Strategy Construction in Infinite Games with Streett and Rabin Chain Winning Conditions

Nils Buhrke[*]
Helmut Lescow[*]
Jens Vöge[*]

ABSTRACT We consider finite-state games as a model of nonterminating reactive computations. A natural type of specification is given by games with Streett winning condition (corresponding to automata accepting by conjunctions of fairness conditions). We present an algorithm which solves the problem of program synthesis for these specifications. We proceed in two steps: First, we give a reduction of Streett automata to automata with the Rabin chain (or parity) acceptance condition. Secondly, we develop an inductive strategy construction over Rabin chain automata which yields finite automata that realize winning strategies. For the step from Rabin chain games to winning strategies examples are discussed, based on an implementation of the algorithm.

1 Introduction

In recent years, methods of automatic verification for finite–state programs have been applied successfully, which have clearly reached the level of practical use. For the existing automata theoretic results on finite–state program synthesis the situation is quite different. Not only explicit formulations of algorithms but also experience in nontrivial examples are missing. The present paper offers an algorithm for finite–state program synthesis from automaton specifications; we also discuss some examples based on an implementation.

The foundations of finite–state program synthesis were laid in the work of Büchi and Landweber [BL69]. Papers exploiting further the approach include Pnueli, Rosner [PR89], Abadi, Lamport, Wolper [ALW89], and Nerode, Yakhnis, Yakhnis [NYY92]. Recently, the paradigm of program synthesis has attracted increasing attention in the context of discrete event

[*]Institut für Informatik und Praktische Mathematik, Christian-Albrechts-Universität Kiel, D-24098 Kiel, email: {nb, hel, jv}@informatik.uni-kiel.de. Supported by Deutsche Forschungsgemeinschaft, projects Th 352/3-2 and Th 352/5-1.

and discrete control systems ([RW89], [KG95], [AMP95]). Unfortunately, however, these papers do not present directly implementable synthesis algorithms; and also the original Büchi-Landweber paper, while describing an algorithm, is not feasible.

We adopt the following game theoretic framework (see [Tho95] for more detailed background). The interaction between two parties (say, a program and its environment) is modeled by two players, called 0 and 1 here, of an "infinite game" (or "Gale-Stewart game" [GS53]). In a play of the game, both parties perform actions in turn, thus building up an infinite computation. In the state-based description to be assumed in this paper, an action causes a transition in a state graph, and referring to the alternation between the two players, we suppose that this state graph is partitioned into two sets such that the transition relation induces a bipartite graph with respect to these two sets. The two players' actions can then be viewed as movements of a token through this "game graph" along the graph edges. By conditions on the visited (or infinitely often visited) states it can be specified which computations ("plays") cause a win of player 0. As we identify player 0 with the party "program" and player 1 with the "environment", this winning condition for player 0 defines the set of "desired computations" as given by the specification.

The synthesis problem for these games asks for programs that realize winning strategies for player 0, i.e. allow player 0 to choose transitions that ensure his win whatever player 1 does. Büchi and Landweber [BL69] showed that one can effectively compute the set of states from where (as start of a play) player 0 wins, and that such a winning strategy is always realizable by a finite automaton.

In applications (e.g. in problems of discrete control) a natural type of winning condition is the so-called "Streett form" [Str82], i.e. a conjunction of conditions of the form

> "if state set U_k is visited infinitely often then state set L_k is visited infinitely often".

Short:

$$(*) \quad \bigwedge_{1 \leq k \leq r} (Inf(\pi) \cap U_k \neq \emptyset \Rightarrow Inf(\pi) \cap L_k \neq \emptyset).$$

Finite game graphs with such winning conditions (for player 0) can be viewed as finite Streett automata (in the sense of ω-automata theory [Tho90]), known to characterize the regular ω-languages.

The computational difficulties in constructing winning strategies for such games are due to two phenomena: First the size of memory needed in winning strategies, secondly the recursive descent by which the given game graph is reduced when a strategy is constructed.

In this paper we develop an algorithm for strategy synthesis in Streett games which proceeds in two steps (by which the above mentioned phenomena are handled separately). First, we transform Streett games to games with "Rabin chain winning condition" (or "parity winning condition" [Mos91]). The blow–up of the game graphs in this step corresponds to the introduction of sufficient memory for winning strategies. The second step is the construction of memory–less winning strategies for these Rabin chain games; it involves a decomposition of the game graph. Other approaches might merge these two aspects (e.g. following [McN93], [TW94], [YY93]), but for experiments it seems useful to be able to study the two effects separately.

The first step has some resemblance to the reduction of games with the Muller winning condition to the Rabin chain condition (see e.g. Emerson, Jutla [EJ91], Thomas [Tho95]). We use a new version of the "latest appearance record". This data structure is hidden already in Rabin's paper [Rab69, lemma 3.8], and appeared in many forms (Gurevich, Harrington [GH82], Büchi [Büc83], Muchnik [Muc92], Safra [Saf92], Muller, Schupp [MS95]). We use a form, called "index appearance record" here, where in a Streett game with winning condition $(*)$ the indices of referenced sets L_k in the order of their last visits are kept in a record, together with two pointers. This is similar to Büchi's "order vector with hit position" [Büc83]. As a result we obtain a rather simple transformation with the (previously known) time complexity polynomial in the number of states but exponential in the number of pairs.

The actual synthesis algorithm starting from a game with Rabin chain or parity winning condition involves an induction (over the cardinality of the state space of the game) following the approach of [McN93, section 3] and [Tho95]. The time complexity of this synthesis problem is still open; our algorithm has an exponential upper bound but seems to allow practical use.

The paper is structured as follows: Section 2 introduces the terminology, section 3 the transition from Streett games to Rabin chain games, and section 4 presents the synthesis algorithm. In the final section we discuss the synthesis algorithm in a small example. The algorithm is implemented: more experiments with examples seem necessary to find appropriate heuristics for choosing pivot states which help to minimize backtracking in the synthesis algorithm.

