



HAL
open science

Abstract geometrical computation 7: Geometrical accumulations and computably enumerable real numbers

Jérôme Durand-Lose

► **To cite this version:**

Jérôme Durand-Lose. Abstract geometrical computation 7: Geometrical accumulations and computably enumerable real numbers. *Natural Computing*, 2012, 11 (4), pp.609-622. 10.1007/s11047-012-9335-8 . hal-00691466

HAL Id: hal-00691466

<https://hal.science/hal-00691466>

Submitted on 7 Nov 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Abstract geometrical computation 7: Geometrical accumulations and computably enumerable real numbers

Jérôme Durand-Lose*[†]
LIFO, Université d'Orléans,
B.P. 6759, F-45067 ORLÉANS Cedex 2.

August 8, 2012

Abstract

Using rules to automatically extend a drawing on an Euclidean space might lead to accumulating drawings into a single point. Such points are characterized in the context of Abstract geometrical computation.

Colored line segments (traces of signals) are drawn according to rules: signals with similar color are parallel and when they intersect, they are replaced according to their colors. Time and space are continuous and accumulations can happen. Constructions exist to unboundedly accelerate a computation and provide, in a finite duration, exact analog values as limits/accumulations.

Starting with rational numbers for coordinates and speeds, the time of any isolated accumulation is a *c.e.* (*computably enumerable*) real number. There is a signal machine and an initial configuration that accumulates at any *c.e.* time. Similarly, the spatial positions of isolated accumulations are exactly the *d-c.e.* (*difference of computably enumerable*) numbers. Moreover, there is a signal machine that can accumulate at any *c.e.* time or *d-c.e.* position depending only on the initial configuration.

These existence results rely on a two-level construction: an inner structure simulates a Turing machine that output orders to the outer structure which handles the accumulation.

*<http://www.univ-orleans.fr/lifo/Members/Jerome.Durand-Lose>,
Lose@univ-orleans.fr

Jerome.Durand-

[†]This work was partially funded by the ANR project AGAPE, ANR-09-BLAN-0159-03.

Key-words.

Abstract geometrical computations; Computable analysis; Geometrical accumulations; c.e. and d-c.e. real numbers; Signal machine.

1 Introduction

Starting from a few aligned points, lines are initiated. When they intersect, they end and new line segments start. Each segment is given a color and segments with the same color should be parallel. The new segments are colored according to the colors of the removed ones.

Segments extension is made so as to always go further away from the initial alignment line. All the updates (lengthening and replacing) are done on a line going away from the initial one. So that there is some notion of past and future.

What can kind of figure can one build with finitely many colors? Could this system compute in any way?

Indeed such a system computes. It does in the understandings of both *Turing* computability (Durand-Lose, 2005), the original *Blum, Shub and Smale* model (Blum et al., 1989; Durand-Lose, 2007, 2008a) and *Computable analysis* (Weihrauch, 2000; Durand-Lose, 2009b, 2011a). The so-called *Black-hole model* of computation can be embedded too (Etesi and Némethi, 2002; Hogarth, 2004; Lloyd and Ng, 2004; Andréka et al., 2009; Durand-Lose, 2006a, 2009a).

Given that the underlying space and time are Euclidean, thus continuous, can there be any accumulation? What can be said about them?

Geometrical accumulation is a common phenomenon (as in Fig. 1; time/propagation is always elapsing upward). With a shift and a rescaling, it could happen anywhere. It is the key to embedding analog computing as well as the Black-hole model. As shown in Fig. 1(b), accumulation is not always limited to one point. In this example, it accumulates on a whole line. Stranger patterns also exist as illustrated in Fig. 2.

In the present article, we show that if the system is based on rational numbers then the temporal and spatial coordinates of any isolated accumulation belong to some particular sets of real numbers. The times are exactly the *computably enumerable numbers* (c.e. numbers for short): the limits of (converging) increasing computable sequences of rational numbers. The spatial positions are exactly the differences of two such numbers (*d-c.e.* numbers).

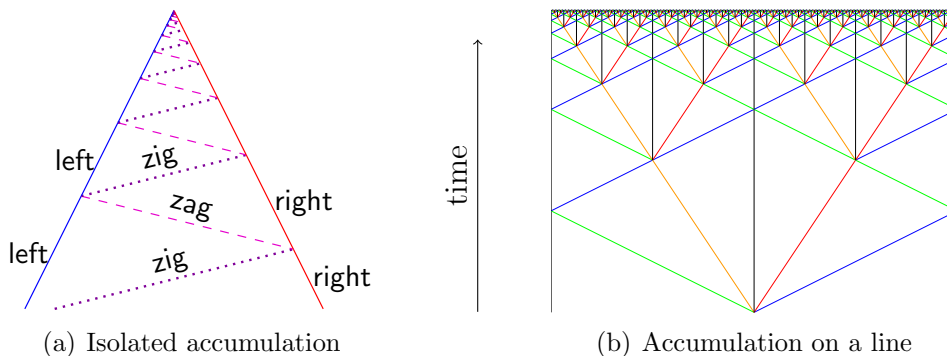


Figure 1: Space-time diagrams with accumulations.

Usual *computable* numbers (limits of effectively converging computable sequences of rational) are a strict subset of *c.e.* numbers which is a strict subset of *d-c.e.* numbers (Zheng, 2006).

The geometric system described above is a *signal machine* in the context of *abstract geometrical computations*. It is inspired by a continuous time and space counterpart of cellular automata emphasizing on the notion of signal which is the key to CA dynamics (Durand-Lose, 2008b) and is somehow related to the approaches of Takeuti (2005) and Hagiya (2005). Geometrically, it is related to the “colored spaces” or “Mondrian spaces” of Jacopini and Sontacchi (1990) which involves polyhedra in Euclidean spaces with a distinguished direction for time. It can also be understood as an idealization of collision computing (Adamatzky, 2002; Adamatzky and Durand-Lose, 2012).

