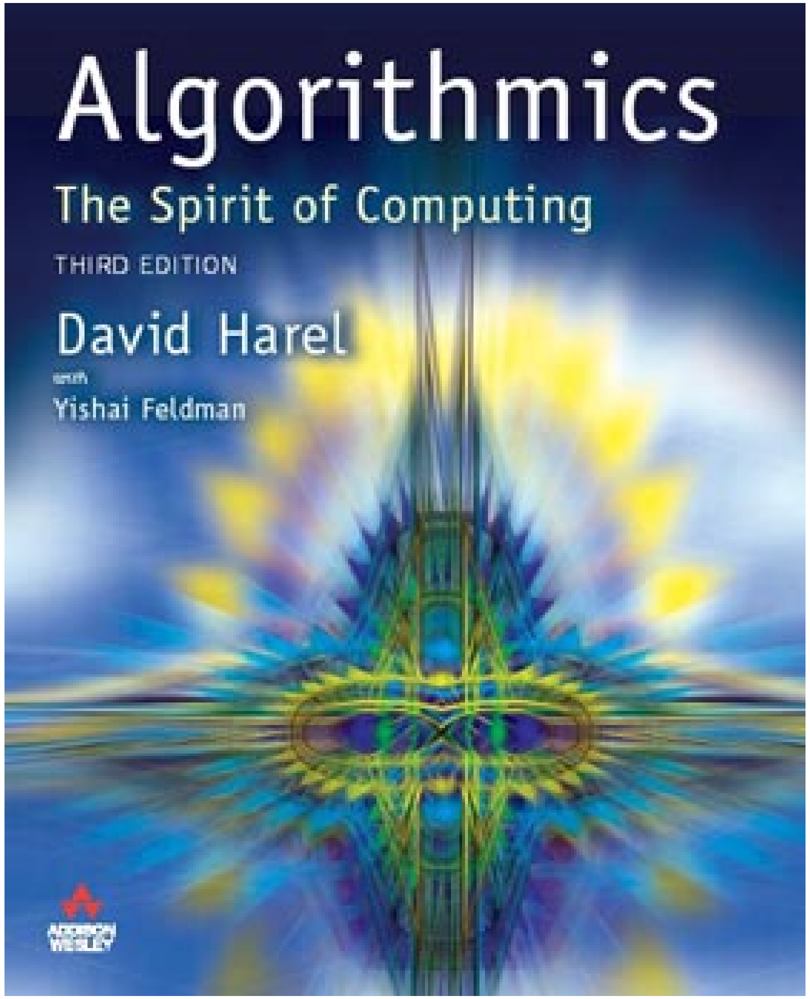
Computers are Not Omnipotent

David Harel

The Weizmann Institute of Science

Rehovot, Israel

For more, see:



1987/2004

2001/2004

TIME *magazine* (April 1984)

"Put the right kind of
software into a computer,
and it will do whatever you
want it to. There may be
limits on what you can do
with the machines themselves,
but there are no limits on
what you can do with software."

Not so !!

Disclaimer:

-Can computers run companies?

-Can computers make good decisions?

-Can computers diagnose?

-Can computers love?

Disclaimer:

~~-Can computers run companies?~~

~~-Can computers make good decisions?~~

~~-Can computers diagnose?~~

~~-Can computers love?~~

* Non-analytic questions

* Pseudo-scientific issues

* Require heuristics

⇓

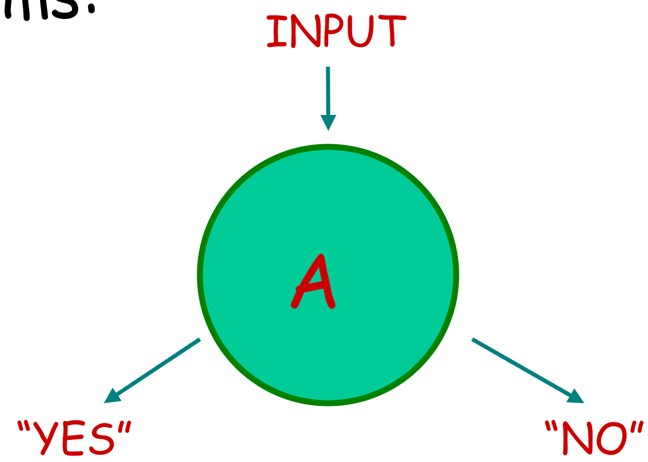
"SOFT" Research

(Artificial Intelligence)

Algorithmic problems:

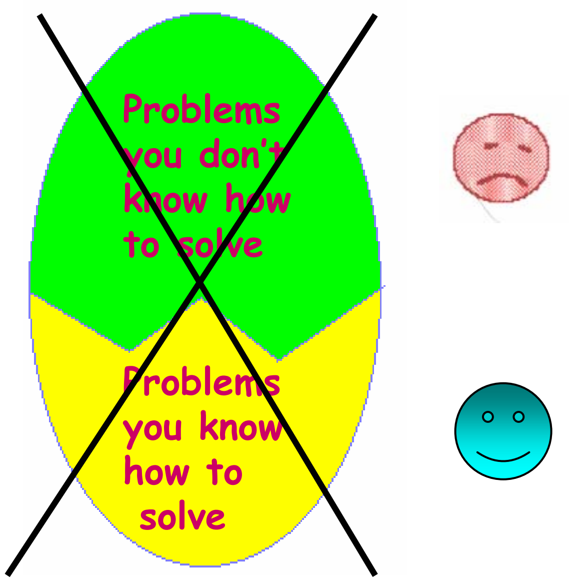
* Set of legal inputs
* Specification of desired output as function of input

Decision problems:

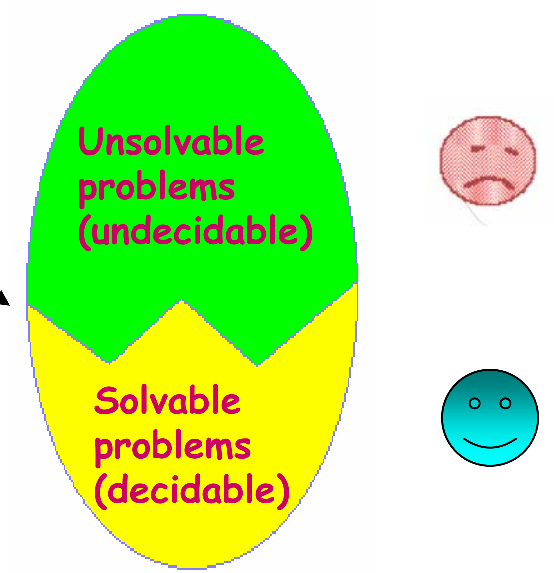


- The algorithm *A* involves "effectively executable" elementary operations, each taking bounded time and bounded resources.

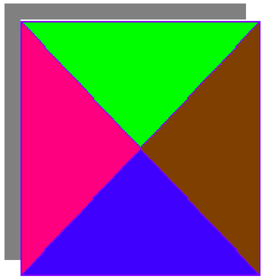
- The algorithm *A* halts for every legal input, and answers the question correctly.



ROBUSTNESS
FOLLOWS
FROM THE
CHURCH/TURING
THESIS, 1936
(All computers are equal…)



Tiling (Domino) Problems:

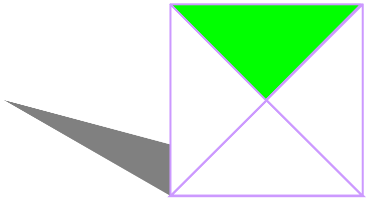


Tile

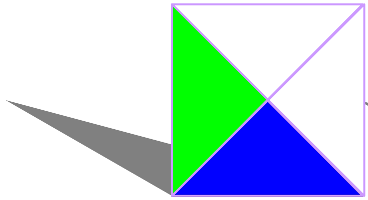
1x1
non-rotatable,
non-reversible

* These problems involve tiling portions of the integer grid Z^2, such that adjacent edges are monochromatic.

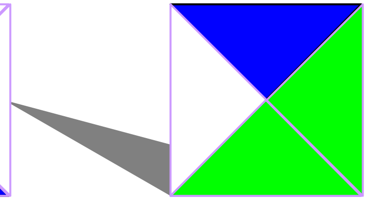
* Inputs include a finite set T of tile types; there are infinitely many copies of each.



