

Converting a Büchi alternating automaton to a usual nondeterministic one

AMAR ISLI

LIPN - CNRS URA 1507, Inst. Galilée, Université Paris-Nord Av. J.B. Clément,
93430 Villetaneuse, France

E-mail: Amar.Isli@lipn.univ-paris13.fr

Abstract. We first give a method for simulating, in the case of Büchi, an alternating automaton by a usual nondeterministic one. Then, to make the satisfiability problem of Linear Propositional Temporal Logic (LPTL) use this result, we give a method for translating any formula of this logic into an equivalent Büchi alternating automaton.

Keywords. Büchi alternating automaton; nondeterministic automaton; linear propositional temporal logic.

1. Introduction

Automata on infinite sequences were introduced in the early sixties, first by Büchi (1960) and then by Muller (1963). Initially, the two types of automata on infinite words, Büchi automata and Muller automata, seemed not to have the same expressiveness. McNaughton (1966) showed that Büchi nondeterministic automata have the same expressive power than Muller automata.

The theory of automata on infinite sequences is used in many areas of computer science. The satisfiability problem of many logics, which consists of testing whether a formula is satisfiable, can be translated into the emptiness problem of an automaton on infinite words; that is, the problem of testing whether such an automaton accepts a nonempty language.

In the early eighties, alternating automata have been introduced as an extension to usual nondeterministic automata (Chandra *et al* 1981; Miyano & Ayashi 1984; Muller & Schupp 1985, 1987). With the latter automata, states are only existential. The new with the former automata is that states can also be universal. The advantage of using alternating automata, especially as they are defined in (Muller & Schupp 1985, 1987), is that they offer a natural and straightforward way of translating a temporal formula into such an automaton Isli (1994). Another advantage is that complementation is easy, and requires linear time; it is performed by dualizing the transition function and complementing the accepting condition (complementation theorem (Muller & Schupp 1985, 1987)).

In this paper, we first give a method for simulating a Büchi alternating automaton by a usual nondeterministic one. Then we describe a method associating to any formula of

Linear Propositional Temporal Logic (LPTL (Pnueli 1981)) a Büchi alternating automaton accepting its models. This clearly leads to an alternating automata approach for the satisfiability problem of LPTL, which first translates the input formula into a Büchi alternating automaton and then simulates the alternating automaton by a usual nondeterministic one.

2. Büchi automata on ω -words: simulating an alternating by a usual nondeterministic

Alternating automata (Chandra *et al* 1981; Miyano & Ayashi 1984; Muller & Schupp 1985; Muller & Schupp 1987) have been introduced as an extension to usual nondeterministic automata. With the usual nondeterministic automata, states are only existential, while with the alternating automata the states can also be universal. As defined in (Chandra *et al* 1981; Miyano & Ayashi 1984), a state of an alternating automaton is either existential or universal, and cannot be intermediate: if Σ and Q are the alphabet and the set of states, respectively, then the transition function maps a pair (a, q) of $\Sigma \times Q$ into a subset of Q which has to be interpreted either universally or existentially. The definition of alternation in (Muller & Schupp 1985, 1987) is more natural, and allows states to be, say, existential-then-universal: the transition function maps (a, q) into a set of subsets of Q ,¹ the set being existential and each subset universal (first choose, nondeterministically, a subset, and then all the states of the chosen subset). Another advantage of defining alternating automata as in (Muller & Schupp 1985, 1987) is that translating a temporal formula into such an automaton is a straightforward and extremely easy task (the translation is given in § 3).

A run of an alternating automaton on an infinite word is not, as it is the case for usual nondeterministic automata, necessarily an infinite sequence of states, but, in general, an infinite tree. The accepting condition concerns not only a unique infinite sequence but each branch of the tree form of the run.

In this section, we propose, in the case of Büchi, an effective construction mapping an alternating automaton into a usual nondeterministic one simulating it. This transforms the emptiness problem of the former automaton to the well known emptiness problem of the latter.

2.1 Alternating automata

For reasons mentioned above, our definition of alternating automata follows (Muller & Schupp 1985, 1987).

DEFINITION 1

Let E be a countable set. The free distributive lattice $\mathcal{L}(E)$ generated by E is defined as the least set satisfying:

- (1) each element of E belongs to $\mathcal{L}(E)$, and
- (2) if f and g belong to $\mathcal{L}(E)$, then so do both $f \wedge g$ and $f \vee g$.

¹In fact, the transition function maps (a, q) into an element of the free distributive lattice $\mathcal{L}(Q)$ generated by Q ; the disjunctive normal form of this element of $\mathcal{L}(Q)$, written in a set form, is a set of subsets of Q .

DEFINITION 2

A Büchi alternating automaton on infinite words is a 5-tuple $M = \langle \mathcal{L}(Q), \Sigma, \delta, q_0, F \rangle$ defined in the following manner:

- (1) Q is a finite state set,
- (2) $\mathcal{L}(Q)$ is the free distributive lattice generated by Q ,
- (3) Σ is the input alphabet,
- (4) $\delta : \Sigma \times Q \longrightarrow \mathcal{L}(Q)$ is a transition function,
- (5) $q_0 \in Q$ is the initial state of M , and
- (6) F is a set defining the accepting condition.

The set F defining the accepting condition is defined in the same fashion as in the usual nondeterministic case. That is, F is a subset of Q , called set of distinguished states.

