# Equivalence of Alternating and Non-deterministic Büchi Automata

Rafael Peñaloza
rpenalozan@yahoo.com

December 2, 2005

**Abstract**

We show that alternating Büchi automata, a generalization of non-deterministic Büchi automata, define the same class of languages as non-deterministic Büchi automata. This way, non-deterministic automata can be used to solve problems for alternating automata, such as the emptyness problem.

The proof of equivalence will be done with the help of strategy trees, which tell on every node of the input tree which transition should be used in order to obtain an accepting run.

The translation yields an exponential blow-up in the number of states of the automaton. This means that any algorithm in which an alternating automaton is first translated into its equivalent non-deterministic automaton will be exponential in the best case. This cannot be improved as, for example, the emptyness problem of alternating automata is Exp-Hard; thus, the translation is an optimal algorithm.

## 1 Introduction

The concept of alternating automata is a generalization of non-deterministic automata. Alternating automata have been used to solve the satisfiability problem of different kinds of logics.

This aplication requires an emptyness test of the alternating automaton. As the emptyness problem for non-deterministic automata has been very well studied, one would like to use the techniques developed for non-deterministic automata to make the test for alternating automata.

It turns out that alternating and non-deterministic automata are equivalent. This means that the emptyness test may be performed for an equivalent non-deterministic automaton.

As the emptyness problem of non-deterministic automata can be solved in polynomial time, while the same problem for alternating automata is Exp-Hard, a non-deterministic automaton must necessarily be exponentially larger than its equivalent alternating automaton.

Here, a proof of equivalence between alternating Büchi automata and non-deterministic Büchi automata is given. As has been pointed out before, every non-deterministic automaton is also an alternating automaton. Thus, the equivalence proof requires only to construct, for every alternating Büchi automaton, a non-deterministic Büchi automaton that accepts the same language.

To achieve this goal, some restricted cases are dealt with first, aiming at giving a better understanding of the general construction appearing at the end of Section 3.

The concepts needed for constructing and proving correctess of the equivalent non-deterministing automata are given in Section 2. The forementioned constructions appear in Section 3. At the end, in Section 4, some conclusions are given.

## 2 Preliminaries

**Definition 1** *Let $k \in \mathbb{N}$. A* $k$-ary tree *is a set $T \subseteq \{1, \ldots, k\}^*$ such that if $x \cdot n \in T$, for $x \in \mathbb{N}^*, n \in \mathbb{N}$, then $x \in T$ and $x \cdot m \in T$ for all $0 < m < n$. A $k$-ary tree $T$ is* full *if $T = \{1, \ldots, k\}^*$.*

*A* labeled $k$-ary tree over an alphabet $\Sigma$ *is a pair $(T, V)$, where $T$ is a $k$-ary tree and $V : T \to \Sigma$. If $\Sigma = \Gamma \times \Delta$ then $V$ will be expressed as two functions $V = (V_1, V_2)$, and $(T, (V_1, V_2))$ will be denoted by $(T, V_1, V_2)$.*

*An* infinite path *on a $k$-ary tree $T$ is a set $P \subseteq T$ such that $\epsilon \in P$ and for every $x \in P$ there exists a unique $i \in \{1, \ldots, k\}$ such that $x \cdot i \in P$.*

Trees that are not full will be used to define the concept of *run* over alternating automata. Apart from that, all the trees used used as input for the automata will be assumed to be full.

**Definition 2** *Let $k \in \mathbb{N}$. A* non-deterministic Büchi Automaton over $k$-ary trees *is a tuple $M = (Q, \Sigma, \Delta, I, F)$, where $Q$ is a finite set of states, $\Sigma$ is an alphabet, $\Delta \subseteq Q \times \Sigma \times Q^k$ is the transition relation, $I \subseteq Q$ is the set of initial states, and $F \subseteq Q$ is the set of final states.*

*A* run of a non-deterministic Büchi Automaton $M = (Q, \Sigma, \Delta, I, F)$ on a full labeled $k$-ary tree $(T, V)$ over $\Sigma$ *is a full labeled $k$-ary tree $(T, r)$ over $Q$ such that $r(\epsilon) \in I$ and for all $x \in \{1, \ldots, k\}^*$, $(r(x), V(x), r(x \cdot 1), \ldots, r(x \cdot k)) \in \Delta$. Such a run $r$ is* accepting *if and only if for every infinite path $P$ in $(T, r)$, there exist infinitely many $x \in P$ such that $r(x) \in F$.*

*The* language accepted *by a non-deterministic Büchi Automaton $M$ is the set $L(M)$ of all full labeled $k$-ary trees $(T, V)$ such that there is an accepting run of $M$ on $(T, V)$.*

*A* non-deterministic looping Automaton *is a non-deterministic Büchi Automaton in which all the states are final states. Thus, the final states are not explicitely stated in it.*

