Kolmogorov Complexity

imagine you are told at a bar
(during a conference):

Chaitin defined a string to be random
if it is its own shortest description

Isn’t that neat?

imagine you are told at a bar
(during a conference):

Chaitin defined a string to be random
if it is its own shortest description

Isn’t that neat?

getting history right:

- Chaitin had rediscovered Kolmogorov Complexity
- which had previously been described by Solomonov

imagine you are told at a bar
(during a conference):

Chaitin defined a string to be random
if it is its own shortest description

Isn’t that neat?

getting history right:

- Chaitin had rediscovered Kolmogorov Complexity
- which had previously been described by Solomonov

how would you have defined ‘description’?

Hint: this is I2TC

1 Information content of a bit string $x \in \mathbb{B}^n$

examples:

- $x = 000000000000000000000000000000$:

$$x = 0^{30} \quad \text{or} \quad x = 0 \text{ (30 times)}$$

compression is
(intuitively) possible

- $x = 01010101010101010101010101010101$

$$x = (01)^{16}$$

1 Information content of a bit string $x \in \mathbb{B}^n$

examples:

- $x = 000000000000000000000000000000$:

$$x = 0^{30} \quad \text{or} \quad x = 0 \text{ (30 times)}$$

- $x = 01010101010101010101010101010101$

$$x = (01)^{16}$$

compression is (intuitively) possible

information content/complexity $K(x)$ of x, idea:

- $K(x)$ is the length of the shortest description of x.

1 Information content of a bit string $x \in \mathbb{B}^n$

examples:

- $x = 000000000000000000000000000000$:

$$x = 0^{30} \quad \text{or} \quad x = 0 \text{ (30 times)}$$

compression is (intuitively) possible

- $x = 01010101010101010101010101010101$

$$x = (01)^{16}$$

information content/complexity $K(x)$ of x, idea:

- $K(x)$ is the length of the shortest description of x.

- Let Y be the lexicographically smallest $y \in \mathbb{B}^*$ which cannot be described in 3 lines ($K(y) \geq 4$ lines).

1 Information content of a bit string $x \in \mathbb{B}^n$

examples:

- $x = 000000000000000000000000000000$:

$$x = 0^{30} \quad \text{or} \quad x = 0 \text{ (30 times)}$$

compression is
(intuitively) possible

- $x = 01010101010101010101010101010101$

$$x = (01)^{16}$$

information content/complexity $K(x)$ of x, idea:

- $K(x)$ is the length of the shortest description of x.

- Let Y be the lexicographically smallest $y \in \mathbb{B}^*$ which cannot be described in 3 lines ($K(y) \geq 4$ lines).

- we just described Y in 2 lines

1 Information content of a bit string $x \in \mathbb{B}^n$

examples:

- $x = 000000000000000000000000000000$:

$$x = 0^{30} \quad \text{or} \quad x = 0 \text{ (30 times)}$$

- $x = 01010101010101010101010101010101$

$$x = (01)^{16}$$

compression is
(intuitively) possible

information content/complexity $K(x)$ of x, idea:

- $K(x)$ is the length of the shortest description of x.

- Let Y be the lexicographically smallest $y \in \mathbb{B}^*$ which cannot be described in 3 lines ($K(y) \geq 4$ lines).

- we just described Y in 2 lines

- what is a description??

1 Information content of a bit string $x \in \mathbb{B}^n$

examples:

- $x = 000000000000000000000000000000$:

$$x = 0^{30} \quad \text{or} \quad x = 0 \text{ (30 times)}$$

- $x = 01010101010101010101010101010101$

$$x = (01)^{16}$$

information content/complexity $K(x)$ of x, idea:

- $K(x)$ is the length of the shortest description of x.

def: $K(x)$ almost:

- a description of x is a string $u\#v$ such that M_u started with v prints x and halts.

$$K(x) = \min\{|u\#v| \;:\; M_u \text{ started with } v \text{ outputs } x \text{ and halts}\}$$

- Let Y be the lexicographically smallest $y \in \mathbb{B}^*$ which cannot be described in 3 lines ($K(y) \geq 4$ lines).

- we just described Y in 2 lines

- what is a description??

1 Information content of a bit string $x \in \mathbb{B}^n$

examples:

- $x = 000000000000000000000000000000$:

$$x = 0^{30} \quad \text{or} \quad x = 0 \text{ (30 times)}$$

- $x = 01010101010101010101010101010101$

$$x = (01)^{16}$$

information content/complexity $K(x)$ of x, idea:

- $K(x)$ is the length of the shortest description of x.

- Let Y be the lexicographically smallest $y \in \mathbb{B}^*$ which cannot be described in 3 lines ($K(y) \geq 4$ lines).
- we just described Y in 2 lines

- what is a description??

def: $K(x)$ almost:

- a description of x is a string $u\#v$ such that M_u started with v prints x and halts.

$$K(x) = \min\{|u\#v| \ : \ M_u \text{ started with } v \text{ outputs } x \text{ and halts}\}$$

- unit: not bits, as $\# \notin \mathbb{B}$. Position of # contains information.

def: binary comma u'; self delimiting strings

- for $w \in \{0,1,\#\}^*$ obtain (as usual)

$$h(0) = 00 \ , \ h(1) = 11 \ , \ h(\#) = 10$$

$$h(w[1:n]) = h(w_1)\dots h(w_n)$$

- for $u \in \mathbb{B}^*$ define

$$u' = h(bin(|u|)\#)u$$

example $u = 01$

$$\begin{aligned} bin(|u|) &= 10 \\ h(bin(|u|)) &= 1100 \\ u' &= 11001001 \end{aligned}$$

def: binary comma u'; self delimiting strings

- for $w \in \{0,1,\#\}^*$ obtain (as usual)

$$h(0) = 00\ ,\ h(1) = 11\ ,\ h(\#) = 10$$

$$h(w[1:n]) = h(w_1)\dots h(w_n)$$

- for $u \in \mathbb{B}^*$ define

$$u' = h(bin(|u|)\#)u$$

example $u = 01$

$$\begin{aligned} bin(|u|) &= 10 \\ h(bin(|u|)) &= 1100 \\ u' &= 11001001 \end{aligned}$$

Lemma 1.

$$|u'| = |u| + O(\log(|u|))$$

Proof.

$$|u'| = 2\lceil \log(|u|) \rceil + 2 + |u|$$

def: binary comma u'; self delimiting strings

- for $w \in \{0,1,\#\}^*$ obtain (as usual)

$$h(0) = 00 \; , \; h(1) = 11 \; , \; h(\#) = 10$$

$$h(w[1:n]) = h(w_1) \dots h(w_n)$$

- for $u \in \mathbb{B}^*$ define

$$u' = h(bin(|u|)\#)u$$

example $u = 01$

$$\begin{aligned} bin(|u|) &= 10 \\ h(bin(|u|)) &= 1100 \\ u' &= 11001001 \end{aligned}$$

Lemma 1.

$$|u'| = |u| + O(\log(|u|))$$

Proof.

$$|u'| = 2\lceil \log(|u|) \rceil + 2 + |u|$$

exercise: can you define binary comma, such that

$$|u'| = |u| + \log(|u|) + O(\log\log(|u|))$$

def: binary comma u'; self delimiting strings

- for $w \in \{0,1,\#\}^*$ obtain (as usual)

$$h(0) = 00 \ , \ h(1) = 11 \ , \ h(\#) = 10$$

$$h(w[1:n]) = h(w_1) \dots h(w_n)$$

- for $u \in \mathbb{B}^*$ define

$$u' = h(bin(|u|)\#)u$$

example $u = 01$

$$\begin{aligned} bin(|u|) &= 10 \\ h(bin(|u|)) &= 1100 \\ u' &= 11001001 \end{aligned}$$

