Safety and Liveness Properties for Real Traces and a Direct Translation from LTL to Monoids

Volker Diekert[1] and Paul Gastin[2⋆]

[1] Inst. für Informatik, Universität Stuttgart, Breitwiesenstr. 20-22, D-70565 Stuttgart
diekert@informatik.uni-stuttgart.de
[2] LIAFA, Université Paris 7, 2, place Jussieu, F-75251 Paris Cedex 05
Paul.Gastin@liafa.jussieu.fr

Abstract. For infinite words there are well-known characterizations of safety and liveness properties. We extend these results to real Mazurkiewicz traces. This is possible due to a result, which has been established recently: Every first-order definable real trace language is definable in linear temporal logic using future tense operators, only. We show that the canonical choice for a topological characterization of safety and liveness properties is given by the Scott topology. In this paper we use an algebraic approach where we work with aperiodic monoids. Therefore we also give a direct translation from temporal logic to aperiodic monoids which is of independent interest.
Keywords Mazurkiewicz traces, temporal logics, safety and liveness properties, concurrency, aperiodic monoids.

1 Introduction

Trace theory has been initiated in computer science by Antoni Mazurkiewicz [21] and early works of Robert Keller [19]. Since then many researchers have made important contributions to trace theory, in particular let us mention the work of Grzegorz Rozenberg, see e.g. [1, 11, 12, 26, 27]. Nowadays, it is one of the most popular settings to study concurrency. The richness of trace theory led to the monograph [8], which contains more than 250 bibliographical entries related to trace theory. One of the basic ideas is that the behavior of a concurrent process is not represented by a string, but more accurately by some labeled partial order, which, in the setting of trace theory, means by its dependence graph. We refer to [18] for a systematic study about dependence graphs and to [14] for a systematic study about real traces.

A suitable way for formal specifications of concurrent systems is given by temporal logic formulae which in turn have a direct interpretation for dependence graphs. It is therefore no surprise that temporal logics for traces have received quite an attention, see [4, 7, 22–25, 28–32].

In [30] it was shown that the basic linear temporal logic with future tense operators and only past tense constants is expressively complete with respect

⋆ Support of ADVANCE, CEFIPRA, and PROCOPE is gratefully acknowledged.

to the first order theory of finite and infinite traces (*real traces*). Finally, in [7] we were able to show (by a quite different approach) that the fragment with future tense operators only is expressively complete. This has answered positively a long standing open question [9, 30]. The present paper shows how to use our result [7] in order to establish the very same characterizations for safety and liveness properties of real traces, which are well-known for infinite words. The characterizations are related to some topological concepts. But as soon as we have independent letters a and b, $a \neq b$, we should not use any topology being Hausdorff. This is perhaps a surprise, because in the case of infinite words one might also use the Cantor topology on Σ^ω which can be defined by the usual prefix metric. The single infinite string $(ab)^\omega$ defines a safety property over infinite words, but if a and b are independent, then every safety property containing $(ab)^\omega$ should also contain a^*b^ω and b^*a^ω. This is exactly captured by the Scott topology, which turns out be the appropriate choice. Safety means Scott-closed and liveness means Scott-dense. Of course, in the special case of infinite words, we obtain the classical result [2].

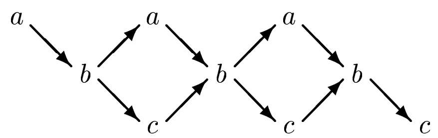
For the syntactic (or nearly) syntactic characterizations of safety and liveness formulae we prefer an algebraic approach using aperiodic monoids. Therefore, we give first a direct translation from temporal logic to aperiodic monoids. In the setting of trace theory these constructions are more involved than in the word case, and the translation seems to be new. But in any case, this direct translation from temporal logic to aperiodic monoids is still much simpler than converting first to first-order logic and then passing from first-order logic to aperiodic monoids. Since we give the translation for formulae having simultaneously future tense and past tense operators, we introduce the new notion of a homomorphism recognizing a formula. Once we are in the framework of aperiodic monoids, a separation theorem (in the spirit of Gabbay [13]) is a direct consequence of our result in [7], and the syntactic descriptions follow.

2 Real traces

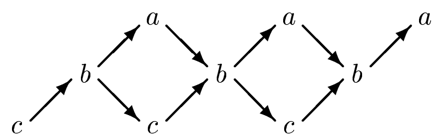
By (Σ, I) we mean an *independence alphabet* where Σ denotes a finite alphabet and $I \subseteq \Sigma \times \Sigma$ is an irreflexive and symmetric relation called the *independence relation*. The complementary relation $D = (\Sigma \times \Sigma) \setminus I$ is called the *dependence relation*. It is reflexive and symmetric. A dependence graph is (an isomorphism class of) a node–labeled, acyclic, directed graph $[V, E, \lambda]$, where V is a set of nodes, each node $i \in V$ is labeled by $\lambda(i) \in \Sigma$, and we have $(\lambda(i), \lambda(j)) \in D$ if and only if $(i, j) \in E \cup E^{-1} \cup \mathrm{id}_V$. A *real trace* is a dependence graph $[V, E, \lambda]$ such that for all $j \in V$ the set $\{i \in V \mid iE^*j\}$ of elements below j is finite. In the following we consider real traces, only. A *real trace* can be finite or infinite. The *empty trace* is denoted by ε, it corresponds to the empty graph. The alphabet of a real trace $x = [V, E, \lambda]$ is $\mathrm{alph}(x) = \lambda(V)$. The set of real traces is denoted by $\mathbb{R}$ and the set of finite traces is denoted by $\mathbb{M}$. We also write $(x, y) \in I$, if in fact $\mathrm{alph}(x) \times \mathrm{alph}(y) \subseteq I$, so I becomes a relation between real traces.

The concatenation of a finite trace $[V_1, E_1, \lambda_1]$ with a real trace $[V_2, E_2, \lambda_2]$ yields the real trace $[V, E, \lambda]$, which is defined by the disjoint union of the labeled graphs together with additional arcs from all vertices $v_1 \in V_1$ to $v_2 \in V_2$ where $(\lambda(v_1), \lambda(v_2)) \in D$. By this the set of finite traces becomes a monoid with the empty trace ε as unit. It is also clear how to define the infinite product $x = x_0 x_1 x_2 \cdots$ of a sequence $(x_i)_{i \geq 0}$ of finite traces. This infinite product is a finite trace if and only if $x_i = \varepsilon$ for almost all $i \geq 0$. As usual, we denote by x^ω the ω-product of a finite trace x. A *language* L is a subset of $\mathbb{R}$, it is called *finitary*, if it is a subset of $\mathbb{M}$.

