An expressively complete temporal logic without past tense operators for Mazurkiewicz traces

Volker Diekert[1] and Paul Gastin[2]

[1] Inst. für Informatik, Universität Stuttgart, Breitwiesenstr. 20-22, D-70565 Stuttgart
`diekert@informatik.uni-stuttgart.de` Fax: +49 711 78 16 310
[2] LIAFA, Université Paris 7, 2, place Jussieu, F-75251 Paris Cedex 05, France
`Paul.Gastin@liafa.jussieu.fr`

Abstract. Mazurkiewicz traces are a widely accepted model of concurrent systems. We introduce a linear time temporal logic $\mathrm{LTL_f}$ which has the same expressive power as the first order theory $\mathrm{FO}(<)$ of finite (infinite resp.) traces. The main contribution of the paper is that we only use future tense modalities in order to obtain expressive completeness. Our proof is direct and uses no reduction to words. As a formal consequence Kamp's theorem for both finite and infinite words becomes a corollary. This direct approach became possible due to a new proof technique of Wilke developed for the case of finite words.
Keywords Temporal logics, Mazurkiewicz traces, concurrency

1 Introduction

The verification of programs is essential for the conception of critical systems, especially for concurrent systems. The model checking approach starts with the abstraction of the actual system into some automata based model. Then the specification to be checked is expressed in some suitable logic, mainly temporal logics. Finally, a tool (model checker) is used to determine whether the system (the automaton) meets its specification (satisfies the formula).

Usually, concurrent systems are reduced to sequential ones by considering all possible linearizations of concurrent behaviors. Then, one can use both the temporal logics for sequences and the existing tools to specify and verify properties of systems. The main problem with this approach is the state explosion induced by these many linearizations. Recently, partial orders have been proposed to improve the efficiency of model checkers applied to concurrent systems [9, 15].

An alternative approach is to use truly concurrent models such as Mazurkiewicz traces and to introduce and study logics over these traces. A lot of work has already been done along this line in the past years [1, 13, 14, 16, 18, 21, 22]. See also [3] for the general background of trace theory, and in particular [17] for traces and logic and [8] for infinite traces.

The various linear time temporal logics differ by the kind of modalities allowed. These modalities may be classified into two main classes: future modalities (next, eventually, until, ...) and past modalities (previous, since, ...). For sequential systems, a crucial result states that linear time temporal logics are expressively complete, i.e., have the same expressive power as the first order theory of words $\mathrm{FO}(<)$. This is known now as Kamp's theorem, [10]. The result that we can avoid past tense operators was shown, however, much later; it relies on Gabbay's separation theorem [6].

Since then simplified versions of the proof of Kamp's theorem were found. Let us mention [2], which contains an elegant proof based on Krohn-Rhodes decomposition. The most recent development is due to Wilke [25] who gave an elementary proof based on the classical fact that FO languages are aperiodic and hence given by some counter–free automaton [11, 19]. Our approach follows the same lines as Wilke's proof for the corresponding result over finite words.

Kamp's theorem has been generalized by Ebinger [4] to Mazurkiewicz traces, but he used a temporal logic with both past and future modalities, and the proof failed for infinite traces. Then, Thiagarajan and Walukiewicz [23] have introduced a temporal logic LTrL for traces with the usual future modalities and also past tense modalities in the weak form of *previous* constants. They proved (via a reduction to the word case) that this logic is expressively complete, both for finite and infinite traces. It is open whether the fragment of LTrL without the previous constants is still expressively complete. A positive answer to this question was claimed for finite traces in [12], but the proof contained a serious flaw, which has not yet been fixed.

In the present paper we work with a linear time temporal logic for traces having future modalities only. In addition to the classical *next* and *until* modalities, we introduce a new (but natural) operator $\langle B^*\rangle\varphi$ which means that we can reach a configuration satisfying φ by using actions from some given set B only. We can think that the operator works as a *filter* which is reflected in the index f in the notation $\mathrm{LTL_f}$. In the word case, $\langle B^*\rangle\varphi$ means nothing but $\langle B\rangle\top\,\mathcal{U}\,\varphi$ and we can read $\langle B^*\rangle\varphi$ as a simple macro. Hence over words $\mathrm{LTL_f}$ becomes the classical linear time logic LTL. From this we may deduce that the new operator $\langle B^*\rangle\varphi$ can be avoided in the case of direct products of free monoids, but up to now we do not know whether this is possible for traces, in general.

We prove that $\mathrm{LTL_f}$ has the same expressive power as the first order theory of traces $\mathrm{FO}(<)$. The result holds for both finite and infinite traces. This solves an open problem of [4, 23] in the sense that we have a temporal logic using future modalities only. We would like to stress that, contrary to previous works, we are not reducing the problem to the word case. We give a direct proof for finite and infinite traces such that, formally, the result for words becomes a corollary.

The hard part of the proof is to find an $\mathrm{LTL_f}(\Sigma)$ formula which is equivalent to some given $\mathrm{FO}(<)$ formula. For this, we use the equivalence between first–order and aperiodic trace languages [5].

All our constructions are effective, and logics like LTrL or our $\mathrm{LTL_f}$ are clearly decidable. But one should be aware of the complexities involved. The satisfiability problem of $\mathrm{FO}(<)$ is non-elementary in the word and in the trace case, whereas the satisfiability problem of linear time temporal logic is PSPACE-complete in the word case [20]. In the presence of concurrency the situation is more complex. Walukiewicz [24] has shown that the satisfiability problem for the fragment without previous constants of LTrL is non-elementary over an independence alphabet with four letters. Hence the same statement holds for $\mathrm{LTL_f}$.

2 Preliminaries

Throughout the first part we speak of finite traces only. The infinitary case is treated in Sect. 4. The separation will make it necessary to repeat some parts, which could be treated uniformly otherwise. We have decided to do so for at least two reasons: The result for finite traces is used in order to obtain the corresponding result for infinite traces. Second, the subject is technically involved. So we prefer to postpone the notions needed to deal with infinite traces as long as possible hoping that the first part becomes then more easily accessible.

By (Σ, I) we mean a finite *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*. The monoid of *finite traces* $\mathbb{M}(\Sigma, I)$ is defined as a quotient monoid with respect to the congruence relation induced by I, i.e., $\mathbb{M}(\Sigma, I) = \Sigma^*/\{\, ab = ba \mid (a, b) \in I\,\}$. We also write $\mathbb{M}$ instead of $\mathbb{M}(\Sigma, I)$. For $A \subseteq \Sigma$ we denote by $\mathbb{M}_A$ the submonoid of $\mathbb{M}(\Sigma, I)$ generated by A:

$$\mathbb{M}_A = \mathbb{M}(A, I \cap A \times A) = \{x \in \mathbb{M}(\Sigma, I) \mid \mathrm{alph}(x) \subseteq A\}.$$

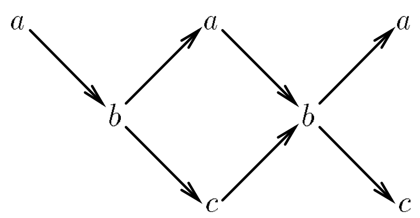
A trace $x \in \mathbb{M}$ is given by a congruence class of a word $a_1 \cdots a_n \in \Sigma^*$ where $a_i \in \Sigma$, $1 \leq n$. By abuse of language, but for simplicity we denote a trace x by one of its representing words $a_1 \cdots a_n$. The number n is called the length of x, it is denoted by $|x|$. For $n = 0$ we obtain the *empty* trace, it is denoted by 1. The *alphabet* $\mathrm{alph}(x)$ of a trace x is the set of letters occurring in x. A *trace language* is a subset $L \subseteq \mathbb{M}$. The concatenation is defined as usual:

$$KL = \{xy \in \mathbb{M} \mid x \in K, y \in L\}.$$

The Kleene–star L^* refers to the submonoid of $\mathbb{M}$ which is generated by the set L. We have $L^* = \bigcup_{i \geq 0} L^i$ where $L^0 = \{1\}$ and L^i is the i-fold iteration of L with itself, $i \geq 0$. For $A \subseteq \Sigma$ we have $A^* = \{x \in \mathbb{M} \mid \mathrm{alph}(x) \subseteq A\}$, which is ambiguous, because A^* could also denote the free monoid generated by the set A. It will be clear from the context what we mean.

Every trace $a_1 \cdots a_n \in \mathbb{M}$ can be identified with its dependence graph. This is (an isomorphism class of) a node–labeled, acyclic, directed graph $[V, E, \lambda]$, where $V = \{1, \ldots, n\}$ is a set of vertices, each $i \in V$ is labeled by $\lambda(i) = a_i$, and there is an edge $(i, j) \in E$ if and only if both $i < j$ and $(\lambda(i), \lambda(j)) \in D$. In pictures it is common to draw the Hasse diagram only. Thus, all redundant edges are omitted.

