Drawing numerable linear orderings

Jérôme Durand-Lose

LABORATOIRE D'INFORMATIQUE FONDAMENTALE D'ORLEANS

Laboratoire d'Informatique Fondamentale d'Orléans
ÉA 4022
Université d'Orléans, Orléans, FRANCE

Funded by **ANR ∂IFFERENCE (ANR-20-CE40-0002)**

Université d'Orléans

Numerable linear orderings $(S, \leqslant_S)$

- S is a numerable set
- $\leqslant_S$ is a reflexive, anti-symmetric and transitive relation on S
- any two elements of S are related/comparable
- examples:
  - ω : the ordering of the natural numbers $(0, 1, 2, \cdots)$
  - ζ : the ordering of the integers $(\cdots, -2, -1, 0, 1, 2, \cdots)$
- operations
  - addition: set union and all the elements from a set before the elements from the other
  - product: set product and with lexicographical order
  - $\{a, b\}^*$ with lexical order
- examples:
  - $\omega + \omega$ is $0, 1, 2, \cdots 0', 1', 2', \cdots$ ($0'$ is a distinct copy of 0)
  - $\omega \cdot \omega$ is $(0, 0), (0, 1), (0, 2), \cdots$ $(1, 0), (1, 1), (1, 2), \cdots$ $(2, 0), (2, 1), (2, 2), \cdots$

Ordinals

- well founded orderings
- examples:
  - $\omega, \quad \omega+\omega, \quad \omega+\omega+\omega, \quad \omega\cdot\omega, \quad \omega\cdot\omega\cdot\omega+\omega\cdot\omega+\omega$
- non-ordinal linear orderings:
  - ζ has the infinite decreasing sequence $(\cdots, -2, -1, 0)$
  - $\{a, b\}^*$ with lexical order has the infinite decreasing sequence $(\cdots, aab, ab, b)$

- More on linear orderings ⇝ Rosenstein [1982]

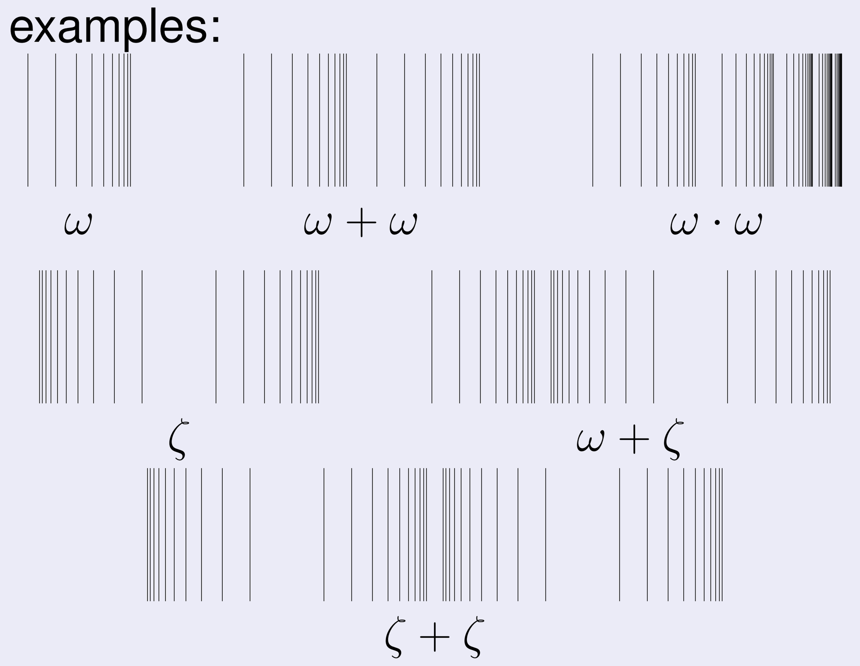
Decidable

- $s_0, s_1, s_2, \cdots$: an enumeration of the elements of S
- A Turing machine decides the relation

$$i, j \mapsto s_i \leqslant_S s_j$$

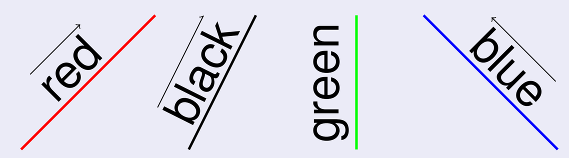
Graphical representation

- by parallel verticals lines s.t.:
  - one line per element in S
  - if $x \leqslant_S y$ then x is on the left of y
  - there is an empty space between x and y ($x \leqslant_S y$) iff x is the immediate predecessor y iff y is the immediate successor of x iff $\forall z, x \leqslant_S z \leqslant_S y \quad \rightarrow \quad x = z \vee z = y$
- examples:



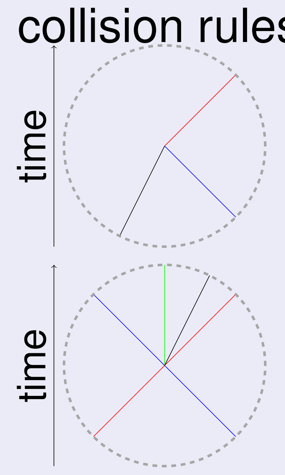
Signal Machines

- signals



constant speed for each kind

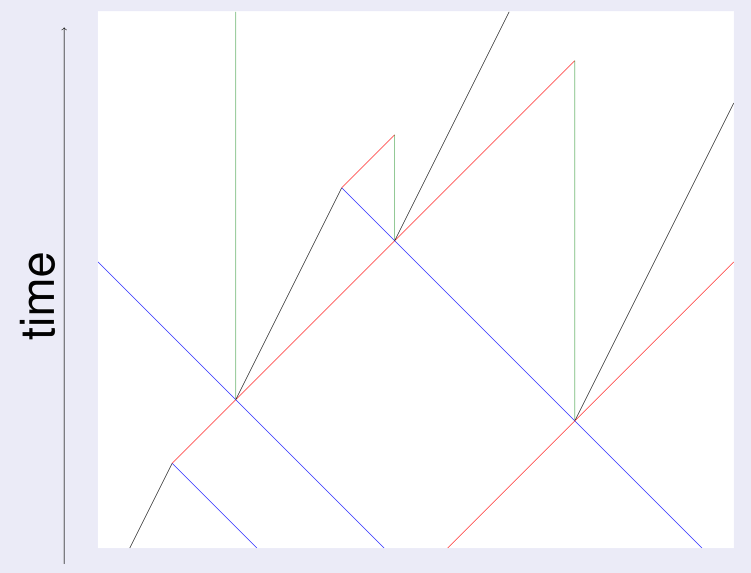
- collision rules



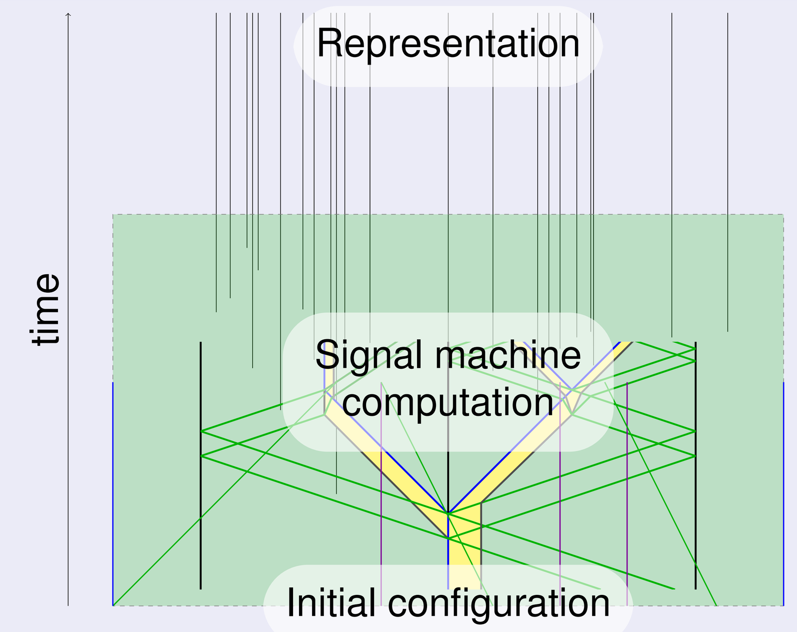
$\{\overline{\text{black}}, \overline{\text{blue}}\} \rightarrow \{\overline{\text{red}}\}$

$\{\overline{\text{red}}, \overline{\text{blue}}\} \rightarrow \{\overline{\text{blue}}, \text{green}, \overline{\text{black}}, \overline{\text{red}}\}$

