Jurdzinski-ing Rabin and Streett*

Nir Piterman[†‡] Amir Pnueli[§]

September 22, 2005

Abstract

In this paper we give a recursive fixpoint characterization of the winning regions in Rabin and Streett games. We use this characterization in two ways. First, we introduce direct Rabin and Streett ranking that are a sound and complete way to characterize the winning sets in the respective games. By computing directly and explicitly the ranking we can solve such games in time $O(mn^{k+1}kk!)$ and space $O(nk)$ for Rabin and $O(nkk!)$ for Streett where n is the number of states, m the number of transitions, and k the number of pairs in the winning condition. Second, by keeping intermediate values during the fixpoint evaluation, we can solve such games symbolically in time $O(n^{k+1}k!)$ and space $O(n^{k+1}k!)$. These results improve on the current bounds of $O(mn^{2k}k!)$ time in the case of direct (symbolic) solution or $O(m(nk^2k!)^k)$ in the case of reduction to parity games.

Keywords: Verification, Games, Rabin Games, Streett Games

*This research was supported in part by the Israel Science Foundation (grant no.106/02-1), European community project Prosyd, and the John von-Neumann Minerva center for Verification of Reactive Systems.

†EPFL - I&C - MTC, Lausanne, Switzerland. Email: firstname.lastname@epfl.ch

‡Part of this research was done while this author was visiting the Weizmann Institute of Science.

§Weizmann Institute, Rehovot, Israel. Email: firstname.lastname@weizmann.ac.il

1 Introduction

One of the most ambitious and challenging problems in reactive system construction is the automatic synthesis of programs and (digital) designs from logical specifications. First identified as Church's problem [Chu63], several methods have been proposed for its solution (cf. [BL69, Rab72]). The two prevalent approaches to solving the synthesis problem were by reducing it to the emptiness problem of tree automata, and viewing it as the solution of a two-person game. These two problems are essentially equivalent with efficient reductions between them [Wil01].

A *two-player game* is a finite or infinite directed graph where the vertices are partitioned between the two players. A *play* proceeds by moving a token between the vertices of the graph. If the token is found on a vertex of player 0, she chooses an outgoing edge and moves the token along that edge. If the token is found on a vertex of player 1, she gets to choose the outgoing edge. The result is an infinite sequence of vertices. In order to determine the winner in a play we consider the *infinity set*, the set of states occurring infinitely often in the play. Then, there are several methods to define which infinity sets are winning for which player.

Two of the most natural such acceptance conditions are *Rabin* [Rab69] and *Streett* [Str82]. Both conditions are defined using a set of pairs of subsets of the vertices of the graph. In order to win the Rabin condition over $\{\langle E_1, F_1\rangle, \ldots, \langle E_k, F_k\rangle\}$ the infinity set has to be contained in E_i and intersect F_i for some i. In order to win the Streett condition over $\{\langle E_1, F_1\rangle, \ldots, \langle E_k, F_k\rangle\}$ the infinity set has to either be disjoint from E_i or to intersect F_i for every i. Both Rabin and Streett acceptance conditions are as general as every other ω-*regular* acceptance condition. That is, if the winning condition is defined using some automaton over infinite words (cf. [Tho90]) or as the set of possible infinity sets (*Müller* condition) there is a way to augment the game with a *deterministic monitor* such that the winning condition over the states of the monitor is either Rabin or Streett. Another general acceptance condition is the *parity* acceptance condition [EJ91]. In the parity condition , every vertex has a priority and a play is won if the minimal priority visited infinitely often is even. We mention parity games because our algorithms are derived from similar algorithms that solve parity games.

Rabin conditions arise naturally when the winning condition is supplied in the form of a nondeterminitic Büchi automaton over infinite words. In such a case, the standard approach to solving the game is by converting the nondeterminitic Büchi automaton to a deterministic Rabin automaton [Saf88]. A solution to the Rabin game is then used to solve the original game. Streett conditions arise naturally when considering synthesis of controllers from logic specifications. In many such cases, the controller has to supply *strong fairness*, that is, if some transition / resource is enabled / requested infinitely often it should be taken / granted infinitely often. These kind of requirements translate naturally to Streett conditions.

Consider for example the following specification of an arbiter. The arbiter, controls the grant lines for n agents. Each agent, has a request line r_i which he may raise at will. Once raised, the agent may withdraw the request but only after at least one cycle. The controller has to supply grants so that no two agents may use the resource at the same time (*mutual exclusion*) and so that every agent that requests the resource infinitely often is granted the resource infinitely often. The natural translation of this scenario into a game results in a Streett game with one strong fairness for every grant.

Rabin and Streett games are known to be NP-complete and co-NP-complete respectively [EJ88]. Emerson and Jutla [EJ88] and independently Pnueli and Rosner [PR89] proposed algorithms that solve Rabin and Streett games in time $O((nk)^3 k)$ where n is the number of vertices and k the number of pairs. This was later improved by Kupferman and Vardi to $O(mn^{2k}k!)$ where m is the number of edges [KV98]. Recently, a different solution with the same complexity was given by Horn [Hor05]. It is also possible to solve Rabin and Streett games by reducing them to parity games [EJ91]. This reduction is by adding a deterministic monitor with $k^2k!$ states. The resulting parity game has $nk^2k!$ states and $2k$ priorities. Using the best current solution to parity game, we can solve Rabin and Streett games in time $O((nk^2k!)^k)$ (enumerative algorithm).

Solutions for parity games passed also a long line of improvements. For many years, the best solution to parity games had been the symbolic fixpoint evaluation algorithm of Emerson and Lei [EL86, EJ91]. The complexity of solving a parity game using this approach is mn^k where k is the number of priorities. One major improvement of the classical algorithm has been the observation of Long et al. that by saving intermediate values of the fixpoint computation the run time can be improved to the square root, i.e., $O(n^{\frac{k}{2}})$ [LBC$^+$94]. Long et al. show that by storing intermediate values of the fixpoint computation they can start fixpoint evaluations from better approximations. Unfortunately, the space complexity of this algorithm matches its time complexity.

Jurdzinski matched the square root upper bound while reducing space complexity to linear [Jur00]. His algorithm computes the winning region in a parity game by computing ranks for each vertex. Every vertex with a finite rank is winning and all the rest are losing. The direct rank computation can be accomplished in time $O(mn^{\frac{k}{2}})$. A disadvantage of this approach is that it cannot be applied symbolically. Thus, forcing enumerative approach of the vertices of the game.

Here we generalize these two approaches to Rabin and Streett games. We give an enumerative algorithm that solves Rabin and Streett games in time $O(mn^{k+1}kk!)$ and $O(mn^k kk!)$ respectively and space $O(nk)$ and $O(nkk!)$ respectively. We give a symbolic algorithm that solves Rabin and Streett games in time $O(n^{k+1}k!)$ and space $O(n^{k+1}k!)$.

We start by giving alternative algorithms for solving Rabin and Streett games. The complexity of our algorithms matches the best previous upper bounds of $O(mn^{2k}k!)$. Our algorithm is a recursive fixpoint evaluation algorithm which resembles the fixpoint characterization of parity games [EJ91]. We use the resemblance between our algorithm and the fixpoint characterization of parity games in two ways as explained below.

We introduce Rabin and Streett ranking which resemble Jurdzinski's ranking in that every winning state has a finite rank and the ranking induces a winning strategy. The direct computation of these ranks requires the square root of the time of previous algorithms. Namely, we solve Rabin and Streett games in time $O(mn^{k+1}kk!)$ and $O(mn^k kk!)$ respectively and in space $O(nk)$ and $O(nkk!)$ respectively. Recall that in the worst case a strategy to win a Streett game may require a memory of size $k!$ [DJW97, Hor05]. Thus, it seems that the memory consumption of the Streett algorithm is close to optimal.

We also show that by using Long et al.'s method of fixpoint acceleration we achieve similar results for Rabin and Streett games. That is, by storing the intermediate values of the fixpoints in our algorithm, we can accelerate the fixpoint computation by starting the computation of fixpoints from better approximations. The result is a symblic algorithm that works in time $O(n^k k!)$ and space $O(n^k k!)$.

From our algorithms it follows that Rabin and Streett games are in fact parity games with different orders on the pairs. This has been implicit in the conversion of Rabin and Streett games to parity games, as well as in the solution of Kupferman and Vardi for Rabin games [KV98]. We are the first to take advantage of this connection to improve the run time of the algorithms for Rabin and Streett games almost to a factor of $k!$. We conjecture that similar generalizations can be applied to other algorithms that solve parity games [VJ00, BSV03, JPZ05].

2 Preliminaries

2.1 Linear Temporal Logic

We assume some set of Boolean variables (propositions) P. LTL formulas are constructed as follows.

$$\varphi ::= p \in P \mid \neg\varphi \mid \varphi \vee \varphi \mid \bigcirc \varphi \mid \varphi U \varphi$$

As usual we denote $\neg(\neg\varphi \vee \neg\psi)$ by $\varphi \wedge \psi$, $\mathrm{T} U\varphi$ by $\diamondsuit \varphi$ and $\neg \diamondsuit \neg\varphi$ by $\square \varphi$.

A *model* (alternatively, *word*) w for a formula φ is an infinite sequence of truth assignments to propositions. Namely, a word in $(2^P)^\omega$ is a model. We denote by w_i the set of propositions that are true in location i, that is $w = w_0 \cdot w_1 \cdots$. We present an inductive definition of when a formula holds in model w at time i.

- For $p \in P$ we have $w, i \models p$ iff $w_i(p) = 1$.
- $w, i \models \neg\varphi$ iff $w, i \not\models \varphi$
- $w, i \models \varphi \vee \psi$ iff $w, i \models \varphi$ or $w, i \models \psi$
- $w, i \models \bigcirc \varphi$ iff $w, i+1 \models \varphi$
- $w, i \models \varphi U \psi$ iff there exists $k \geq i$ such that $w, k \models \psi$ and forall $i \leq j < k$ we have $w, j \models \varphi$

For a formula φ and a position $j \geq 0$ such that $w, j \models \varphi$, we say that φ *holds at position* j of w. If $w, 0 \models \varphi$ we say that φ *holds* on w and denote it by $w \models \varphi$. We denote by $L(\varphi)$ the set of models that satisfy φ.

2.2 Games

A *game* is a tuple $G = \langle V, E, W \rangle$ where V is the set of states of the game, V is partitioned to V_E and V_A the sets of states of Eve and Adam respectively, $E \subseteq V \times V$

is the transition relation, and $W \subseteq V^\omega$ is the winning condition of Eve. We assume that for every $v \in V$ there exists some state $v' \in V$ such that $(v, v') \in E$.

A *play* in G is a maximal (hence infinite) sequence of locations $p = v_0 v_1 \cdots$ such that forall $i \geq 0$ we have $(v_i, v_{i+1}) \in E$. A play p is winning for Eve if $p \in W$. Otherwise, Adam wins.

