Co-Büching Them All

U. Boker and O. Kupferman

School of Computer Science and Engineering
Hebrew University, Israel
{udiboker,orna}@cs.huji.ac.il

Abstract. We solve the open problems of translating, when possible, all common classes of nondeterministic word automata to deterministic and nondeterministic co-Büchi word automata. The handled classes include Büchi, parity, Rabin, Streett and Muller automata. The translations follow a unified approach and are all asymptotically tight.

The problem of translating Büchi automata to equivalent co-Büchi automata was solved in [2], leaving open the problems of translating automata with richer acceptance conditions. For these classes, one cannot easily extend or use the construction in [2]. In particular, going via an intermediate Büchi automaton is not optimal and might involve a blow-up exponentially higher than the known lower bound. Other known translations are also not optimal and involve a doubly exponential blow-up.

We describe direct, simple, and asymptotically tight constructions, involving a $2^{\Theta(n)}$ blow-up. The constructions are variants of the subset construction, and allow for symbolic implementations. Beyond the theoretical importance of the results, the new constructions have various applications, among which is an improved algorithm for translating, when possible, LTL formulas to deterministic Büchi word automata.

1 Introduction

Finite *automata on infinite objects* are widely used in formal verification and synthesis of nonterminating systems. The automata-theoretic approach to verification reduces questions about systems and their specifications to automata-theoretic problems like language containment and emptiness [10, 18]. Recent industrial-strength specification-languages such as Sugar, ForSpec and PSL 1.01 include regular expressions and/or automata, making automata-theory even more essential and popular [1].

There are various classes of automata, characterized by their branching mode and acceptance condition. Each class has its advantages, disadvantages, and common usages. Accordingly, an important challenge in the the study of automata on infinite objects is to provide algorithms for translating between the different classes. For most translations, our community was able to come up with satisfactory solutions, in the sense that the state blow-up involved in the algorithm is proved to be unavoidable. Yet, for some translations there is still a significant gap between the best known algorithm and the corresponding lower bound.

Among these open problems are the translations of nondeterministic automata to equivalent deterministic and nondeterministic co-Büchi automata (NCW and DCW),

when possible. [1] In [2], we introduced the *augmented subset construction* and used it for translating a nondeterministic Büchi automaton (NBW) to NCW and DCW, when possible. We left open the problems of translating automata with richer acceptance conditions (parity, Rabin, Streett and Muller) to co-Büchi automata. For these classes, one cannot easily extend or use the construction in [2], and the gap between the lower and upper bounds is still significant (for some of the classes it is even exponential). In this paper, we solve these problems and study the translation of nondeterministic parity (NPW), Streett (NSW), Rabin (NRW), and Muller (NMW) word automata to NCW and to DCW.

A straightforward approach is to translate an automaton of the richer classes via an intermediate NBW. This approach, however, is not optimal. For example, starting with an NSW with n states and index k, the intermediate NBW has $n2^k$ states, thus the NCW would have $n2^{k+n2^k}$ states, making the dependency in k doubly-exponential. Note that the exponential blow-up in the translation of NSW or NMW to NBW cannot be avoided [15]. A different approach is to translate the original automaton, for example an NRW, to an equivalent DPW, which can then be translated to an equivalent DCW over the same structure [5]. However, translating an NRW to an equivalent DPW might be doubly exponential [4], with no matching lower bound, even for the problem of translating to a DCW, let alone translating to NCW.

Thus, the approaches that go via intermediate automata are far from optimal, and our goal is to find a direct translation of these stronger classes of automata to NCW and DCW. We first show that for NSW, an equivalent NCW can be defined on top of the augmented subset construction (the product of the original automaton with its subset construction). The definition of the corresponding co-Büchi acceptance condition is more involved in this case than in the case of translating an NBW, but the blow-up stays the same. Thus, even though NSW are exponentially more succinct than NBW, their translation to NCW is of exactly the same state complexity as is the one for NBW! This immediately provides an $n2^n$ upper bound for the translation of NSW to NCW. As in the case of translating an NBW, we can further determinize the resulting augmented subset construction, getting a 3^n upper bound for the translation of NSW to DCW. Both bounds are asymptotically tight, having matching lower bounds by the special cases of translating NBW to NCW [2] and NCW to DCW [3]. The above good news apply also to the parity and the generalized-Büchi acceptance conditions, as they are special cases of the Streett condition.

For NRW and NMW, the situation is more complicated. Unfortunately, an equivalent NCW cannot in general be defined on top of the augmented subset construction. Moreover, even though the results on NSW imply a translation of NRW[1] (that is, a nondeterministic Rabin automaton with a single pair) to NCW, one cannot hope to proceed via a decomposition of an NRW with index k to k NRW[1]s. Indeed, the underlying NRW[1]s may not be NCW-realizable, even when the NRW is, and the same for NMWs. We show that still, the NCW can be defined on top of k copies of the augmented subset construction, giving rise to a $kn2^n$ upper bound for the translation to NCW. Moreover, we show that when translating to an equivalent DCW, the k copies

[1] The co-Büchi condition is weaker than the Büchi acceptance condition, and not all ω-regular languages are NCW-recognizable, hence the "when possible".

can be determinized separately, while connected in a round-robin fashion, which gives rise to a $k3^n$ blow-up. As with the other cases, the blow-up involved in the translations is asymptotically tight. The state blow-up involved in the various translations is summarized in Table 1 of the Section 6.

Beyond the theoretical challenge in tightening the gaps, and the fact they are related to other gaps in our knowledge [6], these translations have immediate important applications in formal methods. The interest in the co-Büchi condition follows from its simplicity and its duality to the Büchi acceptance condition. The interest in the stronger acceptance conditions follows from their richness and succinctness. In particular, standard translations of LTL to automata go via intermediate generalized Büchi automata, which are then being translated to Büchi automata. For some algorithms, it is possible to give up the last step and work directly with the generalized Büchi automaton [8]. It follows from our results that the same can be done with the algorithm of translating LTL formulas to NCW and DCW. By the duality of the co-Büchi and Büchi conditions, one can construct a DBW for ψ by dualizing the DCW for $\neg\psi$. Thus, since the translation of LTL to NSW may be exponentially more succinct than a translation to NBW, our construction suggests the best known translation of LTL to DBW, when exists.

An important and useful property of our constructions is the fact they have only a one-sided error when applied to automata whose language is not NCW-recognizable. Thus, given an automaton $\mathcal{A}$, the NCW $\mathcal{C}$ and the DCW $\mathcal{D}$ we construct are always such that $L(\mathcal{A}) \subseteq L(\mathcal{C}) = L(\mathcal{D})$, while $L(\mathcal{A}) = L(\mathcal{C}) = L(\mathcal{D})$ in case $\mathcal{A}$ is NCW-recognizable. Likewise, given an LTL formula ψ, the DBW $\mathcal{D}_\psi$ we construct is always such that $L(\mathcal{D}_\psi) \subseteq L(\psi)$, while $L(\mathcal{D}_\psi) = L(\psi)$ in case ψ is DBW-recognizable. As specified in Section 5, this enables us to extend the scope of the applications also to specifications that are not NCW-realizable.

