

Incompleteness and synchronization in finite automata and codes

Arturo Carpi

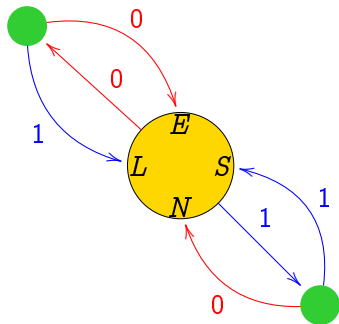
Dipartimento di Matematica e Informatica
Università di Perugia

Colloque en l'honneur de
Marcel-Paul Schützenberger

Bordeaux, March 21–25, 2016

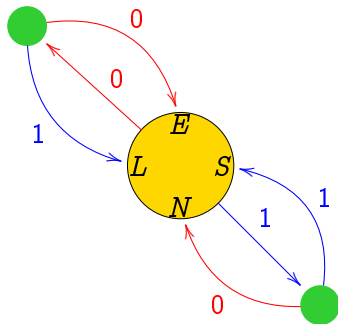
A decoder

<i>E</i>	00
<i>L</i>	01
<i>N</i>	10
<i>S</i>	11



A decoder

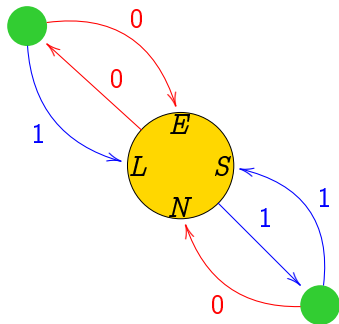
<i>E</i>	00
<i>L</i>	01
<i>N</i>	10
<i>S</i>	11



SENSELESS \rightarrow ~~1~~10010110001001111

A decoder

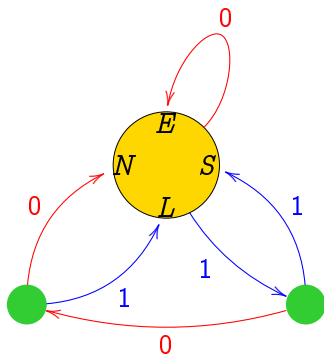
<i>E</i>	00
<i>L</i>	01
<i>N</i>	10
<i>S</i>	11



SENSELESS \rightarrow ~~1~~10010110001001111 \rightarrow NLNLENLS

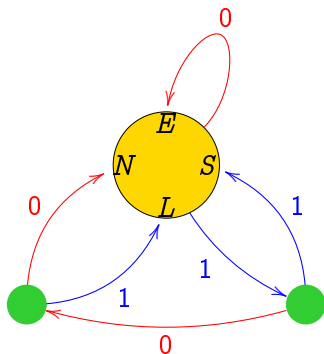
A synchronizing decoder

<i>E</i>	0
<i>L</i>	101
<i>N</i>	100
<i>S</i>	11



A synchronizing decoder

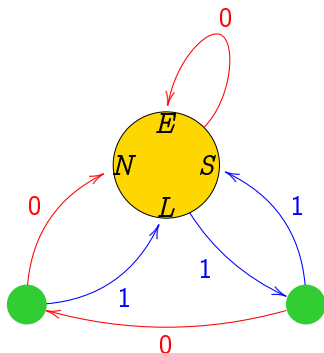
<i>E</i>	0
<i>L</i>	101
<i>N</i>	100
<i>S</i>	11



SENSELESS \rightarrow ~~1~~1010011010101111

A synchronizing decoder

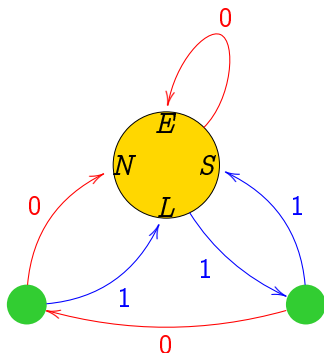
<i>E</i>	0
<i>L</i>	101
<i>N</i>	100
<i>S</i>	11



SENSELESS \rightarrow ~~1~~1010011010101111 \rightarrow LEESELESS

A synchronizing decoder

<i>E</i>	0
<i>L</i>	101
<i>N</i>	100
<i>S</i>	11



SENSELESS \rightarrow ~~1~~1010011010101111 \rightarrow LEESELESS

00 "resets"

Synchronization problem

Definition

A word w is a **reset word** of an automaton \mathcal{A} if there is a **reset state** r such that for all state q , $qw = r$.

