Complexity of a Problem Concerning Reset Words for Eulerian Binary Automata

Vojtěch Vorel

Charles University
Prague, Czech Republic

LATA 2014

Outline

Outline

1 General Introduction
- Finite Automata and Synchronization
- Computational Tasks

2 NP-Completeness of SYN

3 Present Result

Finite Automata

- *Automaton* is a triple $A = (Q, X, \delta)$
  - Q ... finite set of *states*
  - X ... finite set of *letters* (the *alphabet*)
  - δ ... total function $Q \times X \to Q$ (*transition function*)
- Extended transition function:

$$\delta : 2^Q \times X^\star \to 2^Q$$

Finite Automata

- *Automaton* is a triple $A = (Q, X, \delta)$
  - Q ... finite set of *states*
  - X ... finite set of *letters* (the *alphabet*)
  - δ ... total function $Q \times X \to Q$ (*transition function*)
- Extended transition function:

$$\delta : 2^Q \times X^\star \to 2^Q$$

Reset Words

- $w \in X^{\star}$ is a *reset word* of A if

$$|\delta(Q, w)| = 1,$$

i.e. if

$$(\forall q \in Q)\, q \xrightarrow{w} r$$

for some $r \in Q$.

- if A has some reset word, we call it *synchronizing*

Reset Words

- $w \in X^{\star}$ is a *reset word* of A if

  $$|\delta(Q, w)| = 1,$$

  i.e. if

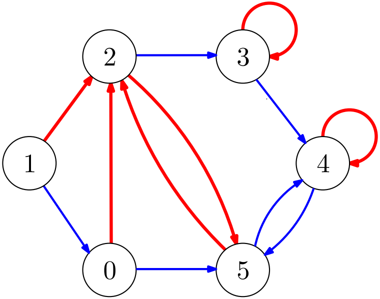
  $$(\forall q \in Q)\, q \xrightarrow{w} r$$

  for some $r \in Q$.

- if A has some reset word, we call it *synchronizing*

Reset Words: an Example

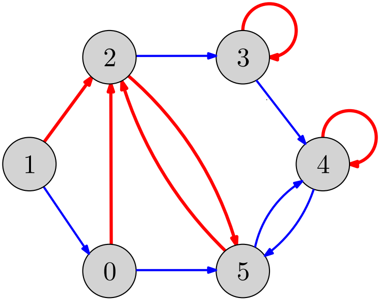
$Q = \{0, 1, 2, 3, 4, 5\}$
$X = \{a, b\}$



