RAIRO. THEORETICAL INFORMATICS AND APPLICATIONS

OLIVIER CARTON
RAMÓN MACEIRAS

Computing the Rabin index of a parity automaton

RAIRO. Theoretical Informatics and Applications, tome 33, nº 6 (1999), p. 495-505
<http://www.numdam.org/item?id=ITA_1999__33_6_495_0>

Theoretical Informatics and Applications
Theoret. Informatics Appl. **33** (1999) 495–505

COMPUTING THE RABIN INDEX OF A PARITY AUTOMATON

Olivier Carton[1] and Ramón Maceiras[1]

Abstract. The Rabin index of a rational language of infinite words given by a parity automaton with n states is computable in time $O(n^2c)$ where c is the cardinality of the alphabet. The number of values used by a parity acceptance condition is always greater than the Rabin index and conversely, the acceptance condition of a parity automaton can always be replaced by an equivalent acceptance condition whose number of used values is exactly the Rabin index. This new acceptance condition can also be computed in time $O(n^2c)$.

AMS Subject Classification. 68Q45, 68Q68, 3D15.

1. Introduction

Since Büchi introduced automata on infinite words in [1], several acceptance conditions for paths have been considered. The acceptance condition presented by Muller in 1963 [8] explicitly specifies the set of infinitely often repeated states. From McNaughton's theorem [5], it is known that any rational set of infinite words is recognized by a deterministic automaton with a Muller acceptance condition. The Rabin condition [9] was first introduced for automata on infinite binary trees but has since been considered for automata on infinite words. This acceptance condition defines a hierarchy among rational sets of infinite words based on the number of accepting pairs required to recognize a given set of infinite words. This number of pairs is usually called the *Rabin index* of the set. It turns out that the classes of this hierarchy are particular classes of the more general hierarchy discovered by Wagner [13] and that this hierarchy has a topological interpretation. Another acceptance condition is the parity acceptance condition also known as the Rabin "chain" condition. It has been first introduced by Mostowski in [6]. It was

[1] Institut Gaspard Monge, Université de Marne-la-Vallée, 5 boulevard Descartes, Champs-sur-Marne, 77454 Marne-la-Vallée Cedex 2, France;
e-mail: `Olivier.Carton@univ-mlv.fr`

used independently to obtain forgetful strategies in [7] and [3]. This acceptance condition is also convenient for boolean operations [2].

Krishnan *et al.* showed that the computation of the Rabin index of languages given by deterministic Rabin (or Streett) automata is NP-complete [4]. The situation is different with Muller automata. The methods developed by Krishnan *et al.* can be used to design an algorithm to compute the Rabin index in polynomial time, provided the languages are given by deterministic Muller automata. Wilke and Yoo have presented an efficient algorithm that computes the Rabin index of a deterministic Muller automaton with n states and m accepting sets in time $O(m^2nc)$ where c is the cardinality of the alphabet [14, 15]. The upper bound obtained using the methods of [4] would be $O(m^2n^2c)$. In this paper, we introduce an algorithm which computes the Rabin index of a parity automaton in time $O(n^2c)$. At the same time, this algorithm also computes an acceptance condition for this automaton which uses the least number of values.

This paper is organized as follows. The basic definitions for automata on infinite words are recalled in Section 2. The Rabin index and the alternating chains used to compute it are defined in Section 3. Section 4 presents an algorithm to compute the Rabin index of a parity automaton and to minimizes its acceptance condition. Finally, it is proved in Section 5 that the algorithm is correct. The complexity of the algorithm is also studied in this section.