We thank Wolfgang Thomas for many helpful discussions during the preparation of this paper.

2 Definitions and Notation

In the context of infinite games, it proved to be useful to consider a type of "ω-automaton" introduced by McNaughton [McN93] where we abstract from the labels of transitions, since the acceptance (or winning) condition depends only on visits of states; moreover, each state is associated with just the player whose turn it is to make the next move.

So a *finite-state game* is given by a bipartite finite directed graph G and a "winning condition". The idea is that two players, called player 0 and player 1, are moving a token alternatively from vertex to vertex along edges in the game graph. A game graph is of the form $G = (Q, Q_0, Q_1, E)$ with $Q = Q_0 \dot{\cup} Q_1$ and $E \subseteq (Q_0 \times Q_1) \cup (Q_1 \times Q_0)$ where Q_i is the set of vertices where it is the turn of player i to move the token. Since we are interested only in infinite computations it is convenient to demand that each vertex of the game graph has at least one outgoing edge.

A *play* π is an infinite sequence of states from Q visited by the token in successive moves, i.e., a sequence $\pi \in Q^\omega$ with $(\pi(t), \pi(t+1)) \in E$ for all t. The winner of a play is declared by a *winning (or accepting) condition* $Win : Q^\omega \rightarrow \{0, 1\}$ that maps a play π to i iff the play π is won by player i. So a game Γ is given by a pair $\Gamma = (G, Win)$. In the sequel, we also use the term "state" for "vertex".

In this paper we deal with Rabin games and Streett games (referring to the analogous acceptance conditions for ω-automata). These games are characterized by special winning conditions. A *Rabin condition* is given by a system $\Omega \subseteq 2^Q \times 2^Q$, and the function Win defined by $Win(\pi) = 0$ iff

$$\exists (L, U) \in \Omega \quad \text{s.t.} \quad Inf(\pi) \cap L = \emptyset \ \wedge Inf(\pi) \cap U \neq \emptyset$$

where $Inf(\pi)$ is the set of vertices visited infinitely often during the play π. A Rabin game is called *Rabin chain game* if the set of accepting pairs Ω consists of pairs (E_k, F_k) that form a chain by set inclusion: $E_1 \subseteq F_1 \subseteq E_2 \subseteq \ldots \subseteq F_r$ (cf. [Mos91]).

The *Streett condition* (following [Str82]) is dual to the Rabin condition in the sense that a play π is won by player 0 in a Rabin game iff π is accepted for player 1 in the corresponding Streett game. So the Streett acceptance condition is given by a system $\Omega \subseteq 2^Q \times 2^Q$ where the function Win is defined by $Win(\pi) = 0$ iff

$$\forall (L, U) \in \Omega : Inf(\pi) \cap U \neq \emptyset \ \Rightarrow Inf(\pi) \cap L \neq \emptyset.$$

A *strategy for player i* in the game Γ is a function which associates with a partial play ending in Q_i a state in Q_{1-i}. W.l.o.g. a strategy may be defined as a partial function $\sigma : Q^* Q_i \rightarrow Q_{1-i}$ with $\sigma(p_1, \ldots, p_k) = p_{k+1}$ such that $(p_k, p_{k+1}) \in E$. The strategy σ is called a *winning strategy* if, for any choice of moves of player $1 - i$, it induces a play won by player i.

Büchi and Landweber showed that finite automata (with fixed finite memory M) are sufficient to realize winning strategies for games on finite graphs (cf. [BL69]). Due to this a strategy for player i can be defined, using a finite (memory) set M, as a function $\sigma : Q_i \times M \rightarrow Q_{1-i}$ and using a function $\varphi : Q \times M \rightarrow M$ for updating the memory with each move of the token.

For a finite game graph $G = (Q, Q_0, Q_1, E)$ we call $\hat{G} = (\hat{Q}, \hat{Q}_0, \hat{Q}_1, \hat{E})$ a *game-extension by memory* if there is a finite set M and a memory-update function $\varphi : Q \times M \to M$, s.t. $\hat{G}$ is characterized by M and φ as follows

$$\begin{array}{rrcl} (i) & \hat{Q} & = & Q \times M, \\ (ii) & \hat{Q}_i & = & Q_i \times M, \text{ for all } i \in \{0,1\}, \\ (iii) & \hat{E} & = & \{((q,m),(q',m')) : (q,q') \in E \text{ and } \varphi(q',m) = m'\}. \end{array}$$

The game $\Gamma = (G, Win)$ may be simulated by the game $\hat{\Gamma} = (\hat{G}, \widehat{Win})$ when we define $\widehat{Win}(\hat{\pi}) := Win(Pr_1(\hat{\pi}))$ for all $\hat{\pi} \in \hat{Q}^\omega$, where Pr_j is the projection to the j-th components of an ω-sequence of tuples.

Thus, a strategy is describable by an extended game graph where all edges from states in Q_i not of the form $\big((q,m),\ (\sigma(q,m),\varphi(q,m))\big)$ are deleted. We call a strategy for player i *no-memory* if no additional memory is required, i.e. the strategy is a function $Q_i \to Q_{1-i}$. It can be described by the game graph where all but one outgoing edges from states in Q_i are deleted.

3 From Streett games to Rabin chain games

Between the games given in normal form described by Büchi [Büc83, §5] and the games with Rabin chain condition introduced by Mostowski [Mos91] is a close relation. If these games are played on finite graphs the winning strategies may be given as subgraphs on the game graph, which are called no-memory strategies. Büchi [Büc83, §12] showed how to transform Muller games into games in normal form. The same technique is presented (in simpler terminology and in the simpler case of finite-state games) in Thomas [Tho95] for constructing Rabin chain games out of Muller games. In both cases the memory needed for constructing winning strategies in the main part consists of the well known "latest appearance record" or "order vector".

