On the complementation of asynchronous cellular Büchi automata *

Anca Muscholl

Universität Stuttgart, Institut für Informatik
Breitwiesenstr. 20-22, 70565 Stuttgart, Germany

Abstract

We present direct subset automata constructions for asynchronous (asynchronous cellular, resp.) automata. This provides a solution to the problem of direct determinization for automata with distributed control for languages of finite traces. We use the subset automaton construction and apply Klarlund's progress measure technique in order to complement non-deterministic asynchronous cellular Büchi automata for infinite traces. Both constructions yield a super-exponential blow-up in the size of local states sets.

1 Introduction

Infinite Mazurkiewicz traces provide a sound framework for studying non-terminating concurrent systems, such as e.g. distributed operating systems or transaction systems. Basically, a concurrent system is viewed as a labelled partial order of a special form. The labelling corresponds to a (finite) set Σ of atomic actions. The partial ordering is based on a fixed symmetric, reflexive dependence relation $D \subseteq \Sigma \times \Sigma$, denoting pairs of actions which cannot be executed in parallel. Especially interesting are systems where the behaviour can be described by finite state devices. The family of recognizable languages of infinite traces, $\mathrm{Rec}(\mathbb{R}(\Sigma, D))$, has been introduced by means of recognizing homomorphisms [9]. Various characterizations have been obtained for this class, including automata-theoretic and logical aspects [10, 6, 8], which generalize the well-understood framework of ω-languages. We are interested in automata with distributed control, more precisely in asynchronous (asynchronous cellular, resp.) automata. They play a basic role in the theory of Mazurkiewicz traces, as finite automata do for sequential systems. For example, Zielonka's important theorem states the equivalence between

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

recognizability for languages of finite traces and acceptance by deterministic asynchronous cellular automata. For languages of infinite traces a natural counterpart of the classical Büchi acceptance condition [10] yields the equivalence between $\mathrm{Rec}(\mathbb{R}(\Sigma, D))$ and the class of languages of infinite traces which are accepted by non-deterministic Büchi asynchronous cellular automata. As in the special case of ω-languages, recognizability of languages of infinite traces can also be characterized by deterministic automata, by using the analogue of the more powerful Muller acceptance condition [6]. This generalizes McNaughton's theorem to languages of infinite traces. All these results hold also for the asynchronous automaton model, due to a straightforward transformation from asynchronous cellular to asynchronous automata.
The closure of $\mathrm{Rec}(\mathbb{R}(\Sigma, D))$ under complementation is easily seen from the definition by recognizing homomorphisms (or the acceptance by deterministic Muller automata). We present in this paper a direct proof based on automata by exhibiting a complementation procedure for non-deterministic asynchronous cellular Büchi automata. We use the notion of progress measures, which has been introduced by Klarlund [11] for complementing Büchi (and Streett) ω-automata. We will apply progress measures locally to computation subgraphs of asynchronous automata.
A basic component of several constructions for ω-automata [20, 11] is the usual subset automaton of Rabin and Scott. In the case of asynchronous automata, no subset construction has been so far available, in spite of several attempts [18, 5]. We present in Sect. 3 subset constructions for asynchronous (asynchronous cellular, resp.) automata relying on Cori/Métivier's notion of asynchronous mapping [2]. Based on the bounded time-stamping of Zielonka's construction we exhibit natural mappings which turn out to be asynchronous and thus can be directly translated into asynchronous (asynchronous cellular, resp.) automata. The deterministic automata obtained provide the full information of asynchronous subset automata. The highly technical part is the proof of correctness, i.e. showing that the above mappings are asynchronous, which requires a detailed analysis of prefix relations in a given trace.
We consider here two models of trace automata with distributed control (asynchronous and asynchronous cellular, resp.), due to size considerations which may be significant and determine their practical use. Asynchronous automata seem to be more advantageous, since they are more compact, in general; the cellular model is easier to understand, due to canonical relations to the prefix structure of a trace. The determinization ideas are closely related and in both cases a superexponential blow-up of size results. However we are interested in the precise size of the subset automata obtained.
Related ideas with respect to determinization have been developed independently in [12], where a subset construction for asynchronous automata essentially matching the lower bound is presented. We note that the superexponential lower bound for the size blow-up obtained in [12] holds by the same argument for the cellular

model, too.
We consider infinite traces only in Sect. 4, where we use the subset automaton introduced previously and apply the progress measure of Klarlund [11] to asynchronous cellular Büchi automata. We obtain a size blow-up for the global state space of $2^{N^{O(1)}}$. Note that we can obtain a deterministic automaton for the complementation problem using deterministic asynchronous Muller automata, applying the algebraic construction given in [6]. This yields however a blow-up for the set of global states at least doubly exponential, since we have to compute the syntactic monoid and to apply usual determinization, as well as Zielonka's construction.
A preliminary version of this paper appeared in [16]. In this paper we additionally consider the asynchronous model with regard to determinization constructions. Moreover, we present here a more involved subset construction which improves the size of the subset automata.

2 Preliminaries

Throughout this paper we denote by (Σ, D) a finite dependence alphabet, i.e. a finite alphabet Σ together with a reflexive, symmetric dependence relation $D \subseteq \Sigma\times\Sigma$. The complementary relation $I = (\Sigma\times\Sigma)\setminus D$, called *independence relation*, induces an equivalence relation $\equiv_I$ on Σ^*, generated by the pairs $(uabv, ubav)$, with $u, v \in \Sigma^*$, $(a, b) \in I$. The relation $\equiv_I$ turns out to be a congruence. The quotient monoid $\mathbb{M}(\Sigma, D) = \Sigma^*/\equiv_I$ was called *trace monoid* by Mazurkiewicz [13] and it was first used in combinatorics for rearrangement problems [1]. The canonical surjective homomorphism associated to $\equiv_I$ will be denoted by $\varphi\colon \Sigma^* \to \mathbb{M}(\Sigma, D)$. For $a \in \Sigma$, $A \subseteq \Sigma$, let $D(a) = \{b \in \Sigma \mid (a, b) \in D\}$ and $D(A) = \bigcup_{a\in A} D(a)$.
By definition, a trace t is a congruence class of words and it may be represented by words. A natural, unique representation is given by considering labelled partial orders. We identify a trace t with a *dependence graph*, i.e. with a (isomorphism class of a) labelled directed, acyclic graph $G = [V, E, \lambda]$, where $\lambda\colon V \to \Sigma$ 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 the trace $t = [a_1 \cdots a_n]$, we define the associated dependence graph by taking n vertices $V = \{1, \ldots , n\}$ labelled as $\lambda(i) = a_i$, with edges $(i, j) \in E$ for $1 \le i < j \le n$ whenever $(a_i, a_j) \in D$.
In this paper we consider finite and infinite 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 $1 = [\emptyset, \emptyset, \emptyset]$. The requirement that any subset of vertices with

the same labelling is well-ordered allows us to represent vertices as pairs (a, i), with $a \in \Sigma$ and $i \geq 0$ a countable ordinal, where (a, i) represents the $(i+1)$-th node labelled by the letter a. This is called the *standard representation*.
We use here the usual notion of infinite traces, where every vertex in the dependence graph has a finite past. This corresponds to the ideal completion of $\mathbb{M}(\Sigma, D)$ under the prefix ordering, which has been already considered by Mazurkiewicz [14]. Infinite traces with this property are called *real traces* and the set of real traces is denoted $\mathbb{R}(\Sigma, D)$. Equivalently, real traces correspond to finite and infinite dependence graphs having a representation by words; i.e. with $\varphi\colon \Sigma^\infty \to \mathbb{G}(\Sigma, D)$ being the extension of the canonical mapping to 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 $a^\omega b \notin \mathbb{R}(\Sigma, D)$).
The notion of recognizability provides an interesting language class in various contexts of finite and infinite objects: words, traces, trees, graphs. For real traces a language L is *recognizable* if $\varphi^{-1}(L) \subseteq \Sigma^\infty$ is recognizable in the usual sense for languages over Σ^∞. The class of recognizable real trace languages is denoted by $\mathrm{Rec}(\mathbb{R}(\Sigma, D))$.
A natural finite state device for trace languages is the asynchronous model introduced by Zielonka [21]. By asynchronous automata we mean two types of automata of equal expressive power, both of which have distributed control and memory. The difference consists mainly in the kind of restriction imposed on the concurrent access to common data. Asynchronous automata belong to the Exclusive-Read-Exclusive-Write type, while asynchronous cellular automata correspond to the Concurrent-Read-Owner-Write access restriction.
An *asynchronous cellular automaton* $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0, F)$ has for each $a \in \Sigma$ a set of local states Q_a and a local transition relation $\delta_a \subseteq (\prod_{b\in D(a)} Q_b) \times Q_a$. The set of global states is $\prod_{a\in\Sigma} Q_a$, and $q_0 \in \prod_{a\in\Sigma} Q_a$ resp. $F \subseteq \prod_{a\in\Sigma} Q_a$ denotes the initial state, resp. the set of final states of $\mathcal{A}$. 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

$$q' \in \delta(q, a) \quad :\Longleftrightarrow \quad \begin{array}{ll} q'_a \in \delta_a((q_b)_{b\in D(a)}) & \text{and} \\ q'_c = q_c, \quad \text{for } c \neq a\,. & \end{array}$$

Thus, an a-transition changes only the local a-state and the change depends only on the local states of letters b with $(b, a) \in D$. Note that for any $q, q' \in \prod_{a\in\Sigma} Q_a$, $u, v \in \Sigma^*$ with $u \equiv_I v$ and $q' \in \delta(q, u)$ also $q' \in \delta(q, v)$ holds. Hence, we may define the trace language accepted by $\mathcal{A}$ by $L(\mathcal{A}) = \{t \in \mathbb{M}(\Sigma, D) \mid \delta(q_0, t) \cap F \neq \emptyset\}$.
The asynchronous automaton model, as originally considered by Zielonka [21], associates to each letter $a \in \Sigma$ a set $\mathrm{dom}(a) \subseteq \{1, \ldots, m\}$ of processors representing the read- and write-domain, such that $(a, b) \in I \Leftrightarrow \mathrm{dom}(a) \cap \mathrm{dom}(b) = \emptyset$. An *asynchronous automaton* $\mathcal{A}$ is a tuple $((Q_i)_{i=1}^m, (\delta_a)_{a\in\Sigma}, q_0, F)$, with a local transition relation $\delta_a \subseteq \prod_{i\in\mathrm{dom}(a)} Q_i \times \prod_{i\in\mathrm{dom}(a)} Q_i$ for each letter $a \in \Sigma$. The

set of global states is $\prod_{i=1}^m Q_i$, and $q_0 \in \prod_{i=1}^m Q_i$, resp. $F \subseteq \prod_{i=1}^m Q_i$ denotes the initial state, resp. the set of final states of $\mathcal{A}$. A global transition step $q' \in \delta(q,a)$, $q = (q_i)_{1 \le i \le m}$, $q' = (q'_i)_{1 \le i \le m}$, is defined by

$$q' \in \delta(q,a) \quad :\Longleftrightarrow \quad \begin{array}{l} (q'_i)_{i \in \mathrm{dom}(a)} \in \delta_a((q_i)_{i \in \mathrm{dom}(a)}) \quad \text{and} \\ q'_j = q_j, \qquad \text{for } j \notin \mathrm{dom}(a)\,. \end{array}$$

