A mixture of computability and ordinals, the infinite time Turing machines

Sabrina Ouazzani

Paris-Est Créteil University

Avril 2017

Computability

- Describe what is a computation.
- Describe what runs a computation.

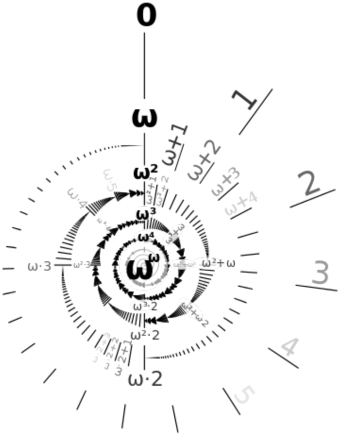
Ordinals: counting through the infinite

We denote ω the set of all natural numbers. But ω is not the only infinite...

Ordinals: counting through the infinite

We denote ω the set of all natural numbers. But ω is not the only infinite...

We can carry on counting!



1 Compute?

2 Infinite time Turing machines

3 Some particularities of infinite time

4 Conclusion

Compute

Definition (Algorithm).

Sequence of instructions:

- finite;
- not ambiguous;
- allows to solve a problem.

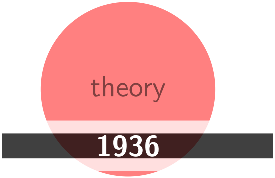
Example: compute the n first terms of the hailtstone sequence (Collatz conjecture)

Variables: counter k
for k *from* 0 *to* n **do**
  if m *is even* **then**
    $m \leftarrow m/2$;
  else
    $m \leftarrow m \times 3 + 1$;
  end
  $k \leftarrow k + 1$;
end

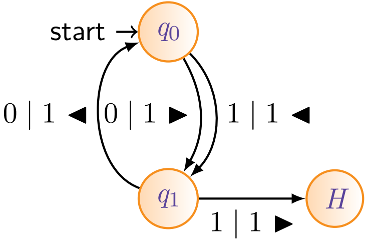
With $n = 7$ and $m = 10$ we obtain the sequence 10, 5, 16, 8, 4, 2, 1.

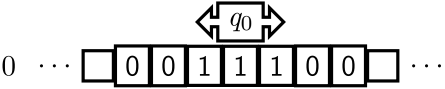
Open question (conjecture, 1952): for all m, does we always reach a 1?

Compute

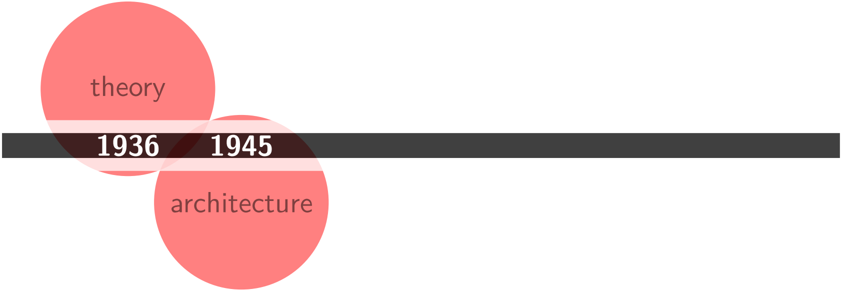


Turing machine.