Reset Words: an Example

$Q = \{0, 1, 2, 3, 4, 5\}$
$X = \{a, b\}$

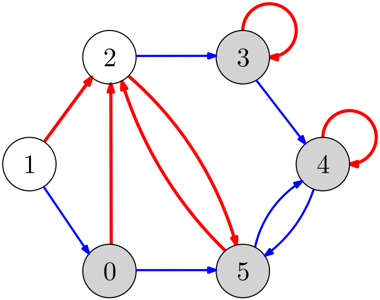
$w =$



Reset Words: an Example

$Q = \{0, 1, 2, 3, 4, 5\}$
$X = \{a, b\}$

$w = b$

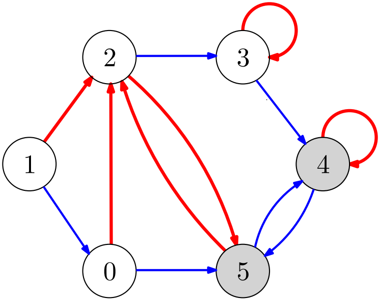


$$\left| \boxed{\delta(Q, w)} \right| = 4$$

Reset Words: an Example

$Q = \{0, 1, 2, 3, 4, 5\}$
$X = \{a, b\}$

$w = bb$

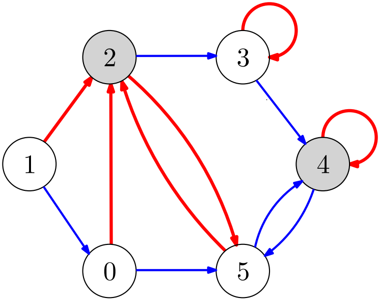


$$\left| \delta(Q, w) \right| = 2$$

Reset Words: an Example

$Q = \{0, 1, 2, 3, 4, 5\}$
$X = \{a, b\}$

$w = bba$

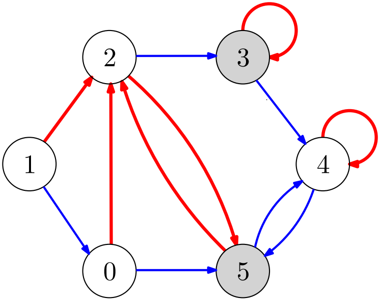


$$\left| \boxed{\delta(Q, w)} \right| = 2$$

Reset Words: an Example

$Q = \{0, 1, 2, 3, 4, 5\}$
$X = \{a, b\}$

$w = bbab$

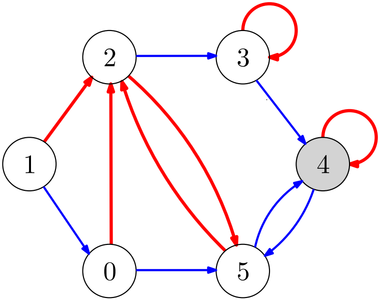


$$\left| \boxed{\delta(Q, w)} \right| = 2$$

Reset Words: an Example

$Q = \{0, 1, 2, 3, 4, 5\}$
$X = \{a, b\}$

$w = bbabb$
is a reset word



$$\left| \boxed{\delta(Q, w)} \right| = 1$$

Short Reset Words

- **Černý conjecture:**
  - If A is synchronizing, it has a reset word of length at most $(|Q|-1)^2$
- Known bounds:
  - $\frac{1}{3}|Q|^3 - n^2 + \frac{5}{3}n - 1$ (Kohavi, 1970)
  - $\frac{1}{6}|Q|^3 - \frac{1}{6}n$ (Pin, 1983)
  - $\cancel{\frac{7}{48}|Q|^3 + \frac{1}{8}n^2 - \frac{1}{3}n}$ (Trakhtman, 2011)

Short Reset Words

- **Černý conjecture:**
  - If A is synchronizing, it has a reset word of length at most $(|Q|-1)^2$
- Known bounds:
  - $\frac{1}{3}|Q|^3 - n^2 + \frac{5}{3}n - 1$ (Kohavi, 1970)
  - $\frac{1}{6}|Q|^3 - \frac{1}{6}n$ (Pin, 1983)
  - $\cancel{\frac{7}{48}|Q|^3 + \frac{1}{8}n^2 - \frac{1}{3}n}$ (Trakhtman, 2011)

Outline

1. General Introduction
   - Finite Automata and Synchronization
   - Computational Tasks
2. NP-Completeness of SYN
3. Present Result

Computational tasks

EX-SYN: Given an automaton A, decide if it has **any** reset word.

- *Solvable in polynomial time*

MIN-SYN: Given an automaton A and a number d, decide if d is the length of **shortest** reset words of A.

- *Both NP-hard and coNP-hard*

SYN: Given an automaton A and a number d, decide if A has a reset word **of length at most** d.

- *NP-complete*

Computational tasks

EX-SYN: Given an automaton A, decide if it has **any** reset word.

- *Solvable in polynomial time*

MIN-SYN: Given an automaton A and a number d, decide if d is the length of **shortest** reset words of A.

- *Both NP-hard and coNP-hard*

SYN: Given an automaton A and a number d, decide if A has a reset word **of length at most** d.

- *NP-complete*

Computational tasks

EX-SYN: Given an automaton A, decide if it has **any** reset word.

- *Solvable in polynomial time*

MIN-SYN: Given an automaton A and a number d, decide if d is the length of **shortest** reset words of A.

- *Both NP-hard and coNP-hard*

SYN: Given an automaton A and a number d, decide if A has a reset word **of length at most** d.

- *NP-complete*

Computational tasks

EX-SYN: Given an automaton A, decide if it has **any** reset word.

- *Solvable in polynomial time*

MIN-SYN: Given an automaton A and a number d, decide if d is the length of **shortest** reset words of A.

- *Both NP-hard and coNP-hard*

SYN: Given an automaton A and a number d, decide if A has a reset word **of length at most** d.

- *NP-complete*

Computational tasks

EX-SYN: Given an automaton A, decide if it has **any** reset word.

- *Solvable in polynomial time*

MIN-SYN: Given an automaton A and a number d, decide if d is the length of **shortest** reset words of A.

- *Both NP-hard and coNP-hard*

SYN: Given an automaton A and a number d, decide if A has a reset word **of length at most** d.

- *NP-complete*

Computational tasks

EX-SYN: Given an automaton A, decide if it has **any** reset word.

- *Solvable in polynomial time*

MIN-SYN: Given an automaton A and a number d, decide if d is the length of **shortest** reset words of A.

- *Both NP-hard and coNP-hard*

SYN: Given an automaton A and a number d, decide if A has a reset word **of length at most** d.

- *NP-complete*

Computational tasks

EX-SYN: Given an automaton A, decide if it has **any** reset word.

- *Solvable in polynomial time*

MIN-SYN: Given an automaton A and a number d, decide if d is the length of **shortest** reset words of A.

- *Both NP-hard and coNP-hard*

SYN: Given an automaton A and a number d, decide if A has a reset word **of length at most** d.

