Effective Complexity

Markus Müller*, Arleta Szkoła*, Nihat Ay*†

*Max Planck Institute for Mathematics in the Sciences, Leipzig
†External Faculty, Santa Fe Institute

MPI Leipzig, October 2008

Outline of the talk:

Outline of the talk:

- 1. Definition of Effective Complexity

Outline of the talk:

- 1. Definition of Effective Complexity
- 2. Random strings are effectively simple

Outline of the talk:

- 1. Definition of Effective Complexity
- 2. Random strings are effectively simple
- 3. Existence of effectively complex strings

Outline of the talk:

- 1. Definition of Effective Complexity
- 2. Random strings are effectively simple
- 3. Existence of effectively complex strings
- 4. Effective Complexity and Logical Depth

Algorithmic Information Content (AIC):

Algorithmic Information Content (AIC):

If U is a universal computer, mapping binary strings $\{0,1\}^* = \{\lambda, 0, 1, 00, 01, \ldots\}$ to binary strings, then

$$C(x) := \min\{\ell(p) \mid U(p) = x\}$$

is the "algorithmic information content" of x, also called "Kolmogorov complexity" of the string x.

Algorithmic Information Content (AIC):

If U is a universal computer, mapping binary strings $\{0,1\}^* = \{\lambda, 0, 1, 00, 01, \ldots\}$ to binary strings, then

$$C(x) := \min\{\ell(p) \mid U(p) = x\}$$

is the "algorithmic information content" of x, also called "Kolmogorov complexity" of the string x.

Kolmogorov complexity $C(x)$ (and its prefix-free version $K(x)$)

Algorithmic Information Content (AIC):

If U is a universal computer, mapping binary strings $\{0,1\}^* = \{\lambda, 0, 1, 00, 01, \ldots\}$ to binary strings, then

$$C(x) := \min\{\ell(p) \mid U(p) = x\}$$

is the "algorithmic information content" of x, also called "Kolmogorov complexity" of the string x.

Kolmogorov complexity $C(x)$ (and its prefix-free version $K(x)$)

- have important applications in math and computer science,

Algorithmic Information Content (AIC):

If U is a universal computer, mapping binary strings $\{0,1\}^* = \{\lambda, 0, 1, 00, 01, \ldots\}$ to binary strings, then

$$C(x) := \min\{\ell(p) \mid U(p) = x\}$$

is the "algorithmic information content" of x, also called "Kolmogorov complexity" of the string x.

Kolmogorov complexity $C(x)$ (and its prefix-free version $K(x)$)

- have important applications in math and computer science,
- are small if x is simple: if $x = \underbrace{0101\ldots01}_{n}$ then $K(x) \approx \log n$.

Algorithmic Information Content (AIC):

If U is a universal computer, mapping binary strings $\{0,1\}^* = \{\lambda, 0, 1, 00, 01, \ldots\}$ to binary strings, then

$$C(x) := \min\{\ell(p) \mid U(p) = x\}$$

is the "algorithmic information content" of x, also called "Kolmogorov complexity" of the string x.

Kolmogorov complexity $C(x)$ (and its prefix-free version $K(x)$)

- have important applications in math and computer science,
- are small if x is simple: if $x = \underbrace{0101\ldots01}_{n}$ then $K(x) \approx \log n$.
- are large if x is random: if $x = \underbrace{01001010\ldots}_{n}$ (fair coin tosses) then $K(x) \approx n$.

Algorithmic Information Content (AIC):

If U is a universal computer, mapping binary strings $\{0,1\}^* = \{\lambda, 0, 1, 00, 01, \ldots\}$ to binary strings, then

$$C(x) := \min\{\ell(p) \mid U(p) = x\}$$

is the "algorithmic information content" of x, also called "Kolmogorov complexity" of the string x.

Kolmogorov complexity $C(x)$ (and its prefix-free version $K(x)$)

- have important applications in math and computer science,
- are small if x is simple: if $x = \underbrace{0101\ldots01}_{n}$ then $K(x) \approx \log n$.
- are large if x is random: if $x = \underbrace{01001010\ldots}_{n}$ (fair coin tosses)

  then $K(x) \approx n$. $\Rightarrow$ K **is not really a complexity measure!**

Effective Complexity (M. Gell-Mann & S. Lloyd)

Effective Complexity (M. Gell-Mann & S. Lloyd)

Idea: Instead of the AIC $K(x)$, define the effective complexity of x as the AIC of its regularities ($\longrightarrow$ discard the random aspects).

Effective Complexity (M. Gell-Mann & S. Lloyd)

Idea: Instead of the AIC $K(x)$, define the effective complexity of x as the AIC of its regularities ($\longrightarrow$ discard the random aspects).

- Regularities in x: a computable ensemble $\mathbb{E}$ of strings, describing a process that possibly generated x.

Effective Complexity (M. Gell-Mann & S. Lloyd)

Idea: Instead of the AIC $K(x)$, define the effective complexity of x as the AIC of its regularities ($\longrightarrow$ discard the random aspects).

- Regularities in x: a computable ensemble $\mathbb{E}$ of strings, describing a process that possibly generated x.
- AIC of x's regularities: $K(\mathbb{E}) =: \mathcal{E}(x)$.

Effective Complexity (M. Gell-Mann & S. Lloyd)

Idea: Instead of the AIC $K(x)$, define the effective complexity of x as the AIC of its regularities ($\rightarrow$ discard the random aspects).

- Regularities in x: a computable ensemble $\mathbb{E}$ of strings, describing a process that possibly generated x.
- AIC of x's regularities: $K(\mathbb{E}) =: \mathcal{E}(x)$.
- Random part of x: identification of x inside the ensemble $\mathbb{E}$ $\rightarrow$ entropy $H(\mathbb{E}) := -\sum_{x\in\{0,1\}^*} \mathbb{E}(x) \log \mathbb{E}(x)$.

Effective Complexity (M. Gell-Mann & S. Lloyd)

Idea: Instead of the AIC $K(x)$, define the effective complexity of x as the AIC of its regularities ($\rightarrow$ discard the random aspects).

- Regularities in x: a computable ensemble $\mathbb{E}$ of strings, describing a process that possibly generated x.
- AIC of x's regularities: $K(\mathbb{E}) =: \mathcal{E}(x)$.
- Random part of x: identification of x inside the ensemble $\mathbb{E}$ $\rightarrow$ entropy $H(\mathbb{E}) := -\sum_{x\in\{0,1\}^*} \mathbb{E}(x) \log \mathbb{E}(x)$.

Example: $x = \underbrace{011010001001011101\ldots}_{n \text{ bits}}$ from fair coin tosses.