Lemma 1.

$$|u'| = |u| + O(\log(|u|))$$

Proof.

$$|u'| = 2\lceil \log(|u|) \rceil + 2 + |u|$$

exercise: can you define binary comma, such that

$$|u'| = |u| + \log(|u|) + O(\log\log(|u|))$$

def: Kolmogorov complexity $K(x)$

$$K(x) = \min\{|u'v| \ : \ M_u \text{ started with } v \text{ outputs } x \text{ and halts}\}$$

def: binary comma u'; self delimiting strings

- for $w \in \{0,1,\#\}^*$ obtain (as usual)

$$h(0) = 00\ ,\ h(1) = 11\ ,\ h(\#) = 10$$

$$h(w[1:n]) = h(w_1)\dots h(w_n)$$

- for $u \in \mathbb{B}^*$ define

$$u' = h(bin(|u|)\#)u$$

example $u = 01$

$$\begin{aligned} bin(|u|) &= 10 \\ h(bin(|u|)) &= 1100 \\ u' &= 11001001 \end{aligned}$$

Lemma 1.

$$|u'| = |u| + O(\log(|u|))$$

Proof.

$$|u'| = 2\lceil \log(|u|) \rceil + 2 + |u|$$

exercise: can you define binary comma, such that

$$|u'| = |u| + \log(|u|) + O(\log\log(|u|))$$

def: Kolmogorov complexity $K(x)$

$$K(x) = \min\{|u'v| \ :\ M_u \text{ started with } v \text{ outputs } x \text{ and halts}\}$$

Lemma 2. *For all $x \in \mathbb{B}^*$:*

$$K(x) \leq |x| + O(1)$$

Proof. idea: you can always specify a string by just writing it down (+ saying: this is it)

def: binary comma u'; self delimiting strings

- for $w \in \{0,1,\#\}^*$ obtain (as usual)

$$h(0) = 00\ ,\ h(1) = 11\ ,\ h(\#) = 10$$

$$h(w[1:n]) = h(w_1)\dots h(w_n)$$

- for $u \in \mathbb{B}^*$ define

$$u' = h(bin(|u|)\#)u$$

example $u = 01$

$$\begin{aligned} bin(|u|) &= 10 \\ h(bin(|u|)) &= 1100 \\ u' &= 11001001 \end{aligned}$$

Lemma 1.

$$|u'| = |u| + O(\log(|u|))$$

Proof.

$$|u'| = 2\lceil\log(|u|)\rceil + 2 + |u|$$

exercise: can you define binary comma, such that

$$|u'| = |u| + \log(|u|) + O(\log\log(|u|))$$

def: Kolmogorov complexity $K(x)$

$$K(x) = \min\{|u'v|\ :\ M_u \text{ started with } v \text{ outputs } x \text{ and halts}\}$$

Lemma 2. *For all $x \in \mathbb{B}^*$:*

$$K(x) \le |x| + O(1)$$

Proof. idea: you can always specify a string by just writing it down (+ saying: this is it)

- M_u: ignores input and halts.
- $u'x$ describes x

$$|u'x| = |x| + O(\log(|u|)) = |x| + O(1)$$

$$+O(|u|)$$

def: binary comma u'; self delimiting strings

- for $w \in \{0,1,\#\}^*$ obtain (as usual)

$$h(0) = 00\ ,\ h(1) = 11\ ,\ h(\#) = 10$$

$$h(w[1:n]) = h(w_1)\dots h(w_n)$$

- for $u \in \mathbb{B}^*$ define

$$u' = h(bin(|u|)\#)u$$

example $u = 01$

$$\begin{aligned} bin(|u|) &= 10 \\ h(bin(|u|)) &= 1100 \\ u' &= 11001001 \end{aligned}$$

Lemma 1.

$$|u'| = |u| + O(\log(|u|))$$

Proof.

$$|u'| = 2\lceil \log(|u|) \rceil + 2 + |u|$$

exercise: can you define binary comma, such that

$$|u'| = |u| + \log(|u|) + O(\log\log(|u|))$$

def: Kolmogorov complexity $K(x)$

$$K(x) = \min\{|u'v| \ :\ M_u \text{ started with } v \text{ outputs } x \text{ and halts}\}$$

Lemma 2. *For all $x \in \mathbb{B}^*$:*

$$K(x) \leq |x| + O(1)$$

Proof. idea: you can always specify a string by just writing it down (+ saying: this is it)

- M_u: ignores input and halts.
- $u'x$ describes x

$$|u'x| = |x| + O(\log(|u|)) = |x| + O(1)$$

$$+O(|u|)$$

Lemma 3.

$$K(1^n) = \log(n) + O(1)$$

- M_u started with $bin(n)$ prints 1^n and halts
- $u'bin(n)$ describes 1^n

$$|u'bin(n)| = \log(n) + O(1)$$

2 Random Strings

idea: a string x is random, if it is its own shortest description.

def: random strings

$$x \in \mathbb{B}^* \text{ random } \leftrightarrow K(x) \geq |x|$$

2 Random Strings

idea: a string x is random, if it is its own shortest description.

def: random strings

$$x \in \mathbb{B}^* \text{ random } \leftrightarrow K(x) \geq |x|$$

Lemma 4. *For all n there are random strings $x \in \mathbb{B}^n$*

2 Random Strings

idea: a string x is random, if it is its own shortest description.

def: random strings

$$x \in \mathbb{B}^* \text{ random } \leftrightarrow K(x) \geq |x|$$

Lemma 4. *For all n there are random strings $x \in \mathbb{B}^n$*

- assume for all $x \in \mathbb{B}^n$

$$K(x) \leq n - 1$$

- the number N of descriptions with length $i < n$ is at most

$$\begin{aligned} N &= |\{u'v : |u'v| \leq n-1\}| \\ &\leq \sum_{i=0}^{n-1} |\mathbb{B}^i| \\ &= 2^n - 1 \\ &< |\mathbb{B}^n| \end{aligned}$$

not enough short descriptions

exercise: for how many bit strings x of length n
must hold $K(x) \geq n - c$?

2 Random Strings

idea: a string x is random, if it is its own shortest description.

def: random strings

$$x \in \mathbb{B}^* \text{ random } \leftrightarrow K(x) \geq |x|$$

Lemma 4. *For all n there are random strings $x \in \mathbb{B}^n$*

- assume for all $x \in \mathbb{B}^n$
$$K(x) \leq n-1$$
- the number N of descriptions with length $i < n$ is at most

$$\begin{aligned} N &= |\{u'v : |u'v| \leq n-1\}| \\ &\leq \sum_{i=0}^{n-1} |\mathbb{B}^i| \\ &= 2^n - 1 \\ &< |\mathbb{B}^n| \end{aligned}$$

not enough short descriptions

exercise: for how many bit strings x of length n must hold $K(x) \geq n - c$?

3 Independence of machine model

- Definition of $K(x)$ depends on coding of 1-Tape Turing machines M_u. So we really defined

$$K(x) = K_{1-tape-TM}(x)$$

- What if we use (binary coded) C-programs instead do define $K_C(x)$?
- what if we compare $K_M(x), K_{M'}(x)$ for arbitrary machine models M and M', each capable to compute the computable functions.

2 Random Strings

idea: a string x is random, if it is its own shortest description.

def: random strings

$$x \in \mathbb{B}^* \text{ random } \leftrightarrow K(x) \geq |x|$$

Lemma 4. *For all n there are random strings $x \in \mathbb{B}^n$*

- assume for all $x \in \mathbb{B}^n$

$$K(x) \leq n-1$$

- the number N of descriptions with length $i < n$ is at most

$$\begin{aligned} N &= |\{u'v \;:\; |u'v| \leq n-1\}| \\ &\leq \sum_{i=0}^{n-1} |\mathbb{B}^i| \\ &= 2^n - 1 \\ &< |\mathbb{B}^n| \end{aligned}$$

not enough short descriptions

exercise: for how many bit strings x of length n must hold $K(x) \geq n - c$?