For building Rabin chain games out of Streett games in our case we extend the "index appearance record", introduced by Muller and Schupp [MS95], by two pointers as done by Safra [Saf92]. For this transformation we present here an easily implementable algorithm of time complexity polynomial in the size of states and exponential in the number of Streett conjunctions. The same can be done for Rabin games.

Let us introduce some conventions to talk about the properties of winning plays in a Streett game. Usually, we assume the following situation:

(*) Let $\Gamma = (G, Win)$ be a Streett game defined over a finite game graph $G = (Q, Q_0, Q_1, E)$ and a finite sequence of Streett pairs $\Omega = (L_k, U_k)_{k \in \{1,\dots,r\}}$ with $L_k, U_k \subseteq Q$, for $k \in \{1, \dots, r\}$, and $L_r = U_r = Q$, s.t. for all plays $\pi \in Q^\omega$ we have $Win(\pi) = 0$ iff for all $k \in \{1, \dots, r\}$ the following condition holds: $Inf(\pi) \cap U_k \neq \emptyset$ implies $Inf(\pi) \cap L_k \neq \emptyset$.

For a given play $\pi \in Q^\omega$ we say that player 0 *reaches* an index $k \in \{1, \dots, r\}$ in the play π at position t if $\pi(t) \in L_k$, and player 1 reaches the index k in the play π at t if $\pi(t) \in U_k$. An index $k \in \{1, \dots, r\}$ reached infinitely often in a play $\pi \in Q^\omega$ by player $i \in \{0, 1\}$ is called *frequent* for player i on the play π.

It is quite clear that for a play $\pi \in Q^\omega$, player 0 wins iff the indices that are frequent for player 1 are also frequent for player 0 on the play π, formally

$$\{1 \leq k \leq r : Inf(\pi) \cap U_k \neq \emptyset\} \subseteq \{1 \leq k \leq r : Inf(\pi) \cap L_k \neq \emptyset\}.$$

In a play of a Streett game as in (*) we associate to each position an *index appearance record* (iar) consisting of permutations. The record at a position is given by shifting the indices reached for player 0 in the previous record to the right keeping the remainder in its previous ordering. A start record for the first position is defined. As an auxiliary notion we need a so–called shift function.

Definition 1 *Assume (*). For each $q \in Q$ and a permutation $iar = (i_1, i_2, \dots, i_r)$ of $\{1, \dots, r\}$, let $L(iar, q)$ be the subsequence of iar consisting of those i_j where $q \in L_{i_j}$. Similarly, let $\overline{L}(iar, q)$ be the (complementary) subsequence of iar with the i_j such that $q \notin L_{i_j}$. The value of the shift function $sh : Perm(\{1, \dots, r\}) \times Q \to Perm(\{1, \dots, r\})$ is defined by*

$$sh(iar, q) := \overline{L}(iar, q) L(iar, q).$$

The crucial point in defining the Rabin chain winning condition are the "hit positions" for each player. Hit functions were invented by Büchi [Büc83] to characterize games by a winning condition in normal form. In our case it would be sufficient to use one hit function which denotes the least hit position of both players and an extra function for the owner of the hit position, which is Büchi's original version. For a simpler argumentation, however, we proceed with two hit functions, one for each player.

Definition 2 *Assume (*). The hit function $hit_i : Perm(\{1, \dots, r\}) \times Q \to \{1, \dots, r\}$ for player $i \in \{0, 1\}$ over $t < \omega$ is defined by*

$$hit_i(iar, q) := \mu p \in \{1, \dots, r\} : q \in \begin{cases} L_{iar(p)} & \text{if } i = 0, \\ U_{iar(p)} & \text{if } i = 1. \end{cases}$$

The hit functions for both players are well-defined, since the Streett condition contains the pair $(L_r, U_r) = (Q, Q)$. The pair may be added to every Streett condition without changing the winning plays of the game.

For the whole play in the Streett game a sequence of extended index appearance records is defined as follows:

Definition 3 *Assume $(*)$. The function $IAR : Q^\omega \to Perm(\{1, \ldots, r\})^\omega$ associates with any $\pi \in Q^\omega$ an ω-sequence $IAR(\pi)$ of iar's defined by*

$$\begin{aligned} &IAR(\pi)(0) := (1, 2, \ldots, r), \\ &IAR(\pi)(t) := sh(IAR(\pi)(t-1), \pi(t)), \ \textit{for}\ t > 0. \end{aligned}$$

The sequence of hit positions for each player on a play is given by the following definition.

Definition 4 *Assume $(*)$. The hit function $HIT_i : Q^\omega \to \{1, \ldots, r\}^\omega$ over ω-sequences is defined for player $i \in \{0, 1\}$ by*

$$\begin{aligned} HIT_i(\pi)(0) &:= r, \\ HIT_i(\pi)(t) &:= hit_i(IAR(\pi)(t-1), \pi(t)), \ \textit{for}\ t > 0. \end{aligned}$$

Before we can convert Street games to Rabin chain games we need the following lemma, which give us a link between the least infinite-hit positions of each player and the winner of each play.

Lemma 5 *Assume $(*)$, let $\pi \in Q^\omega$ be a play in Γ and let*

$$p_0 := \mu p : p \in Inf(HIT_0(\pi))$$

be the least infinite-hit position of player 0 in the play π. Then the play π is won by player 0 iff for all $p_1 \in Inf(HIT_1(\pi))$ we have $p_0 \leq p_1$.

Proof. Let $\pi \in Q^\omega$ be a play in the game Γ and $\alpha := IAR(\pi)$ the infinite sequence of the index appearance records on π and $p_0 := \mu p : p \in Inf(HIT_0(\pi))$.

Let $t_0 < \omega$ be a position in π such that from t_0 onwards all the indices reached for both players are frequent ones, and $p_0 \leq HIT_0(\pi)(t)$ for all $t > t_0$. Then for all $t \geq t_0$ the sets

$$F(t) := \{\alpha(t)(p) \in \{1, \ldots, r\} : p_0 \leq p \leq r\}$$

are equal to the set $F(t_0)$ and consist of the frequent indices for player 0 on π. In the following we will prove that a play $\pi \in Q^\omega$ is a win for player 0 if and only if for all infinite-hit positions $p_1 \in Inf(HIT_1(\pi))$ of player 1 we have $p_0 \leq p_1$. This is done in two steps.