Effective Complexity (M. Gell-Mann & S. Lloyd)

Idea: Instead of the AIC $K(x)$, define the effective complexity of x as the AIC of its regularities ($\rightarrow$ discard the random aspects).

- Regularities in x: a computable ensemble $\mathbb{E}$ of strings, describing a process that possibly generated x.
- AIC of x's regularities: $K(\mathbb{E}) =: \mathcal{E}(x)$.
- Random part of x: identification of x inside the ensemble $\mathbb{E}$ $\rightarrow$ entropy $H(\mathbb{E}) := -\sum_{x\in\{0,1\}^*} \mathbb{E}(x) \log \mathbb{E}(x)$.

Example: $x = \underbrace{011010001001011101\ldots}_{n \text{ bits}}$ from fair coin tosses.

- Ensemble $\mathbb{E}$=uniform distribution on all strings of length n.

Effective Complexity (M. Gell-Mann & S. Lloyd)

Idea: Instead of the AIC $K(x)$, define the effective complexity of x as the AIC of its regularities ($\rightarrow$ discard the random aspects).

- Regularities in x: a computable ensemble $\mathbb{E}$ of strings, describing a process that possibly generated x.
- AIC of x's regularities: $K(\mathbb{E}) =: \mathcal{E}(x)$.
- Random part of x: identification of x inside the ensemble $\mathbb{E}$ $\rightarrow$ entropy $H(\mathbb{E}) := -\sum_{x\in\{0,1\}^*} \mathbb{E}(x) \log \mathbb{E}(x)$.

Example: $x = \underbrace{0110100010010111101\ldots}_{n \text{ bits}}$ from fair coin tosses.

- Ensemble $\mathbb{E}$=uniform distribution on all strings of length n.
- Effective Complexity $\mathcal{E}(x) = K(\mathbb{E}) \approx K(n) \lesssim \log n$

Effective Complexity (M. Gell-Mann & S. Lloyd)

Idea: Instead of the AIC $K(x)$, define the effective complexity of x as the AIC of its regularities ($\rightarrow$ discard the random aspects).

- Regularities in x: a computable ensemble $\mathbb{E}$ of strings, describing a process that possibly generated x.
- AIC of x's regularities: $K(\mathbb{E}) =: \mathcal{E}(x)$.
- Random part of x: identification of x inside the ensemble $\mathbb{E}$ $\rightarrow$ entropy $H(\mathbb{E}) := -\sum_{x\in\{0,1\}^*} \mathbb{E}(x) \log \mathbb{E}(x)$.

Example: $x = \underbrace{011010001001011101\ldots}_{n \text{ bits}}$ from fair coin tosses.

- Ensemble $\mathbb{E}$=uniform distribution on all strings of length n.
- Effective Complexity $\mathcal{E}(x) = K(\mathbb{E}) \approx K(n) \lessapprox \log n$ (cf. $K(x) \approx n + K(n)$).

Effective Complexity (M. Gell-Mann & S. Lloyd)

Idea: Instead of the AIC $K(x)$, define the effective complexity of x as the AIC of its regularities ($\rightarrow$ discard the random aspects).

- Regularities in x: a computable ensemble $\mathbb{E}$ of strings, describing a process that possibly generated x.
- AIC of x's regularities: $K(\mathbb{E}) =: \mathcal{E}(x)$.
- Random part of x: identification of x inside the ensemble $\mathbb{E}$ $\rightarrow$ entropy $H(\mathbb{E}) := -\sum_{x\in\{0,1\}^*} \mathbb{E}(x) \log \mathbb{E}(x)$.

Example: $x = \underbrace{0110100010010111 01\ldots}_{n \text{ bits}}$ from fair coin tosses.

- Ensemble $\mathbb{E}$=uniform distribution on all strings of length n.
- Effective Complexity $\mathcal{E}(x) = K(\mathbb{E}) \approx K(n) \lesssim \log n$ (cf. $K(x) \approx n + K(n)$).
- Randomness $H(\mathbb{E}) \approx n$.

Effective Complexity (M. Gell-Mann & S. Lloyd)

Example: $x = \underbrace{0110100010010111 01\ldots}_{n \text{ bits}}$ from fair coin tosses.

- Ensemble $\mathbb{E}$=uniform distribution on all strings of length n with number of zeroes ≈ number of ones.
- Effective Complexity $\mathcal{E}(x) = K(\mathbb{E}) \approx \log n$ (cf. $K(x) \approx n$).
- Randomness $H(\mathbb{E}) \approx n$.

Effective Complexity (M. Gell-Mann & S. Lloyd)

Example: $x = \underbrace{01101000100101101\ldots}_{n \text{ bits}}$ from fair coin tosses.

- Ensemble $\mathbb{E}$=uniform distribution on all strings of length n with number of zeroes ≈ number of ones.
- Effective Complexity $\mathcal{E}(x) = K(\mathbb{E}) \approx \log n$ (cf. $K(x) \approx n$).
- Randomness $H(\mathbb{E}) \approx n$.

But: What to do if x is given without knowing the process that generated it?

Effective Complexity (M. Gell-Mann & S. Lloyd)

Example: $x = \underbrace{011010001001011101\ldots}_{n \text{ bits}}$ from fair coin tosses.

- Ensemble $\mathbb{E}$=uniform distribution on all strings of length n with number of zeroes $\approx$ number of ones.
- Effective Complexity $\mathcal{E}(x) = K(\mathbb{E}) \approx \log n$ (cf. $K(x) \approx n$).
- Randomness $H(\mathbb{E}) \approx n$.

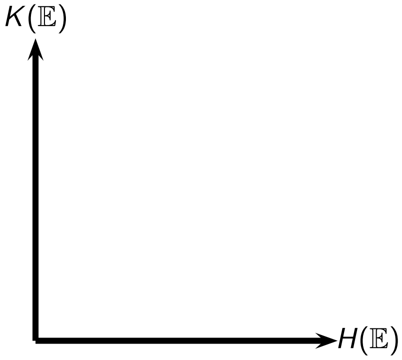
But: What to do if x is given without knowing the process that generated it?
How to decide which ensemble $\mathbb{E}$ to take?

How to decide which ensemble $\mathbb{E}$ to take for a given string x?

How to decide which ensemble $\mathbb{E}$ to take for a given string x?

There is a countably-infinite number of computable ensembles $\mathbb{E}$.

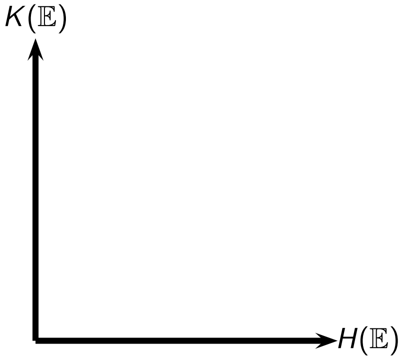
How to decide which ensemble $\mathbb{E}$ to take for a given string x?
There is a countably-infinite number of computable ensembles $\mathbb{E}$.