Example 1. Let $(\Sigma, D) = a \;\text{——}\; b \;\text{——}\; c$, i.e., $I = \{(a, c), (c, a)\}$. Then the trace $x = abcabca$ is given by



By $\min(x)$ and $\max(x)$ we refer to the *minimal* and *maximal letters* in the dependence graph. In the example above $\min(x) = \{a\}$ and $\max(x) = \{a, c\}$. Formally:

$$\min(x) = \{a \in \Sigma \mid x \in a\mathbb{M}\},$$
$$\max(x) = \{a \in \Sigma \mid x \in \mathbb{M}a\}.$$

For $B, C \subseteq \Sigma$ and $\# \in \{\subseteq, =, \supseteq, \neq\}$ we define:

$$\begin{aligned}
I(B) &= \{a \in \Sigma \mid \forall b \in B : (a,b) \in I\},\\
D(B) &= \{a \in \Sigma \mid \exists b \in B : (a,b) \in D\},\\
(\mathrm{Min} \# B) &= \{x \in \mathbb{M} \mid \min(x) \# B\},\\
(\mathrm{Max} \# B) &= \{x \in \mathbb{M} \mid \max(x) \# B\},\\
[B, C] &= \{x \in \mathbb{M} \mid \mathrm{alph}(x) = B,\ \max(x) \subseteq C\}.
\end{aligned}$$

If in the notations above B (or C) is a singleton, then we usually omit braces and write e.g. $I(b)$ or $\mathrm{Max} = b$.

A trace language of the form $K = B_1^* b_1 \cdots B_k^* b_k$ with $k \geq 0$, is called a *max–filter*, if $B_i = \{b_i, \ldots, b_k\}$ for $1 \leq i \leq k$. In this case, we have $\max(x) = \max(b_1 \cdots b_k)$ for all $x \in K$. The languages above $(\mathrm{Max} \# B)$ and $[B, C]$ are finite unions of max–filters. For example, $[B, C]$ is the finite union over all max–filters $B_1^* b_1 \cdots B_k^* b_k$ such that $B = \{b_1, \ldots, b_k\}$ and $\mathrm{Max}(b_1 \cdots b_k) \subseteq C$.

The syntax of the temporal logic $\mathrm{LTL_f}(\Sigma)$ 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*, for each $B \subseteq \Sigma$ a unary operator $\langle B^* \rangle$ called *B-filter*, and a binary operator $\mathcal{U}$ called *until*. Thus, the syntax is given by:

$$\varphi ::= \perp \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle a \rangle \varphi \mid \langle B^* \rangle \varphi \mid \varphi \,\mathcal{U}\, \varphi,$$

where $a \in \Sigma$ and $B \subseteq \Sigma$.

Usually, the semantics is defined by saying when some formula φ is satisfied by some trace z at some configuration (i.e., prefix) x; hence by defining $(z, x) \models \varphi$. Since our temporal logic uses future modalities only, we have $(z, x) \models \varphi$ if and only if $(y, 1) \models \varphi$, where y is the unique trace satisfying $z = xy$. Therefore, we do not need to introduce configurations and it is enough to say when a trace satisfies a formula at the empty configuration, denoted simply by $z \models \varphi$. This is done inductively on the formula as follows:

$$\begin{array}{ll}
z \not\models \perp, & \\
z \models \neg\varphi & \text{if } z \not\models \varphi,\\
z \models \varphi \vee \psi & \text{if } z \models \varphi \text{ or } z \models \psi,\\
z \models \langle a \rangle \varphi & \text{if } z = ay \text{ and } y \models \varphi,\\
z \models \langle B^* \rangle \varphi & \text{if } z = xy,\ x \in \mathbb{M}_B \text{, and } y \models \varphi,\\
z \models \varphi \,\mathcal{U}\, \psi & \text{if } z = xy,\ y \in \mathbb{M},\ y \models \psi, \text{ and } x = x'x'',\ x'' \neq 1 \text{ implies } x''y \models \varphi.
\end{array}$$

As usual, we define $\mathrm{L}_{\mathbb{M}}(\varphi) = \{x \in \mathbb{M} \mid x \models \varphi\}$. We say that a trace language $L \subseteq \mathbb{M}$ is *expressible in* $\mathrm{LTL_f}(\Sigma)$, if there exists a formula $\varphi \in \mathrm{LTL_f}(\Sigma)$ such that $L = \mathrm{L}_{\mathbb{M}}(\varphi)$. Equivalently, we can define inductively the language $\mathrm{L}_{\mathbb{M}}(\varphi)$ as follows:

$$\begin{array}{rlrl}
\mathrm{L}_{\mathbb{M}}(\perp) &= \emptyset & \mathrm{L}_{\mathbb{M}}(\langle a \rangle \varphi) &= a\mathrm{L}_{\mathbb{M}}(\varphi)\\
\mathrm{L}_{\mathbb{M}}(\neg\varphi) &= \mathbb{M} \setminus \mathrm{L}_{\mathbb{M}}(\varphi) & \mathrm{L}_{\mathbb{M}}(\langle B^* \rangle \varphi) &= \mathbb{M}_B\, \mathrm{L}_{\mathbb{M}}(\varphi)\\
\mathrm{L}_{\mathbb{M}}(\varphi \vee \psi) &= \mathrm{L}_{\mathbb{M}}(\varphi) \cup \mathrm{L}_{\mathbb{M}}(\psi) & \mathrm{L}_{\mathbb{M}}(\varphi \,\mathcal{U}\, \psi) &= \mathrm{L}_{\mathbb{M}}(\varphi) \,\mathcal{U}\, \mathrm{L}_{\mathbb{M}}(\psi)
\end{array}$$

where the *until* operator $\mathcal{U}$ is defined on trace languages by

$$L \,\mathcal{U}\, K = \{xy \mid x \in \mathbb{M},\ y \in K, \text{ and } x = x'x'',\ x'' \neq 1 \text{ implies } x''y \in L\}.$$

As an easy exercise let us state the following lemma, which will be used frequently.

Lemma 1. *Let $L \subseteq \mathbb{M}$ be expressible in* $\mathrm{LTL_f}(\Sigma)$ *and $B, C \subseteq \Sigma$. Then the languages* $(\mathrm{Min}\ \#\ B)$, $(\mathrm{Max}\ \#\ B)$, $[B, C]$, *and* $[B, C]L$ *are expressible in* $\mathrm{LTL_f}(\Sigma)$.

The following operators are standard abbreviations. They serve as macros.

$$\begin{array}{rll} \top &:= \neg\bot & \textit{true}, \\ \langle B\rangle\varphi &:= \bigvee_{b\in B}\langle b\rangle\varphi & \text{for } B \subseteq \Sigma, \\ \mathrm{X}\,\varphi &:= \langle\Sigma\rangle\varphi & \textit{neXt } \varphi, \\ \mathcal{F}\,\varphi &:= \top\,\mathcal{U}\,\varphi & \textit{future} \text{ (or } \textit{eventually}) \ \varphi, \\ \mathcal{G}\,\varphi &:= \neg\,\mathcal{F}\,\neg\varphi & \textit{globally } \varphi. \end{array}$$

Remark 1. For comparison let us mention that the syntax and semantics of the logic LTrL defined in [23] is very similar; the difference is only that instead of the modalities $\langle B^*\rangle\varphi$ there is for each letter $a \in \Sigma$ a constant $\langle a^{-1}\rangle\top$. Since the constant $\langle a^{-1}\rangle\top$ refer to the past, we need to use configurations to define its semantics: $(z, x) \models \langle a^{-1}\rangle\top$ if $a \in \max(x)$. It is not clear whether there is a direct translation of $\mathrm{LTL_f}(\Sigma)$ to LTrL or vice versa.

Remark 2. In the case of words, $I = \emptyset$, the meaning of $\langle B^*\rangle\varphi$ is equivalent to the formula $(\langle B\rangle\top)\,\mathcal{U}\,\varphi$. Thus, over words $\langle B^*\rangle\varphi$ is nothing but a simple macro. For traces this is not the case. We always have $\mathrm{L}_{\mathbb{M}}(\langle B^*\rangle\varphi) \subseteq \mathrm{L}_{\mathbb{M}}((\langle B\rangle\top)\,\mathcal{U}\,\varphi)$, but the reverse inclusion fails in general: Let $B = \{b\}$ and $a \in \Sigma$ such that $(a, b) \in I$. Then

$$\mathrm{L}_{\mathbb{M}}(\langle b^*\rangle\langle b\rangle\neg\,\mathrm{X}\,\top) = b^+ \neq \{a, b\}^*b \subseteq \mathbb{M}_{I(b)}b^+ = \mathrm{L}_{\mathbb{M}}((\langle b\rangle\top)\,\mathcal{U}\,(\langle b\rangle\neg\,\mathrm{X}\,\top)).$$

The operator $\langle B^*\rangle$ works like a *filter*, given a formula $\langle B^*\rangle\varphi$ the actions from B may pass (but nothing else) and what remains has to satisfy φ. It is open whether $\langle B^*\rangle\varphi$ can be expressed in terms of the other operators.
There is an operator $\langle B^*\rangle$ for all subsets of Σ, but not all of them are needed. For example, if $\mathbb{M}$ is a direct product of free monoids, then no $\langle B^*\rangle$ is needed at all, see Step 2 in the proof of Thm. 2.
Finally let us mention that the language $\mathrm{L}_{\mathbb{M}}(\langle B^*\rangle\varphi)$ has also a fixed-point definition since it is the unique solution to the equation $Z = \mathrm{L}_{\mathbb{M}}(\varphi) \cup BZ$.

