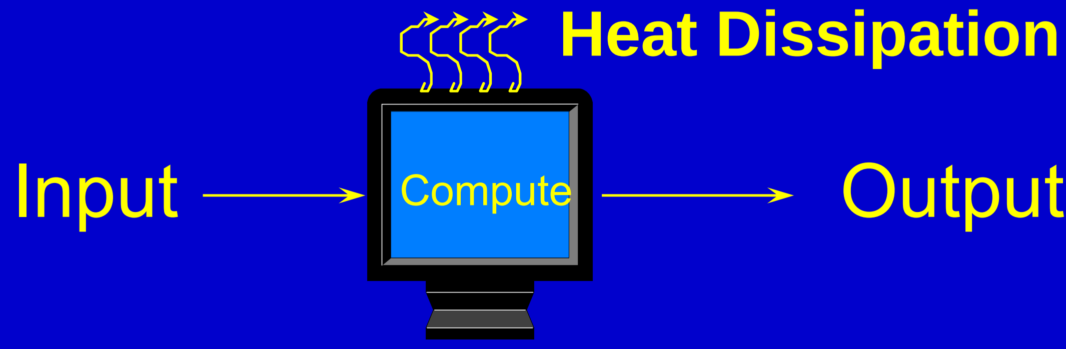
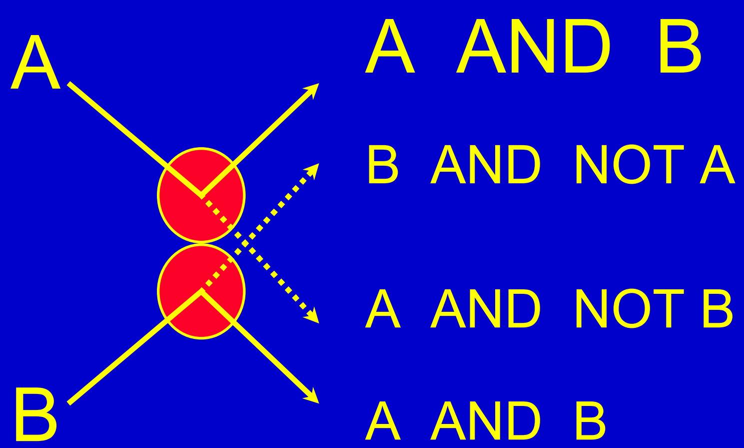
Time, Space, and Energy in Reversible Computing

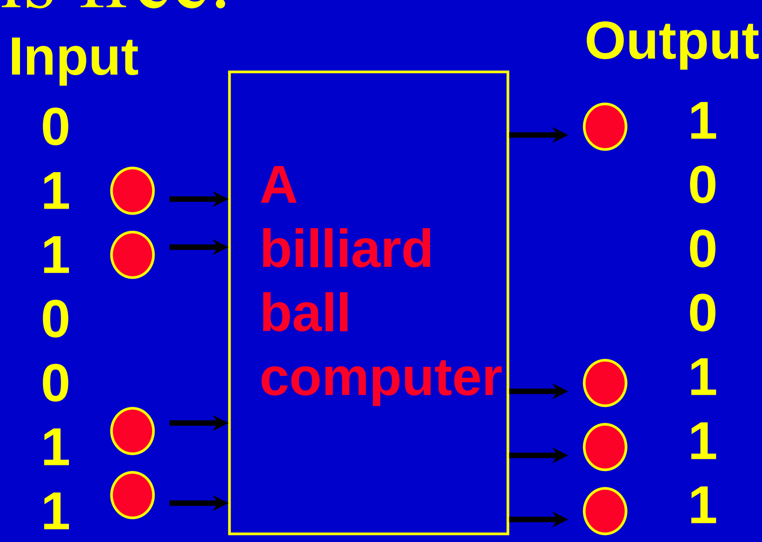
Paul Vitanyi, CWI, University of Amsterdam, National ICT Australia

