Weak Muller Acceptance Conditions for Tree Automata *

Salvatore La Torre[1,2], Aniello Murano[2,3], and Margherita Napoli[2]

[1] University of Pennsylvania
[2] Università degli Studi di Salerno
[3] Rice University
{sallat,murano,napoli}@unisa.it

Abstract. Over the last decades the theory of finite automata on infinite objects has been an important source of tools for the specification and the verification of computer programs. Trees are more suitable than words to model nondeterminism and thus concurrency. In the literature, there are several examples of acceptance conditions that have been proposed for automata on infinite words and then have been fruitfully extended to infinite trees (*Büchi*, *Rabin*, and *Muller* conditions). The type of acceptance condition can influence both the succinctness of the corresponding class of automata and the complexity of the related decision problems. Here we consider, for automata on infinite trees, two acceptance conditions that are obtained by a relaxation of the Muller acceptance condition: the *Landweber* and the *Muller-Superset* conditions. We prove that Muller-Superset tree automata accept the same class of languages as Büchi tree automata, but using more succinct automata . Landweber tree automata, instead, define a class of languages which is not comparable with the one defined by Büchi tree automata. We prove that, for this class of automata, the emptiness problem is decidable in polynomial time, and thus we expand the class of automata with a tractable emptiness problem.

1 Introduction

Since its early days the theory of finite automata had an astonishing impact in computer science. Several models of automata have been extensively studied and applied to many fields. In the sixties, with their pioneering work, Büchi [1, 2], McNaughton [11], and Rabin [12] enriched this theory by introducing finite automata on infinite objects. The connections between such automata and the logic theories have been fruitfully investigated and have originated automata-theoretic approaches to reduce decision problems in the field of mathematical logics to automata decision problems.

* This research was partially supported by the NSF award CCR99-70925, NSF grant CCR-9988322, SRC award 99-TJ-688, DARPA ITO Mobies award F33615-00-C-1707, NSF ITR award, and the MURST in the framework of project "Metodi Formali per la Sicurezza" (MEFISTO)

Automata on infinite words and trees turned out to be very useful for those areas of computer science where nonterminating computations are studied. They give a unifying paradigm to specify, verify, and synthesize nonterminating systems [7, 15, 16]. A system specification can be translated to an automaton, and thus, questions about systems and their specifications are reduced to decision problems in the automata theory. For example, the satisfiability of a specification and the correctness of a system with respect to its specification can be often reduced to the nonemptiness problem and the containment of languages accepted by automata. It is thus important to study classes of automata for which checking for the emptiness of a language is not computationally expensive and the closure under complementation and intersection hold.

As abstract models of systems, trees are more suitable than words to model the nondeterminism, which is also useful to model concurrent programs (nondeterministic interleaving of atomic processes). It is worth noticing that some concurrent programs, such as operating systems, communication protocols, and air-traffic control systems, are intrinsically nondeterministic and nonterminating. Moreover, by using trees we can express the existential path quantifier, and thus we are able to express lower bounds on nondeterminism and concurrency. This feature turns out to be greatly helpful in applications such as program synthesis [3, 4].

In the literature, several acceptance conditions on infinite words have been fruitfully extended to infinite trees, such as Büchi, Muller, and Rabin conditions. The kind of acceptance condition we choose usually influences both the succinctness of the model and the complexity of the decision algorithms. While for Büchi tree automata the emptiness problem is decidable in polynomial time, for Rabin tree automata it is NP-complete. On the other hand, Büchi tree automata are not closed under language complementation, while Rabin tree automata are. Since Rabin tree automata are strictly more expressive than Büchi tree automata, in terms of the class of accepted languages, it is worth searching for new models of automata with interesting closure properties and tractable decision problems, that capture languages besides those characterized by the Büchi paradigm.

For automata on infinite objects, the acceptance is defined with respect to the set of states which are visited infinitely often while reading the input. For example, for a Büchi tree automaton some of the states are declared accepting, and a tree t is accepted if and only if on all the paths of t at least an accepting state is visited infinitely often. For Muller tree automata, a family of set of states $F = \{F_1, F_2, \ldots, F_n\}$ is declared accepting, and a tree t is accepted if and only if on each path of t the set of states which are visited infinitely often belongs to F, that is, it is one of the accepting sets. In this paper, we study two new acceptance conditions for tree automata: *Landweber* and *Muller-Superset* acceptance conditions. These conditions are obtained by relaxing the Muller condition in the following way. The Landweber condition requires that, on each path of the input tree, the set of states which are visited infinitely often is contained in one of the accepting sets, while the Muller-Superset condition requires the opposite, that is, on each path of the input tree, one of the accepting sets is contained in the set of states which are visited infinitely often. With Landweber tree automata, we extend to infinite trees the acceptance condition introduced by Landweber in 1969, relatively to deterministic finite automata on infinite words [8]. Hossley studied the nondeterministic version of such automata [6]. Here, we study both the deterministic and the nondeterministic version of the corresponding tree automata.

For Landweber tree automata, we prove that the class of languages which are accepted by the deterministic model is strictly contained in the class defined by the nondeterministic one. We also prove that both classes are closed under union and intersection but not under complementation. We compare these classes to those accepted by Büchi and Muller tree automata in both deterministic and nondeterministic paradigms. It is also worth noticing that the class of languages accepted by Büchi tree automata is not comparable with that accepted by Landweber tree automata. We prove that the emptiness problem for Landweber tree automata is decidable in polynomial time. From the result on the comparison with Büchi tree automata, we thus obtain an alternative class of tree languages for which there exists a polynomial time decision algorithm.

The class of languages accepted by Muller-Superset tree automata turns out to coincide with the class of languages accepted by Büchi tree automata, in both the deterministic and the nondeterministic versions. An interesting feature of this paradigm is that automata from this class are usually more succinct than Büchi tree automata. We prove that for every language L accepted by a minimal Muller-Superset tree automata S, L is accepted by a minimal Büchi tree automata B such that $size(S) \leq size(B) \leq 2^{O(size(S))}$.

