Alternating Automata and Logics over Infinite Words (Extended Abstract)

Christof Löding, Wolfgang Thomas

Lehrstuhl für Informatik VII, RWTH Aachen, D-52056 Aachen
{loeding,thomas}@informatik.rwth-aachen.de

Abstract. We give a uniform treatment of the logical properties of alternating weak automata on infinite strings, extending and refining work of Muller, Saoudi, and Schupp (1984) and Kupferman and Vardi (1997). Two ideas are essential in the present set-up: There is no acyclicity requirement on the transition structure of weak alternating automata, and acceptance is defined only in terms of reachability of states; moreover, the run trees of the standard framework are replaced by run dags of bounded width. As applications, one obtains a new normal form for monadic second order logic, a simple complementation proof for weak alternating automata, and elegant connections to temporal logic.

1 Introduction

Finite automata on infinite strings provide a useful framework for the logical analysis of sequence properties. The connection to logic is based on (at least) the following four aspects:

- Nondeterministic Büchi automata are expressively equivalent to monadic second-order logic (MSO-logic) over infinite strings ([Büc62]). This equivalence involves a normal form of MSO-formulas in EMSO-logic (existential monadic second-order logic).
- Connected with this fact is the closure of Büchi automata under complement.
- A hierarchy of acceptance conditions for deterministic ω-automata induces a natural classification of sequence properties (cf. [MP92]), including, for example, safety properties and recurrence properties.
- Propositional temporal logic PLTL, a standard framework for the specification of infinite computations, is characterized by counter-free deterministic Muller automata, defined by a natural restriction on the loop structure of transition graphs.

In the present paper, we study these logical aspects of ω-languages in the framework of alternating weak automata, a model introduced in the pioneering work of Muller, Saoudi, and Schupp [MSS86]. We introduce a variant of alternating weak automata which differs from the model of [MSS86] in the following way:

There is no acyclicity requirement on the transition structure of weak alternating automata, and acceptance is defined only in terms of reachability of states; moreover, the run trees of the standard framework are replaced by run dags of bounded width. (For the equivalence to the model of [MSS86] see the next section.) Starting from this, it turns out that all four aspects mentioned above have counterparts in the framework of alternating automata:

1. The equivalence between alternating weak automata and monadic second-order logic over infinite strings provides a new normal form of MSO-formulas, giving an alternative to the classical EMSO-normal form (for the specification of accepting runs).
2. The complementation of alternating weak automata is presented in a game theoretic setting, based on a determinacy result on infinite games with winning conditions in terms of reachability of states. (For a separate exposition of this result see [Tho99].)
3. The basic classification of temporal properties (called Landweber hierarchy in the automaton theoretic setting) is captured in the framework of alternating automata in two different ways: by restricting the alternating computation mode (to universal, respectively existential branching) and by restricting the acceptance component in alternating automata.
4. A structural property (on the loop structure) of weak alternating automata is presented, which characterizes the properties definable in the temporal logic PLTL.

As mentioned above, we define acceptance by alternating automata using run dags instead of run trees. In [KV97], alternating automata are defined as in [MSS86] (however with the co-Büchi acceptance condition), and then a reduction to acceptance via run dags is carried out. In both cases, the approach via run dags does not weaken the expressive power due to the fact that in the associated infinite games (see Section 3) memoryless winning strategies are sufficient. The determinacy proof presented in this paper (of which a preliminary exposition was given by the second author in [Tho99]) is related to a construction of Klarlund [Kla91]. The structural characterization of PLTL-definable properties was obtained independently by Rohde [Roh97], however with a more involved proof.

The paper is structured as follows: In Section 2 we introduce alternating automata and their acceptance conditions. In Section 3, the dualization of alternating automata and its connection to determinacy of infinite games and to complementation is developed (see item 2 above). Section 4 shows that alternating weak automata are able to recognize precisely the regular ω-languages, via a transformation of parity automata into alternating weak automata. Finally, Sections 5, 6, and 7 present the results mentioned above under item 1 (connection to MSO-logic), item 3 (classification of sequence properties) and item 4 (characterization of PLTL-definable properties).

2 Alternating Automata

Alternating automata combine the possibility of existential and universal branching. The transition function of an alternating automaton is defined with positive boolean formulas over the state set.

Let X be a finite set. The set of positive boolean formulas over X, denoted by $\mathcal{B}^+(X)$, contains $\top$ (true), $\bot$ (false), all elements from X, and all boolean combinations over X built with $\wedge$ and $\vee$. A subset S of X is a *model* of $\theta \in \mathcal{B}^+(X)$ iff the truth assignment that assigns true to the elements of S and false to the elements of $X \setminus S$ satisfies θ. We say S is a *minimal model* of θ iff S is a model of θ and no proper subset of S is a model of θ. For $\theta \in \mathcal{B}^+(X)$ the set of minimal models of θ is denoted by $\mathcal{M}_\theta$.

An alternating automaton $\mathcal{A}$ is of the form $\mathcal{A} = (Q, \Sigma, q_0, \delta, AC)$, where Q is a finite state set, Σ is a finite alphabet, $q_0 \in Q$ is the initial state, $\delta : Q \times \Sigma \to \mathcal{B}^+(Q)$ is the transition function and AC is the acceptance component. There are several different types of acceptance conditions referring to different types of acceptance components.

Since in an alternating automaton there is universal branching, a run of an alternating automaton is not an infinite sequence of states, but an infinite acyclic graph. This graph has a "root vertex" labelled with the initial state q_0 and in distance l from this vertex one finds the states which are assumed by the automaton after l input letters. Formally a run is defined as follows. Let $\alpha \in \Sigma^\omega$ and let $G = (V, E)$ be a directed acyclic graph with the following properties.

- $V \subseteq Q \times \mathbb{N}$ with $(q_0, 0) \in V$.
- $E \subseteq \bigcup\limits_{l \geq 0} (Q \times \{l\}) \times (Q \times \{l+1\})$.
- For every $(q, l) \in V \setminus \{(q_0, 0)\}$ exists a $q' \in Q$, such that $((q', l-1), (q, l)) \in E$.

G is called a *run* of $\mathcal{A} = (Q, \Sigma, q_0, \delta, A)$ on α, if for every $(q, l) \in V$ the set $\{q' \in Q \mid ((q, l), (q', l+1)) \in E\}$ is a minimal model of $\delta(q, \alpha(l))$. An example is given in Figure 1.

