NONUNIVERSALITY IN COMPUTING

Selim G. Akl

School of Computing
Queen's University
Kingston, Ontario, Canada

The Universal Computer $\mathcal{U}$

A Universal Computer $\mathcal{U}$ is defined as a computing system with the following capabilities:

1. Input/Output

   - A means of communicating with the outside world with the purpose of receiving input and producing output,

     at any time during a computation.

2. Elementary Operations

   - The ability to perform all elementary arithmetic and logical operations

     (such as addition, subtraction, logical AND, and so on).

3. Program

   - A program made up of basic input, output, arithmetic, and logical operations.

4. Memory

   - An unlimited memory in which the program, the input, intermediate results, and the output are stored and can be retrieved.

Furthermore, $\mathcal{U}$ obeys the

Finiteness condition:

In one step, requiring one unit of time, $\mathcal{U}$ can execute a finite and fixed number of basic operations.

Specifically, it can:

1. Read a finite and fixed number of finite and fixed-sized inputs;

2. Perform a finite and fixed number of elementary arithmetic and logical operations on a finite and fixed number of finite and fixed-sized data;

3. Return a finite and fixed number of finite and fixed-sized outputs.

We consider this to be a 'reasonable' model of computation.

What makes computer $\mathcal{U}$ 'universal' is its supposed ability to simulate any computation performed on any other model of computation:

Anything that can be computed on some model, can be computed on $\mathcal{U}$.

There is no bound on the number of steps that $\mathcal{U}$ can perform to solve a problem:

a simulation can run for as long as required.

- Is simulation always possible?

- Is universality true?

NO FINITE COMPUTER IS UNIVERSAL

Evolving

Computational Systems

- Our universe is constantly undergoing change
  - Every moment there is a transformation that modifies the state of the world

- In computing, the world is assumed to be static
  - Read data
  - Apply algorithm
  - Output results.

What if

- the data
- or the algorithm
- or the results sought

happen to vary *during* the computation?

In an

Evolving Computational System

everything in the computational process is subject to change:

- the inputs
- the algorithms
- the outputs
- even the computing agents.

1. Study the effect of changes that take place during a computation, affecting

   - the data required to solve a problem

   - the complexity of the algorithm used in the solution

2. Example of a computer capable of evolving with a changing computation.

- Computational Models
- Unconventional Computations

  1. Time-Varying Variables
  2. Time-Varying Complexity
  3. Rank-Varying Complexity
  4. Interacting Variables
  5. Mathematical Constraints

Computational Models

Time unit:

Length of time required to perform a step consisting of three elementary operations on a constant number of fixed-size data:

1. Read
2. Calculate
3. Write

Computational Models

What does it mean to compute?

Any form of information processing
(whether occurring spontaneously in nature,
or performed on a computer built by humans)
is a computation.

Thus,

1. Measuring a physical quantity
2. Adding a pair of numbers
3. Setting the value of a physical quantity

are all instances of computational processes.

"The history of the universe is, in effect, a huge and ongoing quantum computation.
The universe is a quantum computer."

S. Lloyd, *Programming the Universe*, 2006.

"Think of all our knowledge-generating processes, our whole culture and civilization,
and all the thought processes in the minds of every individual,
and indeed the entire evolving biosphere as well, as being a gigantic computation.
The whole thing is executing a self-motivated, self-generating computer program."

D. Deutsch, *The Fabric of Reality*, 1997.

"Life is a form of information processing."

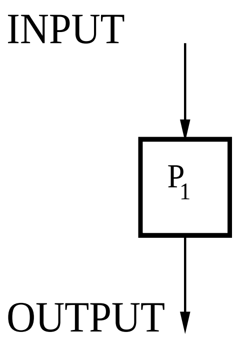
F. J. Tipler, *The Physics of Immortality*, 1995.

Computational Models

Generic Conventional Model

Sequential Computer used in the design and analysis of sequential algorithms

1. Single processor for arithmetic and logic
2. Memory for programs and data
3. Input and output units



In one time unit, the processor:

1. Receives a constant number of data as input
2. Executes a constant number of calculations
3. Returns a constant number of results as output.

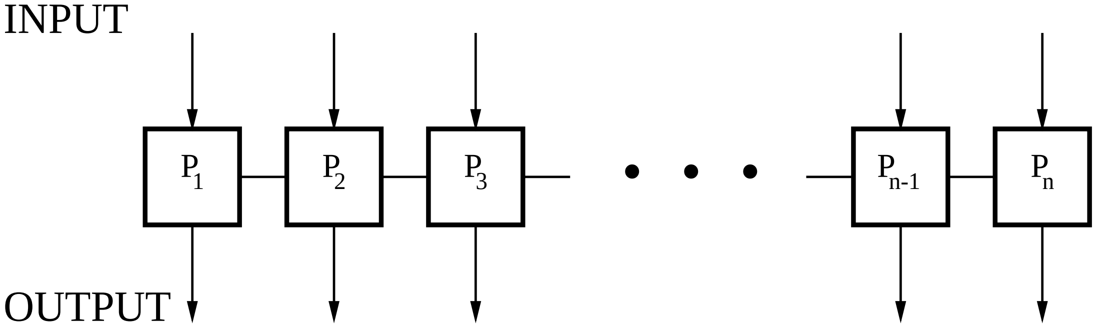
Computational Models

Generic Unconventional Model

Parallel Computer

1. n processors (each a sequential computer)

2. Connected in some fashion for communication



In one time unit, a processor:

1. Receives a constant number of data as input

2. Executes a constant number of calculations

3. Returns a constant number of results as output.

UNCONVENTIONAL COMPUTATIONS

Given n spatially and temporally connected variables

$$x_0, x_1, \ldots, x_{n-1}$$

Compute

$$\mathcal{F}(x_0, x_1, \ldots, x_{n-1})$$

NO FINITE MACHINE CAN COMPUTE $\mathcal{F}$

for all values of n

even if allowed infinite time and memory

PARADIGM 1

1. Time-Varying Variables

$$F_0(x_0),\ F_1(x_1),\ \ldots,\ F_{n-1}(x_{n-1})$$

The x_i are themselves functions that vary with time:

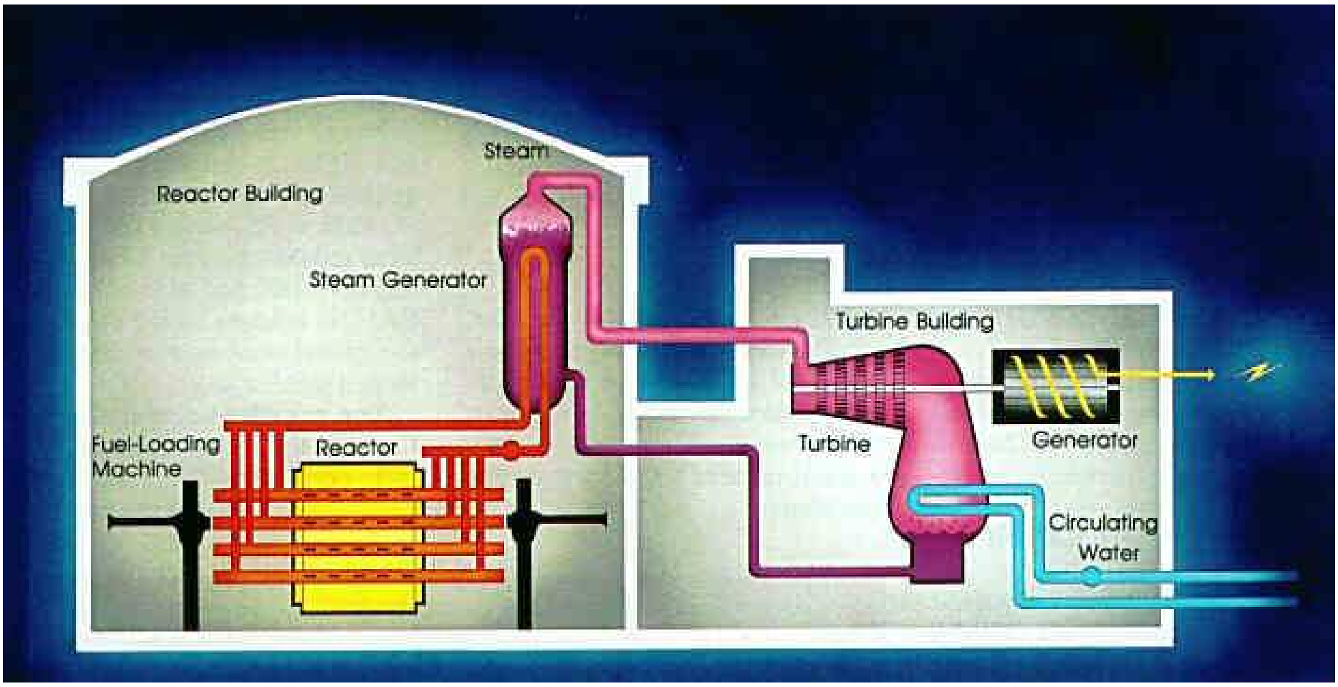
$$x_0(t),\ x_1(t),\ \ldots,\ x_{n-1}(t)$$

$$F_0(x_0(t_0)), F_1(x_1(t_0)), \ldots, F_{n-1}(x_{n-1}(t_0))$$

Each $x_i(t_0)$ is a physical variable, available in its natural environment, ready to be operated on.