A strategy for Eve is a partial function $f : V^* \times V_{_E} \to V$ such that whenever $f(pv)$ is defined $(v, f(pv)) \in E$. We say that a play $p = v_0 v_1 \cdots$ is *f-conform* if whenever $v_i \in V_{_E}$ we have $v_{i+1} = f(v_0 \cdots v_i)$. The strategy f is *winning from* v if every f-conform play that starts in v is winning for Eve. We say that *Eve wins* from v if she has a winning strategy. The *winning region* of Eve, is the set of states from which Eve wins. We denote the winning region of Eve by $W_{_E}$. A strategy, winning strategy, win, and winning region are defined dually for Adam. We *solve* a game by computing the winning regions $W_{_E}$ and $W_{_A}$. For the kind of games handled in this paper $W_{_E}$ and $W_{_A}$ form a partition of V [GH82].

In this paper we are interested in winning conditions defined by LTL formulas. With a condition we specify whether it is the winning condition for Eve or for Adam. In the case that some LTL formula is the winning condition for Eve its negation is the winning condition for Adam. In order to define the winning condition we assume that the game is equipped with a set of propositions $\mathcal{V}$ and a labeling $L : V \to 2^{\mathcal{V}}$ that labels every state v with the set of propositions that are true in it. We extend L to finite and infinite sequences of states in V and to sets of sequences of states in V in the natural way. When the winning condition for Eve is φ then $W = \{p \mid L(p) \in L(V^\omega) \cap L(\varphi)\}$. When the winning condition for Adam is φ then $W = \{p \mid L(p) \in L(V^\omega) - L(\varphi)\}$.

In this paper we solve Rabin and Streett games. Whenever we work with a Rabin game, the Rabin condition is Eve's goal and whenever we work with a Streett game, the Streett condition is Adam's goal. Formally, we have the following. Both conditions are given by a set $\alpha = \{\langle p_1, q_1\rangle, \ldots, \langle p_k, q_k\rangle\}$ of pairs of propositions. The Rabin condition over α is

$$\bigvee_{i=1}^{k} (\Diamond \Box p \wedge \Box \Diamond q)$$

and the *Streett* condition over α is

$$\bigwedge_{i=1}^{k} (\Diamond \Box p \vee \Box \Diamond q)$$

Notice that the Rabin condition α is **not** the dual of the Streett condition α. We abuse notations and write $G = \langle V, E, \alpha\rangle$ for a game with the Rabin condition α and $G = \langle V, E, \overline{\alpha}\rangle$ for a game with the Streett condition α. The reason for this non-standard notation is that in order to maintain Streett and Rabin as duals we have to introduce negations in front of the propositions. This would burden the notations beyond necessary.

2.3 μ-calculus over Game Structures

We define μ-calculus [Koz83] over game structures. Consider a game $G = \langle V, E, \alpha\rangle$ where V is the disjoint union of $V_{_E}$ and $V_{_A}$ the states of Eve and Adam, respectively.

For every proposition p the formula p is an *atomic formula*. Let $Var = \{X, Y, \ldots\}$ be a set of *relational variables*. The μ-calculus formulas are constructed as follows.

$$\varphi ::= p \mid \neg p \mid X \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond \varphi \mid \Box \varphi \mid \mu X \varphi \mid \nu X \varphi$$

A formula φ is interpreted as the set of states in V in which φ is true. We write such set of states as $[[\varphi]]_G^e$ where G is the game and $e : Var \to 2^V$ is an *environment*. The environment assigns to each relational variable a subset of V. We denote by $e[X \leftarrow V']$ the environment such that $e[X \leftarrow V'](X) = V'$ and $e[X \leftarrow V'](Y) = e(Y)$ for $Y \neq X$. The set $[[\varphi]]_G^e$ is defined inductively as follows[1].

- $[[p]]_G^e = \{s \in V \mid s \models p\}$
- $[[\neg p]]_G^e = \{s \in V \mid s \not\models p\}$
- $[[X]]_G^e = e(X)$
- $[[\varphi \vee \psi]]_G^e = [[\varphi]]_G^e \cup [[\psi]]_G^e$.
- $[[\varphi \wedge \psi]]_G^e = [[\varphi]]_G^e \cap [[\psi]]_G^e$.
- $[[\Diamond \varphi]]_G^e = \left\{ v \in V \;\middle|\; \begin{array}{l} v \in V_E \text{ and } \exists v' \text{ s.t. } (v, v') \in E \text{ and } v' \in [[\varphi]]_G^e \text{ or} \\ v \in V_A \text{ and } \forall v' \text{ s.t. } (v, v') \in E \text{ we have } v' \in [[\varphi]]_G^e \end{array} \right\}$.

  A state v is included in $[[\Diamond \varphi]]_G^e$ if Eve can force the play to reach a state in $[[\varphi]]_G^e$. That is, either v is a state of Eve and has a successor in $[[\varphi]]_G^e$ or v is a state of Adam and all its successors are in $[[\varphi]]_G^e$.
- $[[\Box \varphi]]_G^e = \left\{ v \in V \;\middle|\; \begin{array}{l} v \in V_A \text{ and } \exists v' \text{ s.t. } (v, v') \in E \text{ and } v' \in [[\varphi]]_G^e \text{ or} \\ v \in V_E \text{ and } \forall v' \text{ s.t. } (v, v') \in E \text{ we have } v' \in [[\varphi]]_G^e \end{array} \right\}$.

  A state v is included in $[[\Box \varphi]]_G^e$ if Adam can force the play to reach a state in $[[\varphi]]_G^e$. That is, either v is a state of Adam and has a successor in $[[\varphi]]_G^e$ or v is a state of Eve and all its successors are in $[[\varphi]]_G^e$.
- $[[\mu X \varphi]]_G^e = \cup_i S_i$ where $S_0 = \emptyset$ and $S_{i+1} = [[\varphi]]_G^{e[X \leftarrow S_i]}$.
- $[[\nu X \varphi]]_G^e = \cap_i S_i$ where $S_0 = V$ and $S_{i+1} = [[\varphi]]_G^{e[X \leftarrow S_i]}$

When all the variables in φ are bound by either μ or ν the initial environment is not important and we simply write $[[\varphi]]_G$. In case that G is clear from the context we simply write $[[\varphi]]$.

Consider for example a game $G = \langle V, E, W \rangle$ and the formula $\varphi = \nu X(p \wedge \Diamond X)$. A state $v \in V$ is in $[[\nu X(p \wedge \Diamond X)]]$ if Eve can force the game to remain in p states forever. Indeed Eve can force the game to another state in $[[\nu X(p \Diamond X)]]$ and so on ad-infinitum.

The formula $\psi = \mu X(\neg p \vee \Box X)$ characterizes the set of states from which Adam can force a visit to a $\neg p$ state. Indeed, Adam can force the game in a finite number of steps to the set $[[\neg p]]$.

[1]Only for games with a finite number of states.

We freely use μ-calculus formulas with complex operators that compute sets of states. In such a case we simply use the set returned by the operator in the inductive definition of the μ-calculus. For a full exposition of μ-calculus we refer the reader to [Eme97]. We often abuse notations and write a μ-calculus formula φ instead of the set $[[\varphi]]$.

3 Recursive Algorithm

In this section we present recursive algorithms for computing the winning sets in Rabin and Streett games. Our algorithms match the complexity bounds for Rabin and Streett games established in [KV98, Hor05].

3.1 Rabin Games

We present a recursive algorithm that solves Rabin games. Let $G = \langle V, E, \alpha \rangle$ where $\alpha = \{\langle p_1, q_1\rangle, \ldots, \langle p_k, q_k\rangle\}$ is a Rabin winning condition. An infinite play is winning according to the Rabin condition on α if it satisfies $\bigvee_{i=1}^{k}(\Diamond \Box p_i \wedge \Box \Diamond q_i)$. Intuitively, the algorithm chooses a first pair $\langle p, q\rangle$ from α, it collects recursively all the states that win according to the rest of the pairs while satisfying $\Box p$. We now add states that can visit q infinitely often or get to the previously computed states. We repeat the process for the choice of other pairs as first pair. Here `cpred` denotes the control predecessor. That is, when computing the winning region for Eve `cpred(`X`)` is the set of Eve states that have a successor in X union Adam states all whose successors are in X. The operator `cpred` is dual when computing the winning region for Adam.

Theorem 1 *The algorithm in Fig. 1 computes the winning set of Eve according to the Rabin winning condition.*

Proof: We characterize the set of states returned by `Rabin(S,`φ`,W)`. We show that this is the winning set in a game with a 'simpler' winning condition. We then show how the function `main_Rabin` wraps things up.

Given a set of pairs $S = \{\langle p_1, q_1\rangle, \ldots, \langle p_k, q_k\rangle\}$ we denote by $ltl\_rabin(S)$ the formula $\bigvee_{\langle p,q\rangle \in S}(\Box \Diamond p \wedge \Diamond \Box q)$.

Claim 2 *The function* `Rabin(S,`φ`,W)` *computes the set of states winning for Eve in the game whose winning condition is*

$$win(S, \varphi, W) = \bigvee_{\langle p,q\rangle \in S} [[(\varphi \wedge p)\mathcal{U}W] \vee \Box(\varphi \wedge p \wedge \Diamond q) \vee [ltl\_rabin(S - \langle p, q\rangle) \wedge \Box(\varphi \wedge p)]]$$

Proof: We prove the claim by induction on the number of pairs in S. Suppose $S = \{\langle p, q\rangle\}$, then $Rabin(S, \varphi, W)$ returns the following fixpoint.

$$\nu Y \mu X(W \vee \varphi \wedge p \wedge q \wedge \circledcirc Y \vee \varphi \wedge p \wedge \circledcirc X)$$

```
Func main_Rabin(Set);
LeastFix(Z)
 My r1 := cpred(Z);
 Z := Rabin(Set,true,r1);
End -- LeastFix(Z)
Return Z;
End -- Func main_Rabin(Set)

Func Rabin(Set,seqp,right);
 My U := 0;
 Foreach (<p,q> in Set)
  My nSet := Set-<p,q>;
  GreatestFix(Y)
   My r2 := right |
   seqp & p & q & cpred(Y);
   LeastFix(X)
    My r3 := r2 |
     seqp & p & cpred(X);
     If (|nSet|=0)
      X := r3;
     Else
      X := Rabin(nSet,
            seqp & p,r3);
     End -- If (|nset|=0)
    End -- LeastFix(X)
    Let Y := X;
  End -- GreatestFix(Y)
  Let U := U | Y;
 End -- Foreach (<p,q>
 Return U;
End -- Func Rabin
```

Figure 1: Recursive Algorithm for Rabin: $\bigvee_{i=1}^{k}(\Diamond\Box p_i \wedge \Box\Diamond q_i)$

Let $\hat{Y}$ denote the set computed by this fixpoint. Let $X_0 = \emptyset$ and let

$$X_{i+1} = W \vee (\varphi \wedge p \wedge q \wedge \circledDiamond \hat{Y}) \vee (\varphi \wedge p \wedge \circledDiamond X_i)$$

It follows that $\hat{Y} = \bigcup_{i=1}^{\infty} X_i$. We associate every $v \in \hat{Y}$ a rank that is the minimal i such that $v \in X_i$.

If $v \in X_1$ then either $v \in W$ or $V \models \varphi \wedge p \wedge q$ and Eve can force the play in the next move to some state in $\hat{Y}$. If $v \in X_i$ for $i > 1$ then $V \models \varphi \wedge p$ and Eve can force the play to some state in X_{i-1}. It follows that Eve has a strategy to win $[(\varphi \wedge p)\mathcal{U}W] \vee \Box(p \wedge \varphi \wedge \Diamond q)$. So in the case that $|S| = 1$ the claim is sound.