Remark 3. Later we shall perform an induction on the size of Σ leading to formulae $\varphi \in \mathrm{LTL_f}(A)$ for $A \subseteq \Sigma$. We note that the interpretation over $\mathbb{M}(\Sigma, I)$ yields $\mathrm{L}_{\mathbb{M}_A}(\varphi) = \mathrm{L}_{\mathbb{M}}(\varphi) \cap \mathbb{M}_A$. Since $\mathbb{M}_A = \mathrm{L}_{\mathbb{M}}(\neg\,\mathcal{F}\langle\Sigma \setminus A\rangle\top) = \mathrm{L}_{\mathbb{M}}(\langle A^*\rangle\neg\,\mathrm{X}\,\top)$, a language $\mathrm{L}_{\mathbb{M}_A}(\varphi)$ is expressible in $\mathrm{LTL_f}(\Sigma)$. Another trivial observation is that if $\psi \in \mathrm{LTL_f}(\Sigma)$, then we can construct a formula $\psi_A \in \mathrm{LTL_f}(A)$ such that $\mathrm{L}_{\mathbb{M}}(\psi) \cap \mathbb{M}_A = \mathrm{L}_{\mathbb{M}_A}(\psi_A)$.

The first order theory of traces $\mathrm{FO}(<)$ is given by the syntax:

$$\varphi ::= P_a(x) \mid x < y \mid \neg\varphi \mid \varphi \vee \varphi \mid (\exists x)\varphi,$$

where $a \in \Sigma$ and $x, y \in \mathrm{Var}$ are first order variables. Given a trace $t = [V, E, \lambda]$ and a valuation of the free variables into the vertices $\sigma : \mathrm{Var} \to V$, the semantics is obtained by interpreting the relation $<$ as the transitive closure of E and the predicate $P_a(x)$ by $\lambda(\sigma(x)) = a$. Then we can say when $(t, \sigma) \models \varphi$. If φ is a closed formula (a sentence), then the valuation σ has an empty domain and we define the language $\mathrm{L}_{\mathbb{M}}(\varphi) = \{t \in \mathbb{M} \mid t \models \varphi\}$. We say that a trace language $L \subseteq \mathbb{M}$ is expressible in $\mathrm{FO}(<)$ if there exists some sentence $\varphi \in \mathrm{FO}(<)$ such that $L = \mathrm{L}_{\mathbb{M}}(\varphi)$.

Passing from a temporal logic formula to an FO($<$) one is not very difficult. It is well-known or belongs to folklore. The transformation relies on the fact that a prefix (configuration) p of a trace t can be defined by its maximal vertices. Such a set of maximal vertices is bounded by the maximal number of pairwise independent letters in Σ. Therefore, a prefix inside a trace can be defined using a bounded number of first order variables.
Our new modality $\langle B^*\rangle$ yields no extra difficulty: For instance, if some $\mathrm{LTL_f}(\Sigma)$–formula φ is equivalent to the FO–formula $\tilde{\varphi}$, then the $\mathrm{LTL_f}(\Sigma)$–formula $\langle B^+\rangle\varphi$ can be expressed by the FO–formula

$$\exists x_1 \cdots \exists x_k \left(\tilde{\varphi}(x_1, \ldots, x_k) \wedge \forall x \Big(\bigvee_{1\le i\le k} x \le x_i \Big) \longrightarrow \Big(\bigvee_{b\in B} P_b(x) \Big) \right)$$

where $k = |B|$ and $\tilde{\varphi}(x_1, \ldots, x_k)$ is the classical relativization of the formula $\tilde{\varphi}$ to the vertices which are not in the past of $x_1, \ldots, x_k$. Hence, we can state:

Proposition 1. *If a trace language is expressible in* $\mathrm{LTL_f}(\Sigma)$*, then it is expressible in* FO($<$).

As in the case of LTrL, this translation yields a non-elementary decision procedure for the uniform satisfiability problem of $\mathrm{LTL_f}$. (See also [7] for a modular decision procedure based on automata constructions.) For the lower bound, we can use [24], since the lower bound is given there for the fragment of LTrL without the previous constant $\langle a^{-1}\rangle\top$. Putting this together the result of Walukiewicz becomes:

Proposition 2 ([24]). *The satisfiability problem for both logics,* LTrL *and* $\mathrm{LTL_f}$*, is non-elementary over Mazurkiewicz traces.*

In the remainder we shall use the well-known equivalence between FO($<$)–definability and aperiodic languages. Recall that a finite monoid S is *aperiodic*, if there is some $n \ge 0$ such that $s^n = s^{n+1}$ for all $s \in S$. A trace language $L \subseteq \mathbb{M}$ is *aperiodic*, if there exists a morphism to some finite aperiodic monoid $h : \mathbb{M} \to S$ such that $L = h^{-1}(h(L))$.

Proposition 3 ([4, 5]). *A trace language is expressible in* FO($<$) *if and only if it is aperiodic.*

3 Kamp's theorem for finite traces

Theorem 1. *A trace language is expressible in* FO($<$) *if and only if it is expressible in* $\mathrm{LTL_f}(\Sigma)$.

By Props. 1 and 3 it is enough to show that all aperiodic languages in $\mathbb{M}(\Sigma, I)$ are expressible in $\mathrm{LTL_f}(\Sigma)$. This will cover the rest of this section.
In the following $h : \mathbb{M} \to S$ denotes a homomorphism to some finite aperiodic monoid S. A divisor of S is a homomorphic image of some submonoid of S. We shall use two simple facts. First, if $h : S \to G$ is a homomorphism to some group G, then $h(S)$ is trivial. Second, every divisor T of S is itself aperiodic. For a finite set (of states) Q we denote by $\mathrm{Trans}(Q)$ the monoid of mappings from Q to Q. The multiplication is the composition of mappings and the neutral element is the identity id_Q. Every finite monoid can be realized as a submonoid of some $\mathrm{Trans}(Q)$ where $|Q| \le |S|$. Hence we

assume $h : \mathbb{M} \to S \subseteq \mathrm{Trans}(Q)$ and we proceed by induction on $(|Q|, |\Sigma|)$. It is enough to construct a formula for $h^{-1}(s)$ where $s \in S$ and $h^{-1}(s) \neq \emptyset$. If $h(a) = \mathrm{id}_Q$ for all $a \in \Sigma$, which is in particular the case when $|Q| = 1$, then $s = \mathrm{id}_Q$ and $h^{-1}(\mathrm{id}_Q) = \mathbb{M}(\Sigma, I) = \mathrm{L}_{\mathbb{M}}(\top)$.

Hence we may assume that $h(b) \neq \mathrm{id}_Q$ for at least one $b \in \Sigma$ and we fix such a letter $b \in \Sigma$. The crucial observation here is that $h(b)$ is no permutation of Q. Indeed, if $h(b)$ were a permutation, then $h(b)$ would generate a non–trivial subgroup, which is impossible since S is aperiodic. Hence, $h(b)(Q) = Q'$ for some $Q' \subseteq Q$ with $|Q'| < |Q|$. Define $A = \Sigma \setminus \{b\}$ and let $g : \mathbb{M}_A \to S$ be the restriction of h to the submonoid $\mathbb{M}_A \subset \mathbb{M}$. By induction on $|\Sigma|$ we may assume that $g^{-1}(u)$ is expressible in $\mathrm{LTL_f}(A)$ (and hence in $\mathrm{LTL_f}(\Sigma)$ by Rem. 3) for all $u \in S$. Since $h^{-1}(s) = g^{-1}(s) \cup (h^{-1}(s) \cap \mathbb{M}b\mathbb{M})$, it is enough to construct a formula for $h^{-1}(s) \cap \mathbb{M}b\mathbb{M}$.

