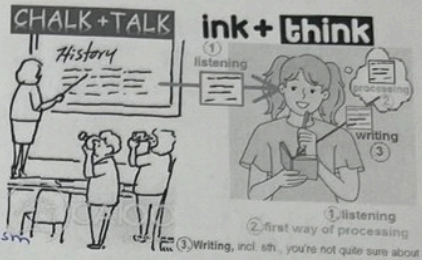
+0.1 11.2.2026



moving from school-level approach to the university-level formalism. And he said that understanding probability requires embracing formalism

School ⬇ gravity ⬇ MOTION ==formalism==> University $E=mc^2$ $\oint \vec{J} d\vec{S}$

CONCRETE AND ABSTRACT THINKING

based on intuition, observation

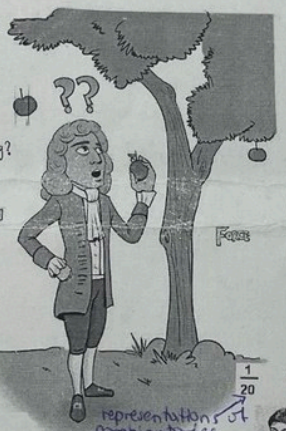
start with ⬇ gravity ⬇

no strict definitions → Why falling?

MOTION

ACCELERATION

University of Cambridge



ISAAC NEWTON

uses proofs, axioms, definitions

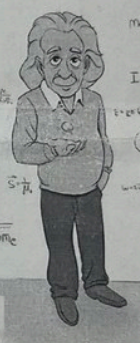
$E=mc^2$

$W=2\mu f$

$\beta=\frac{\Delta I_E}{\Delta I_B}$

$\lambda=\frac{h}{\sqrt{2em_e}}$

$\oint \vec{J} d\vec{S}$



ALBERT EINSTEIN

Princeton University

representation of combinatorics

Galois - abstract mathematical structure.

Hugo - human stories.

forecast becomes a math model.

We can calculate rainy days.

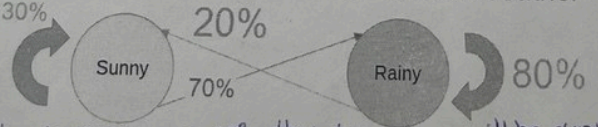
Motivation: 80% chance of rain

Let A_j be the event of rain on day j of this term, $1 \le j \le n$

Suppose the events A_j ... each have probability p, independently

Markoff Chain Probability Model for Oxford Weather

tomorrow's weather depends only on today's.



If today is sunny → 30% that tomorrow will be sunny and 70% - rainy. And for rainy day the same logic

$(a+b)^8 = a^8 + 8a^7b + 28a^6b^2 + 56a^5b^3 + 70a^4b^4 + 28a^3b^5 + \ldots$

56?

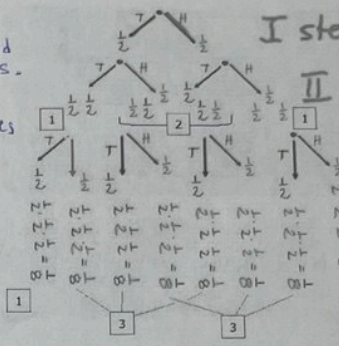
$e^{j\pi} = -1$

Pascal's triangle

to solve problems in probability and combinatorics. So just to calculate probabilities.

						1									1
					1		1								2
				1		2		1							4
			1		3		3		1						8
		1		4		6		4		1					16
	1		5		10		10		5		1				32
1		6		15		20		15		6		1			64
1	7		21		35		35		21		7		1		128

8 28 56 70 56 28 8



I step

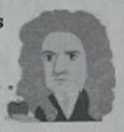
II steps

III steps

probability free

Compute the coefficients in the binomial expansion

Newton's Binomial



$(a+b)^0 = 1$

$(a+b)^1 = a+b$

$(a+b)^2 = a^2 + 2ab + b^2$

$(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$

$(a+b)^4 = a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4$

$(a+b)^5 = a^5 + 5a^4b + 10a^3b^2 + 10a^2b^3 + 5ab^4 + b^5$

$(a+b)^6 = a^6 + 6a^5b + 15a^4b^2 + 20a^3b^3 + 15a^2b^4 + 6ab^5 + b^6$

Events and probabilities

Consider an experiment wich has a set of Ω of outcomes

$\omega \in \Omega$ ← sample space

Ford example

a. toss in a coin $\Omega = \{H, T\}$ ← tails, head

B. throwing a dice

$\Omega \{(i,j)\ \ i, j \in [1,2,3,4,5,6]$

A subset of Ω is called an event

a) coin comes up tail $A = \{T\}$

b) We observe at total of 4

$A = \{(3,6), (4,3), (5,4), (6,3)\}$ - events

If $\omega \in \Omega$ is the outcome, we say A occur if $\omega \in A$

Compliment of A (A^c) occurs if A does not occur

$\bar{A}$"

