The averaging trick and the Černý conjecture

Benjamin Steinberg

Carleton University

bsteinbg@math.carleton.ca

http://www.mathstat.carleton.ca/∼bsteinbg

DLT August 20, 2010

Background

- All automata in this talk are deterministic without initial or final states.
- An automaton (Q, A) is said to be synchronizing if there is a word $w \in A^*$ such that $|Qw| = 1$.
- Such a word w is called a reset word.
- The Černý conjecture is concerned with bounding the length of a reset word as a function of the number of states.

Conjecture (Černý)

An n-state synchronizing automaton admits a reset word of length at most $(n-1)^2$.

Background

- All automata in this talk are deterministic without initial or final states.
- An automaton (Q, A) is said to be synchronizing if there is a word $w \in A^*$ such that $|Qw| = 1$.
- Such a word w is called a reset word.
- The Černý conjecture is concerned with bounding the length of a reset word as a function of the number of states.

Conjecture (Černý)

An n-state synchronizing automaton admits a reset word of length at most $(n-1)^2$.

Background

- All automata in this talk are deterministic without initial or final states.
- An automaton (Q, A) is said to be synchronizing if there is a word $w \in A^*$ such that $|Qw| = 1$.
- Such a word w is called a reset word.
- The Černý conjecture is concerned with bounding the length of a reset word as a function of the number of states.

Conjecture (Černý)

An n-state synchronizing automaton admits a reset word of length at most $(n-1)^2$.

Background

- All automata in this talk are deterministic without initial or final states.
- An automaton (Q, A) is said to be synchronizing if there is a word $w \in A^*$ such that $|Qw| = 1$.
- Such a word w is called a reset word.
- The Černý conjecture is concerned with bounding the length of a reset word as a function of the number of states.

Conjecture (Černý)

An n-state synchronizing automaton admits a reset word of length at most $(n-1)^2$.

Background

- All automata in this talk are deterministic without initial or final states.
- An automaton (Q, A) is said to be synchronizing if there is a word $w \in A^*$ such that $|Qw| = 1$.
- Such a word w is called a reset word.
- The Černý conjecture is concerned with bounding the length of a reset word as a function of the number of states.

Conjecture (Černý)

An n-state synchronizing automaton admits a reset word of length at most $(n-1)^2$.

Černý's examples

- Černý showed that the shortest length reset word for the n-state synchronizing automaton with transitions

$$a = \begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ 2 & 3 & 4 & \cdots & 1 \end{pmatrix},\ b = \begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ 2 & 2 & 3 & \cdots & n \end{pmatrix}$$

is $(n-1)^2$.

- The Černý automaton for $n = 4$:

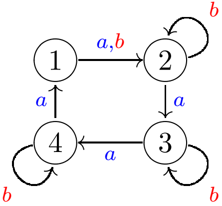
- The word $b(a^3b)^2$ resets to state 2.

Černý's examples

- Černý showed that the shortest length reset word for the n-state synchronizing automaton with transitions

$$a = \begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ 2 & 3 & 4 & \cdots & 1 \end{pmatrix}, \; b = \begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ 2 & 2 & 3 & \cdots & n \end{pmatrix}$$

  is $(n-1)^2$.
- The Černý automaton for $n = 4$:



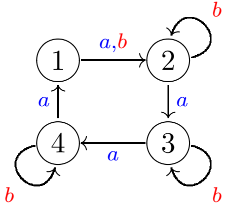
- The word $b(a^3b)^2$ resets to state 2.

Černý's examples

- Černý showed that the shortest length reset word for the n-state synchronizing automaton with transitions

$$a = \begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ 2 & 3 & 4 & \cdots & 1 \end{pmatrix}, \ b = \begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ 2 & 2 & 3 & \cdots & n \end{pmatrix}$$

  is $(n-1)^2$.
- The Černý automaton for $n = 4$:



- The word $b(a^3b)^2$ resets to state 2.

Pin's bound

- It is straightforward to obtain a cubic upper bound of $\frac{n^3-n}{3}$ on reset words for synchronizing automata.
- The best known upper bound is $\frac{n^3-n}{6}$, which was proved by Pin modulo an extremal set theory result of Frankl.
- Probabilistically speaking, all automata are synchronizing with reset word of length at most $2n$.
- The majority of the Černý literature consists of a vast array of special, but interesting, cases.
- I want to abstract here an argument that has been used in an increasing number of papers.

Pin's bound

- It is straightforward to obtain a cubic upper bound of $\frac{n^3-n}{3}$ on reset words for synchronizing automata.
- The best known upper bound is $\frac{n^3-n}{6}$, which was proved by Pin modulo an extremal set theory result of Frankl.
- Probabilistically speaking, all automata are synchronizing with reset word of length at most $2n$.
- The majority of the Černý literature consists of a vast array of special, but interesting, cases.
- I want to abstract here an argument that has been used in an increasing number of papers.

Pin's bound

- It is straightforward to obtain a cubic upper bound of $\frac{n^3-n}{3}$ on reset words for synchronizing automata.
- The best known upper bound is $\frac{n^3-n}{6}$, which was proved by Pin modulo an extremal set theory result of Frankl.
- Probabilistically speaking, all automata are synchronizing with reset word of length at most $2n$.
- The majority of the Černý literature consists of a vast array of special, but interesting, cases.
- I want to abstract here an argument that has been used in an increasing number of papers.

Pin's bound

- It is straightforward to obtain a cubic upper bound of $\frac{n^3-n}{3}$ on reset words for synchronizing automata.
- The best known upper bound is $\frac{n^3-n}{6}$, which was proved by Pin modulo an extremal set theory result of Frankl.
- Probabilistically speaking, all automata are synchronizing with reset word of length at most $2n$.
- The majority of the Černý literature consists of a vast array of special, but interesting, cases.
- I want to abstract here an argument that has been used in an increasing number of papers.

Pin's bound

- It is straightforward to obtain a cubic upper bound of $\frac{n^3-n}{3}$ on reset words for synchronizing automata.
- The best known upper bound is $\frac{n^3-n}{6}$, which was proved by Pin modulo an extremal set theory result of Frankl.
- Probabilistically speaking, all automata are synchronizing with reset word of length at most $2n$.
- The majority of the Černý literature consists of a vast array of special, but interesting, cases.
- I want to abstract here an argument that has been used in an increasing number of papers.

The basic strategy: expansion

- The basic strategy is to work backwards.
- Let (Q, A) be an n-state synchronizing automaton.
- Some letter, say a, must collapse two states to a state q.
- So $|qa^{-1}| \geq 2$.
- Suppose that, for each subset $S \subset Q$ with $1 < |S| < |Q|$, we can find a word $w \in A^*$ with $|Sw^{-1}| > |S|$.
- Then we can find a reset word of length at most $1 + (n-2)k$.
- Indeed, we can find a word w_1 with $|w_1| \leq k$ and $|qa^{-1}w_1^{-1}| > |qa^{-1}| \geq 2$.
- Continuing in this fashion, we can find words $w_2, \ldots, w_m$ with $m \leq n-2$ and $|w_i| \leq k$ such that $|qa^{-1}w_1^{-1} \cdots w_m^{-1}| = |Q|$.
- If $k = n$, then we obtain $1 + (n-2)n = (n-1)^2$.

The basic strategy: expansion

- The basic strategy is to work backwards.
- Let (Q, A) be an n-state synchronizing automaton.
- Some letter, say a, must collapse two states to a state q.
- So $|qa^{-1}| \geq 2$.
- Suppose that, for each subset $S \subset Q$ with $1 < |S| < |Q|$, we can find a word $w \in A^*$ with $|Sw^{-1}| > |S|$.
- Then we can find a reset word of length at most $1 + (n-2)k$.
- Indeed, we can find a word w_1 with $|w_1| \leq k$ and $|qa^{-1}w_1^{-1}| > |qa^{-1}| \geq 2$.
- Continuing in this fashion, we can find words $w_2, \ldots, w_m$ with $m \leq n-2$ and $|w_i| \leq k$ such that $|qa^{-1}w_1^{-1} \cdots w_m^{-1}| = |Q|$.
- If $k = n$, then we obtain $1 + (n-2)n = (n-1)^2$.

