The Shannon effect

Bjørn Kjos-Hanssen

May 14, 2025
ASL annual meeting

Shannon effect

- *almost all Boolean functions have nearly the same circuit complexity as the hardest function*
- *'most objects are almost maximally complicated*

Intuition

Random numbers between 0 and 1000000. Are any of them "simple"?

- 549313
- 502773
- 633402
- 141214
- 559494

Let $\mathcal{B}_n = \{0,1\}^{\{0,1\}^n}$ the class of Boolean functions $f : \{0,1\}^n \to \{0,1\}$.

Definition (Wegener [Weg87, 4.1.1])

Almost all functions f of a class $\mathcal{F}_n \subseteq \mathcal{B}_n$ have property P

means:

$$\lim_{n\to\infty} \frac{|\{f \in \mathcal{F}_n \mid f \text{ has } P\}|}{|\mathcal{F}_n|} = 1.$$

Definition (Wegener [Weg87, 4.1.2])

For a complexity measure CM and a class of functions $\mathcal{F}_n$,

$$\mathrm{CM}(\mathcal{F}_n) = \max_{f\in\mathcal{F}_n} \mathrm{CM}(f).$$

Definition (Wegener [Weg87, 4.1.3])

The *Shannon effect* is valid for a class of functions $\mathcal{F}_n$ and a complexity measure CM if

$$\mathrm{CM}(f) \geq \mathrm{CM}(\mathcal{F}_n) - o(\mathrm{CM}(\mathcal{F}_n))$$

for almost all $f \in \mathcal{F}_n$.

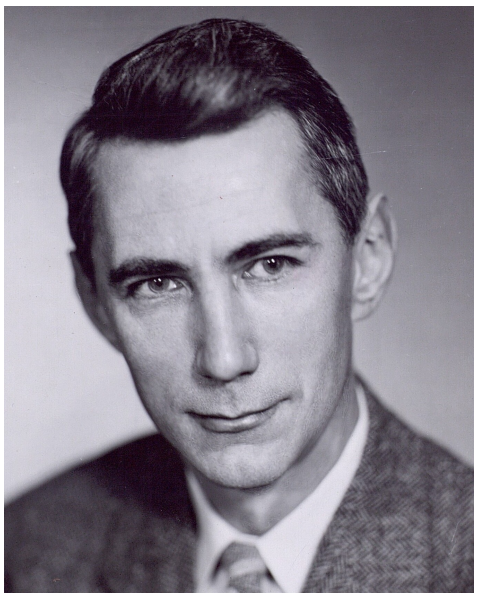
(Presumably it means that for each $\varepsilon > 0$,

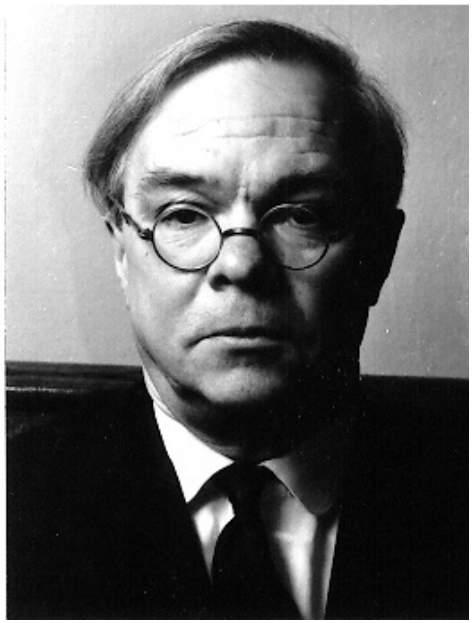
$$\frac{\mathrm{CM}(\mathcal{F}_n) - \mathrm{CM}(f)}{\mathrm{CM}(\mathcal{F}_n)} \leq \varepsilon$$

holds for almost all f.)

History

- Shannon (1916–2001) [Sha49] conjectures the effect
- Lupanov (1932–2006) [Lup58] proves and [Lup70] names the effect