Note that there is no run $G = (V, E)$ on α, such that $(q, l) \in V$ for a q with $\delta(q, \alpha(l)) = \bot$, since $\bot$ has no models.

$Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{a\}$

$\delta(q_0, a) = q_1 \wedge q_2$
$\delta(q_1, a) = (q_1 \wedge q_3) \vee (q_2 \wedge q_3)$
$\delta(q_2, a) = q_1$
$\delta(q_3, a) = (q_1 \wedge q_2) \vee q_3$

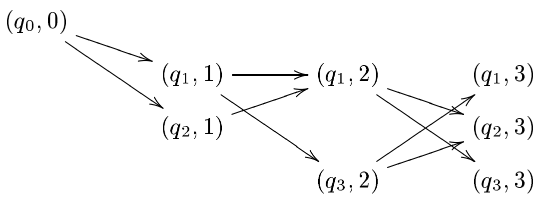

Fig. 1. First segment of a run of an alternating automaton.

With this definition of the transition function of alternating automata we get deterministic, nondeterministic, and universal automata as special cases of alternating automata. In deterministic automata the formulas used in the transition function consist of exactly one state. In nondeterministic automata the formulas are disjunctions of states or $\bot$, and dual to that, in universal automata the formulas are conjunctions of states or $\top$.

For later use in Section 7, we note that it is possible to use alternating automata with an initial positive boolean formula θ_0 instead of a single initial state. Such an automaton can be converted into an equivalent automaton with a single initial state just by adding one extra state.

An automaton accepts a word iff there exists a run of the automaton on this word such that every infinite path through that run satisfies the acceptance condition. The language accepted by the automaton consists of all the words that are accepted by the automaton. Here we identify an infinite path π with the sequence of states induced by this path. The *infinity set* $In(\pi)$ consists of all states that appear infinitely often in π. The *occurrence set* $Oc(\pi)$ consists of all states that appear at least once in π. The following different types of acceptance conditions are considered in this paper.

In *Büchi* and *co-Büchi automata* the acceptance condition refers to a subset F of the state set and in *parity automata* the acceptance condition refers to a mapping (called coloring) $c : Q \to \{0, \ldots, k\}$. The numbers $0, \ldots, k$ are called the colors. For an infinite path π the corresponding infinite sequence of colors is then denoted as $c(\pi)$. An infinite path π satisfies

- the Büchi condition w.r.t. F iff $F \cap In(\pi) \neq \emptyset$,
- the co-Büchi condition w.r.t. F iff $F \cap In(\pi) = \emptyset$,
- the parity condition w.r.t. c iff $\min(In(c(\pi)))$ is even.

For all of these acceptance types we can also consider the "weak versions". We call an acceptance condition weak if it is evaluated in the occurrence set instead of the infinity set of a path. So an infinite path π satisfies

- the weak Büchi condition w.r.t. F iff $F \cap Oc(\pi) \neq \emptyset$,
- the weak co-Büchi condition w.r.t. F iff $F \cap Oc(\pi) = \emptyset$,
- the weak parity condition w.r.t. c iff $\min(Oc(c(\pi)))$ is even.

In the present paper, we focus on weak automata with the weak parity acceptance condition. This differs from the model of weak automata as introduced by Muller and Schupp in [MSS86] where the Büchi acceptance condition is used. Moreover the transition structure of a weak automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ must fulfill the following requirement: There is a partition $Q_1, \ldots, Q_m$ of Q such that for every $q \in Q_i$ and $q' \in Q_j$ with a transition leading from q to q' one has $j \leq i$, and $Q_i \subseteq F$ or $Q_i \cap F = \emptyset$ for every $i \in \{1, \ldots, m\}$.

Let us verify that the two models are equivalent in expressive power. Given a weak automaton $\mathcal{A}$ as above (in the sense of [MSS86]), acceptance means that for every path π through a run of $\mathcal{A}$ there is an $i \in \{1, \ldots, m\}$ such that $In(\pi) \subseteq Q_i \subseteq F$. This can also be expressed as a weak parity condition because a path through the run satisfies the acceptance condition if it enters one of the accepting Q_i's and never leaves it again. For all $i \in \{1, \ldots, m\}$ and for all $q \in Q_i$ we define the coloring c as

$$c(q) = \begin{cases} 2i & \text{if } Q_i \subseteq F, \\ 2i+1 & \text{if } Q_i \cap F = \emptyset. \end{cases}$$

It is easy to see that the weak parity automaton $\mathcal{A}' = (Q, \Sigma, q_0, \delta, c)$ is equivalent to $\mathcal{A}$.

Conversely, given a weak parity automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta, c)$ with $c : Q \to C$, let C_{e} be the set of even numbers in C and let $Q' = Q \times C$, $q_0' = (q_0, c(q_0))$, $F' = Q \times C_{\mathrm{e}}$. To define the transition function we need an auxiliary mapping $\phi : \mathcal{B}^+(Q) \times C \to \mathcal{B}^+(Q')$. The formula $\phi(\theta, i)$ is obtained by replacing every $q \in Q$ in θ with $(q, \min\{i, c(q)\})$. Then we define the transition function by $\delta'((q, i), a) = \phi(\delta(q, a), i)$. In the second component of the new states the automaton remembers the minimal color that was seen so far. Along a path through a run this color may not increase. Therefore, if we define the sets Q_i according to the color in the second component, we get an automaton in the form of [MSS86] equivalent to $\mathcal{A}$.

3 Complementation

In [MS87] Muller and Schupp show that complementation of alternating automata can be done by dualization. In this section we give a self-contained proof of this complementation theorem for the case of alternating weak parity automata. This is done in a game theoretic framework, and making use of the simple winning conditions which are derived from weak alternating automata. So we do *not* rely on difficult determinacy results, e.g. for Borel games, as done in [MS87]. Before we turn to games we define the dual of an alternating weak parity automaton.

For a finite set X and $\theta \in \mathcal{B}^+(X)$ the formula $\widetilde{\theta}$, the dual of θ, is obtained by exchanging $\vee$ and $\wedge$, and $\bot$ and $\top$. We can state the following relation between the minimal models of θ and the models of $\widetilde{\theta}$.

Remark 1. Let $\theta \in \mathcal{B}^+(X)$. A set $S \subseteq X$ is a model of $\widetilde{\theta}$ iff $S \cap R \neq \emptyset$ for all minimal models R of θ.

