

Algebraic Structure Theory of Tail-Biting Trellises

Priti Shankar

Department of Computer Science and Automation
Indian Institute of Science

Outline

- 1 Some History
- 2 Some Old Theorems
- 3 Tailbiting Trellises

Deterministic Finite Automata

- Q : A Finite number of states
- Σ A Finite alphabet set
- δ : A Transition function that maps $Q \times \Sigma \mapsto Q$
- q_0 : Start state
- $F \subseteq Q$: A set of final states

Deterministic Finite Automata

- Q : A Finite number of states
- Σ A Finite alphabet set
- δ : A Transition function that maps $Q \times \Sigma \mapsto Q$
- q_0 : Start state
- $F \subseteq Q$: A set of final states

Deterministic Finite Automata

- Q : A Finite number of states
- Σ A Finite alphabet set
- δ : A Transition function that maps $Q \times \Sigma \mapsto Q$
- q_0 : Start state
- $F \subseteq Q$: A set of final states

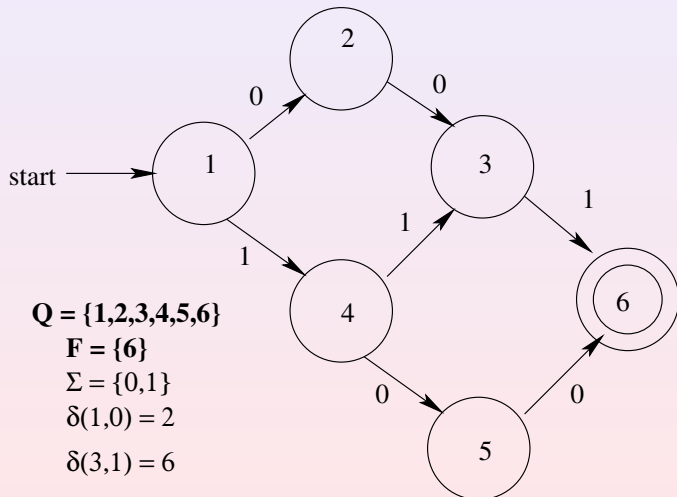
Deterministic Finite Automata

- Q : A Finite number of states
- Σ A Finite alphabet set
- δ : A Transition function that maps $Q \times \Sigma \mapsto Q$
- q_0 : Start state
- $F \subseteq Q$: A set of final states

Deterministic Finite Automata

- Q : A Finite number of states
- Σ A Finite alphabet set
- δ : A Transition function that maps $Q \times \Sigma \mapsto Q$
- q_0 : Start state
- $F \subseteq Q$: A set of final states

Example



Right invariance and bi-invariance

Let R be a binary relation on Σ^* . R is said to be **right invariant** whenever $x R y \Rightarrow xa R ya$; $x, y \in \Sigma^*, \forall a \in \Sigma$.

The relation R is **bi-invariant** if in addition to the above property, R also satisfies *right-cancellation* i.e. $xa R ya \Rightarrow x R y$, $\forall a \in \Sigma$.

The Myhill Nerode Theorem (1958)

The following statements are equivalent:

- 1 $L \subseteq \Sigma^*$ is accepted by a DFA
- 2 L is the union of some of the equivalence classes of a right invariant equivalence relation of finite index.
- 3 Define a relation R_L as follows. For $x, y, \in \Sigma^*$ $(x, y) \in R_L$ iff

$$\forall z \in \Sigma^* \quad xz \in L \iff yz \in L$$

(That is, x and y share all continuations to the language).
Then R_L has finite index.

The Myhill Nerode Theorem (1958)

The following statements are equivalent:

- 1 $L \subseteq \Sigma^*$ is accepted by a DFA
- 2 L is the union of some of the equivalence classes of a right invariant equivalence relation of finite index.
- 3 Define a relation R_L as follows. For $x, y, \in \Sigma^*$ $(x, y) \in R_L$ iff

$$\forall z \in \Sigma^* \quad xz \in L \iff yz \in L$$

(That is, x and y share all continuations to the language).
Then R_L has finite index.

