Appeared in: Computer Science Logic 1995, LNCS vol. 1092, Springer, 1996, 233-251.

Languages and Logical Definability in Concurrency Monoids*

Manfred Droste and Dietrich Kuske

Abteilung Mathematik, Technische Universität Dresden, D–01062 Dresden, Germany,
E-mail: {droste,kuske}@math.tu-dresden.de

Abstract. Automata with concurrency relations $\mathcal{A}$ are labeled transition systems with a collection of binary relations describing when two actions in a given state of the automaton can occur independently of each other. The concurrency monoid $\mathrm{M}(\mathcal{A})$ comprises all finite computation sequences of $\mathcal{A}$, modulo a canonical congruence induced by the concurrency relations, with composition as monoid operation; its elements can be represented by labeled partially ordered sets. Under suitable assumptions on $\mathcal{A}$, we show that a language L in $\mathrm{M}(\mathcal{A})$ is recognizable iff it is definable by a formula of monadic second order logic. We also investigate the relationship between aperiodic and first-order definable languages in $\mathrm{M}(\mathcal{A})$. This generalizes various recent results in trace theory.

1 Introduction

In the literature, classical logical definability results of recognizable word languages (Büchi [Bü60]) and aperiodic or starfree languages (McNaughton and Papert [MP71]) have been generalized in various directions, including tree languages (*cf., e.g.,* [T90a]) and, recently, languages in trace monoids. For the latter, in particular, it was shown that a trace language is recognizable iff it is definable by a sentence of monadic second order logic ([T90b]), and it is aperiodic iff it is starfree ([GRS92]) iff it is definable by a first-order sentence ([T90b], *cf.* [EM93]). It is the aim of this paper to extend these results to large classes of even more general monoids, called concurrency monoids.

Trace theory (*cf.* [AR88, Di90, DR95] for surveys) provides a mathematical model for the sequential behaviour of a parallel system in which the order of two independent actions is regarded as irrelevant; one considers a set E of actions together with a single binary relation representing the concurrency of two actions. Here, we will consider a more general model consisting of labeled transition systems in which the concurrency relation depends not only on the two arriving actions, but also on the present state of the system. An *automaton with concurrency relations* is a tuple $\mathcal{A} = (Q, E, T, \|)$ where Q is the set of states, E as before the set of events or actions, $T \subseteq Q \times E \times Q$ the transition relation (assumed deterministic), and $\| = (\|_q)_{q \in Q}$ is a collection of concurrency relations $\|_q$ for E, indexed by the possible states $q \in Q$. Similarly as in trace theory, we declare two sequences $(p, a, q)(q, b, r)$ and $(p, b, q')(q', a, r)$ equivalent, if $a \|_p b$. This induces

* Research supported by the German Research Foundation (DFG)

a congruence $\sim$ on the set $\mathrm{CS}(\mathcal{A})$ of all finite computation sequences; thus, intuitively, two computation sequences are equivalent, if they represent "interleaved views" of a single computation. The quotient $\mathrm{M}(\mathcal{A}) = \mathrm{CS}(\mathcal{A})/{\sim} \cup \{0\}$ is called the *concurrency monoid associated with* $\mathcal{A}$.

Automata with concurrency relations were introduced and studied in [Dr90, Dr92, BD93, BD94]. Their domains of computation sequences are closely related with event domains and dI–domains arising in denotational semantics of programming languages. These automata also generalize asynchronous transition systems ([Sh85, Be87, WN94]). Very recently, a formalization using several independence relations of the operational semantics of Occam was given in [BR94].

Generalizing results of [Oc85, GRS92] in trace theory, a Kleene-type characterization of the recognizable languages in concurrency monoids $\mathrm{M}(\mathcal{A})$ has been given in [Dr94a], and aperiodic and starfree languages were investigated in [Dr94b]. In [BDK96, BDK95], it was shown how to represent elements of $\mathrm{M}(\mathcal{A})$ by certain labeled graphs, or labeled partially ordered sets. There is a multiplication of (isomorphism classes of) such graphs, yielding a monoid which turns out to be isomorphic to the concurrency monoid $\mathrm{M}(\mathcal{A})$. Therefore here we will identify elements of $\mathrm{M}(\mathcal{A})$ with their graph-theoretical representation. The partial order defined on the enumerated occurrences of actions in a computation denotes that a "smaller" event has to occur before a "larger" one, or, in other words, is a necessary condition for the larger event. Here, we introduce logical languages that allow us to make statements on causal dependencies of events and on the initial state of a computation. Because of the representation of computations we can use the canonical logical languages for labeled partial orders. The satisfaction relation is defined via the representation of a computation by a labeled partially ordered set. We let MSO be the corresponding monadic second order language. For the subsequent results, a useful (and almost necessary) assumption is that $\mathcal{A}$ is stably concurrent (see Def. 4). This means that the concurrency relations of $\mathcal{A}$ depend locally (but not globally) on each other. We will show

Theorem 1. *Let $\mathcal{A}$ be a finite stably concurrent automaton and $L \subseteq \mathrm{M}(\mathcal{A})$. Then L is recognizable if and only if it is definable in* MSO*.*

Next we turn to aperiodic, starfree and first-order definable languages in $\mathrm{M}(\mathcal{A})$. Aperiodic languages are starfree, and the converse was derived in [Dr94b] for a large class of stably concurrent automata $\mathcal{A}$ (it fails in general). Here, we obtain the following; for undefined notions we refer the reader to Sect. 5.

Theorem 2. *Let $\mathcal{A}$ be a finite stably concurrent automaton.*
(a) If $\mathcal{A}$ is counter free, then each aperiodic language in $\mathrm{M}(\mathcal{A})$ *is first-order definable.*
(b) If $\mathcal{A}$ has no commuting loops or is an automaton with global independence, then each first-order definable language in $\mathrm{M}(\mathcal{A})$ *is aperiodic.*

Examples show that neither in (a) nor in (b) the additional assumptions on $\mathcal{A}$ can be omitted. However, there are large classes of stably concurrent automata satisfying these various assumptions, or even their conjunctions, and these theorems contain the corresponding results from trace theory as a special case. Counter free automata have been well studied in the literature, *cf.* [MP71]. We note that in an automaton with global independence the concurrency of actions can be described by a single binary relation. These automata are only slight variants of the full trace automata and the asynchronous transition systems of Bednarczyk ([Be87]) and Shields ([Sh85]) and generalize trace alphabets; also, see *e.g.* [S89a, S89b, BCS93, WN94] for further results on this well-studied class of automata. Even in this class, there are proper inclusions between the classes of first-order definable and aperiodic, and between aperiodic and starfree languages. As shown in [Dr94b], in an automaton without commuting loops the aperiodic languages are precisely the starfree ones. However, these automata are complementary to trace alphabets.

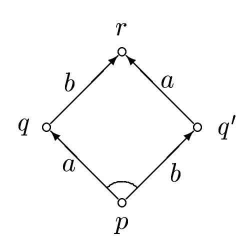
The proofs of Thms. 1 and 2 rest on a detailed analysis of the partial order structure of the representation of elements of $\mathrm{M}(\mathcal{A})$ as well as on Büchi's and McNaughton and Papert's classical results for (finite) words; the use of particular accepting devices sometimes employed in the trace theoretic setting (*e.g.* asynchronous automata in [T90b]) is avoided. An extension of the present results to monadically definable languages of infinite computations of stably concurrent automata has recently been achieved in [DK96].

2 Preliminaries

Definition 3. An *automaton with concurrency relations* is a quadruple $\mathcal{A} = (Q, E, T, \|)$ such that

1. Q and E are finite sets of *states* and *events* or *actions*, respectively.
2. $T \subseteq Q \times E \times Q$ is a set of *transitions* assumed deterministic, *i.e.* whenever $(p, a, q), (p, a, r) \in T$, then $q = r$.
3. $\| = (\|_q)_{q \in Q}$ is a collection of irreflexive, symmetric binary relations on E; it is required that whenever $a \|_p b$, then there exist transitions (p, a, q), (p, b, q'), (q, b, r) and (q', a, r) in T.

Note that we consider only *finite* automata $\mathcal{A}$. A transition $\tau = (p, a, q)$ intuitively represents a potential computation step in which event a happens in state p of $\mathcal{A}$ and $\mathcal{A}$ changes from state p to state q. We write $\mathrm{ev}(\tau) = a$, the event of τ. The concurrency relations $\|_p$ describe the concurrency information for pairs of events at state p. The last requirement can be seen as in the diagram:



The angle at p indicates that $a \parallel_p b$ holds.

A *computation sequence* in $\mathcal{A}$ is either empty (denoted by ϵ), or a finite sequence $\gamma = \sigma_1 \ldots \sigma_n$ of transitions σ_i of the form $\sigma_i = (q_{i-1}, a_i, q_i)$ for $i = 1 \ldots n$; it can be depicted as

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} q_n.$$

We call q_0 the *domain* of γ, denoted by $\operatorname{dom}\gamma$, q_n the *codomain*, denoted by $\operatorname{cod}\gamma$, and n the length $|\gamma|$ of γ. The sequence $a_1a_2 \ldots a_n$ is called *event sequence of* γ, denoted by $\operatorname{evseq}\gamma$. Also, for $w \in E^\star$ and $q \in Q$, we write $q.w = r$ if there exists a computation sequence γ with $\operatorname{dom}\gamma = q$, $\operatorname{evseq}\gamma = w$ and $\operatorname{cod}\gamma = r$. Otherwise, $q.w$ is undefined. We let $\mathrm{CS}(\mathcal{A})$ denote the set of all computation sequences of $\mathcal{A}$. The *composition* $\gamma\delta$ is defined in the natural way by concatenating γ and δ if $\operatorname{cod}\gamma = \operatorname{dom}\delta$. Formally we put $\gamma\epsilon = \epsilon\gamma = \gamma$.