Thermodynamics of Computing



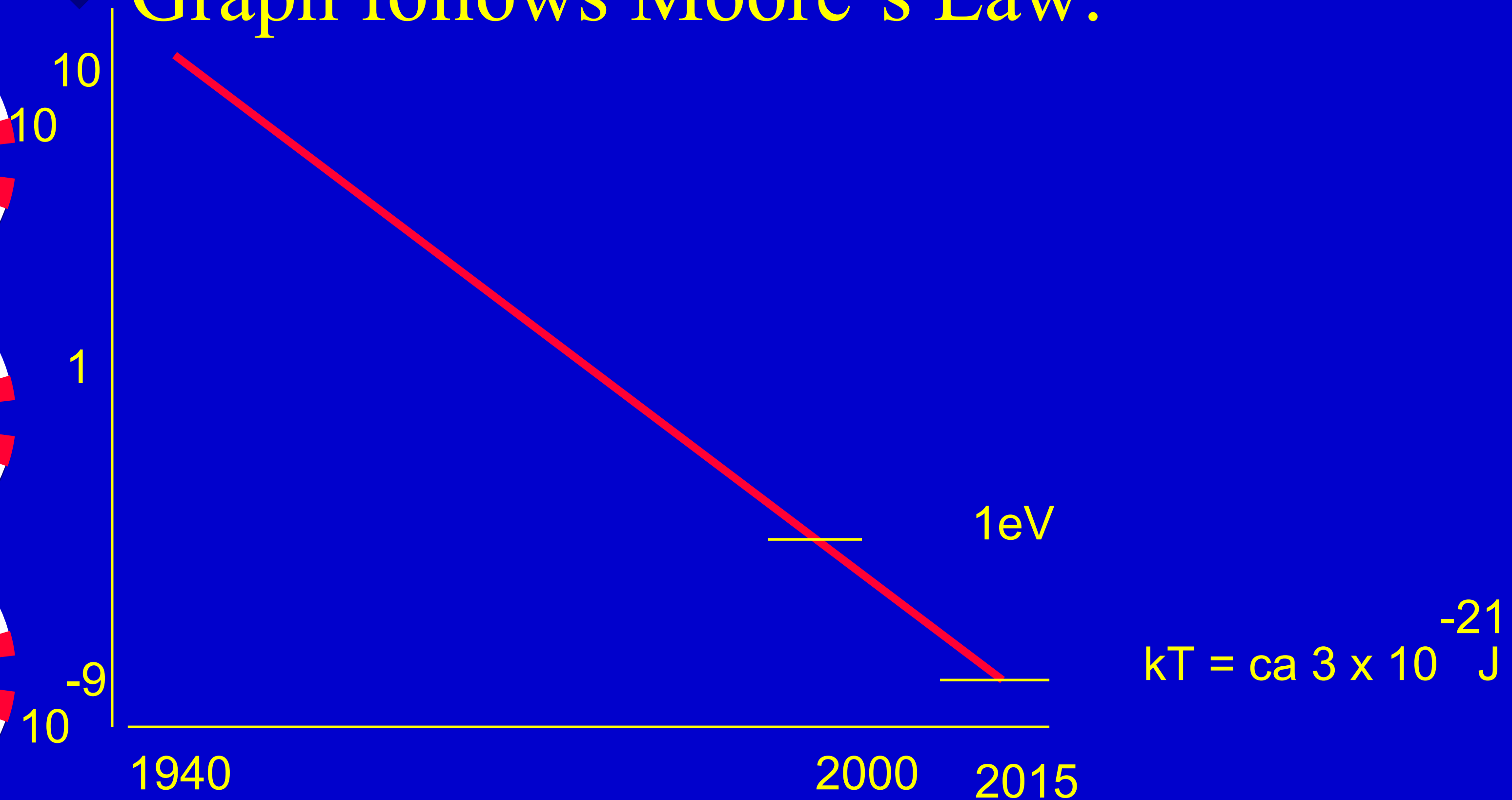
- Physical Law: 1 kT ln 2 is needed to irreversibly process 1 bit (Von Neumann, Landauer)
- Reversible computation is free.





