International Journal of Foundations of Computer Science


# Typeness for $\omega$-Regular Automata*

Orna Kupferman
*School of Engineering and Computer Science*
*Hebrew University, Jerusalem 91904, Israel*
*orna@cs.huji.ac.il*

Gila Morgenstern
*School of Engineering and Computer Science*
*Hebrew University, Jerusalem 91904, Israel*
*gila@cs.huji.ac.il*

Aniello Murano†
*Dipartimento di Informatica ed Applicazioni*
*Università degli Studi di Salerno, 84081 Baronissi, Italy*
*murano@unisa.it*

Received (received date)
Revised (revised date)
Communicated by Editor's name

ABSTRACT

We introduce and study three notions of *typeness* for automata on infinite words. For an acceptance-condition class $\gamma$ (that is, $\gamma$ is weak, Büchi, co-Büchi, Rabin, or Streett), *deterministic $\gamma$-typeness* asks for the existence of an equivalent $\gamma$-automaton on the same deterministic structure, nondeterministic *$\gamma$-typeness* asks for the existence of an equivalent $\gamma$-automaton on the same structure, and *$\gamma$-powerset-typeness* asks for the existence of an equivalent $\gamma$-automaton on the (deterministic) powerset structure – one obtained by applying the subset construction. The notions are helpful in studying the complexity and complication of translations between the various classes of automata. For example, we prove that deterministic Büchi automata are co-Büchi type; it follows that a translation from deterministic Büchi to deterministic co-Büchi automata, when exists, involves no blow up. On the other hand, we prove that nondeterministic Büchi automata are not co-Büchi type; it follows that a translation from a nondeterministic Büchi to nondeterministic co-Büchi automata, when exists, should be more complicated than just redefining the acceptance condition. As a third example, by proving that nondeterministic co-Büchi automata are Büchi-powerset type, we show that a translation of nondeterministic co-Büchi to deterministic Büchi automata, when exists, can be done applying the subset construction. We give a complete picture of typeness for the weak, Büchi, co-Büchi, Rabin, and Streett acceptance conditions, and discuss its usefulness.

*Keywords:* Automata on Infinite Words, Acceptance Conditions

## 1. Introduction

*A preliminary version of this paper appears in the Proceedings of the 2nd International Symposium on Automated Technology for Verification and Analysis, 2004.

†This work was done while the author was visiting the Hebrew University.

Finite *automata on infinite objects* were first introduced in the 60's. Motivated by decision problems in mathematics and logic, Büchi, McNaughton, and Rabin developed a framework for reasoning about infinite word and infinite trees [4, 14, 18]. The framework has proved to be very powerful. Automata, and their tight relation to second-order monadic logics, were the key to the solution of several fundamental decision problems in mathematics and logic [22]. Today, automata on infinite objects are used for *specification* and *verification* of nonterminating systems. In the automata-theoretic approach to verification, we reduce questions about systems and their specifications to questions about automata. More specifically, questions such as satisfiability of specifications and correctness of systems with respect to their specifications are reduced to questions such as nonemptiness and language containment [23, 10, 24]. The automata-theoretic approach separates the logical and the combinatorial aspects of reasoning about systems. The translation of specifications to automata handles the logic and shifts all the combinatorial difficulties to automata-theoretic problems. Recent industrial-strength property-specification languages such as Sugar [2], ForSpec [1], and the recent standard PSL 1.01 [www.accellera.org] include regular expressions and/or automata, making the automata-theoretic approach even more essential.

Since a run of an automaton on an infinite word does not have a final state, acceptance is determined with respect to the set of states visited infinitely often during the run. There are many ways to classify an automaton on infinite words. One is the class of its acceptance condition. For example, in *Büchi* automata, some of the states are designated as accepting states, and a run is accepting iff it visits states from the accepting set infinitely often [4]. Dually, in *co-Büchi* automata, a run is accepting iff it visits states from the accepting set only finitely often. More general are *Rabin* automata. Here, the acceptance condition is a set $\alpha = \{\langle G_1, B_1\rangle, \ldots, \langle G_k, B_k\rangle\}$ of pairs of sets of states, and a run is accepting if there is a pair $\langle G_i, B_i\rangle$ for which the set of states visited infinitely often intersects $G_i$ and is disjoint to $B_i$. The condition $\alpha$ can also be viewed as a *Streett* condition, in which case a run is accepting if for all pairs $\langle G_i, B_i\rangle$, if the set of states visited infinitely often intersects $G_i$, then it also intersects $B_i$. The number $k$ of pairs in $\alpha$ is referred to as the *index* of the automaton. Another way to classify an automaton is by the type of its branching mode. In a *deterministic* automaton, the transition function $\delta$ maps a pair of a state and a letter into a single state. The intuition is that when the automaton is in state $q$ and it reads a letter $\sigma$, then the automaton moves to state $\delta(q, \sigma)$, from which it should accept the suffix of the word. When the branching mode is *nondeterministic*, $\delta$ maps $q$ and $\sigma$ into a set of states, and the automaton should accept the suffix of the word from one of the states in the set.

The applications of automata theory in reasoning about systems have led to the development of new classes of automata. In [17], Muller et al. introduced *weak automata*. Weak automata can be viewed as a special case of Büchi or co-Büchi automata in which every strongly connected component in the graph induced by the structure of the automaton is either contained in the accepting set or is disjoint from it. Since reasoning about specifications is often done by recursively reasoning about their sub-specifications, known translations of temporal-logic specifications to Büchi automata actually result in weak automata [17, 9, 7]. The special structure of weak automata is reflected in their attractive computational properties and makes them very appealing. Essentially, while the formulation of acceptance by a Büchi or a co-Büchi automaton involves alternation between least and greatest fixed-points, no alternation is required for specifying acceptance by a weak automaton [9]. Deterministic weak automata have recently being used to represent real numbers. A real number $x$ in base $r$ is represented by a word in the form $w_i \bullet w_f$ where $w_i$ is the integer part of $x$ and $w_f$ is the float part of $x$, and both are words over the alphabet $\{0, 1, ..., r-1\}$. This way for instance, the real number $5\frac{1}{2}$ in base $r = 10$ is represented by $0^*5 \bullet 50^\omega$ or by $0^*5 \bullet 49^\omega$. In a similar way, a vector $v = \langle x_1, x_2, ..., x_n \rangle$ of real numbers is represented by a word of the form $W_i \bullet W_f$ where $W_i$ is in $(\{0, 1, ..., r-1\}^n)^*$ and $W_f$ is in $(\{0, 1, ..., r-1\}^n)^\omega$. As real numbers may have several representations, real vectors may have several representations too. A *real vector automaton* is a Büchi automaton that either accepts all the representations of some vector $v \in \mathbb{R}^n$ or none of them. It is proved in [3] that an RVA is a deterministic weak automaton.

