ACTS 2011

Determinization of ω-automata unified

Hrishikesh Karmarkar and Supratik Chakraborty

IIT Bombay

Plan

- Background
- Determinization of Streett automata
- Determinization of ω-automata
- Strategy for improved upper-bound
- Summary

ω-automata

- An ω-automaton is a finite state automaton that *accepts* infinite words or ω-words and extends classical NFA/DFA.

- $\mathcal{A} = (\Sigma, Q, Q_0, \delta, \phi)$ is an ω-automaton, where ...
  - Σ is a finite alphabet.
  - Q is a finite set of states.
  - $Q_0 \subseteq Q$ is the set of initial states.
  - $\delta : Q \times \Sigma \rightarrow 2^Q$ is a transition function.
  - ϕ is the acceptance condition.

- The acceptance condition determines the set of infinite words that are accepted by $\mathcal{A}$ i.e the *language* of $\mathcal{A}$.

Acceptance conditions

- We deal with acceptance conditions based on the notion of infinity sets i.e. the set of states visited infinitely often in a run of the automaton.
- Acceptance condition ϕ can be thought of as defining a predicate P_ϕ over 2^Q.

Definition

For every $X \subseteq Q$, we say $P_\phi(X) = \text{True}$ iff X, viewed as the infinity set of a run of $\mathcal{A}$, satisfies the properties specified by ϕ.

Acceptance conditions

Examples of acceptance conditions ...

- Streett condition:
  1. ϕ is given by a table of pairs. $\mathcal{T} = \{(E_1, F_1), (E_2, F_2), \ldots (E_h, F_h)\}$, where $E_i, F_i \subseteq Q$ for all $i \in \{1, \ldots, h\}$.
  2. $P_\phi(X) =$ True iff for all $i \in \{1 \ldots h\}$, $X \cap F_i \neq \emptyset \Rightarrow X \cap E_i \neq \emptyset$
- Parity condition:
  1. ϕ is given by a sequence of sets $\mathcal{F} = \langle F_0, F_1, \ldots F_h \rangle$, where $F_i \subseteq Q$ for all $i \in \{0, \ldots h\}$.
  2. $P_\phi(X) =$ True iff for some even number $j \in \{0, \ldots h\}$, $X \cap F_j \neq \emptyset$ and for all $m \in \{0, \ldots j - 1\}$, $X \cap F_m = \emptyset$.

Determinization of ω-automata

- Many constructions exist to construct deterministic ω-automata from nondeterministic ω-automata.
- All known constructions are tailor-made to work for non-deterministic automata, each with a specific kind of acceptance condition.
- It is very hard to adapt the construction for one type of automaton to another.

Determinization of ω-automata

Construction	**NBW**	**NSW**	**NMW**	**Output**
Safra+Piterman	$\checkmark$	$\checkmark$	$\times$	DPW
Muller-Schupp	$\checkmark$	$\times$	$\times$	DRW
Kähler and Wilke + Piterman	$\checkmark$	$\times$	$\times$	DPW

N: non-deterministic | D: deterministic
B,S,M,R,P: Büchi, Streett, Muller, Rabin, parity
W : over infinite words.

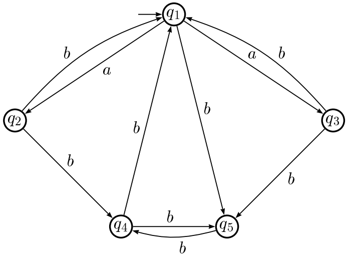
Construction	NBW	NSW	NMW	Output
Safra+Piterman	$\checkmark$	$\checkmark$	$\times$	DPW
Muller-Schupp	$\checkmark$	$\times$	$\times$	DRW
Kähler and Wilke + Piterman	$\checkmark$	$\times$	$\times$	DPW

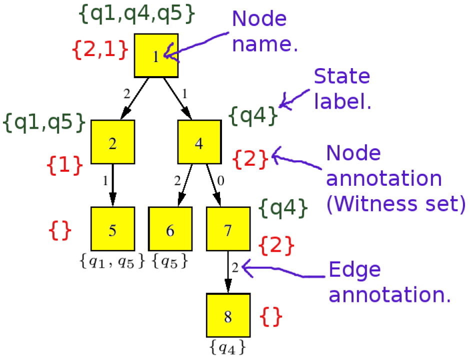
N: non-deterministic | D: deterministic
B,S,M,R,P: Büchi, Streett, Muller, Rabin, parity
W : over infinite words.

