Title: On the concepts of universality in physics and computer science

Date: Apr 10, 2018 09:30 AM

URL: http://pirsa.org/18040107

Abstract: A central fact in computer science is that there are universal machines, that is machines that can run any other program. Recently, a somewhat similar notion of universality has been discovered in physics, by which some spin models can simulate all other models. In this work we shed light on the relation between the two concepts of universality

On the concepts of universality in physics and computer science

Work in progress: mostly thoughts, perspectives, …

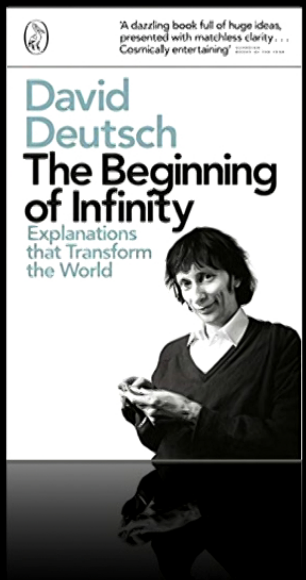
Gemma De las Cuevas (Innsbruck, Austria)

Joint work with Georg Moser

Algorithmic Information, Induction and Observers in Physics. PI, April 10, 2018

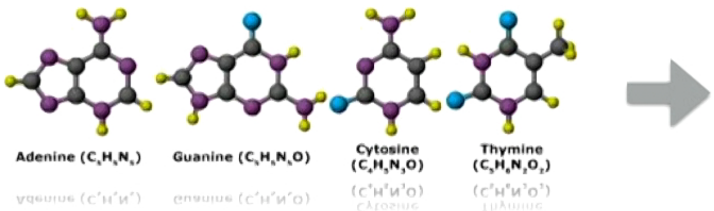
“Our universe is of astonishing simplicity: almost all physical observations can in principle be described by a few theories that have short mathematical descriptions.”

The concept of universality

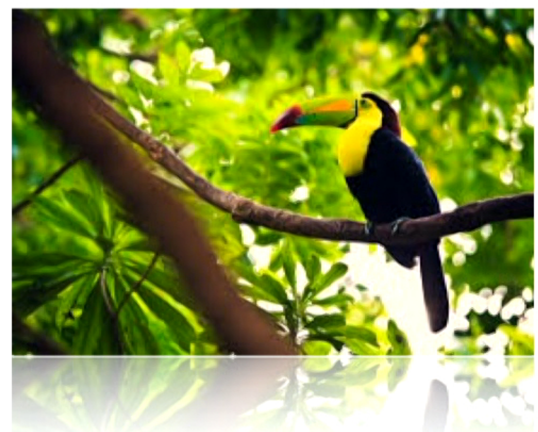


How “easy” it is to generate complexity

Genetic code



Biosphere



How “easy” it is to generate complexity

Alphabet

А Б В Г Д Е
Ё Ж З И Й К
Л М Н О П Р
С Т У Ф Х Ц
Ч Ш Щ Ъ Ы Ь
Э Ю Я

Language


БРАТЬЯ КАРАМАЗОВЫ

РОМАНЪ

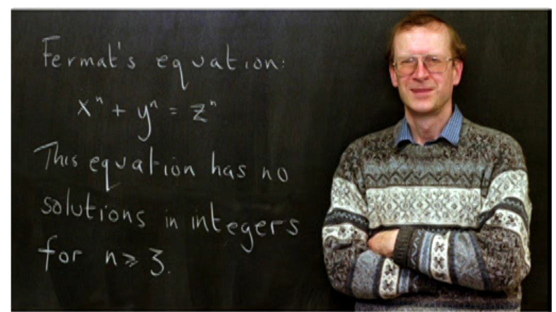
ОТЪ АВТОРА.

How “easy” it is to generate complexity

Axioms of the natural numbers

- 0 is a natural number
- every natural number has a successor
- 0 is not the successor of any natural number
- If the successor of x equals the successor of y, then x equals y
- the axiom of induction: if a statement is true of 0, and if the truth of that statement for a number implies its truth for the successor of that number, then the statement is true for every natural number.

Arithmetic



The jump to universality

When systems gradually become more complex,

they suddenly undergo a large change of functionality.

The jump to universality

When systems gradually become more complex,

they suddenly undergo a large change of functionality.

They can suddenly can capture all possible complexity.

The jump to universality

When systems gradually become more complex,

they suddenly undergo a large change of functionality.

They can suddenly can capture all possible complexity.

They become *universal.*

The jump to universality

When systems gradually become more complex,

they suddenly undergo a large change of functionality.

They can suddenly can capture all possible complexity.

They become *universal.*

They are at the beginning of infinity.

Universality in writing

- Writing systems based on sounds: Universal

А Б В Г Д Е
Ё Ж З И Й К
Л М Н О П Р
С Т У Ф Х Ц
Ч Ш Щ Ъ Ы Ь
Э Ю Я

БРАТЬЯ КАРАМАЗОВЫ

РОМАНЪ

ОТЪ АВТОРА.

Universality in writing

- Writing systems based on sounds: Universal

А Б В Г Д Е
Ё Ж З И Й К
Л М Н О П Р
С Т У Ф Х Ц
Ч Ш Щ Ъ Ы Ь
Э Ю Я


БРАТЬЯ КАРАМАЗОВЫ

РОМАНЪ

ОТЪ АВТОРА.

capable of representing every possible word with a finite amount of symbols

Universality in writing

- Writing systems based on pictograms: Not universal

Universality in writing

- Writing systems based on pictograms: Not universal

Universality in writing

- Writing systems based on pictograms: Not universal

Imagine you want to write TREASON but have no pictogram for it:

Universality in writing

- Writing systems based on pictograms: Not universal

Imagine you want to write TREASON but have no pictogram for it:

- You can invent a new pictogram.

Universality in writing

- Writing systems based on pictograms: Not universal

Imagine you want to write TREASON but have no pictogram for it:

- You can invent a new pictogram.
- Or you can invent a rule:

Use the pictograms for **T**ent **R**ock **EA**gle **Z**ebra **N**ose

This rule triggers the jump to universality.

Universality in numeral systems

- Arabic numeral system: Universal

"2" can take infinitely many meanings: 2

20

2000

….

Universality in numeral systems

- Arabic numeral system: Universal

"2" can take infinitely many meanings: 2

20

2000

….

capable of representing every possible number with a finite amount of symbols (ten)

Universality in numeral systems

- Roman numeral system: Not universal

Universality in numeral systems

- Roman numeral system: Not universal

  For numbers larger than a thousand, MMMMM….. Like tally marks

Universality in numeral systems

- Roman numeral system: Not universal

  For numbers larger than a thousand, MMMMM….. Like tally marks

  Imagine you want to write one million:

Universality in numeral systems

- Roman numeral system: Not universal

  For numbers larger than a thousand, MMMMM….. Like tally marks

  Imagine you want to write one million:

  - You could invent a new symbol

Universality in numeral systems

- Roman numeral system: Not universal

  For numbers larger than a thousand, MMMMM….. Like tally marks

  Imagine you want to write one million:

  - You could invent a new symbol
  - Or invent a new rule that triggers the jump to universality

Universality in acquisition of knowledge

- Critical mind: Universal

Universality in acquisition of knowledge

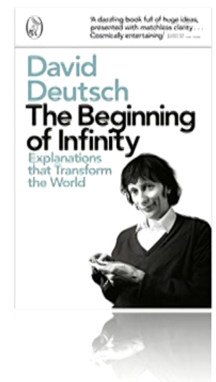
- Critical mind: Universal

"Problems can be solved" ➡ Open-ended generation of knowledge

Universality in acquisition of knowledge

- Critical mind: Universal

“Problems can be solved” → Open-ended generation of knowledge

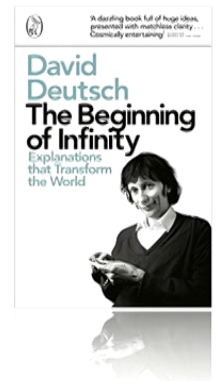


People become universal explainers.

Universality in acquisition of knowledge

- Critical mind: Universal

"Problems can be solved" → Open-ended generation of knowledge



People become universal explainers.

- Knowledge given by authority (e.g. religion): Not universal.

The jump to universality

Some lessons

- It requires very little complexity of the underlying elements, but the correct "rules".

The jump to universality

Some lessons

- It requires very little complexity of the underlying elements,

  but the correct "rules".

- It can only happen in digital systems

  since error correction is only possible in digital systems

  (in analogue systems every value can be correct)

The jump to universality

Some lessons

- It requires very little complexity of the underlying elements,

  but the correct “rules”.

- It can only happen in digital systems

  since error correction is only possible in digital systems

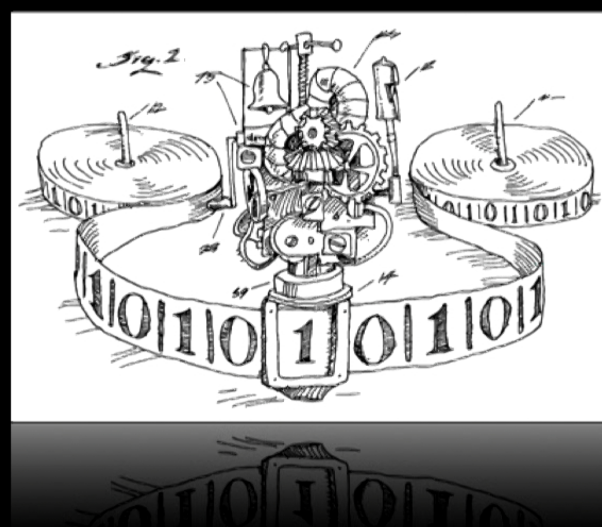
  (in analogue systems every value can be correct)

  The writing system a, b, c, …

  The numeral system 0, 1, 2 , …

  The genetic code A G T C

Universality in computer science



Universality in computer science

In the early days of computer science….

- Formalization of the notion of computation:

  Turing machines, Post systems,

  Recursive functions, lambda calculus, …

Universality in computer science

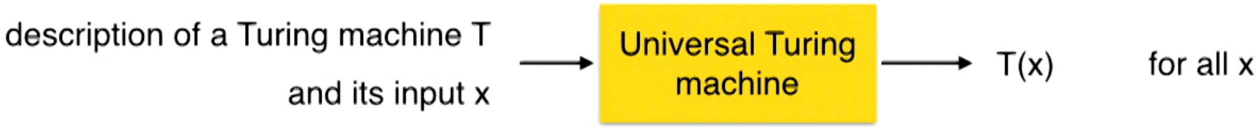
In the early days of computer science….

- Formalization of the notion of computation:

  Turing machines, Post systems,

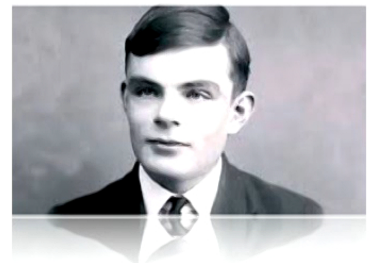
  Recursive functions, lambda calculus, …

- Certain machines are “universal”:



Universality in computer science

In the early days of computer science….



- Formalization of the notion of computation:

  Turing machines, Post systems,

  Recursive functions, lambda calculus, …

- Certain machines are "universal":

description of a Turing machine T and its input x → **Universal Turing machine** → T(x) for all x

it has finitely many rules, yet it can run any possible computation

Universality in computer science

A program can be interpreted as data, and thus fed to another program.

Universality in computer science

A program can be interpreted as data, and thus fed to another program.