3 Independence of machine model

- Definition of $K(x)$ depends on coding of 1-Tape Turing machines M_u. So we really defined

$$K(x) = K_{1-tape-TM}(x)$$

- What if we use (binary coded) C-programs instead do define $K_C(x)$?
- what if we compare $K_M(x), K_{M'}(x)$ for arbitratry machine models M and M', each capable to compute the computable functions.

Lemma 5. *Kolmogorov complexity of strings x differs for different models of computation only by an additive constent*

$$K_M(x) \leq K_{M'}(x) + O(1)$$

2 Random Strings

idea: a string x is random, if it is its own shortest description.

def: random strings

$$x \in \mathbb{B}^* \text{ random } \leftrightarrow K(x) \geq |x|$$

Lemma 4. *For all n there are random strings $x \in \mathbb{B}^n$*

- assume for all $x \in \mathbb{B}^n$
$$K(x) \leq n-1$$
- the number N of descriptions with length $i < n$ is at most

$$\begin{aligned} N &= |\{u'v \,:\, |u'v| \leq n-1\}| \\ &\leq \sum_{i=0}^{n-1} |\mathbb{B}^i| \\ &= 2^n - 1 \\ &< |\mathbb{B}^n| \end{aligned}$$

not enough short descriptions

exercise: for how many bit strings x of length n must hold $K(x) \geq n - c$?

3 Independence of machine model

- Definition of $K(x)$ depends on coding of 1-Tape Turing machines M_u. So we really defined

$$K(x) = K_{1-tape-TM}(x)$$

- What if we use (binary coded) C-programs instead do define $K_C(x)$?
- what if we compare $K_M(x), K_{M'}(x)$ for arbitrary machine models M and M', each capable to compute the computable functions.

Lemma 5. *Kolmogorov complexity of strings x differs for different models of computation only by an additive constent*

$$K_M(x) \leq K_{M'}(x) + O(1)$$

- Let $u'v$ be a shortest description of x in model M.
- let J be an interpreter of programs for M written in M', i.e. J started with $a'b$ simulates program a on input b. (Church's thesis)
- then $J'u'v$ is a description of x in model M'

$$K_{M'}(x) = |J'u'v| = O(1) + |u'v| = O(1) + K_M(x)$$

4 Kolmogorov complexity is not computable

$\boxed{\textbf{Lemma 6. } K(x) \textit{ is not computable}}$

4 Kolmogorov complexity is not computable

$\boxed{\textbf{Lemma 6. } K(x) \textit{ is not computable}}$

assume otherwise.

- M_u with input $bin(n)$
  1. enumerates in lex. order $x \in \mathbb{B}^n$ and computes $K(x)$
  2. once it finds an x with $K(x) \geq n$ it outputs x and halts. This will happen by lemma 4.

4 Kolmogorov complexity is not computable

Lemma 6. *$K(x)$ is not computable*

assume otherwise.

- M_u with input $bin(n)$

  1. enumerates in lex. order $x \in \mathbb{B}^n$ and computes $K(x)$
  2. once it finds an x with $K(x) \geq n$ it outputs x and halts. This will happen by lemma 4.

- $u'bin(n)$ describes a random string in $x \in \mathbb{B}^n$. Thus

$$\begin{aligned} n &\leq K(x) \\ &\leq |u'bin(n)| \\ &\leq O(1) + \log(n) \end{aligned}$$

5 Conditional Kolmogorov complexity

question: How much does a given string y help to describe x?

5 Conditional Kolmogorov complexity

question: How much does a given string y help to describe x?

def: $K(x|y)$ For $x, y \in \mathbb{B}^*$ we define

$$K(x|y) = \min\{u'v \; : \; M_u \text{started with } v\#y \text{ outputs } x \text{ and halts}\}$$

Here we can afford to indicate, what is given with $\# \notin \mathbb{B}$.

5 Conditional Kolmogorov complexity

question: How much does a given string y help to describe x?

def: $K(x|y)$ For $x,y \in \mathbb{B}^*$ we define

$$K(x|y) = \min\{u'v \;:\; M_u \text{started with } v\#y \text{ outputs } x \text{ and halts}\}$$

Here we can afford to indicate, what is given with $\# \notin \mathbb{B}$.

example:

Lemma 7. $K(x|\overline{x}) = O(1)$

- M_u started with y flips all bits of y and halts.
- $u'\#\overline{x}$ describes x
- $|u'| = O(1)$

5 Conditional Kolmogorov complexity

question: How much does a given string y help to describe x?

def: $K(x|y)$ For $x, y \in \mathbb{B}^*$ we define

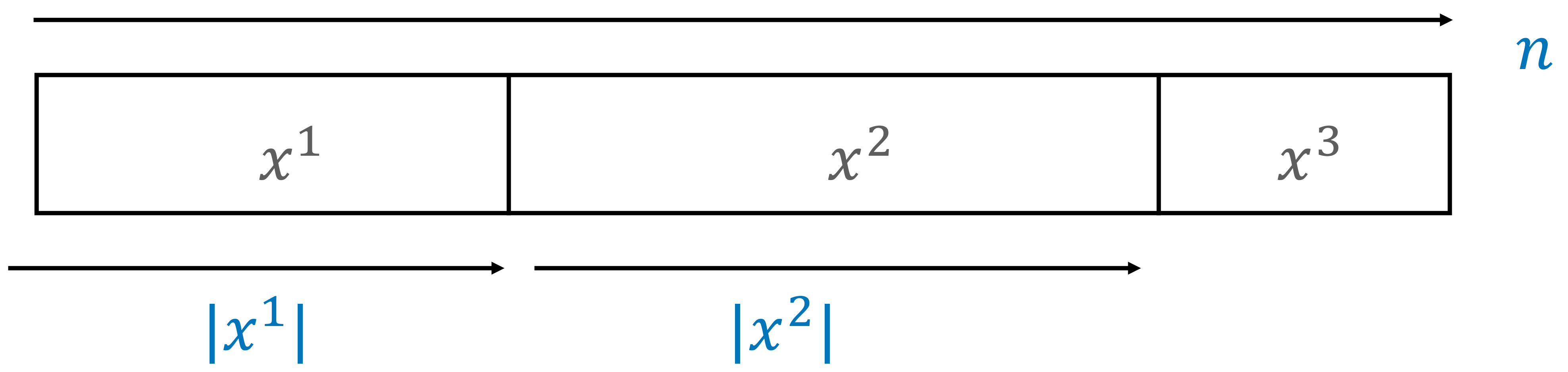
$$K(x|y) = \min\{u'v \ : \ M_u \text{started with } v\#y \text{ outputs } x \text{ and halts}\}$$

Here we can afford to indicate, what is given with $\# \notin \mathbb{B}$.

example:

Lemma 7. $K(x|\overline{x}) = O(1)$

- M_u started with y flips all bits of y and halts.
- $u'\#\overline{x}$ describes x
- $|u'| = O(1)$



6 Substrings of random strings

Lemma 8. *Substrings of random strings are almost random, even if the remainder of the string is given. Let*

$$x = x^1x^2x^3 \in \mathbb{B}^n \text{ be random}$$

Then

$$K(x^2|x^1x^3) \geq |x^2| - O(\log(n))$$

5 Conditional Kolmogorov complexity

question: How much does a given string y help to describe x?