For a graphical representation it is useful to draw the Hasse diagram. For instance, let $\Sigma = \{a, b, c\}$ and $I = \{(a, c), (c, a)\}$. Then $x = (abc)^3$ is given by the following picture.



We get the reverse of a finite trace by reading it from right to left. Formally, let $x = [V, E, \lambda]$ be a finite trace. The *reverse* $\widetilde{x}$ is defined by $\widetilde{x} = [V, E^{-1}, \lambda]$, thus is the dependence graph where the direction of arcs has been reversed. The operation is extended to finitary languages by $\widetilde{K} = \{x \in \mathbb{M} \mid \widetilde{x} \in K\}$. The reverse of the trace $(abc)^3$ above is $(cba)^3$ and given by:



A finite trace $x \in \mathbb{M}$ is a prefix of a real trace $z \in \mathbb{R}$, denoted by $x \leq z$, if $z = xy$ for some (unique) $y \in \mathbb{R}$. The prefix relation is generally defined for two real traces $x, z \in \mathbb{R}$ but here we use it only when the first trace x is finite.

The set $\mathbb{R}$ comes with natural topologies. For our purposes (in the framework of temporal logic) the topology of interest is the Scott topology. A language $L \subseteq \mathbb{R}$ is called *Scott-open*, if for all $z \in L$ there is some finite prefix $x \leq z$ such that $x\mathbb{R} \subseteq L$. The intersection $L_1 \cap L_2$ of two Scott-open sets is Scott-open, because $x_1 \leq z$ and $x_2 \leq z$ with $x_1, x_2 \in \mathbb{M}$ implies that $x_1 \leq x$ and $x_2 \leq x$ for some finite prefix $x \leq z$ (we can take for x the least upper bound of x_1, x_2 which is finite). Moreover, an arbitrary union of Scott-open sets is also clearly a Scott-open set. So, the Scott-open sets form indeed a topology. As usual, a language is called *Scott-closed*, if its complement $\mathbb{R} \setminus L$ is Scott-open. The *Scott-closure* $\overline{L}^\sigma$ of a language $L \subseteq \mathbb{R}$ is the smallest Scott-closed set containing L. It is given by

$$\overline{L}^\sigma = \{z \in \mathbb{R} \mid \forall x \in \mathbb{M}, x \leq z \text{ implies } x \leq y \text{ for some } y \in L\}.$$

Remark 1. The Scott topology (as well as the Lawson topology mentioned below) can be defined in more general settings, see [16] for a standard reference.

For an algebraic dcpo (directed complete partial order), the upper sets of compact elements form a basis of open sets for the Scott topology. Since the set of real traces with the prefix order is an algebraic dcpo with the finite traces as the compact elements [14], these upper sets are exactly the sets $x\mathbb{R}$ with $x \in \mathbb{M}$.

The Lawson topology uses the upper sets of compact elements as a sub-basis of open and closed sets. For the set of real traces this topology is also defined by the prefix metric. Intuitively two words or traces are close if they have the same prefixes up to some length. This prefix metric has many good properties: The metric space of real traces is compact (whence complete) and the set of finite traces is both dense and discrete. Probably, it is also the most frequently used topology on finite and infinite words. Restricted to Σ^ω this topology becomes the Cantor topology, i.e., it is the product topology over the discrete space Σ. However, the topology defined by the prefix metric is not appropriate for us as soon as we have independent letters (see the next remark) or as soon as we wish to deal with finite and infinite words simultaneously (see Remark 4 at the end of the paper).

Remark 2. We are looking for a topology on real traces for which the safe properties define closed sets. We show here that such a topology cannot be Hausdorff (hence cannot be defined by any metric) as soon as there are two independent letters.

Let $\Sigma = \{a, b\}$ with $(a, b) \in I$ and assume (on an informal level) that $(ab)^\omega$ defines a safe execution, then indeed all prefixes of $(ab)^\omega$ should be safe as well. Since $(a, b) \in I$, all finite traces are prefixes of $(ab)^\omega$ and the closure of the singleton set $L = \{(ab)^\omega\}$ should contain $\mathbb{M}$. This is not the case for an Hausdorff topology since then L itself is closed. On the other hand, the Scott-closure of L is $\mathbb{R}$ entirely, which is indeed safe.

We also see that the Scott topology restricted to infinite traces $\mathbb{R} \setminus \mathbb{M}$ is Hausdorff if and only if $I = \emptyset$.

3 Temporal logic for traces

The syntax of linear temporal logic $\mathrm{TL}(\Sigma)$, as we shall use it here, is defined as follows. There are a constant symbol $\perp$ representing *false*, the logical connectives $\neg$ (not) and $\vee$ (or), for each $a \in \Sigma$ a unary operator $\langle a \rangle$, called *next-a*, a unary operator $\langle a \rangle^{-1}$, called *yesterday–a*, and binary operators $\mathbf{U}$ and $\mathbf{S}$, called *until* and *since* respectively. Formally, the syntax for $\mathrm{TL}(\Sigma)$ is given by

$$\varphi ::= \perp \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle a \rangle\varphi \mid \varphi \,\mathbf{U}\, \varphi, \mid \langle a \rangle^{-1}\varphi \mid \varphi \,\mathbf{S}\, \varphi.$$

The semantics is defined by saying when some formula φ is satisfied by some real trace $z \in \mathbb{R}$ at some configuration (i.e., at some finite prefix) $x \in \mathbb{M}$; hence by defining $z, x \models \varphi$. This is done inductively:

$$
\begin{array}{ll}
z, x \not\models \bot, & \\
z, x \models \neg\varphi & \text{if } z, x \not\models \varphi, \\
z, x \models \varphi \vee \psi & \text{if } z, x \models \varphi \text{ or } z, x \models \psi, \\
z, x \models \langle a \rangle \varphi & \text{if } xa \leq z \text{ and } z, xa \models \varphi, \\
z, x \models \varphi \,\mathbf{U}\, \psi & \text{if } \exists y \in \mathbb{M} \text{ such that } xy \leq z,\ z, xy \models \psi, \text{ and} \\
& \qquad y = y'y'',\ y'' \neq 1 \text{ implies } z, xy' \models \varphi. \\
z, x \models \langle a \rangle^{-1} \varphi & \text{if } x = x'a \text{ and } z, x' \models \varphi, \\
z, x \models \varphi \,\mathbf{S}\, \psi & \text{if } \exists y \in \mathbb{M} \text{ such that } x = x'y,\ z, x' \models \psi, \text{ and} \\
& \qquad y = y'y'',\ y' \neq 1 \text{ implies } z, xy' \models \varphi.
\end{array}
$$

As usual, we define $\mathrm{L}_{\mathbb{R}}(\varphi)$ by the set of all real traces $z \in \mathbb{R}$ satisfying $z, \varepsilon \models \varphi$. Two formulae are *equivalent*, if they define the same language.

