Inherent Size Blowup in ω-Automata*

Udi Boker

Interdisciplinary Center (IDC) Herzliya, Israel

Abstract. We clarify the succinctness of the different ω-automata types and the size blowup involved in boolean operations on them. We argue that there are good reasons for the classic acceptance conditions, while there is also place for additional acceptance conditions, especially in the deterministic setting; Boolean operations on deterministic automata with the classic acceptance conditions involve an exponential size blowup, which can be avoided by using stronger acceptance conditions. In particular, we analyze the combination of hyper-Rabin and hyper-Streett automata, which we call *hyper-dual*, and show that in the deterministic setting it allows for exponential succinctness compared to the classic types, boolean operations on it only involve a quadratic size blowup, and its nonemptiness, universality, and containment checks are in PTIME.

Keywords: ω-automata · Size blowup · Acceptance conditions

1 Introduction

Automata on infinite words, often called ω-automata, were introduced in the 1960s in the course of solving decision problems in logic, and since the 1980s they play a key role in formal verification of reactive systems. Unlike automata on finite words, these automata have various acceptance conditions (types), the most classic of which are weak, Büchi, co-Büchi, parity, Rabin, Streett, and Muller.

There are good reasons for having multiple acceptance conditions in ω-automata: each is closely connected to some other formalisms and logics, and has its advantages and disadvantages with respect to succinctness and to the complexity of resolving decision problems on it (see [6]).

There is a massive literature on the translations between the different automata types, accumulated along the past 55 years, and continuing to these days. (See, for example, [14, 33, 34, 25, 32, 41, 17, 27, 36, 8, 37, 4].) Having "only" seven classic types, where each can be deterministic or nondeterministic, we have 175 possible non-self translations between them, which has become difficult to follow. Moreover, it turns out that there is inconsistency in the literature results concerning the size of automata—Some only consider the number of states, some also take into account the index (namely, the size of the acceptance condition), while ignoring the alphabet size, and some do consider the alphabet size, but ignore the index.

* This work was supported by the Israel Science Foundation grant 1373/16.

To make an order with all of these results, we maintain a website [3] that provides information and references for each of the possible translations. The high-level tables of the size blowup and of the state blowup involved in the translations are given in Table 1.

There are many works on the complementation of nondeterministic ω-automata (see [38] for a survey until 2007, after which there are yet many new results), while very few on boolean operations on deterministic ω-automata. This is possibly because nondeterministic automata are exponentially more succinct than deterministic ones and are adequate for model checking. However, in recent years there is a vast progress in synthesis and in probabilistic model checking, which require deterministic or almost deterministic automata, such as limit-deterministic [39] or good-for-games automata [21, 7, 9].

In [6], we completed the picture of the size blowup involved in boolean operations on the classic ω-automata types, as summarized in Table 2. Observe that all of the classic ω-regular-complete automata types, namely parity, Rabin, Streett, and Muller, admit in the deterministic setting an exponential size blowup on boolean operations, even on the positive ones of union and intersection.

Indeed, the problem with boolean operations on classic deterministic automata and the current interest in the deterministic setting, may explain the emergence of new, or renewed, automata types in the past seven years. Among these are "Emerson-Lei" (EL), which was presented in 1985 [18], and was recently "rediscovered" within the "Hanoi" format [1], "generalized-Rabin" [24], and "generalized-Streett" [2]. The EL condition allows for an arbitrary boolean formula over sets of states that are visited finitely or infinitely often, generalized-Rabin extends the Rabin pairs into lists, and generalized-Streett analogously extends Streett pairs.

While boolean operations on EL automata are obviously simple, it is known that its nonemptiness check is NP-complete [18] and its universality check is EXPSPACE-complete [20].

We analyzed in [6] additional non-classic acceptance conditions, and showed that there is no inherent reason for having an exponential size blowup in positive boolean operations on deterministic ω-regular-complete automata that admit a PTIME nonemptiness check: We observed that generalized-Rabin is a special case of a disjunction of Streett conditions, which was already considered in 1985 under the name "canonical form" [18], and which we dubbed "hyper-Rabin". We showed that it may be exponentially more succinct than the classic types, it allows for union and intersection with only a quadratic size blowup, and its nonemptiness check is in PTIME (see Tables 2 and 3). Indeed, there seem to also be practical benefits for generalized-Rabin automata [24, 13, 19], which may possibly be extended to the more general hyper-Rabin condition.

We further analyze in Section 5 the possibility of deterministic ω-regular-complete automata that admit PTIME algorithms for nonemptiness, universality, and containment checks, and for which all boolean operations, including complementation, only involve a quadratic size blowup. We show that it is indeed possible with an approach that upfront seems to only bring redundancy—

maintaining a pair of equivalent automata, one with the hyper-Rabin condition and one with its dual (hyper-Streett) condition. We call such a pair a *hyper-dual* automaton. Observe that in the deterministic setting, it is the same as a pair of hyper-Rabin automata, one for a language L and one for its complement $\overline{L}$.