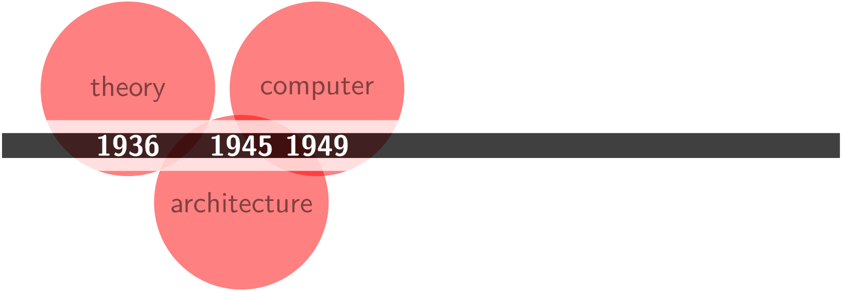




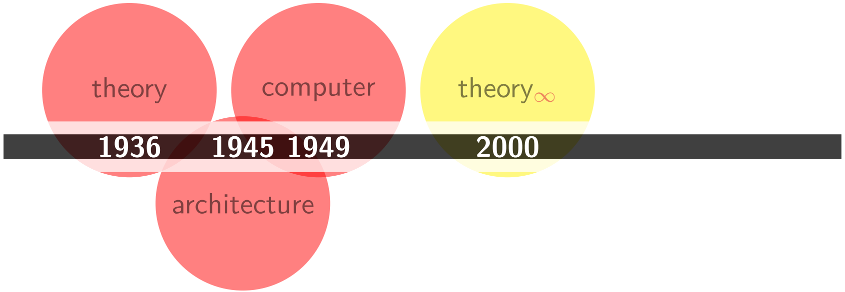
Timeline



Timeline

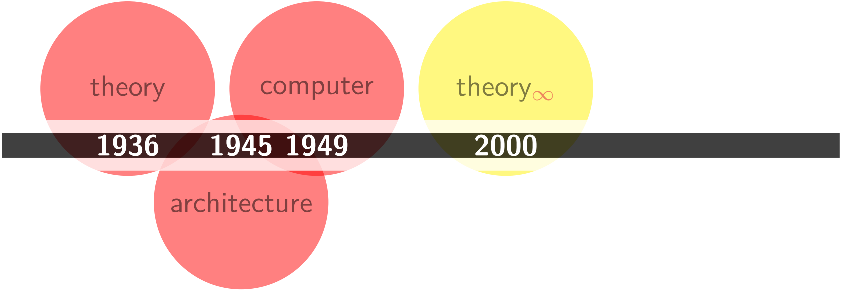


Timeline

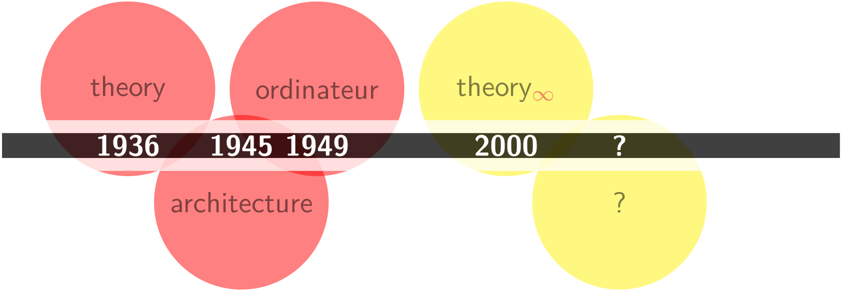

theory
computer
theory$_\infty$
1936
1945
1949
2000
architecture

- Solve the Collatz conjecture.
- For all the natural numbers, apply the algorithm.

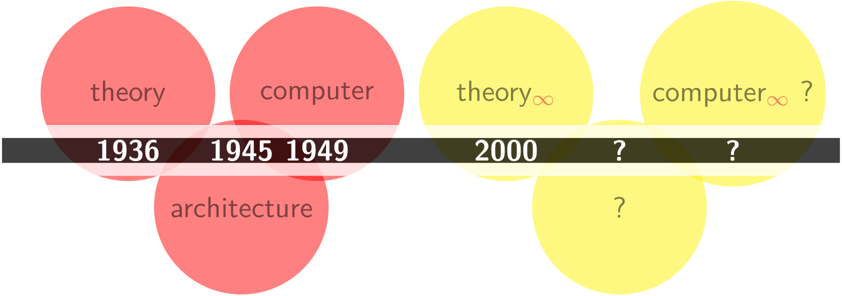
Timeline


theory
computer
theory$_\infty$
1936
1945
1949
2000
architecture

Timeline


theory
ordinateur
theory$_\infty$
1936
1945
1949
2000
?
architecture
?

Timeline


theory
computer
theory$_\infty$
computer$_\infty$?
1936
1945
1949
2000
?
?
architecture
?

Motivations: build links between Computer Science and Logic

- Ordinals as time for computation.
- Peculiar ordinal properties.
- Proof of mathematical properties from an algorithmic point of view.

