Two odd things about computation

Daniel Murfet
University of Melbourne

Before I get started with the talk, I'd like to make a few meta comments. Firstly, this was originally a colloquium talk, which I gave last year to the University of Vienna mathematics department, at the invitation of my friend and collaborator Nils Carqueville, who is a mathematical physicist. So the talk originally had a kind of dual purpose. On the one hand, I wanted to entertain a room of non-logicians and maybe convince them there was something interesting about logic, and on the other hand I wanted to explain to my friend Nils some of the reasons why I had suddenly become fascinated with sequent calculus and linear logic.

Now today's audience is of course already convinced logic is interesting, and there are people here that know far more than me about sequent calculus and substructural logics, so this repeat of the talk is preaching to the choir in many respects. But it's my hope that at least one of the motivating examples I chose to tell my mathematical friends about is still new to you.

Two odd things

I. Maxwell's demon (1871)

Energy cost of computation

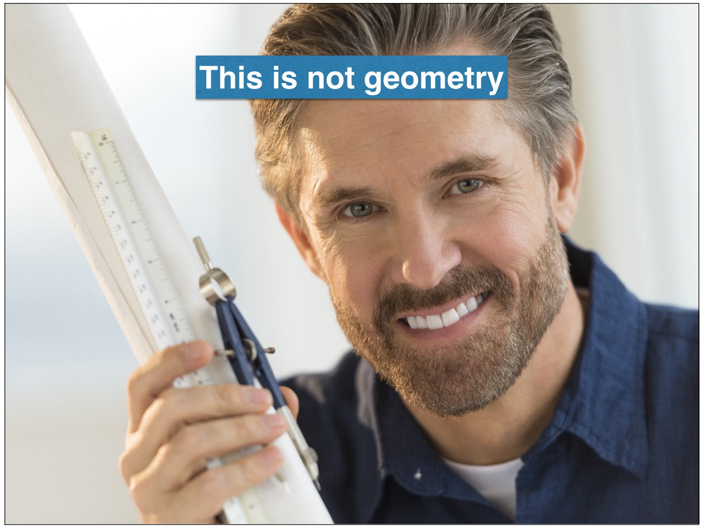
II. Russell's paradox (1901)

Time and space cost of computation

So here are my two examples. They are both paradoxes, and they are both old. And both of them played a fundamental role in the century after they were formulated. In the former case, in physics and the theory of information, and in the second case in logic. For the purpose of today's talk, the link between these two paradoxes is that they both eventually told us something deep about computation.

Now, computation is one of those words that is very hard to define precisely. And I'm not going to try. However, it's important for today's talk that we admit a concept of computation which is not overly tied to any particular choice of mechanism which embodies computational processes. I'd like to suggest why this is important using an analogy with geometry.

[0:40]



Now these tools were undoubtedly important in the discovery of basic geometric concepts, but after hundreds of years of development they no longer look so fundamental.

[0:40]

Similarly, while it's certainly true that these physical tools and practical problems involving them continue to teach us new questions to ask about computation, there is an abstract mathematical theory of computation which both predates them and also involves questions not of immediate practical relevance. It is this broader notion of computation that we're going to be involved with today.

[0:40]

I. Maxwell’s demon

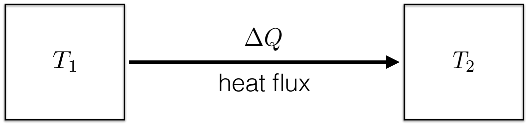
How much energy must be used
in carrying out a computation?

With that preface, let’s get into the first of my examples, a paradox known as Maxwell’s demon. This is the same Maxwell from Maxwell’s equations, and his demon is a thought experiment which played an important role in thermodynamics. From the perspective of computation, it plays a decisive historical role in the development of our current understanding of the minimal energy cost of carrying out a computation.

