An improved analysis of Piterman's Streett determinization construction.

Thomas Varghese*

Abstract

In 2006, Piterman proposed improved constructions for determinizing Büchi and Streett automata to parity automata. An improved counting of the states analysis was provided for the construction soon after for the determinization construction from Büchi to parity automata. In this manuscript, we first describe Piterman's construction and show how the improved counting was performed. We adapt the techniques that are used to improve this estimation to the Streett determinization construction.

1 Preliminaries

We assume familiarity with the notions of regular and ω-regular languages, nondeterministic and deterministic automata, their transformations, runs, transitions etc. We describe here the types of acceptance we study:

Büchi condition. A run of an ω-automaton is accepting according to the Büchi condition α if the set of infinitely recurring states intersects the set of accepting states. We can represent this as

$$\bigvee_{s\in\alpha} s.$$

In other words, at least one of the states in α has to be visited infinitely often during the run of the automaton.

Rabin condition. Let $\mathcal{A} = (S, \Sigma, \delta, s_0, \alpha)$ be an ω-automaton. The Rabin acceptance condition α is a set of pairs $(E_1, F_1), \ldots (E_k, F_k)$, where for all i, we have $E_i \subseteq S$ and $F_i \subseteq S$. We call k the index of the Rabin condition. A run is accepting according to the Rabin condition α if for some i, there exists a pair (E_i, F_i) such that the run visits F_i infinitely many times and visits E_i only finitely often. We can represent this as

$$\bigvee_{i} F_i \wedge \neg E_i.$$

The Rabin condition is also known as the 'pairs condition'.

Streett condition. The Streett acceptance condition is the dual of the Rabin condition. Let $\mathcal{A} = (S, \Sigma, \delta, s_0, \alpha)$ be an ω-automaton. The Streett acceptance condition α is a set of pairs $(E_1, F_1, \ldots (E_k, F_k)$, where for all i, we have $E_i \subseteq S$ and $F_i \subseteq S$. We call k the index of the Streett condition. A run is accepting according to the Streett condition α, if for every pair (E_i, F_i) the run either visits F_i finitely often or visits E_i infinitely often. We can also say that the run is accepting if for all

*This work stems from research for the author's Master thesis while at the University of Leicester supervised by Nir Piterman who proposed the high level ideas and provided intuition.

pairs i, if the run visits F_i infinitely many times, it must also visit E_i infinitely many times. We can write this as

$$\bigwedge_i F_i \rightarrow E_i.$$

The Streett condition is also known as the 'complemented pairs condition'.

Parity condition. Let $\mathcal{A} = (S, \Sigma, \delta, s_0, \alpha)$ be an ω-automaton. The parity acceptance condition α is a partition $\{F_0, \ldots F_k\}$ of S. We call k the index of the parity condition. A run is accepting according to the parity condition α, if for some even i, we have $inf(r) \cap F_i \neq \emptyset$ and for all $i' < i$ we have $inf(r) \cap F_{i'} = \emptyset$.

Notation. We use after [1] the following abbreviations in $\{N, D\} \times \{B, M, R, S, P\} \times \{W, T\}$ to denote automata. N and D stand for non-deterministic and deterministic respectively. The second symbol denotes the acceptance condition: B – Büchi, M – Müller, R – Rabin, S – Streett and P – Parity. The last symbol stands for the object the automaton is reading, ie. W for words and T for trees. So, for example, an NSW is a non-deterministic Streett word automaton and a DPW is a deterministic parity word automaton.

2 From NBW to DPW - Piterman's construction

This determinisation construction is an extension of Safra's construction [2]. As in Safra's determinisation construction, the principal structure of the states of the resulting deterministic automaton is a tree called a *compact Safra tree* [1]. The difference from Safra's construction is that we replace the static node names with dynamic names that decrease as nodes below get erased from the tree, essentially getting rid of the 'older-than' function. Thus, older nodes have smaller names. A natural parity condition is enforced, with the erasure of nodes being seen as a bad event, while finding that a node's label is the same as the union of labels of its descendants is a good event. Using dynamic node names allows us to simulate the *index appearance record* (which allows for translation of Rabin and Streett conditions to parity conditions) within the deterministic automaton. Piterman's construction takes a Büchi automaton with n states and produces a deterministic parity automaton with $2n^n n!$ states and $2n$ parity indices [1] [1].