Safra's NSW determinization construction

- Uses a tree data structures similar to Safra's determinization construction for NBW.
- States are trees, called Safra-Streett trees[Schwoon2001] with additional book-keeping information.
- Piterman describes a way to reuse Safra's method to construct DPW.

Safra-Streett trees

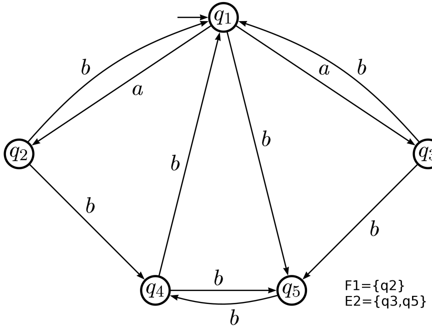




For every run examined by a node check if for every j in witness-set J . . .

1. Some state in E_j is visited infinitely often.
2. No state in $F_{j'}$ is visited infinitely often for $j' \notin J$

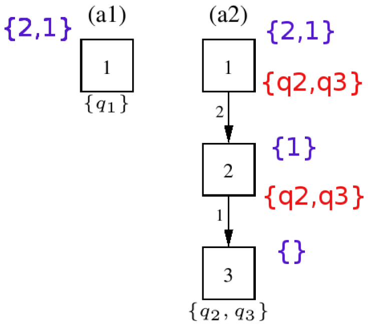
The NSW determinization construction: An example

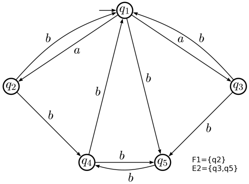


Example from - Stefan Schwoon: Determinization and Complementation of Streett Automata. Automata, Logics, and Infinite Games 2001: 79-91

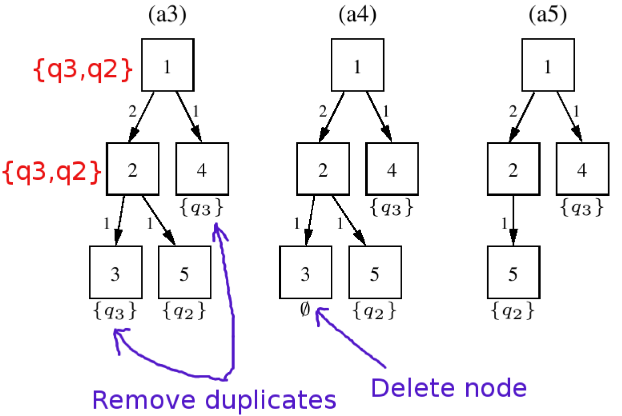
$\Sigma = \{a, b\}$. Word $\alpha = abb\ldots$.

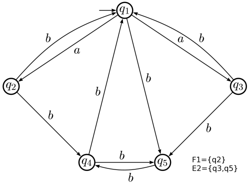
The NSW determinization construction: An example