A formula φ not using any past modality $\langle a \rangle^{-1}$ or $\mathbf{S}$ is called a *pure future* formula. If it is not using any future modality $\langle a \rangle$ or $\mathbf{U}$, then it is called a *pure past* formula. We shall also use macros: $\top = \neg\bot$ (true), $\mathbf{F}\,\varphi = \top\,\mathbf{U}\,\varphi$ (eventually φ or future φ) and $\mathbf{G}\,\varphi = \neg\,\mathbf{F}\,\neg\varphi$ (always φ or globally φ). The operators $\mathbf{F}$ and $\mathbf{G}$ belong both to future tense.

Remark 3. Let $L \subseteq \mathbb{M}$ be a finitary language such that $L = \mathrm{L}_{\mathbb{R}}(\varphi)$ for some pure future formula φ. Then there is a pure past formula $\widetilde{\varphi}$ such that the reverse language $\widetilde{L}$ satisfies $\widetilde{L} = \{x \in \mathbb{M} \mid x, x \models \widetilde{\varphi}\}$. The formula $\widetilde{\varphi}$ is obtained by replacing in φ the future tense modalities by their corresponding past modalities.

4 Recognizing homomorphisms

Every language $L \subseteq \mathbb{R}$ which can be specified by a formula φ in temporal logic, can also be specified by some first-order formula over dependence graphs. Using well-known results in trace theory [17, 10] these languages are aperiodic.

Proposition 1 ([17, 10]). *Let $L = \mathrm{L}_{\mathbb{R}}(\varphi)$ for some temporal logic formula φ. Then L is aperiodic.*

However, the results of [17, 10] are far from being trivial. Therefore a direct proof of Proposition 1 is of interest. In this section we provide a direct translation from temporal logic formulae to aperiodic monoids. Recall that a finite monoid S is called *aperiodic*, if for some (threshold value) $n \geq 0$ we have $s^{n+1} = s^n$ for all $s \in S$. An example is the power set $\mathcal{P}(\Sigma)$ where the monoid operation is union. It is commutative and aperiodic with $n = 1$.

A language $L \subseteq \mathbb{R}$ is *recognized by a homomorphism* h from $\mathbb{M}$ to some finite monoid S if for all $y = y_0 y_1 y_2 \cdots \in L$ and $z = z_0 z_1 z_2 \cdots \in \mathbb{R}$ with $h(y_i) = h(z_i)$ for all $i \geq 0$, we have $z \in L$, too. The language $L \subseteq \mathbb{R}$ is called *aperiodic* if it is recognized by a homomorphism to some finite aperiodic monoid.

For $A \subseteq \Sigma$, we let $D(A) = \{b \in \Sigma \mid (a, b) \in D \text{ for some } a \in A\}$ and for $x \in \mathbb{M}$, we let $D(x) = D(\mathrm{alph}(x))$. For $x, y \in \mathbb{M}$, we have $D(xy) = D(x) \cup D(y)$ and $D(x) = \emptyset$ if and only if $x = \varepsilon$. Also, if $D(x) = D(x')$ and $D(y) = D(y')$ then $\mathrm{alph}(x) \times \mathrm{alph}(y) \subseteq I$ if and only if $\mathrm{alph}(x') \times \mathrm{alph}(y') \subseteq I$. Indeed, assume that

$(a', b') \in D$ for some $a' \in \text{alph}(x')$ and $b' \in \text{alph}(y')$. Then, $b' \in D(x') = D(x)$ and we find $a \in \text{alph}(x)$ with $(a, b') \in D$. Next, $a \in D(y') = D(y)$ and we find $b \in \text{alph}(y)$ with $(a, b) \in D$.

A homomorphism $h : \mathbb{M} \to S$ is called a *D-morphism* if for all $x, y \in \mathbb{M}$, $h(x) = h(y)$ implies $D(x) = D(y)$. It is worth noting that if a real trace language $L \subseteq \mathbb{R}$ is recognized by some homomorphism $h : \mathbb{M} \to S$, it is also recognized by the D-morphism $h' : \mathbb{M} \to S' = S \times \mathcal{P}(\Sigma)$ defined by $h'(x) = (h(x), D(x))$. If S is aperiodic, then the monoid S' is aperiodic with the same threshold value as S (in case S was non trivial). Thus, whenever convenient, we may assume that an aperiodic language is recognized by some (surjective) D-morphism.

Let $h : \mathbb{M} \to S$ be a surjective D-morphism. For $s \in S$, we let $D(s) = D(x)$ if $h(x) = s$. This is well-defined since h is a surjective D-morphism. Hence we may define the relation I on S. For $s, t \in S$ with $s = h(x)$ and $t = h(y)$ we set $(s, t) \in I$ if $\text{alph}(x) \times \text{alph}(y) \subseteq I$. Again, this is well-defined since h is a surjective D-morphism. Note that if $s, t \in S$ with $(s, t) \in I$ then $st = ts$. In fact, we are even more liberal, let $f : \mathbb{M} \to S$ and $g : \mathbb{M} \to T$ be two surjective D-morphisms, then $(s, t) \in I$ for $(s, t) \in S \times T$ means $\text{alph}(f^{-1}(s)) \times \text{alph}(g^{-1}(t)) \subseteq I$.

