COMPUTING AT THE SPEED OF LIGHT

Cristian S. Calude

Department of Computer Science
University of Auckland, New Zealand
www.cs.auckland.ac.nz/˜cristian

Natural Processes and Models of Computation
Bolgona, 2005

Outline

Outline

Outline

Outline

Outline

Photons vs Electrons

Light computing is digital computing using laser light instead of electricity, and holograms instead of silicon computer chips.

Why use laser light instead of electricity?

- *First*, light travels thousands of times faster than electrons (more precisely, electronic signals) in computer chips.
- *Secondly*, light computers could be made of inexpensive plastic and glass that are easier to manufacture than the sand from which electronic computer chips are made (via a very expensive refining).
- *Thirdly*, sophisticated programs that run slowly even on today's supercomputers (for applications in weather prediction, speech recognition, high resolution graphic) could be adequately run on light computers.

Photons vs Electrons

Light computing is digital computing using laser light instead of electricity, and holograms instead of silicon computer chips.
Why use laser light instead of electricity?

- *First*, light travels thousands of times faster than electrons (more precisely, electronic signals) in computer chips.
- *Secondly*, light computers could be made of inexpensive plastic and glass that are easier to manufacture than the sand from which electronic computer chips are made (via a very expensive refining).
- *Thirdly*, sophisticated programs that run slowly even on today's supercomputers (for applications in weather prediction, speech recognition, high resolution graphic) could be adequately run on light computers.

Photons vs Electrons

Light computing is digital computing using laser light instead of electricity, and holograms instead of silicon computer chips.
Why use laser light instead of electricity?

- *First*, light travels thousands of times faster than electrons (more precisely, electronic signals) in computer chips.
- *Secondly*, light computers could be made of inexpensive plastic and glass that are easier to manufacture than the sand from which electronic computer chips are made (via a very expensive refining).
- *Thirdly*, sophisticated programs that run slowly even on today's supercomputers (for applications in weather prediction, speech recognition, high resolution graphic) could be adequately run on light computers.

Photons vs Electrons

Light computing is digital computing using laser light instead of electricity, and holograms instead of silicon computer chips.
Why use laser light instead of electricity?

- *First*, light travels thousands of times faster than electrons (more precisely, electronic signals) in computer chips.
- *Secondly*, light computers could be made of inexpensive plastic and glass that are easier to manufacture than the sand from which electronic computer chips are made (via a very expensive refining).
- *Thirdly*, sophisticated programs that run slowly even on today's supercomputers (for applications in weather prediction, speech recognition, high resolution graphic) could be adequately run on light computers.

Photons vs Electrons

Light computing is digital computing using laser light instead of electricity, and holograms instead of silicon computer chips.
Why use laser light instead of electricity?

- *First*, light travels thousands of times faster than electrons (more precisely, electronic signals) in computer chips.
- *Secondly*, light computers could be made of inexpensive plastic and glass that are easier to manufacture than the sand from which electronic computer chips are made (via a very expensive refining).
- *Thirdly*, sophisticated programs that run slowly even on today's supercomputers (for applications in weather prediction, speech recognition, high resolution graphic) could be adequately run on light computers.

Outline

RainbowSort

RainbowSort, proposed by D. Schultes, is inspired by the observation that light that traverses a prism is sorted by wavelength. It is based on the physical concepts of refraction R and dispersion. D

If there are n elements to be sorted, then

- RainbowSort's time complexity is $\Theta(n)$,
- RainbowSort's space complexity is $\Theta(n)$.

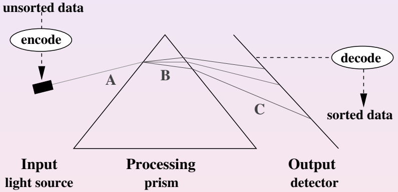
RainbowSort

RainbowSort, proposed by D. Schultes, is inspired by the observation that light that traverses a prism is sorted by wavelength. It is based on the physical concepts of refraction ▸ R and dispersion. ▸ D

If there are n elements to be sorted, then

- RainbowSort's time complexity is $\Theta(n)$,
- RainbowSort's space complexity is $\Theta(n)$.

