Canonicity in GFG and Transition-Based Automata

Bader Abu Radi and Orna Kupferman
School of Computer Science and Engineering, The Hebrew University, Israel
bader.aburadi@gmail.com orna@cs.huji.ac.il

Minimization of deterministic automata on finite words results in a *canonical* automaton. For deterministic automata on infinite words, no canonical minimal automaton exists, and a language may have different minimal deterministic Büchi (DBW) or co-Büchi (DCW) automata.

In recent years, researchers have studied *good-for-games* (GFG) automata – nondeterministic automata that can resolve their nondeterministic choices in a way that only depends on the past. Several applications of automata in formal methods, most notably synthesis, that are traditionally based on deterministic automata, can instead be based on GFG automata.

The *minimization* problem for DBW and DCW is NP-complete, and it stays NP-complete for GFG Büchi and co-Büchi automata. On the other hand, minimization of GFG co-Büchi automata with *transition-based* acceptance (GFG-tNCWs) can be solved in polynomial time. In these automata, acceptance is defined by a set α of transitions, and a run is accepting if it traverses transitions in α only finitely often. This raises the question of canonicity of minimal deterministic and GFG automata with transition-based acceptance.

In this paper we study this problem. We start with GFG-tNCWs and show that the safe components (that is, these obtained by restricting the transitions to these not in α) of all minimal GFG-tNCWs are isomorphic, and that by saturating the automaton with transitions in α we get isomorphism among all minimal GFG-tNCWs. Thus, a canonical form for minimal GFG-tNCWs can be obtained in polynomial time. We continue to DCWs with transition-based acceptance (tDCWs), and their dual tDBWs. We show that here, while no canonical form for minimal automata exists, restricting attention to the safe components is useful, and implies that the only minimal tDCWs that have no canonical form are these for which the transition to the GFG model results in strictly smaller automaton, which do have a canonical minimal form.

1 Introduction

Automata theory is one of the longest established areas in computer science. A classical problem in automata theory is *minimization*: generation of an equivalent automaton with a minimal number of states. For deterministic automata on finite words, a minimization algorithm, based on the Myhill-Nerode right congruence [18, 19], generates in polynomial time a canonical minimal deterministic automaton [11]. Essentially, the canonical automaton, a.k.a. the *quotient automaton*, is obtained by merging equivalent states.

A prime application of automata theory is specification, verification, and synthesis of reactive systems [25, 13]. Since we care about the on-going behaviors of nonterminating systems, the automata run on infinite words and define ω-regular languages. Acceptance in such automata is determined according to the set of states that are visited infinitely often during the run. In Büchi automata [5] (NBW and DBW, for nondeterministic and deterministic Büchi word automata, respectively), the acceptance condition is a subset α of states, and a run is accepting iff it visits α infinitely often. Dually, in co-Büchi automata (NCW and DCW), a run is accepting iff it visits α only finitely often.

For ω-regular languages, no canonical minimal deterministic automaton exists, and a language may have different minimal DBWs or DCWs. Consider for example the DCWs $\mathcal{A}_1$ and $\mathcal{A}_2$ appearing in

Figure 1. Both are minimal DCWs for the language $L = (a+b)^* \cdot (a^\omega + b^\omega)$ ("only finitely many a's or only finitely many b's"; it is easier to see this by considering the dual DBWs, for "infinitely many a's and infinitely many b's").

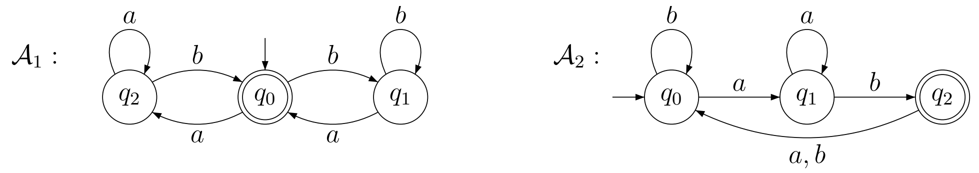

Figure 1: The DCWs $\mathcal{A}_1$ and $\mathcal{A}_2$.

Since all the states of $\mathcal{A}_1$ and $\mathcal{A}_2$ recognize the language L and may serve as initial states, Figure 1 actually presents six different DCWs for L, and more three-state DCWs for L exist. The DCWs $\mathcal{A}_1$ and $\mathcal{A}_2$, are however "more different" than variants of $\mathcal{A}_1$ obtained by changing the initial state: they have a different structure, or more formally, there is no isomorphism between their graphs.

In some applications of automata on infinite words, such as model checking, algorithms can proceed with nondeterministic automata. In other applications, such as synthesis and control, they cannot. The algorithms for these applications involve solving a game that is played on an arena that is based on the automaton. The difficulty in using nondeterministic automata in such game-based algorithms lies in the fact that when a player resolves nondeterminism, her choices should accommodate all possible futures.

A study of nondeterministic automata that can resolve their nondeterministic choices in a way that only depends on the past started in [14], where the setting is modeled by means of tree automata for derived languages. It then continued by means of *good for games* (GFG) automata [10].[1] A nondeterministic automaton $\mathcal{A}$ over an alphabet Σ is GFG if there is a strategy g that maps each finite word $u \in \Sigma^*$ to the transition to be taken after u is read; and following g results in accepting all the words in the language of $\mathcal{A}$. Note that a state q of $\mathcal{A}$ may be reachable via different words, and g may suggest different transitions from q after different words are read. Still, g depends only on the past, namely on the word read so far. Obviously, there exist GFG automata: deterministic ones, or nondeterministic ones that are *determinizable by pruning* (DBP); that is, ones that just add transitions on top of a deterministic automaton. In fact, the GFG automata constructed in [10] are DBP.[2]

In terms of expressive power, it is shown in [14, 20] that GFG automata with an acceptance condition γ (e.g., Büchi) are as expressive as deterministic γ automata. The picture in terms of succinctness is diverse. For automata on finite words, GFG automata are always DBP [14, 17]. For automata on infinite words, in particular NBWs and NCWs, GFG automata need not be DBP [3]. Moreover, the best known determinization construction for GFG-NBWs is quadratic, whereas determinization of GFG-NCWs has an exponential blow-up lower bound [12]. Thus, GFG automata on infinite words are more succinct (possibly even exponentially) than deterministic ones.[3] Further research studies characterization, typeness, complementation, and further constructions and decision procedures for GFG automata [12, 4, 2], as well as an extension of the GFG setting to pushdown ω-automata [15].

Recall that for automata on finite words, a minimal deterministic automaton can be obtained by merging equivalent states. For general DBWs (and hence, also DCWs, as the two dualize each other), merging equivalent states fails, and minimization is NP-complete [21]. Proving NP-hardness, Schewe

[1]GFGness is also used in [6] in the framework of cost functions under the name "history-determinism".

[2]As explained in [10], the fact that the GFG automata constructed there are DBP does not contradict their usefulness in practice, as their transition relation is simpler than the one of the embodied deterministic automaton and it can be defined symbolically.

[3]We note that some of the succinctness results are known only for GFG automata with *transition-based* acceptance.

used a reduction from the vertex-cover problem [21]. Essentially, the choice of a vertex cover in a given graph G is reduced to a choice of a set of states that should be duplicated in a DBW induced by G. The duplication is needed for the definition of the acceptance condition, and is not needed when the DBW is defined with a *transition-based* acceptance condition. In such automata, the acceptance condition is given by a subset α of the transitions, and a run is required to traverse transitions in α infinitely often (in Büchi automata, denoted tNBW), or finitely often (in co-Büchi automata, denoted tNCW). Thus, while minimization is NP-complete for DBW and DCW, its complexity is open for tDBWs and tDCWs. Beyond the theoretical interest, there is recently growing use of transition-based automata in practical applications, with evidences they offer a simpler translation of LTL formulas to automata and enable simpler constructions and decision procedures [8, 7, 23, 16].

In [1], we described a polynomial-time algorithm for the minimization of GFG-tNCWs. Consider a GFG-tNCW $\mathcal{A}$. Our algorithm is based on an analysis of the *safe components* of $\mathcal{A}$, namely its strongly connected components obtained by removing transitions in α. Note that every accepting run of $\mathcal{A}$ eventually reaches and stays forever in a safe component. We showed that a minimal GFG-tNCW equivalent to $\mathcal{A}$ can be obtained by defining an order on the safe components, and applying the quotient construction on a GFG-tNCW obtained by restricting attention to states that belong to components that form a frontier in this order. Considering GFG-tNCWs rather than DBWs involves two modifications of the original question: a transition to GFG rather than deterministic automata, and a transition to transition-based rather than state-based acceptance. A natural question that arises is whether both modifications are crucial for efficiency. It was shown recently [22] that the NP-completeness proof of Schewe for DBW minimization can be generalized to GFG-NBWs and GFG-NCWs. This suggests that the consideration of transition-based acceptance has been crucial, and makes the study of tDBW and tDCW very appealing.

Minimization and its complexity are tightly related to the canonicity question. Recall that ω-regular languages do not have a unique minimal DBW or DCW. In this paper we study canonicity for GFG and transition-based automata. We start with GFG-tNCWs and show that all minimal GFG-tNCWs are *safe isomorphic*, namely their safe components are isomorphic[4]. More formally, if $\mathcal{A}_1$ and $\mathcal{A}_2$ are minimal GFG-tNCWs for the same language, then there exists a bijection between the state spaces of $\mathcal{A}_1$ and $\mathcal{A}_2$ that induces a bijection between their $\bar{\alpha}$-transitions (these not in α). We then show that by saturating the GFG-tNCW with α-transitions we get isomorphism among all minimal automata. We suggest two possible saturations. One adds as many α-transitions as possible, and the second does so in a way that preserves α-homogeneity, thus for every state q and letter σ, all the transitions labeled σ from q are α-transitions or are all $\bar{\alpha}$-transitions. Since the minimization algorithm of [1] generates minimal α-homogenous GFG-tNCWs, it follows that both forms of canonical minimal GFG-tNCW can be obtained in polynomial time.

We then show that, as has been the case with minimization, GFGness is not a sufficient condition for canonicity, raising the question of canonicity in tDCWs. Note that unlike the GFG-tNCW setting, here dualization of the acceptance condition complements the language of an automaton, and thus our results apply also to canonicity of tDBWs. We start with some bad news, showing that as has been the case with DCWs and DBW, minimal tDCWs and tDBWs need not be isomorphic. Moreover, being deterministic, we cannot saturate their transitions and make them isomorphic. On the positive side, safe isomorphism is helpful also in the tDCW setting: Consider an ω-regular language L. Recall that the minimal GFG-tNCW for L may be smaller than a minimal tDCW for L [12]. We say that L is *tDCW-positive* if this is not the case. We prove that all minimal tDCWs for a tDCW-positive ω-regular language are safe isomorphic.