2. Automata on infinite words

In this paper, we consider automata recognizing sets of infinite words also called ω-words. We refer the reader to [11] for a complete introduction to such automata but we recall here the main definitions. A finite automaton $\mathcal{A}$ is an automaton (Q, A, E, q_0, Φ) where Q is a finite set of states, E is the set of transitions, q_0 is the unique initial state and Φ is the acceptance condition. All automata considered in this paper are deterministic. The accepting condition determines a family of sets of states which are said to be *accepting*. Subsets of states which are not accepting are said to be *rejecting*. For an infinite path γ in the automaton, we denote by $\inf(\gamma)$ the subset of states which appear infinitely often along γ. Since the number of states is finite, the subset $\inf(\gamma)$ is always nonempty. A path γ in the automaton is *successful* if it starts at the initial state and if $\inf(\gamma)$ is an accepting subset of states. Many different kinds of acceptance conditions have been studied in the literature. In this paper, we are mainly interested in automata with a parity acceptance condition but we also consider automata with a Muller or a Rabin acceptance condition.

We recall here briefly the definition of the different acceptance conditions that we consider in the paper.

A *Muller automaton* is a deterministic automaton $(Q, A, E, q_0, \mathcal{F})$ where the acceptance condition $\mathcal{F}$ is a family of subsets of states. A subset of states is then accepting if it belongs to the family $\mathcal{F}$. A Muller condition is thus an explicit description of the family of accepting subsets. Any automaton can be viewed as a Muller automaton whose acceptance condition $\mathcal{F}$ is the family of accepting subsets. McNaughton's theorem [5] states that any rational set of infinite words is recognized by a Muller automaton.

A *Rabin automaton* is a deterministic automaton $(Q, A, E, q_0, \mathcal{R})$ where the acceptance condition $\mathcal{R}$ is a family $\{(L_1, U_1), \dots, (L_m, U_m)\}$ of pairs of subsets of states. A subset R is then accepting if $R \cap L_i = \varnothing$ and $R \cap U_i \neq \varnothing$ for some pair (L_i, U_i) of the acceptance condition. Any rational set of infinite words is recognized by a Rabin automaton [9].

A *parity automaton* is a deterministic automaton (Q, A, E, q_0, π) where the acceptance condition π is a function from Q to $\mathbb{N}$ which associates an integer with each state of the automaton. The function π is naturally extended to subsets of states by setting $\pi(R) = \max_{q \in R} \pi(q)$ for any subset R of states. A subset R of states is then accepting if $\pi(R)$ is odd. Notice that a parity automaton is a particular case of a Rabin automaton. If the sets L_i and U_i are respectively defined by $L_i = \{q \mid \pi(q) \geq 2i\}$ and $U_i = \{q \mid \pi(q) = 2i - 1\}$, the Rabin acceptance condition $\{(L_1, U_1), \dots, (L_m, U_m)\}$ for $m = \lfloor (\pi(Q) + 1)/2 \rfloor$ is equivalent to the parity acceptance condition. Conversely, any rational set of infinite words is recognized by a parity automaton [6]. Parity automata are also called chain automata in the literature [2].

A subset R of states is said to be *essential* if it is equal to $\inf(\gamma)$ for some infinite path γ which starts at the initial state. Clearly, a subset R of states is essential iff there is a cycle c in the automaton, which is accessible from the initial state, such that the set of states encountered along the cycle c is exactly R. The successful paths are then defined by the accepting essential subsets of states. This justifies the terminology. Note that if two essential subsets R and S intersect non trivially, the union $R \cup S$ is also an essential subset.

3. Rabin index and alternating chains

In this section, we recall the definition of the Rabin index of a set of infinite words. We explain how this integer measures the complexity of a rational set of infinite words from the automata-theoric point of view and from the topological point of view. We also recall how it can be computed using alternating chains in automata.

The *Rabin index* $\operatorname{ind}(X)$ of a rational set X of infinite words is the minimal number of pairs needed in a Rabin acceptance condition to recognize the set X. More formally, the integer $\operatorname{ind}(X)$ is defined by

$$\operatorname{ind}(X) = \min\{\operatorname{card}(\mathcal{R}) \mid \exists \mathcal{A} = (Q, A, E, q_0, \mathcal{R}) \text{ such that } \mathcal{A} \text{ recognizes } X\}\cdot$$

The Rabin index of X measures the size of a Rabin acceptance condition needed to recognize X. It also has a topological interpretation. It is known that the rational sets of infinite words lie very low in the Borel hierarchy. More precisely, any rational set of infinite words is equal to a boolean combination of G_δ-sets. The Rabin index measures then the size of the boolean combination which is equal to X. The Rabin index of X is indeed the smallest n such that there are two sequences $(X_i)_{1\leq i\leq n}$ and $(Y_i)_{1\leq i\leq n}$ of n G_δ-sets such that $X = \bigcup_{i=1}^{n} X_i - Y_i$.