- *NP-complete*

Reductions from SAT

$$\text{Propositional formula } \varphi \text{ in CNF}$$
$$\downarrow$$
$$\text{Automaton } A \text{ and number } d$$

such that

$$\varphi \text{ is satisfiable}$$
$$\Updownarrow$$
$$A \text{ has a reset word of legth at most } d$$

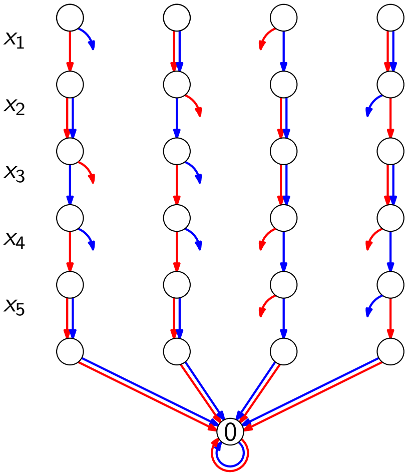
$$\varphi = (x_1 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_2} \vee x_3 \vee x_4) \wedge (\overline{x_1} \vee \overline{x_4} \vee \overline{x_5}) \wedge (x_2 \vee \overline{x_4} \vee x_5)$$

$$\downarrow$$

$X = \{1, 0\}$

$d = 5$

Is there a reset word of length 5?



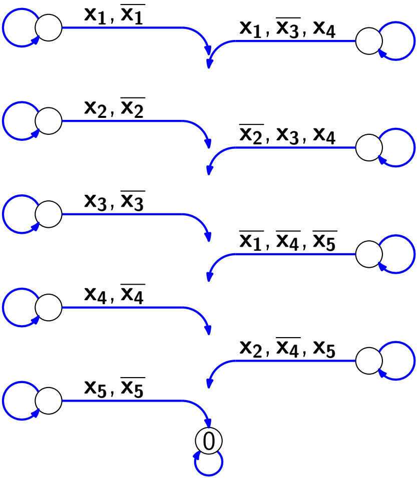
$$\varphi = (x_1 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_2} \vee x_3 \vee x_4) \wedge (\overline{x_1} \vee \overline{x_4} \vee \overline{x_5}) \wedge (x_2 \vee \overline{x_4} \vee x_5)$$

$$\downarrow$$

$X = \{\mathbf{x_1}, \ldots, \mathbf{x_n}, \overline{\mathbf{x_1}}, \ldots, \overline{\mathbf{x_n}}\}$

$d = 5$

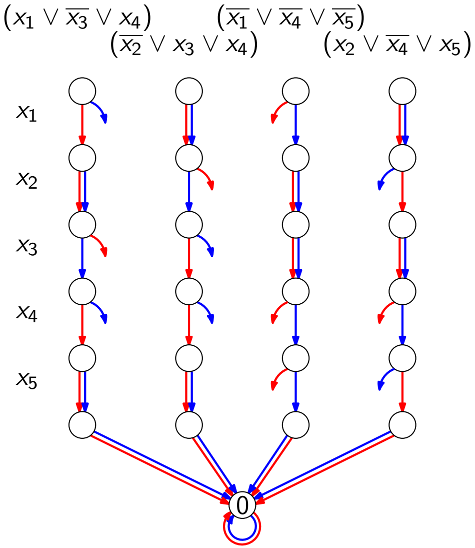
Is there a reset word of length 5?



An automaton $A = (Q, X, \delta)$ is ***Eulerian*** if each state has $|X|$ incoming transitions.

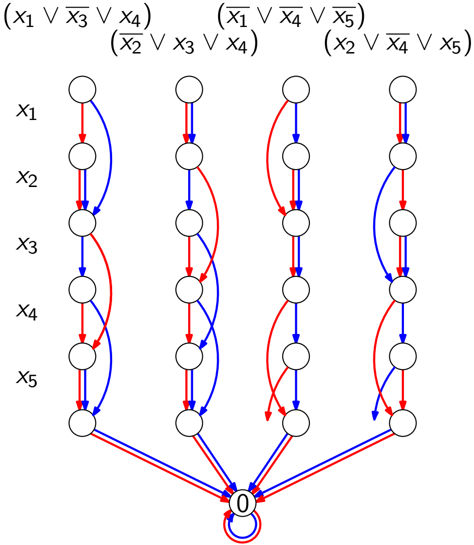
$X = \{1, 0\}$

d=5


$(x_1 \vee \overline{x_3} \vee x_4)$
$(\overline{x_2} \vee x_3 \vee x_4)$
$(\overline{x_1} \vee \overline{x_4} \vee \overline{x_5})$
$(x_2 \vee \overline{x_4} \vee x_5)$
x_1
x_2
x_3
x_4
x_5
0

