On Deciding if Deterministic Rabin Language Is in Büchi Class

Tomasz Fryderyk Urbański

Institute of Computer Science, University of Warsaw,
ul.Banacha 2, 02-097 Warsaw, Poland
tu153507@zodiac.mimuw.edu.pl

Abstract. In this paper we give a proof that it is decidable for a deterministic tree automaton on infinite trees with Rabin acceptance condition, if there exists an equivalent nondeterministic automaton with Büchi acceptance condition. In order to prove this we transform an arbitrary deterministic Rabin automaton to a certain canonical form. Using this canonical form we are able to say if there exists a Büchi automaton equivalent to the initial one. Moreover, if it is the case, the canonical form allows us also to find a respective Büchi automaton.

AMS Subject Classification 03D05 03B70

1 Introduction

Automata on infinite sequences and infinite trees were first introduced by J.R. Büchi (1962) [1] and M.O. Rabin (1969) [5] in order to find the proof of decidability of monadic logic of one and many successors respectively. Later they have also proved useful in modeling finite state systems whose behaviour can be infinite. They have found applications in describing behaviour of concurrent processes in real time systems or in construction of automatic program verification systems. There exist several classes of finite automata having different expressive power. Among them a meaningful role is played by automata with Büchi and Rabin acceptance conditions (see [6,7]). The theory of these automata gives rise to many natural questions which have not been answered until now. One of these is treated in this paper. It is known that the classes of tree languages recognized by Rabin automata with successive indices induce a proper hierarchy (see [3]). Languages recognized by Büchi automata are located at a low level of this classification. In this context, it is natural to ask if, given some Rabin automaton $\mathcal{A}$ for a language L, we can determine the minimal index of a Rabin automaton recognizing L. For deterministic automata on infinite words this problem has been solved. T.Wilke and H.Yoo [8] gave an algorithm which computes the aforementioned minimal index for any Rabin language consisting of infinite words. At the same time there have been attempts to consider a similar problem for automata on infinite trees in case of some more restricted subclasses of Rabin automata. In the paper [4] by D.Niwiński and I.Walukiewicz there can be found a method of computing the minimal Rabin index for the class

U. Montanari et al. (Eds.): ICALP 2000, LNCS 1853, pp. 663–674, 2000.

of infinite tree languages, whose all paths are in some $\omega-$ regular set of infinite words. The problem in general has not been yet solved for automata on infinite trees. In this paper we make one step in this direction, giving a procedure which determines if a language accepted by a deterministic Rabin automaton can be accepted by a (possibly nondeterministic) Büchi automaton. In terms of the hierarchy described below it means determining if the minimal index of a language accepted by a deterministic tree Rabin automaton is equal 1. Our problem is related to the well-known open question of determining the alternation level of a formula of the μ-calculus. Recently M. Otto [9] showed a decision procedure to determine whether a formula is equivalent to an alternation-free formula. It is known that the expressive power of Büchi automata corresponds to that of the formulas of the alternation level $\nu\mu$, however we do not know of any direct μ-calculus characterization of deterministic automata. Nevertheless we may hope that extension of our techniques can help to solve, at least to some extent, the aforementioned μ-calculus problem.

2 Basic Definitions and Notations

2.1 Mathematical Preliminaries

Let ω denote the set of natural numbers. For an arbitrary function f, we denote its domain by $\mathrm{Dom}(f)$. Throughout the paper Σ denotes a finite alphabet. For a set X, X^* is the set of finite words over X, including the empty word ϵ. We denote the length of a word w by $|w|$ ($|\epsilon|=0$). We let X^ω denote the set of infinite words over X. The concatenation wu of words $w \in X^*$ and $u \in X^* \cup X^\omega$ is defined as usual. A word $w \in X^*$ is a prefix of $v \in X^* \cup X^\omega$, in symbols: $w \leq v$, if there exists u such that $v = wu$; the symbols $\geq, >$ etc. have their usual meaning.

An infinite binary tree over Σ is any function $t : \{0,1\}^\star \to \Sigma$. We denote the set of all such functions by T_Σ. An incomplete binary tree over Σ is a function $t : X \to \Sigma$, where $X \subseteq \{0,1\}^\star$ is closed under prefixes, i.e. $w \in X \wedge w' \leq w \Rightarrow w' \in X$ and satisfies the following condition: $\forall w \in X (w0 \in X \wedge w1 \in X) \vee (w0 \notin X \wedge w1 \notin X)$. The set of incomplete binary trees over Σ is denoted by TP_Σ. Note that $T_\Sigma \subseteq TP_\Sigma$. A node $w \in Dom(t)$ is a leaf of t if $w0 \notin Dom(t)$ and $w1 \notin Dom(t)$. We denote the set of leaves of t by $\mathrm{Fr}(t)$. A node which is not a leaf is called the inner node. Let $t \in TP_\Sigma$ and $w \in t$. Then the symbol $t.w$ denotes a subtree of the tree t induced by a node w defined by

$$Dom(t.w) = \{v \mid wv \in Dom(t)\}$$
$$t.w(v) = t(vw), \text{ for } v \in Dom(t.w)$$

A word π from the set $\{0,1\}^\omega$ is a path in a tree $t \in TP_\Sigma$ if all its finite prefixes are in $Dom(t)$. A word π from the set $\{0,1\}^\star$ is a path in the tree $t \in TP_\Sigma$ if $\pi \in Dom(t)$ and $\pi \in Fr(t)$. Throughout this paper whenever we use the symbol π we will mean a path in the sense of the above definitions.