$F_i(x_i(t_0))$ can be computed in one time unit if there is a computer to perform the calculation.

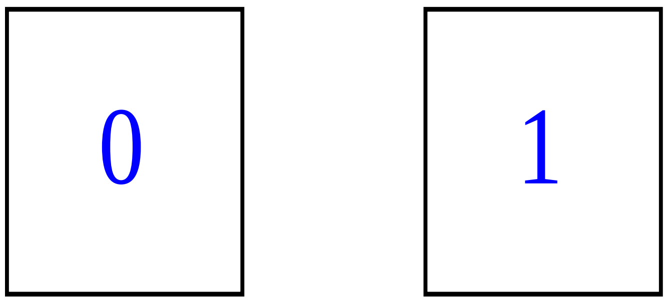
Example 1:
Sensor Network



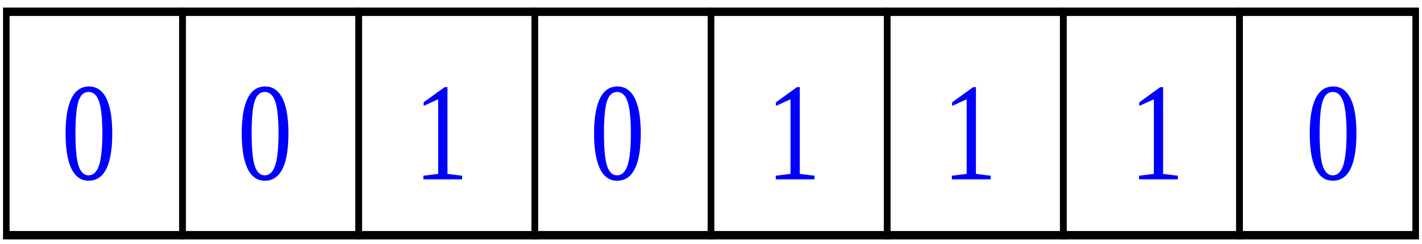
CANDU

$$F_0(x_0(t_0)), F_1(x_1(t_0)), \ldots, F_{n-1}(x_{n-1}(t_0))$$

Example 2:
Quantum Decoherence



Classical Bit



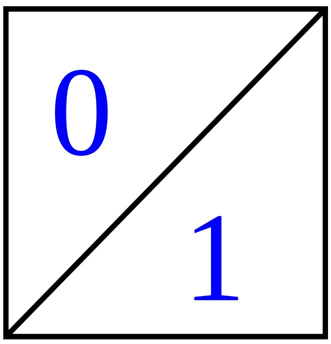
Classical Register

In quantum computing,

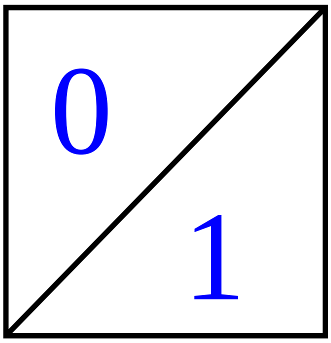
a qubit is a physical entity

that is in a superposition

of the two values 0 and 1.



Quantum Bit



Quantum Bit

Measurement yields

- 0 with probability p

- 1 with probability $1 - p$.

Measurement destroys

the superposition.

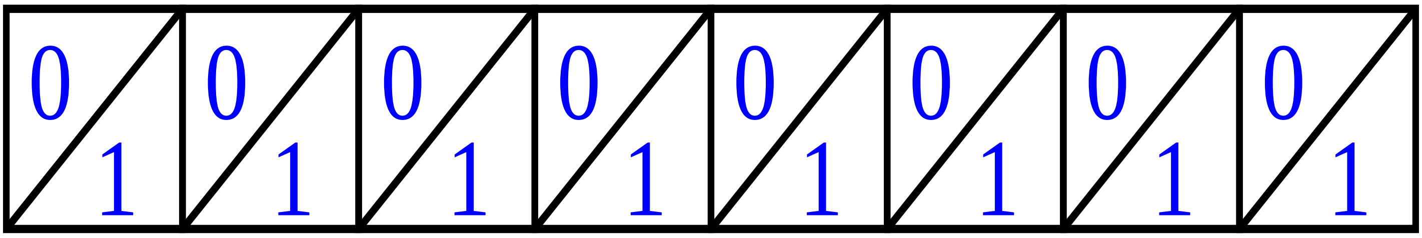
Quantum Decoherence

Through

1. Either measurement

2. Or prolonged exposure to its environment

the qubit loses its coherence: it no longer possesses its quantum properties and behaves as a classical bit.



Quantum Register

Suppose that a quantum system consists of

n independent qubits, x_0, x_1, ..., x_{n-1},

each in a state of superposition.

Their respective values at time t_0 are to be used to compute

$$F_0(x_0(t_0)), F_1(x_1(t_0)), \ldots, F_{n-1}(x_{n-1}(t_0))$$

IMPORTANT:

$x_i(t)$ changes with the passage of time

If $x_i(t)$ is not operated on at time $t = t_0$, then after one time unit

$$x_i(t_0)$$

becomes

$$x_i(t_0 + 1)$$

and after two time units it is

$$x_i(t_0 + 2)$$

and so on.

Furthermore

$$x_i(t_0)$$

cannot be recovered from

$$x_i(t_0 + k)$$

Conventional Solution

Suppose that $x_0(t_0)$ is read initially.

- $F_0(x_0(t_0))$

  can be computed correctly

- But when $x_1(t_0)$ is to be read,

  $x_1(t_0+1)$ is obtained, not $x_1(t_0)$.

- Then

$$x_2(t_0+2), \text{ not } x_2(t_0)$$

$$x_3(t_0+3), \text{ not } x_3(t_0)$$

$$\vdots$$

$$x_{n-1}(t_0+n-1), \text{ not } x_{n-1}(t_0)$$

  are read from the input.