[0:20]

Quasi-static flux of heat Q into a system at temperature T is associated with an increase of entropy S of that system.

$$\Delta Q = T\Delta S$$



$$-T_1\Delta S_1 = \Delta Q = T_2\Delta S_2$$

$$\Delta S = \Delta S_2 + \Delta S_1 = \left(1 - \frac{T_2}{T_1}\right)\Delta S_2 \;\; = \begin{cases} > 0 & T_1 > T_2 \\ < 0 & T_1 < T_2 \end{cases}$$

To understand Maxwell's demon we need just a little bit of undergraduate physics. Entropy is a difficult concept, but for our purposes it is just a scalar quantity attached to certain kinds of physical systems, which increases or decreases as heat is added or removed from the system. Now, the catch is that the increase of decrease of entropy for a given unit of heat depends on the current temperature according to this equation.

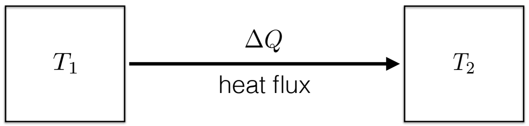
Suppose we have two systems at temperatures T1, T2 and that a unit of heat delta Q flows from the left to the right. Here delta Q is positive. This causes a decrease in the entropy of the left hand system and an increase in the entropy of the right hand system, but the decrease and increase are not of the same magnitude if the temperatures are different. With delta Q fixed, the larger T2 is, for instance, the smaller the increase in entropy of the right hand system.

The total change in entropy of the system is delta S2 plus delta S1, which a little algebra will tell you is given by this expression which is positive if T1 > T2 and negative otherwise. That is, the total change in entropy will be positive if heat flows from high to low temperature, and negative if it flows the other way.

[2:00]

Second law: it is impossible to derive an engine which, working in a cycle, shall produce no effect other than the transfer of heat from a colder to a hotter body.

i.e. in a closed system always $\Delta S_{tot} \geq 0$



$$-T_1 \Delta S_1 = \Delta Q = T_2 \Delta S_2$$

$$\Delta S = \Delta S_2 + \Delta S_1 = \left(1 - \frac{T_2}{T_1}\right) \Delta S_2 \quad = \begin{cases} > 0 & T_1 > T_2 \\ < 0 & T_1 < T_2 \end{cases}$$

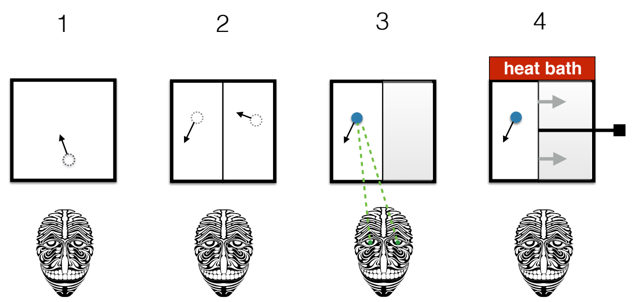
This leads us to the second law of thermodynamics, which says that there is no closed cycle which has the effect of moving heat from a colder to a hotter body, that is, which has negative total entropy. Of course such flows are possible, but they require work to be done from the outside; for instance, moving heat from a cold body such as a refrigerator to a hot body such as the outside air, requires work transferred via electricity from a power plant.

[1:20]

“If someone points out to you that your pet theory of the universe is in disagreement with Maxwell’s equations — then so much the worse for Maxwell’s equations. If it is found to be contradicted by observation — well these experimentalists do bungle things sometimes. But if your theory is found to be against the second law of thermodynamics I can give you no hope; there is nothing for it but to collapse in deepest humiliation.”

–Arthur Stanley Eddington

[0:40]



$$\Delta W = \Delta Q = -kT \ln 2 \qquad \Delta S_{bath} = -k \ln 2$$

In the cycle 1-2-3-4-1, Maxwell's demon seems to violate the second law of thermodynamics

So here is Maxwell's demon. The original paradox is due to Maxwell in 1871 and involves a multi-body gas, but the equivalent version due to Szilard with a single-particle gas is more useful for emphasising the connection to information, and dates from 1929.