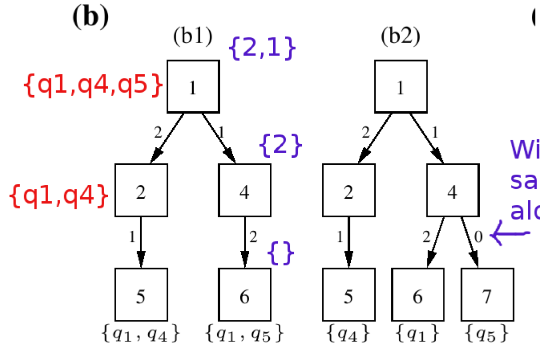


The NSW determinization construction: An example

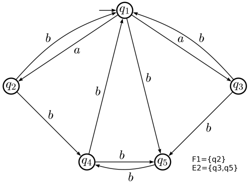




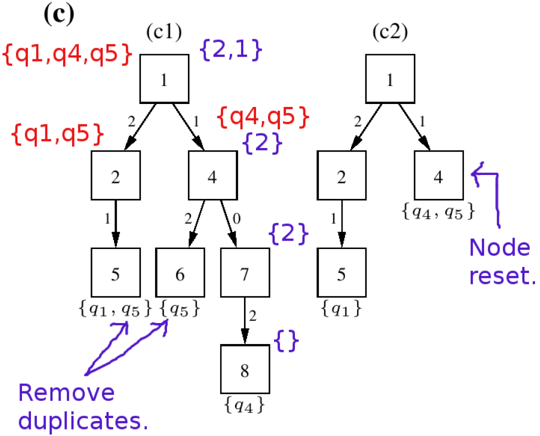
The NSW determinization construction: An example

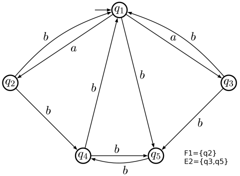


Witness set satisfied along run.



The NSW determinization construction: An example





Theorem

There is an accepting run of the NSW $\mathcal{A}$ on a word α iff there is an infinite run of Streett-Safra trees in the DPW $\mathcal{D}$ along which some node is reset infinitely often and removed finitely often.

Theorem

The number of states of the DPW $\mathcal{D}$ is $2^{O(nh\log(nh))}$, while the number of parity indices in $O(nh)$, where n is the number of states and h is the number of pairs of the NSW $\mathcal{A}$.

Observations on ω-determinization

- All asymptotically optimal determinization constructions for ω-automata use state trees (either Safra or Muller-Schupp).
- All commonly used acceptance conditions are described using the notion of infinity sets.

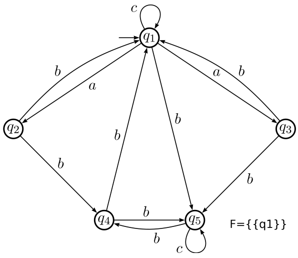
Generalized witness sets

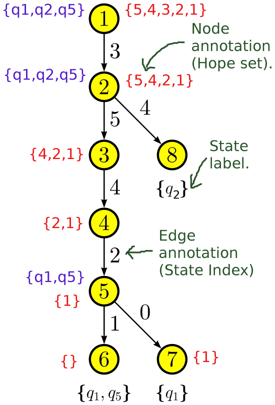
Definition (Generalized Witness Set (GWS))

A set $W \subseteq [n]$, $n = |Q|$ and $[n] = \{1, 2, \ldots, n\}$ is a generalized witness set for a run ρ of $\mathcal{A}$ iff $\mathit{inf}(\rho) = Q_W$ and $P_\phi(Q_W) =$ True, where $Q_W = \{q_i \mid q \in Q, i \in W\}$.

- Safra-Streett witness set : Subset of indices of Streett acceptance pairs
- Generalized witness set: Subset of indices of ω-states.
- A run of a Streett automaton has at most one GWS, but possibly many Safra-Streett witness sets.

Generalized Safra trees (GST)





For every run examined by a node it checks if q_i, for all i in hope-set W is visited.