- dynamics ⇝ space-time diagrams

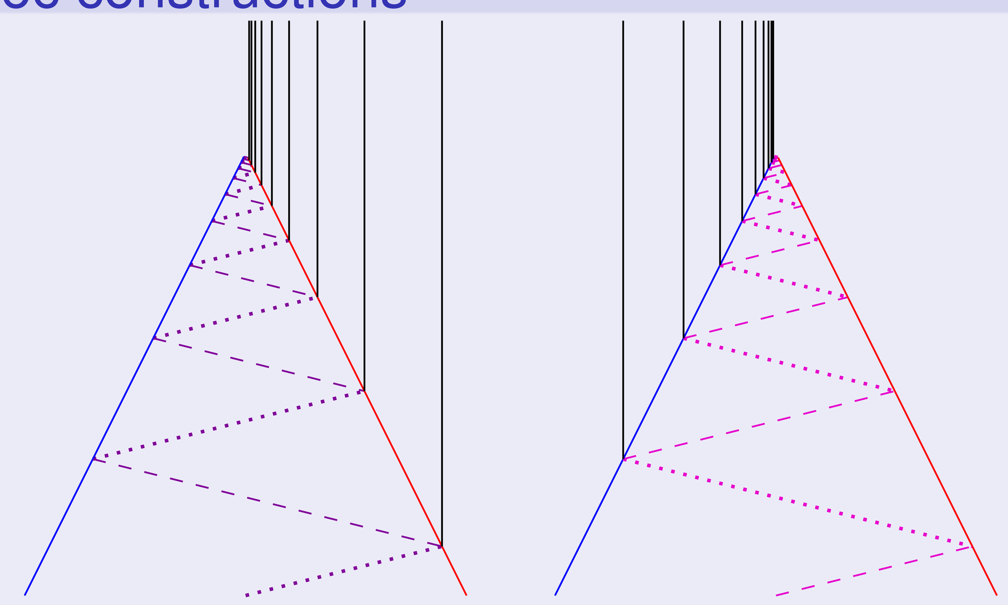


Global scheme



Ad hoc constructions

- ζ



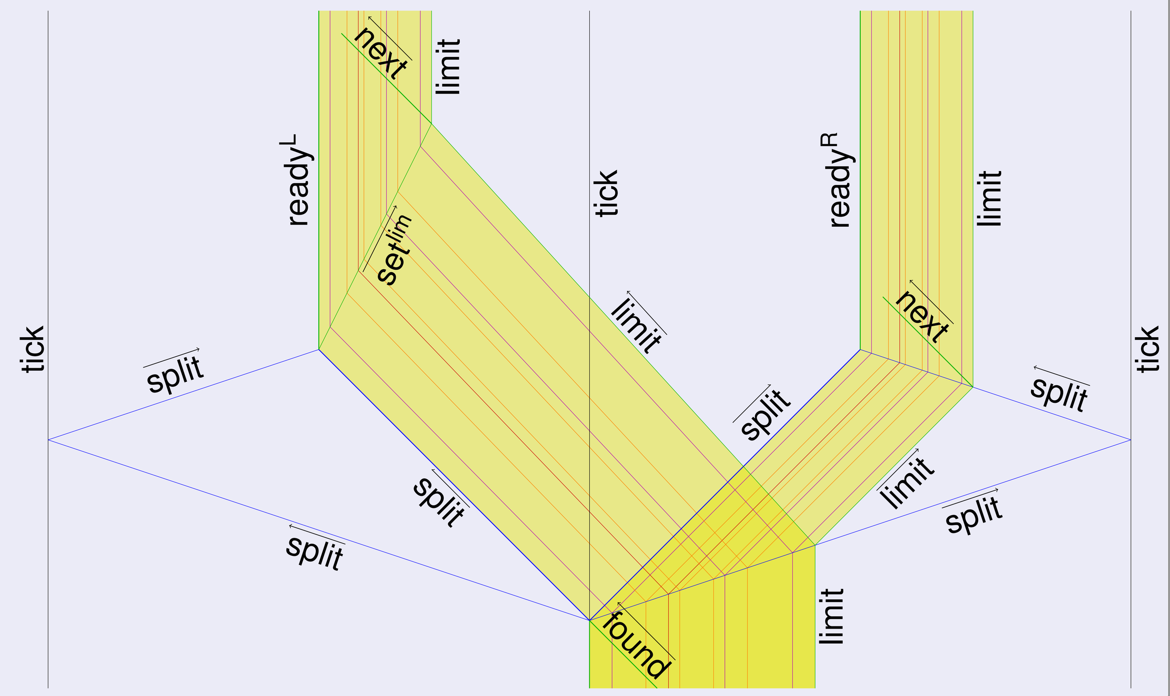
Algorithm

- let i, j be two indices s.t. lines are set
- find least k such that:

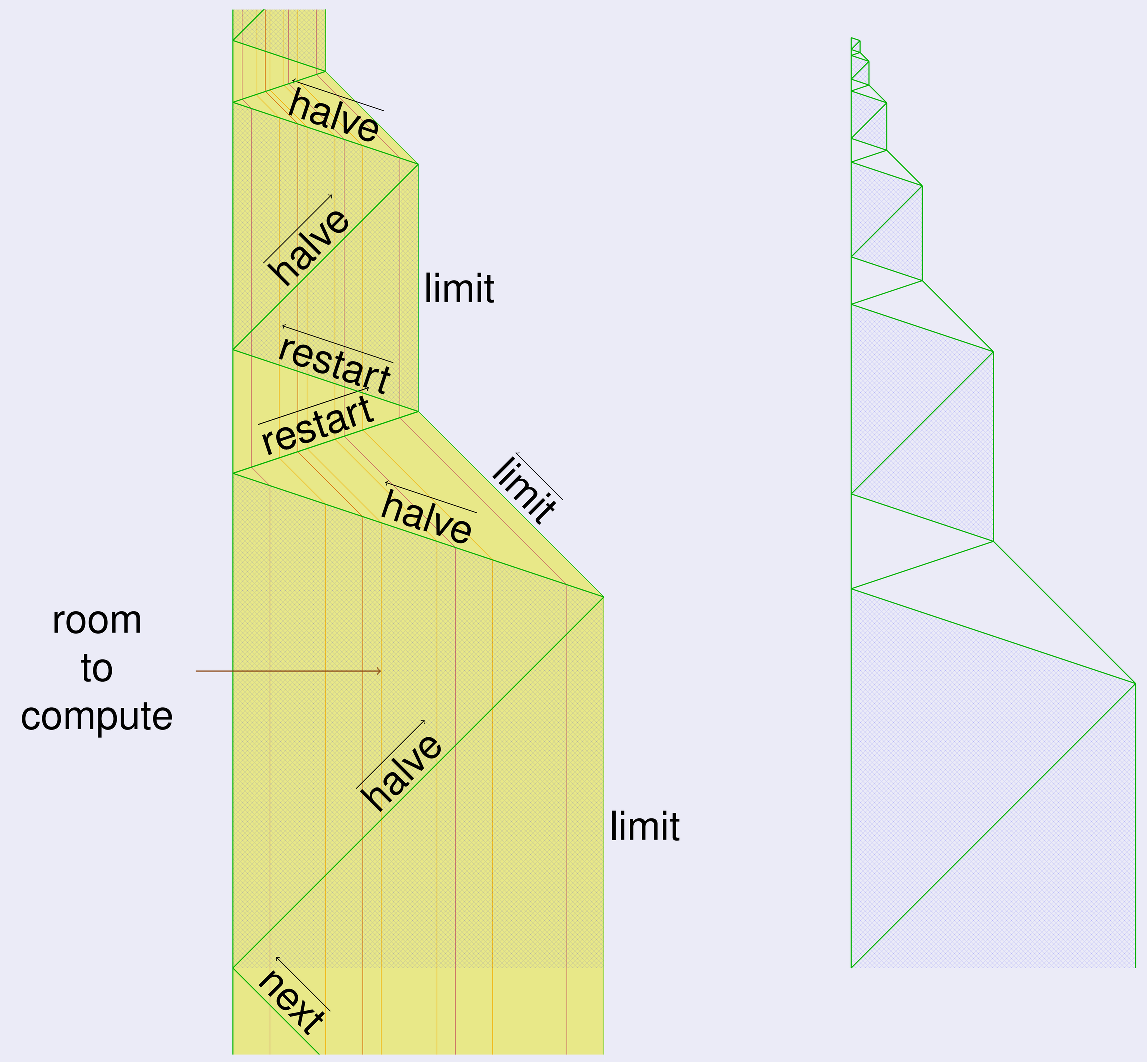
$$k \neq i \wedge k \neq j \wedge s_i \leqslant_S s_k \leqslant_S s_j$$