There is also a reverse operation $\widetilde{\ }$ for monoids: $\widetilde{S}$ is the same set as S, but the new multiplication is defined by $s \circ t = t \cdot s$. If S is aperiodic, then $\widetilde{S}$ has the same property. If a homomorphism $h : \mathbb{M} \to S$ recognizes some finitary language $L \subseteq \mathbb{M}$, then the homomorphism $\tilde{h} : \mathbb{M} \to \tilde{S}$ defined by $\tilde{h}(a) = h(a)$ recognizes $\widetilde{L}$. Note also that if h is a D-morphism then so is $\tilde{h}$.

In the proof of the main proposition, we use twice the I-diamond product introduced in [15] and which generalizes the Schützenberger product. We modify the version of the I-diamond product as given in [14, page 459] to accommodate D-morphisms. Originally, the definition was given for *alphabetic* homomorphisms, but the precise alphabetic information is actually not needed. Considering D-morphisms allows to use smaller monoids, and our construction does not introduce unnecessary information in the case where the independence relation is empty.

Let $f : \mathbb{M} \to S$ and $g : \mathbb{M} \to T$ be two surjective D-morphisms. Consider the power set $\mathcal{P}(S \times T)$ and equip this set with the following multiplication:

$$P \cdot Q = \{(rs, tu) \mid (r, t) \in P, (s, u) \in Q \text{ with } (s, t) \in I\}.$$

It was shown in [14] that this multiplication is associative with $\{(1, 1)\}$ as unit and that the mapping h from $\mathbb{M}$ to $\mathcal{P}(S \times T)$ defined by $h(x) = \{(f(x'), g(x'')) \mid x = x'x''\}$ is a homomorphism. The mapping h is also a D-morphism. Note that $h(x) = h(y)$ implies both $f(x) = f(y)$ and $g(x) = g(y)$. Indeed, $(f(x), f(\varepsilon)) \in h(x) = h(y)$ and therefore $y = y'y''$ with $f(y') = f(x)$ and $f(y'') = f(\varepsilon)$. Since f is a D-morphism, we deduce that $y'' = \varepsilon$ and $y' = y$. The argument is similar for g.

We define the I-diamond product $\diamondsuit_I(f, g)$ to be the homomorphism from $\mathbb{M}$ to the submonoid $h(\mathbb{M})$ of $\mathcal{P}(S \times T)$ defined by $\diamondsuit_I(f, g)(x) = h(x)$.

Lemma 1. *If $f : \mathbb{M} \to S$ and $g : \mathbb{M} \to T$ are two surjective D-morphisms onto aperiodic monoids, then the image $\diamondsuit_I(f,g)(\mathbb{M})$ is an aperiodic monoid and $\diamondsuit_I(f,g)$ is a surjective D-morphism.*

Proof. We use the notation introduced above. Assume that S and T are aperiodic, let $m, n \geq 0$ such that $s^{m+1} = s^m$ and $t^{n+1} = t^n$ for all $s \in S$ and $t \in T$. We claim that $\diamondsuit_I(f,g)(\mathbb{M}) = h(\mathbb{M})$ is aperiodic with threshold value $k = m+n+|\Sigma|$. Thus, we have to show $P^{k+1} = P^k$ for $P = h(x)$. We use some standard reasoning from trace theory. First of all it is enough to consider the case where x is connected, this means that the dependence graph of x is connected in the usual graph theoretical sense. Indeed, if x is not connected, then $x = x_1x_2 = x_2x_1$ with $x_1 \neq \varepsilon$ and $x_2 \neq \varepsilon$. Hence by induction on the size of the alphabet $h(x_i)^{k+1} = h(x_i)^k$ for $i = 1, 2$ and then $h(x)^{k+1} = h(x_1)^{k+1}h(x_2)^{k+1} = h(x_1)^k h(x_2)^k = h(x)^k$. Hence let x be connected and consider some factorization $yz = x^{k+1}$. We assume that $(f(y), g(z)) \in P^{k+1}$ and we show that $(f(y), g(z)) \in P^k$.

Using the generalization of Levi's Lemma (see e.g. [6, Prop. 3.2.3]) we have $y = y_1 \cdots y_{k+1}$, $z = z_1 \cdots z_{k+1}$ with $x = y_iz_i$ for all $1 \leq i \leq k+1$ and $(y_j, z_i) \in I$ for all $1 \leq i < j \leq k+1$. If x^{m+1} is not a prefix of y then $z_p \neq \varepsilon$ for some $p \leq m+1$ and using the fact that x is connected, we deduce that

$$\emptyset \neq \mathrm{alph}(z_p) \subsetneq \cdots \subsetneq \mathrm{alph}(z_{p+q}) = \mathrm{alph}(x)$$

for some $0 \leq q < |\Sigma|$. Then, we have $z_\ell = x$ for all $\ell > p+q$ and x^{n+1} is a suffix of z. Therefore, either x^{m+1} appears as a prefix of y or x^{n+1} appears as a suffix of z. In both cases, the pair $(f(y), g(z))$ appears in P^k, too. This shows $P^{k+1} \subseteq P^k$. The same argument applied to a factorization $yz = x^k$ shows $P^k \subseteq P^{k+1}$. Hence $P^{k+1} = P^k$, and $h(\mathbb{M})$ is aperiodic. $\square$

We turn now to the direct proof for Proposition 1. Since φ may involve simultaneously past and future modalities it is not enough to recognize languages. Therefore, we define that a homomorphism $h : \mathbb{M} \to S$ recognizes a formula φ, if for all $y = y_0y_1y_2 \cdots \in \mathbb{R}$ and $z = z_0z_1z_2 \cdots \in \mathbb{R}$ with $h(y_i) = h(z_i)$ for all $i \geq 0$, we have $y, y_0 \models \varphi$ if and only if $z, z_0 \models \varphi$. If h recognizes φ, then h recognizes $\mathrm{L}_{\mathbb{R}}(\varphi)$. Therefore Proposition 1 follows from the next lemma.