Below we present some notations used throughout this paper for certain subsets of $\{0,1\}^\star$:

$$[w, v] = \{u \in \{0,1\}^\star \mid w \leq u \leq v\}$$
$$(w, v) = \{u \in \{0,1\}^\star \mid w < u < v\}$$
$$\pi_w = \{u \in \{0,1\}^\star \mid w \leq u < \pi\}$$

We write $\exists! x\phi(x)$ to mean that there exists exactly one x satisfying formula $\phi(x)$, similarly a formula $\exists^\infty x\phi(x)$ means that there exist infinitely many x's satisfying $\phi(x)$. A symbol $\dot{\vee}$ will denote exclusive or. For a n-tuple $x = (g_1, ..., g_n)$ we define $x|_i = g_i$.
Now we will introduce some notations concerning directed graphs, used throughout this paper. A cycle S in an arbitrary graph $\mathcal{G}(V, E)$ is a sequence of nodes $a_1, a_2, \ldots, a_k$ such that $k \geq 2$, $a_1 = a_k$, and $(a_i, a_{i+1}) \in E$, for $i \in \{1, \ldots, k-1\}$; it will be denoted by $[a_1, a_2, \ldots, a_k]$. For a cycle $S = [a_1, a_2, \ldots, a_k]$ a notation $\langle S\rangle$ denotes the set $\{a_1, a_2, \ldots, a_k\}$. Given an arbitrary vertex subset X and a cycle S in a graph $\mathcal{G}(V, E)$ we say that the cycle S omits the set X if $\langle S\rangle \cap$ X $= \emptyset$. A connected component of a vertex $p \in V$ with respect to the set $X \subseteq V$ is defined as follows:

$$\mathrm{Con}(X, p) = \{q \in V \mid \exists S - \text{ a cycle in a graph } \mathcal{G} \text{ such that } (\langle S\rangle \cap X = \emptyset \wedge p, q \in \langle S\rangle)\}$$

The set of all connected components with respect to some set X in a graph $\mathcal{G}$ is denoted by $\mathrm{Con}(X, \mathcal{G})$.

2.2 Stopping Automata

Definition 1. A nondeterministic stopping automaton with Rabin acceptance condition *is a 6-tuple* $= \langle Q, \Sigma, q_0, \Delta, \Omega, A\rangle$, *whose components satisfy the following conditions, respectively:*

- $q_0 \in Q$
- $\Delta \subseteq Q \times \Sigma \times (Q \cup A) \times (Q \cup A)$
- $\Omega = \{(L_1, U_1), \ldots, (L_n, U_n), (\emptyset, U_{n+1})\}$ *where* $\forall i \in \{1, \ldots, n+1\}\, L_i, U_i \subseteq Q$
- $Q \cap A = \emptyset$

Q is a finite set of states. A describes a finite set of stopping states disjoint with Q. Ω will be called the set of acceptance conditions. Sometimes when the set of stopping states is empty we will omit the last component in the above 6-tuple and we will not call such an automaton a stopping one.
In this paper there will also appear another type of stopping Rabin automaton, whose initial state is omitted, what we denote by $\langle Q, \Sigma, -, \Delta, \Omega, A\rangle$. If a construction requires giving the initial state, we will use the following notation:

Definition 2. *For an arbitrary stopping automaton* $\mathcal{A}$, $\mathcal{A}[p]$ *denotes a stopping automaton* $\mathcal{A}$ *with the initial state set to* p.

A run of a stopping automaton $\mathcal{A}$ on a tree $t \in T_\Sigma$ is an incomplete tree $r_{t,A} \in TP_{Q\cup A}$ defined by conditions:

- $r_{t,A}(\epsilon) = q_0$

- $\forall w \in (Dom(r_{t,A}) \setminus \mathrm{Fr}(r_{t,A}))\ (r_{t,A}(w), t(w), r_{t,A}(w0), r_{t,A}(w1)) \in \Delta$.
- $\forall w \in \mathrm{Fr}(r_{t,A})\ r_{t,A}(w) \in A$
- $\forall w \in (Dom(r_{t,A}) \setminus \mathrm{Fr}(r_{t,A}))\ r_{t,A}(w) \in Q$

Note that once the automaton reaches a stopping state it stops operating. For a run $r_{t,A}$ and an arbitrary set $X \subseteq \{0,1\}^\star$ a notation $r_{t,A}(X)$ will be used to denote a set $\{r_{t,A}(w) \mid\ w \in X\}$. We also introduce another notation concerning the runs of stopping automata:

$$In(r_{t,A}|\pi) = \{q \mid \exists^\infty\ w < \pi\ (r_{t,A}(w) = q)\ \}$$

We say that a run $r_{t,A}$ is accepting, what we denote by $r_{t,A} \in Acc$, if:

$$\forall \pi\ (\exists i \in \{1, \ldots n+1\}\ (\mathrm{In}(r_{t,A}|\pi) \cap L_i = \emptyset \wedge \mathrm{In}(r_{t,A}|\pi) \cap U_i \neq \emptyset))$$

We reserve the symbol $\mathrm{L}(\mathcal{A})$ for a language accepted by a stopping automaton $\mathcal{A}$ defined by:

$$L(\mathcal{A}) = \{t \in T_\Sigma \mid \exists r_{t,A} \in Acc\}$$

In this paper we assume that all stopping automata in consideration will satisfy the following conditions:

- $\forall a \in \Sigma\ \forall q \in Q\ \exists p, r \in (Q \cup A)\ (q, a, p, r) \in \Delta$
- $\Omega = \{(L_1, U_1), (L_2, U_2), \ldots, (L_n, U_n), (\emptyset, U_{n+1})\}$,where:
  $(\forall i \in \{1 \ldots n\}\ (L_i \neq \emptyset\ \wedge\ L_i \cap U_i = \emptyset))\ \wedge\ (U_{n+1} \cap \bigcup_{i=1}^{n} (L_i \cup U_i) = \emptyset)$
- $\forall x \in \bigcup_{i=1}^{n} L_i \cup U_{n+1}\ \exists t \in T_\Sigma\ \exists r_{t,A} \in Acc\ \exists w \in \{0,1\}^\star\ r_{t,A}(w) = x$

Observe that the first dependency guarantees the existence of a run of a stopping automaton over every tree. States which satisfy the last condition are said to be reachable in some accepting run. It is easy to see that the imposed conditions do not diminish the expressive power of the stopping automata. We would like to emphasize that sets U_i for $i \in \{1, \ldots, n+1\}$ can be empty. The index of a stopping automaton $\mathcal{A}$ is denoted by:

$$Ind(\mathcal{A}) = \begin{cases} 2n+1 & if\ U_{n+1} \neq \emptyset \\ 2n & if\ U_{n+1} = \emptyset \end{cases}$$

We write $\mathrm{Ind}(\Omega)$ instead of $\mathrm{Ind}(\mathcal{A})$, where Ω is the set of acceptance conditions of a stopping automaton $\mathcal{A}$. Moreover for the elements of the set Ω we use the subsequent notation:

$$\forall X = (L_i, U_i) \in \Omega\ \ (X|_1 = L_i\ \wedge X|_2 = U_i)$$

Definition 3. *Stopping automata with index 1 are called* **Büchi automata**.

Languages accepted by some stopping automaton with Rabin (Büchi) acceptance condition are called Rabin (Büchi) languages. It is worth noting that classes of languages recognized by automata with Rabin acceptance condition having successive indices induce a proper hierarchy (see [3]).

Definition 4. *A stopping automaton $\mathcal{A}$ is* **deterministic** *if it satisfies the following condition:*

$$\forall q \in Q \ \forall q', q'', p', p'' \in (Q \cup A) \ \ \forall a \in \Sigma \, (\, (q, a, q', q'') \in \Delta \wedge (q, a, p', p'') \in \Delta \Rightarrow \\ \Rightarrow q' = p' \wedge q'' = p'' \,)$$

Note that if we deal with deterministic automata then we can assume that there exists exactly one run of such an automaton over some tree.

Definition 5. *We say that a stopping automaton $\mathcal{A}$ is* **frontier deterministic** *if the following dependency holds true:*

$$\forall t \in T_\Sigma \, \forall r_{t,A}, s_{t,A} \ (\ Fr \ (r_{t,A}) = \ Fr(s_{t,A}) \ \ \wedge (\, \forall w \in Fr(r_{t,A}) \, r_{t,A}(w) = s_{t,A}(w)))$$

Consequently, frontier determinism guarantees that for some complete tree all possible runs stop in the same nodes reaching the same states.
We will use the notation $\mathcal{A} \simeq \mathcal{B}$ to denote **equivalence** between frontier deterministic stopping automata $\mathcal{A}$ and $\mathcal{B}$. This concept is expressed formally as follows:

- $L(\mathcal{A}) = L(\mathcal{B})$
- $\forall t \in T_\Sigma \, \forall r_{t,A}, s_{t,B} \ (\mathrm{Fr}(r_{t,A}) = \mathrm{Fr}(s_{t,B}) \ \wedge \ \forall \, w \in \mathrm{Fr}(r_{t,A}) \, r_{t,A}(w) = s_{t,B}(w))$

Observe that the second condition refers to all runs of automata $\mathcal{A}$ and $\mathcal{B}$, not only to the accepting ones.
Set $X \subseteq Dom(t)$ is an **antichain** with respect to the order relation $\leq$ if any two elements of X are incomparable. Let f be a function associating trees $t(w) \in T_\Sigma$ with elements w from X. Then we denote the **substitution** by $t[f]$, the **limit** of the sequence t_n by lim t_n and the **iteration** of t along the interval $[v, w]$ by $t^{[v,w]}$. Definitions of the above concepts and also the concept of the **trace of iteration** are well known and can be found e.g. in the paper by Niwinski [3]

3 Relevant States

Let $\mathcal{A}$=$\langle Q, \Sigma, q_0, \Delta, \Omega, A\rangle$ denote an arbitrary stopping automaton, where: $\Omega = \{(L_1, U_1), (L_2, U_2), \ldots, (L_n, U_n), (\emptyset, U_{n+1})\}$.

Definition 6. *We say that some state $q \in \bigcup_{i=1}^{n} L_i$ is* **irrelevant in a run** $r_{t,A}$ *if its occurrences are covered by a finite number of paths in this run:*

$$\exists \, \pi_1, \ldots, \pi_k \in \{0, 1\}^\omega \ (r_{t,A}(w) = q \Rightarrow \exists \, i \in \{1, \ldots, k\} \, w < \pi_i) \tag{1}$$

Furthermore we say that a state $q \in \bigcup_{i=1}^{n} L_i$ is **irrelevant** *for an automaton $\mathcal{A}$ if its occurrences are irrelevant in an arbitrary accepting run of this automaton.*

A state which is not irrelevant is called **relevant**. Observe that if a state p is irrelevant for a stopping automaton $\mathcal{A}$ then for an arbitrary tree $t \in L(\mathcal{A})$ and an arbitrary accepting run $r_{t,A}$ there exists a natural number K such that:

$$\forall\, w \in \{0,1\}^\star (r_{t,A}(w) = p \wedge |w| \geq K) \Rightarrow \exists!\pi\, (w < \pi \wedge p \in In(r_{t,A}|\pi)) \tag{2}$$

Less formally, if in some node below level K the automaton reaches a state p then this state must occur infinitely often on some uniquely determined path going through this node.