The transition relation of a non-determistic Büchi automaton can be seen as a function $\tilde{\delta} : Q \times \Sigma \to \mathcal{P}(Q^k)$, where $\tilde{\delta}(q, \sigma) = \{(q_1, \ldots, q_k) \mid (q, \sigma, q_1, \ldots, q_k) \in \Delta\}$. On the other side, a set $P = \{(q_1^1, \ldots, q_k^1), \ldots, (q_1^n, \ldots, q_k^n)\} \in \mathcal{P}(Q^k)$ can be seen as the positive Boolean formula $((1, q_1^1) \wedge \ldots \wedge (k, q_k^1)) \vee \ldots \vee ((1, q_1^n) \wedge \ldots \wedge (k, q_k^n))$. Intuitively, a pair $(i, q) \in \{1, \ldots, k\} \times Q$ means that the automaton should go to the $i$-th descendant with the state $q$.

Hence, if $\mathcal{B}^+(X)$ denotes the set of all positive Boolean formulas over a set $X$, then the transition relation can be seen as a function $\delta : Q\times\Sigma \to \mathcal{B}^+(\{1,\ldots,k\}\times Q)$ where for all $q \in Q, \sigma \in \Sigma$, $\delta(q,\sigma)$ has the form $((1,q_1^1)\wedge\ldots\wedge(k,q_k^1))\vee\ldots\vee((1,q_1^n)\wedge\ldots\wedge(k,q_k^n))$.

Alternating Büchi automata generalize non-deterministic Büchi automata by allowing $\delta$ to be an arbitrary function to the set of positive formulas, and also by allowing a transition that stays in the same node, which will be called 0-transition. This is stated formally in the next definition.

**Definition 3** *Let $k \in \mathbb{N}$. An* Alternating Büchi Automaton over $k$-ary trees *is a tuple $A = (\Sigma, Q, \delta, q_0, F)$, where $\Sigma$ is an alphabet, $Q$ is a finite set of states, $\delta : Q \times \Sigma \to \mathcal{B}^+(\{0,\ldots,k\}\times Q)$ is the transition function, $q_0 \in Q$ is an initial state, and $F \subseteq Q$ is the set of final states.*

*If $\delta : Q\times\Sigma \to \mathcal{B}^+(\{1,\ldots,k\}\times Q)$, then the automaton is said to have* no 0 transitions.

*A* run of an alternating Büchi Automaton $A = (\Sigma, Q, \delta, q_0, F)$ on a full labeled $k$-ary tree $(T,V)$ over $\Sigma$ *is a labeled tree $\langle T_r, r\rangle$ over $T \times Q$ such that:*

1. *$\epsilon \in T_r$ and $r(\epsilon) = (\epsilon, q_0)$.*
2. *if $y \in T_r$ with $r(y) = (x,q)$ and $\delta(q, V(x)) = \theta$, then there is a (possibly empty) set $S = \{(c_1,q_1),\ldots,(c_n,q_n)\} \subseteq \{0,\ldots,k\}\times Q$, such that the following hold:*
   - *$S$ satisfies $\theta$, and*
   - *for all $1 \le i \le n, y\cdot i \in T_r$, and $r(y\cdot i) = (x\cdot c_i, q_i)$, where for all $x \in \{1,\ldots,k\}^*, x\cdot 0 = x$.*

*A run $\langle T_r, r\rangle$ is* accepting *if and only if for every infinite path $P$ in $\langle T_r, r\rangle$, there exist infinitely many $x \in P$ such that $r(x) \in F$.*

*The* language accepted *by an Alternating Büchi Automaton $A$ is the set $L(A)$ of all full labeled $k$-ary trees $(T,V)$ such that there is an accepting run of $A$ on $(T,V)$.*

*An* Alternating looping Automaton *is an Alternating Büchi Automaton in which all the states are final states. Thus, the final states are not explicitely stated in it.*

To prove the equivalence between Alternating Büchi Automata and non-deterministic Büchi Automata, an additional structure called *strategy tree* will be used, along with a result of Vardi [1].

**Definition 4** *Let $A = (\Sigma, Q, \delta, q_0, F)$ be an alternating Büchi automaton on $k$-ary trees. A* strategy tree *$\tau$ for $A$ is a full labeled $k$-ary tree over the alphabet $2^{Q\times\{0,\ldots,k\}\times Q}$. For each $\zeta \in 2^{Q\times\{0,\ldots,k\}\times Q}$, $state(\zeta) := \{u \mid (u,i,v) \in \zeta\}$.*

*A strategy tree $\tau$ is* on *a $k$-ary input tree $(\{1,\ldots,k\}^*, V)$ if and only if the following conditions hold:*

