Kolmogorov structure functions for automatic complexity in computational statistics

Bjørn Kjos-Hanssen (University of Hawai'i at Mānoa)

December 21, 2014 – Combinatorial Optimization and Applications (COCOA)

History

- 1936: Universal Turing machine / programming language / computer

History

- 1936: Universal Turing machine / programming language / computer
- 1965: Kolmogorov complexity of a string $x = 0111001$, say, $=$ the length of the shortest program printing x

History

- 1936: Universal Turing machine / programming language / computer
- 1965: Kolmogorov complexity of a string $x = 0111001$, say, = the length of the shortest program printing x
- 1973: Structure function (Kolmogorov)

History

- 1936: Universal Turing machine / programming language / computer
- 1965: Kolmogorov complexity of a string $x = 0111001$, say, = the length of the shortest program printing x
- 1973: Structure function (Kolmogorov)
- 2001: Automatic complexity $A(x)$ (Shallit and Wang) (deterministic)

History

- 1936: Universal Turing machine / programming language / computer
- 1965: Kolmogorov complexity of a string $x = 0111001$, say, $=$ the length of the shortest program printing x
- 1973: Structure function (Kolmogorov)
- 2001: Automatic complexity $A(x)$ (Shallit and Wang) (deterministic)
- 2013: Nondeterministic automatic complexity $A_N(x)$ (Hyde, M.A. thesis, University of Hawai‘i): $n/2$ upper bound.

History

- 1936: Universal Turing machine / programming language / computer
- 1965: Kolmogorov complexity of a string $x = 0111001$, say, = the length of the shortest program printing x
- 1973: Structure function (Kolmogorov)
- 2001: Automatic complexity $A(x)$ (Shallit and Wang) (deterministic)
- 2013: Nondeterministic automatic complexity $A_N(x)$ (Hyde, M.A. thesis, University of Hawai‘i): $n/2$ upper bound.
- 2014: Structure function for automatic complexity: entropy-based upper bound.

History

- 1936: Universal Turing machine / programming language / computer
- 1965: Kolmogorov complexity of a string $x = 0111001$, say, = the length of the shortest program printing x
- 1973: Structure function (Kolmogorov)
- 2001: Automatic complexity $A(x)$ (Shallit and Wang) (deterministic)
- 2013: Nondeterministic automatic complexity $A_N(x)$ (Hyde, M.A. thesis, University of Hawai‘i): $n/2$ upper bound.
- 2014: Structure function for automatic complexity: entropy-based upper bound.

Definition

- Let M be an NFA having q states and no $\epsilon-$transitions. If there is exactly one path through M of length $|x|$ leading to an accept state, and x is the string read along the path, then we say $A_N(x) \leq q$.

Definition

- Let M be an NFA having q states and no $\epsilon-$transitions. If there is exactly one path through M of length $|x|$ leading to an accept state, and x is the string read along the path, then we say $A_N(x) \leq q$.

- Such an NFA is said to witness the complexity of x being no more than q.

Upper Bound

Theorem 1

Let $|\Sigma| = k \geq 2$ be fixed and suppose $x \in \Sigma^n$. Then $A_N(x) \leq \frac{n}{2} + 1$.

Upper Bound

Theorem 1

Let $|\Sigma| = k \geq 2$ be fixed and suppose $x \in \Sigma^n$. Then $A_N(x) \leq \frac{n}{2} + 1$.