Another important way to construct in an Euclidean space is the compass and straightedge approach. This provides the primitives for the Geometric machines of Hucckenbeck (1989, 1991). The corresponding set of constructible points is not easy to express and goes far beyond the intended scope of the present article.

A *signal machine* gathers the definition of meta-signals (*colors*, like **zig** and **right** in Fig. 1(a)) and collision rules (like $\{\text{zig}, \text{right}\} \rightarrow \{\text{zag}, \text{right}\}$). An instance of a meta-signal is a dimensionless point called a *signal*. Each signal moves uniformly, its speed only depends on the associated meta-signal. The traces of signals on the space-time diagrams form line segments and as soon as they correspond to the same meta-signal, they are parallel. When signals meet, they are removed and replaced by new signals. The emitted signals only depend on the nature of colliding ones.

One key feature of AGC is that space and time are continuous. This has been used to do fractal parallelism (Duchier et al., 2010, 2012). More-

over, Zeno effects can be implemented to generate unbounded acceleration; in particular to allow infinitely many discrete transitions during a finite duration. This has been used to decide the halting problem and to implement the Black-hole model (Durand-Lose, 2009a). It has also been used to carry out exact analog computations (Durand-Lose, 2008a, 2009b).

This is achieved with rational signal machines: speeds as well as initial positions are rational numbers. Since the positions of collisions are defined by linear equations in rational numbers, the collisions all happen at rational positions. This is important since rational numbers, and thus signal machines, can be handled exactly in classical discrete computability.

One early question in the field was whether, starting from a rational signal machine, accumulation could lead to an irrational coordinate. An accumulation at $\sqrt{2}$ was provided in Durand-Lose (2007). The question became to characterize all the possible accumulation points. Please note that forecasting accumulation for a rational signal machine is as undecidable as the strict partiality of a computable function: it is Σ_2^0 -complete in the arithmetical hierarchy (Durand-Lose, 2006b).

The present article focuses on *isolated accumulations*: in the space-time diagram, sufficiently close to it, there is no accumulation point and nothing except in the casual past—or backward light cone—as in Fig. 1(a). The accumulation on Fig. 2(a) is not isolated because of infinitely many left signals on the right. Please note that there are also accumulation of left signals on the left at the top of the space-time diagram. These accumulations are *static* and happen away from any collision.

The accumulations in Fig. 2(b) form a cantor set. The ones in Fig. 2(c) are on a curve (right upper limit) and are almost all accumulations of signals away from any collision: they are the above limit of infinitely many signals.

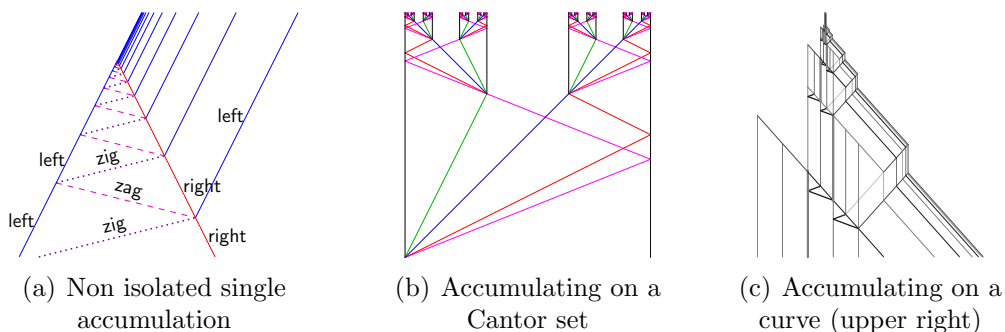


Figure 2: Non-isolated accumulations.

Rational signal machines can be simulated exactly on a computer. The

time of an accumulation is thus a *c.e.* number since it is the above limit of a computable Zeno phenomenon. For the spatial position, by slanting the space-time diagram, an increasing sequence is exhibited. The drifting correction provides a negative *c.e.* term. So that the spatial location is a *d-c.e.* number as the difference of two *c.e.* numbers.

Proving that indeed any *c.e.* (resp. *d-c.e.*) number can be the time (resp. place) of an accumulation is much more involving. To achieve this, a two-level construction scheme is used. It is similar in spirit to the one used in Durand-Lose (2009b) to implement Computable analysis. The control structure simulates a Turing machine that output orders to the outer structure. It is composed of the control (simulation) embedded inside an accelerating structure (inner structure) to ensure that the output is delivered in a bounded time. The outer structure undergoes a shrinking process (generating the accumulation). Before performing any shrinking step, it makes an elemental movement according to the received orders. Specially designed control and outer structures provide (rational) signal machines and initial configurations that accumulate at any *c.e.* time (resp. *d-c.e.* spatial position).

This article is the full version of Durand-Lose (2011b) which goes beyond Durand-Lose (2010) which ended up having a major flaw: the accumulation positions do not need to be computable as is proven here by accessing all *c.e.*

Definitions are gathered in Sect.2. Section 3 shows that the temporal (resp. spatial) coordinate of isolated accumulations is always *c.e.* (resp. *d-c.e.*). Section 4 presents the multilevel architecture uses as well as some naming conventions. Section 5 presents the control part: a simulation of a Turing machine with input and output for *d-c.e.* numbers. Section 6 presents the inner structure with unbounded acceleration. Section 7 presents what is to be expected from the outer structure. Sections 8 and 9 present outer structures to accumulate respectively at a *c.e.* time and at a *d-c.e.* spatial position. Section 10 concludes the article.