It turns out that different classes of automata have different *expressive power*. For example, unlike automata on finite words, where deterministic and nondeterministic automata have the same expressive power, deterministic Büchi automata are strictly less expressive than nondeterministic Büchi automata [11]. That is, there exists a language $\mathcal{L}$ over infinite words such that $\mathcal{L}$ can be recognized by a nondeterministic Büchi automaton but cannot be recognized by a deterministic Büchi automaton. It also turns out that some classes of automata may be more *succinct* than other classes. For example, translating a nondeterministic co-Büchi automaton into a deterministic one is possible [16], but involves an exponential blow up. As another example, translating a nondeterministic Rabin automaton with $n$ states and index $k$, into an equivalent nondeterministic Büchi automaton may result in an automaton with $O(k \cdot n)$ states, and if we start with a Streett automaton, the Büchi automaton may have $n \cdot 2^{O(k)}$ states [21]. Note that expressiveness and succinctness depend in both the branching type of the automaton as well as the class of its acceptance condition.

There has been extensive research on expressiveness and succinctness of automata on infinite words [22]. In particular, since reasoning about deterministic automata is simpler than reasoning about nondeterministic ones, questions like deciding whether a nondeterministic automaton has an equivalent deterministic one, and the blow-up involved in determinization are of particular interest. These questions get further motivation with the discovery that many natural specifications correspond to the deterministic fragments: it is shown in [8] that an LTL formula $\psi$ has an equivalent alternation-free $\mu$-calculus formula iff $\psi$ can be recognized by a deterministic Büchi automaton, and, as mentioned above, real vector automata are deterministic weak automata.

For deterministic automata, where Büchi and co-Büchi automata are less expressive than Rabin and Streett automata, researchers have come up with the notion of a deterministic automaton being *Büchi type*, namely it has an equivalent Büchi automaton on the same structure [6]. It is shown in [6] that Rabin automata are Büchi type. Thus, if a deterministic Rabin automaton $\mathcal{A}$ recognizes a language that can be recognized by a deterministic Büchi automaton, then $\mathcal{A}$ has an equivalent deterministic Büchi automaton on the same structure. On the other hand, Streett automata are not Büchi type: there is a deterministic Streett automaton $\mathcal{A}$ that recognizes a language that can be recognized by a deterministic Büchi automaton, but all the possibilities of defining a Büchi acceptance condition on the structure of $\mathcal{A}$ result in an automaton recognizing a different language.

As discussed in [6], Büchi-typeness is a very useful notion. In particular, a Büchi-type deterministic automaton can be translated to an equivalent deterministic Büchi automaton with no blow up. In this work, we study *typeness* in general: we consider both nondeterministic and deterministic automata, for the five classes $\gamma$ of acceptance conditions described above ($\gamma$ is either Büchi, co-Büchi, Rabin, Streett, or weak). We define and examine three notion of typeness:

1. *Deterministic $\gamma$-typeness* asks for which classes of deterministic automata, the existence of some equivalent deterministic $\gamma$ automaton implies the existence of an equivalent deterministic $\gamma$ automaton on the same structure. For example, we show that all deterministic automata are weak type.

2. *Nondeterministic $\gamma$-typeness* asks for which classes of nondeterministic automata, the existence of some equivalent nondeterministic $\gamma$ automaton implies the existence of an equivalent nondeterministic $\gamma$ automaton on the same structure. For example, we show that nondeterministic Büchi automata are not co-Büchi type. This answers a question on translating Büchi to co-Büchi automata that was left open in [8].

3. *$\gamma$-powerset-typeness* asks for which classes of nondeterministic automata, the existence of some equivalent deterministic $\gamma$ automaton implies the existence of an equivalent deterministic $\gamma$ automaton on the structure obtained by applying the subset construction to the original automaton. For example, while deterministic Rabin automata are Büchi-type, nondeterministic Rabin automata are not Büchi powerset-type. The notion of powerset-typeness is important for the study of the blow-up involved in the translation of automata to equivalent deterministic ones. While for some classes a $2^{O(n \log n)}$ lower bound is known, powerset-typeness implies a $2^n$ upper bound for other classes. We also examine *finite-typeness* for nondeterministic Büchi automata – cases where the limit language of the automaton when viewed as an automaton on finite words is equivalent to that of the Büchi automaton, and we relate finite-typeness with powerset-typeness.

Our results, along with previously known results, are described in Figures 2, 3, and 5.

## 2. Preliminaries

Given an alphabet $\Sigma$, an *infinite word* over $\Sigma$ is an infinite sequence $w = \sigma_0 \cdot \sigma_1 \cdot \sigma_2 \cdots$ of letters in $\Sigma$. We denote the set of all infinite words over $\Sigma$ by $\Sigma^\omega$. A *language* $\mathcal{L}$ is a set of words from $\Sigma^\omega$. An *automaton over infinite words* is a tuple $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$, where $\Sigma$ is the input alphabet, $Q$ is a finite set of states, $\delta : Q \times \Sigma \to 2^Q$ is a transition function, $Q_0 \subseteq Q$ is a set of initial states, and $\alpha$ is an acceptance condition which is a condition that defines a subset of $Q^\omega$. We define several acceptance conditions below. Intuitively, $\delta(q, \sigma)$ is the set of states that $\mathcal{A}$ may move into when it is in the state $q$ and it reads the letter $\sigma$. The automaton $\mathcal{A}$ may have several initial states and the transition function may specify many possible transitions for each state and letter, and hence we say that $\mathcal{A}$ is *nondeterministic*. In the case where $|Q_0| = 1$ and for every $q \in Q$ and $\sigma \in \Sigma$, we have that $|\delta(q, \sigma)| = 1$, we say that $\mathcal{A}$ is *deterministic*.

Given an input infinite word $w = \sigma_0 \cdot \sigma_1 \cdot \sigma_2 \cdots \in \Sigma^\omega$, a *run* of $\mathcal{A}$ on $w$ can be viewed as a function $r : \mathbb{N} \to Q$ where $r(0) \in Q_0$, i.e., the run starts in one of the initial states, and for every $i \geq 0$, we have that $r(i+1) \in \delta(r(i), \sigma_i)$, i.e., the run obeys the transition function. Note that while a deterministic automaton has a single run on an input word $w$, a nondeterministic automaton may have several runs on $w$ or none at all. Each run $r$ induces a set $inf(r)$ of states that $r$ visits infinitely often. Formally, $inf(r) = \{q \in Q \ : \ \text{for infinitely many } i \in \mathbb{N}, \text{ we have } r(i) = q\}$. As $Q$ is finite, it is guaranteed that $inf(r) \neq \emptyset$. The run $r$ is *accepting* iff the set $inf(r)$ satisfies the acceptance condition $\alpha$. A run that is not accepting is *rejecting*. We consider the following acceptance conditions.

