On the complementation of Büchi asynchronous cellular automata *

Anca Muscholl

Universität Stuttgart
Institut für Informatik
Breitwiesenstr. 20-22
D-70565 Stuttgart

Abstract. We present a subset automaton construction for asynchronous cellular automata. This provides an answer to the problem of direct determinization for asynchronous cellular automata for finite traces, which has been open for several years. We use the subset automaton construction in order to extend Klarlund's progress measure technique of complementation to non-deterministic asynchronous cellular Büchi automata for infinite traces. Our automaton for the complement (as well as the subset automaton) has $2^{O(N^{|\Sigma|})}$ global states.

1 Introduction

The closure of the class of recognizable languages of infinite traces under complementation is easily derived from the definition by recognizing morphisms (or the recognition by deterministic Muller automata). We present in this paper an automata-theoretic proof by exhibiting a complementation procedure for non-deterministic asynchronous cellular Büchi automata. We use the efficient progress-measure technique introduced by N. Klarlund [Kla91] for complementing Büchi (and Streett) ω-word automata, by extending it to asynchronous cellular automata.

Klarlund's complementation construction, as well as other related constructions (e.g. Safra's determinization construction [Saf88]), requires a construction for subset automata. In the case of asynchronous automata, no determinization construction has been so far available, in spite of several attempts [Pig93a, DE92][2]. We present in Sect. 3 a determinization construction for asynchronous cellular automata based on Cori/Métivier's notion of asynchronous mapping [CM88], which leads directly to deterministic asynchronous cellular automata. Using Zielonka's bounded time-stamping we exhibit a natural asynchronous mapping which is directly translated into the subset automaton. However, several other new ideas are necessary in order to obtain our result.

* Research supported by the ESPRIT Basic Research Action No. 6317 (ASMICS 2).

[2] Klarlund et. al. have announced another subset construction for asynchronous automata. See [KMS94] in this volume.

We work on infinite traces only in Sect. 4, where we use the constructed subset automaton and extend Klarlund's complementation construction to asynchronous cellular automata. We obtain a blow-up of global states of $2^{O(N^{|\Sigma|})}$, where Σ denotes the alphabet. We believe that this is essentially optimal. It is worth noting that we can obtain a deterministic automaton for the complementation by means of Muller automata, applying the algebraic construction given in [DM93]. This yields however a blow-up of global states at least double exponential.

2 Preliminaries

Throughout this paper we denote by (Σ, D) a finite dependence alphabet, where Σ is a finite alphabet and $D \subseteq \Sigma \times \Sigma$ a reflexive, symmetric dependence relation. The complementary relation $I = (\Sigma \times \Sigma) \setminus D$, called *independence relation*, induces an equivalence relation on Σ^*, generated by pairs $(uabv, ubav)$, with $u, v \in \Sigma^*$, $(a,b) \in I$. The above equivalence relation is in fact a congruence, hence it generates the quotient monoid (free partially commutative monoid) $\mathbb{M}(\Sigma, D) = \Sigma^*/\{ab = ba \mid (a,b) \in I\}$ called *trace monoid* by Mazurkiewicz [Maz77] and first used in combinatorics [CF69]. The associated canonical surjective homomorphism will be denoted by $\varphi : \Sigma^* \to \mathbb{M}(\Sigma, D)$. For $a \in \Sigma$, $A \subseteq \Sigma$ let $D(a) = \{b \in \Sigma \mid (a,b) \in D\}$ and $D(A) = \bigcup_{a \in A} D(a)$.

So far, traces are equivalence classes of words and they may be represented by words. A natural and more useful representation is the graphical representation of the labelled partial order given by a trace. We identify traces with dependence graphs, i.e. with (isomorphism classes of) labelled directed, acyclic graphs $G = [V, E, \lambda]$, where $\lambda : V \to \Sigma$ is the labelling of the vertex set and edges exist between (different) vertices with dependent labellings, i.e. for every $u, v \in V$ we have $(\lambda(u), \lambda(v)) \in D$ if and only if $u = v$ or $(u,v) \in E \cup E^{-1}$. Given a trace $t = [a_1 \ldots a_n]$, we define the associated dependence graph by taking n vertices $V = \{1, \ldots, n\}$ labelled as $\lambda(i) = a_i$ and edges (i,j) for $1 \le i < j \le n$ whenever $(a_i, a_j) \in D$.

In this paper we consider (in)finite dependence graphs with countable vertex sets, such that $\lambda^{-1}(a)$ is well-ordered for every $a \in \Sigma$. The set of dependence graphs satisfying these properties is denoted by $\mathbb{G}(\Sigma, D)$. It forms a monoid with the multiplication $[V_1, E_1, \lambda_1][V_2, E_2, \lambda_2] = [V_1 \dot{\bigcup} V_2, E, \lambda_1 \dot{\bigcup} \lambda_2]$, where $E = E_1 \dot{\bigcup} E_2 \dot{\bigcup} \{(v_1, v_2) \in V_1 \times V_2 \mid (\lambda_1(v_1), \lambda_2(v_2)) \in D\}$. The identity is the empty graph, denoted by 1. The requirement that any subset of vertices with the same label should be well-ordered allows us to represent vertices as pairs (a, i), with $a \in \Sigma$ and i a countable ordinal, where (a, i) represents the $(i+1)$-th node labelled by the letter a. This is called the *standard representation*.

We are concerned here with a subset of $\mathbb{G}(\Sigma, D)$ consisting of dependence graphs where every vertex has a finite past, i.e. with *real traces*. The set of real traces is denoted by $\mathbb{R}(\Sigma, D)$. Equivalently, real traces correspond to (in)finite dependence graphs having a representation by (in)finite words; i.e. with $\varphi : \Sigma^\infty \to \mathbb{G}(\Sigma, D)$ being the extension of the canonical mapping on the set of finite and infinite words $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$, we have $\mathbb{R}(\Sigma, D) = \varphi(\Sigma^\infty)$. Note that $\mathbb{R}(\Sigma, D)$

is not a submonoid of $\mathbb{G}(\Sigma, D)$ (e.g. for $(a,b) \in D$ we have $a^\omega, b \in \mathbb{R}(\Sigma, D)$, but $a^\omega b \notin \mathbb{R}(\Sigma, D)$).

Recognizable languages form a well-studied language class in various contexts of (in)finite objects: words, traces, trees. A real trace language $L \subseteq \mathbb{R}(\Sigma, D)$ is *recognizable* if $\varphi^{-1}(L) \subseteq \Sigma^\infty$ is recognizable in the classical sense of ω-word languages. The class of recognizable real trace languages is denoted by $\mathrm{Rec}(\mathbb{R}(\Sigma, D))$.