2.1 Construction

Let $\mathcal{N} = (S, \Sigma, s_0, \delta, \alpha)$ be an NBW with $|S| = n$ states. We construct an equivalent DPW $\mathcal{D} = (D, \Sigma, d_0, \rho, \alpha')$. We first define the structure of the states of the resulting DPW.

A *compact Safra tree t* over S is $(N, 1, p, l, c)$ defined as follows.

- $N \subseteq [n]$ is a finite set of nodes such that if $i \in N$, then $i - 1 \in N$. We may temporarily have to use nodes in $[n + 1 \ldots 2n]$
- $1 \in N$ is the root.
- $p : (N \setminus \{1\}) \rightarrow N$ is the parent function, such that for every $i > 1, p(i) < i$.
- $l : N \rightarrow 2^S$ is the labelling function such that for every $i, l(i) \subset l(p(i))$, and, if $p(i) < p(j)$, $\exists s$ such that $s \in l(i)$ and $s \notin l(j)$.
- $c \in \{-|V|, ..., |V|\} \setminus \{-1\}$ is the colouring function, used to define the parity acceptance condition.

[1] In [5], we prove that this construction is tight to the state.

The components of the DPW $\mathcal{D} = (D, \Sigma, d_0, \rho, \alpha')$ are as follows:

- D is the set of compact Safra trees over S.
- d_0 is the tree with the single node 1 labelled by $\{s_0\}$.
- For every tree $d \in D$ and $\sigma \in \Sigma$, the transition $d' = \rho(d, \sigma)$ is the result of the following transformations on d. For the purposes of evaluating the colouring function c, we introduce two temporary values e and f.
  1. For every node $v \in N$ with label S', replace S' by $\delta(S', \sigma)$.
  2. For every node $v \in N$ with label S' such that $S' \cap \alpha \neq \emptyset$, create a new son v' of v and let v' be the minimal node not used so far. Let $l(v) = S' \cap \alpha$. We may temporarily have to use nodes in $[n+1 \ldots 2n]$
  3. For every node v with label S' and state $s \in S$, if there exists a $v' \in N$, such that $p(v') = p(v)$ and $v' < v$ and $s \in l(v')$, remove s from the label of v and all its descendants.
  4. For every node v whose label is equal to the union of the labels of its children, remove all descendants of v and call such nodes *green*. Set f to the minimum of $n + 1$ and the names of green nodes. No node in $[(n + 1), ...(2n)]$ can be green.
  5. Remove all nodes with empty labels. Set e to the minimum of $n + 1$ and the names of nodes removed during all stages of the transformation. Green nodes that are removed cannot make a state of even priority.
  6. If $f < e$, set $c = f$; if $e < f$, set $c = -e$. Else, c = 0. Note that $c = -1$ is not considered as this is the case where the label of the root is empty. This is a dead end and is thus a rejecting sink state.
- The parity acceptance condition is a partition $\alpha' = (F_0, ..., F_{2n-1})$ defined as follows:
  - $F_0 = \{d \in D | c = 1\}$
  - $F_{2i+1} = \{d \in D | c = -i - 2\}$ for all $0 \leq i < n - 1$.
  - $F_{2i+2} = \{d \in D | c = i + 2\}$ for all $0 \leq i < n - 1$.
  - $F_{2n-1} = \{d \in D | c = 0\}$.

Theorem 2.1 $\mathcal{L}(\mathcal{D}) = \mathcal{L}(\mathcal{N})$ *[1]* □

3 From NBW to DPW - a tight analysis

In this section, we present a tight analysis of Piterman's construction. This analysis was originally provided by Schewe in [4].

Lemma 3.1 *The number of compact Safra trees is at most* $\frac{4n}{n-1}(n!)^2$.