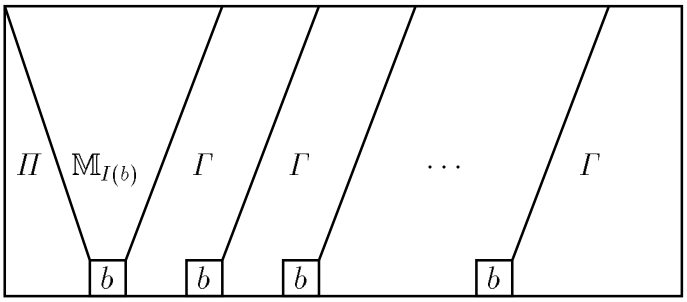
With respect to the letter b we define two more subsets of $\mathbb{M}$.

$$\Gamma = \{x \in \mathbb{M}_A \mid \min(x) \subseteq D(b)\},$$
$$\Pi = \{x \in \mathbb{M}_A \mid \max(x) \subseteq D(b)\}.$$

The notation Π is chosen since Πb are exactly the *pyramids* of $\mathbb{M}$ where the unique maximal element is b. It should be noted that $(b\Gamma)^*$ and $(\Pi b)^*$ are in fact free submonoids of $\mathbb{M}$, being infinitely generated if $D(b) \neq \{b\}$. We have the following unambiguous decomposition:

$$\mathbb{M}b\mathbb{M} = \Pi\, \mathbb{M}_{I(b)}\, b\, (\Gamma b)^*\, \Gamma.$$

This decomposition is best visualized by the following picture; it is in some sense the guide for the modular construction of the formula $h^{-1}(s) \cap \mathbb{M}b\mathbb{M}$.



Each $s \in h((\Gamma b)^*)$ maps the subset Q' to Q'. Hence we may define subsets $T, T' \subseteq \mathrm{Trans}(Q')$ by $T = \{s\lceil_{Q'} \mid s \in h(\Gamma b)\}$ and $T' = \{s\lceil_{Q'} \mid s \in h((\Gamma b)^*)\}$. Since $h((\Gamma b)^*)$ is a submonoid of S, the set T' is a monoid and T is a set of generators, the monoid T' is a divisor of S, hence it is aperiodic. By T^* we denote the free monoid generated by the set T (here T is viewed as an alphabet). The inclusion $T \subseteq T'$ induces a canonical homomorphism $e : T^* \to T'$ which is called the *evaluation*. Since T' is a submonoid of $\mathrm{Trans}(Q')$ and $|Q'| < |Q|$, we may use induction (although we might have $|T| > |\Sigma|$). Hence $e^{-1}(t) \subseteq T^*$ is expressible in $\mathrm{LTL_f}(T)$ for all $t \in T'$. (As a matter of fact, $e^{-1}(t)$ is a language of words in the free monoid T^*.)

We need some further notations. The mapping $\sigma : \Gamma b \to T$ defined by $\sigma(x) = h(x)\lceil_{Q'}$ induces a homomorphism $\sigma : (\Gamma b)^* \to T^*$ between free monoids. Therefore, we also have the morphism $e \circ \sigma : (\Gamma b)^* \to T'$. Note that for all $x \in (\Gamma b)^*$ we have $e \circ \sigma(x) = h(x)\lceil_{Q'}$. Now, for all $u, v, w \in S$ and $t \in T' \subseteq \mathrm{Trans}(Q')$ the product $uvh(b)tw$ is a well–defined mapping from Q to Q, since $h(b)(Q) = Q'$. Hence $uvh(b)tw$ is an element of S.

Therefore, using the unambiguous factorization $\mathbb{M} b\, \mathbb{M} = \Pi \mathbb{M}_{I(b)} b(\Gamma b)^* \Gamma$, the language $h^{-1}(s) \cap \mathbb{M} b\, \mathbb{M}$ can be written as the following finite union:

$$\bigcup_{uv\,h(b)tw=s} \left(g^{-1}(u) \cap \Pi\right)\left(g^{-1}(v) \cap \mathbb{M}_{I(b)}\right) b \left(\sigma^{-1}((e^{-1}(t))\right) \left(g^{-1}(w) \cap \Gamma\right).$$

It remains to show that if $L_1, L_2, L_3 \subseteq \mathbb{M}_A$ are trace languages expressible in $\mathrm{LTL_f}(A)$ and if $K = \sigma^{-1}(K')$ for some $K' \subseteq T^*$ being expressible in $\mathrm{LTL_f}(T)$, then the language

$$(L_1 \cap \Pi)\,(L_2 \cap \mathbb{M}_{I(b)})\, b\, K\, (L_3 \cap \Gamma)$$

is expressible in $\mathrm{LTL_f}(\Sigma)$.
This is done in the following technical lemmas.

Lemma 2. *Let $\varphi \in \mathrm{LTL_f}(A)$. Then the language $(\mathrm{L}_{\mathbb{M}_A}(\varphi) \cap \Pi)b\mathbb{M}$ is expressible in $\mathrm{LTL_f}(\Sigma)$.*

Proof. The assertion is trivial for $\varphi = \bot$. By structural induction we have to consider formulas of type $\neg\varphi, \varphi \vee \psi, \langle a\rangle\varphi, \langle B^*\rangle\varphi$, and $\varphi\, \mathcal{U}\, \psi$, where $a \in A$ and $B \subseteq A$. We deal with $\langle B^*\rangle\varphi$ and $\varphi\, \mathcal{U}\, \psi$ only, since the constructions for the other formulas are simpler.

- $\langle B^*\rangle\varphi$: We have

$$(\mathrm{L}_{\mathbb{M}_A}(\langle B^*\rangle\varphi) \cap \Pi)b\mathbb{M} = \bigcup_{B' \subseteq B, C \subseteq A} [B', D(C \cup \{b\})]\Big((\mathrm{L}_{\mathbb{M}_A}(\varphi) \cap \Pi)b\mathbb{M} \cap [C, D(b)]b\mathbb{M}\Big).$$

- $\varphi\, \mathcal{U}\, \psi$: We have

$$(\mathrm{L}_{\mathbb{M}_A}(\varphi\, \mathcal{U}\, \psi) \cap \Pi)b\mathbb{M} = ((\mathrm{L}_{\mathbb{M}_A}(\varphi) \cap \Pi)b\mathbb{M} \setminus b\mathbb{M})\, \mathcal{U}\, (\mathrm{L}_{\mathbb{M}_A}(\psi) \cap \Pi)b\mathbb{M}.$$

To see this, consider first $uvbw$ such that $uv \in \Pi, v \in \mathrm{L}_{\mathbb{M}_A}(\psi)$, and where for all $u = u'u'', u'' \neq 1$ we have $u''v \in \mathrm{L}_{\mathbb{M}_A}(\varphi)$. Then $u''v \in \Pi \setminus \{1\}$ and hence $u''vbw \notin b\mathbb{M}$. Therefore $uvbw$ is an element of the right–hand side.
Now, let z be an element of the right–hand side. Consider a factorization $z = uy$ with $|u|$ minimal such that $y \in (\mathrm{L}_{\mathbb{M}_A}(\psi) \cap \Pi)b\mathbb{M}$ and for all $u = u'u''$ with $u'' \neq 1$, we have $u''y \in (\mathrm{L}_{\mathbb{M}_A}(\varphi) \cap \Pi)b\mathbb{M} \setminus b\mathbb{M}$. Then, $y = vbw$ with $v \in \mathrm{L}_{\mathbb{M}_A}(\psi) \cap \Pi$ and also $b \notin \mathrm{alph}(u)$. Consider a factorization $u = u'u''$ with $u'' \neq 1$. We have $u''vbw \in (\mathrm{L}_{\mathbb{M}_A}(\varphi) \cap \Pi)b\mathbb{M}$. We have to show that $u''v \in \Pi$, because then $u''v \in \mathrm{L}_{\mathbb{M}_A}(\varphi)$, hence $uv \in \mathrm{L}_{\mathbb{M}_A}(\varphi\, \mathcal{U}\, \psi) \cap \Pi$. Let us write $u''vbw = xbw'$ with $x \in \Pi$. Since $b \notin \mathrm{alph}(u'')$ and $v \in \Pi$ we have $x = x'v$ and $u'' = x'x''$ for some x' and some x'' independent of v and b. We show that $x'' = 1$. Assume on the contrary that $x'' \neq 1$. Then $|x'| < |u''|$ and $z = (u'x')(vbx''w)$ is another possible factorization contradicting the minimality of $|u|$.

Lemma 3. *Assume that $C \times \{b\} \subseteq I$ and let $\varphi \in \mathrm{LTL_f}(C)$. Then the language $\mathrm{L}_{\mathbb{M}_C}(\varphi)(\mathrm{Min} = b)$ is expressible in $\mathrm{LTL_f}(\Sigma)$.*

Proof. Again, we proceed by structural induction on φ. Everything is straightforward, up to the until–operator.