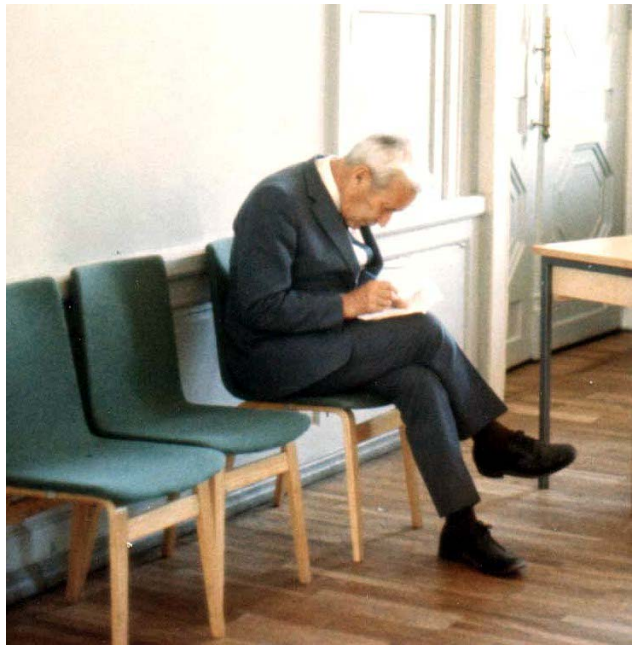




Example: Kolmogorov complexity

$C(w)$ =the length of the shortest program that outputs w. This is machine-dependent but only up to an additive constant.

Kolmogorov (1903–1987) [Kol65] proved $C(w) \leq |w| + O(1)$.



Kolmogorov preparing for a lecture.

Solomonoff (1926–2009) [Sol64a, Sol64b] and Kolmogorov observed $|\{w \in \{0,1\}^n : C(w) \geq n - k\}| > 2^n(1 - 2^{-k})$.

Prefix-free Kolmogorov complexity

Definition

$K(w)$ =shortest program outputting w for a programming language where no prefix of a valid program is valid.

Theorem (Levin 1971/1974)

$K(w) \leq |w| + 2\log|w| + O(1)$ *and* $C(w) \leq K(w) + O(1)$.

The theorem implies the Shannon effect for K as well.

> *But Kolmogorov complexity is not a computable function, so we turn to...*

Finite automata

A deterministic finite automaton (DFA) is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$ where

- Q is a finite set of *states*;
- Σ is a finite *alphabet*;
- $\delta : Q \times \Sigma \to Q$ is a *transition function*;
- $q_0 \in Q$ is an *initial state*;
- $F \subseteq Q$ is a set of *final states*.

Language recognized by DFA

Let ε be the empty word. We define $\delta^* : Q \times \Sigma^* \to Q$ by recursion:

$$\begin{aligned} \delta^*(q, \varepsilon) &= q \\ \delta^*(q, xa) &= \delta(\delta^*(q, x), a) \end{aligned}$$

for $x \in \Sigma^*$ and $a \in \Sigma$.

$$L(M) = \{x : \delta^*(q_0, x) \in F\}$$

is the language recognized by M.

Automatic complexity of distinguishing

Definition (Shallit and Wang 2001)

Let $x \in \{0,1\}^n$. The automatic complexity $A(x)$ of x is the least $|Q|$ over all DFAs M with $L(M) \cap \{0,1\}^n = \{x\}$.

Example

$A(\texttt{0101010101}) = 3$;
$A(\texttt{01}) = 2$.

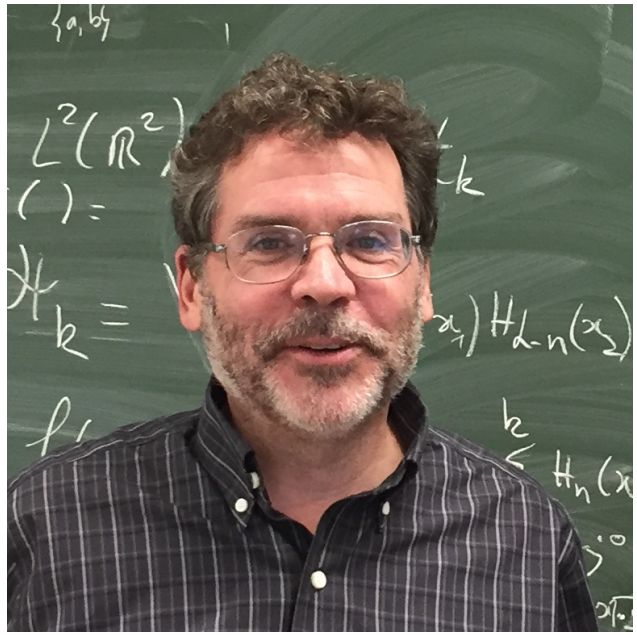
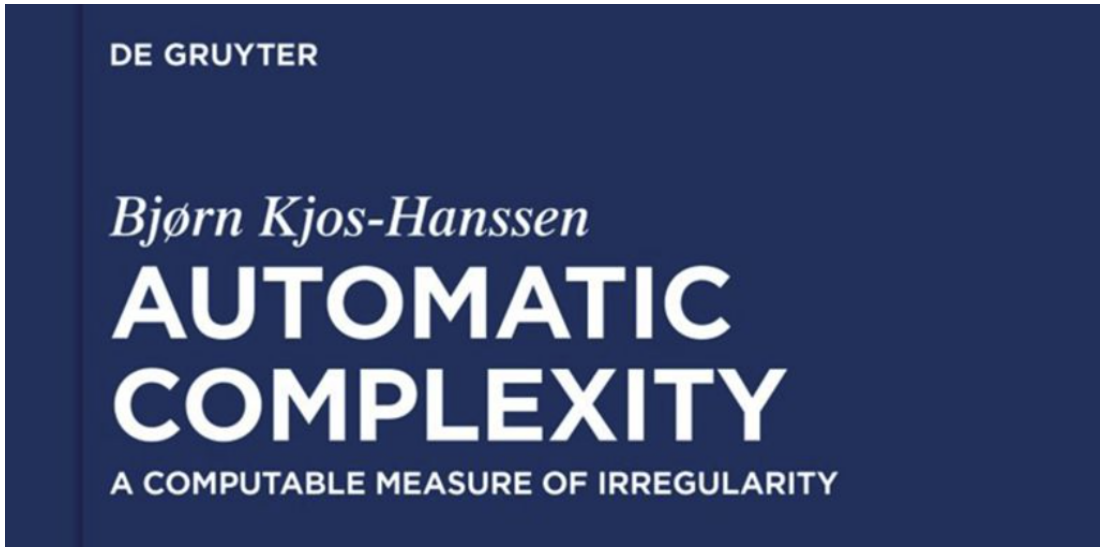