Lemma 2. *If φ is a temporal logic formula, then there is a D-morphism h from $\mathbb{M}$ to some finite aperiodic monoid S which recognizes φ.*

Proof. The proof is by structural induction on φ. The formula $\varphi = \bot$ is recognized by the D-morphism $D : \mathbb{M} \to \mathcal{P}(\Sigma)$. Next, if a homomorphism recognizes some formula φ, then it recognizes its negation as well. Also, assume that φ_1 and φ_2 are recognized by the D-morphisms $h_1 : \mathbb{M} \to S_1$ and $h_2 : \mathbb{M} \to S_2$ with S_1, S_2 aperiodic. Let $h : \mathbb{M} \to S = S_1 \times S_2$ be the homomorphism defined by $h(x) = (h_1(x), h_2(x))$. Then, S is aperiodic, h is a D-morphism and recognizes simultaneously φ_1, φ_2, and all Boolean combinations of φ_1 and φ_2.

In order to emphasize that we include past tense modalities let us begin with $\varphi = \langle a \rangle^{-1}\psi$. By induction, there is some surjective D-morphism $h : \mathbb{M} \to S$ with S aperiodic which recognizes ψ.

Let $\mathcal{C} \subseteq \mathcal{P}(\Sigma)$ be the family of subsets $A \subseteq \Sigma$ such that A consists of pair-wise independent letters. The meaning for sets in $\mathcal{C}$ is to remember the labels of maximal elements of finite traces. Note that for $A \in \mathcal{C}$ the product $\prod_{a \in A} h(a) \in S$ is well-defined, because $(a, b) \in I$ implies $h(a)h(b) = h(b)h(a)$ in S. For simplicity, we denote this product by $h(A)$.

We define an operation on $S' = S \times \mathcal{C}$ by:

$$(r, A) \cdot (s, B) := (r \cdot h(A \cap (D(s) \cup D(B))) \cdot s\,, (A \setminus (D(s) \cup D(B))) \cup B).$$

It can be verified that this operation is associative and that $(1, \emptyset)$ is a unit. Therefore, $(S', \cdot)$ is a monoid. To see that S' is aperiodic let n be the threshold for the monoid S. Hence $s^{n+1} = s^n$ for all $s \in S$. Consider an element $(s, B) \in S'$. Then, we have $(s, B)^{n+2} = ((sh(B))^{n+1}s, B) = ((sh(B))^n s, B) = (s, B)^{n+1}$.

We define a homomorphism $h' : \mathbb{M} \to S'$ by $h'(a) = (1, \{a\})$. The invariant is that if $x = x' \prod_{a \in A} a$ and A is the set of labels of maximal elements of $x \in \mathbb{M}$, then we have $h'(x) = (h(x'), A)$. From this, it follows that h' is a D-morphism. We show that h' recognizes $\varphi = \langle a \rangle^{-1} \psi$.

Let $y = y_0 y_1 y_2 \cdots \in \mathbb{R}$ and $z = z_0 z_1 z_2 \cdots \in \mathbb{R}$ with $h'(y_i) = h'(z_i) = (r_i, A_i)$ for all $i \geq 0$. Assume that $y, y_0 \models \langle a \rangle^{-1} \psi$. We have $y_0 = y_0' a$ and $y, y_0' \models \psi$. Since $h'(y_0) = h'(z_0) = (r_0, A_0)$ we deduce from the invariant that $a \in A_0$, $z_0 = z_0' a$ and $h(y_0') = r_0 h(A_0 \setminus \{a\}) = h(z_0')$. Also, $h(y_i) = r_i h(A_i) = h(z_i)$ for all $i \geq 1$. Since h recognizes ψ, we obtain $z, z_0' \models \psi$ and therefore $z, z_0 \models \langle a \rangle^{-1} \psi$.

The situation for $\varphi = \langle a \rangle \psi$ is symmetric and therefore omitted.

We give now the construction for a formula of type $\varphi \,\mathbf{U}\, \psi$. We may assume that φ and ψ are both recognized by the same surjective D-morphism f from $\mathbb{M}$ onto some finite aperiodic monoid S. We use a nested construction of the $\diamondsuit_I$-product, and we consider

$$g = \diamondsuit_I(f, f) \qquad \text{and} \qquad h = \diamondsuit_I(g, f).$$

By Lemma 1, h is a surjective D-morphism onto some finite aperiodic monoid. We claim that h recognizes the formula $\varphi \,\mathbf{U}\, \psi$. Let $y = y_0 y_1 y_2 \cdots \in \mathbb{R}$ and $z = z_0 z_1 z_2 \cdots \in \mathbb{R}$ such that $h(y_i) = h(z_i)$ for all $i \geq 0$. Assume that $y, y_0 \models \varphi \,\mathbf{U}\, \psi$. Let us show that $z, z_0 \models \varphi \,\mathbf{U}\, \psi$. For some finite prefix $y_{1,1} \leq y_1 y_2 \cdots$ we have $y, y_0 y_{1,1} \models \psi$. Grouping factors together we may assume that $y_{1,1} \leq y_1$ and therefore $y_1 = y_{1,1} y_{1,2}$. Using $h(y_1) = h(z_1)$ we find a factorization $z_1 = z_{1,1} z_{1,2}$ with $g(y_{1,1}) = g(z_{1,1})$ and $f(y_{1,2}) = f(z_{1,2})$. Therefore we have $f(y_0 y_{1,1}) = f(z_0 z_{1,1})$ and $f(y_i) = f(z_i)$ for all $i > 1$. Since $y, y_0 y_{1,1} \models \psi$ and f recognizes ψ, we deduce that $z, z_0 z_{1,1} \models \psi$. Now, let $z_{1,1} = z' z''$ with $z'' \neq \varepsilon$. Using $g(y_{1,1}) = g(z_{1,1})$, we find a factorization $y_{1,1} = y' y''$ with $y'' \neq \varepsilon$ such that $f(y') = f(z')$ and $f(y'') = f(z'')$. Since $y, y_0 y' \models \varphi$ and f recognizes φ we obtain $z, z_0 z' \models \varphi$. Hence the claim.

The construction for $\varphi \,\mathbf{S}\, \psi$ is symmetric. This proves Lemma 2 and Proposition 1. $\square$

5 Safety and liveness

According to [20] a formula φ is called a *safety formula*, if φ can be written as $\varphi = \mathbf{G}\, p$ where p is a pure past formula. A *safety property* is a language $L \subseteq \mathbb{R}$ which can be specified by some safety formula. Assume that $z, \varepsilon \not\models \mathbf{G}\, p$, then $z, x \not\models p$ for some finite prefix $x \leq z$. Since p is a pure past formula, $y, x \not\models p$ for all y in the Scott-open set $x\mathbb{R}$. Therefore, a safety property is Scott-closed.