- *$q_0 \in state(\tau(\epsilon))$*
- *for every $x \in \{1,\ldots,k\}^*$ and every $q \in state(\tau(x))$:*
  - *the set $\{(c,q') \mid (q,c,q') \in \tau(x)\}$ satisfies $\delta(q, V(x))$, and*
  - *if $(q,c,q') \in \tau(x)$, then $q' \in state(\tau(x\cdot c))$.*

*A* path $\beta$ *in a strategy tree* $\tau$ *is a sequence* $(u_1, q_1), (u_2, q_2), \ldots$ *of pairs taken from* $\{1, \ldots, k\}^* \times Q$ *such that* $u_1 = \epsilon$*, and for all* $i > 0$ *there is a* $c_i \in \{0, \ldots, k\}$ *such that* $(q_i, c_i, q_{i+1}) \in \tau(u_i)$ *and* $u_{i+1} = u_i \cdot c_i$*. The set of all states in* $Q$ *that occur infinitely often in* $\beta$ *is denoted by* $inf(\beta)$*.* $\tau$ *is* accepting *if and only if every infinite path* $\beta$ *satisfies* $inf(\beta) \cap F \neq \emptyset$*.*

*A strategy tree* $\tau$ *is* irreducible *if and only if for all* $x \in \{1, \ldots, k\}^*$*, all* $c \in \{1, \ldots, k\}$*, and all* $q \in Q$ *the following holds:*

- $q \in state(\tau(\epsilon))$ *if and only if there exist a sequence* $q_0, q_1, \ldots, q_n$ *of elements of* $Q$ *such that*
  - *for all* $0 \leq i < n, (q_i, 0, q_{i+1}) \in \tau(\epsilon)$ *and*
  - $q_n = q$
- $q \in state(\tau(x \cdot c))$ *if and only if there exist a sequence* $q_1, q_2, \ldots, q_n$ *of elements of* $Q$ *such that*
  - $(q_1, c, q_2) \in \tau(x)$*,*
  - *for all* $1 < i < n, (q_i, 0, q_{i+1}) \in \tau(x \cdot c)$ *and*
  - $q_n = q$*.*

Intuitively, a strategy tree shows, for every node of the tree, a set of transitions that can be seen as a strategy to satisfy the transition function. A path is then obtained by following the strategy given by the tree. So, the strategy tree will be accepting if for every selection of transitions chosen, final states are visited infinitely often.

Note that for an alternating looping automaton, every strategy tree is accepting.

**Proposition 5** *An alternating Büchi automaton accepts an input tree if and only if it has an accepting strategy tree on the input tree*

**Lemma 6** *An alternating Büchi automaton has an accepting strategy tree on a k-ary input tree* $(T, V)$ *if and only if it has an irreducible accepting stategy tree on* $(T, V)$*.*

**Proof.** If it has an irreducible accepting strategy tree, then it has an accepting strategy tree. Hence, it is enough to show only the "only if" part of the lemma.

Let $\tau$ be an accepting strategy tree on $(T, V)$. Build a new strategy tree $\tau'$ as follows:

- $\tau'(\epsilon) \coloneqq \tau(\epsilon)$
- for every $x \in \{1, \ldots, k\}^*, c \in \{1, \ldots, k\}$:
  - $Q_{x,c} \coloneqq \{q \in Q \mid \exists q' \in Q$ such that $(q', c, q) \in \tau'(x)\}$
  - $\tau'(x \cdot c) \coloneqq \tau(x \cdot c) \cap (Q_{x,c} \times \{0, \ldots, k\} \times Q)$

Since $\tau$ is an strategy tree on $(T, V)$, $q_0 \in state(\tau(\epsilon)) = state(\tau'(\epsilon))$. Furthermore, for every $x \in \{1, \ldots, k\}^*$ and every $q \in state(\tau'(x))$, the following two conditions hold:

- $\{(c, q') \mid (q, c, q') \in \tau'(x)\} = \{(c, q') \mid (q, c, q') \in \tau(x)\}$, and hence satisfies $\delta(q, V(x))$;

- if $(q,c,q') \in \tau'(x)$, then $(q,c,q') \in \tau(x)$ and $q' \in Q_{x,c}$. Hence, $q' \in state(\tau'(x \cdot c))$.

Thus, $\tau'$ is a strategy tree on $(T,V)$

Notice now that $\tau'$ is irreducible, as for every $q \in state(\tau'(x \cdot c))$, there must exist a $(q',c,q) \in \tau'(x)$; otherwise, $q \notin Q_{x,c}$, and so it is impossible for $q$ to be an element of $state(\tau'(x \cdot c))$.

It remains only to show that $\tau'$ is accepting. Let $\beta$ be an infinite path in $\tau'$. Then $\beta$ is an infinite path in $\tau$ and, as $\tau$ is accepting, $inf(\beta) \cap F \neq \emptyset$. Thus, $\tau'$ is accepting. ∎

# 3 Equivalence of alternating and non-deterministic automata

In this section it will be shown that for every alternating Büchi automaton there is an equivalent non-deterministic Büchi automaton. The next construction is for the restricted case of looping automata.