- $\varphi\, \mathcal{U}\, \psi$: We have

$$\mathrm{L}_{\mathbb{M}_C}(\varphi\, \mathcal{U}\, \psi)(\mathrm{Min} = b) \;=\; \Big(\mathrm{L}_{\mathbb{M}_C}(\varphi)(\mathrm{Min} = b)\Big)\, \mathcal{U}\, \Big(\mathrm{L}_{\mathbb{M}_C}(\psi)(\mathrm{Min} = b)\Big).$$

The inclusion "$\subseteq$" is trivial. Therefore let z be an element of the right–hand side. Consider a factorization $z = uy$ with $|u|$ minimal such that $y \in \mathrm{L}_{\mathbb{M}_C}(\psi)(\mathrm{Min} = b)$ and

for all $u = u'u''$ with $u'' \neq 1$, we have $u''y \in \mathrm{L}_{\mathbb{M}_C}(\varphi)(\mathrm{Min} = b)$. Then, $y = vw$ with $v \in \mathrm{L}_{\mathbb{M}_C}(\psi)$ and $\mathrm{Min}(w) = \{b\}$.
Assume that $\mathrm{alph}(u) \not\subseteq C$ and let $u = u'u''$ with $u'' \in d\mathbb{M}_C$ and $\mathrm{Min}(u'') = \{d\}$ for some $d \notin C$. Since $d \in \min(u''vw) \subseteq C \cup \{b\}$, we must have $b = d$. Hence $u'' = b$ since $C \times \{b\} \subseteq I$. Therefore $z = u'(vbw)$ is another possible factorization of z contradicting the minimality of $|u|$.
It follows that $\mathrm{alph}(u) \subseteq C$ and for all $u = u'u''$ with $u'' \neq 1$, $(u''v)w$ is the unambiguous factorization in $\mathbb{M}_C\,(\mathrm{Min} = b)$. We deduce that $u''v \in \mathrm{L}_{\mathbb{M}_C}(\varphi)$ since $u''vw \in \mathrm{L}_{\mathbb{M}_C}(\varphi)(\mathrm{Min} = b)$.

Lemma 4. *Let $L_1, L_2, L_3 \subseteq \mathbb{M}_A$ be trace languages expressible in $\mathrm{LTL_f}(A)$ and let $K \subseteq (\Gamma b)^*$ be a trace language such that $K\Gamma$ is expressible in $\mathrm{LTL_f}(\Sigma)$. Then the language*

$$(L_1 \cap \Pi)\,(L_2 \cap \mathbb{M}_{I(b)})\,b\,K\,(L_3 \cap \Gamma)$$

is expressible in $\mathrm{LTL_f}(\Sigma)$.

Proof. Since the product $\Pi\, b\, \mathbb{M}$ is unambiguous, we can write:

$$\begin{aligned}(L_1 \cap \Pi)(L_2 \cap \mathbb{M}_{I(b)})bK(L_3 \cap \Gamma) &= (L_1 \cap \Pi)b(L_2 \cap \mathbb{M}_{I(b)})K(L_3 \cap \Gamma)\\ &= (L_1 \cap \Pi)b\mathbb{M} \cap \Pi b(L_2 \cap \mathbb{M}_{I(b)})K(L_3 \cap \Gamma).\end{aligned}$$

By Lemma 2 and 1 it is enough to show that $b(L_2 \cap \mathbb{M}_{I(b)})K(L_3 \cap \Gamma)$ is expressible in $\mathrm{LTL_f}(\Sigma)$. Since the product $\mathbb{M}_{I(b)}\,(\mathrm{Min} = b)$ is unambiguous, we have

$$\begin{aligned}b(L_2 \cap \mathbb{M}_{I(b)})K(L_3 \cap \Gamma) &= (L_2 \cap \mathbb{M}_{I(b)})bK(L_3 \cap \Gamma)\\ &= (L_2 \cap \mathbb{M}_{I(b)})(\mathrm{Min} = b) \cap \mathbb{M}_{I(b)}bK(L_3 \cap \Gamma).\end{aligned}$$

By Lemma 3 it is enough to consider $K(L_3 \cap \Gamma)$. Finally, since the product $(\mathrm{Max} \supseteq b)\Gamma$ is unambiguous, we obtain if $1 \in K$ (which is equivalent with $1 \in K\Gamma$)

$$K(L_3 \cap \Gamma) = (L_3 \cap \Gamma) \cup \Big(K\Gamma \cap (\mathrm{Max} \supseteq b)I(b)^*(L_3 \cap \Gamma)\Big).$$

and if $1 \notin K$

$$K(L_3 \cap \Gamma) = K\Gamma \cap (\mathrm{Max} \supseteq b)I(b)^*(L_3 \cap \Gamma).$$

The assertion follows from Lemma 1 since $(\mathrm{Max} \supseteq b)$ is a finite union of max–filters, and Γ is expressible in $\mathrm{LTL_f}(\Sigma)$.

By Lemma 4 it remains to show the next lemma.

Lemma 5. *Let $\varphi \in \mathrm{LTL_f}(T)$. Then the trace language $\sigma^{-1}(L_T(\varphi))\Gamma$ is expressible in $\mathrm{LTL_f}(\Sigma)$.*

Proof. For a formula $\varphi \in \mathrm{LTL_f}(T)$ let us denote in this proof by $\sigma^{-1}(\varphi)$ the language $\sigma^{-1}(L_T(\varphi))$. We use structural induction on φ, the basis $\varphi = \perp$ being trivial. Since T^* is a free monoid, it is enough to consider formulas of the form $\neg\varphi$, $\varphi \vee \psi$, $\langle t \rangle \varphi$ where $t \in T$, and $\varphi\, \mathcal{U}\, \psi$.
• $\neg\varphi$: We have $\sigma^{-1}(\neg\varphi)\Gamma = ((\Gamma b)^* \setminus \sigma^{-1}(\varphi))\Gamma = (\Gamma b)^*\Gamma \setminus \sigma^{-1}(\varphi)\Gamma$ since the product $(\Gamma b)^*\Gamma$ is unambiguous. Moreover, the language $(\Gamma b)^*\Gamma = (\mathrm{Min} \subseteq D(b))$ is expressible in $\mathrm{LTL_f}(\Sigma)$ by Lemma 1.
• $\varphi \vee \psi$: Trivial.

• $\langle t\rangle\varphi$ where $t \in T$: Using the unambiguous decomposition

$$(\Gamma b)(\Gamma b)^*\Gamma = (\mathrm{Min} \subseteq D(b)) \cap \Pi \mathbb{M}_{I(b)} b(\Gamma b)^*\Gamma$$

we deduce that

$$\sigma^{-1}(\langle t\rangle\varphi)\Gamma = \bigcup_{uvh(b)=t} \Big(\mathrm{Min} \subseteq D(b)\Big) \cap \Big(g^{-1}(u) \cap \Pi\Big) \Big(g^{-1}(v) \cap \mathbb{M}_{I(b)}\Big) b\sigma^{-1}(\varphi)\Gamma.$$

By induction, $\sigma^{-1}(\varphi)\Gamma$ is expressible in $\mathrm{LTL_f}(\Sigma)$. We may apply Lemma 4 to conclude this case.
• $\varphi\,\mathcal{U}\,\psi$: An until–formula $\varphi\,\mathcal{U}\,\psi$ over words is equivalent with $\psi \vee \varphi \wedge \mathrm{X}(\varphi\,\mathcal{U}\,\psi)$. Thus it is enough to consider $\mathrm{X}(\varphi\,\mathcal{U}\,\psi)$. We claim that

$$\sigma^{-1}(\mathrm{X}(\varphi\,\mathcal{U}\,\psi))\Gamma = (\Gamma b)^*\Gamma \cap \bigg(\Big(b\sigma^{-1}(\varphi)\Gamma \cup (\mathrm{Min} \neq b)\Big)\,\mathcal{U}\,\Big(b\sigma^{-1}(\psi)\Gamma\Big)\bigg).$$

To see the claim, let first $w = u_1 b \cdots u_k b z \in \sigma^{-1}(\mathrm{X}(\varphi\,\mathcal{U}\,\psi))\Gamma$ such that $1 \le k$, $u_1, \ldots, u_k, z \in \Gamma$, and $t_1 \cdots t_k \in \mathrm{L}_{T^*}(\mathrm{X}(\varphi\,\mathcal{U}\,\psi))$ where $t_i = \sigma(u_i b)$ for $1 \le i \le k$. For some $1 \le j \le k$ we have $t_{j+1} \cdots t_k \in \mathrm{L}_{T^*}(\psi)$ and for all $2 \le i \le j$ we have $t_i \cdots t_k \in \mathrm{L}_{T^*}(\varphi)$. We can write $bu_{j+1}b \cdots u_k b z \in b\sigma^{-1}(\psi)\Gamma$. Consider a factorization $u_1 b \cdots u_{j-1} b u_j = xy$ such that $y \neq 1$. We have to show that $\min(y) = b$ implies $y \in b\sigma^{-1}(\varphi)\Gamma$. However this is clear, because $\min(y) = b$ implies $y = bu_i \cdots bu_j$ for some $2 \le i \le j$.
Now, let w be an element of the right–hand side. Since the factorization $(\Gamma b)^*\Gamma$ is unambiguous, there is a unique decomposition $w = u_1 b \cdots u_k b z$ with u_i, $z \in \Gamma$. For $1 \le i \le k$, we let $t_i = \sigma(u_i b)$. Let $1 \le j \le k$ be minimal such that $bu_{j+1}b \cdots u_k b z \in b\sigma^{-1}(\psi)\Gamma$. We must have $bu_i b \cdots u_k b z \in b\sigma^{-1}(\varphi)\Gamma$ for all $2 \le i \le j$. It follows that $t_1 \cdots t_k \in \mathrm{L}_{T^*}(\mathrm{X}(\varphi\,\mathcal{U}\,\psi))$.