Automata-theoretic characterizations of $\mathrm{Rec}(\mathbb{R}(\Sigma, D))$ focus on asynchronous cellular automata, which we describe in the following. An *asynchronous cellular automaton* $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0, F)$ has for each letter $a \in \Sigma$ a set of local states Q_a and a local transition relation $\delta_a \subseteq (\prod_{b\in D(a)} Q_b) \times Q_a$, together with a global initial state $q_0 \in \prod_{a\in\Sigma} Q_a$ and a set of final states $F \subseteq \prod_{a\in\Sigma} Q_a$. The global transition relation $\delta \subseteq \prod_{a\in\Sigma} Q_a \times \Sigma \times \prod_{a\in\Sigma} Q_a$ is defined for $q = (q_a)_{a\in\Sigma}$, $q' = (q'_a)_{a\in\Sigma}$ by letting $q' \in \delta(q, a)$ if and only if

$$q'_a \in \delta_a((q_b)_{b\in D(a)}) \text{ and } q_c{}' = q_c, \text{ for all } c \neq a.$$

Thus, an a-transition affects only the local a-state and the change depends only on the local states of letters b with $(b,a) \in D$. The trace language accepted by $\mathcal{A}$ is $L(\mathcal{A}) = \{t \in \mathbb{M}(\Sigma, D) \mid \delta(q_0, t) \cap F \neq \emptyset\}$.

Asynchronous cellular automata are an inherently distributed recognition device, where independent transitions can be performed concurrently, in a Concurrent Read Owner Write way. Note that once an asynchronous cellular automaton has been constructed, we may transform it easily by a standard method (see e.g. [Pig93b]) into an equivalent, polynomially related asynchronous one, as originally defined by Zielonka [Zie87].

The interest in asynchronous cellular automata derives from Zielonka's theorem stating the equivalence between $\mathrm{Rec}(\mathbb{M}(\Sigma, D))$, the class of recognizable languages of finite traces, and the class of languages recognized by *deterministic* asynchronous cellular automata.

Asynchronous cellular automata showed to be an appropriate recognition device for languages of real traces, too. A distributed, i.e. on locality based counterpart of the classical Büchi and Muller acceptance conditions has been introduced in [GP92]. The idea is to augment an asynchronous cellular automaton by a table $\mathcal{T} \subseteq \prod_{a\in\Sigma} \mathcal{P}(Q_a)$ specifying local states which have to be repeated infinitely often. We omit further details here and refer to the definition given in Sect. 4.

We close this section with some general notations. For $t \in \mathbb{R}(\Sigma, D)$, $a \in \Sigma$ we denote by $|t|_a$ the number of occurrences of a in t; by $\mathrm{alph}(t) = \{a \in \Sigma \mid |t|_a > 0\}$ the set of letters occurring in t; finally, by $\mathrm{alphinf}(t) = \{a \in \Sigma \mid |t|_a = \infty\}$ the alphabet at infinity of t. For $A \subseteq \Sigma$ let $\mathrm{Inf}(A) = \{t \in \mathbb{R}(\Sigma, D) \mid \mathrm{alphinf}(t) = A\}$. For $t \in \mathbb{M}(\Sigma, D)$ let $\max(t) = \{a \in \Sigma \mid \exists w \in \Sigma^* : t = \varphi(wa)\}$ be the labellings of the maximal elements of t. A subalphabet $A \subseteq \Sigma$ is called *connected* if $(A, D|_{A\times A})$ is a connected graph. A trace t is connected if $\mathrm{alph}(t)$ is connected. The prefix order on $\mathbb{R}(\Sigma, D)$ is defined by $t \leq u$ if $u = tv$ for some $v \in \mathbb{R}(\Sigma, D)$. In this context, we denote v by $t^{-1}u$. As usual, for u, v we denote by $u \sqcap v$ the greatest lower bound of u, v with respect to the prefix order. Whenever it exists, the least upper bound of u, v is denoted by $u \sqcup v$.

3 Determinization for asynchronous cellular automata

This section presents a determinization construction for asynchronous cellular automata relying on the concept of asynchronous mapping, which forms the basis for Zielonka's construction. An asynchronous mapping can be computed stepwise in a distributed way, thus being easily transformed into (equivalent) deterministic asynchronous cellular automata. Before recalling the definition, let us introduce some further notations. For $t \in \mathbb{M}(\Sigma, D)$, $a \in \Sigma$ and $A \subseteq \Sigma$ let

$$\partial_a(t) = \sqcap\{\, u \leq t \mid |t|_a = |u|_a \,\} \quad \text{and} \quad \partial_A(t) = \bigsqcup_{a \in A} \partial_a(t)$$
$$\text{(In particular } \partial_\emptyset(t) = 1 \text{ and } \partial_\Sigma(t) = \partial_{\max(t)} = t.)$$

Thus, $\partial_a(t)$ resp. $\partial_A(t) = \bigsqcup\{u \leq t \mid \max(u) \subseteq A\}$ is the minimal prefix of t containing all a, resp. all letters $a \in A$ from t. Especially we have $\partial_a(ta) = \partial_{D(a)}(t)a$. In the following we will use the notation $\partial_{a,A}(t)$ instead of $\partial_a(\partial_A(t))$ (or simply $\partial_{a,b}(t)$, if $A = \{b\}$).

Definition 1 [CM88]. A mapping $\mu : \mathbb{M}(\Sigma, D) \longrightarrow S$ is called *asynchronous* if for every $t \in \mathbb{M}(\Sigma, D)$, $a \in \Sigma$ and $A, B \subseteq \Sigma$ the following conditions are satisfied.

- The value $\mu(\partial_{D(a)}(t))$ and the letter a uniquely determine the value $\mu(\partial_a(ta))$.
- The values $\mu(\partial_A(t))$ and $\mu(\partial_B(t))$ uniquely determine the value $\mu(\partial_{A \cup B}(t))$.