Definition 7. *A* ***gadget*** *of states p and q admitted by an automaton $\mathcal{A}$ is a 4-tuple $G(p,q) = (g_1, g_2, g_3, g_4)$ with $g_i \in \{0,1\}^\star$ satisfying the following conditions:*

- $g_1 < g_2 < g_3 \ \wedge\ g_1 < g_2 < g_4$
- $g_3 \not\leq g_4 \ \wedge\ g_3 \not\geq g_4$
- $\exists\, t \in L(\mathcal{A})\, \exists r_{t,A} \in Acc\, \exists i \in \{1,\ldots,n\}\, (\, r_{t,A}(g_1) = p \ \wedge\ r_{t,A}(g_4) = q \ \wedge$ $r_{t,A}(g_2) = r_{t,A}(g_3) = u \in U_i \ \wedge\ r_{t,A}([g_2, g_3]) \cap\ L_i = \emptyset\,)$
- $r_{t,A}([g_1, g_3]) \cap U_{n+1} = \emptyset \ \wedge\ r_{t,A}([g_1, g_4]) \cap U_{n+1} = \emptyset$

¿From now on when we say that an automaton $\mathcal{A}$ admits a gadget G in a tree t in a run $r_{t,A}$, we mean that t and $r_{t,A}$ satisfy the two conditions above. Note that in this case the language accepted by the automaton $\mathcal{A}$ includes also the iteration of the tree t along the interval $[g_2, g_3]$.

Definition 8. *An automaton $\mathcal{A}$ has* **the accessibility property** *iff*

$$\forall p, p' \in \bigcup_{i=1}^{n} L_i\, \exists t \in L(\mathcal{A}[p])\, \exists r_{t,A[p]} \in Acc\ \exists w \in \{0,1\}^\star\, (r_{t,A[p]}(w) = p' \ \wedge\ w \neq \epsilon)$$

Clearly, if $Ind(\mathcal{A}) = 1$ then $\bigcup_{i=1}^{n} L_i = \emptyset$ and therefore $\mathcal{A}$ has the accessibility property. It is easy to prove that if an automaton $\mathcal{A}$ has the accessibility property, admits a gadget $G(p,q)$ for some $p, q \in \bigcup_{i=1}^{n} L_i$ and $Ind(\mathcal{A})$ is even (hence $U_{n+1} = \emptyset$), then it actually admits a gadget $G(p,q)$ for any $p, q \in \bigcup_{i=1}^{n} L_i$. The subsequent lemma characterizes relevantness of states using the notion of a gadget.

Lemma 1. *Let $\mathcal{A}$ be a frontier deterministic stopping automaton of the form $\langle Q, \Sigma, q, \Delta, \Omega, A\rangle$, whose index is even. Then the initial state q is relevant for the automaton $\mathcal{A}$ if and only if this automaton admits a gadget $G(q,q)$.*

If the automaton $\mathcal{A}$ admits a gadget $G(q,q)$ then using the iteration it is easy to obtain a tree accepted by $\mathcal{A}$ in which the occurrences of the state q are relevant. The reverse implication is only slightly more difficult - for the details see [10]. Presently we will prove a lemma, which allows us to transform Rabin's automaton into Büchi form if we impose on all states in the set $\bigcup_{i=1}^{n} L_i$ some condition, which uses the notion of relevantness.

Lemma 2. *Let $\mathcal{A}$ be a deterministic stopping automaton of the form $\langle Q, \Sigma, q, \Delta, \Omega, A\rangle$, where $\Omega = \{(L_1, U_1), \ldots, (L_n, U_n), (\emptyset, U_{n+1})\}$ and $n \geq 1$. If for any state $s \in \bigcup_{i=1}^{n} L_i$ s is irrelevant for an automaton $\mathcal{A}[s]$, then there exists frontier deterministic stopping automaton $\mathcal{B} = \langle Q', \Sigma, q', \Delta', \Omega', A\rangle$ with index 1 equivalent to the automaton $\mathcal{A}$.*

For the proof see [10].

4 Canonical Decomposition of Deterministic Automaton

4.1 On Representation of Automata in Form of a Composition

Consider a sequence of stopping automata $(\mathcal{A})_{i\in\{1,\ldots,k\}}$ defined by dependencies:

$$\forall i \in \{1, \ldots, k\}\ \mathcal{A}_i = \langle Q_i, \Sigma, p_i, \Delta_i, \Omega_i, A_i\rangle \tag{3}$$

$$\forall i \in \{1, \ldots, k\}\ \Omega_i = \{(L_1^i, U_1^i), \ldots, (L_{n_i}^i, U_{n_i}^i), (\emptyset, U_{n_i+1}^i)\}. \tag{4}$$

$$\forall i, j \in \{1, \ldots, k\}\ (i \neq j \Rightarrow Q_i \cap Q_j = \emptyset) \tag{5}$$