The Myhill Nerode Theorem (1958)

The following statements are equivalent:

- 1 $L \subseteq \Sigma^*$ is accepted by a DFA
- 2 L is the union of some of the equivalence classes of a right invariant equivalence relation of finite index.
- 3 Define a relation R_L as follows. For $x, y, \in \Sigma^*$ $(x, y) \in R_L$ iff

$$\forall z \in \Sigma^* \quad xz \in L \iff yz \in L$$

(That is, x and y share all continuations to the language).
Then R_L has finite index.

Bideterministic Languages

- Let Σ be a finite alphabet, and L be a language over Σ accepted by a finite state automaton.
- Let L^R be the language obtained by reversing all strings in L .
- A finite state language is **bideterministic** if there exists a deterministic finite state automaton (DFA) M accepting L , such that a dfa for L^R is obtained by simply reversing all edges of M .

Bideterministic Languages

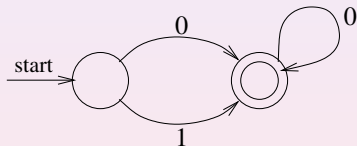
- Let Σ be a finite alphabet, and L be a language over Σ accepted by a finite state automaton.
- Let L^R be the language obtained by reversing all strings in L .
- A finite state language is **bideterministic** if there exists a deterministic finite state automaton (DFA) M accepting L , such that a dfa for L^R is obtained by simply reversing all edges of M .

Bideterministic Languages

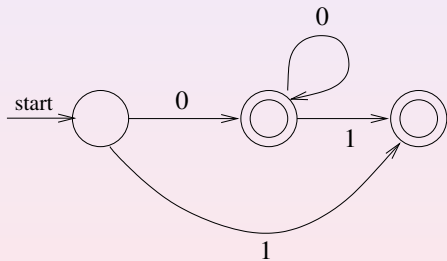
- Let Σ be a finite alphabet, and L be a language over Σ accepted by a finite state automaton.
- Let L^R be the language obtained by reversing all strings in L .
- A finite state language is **bideterministic** if there exists a deterministic finite state automaton (DFA) M accepting L , such that a dfa for L^R is obtained by simply reversing all edges of M .

DFA for a non bi-deterministic language

The language accepted by DFA (a) is non bideterministic.



(a)



(b)

Is there a “stronger” Myhill-Nerode like theorem for such languages?

Theorem

The following are equivalent statements

- (i) *L is a bideterministic finite state language over Σ^* .*
- (ii) *L is an equivalence class of a bi-invariant equivalence relation over Σ^* of finite index.*
- (iii) *Define R_L as follows: $(x, y) \in R_L$ iff*
 - (a) *$\forall z \in \Sigma^*, [xz \in L \text{ whenever } yz \in L]$;*
(if two strings are in the same equivalence class then they share all continuations)
 - (b) *$(x, y) \notin R_L$ iff $\forall z \in \Sigma^* [xz \in L \Rightarrow yz \notin L]$ and .*
*(if two strings are in different equivalence classes, they share no continuations)**Then R_L is a bi-invariant equivalence relation of finite index with a unique class corresponding to L .*

Theorem

The following are equivalent statements

- (i) L is a bideterministic finite state language over Σ^* .
- (ii) L is an equivalence class of a bi-invariant equivalence relation over Σ^* of finite index.
- (iii) Define R_L as follows: $(x, y) \in R_L$ iff
 - (a) $\forall z \in \Sigma^*, [xz \in L \text{ whenever } yz \in L]$;
(if two strings are in the same equivalence class then they share all continuations)
 - (b) $(x, y) \notin R_L$ iff $\forall z \in \Sigma^* [xz \in L \Rightarrow yz \notin L]$ and .
(if two strings are in different equivalence classes, they share no continuations)Then R_L is a bi-invariant equivalence relation of finite index with a unique class corresponding to L .

