DETERMINISTIC ω-REGULAR LIVENESS PROPERTIES

Frank Nießner
Computer Science Department, University of Frankfurt
Robert-Mayer-Str. 11-15, D-60325 Frankfurt, Germany
e-mail: niessner@informatik.uni-frankfurt.de

Ulrich Nitsche
Computer Science Department, University of Zürich
Winterthurerstr. 190, CH-8057 Zürich, Switzerland
e-mail: nitsche@ifi.unizh.ch

Peter Ochsenschläger
Institute for Telecooperation Technology
GMD – German National Research Centre for Information Technology
Rheinstr. 75, D-64295 Darmstadt, Germany
e-mail: ochsenschlaeger@darmstadt.gmd.de

Abstract. A major drawback for the use of automated verification techniques is the complexity of verification algorithms in general. One of the sources of the algorithms' complexity is the difference between the language classes accepted by deterministic and nondeterministic Büchi-automata respectively. This difference causes the problem of complementing Büchi-automata and hence deciding subset conditions on regular ω-languages to be PSPACE-complete. We investigate in this paper whether nontrivial property classes exist that can be characterized by deterministic Büchi-automata and hence be complemented rather easily. Since the class of safety properties is known to be representable deterministically, taking into account that safety properties are the closed sets in the Cantor topology, it suffices for us to identify nontrivial deterministic ω-regular liveness properties.

1 Introduction

The behaviour of reactive systems [6], i.e. *non*-terminating systems reacting to some outside actions, can be characterized in a suitable way by regular ω-languages [8]. According to specific demands and constraints of practical relevance, indeed Eilenberg-limits [3] of prefix-closed regular languages suffice for the representation of reactive systems' behaviours. The notion *verification* of reactive systems describes, in this context, the process of checking whether all

ω-words in an Eilenberg-limit of a prefix-closed regular language satisfy particular properties. Formally, this is done by checking whether the behaviour is a subset of an ω-language that represents all correct behaviours that a system, defined with respect to a particular alphabet *of actions*, may perform. In these formal terms, the ω-language for which we check whether a system's behaviour is one of its subsets is simply called a *property*.

Due to the difference between the class of regular ω-languages acceptable by deterministic and nondeterministic Büchi-automata [2], verification is a complex task since the related problem of complementing Büchi-automata is also complex (PSPACE-complete) [8]. Nevertheless, if a property is accepted by a deterministic Büchi-automaton, then complementation is easy (we present the construction of the complement automaton briefly in this paper). The question thus is: Can we identify non-trivial, practically relevant deterministic ω-regular properties? Here, by "practically relevant", we refer to the fact that characterizations of deterministic Büchi-languages are given so far in terms of G_δ sets in the Cantor topology [5] or by ω-regular expressions on regular prefix-codes [3]. These characterizations do not reveal any hints about deterministic acceptability of properties in the usual way properties are represented in practice. We will answer the above question positively in this paper by presenting deterministic ω-regular properties of practical interest.

Before starting with the main topics we have to explain that considering a particular class of properties called *liveness properties* [1, 4] is sufficient. In [4], two property classes are discussed, *safety and liveness properties*, that are formally defined in [1]. For ω-regular safety properties it is easy to show that they are deterministic: they are the closed sets in the Cantor topology. Additionally, all ω-regular properties are the intersection of an ω-regular safety and an ω-regular liveness property. Taking into account that determinism of regular ω-languages is preserved by intersection, we obtain that a characterization of deterministic ω-regular liveness properties is sufficient to characterize all deterministic ω-regular properties. We take in this paper a first step towards a characterization of deterministic ω-regular liveness properties by presenting two practically relevant subclasses and showing how to construct additional deterministic ω-regular liveness properties from known ones. Up to now, we do not know whether this establishes already an exhaustive characterization of deterministic ω-regular liveness propterties.