How to decide which ensemble $\mathbb{E}$ to take for a given string x?

There is a countably-infinite number of computable ensembles $\mathbb{E}$.

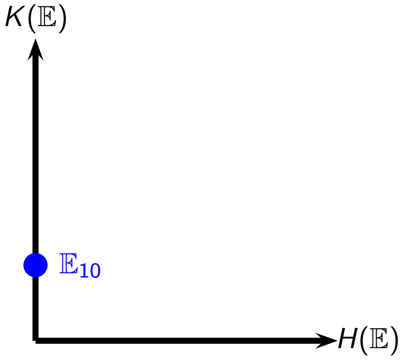
E.g. $\mathbb{E}_{10}(x) := \begin{cases} 1 & \text{if } x = 10 \\ 0 & \text{else} \end{cases}$



How to decide which ensemble $\mathbb{E}$ to take for a given string x?

There is a countably-infinite number of computable ensembles $\mathbb{E}$.

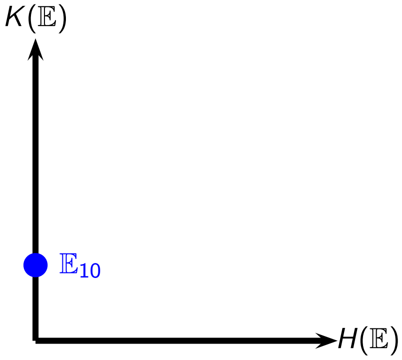
E.g. $\mathbb{E}_{10}(x) := \begin{cases} 1 & \text{if } x = 10 \\ 0 & \text{else} \end{cases}$



How to decide which ensemble $\mathbb{E}$ to take for a given string x?

There is a countably-infinite number of computable ensembles $\mathbb{E}$.

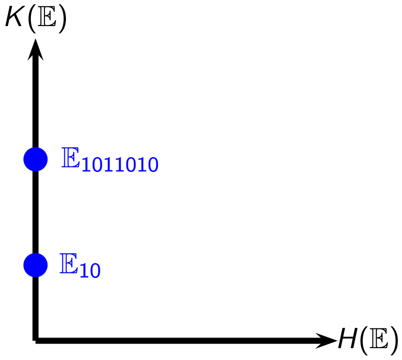
E.g. $\mathbb{E}_{1011010}(x) := \begin{cases} 1 & \text{if } x = 1011010 \\ 0 & \text{else} \end{cases}$



How to decide which ensemble $\mathbb{E}$ to take for a given string x?

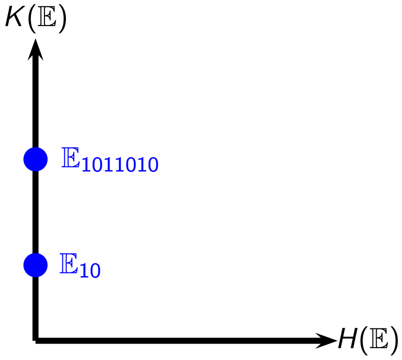
There is a countably-infinite number of computable ensembles $\mathbb{E}$.

E.g. $\mathbb{E}_{1011010}(x) := \begin{cases} 1 & \text{if } x = 1011010 \\ 0 & \text{else} \end{cases}$



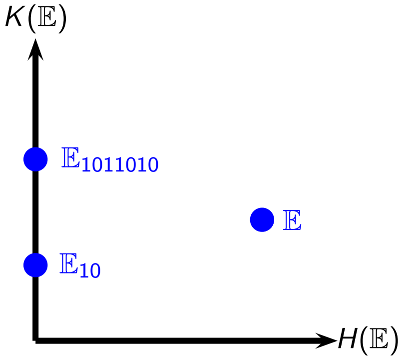
How to decide which ensemble $\mathbb{E}$ to take for a given string x?
There is a countably-infinite number of computable ensembles $\mathbb{E}$.

E.g. $\mathbb{E}$:=uniform distribution on all strings of length 42



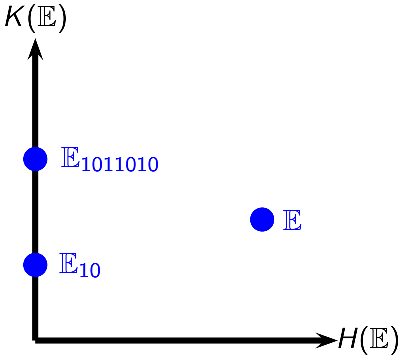
How to decide which ensemble $\mathbb{E}$ to take for a given string x?
There is a countably-infinite number of computable ensembles $\mathbb{E}$.

E.g. $\mathbb{E}$:=uniform distribution on all strings of length 42



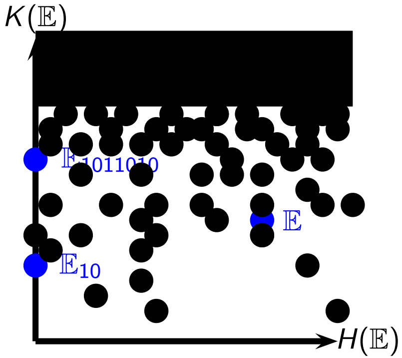
How to decide which ensemble $\mathbb{E}$ to take for a given string x?
There is a countably-infinite number of computable ensembles $\mathbb{E}$.

Plotting all computable $\mathbb{E}\ldots$



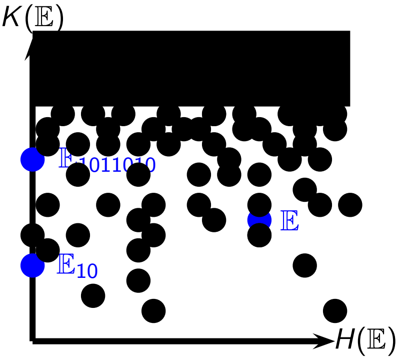
How to decide which ensemble $\mathbb{E}$ to take for a given string x?
There is a countably-infinite number of computable ensembles $\mathbb{E}$.

Plotting all computable $\mathbb{E}$...



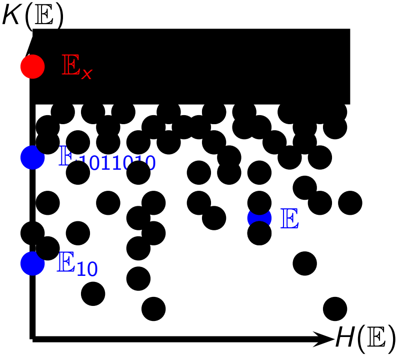
How to decide which ensemble $\mathbb{E}$ to take for a given string x?