Since the function according to which each x_i changes with time is not known,

- it is impossible to recover

$$x_i(t_0) \text{ from } x_i(t_0 + i),$$

$$\text{for } i = 1,\ 2,\ \ldots,\ n-1.$$

This approach cannot produce

$$F_1(x_1(t_0)),$$

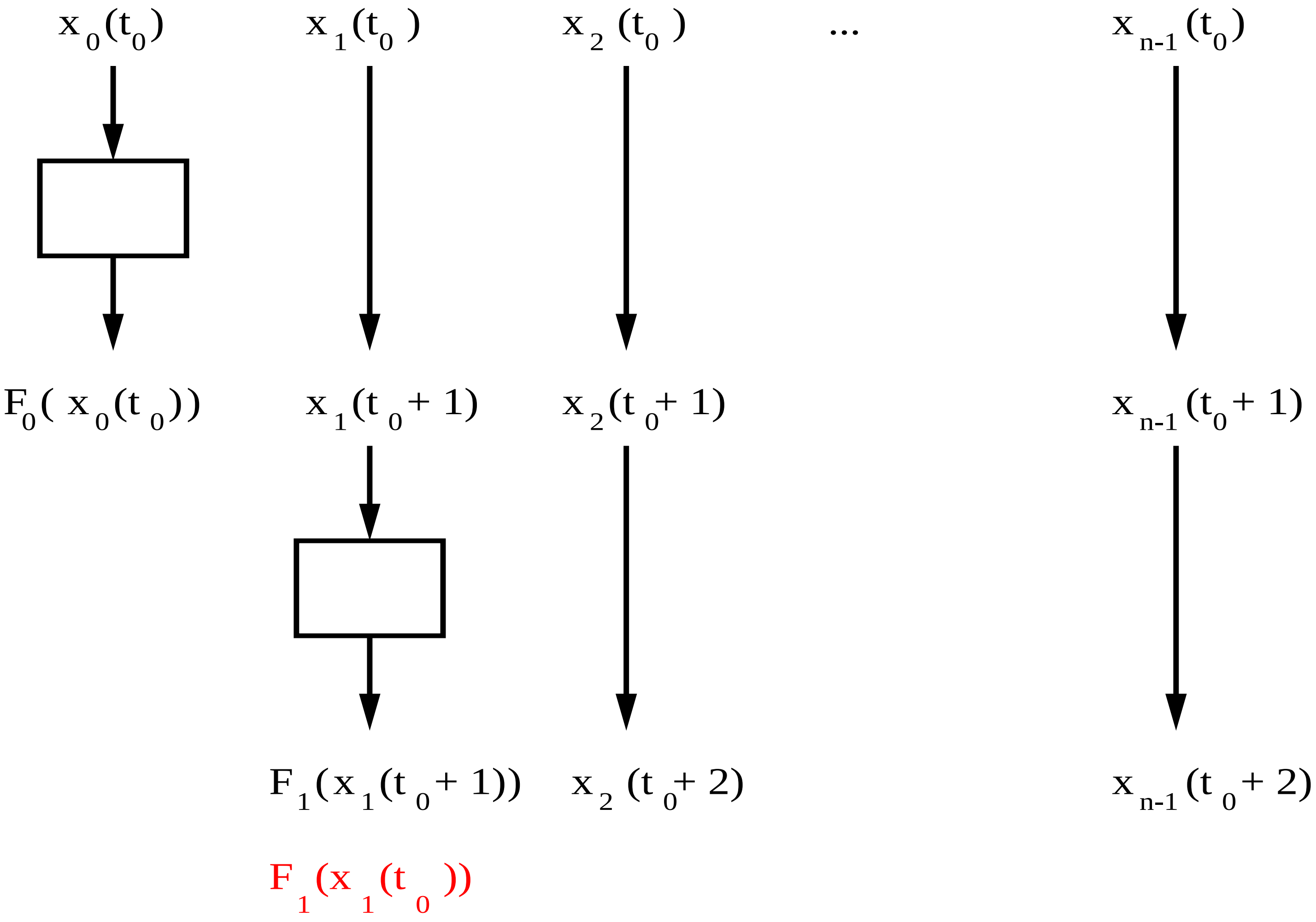
$$F_2(x_2(t_0)),$$

$$\vdots$$

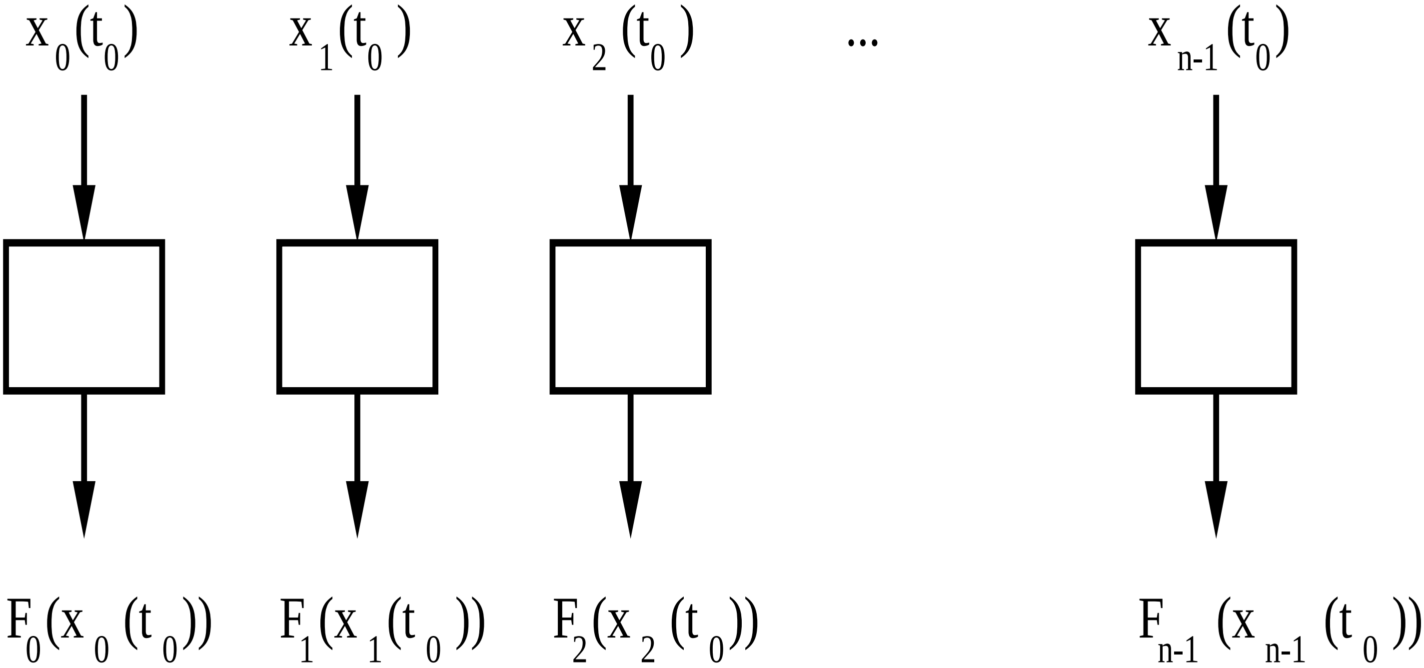
$$F_{n-1}(x_{n-1}(t_0)),$$

as required.

CONVENTIONAL SOLUTION



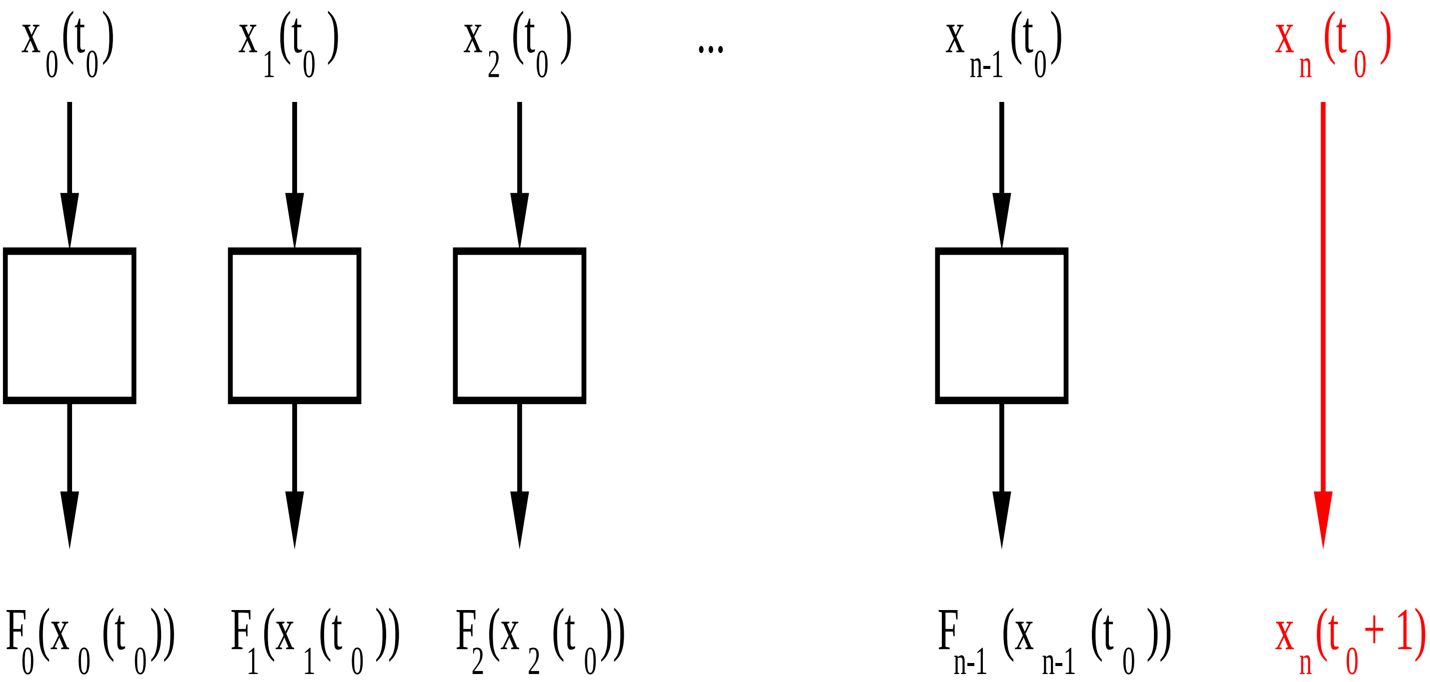
UNCONVENTIONAL SOLUTION



However,

for $n+1$ inputs

n processors do not suffice!



Simulating the parallel solution

on any computer capable of

fewer than n operations

per time unit is impossible,

regardless of how much time

is available to perform the simulation.