Trend

- Graph follows Moore's Law:



Even at kT, 20 C, 100 Gigahertz, 10 Tera gates/cm3 → 3000 Wat

Relevance:

- Mobilization ←→ Battery improvement
  (20% in 10 years)
- Miniaturization computing devices
- Improved speed + higher density (linear+square)
- More heat = dissipation /cm2 (cubic)
- Matter migration of circuits
- Flaky performance requiring reduced clock speed.

Reversible Universal Computer

- We use Turing Machines for convenience
- It is Reversible if for every operation there is an inverse operation that undoes the effect of the former operation.
- We can effectively enumerate a family of reversible TMs.
- There is a Universal Reversible TM that simulates all others (Lecerf 63, Bennett 73)

Energy

- If we compute x → y reversibly, then actually we do

$$p,x \leftrightarrow y,q$$

Program garbage (p) — Garbage Program (q)

• What is a program at one end, is garbage at the other end. We need to supply the program bits and erase the garbage bits. The best way to do this is to compact them as much as possible (like the garbage collection truck does).

Andrey Nikolaevich Kolmogorov (1903-1987, Tambov, Russia)

- Measure Theory
- Probability
- Analysis
- Intuitionistic Logic
- Cohomology
- Dynamical Systems
- Hydrodynamics
- Kolmogorov complexity

Kolmogorov Complexity

- Defines the ultimate data compression limit: gzip, bzip2, PPMZ → ... → KC. Therefore it gives us the thermodynamically cheapest way to get rid of data garbage.
- Getting rid of 11...1 (n ``1''s) costs n kT ln 2 joule; by first compressing it reversibly to log n bits, erasing costs only (log n)kT ln 2 Joule.

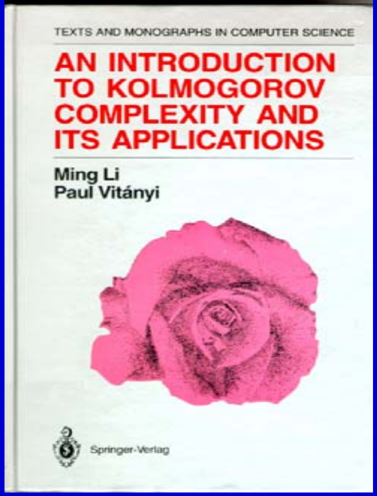
Kolmogorov Complexity

Solomonoff (1960)-Kolmogorov (1965)-Chaitin (1969):
The amount of information in a string is the size of the smallest program generating that string.

$$K(x) = \min_p \{|p| : U(p) = x\}$$

Invariance Theorem: It does not matter which universal Turing machine U we choose. I.e. all “encoding methods” are ok.

Kolmogorov complexity



- $K(x)$ = length of shortest description of x
- $K(x|y)$=length of shortest description of x given y.

A string is (effectively) *random* if it is already compressed to the limit, that is, if $K(x) \geq |x|$.

To erase such a string costs the maximal entropy/energy $|x|\ kT \ln 2$.

- GENERALLY: The ultimate thermodynamic cost of erasing x *is reached by*:

  "Reversibly compress" x to x^*

  Then erase x^*. Cost $\sim K(x)$ bits.

  The longer you compute, the less heat dissipation.

- Thermodynamic cost of converting: $x \leftrightarrow y$

  $E(x,y) = min \{ |p| : U(x,p) = y, U(y,p) = x \}$.

Theorem. $\boldsymbol{E(x,y) = \max\{ K(x|y), K(y|x) \}}$

(Bennett, Gacs, Li, Vitanyi, Zurek STOC'91)

Cost of Erasure

- Trivially, $E(x,\varepsilon) = K(x)$ (i.e., erasing x)

Theorem. $E^t(x, \varepsilon) \geq K^t(x)$

Cost of erasing
x using t time

Length shortest program
Computing x in t time

Question: Is there a hierarchy for growing t?