Identifying deterministic ω-regular liveness properties is not only relevant for verification task concerning the linear satisfaction of properties. It is also of interest with respect to another satisfaction notion for properties, namely *approximate satisfaction* [7]. To check whether a property is approximately satisfied, we have to check subset conditions on regular languages containing finitely long words. Considering properties that are represented by deterministic automata avoids a subset construction that would be necessary, if the properties were repesented by nondeterministic automata. Hence, looking at deterministic ω-regular liveness properties also reduces the complexity of checking their approximate satisfaction.

2 Preliminaries

By $\mathbb{N}$ we designate the set $\{1, 2, 3, \ldots\}$ of natural numbers. $\mathbb{N}_0$ denotes the set $\{0, 1, 2, \ldots\}$ of non-negative integers. For an alphabet Σ, Σ^ω designates the set of ω-words, i.e. infinitely long words, over Σ. Subsets of Σ^ω are called ω-languages over Σ. The set of all words over Σ is denoted by Σ^*.

Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ be a finite automaton and let $x = x_1 x_2 x_3 \ldots \in \Sigma^\omega$ be an ω-word in Σ^ω. A run of $\mathcal{A}$ on x is a sequence $q_1 q_2 q_3 \ldots \in Q^\omega$ of states such that q_1 is the initial state of $\mathcal{A}$, i.e. $q_1 = q_0$, and $\forall i \in \mathbb{N} : (q_i, x_i, q_{i+1}) \in \delta$. For a run $r \subseteq Q^\omega$ of $\mathcal{A}$ on x, $\omega(r)$ denotes the set of all states that repeat infinitely often in r.

Intuitively, a run of $\mathcal{A}$ on $x \in \Sigma^\omega$ is a sequence of states $\mathcal{A}$ may run through when reading x. A run is successful if and only if $\mathcal{A}$ visits at least one accepting state infinitely often. Formally this is defined by Büchi-acceptance [2]:

We call a run r successful if and only if $\omega(r) \cap F \neq \emptyset$. We say that $\mathcal{A}$ Büchi-accepts x if and only if there exists an accepting run of $\mathcal{A}$ on x [2, 8]. ω-Languages that are Büchi-acceptable by some finite automaton are called regular ω-languages [8].

If, for all states q in Q and all letters a in Σ, the successor state of q with respect to a is uniquely determined, then $\mathcal{A}$ is called deterministic. In the infinite case, i.e. for ω-languages, the known equivalence of the language classes accepted by deterministic and nondeterministic automata does not hold. An example of a regular ω-language that demands a nondeterministic Büchi-automaton to accept it is the ω-language that contains all ω-words over $\{a, b\}$ which contain only finitely many occurences of letter a [8].

Let $w \in \Sigma^*$, let $x \in \Sigma^\omega$, let $L \subseteq \Sigma^*$, and let $L_\omega \subseteq \Sigma^\omega$. The set of prefixes of w is $pre(w) = \{w_1 w_2 ... w_i \mid i \in [0, |w|]\}$. For $i = 0$, $w_1 w_2 ... w_i = \varepsilon$. The set of (finite) prefixes of x is $pre(x) = \{x_1 x_2 ... x_i \mid i \in \mathbb{N}_0\}$. The sets of prefixes of L and L_ω respectively are $pre(L) = \bigcup_{w \in L} pre(w)$ and $pre(L_\omega) = \bigcup_{x \in L_\omega} pre(x)$.

Let $V \subseteq \Sigma^*$. Then V^ω designates the set of all ω-words $v_1 v_2 v_3 \ldots$ such that $v_1, v_2, v_3, \ldots \in V$. V is called a prefix-code if and only if no word in V is a proper prefix of another word in V, i.e. if and only if $V = V \setminus (V \cdot \Sigma^+)$ [3] (in [3], prefix-codes are simply called prefixes and what we call a prefix is called an initial segment). We denote the operation $V \setminus (V \cdot \Sigma^+)$ that establishes the prefix-code corresponding to V by $\pi(V)$. We can use the notions of prefix-codes and the ω-power of languages to establish a characterization of deterministic regular ω-languages.

