Specification of Deterministic Regular Liveness Properties

Frank Nießner*

telecommunications, networks & *security* Research Group
Department of Computer Science, University of Fribourg
Boulevard de Pérolles 90, CH–1700 Fribourg, Switzerland

Abstract. Even up-to-date automated verfication techniques are affected by the fundamental complexity of verification algorithms which is caused by necessity to decide subset conditions on certain languages. These languages are recognizable by nondeterministic Büchi automata and represent a system behavior and the desired properties of the system. The involved complementation process may lead to an exponential blow-up in the size of the automata. In this paper we specify the structure of a rich subclass of languages that can be characterized by deterministic Büchi automata and hence be complemented rather easily. Furthermore, we present examples of practically relevant properties belonging to this language class.

1 Introduction

The behavior of systems that exhibit temporary perpetual behavior and have the ability to react to their environment [7] can be appropriately described by regular ω-languages [10] which are Eilenberg-limits [3] of prefix-closed regular languages. Here, the fundamental alphabet is the set of actions that may be performed by the considered system and the system behavior is the set of all infinite action-sequences the system may perform. In this context, verification describes the process of checking whether the behavior is a subset of an ω-language that contains all the action-sequences representing the correct behavior of the system. We call this latter language a property.

Properties are as well be characterized by regular ω-languages and Büchi automata respectively. Due to the difference between the language-classes which are recognizable by deterministic and nondeterministic Büchi automata (the deterministic and nondeterministic regular ω-languages) [2], verification becomes a different task since it might be essential to 'complement' the, in general, nondeterministic property-automaton. This can cause an automaton that is exponentially larger. However, we are able to compute the complement of a property-automaton rather easily if it is deterministic [8].

Therefore, we investigate deterministic automata and deterministic properties respectively. Even though we consider just a proper subset of all regular ω-languages, this is no major drawback since deterministic properties contain a large class of practically

* The author is supported by the *Hasler Foundation* under grant number 1922.

Nießner F. (2006).
Specification of Deterministic Regular Liveness Properties.
In *Proceedings of the 4th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems*, pages 173-178
DOI: 10.5220/0002482601730178

relevant properties in general [8]. Furthermore, before we begin our considerations, we will briefly explain that it is sufficient to focus on the particular class of properties called liveness properties. Properties can be separated due to their intuitive meaning into safety and liveness properties [1]. It is easy to show that safety properties are closed sets in the Cantor topology and therefore are always deterministic [6]. Additionally, every property can be represented by an intersection of a safety and a liveness property [1]. Taking into account that determinism of regular ω-languages is preserved under intersection, we obtain that a characterization of deterministic regular liveness properties suffices to characterize all deterministic regular properties.

In this paper we will present the structure of deterministic regular liveness properties in terms of regular prefix-free languages and, additionally, we will give some examples of practically relevant subsets of this class.

2 Preliminaries

We assume the reader is familiar with the common notions of formal language and automata theory as presented in [4]. For a finite set of *actions* Σ, let Σ^* be the set of all finitely long sequences on Σ, let Σ^ω be the set of all infinitely long sequences, and let $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$. A set $L \subseteq \Sigma^*$ is called a (finitary) language, a set $L_\omega \subseteq \Sigma^\omega$ is called an *ω-language*.

Let $L_\infty \subseteq \Sigma^\infty$. Then $pre(L_\infty) = \{v \in \Sigma^* \mid \exists w \in \Sigma^\infty : vw \in L_\infty\}$ denotes the set of all prefixes of L_∞. We call a language $L \subseteq \Sigma^*$ *prefix-closed* if and only if $pre(L) = L$. Further, the *Eilenberg-limit* [3] (or limit for short) of a language $L \subseteq \Sigma^*$ is given by $lim(L) = \{w \in \Sigma^\omega \mid \exists^\infty v \in pre(w) : v \in L\}$.[1] Let $w = \sigma_1\sigma_2 \ldots \in \Sigma^\omega$. Then we define $Inf(w) = \{\sigma \in \Sigma \mid \exists^\infty i : \sigma_i = \sigma\}$.