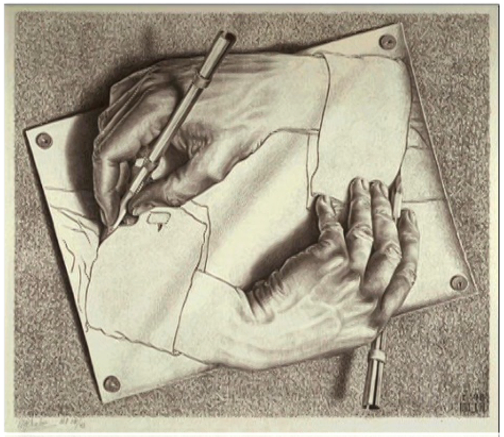
Two concepts which were different (data and programs) are put at the same level

Universality in computer science

A program can be interpreted as data, and thus fed to another program.

Two concepts which were different (data and programs) are put at the same level

➡ self-reference



Undecidability

Self-reference and negation

“I am a liar”

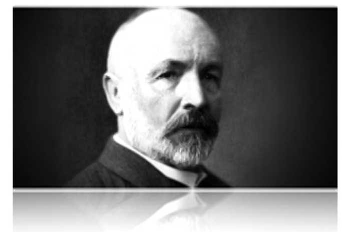
Undecidability

Self-reference and negation

“I am a liar”

Cantor’s diagonal argument:

	1	2	3	4	5	….
1st set	0	1	1	1	1	….
2nd set	1	0	0	1	1	….
3rd set	0	0	0	0	1	….
4th set	1	1	1	1	1	….
…						



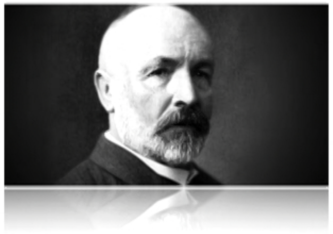
Undecidability

Self-reference and negation

“I am a liar”

Cantor’s diagonal argument:

	1	2	3	4	5	....
1st set	0	1	1	1	1	....
2nd set	1	0	0	1	1	....
3rd set	0	0	0	0	1	....
4th set	1	1	1	1	1	....
…						



→ The natural numbers and its power set cannot be put in one-to-one correspondence.

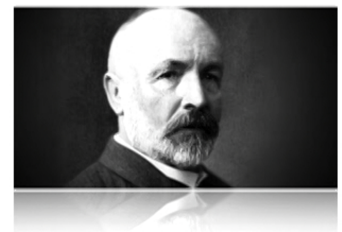
Undecidability

Self-reference and negation

“I am a liar”

Cantor’s diagonal argument:

	1	2	3	4	5	....
1st set	0	1	1	1	1	....
2nd set	1	0	0	1	1	....
3rd set	0	0	0	0	1	....
4th set	1	1	1	1	1	....
…						



➡ The natural numbers and its power set cannot be put in one-to-one correspondence.

Gödel’s proof is also a diagonal argument.

Undecidability

Some lessons

- Crucial that the set is infinite

Undecidability

Some lessons

- Crucial that the set is infinite

- Against reductionism:

  this is a "fundamental truth" that appears at a higher level

  it happens whenever the system is capable of doing self-reference

Undecidability in maths

Undecidability in maths

- Almost every function is uncomputable:

  Set of Turing machines $\mathbb{N}$

  Set of functions $f : \mathbb{N} \to \mathbb{N}$

  Thus, most functions are uncomputable.

Undecidability in maths

- Almost every function is uncomputable:

  Set of Turing machines $\mathbb{N}$

  Set of functions $f : \mathbb{N} \to \mathbb{N}$

  Thus, most functions are uncomputable.

But most of these functions are irrelevant…

Undecidability in maths

- Almost every function is uncomputable:

  Set of Turing machines $\mathbb{N}$

  Set of functions $f : \mathbb{N} \to \mathbb{N}$

  Thus, most functions are uncomputable.

  But most of these functions are irrelevant…

- Many "easy" problems are undecidable:

  Take 5 3x3 matrices with integer entries: $M_1 = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$ ….. $M_5 = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$

  Is there a product of them which gives the 0 matrix? $M_1 M_2 M_2 M_1 M_5 \ldots$

Undecidability in maths

- Almost every function is uncomputable:

  Set of Turing machines $\mathbb{N}$

  Set of functions $f : \mathbb{N} \to \mathbb{N}$

  Thus, most functions are uncomputable.

  But most of these functions are irrelevant…

- Many "easy" problems are undecidable:

  Take 5 3x3 matrices with integer entries: $M_1 = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$ ….. $M_5 = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$

  Is there a product of them which gives the 0 matrix? $M_1 M_2 M_2 M_1 M_5 \ldots$

- By making reductions, these problems spread everywhere

Undecidability in computer science

Undecidability in computer science

- A language is a set of strings (over a finite alphabet).

Undecidability in computer science

- A language is a set of strings (over a finite alphabet).
- Given a string x, is it in the language L?

Undecidability in computer science

- A language is a set of strings (over a finite alphabet).
- Given a string x, is it in the language L?

You can pose this question to a machine:

Undecidability in computer science

- A language is a set of strings (over a finite alphabet).
- Given a string x, is it in the language L?

You can pose this question to a machine:

"The machine M recognizes language L"

Undecidability in computer science

- A language is a set of strings (over a finite alphabet).
- Given a string x, is it in the language L?

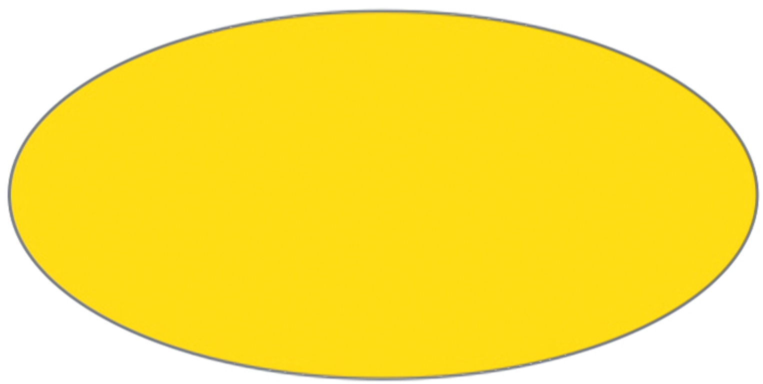
You can pose this question to a machine:

"The machine M recognizes language L"

- A language is called recursively enumerable if it is recognized by a Turing machine.

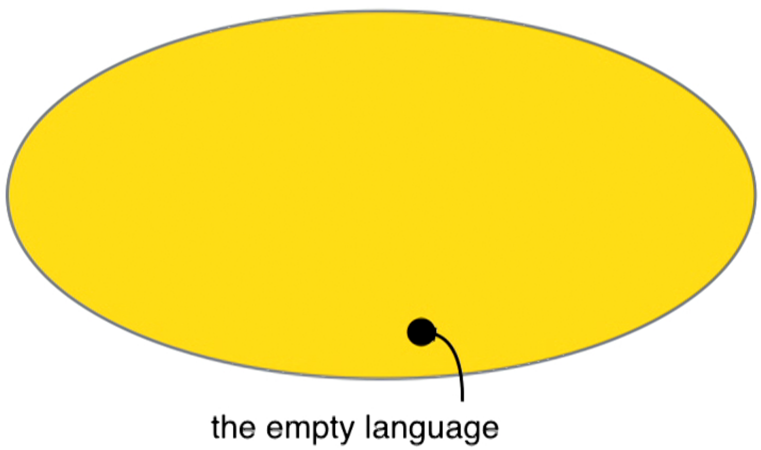
Undecidability in computer science

- Consider the set of recursively enumerable languages:



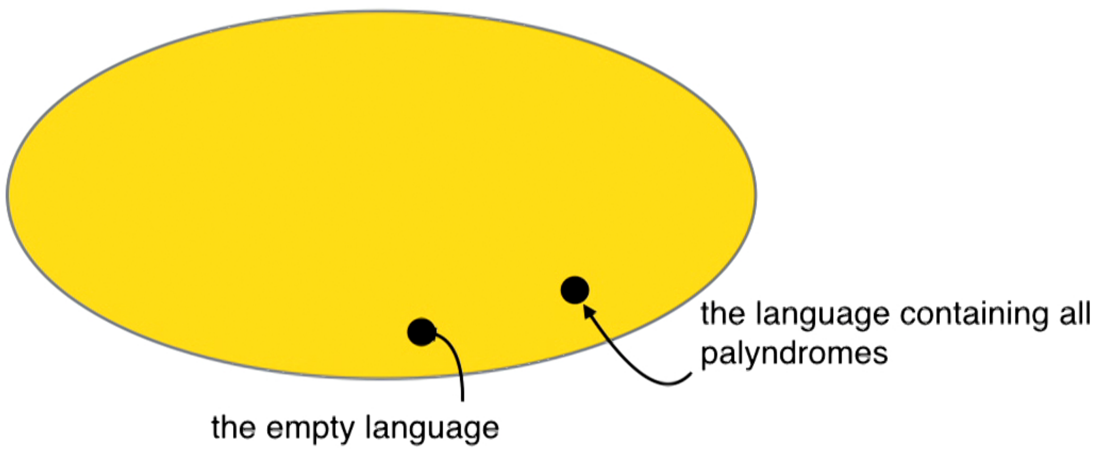
Undecidability in computer science

- Consider the set of recursively enumerable languages:



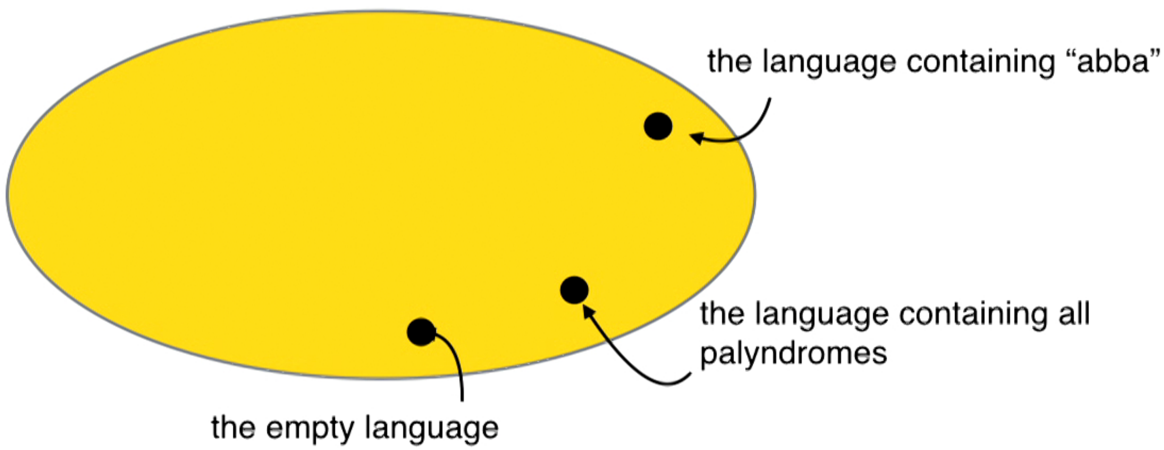
Undecidability in computer science

- Consider the set of recursively enumerable languages:



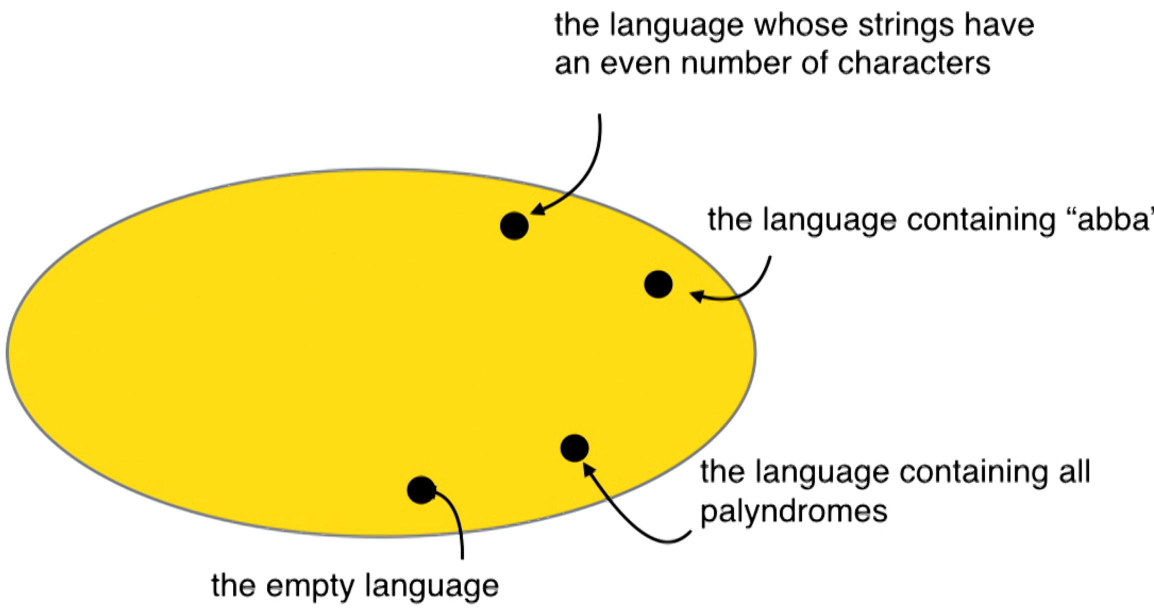
Undecidability in computer science

- Consider the set of recursively enumerable languages:



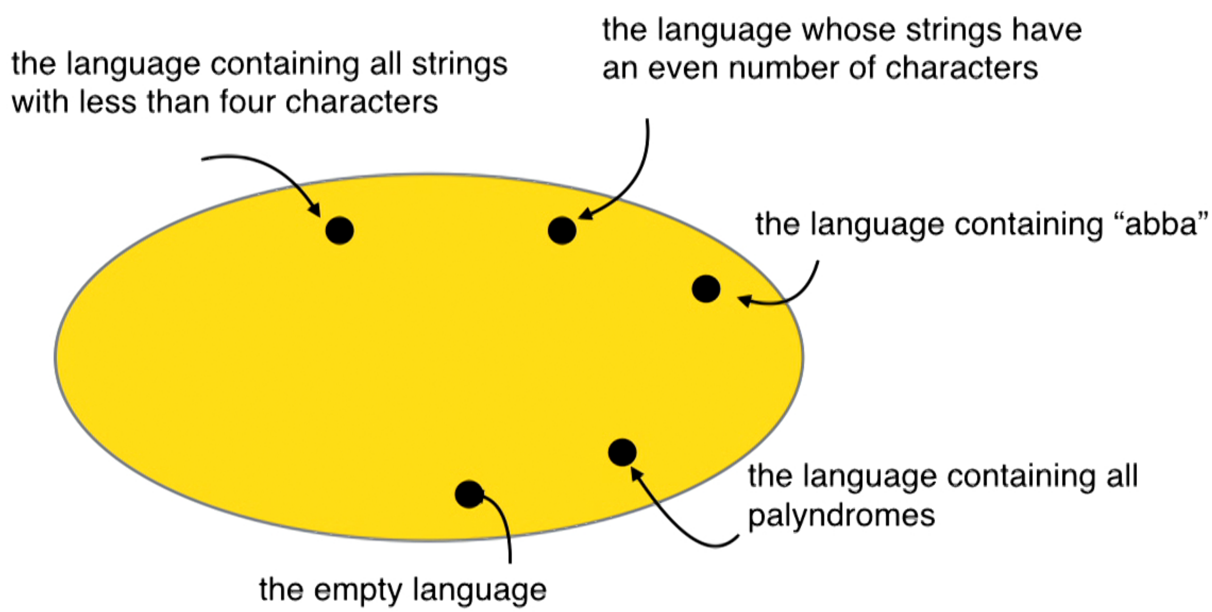
Undecidability in computer science

- Consider the set of recursively enumerable languages:



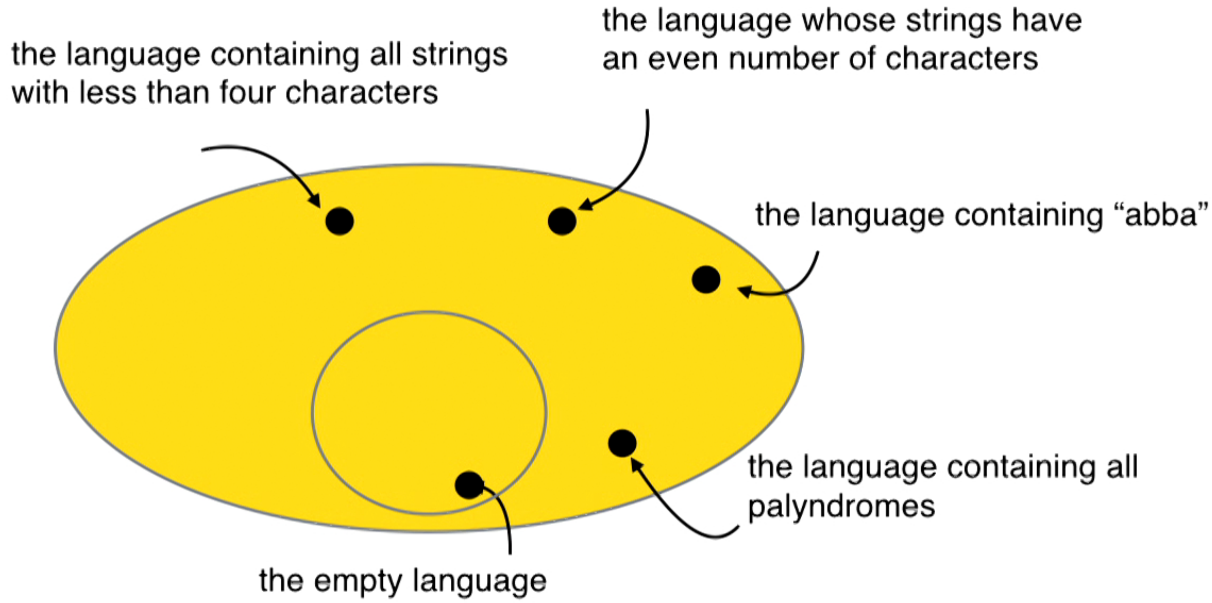
Undecidability in computer science

- Consider the set of recursively enumerable languages:



Undecidability in computer science

- Consider the set of recursively enumerable languages:



- A non-trivial property is a proper subset of this set.

Undecidability in computer science

- Rice's theorem: *Every* non-trivial property of the set of recursively enumerable languages is undecidable.

Undecidability in computer science

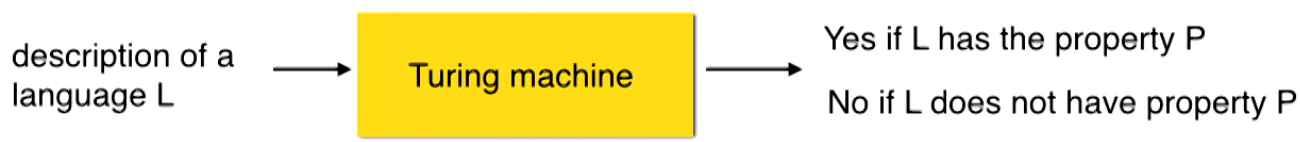
- Rice's theorem: *Every* non-trivial property of the set of recursively enumerable languages is undecidable.

Imagine that you want to decide whether a recursively enumerable language L has the non-trivial property P.

Undecidability in computer science

- Rice's theorem: *Every* non-trivial property of the set of recursively enumerable languages is undecidable.

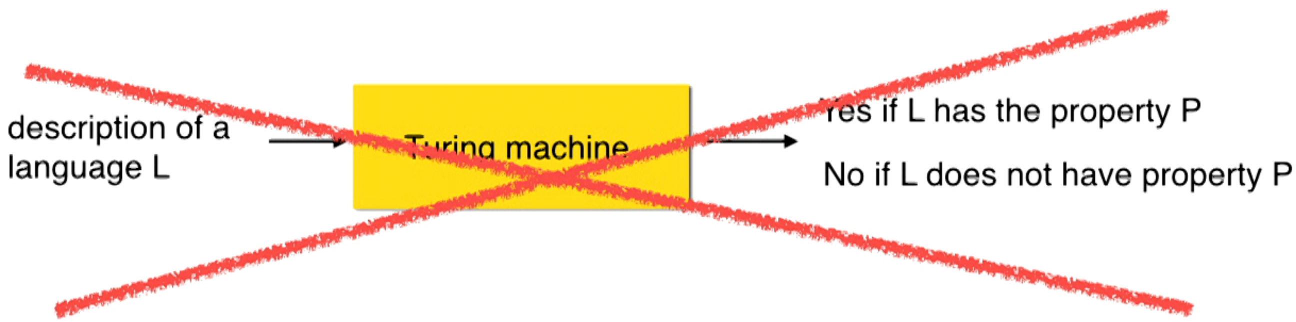
Imagine that you want to decide whether a recursively enumerable language L has the non-trivial property P.



Undecidability in computer science

- Rice's theorem: *Every* non-trivial property of the set of recursively enumerable languages is undecidable.

Imagine that you want to decide whether a recursively enumerable language L has the non-trivial property P.



There is no Turing machine that does this job.

Undecidability in physics

- Largely unexplored

Undecidability in physics

- Largely unexplored

  Undecidability of the spectral gap. T. S. Cubitt, D. Perez-Garcia, M. M. Wolf, Nature (2015)

Undecidability in physics

- Largely unexplored

  Undecidability of the spectral gap. T. S. Cubitt, D. Perez-Garcia, M. M. Wolf, Nature (2015)

- What is the importance of proving that a physical problem is undecidable?

Undecidability in physics

- Largely unexplored

  Undecidability of the spectral gap. T. S. Cubitt, D. Perez-Garcia, M. M. Wolf, Nature (2015)

- What is the importance of proving that a physical problem is undecidable?

  can we do some other mathematical abstraction of our physical problem
  which allows us to circumvent these limitations?

Undecidability in physics

- Largely unexplored

  Undecidability of the spectral gap. T. S. Cubitt, D. Perez-Garcia, M. M. Wolf, Nature (2015)

- What is the importance of proving that a physical problem is undecidable?

  can we do some other mathematical abstraction of our physical problem which allows us to circumvent these limitations?

  undecidability requires infinite sets, but is there infinity in physics?

Undecidability in physics

- Largely unexplored

  Undecidability of the spectral gap. T. S. Cubitt, D. Perez-Garcia, M. M. Wolf, Nature (2015)

- What is the importance of proving that a physical problem is undecidable?

  can we do some other mathematical abstraction of our physical problem which allows us to circumvent these limitations?

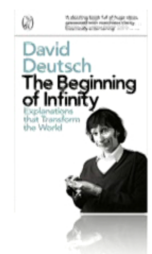
  undecidability requires infinite sets, but is there infinity in physics?