Lemma 1. *$L_\omega \subseteq \Sigma^\omega$ can be accepted by a deterministic Büchi-automaton if and only if there exist regular prefix-codes U_i and V_i, $1 \leq i \leq n$, $n \in \mathbb{N}$, such that $L_\omega = \bigcup_{i=1}^{n} (U_i \cdot V_i^\omega)$ ([3], p. 381, Theorem 1.3.).*

Let $L \subseteq \Sigma^*$. The Eilenberg-limit [3] $lim(L)$ of L is defined as $lim(L) = \{x \in \Sigma^\omega \mid \exists^\infty w \in pre(x) : w \in L\}$. "$\exists^\infty$..." abbreviates "there exist infinitely many different ...". We may say briefly limit instead of Eilenberg-limit.

Lemma 2. *$L_\omega \subseteq \Sigma^\omega$ can be accepted by a deterministic Büchi-automaton if and only if there exists a regular set $L \subseteq \Sigma^*$ such that $L_\omega = lim(L)$ [8].*

Considering a behaviour of a reactive system, it would not be meaningful to allow finite computations to have prefixes that are not finite computations of the system. $L \subseteq \Sigma^*$ is called prefix-closed if and only if $pre(L) = L$. Prefix-closed regular languages are accepted by finite automata with only accepting states.

Because the (infinite) behaviour of a reactive system is its "infinitely continued" finite behaviour, we consider a behaviour to be the Eilenberg-limit of a prefix-closed regular language.

Intuitively, a property partitions the set Σ^ω into the set $Y \subseteq \Sigma^\omega$ of the computations that satisfy the property and the set $N \subseteq \Sigma^\omega$ of computations that do not. To define a property formally, we simply identify the property with the set Y of computations that satisfy it: a property is a subset of Σ^ω. $L_\omega \subseteq \Sigma^\omega$ satisfies the property P if and only if $L_\omega \subseteq P$.

We can characterize properties in regard to their intuitive meaning [1, 4]. There are properties that demand the absence of bad situations; i.e. the absence of obvious errors. This type of properties is referred to as *safety* properties. What can we say about safety properties? The "obvious" error is an error that, if occured, could not be fixed. If such an error occurs in a computation, meaning that the computation does not satisfy the safety property, there must be a point in the sequence of actions where, independent of the further actions, the property cannot be satisfied. In more formal terms: there must be a prefix of the ω-word, whereat, independent of the following suffix, the safety property cannot be satisfied. We give a definition equivalent to the one in [1].

A property $P \subseteq \Sigma^\omega$ is called a safety property if and only if, for all $x \in \Sigma^\omega$, $x \notin P$ implies that there exists a prefix $w \in pre(x)$, such that, for all $y \in \Sigma^\omega$, we have $wy \notin P$.

Other properties of interest are the ones that demand that an intended condition will occur eventually ("something good will happen eventually"). These properties are called *liveness* properties. For a liveness property, the possible satisfaction of the property must be independent of the so far partial computation (i.e. finitely long computation). Hence, for all partial computations, there must exist a computation such that, if the partial computation and the computation are concatenated, the resulting computation will satisfy the property. We present the definition of liveness of [1].

A property $P \subseteq \Sigma^\omega$ is called a liveness property if and only if, for all $w \in \Sigma^*$, there exists an ω-word $x \in \Sigma^\omega$ such that $wx \in P$.

Lemma 3. *$P \subseteq \Sigma^\omega$ is a liveness property if and only if $pre(P) = \Sigma^*$.*

A liveness property is a *progress* property, since it demands progress in direction to meet specific requirements. An interesting result about liveness and safety is that all properties can be represented as the intersection of a liveness and a safety property.