The language accepted by $\mathcal{A}$ is $L(\mathcal{A}) = \{t \in \mathbb{M}(\Sigma, D) \mid \delta(q_0, t) \cap F \neq \emptyset\}$. An automaton is called *complete*, if all transition relations are totally defined.
By standard methods, asynchronous and asynchronous cellular automata are shown to be equivalent. Starting e.g. with an asynchronous cellular automaton with local states sets Q_a, $a \in \Sigma$, an equivalent asynchronous automaton is obtained by embedding $\mathcal{A}$ through $Q_i := \prod_{i \in \mathrm{dom}(a)} Q_a$. Note that the number of (reachable) global states does not change. Conversely, given an asynchronous automaton with local states Q_i, $1 \le i \le m$, a simple time-stamping for each set $A_i = \{a \in \Sigma \mid i \in \mathrm{dom}(a)\}$ can be used for determining the last occurrence of letters from A_i (note $A_i \times A_i \subseteq D$): the sum modulo $|A_i|$ of the time-stamps associated to $a \in A_i$ yields the last occurrence from A_i (see also [19]). Let the size of an asynchronous (cellular) automaton denote the total number of local states. Then an asynchronous automaton of size n with m processors is transformed into an equivalent cellular one of size $O(|\Sigma| n)^m$; the converse transformation yields an asynchronous automaton with m processors and size $O(m \cdot n^{|\Sigma|})$.
Zielonka's theorem [21] characterizes the class of recognizable languages of finite traces by *deterministic* asynchronous automata, providing a suitable recognition device for traces. Asynchronous automata showed to be an appropriate finite state model for languages of real traces, too. A distributed (i.e., local) analogue of the usual Büchi (resp. Muller) acceptance condition has been proposed in [10]. The idea is to augment e.g. an asynchronous cellular automaton by a table $\mathcal{T} \subseteq \prod_{a \in \Sigma} \mathcal{P}(Q_a)$, i.e. we are given a tuple $\mathcal{A} = ((Q_a)_{a \in \Sigma}, (\delta_a)_{a \in \Sigma}, q_0, F, \mathcal{T})$.
Consider an infinite transition path $\pi = (q^0, a_0, q^1, a_1, \dots)$ in $\mathcal{A}$, with $q^n \in \prod_{a \in \Sigma} Q_a$, $a_n \in \Sigma$ and $q^{n+1} \in \delta(q^n, a_n)$ for $n \ge 0$. For each $a \in \Sigma$ we are interested in the set of local a-states which occur infinitely often in π, i.e. in the set $\inf_a(\pi) = \{q_a \in Q_a \mid (q^n)_a = q_a \text{ for infinitely many } n\}$.
The path π is accepted by $\mathcal{A}$ with the *Büchi* acceptance condition if for some $T = (T_a)_{a \in \Sigma} \in \mathcal{T}$ we have $\inf_a(\pi) \supseteq T_a$ for every $a \in \Sigma$. (Viewing $\mathcal{A}$ as a *Muller* automaton, π is accepted if for some $T = (T_a)_{a \in \Sigma} \in \mathcal{T}$, $\inf_a(\pi) = T_a$ for every $a \in \Sigma$). An infinite trace $t \in \mathbb{R}(\Sigma, D)$ is accepted if there exists a path π as above labelled by some representing word of t, i.e. $t = \varphi(a_0 a_1 \dots)$, which is accepted. (The local acceptance condition ensures that this notion of acceptance is well-defined, i.e. it does not depend on the representing word, [10]).
Finite traces are accepted in the usual way, by reaching a final state from F.
Similar definitions apply to the asynchronous model, where the acceptance depends on the sets $\inf_i(\pi) = \{q_i \in Q_i \mid (q^n)_i = q_i \text{ for infinitely many } n\}$.

We denote by $R_{\mathcal{A}}(t)$ the set of runs of an asynchronous cellular automaton $\mathcal{A}$ on $t \in \mathbb{R}(\Sigma, D)$, starting with the initial state q_0 of $\mathcal{A}$. We view a run $r \in R_{\mathcal{A}}(t)$ as a labelling $r\colon V(t) \to \cup_{a\in\Sigma} Q_a$ of the dependence graph of t, $[V(t), E(t), \lambda(t)]$, by local states, such that r is consistent with the alphabetical labelling $\lambda(t)$ and with the transition relations $(\delta_a)_{a\in\Sigma}$. Concretely, consider $(a, n_a) \in V(t)$ and let for $b \in D(a)$: $q_b = r(b, n_b)$, where $((b, n_b), (a, n_a)) \in E(t)$ with n_b maximal, respectively $q_b = (q_0)_b$, if no such (b, n_b) exists. Then we have $r(a, n_a) \in \delta_a((q_b)_{b\in D(a)})$. For $u \in \mathbb{M}(\Sigma, D)$, $r \in R_{\mathcal{A}}(u)$, let $\delta(r, u)$ denote the global state reached in the run r on u.

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 alphabet of 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 subgraph of (Σ, D). A trace t is connected if $\mathrm{alph}(t)$ is connected. The prefix order on $\mathbb{R}(\Sigma, D)$ is defined by $u \leq t$ if $t = uv$ for some $v \in \mathbb{R}(\Sigma, D)$. For $t = uv$ let $u^{-1}t = v$. As usual, $u \sqcap v$ is the greatest lower bound of u, v. Whenever it exists, the least upper bound of u, v is denoted by $u \sqcup v$. For $m \in \mathbb{N}$ let $[m]$ be the set $\{1, \ldots, m\}$. The complement of a set X is denoted $\overline{X}$, while $\mathcal{P}(X)$ denotes the powerset of X.

3 Determinization for asynchronous automata

We present in this section subset constructions for asynchronous (asynchronous cellular, resp.) automata relying on the notion of asynchronous mapping introduced by Cori/Métivier [2]. While the underlying idea of the first construction is more intuitive, the second one achieves a better bound for the size of the subset automata obtained. Asynchronous mappings reflect the functional aspect of asynchronous automata. As defined below, a mapping $\mu\colon \mathbb{M}(\Sigma, D) \to S$ is asynchronous if it can be computed stepwise in a distributed way, thus being easily transformed into an equivalent deterministic asynchronous (cellular) automaton. (For more details and general notions on traces and asynchronous automata see Ch. 7,8 in [7].) Before recalling the definition, let us introduce a basic notation for trace prefixes. For $t \in \mathbb{M}(\Sigma, D)$, $a \in \Sigma$ and $A \subseteq \Sigma$ let

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

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 3.1 ([2]) *A mapping $\mu \colon \mathbb{M}(\Sigma, D) \to S$ 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 $\mu(\partial_a(ta))$.*
- *The values $\mu(\partial_A(t))$, $\mu(\partial_B(t))$ and A, B uniquely determine $\mu(\partial_{A\cup B}(t))$.*

For example, $\mathrm{alph}(\cdot)$ is an asynchronous mapping. On the other hand, $\max(\cdot)$ is not asynchronous (let e.g. $(\Sigma, D) = a — b — c$ and consider $t_1 = ac$, $t_2 = acb$, resp. $t'_1 = bac$, $t'_2 = b$: $\max(t_1 \sqcup t_2) \neq \max(t'_1 \sqcup t'_2)$).
Suppose we are given an asynchronous mapping $\mu \colon \mathbb{M}(\Sigma, D) \to S$ and a subset $R \subseteq S$. Then let $\mathcal{A}_\mu = ((Q_a)_{a\in\Sigma}, \delta, q_0, F)$ be defined by $Q_a = \{\mu(\partial_a(t)) \mid t \in \mathbb{M}(\Sigma, D)\}$, $a \in \Sigma$, $q_0 = (\mu(1))_{a\in\Sigma}$, $F = \{(\mu(\partial_a(t)))_{a\in\Sigma} \mid \mu(t) \in R\}$, and for $q, q' \in \prod_{a\in\Sigma} Q_a$:

$$q' = \delta(q, a) \iff q_b = \mu(\partial_b(t)),\ q'_b = \mu(\partial_b(ta)),\ b \in \Sigma, \quad \text{for some } t \in \mathbb{M}(\Sigma, D).$$

Since $\partial_a(ta) = \partial_{D(a)}(t)a$ and $\partial_b(ta) = \partial_b(t)$ for $b \neq a$, it is immediate that δ is the transition function of an asynchronous cellular automaton. Moreover, with $\delta(q_0, t) = (\mu(\partial_a(t)))_{a\in\Sigma}$, it is clear that $L(\mathcal{A}_\mu) = \mu^{-1}(R)$ [3, 4].

Remark 3.2 1. Asynchronous mappings can be easily translated to asynchronous automata, too. Let $\mathrm{dom}(a) \subseteq [m]$ denote the domain of $a \in \Sigma$, and let $A_i = \{a \in \Sigma \mid i \in \mathrm{dom}(a)\}$, $i \in [m]$. Consider the automaton $\mathcal{A} = ((Q_i)_{i\in[m]}, (\delta_a)_{a\in\Sigma}, q_0, F)$ with $Q_i = \{\mu(\partial_{A_i}(t)) \mid t \in \mathbb{M}(\Sigma, D)\}$, $q_0 = (\mu(1))_{i\in[m]}$, $F = \{(\mu(\partial_{A_i}(t)))_{i\in[m]} \mid \mu(t) \in R\}$. The transition $q' = \delta(q, a)$ is defined if $q_i = \mu(\partial_{A_i}(t))$ and $q'_i = \mu(\partial_{A_i}(ta))$, $i \in [m]$, holds for some $t \in \mathbb{M}(\Sigma, D)$. It is left to the reader to verify that δ is the transition function of an asynchronous automaton accepting $\mu^{-1}(R)$.

2. Consider a non-deterministic asynchronous (asynchronous cellular, resp.) automaton $\mathcal{A}$. It will suffice to obtain an asynchronous mapping $\mu \colon \mathbb{M}(\Sigma, D) \to S$ based on $\mathcal{A}$, with S finite and such that $\mu^{-1}\mu(L(\mathcal{A})) = L(\mathcal{A})$. The asynchronous (cellular) subset automaton will be then directly constructed as just described.