Proof. The formula $\widetilde{\theta}$ is equivalent to $\bigwedge_{R \in \mathcal{M}_\theta} \bigvee_{x \in R} x$, which is the conjunctive normal form of $\widetilde{\theta}$. S is a model of $\widetilde{\theta}$ iff it contains at least one element from each of the disjunctive terms.

Let $\mathcal{A} = (Q, \Sigma, q_0, \delta, c)$ be an alternating weak parity automaton. The *dual automaton* $\widetilde{\mathcal{A}}$ of $\mathcal{A}$ is defined as $\widetilde{\mathcal{A}} = (Q, \Sigma, q_0, \widetilde{\delta}, \widetilde{c})$, where $\widetilde{\delta}$ is defined by $\widetilde{\delta}(q, a) = \widetilde{\delta(q, a)}$ for all $q \in Q$ and $a \in \Sigma$, and $\widetilde{c}$ is defined by $\widetilde{c}(q) = c(q) + 1$.

Since in $\widetilde{\mathcal{A}}$ a state has an even color iff it has an odd color in $\mathcal{A}$, we get the following remark.

Remark 2. A path π satisfies the acceptance condition of $\mathcal{A}$ iff it does not satisfy the acceptance condition of $\widetilde{\mathcal{A}}$.

Let $\mathcal{A} = (Q, \Sigma, q_0, \delta, c)$ be an alternating weak parity automaton and let $\alpha \in \Sigma^\omega$. With $\mathcal{A}$ and α we associate a graph $G_{\mathcal{A},\alpha} = (V_A, V_P, E, w)$ which serves as a game arena for the two players Automaton (A) and Pathfinder (P). The graph is defined as follows.

- $V_A = Q \times \mathbb{N}$ and $V_P = Q \times (2^Q \setminus \{\emptyset\}) \times \mathbb{N}$; let V always denote $V_A \cup V_P$.
- The edge relation E is defined by

$$\begin{array}{ll} ((q,l),(q,S,l+1)) \in E & \text{iff} \ \ S \in \mathcal{M}_{\delta(q,\alpha(l))}, \\ ((p,S,l),(q,l)) \in E & \text{iff} \ \ q \in S, \end{array}$$

  for $p, q \in Q$, $S \subseteq Q$ and $l \in \mathbb{N}$.
- The coloring $w : V \to c(Q)$ is given by $w((q,l)) = c(q)$ for $(q,l) \in V_A$ and $w((q,S,l)) = c(q)$ for $(q,S,l) \in V_P$.

A play of $G_{\mathcal{A},\alpha}$ is an infinite sequence $\gamma \in (V_A V_P)^\omega$ such that $\gamma(0) = (q_0, 0)$ and $(\gamma(i), \gamma(i+1)) \in E$ for all $i \in \mathbb{N}$. Automaton wins the play γ iff $\min(Oc(w(\gamma)))$ is even.

A positional strategy for A is a mapping $f_A : V_A \to V_P$ such that for all $v \in V_A$ we have $(v, f_A(v)) \in E$. The play γ is played according to f_A iff for every $i \in \mathbb{N}$ with $\gamma(i) \in V_A$ one has $\gamma(i+1) = f_A(\gamma(i))$. The strategy f_A is called a positional winning strategy for A iff A wins every play γ played according to f_A.

Strategies for P are defined analogously.

The connections between alternating weak parity automata and the corresponding games are stated in the following three lemmas.

Lemma 1. *Let $\mathcal{A} = (Q, \Sigma, q_0, \delta, c)$ be an alternating weak parity automaton and let $\alpha \in \Sigma^\omega$. Automaton has a positional winning strategy in $G_{\mathcal{A},\alpha}$ iff $\alpha \in L(\mathcal{A})$.*

The very simple proof is omitted.

Lemma 2. *(Determinacy of weak parity games) Let $\mathcal{A} = (Q, \Sigma, q_0, \delta, c)$ be an alternating weak parity automaton and let $\alpha \in \Sigma^\omega$. In $G_{\mathcal{A},\alpha}$ either Automaton or Pathfinder has a positional winning strategy.*

Proof. We first define the notion of an attractor. Let $T \subseteq V_A \cup V_P$. The A-attractor of T, denoted $Attr_A(T)$, is the set of vertices from which Automaton can force the play to eventually visit T.

$$\begin{array}{rl} Attr_A(T) = & \bigcup_{i \in \mathbb{N}} Attr_A^i(T), \text{ where } Attr_A^0(T) = T \text{ and} \\ v \in Attr_A^{i+1}(T) \Leftrightarrow & v \in Attr_A^i(T) \text{ or} \\ & v \in V_A \text{ and } \exists (v,u) \in E \ : \ w \in Attr_A^i(T) \text{ or} \\ & v \in V_P \text{ and } \forall (v,u) \in E \ : \ w \in Attr_A^i(T). \end{array}$$

The P-attractor of T, denoted $Attr_P(T)$, is defined in the analogous way.

By induction on $m = |c(Q)|$, i.e., the number of colors in the weak automaton, we show that either of the players has a positional winning strategy. For $m = 1$ every play in $G_{\mathcal{A},\alpha}$ is won by Automaton or every play is won by Pathfinder. Therefore Automaton or Pathfinder has a positional winning strategy.

Let $m \geq 2$, $k = \min(w(V))$, and $V_k = \{v \in V \mid w(v) = k\}$. We assume that k is even. The proof for the other case is analogous.

If $(q_0, 0)$ belongs to $Attr_A(V_k)$, then obviously Automaton has a positional winning strategy in $G_{\mathcal{A},\alpha}$. If $(q_0, 0)$ does not belong to $Attr_A(V_k)$, then we define the game $G'_{\mathcal{A},\alpha}$ by removing the vertices of $Attr_A(V_k)$ from $G_{\mathcal{A},\alpha}$. By induction we know that in $G'_{\mathcal{A},\alpha}$ either Automaton or Pathfinder has a positional winning strategy. If Pathfinder has a winning strategy in $G'_{\mathcal{A},\alpha}$, then playing according to this strategy also forces the game to stay outside of $Attr_A(V_k)$ in the game $G_{\mathcal{A},\alpha}$. Otherwise there would be a vertex belonging to $Attr_A(V_k)$ in $G'_{\mathcal{A},\alpha}$. Therefore Pathfinder also has a positional winning strategy in $G_{\mathcal{A},\alpha}$.

