Some Thoughts on Normality, Algorithmic Randomness, and Analog Computing

Xiang Huang, Ph.D. Candidate

March 29, 2019

Computer Science Department, Iowa State University

A lunch talk

Where to draw the line between Mathematics/Logic and Theoretical Computer Science (TCS)?

- TCS people care about *how fast* one can achieve something.

A lunch talk

Where to draw the line between Mathematics/Logic and Theoretical Computer Science (TCS)?

- TCS people care about *how fast* one can achieve something.
- TCS people care about *what resources* one uses to achieve something.

A lunch talk

Where to draw the line between Mathematics/Logic and Theoretical Computer Science (TCS)?

- TCS people care about *how fast* one can achieve something.
- TCS people care about *what resources* one uses to achieve something.
- Anything beyond computability is not what TCS people care about.

A lunch talk

Where to draw the line between Mathematics/Logic and Theoretical Computer Science (TCS)?

- TCS people care about *how fast* one can achieve something.
- TCS people care about *what resources* one uses to achieve something.
- Anything beyond computability is not what TCS people care about.
- Anyone who talks about polynomial time is not a mathematician.

It is just some lunch talk. Don't look too much into it. But there is some truth in it.

A lunch talk

Where to draw the line between Mathematics/Logic and Theoretical Computer Science (TCS)?

- TCS people care about *how fast* one can achieve something.
- TCS people care about *what resources* one uses to achieve something.
- Anything beyond computability is not what TCS people care about.
- Anyone who talks about polynomial time is not a mathematician.

It is just some lunch talk. Don't look too much into it. But there is some truth in it.

A repeated theme in this talk is to demonstrate research areas on how we take care of the "**how fast**" and "**what resources**" concerns.

Outline

1. Normality and Automata

Outline

1. Normality and Automata
2. Algorithmic Randomness

Outline

1. Normality and Automata
2. Algorithmic Randomness
3. Analog Computing

Normality and Automata

Simply normal

Let's consider an infinite sequence of 0's and 1's, obtained by tossing a fair coin:

Simply normal

Let's consider an infinite sequence of 0's and 1's, obtained by tossing a fair coin:

the sequence: $1\ 1\ 1\ 1\ 1\ 1\ \cdots$

Simply normal

Let's consider an infinite sequence of 0's and 1's, obtained by tossing a fair coin:

the sequence: $1\ 1\ 1\ 1\ 1\ 1\ \cdots$ is not "**normal**".

Simply normal

Let's consider an infinite sequence of 0's and 1's, obtained by tossing a fair coin:

the sequence: $1\ 1\ 1\ 1\ 1\ 1\ \cdots$ is not "**normal**".

Then what it means by saying something is "normal"?

Simply normal

Let's consider an infinite sequence of 0's and 1's, obtained by tossing a fair coin:

the sequence: $1\ 1\ 1\ 1\ 1\ 1\ \cdots$ is not "**normal**".

Then what it means by saying something is "normal"?

Well, how about "having (roughly) the same number of 0's and 1's?"

Simply normal

Let's consider an infinite sequence of 0's and 1's, obtained by tossing a fair coin:

the sequence: $1\ 1\ 1\ 1\ 1\ 1\ \cdots$ is not "**normal**".

Then what it means by saying something is "normal"?

Well, how about "having (roughly) the same number of 0's and 1's?"

That is, *the frequency of 0's* $=$ *the frequency of 1's* $= 1/2$.

Simply normal

Let's consider an infinite sequence of 0's and 1's, obtained by tossing a fair coin:

the sequence: $1\ 1\ 1\ 1\ 1\ 1\ \cdots$ is not "**normal**".

Then what it means by saying something is "normal"?

Well, how about "having (roughly) the same number of 0's and 1's?"

That is, *the frequency of 0's* $=$ *the frequency of 1's* $= 1/2$.

We call such a property **simply normal**.

Beyond simply normal

Consider the sequence x=01 01 01 01 ...

Beyond simply normal

Consider the sequence x=$\underline{01}\ \underline{01}\ \underline{01}\ \underline{01}$ ...

You won't believe someone get it by tossing a fair coin,

Beyond simply normal

Consider the sequence x=$\underline{01}\ \underline{01}\ \underline{01}\ \underline{01}$ ...

You won't believe someone get it by tossing a fair coin, because the blocks "00" and "11" never occur!

Beyond simply normal

Consider the sequence x=$\underline{01}\ \underline{01}\ \underline{01}\ \underline{01}$ ...

You won't believe someone get it by tossing a fair coin, because the blocks "00" and "11" never occur!

Let's apply our previous idea again to rule out this sequence. That is to say, we require a "normal" sequence must have

$$\text{Freq}(00) \approx \text{Freq}\ (01) \approx \text{Freq}\ (10) \approx \text{Freq}\ (11) \approx 1/4 = 1/2^2.$$

Beyond simply normal

Consider the sequence x=$\underline{01}\ \underline{01}\ \underline{01}\ \underline{01}$ ...

You won't believe someone get it by tossing a fair coin, because the blocks "00" and "11" never occur!

Let's apply our previous idea again to rule out this sequence. That is to say, we require a "normal" sequence must have

$$\text{Freq}(00) \approx \text{Freq}\ (01) \approx \text{Freq}\ (10) \approx \text{Freq}\ (11) \approx 1/4 = 1/2^2.$$

But why do we stop at considering length 2 blocks? We can do

$$\text{Freq}(000) \approx \text{Freq}\ (001) \approx \cdots \approx \text{Freq}\ (111) \approx 1/8 = 1/2^3.$$

Beyond simply normal

Consider the sequence x=$\underline{01}\ \underline{01}\ \underline{01}\ \underline{01}$ ...

You won't believe someone get it by tossing a fair coin, because the blocks "00" and "11" never occur!

Let's apply our previous idea again to rule out this sequence. That is to say, we require a "normal" sequence must have

$$\text{Freq}(00) \approx \text{Freq}\ (01) \approx \text{Freq}\ (10) \approx \text{Freq}\ (11) \approx 1/4 = 1/2^2.$$

But why do we stop at considering length 2 blocks? We can do

$$\text{Freq}(000) \approx \text{Freq}\ (001) \approx \cdots \approx \text{Freq}\ (111) \approx 1/8 = 1/2^3.$$

$$\text{Freq}(0000) \approx \text{Freq}\ (0001) \approx \cdots \approx \text{Freq}\ (1111) \approx 1/16 = 1/2^4.$$

and so on. Note that 2, 3, 4 are the **length**s of the blocks.

Normal sequences/numbers

Definition (Normality[1])
A sequence x is normal, if for any length ℓ, and any block of string s of length ℓ, Freq(s, x)$=\frac{1}{2^\ell}$.

Remark
If you want to talk about normality of base-10 numbers, just change the right-hand side of the above to $\frac{1}{10^\ell}$*, and only focus on the* ***fraction part*** *of the number.*

[1] Borel, E. (1909), "Les probabilités dénombrables et leurs applications arithmétiques".

Some facts about normal numbers

- Champernowne constant[2]: 0.12345678910111213141516... is normal. Or similarly in binary it is 0.$\underline{0}$ $\underline{1}$ $\underline{00}$ $\underline{01}$ $\underline{10}$ $\underline{11}$...

[2]Champernowne, D. G. (1933), "The construction of decimals normal in the scale of ten."

[3]Copeland, A. H.; Erdös, P. (1946), "Note on Normal Numbers".

Some facts about normal numbers

- Champernowne constant[2]: 0.12345678910111213141516... is normal. Or similarly in binary it is 0.$\underline{0}$ $\underline{1}$ $\underline{00}$ $\underline{01}$ $\underline{10}$ $\underline{11}$...
- Copeland–Erdös number[3]: $0.2\ 3\ 5\ 7\ 11\ 13\ 17\cdots$ is normal.

[2]Champernowne, D. G. (1933), "The construction of decimals normal in the scale of ten."

[3]Copeland, A. H.; Erdös, P. (1946), "Note on Normal Numbers".

Some facts about normal numbers

- Champernowne constant[2]: 0.12345678910111213141516... is normal. Or similarly in binary it is 0.$\underline{0}$ $\underline{1}$ $\underline{00}$ $\underline{01}$ $\underline{10}$ $\underline{11}$...
- Copeland–Erdös number[3]: $0.2\ 3\ 5\ 7\ 11\ 13\ 17\cdots$ is normal.
- **Almost all** real numbers are normal numbers.

[2]Champernowne, D. G. (1933), "The construction of decimals normal in the scale of ten."
[3]Copeland, A. H.; Erdös, P. (1946), "Note on Normal Numbers".

Some facts about normal numbers

- Champernowne constant[2]: 0.12345678910111213141516... is normal. Or similarly in binary it is 0.$\underline{0}$ $\underline{1}$ $\underline{00}$ $\underline{01}$ $\underline{10}$ $\underline{11}$...
- Copeland–Erdös number[3]: 0.2 3 5 7 11 13 17 $\cdots$ is normal.
- **Almost all** real numbers are normal numbers.
- **Open problem:** $\sqrt{2}, \pi, e$ normal?

[2]Champernowne, D. G. (1933), "The construction of decimals normal in the scale of ten."

[3]Copeland, A. H.; Erdös, P. (1946), "Note on Normal Numbers".

Some facts about normal numbers

- Champernowne constant[2]: 0.12345678910111213141516... is normal. Or similarly in binary it is 0.$\underline{0}$ $\underline{1}$ $\underline{00}$ $\underline{01}$ $\underline{10}$ $\underline{11}$...
- Copeland–Erdös number[3]: 0.2 3 5 7 11 13 17 $\cdots$ is normal.
- **Almost all** real numbers are normal numbers.
- **Open problem:** $\sqrt{2}, \pi, e$ normal?
  - Some popular culture e.g. movies, say that π contains everything (if converted into ASCII text):

[2]Champernowne, D. G. (1933), "The construction of decimals normal in the scale of ten."

[3]Copeland, A. H.; Erdös, P. (1946), "Note on Normal Numbers".

Some facts about normal numbers

- Champernowne constant[2]: 0.12345678910111213141516... is normal. Or similarly in binary it is 0.$\underline{0}$ $\underline{1}$ $\underline{00}$ $\underline{01}$ $\underline{10}$ $\underline{11}$...
- Copeland–Erdös number[3]: 0.2 3 5 7 11 13 17 $\cdots$ is normal.
- **Almost all** real numbers are normal numbers.
- **Open problem:** $\sqrt{2}, \pi, e$ normal?
  - Some popular culture e.g. movies, say that π contains everything (if converted into ASCII text):
    - the name of every person you will ever love
    - the date, time and manner of your death
    - the answers to all the great questions of the universe

[2]Champernowne, D. G. (1933), "The construction of decimals normal in the scale of ten."