- A set $S$ satisfies a *Büchi* acceptance condition $\alpha \subseteq Q$ if and only if $S \cap \alpha \neq \emptyset$.
- A set $S$ satisfies a *co-Büchi* acceptance condition $\alpha \subseteq Q$ if and only if $S \cap \alpha = \emptyset$.
- A set $S$ satisfies a *Rabin* acceptance condition $\alpha = \{\langle G_1, B_1 \rangle, \ldots, \langle G_k, B_k \rangle\} \subseteq 2^Q \times 2^Q$ if and only if there exists a pair $\langle G_i, B_i \rangle \in \alpha$ for which $S \cap G_i \neq \emptyset$ and $S \cap B_i = \emptyset$.
- A set $S$ satisfies a *Streett* acceptance condition $\alpha = \{\langle G_1, B_1 \rangle, \ldots, \langle G_k, B_k \rangle\} \subseteq 2^Q \times 2^Q$ if and only if for all pairs $\langle G_i, B_i \rangle \in \alpha$ we have that $S \cap G_i = \emptyset$ or $S \cap B_i \neq \emptyset$.

Note that the Büchi acceptance condition is dual to the co-Büchi acceptance condition: a set $S$ satisfies a Büchi acceptance condition $\alpha$ iff $S$ does not satisfy $\alpha$ as a co-Büchi acceptance condition. Similarly, the Rabin acceptance condition is dual to the Streett acceptance condition. The number $k$ appearing in the Rabin and Street conditions is called the *index* of the automaton. An automaton $\mathcal{A}$ accepts an input word $w$ iff there exists an accepting run of $\mathcal{A}$ on $w$. The language of $\mathcal{A}$, denoted $\mathcal{L}(\mathcal{A})$, is the set of all infinite words that $\mathcal{A}$ accepts.

The transition function $\delta$ induces a relation $R_\delta \subseteq Q \times Q$, where $R_\delta(q, q')$ iff there is $\sigma \in \Sigma$ with $\delta(q, \sigma) = q'$. Accordingly, the automaton $\mathcal{A}$ induces a graph $G_\mathcal{A} = \langle Q, R_\delta \rangle$. For two states, $q$ and $q'$ of $\mathcal{A}$, we say that $q'$ is *reachable* from $q$ if there is a (possibly empty) path in $G_{\mathcal{A}}$ from $q$ to $q'$. A *strongly connected component* (*SCC*, for short) in $G_{\mathcal{A}}$ is a set $C \subseteq Q$ such that for all states $q$ and $q'$ in $C$, we have that $q$ is reachable from $q'$. The SCC $C$ is *non-trivial* if the restriction of $G_{\mathcal{A}}$ to $C$ contains a cycle; that is, either $C$ has at least two states, or $C$ has a state with a self loop. A *maximal strongly connected component* (*MSCC*, for short[a]) is an SCC $C$ that is maximal in the sense that we cannot add to $C$ states and stay with an SCC. Thus, for all $C' \subseteq Q \setminus C$, the set $C \cup C'$ is not an SCC. Note that a run of an automaton $\mathcal{A}$ eventually get trapped in an MSCC of $G_{\mathcal{A}}$. We say that a Büchi automaton $\mathcal{A}$ is *weak* if for each MSCC $C$ of $G_{\mathcal{A}}$, either $C \subseteq \alpha$ (in which case we say that $C$ is an *accepting component*) or $C \cap \alpha = \emptyset$ (in which case we say that $C$ is a *rejecting component*). Note that a weak automaton can be viewed as both a Büchi and a co-Büchi automaton. Indeed, a run of $\mathcal{A}$ visits $\alpha$ infinitely often iff it gets trapped in an accepting component, which happens iff it visits states in $Q \setminus \alpha$ only finitely often.

We denote the different types of automata by three letters acronyms in $\{\text{D}, \text{N}\} \times \{\text{F, B, C, R, S, W}\} \times \{\text{W,T}\}$. The first letter stands for the branching mode of the automaton (deterministic or nondeterministic); the second letter stands for the acceptance-condition type (finite, Büchi, co-Büchi, Rabin, Streett, or weak). The third letter stands for the objects on which the automata run (words or trees). For Rabin and Streett automata, we sometimes also indicate the index of the automaton. In this way, for example, NBW are nondeterministic Büchi word automata, and DRW[1] are deterministic Rabin automata with index 1.

### *2.1. Expressiveness and Typeness*

For two automata $\mathcal{A}$ and $\mathcal{A}'$, we say that $\mathcal{A}$ and $\mathcal{A}'$ are *equivalent* if $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$. For an automaton type $\beta$ (e.g., DBW) and an automaton $\mathcal{A}$, we say that $\mathcal{A}$ is $\beta$-realizable if there is a $\beta$-automaton equivalent to $\mathcal{A}$. In Figure 1 below we describe the known expressiveness hierarchy for automata on infinite words. As described in the figure, DRW and DSW are as expressive as NRW, NSW, and NBW, which recognize all $\omega$-regular language [14]. On the other hand, DBW are strictly less expressive than NBW, and so are DCW. In fact, since by dualizing a Büchi automaton we get a co-Büchi automaton, the two internal ovals complement each other. The intersection of DBW and DCW is DWW (note that while a DWW is clearly both a DBW and DCW, the other direction is not trivial, and is proven in [3]). Finally, NCW can be determinized (when applied to universal Büchi automata, the translation in [16], of alternating Büchi automata into NBW, results in DBW. By dualizing it, one gets a translation of NCW to DCW). In addition to the results described in the figure, the index of DRW and DSW also induces a hierarchy, thus DRW$[k+1]$ are strictly more expressive than DRW$[k]$, and similarly for DSW [5].

Consider an automaton $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$. We refer to $\langle Q, \delta, Q_0 \rangle$ as the *structure* of the automaton. The *powerset structure* induced by $\mathcal{A}$ is $\mathcal{P}(\mathcal{A}) = \langle 2^Q, \delta_{\mathcal{P}}, \{Q_0\} \rangle$, where for all $S \in 2^Q$ and $\sigma \in \Sigma$, we have that $\delta_{\mathcal{P}}(S, \sigma) = \bigcup_{s \in S} \delta(s, \sigma)$.

[a]The notation SCC is sometimes used in the literature to denote maximal SCC. Here, we use both MSCC (maximal SCC) and SCC (not necessarily maximal SCC).

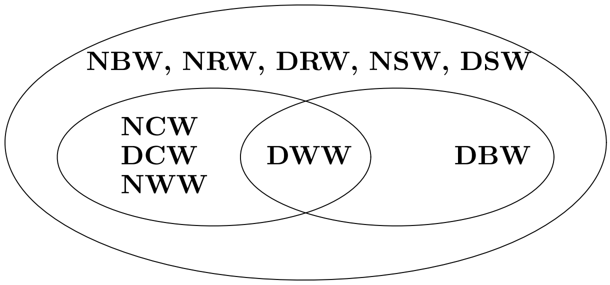


Figure 1: The expressiveness hierarchy for $\omega$-regular automata.