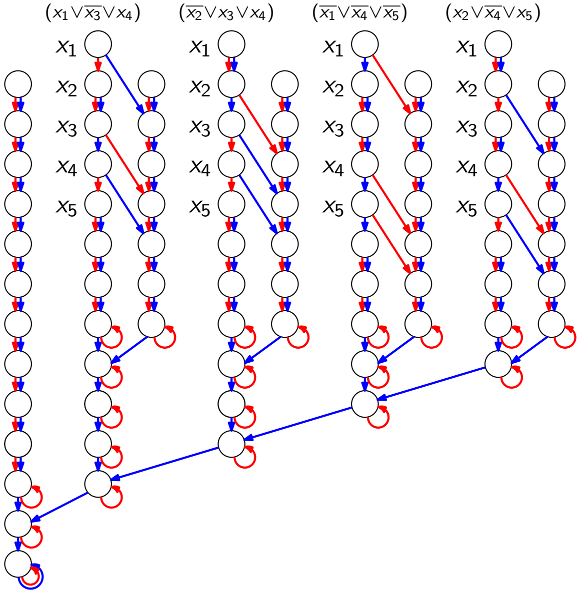
$X = \{1, 0\}$

d=5


$(x_1 \vee \overline{x_3} \vee x_4)$
$(\overline{x_2} \vee x_3 \vee x_4)$
$(\overline{x_1} \vee \overline{x_4} \vee \overline{x_5})$
$(x_2 \vee \overline{x_4} \vee x_5)$
x_1
x_2
x_3
x_4
x_5
0

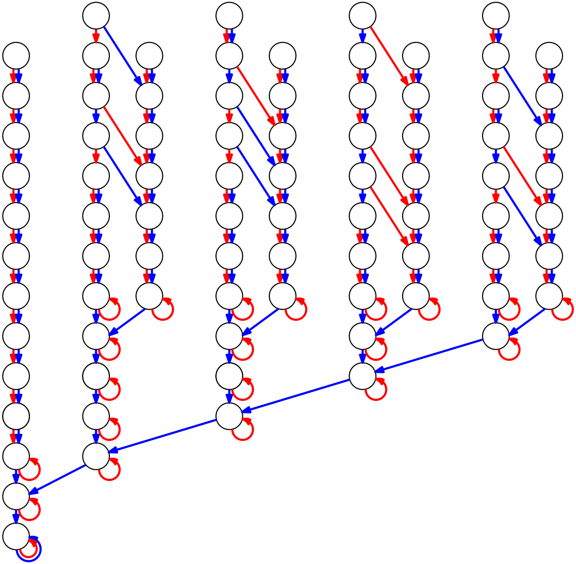
$X = \{1, 0\}$

d=12


$(x_1 \vee \overline{x_3} \vee x_4)$
$(\overline{x_2} \vee x_3 \vee x_4)$
$(\overline{x_1} \vee \overline{x_4} \vee \overline{x_5})$
$(x_2 \vee \overline{x_4} \vee x_5)$
x_1
x_2
x_3
x_4
x_5

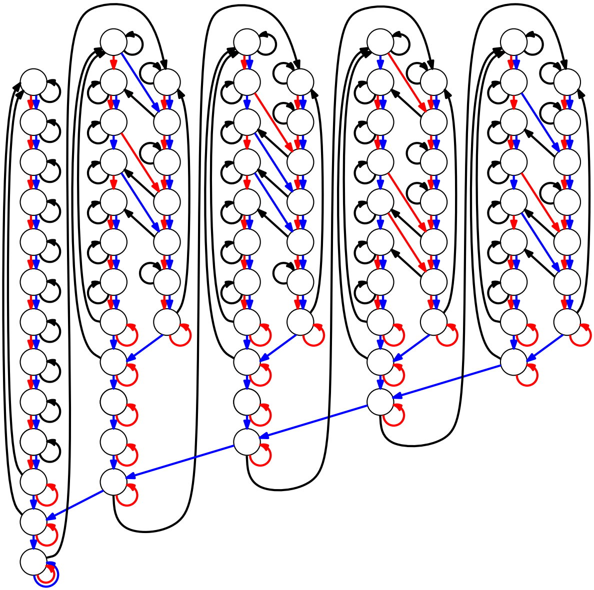
$X = \{1, 0\}$

d=12



$X = \{1, 0, c\}$

d=12



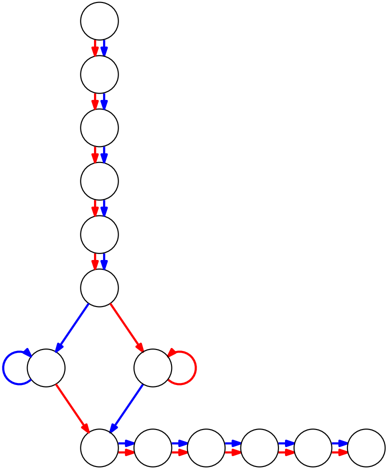
Present Result

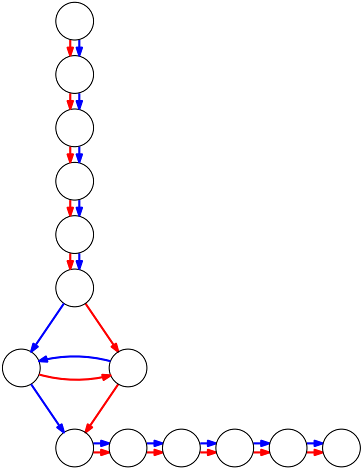
SYN is NP-complete even if restricted to Eulerian automata with a two-letter alphabet.