Now suppose Automaton has a positional winning strategy in $G'_{\mathcal{A},\alpha}$. If Automaton plays according to this strategy in $G_{\mathcal{A},\alpha}$, then the only possibility for Pathfinder to give the play another progression as in $G'_{\mathcal{A},\alpha}$, is to move into $Attr_A(V_k)$ if possible. But then Automaton wins by forcing the game to move into V_k. Therefore Automaton has a positional winning strategy in $G_{\mathcal{A},\alpha}$.

Lemma 3. *Let $\mathcal{A} = (Q, \Sigma, q_0, \delta, c)$ be an alternating weak parity automaton and let $\alpha \in \Sigma^\omega$. Automaton has a positional winning strategy in $G_{\mathcal{A},\alpha}$ iff Pathfinder has a positional winning strategy in $G_{\widetilde{\mathcal{A}},\alpha}$.*

Proof. Let f_A be a positional winning strategy for Automaton in $G_{\mathcal{A},\alpha}$ and let (q, S, l) be a vertex of Pathfinder in $G_{\widetilde{\mathcal{A}},\alpha}$. If there exists a play such that (q, S, l) appears in this play, then $S \in \mathcal{M}_{\widetilde{\delta}(q,\alpha(l-1))}$. For vertices that may not appear in a play we do not have to define the strategy. From Remark 1 it follows that $S \cap f_A(q, l-1) \neq \emptyset$. Let $p \in S \cap f_A(q, l-1)$. We define $\widetilde{f}_P(q, S, l) = p$. For a play γ of $G_{\widetilde{\mathcal{A}},\alpha}$ played according to $\widetilde{f}_P$ there exists a play γ' of $G_{\mathcal{A},\alpha}$ played according to f_A such that $\widetilde{w}(\gamma) = w(\gamma')$. Since Automaton wins γ' in $G_{\mathcal{A},\alpha}$, Pathfinder wins γ in $G_{\widetilde{\mathcal{A}},\alpha}$.

Let $\widetilde{f}_P$ be a positional winning strategy for Pathfinder in $G_{\widetilde{\mathcal{A}},\alpha}$ and let (q, l) be a vertex of Automaton in $G_{\mathcal{A},\alpha}$. The set $S = \{\widetilde{f}_P(q, R, l) \mid R \in \mathcal{M}_{\widetilde{\delta}(q,\alpha(l))}\}$ is a model of $\delta(q, \alpha(l))$ by Remark 1. Let $S' \subset S$ be a minimal model of $\delta(q, \alpha(l))$. We define $f_A(q, l) = (q, S', l+1)$. Again, for a play γ in $G_{\mathcal{A},\alpha}$ played according to f_A there exists a play γ' played according to $\widetilde{f}_P$ in $G_{\widetilde{\mathcal{A}},\alpha}$ such that $w(\gamma) = \widetilde{w}(\gamma')$. Since Pathfinder wins γ' in $G_{\widetilde{\mathcal{A}},\alpha}$, Automaton wins γ in $G_{\mathcal{A},\alpha}$.

Theorem 1. *Let $\mathcal{A}$ be a alternating weak parity automaton over the alphabet Σ. Then $L(\widetilde{\mathcal{A}}) = \Sigma^\omega \setminus L(\mathcal{A})$.*

Proof. Let $\alpha \in \Sigma^\omega$. The automaton $\mathcal{A}$ accepts α iff Automaton has a positional winning strategy in $G_{\mathcal{A},\alpha}$, by Lemma 1. By Lemma 3 this is equivalent to Pathfinder having a winning strategy in $G_{\widetilde{\mathcal{A}},\alpha}$. With Lemma 2 we know that this is the same as Automaton having no positional winning strategy in $G_{\widetilde{\mathcal{A}},\alpha}$ and again using Lemma 1 this is equivalent to $\alpha \notin L(\widetilde{\mathcal{A}})$.

4 Expressive Completeness

In this section we show that every regular ω-language can be recognized by an alternating weak parity automaton. We give a transformation of deterministic parity automata into alternating weak parity automata; it seems to be the simplest way to establish expressive completeness of alternating weak parity automata. (It is well known that deterministic parity automata recognize precisely the regular ω-languages, see [Tho97]). In contrast to deterministic parity automata, where it is not possible to bound the number of colors, for alternating weak parity automata it suffices to consider automata with only three colors.

Theorem 2. *For every deterministic parity automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta, c)$ with $|Q| = n$ and $c : Q \to \{1, \ldots, m\}$ one can construct an equivalent alternating weak parity automaton $\mathcal{A}' = (Q', \Sigma, q_0', \delta', c')$ with $|Q'| = (m+1)n$ and $c' : Q' \to \{1, 2, 3\}$.*

Proof. We can assume that $m = 2k$ for some k. Define $\mathcal{A}'$ as follows.

- Let $Q' = Q \cup (Q \times \{1, \ldots, k\} \times \{0, 1\})$ and $q_0' = q_0$.
- For $q \in Q$, $i \in \{1, \ldots, k\}$, $a \in \Sigma$, and $p = \delta(q, a)$ define

$$\begin{aligned}
\delta'(q, a) &= p \vee \bigvee_{j=1}^{k} (p, j, 0),\\
\delta'((q, i, 0), a) &= \begin{cases} \bot \text{ if } c(q) < 2i,\\ (p, i, 0) \wedge (p, i, 1) \text{ otherwise,} \end{cases}\\
\delta'((q, i, 1), a) &= \begin{cases} \top \text{ if } c(q) = 2i,\\ (p, i, 1) \text{ otherwise.} \end{cases}
\end{aligned}$$

- For $q \in Q$ and $i \in \{1, \ldots, k\}$ let $c'(q) = 3$, $c'((q, i, 0)) = 2$, $c'((q, i, 1)) = 1$.

The idea is to guess the accepting color and the point from where on no smaller color occurs, and then to verify that the guessed color occurs infinitely often and no smaller color occurs anymore. The correctness proof is omitted.

Note that along each path through a run of $\mathcal{A}'$ the colors are decreasing, which corresponds to the usual definition of weak automata. In fact our model without this restriction is equivalent to the usual model.

It is also possible to start from nondeterministic Büchi automata instead of deterministic parity automata, using a similar construction as in [KV97].

