

Stringological Sequence Prediction

Vanessa Kosoy

Faculty of Mathematics, Technion

Computational Rational Agents Laboratory

One World Combinatorics on Words Seminar

Based on

I Efficient algorithms for predicting highly repetitive sequences

arXiv:2603.26852

II Right-to-left automaticity and related complexity measures

arXiv:2607.17369

III Layered ziplines and a tradeoff between efficiency and expressivity

arXiv:2609.19940

Sequence prediction

Given a prefix of a word, guess the next symbol.

Requirement: on *simple* words, i.e. words with simple patterns, make few mistakes.

simple

0110100110010110100101100110100?

random-looking

1011000111010010111100101000110?

Predictors

DEFINITION 2.1

Fix a finite alphabet Σ . A *predictor* is a tuple $\Pi = (S, s_{\text{init}}, U, P)$ where

$S \subseteq \{0, 1\}^*$ *set of states*

$s_{\text{init}} \in S$ *initial state*

$U : S \times \Sigma \rightarrow S$ *update function*

$P : S \rightarrow \Sigma$ *prediction function*

Deterministic sequence prediction

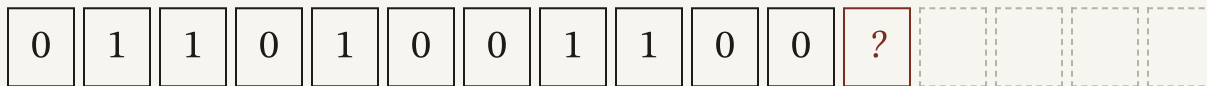
Run Π on $x \in \Sigma^\omega$: $s_0 := s_{\text{init}}$, and for $t = 0, 1, 2, \dots$

1. $\hat{x}_t := P(s_t)$
2. $x[t]$ is revealed
3. $s_{t+1} := U(s_t, x[t])$

mistakes on $u \in \Sigma^T$ $M_\Pi(u) := |\{ t < T : \hat{x}_t \neq u[t] \}|$

state size $S_\Pi(u) := \max_{t < T} |s_t|$

Thue–Morse, $t = 11$



Word complexity measures

A function $C : \Sigma^* \rightarrow \mathbb{N}$ such that

polynomial bound $C(u) \leq \text{poly}(|u|)$

approximate monotonicity $C(u[:k]) \leq (C(u) + 1) \cdot \text{poly}(\log |u|, \log C(u))$

Low complexity = highly structured.

Statistical efficiency

DEFINITION 2.3

A predictor Π is *statistically efficient* w.r.t. C if for all $u \in \Sigma^*$

$$M_{\Pi}(u) \leq (C(u) + 1) \cdot \text{poly}(\log C(u), \log |u|)$$

quasilinear in the complexity, polylogarithmic in the length

if $C(x[:n]) = \text{polylog } n$, then $\text{polylog } n$ mistakes

The counting criterion

COUNTING COMPLEXITY

$$N_C(n, m) := \log |\{ u \in \Sigma^n : C(u) \leq m \}|$$

THEOREM 2.6

There exists a statistically efficient predictor for C if and only if

$$N_C(n, m) \leq (m + 1) \cdot \text{poly}(\log m, \log n)$$

Computational efficiency

DEFINITION 2.4 (II)

Π is *computationally feasible* if

- $P(s_t)$ is computable in time $\text{poly}(|s_t|)$
- $U(s_t, x[t])$ is computable in time $\text{poly}(|s_t|)$
- $S_\Pi(x) \leq \text{poly}(|x|)$

DEFINITION 2.5 (II)

Π is *computationally efficient* w.r.t. C if it is computationally feasible and for all $x \in \Sigma^*$

$$S_\Pi(x) \leq \text{poly}(C(x), \log |x|)$$

Automaticity

DEFINITION 3.1 (II)