**Theorem 7** *For every Alternating looping Automaton $A = (\Sigma, Q, \delta, q_0)$ over k-ary trees, there exists a non deterministic looping automaton $M = (S, \Sigma, \Delta, I)$ over k-ary trees such that $L(A) = L(M)$.*

**Proof.** The idea is to construct a non-deterministic looping automaton for which a run is a strategy tree of the alternating looping automaton. Given $A = (\Sigma, Q, \delta, q_0)$, construct $M = (S, \Sigma, \Delta, I)$ as follows:

- $S = 2^{Q \times \{0,\ldots,k\} \times Q}$
- $I = \{s \in S \mid q_0 \in state(s)\}$
- $\Delta = \{(s, \sigma, s_1, \ldots, s_k) \in S \times \Sigma \times S^k \mid$
  1. $\forall q \in state(s), \{(i,q') \mid (q,i,q') \in s\}$ satisfies $\delta(q,\sigma)$
  2. if $(q,i,q') \in s$, then $q' \in state(s_i)$

  $\}$

Let $(T,V)$ be a full labeled $k$-ary tree over $\Sigma$. If $(T,V) \in L(M)$, then there is a run $\tau$ of $M$ on $T$. By definition, this run satisfies:

- $\tau(\epsilon) \in I = \{s \in S \mid q_0 \in state(s)\}$. Hence, $q_0 \in state(\tau(\epsilon))$.
- $\forall x \in \{1,\ldots,k\}^*$, $(\tau(x), V(x), \tau(x \cdot 1), \ldots, \tau(x \cdot k)) \in \Delta$, and thus
  1. $\forall q \in state(\tau(x)), \{(i,q') \mid (q,i,q') \in \tau(x)\}$ satisfies $\delta(q, V(x))$
  2. if $(q,i,q') \in \tau(x)$, then $q' \in state(\tau(x \cdot i))$.

Thus, $\tau$ is a strategy tree on $(T,V)$. Hence, by Proposition 5, $(T,V) \in L(A)$.

Conversely, if $(T,V) \in L(A)$, then, there is a strategy tree $\tau$ on $(T,V)$. Hence, $\tau$ satisfies:

1. $q_0 \in \mathit{state}(\tau(\epsilon))$
2. for every $x \in \{1, \ldots, k\}^*$ and every $q \in \mathit{state}(\tau(x)), \{(c, q') \mid (q, c, q') \in \tau(x)\}$ satisfies $\delta(q, V(x))$ and if $(q, c, q') \in \tau(x)$, then $q' \in \mathit{state}(\tau(x \cdot c))$.

Condition 1 ensures that $\tau(\epsilon) \in I$. On the other hand, by Condition 2, for every $x \in \{1, \ldots, k\}^*, (\tau(x), V(x), \tau(x \cdot 1), \ldots, \tau(x \cdot k)) \in \Delta$. Hence, $\tau$ is a run of $M$ on $T$. Thus, $(T, V) \in L(M)$. ■

To prove the equivalence between alternating Büchi automata and non-deterministic Büchi automata, two automata will be constructed. The first one accepts doubly labeled trees in which the second label corresponds to an irreducible strategy tree over the first. The second automaton works also on doubly labeled trees and simply verifies that a strategy tree, represented in the second label, is accepting.

The composition of both automata yield a new automaton that accepts all doubly labeled trees in which the second label is an accepting strategy tree over the first label. By projecting out the second component, one gets the language accepted by the original alternating automaton.

**Lemma 8** *Let $A = (\Sigma, Q, \delta, q_0, F)$ be an alternating Büchi automaton on k-ary trees. Then there exists a non-deterministic looping automaton on k-ary trees $M_1$ that accepts a tree $(T, V, \tau)$ iff $(T, \tau)$ is an irreducible strategy tree for $A$ on $(T, V)$.*

**Proof.** Let $M_1 = (2^Q, \Sigma \times S, \Delta_1, I)$ where:

- $S = 2^{Q \times \{1, \ldots, k\} \times Q}$,
- $I = \{P \subseteq Q \mid q_0 \in P\}$,
- $\Delta_1 = \{(P, \sigma, s, P_1, \ldots, P_k) \in 2^Q \times \Sigma \times S \times (2^Q)^k \mid$
    1. $P = \mathit{state}(s)$,
    2. $\forall q \in P, \{(c, q') \mid (q, c, q') \in s\}$ satisfies $\delta(q, \sigma)$,
    3. $\forall i \in \{1, \ldots, k\}, \{q' \in Q \mid \exists q \in Q$ such that $(q, i, q') \in s\} = P_i$

  $\}$

Suppose first that $(T, \tau)$ is an irreducible strategy tree on $(T, V)$. Then, it is possible to build an accepting run $r$ of $M_1$ on $(T, V, \tau)$ as follows:

For every $x \in T, r(x) = \mathit{state}(\tau(x))$. Then,

- As $(T, \tau)$ is a strategy tree on $(T, V), q_0 \in \mathit{state}(\tau(\epsilon)) = r(\epsilon)$. Hence, $r(\epsilon) \in Q_0$.
- For every $x \in \{1, \ldots, k\}^*, (r(x), V(x), \tau(x), r(x \cdot 1), \ldots, r(x \cdot k)) \in \Delta_1$:
    - $r(x) = \mathit{state}(\tau(x))$ (by definition of $r$),
    - for every $q \in r(x) = \mathit{state}(\tau(x))$, the set $\{(c, q') \mid (q, c, q') \in \tau(x)\}$ satisfies $\delta(q, V(x))$ (by definition of strategy tree)

- if $(q, c, q') \in \tau(x)$, then $q' \in state(\tau(x \cdot c)) = r(x \cdot c)$ (by definition of strategy tree)
- if there is no $q \in Q$ such that $(q, c, q') \in \tau(x)$, then $q' \notin r(x \cdot c)$ since $q' \notin state(\tau(x \cdot c))$ (by definition of irreducible strategy tree).

Thus, $(T, V, \tau) \in L(M_1)$.

Conversely, if $(T, V, \tau) \in L(M_1)$, then there is an accepting run $r$ such that for every $x \in \{1, \ldots, k\}^*, (r(x), V(x), \tau(x), r(x \cdot 1), \ldots, r(x \cdot k)) \in \Delta_1$. Then, by Condition 1, $r(x) = state(\tau(x))$. From this, the following hold:

- $r(\epsilon) \in I$, thus $q_0 \in r(\epsilon) = state(\tau(\epsilon))$.
- for every $q \in r(x) = state(\tau(x))$:
  - $\{(c, q') \mid (q, c, q') \in \tau(x)\}$ satisfies $\delta(q, V(x))$ (Condition 2),
  - if $(q, c, q') \in \tau(x)$, then $q' \in r(x \cdot c) = state(\tau(x \cdot c))$ (Condition 3), and
  - $\{q' \in Q \mid \exists q \in Q$ such that $(q, c, q') \in \tau(x)\} = state(\tau(x \cdot c))$ (Condition 3).

Hence, $(T, \tau)$ is an irreducible strategy tree on $(T, V)$. ■

The construction of the second automaton will be done in parts. First, a construction for a restricted case will be shown. The restriction consists in requiring that all the paths must go downwards; that is, the alternating Büchi automaton must have no 0 transitions.

**Definition 9** *Let $A = (\Sigma, Q, \delta, q_0, F)$ be an alternating Büchi automaton with no 0 transitions over k-ary trees. The non-deterministic Büchi automaton $M_A$ over k-ary trees is given by $M_A = (2^Q \times 2^Q, \Sigma \times S, \Delta_A, 2^Q \times \{\emptyset\}, \{\emptyset\} \times 2^Q)$, where:*

- $S = 2^{Q \times \{1, \ldots, k\} \times Q}$,
- $\Delta_A = \{((U, V), \sigma, s, (U_1, V_1), \ldots, (U_k, V_k)) \mid$
  - $state(s) \subseteq U \cup V$
  - *if* $U \neq \emptyset$
    - $X_i = \{q' \mid \exists q \in U$ *such that* $(q, i, q') \in s\}$
    - $Y_i = \{q' \mid \exists q \in V$ *such that* $(q, i, q') \in s\}$
    - $U_i = X_i \setminus F$
    - $V_i = Y_i \cup (X_i \cap F)$
  - *if* $U = \emptyset$
    - $Y_i = \{q' \mid \exists q \in V$ *such that* $(q, i, q') \in s\}$
    - $U_i = Y_i \setminus F$
    - $V_i = Y_i \cap F$

  $\}$

Intuitively, this non-deterministic automaton keeps as states pairs of sets of states of the original alternating automaton. The first element stores the states of paths of the strategy tree which have not yet reached a final state, while the second keeps track of the paths that have already reached a final state. When the first element is the empty set, then all the paths to the current point have reached a final state; at that point, the machinery is reset, and starts looking again for final states in the paths.

A tree is accepted if the first component is the empty set infinitely often in every branch of the tree; that is, if every path in the strategy tree reaches a final state infinitely often, which is the condition for it to be accepting.

The next lemma states this intuition formally.

