

Equivalence and Inclusion Problem for Strongly Unambiguous Büchi Automata

Nicolas Bousquet¹ and Christof Löding²

¹ ENS Cachan, France

`nbousque@dptinfo.ens-cachan.fr`

² RWTH Aachen, Informatik 7, 52056 Aachen, Germany

`loeding@cs.rwth-aachen.de`

Abstract. We consider the inclusion and equivalence problem for unambiguous Büchi automata. We show that for a strong version of unambiguity introduced by Carton and Michel these two problems are solvable in polynomial time. We generalize this to Büchi automata with a fixed finite degree of ambiguity in the strong sense. We also discuss the problems that arise when considering the decision problems for the standard notion of ambiguity for Büchi automata.

1 Introduction

The model of unambiguous automata is located between deterministic and nondeterministic automata. An unambiguous automaton is a nondeterministic automaton such that each input that is accepted has a unique accepting run. The concept of unambiguity also occurs in other areas of theoretical computer science, for example in complexity theory. The problems solvable in polynomial time by unambiguous (nondeterministic) Turing machines are collected in the subclass UP (Unambiguous Polynomial time) of NP [1].

There are two aspects in the study of unambiguous automata: expressiveness and computational complexity. Concerning expressiveness, because of the well-known equivalence between deterministic and nondeterministic finite automata over finite words and trees, unambiguous automata can recognize all the regular languages over these two domains. For automata over ω -words it is known that deterministic Büchi automata are strictly less expressive than nondeterministic ones [2]. However, Arnold showed that all the ω -regular languages can be recognized by an unambiguous Büchi automaton [3]. For automata over ω -trees, the class of unambiguous automata is not as expressive as the class of full nondeterministic tree automata (with standard acceptance conditions like parity, Rabin or Muller) [4, 5].

An interesting subclass of the class of unambiguous Büchi automata is considered by Carton and Michel [6]: Their definition requires that for each input (accepted or not) there is a unique run passing infinitely often through a final state (whether from an initial state or not). Thus, an infinite word is accepted if the initial state is the first state of the path passing infinitely often through

a final state. Non-acceptance means that the unique path of this form does not start in an initial state. Carton and Michel show [6] that this restricted class of Büchi automata suffices to capture the class of ω -regular languages. We consider in this paper a slight modification of their definition, and refer to these automata as strongly unambiguous.

The second interesting aspect for unambiguous models is the computational complexity of algorithmic problems. We consider here the equivalence problem (as well as the inclusion problem which turns out to have the same complexity). It is well known that there is a gap in complexity for the equivalence problem between deterministic and nondeterministic automata. The problem can be decided in polynomial time over finite words [7] and finite trees for deterministic automata, whereas the problem is PSPACE-complete over finite words (see Section 10.6 of [8]³) and EXPTIME-complete over finite trees [9] for nondeterministic automata.

As shown by Stearns and Hunt, the equivalence problem for unambiguous finite automata over finite words is still polynomial [10] and Seidl showed the same over finite trees [11]. In the present paper, we show that this result also holds for strongly unambiguous Büchi automata. To our knowledge, this identifies the first subclass of Büchi automata that is expressively complete for the ω -regular languages and at the same time allows a polynomial time equivalence test.

The polynomial time equivalence test over finite words from [10] uses a counting argument: The main idea is that, for unambiguous automata, the number of accepting runs is equal to the number of accepted inputs. Stearns and Hunt proved that it is sufficient to count the number of accepting paths of the given unambiguous automata only up to a certain length and that this can be done in polynomial time. The problem when trying to adapt such an approach to Büchi automata is that runs of Büchi automata are infinite and one cannot simply count the number of accepted words up to a certain length. However, it is possible to restrict the problem of equivalence of regular languages of infinite words to ultimately periodic words [12] (see also [13]). A word is ultimately periodic if it is of the form $u \cdot v^\omega$ where u and v are finite. It turns out that this restriction to ultimately periodic words allows to adapt the counting argument to the case of strongly unambiguous Büchi automata. Instead of presenting a direct adaption of the proof for finite words we show that the equivalence problem for strongly unambiguous automata can be reduced in polynomial time to the equivalence problem for unambiguous automata on finite words.