For infinite words liveness properties have been widely studied, let us mention e.g. the work of Alpern and Schneider [2, 3]. The situation is more complicated. We use the formulation due to [5] and we say that φ is a *liveness formula*, if it is equivalent to a formula of the form $\mathbf{F}(q \vee \bigvee_{t \in T}(p_t \wedge f_t))$ where T is a finite index set, $q = \bigwedge_{t \in T} \neg p_t$, and moreover for all t we have that p_t is a pure past formula and that f_t is a satisfiable pure future formula. Note that this is not really a syntactic description of a liveness formula. The essential non-syntactic part is that we demand f_t to be satisfiable. We may suppress the q for an even less syntactical description. A formula φ is a liveness formula if and only if it is equivalent to a formula of type $\mathbf{F}(\bigvee_{t \in T}(p_t \wedge f_t))$, where T is a finite index set, $\mathbf{G}(\bigvee_{t \in T} p_t)$ is valid, and for all t the formula p_t is a pure past formula and f_t is a satisfiable pure future formula. A formula of the type above is called canonical liveness formula in [5]. The translation between the two variants for liveness formulae is obvious.

If $L = \mathrm{L}_{\mathbb{R}}(\varphi)$ for some liveness formula, then L is called a *liveness property*. Every liveness property is dense in the Scott topology: Indeed, let U be a Scott-open set and let $x \in \mathbb{M}$ with $x\mathbb{R} \subseteq U$. Then for some $t \in T$ we have $x, x \models p_t$, since $\mathbf{G}(\bigvee_{t \in T} p_t)$ is valid (using the canonical form). Since f_t is satisfiable, we find some $z \in \mathbb{R}$ with $z, \varepsilon \models f_t$. Hence $xz, x \models \varphi$ and $x\mathbb{R} \cap \mathrm{L}_{\mathbb{R}}(\varphi) \neq \emptyset$.

The following statement is well-known for infinite words, see [5, Thm. 10]. We have now the necessary tools in order to extend this result to the set of real traces.

Theorem 1.

1. *A language $L \subseteq \mathbb{R}$ is a safety property if and only if it is aperiodic and Scott-closed.*
2. *A language $L \subseteq \mathbb{R}$ is a liveness property if and only if it is aperiodic and Scott-dense.*
3. *Every aperiodic language is the intersection of some safety property and some liveness property.*

We will see that in some sense Theorem 1 is a corollary of the following result (which is the extension of Kamp's Theorem to real traces).

Theorem 2 ([7]). *Let $L \subseteq \mathbb{R}$ be an aperiodic language. Then there is a pure future formula φ such that $L = \mathrm{L}_{\mathbb{R}}(\varphi)$.*

Due to the reverse operation, Theorem 2 and Remark 3 imply:

Corollary 1. *Let $L \subseteq \mathbb{M}$ be a finitary aperiodic language. Then there is a pure past formula φ such that $L = \{x \in \mathbb{M} \mid x, x \models \varphi\}$.*

Using Proposition 1 and the remarks above, we already know that a safety property is an aperiodic language that is Scott-closed and a liveness property is an aperiodic language that is Scott-dense. To get the other claims of Theorem 1 we start with a general observation. Consider an arbitrary aperiodic language $L \subseteq \mathbb{R}$. Let us fix a surjective homomorphism h from $\mathbb{M}$ onto some finite aperiodic monoid S which recognizes L and such that $h^{-1}(1) = \{\varepsilon\}$. We first show in Equation 1 below that L can be expressed by two special formulae of $\mathrm{TL}(\Sigma)$.

For each $t \in S$ let $L(t) = \{z \in \mathbb{R} \mid h^{-1}(t)z \cap L \neq \emptyset\}$ and we let $T = \{t \in S \mid L(t) \neq \emptyset\}$. Since $h^{-1}(1) = \{\varepsilon\}$, we can verify that the homomorphism h recognizes $h^{-1}(t) \subseteq \mathbb{M}$ and $L(t) \subseteq \mathbb{R}$ for all $t \in T$. Moreover, a simple reflection shows $L = \bigcup_{t \in T} h^{-1}(t)L(t)$.

By Corollary 1 and Theorem 2, for each $t \in T$ we may choose a pure past formula p_t and a satisfiable pure future formula f_t such that $h^{-1}(t) = \{x \in \mathbb{M} \mid x, x \models p_t\}$ and $L(t) = \mathrm{L}_{\mathbb{R}}(f_t)$. Hence, from the equality $L = \bigcup_{t \in T} h^{-1}(t)L(t)$, we can derive the following description:

$$L = \mathrm{L}_{\mathbb{R}}\left(\mathbf{G}\left(\bigvee_{t \in T}(p_t \wedge f_t)\right)\right) = \mathrm{L}_{\mathbb{R}}\left(\mathbf{F}\left(\bigvee_{t \in T}(p_t \wedge f_t)\right)\right). \tag{1}$$

We are now looking more closely at safety and liveness properties. We know that if L is a liveness property, then it is Scott-dense. Let us show that if L is Scott-dense, then $T = S$. By contradiction, assume that $T \neq S$ and consider $s \in S$ with $L(s) = \emptyset$. Since h is surjective, we find some $x \in h^{-1}(s)$ and $L(s) = \emptyset$ implies that L does not intersect the Scott-open set $x\mathbb{R}$ which is not possible since L is dense. Hence, we have shown that if L is Scott-dense, then $T = S$. Finally, if $T = S$, then $\mathbf{G}(\bigvee_{t \in T} p_t)$ is valid, so the description $\mathbf{F}(\bigvee_{t \in T}(p_t \wedge f_t))$ is a canonical liveness formula since, by definition of T, all f_t are satisfiable. Therefore, L is a liveness property. Hence, we already have the proof of Theorem 1 (ii).

Note that every liveness property has a description $\mathbf{G}(\bigvee_{t \in T}(p_t \wedge f_t))$, but we have to be careful, in general $\mathbf{G}(\bigvee_{t \in T}(p_t \wedge f_t))$ does not necessarily specify any dense set.