Given an asynchronous mapping $\mu : \mathbb{M}(\Sigma, D) \longrightarrow S$ and a subset $R \subseteq S$, a deterministic asynchronous cellular automaton $\mathcal{A}_\mu$ with $L(\mathcal{A}_\mu) = \mu^{-1}(R)$ is easily defined as follows: Let $\mathcal{A}_\mu = ((\tilde{Q}_a)_{a\in\Sigma}, (\tilde{\delta}_a)_{a\in\Sigma}, \tilde{q}_0, \tilde{F})$ with local states sets $\tilde{Q}_a = \{\mu(\partial_a(t)) \mid t \in \mathbb{M}(\Sigma, D)\}$, global transition function $\tilde{\delta}((\mu(\partial_b(t)))_{b\in\Sigma}, a) = (\mu(\partial_b(ta)))_{b\in\Sigma}$, $t \in \mathbb{M}(\Sigma, D)$, and $\tilde{q}_0 = (\mu(1))_{a\in\Sigma}$, $\tilde{F} = \{(\mu(\partial_a(t)))_{a\in\Sigma} \mid \mu(t) \in R\}$. It can be easily verified that $\mathcal{A}_\mu$ is asynchronous cellular and accepts exactly $\mu^{-1}(R)$ [CMZ93, Die90].
Consider a non-deterministic automaton $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0, F)$. It will suffice to exhibit a suitable asynchronous mapping μ based on $\mathcal{A}$, i.e. such that $\mu^{-1}\mu(L) = L$, where $L = L(\mathcal{A})$. The subset automaton will be then directly constructed as described above.
Recall now Zielonka's labelling function $\nu : \mathbb{M}(\Sigma, D) \longrightarrow \{0, \ldots, |\Sigma|\}^{\Sigma\times\Sigma}$, which is defined inductively as follows:

- $\nu(1)(a, b) = 0$.
- If $t \neq \partial_{a,b}(t)$ then $\nu(t)(a, b) = \nu(\partial_{a,b}(t))(a, a)$.
- If $t = \partial_a(t)$ and $t \neq 1$ then

$$\nu(t)(a, a) = \min\{n > 0 \mid n \neq \nu(t)(a, c) \text{ for every } c \neq a\}.$$

The labelling function ν is a time-stamping function, which allows to determine the actuality of information received in a distributed way. Especially, it provides crucial information about ordering of prefixes of the form $\partial_a(t)$. Formally, for every $t \in \mathbb{M}(\Sigma, D)$ and $a, b, c \in \Sigma$ we have $\partial_{c,a}(t) = \partial_{c,b}(t)$ if and only if

$\nu(t)(c,a) = \nu(t)(c,b)$. Note also $\nu(\partial_A(t))(c,a) = \nu(t)(c,a)$ for every $a \in A$. Moreover, given $t \in \mathbb{M}(\Sigma, D)$, $A, B \subseteq \Sigma$, suppose the sets $C_{a,b} = \{c \in \Sigma \mid \partial_{c,a}(t) = \partial_{c,b}(t)\}$ are known for every $a, b \in A \cup B$ (e.g. by knowing either $\nu(t)$ or the values $\nu(\partial_A(t))$, $\nu(\partial_B(t))$). Then we can easily determine for every $c \in \Sigma$ which of the three different situations $\partial_{c,A}(t) = \partial_{c,B}(t)$, $\partial_{c,A}(t) < \partial_{c,B}(t)$ resp. $\partial_{c,B}(t) < \partial_{c,A}(t)$ occurs.

Fact 2 ([CMZ93, Die90]) *The mapping ν is asynchronous.*

The crucial point in our determinization construction is to augment the mapping ν by a mapping ρ depending on the given non-deterministic automaton $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0, F)$.
For the rest of the paper we abbreviate the set of global states $\prod_{a\in\Sigma} Q_a$ by Q. Furthermore for $t \in \mathbb{R}(\Sigma, D)$ we denote by $R_{\mathcal{A}}(t)$ the set of runs of $\mathcal{A}$ on t, starting with the initial state q_0. We view a run $r \in R_{\mathcal{A}}(t)$ as a labelling $r : V_t \to \bigcup_{a\in\Sigma} Q_a$ of the dependence graph of t, $[V_t, E_t, \lambda_t]$, by local states, such that r is consistent with the alphabetical labelling λ_t and with the transition relations $(\delta_a)_{a\in\Sigma}$. To be precise, let $(a, n_a) \in V_t$ and let $q_b = r(b, n_b)$, where (b, n_b) is the last vertex with label b before (a, n_a) if it exists, and $q_b = (q_0)_b$ otherwise. Then we have $r(a, n_a) \in \delta_a((q_b)_{b\in D(a)})$. Finally, for $u \in \mathbb{M}(\Sigma, D)$, $r \in R_{\mathcal{A}}(u)$, we denote by $\delta(r, u) \in Q$ the global state reached in the run r on u.
Let now $\rho : \mathbb{M}(\Sigma, D) \to \mathcal{P}(Q^\Sigma)$ be defined for $t \in \mathbb{M}(\Sigma, D)$ by

$$\rho(t) = \{f \in Q^\Sigma \mid \exists r \in R_{\mathcal{A}}(t) \text{ s.t. for every } a \in \Sigma : \ f(a) = \delta(r, \partial_a(t))\}$$

Thus, every element $f : \Sigma \to Q$ of $\rho(t)$ corresponds to a run r on t, such that for every $a, b \in \Sigma$, $f(a)_b$ represents the local b-state reached in the run r on the prefix $\partial_{b,a}(t)$ of t.

Proposition 3. *The mapping (ν, ρ) is asynchronous. Moreover, we have $L(\mathcal{A}) = \{t \in \mathbb{M}(\Sigma, D) \mid \exists q = (q_a)_{a\in\Sigma} \in F,\ \exists f \in \rho(t)$ s.t. $f(a)_a = q_a$ for every $a \in \Sigma\}$.*

Proof. By Fact 2 we know that ν is asynchronous.
Let $t \in \mathbb{M}(\Sigma, D)$, $a \in \Sigma$, $A, B \subseteq \Sigma$. Given $\nu(\partial_{D(a)}(t))$, $\rho(\partial_{D(a)}(t))$ and a, we define $R \subseteq Q^\Sigma$ by letting $g \in R$ if and only if for some $f \in \rho(\partial_{D(a)}(t))$:

- $g(b) = f(b)$, for $b \neq a$;
- $g(a) = q' = (q_x{}')_{x\in\Sigma}$, where
  1. $q_x{}' = f(c)_x$ for $x \neq a$, with $c \in D(a)$ such that $\partial_{x,b}(t) \leq \partial_{x,c}(t)$ for every $b \in D(a)$ (i.e. $\partial_{x,c}(t) = \partial_{x,D(a)}(t)$);
  2. $q_a{}' \in \delta_a((f(b)_b)_{b\in D(a)})$

For each pair f, g as above, with $q = f(a)$, $q' = g(a)$, we will use in the next section the notation $q \xrightarrow{(a)} q'$. This means that a run $r \in R_{\mathcal{A}}(\partial_a(ta))$ exists, such that $q = \delta(r, \partial_a(t))$ and $q' = \delta(r, \partial_a(ta))$.
It is straightforward to see that $R = \rho(\partial_a(ta))$, since runs on $\partial_a(ta)$ are exactly extensions of runs on $\partial_{D(a)}(t)$ by an a-transition. Note also that the above conditions can be checked using $\nu(\partial_{D(a)}(t))$ and the property of ν mentioned before stating Fact 2, together with $\nu(t)(x,b) = \nu(\partial_{D(a)}(t))(x,b)$ for every $b \in D(a)$.

