A Pure Future Local Temporal Logic Beyond Cograph-Monoids

Volker Diekert*

Institut für Formale Methoden der Informatik
Universität Stuttgart, Breitwiesenstr. 20-22, D-70565 Stuttgart
diekert@informatik.uni-stuttgart.de

Keywords Temporal logics, Mazurkiewicz traces, pomsets.

Introduction

Temporal logics defined for labeled partial orders (pomsets) come in two kinds. The evaluation of a formula can be defined at cuts or, locally at vertices. For suitable logics both approaches remain within first-order definability, but it is a difficult task for a given temporal logic to find a large and natural class of pomsets where the logic is expressively complete, i.e., where every first-order property can be expressed. Kamp's Theorem says that the pure future fragment of linear temporal logic LTL is expressively complete for finite and infinite words [9, 7]. This result has been extended in [3] to Mazurkiewicz traces when formulae are evaluated at cuts, see also [12] for a related result. However, evaluation at cuts has a price. The satisfiability problem becomes non-elementary, as shown in [13]. This is one of the motivations to consider a local temporal logic: we are interested in a logic where the satisfiability problem is in PSPACE. The problem is that we do not know whether we lose expressive power. When we restrict ourselves to a linear temporal logic with evaluation at vertices and the future modalities next and until, only, then the best result which has been published so far covers cograph monoids, [2].

Cograph monoids are built up from free monoids by taking direct and free products. We obtain free monoids Σ^*, direct products of free monoids like $\mathbb{N}^k$, $\Sigma^* \times \mathbb{N}^k$, or nested objects like $((\mathbb{N}^{k_1} * \mathbb{N}^{k_2}) \times \Sigma_1^*) * (\Sigma_2^* \times \mathbb{N})$. The elements of cograph monoids are traces which have a representation as series parallel posets. It turns out that the linear temporal logic LTL is expressively complete for cograph monoids and that the satisfiability

* Partially supported by PROCOPE project D/9910318 (MoVe)

problem of LTL is in PSPACE for cograph monoids. This is the best we can expect since the satisfiability problem of LTL is PSPACE complete for words.

In this paper we define $\mathrm{LTL}_\sharp$-definable languages and we show that we can use $\mathrm{LTL}_\sharp$ for a class of trace monoids which is strictly larger than the class of cograph monoids. We restrict ourselves to finitary languages in order to focus on the essential ideas. However, since the monoid structure is not always used, we develop the concepts in terms of labeled partial orders (pomsets).[1]

1 Aperiodic languages

For every monoid M there is a notion of *aperiodic language.* A subset $L \subseteq M$ is called *aperiodic*, if there is a homomorphism $h : M \to S$ to some finite aperiodic monoid S such that $L = h^{-1}h(L)$. Our main interest concerns trace monoids, i.e., finitely generated free partially commutative monoids [4]. For free monoids Σ^* or, more generally for trace monoids, it is well-known that a language L is first-order definable if and only if it is aperiodic [11, 10, 8, 5, 6]. Therefore we concentrate on aperiodic languages and the challenge is to define a suitable temporal logic TL which is first-order definable and which is rich enough to specify all aperiodic languages of M. We say that TL is *expressively complete for M* in this case.

If $M = \mathbb{N}$ is the monoid of natural numbers with addition, then the aperiodic languages are just the finite or cofinite subsets. So, the task is trivial for $\mathbb{N}$.

One way to see Kamp's result on words is to view it as a closure property: If LTL is expressively complete for monoids M_0 and M_1, then it is for the free product $M = M_0 * M_1$. If a temporal logic TL is properly defined, then we may add another (rather trivial) closure property: If TL is expressively complete for monoids M_0 and M_1, then it is for the direct product $M_0 \times M_1$. The reason why it works nicely with aperiodic languages is as follows: Let $h : M_0 \times M_1 \to S$ be a homomorphism to some finite (aperiodic) monoid S. Define $h_i : M_i \to S$ by $h_0(u) = h(u, 1)$ and $h_1(v) = h(1, v)$. Then $h^{-1}(s)$ is a finite union of languages of the form $h_0^{-1}(s_0) \times h_1^{-1}(s_1)$ for each $s \in S$. Indeed, it is enough to consider all pairs $(s_0, s_1) \in S \times S$ with $s_0 s_1 = s$ in S. This observation is closely related to what is known as Mezei's Theorem.