One may wonder what benefit can we have from a deterministic hyper-dual automaton, having inner automata for both L and $\overline{L}$, rather than having an automaton only for L, and complementing it when necessary. We list below some of the benefits:

- A deterministic hyper-dual automaton, despite having a pair of inner automata for both L and $\overline{L}$, is at most twice the size of classic automata, such as Rabin and Streett, and may be exponentially more succinct than them. (Propositions 2 and 7).
- The approach of maintaining an automaton and its complement obviously allows for "free" complementation, yet it might have a price in union and intersection. For example, a pair of equivalent deterministic automata, one with the Rabin condition and one with its dual (Streett) condition, would have an exponential size blowup on both union and intersection. The hyper-dual combination is strong enough to prevent this price, and not too strong for preserving decision problems in PTIME.
- In some scenarios, an automaton is generated iteratively, starting with a basic one, and enlarging it with consecutive boolean operations. In such scenarios, a hyper-dual automaton may have a big advantage—its initial generation is not more difficult than of Rabin or Streett automata, and each boolean operation only involves up to a quadratic size blowup, while preserving the ability to check nonemptiness, universality, and containment in PTIME.
- Compared to complementing a hyper-Rabin automaton on demand:
  - In theory, a deterministic hyper-Rabin automaton can always be complemented into a hyper-Streett automaton that is not bigger than the corresponding hyper-dual automaton. Yet, the complementation procedure is exponential, and does not guarantee the smallest possible hyper-Streett automaton. Hence, having in a hyper-dual automaton a small hyper-Streett automaton in addition to the hyper-Rabin automaton provides a significant potential advantage.
  - In iterative generations, complementation need not be made over and over again, size optimizations would take into account both the hyper-Rabin and hyper-Streett conditions, and progress is guaranteed to be homogenous with no heavy steps in the middle.
  - When expressing some property with only a hyper-Rabin automaton in mind, it might be that we generate a small initial automaton whose complementation would involve an exponential size blowup. Targeting hyper-dual automata, we may limit ourselves to properties that can be expressed with small hyper-dual automata, which then guarantees easy boolean operations.

2 ω-Automata and their Acceptance Conditions

A *nondeterministic automaton* is a tuple $\mathcal{A} = \langle \Sigma, Q, \delta, \iota, \alpha \rangle$, where Σ is the input alphabet, Q is a finite set of states, $\delta : Q \times \Sigma \to 2^Q$ is a transition function, $\iota \subseteq Q$ is a set of initial states, and α is an acceptance condition. If $|\iota| = 1$ and for every $q \in Q$ and $\sigma \in \Sigma$, we have $|\delta(q, \sigma)| \leq 1$, we say that $\mathcal{A}$ is *deterministic*.

A *run* $r = r(0), r(1), \cdots$ of $\mathcal{A}$ on an infinite word $w = w(0) \cdot w(1) \cdots \in \Sigma^\omega$ is an infinite sequence of states such that $r(0) \in \iota$, and for every $i \geq 0$, we have $r(i+1) \in \delta(r(i), w(i))$. An automaton accepts a word if it has an accepting run on it (as defined below, according to the acceptance condition). The language of $\mathcal{A}$, denoted by $L(\mathcal{A})$, is the set of words that $\mathcal{A}$ accepts. We also say that $\mathcal{A}$ *recognizes* the language $L(\mathcal{A})$. Two automata, $\mathcal{A}$ and $\mathcal{A}'$, are *equivalent* iff $L(\mathcal{A}) = L(\mathcal{A}')$.

Acceptance is defined with respect to the set $\mathit{inf}(r)$ of states that the run r visits infinitely often. Formally, $\mathit{inf}(r) = \{q \in Q \mid$ for infinitely many $i \in \mathbb{N}$, we have $r(i) = q\}$.

We start with describing the most classic acceptance conditions, after which we will describe some additional ones.

- *Büchi*, where $\alpha \subseteq Q$, and r is accepting iff $\mathit{inf}(r) \cap \alpha \neq \emptyset$.
- *co-Büchi*, where $\alpha \subseteq Q$, and r is accepting iff $\mathit{inf}(r) \cap \alpha = \emptyset$.
- *weak* is a special case of the Büchi condition, where every strongly connected component of the automaton is either contained in α or disjoint to α.
- *parity*, where $\alpha = \{S_1, S_2, \ldots, S_{2k}\}$ with $S_1 \subset S_2 \subset \cdots \subset S_{2k} = Q$, and r is accepting iff the minimal i for which $\mathit{inf}(r) \cap S_i \neq \emptyset$ is even.
- *Rabin*, where $\alpha = \{\langle B_1, G_1 \rangle, \langle B_2, G_2 \rangle, \ldots, \langle B_k, G_k \rangle\}$, with $B_i, G_i \subseteq Q$ and r is accepting iff for some $i \in [1..k]$, we have $\mathit{inf}(r) \cap B_i = \emptyset$ and $\mathit{inf}(r) \cap G_i \neq \emptyset$.
- *Streett*, where $\alpha = \{\langle B_1, G_1 \rangle, \langle B_2, G_2 \rangle, \ldots, \langle B_k, G_k \rangle\}$, with $B_i, G_i \subseteq Q$ and r is accepting iff for all $i \in [1..k]$, we have $\mathit{inf}(r) \cap B_i = \emptyset$ or $\mathit{inf}(r) \cap G_i \neq \emptyset$.
- *Muller*, where $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_k\}$, with $\alpha_i \subseteq Q$ and r is accepting iff for some $i \in [1..k]$, we have $\mathit{inf}(r) = \alpha_i$.