1 Compute?

2 Infinite time Turing machines

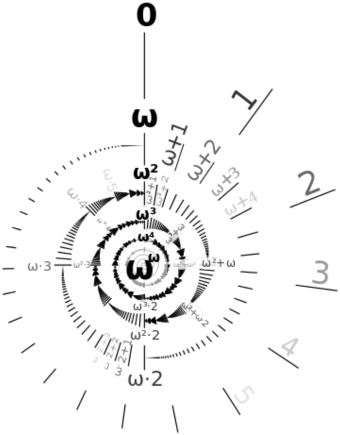
3 Some particularities of infinite time

4 Conclusion

Ordinals: counting through the infinite

We denote ω the set of all natural numbers. But ω is not the only infinite...

We can carry on counting!



Ordinals

Definition (Ordinal)**.**

Transitive well-ordered set for the membership relation.

$$0 := \emptyset$$
$$1 := \{0\} = \{\emptyset\}$$
...
$$\omega := \{0, 1, 2, 3, \cdots\}$$
$$\omega + 1 := \{0, 1, 2, 3, \cdots, \omega\}$$
...
$$\omega.2 := \{0, 1, 2, \cdots, \omega, \omega + 1, \omega + 2 \ldots\}$$

- If α is an ordinal, then $\alpha \cup \{\alpha\}$, denoted $\alpha + 1$ is called successor of α and is an ordinal;
- let A be a set of ordinal numbers, then $\alpha = \bigcup_{\beta \in A} \beta$ is a limit ordinal.

Encoding countable ordinals

Countable ordinal = well order on $\mathbb{N}$.

Codage 1 (Encoding countable ordinals by reals)**.**

Let $<$ be an order on the natural numbers.
The real r is a code for the order-type of $<$ if, for $i = \langle x, y\rangle$, the i-th bit of r is **1** if and only if $x < y$.

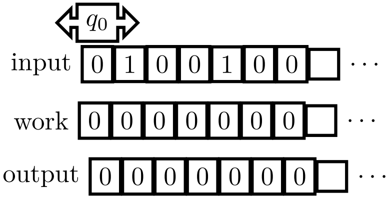
Example: $\omega.2 = \omega + \omega \rightsquigarrow$ even integers lower than odd integers.

$$0 = \langle 0, 0\rangle \ 1 = \langle 0, 1\rangle \ \cdots \ r = 0_0 1_1 0_2 0_3 0_4 1_5 0_6 1_7 1_8 1_9 1_{10} \cdots$$

Structure of infinite time Turing machines (ITTM)

- 3 right-infinite tapes
- a single head
- binary alphabet $\{0, 1\}$
- additional special limit state $\lim$
- computation steps are indexed by ordinals

Configuration

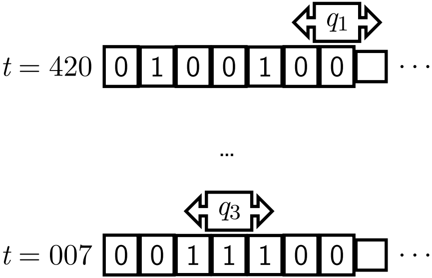


Operating an ITTM

Configuration at $\alpha + 1$.

$\rightsquigarrow$

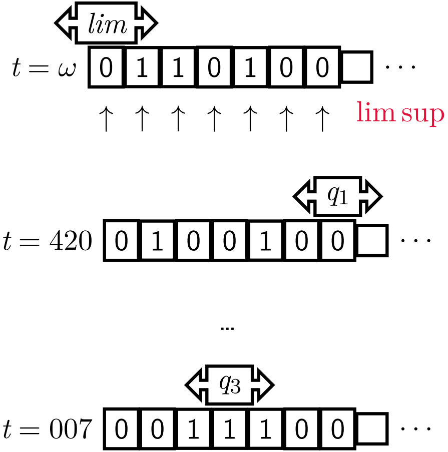
Configuration at α.



Operating an ITTM

Configuration limit:

- head: initial position;
- state: $\lim$;
- each cell: *lim sup* of cell values before.



Halting

- Machines halt when they reach the halting state.
- We consider the strong stabilisation of cells at 0.