Now we want the concurrency relations of $\mathcal{A}$ to induce an equivalence relation on $\mathrm{CS}(\mathcal{A})$ so that equivalent computation sequences are not differentiated by the order in which concurrent events appear. For this, we let $\sim$ be the smallest congruence with respect to composition on $\mathrm{CS}(\mathcal{A})$ making all computation sequences $p \xrightarrow{a} q \xrightarrow{b} r$ and $p \xrightarrow{b} q' \xrightarrow{a} r$ with $a \parallel_p b$ equivalent. We let $[\gamma]$ denote the equivalence class of γ with respect to $\sim$. Also, we let $1 := [\epsilon]$. Defining $\mathrm{M}(\mathcal{A}) = \mathrm{CS}(\mathcal{A})/{\sim} \cup \{0\}$, we now obtain the *monoid* $\mathrm{M}(\mathcal{A})$ *of computations associated with* $\mathcal{A}$, where 0 is an additional symbol. That is, for $\gamma, \delta \in \mathrm{CS}(\mathcal{A})$ we have $[\gamma] \cdot [\delta] = [\gamma\delta]$ if $\operatorname{cod}\gamma = \operatorname{dom}\delta$ and $[\gamma] \cdot [\delta] = 0$ otherwise. Also, $x \cdot 0 = 0 \cdot x = 0$ and $x \cdot 1 = 1 \cdot x = x$ for any $x \in \mathrm{M}(\mathcal{A})$. Clearly, with this operation $\mathrm{M}(\mathcal{A})$ is a monoid with 1 as unit (and with 0 as zero).

Next we show why these automata and their monoids provide a generalization of trace alphabets and trace monoids. A trace alphabet [Ma77, Ma86, Ma88] is a pair (E, D) where E is a nonempty finite set and $D \subseteq E \times E$ a reflexive, symmetric dependence relations on E. The congruence with respect to (E, D) on the free monoid $E^\star$ is the smallest congruence $\sim$ that identifies ab and ba whenever $(a, b) \notin D$. Then $\mathcal{A} = (Q, E, T, \|)$ with $Q = \{q\}$, $T = Q \times E \times Q$ and $\parallel_q = (E \times E) \setminus D$ is an automaton with concurrency relations. This automaton will be called *automaton induced by* (E, D). Obviously, $\operatorname{evseq} : \mathrm{CS}(\mathcal{A}) \longrightarrow E^\star$ is a bijection. Moreover, for $\gamma, \delta \in \mathrm{CS}(\mathcal{A})$ we have $\gamma \sim \delta$ iff $\operatorname{evseq}\gamma \sim \operatorname{evseq}\delta$ with respect to (E, D). Hence, evseq induces a monoid-isomorphism from $\mathrm{M}(\mathcal{A}) \setminus \{0\}$ onto $M(E, D)$. Thus, automata with concurrency relations generalize trace alphabets.

Here, we will investigate recognizable and aperiodic languages in $\mathrm{M}(\mathcal{A})$. A language $L \subseteq \mathrm{M}(\mathcal{A})$ is *recognizable* if there exists a finite $\mathrm{M}(\mathcal{A})$-automaton that recognizes L, or, equivalently, if there are only finitely many sets $x^{-1}L := \{y \in \mathrm{M}(\mathcal{A}) \mid x \cdot y \in L\}$ $(x \in \mathrm{M}(\mathcal{A}))$. Furthermore, a recognizable language $L \subseteq \mathrm{M}(\mathcal{A})$ is *aperiodic* if there exists $n \in \mathbb{N}$ such that $x \cdot y^n \cdot z \in L$ iff $x \cdot y^{n+1} \cdot z \in L$ for any $x, y, z \in \mathrm{M}(\mathcal{A})$. The smallest natural number n that meets this requirement is the *index* of L.

Now we recall basic definitions, constructions and results from [BDK95, BDK96].

Let $\mathcal{A}$ be an automaton with concurrency relations, and let $\gamma \in \mathrm{CS}(\mathcal{A})$. Analogously to trace theory we define a dependence order on those events that appear in γ. This order should reflect that a "smaller" event has to appear before a "larger" one, i.e. the "smaller" event is a necessary condition for the "larger" one. If two events are incomparable they can appear in any order or even in parallel. Since an event a can appear several times in γ we have to distinguish between the first, the second, ... appearance of a. For $a \in E$ let $|\gamma|_a$ denote the number of transitions σ in γ with $\mathrm{ev}\,\sigma = a$, i.e. the number of a's in the word $\mathrm{evseq}\,\gamma \in E^\star$. We abbreviate $a^i = (a, i)$ for $a \in E$ and $i \in \mathbb{N}$. The precise definition of the dependence order of γ can now be given as follows. Let $\mathrm{O}(\gamma) = \{a^i \mid a \in \mathrm{ev}\,\gamma,\ 1 \le i \le |\gamma|_a\}$. Then, obviously, $|\mathrm{O}(\gamma)| = |\gamma|$. For $a^i, b^j \in \mathrm{O}(\gamma)$ let $a^i \sqsubseteq_\gamma b^j$ iff the i–th appearance of a in γ occurs before the j–th appearance of b, i.e., formally, there are words $u, v, w \in E^\star$ with $\mathrm{evseq}\,\gamma = uavbw$, $|u|_a = i - 1$ and $|uav|_b = j - 1$. Then $\sqsubseteq_\gamma$ is a linear order on $\mathrm{O}(\gamma)$. Since for equivalent computation sequences γ and δ we always have $\mathrm{O}(\gamma) = \mathrm{O}(\delta)$, a partial order on $\mathrm{O}(\gamma)$ can be defined by:

$$\sqsubseteq := \bigcap \{\sqsubseteq_\delta \mid \delta \sim \gamma\}.$$

Hence, $a^i \sqsubseteq b^j$ if and only if the i–th a appears before the j–th b in *any* computation sequence equivalent with γ. For $a \in E$, let E_a comprise all elements of $\mathrm{O}(\gamma)$ of the form a^i. Then $\mathrm{DO}(\gamma) = (\mathrm{O}(\gamma), \sqsubseteq, (E_a)_{a \in E}, \mathrm{dom}\,\gamma)$ is a relational structure with one constant from Q. We call $\mathrm{DO}(\gamma)$ the *dependence order associated with* γ. A sequence $A = (x_1, x_2, \ldots, x_n)$ with $x_i \in \mathrm{O}(\gamma)$ is an *order-preserving enumeration of* $\mathrm{DO}(\gamma)$ if it is an enumeration of $\mathrm{O}(\gamma)$ and $x_i \sqsubseteq x_j$ implies $i \le j$. Then a computation sequence δ is the *linearisation of* $\mathrm{DO}(\gamma)$ *induced by* A if $\mathrm{dom}\,\gamma = \mathrm{dom}\,\delta$ and $\mathrm{evseq}\,\delta = a_1 a_2 \ldots a_n$ with $x_i \in E_{a_i}$ for $i = 1, 2, \ldots, n$. There may exist order-preserving enumerations of $\mathrm{DO}(\gamma)$ that do not induce any linearisation. But, since $\mathcal{A}$ is deterministic, any order-preserving enumeration induces at most one linearisation. We call δ *a linearisation of* $\mathrm{DO}(\gamma)$ if it is the linearisation induced by some order-preserving enumeration. Let $\mathrm{Lin}\,\mathrm{DO}(\gamma)$ comprise all linearisations of $\mathrm{DO}(\gamma)$. Then it is easy to see that $[\gamma] \subseteq \mathrm{Lin}\,\mathrm{DO}(\gamma)$ for any automaton with concurrency relations (*cf.* [BDK96, BDK95]). To prove the converse of this inclusion, we need further assumptions on the underlying automaton:

Definition 4. Let $\mathcal{A}$ be an automaton with concurrency relations. $\mathcal{A}$ is called *stably concurrent automaton*, if for all $q \in Q$ and all $a, b, c \in E$ the following equivalence holds:

$$a \parallel_q b,\ b \parallel_q c \text{ and } a \parallel_{q.b} c \iff a \parallel_q c,\ b \parallel_{q.a} c \text{ and } a \parallel_{q.c} b$$

The equivalence in this definition is depicted by Fig. 1.

The requirement of the implication "⇒" has also been called cube axiom, and the implication "⇐" is called the inverse cube axiom. In various forms, these axioms arose several times in the literature, see *e.g.* [S89a, PSS90, Dr90, Dr92, Ku94a, Ku94b, Dr94a, Dr94b]. In [Dr94a, Dr94b] the recognizable and the aperiodic languages in $\mathrm{M}(\mathcal{A})$ for $\mathcal{A}$ stably concurrent have been characterized. In [Ku94b] it has been shown that stably concurrent automata precisely generate

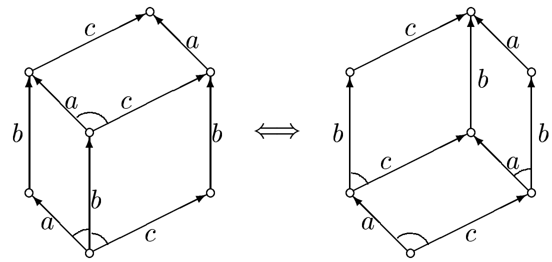


Fig. 1

dI-domains [Cu93] via their domains of finite and infinite computations. This distributivity is the basis for the proofs in [BDK96, BDK95]

Observe that the automaton induced by a trace alphabet is always stably concurrent.