PARADIGM 2

ALEXANDRA KOSTENIUK

CHESS GRANDMASTER


Photo Pufichek (c) www.kosteniuk.com

Like ALEXANDRA
who faces a harder challenge
at every move

A COMPUTER
may find that life gets harder
at every step.

2. Time-Varying Computational Complexity

An algorithm consists of a number of stages:

A stage executed at time t requires $C(t)$ constant-time operations.

COMPUTING WITH DEADLINES

$$f_0(x_0),\ f_1(x_1),\ \ldots,\ f_{n-1}(x_{n-1})$$

1. All n functions are entirely independent.

2. Computing $f_i(x_i)$ at time t requires

   $C(t) = 3^t$ operations, $t \geq 0$.

3. All n values must be returned when $t = 3$.

CONVENTIONAL SOLUTION

Impossible for $n \geq 2$:

$f_0(x_0)$ takes $3^0 = 1$ time unit,

and $f_1(x_1)$ another $3^1 = 3$.

Now $t = 4$ and the deadline is missed!

UNCONVENTIONAL SOLUTION

In one time unit with n processors:

Processor i computes $f_i(x_i)$ at time $t = 0$.

Impossible with fewer than n processors!

Even $n - 1$ processors require 4 time units

and fail to meet the deadline.

COMPUTING WITHOUT DEADLINES

$$f_0(x_0),\ f_1(x_1),\ \ldots,\ f_{n-1}(x_{n-1})$$

1. All n functions are entirely independent.
2. Computing $f_i(x_i)$ at time t requires

$$C(t) = 3^t \text{ operations, } t \geq 0.$$

For $n \geq 4$,

any computer (conventional or unconventional)

capable of no more than $n/4$ operations per time unit

would require more than the age of the Universe!

(Assuming one time unit is one second).

PARADIGM 3

3. Rank-Varying

Computational Complexity

An algorithm consists
of a number of stages

Rank of a stage =
order of execution of that stage

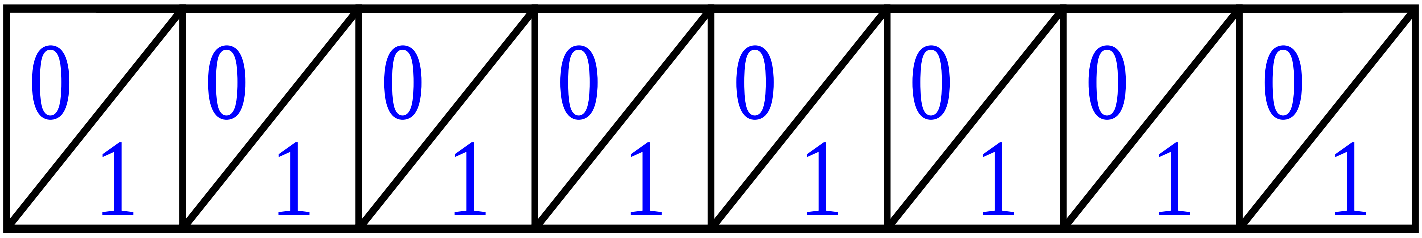
A stage whose rank is i
requires $C(i)$
constant-time operations.

COMPUTING THE INVERSE QUANTUM FOURIER TRANSFORM

A quantum register of n qubits

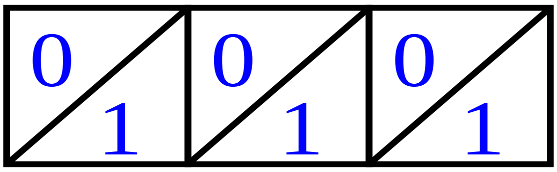
has 2^n computational basis vectors:

$$
\begin{array}{rcl}
|0\rangle & = & |000\cdots 00\rangle, \\
|1\rangle & = & |000\cdots 01\rangle, \\
& \vdots & \\
|2^n - 1\rangle & = & |111\cdots 11\rangle.
\end{array}
$$



Quantum Register

For 3 qubits the register has 2^3 basis vectors.



0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1

Let

$$|j\rangle = |j_1 j_2 j_3 \cdots j_{n-1} j_n\rangle,$$

be one of these vectors.

For $j = 0, 1, \ldots, 2^n - 1$,
the quantum Fourier transform of $|j\rangle$ is given by

$$\frac{(|0\rangle + e^{2\pi i 0.j_n}|1\rangle) \otimes (|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle) \otimes \cdots \otimes (|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle)}{2^{n/2}}$$

where

1. Each of $0.j_n$, $0.j_{n-1}j_n$, $\ldots$, $0.j_1 j_2 \cdots j_n$,
   is a rotation

2. The operator $\otimes$ is a tensor product

$$(a_1|0\rangle + b_1|1\rangle) \otimes (a_2|0\rangle + b_2|1\rangle) =$$

$$a_1 a_2|00\rangle + a_1 b_2|01\rangle + b_1 a_2|10\rangle + b_1 b_2|11\rangle.$$

INVERSE QFT:

Obtain the original vector

$$|j\rangle = |j_1 j_2 j_3 \cdots j_{n-1} j_n\rangle$$

from its given quantum Fourier transform

$$\frac{(|0\rangle + e^{2\pi i 0.j_n}|1\rangle) \otimes (|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle) \otimes \cdots \otimes (|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle)}{2^{n/2}}$$

CONVENTIONAL SOLUTION

1. Compute j_n from $|0\rangle + e^{2\pi i 0.j_n}|1\rangle$

2. Compute j_{n-1} from $|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle$

$\vdots$

k. Compute j_k from $|0\rangle + e^{2\pi i 0.j_k j_{k+1} \cdots j_n}|1\rangle$

$\vdots$

n. Compute j_1 from $|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle$

for $k = n$ downto 1 do

$$|j_k\rangle \leftarrow \frac{1}{\sqrt{2}}\left(\begin{array}{l} |0\rangle \\ e^{2\pi i 0.j_k j_{k+1}\cdots j_n}|1\rangle \end{array} \right)$$

for $m = k + 1$ to n do

if $j_{n+k+1-m} = 1$ then

$$|j_k\rangle \leftarrow |j_k\rangle \begin{pmatrix} 1 & 0 \\ 0 & e^{-2\pi i/2^{n-m+2}} \end{pmatrix}$$