The basic strategy: expansion

- The basic strategy is to work backwards.
- Let (Q, A) be an n-state synchronizing automaton.
- Some letter, say a, must collapse two states to a state q.
- So $|qa^{-1}| \geq 2$.
- Suppose that, for each subset $S \subset Q$ with $1 < |S| < |Q|$, we can find a word $w \in A^*$ with $|Sw^{-1}| > |S|$.
- Then we can find a reset word of length at most $1 + (n-2)k$.
- Indeed, we can find a word w_1 with $|w_1| \leq k$ and $|qa^{-1}w_1^{-1}| > |qa^{-1}| \geq 2$.
- Continuing in this fashion, we can find words $w_2, \ldots, w_m$ with $m \leq n-2$ and $|w_i| \leq k$ such that $|qa^{-1}w_1^{-1} \cdots w_m^{-1}| = |Q|$.
- If $k = n$, then we obtain $1 + (n-2)n = (n-1)^2$.

The basic strategy: expansion

- The basic strategy is to work backwards.
- Let (Q, A) be an n-state synchronizing automaton.
- Some letter, say a, must collapse two states to a state q.
- So $|qa^{-1}| \geq 2$.
- Suppose that, for each subset $S \subset Q$ with $1 < |S| < |Q|$, we can find a word $w \in A^*$ with $|Sw^{-1}| > |S|$.
- Then we can find a reset word of length at most $1 + (n-2)k$.
- Indeed, we can find a word w_1 with $|w_1| \leq k$ and $|qa^{-1}w_1^{-1}| > |qa^{-1}| \geq 2$.
- Continuing in this fashion, we can find words $w_2, \ldots, w_m$ with $m \leq n-2$ and $|w_i| \leq k$ such that $|qa^{-1}w_1^{-1} \cdots w_m^{-1}| = |Q|$.
- If $k = n$, then we obtain $1 + (n-2)n = (n-1)^2$.

The basic strategy: expansion

- The basic strategy is to work backwards.
- Let (Q, A) be an n-state synchronizing automaton.
- Some letter, say a, must collapse two states to a state q.
- So $|qa^{-1}| \geq 2$.
- Suppose that, for each subset $S \subset Q$ with $1 < |S| < |Q|$, we can find a word $w \in A^*$ with $|Sw^{-1}| > |S|$.
- Then we can find a reset word of length at most $1 + (n-2)k$.
- Indeed, we can find a word w_1 with $|w_1| \leq k$ and $|qa^{-1}w_1^{-1}| > |qa^{-1}| \geq 2$.
- Continuing in this fashion, we can find words $w_2, \ldots, w_m$ with $m \leq n-2$ and $|w_i| \leq k$ such that $|qa^{-1}w_1^{-1} \cdots w_m^{-1}| = |Q|$.
- If $k = n$, then we obtain $1 + (n-2)n = (n-1)^2$.

The basic strategy: expansion

- The basic strategy is to work backwards.
- Let (Q, A) be an n-state synchronizing automaton.
- Some letter, say a, must collapse two states to a state q.
- So $|qa^{-1}| \geq 2$.
- Suppose that, for each subset $S \subset Q$ with $1 < |S| < |Q|$, we can find a word $w \in A^*$ with $|Sw^{-1}| > |S|$.
- Then we can find a reset word of length at most $1 + (n-2)k$.
- Indeed, we can find a word w_1 with $|w_1| \leq k$ and $|qa^{-1}w_1^{-1}| > |qa^{-1}| \geq 2$.
- Continuing in this fashion, we can find words $w_2, \ldots, w_m$ with $m \leq n-2$ and $|w_i| \leq k$ such that $|qa^{-1}w_1^{-1} \cdots w_m^{-1}| = |Q|$.
- If $k = n$, then we obtain $1 + (n-2)n = (n-1)^2$.

The basic strategy: expansion

- The basic strategy is to work backwards.
- Let (Q, A) be an n-state synchronizing automaton.
- Some letter, say a, must collapse two states to a state q.
- So $|qa^{-1}| \geq 2$.
- Suppose that, for each subset $S \subset Q$ with $1 < |S| < |Q|$, we can find a word $w \in A^*$ with $|Sw^{-1}| > |S|$.
- Then we can find a reset word of length at most $1 + (n-2)k$.
- Indeed, we can find a word w_1 with $|w_1| \leq k$ and $|qa^{-1}w_1^{-1}| > |qa^{-1}| \geq 2$.
- Continuing in this fashion, we can find words $w_2, \ldots, w_m$ with $m \leq n-2$ and $|w_i| \leq k$ such that $|qa^{-1}w_1^{-1} \cdots w_m^{-1}| = |Q|$.
- If $k = n$, then we obtain $1 + (n-2)n = (n-1)^2$.

The basic strategy: expansion

- The basic strategy is to work backwards.
- Let (Q, A) be an n-state synchronizing automaton.
- Some letter, say a, must collapse two states to a state q.
- So $|qa^{-1}| \geq 2$.
- Suppose that, for each subset $S \subset Q$ with $1 < |S| < |Q|$, we can find a word $w \in A^*$ with $|Sw^{-1}| > |S|$.
- Then we can find a reset word of length at most $1 + (n-2)k$.
- Indeed, we can find a word w_1 with $|w_1| \leq k$ and $|qa^{-1}w_1^{-1}| > |qa^{-1}| \geq 2$.
- Continuing in this fashion, we can find words $w_2, \ldots, w_m$ with $m \leq n-2$ and $|w_i| \leq k$ such that $|qa^{-1}w_1^{-1} \cdots w_m^{-1}| = |Q|$.
- If $k = n$, then we obtain $1 + (n-2)n = (n-1)^2$.

The basic strategy: expansion

- The basic strategy is to work backwards.
- Let (Q, A) be an n-state synchronizing automaton.
- Some letter, say a, must collapse two states to a state q.
- So $|qa^{-1}| \geq 2$.
- Suppose that, for each subset $S \subset Q$ with $1 < |S| < |Q|$, we can find a word $w \in A^*$ with $|Sw^{-1}| > |S|$.
- Then we can find a reset word of length at most $1 + (n-2)k$.
- Indeed, we can find a word w_1 with $|w_1| \leq k$ and $|qa^{-1}w_1^{-1}| > |qa^{-1}| \geq 2$.
- Continuing in this fashion, we can find words $w_2, \ldots, w_m$ with $m \leq n-2$ and $|w_i| \leq k$ such that $|qa^{-1}w_1^{-1} \cdots w_m^{-1}| = |Q|$.
- If $k = n$, then we obtain $1 + (n-2)n = (n-1)^2$.

One-cluster automata

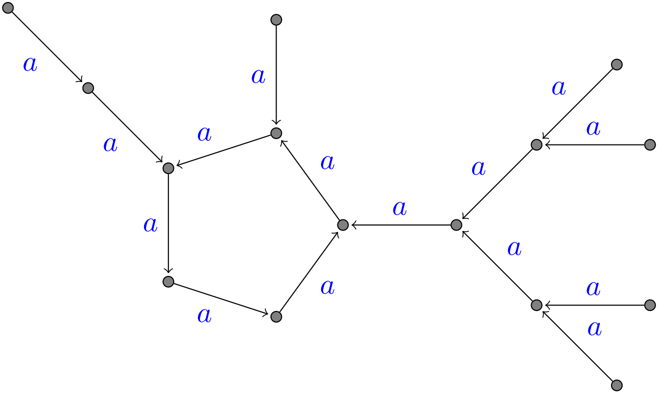
- Béal and Perrin introduced a modified version of the basic strategy in the context of one-cluster automata.
- (Q, A) is a one-cluster automaton if there is a letter a such that the subgraph consisting of edges labelled by a is connected. For example:

One-cluster automata

- Béal and Perrin introduced a modified version of the basic strategy in the context of one-cluster automata.
- (Q, A) is a one-cluster automaton if there is a letter a such that the subgraph consisting of edges labelled by a is connected. For example:

One-cluster automata

- Béal and Perrin introduced a modified version of the basic strategy in the context of one-cluster automata.
- (Q, A) is a one-cluster automaton if there is a letter a such that the subgraph consisting of edges labelled by a is connected. For example:



The modified basic strategy

- Suppose one has a special subset $C \subseteq Q$ and a "short" word u with $Qu \subseteq C$.
- For one-cluster automata, C is the a-cycle and $u = a^{|Q|-|C|}$.
- Since one can "push" Q into C efficiently, we just need to expand subsets of C.
- Suppose that, for each $\emptyset \neq S \subsetneq C$, we can find a word w of length at most k such that $|Sw^{-1} \cap C| > |S|$.
- Then we can find a reset word of length at most $(|C|-1)k + |u|$.
- First we expand from one state of C to all of C using repeatedly words of length at most k and then we apply u^{-1}.
- We lose here the ability to obtain the first expansion with a single letter: a serious problem!