DEFINITION 3

A run of a Büchi alternating automaton $M = \langle \mathcal{L}(Q), \Sigma, \delta, q_0, F \rangle$ on an infinite word $u = a_0a_1 \dots \in \Sigma^\omega$ is an infinite labeled tree, with no leaf, verifying the following:

- (1) the labels (of the nodes) belong to the state set of M ,
- (2) the label of the root is q_0 , the initial state of M , and
- (3) if
 - v is a node of level n labeled by q ,
 - a_n is the n^{th} letter of the word u ,
 - $\{q_1, \dots, q_m\}$ is the set of labels of the immediate successors of v , and
 - $t_1 \vee t_2 \vee \dots \vee t_r$ is the disjunctive normal form of $\delta(a_n, q)$,
 then there exists $j \in \{1, \dots, r\}$ verifying $\{q_1, \dots, q_m\} = \{p \in Q : p \text{ occurs in } t_j\}$; that is, the labels of the immediate successors of v are exactly the states occurring in a certain disjunct, t_j , of the disjunctive normal form of $\delta(a_n, q)$.

Figure 1 illustrates part (3) of definition 3.

Uniform run

A run of a Büchi alternating automaton on an infinite word is said to be uniform if for all nodes v_1 and v_2 at the same level and having the same label, and for all state q of the automaton, the following condition holds: v_1 has an immediate successor labelled by q if and only if v_2 has an immediate successor labelled by q .

History

A history of a Büchi alternating automaton is an infinite sequence $q_{i_0}q_{i_1} \dots q_{i_n} \dots$ of states of the automaton.

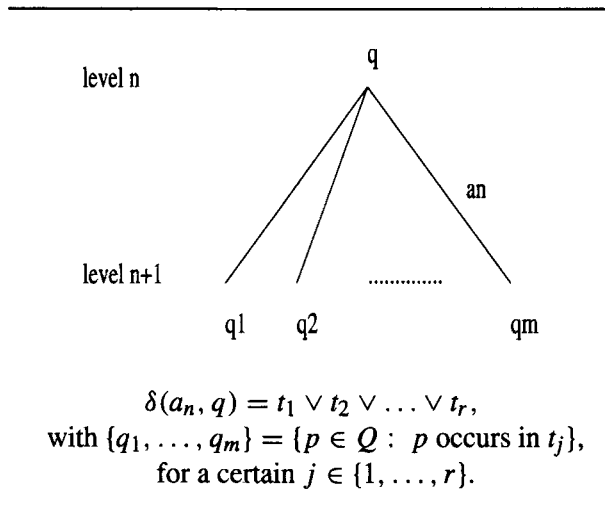


Figure 1. The immediate successors of a node in a run.

History lying on a branch

Let M be a Büchi alternating automaton, and t a run of M on an infinite word u . A branch of t is an infinite path starting at its root. The history lying on a branch β of the run t is the history $q_{i_0}q_{i_1} \dots q_{i_j} \dots$ of the automaton M such that its j th element q_{i_j} , $j \geq 0$, is the label of the j th node of the branch β .

Accepting history

To a history h of a Büchi alternating automaton $M = \langle \mathcal{L}(Q), \Sigma, \delta, q_0, F \rangle$ are associated:

- the mapping $c_h : \mathbb{N} \longrightarrow Q$ such that $c_h(n)$ is the n th letter of h , and
- the set $Inf(h) = \{q \in Q : |c_h^{-1}(q)| = \omega\}$, where $c_h^{-1}(q) = \{n \in \mathbb{N} : c_h(n) = q\}$.

Then a history is accepting if $Inf(h) \cap F \neq \emptyset$.

Accepting run

A branch of a run is accepting if the history lying on it is accepting. A run is accepting if all its branches are accepting.

An infinite word u is accepted by a Büchi alternating automaton M if there exists an accepting run of M on u . The language accepted by M , denoted by $L(M)$, is the set of infinite words accepted by M .

Remark 1. Let t be a uniform run of a Büchi alternating automaton M on an infinite word. If v_1 and v_2 are two nodes at the same level and with the same label, then a history lies on a branch of the subtree t/v_1 of t at v_1 if and only if it lies on a branch of the subtree t/v_2 of t at v_2 .

Theorem 1 below shows that in an alternating automaton we can, without loss of generality, restrict our attention to uniform runs only.

Theorem 1. *Let M be an alternating automaton. For each word u accepted by M , there exists an accepting uniform run of M on u .*

Proof. see appendix A. □

2.2 Run DAG of an alternating automaton

We now need, in order to get our simulation method, to transform a uniform run of a Büchi alternating automaton from its tree form into a directed acyclic graph (DAG) form. This new form of a run will be called **run DAG**, and is defined in the following manner. A run DAG of a Büchi alternating automaton M on an infinite word is the quotient of a uniform run t of M on that word modulo the following equivalence relation R_t defined on the set of nodes of t :

$$v_1 R_t v_2$$

if and only if

$$v_1 \text{ and } v_2 \text{ are at the same level, and labelled by a same state.}$$

2.2a Distinguished levels of a run DAG: Let G be a run DAG of a Büchi alternating automaton $M = \langle \mathcal{L}(Q), \Sigma, \delta, q_0, F \rangle$ on an infinite word u . G is said to be accepting if the uniform run of which it is a quotient is itself accepting. It is easily seen that G is accepting if and only if on all infinite path leaving the root we meet infinitely often an element of F . Let us now have a traversal of G .