[1] The work in this note is based on a collaboration with Paul Gastin. The techniques here are very close to those of [2].

So both closure properties lead to an expressively complete temporal logic for cograph monoids. By well-known closure properties of cographs [1], a trace monoid is not a cograph monoid if it contains the trace monoid $M(P_4) = \{a, b, c, d\}^*/\{ac = ca, ad = da, bd = db\}$ as a submonoid. (This means the independence relation contains a P_4 in the graph theoretical sense, i.e., a path of four vertices, hence the name.)

So, if our favorite temporal logic TL captures finite and cofinite sets as well as all aperiodic sets of $M(P_4)$ and if moreover we can show closure properties for direct and free products, then we are already beyond Kamp's Theorem for cograph monoids. This is the purpose of the logic $\mathrm{LTL}_\sharp$ defined below.

2 Pomsets

For set theoretical convenience let us fix some alphabet Γ of large enough cardinality which can be viewed as our universe. By Σ we always mean a finite subset of Γ. A *pomset* is defined here to be a finite partial order $(V, \leq)$ together with a labeling function $\lambda : V \to \Sigma$. Strictly speaking a pomset t is an isomorphism class of a labeled partial order, $t = [V, \leq, \lambda]$. By $\lessdot$ we denote the direct successor relation in the Hasse diagram of t. Thus, $x \lessdot z$, if $x \leq y \leq z$ implies either $x = y$ or $y = z$, but not both. The empty pomset is denoted by 1 and the set of all pomsets by $\mathbb{P}$. The set of all pomsets $[V, \leq, \lambda]$ with $\lambda(V) \subseteq \Sigma$ is called $\mathbb{P}(\Sigma)$, by $\mathbb{P}^+(\Sigma)$ we mean $\mathbb{P}(\Sigma) \setminus \{1\}$. The sets $\mathbb{P}$ and $\mathbb{P}(\Sigma)$ are monoids by taking the complex product: $[V_1, \leq_1, \lambda_1] \cdot [V_2, \leq_2, \lambda_2] = [V, \leq, \lambda]$ where V is the disjoint union of V_1 and V_2, $\lambda = \lambda_1 \cup \lambda_2$ and $\leq$ is the transitive closure of the relation $\leq_1 \cup \leq_2 \cup V_1 \times V_2$. The empty pomset 1 is the unit element and $\mathbb{P}(\Sigma)$ is a submonoid of $\mathbb{P}$.

If $a \in \Gamma$ is a letter, then we also view a as a pomset consisting of one vertex labeled by a. Hence, for $a \in \Gamma$, $t \in \mathbb{P}$, the pomset $a \cdot t$ is a pomset with a single minimal vertex. In particular, the free monoid Σ^* becomes a submonoid of $\mathbb{P}(\Sigma)$.

Let $t = [V, \leq, \lambda] \in \mathbb{P}$ be a pomset. The set of minimal elements is denoted by $\min(t)$, by $\mathrm{Min}(t)$ we mean $\lambda(\min(t))$. Hence $\min(t) \subseteq V$ and $\mathrm{Min}(t) \subseteq \Sigma$. For $a \in \Gamma$ and $t \in \mathbb{P}$ we have $\mathrm{Min}(a \cdot t) = \{a\}$. By abuse of language, if $t = [V, \leq, \lambda]$ and $x \in V$, then we also write $x \in t$. For $s = a \cdot t$ the letter a means also the unique minimal vertex of s. For $A \subseteq \Sigma$ we denote by $(\mathrm{Min} \subseteq A)$ the set $\{t \in \mathbb{P} \mid \mathrm{Min}(t) \subseteq A\}$, finally we let $(\mathrm{Min} = A)$ the set $\{t \in \mathbb{P} \mid \mathrm{Min}(t) = A\}$.