Thus, the powerset structure is obtained by the usual subset construction [19].

For an acceptance-condition class $\gamma$ (e.g., Büchi), we say that $\mathcal{A}$ is *$\gamma$-type* if $\mathcal{A}$ has an equivalent $\gamma$ automaton with the same structure as $\mathcal{A}$. That is, there is an automaton $\mathcal{A}' = \langle \Sigma, Q, \delta, Q_0, \alpha' \rangle$ such that $\alpha'$ is an acceptance condition of class $\gamma$ and $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$. We say that $\mathcal{A}$ is *$\gamma$-powerset-type* if $\mathcal{A}$ has an equivalent $\gamma$ automaton with the same structure as the powerset structure of $\mathcal{A}$. That is, there is an automaton $\mathcal{A}_\mathcal{P} = \langle \Sigma, 2^Q, \delta_\mathcal{P}, \{Q_0\}, \alpha_\mathcal{P} \rangle$ such that $\alpha_\mathcal{P}$ is an acceptance condition of class $\gamma$ and $\mathcal{L}(\mathcal{A}_\mathcal{P}) = \mathcal{L}(\mathcal{A})$. Note that the automaton $\mathcal{A}_\mathcal{P}$ is deterministic.

## 3. Typeness for Deterministic Automata

In this section we consider the following problem: given two acceptance-condition types $\beta$ and $\gamma$, is it true that every D$\beta$W that is D$\gamma$W-realizable, is also $\gamma$-type? We then say that D$\beta$W are $\gamma$-type. In other words, D$\beta$W are $\gamma$-type if every deterministic $\beta$-automaton that has an equivalent deterministic $\gamma$-automaton, also has an equivalent deterministic $\gamma$-automaton on the same structure.

Our results are described in Figure 2 below. Some results are immediate. For example, since the Büchi and the co-Büchi acceptance conditions are special cases of Rabin and Streett conditions (a Büchi condition $\alpha$ is equivalent to the Rabin condition $\{\langle \alpha, \emptyset \rangle\}$ and to the Streett condition $\{\langle Q, \alpha \rangle\}$, and dually for co-Büchi), it is clear that DBW and DCW are Rabin-type and Streett-type. Similarly, since weak automata can be viewed as Büchi or co-Büchi automata, they can also be viewed as a special case of Rabin and Streett automata. Thus, DWW are $\gamma$-type for all the types $\gamma$ we consider. Such cases, where a translation of the acceptance condition exists, and is independent of the automaton, are indicated in the table by $\hookleftarrow$. Some results are known, or obtained easily by dualizing known results, and the table contains the appropriate reference. Below we prove the new results.

**Lemma 1** *D$\beta$W are weak-type for all acceptance-condition types $\beta$.*

**Proof.** In [3], the authors introduce the notion of a deterministic automaton being *inherently weak* (the definition in [3] is for DBW, and is easily extended to D$\beta$W for all acceptance-condition types $\beta$). A D$\beta$W is inherently weak if none of its reachable MSCC contains both accepting and rejecting non-trivial SCCs. It is easy to see that an inherently weak automaton has an equivalent DWW on the same

| | DWW | DBW | DCW | DRW | DSW |
|---|---|---|---|---|---|
| DWW | | YES<br>Lemma 1 | YES<br>Lemma 1 | YES<br>Lemma 1 | YES<br>Lemma 1 |
| DBW | YES<br>$\hookleftarrow$ | | YES<br>Lemma 2 | YES<br>[6] | NO<br>[6] |
| DCW | YES<br>$\hookleftarrow$ | YES<br>Lemma 2 | | NO<br>dualizing [6] | YES<br>dualizing [6] |
| DRW | YES<br>$\hookleftarrow$ | YES<br>$\hookleftarrow$ | YES<br>$\hookleftarrow$ | | NO<br>Lemma 3 |
| DSW | YES<br>$\hookleftarrow$ | YES<br>$\hookleftarrow$ | YES<br>$\hookleftarrow$ | NO<br>Lemma 3 | |
| DRW[$k$] are not Rabin[$k-1$]-type, DSW[$k$] are not Streett[$k-1$]-type.<br>Lemma 4 | | | | | |

Figure 2: Typeness for deterministic automata.

structure. Indeed, by definition, each of the MSCC of the automaton can be made accepting or rejecting according to the classification of all its non-trivial SCCs.

Let $\mathcal{A}$ be a DWW-realizable D$\beta$W. Then, $\mathcal{A}$ is both DBW-realizable and DCW-realizable. Assume by the way of contradiction that $\mathcal{A}$ is not weak type. Then, $\mathcal{A}$ is not inherently weak, so there exists a reachable MSCC $C$ of $\mathcal{A}$ such that $C$ contains both an accepting non-trivial SCC $S$ and a rejecting non-trivial SCC $R$. Since $\mathcal{A}$ is DBW-realizable, then, by [11], every SCC $S' \supseteq S$ is accepting. In particular, $C$ is accepting. Dually, Since $\mathcal{A}$ is DCW-realizable, then every SCC $R' \supseteq R$ is rejecting. In particular, $C$ is rejecting. It follows that $C$ is both accepting and rejecting, and we reach a contradiction. □

We note that [3] prove that every DBW that accepts a language in $F_\sigma \cap G_\delta$ is inherently weak. The proof there, however, does not make a direct use of [11], and is therefore much more complicated.

**Lemma 2** *DCW are Büchi-type, and DBW are co-Büchi-type.*

**Proof.** Since a DCW can be viewed as a DRW, and DRW are Büchi type [6], DCW are Büchi type too. Dually, DBW are co-Büchi-type. □

Note that if a DCW $\mathcal{A}$ is DBW-realizable, then it is also DWW-realizable. Indeed, by [3], DCW $\cap$ DBW = DWW. Hence, by Lemma 1, $\mathcal{A}$ has an equivalent deterministic weak automaton on the same structure. Thus, Lemma 2 can be strengthened: a DCW that is DBW-realizable (dually, a DBW that is DCW-realizable) has an equivalent deterministic weak automaton on the same structure.