RainbowSort Principle



Use the runtime of the rays to the detector, which increases for decreasing wavelengths, as sorting criterion: rays with longer wavelengths arrive earlier than rays with smaller wavelengths.

▸ RSA

Outline

Deutsch's Problem

Given a function $f(x)$ from $\{0, 1\}$ into itself, as a black box, determine whether f is "constant" ($f(0) = f(1)$) or "balanced" ($f(0) \neq f(1)$) by computing f on one input.

One bit of information is enough to distinguish the two sets, but classically there's no way of calculating the bit we need: it is the XOR of two function outputs.

Can we do it better? The positive answer was given by Deutsch's algorithm.

Deutsch's Problem

Given a function $f(x)$ from $\{0, 1\}$ into itself, as a black box, determine whether f is "constant" ($f(0) = f(1)$) or "balanced" ($f(0) \neq f(1)$) by computing f on one input.

One bit of information is enough to distinguish the two sets, but classically there's no way of calculating the bit we need: it is the XOR of two function outputs.

Can we do it better? The positive answer was given by Deutsch's algorithm.

Deutsch's Problem

Given a function $f(x)$ from $\{0, 1\}$ into itself, as a black box, determine whether f is "constant" ($f(0) = f(1)$) or "balanced" ($f(0) \neq f(1)$) by computing f on one input.

One bit of information is enough to distinguish the two sets, but classically there's no way of calculating the bit we need: it is the XOR of two function outputs.

Can we do it better? The positive answer was given by Deutsch's algorithm.

Outline

Deutsch's Algorithm

A "quantum" black box can compute $f(0)$ and $f(1)$, but also, and more importantly, it can extract some information about f which tells us whether $f(0)$ is equal or not to $f(1)$.

To do this we use the black box on a superposition of the quantum bits $|0\rangle$ and $|1\rangle$. A mirror can be used for implementation. ▸ PPict

Deutsch's Algorithm: Mirror's Implementation

M. Stay's implementation of Deutsch's algorithm:

1. Turn the input polarizer to 45 degrees.
2. Turn the output polarizer to 45 degrees.
3. Shine light into the box. If we don't see any light coming out, it's a constant function; otherwise it's a balanced function.

Deutsch's Algorithm: Mirror's Implementation

M. Stay's implementation of Deutsch's algorithm:

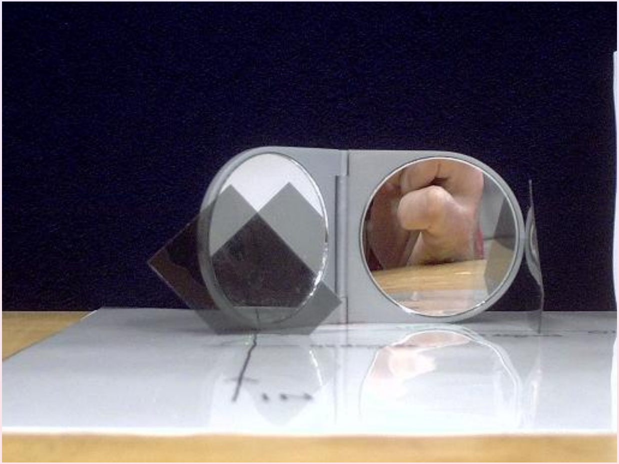
1. `Turn the input polarizer to 45 degrees.`
2. `Turn the output polarizer to 45 degrees.`
3. `Shine light into the box. If we don't see any light coming out, it's a constant function; otherwise it's a balanced function.`