3 Linear temporal logic

The logic $\mathrm{LTL}(\Sigma)$ is given by the following syntax:

$$\varphi ::= \bot \mid a \in \Sigma \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi.$$

The symbol $\bot$ means *false*, $\mathsf{X}\,\varphi$ claims existentially that φ holds for some immediate successor of the current vertex; $\varphi\,\mathsf{U}\,\psi$ is an until claiming that ψ holds for some vertex in the future of the current one and *universally* that φ holds for all vertices in between. More precisely, let $t = [V, \leq, \lambda] \in \mathbb{P}$ and let $x \in t$. The semantics is inductively given as follows:

$$\begin{array}{lll}
t, x \models a & \text{if} & \lambda(x) = a \\
t, x \models \neg\varphi & \text{if} & t, x \not\models \varphi \\
t, x \models \varphi \vee \psi & \text{if} & t, x \models \varphi \ \vee \ t, x \models \psi \\
t, x \models \mathsf{X}\,\varphi & \text{if} & \exists y,\ x \lessdot y \ \&\ t, y \models \varphi \\
t, x \models \varphi\,\mathsf{U}\,\psi & \text{if} & \exists z,\ x \leq z \ \&\ t, z \models \psi \ \&\ \forall x \leq y < z,\ t, y \models \varphi
\end{array}$$

We define $\top = \neg\bot$, hence $\top$ means *true*. We derive some more operators from the above ones. *Eventually* (or *future*) φ claims the existence of some vertex where φ holds in the future of the current one: $\mathsf{F}\,\varphi = \top\,\mathsf{U}\,\varphi$. Its dual operator, *always* or *globally* φ, means that φ holds at all positions in the future of the current one: $\mathsf{G}\,\varphi = \neg\,\mathsf{F}\,\neg\varphi$. Finally, for $A \subseteq \Sigma$ we let $A \in \mathrm{LTL}(\Sigma)$ also denote the formula $A = \bigvee_{a \in A} a$.

Note that the semantics of the until operator does not use any path formula; and it can be defined in first-order logic. The question is how to define a language $L(\varphi) \subseteq \mathbb{P}$ for each $\varphi \in \mathrm{LTL}(\Sigma)$. If $t \in \mathbb{P}$ has a unique minimal vertex x_0, then it is rather clear that we should have $t \in L(\varphi)$ if and only if $t, x_0 \models \varphi$. But what do we do if t has no minimal vertex, i.e., $t = 1$, or more importantly, if t has several minimal vertices?

There are several choices to resolve this problem which may lead indeed to different language classes. Here we proceed as follows: Given $\varphi \in \mathrm{LTL}(\Sigma)$, choose some letter $c \in \Gamma \setminus \Sigma$ (so, c does not occur in the formula φ). Then define $L_c(\varphi) = \{t \in \mathbb{P} \mid c \cdot t, c \models \varphi\}$. This means φ is evaluated at the unique minimal vertex of the pomset $c \cdot t$. By induction on φ, we have $L_c(\varphi) = L_d(\varphi)$ for all $c, d \in \Gamma \setminus \Sigma$. Therefore, we denote $L_c(\varphi)$ rather by $L_\sharp(\varphi)$, where $\sharp$ is some special symbol not used otherwise.

Some sets are easy to express:

$$
\begin{aligned}
(\mathrm{Min} \subseteq A) &= \mathrm{L}_\sharp(\neg\, \mathsf{X}\, \neg A)),\\
(\mathrm{Min} = A) &= \mathrm{L}_\sharp(\neg\, \mathsf{X}\, \neg A \wedge \textstyle\bigwedge_{a \in A} \mathsf{X}\, a),\\
\mathbb{P}(\Sigma) &= \mathrm{L}_\sharp(\textstyle\bigvee_{A \subseteq \Sigma}(\neg\, \mathsf{X}\, \neg A \wedge \bigwedge_{a \in A} \mathsf{X}(a \wedge \mathsf{G}\, \Sigma))).
\end{aligned}
$$

For a subset $M \subseteq \mathbb{P}$ we denote by $\mathrm{LTL}_\sharp(M)$ the class of all languages $L \subseteq M$ where there is some $\varphi \in \mathrm{LTL}(\Sigma)$ such that $L = \mathrm{L}_\sharp(\varphi) \cap M$.