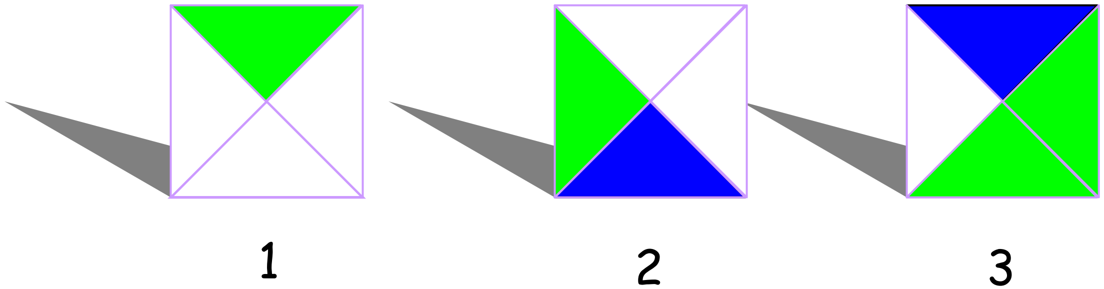
1



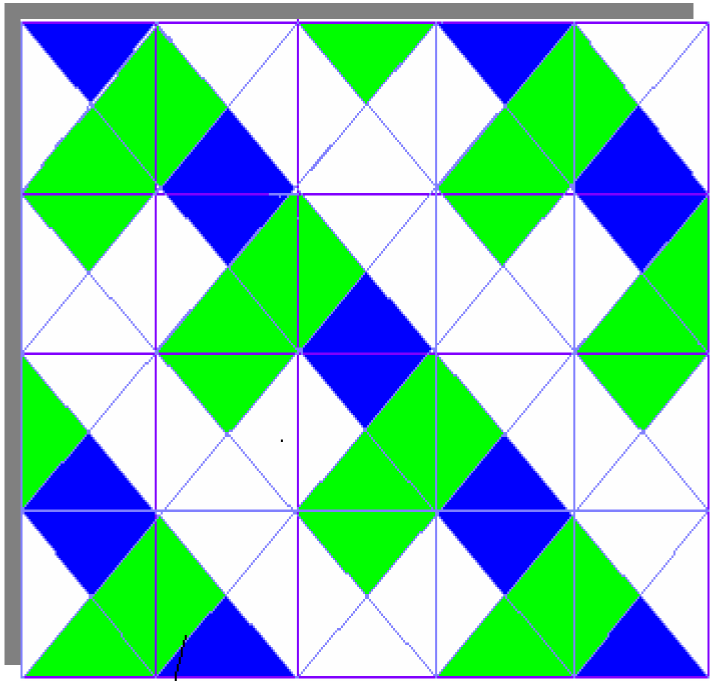
2

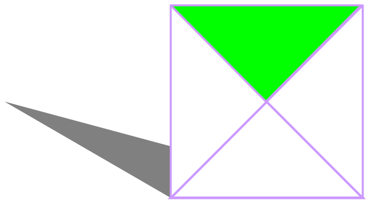


3

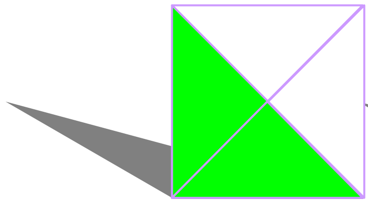


Can tile entire plane Z^2

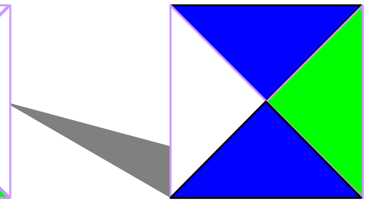




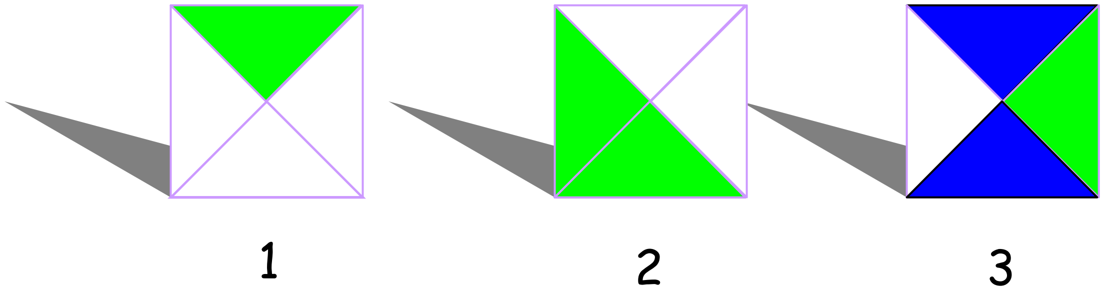
1



2



3

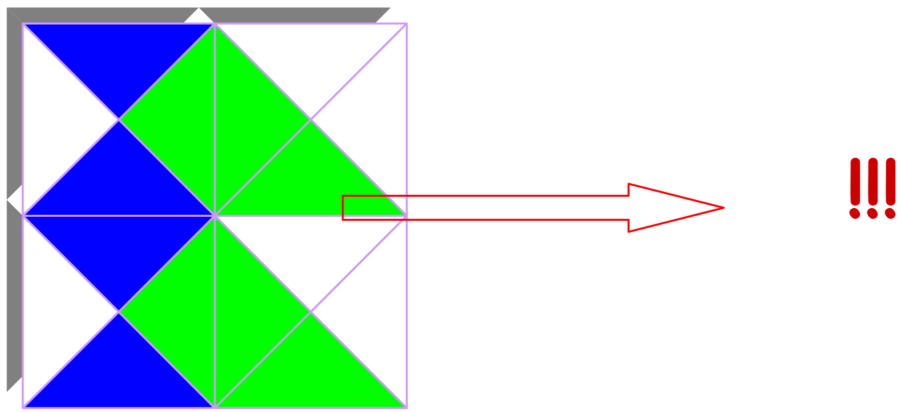


Can't tile even a 3x3 square !

Proof:

1. tile #3 must appear.

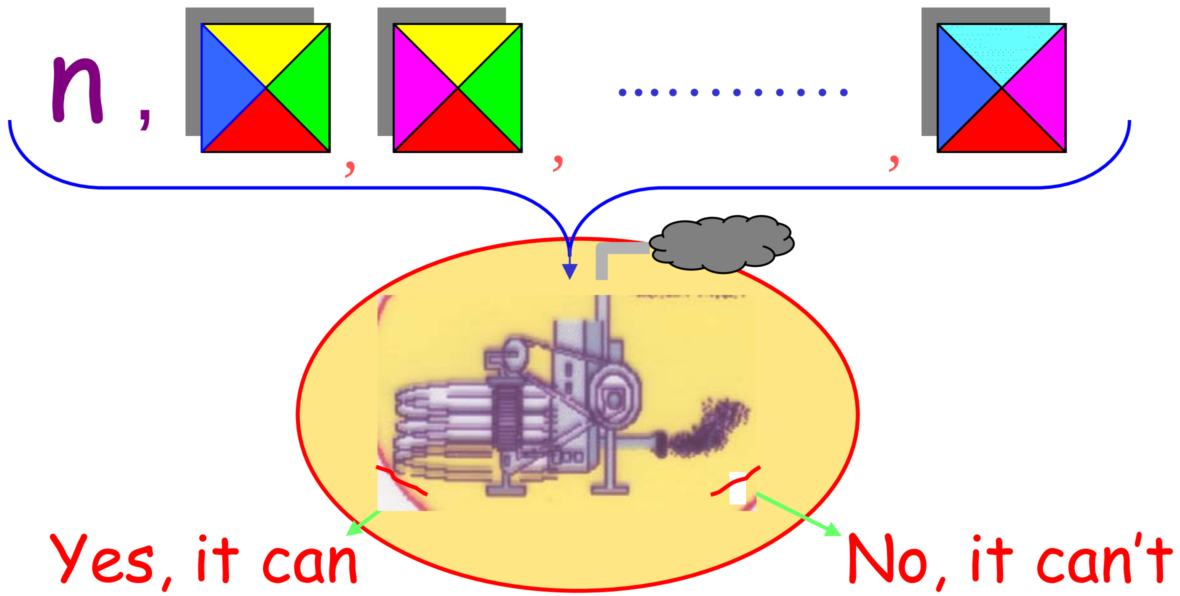
2.



Basic unbounded tiling (domino) problem:

Given T, can T tile Z^2 ?

(equivalent to "can T tile any k x k square")

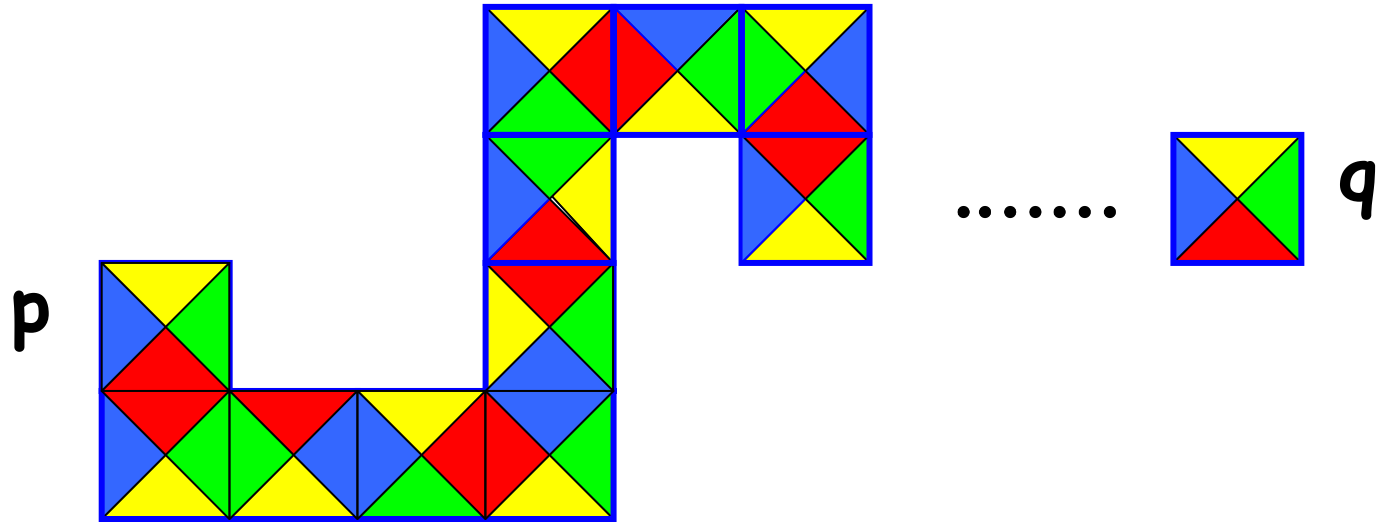


No such machine
No such algorithm
No such recipe
No such program

The domino problem is undecidable !!

Unbounded nature of problem is misleading

Given T and two points, p, q,
can T form "snake" connecting p and q?



UNDECIDABLE in $Z^2/2$ (positive half-plane)

DECIDABLE in Z^2 !!