An automaton is **synchronizing** if it has a reset word.

Synchronization problem

Definition

A word w is a **reset word** of an automaton \mathcal{A} if there is a **reset state** r such that for all state q , $qw = r$.

An automaton is **synchronizing** if it has a reset word.

Problem

Given a synchronizing automaton, what is (the length of) the shortest reset word ?

Synchronization problem

Definition

A word w is a **reset word** of an automaton \mathcal{A} if there is a **reset state** r such that for all state q , $qw = r$.

An automaton is **synchronizing** if it has a reset word.

Problem

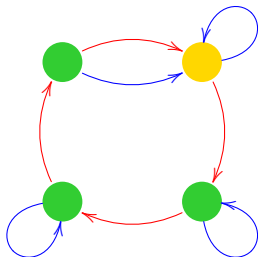
Given a synchronizing automaton, what is (the length of) the shortest reset word ?

NP-complete (Eppstein, 1990)

 Search for bounds w.r.t. the number of states

Theorem (Černý, 1964)

For all $n \geq 1$, there exists an n -state automaton whose shortest reset word has length $(n - 1)^2$



Shortest reset word:

baaabbaaab

Theorem (Černý, 1964)

For all $n \geq 1$, there exists an n -state automaton whose shortest reset word has length $(n - 1)^2$

Conjecture (Černý, 1964)

Any synchronizing n -state automaton has a reset word of length $(n - 1)^2$

Theorem (Černý, 1964)

For all $n \geq 1$, there exists an n -state automaton whose shortest reset word has length $(n - 1)^2$

Conjecture (Černý, 1964)

Any synchronizing n -state automaton has a reset word of length $(n - 1)^2$

Theorem (Pin, 1978; Frankl, 1982)

Any n -state synchronizing automaton has a reset word of length

$$\frac{n^3 - n}{6}$$

Černý conjecture

Černý conjecture is true for

Černý conjecture

Černý conjecture is true for

 Monotonic automata [Eppstein, 1990]

Černý conjecture

Černý conjecture is true for

- Monotonic automata [Eppstein, 1990]
- Circular automata [Dubuc, 1998]

Černý conjecture is true for

- Monotonic automata [Eppstein, 1990]
- Circular automata [Dubuc, 1998]
- Eulerian automata [Kari, 2003]

Černý conjecture is true for

- Monotonic automata [Eppstein, 1990]
- Circular automata [Dubuc, 1998]
- Eulerian automata [Kari, 2003]
 - unambiguous Eulerian automata [C.-D'A., 2009]

Černý conjecture is true for

- Monotonic automata [Eppstein, 1990]
- Circular automata [Dubuc, 1998]
- Eulerian automata [Kari, 2003]
 - unambiguous Eulerian automata [C.-D'A., 2009]

A quadratic bound has been found also for

- 1-cluster automata [Béal, Perrin, 2009]
(synchronizing decoders)

Incomplete automata

Definition

A word w is **incompletable** in an automaton \mathcal{A} if for all state q , $qw = \emptyset$.

An automaton is **incomplete** if it has an incompletable word.

Incomplete automata

Definition

A word w is **incompletable** in an automaton \mathcal{A} if for all state q , $qw = \emptyset$.

An automaton is **incomplete** if it has an incompletable word.

Problem

Given an incomplete automaton, what is (the length of) the shortest reset word ?

Incomplete automata

Definition

A word w is **incompletable** in an automaton \mathcal{A} if for all state q , $qw = \emptyset$.

An automaton is **incomplete** if it has an incompletable word.

Problem

Given an incomplete automaton, what is (the length of) the shortest reset word ?

Proposition

Any incomplete deterministic n -state automaton has an incompletable word of length $n(n - 1)/2$

Beyond deterministic automata

- For all n , there are **non deterministic** n -state automata such that the minimal length of a reset (resp. incompletable) word is $O(\sqrt{n}!)$ [Goralčik, Hedrlin, Koubek, Ryšlinková, 1982]

Beyond deterministic automata

- For all n , there are **non deterministic** n -state automata such that the minimal length of a reset (resp. incompletable) word is $O(\sqrt{n}!)$ [Goralčík, Hedrlín, Koubek, Ryšlinková, 1982]
- Any complete transitive synchronizing **unambiguous** n -state automaton has a reset word of length $n^3/2$ [C., 1988]