Let n_0 the least level of G such that:

every path joining the root to a node of level n_0 has at least one label from F .

Let n_{i+1} be the least level such that:

1. $n_i < n_{i+1}$, and
2. every path starting at a node of level $(n_i + 1)$, and terminating at a node at level n_{i+1} , has at least one label from F .

In the sequel, such levels n_i ($i \geq 0$) are called **distinguished levels** of G .

Theorem 2. *A run DAG of a Büchi alternating automaton is accepting if and only if the number of its distinguished levels is infinite.*

Proof. Straightforward. Left to the reader. □

2.2b Characterizing the distinguished levels: The problem to deal with now is how to characterize the distinguished levels of a run DAG. The solution we propose to solve this problem consists of keeping track of the states met while traversing a run DAG, and

to give a special attention to the distinguished states. For this purpose, we make use of the booleans “0” and “1”. Intuitively, these booleans will serve, respectively, to say “we are still soliciting a distinguished state” and “we have met a distinguished state”. But another problem appears, which consists of what to do when on every path we have met a distinguished state. As it will be seen, the solution to this new problem consists of reinitializing the process of soliciting a distinguished state along every path.

Let us consider a run DAG G of a Büchi alternating automaton $M = \langle \mathcal{L}(Q), \Sigma, \delta, q_0, F \rangle$ on an infinite word u . We define a mapping u_G associating with each node of G an element from the cross product $Q \times \{0, 1\}$. A node v labelled by q will be such that $u_G(v) = (q, 0)$ or $u_G(v) = (q, 1)$. The definition of the mapping u_G is as follows. Let r be the root of G :

$$u_G(r) = \begin{cases} (q_0, 1) & \text{if } q_0 \in F, \\ (q_0, 0) & \text{otherwise.} \end{cases}$$

If the mapping u_G is known at each node at level $n \geq 0$, then for a node v at level $(n + 1)$ labelled by q we have the following:

- if every node v' of level n satisfies $u_G(v') \in Q \times \{1\}$, then:

$$u_G(v) = \begin{cases} (q, 1) & \text{if } q \in F, \\ (q, 0) & \text{otherwise.} \end{cases}$$

- otherwise:

$$u_G(v) = \begin{cases} (q, 1) & \text{if } q \in F, \text{ or all immediate predecessor } v' \\ & \text{of } v \text{ verifies } u_G(v') \in Q \times \{1\}, \\ (q, 0) & \text{otherwise.} \end{cases}$$

Clearly, a level n of G is a distinguished level if and only if for all node v of level n we have the following:

$$u_G(v) \in Q \times \{1\}.$$

2.3 The simulating method

A simulation, in the case of Büchi, of an alternating automaton by a usual nondeterministic one follows straightforwardly from the characterization, given in § 2.2b, of the distinguished levels of a run DAG. Let us consider a Büchi alternating automaton $M = \langle \mathcal{L}(Q), \Sigma, \delta, q_0, F \rangle$. We denote by $E_G(n)$, for a run DAG G and a level n , the set $\{u_G(v) : v \text{ node of level } n\}$. We also consider the set \mathcal{G} of all possible run DAGs of M , and define the equivalence relation R_M on the cross product $\mathbb{N} \times \mathcal{G}$ in the following manner:

$$(n_1, G_1)R_M(n_2, G_2) \text{ if and only if } E_{G_1}(n_1) = E_{G_2}(n_2).$$

Intuitively, the equivalence class R_M puts together the levels having, say, the same history. The equivalence class, $R_M((n, G))$, of an element (n, G) of the cross product $\mathbb{N} \times \mathcal{G}$, is represented by $E_G(n)$. This leads to the following theorem which gives an effective construction of a usual Büchi nondeterministic automaton simulating an alternating one.

In the theorem, the notation $transition(a, Q_1)$ stands for the set of subsets of $Q \times \{0, 1\}$ defined as follows:

$$\begin{aligned}
 S &\in transition(a, Q_1) \\
 &\Updownarrow \\
 \exists u \text{ (infinite word), } n \text{ (nonnegative integer) such that} \\
 Q_1 &= u_G(n), S = u_G(n+1), \text{ and} \\
 u &= b_0 b_1 \dots b_n \dots, \text{ with } b_n = a.
 \end{aligned}$$

Theorem 3. Let $M = \langle \mathcal{L}(Q), \Sigma, \delta, q_0, F \rangle$ be a Büchi alternating automaton. The usual Büchi nondeterministic automaton $B = \langle \mathcal{Q}, \Sigma, \delta', Q_0, F' \rangle$ defined as follows:

- (1) $\mathcal{Q} = 2^{Q \times \{0,1\}}$, the set of subsets of $Q \times \{0, 1\}$,
- (2) $\delta'(a, Q_1) = transition(a, Q_1)$, for all $(a, Q_1) \in \Sigma \times \mathcal{Q}$,
- (3) $Q_0 = \begin{cases} \{(q_0, 1)\} & \text{if } q_0 \in F, \\ \{(q_0, 0)\} & \text{otherwise, and} \end{cases}$
- (4) $F' = 2^{Q \times \{1\}}$,

simulates M .