Energy-Time Tradeoff Hierarchy

- For each large n and $m=\sqrt{n}/2$,

there is an x of length n, and

$t_1(n) < t_2(n) < \ldots t_m(n)$ such that

$$E^{t_1}(x,\varepsilon) > E^{t_2}(x,\varepsilon) > \ldots > E^{t_m}(x,\varepsilon)$$

Note: $t_i(n) = 2^{in}$

General Reversible Simulation

- To do all (a priori unknown) computing tasks reversibly, we need reversibly compile irreversible programs into reversible ones, and then run a reversible computation.
- This implies general reversible simulation of irreversible computations.

Space Hungry (Bennett 1973)

- Irreversible Computation:

Input → Output

Time 0

Time T

- Reversible Simulation:

Instantaneous description = complete computer configuration description

ID

ID'

α

Exists α^{-1}

- Auxiliary Record:

α_1, α_2 , ..., α_T

Algorithm: Input x:

- Reversibly Compute $x \rightarrow f(x)$
- Reversibly Copy f(x)
- Reversibly ``Uncompute'' $x \leftarrow f(x)$

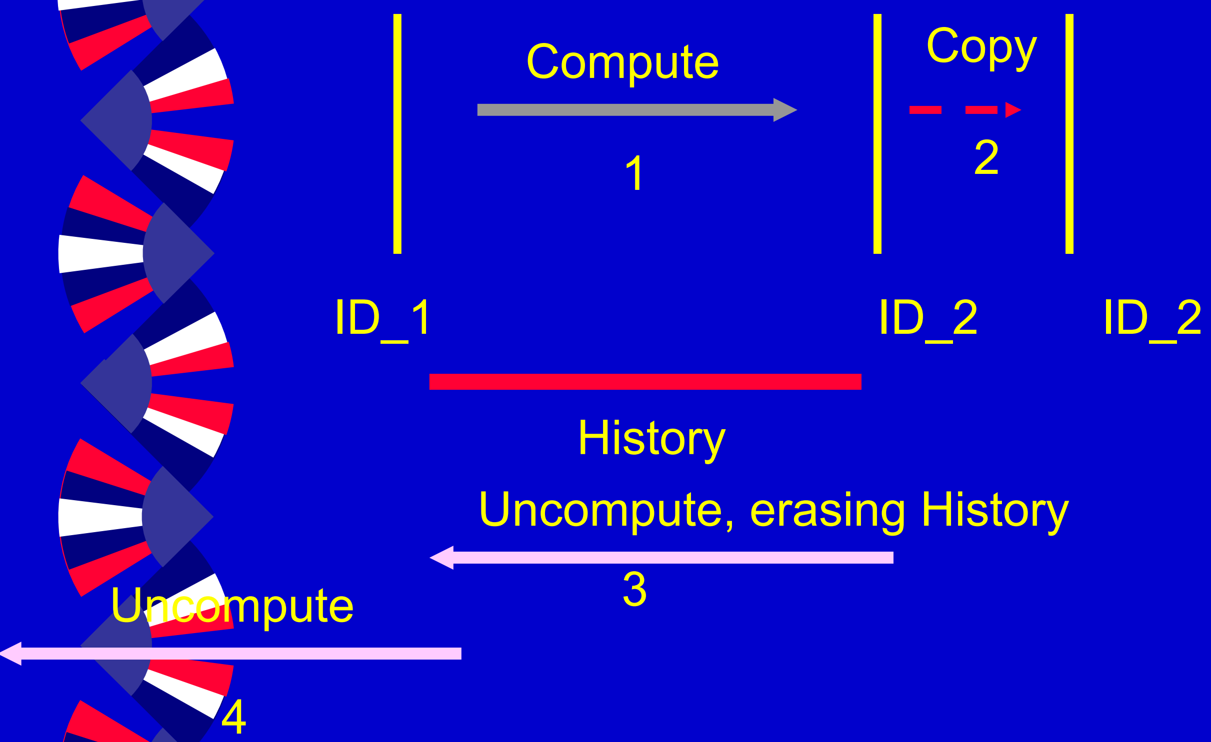
  Erasing $\alpha\_1, \alpha\_2, \ldots, \alpha\_T$ History
- Output <x,f(x)>