Union: $A \cup B$ occurs if A or B occur

Intersection: $A \cap B$ occurs if both A and B occur

set difference: $A \setminus B = A \cap B^c$ ← B doesn't occurs

A occurs

Disjoint: A and B disjoint if $A \cap B = \emptyset$, A and B cannot occur together

• We assign a probab P(A) of each A simplest case:

Ω is finite and all outcomes are equally likely

then $P(A) = \frac{|A|}{|\Omega|}$

a) $|\Omega| = 2$, $|A| = 1$ → $P(A) = \frac{1}{2}$

b) $|\Omega| = 36$, $|A| = 4$ → $P(A) = \frac{4}{36} = \frac{1}{9}$

Elementary combinat

next in b/w paper

C_m^n binomial coefficient

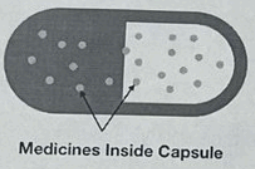
+0.1 16.9 +0.1

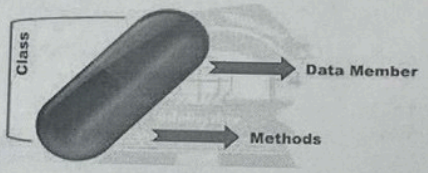
This is the program we need to write today

```
class ABBA
 {
 static void Main(string[] args)
                    // Here's a method called Main.
 {
     System.Console.WriteLine("ABBA!");
 }
}
```

So there's the keyword class. Unlike C++, in C# all code must be placed in a class.

Encapsulated in a class.

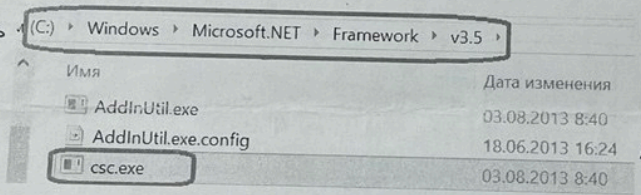




A ≠ a

C# is case sensitive

C:\WINDOWS\Microsoft.NET\Framework\v3.5\ csc.exe



ABBA.cs => csc.exe = ABBA.exe

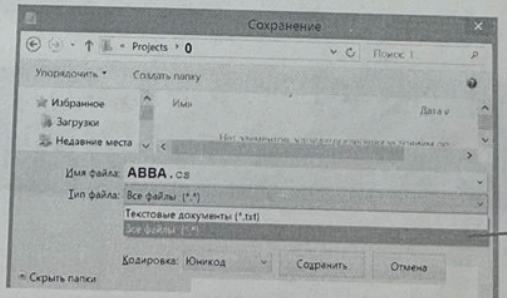
C# Compiler

Step 1. And on my HDD, I also make a folder with the same name D:\ IT

Step 2. In the folder **E:** IT\ we make the folder of the **Projects** - **E:\ IT\Projects**

And in the **Project** folder make folder **0 - E:\ IT\Projects\0** where our today's practical work will be stored

Step 3. As I mentioned above, C# is a built-in language of Windows. Notepad is enough to write a program



You need to switch from *.txt (Text documents) to *.* (all files)

Otherwise, notepad with *.txt extension

Step 4. Entering command mode

Start=>Run=>cmd

cd E: - After that go to the folder IT/Projects/0/

cd IT -Then go to the folder Projects

cd projects - Then go to the folder 0

cd 0 –

```
E:\>cd IT
E:\IT>cd Projects
E:\IT\Projects>cd 0
E:\IT\Projects\0>
```

View Model() {
View();
Model();
}

void Model() {
}

void View() {
}

Step 5.