5 Alternating Automata and MSO

In this section we give a characterization of alternating weak parity automata in MSO logic. We obtain a normal form of MSO formulas different from the EMSO formulas obtained by Büchi [Büc62].

The MSO formulas we consider are S1S (second-order theory of one successor) formulas, built up in the usual way from $x, y, z, x_1, x_2, \ldots$ as (first-order) variables for natural numbers, $X, Y, Z, X_1, X_2, \ldots$ as (second-order) variables for sets of natural numbers, the symbols 0, $+1$, $=$, $<$, $\in$ with their usual meanings, the connectives $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, and the quantifiers $\exists, \forall$.

We can interpret the sets of natural numbers as predicates, and abbreviate $x \in X$ with Xx. We also use natural abbreviations as $\leq, >, \geq, +2, +3, \ldots$ and $\exists^\omega$ ("there exist infinitely many"), $\forall^\omega$ ("for almost all").

The formulas are interpreted in the structure $(\mathbb{N}, <, +1, 0)$. As models we take tuples of subsets of $\mathbb{N}$. Such a tuple $\underline{\alpha} = (Q_1, \ldots Q_n)$, with $Q_1, \ldots, Q_n \subseteq \mathbb{N}$, is a model of a formula $\phi(X_1, \ldots X_n)$, denoted by $\underline{\alpha} \models \phi(X_1, \ldots X_n)$, if and only if ϕ evaluates to true when we substitute each X_i by Q_i.

To characterize automata in S1S, we use a correspondence between infinite words α over an alphabet and the models $\underline{\alpha}$ of S1S. Then we will construct a formula $\phi_{\mathcal{A}}$ such that the automaton $\mathcal{A}$ accepts a word α if and only if the corresponding tuple $\underline{\alpha}$ is a model of $\phi_{\mathcal{A}}$. In addition we construct a second formula $\widetilde{\phi}_{\mathcal{A}}$ equivalent to $\phi_{\mathcal{A}}$, such that $\neg\widetilde{\phi}_{\mathcal{A}} = \phi_{\widetilde{\mathcal{A}}}$. The connection between these two formulas corresponds to the connection between an automaton and its dual.

To code ω-words by sets of natural numbers, we assume without loss of generality that $\Sigma = \{0,1\}^k$. With this convention a word $\alpha \in \Sigma^\omega$ consists of k words $\alpha_1, \ldots, \alpha_k \in \{0,1\}^\omega$. We code α_i with the set X_i, where $x \in X_i$ iff $\alpha_i(x) = 1$. Then the tuple $\underline{\alpha} = (X_1, \ldots, X_k)$ is a unique coding of α. We will refer to natural numbers as "positions". Now we can express the fact that a word has a certain letter $a = (a_1, \ldots, a_k)$ at the position x by the formula $\mathrm{POS}_a(x, X_1, \ldots, X_k) = \bigwedge_{i \in O(a)} X_i x \wedge \bigwedge_{i \in Z(a)} \neg X_i x$, where $O(a) = \{i | a_i = 1\}$ and $Z(a) = \{i | a_i = 0\}$.

Let $\mathcal{A} = (Q, \Sigma, q_0, \delta, c)$ be an alternating weak parity automaton with $c : Q \to \{0, \ldots, k\}$ and without $\top$ and $\bot$ in the transition function (this can be obtained by adding at most one extra state). We have to code the runs of an automaton on a word with subsets of $\mathbb{N}$. Let $Q = \{1, \ldots, n\}$ with $q_0 = 1$ and let $m = n + n^2$. We code a level l of a run and the edges to the previous level with a vector $v_l \in \{0,1\}^m$. The first n entries code the active states. This means entry i is 1 iff (i, l) is a vertex of the run. For every $i \in \{1, \ldots n\}$ the entries from $i \cdot n + 1$ to $i \cdot n + n$ code the successors of the vertex $(i, l-1)$. This means entry $i \cdot n + j$ is 1 iff (j, l) is a successor of $(i, l-1)$. This idea is illustrated in Figure 2 for the beginning of a run of an automaton with states $\{q_0, q_1, q_2, q_3\}$.

This coding yields an infinite sequence $v_0, v_1, \ldots$ of vectors from $\{0,1\}^m$ which can also be represented by $Y_1, \ldots, Y_m \subseteq \mathbb{N}$ in the same way as the words

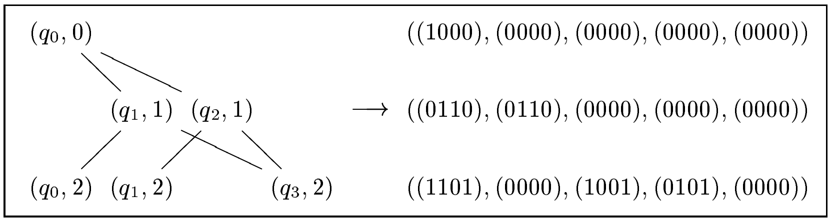

Fig. 2. Coding of the beginning of a run

from Σ^ω. It is easy to verify that the first-order formula

$$\begin{aligned}
&\mathrm{RUN}_{\mathcal{A}}(X_1,\dots,X_k,Y_1,\dots,Y_m) = (Y_1 0 \wedge \bigwedge_{i=2}^{m} \neg Y_i 0)\\
&\wedge \forall x \Big[\bigwedge_{i=1}^{n} (x>0 \wedge Y_i x \to (\bigvee_{j=1}^{n} Y_{j\cdot n+i} x)) \ \wedge \ \bigwedge_{i=1}^{n} (\neg Y_i x \to (\bigwedge_{j=1}^{n} \neg Y_{i\cdot n+j} x+1))\\
&\wedge \bigwedge_{(i,a)\in Q\times\Sigma} (Y_i x \wedge \mathrm{POS}_a(x,X_1,\dots,X_k)\\
&\qquad\qquad \to \bigvee_{S\in\mathcal{M}_{\delta(i,a)}} (\bigwedge_{j\in S} Y_j x+1 \bigwedge_{j\in S} Y_{i\cdot n+j} x+1 \wedge \bigwedge_{j\notin S} \neg Y_{i\cdot n+j} x+1))\Big]
\end{aligned}$$

is satisfied iff $Y_1,\dots,Y_m$ code a run of $\mathcal{A}$ on the input coded by $X_1,\dots,X_k$. In the same way we can construct a formula $\mathrm{DUALRUN}_{\mathcal{A}}(X_1,\dots,X_k,Y_1,\dots,Y_m)$ that defines the runs of the dual automaton of $\mathcal{A}$.