The rest of the paper is organized as follows. In Section 2, we give the definitions and recall some results on the theory of finite automata on infinite trees. In Section 3, we study the Muller-Superset tree automata and compare them with the Büchi tree automata. In Section 4, we extend the Landweber acceptance condition to tree automata, and study the corresponding model with a major emphasis on the main closure properties and the comparison between deterministic and nondeterministic paradigms. Relationships among Büchi, Landweber and Muller classes of languages are studied in Section 5. In Section 6, we prove that the emptiness problem for Landweber tree automata is decidable in polynomial time. Finally, we give a few conclusions in Section 7.

2 Automata on ω-trees

In this section, we introduce some notations that will be used in the rest of this paper. We also recall the definitions and the main results concerning Büchi, Muller, and Rabin tree automata.

Let Σ be a finite alphabet and $\textsc{Dom} = \{0, 1, \ldots, k-1\}^*$. We define an infinite k-ary Σ-tree t as a map $t : \textsc{Dom} \rightarrow \Sigma$. In the following, unless differently stated, an infinite k-ary Σ tree will be referred simply as a tree. For each tree t, the element in $\textsc{Dom}$ are the nodes of the tree and the empty word ϵ corresponds to the root. If u is a node of a tree then ui is the i-th child of the node u. We say that a symbol $a \in \Sigma$ occurs in a tree t if there exists $u \in \textsc{Dom}$ such that $t(u) = a$. Let u, $v \in \textsc{Dom}$, we say that *u precedes v*, denoted as $u < v$, if there exists an $x \in \textsc{Dom}$ such that $v = ux$. Let $\pi \subseteq \textsc{Dom}$, π is a path of t if it is a maximal subset of $\textsc{Dom}$ linearly ordered by $<$. If $\pi \subseteq \textsc{Dom}$ is a path of t, then t/π denotes the restriction of the function t to the set π. We say that a symbol $a \in \Sigma$ occurs in a tree t if there exists $u \in \textsc{Dom}$ such that $t(u) = a$. Moreover, we say that a occurs infinitely often in t/π if there exists an infinite number of nodes $u \in \pi$ such that $t(u) = a$. Let t be a tree and π a path of t, with $Inf(t/\pi)$ we denote the set $\{a \in \Sigma \,|\, a \text{ occurs infinitely often in } t/\pi\}$. Given a tree t and a node $u \in \textsc{Dom}$, we define the *subtree* of t rooted in u as the tree t_u such that $t_u(v) = t(uv)$ for $v \in \textsc{Dom}$. Let Σ be a finite alphabet, we denote by T^ω_Σ the set of Σ-valued trees. A language is a subset of T^ω_Σ. In the following we will deal exclusively with infinite binary trees ($\textsc{Dom} = \{0, 1\}^*$). All the results we obtain also hold for k-ary trees. According to the definition of a subtree, we use t_0 and t_1 to denote, respectively, the left and right subtree of t. Moreover, we denote by π_0 the leftmost path in a tree, that is $\pi_0 = \{0\}^*$, and by π_1 the rightmost one, i.e. $\pi_1 = \{1\}^*$.

Now we give some basic definitions from the theory of automata on infinite trees. We start by recalling some well-known acceptance conditions. In the literature several acceptance conditions on infinite trees have been considered. Here we recall those defining Büchi, Muller, and Rabin automata. Other paradigms such as Streett automata, turned out to be equivalent to Muller automata.

Definition 1. *A finite automaton on infinite trees (TA) is a tuple $A = \langle Q, \Sigma, \delta, Q_0, F\rangle$ where $Q \neq \emptyset$ is a finite set of states, Σ is an alphabet, $\delta \subseteq Q \times \Sigma \times Q \times Q$ is the transition relation, $Q_0 \subseteq Q$ is the set of initial states, and F is an acceptance condition.*

A deterministic automaton on infinite trees (DTA) is a TA with $|Q_0| = 1$ and δ is a total function $\delta : Q \times \Sigma \to Q \times Q$. To describe the behaviour of a TA, we recall the notion of run.

Definition 2. *Let $A = \langle Q, \Sigma, \delta, Q_0, F\rangle$ be a TA and $t : \textsc{Dom} \to \Sigma$ be an infinite tree. A run r of A on t is a tree $r : \textsc{Dom} \to Q$ such that $r(\epsilon) \in Q_0$, and $(r(u), t(u), r(u0), r(u1)) \in \delta$, for all $u \in \textsc{Dom}$.*

With $Run_A(t)$ we denote the set of runs of A on t. Clearly, if A is deterministic, then $|Run_A(t)| = 1$. An infinite tree is accepted by A if there exists a *successful* run r of A on t, that is r satisfies the acceptance conditions on all its paths. Moreover, with $T(A)$ we denote the language recognized by A.

The Büchi tree automaton (BTA) was defined by Rabin [13] who gave to these automata the name of *special automata*. A (D)BTA B is a (D)TA with a Büchi acceptance condition expressed by a set of states F. This condition requires that at least one accepting state occurs infinitely often. Thus, given a $t \in T^\omega_\Sigma$, a run $r \in Run_B(t)$ is successful if and only if for each path π of r, $Inf(r/\pi) \cap F \neq \emptyset$. With (D)BTA we denote also the class of languages accepted by (deterministic) Büchi tree automata.

The Muller condition is given as a family of sets of states $F = \{F_1, \ldots, F_m\}$ and requires that the states that repeat infinitely often are exactly the states from one of accepting sets. Thus, a (deterministic) Muller tree automaton ((D)MTA) M is a (D)TA with family $F = \{F_1, \ldots, F_m\}$. Given a $t \in T^\omega_\Sigma$, a run $r \in Run_M(t)$ is successful if and only if for each path π of r, there exists a set $F_i \in F$ such that $Inf(r/\pi) = F$. With (D)MTA we denote also the class of languages accepted by (deterministic) Muller tree automata.