[3]Copeland, A. H.; Erdös, P. (1946), "Note on Normal Numbers".

Some facts about normal numbers

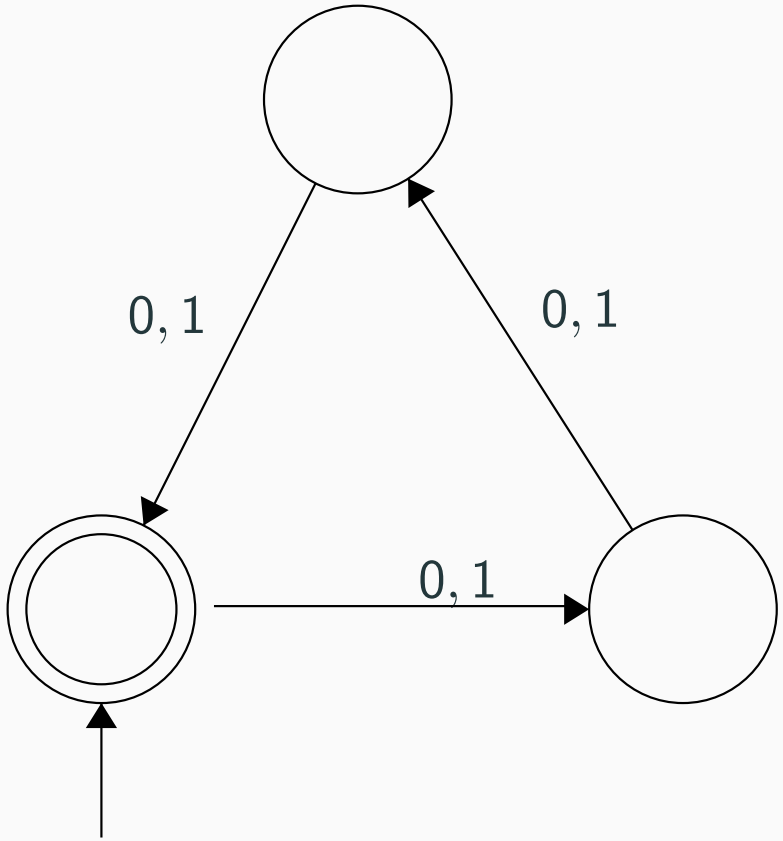
- Champernowne constant[2]: 0.12345678910111213141516... is normal. Or similarly in binary it is 0.$\underline{0}$ $\underline{1}$ $\underline{00}$ $\underline{01}$ $\underline{10}$ $\underline{11}$...
- Copeland–Erdös number[3]: 0.2 3 5 7 11 13 17 $\cdots$ is normal.
- **Almost all** real numbers are normal numbers.
- **Open problem:** $\sqrt{2}, \pi, e$ normal?
  - Some popular culture e.g. movies, say that π contains everything (if converted into ASCII text):
    - the name of every person you will ever love
    - the date, time and manner of your death
    - the answers to all the great questions of the universe
  - All these just simply say that, π is normal. But we don't really know that, yet.

[2]Champernowne, D. G. (1933), "The construction of decimals normal in the scale of ten."

[3]Copeland, A. H.; Erdös, P. (1946), "Note on Normal Numbers".

Finite State Selector

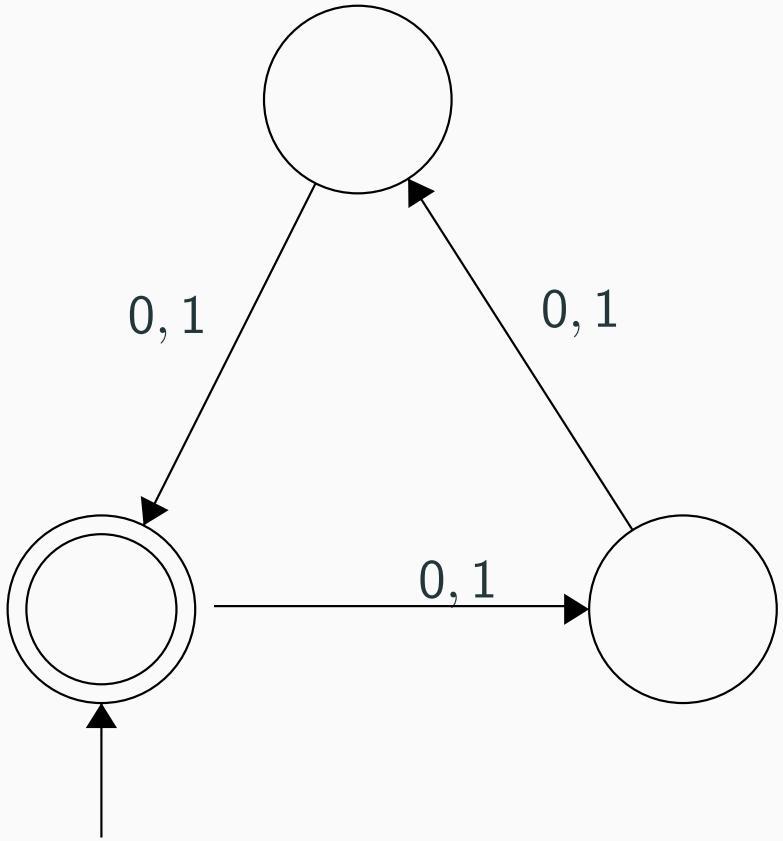
Rule of the game: hit green (accepting state), spit next bit.



$C_2 = \mathbf{0}10\mathbf{0}01\mathbf{1}01\mathbf{1}000\cdots \Longrightarrow 0\ 0\ 1\ 1\cdots$

Finite State Selector

Rule of the game: hit green (accepting state), spit next bit.



$C_2 = \mathbf{0}10\mathbf{0}01\mathbf{1}01\mathbf{1}000\cdots \Longrightarrow 0\ 0\ 1\ 1\cdots$

Question: Is the resulting sequence normal?

Agafonov's Theorem

Theorem (Agafonov[4])
Finite state automata preserves normality.

[4] Agafonov, V. N. (1968), "Normal sequences and finite automata".

Selection: Cont.

Definition (Selection Rule)
A selection rule S is a subset of $\{0,1\}^*$.

- Whenever a prefix of x falls in S, we select the next digit.

Selection: Cont.

Definition (Selection Rule)
A selection rule S is a subset of $\{0,1\}^*$.

- Whenever a prefix of x falls in S, we select the next digit.
- We only discuss normality of the resulting subsequence when it has infinitely many digits.

Selection: Cont.

Definition (Selection Rule)
A selection rule S is a subset of $\{0,1\}^*$.

- Whenever a prefix of x falls in S, we select the next digit.
- We only discuss normality of the resulting subsequence when it has infinitely many digits.
- Agafonov theorem says that **regular** selection rules preserve normality.

Preservation of Normality: Kamae-Weiss language

Let $S \subseteq \{0,1\}^*$ be a selection rule. We define an equivalence relation (Myhill-Nerode Relation) $\equiv_S$ on $\{0,1\}^*$ as follows:

$$u \equiv_S v \quad \text{if} \quad \{w \mid uw \in S\} = \{w \mid vw \in S\}.$$

Preservation of Normality: Kamae-Weiss language

Let $S \subseteq \{0,1\}^*$ be a selection rule. We define an equivalence relation (Myhill-Nerode Relation) $\equiv_S$ on $\{0,1\}^*$ as follows:

$$u \equiv_S v \quad \text{if} \quad \{w \mid uw \in S\} = \{w \mid vw \in S\}.$$

Consider the quotient set $S/\equiv_S$ and $\{0,1\}^*/\equiv_S$, we have

Preservation of Normality: Kamae-Weiss language

Let $S \subseteq \{0,1\}^*$ be a selection rule. We define an equivalence relation (Myhill-Nerode Relation) $\equiv_S$ on $\{0,1\}^*$ as follows:

$$u \equiv_S v \quad \text{if} \quad \{w \mid uw \in S\} = \{w \mid vw \in S\}.$$

Consider the quotient set $S/\equiv_S$ and $\{0,1\}^*/\equiv_S$, we have

Definition (Kamae-Weiss language)
If $|S/\equiv_S| < \infty$, then we say the language S is a Kamae-Weiss language.

Preservation of Normality: Kamae-Weiss language

Let $S \subseteq \{0,1\}^*$ be a selection rule. We define an equivalence relation (Myhill-Nerode Relation) $\equiv_S$ on $\{0,1\}^*$ as follows:

$$u \equiv_S v \quad \text{if} \quad \{w \mid uw \in S\} = \{w \mid vw \in S\}.$$

Consider the quotient set $S/\equiv_S$ and $\{0,1\}^*/\equiv_S$, we have

Definition (Kamae-Weiss language)
If $|S/\equiv_S| < \infty$, then we say the language S is a Kamae-Weiss language.

Remark
Note that if $\Sigma^*/\equiv_S$ is also finite, then S is regular, i.e., it can be recognized by a finite state automaton.

Preservation of Normality: Kamae-Weiss language

Let $S \subseteq \{0,1\}^*$ be a selection rule. We define an equivalence relation (Myhill-Nerode Relation) $\equiv_S$ on $\{0,1\}^*$ as follows:

$$u \equiv_S v \quad \text{if} \quad \{w \mid uw \in S\} = \{w \mid vw \in S\}.$$

Consider the quotient set $S/\equiv_S$ and $\{0,1\}^*/\equiv_S$, we have

Definition (Kamae-Weiss language)
If $|S/\equiv_S| < \infty$, then we say the language S is a Kamae-Weiss language.

Remark
Note that if $\Sigma^*/\equiv_S$ is also finite, then S is regular, i.e., it can be recognized by a finite state automaton.

Remark
Regular Languages $\subsetneq$ Kamae-Weiss Languages

Kamae-Weiss languages preserves normality

Theorem (Kamae and Weiss 1975[5])
Let S be a selection rule. If $S/\equiv_S$ is a finite set, then for any normal number x, if digits in x get selected "frequently", then the resulting subsequence, x_S, is also normal.

[5] *Normal Numbers and Selection Rules*
[6] Alexander Shen (2017), "Automatic Kolmogorov Complexity and Normality Revisited"

Kamae-Weiss languages preserves normality

Theorem (Kamae and Weiss 1975[5])
Let S be a selection rule. If $S/\equiv_S$ is a finite set, then for any normal number x, if digits in x get selected "frequently", then the resulting subsequence, x_S, is also normal.