Proof. Let us represent a compact Safra tree with m nodes by a sequence of at most $(m - 1)$ pointers, such that the ith pointer points to a position in 1,...,i-1. This represents the parenthood relation. There are at most $(m-1)!$ sequences of this sort. This can be proved by induction over m. When, m = 1, there is obviously only one sequence. We assume that this claim is true for some m = $(k - 1)$. To produce a sequence with k nodes, we construct a sequence with $(k - 1)$ nodes. We then select a parent for the kth node. There are $(k - 1)$ possible choices for the parent. By the induction hypothesis, the number of sequences is $(k - 1) \times (k - 2)!$ = $(k - 1)!$.

To analyse the resulting number of *compact Safra trees* , we consider two cases. First, we look at trees where the root is labelled with the complete set of states S of the NBW. The number of compact

Safra trees is the product of the number of ordered trees and the number of labelling functions. The labelling function $l : S \to [m]$ is a function from S onto $\{1, ..., m\}$ that associates a state with the minimal node to which it belongs. Let $t(n, m)$ denote the number of labelled trees with m nodes and $n = |S|$ states in the label of the root. We see that $t(n, n) = (n-1)!n!$ as there are $(n-1)!$ ordered trees/sequences and $n!$ ways to label them. We consider $t(n, m-1)$ which is the number of trees labelled by states as in compact Safra trees, where the root is labelled by all states. We find that for every $m \in [2, \ldots, n]$, we have $t(n, m-1) \leq \frac{1}{2}\, t(n, m)$ as follows.

Let $g : S \to [m-1]$ be a function from S onto $\{1, ..., m-1\}$ and $f : S \to [m]$ be a function from S onto $\{1, ..., m\}$. Let $X \subseteq [1, \ldots, m-1]$. We say that f "extends" g if $f(x) = g(x)$ for all $x \in X$. So, for every f, there are at most $m-1$ extensions and inversely, for every g, there are at least 2 subfunctions. If we visualise this connection as a directed graph, then, there are $(m-1)$ edges from g to f. Inversely, we see that there are at least two edges from f to g.

Let $\alpha(a, b)$ be the number of functions from a set with $[a]$ elements onto another set with $[b]$ elements.

Lemma 3.2 $\alpha(a, b-1) \leq \frac{b-1}{2}\alpha(a, b)$

We can relate $\alpha(a, b-1)$ and $\alpha(a, b)$ by means of a directed graph. A coarse estimate from [4] says that

$$\alpha(a, b-1) \leq \frac{b-1}{2}\alpha(a, b)$$

as there are $b-1$ edges from $\alpha(a, b-1)$ to $\alpha(a, b)$ and inversely at least two edges from $\alpha(a, b)$.

Hence, the number of labelling functions increases by a factor of $\frac{m-1}{2}$, while the number of trees decreases by a factor of $m-1$. Overall, the number of trees decreases by a factor of $\frac{1}{2}$.

So, the number of such trees is $\sum_{i=1}^{n} t(n, i) \leq 2t(n, n)$. Thus, $\sum_{i=1}^{n} t(n, i) \leq 2(n\text{-}1)!n!$

Next, we consider the case where the root is not labelled with the complete set of states S of the NBW. We proceed in a similar direction as in the above case. The number of compact Safra trees is the product of the number of ordered trees and the number of labelling functions. The labelling function $l : S \to [m]$ is a function from S onto $\{1, ..., m\} \cup \{\perp\}$. Let $t'(n, m)$ denote the number of labelled trees such that the root is not labelled with S and $|N|$=m. We see that $t'(n, n-1) = (n-2)!n!$ as there are $(n-2)!$ ordered trees/sequences and $n!$ ways to label them. We then consider $t'(n, m-1)$ which is the number of trees labelled by states as in compact Safra trees, where the root is not labelled by all states. We find that $t'(n, m-1) \leq \frac{1}{2}\, t'(n, m)$ as follows.

We use **Lemma 3.2**, in a similar fashion to the above case. We see that $t'(n, n-1) = (n-2)!n!$ and $t'(n, m-1) \leq \frac{1}{2}\, t'(n, m)$.

So, The number of such trees is $\sum_{i=1}^{n-1} t'(n, i) \leq 2t'(n, n-1)$. Thus, $\sum_{i=1}^{n-1} t'(n, i) \leq 2(n\text{-}2)!n!$.