Finally, a (deterministic) Rabin tree automaton ((D)RTA) R is a (D)TA along with a family of set of states $F = \{(F_1, G_1), \ldots, (F_m, G_m)\}$. Given a tree t, a run $r \in Run_R(t)$ is successful if and only if for each path π of r, there exists a pair $(F_i, G_i) \in F$ such that $Inf(r/\pi) \cap F_i \neq \emptyset$ and $Inf(r/\pi) \cap G_i = \emptyset$. With RTA we denote also the class of languages accepted by a Rabin tree automata.

We recall the main results for the introduced classes of tree automata. Since the class of languages accepted by Rabin tree automata coincides with that accepted by Muller tree automata, when we compare the classes of languages we will refer only to MTA. In Figure 1 we summarize the relationships among the considered classes of tree automata [9, 10, 13].

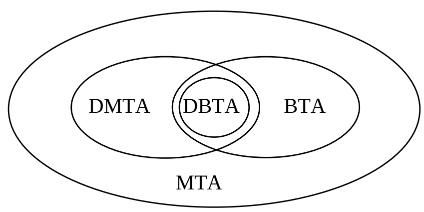

Fig. 1. Relationships among (D)BTA and (D)MTA.

In Figure 2 we list some languages along with their ranking relatively to the classification illustrated in Figure 1. For all of these languages, we assume that $\Sigma = \{a, b\}$. We will use them in the following.

Languages	Ranking
$T_1 = \{t \in T^\omega_\Sigma \mid \exists\ \pi$ such that $a \notin Inf(t/\pi)\}$	BTA - DMTA
$T_2 = \{t \in T^\omega_\Sigma \mid \forall\ \pi$, either $a \notin Inf(t/\pi)$ or $b \notin Inf(t/\pi)\}$	$DMTA - BTA$
$T_3 = \{t \in T^\omega_\Sigma \mid \forall\ x \in \pi_0,\ t/\pi_0(x) = a\}$	$DBTA$
$T_4 = \{t \in T^\omega_\Sigma \mid \forall\ \pi,\ a \in Inf(t/\pi)$ and $b \in Inf(t/\pi)\}$	$DBTA$
$T_5 = \{t \in T^\omega_\Sigma \mid a \notin Inf(t/\pi_0)\}$	$BTA - DBTA$
$T_6 = \{t \in T^\omega_\Sigma \mid \forall\ \pi,\ a \in Inf(t/\pi)\}$	$DBTA$

Fig. 2. Some tree languages and their classification.

In the following remark, we summarize the closure properties of the above classes of automata and languages.

Remark 1.

- DMTA, DBTA, and BTA are closed under intersection and union, but they are not closed under complementation.
- MTA is closed under intersection, union, and complementation.

In the following remark, we recall some known results on the decision problems of Büchi and Rabin tree automata.

Remark 2.

- The emptiness problem for BTA is decidable [13], and is LOGSPACE-complete for PTIME [15].
- The non-emptiness problem for RTA is NP-complete [12, 5].

Given a TA $A = \langle Q, \Sigma, \delta, Q_0, F\rangle$, we define the size of A, denoted by $Size(A)$, as the sum of the sizes of Q, δ, and F. Let T be a language, we define the size of T, denoted by $Size_{TA}(T)$, as the minimum over the sizes of the TA's recognizing T, that is, $Size_{TA}(T) = Size(A)$ where A is a TA such that $T(A) = T$ and for every TA A' such that $T(A') = T$, $Size(A) \leq Size(A')$. We refer to such a TA A as a minimal TA for T.

3 Muller-Superset tree automata

In this section, we study the Muller-Superset acceptance condition, a new acceptance condition for tree automata obtained by using the superset operation, instead of the equality, in the Muller acceptance paradigm. We prove that Muller-Superset tree automata define the same class of languages as Büchi tree automata, in both the deterministic and nondeterministic cases. We also show that Muller-Superset tree automata are at least as much succinct as Büchi tree automata and in some cases the gain can be exponential. We start by defining this class of automata.

A Muller-Superset tree automaton (*STA*) S is a TA with acceptance condition $F = \{F_1, \ldots, F_m\}$. A tree $t \in T^{\omega}_{\Sigma}$ is accepted by an STA S if and only if there exists a run $r \in Run_S(t)$ satisfying the condition that for every path π of r, the inclusion $Inf(r/\pi) \supseteq F_i$ holds for some $F_i \in F$. Moreover, with (D)STA we also denote the class of languages accepted by (deterministic) Muller-Superset tree automata. In the next example we give a DSTA accepting the language T_4.

Example 1. Let $S = \langle\{q_0, q_a, q_b\}, \Sigma, \delta, \{q_0\}, \{\{q_a, q_b\}\}\rangle$ be a DSTA where $\delta = \{(q, x, q_x, q_x) \,|\, q \in Q \text{ and } x \in \Sigma\}$. A tree $t \in T(S)$ if and only if there exists a run $r \in Run_S(t)$ such that for all paths π of r, the set $\{q_a, q_b\}$ is contained by $Inf(r/\pi)$, and thus if and only if for all paths π of t, it holds $\{a, b\} \subseteq Inf(t/\pi)$. Hence, we have that $T(S) = T_4$.

In the following theorem, we show that the classes DBTA and DSTA are equivalent. We prove it by showing that for every DBTA B that recognizes a language T, there exists a DSTA S that recognizes the same language T and vice-versa. Besides, using the corresponding reductions, we show that for every language T accepted by a DSTA, $Size_{DSTA}(T) \leq Size_{DBTA}(T) \leq 2^{\mathcal{O}(Size_{DSTA}(T))}$.