[4]In our results, we assume the GFG-tNCWs are *nice*: they satisfy some syntactic and semantic properties that can be easily obtained from every GFG-tNCW.

Note that for such languages, we also know how to generate a minimal tDCW in polynomial time. For ω-regular languages that are not tDCW-positive, safe isomorphism is left open. For such languages, however, we care more about minimal GFG-tNCWs, which do have a canonical form. Also, all natural ω-regular languages are tDCW-positive, and in fact the existence of ω-regular languages that are not tDCW-positive has been open for quite a while [3]. Accordingly, we view our results as good news about canonicity in deterministic automata with transition-based acceptance.

2 Preliminaries

For a finite nonempty alphabet Σ, an infinite *word* $w = \sigma_1 \cdot \sigma_2 \cdots \in \Sigma^\omega$ is an infinite sequence of letters from Σ. A *language* $L \subseteq \Sigma^\omega$ is a set of words. We denote the empty word by ε, and the set of finite words over Σ by Σ^*. For $i \geq 0$, we use $w[1,i]$ to denote the (possibly empty) prefix $\sigma_1 \cdot \sigma_2 \cdots \sigma_i$ of w and use $w[i+1,\infty]$ to denote its suffix $\sigma_{i+1} \cdot \sigma_{i+2} \cdots$.

A *nondeterministic automaton* over infinite words is $\mathcal{A} = \langle \Sigma, Q, q_0, \delta, \alpha \rangle$, where Σ is an alphabet, Q is a finite set of *states*, $q_0 \in Q$ is an *initial state*, $\delta : Q \times \Sigma \to 2^Q \setminus \emptyset$ is a *transition function*, and α is an *acceptance condition*, to be defined below. For states q and s and a letter $\sigma \in \Sigma$, we say that s is a σ-successor of q if $s \in \delta(q,\sigma)$. The *size* of $\mathcal{A}$, denoted $|\mathcal{A}|$, is defined as its number of states, thus, $|\mathcal{A}| = |Q|$. Note that $\mathcal{A}$ is *total*, in the sense that it has at least one successor for each state and letter, and that $\mathcal{A}$ may be *nondeterministic*, as the transition function may specify several successors for each state and letter. If $|\delta(q,\sigma)| = 1$ for every state $q \in Q$ and letter $\sigma \in \Sigma$, then $\mathcal{A}$ is *deterministic*.

When $\mathcal{A}$ runs on an input word, it starts in the initial state and proceeds according to the transition function. Formally, a *run* of $\mathcal{A}$ on $w = \sigma_1 \cdot \sigma_2 \cdots \in \Sigma^\omega$ is an infinite sequence of states $r = r_0, r_1, r_2, \ldots \in Q^\omega$, such that $r_0 = q_0$, and for all $i \geq 0$, we have that $r_{i+1} \in \delta(r_i, \sigma_{i+1})$. We sometimes extend δ to sets of states and finite words. Then, $\delta : 2^Q \times \Sigma^* \to 2^Q$ is such that for every $S \in 2^Q$, finite word $u \in \Sigma^*$, and letter $\sigma \in \Sigma$, we have that $\delta(S,\varepsilon) = S$, $\delta(S,\sigma) = \bigcup_{s \in S} \delta(s,\sigma)$, and $\delta(S, u \cdot \sigma) = \delta(\delta(S,u),\sigma)$. Thus, $\delta(S,u)$ is the set of states that $\mathcal{A}$ may reach when it reads u from some state in S.

The transition function δ induces a transition relation $\Delta \subseteq Q \times \Sigma \times Q$, where for every two states $q,s \in Q$ and letter $\sigma \in \Sigma$, we have that $\langle q,\sigma,s \rangle \in \Delta$ iff $s \in \delta(q,\sigma)$. We sometimes view the run $r = r_0, r_1, r_2, \ldots$ on $w = \sigma_1 \cdot \sigma_2 \cdots$ as an infinite sequence of successive transitions $\langle r_0, \sigma_1, r_1 \rangle, \langle r_1, \sigma_2, r_2 \rangle, \ldots \in \Delta^\omega$. The acceptance condition α determines which runs are "good". We consider here *transition-based* automata, in which α refers to the set of transitions that are traversed infinitely often during the run; specifically, $\alpha \subseteq \Delta$. We use the terms *α-transitions* and *$\bar{\alpha}$-transitions* to refer to transitions in α and in $\Delta \setminus \alpha$, respectively. We also refer to restrictions δ^α and $\delta^{\bar{\alpha}}$ of δ, where for all $q,s \in Q$ and $\sigma \in \Sigma$, we have that $s \in \delta^\alpha(q,\sigma)$ iff $\langle q,\sigma,s \rangle \in \alpha$, and $s \in \delta^{\bar{\alpha}}(q,\sigma)$ iff $\langle q,\sigma,s \rangle \in \Delta \setminus \alpha$. For a run $r \in \Delta^\omega$, let $\mathit{inf}(r) \subseteq \Delta$ be the set of transitions that r traverses infinitely often. Thus, $\mathit{inf}(r) = \{\langle q,\sigma,s \rangle \in \Delta : q = r_i, \sigma = \sigma_{i+1}$ and $s = r_{i+1}$ for infinitely many i's$\}$. In *co-Büchi* automata, a run r is *accepting* iff $\mathit{inf}(r) \cap \alpha = \emptyset$, thus if r traverses transitions in α only finitely often. A run that is not accepting is *rejecting*. A word w is accepted by $\mathcal{A}$ if there is an accepting run of $\mathcal{A}$ on w. The language of $\mathcal{A}$, denoted $L(\mathcal{A})$, is the set of words that $\mathcal{A}$ accepts. Two automata are *equivalent* if their languages are equivalent. We use tNCW and tDCW to abbreviate nondeterministic and deterministic transition-based co-Büchi automata over infinite words, respectively.

We continue to definitions and notations that are relevant to our study. See Section 7 for a glossary. For an automaton $\mathcal{A} = \langle \Sigma, Q, q_0, \delta, \alpha \rangle$, and a state $q \in Q$, we define $\mathcal{A}^q$ to be the automaton obtained from $\mathcal{A}$ by setting the initial state to be q. Thus, $\mathcal{A}^q = \langle \Sigma, Q, q, \delta, \alpha \rangle$. We say that two states $q,s \in Q$ are *equivalent*, denoted $q \sim_{\mathcal{A}} s$, if $L(\mathcal{A}^q) = L(\mathcal{A}^s)$. The automaton $\mathcal{A}$ is *semantically deterministic* if different nondeterministic choices lead to equivalent states. Thus, for every state $q \in Q$ and letter $\sigma \in \Sigma$, all the σ-successors of q are equivalent: for every two states $s, s' \in Q$ such that $\langle q, \sigma, s \rangle$ and $\langle q, \sigma, s' \rangle$ are in Δ, we have that $s \sim_{\mathcal{A}} s'$. The following proposition follows immediately from the definitions.

Proposition 2.1. *Consider a semantically deterministic automaton $\mathcal{A}$, states $q, s \in Q$, letter $\sigma \in \Sigma$, and transitions $\langle q, \sigma, q' \rangle, \langle s, \sigma, s' \rangle \in \Delta$. If $q \sim_{\mathcal{A}} s$, then $q' \sim_{\mathcal{A}} s'$.*

An automaton $\mathcal{A}$ is *good for games* (*GFG*, for short) if its nondeterminism can be resolved based on the past, thus on the prefix of the input word read so far. Formally, $\mathcal{A}$ is *GFG* if there exists a *strategy* $f : \Sigma^* \to Q$ such that the following holds:

1. The strategy f is consistent with the transition function. That is, for every finite word $u \in \Sigma^*$ and letter $\sigma \in \Sigma$, we have that $\langle f(u), \sigma, f(u \cdot \sigma) \rangle \in \Delta$.
2. Following f causes $\mathcal{A}$ to accept all the words in its language. That is, for every infinite word $w = \sigma_1 \cdot \sigma_2 \cdots \in \Sigma^\omega$, if $w \in L(\mathcal{A})$, then the run $f(w[1,0]), f(w[1,1]), f(w[1,2]), \ldots$, which we denote by $f(w)$, is accepting.

We say that the strategy f *witnesses* $\mathcal{A}$'s GFGness. For an automaton $\mathcal{A}$, we say that a state q of $\mathcal{A}$ is *GFG* if $\mathcal{A}^q$ is GFG. Note that every deterministic automaton is GFG. We say that a GFG automaton $\mathcal{A}$ is *determinizable by prunning* (DBP) if we can remove some of the transitions of $\mathcal{A}$ and get a deterministic automaton that recognizes $L(\mathcal{A})$.

Consider a directed graph $G = \langle V, E \rangle$. A *strongly connected set* in G (SCS, for short) is a set $C \subseteq V$ such that for every two vertices $v, v' \in C$, there is a path from v to v'. A SCS is *maximal* if it is maximal w.r.t containment, that is, for every non-empty set $C' \subseteq V \setminus C$, it holds that $C \cup C'$ is not a SCS. The *maximal strongly connected sets* are also termed *strongly connected components* (SCCs, for short). The *SCC graph of G* is the graph defined over the SCCs of G, where there is an edge from a SCC C to another SCC C' iff there are two vertices $v \in C$ and $v' \in C'$ with $\langle v, v' \rangle \in E$. A SCC is *ergodic* iff it has no outgoing edges in the SCC graph. The SCC graph of G can be computed in linear time by standard SCC algorithms [24].

An automaton $\mathcal{A} = \langle \Sigma, Q, q_0, \delta, \alpha \rangle$ induces a directed graph $G_{\mathcal{A}} = \langle Q, E \rangle$, where $\langle q, q' \rangle \in E$ iff there is a letter $\sigma \in \Sigma$ such that $\langle q, \sigma, q' \rangle \in \Delta$. The SCSs and SCCs of $\mathcal{A}$ are those of $G_{\mathcal{A}}$. We say that a tNCW $\mathcal{A}$ is *safe deterministic* if by removing its α-transitions, we get a (possibly not total) deterministic automaton. Thus, $\mathcal{A}$ is *safe deterministic* if for every state $q \in Q$ and letter $\sigma \in \Sigma$, it holds that $|\delta^{\bar{\alpha}}(q, \sigma)| \leq 1$. We refer to the SCCs we get by removing $\mathcal{A}$'s α-transitions as the *safe components* of $\mathcal{A}$; that is, the *safe components* of $\mathcal{A}$ are the SCCs of the graph $G_{\mathcal{A}^{\bar{\alpha}}} = \langle Q, E^{\bar{\alpha}} \rangle$, where $\langle q, q' \rangle \in E^{\bar{\alpha}}$ iff there is a letter $\sigma \in \Sigma$ such that $q' \in \delta^{\bar{\alpha}}(q, \sigma)$. We denote the set of safe components of $\mathcal{A}$ by $\mathcal{S}(\mathcal{A})$. For a safe component $S \in \mathcal{S}(\mathcal{A})$, the *size* of S, denoted $|S|$, is the number of states in S. Note that an accepting run of $\mathcal{A}$ eventually gets trapped in one of $\mathcal{A}$'s safe components. A tNCW $\mathcal{A}$ is *normal* if there are no $\bar{\alpha}$-transitions connecting different safe components. That is, for all states q and s of $\mathcal{A}$, if there is a path of $\bar{\alpha}$-transitions from q to s, then there is also a path of $\bar{\alpha}$-transitions from s to q.