def: $K(x|y)$ For $x, y \in \mathbb{B}^*$ we define

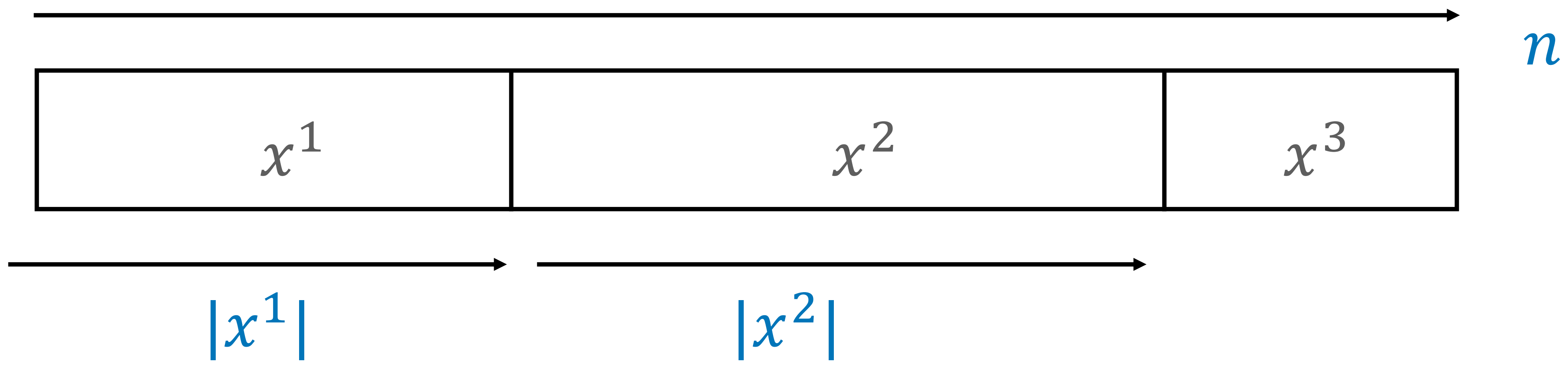
$$K(x|y) = \min\{u'v \ : \ M_u \text{started with } v\#y \text{ outputs } x \text{ and halts}\}$$

Here we can afford to indicate, what is given with $\# \notin \mathbb{B}$.

example:

Lemma 7. $K(x|\overline{x}) = O(1)$

- M_u started with y flips all bits of y and halts.
- $u'\#\overline{x}$ describes x
- $|u'| = O(1)$



6 Substrings of random strings

Lemma 8. *Substrings of random strings are almost random, even if the remainder of the string is given. Let*

$$x = x^1 x^2 x^3 \in \mathbb{B}^n \text{ be random}$$

Then

$$K(x^2|x^1x^3) \geq |x^2| - O(\log(n))$$

- let $u'z$ be a shortest description of x^2 given x^1x^3. i.e. M_u started with $z\#x^1x^3$ prints x^2 and halts

$$K(x^2|x^1x^3) = |u'z|$$

5 Conditional Kolmogorov complexity

question: How much does a given string y help to describe x?

def: $K(x|y)$ For $x, y \in \mathbb{B}^*$ we define

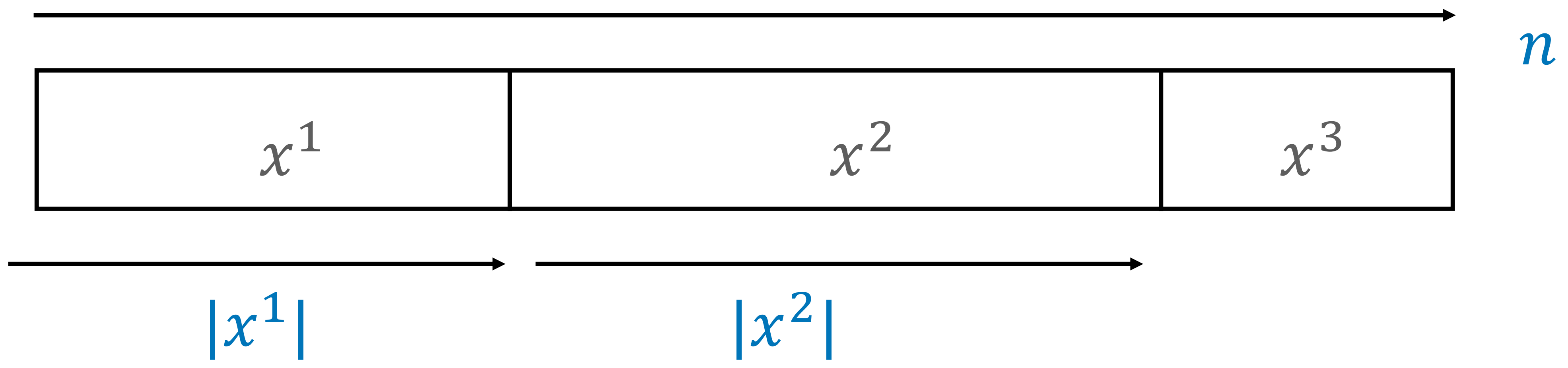
$$K(x|y) = \min\{u'v \ : \ M_u \text{started with } v\#y \text{ outputs } x \text{ and halts}\}$$

Here we can afford to indicate, what is given with $\# \notin \mathbb{B}$.

example:

Lemma 7. $K(x|\overline{x}) = O(1)$

- M_u started with y flips all bits of y and halts.
- $u'\#\overline{x}$ describes x
- $|u'| = O(1)$



6 Substrings of random strings

Lemma 8. *Substrings of random strings are almost random, even if the remainder of the string is given. Let*

$$x = x^1x^2x^3 \in \mathbb{B}^n \textit{ be random}$$

Then

$$K(x^2|x^1x^3) \geq |x^2| - O(\log(n))$$

- let $u'z$ be a shortest description of x^2 given x^1x^3. i.e. M_u started with $z\#x^1x^3$ prints x^2 and halts

$$K(x^2|x^1x^3) = |u'z|$$

- M_v started with $u'z'bin(|x^1|)'x^1x^3$
  1. generates $z\#x^1x^3$
  2. simulates M_u with input $z\#x^1x^3$
  3. outputs x^2
  4. inserts it behind x^1 to obtain $x^1x^2x^3$

5 Conditional Kolmogorov complexity

question: How much does a given string y help to describe x?

def: $K(x|y)$ For $x, y \in \mathbb{B}^*$ we define

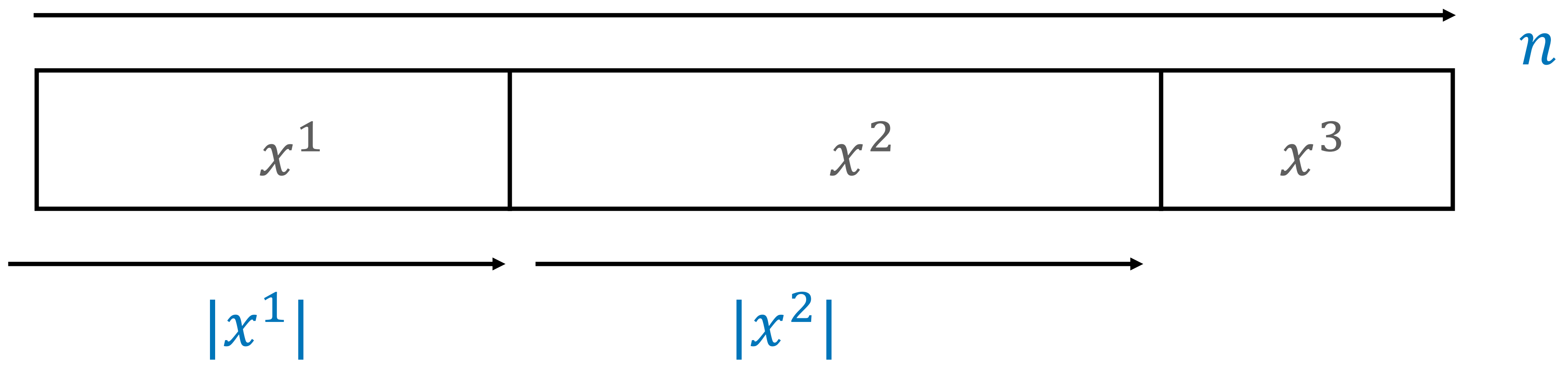
$$K(x|y) = \min\{u'v \ : \ M_u \text{started with } v\#y \text{ outputs } x \text{ and halts}\}$$