Theorem 1. *The classes DBTA and DSTA coincide. Moreover, for a given tree language T, $Size_{DSTA}(T) \leq Size_{DBTA}(T) \leq 2^{\mathcal{O}(Size_{DSTA}(T))}$.*

Proof. To prove that $DBTA \subseteq DSTA$, we use a simple automata construction involving only the accepting set. Given a DBTA $B = \langle Q, \Sigma, \delta, Q_0, F\rangle$, we construct a DSTA $S = \langle Q, \Sigma, \delta, Q_0, F'\rangle$ where $F' = \{\{q_i\} \,|\, q_i \in F\}$. A tree $t \in T(B)$ if and only if there exists an $r \in Run_B(t)$ such that for each path π of r, we get $Inf(r/\pi) \cap F \neq \emptyset$, and thus if and only if there exists a $F_i \in F'$ such that $F_i \subseteq Inf(r/\pi)$. Hence, $t \in T(B)$ if and only if $t \in T(S)$. If B is a minimal $DBTA$ such that $T(B) = T$, by the above construction and the fact that $Size_{DSTA}(T) \leq Size(S)$, we get $Size_{DSTA}(T) \leq Size_{DBTA}(T)$.

Now, let us prove that $DBTA \supseteq DSTA$. Let $S = \langle Q, \Sigma, \delta, Q_0, F\rangle$ be a DSTA and $H = \{W \in \mathcal{P}(Q) \,|\, \forall\, F_i \in F, F_i \not\subseteq W\}$, that is H contains only sets of states W such that for any F_i W is either strictly contained in F_i or not comparable with it. Let $B = \langle\, Q \times H, \Sigma, \delta', Q_0 \times \emptyset, Q \times \emptyset\rangle$ be a DBTA such that $((q, X), a, (q_1, X_1), (q_2, X_2)) \in \delta'$ if $(q, a, q_1, q_2) \in \delta$ and for $i = 1, 2$:

$$X_i = \begin{cases} \emptyset & \text{if } \exists\, F_j \in F \text{ such that } F_j \subseteq X \cup \{q_i\}, \\ X \cup \{q_i\} & \text{otherwise.} \end{cases}$$

The automaton B enters an accepting state if and only if all the states of an accepting set $F_j \in F$ have been visited. Thus, a tree $t \in T(S)$ if and only if $t \in T(B)$. Since the number of sets in H is at most exponential in the size of Q, we have that $Size(B)$ is bounded above by $2^{\mathcal{O}(Size_{DSTA}(T))}$. Thus, $Size_{DBTA}(T) \leq 2^{\mathcal{O}(Size_{DSTA}(T))}$. $\square$

Given a DSTA S accepting a language T, using the above algorithm we can construct a DBTA B accepting T with an exponential blow-up in the size of the automaton S. In the next example, we show a matching lower-bound for this construction. In particular we define a class of languages and a corresponding Muller-Superset tree automata S such that a minimal DBTA B accepting $T(S)$ is such that $Size(B)$ is at least exponential in $Size(S)$.

Example 2. Let $\Sigma = \Sigma_a \cup \Sigma_b$ where $\Sigma_a = \{a_1, \ldots, a_n\}$ and $\Sigma_b = \{b_1, \ldots, b_n\}$, and $T = \{t \in T^{\omega}_{\Sigma} \,|\, \forall\, \pi \text{ of } t, \{a_i, b_i\} \in Inf(t/\pi) \text{ for some } i \text{ such that } 1 \leq i \leq n\}$. We show that T is accepted by a DSTA S with $Size(S) = 4n^2 + 4n$ and a minimal DBTA B accepting T is such that $Size(B) \geq 2^n$. Let $S = \langle Q, \Sigma, \delta, \{q_{a_1}\}, F\rangle$ be a DSTA with $Q = \{q_x \,|\, x \in \Sigma\}$, $F = \bigcup_{i=1}^{n} \{q_{a_i}, q_{b_i}\}$, and $\delta = \{(q, x, q_x, q_x) \,|\, q \in Q$ and $x \in \Sigma\}$. Trivially $T(S) = T$ and $Size(S) = 4n^2 + 4n$.

Now we prove by contradiction that $Size_{DBTA}(T) \geq 2^n$. Assume that B is a DBTA with less than 2^n states such that $T = T(B)$. Notice that any tree t such that for any path π, $t/\pi = (wb_i)^{\omega}$, where $w \in \Sigma_a^*$, $b_i \in \Sigma_b$ and w contains a_i, belongs to T. For a word $w \in \Sigma_a^*$, we define $symbols(w) = \{x_i \,|\, x_i \in \Sigma_a$ and $w = w' x_i w''\}$, that is the set of symbols from Σ_a which occur in w. Define the equivalence $\equiv$ over Σ_a^* as $w \equiv w'$ if and only if $symbols(w) = symbols(w')$. Clearly $\equiv$ has index 2^n. Given a word w, consider a tree t^w such that w is the prefix of the word labeling the leftmost path of t^w. Since B has less than 2^n

states, there exist two words w and w' such that: $w \not\equiv w'$ and, the runs of B on t^w and $t^{w'}$ reach, on the leftmost path, the same state after reading, respectively, w and w'. Let $a_i \in \mathit{symbols}(w)$ and $a_i \notin \mathit{symbols}(w')$. Consider now a tree t whose leftmost path is labeled by $(wb_i)^\omega$ and a tree t' whose leftmost path is labeled by $(w'b_j)^\omega$ for some b_j such that $a_j \in \mathit{symbols}(w')$. Clearly both t and t' belong to T, thus, the corresponding runs r and r' of B are accepting runs. Let t'' be a tree which differs from t' only in the leftmost path which is labeled by $(w'b_i)^\omega$. Such t'' does not belong to T but by the above assumptions we have that the only run of B on t'' is accepting, and thus we have a contradiction.

We end this section showing that Büchi and Muller-Superset tree automata capture the same class of languages also in the nondeterministic case.

Theorem 2. *The classes BTA and STA coincide. Moreover, for a given tree language T, $Size_{STA}(T) \leq Size_{BTA}(T) \leq (Size_{STA}(T))^3$.*

Proof. To prove that $BTA \subseteq STA$, we can use the same construction as we have used in the deterministic case. We recall that that construction also proves $Size_{STA}(T) \leq Size_{BTA}(T)$.