Proof. Straightforward from the characterization of the distinguished levels of a run DAG, described in § 2.2b. \square

2.4 Size of the simulating automaton

Let $M = \langle \mathcal{L}(Q), \Sigma, \delta, q_0, F \rangle$ be a Büchi alternating automaton, and n the cardinality of Q . From the construction of the nondeterministic automaton $B = \langle \mathcal{Q}, \Sigma, \delta', Q_0, F' \rangle$ simulating M , in particular the construction of its transition function, it follows that for any state q of M , and for any reachable state Q_1 of B , $(q, 0)$ and $(q, 1)$ cannot be simultaneously in Q_1 . It follows that the cardinality of the state set \mathcal{Q} of B is bounded by the number of mappings f from Q into $\{0, 1, 2\}$ defined as follows:

$$f(q) = \begin{cases} 0 & \text{if } (q, 0) \in Q_1, \\ 1 & \text{if } (q, 1) \in Q_1, \\ 2 & \text{otherwise.} \end{cases}$$

The number of such mappings is clearly 3^n .

The following two points imply that this upper bound is better than 3^n :

$$\text{for all } q \in F, \text{ and for all } Q_1 \in \mathcal{Q} : (q, 0) \notin Q_1.$$

In summary, given a Büchi alternating automaton M whose state set is of size n , and whose set of distinguished states is of size d , the size of the Büchi nondeterministic automaton simulating M , constructed by theorem 3 is bounded by $2^d 3^{n-d}$, i.e. $(2/3)^d 3^n$.

3. Mapping an LPTL formula into a Büchi alternating automaton

To make the satisfiability problem of LPTL use the result of the last section on the emptiness problem of Büchi alternating automata, we investigate in this section the problem of

mapping an LPTL formula into a Büchi alternating automaton accepting its models.

3.1 Background: the logic LPTL

LPTL is an extension of the classical propositional logic. This extension is obtained by the adding of the temporal operators \bigcirc (the “next”), \diamond (the “eventually”) and U (the “until”).

Syntax

LPTL formulas are built from the following alphabet:

- a countable set \mathcal{P} of atomic propositions p, q, r, \dots ,
- the boolean constructors \wedge and \neg , and
- the temporal operators \bigcirc, \diamond and U .

The set of LPTL (well formed) formulas is defined as the least set verifying:

- (a) every atomic proposition $p \in \mathcal{P}$ is a formula, and
- (b) if f and g are formulas, then so are $f \wedge g, \neg f, \bigcirc f, \diamond f$ and fUg ,

Remark 2. The temporal operator G (the “always”) is used as an abbreviation of $\neg\diamond\neg$: $Gf \equiv \neg\diamond\neg f$.

Semantics

LPTL is complete for the class \mathcal{K} of structures $\xi = (S, N, \pi)$ defined as follows (Manna & Wolper 1984; Wolper 1983):

- (a) S is a countable state set,
- (b) $N: S \longrightarrow S$ is a successor function mapping each state s into a unique successor state $N(s)$, and
- (c) $\pi: S \longrightarrow 2^{\mathcal{P}}$ is a function mapping each state s into a set of atomic propositions.

Remark 3. In a structure $\xi = (S, N, \pi)$, the function π partitions in each state $s \in S$ the set \mathcal{P} of atomic propositions into the set $\pi(s)$ of atomic propositions true in s , and the set $\mathcal{P} \setminus \pi(s)$ of atomic propositions false in s . Hence, π is a function assigning truth values to atomic propositions in each state.

Satisfiability

Let $\xi = (S, N, \pi)$ be a structure of the class \mathcal{K} , and s a state from S . The satisfiability of an LPTL formula f by the state s of the structure, denoted by $\langle \xi, s \rangle \models f$, is defined recursively as follows:

- (a) if f is an atomic proposition then: $\langle \xi, s \rangle \models f$ if and only if $f \in \pi(s)$,
- (b) $\langle \xi, s \rangle \models f_1 \wedge f_2$ if and only if $\langle \xi, s \rangle \models f_1$ and $\langle \xi, s \rangle \models f_2$,

- (c) $\langle \xi, s \rangle \models \neg f_1$ if and only if *not* ($\langle \xi, s \rangle \models f_1$),
- (d) $\langle \xi, s \rangle \models \bigcirc f_1$ if and only if $\langle \xi, N(s) \rangle \models f_1$,
- (e) $\langle \xi, s \rangle \models \diamond f_1$ if and only if $(\exists i \geq 0)(\langle \xi, N^i(s) \rangle \models f_1)$, and
- (f) $\langle \xi, s \rangle \models f_1 U f_2$ if and only if
 - $(\forall i \geq 0)(\langle \xi, N^i(s) \rangle \models f_1)$, or
 - $(\exists i \geq 0)(\langle \xi, N^i(s) \rangle \models f_2$ and $\forall j(0 \leq j < i \Rightarrow \langle \xi, N^j(s) \rangle \models f_1))$.

Remark 4. In the definition above of satisfiability:

- $N^0(s)$ stands for s , and
- $N^{i+1}(s)$, $i \geq 0$, stands for $N(N^i(s))$.

An interpretation ι consists of a structure $\xi = (S, N, \pi)$ and an initial state s_0 : it is denoted by

$$\iota = \langle \xi, s_0 \rangle.$$

An interpretation $\iota = \langle \xi, s_0 \rangle$ satisfies a formula f if and only if

$$\langle \xi, s_0 \rangle \models f.$$

An interpretation satisfying a formula is a model of that formula.

The satisfiability problem for LPTL consists of answering the question of whether a given formula of this logic is satisfiable; that is, whether it has a model.

3.2 Büchi automata on interpretations

The automata we are concerned with in this section are automata on interpretations. That is, automata of infinite sequences of sets of atomic propositions.