Complexity

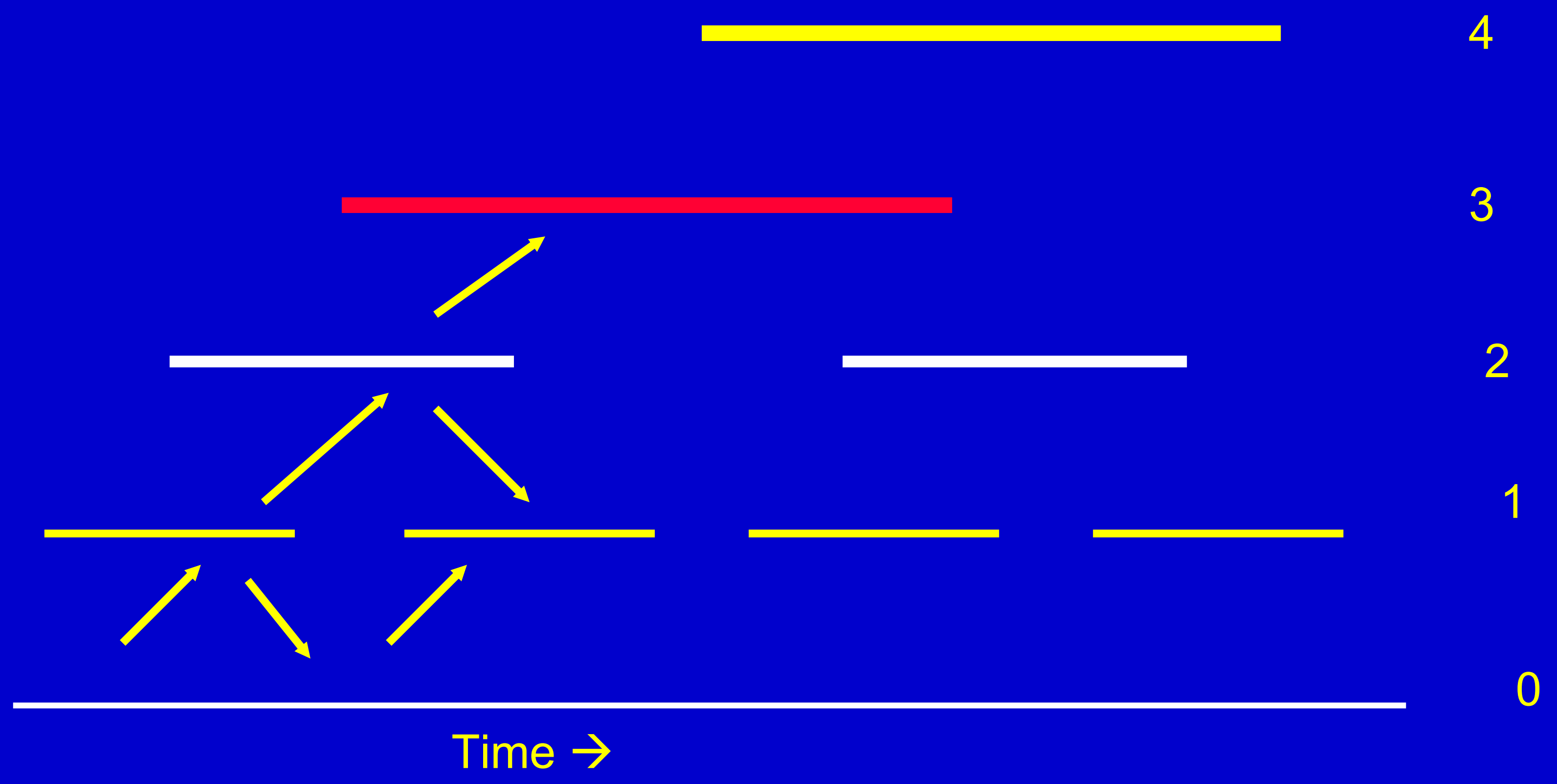
- Each Reversible Computation is 1:1
- Reversible Simulation $x \rightarrow f(x)$ using T time and S space is $x \rightarrow \langle x, f(x) \rangle$ with

$T' = \Theta(T)$ and $S' = \Theta(T+S)$

Space Parsimonious (Bennett 89)