2 Definitions

2.1 Abstract geometrical computation

A *signal machine* collects the definitions of available meta-signals, their speeds (positive for rightward signals and negative for leftward ones) and the collision rules. For example, the machine to generate Fig.1(a) is composed of the following meta-signals (with speed): **left** ($\frac{1}{2}$), **zig** (4), **zag** (-4),

and right ($-\frac{1}{2}$). Two collision rules are defined:

$$\{\text{left, zag}\} \longrightarrow \{\text{left, zig}\} \quad \text{and} \quad \{\text{zig, right}\} \longrightarrow \{\text{zag, right}\} .$$

It might happen that exactly three (or more) meta-signals meet. In such a case, collisions rules involving three (or more) meta-signals are used. There can be any number of meta-signals in the range of a collision rule, as long as their speeds differ (i.e. they are not parallel).

Definition 1 (Signal machine) A *Signal machine* is defined by (M, S, R) where:

1. M is a finite set of *meta-signals*,
2. S is a function from meta-signals to real numbers, it assigns *speeds* to meta-signals, and
3. R is a deterministic set of *collision rules*. A collision rule is written $\rho = \rho^- \rightarrow \rho^+$ where ρ^- and ρ^+ are sets of meta-signals of all different speeds, and ρ^- must have at least two meta-signals. R is deterministic: $\rho \neq \rho'$ implies that $\rho^- \neq \rho'^-$. When appropriate, ρ can be considered as a partial function, a rule is then written $R(\rho^-) = \rho^+$.

A *configuration* is a function from the real line (space) into the set of meta-signals and collision rules plus two extra values: \emptyset (for nothing there) and \ast (for accumulation). Any signal or collision must be spatially isolated: there is nothing else but \emptyset arbitrarily closed. The accumulation points of non- \emptyset locations must be \ast . These are spatial, *static*, accumulations like on top of figures 2(a) and 2(c).

Definition 2 (Configuration) A *configuration*, c , is a function from the real line into meta-signals, rules, \emptyset and \ast (let $V = M \cup R \cup \{\emptyset, \ast\}$ so that $c : \mathbb{R} \rightarrow V$) such that:

1. all signals and collisions are isolated: $\forall x, c(x) \in M \cup R$ implies that $\exists \varepsilon, 0 < \varepsilon$ and $\forall y, 0 < |x - y| < \varepsilon, c(y) = \emptyset$, and
2. spatial accumulation are marked accordingly: any x that is an accumulation point of $c^{-1}(V \setminus \{\emptyset\})$ verifies $c(x) = \ast$ (x in \mathbb{R} is an accumulation point of a subset E of \mathbb{R} iff $\forall \varepsilon, 1 < |E \cap (x - \varepsilon, x + \varepsilon)|$).

The first condition implies that there is at most countably many signals and collisions on the line. In the rest of the paper it is always considered

that this number is finite since we are dealing with isolated accumulations. The last condition implies that \circlearrowleft -valued locations form an open set.

If there is a signal of speed s at x , then, unless there is a collision before, after a duration Δt , its position is $x + s \cdot \Delta t$. At a collision, all incoming signals are immediately replaced by outgoing signals in the following configurations according to collision rules.

Definition 3 (Dynamics) Considering a configuration, c , the *time to the next collision*, $\Delta(c)$, is equal to the minimum of the positive real numbers d such that:

$$\exists x_1, x_2 \in \mathbb{R}, \exists \mu_1, \mu_2 \in M \left\{ \begin{array}{l} x_1 + d \cdot S(\mu_1) = x_2 + d \cdot S(\mu_2) \text{ ,} \\ c(x_1) = \mu_1 \vee (c(x_1) = \rho^- \rightarrow \rho^+ \wedge \mu_1 \in \rho^+) \text{ ,} \\ c(x_2) = \mu_2 \vee (c(x_2) = \rho^- \rightarrow \rho^+ \wedge \mu_2 \in \rho^+) \text{ .} \end{array} \right.$$

It is $+\infty$ if there is no such d .

Let c_t the configuration at time t , for t' between t and $t + \Delta(c_t)$, the configuration at t' is defined as follows. First, signals are set according to $c_{t'}(x') = \mu$ iff $c_t(x) = \mu \vee (c_t(x) = \rho^- \rightarrow \rho^+ \wedge \mu \in \rho^+)$ where $x = x' + (t - t') \cdot S(\mu)$. There is no collision to set (t' is before the next collision). Then (static) accumulations are set: $c_{t'}(x') = *$ iff x' is an accumulation point of $c_{t'}^{-1}(M)$. It is \circlearrowleft everywhere else.

For the configuration at $t' = t + \Delta(c_t)$, collisions are set first: $c_{t'}(x') = \rho^- \rightarrow \rho^+$ iff for all $\mu \in \rho^-$, $c_t(x_\mu) = \mu \vee (c_t(x_\mu) = \rho^- \rightarrow \rho^+ \wedge \mu \in \rho^+)$ where $x_\mu = x' + (t - t') \cdot S(\mu)$. Then meta-signals are set (where there is not already a collision), then (static) accumulations.

The *sequence of collision times* is defined by: $t_0 = 0$, $t_{n+1} = t_n + \Delta(c_{t_n})$. This sequence is finite if there is an n such that $\Delta(c_{t_n}) = +\infty$. Otherwise, since it is non-decreasing, it admits a limit. If the sequence is finite or its limit is infinite, then the whole space-time diagram is defined. These cases are of no interest here since there is no non-static accumulation.