Potential project:

[5] *Normal Numbers and Selection Rules*
[6] Alexander Shen (2017), "Automatic Kolmogorov Complexity and Normality Revisited"

Kamae-Weiss languages preserves normality

Theorem (Kamae and Weiss 1975[5])
Let S be a selection rule. If $S/\equiv_S$ is a finite set, then for any normal number x, if digits in x get selected "frequently", then the resulting subsequence, x_S, is also normal.

Potential project:

- Proved by complicated ergodic theory argument.

[5] *Normal Numbers and Selection Rules*
[6] Alexander Shen (2017), "Automatic Kolmogorov Complexity and Normality Revisited"

Kamae-Weiss languages preserves normality

Theorem (Kamae and Weiss 1975[5])
Let S be a selection rule. If $S/\equiv_S$ is a finite set, then for any normal number x, if digits in x get selected "frequently", then the resulting subsequence, x_S, is also normal.

Potential project:

- Proved by complicated ergodic theory argument.
- Wanted: a combinatorial proof.

[5] *Normal Numbers and Selection Rules*
[6] Alexander Shen (2017), "Automatic Kolmogorov Complexity and Normality Revisited"

Kamae-Weiss languages preserves normality

Theorem (Kamae and Weiss 1975[5])
Let S be a selection rule. If $S/\equiv_S$ is a finite set, then for any normal number x, if digits in x get selected "frequently", then the resulting subsequence, x_S, is also normal.

Potential project:

- Proved by complicated ergodic theory argument.
- Wanted: a combinatorial proof.
- There is a "combinatorial" proof[6] for Agafonov theorem, adapt that?

[5] *Normal Numbers and Selection Rules*
[6] Alexander Shen (2017), "Automatic Kolmogorov Complexity and Normality Revisited"

Selection rule that does not preserve normality

Consider, again, the $C_2 = 0.0\ 1\ 00\ 01\ 10\ 11\ 000 \cdots 111\ 0000 \cdots$.

[7] Wolfgang Merkle and Jan Reimann (2005), " Selection Functions that Do Not Preserve Normality."

Selection rule that does not preserve normality

Consider, again, the $C_2 = 0.0\ 1\ 00\ 01\ 10\ 11\ 000 \cdots 111\ 0000 \cdots$.

Let $S = \{s \in \{0,1\}^* \mid \#(0,s) = \#(1,s)\}$ be the selection rule. That is, whenever we run into a prefix with the same number of 0's and 1's, we select the next bit.

[7] Wolfgang Merkle and Jan Reimann (2005), " Selection Functions that Do Not Preserve Normality."

Selection rule that does not preserve normality

Consider, again, the $C_2 = 0.0\ 1\ 00\ 01\ 10\ 11\ 000 \cdots 111\ 0000 \cdots$.

Let $S = \{s \in \{0,1\}^* \mid \#(0,s) = \#(1,s)\}$ be the selection rule. That is, whenever we run into a prefix with the same number of 0's and 1's, we select the next bit.

What do we get after selection?

[7] Wolfgang Merkle and Jan Reimann (2005), " Selection Functions that Do Not Preserve Normality."

Selection rule that does not preserve normality

Consider, again, the $C_2 = 0.0\ 1\ 00\ 01\ 10\ 11\ 000 \cdots 111\ 0000 \cdots$.

Let $S = \{s \in \{0,1\}^* \mid \#(0,s) = \#(1,s)\}$ be the selection rule. That is, whenever we run into a prefix with the same number of 0's and 1's, we select the next bit.

What do we get after selection?

Well, $C_{2,S} = 0.0\ 0\ 0\ \cdots$, which is obviously not normal[7].

[7] Wolfgang Merkle and Jan Reimann (2005), " Selection Functions that Do Not Preserve Normality."

Selection rule that does not preserve normality

Consider, again, the $C_2 = 0.0\ 1\ 00\ 01\ 10\ 11\ 000 \cdots 111\ 0000 \cdots$.

Let $S = \{s \in \{0,1\}^* \mid \#(0,s) = \#(1,s)\}$ be the selection rule. That is, whenever we run into a prefix with the same number of 0's and 1's, we select the next bit.

What do we get after selection?

Well, $C_{2,S} = 0.0\ 0\ 0\ \cdots$, which is obviously not normal[7].

Remark
Note that S can be recognized by a stack machine (pushdown automata). Hence this says stack machines do not preserve normality.

[7] Wolfgang Merkle and Jan Reimann (2005), " Selection Functions that Do Not Preserve Normality."

An open problem

What is the largest class of languages (as selectors) that preserves normality?

- No weaker than finite state machines
- It contains some Kamae-Weiss languages.
- Weaker than stack machine.

Remark
This is a question asking what is the right amount of resources we need to preserve normality.

Finite-state gambler and strong dichotomy

Finite-state Gambler

Definition
A *finite-state gambler (FSG)* is a 4-tuple

$$G = (Q, \delta, s, B),$$

where Q is a finite set of *states*, $\delta : Q \times \Sigma \to Q$ is the *transition function*, $s \in Q$ is the *start state*, and $B : Q \to \Delta_{\mathbb{Q}}(\Sigma)$ is the *betting function*.

Finite-state Gambler

Definition
A *finite-state gambler (FSG)* is a 4-tuple

$$G = (Q, \delta, s, B),$$

where Q is a finite set of *states*, $\delta : Q \times \Sigma \to Q$ is the *transition function*, $s \in Q$ is the *start state*, and $B : Q \to \Delta_{\mathbb{Q}}(\Sigma)$ is the *betting function*.

The payoffs in the betting are determined by a *payoff probability measure* $\alpha \in \Delta(\Sigma)$.

Finite-state Gambler

Definition
A *finite-state gambler (FSG)* is a 4-tuple

$$G = (Q, \delta, s, B),$$

where Q is a finite set of *states*, $\delta : Q \times \Sigma \to Q$ is the *transition function*, $s \in Q$ is the *start state*, and $B : Q \to \Delta_{\mathbb{Q}}(\Sigma)$ is the *betting function*.

The payoffs in the betting are determined by a *payoff probability measure* $\alpha \in \Delta(\Sigma)$.

We write $d_{G,\alpha}(w)$ for the gambler G's capital (amount of money) after betting on the successive bits of a prefix $w \sqsubseteq S$, and we assume that the initial capital is $d_{G,\alpha}(\lambda) = 1$.

Finite-state Gambler

Definition
A *finite-state gambler (FSG)* is a 4-tuple

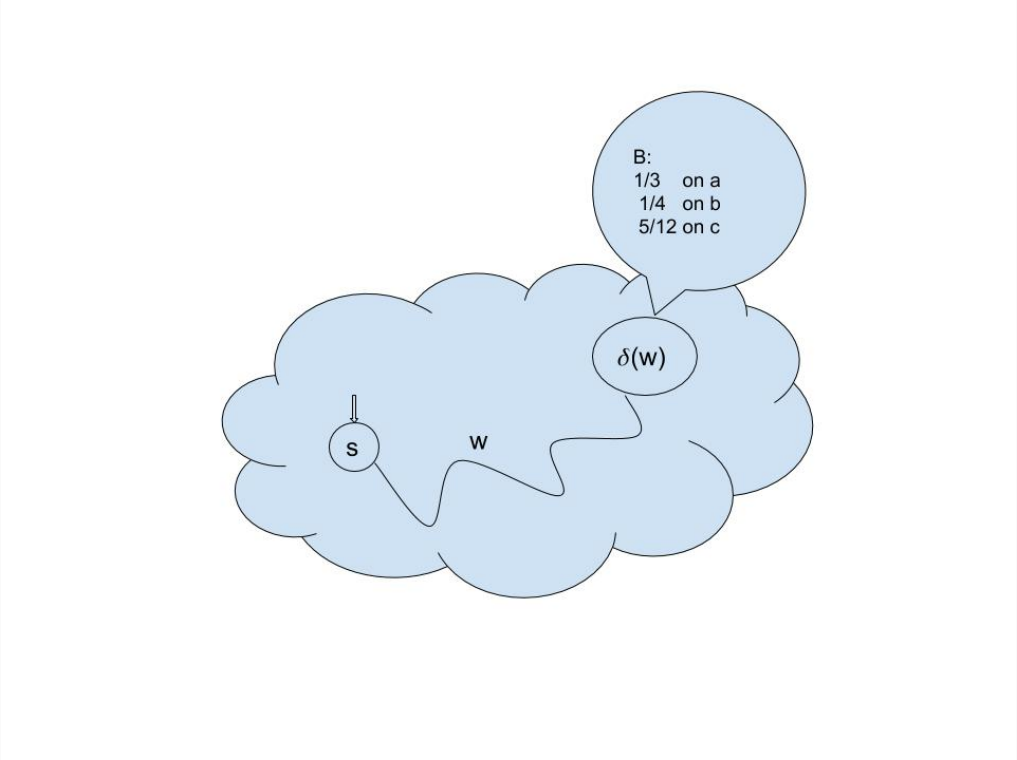
$$G = (Q, \delta, s, B),$$

where Q is a finite set of *states*, $\delta : Q \times \Sigma \to Q$ is the *transition function*, $s \in Q$ is the *start state*, and $B : Q \to \Delta_{\mathbb{Q}}(\Sigma)$ is the *betting function*.

The payoffs in the betting are determined by a *payoff probability measure* $\alpha \in \Delta(\Sigma)$.

We write $d_{G,\alpha}(w)$ for the gambler G's capital (amount of money) after betting on the successive bits of a prefix $w \sqsubseteq S$, and we assume that the initial capital is $d_{G,\alpha}(\lambda) = 1$.

Finite-state Gambler



Let our capital after seeing $w \sqsubseteq S$ be $d_{G,\alpha}(w)$. The gambler's bets $B(\delta(w))(a)$ of the money that the next symbol of S is an a. If it then turns out to be the case, the gambler's capital will be

$$d_{G,\alpha}(wa) = d_{G,\alpha}(w)\frac{B\big(\delta(w)\big)(a)}{\alpha(a)}. \tag{1}$$

Schnorr-Stimm Dichotomy Theorem

Theorem (Schnorr and Stimm 72')

1. *If S is not normal, then there is a finite-state gambler that wins money at an infinitely-often exponential rate betting on S.*
2. *If S is normal, then any finite-state gambler betting on S loses money at an exponential rate betting on S.*

Schnorr-Stimm Dichotomy Theorem

Theorem (Schnorr and Stimm 72')

1. *If S is not normal, then there is a finite-state gambler that wins money at an infinitely-often exponential rate betting on S.*
2. *If S is normal, then any finite-state gambler betting on S loses money at an exponential rate betting on S.*

Question: What are these exponential rates? How to interpret them?

Information Theory: a Short Introduction

Self-information

A long time ago, in a galaxy far, far away...