**Lemma 3** *DRW are not Streett-type, and DSW are not Rabin-type.*

**Proof.** Since DSW can recognize all $\omega$-regular languages, DSW being Rabin-type means that every DSW has an equivalent DRW on the same structure. In [12], Löding shows that a translation of a DSW to an equivalent DRW may involve an exponential blow up, thus typeness obviously cannot hold. The argument for DRW is dual. □

In addition to the results in the table, we prove that the expressiveness hierarchy known for the indices of DRW and DSW induces a typeness hierarchy:

**Lemma 4** *For all $k \geq 2$, we have that DRW[k] are not Rabin[$k-1$]-type, and DSW[k] are not Streett[$k-1$]-type.*

**Proof.** Let $\Sigma_k = \{1, 2, \ldots, k\}$. Consider the languages $L_k$ of exactly all words containing infinitely many $i$'s, for all $1 \leq i \leq k$. Consider the DSW[$k$] $\mathcal{A}_k = \langle \Sigma_k, \Sigma_k, \delta, \{1\}, \alpha_k \rangle$, with $\delta(q, i) = i$, for all $q, i \in \Sigma_k$, and $\alpha_k = \{\langle \Sigma_k, \{1\}\rangle, \langle \Sigma_k, \{2\}\rangle, \ldots, \langle \Sigma_k, \{k\}\rangle\}$. Thus, whenever $\mathcal{A}_k$ reads a letter $i$, it moves to state $i$, and the acceptance condition requires an accepting run to visit all states infinitely often. It is easy to see that $\mathcal{A}_k$ recognizes $L_k$. Also, since $L_k$ can be viewed as the intersection of $k$ DBWs $\mathcal{D}_i$, each for the language "infinitely many $i$'s," we know that $L_k$ is DBW-recognizable, and hence also DSW[$k-1$]-realizable. On the other hand, it is impossible to define a Streett[$k-1$] acceptance condition $\alpha'_k$ so that $\mathcal{A}_k$ with condition $\alpha'_k$ recognizes $L_k$. To see this, note that for each letter $i \in \Sigma_k$, the DSW $\mathcal{A}_k$ accepts $(1 \cdot 2 \cdots k)^\omega$ and rejects $(1 \cdot 2 \cdots i-1 \cdot i+1 \cdots k)^\omega$. For that, $\mathcal{A}_k$ must contain, for each $i \in \Sigma_k$, a pair $\langle G_i, B_i \rangle$ such that $G_i \cap \Sigma_k \neq \emptyset$ and $B_i = \{i\}$. Thus, $\mathcal{A}_k$ must contain at least $k$ pairs, and we are done. It follows that DSW[$k$] are not Streett[$k-1$]-type. The argument for Rabin automata is dual, and considers the complement of $L_n$. □

## 4. Typeness for Nondeterministic Automata

In this section we consider the following problem: given two acceptance-condition types $\beta$ and $\gamma$, is it true that every N$\beta$W that is N$\gamma$W-realizable, is also $\gamma$-type? We then say that N$\beta$W are $\gamma$-type. In other words, N$\beta$W are $\gamma$-type if every nondeterministic $\beta$-automaton that has an equivalent nondeterministic $\gamma$-automaton, also has an equivalent nondeterministic $\gamma$-automaton on the same structure.

Our results are described in Figure 3 below. As in Section 3, some results follow immediately from translations of the acceptance condition, and are indicated in the table by $\hookleftarrow$. The new results are proven in Lemmas 5, 6, and 7. When the results follow from applying translations to results proven in the Lemmas, we indicate it with $\hookleftarrow$ too.

| | NWW | NBW | NCW | NRW | NSW |
|---|---|---|---|---|---|
| NWW | | NO<br>Lemma 5 | NO<br>Lemma 6 | NO<br>Lemmas 5 and 6 $\hookleftarrow$ | NO<br>Lemmas 5 and 6 $\hookleftarrow$ |
| NBW | YES<br>$\hookleftarrow$ | | NO<br>Lemma 6 | NO<br>Lemma 6 $\hookleftarrow$ | NO<br>Lemma 6 $\hookleftarrow$ |
| NCW | YES<br>$\hookleftarrow$ | NO<br>Lemma 5 | | NO<br>Lemma 5 $\hookleftarrow$ | NO<br>Lemma 5 $\hookleftarrow$ |
| NRW | YES<br>$\hookleftarrow$ | YES<br>$\hookleftarrow$ | YES<br>$\hookleftarrow$ | | NO<br>Lemma 7 |
| NSW | YES<br>$\hookleftarrow$ | YES<br>$\hookleftarrow$ | YES<br>$\hookleftarrow$ | NO<br>Lemma 7 | |

Figure 3: Typeness for nondeterministic automata.

**Lemma 5** *NBW are neither co-Büchi- nor weak-type.*

**Proof.** Consider the NBW $\mathcal{A}_1$ described in Figure 4. The NBW recognizes the language $a^* \cdot b \cdot (a+b)^*$ (at least one $b$). This language is in NWW and NCW, yet it is easy to see that there is no NCW (and hence also no NWW) recognizing $L$ on the same structure. □

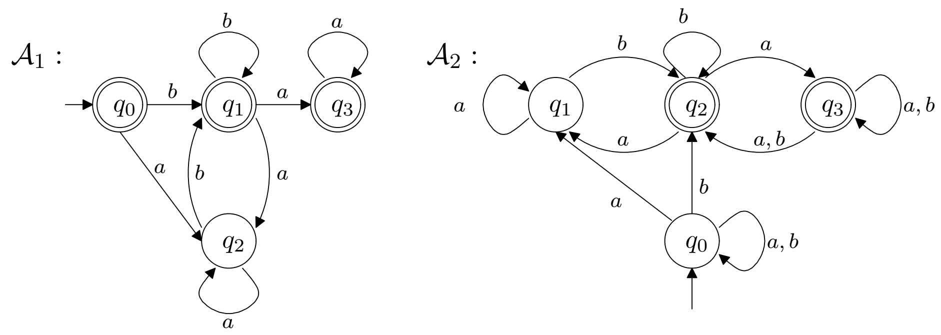


Figure 4: NBWs for $a^* \cdot b \cdot (a+b)^*$.

We note that the automaton in Figure 4 is a *single-run* automaton: every word accepted by it has a single accepting run. This is of particular interest in the context of specification and verification, as the NBW described in [24] for LTL formulas are single-run automata. Our example shows that even such automata are neither co-Büchi- nor weak-type. It is shown in [8] that an LTL formula $\psi$ has an equivalent alternation-free $\mu$-calculus formula $\psi'$ iff the language of $\psi$ can be recognized by a DBW $\mathcal{A}_\psi$. The construction of the formula $\psi'$ in [8] goes via $\mathcal{A}_\psi$, and therefore it involve a doubly-exponential blow-up. The construction of $\psi'$ may also go via an NCW $\tilde{\mathcal{A}}_\psi$, for $\neg\psi$. While $\psi'$ is of length linear in the size of $\tilde{\mathcal{A}}_\psi$, the best known translation of LTL to NCW (when exists) actually constructs a DCW and is doubly-exponential. It is conjectured in [8] that single-run NBW can be translated to NCW with only a linear blow up, leading to an exponential translation of LTL to alternation-free $\mu$-calculus. In particular, the question of obtaining the NCW by modifying the acceptance condition of the NBW is left open in [8]. Our result here answers the question negatively.