Only the last case is considered from now on: there is a finite limit, say \tilde{t} . The configuration at \tilde{t} is defined as follows. First (dynamic) accumulations are set: $c_{\tilde{t}}(x) = *$ iff $\forall \varepsilon, 0 < \varepsilon$ then there exists x' and t' such that $|x - x'| < \varepsilon$, $\tilde{t} - \varepsilon < t' < \tilde{t}$ and $c_{t'}(x') \in R$. Then collisions are set: $c_{\tilde{t}}(x) = \rho^- \rightarrow \rho^+$ iff for all $\mu \in \rho^-$, $\exists \varepsilon, \forall \varepsilon', 0 < \varepsilon' < \varepsilon$, holds $c_{\tilde{t}-\varepsilon'}(x' - \varepsilon' \cdot S(\mu)) = \mu$. Then meta-signals are set: $c_{\tilde{t}}(x) = \mu$ iff $\exists \varepsilon, \forall \varepsilon', 0 < \varepsilon' < \varepsilon$, then $c_{\tilde{t}-\varepsilon'}(x' - \varepsilon' \cdot S(\mu)) = \mu$. Finally, static accumulations are set.

Each time, a position is set only if it is not already set. At the end every unset position is assigned \circlearrowleft . Please note that this definition does not always define an extension to the computation (when there are infinitely

many signals, $\Delta(c)$ is an infimum that could be equal to zero), nevertheless it does in the cases considered here.

A *space-time diagram* is the collection of consecutive configurations which forms a two dimensional picture. Figure 3(a) provides a space-time diagram with three signals in the initial configuration (normal, thick and dashed) and three collisions at time t_1 , t_2 and t_3 (little circles). The same meta-signals are involved in the first and third collisions (normal and thick) so that the same meta-signals (thick, normal and dotted) emerges. Collisions are deterministic.

Definition 4 (Casual past and isolated accumulation) Let S_{max} and S_{min} (maximal left speed) be the maximum and minimum values taken by the speed function S . The value at position (x, t) in the space-time diagram only depends on the values at the position on the *casual past* or *backward light cone*:

$$I^-(x, t) = \left\{ (x', t') \mid t' < t \wedge S_{max} \cdot (t' - t) < x' - x < S_{min} \cdot (t' - t) \right\} .$$

This is illustrated in Fig. 3(b). An accumulation at (x_0, t_0) is *isolated* if, sufficiently close to (x_0, t_0) :

- there is nothing but \emptyset out of the casual past, and
- there are infinitely many signals and collisions but no accumulation in the casual past.

It is a purely dynamical (and local) accumulation.

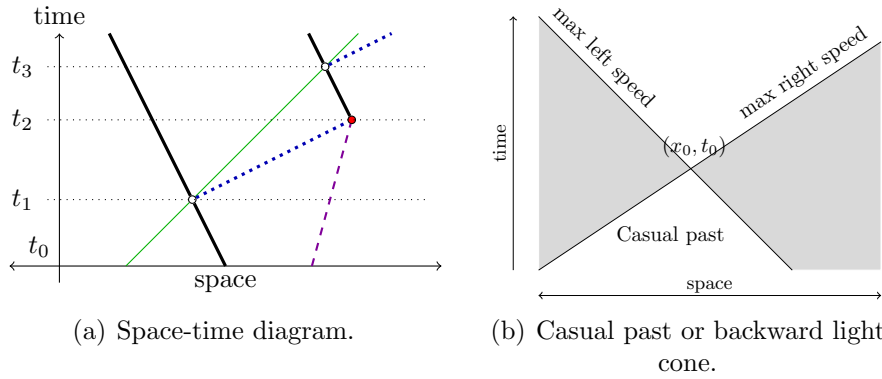


Figure 3: Example of a space-time diagram and casual past.

Definition 5 (Rational signal machine) A signal machine is *rational* if all speeds are rational numbers and non- \emptyset positions in any initial configuration must also be rational numbers.

Since the position of collisions are solutions of systems of rational linear equations, they are rational. In any space-time diagram of a rational signal machine, as long as there is no accumulation, the coordinates of all collisions are rational.

The dynamics is uniform in both space and time. Space and time are continuous; there is no absolute scale. So that if the initial configuration is shifted or scaled so is the whole space-time diagram.

2.2 *c.e.* and *d-c.e.* real numbers

A *computable sequence* is defined by a Turing machine (or any equivalent model of computation) that on input n outputs the n th term of the sequence. Equivalently, it is also defined by a Turing machine that outputs the infinite sequence on a write-once tape in infinite time. A rational number may be written in any *reasonable* encoding.

Computable real numbers are the limits of computable real sequences of rational numbers such that the precision is known and decreases to zero (e.g. $|x - x_n| < 2^{-n}$). Such a convergence is called *effective*.

Definition 6 (*c.e.* and *d-c.e.* numbers) A real number is *c.e.* (*computably enumerable*) if there is an increasing computable sequence of rational numbers that converges to it.

A real number is *d-c.e.* (*difference of computably enumerable*) if it is the difference of two *c.e.* numbers.

An effective sequence can be turned into an effective increasing sequence by considering at each stage the least possible value for the approximation considering previously output approximations. But computable increasing converging sequences cannot be turned into effective one since there is no information on the precision. To understand consider the Chaitin's constant Ω equals to the sum of 2^{-n} over n 's such that the n th Turing machine stops on the empty entry. To built a computable increasing sequence that converges to it, it is enough to enumerate \mathbb{N}^2 considering each entry as a pair: (number of Turing machine n , number of iterations m) and add 2^{-n} each time the n th Turing machine stops on the empty entry in exactly m iterations. If there were any effective sequence converging to Ω , the halting problem would be decidable.

On the one side, the *c.e.* numbers are closed by rational addition and positive rational multiplication but they not closed under subtraction. If it were closed by subtraction, then $-\Omega$ would be the limit of a computable increasing sequence, so that Ω would also be the limit of a decreasing sequence.

Since it is possible to construct an effective sequence from an increasing and a decreasing ones, Ω would be the limit of an effective sequence.