Example 1. Let $\Sigma = \{a, b\}$ with $(a, b) \in D$. Consider the formula

$$\varphi = \mathbf{G}\left(\left(\neg\langle a\rangle^{-1}\top \wedge \neg\langle b\rangle^{-1}\top\right) \vee \left(\langle a\rangle^{-1}\top \wedge \langle a\rangle\langle b\rangle\top\right) \vee \left(\langle b\rangle^{-1}\top \wedge \langle a\rangle\top\right)\right).$$

This formula is of type $\mathbf{G}(\bigvee_{t \in T}(p_t \wedge f_t))$, where $\mathbf{G}(\bigvee_{t \in T} p_t)$ is valid, and all f_t are satisfiable. However $\mathrm{L}_{\mathbb{R}}(\varphi) = \{\varepsilon\}$ which is not dense.

On the other hand, if $\Sigma = \{a, b\}$ with $(a, b) \in I$, then $\mathrm{L}_{\mathbb{R}}(\varphi) = \{\varepsilon, (ab)^\omega\}$, which is a liveness property in this case.

Once, we have Equation 1, we can derive the Scott-closure $\overline{L}^\sigma$ of L. Recall that $\overline{L}^\sigma = \{z \in \mathbb{R} \mid \forall x \in \mathbb{M}, x \leq z \text{ implies } x \leq y \text{ for some } y \in L\}$. We may assume that all f_t are satisfiable (if not, replace T by $\{t \in T \mid \mathrm{L}_{\mathbb{R}}(f_t) \neq \emptyset\}$). We claim that $\overline{L}^{\sigma} = \mathrm{L}_{\mathbb{R}}(\mathbf{G}(\bigvee_{t \in T} p_t))$. Indeed, let $z \in \overline{L}^{\sigma}$. For each finite prefix $x \leq z$, we find $y \in L$ with $x \leq y$. Using $L = \mathrm{L}_{\mathbb{R}}(\mathbf{G}(\bigvee_{t \in T}(p_t \wedge f_t)))$ we deduce that $y, x \models p_t \wedge f_t$ for some $t \in T$. Since p is a pure past formula, we get $z, x \models p_t$ and $z, \varepsilon \models \mathbf{G}(\bigvee_{t \in T} p_t)$. Conversely, assume that $z, \varepsilon \models \mathbf{G}(\bigvee_{t \in T} p_t)$. For each finite prefix $x \leq z$, we find $t \in T$ with $z, x \models p_t$. Now, since f_t is satisfiable, there exists x' with $x', \varepsilon \models f_t$. Using that f_t is pure future, we get $y, x \models (p_t \wedge f_t)$ with $y = xx'$ and therefore $y \in \mathrm{L}_{\mathbb{R}}(\mathbf{F}(\bigvee_{t \in T}(p_t \wedge f_t))) = L$ concluding the proof of the claim.

The claim implies that the Scott-closure of the aperiodic language L is a safety property (hence is still aperiodic) that is expressed by the safety formula $\mathbf{G}(\bigvee_{t \in T} p_t)$. As a corollary, we also get that if L is aperiodic and Scott-closed then L is a safety property, completing the proof of Theorem 1 (ii).

Finally, for Theorem 1 (ii), we want to see how to write L as an intersection of a safety and a liveness property. To do so, we define $q = \bigwedge_{t \in T} \neg p_t$. This is a pure past formula and by the very definition, $\mathbf{F}(q \vee \bigvee_{t \in T}(p_t \wedge f_t))$ is a liveness formula. Of course, $\mathbf{G}(\bigvee_{t \in T} p_t)$ is a safety formula and

$$\mathrm{L}_{\mathbb{R}}\left(\mathbf{G}\left(\bigvee_{t \in T} p_t\right)\right) \cap \mathrm{L}_{\mathbb{R}}\left(\mathbf{F}\left(q \vee \bigvee_{t \in T}(p_t \wedge f_t)\right)\right) = \mathrm{L}_{\mathbb{R}}\left(\mathbf{F}\left(\bigvee_{t \in T}(p_t \wedge f_t)\right)\right) = L.$$

Summarizing what we have done so far, we may restate Theorem 1 in a more precise version.

Theorem 3. *Let $L \subseteq \mathbb{R}$ be recognizable by some aperiodic monoid S. Then for some subset $T \subseteq S$ there are for all $t \in T$ a satisfiable pure past formula p_t and a satisfiable pure future formula f_t such that the following conditions hold:*

$$L = \mathrm{L}_{\mathbb{R}}\left(\mathbf{F}\left(\bigvee_{t \in T}(p_t \wedge f_t)\right)\right) = \mathrm{L}_{\mathbb{R}}\left(\mathbf{G}\left(\bigvee_{t \in T}(p_t \wedge f_t)\right)\right),$$

$$\mathrm{L}_{\mathbb{R}}\left(\mathbf{G}\left(\bigvee_{t \in T} p_t\right)\right) \text{ is the Scott-closure of } L.$$

Moreover, let $q = \bigwedge_{t \in T} \neg p_t$. Then $\widehat{L} = \mathrm{L}_{\mathbb{R}}(\mathbf{F}(q \vee \bigvee_{t \in T}(p_t \wedge f_t)))$ is a liveness property. The language L is the intersection of the liveness property $\widehat{L}$ and the Scott-closure $\overline{L}^{\sigma}$ of L (which is a safety property). If L is a liveness property, then $\mathbf{F}(\bigvee_{t \in T}(p_t \wedge f_t)$ is a liveness formula in canonical form.

Remark 4. As a final remark let us comment again on the role of the Scott topology for safety and liveness. A safety property should be prefix closed which is usually not the case for closed sets in a metric space. In the topology defined by the prefix metric every finite subset of finite traces is both open and closed. So, for characterizing safety properties with the prefix metric we need in addition to the topological closure also the prefix closure. But this means we speak about the Scott-closure. Moreover, in the metric space $\mathbb{R}$ a subset is dense if and only if it contains all finite traces, which is again not useful for a characterization of liveness properties.

If however one restricts to the space of infinite words Σ^ω, then the prefix relation is just equality and the difference between the Cantor topology defined by the prefix metric and the Scott topology vanishes. This explains why on ω-words (and on ω-words, only) safety and liveness properties can be characterized using a metric space.