One of them is $\mathbb{E}_x(s) := \begin{cases} 1 & \text{if } s = x, \\ 0 & \text{otherwise.} \end{cases}$



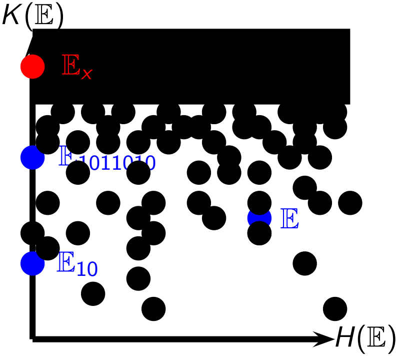
How to decide which ensemble $\mathbb{E}$ to take for a given string x?

One of them is $\mathbb{E}_x(s) := \begin{cases} 1 & \text{if } s = x, \\ 0 & \text{otherwise.} \end{cases}$



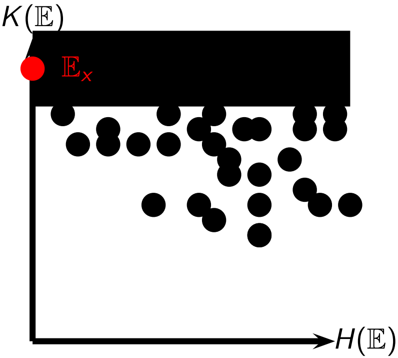
How to decide which ensemble $\mathbb{E}$ to take for a given string x?

Step 1: Allow only those $\mathbb{E}$ with $\mathbb{E}(x) \stackrel{\approx}{\geq} 2^{-H(\mathbb{E})}$.



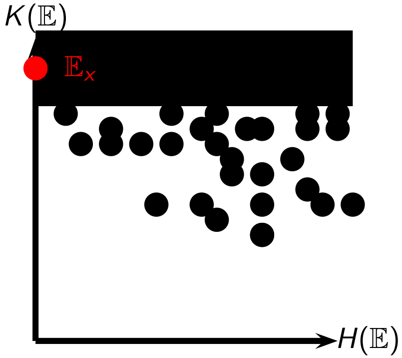
How to decide which ensemble $\mathbb{E}$ to take for a given string x?

Step 1: Allow only those $\mathbb{E}$ with $\mathbb{E}(x) \overset{\approx}{\geq} 2^{-H(\mathbb{E})}$.



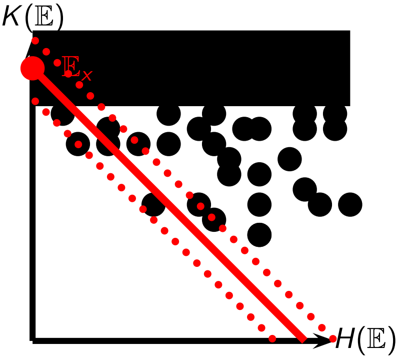
How to decide which ensemble $\mathbb{E}$ to take for a given string x?

Observation: The remaining ensembles all have total information $\Sigma(\mathbb{E}) := K(\mathbb{E}) + H(\mathbb{E}) \geq K(\mathbb{E}_x) - \mathcal{O}(1) = K(x) - \mathcal{O}(1)$.



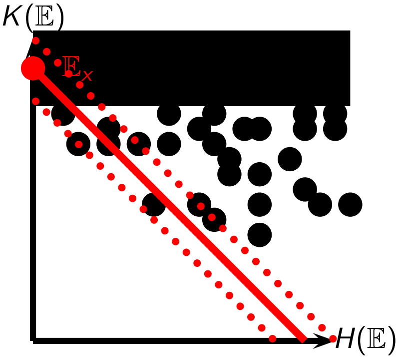
How to decide which ensemble $\mathbb{E}$ to take for a given string x?

Observation: The remaining ensembles all have total information $\Sigma(\mathbb{E}) := K(\mathbb{E}) + H(\mathbb{E}) \geq K(\mathbb{E}_x) - \mathcal{O}(1) = K(x) - \mathcal{O}(1)$.



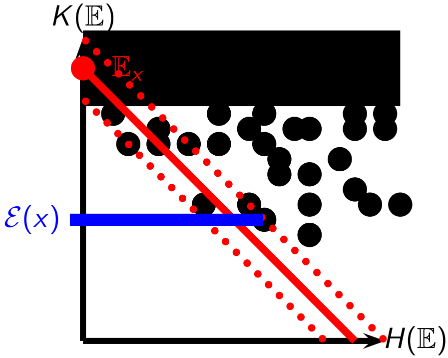
How to decide which ensemble $\mathbb{E}$ to take for a given string x?

Step 2: Find the ensemble $\mathbb{E}^*$ with minimal $K(\mathbb{E})$ along this approximate line $\rightarrow$ we are done: $\mathcal{E}(x) := K(\mathbb{E}^*)$.



How to decide which ensemble $\mathbb{E}$ to take for a given string x?

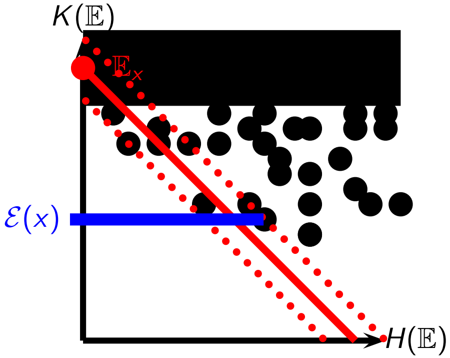
Step 2: Find the ensemble $\mathbb{E}^*$ with minimal $K(\mathbb{E})$ along this approximate line $\rightarrow$ we are done: $\mathcal{E}(x) := K(\mathbb{E}^*)$.



How to decide which ensemble $\mathbb{E}$ to take for a given string x?

Definition (Effective Complexity)

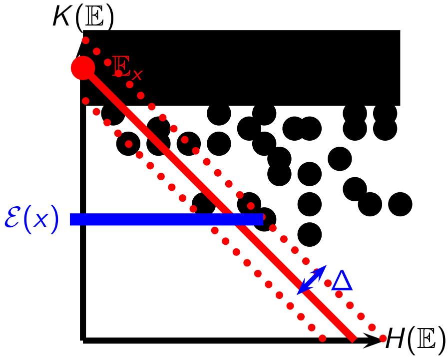
$\mathcal{E}_{\delta,\Delta}(x)$ is defined as the minimal $K(\mathbb{E})$ of all ensembles $\mathbb{E}$ with $\mathbb{E}(x) \geq 2^{-H(\mathbb{E})(1+\delta)}$ and $K(\mathbb{E}) + H(\mathbb{E}) \leq K(x) + \Delta$.