Having managed to count the number of trees with all states labelled in the root and the number of trees without all states, labelled in the root, we proceed to count the overall number of such trees. Overall, the number of ordered trees is $\sum_{i=1}^{n} t(n, i) + \sum_{i=1}^{n-1} t'(n, i) \leq 2(n\text{-}1)!n! + 2(n\text{-}2)!n! \leq (2(n-1)+2)(n-2)!n! \leq 2n(n-2)!n!$. There are $2n$ possible values to define the acceptance condition. So the number of states of the resulting DPW is at most $2n(n-2)!n!$ times $2n$ giving at most $\frac{4n}{n-1}(n!)^2$ states. □

We will use the technique used in this analysis to analyse the resulting number of states for the Streett determinisation construction.

4 From NSW to DPW - Piterman's construction

This determinisation construction is an extension of Safra's construction [3]. As in Safra's determinisation construction, the principal structure of the states of the resulting deterministic automaton is a tree called a *compact Streett Safra tree* [1]. The difference from Safra's construction is that we replace the static node names with dynamic names that decrease as nodes below get erased from the tree, essentially getting rid of the 'older-than' function. Thus, older nodes have smaller names. A natural parity condition is enforced. Using dynamic node names allows us to simulate the *index appearance record* (which allows for translation of Rabin and Streett conditions to parity conditions) within the deterministic automaton. Piterman's construction takes a Streett automaton with n states and k accepting pairs and produces a deterministic parity automaton with $2n^n m!(k+1)^m$ states and $2m$ parity indices(where $m = n(k+1)$[1].

4.1 Construction

Let $\mathcal{S}$ = $(S, \Sigma, s_0, \delta, \alpha)$ be an NSW with $|S| = n$ states and α = $\{(R_1, G_1), ..., (R_k, G_k)\}$. We construct an equivalent DPW $\mathcal{D}$ = $(D, \Sigma, d_0, \rho, \alpha')$. We first define the structure of the states of the resulting DPW. Let $m = n(k+1)$.

A compact Streett Safra tree over S is $(N, 1, p, l, h, e, f)$ defined as follows:

- $N \subseteq [m]$ such that if $i \in N$, then $i - 1 \in N$.
- $1 \in N$ is the root.
- $p : (N \setminus \{1\}) \to N$ is the parent function, such that for every $i > 1, p(i) < i$.
- $l : N \to 2^S$ is the labelling function such that for every $i, l(i) \subset l(p(i))$, and, $\exists s$ such that $s \in l(i)$ and $s \notin l(j)$, for $j \neq i$.
- $h : N \to 2^{[k]}$ is the annotation function that annotates every node with a set of indices from $[k]$. The root is annotated by $[k]$. The annotation of every node is a subset of the annotation of its parent and it misses at most one element from the parent node's annotation. Every non-leaf node has at least one son with a strictly smaller annotation.
- $e, f \in [m+1]$ are used to define the parity acceptance condition.

The components of the DPW $\mathcal{D}$ = $(D, \Sigma, d_0, \rho, \alpha')$ are as follows:

- D is the set of compact Streett Safra trees over $\mathcal{S}$.
- d_0 is the tree with the single node 1 labelled by $\{s_0\}$ and annotated $[k]$. We set $e = 2$ and $f = 1$.
- For every tree $d \in D$ and $\sigma \in \Sigma$, the transition $d' = \rho(d, \sigma)$ is the result of the following recursive transformations on d starting from the root. Before we start, we set e and f to $m + 1$ and replace the label of every node by $\delta(l(v), \sigma)$.
  1. If v is a leaf such that $h(v) = \emptyset$, stop.
  2. If v is a leaf such that $h(v) \neq \emptyset$, add to v a new son $v' \in V'$. Set $l(v') = l(v)$ and $h(v') = h(v) - \{max(h(v))\}$.
  3. Let $v_1..., v_l$ be the sons of v(ordered from oldest to youngest) and let $j_1...j_l$ be the indices such that $j_i \in h(v) - h(v_i)$. Call the procedure recursively on $v_1...v_l$ including sons created in the previous step.
     (a) If $s \in R_j$, remove s from the label of v_i and all its descendants. Add a new son $v' \in [m]$ to v. Set $l(v') = \{s\}$ and $h(v') = h(v) - \{max(\{0\} \cup (h(v) \cap \{1, ..., j_i - 1\}))\}$.
     (b) If $s \in G_j$, remove s from the label of v_i and all its descendants. Add a new son $v' \in [m]$ to v. Set $l(v') = \{s\}$ and $h(v') = h(v) - \{j_i\}$.
  4. If a state appears in $l(v_i)$ and $l(v_{i'})$ and $j_i < j_{i'}$, remove s from the label of $v_{i'}$ and all its descendants.
  5. If a state appears in $l(v_i)$ and $l(v_{i'})$ and $j_i = j_{i'}$, remove s from the label of the younger sibling and all its descendants.