Figure: Jeff Shallit

Automatic complexity of distinguishing

De Gruyter Series in Logic and its Applications (2024)



All exercises come with Lean solutions (but no English solutions)

Automatic complexity

Shallit and Wang proved that for almost all x of any sufficiently large length n, $A(x) \leq (3/4)n + (\log n)(n/8)^{1/2}$ and $A(x) \geq n/13$.

Open problem: Is there a Shannon effect for automatic complexity?

It turns out to be easier to understand a more complicated notion...

Nondeterminism

A nondeterministic finite automaton (DFA) is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$ where

- Q is a finite set of *states*;
- Σ is a finite *alphabet*;
- $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ is a *transition function*;
- $q_0 \in Q$ is an *initial state*;
- $F \subseteq Q$ is a set of *final states*.

Language recognized by NFA

Let ε be the empty word. We define $\delta^* : Q \times \Sigma^* \to \mathcal{P}(Q)$ by

$$\begin{aligned} \delta^*(q, \varepsilon) &= \{q\} \\ \delta^*(q, xa) &= \bigcup_{r \in \delta^*(q,x)} \delta(r, a) \end{aligned}$$

for $x \in \Sigma^*$ and $a \in \Sigma$.

$$L(M) = \{x : \delta^*(q_0, x) \in F\}$$

is the language recognized by M.

State sequences

Let M be an NFA. An *accepting state sequence* for $x = x_0 \dots x_{n-1}$ in M, where $x_i \in \Sigma$ for $0 \leq i < n$, is a sequence $(q_0, \dots, q_n)$ where $q_i \in Q$ and

$$q_{i+1} \in \delta^*(q_i, x_i)$$

for each $0 \leq i \leq n$, and $q_n \in F$.
Let $P(x, M)$ be the set of accepting state sequences for x in M.

Nondeterministic automatic complexity A_N

We say that an NFA M uniquely accepts x if

- $|P(x, M)| = 1$;
- $|P(x, M)| = 0$ for all $y \neq x$, $|y| = |x|$.

Definition

$A_N(x)$ is the minimum of $|Q|$ over all NFAs M which uniquely accept x.

Equivalently (?)...

Definition

$A_{Ne}(x)$ is the minimum of $|Q|$ over all NFAs M with $L(M) \cap \{0,1\}^{|x|} = \{x\}$.

The e in Ne stands for “exactly accepts”.

Lemma

$A_{Ne}(x) \leq A_N(x)$.

Since "only one path implies only one word".