Fix $k \geq 2$, and let $\langle t \rangle_k^m \in [k]^m$ be the base- k representation of t , most significant digit first, padded to length m . For $x \in \Sigma^*$, the minimal number of states of a DFAO M such that for some m with $|x| \leq k^m$, for all $t < |x|$:

$$x[t] = M(\langle t \rangle_k^m)$$

$$\text{LTR } k\text{-automaticity } \text{AC}_k^{\text{L}}(x)$$

$$x[t] = M((\langle t \rangle_k^m)^{\text{R}})$$

$$\text{RTL } k\text{-automaticity } \text{AC}_k^{\text{R}}(x)$$

Example

Any prefix x of Thue–Morse with $|x| \geq 2$: $\text{AC}_2^{\text{L}}(x) = \text{AC}_2^{\text{R}}(x) = 2$

Predicting w.r.t. LTR automaticity

THEOREM 3.2 (I)

The Hierarchical Dictionary Plurality (HDP) predictor is statistically and computationally efficient w.r.t. AC_k^L . For $x \in \Sigma^n$, $m := AC_k^L(x)$, $\sigma := |\Sigma|$:

$$\text{mistakes} \quad M_{\text{HDP}}(x) = O(km \log m (\log n)^2)$$

$$\text{state size} \quad S_{\text{HDP}}(x) = O(km (\log m \log n + \log \sigma))$$

Predicting w.r.t. RTL automaticity

THEOREM 3.4 (II)

The Tabular Plurality (TP) predictor is statistically and computationally efficient w.r.t. AC_k^R . For $n \geq 2$, $x \in \Sigma^n$, $m := AC_k^R(x)$, $\sigma := |\Sigma|$:

$$\text{mistakes} \quad M_{TP}(x) = O(km \log(km) \log n)$$

$$\text{state size} \quad S_{TP}(x) = O(m (k \log m \log n + (\log n)^2 + \log \sigma))$$

Straight-line programs

DEFINITION 4.1 (I)

An SLP over Σ is $P = (Q, q_0, \delta)$ with $\delta : Q \rightarrow (Q \sqcup \Sigma)^*$ such that

- $|\delta(q)| \geq 2$ for all $q \neq q_0$
- δ is acyclic, and q_0 is the unique state not appearing in any $\delta(q)$

$\text{val}(q) := \text{val}(\delta(q)[0]) \cdots \text{val}(\delta(q)[l-1])$, $\text{val}(P) := \text{val}(q_0)$, $|P| := \sum_q |\delta(q)|$

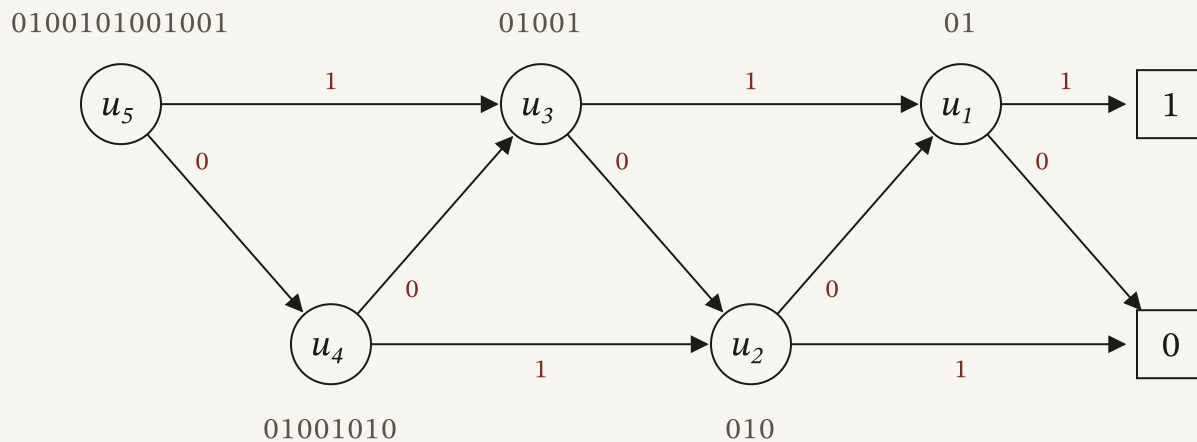
DEFINITION 4.2 (I)