Self-information

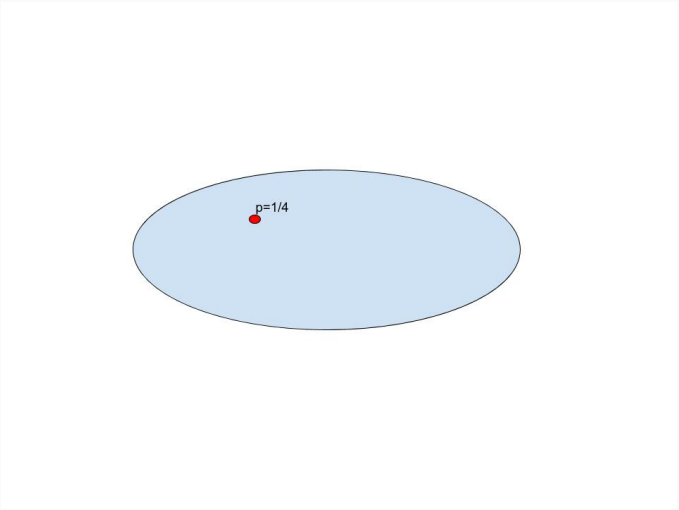
A long time ago, in a galaxy far, far away...

There is a point. And he was told he got $p = 1/4$ chance selected in some process.

Self-information

A long time ago, in a galaxy far, far away...

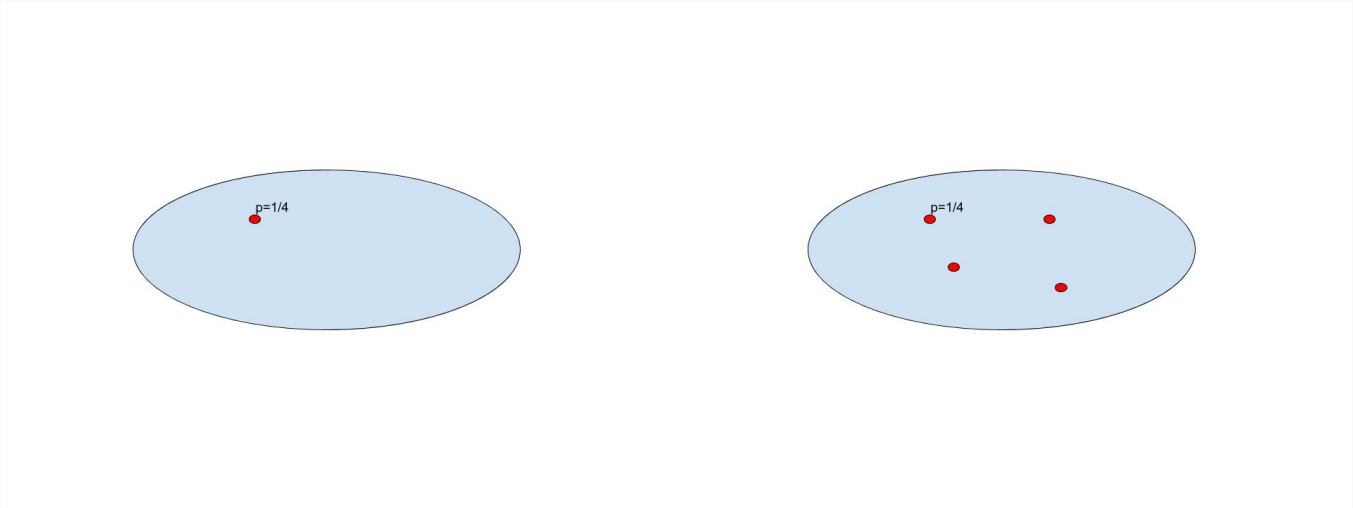
There is a point. And he was told he got $p = 1/4$ chance selected in some process.



Self-information

A long time ago, in a galaxy far, far away...

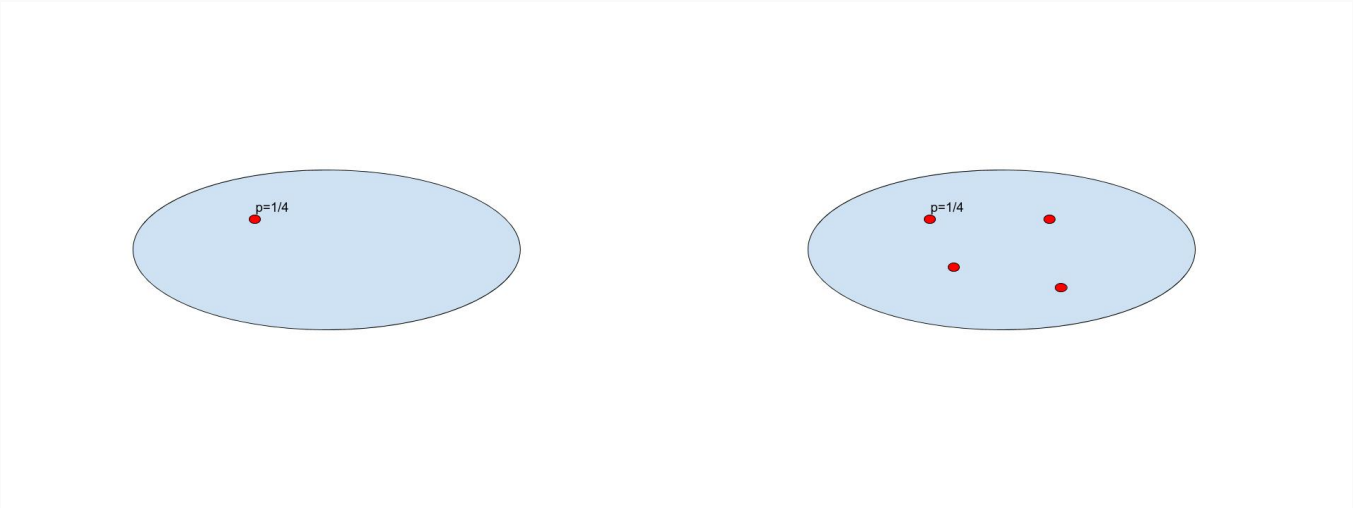
There is a point. And he was told he got $p = 1/4$ chance selected in some process.



Self-information

A long time ago, in a galaxy far, far away...

There is a point. And he was told he got $p = 1/4$ chance selected in some process.

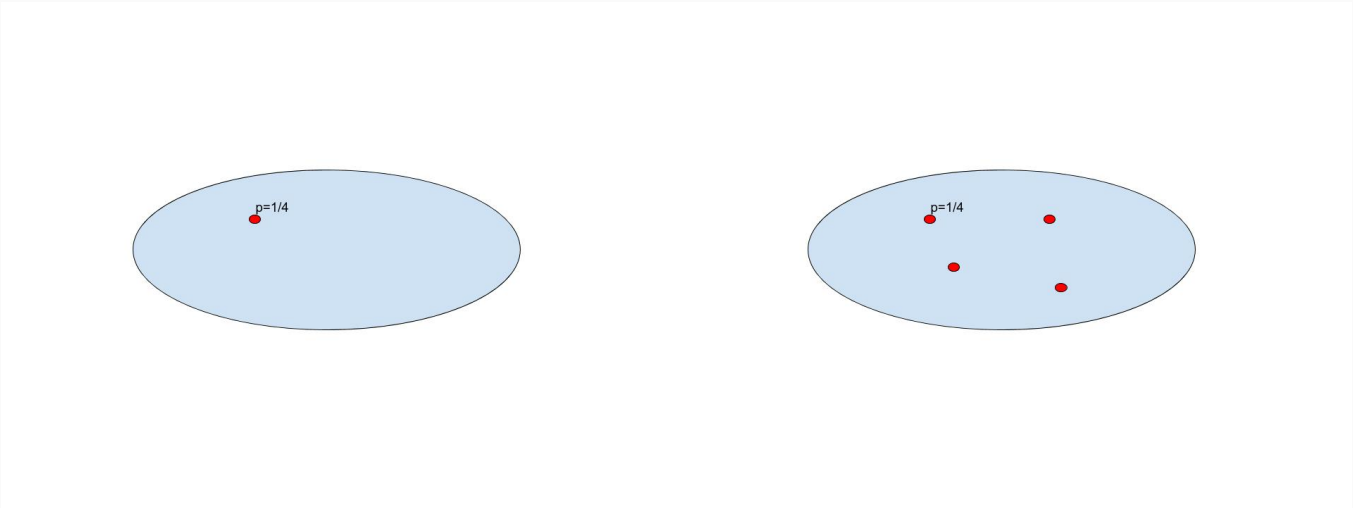


So he therefore believes he needs to have $\log \frac{1}{1/4} = 2$ bits to encode the objects in his galaxy. He can encode the four points “00”, “01”, “10” and “11”. And that is only **his** opinion.

Self-information

A long time ago, in a galaxy far, far away...

There is a point. And he was told he got $p = 1/4$ chance selected in some process.



So he therefore believes he needs to have $\log \frac{1}{1/4} = 2$ bits to encode the objects in his galaxy. He can encode the four points "00", "01", "10" and "11". And that is only **his** opinion.

We call $\log \frac{1}{p}$ this point's *self-information*.

Entropy

A *(discrete) probability measure* on a nonempty finite set Ω is a function $\pi : \Omega \to [0, 1]$ satisfying