Let $(P, \leq)$ be a partially ordered set and $x \in P$. Then $x\!\uparrow$ denotes the subset of P comprising all elements y with $x \leq y$. Similarly, $x\!\downarrow = \{y \in P \mid y \leq x\}$. A subset $M \subseteq P$ is *downward closed* iff $x\!\downarrow \subseteq M$ for each $x \in M$.

The following proposition is central in the proofs in [BDK96, BDK95] and will be used here, too. Let $\mathcal{A}$ be an automaton with concurrency relations, $\gamma \in \mathrm{CS}(\mathcal{A})$ and $M = \{a_1^{n_1}, a_2^{n_2}, \ldots, a_k^{n_k}\} \subseteq \mathrm{O}(\gamma)$ with $a_i^{n_i} \sqsubseteq_\gamma (a_{i+1})^{n_{i+1}}$. Then $\gamma(M)$ denotes the word $a_1 a_2 \ldots a_k \in E^\star$.

Proposition 5 [BDK96]. *Let $\mathcal{A}$ be a stably concurrent automaton and $\gamma \in \mathrm{CS}(\mathcal{A})$. Let $M \subseteq \mathrm{O}(\gamma)$ be downward closed in $(\mathrm{O}(\gamma), \sqsubseteq)$. Then $q = (\mathrm{dom}\,\gamma).\gamma(M)$ is defined. Suppose furthermore $x, y \in \mathrm{O}(\gamma)$ with $x \in E_a$, $y \in E_b$ and $(x\!\downarrow \cup y\!\downarrow) \setminus \{x, y\} \subseteq M \subseteq \mathrm{DO}(\gamma) \setminus (x\!\uparrow \cup y\!\uparrow)$ (where $\downarrow$ and $\uparrow$ are to be understood in $(\mathrm{O}(\gamma), \sqsubseteq)$). Then x and y are incomparable with respect to $\sqsubseteq$ if and only if $a \parallel_q b$.*

In [BDK95] the following is proved.

Theorem 6 [BDK95]. *Let $\mathcal{A}$ be a stably concurrent automaton and $\gamma \in \mathrm{CS}(\mathcal{A})$. Then $[\gamma] = \mathrm{Lin}(\mathrm{DO}(\gamma))$. Furthermore, any order-preserving enumeration of $\mathrm{DO}(\gamma)$ induces a linearisation of $\mathrm{DO}(\gamma)$.*

Hence, we have the following equivalences for a stably concurrent automaton $\mathcal{A}$ and computation sequences $\gamma, \delta \in \mathrm{CS}(\mathcal{A})$:

$$\mathrm{DO}(\gamma) = \mathrm{DO}(\delta) \iff \gamma \sim \delta \iff \delta \in \mathrm{Lin}(\mathrm{DO}(\gamma)).$$

This result enables us to represent computations, *i.e.* elements of $\mathrm{M}(\mathcal{A}) \setminus \{0, 1\}$, by dependence orders. To include 0 and 1, formally we put $\mathrm{dom}\, 0 = \bot$ and $\mathrm{dom}\, 1 = \top$ where $\bot$ and $\top$ are additional symbols not belonging to Q, and, using this, define $\mathrm{DO}(0)$ and $\mathrm{DO}(1)$ similarly as before. Since dependence orders are relational structures, we can define a logical language to describe properties of these dependence orders. The corresponding first-order language FO has variables $x, y, \ldots$ for elements of $\mathrm{O}(\gamma)$. The atomic formulas are $x \leq y$, $E_a(x)$ for

$a \in E$, and constants D_q for $q \in Q \cup \{\bot, \top\}$. Then formulas are built up from atomic formulas by the connectives $\neg$ and $\vee$ and the quantor $\exists$. In the monadic second-order language, also set variables $X, Y, \ldots$, quantification of them and atomic formulas $X(x)$ are admitted. Also, we will use several abbreviations like $x < y$, $x = y$, $\phi \to \psi$, $\forall x\, \phi$ etc. with the usual interpretation. Additionally, $x \parallel y$ stands for $\neg(x \leq y) \wedge \neg(y \leq x)$. This monadic second-order logic is denoted by MSO. A sentence of MSO is a formula without free variables. The satisfaction relation $\mathrm{DO}(\gamma) \models \phi$ between dependence orders and sentences is defined as usually: $x \leq y$ is satisfied iff $x \sqsubseteq y$, $E_a(x)$ iff x is of the form a^i, D_q iff $\operatorname{dom} \gamma = q$ and $X(x)$ iff $x \in X$. Now let ϕ be a sentence of MSO. Then $L(\phi)$ denotes the set of all $[\gamma] \in \mathrm{M}(\mathcal{A})$ such that $\mathrm{DO}(\gamma) \models \phi$. Since $\gamma \sim \delta$ implies $\mathrm{DO}(\gamma) = \mathrm{DO}(\delta)$, the language $L(\phi)$ is welldefined. For a language $L \subseteq \mathrm{M}(\mathcal{A})$ and a sentence ϕ of MSO, we say ϕ *defines* L if $L = L(\phi)$. The language L is *definable in* MSO if there exists a sentence ϕ in MSO that defines L. In the following two sections, we will prove the two implications of

Theorem 1. *Let $\mathcal{A}$ be a finite stably concurrent automaton and $L \subseteq \mathrm{M}(\mathcal{A})$. Then L is recognizable if and only if it is definable in* MSO.

Suppose, $\mathcal{A}$ is the automaton induced by the trace alphabet (E, D). Then the monoids $\mathrm{M}(\mathcal{A}) \setminus \{0\}$ and $M(E, D)$ are isomorphic. Hence, Thm. 1 generalizes the result of [T90b].

3 Definability implies recognizability

In all of this section *let $\mathcal{A}$ be a stably concurrent automaton.* Furthermore, let ϕ be a sentence of MSO and $L = L(\phi) \subseteq \mathrm{M}(\mathcal{A})$ a definable language. We will show that L is recognizable. To prove this, we show that $L^T := \{\gamma \in \mathrm{CS}(\mathcal{A}) \mid [\gamma] \in L\}$ is definable in a monadic second-order language in $T^\star$ where words are considered as finite linear orders. Hence, by [Bü60], L^T is recognizable in $T^\star$. This implies that L is recognizable in $\mathrm{M}(\mathcal{A})$.

Therefore, we need another monadic second-order language to describe properties of words over T. Let $\gamma = \sigma_1 \sigma_2 \ldots \sigma_n$ be a word over T (not necessarily a computation sequence). We will identify the word γ with the structure $(\mathrm{O}(\gamma), \sqsubseteq_\gamma, (T_t)_{t \in T})$ where $x \in T_t$ iff for some i, x is the i-th element in the finite linear order $(\mathrm{O}(\gamma), \sqsubseteq_\gamma)$ and $\sigma_i = t$. To describe properties of such structures we use the monadic second-order language MSO_T with atomic formulas $x \leq y$, $X(x)$ and $T_t(x)$ for $t \in T$ where x and y are first-order variables and X is a second-order variable. To become familiar with the language MSO_T consider the following sentence :

$$\mathrm{CompSeq} = \forall x \forall y \bigwedge_{t \in T} [(T_t(x) \wedge \mathrm{next}(x, y)) \to \bigvee_{\substack{t' \in T \\ \operatorname{dom} t' = \operatorname{cod} t}} T_{t'}(y))]$$

where $\mathrm{next}(x, y)$ denotes the formula $(x < y) \wedge \forall z((x \leq z \wedge z < y) \to x = z)$. Then $\gamma \in T^\star$ satisfies CompSeq (denoted by $\gamma \models \mathrm{CompSeq}$) iff $\gamma \in \mathrm{CS}(\mathcal{A})$, *i.e.* $\mathrm{CS}(\mathcal{A})$ can be defined by a sentence of the first-order fragment of MSO_T. This sentence CompSeq will be used later again.

Lemma 7. *Let $r \in Q$. Then there exists a formula Cod_r in MSO_T with a free monadic variable such that for any $\gamma \in \mathrm{CS}(\mathcal{A})$ and any $N \subseteq \mathrm{O}(\gamma)$ the following are equivalent:*

1. $(\mathrm{dom}\,\gamma).\gamma(N) = r$.
2. $\gamma \models Cod_r(N)$.