We show that in the case that $|S| = 1$ the claim is complete. Let W_E denote the winning set for Eve according to the winning condition $win(S, \varphi, W)$. Let $\hat{Y}$ denote some set such that $W_E \subseteq \hat{Y}$. We show that every state from which Eve wins the game with a simpler winning condition is maintained in the computation of the greatest fixpoint. As the winning condition $win(S, \varphi, W)$ implies this simpler winning condition it follows that the greatest fixpoint does not loose winning states according to win. Consider the following winning condition.

$$\psi = [(\varphi \wedge p)\mathcal{U}W] \vee [(\varphi \wedge p)\mathcal{U}(\varphi \wedge p \wedge q \wedge \circledDiamond \hat{Y})]$$

We show that every state winning according to ψ is maintained in the next iteration of the greatest fixpoint. Let X_i for $i \geq 0$ be the sets defined above. For a state v from which Eve wins according to ψ let i denote the maximal number of steps that are taken until a state in W or in $(\varphi \wedge p \wedge q \wedge \circledDiamond \hat{Y})$ is reached. If $i = 1$ then clearly $v \in X_1$. If $i > 1$ then if v is a state of Eve there exists a successor of v whose distance from $W \vee (\varphi \wedge p \wedge q \wedge \circledDiamond \hat{Y})$ is at most $i - 1$. This successor is in X_{i-1} by induction and $v \in X_i$. If v is a state of Adam then all successors of v are in X_{i-1}. This completes the proof of the base case of the induction.

Suppose that the claim is true for sets S of size i. We prove the claim for sets of size $i + 1$. We concentrate on one pair $\langle p, q\rangle \in S$ and denote $S' = S - \langle p, q\rangle$. The largest fixpoint in `Rabin(`S`,`φ`,`W`)` computes the following set.

$$\nu Y \mu X[Rabin(S', \varphi \wedge p, W \vee (\varphi \wedge p \wedge q \wedge \circledDiamond Y) \vee (\varphi \wedge p \wedge \circledDiamond X))]$$

As before let $\hat{Y}$ denote the result of this fixpoint, let X_0 be the empty set and let

$$X_{i+1} = Rabin(S', \varphi \wedge p, W \vee (\varphi \wedge p \wedge q \wedge \circledDiamond \hat{Y}) \vee (\varphi \wedge p \wedge \circledDiamond X_i))$$

from the induction assumption

$$x_{i+1} = win(S', \varphi \wedge p, W \vee (\varphi \wedge p \wedge q \wedge \circledDiamond \hat{Y}) \vee (\varphi \wedge p \wedge \circledDiamond X_i))$$

Suppose that $v \in X_1$. By induction, from v Eve wins with the winning condition

$$\bigvee_{\langle p', q'\rangle \in S'} \left[\begin{array}{ll} ((\varphi \wedge p \wedge p')\mathcal{U}(W \vee (\varphi \wedge p \wedge q \wedge \circledDiamond \hat{Y}))) & \vee \\ \Box(\varphi \wedge p \wedge p' \wedge \Diamond q') & \vee \\ (ltl\_rabin(S' - \langle p', q'\rangle) \wedge G(\varphi \wedge p \wedge p')) & \end{array} \right]$$

which is equivalent to

$$\bigvee_{\langle p',q'\rangle\in S'}\left[\begin{array}{ll}((\varphi\wedge p\wedge p')\mathcal{U}(\varphi\wedge p\wedge q\wedge \bigcirc\hat{Y})) & \vee\\ ((\varphi\wedge p\wedge p')\mathcal{U}W) & \vee\\ \Box(\varphi\wedge p\wedge p'\wedge\Diamond q') & \vee\\ (ltl\_rabin(S'-\langle p',q'\rangle)\wedge G(\varphi\wedge p\wedge p')) & \end{array}\right]$$

So Eve can either (a) force the game to a state in W while maintaining $\varphi\wedge p$, (b) force the game to a state that satisfies $\varphi\wedge p\wedge q$ while maintaining $\varphi\wedge p$ and then force the game in the next move to $\hat{Y}$ or (c) win according to the rest of the condition.

Suppose that $v\in X_i$ for $i>1$. By induction, from v Eve wins with the winning condition

$$\bigvee_{\langle p',q'\rangle\in S'}\left[\begin{array}{ll}((\varphi\wedge p\wedge p')\mathcal{U}(W\vee(\varphi\wedge p\wedge q\wedge\bigcirc\hat{Y})\vee(\varphi\wedge p\wedge\bigcirc X_{i-1}))) & \vee\\ \Box(\varphi\wedge p\wedge p'\wedge\Diamond q') & \vee\\ (ltl\_rabin(S'-\langle p',q'\rangle)\wedge G(\varphi\wedge p\wedge p')) & \end{array}\right]$$

which is equivalent to

$$\bigvee_{\langle p',q'\rangle\in S'}\left[\begin{array}{ll}((\varphi\wedge p\wedge p')\mathcal{U}(\varphi\wedge p\wedge\bigcirc X_{i-1})) & \vee\\ ((\varphi\wedge p\wedge p')\mathcal{U}(\varphi\wedge p\wedge q\wedge \bigcirc\hat{Y})) & \vee\\ ((\varphi\wedge p\wedge p')\mathcal{U}W) & \vee\\ \Box(\varphi\wedge p\wedge p'\wedge\Diamond q') & \vee\\ (ltl\_rabin(S'-\langle p',q'\rangle)\wedge\Box(\varphi\wedge p\wedge p')) & \end{array}\right]$$

So Eve has a strategy that either (a) forces the game to a state in W while maintaining $\varphi\wedge p$, (b) forces the game to a state in $\varphi\wedge p\wedge q\wedge\bigcirc\hat{Y}$ while maintaining $\varphi\wedge p$, (c) forces the game to X_{i-1} while maintaining $\varphi\wedge p$, or (d) wins according to

$$\psi := \bigvee_{\langle p',q'\rangle\in S}\left[\begin{array}{ll}\Box(\varphi\wedge p\wedge p'\wedge\Diamond q') & \vee\\ (ltl\_rabin(S-\langle p',q'\rangle)\wedge\Box(\varphi\wedge p\wedge p')) & \end{array}\right]$$

We combine these strategies as follows. Consider a state $v\in\hat{Y}$. Let i be the minimal such that $v\in X_i$. Eve applies the ith strategy. Either the play remains in X_i indefinitely or it reaches X_{i-1} and Eve switches to the $i-1$th strategy. If while playing according to some strategy the play reaches $\varphi\wedge p\wedge q$ then Eve chooses some successor in $\hat{Y}$ and continues with the appropriate strategy. Consider an infinite play according to the combination of the strategies as explained above. Either for some i the play stays indefinitely in X_i and wins according to ψ or infinitely often the play reaches X_1 and wins according to $\Box(\varphi\wedge p\wedge\Diamond q)$.

Every play is won according to one of the following.

- The play starts with a finite prefix of $\varphi\wedge p$ and stays eventually always within some X_i and wins according to ψ.
- The play visits X_1 infinitely often and satisfies $\Box(\varphi\wedge p\wedge\Diamond q)$.
- The play gets to W along a $\varphi\wedge p$ path.

This means that the Rabin player wins according to the following condition.

$$\bigvee_{\langle p',q'\rangle\in S'}\left[\begin{array}{ll}\Box(p\wedge\varphi\wedge\Diamond q) & \vee\\ (\varphi\wedge p)\mathcal{U}W & \vee\\ (\Diamond\Box p'\wedge\Box\Diamond q')\wedge\Box(p\wedge\varphi) & \vee\\ (ltl\_rabin(S'-\langle p',q'\rangle)\wedge\Box(\varphi\wedge p)) & \end{array}\right]$$

or equivalently

$$\begin{array}{ll}\Box(p\wedge\varphi\wedge\Diamond q) & \vee\\ (\varphi\wedge p)\mathcal{U}W & \vee\\ (ltl\_rabin(S')\wedge\Box(\varphi\wedge p)) & \end{array}$$

We note that the greatest fixpoint in `Rabin` is nested in a loop going over all pairs in S. We conclude that the winning condition is of the wanted form.

We now prove the completeness of the induction step. We show that every iteration of the greatest fixpoint maintains all the states winning according to a simpler winning condition ψ. As $win(S,\varphi,W)$ implies ψ it follows that every state winning according to win remains in the greatest fixpoint. Consider some pair $\langle p,q\rangle\in S$ and denote $S'=S-\langle p,q\rangle$. Let W_E denote the winning set for Eve according to the disjunct of $\langle p,q\rangle$ in $win(S,\varphi,W)$. Let $\hat{Y}$ denote some set such that $W_E\subseteq\hat{Y}$. We show that every state from which Eve wins the game whose winning condition is

$$\psi=\begin{array}{ll}(\varphi\wedge p)\mathcal{U}(\varphi\wedge p\wedge q\wedge\bigcirc\!\!\!\!\Diamond\,\hat{Y}) & \vee\\ (\varphi\wedge p)\mathcal{U}W & \vee\\ (ltl\_rabin(S')\wedge\Box(\varphi\wedge p)) & \end{array}$$

is maintained by the greatest fixpoint.

Denote the winning region for Eve according to the winning condition ψ by T. We analyze the form of T, our methods remind the methods in [KV01]. We show that as long as $\hat{X}$ is not equal to T the equation $rabin(S',\varphi\wedge p,W\vee(\varphi\wedge p\wedge q\wedge\bigcirc\!\!\!\!\Diamond\,\hat{Y})\vee(\varphi\wedge p\wedge\bigcirc\!\!\!\!\Diamond\,\hat{X}))$ increases the size of X. As T is finite it follows that eventually the minimal fixpoint equals T.

It is clear that every state in T satisfies $\varphi\wedge p$. Suppose that there exists some state $v\in T-\hat{X}$ such that Eve can control the play to reach $\hat{X}$ in one step then v is included in the next value of the fixpoint. Suppose that no such state exists. We show that there exists a state from which Eve wins according to $win(S',\varphi\wedge p,W\vee(\varphi\wedge p\wedge\hat{X}))$. That is, Eve's strategy on $T-\hat{X}$ maintains $\Box(\varphi\wedge p)$ and wins according to $ltl\_rabin(S')$. We show that there exists a node $v\in T-\hat{X}$ such that Eve's winning strategy maintains $\Box p'$ for some pair $\langle p',q'\rangle\in S'$. This state is included in $win(S',\varphi\wedge p,W\vee(\varphi\wedge p\wedge\bigcirc\!\!\!\!\Diamond\,\hat{X}))$ by the induction assumption. Suppose that there does not exist a state $v\in T-\hat{X}$ such that for some $\langle p',q'\rangle\in S'$ Eve's winning strategy maintains $\Box p'$ on all plays continuing from v. Let $\langle p',q'\rangle$ be the first pair in S'. By assumption there does not exist a state from which Eve maintains $\Box p'$. Let $v_0\in T-\hat{X}$ be some state such that $v_0\models\neg p'$. We recall that Eve cannot force an immediate visit to $\hat{X}$. There exists a successor v_1 that is either chosen by Eve (in case that v_0 is a state of Eve) or it is some successor of v_0 in $T-\hat{X}$ (in case that v_0 is a state of Adam). We construct by induction an infinite path in $T-\hat{X}$ that visits $\neg p'$ for every $\langle p',q'\rangle\in S'$ infinitely often. This path cannot be winning according to ψ. We conclude that there exists a node v from which Eve's winning strategy maintains $\Box\, p'$ for some $\langle p', q'\rangle \in S'$. This state is winning according to $win(S', \varphi \wedge p, W \vee (\varphi \wedge p \wedge \circledcirc\, \hat{X}))$ and it is included in the next iteration of the fixpoint. This concludes completeness of the induction step. $\square$

We handle the function `main_Rabin`. From the previous proof it immediately follows that every state returned by `main_Rabin` is winning for Rabin. We have to show that every state winning for Rabin is included. Similar to the completeness proof above, we analyze the winning region for Eve in the Rabin game. We claim that there exists a region in the winning region of Eve that satisfies $\Box\, p$ for some $\langle p, q\rangle \in S$. Such a region satisfies the condition $win(S, true, \emptyset)$. It follows that it is returned in a call to `Rabin(S,true,∅)`. Then the minimal fixpoint collects all states that can reach these regions in a finite number of states and collects other such regions. As before, the game is finite so it is eventually depleted.