We now combine several properties defined above and say that a GFG-tNCW $\mathcal{A}$ is *nice* if all the states in $\mathcal{A}$ are reachable and GFG, and $\mathcal{A}$ is normal, safe deterministic, and semantically deterministic. As Theorem 2.2 below shows, each of these properties can be obtained in at most polynomial time, and without the properties being conflicting.

Theorem 2.2. [12, 1] *Every GFG-tNCW $\mathcal{A}$ can be turned, in polynomial time, into an equivalent nice GFG-tNCW $\mathcal{B}$ such that $|\mathcal{B}| \leq |\mathcal{A}|$.*

Consider a tNCW $\mathcal{A} = \langle \Sigma, Q, q_0, \delta, \alpha \rangle$. A run r of $\mathcal{A}$ is *safe* if it does not traverse α-transitions. The *safe language* of $\mathcal{A}$, denoted $L_{safe}(\mathcal{A})$, is the set of infinite words w, such that there is a safe run of $\mathcal{A}$ on

w. Recall that two states $q,s \in Q$ are equivalent ($q \sim_{\mathcal{A}} s$) if $L(\mathcal{A}^q) = L(\mathcal{A}^s)$. Then, q and s are *strongly-equivalent*, denoted $q \approx_{\mathcal{A}} s$, if $q \sim_{\mathcal{A}} s$ and $L_{safe}(\mathcal{A}^q) = L_{safe}(\mathcal{A}^s)$. Finally, q is *subsafe-equivalent to* s, denoted $q \precsim_{\mathcal{A}} s$, if $q \sim_{\mathcal{A}} s$ and $L_{safe}(\mathcal{A}^q) \subseteq L_{safe}(\mathcal{A}^s)$. Note that the three relations are transitive. When $\mathcal{A}$ is clear from the context, we omit it from the notations, thus write $L_{safe}(q)$, $q \precsim s$, etc. The tNCW $\mathcal{A}$ is *safe-minimal* if it has no strongly-equivalent states. Then, $\mathcal{A}$ is *safe-centralized* if for every two states $q,s \in Q$, if $q \precsim s$, then q and s are in the same safe component of $\mathcal{A}$. Finally, $\mathcal{A}$ is *α-homogenous* if for every state $q \in Q$ and letter $\sigma \in \Sigma$, either $\delta^{\alpha}_{\mathcal{A}}(q,\sigma) = \emptyset$ or $\delta^{\bar{\alpha}}_{\mathcal{A}}(q,\sigma) = \emptyset$. Thus, either all the σ-labeled transitions from q are α-transitions, or they are all $\bar{\alpha}$-transitions.

Example 2.1. Consider the tDCW $\mathcal{A}$ appearing in Figure 2. The dashed transitions are α-transitions. All the states of $\mathcal{A}$ are equivalent, yet they all differ in their safe language. Accordingly, $\mathcal{A}$ is safe-minimal. Since $a^{\omega} = L_{safe}(\mathcal{A}^{q_2}) \subseteq L_{safe}(\mathcal{A}^{q_0})$, we have that $q_2 \precsim q_0$. Hence, as q_0 and q_2 are in different safe components, the tDCW $\mathcal{A}$ is not safe-centralized. □

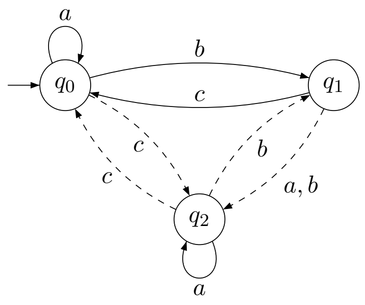

Figure 2: The tDCW $\mathcal{A}$.

The following properties of nice GFG-tNCWs are proven in [1].

Proposition 2.3. *Consider a nice GFG-tNCW $\mathcal{A}$ and states q and s of $\mathcal{A}$ such that $q \approx s$ ($q \precsim s$). For every letter $\sigma \in \Sigma$ and $\bar{\alpha}$-transition $\langle q,\sigma,q'\rangle$, there is an $\bar{\alpha}$-transition $\langle s,\sigma,s'\rangle$ such that $q' \approx s'$ ($q' \precsim s'$, respectively).*

Proposition 2.4. *Let $\mathcal{A}$ and $\mathcal{B}$ be equivalent nice GFG-tNCWs. For every state p of $\mathcal{A}$, there are states q of $\mathcal{A}$ and s of $\mathcal{B}$, such that $p \precsim q$ and $q \approx s$.*

Lemma 2.5. *Consider a nice GFG-tNCW $\mathcal{A}$. If $\mathcal{A}$ is safe-centralized and safe-minimal, then for every nice GFG-tNCW $\mathcal{B}$ equivalent to $\mathcal{A}$, there is an injection $\eta : \mathcal{S}(\mathcal{A}) \to \mathcal{S}(\mathcal{B})$ such that for every safe component $T \in \mathcal{S}(\mathcal{A})$, it holds that $|T| \leq |\eta(T)|$.*

3 Minimizing GFG-tNCW

A GFG-tNCW $\mathcal{A}$ is *minimal* if for every equivalent GFG-tNCW $\mathcal{B}$, it holds that $|\mathcal{A}| \leq |\mathcal{B}|$. In this section, we review the minimization construction of [1], highlighting its properties that are important for the canonization results. The algorithm is based on the following theorem.

Theorem 3.1. *Consider a nice GFG-tNCW $\mathcal{A}$. If $\mathcal{A}$ is safe-centralized and safe-minimal, then $\mathcal{A}$ is a minimal GFG-tNCW for $L(\mathcal{A})$.*

Thus, minimization involves two steps: safe centralization and safe minimization.

Step 1: Safe centralization Consider a nice GFG-tNCW $\mathcal{A} = \langle \Sigma, Q_\mathcal{A}, q^0_\mathcal{A}, \delta_\mathcal{A}, \alpha_\mathcal{A} \rangle$. Recall that $\mathcal{S}(\mathcal{A})$ denotes the set of safe components of $\mathcal{A}$. Let $H \subseteq \mathcal{S}(\mathcal{A}) \times \mathcal{S}(\mathcal{A})$ be such that for all safe components $S, S' \in \mathcal{S}(\mathcal{A})$, we have that $H(S,S')$ iff there exist states $q \in S$ and $q' \in S'$ such that $q \precsim q'$. The relation H is transitive: for every safe components $S, S', S'' \in \mathcal{S}(\mathcal{A})$, if $H(S,S')$ and $H(S',S'')$, then $H(S,S'')$. We say that a set $\mathcal{S} \subseteq \mathcal{S}(\mathcal{A})$ is a *frontier of* $\mathcal{A}$ if for every safe component $S \in \mathcal{S}(\mathcal{A})$, there is a safe component $S' \in \mathcal{S}$ with $H(S,S')$, and for all safe components $S, S' \in \mathcal{S}$ such that $S \neq S'$, we have that $\neg H(S,S')$ and $\neg H(S',S)$. Once H is calculated, a frontier of $\mathcal{A}$ can be found in linear time. For example, as H is transitive, we can take one vertex from each ergodic SCC in the graph $\langle \mathcal{S}(\mathcal{A}), H \rangle$. Note that all frontiers of $\mathcal{A}$ are of the same size, namely the number of ergodic SCCs in this graph.

Proposition 3.2. *Consider safe components $S, S' \in \mathcal{S}(\mathcal{A})$ such that $H(S,S')$. Then, for every state $p \in S$ there is a state $p' \in S'$, such that $p \precsim p'$.*

Given a frontier $\mathcal{S}$ of $\mathcal{A}$, we define the automaton $\mathcal{B}_\mathcal{S} = \langle \Sigma, Q_\mathcal{S}, q^0_\mathcal{S}, \delta_\mathcal{S}, \alpha_\mathcal{S} \rangle$, where $Q_\mathcal{S} = \{q \in Q_\mathcal{A} : q \in S \text{ for some } S \in \mathcal{S}\}$, and the other elements are defined as follows. The initial state $q^0_\mathcal{S}$ is chosen such that $q^0_\mathcal{S} \sim_\mathcal{A} q^0_\mathcal{A}$. Specifically, if $q^0_\mathcal{A} \in Q_\mathcal{S}$, we take $q^0_\mathcal{S} = q^0_\mathcal{A}$. Otherwise, by Proposition 3.2 and the definition of $\mathcal{S}$, there is a state $q' \in Q_\mathcal{S}$ such that $q^0_\mathcal{A} \precsim q'$, and we take $q^0_\mathcal{S} = q'$. The transitions in $\mathcal{B}_\mathcal{S}$ are either $\bar{\alpha}$-transitions of $\mathcal{A}$, or α-transitions that we add among the safe components in $\mathcal{S}$ in a way that preserves language equivalence. Formally, consider a state $q \in Q_\mathcal{S}$ and a letter $\sigma \in \Sigma$. If $\delta^{\bar{\alpha}}_\mathcal{A}(q,\sigma) \neq \emptyset$, then $\delta^{\bar{\alpha}}_\mathcal{S}(q,\sigma) = \delta^{\bar{\alpha}}_\mathcal{A}(q,\sigma)$ and $\delta^{\alpha}_\mathcal{S}(q,\sigma) = \emptyset$. If $\delta^{\bar{\alpha}}_\mathcal{A}(q,\sigma) = \emptyset$, then $\delta^{\bar{\alpha}}_\mathcal{S}(q,\sigma) = \emptyset$ and $\delta^{\alpha}_\mathcal{S}(q,\sigma) = \{q' \in Q_\mathcal{S} : \text{ there is } q'' \in \delta^{\alpha}_\mathcal{A}(q,\sigma) \text{ such that } q' \sim_\mathcal{A} q''\}$. Note that $\mathcal{B}_\mathcal{S}$ is α-homogenous.