Here we can afford to indicate, what is given with $\# \notin \mathbb{B}$.

example:

Lemma 7. $K(x|\overline{x}) = O(1)$

- M_u started with y flips all bits of y and halts.
- $u'\#\overline{x}$ describes x
- $|u'| = O(1)$



6 Substrings of random strings

Lemma 8. *Substrings of random strings are almost random, even if the remainder of the string is given. Let*

$$x = x^1x^2x^3 \in \mathbb{B}^n \textit{ be random}$$

Then

$$K(x^2|x^1x^3) \geq |x^2| - O(\log(n))$$

- let $u'z$ be a shortest description of x^2 given x^1x^3. i.e. M_u started with $z\#x^1x^3$ prints x^2 and halts

$$K(x^2|x^1x^3) = |u'z|$$

- M_v started with $u'z'bin(|x^1|)'x^1x^3$
  1. generates $z\#x^1x^3$
  2. simulates M_u with input $z\#x^1x^3$
  3. outputs x^2
  4. inserts it behind x^1 to obtain $x^1x^2x^3$
-

$$\begin{aligned}
n &\leq K(x^1x^2x^3) \\
&\leq |v'u'z'bin(|x^1|)'x^1x^3| \\
&= O(1) + K(x^2|x^1x^3) + O(\log n) + |x^1x^3| \\
|x^2| - O(\log(n)) &\leq K(x^2|x^1x^3)
\end{aligned}$$

7 Crossing sequences

Hennie 1965

- exploiting communication bottleneck of 1-tape Turing machines
- can transport information across a cell boundary only in the state $z \in Z$

7 Crossing sequences

Hennie 1965

Fix 1-tape TM

$$M = (Z, \Sigma, \delta, z_0, Z_A)$$

and w.l.o.g assume that computations of M end with head on left end of tape inscription.

def: crossing sequence For $w \in A^*$ and numbers i of tape cells the crossing sequence $CS(w,i)$ is the sequence of states of M started with w before its head crosses the border between tape cells i and $i+1$

7 Crossing sequences

Hennie 1965

Fix 1-tape TM

$$M = (Z, \Sigma, \delta, z_0, Z_A)$$

and w.l.o.g assume that computations of M end with head on left end of tape inscription.

def: crossing sequence For $w \in A^*$ and numbers i of tape cells the crossing sequence $CS(w,i)$ is the sequence of states of M started with w before its head crosses the border between tape cells i and $i+1$

Lemma 9. *Let $u, v, x, y \in A^*$ and assume that uv and wx produce the same crossing sequences between them.*

$$CS(uv, |u|) = CS(wx, |w|)$$

Then

$$ux \in L(M) \leftrightarrow uv \in L(M)$$

Fix 1-tape TM

$$M = (Z, \Sigma, \delta, z_0, Z_A)$$

and w.l.o.g assume that computations of M end with head on left end of tape inscription.

def: crossing sequence For $w \in A^*$ and numbers i of tape cells the crossing sequence $CS(w,i)$ is the sequence of states of M started with w before its head crosses the border between tape cells i and $i+1$

Lemma 9. *Let $u,v,x,y \in A^*$ and assume that uv and wx produce the same crossing sequences between them.*

$$CS(uv, |u|) = CS(wx, |w|)$$

Then

$$ux \in L(M) \leftrightarrow uv \in L(M)$$

- consider head movements accepting computations of M with inputs uv and ux as shown in figures 1 and 2

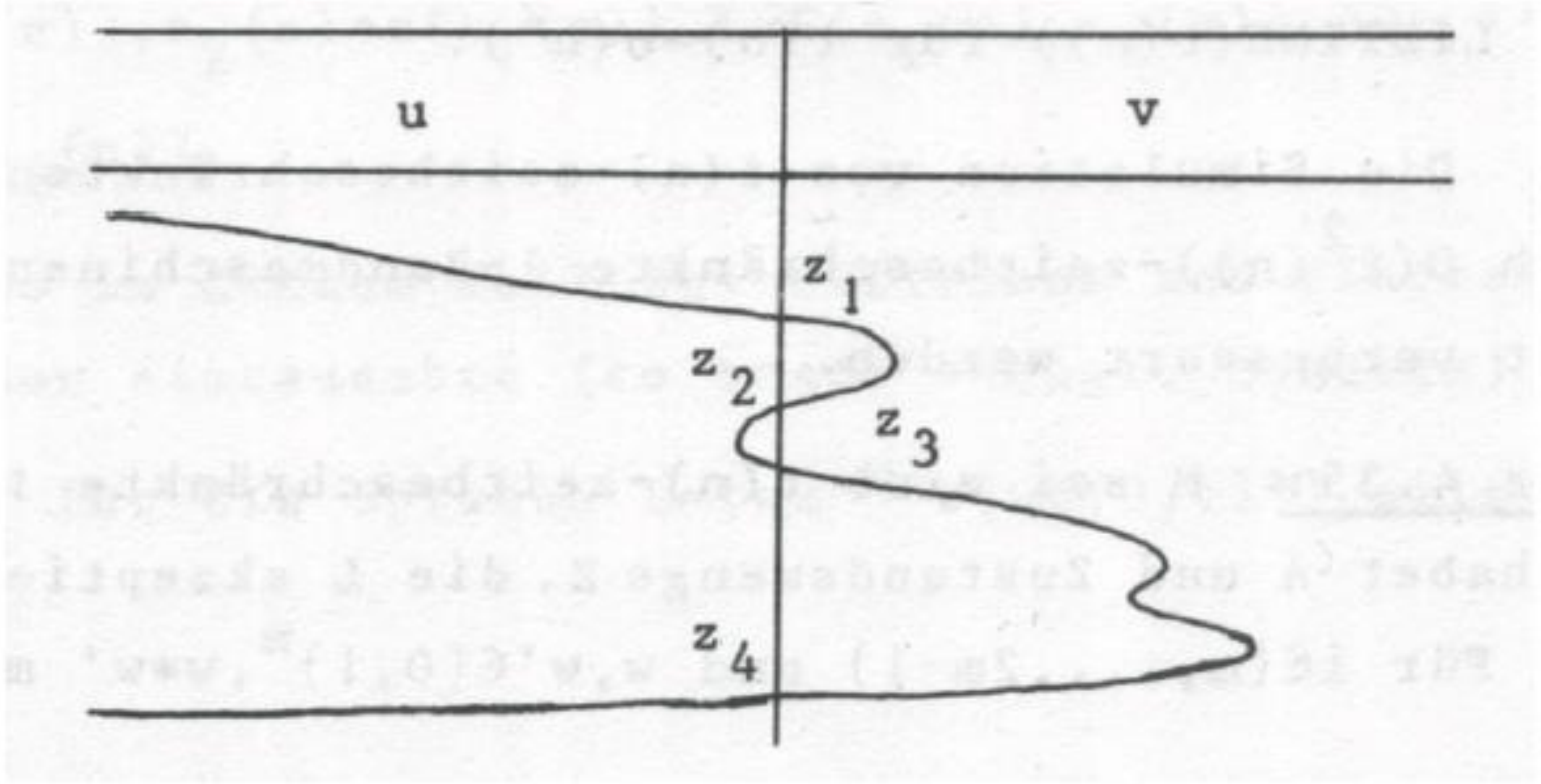


Figure 1: head movement and states of the computation with input uv. Time axis is pointing downward.