Now we need to compile the file using the compiler csc.exe which is in the folder

C:\Windows\Microsoft.NET\Framework\v3.5

C:\WINDOWS\Microsoft.NET\Framework\v3.5\csc.exe ABBA.cs

the result is a file ABBA.exe that can already be run (which is located in the same folder). If you have taken the 1st step, then this means that the education-process has begun. This is victory. csc /target:library ABBA.cs - will make ABBA.dll.

```
>ABBA
ABBA
```

Step 6. Modify the file as follows

```
using System;

class ABBA
{
  static int Factorial(int n)
  {
    if (n == 1) return 1;
    return n * Factorial(n - 1);
  }

  static void Main(string[] args)
    // Here's a method called main.
  {
        System.Console.WriteLine("ABBA -"+
                                 Factorial(4));
  }
  }
```

```
>C:\WINDOWS\Microsoft.NET\Framework\v3.5\csc.exe ABBA.cs

>ABBA
ABBA – 24
```

Step 7. Modify the file as follows

```
using System;

class ABBA
{
  static int Factorial(int n)
  {
   if (n == 1) return 1;
    return n * Factorial(n - 1);
  }

  public static void Main()
  {
   System.Console.WriteLine("ABBA");
   System.Console.Write("Enter number of Vovels:");
   string s = Console.ReadLine();
   int vovels=int.Parse(s);
   System.Console.Write("Enter number of Consonants:");
   s=System.Console.ReadLine();
   int consonants=int.Parse(s);
   Console.WriteLine(Factorial(vovels)*Factorial(consonants) );
   }
}
```

```
>ABBA
ABBA
Enter number of Vovels:2
Enter number of Consonants:2
4
```

```
>ABBA
ABBA
Enter number of Vovels:3
Enter number of Consonants:3
36
```

1 2 3 4

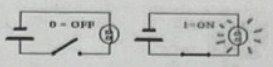
Information theory

entropy

Massachusetts Institute of Technology (MIT)

Lecture by Pr. Bob Gallagher

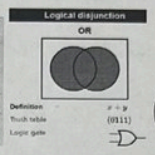
Boole (1815-1864) & **Shannon** (1916-2001)



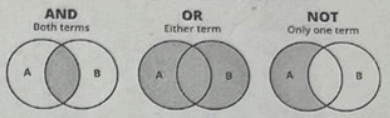
Logical addition (disjunction)