Robustness question

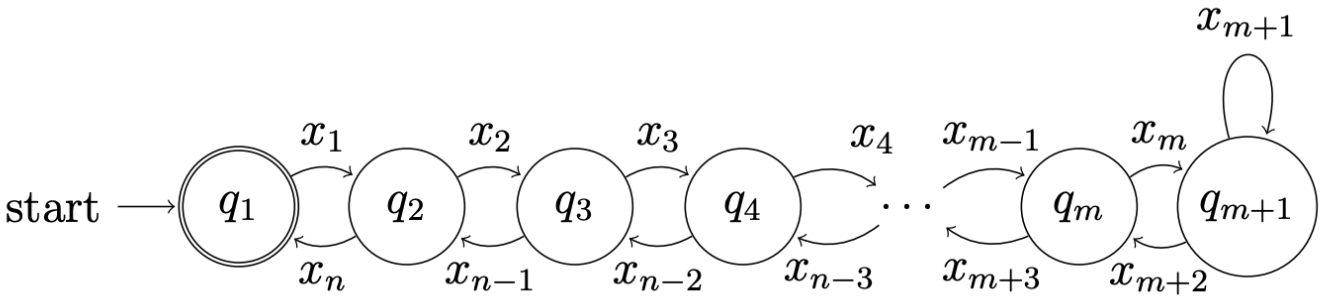
- A_{Ne} is a direct analogue of A;
- A_N is easier to compute in practice*.

*Python

Question

Is $A_N = A_{Ne}$*?*

Shannon effect for A_N

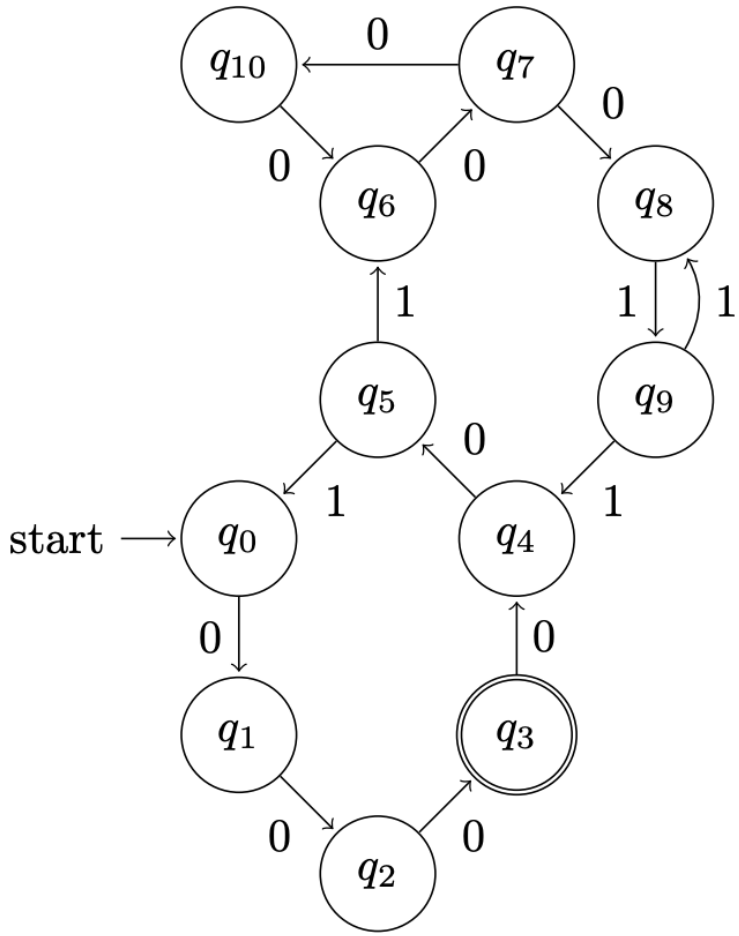


Kayleigh Hyde, MA Mathematics from UH, 2013

- Hyde 2013: For all words x, $A_N(x) \leq |x|/2 + 1$.
- K. 2021: $A_N(x) \geq |x|/(2 + \varepsilon)$ for almost all words x.

Understanding A_N

The witnessing automata will always be "forests of cycles" laid out in Kleene–Brouwer order. Example for $0^5 10^5 1^6 0 10^3$



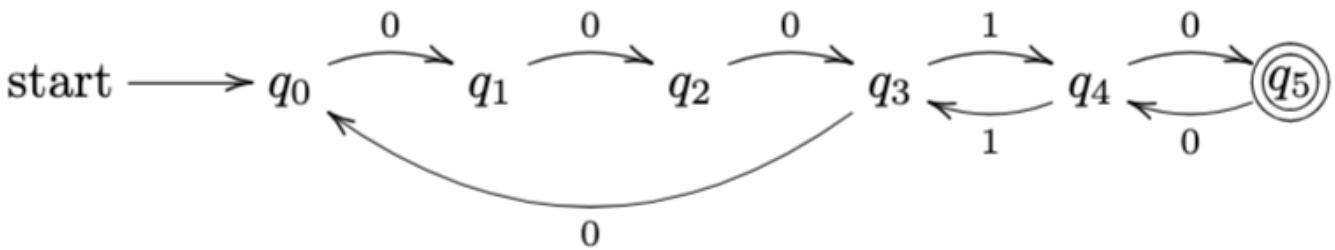
Permutation automatic complexity

Definition

$A^{\text{perm}}(x)$ is the minimum number of states of a DFA accepting x and no other word of length $|x|$ — and such that each function $q \mapsto \delta(a, q)$ is bijective.