ω-determinization: An example

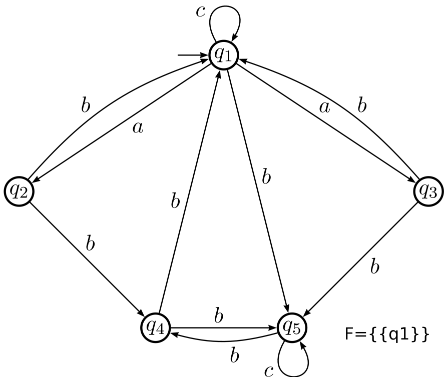
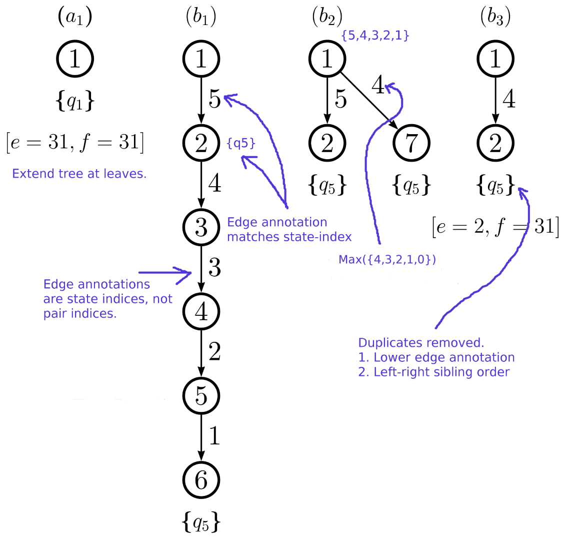
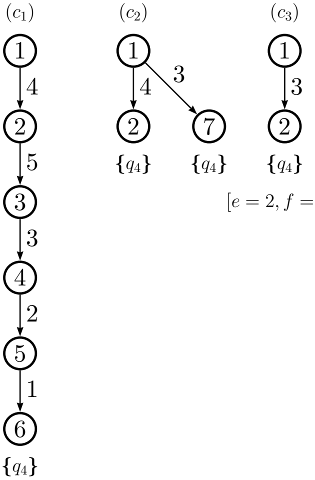


Figure: Muller automaton with $F = \{\{q_1\}\}$, Word: $bbbc^{\omega}$

ω-determinization: An example

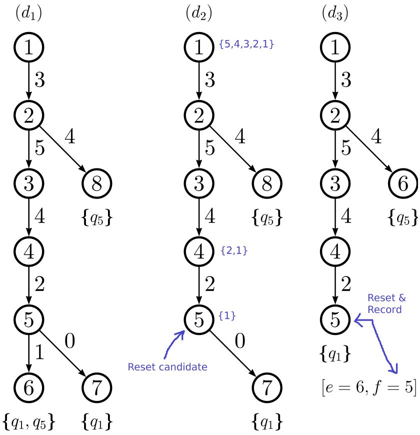


ω-determinization: An example



$[e = 2, f = 31]$

ω-determinization: An example



Theorem

Given an ω-automaton $\mathcal{A}$ with n states, the deterministic parity automaton $\mathcal{D}$ constructed above has at most $n^{O(n^2)}$ states and $2n^2$ parity indices.

- Construction gives a uniform view of the determinization process.
- New upper-bound on states and parity indices is independent of the number of pairs.
- Turns out to be more powerful than expected - There are classes of NSW for which new bound is better!

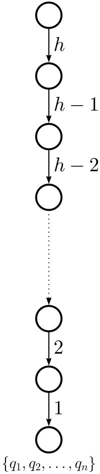
An improved bound for Streett determinization

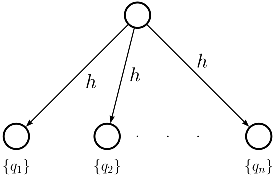
Theorem

There exists a family A_S *of NSW where each NSW* $\mathcal{A}_S \in \mathsf{A}_S$ *has* $3n+1$ *states and* $2^n + 1$ *accepting pairs for which the Safra-Streett+Piterman construction constructs a DPW with* $2^{\Omega(n^3)}$ *states, while our construction constructs a DPW with* $2^{O(n^2 \log n)}$ *states.*