How to decide which ensemble $\mathbb{E}$ to take for a given string x?

Definition (Effective Complexity)

$\mathcal{E}_{\delta,\Delta}(x)$ is defined as the minimal $K(\mathbb{E})$ of all ensembles $\mathbb{E}$ with $\mathbb{E}(x) \geq 2^{-H(\mathbb{E})(1+\delta)}$ and $K(\mathbb{E}) + H(\mathbb{E}) \leq K(x) + \Delta$.



Random Strings are Effectively Simple:

Random Strings are Effectively Simple:

A string x of length n is "random" or r-incompressible, if

$$K(x) \geq n + K(n) - r.$$

Random Strings are Effectively Simple:

A string x of length n is "random" or r-incompressible, if

$$K(x) \geq n + K(n) - r.$$

- Most strings are r-incompressible (only a fraction of $2^{-r+\mathcal{O}(1)}$ is *not*).

Random Strings are Effectively Simple:

A string x of length n is "random" or r-incompressible, if

$$K(x) \geq n + K(n) - r.$$

- Most strings are r-incompressible (only a fraction of $2^{-r+\mathcal{O}(1)}$ is *not*).
- Example: Typical outcomes of fair coin tosses.

Random Strings are Effectively Simple:

A string x of length n is "random" or r-incompressible, if

$$K(x) \geq n + K(n) - r.$$

- Most strings are r-incompressible (only a fraction of $2^{-r+\mathcal{O}(1)}$ is *not*).
- Example: Typical outcomes of fair coin tosses.

Theorem (after Gell-Mann and Lloyd 1996)

There is a constant $c > 0$ such that

$$\mathcal{E}_{\delta,\Delta}(x) \leq \log n + \mathcal{O}(\log\log n)$$

for all r-incompressible strings x of length n, $\delta \geq 0$ and $\Delta \geq r + c$.

Random Strings are Effectively Simple:

Theorem (after Gell-Mann and Lloyd 1996)

There is a constant $c > 0$ such that

$$\mathcal{E}_{\delta,\Delta}(x) \leq \log n + \mathcal{O}(\log \log n)$$

for all r-incompressible strings x of length n, $\delta \geq 0$ and $\Delta \geq r + c$.

Random Strings are Effectively Simple:

Theorem (after Gell-Mann and Lloyd 1996)

There is a constant $c > 0$ such that

$$\mathcal{E}_{\delta,\Delta}(x) \leq \log n + \mathcal{O}(\log\log n)$$

for all r-incompressible strings x of length n, $\delta \geq 0$ and $\Delta \geq r + c$.

Reminder for later: even $\mathcal{E}_{\delta,\Delta}(x) \leq K(C(x)) + r + \mathcal{O}(1)$.

Random Strings are Effectively Simple:

Theorem (after Gell-Mann and Lloyd 1996)

There is a constant $c > 0$ such that

$$\mathcal{E}_{\delta,\Delta}(x) \leq \log n + \mathcal{O}(\log \log n)$$

for all r-incompressible strings x of length n, $\delta \geq 0$ and $\Delta \geq r + c$.

Reminder for later: even $\mathcal{E}_{\delta,\Delta}(x) \leq K(C(x)) + r + \mathcal{O}(1)$.

Idea of Proof:

Random Strings are Effectively Simple:

Theorem (after Gell-Mann and Lloyd 1996)

There is a constant $c > 0$ such that

$$\mathcal{E}_{\delta,\Delta}(x) \leq \log n + \mathcal{O}(\log\log n)$$

for all r-incompressible strings x of length n, $\delta \geq 0$ and $\Delta \geq r + c$.

Reminder for later: even $\mathcal{E}_{\delta,\Delta}(x) \leq K(C(x)) + r + \mathcal{O}(1)$.

Idea of Proof:

- r-incompressible strings x have $K(x) \geq n + K(n) - r$.

Random Strings are Effectively Simple:

Theorem (after Gell-Mann and Lloyd 1996)

There is a constant $c > 0$ such that

$$\mathcal{E}_{\delta,\Delta}(x) \leq \log n + \mathcal{O}(\log\log n)$$

for all r-incompressible strings x of length n, $\delta \geq 0$ and $\Delta \geq r + c$.

Reminder for later: even $\mathcal{E}_{\delta,\Delta}(x) \leq K(C(x)) + r + \mathcal{O}(1)$.

Idea of Proof:

- r-incompressible strings x have $K(x) \geq n + K(n) - r$.
- $\mathbb{E}$:=uniform distribution on $\{0,1\}^n$, then

Random Strings are Effectively Simple:

Theorem (after Gell-Mann and Lloyd 1996)

There is a constant $c > 0$ such that

$$\mathcal{E}_{\delta,\Delta}(x) \leq \log n + \mathcal{O}(\log\log n)$$

for all r-incompressible strings x of length n, $\delta \geq 0$ and $\Delta \geq r + c$.

Reminder for later: even $\mathcal{E}_{\delta,\Delta}(x) \leq K(C(x)) + r + \mathcal{O}(1)$.

Idea of Proof:

- r-incompressible strings x have $K(x) \geq n + K(n) - r$.
- $\mathbb{E}$:=uniform distribution on $\{0,1\}^n$, then
- $\mathbb{E}(x) = 2^{-n} = 2^{-H(\mathbb{E})} \geq 2^{-H(\mathbb{E})(1+\delta)}$ for every $\delta \geq 0$.

Random Strings are Effectively Simple:

Theorem (after Gell-Mann and Lloyd 1996)

There is a constant $c > 0$ such that

$$\mathcal{E}_{\delta,\Delta}(x) \leq \log n + \mathcal{O}(\log\log n)$$

for all r-incompressible strings x of length n, $\delta \geq 0$ and $\Delta \geq r + c$.

Reminder for later: even $\mathcal{E}_{\delta,\Delta}(x) \leq K(C(x)) + r + \mathcal{O}(1)$.

Idea of Proof:

- r-incompressible strings x have $K(x) \geq n + K(n) - r$.
- $\mathbb{E}$:=uniform distribution on $\{0,1\}^n$, then
- $\mathbb{E}(x) = 2^{-n} = 2^{-H(\mathbb{E})} \geq 2^{-H(\mathbb{E})(1+\delta)}$ for every $\delta \geq 0$.
- $\Sigma(\mathbb{E}) \equiv H(\mathbb{E}) + K(\mathbb{E}) \leq n + K(n) + c$

Random Strings are Effectively Simple:

Theorem (after Gell-Mann and Lloyd 1996)

There is a constant $c > 0$ such that

$$\mathcal{E}_{\delta,\Delta}(x) \leq \log n + \mathcal{O}(\log\log n)$$

for all r-incompressible strings x of length n, $\delta \geq 0$ and $\Delta \geq r + c$.