We define an automaton denoted by a symbol $\mathcal{A}_1 \oplus \ldots \oplus \mathcal{A}_k$ as a stopping automaton $\langle Q', \Sigma, -, \Delta', \Omega', A'\rangle$, whose construction is presented below:

- $Q' = \bigcup_{i=1}^{k} Q_i$
- $\Delta' = \bigcup_{i=1}^{k} \Delta_i$
- $A' = (\bigcup_{i=1}^{k} A_i) \setminus Q'$

The set Ω' is a little more tedious to define and comes as a result of the following construction: Let $m\ = \max\{\, n_i \mid i \in \{1, \ldots, k\}\} + 1$. We change each set Ω_i into a sequence consisting of its elements keeping the same notation: $\Omega_i(j)$ for $j \in \{1, \ldots, n_i + 1\}$, taking care that the only pair of the form $(\emptyset, U)$ is at the position $n_i + 1$. Now we define a sequence $\Omega'(j)$ for $j \in \{1, \ldots, m\}$:

$$\Omega'(j) = \begin{cases} \Big(\bigcup\limits_{\substack{1\leq l\leq k \\ j\leq n_l}} (\Omega_l(j)|_1),\ \bigcup\limits_{\substack{1\leq l\leq k \\ j\leq n_l}} (\Omega_l(j)|_2)\ \Big), \text{for } j < m \\ (\emptyset, (\bigcup\limits_{1\leq l\leq k} (\Omega_l(n_l + 1)|_2))) \qquad\qquad \text{for } j = m \end{cases} \tag{6}$$

As can be seen, we sum the conditions pointwise except for the last pairs, which summed separately form the last condition. Now let Ω' be a set of the elements of the sequence $\Omega'(i)$ for $i \in \{1, \ldots, m\}$. Note that the above construction is not unique. Transforming the set Ω_i into a sequence we can arrange its elements in many different ways. This ambiguity however will have no influence on the proofs and can be easily removed, if we define the set of acceptance conditions of a stopping automaton as a sequence.

Definition 9. *The automaton* $\mathcal{A}_1 \oplus \ldots \oplus \mathcal{A}_k$ *described above will be called* **the composition of a sequence** $(\mathcal{A})_{i\in\{1,\ldots,k\}}$.

Note that the following inequalities hold true:

$$\max\{Ind(\Omega_i) \mid 1 \leq i \leq k\} \leq Ind(\Omega') \leq \max\{Ind(\Omega_i) \mid 1 \leq i \leq k\} + 1 \quad (7)$$

Note that the composition is a Büchi automaton if so is each automaton $\mathcal{A}_i$. For a particular composition $\mathcal{B} = \mathcal{A}_1 \oplus \ldots \oplus \mathcal{A}_k$ we define a function **SwitchingStates**^{B} , which associates with each automaton $\mathcal{A}_i = \langle Q_i, \Sigma, -, \Delta_i, \Omega_i, A_i\rangle$ some subset of Q_i in the following way:

$$\text{SwitchingStates}^B(\mathcal{A}_i) = Q_i \cap \bigcup_{j=1}^{k} A_j$$

The function SwitchingStates^B describes, which states of the given automaton are the stopping states of the other components of the composition. In this paper often compositions of the form $\mathcal{B} = \mathcal{A}_1 \oplus \ldots \oplus \mathcal{A}_k$ (where $\mathcal{A}_i$ are defined as in conditions 3,4) will satisfy additional conditions, stated below.

Definition 10. *Assume that there exists a partition of a set* $\{1, \ldots, k\}$ *into two sets* X, Y*, which satisfy the conditions:*

- $\forall i \in X \; Ind(\mathcal{A}_i) = 1$
- $\forall t \in T_\Sigma \forall r_{t,B} \forall\, \pi \in \{0,1\}^\omega \, (\exists i \in Y\, (In(r_{t,B}|\pi) \cap Q_i \neq \emptyset \; \wedge In(r_{t,B}|\pi) \not\subseteq Q_i))$ $\Rightarrow \; (\exists i \in X\; (In(r_{t,B}|\pi) \cap U_1^i \neq \emptyset))$
- $\forall i \in Y \; (\; Ind(\mathcal{A}_i) > 1 \; \Rightarrow SwitchingStates^B(\mathcal{A}_i) = \bigcup_{j=1}^{n_i} L_j^i \;)$
- $\forall i \in Y$ *SwitchingStates*$^B(\mathcal{A}_i)$ *are reachable in the accepting runs of the automaton* $\mathcal{B}$

Composition of a sequence of stopping automata is proper *if there exists a partition* X, Y *such that the above conditions hold true.*

Intuitively the second condition means: if in some run (not necessarily accepting) on some path the automaton $\mathcal{B}$ visits states of some automaton from the group Y infinitely often, but will never remain in its states forever, then the acceptance conditions are satisfied with regard to some automaton from the group X.
Again assume that $\mathcal{B} = \mathcal{A}_1 \oplus \ldots \oplus \mathcal{A}_k = \langle Q, \Sigma, -, \Delta, \Omega, A\rangle$. Let us fix for some $i \in \{1, \ldots, k\}$ an automaton $\mathcal{A}_i = \langle Q_i, \Sigma, -, \Delta_i, \Omega_i, A_i\rangle$. Moreover let there exist an automaton $\mathcal{C} = \langle Q', \Sigma, -, \Delta', \Omega', A'\rangle$ such that $Q' \cap Q = \emptyset$ and a function $f : Q_i \overset{1-1}{\to} Q'$. Then for an arbitrary $q \in Q$ **a substitution** $\mathcal{C} \overset{f}{\mapsto} \mathcal{A}_i$ **into the composition** $\mathcal{A}_1 \oplus \ldots \oplus \mathcal{A}_k\, [q]$ is a composition $\mathcal{A}_1 \oplus \ldots \oplus \mathcal{A}_{i-1} \oplus \mathcal{C} \oplus \mathcal{A}_{i+1} \oplus \ldots \oplus \mathcal{A}_k[q']$, which we change in the following manner:

- if $q \in Q_i$, we take $q' = f(q)$, otherwise $q' = q$,
- all occurrences of states p from the set SwitchingStates$^{\mathcal{B}}(\mathcal{A}_i)$ are replaced by states $f(p)$

4.2 Automaton Graph

Let $\mathcal{A} = \langle Q, \Sigma, q_0, \Delta, \Omega, A\rangle$ denote an arbitrary stopping automaton such that $\Omega = \{(L_1, U_1), (L_2, U_2), \ldots, (L_n, U_n), (\emptyset, U_{n+1})\}$, where: $n \geq 1$. We will construct **a directed automaton graph** $\mathcal{A}$, which will be denoted by $\mathcal{G}_{\mathcal{A}}$. Initially, we define a directed graph: $\mathcal{H}_{\mathcal{A}} = \langle V, E\rangle$, where $V = \bigcup_{i=1}^{n} L_i \cup U_{n+1}$ and $E \subseteq V \times V$ has the following form:

$$E = \{(x, y) \in V \times V \mid \exists\, t \in L(\mathcal{A})\ \ \exists r_{t,A} \in Acc\, \exists w, w' \in \{0,1\}^{\star}$$
$$(r_{t,A}(w) = x \wedge\ r_{t,A}(w') = y\ \ \wedge w < w'\ \wedge\ r_{t,A}((w, w')) \cap V = \emptyset)\}$$

The above graph will be helpful in defining a directed stopping automaton graph $\mathcal{A}$, $\mathcal{G}_{\mathcal{A}} = \langle$V',E'$\rangle$. Namely, we define:

$$\text{V'} = \ \text{Con}(U_{n+1}, \mathcal{H}_{\mathcal{A}}) \cup \{\ \{q\} \mid \text{Con}(U_{n+1}, q) = \emptyset\ \wedge\ q \in V\}\ ,$$
$$\text{E'} = \{(X, Y) \in V' \times V' \mid \exists x \in X\ \exists y \in Y\ (x, y) \in E\},$$

Let us note that considering the character of the above construction and a condition $U_{n+1} \cap \bigcup_{i=1}^{n} L_i = \emptyset$, which holds true, the following dependency is satisfied:

$$\forall X, Y \in V'\ (X \cap Y = \emptyset) \wedge \forall X \in V'\ \ (X \cap U_{n+1} = \emptyset \vee X \cap \bigcup_{i=1}^{n} L_i = \emptyset)$$

4.3 Canonical Form of Rabin Automaton

Lemma 3. *Consider an arbitrary deterministic stopping automaton $\mathcal{A}$ with index greater than 1, $\langle Q, \Sigma, -, \Delta, \Omega, A\rangle$, where $\Omega = \{(L_1, U_1), \ldots, (L_n, U_n), (\emptyset, U_{n+1})\}$. Moreover we require that if its index is even, the automaton does not have the accessibility property. Then there exists a sequence $(\mathcal{B})_{i\in\{1,\ldots,k\}}$ ($k \geq 2$) of deterministic stopping automata whose composition $\mathcal{B} = \mathcal{B}_1 \oplus \ldots \oplus \mathcal{B}_k = \langle Q', \Sigma, -, \Delta', \Omega', A'\rangle$, is proper and satisfies conditions:*

- *$\forall i \in \{1, \ldots, k\}\ \mathcal{B}_i = \langle Q_i, \Sigma,\ s_i, \Delta_i, \Omega_i, A_i\rangle$ has the accessibility property,*
- *$\forall i \in \{1, .., k\}\ Ind(\mathcal{B}_i) \leq Ind\ (\mathcal{A})$ (if $Ind(\mathcal{A})$ is odd, $Ind(\mathcal{B}_i) < Ind\ (\mathcal{A})$),*
- *$\exists f : Q \overset{1-1}{\to} Q'\ (\forall\, p \in Q\ (\mathcal{A}[p] \simeq \mathcal{B}[f(p)]))$*

Moreover, if $\Omega' = \{(L'_1, U'_1), \ldots, (L'_m, U'_m), (\emptyset, U'_{m+1})\}$, then the function f from the last condition satisfies a dependency:

$$\bigcup_{j=1}^{m} L'_j \subseteq f(\bigcup_{i=1}^{n} L_i). \tag{8}$$

$$(\, p \in f(\bigcup_{i=1}^{n} L_i) \setminus \bigcup_{j=1}^{m} L'_j\ \wedge\ \exists\, i \in \{1, \ldots, k\}\ f(p) \in \mathcal{B}_i\ ) \Rightarrow Ind(\mathcal{B}_i) = 1 \tag{9}$$

For the proof see [10].