Example 3.1. Consider the nice tDCW $\mathcal{A}$ from Figure 2. By removing the α-transitions of $\mathcal{A}$, we get the safe components described in Figure 3. Since $q_2 \precsim q_0$, we have that $\mathcal{A}$ has a single frontier $\mathcal{S} = \{\{q_0, q_1\}\}$. The automaton $\mathcal{B}_\mathcal{S}$ appears in Figure 4. As all the states of $\mathcal{A}$ are equivalent, we direct a σ-labeled α-transition to q_0 and to q_1, for every state with no σ-labeled transition in $\mathcal{A}$. □

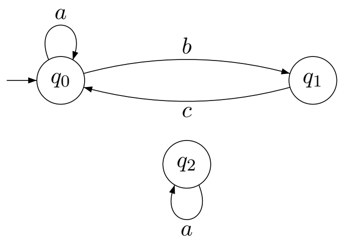

Figure 3: The safe components of $\mathcal{A}$.

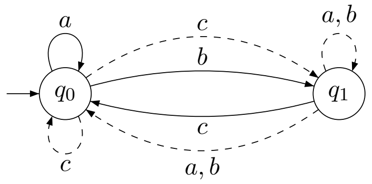

Figure 4: The tNCW $\mathcal{B}_\mathcal{S}$ for $\mathcal{S} = \{\{q_0, q_1\}\}$.

Proposition 3.3. *Let q and s be states of $\mathcal{A}$ and $\mathcal{B}_\mathcal{S}$, respectively, with $q \sim_\mathcal{A} s$. It holds that $\mathcal{B}^s_\mathcal{S}$ is a GFG-tNCW equivalent to $\mathcal{A}^q$.*

Proposition 3.4. *For every frontier $\mathcal{S}$, the automaton $\mathcal{B}_\mathcal{S}$ is a nice, safe-centralized, and α-homogenous GFG-tNCW equivalent to $\mathcal{A}$.*

Step 2: Safe minimization Let $\mathcal{B} = \langle \Sigma, Q, q_0, \delta, \alpha \rangle$ be a nice, safe-centralized, and α-homogenous GFG-tNCW. For $q \in Q$, define $[q] = \{q' \in Q : q \approx_\mathcal{B} q'\}$. We define the tNCW $\mathcal{C} = \langle \Sigma, Q_\mathcal{C}, [q_0], \delta_\mathcal{C}, \alpha_\mathcal{C} \rangle$ as follows. First, $Q_\mathcal{C} = \{[q] : q \in Q\}$. Then, the transition function is such that $\langle [q], \sigma, [p] \rangle \in \Delta_\mathcal{C}$ iff there are $q' \in [q]$ and $p' \in [p]$ such that $\langle q', \sigma, p' \rangle \in \Delta$, and $\langle [q], \sigma, [p] \rangle \in \alpha_\mathcal{C}$ iff $\langle q', \sigma, p' \rangle \in \alpha$. Note that $\mathcal{B}$ being α-homogenous implies that $\alpha_\mathcal{C}$ is well defined; that is, independent of the choice of q' and p'. To see why, assume that $\langle q', \sigma, p' \rangle \in \bar{\alpha}$, and let q'' be a state in $[q]$. As $q' \approx_\mathcal{B} q''$, we have, by Proposition 2.3, that there is $p'' \in [p]$ such that $\langle q'', \sigma, p'' \rangle \in \bar{\alpha}$. Thus, as $\mathcal{B}$ is α-homogenous, there is no σ-labeled

α-transition from q'' in $\mathcal{B}$. In particular, there is no σ-labeled α-transition from q'' to a state in $[p]$. Note that, by the above, the tNCW $\mathcal{C}$ is α-homogenous.

Proposition 3.5. *For every $[p] \in Q_\mathcal{C}$ and $s \in [p]$, we have that $\mathcal{C}^{[p]}$ is a GFG-tNCW equivalent to $\mathcal{B}^s$.*

Proposition 3.6. *The GFG-tNCW $\mathcal{C}$ is a nice, safe-centralized, safe-minimal, and α-homogenous GFG-tNCW equivalent to $\mathcal{A}$.*

Example 3.2. The safe languages of the states q_0 and q_1 of the GFG-tNCW $\mathcal{B}_\mathcal{S}$ from Figure 4 are different. Thus, $q_0 \not\approx q_1$, and applying safe minimization to $\mathcal{B}_\mathcal{S}$ results in the GFG-tNCW $\mathcal{C}$ identical to $\mathcal{B}_\mathcal{S}$. □

4 Canonicity in GFG-NCWs

In this section we study canonicity for GFG-tNCWs. We first show that the sufficient conditions for minimality of nice GFG-tNCWs specified in Theorem 3.1 are necessary.

Theorem 4.1. *Nice minimal GFG-tNCWs are safe-centralized and safe-minimal.*

Proof. Consider a nice minimal GFG-tNCW $\mathcal{A}$. We argue that if $\mathcal{A}$ is not safe-centralized or not safe-minimal, then it can be minimized further by the minimization construction of [1]. Assume first that $\mathcal{A}$ is not safe-centralized. Then, there are two different safe components $S, S' \in \mathcal{S}(\mathcal{A})$ and states $q \in S$ and $q' \in S'$ such that $q \precsim q'$. Then, $H(S,S')$, implying that the safe components in $\mathcal{S}(\mathcal{A})$ are not a frontier. Then, Step 1 of the construction minimizes $\mathcal{A}$ further. Indeed, in the transition to the automaton $\mathcal{B}_\mathcal{S}$, at least one safe component in $\mathcal{S}(\mathcal{A})$ is removed from $\mathcal{A}$ when the frontier $\mathcal{S}$ is computed. Assume now that $\mathcal{A}$ is safe-centralized. Then, for every two different safe components $S, S' \in \mathcal{S}(\mathcal{A})$, it holds that $\neg H(S,S')$ and $\neg H(S',S)$. Hence, every strict subset of $\mathcal{S}(\mathcal{A})$ is not a frontier. Thus, $\mathcal{S}(\mathcal{A})$ is the only frontier of $\mathcal{A}$. Hence, the automaton $\mathcal{B}_\mathcal{S}$ constructed in Step 1 has $\mathcal{S} = \mathcal{S}(\mathcal{A})$, and is obtained from $\mathcal{A}$ by adding α-transitions that do not change the languages and safe languages of its states. Accordingly, $\mathcal{B}_\mathcal{S}$ is safe-minimal iff $\mathcal{A}$ is safe-minimal. Therefore, if $\mathcal{A}$ is not safe-minimal, then applying Step 2 in the construction to $\mathcal{B}_\mathcal{S}$ merges at least two different states. Hence, also in this case, $\mathcal{A}$ is minimized further. □

We formalize relations between tNCWs by means of *isomorphism* and *safe isomorphism*. Consider two tNCWs $\mathcal{A} = \langle \Sigma, Q_\mathcal{A}, q^0_\mathcal{A}, \delta_\mathcal{A}, \alpha_\mathcal{A} \rangle$ and $\mathcal{B} = \langle \Sigma, Q_\mathcal{B}, q^0_\mathcal{B}, \delta_\mathcal{B}, \alpha_\mathcal{B} \rangle$, and a bijection $\kappa : Q_\mathcal{A} \to Q_\mathcal{B}$. We say that κ is:

- *α-transition respecting*, if κ induces a bijection between the α-transitions of $\mathcal{A}$ and $\mathcal{B}$. Formally, for all states $q, q' \in Q_\mathcal{A}$ and letter $\sigma \in \Sigma$, we have that $q' \in \delta^{\alpha_\mathcal{A}}_\mathcal{A}(q,\sigma)$ iff $\kappa(q') \in \delta^{\alpha_\mathcal{B}}_\mathcal{B}(\kappa(q),\sigma)$.
- *$\bar{\alpha}$-transition respecting*, if κ induces a bijection between the $\bar{\alpha}$-transitions of $\mathcal{A}$ and $\mathcal{B}$. Formally, for all states $q, q' \in Q_\mathcal{A}$ and letter $\sigma \in \Sigma$, we have that $q' \in \delta^{\bar{\alpha}_\mathcal{A}}_\mathcal{A}(q,\sigma)$ iff $\kappa(q') \in \delta^{\bar{\alpha}_\mathcal{B}}_\mathcal{B}(\kappa(q),\sigma)$.

Then, $\mathcal{A}$ and $\mathcal{B}$ are *safe isomorphic* if there is a bijection $\kappa : Q_\mathcal{A} \to Q_\mathcal{B}$ that is $\bar{\alpha}$-transition respecting. If, in addition, κ is α-transition respecting, then $\mathcal{A}$ and $\mathcal{B}$ are *isomorphic*. Note that if κ is $\bar{\alpha}$-transition respecting, then for every state $q \in Q_\mathcal{A}$, we have that q and $\kappa(q)$ are safe equivalent. Also, if κ is both α-transition respecting and $\bar{\alpha}$-transition respecting, then for every state $q \in Q_\mathcal{A}$, we have that $q \approx \kappa(q)$.

4.1 Safe isomorphism

Theorem 4.2. *Every two equivalent, nice, and minimal GFG-tNCWs are safe isomorphic.*

Proof. Consider two equivalent, nice, and minimal GFG-tNCWs $\mathcal{A}$ and $\mathcal{B}$. By Theorem 4.1, $\mathcal{A}$ is safe-minimal and safe-centralized. Hence, by Lemma 2.5, there is an injection $\eta : \mathcal{S}(\mathcal{A}) \to \mathcal{S}(\mathcal{B})$ such that for every safe component $T \in \mathcal{S}(\mathcal{A})$, it holds that $|T| \leq |\eta(T)|$. For a safe component $T \in \mathcal{S}(\mathcal{A})$, let p_T be some state in T. By Proposition 2.4, there are states $q_T \in Q_\mathcal{A}$ and $s_T \in Q_\mathcal{B}$ such that $p_T \precsim q_T$ and $q_T \approx s_T$. Since $\mathcal{A}$ is safe-centralized, the state q_T is in T, and in the proof of Lemma 2.5, we defined $\eta(T)$ to be the safe component of s_T in $\mathcal{B}$. Likewise, $\mathcal{B}$ is safe-minimal and safe-centralized, and there is an injection $\eta' : \mathcal{S}(\mathcal{B}) \to \mathcal{S}(\mathcal{A})$. The existence of the two injections implies that $|\mathcal{S}(\mathcal{A})| = |\mathcal{S}(\mathcal{B})|$. Thus, the injection η is actually a bijection. Hence,

$$|\mathcal{A}| = \sum_{T \in \mathcal{S}(\mathcal{A})} |T| \leq \sum_{T \in \mathcal{S}(\mathcal{A})} |\eta(T)| = \sum_{T' \in \mathcal{S}(\mathcal{B})} |T'| = |\mathcal{B}|$$

Indeed, the first inequality follows from the fact $|T| \leq |\eta(T)|$, and the second equality follows from the fact that η is a bijection. Now, as $\mathcal{A}$ and $\mathcal{B}$ are both minimal, we have that $|\mathcal{A}| = |\mathcal{B}|$, and so it follows that for every safe component $T \in \mathcal{S}(\mathcal{A})$, we have that $|T| = |\eta(T)|$. We use the latter fact in order to show that η induces a bijection $\kappa : Q_\mathcal{A} \to Q_\mathcal{B}$ that is $\bar{\alpha}$-transition respecting.