A finite state automaton is capable of accepting strings. It is given by a quintuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, where Q is a non-empty finite set of states, Σ is a non-empty finite set of input symbols, $q_0 \in Q$ is the initial state, $F \subseteq Q$ is a set of final states, and $\delta : Q \times \Sigma \to 2^Q$ denotes the transition function. We assume the transition function δ to be extended to $2^Q \times \Sigma^* \to 2^Q$ as usual. $\mathcal{A}$ is *deterministic* if $|\delta(q, \sigma)| \leq 1$, for each $q \in Q$, $\sigma \in \Sigma$. A tripel $(q, \sigma, p) \in Q \times \Sigma \times Q$ s.t. $\delta(q, \sigma) \ni p$ is a transition of $\mathcal{A}$.

Let $v = \sigma_1\sigma_2 \ldots \sigma_n \in \Sigma^*$ and $w = \sigma_1\sigma_2 \ldots \in \Sigma^\omega$. A finite state sequence $\rho(v) = r_0r_1 \ldots r_n \in Q^*$ denotes a *finite run* of $\mathcal{A}$ on v if $\delta(r_i, \sigma_{i+1}) \ni r_{i+1}$ for $0 \leq i < n$. The finite run $\rho(v)$ is successful if $r_0 = q_0$ and $r_n \in F$. An infinite state sequence $\rho(w) = r_0r_1 \ldots \in Q^\omega$ denotes a *run* of $\mathcal{A}$ on w if $\delta(r_i, \sigma_{i+1}) \ni r_{i+1}$ for $0 \leq i$. The run $\rho(w)$ is successful if $r_0 = q_0$ and $Inf(\rho(w)) \cap F \neq \emptyset$.

Subject to its acceptance condition, $\mathcal{A}$ turns into a finite automaton or a Büchi automaton. If we define $\mathcal{A}$ to accept all $v \in \Sigma^*$ such that $\delta(q_0, v) \cap F \neq \emptyset$, then $\mathcal{A}$ is a *finite automaton (FA)* and $L(\mathcal{A}) = \{v \in \Sigma^* \mid \text{ There is a successful finite run of } \mathcal{A} \text{ on } v\}$ is a *regular language*. If we define $\mathcal{A}$ to accept each $w \in \Sigma^\omega$ such that there are infinitely many different $v \in pre(w)$ such that $\delta(q_0, v) \cap F \neq \emptyset$, then $\mathcal{A}$ turns into a *Büchi automaton (BA)* and $L_\omega(\mathcal{A}) = \{w \in \Sigma^\omega \mid \text{ There is a successful run of } \mathcal{A} \text{ on } w\}$ is a *regular ω-language*. Throughout this paper we assume our automata to be reduced, i.e., they don't have useless states or transitions.

[1] Read '$\exists^\infty$... : ...' as 'there exist infinitely many different ... such that ...'.

3 System Behavior, Properties and Verification

When we consider the behavior of a reactive system, it would make no sense to allow for finite computations having prefixes that are not a finite behavior as well. Thus, the language representing the finite behavior of the system is prefix-closed and can be recognized by a finite automaton with only accepting states. Furthermore, since it is reasonable to consider the behavior to be the 'infinitely continued' finite behavior of the system, we consider the behavior to be the limit of a prefix-closed language.

Intuitively, a property partitions Σ^ω into the set $Y \subseteq \Sigma^\omega$ of sequences that satisfy the property and the set $N \subseteq \Sigma^\omega$ of sequences that do not. For a formal definition of a property we simply identify it with the set $Y \subseteq \Sigma^\omega$ that satisfies it. A system behavior B satisfies a property $P \subseteq \Sigma^\omega$ linearly if and only if $B \subseteq P$.

Properties can be classified by their intuitive meaning [5, 1]. There are properties demanding that 'nothing undesired will happen'. We call these properties safety properties. If an undesired action occurs in a computation then this computation, independently of further actions, does not satisfy the property. Thus, a property $P \subseteq \Sigma^\omega$ is called a safety property if and only if from $w \notin P$ follows the existence of a $u \in pre(w)$, such that $uv \notin P$ for all $w, v \in \Sigma^\omega$ [1]. Another important class of properties are the liveness properties. These properties demand that 'a desired action or action sequence occurs eventually' but without specifying the point in time and the number of occurences (once or repeatedly). Furthermore, the possible satisfaction of the property must be independent of the computation performed so far. This means that for all finite computations $v \in \Sigma^*$ there must exist an infinite continuation $w \in \Sigma^\omega$ such that $vw \in P$. A reformulation yields [1]: $P \subseteq \Sigma^\omega$ is a liveness property if and only if $pre(P) = \Sigma^*$. The classification of properties into safety and liveness properties is well-founded since a common result from topology states that every property is the intersection of a safety and a liveness property [1].