2 Preliminaries

Given an alphabet Σ, a *word* over Σ is a (possibly infinite) sequence $w = w_1 \cdot w_2 \cdots$ of letters in Σ. For two words, x and y, we use $x \preceq y$ to indicate that x is a prefix of y and $x \prec y$ to indicate that x is a strict prefix of y. An *automaton* is a tuple $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$, where Σ is the input alphabet, Q is a finite set of states, $\delta : Q \times \Sigma \to 2^Q$ is a transition function, $Q_0 \subseteq Q$ is a set of initial states, and α is an acceptance condition. 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 σ. 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 $|\delta(q, \sigma)| \leq 1$, we say that $\mathcal{A}$ is *deterministic*. The transition function extends to sets of states and to finite words in the expected way, thus for a set of states S and a finite word x, $\delta(S, x)$ is the set of states that $\mathcal{A}$ may move into when it is in a state in S and it reads x. Formally, $\delta(S, \epsilon) = S$ and $\delta(S, w \cdot \sigma) = \bigcup_{q \in \delta(S,w)} \delta(q, \sigma)$. We abbreviate $\delta(Q_0, x)$ by $\delta(x)$, thus $\delta(x)$ is the set of states that $\mathcal{A}$ may visit after reading x. For an automaton $\mathcal{A}$ and a state q of $\mathcal{A}$, we denote by $\mathcal{A}^q$ the automaton that is identical to

$\mathcal{A}$, except for having $\{q\}$ as its set of initial states. An automaton without an acceptance condition is called a *semi-automaton*.

A run $r = r_0, r_1, \cdots$ of $\mathcal{A}$ on $w = w_1 \cdot w_2 \cdots \in \Sigma^\omega$ is an infinite sequence of states such that $r_0 \in Q_0$, and for every $i \geq 0$, we have that $r_{i+1} \in \delta(r_i, w_{i+1})$. Note that while a deterministic automaton has at most a single run on an input word, a nondeterministic automaton may have several runs on an input word. We sometimes refer to r as a word in Q^ω or as a function from the set of prefixes of w to the states of $\mathcal{A}$. Accordingly, we use $r(x)$ to denote the state that r visits after reading the prefix x.

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

Several acceptance conditions are studied in the literature. We consider here six:

- *Büchi*, where $\alpha \subseteq Q$, and r is accepting iff $inf(r) \cap \alpha \neq \emptyset$.
- *co-Büchi*, where $\alpha \subseteq Q$, and r is accepting iff $inf(r) \subseteq \alpha$. Note that the definition we use is less standard than the $inf(r) \cap \alpha = \emptyset$ definition; clearly, $inf(r) \subseteq \alpha$ iff $inf(r) \cap (Q \setminus \alpha) = \emptyset$, thus the definitions are equivalent. We chose to go with this variant as it better conveys the intuition that, as with the Büchi condition, a visit in α is a "good event".
- *parity*, where $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_{2k}\}$ with $\alpha_1 \subset \alpha_2 \subset \cdots \subset \alpha_{2k} = Q$, and r is accepting if the minimal index i for which $inf(r) \cap \alpha_i \neq \emptyset$ is even.
- *Rabin*, where $\alpha = \{\langle \alpha_1, \beta_1 \rangle, \langle \alpha_2, \beta_2 \rangle, \ldots, \langle \alpha_k, \beta_k \rangle\}$, with $\alpha_i, \beta_i \subseteq Q$ and r is accepting iff for some $1 \leq i \leq k$, we have that $inf(r) \cap \alpha_i \neq \emptyset$ and $inf(r) \cap \beta_i = \emptyset$.
- *Streett*, where $\alpha = \{\langle \beta_1, \alpha_1 \rangle, \langle \beta_2, \alpha_2 \rangle, \ldots, \langle \beta_k, \alpha_k \rangle\}$, with $\beta_i, \alpha_i \subseteq Q$ and r is accepting iff for all $1 \leq i \leq k$, we have that $inf(r) \cap \beta_i = \emptyset$ or $inf(r) \cap \alpha_i \neq \emptyset$.
- *Muller*, where $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_k\}$, with $\alpha_i \subseteq Q$ and r is accepting iff for some $1 \leq i \leq k$, we have that $inf(r) = \alpha_i$.

The number of sets in the parity and Muller acceptance conditions or pairs in the Rabin and Streett acceptance conditions is called the *index* of the automaton. An automaton accepts a word if it has an accepting run on it. The language of an automaton $\mathcal{A}$, denoted $L(\mathcal{A})$, is the set of words that $\mathcal{A}$ accepts. We also say that $\mathcal{A}$ *recognizes* the language $L(\mathcal{A})$. For two automata $\mathcal{A}$ and $\mathcal{A}'$, we say that $\mathcal{A}$ and $\mathcal{A}'$ are *equivalent* if $L(\mathcal{A}) = L(\mathcal{A}')$.

We denote the different classes of automata by three letter acronyms in $\{\mathrm{D}, \mathrm{N}\}$ $\times$ {B, C, P, R, S, M} $\times$ {W}. The first letter stands for the branching mode of the automaton (deterministic or nondeterministic); the second letter stands for the acceptance-condition type (Büchi, co-Büchi, parity, Rabin, Streett, or Muller); and the third letter indicates that the automaton runs on words. We say that a language L is γ-*recognizable* or γ-*realizable* if L can be recognized by an automaton in the class γ.

Different classes of automata have different expressive power. In particular, while NBWs recognize all ω-regular languages [12], DBWs are strictly less expressive than NBWs, and so are DCWs [11]. In fact, a language L is in DBW iff its complement is in DCW. Indeed, by viewing a DBW as a DCW and switching between accepting and non-accepting states, we get an automaton for the complementing language, and vice

versa. The expressiveness superiority of the nondeterministic model over the deterministic one does not apply to the co-Büchi acceptance condition. There, every NCW has an equivalent DCW [13]. As for parity, Rabin, Streett and Muller automata, both the deterministic and nondeterministic models recognize all ω-regular languages [17].

Our constructions for translating the various automata to co-Büchi automata will use the *augmented subset construction* [2], which is the product of an automaton with its subset construction.