Proof Ideas

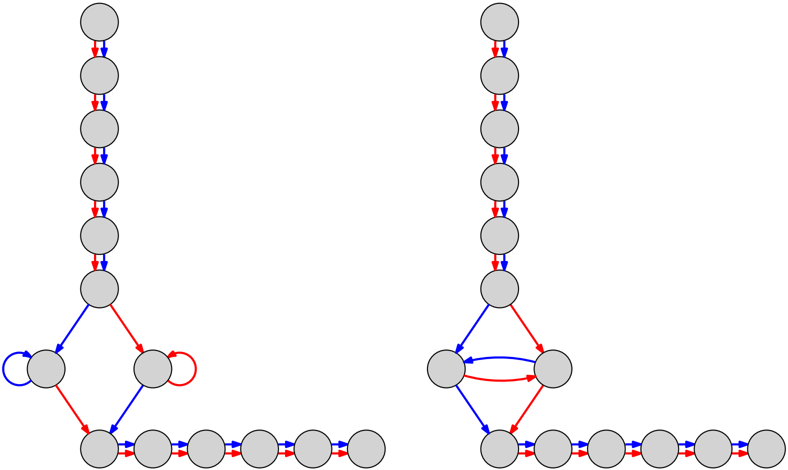
- φ in 3-CNF
- *Recording*
- *Testing*

Recording

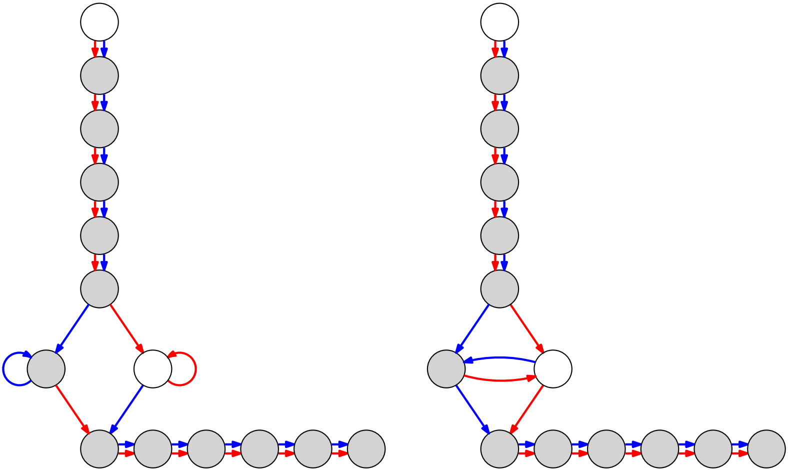




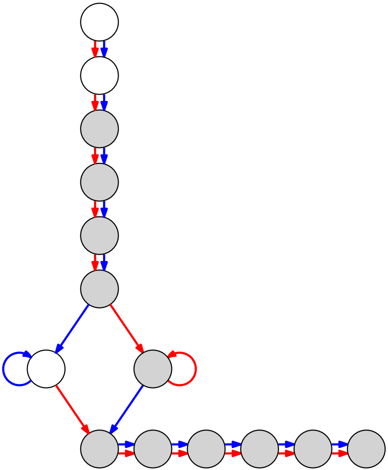
Recording



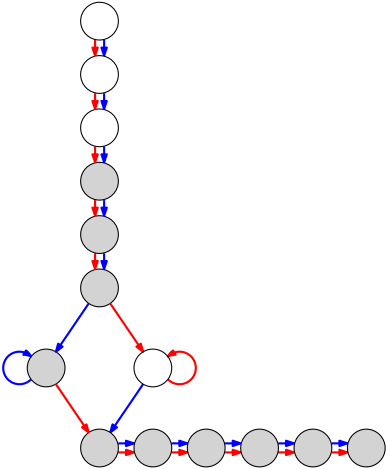
$w = a$



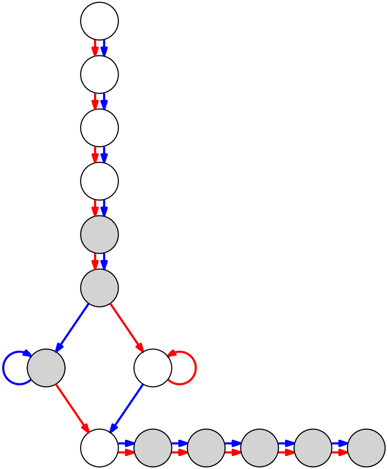
$w = ab$

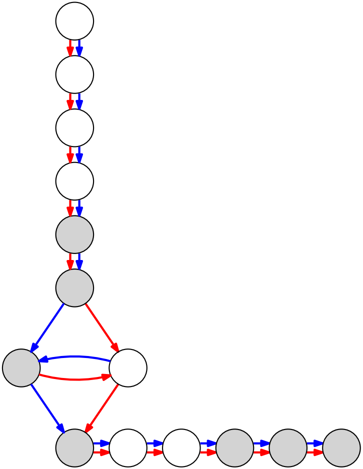


$w = aba$

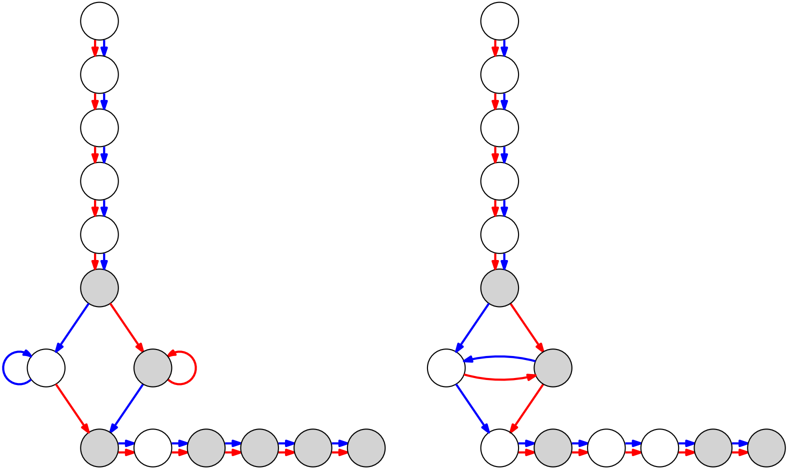


$w = abaa$

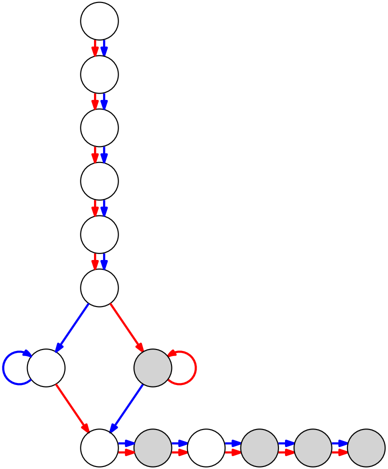