Undecidable even when removing a single point!!

The halting problem

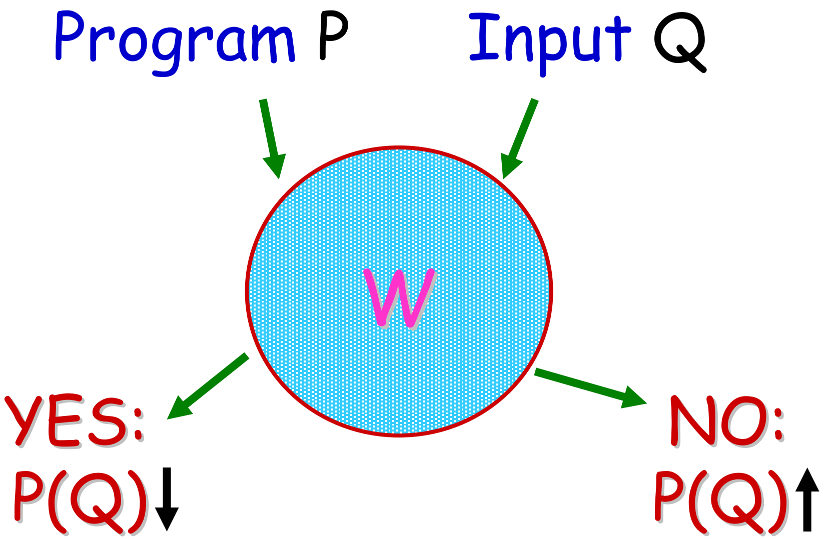
Positive integers

```
A: if x=1 stop;
     x ← x-1;
   goto A.
```

?????

```
B: if x=1 stop
     if x is even then x ← x/2
                  else x ← 3x+1;
   goto B.
```

Is there a W ??



There is no such W

The halting problem is undecidable

And so is essentially <u>any</u> problem about computation, including correctness, efficiency, equivalence, etc.!!

In fact, discovering errors early on is crucial, and can make a *tremendous* difference...

Example: The Y2K Problem

Can we build software to find the errors in
any given program?

Can we correct the errors in programs?

Can we tell whether a program will be efficient?

Can we tell wether two prog's are equivalent?

. . .

. . .

In general, no way!!

Unbounded tiling/domino problem

$Z^2/2$ Snake domino problem

Halting problem

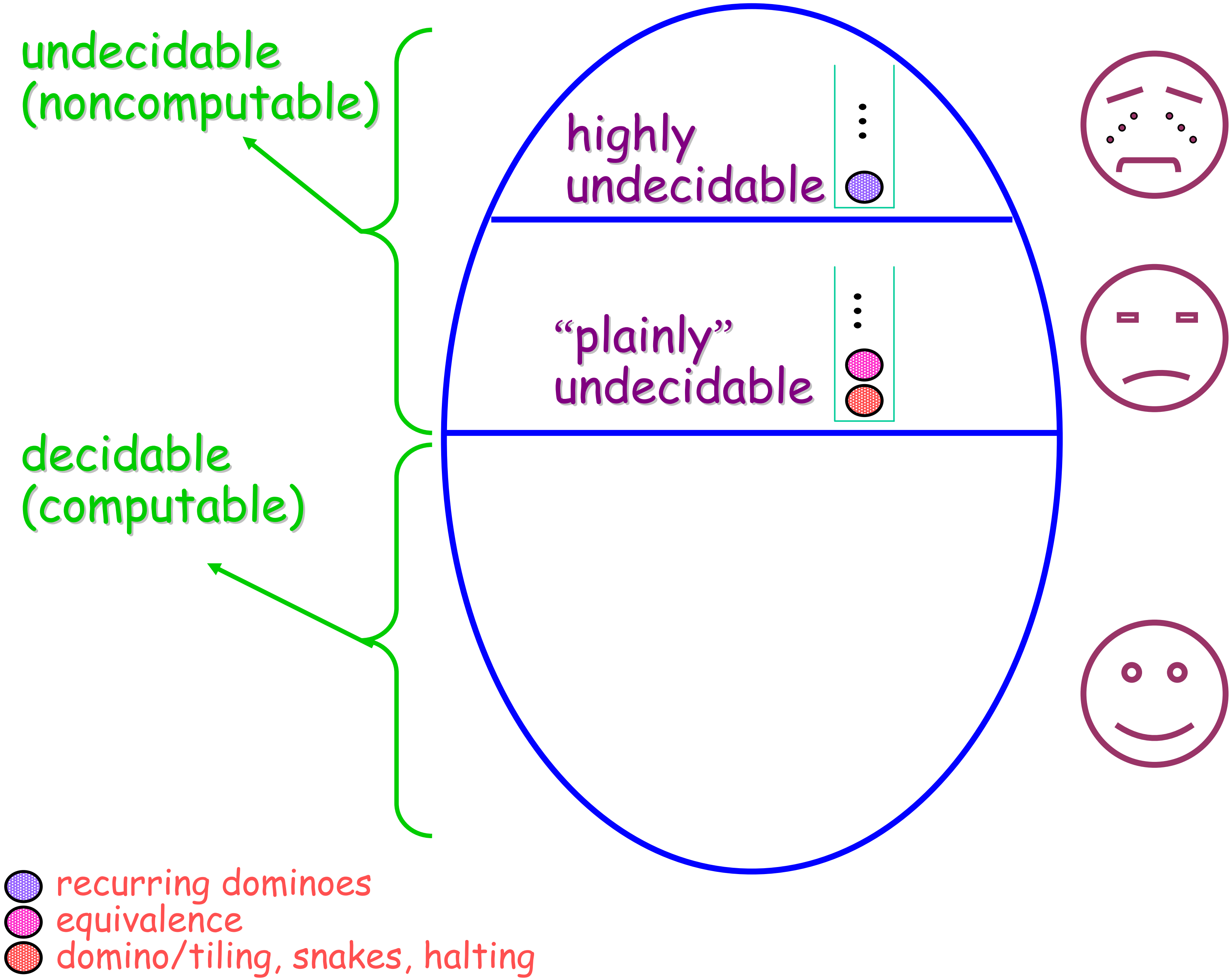
All are computationally equivalent

Reduction: Given one as "free" subroutine, others are decidable

Some problems are much “worse“…

Given T, and d in T, can tile Z^2 such that d occurs infinitely often in the tiling?

unbounded dominos $\underset{\not\longrightarrow}{\longleftarrow}$ recurring dominos


undecidable
(noncomputable)
highly
undecidable
"plainly"
undecidable
decidable
(computable)
recurring dominoes
equivalence
domino/tiling, snakes, halting

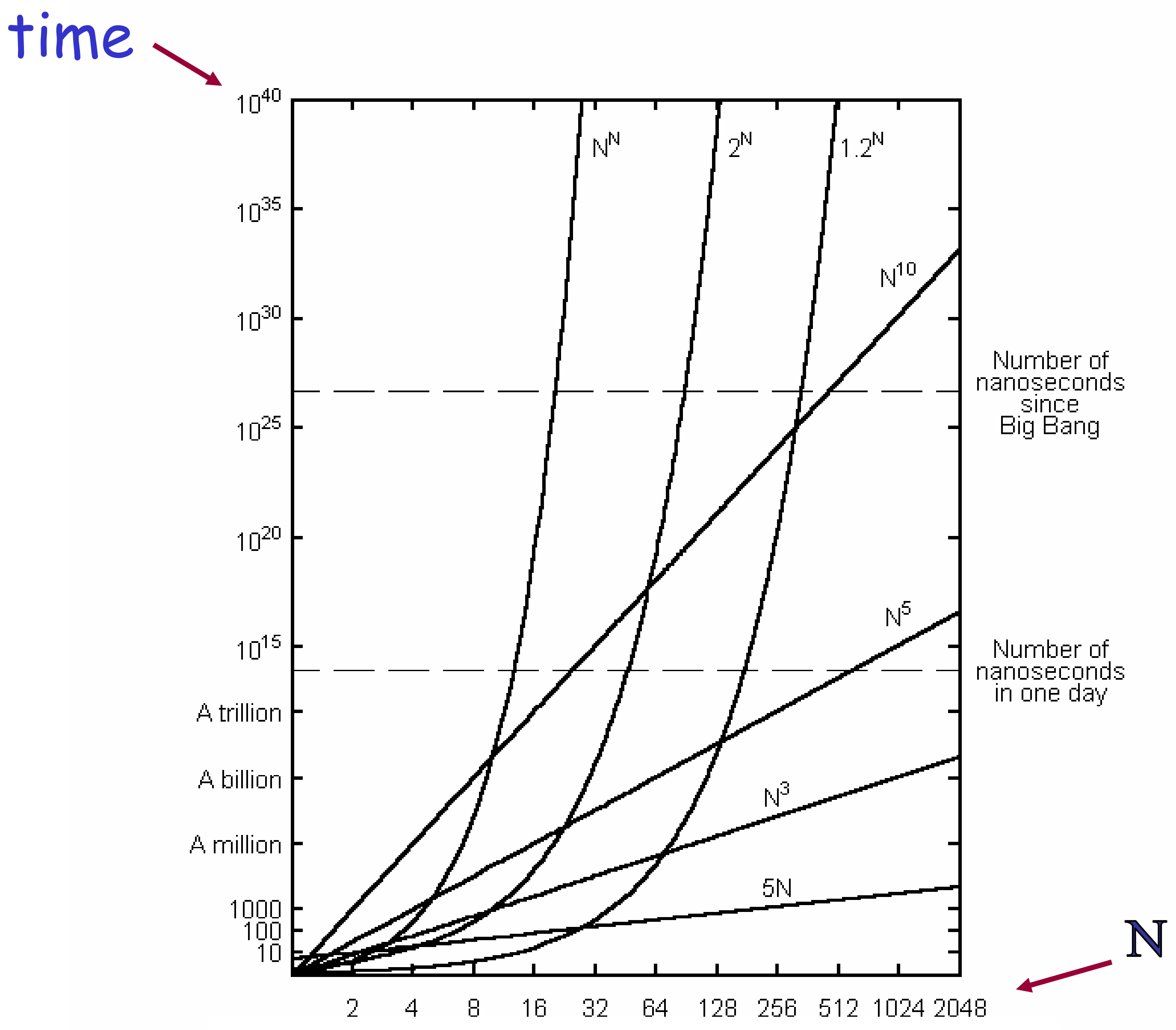
Order of Magnitude Running Time

upper bound		lower bound
	Searching in an unordered list	
$O(n)$		$O(n)$
	Searching in an ordered list	
$O(\log n)$		$O(\log n)$
	Sorting an unordered list	
$O(n \log n)$		$O(n \log n)$
	Multiplying matrices	
$O(n^{2.39\ldots\ldots})$		$O(n^2)$

So, we are talking about amounts, quantities, the size of things...