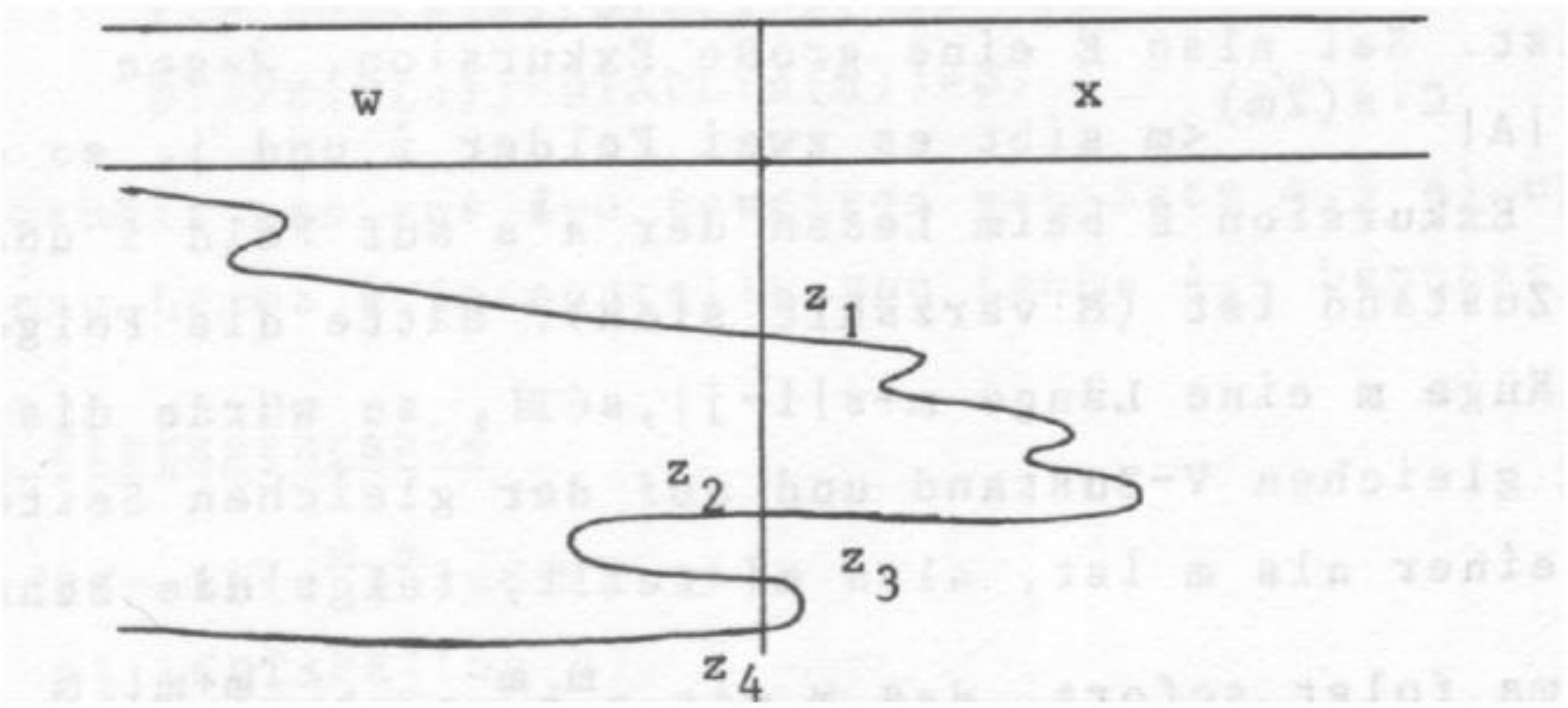


Figure 2: head movement and states of the computation with input wx

Fix 1-tape TM

$$M = (Z, \Sigma, \delta, z_0, Z_A)$$

and w.l.o.g assume that computations of M end with head on left end of tape inscription.

def: crossing sequence For $w \in A^*$ and numbers i of tape cells the crossing sequence $CS(w,i)$ is the sequence of states of M started with w before its head crosses the border between tape cells i and $i+1$

Lemma 9. *Let $u, v, x, y \in A^*$ and assume that uv and wx produce the same crossing sequences between them.*