This kind of reduction does not seem to work for unambiguous Büchi automata. We show that deciding whether an unambiguous Büchi automaton (even a deterministic one) accepts some periodic word v^ω , where v is of a given length n , is NP-complete. Although this proof does not show that the equivalence prob-

³ In [8] the PSPACE-hardness of the non-universality problem for regular expressions is shown. This can easily be turned into a PSPACE-hardness proof for the equivalence problem for nondeterministic finite automata.

lem for unambiguous Büchi automata is difficult, it shows that different methods are required.

The remainder of the paper is structured as follows. In the second section we give some definitions and simple properties of Büchi automata. In Section 3 we show how to reduce the equivalence problem for strongly unambiguous Büchi automata to the case of unambiguous automata on finite words. In Section 4 we extend these results to strongly k -ambiguous automata, a relaxed notion of strong unambiguity, where each word can have at most k final paths. In Section 5 we show that deciding if a deterministic Büchi automaton accepts periodic words of a given length is NP-complete. We conclude in the last section.

2 Definitions and Background

For an alphabet Σ we denote as usual the set of finite words over Σ by Σ^* , the set of nonempty finite words by Σ^+ , and the set of infinite words by Σ^ω . The length of a finite word $u \in \Sigma^*$ is denoted by $|u|$. For an infinite word $\alpha \in \Sigma^\omega$ we denote the j th letter by $\alpha(j)$, i.e., $\alpha = \alpha(0)\alpha(1)\cdots$.

An infinite word of the form $uv^\omega = uvvv\cdots$ for finite words u, v is called ultimately periodic.

We consider nondeterministic finite automata (NFA) on finite words of the form $A = (Q, \Sigma, Q_{\text{in}}, \Delta, F)$, where Q is a finite set of states, Σ is the input alphabet, $Q_{\text{in}} \subseteq Q$ is the set of initial states, $\Delta \subseteq Q \times \Sigma \times Q$ is the transition relation, and $F \subseteq Q$ is the set of final states. We use the standard terminology for NFAs (see e.g. [14]) and denote the language of words accepted by A by $L(A)$.

A Büchi automaton $A = (Q, \Sigma, Q_{\text{in}}, \Delta, F)$ is of the same form as an NFA. In contrast to NFAs, a Büchi automaton defines a language of infinite words. A *path* for the infinite word $\alpha \in \Sigma^\omega$ is an infinite sequence of states $q_0q_1\cdots$ such that for all $j \in \mathbb{N}$, $(q_{j-1}, \alpha(j), q_j) \in \Delta$. A *final path* is a path that passes infinitely often through a final state. A path begins in q if $q_0 = q$. If a final path for α begins in some $q_0 \in Q_{\text{in}}$, then the word α is *accepted* by A . So a final path is *accepting* if it starts with an initial state. If an accepting path for α exists, then A *accepts* α . The *language* $L(A)$ is the set of infinite words α accepted by A . For an automaton $A = (Q, \Sigma, Q_{\text{in}}, \Delta, F)$ we denote by $L_A(q)$ the language accepted by $(Q, \Sigma, \{q\}, \Delta, F)$. The class of languages that can be accepted by Büchi automata is called the class of *ω -regular languages*.

For a finite word u and two states q, q' of A we write $A : q \xrightarrow{u} q'$ if one can reach q' from q on reading u , and we write $A : q \xrightarrow{u, F} q'$ if one can reach q' from q on reading u and by passing through a final state on the way.

It is well known that the equivalence problem for NFAs, i.e., the question whether two given NFAs accept the same language, is PSPACE-complete (see [8]). The same holds for the inclusion problem because equivalence can easily be tested by checking for both inclusions. Furthermore, the lower bound on the complexity easily extends to Büchi automata.

In [10] unambiguous NFAs are considered and it is shown that the equivalence and inclusion problem for these automata are solvable in polynomial time. Unambiguous automata are nondeterministic automata in which for each word there is at most one accepting path.

Our aim is to see to what extent these results can be lifted to Büchi automata. We introduce two notions of unambiguity, the standard one and a stronger notion introduced in [6].