In [2] we have considered a slightly different syntax using *initial* formulae. In the framework here this means that we have considered a restricted class of formulae, where φ is a Boolean combination of $\mathsf{X}\,\psi$ formulae with $\psi \in \mathrm{LTL}(\Sigma)$. The reason is that on one hand for these formulae we can avoid the artificial introduction of a single minimal point and on the other hand the restricted class is still rich enough to express all first-order languages of cograph monoids. However, in the restricted class some languages cannot be expressed anymore. To give an idea why this holds let $M \subseteq \mathbb{P}(\Sigma)$ be the set of all pomsets without auto concurrency, i.e., vertices with the same label are always ordered. Consider $\Sigma = \{a, b, c, d\}$ and let

$$
L = \left\{ \begin{tikzcd} a \arrow[r] & b \arrow[r] & c \arrow[r] & b \arrow[r] & c \cdots b \arrow[r] & c \\ & d \arrow[ur] \end{tikzcd} \right\} \subseteq \mathbb{P}(\Sigma).
$$

Inside $(\mathrm{Min} = \{a, d\})$ a Boolean combination of $\mathsf{X}\,\psi$ formulae can be transformed into a disjunction of conjunctions of $\mathsf{X}\,\psi$ formulae. Now, if infinitely many $t \in L$ are in $\mathrm{L}_\sharp(\mathsf{X}\,\psi)$, then there are also infinitely many $s \in \mathrm{L}_\sharp(\mathsf{X}\,\psi)$ where s has the form

$$
\begin{tikzcd} & a \arrow[r] & b \arrow[r] & c \arrow[r] & b \arrow[r] & c \cdots b \arrow[r] & c \\ d \arrow[r] & c \arrow[ur] \end{tikzcd}
$$

On the other hand, we have $L \in \mathrm{LTL}_\sharp(M)$ by a formula which is a conjunction $\varphi \wedge \neg\, \mathsf{X}\, \neg\{a, d\} \wedge \mathsf{X}\,\psi_a \wedge \mathsf{X}\,\psi_d$ where $\varphi = \neg c\, \mathsf{U}\, b$ and for $e = a, d$ the formula ψ_e states e, and after e there is b (c resp.), globally after b there is c, and globally after c there is either b or nothing. This shows that $\mathrm{LTL}_\sharp(M)$ contains more languages than the class investigated in [2].

The following fact can be shown.

Proposition 1. *The class* $\mathrm{LTL}_\sharp(M(P_4))$ *is the class of all aperiodic languages of the trace monoid* $M(P_4)$.

In the following we say $\mathrm{LTL}_\sharp$ is *expressively complete* for a monoid $M \subseteq \mathbb{P}$, if all aperiodic languages $L \subseteq M$ can be represented as $\mathrm{L}_\sharp(\varphi) \cap M$ for some suitable $\varphi \in \mathrm{LTL}(\Sigma)$. Thus by the proposition above, $\mathrm{LTL}_\sharp$ is expressively complete for $M(P_4)$. The result we are interested in here can be stated as follows.

Theorem 1. *Let $M_0, M_1 \subseteq \mathbb{P}$ be submonoids such that the logic $\mathrm{LTL}_\sharp$ is expressively complete for both M_0 and M_1. Then $\mathrm{LTL}_\sharp$ is expressively complete for the direct product $M_0 \times M_1$ and for the free product $M_0 * M_1$.*