end if

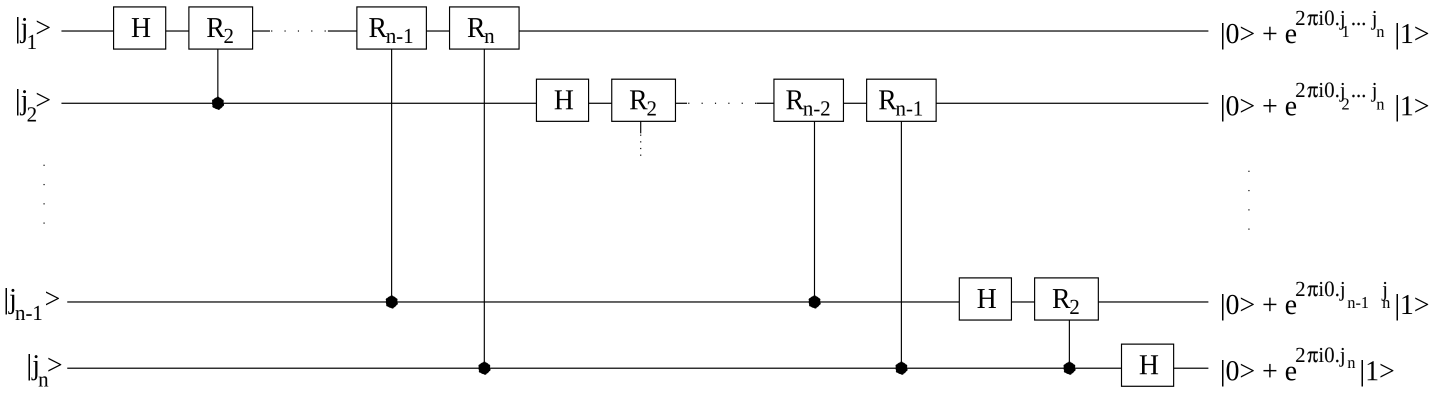
end for

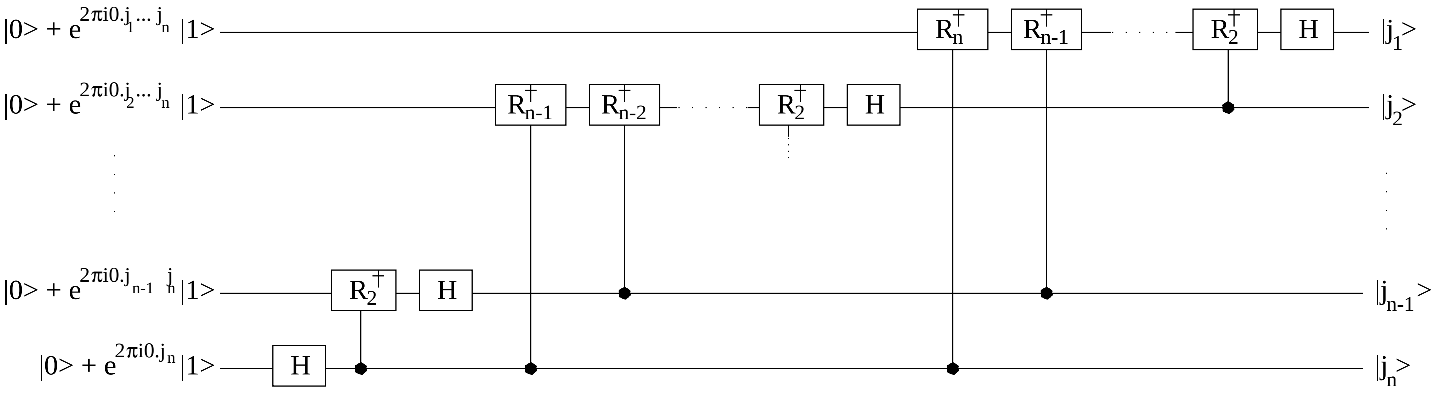
$$|j_k\rangle \leftarrow |j_k\rangle \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

end for. ∎

Number of time units required to compute $j_1, j_2, \ldots, j_n$

$$1 + 2 + \cdots + n = n(n+1)/2$$





For $k = n, n-1, \ldots, 2$, once j_k is known,
all operations involving j_k in the computation of

$$j_1, j_2, \ldots, j_{k-1},$$

can be performed simultaneously, each being a rotation.

for $k = 1$ to n do in parallel

$$|j_k\rangle \leftarrow \frac{1}{\sqrt{2}} \begin{pmatrix} |0\rangle \\ e^{2\pi i 0.j_k j_{k+1} \cdots j_n} |1\rangle \end{pmatrix}$$

end for

$$|j_n\rangle \leftarrow |j_n\rangle \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

for $k = n - 1$ downto 1 do

if $j_{k+1} = 1$ then

for $m = 1$ to k do in parallel

$$|j_m\rangle \leftarrow |j_m\rangle \begin{pmatrix} 1 & 0 \\ 0 & e^{-2\pi i / 2^{n-m+1}} \end{pmatrix}$$

end for

end if

$$|j_k\rangle \leftarrow |j_k\rangle \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

end for. ∎

Number of time units required to compute $j_1, j_2, \ldots, j_n$:

$$2n - 1$$

More importantly,

if decoherence takes place within δ time units,

$$2n - 1 < \delta < n(n+1)/2$$

unconventional computing succeeds,

conventional computing fails.

However,

if instead of n

only $n - 1$ operations

can be performed simultaneously

then decoherence is inevitable.

PARADIGM 4

4. Interacting Variables

- Computations whose characteristics are akin to certain unique phenomena that are observed in different domains of science.

- Systems whose computations are subject to natural law:

  - the system variables are altered unpredictably whenever one of these variables is measured or modified.

- Examples

  - Heisenberg's uncertainty principle of quantum physics:

    puts a limit on our ability to measure accurately pairs of 'complementary' variables.

  - Le Châtelier's principle of chemical systems under stress:

    if a system at equilibrium is subjected to a stress, the system will shift to a new equilibrium in an attempt to reduce the stress.

  - the homeostatic principle in biology:

    maintains the equilibrium necessary for the survival of organisms

$$x_0,\ x_1,\ \ldots,\ x_{n-1},$$

are the variables of a physical system.

They need to be measured in order to compute

$$F_0(x_0),\ F_1(x_1),\ \ldots,\ F_{n-1}(x_{n-1}),$$

The physical system has the property that

- measuring one variable disturbs any number of the remaining variables unpredictably
  - meaning that we cannot tell which variables have changed value, and by how much.

Mathematical model

$$
\begin{aligned}
x_0(t+1) &= g_0(x_0(t), x_1(t), \ldots, x_{n-1}(t)) \\
x_1(t+1) &= g_1(x_0(t), x_1(t), \ldots, x_{n-1}(t)) \\
&\vdots \\
x_{n-1}(t+1) &= g_{n-1}(x_0(t), x_1(t), \ldots, x_{n-1}(t))
\end{aligned}
$$

Note:

1. When the system is in a state of equilibrium, its variables do not change with the passage of time.

2. The functions $g_0, g_1, \ldots, g_{n-1}$ are too complex to capture mathematically.

Suppose we wish to measure

$$x_0(t_0), x_1(t_0), \ldots, x_{n-1}(t_0)$$

at moment t_0, when the system is in a stable state, in order to compute

$$F_i(x_i(t_0)), 0 \leq i \leq n-1.$$

It is easy to measure $x_0(t_0)$.

This measurement operation will change the state of the system from

$$(x_0(t_0), x_1(t_0), \ldots, x_{n-1}(t_0))$$

to

$$(x'_0(t_0), x_1(t_0), \ldots, x_{n-1}(t_0)),$$

where $x'_0(t_0)$ denotes the value of variable x_0 after measurement.

Therefore at time $t_0 + 1$,

$$\begin{aligned}
x_0(t_0+1) &= g_0(x'_0(t_0), x_1(t_0), \ldots, x_{n-1}(t_0)) \\
x_1(t_0+1) &= g_1(x'_0(t_0), x_1(t_0), \ldots, x_{n-1}(t_0)) \\
&\vdots \\
x_{n-1}(t_0+1) &= g_{n-1}(x'_0(t_0), x_1(t_0), \ldots, x_{n-1}(t_0))
\end{aligned}$$

CONVENTIONAL SOLUTION

Fails to compute the required F_i.

Suppose that x_0 is measured first.

- This allows a correct evaluation of

$$F_0(x_0), \text{ at time } t_0$$

- but affects any number of the remaining variables

$$x_1,\ x_2,\ \ldots,\ x_{n-1}$$

irremediably.

Since we cannot recover the original values of

$$x_1,\ x_2,\ \ldots,\ x_{n-1},$$

the computation of

$$\begin{gathered} F_1(x_1), \\ F_2(x_2), \\ \vdots \\ F_{n-1}(x_{n-1}), \end{gathered}$$

is impossible.

UNCONVENTIONAL SOLUTION

A parallel computer with n processors,

will measure all the variables

$$x_0,\ x_1,\ \ldots,\ x_{n-1}$$

simultaneously (one value per processor), and compute

$$F_0(x_0),\ F_1(x_1),\ \ldots,\ F_{n-1}(x_{n-1}),$$

at t_0 as required.

DISTINGUISHING AMONG 2^n QUANTUM STATES

A register of n qubits
is in one of the following 2^n states:

$$\frac{1}{\sqrt{2}}(|000\cdots 0\rangle \ \pm \ |111\cdots 1\rangle),$$

$$\frac{1}{\sqrt{2}}(|000\cdots 1\rangle \ \pm \ |111\cdots 0\rangle),$$

$$\vdots$$

$$\frac{1}{\sqrt{2}}(|011\cdots 1\rangle \ \pm \ |100\cdots 0\rangle).$$

It is impossible

through single measurement

to distinguish among these 2^n states.

If after one qubit is read the superposition collapses to

$$|000\cdots 0\rangle,$$

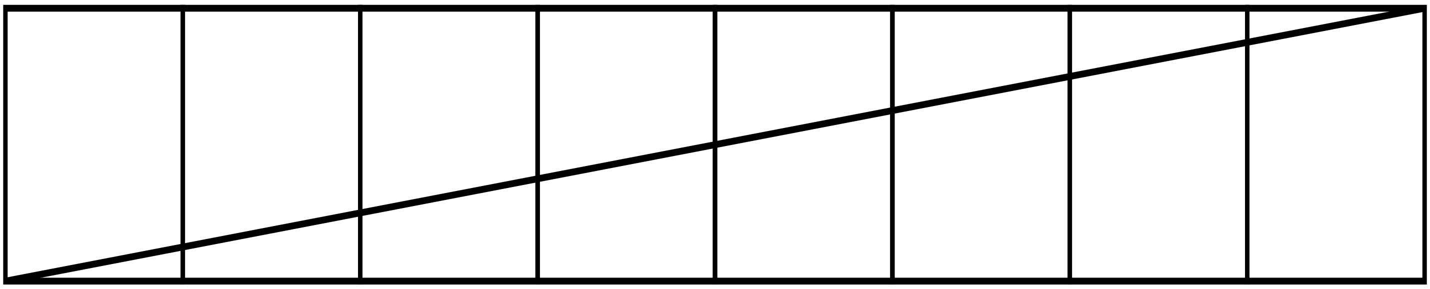
we will have no way of telling
which of the two superpositions,

$$\frac{1}{\sqrt{2}}(|000\cdots 0\rangle + |111\cdots 1\rangle),$$

or

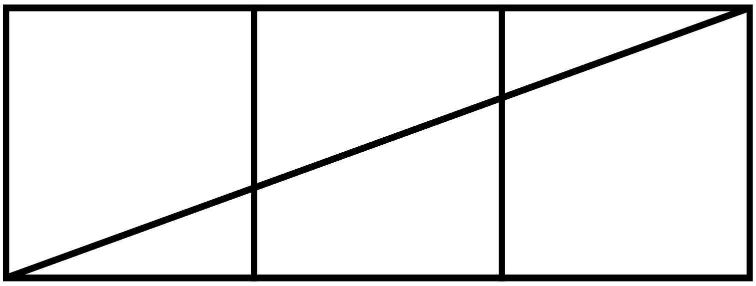
$$\frac{1}{\sqrt{2}}(|000\cdots 0\rangle - |111\cdots 1\rangle),$$

existed in the register prior to the measurement.