Deutsch's Algorithm: Mirror's Implementation

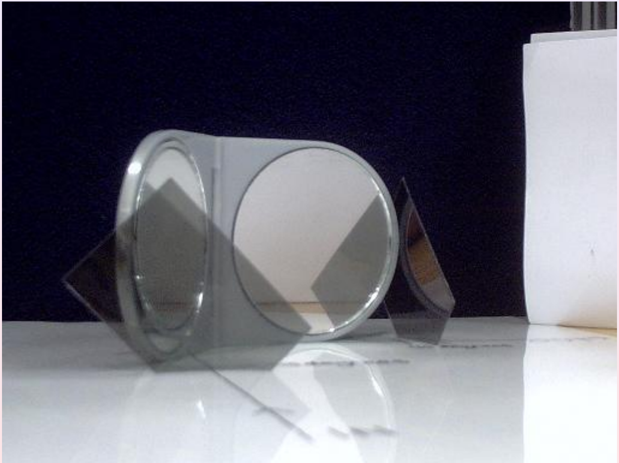
M. Stay's implementation of Deutsch's algorithm:

1. `Turn the input polarizer to 45 degrees.`
2. `Turn the output polarizer to 45 degrees.`
3. `Shine light into the box. If we don't see any light coming out, it's a constant function; otherwise it's a balanced function.`

Deutsch's Algorithm: Balanced Function



Deutsch's Algorithm: Constant Function



Outline

Three Forms of Randomness

- John von Neumann: 'Anyone who considers arithmetical [read: software generated] methods of producing random digits is, of course, in a state of sin'.
- Algorithmic randomness means "computational incompressibility".
- Which slit the electron went through in the double slit experiment is "random". Randomness is intrinsically part of the standard model of quantum mechanics.

Three Forms of Randomness

- John von Neumann: 'Anyone who considers arithmetical [read: software generated] methods of producing random digits is, of course, in a state of sin'.
- Algorithmic randomness means "computational incompressibility".
- Which slit the electron went through in the double slit experiment is "random". Randomness is intrinsically part of the standard model of quantum mechanics.

Three Forms of Randomness

- John von Neumann: 'Anyone who considers arithmetical [read: software generated] methods of producing random digits is, of course, in a state of sin'.
- Algorithmic randomness means "computational incompressibility".
- Which slit the electron went through in the double slit experiment is "random". Randomness is intrinsically part of the standard model of quantum mechanics.

Outline

Characteristics of Algorithmic Randomness

- It satisfies all computable enumerable statistical properties of randomness.
- It is unpredictable.
- It is Turing uncomputable.
- Every "decent" Monte Carlo simulation algorithm (like Rabin's primality test) powered with algorithmic randomness produces the result not only true with high probability, but rigourously correct.

Characteristics of Algorithmic Randomness

- It satisfies all computable enumerable statistical properties of randomness.
- It is unpredictable.
- It is Turing uncomputable.
- Every "decent" Monte Carlo simulation algorithm (like Rabin's primality test) powered with algorithmic randomness produces the result not only true with high probability, but rigourously correct.

Characteristics of Algorithmic Randomness

- It satisfies all computable enumerable statistical properties of randomness.
- It is unpredictable.
- It is Turing uncomputable.
- Every "decent" Monte Carlo simulation algorithm (like Rabin's primality test) powered with algorithmic randomness produces the result not only true with high probability, but rigourously correct.

Characteristics of Algorithmic Randomness

- It satisfies all computable enumerable statistical properties of randomness.
- It is unpredictable.
- It is Turing uncomputable.
- Every "decent" Monte Carlo simulation algorithm (like Rabin's primality test) powered with algorithmic randomness produces the result not only true with high probability, but rigourously correct.

Outline

Characteristics of Quantum Randomness

- It has been confirmed by theoretical and experimental research.
- It passes all reasonable statistical properties of randomness.
- It is Turing uncomputable.
- Can be easily and reliably produced: A photon generated by a source beamed to a semitransparent mirror is reflected or transmitted with 50 per cent chance, and these measurements can be translated into a string of quantum random bits.