On the other side *d-c.e.* numbers form a closed field (Ambos-Spies et al., 2000) and are also characterized by:

Theorem 7 (Ambos-Spies et al. (2000)) *A real number is d-c.e. iff there is a computable sequence (x_n) that weakly effectively converges to it in the sense that the sum $\sum_{n \in \mathbb{N}} |x_{n+1} - x_n|$ converges.*

This representation is used for defining the control in Subsect. 5.1 and for the spatial constructions in Sect. 9.

3 Only (*d-*)*c.e.* coordinates

Let us consider any isolated accumulation at (x_0, t_0) on a rational signal machine. The configuration is “clipped” sufficiently closed to the accumulation so that there is nothing out of the casual past. It is rational and finite.

From a (rational) signal machine and a (finite) configuration, it is straightforward to build a Turing machine that treats the collisions and updates the configuration forever. This follows the dynamics at collision times as defined in Def. 3, the function $\Delta(\cdot)$ is easy to compute since there are finitely many signals and computations with rational numbers are made with exact precision. (Indeed this has been programmed in java to generate the illustrations.)

Let modify the machine so that it outputs the sequence of the (rational) time of collisions. This (computable) sequence is increasing and converges to the time of the accumulation.

Lemma 8 *The time of any (rational) isolated accumulation is c.e.*

A space-time diagram can be slanted by adding the same *drift* to all signals. This is done by increasing all their speeds of the meta-signals by the same amount. For example, starting from Fig. 1(a), by adding 1, 2 and 4 to all the speeds, the diagrams of Fig. 4 are generated.

Let d be any integer, the d -drift of a signal machine A is a signal machine \hat{A} defined like A but all the speeds are incremented by d , i.e. $\hat{S}(\mu) = S(\mu) + d$. For any A -configuration c , the corresponding \hat{A} -configuration \hat{c} is identical but the dynamics is drifting: $\hat{c}_t(c) = c_t(c - d \cdot t)$ (proof by induction on t_n).

With a sufficiently large integer drift, all speeds become positive, so that the configuration has to move to the right, or at least, the position of its leftmost signal does. Consider a modification of the above Turing machine so that each collision time, it outputs the spatial position of the leftmost

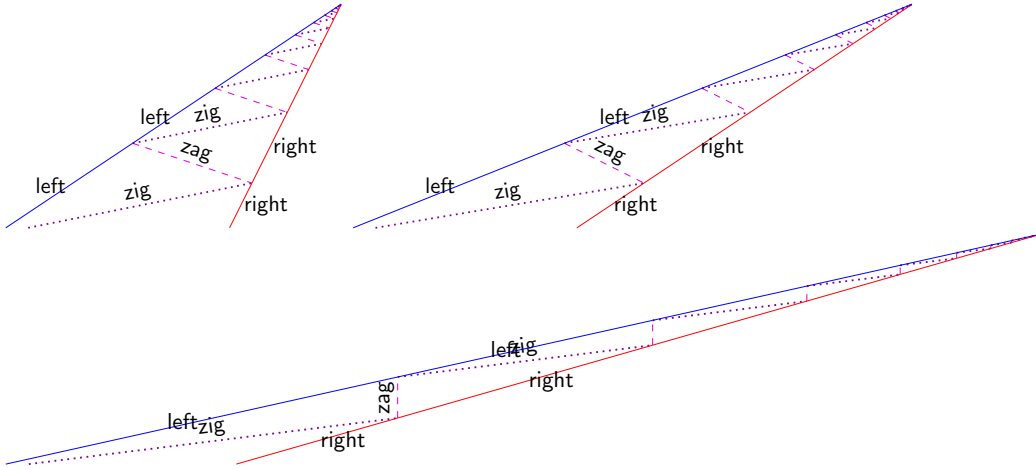


Figure 4: Examples of drifts by 1, 2 and 4.

signal. This produces an increasing sequence that converges to the spatial position of the drifted accumulation. This position, \hat{x}_0 , is *c.e.*

While the times remain unchanged, the spatial positions of signals and collisions (and hence accumulation) are moved by $d \cdot t$ where d is the drift and t is the time. To correct the drift for the accumulation and recover x_0 , the spatial position in the original dynamics, $d \cdot t_0$ has to be removed where t_0 is the time of the accumulation. Since t_0 is *c.e.* and d is an integer, $d \cdot t_0$ is *c.e.* So that $x_0 = \hat{x}_0 - d \cdot t_0$ is *d-c.e.*

With a sufficiently large negative drift, a decreasing converging sequence is generated. This generates the opposite of a *c.e.* (a *co-c.e.*) real number.

Lemma 9 *The spatial position of any (rational) isolated accumulation is d-c.e. The coordinates of an isolated accumulation can be expressed as $(y - d \cdot t_0, t_0)$ or $(d \cdot t_0 - y', t_0)$ where y, y' and t_0 are c.e. and d is an integer.*

4 Global architecture and conventions

This section presents the general scheme to provoke an accumulation at a given location. It is based on a two-level structure: one handles the elementary movements and the accumulation, the other provides the shifting instructions to the first. The first one is called the *outer structure*, it moves and shrinks. The second one is called the *control structure*, it simulates a Turing machine and ensures that the output come in due time. The latter is composed of the *control* that does the computation and the *inner structure* that repeatedly accelerates the control to ensure that whatever the compu-

tation length, the output comes in bounded time. They are activated in sequence as shown in Fig. 5. This cycle never stops.