**Lemma 10** *Let $A = (\Sigma, Q, \delta, q_0, F)$ be an alternating Büchi automaton with no 0 transitions over k-ary trees, then the non-deterministic Büchi automaton $M_A$ accepts a tree $(T, V, \tau)$, with $(T, \tau)$ an irreducible strategy tree for A iff $(T, \tau)$ is accepting.*

**Proof.** Suppose first that $(T, V, \tau) \in L(M_A)$, and that $(T, \tau)$ is an irreducible strategy tree for $A$. Let $\beta = (u_1, q_1), (u_2, q_2), \ldots$ be an infinite path in $(T, \tau)$, $(u_i, q_i) \in \{1, \ldots, k\}^* \times Q$. It is sufficient to show that $inf(\beta) \cap F \neq \emptyset$.

Suppose that $inf(\beta) \cap F = \emptyset$, then there exists an $n \in \mathbb{N}$ such that $q_m \notin F$, for all $m \geq n$. As $(T, V, \tau) \in L(M_A)$, there is an accepting run $r$ of $M_A$ on $(T, V, \tau)$. Consider the following path in $(T, r) : (u_1, r(u_1)), (u_2, r(u_2)), \ldots$, and let $r(u_i) = (U_i, V_i)$. As $r$ is accepting, this path has to reach final states infinitely often. In particular, there exist $m_0 \geq n$ such that $r(u_{m_0}) = (\emptyset, V_{m_0})$.

On the other hand, as $\beta$ is an infinite path, for every $i > 0, q_i \in state(\tau(u_i)) \subseteq U_i \cup V_i$. Hence, $q_{m_0} \in V_{m_0}$. As $q_{m_0+1} \notin F$, then $q_{m_0+1} \in U_{m_0+1}$, thus $U_{m_0+1} \neq \emptyset$.

Now, for every $m > m_0 \geq n, q_{m+1} \in U_{m+1}$ because $q_m \in U_m$ and $q_{m+1} \notin F$. Then, for all $m > m_0, U_m \neq \emptyset$. This contradicts the fact that $r$ is an accepting run of $M_A$. Thus, $inf(\beta) \cap F \neq \emptyset$. This means then that $(T, \tau)$ is an accepting strategy tree.

Conversely, suppose that $(T, \tau)$ is an accepting strategy tree. Notice that the transition relation in $M_A$ is functional; that is, given the state in the root, there is at most one run of the automaton over a given tree.

Construct a run $r$ of $M_A$ by setting $r(\epsilon) = (state(\tau(\epsilon)), \emptyset)$. This run exists since it was assumed that $(T, \tau)$ is irreducible, and hence for all $x \in T$, if $r(x) = (U, V)$ then $state(\tau(x)) \subseteq U \cup V$.

It remains to prove that $r$ is accepting. Suppose on the contrary that it is not accepting. Then, there exist a path in $r, (u_1, (U_1, V_1)), (u_2, (U_2, V_2)), \ldots$, and a $n \in \mathbb{N}$ such that for all $m \geq n, U_m \neq \emptyset$. By the definition of the transition relation, for every $m \geq n$, if $u_{m+1} = u_m \cdot c$ and $q \in U_{m+1}$, then there exists a $q' \in U_m$ such that $(q', c, q) \in \tau(u_m)$. Furthermore, for all $m \geq n, U_m \cap F = \emptyset$. Hence, it is possible to construct an infinite path $\beta = (u_1, q_1), (u_2, q_2), \ldots$ such that for all $m \geq n, q_m \in U_m$, but then, $q_m \notin F$. This contradicts the fact that $(T, \tau)$ is accepting.

Hence, $r$ is an accepting run, and thus, $(T, V, \tau) \in L(M_A)$. ■

The requirement of all the paths going downwards is necessary in the first part of the proof, when an infinite path on $\tau$ is used to construct a path on the run $\langle T, r\rangle$. If 0-transitions were allowed, this construction would not be possible.

**Theorem 11** *Let $A = (\Sigma, Q, \delta, q_0, F)$ be an alternating Büchi automaton with no 0-transitions. Then, there is a non-deterministic Büchi automaton $M$ such that $L(A) = L(M)$.*

**Proof.** Given $A$, construct the automata $M_1$ and $M_A$. The usual product construction yields an automaton $M'$ that accepts $L(M_1) \cap L(M_A)$. That is, $M'$ accepts all trees with an accepting strategy tree for $A$. By Proposition 5, $M'$ accepts all the trees in $L(A)$ with their corresponding accepting strategy trees for $A$.

Projecting out the second component, that is, the strategy tree, one gets the desired automaton $M$. ■

This result can be generalized to alternating automata with 0-transitions. The main idea is still the same: keep track of all the paths which have not reached a final state. Nonetheless, the construction of the automaton that verifies the acceptance of the strategy tree is more complex, as it has to deal with transitions that stay in the same node of the tree, which are not possible in alternating automata with no 0 transitions. A simple generalization of the construction given in Definition 9 is enough to deal with paths that make only a finite number of steps on every node.

Additional difficulties arise when a path loops in the same node forever: the accepting condition must be verified for these paths, too. The next definition will be used to construct the desired automaton, to check this kind of paths.