Constructing family A_S: A Strategy

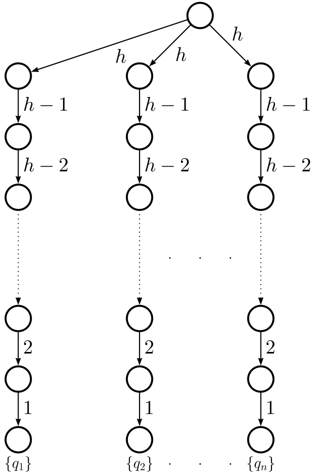
- From a Safra-Streett tree with a single path, obtain a tree with n disjoint paths.
- Ensure that $q_1, \ldots, q_n$ all belong to the F_h in the Streett pair (E_h, F_h).





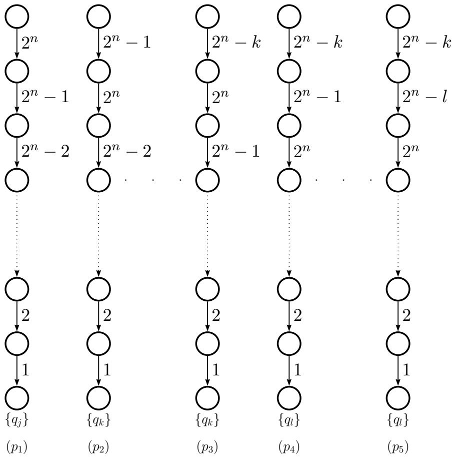
Constructing family A_S: A Strategy

- Tree with *n* disjoint paths gets extended as the computation of the next Safra-Streett tree proceeds.



Constructing family A_S: A Strategy

- Permute edge annotations for each disjoint path.
- Ensure right distribution of Streett states among the Streett pairs.
- E.g. To change 2^n to $2^n - 1$ via q_k, ensure $q_k \in E_h$ of pair (E_h, F_h).



Constructing family A_S: A strategy

We permute n pair indices with the following conditions.

- Pick $k = \lfloor \frac{2^n}{n} \rfloor$ blocks $B_1, B_2, \ldots, B_{k-2}, B_{k-1}, B_k$.
- From each block pick exactly one pair index. The i^{th} pair in block B_j is $(E_{2^n-(j-1)n+(i-1)}, F_{2^n-(j-1)n+(i-1)})$, and has index idx_j^i.
- If pair index idx_j^i is already picked from block j, then do not pick idx_l^i for $l \neq j$, for every pair of blocks B_j and B_l that are picked.

Constructing family A_S: A strategy

- We start out with $h = 2^n$ pairs in the NSW.
- Partition 2^n pairs into $\lfloor \frac{2^n}{n} \rfloor$ *blocks* of n pairs each.

Block 1

$$B_1 = \langle (E_{2^n}, F_{2^n}), (E_{2^n-1}, F_{2^n-1}), \ldots, (E_{2^n-(n-1)}, F_{2^n-(n-1)}) \rangle$$

Block 2

$$B_2 = \langle (E_{2^n-(n)}, F_{2^n-(n)}), (E_{2^n-(n+1)}, F_{2^n-(n+1)}), \ldots, (E_{2^n-(2n-1)}, F_{2^n-(2n-1)}) \rangle$$

Let $\lfloor \frac{2^n}{n} \rfloor = k$, then last or k^{th} block

$$B_k = (E_{2^n-((k-1)n)}, F_{2^n-((k-1)n)}), (E_{2^n-((k-1)n+1)}, F_{2^n-((k-1)n+1)}), \ldots,$$
$$(E_{2^n-(kn-1)}, F_{2^n-(kn-1)})$$