Proof The idea behind the following formula is that X_q comprises all elements $x \in N$ for which the elements of N before x change the state of $\mathcal{A}$ from $\mathrm{dom}\,\gamma$ to q; clearly, these sets X_q ($q \in Q$) are pairwise disjoint. We write $\mathrm{ev}(x) = a$ as abbreviation for $\bigvee_{t=(p,a,q)\in T} T_t(x)$ and $\mathrm{dom}(y) = q$ as abbreviation for $\bigvee_{t=(q,a,r)\in T} T_t(y)$. For $x, y \in N$ we say that y is the successor of x in N, if $x < y$ and there is no $z \in N$ with $x < z < y$; in this case, if $x \in X_p$ and $\mathrm{ev}(x) = a$, then $y \in X_{p.a}$. Now let $Q = \{q_1, q_2, \ldots, q_n\}$ and Cod_r be the following (informally described) formula:

$$
\begin{aligned}
\exists X_{q_1}, \ldots, X_{q_n}\ [\ & (\bigwedge_{i\neq j} X_{q_i} \text{ and } X_{q_j} \text{ are disjoint})\\
& \wedge \bigwedge_{q\in Q} (\text{the minimal element } y \text{ of } (O(\gamma), \sqsubseteq_\gamma) \text{ satisfies } \mathrm{dom}(y) = q\\
& \qquad \longrightarrow \text{ the minimal element } x \text{ of } N \text{ satisfies } X_q(x))\\
& \wedge \forall x \forall y (y \text{ is the successor of } x \text{ in } N\\
& \qquad \longrightarrow \bigvee_{(p,a,q)\in T} (X_p(x) \wedge \mathrm{ev}(x) = a \wedge X_q(y)))\\
& \wedge \bigvee_{(p,a,r)\in T} (\text{the maximal element } x \text{ of } N \text{ satisfies}\\
& \qquad\qquad X_p(x) \wedge \mathrm{ev}(x) = a)]
\end{aligned}
$$

This can be easily translated into the formal language MSO_T, and the result follows. □

The following lemma characterizes downward closed subsets of $\mathrm{DO}(\gamma)$. For this characterization, if $M \subseteq \mathrm{O}(\gamma)$, $x = a^i \in M$ and $y = b^j \in \mathrm{O}(\gamma) \setminus M$, let $M(x,y) := \{z \in \mathrm{O}(\gamma) \mid z \sqsubset_\gamma y\} \cup \{z \in M \mid z \sqsubset_\gamma x\}$.

Lemma 8. *Let $\gamma \in \mathrm{CS}(\mathcal{A})$ and $M \subseteq \mathrm{O}(\gamma)$. Then the following are equivalent:*

1. *M is downward closed in $\mathrm{DO}(\gamma)$.*
2. *Whenever $x = a^i \in M$, $y = b^j \in \mathrm{O}(\gamma) \setminus M$, $y \sqsubset_\gamma x$ and $r = (\mathrm{dom}\,\gamma).\gamma(M(x,y))$, then $a \parallel_r b$.*

Proof Let M be downward closed in $\mathrm{DO}(\gamma)$, $x \in M$ and $y \in \mathrm{O}(\gamma) \setminus M$ with $y \sqsubset_\gamma x$. Then x and y are incomparable in $\mathrm{DO}(\gamma)$. Hence, $M(x,y)$ is downward closed in $\mathrm{DO}(\gamma)$ with $(x{\downarrow} \cup y{\downarrow}) \setminus \{x,y\} \subseteq M(x,y) \subseteq \mathrm{DO}(\gamma) \setminus (x{\uparrow} \cup y{\uparrow})$. This implies $a \parallel_r b$ by Prop. 5.

Conversely assume the second statement. Let $x \in M$. We claim that $x{\downarrow} \subseteq M$. We may assume that $x'{\downarrow} \subseteq M$ for all $x' \in M$ with $x' \sqsubset_\gamma x$. Suppose there exists $y' \in \mathrm{O}(\gamma) \setminus M$ with $y' \sqsubset x$. Then there exists an element x' of $\mathrm{O}(\gamma)$ with $y' \sqsubseteq x' \sqsubset x$ such that there are no elements between x' and x, *i.e.* x is a direct cover of x'. If $x' \in M$ then $y' \in M$ by our assumption. Hence $Y = \{x' \in \mathrm{O}(\gamma) \setminus M \mid x \text{ is a direct cover of } x'\}$ is not empty. Let y be the maximum of Y with respect to the linear ordering $\sqsubseteq_\gamma$. We show that this leads to a contradiction.

First we show that $M(x,y)$ is downward closed in $\mathrm{DO}(\gamma)$: Suppose $z \in M(x,y)$, $z' \in \mathrm{O}(\gamma)$ and $z' \sqsubset z$. If $z \sqsubset_\gamma y$, we obtain $z' \sqsubset_\gamma y$ showing $z' \in M(x,y)$. Otherwise, $z \in M$ and $z \sqsubset_\gamma x$. By our assumption, this implies $z' \in M$ showing $z' \in M(x,y)$.

Now suppose $z \in \mathrm{O}(\gamma)$ and $z \sqsubset x$. For $z \in M$ we immediately have $z \in M(x,y)$. Otherwise there exists $x' \in Y$ with $z \sqsubseteq x' \sqsubseteq_\gamma y$ since $y = \max(Y, \sqsubseteq_\gamma)$. Hence $z \in M(x,y)$. Thus, we have $(x{\downarrow} \cup y{\downarrow}) \setminus \{x,y\} \subseteq M(x,y)$.

The inclusion $M(x,y) \subseteq \mathrm{DO}(\gamma) \setminus (x{\uparrow} \cup y{\uparrow})$ follows from the construction of $M(x,y)$ by $\sqsubseteq\ \subseteq\ \sqsubseteq_\gamma$. Choose $a,b \in E$ with $x \in E_a$, $y \in E_b$, and put $r = (\mathrm{dom}\,\gamma).\gamma(M(x,y))$. By Prop. 5 we can conclude that $a \parallel_r b$ does not hold, contradicting the assumption. □

Next we characterize the partial order relation of the dependence order $\mathrm{DO}(\gamma)$ by an MSO_T-formula for $\gamma \in \mathrm{CS}(\mathcal{A})$. In Sect. 5 we will see that this is, in general and in contrast to trace theory, not possible using first-order formulas.

Proposition 9. *There exists a formula LE (for "Less or Equal") in* MSO_T *with two free variables such that for any* $\gamma \in \mathrm{CS}(\mathcal{A})$ *and any* $x, y \in \mathrm{O}(\gamma)$ *the following are equivalent:*

1. $x \sqsubseteq y$ *in* $\mathrm{DO}(\gamma)$.
2. $\gamma \models LE(x,y)$.

Proof By Lemmas 8 and 7 we find a formula DC in MSO_T with one free monadic variable such that $M \subseteq \mathrm{O}(\gamma)$ is downward closed in $\mathrm{DO}(\gamma)$ iff $\gamma \models \mathrm{DC}(M)$. Now let $\mathrm{LE} = \forall M(M(y) \wedge \mathrm{DC}(M) \to M(x))$. Clearly, this formula meets the requirements. □

Now we are able to prove that any language definable in MSO is recognizable.

Theorem 10. *Let* ϕ *be a sentence of* MSO*. Then* $L = L(\phi)$ *is recognizable in* $\mathrm{M}(\mathcal{A})$.

Proof We may assume that $0 \notin L$. In a first step we show that $L^T = \{\gamma \in \mathrm{CS}(\mathcal{A}) \mid [\gamma] \in L\}$ is definable in MSO_T. In the sentence ϕ replace all subformulas "$x \leq y$" by "$\mathrm{LE}(x,y)$", "$E_a(x)$" by "$\bigvee_{t \in T, \mathrm{ev}(t)=a} T_t(x)$" and "$D_q$" (for

$q \in Q$) by "$\exists x(\forall y\ x \leq y \ \wedge \bigvee_{\substack{t \in T \\ \mathrm{dom}\, t = q}} T_t(x))$". Also, replace "$D_\top$" by "$\neg\exists x(x \leq x)$" and "$D_\bot$" by "$\neg$CompSeq". Denote this new formula by $\bar{\phi}$. Clearly, $\bar{\phi}$ is a sentence of MSO_T. For $\gamma \in \mathrm{CS}(\mathcal{A})$ we have $\gamma \models \bar{\phi}$ iff $\mathrm{DO}(\gamma) \models \phi$, *i.e.* iff $\gamma \in L^T$. Hence, $\phi^T = \bar{\phi} \wedge \mathrm{CompSeq}$ defines L^T in $T^\star$. Thus, L^T is recognizable in $T^\star$ by [Bü60]. By $0 \notin L$, we have $L^T = [.]^{-1}(L)$, and since the morphism $[.] : T^\star \longrightarrow \mathrm{M}(\mathcal{A})$ has either $\mathrm{M}(\mathcal{A})$ or $\mathrm{M}(\mathcal{A}) \setminus \{0\}$ as its image, L is recognizable in $\mathrm{M}(\mathcal{A})$. □

4 Recognizability implies definability

In this section, we need a monadic second order language to describe properties of words over E. This language MSO_E is defined similarly to MSO_T. A word $w = a_1 a_2 \dots a_n$ in $E^\star$ is identified with the structure $(\mathrm{O}(w), \sqsubseteq_w, (E_a)_{a \in E})$ where $x \in E_a$ iff for some i, x is the i-th element in the finite linear order $(\mathrm{O}(w), \sqsubseteq_w)$ and $a = a_i$. Therefore, the language MSO_E has the following atomic formulas: $x \leq y$, $X(x)$ and $E_a(x)$ with first-order variables x and y, second-order variable X and $a \in E$. The first-order fragment of MSO_E is defined as usual and denoted by FO_E.

Also, we will use the lexicographic normal form of a computation sequence of $\mathcal{A}$. Throughout this section, let $\preceq$ be a fixed linear order on E. Let $\gamma \in \mathrm{CS}(\mathcal{A})$. Then $\mathrm{evseq}([\gamma]) := \{\mathrm{evseq}\,\delta \mid \delta \sim \gamma\}$ contains a smallest element w with respect to the lexicographic order on $E^\star$ induced by $\preceq$. The *lexicographic normal form* of γ is defined to be the computation sequence $\delta \sim \gamma$ with $\mathrm{evseq}\,\delta = w$. Let $\mathrm{CS}_{\min}(\mathcal{A})$ comprise all computation sequences that are lexicographic normal forms.