Given a set A of literals built from a set \mathcal{P} of atomic propositions, we define $\neg A$ as being the following set of literals:

$$\neg A = \{p : p \in \mathcal{P} \text{ and } \neg p \in A\} \cup \{\neg p : p \in \mathcal{P} \cap A\}.$$

DEFINITION 4

A Büchi nondeterministic automaton on interpretations is a 5-tuple $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ defined as follows:

- Σ is the input alphabet: $\Sigma = \mathcal{P} \cup \neg \mathcal{P}$, \mathcal{P} being a countable set of atomic propositions,
- Q is a finite state set,
- $\delta : Q \longrightarrow 2^{2^\Sigma \times Q}$ is a transition function,
- $q_0 \in Q$ is the initial state of M , and
- $F \subseteq Q$ is a set of distinguished states.

q	$\delta(q)$
q_0	$\{(\{\}, q_1)\}$
q_1	$\{(\{\}, q_2), (\{\}, q_3)\}$
q_2	$\{(\{j\}, q_2), (\{\neg c, j\}, q_4), (\{\neg e, j\}, q_4), (\{\neg c, j\}, q_5), (\{\neg e, j\}, q_5)\}$
q_3	$\{(\{j\}, q_2), (\{j\}, q_3), (\{j\}, q_6)\}$
q_4	$\{(\{j\}, q_2), (\{\neg c, j\}, q_3), (\{\neg e, j\}, q_3)\}$
q_5	$\{(\{j\}, q_2), (\{j\}, q_3)\}$
q_6	$\{(\{j\}, q_2), (\{\neg c, j\}, q_3), (\{\neg e, j\}, q_3), (\{j\}, q_6)\}$

Figure 2. the transition function δ of the automaton M of the example.

The transition function maps each state into a set of subsets of $2^\Sigma \times Q$; that is, $\delta(q)$ has the form $\{(A_{i_1}, q_{j_1}), \dots, (A_{i_m}, q_{j_m}) \mid A_{i_k} \subseteq \Sigma, q_{j_k} \in Q, \forall k = 1, \dots, m\}$. As shown in the definition below, if $(A, q) \in \delta(q)$ then the intuitive meaning of the literal in A is the following: the positive literals give atomic propositions that are necessary, and the negative literals give the atomic propositions that are forbidden.

DEFINITION 5

A run of a Büchi nondeterministic automaton $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ on an interpretation $\iota = e_0 e_1 e_2 \dots e_n \dots$ is an infinite sequence $c = q_{i_0} q_{i_1} q_{i_2} \dots q_{i_n} \dots$ of states of M verifying the following:

1. $q_{i_0} = q_0$,
2. $\forall n \geq 0 \exists (A, q) \in \delta(q_{i_n})$ such that :
 - (a) $q = q_{i_{n+1}}$, and
 - (b) for all atomic proposition p :
 - i. $(p \in A) \Rightarrow (p \in e_n)$,
 - ii. $(\neg p \in A) \Rightarrow (p \notin e_n)$.

The point 5 says intuitively that the positive literals of A are necessary, and that the negative ones forbid their corresponding atomic propositions. The presence or absence of an atomic proposition for which neither of the two corresponding literals (neither the positive nor the negative) belong to A is irrelevant.

Example. Let us consider the following Büchi nondeterministic automaton $M' = \langle Q, \Sigma, \delta, q_0, F \rangle$:

- $\Sigma = \mathcal{P} \cup \neg\mathcal{P}$, with $\mathcal{P} = \{c, e, j\}$,
- $Q = \{q_0, q_1, \dots, q_6\}$,
- the initial state is q_0 ,
- the transition function is given by the table of figure 2, and
- $F = \{q_0, q_1, q_4, q_5\}$.

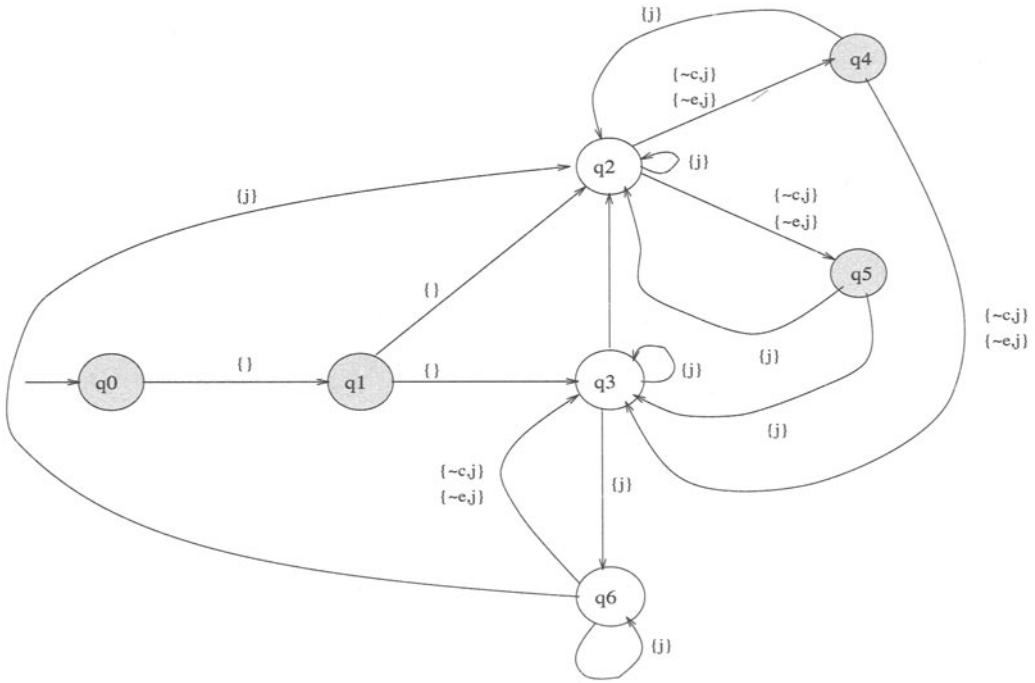


Figure 3. The graphical representation of the automaton M of the example.