"$\Rightarrow$": Let π be a winning play for player 0 and $p_1 \in Inf(HIT_1(\pi))$. Then there exist infinitely many indices $t_1 < t_2 < \ldots < \omega$ with $t_0 < t_1$ and $HIT_1(\pi)(t_i) = p_1$. Since π is winning for player 0 all indices reached after t_0 are in $F(t)$, so we get $p_0 \leq p_1$.

"$\Leftarrow$": Assume now that there exists a $p_1 \in Inf(HIT_1(\pi))$ with $p_1 < p_0$. For all $p < p_0$ and $t_0 < t$ we have that $\alpha(t_0)(p) = \alpha(t)(p)$, which means that α is not changed for positions less than p_0 after the position t_0. So player 1 reaches infinitely often an index which is not in the set of the frequent indices for player 0 on the play π. So we get $Win(\pi) = 1$.

□

Theorem 6 *Assume* $(*)$. *Then there is a memory extension* $\hat{\Gamma} = (\hat{G}, \widehat{Win})$ *of the game* Γ *with the finite game graph* $\hat{G} = (\hat{Q}, \hat{Q}_0, \hat{Q}_1, \hat{E})$ *and where* $\widehat{Win}$ *is a Rabin chain winning condition s.t. for all plays* $\hat{\pi} \in \hat{Q}^\omega$ *in the game* $\hat{\Gamma}$ *we have*

$$(+) \qquad Win(Pr_1(\hat{\pi})) = 0 \textit{ iff } \widehat{Win}(\hat{\pi}) = 0.$$

Proof. We have to construct in the following a finite memory set M, a memory update function $\varphi : Q \times M \to M$ and the sequence of Rabin chain pairs. Then we have to prove that the two games are equivalent in the sense of $(+)$.

The finite set M of memory is given by

$$M := Perm(\{1,\ldots,r\}) \times \{1,\ldots,r\} \times \{1,\ldots,r\}.$$

The function $\varphi : Q \times M \to M$ is given for $(q, iar, h_0, h_1) \in Q \times M$ by

$$\varphi(q, iar, h_0, h_1) := (sh(iar,q), hit_0(iar,q), hit_1(iar,q)).$$

The Rabin chain acceptance condition $(E_1, F_1, E_2, F_2, \ldots, E_r, F_r)$ is built up inductively for $k \in \{1,\ldots,r\}$ and $F_0 := \emptyset$ by

$$\begin{aligned} E_k &:= F_{k-1} \cup \{(q, iar, h_0, h_1) \in Q \times M : h_1 = k-1 \text{ and } k \leq h_0 \leq r\}, \\ F_k &:= E_k \quad \cup \{(q, iar, h_0, h_1) \in Q \times M : h_0 = k \qquad \text{and } k \leq h_1 \leq r\}. \end{aligned}$$

Note that E_1 is the empty set. It is easy to see that for all $k \in \{1,\ldots,r\}$ the sets E_k consist exactly of this states for which the hit position for at least one of the players is less than k.

Let $\hat{G} = (\hat{Q}, \hat{Q}_0, \hat{Q}_1, \hat{E})$ be the memory–extended game graph by the finite memory set M and the memory–update function φ. Define $\widehat{Win} : \hat{Q}^\omega \to \{0,1\}$ over the Rabin chain $(E_1, \ldots, F_r)$. Then $\hat{\Gamma} = (\hat{G}, \widehat{Win})$ is a Rabin chain game. Finally we show that the two games are equivalent, w.r.t. $(+)$.

Therefore let $\hat{\pi} \in \hat{Q}^\omega$. Define $\pi := Pr_1(\hat{\pi})$. Then $\pi \in Q^\omega$, $HIT_0(\pi) = Pr_3(\hat{\pi})$ and $HIT_1(\pi) = Pr_4(\hat{\pi})$. By Lemma 5, $Win(\pi) = 0$ iff for all $p_1 \in Inf(HIT_1(\pi))$ the least infinite–hit position for player 0 is less or equal than p_1 iff there exists an index $k \in \{1,\ldots,r\}$ with $Inf(\hat{\pi}) \cap F_k \neq \emptyset$ and $Inf(\hat{\pi}) \cap E_k = \emptyset$. The last equivalence follows from the construction of the Rabin chain. Therefore $Win(\pi) = 0$ iff $\widehat{Win}(\hat{\pi}) = 0$. □

From Theorem 6 it follows directly that the generation of the states, of the edges and of the winning condition for a Rabin chain game based on a Streett game can be done separately. For the states and the winning condition it suffices to use simple enumeration routines with time complexity $O(n2^{r\log r})$. In the case of the edges the construction is also done by enumeration in time $O(n^2 2^{r\log r})$, whereby we have to use the hit–function and the shift–function as in Theorem 6.

Theorem 7 *The above construction yields, starting from a Streett game with n states and r accepting pairs, an equivalent Rabin chain game in time* $O(n^2 2^{r\log r})$.

Let us add a remark on games with special type of Streett winning condition: Assume a game $(*)$ has a Streett winning condition such that $L_1 \subseteq L_2 \subseteq \ldots \subseteq L_r$. In this special case we can extract a Rabin chain game immediately by defining the following winning condition assuming that $F_0 := \emptyset$ and $U_0 := \emptyset$:

$$E_k := F_{k-1} \cup U_{k-1} \text{ and } F_k := E_k \cup L_k, \text{ for all } k \in \{1, \ldots, r\}.$$

The proof is similar to the proof of Theorem 6 by defining the hit functions over the states $q \in Q$ for both players by

$$hit_0(q) := \mu k \leq r : q \in L_k \qquad \text{and} \qquad hit_1(q) := \mu k \leq r : q \in U_k.$$

It follows from the next section that Rabin chain games allow no–memory winning strategies. Hence in the case of Streett games with monotone increasing L–sets both players have a no–memory strategy.