A	B	F=A∨B
0	0	0
0	1	1
1	0	1
1	1	1

A	B	$A \vee B$
True	True	True
True	False	True
False	True	True
False	False	False



BOOLEAN LOGIC



Good logic

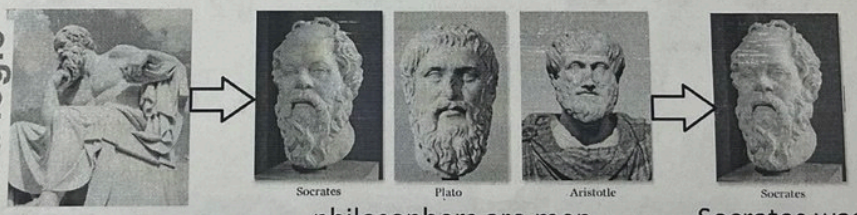
Socrates was a philosopher

philosophers are men

Socrates was a man

$S \in \Phi$ $\Phi \in A$ $S \in A$

Bad logic

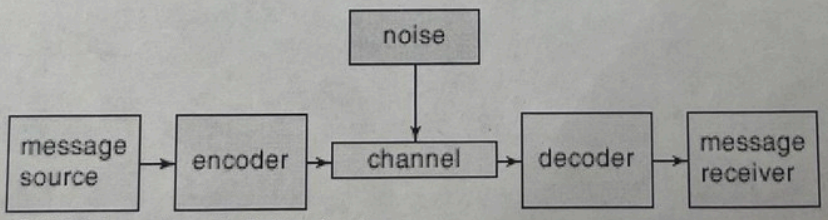


Socrates was a man

philosophers are men

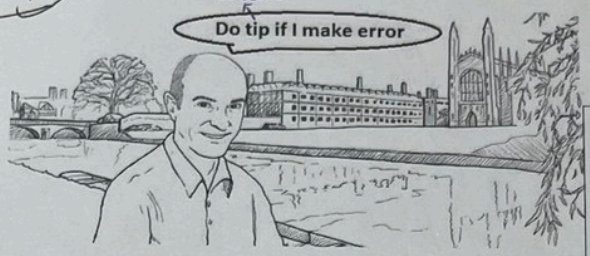
Socrates was a philosopher

$S \in A$ $\Phi \in A$ $S \in \Phi$

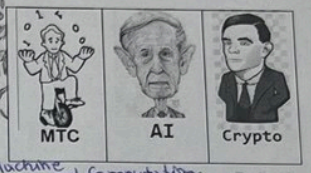


Shannon communication model

+0.1 Torn +0.1

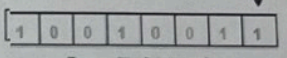


Sir **Dr. D. MacKay**,
University of Cambridge
(22 April 1967 – 14 April 2016)

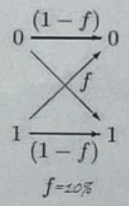


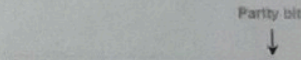
scientific reasoning → "I believe in clean energy, but I also believe in mathematics" ← emotional activism

Machine Theory and Computation



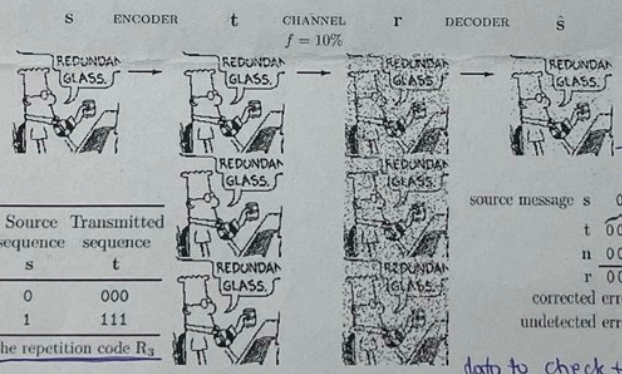
noisy channel can flip each bit with probability f.





+0.1 A parity bit is an extra bit added to data to make the number of 1 even or odd.

+0.1 It detects single-bit errors, but doesn't correct them.



The decoder uses majority voting it corrects single-bit errors.

Source sequence s	Transmitted sequence t
0	000
1	111

The repetition code R_3

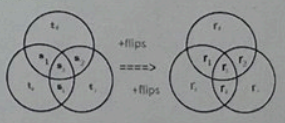
source message	s	0	0	1	0	1	1	0
	t	000	000	111	000	111	111	000
	n	000	001	000	000	101	000	000
	r	000	001	111	000	010	111	000
corrected errors			*			*		
undetected errors						*		

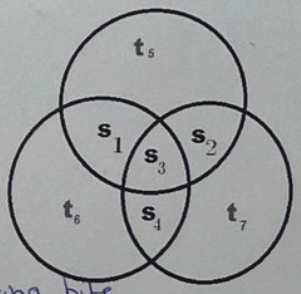
repeat every single bite 3 times for an accurate message

7.4. Hamming code.

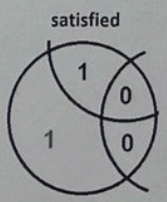
data to check + data bits

$$\frac{4}{\Sigma} \rightarrow \frac{7}{t}$$

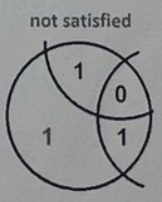




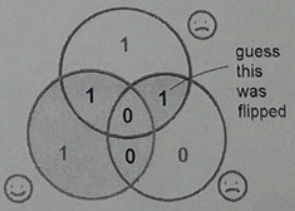
s_{1-4} - checking bits



1+1+0+0=**0**
even



1+1+0+1=1
odd



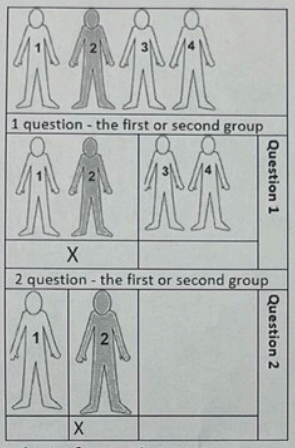
0,1 +0,1

Say **NO** to the **first**

Say **YES** to the **second** if it is better than the first

Say **NO** to the **third** only if it is worse than all the others

shannon entropy



Average number of questions =

1*0.5+	2*0.25+	3*0,125+	3*0,125
			

Question 1. Is this Zuckerberg?	 50%	1*0.5
Question 2. Is this Sergey Brin?	 25%	2*0.25
Question 3. Is this Stefan from BMW?	 12,5%	3*0,125
So Prince Saud	 12,5%	3*0,125
Average number of questions = 1,75		