Again, let $\gamma \in \mathrm{CS}(\mathcal{A})$ and $A = (x_1, x_2, \dots, x_n)$ be the order-preserving enumeration of $\mathrm{DO}(\gamma)$ that induces γ. Let X_i comprise all minimal elements of $\mathrm{O}(\gamma) \setminus \{x_1, x_2, \dots, x_i\}$ with respect to $\sqsubseteq$ for $i = 0, 1, \dots, n-1$. Then $x_{i+1} \in X_i$. Since X_i is an antichain in $\mathrm{DO}(\gamma)$, the elements of X_i carry mutually different actions, *i.e.* $a^j, a^k \in X_i$ imply $j = k$, as can be derived from Prop. 5. Using Thm. 6, it is easy to see that γ is a lexicographic normal form iff x_{i+1} carries the smallest action with respect to $\preceq$ in X_i for any $i = 0, 1, \dots, n-1$. This observation will be used in this section.

Lemma 11. *For any $\gamma \in \mathrm{CS}_{\min}(\mathcal{A})$ and any $a^i, b^j \in \mathrm{O}(\gamma)$, the following are equivalent:*

1. *$a^i \sqsubseteq_\gamma b^j$.*
2. *$a^i \sqsubseteq b^j$ or there exists $c^k \in \mathrm{O}(\gamma)$ with $a \prec c$ such that $a^i \sqsubseteq_\gamma c^k \sqsubseteq b^j$.*

Proof The second statement implies in particular $a^i \sqsubseteq_\gamma b^j$.

Conversely suppose $a^i \sqsubseteq_\gamma b^j$. Let $A = (x_1, x_2, \dots, x_n)$ be the order-preserving enumeration of $\mathrm{DO}(\gamma)$ that induces γ, *i.e.* $x_k \sqsubseteq_\gamma x_l \iff k \leq l$. Then there exists l with $a^i = x_l$. Let M denote the set of all minimal elements with respect to $\sqsubseteq$ of $\mathrm{O}(\gamma) \setminus \{x_1, x_2, \ldots, x_{l-1}\}$. Since A is order-preserving, a^i is an element of M. Since A induces γ which is the lexicographic normal form, a is the minimal action occurring in M. Since $a^i \sqsubseteq_\gamma b^j$ and A induces γ, the event b^j is not contained in $\{x_1, x_2, \ldots, x_{l-1}\}$. Hence there exists c^k in M with $c^k \sqsubseteq b^j$. Since M is an antichain with respect to $\sqsubseteq$ it contains events with mutually different actions. So, if $c = a$ we get $a^i = c^k \sqsubseteq b^j$. Otherwise we have $a \prec c$ since a is minimal in M with respect to $\prec$. Hence $a^i \sqsubseteq_\gamma c^k$. □

To simplify the notation, let $E = \{1, 2, \ldots, r\}$ with $\preceq$ the usual linear order. Now, we define inductively a class of first-order formulas for $s = 1, 2, \ldots, r-1$ with two free variables x and y as follows:

$$\psi_r = (x \leq y)$$

$$\psi_s = (x \leq y) \vee \exists z \left(\bigvee_{\substack{d,c \in E \\ d \prec c}} (E_d(x) \wedge E_c(z)) \wedge \neg \psi_{s+1}(z, x) \wedge z \leq y \right)$$

Note that the formula ψ_1 contains $|E| - 1$ quantifiers. Furthermore, since the existential quantifier in $\psi_{s+1}(z, x)$ occurs in the scope of the negation, the prenex normal form has an alternating sequence of existential and universal quantifiers. Hence ψ_1 is a formula in $\Sigma_{|E|-1}$.

Proposition 12. *For any $\gamma \in \mathrm{CS}_{\min}(\mathcal{A})$, any $a^i, b^j \in \mathrm{O}(\gamma)$ and any $s \leq a$ the following are equivalent:*

1. $a^i \sqsubseteq_\gamma b^j$
2. $\mathrm{DO}(\gamma) \models \psi_s(a^i, b^j)$

Proof The proof is done by induction on a.

So let $a = r$ and $s \leq r$. Then by Lemma 11 $a^i \sqsubseteq_\gamma b^j$ iff $a^i \sqsubseteq b^j$ since a is the maximal action with respect to $\preceq$. Because of this maximality, the disjunction in the scope of "$\exists z$" in the second part of ψ_s cannot hold for $x = a^i$. Thus, for $a = r$ we showed the equivalence for any $s \leq r$.

Now suppose we have $c^k \sqsubseteq_\gamma d^l \iff \mathrm{DO}(\gamma) \models \psi_s(c^k, d^l)$ for any $c^k, d^l \in \mathrm{O}(\gamma)$ with $a \prec c$ and $s \leq c$. Let $a^i, b^j \in \mathrm{O}(\gamma)$ and $s \leq a$.

By Lemma 11, $a^i \sqsubseteq_\gamma b^j$ implies $a^i \sqsubseteq b^j$ or there exists $c^k \in \mathrm{O}(\gamma)$ with $a \prec c$ such that $a^i \sqsubseteq_\gamma c^k \sqsubseteq b^j$. Thus in case a^i and b^j are comparable we have $\mathrm{DO}(\gamma) \models \psi_s(a^i, b^j)$. Now let a^i and b^j be incomparable. Then there exists $c^k \in \mathrm{O}(\gamma)$ with $a \prec c$ such that $a^i \sqsubseteq_\gamma c^k \sqsubseteq b^j$. Since $s \leq a \prec c$ we have $s+1 \leq c$ and therefore the induction hypothesis yields $\mathrm{DO}(\gamma) \models \neg\psi_{s+1}(c^k, a^i)$. Thus we get $\mathrm{DO}(\gamma) \models \psi_s(a^i, b^j)$. Conversely, suppose $\mathrm{DO}(\gamma) \models \psi_s(a^i, b^j)$. If $a^i \sqsubseteq b^j$ we have immediate $a^i \sqsubseteq_\gamma b^j$. Otherwise there exists c^k in $\mathrm{O}(\gamma)$ with $a \prec c$, $\mathrm{DO}(\gamma) \models \neg\psi_{s+1}(c^k, a^i)$ and $c^k \leq b^j$. By the induction hypothesis, we obtain $a^i \sqsubseteq_\gamma c^k$ and therefore $a^i \sqsubseteq_\gamma b^j$. □

Now we can prove that any recognizable language in $\mathrm{M}(\mathcal{A})$ is definable by a sentence of MSO.

Theorem 13. *Let L be a recognizable language in* $\mathrm{M}(\mathcal{A})$*. Then there exists a sentence ϕ of* MSO *such that $L = L(\phi)$.*

Proof Since $\{0\}$ and $\{1\}$ are definable, we may assume that $0, 1 \notin L$. Let $q \in Q$, $L_q^E = \{\mathrm{evseq}\,\gamma \mid [\gamma] \in L \text{ and } \mathrm{dom}\,\gamma = q\}$ and $x \in E^\star$. If $x^{-1}L_q^E \neq \emptyset$ then there exists a uniquely determined computation sequence $\gamma \in \mathrm{CS}(\mathcal{A})$ with $\mathrm{dom}\,\gamma = q$ and $\mathrm{evseq}\,\gamma = x$. Furthermore, $z \in x^{-1}L_q^E$ iff there exists $\delta \in \mathrm{CS}(\mathcal{A})$ with $\mathrm{evseq}\,\delta = z$ and $[\gamma\delta] \in L$. Hence, $x^{-1}L_q^E = \{\mathrm{evseq}\,\delta \mid [\delta] \in [\gamma]^{-1}L\}$. Since L is recognizable, there are only finitely many sets $[\gamma]^{-1}L$. Hence, $\{x^{-1}L_q^E \mid x \in E^\star\}$ is finite, *i.e.* L_q^E is recognizable in the free monoid $E^\star$.

By [Bü60] there exists a sentence ϕ_q^E of MSO_E such that $L_q^E = \{w \in E^\star \mid w \models \phi_q^E\}$. We construct a sentence ϕ_q^1 of MSO from ϕ_q^E by replacing all subformulas of the form "$x \leq y$" by "$\psi_1(x, y)$". Then put $\phi_q = \phi_q^1 \wedge D_q$.

We show that ϕ_q defines $L_q = \{[\gamma] \in L \mid \mathrm{dom}\,\gamma = q\}$: Let $\gamma' \in \mathrm{CS}(\mathcal{A})$ and γ be the lexicographic normal form of γ', *i.e.* $\gamma \sim \gamma'$ and $\gamma \in \mathrm{CS}_{\min}(\mathcal{A})$. Then, using Prop. 12, we have the following equivalences:

$$\begin{aligned}
\mathrm{DO}(\gamma') \models \phi_q &\iff \mathrm{DO}(\gamma) \models \phi_q \\
&\iff \mathrm{evseq}\,\gamma \models \phi_q^E \text{ and } \mathrm{dom}\,\gamma = q \\
&\iff \mathrm{evseq}\,\gamma \in L_q^E \text{ and } \mathrm{dom}\,\gamma = q \\
&\iff [\gamma] \in L_q \iff [\gamma'] \in L_q.
\end{aligned}$$

Now clearly the sentence $\bigvee_{q \in Q} \phi_q$ of MSO defines $L = \bigcup_{q \in Q} L_q$. □

Now Thm. 1 is immediate by Thm. 10 and Thm. 13.