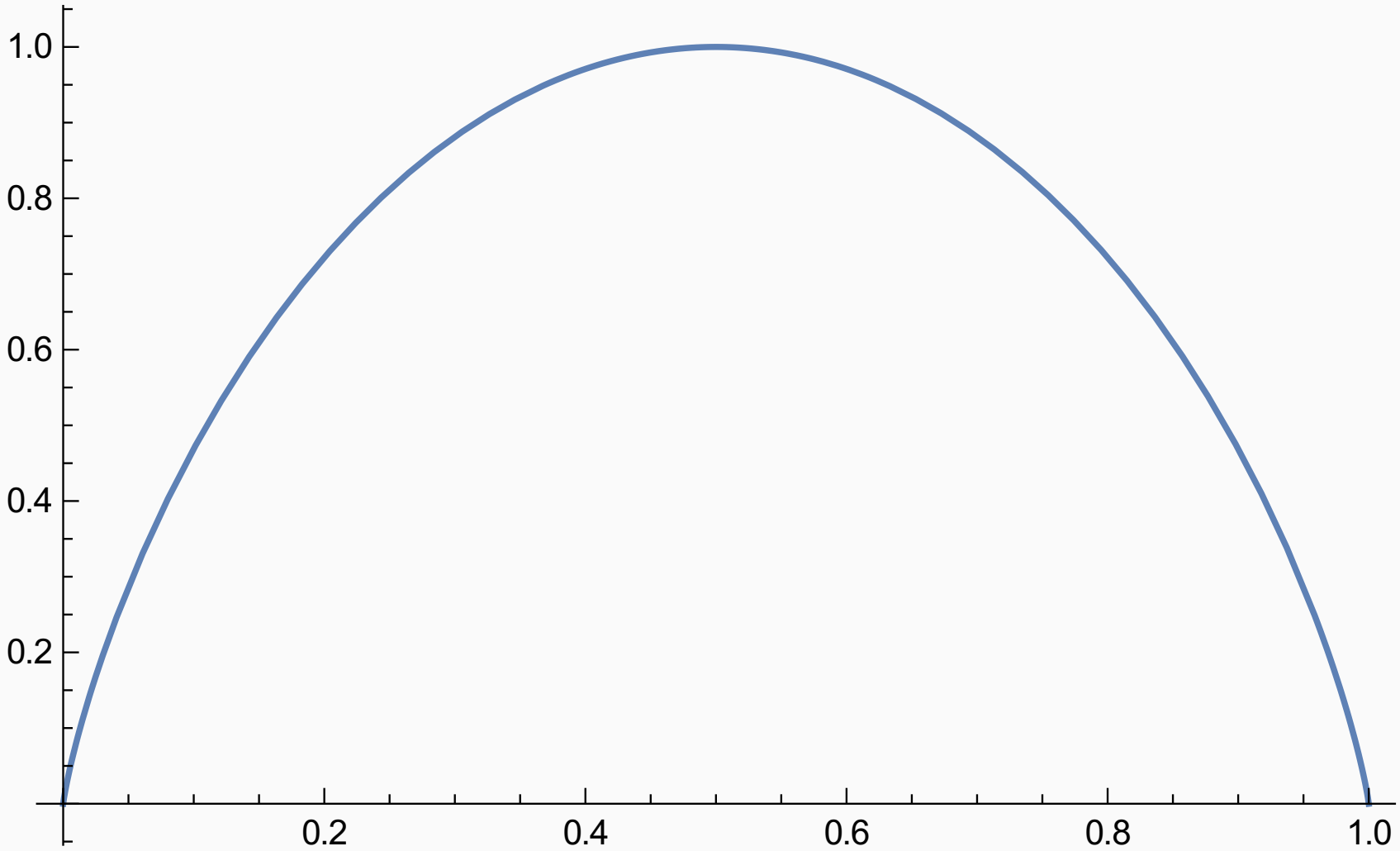
$$\sum_{\omega \in \Omega} \pi(\omega) = 1. \tag{2}$$

We write $\Delta(\Omega)$ for the set of all probability measures on Ω. Then the Shannon entropy of π is define as:

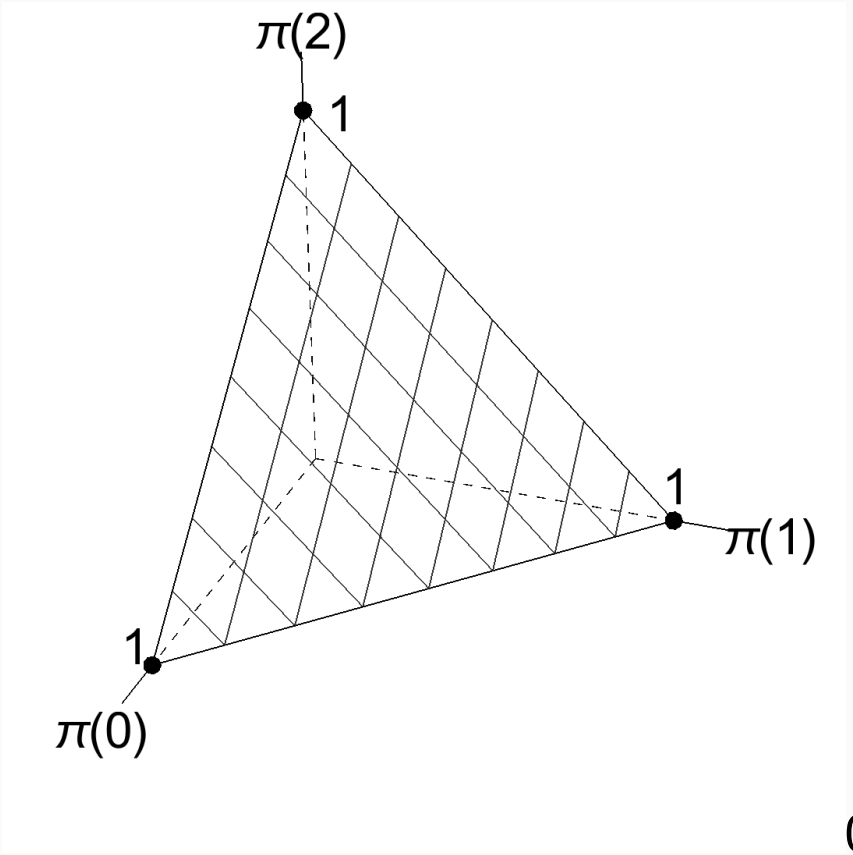
$$H(\pi) = \sum_{\omega \in \Omega} \pi(\omega) \log \frac{1}{\pi(\omega)} \tag{3}$$

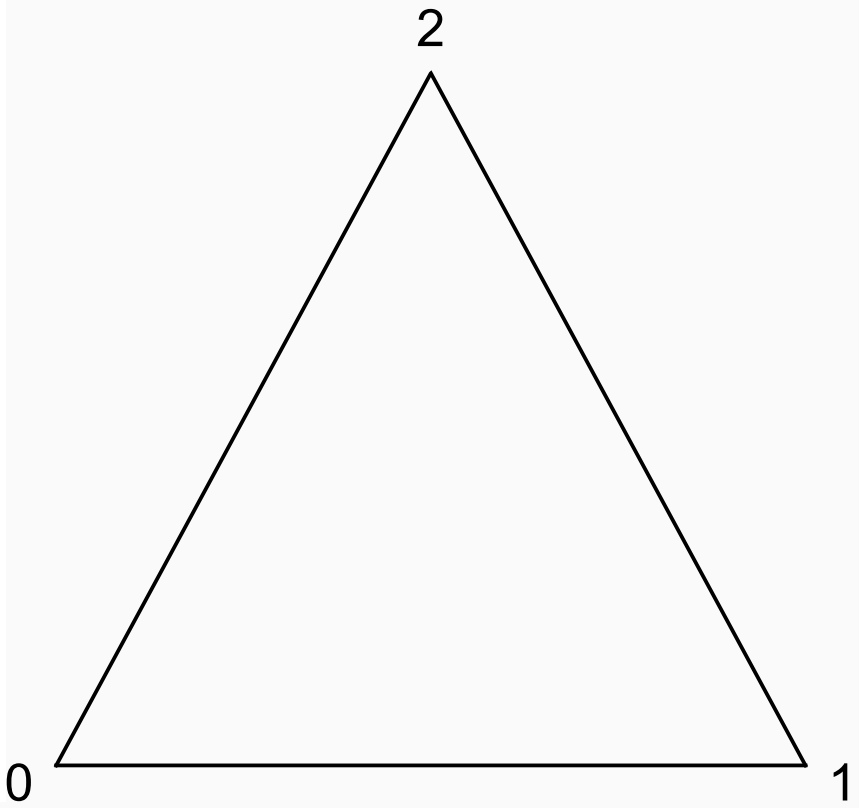
That is, a (weighted) average of the self-information of all points.

Entropy: 2D-plots



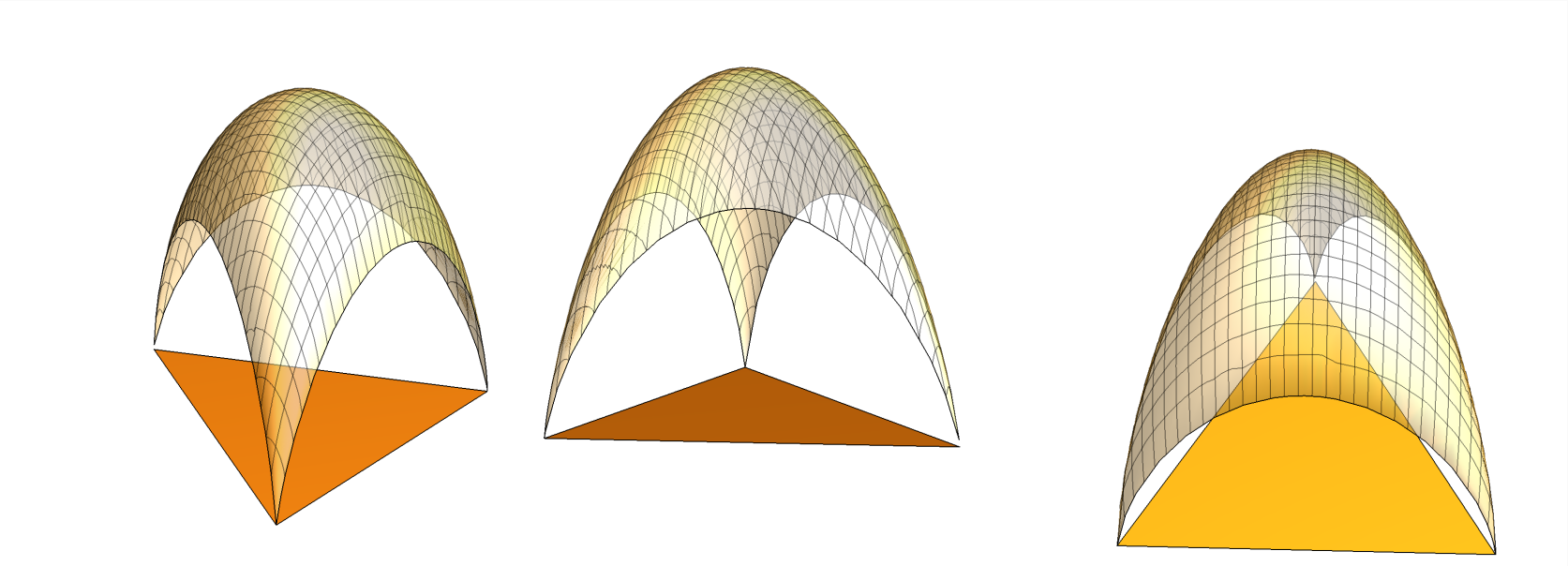
Entropy: 3D Probability Measures





Two views of the simplex $\Delta(\{0,1,2\})$. The simplex is part of the plane $x+y+z=1$.

Entropy: 3D-entropy



3D-entropy plotted on the previous equilateral triangle.