The modified basic strategy

- Suppose one has a special subset $C \subseteq Q$ and a "short" word u with $Qu \subseteq C$.
- For one-cluster automata, C is the a-cycle and $u = a^{|Q|-|C|}$.
- Since one can "push" Q into C efficiently, we just need to expand subsets of C.
- Suppose that, for each $\emptyset \neq S \subsetneq C$, we can find a word w of length at most k such that $|Sw^{-1} \cap C| > |S|$.
- Then we can find a reset word of length at most $(|C|-1)k + |u|$.
- First we expand from one state of C to all of C using repeatedly words of length at most k and then we apply u^{-1}.
- We lose here the ability to obtain the first expansion with a single letter: a serious problem!

The modified basic strategy

- Suppose one has a special subset $C \subseteq Q$ and a "short" word u with $Qu \subseteq C$.
- For one-cluster automata, C is the a-cycle and $u = a^{|Q|-|C|}$.
- Since one can "push" Q into C efficiently, we just need to expand subsets of C.
- Suppose that, for each $\emptyset \neq S \subsetneq C$, we can find a word w of length at most k such that $|Sw^{-1} \cap C| > |S|$.
- Then we can find a reset word of length at most $(|C|-1)k + |u|$.
- First we expand from one state of C to all of C using repeatedly words of length at most k and then we apply u^{-1}.
- We lose here the ability to obtain the first expansion with a single letter: a serious problem!

The modified basic strategy

- Suppose one has a special subset $C \subseteq Q$ and a "short" word u with $Qu \subseteq C$.
- For one-cluster automata, C is the a-cycle and $u = a^{|Q|-|C|}$.
- Since one can "push" Q into C efficiently, we just need to expand subsets of C.
- Suppose that, for each $\emptyset \neq S \subsetneq C$, we can find a word w of length at most k such that $|Sw^{-1} \cap C| > |S|$.
- Then we can find a reset word of length at most $(|C|-1)k + |u|$.
- First we expand from one state of C to all of C using repeatedly words of length at most k and then we apply u^{-1}.
- We lose here the ability to obtain the first expansion with a single letter: a serious problem!

The modified basic strategy

- Suppose one has a special subset $C \subseteq Q$ and a "short" word u with $Qu \subseteq C$.
- For one-cluster automata, C is the a-cycle and $u = a^{|Q|-|C|}$.
- Since one can "push" Q into C efficiently, we just need to expand subsets of C.
- Suppose that, for each $\emptyset \neq S \subsetneq C$, we can find a word w of length at most k such that $|Sw^{-1} \cap C| > |S|$.
- Then we can find a reset word of length at most $(|C|-1)k + |u|$.
- First we expand from one state of C to all of C using repeatedly words of length at most k and then we apply u^{-1}.
- We lose here the ability to obtain the first expansion with a single letter: a serious problem!

The modified basic strategy

- Suppose one has a special subset $C \subseteq Q$ and a "short" word u with $Qu \subseteq C$.
- For one-cluster automata, C is the a-cycle and $u = a^{|Q|-|C|}$.
- Since one can "push" Q into C efficiently, we just need to expand subsets of C.
- Suppose that, for each $\emptyset \neq S \subsetneq C$, we can find a word w of length at most k such that $|Sw^{-1} \cap C| > |S|$.
- Then we can find a reset word of length at most $(|C|-1)k + |u|$.
- First we expand from one state of C to all of C using repeatedly words of length at most k and then we apply u^{-1}.
- We lose here the ability to obtain the first expansion with a single letter: a serious problem!

The modified basic strategy

- Suppose one has a special subset $C \subseteq Q$ and a "short" word u with $Qu \subseteq C$.
- For one-cluster automata, C is the a-cycle and $u = a^{|Q|-|C|}$.
- Since one can "push" Q into C efficiently, we just need to expand subsets of C.
- Suppose that, for each $\emptyset \neq S \subsetneq C$, we can find a word w of length at most k such that $|Sw^{-1} \cap C| > |S|$.
- Then we can find a reset word of length at most $(|C|-1)k + |u|$.
- First we expand from one state of C to all of C using repeatedly words of length at most k and then we apply u^{-1}.
- We lose here the ability to obtain the first expansion with a single letter: a serious problem!

The averaging trick: a summary

- Most papers that use this expansion technique rely on the following two simple observations:

Observation

A non-constant function must exceed its average value.

Observation

A strict chain of non-zero subspaces of an n-dimensional vector space has length at most n.

- Some papers use the language of rational power series to avoid discussing strict chains of subspaces.

The averaging trick: a summary

- Most papers that use this expansion technique rely on the following two simple observations:

Observation

A non-constant function must exceed its average value.

Observation

A strict chain of non-zero subspaces of an n-dimensional vector space has length at most n.

- Some papers use the language of rational power series to avoid discussing strict chains of subspaces.

The averaging trick: a summary

- Most papers that use this expansion technique rely on the following two simple observations:

Observation

A non-constant function must exceed its average value.

Observation

A strict chain of non-zero subspaces of an n-dimensional vector space has length at most n.

- Some papers use the language of rational power series to avoid discussing strict chains of subspaces.

The averaging trick: a summary

- Most papers that use this expansion technique rely on the following two simple observations:

Observation

A non-constant function must exceed its average value.

Observation

A strict chain of non-zero subspaces of an n-dimensional vector space has length at most n.

- Some papers use the language of rational power series to avoid discussing strict chains of subspaces.

An incomplete list of authors using this technique

- Pin for circular automata with a prime number of states.
- Rystsov for regular automata.
- Dubuc for circular automata in general.
- Kari for Eulerian automata.
- Béal, Berlinkov and Perrin for one-cluster automata.
- D'Alessandro and Carpi for strongly transitive and locally strongly transitive automata.
- The speaker for Cayley graphs of groups and for one-cluster automata.

An incomplete list of authors using this technique

- Pin for circular automata with a prime number of states.
- Rystsov for regular automata.
- Dubuc for circular automata in general.
- Kari for Eulerian automata.
- Béal, Berlinkov and Perrin for one-cluster automata.
- D'Alessandro and Carpi for strongly transitive and locally strongly transitive automata.
- The speaker for Cayley graphs of groups and for one-cluster automata.

An incomplete list of authors using this technique

- Pin for circular automata with a prime number of states.
- Rystsov for regular automata.
- Dubuc for circular automata in general.
- Kari for Eulerian automata.
- Béal, Berlinkov and Perrin for one-cluster automata.
- D'Alessandro and Carpi for strongly transitive and locally strongly transitive automata.
- The speaker for Cayley graphs of groups and for one-cluster automata.

An incomplete list of authors using this technique

- Pin for circular automata with a prime number of states.
- Rystsov for regular automata.
- Dubuc for circular automata in general.
- Kari for Eulerian automata.
- Béal, Berlinkov and Perrin for one-cluster automata.
- D'Alessandro and Carpi for strongly transitive and locally strongly transitive automata.
- The speaker for Cayley graphs of groups and for one-cluster automata.

An incomplete list of authors using this technique

- Pin for circular automata with a prime number of states.
- Rystsov for regular automata.
- Dubuc for circular automata in general.
- Kari for Eulerian automata.
- Béal, Berlinkov and Perrin for one-cluster automata.
- D'Alessandro and Carpi for strongly transitive and locally strongly transitive automata.
- The speaker for Cayley graphs of groups and for one-cluster automata.

An incomplete list of authors using this technique

- Pin for circular automata with a prime number of states.
- Rystsov for regular automata.
- Dubuc for circular automata in general.
- Kari for Eulerian automata.
- Béal, Berlinkov and Perrin for one-cluster automata.
- D'Alessandro and Carpi for strongly transitive and locally strongly transitive automata.
- The speaker for Cayley graphs of groups and for one-cluster automata.

An incomplete list of authors using this technique

- Pin for circular automata with a prime number of states.
- Rystsov for regular automata.
- Dubuc for circular automata in general.
- Kari for Eulerian automata.
- Béal, Berlinkov and Perrin for one-cluster automata.
- D'Alessandro and Carpi for strongly transitive and locally strongly transitive automata.
- The speaker for Cayley graphs of groups and for one-cluster automata.

How it works: a high level description

- Recall our setup: (Q, A) is a synchronizing automaton with n states.
- $\emptyset \subsetneq S \subsetneq Q$.
- We want a word w of length at most k with $|Sw^{-1}| > |S|$.
- If w is a reset word, then $|Sw^{-1}| = |Q| > |S|$.
- Typically one shows that under the hypothesis of one's paper, the average value of $|Sw^{-1}|$ on some well-chosen subset X of words of length at most k is $|S|$.
- One has $|Sw^{-1}| = [Q]w[S]^T$ where $[Y]$ denotes the characteristic vector of $Y \subseteq Q$.
- One deduces that $|Sw^{-1}|$ is not constant for some word $w \in X$ by an ascending chain of subspaces argument.
- Therefore, $|Sw^{-1}| > |S|$ for some $w \in X$
- Let's formalize this!