We now come to the definition of alternating chains in automata. This definition does not depend on a particular acceptance condition. A chain in $\mathcal{A}$ is an alternating chain with respect to set inclusion of accepting and rejecting subsets of states. More formally, a *chain* in $\mathcal{A}$ of length m is an increasing sequence

$$R_1 \subset R_2 \subset \cdots \subset R_m \tag{1}$$

of m essential subsets of Q such that R_1 is an accepting set and such that, for $1 \leq i \leq m$, R_i are alternately accepting and rejecting. This means that R_2 is rejecting, R_3 is accepting and so on. Notice that it is important in the definition that the subsets R_i are essential.

It is usually not assumed in the literature that the first set R_1 of a chain is accepting. Chains are usually called positive or negative according as R_1 is accepting or rejecting. However, we are in this paper only interested in positive chains and we always assume that R_1 is accepting. We denote by $m(\mathcal{A})$ the maximal length of (positive) chains in $\mathcal{A}$. By convention, we set $m(\mathcal{A}) = 0$ if there is no chain in $\mathcal{A}$, that is, if the automaton $\mathcal{A}$ has no accepting essential set. It is obvious by definition that $m(\mathcal{A})$ is finite for any finite automaton $\mathcal{A}$. Indeed, one has the inequality $m(\mathcal{A}) \leq \text{card}(Q)$ since the inclusions in (1) are strict. When the automaton $\mathcal{A}$ is a parity automaton (Q, A, E, q_0, π), this upper bound can be sharpened. Indeed, the sequence $\pi(R_1), \ldots, \pi(R_m)$ is then a strictly increasing sequence of integers and thus $m(\mathcal{A}) \leq \pi(Q)$.

Note that $m(\mathcal{A}) = 0$ iff $\mathcal{A}$ has no accepting essential set and that $m(\mathcal{A}) = 1$ iff the family of accepting essential sets is closed by taking superset. Thus $m(\mathcal{A}) = 0$ iff the set recognized by the automaton is empty and $m(\mathcal{A}) = 1$ iff the set recognized by the automaton is G_δ by Landweber's theorem.

The following theorem relates the maximal length $m(\mathcal{A})$ of chains in an automaton and the Rabin index of the set X recognized by the automaton [12,13]. It shows in particular that the Rabin index is computable.

Theorem 1 (Wagner 77). *The Rabin index of a rational set X of infinite words recognized by a deterministic automaton $\mathcal{A}$ is given by the expression:*

$$\text{ind}(X) = \lfloor (m(\mathcal{A}) + 1)/2 \rfloor$$

where $\lfloor \alpha \rfloor$ denotes the greatest integer not greater than α.

The previous theorem shows that the Rabin index can be computed. Indeed, there are finitely many chains in an automaton which can be effectively enumerated. Thus, the integer $m(\mathcal{A})$ can be effectively computed. However, the complexity of this computation highly depends on the acceptance condition. It was shown in [4] that the computation of the Rabin index is NP-complete for a Rabin automaton while it was shown in [14, 15] that it is polynomial for a Muller automaton. We prove in this paper that it is polynomial for a parity automaton. We describe a polynomial algorithm which takes a parity automaton (Q, A, E, q_0, π) and outputs the Rabin index of the set recognized by the automaton. The algorithm also outputs another acceptance condition π' such that (Q, A, E, q_0, π') recognizes the same set and which is minimal in a sense explained below.