Theorem

The following are equivalent statements

- (i) L is a bideterministic finite state language over Σ^* .
- (ii) L is an equivalence class of a bi-invariant equivalence relation over Σ^* of finite index.
- (iii) Define R_L as follows: $(x, y) \in R_L$ iff
 - (a) $\forall z \in \Sigma^*, [xz \in L \text{ whenever } yz \in L]$;
(if two strings are in the same equivalence class then they share all continuations)
 - (b) $(x, y) \notin R_L$ iff $\forall z \in \Sigma^* [xz \in L \Rightarrow yz \notin L]$ and .
(if two strings are in different equivalence classes, they share no continuations)Then R_L is a bi-invariant equivalence relation of finite index with a unique class corresponding to L .

Theorem

The following are equivalent statements

- (i) L is a bideterministic finite state language over Σ^* .
- (ii) L is an equivalence class of a bi-invariant equivalence relation over Σ^* of finite index.
- (iii) Define R_L as follows: $(x, y) \in R_L$ iff
 - (a) $\forall z \in \Sigma^*, [xz \in L \text{ whenever } yz \in L]$;
(if two strings are in the same equivalence class then they share all continuations)
 - (b) $(x, y) \notin R_L$ iff $\forall z \in \Sigma^* [xz \in L \Rightarrow yz \notin L]$ and .
(if two strings are in different equivalence classes, they share no continuations)

Then R_L is a bi-invariant equivalence relation of finite index with a unique class corresponding to L .

Linear Block Codes

Linear Block Codes are a special subclass of bideterministic languages

A linear (n,k) block code is a k -dimensional subspace of an n -dimensional vector space. A generator matrix G has rows corresponding to k basis vectors for the code. A parity check matrix H has $n-k$ rows which are basis vectors of the orthogonal space called the dual code.

G and H matrices for a $(7,4)$ binary Hamming code

$$G = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Conventional Trellises

A conventional trellis $T = (V, E, \Sigma)$ of depth n is an edge-labeled directed graph with the property that the set V can be partitioned into $n + 1$ vertex classes

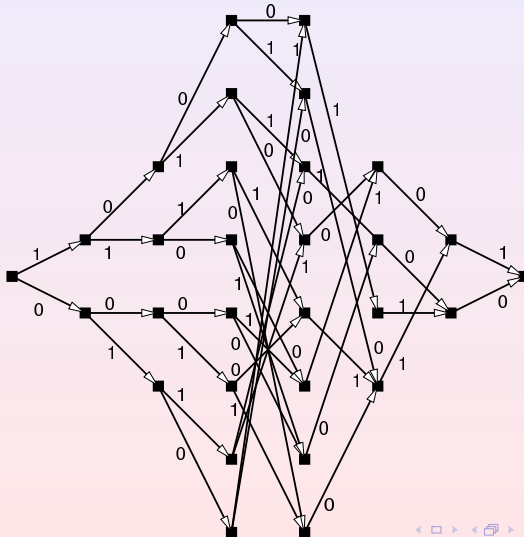
$$V = V_0 \cup V_1 \cup \dots \cup V_n$$

where $|V_0| = |V_n| = 1$, such that every edge in T is labeled with a symbol from the alphabet Σ , and begins at a vertex of V_i and ends at a vertex of V_{i+1} , for some $i \in \{0, 1, \dots, n - 1\}$.

The integers $\{0, 1, 2, \dots, n\}$ are called **time indices**.

The conventional trellis is just an automaton recognizing some code over Σ . Trellises are useful for maximum likelihood decoding.

A conventional trellis for a (7,4) Hamming Code



Bideterministic languages have the property that the minimal automaton (among all non-deterministic as well as deterministic automata) accepting the language is the unique DFA for the language. This is not true in general as a minimal DFA can be exponential in the size of an NFA for the same language.

Linear Block Codes are a special subclass of the class of bideterministic languages. Their minimal trellises are said to be biproper.