- Physics determines what is undecidable

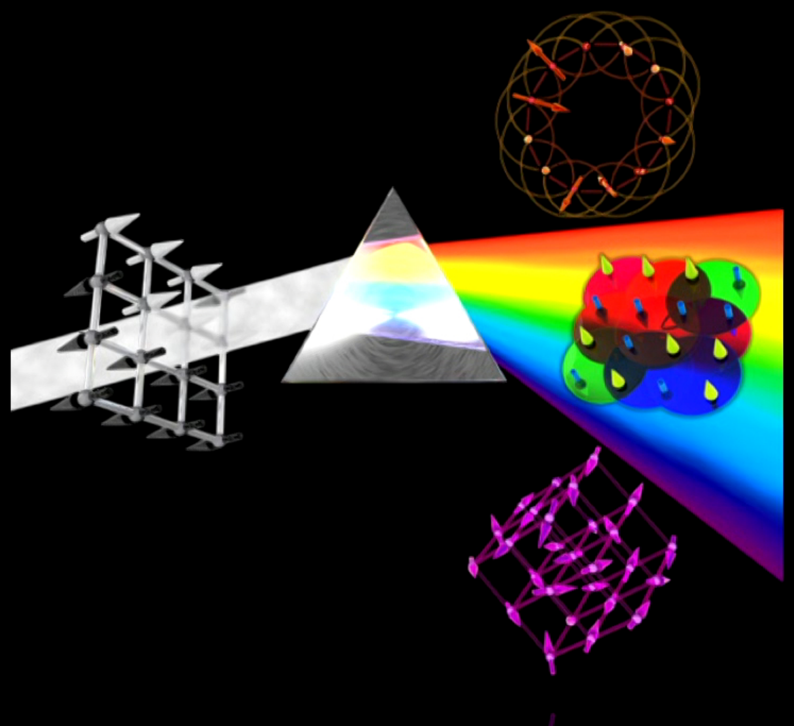
  All knowledge is generated by physical processes: e.g. a computation, a proof.

  Thus, its scope and limitations are conditioned by the laws of nature.

  E.g. only a finite amount of information can be processed in a finite amount of time.

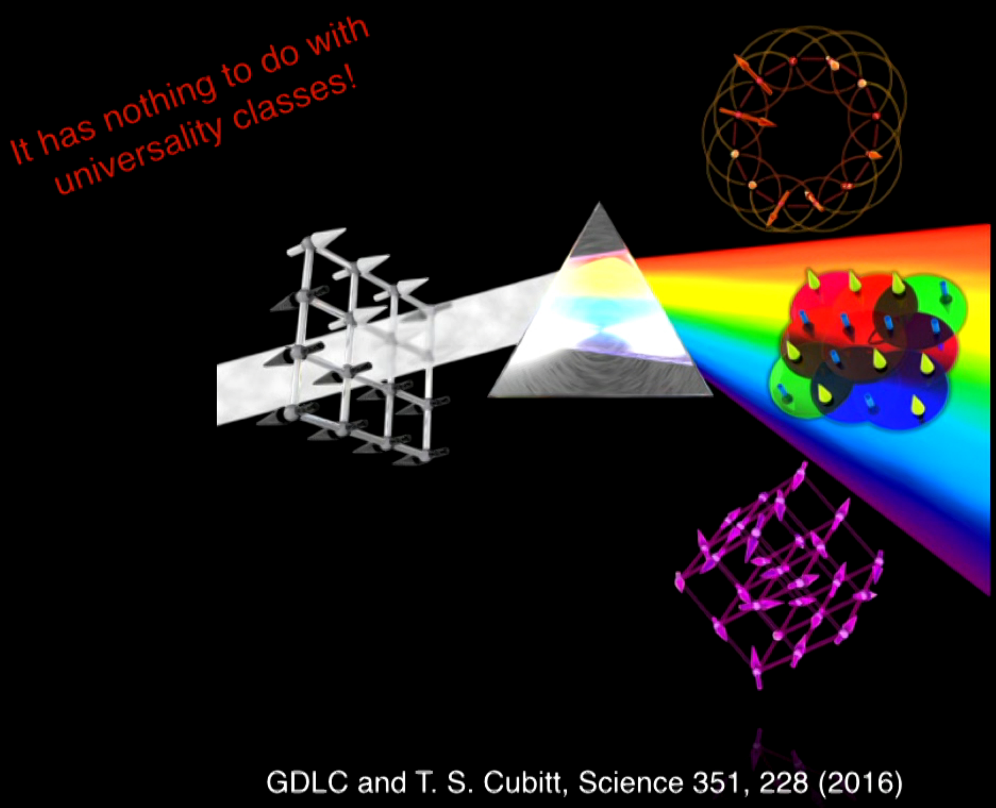


Universality in physics



GDLC and T. S. Cubitt, Science 351, 228 (2016)

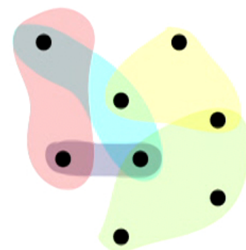
Universality in physics



GDLC and T. S. Cubitt, Science 351, 228 (2016)

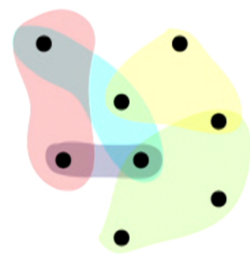
Spin models

A spin model:
- a set of classical degrees of freedom (the spins)
- a cost function (the hamiltonian)



Spin models

A spin model:
- a set of classical degrees of freedom (the spins)
- a cost function (the hamiltonian)



- Tool to model complexity “bottom up”

Spin models

- The 1D Ising model with coupling *J*:

Spin configuration	$\mapsto$	Energy	for any *n*
$s_1, \ldots, s_n$		$J \sum_{i=1}^{n-1} s_i s_{i+1}$	

Spin models

- The 1D Ising model with coupling *J*:

Spin configuration $\mapsto$ Energy for any *n*

$s_1, \ldots, s_n$

$$J \sum_{i=1}^{n-1} s_i s_{i+1}$$

x = Spin configuration → The 1D Ising model with coupling J → Energy of input x

for all x

Spin models

- The 2D Ising model with fields:

Spin configuration and coupling strengths	$\mapsto$	Energy	for any *n*
$s_1, \ldots, s_n \# h_1, \ldots, h_n \# J_{1,2}, \ldots, J_{n-1,n}$		$\sum_{(i,j)} J_{i,j} s_i s_j + \sum_i h_i s_i$	

Spin models

- The 2D Ising model with fields:

Spin configuration and coupling strengths $\mapsto$ Energy for any *n*

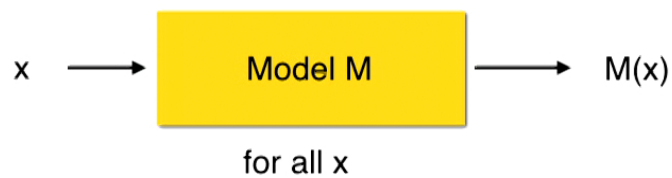
$$s_1, \ldots, s_n \# h_1, \ldots, h_n \# J_{1,2}, \ldots, J_{n-1,n} \qquad \sum_{(i,j)} J_{i,j} s_i s_j + \sum_i h_i s_i$$

x = Spin configuration and coupling strengths → **The 2D Ising model with fields** → Energy of input x

for all x

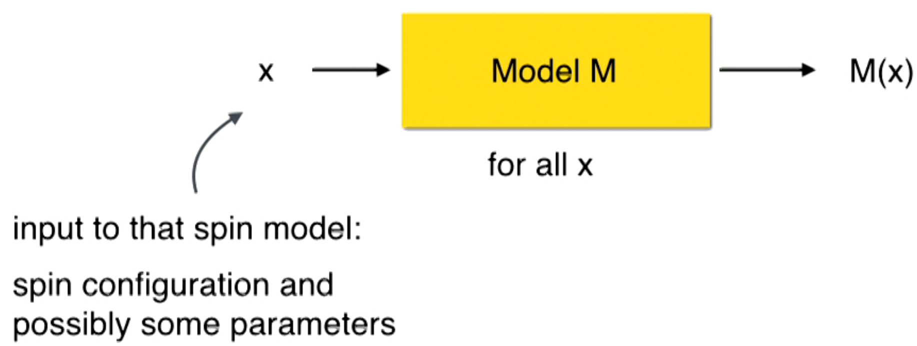
Spin models

- Spin model M:



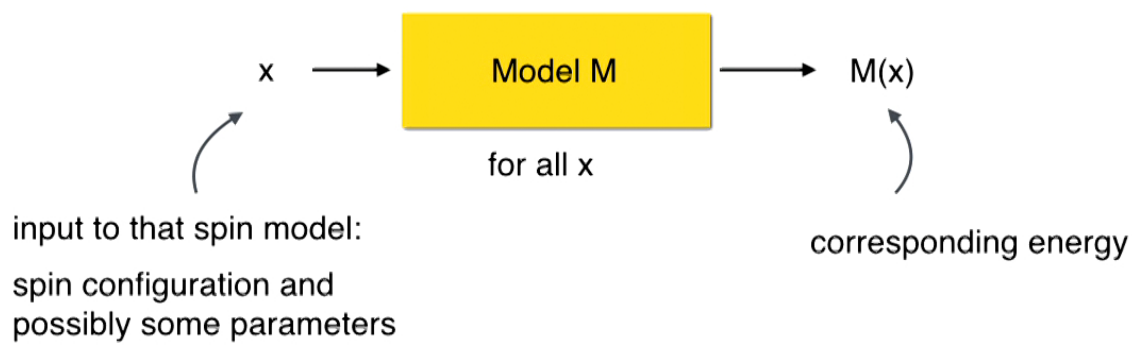
Spin models

- Spin model M:



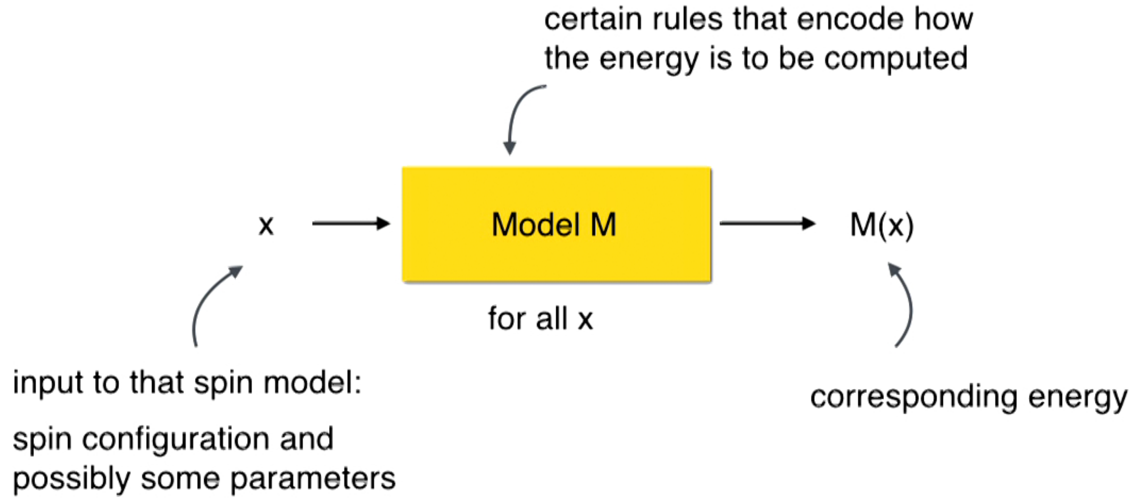
Spin models

- Spin model M:



Spin models

- Spin model M:



Simulation of spin models

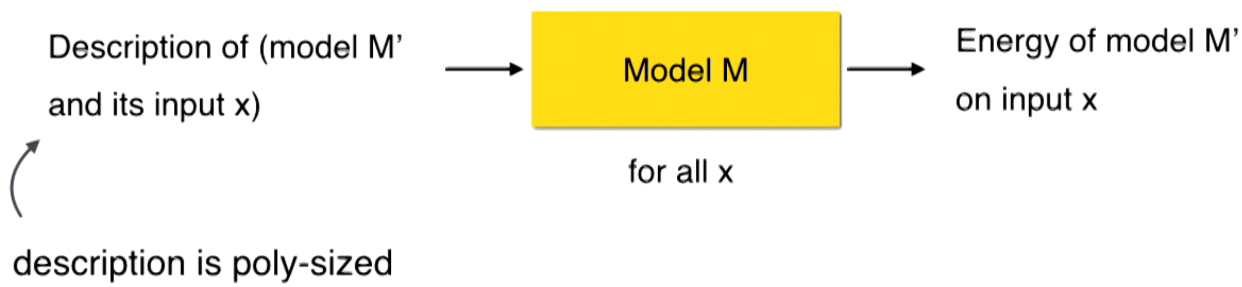
- A spin model M *simulates* a spin model M' if there is a description such that

Description of (model M' and its input x) → Model M → Energy of model M' on input x

for all x

Simulation of spin models

- A spin model M *simulates* a spin model M’ if there is a description such that



Simulation of spin models

- Trivial: “The 3D Ising model simulates the 2D Ising model”

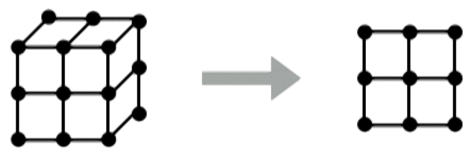
Simulation of spin models

- Trivial: "The 3D Ising model simulates the 2D Ising model"

Description of (2D Ising model and its input x) → **The 3D Ising model** → Energy of 2D Ising model on input x

for all x

Put all couplings outside a layer to 0



Simulation of spin models

- Non-trivial: “The 2D Ising model with fields can simulate the 3D Ising model”

Description of (3D Ising model and its input x) → **The 2D Ising model with fields** → Energy of 3D Ising model on input x

for all x

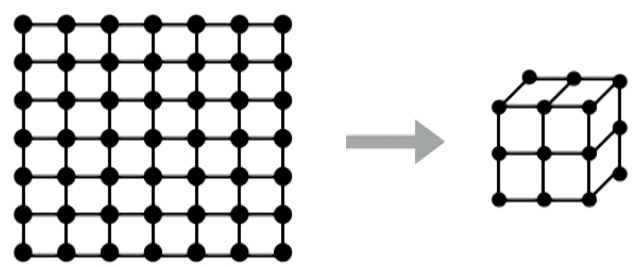
Simulation of spin models

- Non-trivial: "The 2D Ising model with fields can simulate the 3D Ising model"

Description of (3D Ising model and its input x) → **The 2D Ising model with fields** → Energy of 3D Ising model on input x

for all x

Project 3D lattice to 2D, use auxiliary spins, and coupling strengths…..



Universal spin models

- A spin model is *universal* if it can simulate all other spin models.

Universal spin models

- A spin model is *universal* if it can simulate all other spin models.

  A spin model M is universal if for every model M' there is a description d such that M(d(M',x)) = M'(x) for all x.

Universal spin models

- A spin model is *universal* if it can simulate all other spin models.

  A spin model M is universal if for every model M' there is a description d such that M(d(M',x)) = M'(x) for all x.

- Note: in a universal Turing machine, the description must be common

Universal spin models

- A spin model is *universal* if it can simulate all other spin models.

  A spin model M is universal if for every model M' there is a description d such that M(d(M',x)) = M'(x) for all x.

- Note: in a universal Turing machine, the description must be common

  these spin models are machines that only need to compute energies…

Universal spin models

- The 2D Ising model with fields is universal.

Universal spin models

- The 2D Ising model with fields is universal.

Description of (other model and its input x) → **The 2D Ising model with fields** → Energy of other model on input x

for all x

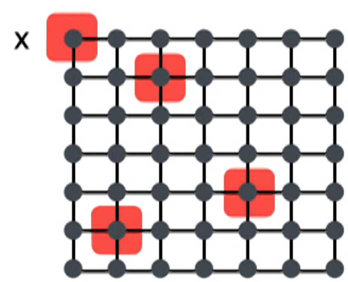
Universal spin models

- The 2D Ising model with fields is universal.

Description of (other model and its input x) → **The 2D Ising model with fields** → Energy of other model on input x

for all x

spin configuration and coupling strengths of 2D Ising model



Universal spin models

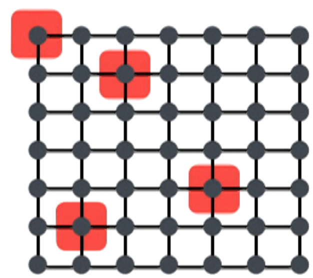
- The 2D Ising model with fields is universal.

Description of (other model and its input x) → **The 2D Ising model with fields** → Energy of other model on input x

for all x

spin configuration and coupling strengths of 2D Ising model

x



- Apparently simple model. Enough to reach this form of universality.

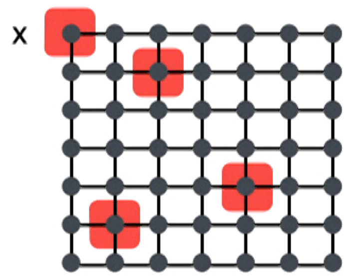
Universal spin models

- The 2D Ising model with fields is universal.

Description of (other model and its input x) → **The 2D Ising model with fields** → Energy of other model on input x

for all x

spin configuration and coupling strengths of 2D Ising model



- Apparently simple model. Enough to reach this form of universality.
- 2D Ising model without fields is not universal….

Universal spin models

- Full characterization of universality in physics:

  A spin model is universal if and only if

  - its ground state energy problem admits a faithful polynomial-time reduction from SAT, and
  - it is closed.

Universal spin models

- Full characterization of universality in physics:

  A spin model is universal if and only if

  - its ground state energy problem admits a faithful polynomial-time reduction from SAT, and
  - it is closed.

- In computer science, it is undecidable to determine whether a Turing machine is universal.

Universal spin models

- Full characterization of universality in physics:

  A spin model is universal if and only if

  - its ground state energy problem admits a faithful polynomial-time reduction from SAT, and
  - it is closed.

- In computer science, it is undecidable to determine whether a Turing machine is universal.

- I will argue that our notion of universality is weaker than that in computer science.

Connections between the two notions of universality

Work in progress

with Georg Moser

Machines and languages

- A computation " x —> f(x)" can be seen as the relation "(x, f(x)) is true"

Machines and languages

- A computation " x —> f(x)" can be seen as the relation "(x, f(x)) is true"