Definition 1 (Augmented subset construction). *[2] Let $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0 \rangle$ be a semi-automaton. The* augmented subset construction *$\mathcal{A}'$ of $\mathcal{A}$ is the product of $\mathcal{A}$ with its subset construction. Formally, $\mathcal{A}' = \langle \Sigma, Q', \delta', Q'_0 \rangle$, where*

- *$Q' = Q \times 2^Q$. That is, the states of $\mathcal{A}'$ are all the pairs $\langle q, E \rangle$ where $q \in Q$ and $E \subseteq Q$.*
- *For all $\langle q, E \rangle \in Q'$ and $\sigma \in \Sigma$, we have $\delta'(\langle q, E \rangle, \sigma) = \delta(q, \sigma) \times \{\delta(E, \sigma)\}$. That is, $\mathcal{A}'$ nondeterministically follows $\mathcal{A}$ on its Q-component and deterministically follows the subset construction of $\mathcal{A}$ on its 2^Q-component.*
- *$Q'_0 = Q_0 \times \{Q_0\}$.*

3 Translating to NCW

In this section we study the translation, when possible, of NPWs, NRWs, NSWs, and NMWs to NCWs. Since the Büchi acceptance condition is a special case of these stronger conditions, the $2^{\Omega(n)}$ lower bound from [2] applies, and the challenge is to come up with matching upper bounds. While nondeterministic Rabin, Streett, and Muller automata are not more expressive than nondeterministic Büchi automata, they are more succinct: translating an NRW, NSW, and NMW with n states and index k to an NBW, results in an NBW with $O(nk)$, $O(n2^k)$, and $O(n^2k)$ states, respectively [15, 16]. Note that an NPW is a special case of both an NSW and an NRW.

A first attempt to translate NRWs, NSWs, and NMWs to NCWs is to go via intermediate NBWs, which can be translated to NCWs by the augmented subset construction [2]. By the blow-ups above, however, this results in NCWs that are far from optimal. A second attempt is to apply the augmented subset construction directly on the input automaton, and check the possibility of defining on top of it a suitable co-Büchi acceptance condition.

It is not hard to see that this second attempt does not work for all automata. Consider for example the Rabin acceptance condition. Note that the augmented subset construction does not alter a deterministic automaton. Also, DRWs are not DCW-type [7] (that is, there is a DRW $\mathcal{A}$ whose language is DCW-recognizable, but still no DCW equivalent to $\mathcal{A}$ can be defined on top of the structure of $\mathcal{A}$). It follows that there are NRWs whose language is NCW-recognizable, but still no NCW recognizing them can be defined on top of the automaton obtained by applying the augmented subset construction on them (see Theorem 2 for a concrete example).

With this in mind, this section is a collection of good news. First, we show in Subsection 3.1 that NSWs (and NPWs) can be translated to NCWs on top of the augmented subset construction. Second, while this is not valid for NRWs and NMWs, we show

in Subsection 3.2 that they can be translated to NCWs on top of a union of copies of the augmented subset construction. Moreover, the translation of the obtained NCWs to equivalent DCWs does not involve an additional exponential blow-up (see Section 4).

We first provide some basic lemmata from [2]. We start with a property relating states of a DCW (in fact, any deterministic automaton) that are reachable via words that lead to the same state in the subset construction of an equivalent nondeterministic automaton.

Lemma 1. *[2] Consider a nondeterministic automaton $\mathcal{A}$ with a transition function $\delta_{\mathcal{A}}$ and a DCW $\mathcal{D}$ with a transition function $\delta_{\mathcal{D}}$ such that $L(\mathcal{A}) = L(\mathcal{D})$. Let d_1 and d_2 be states of $\mathcal{D}$ such that there are two finite words x_1 and x_2 such that $\delta_{\mathcal{D}}(x_1) = d_1$, $\delta_{\mathcal{D}}(x_2) = d_2$, and $\delta_{\mathcal{A}}(x_1) = \delta_{\mathcal{A}}(x_2)$. Then, $L(\mathcal{D}^{d_1}) = L(\mathcal{D}^{d_2})$.*

For automata on finite words, if two states of the automaton have the same language, they can be merged without changing the language of the automaton. While this is not the case for automata on infinite words, the lemma below enables us to do take advantage of such states.

Lemma 2. *[2] Consider a DCW $\mathcal{D} = \langle \Sigma, D, \delta, D_0, \alpha \rangle$. Let d_1 and d_2 be states in D such that $L(\mathcal{D}^{d_1}) = L(\mathcal{D}^{d_2})$. For all finite words u and v, if $\delta(d_1, u) = d_1$ and $\delta(d_2, v) = d_2$ then for all words $w \in (u + v)^*$ and states $d \in \delta(d_1, w) \cup \delta(d_2, w)$, we have $L(\mathcal{D}^d) = L(\mathcal{D}^{d_1})$.*

The next lemma takes further advantage of DCW states recognizing the same language.

Lemma 3. *[2] Let $\mathcal{D} = \langle \Sigma, D, \delta, D_0, \alpha \rangle$ be a DCW. Consider a state $d \in D$. For all nonempty finite words v and u, if $(v^* \cdot u^+)^\omega \subseteq L(\mathcal{D}^d)$ and for all words $w \in (v + u)^*$ and states $d' \in \delta(d, w)$, we have $L(\mathcal{D}^{d'}) = L(\mathcal{D}^d)$, then $v^\omega \in L(\mathcal{D}^d)$.*

3.1 From NSW to NCW

The translation of an NSW to an NCW, when exists, can be done on top of the augmented subset construction, generalizing the acceptance condition used for translating an NBW to an NCW.

In the translation of an NBW to an NCW, we start with an NBW $\mathcal{B}$ and define a state $\langle b, E \rangle$ of the augmented subset construction to be co-Büchi accepting if there is some path p in $\mathcal{B}$, taking $\langle b, E \rangle$ back to itself via a Büchi accepting state. The correctness of the construction follows from the fact that an NCW-recognizable language is closed under pumping such cycles. Thus, if $\mathcal{B}$ accepts a word that includes a subword along which p is read, then $\mathcal{B}$ also accepts words obtained by pumping the subword along which p is read. In turns out that this intuition is valid also when we start with an NSW $\mathcal{S}$: a state $\langle s, E \rangle$ of the augmented subset construction is co-Büchi accepting if there is some path p in $\mathcal{S}$, taking $\langle s, E \rangle$ back to itself, such that p visits α_i or avoid β_i for every pair i in the Streett acceptance condition. This guarantees that pumping p infinitely often results in a run that satisfies the Streett condition, which in turn implies that an NCW-recongnizable language is closed under such pumping.

We formalize and prove this idea below.

Theorem 1. *For every NSW $\mathcal{S}$ with n states that is NCW-recognizable, there is an equivalent NCW $\mathcal{C}$ with at most $n2^n$ states.*

Proof. Let $\mathcal{S} = \langle \Sigma, S, \delta_{\mathcal{S}}, S_0, \langle \beta_1, \alpha_1 \rangle, \ldots \langle \beta_k, \alpha_k \rangle \rangle$. We define the NCW $\mathcal{C} = \langle \Sigma, C, \delta_{\mathcal{C}}, C_0, \alpha_{\mathcal{C}} \rangle$ as the augmented subset construction of $\mathcal{S}$ with the following acceptance condition: a state is a member of $\alpha_{\mathcal{C}}$ if it is reachable from itself along a path whose projection on S visits α_i or avoids β_i for every $1 \leq i \leq k$.