Theorem 1 (Hamkins, Lewis [HL00])**.**

Either an ITTM halts in a countable numer of steps, either it begins looping in a **countable number of steps**.

- We focus on the halting problem on 0.

1 Compute?

2 Infinite time Turing machines

3 Some particularities of infinite time

4 Conclusion

Clockable and writable ordinals

Two natural notions:

Definition (Clockable ordinal)**.**

α clockable: there exists an ITTM that **halts** on input $000\dots$ in exactly α steps of computation.

Definition (Writable ordinal)**.**

α writable: there exists an ITTM that **writes a code** for α on input $000\dots$ and halts.

Supremum

Theorem 2 (Welch [Wel09]).

The supremum of the clockable ordinals is equal to the supremum of the writable ordinals. It is called λ.

λ is a rather large countable ordinal...

Let's count!

Count with a clockable ordinal $\rightsquigarrow$ Clock.

Like an hourglass, execute operations while clocking the desired ordinal.

Speed-up lemma (Hamkins, Lewis [HL00]).

If p halts on 0 in $\alpha + n$ steps, then there exists p' which halts on 0 in α steps (and computes the same). $\rightsquigarrow$ **limit** ordinals

Count with a writable ordinal $\rightsquigarrow$ Empty an order.

It is about counting through the encoding of an ordinal.

...

What about the particularities of these ordinals?

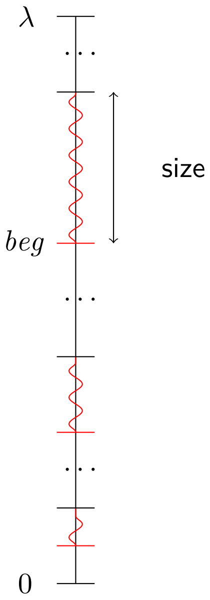
Gap

There exist writable ordinals that are not clockable such that:

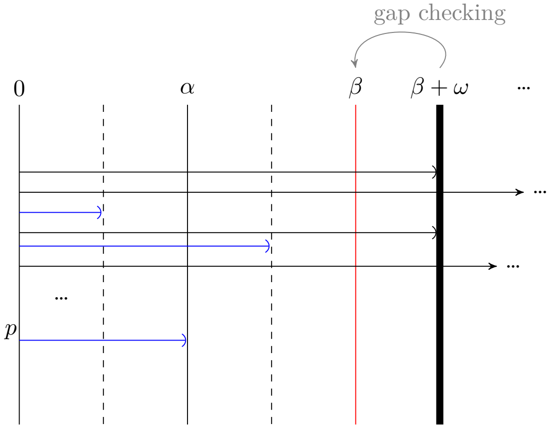
- they form intervalles;
- these intervalles have limit sizes.

Definition (Gap).

Intervalles of not clockable ordinals.


λ
...
size
beg
...
...
0

Proof of gap existence



Simulation of all programs on input 0.

In blue: halting programs. In red: limit step, begins a gap?