Consider a safe component $T \in \mathcal{S}(\mathcal{A})$. We define a bijection $\kappa_T : T \to \eta(T)$. The desired bijection κ is then the union of the bijections κ_T for $T \in \mathcal{S}(\mathcal{A})$. By Lemma 2.5, we have that $|T| \leq |\eta(T)|$. The proof of the lemma associates with a safe run $r_T = q_0, q_1, \ldots q_m$ of $\mathcal{A}$ that traverses all the states in the safe component T, a safe run $r_{\eta(T)} = s_0, s_1, \ldots s_m$ of $\mathcal{B}$ that traverses states in $\eta(T)$ and $q_i \approx s_i$, for every $1 \leq i \leq m$. Moreover, if $1 \leq i_1, i_2 \leq m$ are such that $q_{i_1} \not\approx q_{i_2}$, then $s_{i_1} \not\approx s_{i_2}$. Now, as $\mathcal{A}$ is safe-minimal, every two states in T are not strongly equivalent. Therefore, the function κ_T that maps each state q_i in r_T to the state s_i in $r_{\eta(T)}$ is an injection from T to $\eta(T)$. Thus, as $|T| = |\eta(T)|$, the injection κ_T is actually a bijection.

Clearly, as $\eta : \mathcal{S}(\mathcal{A}) \to \mathcal{S}(\mathcal{B})$ is a bijection, the function κ that is the union of the bijections κ_T is a bijection from $Q_\mathcal{A}$ to $Q_\mathcal{B}$. We prove that κ is $\bar{\alpha}$-transition respecting. Consider states $q, q' \in Q_\mathcal{A}$ and a letter $\sigma \in \Sigma$ such that $\langle q, \sigma, q' \rangle$ is an $\bar{\alpha}$-transition of $\mathcal{A}$. Let T be q's safe component. By the definition of κ_T, we have that $q \approx \kappa_T(q)$. By Proposition 2.3, there is an $\bar{\alpha}$-transition of $\mathcal{B}$ of the form $t = \langle \kappa(q), \sigma, s' \rangle$, where $q' \approx s'$. As t is an $\bar{\alpha}$-transition of $\mathcal{B}$, we know that s' is in $\eta(T)$. Recall that $\mathcal{B}$ is safe-minimal; in particular, there are no strongly-equivalent states in $\eta(T)$. Hence, $s' = \kappa(q')$, and so $\langle \kappa(q), \sigma, \kappa(q') \rangle$ is an $\bar{\alpha}$-transition of $\mathcal{B}$. Likewise, if $\langle \kappa(q), \sigma, \kappa(q') \rangle$ is an $\bar{\alpha}$-transition of $\mathcal{B}$, then $\langle q, \sigma, q' \rangle$ is an $\bar{\alpha}$-transition of $\mathcal{A}$, and so we are done. $\square$

4.2 Isomorphism

Theorem 4.2 implies that all nice minimal GFG-tNCWs for a given language are safe isomorphic. We continue and show that it is possible to make these GFG-tNCWs isomorphic. We propose two canonical forms that guarantee isomorphism. Both are based on saturating the GFG-NCW with α-transitions. One adds as many α-transitions as possible, and the second does so in a way that preserves α-homogeneity.

Consider a nice GFG-tNCW $\mathcal{A} = \langle \Sigma, Q, q_0, \delta, \alpha \rangle$. We say that a triple $\langle q, \sigma, s \rangle \in Q \times \Sigma \times Q$ is an *allowed transition* in $\mathcal{A}$ if there is a state $s' \in Q$ such that $s \sim s'$ and $\langle q, \sigma, s' \rangle \in \Delta$. Thus, $\langle q, \sigma, s \rangle$ is allowed if there is a state s' equivalent to s such that $s' \in \delta(q, \sigma)$. We now define two types of α-maximality:

- We say that $\mathcal{A}$ is *α-maximal* if all allowed transitions in $Q \times \Sigma \times Q$ are in Δ.
- We say that $\mathcal{A}$ is *α-maximal up to homogeneity* if $\mathcal{A}$ is α-homogenous, and for every state $q \in Q$ and letter $\sigma \in \Sigma$, if q has no outgoing σ-labeled $\bar{\alpha}$-transitions, then all allowed transitions in $\{q\} \times \{\sigma\} \times Q$ are in Δ.

Thus, α-maximal automata include all allowed transitions, and α-maximal up to homogeneity automata include all allowed transitions as long as their inclusion does not conflict with α-homogeneity.

Example 4.1. Recall the minimal GFG-tNCW $\mathcal{B}_S$ appearing in Figure 4. The GFG-tNCWs $\mathcal{C}_1$ and $\mathcal{C}_2$ in Figure 5 are obtained from $\mathcal{B}_S$ by removing a c-labeled α-transition from q_0. This does not change the language and result in two minimal equivalent GFG-tNCWs that are safe isomorphic yet are not α-maximal nor α-maximal up to homogeneity. □

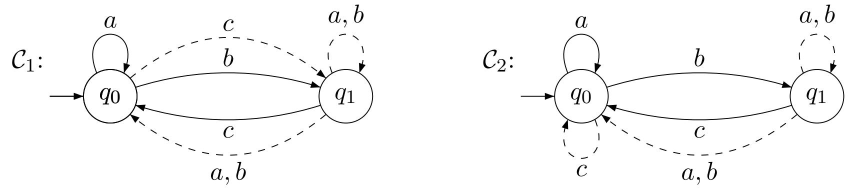

Figure 5: Two safe-isomorphic yet not isomorphic minimal equivalent GFG-tNCWs.

We now see that both types of α-maximality guarantee isomorphism.

Theorem 4.3. *Every two equivalent, nice, minimal, and α-maximal GFG-tNCWs are isomorphic.*

Proof. Consider two equivalent, nice, minimal, and α-maximal GFG-tNCWs $\mathcal{C}_1$ and $\mathcal{C}_2$. By Theorem 4.2, we have that $\mathcal{C}_1$ and $\mathcal{C}_2$ are safe isomorphic. Thus, there is a bijection $\kappa : Q_{\mathcal{C}_1} \to Q_{\mathcal{C}_2}$ that is $\bar{\alpha}$-transition respecting. The bijection κ was defined such that $q \approx \kappa(q)$, for every state $q \in Q_{\mathcal{C}_1}$. We show that κ is also α-transition respecting. Let $\langle q, \sigma, s \rangle$ be an α-transition of $\mathcal{C}_1$. Then, as κ is $\bar{\alpha}$-transition respecting, and $\langle q, \sigma, s \rangle$ is not an $\bar{\alpha}$-transition in $\mathcal{C}_1$, the triple $\langle \kappa(q), \sigma, \kappa(s) \rangle$ cannot be an $\bar{\alpha}$-transition in $\mathcal{C}_2$. We show that $\langle \kappa(q), \sigma, \kappa(s) \rangle$ is a transition in $\mathcal{C}_2$, and thus it has to be an α-transition. As $\mathcal{C}_2$ is nice, in particular total, there is a transition $\langle \kappa(q), \sigma, s' \rangle$ in $\mathcal{C}_2$. As $q \sim \kappa(q)$ and both automata are nice, in particular, symantically deterministic, Proposition 2.1 then implies that $s \sim s'$. Now since $s \sim \kappa(s)$, we get by the transitivity of $\sim$ that $s' \sim \kappa(s)$. Therefore, the existence of the transition $\langle \kappa(q), \sigma, s' \rangle$ in $\mathcal{C}_2$, implies that the transition $\langle \kappa(q), \sigma, \kappa(s) \rangle$ is an allowed transition, and so α-maximality of $\mathcal{C}_2$ implies that it is also a transition in $\mathcal{C}_2$. Likewise, if $\langle \kappa(q), \sigma, \kappa(s) \rangle$ is an α-transition in $\mathcal{C}_2$, then $\langle q, \sigma, s \rangle$ is an α-transition in C_1, and so we are done. □

Theorem 4.4. *Every two equivalent, nice, minimal, and α-maximal up to homogeneity GFG-tNCWs are isomorphic.*

Proof. The proof is identical to that of Theorem 4.3, except that we also have to prove that $\kappa(q)$ has no outgoing σ-labeled $\bar{\alpha}$-transitions in $\mathcal{C}_2$. To see this, assume by way of contradiction that there is an $\bar{\alpha}$-transition $\langle \kappa(q), \sigma, s' \rangle$ in $\mathcal{C}_2$. Then, as $q \approx \kappa(q)$, Proposition 2.3 implies that q has an outgoing σ-labeled $\bar{\alpha}$-transition in $\mathcal{C}_1$, contradicting the fact that $\mathcal{C}_1$ is α-homogenous. □

5 Obtaining Canonical Minimal GFG-tNCWs

In this section we show how the two types of canonical minimal GFG-tNCWs can be obtained in polynomial time. We start with α-maximality up to homogeneity and show that such an α-maximization is performed by the minimization construction of [1]. We continue with α-maximality, show that adding allowed transitions to a GFG-tNCW does not change its language, and conclude that α-maximization can be performed on top of the minimization construction of [1].