- if k is found, the interval is split and the algorithm is restarted on i and k on the left and k and j on the right
- if no such k exists, the computation vanishes
- this is done in finite time

Splitting

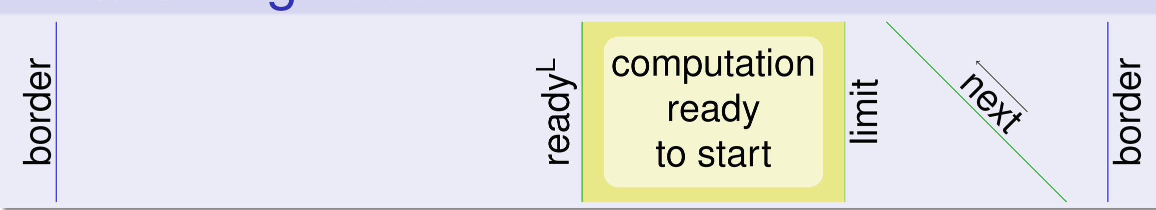


Scaling the computation at each step



- completion ensured in finite duration

Initial configuration

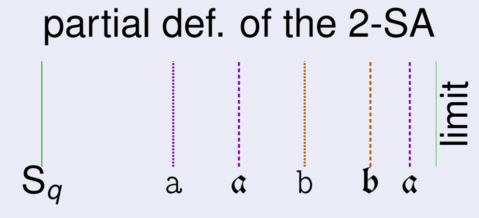


Working on $\mathbb{Q}$

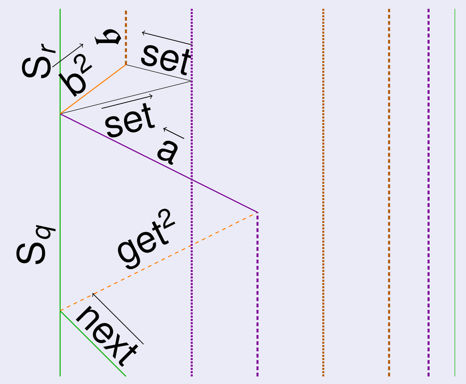
There is a rational signal machine able to generate the representation of any decidable countable linear ordering.

- Simulating multi-stack automaton

$p(q) = 2$
$t(q, \mathtt{a}) = (\mathtt{b}, 2)$
$\delta(q, \mathtt{a}) = r$
partial def. of the 2-SA