Notice that Büchi and co-Büchi are special cases of the parity condition, which is in turn a special case of both the Rabin and Streett conditions.

Two additional types that are in common usage are:

- *very weak* (*linear*) is a special case of the Büchi (and weak) condition, where all cycles are of size one (self loops).
- *generalized Büchi*, where $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_k\}$, with $\alpha_i \subseteq Q$ and r is accepting iff for every $i \in [1..k]$, we have $\mathit{inf}(r) \cap \alpha_i \neq \emptyset$.

A general way of describing an acceptance condition was given by Emerson and Lei in 1985 [18]: For a set S of states, we define that $\mathit{Inf}(S)$ holds in a run r if $S \cap \mathit{inf}(r) \neq \emptyset$ and $\mathit{Fin}(S)$ holds otherwise. Then,

- *Emerson-Lei* is an arbitrary boolean formula over *Fin* and *Inf* of sets of states. (A positive boolean formula is enough, as $\neg \mathit{Fin}(S) = \mathit{Inf}(S)$.)

Using the Emerson-Lei notation, we define below some additional types that were defined (or renewed) in recent years.

- *Generalized-Rabin*: $\bigvee_{i=1}^{n} Fin(B_i) \wedge Inf(G_{i_1}) \wedge Inf(G_{i_2}) \wedge \ldots \wedge Inf(G_{i_{k_i}})$.
- *Generalized-Streett*: $\bigwedge_{i=1}^{n} Inf(G_i) \vee Fin(B_{i_1}) \vee Fin(B_{i_2}) \vee \ldots \vee Fin(B_{i_{k_i}})$.
- *Hyper-Rabin*: $\bigvee_{i=1}^{n} \bigwedge_{j=1}^{m} Fin(B_{i,j}) \vee Inf(G_{i,j})$.
- *Hyper-Streett*: $\bigwedge_{i=1}^{n} \bigvee_{j=1}^{m} Fin(B_{i,j}) \wedge Inf(G_{i,j})$.

Another related type is *circuit* [22], which further shortens Emerson-Lei, by representing the acceptance formula as a boolean circuit. In Section 5, we also consider the combination of hyper-Rabin and hyper-Streett automata, which we call *hyper-dual*. Very-weak, weak, and co-Büchi automata, as well as deterministic Büchi automata, are less expressive than the other automata types, which recognize all ω-regular languages.

The *index* of an automaton is the length of the boolean formula describing its acceptance condition. For the more standard types, this definition coincides with the standard definition of index: The number of sets in the generalized-Büchi, parity, and Muller conditions, the number of pairs in the Rabin and Streett conditions, and 1 in the very-weak, weak, co-Büchi, and Büchi conditions.

The *size* of an automaton is the maximum size of its elements; that is, it is the maximum of the alphabet size, the number of states, the number of transitions, and the index.

3 Succinctness

Size versus number of states. Out of the four elements that constitute the size of an automaton, the number of states and the index are the dominant ones.

Considering the alphabet, the common practice is to provide the upper bounds for arbitrary alphabets and to seek lower bounds with fixed alphabets. For example, [28] strengthen the lower bound of [30] by moving to a fixed alphabet, and [41] starts with automata over a rich alphabet and then moves to a fixed alphabet. It turns out that this approach works well for all relevant translations, eliminating the influence of the alphabet. As for the number of transitions, they are bounded by the size of the alphabet times quadratically the number of states, and the transition blowup tends to go hand in hand with the state blowup.

Considering the number of states and index, one cannot get the full picture by studying their blowup separately, as they are interconnected, and sometimes have a trade-off between them. For example, one can translate a Streett automaton to a Rabin automaton with an exponential state blowup and no index blowup [14] as well as with only a quadratic state blowup and an exponential index blowup [5]. Therefore, there is only a quadratic inevitable state blowup and no inevitable index blowup. Yet, there is an exponential inevitable size blowup [5].

The high-level tables of the size blowup and of the state blowup involved in the translations of automata with classic acceptance conditions are given in Table 1. The size blowup relates to an automaton of size n, and the state blowup to an automaton with n states (and index as large as desired). The

Size Blowup From	To	Det. W	Det. C	Det. B	Det. P	Det. R	Det. S	Det. M	Nondet. W	Nondet. C	Nondet. B	Nondet. P	Nondet. R	Nondet. S	Nondet. M
Det.	W	·							·						
	C		·							·		$\Theta(n)$			
	B			·	$\Theta(n)$						·				
	P				·			$2^{\Theta(n)}$			$\Theta(n^2)$	·			$2^{\Theta(n)}$
	R		$2^{\Theta(n)}$			· $\Theta(2^{n \log n})$			$O(n^2)$ $\Omega(n)$			$O(n^2)$ $\Omega(n)$	·	$O(n^2)$ $\Omega(n)$	
	S			$2^{\Theta(n)}$			·		$\Theta(n)$			$2^{\Theta(n)}$		·	
	M		$O(n^2)$ $\Omega(n)$	?		$O(2^{n \log n})$ $2^{\Omega(n)}$		·	$O(n^2)$ $\Omega(n)$		$\Theta(n^3)$		$\Theta(n^2)$		·
Non-Det.	W								·						
	C		$2^{\Theta(n)}$							·		$\Theta(n)$			
	B				$2^{\Theta(n \log n)}$			$*^1$			·				
	P							$2^{2^{\Omega(n)}}$			$\Theta(n^2)$	·			$2^{\Theta(n)}$
	R			$*^2$	$2^{\Theta(n^2 \log n)}$				$2^{\Theta(n)}$			$O(n^2)$ $\Omega(n)$	·	$O(n^2)$ $\Omega(n)$	
	S													·	
	M			$*^3$	$2^{O(n^3 \log n)}$ $2^{\Omega(n \log n)}$						$\Theta(n^3)$		$\Theta(n^2)$		·