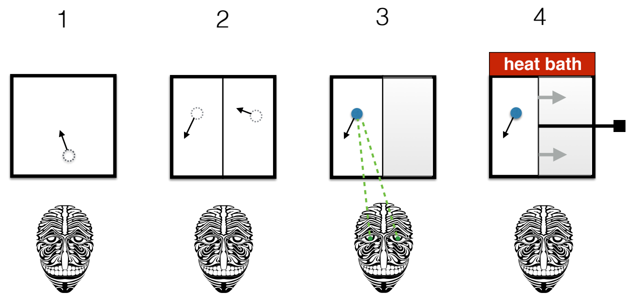
It's a paradox because it is a hypothetical system which can transfer heat from one body to another, even if the second body is at a higher temperature than the first, thus violating the second law of thermodynamics.

The system consists of (A) a box containing a single particle of some ideal gas, at some temperature, bouncing around inside the box in such a way that it reflects elastically off the walls, (B) the demon, who is able to make certain observations of the system and perform certian mechanical actions, that we'll enumerate in a moment and (C) finally, there is a heat bath that will appear in the final step of the process, and which I'll explain then.

The system will be taken through 4 steps. Step 1 is the initial state. In Step 2, the demon inserts an immovable partition quickly in the middle of the box. He does not observe the particle before doing so, so after the partition is added the particle could be in either the left or right half, and he does not know which.

In Step 3 he observes on which side the particle is located. Suppose it is on the left. This means the right hand part of the box is a vacuum at zero pressure, while the left hand side has some finite pressure on the walls from the particle bouncing around. The idea is that we can use this pressure to do mechanical work.

In Step 4 we do two things. Firstly, we hook up a piston to the interior partition, which extends outside the box. If we let the interior partition move quasi-statically, then the pressure on the left will push it to the right, moving the piston and allowing us to do work on some external system.



$$\Delta W = \Delta Q = -kT \ln 2 \qquad \Delta S_{bath} = -k \ln 2$$

In the cycle 1-2-3-4-1, Maxwell's demon seems to violate the second law of thermodynamics

For instance, we could use this work to turn a fan and heat up a pool of water. Now the temperature of the particle and thus the energy drops as it hits the interior partition and does this work, but if the box is connected to a heat bath it is possible to keep restoring this lost energy so that the system remains at a fixed temperature. Now, this continues until the interior partition reaches the right hand wall.

If the fixed temperature is T, a little calculation using the ideal gas law shows that the work done is kTln2, where k is Boltzmann's constant. The work is negative because it represents a flow of energy out of the system. Moreover, the heat transferred from the heat bath to the particle to keep it at a fixed temperature equals this amount of work, which can be used to heat up our external pool of water by the same amount kTln2. In conclusion, then, we have managed to transfer an amount of heat delta Q from the heat bath to that external pool of water, via this cycle 1-2-3-4.

BUT, the heat bath could have been at a lower temperature than the external pool of water! In this case, we have designed a cycle which contradicts the second law of thermodynamics. That is, the total entropy of the cycle is a negative number, -k ln 2.

To be clear, the total system includes the box, particle, heat bath, demon and the pool of water we're heating up. Somehow this closed system underwent a cycle with negative entropy. Now, since we believe in the second law of thermodynamics, we believe that this analysis must be incomplete, that there must be some part of the system which secretly INCREASED its entropy, so that the overall change was in fact positive or zero. But, where is the gap? Which part?

[7:00]

What generates the missing entropy?