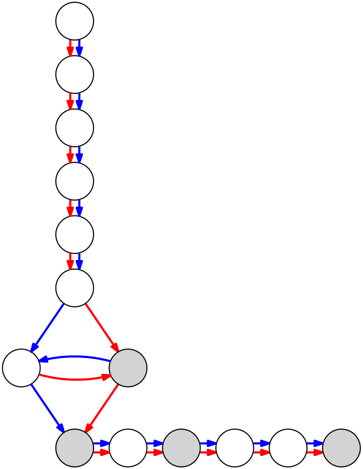


$w = abaab$

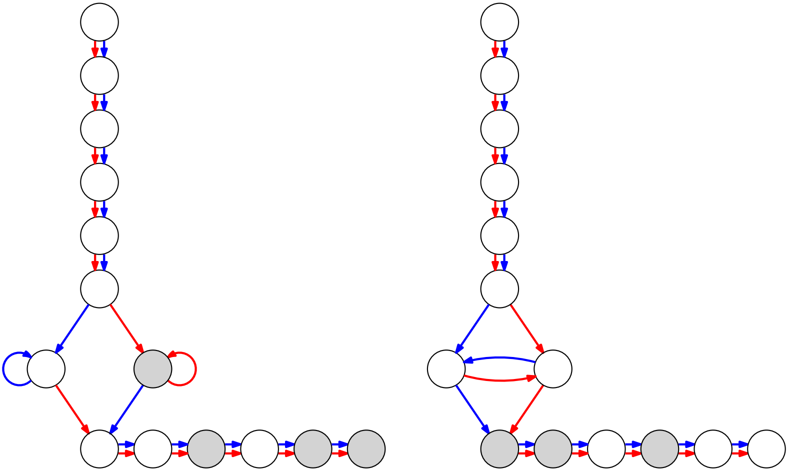


$w = abaabb$

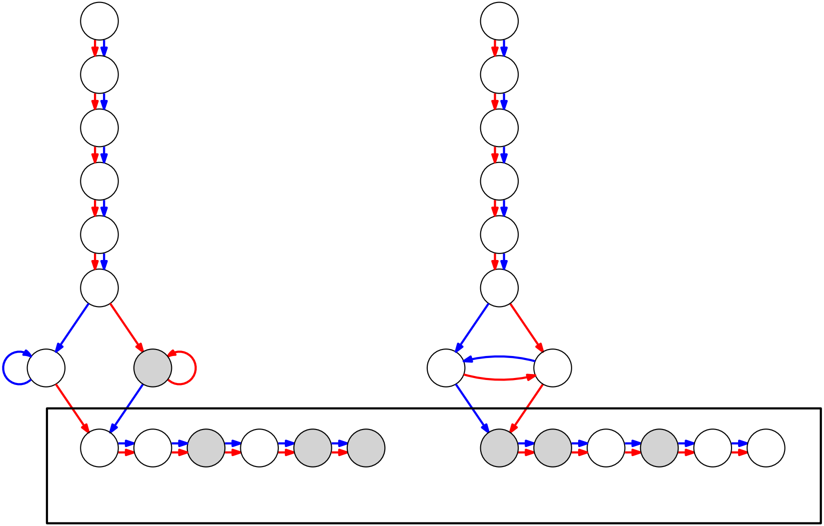




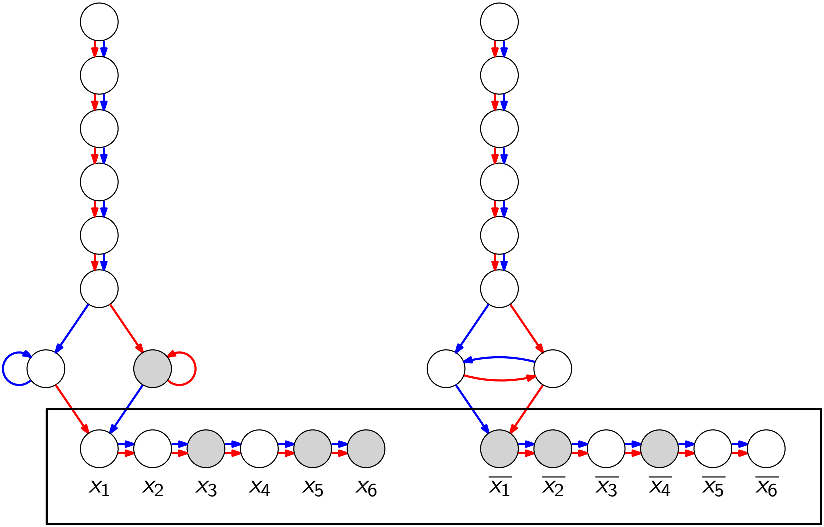
$w = abaabbb$

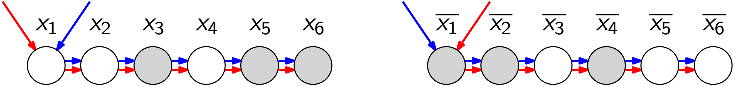


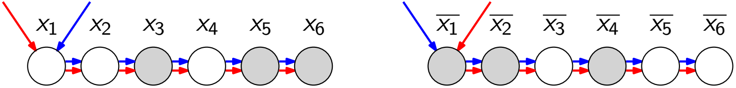
$w = abaabbb$



$w = abaabbb$

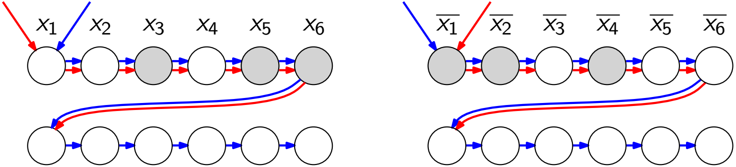

x_1 x_2 x_3 x_4 x_5 x_6
$\overline{x_1}$ $\overline{x_2}$ $\overline{x_3}$ $\overline{x_4}$ $\overline{x_5}$ $\overline{x_6}$