The entire quantum register is in a

superposition.



0 0 0 + 1 1 1	0 0 0 - 1 1 1
0 0 1 + 1 1 0	0 0 1 - 1 1 0
0 1 0 + 1 0 1	0 1 0 - 1 0 1
0 1 1 + 1 0 0	0 1 1 - 1 0 0

The qubits are entangled:

measure one,

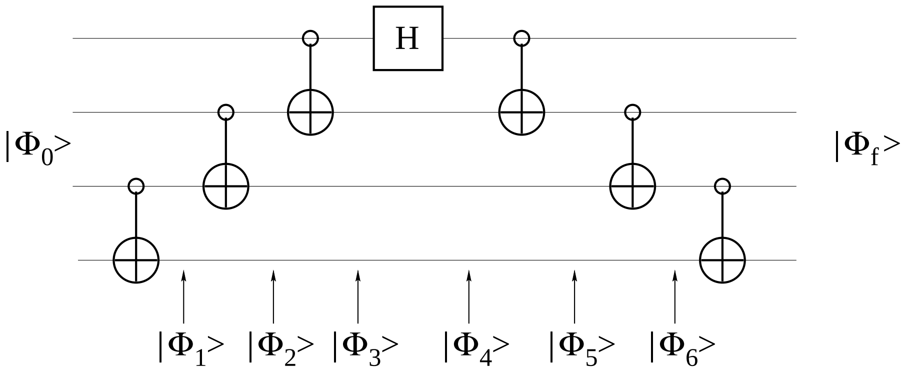
and the superposition collapses.

0	0	0

0 0 0 + 1 1 1 ?

0 0 0 - 1 1 1 ?

$$
\begin{aligned}
\frac{1}{\sqrt{2}}(|000\cdots 0\rangle + |111\cdots 1\rangle) &\longleftrightarrow |000\cdots 0\rangle, \\
\frac{1}{\sqrt{2}}(|000\cdots 0\rangle - |111\cdots 1\rangle) &\longleftrightarrow |111\cdots 1\rangle, \\
\frac{1}{\sqrt{2}}(|000\cdots 1\rangle + |111\cdots 0\rangle) &\longleftrightarrow |000\cdots 1\rangle, \\
\frac{1}{\sqrt{2}}(|000\cdots 1\rangle - |111\cdots 0\rangle) &\longleftrightarrow |111\cdots 0\rangle, \\
&\vdots \\
\frac{1}{\sqrt{2}}(|011\cdots 1\rangle + |100\cdots 0\rangle) &\longleftrightarrow |011\cdots 1\rangle, \\
\frac{1}{\sqrt{2}}(|011\cdots 1\rangle - |100\cdots 0\rangle) &\longleftrightarrow |100\cdots 0\rangle.
\end{aligned}
$$



$|\Phi_0\rangle = \frac{1}{\sqrt{2}}|0000\rangle + \frac{1}{\sqrt{2}}|1111\rangle$

$|\Phi_1\rangle = \frac{1}{\sqrt{2}}|0000\rangle + \frac{1}{\sqrt{2}}|1110\rangle$

$|\Phi_2\rangle = \frac{1}{\sqrt{2}}|0000\rangle + \frac{1}{\sqrt{2}}|1100\rangle$

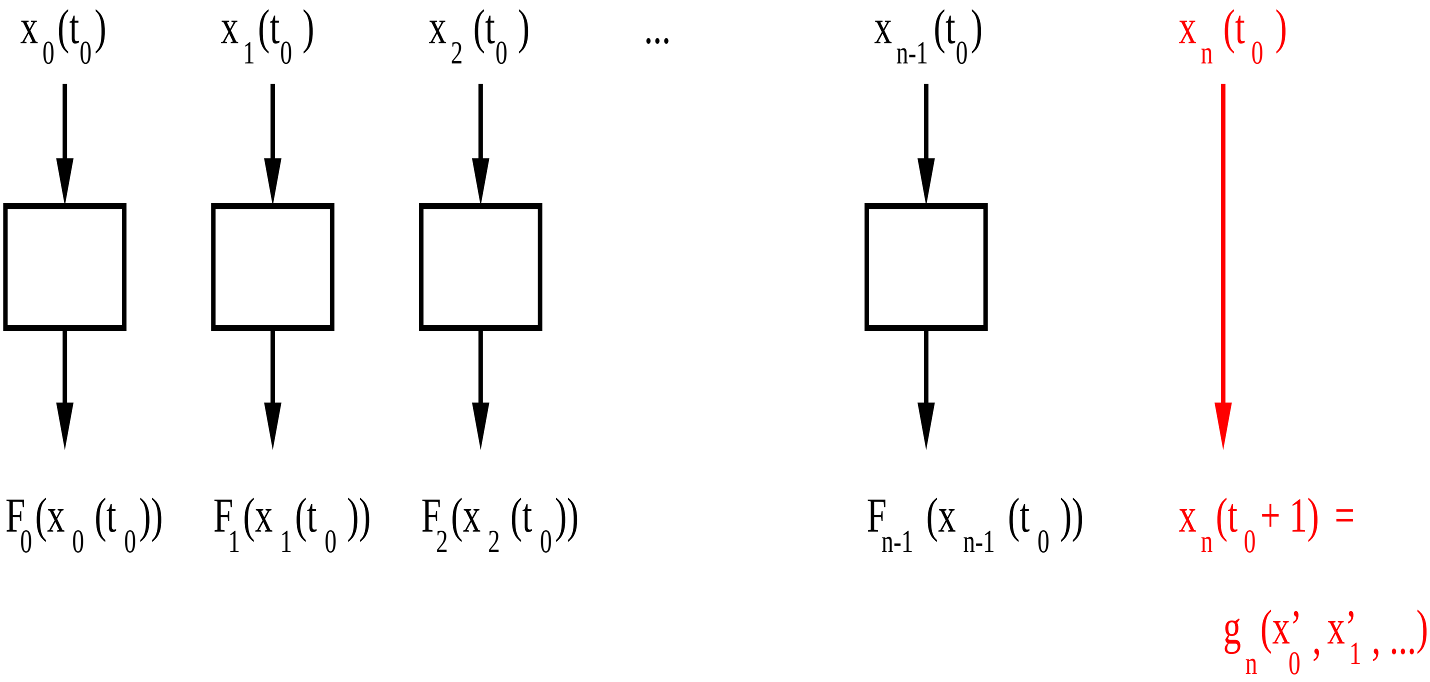
$|\Phi_3\rangle = \frac{1}{\sqrt{2}}|0000\rangle + \frac{1}{\sqrt{2}}|1000\rangle = (\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle) \otimes |000\rangle$

$|\Phi_4\rangle = |\Phi_5\rangle = |\Phi_6\rangle = |\Phi_f\rangle = |0000\rangle.$

However,

for $n+1$ inputs

n processors do not suffice!



PARADIGM 5

5. Mathematical Constraints

$$x_0, \; x_1, \; \ldots, \; x_{n-1},$$

are all available

- they even already reside in memory.