4 Strategy construction for Rabin chain games

In this section we construct a deterministic winning strategy for a Rabin chain game. In the following we always mean by strategy a no–memory strategy.

Strategies will be presented by functions $\sigma : Q \to Q \cup \{\perp\}$. Such a function contains two strategies (given by the induced maps from Q_0 to $Q_1 \cup \{\perp\}$, resp. from Q_1 to $Q_0 \cup \{\perp\}$). If a player has no winning possibility in a state q this is indicated by $\sigma(q) = \perp$. Otherwise his strategy is to select $\sigma(q) \in Q$.

For each game (G, Win) with $\Omega = (E_1, F_1, \ldots, E_r, F_r)$ there is a dual game $(\overline{G}, \overline{Win})$, where players are changed: $\overline{G} = (Q, Q_1, Q_0, E)$. The acceptance condition can be complemented by shifting the state sets by one:

$$\overline{\Omega} = (\emptyset, E_1, F_1, E_2, \ldots, F_{r-1}, E_r, F_r, Q).$$

We construct a strategy by induction on the size of game graphs. So we need a notion of subgame. A game (G', Win') is called a subgame of

(G, Win) if $Q' \subseteq Q$ and $G' = (Q',\ Q_0 \cap Q',\ Q_1 \cap Q',\ E \cap (Q' \times Q'))$. The acceptance condition must be $\Omega' = \emptyset$ if $F_r \cap Q' = \emptyset$ and otherwise $\Omega' = (E'_l, F'_l, \ldots, E'_r, F'_r)$ with $l = \min\{k \in \{1, \ldots r\} | F_k \cap Q' \neq \emptyset\}$ and for $k \in \{l, \ldots, r\}$: $E'_k = E_k \cap Q'$ and $F'_k = F_k \cap Q'$. Furthermore, in the underlying graph at least one transition must leave each state. This is the only condition for a subset $Q' \subseteq Q$ to induce a subgame.

We define a function *reach*() which over Q computes the set R of states from which a player i can force a visit in a given "target set" T. More precisely, the function $reach(Q_i, E, T)$ in Figure 1 determines the states $R \subseteq Q \setminus T$ from which player i can force a visit in T; simultaneously a function $\sigma : R \to Q \cup \{\perp\}$ is computed which defines a "strategy" to ensure this. The word "strategy" here only refers to the domain R which is computed by *reach*().

```
function $reach(Q_i, E, T)$ :
  return $Reach(Q_i, E, T, T)$

function $Reach(Q_i, E, T, T')$ :
  $R := \emptyset$
  $\rho := \emptyset$
  if $T' = \emptyset$ then
    return $(R, \rho)$
  else
    for each $q \in T'$ do
      for each $p \in E^{-1}(q) \setminus T$ do
        if $p \in Q_i$ then
          $R := R \cup \{p\}$
          $\rho := \rho \cup (p \mapsto q)$
        else
          $t := true$
          for each $q' \in E(p)$ do
            if $q' \notin T$ then
              $t := false$
          if $t$ then
            $R := R \cup \{p\}$
            $\rho := \rho \cup (p \mapsto \perp)$
    $(R', \rho') := Reach(Q_i,\ E,\ T \cup R,\ R)$
    return $(R \cup R',\ \rho \cup \rho')$
```

FIGURE 1. Function *reach*()

In the functions *reach*() and *strategy*() the notations $E(p) := \{q | (p, q) \in E\}$ and $E^{-1}(q) := \{p | (p, q) \in E\}$ are used for describing the transition relation E.

The computation of function *reach*() starts with setting $R = \emptyset$. If there is a state $q \in Q_0 \setminus (T \cup R)$ with a transition leading into $T \cup R$ this state is added to R. If there is a state $q \in Q_1 \setminus (T \cup R)$ with all transitions leading into $T \cup R$ this state is also added to R. These steps are repeated until there is no further state to add. This construction ensures that $Q' = Q - R$ induces a subgame.

A strategy for a Rabin chain game (G, Win) with $G = (Q, Q_0, Q_1, E)$ and $\Omega = (E_1, F_1, \ldots, E_r, F_r)$ can be determined by the recursively defined function *strategy*() of Figure 2: It returns a triple (V_0, V_1, σ), where V_i is the winning set for player i and $\sigma : Q \to Q \cup \{\bot\}$ is a strategy function. We divide the problem in four different cases of which three are quite simple.

(a) The trivial case is $|Q| = 0$. On this graph winning sets are empty and the strategy function is also empty.

(b) If the acceptance condition is empty ($\Omega = \emptyset$), player 0 cannot win. Thus winning sets are $V_0 = \emptyset$ and $V_1 = Q$. For the states of player 0 the value of the strategy function is $\bot$, for the states of player 1 any of the directly reachable states can be chosen.

(c) If the graph has a non–empty acceptance condition with $E_1 = \emptyset$ the case can be reduced to the dual game $(\overline{G}, \overline{\mathit{Win}})$ with $\overline{G} = (Q, Q_1, Q_0, E)$ and

$$\begin{array}{ll} \overline{\Omega} = (F_1, E_2, \ldots, F_{r-1}, E_r, F_r, Q) & \text{if } F_r \neq Q \text{ or} \\ \overline{\Omega} = (F_1, E_2, \ldots, F_{r-1}, E_r) & \text{if } F_r = Q. \end{array}$$