State Blowup From	To	Det. W	Det. C	Det. B	Det. P	Det. R	Det. S	Det. M	Nondet. W	Nondet. C	Nondet. B	Nondet. P	Nondet. R	Nondet. S	Nondet. M
Det.	W	·							·						
	C		·		$\Theta(n)$					·		$\Theta(n)$			
	B			·							·				
	P				·						$\Theta(n^2)$	·			
	R		$2^{\Theta(n)}$			·							·		
	S			$2^{\Theta(n)}$		$\Theta(2^{n \log n})$	·		$\Theta(n)$		$2^{\Theta(n)}$		$\Theta(n^2)$	·	
	M							·							·
Non-Det.	W								·						
	C		$2^{\Theta(n)}$							·		$\Theta(n)$			
	B					$2^{\Theta(n \log n)}$					·				
	P										$\Theta(n^2)$	·			
	R			?		$2^{2^{\Theta(n)}}$			$2^{\Theta(n)}$				·		
	S			$*^4$	?	$*^5$	?	$*^4$					$\Theta(n^2)$	·	
	M			?		$2^{2^{\Theta(n)}}$									·

$*^1$: Upper bounds between $2^{2^{O(n)}}$ and $2^{2^{O(n^3 \log n)}}$ $*^2$: To DBW: $2^{\Omega(n \log n)}$
$*^3$: Lower bound to DBW: $2^{\Omega(n)}$ $*^4$: $2^{O(n^2 \log n)}$ and $2^{\Omega(n \log n)}$ $*^5$: $2^{\Theta(n^2 \log n)}$

Table 1. Size blowup and state blowup involved in automata translations [3].

capital letters stand for the type names: Weak, Co-Büchi, Büchi, etc. A question mark in the tables stands for an exponential gap between the currently known lower and upper bounds. The size blowup involved in the translations of the stronger acceptance conditions, as discussed in Section 5, is given in Table 4.

Inevitable: Succinctness + Complementation $\geq$ Double-Exp. Aside from the translations between specific automata types, one may wonder what might be the succinctness of an arbitrary, possibly yet unknown, automaton type. It turns out that there is an inherent tradeoff between the succinctness of an automaton and the size blowup involved in its complementation—It is shown in [35] that there is a family of ω-regular languages $\{L_n\}_{n\geq 1}$, such that for every n, there is an Emerson-Lei automaton of size n for L_n, while every ω-automaton for $\overline{L_n}$ has at least 2^{2^n} states.

Hence, for an automaton of some type T whose complementation only involves a single-exponential size blowup, there must also be at least a single-exponential size blowup in translating arbitrary ω-automata into T-automata. Analogously, if we aim for a single-exponential blowup in determinization, and no blowup in the complementation of deterministic automata, there must be at least a double-exponential size blowup in translating arbitrary automata into deterministic T-automata.

In this sense, the classic types, except for Muller, provide a reasonable tradeoff between their succinctness and the size blowup involved in their determinization and complementation, having all of these measures singly exponential.

Proposition 1 ([6]). *For every $n \in \mathbb{N}$ and nondeterministic ω-automaton of size n, there is an equivalent nondeterministic Büchi automaton of size in $2^{O(n)}$ and an equivalent deterministic parity automaton of size in $2^{2^{O(n)}}$.*

4 Boolean Operations and Decision Problems

In the nondeterministic setting, boolean operations on the classic automata types, except for Muller, roughly involve an asymptotically optimal size blowup: linear for union, quadratic for intersection, and singly exponential for complementation. These blowups are inevitable already in automata over finite words. In the deterministic setting, however, the picture is different, having an exponential size blowup on union or intersection for all of the classic ω-regular-complete types. In this setting, the stronger acceptance conditions, as elaborated on in Section 5, match the inevitable blowups, having only quadratic size blowup. The size blowup involved in boolean operations is summarized in Table 2.

Seeking small size blowup on boolean operations is only one side of the equation—one should consider it in conjunction with the succinctness of the automaton type and the complexity of the nonemptiness and universality problems. The EL acceptance condition, for example, is very flexible and there is a small size blowup in boolean operations on deterministic EL automata, however at the cost of a high complexity of the decision problems.