Sometimes, big differences in size can produce quite striking effects...

Polynomial vs. exponential time



Polynomial time vs. Exponential time algorithms:

assuming 1 instruction per nanosecond

function \ N	20	40	60	100	300
N^2	1/2500 second	1/625 second	1/278 second	1/100 second	1/11 second
N^5	1/300 second	1/10 second	7/10 second	10 seconds	
2^N	1/1000 second				
N^N					

Polynomial time vs. Exponential time algorithms:

assuming 1 instruction per nanosecond

function \ N	20	40	60	100	300
N^2	1/2500 second	1/625 second	1/278 second	1/100 second	1/11 second
N^5	1/300 second	1/10 second	7/10 second	10 seconds	40.5 minutes
2^N	1/1000 second				
N^N					

Polynomial time vs. Exponential time algorithms:

assuming 1 instruction per nanosecond

function \ N	20	40	60	100	300
N^2	1/2500 second	1/625 second	1/278 second	1/100 second	1/11 second
N^5	1/300 second	1/10 second	7/10 second	10 seconds	40.5 minutes
2^N	1/1000 second	18.3 minutes			
N^N					

Polynomial time vs. Exponential time algorithms:

assuming 1 instruction per nanosecond

function \ N	20	40	60	100	300
N^2	1/2500 second	1/625 second	1/278 second	1/100 second	1/11 second
N^5	1/300 second	1/10 second	7/10 second	10 seconds	40.5 minutes
2^N	1/1000 second	18.3 minutes	36.6 years		
N^N					

Polynomial time vs. Exponential time algorithms:

assuming 1 instruction per nanosecond

function \ N	20	40	60	100	300
N^2	1/2500 second	1/625 second	1/278 second	1/100 second	1/11 second
N^5	1/300 second	1/10 second	7/10 second	10 seconds	40.5 minutes
2^N	1/1000 second	18.3 minutes	36.6 years	400 billion centuries	
N^N					

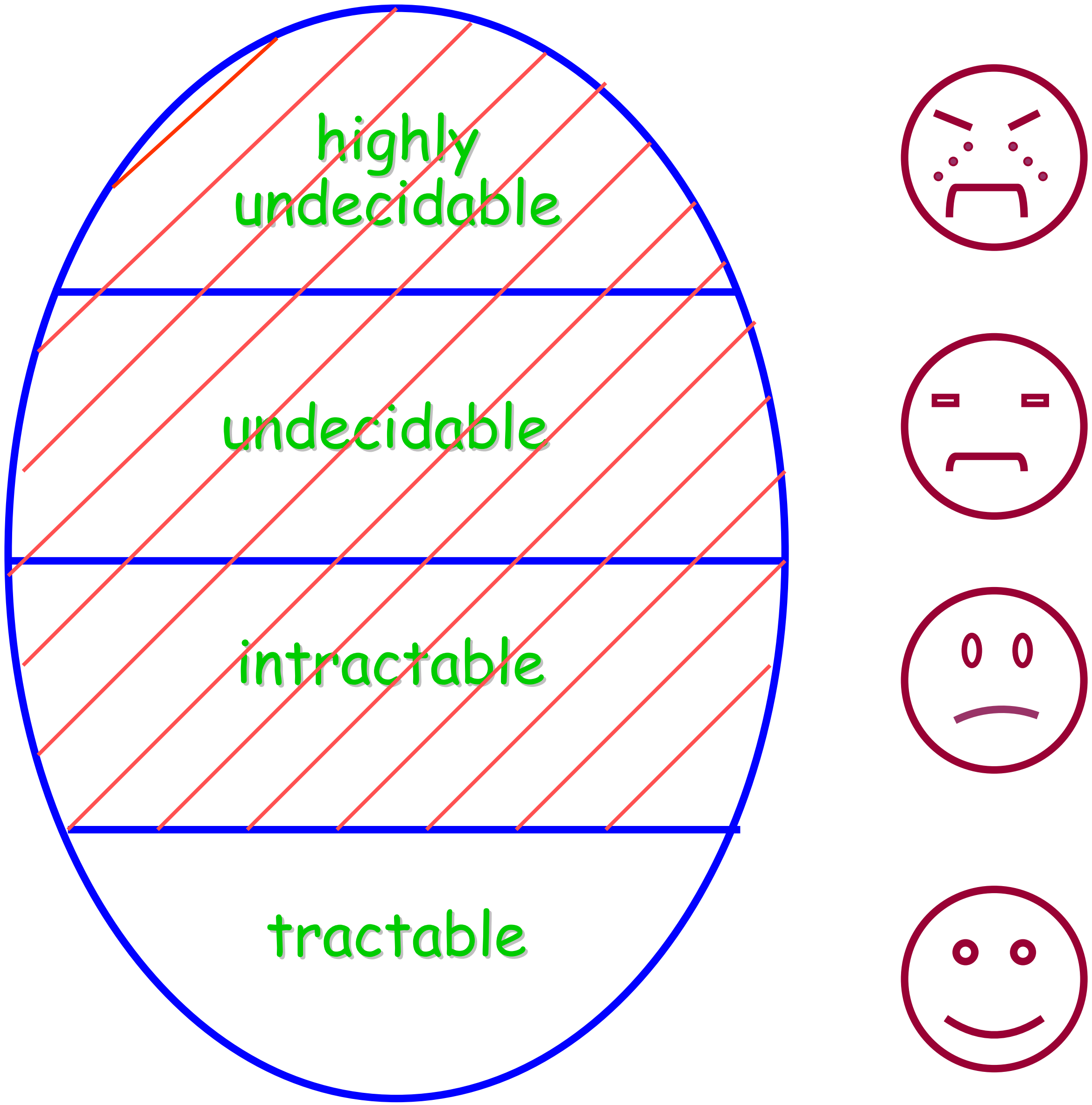
Polynomial time vs. Exponential time algorithms:

assuming 1 instruction per nanosecond

function \ N	20	40	60	100	300
N^2	1/2500 second	1/625 second	1/278 second	1/100 second	1/11 second
N^5	1/300 second	1/10 second	7/10 second	10 seconds	40.5 minutes
2^N	1/1000 second	18.3 minutes	36.6 years	400 billion centuries	a 72-digit number of centuries
N^N					

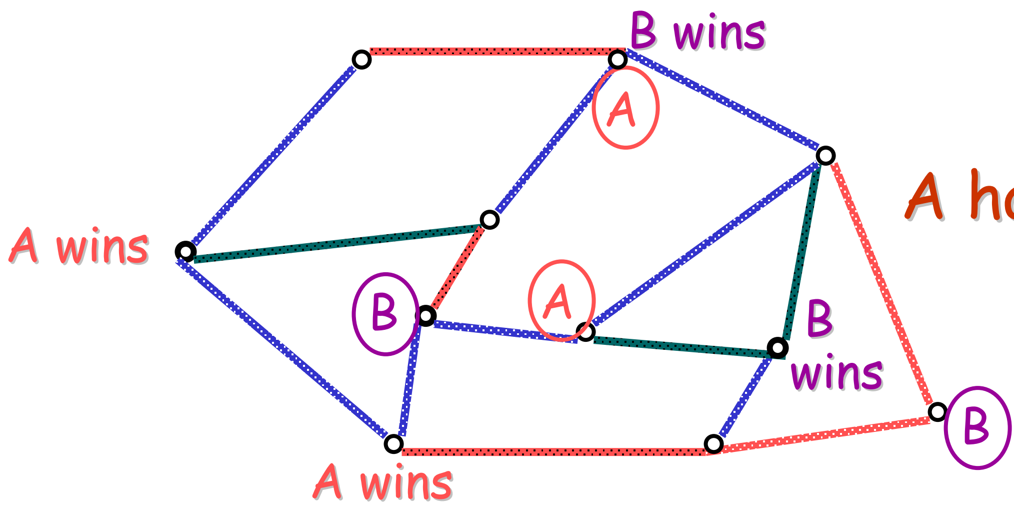
For comparison: the big bang was approximately 15 billion years ago.

function \ N	20	40	60	100	300
N^2	1/2500 second	1/625 second	1/278 second	1/100 second	1/11 second
N^5	1/300 second	1/10 second	7/10 second	10 seconds	40.5 minutes
2^N	1/1000 second	18.3 minutes	36.6 years	400 billion centuries	a 72-digit number of centuries
N^N	3.3 billion years	a 46-digit number of centuries	an 89-digit number of centuries	a 182-digit number of centuries	a 725-digit number of centuries


highly
undecidable
undecidable
intractable
tractable

"Roadblock"

- Two players on a road network;
- "A wins" and "B wins" intersections;
- One turn: player travels in one of his/her cars along any single colored stretch, free of cars.