- Consider a machine that accepts or rejects strings

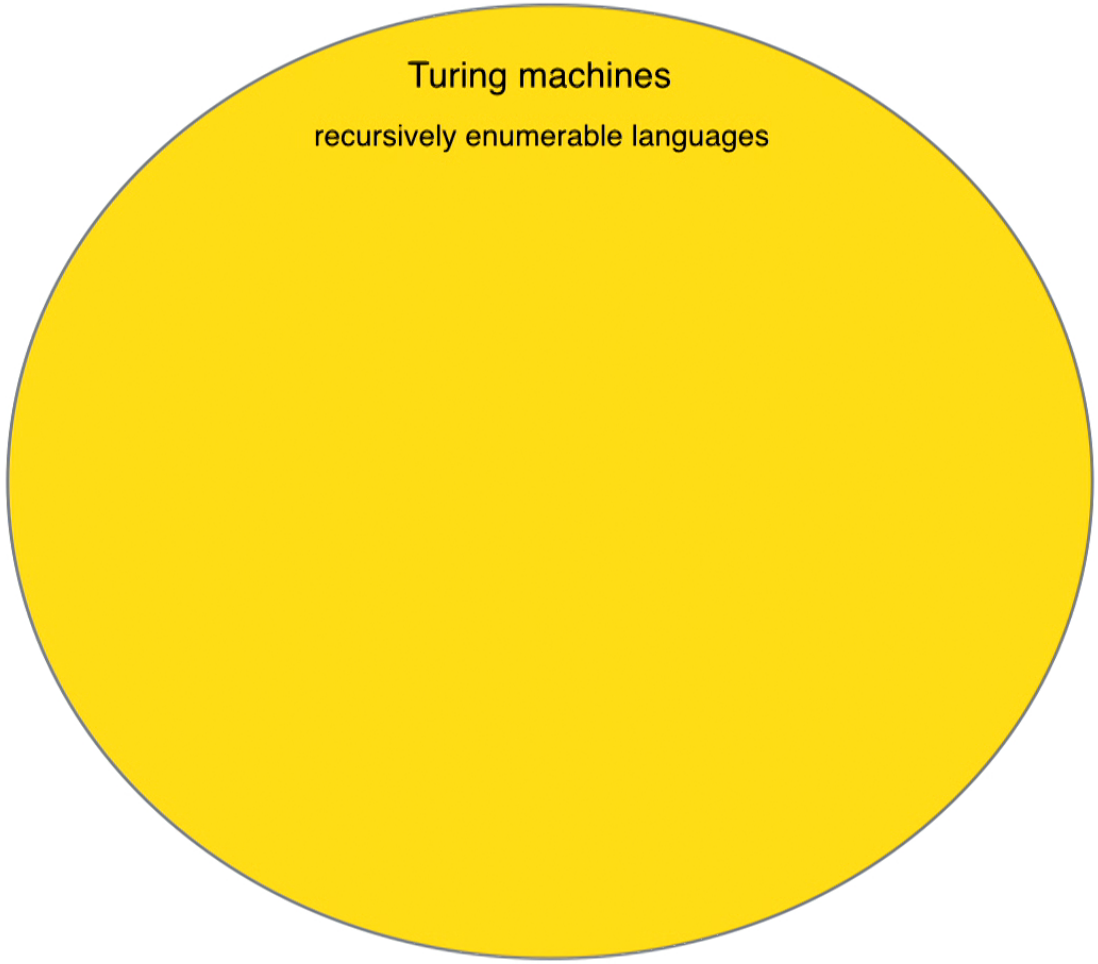
Machines and languages

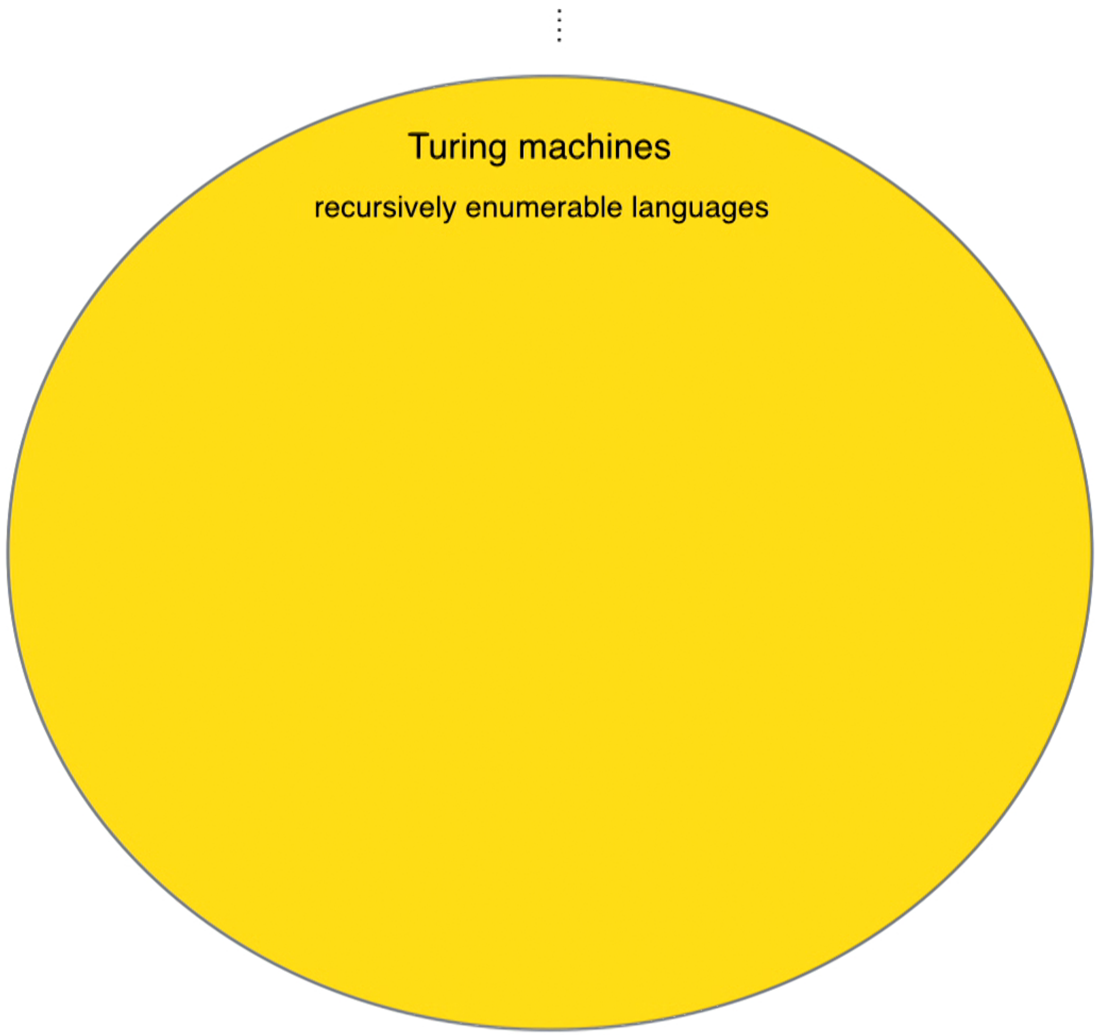
- A computation " x —> f(x)" can be seen as the relation "(x, f(x)) is true"

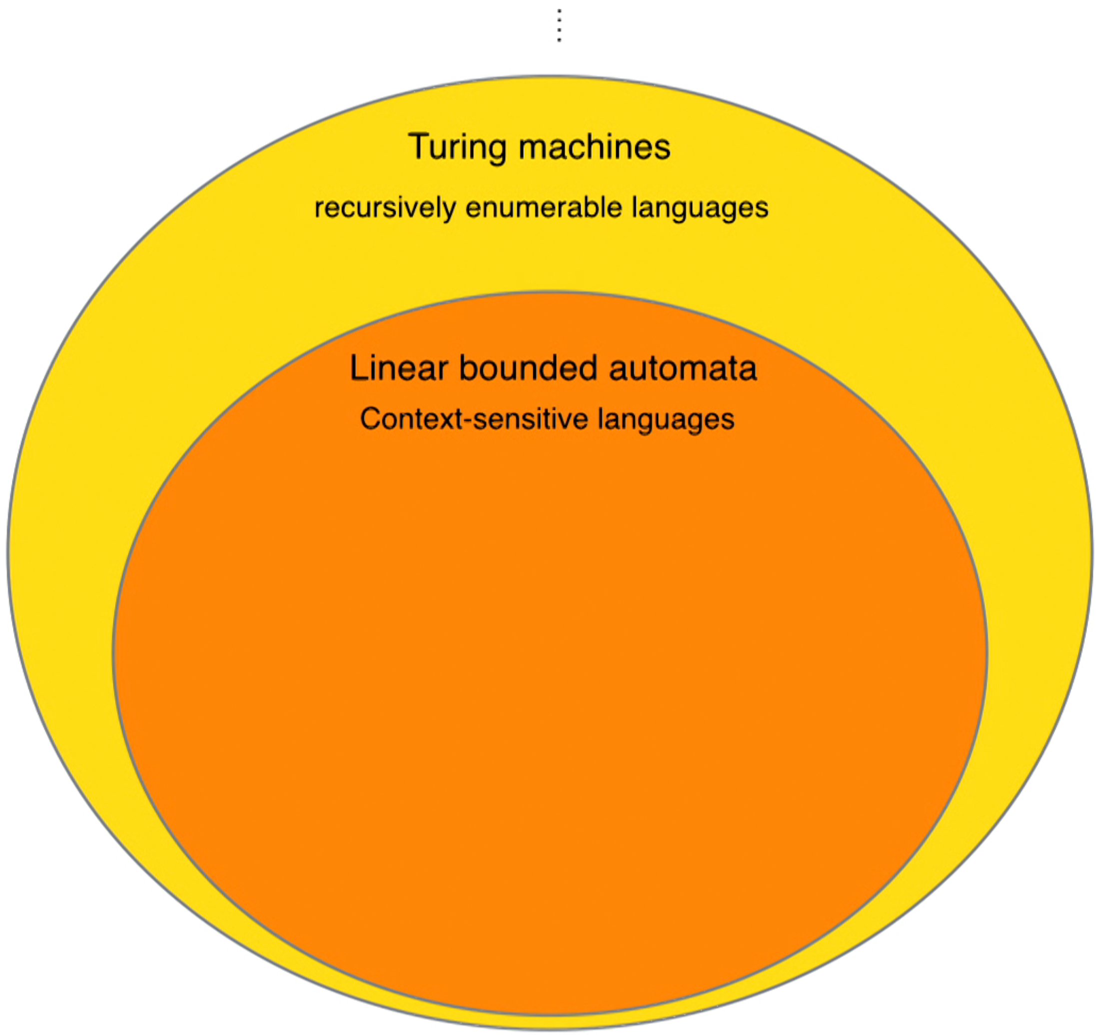
- Consider a machine that accepts or rejects strings

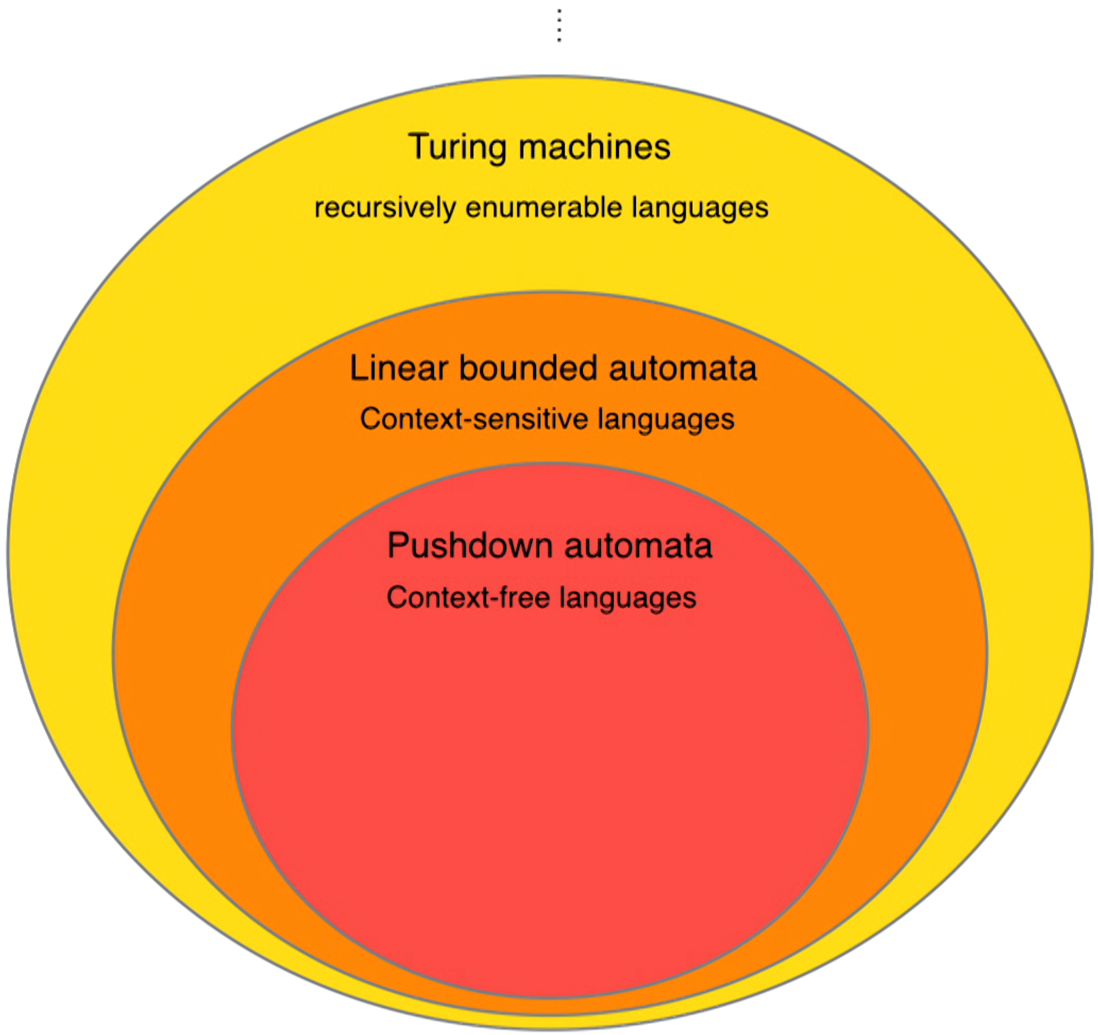
- The language associated to a machine is the set of strings accepted by that machine