Some definitions for trellises for linear block codes

- Support of a codeword
- State Cardinality profile
- State Complexity Profile
- Width of the trellis
- Section of a trellis
- Span of a codeword
- Atomic codewords
- Labeled trellis
- Past and future codewords

Some definitions for trellises for linear block codes

- Support of a codeword
- State Cardinality profile
- State Complexity Profile
- Width of the trellis
- Section of a trellis
- Span of a codeword
- Atomic codewords
- Labeled trellis
- Past and future codewords

Some definitions for trellises for linear block codes

- Support of a codeword
- State Cardinality profile
- State Complexity Profile
- Width of the trellis
- Section of a trellis
- Span of a codeword
- Atomic codewords
- Labeled trellis
- Past and future codewords

Some definitions for trellises for linear block codes

- Support of a codeword
- State Cardinality profile
- State Complexity Profile
- Width of the trellis
- Section of a trellis
- Span of a codeword
- Atomic codewords
- Labeled trellis
- Past and future codewords

Some definitions for trellises for linear block codes

- Support of a codeword
- State Cardinality profile
- State Complexity Profile
- Width of the trellis
- Section of a trellis
- Span of a codeword
- Atomic codewords
- Labeled trellis
- Past and future codewords

Some definitions for trellises for linear block codes

- Support of a codeword
- State Cardinality profile
- State Complexity Profile
- Width of the trellis
- Section of a trellis
- Span of a codeword
- Atomic codewords
- Labeled trellis
- Past and future codewords

Some definitions for trellises for linear block codes

- Support of a codeword
- State Cardinality profile
- State Complexity Profile
- Width of the trellis
- Section of a trellis
- Span of a codeword
- Atomic codewords
- Labeled trellis
- Past and future codewords

Some definitions for trellises for linear block codes

- Support of a codeword
- State Cardinality profile
- State Complexity Profile
- Width of the trellis
- Section of a trellis
- Span of a codeword
- Atomic codewords
- Labeled trellis
- Past and future codewords

Some definitions for trellises for linear block codes

- Support of a codeword
- State Cardinality profile
- State Complexity Profile
- Width of the trellis
- Section of a trellis
- Span of a codeword
- Atomic codewords
- Labeled trellis
- Past and future codewords

Some important results for conventional trellises for linear block codes

- The state space at any time index i has dimension $s_i = \dim[C] - \dim[C_{i-}] - \dim[c_{i+}]$ (Forney 1994)
- The minimal trellis can be constructed as a cartesian product of trellises for atomic codewords with spans not beginning or ending at the same time index. (Kschischang and Sorokine(1995))
- The state spaces can be labeled so that the labels themselves form a linear vector space, the dimension being the state complexity. (Bahl, Cocke, Jelinek, Raviv, 1974). This labeling scheme produces the minimal trellis—the BCJR trellis. The dual trellis has the same state complexity profile.

Some important results for conventional trellises for linear block codes

- The state space at any time index i has dimension $s_i = \dim[C] - \dim[C_{i-}] - \dim[c_{i+}]$ (Forney 1994)
- The minimal trellis can be constructed as a cartesian product of trellises for atomic codewords with spans not beginning or ending at the same time index. (Kschischang and Sorokine(1995))
- The state spaces can be labeled so that the labels themselves form a linear vector space, the dimension being the state complexity. (Bahl, Cocke, Jelinek, Raviv, 1974). This labeling scheme produces the minimal trellis—the BCJR trellis. The dual trellis has the same state complexity profile.

Some important results for conventional trellises for linear block codes

- The state space at any time index i has dimension $s_i = \dim[C] - \dim[C_{i-}] - \dim[c_{i+}]$ (Forney 1994)
- The minimal trellis can be constructed as a cartesian product of trellises for atomic codewords with spans not beginning or ending at the same time index. (Kschischang and Sorokine(1995))
- The state spaces can be labeled so that the labels themselves form a linear vector space, the dimension being the state complexity. (Bahl, Cocke, Jelinek, Raviv, 1974). This labeling scheme produces the minimal trellis—the BCJR trellis. The dual trellis has the same state complexity profile.

