or will visit \( s \) exactly once since the next move after a visit to \( s \) must be to a node in \( V - X_B \) after which the play must stay forever in \( V - X_B \). Thus \( \text{perm}(p) = \emptyset \) for all plays where Red follows \( f_r \). Thus \( f_r \) is a no memory winning strategy for Red from \( V \). Hence \( \text{Win}_B(\mathcal{G}_M) = \emptyset \) and \( \text{Win}_R(f_r) = V \).

Subcase 2.4: \( s \in \mathcal{V}_R \) and \( \text{Out}(s) \cap (V - X_B) \neq \emptyset \).

Let \( f_r \) be defined so that \( f_r(\eta) = g_{\{s\},0,B}(\eta) \) if \( \eta \neq s \) and \( f_r(s) = \gamma \) where \( \gamma \) is the least \( \delta \) in \( \text{Out}(s) \cap (V - X_B) \) otherwise. Then just like in subcase 2.3, we can show that any play \( p \) in which Red is following \( f_r \) can visit \( s \) at most once. It follows that \( \text{Win}_B(\mathcal{G}_M) = \emptyset \), \( \text{Win}_R(\mathcal{G}_M) = V \), and \( f_r \) is a no memory winning strategy for Red from \( V \).
Case 3: \( \Omega = \emptyset \).

In this case every play must be win for Red. Thus \( \text{Win}_B(\mathcal{G}_M) = \emptyset \) and \( \text{Win}_R(\mathcal{G}_M) = V \). As defined in Case 1 will be a no memory strategy for Red from \( V \).

Case 4: \( \Omega = \{\emptyset\} \).

The analysis in this case is exactly the same as in Case 2 except that one reverses the role of Red and Black.

It is now easy to see that since \(|E| > |V|\), it takes at most \( O(|E|) \) steps to define the functions \( f_b \) and \( f_r \) from \( \mathcal{X}_B \) in cases 1 and 3. In cases 2 and 4, note that we can compute \( \mathcal{X}_B \) and \( \mathcal{X}_R \) and their corresponding no memory strategies steps by Lemma 4.8. It easily follows that we can compute \( \text{Win}_B(\mathcal{G}_M), \text{Win}_R(\mathcal{G}_M), f_b \) and \( f_r \) in \( O(|E|) \) steps in cases 2 and 4.

Next we shall outline the recursive algorithm that McNaughton used to prove Theorem 4.1. This outline will be followed by an analysis of the complexity of each of the steps in the algorithm. Let \( \mathcal{G}_M = (G, S, \Omega) \) be a McNaughton game where \( G = (V, V_B, V_R, E) \) and \( S = \{s_1, \ldots, s_k\} \). Note that since there is a basic symmetry between the roles of Black and Red in a McNaughton game, there is no loss in generality in assuming \( S \subseteq \Omega \).

Step 1: Construct the sets \( N_{i,B} \) and \( N_{i,R} \) for \( i = 1, \ldots, k \).

Here \( N_{i,B} \) is the set of nodes \( \eta \) such that Black has a LVR strategy \( f \) such that in any play \((\eta_0 = \eta, \eta_1, \ldots)\) where Black follows \( f \), Black wins and \( s_i \) does not appear in \((\eta_0, \eta_1, \ldots)\). Thus \( \eta \in N_{i,B} \) iff Black has a LVR winning strategy from \( \{\eta\} \) which avoids \( s_i \). Similarly, \( N_{i,R} \) is the set of nodes \( \eta \) such that Red has a LVR strategy \( g \) such that in any play \((\eta_0 = \eta, \eta_1, \ldots)\) where Red follows \( g \), Red wins and \( s_i \) does not appear in \((\eta_0, \eta_1, \ldots)\).

Step 2: Construct the sets \( N_B \) and \( N_R \).

Here \( N_B \) is the set of nodes \( \eta \) such that Black has a no memory strategy which can force any play which starts at \( \eta \) into \( \bigcup_{i=1}^{k} N_{i,B} \) in zero or more moves. Hence \( N_B \) is a set of nodes from which Black can win. Similarly \( N_R \) is the set of nodes \( \eta \) such that Red has no no memory strategy which can force any play which starts at \( \eta \) into \( \bigcup_{i=1}^{k} N_{i,R} \) in 0 or more moves. Again \( N_R \) represents a set of nodes from which Red can win.

Step 3: Step 3 has 3 cases depending on the sets \( N_B \) and \( N_R \).

Case 3.1: \( N_B \cup N_R = V \).

In this case, we are done, i.e., \( \text{Win}_B(\mathcal{G}_M) = N_B \) and \( \text{Win}_R(\mathcal{G}_M) = N_R \).

Case 3.2: \( N_B = N_R = \emptyset \).

In this case, \( \text{Win}_B(\mathcal{G}_M) = V \) and we can directly construct an LVR winning strategy for Black from \( V \).

Case 3.3: Not Case 1 or 2.

In this case, we show that \( V - (N_B \cup N_R) \) determines a subgame of \( \mathcal{G}_M \) and we can solve the subgame \( \mathcal{G}_M(V - (N_B \cup N_R)) \) by recursion.

We now consider the complexity of each of these steps.
The complexity of Step 1

To construct the sets $N_{i,B}$ and $N_{i,R}$, we proceed as follows. For $i = 1, \ldots, k$, construct $X_{i,B} = X_{\{s_i\},0,B}$ and $X_{i,R} = X_{\{s_i\},0,R}$ as in Lemma 4.8. By Lemma 4.8, the sets $X_{i,B}$ and $X_{i,R}$ can each be constructed in $c_1|E|$ steps for some fixed constant $c_1$. The importance of the sets $X_{i,B}$ and $X_{i,R}$ is due to the following result of McNaughton.

**Lemma 4.11.** For each $i$, $V - X_{i,B}$ and $V - X_{i,R}$ determine subgames of $G_M$. Moreover,

1. $N_{i,B} = \text{Win}_B(G_M(V - X_{i,R}))$ and
2. $N_{i,R} = \text{Win}_R(G_M(V - X_{i,B}))$.

Let $F(k,n)$ denote the number of steps it takes to find the sets $\text{Win}_B(G_M^*)$ and $\text{Win}_R(G_M^*)$ in a McNaughton game $G_M^* = (S^*, V^*, E^*)$ where $k = |S^*|$, $n = |E^*|$, and $G^* = (V^*, V_B^*, V_R^*, E^*)$. Similarly let $H(k,n)$ denote the number of steps it takes to find the sets $\text{Win}_B(G_M^*)$, $\text{Win}_R(G_M^*)$, a LVR winning strategy $f^*_i$ for Black from $\text{Win}_B(G_M^*)$, and a LVR winning strategy $g^*_i$ for Red from $\text{Win}_R(G_M^*)$.

We have specified that $|S| = k$. Suppose that $|E| = n$. Now for each $i$, $s_i \in X_{i,B}$ and $s_i \in X_{i,R}$, so that in the subgames determined by $V - X_{i,B}$ and $V - X_{i,R}$, the size of both the set of significant nodes and the set of edges has decreased. Thus for each $i$, it takes at most $2F(k - 1, n - 1)$ steps to find $N_{i,B} = \text{Win}_B(G_M(V - X_{i,R}))$ and to find $N_{i,R} = \text{Win}_R(G_M(V - X_{i,B}))$. Hence to find the sets $N_{i,B}$ and $N_{i,R}$ for $i = 1, \ldots, k$ takes at most

$$2kc_1|E| + 2kF(k - 1, n - 1) = 2c_1|S||E| + 2|S|F(|S| - 1, |E| - 1)$$

steps.

We may assume that it takes $2kH(k - 1, n - 1)$ steps to compute the set $N_{i,B} = \text{Win}_B(G_M(V - X_{i,R}))$ and its corresponding winning LVR strategy $f_i$ for Black from $N_{i,B}$ and the set $N_{i,R} = \text{Win}_R(G_M(V - X_{i,B}))$ and its corresponding winning LVR strategy $g_i$ for Red from $N_{i,B}$ for $i = 1, \ldots, k$. By our remarks in Example 2.1, if we define for $(L, \delta) \in L \times V_B$,

$$\bar{f}_i(L, \delta) = \begin{cases} f_i(L, \delta) & \text{if } (L, \delta) \text{ is a position in } G_M(V - X_{i,R}), \\ \delta & \text{where } \delta \text{ is the least element in } \text{Out}(\delta), \text{otherwise} \end{cases}$$

and for $(L, \delta) \in L \times V_R$,

$$\bar{g}_i(L, \delta) = \begin{cases} g_i(L, \delta) & \text{if } (L, \delta) \text{ is a position in } G_M(V - X_{i,B}), \\ \delta & \text{where } \delta \text{ is the least element in } \text{Out}(\delta), \text{otherwise} \end{cases}$$

then $\bar{f}_i$ is a winning LVR strategy for Black from $\text{Win}_B(G_M(V - X_{i,R}))$ in $G_M$ and $\bar{g}_i$ is a winning LVR strategy for Red from $\text{Win}_R(G_M(V - X_{i,B}))$ in $G_M$.

Now it takes at most $c_2|V||S|$ to define $\bar{f}_i$ from $f_i$ or to define $\bar{g}_i$ from $g_i$ for some fixed constant $c_2$. Thus it takes $2kH(k - 1, n - 1) + 2kc_2|V||S|$ steps to produce $\bar{f}_1, \ldots, \bar{f}_k$, $\bar{g}_1, \ldots, \bar{g}_k$.

Next we can apply the construction of Theorem 4.4 to $\bar{f}_1, \ldots, \bar{f}_k$ to compute a single LVR winning strategy $\bar{f}$ for Black from $\bigcup_{i=1}^k N_{i,B}$ in $O(|S||E||S|)$ steps. Similarly we
can compute a single LVR winning strategy for Red from $\bigcup_{i=1}^{k} N_{i,R}$ using $\overline{g}_1, \ldots, \overline{g}_k$ in $O(\max\{c_1, c_2\})$ steps. It follows that there is some fixed constant $c_3 \geq 2 \max(c_1, c_2)$ such that we can compute $\bigcup_{i=1}^{k} N_{i,B}, \bigcup_{i=1}^{k} N_{i,R}, \overline{f}$, and $\overline{g}$ in at most
\[
2c_3k|E| + 2kH(k - 1, n - 1) + 2kc_3|V||S|! + 2k|E||S|!
\leq 4c_3|S||E||S|! + 2|S|H(|S| - 1, |E| - 1)
\]
(8)
steps.