x_1
x_2
x_3
x_4
x_5
x_6
$\overline{x_1}$
$\overline{x_2}$
$\overline{x_3}$
$\overline{x_4}$
$\overline{x_5}$
$\overline{x_6}$



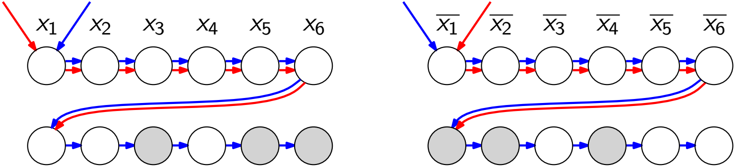
word w:

- recording part (fix an assignment)
- transpose part a^k
- test part (prove that the assignment works)
- ...



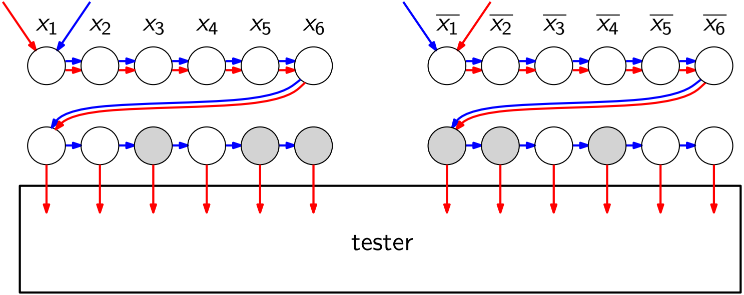
word w:

- recording part (fix an assignment)
- transpose part a^k
- test part (prove that the assignment works)
- ...