Formally, $\langle s, E \rangle \in \alpha_{\mathcal{C}}$ if there is a finite word $z = z_1 z_2 \cdots z_m$ of length m and a sequence of $m+1$ states $\langle s_0, E_0 \rangle \ldots \langle s_m, E_m \rangle$ such that $\langle s_0, E_0 \rangle = \langle s_m, E_m \rangle = \langle s, E \rangle$, and for all $0 \leq l < m$ we have $\langle s_{l+1}, E_{l+1} \rangle \in \delta_C(\langle s_l, E_l \rangle, z_{l+1})$, and for every $1 \leq i \leq k$, either there is $0 \leq l < m$ such that $s_l \in \alpha_i$ or $s_l \notin \beta_i$ for all $0 \leq l < m$. We refer to z as the *witness* for $\langle s, E \rangle$. Note that z may be the empty word.

We prove the equivalence of $\mathcal{S}$ and $\mathcal{C}$. Note that the 2^S-component of C proceeds in a deterministic manner. Therefore, each run r of $\mathcal{S}$ induces a single run of $\mathcal{C}$ (the run in which the S-component follows r). Likewise, each run r of $\mathcal{C}$ induces a single run of $\mathcal{S}$, obtained by projecting r on its S-component.

We first prove that $L(\mathcal{S}) \subseteq L(\mathcal{C})$. Note that this direction is always valid, even if $\mathcal{S}$ is not NCW-recognizable. Consider a word $w \in L(\mathcal{S})$. Let r be an accepting run of $\mathcal{S}$ on w. We prove that the run r' induced by r is accepting. Let $J \subseteq \{1, \ldots, k\}$ denote the set of indices of acceptance-pairs whose β-element is visited infinitely often by r. That is, $J = \{j \mid \beta_j \cap inf(r) \neq \emptyset\}$. Consider a state $\langle s, E \rangle \in inf(r')$. We prove that $\langle s, E \rangle \in \alpha_{\mathcal{C}}$. Since $\langle s, E \rangle$ appears infinitely often in r' and r is accepting, it follows that there are two (not necessarily adjacent) occurrences of $\langle s, E \rangle$, between which r visits α_j for all $j \in J$ and avoids β_i for all $i \notin J$. Hence, we have the required witness for $\langle s, E \rangle$, and we are done.

We now prove that $L(\mathcal{C}) \subseteq L(\mathcal{S})$. Consider a word $w \in L(\mathcal{C})$, and let r be an accepting run of $\mathcal{C}$ on w. Let $J \subseteq \{1, \ldots, k\}$ denote the set of indices of acceptance-pairs whose β-element is visited infinitely often by r. That is, $J = \{j \mid (\beta_j \times 2^S) \cap inf(r) \neq \emptyset\}$. If J is empty then the projection of r on its $\mathcal{S}$-component is accepting, and we are done. Otherwise, we proceed as follows. For every $j \in J$, let $\langle s_j, E_j \rangle$ be a state in $(\beta_j \times 2^S) \cap inf(r)$.

By the definition of J, all the states $\langle s_j, E_j \rangle$, with $j \in J$, are visited infinitely often in r, whereas states whose $\mathcal{S}$-component is in β_i, for $i \notin J$, are visited only finitely often in r. Accordingly, the states $\langle s_j, E_j \rangle$, with $j \in J$, are strongly connected via a path that does not visit β_i, for $i \notin J$. In addition, for every $\langle s_j, E_j \rangle$, with $j \in J$, there is a witness z_j for the membership of $\langle s_j, E_j \rangle$ in $\alpha_{\mathcal{C}}$, going from $\langle s_j, E_j \rangle$ back to itself via α_j and either avoiding β_i or visiting α_i, for every $1 \leq i \leq k$. Let $\langle s, E \rangle$ be one of these $\langle s_j, E_j \rangle$ states, and let x be a prefix of w such that $r(x) = \langle s, E \rangle$. Then, there is a finite word z along which there is a path from $\langle s, E \rangle$ back to itself, visiting all α_j for $j \in J$ and either avoiding β_i or visiting α_i for every $1 \leq i \leq k$. Therefore, $x \cdot z^\omega \in L(\mathcal{S})$.

Recall that the language of $\mathcal{S}$ is NCW-recognizable. Let $\mathcal{D} = \langle \Sigma, D, \delta_D, D_0, \alpha_{\mathcal{D}} \rangle$ be a DCW equivalent to $\mathcal{S}$. Since $L(\mathcal{S}) = L(\mathcal{D})$ and $x \cdot z^\omega \in L(\mathcal{S})$, it follows that the run ρ of $\mathcal{D}$ on $x \cdot z^\omega$ is accepting. Since D is finite, there are two indices, l and m, such that $l < m$, $\rho(x \cdot z^l) = \rho(x \cdot z^m)$, and for all prefixes y of $x \cdot z^\omega$ such that $x \cdot z^l \preceq y$, we have $\rho(y) \in \alpha_{\mathcal{D}}$. Let q be the state of $\mathcal{D}$ such that $q = \rho(x \cdot z^l)$.

Consider the run η of $\mathcal{D}$ on w. Since r visits $\langle s, E\rangle$ infinitely often and D is finite, there must be a state $d \in D$ and infinitely many prefixes $p_1, p_2, \ldots$ of w such that for all $i \geq 1$, we have $r(p_i) = \langle s, E\rangle$ and $\eta(p_i) = d$. We claim that the states q and d of $\mathcal{D}$ satisfy the conditions of Lemma 1 with x_0 being p_1 and x_1 being $x \cdot z^l$. Indeed, $\delta_{\mathcal{D}}(p_1) = d$, $\delta_{\mathcal{D}}(x \cdot z^l) = q$, and $\delta_{\mathcal{S}}(p_1) = \delta_{\mathcal{S}}(x \cdot z^l) = E$. For the latter equivalence, recall that $\delta_{\mathcal{S}}(x) = E$ and $\delta_{\mathcal{S}}(E, z) = E$. Hence, by Lemma 1, we have that $L(\mathcal{D}^q) = L(\mathcal{D}^d)$.

Recall the sequence of prefixes $p_1, p_2, \ldots$. For all $i \geq 1$, let $p_{i+1} = p_i \cdot t_i$. We now claim that for all $i \geq 1$, the state d satisfies the conditions of Lemma 3 with u being z^{m-l} and v being t_i. The second condition is satisfied by Lemma 2. For the first condition, consider a word $w' \in (v^* \cdot u^+)^\omega$. We prove that $w' \in L(\mathcal{D}^d)$. Recall that there is a run of $\mathcal{S}^s$ on v that goes back to s while avoiding β_i for all $i \notin J$ and there is a run of $\mathcal{S}^s$ on u that goes back to s while visiting α_j for all $j \in J$ and either visiting α_i or avoiding β_i for all $i \notin J$. (Informally, u "fixes" all the problems of v, by visiting α_j for every β_j that v might visit.) Recall also that for the word p_1, we have that $r(p_1) = \langle s, E\rangle$ and $\eta(p_1) = d$. Hence, $p_1 \cdot w' \in L(\mathcal{S})$. Since $L(\mathcal{S}) = L(\mathcal{D})$, we have that $p_1 \cdot w' \in L(\mathcal{S})$. Therefore, $w' \in L(\mathcal{D}^d)$.