How it works: a high level description

- Recall our setup: (Q, A) is a synchronizing automaton with n states.
- $\emptyset \subsetneq S \subsetneq Q$.
- We want a word w of length at most k with $|Sw^{-1}| > |S|$.
- If w is a reset word, then $|Sw^{-1}| = |Q| > |S|$.
- Typically one shows that under the hypothesis of one's paper, the average value of $|Sw^{-1}|$ on some well-chosen subset X of words of length at most k is $|S|$.
- One has $|Sw^{-1}| = [Q]w[S]^T$ where $[Y]$ denotes the characteristic vector of $Y \subseteq Q$.
- One deduces that $|Sw^{-1}|$ is not constant for some word $w \in X$ by an ascending chain of subspaces argument.
- Therefore, $|Sw^{-1}| > |S|$ for some $w \in X$
- Let's formalize this!

How it works: a high level description

- Recall our setup: (Q, A) is a synchronizing automaton with n states.
- $\emptyset \subsetneq S \subsetneq Q$.
- We want a word w of length at most k with $|Sw^{-1}| > |S|$.
- If w is a reset word, then $|Sw^{-1}| = |Q| > |S|$.
- Typically one shows that under the hypothesis of one's paper, the average value of $|Sw^{-1}|$ on some well-chosen subset X of words of length at most k is $|S|$.
- One has $|Sw^{-1}| = [Q]w[S]^T$ where $[Y]$ denotes the characteristic vector of $Y \subseteq Q$.
- One deduces that $|Sw^{-1}|$ is not constant for some word $w \in X$ by an ascending chain of subspaces argument.
- Therefore, $|Sw^{-1}| > |S|$ for some $w \in X$
- Let's formalize this!

How it works: a high level description

- Recall our setup: (Q, A) is a synchronizing automaton with n states.
- $\emptyset \subsetneq S \subsetneq Q$.
- We want a word w of length at most k with $|Sw^{-1}| > |S|$.
- If w is a reset word, then $|Sw^{-1}| = |Q| > |S|$.
- Typically one shows that under the hypothesis of one's paper, the average value of $|Sw^{-1}|$ on some well-chosen subset X of words of length at most k is $|S|$.
- One has $|Sw^{-1}| = [Q]w[S]^T$ where $[Y]$ denotes the characteristic vector of $Y \subseteq Q$.
- One deduces that $|Sw^{-1}|$ is not constant for some word $w \in X$ by an ascending chain of subspaces argument.
- Therefore, $|Sw^{-1}| > |S|$ for some $w \in X$
- Let's formalize this!

How it works: a high level description

- Recall our setup: (Q, A) is a synchronizing automaton with n states.
- $\emptyset \subsetneq S \subsetneq Q$.
- We want a word w of length at most k with $|Sw^{-1}| > |S|$.
- If w is a reset word, then $|Sw^{-1}| = |Q| > |S|$.
- Typically one shows that under the hypothesis of one's paper, the average value of $|Sw^{-1}|$ on some well-chosen subset X of words of length at most k is $|S|$.
- One has $|Sw^{-1}| = [Q]w[S]^T$ where $[Y]$ denotes the characteristic vector of $Y \subseteq Q$.
- One deduces that $|Sw^{-1}|$ is not constant for some word $w \in X$ by an ascending chain of subspaces argument.
- Therefore, $|Sw^{-1}| > |S|$ for some $w \in X$
- Let's formalize this!

How it works: a high level description

- Recall our setup: (Q, A) is a synchronizing automaton with n states.
- $\emptyset \subsetneq S \subsetneq Q$.
- We want a word w of length at most k with $|Sw^{-1}| > |S|$.
- If w is a reset word, then $|Sw^{-1}| = |Q| > |S|$.
- Typically one shows that under the hypothesis of one's paper, the average value of $|Sw^{-1}|$ on some well-chosen subset X of words of length at most k is $|S|$.
- One has $|Sw^{-1}| = [Q]w[S]^T$ where $[Y]$ denotes the characteristic vector of $Y \subseteq Q$.
- One deduces that $|Sw^{-1}|$ is not constant for some word $w \in X$ by an ascending chain of subspaces argument.
- Therefore, $|Sw^{-1}| > |S|$ for some $w \in X$
- Let's formalize this!

How it works: a high level description

- Recall our setup: (Q, A) is a synchronizing automaton with n states.
- $\emptyset \subsetneq S \subsetneq Q$.
- We want a word w of length at most k with $|Sw^{-1}| > |S|$.
- If w is a reset word, then $|Sw^{-1}| = |Q| > |S|$.
- Typically one shows that under the hypothesis of one's paper, the average value of $|Sw^{-1}|$ on some well-chosen subset X of words of length at most k is $|S|$.
- One has $|Sw^{-1}| = [Q]w[S]^T$ where $[Y]$ denotes the characteristic vector of $Y \subseteq Q$.
- One deduces that $|Sw^{-1}|$ is not constant for some word $w \in X$ by an ascending chain of subspaces argument.
- Therefore, $|Sw^{-1}| > |S|$ for some $w \in X$
- Let's formalize this!

How it works: a high level description

- Recall our setup: (Q, A) is a synchronizing automaton with n states.
- $\emptyset \subsetneq S \subsetneq Q$.
- We want a word w of length at most k with $|Sw^{-1}| > |S|$.
- If w is a reset word, then $|Sw^{-1}| = |Q| > |S|$.
- Typically one shows that under the hypothesis of one's paper, the average value of $|Sw^{-1}|$ on some well-chosen subset X of words of length at most k is $|S|$.
- One has $|Sw^{-1}| = [Q]w[S]^T$ where $[Y]$ denotes the characteristic vector of $Y \subseteq Q$.
- One deduces that $|Sw^{-1}|$ is not constant for some word $w \in X$ by an ascending chain of subspaces argument.
- Therefore, $|Sw^{-1}| > |S|$ for some $w \in X$
- Let's formalize this!

How it works: a high level description

- Recall our setup: (Q, A) is a synchronizing automaton with n states.
- $\emptyset \subsetneq S \subsetneq Q$.
- We want a word w of length at most k with $|Sw^{-1}| > |S|$.
- If w is a reset word, then $|Sw^{-1}| = |Q| > |S|$.
- Typically one shows that under the hypothesis of one's paper, the average value of $|Sw^{-1}|$ on some well-chosen subset X of words of length at most k is $|S|$.
- One has $|Sw^{-1}| = [Q]w[S]^T$ where $[Y]$ denotes the characteristic vector of $Y \subseteq Q$.
- One deduces that $|Sw^{-1}|$ is not constant for some word $w \in X$ by an ascending chain of subspaces argument.
- Therefore, $|Sw^{-1}| > |S|$ for some $w \in X$
- Let's formalize this!

A formalization

- $\mathbb{R}A$ is the ring of polynomials in non-commuting variables A over $\mathbb{R}$.
- A probability on A^* is an element

$$P = \sum_{w \in A^*} P(w)w \in \mathbb{R}A$$

such that: $P(w) \geq 0$ for all $w \in A^*$, and

$$\sum_{w \in A^*} P(w) = 1.$$

- The support of P is

$$\sigma(P) = \{w \in A^* \mid P(w) > 0\}.$$

- If P_1 and P_2 are probabilities, then so is P_1P_2 and moreover $\sigma(P_1P_2) = \sigma(P_1)\sigma(P_2)$.

A formalization

- $\mathbb{R}A$ is the ring of polynomials in non-commuting variables A over $\mathbb{R}$.
- A probability on A^* is an element

$$P = \sum_{w \in A^*} P(w)w \in \mathbb{R}A$$

such that: $P(w) \geq 0$ for all $w \in A^*$, and

$$\sum_{w \in A^*} P(w) = 1.$$

- The support of P is

$$\sigma(P) = \{w \in A^* \mid P(w) > 0\}.$$

- If P_1 and P_2 are probabilities, then so is $P_1 P_2$ and moreover $\sigma(P_1 P_2) = \sigma(P_1)\sigma(P_2)$.

A formalization

- $\mathbb{R}A$ is the ring of polynomials in non-commuting variables A over $\mathbb{R}$.
- A probability on A^* is an element

$$P = \sum_{w \in A^*} P(w)w \in \mathbb{R}A$$

such that: $P(w) \geq 0$ for all $w \in A^*$, and

$$\sum_{w \in A^*} P(w) = 1.$$

- The support of P is

$$\sigma(P) = \{w \in A^* \mid P(w) > 0\}.$$

- If P_1 and P_2 are probabilities, then so is P_1P_2 and moreover $\sigma(P_1P_2) = \sigma(P_1)\sigma(P_2)$.