Lemma 4. *Let* $\mathcal{A}=\mathcal{A}_1 \oplus \ldots \oplus \mathcal{A}_k = \langle Q, \Sigma, q, \Delta, \Omega, A\rangle$*, where* $\Omega = \{(L_1, U_1), \ldots, (L_n, U_n), (\emptyset, U_{n+1})\}$*, be a proper composition of frontier deterministic stopping automata. Assume that there exists* $i \in \{1, \ldots, k\}$ *such that the automaton* $\mathcal{A}_i$ *is of the form :* $\langle Q_i, \Sigma, -, \Delta_i, \Omega_i, A_i\rangle$ *and its index is greater than 1. Let:* $\Omega_i = \{(L^i_1, U^i_1), \ldots, (L^i_{n_i}, U^i_{n_i}),\ (\emptyset, U^i_{n_i+1})\}$*. Suppose furthermore that there exists a proper composition of frontier deterministic stopping automata* $\mathcal{B} = \mathcal{B}_1 \oplus \ldots \oplus \mathcal{B}_l = \langle Q', \Sigma, -, \Delta', \Omega', A'\rangle$ *and a one-to-one function* f *from* Q_i *to* Q'*, which satisfies the two conditions 8, 9 and such that for each* $p \in Q_i$ *the automaton* $\mathcal{A}_i[p]$ *is equivalent to the automaton* $\mathcal{B}[f(p)]$*. Then the substitution* $\mathcal{B} \overset{f}{\mapsto} \mathcal{A}_i$ *into the composition* $\mathcal{A}_1 \oplus \ldots \oplus \mathcal{A}_k[q]$ *is proper and equivalent to the composition* $\mathcal{A}$*.*

For the proof see [10].

Theorem 1. *For an arbitrary deterministic stopping automaton* $\mathcal{A}$ *with the index greater than 1 there exist two sequences of stopping automata having the accessibility property* $(\mathcal{A}_i)_{i \leq l}$ *and* $(\mathcal{B}_j)_{j \leq k}$*, where* $l \cdot k \neq 0$*, such that the composition* $\mathcal{A}_1 \oplus \ldots \oplus \mathcal{A}_l \oplus \mathcal{B}_1 \oplus \ldots \oplus \mathcal{B}_k[q']$ *is equivalent to the automaton* $\mathcal{A}$ *for some specifically chosen state* q'*. Furthermore, this composition satisfies the following conditions:*

1. *for each* $i \in \{1, \ldots, l\}$ *the automaton* $\mathcal{A}_i$ *is frontier deterministic and its index is 1*
2. *for each* $i \in \{1, \ldots, k\}$ *the automaton* $\mathcal{B}_i$ *is deterministic, and its index is even*
3. $Ind(\mathcal{B}_1) \leq \ldots \leq Ind(\mathcal{B}_k)$
4. *for each* $i \in \{1, \ldots, k\}$ $\mathcal{B}_i = \langle\ Q_i, \Sigma, -, \Delta_i, \Omega_i, A_i\rangle$*, where* $\Omega_i = \{(L^i_1, U^i_1), \ldots, (L^i_{n_i}, U^i_{n_i})\}$*, and for any* $p, r \in \bigcup_{j=1}^{n_i} L^i_j$ *the automaton* $\mathcal{B}_i[p]$ *admits a gadget* $G(p, r)$*.*

Moreover, we can assume that the sequences $(\mathcal{A}_i)_{i \leq l}$ *and* $(\mathcal{B}_j)_{j \leq k}$*, satisfy an additional condition:* $l \leq 1$ *and* $k \leq |\bigcup_{i=1}^{n} L_i|$*.*
The representation of the automaton $\mathcal{A}$ *in the form of a composition satisfying the above conditions will be called* **the canonical form***.*

Proof. If the index of the automaton $\mathcal{A}$ is odd or it is even, but the automaton does not have the accessibility property then according to the lemma 3 there exists a sequence of deterministic stopping automata having the accessibility property $(\mathcal{C}_i)_{i \leq m}$, whose composition $\mathcal{C} = \mathcal{C}_1 \oplus \ldots \oplus \mathcal{C}_m[r]$ is equivalent to the automaton $\mathcal{A}$ for some state r. Assume that we can find in the set $\{1, \ldots, m\}$ such j that $\mathrm{Ind}(\mathcal{C}_j)$ is odd and greater than 1. If we again use the lemma 3, this time for the automaton $\mathcal{C}_j$, and obtain a composition $\mathcal{D} = \mathcal{D}_1 \oplus \ldots \oplus \mathcal{D}_{m'}$ and a function f, which establishes equivalence between $\mathcal{C}_j$ and the composition $\mathcal{D}$, then by lemma 4 we can construct the substitution $\mathcal{D} \overset{f}{\mapsto} \mathcal{C}_j$ into the composition $\mathcal{C}$. Thus we obtain a new proper composition, whose all elements have the accessibility property and which is equivalent to the automaton $\mathcal{A}$. This procedure can be repeated as long as among the components of the composition there are automata, whose index is even and greater than 1. Let us note however that this can be done only finitely many times since according to the lemma 3 each time we introduce into the composition automata with smaller indices than the index of the replaced automaton. Eventually we obtain a sequence $(\mathcal{E}_i)_{i\leq m''}$ of deterministic stopping automata having the accessibility property, whose index is either even or equal 1 and whose composition is equivalent to the automaton $\mathcal{A}$. Now let us take an arbitrary automaton from the above composition $\mathcal{E}_i$ with an even index (if there is such). Assume $\mathcal{E}_i = \langle Q', \Sigma, -, \Delta', \Omega', A'\rangle$, where $\Omega' = \{(L'_1, U'_1), \ldots, (L'_{n'}, U'_{n'})\}$. If for each $p \in \bigcup_{j=1}^{n'} L'_j$ the state p is irrelevant for the automaton $\mathcal{E}_i[p]$ then by lemma 2 there exists a frontier deterministic stopping automaton with index 1 equivalent to $\mathcal{E}_i$ (the construction from the lemma 2 does not depend on the initial state of the initial automaton, so there exists a function f establishing an equivalence between the above automata) Note that a single-element sequence consisting of a Büchi automaton $\mathcal{E}'$ forms a proper composition and we can use the lemma 4 constructing a substitution $\mathcal{E}' \overset{f}{\mapsto} \mathcal{E}_i$ into the composition $\mathcal{E}_1 \oplus \ldots \oplus \mathcal{E}_{m''}$. We continue this procedure as long as it is possible. Next we split the sequence $(\mathcal{E}_i)_{i\leq m''}$ into two subsequences $(\mathcal{A}_i)_{i\leq l}$ and $(\mathcal{B}_j)_{j\leq k}$, putting into the first one automata with index 1 and to the latter one the automata with an even index, thus we obtain the representation of the automaton $\mathcal{A}$ in the form of the composition satisfying the conditions of the lemma. Let us observe that we can compute the composition of all component automata with index 1 and satisfy the condition $l \leq 1$. Moreover let us note that the fourth condition also holds. We know that for any $i \in \{1, \ldots, k\}$ there exists $p(i) \in \bigcup_{j=1}^{n_i} L^i_j$ such that the state $p(i)$ is relevant for the automaton $\mathcal{B}_i[p(i)]$. It follows from the lemma 1 that the automaton $\mathcal{B}_i[p(i)]$ admits a gadget $G(p(i), p(i))$. However since the automaton $\mathcal{B}_i[p(i)]$ has the accessibility property then it also admits gadgets of the form $G(p(i), q)$ for any $q \in \bigcup_{j=1}^{n_i} L^i_j$. Finally, by the definition of the accessibility property and the fact that the index of the automaton $\mathcal{B}_i$ is even, the fourth condition is satisfied. The remaining parts of the thesis are easy observations.