Constructing family A_S: A strategy

- To construct a permutation of length n. E.g. $\langle 3, 2, \ldots, n \rangle$

$$B_1 = \langle (E_{2^n}, F_{2^n}), (E_{2^n-1}, F_{2^n-1}), (E_{2^n-2}, F_{2^n-2}), (E_{2^n-3}, F_{2^n-3}) \ldots, \\ (E_{2^n-(n-1)}, F_{2^n-(n-1)}) \rangle$$

$$B_2 = \langle (E_{2^n-(n)}, F_{2^n-(n)}), (E_{2^n-(n+1)}, F_{2^n-(n+1)}), \ldots, (E_{2^n-(2n-1)}, F_{2^n-(2n-1)}) \rangle$$

$$\vdots$$

$$B_k = (E_{2^n-((k-1)n)}, F_{2^n-((k-1)n)}), (E_{2^n-((k-1)n+1)}, F_{2^n-((k-1)n+1)}), \ldots, \\ (E_{2^n-(kn-1)}, F_{2^n-(kn-1)})$$

Constructing family A_S: A strategy

- To construct a permutation of length n. E.g. $\langle 3, 2, \ldots, n \rangle$

$$B_1 = \langle (E_{2^n}, F_{2^n}), (E_{2^n-1}, F_{2^n-1}), (E_{2^n-2}, F_{2^n-2}), (E_{2^n-3}, F_{2^n-3}) \ldots, \\ (E_{2^n-(n-1)}, F_{2^n-(n-1)}) \rangle$$

$$B_2 = \langle (E_{2^n-(n)}, F_{2^n-(n)}), (E_{2^n-(n+1)}, F_{2^n-(n+1)}), \ldots, (E_{2^n-(2n-1)}, F_{2^n-(2n-1)}) \rangle$$

$$\vdots$$

$$B_k = (E_{2^n-((k-1)n)}, F_{2^n-((k-1)n)}), (E_{2^n-((k-1)n+1)}, F_{2^n-((k-1)n+1)}), \ldots, \\ (E_{2^n-(kn-1)}, F_{2^n-(kn-1)})$$

- To construct a permutation of length n. E.g. $\langle 3, 2, \ldots, n \rangle$

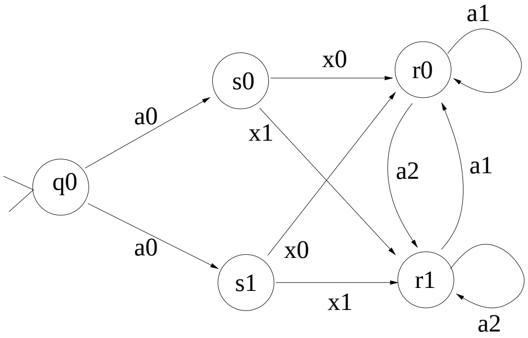
$$B_1 = \langle (E_{2^n}, F_{2^n}), (E_{2^n-1}, F_{2^n-1}), (E_{2^n-2}, F_{2^n-2}), (E_{2^n-3}, F_{2^n-3}) \ldots, \\ (E_{2^n-(n-1)}, F_{2^n-(n-1)}) \rangle$$

$$B_2 = \langle (E_{2^n-(n)}, F_{2^n-(n)}), (E_{2^n-(n+1)}, F_{2^n-(n+1)}), \ldots, (E_{2^n-(2n-1)}, F_{2^n-(2n-1)}) \rangle$$

$\vdots$

$$B_k = (E_{2^n-((k-1)n)}, F_{2^n-((k-1)n)}), (E_{2^n-((k-1)n+1)}, F_{2^n-((k-1)n+1)}), \ldots, \\ (E_{2^n-(kn-1)}, F_{2^n-(kn-1)})$$