The present computation has three distinguishing properties:

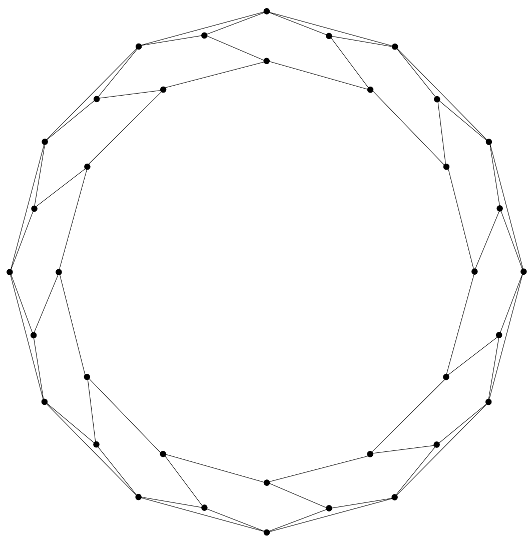
1. All the x_i obey a certain global condition (a mathematical property).
   - This condition must hold throughout the computation.
   - Otherwise, the computation fails.

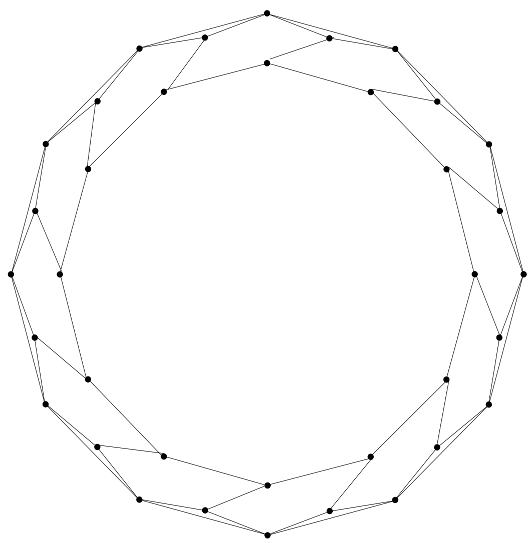
2. Applying F_i to x_i produces a new value for x_i:

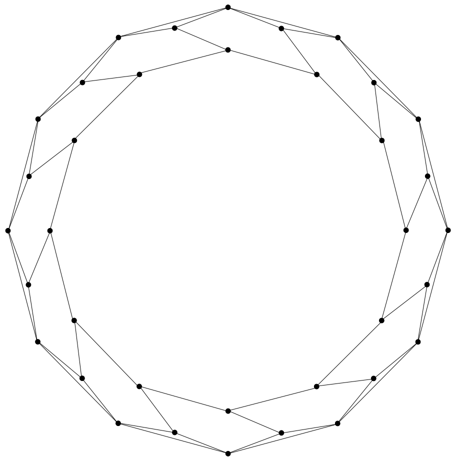
$$x_i^{\text{new}} = F_i(x_i), \quad 0 \leq i \leq n-1.$$

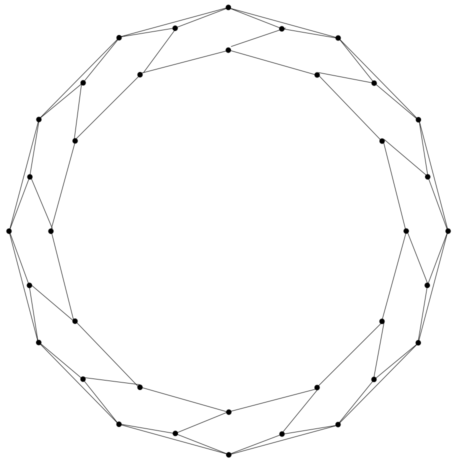
3. IF $F_i(x_i)$ is computed for any one of the variables the global condition is no longer be satisfied.

COMPUTING SUBJECT TO A GEOMETRIC CONSTRAINT







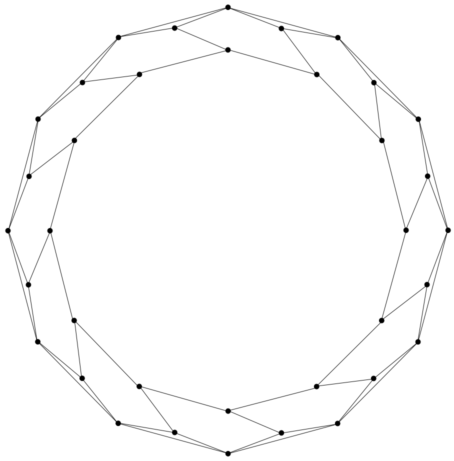


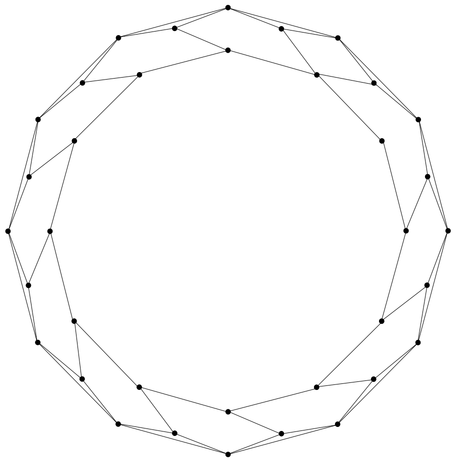
The transformation is effected by removing edges and replacing them with other edges.

Condition for a successful transformation:

each intermediate figure must be a convex subdivision with the same number of edges

There are n edges in the original subdivision that can be removed and replaced with another n edges to produce the new subdivision.





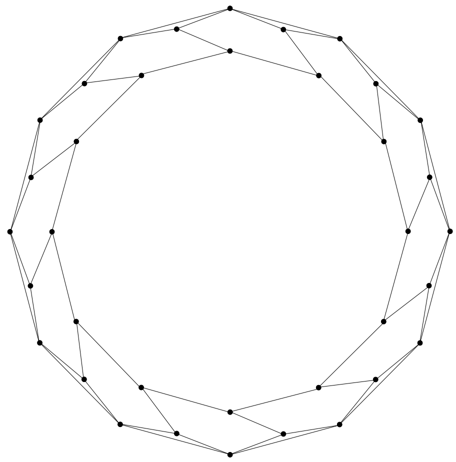
However, removing any one of these n edges and replacing it with another

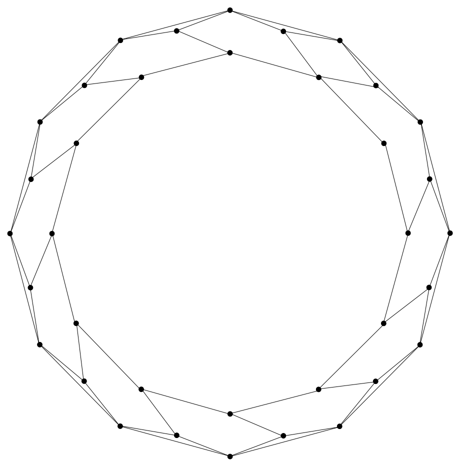
creates a concavity,

thus violating the condition.

CONVENTIONAL SOLUTION

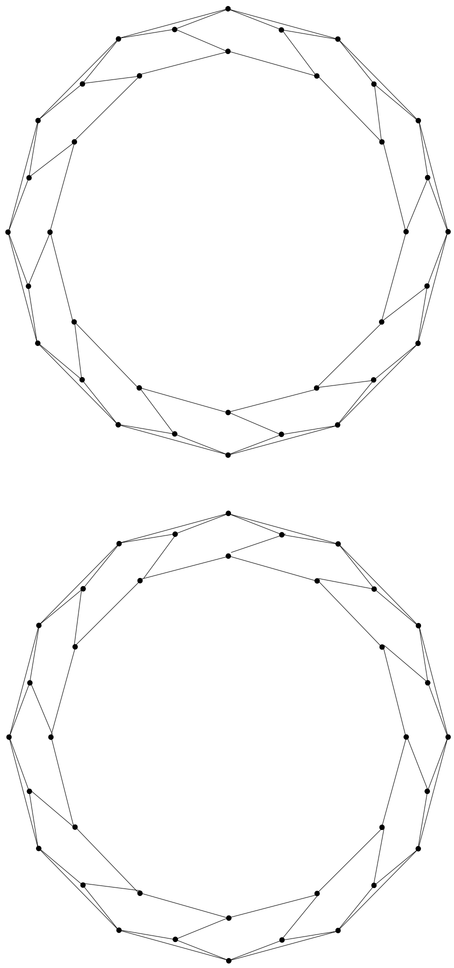
Only one edge of the subdivision can be replaced at a time.





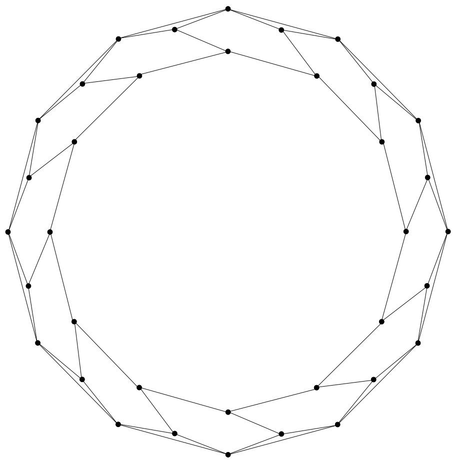
Once any one of the n candidate edges is replaced, the global condition of convexity no longer holds.