encoding of $(q, \mathtt{ab}, \mathtt{aba})$

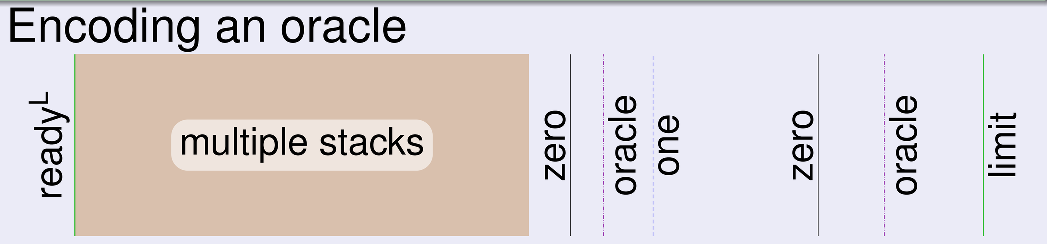


one transition step

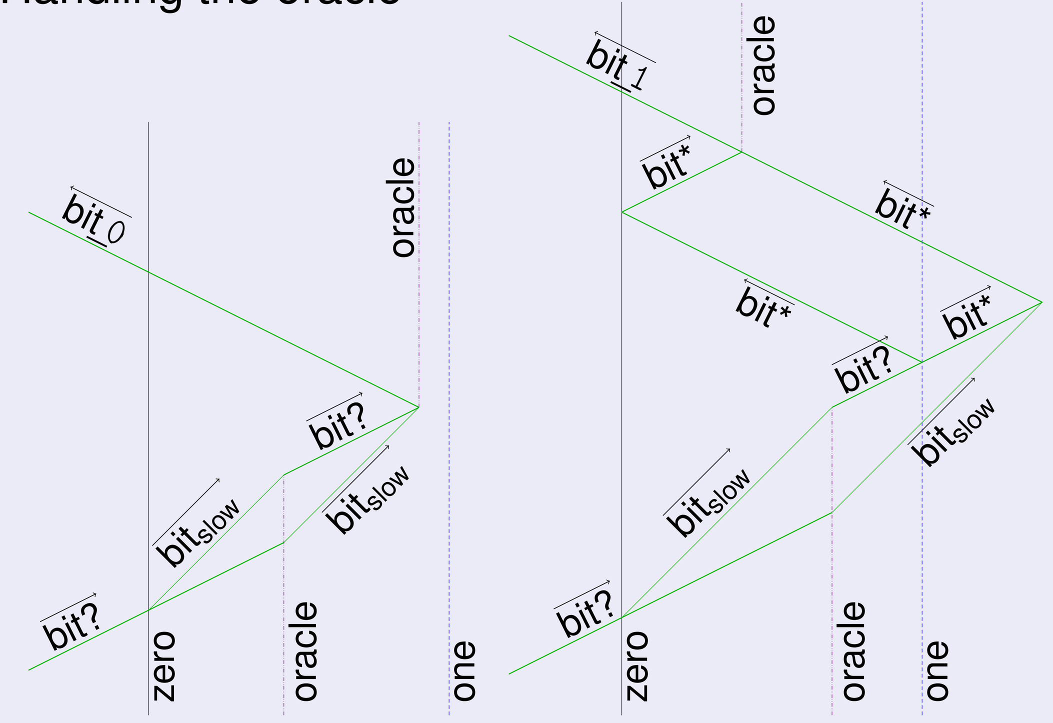
Working on $\mathbb{R}$

There is a signal machine able to generate the representation of any countable linear ordering.

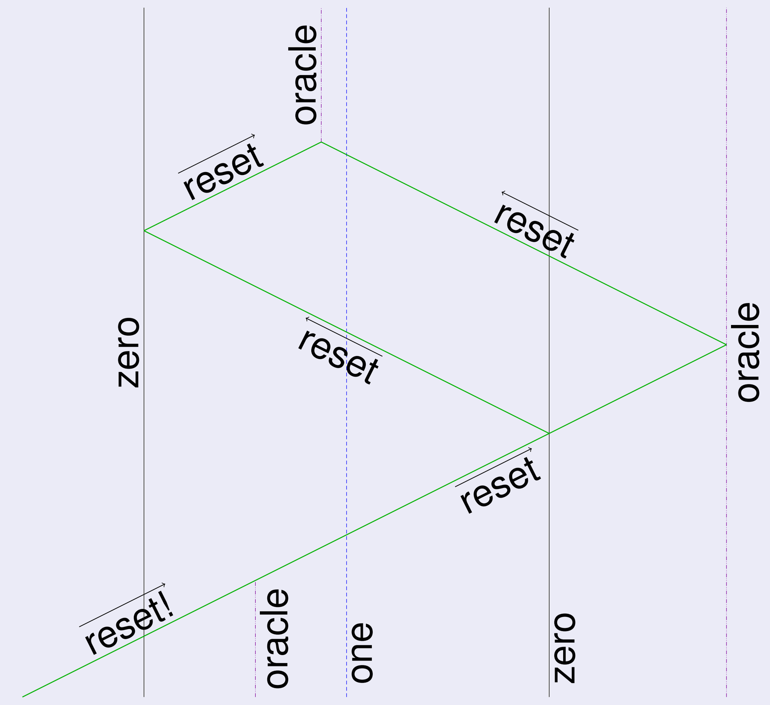
- Encoding an oracle



- Handling the oracle



(a) next bit (case 0) (b) next bit (case 1)



(c) resetting the oracle

References

J. Durand-Lose. Ways to Compute in Euclidean Frameworks. In M. J. Patitz and M. Stannett, editors, *Unconventional Computation and Natural Computation - 16th Int. Conf., UCNC 2017, Fayetteville, AR*, volume 10240 of *LNCS*, pages 8–25. Springer, 2017. doi: 10.1007/978-3-319-58187-3_2.

J. G. Rosenstein. *Linear ordering*. Academic Press, 1982.