We make one more remark about the sets $N_{i,B}$ and $N_{i,R}$. Since $N_{i,B} = W_{in_{B}}(G_M(V - X_{i,R}))$, it must be the case that for any play $p = (\eta_0, \eta_1, \ldots)$ in which Black follows $\overline{f}$, where $\eta_0 \in N_{i,B}$, all the nodes $\eta_i$ for $i \geq 1$ must be in $N_{i,B}$. This is because by Lemma 4.7, there is an LVR winning strategy for Black from $\{\eta_i\}$ and hence $\eta_i \in W_{in_{B}}(G_M(V - X_{i,R}))$. Thus it follows that if $\eta \in \bigcup_{i=1}^{k} N_{i,B}$, then any play $p$ in which Black follows $\overline{f}$, where $p$ starts at $\eta$, will remain entirely in $\bigcup_{i=1}^{k} N_{i,B}$. Thus if $\delta \notin \bigcup_{i=1}^{k} N_{i,B}$ or $L$ contains an element which is not in $\bigcup_{i=1}^{k} N_{i,R}$ then $(L, \delta)$ is not $\overline{f}$-reachable for Black from $\bigcup_{i=1}^{k} N_{i,B}$ for any $i \in \mathcal{P}$. A similar remark holds for $\overline{g}$ and $\bigcup_{i=1}^{k} N_{i,R}$.

The complexity of Step 2

Again we can apply Lemma 4.8 to construct the sets $N_B$ and $N_R$. Thus we can find $N_B$ and $N_R$ in less than or equal to $2c_1|E|$ steps.

Note that in $2c_1|E|$ steps, we can also construct no memory strategies $f$ and $g$ such that any play $p$, in which Black follows $f$ and which starts in $N_B - \bigcup_{i=1}^{k} N_{i,B}$, will visit $\bigcup_{i=1}^{k} N_{i,B}$ in 1 or more moves and any play $p'$, in which Red follows $g$ and which starts in $N_R - \bigcup_{i=1}^{k} N_{i,R}$, will visit $\bigcup_{i=1}^{k} N_{i,R}$ in 1 or more moves. Next we shall describe how we can use $f$ and $\overline{f}$ from step 1 to construct a LVR winning strategy $f_{N_{B}}$ for Black from $N_{B}$. The idea to construct $f_{N_{B}}$ is quite simple. That is, if we start at $\eta \in N_{B} - \bigcup_{i=1}^{k} N_{i,B}$ and follows $f$, then we will generate a partial play $(\eta_0, \eta_1, \ldots, \eta_s)$, where $\eta_s \in \bigcup_{i=1}^{k} N_{i,B}$ and $\eta_i \in N_{B} - \bigcup_{i=1}^{k} N_{i,B}$ for $i < s$. Then we will want to essentially follow $\overline{f}$. There is one catch however. If $(LVR(\eta_0, \ldots, \eta_s), \eta_s)$ is $\overline{f}$-reachable from $\bigcup_{i=1}^{k} N_{i,B}$, then Black can literally just follow $\overline{f}$ once we reach $\eta_s$. However, it may be the case that $L = LVR(\eta_0, \ldots, \eta_{s-1}) \neq \emptyset$ so that $(LVR(\eta_0, \ldots, \eta_s), \eta_s)$ is not $\overline{f}$-reachable from $\bigcup_{i=1}^{k} N_{i,B}$. Then Black cannot just follow $\overline{f}$. What Black needs to do in any continuation of the play $(\eta_0, \ldots, \eta_s, \eta_{s+1}, \eta_{s+2}, \ldots)$ is to act as if the play started at $\eta_s$. That is, if $k \geq s$ and $\eta_k \in V_B$, then Black’s next move should be $\overline{f}(LVR(\eta_s, \ldots, \eta_k), \eta_k)$. Thus we would like to define $f_{N_{B}}(LVR(\eta_0, \ldots, \eta_k), \eta_k)) = \overline{f}(LVR(\eta_s, \ldots, \eta_k), \eta_k)$. Fortunately in this case there is no conflict. That is, let $LVR(\eta_0, \ldots, \eta_{s-1}) = L'$. Note that the partial play $\eta_0, \ldots, \eta_{s-1}$ is entirely in $N_B - \bigcup_{i=1}^{k} N_{i,B}$ and any partial play $(\eta_s, \ldots, \eta_k)$ in which Black follows $\overline{f}$ stays entirely in $\bigcup_{i=1}^{k} N_{i,B}$ by our remarks at the end of step 1. Thus $L = LVR(\eta_0, \ldots, \eta_k)$ and $L'$ have no nodes in common if Black was following $\overline{f}$ in $(\eta_s, \ldots, \eta_k)$. Hence in such a situation, $LVR(\eta_0, \ldots, \eta_{s-1}, \eta_s, \ldots, \eta_k) = L'L$ but then $L$ consists of elements entirely in $N_B = \bigcup_{i=1}^{k} N_{i,B}$ so that again by our remarks at the end of step 1, $(L'L, \eta_k)$ is not $\overline{f}$-reachable for Black from $\bigcup_{i=1}^{k} N_{i,B}$. Thus there is no harm to the condition that $\overline{f}$ is a LVR winning strategy for Black from $\bigcup_{i=1}^{k} N_{i,B}$ by
changing the value of $\overline{f}$ at $(L'L, \eta_k)$. It follows that our desired LVR winning strategy for Black from $N_B$ can be defined as follows. For $L \in \mathcal{L}$, write $L \subseteq A$ iff each element of $L$ is in $A$. Then for $(L, \eta) \in \mathcal{L} \times V_B$, set

$$f_{N_B}(L, \eta) = \begin{cases} \overline{f}(L', \eta) & \text{if } L = L'L'' \text{ where } L' \subseteq N_B - \bigcup_{i=1}^{k} N_{i,B}, \\ \text{otherwise.} & L'' \subseteq \bigcup_{i=1}^{k} N_{i,B}, \text{ and } \eta \in \bigcup_{i=1}^{k} N_{i,B}. \end{cases}$$

Just as was the case with $\overline{f}$ and $\bigcup_{i=1}^{k} N_{B,i}$, any play $p$, in which Black follows $f_{N_B}$ and which starts in $N_B$, will stay entirely within $N_B$. Thus if either $\delta \notin N_B$ or $L \not\subseteq N_B$, then $(L, \delta)$ is not $f_{N_B}$-reachable for Black from $N_B$.

By the same argument, we can construct a LVR winning strategy $g_{N_R}$ for Red from $N_R$ from $g$ and $\overline{g}$. That is, for $(L, \eta) \in \mathcal{L} \times V_R$, set

$$g_{N_R}(L, \eta) = \begin{cases} g(\eta) & \text{if } \eta \in N_R - \bigcup_{i=1}^{k} N_{i,R}, \\ \overline{g}(L'', \eta) & \text{if } L = L'L'' \text{ where } L' \subseteq N_R - \bigcup_{i=1}^{k} N_{i,R}, \\ \text{otherwise.} & L'' \subseteq \bigcup_{i=1}^{k} N_{i,R}, \text{ and } \eta \in \bigcup_{i=1}^{k} N_{i,R}. \end{cases}$$

Then $g_{N_R}$ will be a LVR winning strategy for Red from $N_R$. Moreover any play, in which Red follows $g_{N_R}$ and which starts in $N_R$, will stay entirely within $N_R$. Thus if either $\delta \notin N_R$ or $L \not\subseteq N_R$, then $(L, \delta)$ is not $g_{N_R}$-reachable for Red from $N_R$.

Note that given $\overline{f}$ and $\bigcup_{i=1}^{k} N_{B,i}$, to find $f_{N_B}(L, \eta)$, we must check whether $\eta \in N_B$ or $\bigcup_{i=1}^{k} N_{i,B}$ and scan $L$ to see if it is of the form $L'L''$, where $L' \subseteq N_B - \bigcup_{i=1}^{k} N_{i,B}$ and $L'' \subseteq \bigcup_{i=1}^{k} N_{i,B}$. Thus it takes at most $c_4|S|$ steps for some fixed constant $c_4$ to find $f(L, \eta)$. Hence it takes at most $c_4S/2|V||S|$ additional steps to construct $f_{N_B}$. Similarly, it takes almost $c_4S/2|V||S||S|$ additional steps to construct $g_{N_R}$. Thus, the total number of steps to construct $N_B$ and $N_R$ in Step 2 is just

$$2c_1|E|. \quad (9)$$

The total number of steps to construct $N_B, N_R, f_{N_B},$ and $g_{N_R}$ in step 2 is

$$2c_1|E| + 5c_4S|V||S||S|. \quad (10)$$

The complexity of Step 3

In case 3.1, there is nothing left to do. That is, $Win_B(\mathcal{G}_{M}) = N_B$, $Win_R(\mathcal{G}_{M}) = N_R$, $f_b = f_{N_B}$, and $f_r = g_{N_R}$.