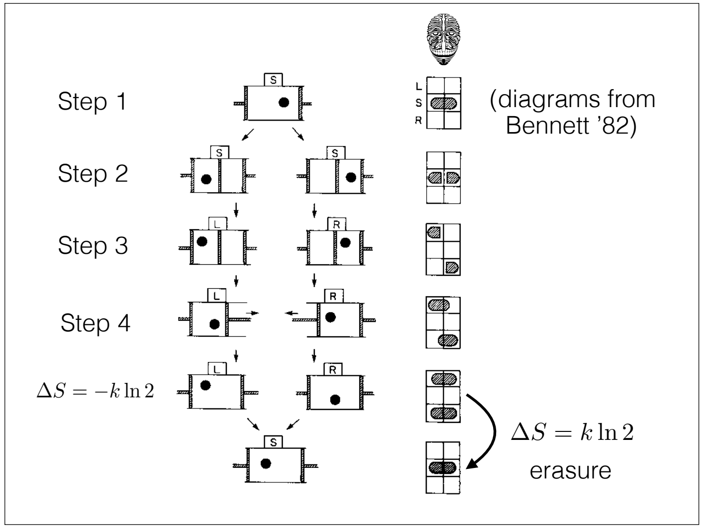
- It's the measurement! - Brillouin '51, Gabor '61.
- No, it's not the measurement! It's the change in *internal state* of the demon, viewed as changing over the course of a cyclic computational process - Landauer '61, Fredkin & Toffoli '82, Bennett '82.

For a long time Maxwell's demon bothered people, but it was generally believed since at least the 50's and probably earlier that the culprit responsible for the secret increase in entropy was the measurement process. This is plausible: for example, you might think that to measure which half contains the particle you need to shine light on it, and in order for that measurement to succeeed that light will have to be sufficiently coherent, and producing this kind of light can be seen to generate entropy. Many other examples of measurement schemes can also be seen to unavoidably generate entropy, so it seems convincing that this is the explanation for Maxwell's demon. That is, you might be able to transfer heat from a cold body to a hot body in one part of the system, but only because you used energy from another part, and overall the second law of thermodynamics is preserved.

But very clever arguments by Landauer, Fredkin & Toffoli and Bennett showed firstly that there exist measurement schemes which DO NOT generate any entropy, and secondly that the correct resolution of the paradox is more subtle, and has to do with how we view the demon as a "computational process" in its own right, with an internal state which evolves over the course of the cycle.

[1:10]

[Cumulative 19 min]

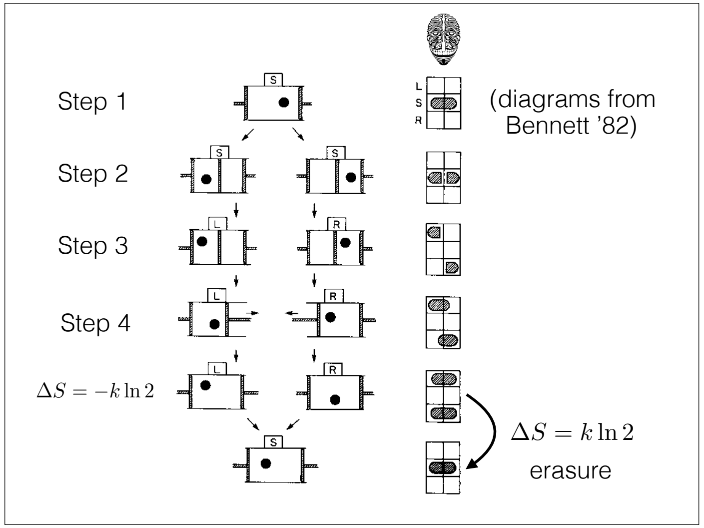


I'd like to explain the resolution of Maxwell's paradox due to Landauer and Bennett, which seems to continue to be generally accepted.

Recall the four steps. But now in addition to the case where the particle was found to be on the left in Step 2 we also consider the possibility that the particle was found on the right. Now, the left hand part of this diagram just represents what we already discussed - Step 2 is partition, Step 3 is observe and Step 4 is extract work. Note that in Step 4 if the particle is on the right then of course the piston connecting to the internal partition comes out to the left. Once the work is done we are back to a box with no partition, and as discussed, the overall entropy (as far as we understood it before) was -kln2.