Now let us prove that $BTA \supseteq STA$. Let $S = \langle Q, \Sigma, \delta, Q_0, F\rangle$ be an STA, where $F = \{F_1, \ldots, F_n\}$. We assume that the states from F_h are numbered from 0 to $n_h - 1$, where $n_h = |F_h|$, according to any arbitrary order. Define $B = \langle Q', \Sigma, \delta', Q'_0, F'\rangle$ as the BTA with set of states $Q' = \{[q, h, i] \mid q \in Q, 1 \leq h \leq n$ and $0 \leq i \leq n_h\}$, set of initial states $Q'_0 = \{[q, h, 0] \mid q \in Q_0$ and $1 \leq h \leq n\}$, accepting set $F' = \{[q, h, n_h] \mid q \in Q$ and $1 \leq h \leq n\}$ and transition relation δ' defined as follows. For any transition $(q, a, q_1, q_2) \in \delta$, we add the tuples $([q, h, i], a, [q_1, h_1, i_1], [q_2, h_2, i_2])$ such that at least one between h_1 and h_2 is equal to h, and for $j = 1, 2$ we have that if $h_j \neq h$ then $i_j = 0$, otherwise if $h_j = h$ then

- $i_j = i + 1$, if $i < n_h$ and q is the i-th state of the set F_h,
- $i_j = 0$, if $i = n_h$, and
- $i_j = i$, in all the other cases.

For each path, the automaton B nondeterministically guesses a set of accepting states F_h and checks that all the states from this set are visited infinitely often. Thus, $T(B) = T(S)$. By the above construction we have that the size of Q' is $|F|\,|Q|$, $\delta' = O(n\,|\delta|\,|F|)$ and $|F'| = n\,|Q|$. Thus, we get that $Size(B) \leq (Size(S))^3$, and $Size_{BTA}(T) \leq (Size_{STA}(T))^3$ holds. □

4 Landweber tree automata

In this section, we introduce the *Landweber* acceptance condition, a new acceptance condition on infinite trees obtained by using the subset operation, instead of the equality, in the Muller acceptance condition. We study for the corresponding class of languages the main closure properties and we compare the classes of languages accepted by the deterministic and the nondeterministic Landweber tree automata. The considered model extends to trees, in the usual way, the acceptance condition introduced by Landweber for automata on ω-words [8].

A *Landweber* tree automaton (LTA) L is a TA with accepting condition $F = \{F_1, \ldots, F_m\}$. A tree $t \in T_\Sigma^\omega$ is accepted by L if and only if there exists a run $r \in Run_L(t)$ satisfying the condition that for every path π of r, the containment $Inf(r/\pi) \subseteq F_i$ holds for some $F_i \in F$. Moreover, with (D)LTA we denote the class of languages accepted by (deterministic) Landweber tree automata. Directly from the definition we have that LTA is contained in MTA.

In the following example, we give a DLTA accepting the language T_2.

Example 3. Let $L = \langle \{q_a, q_b\}, \{a, b\}, \delta, \{q_a\}, \{\{q_a\}, \{q_b\}\}\rangle$ be a DLTA where $\delta = \{(q, a, q_a, q_a), (q, b, q_b, q_b) \mid q \in Q\}$. The automaton L changes its state to q_a (respectively, q_b) whenever an occurrence of a (respectively, b) is read. Thus, $t \in T(L)$ if and only if on each path there are either finitely many occurrences of b or finitely many occurrences of a, that is if and only if $t \in T_2$. Hence, $T_2 \in$DLTA.

We show now that, also for this new acceptance condition, nondeterminism strictly increases the accepting power of the model.

Theorem 3. *$DLTA \subset LTA$.*

Proof. To prove this result we provide a language in LTA which is not in DLTA. Namely, we show that $T_1 \in$LTA-DLTA.

Let L be the LTA $\langle \{q_0, q_1, q_2\}, \{a, b\}, \delta, \{q_0\}, \{\{q_1, q_2\}\}\rangle$ such that δ is given by $\{(q_0, x, q_0, q_1), (q_0, x, q_1, q_0), (q_1, x, q_1, q_1), (p, b, q_1, q_2), (p, b, q_2, q_1) \mid x \in \{a, b\}$ and $p \in \{q_0, q_2\}\}$. L nondeterministically chooses a path in the input tree and checks if there is a finite number of a's. This check is carried out by guessing the last occurrence of a on the selected path. Thus, a tree is accepted if and only if it is possible to select a path with a finite number of a's. Hence, $T_1 = T(L)$.

To complete the proof, suppose that there is a DLTA L accepting T_1. Let $t \in T_1$ such that t_0 is entirely labeled by a, and t_1 is entirely labeled by b and let r be a successful run from $Run_L(t)$. Consider now the tree t' obtained from t by exchanging the two subtrees t_0 and t_1, that is $t'_0 = t_1$ and $t'_1 = t_0$. Obviously, $t' \in T_1$, and let r' be a successful run of L on t'. Replace t'_1 for t_1 in t, and call the obtained tree t''. Clearly, $t'' \notin T_1$. Since L is deterministic, and r and r' are successful runs, we have that replacing r'_1 for r_1 in r, we obtain a successful run of L on t''. Thus, $t'' \in T(L)$ and hence we get a contradiction. $\square$

By standard automata constructions we can prove the following theorem:

Theorem 4. *(D)LTA is closed under union and intersection.*

Now we informally introduce a new notation that will be used to prove the non closure under complementation of (D)LTA. Let Σ be a finite alphabet and let $c \notin \Sigma$ be a new symbol, we denote by $T_\Sigma^{(c)}$ the set of trees with both infinite and finite paths valued on $\Sigma \cup \{c\}$ with the restriction that c cannot label internal nodes and all the symbols in Σ can only label internal nodes. Let $t, t' \in T_\Sigma^{(c)}$, we denote by $t \cdot_c t'$ the tree obtained from t by replacing each occurrence of c with t' (*c-concatenation*). For $T \in T_\Sigma^{(c)}$, the ωc-concatenation of t denoted by $t^{\omega c}$ is the tree in T_Σ^ω obtained by iterating the *c-concatenation* on t.