Straight-line complexity $\text{SLC}(x) := \min \{ |P| : \text{val}(P) = x \}$

Example

Fibonacci word: $u_0 = 0$, $u_1 = 01$, $u_{j+2} = u_{j+1}u_j \implies \text{SLC}(u_\infty[:n]) = O(\log n)$

A straight-line program for the Fibonacci word



$$\delta(u_j) = (u_{j-1}, u_{j-2}), \quad \delta(u_1) = (0, 1) \quad \text{val}(u_5) = 0100101001001 \quad |P| = 10$$

Straight-line complexity and automaticity

PROPOSITION 4.5 (I)

For any base $k \geq 2$ and word x ,

$$\text{SLC}(x) = O(k \cdot \text{AC}_k^{\text{L}}(x) \cdot \log |x|)$$

a predictor efficient w.r.t. SLC is efficient w.r.t. AC_k^{L} for every k

SLC does not depend on a choice of base

LZ77 factorization

DEFINITION 4.4 (I)

The LZ77 factorization of x is $x = f_0 f_1 \cdots f_{z-1}$, where each f_j is

- a single letter occurring for the first time, or
- the longest prefix of the remaining suffix with an earlier occurrence in x (overlap allowed)

LZ complexity $\text{LZC}(x) := z$

Fibonacci word, first 32 letters

0	1	0	010	10010	01010010	1001001010010
---	---	---	-----	-------	----------	---------------

Straight-line complexity and LZ complexity

THEOREM (RYTTER 2003)

$$\text{LZC}(x) \leq \text{SLC}(x) = O(\text{LZC}(x) \cdot \log |x|)$$

statistically (resp. computationally) efficient w.r.t. SLC $\Leftrightarrow$ w.r.t. LZC

LZ complexity of morphic words

$h : \Gamma^* \rightarrow \Gamma^*$ non-erasing, prolongable on a , $w = h^\omega(a)$, growth $h_a(n) := |h^n(a)| = \Theta(n^e \rho^n)$

THEOREM (CONSTANTINESCU-ILIE 2006)

w ultimately periodic

$$\text{LZC}(w[:n]) = \Theta(1)$$

not ultimately periodic, $\rho > 1$

$$\text{LZC}(w[:n]) = \Theta(\log n)$$

not ultimately periodic, $\rho = 1$, $e \geq 2$

$$\text{LZC}(w[:n]) = \Theta(n^{1/e})$$

Straight-line complexity of characteristic words

For $\theta \in (0, 1)$: $y[t] := \lfloor (t+2)\theta \rfloor - \lfloor (t+1)\theta \rfloor$

PROPOSITION G.3 (I)

For any $\theta \in (0, 1)$ and $n \in \mathbb{N}$,

$$\text{SLC}(y[:n]) \leq 24 \log n + 8$$

morphic only when θ is a quadratic irrational

idea: $s_{j+1} = s_j^{a_{j+1}} s_{j-1}$ for $\theta = [0; a_1, a_2, \dots]$, powers by repeated squaring