In case 3.2 where $N_B = N_R = \emptyset$, McNaughton showed that $Win_B(\mathcal{G}_{M}) = V$. The LVR winning strategy for Black can be constructed as follows. Consider the sets $X_{i,B}$ from step 1. In $O(|E|)$ steps, we can construct $X_{i,B}$ and a no memory strategy $h_i = f_{\{s_i\}, \emptyset, B}$ such that if Black follows $h_i$ in any play which starts in $X_{i,B}$, Black can force a visit to $s_i$ in 0 or more moves. By Lemma 4.11, $V - X_{i,B}$ determines a subgame of $\mathcal{G}_{M}$. Since $Win_R(\mathcal{G}_{M}(V - X_{i,B})) = N_{i,R} \subseteq N_R = \emptyset$, it follows that $Win_B(\mathcal{G}_{M}(V - X_{i,B})) = V - X_{i,B}$. Moreover in at most $H(k - 1, n - 1)$ steps, we can find a LVR winning strategy $t_i$ for
Black from $V - X_i, B$ in $G_M(V - X_i, B)$. We claim that the LVR winning strategy $f_b$ for Black from $V$ in $G_M$ can be defined using $h_1, t_1, \ldots, h_k, t_k$ as follows. First we want to modify the LVR strategies $t_1, \ldots, t_k$ slightly. Namely in a subgame $G_M(V - X_i, B)$, there are some positions $(L, \eta)$ which are not $t_i$-reachable from $V - X_i, B$ on which $t_i(L, \eta)$ can be arbitrary. Let us define $\text{reach}_i(L, \eta)$ as the longest cofinal subsequence $L'$ of $L$ such that $(L', \eta)$ is $t_i$-reachable from $V - X_i, B$. Note that by Lemma 4.3, we can construct the set of all $(L'', \beta)$ which are $t_i$-reachable from $V - X_i, B$ in the subgame $G_M(V - X_i, B)$ in at most $O((E - 1)(S - 1)!)$ steps. Having constructed the set of positions which are $t_i$-reachable from $V - X_i, B$ in the subgame $G_M(V - X_i, B)$, we then only have to ask $|L| + 1$ membership questions to find $\text{reach}_i(L, \eta)$. Thus for each $\eta \in V - X_i, B$, it take $d(1 + 2(|S| - 1) + 3(|S| - 1)(|S| - 2) + \cdots + |S|(|S| - 1)!)$ steps for some fixed constant $d$ to find the values of $\text{reach}_i(L, \eta)$ for $L \in \mathcal{L}$. Hence, we can construct the function $\text{reach}_i$ in

$$d_1(((E - 1)(|S| - 1)!)) + |V - X_i, B|(|S|)!$$

(11)

steps for each $i$ for some fixed constant $d_1$. We then can define a new LVR winning strategy $\ell_i$ for Black from $V - X_i, B$ in $G_M(V - X_i, B)$ by letting

$$\ell_i(L, \eta) = t_i(\text{reach}_i(L, \eta), \eta).$$

The advantage of the strategies $\ell_i$ is that if $p = (\eta_0, \eta_1, \ldots)$ is any play in $G_M$ such that in some cofinal sequence $(\eta_k, \eta_{k+1}, \ldots)$, we follow the LVR strategy which maps $(LVR(\eta_0, \ldots, \eta_r), \eta_r)$ to $\ell_i(LVR(\eta_0, \ldots, \eta_r) \cap (V - X_i, B), \eta_r)$ for each $\eta_r \in V_B$, where $r \geq k$ and $LVR(\eta_0, \ldots, \eta_r) \cap (V - X_i, B)$ denotes the subsequence of $LVR(\eta_0, \ldots, \eta_r)$ formed by all elements which lie in $V - X_i, B$, then $p$ will have a perm set of some play in the subgame $G_M(V - X_i, B)$ and hence be a win for Black.

This given, we shall define a LVR winning strategy from $V$ for Black from $h_1, \ldots, h_k$ and $\ell_1, \ldots, \ell_k$ as follows. Suppose that $L = (s_{\sigma_1}, \ldots, s_{\sigma_q}) \in \mathcal{L}$. Define the target of $L$, $t(L)$, to be equal to $s_{\sigma_q}$ if $q = k$ or to be equal to $s_j$ where $j$ is the least $i$ such that $i \notin \{\sigma_1, \ldots, \sigma_q\}$ if $q < k$. Recall that we are assuming that $S \in \Omega$ so that Black wins any play $p$ in which the perm set of $p$ is $S$. Hence we would like to force a play to visit every node infinitely often. We can thus think of $t(L)$ as the next node we would like to visit. That is, if $k = q$, then when we have reached a position $(L, \eta)$ in a play, we must have visited all the nodes in $S - \{s_{\sigma_1}\}$ since we visited $s_{\sigma_1}$. If $q < k$, then when we reach $(L, \eta)$ in a play, we have not yet visited $t(L)$. The definition of $f_r(L, \eta)$ depends mainly on $t(L)$. Given an last visitation record $L \in \mathcal{L}$ and set $X \subseteq V$, we let $L/X$ denote the last visitation record which results from $L$ by removing all elements which lie in $X$ from $L$. Say $t(L) = s_i$, then

$$f_b(L, \eta) = \begin{cases} h_i(\eta) & \text{if } \eta \in X_i, B, \\ \ell_i(L/X_i, B, \eta) & \text{if } \eta \in V - X_i, B. \end{cases}$$

Suppose Black follows this strategy in a play $p = (\eta_0, \eta_1, \ldots)$ and assume that $LVR(\eta_0, \ldots, \eta_i, \eta_i) = (L, \eta)$ where $t(L) = s_i$. Now if $\eta \in X_i, B$, then by following $f_b$, Black will force a visit to $s_i$ in 0 or more moves. If $\eta \in V - X_i, B$, then Black is following the
strategy $\ell_i$ for some position which is reachable from a play which starts in $V - X_i,B = \text{Win}_B(\mathcal{G}_M(V - X_i,B))$ in the subgame $\mathcal{G}_M(V - X_i,B)$. Now either there will be some partial play $(\eta_i, \ldots, \eta_k)$, where $\eta_k \in V_R$ and Red’s next move puts $\eta_{k+1} \in X_i,B$ in which case Black will subsequently force the play to visit $s_i$ or Red will never move out of $V - X_i,B$ in which case Black automatically wins because the effect of following $f_b$ will be to follow $\ell_i$ in some cofinal sequence of play which starts in $\text{Win}_B(\mathcal{G}_M(V - X_i,B))$ in the subgame $\mathcal{G}_M(V - X_i,B)$. Thus in any play $p = (\eta_0, \eta_1, \ldots)$ in which Black follows $f_b$, either Black forces $\text{perm}(p) = S$ or for some $i$ and $j$, $(\eta_j, \eta_{j+1}, \ldots)$ is a cofinal sequence in a play in the subgame $\mathcal{G}_M(V - X_i,B)$ which starts from $\text{Win}_B(\mathcal{G}_M(V - X_i,B))$ and in which Black is following $\ell_i$ so that $\text{perm}(p) \in \Omega(V - X_i,B) \subseteq \Omega$. Thus $f_b$ is a LVR winning strategy for Black from $V$ in $\mathcal{G}_M$.

Note that we may assume that for $i = 1, \ldots, k$, the set $X_i,B$, the no memory strategy $h_i$ and the LVR winning strategy $t_i$ for Black from $V - X_i,B$ in $\mathcal{G}_M(V - X_i,B)$ was constructed in step 1. Given the $X_i,B$'s, $h_i$'s, and $t_i$'s, it then takes at most $|S|d_1(((|E| - 1)(|S| - 1)! + |V - X_i,B|(|S|!))$ steps to construct $\ell_1, \ldots, \ell_k$. Finally, given $X_i,B$'s, $h_i$'s, and $t_i$'s, it takes an additional $d_2|S|$ steps to find $f(L, \eta)$ for any $(L, \eta) \in \mathcal{L} \times V$ for some fixed constant $d_2$. Thus, note that since $|V| \leq |E|$, it takes at most

$$|S|(d_1(((|E| - 1)(|S| - 1)! + |V - X_i,B|(|S|!)) + d_2|S|) = c_5$$

additional steps to compute $f_b$ in this case for some fixed constant $c_5$.

If all we are interested in is computing $\text{Win}_B(\mathcal{G}_M)$ and $\text{Win}_R(\mathcal{G}_M)$, there is essentially nothing to do in this case. That is, once we know $N_R = N_B = \emptyset$, then $\text{Win}_B(\mathcal{G}_M) = V$ and $\text{Win}_R(\mathcal{G}_M) = \emptyset$.

Finally, we consider case 3.3 of step 3. First we claim $V - (N_B \cup N_R)$ determines a subgame of $\mathcal{G}_M$. That is, if $v \in V_R - (N_B \cup N_R)$, then there can be no edge $(v, w) \in E$ such that $w \in N_B$ since otherwise the construction of $N_B$ via Lemma 4.8 would force $v \in N_B$. Similarly, it cannot be that for all edges $(v, w) \in E$, $w \in N_R$ since otherwise $v$ would be forced into $N_R$. Thus there is at least one edge $(v, w) \in E$ such that $w \in V - (N_B \cup N_R)$. A similar argument will show that for every $v \in V_R - (N_B \cup N_R)$, there is an edge $(v, w) \in E$ such that $w \in V - (N_B \cup N_R)$. Then since $N_B \cup N_R \neq \emptyset$, we can by recursion construct the sets $\text{Win}_B(\mathcal{G}_M(V - (N_B \cup N_R)))$, $\text{Win}_R(\mathcal{G}_M(V - (N_B \cup N_R)))$, a LVR winning strategy $h_b$ for Black from $\text{Win}_B(\mathcal{G}_M(V - (N_B \cup N_R)))$, and a LVR winning strategy $h_r$ for Red from $\text{Win}_R(\mathcal{G}_M(V - (N_B \cup N_R)))$. Then we claim that $\text{Win}_B(\mathcal{G}_M) = N_B \cup \text{Win}_B(\mathcal{G}_M(V - (N_B \cup N_R)))$ and $\text{Win}_R(\mathcal{G}_M) = N_R \cup \text{Win}_R(\mathcal{G}_M(V - (N_B \cup N_R)))$. The winning LVR strategy $f_b$ for Black is defined as follows. For any $(L, \eta) \in \mathcal{L} \times V$, let

$$f_b(L, \eta) = \begin{cases} h_b(L, \eta) & \text{if } \eta \in V - (N_B \cup N_R) \\ f_{N_b}(L'', \eta) & \text{if } \eta \in N_B \text{ and } L = L' L''', \\ f_{N_b}(L, \eta) & \text{if otherwise.} \end{cases}$$
Now suppose that $p = (\eta_0, \eta_1, \ldots)$ is a play in which Black follows $f_b$. If $\eta_0 \in N_B$, then Black will just be following $f_{N_B}$. By our remarks at the end of step 2, the entire play will be in $N_R$ in this case and Black will win since $f_{N_B}$ is a LVR winning strategy for Black from $N_B$ in $G_M$. Next assume $\eta_0 \in Win_B(G_M(V - (N_B \cup N_R)))$. Then Black starts out following $h_b$. As long as Red continues to play in the subgame $G_M(V - (N_B \cup N_R))$, Black will continue to follow $h_b$ keeping the play in the subgame $G_M(V - (N_B \cup N_R))$. Now suppose that there is a $k$ such that $\eta_k \in V_R - (N_B \cup N_R)$ but $\eta_{k+1} \in N_B \cup N_R$. Note that it can not be that $\eta_{k+1} \in N_R$ since otherwise $\eta_k \in N_R$ by our construction of $N_R$. Thus $\eta_{k+1} \in N_B$. But then just like the situation in Case 3.2 of step 3, our definition of $f_b$ ensures that Black then plays like he is following $f_{N_B}$ in the play $(\eta_{k+1}, \eta_{k+2}, \ldots)$. Moreover, as Red has no move from a node in $N_B$ to a node out of $N_B$, the play will continue in $N_B$. Thus in this case, $perm(\eta_0, \eta_1, \ldots) = perm(\eta_{k+1}, \eta_{k+2}, \ldots) \in \Omega$ since $f_{N_B}$ is a winning LVR strategy for Black from $N_B$. Finally, if there is no such $k$, then the entire play $p$ occurs in the subgame $G_M(V - (N_B \cup N_R))$ and again Black must win $p$ because he is following $h_b$ which is a LVR winning strategy for Black from $Win_B(G_M(V - (N_B \cup N_R)))$ in $G_M(V - (N_B \cup N_R))$. Thus $f_b$ is a LVR winning strategy for Black from $N_B \cup Win_B(G_M(V - (N_B \cup N_R)))$. A similar argument will show that $f_r$, is a winning LVR strategy for Red from $N_R \cup Win_R(G_M(V - (N_B \cup N_R)))$ where for $(L, \eta) \in \mathcal{L} \times V$, we define