A formalization

- $\mathbb{R}A$ is the ring of polynomials in non-commuting variables A over $\mathbb{R}$.
- A probability on A^* is an element

$$P = \sum_{w \in A^*} P(w)w \in \mathbb{R}A$$

such that: $P(w) \geq 0$ for all $w \in A^*$, and

$$\sum_{w \in A^*} P(w) = 1.$$

- The support of P is

$$\sigma(P) = \{w \in A^* \mid P(w) > 0\}.$$

- If P_1 and P_2 are probabilities, then so is P_1P_2 and moreover $\sigma(P_1P_2) = \sigma(P_1)\sigma(P_2)$.

A formalization

- $\mathbb{R}A$ is the ring of polynomials in non-commuting variables A over $\mathbb{R}$.
- A probability on A^* is an element
$$P = \sum_{w \in A^*} P(w)w \in \mathbb{R}A$$
such that: $P(w) \geq 0$ for all $w \in A^*$, and
$$\sum_{w \in A^*} P(w) = 1.$$
- The support of P is
$$\sigma(P) = \{w \in A^* \mid P(w) > 0\}.$$
- If P_1 and P_2 are probabilities, then so is P_1P_2 and moreover $\sigma(P_1P_2) = \sigma(P_1)\sigma(P_2)$.

Expected value

- If $X\colon A^* \to \mathbb{R}$ is a random variable, the expected value of X with respect to the probability P is:

$$\mathbf{E}_P(X) = \sum_{w\in A^*} P(w)X(w) = \sum_{w\in\sigma(P)} P(w)X(w).$$

- A random variable that is not almost surely constant, must exceed it expected value with positive probability.
- More precisely, if X is not constant on the support of P, then it exceeds its expected value somewhere on the support of P.

Expected value

- If $X\colon A^* \to \mathbb{R}$ is a random variable, the expected value of X with respect to the probability P is:

$$\mathbf{E}_P(X) = \sum_{w\in A^*} P(w)X(w) = \sum_{w\in\sigma(P)} P(w)X(w).$$

- A random variable that is not almost surely constant, must exceed it expected value with positive probability.
- More precisely, if X is not constant on the support of P, then it exceeds its expected value somewhere on the support of P.

Expected value

- If $X\colon A^* \to \mathbb{R}$ is a random variable, the expected value of X with respect to the probability P is:

$$\mathbf{E}_P(X) = \sum_{w\in A^*} P(w)X(w) = \sum_{w\in\sigma(P)} P(w)X(w).$$

- A random variable that is not almost surely constant, must exceed it expected value with positive probability.
- More precisely, if X is not constant on the support of P, then it exceeds its expected value somewhere on the support of P.

The matrix representation

- Suppose that $Q = \{1, \ldots, n\}$.
- For each $w \in A$, we have the associated matrix $M(w)$ where

$$M(w)_{ij} = \begin{cases} 1 & iw = j \\ 0 & \text{else.} \end{cases}$$

- This extends to a homomorphism $\pi\colon \mathbb{R}A \to M_n(\mathbb{R})$ by

$$\pi\left(\sum_{w \in A^*} f(w)w\right) = \sum_{w \in A^*} f(w)M(w).$$

- If P is a probability on A^*, then $\pi(P)$ is a stochastic matrix.
- That is, each row of $\pi(P)$ is a probability vector.

The matrix representation

- Suppose that $Q = \{1, \ldots, n\}$.
- For each $w \in A$, we have the associated matrix $M(w)$ where

$$M(w)_{ij} = \begin{cases} 1 & iw = j \\ 0 & \text{else.} \end{cases}$$

- This extends to a homomorphism $\pi \colon \mathbb{R}A \to M_n(\mathbb{R})$ by

$$\pi\left(\sum_{w \in A^*} f(w)w\right) = \sum_{w \in A^*} f(w)M(w).$$

- If P is a probability on A^*, then $\pi(P)$ is a stochastic matrix.
- That is, each row of $\pi(P)$ is a probability vector.

The matrix representation

- Suppose that $Q = \{1, \ldots, n\}$.
- For each $w \in A$, we have the associated matrix $M(w)$ where

$$M(w)_{ij} = \begin{cases} 1 & iw = j \\ 0 & \text{else.} \end{cases}$$

- This extends to a homomorphism $\pi \colon \mathbb{R}A \to M_n(\mathbb{R})$ by

$$\pi\left(\sum_{w \in A^*} f(w)w\right) = \sum_{w \in A^*} f(w)M(w).$$

- If P is a probability on A^*, then $\pi(P)$ is a stochastic matrix.
- That is, each row of $\pi(P)$ is a probability vector.

The matrix representation

- Suppose that $Q = \{1, \ldots, n\}$.
- For each $w \in A$, we have the associated matrix $M(w)$ where

$$M(w)_{ij} = \begin{cases} 1 & iw = j \\ 0 & \text{else.} \end{cases}$$

- This extends to a homomorphism $\pi \colon \mathbb{R}A \to M_n(\mathbb{R})$ by

$$\pi\left(\sum_{w \in A^*} f(w)w\right) = \sum_{w \in A^*} f(w)M(w).$$

- If P is a probability on A^*, then $\pi(P)$ is a stochastic matrix.
- That is, each row of $\pi(P)$ is a probability vector.

The matrix representation

- Suppose that $Q = \{1, \ldots, n\}$.
- For each $w \in A$, we have the associated matrix $M(w)$ where

$$M(w)_{ij} = \begin{cases} 1 & iw = j \\ 0 & \text{else.} \end{cases}$$

- This extends to a homomorphism $\pi\colon \mathbb{R}A \to M_n(\mathbb{R})$ by

$$\pi\left(\sum_{w\in A^*} f(w)w\right) = \sum_{w\in A^*} f(w)M(w).$$

- If P is a probability on A^*, then $\pi(P)$ is a stochastic matrix.
- That is, each row of $\pi(P)$ is a probability vector.

The technical statement: version 1

- Let us first state a variant of our result in the context of the original version of the basic strategy.

Lemma (Averaging Lemma 1)

Let $\mathscr{A} = (Q, A)$ be a strongly connected synchronizing automaton with n states and let P_1, P_2 be probabilities on A^* such that:

1. *P_2 has support $A^{\leq n-1}$;*
2. *$[Q]P_2P_1 = [Q]$.*

Then $\mathscr{A}$ has a reset word of length at most $1 + (n-2)(n-1+L)$ where L is the maximum length of a word in $\sigma(P_1)$.

The technical statement: version 1

- Let us first state a variant of our result in the context of the original version of the basic strategy.

Lemma (Averaging Lemma 1)

Let $\mathscr{A} = (Q, A)$ be a strongly connected synchronizing automaton with n states and let P_1, P_2 be probabilities on A^* such that:

1. *P_2 has support $A^{\leq n-1}$;*
2. *$[Q]P_2P_1 = [Q]$.*

Then $\mathscr{A}$ has a reset word of length at most $1 + (n-2)(n-1+L)$ where L is the maximum length of a word in $\sigma(P_1)$.

The technical statement: version 1

- Let us first state a variant of our result in the context of the original version of the basic strategy.

Lemma (Averaging Lemma 1)

Let $\mathscr{A} = (Q, A)$ be a strongly connected synchronizing automaton with n states and let P_1, P_2 be probabilities on A^* such that:

1. P_2 *has support* $A^{\leq n-1}$;
2. $[Q]P_2P_1 = [Q]$.

Then $\mathscr{A}$ has a reset word of length at most $1 + (n-2)(n-1+L)$ where L is the maximum length of a word in $\sigma(P_1)$.

The technical statement: version 1

- Let us first state a variant of our result in the context of the original version of the basic strategy.

Lemma (Averaging Lemma 1)

Let $\mathscr{A} = (Q, A)$ be a strongly connected synchronizing automaton with n states and let P_1, P_2 be probabilities on A^* such that:

1. *P_2 has support $A^{\leq n-1}$;*
2. *$[Q]P_2P_1 = [Q]$.*

Then $\mathscr{A}$ has a reset word of length at most $1 + (n-2)(n-1+L)$ where L is the maximum length of a word in $\sigma(P_1)$.

The technical statement: version 1

- Let us first state a variant of our result in the context of the original version of the basic strategy.