Characteristics of Quantum Randomness

- It has been confirmed by theoretical and experimental research.
- It passes all reasonable statistical properties of randomness.
- It is Turing uncomputable.
- Can be easily and reliably produced: A photon generated by a source beamed to a semitransparent mirror is reflected or transmitted with 50 per cent chance, and these measurements can be translated into a string of quantum random bits.

Characteristics of Quantum Randomness

- It has been confirmed by theoretical and experimental research.
- It passes all reasonable statistical properties of randomness.
- It is Turing uncomputable.
- Can be easily and reliably produced: A photon generated by a source beamed to a semitransparent mirror is reflected or transmitted with 50 per cent chance, and these measurements can be translated into a string of quantum random bits.

Characteristics of Quantum Randomness

- It has been confirmed by theoretical and experimental research.
- It passes all reasonable statistical properties of randomness.
- It is Turing uncomputable.
- Can be easily and reliably produced: A photon generated by a source beamed to a semitransparent mirror is reflected or transmitted with 50 per cent chance, and these measurements can be translated into a string of quantum random bits.

JPict

Quantis

Quantis: quantum mechanical random number generator produced and sold by *id Quantique* of the University of Geneva

Outline

The smallest number expressible as the sum of two cubes in n different ways is called $\mathit{Taxicab}(n)$.

$\mathit{Taxicab}(2) = 1729$; $\mathit{Taxicab}(5) = 48988659276962496$.

The value of $\mathit{Taxicab}(6)$ is not known.

Using a sample of 562,500 quantum random integers drawn by Quantis from the interval $[10^{18}, 24153319581254312065344]$, C.S.Calude, E. Calude and M.J. Dinneen have proved that with probability greater than 99.8%,

$$\mathit{Taxicab}(6) = 24153319581254312065344.$$

The smallest number expressible as the sum of two cubes in n different ways is called *Taxicab*(n).

Taxicab$(2) = 1729$; *Taxicab*$(5) = 48988659276962496$.

The value of *Taxicab*(6) is not known.

Using a sample of 562,500 quantum random integers drawn by Quantis from the interval $[10^{18}, 24153319581254312065344]$, C.S.Calude, E. Calude and M.J. Dinneen have proved that with probability greater than 99.8%,

$$Taxicab(6) = 24153319581254312065344.$$

The smallest number expressible as the sum of two cubes in n different ways is called $Taxicab(n)$.

$Taxicab(2) = 1729$; $Taxicab(5) = 48988659276962496$.

The value of $Taxicab(6)$ is not known.

Using a sample of 562,500 quantum random integers drawn by Quantis from the interval $[10^{18}, 24153319581254312065344]$, C.S.Calude, E. Calude and M.J. Dinneen have proved that with probability greater than 99.8%,

$$Taxicab(6) = 24153319581254312065344.$$

The smallest number expressible as the sum of two cubes in n different ways is called $Taxicab(n)$.

$Taxicab(2) = 1729$; $Taxicab(5) = 48988659276962496$.

The value of $Taxicab(6)$ is not known.

Using a sample of 562,500 quantum random integers drawn by Quantis from the interval $[10^{18}, 24153319581254312065344]$, C.S.Calude, E. Calude and M.J. Dinneen have proved that with probability greater than 99.8%,

$$Taxicab(6) = 24153319581254312065344.$$

Outline

Open Questions

The hybrid computer "PC plus Quantis" (used theoretically to generate an infinite sequence of quantum random bits) trespasses the Turing barrier. The machine exists and was used.

1. What is the computational power of the hybrid machine "PC plus Quantis"? If Quantis could generate an "algorithmically random sequence", then the machine "PC plus Quantis" would solve the Halting Problem.
2. How random is quantum randomness? More precisely, to what extent is quantum randomness algorithmic randomness?
3. How accurate are simulations powered by quantum random bits?