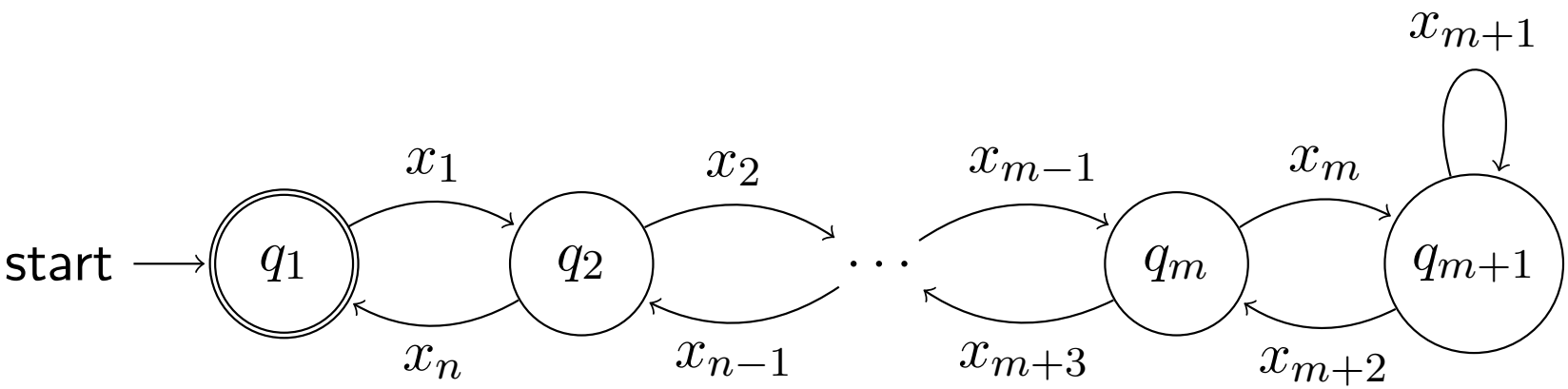

Figure 1 : An NFA uniquely accepting $x = x_1 x_2 \dots x_n$, $n = 2m + 1$

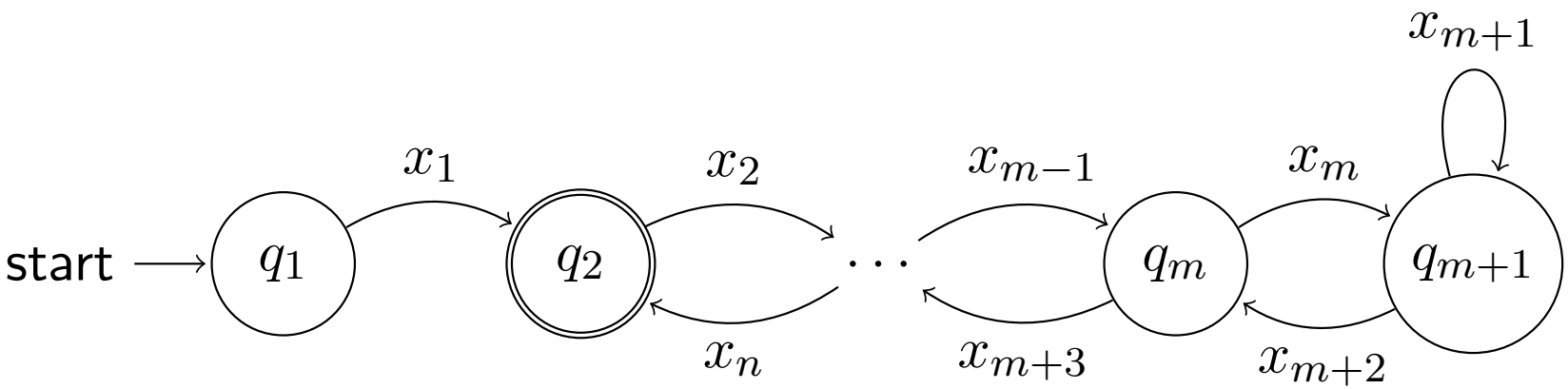

Figure 2 : An NFA uniquely accepting $x = x_1 x_2 \dots x_n$, $n = 2m$

Examples:

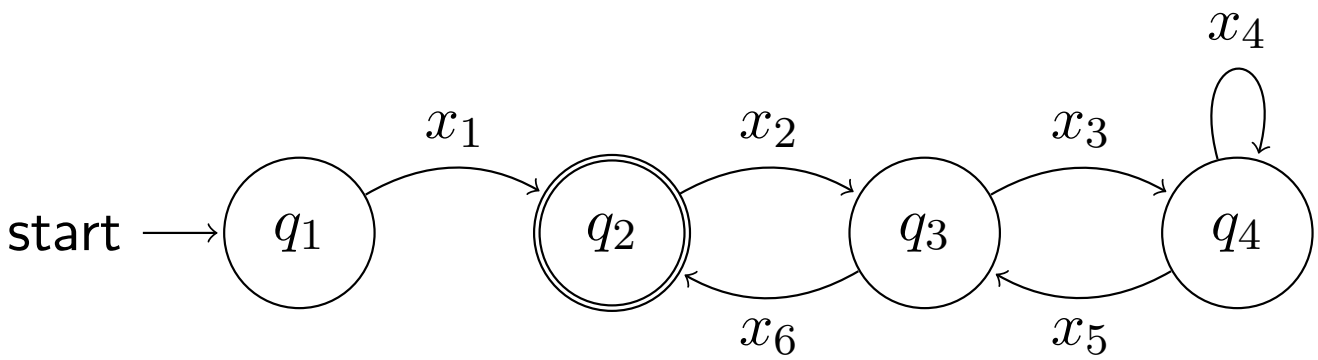

Figure 3 : An NFA uniquely accepting $x = x_1x_2x_3x_4x_5x_6$. $A_N(x) \leq 4$.

Examples:

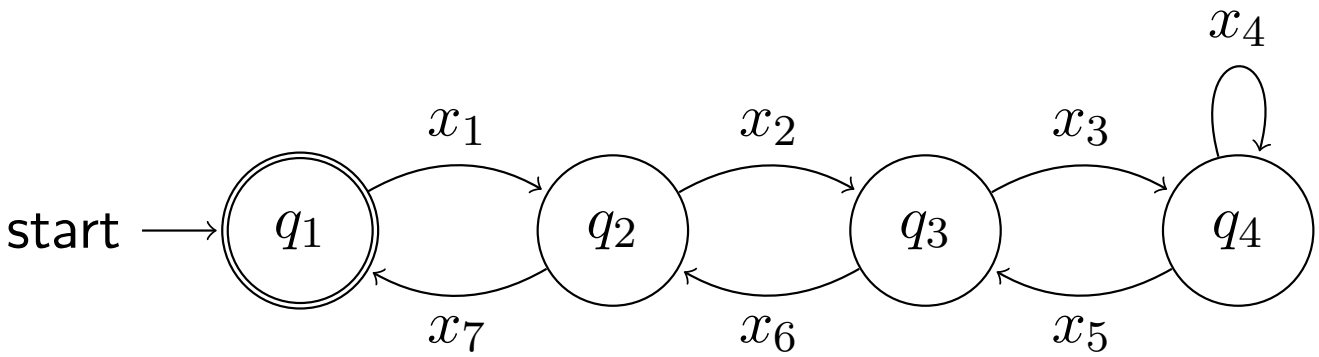

Figure 4 : An NFA uniquely accepting $x = x_1x_2x_3x_4x_5x_6x_7$. $A_N(x) \leq 4$.

Examples:

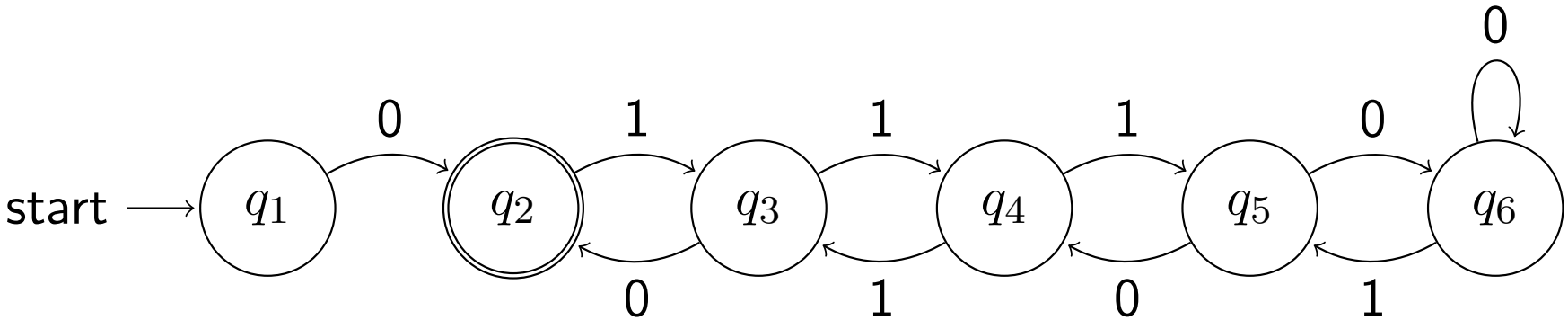

Figure 5 : An NFA uniquely accepting $x = 0111001010$, $|x| = 10$, $A_N(x) \leq 6$.

Examples:

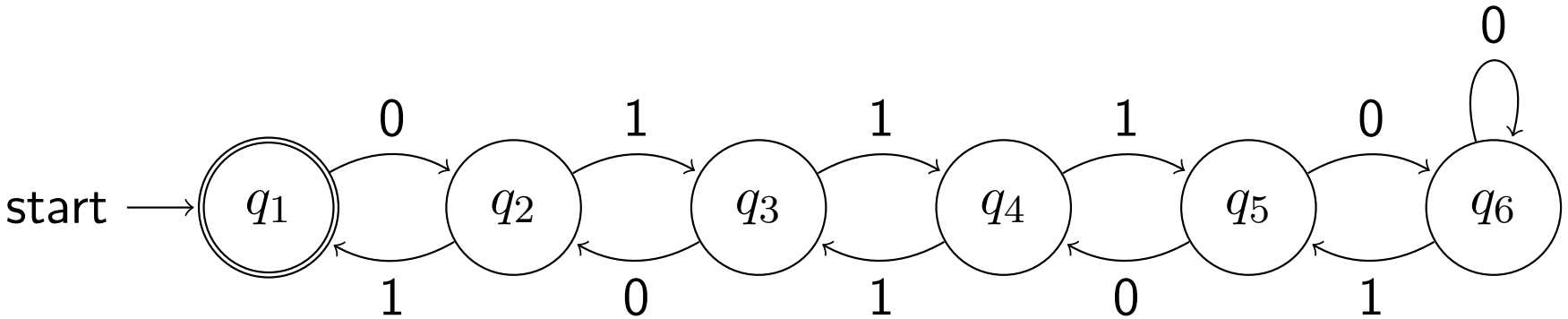

Figure 6 : An NFA uniquely accepting $x = 01110010101$, $|x| = 11$, $A_N(x) \leq 6$.

The Kolmogorov complexity of a finite word w is roughly speaking the length of the shortest description w^* of w in a fixed formal language. The description w^* can be thought of as an optimally compressed version of w. Motivated by the non-computability of Kolmogorov complexity, Shallit and Wang studied a deterministic finite automaton analogue.

Definition 1 (Shallit and Wang [1])

The automatic complexity of a finite binary string $x = x_1 \dots x_n$ is the least number $A_D(x)$ of states of a deterministic finite automaton M such that x is the only string of length n in the language accepted by M.

This complexity notion has two minor deficiencies:

1. Most of the relevant automata end up having a "dead state" whose sole purpose is to absorb any irrelevant or unacceptable transitions.
2. The complexity of a string can be changed by reversing it. For instance,

$$A_D(011100) = 4 < 5 = A_D(001110).$$

Definition 2

The nondeterministic automatic complexity $A_N(w)$ of a word w is the minimum number of states of an NFA M, having no ϵ-transitions, accepting w such that there is only one accepting path in M of length $|w|$.