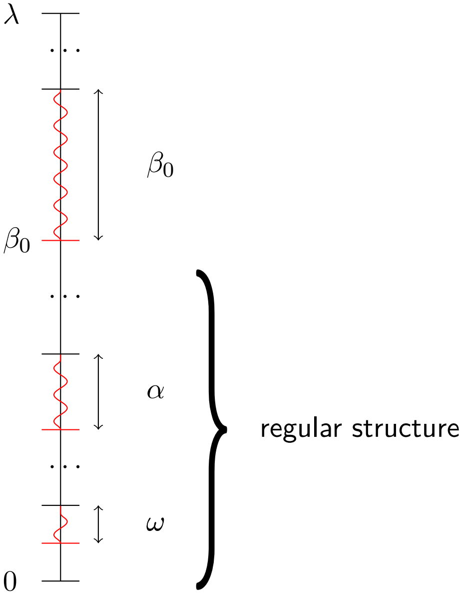
Proof of gap existence

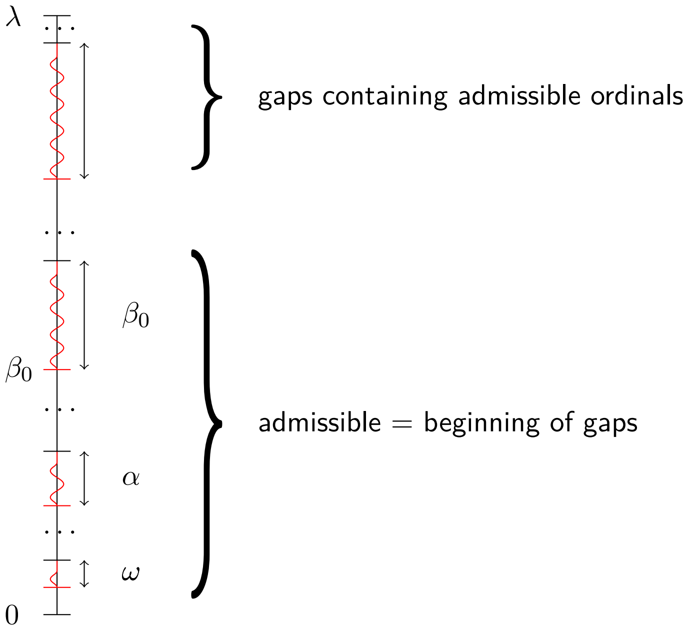
But ...does the algorithm halt?

Halting of the algorithm, proof by contradiction:

- Above λ, by definition, there are no clockable ordinals.
- If no gaps before λ, thus beginnning of gap detected **at** λ.
- **Contradiction**.

What can we say about gaps?


λ
β_0
β_0
α
ω
0
regular structure


λ
β0
α
ω
0
gaps containing admissible ordinals
admissible = beginning of gaps

infinite time Turing machines

=

model for algorithms proving logical properties

Thank you for your attention.

Some references:

Joel D. Hamkins and Andrew Lewis.
Infinite time turing machines.
Journal of Symbolic Logic, 65(2):567–604, 2000.

Philip D. Welch.
Characteristics of discrete transfinite time turing machine models: Halting times, stabilization times, and normal form theorems.
Theoretical Computer Science, 410(4-5):426–442, 2009.

Computational power of ITTM

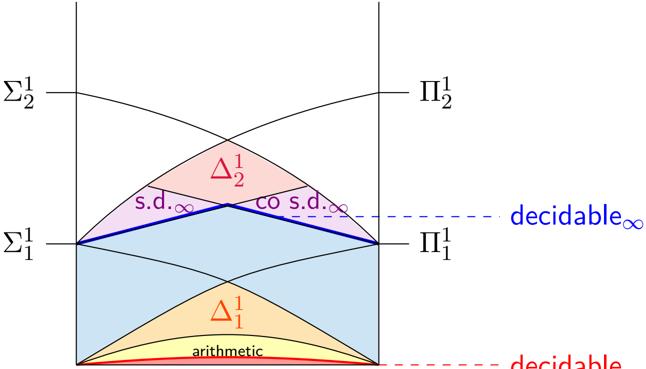


Figure: Projective hierarchy

Admissible ordinals

Proposition 3.

A limit ordinal α is admissible if and only if there **doesn't exist** a function f from $\gamma < \alpha$ to α such that:

- f is unbounded (no greatest element in α) and
- f is Σ_1-definable in L_α.

Constructible hierarchy

Definition (Constructible hierarchy L)**.**

- $L_0 = \emptyset$;
- $L_{\alpha+1} = def(L_\alpha)$;
- if α is a limit ordinal, $L_\alpha = \bigcup_{\beta<\alpha} L_\beta$;

Application: reals of L_λ are the writable reals.

Definability

Let M be a set and F be the set of the formulas of the language $\{\in\}$.

Definition (Definability).

X is definable on a model $(M, \in)$ if:

- there exists a formula $\varphi \in F$,
- there exists $a_1, \ldots, a_n \in M$

such that $X = \{x \in M : \varphi(x, a_1, \ldots, a_n)$ is true in $(M, \in)\}$.

$def(M) = \{X \subset M : X$ is definable on $(M, \in)\}$.

Admissible ordinals

Proposition 4.

A limit ordinal α is admissible if and only if there **doesn't exist** a function f from $\gamma < \alpha$ to α such that:

- f is unbounded (no greatest element in α) and
- f is Σ_1-definable in L_α.