The following definition captures the fact that two parity functions π and π' define the same set of accepting sets. Let $\mathcal{A} = (Q, A, E, q_0)$ be an automaton. Two functions π and π' are said to be *equivalent* iff for any essential set R, $\pi(R) \equiv \pi'(R) \bmod 2$. This definition is motivated by the fact that if π and π' are equivalent, then both parity automata $\mathcal{A} = (Q, A, E, q_0, \pi)$ and $\mathcal{A}' = (Q, A, E, q_0, \pi')$ have the same accepting essential sets and thus recognize the same set of infinite words.

The algorithm that we give in the next section computes the Rabin index of the set recognized by a parity automaton. It also outputs another parity acceptance condition whose existence is stated by the following theorem.

Theorem 2. *Let $\mathcal{A} = (Q, A, E, q_0, \pi)$ be a parity automaton recognizing a set X. There exists another function π' from Q to the set $\{0, \dots, m(\mathcal{A})\}$ which satisfies $\pi'(q) \leq \pi(q)$ for any state q and which is equivalent to π.*

For a proof, see [2]. The algorithm that we describe in the next section provides another proof of this result. Since the parity function π' given by the theorem is equivalent to π, the automaton $\mathcal{A}' = (Q, A, E, q_0, \pi')$ recognizes the same set of infinite words. The greatest integer used by the parity condition, that is $\pi(Q) = \max_{q \in Q} \pi(q)$ may be considered as the size of the condition. It corresponds to the number of pairs (with a factor 2) if this parity condition is viewed as a Rabin condition. We have already mentioned that for any parity automaton the inequality $m(\mathcal{A}) \leq \pi(Q)$ holds. Thus, the parity function π' given by the previous theorem uses the least number of values as possible. For a given Rabin automaton, it is not possible in general to modify the acceptance condition to have the minimal number of pairs (*i.e.*, the Rabin index), although it is possible to find another Rabin automaton recognizing the same language and which has the minimal number of pairs in its acceptance condition. In contrary, it is always possible to modify the acceptance condition of a parity automaton such that the greatest integer used by the new condition is minimal.

4. The algorithm

In this section, we describe the algorithm which computes the Rabin index of a set recognized by a parity automaton $\mathcal{A} = (Q, A, E, q_0, \pi)$. Furthermore, the algorithm computes the parity function π' given by Theorem 2. We have then the following theorem which is the main result of the paper.

Theorem 3. *The Rabin index of a set of infinite words recognized by a parity automaton $\mathcal{A} = (Q, A, E, q_0, \pi)$ and the reduced parity function π' which is equivalent to π can be computed in time $O(|Q|^2|A|)$.*

The algorithm is based on some results about parity automata that we first establish.

For an automaton $\mathcal{A} = (Q, A, E, q_0, \pi)$, the integer $m(\mathcal{A})$ is equal to the maximal length of chains in $\mathcal{A}$. We extend the function m to a function from $\mathcal{P}(Q)$ to $\mathbb{N}$ which assigns to every subset P of Q, the maximal length of a chain contained in P. More formally, $m(P)$ is the greatest integer m such that there is a chain $R_1, \ldots, R_m$ with $R_m \subset P$. By convention, we set $m(P) = 0$, if there is no chain contained in P, that is, if no accepting essential subset is included in P. Clearly, we have $m(\mathcal{A}) = m(Q)$. Note also that if $P' \subset P$, then $m(P') \leq m(P)$.

The algorithm is based on the following two propositions. The first proposition essentially states that if suffices to compute the values of $m(R)$ when R is a maximal essential subset.