Some important results for conventional trellises (continued)

- Sectioning a trellis can reduce its state complexity. (Vardy and Be'ery)
- Permuting the time indices can greatly reduce width, but getting the best permutation is NP-hard(Kashyap 2007)

Some important results for conventional trellises (continued)

- Sectioning a trellis can reduce its state complexity. ([Vardy and Be'ery](#))
- Permuting the time indices can greatly reduce width, but getting the best permutation is NP-hard([Kashyap 2007](#))

Group-theoretic characterization of a conventional trellis

Let C be a code of length n over a finite alphabet A . We define two shortened versions of C as follows:

$$\begin{aligned} \mathcal{P}_i &= \{(c_1, c_2, \dots, c_i) : \exists (c_{i+1} = c_{i+2} \dots = c_n = 0) \\ &\quad \ni (c_1, c_2, \dots, c_i, c_{i+1}, c_{i+2} \dots c_n) \in C\} \end{aligned}$$

The above is the past subcode of C .

$$\begin{aligned} \mathcal{F}_i &= \{c_{i+1}, c_{i+2} \dots c_n : \exists (c_1 = c_2 = \dots = c_i = 0) \\ &\quad \ni (c_1, c_2, \dots, c_i, c_{i+1}, c_{i+2} \dots c_n) \in C\} \end{aligned}$$

The above is the future subcode of C at i . Their direct sum $\mathcal{P}_i \oplus \mathcal{F}_i$ is a linear subcode of C .

The Forney Algebraic Characterization of Conventional Trellises: (Forney 1988) The set of vertices at time index i is given by

$$V_i = C/\mathcal{P}_i \oplus \mathcal{F}_i$$

the set of cosets of

$$\mathcal{P}_i \oplus \mathcal{F}_i$$

in C .

$$\mathcal{P}_n = \mathcal{F}_0 = C$$

$$\mathcal{P}_0 = \mathcal{F}_n = \{0\}$$

by convention.

There is an edge from a vertex v in V_i to v' in V_{i+1} if there is a codeword $c \in v \cap v'$.

The Forney construction gives the minimal trellis.

Definition

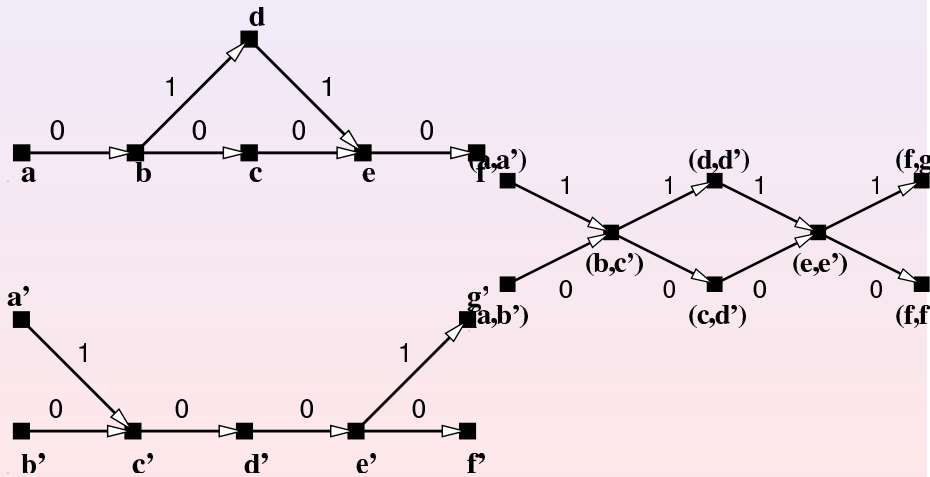
A *tail-biting trellis* $T = (V, E, \Sigma)$ of depth n is an edge-labeled directed graph with the property that the set V can be partitioned into n vertex classes