5.1 Obtaining canonical minimal α-maximal up to homogeneity GFG-tNCWs

Theorem 5.1. *Consider a nice GFG-tNCW $\mathcal{A}$, and let $\mathcal{C}$ be the minimal GFG-tNCW produced from $\mathcal{A}$ by the minimization construction of [1]. Then, $\mathcal{C}$ is α-maximal up to homogeneity.*

Proof. Consider the minimization construction of [1]. We first show that the safe-centralized GFG-tNCW $\mathcal{B}_\mathcal{S}$, defined in Step 1, is α-maximal up to homogeneity. Then, we show that α-maximality up to homogeneity is maintained in the transition to the GFG-tNCW $\mathcal{C}$, defined in Step 2. By Theorem 3.4, we know that $\mathcal{B}_\mathcal{S}$ is α-homogenous. Assume that q is a state in $\mathcal{B}_\mathcal{S}$ with no outgoing σ-labeled $\bar{\alpha}$-transitions, and assume that $\langle q,\sigma,s\rangle$ is an allowed transition. We need to show that $\langle q,\sigma,s\rangle$ is a transition in $\mathcal{B}_\mathcal{S}$. As $\langle q,\sigma,s\rangle$ is an allowed transition, there is a transition $\langle q,\sigma,s'\rangle$ in $\mathcal{B}_\mathcal{S}$ with $s\sim_{\mathcal{B}_\mathcal{S}} s'$, and by the assumption, $\langle q,\sigma,s'\rangle$ has to be an α-transition. By the definition of the transition function of $\mathcal{B}_\mathcal{S}$, we have that $s'\sim_\mathcal{A} q'$ for some state $q'\in\delta^\alpha_\mathcal{A}(q,\sigma)$. As $\mathcal{A}$ is semantically deterministic, we get that the state s' is $\mathcal{A}$-equivalent to every state in $\delta^\alpha_\mathcal{A}(q,\sigma)$. So again, by the definition of the transition function of $\mathcal{B}_\mathcal{S}$, we can write $\delta^\alpha_\mathcal{S}(q,\sigma)=\{p\in Q_\mathcal{S} : p\sim_\mathcal{A} q'\}$. Now, as $s\sim_{\mathcal{B}_\mathcal{S}} s'$, Proposition 3.3 implies that $L(\mathcal{A}^s)=L(\mathcal{B}^s_\mathcal{S})=L(\mathcal{B}^{s'}_\mathcal{S})=L(\mathcal{A}^{s'})$; that is, $s\sim_\mathcal{A} s'$, and since $s'\sim_\mathcal{A} q'$, we get by the transitivity of $\sim_\mathcal{A}$ that $s\sim_\mathcal{A} q'$, and so $\langle q,\sigma,s\rangle$ is a transition in $\mathcal{B}_\mathcal{S}$.

We show next that the GFG-tNCW $\mathcal{C}$ is α-maximal up to homogeneity. By Theorem 3.6, we have that $\mathcal{C}$ is α-homogenous. Assume that $[q]$ is a state in $\mathcal{C}$ with no outgoing σ-labeled $\bar{\alpha}$-transitions, and assume that $\langle[q],\sigma,[s]\rangle$ is an allowed transition. We need to show that $\langle[q],\sigma,[s]\rangle$ is a transition in $\mathcal{C}$. As $\langle[q],\sigma,[s]\rangle$ is an allowed transition, there is a transition $\langle[q],\sigma,[s']\rangle$ in $\mathcal{C}$ with $[s]\sim[s']$. Thus, by Proposition 3.5, we have that $L(\mathcal{B}^s_\mathcal{S})=L(\mathcal{C}^{[s]})=L(\mathcal{C}^{[s']})=L(\mathcal{B}^{s'}_\mathcal{S})$; that is, $s'\sim_{\mathcal{B}_\mathcal{S}} s$. By the assumption, $\langle[q],\sigma,[s']\rangle$ has to be an α-transition. Therefore, by the definition of $\mathcal{C}$, there are states $q''\in[q]$ and $s''\in[s']$, such that $\langle q'',\sigma,s''\rangle$ is an α-transition in $\mathcal{B}_\mathcal{S}$. Now, by transitivity of $\sim_{\mathcal{B}_\mathcal{S}}$ and the fact that $s''\sim_{\mathcal{B}_\mathcal{S}} s'$, we get that $s''\sim_{\mathcal{B}_\mathcal{S}} s$. Finally, as $\mathcal{B}_\mathcal{S}$ is α-homogenous, we get that q'' has no outgoing σ-labeled $\bar{\alpha}$-transitions in $\mathcal{B}_\mathcal{S}$, and so by the α-maximality up to homogeneity of $\mathcal{B}_\mathcal{S}$, we have that $\langle q'',\sigma,s\rangle$ is a transition in $\mathcal{B}_\mathcal{S}$. Therefore, by the definition of $\mathcal{C}$, we have that $\langle[q],\sigma,[s]\rangle$ is a transition in $\mathcal{C}$, and we are done. □

We can thus conclude with the following.

Theorem 5.2. *Every GFG-tNCW $\mathcal{A}$ can be canonized into a nice minimal α-maximal up to homogeneity GFG-tNCW in polynomial time.*

5.2 Obtaining canonical minimal α-maximal GFG-tNCWs

Consider a nice GFG-tNCW $\mathcal{A}=\langle\Sigma,Q,q_0,\delta,\alpha\rangle$. We say that a set of triples $\mathcal{E}\subseteq Q\times\Sigma\times Q$ is an *allowed set* if all the triples in it are allowed transitions in $\mathcal{A}$. For every set $\mathcal{E}\subseteq Q\times\Sigma\times Q$, we define the tNCW $\mathcal{A}_\mathcal{E}=\langle\Sigma,Q,q_0,\delta_\mathcal{E},\alpha_\mathcal{E}\rangle$, where $\Delta_\mathcal{E}=\Delta\cup\mathcal{E}$ and $\alpha_\mathcal{E}=\alpha\cup\mathcal{E}$. Clearly, as $\mathcal{A}$ and $\mathcal{A}_\mathcal{E}$ have the same set of states and the same set of $\bar{\alpha}$-transitions, they are safe equivalent.

In Propositions 5.4 and 5.5 below, we prove that for every allowed set $\mathcal{E}$, we have that $\mathcal{A}_{\mathcal{E}}$ is a nice GFG-tNCW equivalent to $\mathcal{A}$. We first extend Proposition 2.1 to the setting of $\mathcal{A}$ and $\mathcal{A}_{\mathcal{E}}$:

Proposition 5.3. *Consider states q and s of $\mathcal{A}$ and $\mathcal{A}_{\mathcal{E}}$, respectively, a letter $\sigma \in \Sigma$, and transitions $\langle q,\sigma,q'\rangle$ and $\langle s,\sigma,s'\rangle$ of $\mathcal{A}$ and $\mathcal{A}_{\mathcal{E}}$, respectively. If $q \sim_{\mathcal{A}} s$, then $q' \sim_{\mathcal{A}} s'$.*

Proof. If $\langle s,\sigma,s'\rangle \notin \mathcal{E}$, then, by the definition of $\Delta_{\mathcal{E}}$, it is also a transition of $\mathcal{A}$. Hence, since $q \sim_{\mathcal{A}} s$ and $\mathcal{A}$ is nice, in particular, semantically deterministic, Proposition 2.1 implies that $q' \sim_{\mathcal{A}} s'$. If $\langle s,\sigma,s'\rangle \in \mathcal{E}$, then, by the definition of $\Delta_{\mathcal{E}}$, it is an allowed transition of $\mathcal{A}$. Therefore, there is a state $p' \in Q$ such that $s' \sim_{\mathcal{A}} p'$ and $\langle s,\sigma,p'\rangle \in \Delta$. As $q \sim_{\mathcal{A}} s$ and $\mathcal{A}$ is semantically deterministic, Proposition 2.1 implies that $q' \sim_{\mathcal{A}} p'$. Therefore, using the fact that $p' \sim_{\mathcal{A}} s'$, the transitivity of $\sim_{\mathcal{A}}$ implies that $q' \sim_{\mathcal{A}} s'$, and so we are done. □

Proposition 5.4. *Let p and s be states of $\mathcal{A}$ and $\mathcal{A}_{\mathcal{E}}$, respectively, with $p \sim_{\mathcal{A}} s$. Then, $\mathcal{A}^s_{\mathcal{E}}$ is a GFG-tNCW equivalent to $\mathcal{A}^p$.*

Proof. We first prove that $L(\mathcal{A}^s_{\mathcal{E}}) \subseteq L(\mathcal{A}^p)$. Consider a word $w = \sigma_1\sigma_2\ldots \in L(\mathcal{A}^s_{\mathcal{E}})$, and let $s_0,s_1,s_2,\ldots$ be an accepting run of $\mathcal{A}^s_{\mathcal{E}}$ on w. Then, there is $i \geq 0$ such that $s_i,s_{i+1},\ldots$ is a safe run of $\mathcal{A}^{s_i}_{\mathcal{E}}$ on the suffix $w[i+1,\infty]$. Let $p_0,p_1,\ldots p_i$ be a run of $\mathcal{A}^p$ on the prefix $w[1,i]$. Since $p_0 \sim_{\mathcal{A}} s_0$, we get, by an iterative application of Proposition 5.3, that $p_i \sim_{\mathcal{A}} s_i$. In addition, as the run of $\mathcal{A}^{s_i}_{\mathcal{E}}$ on the suffix $w[i+1,\infty]$ is safe, it is also a safe run of $\mathcal{A}^{s_i}$. Hence, $w[i+1,\infty] \in L(\mathcal{A}^{p_i})$, and thus $p_0,p_1,\ldots,p_i$ can be extended to an accepting run of $\mathcal{A}^p$ on w.

Next, as $\mathcal{A}$ is nice, all of its states are GFG, in particular, there is a strategy f^s witnessing $\mathcal{A}^s$'s GFGness. Recall that $\mathcal{A}$ is embodied in $\mathcal{A}_{\mathcal{E}}$. Therefore, every run in $\mathcal{A}$ exists also in $\mathcal{A}_{\mathcal{E}}$. Thus, as $p \sim_{\mathcal{A}} s$, we get that for every word $w \in L(\mathcal{A}^p)$, the run $f^s(w)$ is an accepting run of $\mathcal{A}^s$ on w, and thus is also an accepting run of $\mathcal{A}^s_{\mathcal{E}}$ on w. Hence, $L(\mathcal{A}^p) \subseteq L(\mathcal{A}^s_{\mathcal{E}})$ and f^s witnesses $\mathcal{A}^s_{\mathcal{E}}$'s GFGness.