Definition 1. *A Büchi automaton A is called unambiguous if every infinite word has at most one accepting path in A , and it is called strongly unambiguous if every infinite word has at most one final path.*

Clearly, if A is strongly unambiguous, then for each infinite word α there is at most one state q such that $\alpha \in L_A(q)$. We state this observation as a remark for later reference.

Remark 1. Let A be a strongly unambiguous Büchi automaton and α be an infinite word. If $\alpha \in L_A(q_1)$ and $\alpha \in L_A(q_2)$ then $q_1 = q_2$.

Consider, for example, the automata shown in Figure 1. Both automata accept the language over $\{a, b\}$ consisting of all words that contain infinitely many b . The automaton on the left-hand side is deterministic (where deterministic automata as usual only have a single initial state and for each state and letter at most one outgoing transition) and therefore unambiguous, but it is not strongly unambiguous: the word b^ω is accepted from both states q_0 and q_1 . The automaton on the right-hand side is strongly unambiguous. It accepts the same language as the deterministic automaton, but from state p_0 all accepted words start with a , and from p_1 all accepted words start with b .

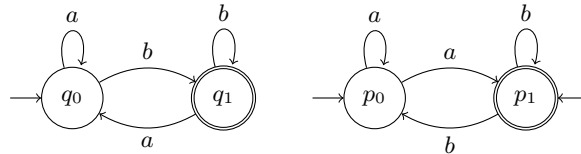


Fig. 1. Example for a deterministic Büchi automaton (left-hand side) and a strongly unambiguous Büchi automaton (right-hand side) for the same language

Note that each strongly unambiguous automaton is unambiguous because each accepting path is also a final path. It has been shown in [3] that each ω -regular language can be accepted by an unambiguous Büchi automaton. The class of strongly unambiguous automata has been introduced in [6]⁴ and it has been shown that this class is expressively complete for the ω -regular languages.

⁴ The definition in [6] is even more restrictive: It is required that each word has exactly one final path. This allows an easy complementation by complementing the set of

Theorem 1 ([6]). *Every ω -regular language can be recognized by a strongly unambiguous Büchi automaton.*

This expressive completeness makes strongly unambiguous Büchi automata an interesting class. It is also worth noting that strongly unambiguous Büchi automata naturally occur in the translation from linear temporal logic formulas into Büchi automata. In the standard approach for this translation the Büchi automaton guesses valuations of all subformulas of the given formula and verifies that the guesses are correct (see [15]). An input word is accepted from the unique state that evaluates all subformulas correctly. Hence the automaton that is constructed in this standard way is strongly unambiguous.

Before we turn to the decision problems for strongly unambiguous Büchi automata, we compare them to deterministic automata. Note that deterministic Büchi automata do not capture the full class of ω -regular languages, but using extended acceptance conditions like the Muller acceptance condition, deterministic automata become expressively complete [16] (see also [17]).

The example from Figure 1 already shows that deterministic Büchi automata need not to be strongly unambiguous. In fact, there is no deterministic Büchi automaton that is strongly unambiguous and equivalent to the one from Figure 1: Assume A is a deterministic and strongly unambiguous Büchi automaton accepting all words containing infinitely many b . Then the word b^ω is accepted from the unique initial state. Then b^ω is not accepted from any other state by Remark 1. Thus, the initial state is final and has a b -loop. The only way to accept ab^ω would be to also have an a -loop on the initial state. This would mean that A accepts all ω -words over $\{a, b\}$. Note that we only used the fact that a deterministic automaton has only a single initial state. So this example shows that a set of initial states is necessary for strongly unambiguous Büchi automata, in general.

The next example shows that strongly unambiguous automata can be exponentially more succinct than deterministic ones. We formulate the following remark for deterministic Muller automata, because the Muller condition is the most general one of the standard acceptance conditions that are usually considered: A Muller condition is specified by a family \mathcal{F} of state sets. A run is accepting if the set of states that appear infinitely often in this run is a member of \mathcal{F} .

Remark 2. There is a family $(L_n)_{n \geq 1}$ of ω -languages over the alphabet $\{a, b\}$ such that each L_n can be accepted by a strongly unambiguous Büchi automaton with $n + 2$ states, and each deterministic Muller automaton for L_n needs at least 2^n states.