6. Remove sons with empty label. Set e to the minimum of its previous value and the minimal removed descendant.
7. If all sons are annotated by $h(v)$, remove all the sons and their descendants. Set e to the minimum of its previous value and the minimal removed descendant. Set f to the minimum of its previous value and v.

- The parity acceptance condition is a partition $\alpha' = (F_0, ..., F_{2m-1})$ defined as follows:
  - $F_0 = \{d \in D | f = 1 \text{ and } e > 1\}$
  - $F_{2i+1} = \{d \in D | e = i + 2 \text{ and } f \geq e\}$.
  - $F_{2i+2} = \{d \in D | f = i + 2 \text{ and } e > f\}$.

Theorem 4.1 *$\mathcal{L}(\mathcal{D}) = \mathcal{L}(\mathcal{S})$ [1]* □

5 An improved state analysis of Piterman's Streett determinisation construction

Lemma 5.1 *For a compact Streett Safra tree with 'i' states, at most $nk + i$ node names are sufficient [1]*

There exists a path from the root to a leaf where no edge is annotated by 0. Now, for every edge annotated with 0, there is a path from the target node of this edge to a leaf where there is no edge annotated by 0. It follows that there are at most $i - 1$ edges annotated by 0. All other edges are annotated by some $j \in [k]$. There are at most nk such edges. Thus, $nk + i$ node names are sufficient.

Lemma 5.2 *The number of compact Streett Safra trees is at most $8(\frac{m!}{n})(n + 1)!(k + 1)^m$*

Proof. We define the notation $maxnodes_k(i)$ to be the maximum number of nodes in such a compact Streett Safra tree.

Let us represent a compact Safra tree with m nodes by a sequence of at most $(m - 1)$ pointers, such that the ith pointer points to a position in 1,...,i-1. This represents the parenthood relation. There are at most $(m - 1)!$ sequences(trees) of this sort. This can be proved by induction over m. When, m = 1, there is obviously only one sequence. We assume that this claim is true for some m = $(k - 1)$. To produce a sequence with k nodes, we construct a sequence with $(k - 1)$ nodes. We then select a parent for the kth node. There are $(k - 1)$ possible choices for the parent. By the induction hypothesis, the number of sequences is $(k - 1) \times (k - 2)!$ = $(k - 1)!$.

here are also at most $n!$ labels. The edge annotation can be represented by a function $h : [m] \rightarrow [0, ..., k]$. There are $(k + 1)^m$ such annotations.

Having estimated the number of the above functions, let us now define $t_k(i, j)$ as the number of compact Streett Safra trees with i nodes, j leaves and k Streett acceptance pairs(indices) for the case where the label of the root contains all states of $\mathcal{S}$. Using the above estimates of the number of functions, it follows that $t_k(m, n)$ = $(m - 1)!n!(k + 1)^m$.

In order to count the number of resulting compact Streett Safra trees, we proceed as follows. Firstly, we consider $\sum_{i=1}^{m} t_k(i, j)$ where the ordered tree has i nodes and exactly $j = n$ leaves. We estimate that $t_k(i - 1, j) \leq \frac{1}{2} t_k(i, j)$ as follows.

Recall that if $\alpha(a, b)$ is the number of functions from a set with $[a]$ elements onto another set with $[b]$ elements, then by Lemma 3.2, $\alpha(a, b - 1) \leq \frac{b-1}{2}$. We can thus relate $t_k(i - 1, j)$ and $t_k(i, j)$ as below.