□

Proposition 5.5. *For every allowed set $\mathcal{E}$, the GFG-tNCW $\mathcal{A}_{\mathcal{E}}$ is nice.*

Proof. It is easy to see that the fact $\mathcal{A}$ is nice implies that $\mathcal{A}_{\mathcal{E}}$ is normal and safe deterministic. Also, as $\mathcal{A}$ is embodied in $\mathcal{A}_{\mathcal{E}}$ and both automata have the same state-space and initial states, then all the states in $\mathcal{A}_{\mathcal{E}}$ are reachable. Finally, Proposition 5.4 implies that all the states in $\mathcal{A}_{\mathcal{E}}$ are GFG. To conclude that $\mathcal{A}_{\mathcal{E}}$ is nice, we prove below that it is semantically deterministic. Consider transitions $\langle q,\sigma,s_1\rangle$ and $\langle q,\sigma,s_2\rangle$ in $\Delta_{\mathcal{E}}$. We need to show that $s_1 \sim_{\mathcal{A}_{\mathcal{E}}} s_2$. By the definition of $\Delta_{\mathcal{E}}$, there are transitions $\langle q,\sigma,s'_1\rangle$ and $\langle q,\sigma,s'_2\rangle$ in Δ for states s'_1 and s'_2 such that $s_1 \sim_{\mathcal{A}} s'_1$ and $s_2 \sim_{\mathcal{A}} s'_2$. As $\mathcal{A}$ is nice, in particular, semantically deterministic, we have that $s'_1 \sim_{\mathcal{A}} s'_2$. Hence, as $s_1 \sim_{\mathcal{A}} s'_1$ and $s'_2 \sim_{\mathcal{A}} s_2$, we get by the transitivity of $\sim_{\mathcal{A}}$ that $s_1 \sim_{\mathcal{A}} s_2$. Then, Proposition 5.4 implies that $L(\mathcal{A}^{s_1}) = L(\mathcal{A}^{s_1}_{\mathcal{E}})$ and $L(\mathcal{A}^{s_2}) = L(\mathcal{A}^{s_2}_{\mathcal{E}})$, and so we get that $s_1 \sim_{\mathcal{A}_{\mathcal{E}}} s_2$. Thus, $\mathcal{A}_{\mathcal{E}}$ is semantically deterministic. □

Let $\mathcal{C}$ be a nice minimal GFG-tNCW equivalent to $\mathcal{A}$, and let $\hat{\mathcal{E}}$ be the set of all allowed transitions in $\mathcal{C}$. By Propositions 5.4 and 5.5, we have that $\mathcal{C}_{\hat{\mathcal{E}}}$ is a nice minimal GFG-tNCW equivalent to $\mathcal{A}$. Below we argue that it is also α-maximal.

Proposition 5.6. *Let $\mathcal{C}$ be a nice GFG-tNCW, and let $\hat{\mathcal{E}}$ be the set of all allowed transitions in $\mathcal{C}$. Then, $\mathcal{C}_{\hat{\mathcal{E}}}$ is α-maximal.*

Proof. Let $\mathcal{C} = \langle \Sigma, Q, q_0, \delta, \alpha \rangle$, and consider an allowed transition $\langle q, \sigma, s \rangle \in Q \times \Sigma \times Q$ in $\mathcal{C}_{\hat{\mathcal{E}}}$. We prove that $\langle q, \sigma, s \rangle$ is an allowed transition also in $\mathcal{C}$. Hence, it is in $\hat{\mathcal{E}}$, and thus is a transition in $\mathcal{C}_{\hat{\mathcal{E}}}$.

By the definition of allowed transitions, there is a state $s' \in Q$ with $s \sim_{\mathcal{C}_{\hat{\mathcal{E}}}} s'$ such that $s' \in \delta_{\hat{\mathcal{E}}}(q, \sigma)$. Proposition 5.4 implies that $L(\mathcal{C}^s) = L(\mathcal{C}^s_{\hat{\mathcal{E}}}) = L(\mathcal{C}^{s'}_{\hat{\mathcal{E}}}) = L(\mathcal{C}^{s'})$, and thus $s \sim_{\mathcal{C}} s'$. Also, by the definition of $\delta_{\hat{\mathcal{E}}}$, there is a state $s'' \in Q$ such that $s'' \sim_{\mathcal{C}} s'$ and $s'' \in \delta(q, \sigma)$. Therefore, as the transitivity of $\sim_{\mathcal{C}}$ implies that $s \sim_{\mathcal{C}} s''$, we have that $\langle q, \sigma, s \rangle$ is also an allowed transition in $\mathcal{C}$, and we are done. □

Since the relation $\sim$ can be calculated in polynomial time [9, 12], and so checking if a triple in $Q \times \Sigma \times Q$ is an allowed transition can be done in polynomial time, then applying α-maximization on top of the minimization construction of [1] is still polynomial. We can thus conclude with the following.

Theorem 5.7. *Every GFG-tNCW $\mathcal{A}$ can be canonized into a nice minimal α-maximal GFG-tNCW in polynomial time.*

Example 5.1. By applying α-maximization to the GFG-tNCW $\mathcal{B}_{\mathcal{S}}$, we obtained the α-maximal GFG-tNCW $\mathcal{C}_{\hat{\mathcal{E}}}$ appearing in Figure 6 □

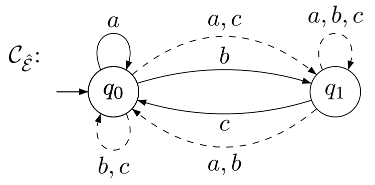

Figure 6: The α-maximal GFG-tNCW for the GFG-tNCW $\mathcal{B}_{\mathcal{S}}$ in Figure 4.

6 Canonicity in tDCW and tDBW

For deterministic automata with state-based acceptance, an analogue definition of isomorphism between automata $\mathcal{A}$ and $\mathcal{B}$ with acceptance conditions $\alpha_{\mathcal{A}} \subseteq Q_{\mathcal{A}}$ and $\alpha_{\mathcal{B}} \subseteq Q_{\mathcal{B}}$, seeks a bijection $\kappa : Q_{\mathcal{A}} \to Q_{\mathcal{B}}$ such that for every $q \in Q_{\mathcal{A}}$, we have that $q \in \alpha_{\mathcal{A}}$ iff $\kappa(q) \in \alpha_{\mathcal{B}}$, and for every letter $\sigma \in \Sigma$, and state $q' \in Q_{\mathcal{A}}$, we have that $q' \in \delta_{\mathcal{A}}(q, \sigma)$ iff $\kappa(q') \in \delta_{\mathcal{B}}(\kappa(q), \sigma)$. It is easy to see that the DCWs $\mathcal{A}_1$ and $\mathcal{A}_2$ from Figure 1 are not isomorphic, which is a well known property of DCWs and DBWs [13]. In Theorem 6.1 below, we extend the "no canonicity" result to GFG-NCWs.

Theorem 6.1. *Nice, equivalent, and minimal GFG-NCWs need not be isomorphic.*

Proof. Consider the language $L = (a + b)^* \cdot (a^\omega + b^\omega)$. In Figure 1, we described the non-isomorphic DCWs $\mathcal{A}_1$ and $\mathcal{A}_2$ for L. The DCWs $\mathcal{A}$ and $\mathcal{B}$ can be viewed as nice GFG-NCWs. It is not hard to see that there is no 2-state GFG-NCW for L, implying that $\mathcal{A}$ and $\mathcal{B}$ are nice, equivalent, and minimal GFG-NCWs that are not isomorphic, as required. □

In Example 4.1 we saw that nice, equivalent, and minimal GFG-tNCWs need not be isomorphic too, yet they may be made isomorphic by α-maximization. For the GFG-NCWs in the proof of Theorem 6.1, this does not work for every definition of α-maxization that makes sense: we cannot add transitions and make the automata isomorphic. This suggests that the consideration of automata with transition-based acceptance is more crucial for canonization than the consideration of GFG automata, and makes the study of canonization for tDCWs very interesting. In particular, unlike the case of GFG automata, here results on tDCWs immediately apply also to tDBWs. We start with some bad news, showing that there is no canonicity also in the transition-based setting.

Theorem 6.2. *Nice, equivalent, and minimal tDCWs and tDBWs need not be isomorphic.*

Proof. The GFG-tNCW $\mathcal{B}_{\mathcal{S}}$ from Figure 4 is DBP. In Figure 7 below, we describe two tDCWs obtained from it by two different prunnings. It is not hard to see that both tDCWs are equivalent to $\mathcal{B}_{\mathcal{S}}$, yet are not isomorphic.

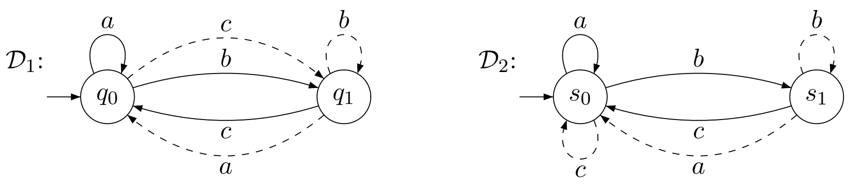

Figure 7: Two non-isomorphic equivalent minimal nice tDCWs, obtained by different prunnings of $\mathcal{B}_{\mathcal{S}}$.

By removing the a-labeled transitions from the tDCWs in Figure 7, we obtain a simpler example. Consider the tDCWs $\mathcal{D}'_1$ and $\mathcal{D}'_2$ in Figure 8. It is easy to see that $L(\mathcal{D}'_1) = L(\mathcal{D}'_2) = (b+c)^* \cdot (b \cdot c)^\omega$. Clearly, there is no single-state tDCW for this language. Also, the tDCWs are not isomorphic, as a candidate bijection κ has to be $\bar{\alpha}$-transition respecting, and thus have $\kappa(q_0) = s_0$ and $\kappa(q_1) = s_1$, yet then it is not α-transition respecting. By dualizing the acceptance condition of $\mathcal{D}'_1$ and $\mathcal{D}'_2$, we obtain two non-isomorphic tDBWs for the complement language, of all words with infinitely many occurrences of *bb* or *cc*. □

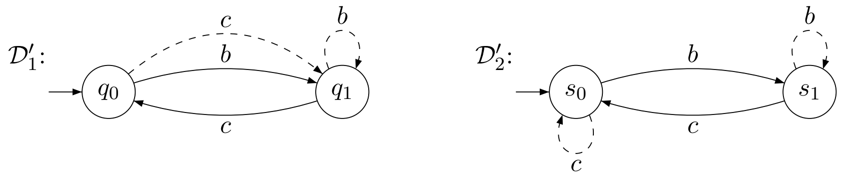

Figure 8: Two nice, minimal, equivalent, and non-isomorphic tDCWs

The GFG-NCWs used in the proof of Theorem 6.1 cannot be made isomorphic by changing membership of states in α or by adding transitions. Likewise, since tDCWs cannot be α-maximized, as adding transitions conflicts with determinism, the tDCWs used in the proof of Theorem 6.2 cannot be made isomorphic either. Hence, we have the following.