$$f_r(L, \eta) = \begin{cases} h_r(L, \eta) & \text{if } \eta \in V - (N_B \cup N_R) \text{ and } L \subseteq V - (N_B \cup N_R), \\ g_{N_R}(L'', \eta) & \text{if } \eta \in N_R \text{ and } L = L''L'', \text{ where } \\ & L' \subseteq V - (N_B \cup N_R) \text{ and } L'' \subseteq N_R, \\ g_{N_B}(L, \eta) & \text{otherwise.} \end{cases}$$

Finally, we consider the complexity in this case. There are two subcases.

**Subcase 3.3.1:** $(N_B \cup N_R) \cap S = \emptyset$.

In this subcase, McNaughton proves that in the subgame $G'_M = G_M(V - (N_B \cup N_R))$, the analogues of $N_B$ and $N_R$ are empty. That is, he proves the following.

**Theorem 4.12.** Suppose $G_M$ is a McNaughton game and $N_B(G_M) = N_B$ and $N_R(G_M) = N_R$ is defined as in step 2 of our recursive algorithm. If $(N_B \cup N_R) \cap S = \emptyset$, then in the subgame $G'_M = G_M(V - (N_B \cup N_R))$, $N_B(G'_M) \cup N_R(G'_M) = \emptyset$.

The import of Theorem 4.12 is that in subcase 3.3.1, the subgame $G'_M$ falls in case 3.2 of step 3 so that $Win_B(G'_M(V - (N_B \cup N_R))) = V - (N_B \cup N_R)$ and $Win_R(G'_M(V - (N_B \cup N_R))) = \emptyset$. Thus there is essentially nothing to do in this subcase to find $Win_B(G'_M)$ and $Win_R(G'_M)$ since we know that

$$Win_B(G_M) = V - N_R \text{ and } Win_R(G_M) = N_R.$$
and find their corresponding winning LVR strategies for Black. Note that since $N_B \cup N_R \neq \emptyset$ and $s_1 \in X_{i,B}(\mathcal{G}'_M)$, $|E(V - (N_B \cup N_R \cup X_{i,B}(\mathcal{G}'_M))|$ is less than or equal to $n - 2$. Thus it takes $2kc_1(|E| - 1) + 2kH(k - 1, n - 2)$ steps to find these sets and solve the subgames. Then the strategy $h_b$ can be constructed from no memory strategies for corresponding to $X_{i,B}(\mathcal{G}'_M)$'s and the winning LVR strategies for Black in the subgames $\mathcal{G}_M(V - (N_B \cup N_R \cup X_{i,B}(\mathcal{G}'_M)))$ in $c_5|S|(|E| - 1)|S|!$ steps. Finally, it takes $d_5|S||\mathcal{L} \times V|$ steps to construct $f_b$ from $h_b$ and $f_{N_R}$ for some fixed constant $d_5$. Clearly, we can let $f_r = f_{N_R}$ in this subcase. Thus to construct $f_r$ and $f_b$, requires at most

$$2kc_1|E| + 2kH(k - 1, n - 2) + c_5(k(n - 1)k!) + d_5k5/2n(k!)$$

steps for some fixed constant $c_6$.

**Subcase 3.3.2:** $(N_B \cup N_R) \cap S \neq \emptyset$.

In this case to find the sets $\text{Wins}(\mathcal{G}'_M(V - (N_B \cup N_R)))$ and $\text{Wins}(\mathcal{G}_M(V - (N_B \cup N_R)))$ requires at most $F(k - 1, n - 1)$ steps. Thus to find the sets $\text{Wins}(\mathcal{G}_M)$ and $\text{Wins}(\mathcal{G}_M)$ requires at most

$$2c_7|E| + F(|S| - 1, |E| - 1)$$

steps where the factor $2c_7|V| \leq 2c_7|E|$ is the cost of forming the unions $N_B \cup \text{Wins}\left(\mathcal{G}_M(V - (N_B \cup N_R))\right)$ and $N_R \cup \text{Wins}\left(\mathcal{G}_M(V - (N_B \cup N_R))\right)$.

The number of steps required to construct the LVR strategies $f_b$ and $f_r$ is $H(k - 1, n - 1)$ steps to find the strategies $h_b$ and $h_r$ in the subgames $\mathcal{G}_M(V - (N_B \cup N_R))$ plus $c(|S||\mathcal{L} \times V|)$ steps to construct the strategies $f_b$ from $f_{N_R}$ and $h_b$ and $f_r$ from $g_{N_R}$ and $h_r$. Hence in this case, to find $\text{Wins}(\mathcal{G}_M)$, $\text{Wins}(\mathcal{G}_M)$, $f_b$, and $f_r$ takes at most

$$2c_7|E| + H(|S| - 1, |E| - 1) + c_6|S||V||S|!$$

steps.

Putting together our estimates from (7)–(15) in the three steps of our recursive algorithm, we have shown that for some fixed constants $c$ and $c'$

$$F(|S|, |E|) \leq c|S||E| + c|E| + |V| + (2|S| + 1)F(|S| - 1, |E| - 1)$$

$$\leq c|S||E| + 2c|E| + (2|S| + 1)F(|S| - 1, |E| - 1)$$

for $|S| > 1$. Thus

$$ckn + 2cn + (2k + 1)F(k - 1, n - 1) \geq F(k, n).$$

Moreover by Theorem 4.10, for some constant $d$

$$F(1, n) = dn.$$
Now assume by induction that $F(k - 1, m) \leq dk m \text{odd}!(k)$ for all $m$. Then by (17), if $d \geq 3c$,

$$F(n, k) \leq ckn + 2cn + (2k + 1)F(k - 1, n - 1)$$
$$\leq ckn + 2cn + (2k + 1)d(k - 1)n \text{odd}!(k - 1)$$
$$\leq ckn + 2cn + d(k - 1)n \text{odd}!(k)$$
$$\leq n \text{odd}!(k) \left( \frac{ck}{\text{odd}!(k)} + \frac{2c}{\text{odd}!(k)} + dk - d \right)$$
$$\leq dkn \text{odd}!(k)$$

(19)

Thus, we have established part (a) of Theorem 4.5.

Similarly, we have established that for some fixed constant $c'$,

$$c'|S||E|(|S|! + c'|E| + c'|S||V|(|S|! + c'|S||E|$$
$$+(2|S| + 1)H((|S| - 1, |E| - 1) + 2|S|H(|S| - 1, |E| - 2))$$
$$\geq H(|S|, |E|).$$

(20)

Thus by using the facts that $|V| \leq |E|$ and $|S| \geq 1$, we have shown that

$$2c'kn(k!) + 2c'kn + (2k + 1)H(k - 1, n - 1) + 2kH(k - 1, n - 2) \geq H(k, n).$$

(21)

Again Theorem 4.10 establishes that for some fixed constant $d'$

$$H(1, n) = d'n.$$ 

(22)

Now assume by induction that $H(k - 1, m) \leq d'(k - 1)2^{k-1}m \text{odd}!(k - 1)$. Then by (21), if $d' \geq 4c'$, we have that

$$H(n, k) \leq 2c'kn(k!) + 2c'kn + (2k + 1)d'(k - 1)2^{k-1}(n - 1) \text{odd}!(k - 1)$$
$$+ 2kd'(k - 1)2^{k-1}(n - 2) \text{odd}!(k - 1)$$
$$\leq 2c'kn(k!) + 2c'kn + d'(k - 1)2^k n \text{odd}!(k)$$
$$= 2^k n \text{odd}!(k) \left[ \frac{2c'k(k!)}{2^k \text{odd}!(k)} + \frac{2c'k}{2^k \text{odd}!(k)} + d'k - d \right]$$
$$\leq d'k2^k n \text{odd}!(k).$$

(23)

Thus we have established part (b) of Theorem 4.5. □

5. Tractable cases of McNaughton games

The bound for the running time of the algorithm in Theorem 4.5 is useful for only very small values of $|S|$. Even for $|S| = 6$, the factor $|S|2^{|S|}\text{odd}!(|S|)$ is already equal to $37671480$. Part of the problem is that the general recursive algorithm used to prove Theorem 4.5 does not use any properties of the set of winning sets $\Omega$. In this section, we shall give two examples where if we know something about the form of $\Omega$, then
we can construct $Win_B(\mathcal{G}_M)$, $Win_R(\mathcal{G}_M)$, $f_b$ and $f_r$ in at most $O(|S||E|)$ steps.

Before giving our two examples, we start with an example due to McNaughton.

**Definition 5.1.** Say that a pair $(S, \Omega)$, where $S$ is a finite set and $\Omega \subseteq 2^S$ has a **split** if there exists $\alpha, \beta \subseteq S$ such that either

(i) $\alpha \cup \beta \in \Omega$ and $\alpha, \beta \not\in 2^S - \Omega$ or

(ii) $\alpha \cup \beta \not\in \Omega$ and $\alpha, \beta \in \Omega$.