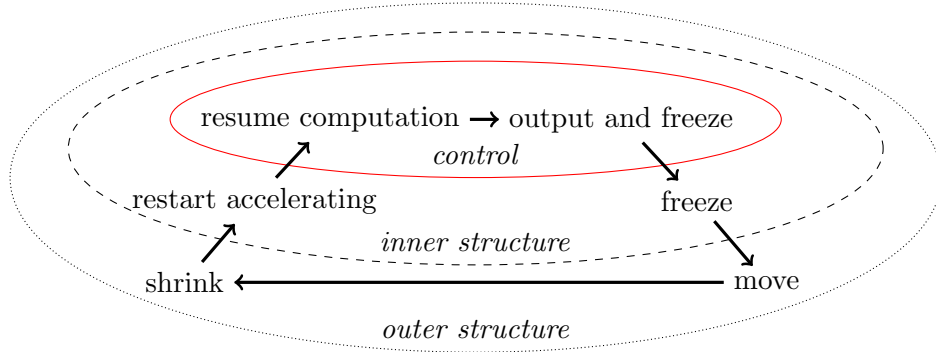


Figure 5: Two-level structure dynamics.

On each iteration, the outer structure ensures that the structure and everything that is embedded inside it is first moved so as to get to the right location at the limit and then scaled down by a constant factor (here one half). The movements depend on the kind of location aimed at. This is presented in more details in Sect. 7 and concrete constructions are provided in sections 8 and 9.

These shrinking step is illustrated with three steps in Fig. 6(c). This process is detailed in Durand-Lose (2011a). It is repeated forever generating an accumulation.

Shrinking a computation accelerates it. Thus the iterated shrinking scheme also provides an unlimited acceleration. This is used in the inner structure in order to guaranty an output within a given delay. But this iterated shrinking has to be stopped once the output is generated since otherwise it would provoke an unwanted accumulation.

The control simulates a Turing machine with output as described in Sect. 5. The control is stopped after each output so that there is no anticipated output to handle.

4.1 Graphical and naming conventions

In the illustrations, time elapses upward. Some space-time diagrams have different space and time scales for readability purposes. Even if the speed indicated do not match the illustrations, the sequence of collisions are correct as well as proportions following each axis. When the meta-signals are indicated on a space-time diagram, collision rules can be read directly.

The following drawing conventions are used. The meta-signals for the control are drawn solid while the ones for the inner structure are dashed and for the output structure dotted. The only exception is for the signals corresponding to input/output between structures that are drawn solid.

Similarly, the names of the meta-signals for the inner structure are prefixed by an i index (like in $\overrightarrow{i\text{back}}$). For the outer structure, the prefix is o (like in $\overrightarrow{o\text{back}}$). These prefixes are important since both structures include interactive shrinking, and thus there is a lot of corresponding meta-signals with similar semantics and names.

Over arrows are used to indicate the direction of meta-signals (*i.e.* left for negative speed and right for positive speed) like in $\overleftarrow{q_i}$ and $\overrightarrow{q_i}$ and no arrow for motionless (*i.e.* null speed) meta-signals.

The index s and f indicate *slow* and *fast* meta-signals that works in pair. The index l and h indicate *low* and *high* in a pair of signals that work in conjunction.

A collision rule is *blank* when outgoing meta-signals are exactly incoming meta-signals. Such rules are never listed. It is understood that undefined rules are blank.

5 Control

The control is a type-2 Turing machine with only one working tape, it provides an infinite sequence of finite output. After sending an output, the machine enters some waiting state. It remains in such a state until it receives some acknowledgment. It then resumes its computation and then sends the next output. Outputting is thus blocking.

A Turing machine can be simulated quite easily with signal machines as depicted in Fig.6(b) where the computation in Fig.6(a) is simulated. In the example, the value output is $\overline{11}$. Each vertical line/motionless signal encodes a cell of the tape. The head location and state are encoded by a moving signal. One collision carries out one iteration of the Turing machine.

Finitely many signals are used since the written part of the tape is finite. A $\overline{\#}$ signal is used to mark the last cell on the right. Each time the tape might need to be enlarged, the $\overline{\#}$ signal is turned to $\#$ and a new $\overline{\#}$ signal is added further right. This is done so that the distance to the new cell is only a fraction of the previous inter-cell distance. This ensures that, even though the number of cells is not bounded, the space used to encode the tape (and the whole Turing machine) is bounded. The continuity of space allows to stack an unbound number of signals in a bounded space.

Output is sent on the left as signals, one signal per symbol. Each time a

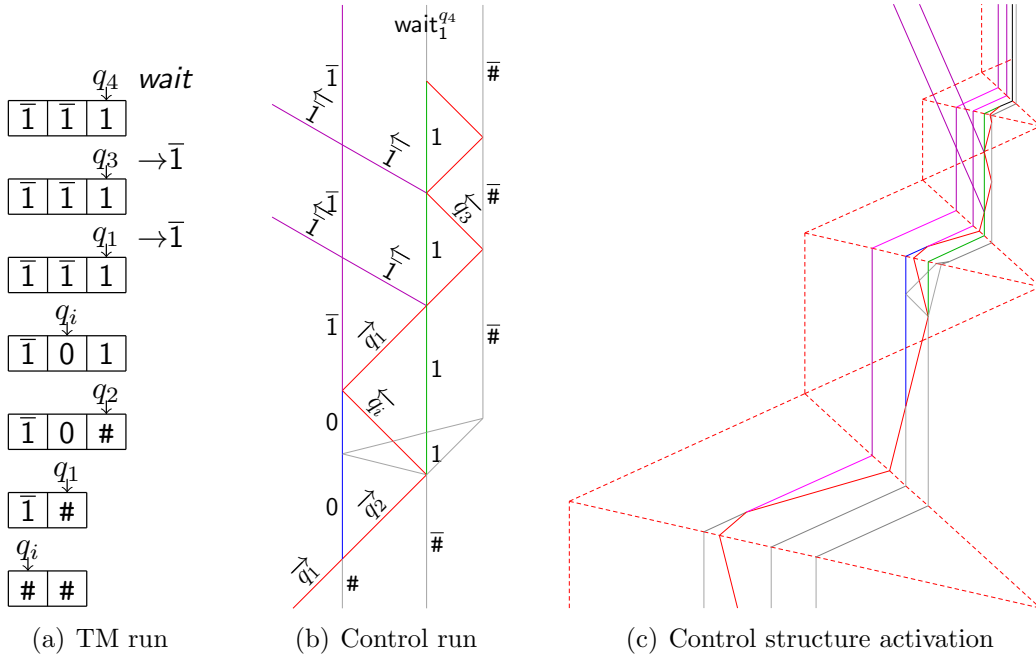


Figure 6: From Turing machine to control structure.

symbol is output, the signal encoding the head is sent on the right to bounce and come back. In the Turing machine, the head does not move when an output is done. In the simulation, the bouncing produces the iteration delay.