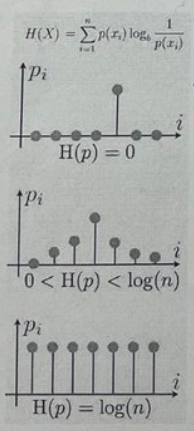
We use yes/no questions

$\log_2 4 = 2$

Average number of questions =

2*0.25+ 2*0.25+ 2*0,25+ 2*0,25 =2

Entropy is how many questions it takes to find the answer (or) how many different probabilities events have



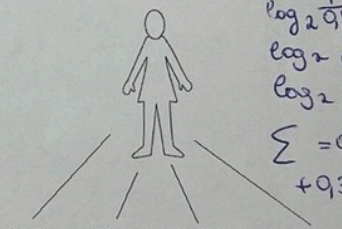
$$\sum_{i=1}^{n} p(i) \log_2 \frac{1}{p(i)}$$

Quantifying information

$$I(x_i) = \log_2\left(\frac{1}{p_i}\right)$$

number of bits required to encode choice

$$\sum_{i=1}^{n} p(x_i)\, I(x_i)$$



Mark Zukerberg P(1)= 50%

Sergey Brin P(2)= 25%

Stefan Quandt P(3)= 12,5%

Prince Al Saud P(4)= 12,5%

$\log_2 \frac{1}{0,5} = 1$

$\log_2 \frac{1}{0,3} = 1,74$

$\log_2 \frac{1}{0,1} = 3,32$

$\sum = 0,5 \cdot 1 +$

$+ 0,3 \cdot 1,74 +$

$+ 0,1 \cdot 3,32 +$

$+ 0,1 \cdot 3,32 =$

$= 0,5 +$

$+ 0,522 +$

$+ 0,332 \cdot 2 =$

$= 1,354$

1. 0.5 = 0.5

2. 0.3 0.6

3. 0.1 0.3

$= 0,5 + 0,6 + 0,3 + 0,3 = 1,7$

30% ★★★

10% ★

10% ✵

50% ✦✦✦✦✦

Shannon entropy

1.7

1.05

0.5 · 1 0.5

0.3 · 1.74 0.5.22

0.1 3.32 0.1.32

0.1 3.32 0.3.2

4.6

Resume of Lecture by Pr. Bob Gallagher from MIT

Massachusetts Institute of Technology (MIT)

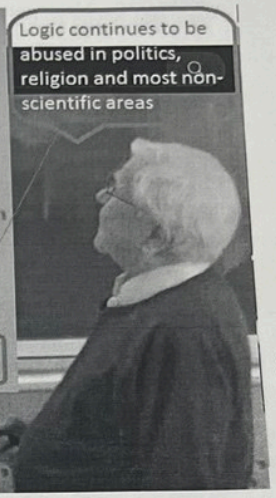
George Boole (1815-1864) developed Boolean logic

The principles of logical thinking have been understood (and occasionally used) since the Hellenic era.

Boole's contribution was to show how to systemize these principles and express them in equations (called Boolean logic or Boolean algebra).

Claude Shannon (1916-2001) showed how to use Boolean algebra as the basis for switching technology. This conribution systemized logical thinking for computer and communication systems, both for the design and programming of the systems and their applications.

Logic continues to be abused in politics, religion, and most non-scientific areas.



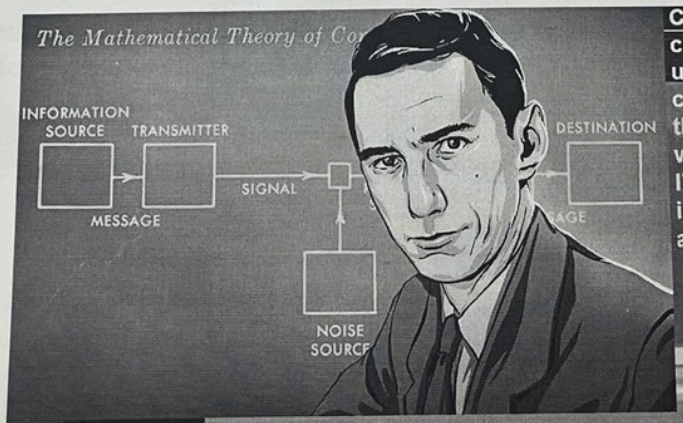
in there areas people rely on emotions instead of logical reasoning