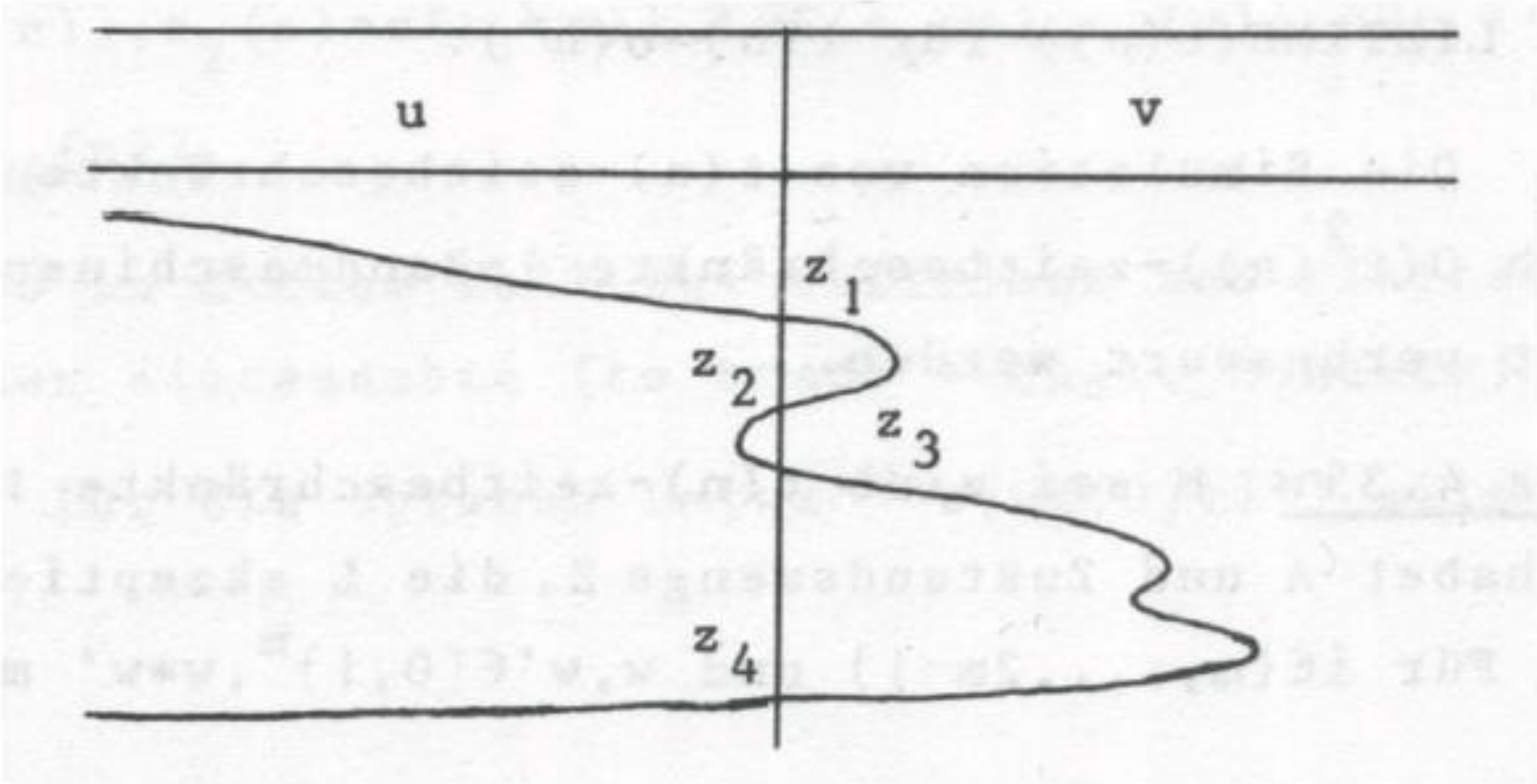
$$CS(uv, |u|) = CS(wx, |w|)$$

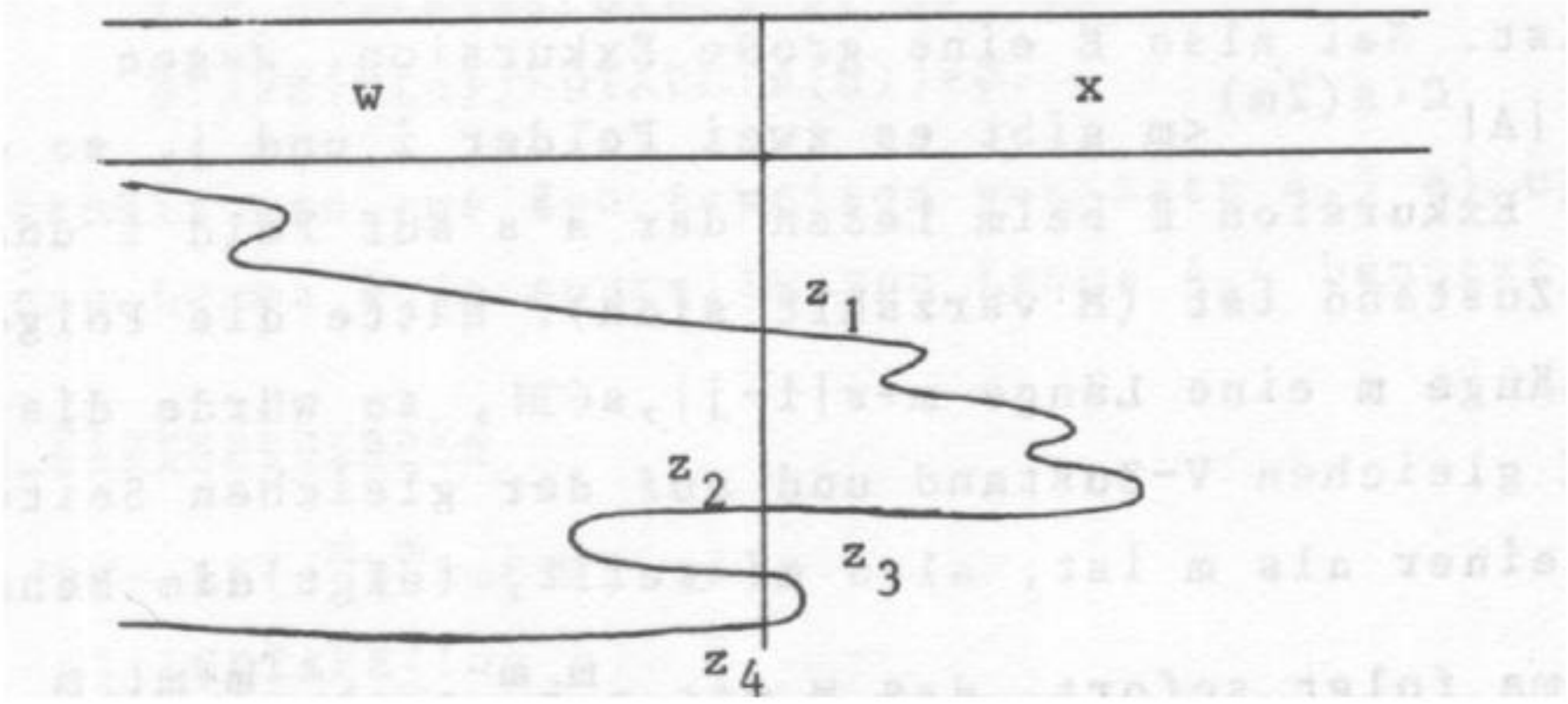
Then

$$ux \in L(M) \leftrightarrow uv \in L(M)$$

- consider head movements accepting computations of M with inputs uv and ux as shown in figures 1 and 2

- 'cut' at border and glue together as shown in figure 3 gives an computation of ux. The decision whether to accept is done on the left side and is the same as for uv





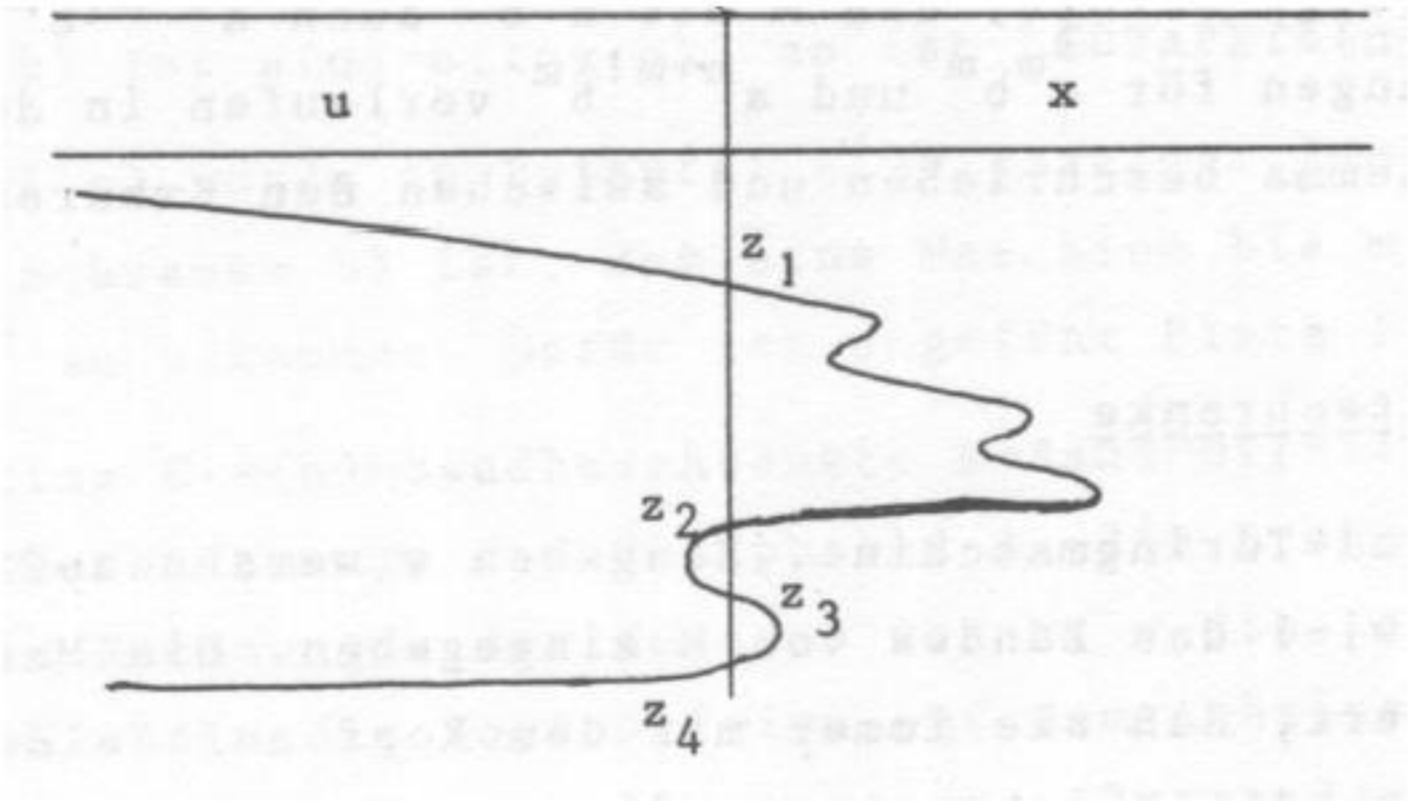


Figure 3: head movement and states of the computation with input ux

Lemma 10. *Let*

$$L = \{u\#^m u \ : \ u \in \mathbb{B}^m \ , \ m \in \mathbb{N}_0\}$$

Let M be a $t(n)$- time bounded 1-tape TM and $t(n) = o(n^2)$. Then M does not accept L