Reminder for later: even $\mathcal{E}_{\delta,\Delta}(x) \leq K(C(x)) + r + \mathcal{O}(1)$.

Idea of Proof:

- r-incompressible strings x have $K(x) \geq n + K(n) - r$.
- $\mathbb{E} :=$uniform distribution on $\{0,1\}^n$, then
- $\mathbb{E}(x) = 2^{-n} = 2^{-H(\mathbb{E})} \geq 2^{-H(\mathbb{E})(1+\delta)}$ for every $\delta \geq 0$.
- $\Sigma(\mathbb{E}) \equiv H(\mathbb{E}) + K(\mathbb{E}) \leq n + K(n) + c \leq K(x) + \Delta$. □

Existence of Effectively Complex Strings:

Existence of Effectively Complex Strings:

Most strings are r-incompressible and have $\mathcal{E}(x) \leq \mathcal{O}(\log n)$.

Existence of Effectively Complex Strings:

Most strings are r-incompressible and have $\mathcal{E}(x) \leq \mathcal{O}(\log n)$. Are there any effectively complex strings at all?

Existence of Effectively Complex Strings:

Most strings are r-incompressible and have $\mathcal{E}(x) \leq \mathcal{O}(\log n)$. Are there any effectively complex strings at all?

- $\mathcal{E}_{\delta,\Delta}(x) = K(\mathbb{E}^*) \leq K(\mathbb{E}^*) + H(\mathbb{E}^*) \leq K(x) + \Delta \leq n + \mathcal{O}(1)$.

Existence of Effectively Complex Strings:

Most strings are r-incompressible and have $\mathcal{E}(x) \leq \mathcal{O}(\log n)$. Are there any effectively complex strings at all?

- $\mathcal{E}_{\delta,\Delta}(x) = K(\mathbb{E}^*) \leq K(\mathbb{E}^*) + H(\mathbb{E}^*) \leq K(x) + \Delta \leq n + \mathcal{O}(1)$.
- Are there strings x with $\mathcal{E}(x) \approx n$?

Existence of Effectively Complex Strings:

Most strings are r-incompressible and have $\mathcal{E}(x) \leq \mathcal{O}(\log n)$. Are there any effectively complex strings at all?

- $\mathcal{E}_{\delta,\Delta}(x) = K(\mathbb{E}^*) \leq K(\mathbb{E}^*) + H(\mathbb{E}^*) \leq K(x) + \Delta \leq n + \mathcal{O}(1)$.
- Are there strings x with $\mathcal{E}(x) \approx n$?

Theorem (MM, A. Szkoła, N. Ay '08)

For every $\delta, \Delta \geq 0$ and $n \in \mathbb{N}$, there is a string x of length n with

$$\mathcal{E}_{\delta,\Delta}(x) \geq (1-\delta)n - \mathcal{O}(\log n).$$

Existence of Effectively Complex Strings:

Most strings are r-incompressible and have $\mathcal{E}(x) \leq \mathcal{O}(\log n)$. Are there any effectively complex strings at all?

- $\mathcal{E}_{\delta,\Delta}(x) = K(\mathbb{E}^*) \leq K(\mathbb{E}^*) + H(\mathbb{E}^*) \leq K(x) + \Delta \leq n + \mathcal{O}(1)$.
- Are there strings x with $\mathcal{E}(x) \approx n$?

Theorem (MM, A. Szkoła, N. Ay '08)

For every $\delta, \Delta \geq 0$ and $n \in \mathbb{N}$, there is a string x of length n with

$$\mathcal{E}_{\delta,\Delta}(x) \geq (1-\delta)n - \mathcal{O}(\log n).$$

- Is also true for $\mathcal{E}$ with constraints: adding constraits makes effective complexity increase.

Existence of Effectively Complex Strings:

Most strings are r-incompressible and have $\mathcal{E}(x) \leq \mathcal{O}(\log n)$. Are there any effectively complex strings at all?

- $\mathcal{E}_{\delta,\Delta}(x) = K(\mathbb{E}^*) \leq K(\mathbb{E}^*) + H(\mathbb{E}^*) \leq K(x) + \Delta \leq n + \mathcal{O}(1)$.
- Are there strings x with $\mathcal{E}(x) \approx n$?

Theorem (MM, A. Szkoła, N. Ay '08)

For every $\delta, \Delta \geq 0$ *and* $n \in \mathbb{N}$, *there is a string* x *of length* n *with*

$$\mathcal{E}_{\delta,\Delta}(x) \geq (1-\delta)n - \mathcal{O}(\log n).$$

- Is also true for $\mathcal{E}$ with constraints: adding constraits makes effective complexity increase.
- Main ingredient: Gács '01: There are ("non-stochastic") strings x with a strange property:

Existence of Effectively Complex Strings:

Most strings are r-incompressible and have $\mathcal{E}(x) \leq \mathcal{O}(\log n)$. Are there any effectively complex strings at all?

- $\mathcal{E}_{\delta,\Delta}(x) = K(\mathbb{E}^*) \leq K(\mathbb{E}^*) + H(\mathbb{E}^*) \leq K(x) + \Delta \leq n + \mathcal{O}(1)$.
- Are there strings x with $\mathcal{E}(x) \approx n$?

Theorem (MM, A. Szkoła, N. Ay '08)

For every $\delta, \Delta \geq 0$ and $n \in \mathbb{N}$, there is a string x of length n with

$$\mathcal{E}_{\delta,\Delta}(x) \geq (1-\delta)n - \mathcal{O}(\log n).$$

- Is also true for $\mathcal{E}$ with constraints: adding constraits makes effective complexity increase.
- Main ingredient: Gács '01: There are ("non-stochastic") strings x with a strange property: For *every* simple set S, there is always a simpler way to describe x given S then to specify its position (index) within S.

Existence of Effectively Complex Strings:

Open Problem: Find more intuitive examples, or give a better interpretation! ($\rightarrow$ Gell-Mann, Lloyd: DNA?)

Theorem (MM, A. Szkoła, N. Ay '08)

For every $\delta, \Delta \geq 0$ *and* $n \in \mathbb{N}$, *there is a string* x *of length* n *with*

$$\mathcal{E}_{\delta,\Delta}(x) \geq (1-\delta)n - \mathcal{O}(\log n).$$

- Is also true for $\mathcal{E}$ with constraints: adding constraits makes effective complexity increase.
- Main ingredient: Gács '01: There are ("non-stochastic") strings x with a strange property: For *every* simple set S, there is always a simpler way to describe x given S then to specify its position (index) within S.