We also note that NCW-typeness and weak-typeness do not coincide. Figure 4 also describes a different NBW, $\mathcal{A}_2$, for $L$. This NBW is NCW-type: an NCW with the same structure but with the acceptance condition $\alpha = \{q_0, q_1\}$ accepts $L$. Yet, it is not weak-type.

**Lemma 6** *NCW are neither Büchi- nor weak-type.*

**Proof.** Consider the two-state DCW $\mathcal{A}$ for the language $L$ of all words with finitely many $a$'s. Since $L$ is not DBW-realizable, and $\mathcal{A}$ is deterministic, $\mathcal{A}$ is not Büchi-type. The language $L$ is NWW-realizable. But again, since $\mathcal{A}$ is deterministic and $L$ is not DWW-realizable, it is not weak-type. □

**Lemma 7** *NRW are not Streett-type, and NSW are not Rabin-type.*

**Proof.** By Lemma 3, DRW are not Streett-type. Hence, there are DRW that are DSW-realizable but do not have an equivalent DSW on the same structure. Since DRW are a special case of NRW, it follows that NRW are not Streett-type. The proof for NSW not being Rabin-type is similar. □

By Lemma 4, DRW[$k$] are not Rabin[$k-1$]-type, and DSW[$k$] are not Streett[$k-1$]-type, for all $k \geq 2$. Thus, following the same considerations as in the proof of Lemma 7, we get that NRW[$k$] are not Rabin[$k-1$]-type, and NSW[$k$] are not Streett[$k-1$]-type.

## 5. Powerset-Typeness for Nondeterministic Automata

In this section we consider the following problem: given two acceptance-condition types $\beta$ and $\gamma$, is it true that every N$\beta$W that is D$\gamma$W-realizable, is also $\gamma$-powerset-type? We then say that N$\beta$W are $\gamma$-powerset-type. In other words, N$\beta$W are $\gamma$-type if every nondeterministic $\beta$-automaton that has an equivalent deterministic $\gamma$-automaton, also has an equivalent deterministic $\gamma$-automaton on the powerset structure.

Our results are described in Figure 5 below. Since $\mathcal{A} = \mathcal{P}(\mathcal{A})$ for a deterministic automaton $\mathcal{A}$, we know that N$\beta$W cannot be $\gamma$-powerset-type if D$\beta$W are not $\gamma$-type. Thus, the negative cases in Figure 2 immediately induce negative cases here. In particular, for all $k \geq 2$, we have that NRW[$k$] are not Rabin[$k-1$]-powerset-type, and NSW[$k$] are not Streett[$k-1$]-powerset-type.

| | NWW | NBW | NCW | NRW | NSW |
|---|---|---|---|---|---|
| DWW | YES [13] | YES [13] | YES [13] | YES [13] | YES [13] |
| DBW | YES Lemma 8 | NO Lemma 9 | YES Lemma 8 | NO Lemma 9 ↩ | NO Lemma 9 ↩ |
| DCW | NO Lemma 10 | NO Lemma 10 ↩ | NO Lemma 10 ↩ | NO Lemma 10 ↩ | NO Lemma 10 ↩ |
| DRW | NO Lemma 10 | NO Lemma 10 ↩ | NO Lemma 10 ↩ | NO Lemma 10 ↩ | NO Lemma 10 ↩ |
| DSW | NO Lemma 10 | NO Lemma 10 ↩ | NO Lemma 10 ↩ | NO Lemma 10 ↩ | NO Lemma 10 ↩ |

Figure 5: Powerset-typeness for nondeterministic automata.

**Lemma 8** *NWW and NCW are Büchi-powerset-type.*

**Proof.** Consider an NCW $\mathcal{A}$. Recall that $\mathcal{A}$ is DCW-realizable. Therefore, if $\mathcal{A}$ is DBW-realizable, then it is also DWW-realizable. Hence, as NCW are weak-powerset-type, there is a DWW, and thus also a DBW, equivalent to $\mathcal{A}$ with structure $\mathcal{P}(\mathcal{A})$. Thus, NCW are Büchi-powerset-type. Since NWW are a special case of NCW, the result for NWW follows. □

**Lemma 9** *NBW are not Büchi-powerset-type.*

**Proof.** The NBW $\mathcal{A}$ in Figure 6 recognizes the language of all words with infinitely many occurrences of the subword $ab$. The language can be recognized by a DBW, yet no DBW for it can be defined on top of $\mathcal{P}(\mathcal{A})$. □

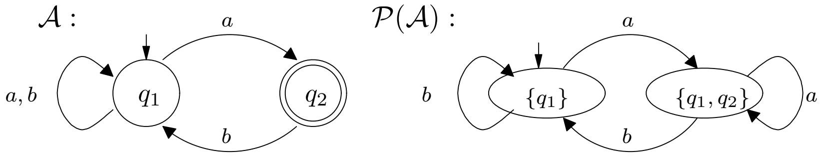


Figure 6: An NBW for $((a + b)^* \cdot a \cdot b)^\omega$ that is not Büchi-powerset-type.

**Lemma 10** *NWW are neither co-Büchi-, Rabin-, nor Streett-powerset-type.*

**Proof.** The NWW $\mathcal{A}$ in Figure 7 recognizes the language of all words with an $(a \cdot b)^\omega$ tail. The language can be recognized by a DCW, and hence also by a DRW and DSW. Yet, no DCW, DRW, or DSW for it can be defined on top of $\mathcal{P}(\mathcal{A})$. □

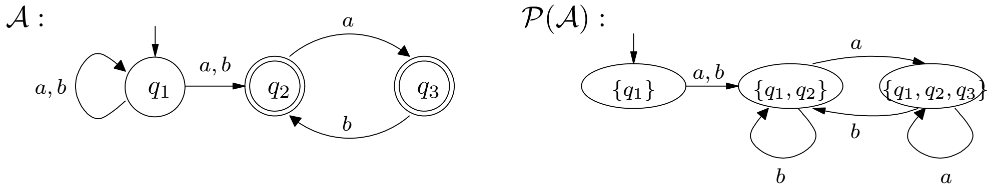


Figure 7: An NBW for $(a + b)^* \cdot (a \cdot b)^\omega$ that is neither co-Büchi-, Rabin-, nor Streett-powerset-type.