Kullback-Leibler Divergence

Definition
Let $\alpha, \beta \in \Delta(\Omega)$, where Ω is a nonempty finite set. The *Kullback-Leibler divergence* (or *KL-divergence*) *of* β *from* α is

$$D(\alpha||\beta) = \sum_{\omega \in \Omega} \alpha(\omega) \log \frac{\alpha(\omega)}{\beta(\omega)} = E_\alpha \log \frac{\alpha}{\beta}, \tag{4}$$

where the logarithm is base-2.

Kullback-Leibler Divergence

Definition
Let $\alpha, \beta \in \Delta(\Omega)$, where Ω is a nonempty finite set. The *Kullback-Leibler divergence* (or *KL-divergence*) *of* β *from* α is

$$D(\alpha||\beta) = \sum_{\omega \in \Omega} \alpha(\omega) \log \frac{\alpha(\omega)}{\beta(\omega)} = E_\alpha \log \frac{\alpha}{\beta}, \tag{4}$$

where the logarithm is base-2.

Note that we can view
$D(\alpha||\beta) = \sum_{\omega \in \Omega} \alpha(\omega) \log \frac{1}{\beta(\omega)} - \sum_{\omega \in \Omega} \alpha(\omega) \log \frac{1}{\alpha(\omega)}$

Kullback-Leibler Divergence

Definition
Let $\alpha, \beta \in \Delta(\Omega)$, where Ω is a nonempty finite set. The *Kullback-Leibler divergence* (or *KL-divergence*) *of* β *from* α is

$$D(\alpha||\beta) = \sum_{\omega \in \Omega} \alpha(\omega) \log \frac{\alpha(\omega)}{\beta(\omega)} = E_\alpha \log \frac{\alpha}{\beta}, \tag{4}$$

where the logarithm is base-2.

Note that we can view
$D(\alpha||\beta) = \sum_{\omega \in \Omega} \alpha(\omega) \log \frac{1}{\beta(\omega)} - \sum_{\omega \in \Omega} \alpha(\omega) \log \frac{1}{\alpha(\omega)}$

- $\beta(\omega)$ can be view as "subjective" probability, or a coding scheme.

Kullback-Leibler Divergence

Definition
Let $\alpha, \beta \in \Delta(\Omega)$, where Ω is a nonempty finite set. The *Kullback-Leibler divergence* (or *KL-divergence*) *of* β *from* α is

$$D(\alpha||\beta) = \sum_{\omega \in \Omega} \alpha(\omega) \log \frac{\alpha(\omega)}{\beta(\omega)} = E_\alpha \log \frac{\alpha}{\beta}, \tag{4}$$

where the logarithm is base-2.

Note that we can view
$D(\alpha||\beta) = \sum_{\omega \in \Omega} \alpha(\omega) \log \frac{1}{\beta(\omega)} - \sum_{\omega \in \Omega} \alpha(\omega) \log \frac{1}{\alpha(\omega)}$

- $\beta(\omega)$ can be view as "subjective" probability, or a coding scheme.
- $\alpha(\omega)$ can be view as the "real" probability.

Kullback-Leibler Divergence

Definition
Let $\alpha, \beta \in \Delta(\Omega)$, where Ω is a nonempty finite set. The *Kullback-Leibler divergence* (or *KL-divergence*) *of* β *from* α is

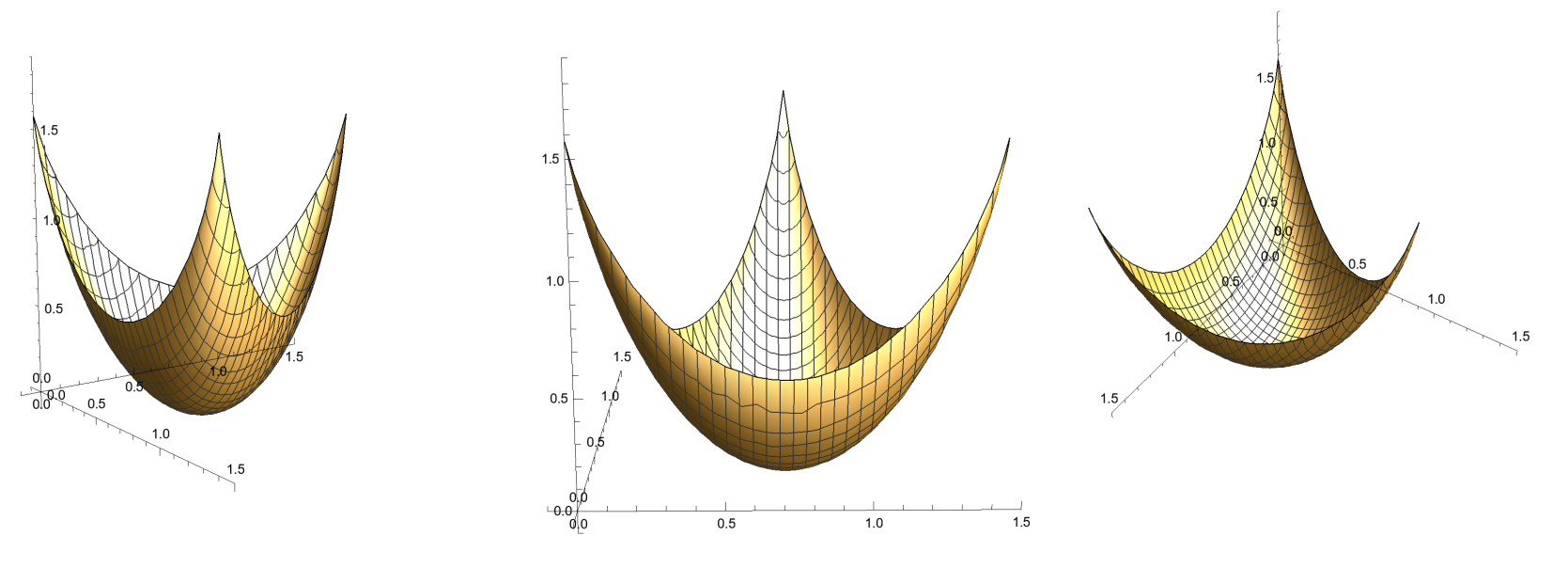
$$D(\alpha||\beta) = \sum_{\omega \in \Omega} \alpha(\omega) \log \frac{\alpha(\omega)}{\beta(\omega)} = E_\alpha \log \frac{\alpha}{\beta}, \tag{4}$$

where the logarithm is base-2.

Note that we can view
$D(\alpha||\beta) = \sum_{\omega \in \Omega} \alpha(\omega) \log \frac{1}{\beta(\omega)} - \sum_{\omega \in \Omega} \alpha(\omega) \log \frac{1}{\alpha(\omega)}$

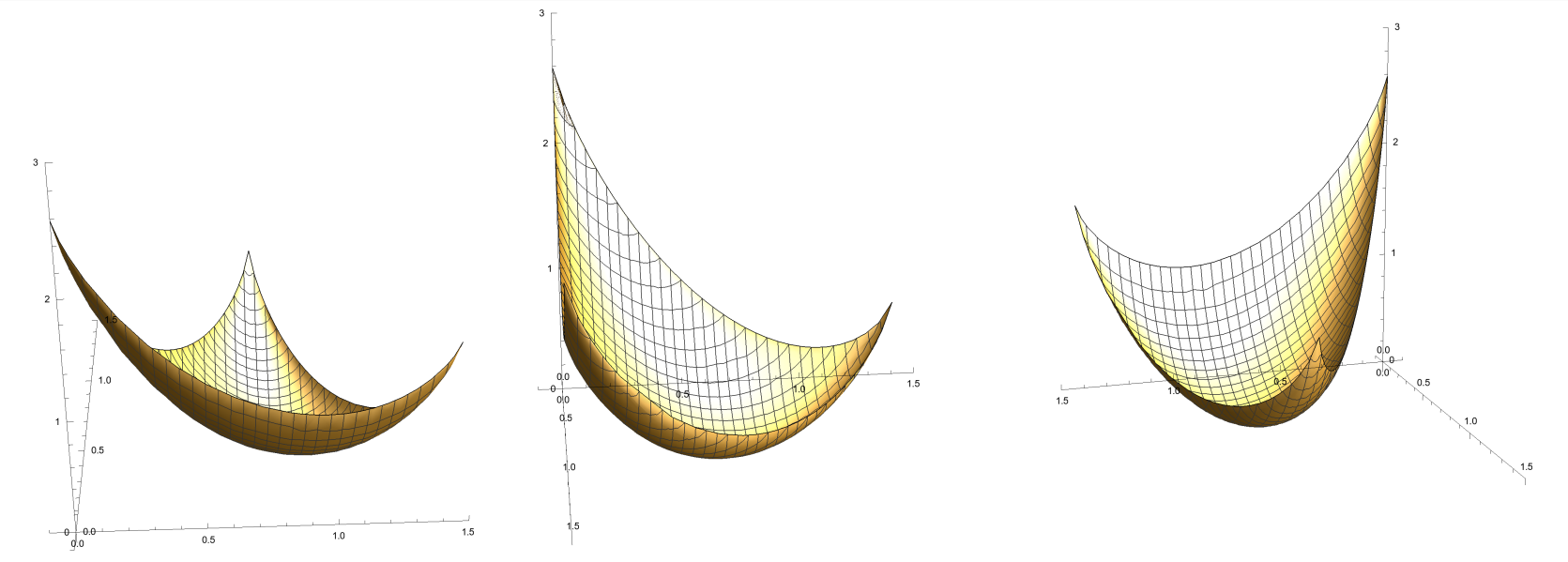
- $\beta(\omega)$ can be view as "subjective" probability, or a coding scheme.
- $\alpha(\omega)$ can be view as the "real" probability.
- $D(\alpha||\beta)$ measures the difference between the average code-length of a coding scheme and an optimal coding. So it is always ≥ 0.