Effective Complexity and Logical Depth (informal)

Effective Complexity and Logical Depth (informal)

Bennett '88: The logical depth of a string x is the minimal number of time steps required by a universal computer to produce x from an almost-minimal program.

Effective Complexity and Logical Depth (informal)

Bennett '88: The logical depth of a string x is the minimal number of time steps required by a universal computer to produce x from an almost-minimal program.

- Random strings x of length n have $C(x) \approx n$, and thus small depth: "Print the following digits: $\underbrace{010010\ldots}_{x}$" is an almost-minimal program, so $\mathrm{Depth}(x) \approx n$ (x is "shallow").

Effective Complexity and Logical Depth (informal)

Bennett '88: The logical depth of a string x is the minimal number of time steps required by a universal computer to produce x from an almost-minimal program.

- Random strings x of length n have $C(x) \approx n$, and thus small depth: "Print the following digits: $\underbrace{010010\ldots}_{x}$" is an almost-minimal program, so $\mathrm{Depth}(x) \approx n$ (x is "shallow").
- $x = x_1 x_2 \ldots x_n$ with $x_k := 2^{2^k}$-th binary digit of $\log(\pi + e)$ is probably deep: $\mathrm{Depth}(x) \approx 2^{2^n}$.

Effective Complexity and Logical Depth (informal)

Bennett '88: The logical depth of a string x is the minimal number of time steps required by a universal computer to produce x from an almost-minimal program.

- Random strings x of length n have $C(x) \approx n$, and thus small depth: "Print the following digits: $\underbrace{010010\ldots}_{x}$" is an almost-minimal program, so $\mathrm{Depth}(x) \approx n$ (x is "shallow").
- $x = x_1 x_2 \ldots x_n$ with $x_k := 2^{2^k}$-th binary digit of $\log(\pi + e)$ is probably deep: $\mathrm{Depth}(x) \approx 2^{2^n}$.
- The Quark and the Jaguar: Gell-Mann discusses interrelations between algorithmic information content and effective complexity as well as logical depth.
  $\longrightarrow$ relation between depth and complexity?

Effective Complexity and Logical Depth:

Effective Complexity and Logical Depth:

Theorem (MM, A. Szkoła, N. Ay '08)

If $f : \mathbb{N} \to \mathbb{N}$ *is a strictly increasing, computable function and* x *is a string then*

$$\mathcal{E}(x) > K(C(x)) + K(f) + \mathcal{O}(1)$$

implies $\mathrm{Depth}(x) > f(C(x))$.

Effective Complexity and Logical Depth:

Theorem (MM, A. Szkoła, N. Ay '08)

If $f : \mathbb{N} \to \mathbb{N}$ is a strictly increasing, computable function and x is a string then

$$\mathcal{E}(x) > K(C(x)) + K(f) + \mathcal{O}(1)$$

implies $\mathrm{Depth}(x) > f(C(x))$.

- f can be simple (i.e. $K(f)$ small), but absurdly rapidly growing, i.e. $f(n) := n^{n^{n^{\cdots}}}$ (power tower of height n).

Effective Complexity and Logical Depth:

Theorem (MM, A. Szkoła, N. Ay '08)

If $f : \mathbb{N} \to \mathbb{N}$ is a strictly increasing, computable function and x is a string then

$$\mathcal{E}(x) > K(C(x)) + K(f) + \mathcal{O}(1)$$

implies $\mathrm{Depth}(x) > f(C(x))$.

- f can be simple (i.e. $K(f)$ small), but absurdly rapidly growing, i.e. $f(n) := n^{n^{n^{n^{\cdots}}}}$ (power tower of height n).
- $K(C(x)) = \mathcal{O}(\log n)$, so the inequality condition is weak.

Effective Complexity and Logical Depth:

Theorem (MM, A. Szkoła, N. Ay '08)

If $f : \mathbb{N} \to \mathbb{N}$ is a strictly increasing, computable function and x is a string then

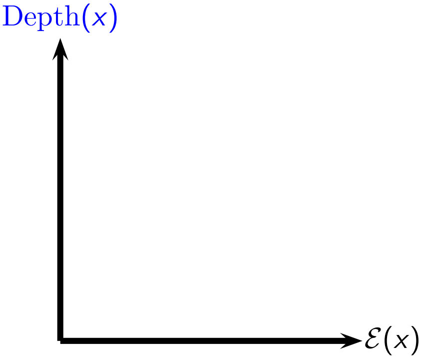
$$\mathcal{E}(x) > K(C(x)) + K(f) + \mathcal{O}(1)$$

implies $\mathrm{Depth}(x) > f(C(x))$.

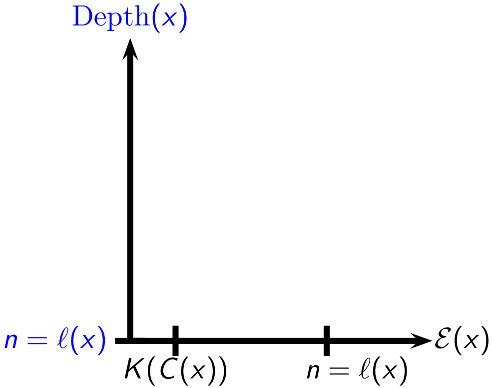
- f can be simple (i.e. $K(f)$ small), but absurdly rapidly growing, i.e. $f(n) := n^{n^{n^{\cdots}}}$ (power tower of height n).
- $K(C(x)) = \mathcal{O}(\log n)$, so the inequality condition is weak.
- Holds also for effective complexity with constraints (under mild assumptions on the constraints).

Effective Complexity and Logical Depth:

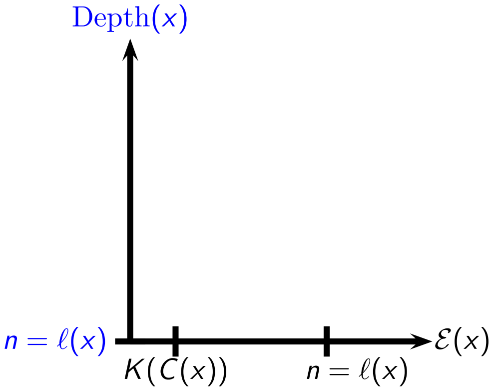
Effective Complexity and Logical Depth:



Effective Complexity and Logical Depth:

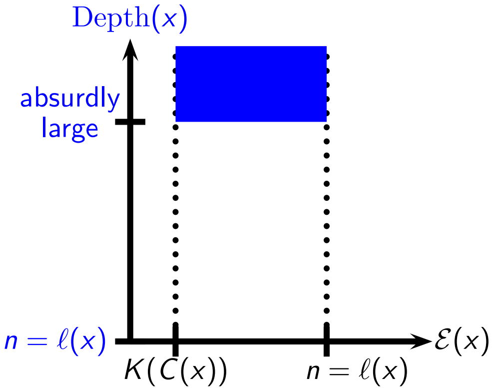


Effective Complexity and Logical Depth:



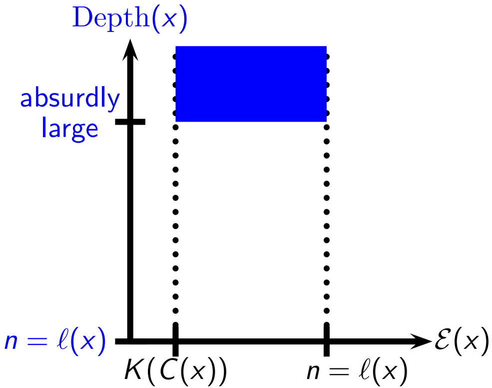
- If $\mathcal{E}(x) > K(C(x)) + K(f) + \mathcal{O}(1)$, then Depth$(x) > f(C(x))$.

Effective Complexity and Logical Depth:



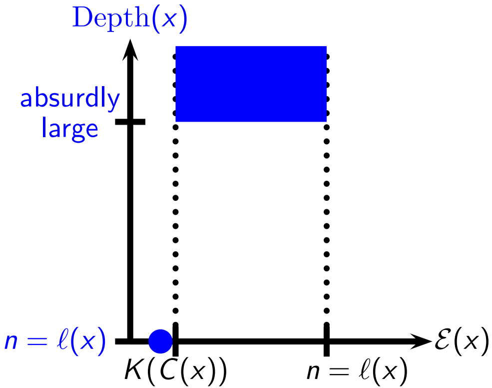
- If $\mathcal{E}(x) > K(C(x)) + K(f) + \mathcal{O}(1)$, then $\mathrm{Depth}(x) > f(C(x))$.

Effective Complexity and Logical Depth:



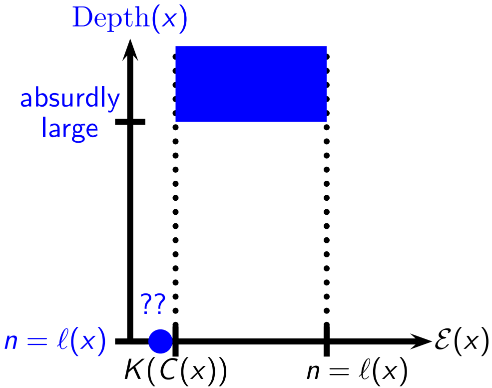
- If x is incompressible, then $\mathcal{E}(x) \leq K(C(x)) + \mathcal{O}(1)$. But x is random hence shallow, so $Depth(x) = n + \mathcal{O}(1)$.

Effective Complexity and Logical Depth:



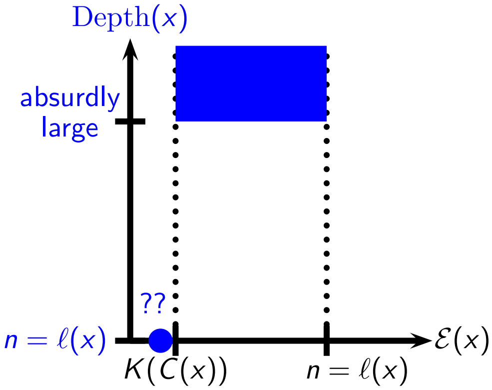
- If x is incompressible, then $\mathcal{E}(x) \leq K(C(x)) + \mathcal{O}(1)$. But x is random hence shallow, so $Depth(x) = n + \mathcal{O}(1)$.

Effective Complexity and Logical Depth:



- If x is incompressible, then $\mathcal{E}(x) \leq K(C(x)) + \mathcal{O}(1)$. But x is random hence shallow, so $Depth(x) = n + \mathcal{O}(1)$.

Effective Complexity and Logical Depth:



- The "edge of depth" at $K(C(x))$. Happens also for $\mathcal{E}$ with constraints.

Conclusions

Conclusions

- M. Gell-Mann and S. Llyod have proposed effective complexity as a complexity measure, to overcome limitations of Algorithmic Information Content ("Kolmogorov Complexity").

Conclusions

- M. Gell-Mann and S. Llyod have proposed effective complexity as a complexity measure, to overcome limitations of Algorithmic Information Content ("Kolmogorov Complexity").
- We have proposed a formal definition $\mathcal{E}_{\delta,\Delta}(x)$ of effective complexity, including all the additive constants, and examined its properties. In particular, we have

Conclusions

- M. Gell-Mann and S. Llyod have proposed effective complexity as a complexity measure, to overcome limitations of Algorithmic Information Content ("Kolmogorov Complexity").
- We have proposed a formal definition $\mathcal{E}_{\delta,\Delta}(x)$ of effective complexity, including all the additive constants, and examined its properties. In particular, we have
  - reproduced the result by Gell-Mann and Lloyd that incompressible (random) strings x are simple, i.e. have $\mathcal{E}(x) \leq \log n + \mathcal{O}(\log \log n)$,

Conclusions

- M. Gell-Mann and S. Llyod have proposed effective complexity as a complexity measure, to overcome limitations of Algorithmic Information Content ("Kolmogorov Complexity").
- We have proposed a formal definition $\mathcal{E}_{\delta,\Delta}(x)$ of effective complexity, including all the additive constants, and examined its properties. In particular, we have
  - reproduced the result by Gell-Mann and Lloyd that incompressible (random) strings x are simple, i.e. have $\mathcal{E}(x) \leq \log n + \mathcal{O}(\log\log n)$,
  - shown that there exist effectively complex strings x with $\mathcal{E}(x) \geq n - \mathcal{O}(\log n)$,

Conclusions

- M. Gell-Mann and S. Llyod have proposed effective complexity as a complexity measure, to overcome limitations of Algorithmic Information Content ("Kolmogorov Complexity").
- We have proposed a formal definition $\mathcal{E}_{\delta,\Delta}(x)$ of effective complexity, including all the additive constants, and examined its properties. In particular, we have
  - reproduced the result by Gell-Mann and Lloyd that incompressible (random) strings x are simple, i.e. have $\mathcal{E}(x) \leq \log n + \mathcal{O}(\log\log n)$,
  - shown that there exist effectively complex strings x with $\mathcal{E}(x) \geq n - \mathcal{O}(\log n)$,
  - found a relation between effective complexity and Bennett's logical depth: If $\mathcal{E}(x) > K(C(x)) + \mathcal{O}(1)$, then $\mathrm{Depth}(x)$ is astronomically large. Otherwise, it can be arbitrarily small.