As previously mentioned, we will use throughout our determinization constructions Zielonka's labelling function $\nu \colon \mathbb{M}(\Sigma, D) \to \{0, \ldots, |\Sigma|\}^{\Sigma\times\Sigma}$, which is defined inductively for $a, b \in \Sigma$, $t \in \mathbb{M}(\Sigma, D)$ by:

- $\nu(1)(a, b) = 0$.
- 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

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

Note that $\nu(\partial_A(t))(c,a) = \nu(t)(c,a)$ for every $a \in A$.
The labelling function ν is a time-stamping function, allowing to determine the actuality of information received in a distributed way. In particular, it provides information about ordering of prefixes of the form $\partial_a(t)$. Consider e.g. the dependence alphabet $(\Sigma = \{a,b,c,d\}, D)$ with $D = (\Sigma \times \Sigma) \setminus \{(a,c),(c,a),(b,d),(d,b)\}$, and the trace $t = [ada^n cb]$, $n \geq 1$. Then using ν we are able to determine whether b or c have the most recent information about a. In the following we point out some basic properties of ν:

Fact 3.3 ([3, 4]) *Let $t \in \mathbb{M}(\Sigma, D)$, $a,b,c \in \Sigma$, $A, B \subseteq \Sigma$. Then:*

1. $\nu(t)(c,a) = \nu(t)(c,b) \iff \partial_{c,a}(t) = \partial_{c,b}(t)\,.$

2. *Suppose that we are given the value $\nu(t)$ or both values $\nu(\partial_A(t))$, $\nu(\partial_B(t))$. Then we can determine the set $C_{a,b} := \{c \in \Sigma \mid \partial_{c,a}(t) = \partial_{c,b}(t)\}$, for every $a,b \in A \cup B$. Moreover, we can determine for every $c \in \Sigma$ which of the following holds:*

$$\partial_{c,A}(t) = \partial_{c,B}(t)\,, \quad \textit{resp. } \partial_{c,A}(t) < \partial_{c,B}(t)\,, \quad \textit{resp. } \partial_{c,B}(t) < \partial_{c,A}(t)\,.$$

3. *ν is an asynchronous mapping.*

3.1 A simple determinization construction

The crucial idea for the determinization constructions presented in the sequel is to augment the time-stamping mapping ν by a mapping ρ depending on the given non-deterministic asynchronous (asynchronous cellular, resp.) automaton $\mathcal{A}$.
We start considering the cellular model and denote in the following by Q the set of local states, $Q = \dot{\bigcup}_{a \in \Sigma} Q_a$ of an asynchronous cellular automaton $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0, F)$.
Let now $\rho \colon \mathbb{M}(\Sigma, D) \to \mathcal{P}(Q^{\Sigma\times\Sigma})$ be given for $t \in \mathbb{M}(\Sigma, D)$ as

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

Thus, every element $f \colon \Sigma \times \Sigma \to Q$ of $\rho(t)$ is associated to a run r on t, such that for every $a, b \in \Sigma$, $f(b,a)$ represents the local b-state reached in the run r on the prefix $\partial_a(t)$ of t.

Proposition 3.4 *Given a non-deterministic asynchronous cellular automaton $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0, F)$. Then the associated mapping (ν, ρ) is asynchronous.*
Moreover, we have

$$L(\mathcal{A}) = \{t \in \mathbb{M}(\Sigma, D) \mid \exists q = (q_a)_{a\in\Sigma} \in F,\ \exists f \in \rho(t) \textit{ s.t. } f(a,a) = q_a,\ \forall a \in \Sigma\}.$$

For the proof, let us start with a technical lemma.

Lemma 3.5 *Let $A, B \subseteq \Sigma$, $t = \partial_{A\cup B}(t)$ and $t_1 = \partial_A(t)$, $t_2 = \partial_B(t)$. Let $a, b \in \Sigma$ such that $\partial_a(t_1) < \partial_a(t_2)$ and $\partial_{b,a}(t) \leq t_1 \sqcap t_2$.*
Then there exists some $d \in \Sigma$ such that $\partial_{b,a}(t) = \partial_{b,d}(t_1)$. Moreover, a letter d with this property can be computed effectively, given the values $\nu(t_1)$ and $\nu(t_2)$.

Proof: By assumption, we have $\partial_{b,a}(t) = \partial_{b,a}(t_2) \leq t_1 \sqcap t_2$. Using ν the common prefix $s := t_1 \sqcap t_2$ can be determined, since we have $\max(s) \subseteq C$, with $C = \{c \in \Sigma \mid \partial_c(t_1) = \partial_c(t_2)\} = \{c \in \Sigma \mid \nu(t_1)(c, c) = \nu(t_2)(c, c)\}$ [3]. By the very definition of C we have $\partial_c(s) = \partial_c(t_i)$, for every $c \in C$, $i = 1, 2$. Let further $u, v \in \mathbb{M}(\Sigma, D)$ be such that $t_1 = su$, $t_2 = sv$, hence $\mathrm{alph}(u) \times \mathrm{alph}(v) \subseteq I$.
Assume first $\partial_{b,a}(t) \neq 1$ and consider the set $C' = \{d \in \Sigma \mid \partial_{d,a}(t_2) = \partial_d(s)\}$. We denote $s' = s \sqcap \partial_a(t_2)$, with $s = s'x$, $\partial_a(t_2) = s'y$ for suitable $x, y \in \mathbb{M}(\Sigma, D)$, where $\mathrm{alph}(x) \times \mathrm{alph}(y) \subseteq I$ (see also Fig. 1). Again, we have $s' = \partial_{C'}(s') = \partial_{C'}(s) = \partial_{C',a}(t_2)$.
From $\partial_{b,a}(t_2) \leq \partial_b(s)$ we obtain directly $\partial_{b,a}(t_2) = \partial_b(s')$. (In particular, $s' \neq 1$, thus $C' \neq \emptyset$.) Together with $s' = \partial_{C'}(s)$ we have $\partial_{b,a}(t_2) = \partial_{b,d}(s)$ for a suitable $d \in \max(s') \subseteq C'$.
We show now $\partial_d(s) = \partial_d(t_1)$. Since d belongs to a path from b to a, where the vertex labelled a lies in y, there is some $e \in \mathrm{alph}(y)$ with $(d, e) \in D$ (see Fig. 1). Due to $y \leq v$, the assumption $d \in \mathrm{alph}(u)$ would contradict $\mathrm{alph}(u) \times \mathrm{alph}(v) \subseteq I$. (Note that $y \neq 1$, otherwise we would have $\partial_a(t_2) \leq s$.) By the definition of C' we have $C' \cap \mathrm{alph}(x) = \emptyset$. From $t_1 = s'xu$ we conclude $\partial_d(t_1) = \partial_d(s') = \partial_d(s)$. Note also that $\partial_{b,a}(t_2) = \partial_{b,C''}(s)$, where $C'' := C' \cap D(\mathrm{alph}(v))$.
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 $\mathrm{alph}(v)$ [3]; using again $\nu(t_2)$ we compute C'. Finally we choose $d \in C''$ such that $\partial_{b,d}(t_1) = \partial_{b,C''}(t_1)$ and obtain

$$\partial_{b,d}(t_1) = \partial_{b,C''}(t_1) \overset{C''\subseteq D(\mathrm{alph}(v))}{=} \partial_{b,C''}(s) = \partial_{b,a}(t_2)\,.$$

Finally, if $\partial_{b,a}(t_2) = 1$, then $d = a$ satisfies the requirement $\partial_{b,a}(t_2) = \partial_{b,d}(t_1)$.
□

Remark 3.6 Consider now a clique covering of $(\Sigma, D) = (\bigcup_{i\in[m]} A_i, \bigcup_{i\in[m]} A_i \times A_i)$. A closer look at the proof of Lem. 3.5 yields the existence of a clique A_k such that $\partial_{b,a}(t) = \partial_{b,A_k}(t_1)$. For this, let us assume $\partial_{b,a}(t) \neq 1$ and consider a clique A_k containing both d, e, with the notations from the last proof. We observe that $\partial_{A_k}(t_1) = \partial_{A_k}(s') = \partial_d(s')$, since $A_k \subseteq D(e) \subseteq D(\mathrm{alph}(y))$ and $d \in A_k \cap \max(s')$. For $\partial_{b,a}(t) = 1$ we choose A_k with $a \in A_k$ and note $a \notin \mathrm{alph}(u)$. Hence, it can be easily verified that for any $i, j \in [m]$ with $\partial_{A_i}(t_1) < \partial_{A_i}(t_2)$ and $\partial_{A_j,A_i}(t) \leq t_1 \sqcap t_2$ some $k \in [m]$ effectively exists such that $\partial_{A_j,A_i}(t) = \partial_{A_j,A_k}(t_1)$.

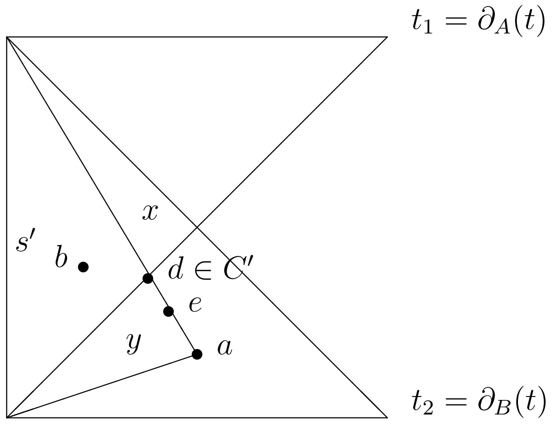

Figure 1: $\partial_{b,d}(t_1) = \partial_{b,a}(t_2) = \partial_{b,a}(t)$.

Proof of 3.4: By Fact 3.3 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 $R \subseteq Q^{\Sigma\times\Sigma}$ by letting $g \in R$ if for some $f \in \rho(\partial_{D(a)}(t))$:

- $g(c, b) = f(c, b)$, for $b \neq a$ and $c \in \Sigma$;
- $g(c, a) = q_c \in Q_c$, where $q_c = f(c, c)$ for $c \neq a$, and $q_a \in \delta_a((f(b, b))_{b\in D(a)})$.

It is straightforward to see $R = \rho(\partial_a(ta))$, since runs on $\partial_a(ta)$ represent exactly extensions of runs on $\partial_{D(a)}(t)$ by an a-transition. The details are left to the reader.
Consider now $A, B \subseteq \Sigma$ and $t = \partial_{A\cup B}(t)$, with $t_1 = \partial_A(t)$, $t_2 = \partial_B(t)$, $s = t_1 \sqcap t_2$. Hence, $t_1 = su$ and $t_2 = sv$ with $\mathrm{alph}(u) \times \mathrm{alph}(v) \subseteq I$. We denote $C = \{c \in \Sigma \mid \partial_c(t_1) = \partial_c(t_2)\} = \{c \in \Sigma \mid \nu(t_1)(c, c) = \nu(t_2)(c, c)\}$. Let $\nu(t_i), \rho(t_i)$ be given and define $R \subseteq Q^{\Sigma\times\Sigma}$ by $f \in R$ if for some $f_i \in \rho(t_i)$ $(i = 1, 2)$ satisfying $f_1(c', c) = f_2(c', c)$ for every $c \in C$, $c' \in \Sigma$, we have for all $a, b \in \Sigma$:

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

By Lemma 3.5 f is well-defined in the last case, too. The definition of f is based on the idea of combining pairs of runs on t_1, t_2 if they yield the same global state on the common prefix $s = t_1 \sqcap t_2$.
More precisely, let $r_i \in R_A(t_i)$ be given such that $\delta(r_1, \partial_c(t_1)) = \delta(r_2, \partial_c(t_2))$, for every $c \in C$. In particular, we also have $\delta(r_1, s) = \delta(r_2, s)$, since $s = \partial_C(s)$. Denoting by $[V_i, E_i, \lambda_i]$ the dependence graph of t_i it follows easily that the mapping $r \colon V_1 \cup V_2 \to Q$ with

$$r(v) = \begin{cases} r_1(v) & \text{if } v \in V_1 \\ r_2(v) & \text{if } v \in V_2 \setminus V_1 \end{cases}$$

is a well-defined run on t. Hence, for the inclusion $R \subseteq \rho(t)$ it suffices to show that the mappings $f \in Q^{\Sigma\times\Sigma}$ defined above correspond exactly to runs $r\colon V_1 \cup V_2 \to Q$ as just described. The converse inclusion $\rho(t) \subseteq R$ is immediate.
For $a, b \in \Sigma$ we distinguish whether $\partial_{b,a}(t) = \partial_{b,a}(t_1)$ or $\partial_{b,a}(t) = \partial_{b,a}(t_2)$ holds, where in the second case the position of the maximal vertex of $\partial_{b,a}(t)$ (if any) w.r.t. v or s is taken into account:

$$\partial_{b,a}(t) = \begin{cases} \partial_{b,a}(t_1) & \text{if } \partial_a(t_2) \le \partial_a(t_1) \\ \partial_{b,a}(t_2) & \text{if } \partial_b(s) < \partial_{b,a}(t_2) \\ \partial_{b,d}(t_1) & \text{otherwise, for } d \in \Sigma \text{ s.t. } \partial_{b,a}(t_2) = \partial_{b,d}(t_1)\,. \end{cases}$$

To conclude the proof, note that the values $\nu(t_1), \nu(t_2)$ allow to distinguish the three cases considered in the definition of f, due to $s = \partial_C(t_2)$, together with Lem. 3.5. □

Notation: We will use in the next section the abbreviation $q \overset{(a)}{\to} q'$, for $t \in \mathbb{M}(\Sigma, D)$ and states $q, q' \in \prod_{a\in\Sigma} Q_a$ in an asynchronous cellular automaton $\mathcal{A}$. This means that $t \in \mathbb{M}(\Sigma, D)$ and a run $r \in R_{\mathcal{A}}(\partial_a(ta))$ exist, such that $q = \delta(r, \partial_a(t))$ and $q' = \delta(r, \partial_a(ta))$. Note that the subset automaton based on Prop. 3.4 (see also Prop. 3.9) provides this information, given the values $\nu(\partial_b(t))$, $\rho(\partial_b(t))$, for all $b \in D(a)$.

Remark 3.7 The idea of the construction in Prop. 3.4 can be easily adapted to asynchronous automata. Let $\mathcal{A} = ((Q_i)_{i=1}^m, (\delta_a)_{a\in\Sigma}, q_0, F)$ be a non-deterministic asynchronous automaton, and let $Q = \bigcup_{i\in[m]} Q_i$. Let $\rho\colon \mathbb{M}(\Sigma, D) \to \mathcal{P}(Q^{[m]\times[m]})$ be defined for $t \in \mathbb{M}(\Sigma, D)$ by:

$$\rho(t) := \{f \in Q^{[m]\times[m]} \mid \exists r \in R_{\mathcal{A}}(t) :\ \delta(r, \partial_{A_i}(t))_j = f(j, i)\}.$$

Then the mapping $\mu = (\nu, \rho)$ is asynchronous. Consider e.g. the case where $\mu(t_1)$, $\mu(t_2)$ are given, with $t_1 = \partial_A(t) = su$, $t_2 = \partial_B(t) = sv$, $\mathrm{alph}(u) \times \mathrm{alph}(v) \subseteq I$. Assume that both u and v are non-empty and let $C = \{c \in \Sigma \mid \partial_c(t_1) = \partial_c(t_2)\}$. We show how runs $r_i \in R_{\mathcal{A}}(t_i)$, $i = 1, 2$, can be chosen with $\delta(r_1, s) = \delta(r_2, s)$. For $k \in [m]$ let $c \in \max(s)$ be such that $\partial_{A_k}(s) = \partial_{A_k,c}(s)$. Due to $u \ne 1$, $v \ne 1$ we have $c \in D(\mathrm{alph}(u)) \cap D(\mathrm{alph}(v))$, hence there exist $i, j \in [m]$ with $A_i \cap \mathrm{alph}(u) \ne \emptyset$, $A_j \cap \mathrm{alph}(v) \ne \emptyset$ and $c \in A_i \cap A_j$. Clearly, $A_i \cap \mathrm{alph}(v) = A_j \cap \mathrm{alph}(u) = \emptyset$. This yields $\partial_c(s) = \partial_{A_i}(t_2) = \partial_{A_j}(t_1)$, hence $\partial_{A_k}(s) = \partial_{A_k,A_i}(t_2) = \partial_{A_k,A_j}(t_1)$. Moreover, C, c and i, j can be effectively determined from $\nu(t_1)$, $\nu(t_2)$. For the definition of the combined run on $t = t_1 \sqcup t_2$ we require $f_1(k, i) = f_2(k, j)$ for all k and we use Rem. 3.6. The details and the definition of a subset $R \subseteq \{0, \dots, |\Sigma|\}^{\Sigma\times\Sigma} \times \mathcal{P}(Q^{[m]\times[m]})$ with $L(\mathcal{A}) = \mu^{-1}(R)$ are left to the reader.

The asynchronous mappings considered in Prop. 3.4 and Rem. 3.7 lead together with Rem. 3.2(2) to determinization constructions for both asynchronous automata models as stated in the following theorem. Recall that the size of an asynchronous (cellular) automaton corresponds to the total number of local states.

Theorem 3.8 *1. Let $\mathcal{A}$ be a non-deterministic asynchronous cellular automaton of size n. Then a deterministic asynchronous cellular subset automaton $\tilde{\mathcal{A}}$ of size $2^{O(n^{|\Sigma|^2})}$ effectively exists with $L(\mathcal{A}) = L(\tilde{\mathcal{A}})$.*

2. Let $\mathcal{A}$ be a non-deterministic asynchronous automaton of size n with m processors. Then a deterministic asynchronous subset automaton $\tilde{\mathcal{A}}$ of size $2^{O(n^{m^2})}$ effectively exists with $L(\mathcal{A}) = L(\tilde{\mathcal{A}})$, having the same read-and-write-domains as $\mathcal{A}$.

3.2 An improved determinization construction

We consider in this section further mappings for both asynchronous models, which lead to determinization constructions with improved bounds w.r.t. size. More precisely, we obtain a size blow-up of $2^{n^{O(|\Sigma|)}}$ for the asynchronous cellular model, resp. $2^{n^{O(m)}}$ for the asynchronous one, where m is the number of processors. However, the correctness proof is more involved and the definition of these mappings is less intuitive.
We denote in the following by Q_Σ the set of global states of an asynchronous cellular automaton $\mathcal{A}$, $Q_\Sigma = \prod_{a\in\Sigma} Q_a$.

Proposition 3.9 *Given a complete non-deterministic asynchronous cellular automaton $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0, F)$.*
Let $\rho\colon \mathbb{M}(\Sigma, D) \to \mathcal{P}(Q_\Sigma \times Q_\Sigma \times \mathcal{P}(\Sigma) \times \Sigma)$ be defined for $t \in \mathbb{M}(\Sigma, D)$ by:

$$\rho(t) := \{(q, q', A, a) \mid \exists r \in R_{\mathcal{A}}(t) :\ \delta(r, \partial_A(t)) = q \text{ and } \delta(r, \partial_{A\cup\{a\}}(t)) = q'\}.$$

Then the mapping $\mu = (\nu, \rho)$ is asynchronous.
Moreover, we have

$$L(\mathcal{A}) = \{t \in \mathbb{M}(\Sigma, D) \mid \exists q \in F :\ (q, q, \Sigma, a) \in \rho(t), \forall a \in \Sigma\}.$$

Proof: Let first $t = \partial_{D(a)}(t)$ and $t' = ta$, for $a \in \Sigma$. Note that for $X \subseteq \Sigma$, $\partial_X(t') = \partial_X(t)$ if $a \notin X$, resp. $\partial_X(t') = t'$ if $a \in X$. For $B \subseteq \Sigma$, $b \in \Sigma$, and given $\mu(t)$ and a we define a set R as the least subset of $Q_\Sigma \times Q_\Sigma \times \mathcal{P}(\Sigma) \times \Sigma$ satisfying the following:

1. If $a \neq b$ and $a \notin B$, then let $(q, q', B, b) \in R$ if $(q, q', B, b) \in \rho(t)$.
2. If $b = a \notin B$, then let $(q, q', B, b) \in R$ if some $q'' \in Q_\Sigma$ exists such that $(q'', q'', \Sigma, a) \in \rho(t)$, $q' \in \delta(q'', a)$ and $q'' \in \delta(q, \partial_B(t)^{-1}t)$. Note that the last condition can be checked by augmenting B stepwise by letters from $D(a)$.
3. If $a \in B$, then let $(q, q, B, a) \in R$ if $q \in \delta(q', a)$, for some $(q', q', \Sigma, a) \in \rho(t)$.

By the previous remark, it is easy to see that $R \subseteq \rho(t')$ holds. The converse inclusion $\rho(t') \subseteq R$ is immediate.
For the second property of asynchronous mappings let $A, B \subseteq \Sigma$ and consider $t = \partial_{A\cup B}(t)$ with $t_1 = \partial_A(t) = su$, $t_2 = \partial_B(t) = sv$, where $\mathrm{alph}(u) \times \mathrm{alph}(v) \subseteq I$. We denote by C the alphabet $C = \{c \in \Sigma \mid \partial_c(t_1) = \partial_c(t_2)\}$, and recall that C can be computed from the values $\nu(t_1), \nu(t_2)$ of the labelling function ν. Recall further that $s = \partial_C(s) = \partial_C(t_i)$, $i = 1, 2$.
Let $E \subseteq \Sigma$, $a \in \Sigma$ and assume without loss of generality $\partial_a(t_2) \leq \partial_a(t_1)$. Let $y = \partial_E(v)$ and note that we have $\partial_E(t) = \partial_E(t_1 v) = \partial_{E\cup F}(t_1)y$, where $F = D(\mathrm{alph}(y))$.
The idea of this construction is to determine runs on $\partial_E(t)$ by applying to more information than necessary in order to build the run (see also Fig. 2): we combine a run $r_1 \in R_{\mathcal{A}}(\partial_{E\cup F}(t_1))$ with a run $r_2 \in R_{\mathcal{A}}(\partial_{C\cup E}(t_2))$ to a run r on $\partial_E(t)$ corresponding to r_1 on $\partial_{E\cup F}(t_1)$, respectively to r_2 on y. For the consistency note first that $\partial_f(t_1) = \partial_f(s)$ for every $f \in F$, since $F \subseteq D(\mathrm{alph}(v))$ and $\mathrm{alph}(u) \times \mathrm{alph}(v) \subseteq I$. Moreover, in order to combine consistent runs we need the state reached by a run on $\partial_E(t_2) = \partial_{E\cup F}(s)y$ on the prefix $\partial_{E\cup F}(s)$ (actually, only the F-components are needed). This information cannot be provided by supplying the global state on $\partial_E(t_2)$, only. Therefore, we consider the run r_2 on the larger prefix $\partial_{C\cup E}(t_2) = sy$, and we additionally assume the existence of a third run r_3 on s, such that r_2 is an extension of r_3 and r_1 agrees with r_3 in the F-components.