Definition 3

The complexity deficiency of a word x of length n is

$$D_n(x) = D(x) = b(n) - A_N(x).$$

Length n	$\mathbb{P}(D_n > 0)$
0	0.000
2	0.500
4	0.500
6	0.531
8	0.617
10	0.664
12	0.600
14	0.687
16	0.657
18	0.658
20	0.641
22	0.633
24	0.593

(a) Even lengths.

Length n	$\mathbb{P}(D_n > 0)$
1	0.000
3	0.250
5	0.250
7	0.234
9	0.207
11	0.317
13	0.295
15	0.297
17	0.342
19	0.330
21	0.303
23	0.322
25	0.283

(b) Odd lengths.

Table 1 : Probability of strings of having positive complexity deficiency D_n, truncated to 3 decimal digits.

Definition 4

Let $\mathrm{DEFICIENCY}$ *be the following decision problem.*
Given a binary word w *and an integer* $d \geq 0$*, is* $D(w) > d$*?*

Theorem 5

$\mathrm{DEFICIENCY}$ *is in NP.*

We do not know whether $\mathrm{DEFICIENCY}$ is NP-complete.

Let

$$S_x = \{(q, m) \mid \exists\ q\text{-state NFA } M, x \in L(M) \cap \Sigma^n, |L(M) \cap \Sigma^n| \leq b^m\}.$$

Then S_x has the upward closure property

$$q \leq q', m \leq m', (q, m) \in S_x \quad \Longrightarrow \quad (q', m') \in S_x.$$

Definition 6 (Vereshchagin, personal communication, 2014)

In an alphabet Σ containing b symbols, we define

$$h_x^*(m) = \min\{k : (k, m) \in S_x\} \quad \textit{and}$$

$$h_x(k) = \min\{m : (k, m) \in S_x\}.$$

Definition 7

The entropy function $\mathcal{H} : [0,1] \to [0,1]$ *is given by*

$$\mathcal{H}(p) = -p \log_2 p - (1-p) \log_2 (1-p).$$

A useful well-known fact:

Theorem 8

For $0 \leq k \leq n$,

$$\log_2 \binom{n}{k} = \mathcal{H}(k/n)n + O(\log n).$$

Definition 9

Let

$$\tilde{h}(p) = \limsup_{n\to\infty} \max_{|x|=n} \frac{h_x([p\cdot n])}{n}, \quad \tilde{h} : [0, 1/2] \to [0, 1]$$

where $[x]$ *is the nearest integer to* x*.*

We are interested in upper bounds on $\tilde{h}$.

$$y=\begin{cases} \min\left(2-\frac{2\ln(2+\sqrt{3})}{\ln(2)}x, 1-x\right) & \frac{\sqrt{3}}{4}>x \wedge x\geq 0 \\ \frac{\left(-\left(\frac{1}{2}-x\right)\ln\left(\frac{1}{2}-x\right)-\left(\frac{1}{2}+x\right)\ln\left(\frac{1}{2}+x\right)\right)}{\ln(2)} & \frac{\sqrt{3}}{4}\leq x \end{cases}$$