Thus, by Lemma 3, for all $i \geq 1$ we have that $t_i^\omega \in L(\mathcal{D}^d)$. Since $\delta_{\mathcal{D}}(d, t_i) = d$, it follows that all the states that $\mathcal{D}$ visits when it reads t_i from d are in $\alpha_{\mathcal{D}}$. Note that $w = p_1 \cdot t_1 \cdot t_2 \cdots$. Hence, since $\delta_{\mathcal{D}}(p_1) = d$, the run of $\mathcal{D}$ on w is accepting, thus $w \in L(\mathcal{D})$. Since $L(\mathcal{D}) = L(\mathcal{S})$, it follows that $w \in L(\mathcal{S})$, and we are done. □

Two common special cases of the Streett acceptance condition are the parity and the *generalized Büchi* acceptance conditions. In a generalized Büchi automaton with states Q, the acceptance condition is $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_k\}$ with $\alpha_i \subseteq Q$, and a run r is accepting if $inf(r) \cap \alpha_i \neq \emptyset$ for all $1 \leq i \leq k$. Theorem 1 implies that an NCW-recognizable nondeterministic parity or generalized Büchi automaton with n states can be translated to an NCW with $n2^n$ states, which can be defined on top of the augmented subset construction.

3.2 From NRW and NMW to NCW

In this section we study the translation of NRWs and NMWs to NCWs, when exists. Unfortunately, for these automata classes we cannot define an equivalent NCW on top of the augmented subset construction. Intuitively, the key idea of Subsection 3.1, which is based on the ability to pump paths that satisfy the acceptance condition, is not valid in the Rabin and the Muller acceptance conditions, as in these conditions, visiting some "bad" states infinitely often need not be compensated by visiting some "good" ones infinitely often. We formalize this in the example below, which consists of the fact that DRWs are not DCW-type [7].

Theorem 2. *There is an NRW and an NMW that are NCW-recognizable but an equivalent NCW for them cannot be defined on top of the augmented subset construction.*

Proof. Consider the NRW $\mathcal{A}$ appearing in Figure 1. The language of $\mathcal{A}$ consists of all words over the alphabet $\{0, 1\}$ that either have finitely many 0's or have finitely

many 1's. This language is clearly NCW-recognizable, as it is the union of two NCW-recognizable languages. Since $\mathcal{A}$ is deterministic and the augmented subset construction does not alter a deterministic automaton, it suffices to show that there is no co-Büchi acceptance condition α' that we can define on the structure of $\mathcal{A}$ and get an equivalent language. Indeed, α' may either be $\emptyset$, $\{q_0\}$, $\{q_1\}$, or $\{q_0, q_1\}$, none of which provides the language of $\mathcal{A}$. Since every NRW has an equivalent NMW over the same structure, the above result also applies to the NMW case. □

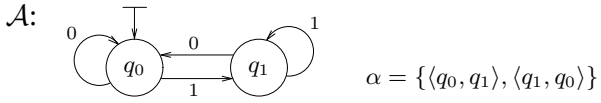

Fig. 1. The NRW $\mathcal{A}$, having no equivalent NCW on top of its augmented subset construction.

Consider an NRW or an NMW $\mathcal{A}$ with index k. Our approach for translating $\mathcal{A}$ to an NCW is to decompose it to k NSWs over the same structure, and apply the augmented subset construction on each of the components. Note that the components may not be NCW-realizable even when $\mathcal{A}$ is, thus, we should carefully analyze the proof of Theorem 1 and prove that the approach is valid.

We now formalize and prove the above approach. We start with the decomposition of an NRW or an NMW with index k into k NSWs over the same structure.

Lemma 4. *Every NRW or NMW $\mathcal{A}$ with index k is equivalent to the union of k NSWs over the same structure as $\mathcal{A}$.*

Proof. An NRW $\mathcal{A}$ with states A and index k is the union of k NRWs with index 1 over the same structure as $\mathcal{A}$. Since a single-indexed Rabin acceptance condition $\{\langle \alpha_1, \beta_1\rangle\}$ is equivalent to the Streett acceptance condition $\{\langle \alpha_1, \emptyset\rangle, \langle A, \beta_1\rangle\}$, we are done.

An NMW $\mathcal{A}$ with states A and index k is the union of k NMWs with index 1 over the same structure as $\mathcal{A}$. Since a single-indexed Muller acceptance condition $\{\alpha_1\}$ is equivalent to the Streett acceptance condition $\{\langle A \setminus \alpha_1, \emptyset\rangle\} \cup \bigcup_{s\in\alpha_1} \{\langle A, \{s\}\rangle\}$, we are done. □

Next we show that a union of k NSWs can be translated to a single NSW over their union.

Lemma 5. *Consider k NSWs, $\mathcal{S}_1, \ldots, \mathcal{S}_k$, over the same structure. There is an NSW $\mathcal{S}$ over the disjoint union of their structures, such that $L(\mathcal{S}) = \bigcup_{i=1}^{k} L(\mathcal{S}_i)$.*

Proof. We obtain the Streett acceptance condition of $\mathcal{S}$ by taking the union of the Streett acceptance conditions of the NSWs $\mathcal{S}_1, \ldots, \mathcal{S}_k$. Note that while the underlying NSWs are interpreted disjunctively (that is, in order for a word to be accepted by the union, there should be an accepting run on it in some $\mathcal{S}_i$), the pairs in the Streett condition are interpreted conjunctively (that is, in order for a run to be accepting, it has to satisfy the constraints by all the pairs in the Streett condition). We prove that still $L(\mathcal{S}) =$

$\bigcup_{i=1}^{k} L(\mathcal{S}_i)$. First, if a run r of $\mathcal{S}$ is an accepting run of an underlying NSW $\mathcal{S}_i$, then the acceptance conditions of the other underlying NSWs are vacuously satisfied. Hence, if a word is accepted by $\mathcal{S}_i$ for some $1 \leq i \leq k$, then $\mathcal{S}$ accepts it too. For the other direction, if a word w is accepted in $\mathcal{S}$, then its accepting run in $\mathcal{S}$ is also an accepting run of one of the underlying NSWs, thus w is in $\bigcup_{i=1}^{k} L(\mathcal{S}_i)$. □

Finally, we combine the translation to Streett automata with the augmented subset construction and get the required upper bound for NRW and NMW.