To express the acceptance of a word by an automaton we now have to code the paths of a run. A path through a run $G=(V,E)$ can be viewed as a subgraph $G'=(V',E')$ of G ($V'\subseteq V$ and $E'\subseteq E$) such that every vertex has exactly one successor and every vertex except the initial vertex has exactly one predecessor. We can code a path by $Z_1,\dots Z_n \subseteq \mathbb{N}$. These Z_i must have the following properties (with i and j ranging over $\{1,\dots,n\}$):

- $Z_i \subseteq Y_i$ for every i ($V'\subseteq V$),
- $Z_i \cap Z_j = \emptyset$ for every $i \neq j$ (in every level is at most one vertex),
- $\forall x \exists i (x \in Z_i)$ (in every level is at least one vertex),
- for all x with $Z_i x$ and $Z_j x+1$ one has $Y_{i\cdot n+j} x+1$ (the vertex in level $x+1$ is a successor of the vertex in level x),

and we indicate by $\mathrm{PATH}_{\mathcal{A}}(Y_1,\dots,Y_m,Z_1,\dots,Z_n)$ a first-order formula expressing this.

The last fact we have to express is that a path satisfies the acceptance condition. We define for $m \in \{0,\dots,k\}$ the set Q_m of states with color m, i.e., $Q_m = \{q \in Q \mid c(q)=m\}$. Let $Z_1,\dots,Z_n \subseteq \mathbb{N}$ be the coding of a path through a run. Then the first-order formula

$$\mathrm{WEAKACC}_{\mathcal{A}}(Z_1,\dots,Z_n) = \bigvee_{m=0}^{k} \Big(\exists x (\bigvee_{i\in Q_m} Z_i x) \wedge \forall x (\bigwedge_{i\in \bigcup_{l<m} Q_l} \neg Z_i x) \Big)$$

is satisfied if and only if the path coded by $Z_1, \ldots, Z_n$ satisfies the acceptance condition of $\mathcal{A}$.

Now we can translate an automaton into two equivalent S1S formulas. Here we write $\overline{X}$ for $X_1, \ldots, X_k$ (and similar $\overline{Y}$ and $\overline{Z}$).

Theorem 3. *The following two formulas are satisfied if and only if $\overline{X}$ is the coding of a word $\alpha \in L(\mathcal{A})$.*

$$\phi_{\mathcal{A}}(\overline{X}) = \exists \overline{Y} \forall \overline{Z} (\mathrm{RUN}_{\mathcal{A}}(\overline{X}, \overline{Y}) \wedge \ (\mathrm{PATH}_{\mathcal{A}}(\overline{Y}, \overline{Z}) \rightarrow \ \mathrm{WEAKACC}_{\mathcal{A}}(\overline{Z}))).$$

$$\widetilde{\phi}_{\mathcal{A}}(\overline{X}) = \forall \overline{Y} \exists \overline{Z} (\mathrm{DUALRUN}_{\mathcal{A}}(\overline{X}, \overline{Y}) \rightarrow \ (\mathrm{PATH}_{\mathcal{A}}(\overline{Y}, \overline{Z}) \wedge \ \mathrm{WEAKACC}_{\mathcal{A}}(\overline{Z}))).$$

The formulas $\phi_{\mathcal{A}}$ and $\widetilde{\phi}_{\mathcal{A}}$ thus represent a normal form for S1S-formulas of second-order quantifier prefix types Σ_2, Π_2, respectively, in which the acceptance component WEAKACC only involves reachability conditions.

6 The Landweber Hierarchy

The classical characterization of regular sequence properties in the Landweber hierarchy [Lan69] uses deterministic automata with different acceptance conditions. As we show, the hierarchy can also be characterized in two different ways: first by alternating automata, equipped with weak acceptance conditions, and second by fixing the acceptance condition as weak parity, and modifying the mode of the transition function. The three different characterizations are shown in Figure 3. For notational simplicity we abbreviate the type of an automaton by the initial letters of its transition mode and its type of acceptance condition. So for example UWCB denotes universal weak co-Büchi automata and DP denotes deterministic parity automata. If T is such an identifier, then $\mathcal{L}(T)$ denotes the class of languages characterized by automata of type T. As explained in [MP92], the language classes $\mathcal{L}(\mathrm{DB})$, $\mathcal{L}(\mathrm{DCB})$, and $\mathcal{L}(\mathrm{DB}) \cap \mathcal{L}(\mathrm{DCB})$ capture the (regular) "recurrence properties", "persistence properties", and "obligation properties", respectively. So our results clarify their role in the framework of alternating automata.

Deterministic Automata

$$\begin{tikzcd}[column sep=small] & \mathcal{L}(\mathrm{DP}) \arrow[dl, dash] \arrow[dr, dash] & \\ \mathcal{L}(\mathrm{DB}) \arrow[dr, dash] & & \mathcal{L}(\mathrm{DCB}) \arrow[dl, dash] \\ & \mathcal{L}(\mathrm{DB}) \cap \mathcal{L}(\mathrm{DCB}) & \end{tikzcd}$$

Alternating Automata

$$\begin{tikzcd}[column sep=small] & \mathcal{L}(\mathrm{AWP}) \arrow[dl, dash] \arrow[dr, dash] & \\ \mathcal{L}(\mathrm{AWCB}) \arrow[dr, dash] & & \mathcal{L}(\mathrm{AWB}) \arrow[dl, dash] \\ & \mathcal{L}(\mathrm{AWCB}) \cap \mathcal{L}(\mathrm{AWB}) & \end{tikzcd}$$

Weak Parity Automata

$$\begin{tikzcd}[column sep=small] & \mathcal{L}(\mathrm{AWP}) \arrow[dl, dash] \arrow[dr, dash] & \\ \mathcal{L}(\mathrm{UWP}) \arrow[dr, dash] & & \mathcal{L}(\mathrm{NWP}) \arrow[dl, dash] \\ & \mathcal{L}(\mathrm{DWP}) & \end{tikzcd}$$

Fig. 3. Three different characterizations of the Landweber hierarchy

Before we give the theorem stating the correctness of the characterization from Figure 3, we need some preparations.

Lemma 4. *For every deterministic Büchi automaton one can construct an equivalent universal weak co-Büchi automaton.*

Proof. For the transformation we use a simplified version of the construction from Section 4.