This finishes the proof of Thm. 1.

4 Kamp's theorem for infinite traces

An *infinite trace* is an infinite dependence graph $[V, E, \lambda]$ such that for all $j \in V$ the set $\downarrow j = \{i \in V \mid i \le j\}$ is finite. A *real trace* is a finite or infinite trace. The set of real traces is denoted by $\mathbb{R}(\Sigma, I)$ or simply by $\mathbb{R}$. For a real trace $x = [V, E, \lambda]$ the *alphabet at infinity* is defined by the set of letters occurring infinitely many times in x, i.e., $\mathrm{alphinf}(x) = \{a \in \Sigma \mid |\lambda^{-1}(a)| = \infty\}$. We refer to [8] for details about infinite traces.
The aim of this section is to generalize Kamp's theorem for infinite words (i.e., for ω-words) to real traces. The expressiveness result for ω-words is shown in [6]. It is worth mentioning that we do not use this fact on ω-words, which becomes again a formal corollary.
We shall use the syntax of $\mathrm{LTL_f}(\Sigma)$. The semantics of $\mathrm{LTL_f}(\Sigma)$ is defined exactly as in the finitary case. For each $\varphi \in \mathrm{LTL_f}(\Sigma)$ there is a language $\mathrm{L}_{\mathbb{R}}(\varphi)$ and we have $\mathrm{L}_{\mathbb{M}}(\varphi) = \mathrm{L}_{\mathbb{R}}(\varphi) \cap \mathbb{M}$. Moreover, we can express $\mathbb{M}$ as the language $\mathrm{L}_{\mathbb{R}}(\mathcal{F} \neg \mathrm{X}\top)$.
Recall also that $x \in L\,\mathcal{U}\,K$, $L, K \subseteq \mathbb{R}$ means that there are y, z with $y \in \mathbb{M}$ such that $z \in K$ and for all $y = y'y''$, $y'' \neq 1$ we have $y''z \in L$. Clearly, as in the finitary case $\mathrm{L}_{\mathbb{R}}(\varphi\,\mathcal{U}\,\psi) = \mathrm{L}_{\mathbb{R}}(\varphi)\,\mathcal{U}\,\mathrm{L}_{\mathbb{R}}(\psi)$.

Theorem 2. *A language over real traces $L \subseteq \mathbb{R}(\Sigma, I)$ is expressible in* FO($<$) *if and only if it is expressible in* $\mathrm{LTL_f}(\Sigma)$.

As for finite traces, we can pass from an $\mathrm{LTL_f}(\Sigma)$ formula to a FO($<$) formula. It is also well-known that a language of real traces is FO($<$) definable if and only if it is aperiodic [5]. Let us recall the notions of recognizable and of aperiodic real trace language.
Let $h = \mathbb{M} \to S$ be a morphism into some finite monoid S. For $x, y \in \mathbb{R}$, we say that x and y are h-similar, denoted by $x \sim_h y$ if we can find infinite factorizations $x = x_1 x_2 \cdots$ and $y = y_1 y_2 \cdots$ into finite traces such that $h(x_i) = h(y_i)$ for all $i \geq 0$. A real trace language $L \subseteq \mathbb{R}$ is *recognized* by h if it is saturated by $\sim_h$. A real trace language $L \subseteq \mathbb{R}$ is *aperiodic* if it is recognized by some morphism into some finite aperiodic monoid.
We denote by $\approx_h$ the transitive closure of $\sim_h$ which is therefore an equivalence relation. It is well-known (by a Ramsey-type argument) that there are only finitely many equivalence classes $[x]_{\approx_h} = \{y \in \mathbb{R} \mid y \approx_h x\}$ and they form a finite partition of $\mathbb{R}$.

Remark 4. Working with real traces it is convenient to consider languages L which contain simultaneously finite and infinite traces. We allow to write a finite trace $x \in \mathbb{M}$ as an infinite product $x = x_1 x_2 \cdots$ where $x_i = 1$ for almost all i. Thus, it may happen that $x \sim_h y$ where x is finite and y is infinite.
This convention is a matter of taste, but it is quite natural in the presence of concurrency where we have independent components. Some of them may stop and other may run forever.

The remaining of this section consists in the proof that for each aperiodic trace language $L \subseteq \mathbb{R}$, there exists a formula φ in $\mathrm{LTL_f}(\Sigma)$ such that $L = \mathrm{L}_{\mathbb{R}}(\varphi)$. Clearly, this will show Thm. 2. Let $h = \mathbb{M} \to S$ be a morphism to some finite aperiodic monoid S. We can realize S as a submonoid of some transformation monoid Trans(Q). We show by induction on $(|Q|, |\Sigma|)$ that every language $L \subseteq \mathbb{R}$ recognized by h is expressible in $\mathrm{LTL_f}(\Sigma)$. We use that for each $s \in S$ the language $h^{-1}(s)$ is expressible in $\mathrm{LTL_f}(\Sigma)$ (Prop. 3, Thm. 1).

Step 1: Assume that $h(a) = \mathrm{id}_Q$ for all $a \in \Sigma$ (this is in particular the case when $|Q| = 1$). Then h recognizes only two languages: $\mathbb{R}$ and $\emptyset$ which are both expressible in $\mathrm{LTL_f}(\Sigma)$ by $\top$ and $\perp$ respectively.

Step 2: Assume that $\Sigma = \Sigma_1 \cup \Sigma_2$ where Σ_1, Σ_2 are nonempty subsets of Σ such that $\Sigma_1 \times \Sigma_2 \subseteq I$. For $i = 1, 2$ consider $\mathbb{M}_i = \mathbb{M}(\Sigma_i, I \cap \Sigma_i \times \Sigma_i)$, $\mathbb{R}_i = \mathbb{R}(\Sigma_i, I \cap \Sigma_i \times \Sigma_i)$, and $h_i : \mathbb{M}_i \to S$ the restriction of h to $\mathbb{M}_i$. We claim that L is a finite union of products of the form $L_1 L_2$ where L_1, L_2 are recognized by h_1, h_2 respectively.
Let $x \in L$, we can write $x = x_1 x_2$ with $x_1 \in \mathbb{R}_1$ and $x_2 \in \mathbb{R}_2$. For $i = 1, 2$, let $L_i = [x_i]_{\approx_{h_i}}$. Clearly, L_i is recognized by h_i and $x \in L_1 L_2$. Now, for $i = 1, 2$, let $y_i, z_i \in \mathbb{R}_i$ be such that $y_i \sim_{h_i} z_i$. Since $\Sigma_1 \times \Sigma_2 \subseteq I$, it is easy to see that $y_1 y_2 \sim_h z_1 z_2$. We deduce that $L_1 L_2 \subseteq L$. The claim follows since there are only finitely many equivalence classes under $\approx_{h_1}$ and $\approx_{h_2}$.
Now, we can conclude the second case. Using the induction on $|\Sigma|$ we know that L_1 and L_2 are expressible in $\mathrm{LTL_f}(\Sigma_1)$ and $\mathrm{LTL_f}(\Sigma_2)$ respectively. We deduce that L is expressible in $\mathrm{LTL_f}(\Sigma)$ using Lemma 6.

Lemma 6. *For $i = 1, 2$, let $L_i \subseteq \mathbb{R}_i$ be a trace language expressible in* $\mathrm{LTL_f}(\Sigma_i)$. *Then the language $L_1 L_2$ is expressible in* $\mathrm{LTL_f}(\Sigma)$.

Proof. The product $\mathbb{R}_1\mathbb{R}_2$ is unambiguous and we have $L_1L_2 = (L_1\mathbb{R}_2) \cap (\mathbb{R}_1 L_2)$. The result follow since we can show by structural induction that for each formula φ in $\mathrm{LTL_f}(\Sigma_1)$ we have $\mathrm{L}_{\mathbb{R}}(\varphi) = \mathrm{L}_{\mathbb{R}_1}(\varphi)\mathbb{R}_2$.