References

1. IJ. J. Aalbersberg and G. Rozenberg. Theory of traces. *Theoretical Computer Science*, 60:1–82, 1988.
2. B. Alpern and F. B. Schneider. Defining liveness. *Information Processing Letters*, 21:181–185, 1985.
3. B. Alpern and F. B. Schneider. Recognizing safety and liveness. *Distributed Computing*, 2:117–126, 1987.
4. R. Alur, D. Peled, and W. Penczek. Model-checking of causality properties. In *Proceedings of LICS'95*, pages 90–100, 1995.
5. E. Chang, Z. Manna, and A. Pnueli. Characterization of temporal property classes. In W. Kuich, editor, *Proceedings of the 19th International Colloquium on Automata, Languages and Programming (ICALP'92), Vienna*, volume 623 of *Lecture Notes in Computer Science*, pages 474–486, Berlin-Heidelberg-New York, 1992. Springer.
6. C. Choffrut. Combinatorics in trace monoids I. In V. Diekert and G. Rozenberg, editors, *The Book of Traces*, chapter 3, pages 71–82. World Scientific, Singapore, 1995.
7. V. Diekert and P. Gastin. LTL is expressively complete for Mazurkiewicz traces. In U. M. et al., editor, *Proceedings of the 27th International Colloquium on Automata, Languages and Programming (ICALP'00), Geneva*, number 1853 in Lecture Notes in Computer Science, pages 211–222. Springer, 2000.
8. V. Diekert and G. Rozenberg, editors. *The Book of Traces*. World Scientific, Singapore, 1995.
9. W. Ebinger. *Charakterisierung von Sprachklassen unendlicher Spuren durch Logiken*. Dissertation, Institut für Informatik, Universität Stuttgart, 1994.
10. W. Ebinger and A. Muscholl. Logical definability on infinite traces. *Theoretical Computer Science*, 154:67–84, 1996. A preliminary version appeared in Proceedings of the 20th International Colloquium on Automata, Languages and Programming (ICALP'93), Lund (Sweden) 1993, Lecture Notes in Computer Science 700, 1993.
11. A. Ehrenfeucht, H. J. Hoogeboom, and G. Rozenberg. Combinatorial properties of dependence graphs. *Information and Computation*, 114(2):315–328, 1994.
12. A. Ehrenfeucht and G. Rozenberg. On the structure of dependence graphs. In K. Voss, H. J. Genrich, and G. Rozenberg, editors, *Concurrency and Nets*, pages 141–170, Berlin-Heidelberg-New York, 1987. Springer.
13. D. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. On the temporal analysis of fairness. In *Conference Record of the 12th ACM Symposium on Principles of Programming Languages*, pages 163–173, Las Vegas, Nev., 1980.
14. P. Gastin and A. Petit. Infinite traces. In V. Diekert and G. Rozenberg, editors, *The Book of Traces*, chapter 11, pages 393–486. World Scientific, Singapore, 1995.

15. P. Gastin, A. Petit, and W. Zielonka. An extension of Kleene's and Ochmański's theorems to infinite traces. *Theoretical Computer Science*, 125:167–204, 1994. A preliminary version was presented at ICALP'91, Lecture Notes in Computer Science 510 (1991).
16. G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. W. Mislove, and D. S. Scott. *A Compendium of Continuous Lattices.* Springer, Berlin-Heidelberg-New York, 1980.
17. G. Guaiana, A. Restivo, and S. Salemi. Star-free trace languages. *Theoretical Computer Science*, 97:301–311, 1992. A preliminary version was presented at STACS'91, Lecture Notes in Computer Science 480 (1991).
18. H. J. Hoogeboom and G. Rozenberg. Dependence graphs. In V. Diekert and G. Rozenberg, editors, *The Book of Traces*, chapter 2, pages 43–67. World Scientific, Singapore, 1995.
19. R. M. Keller. Parallel program schemata and maximal parallelism I. Fundamental results. *Journal of the Association for Computing Machinery*, 20(3):514–537, 1973.
20. Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems, Specification.* Springer, 1991.
21. A. Mazurkiewicz. Concurrent program schemes and their interpretations. DAIMI Rep. PB 78, Aarhus University, Aarhus, 1977.
22. M. Mukund and P. S. Thiagarajan. Linear time temporal logics over Mazurkiewicz traces. In *Proceedings of the 21th MFCS, 1996*, number 1113 in Lecture Notes in Computer Science, pages 62–92. Springer, 1996.
23. P. Niebert. A ν-calculus with local views for sequential agents. In *Proceedings of the 20th MFCS, 1995*, number 969 in Lecture Notes in Computer Science, pages 563–573. Springer, 1995.
24. W. Penczek. Temporal logics for trace systems: On automated verification. *International Journal of Foundations of Computer Science*, 4:31–67, 1993.
25. R. Ramanujam. Locally linear time temporal logic. In *Proceedings of LICS'96*, Lecture Notes in Computer Science, pages 118–128, 1996.
26. G. Rozenberg. Behaviour of elementary net systems. In W. Brauer, editor, *Petri nets: central models and their properties; advances in Petri nets; proceedings of an advanced course, Bad Honnef, 8.-19. Sept. 1986, Vol. 1*, number 254 in Lecture Notes in Computer Science, pages 60–94, Berlin-Heidelberg-New York, 1986. Springer.
27. G. Rozenberg and P. S. Thiagarajan. Petri nets: Basic notions, structure and behaviour. Number 224 in Lecture Notes in Computer Science, pages 585–668, Berlin-Heidelberg-New York, 1986. Springer.
28. P. S. Thiagarajan. A trace based extension of linear time temporal logic. In *Proceedings of LICS'94*, pages 438–447, 1994.
29. P. S. Thiagarajan. A trace consistent subset of PTL. In *Proceedings of CONCUR'95*, number 962 in Lecture Notes in Computer Science, pages 438–452, 1995.
30. P. S. Thiagarajan and I. Walukiewicz. An expressively complete linear time temporal logic for Mazurkiewicz traces. In *Proceedings of LICS'97*, 1997.
31. I. Walukiewicz. Difficult configurations – on the complexity of LTrL. In K. G. Larsen et al., editors, *Proceedings of the 25th International Colloquium on Automata, Languages and Programming (ICALP'98), Aalborg*, number 1443 in Lecture Notes in Computer Science, pages 140–151, Berlin-Heidelberg-New York, 1998. Springer.
32. I. Walukiewicz. Local logics for traces. *Journal of Automata, Languages and Combinatorics*, 2001. To appear.