Formally, assume that W is the set of states computed by the minimal fixpoint in `main_Rabin`. Assume further that $W_{_E}$ is the set of winning states for Eve in the Rabin game and that $W_{_E} - W \neq \emptyset$. Suppose that there exists some state v in $W_{_E} - W$ such that Eve can control the play to reach W in one step. Then v is included in the next value of the fixpoint. Suppose that no such state exists. Then we show that there exists a state from which Eve wins according to $win(S, true, W)$. That is, Eve's strategy on $W_{_E} - W$ maintains $\Box\, p$ for some pair $\langle p, q\rangle \in S$ and in addition wins according to the Rabin condition. Suppose that there does not exist a state $v \in W_{_E} - W$ such that for some $\langle p, q\rangle \in S$ Eve's winning strategy maintains $\Box\, p$. Let v_0 be some state such that $v_0 \models \neg p_1$. We recall that Eve cannot force an immediate visit to W. There exists a successor v_1 that is either chosen by Eve or it is some successor of v_0 in $W_{_E} - W$. We construct by induction an infinite path in $W_{_E} - W$ that visits $\neg p$ for every pair $\langle p, q\rangle \in S$ infinitely often. This path cannot be winning according to the Rabin condition and we conclude that a state from which Eve's winning strategy maintains $\Box\, p$ exists. This state is included in the next iteration of the fixpoint and eventually the minimal fixpoint equals $W_{_E}$. $\square$

3.2 Streett Games

We present a recursive algorithm that solves Streett games. Let $G = \langle V, E, \overline{\alpha}\rangle$ where $\alpha = \{\langle p_1, q_1\rangle, \ldots, \langle p_k, q_k\rangle\}$ is a Streett winning condition. We recall that in our notation, the Rabin condition α is not the dual of the Streett condition α. An infinite play is winning according to the Streett condition on α if it satisfies $\bigvee_{i=1}^{k}(\Diamond\, \Box\, p \vee \Box\, \Diamond\, q)$. Intuitively, the algorithm chooses a pair $\langle p, q\rangle$ in α, it collects all states that eventually visit only p states while making sure recursively that all other pairs are satisfied. We then add states that can visit q infinitely often and do the same for all other pairs.

Theorem 3 *The algorithm in Fig. 2 computes the winning set of Adam according to the Streett winning condition.*

```
Func Streett(Set,seqp,right)
 GreatestFix(Z)
  Foreach (<p,q> in Set)
   My nSet := Set-<p,q>;
   My r1 := right |
    seqp & q & cpred(Z);
   LeastFix(Y)
    My r2 := r1 |
        seqp & cpred(Y);
    If (|nSet|=0)
     Y := m_Streett(
            seqp & p,r2);
    Else
     Y := Streett(nSet,
           seqp & p,r2);
    End -- If (|nSet|=0)
   End -- LeastFix(Y)
   Z := Y;
  End -- Foreach (<p,q>
 End -- GreatestFix(Z)
 Return Z;
End -- Streett

Func m_Streett(seqp,right)
 GreatestFix(X)
  X := right |
      seqp & cpred(X);
 End -- GreatestFix(X)
 Return X;
End -- m_Streett
```

Figure 2: Recursive Algorithm for Streett: $\bigwedge_{i=1}^{k}(\Diamond \Box p_i \vee \Box \Diamond q_i)$

Proof: We characterize the set of states returned by `Streett(S,`φ`,W)`. We show that this is the winning set in a game with a 'simpler' winning condition.

Given a set of pairs $S = \{\langle p_1, q_1\rangle, \ldots, \langle p_k, q_k\rangle\}$ we denote by $ltl\_streett(S)$ the formula $\bigwedge_{\langle p,q\rangle\in S}(\Diamond \Box p \vee \Box \Diamond q)$.

Claim 4 *The function* `Streett(S,`φ`,W)` *computes the set of states winning for Adam in the game whose winning condition is*

$$win(S,\varphi,W) = \begin{array}{ll} (\varphi \mathcal{U} W) & \vee \\ \bigwedge\limits_{\langle p,q\rangle \in S} \left[\begin{array}{ll} \Box(\varphi \wedge \Diamond q) & \vee \\ \varphi \mathcal{U}(\Box(\varphi \wedge p) \wedge ltl\_streett(S - \langle p,q\rangle)) & \end{array} \right] & \end{array}$$

Proof: We prove the claim by induction on the number of pairs in S. Suppose $S = \emptyset$, then $minor\_streett(\emptyset, \varphi, W)$ computes the following fixpoint.

$$\nu X(W \vee \varphi \wedge \bigcirc\!\!\!\!\!\Box\, X)$$

It is simple to see that this is the winning set in the game whose winning condition is $\varphi \mathcal{U} W \vee \Box\, \varphi$.

Suppose that the claim is true for sets S of size i. We prove the claim for sets of size $i+1$. Let $\hat{Z}$ denote the set computed by the greatest fixpoint. It follows that for every $\langle p,q\rangle \in S$ we have

$$\hat{Z} = \mu Y[Streett(S - \langle p,q\rangle, \varphi \wedge p, W \vee (\varphi \wedge q \wedge \bigcirc\!\!\!\!\!\Box\, \hat{Z}) \vee (\varphi \wedge \bigcirc\!\!\!\!\!\Box\, Y))]$$

We concentrate on some $\langle p,q\rangle \in S$ and denote $S' = S - \langle p,q\rangle$. Let $Y_0 = \emptyset$ and let

$$Y_{i+1} = Streett(S', \varphi \wedge p, W \vee (\varphi \wedge q \wedge \bigcirc\!\!\!\!\!\Box\, \hat{Z}) \vee (\varphi \wedge \bigcirc\!\!\!\!\!\Box\, Y_i))$$

For every state $v \in \hat{Z}$ let $r(v)$ be the minimal i such that $v \in Y_i$.

Consider a state v such that $r(v) = 1$. By induction Adam wins from v according to

$$\begin{array}{ll} \varphi \mathcal{U}(W \vee (\varphi \wedge q \wedge \bigcirc\!\!\!\!\!\Box\, \hat{Z})) & \vee \\ \bigwedge\limits_{\langle p',q'\rangle \in S'} \left[\begin{array}{ll} \Box(\varphi \wedge p \wedge \Diamond q') & \vee \\ (\varphi \wedge p) \mathcal{U}(\Box(\varphi \wedge p) \wedge ltl\_streett(S' - \langle p',q'\rangle)) & \end{array} \right] & \end{array}$$

or equivalently

$$\begin{array}{ll} \varphi \mathcal{U}(\varphi \wedge q \wedge \bigcirc\!\!\!\!\!\Diamond\, \hat{Z}) & \vee \\ \varphi \mathcal{U} W & \vee \\ \bigwedge\limits_{\langle p',q'\rangle \in S'} \left[\begin{array}{ll} \Box(\varphi \wedge p \wedge \Diamond q') & \vee \\ (\varphi \wedge p) \mathcal{U}(\Box(\varphi \wedge p) \wedge ltl\_streett(S' - \langle p',q'\rangle)) & \end{array} \right] & \end{array}$$

So there exists a strategy such that Adam either (a) reaches W while staying in φ states, (b) reaches $\varphi \wedge q \wedge \bigcirc\!\!\!\!\!\Diamond\, \hat{Z}$ while staying in φ states, or (c) the play is infinite and it is always in $\varphi \wedge p$ states while satisfying the rest of the Streett pairs.

Consider a state v such that $r(v) = i > 1$. By induction Adam wins from v according to

$$\begin{array}{ll}
\varphi\mathcal{U}(\varphi \wedge \circledcirc Y_{i-1}) & \vee \\
\varphi\mathcal{U}(\varphi \wedge q \wedge \circledcirc \hat{Z}) & \vee \\
\varphi\mathcal{U} W & \vee \\
\bigwedge_{\langle p',q'\rangle \in S'} \left[\begin{array}{ll} \square(\varphi \wedge p \wedge \diamondsuit q') & \vee \\ (\varphi \wedge p)\mathcal{U}(\square(\varphi \wedge p) \wedge ltl_streett(S' - \langle p', q'\rangle)) & \end{array} \right] &
\end{array}$$

So there exists a strategy such that Adam either (a) reaches W while staying in φ states, (b) reaches a state with lower rank or reaches $\varphi \wedge q$ while staying in φ states, or (c) the play is infinite and it is always in $\varphi \wedge p$ states while satisfying the rest of the Streett pairs.

We now combine these strategies to prove the soundness of the induction step. In states whose rank is i Adam plays the ith strategy. When playing according to some strategy and getting to a state in $\varphi \wedge q \wedge \circledcirc \hat{Z}$, Adam chooses some successor and the rank may increase arbitrarily. Every play either stays within some Y_i from some stage onwards and continues indefinitely according to the ith strategy or infinitely often switches between the strategies. In the first case, the play fulfills

$$\varphi\mathcal{U} \bigwedge_{\langle p',q'\rangle \in S'} \left[\begin{array}{ll} \square(\varphi \wedge p \wedge \diamondsuit q') & \vee \\ (\varphi \wedge p)\mathcal{U}(\square(\varphi \wedge p) \wedge ltl\_streett(S' - \langle p', q'\rangle)) & \end{array} \right]$$

which implies $\varphi\mathcal{U}(ltl\_streett(S') \wedge \square(\varphi))$. In the second case, the play fulfills $\square(\varphi \wedge \diamondsuit q)$. Soundness follows.

We now prove the completeness of the induction step. Let W_A denote the winning set of Adam according to $win(S, \varphi, W)$. Let $\hat{Z}$ denote some set such that $W_A \subseteq \hat{Z}$. We concentrate on some pair $\langle p, q\rangle \in S$ and denote $S' = S - \langle p, q\rangle$. We show that every state from which Adam wins the game according to

$$\psi = \begin{array}{ll} \varphi\mathcal{U}(W \vee (\varphi \wedge q \wedge \circledcirc \hat{Z})) & \vee \\ \varphi\mathcal{U} \left[\begin{array}{ll} \square(\varphi \wedge p) & \wedge \\ ltl\_streett(S - \langle p, q\rangle) & \end{array} \right] & \end{array}$$

is maintained by the fixpoint. Clearly, a state winning according to $win(S, \varphi, W)$ is winning according to ψ.