Step 3: For a subset $A \subseteq \Sigma$ we denote by $\overline{A}$ its complement, $\overline{A} = \Sigma \setminus A$, and for $\# \in \{\subseteq, =\}$, we define $(\mathrm{Alphinf}\ \#\ A) = \{x \in \mathbb{R} \mid \mathrm{alphinf}(x)\ \#\ A\}$. We show that if A is a proper subset of Σ then $L \cap (\mathrm{Alphinf} \subseteq A)$ is expressible in $\mathrm{LTL_f}(\Sigma)$. For $s \in S$ we let $L(s) = \{y \in \mathbb{R}_A \mid h^{-1}(s) \cdot y \cap L \neq \emptyset\}$. We first show that

$$L \cap (\mathrm{Alphinf} \subseteq A) = \bigcup_{s \in S} (h^{-1}(s) \cap (\mathrm{Max} \subseteq \overline{A}))L(s).$$

Let $z \in L \cap (\mathrm{Alphinf} \subseteq A)$. Using the unambiguous decomposition $(\mathrm{Alphinf} \subseteq A) = (\mathrm{Max} \subseteq \overline{A})\mathbb{R}_A$ we can write $z = xy$ with $x \in (\mathrm{Max} \subseteq \overline{A})$ and $y \in \mathbb{R}_A$. Let $s = h(x)$, we obtain $z = xy \in (h^{-1}(s) \cap (\mathrm{Max} \subseteq \overline{A}))L(s)$.
Conversely, let $s \in S$, $x \in h^{-1}(s) \cap (\mathrm{Max} \subseteq \overline{A})$ and $y \in L(s)$. Let $x' \in h^{-1}(s)$ be such that $x'y \in L$. Since $xy \sim_h x'y$ and L is recognized by h we deduce that $xy \in L$. Moreover, $\mathrm{alphinf}(xy) \subseteq \mathrm{alph}(y) \subseteq A$.
Then, we show that $L(s)$ is recognized by the restriction h_A of h to $\mathbb{M}_A$. Indeed, let $y, z \in \mathbb{R}_A$ be such that $y \sim_{h_A} z$. Note that this implies $y \sim_h z$. Assume that $y \in L(s)$, then there exists $x \in h^{-1}(s)$ such that $xy \in L$. Since $xy \sim_h xz$, we deduce that $xz \in L$ and then $z \in L(s)$.
From the finitary case we know that each language $h^{-1}(s)$ is expressible in $\mathrm{LTL_f}(\Sigma)$. So is the language $(\mathrm{Max} \subseteq \overline{A})$ as a finite union of max–filters. Using the induction on $|\Sigma|$, we deduce that the language $L(s)$ is expressible in $\mathrm{LTL_f}(A)$. From Lemma 7 we conclude that $L \cap (\mathrm{Alphinf} \subseteq A)$ is expressible in $\mathrm{LTL_f}(\Sigma)$.

Lemma 7. *Let $A \subseteq \Sigma$ and let $K \subseteq \mathbb{M}$ and $L \subseteq \mathbb{R}_A$ be trace languages expressible in* $\mathrm{LTL_f}(\Sigma)$ *and* $\mathrm{LTL_f}(A)$ *respectively. Then the language $(K \cap (\mathrm{Max} \subseteq \overline{A}))L$ is expressible in* $\mathrm{LTL_f}(\Sigma)$ *as well.*

Proof. First the product $(\mathrm{Max} \subseteq \overline{A})\mathbb{R}_A$ is unambiguous and we have

$$(K \cap (\mathrm{Max} \subseteq \overline{A}))L = (K \cap (\mathrm{Max} \subseteq \overline{A}))\mathbb{R}_A \cap (\mathrm{Max} \subseteq \overline{A})L.$$

The language $(\mathrm{Max} \subseteq \overline{A})$ is a union of max–filters. Hence, by Lemma 1, the language $(\mathrm{Max} \subseteq \overline{A})L$ is expressible in $\mathrm{LTL_f}(\Sigma)$.
We show now by induction on φ that the language $(\mathrm{L}_{\mathbb{M}}(\varphi) \cap (\mathrm{Max} \subseteq \overline{A}))\mathbb{R}_A$ is expressible in $\mathrm{LTL_f}(\Sigma)$. The proof is similar to that of Lemma 2.
- The cases $\perp$ and $\varphi \vee \psi$ are trivial.
- $\neg\varphi$: Since the product $(\mathrm{Max} \subseteq \overline{A})\mathbb{R}_A$ is unambiguous, we have

$$(\mathrm{L}_{\mathbb{M}}(\neg\varphi) \cap (\mathrm{Max} \subseteq \overline{A}))\mathbb{R}_A = (\mathrm{Max} \subseteq \overline{A})\mathbb{R}_A \setminus (\mathrm{L}_{\mathbb{M}}(\varphi) \cap (\mathrm{Max} \subseteq \overline{A}))\mathbb{R}_A.$$

Moreover $(\mathrm{Max} \subseteq \overline{A})\mathbb{R}_A$ is the set of traces such that the alphabet at infinity is contained in A and is clearly expressible in $\mathrm{LTL_f}(\Sigma)$.
- $\langle a \rangle\varphi$: Follows from Lemma 1 by writing $(\mathrm{L}_{\mathbb{M}}(\langle a \rangle\varphi) \cap (\mathrm{Max} \subseteq \overline{A}))\mathbb{R}_A$ as

$$\bigcup_{\{C \subseteq \Sigma \mid a \in \overline{A} \cup D(C)\}} a\Big((\mathrm{L}_{\mathbb{M}}(\varphi) \cap (\mathrm{Max} \subseteq \overline{A}))\mathbb{R}_A \cap [C, \overline{A}]\mathbb{R}_A\Big).$$

- $\langle B^*\rangle\varphi$: Similarly, we can write $(\mathrm{L}_{\mathbb{M}}(\langle B^*\rangle\varphi)\cap(\mathrm{Max}\subseteq\overline{A}))\mathbb{R}_A$ as

$$\bigcup_{B'\subseteq B, C\subseteq\Sigma}[B',\overline{A}\cup D(C)]\Big((\mathrm{L}_{\mathbb{M}}(\varphi)\cap(\mathrm{Max}\subseteq\overline{A}))\mathbb{R}_A\cap[C,\overline{A}]\mathbb{R}_A\Big).$$

- $\varphi\,\mathcal{U}\,\psi$: We have

$$(\mathrm{L}_{\mathbb{M}}(\varphi\,\mathcal{U}\,\psi)\cap(\mathrm{Max}\subseteq\overline{A}))\mathbb{R}_A=(\mathrm{L}_{\mathbb{M}}(\varphi)\cap(\mathrm{Max}\subseteq\overline{A}))\mathbb{R}_A\,\mathcal{U}\,(\mathrm{L}_{\mathbb{M}}(\psi)\cap(\mathrm{Max}\subseteq\overline{A}))\mathbb{R}_A.$$

Step 4: We may now assume that the alphabet Σ is connected and we show that for some language L' expressible in $\mathrm{LTL_f}(\Sigma)$ we have

$$L\cap(\mathrm{Alphinf}=\Sigma)\subseteq L'\subseteq L.$$

We proceed as for finite traces by choosing a letter $b\in\Sigma$ such that $h(b)\neq\mathrm{id}_Q$. We know that $Q'=h(b)(Q)$ is a proper subset of Q. We let $A=\Sigma\setminus\{b\}$, $\Pi=\{x\in\mathbb{M}_A\mid\max(x)\subseteq D(b)\}$ and $\Gamma=\{x\in\mathbb{M}_A\mid\min(x)\subseteq D(b)\}$. For $s\in S$ we define $L(s)=\{y\in\mathbb{R}\mid h^{-1}(s)\cdot y\cap L\neq\emptyset\}$. Since L is recognized by h, we deduce easily that $L(s)$ is recognized by h and $h^{-1}(s)L(s)\subseteq L$ for all $s\in S$.
Now we claim that $L\cap(\mathrm{Alphinf}=\Sigma)\subseteq L'\subseteq L$ with

$$L'=\bigcup_{u,v\in S}(h^{-1}(u)\cap\Pi)(h^{-1}(v)\cap\mathbb{M}_{I(b)})(L(uv)\cap(b\Gamma)^\infty).$$

Since Σ is connected and by the factorization $(\mathrm{Alphinf}=\Sigma)\subseteq\Pi\mathbb{M}_{I(b)}(b\Gamma)^\infty$ wich is unambiguous, the first inclusion $L\cap(\mathrm{Alphinf}=\Sigma)\subseteq L'$ follows easily. The second inclusion is clear since $h^{-1}(s)L(s)\subseteq L$ for all $s\in S$.