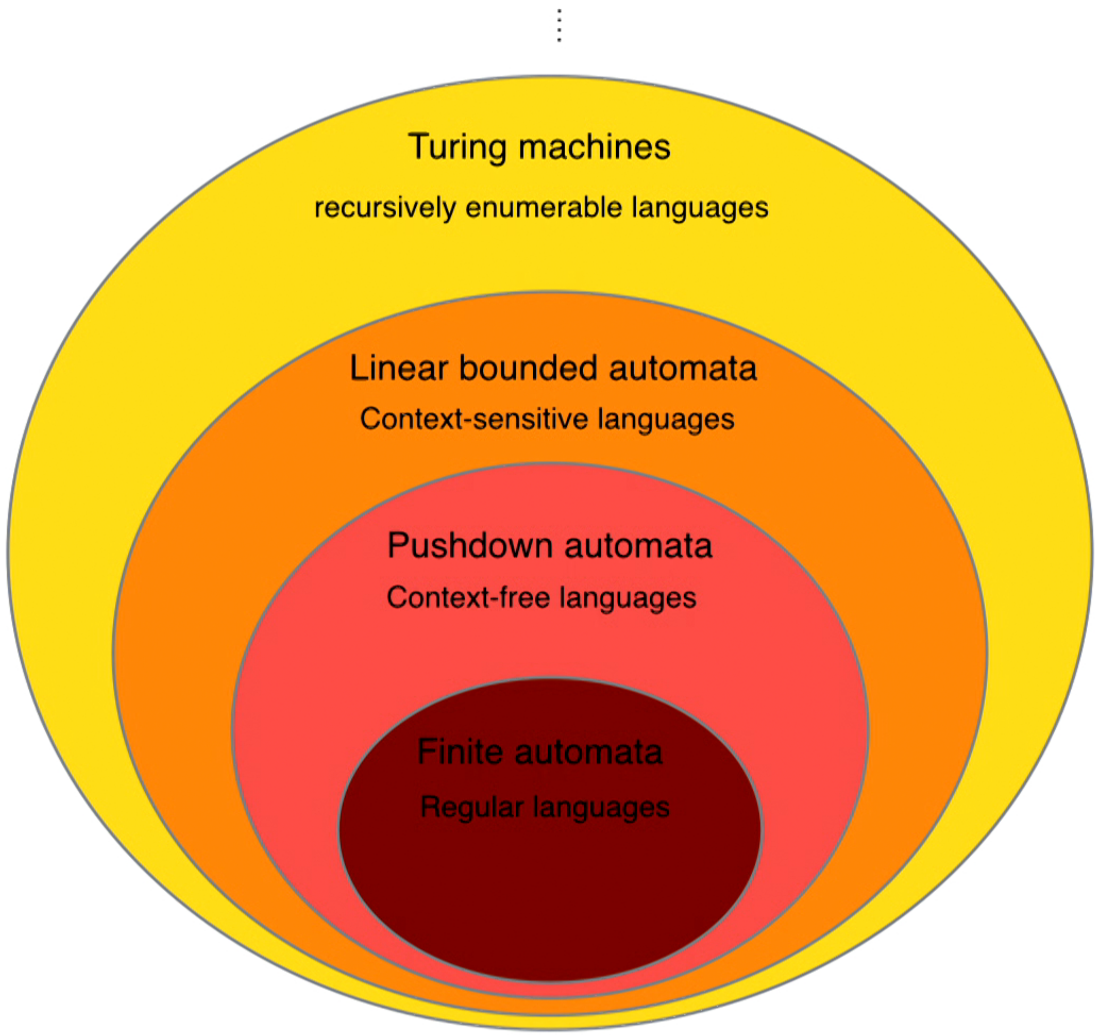
  L(M) = {x accepted by M}


Turing machines
recursively enumerable languages

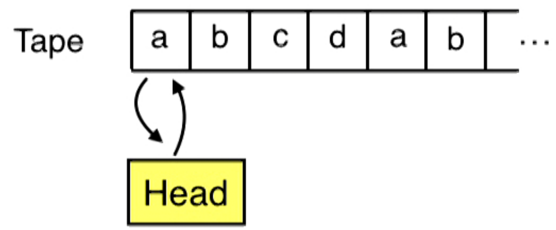

⋮
Turing machines
recursively enumerable languages


⋮
Turing machines
recursively enumerable languages
Linear bounded automata
Context-sensitive languages

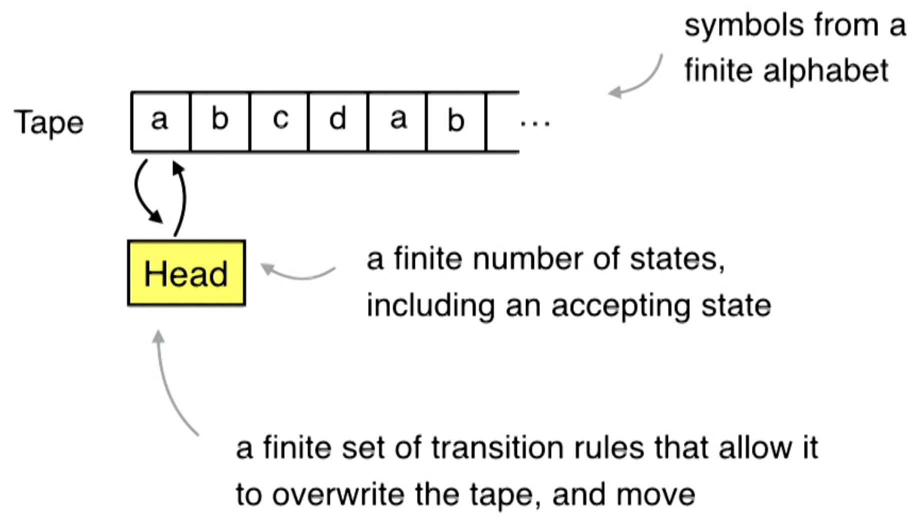

⋮
Turing machines
recursively enumerable languages
Linear bounded automata
Context-sensitive languages
Pushdown automata
Context-free languages


⋮
Turing machines
recursively enumerable languages
Linear bounded automata
Context-sensitive languages
Pushdown automata
Context-free languages
Finite automata
Regular languages

Turing machine

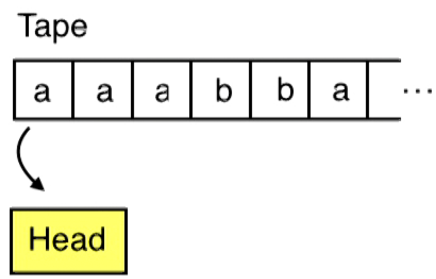


Turing machine



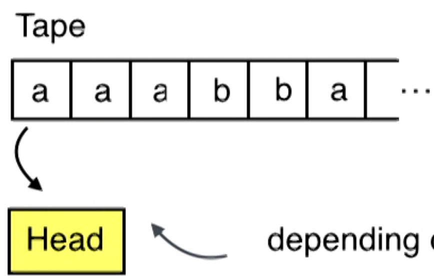
Finite automata

- A finite automaton is a read-only Turing machine



Finite automata

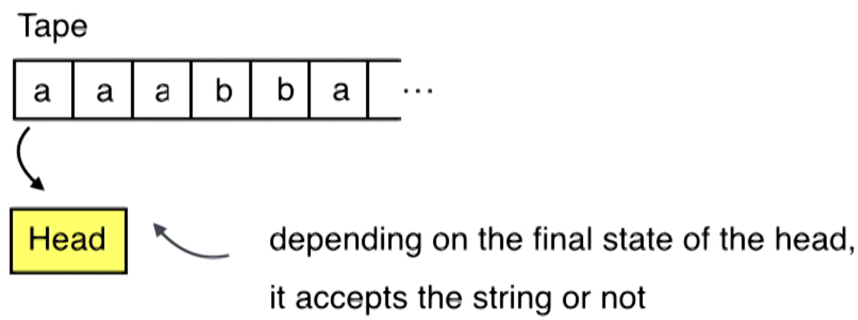
- A finite automaton is a read-only Turing machine



depending on the final state of the head,
it accepts the string or not

Finite automata

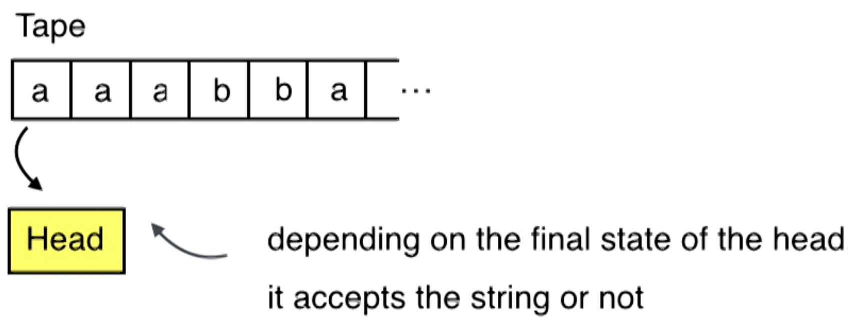
- A finite automaton is a read-only Turing machine



- Finite automata can recognize regular languages.

Finite automata

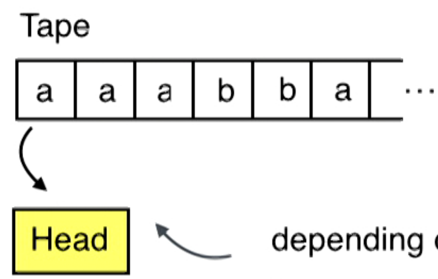
- A finite automaton is a read-only Turing machine



- Finite automata can recognize regular languages.
- They can do pattern recognition.

Finite automata

- A finite automaton is a read-only Turing machine



depending on the final state of the head, it accepts the string or not

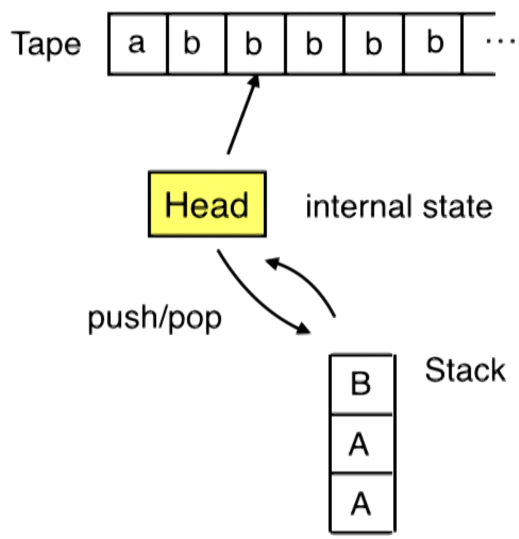
- Finite automata can recognize regular languages.
- They can do pattern recognition.

E.g. recognize L = the set of strings that contain abba

- Every finite language is regular.