Proposition 4. *Let $\mathcal{A} = (Q, A, E, q_0, \pi)$ be a parity automaton. Let P be a subset of Q and let $R_1, R_2, ..., R_n$ be the maximal essential subsets (with respect to set inclusion) included in P. One then has*

$$m(P) = \begin{cases} \max\{m(R_i) \mid 1 \leq i \leq n\} & \text{if } n > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Proof. Any essential subset of a chain in P is included, by definition, in a maximal essential subset included in P. □

We now introduce some definition. Let $\mathcal{A} = (Q, A, E, q_0, \pi)$ be a parity automaton and let R be a subset of states. We define the *derivative* R' of R by $R' = \{q \in R \mid \pi(q) < \pi(R)\}$. The derivative R' of R is thus a strict subset of R and if $\pi(R) > 0$, it satisfies $\pi(R') < \pi(R)$. Furthermore, the complement $R - R'$ of R' in R is equal to $\{q \in R \mid \pi(q) = \pi(R)\}$.

The following proposition relates the value $m(R)$ with the value $m(R')$.

Proposition 5. *Let $\mathcal{A} = (Q, A, E, q_0, \pi)$ be a parity automaton and let R be an essential subset of Q. One then has*

$$m(R) = \begin{cases} 0 & \text{if } \pi(R) = 0 \\ m(R') & \text{if } \pi(R) - m(R') \text{ is even} \\ m(R') + 1 & \text{otherwise.} \end{cases}$$

Proof. If $\pi(R) = 0$, then any essential subset included in R is rejecting and there is no chain contained in R. Thus, one has $m(R) = 0$.

Let m be the integer $m(R')$. By definition of m, there is a chain $R_1, \ldots, R_m$ contained in R'. Since this chain is also contained in R, one has $m(R) \geq m$. Conversely, suppose that $R_1, \ldots, R_{m+1}$ is a chain contained in R. The definition of m implies that R_{m+1} is not included in R'. Thus, the intersection $R_{m+1} \cap (R - R')$ is nonempty and $\pi(R_{m+1}) = \pi(R)$.

We first suppose that $\pi(R) - m$ is even. Since both integers m and $\pi(R)$ have the same parity, both sets R_m and R are either accepting or rejecting. If

$R_1, \ldots, R_{m+1}$ is a chain contained in R, then $\pi(R_{m+1}) = \pi(R)$. Therefore, both sets R_m and R_{m+1} are either accepting or rejecting. This is a contradiction with the alternation of the chain.

We now suppose that $\pi(R) - m$ is odd. Since the integers m and $\pi(R)$ do not have the same parity, one of the two sets R_m and R is accepting and the other is rejecting. The sequence $R_1, \ldots, R_m, R$ is then a chain contained in R. This shows that $m(R) \geq m + 1$. Conversely, if $R_1, \ldots, R_{m+2}$ is a chain contained in R, then $\pi(R_{m+1}) = \pi(R)$ and since $R_{m+1} \subset R_{m+2} \subset R$, one also has $\pi(R_{m+2}) = \pi(R_{m+1})$. This is a contradiction with the alternation of the chain. □

These propositions above motivate the following algorithm computing the Rabin index of a parity automaton and minimizing its acceptance condition. The states which are not accessible from the initial state can be found by a depth first search in time $O(|Q||A|)$. These states are removed and we can assume that any state of the automaton is accessible from the initial state.

The algorithm RabinIndex is the following.

Algorithm RabinIndex
Input $\mathcal{A} = (Q, A, E, q_0, \pi)$
  for each q **in** Q **do**
    $\pi'[q] \leftarrow 0$
  $m(\mathcal{A}) \leftarrow M(Q)$
Output $\lfloor (m(\mathcal{A}) + 1)/2 \rfloor$ **and** $\mathcal{A}' = (Q, A, E, q_0, \pi')$.

The algorithm inputs a parity automaton. The recursive function M computes the value of $m(Q)$. It also computes the reduced acceptance condition π'. The algorithm uses then Theorem 1 to compute the Rabin index. It outputs both the Rabin index and the automaton with the reduced acceptance condition. The function M is computed as follows.

function $M(P)$
  max $\leftarrow 0$
  for each maximal essential subset R_i included in P **do**
    if $\pi(R_i) = 0$ **then**
      $m \leftarrow 0$
    else
      $m \leftarrow M(R'_i)$
      if $\pi(R_i) - m$ is odd **then** $m \leftarrow m + 1$
    for each $q \in R_i - R'_i$ **do**
      $\pi'[q] \leftarrow m$
    max $\leftarrow$ max(max, m)
Return max.