We use the definitions and notations introduced earlier for the sets $T,T'\subseteq\mathrm{Trans}(Q')$, the morphisms $e:T^*\to T'$ and $\sigma:(\Gamma b)^*\to T^*$ and we extend $\sigma:(\Gamma b)^\infty\to T^\infty$ naturally.
For $K\subseteq\mathbb{R}$ recognized by h we define

$$K'=\{\sigma(x)\mid bx\in K\cap b(\Gamma b)^\infty\}\subseteq T^\infty.$$

We show that
1) $K\cap b(\Gamma b)^\infty=b\sigma^{-1}(K')$
2) K' is recognized by the morphism $e:T^*\to T'$.
For the first point, one inclusion is clear. Conversely, let $y=y_1by_2b\cdots\in\sigma^{-1}(K')$ with $y_i\in\Gamma$ and let $bx\in K\cap b(\Gamma b)^\infty$ be such that $\sigma(x)=\sigma(y)$. We write $x=x_1bx_2b\cdots$ with $x_i\in\Gamma$. For all i we have $h(x_ib)\restriction_{Q'}=\sigma(x_ib)=\sigma(y_ib)=h(y_ib)\restriction_{Q'}$ and we deduce that $h(bx_ib)=h(by_ib)$. It follows that $bx\sim_h bz\sim_h by$ with $z=x_1by_2bx_3by_4b\cdots$. Since $bx\in K$ and $bx\approx_h by$, we deduce that $by\in K$ and we have proved the converse inclusion since $by\in b(\Gamma b)^\infty$.
For the second point, let $u,v\in T^\infty$ be such that $u\sim_e v$. We write $u=u_1u_2\cdots$ and $v=v_1v_2\cdots$ with $e(u_i)=e(v_i)$ for all i. Assume that $u\in K'$ and let $bx=bx_1bx_2b\cdots\in K$ with $x_ib\in(\Gamma b)^*$ and $\sigma(x_ib)=u_i$. Let $y=y_1by_2b\cdots$ with $y_ib\in(\Gamma b)^*$ and $\sigma(y_ib)=v_i$. Then, for all i we have $h(x_ib)\restriction_{Q'}=e(u_i)=e(v_i)=h(y_ib)\restriction_{Q'}$ and therefore $h(bx_ib)=h(by_ib)$. Hence, we have $bx\sim_h bz\sim_h by$ with $z=x_1by_2bx_3by_4b\cdots$ and therefore $bx\approx_h by$. Since $bx\in K$ which is recognized by h, it follows that $by\in K$ and $v=\sigma(y)\in K'$.

We have proved that K' is recognized by the morphism $e : T^* \to T' \subseteq \mathrm{Trans}(Q')$. We deduce by induction on $|Q|$ that K' is expressible in $\mathrm{LTL_f}(T)$. Using Lemma 9 (below) we deduce that $K \cap b(\Gamma b)^\infty = b\sigma^{-1}(K')$ is expressible in $\mathrm{LTL_f}(\Sigma)$. Finally, using Lemma 8, we deduce that L' is expressible in $\mathrm{LTL_f}(\Sigma)$.

Lemma 8. *Let $L_1, L_2 \subseteq \mathbb{M}$ and $L_3 \subseteq b(\Gamma b)^\infty$ be expressible in $\mathrm{LTL_f}(\Sigma)$. Then the language $(L_1 \cap \Pi)(L_2 \cap \mathbb{M}_{I(b)})L_3$ is expressible in $\mathrm{LTL_f}(\Sigma)$.*

Proof. Similar to that of Lemma 4. We use the decomposition

$$(L_1 \cap \Pi)(L_2 \cap \mathbb{M}_{I(b)})L_3 = (L_1 \cap \Pi)b\mathbb{R} \cap \Pi(L_2 \cap \mathbb{M}_{I(b)})(\mathrm{Min} = b) \cap \mathbb{M}_A L_3.$$

Lemma 9. *Let $K \subseteq T^\infty$ be a language expressible in $\mathrm{LTL_f}(T)$. Then the language $\sigma^{-1}(K)$ is expressible in $\mathrm{LTL_f}(\Sigma)$.*

Proof. Similar to that of Lemma 5. It is done by induction on the formula φ such that $K = \mathrm{L}_{T^\infty}(\varphi)$. Since the formula φ defines a word language $K \subseteq T^\infty$ we do not have to consider the modality $\langle B^* \rangle$. The cases $\perp$ and $\varphi \vee \psi$ are trivial.
• $\neg\varphi$: We have $\sigma^{-1}(\neg\varphi) = (\Gamma b)^\infty \setminus \sigma^{-1}(\varphi)$. Now $(\Gamma b)^\infty = (\Gamma b)^* \cup (\Gamma b)^\omega$. Then we write $(\Gamma b)^* = (\mathrm{Min} \subseteq D(b)) \cap \mathbb{M}b$ which is easily expressible in $\mathrm{LTL_f}(\Sigma)$. Now $(\Gamma b)^\omega$ is expressed by the formula $(\mathrm{Min} \subseteq D(b)) \wedge \mathcal{G}\mathcal{F}(\mathrm{Min} = b)$.
• $\langle t \rangle\varphi$: The unambiguous decomposition $(\Gamma b)(\Gamma b)^\infty = (\min \subseteq D(b)) \cap \Pi\mathbb{M}_{I(b)}b(\Gamma b)^\infty$, yields that

$$\sigma^{-1}(\langle t \rangle\varphi) = \bigcup_{uv\,h(b)=t} (\min \subseteq D(b)) \cap (h^{-1}(u) \cap \Pi)(h^{-1}(v) \cap \mathbb{M}_{I(b)})b\sigma^{-1}(\varphi).$$

We conclude using Lemma 8.
• $\varphi\,\mathcal{U}\,\psi$: An until-formula $\varphi\,\mathcal{U}\,\psi$ over words is equivalent with $\psi \vee (\varphi \wedge \mathrm{X}(\varphi\,\mathcal{U}\,\psi))$. Thus it is enough to consider $\mathrm{X}(\varphi\,\mathcal{U}\,\psi)$. We have

$$\sigma^{-1}(\mathrm{X}(\varphi\,\mathcal{U}\,\psi)) = (\Gamma b)^\infty \cap \Big(\Big(b\sigma^{-1}(\varphi) \cup (\mathrm{Min} \neq b)\Big)\,\mathcal{U}\,\Big(b\sigma^{-1}(\psi)\Big)\Big).$$

References

1. R. Alur, D. Peled, and W. Penczek. Model-checking of causality properties. In *Proceedings of LICS'95*, pages 90–100, 1995.
2. J. Cohen, D. Perrin, and J. E. Pin. On the expressive power of temporal logic. *Journal of Computer and System Sciences*, 46:271–295, 1993.
3. V. Diekert and G. Rozenberg, editors. *The Book of Traces*. World Scientific, Singapore, 1995.
4. W. Ebinger. *Charakterisierung von Sprachklassen unendlicher Spuren durch Logiken*. Dissertation, Institut für Informatik, Universität Stuttgart, 1994.
5. 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.
6. 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.

7. P. Gastin, R. Meyer, and A. Petit. A (non-elementary) modular decision procedure for ltrl. In L. Brim, F. Gruska, and J. Zlatuska, editors, *Proceedings of the 23rd International Symposium on Mathematical Foundations of Computer Science (MFCS'98)*, number 1450 in Lecture Notes in Computer Science, pages 356–365. Springer, 1998.
8. 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.
9. P. Godefroid. Partial-order methods for the verification of concurrent systems. In *LNCS 1032*, pages 176–185. Springer, 1996.
10. J. A. W. Kamp. *Tense Logic and the Theory of Linear Order.* PhD thesis, University of California, Los Angeles, Calif., 1968.
11. R. McNaughton and S. Papert. *Counter-free Automata.* MIT Press, 1971.
12. R. Meyer and A. Petit. Expressive completeness of LTrL on finite traces: an algebraic proof. In *Proceedings of STACS'98*, number 1373 in Lecture Notes in Computer Science, pages 533–543, 1998.
13. 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.
14. 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.
15. D. Peled. Partial-order reductions: model checking using representatives. In *Proceedings of MFCS'96*, number 1113 in Lecture Notes in Computer Science, pages 93–112, 1996.
16. W. Penczek. Temporal logics for trace systems: On automated verification. *International Journal of Foundations of Computer Science*, 4:31–67, 1993.
17. W. Penczek and R. Kuiper. Traces and logic. In V. Diekert and G. Rozenberg, editors, *The Book of Traces*, chapter 10, pages 307–390. World Scientific, Singapore, 1995.
18. R. Ramanujam. Locally linear time temporal logic. In *Proceedings of LICS'96*, Lecture Notes in Computer Science, pages 118–128, 1996.
19. M. P. Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8:190–194, 1965.
20. A. Sistla and E. Clarke. The complexity of propositional linear time logic. *J. ACM*, 32:733–749, 1985.
21. P. S. Thiagarajan. A trace based extension of linear time temporal logic. In *Proceedings of the 9th Annual IEEE Symposium on Logic in Computer Science (LICS'94)*, pages 438–447, 1994.
22. 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.
23. P. S. Thiagarajan and I. Walukiewicz. An expressively complete linear time temporal logic for Mazurkiewicz traces. In *Proceedings of the 12th Annual IEEE Symposium on Logic in Computer Science (LICS'97)*, 1997.
24. I. Walukiewicz. Difficult configurations – on the complexity of LTrL. In Kim G. Larsen et al., editors, *Proceedings of the 25th International Colloquium on Automata, Languages and Programming (ICALP'98), Aalborg (Denmark) 1998*, number 1443 in Lecture Notes in Computer Science, pages 140–151, Berlin-Heidelberg-New York, 1998. Springer.
25. Th. Wilke. Classifying discrete temporal properties. Habilitationsschrift (post-doctoral thesis), April 1998. The main results will be presented in an Invited Lecture at STACS in Trier. Proc. of STACS'99. Lect. Notes in Comp. Sci. To appear.