**Definition 12** *Let $Q$ be a set, and $S = 2^{Q\times\{0,\dots,k\}\times Q}$. For every $s \in S$, define the set $Q_s := \{q \in Q \mid$ there exists an infinite sequence $q_0, q_1, \dots$ of elements of $Q$ such that $q_0 = q$, for all $i \geq 0, (0, q_{i+1}) \in \{(c, q) \mid (q_i, c, q) \in s\}$, and there exists a $n \in \mathbb{N}$ such that for all $m \geq n, q_m \notin F\}$*

Recall that a strategy tree is a labeled tree over the alphabet $S$. Briefly, the elements of $Q_s$ are those states for which, if a node of the strategy tree was labeled with $s$, the strategy would yield an infinite path that loops in that node, but doesn't reach final states infinitely often. If any of such states by some path leading to the corresponding node, then the tree may not be accepting, and thus, must be rejected.

With the help of these sets, it is now possible to define an automaton that verifies that the strategy tree is accepting. The correctness of the construction is shown afterwards.

**Definition 13** *Let $A = (\Sigma, Q, \delta, q_0, F)$ be an alternating Bc̈hi automaton over k-ary trees. The non-deterministic Büchi automaton $M_A^0$ over k-ary trees is given by $M_A^0 = ((2^Q \times 2^Q), \Sigma \times S, \Delta_A^0, 2^Q \times \{\emptyset\}, \{\emptyset\} \times 2^Q)$, where:*

- $S = 2^{Q\times\{0,\dots,k\}\times Q}$,
- $\Delta_A^0 = \{((U, V), \sigma, s, (\bigcup_{l\geq 1} U_1^l, \bigcup_{l\geq 1} V_1^l), \dots, (\bigcup_{l\geq 1} U_k^l, \bigcup_{l\geq 1} V_k^l)) \mid$
  - $state(s) \subseteq U \cup V$
  - $U_0^0 = U, V_0^0 = V$

– *for* $l > 0, (U_0^{l+1}, V_0^{l+1}), (U_1^{l+1}, V_1^{l+1}), \ldots, (U_k^{l+1}, V_k^{l+1})$ *are given by*
  *if* $U_0^l \neq \emptyset$*, then*
  - $X_i = \{q' \mid \exists q \in U_0^l$ *such that* $(q, i, q') \in s\}$
  - $Y_i = \{q' \mid \exists q \in V_0^l$ *such that* $(q, i, q') \in s\}$
  - $U_i^{l+1} = X_i \setminus F$
  - $V_i^{l+1} = Y_i \cup (X_i \cap F)$

  *if* $U_0^l = \emptyset$*, then*
  - $Y_i = \{q' \mid \exists q \in V_0^l$ *such that* $(q, i, q') \in s\}$
  - $U_i^{l+1} = \emptyset$
  - $V_i^{l+1} = Y_i$

– $(U \cup V) \cap Q_s = \emptyset\}$

The idea behind this construction is the same as in Definition 9. The states of the non-deterministic automaton are pairs of sets of states of the alternating automaton that keep information of the paths that have reached, or not, final states. The transition relation checks the paths that stay finitely in every node. The condition that $(U \cup V) \cap Q_s = \emptyset\}$ ensures that all the paths that loop forever in the current state will reach final states infinitely often.

Notice that, although the construction aparently requires the computation of an infinite union for the transition relation, this union contains always only a finite amount of different sets, for they are all subsets of $Q$, which is finite. Hence, the forementioned union can be computed.

**Lemma 14** *Let* $A = (\Sigma, Q, \delta, q_0, F)$ *be an alternating Büchi automaton over k-ary trees, then the non-deterministic Büchi automaton* $M_A^0$ *accepts a tree* $(T, V, \tau)$ *with* $(T, \tau)$ *an irreducible strategy tree for A iff* $(T, \tau)$ *is accepting.*

**Proof.** Suppose first that $(T, V, \tau) \in L(M_A^0)$, and that $(T, \tau)$ is an irreducible strategy tree for $A$. Then, there exists a run $r$ of $M_A^0$ on $(T, V, \tau)$. Let $\beta = (u_1, q_1), (u_2, q_2), \ldots$ be a path on $\tau$.

Suppose that $inf(\beta) \cap F = \emptyset$. Then, there exists a $n_0 \in \mathbb{N}$ such that $q_m \notin F$ for all $m \geq n_0$. Then, there are two possible cases:

- if for all $n$ exists an $m > n$ such that $u_m \neq u_n$, consider the following path in $(T, r) : (u_{m_1}, r(u_{m_1})), (u_{m_2}, r(u_{m_2})), \ldots$, where $m_1 = 1, m_{i+1} = min\{n \geq m_i \mid u_n \neq u_{m_i}\}$, and let $r(u_{m_i}) = (U_i, V_i)$.