5. Correctness and complexity of the algorithm

In this section, we prove that the algorithm described in the previous section is correct and we analyse its running time. This analysis is based on a labeling of the nodes of the call tree of the function M. This labeling is described in the next section.

5.1. Labeling of the call tree of M

The algorithm is mainly based on the recursive function M whose call tree we now study. Each node of the call tree corresponds to a call $M(P)$ for some subset P of Q. The root of the tree corresponds to the call $M(Q)$. The sons of a node corresponding to a call $M(P)$ correspond to the calls $M(R'_1), \ldots, M(R'_m)$ where $R_1, \ldots, R_m$ are the maximal essential subsets included in P. Notice that the sets R_i are pairwise disjoint. Indeed, if two essential subsets intersect non trivially, their union is still an essential subset and they cannot be maximal in P unless they are equal. Notice also that if $\pi(R_i) > 0$, the derivative R'_i is a strict subset of R_i and thus it is also a strict subset of P. This shows that if the nodes along a branch of the tree correspond to the sequence $M(P_1), \ldots, M(P_k)$ of calls, one has the sequence $P_1 \supsetneq \cdots \supsetneq P_k$ of strict inclusions. Furthermore, since $\pi(R') < \pi(R) \leq \pi(P)$, the sequence $\pi(P_1), \ldots, \pi(P_k)$ is a strictly decreasing sequence of integers. Recall that two sons of the same node correspond to calls $M(P_1)$ and $M(P_2)$ with $P_1 \cap P_2 = \varnothing$. Combining these two properties, one has that there is at most one node corresponding to a call $M(P)$ for any nonempty subset P. Furthermore the nodes corresponding to a call $M(\varnothing)$ are leafs of the trees and these calls always return 1. In the sequel, we identify a node of the call tree with the corresponding call.

We now define a labeling of the nodes of the tree. This labeling will allow us to show that the number of nodes in the whole tree is bounded by the number of states. It will also be used to show the correctness of the algorithm. We label each node of the tree which is different from the root with a nonempty subset of states. Let $M(P)$ be a node of the tree and let $M(R')$ be one of its sons where R is one of the maximal essential subsets included in P. We label the node $M(R')$ with the subset $R - R'$ of states. Observe that this label is a nonempty subset since the derivative R' is a strict subset of R whenever $\pi(R) > 0$. Observe also that the label of the call $M(R')$ is disjoint from R' but it is included in R. By the chain property along the branch, if the node $M(P_1)$ is an ancestor of the node $M(P_2)$, the label of $M(P_2)$ is included in a maximal essential subset R included in P_1.

We claim that the labels of the nodes are pairwise disjoint. Consider two different nodes $M(P_1)$ and $M(P_2)$ of the call tree. Either, these two nodes are on the same branch of the tree and one of them is the ancestor of the other or they have a least common ancestor $M(P)$ in the tree which is different from both $M(P_1)$ and $M(P_2)$.

In the former case, one may suppose that $M(P_1)$ is an ancestor of $M(P_2)$. The label of $M(P_2)$ is then included in P_1 which is disjoint of the label of $M(P_1)$. The labels of the two nodes $M(P_1)$ and $M(P_2)$ are then disjoint.

In the latter case, there are two different maximal essential subsets R_1 and R_2 included in P such that the labels of the call $M(P_1)$ and $M(P_2)$ are respective subsets of R_1 and R_2. The labels of the two nodes $M(P_1)$ and $M(P_2)$ are then disjoint since R_1 and R_2 are disjoint.

5.2. Correctness

We prove in this section that the value returned by any call $M(P)$ of the function M is $m(P)$ and that the function π' which is globally computed by all the recursive calls made by $M(Q)$ satisfies the properties of Theorem 2.