4 Definability of mappings

In the following M denotes a subset of $\mathbb{P}$. We shall prove Theorem 1 from a slightly more general viewpoint. Let $h : M \to S$ be any mapping to some finite set S. We say that h is definable in $\mathrm{LTL}_\sharp(M)$ if $h^{-1}(s) \in \mathrm{LTL}_\sharp(M)$ for all $s \in S$. For example, let $S' = 2^\Sigma$ be the power set of Σ and $S = S' \times S' \cup \{*\}$. Then define $h : \mathbb{P} \to S$ by $h(t) = (\mathrm{Min}(t), \lambda(V))$ if $t = [V, \leq, \lambda]$ with $\lambda(V) \subseteq \Sigma$ and $h(t) = *$ otherwise. Then h is definable in $\mathrm{LTL}_\sharp(M)$. When we work with two structures $M_0, M_1 \subseteq \mathbb{P}$, we shall assume that $M_i \subseteq \mathbb{P}(\Sigma_i)$, $i = 0, 1$ where $\Sigma_0 \cap \Sigma_1 = \emptyset$. This is always possible by some suitable relabeling.

4.1 Closure under direct products

Let $\Sigma = \Sigma_0 \cup \Sigma_1$ be a disjoint union, i.e., $\Sigma_0 \cap \Sigma_1 = \emptyset$. Assume that we have $M_i \subseteq \mathbb{P}(\Sigma_i)$ for $i = 0, 1$. Then, a pair $(u, v) \in M_0 \times M_1$ can be represented by the disjoint union of u and v, and vice versa: a pomset $t \in \mathbb{P}$ which is the disjoint union of pomsets $u \in M_0$ and $v \in M_1$ can be written as a pair $t = (u, v) \in M_0 \times M_1$. Thus, $M_0 \times M_1 \subseteq \mathbb{P}(\Sigma)$. Moreover, if $1 \in M_i$, then $M_{i-1} \subseteq M_0 \times M_1$ for $i = 0, 1$.

Remark 1. There is some $\varphi \in \mathrm{LTL}(\Sigma)$ such that $\mathrm{L}_\sharp(\varphi) = \mathbb{P}(\Sigma_0) \times \mathbb{P}(\Sigma_1)$.

Proof. For $i = 0, 1$ let $\psi \in \mathrm{LTL}(\Sigma)$ with $\mathrm{L}_\sharp(\psi) = \mathbb{P}(\Sigma)$. Define

$$\varphi = \psi \wedge \mathsf{G}\Big(\bigwedge_{i=0,1} (\mathsf{G}\, \Sigma_i \vee \neg \Sigma_i) \Big).$$

$\square$

Lemma 1. *For* $i = 0, 1$ *let* $\alpha_i \in \mathrm{LTL}(\Sigma)$ *with* $\mathrm{L}_\sharp(\alpha_i) \subseteq \mathbb{P}(\Sigma_i)$*. Then there is some* $\alpha \in \mathrm{LTL}(\Sigma)$ *with*

$$\mathrm{L}_\sharp(\alpha) = \mathrm{L}_\sharp(\alpha_0) \times \mathrm{L}_\sharp(\alpha_1) \subseteq \mathbb{P}(\Sigma).$$

Proof. Since $\Sigma_0 \cap \Sigma_1 = \emptyset$ and $\mathrm{L}_\sharp(\alpha_i) \subseteq \mathbb{P}(\Sigma_i)$, the set $\mathrm{L}_\sharp(\alpha_0) \times \mathrm{L}_\sharp(\alpha_1)$ is well-defined as a subset of $\mathbb{P}(\Sigma)$. Let $\varphi \in \mathrm{LTL}(\Sigma)$ from the remark above such that $\mathrm{L}_\sharp(\varphi) = \mathbb{P}(\Sigma_0) \times \mathbb{P}(\Sigma_1)$. By symmetry, it is enough to find $\widetilde{\alpha_0}$ such that

$$\mathrm{L}_\sharp(\widetilde{\alpha_0}) \cap \mathbb{P}(\Sigma_0) \times \mathbb{P}(\Sigma_1) = \mathrm{L}_\sharp(\alpha_0) \times \mathbb{P}(\Sigma_1).$$