Suppose $(S, \Omega)$ has no splits, $S \in \Omega$ and $\Omega \neq 2^S$. Then define $o(S, \Omega)$ be the maximum size of an $X \in 2^S - \Omega$. Thus $0 \leq o(S, \Omega) \leq |S| - 1$. Now if $X \in 2^S - \Omega$ and $|X| = o(S, \Omega)$, then for any $y \in S - X$, it follows that for all $\{y\} \subset Z \subseteq S$, $Z \in \Omega$. That is, if $Z \not\in \Omega$, then by the definition of $o(S, \Omega)$, $X \cup Z \in \Omega$ which would violate the fact that $\Omega$ has no splits.

**Definition 5.2.** We say a pair $(S, \Omega)$, where $\Omega \subseteq 2^S$ is a **no memory pair** if for any McNaughton game $\mathcal{G}_M = (G, S, \Omega)$, Black has a no memory strategy from $Win_B(\mathcal{G}_M)$ and Red has a no memory strategy from $Win_R(\mathcal{G}_M)$.

Now McNaughton proved the following.

**Theorem 5.3.** For any finite set $S$ and $\Omega \subseteq 2^S$, $(S, \Omega)$ is a no memory pair iff $(S, \Omega)$ has no splits.

The importance of Theorem 5.3 is that McNaughton proved one direction of Theorem 5.3 by giving an algorithm to solve McNaughton games $\mathcal{G}_M = (G, S, \Omega)$ and produced the required no memory strategies in the case where $(S, \Omega)$ has no splits.

McNaughton did not analyze the complexity of this algorithm so our next result gives a complexity analysis in this case. Unfortunately, the bound is still exponential in $|S|$ and linear in $|E|$.

**Theorem 5.4.** Suppose $\mathcal{G}_M = (G, S, \Omega)$ is a McNaughton game, where $G = (V, V_B, V_R, E)$ and $(S, \Omega)$ has no splits. Then there is an algorithm, which given $\mathcal{G}_M$, runs in $O(|S||2^S||E|)$ steps and produces $Win_B(\mathcal{G}_M)$, $Win_R(\mathcal{G}_M)$, a no memory winning strategy $f_b$ for Black from $Win_B(\mathcal{G}_M)$, and a no memory winning strategy $f_r$ for Red from $Win_R(\mathcal{G}_M)$.

**Proof.** We shall only sketch the algorithm that McNaughton gives since he verifies in [8] that it works.

First note that $(S, \Omega)$ has no splits iff $(S, 2^S - \Omega)$ has no splits so that there is no loss in assuming that $S \in \Omega$. Now fix $X \in 2^S - \Omega$ of size $o(S, \Omega)$ and $y \in S - X$. By our remarks following the definition of $o(S, \Omega)$, if follows that if $\{y\} \subseteq Z \subseteq S$, then $Z \in \Omega$. Thus Black can win any play $p$ in which he can force $p$ to visit $y$ infinitely often.

Our proof proceeds by induction on $|S|$. Let $F(k,n)$ be the maximum number of steps it requires to find $Win_B(G^*_M)$, $Win_R(G^*_M)$, and the required no memory strategies.
$f_b$ and $f_r$ for any game $G_M^* = (G^*, S^*, \Omega^*)$, where $(S^*, \Omega^*)$ has no splits, $|S^*| = k$, and $G^*$ has $n$ edges. Note that by Theorem 4.10, there is a fixed constant $d$ such that

$$F(1, n) = dn.$$  

(24)

Now assume $|S| > 1$. First construct $X_y = X_{\{y\}}$, $\emptyset, R$ and its corresponding no memory strategy $f_y = f_y(\{y\}, \emptyset, B)$ via Lemma 4.8. We can compute $X_y$ and $f_y$ in $c_1|E|$ steps for some fixed constant $c_1$. Next consider the subgame determined by $V - X_y$, $G_M(V - X_y) = (G_{V - X_y}, S(V - X_y), \Omega(V - X_y))$. It is easy to see that $|S(V - X_y)| < |S|$ and $(S(V - X_y), \Omega(V - X_y))$ has no splits so that by induction, in at most $F(|S| - 1, |E| - 1)$ steps, we can construct $Win_B(G_M'(V - X_y))$, $Win_R(G_M'(V - X_y))$, a no memory strategy $g_b$ for Black from $Win_B(G_M'(V - X_y))$, and a no memory strategy $g_r$ for Red from $Win_R(G_M'(V - X_y))$. There are 2 cases.

Case 1: $y \in V_B$ and $Out(y) \cap (X_y \cup Win_B(G_M'(V - X_y))) \neq \emptyset$ or $y \in V_R$ and $Out(y) \not\subseteq X_y \cup Win_B(G_M'(V - X_y))$.

In this case, it is not difficult to show that $Win_B(G_M) = Win_B(G_M'(V - X_y))$ and $Win_R(G_M) = Win_R(G_M'(V - X_y)) \cup X_y$. The no memory winning strategies can be defined as follows.

For $\eta \in V_R$, let

$$f_r(\eta) = \begin{cases} g_r(\eta) & \text{if } \eta \in V - X_y, \\ \delta & \text{where } \delta \text{ is the least element of } Out(\eta) \text{ if } \eta \in X_y. \end{cases}$$

For $\eta \in V_B$, let

$$f_b(\eta) = \begin{cases} f_y(\eta) & \text{if } \eta \in X_y - \{y\}, \\ g_b(\eta) & \text{if } \eta \in V - X_y, \\ \delta & \text{where } \delta \text{ is the least element of } Out(\eta) \cap (X_y \cup Win_B(G_M'(V - X_y))) \text{ if } \eta = y. \end{cases}$$

Note that it takes $c_2|V| \leq c_2|E|$ steps to construct $f_b$ and $f_r$ given $f_y, g_b,$ and $g_r$ for some fixed constant $c_2$.

Case 2: Not case 1.

Construct $Z = X_{Win_B(G_M'(V - X_y))}, \emptyset, R$ and its corresponding no memory strategy $f_Z$ via Lemma 4.8. Our assumptions ensure $y \in X_{Win_B(G_M'(V - X_y))}, \emptyset, R$ in this case. Again we can construct $Z$ and $f_Z$ in $c_2|E|$ steps. Also $V - Z$ determines a subgame, $G_M(V - Z) = (G_{V - Z}, S(V - Z), \Omega(V - Z))$, where $(S(V - Z), \Omega(V - Z))$ has no splits and $|S(V - Z)| < |S|$. Thus by induction, we can find $Win_B(G_M(V - Z))$, $Win_R(G_M(V - Z))$, a no memory winning strategy $h_b$ for Black from $Win_B(G_M(V - Z))$, and a no memory winning strategy $h_r$ for Red from $Win_R(G_M(V - Z))$ in at most $F(|S| - 1, |E| - 1)$ steps. It is then not difficult to show that $Win_B(G_M) = Win_B(G_M(V - Z))$ and $Win_R(G_M) = Z \cup Win_R(G_M(V - Z))$ and the corresponding no memory strategies in this case are defined as follows. For $\eta \in V_B$,

$$f_b(\eta) = \begin{cases} h_b(\eta) & \text{if } \eta \in V - Z, \\ \delta & \text{where } \delta \text{ is the least element of } Out(\eta) \text{ if } \eta \in Z. \end{cases}$$
For $\eta \in V_R$,

$$f_r(\eta) = \begin{cases} f_Z(\eta) & \text{if } \eta \in Z - \text{Win}_R(\mathcal{G}_M(V - X_y)), \\ g_r(\eta) & \text{if } \eta \in \text{Win}_R(\mathcal{G}_M(V - X_y)), \\ h_r(\eta) & \text{if } \eta \in V - Z. \end{cases}$$

That is, suppose Red follows $f_z$ in a play $(\eta_0, \eta_1, \ldots)$. If $\eta_0 \in Z$, then Red is following $f_z$ and hence will force a move to $\text{Win}_R(\mathcal{G}_M(V - X_y))$ in zero or more steps. Since Black has no move from a node in $V - X_y$ into $X_y$, it follows that the remainder of play will stay in $V - X_y$ and hence Red will win since once the play moves to $\text{Win}_R(\mathcal{G}_M(V - X_y))$, Red follows $g_r$ which is a no memory winning strategy for Red from $\text{Win}_R(\mathcal{G}_M(V - X_y))$ in the subgame determined by $V - X_y$. If $\eta_0 \in \text{Win}_R(\mathcal{G}_M(V - Z))$, then Black will either play forever in $V - Z$ or he will eventually force a move into $Z$. In the first case, Red will win because he will follow $h_r$ which is a winning strategy from $\text{Win}_R(\mathcal{G}_M(V - Z))$ in the subgame determined by $V - Z$. In the second case, Red will win since he wins any game which eventually moves into $Z$.

Next suppose that Black is following $f_b$ in a play $(\beta_0, \beta_1, \ldots)$. Note that Red has no move from a node in $V - Z$ into $Z$. Thus if $\beta_0 \in \text{Win}_B(\mathcal{G}_M(V - Z))$, the entire play will lie in $V - Z$ since Black will be following the strategy $h_b$ which is a winning strategy for Black for $\text{Win}_B(\mathcal{G}_M(V - Z))$. Thus Black will win.

Note that it takes at most $c_3 |V| \leq c_3 |E|$ steps to construct $f_b$ and $f_z$ from $h_r, h_b, f_z$, and $g_r$ in case 2. Putting together the analysis of cases 1 and 2, we see that there is some fixed constant $c$ such that

$$c|E| + 2F(|S| - 1, |E| - 1) \geq F(|S|, |E|). \quad (25)$$

Assuming by induction that $F(k - 1, m) \leq d(k - 1)2^{k-1}m$, equation (25) implies

$$F(k, n) \leq cn + 2d(k - 1)2^{k-1}(n - 1)$$

$$\leq cn + 2^kd(k - 1)n$$

$$\leq 2^k n\left(\frac{c}{2k} + d k - d\right). \quad (26)$$

Thus if we pick $d \geq c$, then we may conclude $F(k, n) \leq dk^{2^k}n$. □

There are many special cases for $\Omega$ where we can substantially improve on the running time bounds in Theorems 4.5 and 5.4. Our next two results will show that if $\Omega$ is an interval, i.e. if $\Omega = \{Y : Z_1 \subseteq Y \subseteq Z_2\}$ for some $Z_1 \subseteq Z_2 \subseteq S$, then we can solve a McNaughton game $G_M = (G, S, \Omega)$ in $\mathcal{O}(|S||E|)$ steps.