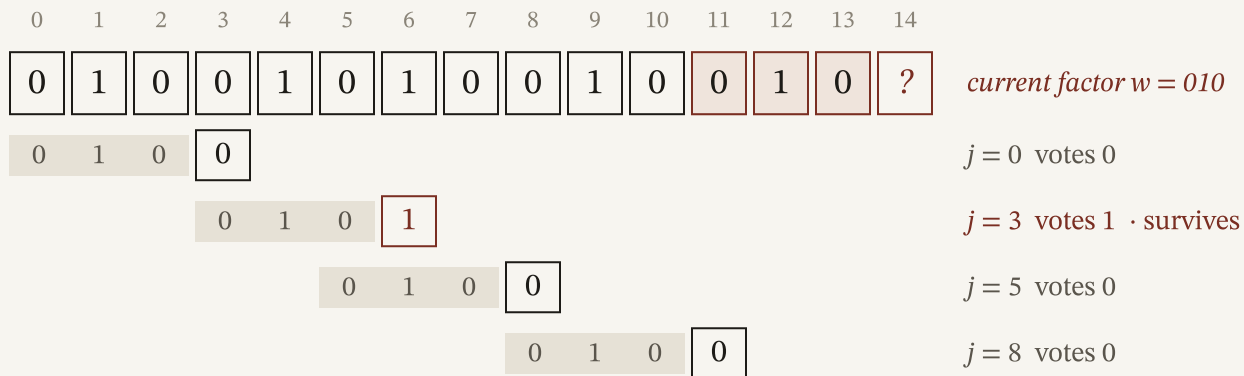
Lempel–Ziv Plurality

History $x[:t]$, *current factor* $w :=$ last factor of its LZ77 factorization

voters positions $j < t - |w|$ with $x[j:j + |w|] = w$

predict voter j votes for $x[j + |w|]$; output the plurality

Fibonacci word, $t = 14$



plurality predicts 0; $x[14] = 1$ is a mistake, and the voters shrink from 4 to 1

Predicting w.r.t. straight-line complexity

THEOREM 4.6 (I)

The Lempel–Ziv Plurality (LZP) predictor is statistically and computationally efficient w.r.t. LZC. For $x \in \Sigma^n$, $m := \text{LZC}(x)$, $\sigma := |\Sigma|$:

$$\text{mistakes} \quad M_{\text{LZP}}(x) = O(m \log n)$$

$$\text{state size} \quad S_{\text{LZP}}(x) = O(m (\log n + \log \sigma))$$

at most $\log n + 1$ mistakes per factor

Zipline programs

ZIP

$\text{zip}(u_0, \dots, u_{r-1})[i] := u_{i \bmod r}[\lfloor i/r \rfloor]$, stopping before the first missing symbol

$$\text{zip}(\text{abc}, 12) = \text{a1b2c}$$

DEFINITION 4.5 (II)

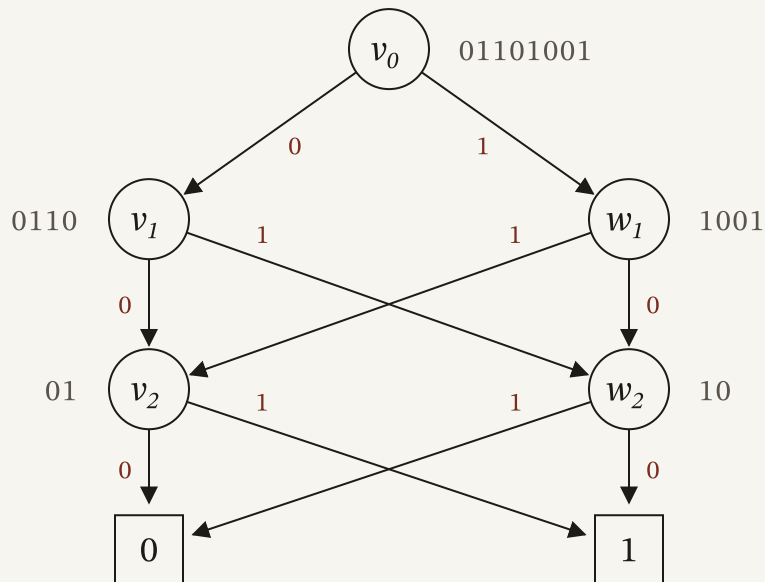
A *zipline program* (ZLP) is $P = (Q, q_0, \delta)$ as for an SLP, with

$$\text{val}(q) := \text{zip}(\text{val}(\delta(q)[0]), \dots, \text{val}(\delta(q)[l-1]))$$

DEFINITION 4.6 (II)