We first claim that the function M computes $m(P)$. The computation of the function M is based on Propositions 4 and 5. The outer loop of the function M computes the maximum value of $M(R_i)$ for all maximal essential subsets R_i included in P as in Proposition 4. The value of $M(R_i)$ is, as in Proposition 5, computed from the value of $M(R'_i)$ which is itself computed by a recursive call of the function M. The function M terminates since any recursive call is made for a strictly smaller set. We actually prove below that the total number of recursive calls is bounded by the number of states of the automaton.

We now study the function π' which is computed by the function M. We claim that for each state q, there is at most one assignment $\pi'[q] \leftarrow m$ in the running of M. If the value $\pi'(q)$ is set by a call $M(P)$, the state q belongs to $R - R'$ where R is a maximal essential subset included in P. The state q belongs then to the label of the son $M(R')$ of $M(P)$. Since the labels of the nodes are pairwise disjoint, there is at most one such assignment. Furthermore, $\pi'(q)$ is then set to $m(R)$ by the previous discussion.

We now claim that the parity function π' computed by the function M is equivalent to π and that $\pi'(q) \leq \pi(q)$ for any state q of the automaton.

We need the following lemma which relates the functions m and π.

Lemma 6. *Let $\mathcal{A} = (Q, A, E, q_0, \pi)$ be a parity automaton and let R be an essential subset of Q, we then have*

$$m(R) \leq \pi(R)$$
$$m(R) \equiv \pi(R) \mod 2.$$

Proof. Let m be $m(R)$. By definition, there is a chain $R_1, \dots, R_m$ contained in R. The sequence $\pi(R_1), \dots, \pi(R_m)$ is a strictly increasing sequence of integers and thus $m(R) \leq \pi(R)$. Furthermore, if m and $\pi(R)$ do not have the same parity, the sequence $R_1, \dots, R_m, R$ is a chain of length $m+1$ which contradicts the definition of m. Thus one has $m(R) \equiv \pi(R) \mod 2$. □

We are now able to prove the two statements about the function π'. We first prove that $\pi'(q) \leq \pi(q)$ for any state q. Let q be state of the automaton. If $\pi'(q) = 0$, the inequality trivially holds. Otherwise, there is a unique node $M(P)$ in the call tree such that q belongs to $R - R'$ for some maximal essential subset R included in P. Since $q \in R - R'$, one has $\pi(q) = \pi(R)$. The value $\pi'(q)$ is then set to $m(R)$ and the inequality holds by the previous lemma.

We finally prove that the functions π and π' are equivalent, that is, $\pi'(S) \equiv \pi(S) \mod 2$ for any essential subset S of the automaton. Let S be an essential subset of the automaton. Note first that if S is included in a subset P, it is then included in one of the maximal essential subsets included P and it is disjoint from all others. By definition, S is included in at least one of the maximal essential subsets. If S intersects non trivially two essential subsets R_1 and R_2, the three subsets $S \cup R_1$, $S \cup R_2$ and $S \cup R_1 \cup R_2$ are also essential. Thus R_1 and R_2 cannot be maximal unless $R_1 = R_2$ and $S \subset R_1$. Furthermore, if S is included in an essential subset R, one has $S \subset R'$ iff $\pi(S) < \pi(R)$ since $R - R' = \{q \mid \pi(q) = \pi(R)\}$. It follows that there is a unique branch $M(P_1), \ldots, M(P_k)$ of nodes in the tree such that $P_1 = Q$, $P_1 \supsetneq \cdots \supsetneq P_k \supseteq S$ and $\pi(P_1) > \cdots > \pi(P_k) = \pi(S)$. Thus, all assignments to $\pi'(q)$ for the states of S are made by calls which belong to the subtree rooted in $M(P_k)$. There is then a maximal essential subset R of P_k such that $S \subseteq R$. Since $\pi(S) = \pi(R) = \pi(P_k)$, the intersection $S \cap (R - R')$ is nonempty. For any state q of $S \cap (R - R')$, $\pi'(q)$ is set to $m(R)$. For any state q of $S \cap R'$, $\pi'(q)$ is set to a value $m(P)$ for some subset P of R' and thus $\pi'(q) \leq m(R)$ since $P \subseteq R$. Finally, $\pi'(S)$ is equal to $m(R)$. By the previous lemma, one has $\pi'(S) \equiv \pi(S) \mod 2$.