The definition of powerset-typeness requires the deterministic automaton to have the powerset structure, but it does not restrict the definition of the set of accepting states. For an automaton $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$, let $\mathcal{S}(\mathcal{A}) = \langle \Sigma, 2^Q, \delta_\mathcal{P}, \{Q_0\}, \alpha_P \rangle$ be the automaton obtained from $\mathcal{A}$ by applying to it the subset construction. Thus, the structure of $\mathcal{S}(\mathcal{A})$ is the powerset-structure of $\mathcal{A}$, and a state is in $\alpha_\mathcal{P}$ if its intersection with $\alpha$ is not empty. We refer to $\mathcal{S}(\mathcal{A})$ as the *subset automaton* of $\mathcal{A}$. Clearly, for an NFW $\mathcal{A}$, we have that $\mathcal{A}$ and $\mathcal{S}(\mathcal{A})$ are equivalent [19]. We say that an NBW $\mathcal{A}$ is Büchi-subset-type if $\mathcal{A}$ and $\mathcal{S}(\mathcal{A})$ are equivalent. Note that if $\mathcal{A}$ is Büchi-subset-type, then it is also Büchi-powerset-type. As we shall see below, the other direction does not necessarily hold.

**Lemma 11** *There is an NBW that is Büchi-powerset-type but not Büchi subset-type.*

**Proof.** The NBW $\mathcal{A}$ in Figure 8 recognizes the language of all words with infinitely many $b$'s but no two successive $b$'s. The DBW obtained by augmenting the powerset structure of $\mathcal{A}$, also described in the figure, with the acceptance condition $\alpha_\mathcal{P} = \{\{q_1\}\}$ is equivalent to $\mathcal{A}$. Thus, $\mathcal{A}$ is powerset type. On the other hand, $\mathcal{S}(\mathcal{A})$ has $\alpha_P = \{\{q_1, q_2\}\}$ and is not equivalent to $\mathcal{A}$. □

*5.1. From NBW to NFW*

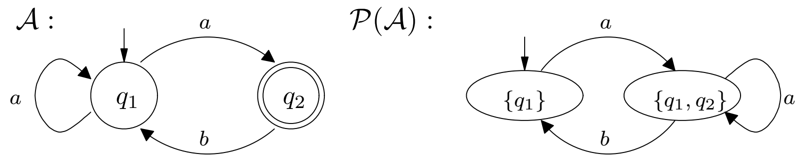


Figure 8: An NBW for $(a^+ \cdot b)^\omega$ that is Büchi-powerset-type but not finite-type.

Recall that DBW are strictly less expressive than NBW. A language $L \subseteq \Sigma^\omega$ can be recognized by a DBW iff there is a regular language $R \subseteq \Sigma^*$ such that $L = limR$; that is, $w \in L$ iff $w$ has infinitely many prefixes in $R$ [11]. An open problem is to construct, given an NBW $\mathcal{A}$ for $L$, such that $\mathcal{A}$ is DBW-realizable, an NFW $\mathcal{A}'$ for the corresponding $R$. An immediate $2^{O(n \log n)}$ upper bound follows from the $2^{O(n \log n)}$ determinization construction of [20] for $\mathcal{A}$ (since DRW are Büchi type, the DRW constructed in [20] can be converted to a DBW on the same structure). While the $2^{O(n \log n)}$ blow up in determinization is tight [15, 12], no super-linear lower bound is known for the translation of $\mathcal{A}$ to $\mathcal{A}'$. The challenges in this problem are similar to these in the problem of translating an NBW that is NCW-realizable to an equivalent NCW. While a $2^{O(n \log n)}$ upper bound is immediate, no super-linear lower bound is known.

Consider an NBW $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$. We say that $\mathcal{A}$ is *finite-type* if there is an NFW $\mathcal{A}' = \langle \Sigma, Q, \delta, Q_0, \alpha' \rangle$ such that $\mathcal{L}(\mathcal{A}) = lim\mathcal{L}(\mathcal{A}')$. Thus, $\mathcal{A}$ is finite-type if there is an NFW with the same structure as $\mathcal{A}$ (but possibly with a different set of accepting states) whose limit language is the language of $\mathcal{A}$. Let $\mathcal{A}_{fin}$ be $\mathcal{A}$ viewed as an NFW. We say that $\mathcal{A}$ is *strong-finite-type* if $\mathcal{L}(\mathcal{A}) = lim\mathcal{L}(\mathcal{A}_{fin})$. Thus, in strong-finite-typeness, we require the NFW to have both the structure and the acceptance condition of $\mathcal{A}$. Obviously, the transition from an NBW $\mathcal{A}$ that is finite-type to an NFW whose limit is $\mathcal{A}$ is linear.

The notion of subset-typeness turns out to be related to finite-typeness:

**Lemma 12** *An NBW is Büchi-subset-type iff it is strong-finite-type.*

**Proof.** Assume first that $\mathcal{A}$ is not Büchi subset-type. Since $\mathcal{S}(\mathcal{A})$ is a DBW, then $\mathcal{L}(\mathcal{S}(\mathcal{A})) = lim\mathcal{L}(\mathcal{S}(\mathcal{A})_{fin})$. Since $\mathcal{A}_{fin}$ is an NFW, then $\mathcal{L}(\mathcal{A}_{fin}) = \mathcal{L}(\mathcal{S}(\mathcal{A})_{fin})$. It follows that $\mathcal{L}(\mathcal{S}(\mathcal{A})) = lim\mathcal{L}(\mathcal{A}_{fin})$. Since $\mathcal{A}$ is not Büchi subset-type, $\mathcal{L}(\mathcal{A}) \neq \mathcal{L}(\mathcal{S}(\mathcal{A}))$. It follows that $\mathcal{L}(\mathcal{A}) \neq lim\mathcal{L}(\mathcal{A}_{fin})$, thus $\mathcal{A}$ is not strong-finite-type.

Assume now that $\mathcal{A}$ is Büchi subset-type. For every NBW $\mathcal{A}$, we have that $\mathcal{L}(\mathcal{A}) \subseteq lim\mathcal{L}(\mathcal{A}_{fin})$. Indeed, an accepting run of $\mathcal{A}$ on a word $w$ points to infinitely many prefixes of $w$ that are accepted by $\mathcal{A}_{fin}$. It is left to prove that $lim\mathcal{L}(\mathcal{A}_{fin}) \subseteq \mathcal{L}(\mathcal{A})$. Consider a word $w \in lim\mathcal{L}(\mathcal{A}_{fin})$. Thus, $w$ has infinitely many prefixes in $\mathcal{L}(\mathcal{A}_{fin})$. Since $\mathcal{A}_{fin}$ is an NFW, then $\mathcal{L}(\mathcal{A}_{fin}) = \mathcal{L}(\mathcal{S}(\mathcal{A})_{fin})$. It follows that $w$ has infinitely many prefixes in $\mathcal{L}(\mathcal{S}(\mathcal{A})_{fin})$, or equivalently, that the run of $\mathcal{S}(\mathcal{A})$ on $w$ visits the set of accepting states infinitely often, implying that $w \in \mathcal{L}(\mathcal{S}(\mathcal{A}))$. Since $\mathcal{A}$ is Büchi-subset-type, $w$ is also accepted by $\mathcal{A}$, and we are done. $\square$

It is worth noting that not all NBW that are DWW-realizable are strong-finite-type. Indeed, as proved in [13], an NBW that is DWW-realizable is also Büchi-powerset-type. On the other hand, there are DBW that are DWW-realizable and are not Büchi-subset-type, and hence also not strong-finite-type. To see this, take the NBW $\mathcal{A}_1$ described in Figure 4, add an accepting state $q$ that is reachable from $q_2$ with a transition labeled $a$, and also add a self-loop labeled $b$ from $q$ to itself. The obtained NBW still accepts $\mathcal{L}(\mathcal{A}_1)$, but the subset construction results in an automaton that accepts $\Sigma^\omega$.