Theorem 3. *For every NCW-recognizable NRW or NMW with n states and index k, there is an equivalent NCW $\mathcal{C}$ with at most $kn2^n$ states.*

Proof. Consider an NRW or an NMW $\mathcal{A}$ with n states and index k. By Lemmas 4 and 5, there is an NSW $\mathcal{S}$ whose structure consists of k copies of the structure of $\mathcal{A}$ such that $L(\mathcal{S}) = L(\mathcal{A})$. Let $\mathcal{C}$ be the NCW equivalent to $\mathcal{S}$, defined over the augmented subset construction of $\mathcal{S}$, as described in Theorem 1. Note that $\mathcal{S}$ has nk states, thus a naive application of the augmented subset construction on it results in an NCW with $kn2^{kn}$ states. The key observation, which implies that we get an NCW with only $kn2^n$ states, is that applying the augmented subset construction on $\mathcal{S}$, the deterministic component of all the underlying NCWs is the same, and it coincides with the subset construction applied to $\mathcal{A}$. To see this, assume that $\mathcal{A} = \langle \Sigma, A, A_0, \delta, \alpha \rangle$. Then, $\mathcal{S} = \langle \Sigma, A \times \{1, \ldots, k\}, A_0 \times \{1, \ldots, k\}, \delta', \alpha' \rangle$, where for all $a \in A, 1 \leq j \leq k$, and $\sigma \in \Sigma$, we have that $\delta'(\langle a, j \rangle, \sigma) = \delta(a, \sigma) \times \{j\}$. Applying the augmented subset construction, we get the product of $\mathcal{S}$ and its subset construction, where the latter has a state for every reachable subset of S. That is, a subset $G' \subseteq S$ is a state of the subset construction if there is a finite word u for which $\delta'(u) = G'$. Since for all $a \in A, 1 \leq j \leq k$, and $\sigma \in \Sigma$, we have that $\delta'(\langle a, j \rangle, \sigma) = \delta(a, \sigma) \times \{j\}$, it follows that G' is of the form $G \times \{j\}$ for all $1 \leq j \leq k$ and some $G \subseteq A$. Hence, there are up to $2^{|A|} = 2^n$ states in the subset construction of $\mathcal{S}$. Thus, when we apply the augmented subset construction on $\mathcal{S}$, we end up with an NCW with only $kn2^n$ states, and we are done. □

4 Translating to DCW

In a first sight, the constructions of Section 3, which translate a nondeterministic word automaton to an NCW, are not useful for translating it to a DCW, as the determinization of an NCW to a DCW has an exponential state blow-up. Yet, we show that the special structure of the constructed NCW allows to determinize it without an additional exponential blow-up. The key to our construction is the observation that the augmented subset construction is transparent to additional applications of the subset construction. Indeed, applying the subset construction on an NCW $\mathcal{C}$ with state space $B \times 2^B$, one ends up in a deterministic automaton with state space $\{\{\langle q, E \rangle \mid q \in E\} : E \subseteq B\}$, which is isomorphic to 2^B.

The standard breakpoint construction [13] uses the subset construction as an intermediate layer in translating an NCW with state space C to a DCW with state space 3^C. Thus, the observation above suggests that applying it on our special NCW $\mathcal{C}$ would

not involve an additional exponential blow-up on top of the one involved in going from some automaton $\mathcal{A}$ to $\mathcal{C}$. As we show in Theorem 4 below, this is indeed the case.

Starting with an NSW, the determinization of the corresponding NCW is straightforward, following [13]'s construction. However, when starting with an NRW or an NMW, the k different parts of the corresponding NCW (see Theorem 3) might cause a doubly-exponential blowup. Fortunately, we can avoid it by determinizing each of the k parts separately and connecting them in a round-robin fashion. We refer to the construction in Theorem 4 as the *breakpoint construction*.

Theorem 4. *For every DCW-recognizable NPW, NSW, NRW, or NMW $\mathcal{A}$ with n states there is an equivalent DCW $\mathcal{D}$ with $O(3^n)$ states.*

Proof. We start with the case $\mathcal{A}$ is an NSW. The DCW $\mathcal{D}$ follows all the runs of the NCW $\mathcal{C}$ constructed in Theorem 1. Let $\alpha_{\mathcal{C}} \subseteq A \times 2^A$ be the acceptance condition of $\mathcal{C}$. The DCW $\mathcal{D}$ accepts a word if some run of $\mathcal{C}$ remains in $\alpha_{\mathcal{C}}$ from some position.[2] At each state, $\mathcal{D}$ keeps the corresponding subset of the states of $\mathcal{C}$, and it updates it deterministically whenever an input letter is read. In order to check that some run of $\mathcal{C}$ remains in $\alpha_{\mathcal{C}}$ from some position, the DCW $\mathcal{D}$ keeps track of runs that do not leave $\alpha_{\mathcal{C}}$. The key observation in [13] is that keeping track of such runs can be done by maintaining the subset of states that belong to these runs.

Formally, let $\mathcal{A} = \langle \Sigma, A, \delta_{\mathcal{A}}, A_0, \alpha_{\mathcal{A}} \rangle$. We define a function $f : 2^A \to 2^A$ by $f(E) = \{a \mid \langle a, E \rangle \in \alpha_{\mathcal{C}}\}$. Thus, when the subset component of $\mathcal{D}$ is in state E, it should continue and check the membership in $\alpha_{\mathcal{C}}$ only for states in $f(E)$. We define the DCW $\mathcal{D} = \langle \Sigma, D, \delta_{\mathcal{D}}, D_0, \alpha_{\mathcal{D}} \rangle$ as follows.

- $D = \{\langle S, O \rangle \mid S \subseteq A \text{ and } O \subseteq S \cap f(S)\}$.
- For all $\langle S, O \rangle \in D$ and $\sigma \in \Sigma$, the transition function is defined as follows.
  - If $O \neq \emptyset$, then $\delta_D(\langle S, O \rangle, \sigma) = \{\langle \delta_{\mathcal{A}}(S, \sigma), \delta_{\mathcal{A}}(O, \sigma) \cap f(S) \rangle\}$.
  - If $O = \emptyset$, then $\delta_D(\langle S, O \rangle, \sigma) = \{\langle \delta_{\mathcal{A}}(S, \sigma), \delta_{\mathcal{A}}(S, \sigma) \cap f(S) \rangle\}$.
- $D_0 = \{\langle A_0, \emptyset \rangle\}$.
- $\alpha_{\mathcal{D}} = \{\langle S, O \rangle \mid O \neq \emptyset\}$.

Thus, the run of $\mathcal{D}$ on a word w has to visit states in $2^A \times \{\emptyset\}$ only finitely often, which holds iff some run of $\mathcal{C}$ on w eventually always visits $\alpha_{\mathcal{C}}$. Since each state of D corresponds to a function from A to the set $\{$ "in $S \cap O$", "in $S \setminus O$", "not in S"$\}$, its number of states is at most $3^{|A|}$.