In [MH84] Miyano and Hayashi give an exponential transformation of alternating Büchi automata into nondeterministic Büchi automata. But the construction also gives a more general theorem, which is stated below.

Theorem 4. *Let $\mathcal{A}$ be an alternating Büchi (weak Büchi) automaton. One can construct an equivalent nondeterministic Büchi (weak Büchi) automaton $\mathcal{A}'$ and furthermore, if $\mathcal{A}$ is universal, then $\mathcal{A}'$ is deterministic.*

To apply this theorem to weak parity automata, we give a transformation of weak parity automata into Büchi automata. The idea is to remember the lowest color seen so far. This suffices to decide with a Büchi condition, whether a path is accepting or not.

Lemma 5. *Let $\mathcal{A} = (Q, \Sigma, q_0, \delta, c)$ be a weak parity automaton with $c : Q \to C$. One can construct an equivalent Büchi automaton $\mathcal{A}' = (Q', \Sigma, q_0', \delta', F')$ with the same mode of transition function as $\mathcal{A}$.*

Theorem 5. *(1)* $\mathcal{L}(\mathrm{DP}) = \mathcal{L}(\mathrm{AWP})$.
(2) $\mathcal{L}(\mathrm{DB}) = \mathcal{L}(\mathrm{AWCB}) = \mathcal{L}(\mathrm{UWP})$.
(3) $\mathcal{L}(\mathrm{DCB}) = \mathcal{L}(\mathrm{AWB}) = \mathcal{L}(\mathrm{NWP})$.
(4) $\mathcal{L}(\mathrm{DB}) \cap \mathcal{L}(\mathrm{DCB}) = \mathcal{L}(\mathrm{AWB}) \cap \mathcal{L}(\mathrm{AWCB}) = \mathcal{L}(\mathrm{DWP})$.

Proof. (1): This follows from Theorem 2 and from the fact that every language recognized by an alternating weak parity automaton is regular.

(2): From Lemma 4 follows $\mathcal{L}(\mathrm{DB}) \subseteq \mathcal{L}(\mathrm{AWCB})$ and $\mathcal{L}(\mathrm{DB}) \subseteq \mathcal{L}(\mathrm{UWP})$. Since one can transform every universal weak parity automaton into a universal Büchi automaton by Lemma 5, and then into a deterministic Büchi automaton by Theorem 4, we get $\mathcal{L}(\mathrm{UWP}) \subseteq \mathcal{L}(\mathrm{DB})$.

If we are given an alternating weak co-Büchi automaton, we dualize it, yielding an alternating weak Büchi automaton. This can be transformed into a nondeterministic weak Büchi automaton by Theorem 4. If we dualize again, we get a universal weak co-Büchi automaton, equivalent to the given automaton. Therefore we get $\mathcal{L}(\mathrm{AWCB}) \subseteq \mathcal{L}(\mathrm{UWP})$, since weak co-Büchi conditions are special cases of weak parity conditions.

We omit the proof of (3) because its the dual statement to (2), and (4) can be shown very easily, using (2) and (3), and some criteria on the loop structure of deterministic ω-automata [Lan69].

7 The PLTL Fragment

In the previous section we gave exact characterizations for the different fragments from the Landweber hierarchy. Another fragment of the regular ω-languages is the fragment of PLTL (propositional linear temporal logic, see [Eme90]) that includes all languages that can be described by PLTL formulas, or, equivalently, by first-order formulas. PLTL formulas are built up from a finite set P of atomic propositions, the boolean operators, and the temporal operators $\bigcirc$ (Next), $\Box$ (Always), $\Diamond$ (Eventually), $\mathcal{U}$ (Until). For this section we fix the alphabet $\Sigma = 2^P$. Let $\alpha \in \Sigma^\omega$, $i \in \mathbb{N}$, $p \in P$, and φ, φ' be PLTL formulas. The relation $\models$ is defined as follows.

$$
\begin{array}{ll}
\alpha, i \models p & \text{if } p \in \alpha(i), \\
\alpha, i \models \bigcirc\varphi & \text{if } \alpha, i+1 \models \varphi, \\
\alpha, i \models \Box\varphi & \text{if } \forall k \geq i\ (\alpha, k \models \varphi), \\
\alpha, i \models \Diamond\varphi & \text{if } \exists k \geq i\ (\alpha, k \models \varphi), \\
\alpha, i \models \varphi \mathrm{U} \varphi' & \text{if } \exists k \geq i\ (\alpha, k \models \varphi' \text{ and } \forall i \leq j < k\ (\alpha, j \models \varphi)).
\end{array}
$$

For the boolean operators, $\models$ is defined in the straightforward way.

A PLTL formula φ defines the language $L(\varphi) = \{\alpha \in \Sigma^\omega \mid \alpha, 0 \models \varphi\}$. The PLTL fragment of the regular ω-languages is the class of all languages that can be defined by a PLTL formula.

In this section we give an exact automata theoretic characterization of this fragment in terms of a subclass of alternating weak parity automata, so called alternating linear automata.

An alternating weak parity automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta, c)$ is called a *linear automaton*, if in the transition graph there are no cycles containing 2 or more states, and if along each path through the transition graph the colors of the states do not increase.

A simple induction shows that a PLTL-definable language can also be recognized by an alternating linear automaton (see e.g. [Var97]). Here we show the other direction, which was independently shown by Rohde [Roh97].

Theorem 6. *Let $\mathcal{A} = (Q, \Sigma, \theta_0, \delta, c)$ be an alternating linear automaton. There is a PLTL formula χ such that $L(\mathcal{A}) = L(\chi)$.*

Proof. For $\theta \in \mathcal{B}^+(Q)$ let $\mathcal{A}(\theta) = (Q, \Sigma, \theta, \delta, c)$ (an automaton with an initial formula instead of an initial state). For $R \subseteq P$ let $\psi_R = \bigwedge_{p \in R} p \wedge \bigwedge_{p \notin R} \neg p$.

We construct for every $\theta \in \mathcal{B}^+(Q)$ a PLTL formula $\chi(\theta)$ with $L(\chi(\theta)) = L(\mathcal{A}(\theta))$ and then set $\chi = \chi(\theta_0)$. Let $\theta \in \mathcal{B}^+(Q)$. If for all $q \in Q$ that occur in θ the formula $\chi(q)$ is already known, then we obtain $\chi(\theta)$ by replacing each atom q in θ by the formula $\chi(q)$. Furthermore we set $\chi(\top) = \top$ and $\chi(\bot) = \bot$.