(d) If none of the previous conditions holds the situation is more complicated. We select a pivot state $p \in E_1$. Player 1 wins if he can visit this state infinitely often. By applying $\mathit{reach}(Q, E, \{p\})$ we compute all states $R \subseteq Q \setminus \{p\}$ from which player 1 can force a visit of p. This function also computes a strategy ρ to get there. Then we determine a strategy for the remaining subgame $(G \setminus (R \cup \{p\}), \mathit{Win}')$ with $\Omega' = \Omega \setminus (R \cup \{p\})$ which results from the original game by removing all states of $R \cup \{p\}$. By induction hypotheses we get winning sets V_0, V_1 and a strategy σ for this subgame. Now we have a strategy for all states but p which leads to two possible cases:

1. Player 1 can prevent player 0 from entering V_0. Then player 0 has a winning strategy on V_0 and player 1 on the remaining set $V_1 \cup R \cup \{p\}$. Let's assume that in state p the choice of $q \in Q \cup \{\bot\}$ is a way for player 1 to avoid entering V_0. Then the strategy for the whole game is $\rho \cup \sigma \cup \{p \mapsto q\}$.
2. Player 1 cannot prevent entering V_0 in state p. Then we only know that player 0 has a winning strategy on V_0. We compute all further states from which player 0 can force entering V_0. We name this set R' and the strategy ρ'. Again the remaining

```
    function $strategy(Q, Q_0, Q_1, E, (E_1, F_1, \ldots, E_r, F_r))$:
(a)   if $size(Q) = 0$ then
        return $(\emptyset, \emptyset, \emptyset)$
(b)   elseif $r = 0$ then
        return $(\emptyset, Q, Q \to Q \cup \{\bot\} : \begin{array}{ll} p \mapsto \bot & \text{if } p \in Q_0, \\ p \mapsto q \in E(p) & \text{if } p \in Q_1 \end{array})$
(c)   elseif $E_1 = \emptyset$ then
        if $F_r = Q$ then
          $(V_1, V_0, \sigma) := strategy(Q, Q_1, Q_0, E, (F_1, E_2, \ldots, F_{r-1}, E_r))$
        else
          $(V_1, V_0, \sigma) := strategy(Q, Q_1, Q_0, E, (F_1, E_2, \ldots, F_{r-1}, E_r, F_r, Q))$
        return $(V_0, V_1, \sigma)$
(d)   else
        select $p \in E_1$
        $(R, \rho) := reach(Q_1, E, \{p\})$
        $(V_0, V_1, \sigma) := strategy(Q, Q_0, Q_1, E, (E_1, F_1, \ldots, E_r, F_r), R \cup \{p\})$
(d1)    if $(p \in Q_0 \ \wedge \ E(p) \cap V_0 = \emptyset) \ \vee \ (p \in Q_1 \ \wedge \ E(p) \setminus V_0 \neq \emptyset)$ then
          if $p \in Q_0$ then
            $q := \bot$
          else
            select $q \in E(p) \cap V_1$
          return $(V_0,\ V_1 \cup \{p\} \cup R,\ \rho \cup \{(p \mapsto q)\} \cup \sigma)$
(d2)    else
          $(R', \rho') := reach(Q_0, E, V_0)$
          $(U_0, U_1, \sigma') := strategy(Q, Q_0, Q_1, E, (E_1, F_1, \ldots, E_r, F_r), R' \cup V_0)$
          return $(U_0 \cup R' \cup V_0,\ U_1,\ \sigma|V_0 \cup \rho' \cup \sigma')$

    function $strategy(Q, Q_0, Q_1, E, (E_1, F_1, \ldots, E_r, F_r), R)$:
      if $F_1 \setminus R = \emptyset$ then
        return $strategy(Q, Q_0, Q_1, E, (E_2, F_2, \ldots, E_r, F_r), R)$
      else
        return $strategy(Q\setminus R,\ Q_0\setminus R,\ Q_1\setminus R,\ E\setminus(R \times R),$
                        $(E_1\setminus R,\ F_1\setminus R,\ \ldots,\ E_r\setminus R,\ F_r\setminus R))$
```

FIGURE 2. Function *strategy*()

subgraph induces a subgame $(G \setminus (V_0 \cup R'), Win')$ with $\Omega' = \Omega \setminus (V_0 \cup R')$. By induction hypotheses we get winning sets U_0, U_1 and a strategy σ' for this subgame . Thus the strategy for the whole game is $\sigma|_{V_0} \cup \rho' \cup \sigma'$.

This function terminates because in each recursive call the number of states or the length of the acceptance condition is reduced. The computation time of *strategy*() excluding its recursive calls is bounded by a polynomial. The function *strategy*() calls itself at most twice for smaller subgames. Thus the strategy algorithm has an exponential worst case complexity of $O(2^{|Q|})$.

5 Applying the synthesis algorithm

5.1 A simple type of Rabin chain games

Let $G = (Q, Q_0, Q_1, E)$ be a game graph. As usual the edge relation E is a subset of $(Q_0 \times Q_1) \cup (Q_1 \times Q_0)$. For the vertices in Q we add

- an irreflexive partial order $\prec$ and
- a vertex labeling T with $T(p) \in \{t, f\}$ for all $p \in Q$

such that

$$(*) \quad \begin{array}{l} \neg(p \prec q \vee q \prec p) \Rightarrow \\ \quad [(r \prec p \iff r \prec q) \wedge (p \prec r \iff q \prec r) \quad \vee \quad T(p) = T(q)]. \end{array}$$

Condition $(*)$ requires that two states which are incomparable w.r.t. $\prec$ have either identical labels or are comparable to the same states.