Consider now for $A, B \subseteq \Sigma$, $t_1 = \partial_A(t)$, $t_2 = \partial_B(t)$, $s = t_1 \sqcap t_2$ with $t_1 = su$ and $t_2 = sv$ ($\mathrm{alph}(u) \times \mathrm{alph}(v) \subseteq I$). Furthermore we denote by C the set $\{c \in \Sigma \mid \partial_c(t_1) = \partial_c(t_2)\} = \{c \in \Sigma \mid \nu(t_1)(c,c) = \nu(t_2)(c,c)\}$. Let $\rho(t_1), \rho(t_2)$ be given and define $R \subseteq Q^\Sigma$ by letting $f \in R$ if and only if for some $f_i \in \rho(t_i)$ ($i = 1, 2$) satisfying $f_1(c) = f_2(c)$ for every $c \in C$, we have for every $a, b \in \Sigma$:

$$f(a)_b = \begin{cases} f_1(a)_b & \text{if } \partial_a(t_2) \leq \partial_a(t_1) \\ f_2(a)_b & \text{if } \partial_{b,C}(t_2) < \partial_{b,a}(t_2) \\ f_1(d)_b & \text{otherwise, with } d \in \Sigma \text{ s.t. } \partial_{b,a}(t_2) = \partial_{b,d}(t_1) \end{cases}$$

We will see below that f is well-defined in the last case, too. The motivation for the above definition is based on the idea of combining runs on t_1 resp. t_2 if they yield the same global state after reading the common prefix s of t_1, t_2. With $\partial_C(t_1) = \partial_C(t_2) = \partial_C(s) = s = t_1 \sqcap t_2$ [CMZ93, Die90] we obtain for two runs $r_i \in R_{\mathcal{A}}(t_i)$ such that $f_i(a) = \delta(r_i, \partial_a(t_i))$, for every $a \in \Sigma$:

$$f_1(c) = f_2(c), \forall c \in C \quad \Longrightarrow \quad \delta(r_1, s) = \delta(r_2, s)$$

(With $\max(s) \subseteq C$ note that $\partial_a(s) = \partial_{a,c}(s) = \partial_{a,c}(t_i)$ for a suitable $c \in C$, $i = 1, 2$.) Hence, the mapping r labelling $t = t_1 \sqcup t_2$ with local states according to r_1 on su and to r_2 on v is a well-defined run of $\mathcal{A}$ on $t_1 \sqcup t_2$. The mapping f defined above corresponds to r, since we have for every $a, b \in \Sigma$:

1. $\partial_{b,a}(t) = \partial_{b,a}(t_1)$, if $\partial_a(t_2) \leq \partial_a(t_1)$;
2. $\partial_{b,a}(t) = \partial_{b,a}(t_2)$, if $\partial_{b,C}(t_2) < \partial_{b,a}(t_2)$. Note in this case $\partial_{b,a}(t) \not\leq s$.
3. In the remaining case we have $\partial_a(t_1) < \partial_a(t_2)$ and $\partial_{b,a}(t_2) \leq \partial_{b,C}(t_2)$. Assume first $\partial_{b,a}(t_2) \neq 1$ and let $C' = \{d \in \Sigma \mid \partial_{d,a}(t_2) = \partial_{d,C}(t_2) = \partial_d(s)\}$. We denote $s' = s \sqcap \partial_a(t_2)$ with $s = s'x$, $\partial_a(t_2) = s'y$ for suitable $x, y \in \mathbb{M}(\Sigma, D)$ with $\mathrm{alph}(x) \times \mathrm{alph}(y) \subseteq I$. Note that $\max(s') \subseteq C'$.
Then with $\partial_{b,a}(t_2) \leq \partial_{b,C}(t_2) = \partial_b(s)$ we obtain $\partial_{b,a}(t_2) = \partial_b(s') = \partial_{b,C'}(s)$ (in particular, $s' \neq 1$, thus $C' \neq \emptyset$), hence $\partial_{b,a}(t_2) = \partial_{b,d}(s') = \partial_{b,d}(s)$ for some $d \in \max(s') \subseteq C'$. It suffices to show now $\partial_d(s) = \partial_d(t_1)$. Due to the definition of C', s', x, y it follows immediately that $C' \cap \mathrm{alph}(x) = \emptyset$. Suppose now $d \in \mathrm{alph}(u)$. Since $d \in \max(s')$ there is some $e \in \mathrm{alph}(y)$, hence $e \in \mathrm{alph}(v)$, such that $(d, e) \in D$, which contradicts the assumption $d \in \mathrm{alph}(u)$ (note $y \neq 1$, otherwise we would obtain $\partial_a(t_2) \leq s$). We conclude by noting that $t_1 = s'xu$ and $d \notin \mathrm{alph}(xu)$ imply $\partial_d(t_1) = \partial_d(s')$ ($= \partial_d(s)$).
Hence, the claim $\partial_{b,a}(t_2) = \partial_{b,d}(t_1)$ holds.

For $\partial_{b,a}(t_2) \neq 1$ (i.e., $\nu(t_2)(b,a) \neq 0$) we compute effectively a letter d with $\partial_{b,a}(t_2) = \partial_{b,d}(t_1)$ as follows: using $\nu(t_1)$, $\nu(t_2)$ we first determine the sets C and $\mathrm{alph}(v)$ [CMZ93]; using again $\nu(t_2)$ we compute C'. Finally we choose $d \in C' \cap D(\mathrm{alph}(v))$ such that $\partial_{b,C}(t_1) \leq \partial_{b,d}(t_1)$ (which can be determined using $\nu(t_1)$). We obtain

$$\partial_{b,d}(t_1) \overset{d \notin \mathrm{alph}(u)}{=} \partial_{b,d}(s) \overset{d \in C'}{=} \partial_{b,d,a}(t_2) \leq \partial_{b,a}(t_2) \leq \partial_{b,C}(t_2) = \partial_{b,C}(t_1) \leq \partial_{b,d}(t_1)$$

Finally, if $\partial_{b,a}(t_2) = 1$, then $d = a$ satisfies $\partial_{b,a}(t_2) = \partial_{b,d}(t_1)$.
To conclude the proof, note that the values $\nu(t_1), \nu(t_2)$ allow to distinguish the three cases considered above.