Lemma 4. *Let $P \subseteq \Sigma^\omega$ be a property over Σ. Then P is the intersection of a liveness property $P_l \subseteq \Sigma^\omega$ and a safety property $P_s \subseteq \Sigma^\omega$ [1].*

The proof of this lemma in [1] is based on a characterization of liveness properties as dense sets and the safety properties as closed sets in the Cantor topology over Σ^ω [3]. Taking into account that the closed sets in the Cantor topology are exactly the Eilenberg-limits of prefix-closed language, we obtain in the ω-regular case that the regular safety properties are all acceptable by deterministic Büchi-automata. Since deterministic ω-regular properties are closed under intersection, we have to consider only deterministic ω-regular liveness properties in our search for deterministic ω-regular properties

3 Complementation of Büchi-Automata

Our motivation for looking at deterministic properties is the complexity of checking subset condition for ω-regular languages. Checking $L_\omega \subseteq P$ is performed algorithmically by checking $L_\omega \cap \overline{P}$ for emptyness ($\overline{P} = \Sigma^\omega \setminus P$). Regular ω-languages are closed under boolean operations, but, unfortunately, complementation of Büchi-automata is complex:

Lemma 5. *The problem of complementing Büchi automata is PSPACE-complete [8].*

For deterministic Büchi-automata, complementation can be done easily. We briefly present the construction of a Büchi-automaton $\mathcal{A}' = (Q', \Sigma, \delta', q_0, F')$ representing $\overline{P}$, given a deterministic Büchi-automaton $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ representing P. Notice that, for deterministic $\mathcal{A}$, if a run on $x \in \Sigma^\omega$ is not successful or there is no run on x, then $x \in \overline{L_\omega(\mathcal{A})}$ (a run is uniquely determined for deterministic Büchi-automata). Let, for all $q \in Q$, $q' \notin Q$ and let $f \notin Q$. We set

$$Q' = Q \cup \{q' \mid q \in Q \setminus F\} \cup \{f\}.$$

$$\delta' = \begin{cases} \delta & \cup \\ \{(q, \varepsilon, q') \mid q \in Q \setminus F\} & \cup \\ \{(q', a, p') \mid q, p \in Q \setminus F \wedge (q, a, p) \in \delta\} & \cup \\ \{(q, a, f) \mid q \in Q \wedge a \in \Sigma \wedge \nexists p \in Q : (q, a, p) \in \delta\} & \cup \\ \{(q', a, f) \mid q \in Q \wedge a \in \Sigma \wedge \nexists p \in Q : (q, a, p) \in \delta\} & \cup \\ \{(f, a, f) \mid a \in \Sigma\}. \end{cases}$$

$$F' = \{q' \mid q \in Q \setminus F\} \cup \{f\}.$$

Lemma 6. $L_\omega(\mathcal{A}') = \overline{L_\omega(\mathcal{A})}$.

In $\mathcal{A}'$, we nondeterministically move to a state from which no state is reachable which corresponds to an accepting state in $\mathcal{A}$. Hence $\mathcal{A}'$ accepts exactly all ω-words which are not accepted by $\mathcal{A}$.

4 Some Deterministic Liveness Properties

We establish in this section, based on a result for maximal regular prefix-codes, two classes of practically relevant deterministic ω-regular liveness properties.

Let $L_{pc} \subseteq \Sigma^*$ be a regular prefix-code. L_{pc} is called *maximal* if and only if, for all $w \in \Sigma^* \setminus L_{pc}$, the language $L_{pc} \cup \{w\}$ is not a prefix-code. Notice that, for all regular languages $L \subseteq \Sigma^*$, $\pi(\Sigma^* \cdot L)$ is a maximal regular prefix-code. Not all maximal regular prefix-codes are of the type $\pi(\Sigma^* \cdot L)$ for some regular L; e.g. $\{aa, aba, abb, b\} \subseteq \{a, b\}^*$ is a maximal prefix-code that cannot be represented in such a way. According to [3] (page 92, Proposition 7.1.) we obtain:

Lemma 7. *$L_{pc} \subseteq \Sigma^*$ is a maximal prefix-code if and only if $pre(L_{pc} \cdot \Sigma^*) = \Sigma^*$.*

According to Lemma 3, we have:

Corollary 8. *$L_{pc} \subseteq \Sigma^*$ is a maximal prefix-code if and only if $L_{pc} \cdot \Sigma^\omega$ is a liveness property.*

Combining Lemma 7 and Corollary 8, we obtain:

Corollary 9. *Let $A, B \subseteq \Sigma^*$ be regular prefix-codes. $A \cdot B^\omega$ is a deterministic ω-regular liveness property if and only if A and B are maximal.*

We use this corollary to establish two classes of deterministic ω-regular liveness properties.

4.1 "Eventually a Regular Pattern" is a Deterministic ω-Regular Liveness Property

Let $L \subseteq \Sigma^*$ be a regular language. We show that the ω-language $P_{evt} = \Sigma^* \cdot L \cdot \Sigma^\omega$ is a deterministic ω-regular liveness property. P_{evt} demands a regular pattern in L to occur eventually.

Lemma 10. *Let $L \subseteq \Sigma^*$ be a regular language. Then $P_{evt} = \Sigma^* \cdot L \cdot \Sigma^\omega$ is a deterministic ω-regular liveness property.*

Proof. Because $\Sigma^* \cdot L \cdot \Sigma^\omega = \pi(\Sigma^* \cdot L) \cdot \Sigma^\omega$ and Σ is a maximal regular prefix-code, we obtain by Corollary 9 that P_{evt} is a deterministic ω-regular liveness property.

4.2 "Infinitely Often Some Regular Pattern" is a Deterministic ω-Regular Liveness Property

Let $L \subseteq \Sigma^*$ be a regular language. We show that the ω-language $P_{inf} = (\Sigma^* \cdot L)^\omega$ is a deterministic ω-regular liveness property. P_{inf} demands regular patterns in L to occur infinitely often.

Lemma 11. *Let $L \subseteq \Sigma^*$ be a regular language. Then $P_{inf} = (\Sigma^* \cdot L)^\omega$ is a deterministic ω-regular liveness property.*

Proof. Because $(\Sigma^* \cdot L)^\omega = \pi((\Sigma^* \cdot L))^\omega = \pi((\Sigma^* \cdot L)) \cdot \pi((\Sigma^* \cdot L))^\omega$, we obtain by Corollary 9 that P_{inf} is a deterministic ω-regular liveness property.

Lemma 10 and Lemma 11 contain inherent subset constructions on the automaton level that, in the worst case, lead to exponential blow-ups in the number of states of the automata representing the properties when compared to the automaton accepting L. For Lemma 11, we present the subset construction in the appendix of this paper.

The subset-construction can be avoided, if for all $a \in \Sigma$, $u \in \Sigma^*$, and $v \in pre(L)$, we have: if $va \in pre(L)$ and $au \in L$, then $vau \in L$. This condition briefly says that, if an initial letter is repeated somewhere later, then the suffix-language starting there is a superset of the sublanguage of L of all words in L which start with this letter. In other words: we cannot miss to recognize another word in L when we try to recognize one that "started earlier" in the ω-word that we try to accept.

An easy subcondition of the one presented above is requiring that $L \subseteq (\Sigma_1 \cdot \Sigma_2^*)$ for Σ_1 and Σ_2 partitioning Σ. Here, letters in Σ_1 indicate at which positions in a given ω-word we have to try to recognize a subword that may be in L.

5 Construction of Deterministic ω-Regular Liveness Properties from Other Deterministic ω-Regular Liveness Properties

We present, in this section, a construction that allows us to construct a deterministic ω-regular liveness property from other deterministic ω-regular liveness properties. Our construction is related to a representation of deterministic regular ω-languages by ω-regular expressions on regular prefix-codes given in [3].