The following theorem states the nonclosure under complementation of both the introduced classes.

Theorem 5. *(D)LTA is not closed under complementation.*

Proof. To prove this result we show that there is a language in DLTA whose complement with respect to T_Σ^ω is not in LTA. In the Example 3, we have given a DLTA accepting T_2. Consider now $T = \{t \in T_\Sigma^\omega \mid \exists\ \pi \textit{ such that } a \in Inf(t/\pi)$ and $b \in Inf(t/\pi)\}$. Clearly T is the complement of T_2 with respect to T_Σ^ω. Suppose that there exists an LTA L such that $T(L) = T$. Thus, there exists $t \in T$ such that $Inf(t/\pi_0) = \{a, b\}$, $Inf(t/\pi) \subset \{a, b\}$ for any path $\pi \neq \pi_0$, and for any natural number h, there exists a natural number $m \geq h$ such that $t(0^i) = b$ for $m \leq i \leq n + m$, where n is the number of states of L. Let r be a successful run of L on t such that $Inf(r/\pi_0) \subseteq F_i$ for a given i. Thus, there exist j and k such that $m \leq j < k \leq m + n$ and $r(0^j) = r(0^k) \in F_i$. We can decompose r (respectively, t) in three parts as follows:

- r' (resp. t') is obtained from r (resp. t) by replacing the subtree r_{0^j} (resp. t_{0^j}) with a sole node labeled by c.
- r'' (resp. t'') is obtained from r_{0^j} (resp. t_{0^j})by replacing r_{0^k} (resp. t_{0^k}) with a sole node labeled by c.
- r''' (resp. t''') is r_{0^k} (resp. t_{0^k}).

Thus, $r = r' \cdot_c r'' \cdot_c r'''$ and $t = t' \cdot_c t'' \cdot_c t'''$. Directly from the above properties of r and t, we get that the run $r' \cdot_c r''^{\omega c}$ is a successful run of L on $t' \cdot_c t''^{\omega c}$. But for the choice of j and k this tree contains only paths with a finite number of a. Hence, we contradict the hypothesis that $T(L) = T$. □

Notice that the above proof is sufficient to show also that T_4 is not LTA.

5 Language comparisons

In this section we compare the classes of languages (D)BTA, (D)MTA, and (D)LTA. To prove our results we use the languages along their rank listed in Figure 2.

The following theorem states the results of the comparisons involving DLTA.

Theorem 6.

1. *DLTA and (D)BTA are not comparable.*
2. *DLTA $\cup$ DBTA $\subset$ DMTA.*

Proof. From example 3 we have that $T_2 \in$DLTA. Since we know that $T_2 \notin$ BTA, and $T_4 \in$DBTA-DLTA, we obtain that DLTA and (D)BTA are not comparable.

From the relationships summarized in Figure 1 and a trivial automata construction we get DLTA $\cup$ DBTA $\subseteq$ DMTA. To prove the strict containment we use $T = \{t \in T^{\omega}_{\Sigma} \,|\, t_0 \in T_2 \text{ and } t_1 \in T_4\}$. Obviously, $T \notin$ DLTA $\cup$ DBTA. We know that T_2 and T_4 is in DMTA and thus part 2 holds. □

Comparisons between LTA and the remaining classes are established in the following lemma.

Theorem 7.

1. *LTA and (D)BTA are not comparable.*
2. *LTA and DMTA are not comparable.*
3. *For $X, Y, Z \in \{LTA,\ BTA,\ DMTA\}$ such that $X \neq Y$, $X \neq Z$ and $Y \neq Z$, we have that $X \not\subseteq Y \cup Z$.*
4. *$LTA \cup BTA \cup DMTA \subset MTA$.*

Proof. Part 1 is a direct consequence of $T_4 \in DBTA - LTA$, and $T_2 \in LTA - BTA$. We know from the table in Figure 2 and the results in section 4 that $T_1 \in$ LTA-DMTA and $T_4 \in$ DMTA-LTA, thus, part 2 holds. To prove part 3 we only need to consider three cases: (a) LTA $\not\subseteq$ BTA $\cup$ DMTA, (b) BTA $\not\subseteq$ LTA $\cup$ DMTA, and (c) DMTA $\not\subseteq$ BTA $\cup$ LTA. Consider the case (a). Let $T = \{t \in T^{\omega}_{\Sigma} \,|\, t_0 \in T_1$ and $t_1 \in T_2\}$. We have that $T \notin$ DMTA $\cup$ BTA since $T_1 \notin$ DMTA and $T_2 \notin$ BTA, but $T \in$ LTA since by the results in section 4 we know that T_1 and T_2 belong to LTA. The proofs of cases (b) and (c) are analogous. In particular, for (b) we may use $T = \{t \in T^{\omega}_{\Sigma} \,|\, t_0 \in T_1 \text{ and } t_1 \in T_4\}$ and for (c) we may use $T = \{t \in T^{\omega}_{\Sigma} \,|\, t_0 \in T_2 \text{ and } t_1 \in T_4\}$. For part 4, we observe that, by a simple automata construction involving only the accepting sets, it is possible to prove that LTA $\subseteq$ MTA. Thus, from Figure 1, we have that DMTA $\cup$ BTA $\cup$ LTA $\subseteq$ MTA. Let $T = \{t \in T^{\omega}_{\Sigma} \,|\, t_0 \in T_1 \text{ and } t_{10} \in T_2 \text{ and } t_{11} \in T_4\}$, we obtain $T \notin$ DMTA $\cup$ BTA $\cup$ LTA since $T_1 \notin$ DMTA, $T_2 \notin$ BTA, and $T_4 \notin$ LTA. It easy to verify that T is instead accepted by an MTA. □

In the following theorem we complete the study of the relationships among the introduced classes of languages by pointing out the existence of languages in particular subclasses.

Theorem 8.