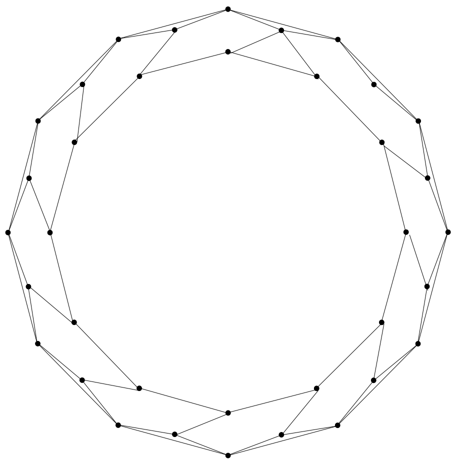
UNCONVENTIONAL SOLUTION



All spokes are replaced simultaneously.

However, $n-1$ processors do not suffice!





UNIVERSALITY

NO FINITE COMPUTER IS UNIVERSAL

Given n spatially and temporally connected variables

$$x_0, x_1, \ldots, x_{n-1}$$

Compute

$$\mathcal{F}(x_0, x_1, \ldots, x_{n-1})$$

NO FINITE MACHINE CAN COMPUTE $\mathcal{F}$

for all values of n

even if allowed infinite time and memory

Suppose there exists a Universal Computer $\mathcal{U}$

capable of n elementary operations per step

(n finite and fixed)

$\mathcal{U}$ fails to perform a computation

requiring n' operations per step

$$n' > n$$

$\mathcal{U}$ is not universal

For each

$$n' > n$$

another computer $\mathcal{U}'$

capable of n' operations per step
succeeds

only to be defeated by a computation

requiring n'' operations per step

$$n'' > n'$$

An n processor parallel computer can solve every problem of size n in

1. Time-Varying Variables
2. Time-Varying Complexity
3. Rank-Varying Complexity
4. Interacting Variables
5. Mathematical Constraints

but not a problem of size $n+1$.

This remains true even if the parallel computer is

- endowed with unlimited memory

- allowed to compute for an indefinite amount of time.

This result applies to all computers obeying the

finiteness condition

All computers capable of only

a finite and fixed number of operations per step.

- Theoretical models
  - Turing Machine
  - Random Access Machine
  - ...
- Practical general-purpose computers
  - Conventional
  - Unconventional
    * Biological
    * Quantum
    * Accelerating
    * ...

WHAT IF WE ALLOWED

THE NUMBER OF OPERATONS

PER STEP

TO BE A VARIABLE?

ACCELERATING MACHINES Computers that increase their speed at every step.

Example 1:

Computers that double the number of operations that they can do at every step.

2, 4, 8, 16, 32, 64, ...

Example 2:

Computers that square the number of operations that they can do at every step.

2, 4, 16, 256, 65536, 4294967296, ...

These machines still obey the finiteness condition: The number of operations per step is finite and fixed, but not a constant.

THE GOLDBACH CONJECTURE

Christian Goldbach (1690 - 1764)

conjectured that every even number greater than 2 is the sum of two prime numbers.

$$
\begin{aligned}
4 &= 2+2 \\
6 &= 3+3 \\
8 &= 3+5 \\
10 &= 5+5 \\
12 &= 5+7 \\
&\vdots
\end{aligned}
$$

FAMOUS OPEN PROBLEM IN MATHEMATICS

Solution by Accelerating Machine

Test every even number greater than 2:

$$1 + 1/2 + 1/4 + 1/8 + \ldots = 2.$$

ACCELERATING MACHINES DESPITE THEIR EXTRAORDINARY ABILITIES CANNOT BE UNIVERSAL

Computational Problem: Time-varying complexity

The ACCELERATING MACHINE can SQUARE the number of operations it can do at every step.

For $n > 2$ and $0 \leq i \leq n-1$,

the evaluation of the function $F(x_i)$

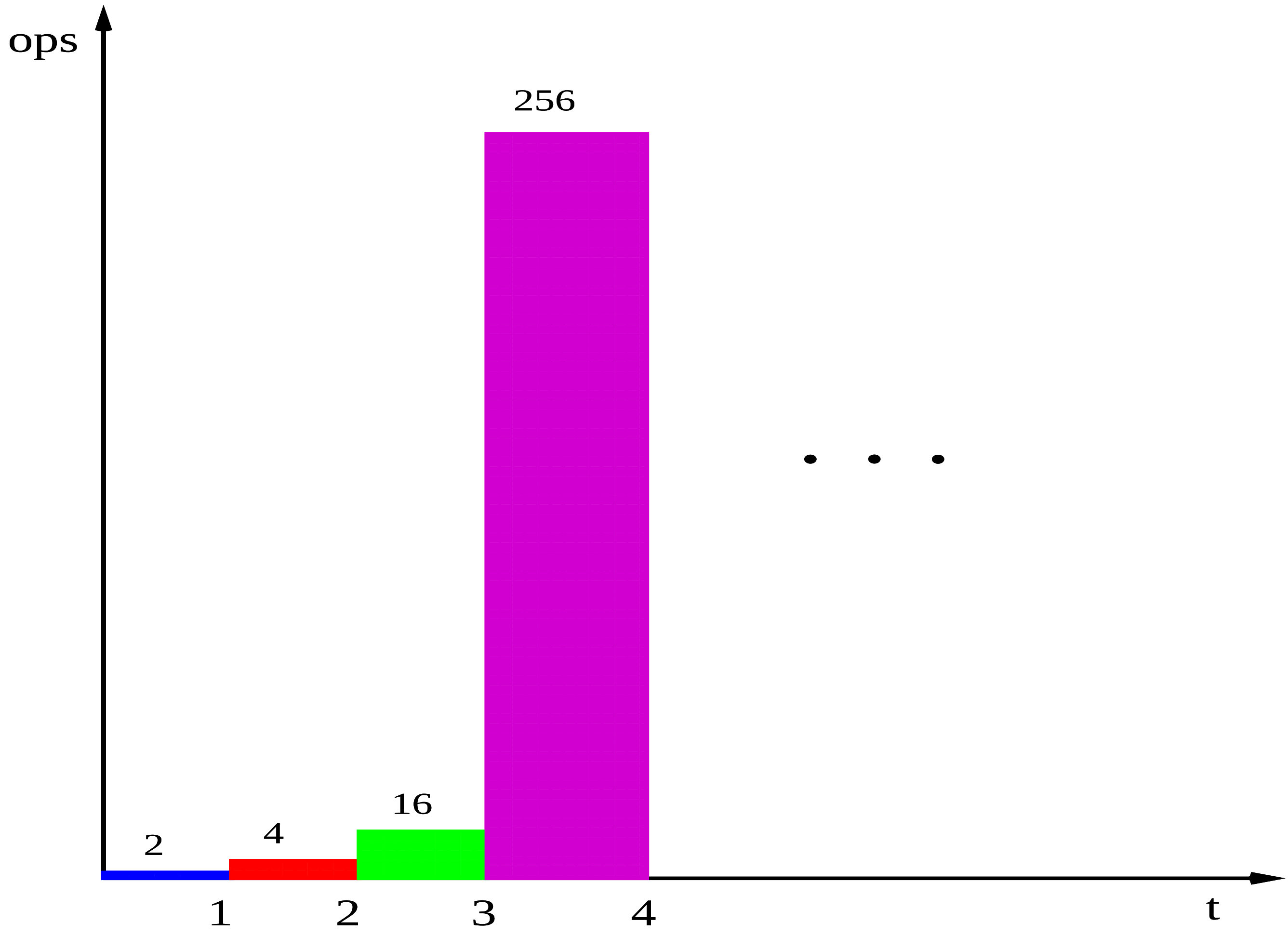
$$\text{requires } 2^{2^t} \text{ operations,}$$

if computed at time t, for $t \geq 0$,

and all of

$$F(x_0), F(x_1), \ldots, F(x_{n-1})$$

must be available at $t = 2$.



This computation is

impossible for the ACCELERATING MACHINE

but trivial for an n-processor computer.

EVEN ACCELERATING MACHINES ARE NOT UNIVERSAL!

"Nowadays, it is difficult to understand how Leibniz could have seriously believed that the universe we inhabit, in all of its complexity, could be reduced to a single symbolic calculus."

Martin Davis, *The Universal Computer*, 2000.

One cannot help but wonder how for seventy years (ever since Turing), we computer scientists have believed that a simple, finite, and fixed 'universal machine' can fully capture the complexity, immensity and ever changing nature of the whole Universe.