Waiting is ensured by having only null-speed signals, the one encoding the cell of the tape also encodes the state (and the position of the head) as $\text{wait}_1^{q_4}$ in the Fig. 6(b).

Computation is resumed on receiving a $\leftarrow \text{scale}_h$ signal (introduced later). This signal is restored (it is needed by the outer structure) together with the result of the waking transition of the Turing machine. (This is not illustrated.)

5.1 Control output for d -c.e.

In this section, a representation of any d -c.e. number by an infinite computable sequence of integers is provided. This sequence is provided in unary by the control.

Let x be any d -c.e. number. By Th. 7, there is a Turing machine that generates a sequence of rational numbers x_n such that this sequence converges to x and $\sum_{n \in \mathbb{N}} |x_{n+1} - x_n|$ converges. If x is c.e., it is also requested that the sequence is increasing (and the last part of the condition is trivial).

Let α be any positive rational number. (It is used as a scaling factor in following sections.) Let us define the sequences:

$$y_0 = \left\lfloor \frac{1}{\alpha} x_0 \right\rfloor \quad y_{n+1} = \left\lfloor \frac{2^{n+1}}{\alpha} (x_{n+1} - x_n + e_n) \right\rfloor \quad (1)$$

$$e_0 = x_0 - \alpha \cdot y_0 \quad e_{n+1} = x_{n+1} - x_n + e_n - \frac{\alpha}{2^{n+1}} y_{n+1} \quad (2)$$

where $\lfloor u \rfloor$ is the greatest integer less or equal to u ($\lfloor u \rfloor \leq u < \lfloor u \rfloor + 1$). It follows that

$$|e_n| < \frac{\alpha}{2^n} . \quad (3)$$

So that e_n converges to 0 and the sequence defined below, z_n , converges to x because, using (2),

$$\begin{aligned} z_n = \sum_{i=0}^n \frac{\alpha}{2^i} y_i &= x_0 - e_0 + \sum_{i=1}^n (-e_i + x_i - x_{i-1} + e_{i-1}) \\ &= x_n - e_n . \end{aligned}$$

z_n is an approximation of x_n up to $\frac{\alpha}{2^n}$. This is not a signed binary extension approximation since y_n is any integer and it is not bounded. Moreover, e_n is the error to x_n , not to x .

Since x_n is a computable sequence of rational numbers, then y_n and e_n are computable sequences of respectively integers and rational numbers.

If x is a positive *c.e.* number, then the computable sequence is increasing, so that only non-negative values are output; y_n is a sequence of natural numbers.

In the following, the control output y_n in unary (with $\bar{1}$ for negative values and with 1 for positive values) then waits. This loop is repeated forever.

6 Inner structure

The control is embedded inside an accelerating/shrinking structure in order to ensure that all the output is generated in a bounded time. The control plus this inner structure form the control structure. Figure 6(c) shows the first activation of a control structure.

The output leaves on the left, unaffected by the inner shrinking. This is done by canceling the collision rules from the shrinking structure that would act upon the meta-signals used for the output. The output is collected and processed by the outer structure.

To ensure that the inner shrinking structure itself does not generate any accumulation, it is also blocked after the output and resume on relaying the

$\overleftarrow{o\text{scale}_h}$ acknowledgment. This is done by freezing the structure when some waiting signal ($\overrightarrow{\text{wait}_1^{q_4}}$ in the example) appears in the control.

At each iteration of the shrinking structure (dashed lines on Fig. 6(c)), one signal, $\overrightarrow{i\text{back}}$, has to cross the configuration from left to right to start the next shrinking iteration (where $\overleftarrow{i\text{scale}_h}$ and $\overleftarrow{i\text{scale}_l}$ signals are released). Since $\overrightarrow{i\text{back}}$ crosses the control, then if any waiting signal is here, it meets it. In such a collision, the waiting signal is restored (not to lose any information) but $\overrightarrow{i\text{back}}$ is not (as can be seen on top of Fig. 6(c)): it disappears and the shrinking is stopped.

A $\overleftarrow{o\text{scale}_h}$ restarts the control structure by awaking the control (at the waiting signal) and then generating a new $\overrightarrow{i\text{back}}$ signal on the left side (at $i\text{border}_L$). The inner structure is restarted after the control. This restarting is illustrated at the top of the space-time diagrams in figures 7 and 9.

It might happen that the $\overrightarrow{i\text{back}}$ signal cannot reach the right side because it meets the new waiting signal before. This is not a problem since the inner structure is only there to accelerate the control and not to ensure any shrinking. The outer and control structures are organized such that everything works perfectly even if the inner structure never performs any shrinking step.

7 Outer structure

The outer structure waits a fixed time before collecting and processing the next output: the time for $\overrightarrow{o\text{back}}$ to cross from left ($o\text{border}_L$) to right ($o\text{border}_R$). The control structure has a limited activation time for outputting. But since it is embedded inside a shrinking structure, the Turing machine has an unlimited number of iterations ahead of it. This unbounded acceleration ensures that the output is available in due time.