Beyond deterministic automata

- For all n , there are **non deterministic** n -state automata such that the minimal length of a reset (resp. incompletable) word is $O(\sqrt{n}!)$ [Goralčik, Hedrlin, Koubek, Ryšlinková, 1982]
- Any complete transitive synchronizing **unambiguous** n -state automaton has a reset word of length $n^3/2$ [C., 1988]
- It is undecidable whether a **stochastic** automaton has a reset word

Beyond deterministic automata

- For all n , there are **non deterministic** n -state automata such that the minimal length of a reset (resp. incompletable) word is $O(\sqrt{n}!)$ [Goralčík, Hedrlín, Koubek, Ryšlinková, 1982]
- Any complete transitive synchronizing **unambiguous** n -state automaton has a reset word of length $n^3/2$ [C., 1988]
- It is undecidable whether a **stochastic** automaton has a reset word
- It is undecidable whether a \mathbb{Z} -automaton has an incompletable word

Some problem

- Is the bound $n(n - 1)/2$ for incomplete deterministic n -state automata optimal?

Some problem

- Is the bound $n(n - 1)/2$ for incomplete deterministic n -state automata optimal?
- Is there a polynomial upper bound for the minimal length of an incompletable word in an unambiguous n -state automaton?

Some problem

- Is the bound $n(n - 1)/2$ for incomplete deterministic n -state automata optimal?
- Is there a polynomial upper bound for the minimal length of an incompletable word in an unambiguous n -state automaton?
- Is there a polynomial upper bound for the minimal length of a synchronizing word in a synchronizing **intransitive** unambiguous n -state automaton?

Incomplete sets of words

A finite alphabet

Incomplete sets of words

A finite alphabet

X finite set of words on alphabet A

Incomplete sets of words

A finite alphabet

X finite set of words on alphabet A

$\ell(X)$ maximal length of the words of X

Incomplete sets of words

A finite alphabet

X finite set of words on alphabet A

$\ell(X)$ maximal length of the words of X

Definition

Incomplete sets of words

A finite alphabet

X finite set of words on alphabet A

$\ell(X)$ maximal length of the words of X

Definition

A pair of words (r, s) is said to be a **completion** of a word w in X if $rhs \in X^*$.

Incomplete sets of words

A finite alphabet

X finite set of words on alphabet A

$\ell(X)$ maximal length of the words of X

Definition

A pair of words (r, s) is said to be a **completion** of a word w in X if $rws \in X^*$.

A word w is **completable** in X if it has a completion in X , **incompletable** otherwise.

Incomplete sets of words

A finite alphabet

X finite set of words on alphabet A

$\ell(X)$ maximal length of the words of X

Definition

A pair of words (r, s) is said to be a **completion** of a word w in X if $rhs \in X^*$.

A word w is **completable** in X if it has a completion in X , **incompletable** otherwise.

The set X is **complete** if all words of A^* are completable in X , **incomplete** otherwise.

Example

$$X = \{ba, bba\}$$

a bba b

Example

$$X = \{ba, bba\}$$

$$ba\ bba\ bba \in X^*$$

(b, ba) is a completion of $abbab$ in X

$$X = \{ba, bba\}$$

$$ba\ bba\ bba \in X^*$$

(b, ba) is a completion of $abbab$ in X

$Y = \{a, ba, bb\}$ is complete (code)

$$X = \{ba, bba\}$$

$$ba\ bba\ bba \in X^*$$

(b, ba) is a completion of $abbab$ in X

$Y = \{a, ba, bb\}$ is complete (code)

Remark

Any maximal code is a minimal complete set. The converse is not generally true.

Definition

A word $w \in X^*$ is a **synchronizing word** of X if for all completion (r, s) of w in X , one has

$$rw, ws \in X^* .$$

Synchronization

Definition

A word $w \in X^*$ is a **synchronizing word** of X if for all completion (r, s) of w in X , one has

$$rw, ws \in X^* .$$



Synchronization

Definition

A word $w \in X^*$ is a **synchronizing word** of X if for all completion (r, s) of w in X , one has

$$rw, ws \in X^* .$$

The set X is **synchronizing** if it has a synchronizing word.



Synchronization

Definition

A word $w \in X^*$ is a **synchronizing word** of X if for all completion (r, s) of w in X , one has

$$rw, ws \in X^* .$$

The set X is **synchronizing** if it has a synchronizing word.