Theorem 4. *Given a non-deterministic asynchronous cellular automaton* $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0, F)$ *we can effectively construct a deterministic asynchronous cellular automaton* $\tilde{\mathcal{A}} = ((\tilde{Q}_a)_{a\in\Sigma}, (\tilde{\delta}_a)_{a\in\Sigma}, \tilde{q}_0, \tilde{F})$ *recognizing the same trace language and having* $2^{O(N^{|\Sigma|})}$ *global states, where* $N = |\prod_{a\in\Sigma} Q_a|$ *is the number of global states of* $\mathcal{A}$.

To conclude this section, note that the direct determinization construction given by Zielonka can be applied to a non-deterministic asynchronous cellular automaton, too, by viewing it as a word automaton and determinizing it. Afterwards, one either computes its transformation monoid, which yields a double exponential blow-up in the number of global states of the given automaton, or better uses directly the deterministic word automaton [CMZ93], which yields a simple exponential blow-up. Note that in the present construction only reachable states are taken into account.

4 Complementing Büchi asynchronous cellular automata

The result presented in this section is an extension of Klarlund's *progress measure* technique [Kla91] to asynchronous cellular automata, using the subset automaton construction given in Sect. 3.

Our starting point is a slightly different Büchi acceptance condition, which specifies for each letter at most one local state to be repeated infinitely often, and in addition, the set of letters occurring infinitely often, i.e. the set alphinf(t). Formally, we consider a tuple $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0, \mathcal{T})$, where $\mathcal{T} \subseteq Q \times \mathcal{P}(\Sigma) \times \mathcal{P}(\Sigma)$ (recall $Q = \prod_{a\in\Sigma} Q_a$). A table element is a triple $(q^f, A, \{a_1, \ldots, a_k\})$ with the following restriction: with $A = \dot{\bigcup}_{i=1}^{k} A_i$ being the decomposition of A in connected components (i.e. every A_i is connected and $A_i \times A_j \subseteq I$ for $i \neq j$), we require $a_i \in A_i$ for every $1 \leq i \leq k$.

A run $r \in R_{\mathcal{A}}(t)$ on t is accepted by table element $(q^f, A, \{a_1, \ldots, a_k\})$ if

- $A = \text{alphinf}(t)$ and
- For every $a \in \bar{A} \cup \{a_1, \ldots, a_k\}$ we have $q^f_a \in \inf_a(r)$, where $\inf_a(r) := \{q_a \in Q_a \mid \forall n < |t|_a \; \exists n \leq m < |t|_a : \; r(a,m) = q_a\}$.

Note that an asynchronous cellular Büchi automaton with acceptance table $\mathcal{T} \subseteq \prod_{a\in\Sigma} \mathcal{P}(Q_a)$ like in Sect. 2 can be easily transformed into an equivalent one with acceptance table like above: once we have fixed the alphabet at infinity A, we use a_i, $1 \leq i \leq k$, in order to gather information about the states from $\bigcup_{a\in A_i} T_a$ which have occurred, where $T = (T_a)_{a\in\Sigma} \in \mathcal{T}$. The converse transformation is straightforward, since we only have to check in addition (non-deterministically) the set of letters occurring infinitely often.

For $t \in \mathbb{R}(\Sigma, D)$, $a \in \Sigma$, $0 \leq n < |t|_a$, let $t[a,n] = \sqcap\{u \leq t \mid |u|_a = n+1\}$ be the least prefix of t containing the first $n+1$ occurrences of a (note $\max(t[a,n]) = \{a\}$). Furthermore, let

$$U_a(t) = \{(q,n) \mid 0 \leq n < |t|_a, \; q \in \delta(q_0, t[a,n])\}$$

be the set of global states reachable by prefixes $t[a,n]$ of t.
For $n+1 < |t|_a$ and $q, q' \in U_a(t)$ with $q' \in \delta(q, t[a,n]^{-1}t[a,n+1])$ we use the notation $(q,n) \xrightarrow{a,t} (q', n+1)$.
Throughout this section we use the notion of computation (sub-) graph with the meaning of (a subgraph of) $(U_a(t), \xrightarrow{a,t})$.
The next proposition gives the basis of the complement automaton. We denote $N := |Q|$ and $F_i := \{q = (q_a)_{a\in\Sigma} \in Q \mid q_{a_i} = q^f_{a_i}\}$, where $q^f = (q^f_a)_{a\in\Sigma}$ will be the first component in the unique table element of the given automaton. Moreover we use the abbreviation $U_i(t)$ for $U_{a_i}(t)$, $1 \le i \le k$.

Proposition 5. *Let $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0, \mathcal{T})$ be an asynchronous cellular Büchi automaton, with table $\mathcal{T} = \{(q^f, A = \dot{\bigcup}_{i=1}^{k} A_i, \{a_1, \ldots, a_k\})\}$ like above. Let $t \in \mathbb{R}(\Sigma, D)$ with* $\mathrm{alphinf}(t) = A$.
Then we have $t \notin L(\mathcal{A})$ if and only if there exists a family of computation subgraphs $G_i(t) = (V_i(t), E_i(t))$, $1 \le i \le k$, together with associated mappings $(\Phi_i)_{1\le i\le k}$, $\Phi_i : U_i(t) \to \{0, 1, \ldots, 2N+1\}$, such that the following conditions hold for every i:

1. – $V_i(t) = \{(q,n) \in U_i(t) \mid \Phi_i(q,n) \ne 2N+1\}$,
   – $E_i(t) \subseteq \{((q,n),(q',n+1)) \in V_i(t)^2 \mid (q,n) \xrightarrow{a_i,t} (q',n+1)\}$,
   – *For every $(q,n) \xrightarrow{a_i,t} (q',n+1)$ with $(q,n) \in V_i(t)$, we have $(q',n+1) \in V_i(t)$, too.*
2. *Each mapping $\Phi_i : U_i(t) \to \{0,1,\ldots,2N+1\}$ satisfies for every (q,n), $(q',n+1) \in U_i(t)$ the following properties:*
   (a) *Φ_i is decreasing w.r.t. the transition relation $\xrightarrow{a_i,t}$, i.e., $(q,n) \xrightarrow{a_i,t} (q',n+1)$ implies $\Phi_i(q,n) \ge \Phi_i(q',n+1)$.*
   (b) *For $(q,n) \xrightarrow{a_i,t} (q',n+1)$ with $\Phi_i(q,n) = \Phi_i(q',n+1)$ we have either $q' \notin F_i$ or $\Phi_i(q,n) \in \{0,2,\ldots,2N\} \cup \{2N+1\}$.*
   (c) *The set $\{0,2,\ldots,2N\}$ contains only values of real progress for Φ_i: Let $((q_n,n))_{n\ge 0} \subseteq V_i(t)$ be an infinite sequence with $(q_n,n) \xrightarrow{a_i,t} (q_{n+1},n+1)$, $n \ge 0$. Then $\lim_{n\to\infty} \Phi_i(q_n,n) \in \{1,3,\ldots,2N-1\}$.*
3. *There exists a finite prefix $t_0 \le t$, $t_0 \in \mathbb{M}(\Sigma,D)$, with $|t_0|_a = |t|_a$ for every $a \in \bar{A}$, such that every run $r \in R_{\mathcal{A}}(t_0)$ satisfies*
   – *either $\delta(r,t_0)_a \ne q^f_a$ for some $a \in \bar{A}$,*
   – *or $(\delta(r, \partial_{a_i}(t_0)), |t_0|_{a_i} - 1) \in V_i(t)$, for some $1 \le i \le k$.*

Remark 6. Following Klarlund [Kla91], each mapping is a quasi-progress measure, for which the values $\{0,2,\ldots,2N\}$ denote real progress. The progress value for a vertex v in a computation subgraph measures how close paths starting in v are from repeating the corresponding component of the final state finitely often.

Condition 3 guarantees that computations *not* covered (locally) by some quasi-progress measures (Φ_i), $1 \le i \le k$, can not be synchronized to a run on t. This means that every run $r \in R_{\mathcal{A}}(t)$ is rejecting either due to $\inf_a(r) \ne \{q^f_a\}$ for some $a \notin \mathrm{alphinf}(t)$, or due to r being covered by some Φ_i, i.e. there is a $n_i \ge 0$ such that $(\delta(r, t[a_i,n_i]), n_i) \in V_i(t)$ (hence, $(\delta(r, t[a_i,n]), n) \in V_i(t)$ for every $n \ge n_i$).

Proof of 5: We omit the implication from right to left for lack of space. For the converse direction, let $t \notin L(\mathcal{A})$ be such that alphinf$(t) = A$. Following Klarlund [Kla91], we first define progress measures $\tilde{\Phi}_i$ with values in the set of countable ordinals ω_1. We denote in the following for $(q,n) \in U_i(t)$ by $N_+(q,n)$ the set of proper successors of (q,n) in $(U_i(t) \setminus V_i(t), \stackrel{a_i,t}{\rightarrow})$, i.e.,

$$N_+(q,n) = \{(q',m) \mid m > n,\ \exists q = q_n, q_{n+1}, \ldots, q_m = q' \in U_i(t) \setminus V_i(t) \\ \text{with } (q_k,k) \stackrel{a_i,t}{\rightarrow} (q_{k+1}, k+1), \text{ for every } n \le k < m\}$$

The computation subgraph $G_i(t) = (V_i(t), E_i(t))$ and the progress measure $\tilde{\Phi}_i : U_i(t) \longrightarrow \omega_1$ are now defined by transfinite induction as follows (see also [Kla91]): Let $V_0 = V_i(t) = \emptyset$. Assume that for $\beta < \omega_1$, the sequence $(V_\alpha)_{\alpha<\beta}$ of pairwise disjoint subsets of $U_i(t)$ with $V_i(t) = \cup_{\alpha<\beta} V_\alpha$ is already defined.
If $N_+(q,n) \cap (F_i \times \mathbb{N}) \neq \emptyset$ for every $(q,n) \in U_i(t) \setminus V_i(t)$, then we let $V_\beta := U_i(t) \setminus V_i(t)$ and $V_\gamma = \emptyset$ for every $\beta < \gamma < \omega_1$.
Otherwise we choose $(q,n) \in U_i(t) \setminus V_i(t)$ such that $N_+(q,n) \cap (F_i \times \mathbb{N}) = \emptyset$ and define

$$V_\beta := \begin{cases} \{(q,n)\} & \text{if } N_+(q,n) = \emptyset \\ N_+(q,n) & \text{otherwise,} \end{cases}$$

updating $V_i(t) = V_i(t) \cup V_\beta$.
The mapping $\tilde{\Phi}_i : U_i(t) \longrightarrow \omega_1$ is now defined by letting $\tilde{\Phi}_i(q,n) := \beta$ for $(q,n) \in V_\beta$ (note that $V_\alpha \cap V_{\alpha'} = \emptyset$ for every $\alpha \neq \alpha' < \omega_1$).
It is not hard to see that $\tilde{\Phi}_i$ is decreasing w.r.t. the transition relation $\stackrel{a_i,t}{\rightarrow}$, due to the definition by means of successors. Moreover, $\tilde{\Phi}_i(q,n) = \tilde{\Phi}_i(q',n+1)$ for $(q,n) \stackrel{a_i,t}{\rightarrow} (q',n+1)$ with $(q,n),(q',n+1) \in V_i(t)$, implies

$$q' \notin F_i. \tag{1}$$

By construction we obtain a countable ordinal $\beta_0 < \omega_1$ with $V_{\beta_0} = U_i(t) \setminus V_i(t)$, such that $\beta_0 = \sqcup\{\alpha < \omega_1 \mid V_\alpha \neq \emptyset\}$. Note that we have either $V_{\beta_0} = \emptyset$, or for every $(s_n,n) \in V_{\beta_0}$ there is an infinite transition path $(s_n,n) \stackrel{a_i,t}{\rightarrow} (s_{n+1},n+1) \stackrel{a_i,t}{\rightarrow} \cdots$ in V_{β_0}, repeating infinitely often some state from F_i, i.e.,

$$|\{m \ge n \mid s_m \in F_i\}| = \infty. \tag{2}$$

In a second step, following again [Kla91], we define a *quasi-progress* measure $\Phi_i : U_i(t) \longrightarrow \{0,1,\ldots,2N+1\}$ by weakening the mapping $\tilde{\Phi}_i$ and obtaining thus a finite value domain. Given $\alpha < \omega_1$, let the predicate const(α) be true, if there is an infinite transition path $(q_n,n) \stackrel{a_i,t}{\rightarrow} (q_{n+1},n+1) \stackrel{a_i,t}{\rightarrow} \cdots$ in $U_i(t)$, such that $\tilde{\Phi}_i(q_m,m) = \alpha$ for every $m \ge n$. The crucial point now is the bounded width of the computation graph $(U_i(t), \stackrel{a_i,t}{\rightarrow})$, which yields with the pigeon-hole principle a set of at most N countable ordinals $0 < \alpha_1 < \ldots < \alpha_M < \omega_1$ $(M \le N)$ with const(α_i), $1 \le i \le M$. Let $\alpha_0 = 0$, $\alpha_{M+1} = \omega_1$ and define $\Phi_i : U_i(t) \longrightarrow \{0,1,\ldots,2N+1\}$ for $(q,n) \in V_i(t)$ as