**Theorem 5.5.** Let $\mathcal{G}_M = (G, S, \Omega)$ be a McNaughton game, where $G = (V, V_B, V_R, E)$. If $\Omega = \{Y : Y \subseteq Z\}$ where $Z \subseteq S$, then there is an algorithm which runs in at most $\mathcal{O}(|S - Z||E|)$ steps and produces $\text{Win}_B(\mathcal{G}_M), \text{Win}_R(\mathcal{G}_M)$, a no memory winning strategy $f_b$ for Black from $\text{Win}_B(\mathcal{G}_M)$ and a no memory winning strategy $f_r$ for Red from $\text{Win}_R(\mathcal{G}_M)$. 

Proof. We proceed by induction on \(|S - Z|\). Note if \(Z = S\), then the game is trivial. That is, if \(Z = S\), then \(\Omega = 2^S\) and Black wins every play no matter what strategy he follows. So assume the result when \(|S - Z| < p\) and let \(Y_0 = S - Z = \{s_1, \ldots, s_p\}\). First construct \(Y_1 = X_{Y_0, 0, R}\), its corresponding no memory strategy for \(f_1 = f_{Y_0, 0, R}\) for Red, and its corresponding no memory strategy \(g_1 = g_{Y_0, 0, R}\) for Black as in Lemma 4.8 and the remarks following Lemma 4.8. Now clearly \(g_1\) is a winning no memory strategy for Black from \(V - Y_1\) in \(G_M(V - Y_1)\). That is, Red has no move from a node \(V - Y_1\) into \(Y_1 = X_{Y_0, 0, R}\). Thus if Black follows \(g_1\) in a play \(p\) which starts in \(V - Y_1\), then Black forces the entire play to stay within \(V - Y_1\). Hence \(\text{perm}(p) \cap Y_1 = \emptyset\) so that certainly \(\text{perm}(p) \cap S - Z = \emptyset\) and Black wins.

Next consider the points \(s_1, \ldots, s_p\). We say that \(s_i\) is bad for Red relative to \(A\) if either

\[
\begin{align*}
(i) & \ s_i \in V_R \cap A \text{ and } \text{Out}(s_i) \subseteq V - A \\
(ii) & \ s_i \in V_B \cap A \text{ and } \text{Out}(s_i) \cap (V - A) \neq \emptyset.
\end{align*}
\]

Now suppose \(B_1\) is the set of elements contained in \(S - Z\) which are bad for Red relative to \(Y_1\). There are 3 cases.

Case 1: \(B_1 = S - Z\).

In this case, we claim \(\text{Win}_B(G_M) = V\). The winning no memory strategy \(f_b\) for Black from \(V\) is defined as follows. For \(\eta \in V_B\), let

\[
f_b(\eta) = \begin{cases} 
g_1(\eta) & \text{if } \eta \in V - Y_1, \\
\delta & \text{where } \delta \text{ is the least element of } \text{Out}(\eta) \cap (V - Y_1), \\
\gamma & \text{where } \gamma \text{ is the least element in } \text{Out}(\eta), \text{ otherwise.}
\end{cases}
\]

That is, suppose \(p = (\eta_0, \eta_1, \ldots)\) is any play in which Black follows \(f_b\). If some \(\eta_k \in V - Y_1\), then in the play \((\eta_k, \eta_{k+1}, \ldots)\), Black is simply following \(g_1\) and since Red has no move from a node in \(V - Y_1\) to \(Y_1\), Black wins \(p\) by our remarks above. Note that by our assumption that \(S - Z\) is bad for \(Y_1\), if some \(\eta_n \in S - Z\), then either \(\eta_n \in V_B\) and hence \(f_b(\eta_n) = g_{\eta_n + 1} \in V - Y_1\) or \(\eta_n \in V_R\) in which case \(\eta_n + 1 \in V - Y_1\) by the fact that \(S - Z\) is bad for \(Y_1\). Thus if the play ever reaches a node in \(S - Z\), the next move must to a node in \(V - Y_1\) so that Black wins \(p\). Of course, if no node \(p\) is ever in \(S - Z\), then certainly \(\text{perm}(p) \subseteq Z\) so that once again Black wins \(p\). Thus Black wins all plays and hence \(\text{Win}_B(G_M) = V\).

Case 2: \(B_1 = \emptyset\).

In this case, we claim \(\text{Win}_B(G_M) = Y_1\) and hence \(\text{Win}_B(G_M) = V - Y_1\) since we have already shown \(g_1\) is a no memory winning strategy for Black from \(V - Y_1\). The fact that \(B_1 = \emptyset\) means that for every \(s_i \in S - Z\) either

\[
\begin{align*}
(a) & \ s_i \in V_B \text{ and } \text{Out}(s_i) \subseteq Y_1 \text{ or} \\
(b) & \ s_i \in V_R \text{ and } \text{Out}(s_i) \cap Y_1 \neq \emptyset.
\end{align*}
\]
The no memory winning strategy $f_r$ for Red from $Y_1$ is then defined for each $\eta \in V_R$ by

$$f_r(\eta) = \begin{cases} 
  f_1(\eta) & \text{if } \eta \notin (S - Z), \\
  \delta & \text{where } \delta \text{ is the least element of } Out(\eta) \cap Y_1 \\
  \delta & \text{if } \eta \in S - Z.
\end{cases}$$

Now suppose $p = (\eta_0, \eta_1, \ldots)$ is a play in which Red follows $f_r$ and which starts in $Y_1$. If $\eta_0 \in Y_1 - (S - Z)$, then Red is following $f_1$ which forces a visit to $S - Z$. Once we are in $S - Z$, our definition of $f_r$ and the fact that $B_1 = \emptyset$ ensures our next move is into $Y_1$, again. Thus either our next move is into $S - Z$ or once again we start in $Y_1 - (S - Z)$ and Red can force another visit to $S - Z$. Thus $p$ will have to visit $S - Z$ infinitely many times and hence $\text{perm}(p) \cap (S - Z) \neq \emptyset$ and Red wins $p$.

Case 3: $\emptyset \neq B_1 \neq S - Z$.

By the same argument as we used in case 1, $(V - Y_1) \cup B_1 \subseteq \text{Win}_B(\mathcal{G}_M)$. That is, let $h_2$ be the following no memory strategy for Black. For $\eta \in V_B$, let

$$h_2(\eta) = \begin{cases} 
  g_1(\eta) & \text{if } \eta \in V - B_1, \\
  \delta & \text{where } \delta \text{ is the least element of } Out(\eta) \cap (V - Y_1), \\
  \delta & \text{if } \eta \in B_1.
\end{cases}$$

Then suppose $p = (\eta_0, \eta_1, \ldots)$ is a play in which Black follows $h_2$. Now if $\eta_0 \in V - Y_1$, then Black is just following $g_1$ and the entire play lies in $V - Y_1$. Thus $\text{perm}(p) \cap (S - Z) = \emptyset$ and Black wins. If $\eta_0 \in B_1$, then our definition of $h_2$ ensures that $\eta_1 \in V - Y_1$ so that in the play $(\eta_1, \eta_2, \ldots)$, Black will once again be following $g_1$ and hence Black will win. Moreover our analysis shows that as long as Black is following $h_2$, then any play which starts in $(V - Y_1) \cup B_1$ stays in $(V - Y_1) \cup B_1$ forever.

Next construct $Y_2 = X_{\gamma_1 \cup B_1 \cup B_2}$ and its corresponding no memory strategy $f_2 = f_{\gamma_1 \cup B_1 \cup B_2}$ for Black. Note that $V - Y_2$ determines a subgame of $\mathcal{G}_M$. Thus in the game $\mathcal{G}_M(V - Y_2) = (G_{V - Y_2}, S(V - Y_2), \Omega(V - Y_2))$, $S(V - Y_2) = S \cap (V - Y_2) \subseteq S - B_1$ and $\Omega(V - Y_2) = \{ Y \in \Omega : Y \subseteq S(V - Y_2) \} = \{ Y : Y \subseteq Z \cap (V - Y_2) \}$. Thus the set of winning sets of $\mathcal{G}_M(V - Y_2)$ has the required form. Moreover, $(S \cap (V - Y_2)) - (Z \cap (V - Y_2)) = (S - Z) \cap (V - Y_2) \subseteq (S - B_1) - Z$. Thus by induction, in $O((S - Z) \cap (V - Y_2)(|E| - 1))$ steps, we can construct $\text{Win}_B(\mathcal{G}_M(V - Y_2))$, $\text{Win}_R(\gamma_2)$, a no memory winning strategy $f_{\gamma_2}$ for Black from $\text{Win}_B(\mathcal{G}_M(V - Y_2))$, and a no memory winning strategy $f_r$ for Red from $\text{Win}_R(\mathcal{G}_M(V - Y_2))$. Then we claim $\text{Win}_B(\mathcal{G}_M) = Y_2 \cup \text{Win}_B(\mathcal{G}_M(V - Y_2))$ and $\text{Win}_R(\mathcal{G}_M) = \text{Win}_R(\mathcal{G}_M(V - Y_2))$.

That is, first extend $f_r$ to a no memory strategy for Red in $\mathcal{G}_M$ by defining for each $\eta \in V_R$,

$$f_r(\eta) = \begin{cases} 
  f_r(\eta) & \text{if } \eta \in V - Y_2, \\
  \delta & \text{where } \delta \text{ is the least element of } Out(\eta) \text{ if } \eta \in Y_2.
\end{cases}$$

By our remarks in Example 4.1, $f_r$ is automatically a no memory winning strategy for Red from $\text{Win}_R(\mathcal{G}_M(V - Y_2))$ in $\mathcal{G}_M$. 
The desired no memory strategy $f_b$ may be defined by setting for each $\eta \in V_B$,

$$f_b(\eta) = \begin{cases} 
    h_2(\eta) & \text{if } \eta \in (V - Y_1) \cup B_1, \\
    f_2(\eta) & \text{if } \eta \in Y_2 - ((V - Y_1) \cup B_1), \\
    \overline{f}_b(\eta) & \text{if } \eta \in V - Y_2.
\end{cases}$$

Now suppose $p$ is any play in which Black is following $f_b$ and which starts in $Win_b(V - Y_2) \cup Y_2$. Now if $p$ ever visits $(V - Y_1) \cup B_1$, then Black will follow $h_2$ for the rest of the play and win. If $p$ ever visits $Y_2 - ((V - Y_1) \cup B_1)$, then Black will follow $f_2$ until he forces a visit to $(V - Y_1) \cup B_1$ and once again Black wins. If $p$ never visits $Y_2$, then Black is following $\overline{f}_b$ in the subgame $\mathcal{G}_M(V - Y_2)$ so again Black must win. Thus $f_b$ is a no memory winning strategy from $Y_2 \cup Win_b \mathcal{G}_M(V - Y_2)$.