We proceed to the case $\mathcal{A}$ is an NRW or an NMW. Here, by Theorem 3, $\mathcal{A}$ has an equivalent NCW $\mathcal{C}$ with $kn2^n$ states. The NCW $\mathcal{C}$ is obtained by applying the augmented subset construction on k copies of $\mathcal{A}$, and thus has k unconnected components, $\mathcal{C}_1, \ldots, \mathcal{C}_k$ that are identical up to their acceptance conditions $\alpha_{\mathcal{C}_1}, \ldots, \alpha_{\mathcal{C}_k}$.

Since the k components of $\mathcal{C}$ all have the same $A \times 2^A$ structure, applying the standard subset construction on $\mathcal{C}$, one ends up with a deterministic automaton that is isomorphic to 2^A. Applying the standard breakpoint construction on $\mathcal{C}$, we could thus

[2] Readers familiar with the construction of [13] may find it easier to view the construction here as one that dualizes a translation of universal co-Büchi automata to deterministic Büchi automata, going through universal Büchi word automata – these constructed by dualizing Theorem 1.

hope to obtain a deterministic automaton with only $3^{|A|}$ states. This construction, however, has to consider the different acceptance conditions α_i, maintaining in each state not only a pair $\langle S, O\rangle$, but a tuple $\langle S, O_1, \ldots, O_k\rangle$, where each $O_i \subseteq S$ corresponds to the standard breakpoint construction with respect to α_i. Such a construction, however, involves a k^n blow-up.

We circumvent this blow-up by determinizing each of the $\mathcal{C}_i$'s separately and connecting the resulting $\mathcal{D}_i$'s in a round-robin fashion, moving from $\mathcal{D}_i$ to $\mathcal{D}_{i \pmod k+1}$ when the set O, which maintains the set of states in paths in which $\mathcal{D}_i$ avoids α_i, becomes empty. Now, there is $1 \leq i \leq k$ such that $\mathcal{C}_i$ has a run that eventually gets stuck in α_i iff there is $1 \leq i \leq k$ such that in the round-robin construction, the run gets stuck in a copy that corresponds to $\mathcal{D}_i$ in states with $O \neq \emptyset$.

Formally, for every $1 \leq i \leq k$, we define a function $f_i : 2^A \to 2^A$ by $f_i(E) = \{a \mid \langle a, E\rangle \in \alpha_{\mathcal{C}_i}\}$. We define the DCW $\mathcal{D} = \langle \Sigma, D, \delta_{\mathcal{D}}, D_0, \alpha_{\mathcal{D}}\rangle$ as follows.

- $D = \{\langle S, O, i\rangle \mid S \subseteq A$, $O \subseteq S \cap f_i(S)$, and $i \in \{1, \ldots k\}\}$.
- For all $\langle S, O, i\rangle \in D$ and $\sigma \in \Sigma$, the transition function is defined as follows.
  - If $O \neq \emptyset$, then $\delta_D(\langle S, O, i\rangle, \sigma) = \{\langle S', O', i'\rangle\}$, where $S' = \delta_{\mathcal{A}}(S, \sigma)$, $O' = \delta_{\mathcal{A}}(O, \sigma) \cap f_i(S)$ and $i' = i \pmod k + 1$ if $O' = \emptyset$ and i otherwise.
  - If $O = \emptyset$, then $\delta_D(\langle S, O, i\rangle, \sigma) = \{\langle S', O', i'\rangle\}$, where $S' = \delta_{\mathcal{A}}(S, \sigma)$, $O' = \delta_{\mathcal{A}}(S, \sigma) \cap f_i(S)$ and $i' = i \pmod k + 1$ if $O' = \emptyset$ and i otherwise.
- $D_0 = \{\langle A_0 \text{ of } \mathcal{C}_1, \emptyset\rangle\}$.
- $\alpha_{\mathcal{D}} = \{\langle S, O, i\rangle \mid O \neq \emptyset\}$.

A run of $\mathcal{D}$ is accepting if it gets stuck in one of the sets of accepting states. Since the different parts of $\mathcal{C}$ are unconnected, we have that a run of $\mathcal{C}$ is accepting iff it gets stuck in the accepting states of one of the $\mathcal{C}_i$'s. Hence, a word is accepted by $\mathcal{C}$ iff it is accepted by $\mathcal{D}$, and we are done.

□

By [3], one cannot avoid the 3^n state blow-up for translating an NCW to a DCW. Since this lower bound clearly holds also for the stronger conditions, we can conclude with the following.

Theorem 5. *The tight bound for the state blow-up in the translation, when possible, of NPW, NSW, NRW and NMW to an equivalent DCW is $\Theta(3^n)$.*

5 Applications

The translations of nondeterministic automata to NCW and DCW are useful in various applications, mainly in procedures that currently involve determinization. The idea is to either use an NCW instead of a deterministic Büchi or parity automaton, or to use a DBW instead of a deterministic parity automaton. We elaborated on these applications in [2], where the starting point was NBWs. In this section we show that the starting point for the applications can be automata with richer acceptance conditions, and that starting with the richer acceptance conditions (and hence, with automata that may be exponentially more succinct!), involves no extra cost.

In addition, all the applications described in [2] that involve a translation of LTL formulas to NCWs, DCWs or DBWs, can now use an intermediate automaton of the richer classes rather than an NBW. Here too, this can lead to an exponential saving. Indeed, the exponential succinctness of NSW with respect to NBW [15] is proved using languages that can be described by LTL formulas of polynomial length. It follows that there are LTL formulas whose translation to NSW would be exponentially more succinct than their translation to NBW. Moreover, in practice, tools that translate LTL to NBW go through intermediate generalized-Büchi automata, which are a special case of NSW. Our results suggest that in the applications described below, one need not blow-up the state space by going all the way to an NBW.

We first note two important features of the translations. The first feature is the fact that the constructions in Theorems 1, 3, and 4 are based on the subset construction, have a simple state space, are amenable to optimizations, and can be implemented symbolically [14]. The second feature has to do with the one-sided error of the construction, when applied to automata that are not NCW-recognizable: Theorems 1, 3 and 4 guarantee that if the given automaton is NCW-recognizable, then the constructions result in equivalent automata. As stated below, if this is not the case, then the constructions have only a one-sided error.

Lemma 6. *For an automaton $\mathcal{A}$, let $\mathcal{C}$ be the NCW obtained by the translations of Theorems 1 and 3, and let $\mathcal{D}$ be the DCW obtained from $\mathcal{A}$ by applying the breakpoint construction of Theorem 4. Then, $L(\mathcal{A}) \subseteq L(\mathcal{C}) = L(\mathcal{D})$.*

Proof. It is easy to see that the proof of the $L(\mathcal{A}) \subseteq L(\mathcal{C})$ direction in Theorems 1 and 3, as well as the equivalence of $\mathcal{C}$ and $\mathcal{D}$ in Theorem 4, do not rely on the assumption that $\mathcal{A}$ is NCW-recognizable. □

Below we list the main applications. More details can be found in [2] (the description of the problems is the same, except that there the input or intermediate automata are NBWs, whereas here we can handle, at the same complexity, all other acceptance conditions).