Verifying a system means deciding subset conditions of the form $B \subseteq P$ which can be algorithmically performed by checking $B \cap \overline{P} = \emptyset$ (where $\overline{L}_\omega = \Sigma^\omega \setminus L_\omega$). However, the problem of complementing Büchi automata is PSPACE-complete [10] and may result in an automaton of size up to $2^{O(n \log n)}$ [9]. An exponential blow-up can be avoided if we restrict the properties to be deterministic. Then, a deterministic Büchi automaton suffices to describe a property P and it can be complemented in linear time, yielding a Büchi automaton (which is not necessarily deterministic anymore) that recognizes $\overline{P}$ and has at most twice as many states (plus one in addition) compared to the original one [8].

4 Deterministic Regular Liveness Properties

We will now have a closer look on the structure of deterministic liveness properties. A good point of origin is the well-known specification for deterministic regular ω-languages which is based on regular prefix-free languages [3]. Here, a language $L \subseteq \Sigma^*$ is called prefix-free if no proper prefix of a string in L is in L. It is called maximal prefix-free if it is prefix-free and for all $w \in \Sigma^* \setminus L$ holds: $L \cup \{w\}$ is not prefix-free anymore. In this case we have $pre(L \cdot \Sigma^*) = \Sigma^*$. The specification is given by

Lemma 1. *A regular ω-language $L_\omega \subseteq \Sigma^\omega$ is deterministic if and only if there exist regular prefix-free languages $U_i, V_i \subseteq \Sigma^*$, $1 \leq i \leq n$ such that $L_\omega = \bigcup\limits_{1\leq i\leq n} U_i \cdot V_i^\omega$.*

A proof can be found in [3]. Instead, we give an intuitive explanation that provides hints on how to improve this characterization towards a specification for deterministic liveness properties.

Obviously, to each deterministic regular ω-language there exists a deterministic Büchi automaton $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ that recognizes it. Hence, this automaton has at least one accepting state, say 1, and, since it accepts infinite action sequences, there must be at least one loop, i.e., a sequence of transitions that starts and ends in this accepting state without passing it intermediately. In general, there might be infinitely many of such sequences. We collect all these sequences in a set, say $V_1 \subseteq \Sigma^*$. In fact, V_1 is a regular language and it is prefix-free, since $\mathcal{A}$ is deterministic. Furthermore, there must exist at least one finite sequence of transitions that start in q_0 and ends up in 1. Again, there might be more than just one such sequence (even infinitely many) and we collect all of them in a set $U_1 \subseteq \Sigma^*$. Notice that U_1 is as well regular und prefix-free since $\mathcal{A}$ is deterministic. Then $U_1 \cdot V_1^\omega$ contains all infinite sequences that $\mathcal{A}$ accepts in state 1. These observations hold for each accepting state from $F = \{1, \ldots, n\}$ of $\mathcal{A}$. Thus, $L(\mathcal{A}) = \bigcup_{1\leq i\leq n} U_i \cdot V_i^\omega$, where $U_i, V_i \subseteq \Sigma^*$ are regular prefix-free languages.

In a certain sense, the determinism is captured in the prefix-freedom of the regular languages $U_i, V_i \subseteq \Sigma^*$ (or vice versa). Due to their determinism, the representation given in Lemma 1 holds as well for deterministic regular liveness properties. However, there must exist additional constraints on $U_i, V_i \subseteq \Sigma^*$ that capture the 'liveness' of a property. Recall that P is a liveness property if and only if $pre(P) = \Sigma^*$. Furthermore, notice that we can add an arbitrary deterministic regular ω-language to P (deterministic regular ω-languages are closed under union) and still have a deterministic regular liveness property.