1. *$DLTA \cap DBTA \cap DMTA \neq \emptyset$.*
2. *$((LTA \cap DMTA) - BTA) \neq \emptyset$, $((BTA \cap LTA) - DMTA) \neq \emptyset$ and $((BTA \cap DMTA) - LTA) \neq \emptyset$.*
3. *$((DLTA \cap BTA) - DBTA) \neq \emptyset$.*
4. *$((BTA \cap DMTA) - (DBTA \cup LTA)) \neq \emptyset$.*

Proof. It is easy to prove that $T_3 \in$ DLTA, thus, from the table in Figure 2, T_3 belongs to the intersection of DLTA, DBTA, and DMTA. Hence, part 1 holds. Part 2 directly follows from the classification of the languages T_1, T_2 and T_4. Part 3 can be proved by using T_5. For part 4, consider $T = \{t \in T^{\omega}_{\Sigma} \,|\, t_0 \in T_4$ and $t_1 \in T_5\}$. Since $T_4 \notin$ LTA and $T_5 \notin$ DBTA we have that $T \notin$ LTA $\cup$ DBTA. Since $T_4, T_5 \in$ DMTA $\cap$ BTA, we have that $T \in$ DMTA $\cap$ BTA. □

It is an open problem to determine whether DLTA = DMTA ∩ LTA. Our conjecture is that this is the case. Our main motivation is that a language to be accepted nondeterministically by an LTA and deterministically by an MTA, cannot contain trees that require nondeterministic guesses on the paths on which the automaton must check for a given property. Thus, nondeterminism can be only used along the paths. Since every path is an infinite word and the nondeterministic and deterministic paradigms of finite automata on infinite sequences of characters with Landweber acceptance condition are equivalent, we expect our conjecture to hold.

Assuming that our conjecture is true the relationships among the introduced classes of languages is reported in Figure 3. Notice that according to the results presented in this section, any region denoted in the diagram is not empty.

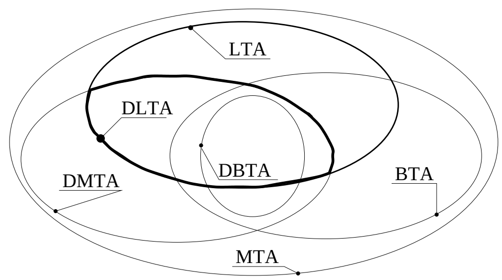

Fig. 3. Relationships between (D)BTA, (D)MTA and (D)LTA.

6 The emptiness problem for *LTA*

In this section we deal with the emptiness problem for Landweber tree automata. From Theorem 7 we have that the class LTA is not comparable with BTA and is strictly contained in MTA. We observe that a Landweber condition can be directly translated to a Rabin condition by a very simple construction. Given a Landweber tree automaton $L = \langle Q, \Sigma, \delta, Q_0, F \rangle$ where $F = \{F_1, \dots, F_m\}$, a Rabin tree automaton accepting $T(L)$ is $R = \langle Q, \Sigma, \delta, Q_0, F' \rangle$, where $F' = \{(F_1, Q - F_1), \dots, (F_m, Q - F_m)\}$. By this construction and the result that any non-empty language accepted by a Rabin tree automaton contains a regular tree [14], we have the following lemma.

Lemma 1. *Any non-empty language accepted by a Landweber tree automaton contains a regular tree.*

Since the constructed automaton R is linear in the size of the starting automaton L, and the non-emptiness problem for Rabin tree automata is NP-complete [5, 12], we get a nondeterministic polynomial time algorithm to solve the non-emptiness problem for Landweber tree automata. In the following, we

Algorithm 1.

$P \leftarrow Q'$
repeat
  $P' \leftarrow P$
  $P \leftarrow \emptyset$
  for $(q, a, q_1, q_2) \in \delta$ such that $q \in P'$ **do**
    if ($p, p' \in \{q_1, q_2\}$, $p \neq p'$, $p \in Q''$ **and** $p' \in P'$) **or** $q_1, q_2 \in P'$ **then**
      $P \leftarrow P \cup \{q\}$
  enddo
until $P' = P$
return $Reach(Q'' \cup P)$

Fig. 4. Algorithm for computing $Accept(Q', Q'')$.

indeed prove that checking for the emptiness of Landweber tree automata can be done in deterministic polynomial time and thus, under the theoretical assumption that "$P \neq NP$", it turns out to be simpler than for Rabin tree automata. Before proving the claimed result on the LTA emptiness problem we give some notation.

Consider a Landweber tree automaton $L = \langle Q, \Sigma, \delta, Q_0, F\rangle$ where $F = \{F_1, \ldots, F_m\}$. A *finite portion* of a run r is a subtree r' of r such that all paths in r' are finite. We refer to the last nodes of all paths in r' as *leaves*. For a set of states $Q' \subseteq Q$, we define $Reach(Q')$ as the set of all the states from which there exists a finite portion of a run whose leaves are labeled with states in Q'. This set can be computed in time linear in $|Q'| + |\delta|$, by using a simple algorithm of backward reachability.

For a transition $e = (q, a, q_1, q_2) \in \delta$, we define the *graph induced* by e as the directed graph whose vertices are q, q_1, and q_2, and whose edges are (q, q_1) and (q, q_2). This definition generalizes to sets of transitions in an obvious way. Moreover, we define a *graph embedded* in L as the graph induced by the transitions from a $\delta' \subseteq \delta$ such that for any $q \in Q$ there exists at most a transition from q in δ'. Given two sets $Q', Q'' \subseteq Q$, by $Accept(Q', Q'')$ we denote a set of states $q \in Q$ such that there exists a graph G embedded in L such that q is a vertex of G and along every maximal simple path of G starting from q either a state in Q'' or a cycle over only states from Q' are reached. Clearly, $Accept(\emptyset, P) = Reach(P)$ and $Accept(P, \emptyset)$ is $Reach(P')$ where $P' = \cup_{i=1}^{k} P_i$ such that for any $i = 1, \ldots, k$, $P_i \subseteq P$ and there exists a graph G_i which is embedded in L, has P_i as set of vertices, and is a strongly connected component (i.e., from any vertex of G all the other vertices of G can be reached). The set $Accept(Q', Q'')$ can be computed by the algorithm in Figure 4.