Proof. We use the standard syntactic right-congruence for ω -languages L , defined by $u \sim_L v$ iff $u\alpha \in L \Leftrightarrow v\alpha \in L$ for all $\alpha \in \Sigma^\omega$. As for automata on finite

initial states. We have chosen the more relaxed notion because the polynomial time equivalence test also works in this setting. Further note that in [6] these automata are simply called unambiguous and not strongly unambiguous.

words (see [14]), one can show that each Muller automaton for L needs at least as many states as there are classes of \sim_L .

The language $L_n = \Sigma^* a \Sigma^{n-1} a b^\omega$ can be recognized by a strongly unambiguous Büchi automaton of size $n + 2$ as shown in Figure 2. The number of \sim_{L_n} classes is at least 2^n , so a deterministic Muller automaton which recognizes L_n has at least 2^n states. \square

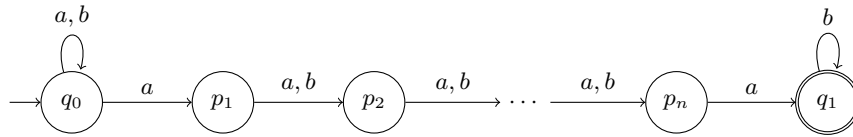


Fig. 2. A strongly unambiguous Büchi automaton for $L = \Sigma^* a \Sigma^{n-1} a b^\omega$

Finally, we would like to mention that there are also deterministic Büchi automata exponentially smaller than strongly unambiguous ones. This is in contrast to unambiguous automata, since each deterministic automaton is unambiguous.

Remark 3. There is a family $(L_n)_{n \geq 1}$ of ω -languages over the alphabet $\{a, b\}$ such that each L_n can be accepted by a deterministic Büchi automaton with $n + 1$ states, and each strongly unambiguous Büchi automaton for L_n needs at least 2^{n-1} states.

Proof. Let L_n be the language of all words in $\{a, b\}^\omega$ in which the n th letter is a . Using $n + 1$ states a deterministic automaton can check this property.

Let A_n be a strongly unambiguous automaton for L_n . Assume that A_n has less than 2^{n-1} states. Then there are two different words $w_1, w_2 \in \{a, b\}^{n-1}$ of length $n - 1$ such that $w_1 a b^\omega$ and $w_2 a b^\omega$ are accepted by A_n from the same state q . Since w_1 and w_2 are different, we can assume w.l.o.g. that $w_1 = w a w'_1$ and $w_2 = w b w'_2$ for some words w, w'_1, w'_2 . Let $m = |w|$ and v be a word of length $n - m - 1$. Then $v w a w'_1 b^\omega$ is in L_n . The corresponding accepting run must reach q after having read v because otherwise there would be two different final paths for $w a w'_1 b^\omega$. Hence, there is also an accepting run for $v w b w'_2 b^\omega$: the one that moves to q on reading v , and then accepting $w b w'_2 b^\omega$ from q . Since $w b w'_2 b^\omega \notin L_n$ we get a contradiction and thus A_n has at least 2^{n-1} states. \square

3 Equivalence for Strongly Unambiguous Büchi Automata

In this section we prove that the equivalence problem for strongly unambiguous Büchi automata can be solved in polynomial time. For the case of finite

words, the proof of Stearns and Hunt [10] uses a counting argument: For an unambiguous automaton over finite words the number of accepted words of a certain length is the same as the number of accepting paths of this length. The idea is to decide if $L(A_1) \cap L(A_2) = L(A_i)$ for $i = 1, 2$ by simply counting the number of accepting paths. Since $L(A_1) \cap L(A_2) \subseteq L(A_i)$, the equality holds if for each n there is the same number of accepting paths in the automaton for $L(A_1) \cap L(A_2)$ and in A_i . The key argument is then that a comparison of the number of accepting paths is sufficient up to a certain bound of the length.

For general infinite words it is impossible to count final paths in a reasonable way. However, there are infinite words that can be represented in a finite way: the ultimately periodic words introduced in the previous section. This class of words is particularly interesting because it can be used to characterize equivalence and inclusion of ω -regular languages. This easily follows from the closure properties of the class of ω -regular languages and the fact that each non-empty ω -regular language contains an ultimately periodic word.