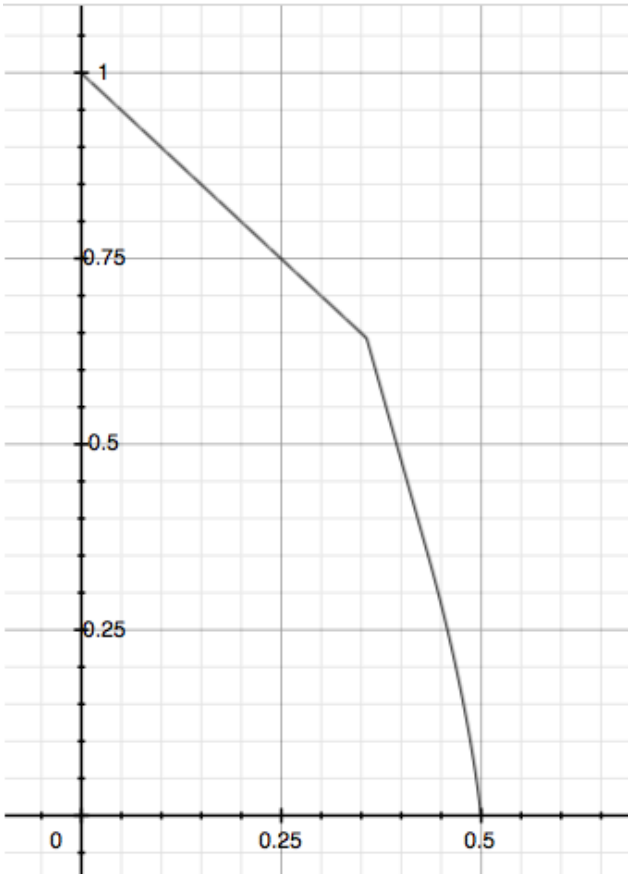

Figure 7 : Bounds for the automatic structure function for alphabet size $b = 2$.

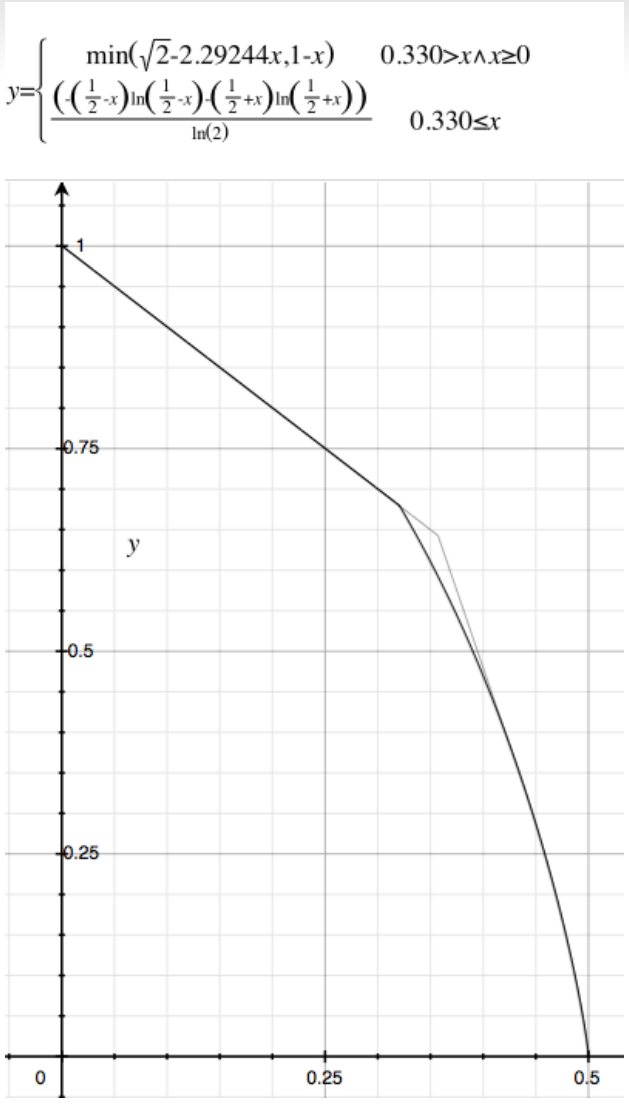

Figure 8 : Bounds for the automatic structure function for alphabet size $b = 2$ with improved bound.

Complexity guessing game

http://math.hawaii.edu/play

References I

Jeffrey Shallit and Ming-Wei Wang.
Automatic complexity of strings.
J. Autom. Lang. Comb., 6(4):537–554, 2001.
2nd Workshop on Descriptional Complexity of Automata, Grammars and Related Structures (London, ON, 2000).