So far, we considered deterministic ω-regular liveness properties where the regular pattern that had to occur was independent of the prefix up to this point. We can also imagine liveness properties where the ω-word we have to concatenate to a finitely long word to obtain an ω-word in the property depends on the finitely long word itself. An example with respect to the alphabet $\Sigma = \{a, b\}$ is the liveness property $a \cdot \Sigma^* \cdot a \cdot \Sigma^\omega + b \cdot \Sigma^* \cdot b \cdot \Sigma^\omega$ which demands that the letter a will eventually occur, if the ω-word starts with letter a, as well as the corresponding condition for letter b.

Let $n \in I\!N$. Let $L_{\omega,i} \subseteq \Sigma^\omega$, $1 \leq i \leq n$, be deterministic ω-regular liveness properties. Let $L_i \subseteq \Sigma^*$, $1 \leq i \leq n$, be regular languages such that the following conditions hold:

- All L_i are pairwise disjoint prefix-codes.
- $\bigcup_{i=1}^{n} L_i$ is a prefix-code.
- $pre((\bigcup_{i=1}^{n} L_i) \cdot \Sigma^*) = \Sigma^*$.

The second and third condition simply require $\bigcup_{i=1}^{n} L_i$ to be a maximal prefix-code.

Lemma 12. *$\bigcup_{i=1}^{n} (L_i \cdot L_{\omega,i})$ is a deterministic ω-regular liveness property.*

Proof. Because we have $pre((\bigcup_{i=1}^{n} L_i) \cdot \Sigma^*) = \Sigma^*$ and $pre(L_{\omega,i}) = \Sigma^*$, we obtain that $pre(\bigcup_{i=1}^{n} (L_i \cdot L_{\omega,i})) = \Sigma^*$. Thus $\bigcup_{i=1}^{n} (L_i \cdot L_{\omega,i})$ is a liveness property.

Because all $L_{\omega,i}$ are deterministic regular ω-languages and all L_i are regular prefix-codes, all $L_i \cdot L_{\omega,i}$ are deterministic regular ω-languages.

Because $\bigcup_{i=1}^{n} L_i$ is a prefix-code, all ω-words in $\bigcup_{i=1}^{n} (L_i \cdot L_{\omega,i})$ can be uniquely related to exactly one $L_i \cdot L_{\omega,i}$ by checking which L_i contains a prefix of this ω-word (there exists exactly one L_i containig such a prefix). Thus we finally obtain $\bigcup_{i=1}^{n} (L_i \cdot L_{\omega,i})$ is a deterministic regular ω-language.

Taking into account that Σ^ω is a deterministic liveness property, for all finite alphabets Σ, we obtain that Lemma 10 is a special case of the application of Lemma 12. Most of the conditions the L_i have to satisfy can be checked easily on a deterministic automaton representing L_i:

- The condition that all L_i are prefix-codes can be checked by testing whether there are no outgoing transitions from accepting states in the automaton.
- To check whether $\bigcup_{i=1}^{n} L_i$ is a prefix-code we can construct a deterministic automaton accepting $\bigcup_{i=1}^{n} L_i$ and then test whether there are no outgoing transitions from accepting states.
- Checking whether $pre((\bigcup_{i=1}^{n} L_i) \cdot \Sigma^*) = \Sigma^*$ can be done by verifying whether the automaton is complete with respect to non-accepting states, i.e. for all non-accepting states and all letters, there exists a transition from the state labelled by the letter.