	$t_k(i-1, j)$	$t_k(i, j)$	Growth factor
No. of trees	$(i-2)!$	$(i-1)!$	$\frac{1}{i-1}$
No. of annotations	$\alpha(k+1, i-1)$	$\alpha(k+1, i)$	$(\frac{i}{i-1})^k$

The number of ordered trees in $t_k(i-1, j)$ decreases by a factor of $i-1$ than $t_k(i, j)$ and the number of annotations increases by a factor of $(\frac{i}{i-1)^k}$. Clearly, we are not interested in trees with 1 or 2 nodes. Overall, we see that $t_k(i-1, j) \leq \frac{1}{i-1}(\frac{i}{i-1})^k t_k(i, j)$. We can roughly infer that $t_k(i-1, j) \leq \frac{1}{2} t_k(i, j)$. Hence $\sum_{i=1}^{m} t_k(i, n) \leq 2(m-1)!n!(k+1)^m$.

Now, we proceed by counting the number of trees with fewer than n leaves. When we go in this direction, for $j < n$, we are clearly not interested in trees with more than $maxnodes_k(i)$ nodes. Hence the most significant figure for $j < n$ is $t_k(maxnodes_k(j), j)$. We estimate that $t_k(maxnodes_k(i-1), i-1) \leq \frac{1}{2} t_k(maxnodes_k(i), i)$ as follows.

	$t_k(maxnodes_k(i-1), i-1)$	$t_k(maxnodes_k(i), i)$	Growth factor
No. of trees	$(nk+i-1)!$	$(nk+i)!$	$\frac{1}{nk+i}$
No. of labeling fns.	$\alpha(n, i-1)$	$\alpha(n, i)$	$\frac{i-1}{2}$
No. of annotations	$(k+1)^{nk+i-1}$	$(k+1)^{nk+1}$	$\frac{1}{k+1}$

From Lemma 5.1, we see that for a compact Streett Safra tree with $'i'$ states, there are at most $nk + i$ nodes. Thus there can be at most $(nk + i - 1)!$ such trees. The number of such trees in $t_k(maxnodes_k(i-1), i-1)$ is lesser than $t_k(maxnodes_k(i), i)$ by a factor of $nk + 1$. The number of labelling functions increases by a factor of $\frac{i-1}{2}$ and the number of annotations decreases by a factor of $k + 1$. Overall we see that $t_k(maxnodes_k(i-1), i-1) \leq \frac{i-1}{2} \frac{1}{nk+i} \frac{1}{k+1} t_k(maxnodes_k(i), i)$ which is roughly $\leq \frac{1}{2} t_k(maxnodes_k(i), i)$. Hence $\sum_{i=1}^{m} t_k(maxnodes_k(i), i) \leq 2(m-1)!n!(k+1)^m$.

Now we consider the case where not all states are labelled in the root. Let us now define $t'_k(i, j)$ as the number of compact Streett Safra trees with i nodes, j leaves and k Streett acceptance pairs(indices) for the case where the label of the root does not contain all states of $\mathcal{S}$. Using the above estimates of the number of functions, it follows that $t'_k(m, n-1) = (m-1)!(n-1)!(k+1)^m$.

In order to count the number of resulting compact Streett Safra trees, we proceed as follows. Firstly, we consider $\sum_{i=1}^{m} t'_k(i, j)$ where the ordered tree has i nodes and exactly $j = n-1$ leaves. We estimate that $t'_k(i-1, j) \leq \frac{1}{2} t'_k(i, j)$ as follows.

Recall that if $\alpha(a, b)$ is the number of functions from a set with $[a]$ elements onto another set with $[b]$ elements, then by Lemma 3.2, $\alpha(a, b-1) \leq \frac{b-1}{2}$. We can thus relate $t'_k(i-1, j)$ and $t'_k(i, j)$ as below.