Divergence: $D(\alpha||\beta)$, where $\beta = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$



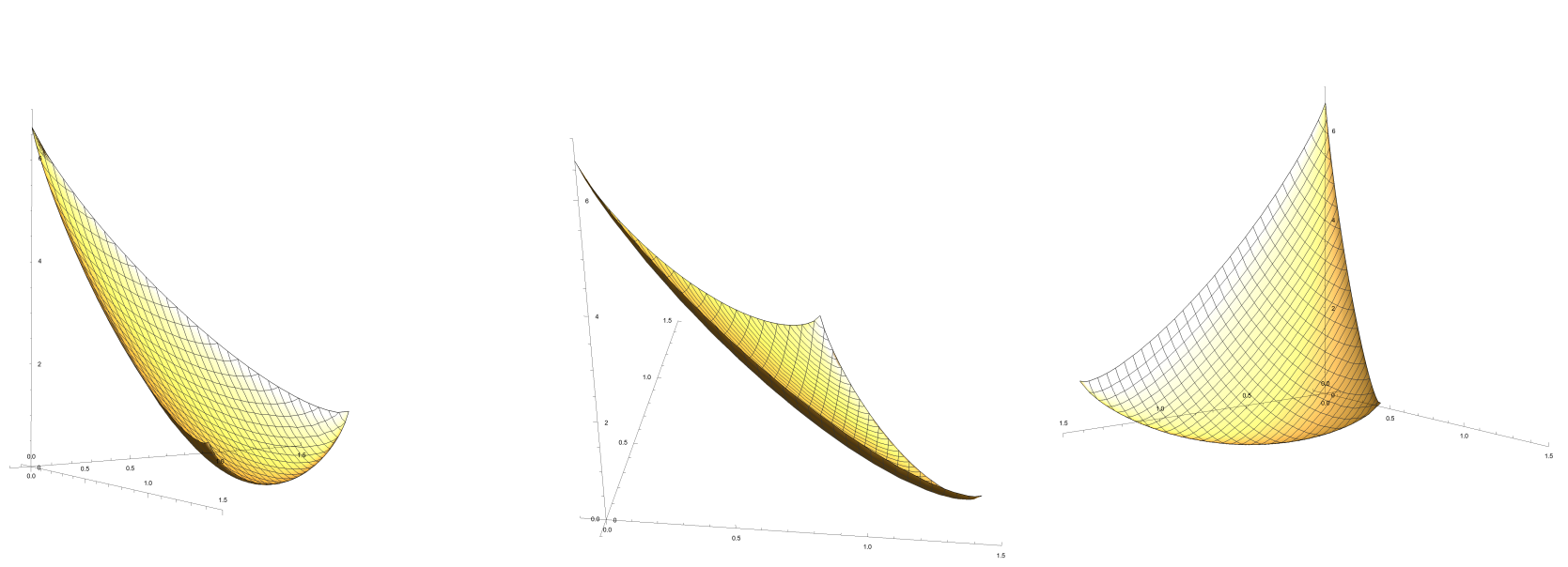
Three views of the function $D(\alpha||\beta)$, where $\beta = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. It is the upside-down copy of the 3D-entropy function.

Divergence: $D(\alpha||\beta)$, where $\beta = (\frac{1}{6}, \frac{1}{3}, \frac{1}{2})$



Three views of the function $D(\alpha||\beta)$, where $\beta = (\frac{1}{6}, \frac{1}{3}, \frac{1}{2})$.

Divergence: $D(\alpha||\beta)$, where $\beta = (\frac{1}{100}, \frac{1}{2}, \frac{49}{100})$



Three views of the function $D(\alpha||\beta)$, where $\beta = (\frac{1}{100}, \frac{1}{2}, \frac{49}{100})$.

Strong Dichotomy Theorem

Modulo asymptotic caveats, We show that[8]

1'. The infinitely-often exponential rate of winning in 1 is $2^{\mathrm{Div}(S||\alpha)|w|}$.

2'. The exponential rate of loss in 2 is $2^{-\mathrm{Risk}_G(w)}$.

[8] Xiang Huang, Jack H. Lutz, Elvira Mayordomo and Donald M. Stull (2019), "Asymptotic Divergences and Strong Dichotomy."

Strong Dichotomy Theorem

Modulo asymptotic caveats, We show that[8]

1'. The infinitely-often exponential rate of winning in 1 is $2^{\mathrm{Div}(S||\alpha)|w|}$.

2'. The exponential rate of loss in 2 is $2^{-\mathrm{Risk}_G(w)}$.

This is a **quantify** version of Schnorr-Stimm's dichotomy theorem.

Remark
$\mathrm{Div}(S||\alpha)$ *is an asymptotic version of divergence* D*, and* $\mathrm{Risk}_G(w)$ *is defined based on divergence* D*.*

[8] Xiang Huang, Jack H. Lutz, Elvira Mayordomo and Donald M. Stull (2019), "Asymptotic Divergences and Strong Dichotomy."

Algorithm Randomness

Algorithmic Randomness

We revisit the number

$$C_2 = 0.0\ 1\ 00\ 01\ 10\ 11\ 000\cdots$$

Algorithmic Randomness

We revisit the number

$$C_2 = 0.0\ 1\ 00\ 01\ 10\ 11\ 000\cdots$$

- It is "random" w.r.t. finite-state automata.

Algorithmic Randomness

We revisit the number

$$C_2 = 0.0\ 1\ 00\ 01\ 10\ 11\ 000\cdots$$

- It is "random" w.r.t. finite-state automata.
- It is not "random" w.r.t. some stack machine.

Algorithmic Randomness

We revisit the number

$$C_2 = 0.0\ 1\ 00\ 01\ 10\ 11\ 000\cdots$$

- It is "random" w.r.t. finite-state automata.
- It is not "random" w.r.t. some stack machine.
- Normality is a **resource bounded** randomness.

Algorithmic Randomness

We revisit the number

$$C_2 = 0.0\ 1\ 00\ 01\ 10\ 11\ 000\cdots$$

- It is "random" w.r.t. finite-state automata.
- It is not "random" w.r.t. some stack machine.
- Normality is a **resource bounded** randomness.
- Is there a notion of randomness that corresponds to the max computing power we have, i.e., Turing Machine?

Algorithmic Randomness

We revisit the number

$$C_2 = 0.0\ 1\ 00\ 01\ 10\ 11\ 000\cdots$$

- It is "random" w.r.t. finite-state automata.
- It is not "random" w.r.t. some stack machine.
- Normality is a **resource bounded** randomness.
- Is there a notion of randomness that corresponds to the max computing power we have, i.e., Turing Machine?

Walk along this path (computable selection rule) we can get to the von Mises-Church definition of algorithmic randomness. It is one of many ways to define algorithmic randomness.

Weakly PSPACE-randomness

Theorem (Weak PSPACE-randomness Characterization[9])
A point x is weakly PSPACE-random if and only if for every polynomial space L_1-computable $f \in L_1([0,1]^n)$, and every polynomial space computable sequence of simple function $\{f_m\}_{m\in\mathbb{N}}$ approximating f,

$$\lim_{m\to\infty} f_m(x) = \lim_{Q\to x} \frac{\int_Q f \mathrm{d}\mu}{\mu(Q)}$$

where the limit is taken over all cubes Q containing x as the diameter of Q tends to 0.

Remark
This is an interaction between resource bounded (polynomial space) randomness and a fundamental result of real analysis, the Lebesgue differentiation theorem.

[9] Xiang Huang and Donald M. Stull (2016), "Polynomial Space Randomness in Analysis"

Analog Computing

General purpose analog computer

The General Purpose Analog Computer (GPAC) is a mathematical model of analog computers first introduced in 1941 by Claude Shannon.

For the purpose of this talk, we can view a GPAC as a polynomial initial value problem (PIVP) $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ with integral coefficients satisfying $y(0) = 0$ and the individual components of $\mathbf{y}$ obey the ODEs

$$\begin{aligned} y_1' &= p_1(y_1, \ldots, y_n), \\ y_2' &= p_2(y_1, \ldots, y_n), \\ &\vdots \\ y_n' &= p_n(y_1, \ldots, y_n). \end{aligned}$$

Real-time analog computability

We say that a real number α is *real-time analog computable*, and we write $\alpha \in \mathbb{R}_{RTA}$, if there exist a polynomial initial value problem (PIVP) $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ with integral coefficients satisfying $\mathbf{y}(0) = \mathbf{0}$ and

1. There is a constant $\beta > 0$ such that $|y_i(t)| \leq \beta$ for all $1 \leq i \leq n$ and $t \in [0, \infty)$, and
2. $|y_1(t) - \alpha| \leq 2^{-t}$ for all $t \in [0, \infty)$.

- "What resources:" zero initial values, integer coefficients, bounded values
- "How fast:" exponentially fast convergent rate.

Chemical reaction network: a restricted model

A **chemical reaction network (CRN)** is a GPAC with the ODE system being

$$\begin{aligned} y_1' &= p_1 - q_1 y_1, \\ y_2' &= p_2 - q_2 y_2, \\ &\vdots \\ y_n' &= p_n - q_n y_n \end{aligned}$$

where $p_i = p_i(y_1, \ldots, y_n)$, and $q_i = q_i(y_1, \ldots, y_n)$ are polynomial with **positive terms**.

That is, negative terms in y_i, must have occurrence of y_i.

Remark
We care about this model because it is an abstractions of chemical reactions. This model has become the language of choice for Molecular Programming.

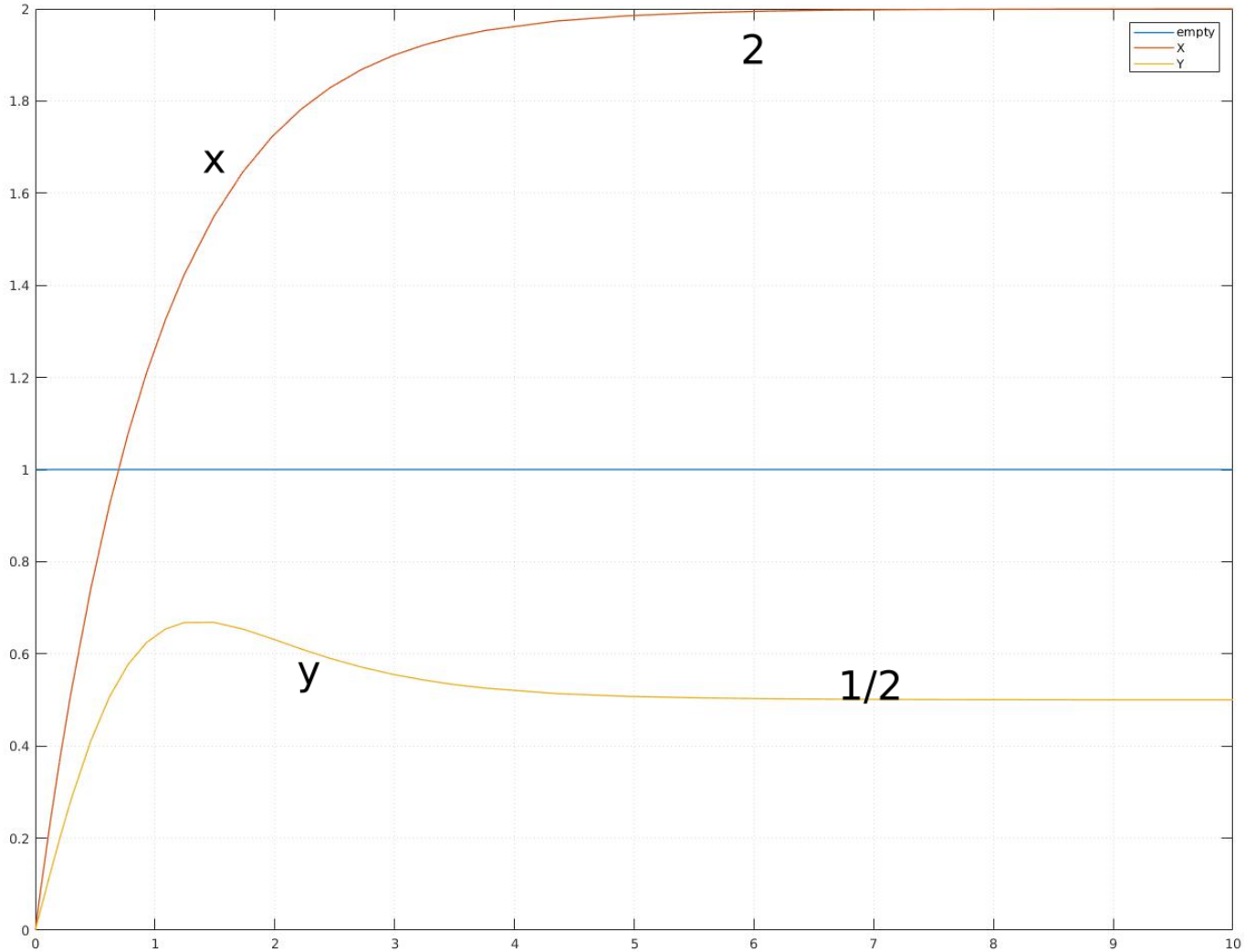
Real-time CRN computable numbers: an example