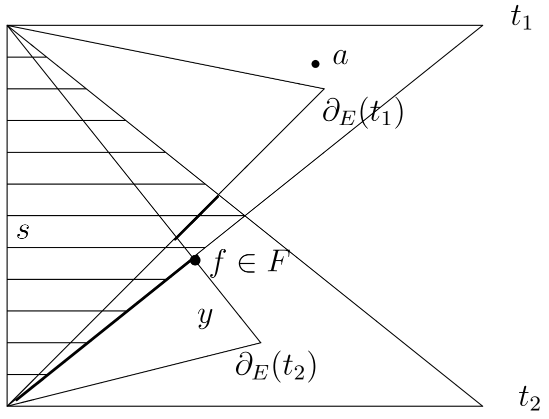

Figure 2: Forming runs on $\partial_E(t)$.

Formally, consider a run $r_3 \in R_{\mathcal{A}}(s)$ and let $q_1 = \delta(r_1, \partial_{E\cup F}(t_1))$, $q_2 = \delta(r_2, \partial_{C\cup E}(t_2))$, and $q_3 = \delta(r_3, s)$. Suppose that $q_2 \in \delta(q_3, y)$ and $(q_1)_f = (q_3)_f$, for every $f \in F$. We claim that the following mapping $r \colon V(\partial_E(t)) \to \bigcup_{a\in\Sigma} Q_a$, which labels the vertex set $V(\partial_E(t))$ of $\partial_E(t)$ by local states, is a well-defined run on $\partial_E(t)$:

$$r(w) := \begin{cases} r_1(w) & \text{if } w \in V(\partial_{E\cup F}(t_1)) \\ r_2(w) & \text{if } w \in V(y), \end{cases}$$

with $V(\partial_{E\cup F}(t_1))$ resp. $V(y)$ denoting the vertex set of $\partial_{E\cup F}(t_1)$ resp. y.
To see the claim, note that given a run r' on $\partial_{C\cup E}(t_2) = sy$ and a run r'' on s with $\delta(r', s)_f = \delta(r'', s)_f$ for all $f \in F$, we can combine r', r'' to a third run r''' on $\partial_{C\cup E}(t_2)$, corresponding to r'' on s, resp. to r' on y. This property is mainly due to $F = D(\text{alph}(y))$, together with the definition of read-domains of an asynchronous cellular automaton. Recall also $\partial_{f,E\cup F}(t_1) = \partial_f(t_1) = \partial_f(s)$, for $f \in F$.
The global state $q = \delta(r, \partial_E(t))$ associated to the run r defined above satisfies:

$$q_e = \begin{cases} (q_2)_e & \text{if } \partial_{e,C}(t_2) < \partial_{e,E}(t_2) \\ (q_1)_e & \text{otherwise.} \end{cases}$$

Recall the assumption $\partial_a(t_2) \leq \partial_a(t_1)$ and let r_4 denote a run on $\partial_{E\cup F\cup\{a\}}(t_1)$ satisfying $\delta(r_4, \partial_{E\cup F}(t_1)) = q_1 = \delta(r_1, \partial_{E\cup F}(t_1))$. Denote by q_4 the state $q_4 = \delta(r_4, \partial_{E\cup F\cup\{a\}}(t_1))$. Similar to the run r, the mapping $r' \colon V(\partial_{E\cup\{a\}}(t)) \to \bigcup_{a\in\Sigma} Q_a$ defined by

$$r'(w) := \begin{cases} r_4(w) & \text{if } w \in V(\partial_{E\cup F\cup\{a\}}(t_1)) \\ r_2(w) & \text{if } w \in V(y) \end{cases}$$

is a well-defined run on $\partial_{E\cup\{a\}}(t)$. Note that the run r' yields the state q on the prefix $\partial_E(t)$. For $q' = \delta(r', \partial_{E\cup F\cup\{a\}}(t))$ we have analogously to q:

$$q'_e = \begin{cases} (q_2)_e & \text{if } \partial_{e,C}(t_2) < \partial_{e,E}(t_2) \\ (q_4)_e & \text{otherwise.} \end{cases}$$

Therefore, for $\partial_a(t_2) \leq \partial_a(t_1)$ we let $(q, q', E, a) \in R$ if the following conditions are satisfied for some states $q_i \in \prod_{a\in\Sigma} Q_a$, $1 \leq i \leq 4$:

1. $(q_1)_f = (q_3)_f$, for every $f \in F = D(\text{alph}(\partial_E(v)))$.
2. Let $E = \{e_1, \dots, e_k\}$, $k \geq 0$, and $C_i = C \cup \{e_1, \dots, e_i\}$, $0 \leq i \leq k$ (with $C = C_0$). Then we require the existence of a sequence $q_3 = p_0, p_1, \dots, p_k = q_2$ of global states satisfying $(p_i, p_{i+1}, C_i, e_{i+1}) \in \rho(t_2)$, for all $0 \leq i < k$.
3. $(q_1, q_4, E \cup F, a) \in \rho(t_1)$.
4. - $q_e = q'_e = (q_2)_e$, if $\partial_{e,C}(t_2) < \partial_{e,E}(t_2)$.
   - $q_e = (q_1)_e$ resp. $q'_e = (q_4)_e$, if $\partial_{e,E}(t_2) \leq \partial_{e,C}(t_2)$.

By the remarks above we see $R \subseteq \rho(t)$, whereas the converse inclusion is again immediate, using the fact that the transition relations of $\mathcal{A}$ are totally defined. Note also that the sets C, F and the conditions concerning the prefixes $\partial_a(t_2)$, $\partial_a(t_1)$, $\partial_{e,E}(t_2)$, $\partial_{e,C}(t_2)$ can be checked using the values $\nu(t_1)$, $\nu(t_2)$. □

Remark 3.10 The idea of the improved construction for asynchronous cellular automata based on Prop. 3.9 can be adapted also to asynchronous automata. Let $\mathcal{A} = ((Q_i)_{i=1}^m, (\delta_a)_{a\in\Sigma}, q_0, F)$ denote an asynchronous automaton with read-and-write domains $\mathrm{dom}(a) \subseteq [m]$, $a \in \Sigma$, and denote by $Q_{[m]}$ the set of global states $\prod_{i\in[m]} Q_i$. We denote in the following by A_i, $i \in [m]$, the dependence clique $\{a \in \Sigma \mid i \in \mathrm{dom}(a)\}$, and let $\partial_i(t)$, resp. $\partial_E(t)$ ($i \in [m]$, $E \subseteq [m]$) denote the prefix $\partial_{A_i}(t)$, resp. $\partial_{A_E}(t)$, where $A_E = \cup_{i\in E} A_i$. Consider the mapping $\rho \colon \mathbb{M}(\Sigma, D) \to \mathcal{P}(Q_{[m]} \times Q_{[m]} \times \mathcal{P}([m]) \times [m])$ given by

$$\rho(t) := \{(q, q', E, i) \mid \exists r \in R_{\mathcal{A}}(t) :\ \delta(r, \partial_E(t)) = q \text{ and } \delta(r, \partial_{E\cup\{i\}}(t)) = q'\}.$$

Since the first property of asynchronous mappings can be easily verified for the above mapping, let us consider the second one. Suppose $t = \partial_{A\cup B}(t)$, with $A, B \subseteq \Sigma$, and let $t_1 = \partial_A(t) = su$, $t_2 = \partial_B(t) = sv$, where $\mathrm{alph}(u) \times \mathrm{alph}(v) \subseteq I$. For $E \subseteq [m]$, $i \in [m]$ assume $u \neq 1$ and $\partial_i(t_2) \leq \partial_i(t_1)$. We denote as in the proof of Prop. 3.9 $y = \partial_E(v)$, and let $F = \bigcup_{b\in\mathrm{alph}(y)} \mathrm{dom}(b)$. Note that $D(\mathrm{alph}(y)) = \bigcup_{i\in F} A_i$, hence $\partial_E(t) = \partial_E(t_1 v) = \partial_{E\cup F}(t_1) y$. Moreover, we have again $\partial_i(t_1) = \partial_i(s)$, for every $i \in F$.
It remains to define the set C accordingly. Let $C' = \{c \in \Sigma \mid \partial_c(t_1) = \partial_c(t_2)\} \supseteq \max(s)$ and define $C = \{j \in [m] \mid A_j \cap C' \neq \emptyset, A_j \cap \mathrm{alph}(u) \neq \emptyset\}$. Since $u \neq 1$, we have for every $c \in \max(s)$: $c \in D(\mathrm{alph}(u))$, hence $\max(s) \subseteq \bigcup_{i\in C} A_i$.
Moreover, $\partial_C(t_2) = \partial_C(s) = s$, and for every $i \in F$: $\partial_{i,C}(t_2) = \partial_i(s) = \partial_i(t_1) = \partial_{i,E\cup F}(t_1)$.
It is not difficult to show that $\mu = (\nu, \rho)$ is asynchronous and that $L(\mathcal{A}) = \mu^{-1}\mu(L(\mathcal{A}))$. The details of the proof are left to the reader.

Prop. 3.9 and the above remark yield the improved constructions for asynchronous (asynchronous cellular, resp.) subset automata:

Theorem 3.11 *1. Let $\mathcal{A}$ be a non-deterministic asynchronous cellular automaton of size n. Then a deterministic asynchronous cellular subset automaton $\tilde{\mathcal{A}}$ of size $2^{n^{O(|\Sigma|)}}$ effectively exists with $L(\mathcal{A}) = L(\tilde{\mathcal{A}})$.*

2. Let $\mathcal{A}$ be a non-deterministic asynchronous automaton of size n with m processors. Then a deterministic asynchronous automaton $\tilde{\mathcal{A}}$ of size $2^{n^{O(m)}}$ effectively exists with $L(\mathcal{A}) = L(\tilde{\mathcal{A}})$, having the same read-and-write-domains as $\mathcal{A}$.

We conclude this section with remarks concerning related subset constructions. As previously mentioned, Klarlund, Mukund and Sohoni presented independently a solution to the determinization problem for asynchronous automata [12]. Their construction yields a blow-up in size of $2^{n^{O(m^3)}}$, where n denotes the size of the input automaton (as the maximal size of local states sets) and m is the number of processors. This paper also contains a nice example for the lower bound of $2^{n^m/m}$.

A less direct subset construction can be achieved by using an alternative determinization procedure given in [3]. Here, we view e.g. a non-deterministic asynchronous cellular automaton $\mathcal{A}$ as a sequential automaton with transition relation $\delta \subseteq Q \times \Sigma^* \times Q$, where $|Q| \leq n^{|\Sigma|}$ denotes the set of global states. After determinizing $\mathcal{A}$ (and possibly minimizing it) we obtain a deterministic automaton with at most $2^{n^{|\Sigma|}}$ states, having the I-diamond property (i.e., for every states q, q' and letters $(a,b) \in I$: $q' = q \cdot ab \Leftrightarrow q' = q \cdot ba$). This sequential automaton can be used as input for the alternative construction of deterministic asynchronous automata mentioned above (see [3]). One obtains an equivalent asynchronous cellular automaton with at most $2^{O(n)^{|\Sigma|}}$ local states. For practical use, this off-line approach may be less efficient than the direct constructions of Thm. 3.8 and [12], since simulating the subset automaton is done by table lookup instead of updating local information. Moreover, direct subset constructions are more flexible w.r.t. modifications of the input automaton or to partial determinization. Note also that direct constructions take only reachable local states into account.