5.3. Complexity

The running time of the algorithm is the running time of the function M. The function needs to compute the maximal essential sets included in the subset P. Let G_P be the oriented graph obtained by restricting the automaton to the states of P. A subset R is a maximal essential subset included in P iff it is a strongly connected component of G_P which is not reduced to a single vertex with no loop. The graph G_P can be computed in time $O(|Q||A|)$ and by Tarjan's algorithm [10], its strongly connected components can be computed in time $O(|Q||A|)$. Finally, the maximal essential subsets included in P can be computed in time $O(|Q||A|)$.

To analyze this running time, it remains to upper bound the number of recursive calls made by M. Since the labels of the nodes are pairwise disjoint, the number of nodes in the call tree is bounded by $|Q|$. We have then proved that the running time of the algorithm is bounded by $O(|Q|^2|A|)$ as stated in Theorem 3.

References

[1] J.R. Büchi, On a decision method in the restricted second-order arithmetic, in *Proc. Int. Congress Logic, Methodology and Philosophy of science, Berkeley 1960*, Stanford University Press (1962) 1–11.

[2] O. Carton, Chain automata. *Theoret. Comput. Sci.* **161** (1996) 191–203.

[3] E.A. Emerson and C.S. Jutla, Tree automata, Mu-calculus and determinacy, in *Proc. 32th Symp. on Foundations of Computer Science* (1991) 368–377.

[4] S.C. Krishnan, A. Puri and R.K. Brayton, Structural complexity of ω-languages, in *STACS '95*, Springer-Verlag, *Lectures Notes in Comput. Sci.* **900** (1995) 143–156.

[5] R. McNaughton, Testing and generating infinite sequences by a finite automaton. *Inform. Control* **9** (1966) 521–530.

[6] A. Mostowski Regular expressions for infinite trees and a standard form for automata, in *Computation theory*, A. Skowron, Ed., Springer-Verlag, Berlin, *Lectures Notes in Comput. Sci.* **208** (1984) 157–168.
[7] A. Mostowski, Hierarchies of weak automata and weak monadic formulas. *Theoret. Comput. Sci.* **83** (1991) 323–335.
[8] D. Muller, Infinite sequences and finite machines, in *Switching Theory and Logical Design*, P. of Fourth Annual IEEE Symp., Ed. (1963) 3–16.
[9] M.O. Rabin, Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. Soc.* **141** (1969) 1–35.
[10] R.E. Tarjan, Depth first search and linear graphs. *SIAM J. Comput.* **1** (1972) 146–160.
[11] W. Thomas, Automata on infinite objects, in *Handbook of Theoretical Computer Science*, J. van Leeuwen, Ed., **B** (Elsevier, 1990) Chapter 4, pp. 133–191.
[12] K. Wagner, Eine topologische Charakteriesierung einiger Klassen regulärer Folgenmengen. *Elektron. Informationsverarb. Kybernet.* **13** (1977) 505–519.
[13] K. Wagner, On ω-regular sets. *Inform. Control* **43** (1979) 123–177.
[14] T. Wilke and H. Yoo, Computing the Wadge degree, the Lipschitz degree, and the Rabin index of a regular language of infinite words in polynomial time, in *Trees in Algebra and Programming – CAAP '95* P. M. *et al.*, Ed., Springer-Verlag, *Lectures Notes in Comput. Sci.* **915** (1995) 288–302.
[15] T. Wilke and H. Yoo, Computing the Rabin index of a regular language of infinite words. *Inform. Comput.* **130** (1996) 61–70.

Communicated by J. Berstel.
Received October, 1997 and May, 1999. Accepted July, 1999.