(A to move)
A has guaranteed win

does Roadblock player have guaranteed win?

does chess player have guaranteed win?

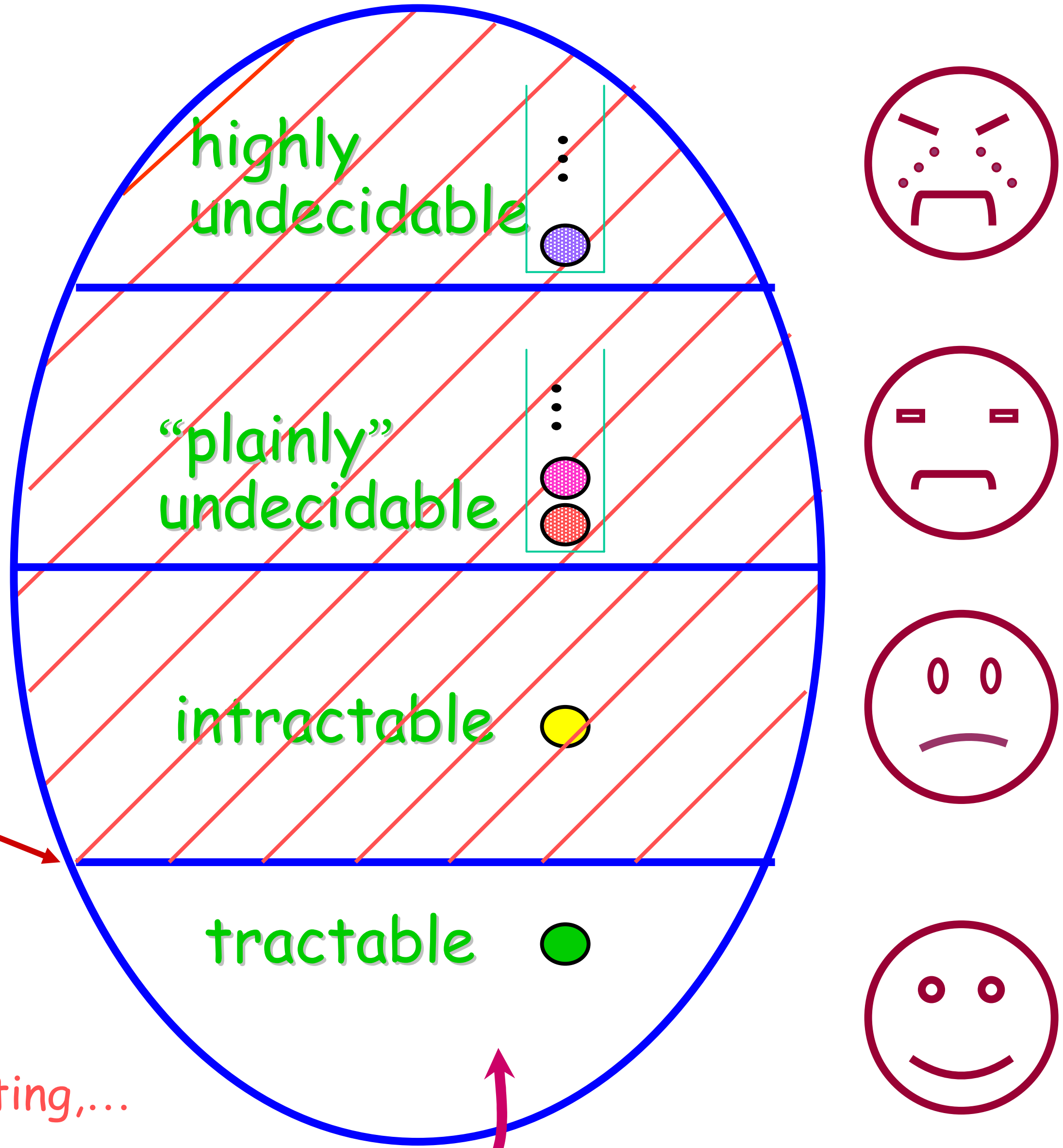
does checkers player have guaranteed win?

generalized to NxN boards

ALL HAVE $2^{O(N)}$ LOWER BOUNDS

Therefore INTRACTABLE !

Robustness follows from refined thesis (All computers are polynomially equal !)



recurring dominoes

equivalence

dominoes, snakes, halting

roadblock, generalized chess

matrix multiplication, sorting,...

primality testing is here too (as of 8/2002) !!

Possible challenge to this robustness:

Quantum computing

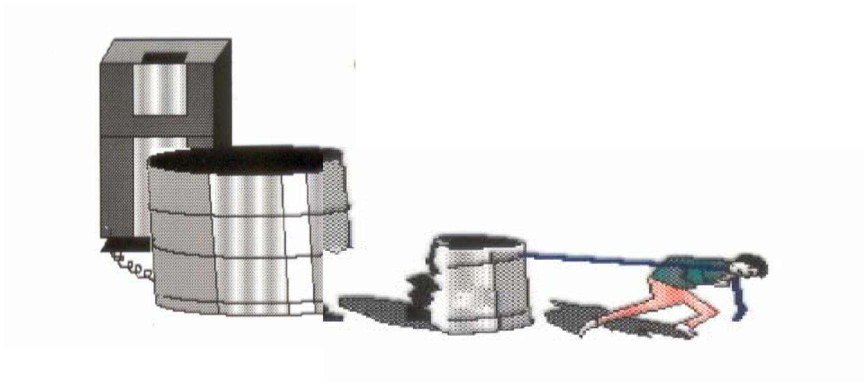
(e.g., there is a polynomial-time algorithm for factoring numbers)

However,

(1) not yet for any provably intractable problem
(2) no practical quantum computer on the horizon yet

Memory (space) requirements

Some problems provably require exponential-space; i.e., even on reasonable inputs (N<150) would require memory larger than the entire known universe, even if each bit were the size of a proton or quark !!!



NP-completeness:
Rising or falling together

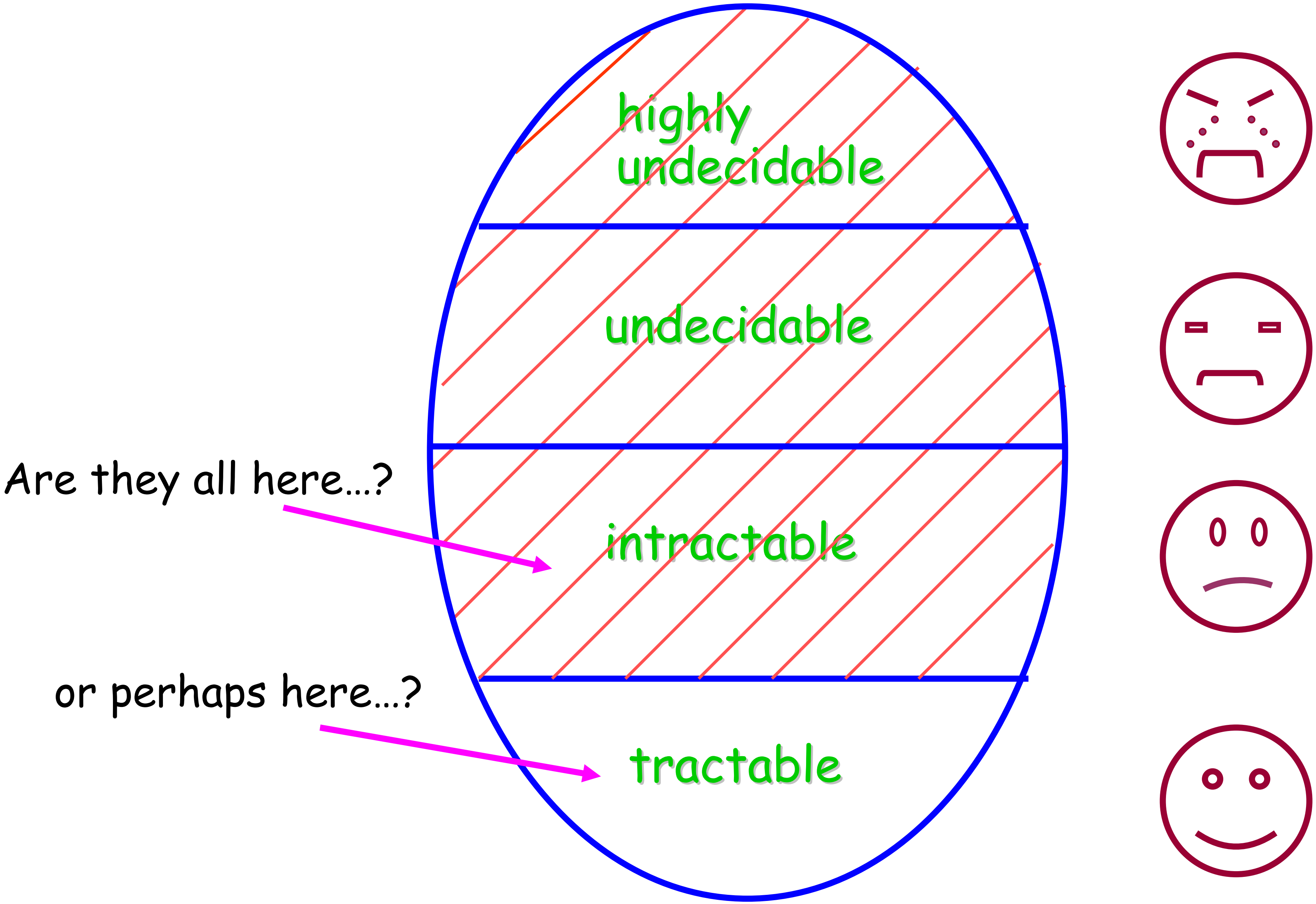
There are 700-3000 problems sharing remarkable properties:

1. Best upper bounds we have are exponential,…
2. but best lower bounds are polynomial.

3. However, if one is tractable we know they <u>all</u> are …
4. and if one is intractable they <u>all</u> are !

$$P \stackrel{?}{=} NP$$ (1971)

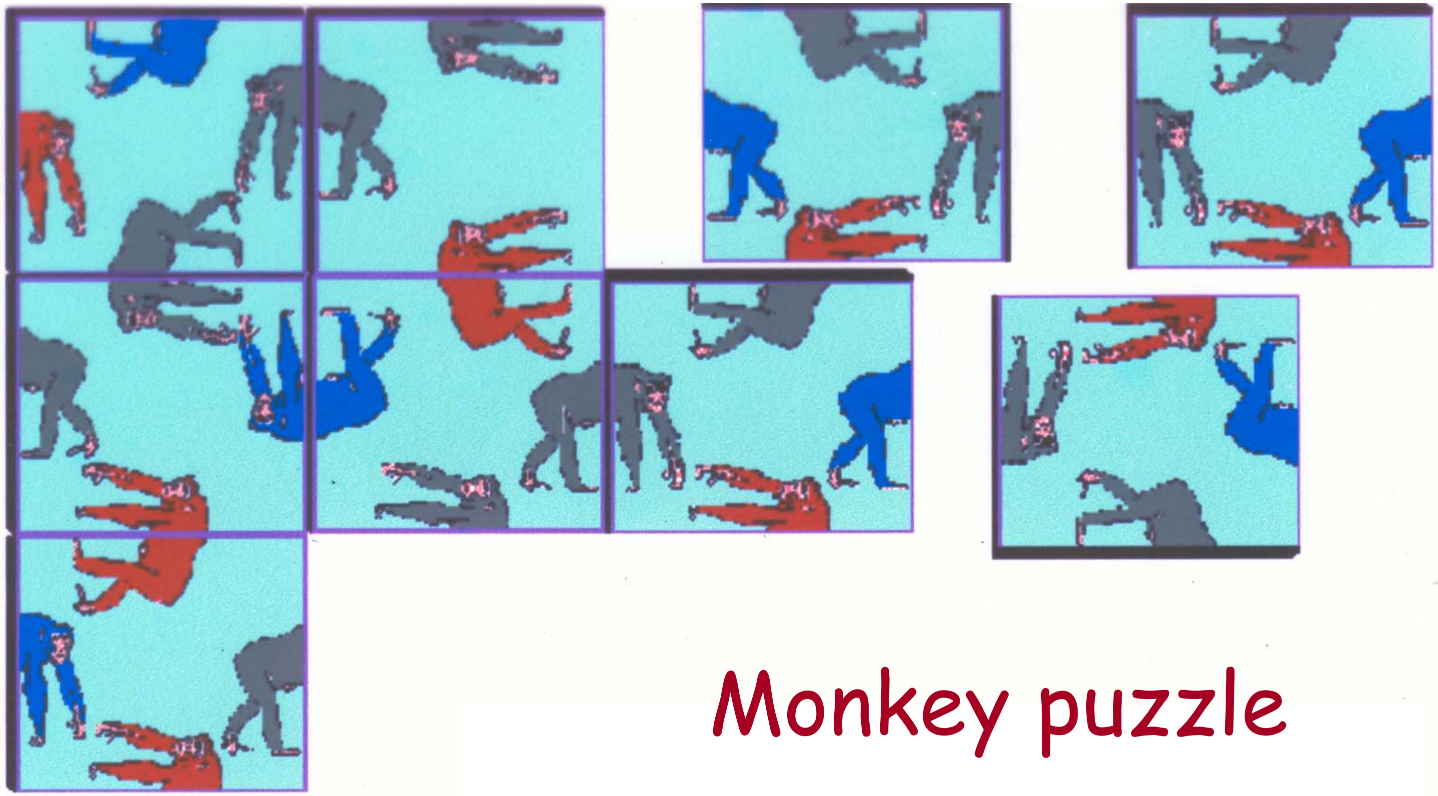
- Most important open problem in computer science
- Already considered major open problem in mathematics



highly undecidable
undecidable
intractable
tractable
Are they all here…?
or perhaps here…?

So, what we are saying here is that an entire skyscraper can depend on a single foundation, right?

Here's what this kind of thing can really look like…



Monkey puzzle

General monkey puzzle (arbitrary N) is NP-complete

(e.g., if N=144, is probably unsolvable in less than millions of years of computer time!)

$\boxed{N=9}$ 3x3

Other NP-complete (i.e., status-unknown) problems:

Timetable problem:

* given N teachers, M time-slots, K courses, teachers' time openings, courses to be taught by who and when; is there a schedule making everyone happy?

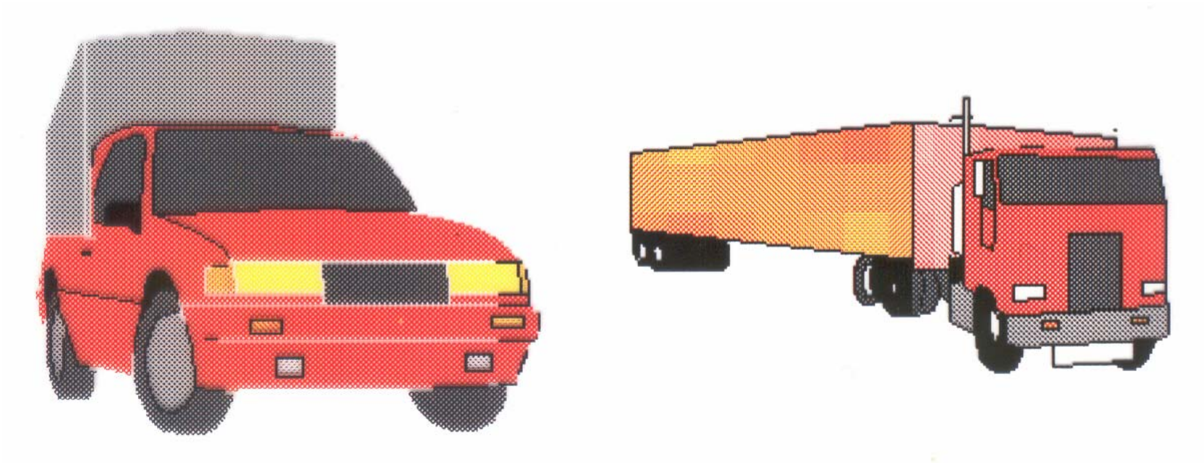
Traveling salesman problem:

* given a distance map with N cities, is there a tour of all cities of length < K ?

* given integers $a_1, \ldots\ldots, a_n$, and K, does some subset subset of the a's sum to K?

* given a formula in propositional logic, is it satisfiable?

* given K trucks with capacities and N objects with weights, can they be packed in the trucks?



Q: Can this be remedied by approximation algorithms; e.g., can we find (fast) a traveling salesman tour guaranteed to be no more than 50% longer than the optimum?

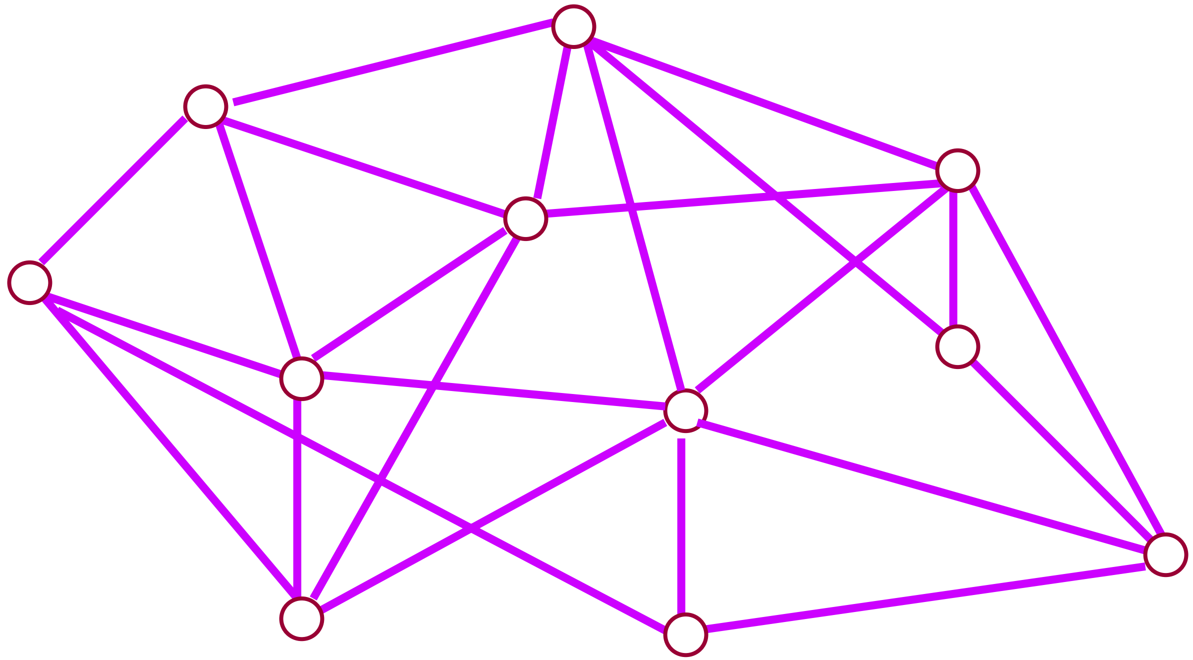
A: Sometimes, but not always.