Denote the winning region for Adam according to ψ by T. We analyze the form of T. We show that as long as $\hat{Y}$ is not equal to T the equation $streett(S', \varphi \wedge p, W \vee (\varphi \wedge \circledcirc \hat{Y}))$ increases the size of Y. As T is finite it follows that eventually the minimal fixpoint equals T.

It is clear that every state in T satisfies φ. Suppose that there exists some state $v \in T$ such that Adam can control the play to reach $\hat{Y}$ in one step, then v is included in the next value of the fixpoint. Suppose that no such state exists. We show that there exists a state from which Adam wins according to $win(S', \varphi \wedge p, W \vee (\varphi \wedge \circledcirc \hat{Y}))$.

As before it is sufficient to prove that there exists a state v such that Adam's winning strategy for ψ maintains $\square(\varphi \wedge p)$. Combining $\square(\varphi \wedge p)$ with ψ gives us a region winning with respect to $win(S', \varphi \wedge p, W \vee (\varphi \wedge \bigcirc \hat{Y}))$. Suppose that there does not exists a state v such that Adam's winning strategy maintains $\Box\, p$. Let v_0 be some state such that $v_0 \models \neg p$. We recall that Adam cannot force an immediate visit to $\hat{Y}$. There exists a successor v_{2i+1} of v_{2i} that is either chosen by Adam (in case that v_{2i} is a state of Adam) or is some successor of v_0 in $T - \hat{Y}$ (in case that v_{2i} is a state of Eve). By assumption Adam does not maintain $\Box\, p$ and there exists a node v_{2i+2} that is reachable from v_{2i+1} using Adam's winning strategy such that $v_{2i+2} \models \neg p$. By induction we construct an infinite path in $T - \hat{X}$ that respects Adam's winning strategy and visits $\neg p$ infinitely often. This path cannot be winning according to ψ. We conclude that there exists a node v from which Adam's winning strategy maintains $\Box\, p$. This state is in addition winning according to ψ. We conclude that this state is winning according to $win(S', \varphi \wedge p, W \vee (\varphi \wedge \bigcirc \hat{Y}))$ and that it is included in the next iteration of the fixpoint. This concludes completeness of the induction step. □

□

From Theorems 1 and 3 it is easy to derive the following bounds. A greatest or least fixpoint collects at least one state in every iteration and hence cannot be repeated more than n times. The inner most fixpoint can be computed in time proportional to m where m is the number of transitions.

Corollary 5 *Rabin and Streett games can be solved in time $O(mn^{2k}k!)$ where n is the number of states, m is the number of transitions, and k is the number of pairs of the winning condition.*

4 Rabin and Streett Ranking

In this section we show how to define Rabin and Streett ranking based on the recursive algorithms presented in the previous section. By computing these ranks directly we achieve algorithms for Rabin and Streett games that require $O(mn^{k+1}kk!)$ time and $O(nk)$ space or $O(mn^k kk!)$ time and $O(nkk!)$ space respectively.

4.1 Rabin Ranking

Consider a game $G = \langle V, E, \alpha \rangle$ where $\alpha = \{\langle p_1, q_1 \rangle, \ldots, \langle p_k, q_k \rangle\}$ is a Rabin winning condition. Eve wins an infinite play if it satisfies $\bigvee_{i=1}^{k} (\Diamond \Box\, p_i \wedge \Box \Diamond\, q_i)$. Let $\Pi(k)$ denote the set of permutations over $[1..k]$. Given a permutation $\pi = j_1 j_2 \cdots j_k \in \Pi(k)$ we denote j_i by π_i.

The *Rabin domain* for α over V is $D_{\scriptscriptstyle R}(\alpha, V) = \{i_0 j_1 i_1 j_2 \cdots j_k i_k \mid i_0 \cdots i_l \in [0..n]^{k+1}$ and $j_1 \cdots j_k \in \Pi(k)\} \cup \{\infty\}$. That is, the domain contains the interleaving of a $k+1$ tuple of integers with a permutation over $[1..k]$. Every integer is bounded by n. For simplicity of notations we write $D_{\scriptscriptstyle R}$ instead of $D_{\scriptscriptstyle R}(\alpha, V)$ and Π instead of $\Pi(k)$. Given $d = i_0 j_1 \cdots j_k i_k \in D_{\scriptscriptstyle R}$ we denote by $\pi(d)$ the permutation $j_1 \cdots j_k$ and by $m(d)$ the tuple $i_0 \cdots i_k \in [0..n]^k$. We order $D_{\scriptscriptstyle R}$ according to the lexicographic ordering with ∞ as maximal element.

A *Rabin ranking* over V is $r : V \to D_{_R}$. Given a node $v \in V$ and a Rabin ranking r we denote by $best(v)$ the rank of the minimal successor of v in case that $v \in V_{_E}$ or the rank of the maximal successor of V in case that $v \in V_{_A}$. Formally,

$$best(v) = \begin{cases} min_{(v,w)\in E}(r(w)) & v \in V_{_E} \\ max_{(v,w)\in E}(r(w)) & v \in V_{_A} \end{cases}$$

We say that a Rabin ranking is *good* if for every state v such that $r(v) \neq \infty$ we have $best(v)$ is *better* than $r(v)$. Let $r(v) = i_0 j_1 i_1 \cdots i_k$ and and $best(v) = i'_0 j'_1 i'_1 \cdots i'_k$. We say that $best(v)$ is *better* than $r(v)$ if $i_0 > i'_0$ or $i_0 = i'_0$ and $best(v)$ is *better*$_1$ than $r(v)$. We say that $best(v)$ is *better*$_l$ than $r(v)$ if one of the following holds.

- $j_l > j'_l$.
- $j_l = j'_l$, $v \models p_{j_l}$, and $i_l > i'_l$.
- $j_l = j'_l$, $v \models p_{j_l}$, and $v \models q_{j_l}$.
- $j_l = j'_l$, $v \models p_{j_l}$, $i_l = i'_l$, and $best(v)$ is better$_{l+1}$ than $r(v)$.

If one of the first three conditions holds we say that $best(v)$ is *strictly better*$_l$ than $r(v)$. It is simple to see that if $v \in V_{_A}$ and $best(v)$ is better than $r(v)$ then for every node w such that $(v, w) \in E$ we have $r(w)$ is better than $r(v)$. This follows from $r(w)$ being at most $best(v)$.

We show that a good Rabin ranking induces a wining strategy for Eve.

Claim 6 *Given a good Rabin ranking r, Eve wins the Rabin game from every state v such that $r(v) \neq \infty$.*

Proof: Consider the following strategy. From a state $v \in V_{_E}$ choose the successor w such that $r(w)$ is minimal. We show that this strategy is winning.

Consider an infinite play $v_0 v_1 \cdots$ that conforms to this strategy. Let $r_0 r_1 \cdots$ denote the sequence of ranks such that $r_m = r(v_m)$ and $r_m = i_0^m j_1^m i_1^m \cdots j_k^m i_k^m$. From the definition of good ranking it follows that it is always the case that r_{m+1} is better than r_m. Let l be the minimal value such that there exist infinitely many m such that r_{m+1} is strictly better$_l$ than r_m. There exists m' such that forall $m > m'$ and forall $l' < l$ we have r_{m+1} is not strictly better$_{l'}$ from r_m. So for all $l' < l$ and forall $m > m'$ we have $j_{l'}^m = j_{l'}^{m+1}$, $v_m \models p_{j_{l'}^m}$, and $i_{l'}^m = i_{l'}^{m+1}$. Similarly, there exists $u \in [1..k]$ and $m'' > m'$ such that forall $m > m'$ we have $j_l^m = u$. Consider the pair $\langle p_u, q_u \rangle$ and the suffix of the play starting from m''. For every $m > m''$ we have r_{m+1} is better than r_m, hence $v_m \models p_u$. Furthermore, whenever r_{m+1} is strictly better$_u$ than r_m then either $i_u^m > i_u^{m+1}$ or $v_m \models q_u$. We conclude that the play is winning according to the pair $\langle p, q \rangle$. ☐

We show that the algorithm in Fig. 1 induces a good Rabin ranking. Thus, proving completeness of the Rabin ranking method.

Claim 7 *For every Rabin game there exists a good Rabin ranking such that for every state v winning for Eve according to the Rabin winning condition we have $r(v) \neq \infty$.*

Proof: Denote by W the set of states returned by the algorithm in Fig. 1. We show how to define a good Rabin ranking on the states in W. In order to define the ranking we analyze the way the computation advances. The analysis is similar to the analysis of the fixpoint computation in [Wal01]. Formally, we have the following.

In every stage of the computation we record the status of the call stack. According to the contents of the call stack we define sets of states whose union includes all the states in W. We then use these sets to give ranks to the states in W. First, let us add a counter to the least fixpoints. We assume that with the minimal fixpoint there is a counter i. This counter is initialized to zero in the first visit to line 1 in the function `main_Rabin` and increased by one in every subsequent visit. Similarly, the counter is initialized to 0 in the first visit to line 6 in the function `Rabin` and increased by one in every subsequent visit.

Consider the state of the call stack when the computation reaches line 1 in function `main_Rabin`. We use the counter i to set Z_j to the value of Z in the iteration where i is incremented to j. It follows that in the first iteration when i is initialized to 0 we have $Z = \emptyset$ and we set $Z_0 = \emptyset$. Furthermore, $Z_{i+1} =$ `Rabin(Set,true,cpred(`Z_i`))`.

We monitor the call stack if every copy of `Rabin` on the call stack is found in the last iteration of the maximal fixpoint. That is, the value of Y (in each copy) is already the value computed by the next iteration. In what follows, every configuration of the call stack is assumed to be in such a state. Consider a configuration of the call stack where the active copy of `Rabin` is in line 6. Let us denote the number of copies of the function `Rabin` on the call stack by l. Let $j_1 \cdots j_l$ be the pairs of the acceptance condition handled by these copies of `Rabin` and let $i_0 \cdots i_l$ be the values of the counter i (where i_0 is the counter in the function `main_Rabin`). We set $X^{i_0 \cdots i_l}_{j_1 \cdots j_l}$ to be the value of X in the active copy of `Rabin` in this state of the call stack. Again, whenever i_l is 0 we have $X^{i_0 \cdots i_l}_{j_1 \cdots j_l}$ is the empty set.

Consider a tuple $i_0 \cdots i_l$ and a prefix of a permutation $j_1 \cdots j_l$. From the structure of the fixpoint it follows that $X^{i_0 \cdots (i_l+1)}_{j_1 \cdots j_l}$ is exactly the union of $X^{i_0 \cdots i_l i}_{j_1 \cdots j_l j}$ for every value of $j \notin \{j_1, \ldots, j_l\}$ and i.

For every state $v \in W$ there exists $d \in D_{_R}$ such that $v \in X^{m(d)}_{\pi(d)}$ and d is minimal according to the ordering on $D_{_R}$. We set $r(v)$ to be this minimal value d. For all states $v \notin W$ we set $r(v) = \infty$. We show that the resulting ranking is a good Rabin ranking.