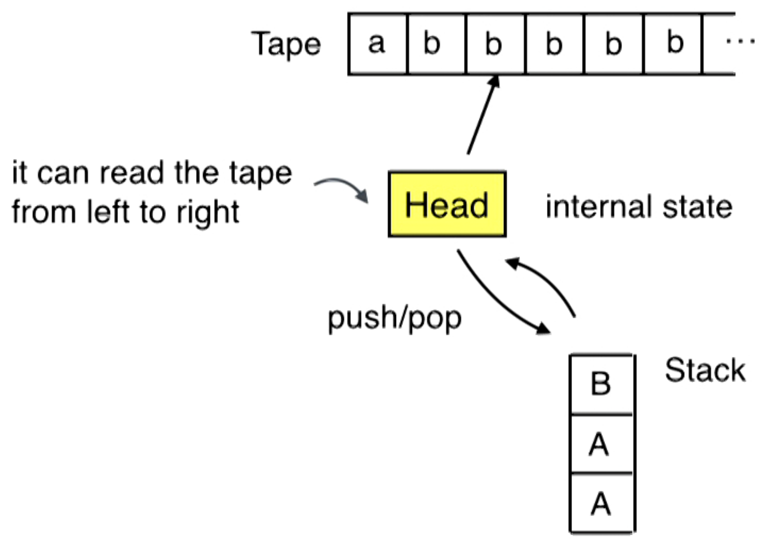
Pushdown automata

- It is a non-deterministic finite automaton with a stack



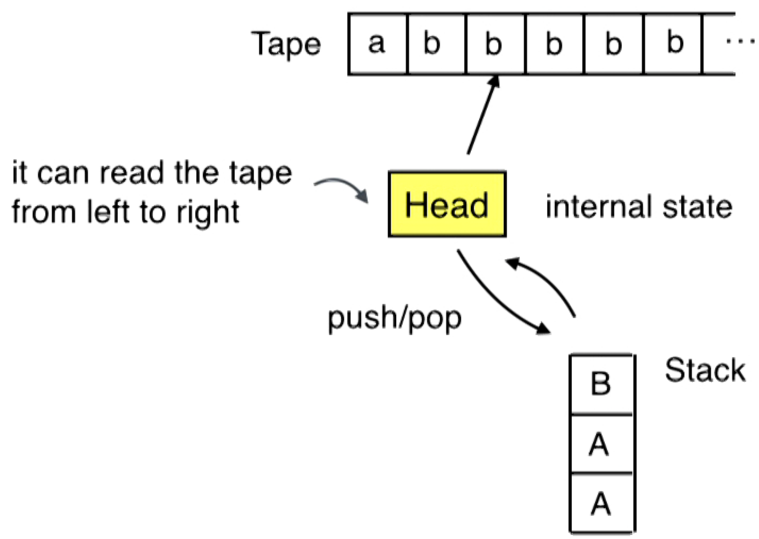
Pushdown automata

- It is a non-deterministic finite automaton with a stack



Pushdown automata

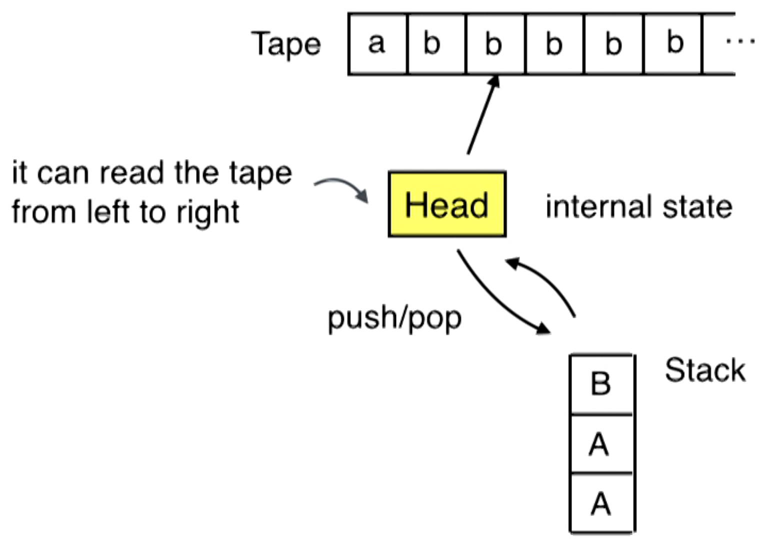
- It is a non-deterministic finite automaton with a stack



- It can recognize context-free languages

Pushdown automata

- It is a non-deterministic finite automaton with a stack

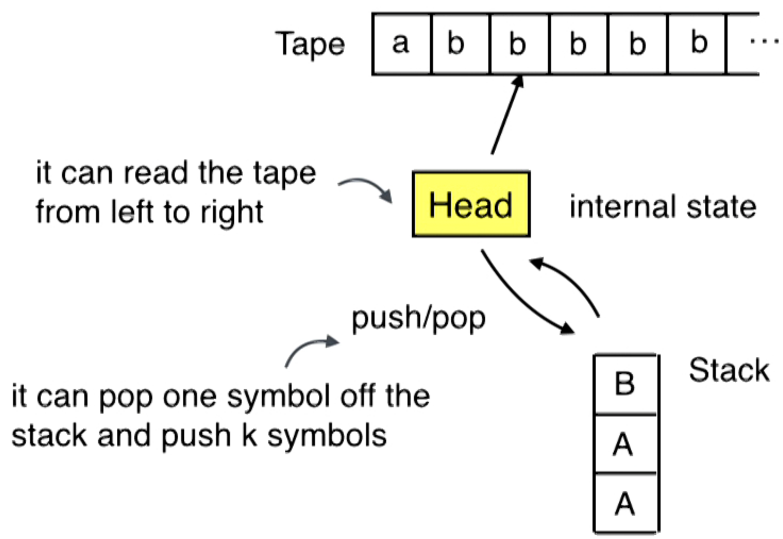


- It can recognize context-free languages

E.g. $L = \{a^n b^n | n \geq 0\}$

Pushdown automata

- It is a non-deterministic finite automaton with a stack



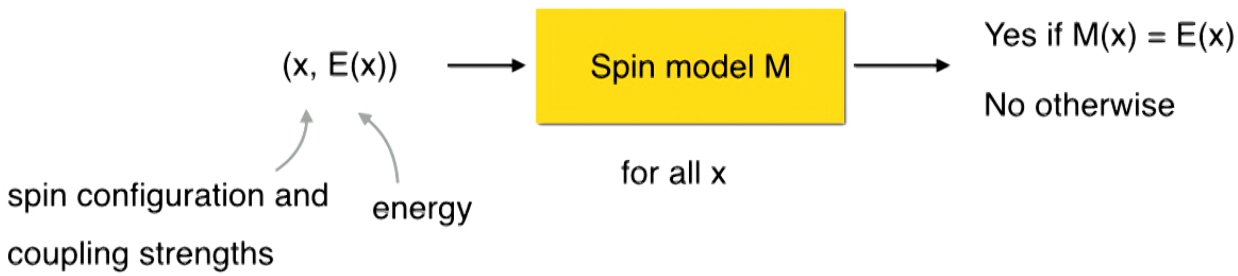
- It can recognize context-free languages

  E.g. $L = \{a^n b^n | n \geq 0\}$

  E.g. the language containing all strings of balanced parantehesis: (), (()), ()(), (()(())), ...

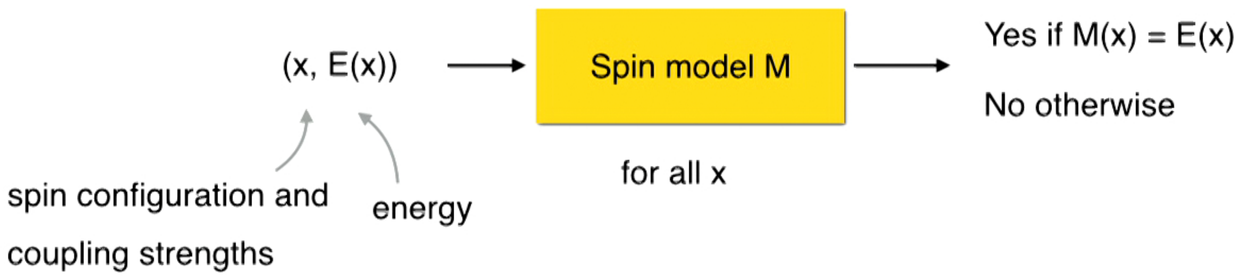
A hamiltonian as a machine

- Recall: a spin model is a machine



A hamiltonian as a machine

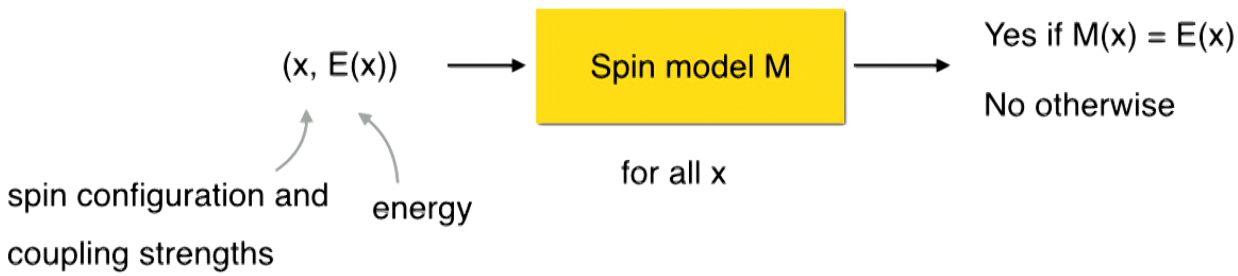
- Recall: a spin model is a machine



- What is the power of this machine?

A hamiltonian as a machine

- Recall: a spin model is a machine



- What is the power of this machine?

  I claim that it need only be a pushdown automaton.

A hamiltonian as a machine

- The hamiltonian as a pushdown automaton:

  We have a function $H = \sum_i h_i$

A hamiltonian as a machine

- The hamiltonian as a pushdown automaton:

We have a function $H = \sum_i h_i$

This acts on k spins. Hence it can take only a finite number of values, which can be coded in the state of the head.

A hamiltonian as a machine

- The hamiltonian as a pushdown automaton:

We have a function $H = \sum_i h_i$

This acts on k spins. Hence it can take only a finite number of values, which can be coded in the state of the head.

The result gets multiplied by the coupling strength, and stored in the stack.

A hamiltonian as a machine

- The hamiltonian as a pushdown automaton:

We have a function $H = \sum_i h_i$

This acts on k spins. Hence it can take only a finite number of values, which can be coded in the state of the head.

The result gets multiplied by the coupling strength, and stored in the stack.

Finally, the energy written in the tape is subtracted from the stack.

A hamiltonian as a machine

- The hamiltonian as a pushdown automaton:

We have a function $H = \sum_i h_i$

This acts on k spins. Hence it can take only a finite number of values, which can be coded in the state of the head.

The result gets multiplied by the coupling strength, and stored in the stack.

Finally, the energy written in the tape is subtracted from the stack.

If the final stack is empty, it accepts. Else, it rejects.

A hamiltonian as a machine

- Simulation between spin models translates to reduction between pushdown automata.
- Implications of our full characterization of simulations between spin models?

Summary &
Perspectives

Summary

- Universality everywhere

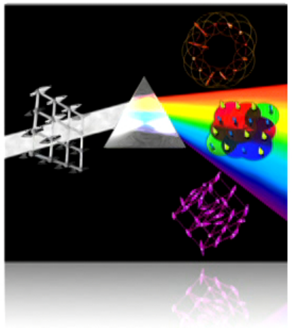
Summary

- Universality everywhere …. and undecidability

Summary

- Universality everywhere …. and undecidability

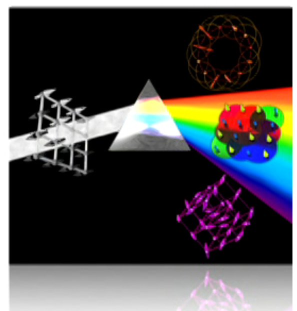
- New universality results in physics



Summary

- Universality everywhere …. and undecidability

- New universality results in physics

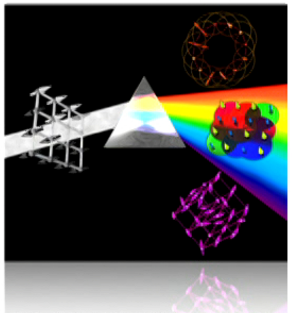


- Is this notion of universality comparable to universality in computer science?

Summary

- Universality everywhere .... and undecidability

- New universality results in physics

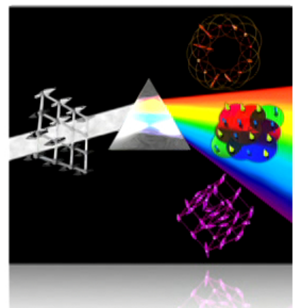


- Is this notion of universality comparable to universality in computer science?
- A hamiltonian as a pushdown automaton

Summary

- Universality everywhere …. and undecidability

- New universality results in physics



- Is this notion of universality comparable to universality in computer science?
- A hamiltonian as a pushdown automaton
- Universality results in physics: characterization of pushdown automata?

More perspectives

More perspectives

- Implications of these results for undecidability in physics?

More perspectives

- Implications of these results for undecidability in physics?
- Implications of universality results for artificial intelligence

More perspectives

- Implications of these results for undecidability in physics?
- Implications of universality results for artificial intelligence
- "Practical" implications of universality results:
  simulation of one model with another

More perspectives

- Implications of these results for undecidability in physics?
- Implications of universality results for artificial intelligence
- "Practical" implications of universality results:
  simulation of one model with another
- Implications of universality results for condensed matter:
  theory of universality classes