For any set $M \subseteq Q$ of vertices the set $min(M)$ of minimal elements of M w.r.t. $\prec$ is uniquely defined.
In order to gain a Rabin chain game we use the winning condition:

$$\text{Player 0 wins a play } \pi \iff \text{for all } p \in min(Inf(\pi)) \, : \, T(p) = t.$$

Let us verify that this is indeed a Rabin chain game. We construct the sets E_i, F_i as follows:

$$\begin{aligned} E_1 &= \{p \in Q \mid \neg\exists q \in Q \;\; T(q) = t \wedge q \prec p\}, \\ F_1 &= \{p \in Q \mid \forall q \in Q \setminus E_1 \;\; T(q) = f \Rightarrow p \prec q\}, \\ E_{i+1} &= \{p \in Q \mid \neg\exists q \in Q \setminus F_i \;\; T(q) = t \wedge q \prec p\}, \\ F_{i+1} &= \{p \in Q \mid \forall q \in Q \setminus E_{i+1} \;\; T(q) = f \Rightarrow p \prec q\}. \end{aligned}$$

Assume we have a play π such that for all $p \in min(Inf(\pi))$ $T(p) = t$ holds. Then there is an i such that $Inf(\pi) \cap F_i \neq \emptyset$ and $Inf(\pi) \cap E_i = \emptyset$. Otherwise there would be a state p in $Inf(\pi)$ with $T(p) = f$ that is minimal in $Inf(\pi)$ w.r.t $\prec$.

Now assume we have a play π and an i such that $Inf(\pi) \cap F_i \neq \emptyset$ and $Inf(\pi) \cap E_i = \emptyset$. By condition $(*)$ we know that

- for all $p \in E_i$ and for all $q \in F_i$ $\quad p \prec q$,
- for all $p \in F_i$ and for all $q \in E_{i+1}$ $\quad p \prec q$.

Then by definition of F_i the $\prec$-minimal vertices p of $Inf(\pi)$ are in $Inf(\pi) \cap F_i$ and for these vertices $T(p) = t$ holds.
So the above definition leads indeed to a Rabin chain game.

Example 8 Let $Q_0 = \{(n,0) \mid n \in \{0,\ldots,7\}\}$ and $Q_1 = \{(n,1) \mid n \in \{0,\ldots,7\}\}$, and let $(n,i) \prec (m,j) \iff n < m$ for all $i,j \in \{0,1\}$. The edge relation E is given is given in terms of auxiliary functions $f_i : Q_0 \to Q_1$ and $g_i : Q_1 \to Q_0$ by the following definition:

$$(p,p') \in E \iff \exists i \ \ (f_i(p) = p' \vee g_i(p) = p').$$

In our example we use the functions:

$$f_1((n,0)) = (n,1), \qquad f_2((n,0)) = \begin{cases} (n+5,1) & \text{if } n \in \{2,6\} \\ (n+1,1) & \text{otherwise,} \end{cases}$$

$$g_1((n,1)) = (n,0), \qquad g_2((n,1)) = \begin{cases} (n+5,0) & \text{if } n \in \{0,4\} \\ (n+1,0) & \text{otherwise.} \end{cases}$$

(Here "+" means addition modulo 8.) So player 0 can use a function f_i to reach a new vertex whereas player 1 uses a function g_i.

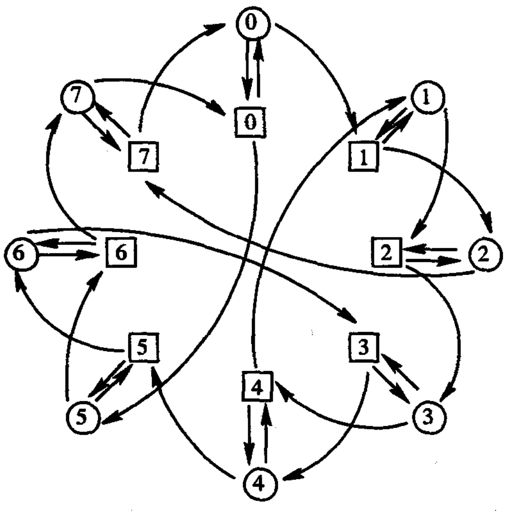

FIGURE 3. Game graph for the Rabin chain game of Example 8

Let T with $T(n,i) = t$ for even n and $T(n,i) = f$ for odd n. By definition player 0 wins iff the minimal number n visited infinitely often is even.

If we represent the states in Q_0 by circles and the ones in Q_1 by squares we get the game graph of Figure 3. Note that by other choices of the functions f_i, g_i many more examples of Rabin chain games can be generated. Beyond the concrete example above, for which an analysis "by hand" is still possible, in the next orders of magnitude we considered various examples from 32 up to 1024 states. We used contrived (however nonrandom) definitions of $\prec$, labeling T, and functions f_i, g_i.

Number of states	"Average time"	"Maximal time"
32	0.1	0.1
48	0.3	0.5
64	0.4	0.6
96	1.3	1.5
128	2.6	4.2
192	5.7	7.7
256	11.0	16.4
384	30.0	49.6
512	74.0	106.9
768	210.0	356.7
1024	469.0	749.5

TABLE 1. CPU time used (in seconds) to compute strategies for example games

The CPU time (on a SUN Sparc 10) for computing the strategies is shown in Table 1. We considered up to ten games of each given size with different edge relations and accepting chains. In the table, the average time as well as the maximal time of the considered cases are noted.

Although the worst case complexity of the algorithm is exponential, it seems applicable to problems of considerable size. Note that the maximal (and average) times we found by our experiments grow moderately even in our preliminary version with an uncontrolled choice of pivots (just using the first in a given enumeration).

5.2 Remarks on the strategy synthesis algorithm

The strategy synthesis algorithm splits a given game graph into two parts, containing the states from which player 0 resp. 1, wins.

The solution for Example 8 is depicted in Figure 4. From the states in the left-hand side, player 0 has a winning strategy as indicated by the chosen displayed edges, from the states in the right-hand part player 1 wins by traversing through the displayed edges.

In general, the critical point in applying the algorithm is the choice of the pivot states (in the "**select**" line of d) in Figure 2). We see two approaches to handle this problem. A possible heuristic is to select states p where $Reach(p)$ is of maximal size; this reduces the size of the subgame to which the induction hypothesis is applied. However in general this greedy method may fail to find the best descent in reducing the game graph: Note that in case d2, a second use of the inductive hypothesis is necessary which may spoil the optimality of reduction. Another approach would be to introduce a preprocessing of the game graph, yielding an order of pivot elements (similar to preprocessing a linear equation system when applying Gauss elimination). The structure theory of game graphs necessary for this is still missing.