Open Questions

The hybrid computer "PC plus Quantis" (used theoretically to generate an infinite sequence of quantum random bits) trespasses the Turing barrier. The machine exists and was used.

1. What is the computational power of the hybrid machine "PC plus Quantis"? If Quantis could generate an "algorithmically random sequence", then the machine "PC plus Quantis" would solve the Halting Problem.
2. How random is quantum randomness? More precisely, to what extent is quantum randomness algorithmic randomness?
3. How accurate are simulations powered by quantum random bits?

Open Questions

The hybrid computer "PC plus Quantis" (used theoretically to generate an infinite sequence of quantum random bits) trespasses the Turing barrier. The machine exists and was used.

1. What is the computational power of the hybrid machine "PC plus Quantis"? If Quantis could generate an "algorithmically random sequence", then the machine "PC plus Quantis" would solve the Halting Problem.
2. How random is quantum randomness? More precisely, to what extent is quantum randomness algorithmic randomness?
3. How accurate are simulations powered by quantum random bits?

Open Questions

The hybrid computer "PC plus Quantis" (used theoretically to generate an infinite sequence of quantum random bits) trespasses the Turing barrier. The machine exists and was used.

1. What is the computational power of the hybrid machine "PC plus Quantis"? If Quantis could generate an "algorithmically random sequence", then the machine "PC plus Quantis" would solve the Halting Problem.
2. How random is quantum randomness? More precisely, to what extent is quantum randomness algorithmic randomness?
3. How accurate are simulations powered by quantum random bits?

Summary

- Light is a viable alternative for classical and quantum computing.
- Computing with quantum randomness is fast and reliable.
- Understanding the capability and limits of computing with light is a challenging and difficult problem.

Summary

- Light is a viable alternative for classical and quantum computing.
- Computing with quantum randomness is fast and reliable.
- Understanding the capability and limits of computing with light is a challenging and difficult problem.

Summary

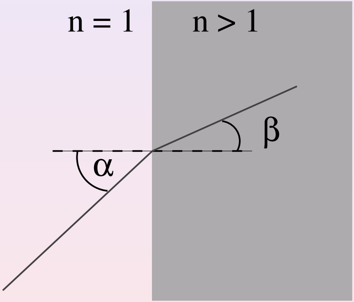
- Light is a viable alternative for classical and quantum computing.
- Computing with quantum randomness is fast and reliable.
- Understanding the capability and limits of computing with light is a challenging and difficult problem.

Further Reading

- C S. Calude. *Information and Randomness*, 2nd Edition, Revised and Extended, Springer-Verlag, Berlin, 2002.
- C. S. Calude. Algorithmic randomness, quantum physics, and incompleteness, in M. Margenstern (ed.). *Proc. MCU'2004*, Lectures Notes in Comput. Sci. 3354, Springer, Berlin, 2005, 1–17.
- C. S. Calude, Elena Calude, M. J. Dinneen. What is the value of *Taxicab*(6)?, An Update, *CDMTCS Research Report* 261, 2005.
- D. Schultes. Rainbow Sort: Sorting at the Speed of Light, *Journal of Natural Computing*, accepted.
- M. Stay. Inexpensive linear-optical implementations of Deutsch's algorthm, *EATCS Bull.* 84 (2004), 195–202.