Consider a state $v \in W$. Let $r(v) = i_0 j_1 i_1 \cdots j_k i_k$. Consider the call stack of the computation at the point where $X^{i_0 \cdots i_k}_{j_1 \cdots j_k}$ is computed. Let $Y_{j_1 j_2 \cdots j_l}$ denote the value of Y in the lth copy of `Rabin` on the call stack (counting from the bottom of the stack). Notice, that we do not have to annotate Y by $i_0 i_1 \cdots i_{l-1}$ as we are considering only the specific rank $r(v)$. From the structure of the fixpoint it follows that $Y_{j_1 \cdots j_l}$ is exactly the union of $X^{i_0 \cdots i_l i}_{j_1 \cdots j_l j}$ for all possible values of i and $j \notin \{j_1 \cdots j_l\}$.

By definition we have the following.

$$
\begin{array}{ll}
X^{i_0\cdots i_k}_{j_1\cdots j_k} = & \\
\bigcirc\!\!\!\!\diamond\, Z_{i_0-1} & \vee \\
\left(p_{j_1} \wedge \bigcirc\!\!\!\!\diamond\, X^{i_0(i_1-1)}_{j_1}\right) \vee (p_{j-1} \wedge q_{j-1} \wedge \bigcirc\!\!\!\!\diamond\, Y_{j_1}) & \vee \\
\left(p_{j_1} \wedge p_{j_2} \wedge \bigcirc\!\!\!\!\diamond\, X^{i_0 i_1 (i_2-1)}_{j_1 j_2}\right) \vee (p_{j_1} \wedge p_{j_2} \wedge q_{j_2} \wedge \bigcirc\!\!\!\!\diamond\, Y_{j_1 j_2}) & \vee \\
\vdots & \\
\vdots & \\
\vdots & \\
\left(\left(\bigwedge_{l=1}^{k-1} p_{j_l}\right) \wedge \bigcirc\!\!\!\!\diamond\, X^{i_0\cdots(i_{k-1}-1)}_{j_1\cdots j_{k-1}}\right) \vee \left(\left(\bigwedge_{l=1}^{k-1} p_{j_l}\right) \wedge q_{j_{k-1}} \wedge \bigcirc\!\!\!\!\diamond\, Y_{j_1\cdots j_{k-1}}\right) & \vee \\
\left(\left(\bigwedge_{l=1}^{k} p_{j_l}\right) \wedge \bigcirc\!\!\!\!\diamond\, X^{i_0\cdots(i_k-1)}_{j_1\cdots j_k}\right) \vee \left(\left(\bigwedge_{l=1}^{k} p_{j_l}\right) \wedge q_{j_k} \wedge \bigcirc\!\!\!\!\diamond\, Y_{j_1\cdots j_k}\right) &
\end{array}
$$

Consider some $v \in X^{i_0\cdots i_k}_{j_1\cdots j_k}$. If v is in the first disjunct $\bigcirc\!\!\!\!\diamond\, Z_{i_0-1}$ then $best(v)$ is better than $r(v)$ (without checking better$_1$).

If v is in the $a+1$th disjunct

$$\left(\left(\bigwedge_{l=1}^{a} p_{j_l}\right) \wedge \bigcirc\!\!\!\!\diamond\, X^{i_0\cdots(i_a-1)}_{j_1\cdots j_a}\right) \vee \left(\left(\bigwedge_{l=1}^{a} p_{j_l}\right) \wedge \bigcirc\!\!\!\!\diamond\, Y_{j_1\cdots j_a}\right)$$

and v is not in all the disjuncts below it then $best(v)$ is better$_{a'}$ from $r(v)$ for all $a' < a$ (but not strictly better$_{a'}$). This follows from $i_0 \cdots i'_a$ being equivalent and from $p_1 \wedge \cdots \wedge p_a$ holding in v. It is also the case that $best(v)$ is strictly better$_a$ than $r(v)$. If $v \in (\bigwedge_{l=1}^{a} p_{j_l}) \wedge \bigcirc\!\!\!\!\diamond\, X^{i_0\cdots i_a}_{j_1\cdots j_a}$ the ath coordinate of the ranking decreases. If $v \in (\bigwedge_{l=1}^{a} p_{j_l}) \wedge q_{j_l} \bigcirc\!\!\!\!\diamond\, Y_{j_1\cdots j_a}$ then v is a q_{j_l} state.

We conclude that from v Eve can control the game so that the successor of v is better than v. $\square$

4.2 Streett Ranking

Consider a game $G = \langle V, E, \overline{\alpha}\rangle$ where $\alpha = \{\langle p_1, q_1\rangle, \ldots, \langle p_k, q_k\rangle$ is a Streett winning condition. We recall that in our notation, the Rabin condition α is not the dual of the Streett condition α. Adam wins an infinite play if it satisfies $\bigwedge_{i=1}^{k}(\Diamond \Box\, p \vee \Box \Diamond\, q)$.

The *Streett domain* for α over V is $[0..n]^k \cup \{\infty\}$, denoted by $D_S(\alpha, V)$. We order $D_S(\alpha, V)$ according to the lexicographic order with ∞ as maximal element. Given $m \in D_S(\alpha, V)$ we denote by m_l the lth entry in m. Consider the set $\Pi(k)$. Let $\pi = j_1 \cdots j_k \in \Pi(k)$ be some permutation. We define what does it mean to increase the lth entry in π. We increase the lth entry by leaving the first $l-1$ entries unchanged. For the lth entry we choose the next available value among the rest of the entries. If the lth entry is already the maximal among these entries then we go back to the minimal. The rest are ordered in increasing order. Let $\pi = j_1 \cdots j_k$. We set $inc_l(\pi)$ to be the permutation $j_1 \cdots j_{l-1} j'_l \cdots j'_k$ such that if $j_l = max(j_l, \ldots j_k)$ then $j'_l = min(j_l, \cdots, j_k)$ and if $j_l < max(j_l, \ldots, j_k)$ then j'_l is set to the minimal value in $j_l, \ldots, j_k$ such that $j'_l > j_l$. Then, we order $\{j_l, \ldots, j_k\} - \{j'_l\}$ in increasing order and this completes the permutation. For example, $inc_k(\pi)$ is π, $inc_1(123)$ is 213, and $inc_2(123)$ is 132. For simplicity of notations we write $D_{_S}$ instead of $D_{_S}(\alpha, V)$ and Π instead of $\Pi(k)$.

A *Streett ranking* over V is $r : V \times \Pi \to D_{_S}$. That is, with every state $v \in V$ and every permutation $\pi \in \Pi$ we associate a rank in $D_{_S}$. For every state v and permutation π, we denote by $best(v, \pi)$ the rank of the minimal successor of v in case that $v \in V_{_A}$ or the rank of the maximal successor of V in case that $v \in V_{_E}$. Formally,

$$best(v, \pi) = \begin{cases} min_{(v,w)\in E}(r(w, \pi)) & v \in V_{_A} \\ max_{(v,w)\in E}(r(w, \pi)) & v \in V_{_E} \end{cases}$$

We say that a Streett ranking is *good* if for every state v and $\pi \in \Pi$ such that $r(v, \pi) \neq \infty$ we have $best(v, \pi)$ is *better* than $r(v, \pi)$. Let $\pi = j_1 \cdots j_k$, $r(v, \pi) = i_1 \cdots i_k$, and $best(v, \pi) = i'_1 \cdots i'_k$. We say that $best(v, \pi)$ is *better* than $r(v, \pi)$ if it is *better*$_1$ than $r(v, \pi)$. We say that $best(v, \pi)$ is *better*$_l$ than $r(v, \pi)$ if one of the following holds.

- $i_l > i'_l$.
- $v \models q_{j_l}$ and $best(v, inc_l(\pi)) \neq \infty$.
- $i_l = i'_l$, $v \models p_{j_l}$, and $best(v, \pi)$ is better$_{l+1}$ than $r(v, \pi)$.

We say that $best(v, \pi)$ is *better*$_{k+1}$ than $r(v, \pi)$ if $best(v, \pi) \neq \infty$. It is simple to see that if $v \in V_{_E}$ and $best(v, \pi)$ is better than $r(v, \pi)$ then for every node w such that $(v, w) \in E$ we have $r(w, \pi)$ is better than $r(v, \pi)$.

We show that a good Streett ranking induces a winning strategy for Adam.

Claim 8 *Given a good Streett ranking r, Adam wins the Streett game from every state v such that for some permutation $\pi \in \Pi$ we have $r(v, \pi) \neq \infty$.*

We show that the algorithm in Fig. 2 induces a good Streett ranking. Thus, proving completeness of the Streett ranking method.

Claim 9 *For every Streett game there exists a good Streett ranking such that for every state v winning for Adam according to the Streett winning condition there exists a permutation π such that $r(v, \pi) \neq \infty$.*

Proof: Denote by W the set of states returned by the algorithm in Fig. 2. We show how to define a good Streett ranking on the states in W. In order to define the ranking we analyze the way the computation advances. The analysis is similar to the analysis of the fixpoint computation in [Wal01]. Formally, we have the following.

In every stage of the computation we record the status of the call stack. According to the contents of the call stack we define sets of states that include all the states in W. We then use these sets to give ranks to the states in W. First, let us add a counter to the least fixpoints. We assume that with the minimal fixpoint there is a counter i. This counter is initialized to zero in the first visit to line 5 in the function `Streett` and increased by one in every subsequent visit.

We monitor the call stack if every copy of `Streett` on the call stack is found in the last iteration of the maximal fixpoint. That is, the value of Z (in each copy) is already the value computed by the next iteration. In what follows, every configuration of the call stack is assumed to be in such a state. Consider a configuration of the call stack where the active copy of `Streett` is in line 5. Let us denote the number of copies of the function `Streett` on the call stack by l. Let $j_1 \cdots j_l$ be the pairs of the acceptance condition handled by these copies of `Streett` and let $i_1 \cdots i_l$ be the values of the counter i. We set $Y^{i_1 \cdots i_l}_{j_1 \cdots j_l}$ to be the value of Y in the active copy of `Streett` in this state of the call stack. It follows that whenever i_l is 0 we have $Y^{i_1 \cdots i_l}_{j_1 \cdots j_l}$ is the empty set.

Consider a tuple $i_0 \cdots i_l$ and a prefix of a permutation $j_1 \cdots j_l$. From the structure of the fixpoint it follows that for every value $j \notin \{j_1, \ldots, j_l\}$ we have $Y^{i_0 \cdots (i_l+1)}_{j_1 \cdots j_l}$ is exactly the union of $Y^{i_0 \cdots i_l i}_{j_1 \cdots j_l j}$ for every possible value i.

For every state $v \in W$ and every permutation $\pi \in \Pi$ such that there exists $d \in D_S$ such that $v \in Y^d_\pi$, we set $r(v, \pi)$ to be the minimal such value d. From the definition of the fixpoint for every value $v \in W$ there exists at least one such permutation $\pi \in \Pi$. For all states $v \notin W$ and forall permutations $\pi \in \Pi$ we set $r(v, \pi) = \infty$. We show that the resulting ranking is a good Streett ranking.