<table>
<tr><th rowspan="2">Operations
Size Blowup</th><th colspan="3">On Deterministic Automata</th><th colspan="3">On Nondeterministic Automata</th></tr>
<tr><th>Union</th><th>Intersect.</th><th>Complement.</th><th>Union</th><th>Intersect.</th><th>Complement.</th></tr>
<tr><td>Weak</td><td colspan="2" rowspan="3">Quadratic</td><td>No blowup</td><td rowspan="7">Linear</td><td rowspan="3">Quad.</td><td rowspan="2">$2^{\Theta(n)}$
[31]
(if possible)</td></tr>
<tr><td>Co-Büchi</td><td rowspan="2">No blowup
[25]
(if possible)</td></tr>
<tr><td>Büchi</td><td rowspan="2">$2^{\Theta(n \log n)}$
$[30, 33, 11]$</td></tr>
<tr><td>Parity</td><td colspan="2">Exponential
$[29, 6]$</td><td>No blowup</td><td>Quad. -
Quartic</td></tr>
<tr><td>Rabin</td><td>Quad.
[6]</td><td>Exp.
[6]</td><td rowspan="2">Exp.
[28]</td><td rowspan="2">Quad.
[6]</td><td rowspan="2">$2^{\Theta(n^2 \log n)}$
$[26, 12, 10]$</td></tr>
<tr><td>Streett</td><td>Exp.
[6]</td><td>Quad.
[6]</td></tr>
<tr><td>Muller</td><td colspan="2">Exp.
[6]</td><td>Exp.
[33]</td><td>Exp.
[6]</td><td>Double-Exp.
[6]</td></tr>
<tr><td>Hyper-Rabin</td><td colspan="2" rowspan="4">Quadratic
Prop. 3, [6]</td><td rowspan="2">Exp.
[6]</td><td rowspan="4">Linear</td><td rowspan="4">Quad.
Prop. 4,
[6]</td><td>Exp.
[6]</td></tr>
<tr><td>Hyper-Streett</td><td>Double-Exp.
[6]</td></tr>
<tr><td>Hyper-dual</td><td rowspan="2">No blowup</td><td>Exp.
Prop. 4</td></tr>
<tr><td>Emerson-Lei</td><td>Double-Exp.
[6]</td></tr>
</table>

Table 2. The size blowup involved in boolean operations.

The best possible complexity for the nonemptiness problem is NLOGSPACE and linear time, taking its lower bound from the reachability problem. It is indeed achieved with Büchi automata. For the stronger classic acceptance conditions, except for Streett, it remains in NLOGSPACE, while exceeding the linear time, and for Streett it is PTIME-complete. The further stronger conditions either remain in PTIME, as Hyper-Rabin and Hyper-dual, or become NP-complete, as hyper-Streett and Emerson-Lei.

The best possible complexity for the universality problem on nondeterministic automata is PSPACE-complete, taking its lower bound from automata on finite words. It is achieved for all the classic types, as well as for the hyper-Rabin and hyper-dual types. A possible way to perform the universality check of an automaton $\mathcal{A}$ of these types is the following: translate $\mathcal{A}$ to a Streett automaton $\mathcal{B}$ with only a polynomial size blowup, then complement $\mathcal{B}$ to a Büchi automaton $\mathcal{C}$ on the fly, having a potential exponential space, and check the nonemptiness of $\mathcal{C}$ in logarithmic space, yielding a PSPACE algorithm in the size of $\mathcal{A}$ [33].

The complexity of the nonemptiness and universality problems (of nondeterministic automata) is summarized in Table 3.

Checks of Nondeterministic	Nonemptiness	Universality
Weak	Linear time, NL-complete [16, 40]	PSPACE-comp. [33]
Co-Büchi		
Büchi		
Parity	$O(m \log k)$ time, NL-comp. [18, 23]	
Rabin	$O(mk)$ time, NL-comp. [40]	
Streett	PTIME-comp. [18]	
Muller	NL-comp. [15]	
Hyper-Rabin	PTIME-comp. [18]	PSPACE-comp. [18]
Hyper-Streett	NP-complete [6]	EXPSPACE-comp. [20]
Hyper-dual	PTIME-comp. Prop. 8	PSPACE-comp. Prop. 8
Emerson-Lei	NP-complete [18]	EXPSPACE-comp. [33]

Table 3. The complexity of the nonemptiness and universality checks of nondeterministic automata. The complexity is w.r.t. the automaton size n, and if specified, w.r.t. m states and index k.

Translations Size Blowup From	To	Deterministic		Nondeterministic	
		H-Rabin	H-Streett	H-Rabin	H-Streett
Det.	Hyper-Rabin	·	Exp.	·	$O(n^2)$
	Hyper-Streett	Exp.	·	Exp.	·
Non-Det.	Hyper-Rabin	Exp.		·	$O(n^2)$
	Hyper-Streett	Double-Exp.		Exp.	·

Table 4. The size blowup involved in translations between hyper-Rabin/Streett automata. The translations to and from generalized-Rabin/Streett automata have the same blowup. All results are from [6].

5 Hyper-Dual

We look in this section into a new automaton type that consists of two equivalent inner automata, one with the hyper-Rabin condition and one with the hyper-Streett condition. In the deterministic setting, it is the same as having two hyper-Rabin automata, one for the requested language and one for its complement.

Despite the first impression that it only brings redundancy, it seems to have an interesting potential in the deterministic setting—It allows for all boolean operations with only a quadratic state blowup, and for polynomial-time algorithms of the decision problems of emptiness, universality, and automata comparison. (A further discussion of its benefits is given at the end of the Introduction.)