Example

$X = \{aab, aba, abaa, baa\}$
 $abaa\ aba\ aab$ is synchronizing

A set X is complete if and only if

$$\forall w \in A^* \exists u, v \in A^* \text{ s.t. } u w v \in X^*$$

A code X is complete and synchronizing if and only if

$$\exists u, v \in A^* \text{ s.t. } \forall w \in A^*, u w v \in X^*$$

Completeness, synchronization and automata

In the sequel, by ‘automaton’ we mean a trim automaton with a unique initial and final state, denoted by 1.

Lemma

Let X be a prefix code. Then X^* is accepted by a deterministic automaton \mathcal{A} . Moreover,

①

X is complete if and only if \mathcal{A} is complete.

②

X is synchronizing if and only if \mathcal{A} is synchronizing.

Completeness, synchronization and automata - 2

With a natural definition of complete and synchronizing unambiguous automata,

Lemma

Let X be a code. Then X^* is accepted by an unambiguous automaton \mathcal{A} . Moreover,

- 1 X is complete if and only if \mathcal{A} is complete.
- 2 X is synchronizing if and only if \mathcal{A} is synchronizing.

Restivo's Problem

Find an upper bound to the minimal length of an incompletable word of an incomplete set

Restivo vs. Cerny

Restivo's Problem

Find an upper bound to the minimal length of an incompletable word of an incomplete set

Cerny's Problem

Find an upper bound to the minimal length of a synchronizing word of a synchronizing set

Let \mathcal{L} be a class of finite languages.

Let \mathcal{L} be a class of finite languages.

For all $n, d > 0$, we denote by $R_{\mathcal{L}}(n, d)$ the least positive integer r satisfying the following condition:

any incomplete set $X \in \mathcal{L}$ on a d -letter alphabet such that $\ell(X) \leq n$ has an incompletable word of length r .

Let \mathcal{L} be a class of finite languages.

For all $n, d > 0$, we denote by $R_{\mathcal{L}}(n, d)$ the least positive integer r satisfying the following condition:

any incomplete set $X \in \mathcal{L}$ on a d -letter alphabet such that $\ell(X) \leq n$ has an incompletable word of length r .

and by $C_{\mathcal{L}}(n, d)$ the least positive integer c satisfying the following condition:

any synchronizing set $X \in \mathcal{L}$ on a d -letter alphabet such that $\ell(X) \leq n$ has a synchronizing word of length $\leq c$.

Let \mathcal{L} be a class of finite languages.

For all $n, d > 0$, we denote by $R_{\mathcal{L}}(n, d)$ the least positive integer r satisfying the following condition:

any incomplete set $X \in \mathcal{L}$ on a d -letter alphabet such that $\ell(X) \leq n$ has an incompletable word of length r .

and by $C_{\mathcal{L}}(n, d)$ the least positive integer c satisfying the following condition:

any synchronizing set $X \in \mathcal{L}$ on a d -letter alphabet such that $\ell(X) \leq n$ has a synchronizing word of length $\leq c$.

Moreover, for all $n > 0$ we set

$$R_{\mathcal{L}}(n) = \sup_{d \geq 1} R_{\mathcal{L}}(n, d), \quad C_{\mathcal{L}}(n) = \sup_{d \geq 1} C_{\mathcal{L}}(n, d).$$

Our problems

We are interested in evaluating the functions introduced above in the case where \mathcal{L} is one of the following classes of languages:

 all finite languages

Our problems

We are interested in evaluating the functions introduced above in the case where \mathcal{L} is one of the following classes of languages:

- all finite languages
- finite codes

Our problems

We are interested in evaluating the functions introduced above in the case where \mathcal{L} is one of the following classes of languages:

- all finite languages
- finite codes
- finite prefix codes

Our problems

We are interested in evaluating the functions introduced above in the case where \mathcal{L} is one of the following classes of languages:

- all finite languages
- finite codes
- finite prefix codes
- finite complete codes

Our problems

We are interested in evaluating the functions introduced above in the case where \mathcal{L} is one of the following classes of languages:

- all finite languages
- finite codes
- finite prefix codes
- finite complete codes
- finite complete prefix codes

Proposition (Restivo, 1978)

Let \mathcal{P} be the class of finite prefix codes. Then

$$R_{\mathcal{P}}(n) \leq 2n^2.$$

Proposition (Restivo, 1978)

Let \mathcal{P} be the class of finite prefix codes. Then

$$R_{\mathcal{P}}(n) \leq 2n^2.$$

Example (Pribavkina, 2009)

Let $X = A^n \setminus \{u\}$, with $u \in A^n$ unbordered. An incompletable word of minimal length is $w = (ua)^{n-1}u$, $a \in A$.