The new part is on the right, which is a diagram representing the internal state of the demon over the course of this cycle. The internal state is represented by a table with three rows and two columns.

The rows stand for the internal state of the demon. He has three possible states: L, S, and R where S is his standard or initial state. The columns stand for the two branches in the diagram on the left. So to begin with the demon is in his internal state, the box is unpartitioned. Then the box is partitioned, but the demon's state does not change. When the particle is observed in Step 3, his state changes to L if he observed the particle on the left and to R if he observes it on the right. As the partition moves the difference between the state of the two branches starts to converge, until at the end of Step 4 they are the same state where the particle's position is unknown.



However, while the state of the box has now returned to its original configuration, the demon's internal state has not. He still remembers which of the two possibilities actually occurred. To complete the cycle he has to reset his internal state to the standard state, and this 2:1 mapping of his state, or if you like the erasure of one bit of information, is well-understood to incur an entropy cost of kln2.

This increase in entropy precisely cancels the decrease in entropy from the heat transfer out of the heat bath, and thus the second law of thermodynamics is saved and Maxwell's demon exorcised.

[4:00]

- **Landauer's principle:** the only fundamental computational operation which generates entropy is *erasure*.

- **Bennett's exorcism**: the resolution of Maxwell's paradox is that, in order to complete a cycle, the demon's internal state must be restored to its original configuration by *erasing its memory* - thus generating the entropy necessary to put the engine back into compatibility with the second law.

This analysis led to the formulation of Landauer's principle. The only computational operation which cannot be done reversibly, and thus which incurs an unavoidable energy cost, is erasure.

[1:30]

[Cumulative 23:00]

II. Russell’s paradox

How much time and space must be used
in carrying out a computation?

My second example of a paradox is Russell’s paradox, which is certainly more familiar to this audience than Maxwell’s paradox, but it may not be clear to all of you what Russell’s paradox has to do with the subtitle here regarding the time used to carry out of a computation. I’d like to explain that, and then tie both Maxwell’s paradox and Russell’s paradox together (in a fun but not very deep way) using a perspective on computation offerred by sequent calculus.

[0:40]

Comprehension considered harmful

$$R = \{x \mid x \notin x\}$$

$$R \in R \vdash R \notin R$$

$$R \notin R \vdash R \in R$$

We're all familiar with Russell's paradox in naive set theory. I'd like to try and perform a more thorough analysis of what exactly goes wrong, using the sequent calculus of classical logic.

[2:00]

The **sequent** $A_1, \ldots, A_n \vdash B_1, \ldots, B_m$ says:
together the A_i prove at least one of the B_i

$$\frac{}{A \vdash A}\ \text{axiom} \qquad \frac{\Gamma \vdash \Delta, A \qquad A, \Gamma' \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\ \text{cut}$$

$$\frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta}\ \text{ctr} \qquad \frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta} \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}$$

$$\frac{\Gamma \vdash B, B, \Delta}{\Gamma \vdash B, \Delta}\ \text{ctr} \qquad \frac{\Gamma \vdash B_1, B_2, \Delta}{\Gamma \vdash B_1 \vee B_2, \Delta} \qquad \frac{\Gamma, A_1, A_2 \vdash \Delta}{\Gamma, A_1 \wedge A_2 \vdash \Delta}$$

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash B, \Delta}\ \text{weak} \qquad \frac{\Gamma \vdash \Delta}{\Gamma, B \vdash \Delta}\ \text{weak} \qquad \bullet\ \bullet\ \bullet$$

Γ, Δ stand for multisets of formulae

[5:30]

Example of a proof in the sequent calculus

$$\dfrac{\dfrac{\dfrac{}{x \vdash x}\,\text{ax} \qquad \dfrac{\dfrac{}{y \vdash y}\,\text{ax}}{x, y \vdash y}\,\text{weak}}{x, y \vdash y}\,\text{cut}}{x \wedge y \vdash y}$$