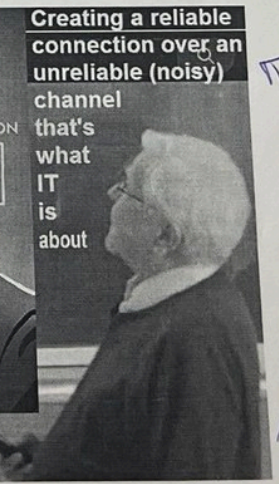
Kant Gauß Goethe

A little nationalistic, but this is an sample of right logic

Kant, Gauss, Goethe are great → Kant, Gauss, Goethe - Germans → Germany Great

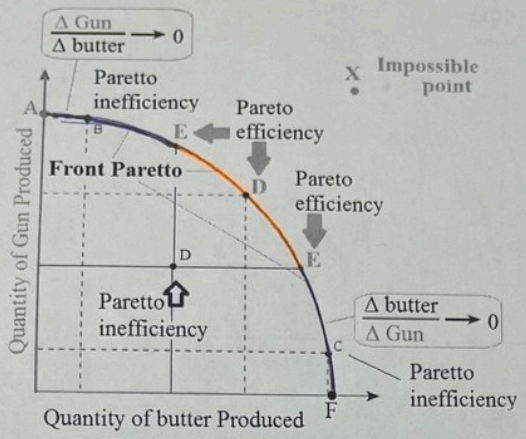
Bad logic (abuse of logic)





Shannon created the mathematical tools (entropy, coding, channel capacity) that make this possible

dilemma
state choice between military spending ('guns') and civilian needs ('butter')
This Production Possibility Curve shows the max combinations of guns and butter.

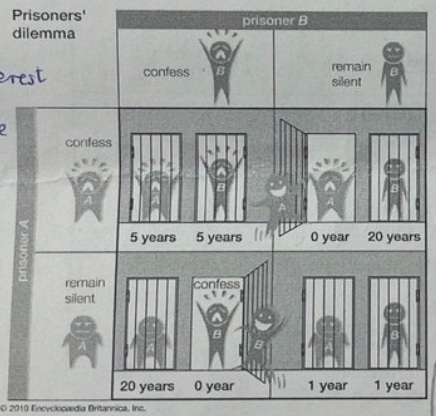


by Vilfredo Pareto
1848-1923

The orange sector E-D-E is the most Pareto efficient - since an increase in one indicator leads to a decrease in another.

Prisoners' dilemma

shows how individual interest leads to a worse collective outcome



© 2010 Encyclopaedia Britannica, Inc.

Game Theory

Nash Equilibrium

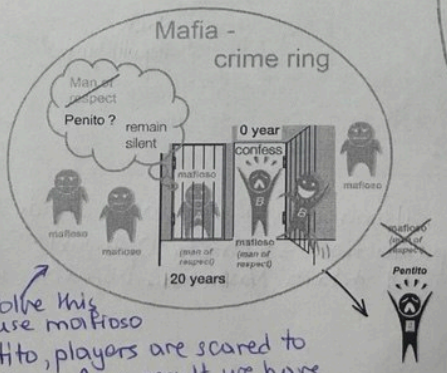
** => Nash equilibrium

$H_2(x)$ / $H_1(x)$	*Player 2*	Recognition;	Non-recognition;
Player 1		1	2
Recognition;	1	-5 / -5	-20 / 0
Non-recognition;	2	0 / -20	-1 / -1

-1-1

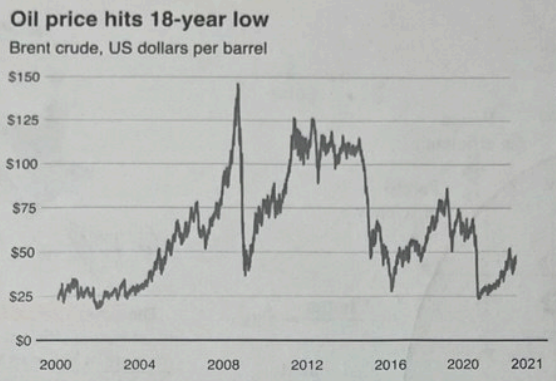
Pareto Optimality

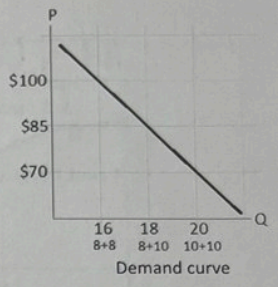
situation where it is impossible to make things better for one player without making them worse for the other