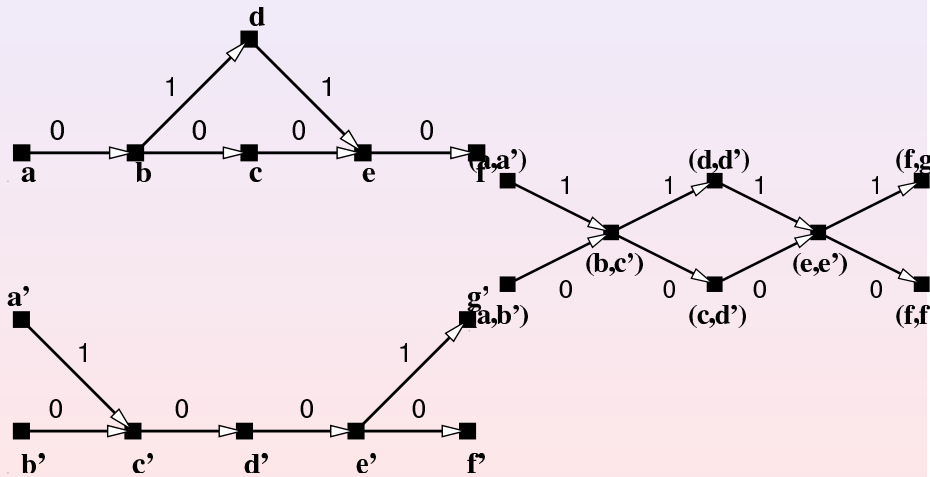
$$V = V_0 \cup V_1 \cup \cdots \cup V_{n-1} \quad (1)$$

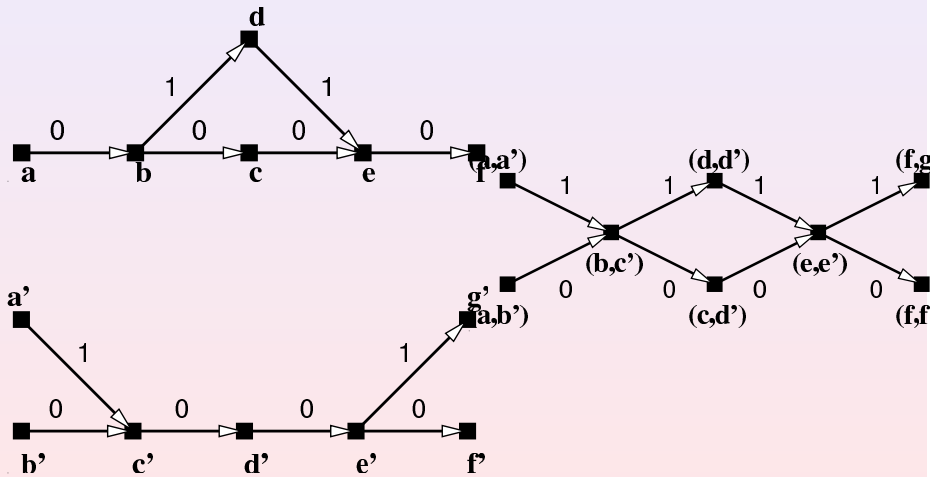
such that every edge in T is labeled with a symbol from the alphabet Σ , and begins at a vertex of V_i and ends at a vertex of V_{i+1} , for some $i \in \{0, 1, \dots, n-1\}$.