The graphical representation of the automaton M is given by figure 3. The dashed states are the distinguished states of M . The incoming arrow shows the initial state.

The infinite sequence $q_0q_1(q_2q_4)^\omega$ is a run of the automaton M on each of the interpretations $\{c, e, j\}^2(\{c, j\}\{e, j\})^\omega$, $\{c, e\}\{c, j\}(\{j\}\{c, j\})^\omega$ and $\{j\}\{c\}(\{j\}\{c, e, j\})^\omega$. The run $q_0q_1(q_2q_4)^\omega$ is accepting, for it repeats infinitely often the distinguished state q_4 .

DEFINITION 6

A Büchi alternating automaton on interpretations is a 5-tuple $M = \langle \mathcal{L}(\Sigma \cup Q), \delta, q_0, q_v, F \rangle$ defined in the following manner:

- Σ is the input alphabet: $\Sigma = \mathcal{P} \cup \neg\mathcal{P}$, \mathcal{P} being a countable set of atomic propositions,
- $Q \cup \{q_v\}$ is a finite state set, $q_v \notin Q$ being a special state called “the valid state” of M ,
- $\mathcal{L}(\Sigma \cup Q)$ is the free distributive lattice generated by $\Sigma \cup Q$,
- $\delta : Q \cup \{q_v\} \longrightarrow \mathcal{L}(\Sigma \cup Q) \cup \{q_v\}$ is a transition function verifying:

$$\delta(q_v) = q_v,$$

- $q_0 \in Q$ is the initial state of M , and
- F is a set of distinguished states including q_v .

DEFINITION 7

A run of a Büchi alternating automaton $M = \langle \mathcal{L}(\Sigma \cup Q), \delta, q_0, q_v, F \rangle$ on an interpretation $\iota = e_0e_1 \dots e_n \dots \in (2^\Sigma)^\omega$ is an infinite labelled tree defined as follows:

1. all node of the tree is labelled by a state of M ,
2. the label of the root is q_0 , the initial state of M , and
3. (a) if
 - i. v is a node of level n labelled by q ,
 - ii. e_n is the n^{th} element of the interpretation ι , i.e. the n^{th} element of the infinite sequence $e_0e_1 \dots e_n \dots$,
 - iii. $\{q_1, \dots, q_m\}$ is the set of labels of of the immediate successors of v , and
 - iv. $t_1 \vee t_2 \vee \dots \vee t_r$ is the disjunctive normal form of $\delta(q)$,
 (b) then there exists $j \in \{1, \dots, r\}$ such that:
 - i. $\{q_1, \dots, q_m\} = \begin{cases} \{p \in Q : p \text{ occurs in } t_j\} & \text{if } \{p \in Q : p \text{ occurs in } t_j\} \neq \emptyset, \\ \{q_v\} & \text{otherwise,} \end{cases}$
 - ii. $\{p \in \mathcal{P} : p \text{ occurs in } t_j\} \subseteq e_n$, and
 - iii. $e_n \cap \{p \in \mathcal{P} : \neg p \text{ occurs in } t_j\} = \emptyset$.

As in the usual nondeterministic case, the points 7 and 7 say intuitively that the positive literals of t_j give the necessary atomic relations, and that the negative ones give the atomic propositions that are forbidden; the presence or absence of the atomic propositions for which neither of the corresponding literals belong to t_j is irrelevant. We need some other definitions before giving the construction mapping an LPTL formula into a Büchi alternating automaton accepting its models.

DEFINITION 8

The set $Subf(f)$ of subformulas of an LPTL formula f is defined recursively as follows:

- if f is an atomic proposition then $Subf(f) = \{f\}$,
- $Subf(\theta f) = \{\theta f\} \cup Subf(f)$, $\theta \in \{\neg, \bigcirc, \diamond\}$,
- $Subf(f\theta g) = \{f\theta g\} \cup Subf(f) \cup Subf(g)$, $\theta \in \{\wedge, U\}$.

DEFINITION 9

An elementary formula is a formula f either of the following forms:

- f is a literal, or
- f is prefixed by the temporal operator \bigcirc , i.e. \bigcirc is the main operator of f (f of the form $\bigcirc g$).

DEFINITION 10

An eventuality is a formula of the form $\diamond f$ or $\neg(fUg)$.