Checkpoint Levels: Can be modelled by ``Reversible Pebble Game''



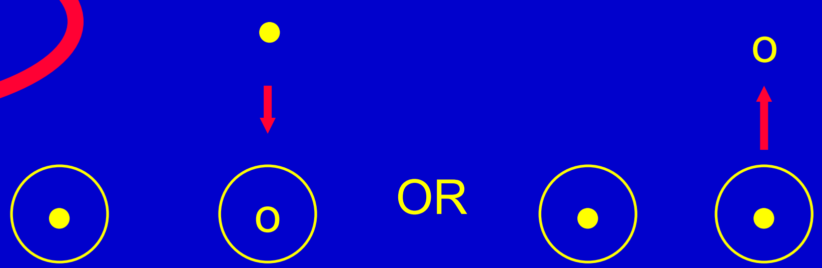
Reversible Pebble Game

- Pebble = ID

1. Initially: (1) (2) . . . (T) Positions

n free pebbles

2. Each step:



3. Win if 1) (T) gets pebbled, and

2) All n pebbles eventually free

Theorem: There is a winning strategy using n pebbles for T=2^n-1

```
1 2 3 4 5 6 7    →   T
o . . . . . .
o o . . . . .
. o . . . . .
. o o . . . .
. o o o . . .
. o  . o . . .
o o  . o . . .
o  .  . o . . .
. . . o o . .
. . . o o o .
. . . o  . o .
. . . o  . o o
. . . o  . o .
. . . o o o .
. . . o o . .
```

T' ↓

n pebbles → $T = 2^n - 1$ in $T' = T^{{}^{2}\mathrm{Log}\,3}$ steps

Theorem (Li, Tromp, Vitanyi 98)

- No reversible pebble game can do better. (Proof is subtle)

We use too much space!!
In the worst case S= (log T)^2 which is pretty bad for large T.

- What if we allow erasures?

Theorem (Space-Erasure LVT 98)

- There is a winning strategy with n+2 pebbles and m-1 erasures for pebble games with T= m2^n for all m ≥ 1 with simulation time T' = 2m^{1 – log 3} T^{log 3}.