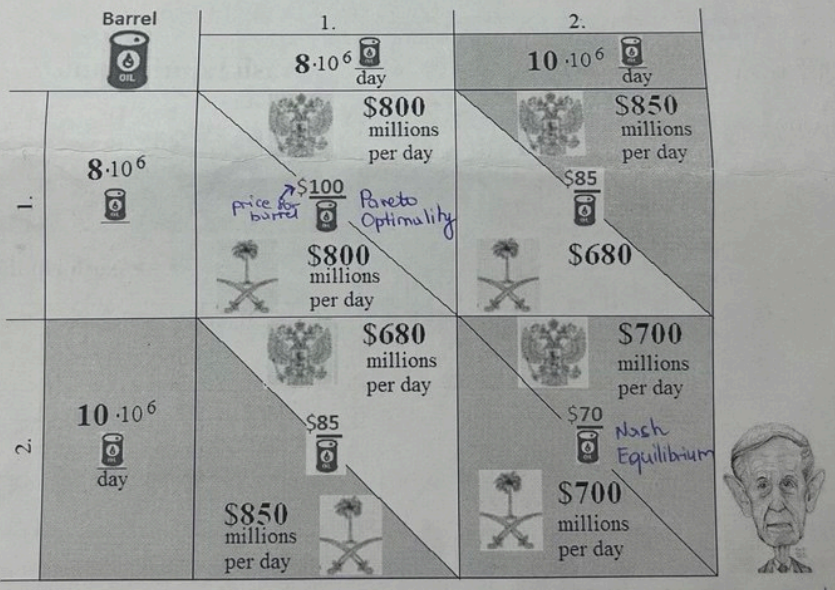


mafia code solve this problem. because mafioso punishes pentito, players are scared to betray each other. As a result we have the best collective outcome Pareto Optimality

situation where no player wants to change their strategy because doing it alone will only make their results worse.

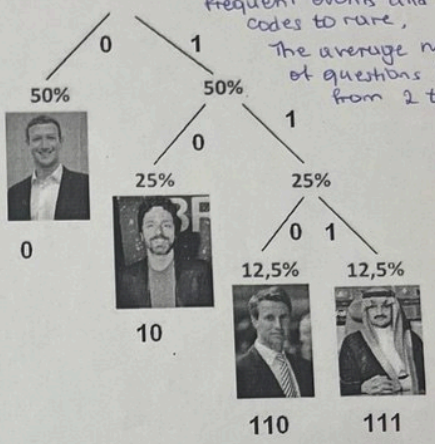






It shows the Prisoner's Dilemma to the oil market (Russia and Saudi Arabia). If both countries cooperate and limit production => Pareto Optimal. But due to self-interest and fear of cheating they fall into a Nash equilibrium where the price and revenue shrinks.

With unequal probabilities, optimal binary questions assign shorter codes to frequent events and longer codes to rare. The average number of questions drops from 2 to 1.75



letters are generated independently based only on their frequency. The child speech at this stage is a chaotic sequence of sounds.

First-order approximation (symbols independent but with frequencies of Belarusian txt).

Мама мыла ра

М - 3 — 30%		1-3 М
а - 4 — 40%		4-7 а
ы -1 — 10%		8 -ы
л - 1 — 10%		9 -л
р - 1 — 10%		10 -р
10		

лламаммар

Мама мыла ра

Ма - 2	22%	1-2 ма
ам - 2	22%	3-4 ам
мы - 1	11%	5 мы
ыл - 1	11%	6 ыл
ла - 1	11%	7 ла
ар - 1	11%	8 ар
ра - 1	11%	9 ра
9		

Second-order approximation (digram (2-symbols) structure as in Belarusian)

0.467319167350

ам ыл ла ам ма ра ма ыл ла ам мы

мылла рама

*Third-order approximation: the choice of each letter is dependent on the 2 previous letters

Each letter depends directly on the previous one. the model remembers pairs of letters. At this stage child's speech begins to form structures.