Construction. Constructing a hyper-dual automaton is not more difficult than constructing any classic automaton—having an automaton $\mathcal{A}$ with the Rabin or Streett condition, a pair $(\mathcal{A}, \mathcal{A})$ is a proper hyper-dual automaton.

Proposition 2. *The Rabin and Streett acceptance conditions are special cases of both the hyper-Rabin and the hyper-Streett conditions.*

Proof. Observe that the Streett condition is the hyper-Rabin condition with a single disjuct, while the Rabin condition is a hyper-Rabin condition, in which every disjunct consists of two elements $Fin(B) \vee Inf(\emptyset)$ and $Fin(Q) \vee Inf(G)$, where Q is the entire set of states in the automaton, and $B, G \subseteq Q$.

As for hyper-Streett, the claim follows from the duality to hyper-Rabin.

Boolean operations. Further generating deterministic hyper-dual automata by boolean operations is easy, involving only a quadratic size blowup—The complement of a deterministic hyper-dual automaton $\mathcal{C} = (\mathcal{A}, \mathcal{B})$ is $\overline{\mathcal{C}} = (\overline{\mathcal{B}}, \overline{\mathcal{A}})$, while the union and intersection of deterministic hyper-dual automata $\mathcal{C}' = (\mathcal{A}', \mathcal{B}')$ and $\mathcal{C}'' = (\mathcal{A}'', \mathcal{B}'')$ is $\mathcal{C} = (\mathcal{A}' \cup \mathcal{A}'', \mathcal{B}' \cup \mathcal{B}'')$ and $\mathcal{C} = (\mathcal{A}' \cap \mathcal{A}'', \mathcal{B}' \cap \mathcal{B}'')$, respectively, involving a quadratic size blowup (Table 2).

Proposition 3. *Complementation of a deterministic hyper-dual automaton involves no size blowup, and the union and intersection of two deterministic hyper-dual automata involve a quadratic size blowup.*

In the nondeterministic setting, it is almost similar to handling only hyper-Rabin automata, for a simple reason—translating a nondeterministic hyper-Rabin automaton into an equivalent hyper-Streett automaton only involves a quadratic size blowup (Table 4).

Proposition 4. *Complementation of a nondeterministic hyper-dual automaton involves a singly-exponential size blowup, and the union and intersection of two nondeterministic hyper-dual automata involve a quadratic size blowup.*

Properness check. Constructing a hyper-dual automaton from a classic automaton and boolean operations guarantees its properness. Checking whether an arbitrary pair of hyper-Rabin and hyper-Streett automata is a proper hyper-dual automaton might not be too interesting and it is also coNP-complete for deterministic automata and EXPSPACE-complete for nondeterministic automata.

Proposition 5. *Given a pair $\mathcal{C} = (\mathcal{A}, \mathcal{B})$ of a deterministic hyper-Rabin automaton $\mathcal{A}$ and a deterministic hyper-Streett automaton $\mathcal{B}$, the problem of deciding whether $\mathcal{C}$ is a proper hyper-dual automaton is coNP-complete.*

Proof. For the upper bound, we should validate that $L(\mathcal{A}) = L(\mathcal{B})$. This is the case iff $L(\mathcal{A}) \subseteq L(\mathcal{B})$ and $L(\mathcal{B}) \subseteq L(\mathcal{A})$, which is the case iff $L(\mathcal{A}) \cap \overline{L(\mathcal{B})} = \emptyset$ and $L(\mathcal{B}) \cap \overline{L(\mathcal{A})} = \emptyset$.

Observe that $\overline{L(\mathcal{B})} = L(\overline{\mathcal{B}})$, and that $\overline{\mathcal{B}}$ is a hyper-Rabin automaton. Thus, checking whether $L(\mathcal{A}) \subseteq L(\mathcal{B})$ is in PTIME: constructing $\mathcal{A} \cap \overline{\mathcal{B}}$ is possible in quadratic time (Table 2), and its emptiness check in PTIME (Table 3).

As for checking whether $L(\mathcal{B}) \subseteq L(\mathcal{A})$, observe that $\overline{L(\mathcal{A})} = L(\overline{\mathcal{A}})$, and that $\overline{\mathcal{A}}$ is a hyper-Streett automaton. Thus, the check is in co-NP: constructing $\mathcal{B} \cap \overline{\mathcal{A}}$ is possible in quadratic time (Table 2), and its nonemptiness check in NP (Table 3).

For the lower bound, consider a pair $\mathcal{C}$ in which $\mathcal{A}$ is an empty automaton. Then $\mathcal{C}$ is proper iff $\mathcal{B}$ is empty, and the nonemptiness check of a DHSW is NP-complete (Table 3).

Proposition 6. *Given a pair $\mathcal{C} = (\mathcal{A}, \mathcal{B})$ of a nondeterministic hyper-Rabin automaton $\mathcal{A}$ and a nondeterministic hyper-Streett automaton $\mathcal{B}$, the problem of deciding whether $\mathcal{C}$ is a proper hyper-dual automaton is EXPSPACE-complete.*

Proof. For the upper bound, we can translate $\mathcal{A}$ and $\mathcal{B}$ to Büchi automata, having an exponential size blowup (Proposition 1), and then check the equivalence of the two Büchi automata in PSPACE.