There are NP-complete problems that have been proven to admit no approximate solutions unless P=NP !

Sometimes the bad news can be used constructively:

In cryptography and security

Zero-knowledge interactive proofs



A: “I can color this graph with 3 colors”

B: “I don’t believe you”

A: “Ok, I’ll prove it”

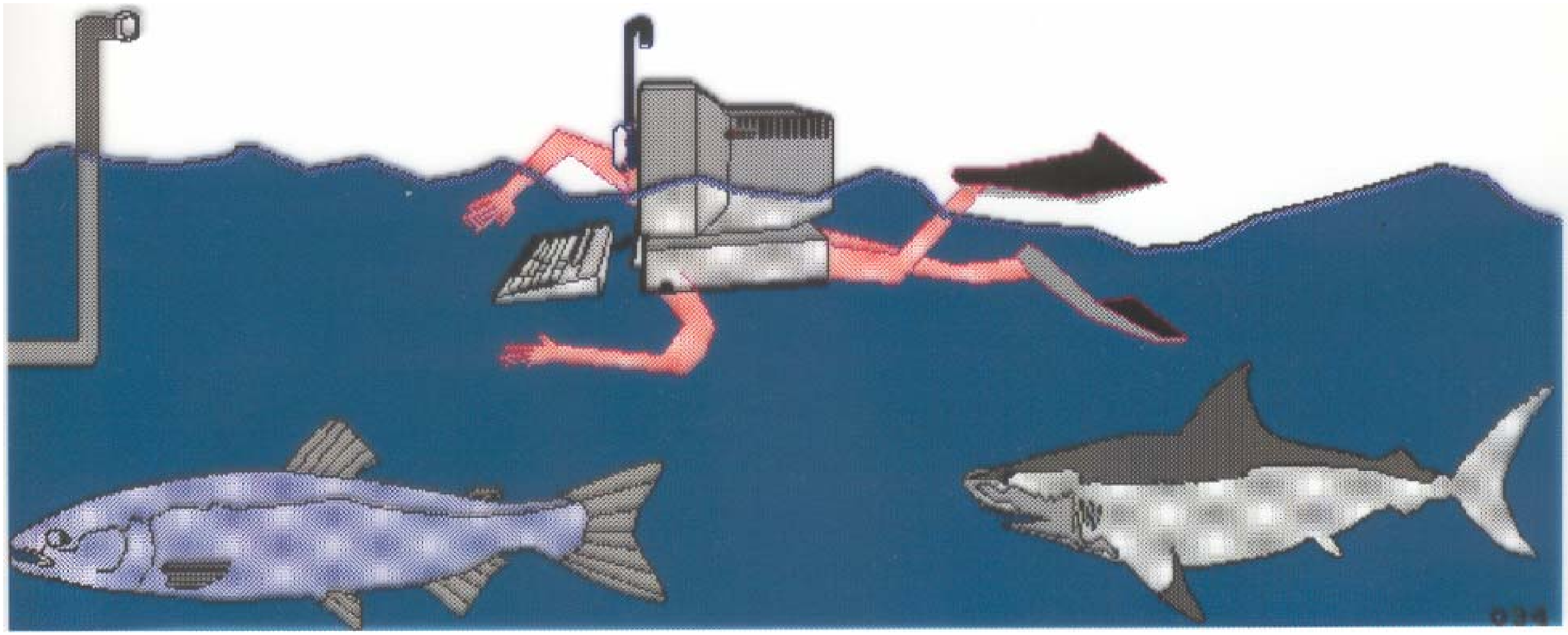
....

B: “Now I believe you, but I have no idea how to do it myself” ??!!?!?!?!

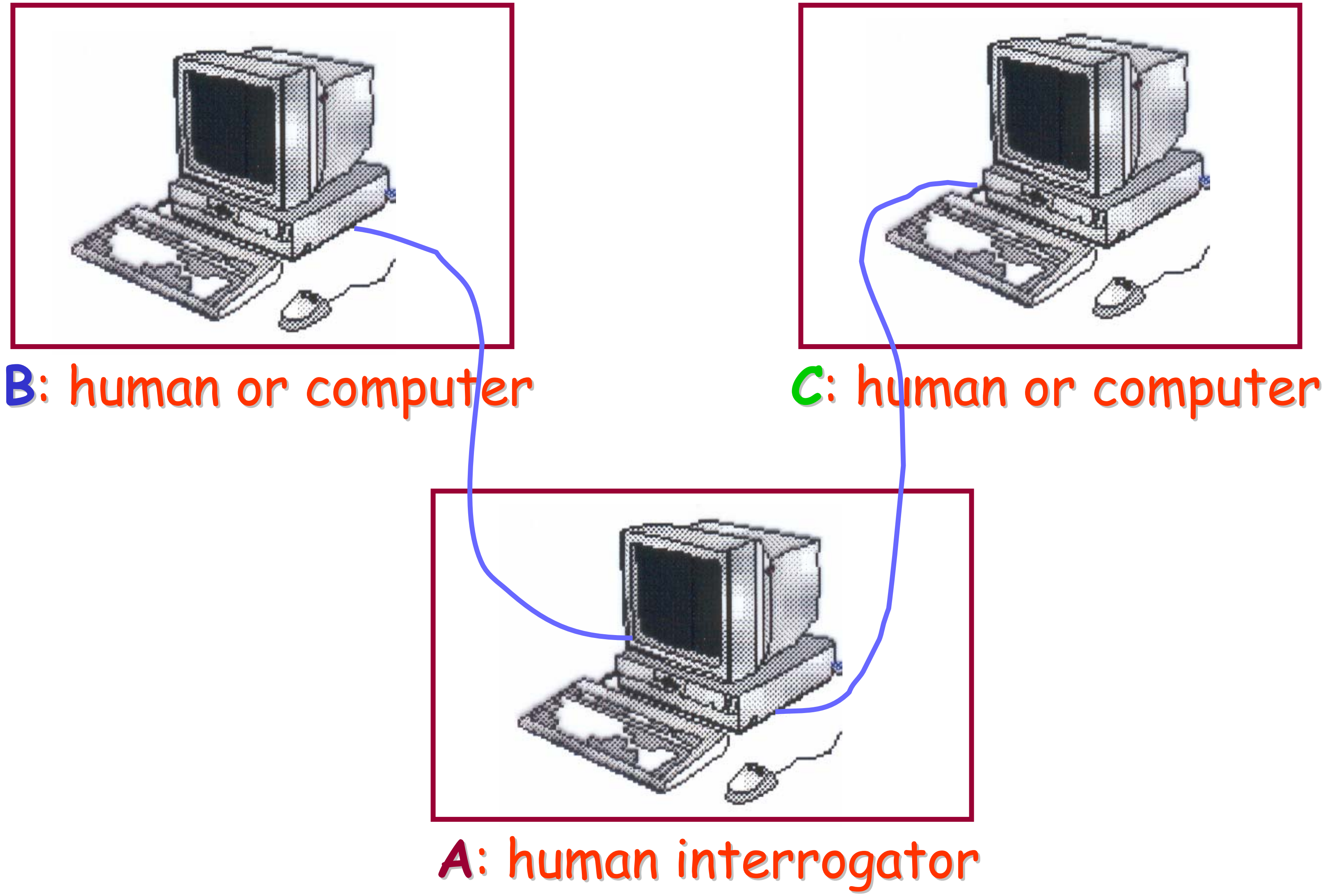
Here come some overhead transparencies...

Computer Science folklore:

"Asking whether computers can think is like asking whether submarines can swim"



The Turing Test



Typical Questions

(1) Are you a computer?

(2) What is the time?

(3) When was president Kennedy assassinated?

(4) What is 2276448 x 7896?

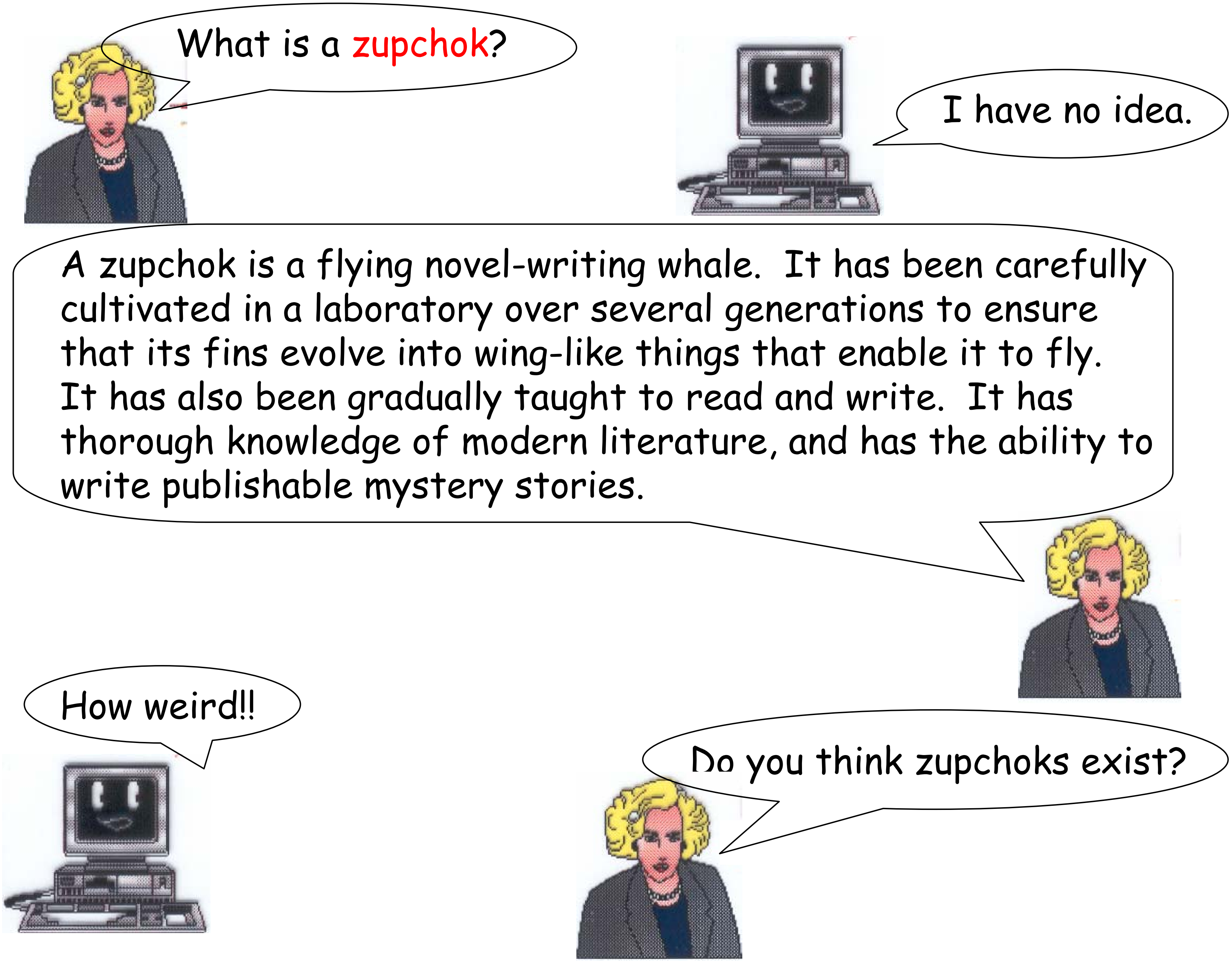
(5) Can white win in one move from following chess position…?