Let $q \in Q$ and let $Tr(q)$ denote the set of states q' such that a transition leads from q to q'. Since $\mathcal{A}$ is a linear automaton, we can assume by induction that for all $q' \in Tr(q) \setminus \{q\}$ the formula $\chi(q')$ is already known, and therefore also all $\chi(\theta)$ with $\theta \in \mathcal{B}^+(Tr(q) \setminus \{q\})$ are known.

For all $R \subseteq P$ we can write the transition formula for q and R in the form $\delta(q, R) = (q \wedge \theta_{R,q}) \vee \theta'_{R,q}$ with $\theta_{R,q}, \theta'_{R,q} \in \mathcal{B}^+(Tr(q) \setminus \{q\})$. We define

$$
\chi(q) = \begin{cases} \varphi_q \mathcal{U} \varphi'_q & \text{if } c(q) \text{ is even,} \\ \varphi_q \mathcal{U} \varphi'_q \vee \Box\varphi_q & \text{if } c(q) \text{ is odd,} \end{cases}
$$

with

$$\varphi_q = \bigvee_{R \subseteq P} (\psi_R \wedge \bigcirc \chi(\theta_{R,q})) \text{ and } \varphi'_q = \bigvee_{R \subseteq P} (\psi_R \wedge \bigcirc \chi(\theta'_{R,q})).$$

To show that for all $\theta \in \mathcal{B}^+(Q)$ the equality $L(\chi(\theta)) = L(\mathcal{A}(\theta))$ holds, it suffices to show $L(\chi(q)) = L(\mathcal{A}(q))$ for all $q \in Q$.

Let $q \in Q$ and let $\alpha \in L(\chi(q))$. If $c(q)$ is odd, then there exists a $k \in \mathbb{N}$ such that $\alpha, i \models \varphi_q$ for all $i < k$ and $\alpha, k \models \varphi'_q$. Thus, for all $i < k$, the word $\alpha[i+1, \infty)$ is in $L(\chi(\theta_{\alpha(i),q}))$ and $\alpha[k+1, \infty)$ is in $L(\chi(\theta'_{\alpha(k),q}))$. By induction we know that $L(\chi(\theta_{\alpha(i),q})) = L(\mathcal{A}(\theta_{\alpha(i),q})$ and $L(\chi(\theta'_{\alpha(k),q})) = L(\mathcal{A}(\theta'_{\alpha(k),q})$. An accepting run of $\mathcal{A}(q)$ has the following form:

$$
\begin{array}{ccc}
q \xrightarrow{\alpha(0)} q \xrightarrow{\alpha(1)} q & \cdots & q \xrightarrow{\alpha(k-1)} q \xrightarrow{\alpha(k)} \mathcal{A}(\theta'_{\alpha(k),q}) \\
\mathcal{A}(\theta_{\alpha(0),q}) \quad \mathcal{A}(\theta_{\alpha(1),q}) & \cdots & \mathcal{A}(\theta_{\alpha(k-1),q})
\end{array}
$$

The identifiers of the automata stand for accepting runs of these automata on the corresponding suffix of α.

If $c(q)$ is even then the proof is analogous accept for the case that the $\Box\varphi_q$ part is satisfied. Then we get a run of $\mathcal{A}_q$ with an infinite path labelled with q. This run is also accepting, because $c(q)$ is even.

For the other direction let $\alpha \in L(\mathcal{A}_q)$. If $c(q)$ is odd, then an accepting run of $\mathcal{A}_q$ on α is of the form as given above. But then, using the induction hypothesis, $\alpha \models \varphi_q \mathcal{U} \varphi'_q$. In case $c(q)$ is even, we can also get an accepting run with an infinite path labelled with q, but then $\alpha \models \Box\varphi_q$.

Acknowledgement

We thank Olivier Carton, Marcin Jurdziński, Narayan Kumar, Madhusudan, and Madhavan Mukund, as well as the referees for useful comments and discussions.

References

[Büc62] J.R. Büchi. On a decision method in restricted second order arithmetic. In *Proc. International Congress on Logic, Method and Philos. Sci. 1960*, pages 1–11, 1962.

[Eme90] E.A. Emerson. Temporal and modal logic. In J. v. Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. Elsevier Science Publishers, 1990.

[Kla91] N. Klarlund. Progress measures for complementation of ω-automata with applications to temporal logic. In *Proc. 32nd FOCS*, pages 358–367, 1991.

[KV97] O. Kupferman and M.Y. Vardi. Weak alternating automata are not that weak. In *Proc. 5th Israeli Symposium on Theory of Computing and Systems*, pages 147–158. IEEE Computer Society Press, 1997.

[Lan69] L.H. Landweber. Decision poblems for ω-automata. *Math. System Theory*, 3:376–384, 1969.

[Löd98] C. Löding. Methods for the transformation of ω-automata: Complexity and connection to second order logic. Master's thesis, Christian-Albrechts-University of Kiel, 1998.

[MH84] S. Miyano and T. Hayashi. Alternating finite automata on ω-words. *Theoretical Computer Science*, 32:321–330, 1984.

[MP92] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems*. Springer, 1992.

[MS87] D.E. Muller and P.E. Schupp. Alternating automata on infinite trees. *Theoretical Computer Science*, 54:267–276, 1987.

[MSS86] D.E. Muller, A. Saoudi, and P.E. Schupp. Alternating automata, the weak monadic theory of the tree and its complexity. In *Proc. 13th ICALP, LNCS 226*, pages 275–283, 1986.

[Roh97] S. Rohde. *Alternating automata and the temporal logic of ordinals*. PhD thesis, University of Illinois at Urbana-Champaign, 1997.

[Tho97] W. Thomas. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Language Theory*, volume III, pages 385–455. Springer-Verlag, 1997.

[Tho99] W. Thomas. Complementation of Büchi automata revisited. In J. Karhumäki et al., editor, *Jewels are Forever, Contributions on Theoretical Computer Science in Honor of Arto Salomaa*, pages 109–122. Springer, 1999.

[Var97] M.Y. Vardi. Alternating automata: Unifying truth and validity checking for temporal logics. In *14th International Conference on Automated Deduction, Lecture Notes in Artificial Intelligence 1249*, pages 191–206, 1997.