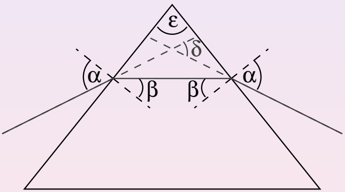
Appendix Outline

RainbowSort



A light ray that moves through vacuum entering a transparent material with a refraction index $n > 1$ gets refracted. ▸ RS

RainbowSort



When a ray traverses a prism, it is refracted twice, once when it enters the prism and once when it leaves it. The angle between the incoming and the outgoing ray is the angular deviation δ. The bigger the refraction index n, the higher is the deviation δ. RS

RainbowSort Algorithm

1. The input is a light source with the unsorted input data encoded into wavelengths: each number of the input is mapped to a wavelength. The light source generates a ray whose spectrum consists exactly of the wavelengths that represent the input data.
2. The processing part consists of a prism onto which the generated ray is sent through. Due to refraction and dispersion, the ray is split into n monochromatic rays that are sorted by wavelength.
3. On the output side a detector receives the incoming rays. It is positioned in such a way that the length of the path from the prism is maximal for the minimum wavelength and minimal for the maximum wavelength. The detector decodes the incoming rays and outputs the sorted data.

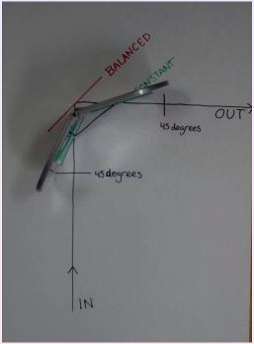
RainbowSort Algorithm

1. The input is a light source with the unsorted input data encoded into wavelengths: each number of the input is mapped to a wavelength. The light source generates a ray whose spectrum consists exactly of the wavelengths that represent the input data.
2. The processing part consists of a prism onto which the generated ray is sent through. Due to refraction and dispersion, the ray is split into *n* monochromatic rays that are sorted by wavelength.
3. On the output side a detector receives the incoming rays. It is positioned in such a way that the length of the path from the prism is maximal for the minimum wavelength and minimal for the maximum wavelength. The detector decodes the incoming rays and outputs the sorted data.

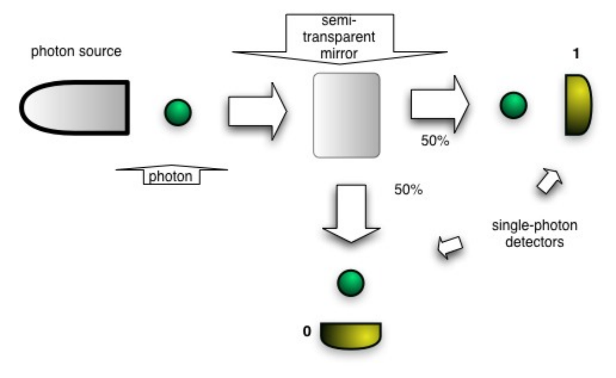
RainbowSort Algorithm

1. The input is a light source with the unsorted input data encoded into wavelengths: each number of the input is mapped to a wavelength. The light source generates a ray whose spectrum consists exactly of the wavelengths that represent the input data.
2. The processing part consists of a prism onto which the generated ray is sent through. Due to refraction and dispersion, the ray is split into n monochromatic rays that are sorted by wavelength.
3. On the output side a detector receives the incoming rays. It is positioned in such a way that the length of the path from the prism is maximal for the minimum wavelength and minimal for the maximum wavelength. The detector decodes the incoming rays and outputs the sorted data.

RSP



A beam of light reflecting off a mirror aligned as above will have left-right polarization switched, while up-down polarization remains unaffected. DA



Optical system for generating quantum random bits

Abstract

Cheap, fast, and reliable computations can be implemented with light. This idea will be illustrated with examples from classical computing (RainbowSort) and quantum computing (a mirror implementation of Deutsch's algorithm). Light can be also used to produce quantum random bits which are qualitatively superior to any software generated pseudo-random bits. The hybrid computer "PC + random quantum generator" is not a 'paper machine', it is a real machine; this machine will be demonstrated with the probabilistic computation of the smallest number expressible as the sum of two cubes in six different ways—currently an open mathematical question. More importantly, the computational power of the hybrid computer trespasses the Turing barrier. This phenomenon raises fascinating open questions like 'Can the hybrid computer "PC + random quantum generator" solve the Halting Problem?'.