Consider a state $v \in W$ and some permutation π such that $r(v, \pi) < \infty$. Let $\pi = j_1 \cdots j_k$ and $r(v, \pi) = i_1 \cdots i_k$. Consider the call stack of the computation at the point where $Y^{r(v,\pi)}_\pi$ is computed. Let $Z_{j_1 \cdots j_l}$ denote the value of Z in the $l + 1$th copy of `Streett` on the call stack (counting from the bottom of the stack). Notice that we do not have to annotate $Z_{j_1 \cdots j_l}$ by $i_1 \cdots i_l$ as we are considering only the specific rank $r(v, \pi)$. Notice as well that according to this notation Z is the winning set computed by the algorithm. Let $X_{j_1 \cdots j_k}$ denote the value of X returned by the function `minor_Streett`. From the structure of the fixpoint it follows that for every $j \notin \{j_1 \cdots j_l\}$ we have $Z_{j_1 \cdots j_l}$ is exactly the union of $Y^{i_1 \cdots i_l i}_{j_1 \cdots j_l j}$ for all possible values of i.

By definition we have the following.

$$
\begin{array}{ll}
Y^{i_1 \cdots i_k}_{j_1 \cdots j_k} = & \\
\quad (q_{j_1} \wedge \bigcirc Z) \vee \left(\bigcirc Y^{(i_1-1)}_{j_1} \right) & \vee \\
\quad (p_{j_1} \wedge q_{j_2} \wedge \bigcirc Z_{j_1}) \vee \left(p_{j_1} \wedge \bigcirc Y^{i_1 \cdots (i_2-1)}_{j_1 j_2} \right) & \vee \\
\quad \vdots & \\
\quad \left(\left(\bigwedge_{l=1}^{k-2} p_{j_l} \right) \wedge q_{j_{k-1}} \wedge \bigcirc Z_{j_1 \cdots j_{k-2}} \right) \vee \left(\left(\bigwedge_{l=1}^{k-2} p_{j_l} \right) \wedge \bigcirc Y^{i_1 \cdots (i_{k-1}-1)}_{j_1 \cdots j_{k-1}} \right) & \vee \\
\quad \left(\left(\bigwedge_{l=1}^{k-1} p_{j_l} \right) \wedge q_{j_k} \wedge \bigcirc Z_{j_1 \cdots j_{k-1}} \right) \vee \left(\left(\bigwedge_{l=1}^{k-1} p_{j_l} \right) \wedge \bigcirc Y^{i_1 \cdots (i_k-1)}_{j_1 \cdots j_k} \right) & \vee \\
\quad \left(\left(\bigwedge_{l=1}^{k} p_{j_l} \right) \wedge \bigcirc X_{j_1 \cdots j_k} \right) &
\end{array}
$$

Consider some $v \in Y^{i_1 \cdots i_k}_{j_1 \cdots j_k}$. Let $\pi = j_1 \cdots j_k$. If v is in the first disjunct $(q_{j_1} \wedge \bigcirc Z) \vee \bigcirc Y^{i_1-1}_{j_1}$ then $best(v, \pi)$ is better$_1$ than $r(v, \pi)$. If $v \in (q_{j_1} \wedge \bigcirc Z_{j_1})$ then v is a q_{j_1} state and as for every j we have Z is equal to $\bigcup_i Y^i_j$ it follows that $best(v, inc_1(\pi)) \neq \infty$.

If v is in the $a+1$th disjunct

$$\left(\left(\left(\bigwedge_{l=1}^{a} p_{j_l}\right) \wedge q_{j_{a+1}} \wedge \bigcirc Z_{j_1\cdots j_a}\right) \vee \left(\left(\bigwedge_{l=1}^{a} p_{j_l}\right) \wedge \bigcirc Y^{i_1\cdots(i_{a+1}-1)}_{j_1\cdots j_{a+1}}\right)\right)$$

and v is not in all the disjuncts above it then $best(v,\pi)$ is better$_{a'}$ from $r(v,\pi)$ for all $a' < a+1$. This follows from the $i_0\cdots i'_a$ being equivalent and from $p_1 \wedge \cdots \wedge p_{a'}$ holding in v. It is also the case that $best(v,\pi)$ is better$_{a+1}$ then $r(v,\pi)$. If $v \in (\bigwedge_{l=1}^{a} p_{j_l}) \wedge q_{j_{a+1}} \wedge \bigcirc Z_{j_1\cdots j_a}$ then v is a $q_{j_{a+1}}$ state. As for every $j \notin \{j_1 \cdots j_a\}$ we have $Z_{j_1\cdots j_a}$ is $\bigcup_i Y^{i_1\cdots i_a i}_{j_1\cdots j_a j}$ it follows that $best(v, inc_{a+1}(\pi)) \neq \infty$ and $best(v,\pi)$ is better$_{a+1}$ than $r(v,\pi)$. If $v \in (\bigwedge_{l=1}^{a} p_{j_l}) \wedge \bigcirc Y^{i_1\cdots(i_a-1)}_{j_1\cdots j_a}$ then the $a+1$th coordinate of $best(v,\pi)$ decreases. Finally, if v is in the $k+1$th disjunct $(\bigwedge_{l=1}^{k} p_{j_1}) \wedge \bigcirc X_{j_1\cdots j_k}$ then $best(v,\pi)$ is better$_{k+1}$ than $r(v,\pi)$.

We conclude that from v Adam can control the game so that the successor of v is better than v. □

5 Computing Ranks Explicitly

In this section we generalize Jurdzinski's explicit ranking computation of parity games to Rabin and Streett ranking [Jur00]. As in the case of parity, the minimal good ranking is a least fixpoint of a monotone operator on a complete lattice. By Knaster-Tarski theorem there exists a least good ranking and there exists a simple lifting algorithm to compute it. From previous section it follows that the least good ranking is defined on the winning region. Etessami et al. show exactly how to encode Jurdzinski's algorithm to get the stated time and space bounds [EWS01].

Consider the set of possible Rabin rankings $r : V \to D_R$. We say that $r_1 \sqsubseteq r_2$ if for every $v \in V$ we have $r_1(v) \leq r_2(v)$. The resulting structure is a complete lattice. We now define the lifting operator. Given a ranking $r : V \to D_R$ and a state $v \in V$ we set $prog(r,v)$ to be the least value $d \in D_R$ such that $best(v)$ is better than d. We define $lift(r,v)$ to be the following function.

$$lift(r,v)(u) = \begin{cases} r(u) & u \neq v \\ max\{r(u), prog(r,u)\} & u = v \end{cases}$$

Obviously, the operator $lift$ is monotone according to $\sqsubseteq$. Every good Rabin ranking r is a pre-fixpoint with respect to $lift(r,v)$ for all states $v \in V$ and every pre-fixpoint with respect to $lift(r,v)$ for all states $v \in V$ is a good Rabin ranking.

Similarly, consider the set of possible Streett rankings $r : V \times \Pi \to D_S$. We say that $r_1 \sqsubseteq r_2$ if for every $v \in V$ and every $\pi \in \Pi$ we have $r_1(v,\pi) \leq r_2(v,\pi)$. The resulting structure is a complete lattice. The Streett lifting operator is defined analogously to the above. Given a ranking $r : V \times \Pi \to D_S$, a state $v \in V$, and a permutation $\pi \in \Pi$ we set $prog(r,v,\pi)$ to be the least value $d \in D_S$ such that $best(v,\pi)$ is better than d. The ranking $lift(r,v,\pi)$ is the following ranking.

$$lift(r,v,\pi)(u) = \begin{cases} r(u,\pi') & u \neq v \text{ or } \pi \neq \pi' \\ max\{r(u), prog(r,u)\} & u = v \text{ and } \pi = \pi' \end{cases}$$

Again, the operator $lift$ is monotone according to $\sqsubseteq$. Every good Streett ranking r is a pre-fixpoint with respect to $lift(r, v, \pi)$ for all states $v \in V$ and permutations $\pi \in \Pi$ and every pre-fixpoint with respect to $lift(r, v, \pi)$ for all $v \in V$ and $\pi \in \Pi$ is a good Streett ranking.

By the Knaster-Tarski theorem the least pre-fixpoint (either for Streett or Rabin) exists and it can be computed by the algorithm in Fig. 3. Let $r0$ denote the following ranking. In the case of Rabin $r0$ is the ranking such that for every $v \in V$ we have $\pi(r(v)) = 12\cdots k$ and $m(r(v)) = 0\cdots 0$. In the case of Streett $r0$ is the ranking such that for every $v \in V$ and $\pi \in \Pi$ we have $r(v, \pi) = 0\cdots 0$.

```
RankingLifting
  Let r := r0;
  While (∃v,π s.t.  r ⊑ lift(r,v,π) and r ≠ lift(r,v,π))
    Let r := lift(r,v,π);
  End -- While(...)
End -- RankingLifting
```

Figure 3: The lifting algorithm.

The procedure in Fig. 3 misses most of the implementation details. A naïve approach to choosing the next $v \in V$ and $\pi \in \Pi$ for performing lifting can take $O(nk!)$ for one lift. Etessami et al. supplied the necessary details for the case of parity games with 3 winning conditions [EWS01]. The generalization of their implementation to the general case is not complicated.

Theorem 10 *Rabin and Streett games can be solved in time $O(mn^{k+1}kk!)$ time and $O(nk)$ space for Rabin and time $O(mn^k kk!)$ time and $O(nkk!)$ space for Streett where n is the number of states, m is the number of edges, and k is the number of pairs.*

Proof: We start with Rabin. The space required is $O(nk)$ as we have to store the ranking for each state $v \in V$ and an entry $d \in D_R$ requires $O(k)$ space. The lifting operator can work in time $O(k \cdot out{-}deg(v))$, where $out{-}deg(v)$ is the out-degree of v. Every state can be lifted at most $|D_R|$ times. The total run time is bounded by

$$O\left(\sum_{v \in V} k \cdot out{-}deg(v) \cdot |D_R|\right) = O(km|D_R|)$$

As $|D_R| = n^{k+1}k!$ the bound follows.

For the case of Streett, the space required is $O(nkk!)$ as we have to store a value $d \in D_S$ for each state $v \in V$ and every permutation $\pi \in \Pi$. An entry $d \in D_S$ requires $O(k)$ space. The lifting operator can work in time $O(k \cdot out{-}deg(v))$. Every state and permutation can be lifted at most $|D_S|$ times. The total run time is bounded by

$$O\left(\sum_{v \in V} \sum_{\pi \in \Pi} k \cdot out{-}deg(v) \cdot |D_S|\right) = O(kmk!|D_S|)$$

As $|D_S| = n^k$ the bound follows. □

As in Jurdzinski's original algorithm this algorithm cannot be applied symbolically (see Section 7). In some cases, this may be a serious restriction.

6 Fast Symbolic Computation

In this section we generalize the method of Long et al. for accelerating the evaluation of fixpoints [LBC+94]. Long et al. show that by maintaining the intermediate values of the fixpoint, they can use these values to start the computation of future fixpoints not from minimal or maximal values but rather from better approximations. They show that with these approximations the worst time complexity of the fixpoint computation is reduced to the square root of the original. Unfortunately, the memory consumption amounts to the other square root.