Zipline complexity $\text{ZLC}(x) := \min \{ |P| : x \sqsubseteq \text{val}(P) \}$

A zipline program for the Thue–Morse word



$$\text{val}(v_0) = \text{zip}(0110, 1001) = 01101001 = u_3 \quad |P| = 10$$

Arithmetic repetition complexity

Arithmetic interval $\alpha = (r, i, a, b)$: $\alpha(t) := r(a + t) + i$ for $t < |\alpha| := b - a$

DEFINITIONS 4.1, 4.2 (II)

An ARS for $u \in \Sigma^n$ is $R = (L, \Phi)$, $L \subseteq \Sigma \times [n]$, Φ a set of pairs of arithmetic intervals:

literals $(a, i) \in L \Rightarrow u[i] = a$

equations $(\alpha, \beta) \in \Phi \Rightarrow |\alpha| = |\beta|$ and $u[\alpha] = u[\beta]$

coverage each $i < n$ is a literal, or $\alpha(t) < \beta(t) = i$ for some $(\alpha, \beta) \in \Phi$

$\text{ARC}(x) := \min |L| + |\Phi|$

Example: Thue–Morse

$x[0] = 0, x[1] = 1, x[t] = x[2t], x[2t + 1] = x[4t + 1], x[t] = x[4t + 3] \Rightarrow \text{ARC}(x[: n]) \leq 5$

Predicting w.r.t. arithmetic repetition complexity

THEOREM 4.11 (II)

ARC-MUWU is statistically efficient and computationally feasible w.r.t. ARC. For $n \geq 2, x \in \Sigma^n, m := \text{ARC}(x)$:

$$\text{mistakes} \quad M_{\text{ARC-MUWU}}(x) = O(m \log n)$$

open: a computationally efficient predictor w.r.t. ARC

ARC and zipline complexity

PROPOSITION 4.7 (II)

For every $n \in \mathbb{N}$ and $x \in \Sigma^n$,

$$\text{ARC}(x) \leq 2 (\sqrt[3]{2n} + 1) \cdot \text{ZLC}(x)$$

also $\text{ARC}(x) \leq \text{LZC}(x)$ (Proposition 4.3)

open: is the factor $\sqrt[3]{n}$ tight?

Synchronous zipline programs

DEFINITION 4.8 (II)

A ZLP is *synchronous* if there is $\phi : Q \rightarrow \mathbb{N}$ with

$$\phi(q_0) = 1, \quad q' \text{ in } \delta(q) \implies \phi(q') = |\delta(q)| \cdot \phi(q)$$

every path from q_0 to q gives a progression with the same step

$$\text{sZLC}(x) := \min \{ |P| : P \text{ synchronous, } x \sqsubseteq \text{val}(P) \}$$

synchronous: Thue–Morse, $\phi(v_j) = \phi(w_j) = 2^j$

$$\delta(v_j) = (v_{j+1}, w_{j+1}), \quad \delta(w_j) = (w_{j+1}, v_{j+1})$$

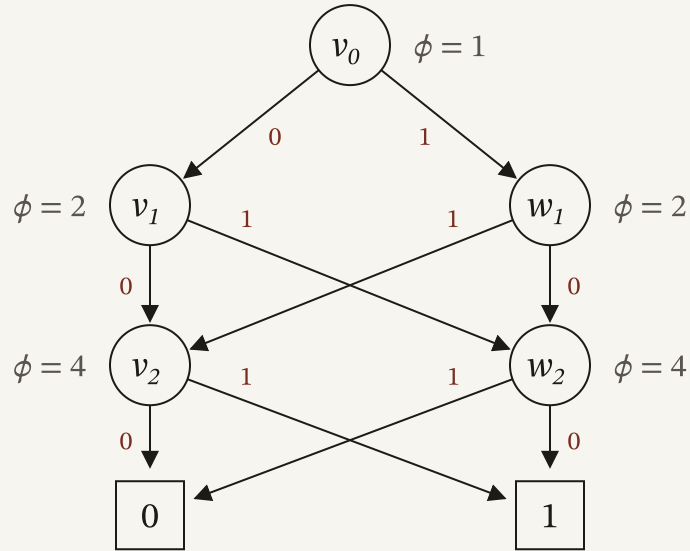
$$\text{sZLC}(u_\infty[:n]) = O(\log n)$$