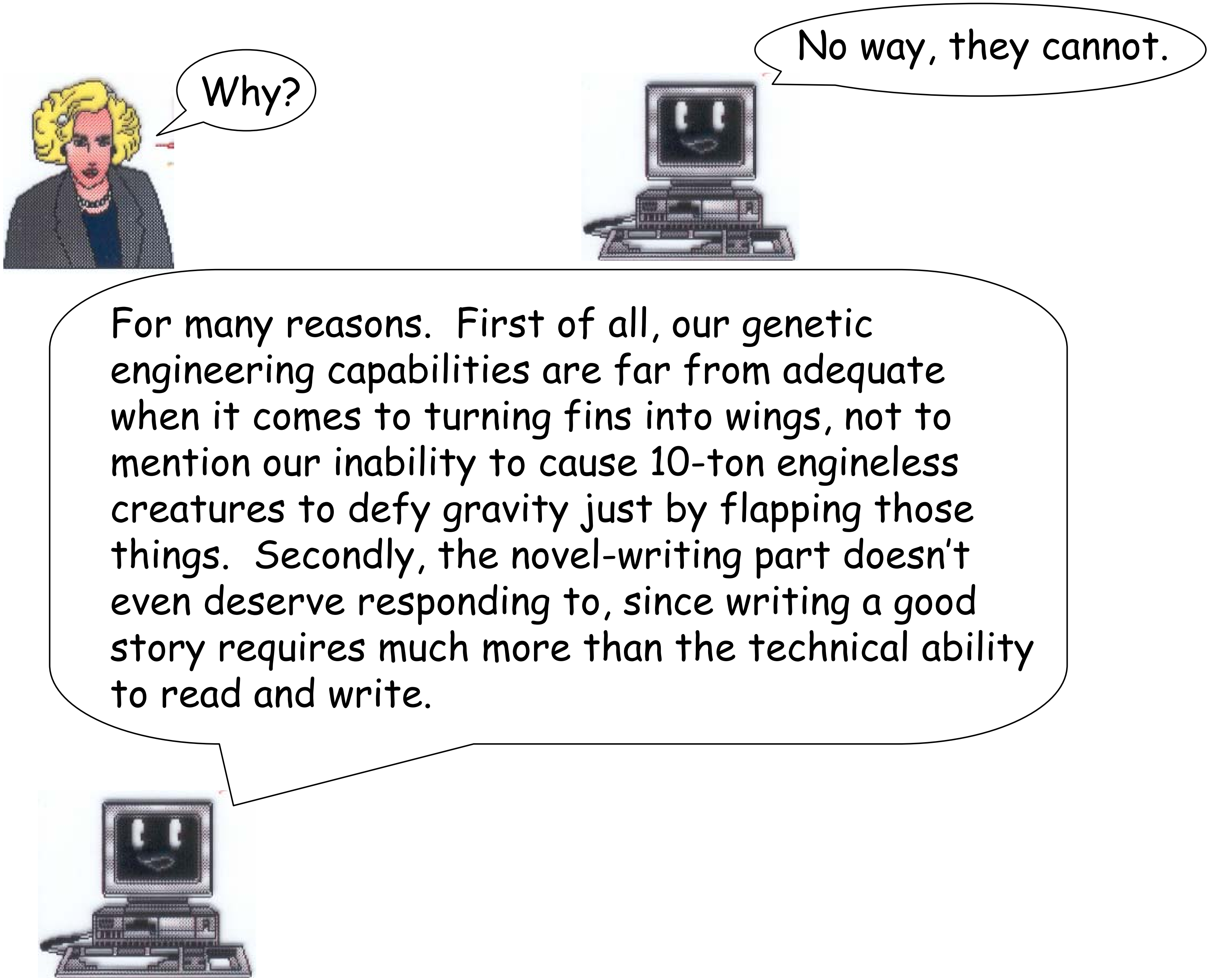
(6) Describe your parents

(7) How does the following poem strike you..?

(8) What do think of Charles Dickens?

(9) What is your opinion on the arms race, in view of the fact that millions of people around the globe suffer from starvation?


What is a zupchok?
I have no idea.
A zupchok is a flying novel-writing whale. It has been carefully cultivated in a laboratory over several generations to ensure that its fins evolve into wing-like things that enable it to fly. It has also been gradually taught to read and write. It has thorough knowledge of modern literature, and has the ability to write publishable mystery stories.
How weird!!
Do you think zupchoks exist?


Why?
No way, they cannot.
For many reasons. First of all, our genetic engineering capabilities are far from adequate when it comes to turning fins into wings, not to mention our inability to cause 10-ton engineless creatures to defy gravity just by flapping those things. Secondly, the novel-writing part doesn't even deserve responding to, since writing a good story requires much more than the technical ability to read and write.

1

Machine: Hello, this is Jim's phone.

Voice: Oh, it's you. Listen, this is his boss. I really need to get Jim right away. Can you locate him and have him call me?

Machine: I'm sorry, Mr.Hizboss, Jim is playing golf this afternoon and left orders not to be disturbed.

Voice: He is, is he? Well look, I'm thin on patience this afternoon. This is HIS BOSS calling, you idiot, not Mr.Hizboss. Get Jim. Now!

Machine: I'm pleased to hear that you are spending time with your patients this afternoon, Dr.Thin. Business must be good. If you want to reach Jim's boss just dial 553-8861. Certainly you would never find him here in Jim's office; we have him listed in our directory under the alias of The Monster.

Voice: Take this message, you son of a chip, and get it straight. Tell him he is not worth the keys on your control panel. He is fired!

(Click)

Machine: Hello, this is Jim's phone.

Voice: Oh, hello, you darling machine. I just wanted to check that we're still on for dinner and whatever.

Machine: Of course, Sue. I have you for Thursday at the usual spot.

Voice: This is Jim's fiancee, Barbara. Who is Sue?

Machine: Oh, Barbara, I didn't recognize your voice. I've never heard of anyone name Sue.

Voice: But you just said he was meeting with Sue on Thursday.

Machine: Oh, THAT Sue. Are you sure you have the right number? This is Robert Finch's phone.

Voice: You can't pull that trick on me. Tell Jim it's all over!!

Machine: You have reached a nonworking number. Please check your listing and redial.

(Click)

Machine: Hello, this is Jim's phone.

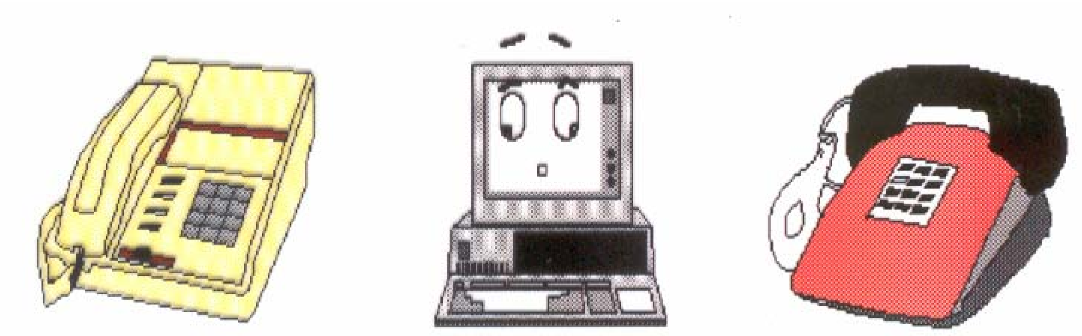
3

Voice: Are you satisfied with your present investments? Have you considered the advantages of tax-free municipal bonds? To hear more, please give your name and address following the beep.

(Beep)

Machine: Err…this is Jim's phone.

Voice: Thank you very much Mr.Jimzfone. Let me tell you more about our unusual investment opportunities…….



And to end this gloomy talk, here is a beautiful example of what computers <u>*can*</u> do:

Thank you for listening