Lemma (Averaging Lemma 1)

Let $\mathscr{A} = (Q, A)$ *be a strongly connected synchronizing automaton with* n *states and let* P_1, P_2 *be probabilities on* A^* *such that:*

1. P_2 *has support* $A^{\leq n-1}$*;*
2. $[Q]P_2P_1 = [Q]$.

Then $\mathscr{A}$ *has a reset word of length at most* $1 + (n-2)(n-1+L)$ *where* L *is the maximum length of a word in* $\sigma(P_1)$.

An example: pseudo-Eulerian automata

- Kari termed an automaton Eulerian if its underlying digraph has a directed Euler path.
- This is equivalent to saying that the in-degree of each vertex is $|A|$, i.e., each row and column sum of the adjacency matrix of (Q, A) is $|A|$.
- Hence if we define a probability

$$P = \frac{1}{|A|}\sum_{a\in A} a$$

then $\pi(P)$ is doubly stochastic.

- That means, each row and column is a probability vector.

Definition (Pseudo-Eulerian)

An automaton (Q, A) is pseudo-Eulerian if there exists a probability P with support the alphabet A such that $\pi(P)$ is doubly stochastic.

An example: pseudo-Eulerian automata

- Kari termed an automaton Eulerian if its underlying digraph has a directed Euler path.
- This is equivalent to saying that the in-degree of each vertex is $|A|$, i.e., each row and column sum of the adjacency matrix of (Q, A) is $|A|$.
- Hence if we define a probability

$$P = \frac{1}{|A|} \sum_{a \in A} a$$

  then $\pi(P)$ is doubly stochastic.
- That means, each row and column is a probability vector.

Definition (Pseudo-Eulerian)

An automaton (Q, A) is pseudo-Eulerian if there exists a probability P with support the alphabet A such that $\pi(P)$ is doubly stochastic.

An example: pseudo-Eulerian automata

- Kari termed an automaton Eulerian if its underlying digraph has a directed Euler path.
- This is equivalent to saying that the in-degree of each vertex is $|A|$, i.e., each row and column sum of the adjacency matrix of (Q, A) is $|A|$.
- Hence if we define a probability

  $$P = \frac{1}{|A|} \sum_{a \in A} a$$

  then $\pi(P)$ is doubly stochastic.
- That means, each row and column is a probability vector.

Definition (Pseudo-Eulerian)

An automaton (Q, A) is pseudo-Eulerian if there exists a probability P with support the alphabet A such that $\pi(P)$ is doubly stochastic.

An example: pseudo-Eulerian automata

- Kari termed an automaton Eulerian if its underlying digraph has a directed Euler path.
- This is equivalent to saying that the in-degree of each vertex is $|A|$, i.e., each row and column sum of the adjacency matrix of (Q, A) is $|A|$.
- Hence if we define a probability

$$P = \frac{1}{|A|} \sum_{a \in A} a$$

  then $\pi(P)$ is doubly stochastic.
- That means, each row and column is a probability vector.

Definition (Pseudo-Eulerian)

An automaton (Q, A) is pseudo-Eulerian if there exists a probability P with support the alphabet A such that $\pi(P)$ is doubly stochastic.

An example: pseudo-Eulerian automata

- Kari termed an automaton Eulerian if its underlying digraph has a directed Euler path.
- This is equivalent to saying that the in-degree of each vertex is $|A|$, i.e., each row and column sum of the adjacency matrix of (Q, A) is $|A|$.
- Hence if we define a probability

$$P = \frac{1}{|A|} \sum_{a \in A} a$$

then $\pi(P)$ is doubly stochastic.
- That means, each row and column is a probability vector.

Definition (Pseudo-Eulerian)

An automaton (Q, A) is pseudo-Eulerian if there exists a probability P with support the alphabet A such that $\pi(P)$ is doubly stochastic.

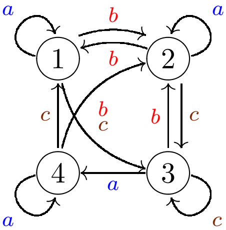
Pseudo-Eulerian automata 2

- Every Eulerian automaton is pseudo-Eulerian.
- The automaton below is pseudo-Eulerian but not Eulerian.

- If $P = a/2 + b/6 + c/3$, $\pi(P) = \begin{bmatrix} \frac{1}{2} & \frac{1}{6} & \frac{1}{3} & 0 \\ \frac{1}{6} & \frac{1}{2} & \frac{1}{3} & 0 \\ 0 & \frac{1}{6} & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{6} & 0 & \frac{1}{2} \end{bmatrix}$ is doubly stochastic.

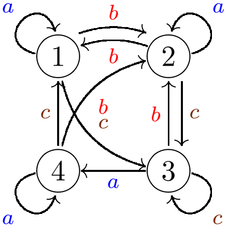
Pseudo-Eulerian automata 2

- Every Eulerian automaton is pseudo-Eulerian.
- The automaton below is pseudo-Eulerian but not Eulerian.



- If $P = a/2 + b/6 + c/3$, $\pi(P) = \begin{bmatrix} \frac{1}{2} & \frac{1}{6} & \frac{1}{3} & 0 \\ \frac{1}{6} & \frac{1}{2} & \frac{1}{3} & 0 \\ 0 & \frac{1}{6} & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{6} & 0 & \frac{1}{2} \end{bmatrix}$ is doubly stochastic.

- Every Eulerian automaton is pseudo-Eulerian.
- The automaton below is pseudo-Eulerian but not Eulerian.



- If $P = a/2 + b/6 + c/3$, $\pi(P) = \begin{bmatrix} \frac{1}{2} & \frac{1}{6} & \frac{1}{3} & 0 \\ \frac{1}{6} & \frac{1}{2} & \frac{1}{3} & 0 \\ 0 & \frac{1}{6} & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{6} & 0 & \frac{1}{2} \end{bmatrix}$ is doubly stochastic.

Pseudo-Eulerian automata 3

- It is a simple linear programming task to determine whether an automaton is pseudo-Eulerian.

Theorem

An n-state synchronizing pseudo-Eulerian automaton admits a reset word of length at most $1 + (n-2)(n-1) < (n-1)^2$.

Proof.

- Let P be a probability on A with $\pi(P)$ doubly stochastic.
- Let P_1 be a point mass at ε.
- Put $P_2 = \frac{1}{n}\sum_{m=0}^{n-1} P^m$; it has support $A^{\leq n-1}$.
- $\pi(P)$ doubly stochastic implies $[Q]P = [Q]$. So $[Q]P_2P_1 = [Q] \cdot \frac{1}{n}\sum_{m=0}^{n-1} P^m = [Q]$.
- The Averaging Lemma yields the desired upper bound.

$\square$

- It is a simple linear programming task to determine whether an automaton is pseudo-Eulerian.

Theorem

An n-state synchronizing pseudo-Eulerian automaton admits a reset word of length at most $1 + (n-2)(n-1) < (n-1)^2$.

Proof.

- Let P be a probability on A with $\pi(P)$ doubly stochastic.
- Let P_1 be a point mass at ε.
- Put $P_2 = \frac{1}{n}\sum_{m=0}^{n-1} P^m$; it has support $A^{\leq n-1}$.
- $\pi(P)$ doubly stochastic implies $[Q]P = [Q]$. So $[Q]P_2P_1 = [Q] \cdot \frac{1}{n}\sum_{m=0}^{n-1} P^m = [Q]$.
- The Averaging Lemma yields the desired upper bound.

Pseudo-Eulerian automata 3

- It is a simple linear programming task to determine whether an automaton is pseudo-Eulerian.

Theorem

An n-state synchronizing pseudo-Eulerian automaton admits a reset word of length at most $1 + (n-2)(n-1) < (n-1)^2$.

Proof.

- Let P be a probability on A with $\pi(P)$ doubly stochastic.
- Let P_1 be a point mass at ε.
- Put $P_2 = \frac{1}{n}\sum_{m=0}^{n-1} P^m$; it has support $A^{\leq n-1}$.
- $\pi(P)$ doubly stochastic implies $[Q]P = [Q]$. So $[Q]P_2P_1 = [Q] \cdot \frac{1}{n}\sum_{m=0}^{n-1} P^m = [Q]$.
- The Averaging Lemma yields the desired upper bound.

Pseudo-Eulerian automata 3

- It is a simple linear programming task to determine whether an automaton is pseudo-Eulerian.

Theorem

An n-state synchronizing pseudo-Eulerian automaton admits a reset word of length at most $1 + (n-2)(n-1) < (n-1)^2$.

Proof.