4 Complementing Büchi asynchronous cellular automata

The complementation procedure for asynchronous cellular Büchi automata presented in this section applies the *progress measure* technique proposed by Klarlund [11], and uses the subset automaton construction given in Sect. 3. Progress measures have been devised in a more general setting for verifying sequential programs, and provide optimal complementation procedures for e.g. Büchi and Streett ω-automata.

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

Throughout this section we use the standard representation for dependence graphs, introduced in Sect. 2.

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

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

Hence, this local acceptance condition specifies halting states for letters $a \notin \mathrm{alphinf}(t)$ and recurrent states for the designated letters a_i. Note that an asynchronous cellular Büchi automaton with acceptance table $\mathcal{T} \subseteq \prod_{a\in\Sigma} \mathcal{P}(Q_a)$ as

defined in Sect. 2 can be easily transformed into an equivalent one with acceptance table as above: for t with $\mathrm{alphinf}(t) = A$ the letters $a_i \in A_i$ can be used for checking that all local states from $\bigcup_{a \in A_i} T_a$ occur infinitely often, where $T = (T_a)_{a\in\Sigma} \in \mathcal{T}$. The converse transformation is straightforward, since we only have to check additionally that the input trace belongs to some (recognizable) set $\mathrm{Inf}(A)$, $A \subseteq \Sigma$.
For $t \in \mathbb{R}(\Sigma, D)$, $a \in \Sigma$, $0 \le n < |t|_a$, let $t[a, n] = \sqcap\{u \le t \mid |u|_a = n + 1\}$ be the least prefix of t containing the first $n+1$ occurrences of a (note $\max(t[a, n]) = \{a\}$, if $|t|_a > 0$). Furthermore, let us define $U_a(t) \subseteq Q \times \mathbb{N}$ by

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

$U_a(t)$ is the vertex set of a directed graph containing the information about the global states reached by prefixes $t[a, n]$, $n \ge 0$. The edges are given by $(q, n) \overset{a,t}{\to} (q', n+1)$ for $q \in U_a(t)$ and $q' \in \delta(q, t[a, n]^{-1}t[a, n+1])$, when $n+1 < |t|_a$. Throughout this section we use the notion of transition graph with the meaning of a subgraph of $(U_a(t), \overset{a,t}{\to})$. Moreover we use the abbreviation $U_i(t)$ for $U_{a_i}(t)$, $1 \le i \le k$.
The basic idea for Klarlund's progress measure method is to express a global property of an infinite (transition) graph by the existence of a suitable mapping, which associates with each vertex a finite amount of information. This information quantifies progress towards satisfying the required condition. For complementing e.g. Büchi automata, the condition expresses that certain states are visited finitely often, i.e. they are not recurrent. In our distributed setting, we have to assume the existence of several progress measures, one for each connected component of $\mathrm{alphinf}(t)$ (i.e., for each designated letter).
The next proposition gives the basis for the complementation procedure. Recall $Q = \prod_{a\in\Sigma} Q_a$ and let $N := |Q|$. Let $F_i := \{q = (q_a)_{a\in\Sigma} \in Q \mid q_{a_i} = p_{a_i}\}$, where $p \in Q$ will be the first component in the unique element of the table of the given automaton.

Proposition 4.1 *Let $A = \dot{\bigcup}_{i=1}^{k} A_i$ be the decomposition in connected components of $A \subseteq \Sigma$, $a_i \in A_i$, for all i, and $p \in Q$. Let $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0, \{T\})$ be an asynchronous cellular Büchi automaton, with $T = (p, A, \{a_1, \dots, a_k\})$.*
Then $t \in \mathrm{Inf}(A) \setminus L(\mathcal{A})$ holds if and only if there exists a family of transition graphs $(V_i(t), E_i(t))$, together with associated mappings $\Phi_i \colon U_i(t) \to \{0, \dots, 2N+1\}$, $1 \le i \le k$, such that the following conditions hold for each i:

1. *$(V_i(t), E_i(t))$ is the subgraph of the transition graph $(U_i(t), \overset{a_i,t}{\to})$ induced by $V_i(t) = U_i(t) \setminus \Phi_i^{-1}(2N + 1)$.*

2. *$\Phi_i \colon U_i(t) \to \{0, 1, \dots, 2N + 1\}$ satisfies for every $(q, n), (q', n + 1) \in U_i(t)$ with $(q, n) \overset{a_i,t}{\to} (q', n + 1)$:*

- *$\Phi_i(q,n) \geq \Phi_i(q',n+1)$*
- *$\Phi_i(q,n) = \Phi_i(q',n+1) \implies q' \notin F_i$ or $\Phi_i(q,n) \in \{0,2,\dots,2N\} \cup \{2N+1\}$.*

3. *Let $((q_n,n))_{n\geq n_0} \subseteq V_i(t)$ be an infinite sequence with $(q_n,n) \overset{a_i,t}{\to} (q_{n+1},n+1)$, $n \geq n_0$. Then*
$$\lim_{n\to\infty} \Phi_i(q_n,n) \in \{1,3,\dots,2N-1\}\,.$$

4. *There exists a finite prefix $t_0 \leq t$, $t_0 \in \mathbb{M}(\Sigma,D)$, 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 $\delta(r,t_0)_a \neq p_a$ for some $a \in \bar{A}$,*
   - *or $(q_i,n_i) \in V_i(t)$ for some $1 \leq i \leq k$, where $q_i = \delta(r,\partial_{a_i}(t_0))$ and $n_i = |t_0|_{a_i} - 1$.*

Remark 4.2 Following Klarlund's construction for Büchi ω-automata [11], each mapping Φ_i is a quasi-progress measure for the subgraph $(V_i(t),E_i(t))$. Note that since Φ_i is decreasing w.r.t the transition relation $\overset{a_i,t}{\to}$, every transition graph $(V_i(t),E_i(t))$ is downward closed, i.e. for $(q,n) \in V_i(t)$ and $(q',n+1)$ with $(q,n) \overset{a_i,t}{\to} (q',n+1)$ we have $(q',n+1) \in V_i(t)$, too.

Condition 4 corresponds to a safety requirement. It guarantees that computations which reach the final halting state for each letter $a \notin A$ and which are *not* captured by some quasi-progress measure Φ_i can not be synchronized to a run on t. The existence of Φ_i ensures that the local state p_{a_i} is not recurrent (for a subset of runs on t). This means that we eliminate every possibility for runs $r \in R_{\mathcal{A}}(t)$ on t to be accepting.

Proof of 4.1: Let $t \in \mathrm{Inf}(A)$. Assume that a family of transition graphs $(V_i(t),E_i(t))$ exists, together with associated mappings Φ_i, $1 \leq i \leq k$, satisfying the conditions of the proposition. Let $r \in R_{\mathcal{A}}(t)$ be a run on t. It suffices to consider the case where for some connected component $1 \leq i \leq k$ and some $n \in \mathbb{N}$ we have $(\delta(r,t[a_i,n]),n) \in V_i(t)$ (hence, also $(\delta(r,t[a_i,m]),m) \in V_i(t)$ holds, for every $m \geq n$). Due to Φ_i being decreasing (Cond. (2)) and taking values in a finite set, we may assume with $q_m := \delta(r,t[a_i,m])$ that we have $\Phi_i(q_m,m) = \Phi_i(q_n,n)$ for every $m \geq n$. With Cond. (3) we obtain $\Phi_i(q_n,n) \notin \{0,2,\dots,2N\} \cup \{2N+1\}$, hence by Cond. (2), $q_m \notin F_i$ for all $m \geq n$. Therefore, the run $r \in R_{\mathcal{A}}(t)$ is rejecting. Hence, $t \notin L(\mathcal{A})$.

For the converse direction, let $t \in \mathrm{Inf}(A) \setminus L(\mathcal{A})$ and let $1 \leq i \leq k$ be fixed.
We follow [11] and define the mappings Φ_i in two steps. First, *progress measures* $\tilde{\Phi}_i$ with values in the set of countable ordinals ω_1 are defined by transfinite induction. In the present setting, we simultaneously define the computation graphs

$(V_i(t), E_i(t))$. For a set $V \subseteq U_i(t)$ we denote in the following by $N_V(q,n)$ the set of proper successors of $(q,n) \in U_i(t)$ from $U_i(t) \setminus V$, i.e.,

$$\begin{aligned} N_V(q,n) \;:=\; & \{(q',m) \mid m > n,\ \exists q = q_n, q_{n+1}, \ldots, q_m = q',\ (q',m) \in\ U_i(t) \setminus V \\ & \text{with } (q_k,k) \overset{a_i,t}{\to} (q_{k+1}, k+1),\ \text{for every } n \le k < m\}\,. \end{aligned}$$

The transition graph $(V_i(t), E_i(t))$ and the progress measure $\tilde{\Phi}_i \colon U_i(t) \to \omega_1$ are now defined inductively: let $V_0 = V_i(t) = \emptyset$. Assume that for $\beta < \omega_1$, the sequence $(V_\alpha)_{\alpha<\beta}$ of pairwise disjoint subsets of $U_i(t)$ is already defined, with $V_i(t) = \cup_{\alpha<\beta} V_\alpha$.
If either $U_i(t) = V_i(t)$, or $N_{V_i(t)}(q,n) \cap (F_i \times \mathbb{N}) \neq \emptyset$ holds for all $(q,n) \in U_i(t) \setminus V_i(t)$, then let $V_\beta := U_i(t) \setminus V_i(t)$ and $V_\gamma := \emptyset$ for every $\beta < \gamma < \omega_1$ and we are done.
Otherwise choose $(q,n) \in U_i(t) \setminus V_i(t)$ satisfying $N_{V_i(t)}(q,n) \cap (F_i \times \mathbb{N}) = \emptyset$ and define

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

updating $V_i(t) := V_i(t) \cup V_\beta$.
The decomposition of $U_i(t)$ in pairwise disjoint sets naturally induces a mapping $\tilde{\Phi}_i \colon U_i(t) \to \omega_1$. Let $\tilde{\Phi}_i(q,n) = \beta$ if $(q,n) \in V_\beta$.
It is easy to see that $\tilde{\Phi}_i$ is decreasing w.r.t. the transition relation $\overset{a_i,t}{\to}$, due to the definition by means of successors in the transition graph. Moreover, suppose $\tilde{\Phi}_i(q,n) = \tilde{\Phi}_i(q', n+1)$ holds for $(q,n) \in V_i(t)$, with $(q,n) \overset{a_i,t}{\to} (q', n+1)$. Then we have

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

By construction we obtain a countable ordinal β_0 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 $(s_n,n) \overset{a_i,t}{\to} (s_{n+1}, n+1) \overset{a_i,t}{\to} \cdots$ in the subgraph induced by V_{β_0}, repeating infinitely often some state from F_i, i.e.,

$$|\{m \ge n \mid (s_m)_{a_i} = p_{a_i}\}| = \infty\,. \tag{2}$$

The second step of Klarlund's construction [11] uses the bounded width N of the given transition graph for modifying the progress measure to a *quasi-progress measure* mapping with finite range. Given $\alpha < \omega_1$, let the predicate $\mathrm{const}(\alpha)$ be true, if there is an infinite path $(q_n,n) \overset{a_i,t}{\to} (q_{n+1}, n+1) \overset{a_i,t}{\to} \cdots$ in the transition graph $U_i(t)$, such that $\tilde{\Phi}_i(q_m,m) = \alpha$ for every $m \ge n$. Since these infinite paths are disjoint one obtains a set of at most N countable ordinals $0 < \alpha_1 < \cdots < \alpha_M < \omega_1$ ($M \le N$) satisfying $\mathrm{const}(\alpha_i)$, $1 \le i \le M$. Let $\alpha_0 = 0$, $\alpha_{M+1} = \omega_1$ and define $\Phi_i \colon U_i(t) \to \{0,1,\ldots,2N+1\}$ for $(q,n) \in V_i(t)$ as