not synchronous

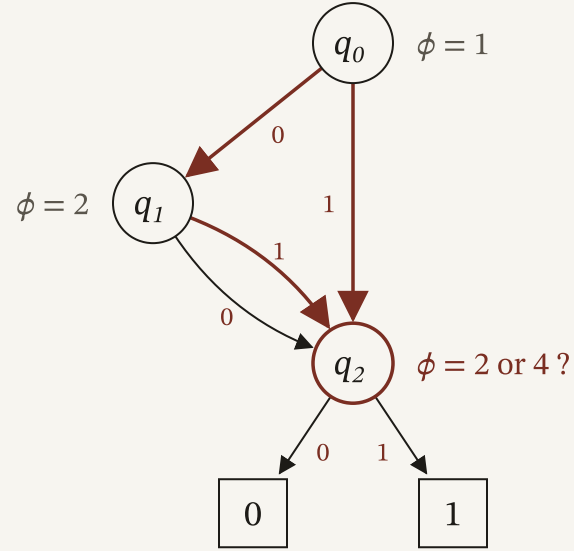
$$\delta(q_0) = (q_1, q_2), \quad \delta(q_1) = (q_2, q_2), \quad \delta(q_2) = (0, 1)$$

$$\text{val} = \text{zip}(0011, 01) = 00011$$

Synchronous and non-synchronous zipline programs



synchronous: step 2^j at level j



not synchronous: paths to q_2 have steps 2 and 4

ARC and synchronous zipline complexity

PROPOSITIONS 4.9, 4.10 (II)

For every base $k \geq 2$, $n \geq 2$ and $x \in \Sigma^n$,

$$\text{ARC}(x) \leq \text{sZLC}(x) \leq k \lceil \log_k n \rceil \cdot \text{AC}_k^{\text{R}}(x)$$

zip is to AC_k^{R} what concatenation is to AC_k^{L}

Mix-automatic sequences

ENDRULLIS–GRABMAYER–HENDRIKS 2013; DEFINITIONS F.3, F.4 (II)

A *mix-automaton* is $M = (Q, q_0, \delta, \tau)$ with $\delta : Q \rightarrow Q^*$, $\tau : Q \rightarrow \Sigma$ such that

- $d_q := |\delta(q)| \geq 2$
- $\tau(\delta(q)[0]) = \tau(q)$

$$\text{val}_M(q)[0] := \tau(q)$$

$$\text{val}_M(q)[d_q t + i] := \text{val}_M(\delta(q)[i])[t] \quad (i < d_q)$$

x is mix-automatic if $x = \text{val}_M(q_0)$ for some mix-automaton M

digits are read least significant first, in a base that depends on the state

k -automatic: $d_q = k$ for all q

Synchronous zipline complexity of mix-automatic sequences

Synchronization rank: $\text{syrk}(M) := \dim_{\mathbb{Q}} \text{span}_{\mathbb{Q}} \{ w(C) : C \text{ a cycle in } M \}$, $w(C) := \sum_j \ln d_{C[j]}$

PROPOSITION F.8 (II)

For a mix-automaton M , $x := \text{val}(M)$, $r := \text{syrk}(M)$ and every $n \geq 2$,

$$\text{sZLC}(x[:n]) = O((\log n)^r)$$

so ARC-MUWU makes $\text{polylog } n$ mistakes on any mix-automatic sequence

Layered zipline programs

DEFINITION 5 (III)

A ZLP $H = (Q, q_0, \delta)$ is *layered* if there is $h : Q \rightarrow \mathbb{N}$ with

- $h(q_0) = 0$
- $q' \text{ in } \delta(q) \implies h(q') = h(q) + 1$
- $h(q) = h(q') \implies \text{ar}(q) = \text{ar}(q')$

Layers $Q_j := h^{-1}(j)$, arity profile $\text{ar}_H(j)$. H is R -adic if $\text{ar}_H(j) \in R$ for all j .