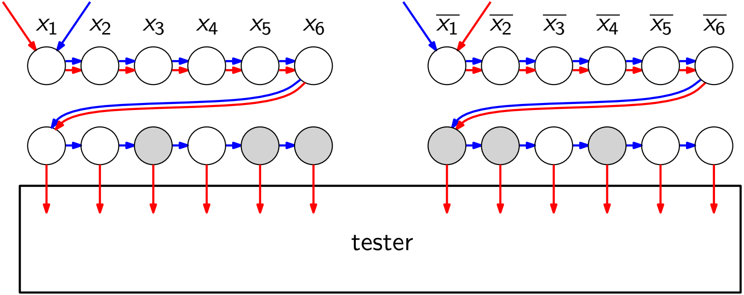
word w:

- recording part (fix an assignment)
- transpose part a^k
- test part (prove that the assignment works)
- ...



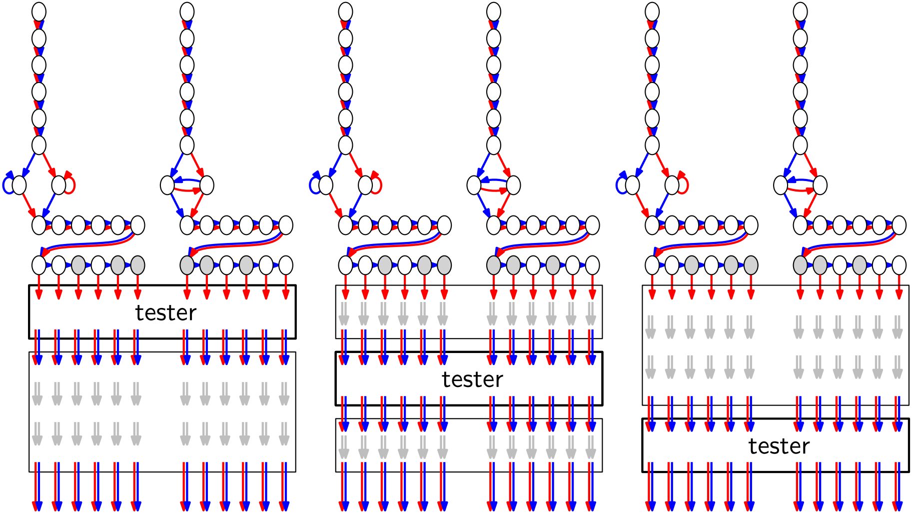
word w:

- recording part (fix an assignment)
- transpose part a^k
- test part (prove that the assignment works)
- ...

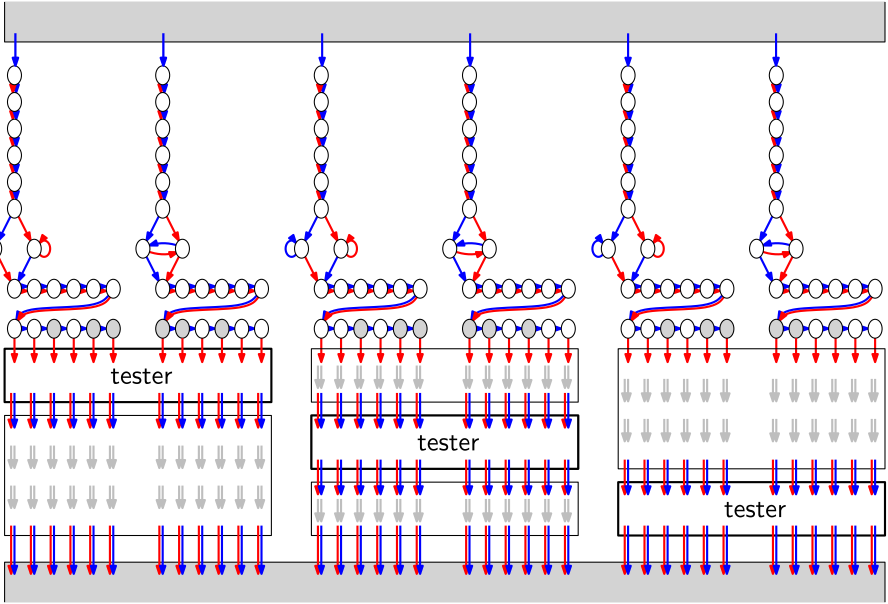


word w:

- recording part (fix an assignment)
- transpose part a^k
- test part (prove that the assignment works)
- ...


tester
tester
tester

Testing



Summary

- A *reset word* takes all the states to some unique state.
- *SYN* - Does A have a reset word of length at most d?

- SYN is NP-complete (Eppstein, 1990).

- Does it remain NP-hard if restricted to Eulerian automata with a constant-size alphabet?
  - For a 3-letter alphabet it does (Martyugin, 2011).
  - For a 2-letter alphabet it does as well.