  As $r$ is an accepting run, there exists $l_0$ such that $m_{l_0} \geq n_0$ and $r(u_{m_{l_0}}) = (\emptyset, V_{l_0})$. As $\beta$ is an infinite path, for all $i > 0, q_{m_i} \in state(\tau(u_{m_i})) \subseteq U_i \cup V_i$. Hence, $q_{m_{l_0}} \in V_{l_0}$. As $q_m \notin F$ for all $m \geq m_{l_0} \geq n, q_{m_{l_0+1}} \in U_{l_0+1}$. Thus $U_{l_0+1} \neq \emptyset$.

  But then, for all $l > l_0, U_l \neq \emptyset$, contradicting the fact that $r$ is an accepting run of $M_A^0$.

- if there exists an $n$ such that $u_m = u_n$ for all $m \geq n$, then as $q_m \notin F$ for all $m \geq n, q_n \in Q_s$. But if this was the case, then $M_A^0$ would not be able to make a transition at node $u_n$, since if $r(u_n) = (U, V)$, then $q_n \in U \cup V$. This contradicts the fact that $r$ is a run.

Hence, for every path $\beta$ in $\tau$, $inf(\beta) \cap F \neq \emptyset$, which means that $(T, \tau)$ is accepting.

Conversely, suppose that $(T, \tau)$ is accepting. Notice that the transition relation of $M_A^0$ is functional. Hence, given the state at the root, there is at most one run of the automaton over a given input tree.

Construct a run $r$ of $M_A^0$ by setting $r(\epsilon) = state(\tau(\epsilon)), \emptyset)$. This run exists since, as $(T, \tau)$ is irreducible, for all $x \in T$, if $r(x) = (U, V)$, then $state(\tau(x)) \subseteq U \cup V$. Furthermore, $(U \cup V) \cap Q_{\tau(x)} = \emptyset$; otherwise, it would be possible to build an infinite path that doesn't reach final states infinitely often, contradicting the fact that $(T, \tau)$ is an accepting strategy tree.

It remains to prove that $r$ is accepting. Suppose on the contrary that it is not. Then, there exist a path in $r, (u_1, (U_1, V_1)), (u_2, (U_2, V_2)), \ldots$ and a $n \in \mathbb{N}$ such that for all $m \geq n, U_m \neq \emptyset$. By definition of the transition relation, for every $m \geq n$, if $u_{m+1} = u_m \cdot c$ and $q \in U_{m+1}$, then there exist a sequence $q_0, q_1, \ldots, q_l$, with $l \geq 0$ such that $(q_i, 0, q_{i+1}) \in \tau(u_m)$ for all $i < l$, and $(q_l, c, q) \in \tau(u_m)$. Furthermore, $U_m \cap F = \emptyset$. Hence, it is possible to construct an infinite path $\beta$ such that $inf(\beta) \cap F = \emptyset$. This contradicts the fact that $(T, \tau)$ is accepting.

Hence, $r$ is an accepting run, and thus $(T, V, \tau) \in L(M_A^0)$. ∎

**Theorem 15** *Let $A = (\Sigma, Q, \delta, q_0, F)$ be an alternating Büchi automaton. Then, there is a non-deterministic Büchi automaton $M$ such that $L(A) = L(M)$.*

**Proof.** Given $A$, construct the automata $M_1$ and $M_A^0$. The usual product construction yields an automaton $M'$ that accepts $L(M_1) \cap L(M_A^0)$. That is, $M'$ accepts all trees with an accepting strategy tree for A. By Proposition 5, $M'$ accepts all the trees in $L(A)$ with their corresponding accepting strategy trees for A.

Projecting out the second component yields the desired automaton $M$. ∎

# 4 Conclusions

Alternating Büchi automata and non-deterministic Büchi automata define the same class of languages. With this result, the emptyness problem of an alternating Büchi automaton can now be solved by making an emptyness test on the equivalent non-deterministic Büchi automaton, which can be done in polynomial time on the size of the automaton.

However, the change from alternating to non-deterministic automata gives an exponential blow-up. This means that this method for checking emptyness gives an exponential algorithm, in the size of the alternating Büchi automaton. This is, in fact, the best that can be done, as the emptyness problem of alternating Büchi automata is Exp-Hard.

It is worth noting that, in [1], the equivalence of automata with parity acceptance condition is proven using a simpler construction. The way it is done is by using a word automaton to find a path that will *not* be accepting. Afterwards, that automaton is co-determinized and used to construct the desired tree automaton.

This method cannot be used with Büchi automata, because not every non-deterministic Büchi automaton can be determinized.

## References

[1] Vardi, M.Y.: Reasoning about The Past with Two-Way Automata.

[2] Muller, M.E. and P.E. Schupp: Alternating Automata on Infinite Trees. *Theoretical Computer Science*, 54, 267-276, 1987.