5 First-order definable and aperiodic languages

In [T90b, EM93] it has been shown that a language in a trace monoid is definable by a first-order formula iff it is aperiodic. By [GRS92], it is aperiodic iff it is starfree. As shown in [Dr94b], any aperiodic language in a concurrency monoid is starfree, but not necessarily conversely. Here we give examples of concurrency monoids that contain aperiodic languages which are not first-order definable, and vice versa. Hence, the classes of aperiodic, starfree and first-order definable languages are in general mutually different in a concurrency monoid. As a positive result, we formulate a sufficient condition on $\mathcal{A}$ such that any aperiodic language in $\mathrm{M}(\mathcal{A})$ is first-order definable. Also, we describe two classes of automata where any first-order definable language is aperiodic. For one of these classes, the aperiodic and the starfree languages coincide ([Dr94b]). Hence we can describe a class of automata where aperiodic, starfree and first-order definable languages coincide. This class contains, besides others, automata with only one state. Since these are precisely the automata induced by a trace alphabet, our result generalizes the result of [T90b] and that of [EM93] on finite traces.

We start with a simple example of a language that is aperiodic but not first-order definable.

Example 1. Consider the stably concurrent automaton $\mathcal{A}$ with $Q = \{p, q\}$, $E = \{a\}$ and transitions $s = (p, a, q)$ and $t = (q, a, p)$. The reader may check that the language $L = (s \cdot \mathrm{M}(\mathcal{A}) \cdot t \cup t \cdot \mathrm{M}(\mathcal{A}) \cdot s) \setminus \{0\} = \{[\gamma] \in \mathrm{M}(\mathcal{A}) \mid \mathrm{dom}\,\gamma = \mathrm{cod}\,\gamma\}$ is aperiodic with index 2. Clearly, for any $\gamma \in \mathrm{CS}(\mathcal{A})$, $(\mathrm{O}(\gamma), \leq)$ is a linear order with $|\gamma|$ elements. Hence, $[\gamma] \in L$ iff $\mathrm{O}(\gamma)$ has an even number of elements. But a first-order formula cannot distinguish between linear orders of even and of odd length. Hence, L is not definable in FO.

The automaton of this example is not counter free as defined below.

Definition 14. An automaton with concurrency relations $\mathcal{A}$ *is counter free* if $q.w^n = q$ implies $q.w = q$ for any $q \in Q$, $w \in E^\star$ and any natural number $n > 0$.

Obviously, any automaton with precisely one state is counter free. Hence these automata generalize trace alphabets. Now, let $\mathcal{A}$ be a counter free automaton, $w \in E^\star$ and $q \in Q$. Since Q is finite, there exist natural numbers m and $n > 0$ with $q.w^m = q.w^{m+n}$. Suppose m is minimal with this property. Thus, the elements of $\{q.w^k \mid k \leq m\}$ are mutually different. Hence, $m \leq |Q|$. Because of $(q.w^m).w^n = q.w^m$ we obtain by the assumption on $\mathcal{A}$ $q.w^{m+1} = q.w^m$. Thus, we have $q.w^{|Q|} = q.w^{|Q|+1}$ for any $q \in Q$ and $w \in E^\star$. Suppose conversely that this holds in a stably concurrent automaton $\mathcal{A}$. Let $q \in Q$, $w \in E^\star$ and $n > 0$ with $q.w^n = q$. Then we have $q = q.(w^n)^{|Q|} = q.w^{n|Q|}$. This equals $q.w^{n|Q|+1}$ since $n|Q| \geq |Q|$. Hence, $q = (q.w^{n|Q|}).w = q.w$, *i.e.* $\mathcal{A}$ is counter free, too.

Thus, an automaton with concurrency relations is counter free iff $q.w^{|Q|} = q.w^{|Q|+1}$ for any $q \in Q$ and any $w \in E^\star$.

Proposition 15. *Let $\mathcal{A}$ be a counter free stably concurrent automaton. Let $L \subseteq \mathrm{M}(\mathcal{A})$ be an aperiodic language with index k. Then, for any state $q \in Q$, $L_q^E = \{\mathrm{evseq}\,\delta \mid [\delta] \in L, \mathrm{dom}\,\delta = q\}$ is aperiodic in $E^\star$ with index at most $2 \cdot \max(k, |Q|)$.*

Proof Let $n = \max(k, |Q|)$. Suppose $uv^{(2n)}w \in L_q^E$. Then there exist $\gamma, \delta, \eta \in \mathrm{CS}(\mathcal{A})$ with $\mathrm{evseq}\,\gamma = u$, $\mathrm{evseq}\,\delta = v^{2n}$, $\mathrm{evseq}\,\eta = w$ and $[\gamma\delta\eta] \in L$. Also, δ can be written as $\delta_1\delta_2\delta_3$ with $\mathrm{evseq}\,\delta_1 = v^n$, $\mathrm{evseq}\,\delta_2 = v$ and $\mathrm{evseq}\,\delta_3 = v^{n-1}$. Let p denote the codomain of γ. Then we have $\mathrm{cod}\,\delta_1 = p.v^n = p.v^{n+1} = \mathrm{cod}\,\delta_2$ since $\mathcal{A}$ is counter free and $n \geq |Q|$. Hence $\mathrm{dom}\,\delta_2 = \mathrm{dom}\,\delta_3$ and therefore $\delta_2\delta_3 = \delta_2^n$. This implies $[\gamma\delta_1\delta_2^{n+1}\eta] \in L$ since $n \geq k$, the index of L. Now we have $uv^{2n+1}w = uv^nv^{n+1}w \in L_q^E$.

Conversely suppose $uv^{2n+1}w \in L_q^E$. We find $\gamma, \delta_1, \delta_2, \eta \in \mathrm{CS}(\mathcal{A})$ with $\mathrm{evseq}\,\gamma = u$, $\mathrm{dom}\,\gamma = q$, $\mathrm{evseq}\,\delta_1 = v^n$, $\mathrm{evseq}\,\delta_2 = v^{n+1}$, $\mathrm{evseq}\,\eta = w$ and $[\gamma\delta_1\delta_2\eta] \in L$. Again, let p denote the codomain of γ. Since $\mathcal{A}$ is counter free, we have $\mathrm{cod}\,\delta_1 = p.v^n = p.v^{n+1} = p.v^{n+2} = \ldots$ Therefore, we find $\delta \in \mathrm{CS}(\mathcal{A})$ with $\mathrm{dom}\,\delta = \mathrm{cod}\,\delta = p.v^n$ and $\mathrm{evseq}\,\delta = v$ such that $\delta_2 = \delta^{n+1}$. Because of

$[\gamma\delta_1\delta^{n+1}\eta] \in L$ and $n \geq k$, we obtain $[\gamma\delta_1\delta^n\eta] \in L$, *i.e.* $uv^{2n}w = uv^n v^n w \in L_q^E$. □

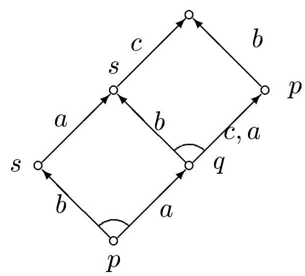
Now we can show that in this case all aperiodic languages are definable by FO.

Theorem 16. *Let $\mathcal{A}$ be a counter free stably concurrent automaton. Let $L \subseteq$ $\mathrm{M}(\mathcal{A})$ be an aperiodic language. Then there exists a sentence ϕ of* FO *such that $L = L(\phi)$.*

Proof Let $q \in Q$. By Prop. 15, L_q^E is aperiodic. By [MP71], there exists a sentence ϕ_q^E of FO_E with $L_q^E = L(\phi_q^E)$. Now, the proof proceeds similarly to the proof of Thm. 13. □

Now, we give an example of a language that can be defined by FO but is not aperiodic.

Example 2. Let $\mathcal{A}$ be the following stably concurrent automaton with $a \parallel_p b$ and $b \parallel_q c$.



Note that $s.a = s$, $p.a^{2n} = p$, $p.ac = p$ and that b is concurrent with both a and c in state q. We consider the following language

$$L = \{[\gamma] \mid \operatorname{dom}\gamma = p \text{ and } \exists n \in \mathbb{N} : \operatorname{evseq}\gamma = ba^{2n}c\}$$

that is not aperiodic. For $w \in E^\star$ let $w@p$ denote the computation sequence δ with $\operatorname{dom}\delta = p$ and $\operatorname{evseq}\delta = w$ (if it exists). Let $\gamma = bac@p$. Then $\gamma \sim abc@p \sim acb@p$. Hence, b^1 and c^1 are incomparable in $\mathrm{DO}(\gamma)$. Now consider $\gamma = baac@p$. Then we have $[\gamma] = \{baac@p, abac@p, aabc@p\}$. Hence $b^1 \sqsubseteq c^1$ in $\mathrm{DO}(\gamma)$. By induction we can show that a natural number k is even iff $b^1 \sqsubseteq c^1$ in $\mathrm{DO}(ba^k c@p)$. Hence, L can be defined by FO.

Additionally, the example shows that the dependence order $\mathrm{DO}(\gamma)$ cannot be defined by a first-order formula from $\gamma \in \mathrm{CS}(\mathcal{A})$: Suppose there exists a formula LE in FO_T that satisfies Prop. 9. Then, in the example above, $ba^k c@p$ satisfies $LE(b^1, c^1)$ iff k is even. Hence, by [MP71], the language $\{ba^{2n}c@p \mid n \in \mathbb{N}\} \subseteq T^\star$ is aperiodic. But this is not the case.