$$\Phi_i(q,n) = \begin{cases} 2k-1 & \text{if } \tilde{\Phi}_i(q,n) = \alpha_k,\ 1 \le k \le M \\ 2k & \text{if } \alpha_k < \tilde{\Phi}_i(q,n) < \alpha_{k+1},\ 0 \le k \le M, \end{cases}$$

otherwise let $\Phi_i(q,n) = 2N+1$.
The mapping Φ_i satisfies Cond. (2c): assume by contradiction that an infinite transition path $(q_n, n) \xrightarrow{a_i,t} (q_{n+1}, n+1) \xrightarrow{a_i,t} \cdots$ exists in $U_i(t)$, such that $\Phi_i(q_m, m) = 2k$ for some $k \le M$, $\forall m \ge n$. With $\tilde{\Phi}_i$ being decreasing, we would obtain an ordinal $\alpha_k < \alpha < \alpha_{k+1}$ and some $n' \ge n$ with $\tilde{\Phi}_i(q_m, m) = \alpha$ for every $m \ge n'$. Thus, const(α) would hold, hence contradicting the definition of $\{\alpha_1, \ldots, \alpha_M\}$.
Finally, we show that runs uncovered by the quasi-progress measures $(\Phi_i)_{1\le i\le k}$ cannot be synchronized (Cond. 3). Let us assume by contradiction that for every $(n_i)_{1\le i\le k} \in \mathbb{N}^k$ there exist global states $(q_i)_{1\le i\le k} \in Q^k$ and a run r on the (finite) prefix $t_0 := (\bigsqcup_{1\le i\le k} t[a_i, n_i]) \sqcup (\bigsqcup_{a\in\bar{A}\cap \mathrm{alph}(t)} t[a, |t|_a - 1])$ of t with

- $(q_i, n_i) \in U_i(t) \setminus V_i(t)$ for every $1 \le i \le k$,
- $\delta(r, t[a_i, n_i]) = q_i$ for every $1 \le i \le k$, and
- $\delta(r, t_0)_a = q_a^f$ for every $a \in \bar{A}$.

Assume now without loss of generality that $\max(t_0) \cap A = \{a_1, \ldots, a_k\}$ and $\mathrm{alph}(t[a_i, n_i]^{-1} t[a_i, n]) \subseteq A_i$, for every $n > n_i$, $1 \le i \le k$. Let $t = t_0 t_1 \cdots t_k$ with $\mathrm{alph}(t_i) = A_i$ for every $1 \le i \le k$. Due to $(q_i, n_i) \in U_i(t) \setminus V_i(t)$ together with (2) we obtain for each i a run r_i on the connected t-suffix t_i starting in q_i and repeating infinitely often some state from F_i, i.e. $q_{a_i}^f \in \mathrm{inf}_{a_i}(r_i)$. Let r' be a run on t corresponding to r on t_0 respectively to r_i on t_i, $1 \le i \le k$. Obviously, $q_a^f \in \mathrm{inf}_a(r')$ for every $a \in \bar{A} \cup \{a_1, \ldots, a_k\}$, hence $t \in L(\mathcal{A})$ contradicting the assumption. □

We are now ready to define an asynchronous cellular Büchi automaton $\mathcal{B}$ with $L(\mathcal{B}) = \overline{L(\mathcal{A})} \cap \mathrm{Inf}(A)$, where $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0, \{(q^f, A, \{a_1, \ldots, a_k\})\})$. Following Klarlund [Kla91], the automaton guesses the values of the quasi-progress measures. Here we guess at the same time the computation subgraphs $(V_i(t), E_i(t))$, "covered" each by the mapping Φ_i.
The complement automaton relies on the subset automaton of the given automaton $((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0)$, defined in Sect. 3 as $\mathcal{A}_\mu$ for $\mu = (\nu, \rho)$. Let $\mathcal{A}_\mu = ((\tilde{Q}_a)_{a\in\Sigma}, (\tilde{\delta}_a)_{a\in\Sigma}, \tilde{q}_0)$ be the subset automaton of $((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0)$ (we omit here final state sets). In the following we denote by $[2N+1]_p^Q$ the set of partial mappings from Q to $\{0, 1, \ldots, 2N+1\}$. Furthermore, dom(f) denotes the domain of $f \in [2N+1]_p^Q$.
Let $\mathcal{B} = ((S_a)_{a\in\Sigma}, (\Delta_a)_{a\in\Sigma}, s_0, \mathcal{T})$ be defined by:

1. $$S_a = \begin{cases} \tilde{Q}_a & \text{if } a \notin \{a_1, \ldots, a_k\} \\ \tilde{Q}_a \times [2N+1]_p^Q \times \mathcal{P}(Q) & \text{otherwise} \end{cases}$$

   Moreover, for every $a \in \{a_1, \ldots, a_k\}$ and $(\tilde{q}_a, \alpha_a, A_a) \in S_a$ we have

   $$\mathrm{dom}(\alpha_a) = \{f(a) \mid f \in R, \text{ where } \tilde{q}_a = (N, R)\},$$

   with N (resp. R) denoting the ν (resp. ρ) component of the local subset automaton state.