Lemma 2. *Algorithm 1 computes* $Accept(Q', Q'')$ *in* $O(|Q'|\,|\delta| + |Q''|)$ *time.*

The following theorem holds.

Theorem 9. *The emptiness problem for LTA is decidable in polynomial time.*

Proof. Consider a Landweber tree automaton $L = \langle Q, \Sigma, \delta, Q_0, F\rangle$ where $F = \{F_1, \ldots, F_m\}$. A procedure to decide "$T(L) = \emptyset$?" is given by the recurrence $\Gamma_0 = \cup_{i=1}^{m} Accept(F_i, \emptyset)$ and $\Gamma_{j+1} = \cup_{i=1}^{m} Accept(F_i, \Gamma_j)$. Let Γ be the least fixed-point of the above recurrence, our decision algorithm outputs YES, if $\Gamma \cap Q_0 = \emptyset$, and NO, otherwise. We observe that the least fixed-point of the above recurrence is reached in at most $|Q|$ iterations. Thus, by Lemma 2 and the fact that $Reach(P)$ can be done in linear time, computing Γ requires $O(m\,|\delta|\,|Q|^2)$ time and hence can be computed in polynomial time. To complete the proof we need to show that $T(L)$ is not empty if and only if $\Gamma \cap Q_0$ is not empty. The converse direction is a direct consequence of the definition of the set Γ and the definition of the Landweber acceptance condition. Consider now the forward direction. ¿From Lemma 1 we have that $T(L)$ is not empty if and only if there exists a regular tree belonging to $T(L)$. Consider now a finite graph G corresponding to an accepting regular run of L (i.e., a run on a regular tree). We claim that G must contain at least a strongly connected component over states from a given $F_i \in F$. To prove this we observe that if this is not the case then there is an infinite path π of G, and thus of the corresponding regular tree, that does not satisfy the acceptance condition, i.e., for any $F_i \in F$ it holds that on π the set of states that repeat infinitely often is not contained in F_i. This property ensures that some of the vertices of such G correspond to states belonging to Γ_0. By a similar argument we can prove that at the i-th iteration of the above algorithm at least one new state, among those corresponding to vertices of G, is added to Γ_i, and this is repeated until all such states are added. Since G corresponds to an accepting run of L, then also a state from Q_0 is eventually added to some Γ_i, and thus, is contained in Γ. Hence, we get that if $T(L)$ is not empty then $\Gamma \cap Q_0$ is not empty, and we are done. □

7 Conclusions

In this paper we have studied two new acceptance conditions for tree automata: the Landweber and the Muller-Superset acceptance conditions. These conditions are relaxations of the Muller acceptance condition. We have shown that Muller-Superset tree automata capture the same class of languages as Büchi tree automata, and that automata from this class can be more succinct than Büchi tree automata. For Landweber tree automata we have studied both the nondeterministic and the deterministic paradigms, and compared the corresponding classes of languages to those characterized by Büchi and Muller tree automata. The class of languages accepted by Landweber tree automata is orthogonal to that accepted by Büchi tree automata. For this class we have studied the main closure properties and the emptiness problem. Our main result is that the emptiness problem for Landweber tree automata is decidable in polynomial time, and thus we enlarge the class of ω-tree languages that can be characterized by automata with a tractable emptiness problem. Besides the intrinsic interest in such a theoretical result, the emptiness problem of automata is closely related to decision problems in mathematical logics and also to the specification and verification of systems. Future developments of this research will take in account these aspects. Finally, the Landweber acceptance condition naturally expresses the complement of the Büchi condition. Given a Büchi condition F its complementation is given by the Landweber condition $\{Q - F\}$, where Q is the set of states of the automaton. This suggests a very simple way of characterizing the complement of deterministic Büchi tree automata by nondeterministic Landweber tree automata with only one accepting set. The nondeterminism is needed to nondeterministically select a path to check for the violation of the Büchi condition.

References

1. J.R. Büchi. Weak second-order arithmetic and finite automata. *Z. Math Logik Grundlag. Math.*, 6:66 – 92, 1960.
2. J.R. Büchi. On a decision method in restricted second-order arithmetic. In *Proc. of the International Congress on Logic, Methodology, and Philosophy of Science 1960*, pages 1 – 12. Stanford University Press, 1962.
3. E.M. Clarke and E.A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Proc. of Workshop on Logic of Programs*, LNCS 131, pages 52 – 71. Springer-Verlag, 1981.
4. E.A. Emerson and J.Y. Halpern. Sometimes and not never revisited: On branching versus linear time. *Journal of the ACM*, 33(1):151 – 178, 1986.
5. E.A. Emerson and C.S. Jutla. The complexity of tree automata and logics of programs. In *Proc. of the 29th IEEE-CS Symposium on Foundations of Computer Science*, pages 328 – 337, 1988.
6. R. Hossley. *Finite tree Automata and ω-Automata*. PhD thesis, MIT, Cambridge, Massachussets, 1970.
7. R.P. Kurshan. *Computer-aided Verification of Coordinating Processes: the automata-theoretic approach*. Princeton University Press, 1994.
8. L. H. Landweber. Decision problems for ω-automata. *Mathematical System Theory*, 3:376 – 384, 1969.
9. S. La Torre and M. Napoli. Finite automata on timed ω-trees. *Theoretical Computer Science*, To appear.
10. S. La Torre and M. Napoli. Timed tree automata with an application to temporal logic. *Acta Informatica*, 38(2):89–116, 2001.
11. R. McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, 9:521 – 530, 1966.
12. M.O. Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. Soc.*, 141:1 – 35, 1969.
13. M.O. Rabin. Weakly definable relations and special automata. *Mathematical Logic and Foundations of Set theory*, 1970.
14. M.O. Rabin. Automata on Infinite Objects and Church's Problem. *Amer. Mathematical Soc.*, 1972.
15. M.Y. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. *Journal of Computer and System Sciences*, 32:182 – 211, 1986.
16. M.Y. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115:1 – 37, 1994.