Recall that both subset-typeness and strong-finite-typeness restrict the set of accepting states. We could have then hoped that the notions of powerset-typeness and finite-typeness, which both do not restrict the set of accepting states, would also be related. As we now show, this is not the case.

**Lemma 13** *Büchi-powerset-type NBW are not finite-type.*

**Proof.** As discussed in the proof of Lemma 11, the NBW $\mathcal{A}$ in Figure 8 is powerset type. On the other hand, there is no way to augment $\mathcal{A}_{fin}$ with an acceptance condition $\alpha'$ that results in an automaton $\mathcal{A}'$ for which $lim(\mathcal{L}(\mathcal{A}')) = \mathcal{L}(\mathcal{A})$. To see this, note that either $\alpha'$ is empty, in which case $\mathcal{L}(\mathcal{A}')$ is empty, or $\alpha'$ is not empty, in which case $\mathcal{L}(\mathcal{A}')$ contains $a^+$, thus $lim(\mathcal{L}(\mathcal{A}'))$ contains $a^\omega$, which is not in $\mathcal{L}(\mathcal{A})$. □

## 6. Discussion

We studied three notions of typeness for automata on infinite words. The notions are helpful in studying the complexity and complication of translations between the various classes of automata. Of special interest is the blow-up involved in a translation of NBW to NCW, when exists. As discussed in Section 4, a polynomial translation will enable an exponential translation of LTL to alternation-free $\mu$-calculus (for formulas that can be expressed in the alternation-free $\mu$-calculus), improving the doubly-exponential known upper bound. Current translations of NBW to NCW actually construct a DCW with $2^{O(n \log n)}$ states (starting with an NBW with $n$ states), whereas even no super-linear lower bound is known.

A related notion has to do with the translation of an NBW to an NFW whose limit language is equivalent to that of the NBW. We studied also this notion, and characterized NBW that are finite-type, and for which a linear translation exists. We hope to relate finite-typeness with co-Büchi typeness, aiming at developing more techniques and understanding for approaching the NBW to NCW problem.

## References


1. R. Armoni, L. Fix, A. Flaisher, R. Gerth, B. Ginsburg, T. Kanza, A. Landver, S. Mador-Haim, E. Singerman, A. Tiemeyer, M.Y. Vardi, and Y. Zbar. The ForSpec temporal logic: A new temporal property-specification logic. In *Proc. 8th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 2280 of *Lecture Notes in Computer Science*, pages 296–211, Grenoble, France, April 2002. Springer-Verlag.
2. I. Beer, S. Ben-David, C. Eisner, D. Fisman, A. Gringauze, and Y. Rodeh. The temporal logic sugar. In *Proc. 13th International Conference on Computer Aided Verification*, volume 2102 of *Lecture Notes in Computer Science*, pages 363–367, Paris, France, July 2001. Springer-Verlag.

3. B. Boigelot, S. Jodogne, and P. Wolper. On the use of weak automata for deciding linear arithmetic with integer and real variables. In *Proc. International Joint Conference on Automated Reasoning (IJCAR)*, volume 2083 of *Lecture Notes in Computer Science*, pages 611–625, Siena, June 2001. Springer-Verlag.

4. J.R. Büchi. On a decision method in restricted second order arithmetic. In *Proc. Internat. Congr. Logic, Method. and Philos. Sci. 1960*, pages 1–12, Stanford, 1962. Stanford University Press.

5. M. Kaminski. A classification of $\omega$-regular languages. *Theoretical Computer Science*, 36:217–229, 1985.

6. S.C. Krishnan, A. Puri, and R.K. Brayton. Deterministic $\omega$-automata vis-a-vis deterministic Büchi automata. In *Algorithms and Computations*, volume 834 of *Lecture Notes in Computer Science*, pages 378–386. Springer-Verlag, 1994.

7. O. Kupferman and M.Y. Vardi. Relating linear and branching model checking. In *IFIP Working Conference on Programming Concepts and Methods*, pages 304 – 326, New York, June 1998. Chapman & Hall.

8. O. Kupferman and M.Y. Vardi. From linear time to branching time. *ACM Trans. on Computational Logic*, 6(2):273–294, 2005.

9. O. Kupferman, M.Y. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the ACM*, 47(2):312–360, March 2000.

10. R.P. Kurshan. *Computer Aided Verification of Coordinating Processes*. Princeton Univ. Press, 1994.

11. L.H. Landweber. Decision problems for $\omega$–automata. *Mathematical Systems Theory*, 3:376–384, 1969.

12. C. Löding. Optimal bounds for the transformation of omega-automata. In *Proc. 19th Conference on the Foundations of Software Technology and Theoretical Computer Science*, volume 1738 of *Lecture Notes in Computer Science*, pages 97–109, December 1999.

13. O. Maler and L. Staiger. On syntactic congruences for $\omega$-languages. *Theoretical Computer Science*, 183(1):93–112, 1997.

14. R. McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, 9:521–530, 1966.

15. M. Michel. Complementation is more difficult with automata on infinite words. CNET, Paris, 1988.

16. S. Miyano and T. Hayashi. Alternating finite automata on $\omega$-words. *Theoretical Computer Science*, 32:321–330, 1984.

17. D.E. Muller, A. Saoudi, and P.E. Schupp. Alternating automata, the weak monadic theory of the tree and its complexity. In *Proc. 13th International Colloquium on Automata, Languages and Programming*, volume 226 of *Lecture Notes in Computer Science*. Springer-Verlag, 1986.

18. M.O. Rabin. Decidability of second order theories and automata on infinite trees. *Transaction of the AMS*, 141:1–35, 1969.

19. M.O. Rabin and D. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3:115–125, 1959.

20. S. Safra. On the complexity of $\omega$-automata. In *Proc. 29th IEEE Symp. on Foundations of Computer Science*, pages 319–327, White Plains, October 1988.

21. S. Safra and M.Y. Vardi. On $\omega$-automata and temporal logic. In *Proc. 21st ACM Symp. on Theory of Computing*, pages 127–137, Seattle, May 1989.

22. W. Thomas. Automata on infinite objects. *Handbook of Theoretical Computer Science*, pages 165–191, 1990.

23. M.Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. In *Proc. 1st Symp. on Logic in Computer Science*, pages 332–344, Cambridge, June 1986.

24. M.Y. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, November 1994.