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

For $(q,n) \in U_i(t) \setminus V_i(t)$ let $\Phi_i(q,n) = 2N+1$.
The mapping Φ_i satisfies the condition of real progress (Cond. (3)): assume by contradiction that an infinite transition path $(q_n,n) \overset{a_i,t}{\to} (q_{n+1},n+1) \overset{a_i,t}{\to} \cdots$ exists in $U_i(t)$, such that $\Phi_i(q_m,m) = 2k$ holds for some k and every $m \geq n$. With $\tilde{\Phi}_i$ being decreasing and by the definition of Φ_i, some $\alpha_k < \alpha < \alpha_{k+1}$ would exist with $\lim_{m\to\infty} \tilde{\Phi}_i(q_m,m) = \alpha$. Thus, $\mathrm{const}(\alpha)$ follows, contradicting the definition of $\{\alpha_1,\ldots,\alpha_M\}$.
The mapping Φ_i is obviously decreasing, since $\tilde{\Phi}_i$ already was. Moreover, suppose we have $(q,n) \overset{a_i,t}{\to} (q',n+1)$ with $\Phi_i(q,n) = \Phi_i(q',n+1) \neq 2N+1$ and $q' \in F_i$. By Eq. (1) we obtain $\tilde{\Phi}_i(q,n) > \tilde{\Phi}_i(q',n+1)$. Hence, $\Phi_i(q,n) \notin \{0,2,\ldots,2N\}$ leads to a contradiction.

Finally, we show that runs which are not covered by the quasi-progress measures $(\Phi_i)_{1\leq i\leq k}$ cannot be synchronized (Cond. 4). Let us assume by contradiction that for every tuple $(n_i)_{1\leq i\leq k} \in \mathbb{N}^k$ there exist global states $(q_i)_{1\leq i\leq k} \in Q^k$ and a run $r \in R_{\mathcal{A}}(t_0)$ on the (finite) prefix $t_0 = \bigsqcup_{1\leq i\leq k} t[a_i,n_i] \sqcup \bigsqcup_{a\in\bar{A}} t[a,|t|_a - 1]$ of t with

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

For large enough values of n_i, $1 \leq i \leq k$, we have $\max(t_0) \cap A = \{a_1,\ldots,a_k\}$, due to $t \in \mathrm{Inf}(A)$ and the way we choosed the designated letters a_i. Let $t = t_0 t_1 \cdots t_k$ with $\mathrm{alph}(t_i) = A_i$ for every i. Due to $(q_i,n_i) \in U_i(t) \setminus V_i(t)$ we obtain by Eq. (2) an infinite path π_i in the subgraph of $(U_i(t), \overset{a_i,t}{\to})$ induced by $U_i(t) \setminus V_i(t)$, such that π_i starts in (q_i,n_i) and visits infinitely often the set $F_i \times \mathbb{N}$. Since $\mathrm{alph}(t_0^{-1}t) = A$ every path π_i defines (a set of) runs on the connected suffix t_i of t, starting with the state q_i . Moreover, every run r_i associated to π_i repeats infinitely often the local state p_{a_i}. For each i we choose a run r_i on t_i associated to π_i as above. With V_i, $0 \leq i \leq k$, denoting the vertex set of t_i in the dependence graph of t, we define $r' \in R_{\mathcal{A}}(t)$ by $r'|_{V_0} := r$ and $r'|_{V_i} := r_i$, $1 \leq i \leq k$. Obviously, we obtained an accepting run on t, since $p_a \in \inf_a(r')$ for every $a \in \bar{A} \cup \{a_1,\ldots,a_k\}$. Hence, a contradiction follows. $\square$

We are now ready to define an asynchronous cellular Büchi automaton $\mathcal{B}$ such that $L(\mathcal{B}) = \overline{L(\mathcal{A})} \cap \mathrm{Inf}(A)$, where $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0, \{T\})$ is an asynchronous cellular Büchi automaton with a single element $T = (p, A, \{a_1,\ldots,a_k\})$ in $\mathcal{T}$. Following [11], the automaton $\mathcal{B}$ guesses the values of the quasi-progress measures. In this setting $\mathcal{B}$ guesses at the same time the transition graphs $(V_i(t), E_i(t))$ covered by Φ_i.
The automaton recognizing the complement language relies on the subset automaton of the given asynchronous cellular automaton $((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0)$, defined in Sect. 3 for an asynchronous mapping $\mu = (\nu, \rho)$ (we omit final states).

Let $\mathcal{A}_\mu = ((\tilde{Q}_a)_{a\in\Sigma}, (\tilde{\delta}_a)_{a\in\Sigma}, \tilde{q}_0)$ denote the subset automaton obtained from μ. In the following we denote by $[2N+1]^Q$ the set of partial mappings from Q to $\{0,1,\ldots,2N+1\}$ (recall $Q = \prod_{a\in\Sigma} Q_a$). Furthermore, $\mathrm{dom}(f)$ denotes the domain of $f \in [2N+1]^Q$. We recall the notation $q \overset{(a)}{\to} q'$ from Sect. 3, which means that for some $t \in \mathbb{M}(\Sigma, D)$ a run $r \in R_{\mathcal{A}}(\partial_a(ta))$ exists, with $q = \delta(r, \partial_a(t))$ and $q' = \delta(r, \partial_a(ta))$.
Let $\mathcal{B} = ((S_a)_{a\in\Sigma}, (\Delta_a)_{a\in\Sigma}, \mathcal{J}, \mathcal{T})$ be defined as follows (with $\mathcal{J}$ denoting the set of initial states):

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

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

   $$\mathrm{dom}(\alpha_a) = \{\delta(r,u) \mid \exists u \in \mathbb{M}(\Sigma,D),\ u = \partial_a(u),\ \tilde{\delta}(\tilde{q}_o, u)_a = \tilde{q}_a \text{ and } r \in R_{\mathcal{A}}(u)\}.$$

   For the subset construction of Prop. 3.4, the above condition is equivalent to the following one: if $\tilde{q}_a = (\nu(u), \rho(u))$, then we let $(f(b,a))_{b\in\Sigma} \in \mathrm{dom}(\alpha_a)$, for some $f \in \rho(u)$. If we use instead the subset construction given in Prop. 3.9 then we simply let $q \in \mathrm{dom}(\alpha_a)$, whenever $(q, q, \{a\}, a) \in \rho(u)$.

2. For $a \notin \{a_1,\ldots,a_k\}$ let $s'_a \in \Delta_a((s_b)_{b\in D(a)})$ if $s'_a = \tilde{\delta}_a((\tilde{q}_b)_{b\in D(a)})$, where $s_b = \tilde{q}_b$ or $s_b = (\tilde{q}_b, \alpha_b, A_b)$ for some $\alpha_b \in [2N+1]^Q$, $A_b \subseteq Q$, for all $b \in D(a)$.

   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

   - $\tilde{q}'_a = \tilde{\delta}_a((\tilde{q}_b)_{b\in D(a)})$, for $\tilde{q}_b = s_b$, $b \in D(a) \setminus \{a\}$ (note that $D(a) \cap \{a_1,\ldots,a_k\} = \{a\}$);
   - For every $q \in \mathrm{dom}(\alpha_a)$, $q' \in \mathrm{dom}(\alpha'_a)$ such that $q \overset{(a)}{\to} q'$ we require that both conditions below hold:
     i. $\alpha_a(q) \ge \alpha'_a(q')$
     ii. $\alpha_a(q) = \alpha'_a(q')$ implies $q'_a \ne p_a$ or $\alpha_a(q) \in \{0,2,\ldots,2N\}$ or $\alpha_a(q) = 2N+1$.
   - $$A'_a = \begin{cases} \mathrm{dom}(\alpha'_a) & \text{if } A_a = \emptyset \\ \{q' \in \mathrm{dom}(\alpha'_a) \mid \exists q \in \mathrm{dom}(\alpha_a) \cap A_a \text{ with} & \\ \qquad q \overset{(a)}{\to} q' \text{ and } \alpha_a(q) = \alpha'_a(q') \in \{0,2,\ldots,2N\}\} & \text{otherwise} \end{cases}$$

3. The table $\mathcal{T}$ is given by $(z, A, \{a_1,\ldots,a_k\}) \in \mathcal{T}$ 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 $\alpha_a \in [2N+1]^Q$, $a \in \{a_1,\ldots,a_k\}$ both conditions below are satisfied:

   1. $z_a = (\tilde{q}_a, \alpha_a, \emptyset)$, for $a \in \{a_1,\ldots,a_k\}$, respectively $z_a = \tilde{q}_a$ for $a \notin \{a_1,\ldots,a_k\}$.

2. Let $M = \bar{A} \cup \{a_1, \ldots, a_k\}$. Then we require for every $r \in R_{\mathcal{A}}(\partial_M(u))$:
   - either $\delta(r, \partial_M(u))_a \neq p_a$, for some $a \in \bar{A}$,
   - or $\alpha_a(\delta(r, \partial_a(u))) \neq 2N+1$, for some $a \in \{a_1, \ldots, a_k\}$.

   The conditions above can be easily checked using our subset automata constructions (Prop. 3.4, 3.9).

4. Let $s_0 \in \mathcal{J}$ be an initial state if $(s_0)_a = (\tilde{q}_0)_a$, resp. $(s_0)_a = ((\tilde{q}_0)_a, \alpha_a, \emptyset)$, for some $\alpha_a \in [2N+1]^Q$.

Proposition 4.3 *Let $\mathcal{A} = ((Q_a)_{a\in\Sigma}, (\delta_a)_{a\in\Sigma}, q_0, \{T\})$ be an asynchronous cellular Büchi automaton with a single table element $T = (p, A, \{a_1, \ldots, a_k\})$. Let $\mathcal{B}$ be defined as above.*
Then $L(\mathcal{B}) = \overline{L(\mathcal{A})} \cap \mathrm{Inf}(A)$.