Theorem 2 ([12]). *Let A_1 and A_2 be two Büchi automata.*

1. A_1 and A_2 accept the same language if and only if they accept the same ultimately periodic words.
2. $L(A_1) \subseteq L(A_2)$ if and only if the ultimately periodic words recognized by A_1 are in $L(A_2)$.

Using this fact one can indeed adapt the counting argument from [10] to count ultimately periodic words accepted by Büchi automata. However, instead of adapting the proof of [10] we can also give a reduction of the equivalence problem for strongly unambiguous Büchi automata to the equivalence problem for unambiguous automata over finite words. The idea of reducing decision problems for Büchi automata to automata over finite words has already been used in [13]. The main difference here is that the reduction is computable in polynomial time when starting from a strongly unambiguous automaton. Before we present the reduction we state a simple but important property of strongly unambiguous Büchi automata that makes our reduction work.

Lemma 1. *Let $A = (Q, \Sigma, Q_{\text{in}}, \Delta, F)$ be a strongly unambiguous Büchi automaton. An ultimately periodic word uv^ω is accepted by A iff there are states $q_0 \in Q_{\text{in}}$ and $q \in Q$ such that $A : q_0 \xrightarrow{u} q \xrightarrow{v}_F q$.*

Proof. Obviously, if $A : q_0 \xrightarrow{u} q \xrightarrow{v}_F q$ for states $q_0 \in Q_{\text{in}}$ and $q \in Q$, then uv^ω is accepted. Now suppose that uv^ω is accepted by A . Then there is an accepting path $\rho = q_0q_1\dots$ on uv^ω that starts in some $q_0 \in Q_{\text{in}}$. Let $q = q_{|u|}$ be the state reached in ρ after reading u , and let $q' = q_{|u|+|v|}$ be the state reached after reading uv . Then v^ω is in $L_A(q)$ and in $L_A(q')$ and thus, by Remark 1 we have $q = q'$. Furthermore, that path $A : q \xrightarrow{v} q$ must pass through a final state because otherwise there would be another accepting path for v^ω that starts from q . By prefixing this accepting path with the v -loop from q to q we would obtain more than one accepting path for v^ω starting in q , contradicting the strong unambiguity of A . \square

For the reduction to unambiguous automata on finite words we now build from a strongly unambiguous Büchi automaton an unambiguous automaton on finite words that accepts precisely the words of the form $u\#v$ such that uv^ω is accepted by A . By Theorem 2 two strongly unambiguous Büchi automata are equivalent iff the resulting finite automata are.

Let $A = (Q, \Sigma, Q_{\text{in}}, \Delta, F)$ be a strongly unambiguous Büchi automaton. The finite automaton we are constructing simulates A on the first part u of the input $u\#v$. When reading $\#$ it stores the current state q and continues reading the input. It accepts if it reaches q again after having read v , and if it has passed through a final state on the way. Since the automaton accepts codings of ultimately periodic words, we call it A_{up} , where the subscript abbreviates ultimately periodic.

Formally, $A_{\text{up}} = (Q', \Sigma \cup \{\#\}, Q_{\text{in}}, \Delta', F')$ is defined as follows:

- $Q' = Q \cup (Q \times Q \times \{0, 1\})$.
- Δ' contains
 - all transitions from Δ ,
 - all transitions of the form $(q, \#, (q, q, 0))$ with $q \in Q$, and
 - all transitions of the form $((q, p, i), a, (q, p', i'))$, where $(p, a, p') \in \Delta$, and

$$i' = \begin{cases} 1 & \text{if } p' \in F, \\ i & \text{if } p' \notin F. \end{cases}$$
- $F' = \{(q, q, 1) \mid q \in Q\}$.

Lemma 2. *The automaton A_{up} over finite words is unambiguous and accepts the language $L(A_{\text{up}}) = \{u\#v \in \Sigma^* \# \Sigma^+ \mid uv^\omega \in L(A)\}$.*

Proof. It is clear from the construction that A_{up} only accepts words of the form $u\#v \in \Sigma^* \# \Sigma^+$.