	$t'_k(i-1, j)$	$t'_k(i, j)$	Growth factor
No. of trees	$(i-2)!$	$(i-1)!$	$\frac{1}{i-1}$
No. of annotations	$\alpha(k+1, i-1)$	$\alpha(k+1, i)$	$(\frac{i}{i-1})^k$

The number of ordered trees in $t'_k(i-1, j)$ decreases by a factor of $i-1$ than $t'_k(i, j)$ and the number of annotations increases by a factor of $(\frac{i}{i-1})^k$. Clearly, we are not interested in trees with 1 or 2 nodes. Overall, we see that $t'_k(i-1, j) \leq \frac{1}{i-1}(\frac{i}{i-1})^k t'_k(i, j)$. We can roughly infer that $t'_k(i-1, j) \leq \frac{1}{2} t'_k(i, j)$. Hence $\sum_{i=1}^{m} t'_k(i, n-1) \leq 2(m-1)!(n-1)!(k+1)^m$.

Now, we proceed by counting the number of trees with fewer than $n-1$ leaves. When we go in this direction, for $j < n$, we are clearly not interested in trees with more than $maxnodes_k(i)$ nodes. Hence the most significant figure for $j < n-1$ is $t'_k(maxnodes_k(j), j)$. We estimate that $t'_k(maxnodes_k(i-1), i-1) \leq \frac{1}{2} t'_k(maxnodes_k(i), i)$ as follows.

	$t'_k(maxnodes_k(i-1), i-1)$	$t'_k(maxnodes_k(i), i)$	Growth factor
No. of trees	$(nk+i-1)!$	$(nk+i)!$	$\frac{1}{nk+i}$
No. of labeling fns.	$\alpha(n, i-1)$	$\alpha(n, i)$	$\frac{i-1}{2}$
No. of annotations	$(k+1)^{nk+i-1}$	$(k+1)^{nk+1}$	$\frac{1}{k+1}$

From Lemma 5.1, we see that for a compact Streett Safra tree with $'i'$ states, there are at most $nk+i$ nodes. Thus there can be at most $(nk+i-1)!$ such trees. The number of such trees in $t_k(maxnodes_k(i-1), i-1)$ is lesser than $t_k(maxnodes_k(i), i)$ by a factor of $nk+1$. The number of labelling functions increases by a factor of $\frac{i-1}{2}$ and the number of annotations decreases by a factor of $k+1$. Hence $\sum_{i=1}^{m} t'_k(maxnodes_k(i), i) \leq 2(m-1)!(n-1)!(k+1)^m$.

Now, we count the overall number of compact Streett Safra trees. This will be $\sum_{i=1}^{m} t_k(i,n) + \sum_{i=1}^{m} t_k(maxnodes_k(i), i) + \sum_{i=1}^{m} t'_k(i, n-1) + \sum_{i=1}^{m} t'_k(maxnodes_k(i), i)$. Let us denote this quantity with the symbol $\mathcal{S}$. Now,

$$\begin{aligned}\mathcal{S} &\leq 2(m-1)!n!(k+1)^m + 2(m-1)!n!(k+1)^m + 2(m-1)!(n-1)!(k+1)^m \\ &\quad +2(m-1)!(n-1)!(k+1)^m \\ &\leq 4(m-1)!n!(k+1)^m + 4(m-1)!(n-1)!(k+1)^m \\ &\leq 4(m-1)!(n-1)!(k+1)^m(n+1) \qquad (1)\end{aligned}$$

We also need $2m$ possible values to define the acceptance condition, so the overall number of compact Safra trees is $2m \times 4(m-1)!(n-1)!(k+1)^m(n+1)$ which is $8(\frac{m!}{n})(n+1)!(k+1)^m$. □

References

[1] N. Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. *Journal of LMCS*, 3(3:5), 2007.

[2] S. Safra. On the complexity of ω-automata. *Proc. of FOCS 1988*, pages 319–327

[3] S. Safra. Exponential determinization for omega-automata with strong-fairness acceptance condition. *Proc. of STOC 1992*, pages 275–282.

[4] S. Schewe. Tighter bounds for the determinisation of Büchi automata. *Proc. of FoSSaCS 2009*, pages 167–181.

[5] S. Schewe and T. Varghese. Determinising parity automata. *Proc. of MFCS 2014*, Volume 1, pages 486–498.