(Indeed, then $\alpha = \widetilde{\alpha_0} \wedge \widetilde{\alpha_1} \wedge \varphi$ satisfies the assertion of this lemma.) We construct $\widetilde{\alpha_0}$ in such a way that for all $(u, v) \in \mathbb{P}(\Sigma_0) \times \mathbb{P}(\Sigma_1)$ and $x \in \# \cdot u$ we have

$$\# \cdot (u, v), x \models \widetilde{\alpha_0} \quad \text{if and only if} \quad \# \cdot u, x \models \alpha_0.$$

The construction is clear for Boolean operations and $a \in \Sigma$. Hence, let $\alpha_0 = \mathsf{X}\,\beta_0$. Then define $\widetilde{\alpha_0} = \mathsf{X}(\beta_0 \wedge \Sigma_0)$. For $\alpha_0 = \beta_0 \,\mathsf{U}\, \gamma_0$, define $\widetilde{\alpha_0} = (\widetilde{\beta_0} \wedge \neg\Sigma_1) \,\mathsf{U}\, (\widetilde{\gamma_0} \wedge \neg\Sigma_1)$. For the correctness observe that the set of vertices in $\# \cdot (u, v)$ satisfying Σ_0 is u, whereas $\neg\Sigma_1$ is true for the set of vertices in $\# \cdot u$. $\square$

Proposition 2. *For* $i = 0, 1$ *let* $h_i : M_i \to S_i$ *be definable in* $\mathrm{LTL}_\sharp(M_i)$*. Then the mapping* $h : M_0 \times M_1 \to S_0 \times S_1$*,* $h(u, v) = (h_0(u), h_1(v))$ *is definable in* $\mathrm{LTL}_\sharp(M_0 \times M_1)$*.*

Proof. Let $(s_0, s_1) \in S_0 \times S_1$ and $\alpha_i \in \mathrm{LTL}(\Sigma_i)$ such that $h_i^{-1}(s_i) = \mathrm{L}_\sharp(\alpha_i) \cap M_i$ for $i = 0, 1$. Take α as in the lemma above. $\square$

Corollary 1. *If* $\mathrm{LTL}_\sharp$ *is expressively complete for submonoids* $M_0, M_1 \subseteq \mathbb{P}$*, then it is expressively complete for the direct product* $M_0 \times M_1$*.*

4.2 Closure under free products

Again, let $\Sigma = \Sigma_0 \cup \Sigma_1$ with $\Sigma_0 \cap \Sigma_1 = \emptyset$ and $M_i \subseteq \mathbb{P}(\Sigma_i)$ for $i = 0, 1$. The free product M of M_0 and M_1 is defined by all sequences $(t_1, \ldots, t_n)$, $n \geq 0$, where with $t_0 = 1$ for all $i = 0, 1$ and $1 \leq j \leq n$ we have $t_{j-1} \in M_i$ implies $t_j \in M_{1-i}$. We identify $(t_1, \ldots, t_n)$ with the (complex) product $t_1 \cdots t_n$. Hence, we view $M \subseteq \mathbb{P}(\Sigma)$. If M_0, M_1 are submonoids of $\mathbb{P}$, then M is the usual free product $M_0 * M_1$.

Lemma 2. *Let* $\alpha \in \mathrm{LTL}(\Sigma)$ *and* $i \in \{0,1\}$. *Then there is a formula* $\alpha_i \in \mathrm{LTL}(\Sigma_i)$ *such that for all letters* $c \in \Gamma \setminus \Sigma_i$ *we have*

$$L_c(\alpha_i) \cap \mathbb{P}^+(\Sigma_i) \cdot (\mathrm{Min} \subseteq \Sigma_{1-i}) = (\mathrm{L}_\sharp(\alpha) \cap \mathbb{P}^+(\Sigma_i)) \cdot (\mathrm{Min} \subseteq \Sigma_{1-i}).$$