Furthermore, one easily sees that A_{up} accepts precisely those $u\#v$ such that there are states $q_0 \in Q_{\text{in}}$ and $q \in Q$ with $A : q_0 \xrightarrow{u} q$ and $A : q \xrightarrow{v} q$. Lemma 1 allows us to conclude that $L(A_{\text{up}}) = \{u\#v \in \Sigma^* \# \Sigma^+ \mid uv^\omega \in L(A)\}$.

The automaton A_{up} simulates the automaton A when reading its input. In particular, if there are two different paths for accepting $u\#v$, then there are also two different paths in A accepting uv^ω . Thus, since A is strongly unambiguous, A_{up} is unambiguous. \square

Since inclusion and equivalence for unambiguous automata on finite words are decidable in polynomial time [10], we obtain the following result for strongly unambiguous Büchi automata.

Theorem 3. *The inclusion and equivalence problem for strongly unambiguous Büchi automata are decidable in polynomial time.*

Proof. Given two strongly unambiguous Büchi automata A and B we transform them into the unambiguous automata A_{up} and B_{up} . For these we can test inclusion or equivalence in polynomial time. By Lemma 2 and Theorem 2 the corresponding result is correct for A and B . \square

4 Extension to Strongly k -Ambiguous Automata

An automaton over finite words is called k -ambiguous if each accepted word has at most k accepting runs. In [10] it is shown that equivalence and inclusion of k -ambiguous automata can be solved in polynomial time (for a fixed k). Following this idea, we extend the result from the previous section to strongly k -ambiguous Büchi automata, as defined below.

Definition 2. *A Büchi automaton A is called k -ambiguous if each infinite word has at most k accepting paths in A , and is called strongly k -ambiguous if each infinite word has at most k final paths in A .*

The accepting paths for ultimately periodic words are not as constrained in strongly k -ambiguous automata as they are in strongly unambiguous automata, but there is a similar characterization. The difference is that the loop on the periodic part can be longer, but it can consist of at most k repetitions of the periodic pattern.

Lemma 3. *Let $A = (Q, \Sigma, Q_{\text{in}}, \Delta, F)$ be a strongly k -ambiguous Büchi automaton. An ultimately periodic word uv^ω is accepted by A iff there are states $q_0 \in Q_{\text{in}}$ and $q_1, \dots, q_{k+1} \in Q$ such that*

$$A : q_0 \xrightarrow{u} q_1 \xrightarrow{v} q_2 \xrightarrow{v} \dots \xrightarrow{v} q_k \xrightarrow{v} q_{k+1} ,$$

$q_{k+1} = q_s$ for some $1 \leq s \leq k$, and $q_i \xrightarrow{v} q_{i+1}$ for some $s \leq i \leq k$.

Proof. The proof is similar to the proof of Lemma 1. Assume that uv^ω is accepted by A , let ρ be an accepting path, and let q_0, \dots, q_{k+1} be such that ρ starts as follows:

$$q_0 \xrightarrow{u} q_1 \xrightarrow{v} q_2 \xrightarrow{v} \dots \xrightarrow{v} q_{k+1}.$$

Then v^ω is in $L_A(q_i)$ for all $i \in \{1, \dots, k+1\}$. Since A is strongly k -ambiguous, there must be some $i \neq j$ such that $q_i = q_j$. Now assume that $q_{k+1} \neq q_s$ for all $s \in \{1, \dots, k\}$. Then we get arbitrarily many different accepting paths of A on uv^ω by repeating the loop between q_i and q_j an arbitrary number of times before continuing the path towards q_{k+1} . Hence, there must exist some $1 \leq s \leq k$ with $q_{k+1} = q_s$.

It remains to show that there is some $i \in \{s, \dots, k\}$ such that $q_i \xrightarrow{v} q_{i+1}$. Assume the contrary. Then we can again produce an arbitrary number of accepting paths for uv^ω by repeating the loop $q_s \xrightarrow{v} \dots q_k \xrightarrow{v} q_s$ (that does not contain a final state) before continuing the path with the accepting part after q_{k+1} in ρ . All these paths are different because we always increase the part that does not contain a final state before continuing with the part of ρ after q_{k+1} that contains infinitely many accepting states. This proves one direction of the claim. The other direction is obvious. \square

We now construct an automaton A_{up}^k that has the same property as A_{up} from the previous section. This automaton, after having read u and when reading $\#$,

guesses the states q_2, \dots, q_k and then verifies that the properties from Lemma 3 are satisfied.