Proof: Let us first assume $t \in \mathrm{Inf}(A) \setminus L(\mathcal{A})$. We apply Prop. 4.1 and obtain transition graphs $(V_i(t), E_i(t))_{1\leq i\leq k}$, together with quasi-progress measures $(\Phi_i)_{1\leq i\leq k}$ satisfying the properties in 4.1. The definition of an accepting $\mathcal{B}$-run r on t follows immediately:

1. For $a \notin \{a_1, \ldots, a_k\}$, $0 \leq n < |t|_a$, let $r(a,n) = \tilde{\delta}(\tilde{q}_0, t[a,n])_a$.

2. For $a = a_i$, $1 \leq i \leq k$ and $n \geq 0$ let $r(a,n) = (\tilde{q}_a, \alpha_a, A_a)$, where
   - $\tilde{q}_a = \tilde{\delta}(\tilde{q}_0, t[a,n])_a$;
   - Let $\tilde{q}_a = (\nu(u), \rho(u))$ for some $u = \partial_a(u)$. Then $\mathrm{dom}(\alpha_a) = \{q \in Q \mid \exists r \in R_{\mathcal{A}}(u) : \delta(r,u) = q\}$. For the construction given in Prop. 3.4 (resp. Prop. 3.9) this means $q \in \mathrm{dom}(\alpha_a)$ if and only if $q = (f(b,a))_{b\in\Sigma}$ for some $f \in \rho(u)$ (resp. $(q, q, \{a\}, a) \in \rho(u)$).
   - For all $q \in \mathrm{dom}(\alpha_a)$ let $\alpha_a(q) = \Phi_i(q,n)$.

The local state component A_a being computed deterministically, we note that the limit condition (Cond. (3)) in Prop. 4.1 implies for some $u \in \mathbb{M}(\Sigma, D)$ and partial mappings α_a, $a \in \{a_1, \ldots, a_k\}$, that the local states $(\tilde{p}_a, \alpha_a, \emptyset)$ with $\tilde{p}_a = (\nu(\partial_a(u)), \rho(\partial_a(u)))$ are repeated infinitely often. Otherwise, by the same argument as in [11] we would obtain using König's Lemma an infinite path in $(U_i(t), \overset{a_i,t}{\to})$ with a constant Φ_i-value from $\{0, 2, \ldots, 2N\}$.
For $a \in \bar{A}$ assume w.l.o.g. that the state $(\nu(\partial_a(u)), \rho(\partial_a(u)))$ labels the last a-vertex in t, i.e. $(a, |t|_a - 1)$. Finally, the synchronization condition (Cond. 4) in 4.1 ensures that there is no run $r \in R_{\mathcal{A}}(\partial_M(u))$ satisfying both $\delta(r, \partial_M(u))_a = p_a$, for all $a \in \bar{A}$, and $\alpha_a(\delta(r, \partial_a(u))) \neq 2N+1$ for all $a \in \{a_1, \ldots, a_k\}$ (recall that $\alpha_{a_i}(q) = 2N+1$, for all $q \notin V_i(t)$).
For the converse let $t \in L(\mathcal{B})$ be accepted by a $\mathcal{B}$-run r by $(z, A, \{a_1, \ldots, a_k\}) \in \mathcal{T}$, hence $\mathrm{alphinf}(t) = A$. Once again, there is a canonical definition for $(V_i(t), E_i(t))$

and Φ_i, $1 \leq i \leq k$. For $a = a_i$, $n \geq 0$, let $(\tilde{q}_a, \alpha_a, A_a) = \Delta(r, t[a,n])_a$. Furthermore, for $q \in \mathrm{dom}(\alpha_a)$ define $(q,n) \in V_i(t)$ if and only if $\alpha_a(q) \neq 2N+1$ and let $\Phi_i(q,n) = \alpha_a(q)$. Since for any $a \in \{a_1, \dots, a_k\}$, $z_a = (\tilde{q}_a, \alpha_a, \emptyset)$ for some $\tilde{q}_a \in \tilde{Q}_a$, $\alpha_a \in [2N+1]^Q$, we directly obtain that the real progress condition (Cond. (3)) from Prop. 4.1 holds. Finally, let $u \in \mathbb{M}(\Sigma, D)$, $u < t$ be such that $\Delta(r,u)_a = z_a$ for $a \in \bar{A} \cup \{a_1, \dots, a_k\}$ and $\max(u) \cap A = \{a_1, \dots, a_k\}$. Then with $n_i = |u|_{a_i} - 1$, $1 \leq i \leq k$, we satisfy Cond. 4 in Prop. 4.1, too. $\square$

For the size of the automaton recognizing the complement language note that the size of every local state set S_a of $\mathcal{B}$ is dominated by the size obtained for asynchronous cellular subset automata. Hence, by the construction based on Prop. 3.9 we obtain the bound $2^{N^{O(1)}}$, where the exponent of N is independent of the alphabet size. The lower bound for the complementation of Büchi ω-automata is $2^{N \log N}$ [15].

Theorem 4.4 *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 $N = |\prod_{a\in\Sigma} Q_a|$ global states.*
Then an asynchronous cellular Büchi automaton $\mathcal{B} = ((S_a)_{a\in\Sigma}, (\Delta_a)_{a\in\Sigma}, s_0, \mathcal{T}')$, with table $\mathcal{T}' \subseteq (\prod_{a\in\Sigma} S_a) \times \mathcal{P}(\Sigma) \times \mathcal{P}(\Sigma)$ effectively exists, such that $L(\mathcal{B}) = \overline{L(\mathcal{A})}$. The automaton $\mathcal{B}$ has $2^{N^{O(1)}}$ global states.

The construction of Prop. 4.3 can be adapted smoothly to asynchronous Büchi automata, similar to our subset constructions. Based on the subset constructions for asynchronous automata, the size blow-up obtained for the complementation procedure is again dominated by the subset automaton. Note that we can also start with an asynchronous Büchi automaton $\mathcal{A}$ (with modified acceptance condition) and use the asynchronous mapping ρ based on the asynchronous automaton $\mathcal{A}$ (given in Rem. 3.10) in the definition of $\mathcal{B}$. This yields an asynchronous cellular automaton $\mathcal{B}$ accepting the complement language and having local states sets of size $2^{n^{O(m)}} = 2^{N^{O(1)}}$, with N denoting the number of global states. Finally, by the standard embedding we transform $\mathcal{B}$ into an equivalent asynchronous automaton $\mathcal{B}'$ still having $2^{N^{O(1)}}$ global states.
Asynchronous (cellular) automata viewed as sequential automata satisfy the I-diamond property, as we already mentioned in Sect. 2. This means that for every pair of states q, q' and independent letters $(a,b) \in I$: $q' \in \delta(q, ab)$ holds if and only if $q' \in \delta(q, ba)$. This property expresses a reduced view of concurrency, the interleaving viewpoint. We note that non-deterministic Büchi ω-automata satisfying the I-diamond property which accept closed languages from Σ^∞ characterize precisely $\mathrm{Rec}(\mathbb{R}(\Sigma, D))$ (a language L is closed if $\varphi^{-1}(\varphi(L)) = L$, with φ denoting the canonical mapping). If we were only interested in Büchi automata $\mathcal{A}$ with I-diamond property, then we could apply for complementing $\mathcal{A}$ a simple construction proposed by Pécuchet [17]. All we need is a homomorphism $h\colon \Sigma^* \to S$ to a finite monoid recognizing $L(\mathcal{A})$ and satisfying $h(ab) = h(ba)$, for

all $(a,b) \in I$. Then the size of the automaton for the complement language obtained in [17] is $O(|S|^2)$. Note that the syntactic morphism of a closed language $L \subseteq \Sigma^\infty$ satisfies this property. Moreover, the size of the syntactic monoid of L is bounded by $2^{O(N^2)}$, with N denoting the size of $\mathcal{A}$.

Acknowledgment: I wish to thank Volker Diekert and Paul Gastin for interesting discussions and Wolfgang Thomas for pointing out the work of J.-P. Pécuchet. I also thank the referees from ICALP'93 and especially from *Theoretical Computer Science* for the comments which led to an improved presentation of this paper.

References

[1] P. Cartier and D. Foata. *Problèmes combinatoires de commutation et réarrangements.* Number 85 in Lecture Notes in Mathematics. Springer, Berlin-Heidelberg-New York, 1969.

[2] R. Cori and Y. Métivier. Approximation of a trace, asynchronous automata and the ordering of events in a distributed system. In *Proceedings of the 15th International Colloquium on Automata, Languages and Programming (ICALP'88)*, number 317 in LNCS, pages 147–161. Springer, 1988.

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

[4] V. Diekert. *Combinatorics on Traces.* Number 454 in Lecture Notes in Computer Science. Springer, Berlin-Heidelberg-New York, 1990.

[5] V. Diekert and W. Ebinger, editors. *Infinite Traces. Proceedings of a workshop of the ESPRIT Basic Research Action No 3166: Algebraic and Syntactic Methods in Computer Science (ASMICS), Tübingen, Germany, 1992*, Bericht 4/92. Universität Stuttgart, Fakultät Informatik, 1992.

[6] V. Diekert and A. Muscholl. Deterministic asynchronous automata for infinite traces. *Acta Informatica*, 31:379–397, 1994. A preliminary version was presented at STACS'93, Lecture Notes in Computer Science 665 (1993).

[7] V. Diekert and G. Rozenberg, editors. *The Book of Traces.* World Scientific, Singapore, 1995.

[8] W. Ebinger and A. Muscholl. Logical definability on infinite traces. *Theoretical Computer Science*, 154:67–84, 1996. A preliminary version was presented at ICALP'93, Lecture Notes in Computer Science 700 (1993).

[9] P. Gastin. Recognizable and rational trace languages of finite and infinite traces. In C. Choffrut et al., editors, *Proceedings of the 8th Annual Symposium on Theoretical Aspects of Computer Science (STACS'91), Hamburg 1991*, number 480 in Lecture Notes in Computer Science, pages 89–104, Berlin-Heidelberg-New York, 1991. Springer.

[10] P. Gastin and A. Petit. Asynchronous automata for infinite traces. In W. Kuich, editor, *Proceedings of the 19th International Colloquium on Automata, Languages and Programming (ICALP'92), Vienna (Austria) 1992*, number 623 in Lecture Notes in Computer Science, pages 583–594, Berlin-Heidelberg-New York, 1992. Springer.

[11] N. Klarlund. Progress measures for complementation of ω-automata with applications to temporal logic. In *Proc. of the 32nd Symposium on Foundations of Computer Science*, Oct. 1-4, 1991, San Juan, Puerto Rico, pages 358–367, 1991.

[12] N. Klarlund, M. Mukund, and M. Sohoni. Determinizing asynchronous automata. In S. Abiteboul and E. Shamir, editors, *Proceedings of the 21st International Colloquium on Automata, Languages and Programming (ICALP'94), Jerusalem (Israel) 1994*, number 820 in Lecture Notes in Computer Science, pages 130–141, 1994.

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

[14] A. Mazurkiewicz. Trace theory. In W. Brauer et al., editors, *Petri Nets, Applications and Relationship to other Models of Concurrency*, number 255 in Lecture Notes in Computer Science, pages 279–324, Berlin-Heidelberg-New York, 1987. Springer.

[15] M. Michel. Complementation is more difficult with automata on infinite words. Manuscript., 1988.

[16] A. Muscholl. On the complementation of Büchi asynchronous cellular automata. In S. Abiteboul and E. Shamir, editors, *Proceedings of the 21st International Colloquium on Automata, Languages and Programming (ICALP'94), Jerusalem (Israel) 1994*, number 820 in Lecture Notes in Computer Science, pages 142–153. Springer, 1994.

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

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

[19] G. Pighizzini. *Recognizable trace languages and asynchronous automata.* PhD thesis, Università di Milano, February 1993.

[20] S. Safra. On the complexity of ω-automata. In *Proc. of the 29th Symposium on Foundations of Computer Science*, pages 319–327, 1988.

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