Proof. We may assume that $\alpha \in \mathrm{LTL}(\Sigma_i)$ and $\mathrm{L}_\sharp(\alpha) \subseteq \mathbb{P}^+(\Sigma_i)$. In particular, $\mathrm{L}_\sharp(\alpha) = L_c(\alpha)$ for all $c \in \Gamma \setminus \Sigma_i$. Consider some $c \in \Gamma \setminus \Sigma_i$, $t \in \mathbb{P}^+(\Sigma_i)$ and $s \in (\mathrm{Min} \subseteq \Sigma_{1-i})$. It is enough to construct $\alpha_i \in \mathrm{LTL}(\Sigma_i)$ such that for all vertices $x \in c \cdot t$ we have:

$$c \cdot t \cdot s, x \models \alpha_i \quad \text{if and only if} \quad c \cdot t, x \models \alpha.$$

Again, this is clear for Boolean operations. For $\alpha = a \in \Sigma_i$, there is nothing to do. For $\alpha = \mathsf{X}\,\beta$ let $\alpha_i = \mathsf{X}(\beta_i \wedge \Sigma_i)$. For $\alpha = \beta \mathsf{U} \gamma$ let $\alpha_i = (\beta_i \wedge \mathsf{X}\,\Sigma_i) \mathsf{U} \gamma_i$. For the correctness observe that $t \in \mathbb{P}^+(\Sigma_i)$. Hence, if $x \in c \cdot t$ is a vertex which is not a maximal vertex of t and if $z \in s$, then every path from x to z in $c \cdot t \cdot s$ passes through some maximal vertex y of t and then $\mathsf{X}\,\Sigma_i$ is not satisfied at y. $\square$

Proposition 3. *Let S be a finite aperiodic monoid and let $h_i : M_i \to S$ be definable in $\mathrm{LTL}_\sharp(M_i)$ for $i = 0, 1$. Define $h : M \to S$ by $h(t_1, \ldots, t_n) = h(t_1) \cdots h(t_n)$. Then h is definable in $\mathrm{LTL}_\sharp(M)$.*

Proof. Let $e : S^* \to S$ be the canonical homomorphism which evaluates a sequence, i.e., $e(s_1, \ldots, s_n) = s_1 \cdots s_n$. Define $f : M \to S^*$ by $f(t_1, \ldots, t_n) = (h(t_1), \ldots, h(t_n))$. Then $h = e \circ f$. Fix some $s \in S$. By Kamp's Theorem for words we have $L(\varphi) = e^{-1}(s) \cap S^+$ for some $\varphi \in \mathrm{LTL}(S)$, where for convenience $L(\varphi) = \mathrm{L}_\sharp(\mathsf{X}\,\varphi) \cap S^+$. We construct a formula $\alpha \in \mathrm{LTL}(\Sigma)$ such that for all $(t_1, \ldots, t_n) \in M$ with $n \geq 1$ we have: $(t_1, \ldots, t_n) \in \mathrm{L}_\sharp(\alpha)$ if and only if $(h(t_1), \ldots, h(t_n)) \in L(\varphi)$. For $i = 0, 1$ we construct a formula α_i such that for all $c \in \Gamma \setminus \Sigma_i$ we have: $(t_1, \ldots, t_n) \in L_c(\alpha_i)$ if and only if both $(h(t_1), \ldots, h(t_n)) \in L(\varphi)$ and $t_1 \in \mathbb{P}^+(\Sigma_i)$. We do this by induction on φ.

The construction is clear for Boolean operations. For $\varphi = s \in S$ we have to consider $t_1 \in h^{-1}(s) \cap (M_i \setminus \{1\}) = h_i^{-1}(s) \cap (M_i \setminus \{1\})$. The resulting formula α_i is given by the lemma above.