- Let P be a probability on A with $\pi(P)$ doubly stochastic.
- Let P_1 be a point mass at ε.
- Put $P_2 = \frac{1}{n}\sum_{m=0}^{n-1} P^m$; it has support $A^{\leq n-1}$.
- $\pi(P)$ doubly stochastic implies $[Q]P = [Q]$. So $[Q]P_2P_1 = [Q] \cdot \frac{1}{n}\sum_{m=0}^{n-1} P^m = [Q]$.
- The Averaging Lemma yields the desired upper bound.

□

Pseudo-Eulerian automata 3

- It is a simple linear programming task to determine whether an automaton is pseudo-Eulerian.

Theorem

An n-state synchronizing pseudo-Eulerian automaton admits a reset word of length at most $1+(n-2)(n-1)<(n-1)^2$.

Proof.

- Let P be a probability on A with $\pi(P)$ doubly stochastic.
- Let P_1 be a point mass at ε.
- Put $P_2 = \frac{1}{n}\sum_{m=0}^{n-1} P^m$; it has support $A^{\leq n-1}$.
- $\pi(P)$ doubly stochastic implies $[Q]P=[Q]$. So $[Q]P_2P_1 = [Q]\cdot\frac{1}{n}\sum_{m=0}^{n-1} P^m = [Q]$.
- The Averaging Lemma yields the desired upper bound.

□

Pseudo-Eulerian automata 3

- It is a simple linear programming task to determine whether an automaton is pseudo-Eulerian.

Theorem

An n-state synchronizing pseudo-Eulerian automaton admits a reset word of length at most $1 + (n-2)(n-1) < (n-1)^2$.

Proof.

- Let P be a probability on A with $\pi(P)$ doubly stochastic.
- Let P_1 be a point mass at ε.
- Put $P_2 = \frac{1}{n}\sum_{m=0}^{n-1} P^m$; it has support $A^{\leq n-1}$.
- $\pi(P)$ doubly stochastic implies $[Q]P = [Q]$. So $[Q]P_2P_1 = [Q] \cdot \frac{1}{n}\sum_{m=0}^{n-1} P^m = [Q]$.
- The Averaging Lemma yields the desired upper bound.

□

Pseudo-Eulerian automata 3

- It is a simple linear programming task to determine whether an automaton is pseudo-Eulerian.

Theorem

An n-state synchronizing pseudo-Eulerian automaton admits a reset word of length at most $1 + (n-2)(n-1) < (n-1)^2$.

Proof.

- Let P be a probability on A with $\pi(P)$ doubly stochastic.
- Let P_1 be a point mass at ε.
- Put $P_2 = \frac{1}{n}\sum_{m=0}^{n-1} P^m$; it has support $A^{\leq n-1}$.
- $\pi(P)$ doubly stochastic implies $[Q]P = [Q]$. So $[Q]P_2P_1 = [Q] \cdot \frac{1}{n}\sum_{m=0}^{n-1} P^m = [Q]$.
- The Averaging Lemma yields the desired upper bound.

□

The technical statement: version 2

- Here is a version adapted to the modified basic strategy.

Lemma (Averaging Lemma 2)

Let $\mathscr{A} = (Q, A)$ be an n-state synchronizing automaton. Suppose there exist probabilities P_1, P_2 on A^* and $R \subsetneq Q$ such that:

1. *P_2 has support $A^{\leq n-1}$;*
2. $[R]P_2P_1 = [R]$;
3. *$R \subseteq qA^*$ for all $q \in R$;*
4. *there exists $w_0 \in A^*$ with $Qw_0 \subseteq R$.*

Then $\mathscr{A}$ has a reset word of length at most

$$(|R| - 1)(n - 1 + L) + |w_0|$$

where L is the maximum length of a word in $\sigma(P_1)$.

- The statement in the paper is even more technical than this!

The technical statement: version 2

- Here is a version adapted to the modified basic strategy.

Lemma (Averaging Lemma 2)

Let $\mathscr{A} = (Q, A)$ be an n-state synchronizing automaton. Suppose there exist probabilities P_1, P_2 on A^* and $R \subsetneq Q$ such that:

1. *P_2 has support $A^{\leq n-1}$;*
2. *$[R]P_2P_1 = [R]$;*
3. *$R \subseteq qA^*$ for all $q \in R$;*
4. *there exists $w_0 \in A^*$ with $Qw_0 \subseteq R$.*

Then $\mathscr{A}$ has a reset word of length at most

$$(|R| - 1)(n - 1 + L) + |w_0|$$

where L is the maximum length of a word in $\sigma(P_1)$.

- The statement in the paper is even more technical than this!

The technical statement: version 2

- Here is a version adapted to the modified basic strategy.

Lemma (Averaging Lemma 2)

Let $\mathscr{A} = (Q, A)$ be an n-state synchronizing automaton. Suppose there exist probabilities P_1, P_2 on A^* and $R \subsetneq Q$ such that:

1. *P_2 has support $A^{\leq n-1}$;*
2. *$[R]P_2P_1 = [R]$;*
3. *$R \subseteq qA^*$ for all $q \in R$;*
4. *there exists $w_0 \in A^*$ with $Qw_0 \subseteq R$.*

Then $\mathscr{A}$ has a reset word of length at most

$$(|R| - 1)(n - 1 + L) + |w_0|$$

where L is the maximum length of a word in $\sigma(P_1)$.

- The statement in the paper is even more technical than this!

The technical statement: version 2

- Here is a version adapted to the modified basic strategy.

Lemma (Averaging Lemma 2)

Let $\mathscr{A} = (Q, A)$ be an n-state synchronizing automaton. Suppose there exist probabilities P_1, P_2 on A^* and $R \subsetneq Q$ such that:

1. P_2 *has support* $A^{\leq n-1}$;
2. $[R]P_2P_1 = [R]$;
3. $R \subseteq qA^*$ *for all* $q \in R$;
4. *there exists* $w_0 \in A^*$ *with* $Qw_0 \subseteq R$.

Then $\mathscr{A}$ has a reset word of length at most

$$(|R| - 1)(n - 1 + L) + |w_0|$$

where L is the maximum length of a word in $\sigma(P_1)$.

- The statement in the paper is even more technical than this!

The technical statement: version 2

- Here is a version adapted to the modified basic strategy.

Lemma (Averaging Lemma 2)

Let $\mathscr{A} = (Q, A)$ be an n-state synchronizing automaton. Suppose there exist probabilities P_1, P_2 on A^* and $R \subsetneq Q$ such that:

1. *P_2 has support $A^{\leq n-1}$;*
2. *$[R]P_2P_1 = [R]$;*
3. *$R \subseteq qA^*$ for all $q \in R$;*
4. *there exists $w_0 \in A^*$ with $Qw_0 \subseteq R$.*

Then $\mathscr{A}$ has a reset word of length at most

$$(|R| - 1)(n - 1 + L) + |w_0|$$

where L is the maximum length of a word in $\sigma(P_1)$.

- The statement in the paper is even more technical than this!

The technical statement: version 2

- Here is a version adapted to the modified basic strategy.

Lemma (Averaging Lemma 2)

Let $\mathscr{A} = (Q, A)$ be an n-state synchronizing automaton. Suppose there exist probabilities P_1, P_2 on A^* and $R \subsetneq Q$ such that:

1. *P_2 has support $A^{\leq n-1}$;*
2. *$[R]P_2P_1 = [R]$;*
3. *$R \subseteq qA^*$ for all $q \in R$;*
4. *there exists $w_0 \in A^*$ with $Qw_0 \subseteq R$.*

Then $\mathscr{A}$ has a reset word of length at most

$$(|R| - 1)(n - 1 + L) + |w_0|$$

where L is the maximum length of a word in $\sigma(P_1)$.

- The statement in the paper is even more technical than this!

The technical statement: version 2

- Here is a version adapted to the modified basic strategy.

Lemma (Averaging Lemma 2)

Let $\mathscr{A} = (Q, A)$ be an n-state synchronizing automaton. Suppose there exist probabilities P_1, P_2 on A^* and $R \subsetneq Q$ such that:

1. *P_2 has support $A^{\leq n-1}$;*
2. *$[R]P_2P_1 = [R]$;*
3. *$R \subseteq qA^*$ for all $q \in R$;*
4. *there exists $w_0 \in A^*$ with $Qw_0 \subseteq R$.*

Then $\mathscr{A}$ has a reset word of length at most

$$(|R| - 1)(n - 1 + L) + |w_0|$$

where L is the maximum length of a word in $\sigma(P_1)$.

- The statement in the paper is even more technical than this!

The technical statement: version 2

- Here is a version adapted to the modified basic strategy.