The second condition we have to check for the automata accepting the L_i is not the easiest to handle. Fortunately we can show that this condition is not needed. First notice that adding ω-words to a liveness property results in a liveness property (if $pre(P) = \Sigma^*$, then $pre(P \cup L_\omega) = \Sigma^*$, for all L_ω) and that deterministic ω-regular languages are closed under set union [3]. Hence we have:

Lemma 13. *Let $P \subseteq \Sigma^\omega$ be a deterministic ω-regular liveness property and let $L_\omega \subseteq \Sigma^\omega$ be a deterministic ω-regular language. Then $P \cup L_\omega$ is a deterministic ω-regular liveness property.*

By this lemma, we obtain that if the second condition in Lemma 12 is not satisfied, then we have added some deterministic regular ω-language to some deterministic liveness property and thus still obtain a deterministic liveness property. Therefore, we can omit the second condition for Lemma 12 and have:

Let $n \in I\!N$. Let $L_{\omega,i} \subseteq \Sigma^\omega$, $1 \leq i \leq n$, be deterministic ω-regular liveness properties. Let $L_i \subseteq \Sigma^*$, $1 \leq i \leq n$, be regular languages such that the following conditions hold:

- All L_i are pairwise disjoint prefix-codes.
- $pre((\bigcup_{i=1}^{n} L_i) \cdot \Sigma^*) = \Sigma^*$.

Corollary 14. $\bigcup_{i=1}^{n} (L_i \cdot L_{\omega,i})$ *is a deterministic ω-regular liveness property.*

Notice the idempotency of Corollary 14: if $\mathcal{P}$ is a class of deterministic ω-regular liveness properties established by the application of Corollary 14 to some other class of deterministic ω-regular liveness properties, i.e. the $L_{\omega,i}$ were taken from this second class, then the application of Corollary 14 to the class $\mathcal{P}$ results again in the class $\mathcal{P}$. "Corollary 14 only needs to be applied once."

6 Conclusions

We have shown that the properties "eventually some regular pattern" and "infinitely often some regular pattern" are deterministic ω-regular liveness properties.

Deterministic ω-regular liveness properties are worth considering because verification algorithms require complementation of the property to check subset conditions, and complementation of deterministic Büchi-automata, though being complex for general Büchi-automata, is easy.

One may, at first glance, ask: why not also consider $\Sigma^* \cdot (L \cdot \Sigma^*)^n \cdot \Sigma^\omega$ such that n is a natural number, Kleene star '$*$', or Kleene plus '$+$'? Since $(L \cdot \Sigma^*)^n$ is itself a regular language, this is already included in Lemma 10. Hence the class of deterministic ω-regular liveness properties characterized by the properties given in Lemma 10 and Lemma 11 is quite rich. In combination with Corollary 14, we obtain a reasonably large class of nontrivial deterministic ω-regular liveness properties.

The remaining question is: what have we missed? A topic for further study is thus the quest for deterministic ω-regular liveness properties that are not characterized by the lemmas we have presented. Alternatively, one may try to show that the lemmas already give a complete characterization of deterministic ω-regular liveness properties. Obviously, it is our final aim to establish a complete characterization of deterministic ω-regular liveness properties in terms of regular sets and regular ω-languages, or in other words: a characterization of deterministic ω-regular liveness properties that is practically more relevant than

the characterizations by the G_δ-sets in the Cantor-topology [5] and by ω-regular expressions on prefix-codes [3] respectively, additionally requiring liveness.

Besides the fact that checking ω-regular liveness properties linearly becomes easy in the deterministic case, one may also ask whether approximate satisfaction [7] of deterministic ω-regular liveness properties becomes easier than in the general case. Here one has to check a condition $pre(L_\omega) \subseteq pre(L_\omega \cap P)$. Thus one does not have to complement Büchi-automata but finite automata accepting finite-word languages. Hence, when considering deterministically represented ω-regular P, we can avoid making a subset construction for checking $pre(L_\omega) \subseteq pre(L_\omega \cap P)$. Even though, the subset construction is "nicer" than the construction necessary to complement Büchi-automata [3, 8], it is an important improvement to avoid it when checking approximate satisfaction of deterministic ω-regular properties. Therefore in both cases, the linear and the approximate case, considering deterministic ω-regular properties which are studied in this paper, drastically reduces the complexity of algorithms for checking the satisfaction of the properties.