For the lower bound, consider a pair $\mathcal{C}$ in which $\mathcal{A}$ is an automaton recognizing Σ^ω. Then $\mathcal{C}$ is proper iff $\mathcal{B}$ is universal, and the universality check of a hyper-Streett automaton is EXPSPACE-complete (Table 3).

Succinctness. Comparing hyper-dual automata to the classic types, there is no size blowup in the translation of Rabin and Streett automata to a hyper-dual automaton (Proposition 2), while there is an exponential size blowup in the other direction when considering deterministic automata—for the translation to a Rabin automaton, we have the lower bound of deterministic Streett to Rabin, by considering two copies of the Streett automaton as a hyper-dual automaton, and analogously to the translation to Streett (Table 1).

Proposition 7. *There is a $2^{\omega(n \log n)}$ size blowup in the translation of deterministic hyper-dual automaton to deterministic Rabin and Streett automata.*

An upper bound for translating a deterministic hyper-dual automaton to deterministic Rabin and Streett automata involves a $2^{O(n^4 \log n)}$ size blowup:

We can consider a hyper-Rabin automaton of size n with n disjuncts in its acceptance condition as the union of n deterministic Streett automata of size n, which can then be viewed as a single nondeterministic Streett automaton of size n^2. A Streett automaton of size m can be translated to deterministic Rabin and Streett automata of size $2^{O(m^2 \log m)}$ (Table 1), providing a total size blowup of $2^{O(n^4 \log n)}$.

In the nondeterministic setting there is an exponential blowup in the translation to a Rabin automaton, due to the lower bound in the translation of a nondeterministic Streett automaton to a nondeterministic Rabin automaton (Table 1), while the translation to a Streett automaton only involves a quadratic size blowup, as a hyper-Rabin automaton with n disjuncts can be viewed as the union of n Streett automata.

Usage and decision procedures. Equipped with both a hyper-Rabin automaton and a hyper-Streett automaton, we can "enjoy both worlds", by choosing which of them to use for each task. As a result, using them for synthesis, model-checking probabilistic automata, or game solving is not more difficult than using a hyper-Rabin or hyper-Streett automaton, and all decision problems in the deterministic setting have polynomial-time algorithms. In the nondeterministic setting, the decision problems are roughly as for hyper-Streett automata due to the quadratic translation of hyper-Rabin to hyper-Streett.

Proposition 8. *Nonemptiness and universality checks of a deterministic hyper-dual automaton, as well as the containment and equivalence problems of two deterministic hyper-dual automata, are in PTIME.*

Proof. Consider a hyper-dual automaton $\mathcal{C} = (\mathcal{A}, \mathcal{B})$. Then $\mathcal{C}$ is empty iff $\mathcal{A}$ is, and it is universal iff $\overline{\mathcal{B}}$ is empty, which reduces to checking the (non-)emptiness of hyper-Rabin automata, which is in PTIME (Table 3).

Consider two hyper-dual automata $\mathcal{C}' = (\mathcal{A}', \mathcal{B}')$ and $\mathcal{C}'' = (\mathcal{A}'', \mathcal{B}'')$. Then $L(\mathcal{C}') \subseteq L(\mathcal{C}'')$ iff $L(\mathcal{A}') \cap \overline{L(\mathcal{B}'')} = \emptyset$. Observe that $\overline{L(\mathcal{B}'')} = L(\overline{\mathcal{B}''})$, and that $\overline{\mathcal{B}''}$ is a hyper-Rabin automaton. Thus, checking whether $L(\mathcal{A}') \subseteq L(\mathcal{B}'')$ is in PTIME: constructing $\mathcal{A}' \cap \overline{\mathcal{B}''}$ is possible in quadratic time (Table 2), and its emptiness check in PTIME (Table 3).

For checking whether $L(\mathcal{C}'') \subseteq L(\mathcal{C}')$, we can analogously consider $\mathcal{A}'' \cap \overline{\mathcal{B}'}$.

6 Conclusions

Automata on infinite words enjoy a variety of acceptance conditions, which are indeed necessary due to the richness of ω-regular languages and their connection to various kinds of other formalisms and logics. In the deterministic setting, which has recently become very relevant, it seems that there is still place for new acceptance conditions. In particular, when the automata are to be involved in positive boolean operations, one may consider the hyper-Rabin condition, and when complementations are also in place, one may consider the hyper-dual type.