Lemma (Averaging Lemma 2)

Let $\mathscr{A} = (Q, A)$ be an n-state synchronizing automaton. Suppose there exist probabilities P_1, P_2 on A^* and $R \subsetneq Q$ such that:

1. P_2 *has support* $A^{\leq n-1}$;
2. $[R]P_2P_1 = [R]$;
3. $R \subseteq qA^*$ *for all* $q \in R$;
4. *there exists* $w_0 \in A^*$ *with* $Qw_0 \subseteq R$.

Then $\mathscr{A}$ has a reset word of length at most

$$(|R| - 1)(n - 1 + L) + |w_0|$$

where L is the maximum length of a word in $\sigma(P_1)$.

- The statement in the paper is even more technical than this!

Some examples

- The next result simultaneously generalizes results of the following authors.
- Rystsov on regular automata.
- Carpi and d'Alessandro on strongly and locally strongly transitive automata (which is the case $k = 1$ below).
- Béal and Perrin on one-cluster automata (which is a special case of locally strongly transitive automata).

Some examples

- The next result simultaneously generalizes results of the following authors.
- Rystsov on regular automata.
- Carpi and d'Alessandro on strongly and locally strongly transitive automata (which is the case $k = 1$ below).
- Béal and Perrin on one-cluster automata (which is a special case of locally strongly transitive automata).

Some examples

- The next result simultaneously generalizes results of the following authors.
- Rystsov on regular automata.
- Carpi and d'Alessandro on strongly and locally strongly transitive automata (which is the case $k = 1$ below).
- Béal and Perrin on one-cluster automata (which is a special case of locally strongly transitive automata).

Some examples

- The next result simultaneously generalizes results of the following authors.
- Rystsov on regular automata.
- Carpi and d'Alessandro on strongly and locally strongly transitive automata (which is the case $k = 1$ below).
- Béal and Perrin on one-cluster automata (which is a special case of locally strongly transitive automata).

A generalization of regular automata

- Let $\mathscr{A} = (Q, A)$ be a synchronizing automaton.
- Suppose there is a set of words $W \subseteq A^*$ and $k \geq 1$ so that, for each state $q \in Q$ and each state $s \in C = QW$, there are exactly k elements of W taking q to s.
- Let ℓ be the length of the shortest word in W and L be the length of the longest.

Theorem

If $C = Q$, then there is a reset word for $\mathscr{A}$ of length at most $2 + (n-2)(n-2+L)$; if $C \subsetneq Q$, then there is a reset word of length at most $(|C| - 1)(n - 2 + L) + \ell + 1$.

A generalization of regular automata

- Let $\mathscr{A} = (Q, A)$ be a synchronizing automaton.
- Suppose there is a set of words $W \subseteq A^*$ and $k \geq 1$ so that, for each state $q \in Q$ and each state $s \in C = QW$, there are exactly k elements of W taking q to s.
- Let ℓ be the length of the shortest word in W and L be the length of the longest.

Theorem

If $C = Q$, then there is a reset word for $\mathscr{A}$ of length at most $2 + (n-2)(n-2+L)$; if $C \subsetneq Q$, then there is a reset word of length at most $(|C|-1)(n-2+L) + \ell + 1$.

A generalization of regular automata

- Let $\mathscr{A} = (Q, A)$ be a synchronizing automaton.
- Suppose there is a set of words $W \subseteq A^*$ and $k \geq 1$ so that, for each state $q \in Q$ and each state $s \in C = QW$, there are exactly k elements of W taking q to s.
- Let ℓ be the length of the shortest word in W and L be the length of the longest.

Theorem

If $C = Q$, then there is a reset word for $\mathscr{A}$ of length at most $2 + (n-2)(n-2+L)$; if $C \subsetneq Q$, then there is a reset word of length at most $(|C|-1)(n-2+L) + \ell + 1$.

A generalization of regular automata

- Let $\mathscr{A} = (Q, A)$ be a synchronizing automaton.
- Suppose there is a set of words $W \subseteq A^*$ and $k \geq 1$ so that, for each state $q \in Q$ and each state $s \in C = QW$, there are exactly k elements of W taking q to s.
- Let ℓ be the length of the shortest word in W and L be the length of the longest.

Theorem

If $C = Q$, then there is a reset word for $\mathscr{A}$ of length at most $2 + (n-2)(n-2+L)$; if $C \subsetneq Q$, then there is a reset word of length at most $(|C|-1)(n-2+L) + \ell + 1$.

Some examples continued

- Our theorem yields better bounds than those in the original papers mentioned above.
- However, improved bounds have been found since.
- In the one-cluster case, we have found a way to deal with the problem of expanding the first state.
- The new idea is that instead of fixing a subset and averaging over a set of words, we average over all choices of the first state.
- This yields an upper bound of $2n^2 - 9n + 14$ on the length of a reset word for n-state one-cluster automata, which is the current record.
- It also leads to the following theorem.

Some examples continued

- Our theorem yields better bounds than those in the original papers mentioned above.
- However, improved bounds have been found since.
- In the one-cluster case, we have found a way to deal with the problem of expanding the first state.
- The new idea is that instead of fixing a subset and averaging over a set of words, we average over all choices of the first state.
- This yields an upper bound of $2n^2 - 9n + 14$ on the length of a reset word for n-state one-cluster automata, which is the current record.
- It also leads to the following theorem.

Some examples continued

- Our theorem yields better bounds than those in the original papers mentioned above.
- However, improved bounds have been found since.
- In the one-cluster case, we have found a way to deal with the problem of expanding the first state.
- The new idea is that instead of fixing a subset and averaging over a set of words, we average over all choices of the first state.
- This yields an upper bound of $2n^2 - 9n + 14$ on the length of a reset word for n-state one-cluster automata, which is the current record.
- It also leads to the following theorem.

Some examples continued

- Our theorem yields better bounds than those in the original papers mentioned above.
- However, improved bounds have been found since.
- In the one-cluster case, we have found a way to deal with the problem of expanding the first state.
- The new idea is that instead of fixing a subset and averaging over a set of words, we average over all choices of the first state.
- This yields an upper bound of $2n^2 - 9n + 14$ on the length of a reset word for n-state one-cluster automata, which is the current record.
- It also leads to the following theorem.

Some examples continued

- Our theorem yields better bounds than those in the original papers mentioned above.
- However, improved bounds have been found since.
- In the one-cluster case, we have found a way to deal with the problem of expanding the first state.
- The new idea is that instead of fixing a subset and averaging over a set of words, we average over all choices of the first state.
- This yields an upper bound of $2n^2 - 9n + 14$ on the length of a reset word for n-state one-cluster automata, which is the current record.
- It also leads to the following theorem.

Some examples continued

- Our theorem yields better bounds than those in the original papers mentioned above.
- However, improved bounds have been found since.
- In the one-cluster case, we have found a way to deal with the problem of expanding the first state.
- The new idea is that instead of fixing a subset and averaging over a set of words, we average over all choices of the first state.
- This yields an upper bound of $2n^2 - 9n + 14$ on the length of a reset word for n-state one-cluster automata, which is the current record.
- It also leads to the following theorem.

One-cluster automata with prime length cycle

Theorem

The Černý conjecture is true for one-cluster automata with prime length cycle.

- This has the following consequence for the hybrid Černý-Road Coloring conjecture:

Corollary

Let Γ be a strongly connected aperiodic digraph with constant out-degree, n vertices and no multiple edges. Suppose moreover that Γ contains a cycle of prime length $p < n$. Then Γ admits a synchronizing coloring with a reset word of length at most $3n - 3p + 1 + (p-2)(2n-p) \leq (n-1)^2$.

One-cluster automata with prime length cycle

Theorem

The Černý conjecture is true for one-cluster automata with prime length cycle.

- This has the following consequence for the hybrid Černý-Road Coloring conjecture:

Corollary

Let Γ be a strongly connected aperiodic digraph with constant out-degree, n vertices and no multiple edges. Suppose moreover that Γ contains a cycle of prime length $p < n$. Then Γ admits a synchronizing coloring with a reset word of length at most $3n - 3p + 1 + (p-2)(2n-p) \leq (n-1)^2$.

One-cluster automata with prime length cycle

Theorem

The Černý conjecture is true for one-cluster automata with prime length cycle.

- This has the following consequence for the hybrid Černý-Road Coloring conjecture:

Corollary

Let Γ be a strongly connected aperiodic digraph with constant out-degree, n vertices and no multiple edges. Suppose moreover that Γ contains a cycle of prime length $p < n$. Then Γ admits a synchronizing coloring with a reset word of length at most $3n - 3p + 1 + (p - 2)(2n - p) \leq (n - 1)^2$.

Thank you for your attention!