The following two definitions are the key points in the construction of a Büchi alternating automaton accepting the models of an LPTL formula. The first concerns the decomposition of a formula into elementary ones, and will be used for finding the transition function of the alternating automaton to associate to a formula. The second definition defines the closure of a formula, which will be used for determining the state set of the alternating automaton. These definitions are based on the following equivalences which are straightforward from the definition of satisfiability:

$$\begin{aligned}
 (\text{equiv 1}) \quad \diamond f &\equiv f \vee \bigcirc \diamond f \\
 (\text{equiv 2}) \quad f_1 U f_2 &\equiv f_2 \vee f_1 \wedge \bigcirc (f_1 U f_2) \\
 (\text{equiv 3}) \quad \neg \diamond f &\equiv \neg f \wedge \bigcirc \neg \diamond f \\
 (\text{equiv 4}) \quad \neg (f_1 U f_2) &\equiv \neg f_2 \wedge (\neg f_1 \vee \bigcirc \neg (f_1 U f_2)) \\
 (\text{equiv 5}) \quad \neg \bigcirc f &\equiv \bigcirc \neg f
 \end{aligned}$$

DEFINITION 11

The decomposition, $\text{elem}(f)$, of an LPTL formula f into elementary formulas is defined recursively as follows:

- if f is a literal: $\text{elem}(f) = f$,
- $\text{elem}(\neg \neg f_1) = \text{elem}(f_1)$,
- $\text{elem}(f_1 \wedge f_2) = \text{elem}(f_1) \wedge \text{elem}(f_2)$,
- $\text{elem}(\neg (f_1 \wedge f_2)) = \text{elem}(\neg f_1) \vee \text{elem}(\neg f_2)$,
- $\text{elem}(\diamond f_1) = \text{elem}(f_1) \vee \bigcirc \diamond f_1$,
- $\text{elem}(f_1 U f_2) = \text{elem}(f_2) \vee \text{elem}(f_1) \wedge \bigcirc (f_1 U f_2)$,
- $\text{elem}(\bigcirc f_1) = \bigcirc f_1$,
- $\text{elem}(\neg \diamond f_1) = \text{elem}(\neg f_1) \wedge \bigcirc \neg \diamond f_1$,
- $\text{elem}(\neg (f_1 U f_2)) = \text{elem}(\neg f_2) \wedge (\text{elem}(\neg f_1) \vee \bigcirc \neg (f_1 U f_2))$,
- $\text{elem}(\neg \bigcirc f_1) = \bigcirc \neg f_1$.

DEFINITION 12

The closure, $\text{cl}(f)$, of an LPTL formula f is defined recursively as follows:

- if f is a literal: $\text{cl}(f) = \emptyset$,
- $\text{cl}(\neg \neg f_1) = \text{cl}(f_1)$,
- $\text{cl}(f_1 \wedge f_2) = \text{cl}(f_1) \cup \text{cl}(f_2)$,
- $\text{cl}(\neg (f_1 \wedge f_2)) = \text{cl}(\neg f_1) \cup \text{cl}(\neg f_2)$,
- $\text{cl}(\diamond f_1) = \text{cl}(f_1) \cup \{\diamond f_1\}$,
- $\text{cl}(f_1 U f_2) = \text{cl}(f_2) \cup \text{cl}(f_1) \cup \{f_1 U f_2\}$,
- $\text{cl}(\bigcirc f_1) = \text{cl}(f_1) \cup \{f_1\}$,
- $\text{cl}(\neg \diamond f_1) = \text{cl}(\neg f_1) \cup \{\neg f_1\}$,
- $\text{cl}(\neg (f_1 U f_2)) = \text{cl}(\neg f_2) \cup \text{cl}(\neg f_1) \cup \{\neg (f_1 U f_2)\}$,
- $\text{cl}(\neg \bigcirc f_1) = \text{cl}(\neg f_1) \cup \{\neg f_1\}$.

3.3 Büchi alternating automaton of an LPTL formula

The following theorem gives an effective construction of a Büchi alternating automaton accepting the models of an LPTL formula.

Theorem 4. *Let f be an LPTL formula. The set of models of f is the language accepted by the Büchi alternating automaton $B_f = \langle \mathcal{L}(\Sigma \cup Q), \delta, q_0, q_v, F \rangle$, called Büchi alternating automaton of f , defined in the following manner:*

- $\Sigma = \mathcal{P} \cup \neg\mathcal{P}$, \mathcal{P} being the set of atomic propositions occurring in f ,
- $Q = \{\langle f \rangle\} \cup \{\langle g \rangle : g \in \text{cl}(f)\}$,
- $\delta(\langle g \rangle)$ is the result obtained by substituting, for every h , $\langle h \rangle$ for each non-nested occurrence² $\bigcirc h$,
- $q_0 = \langle f \rangle$, and
- the set F of distinguished states is:

$$F = \{\langle g \rangle \in Q : g \text{ is not an eventuality formula}\} \cup \{q_v\}.$$

Proof. The only point we clarify is the choice of the set F of distinguished states. For the initial formula to be satisfied, there should exist an interpretation satisfying it, that is a model of it, on which one can construct a run of the automaton verifying the following: every time we meet on a branch a node labelled by a state $\langle g \rangle$ with g being an eventuality formula, we can find, along the suffix of that branch beginning at that node, a state label by $\langle h \rangle$ is such that h is not an eventuality formula. Stated otherwise, the run should repeat infinitely often, on each of its branches, states $\langle g \rangle$ such that g is not an eventuality. Hence the result. \square

4. Related work

The problem of simulating, in the case of Büchi, an alternating automaton by a usual non-deterministic one has been investigated by other authors (Miyano & Ayashi 1984; Muller & Schupp 1993). However, we believe that our method offers an easier implementation. In fact, in (Miyano & Ayashi 1984; Muller & Schupp 1993), the authors were mainly interested in proving that alternating automata have the same expressive power than usual nondeterministic automata, and their purpose was not to give an easily implementable translation.