Lemma 17. *Let $\mathcal{A}$ be a stably concurrent automaton such that a formula LE in FO_T exists with the properties of Prop. 9. Then any first-order definable language in $\mathrm{M}(\mathcal{A})$ is aperiodic.*

Proof Let ϕ be a sentence of FO and $L = L(\phi)$. Following the proof of Thm. 10 we find that L^T can be defined by a sentence of FO_T. By [MP71], L^T is aperiodic in $T^\star$. Hence, L is aperiodic with the same index. $\square$

Thus to show that any first-order definable language is aperiodic, it suffices to show that there exists a formula LE of FO_T that expresses the dependence order. Therefore, we determine two classes of automata that meet this requirement (Def. 18 and 20).

Definition 18. An automaton with concurrency relations $\mathcal{A}$ is an *automaton with global independence* if whenever $a \parallel_p b$ and $q.ab$ is defined then $a \parallel_q b$ for any $a, b \in E$ and $p, q \in Q$.

Note that any automaton induced by a trace alphabet has global independence. It is possible to check that any automaton with global independence is stably concurrent.

Together with Lemma 17, the following proposition implies that for an automaton $\mathcal{A}$ with global independence any first-order definable language in $\mathrm{M}(\mathcal{A})$ is aperiodic.

Proposition 19. *Let $\mathcal{A}$ be an automaton with global independence. Then there exists a formula LE in FO_T with two free variables such that for any $\gamma \in \mathrm{CS}(\mathcal{A})$ and any $x, y \in \mathrm{O}(\gamma)$ the following are equivalent:*

1. $x \sqsubseteq y$ *in* $\mathrm{DO}(\gamma)$.
2. $\gamma \models LE(x, y)$.

Proof Let $I := \bigcup\{\parallel_q \mid q \in Q\}$. Then $(E, E^2 \setminus I)$ is a trace alphabet. Let $\gamma \in \mathrm{CS}(\mathcal{A})$. One can show that, since $\mathcal{A}$ is an automaton with global independence, the dependence graph of evseq γ with respect to $(E, E^2 \setminus I)$ and the partially ordered set $(\mathrm{O}(\gamma), \sqsubseteq)$ coincide. But it is well known that the dependence graph can be defined by the following first-order formula LE:

$$\bigvee_{\substack{\{a_1,\ldots,a_n\}\subseteq E\\ (a_i,a_{i+1})\notin I}} \exists x_1, \ldots, x_n \left[\bigwedge_{i \le n} E_{a_i}(x_i) \wedge \bigwedge_{i \le n-1} x_i < x_{i+1} \wedge (x_1 = x) \wedge (x_n = y) \right].$$

$\square$

[Dr94b, Example 3.1] shows that there exists an automaton with global independence $\mathcal{A}$ and a starfree language in $\mathrm{M}(\mathcal{A})$ that is not aperiodic. In this automaton, it is possible to define a language similar to that of Example 1 that is aperiodic but not first-order definable. Now we define a class of automata where a language is aperiodic iff it is starfree ([Dr94b]) and show that furthermore any first-order definable language is aperiodic.

Let $\mathcal{A}$ be a stably concurrent automaton, $w = a_1 a_2 \ldots a_n \in E^\star$, $q \in Q$ and $a \in E$. We say *a and w commute in q* (denoted by $a \parallel_q w$) if $q.w$ is defined and $a \parallel_{q.a_1 a_2 \ldots a_i} a_{i+1}$ for $i = 0, 1, \ldots, n-1$.

Definition 20. A stably concurrent automaton $\mathcal{A}$ *has no commuting loops* if $a \parallel_q w$ implies $q.w \neq q$ for any $q \in Q$, $w \in E^\star$ and $a \in E$.

Suppose $a \in E$, $w \in E^\star$, $q \in Q$ such that $a \parallel_q w$ and $q.w = q$. Then, there exists a prefix uv of w of length at most $|Q|$ such that $q.uv = q.u$. Let $p := q.u$. Now $a \parallel_p v$ is immediate. Hence, to check whether a stably concurrent automaton has a commuting loop it suffices to consider words w of length at most $|Q|$. Note that if $\mathcal{A}$ is the automaton induced by a trace alphabet (E, D), thus has only one state, and has no commuting loops, then $D = E \times E$. Thus the class of automata without commuting loops forms a model for concurrent systems complementary to trace alphabets.

To show that for a stably concurrent automaton without commuting loops there exists a first-order formula describing the dependence order we need the following lemma which describes when two elements of the dependence order $\mathrm{DO}(\gamma)$ are incomparable. It is valid for any stably concurrent automaton.

Lemma 21. *Let $\mathcal{A}$ be a stably concurrent automaton, $\sigma_x, \sigma_y \in T$, $\gamma_i \in \mathrm{CS}(\mathcal{A})$ for $i = 1, 2, 3$, $\gamma = \gamma_1\sigma_x\gamma_2\sigma_y\gamma_3 \in \mathrm{CS}(\mathcal{A})$, $\mathrm{ev}\,\sigma_x = a$, $\mathrm{ev}\,\sigma_y = b$, $|\gamma_1|_a = i - 1$ and $|\gamma_1\sigma_x\gamma_2|_b = j - 1$. Then $x = a^i$ and $y = b^j$ are incomparable in $\mathrm{DO}(\gamma)$ iff there exist computation sequences δ_1 and δ_2 with $\gamma_2 \sim \delta_1\delta_2$, $a \parallel_{\mathrm{cod}\,\gamma_1} (\mathrm{evseq}\,\delta_1)b$ and $b \parallel_{\mathrm{dom}\,\delta_2} \mathrm{evseq}\,\delta_2$.*

Proof Suppose that x and y are incomparable. Let $w_i = \mathrm{evseq}\,\gamma_i$ for $i = 1, 2, 3$. There exists an order-preserving enumeration

$$A = (x_1, x_2, \ldots, x_k, x, y_1, y_2, \ldots, y_l, y, z_1, z_2, \ldots, z_m)$$

of $\mathrm{DO}(\gamma)$ that induces γ. Let $(y'_1, y'_2, \ldots, y'_n)$ denote the subsequence of $(y_1, y_2, \ldots, y_l)$ comprising all elements y_i with $y_i \leq y$ and let $(y'_{n+1}, \ldots, y'_l)$ denote the remaining subsequence of $(y_1, y_2, \ldots, y_l)$. For $i \leq n$ we have $x \parallel y'_i$, hence $y'_i \notin E_a$. Similarly, for $i > n$, we have $y \parallel y'_i$, implying $y'_i \notin E_b$. Hence

$$B = (x_1, x_2, \ldots, x_k, x, y'_1, y'_2, \ldots, y'_l, y, z_1, z_2, \ldots, z_m),$$
$$C = (x_1, x_2, \ldots, x_k, x, y'_1, y'_2, \ldots, y'_n, y, y'_{n+1}, \ldots, y'_l, z_1, z_2, \ldots, z_m) \text{ and}$$
$$D = (x_1, x_2, \ldots, x_k, y'_1, y'_2, \ldots, y'_n, y, x, y'_{n+1}, \ldots, y'_l, z_1, z_2, \ldots, z_m)$$

are order-preserving enumerations of $\mathrm{DO}(\gamma)$. Let u_1 denote the sequence of actions of $(y'_1, y'_2, \ldots, y'_n)$ and u_2 that of $(y'_{n+1}, \ldots, y'_l)$. Then B induces a computation sequence with event sequence $w_1au_1u_2bw_3$. Hence, there exist computation sequences δ_1 and δ_2 with $\mathrm{evseq}\,\delta_i = u_i$ and $\gamma_2 \sim \delta_1\delta_2$. The enumeration C induces a computation sequence with event sequence $w_1au_1bu_2w_3$. Since $y'_i \notin E_b$ for $i > n$, the word u_2 does not contain any b. Therefore, $b \parallel_p u_2$ with $p = (\mathrm{dom}\,\gamma).w_1au_1 = \mathrm{dom}\,\delta_2$. Finally, D induces a computation sequence with event sequence $w_1u_1bau_2w_3$. Since $y'_i \notin E_a$ for $i \leq n$, and since $a \neq b$, the word u_1b does not contain any a. This implies $a \parallel_q u_1b$ with $q = (\mathrm{dom}\,\gamma).w_1 = \mathrm{cod}\,\gamma_1$.

Suppose, conversely, there exist δ_1 and δ_2 with the properties described above. Then $\gamma \sim \gamma_1\delta'_1\sigma'_y\sigma'_x\delta'_2\gamma_3 =: \delta$ with $\mathrm{evseq}\,\delta_i = \mathrm{evseq}\,\delta'_i$ $(i = 1, 2)$, $\mathrm{ev}\,\sigma_x = \mathrm{ev}\,\sigma'_x$

and $\mathrm{ev}\,\sigma_y = \mathrm{ev}\,\sigma'_y$. By $a \parallel_{\mathrm{cod}\,\gamma_1} (\mathrm{evseq}\,\delta_1)b$, we obtain $|\delta'_1\sigma'_y|_a = |\delta_1|_a + |\sigma_y|_a = 0$. Similarly, $b \parallel_{\mathrm{dom}\,\delta_2} \mathrm{evseq}\,\delta_2$ implies $|\delta'_2|_b = |\delta_2|_b = 0$. Hence $|\gamma_1\delta'_1\sigma'_y|_a = i-1$ and $|\gamma_1\delta'_1|_b = |\gamma_1|_b + |\delta_1|_b = |\gamma_1\sigma_x\delta_1\delta_2|_b = j-1$. Hence $y \sqsubseteq_\delta x$ which implies that x and y are incomparable in $\mathrm{DO}(\gamma)$. □

Now, the following proposition implies that for stably concurrent automata without commuting loops any first-order definable language in $\mathrm{M}(\mathcal{A})$ is aperiodic.