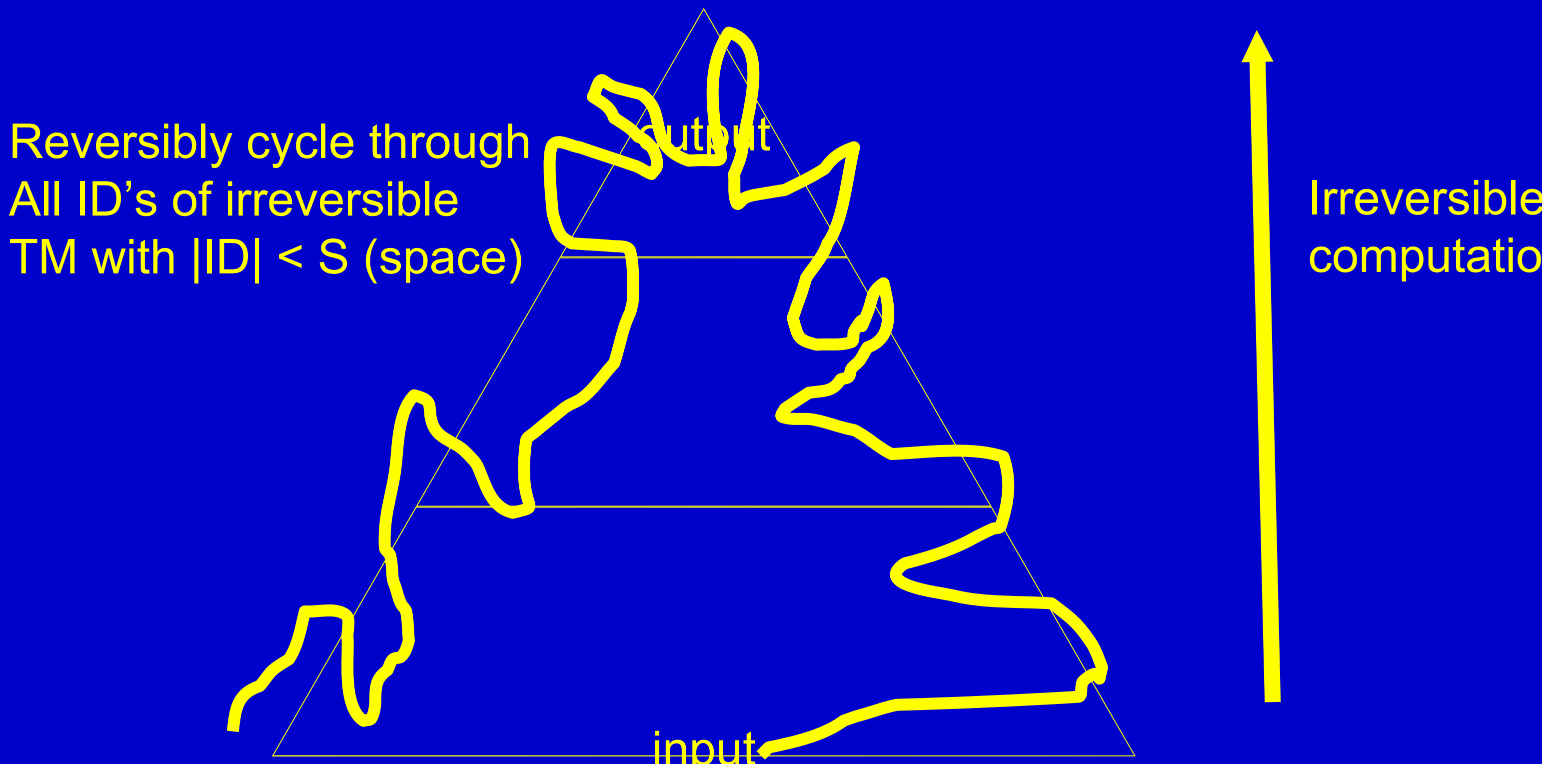
- Corollary: We can use n – log (E-1)
- pebbles and E erasures instead of n pebbles.

Using Less space can only be done by simulations that are not reversible pebble games (LVT98).

- Lange, Mackenzie, Tapp 00:

No extra space but exponential time, by reversible cycling through the configuration tree of the TM computation (connected component of fixed size IDs including the start ID), using a method due to Sipser 90.