Example

$A^{\text{perm}}(00010) \leq 6$ as witnessed by this DFA (missing edges are self-loops):



Permutation automatic "complexity"

- K. 2017: $A^{\mathrm{perm}}(w) \leq |w| + 1$ (using construction that loops back to conclude a cycle) (upon formalization, realized it should be viewed as: loops back to maintain injectivity)
- Anthony Quas 2020: $A^{\mathrm{perm}}(w) \geq |w| + 1$ ("this proof somewhat resembles that of the Morse-Hedlund theorem in symbolic dynamics.")
- Shannon effect is trivial: $A^{\mathrm{perm}}(w) = |w| + 1$ for all w.

Formalization in Lean

I formalized Hyde's theorem. The struggles of doing so revealed that the construction is best viewed via the idea of *the NFA generated by a path* rather than directly as an NFA.

```
/-- Hyde's theorem (2013). -/
theorem A_N_bound {A : Type*} {n : ℕ} (w : Fin n → A) :
  A_N w ≤ n/2+1 := find_le <| hyde_all_lengths w
```

Lean details

```
/-- The transition function δ generated by a labeled path.
δ b r = {s | s is reachable in one step from r reading b}.
-/
def δ_of_path {A Q : Type*} {n : ℕ} (w : Fin n → A)
    (p : Fin (n+1) → Q) (b : A) (r : Q) :=
  {s | ∃ t : Fin n,
    p t.castSucc = r ∧ p t.succ = s ∧ w t = b}
```

```
/-- Kayleigh Hyde's transition function δ. -/
def khδ' {A : Type*} {k : ℕ} (w : Fin (2*k+1) → A) :
    A → Fin (k+1) → Set (Fin (k+1)) :=
  δ_of_path w fun t =>
    dite (t.1 < k + 1) (⟨t.1, .⟩)
    (⟨2 * k + 1 - t.1, flipCast .⟩)
```

Things that need a proof...

If $f : [k] \to [k]$ *with* $f(0) = 0$ *and* $f(k) = k$ *and* $f(x+1) \leq f(x) + 1$ *for all* x*, then* f *is the identity.*

```
/-- Discrete Racecar Principle. -/
lemma exact_racecar {k : ℕ} {f : Fin (k+1) → Fin (k+1)}
    (h₀ : f 0 = 0) (hk : f (Fin.last k) = (Fin.last k))
    (h : ∀ (s : Fin k),
      (f s.succ).1
    ≤ (f s.castSucc).1 + 1) {a : Fin k} :
    f a.castSucc = a.castSucc
```

Recent news about A_{Ne}

Chen, K., Koswara, Richter, Stephan (2025), submitted [CKHK$^+$25] prove the Shannon effect also for A_{Ne}.

4 of the authors are based in Singapore

Conclusions

- The Shannon effect usually holds but a precise statement to the effect that it "always" does is missing.
- Formalizing mathematics can help reveal its structure.

References I


Joey Chen, Bjørn Kjos-Hanssen, Ivan Koswara, Linus Richter, and Frank Stephan.
Languages of words of low automatic complexity are hard to compute, 2025.
Submitted.


A. N. Kolmogorov.
Three approaches to the definition of the concept "quantity of information".
Problemy Peredači Informacii, 1(1):3–11, 1965.


O. B. Lupanov.
The synthesis of contact circuits.
Dokl. Akad. Nauk SSSR (N.S.), 119:23–26, 1958.


O. B. Lupanov.
The schemes of functional elements with delays.
Problemy Kibernet., (23):43–81, 303, 1970.

References II


Claude E. Shannon.
The synthesis of two-terminal switching circuits.
Bell System Tech. J., 28:59–98, 1949.


R. J. Solomonoff.
A formal theory of inductive inference. I.
Information and Control, 7:1–22, 1964.


R. J. Solomonoff.
A formal theory of inductive inference. II.
Information and Control, 7:224–254, 1964.


Ingo Wegener.
The complexity of Boolean functions.
Wiley-Teubner Series in Computer Science. John Wiley & Sons, Ltd., Chichester; B. G. Teubner, Stuttgart, 1987.