Now, let $U_i, V_i \subseteq \Sigma^*$ be regular prefix-free languages such that $P = \bigcup_{1\leq i\leq n} U_i \cdot V_i^\omega$. Hence, we must have $pre(\bigcup_{\mathit{1}\leq i\leq n} U_i \cdot V_i^\omega) = \Sigma^*$. Some of the U_j might be prefixes of a U_i, $i \neq j$, and if this is the case, then we can skip them without destroying the 'liveness' of the rest, i.e., $pre(\bigcup_{s\in S} U_s \cdot V_s^\omega) = \Sigma^*$, where $S = \{1, \ldots, n\} \setminus R$ and R is the set of indices j such that U_j is a prefix of a U_i, $i \neq j$. This implies $pre(\bigcup_{s\in S} U_s) \cdot \Sigma^* = \Sigma^*$ which means that $pre(\bigcup_{s\in S} U_s)$ is maximal prefix-free. Moreover, we must have $pre(V_s^\omega) = \Sigma^*$ for $s \in S$. From $pre(V_s^\omega) = \Sigma^*$ follows $pre(V_s \cdot \Sigma^*) = \Sigma^*$, i.e., the regular languages V_s, $s \in S$, must be maximal prefix-free, whereas the V_r, $r \in R$ are at least prefix-free. The deterministic regular ω-language $\bigcup_{r\in R} U_r \cdot V_r^\omega$ can be considered as addition to that part that still describes a deterministic regular liveness property. Thus we obtain the following lemma

Lemma 2. *L_ω is a deterministic regular liveness property if and only if there exist regular prefix-free languages $U_i, V_i \subseteq \Sigma^*$, $1 \leq i \leq n$, such that $L_\omega = \bigcup_{1\leq i\leq n} U_i \cdot V_i^\omega$ and there exists a subset $L_K = \bigcup_{s\in S} U_s \cdot V_s^\omega$ of L_ω, where $S \subseteq \{1, \ldots, n\}$, such that $\bigcup_{s\in S} U_s$ and the V_s are maximal prefix-free for all $s \in S$.*

We exemplify the concept using the Büchi automaton $\mathcal{A}$ depicted in Figure 1.

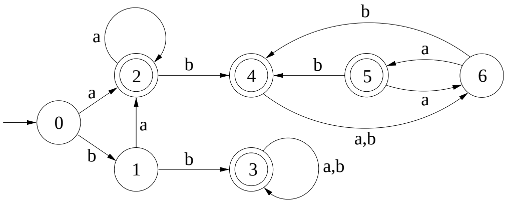

Fig. 1. Büchi automaton $\mathcal{A}$.

Example 1. We have four accepting states (the double-circled ones). The prefix-free sets corresponding to state 2 are $U_1 = (a + ba)$ and $V_1 = a$. Thus, $(a + ba) \cdot a^\omega$ is the set of ω-words that $\mathcal{A}$ recognizes in state 2. The sets corresponding to state 3 are $U_2 = bb$, $V_2 = (a + b)$ and $\mathcal{A}$ recognizes $bb \cdot (a + b)^\omega$ in state 3. For state 4 we obtain $U_3 = (a + ba)a^*b$, $V_3 = (a + b)a^*b$ and the ω-language $a + ba)a^*b \cdot ((a + b)a^*b)^\omega$. The sets corresponding to state 5 are $U_4 = (a+ba)a^*b(a+b)a$, $V_4 = (ab+b)(a+b)a$ and thus the ω-language accepted in state 5 is $U_4 = (a + ba)a^*b(a + b)a \cdot ((ab + b)(a + b)a)^\omega$. We observe that $U_1, U_2 \subseteq pre(U_4)$. Hence, $L(\mathcal{A}) = L_K \cup L_Z$ where $L_K = (bb \cdot (a + b)^\omega) \cup ((a + ba)a^*b(a + b)a \cdot ((ab + b)(a + b)a)^\omega)$ is a deterministic regular liveness property and $L_Z = ((a + ba) \cdot a^\omega) \cup (a + ba)a^*b \cdot ((a + b)a^*b)^\omega)$ is a deterministic regular ω-language.

5 Practically Relevant Deterministic Regular Liveness Properties

This section introduces two classes of practically relevant deterministic regular liveness properties and indicates how they can be extend using Lemma 2. Some of the considerations can as well be found in [8]. From the previous observations follows that the determinism is captured in the prefix-freedom of the involved regular languages. A language L is prefix-free if and only if no word in L is a proper prefix of another word in L or, in other words, if and only if $L \setminus (L \cdot \Sigma^+)$ [3]. We denote the operation $L \setminus (L \cdot \Sigma^+)$ that establishes the prefix-free language corresponding to L by $\pi(L)$. Observe that for all regular languages $L \subseteq \Sigma^*$, $\pi(\Sigma^* \cdot L)$ is a maximal regular prefix-free language. Furthermore, we obtain from Lemma 2 as a special case the following corollary.