Constructing family A_S: An example



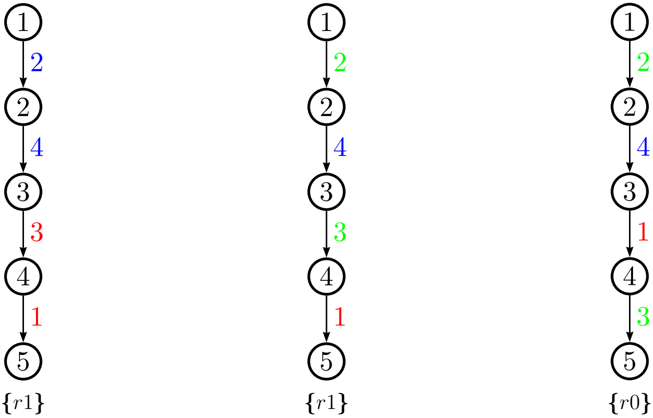
$(E_4, F_4) = (\{r0\}, \{s0, s1\})$ $(E_3, F_3) = (\{r1\}, \{\})$ $(E_2, F_2) = (\{r0\}, \{\})$
$(E_1, F_1) = (\{r1\}, \{\})$

Example

$n = 2$, $2^n = 4$. Permutation of length 2 $\ldots \langle 4, 1\rangle$
$(E_4, F_4) = (\{r0\}, \{s0, s1\})$ $(E_3, F_3) = (\{r1\}, \{\})$ $(E_2, F_2) = (\{r0\}, \{\})$
$(E_1, F_1) = (\{r1\}, \{\})$

Example

$n = 2$, $2^n = 4$. Permutation of length $2 \ldots \langle 2, 3 \rangle$
$(E_4, F_4) = (\{r0\}, \{s0, s1\})$ $(E_3, F_3) = (\{r1\}, \{\})$ $(E_2, F_2) = (\{r0\}, \{\})$
$(E_1, F_1) = (\{r1\}, \{\})$



$s0 \xrightarrow{x0} r0 \xrightarrow{a2} r1$ $\quad$ $r1 \xrightarrow{a2} r1 \xrightarrow{a1} r0$

Counting for the lower bound

- Permutation of length n.
- $k = \lfloor \frac{2^n}{n} \rfloor$ blocks of Streett pairs, with n elements in each block.
- Each element $2^n - ni - j$, for all $i, j \in \{0, \ldots, n-1\}$ can be chosen from one of k blocks. k^n choices.
- $n!$ ways of ordering the elements of blocks. Hence there are $n! \times k^n$ ways of choosing a block permutation in each disjoint branch.
- $(n! \times k^n)^n$ ways of choosing block permutations in all branches.
- Lower bound can be shown to be $2^{\Omega(n^3)}$.

Another lower bound for ω-automata

Rabin Index

Let $\mathcal{L}(k)$ be the set of all ω-regular languages that are accepted by DRW with k or less number of pairs. For any ω-regular language L the smallest k such that $L \in \mathcal{L}(k)$ is called the Rabin index of L.

Theorem

Given an ω-regular language L with Rabin index k, any ω-automaton (deterministic or non-deterministic) that uses an acceptance condition based on infinity sets and accepts L must have at least $\sqrt{k}$ states.

1. DPW can be interpreted as an equivalent DRW with at most n^2 Rabin acceptance pairs.
2. By definition of Rabin index we must have $n^2 \geq k$. It follows that $n \geq \sqrt{k}$.

Summary

- Unified determinization construction for ω-automata with acceptance conditions based on infinity sets.
- Direct determinization construction for non-deterministic Muller automata.
- Beats the best known upper bound for determinization of non-deterministic Streett automata for classes of NSW.
  - Upper bound is a function of only the number of states of the Streett automaton.
  - Earlier upper bound was a function of the number of states and number of acceptance pairs of the Streett automaton.

Summary

- A new lower bound on the number of states of any ω-automaton (deterministic or non-deterministic) accepting a given ω-regular language.
- Cons: Upper bound for Büchi determinization also $2^{O(n^2 \log(n))}$, while best known upper bound is $2^{O(n \log(n))}$.
- Future Work: Show that new bound is better than earlier bound for all $h > n$.

Thank you