Note that by Lemma 4.8, we can compute $Y_1, f_1, g_1, Y_2, f_2$ in $\mathcal{O}(|E|)$ steps. In case 3, we can compute $Win_b(\mathcal{G}_M(V - Y_2)), Win_R(\mathcal{G}_M(V - Y_2)), \overline{f}_b$ and $\overline{f}_b$ in at most $\mathcal{O}((|Z_2| + 1)|E|)$ steps. Finally once we are given all the sets and functions, we can completely compute $Win_b(\mathcal{G}_M), Win_R(\mathcal{G}_M), f_b$, and $f_r$ in an additional $\mathcal{O}(|E|)$ steps. Hence the entire computation takes in $\mathcal{O}(|Z_2| |E|)$ steps. $\square$

Our next result will show that we can also solve all McNaughton games where $Z = \{Y : Z_1 \subseteq Y \subseteq Z_2\}$ and $Z_1 \neq \emptyset$ in at most $\mathcal{O}(|S| |E|)$ steps. In this case, Theorem 5.3 shows that if $|Z_1| > 2$, then such a game cannot always be solved with no memory winning strategies for Black and Red. However we shall see that, we do not require full LVR strategies in this case.

**Theorem 5.6.** Let $\mathcal{G}_M = (G, S, \Omega)$ be a McNaughton game, where $G = (V, V_B, V_R, E)$ and $\Omega = \{Y : Z_1 \subseteq Y \subseteq Z_2\}$ for some $\emptyset \neq Z_1 \subseteq Z_2 \subseteq S$. Then there is an algorithm, which given $\mathcal{G}_M$, runs in at most $\mathcal{O}((2|Z_1| + 1)|E|)$ steps and produces $Win_b(\mathcal{G}_M), Win_R(\mathcal{G}_M)$, a LVR winning strategy $f_b$ for Black from $Win_b(\mathcal{G}_M)$, and a LVR winning strategy $f_r$ for Red from $Win_R(\mathcal{G}_M)$.

**Proof.** The algorithm for computing $Win_b(\mathcal{G}_M), Win_R(\mathcal{G}_M), f_b$ and $f_r$ proceeds as follows. First compute for each $s_i \in Z_1$, $X(s_i), S - Z_2, B$, the corresponding no memory strategy for Black $f_i = f_{\{s_i\}, S - Z_2, B}$, and the corresponding no memory strategy for Red $g_i = g_{\{s_i\}, S - Z_2, B}$. By our remarks above, this takes $\mathcal{O}(|Z_1| |E|)$ steps. There are two cases.

**Case 1:** $|Z_1| = 1$ so that $Z_1 = \{s_1\}$.

In this case, there are two subcases.

**Subcase 1.1:** Either $s_1 \in V_B$ and $Out(s_1) \subseteq V - X(s_1), S - Z_2, B$ or $s_1 \in V_R$ and $Out(s_1) \cap (V - X(s_1), S - Z_2, B) \neq \emptyset$.

In this case, we claim $Win_b(\mathcal{G}_M) = V$. The required LVR winning strategy for Red $f_r$ from $V$ in this case is in fact a no memory strategy. That is, if $s_1 \in V_B$, then let $f_r = g_1$ and if $s_1 \in V_R$, then let $f_r(s_1) = \delta$, where $\delta$ is the least element of $Out(s_1) \cap (V - X(s_1), S - Z_2, B)$ and $f_r(\eta) = g_1(\eta)$ for $\eta \in V_R - \{s_1\}$. It is then easy to see that if $p$ is any play in which Red is following $f_r$, then after every visit to $s_1$, the play next visits $V - X(s_1), S - Z_2, B$ at which point Red starts to follow $g_1$. But this
means that after such a visit, the play must visit some element of $S - Z_2$ before it can again visit $X_{s_1}, S - Z_2, B$ and hence before it can visit $s_1$. Thus either $p$ visits $s_1$ only finitely often so that $s_1 \notin \perm(p)$ or it visits $s_1$ infinitely often in which case $p$ must also visit elements of $S - Z_2$ infinitely often. In the latter case, since $S - Z_2$ is finite, some $x \in S - Z_2$ must be visited infinitely often so that $\perm(p) \cap (S - Z_2) \neq \emptyset$. Thus in either case, it is not the case that $\{s_1\} \subseteq \perm(p) \subseteq Z_2$ so Red wins.

Subcase 1.2: Not subcase 1.1.
Thus if $s_1 \in V_R$, $Out(s_1) \subseteq X_{s_1}, S - Z_2, B$ and if $s_1 \in V_B$, then $Out(s_1) \cap X_{s_1}, S - Z_2, B$. In this case, let $A_1 = X_{s_1}, S - Z_2, B$ and compute $X_{A_1}, \emptyset, B$, its corresponding no memory strategy $f_{A_1} = f_{A_1}, \emptyset, B$ for Black, and its corresponding no memory strategy $g_{A_1} = g_{A_1}, \emptyset, B$ for Red. Then we claim $Win_B(\mathcal{M}) = X_{A_1}, \emptyset, B$ and $Win_R = V - X_{A_1}, \emptyset, B$. Now clearly $g_{A_1}$ is a no memory winning strategy for Red from $V - X_{A_1}, \emptyset, B$ because in any play $p$ in which Red is following $g_{A_1}$ and which starts in $V - X_{A_1}, \emptyset, B$, must stay entirely in $V - X_{A_1}, \emptyset, B$. In particular, since $s_1 \in A_1 \subseteq X_{A_1}, \emptyset, B$, this means that $s_1 \notin \perm(p)$ for any such play so that Red wins $p$. The LVR winning strategy $f_b$ for Black from $X_{A_1}, \emptyset, B, A_1$ is defined so that for each $\eta \in V_B$ and $L \in \mathcal{L}$

$$f_b(L, \eta) = \begin{cases} f_{A_1}(\eta) & \text{if } s_1 \notin L \text{ and } \eta \in X_{A_1}, \emptyset, B - A_1, \\ f_1(\eta) & \text{otherwise,} \end{cases}$$

(27)

where if $s_1 \in V_R$, then $f_1 = f_1$ and if $s_1 \in V_B$, then $f_1(s_1) = \gamma$ is the least element of $Out(s_1) \cap X_{s_1}, S - Z_2, B$ and $f_1(\eta) = f_1(\eta)$ if $\eta \in V_B - \{s_1\}$. Now suppose $p = (\eta_0, \eta_1, \ldots)$ is a play in which Black is following $f_b$ where $\eta_0 \in X_{A_1}, \emptyset, B$. Then either $\eta_0 \in A_1$ or $\eta_0 \notin X_{A_1}, \emptyset, B - A_1$ in which case the effect of Black following $f_b$ is that Black is following the no memory strategy $f_{A_1}$ so that Black will force a visit to $A_1$ in 0 or more moves. Thus in either case, there must be some $i$ such that $\eta_i \in A_1$. Then in the rest of the play $(\eta_i, \eta_{i+1}, \ldots)$, Black will follow $f_1$, which has the effect of forcing a visit to $s_1$ while avoiding $S - Z_2$ and then moving back into $A_1 - \{s_1\}$. Thus in the play $(\eta_i, \eta_{i+1}, \ldots)$, Black will force infinitely many visits to $s_1$ while never visiting and node in $S - Z_2$. Hence $\{s_1\} \subseteq \perm(p) \subseteq Z_2$ and Black wins $p$.

By Lemma 4.8, we can compute $X_{s_1}, S - Z_2, B, f_r, g_1, A_1, \emptyset, B, f_{A_1}$, in $O(2|E|)$ steps. It is then easy to see that it takes an additional $O(|V|)$ steps to compute $Win_B(\mathcal{M})$, $Win_R(\mathcal{M})$, $f_b$ and $f_r$ from this information. Thus certainly we can solve $\mathcal{M}$ in $O(3|E|)$ steps in this case.

Case 2: $Z_1 = \{s_1, \ldots, s_p\}$, where $p \geq 2$.
In this case, there are also two subcases.

Subcase 2.1: $Z_1 \not\subseteq \bigcap_{i=1}^p X_{s_i}, S - Z_2, B$.
In this subcase, there is some fixed $i$ and $j$ less than or equal to $p$ such that $s_j \notin X_{s_i}, S - Z_2, B$. Then we claim $Win_R(\mathcal{M}) = V$ and $f_r = g_i$ is a no memory winning strategy for Red from $V$. That is, let $p = (\eta_0, \eta_1, \ldots)$ be any play in which follows $f_r$. If for some $t$, $\eta_t = s_j \in V - X_{s_i}, S - Z_2, B$ then by following the no memory strategy $g_i$, Red either forces the play to stay entirely within $V - X_{s_i}, S - Z_2, B$. Hence after any visit to $s_j$, Red can force a visit to $S - Z_2$ before the play can again visit $s_i \in X_{s_i}, S - Z_2, B$. Hence, Red wins.
Thus if both $s_i$ and $s_j$ are visited infinitely often during the play $p$, then $S - Z_2$ must be visited infinitely often and hence $\text{perm}(p) \cap (S - Z_2) \neq \emptyset$. Thus for any play, either $\{s_i, s_j\} \subseteq \text{perm}(p)$ or $\text{perm}(p) \cap (S - Z_2) \neq \emptyset$. In either case, Red wins $p$. Thus $f_r$ is a no memory winning strategy for Red from $V$.

**Subcase 2.2:** $Z_1 \subseteq \bigcap_{i=1}^{p} X_{(s_i), S - Z_2, B}$.

In this case, we must compute $Y = X_{Z_1, \emptyset, B}$, its corresponding no memory strategy $f_{Z_1} = f_{Z_1, \emptyset, B}$ for Black, and the corresponding no memory strategy $g_{Z_1} = g_{Z_1, \emptyset, B}$ for Red. In this case, we claim that $\text{Win}_B(\mathcal{G}_M) = Y$ and $\text{Win}_R(\mathcal{G}_M) = V - Y$.