- Deciding whether a given automaton (NSW, NPW, NRW, or NMW) is NCW-recognizable.
- Deciding whether a given LTL formula is NCW- or DBW-recognizable.
- Translating an LTL formula to a DBW: For an LTL formula ψ, let $L(\psi)$ denote the set of computations satisfying ψ. Then, the following is an easy corollary of the duality between DBW and DCW.

  Lemma 7. *Consider an LTL formula ψ that is DBW-recognizable. Let $\mathcal{A}_{\neg\psi}$ be a nondeterministic automaton accepting $L(\neg\psi)$, and let $\mathcal{D}_\psi$ be the DBW obtained by dualizing the breakpoint construction of $\mathcal{A}_{\neg\psi}$. Then, $L(\mathcal{D}_\psi) = L(\psi)$.*

  Note that one need not translate the LTL formula to an NBW, and can instead translate it to a nondeterministic generalized Büchi or even to a Streett automaton, which are more succinct.
- Translating LTL formula to the alternation-free μ-calculus.

Using the one-sided error. The one-sided error of the constructions suggest applications also for specifications that are not NCW-recognizable. The translation to DBW, for example, can be used in a decision procedure for CTL* even when the path formulas are not DBW-recognizable.

We demonstrate below how the one-sided error can be used for solving LTL synthesis. Given an arbitrary LTL formula ψ, let $\mathcal{D}_\psi$ be the DBW constructed as in Lemma 7. Lemma 6 implies that $L(\mathcal{D}_\psi) \subseteq L(\psi)$. The polarity of the error (that is, $\mathcal{D}_\psi$ underapproximates ψ) is the helpful one. If we get a transducer that realizes $\mathcal{D}_\psi$, we know that it also realizes ψ, and we are done. Moreover, as suggested in [9], in case $\mathcal{D}_\psi$ is unrealizable, we can check, again using an approximating DBW, whether $\neg\psi$ is realizable for the environment. Only if both ψ is unrealizable for the system and $\neg\psi$ is unrealizable for the environment, we need precise realizability. Note that then, we can also conclude that ψ is not in DBW.

6 Discussion

The simplicity of the co-Büchi condition and its duality to the Büchi condition makes it an interesting theoretical object. Its many recent applications in practice motivate further study of it. Translating automata of rich acceptance conditions to co-Büchi automata is useful in formal verification and synthesis, yet the state blow-up that such translations involve was a long-standing open problem. We solved the problem, and provided asymptotically tight constructions for translating all common automata classes to nondeterministic and deterministic co-Büchi automata.

All the constructions are extensions of the augmented subset construction and breakpoint construction, which are in turn an extension of the basic subset construction. In particular, the set of accepting states is induced by simple reachability queries in the graph of the automaton. Hence, the constructed automata have a simple state space and are amenable to optimizations and to symbolic implementations.

The state blow-up involved in the various translations is summarized in Table 1.

From $\diagdown$ To	NCW	DCW
NBW, NPW, NSW	$n2^n$	3^n
NRW, NMW	$kn2^n$	$k3^n$

Table 1. The state blow-up involved in the translation, when possible, of a word automaton with n states and index k to an equivalent NCW and DCW.

Since the lower bounds for the translations are known for the special case of the origin automaton being an NBW, this is a "good news" paper, providing matching upper bounds. The new translations are significantly, in some cases exponentially, better than known translations. In particular, they show that the exponential blow-ups in the translation of NSW to NBW and of NBW to NCW are not additive. This is quite rare in the theory of automata on infinite words. The good news is carried over to the applications

of the translations. In particular, our results suggest that one need not go via intermediate NBWs in the translation of LTL formulas to DBWs, and that working instead with intermediate NSWs can result in DBWs that are exponentially smaller.

Acknowledgments

We thank an anonymous reviewer for a delicate observation on the proof of Theorem 4.

References

1. Accellera. Accellera organization inc. http://www.accellera.org, 2006.
2. U. Boker and O. Kupferman. Co-ing Büchi made tight and helpful. In *Proc. 24th IEEE Symp. on Logic in Computer Science*, pages 245–254, 2009.
3. U. Boker, O. Kupferman, and A. Rosenberg. Alternation removal in Büchi automata. In *Proc. 37th Int. Colloq. on Automata, Languages, and Programming*, volume 6199, pages 76–87, 2010.
4. Y. Cai, T. Zhang, and H. Luo. An improved lower bound for the complementation of Rabin automata. In *Proc. 24th IEEE Symp. on Logic in Computer Science*, pages 167–176, 2009.
5. S.C. Krishnan, A. Puri, and R.K. Brayton. Deterministic ω-automata vis-a-vis deterministic Büchi automata. In *Algorithms and Computations*, volume 834 of *Lecture Notes in Computer Science*, pages 378–386. Springer, 1994.
6. O. Kupferman. Tightening the exchange rate between automata. In *Proc. 16th Annual Conf. of the European Association for Computer Science Logic*, volume 4646 of *Lecture Notes in Computer Science*, pages 7–22. Springer, 2007.
7. O. Kupferman, G. Morgenstern, and A. Murano. Typeness for ω-regular automata. In *2nd Int. Symp. on Automated Technology for Verification and Analysis*, volume 3299 of *Lecture Notes in Computer Science*, pages 324–338. Springer, 2004.
8. O. Kupferman, N. Piterman, and M.Y. Vardi. Safraless compositional synthesis. In *Proc. 18th Int. Conf. on Computer Aided Verification*, LNCS vol. 4144, pages 31–44, 2006.
9. O. Kupferman and M.Y. Vardi. Safraless decision procedures. In *Proc. 46th IEEE Symp. on Foundations of Computer Science*, pages 531–540, 2005.
10. R.P. Kurshan. *Computer Aided Verification of Coordinating Processes*. Princeton Univ. Press, 1994.
11. L.H. Landweber. Decision problems for ω–automata. *Mathematical Systems Theory*, 3:376–384, 1969.
12. R. McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, 9:521–530, 1966.
13. S. Miyano and T. Hayashi. Alternating finite automata on ω-words. *Theoretical Computer Science*, 32:321–330, 1984.
14. A. Morgenstern and K. Schneider. From LTL to symbolically represented deterministic automata. In *Proc. 9th Int. Conf. on Verification, Model Checking, and Abstract Interpretation*, volume 4905 of *Lecture Notes in Computer Science*, pages 279–293, 2008.
15. 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.
16. H. Seidl and D. Niwiński. On distributive fixed-point expressions. *Theoretical Informatics and Applications*, 33(4–5):427–446, 1999.
17. W. Thomas. Automata on infinite objects. *Handbook of Theoretical Computer Science*, pages 133–191, 1990.
18. M.Y. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, 1994.