Formally $A_{\text{up}}^k = (Q', \Sigma \cup \{\#\}, Q_{\text{in}}, \Delta', F')$ is constructed as follows:

- $Q' = Q \cup ((Q \times Q)^k \times \{0, \dots, k\})$.
- Δ' contains all transitions from Δ , all transitions of the form

$$(q_1, \#, [(q_1, q_1), \dots, (q_k, q_k), 0])$$

and all transitions of the form (written with an arrow for better readability)

$$[(q_1, p_1), \dots, (q_k, p_k), i] \xrightarrow{a} [(q_1, p'_1), \dots, (q_k, p'_k), i']$$

where $(p_j, a, p'_j) \in \Delta$ for all j , and

$$i' = \max(\{i\} \cup \{j \mid p'_j \in F\}).$$

- $F' = \{[(q_1, q_2), (q_2, q_3), \dots, (q_k, q_s), i] \mid 1 \leq s \leq k \text{ and } s \leq i \leq k\}$.

Lemma 4. *The automaton A_{up}^k over finite words is k -ambiguous and accepts the language $L(A_{\text{up}}^k) = \{u\#v \in \Sigma^* \# \Sigma^+ \mid uv^\omega \in L(A)\}$.*

Proof. The proof is based on Lemma 3 and is similar to the proof of Lemma 2. \square

As in the previous section we conclude:

Theorem 4. *For a fixed k , the inclusion and equivalence problem for strongly k -ambiguous Büchi automata can be decided in polynomial time.*

5 Periodic Words in Deterministic Automata

In this section we show that deciding for a given deterministic Büchi automaton whether it accepts a periodic word with a period of a given length is NP-complete. Since deterministic automata are special cases of unambiguous automata, this shows that it is unlikely that the counting techniques that work for unambiguous automata on finite words and for strongly unambiguous Büchi automata can be transferred to unambiguous Büchi automata.

More formally, consider for a class \mathcal{C} of Büchi automata the following decision problem PERIODIC(\mathcal{C}):

Given: A Büchi automaton A from the class \mathcal{C} and a natural number n .

Question: Does there exist a word v with $|v| \leq n$ such that A accepts v^ω ?

Proposition 1. *The problem PERIODIC(\mathcal{C}) is NP-complete for the class \mathcal{C} of deterministic Büchi automata.*

Proof. Membership in NP can easily be verified: If n is bigger than the number of states of A , then A accepts a periodic word of length n iff the initial state of A is contained in a loop with a final state. This can be checked in polynomial time by standard graph algorithms. If n is smaller than the number of states of A , then we can guess a word v of length at most n and verify in polynomial time whether v^ω is accepted by A .

For the NP-hardness we give a reduction from the satisfiability problem of Boolean formulas in conjunctive normal form (CNF). Let $\varphi = C_1 \wedge \dots \wedge C_k$ be such a formula over n variables x_1, \dots, x_n consisting of clauses C_1, \dots, C_k . A truth assignment of the variables can naturally be encoded by a word of length n over the alphabet $\{0, 1\}$: Position i of the word is 0 if x_i is false, and 1 if x_i is true.

The deterministic Büchi automaton that we construct works over the alphabet $\{0, 1, \#\}$ and only accepts words from $(\{0, 1\}^n \#)^\omega$: It basically consists of a loop of length $(n + 1) \cdot k$ that reads words $v_1 \# v_2 \# \dots \# v_k \#$ where each v_i is from $\{0, 1\}^n$. Each v_i encodes an assignment of the n variables as described above. The automaton A checks whether the assignment coded by v_i satisfies the clause C_i : Assume that A reads letter j of v_i . If this letter is 0 and $\neg x_j$ is contained in C_i or the letter is 1 and x_j is contained in C_i , then A sets a bit indicating that the current clause is satisfied. If it reaches the end of v_i without the bit being set it rejects by moving to a sink state. Otherwise it reads $\#$ and proceeds to v_{i+1} . After having processed k such words, A loops back to the initial state. All states except the sink state are accepting.