Theorem 6.3. *There is no canonicity for minimal GFG-NCWs and for minimal tDCWs.*

The tDCWs in the proof of Theorem 6.2 are safe isomorphic, We continue and study safe-isomorphism between minimal tDCWs. Here too, we restrict attention to *nice* minimal tDCWs. Note that here, some of the properties of nice GFG-tNCWs are trivial: being minimal and deterministic, then clearly all states are reachable and GFG, the automata are semantically deterministic and safe deterministic, and we only have to make them normal by classifying transitions between safe components as α-transitions.

We say that an ω-regular language L is *tDCW-positive* if a minimal tDCW for L is not bigger than a minimal GFG-tNCW for L. Thus, tDCWs for L are as succinct as GFG-tNCWs for it.

Theorem 6.4. *Consider an ω-regular language L. If L is tDCW-positive, then every two nice and minimal tDCWs for L are safe isomorphic.*

Proof. Consider a language L that is tDCW-positive, and consider two nice minimal tDCWs $\mathcal{A}_1$ and $\mathcal{A}_2$ for L. Since L is tDCW-positive, then $\mathcal{A}_1$ and $\mathcal{A}_2$ are also nice minimal GFG-tNCWs for L. Hence, by Theorem 4.2, they are safe isomorphic. $\square$

Note that safe isomorphism for ω-regular languages that are not tDCW-positive is left open. Theorem 6.4 suggests that searching for a language L that has two minimal tDCWs that are not safe isomorphic, we can restrict attention to languages that are not tDCW-positive. Such languages are not natural. Moreover, their canonicity is less crucial, as working with a minimal GFG-tNCW for them is more appealing. Examples of languages that are not tDCW-positive can be found in [12], where it was shown that GFG-tNCWs may be exponentially more succinct than tDCWs.

7 Glossary

All notations and definitions refer to a GFG-tNCW $\mathcal{A} = \langle \Sigma, Q, q_0, \delta, \alpha \rangle$.

Relations between states

- Two states $q, s \in Q$ are *equivalent*, denoted $q \sim s$, if $L(\mathcal{A}^q) = L(\mathcal{A}^s)$.
- Two states $q, s \in Q$ are *safe equivalent* if $L_{safe}(\mathcal{A}^q) = L_{safe}(\mathcal{A}^s)$.
- Two states $q, s \in Q$ are *strongly-equivalent*, denoted $q \approx s$, if $q \sim s$ and $L_{safe}(\mathcal{A}^q) = L_{safe}(\mathcal{A}^s)$.
- A state $q \in Q$ is *subsafe-equivalent to* a state s, denoted $q \precsim s$, if $q \sim s$ and $L_{safe}(\mathcal{A}^q) \subseteq L_{safe}(\mathcal{A}^s)$.

Properties of a GFG-tNCW

- $\mathcal{A}$ is *semantically deterministic* if for every state $q \in Q$ and letter $\sigma \in \Sigma$, all the σ-successors of q are equivalent: for every two states $s, s' \in \delta(q, \sigma)$, we have that $s \sim s'$.
- $\mathcal{A}$ is *safe deterministic* if by removing its α-transitions, we get a (possibly not total) deterministic automaton. Thus, for every state $q \in Q$ and letter $\sigma \in \Sigma$, it holds that $|\delta^{\bar{\alpha}}(q, \sigma)| \leq 1$.
- $\mathcal{A}$ is *normal* if there are no $\bar{\alpha}$-transitions connecting different safe components. That is, for all states q and s of $\mathcal{A}$, if there is a path of $\bar{\alpha}$-transitions from q to s, then there is also a path of $\bar{\alpha}$-transitions from s to q.
- $\mathcal{A}$ is *nice* if all the states in $\mathcal{A}$ are reachable and GFG, and $\mathcal{A}$ is normal, safe deterministic, and semantically deterministic.
- $\mathcal{A}$ is *α-homogenous* if for every state $q \in Q$ and letter $\sigma \in \Sigma$, either $\delta^{\alpha}(q, \sigma) = \emptyset$ or $\delta^{\bar{\alpha}}(q, \sigma) = \emptyset$.
- $\mathcal{A}$ is *safe-minimal* if it has no strongly-equivalent states.
- $\mathcal{A}$ is *safe-centralized* if for every two states $q, s \in Q$, if $q \precsim s$, then q and s are in the same safe component of $\mathcal{A}$.

References

[1] B. Abu Radi & O. Kupferman (2019): *Minimizing GFG Transition-Based Automata*. In: *Proc. 46th Int. Colloq. on Automata, Languages, and Programming*, *LIPIcs* 132, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp. 100:1–100:16, doi:10.4230/LIPIcs.ICALP.2019.100.

[2] M. Bagnol & D. Kuperberg (2018): *Büchi Good-for-Games Automata Are Efficiently Recognizable*. In: *Proc. 38th Conf. on Foundations of Software Technology and Theoretical Computer Science*, *LIPIcs* 122, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp. 16:1–16:14, doi:10.4230/LIPIcs.FSTTCS.2018.16.

[3] U. Boker, D. Kuperberg, O. Kupferman & M. Skrzypczak (2013): *Nondeterminism in the Presence of a Diverse or Unknown Future*. In: *ICALP (2)*, *Lecture Notes in Computer Science* 7966, Springer, pp. 89–100, doi:10.1007/978-3-642-39212-2_11.

[4] U. Boker, O. Kupferman & M. Skrzypczak (2017): *How Deterministic are Good-For-Games Automata?* In: *Proc. 37th Conf. on Foundations of Software Technology and Theoretical Computer Science*, *Leibniz International Proceedings in Informatics (LIPIcs)* 93, pp. 18:1–18:14, doi:10.4230/LIPIcs.FSTTCS.2017.18.

[5] J.R. Büchi (1962): *On a Decision Method in Restricted Second Order Arithmetic*. In: *Proc. Int. Congress on Logic, Method, and Philosophy of Science. 1960*, Stanford University Press, pp. 1–12.

[6] Th. Colcombet (2009): *The theory of stabilisation monoids and regular cost functions*. In: *Proc. 36th Int. Colloq. on Automata, Languages, and Programming*, *Lecture Notes in Computer Science* 5556, Springer, pp. 139–150, doi:10.1007/978-3-642-02930-1_12.

[7] A. Duret-Lutz, A. Lewkowicz, A. Fauchille, Th. Michaud, E. Renault & L. Xu (2016): *Spot 2.0 — a framework for LTL and ω-automata manipulation*. In: *14th Int. Symp. on Automated Technology for Verification and Analysis*, *Lecture Notes in Computer Science* 9938, Springer, pp. 122–129, doi:10.1007/978-3-319-46520-3_8.

[8] D. Giannakopoulou & F. Lerda (2002): *From States to Transitions: Improving Translation of LTL Formulae to Büchi Automata*. In: *Proc. 22nd International Conference on Formal Techniques for Networked and Distributed Systems*, *Lecture Notes in Computer Science* 2529, Springer, pp. 308–326, doi:10.1007/3-540-36135-9_20.

[9] T.A. Henzinger, O. Kupferman & S. Rajamani (2002): *Fair simulation*. *Information and Computation* 173(1), pp. 64–81, doi:10.1006/inco.2001.3085.

[10] T.A. Henzinger & N. Piterman (2006): *Solving Games without Determinization*. In: *Proc. 15th Annual Conf. of the European Association for Computer Science Logic*, *Lecture Notes in Computer Science* 4207, Springer, pp. 394–410, doi:10.1007/11874683_26.

[11] J.E. Hopcroft (1971): *An $n \log n$ algorithm for minimizing the states in a finite automaton*. In Z. Kohavi, editor: *The Theory of Machines and Computations*, Academic Press, pp. 189–196, doi:10.1016/B978-0-12-417750-5.50022-1.

[12] D. Kuperberg & M. Skrzypczak (2015): *On Determinisation of Good-for-Games Automata*. In: *Proc. 42nd Int. Colloq. on Automata, Languages, and Programming*, pp. 299–310, doi:10.1007/978-3-662-47666-6_24.

[13] O. Kupferman (2015): *Automata Theory and Model Checking*. *Handbook of Theoretical Computer Science*.

[14] O. Kupferman, S. Safra & M.Y. Vardi (2006): *Relating word and tree automata*. *Ann. Pure Appl. Logic* 138(1-3), pp. 126–146, doi:10.1016/j.apal.2005.06.009.

[15] K. Lehtinen & M. Zimmermann (2020): *Good-for-games ω-Pushdown Automata*. In: *Proc. 35th IEEE Symp. on Logic in Computer Science*, pp. 689–702, doi:10.1145/3373718.3394737.

[16] W. Li, Sh. Kan & Z. Huang (2017): *A Better Translation From LTL to Transition-Based Generalized Büchi Automata*. *IEEE Access* 5, pp. 27081–27090, doi:10.1109/ACCESS.2017.2773123.

[17] G. Morgenstern (2003): *Expressiveness results at the bottom of the ω-regular hierarchy*. M.Sc. Thesis, The Hebrew University.

[18] J. Myhill (1957): *Finite automata and the representation of events*. Technical Report WADD TR-57-624, pages 112–137, Wright Patterson AFB, Ohio.

[19] A. Nerode (1958): *Linear Automaton Transformations*. *Proceedings of the American Mathematical Society* 9(4), pp. 541–544, doi:10.2307/2033204.

[20] D. Niwinski & I. Walukiewicz (1998): *Relating hierarchies of word and tree automata*. In: *Proc. 15th Symp. on Theoretical Aspects of Computer Science*, *Lecture Notes in Computer Science* 1373, Springer, pp. 320–331, doi:10.1007/BFb0028571.

[21] S. Schewe (2010): *Beyond Hyper-Minimisation—Minimising DBAs and DPAs is NP-Complete*. In: *Proc. 30th Conf. on Foundations of Software Technology and Theoretical Computer Science*, *Leibniz International Proceedings in Informatics (LIPIcs)* 8, pp. 400–411, doi:10.4230/LIPIcs.FSTTCS.2010.400.

[22] S. Schewe (2020): *Minimising Good-for-Games automata is NP complete*. *CoRR* abs/2003.11979.

[23] S. Sickert, J. Esparza, S. Jaax & J. Křetínský (2016): *Limit-Deterministic Büchi Automata for Linear Temporal Logic*. In: *Proc. 28th Int. Conf. on Computer Aided Verification*, *Lecture Notes in Computer Science* 9780, Springer, pp. 312–332, doi:10.1007/978-3-319-41540-6_17.

[24] R.E. Tarjan (1972): *Depth first search and linear graph algorithms*. *SIAM Journal of Computing* 1(2), pp. 146–160, doi:10.1137/0201010.

[25] M.Y. Vardi & P. Wolper (1994): *Reasoning about Infinite Computations*. *Information and Computation* 115(1), pp. 1–37, doi:10.1006/inco.1994.1092.