References

1. T. Babiak, F. Blahoudek, A. Duret-Lutz, J. Klein, J. Křetínský, D. Müller, D. Parker, J. Strejček, and C. S. Păsăreanu. The Hanoi omega-automata format. In *Proc. of CAV*, pages 479–486, 2015.
2. F. Blahoudek. Translation of an LTL fragment to deterministic Rabin and Streett automata. Master's thesis, Masarykova Univerzita, 2012.
3. U. Boker. Word-automata translations, 2010–. URL: `http://www.faculty.idc.ac.il/udiboker/automata`.
4. U. Boker. On the (in)succinctness of Muller automata. In *CSL*, pages 12:1–12:16, 2017.
5. U. Boker. Rabin vs. Streett automata. In *FSTTCS*, pages 17:1–17:15, 2017.
6. U. Boker. Why these automata types? In *Proceedings of LPAR*, pages 143–163, 2018.
7. U. Boker, D. Kuperberg, O. Kupferman, and M. Skrzypczak. Nondeterminism in the presence of a diverse or unknown future. In *Proc. of ICALP*, pages 89–100, 2013.
8. U. Boker and O. Kupferman. Translating to co-Büchi made tight, unified, and useful. *ACM Trans. Comput. Log.*, 13(4):29:1–29:26, 2012.
9. U. Boker, O. Kupferman, and M. Skrzypczak. How deterministic are good-for-games automata? In *Proceedings of FSTTCS*, pages 18:1–18:14, 2017.
10. Y. Cai and T. Zhang. A tight lower bound for Streett complementation. In *Proceedings of FSTTCS*, pages 339–350, 2011.
11. Y. Cai and T. Zhang. Tight upper bounds for Streett and parity complementation. In *Proceedings of CSL*, pages 112–128, 2011.
12. Y. Cai, T. Zhang, and H. Luo. An improved lower bound for the complementation of Rabin automata. In *Proceedings of LICS*, pages 167–176, 2009.
13. K. Chatterjee, A. Gaiser, and J. Křetínský. Automata with generalized Rabin pairs for probabilistic model checking and LTL synthesis. In *Proc. of CAV*, pages 559–575, 2013.
14. Y. Choueka. Theories of automata on ω-tapes: A simplified approach. *Journal of Computer and Systems Science*, 8:117–141, 1974.
15. E. Clarke, I. Browne, and R. Kurshan. A unified approach for showing language containment and equivalence between various types of ω-automata. In *CAAP '90*, pages 103–116, 1990.
16. E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languagues and Systems*, 8(2):244–263, 1986.
17. T. Colcombet and K. Zdanowski. A tight lower bound for determinization of transition labeled Büchi automata. In *Proceedings of ICALP*, pages 151–162, 2009.
18. E.A. Emerson and C.-L. Lei. Modalities for model checking: Branching time logic strikes back. *Science of Computer Programming*, 8:275–306, 1987.
19. J. Esparza, J. Křetínský, and S. Sickert. From LTL to deterministic automata: A Safraless compositional approach. *Formal Methods in System Design*, 49(3):219–271, 12 2016.
20. E. Filiot, R. Gentilini, and J. F. Raskin. Rational synthesis under imperfect information. In *Proceedings of LICS*, pages 422–431, 2018.
21. T.A. Henzinger and N. Piterman. Solving games without determinization. In *Proc. of CSL*, volume 4207 of *LNCS*, pages 394–410. Springer, 2006.

22. P. Hunter and A. Dawar. Complexity bounds for regular games. In *Proc. of MFCS*, pages 495–506, 2005.
23. V. King, O. Kupferman, and M.Y. Vardi. On the complexity of parity word automata. In *Proc. of FoSSaCS*, pages 276–286, 2001.
24. J. Kretínský and J. Esparza. Deterministic automata for the (F,G)-fragment of LTL. In *Proc. of CAV*, volume 7358 of *LNCS*, pages 7–22. Springer, 2012.
25. O. Kupferman, G. Morgenstern, and A. Murano. Typeness for ω-regular automata. In *Proc. of ATVA*, volume 3299 of *LNCS*, pages 324–338. Springer, 2004.
26. O. Kupferman and M.Y. Vardi. Complementation constructions for nondeterministic automata on infinite words. In *Proc. of TACAS*, pages 206–221, 2005.
27. W. Liu and J. Wang. A tighter analysis of Piterman's Büchi determinization. *Inf. Process. Lett.*, 109(16):941–945, 2009.
28. C. Löding. Optimal bounds for the transformation of omega-automata. In *Proceedings of FSTTCS*, volume 1738 of *LNCS*, pages 97–109, 1999.
29. C. Löding and H. Yue. Memory bounds for winning strategies in infinite games, 2008, unpublished.
30. M. Michel. Complementation is more difficult with automata on infinite words. CNET, Paris, 1988.
31. S. Miyano and T. Hayashi. Alternating finite automata on ω-words. *Theoretical Computer Science*, 32:321–330, 1984.
32. N. Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. *Logical Methods in Computer Science*, 3(3):5, 2007.
33. S. Safra. *Complexity of automata on infinite objects*. PhD thesis, Weizmann Institute of Science, 1989.
34. S. Safra. Exponential determinization for ω-automata with strong-fairness acceptance condition. In *Proc. 24th ACM Symp. on Theory of Computing*, 1992.
35. S. Safra and M.Y. Vardi. On ω-automata and temporal logic. In *Proc. 21st ACM Symp. on Theory of Computing*, pages 127–137, 1989.
36. S. Schewe. Büchi complementation made tight. In *Proc. 26th Symp. on Theoretical Aspects of Computer Science*, volume 3 of *LIPIcs*, pages 661–672, 2009.
37. S. Schewe and T. Varghese. Determinising parity automata. In *Proceedings of MFCS*, pages 486–498, 2014.
38. M. Vardi. The Büchi complementation saga. In *Proceedings of STACS*, pages 12–22, 2007.
39. M.Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proceedings of FOCS*, pages 327–338, 1985.
40. M.Y. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, 1994.
41. Qiqi Yan. Lower bounds for complementation of omega-automata via the full automata technique. *Logical Methods in Computer Science*, 4(1), 2008.