If this automaton accepts a periodic word with period of length at most $n + 1$, then it must be of the form $(v\#)^\omega$, where $v \in \{0, 1\}^n$ satisfies all the clauses of φ . This shows that φ is satisfiable iff the constructed automaton accepts a periodic word with period of length at most $n + 1$. \square

This is of course not a proof for the hardness of equivalence for unambiguous Büchi automata. It only shows that the techniques that have been used so far for obtaining polynomial time equivalence tests are unlikely to work for the case of unambiguous Büchi automata.

6 Conclusion

The class of strongly unambiguous Büchi automata is the first known class of Büchi automata as expressive as nondeterministic Büchi automata for which the inclusion and equivalence problem can be decided in polynomial time. However, this class is quite difficult to understand because strongly unambiguous Büchi automata are co-deterministic [6] and we usually think in a deterministic way. In addition, there are deterministic Büchi automata exponentially smaller than strongly unambiguous ones, which is impossible for unambiguous Büchi automata, because every deterministic Büchi automaton is unambiguous. Therefore, it would be interesting to settle the complexity of the equivalence problem for unambiguous Büchi automata.

References

1. Valiant, L.G.: Relative complexity of checking and evaluating. *Inf. Process. Lett.* **5**(1) (1976) 20–23
2. Landweber, L.H.: Decision problems for ω -automata. *Mathematical Systems Theory* **3** (1969) 376–384
3. Arnold, A.: Rational ω -languages are non-ambiguous. *Theoretical Computer Science* **26** (1983) 221–223
4. Niwiński, D., Walukiewicz, I.: Ambiguity problem for automata on infinite trees. (Unpublished note)
5. Carayol, A., Löding, C.: MSO on the infinite binary tree: Choice and order. In: *Proceedings of the 16th Annual Conference of the European Association for Computer Science Logic, CSL 2007*. Volume 4646 of *Lecture Notes in Computer Science.*, Springer (2007) 161–176
6. Carton, O., Michel, M.: Unambiguous Büchi automata. *Theor. Comput. Sci.* **297**(1–3) (2003) 37–81
7. Stockmeyer, L.J.: *The Complexity of Decision Problems in Automata Theory and Logic*. PhD thesis, Dept. of Electrical Engineering, MIT, Boston, Mass. (1974)
8. Aho, A., Hopcroft, J., Ullman, J.: *The Design and Analysis of Computer Algorithms*. Addison-Wesley, New York (1974)
9. Comon, H., Dauchet, M., Gilleron, R., Jacquemard, F., Löding, C., Lugiez, D., Tison, S., Tommasi, M.: *Tree Automata Techniques and Applications*. (Available on <http://tata.gforge.inria.fr/>) Last Release: October 12, 2007.
10. Stearns, R.E., Hunt III, H.B.: On the equivalence and containment problems for unambiguous regular expressions, regular grammars and finite automata. *SIAM Journal on Computing* **14**(3) (1985) 598–611
11. Seidl, H.: Deciding equivalence of finite tree automata. *SIAM J. Comput.* **19**(3) (1990) 424–437
12. Büchi, J.R.: On a decision method in restricted second order arithmetic. In: *International Congress on Logic, Methodology and Philosophy of Science*, Stanford University Press (1962) 1–11
13. Calbrix, H., Nivat, M., Podelski, A.: Ultimately periodic words of rational ω -languages. In: *Mathematical Foundations of Programming Semantics, 9th International Conference, Proceedings, MFPS'93*. Volume 802 of *Lecture Notes in Computer Science.*, Springer (1993) 554–566
14. Hopcroft, J.E., Ullman, J.D.: *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley (1979)
15. Baier, C., Katoen, J.P.: *Principles of Model Checking*. MIT Press (2008)
16. McNaughton, R.: Testing and generating infinite sequences by a finite automaton. *Information and Control* **9**(5) (1966) 521–530
17. Thomas, W.: Languages, automata, and logic. In Rozenberg, G., Salomaa, A., eds.: *Handbook of Formal Language Theory*. Volume III. Springer (1997) 389–455