Proposition (Restivo, 1978)

Let \mathcal{P} be the class of finite prefix codes. Then

$$R_{\mathcal{P}}(n) \leq 2n^2.$$

Example (Pribavkina, 2009)

Let $X = A^n \setminus \{u\}$, with $u \in A^n$ unbordered. An incompletable word of minimal length is $w = (ua)^{n-1}u$, $a \in A$. Thus,

$$n^2 + n - 1 \leq R_{\mathcal{P}}(n, d) \leq R_{\mathcal{P}}(n) \leq 2n^2.$$

Proposition (Restivo, 1978)

Let \mathcal{P} be the class of finite prefix codes. Then

$$R_{\mathcal{P}}(n) \leq 2n^2.$$

Example (Pribavkina, 2009)

Let $X = A^n \setminus \{u\}$, with $u \in A^n$ unbordered. An incompletable word of minimal length is $w = (ua)^{n-1}u$, $a \in A$. Thus,

$$n^2 + n - 1 \leq R_{\mathcal{P}}(n, d) \leq R_{\mathcal{P}}(n) \leq 2n^2.$$

Problem

What is the exact value ?

Example (Ananichev, Gusev, Volkov, 2009)

A synchronizing word of minimal length for the prefix code $X = a(a + b)^{n-1} + b(a + b)^{n-2}$ is

$$(ab^{n-2})^{n-1}$$

Thus,

$$C_{\mathcal{P}}(n, 2) \geq (n - 1)^2 \quad \text{and} \quad C_{\mathcal{M}}(n, 2) \geq (n - 1)^2$$

where \mathcal{M} is the class of maximal finite codes.

Synchronizing codes

Let \mathcal{M} be the class of complete finite codes and \mathcal{F} be the class of finite sets.

Proposition

For all $n, d > 0$,

$$C_{\mathcal{M}}(n, d) \leq 2R_{\mathcal{F}}(n, d + 1) + 2n - 2.$$

Synchronizing codes

Let \mathcal{M} be the class of complete finite codes and \mathcal{F} be the class of finite sets.

Proposition

For all $n, d > 0$,

$$C_{\mathcal{M}}(n, d) \leq 2R_{\mathcal{F}}(n, d + 1) + 2n - 2.$$

Thus, if Restivo's conjecture is true, then

$$C_{\mathcal{M}}(n, d) = O(n^2).$$

Synchronizing codes

Let \mathcal{MP} be the class of complete prefix finite codes.

Proposition

For all $n, d > 0$,

$$C_{\mathcal{MP}}(n, d) \leq R_{\mathcal{F}}(n, d + 1).$$

Synchronizing codes

Let \mathcal{MP} be the class of complete prefix finite codes.

Proposition

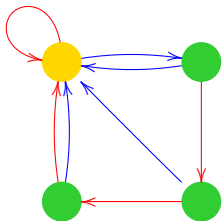
For all $n, d > 0$,

$$C_{\mathcal{MP}}(n, d) \leq R_{\mathcal{F}}(n, d + 1).$$

Remark

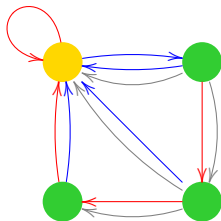
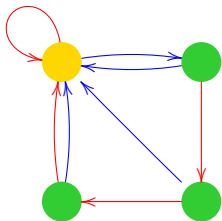
Beal, Perrin proved that a complete prefix code X has a synchronizing word of length $\leq 2\ell(X) \text{Card}(X)$. In view of Proposition above and Restivo's conjecture, this bound seems of no help in evaluating $C_{\mathcal{P}}(n, 2)$, as one may have $\text{Card}(X) \simeq 2^{\ell(X)}$.