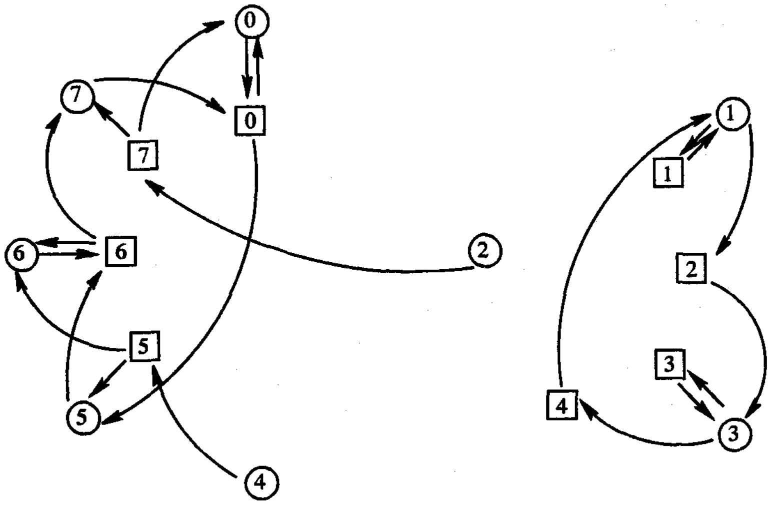

FIGURE 4. Winning strategies for Rabin chain game of example 8

6 Conclusion

We have presented and implemented (for the first time, to our knowledge) an algorithm which synthesizes finite-state winning strategies from automaton specifications of infinite games. In our implementation the necessary pivot choice is still done by the user.

Ongoing work deals with the implementation of algorithms to transform games with other winning conditions into Rabin chain form and of strategy synthesis algorithms in which memory construction and game graph decomposition are combined (as in [McN93], [TW94], and [YY93]).

7 References

[ALW89] M. Abadi, L. Lamport, and P. Wolper. Realizable and unrealizable specifications of reactive systems. In G. Ausiello et al., editor, *Automata, Languages, and Programming*, volume 372 of *LNCS*, pages 1 – 17, Berlin, Heidelberg, New York, 1989. Springer-Verlag.

[AMP95] E. Asarin, O. Maler, and A. Pnueli. Symbolic controller synthesis for discrete and timed systems. In P. Antsaklis et al., editor, *Hybrid Systems II*, volume 999 of *LNCS*, pages 1 – 20, Berlin, Heidelberg, New-York, 1995. Springer-Verlag.

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

[Büc83] J. R. Büchi. State strategies for games in $F_{\sigma\delta} \cap G_{\delta\sigma}$. *J. Symb. Logic*, 48:1171 – 1198, 1983.

[EJ91] E.A. Emerson and C.S. Jutla. Tree automata, mu-calculus and determinacy. In *Proc. 32nd IEEE Symp. on the Foundations of Computing*, pages 368 – 377, 1991.

[GH82] Y. Gurevich and L. Harrington. Trees, automata, and games. In *Proc 14th ACM Symp. on the Theory of computing*, pages 60 – 65, San Fancisco, 1982.

[GS53] D. Gale and F. M. Stewart. Infinite games with perfect information. *Annals of Mathematical Studies*, 28:245 – 266, 1953.

[KG95] R. Kumar and V. K. Garg. *Modeling and Control of Logical Discrete Event Systems*. Kluwer Academic Publishers, Norwell, MA, USA, 1995.

[McN93] R. McNaughton. Infinite games played on finite graphs. *Ann. Pure Appl Logic*, 65:149 – 184, 1993.

[Mos91] A.W. Mostowski. Games with forbidden positions. Technical Report Preprint No. 78, Uniwersytet Gdański, Instytyt Matematyki, 1991.

[MS95] D.E. Muller and P.E. Schupp. Simulating alternating tree automata by nondeterministic automata: New results and new proofs of the theorems of Rabin, McNaughton and Safra. *Theoretical Computer Science*, 141:69 – 107, 1995.

[Muc92] A. Muchnik. Games on infinite trees and automata with dead-ends: A new proof for the decidability of the monadic second order theory of two successors. *Bulletin of the European Association for Theoretical Computer Science*, 48:220 – 267, 1992.

[NYY92] A. Nerode, A. Yakhnis, and V. Yakhnis. Concurrent programs as strategies in games. In Moschovakis Y., editor, *Logic from Computer Science*. Springer, 1992.

[PR89] A. Pnueli and R. Rosner. On the systhesis of a reactive module. In *Proc. 16th ACM Sympos. on Principles of Prog. Lang.*, pages 179 – 190, Austin, 1989.

[Rab69] M. O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1 – 35, 1969.

[RW89] P. J. G. Ramadge and W. M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77, 1:81 – 98, 1989.

[Saf92] S. Safra. Exponential determinization for ω-automata with strong-fairness acceptance condition. In *Proc. 24th ACM Symposium on Theory of Computing (STOC)*, pages 275–282, 1992.

[Str82] R.S. Streett. Propositional dynamic logic of looping and converse. *Information and Control*, 54:121 – 141, 1982.

[Tho90] W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 4, pages 131–191. North-Holland, Amsterdam, 1990.

[Tho95] W. Thomas. On the synthesis of strategies in infinite games. In Ernst W. Mayr and Claude Puech, editors, *STACS 95*, volume 900 of *LNCS*, pages 1 – 13, Berlin, Heidelberg, New-York, 1995. Springer-Verlag.

[TW94] J.G. Thistle and W.M. Wonham. Control of infinite behaviour of finite automata. *SIAM J. of Control and Optimization*, 32(4):1075 – 1097, 1994.

[YY93] A. Yakhnis and V. Yakhnis. Gurevich - Harrington's games defined by finite automata. *Ann. Pure Appl Logic*, 62:265 – 294, 1993.