Acknowledgments

We would like to thank Detlef Wotschke very much for interesting discussions and his support.

Appendix: Subset-Construction for Lemma 11

Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ be the minimal deterministic finite automaton accepting L; i.e. $L(\mathcal{A}) = L$. We construct a deterministic finite automaton $\mathcal{A}' = (Q', \Sigma, \delta', q_0', F')$ that Büchi-accepts P_{inf}. Let $f \notin Q$.

We set $Q' = (Q \times 2^Q) \cup \{f\}$, $q_0' = (q_0, \emptyset)$, and $F' = \{f\}$. We will check "nondeterministically" whether a word in L occurs and will resolve the "nondeterminism" making a subset construction on the state-set. The only final state f will be reached when a pattern in L is detected. Hence an ω-word will be Büchi-accepted only if words in L appear infinitely often.

We simulate $\mathcal{A}$ on an ω-word in the first component of the new states in Q'. If $\mathcal{A}$ accepted, we would reach the final state f, treating it as a new initial state for the simulation of $\mathcal{A}$. In other words: we check whether $\mathcal{A}$ would accept, again and again, finitely long subwords of the ω-word. If in the simulation of $\mathcal{A}$ on an ω-word, $\mathcal{A}$ could not take any transition, we restart the simulation. Since during a not successful simulation of $\mathcal{A}$ on the ω-word, the beginning of a correct pattern (i.e. word in L) could have occured, we keep track on "simulations that started later than the current one" in the second component of the current state in Q', containing all possibly alternative states which a later simulation of $\mathcal{A}$ on the ω-word would have reached. We could have put all state information into a single subset of states in Q, but splitting it into a (state, set of alternative states)-pair directly proves $\mathcal{A}'$ to be deterministic. According to our comments so far, we start with the definition of δ'.

Let $q \in Q$, $O \subseteq Q$, and $a \in \Sigma$. Let $P = \delta(O, a)$. If $\delta(q, a)$ is defined then let $q' = \delta(q, a)$. Otherwise let $q' = q_0$. If $(\{q'\} \cup P) \cap F \neq \emptyset$, we set $\delta'((q, O), a) = f$. Otherwise we set $\delta'((q, O), a) = (q', (P \cup \{q_0\}) \setminus \{q'\})$. For all $\tilde{q} \in Q'$ and $a \in \Sigma$, we set $\delta'(f, a) = \tilde{q}$ if and only if $\delta'((q_0, \emptyset), a) = \tilde{q}$.

$\mathcal{A}'$ is deterministic and, by construction, accepts P_{inf}.

References

1. B. Alpern and F. B. Schneider. Defining liveness. *Information Processing Letters*, 21(4):181–185, October 1985.
2. J. Büchi. On a decision method in restricted second order arithmetic. In E. Nagel et al., editors, *Proceedings of the International Congress on Logic, Methodology and Philosophy of Science 1960*, pages 1–11. Stanford University Press, 1962.
3. S. Eilenberg. *Automata, Languages and Machines*, volume A. Academic Press, New York, 1974.
4. L. Lamport. Proving the correctness of multiprocess programs. *IEEE Transactions on Software Engineering*, SE-3(2):125–143, 1977.
5. Z. Manna and A. Pnueli. A hierarchy of temporal properties. In *Proceedings of the 9th Annual ACM Symposium on Principles of Distributed Computing*, pages 377–408. ACM Press, 1990.
6. Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems—Specification*. Springer Verlag, New York, first edition, 1992.
7. U. Nitsche and P. Ochsenschläger. Approximately satisfied properties of systems and simple language homomorphisms. *Information Processing Letters*, 60:201–206, 1996.
8. W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Formal Models and Semantics*, volume B of *Handbook of Theoretical Computer Science*, pages 133–191. Elsevier, 1990.

This article was processed using the LaTeX macro package with LLNCS style