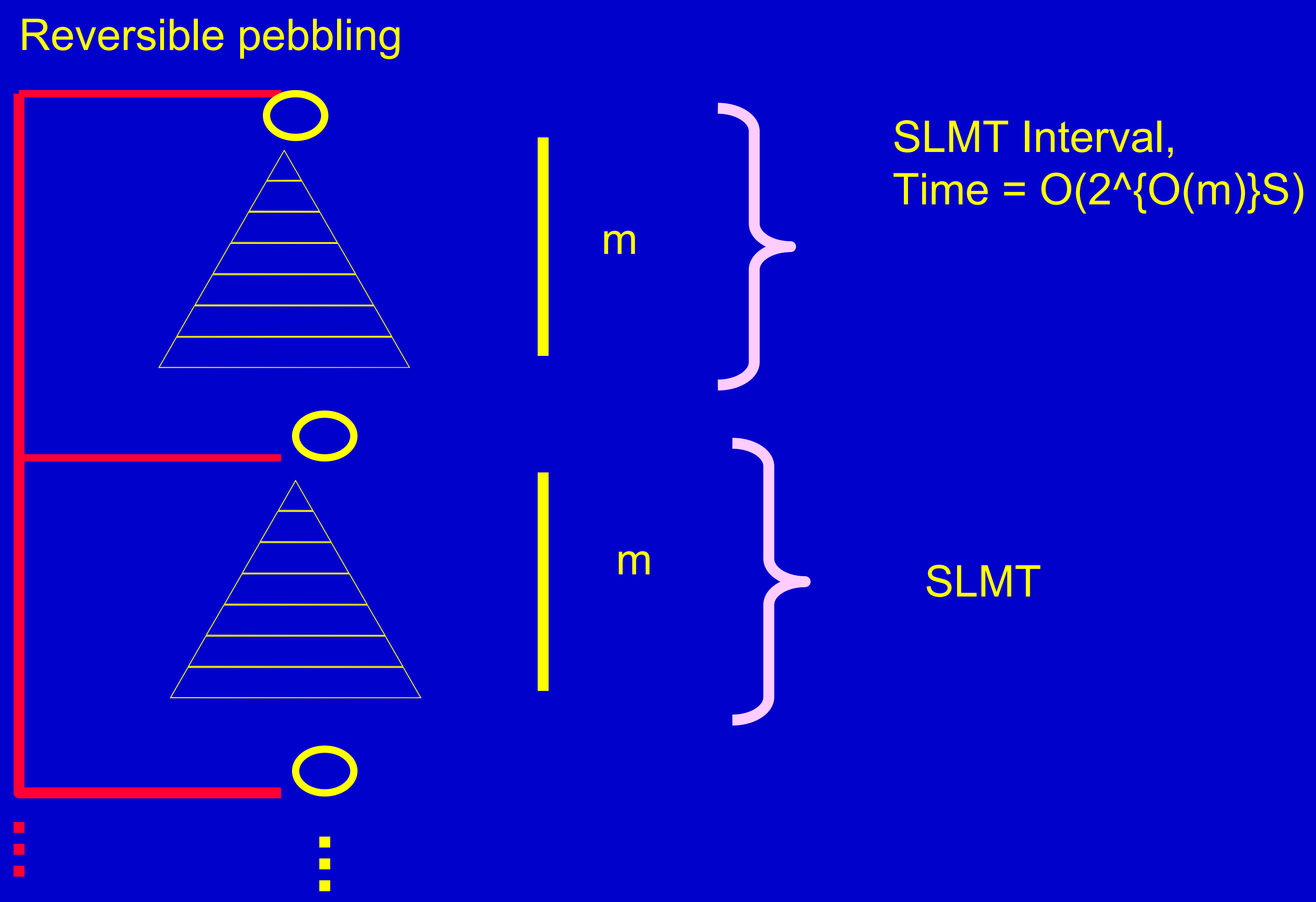
Lemma (LMT00): Reversible simulation time is $O(2^S)$ and reversible simulation space is $S + \log T$



Time-Space Tradeoff, Buhrman Tromp Vitanyi 00,01, Williams 00.

- Reversible Pebbling with long intervals for pebble moves being bridged by LMT simulations.

Algorithm:



Using k pebbles and m=T/2^k

- Lemma: (BTV01) Simulation time is

$$T' = 3^k 2^{O(T/2)^k} S$$

Simulation space is:

S' = S(1+O(k))

Corollary 1:

$$T' = S3^{S'/S}2^{O(T/2^{S'/S})}$$

Some special values:

k=log log T:

$$T' = S(\log T)^{\log 3} 2^{O(T/\log T)}$$

$$S' = S \log\log T \leq S \log S$$

k=√ log T:

$$T' = S\,3^{\sqrt{\log T}}\,2^{O(T/2^{\sqrt{\log T}})}$$

$$S' = S\sqrt{\log T} \leq S\sqrt{S}$$

That is: Both Subexponential Time and Subquadratic Space!

Unknown Computation Time T:

- Simulate first t steps of computation for all values t=2,2^2,2^3,.... Consecutively (and if T<t reversibly undo the simualtion and start with t ← 2t.
- Total time used is T'' ≤ 2T'

$$T'' \leq 2\sum_{t=1}^{\log T} S3^{S'/S}2^{O(2^{t-S'/S})} \leq 2T'$$

Lower Bound (BTV01)

Lemma: If T,S,T',S' are average-case, then every reversible simulation of an Irreversible computation has:

$$T' \geq T \text{ and } S' \geq n + \log T$$

Input length, e.g., n=S

This is sharp, since (S)LMT reaches bound on S' from above.

Big Question:

- Is there a polynomial time * space reversible simulation?

We actually want about linear time And linear space -☺