5 Main Results

Theorem 2. *Let $\mathcal{B} = \mathcal{A}_1 \oplus \ldots \oplus \mathcal{A}_l \oplus \mathcal{B}_1 \oplus \ldots \oplus \mathcal{B}_k$ be a canonical form of the deterministic stopping tree automaton $\mathcal{A}$ $l, k \geq 0$. Then $L(\mathcal{A})$ is in Büchi class if and only if $k = 0$.*

For the proof see [10].

Theorem 3. *For an arbitrary deterministic stopping automaton $\mathcal{A}$ we are able to decide if the language accepted by it is in Büchi class.*

Sketch of the proof. By the preceding theorem to prove the decidability of the considered problem we need to show that we can find a canonical form of an arbitrary deterministic stopping automaton. Therefore it suffices to prove that we are able to give a procedure deciding if a given state is irrelevant for some deterministic stopping automaton $\mathcal{A}$. Let us observe that if we had a procedure determining if a state is irrelevant for an automaton whose set of stopping states is empty, then our task would be completed. It follows from the fact that we can transform any stopping automaton into one of the above form. To do this we add to the sets of states a new state x, replace all stopping states by this state and finally add transitions (x, a, x, x) for any $a \in \Sigma$ and a condition $(\emptyset, \{x\})$ to the set of conditions. Thus constructed automaton with the empty set of stopping states has the following property: states which are irrelevant for it are also irrelevant for the initial automaton and vice versa.
Additionally, we have to be able to construct an automaton graph $\mathcal{G}_{\mathcal{A}}$. The proof of decidability of both of the problems is simple and uses the celebrated Rabin theorem on decidability of S2S logic [2]. For a complete proof see [10].

References

1. J.R. Büchi *On a decision method in restricted second order arithmetic*, in: Logic, Methodology and Philosophy of Science, Proc. (1960) International Congr. (1962), 1-11.
2. An.A. Muchnik *Igry na bieskoniecznych dieriewiach i awtomaty s tupikami. Nowoje dokazatielstwo razrieszimosti monadiczieskoj tieorii dwuch sliedowanij*, in: Siemiotika (1985), 16-40.
3. D. Niwiński *Fixed point characterization of infinite behavior of finite-state systems*, Fundamental Study, Theoretical Computer Science 189 Elsevier (1997), 1-69.
4. D. Niwiński I. Walukiewicz *Relating hierarchies of word and tree automata* , Proceedings of "15th Annual Symposium on Theoretical Aspects of Computer Science" (STACS), Paris, France (1998), 320-331.
5. M.O. Rabin *Decidability of second-order theories and automata on infinite trees*, Trans. Amer. Math. Soc. 141, (1969), 1-35.
6. W. Thomas *Automata on infinite objects*, in: Handbook of Theoretical Computer Science, Elsevier (1990) 133-191.
7. W. Thomas *Languages, automata and logic*, Handbook of Formal Languages, Vol.3, Springer-Verlag (1997).
8. T. Wilke, H. Yoo *Computing the Rabin index of a regular language of infinite words*, Information and Computation Vol.130 (1996), 61-70.
9. M. Otto, *Eliminating Recursion in the mu-Calculus*, in: Proceedings of STACS'99, LNCS 1563, Springer 1999, 531-540.
10. T.F.Urbański, *On decidability of some properties of finite automata on infinite trees*, Master of Sci. Thesis, Warsaw University, 1998 (in Polish), English version `http://zls.mimuw.edu.pl/~niwinski/Prace/TU.ps.gz`.