The acceleration works very similarly for Rabin and Streett games. We explain it here for the case of Rabin. The case of Streett is identical but for the order of the indices. Consider the algorithm in Fig. 1. We add a counter to each of the fixpoints. To each of the minimal fixpoints we add a counter i. It is initialized to 0 in the first visit to the command `LeastFix` and incremented by 1 in every subsequent visit. Similarly, to each of the maximal fixpoint we add a counter p. It is initialized to 0 in the first visit to the command `GreatestFix` and incremented by 1 in every subsequent visit. Consider an active copy of the function `Rabin` with $l-1$ copies of `Rabin` on the store. Suppose that the active copy of `Rabin` is found in line 4. Let $i_0 \cdots i_{l-1}$ be the values of the counters i associated with the least fixpoints in the copies of `Rabin` on the stack (where i_0 is the counter in the function `main_Rabin`). Let $p_1 \cdots p_l$ be the values of the counters p associated with the greatest fixpoints in the copies of `Rabin` on the stack. Let $j_1 \cdots j_l$ denote the number of pairs handled by the different copies of `Rabin`. We set $Y(i_0, \cdots i_{l-1}, p_1 \cdots p_l, j_1 \cdots j_l)$ to be the value of Y when the counter p is set to p_l. When the active copy of `Rabin` is found in line 6 Then the sequence $i_0 \cdots i_l$ includes also the value of the counter i in the active copy of `Rabin`. We set $X(i_0 \cdots i_l, p_1 \cdots p_l, j_1 \cdots j_l)$ to be the value of X when the counter i is set to i_l.

Given two sequences $\alpha = i_0 \cdots i_{l-1}$, $\beta = p_1 \cdots p_l$, and $\gamma = j_1 \cdots j_l$ and $\alpha' = i'_0 \cdots i'_{l-1}$, $\beta' = p'_1 \cdots p'_l$, and $\gamma' = j'_1 \cdots j'_l$ we say that $\alpha\beta\gamma <_\nu \alpha'\beta'\gamma'$ if $\alpha = \alpha'$, $\gamma = \gamma'$ and $\beta < \beta'$ according to the lexicographic order. Similarly, given $\alpha = i_0 \cdots i_l$, $\beta = p_1 \cdots p_l$, and $\gamma = j_1 \cdots j_l$ and $\alpha' = i'_0 \cdots i'_l$, $\beta' = p'_1 \cdots p'_l$, and $\gamma' = j'_1 \cdots j'_l$ we say that $\alpha\beta\gamma <_\mu \alpha'\beta'\gamma'$ if $\beta = \beta'$, $\gamma = \gamma'$ and $\alpha < \alpha'$ according to the lexicographic order. For a fixed $\alpha = i_0 \cdots i_{l-1}$ and $\gamma = j_1 \cdots j_l$, the ordering $<_\nu$ is a total order on l-tuples. Similarly, for a fixed $\beta = j_1 \cdots j_l$ and $\gamma = j_1 \cdots j_l$, the ordering $<_\mu$ is a total order on l-tuples.

For every $\alpha = i_0 \cdots i_{l-1}$, $\beta = p_0 \cdots p_{l-1}$, and $\gamma = j_1 \cdots j_l$ the maximal value p such that $Y(\alpha, \beta p, \gamma)$ is defined is a greatest fixpoint value. Long et al. show that $Y(\alpha, \beta p, \gamma)$ is contained in every set $Y(\alpha, \beta', \gamma)$ such that $\beta' < \beta p$. It follows that the computation of $Y(\alpha, \beta 0, \gamma)$ (which leads to the computation of $Y(\alpha, \beta p, \gamma)$) can start from the minimal set $Y(\alpha, \beta', \gamma)$ such that $\beta' < \beta 0$. Consider now the values of the inner-most greatest fixpoint. That is, the values $Y(\alpha, \beta, \gamma)$ where $|\beta| = k$. It follows that for every value of α and γ there are at most n different values for $Y(\alpha, \beta, \gamma)$.

Dually, for every $\alpha = i_0 \cdots i_{l-1}$, $\beta = p_0 \cdots p_l$, and $\gamma = j_1 \cdots j_l$ the maximal value i such that $X(\alpha i, \beta, \gamma)$ is defined is a least fixpoint value. Long et al. show that $X(\alpha i, \beta, \gamma)$ contains every set $X(\alpha', \beta, \gamma)$ such that $\alpha' < \alpha i$. It follows that the computation of $X(\alpha 0, \beta, \gamma)$ (which leads to the computation of $X(\alpha i, \beta, \gamma)$ can start from the maximal set $X(\alpha', \beta, \gamma)$ such that $\alpha' < \alpha 0$. Consider now the values of the inner-most least fixpoint. That is, the values $X(\alpha,\beta,\gamma)$ where $|\alpha| = k$. If follows that for every value of β and γ there are at most n different values for $X(\alpha,\beta,\gamma)$.

The computation of the inner-most least fixpoint dominates the computation time. It follows that the computation can be concluded in time $O(n^{k+1}k!)$. However, we have to store the Y values for every possible value of α and γ. Notice, that the β values are implicit in every point of the computation. We just have to store the best value for α and γ. Similarly, we store the X values for every possible value of β and γ. Thus, the memory required by the algorithm is $O(n^{k+1}k!)$. Formally, we have the following.

Theorem 11 *Rabin and Streett games can be solved in time $O(n^{k+1}k!)$ and space $O(n^{k+1}k!)$ where n is the number of states and k is the number of pairs of the winning condition.*

On the one hand, the space complexity of the algorithm makes it prohibitively expensive. Implementing an efficient memory system that supports this algorithm makes it less attractive in practice. On the other hand, if we want to use the intermediate fixpoint values for construction of the winning strategy then memorizing some of the intermediate values is necessary anyway.

7 Conclusions

We present recursive fixpoint algorithms for solving Rabin and Streett games. Using these algorithms we can compute the winning region of Rabin (respectively Streett) in time $O(mn^{2k}k!)$ using a symbolic algorithm. This matches the best known upper bounds for Streett and Rabin [KV98, Hor05]. We use these algorithms to two ends.

First, we show how to define direct Rabin and Streett ranking. Our ranking are a complete and sound way to characterize the winning sets in the respective games. Much like in parity games, computing the ranking directly is much more efficient than computing the fixpoint [Jur00]. We can solve Rabin games in time $O(mn^{k+1}kk!)$ and space $O(nk)$ where n is the number of states, m the number of transitions, and k the number of pairs. We can solve Streett games in time $O(mn^{k}kk!)$ and space $O(nkk!)$.

Our second contribution again follows a similar contribution for parity games. Long et al. show that by keeping intermediate values of the fixpoint computation, parity games can be solved more efficiently [LBC^{+}94]. We show that by keeping intermediate values of the fixpoint computation, we can reduce computation time to $O(n^{k+1}k!)$ at the price of using $O(n^{k+1}k!)$ space.

Both the enumerative and symbolic algorithms are borrowed from algorithms for solving parity games. This raises the question whether we can adapt the strategy improvement technique [VJ00] as well as the subexponential algorithm to solve parity games [JPZ05] to Rabin and Streett games. We conjecture that every solution to parity games that works in time $t(m,n,k)$ can be generalized to solve Rabin and Streett games in time $k!t(m,n,2k)$ (recall that a Rabin / Streett game is converted to a parity game with $2k$ priorities).

We mentioned that the direct rank computation cannot be implemented symbolically. This is similar to Jurdzinski's algorithm [Jur00]. Bustan et al. suggested to use

Algebraic Decision Diagrams (ADDs) to represent Jurdzinski's ranking symbolically [BKV04]. We cannot say whether this would be applicable in our case as well.

References

[BKV04] D. Bustan, O. Kupferman, and M.Y. Vardi. A measured collapse of the modal μ-calculus alternation hierarchy. In *21st STACS*, LNCS 2996, 522–533. 2004.

[BL69] J.R. Büchi and L.H. Landweber. Solving sequential conditions by finite-state strategies. *Trans. AMS*, 138:295–311, 1969.

[BSV03] H. Björklund, S. Sandberg, and S. Vorobyov. A discrete subexponential algorithm for parity games. In *20th STACS*, LNCS 2607, 663–674. Springer-Verlag, 2003.

[Chu63] A. Church. Logic, arithmetic and automata. In *1962 Int. Congr. Math.*, 23–25, 1963.

[DJW97] S. Dziembowski, M. Jurdzinski, and I. Walukiewicz. How much memory is needed to win infinite games. In *12th LICS*, 99–110, 1997.

[EJ88] E.A. Emerson and C.S. Jutla. The complexity of tree automata and logic of programs. In *29th FOCS*, 328–337, 1988.

[EJ91] E.A. Emerson and C. Jutla. Tree automata, μ-calculus and determinacy. In *32nd FOCS*, 368–377, 1991.

[EL86] E. A. Emerson and C. L. Lei. Efficient model-checking in fragments of the propositional modal μ-calculus. In *1st LOCS*, 267–278, 1986.

[Eme97] E.A. Emerson. Model checking and the μ-calculus. In *Descriptive Complexity and Finite Models*, 185–214. AMS, 1997.

[EWS01] K. Etessami, Th. Wilke, and R. A. Schuller. Fair simulation relations, parity games, and state space reduction for Büchi automata. In *28th ICALP*, LNCS 2076, 2001.

[GH82] Y. Gurevich and L.A. Harrington. Automata, trees and games. In *14th STOC*, 1982.

[Hor05] F. Horn. Streett games on finite graphs. In *2nd GDV*, 2005.

[JPZ05] M. Jurdzinski, M. Paterson, and U. Zwick. A deterministic subexponential algorithm for solving parity games. Manuscript, 2005.

[Jur00] M. Jurdzinski. Small progress measures for solving parity games. In *17th STACS*, LNCS 1770, 290–301. Springer-Verlag, 2000.

[Koz83] D. Kozen. Results on the propositional μ-calculus. *TCS*, 27:333–354, 1983.

[KV98] O. Kupferman and M.Y. Vardi. Weak alternating automata and tree automata emptiness. In *30th STOC*, 224–233, Dallas, 1998.

[KV01] O. Kupferman and M.Y. Vardi. Weak alternating automata are not that weak. *ACM ToCL*, 2001(2):408–429, July 2001.

[LBC$^+$94] D. Long, A. Brown, E. Clarke, S. Jha, and W. Marrero. An improved algorithm for the evaluation of fixpoint expressions. In *6th CAV*, LNCS 818, 338–350. 1994.

[PR89] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *16th POPL*, 179–190, 1989.

[Rab69] M.O. Rabin. Decidability of second order theories and automata on infinite trees. *Trans. AMS*, 141:1–35, 1969.

[Rab72] M.O. Rabin. *Automata on Infinite Objects and Churc's Problem*, volume 13 of *Regional Conference Series in Mathematics*. AMS, 1972.

[Saf88] S. Safra. On the complexity of ω-automata. In *29th FOCS*, 319–327, 1988.

[Str82] R.S. Streett. Propositional dynamic logic of looping and converse is elementarily decidable. *IC*, 54:121–141, 1982.

[Tho90] W. Thomas. Automata on infinite objects. In *Handbook of TCS*, volume B, pages 165–191. 1990.

[VJ00] J. Voge and M. Jurdzinski. A discrete strategy improvement algorithm for solving parity games. In *12th CAV*, LNCS 1855, 202–215. Springer-Verlag, 2000.

[Wal01] I. Walukiewicz. Pushdown processes: Games and model-checking. *IC*, 164, 2001.

[Wil01] T. Wilke. Alternating tree automata, parity games, and modal μ-calculus. *Bull. Soc. Math. Belg.*, 8(2), May 2001.