Corollary 1. *Let $U, V \subseteq \Sigma^*$ be regular prefix-free languages. Then $U \cdot V^\omega$ is a deterministic regular liveness property if and only if U and V are maximal [8].*

Using this result we establish the following classes of deterministic regular liveness properties. Let $L \subseteq \Sigma^*$ be a regular language. Then the ω-language $P_{evt} = \Sigma^* \cdot L \cdot \Sigma^\omega$ demands a regular pattern in L to occur eventually. P_{evt} is a deterministic regular liveness property since $\Sigma^* \cdot L \cdot \Sigma^\omega = \pi(\Sigma^* \cdot L) \cdot \Sigma^\omega$ and Σ is a maximal regular prefix-free language [8]. Thus, by Corollary 1, the assertion follows. In a similar way we can discuss $P_{inf} = (\Sigma^* \cdot L)^\omega$. P_{inf} demands regular patterns in L to occur infinitely often. Since $(\Sigma^* \cdot L)^\omega = (\pi(\Sigma^* \cdot L))^\omega = \pi(\Sigma^* \cdot L) \cdot (\pi(\Sigma^* \cdot L))^\omega$, we obtain by Corollary 1 that P_{inf} is a deterministic regular liveness property.

Notice that P_{evt} and P_{inf} represent huge classes of properties since there are no restrictions on L (except for regularity). For instance, we could replace L by $(L \cdot \Sigma^*)^n$, where n is a natural number, Kleene star'*' or Kleene plus'+'. All these expressions describe regular languages and thus P_{evt} and P_{inf} would remain deterministic regular liveness properties.

Furthermore, Lemma 2 allows to unify any (finite) number of such deterministic regular liveness properties and we still obtain a deterministic regular liveness property. And there is yet another method to extend these property classes. In the representation given in Lemma 2, we demand the deterministic regular languages V_s to be maximal prefix-free so as to ensure that $pre(V_s^\omega) = \Sigma^*$. However, this holds as well for any deterministic regular liveness property. Thus, in the representation given in Lemma 2 we can replace arbitrarily many V_s^ω by P_{evt}, P_{inf} or any deterministic regular liveness property and the result will be a deterministic regular liveness property. Notice the idempotency of this statement: the resulting deterministic regular liveness property can again replace one (or arbitrarily many) of the maximal prefix-free sets V_s. Hence, the class of deterministic regular liveness properties is rather comprehensive and contains various practically relevant properties.

6 Conclusion

We have considered in detail the deterministic regular liveness properties, since they form a subclass of regular liveness properties for which an exponential blow-up in the number of states can be avoided in the corresponding verification process. We presented a specification for these languages and demonstrated the richness of this language class.

References

1. B. Alpern and F.B. Schneider, Defining liveness, Information Processing Letters, 21(4), pp 181-185, 1985.
2. J.R. 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, pp 1-11. Stanford University Press, 1962.
3. S. Eilenberg, Automata, Languages and Machines, volume A, Academic Press, New York, 1974.
4. J. E. Hopcroft and J. D. Ullman, Introduction to Automata Theory, Languages and Computation, Addison-Wesley Publishing Company, 1979.
5. L. Lamport, Proving the correctness of multiprocess programs, IEEE Transactions on Software Engineering, SE-3(2), pp 125-143, 1977.
6. Z. Manna and A. Pnueli, A hierarchy of temporal properties, Proceedings of the 9th Annual ACM Symposium on Principles of Distributed Computing, ACM Press, pp 377-408, 1990.
7. Z. Manna and A. Pnueli, The Temporal Logic of Reactive and Concurrent Systems-Specification, Springer Verlag, New York, first edition, 1992.
8. F. Nießner, U. Nitsche, and P. Ochsenschläger, Deterministic ω-Regular Liveness Properties, In Symeon Bozapalidis, editor, Proceedings of the 3rd International Conference on Developments in Language Theory (DLT'97), pp 237-247, 1997.
9. S. Safra, On the complexity of ω-automata, Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science, IEEE, pp 319-327, 1988.
10. W. Thomas, Automata on infinite objects, in J. van Leeuwen, editor, Formal Models and Semantics, volume B of Handbook of Theoretical Computer Science, pp 133-191, Elsevier, 1990.