DEFINITION 7 (III)

Width $w(H) := \max_j |Q_j|$. For finite $R \subseteq \{2, 3, 4, \dots\}$:

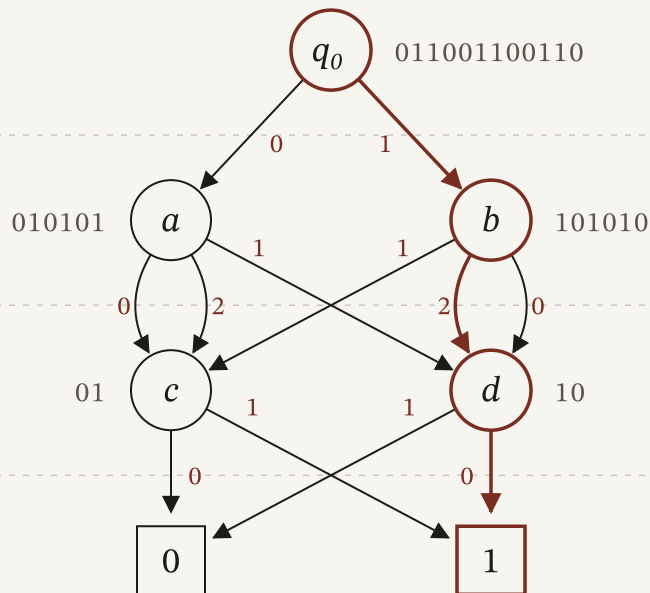
$$\text{LZLC}_R(x) := \min \{ w(H) : H \text{ } R\text{-adic layered, } x \sqsubseteq \text{val}(H) \}$$

A layered zipline program

Q_0 arity 2
 $m = 1$

Q_1 arity 3
 $m = 2$

Q_2 arity 2
 $m = 6$



layers and arities

$R = \{2, 3\}$, $\text{ar}_H = (2, 3, 2)$, $w(H) = 2$

reading position $t = 5$

$5 = 1 \cdot 1 + 2 \cdot 2 + 0 \cdot 6$

$q_0 \rightarrow_1 b \rightarrow_2 d \rightarrow_0 1$

$\text{val}(H)$

011001100110

Layered zipline complexity and other measures

COROLLARY 12 (III)

For finite nonempty $R \subseteq \{2, 3, 4, \dots\}$, $r := \max R$, $n \geq 2$ and $x \in \Sigma^n$,

$$\text{ARC}(x) \leq \text{sZLC}(x) \leq r \lceil \log n \rceil \cdot (\text{LZLC}_R(x) + \sigma)$$

COROLLARY 15 (III)

Let R be finite, $R!$ its multiplicative closure, $x \in \Sigma^*$ and $k \in R!$ with $k \geq 2$. Then

$$\text{LZLC}_R(x) \leq \frac{k}{2} \cdot \text{AC}_k^R(x)$$

Predicting w.r.t. layered zipline complexity

THEOREM 17 (III)

For finite nonempty $R \subseteq \{2, 3, 4, \dots\}$, the predictor P^3 is statistically and computationally efficient w.r.t. LZLC_R . For $n \geq 2$, $x \in \Sigma^n$, $s := \text{LZLC}_R(x)$, $r := \max R$, $\sigma := |\Sigma|$:

$$\text{mistakes} \quad O(r \log r \cdot (s + \sigma) \log(r(s + \sigma)) \cdot \log n \cdot \log(rn))$$

$$\text{state size} \quad O((s + \sigma) \cdot (|R| r \log(s + \sigma) + \log(rn)) \cdot (\log(rn))^{|R|})$$

open: can the exponential dependence on $|R|$ be avoided?