[1:00]

Theorem (Gentzen 1934). There is an algorithm which, given a proof, produces a cut-free proof of the same sequent. This algorithm is called *cut-elimination.*

logic	**programming**
formula	type
sequent	input/output spec
proof	program
cut-elimination	execution
contraction	copying
weakening	erasure

Curry-Howard correspondence

[4:00]

unrestricted comprehension

$$\frac{\Gamma \vdash \Delta, A[t/x]}{\Gamma \vdash \Delta, t \in \{x \mid A\}}\text{comp} \qquad \frac{\Gamma, A[t/x] \vdash \Delta}{\Gamma, t \in \{x \mid A\} \vdash \Delta}\text{comp}$$

$$A = x \notin x \qquad R = \{x \mid A\} = \{x \mid x \notin x\}$$

$$R \notin R = A[R/x]$$

Applying the new comprehension rule yields

$$\frac{\begin{array}{c}\vdots\\ \Gamma \vdash \Delta, R \notin R\end{array}}{\Gamma \vdash \Delta, R \in R}\text{comp} \qquad \frac{\begin{array}{c}\vdots\\ \Gamma, R \notin R \vdash \Delta\end{array}}{\Gamma, R \in R \vdash \Delta}\text{comp}$$

[3:00]

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{R \in R \vdash R \in R}\,\text{ax}}{\vdash R \notin R, R \in R}}{\vdash R \in R, R \in R}\,\text{comp}}{\vdash R \in R}\,\text{ctr}
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{}{R \notin R \vdash R \notin R}\,\text{ax}}{R \notin R, R \in R \vdash}}{R \in R, R \in R \vdash}\,\text{comp}}{R \in R \vdash}\,\text{ctr}}{\dfrac{\vdash}{\vdash C}\,\text{weak}}\,\text{cut}
$$

- The sequent calculus of naive set theory is inconsistent.
- The standard point of view is that the culprit is unrestricted comprehension.

[3:00]

logic	**programming**
formula	type
sequent	input/output spec
proof	program
cut-elimination	execution
contraction	copying
weakening	erasure

Curry-Howard correspondence

[2:00]

Theorem (Girard '98): there is a consistent refinement of the sequent calculus with unrestricted comprehension but *restricted contraction* in which the provably total functions $\mathbb{N} \longrightarrow \mathbb{N}$ are precisely the polynomial time functions.

- This refinement is called *light linear logic.*
- The subject of *implicit computational complexity* views time and space complexity of programs as constraints on the static "geometry" of proofs.
- This has been used to propose versions of System F (the programming language on which Haskell is based) with polytime as a typing constraint (Atassi-Baillot-Terui '07).
- Derived from a close analysis of Russell's paradox.

[3:30]

logic	**programming**	
formula	type	
proof	program	
cut-elimination	execution	
contraction	copying	Russell’s paradox
weakening	erasure	Maxwell’s paradox
restricted contraction	restricted complexity	

What is the physical meaning of complexity?

[3:00]

[Cumulative 2nd half: 28]

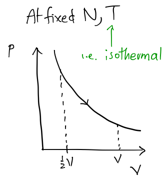
Simple ideal gas

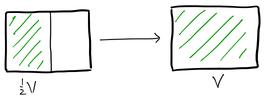
$$PV = NRT \qquad U = cNRT$$

P: pressure; V: volume; N: mole number; R: universal gas constant; T: temperature

U: energy; c: constant, depends on gas

At fixed N, T (i.e. isothermal)





$$W = \int_{\frac{1}{2}V}^{V} P\,dV = \int_{\frac{1}{2}V}^{V} \frac{NRT}{V}\,dV$$

$$= NRT\int_{\frac{1}{2}V}^{V} \frac{1}{V}\,dV = NRT\ln\left(\frac{V}{\frac{1}{2}V}\right)$$

$$= NRT\ln 2$$

$\therefore$ Work per mole $W/N = RT\ln 2$.