Example:

$$\begin{aligned} \emptyset &\xrightarrow{2} X \\ X &\xrightarrow{1} \emptyset \\ \emptyset &\xrightarrow{1} Y \\ X + Y &\xrightarrow{1} X \end{aligned} \qquad \Longleftrightarrow \qquad \begin{aligned} x' &= 2 - x \\ y' &= 1 - xy \end{aligned}$$

with initial value $x(0) = 0$, $y(0) = 0$.

Plot



RTCRN number: algebraic numbers

Theorem (Algebraics[10])
Algebraic numbers are real-time CRN-computable.

Remark
Algebraic numbers are roots of polynomials. For example, $\sqrt{2}$, $\frac{1+\sqrt{5}}{2}, \cdots$

Remark
We write the set of real-time CRN-computable number as $\mathbb{R}_{RTCRN}$.

[10] Xiang Huang, Titus H. Klinge, James I. Lathrop, Jack H. Lutz, and Xiaoyuan Li (2017), "Real-time computability of real numbers by chemical reaction networks."

Theorem (Transcendentals[11])
The number e, π are real-time CRN-computable.

Theorem (Euler constant)
Euler constant γ is real-time CRN-computable.

Remark
The number $\gamma = \lim_{n\to\infty} 1 + \frac{1}{2} + \cdots + \frac{1}{n} - \log n = 0.57721\cdots$

Theorem (Model Equivalence)
$\mathbb{R}_{RTA} = \mathbb{R}_{RTCRN}$

Remark
The weaker model (CRNs), turns out to have the same computing power as GPAC, in terms of computing real numbers in real time.

[11]Xiang Huang, Titus H. Klinge, and James I. Lathrop (2019), "Some Properties of Real-Time Computable Real Numbers by Chemical Reaction Networks", manuscript.

Open question: a possible tower?

$$\vdots$$

$$R_2 = CRN(R_1)$$

$$R_1 = CRN(R_0)$$

$$R_0 = CRN(Integers) = \mathbb{R}_{RTCRN}$$

Question: Do we really have a tower, or all layers collapse to $\mathbb{R}_{RTCRN}$?

Demo: compute π with CRN

Consider the PIVP defined by

$$w'(t) = -w(t), \tag{5}$$
$$x'(t) = -2w(t)x(t)y(t), \tag{6}$$
$$y'(t) = w(t)x(t)^2 - w(t)y(t)^2, \tag{7}$$
$$z'(t) = w(t)x(t), \tag{8}$$

with $w(0) = x(0) = 1$ and $y(0) = z(0) = 0$.

We know $z(t) = \arctan(1 - e^{-t})$ and it converges to $\frac{\pi}{4}$ in real-time.

Demo: compute π with CRN

Consider the PIVP defined by

$$w'(t) = -w(t), \tag{5}$$
$$x'(t) = -2w(t)x(t)y(t), \tag{6}$$
$$y'(t) = w(t)x(t)^2 - w(t)y(t)^2, \tag{7}$$
$$z'(t) = w(t)x(t), \tag{8}$$

with $w(0) = x(0) = 1$ and $y(0) = z(0) = 0$.

We know $z(t) = \arctan(1 - e^{-t})$ and it converges to $\frac{\pi}{4}$ in real-time.

Summer Project: The conversion of an PIVP to CRNs is pretty mechanical. Write a program to do that?

Demo: sample of reactions

$$U_1 + V_2 + V_3 \xrightarrow{2} U_1 + 2V_2 + V_3, \qquad V_2 + V_3 \xrightarrow{2} 2V_2 + V_3,$$
$$V_1 + V_3 \xrightarrow{2} V_1 + V_2 + V_3, \qquad V_1 + U_2 + V_3 \xrightarrow{2} V_1 + U_2 + V_2 + V_3,$$
$$U_1 + U_2 + U_3 \xrightarrow{2} U_1 + U_2 + V_2 + U_3, \qquad U_1 + U_3 \xrightarrow{2} U_1 + V_2 + U_3,$$
$$U_2 + U_3 \xrightarrow{2} U_2 + V_2 + U_3, \qquad U_3 \xrightarrow{2} V_2 + U_3,$$
$$V_1 + V_2 + U_3 \xrightarrow{2} V_1 + 2V_2 + U_3.$$
$$U_2 + V_2 + V_3 \xrightarrow{2} V_3, \qquad U_2 + V_2 + V_3 + U_1 \xrightarrow{2} V_3 + U_1,$$
$$2U_2 + V_2 + V_3 \xrightarrow{2} U_2 + V_3, \qquad U_1 + 2U_2 + V_2 + V_3 \xrightarrow{2} U_1 + U_2 + V_3,$$
$$V_1 + U_2 + 2V_2 + V_3 \xrightarrow{2} V_1 + V_2 + V_3, \qquad U_1 + U_2 + 2V_2 + U_3 \xrightarrow{2} U_1 + V_2 + U_3,$$
$$U_2 + 2V_2 + U_3 \xrightarrow{2} V_2 + U_3, \qquad V_1 + U_2 + V_2 + U_3 \xrightarrow{2} V_1 + U_3,$$
$$V_1 + 2U_2 + V_2 + U_3 \xrightarrow{2} V_1 + U_2 + U_3. \qquad \dots$$

Remark

There are 120+ reactions in the network.

Plot

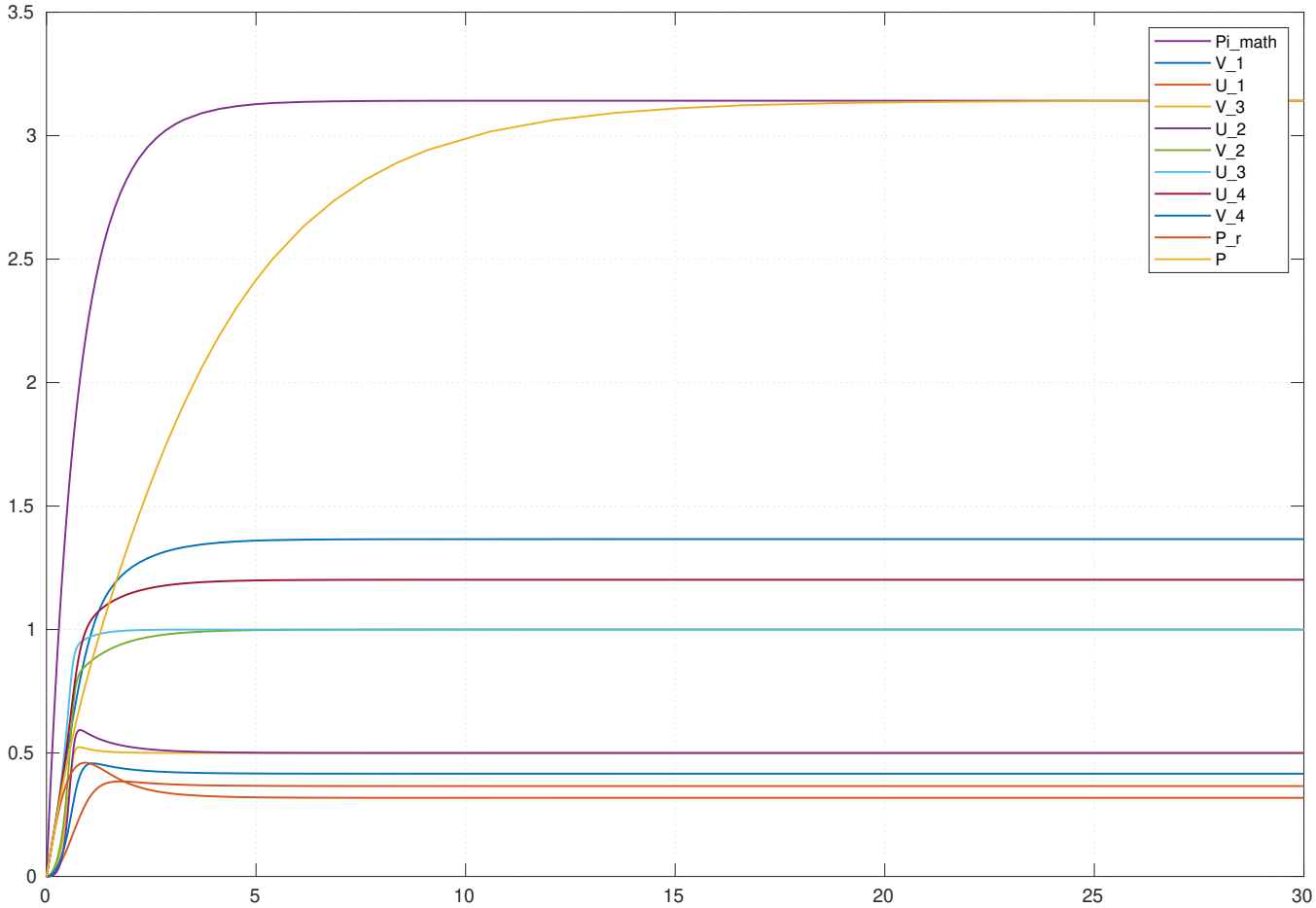


Figure 1: Compute π in real time by CRN

Thank you for your time!