As in the case with conventional trellises, the set of indices $\mathcal{I} = \{0, 1, \dots, n-1\}$ are the *time indices*. We identify \mathcal{I} with Z_n , the residue classes of integers modulo n . An interval of indices $[i, j]$ represents the sequence $\{i, i+1, \dots, j\}$ if $i < j$, and the sequence $\{i, i+1, \dots, n-1, 0, \dots, j\}$ if $i > j$. Every cycle of length n in T starting at a vertex of V_0 defines a vector $a_1, a_2, \dots, a_n \in \Sigma^n$ which is an *edge-label sequence*. We assume that every vertex and every edge in the tail-biting trellis lies on some cycle. The trellis T is said to *represent* a block code C over Σ if the set of all edge-label sequences in T is equal to C . Let $C(T)$ denote the code represented by the trellis T .

A tail-biting trellis can be viewed as a **splitting** a conventional trellis into subtrellises and then **overlaying** the subtrellises so that they share states.







Myhill Nerode like theorem for tail-biting trellises

Theorem

The following statements are equivalent:

- T is a linear one-to-one, biproper tail-biting trellis for a code C over $GF(q)$ corresponding to coset decomposition $C = \{C_1, C_2, \dots, C_{l-1}\}$ with merging intervals $[j_1, j_1], [j_2, j_2], \dots, [j_l, j_l]$, and state complexity profile s_1, s_2, \dots, s_n .*
- C is the union of l equivalence classes of a linear, distributed cyclic bi-invariant equivalence relation with indices j_1, j_2, \dots, j_l , and index profile $q^{s_1}, q^{s_2}, \dots, q^{s_l}$.*

Myhill Nerode like theorem for tail-biting trellises

Theorem

The following statements are equivalent:

- *T is a linear one-to-one, biproper tail-biting trellis for a code C over $GF(q)$ corresponding to coset decomposition $C = \{C_1, C_2, \dots, C_{l-1}\}$ with merging intervals $[i_1, j_1], [i_2, j_2], \dots, [i_l, j_l]$, and state complexity profile s_1, s_2, \dots, s_n .*
- *C is the union of l equivalence classes of a linear, distributed cyclic bi-invariant equivalence relation with indices j_1, j_2, \dots, j_l , and index profile $q^{s_1}, q^{s_2}, \dots, q^{s_l}$.*

Myhill Nerode like theorem for tail-biting trellises

Theorem

The following statements are equivalent:

- *T is a linear one-to-one, biproper tail-biting trellis for a code C over $GF(q)$ corresponding to coset decomposition $C = \{C_1, C_2, \dots, C_{l-1}\}$ with merging intervals $[i_1, j_1], [i_2, j_2], \dots, [i_l, j_l]$, and state complexity profile s_1, s_2, \dots, s_n .*
- *C is the union of l equivalence classes of a linear, distributed cyclic bi-invariant equivalence relation with indices j_1, j_2, \dots, j_l , and index profile $q^{s_1}, q^{s_2}, \dots, q^{s_l}$.*

Some definitions

Definition

P1..An equivalence relation R over Σ^* is said to be cyclic right invariant if there is an integer i such that for all $x, y \in \Sigma^*$ $(x, y) \in R$ either $(xa, ya) \in R$ or $(\text{cyr}^i(xa), \text{cyr}^i(ya)) \in R$.

P2. An equivalence relation R over Σ^* is said to be distributed cyclic right-invariant if there is a set of integers $I = \{i_1, i_2, \dots, i_n\}$, such that for all $x, y \in \Sigma^*$ $(x, y) \in R, a \in \Sigma$, either $(xa, ya) \in R$ or $\exists i \in I$ such that $(\text{cyr}^i(xa), \text{cyr}^i(ya)) \in R$.

P3. an equivalence relation R over Σ^* is distributed cyclic bi-invariant if it satisfies property P2 and is right cancellative.

A Myhill-Nerode like theorem for tail-biting trellises.

Theorem

A language L is accepted by a bideterministic one-to-one tailbiting automaton. with I start states iff L is the union of I equivalence classes of a distributed cyclic bi-invariant equivalence relation of finite index.

The BCJR Trellis

The BCJR trellis (Bahl, Cocke, Jelinek, Raviv) was introduced in 1974.

As before, let G and H be the generator and parity check matrices for a linear block code.