$$L(M) \neq L$$

Lemma 10. *Let*

$$L = \{u\#^m u \ : \ u \in \mathbb{B}^m \ , \ m \in \mathbb{N}_0\}$$

Let M be a $t(n)$- time bounded 1-tape TM and $t(n) = o(n^2)$. Then M does not accept L

$$L(M) \neq L$$

Assume $L = L(M)$

- For $u\#^m u \in L$ and $u'\#^m u' \in L$ with $u \neq u'$ and $i \in [m : 2m]$ crossing sequences at border i must differ

  $$CS(u\#^m u, i) \neq CS(u'\#^m u', i)$$

  otherwise by lemma 9

  $$u\#^m u' \in L(M)$$

Lemma 10. *Let*

$$L = \{u\#^m u \ : \ u \in \mathbb{B}^m \ , \ m \in \mathbb{N}_0\}$$

Let M be a $t(n)$- time bounded 1-tape TM and $t(n) = o(n^2)$. Then M does not accept L

$$L(M) \neq L$$

Assume $L = L(M)$

- For $u\#^m u \in L$ and $u'\#^m u' \in L$ with $u \neq u'$ and $i \in [m : 2m]$ crossing sequences at border i must differ

  $$CS(u\#^m u, i) \neq CS(u'\#^m u', i)$$

  otherwise by lemma 9

  $$u\#^m u' \in L(M)$$

1979 proof:

- code crossing sequences CS in binary in

  $$\tilde{CS}' \in \mathbb{B}^* \ , \ |\tilde{CS}| = \rho \cdot |CS| \ , \ \rho = \lceil \log |Z|) \rceil$$

Lemma 10. *Let*

$$L = \{u\#^m u \;:\; u \in \mathbb{B}^m \;,\; m \in \mathbb{N}_0\}$$

Let M be a $t(n)$- time bounded 1-tape TM and $t(n) = o(n^2)$. Then M does not accept L

$$L(M) \neq L$$

Assume $L = L(M)$

- For $u\#^m u \in L$ and $u'\#^m u' \in L$ with $u \neq u'$ and $i \in [m : 2m]$ crossing sequences at border i must differ

  $$CS(u\#^m u, i) \neq CS(u'\#^m u', i)$$

  otherwise by lemma 9

  $$u\#^m u' \in L(M)$$

1979 proof:

- code crossing sequences CS in binary in

  $$\tilde{CS}' \in \mathbb{B}^* \;,\; |\tilde{CS}| = \rho \cdot |CS| \;,\; \rho = \lceil \log |Z|) \rceil$$

- decription of $u \in \mathbb{B}^m$ by crossing sequence: TM M_v started with $bin(m)'bin(i)'\tilde{CS}$

  1. enumerates all $u \in \mathbb{B}^m$
  2. runs machine M with input $u\#^m u$ and observes if $\tilde{CS}$ appears as code of CS at position i

- then M_v started with $bin(m)'bin(i)'\tilde{CS}(u\#^m u, i)$ detects this crossing sequence for u and only for u. It outputs u and halts.

Lemma 10. *Let*

$$L = \{u\#^m u \ : \ u \in \mathbb{B}^m \ , \ m \in \mathbb{N}_0\}$$

Let M be a $t(n)$- time bounded 1-tape TM and $t(n) = o(n^2)$. Then M does not accept L

$$L(M) \neq L$$

Assume $L = L(M)$

- For $u\#^m u \in L$ and $u'\#^m u' \in L$ with $u \neq u'$ and $i \in [m : 2m]$ crossing sequences at border i must differ

$$CS(u\#^m u, i) \neq CS(u'\#^m u', i)$$

  otherwise by lemma 9

$$u\#^m u' \in L(M)$$

1979 proof:

- code crossing sequences CS in binary in

$$\tilde{CS}' \in \mathbb{B}^* \ , \ |\tilde{CS}| = \rho \cdot |CS| \ , \ \rho = \lceil \log |Z|) \rceil$$

- decription of $u \in \mathbb{B}^m$ by crossing sequence: TM M_v started with $bin(m)'bin(i)'\tilde{CS}$
  1. enumerates all $u \in \mathbb{B}^m$
  2. runs machine M with input $u\#^m u$ and observes if $\tilde{CS}$ appears as code of CS at position i
- then M_v started with $bin(m)'bin(i)'\tilde{CS}(u\#^m u, i)$ detects this crossing sequence for u and only for u. It outputs u and halts.
- $v'bin(m)'bin(i)'\tilde{CS}(u\#^m u, i)$ describes u. If u is random, we have

$$\begin{aligned} m &\leq K(u) \\ &\leq O(1) + O(\log m) + \rho \cdot |CS(u\#^m u, i)| \\ |CS(u\#^m u, i)| &\geq \frac{m - O(\log m)}{\rho} \\ &= \frac{m}{2\rho} \quad \text{for } m \geq m_0 \end{aligned}$$

Lemma 10. *Let*

$$L = \{u\#^m u \ : \ u \in \mathbb{B}^m \ , \ m \in \mathbb{N}_0\}$$

Let M be a $t(n)$- time bounded 1-tape TM and $t(n) = o(n^2)$. Then M does not accept L

$$L(M) \neq L$$

Assume $L = L(M)$

- For $u\#^m u \in L$ and $u'\#^m u' \in L$ with $u \neq u'$ and $i \in [m:2m]$ crossing sequences at border i must differ

$$CS(u\#^m u, i) \neq CS(u'\#^m u', i)$$

otherwise by lemma 9

$$u\#^m u' \in L(M)$$

1979 proof:

- code crossing sequences CS in binary in

$$\tilde{CS}' \in \mathbb{B}^* \ , \ |\tilde{CS}| = \rho \cdot |CS| \ , \ \rho = \lceil \log |Z|) \rceil$$

- decription of $u \in \mathbb{B}^m$ by crossing sequence: TM M_v started with $bin(m)'bin(i)'\tilde{CS}$
  1. enumerates all $u \in \mathbb{B}^m$
  2. runs machine M with input $u\#^m u$ and observes if $\tilde{CS}$ appears as code of CS at position i
- then M_v started with $bin(m)'bin(i)'\tilde{CS}(u\#^m u, i)$ detects this crossing sequence for u and only for u. It outputs u and halts.

- $v'bin(m)'bin(i)'\tilde{CS}(u\#^m u, i)$ describes u. If u is random, we have

$$\begin{aligned} m &\leq K(u) \\ &\leq O(1) + O(\log m) + \rho \cdot |CS(u\#^m u, i)| \\ |CS(u\#^m u, i)| &\geq \frac{m - O(\log m)}{\rho} \\ &= \frac{m}{2\rho} \quad \text{for } m \geq m_0 \end{aligned}$$

- run time is at least sum of lengths of crossing sequences. Let $n = 3m$. Then run time T_n with random $u \in \mathbb{B}^m$ is

$$\begin{aligned} T_n &\geq \sum_{i=m}^{2m} |CS(u\#^m u, i)| \\ &\geq m \cdot m/(2\rho) \\ &= n^2/(18\rho) \end{aligned}$$