Proposition 22. *Let $\mathcal{A}$ be a stably concurrent automaton without commuting loops. Then there exists a formula LE in FO_T with two free variables such that for any $\gamma \in \mathrm{CS}(\mathcal{A})$ and any $x, y \in \mathrm{O}(\gamma)$ the following are equivalent:*

1. $x \sqsubseteq y$ *in* $\mathrm{DO}(\gamma)$.
2. $\gamma \models LE(x, y)$.

Proof Let $x = a^i$ and $y = b^j$. Clearly, $x \sqsubseteq_\gamma y$ is a necessary condition for $x \sqsubseteq y$. Therefore, suppose $x \sqsubseteq_\gamma y$. Then there exist $\gamma_1, \gamma_2, \gamma_3 \in \mathrm{CS}(\mathcal{A})$ and $\sigma_x, \sigma_y \in T$ satisfying the assumptions of Lemma 21. Hence, x and y are incomparable in $\mathrm{DO}(\gamma)$ iff there exist $\delta_1, \delta_2 \in \mathrm{CS}(\mathcal{A})$ with the properties described in Lemma 21. Since $\mathcal{A}$ has no commuting loops, this implies $|\delta_1| < |Q| - 1$ and $|\delta_2| < |Q|$, hence $|\gamma_2| < 2|Q| - 1$. Since $\mathcal{A}$ is finite, there are only finitely many computation sequences of length less than $2|Q| - 1$. Hence it is possible to express by a first-order formula LE that $x \sqsubseteq y$ in $\mathrm{DO}(\gamma)$ holds. □

Summarizing Prop. 19 and 22 and Lemma 17, we obtain:

Theorem 23. *Let $\mathcal{A}$ be either a stably concurrent automaton without commuting loops, or an automaton with global independence. Then each first-order definable language in $\mathrm{M}(\mathcal{A})$ is aperiodic.*

Now Thm. 2 given in the introduction is immediate by Thms. 16 and 23.

Acknowledgment

The authors want to thank V. Diekert whose idea for lexicographic normal forms of traces improved Prop. 12 remarkably. Originally, the authors had a formula with $|E|^2$ quantifiers. He suggested the one given here with only $|E|-1$ quantifiers (*cf.* [DM96]).

References

[AR88] Aalbersberg, I.J. and G. Rozenberg: *Theory of traces.* Theor. Comp. Science 60 (1988), 1-82.

[Be87] Bednarczyk, M.: *Categories of asynchronous systems.* Ph.D. thesis, University of Sussex, 1987.

[Be78] Berry, G.: *Stable models of typed λ-calculi.* In: 5th ICALP, Lect. Notes in Comp. Science vol. 62, Springer, 72-89, 1978.

[BCS93] BOLDI, P., F. CARDONE and N. SABADINI: *Concurrent automata, prime event structures and universal domains*, In: *Semantics of Programming Languages and Model Theory* (M. Droste, Y. Gurevich, eds.), Gordon and Breach Science Publ., OPA Amsterdam, 1993, pp. 89-108.

[BR94] BÖRGER, E. and D. ROSENZWEIG: *Occam: Specification and compiler correctness. Part I: Simple mathematical interpreters*. To appear.

[BD93] BRACHO, F. and M. DROSTE: *From domains to automata with concurrency.* In: 20th ICALP, Lecture Notes in Computer Science vol. 700, Springer, 1993, pp. 669-681.

[BD94] BRACHO, F. and M. DROSTE: *Labelled domains and automata with concurrency*. Theor. Comp. Science 135 (1994), 289-318.

[BDK95] BRACHO, F., M. DROSTE and D. KUSKE: *Dependence orders for computations of concurrent automata*. In: STACS'95, Lecture Notes in Computer Science vol. 900, Springer 1995, 467-678.

[BDK96] BRACHO, F., M. DROSTE and D. KUSKE: *Representation of computations in concurrent automata by dependence orders*. Theor. Comp. Science, 1996, to appear.

[Bü60] BÜCHI, J.R.: *Weak second-order arithmetic and finite automata*. Z. Math. Logik Grundlagen Math. 6 (1960), 66-92.

[Cu93] CURIEN, P.L.: *Categorical Combinators, Sequential Algorithms and Functional Programming*. Progress in Theoretical Computer Science, Birkhäuser, Boston, 1993.

[Di90] DIEKERT, V.: *Combinatorics on Traces*. Lecture Notes in Computer Science vol. 454, Springer, 1990.

[DM96] V. DIEKERT and Y. MÉTIVIER: *Partial commutation and traces*. In: G. Rozenberg and A. Salomaa: Handbook on Formal Languages vol. 3, Springer, to appear, 1996.

[DR95] DIEKERT, V. and G. ROZENBERG: *The Book of Traces*. World Scientific Publ. Co., 1995.

[Dr90] DROSTE, M.: *Concurrency, automata and domains*. In: *17th ICALP*, Lecture Notes in Computer Science vol. 443, Springer, 1990, pp. 195-208.

[Dr92] DROSTE, M.: *Concurrent automata and domains*. Intern. J. of Found. of Comp. Science 3 (1992), 389-418.

[Dr94a] DROSTE, M.: *A Kleene theorem for recognizable languages over concurrency monoids*. In: 21st ICALP, Lecture Notes in Computer Science vol. 820, Springer 1994, pp. 388-399.
Full version: *Recognizable languages in concurrency monoids*, Theor. Comp. Science 150 (1996), pp. 77-109.

[Dr94b] DROSTE, M.: *Aperiodic languages over concurrency monoids*. Information and Computation, 1996, to appear.

[DK96] DROSTE, M. and D. KUSKE: *Recognizable and logically definable languages of infinite computations in concurrency monoids*. Technical report, Inst. für Algebra, TU Dresden, 1995, submitted.

[EM93] EBINGER, W. and A. MUSCHOLL: *Logical definability on infinite traces*. In: 20th ICALP, Lecture Notes in Computer Science vol. 700, Springer, 1993, pp. 335-346.

[GRS92] GUAIANA, G., A. RESTIVO and S. SALEMI: *Star-free trace languages*. Theor. Comp. Science 97 (1992), 301-311.

[Ku94a] KUSKE, D.: *Modelle nebenläufiger Prozesse – Monoide, Residuensysteme und Automaten*. Dissertation, Universität GH Essen, 1994.

[Ku94b] Kuske, D.: *Nondeterministic automata with concurrency relations.* Colloquium on Trees in Algebra and Programming, Lecture Notes in Computer Science vol. 787, Springer, 1994, 202-217.

[Ma77] Mazurkiewicz, A.: *Concurrent program schemes and their interpretation.* DAIMI Report PB-78, Aarhus University, Aarhus, 1977.

[Ma86] Mazurkiewicz, A.: *Trace theory.* In: *Advanced Course on Petri Nets*, Lecture Notes in Computer Science, Springer, 1986, pp. 279-324.

[Ma88] Mazurkiewicz, A.: *Basic notions of trace theory.* In: *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency* (J.W. de Bakker, W.-P. de Roever, G. Rozenberg), Lecture Notes in Computer Science vol. 354, Springer, 1988, pp. 285-363.

[MP71] McNaughton, R. and S. Papert: *Counter-free Automata.* MIT-Press, Cambridge, USA, 1971.

[Oc85] Ochmanski, E.: *Regular behaviour of concurrent systems*, Bull. Europ. Assoc. for Theoret. Comp. Science 27 (1985), 56-67.

[PSS90] Panangaden, P., V. Shanbhogue and E.W. Stark: *Stability and sequentiability in dataflow networks.* In: *17th ICALP*, Lecture Notes in Computer Science vol. 443, Springer, 1990, pp. 308-321.

[Sh85] Shields, M.W.: *Concurrent machines.* Computer Journal 28 (1985), 449-465.

[S89a] Stark, E.W.: *Connections between a concrete and an abstract model of concurrent systems.* In: *Proceedings of the 5th Conf. on the Mathematical Foundations of Programming Semantics*, Lecture Notes in Computer Science vol. 389, Springer, 1989, pp. 52-74.

[S89b] Stark, E.W.: *Compositional relational semantics for indeterminate dataflow networks.* In: Proc. Category Theory and Computer Science, Lecture Notes in Computer Science vol. 389, Springer, 1989, pp. 52-74.

[T90a] Thomas, W.: *Automata on Infinite Objects.* In: *Handbook of Theoretical Computer Science* (J. van Leeuwen, ed.), Elsevier Science Publ. B.V., 1990, pp. 133-191.

[T90b] Thomas, W.: *On logical definability of trace languages.* In: Proc. of the workshop Algebraic Methods in Computer Science, Kochel am See, FRG (1989) (V. Diekert, ed.). Report TUM-I9002, TU Munich, 1990, pp. 172-182.

[WN94] Winskel, G. and M. Nielsen: *Models for concurrency.* In: *Handbook of Logic in Computer Science* (S. Abramsky, D.M. Gabbay, T.S.E. Maibaum, eds.), to appear.