Another automata-based approach to the satisfiability problem of LPTL is well-known in the literature Vardi & Wolper (1986). Vardi and Wolper's method maps an LPTL formula into a Büchi nondeterministic automaton by performing the cross product of a first automaton called "local automaton" and a second automaton called "eventuality automaton". Each of the local automaton and the eventuality automaton (and the final Büchi nondeterministic automaton) is of size single-exponential in the length of the input formula. The Büchi nondeterministic automaton our alternating automata approach associates to an LPTL formula is also of size single-exponential in the length of the input formula. However, our method is more natural; furthermore, the intermediate automaton used (the Büchi alternating automaton) is of size linear in the length of the input formula (the size is the cardinality of closure).

²An occurrence which is not in the scope of a temporal operator.

5. Conclusion

The investigation of this paper can be viewed as an alternating automata approach to the satisfiability problem of Linear Propositional Temporal Logic (LPTL Pnueli (1981)). We first gave a method for translating, in the case of Büchi, any alternating automaton into a usual nondeterministic one. We then provided a method for mapping any LPTL formula into a Büchi alternating automaton accepting its models.

Our current concern is the minimal model property for LPTL: we strongly believe that one can use the decreasing property of very weak alternating automata (see Isli (1993) for the class of very weak alternating automata, and (Muller *et al* 1988; Muller *et al* 1992) for the class of weak alternating automata which has at least an equal expressive power) to improve the minimal model property for LPTL.

I am indebted to my thesis advisor Professor Ahmed SAOUDI, who has been a major contributor to this work. He passed away on August 11, 1993.

References

- Büchi J B 1960 On a decision method in restricted second order theory. *Proc. Int. Congress on Logic, Method and Philosophy of Science* (Stanford, CA: Univ. Press)
- Chandra A K, Kozen D C, Stockmeyer L J 1981 Alternation. *J. Assoc. Comput. Mach.* 28: 114–133
- Isli A 1993 *Automates Alternants et Logiques Temporelles, Satisfaction de Contraintes Temporelles*. Ph D thesis, LIPN, Université Paris 13, France
- Isli A 1994 Mapping an LPTL formula into a Büchi alternating automaton accepting its models. In *Proceedings of the Workshop of the International Conference on Temporal Logic (ICTL)*, Bonn, pp 85–90
- Manna Z, Wolper P 1984 Synthesis of communicating processes from temporal logic specifications. *ACM Trans. Program. Languages Syst.* 6: 68–93
- McNaughton R 1966 Testing and generating infinite sequences by a finite automaton. *Inf. Contr.* 9: 521–530
- Miyano S, Ayashi T 1984 Alternating automata on ω -words. *Theor. Comput. Sci.* 32: 321–330
- Muller D E 1963 Infinite sequences and finite machines. *Switching circuit theory and logical design: Proc. Fourth Annu. Symp.* (New York: IEEE Press)
- Muller D E, Schupp P E 1985 *Alternating automata on infinite objects, determinacy and Rabin's theorem*. LNCS 192 (Berlin: Springer) pp 100–107
- Muller D E, Schupp P E 1987 Alternating automata on infinite trees. *Theor. Comput. Sci.* 54: 267–276
- Muller D E, Schupp P E 1993 Simulating alternating tree automata by nondeterministic automata. Technical report 93.50, LITP, Institut Blaise Pascal, Paris, October
- Muller D E, Saoudi A, Schupp P E 1988 Weak alternating automata give a simple explanation of why most temporal and dynamic logics are decidable in exponential time. In *Proceedings 3rd IEEE Annu. Symp. on Logic in Computer Science* pp 422–427
- Muller D E, Saoudi A, Schupp P E 1992 Alternating automata, The weak monadic theory of trees and its complexity. *Theor. Comput. Sci.* 97: 233–244
- Pnueli A 1981 The temporal semantics of concurrent programs. *Theor. Comput. Sci.* 13: 45–60

- Vardi M Y, Wolper P 1986 An automata-theoretic approach to automatic program verification. *IEEE Symposium on Logic In Computer Science*, pp 332–344
- Wolper P 1983 Temporal logic can be more expressive. *Inf. Contr.* 56: 72–99

Appendix A: Proof of theorem 1

Let $M = \langle \mathcal{L}(Q), \Sigma, \delta, q_0, F \rangle$ be a Büchi alternating automaton, and u an infinite word accepted by M . There exists an accepting run of M on u : let t be such a run. We define a sequence $(t_n)_{n \geq 0}$ of runs of M on u in the following fashion:

- (a) $t_0 = t$,
- (b) for all $n \geq 0$, t_{n+1} is obtained from t_n as follows: let $NODES(t_n, n + 1, q)$ be the set of nodes of t_n of level $n + 1$ labelled by q . For all $q \in Q$ such that $|NODES(t_n, n + 1, q)| \geq 2$, we perform the following operations:
 - (b1) choose a node v from $NODES(t_n, n + 1, q)$,
 - (b2) for every node v' belonging to $NODES(t_n, n + 1, q) \setminus \{v\}$, we substitute the subtree t_n/v of t_n at v for the subtree t_n/v' of t_n at v' .

It is easy to see that for all $n \geq 0$, t_n is an accepting run of M on u . The limit of the sequence $(t_n)_{n \geq 0}$ is clearly an accepting uniform run of M on u . □