Example



● \mathcal{A} accepts X^* with $X = \{a, baaa, baab, bab, bb\}$

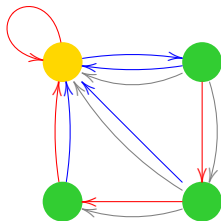
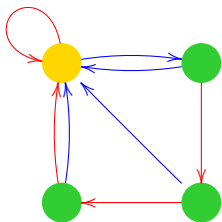
Example






● \mathcal{A} accepts X^* with $X = \{a, baaa, baab, bab, bb\}$

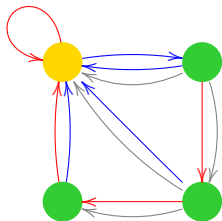
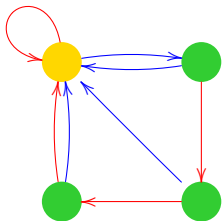
● \mathcal{A}' accepts Y^* with $\ell(X) = \ell(Y)$

Example



-  \mathcal{A} accepts X^* with $X = \{a, baaa, baab, bab, bb\}$
-  \mathcal{A}' accepts Y^* with $\ell(X) = \ell(Y)$
-  X is synchronizing iff Y is incomplete

Example



- \mathcal{A} accepts X^* with $X = \{a, baaa, baab, bab, bb\}$
- \mathcal{A}' accepts Y^* with $\ell(X) = \ell(Y)$
- X is synchronizing iff Y is incomplete
- Replacing c by a in a minimal incomplete word of Y gives a synchronizing word of X

Reduction to binary

Encoding a d -letter alphabet on a suitable binary code one obtains

Proposition

Let \mathcal{L} be the class of finite languages (resp. codes).

Then

$$R_{\mathcal{L}}(n, d) \leq \left\lceil \frac{R_{\mathcal{L}}(\lceil \log_2 d \rceil n, 2)}{\lfloor \log_2 d \rfloor} \right\rceil.$$

Proposition

Let \mathcal{L} be the class of finite complete languages (resp. codes, prefix codes). Then

$$C_{\mathcal{L}}(n, d) \leq \left\lceil \frac{C_{\mathcal{L}}(\lceil \log_2(d+1) \rceil n, 2)}{\lfloor \log_2(d-1) \rfloor} \right\rceil.$$

Reduction to binary

Encoding a d -letter alphabet on a suitable binary code one obtains

Proposition

Let \mathcal{L} be the class of finite languages (resp. codes).
Then

$$R_{\mathcal{L}}(n, d) \leq \left\lceil \frac{R_{\mathcal{L}}(\lceil \log_2 d \rceil n, 2)}{\lfloor \log_2 d \rfloor} \right\rceil.$$

Proposition

Let \mathcal{L} be the class of finite ~~complete~~ languages (resp. codes, prefix codes). Then

$$C_{\mathcal{L}}(n, d) \leq \left\lceil \frac{C_{\mathcal{L}}(\lceil \log_2(d+1) \rceil n, 2)}{\lceil \log_2(d+1) \rceil} \right\rceil.$$

A tool for reduction

Lemma

Let $k_1, \dots, k_n, d > 0$ be such that

$$\gcd(k_1, k_2, \dots, k_n) = 1, \quad \sum_{i=1}^n d^{-k_i} = 1.$$

Then k_1, \dots, k_n are the codeword lengths of a synchronizing prefix code over d letters.

A tool for reduction

Lemma

Let $k_1, \dots, k_n, d > 0$ be such that

$$\gcd(k_1, k_2, \dots, k_n) = 1, \quad \sum_{i=1}^n d^{-k_i} = 1.$$

Then k_1, \dots, k_n are the codeword lengths of a synchronizing prefix code over d letters.

Proof.

From Kraft Theorem and Trahtman Road-coloring Theorem. □

Thank you !

Reduction to binary

Lemma

- Let $h : A^* \rightarrow B^*$ be a prefix encoding and $Y \subseteq A^*$. The set $h(Y)$ is complete if and only if Y and $h(A)$ are complete.
- Let $h : A^* \rightarrow B^*$ be a monomorphism and $Y \subseteq A^*$ be a complete set. The set $h(Y)$ is synchronizing if and only if Y and $h(A)$ are synchronizing.

Proof uses the following

Lemma

If Y is complete, any word has a completion (y, s) with $y \in Y^*$

Lemma

Let X be a synchronizing code such that $\ell(X) = n$ and \mathcal{A} be an unambiguous trim automaton (with unique start and accepting state) accepting X^* . For all $u \in X^*$, there are words $w_1, w_2 \in A^*$ such that

$$|w_1|, |w_2| \leq R_{\mathcal{F}}(n, d + 1), \quad Qw_1 \subseteq Qu, \quad Qw_2^{-1} \subseteq Qu^{-1}.$$

Proof.

Starting from \mathcal{A} , construct a nondeterministic incomplete automaton on a larger alphabet and obtain w_1, w_2 from minimal incompletable words of the accepted language. □