Since we are transforming LTL formulae on words, it is now enough to consider formulae of type $\mathsf{X}(\varphi \,\mathsf{U}\, \psi) \in \mathrm{LTL}(S)$. We have $(h(t_1), \ldots, h(t_n)) \in L(\mathsf{X}(\varphi \,\mathsf{U}\, \psi))$ if and only if there is $1 < k \leq n$ such that for all $1 < j < k$ we have both $(h(t_k), \ldots, h(t_n)) \in L(\psi)$ and $(h(t_j), \ldots, h(t_n)) \in L(\varphi)$. By induction, for $\ell = 0, 1$, there are $\beta_\ell, \gamma_\ell \in \mathrm{LTL}(\Sigma_\ell)$ such that for all

$c \in \Gamma \setminus \Sigma_\ell$ we have: $c \cdot t_k \cdots t_n, c \models \gamma_\ell$ if and only if both $f(t_k, \ldots, t_n) \in L(\psi)$ and $t_k \in \mathbb{P}^+(\Sigma_\ell)$. Analogously, $c \cdot t_j \cdots t_n, c \models \beta_\ell$ if and only if both $f(t_j, \ldots, t_n) \in L(\varphi)$ and $t_j \in \mathbb{P}^+(\Sigma_\ell)$. The resulting formula α_i is almost independent of i. We define:

$$\alpha_i = \mathsf{X}\left(\Sigma_i \wedge \left(\bigvee_{\ell=0,1} \Sigma_\ell \wedge (\mathsf{X}\,\Sigma_\ell \vee \beta_\ell)\right) \mathsf{U} \left(\bigvee_{\ell=0,1} \Sigma_\ell \wedge \neg\,\mathsf{X}\,\Sigma_\ell \wedge \gamma_\ell\right)\right)$$

□

Conclusion: We have seen that $\mathrm{LTL}_\sharp$ is a local and pure future temporal logic. The class of trace monoids where $\mathrm{LTL}_\sharp$ is expressively complete is strictly larger than the class of cograph monoids which has been studied in [2]. The challenging programme remains to see whether or not, $\mathrm{LTL}_\sharp$ is expressively complete for all trace monoids. More general, it would be interesting to study the expressive power of $\mathrm{LTL}_\sharp$ in other classes of pomsets.

References

1. A. Brandstädt, V. B. Le, and J. P. Spinrad. *Graph Classes: A Survey.* SIAM Monographs on Discrete Mathematics and Applications. 1999.
2. V. Diekert and P. Gastin. Local temporal logic is expressively complete for cograph dependence alphabets. In R. Nieuwenhuis and A. Voronkov, editors, *Proc. 8th International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR'01), Havanna (Cuba)*, number 2250 in Lecture Notes in Artificial Intelligence, pages 55–69. Springer, 2001.
3. V. Diekert and P. Gastin. LTL is expressively complete for Mazurkiewicz traces. *Journal of Computer and System Sciences*, 64:396–418, 2002.
4. V. Diekert and G. Rozenberg, editors. *The Book of Traces.* World Scientific, Singapore, 1995.
5. W. Ebinger. *Charakterisierung von Sprachklassen unendlicher Spuren durch Logiken.* Dissertation, Institut für Informatik, Universität Stuttgart, 1994.
6. W. Ebinger and A. Muscholl. Logical definability on infinite traces. *Theoretical Computer Science*, 154:67–84, 1996.
7. 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 (POPL'80), Las Vegas (Nevada), 1980*, pages 163–173, 1980.
8. G. Guaiana, A. Restivo, and S. Salemi. Star-free trace languages. *Theoretical Computer Science*, 97:301–311, 1992.

9. J. A. W. Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, Los Angeles (California), 1968.
10. R. McNaughton and S. Papert. *Counter-Free Automata*. The MIT Press, Cambridge, Mass., 1971.
11. M. P. Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8:190–194, 1965.
12. P. S. Thiagarajan and I. Walukiewicz. An expressively complete linear time temporal logic for Mazurkiewicz traces. In *Proc. 12th Annual IEEE Symposium on Logic in Computer Science (LICS'97), Warsaw (Poland)*, pages 183–194, 1997.
13. I. Walukiewicz. Difficult configurations – on the complexity of LTrL. In K. G. Larsen et al., editors, *Proc. 25th International Colloquium on Automata, Languages and Programming (ICALP'98), Aalborg (Denmark)*, number 1443 in Lecture Notes in Computer Science, pages 140–151, Berlin-Heidelberg-New York, 1998. Springer.