It is easy to see that $V - Y \subseteq \text{Win}_R(\mathcal{G}_M)$. That is, clearly $g_{Z_1}$ is a no memory winning strategy for Red from $V - Y$ since Black has no move from $V - Y$ to $Y$ and hence any play $p$, in which Red is following $g_{Z_1}$ and which starts in $V - Y$, must stay entirely within $V - Y$ and hence $\text{perm}(p) \cap Z_1 = \emptyset$. To define the LVR strategy for Black from $Y$, we must define the notion of the $Z$-target, $t_z(L)$, for a possible last visitation record $L$. Now if $L \subseteq S - Z_1$, we define $t_{Z_1}(L) = \emptyset$. Otherwise we can write $L = (L_0, s_{i_1}, L_1, s_{i_2}, L_2, \ldots, L_{i-1}, s_{i_i}, L_i)$, where $L_0, L_1, \ldots, L_i$ are (possibly empty) sequences from $S - Z_1$ and $i_1, \ldots, i_i \subseteq \{1, \ldots, p\}$. In this case, $t_z(L) = s_{i_t}$ if $t = p$ and $t_z(L) = s_j$, where $j$ is the least element of $\{1, \ldots, p\} - \{i_1, \ldots, i_i\}$ otherwise. Then we claim that the LVR winning strategy $f_b$ for Black from $Y$ can be defined for $(\mathcal{L}, \eta) \in \mathcal{L} \times V_B$ by

$$f_b(L, \eta) = \begin{cases} f_{Z_1}(\eta) & \text{if } t_{Z_1}(L) = \emptyset, \\ f_b(\eta) & \text{if } t_{Z_1}(L) = s_{i_j}. \end{cases} \quad (28)$$

Now consider a play $p = (\eta_0, \eta_1, \ldots)$ in which Black follows $f_b$ and $\eta_0 \in Y$. If $\eta_0 \in Y - Z_1$, then Black will follow the no memory strategy $f_{Z_1}$ until the play reaches some $s_i \in Z_1$. Note that there must be such an $i$ since by construction Black starts our following $f_{Z_1}$ and hence Black can force any play which starts in $Y - Z_1$ to visit $Z_1$ in 0 or more steps. Let $i_0$ be the least $j$ such that $\eta_j \in Z_1$. Then $L(\eta_0, \ldots, \eta_{i_0})$ will be of the form $(L', s_q)$ where $\eta_{i_0} = s_q \in Z_1$. At this point, the effect of Black following $f_b$ is that Black starts following the no memory strategy $f_j$ where $s_j = t((L', s_q))$. Because $\{s_q\} \subseteq X_{(s_j), S - Z_2, B} \subseteq X_{Z_1, \emptyset, B} = Y$, following the no memory strategy $f_j$ will force a visit to $s_j$ in 0 or more moves. Moreover, suppose that $(\eta_{i_0}, \ldots, \eta_m)$ is partial play in which Black is following $f_j$ which has not reached $s_j$. Then it is easy to see that $\text{LVR}(\eta_0, \ldots, \eta_{i_0}, \ldots, \eta_m) = (L_0, s_{i_1}, L_1, \ldots, s_{i_t}, L_t)$, where $L_0, L_1, \ldots, L_t$ are sequences contained in $S - Z_1$, and $q \in \{i_1, \ldots, i_t\} \subseteq \{1, \ldots, p\} - \{j\}$. But since $j = \mu x (s \in \{1, \ldots, p\} - \{q\})$, $j = \mu x (x \in \{1, \ldots, p\} - \{i_1, \ldots, i_t\})$ so that $t(L_{i_0}, \ldots, \eta_{i_1}, \ldots, \eta_{i_t})) = s_j$. This means that in following $f_b$, Black will continue to follow $f_j$ until the play reaches $s_j$. Thus the play will reach $s_j$ by our choice of $f_j$ and hence there will be some least $i_t > i_0$ such that $\eta_{i_t} = s_j$. Moreover during the partial play $(\eta_{i_0}, \ldots, \eta_{i_t})$, we stay entirely within $X_{(s_j), S - Z_1, B}$ so that there are no occurrence of elements of $S - Z_2$ in $(\eta_{i_0}, \ldots, \eta_{i_t})$. More generally suppose that $\eta_n = s_n \in Z_1$, $L_0 = \text{LVR}(\eta_0, \ldots, \eta_n) = (L_0, s_{i_1}, L_1, \ldots, s_{i_t}, L_t, s_n)$, where $L_0$ are sequences contained in $S - Z_1$ and $i_t \leq p$ for $i \leq t$, and $t_{Z_1}(L_n) = s_n$. Note that since $|Z_1| \geq 2$, $m \neq n$. Then starting at $\eta_n$, Black is following $f_m$. Moreover, since $\eta_n = s_n \in X_{(s_n), S - Z_2, B}$, as long
as Black continues to follow $f_m$, Black can force the play to visit $s_m$ before it visits $S - Z_2$. Now suppose $(\eta_1, \ldots, \eta_r)$ is a partial play during which Black is following $f_m$ but has not reached $s_m$. Note that the nodes $(\eta_1, \ldots, \eta_r)$ all lie in $X(s_n), S - Z_2; B - \{s_m\}$ so that there are no occurrences of elements of $S - Z_2 \cup \{s_m\}$ in this partial play. This means that if $s_m \notin \{s_1, \ldots, s_i\}$, then $s_m \notin L_{r} = \text{LVR}(\eta_0, \ldots, \eta_r)$. Also if $H_{r} = \{i: i \leq p$ and $s_i$ occurs $L_{r}\}$, then $\{i_1, \ldots, i_t\} \subseteq H_{r}$. Thus $m = \mu x(x \in \{1, \ldots, p\} - \{i_1, \ldots, i_t\})$ and so that $q_{f_{Z}}(L_{r}) = s_m$. Similarly if $s_m = s_{i_1}$, it is easy to see that $L_{r}$ must be of the form $(L_{0}, s_{i_1}, \ldots, s_{i_r}),$ where $L_{r}$ is a sequence contained in $S - Z_2$ and $j_{r} \leq p$ for $r \leq r$. That is, again $q_{f_{Z}}(L_{r}) = s_m$. It follows that starting at $\eta_1$, Black must continue to follow $f_m$ until it reaches $s_m$. Thus Black will generate a partial play $(\eta_1, \ldots, \eta_r)$, where $(\eta_1, \ldots, \eta_{r-1})$ lies entirely in $X(s_n), S - Z_2; B$. In particular, $(\eta_1, \ldots, \eta_r)$ contains no occurrence of an element of $S - Z_2$. This means that there will be a sequence $i_0 < i_1 < i_2 < \ldots$ such that for all $r \geq 0$, $s_{i_r} \in Z_1$, during the partial play $(\eta_1, \ldots, \eta_r)$, Black is following $f_m$ where $q_{f_{Z}}(\text{LVR}(\eta_0, \ldots, \eta_r)) = s_m$, $s_{i_r+1} = s_m$, and $\eta_1, \ldots, \eta_{i_r}$ occurs entirely within $X(s_n), S - Z_2; B$. This certainly implies $\text{perm}(p) \subseteq Z_2$.

In addition, we claim that $Z_1 \subseteq \text{perm}(p)$. That is, since we always reach the current $Z_1$-target of any sequence $\text{LVR}(\eta_0, \ldots, \eta_r)$ it is easy to see that each $s_i$ with $i \leq p$ will be visited at some point in the play. Thus there must be a $u$ such that $\text{LVR}(\eta_0 \ldots \eta_u) = (L_0, s_{i_1}, L_1, \ldots, s_{i_r}, L_p)$ where $L_i$ is a sequence contained in $S - Z_1$ for $i \leq p$ and $i_1, \ldots, i_p$ is some permutation of $1, \ldots, p$. Now suppose that one of $s_1, \ldots, s_{p}$ is never visited during the play $(\eta_1, \eta_{u+1}, \ldots)$. Then let $a$ be the least $j$ such that $s_j$ is never visited during $(\eta_1, \eta_{u+1}, \ldots)$. Since $s_{i_b}$ for $b < a$ will be visited during $(\eta_1, \eta_{u+1}, \ldots)$, there will be some $v \geq u$ such that $L_v = \text{LVR}(\eta_0 \ldots \eta_v) = (L_0, s_{i_1}, L_1, \ldots, s_{i_r}, L_p)$ where for $i \geq p$, $L_i$ is a sequence contained in $S - Z_1$ and $j_i \leq p$. But then $q_{f_{Z}}(L_v) = s_{i_r}$ and since we have proved that we always reach the $Z_1$-target of $L_v$ during the play $(\eta_1, \eta_{u+2}, \ldots)$, we must visit $s_{i_r}$ during $(\eta_1, \eta_{u+1}, \ldots)$. Thus for every $u$ and every $i \leq p$, $s_i$ occurs in $(\eta_1, \eta_{u+1}, \ldots)$. In particular, this means $Z_1 \subseteq \text{perm}(p) \subseteq Z_2$. Thus, $f_b$ is a LVR winning strategy for Black from $Y$ as claimed.

Note that Eq. (28) represents an algorithm to compute $f_b$ with inputs $f_1, \ldots, f_p$, and $f_Z$. It is easy to see that we could represent $f_b$ by a $|Z_1| + 1 \times |V|$ array in this subcase since $f_b(L, \eta)$ depends only on the $Z_1$-target of $L$. It then easily follows that in either subcase, we can compute $f_1$ and $f_b$ in $C(|Z_1||E|)$ steps given $X(s_n), S - Z_2, B, f_1, g_1, X(s_n), S - Z_2, B, f_i$ and $g_i$ for $i = 1, \ldots, p$. Thus it takes $C((2p + 1)|E|)$ steps to compute $\text{Win}_R(\mathcal{G}_M)$, $\text{Win}_R(\mathcal{G}_M)$ and the LVR winning strategies $f_b$ and $f$ in this case. □

In a subsequent paper, we shall give several other special forms for the set of winning sets $\Omega$ which ensure that we can solve any McNaughton game $\mathcal{G}_M = (G, S, \Omega)$ in $C(|S|^k|E|)$ for some fixed $k$. For example, it is easy to modify the proof of Theorem 5.6 to show that if $\Omega$ is a union of disjoint intervals, then we can solve any McNaughton game $\mathcal{G}_M = (G, S, \Omega)$ in $C(k|S||E|)$ where $k$ is the number of intervals.
References