2. For $a \notin \{a_1, \dots, a_k\}$ let $s'_a \in \Delta_a((s_b)_{b\in D(a)})$ if and only if $s'_a = \tilde{\delta}_a((\tilde{q}_b)_{b\in D(a)})$, where for every $b \in D(a)$: $s_b = \tilde{q}_b$ or $s_b = (\tilde{q}_b, \alpha_b, A_b)$ for some $\alpha_b \in [2N+1]_p^Q$, $A_b \subseteq Q$.
   For $a = a_i$ $(1 \le i \le k)$, let $s_a = (\tilde{q}_a, \alpha_a, A_a)$ and $s'_a = (\tilde{q}'_a, \alpha'_a, A'_a)$. Then $s'_a \in \Delta_a((s_b)_{b\in D(a)})$ if and only if
   - $\tilde{q}'_a = \tilde{\delta}((\tilde{q}_b)_{b\in D(a)})$, for $\tilde{q}_b = s_b$, $b \in D(a) \setminus \{a\}$;
   - For every $q \in \mathrm{dom}(\alpha_a)$, $q' \in \mathrm{dom}(\alpha'_a)$ such that $q \xrightarrow{(a)} q'$ (recall this notation from the the proof of Prop. 3) we have
     (a) $\alpha_a(q) \ge \alpha'_a(q')$ and
     (b) $\alpha_a(q) = \alpha'_a(q')$ implies $q'_a \ne q^f_a$ or $\alpha_a(q) \in \{0, 2, \dots, 2N\}$ or $\alpha_a(q) = 2N+1$.
   - $$A'_a = \begin{cases} \mathrm{dom}(\alpha'_a) & \text{if } A_a = \emptyset \\ \{q' \in \mathrm{dom}(\alpha'_a) \mid \exists q \in \mathrm{dom}(\alpha_a) \cap A_a \text{ with} & \\ \qquad q \xrightarrow{(a)} q' \text{ and } \alpha_a(q) = \alpha'_a(q') \in \{0, 2, \dots, 2N\}\} & \text{otherwise} \end{cases}$$

3. $(s^f, A, \{a_1, \dots, a_k\}) \in \mathcal{T}$ if and only if for some $u \in \mathbb{M}(\Sigma, D)$ with $\tilde{q}_a = (\nu(\partial_a(u)), \rho(\partial_a(u)))$, $a \in \Sigma$, and mappings $\alpha_a \in [2N+1]_p^Q$, $a \in \{a_1, \dots, a_k\}$:
   (a) $s^f_a = (\tilde{q}_a, \alpha_a, \emptyset)$, for every $a \in \{a_1, \dots, a_k\}$, respectively $s^f_a = \tilde{q}_a$ otherwise.
   (b) Let $C = \bar{A} \cup \{a_1, \dots, a_k\}$. Then there is no $f \in \rho(\partial_C(u))$ with
     - $f(a)_a = q^f_a$, for every $a \in \bar{A}$ and
     - $f(a) \in \alpha_a^{-1}(2N+1)$, for every $a \in \{a_1, \dots, a_k\}$.

With Prop. 5 the proof of the following proposition follows without difficulties.

Proposition 7. *Let $\mathcal{B}$ be defined as above. Then $L(\mathcal{B}) = \overline{L(\mathcal{A})} \cap \mathrm{Inf}(A)$.*

Theorem 8. *Given a (non-deterministic) asynchronous cellular Büchi automaton $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0, \mathcal{T})$ with table $\mathcal{T} \subseteq (\prod_{a\in\Sigma} Q_a) \times \mathcal{P}(\Sigma) \times \mathcal{P}(\Sigma)$ and having N global states.*
Then we can construct effectively an asynchronous cellular Büchi automaton $\mathcal{B} = ((S_a)_{a\in\Sigma}, (\Delta_a)_{a\in\Sigma}, s_0, \mathcal{T}')$, with table $\mathcal{T}' \subseteq (\prod_{a\in\Sigma} S_a) \times \mathcal{P}(\Sigma) \times \mathcal{P}(\Sigma)$, such that $L(\mathcal{B}) = \overline{L(\mathcal{A})}$. The automaton $\mathcal{B}$ has $2^{O(N^{|\Sigma|})}$ global states, if $|\Sigma| \ge 2$.

Asynchronous cellular automata have in particular the so-called I-diamond property, i.e. for every two states q, q' and independent letters $(a, b) \in I$, $q \xrightarrow{ab} q'$ holds if and only if $q \xrightarrow{ba} q'$. It is worth noting that if we were only interested in I-diamond word automata, then Pécuchet's method [Péc86] applies, yielding an automaton with $2^{O(N^2)}$ states.

Acknowledgment: I wish to thank Volker Diekert and Paul Gastin for valuable discussions and Wolfgang Thomas for pointing out the work of J.-P. Pécuchet.

References

[CF69] P. Cartier and D. Foata. *Problèmes combinatoires de commutation et réarrangements.* Number 85 in Lecture Notes in Mathematics. Springer, 1969.

[CM88] R. Cori and Y. Métivier. Approximation of a trace, asynchronous automata and the ordering of events in a distributed system. In *Proc. of the 15th ICALP*, LNCS 317, pp. 147–161. 1988.

[CMZ93] R. Cori, Y. Métivier, and W. Zielonka. Asynchronous mappings and asynchronous cellular automata. *Information and Computation*, 106:159–202, 1993.

[DE92] V. Diekert and W. Ebinger, eds. *Infinite Traces. Proc. of a workshop of the ESPRIT BRA No 3166 (ASMICS), Tübingen, 1992*, Ber. 4/92. Univ. Stuttgart.

[Die90] V. Diekert. *Combinatorics on Traces.* No. 454 in LNCS. Springer, 1990.

[DM93] V. Diekert and A. Muscholl. Deterministic asynchronous automata for infinite traces. In *Proc. of the 10th STACS*, LNCS 665, pp. 617–628, 1993.

[Gas91] P. Gastin. Recognizable and rational trace languages of finite and infinite traces. In *Proc. of the 8th STACS*, LNCS 480, pp. 89–104, 1991.

[GP92] P. Gastin and A. Petit. Asynchronous automata for infinite traces. In *Proc. of the 19th ICALP*, LNCS 623, pp. 583–594, 1992.

[Kla91] N. Klarlund. Progress measures for complementation of ω-automata with applications to temporal logic. In *Proc. of the 32nd FOCS*, Oct. 1-4, 1991, San Juan, Puerto Rico, pp. 358–367, 1991.

[KMS94] N. Klarlund, M. Mukund, and M. Sohoni. Determinizing asynchronous automata. This volume.

[Maz77] A. Mazurkiewicz. Concurrent program schemes and their interpretations. DAIMI Rep. PB 78, Aarhus University, Aarhus, 1977.

[Péc86] J.-P. Pécuchet. On the complementation of Büchi automata. *Theor. Comp. Sci.*, (47):95–98, 1986.

[Pig93a] G. Pighizzini. Personal communication, 1993.

[Pig93b] G. Pighizzini. Recognizable trace languages and asynchronous automata. PhD thesis, Dip. Scienze dell'Informazione, Università di Milano, 1993.

[Saf88] S. Safra. On the complexity of ω-automata. In *Proc. of the 29th FOCS*, 1988.

[Zie87] W. Zielonka. Notes on finite asynchronous automata. *R.A.I.R.O. — Informatique Théorique et Applications*, 21:99–135, 1987.