The whole control structure iteration (control, shrinking, and outputting) takes place inside a triangle at the bottom. This triangle is delimited by a rightward signals, $\overrightarrow{o\text{back}}$ (it works for the outer structure identically to $\overrightarrow{i\text{back}}$ for the inner one) and a motionless signal on the right, $o\text{border}_R$ (the last of the configuration). In the following activation of the control structure, it is bounded from below by a leftward signal.

When an order is issued, the control structure is blocked with all its signal parallel as explained. The output of the control structure is the input of outer structure. It encodes an integer in unary notation and shift by shift elementary movements are made by the outer structure consuming one by one all the signals of the input.

Preserving the input is simple: signals have to cross $\overrightarrow{o\text{back}}$. The first signal of the output is removed and $\overrightarrow{o\text{back}}$ is replaced by a signal recording the erasing. All the remaining signals are turned into motionless signals shifted like the control structure signals. After doing the first atomic shift, $\overrightarrow{o\text{back}}$ is reissued on the left. It collects the next input signal. When the input is totally processed, $\overrightarrow{o\text{back}}$ reaches the right end unaffected and a (outer) shrinking starts.

Apart from signals from outer structure, all signals (input and control structure) are parallel. Parallel signals are easy to move preserving their relative position as illustrated on Fig. 6(c) where null-speed signals amounting for the tape cells go from one triangle to the next. Each time their distances are scaled by one half.

The outer structure provides the isolated accumulation but it moves (or waits) at each iteration so that to make the accumulation happens at some position according to the control.

When the order is carried out as well as the (outer) shrinking step, the control structure is restarted as explained above by the passing of the higher (*i.e.* slowest) of the two signals emitted from the right, $\overleftarrow{o\text{scale}_h}$. This signal turns the slanted signals into motionless ones to end the shrinking process as well as awakes the waiting signal and regenerates a (inner) $\overleftarrow{i\text{back}}$ signal. It is fast enough so that the control structure cannot catch up with it.

The whole process is exemplified in the different figures.

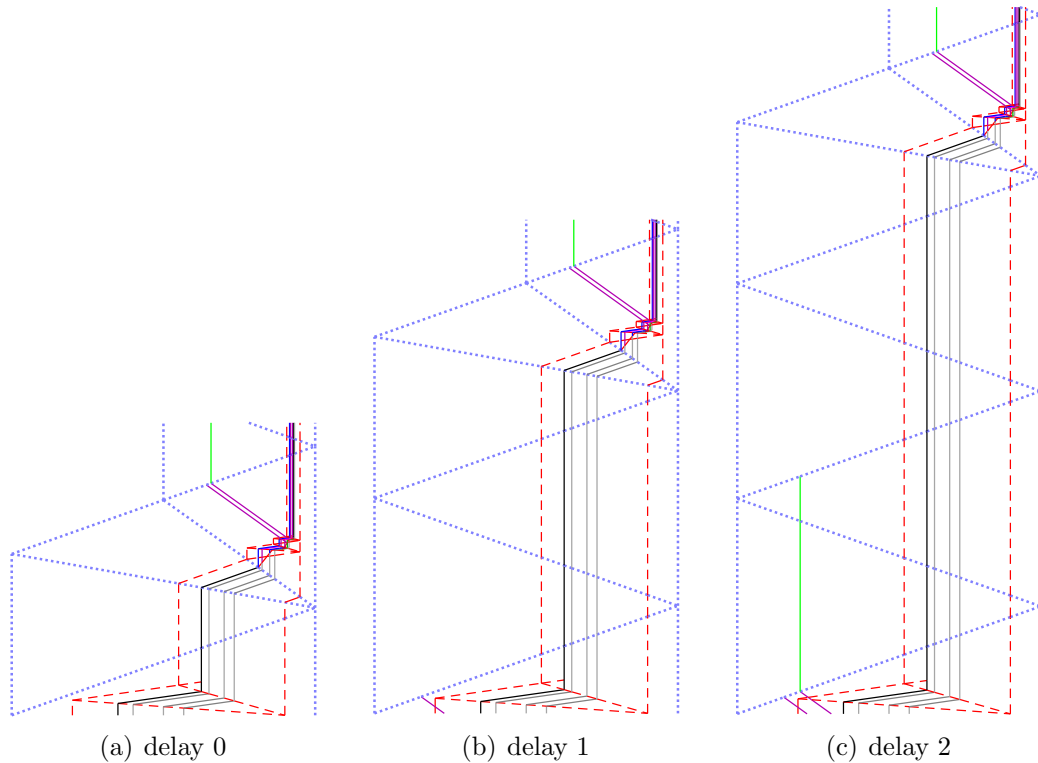
8 Accumulating at a *c.e.* time

On the n th iteration, the outer structure receives y_n as a unary encoded natural number output by the control. It waits y_n time the current delay and then shrinks the whole configuration by one half and wakes up the inner control.

If y_n is 0 (encoded as the empty word), the outer structure just shrinks as in Fig. 7(a): the bottom signal, $\overrightarrow{o\text{back}}$, crosses the configuration from left to right, does not meet any input and starts the shrinking process.

Figure 7(b) illustrates a unit delay: $\overrightarrow{o\text{back}}$ crosses the configuration left to right, collects the unique $\overleftarrow{1}$ input and changes to $\overrightarrow{o\text{loop}}$. It goes forth and, as $\overleftarrow{o\text{loop}}$, back to the left and starts again as $\overrightarrow{o\text{back}}$. But this time it encounters no input and starts the shrinking process.

If the y_n is greater, like in Fig. 7(c), the other $\overleftarrow{1}$'s are stored as motionless signals $\overrightarrow{o1}$ (vertical lines on the examples). Then each time $\overrightarrow{o\text{back}}$ passes it collects one and loops.



(The bottom of each space-time diagram is simplified: the run of the control has been removed and replaced by the output.)

Figure 7: Shrinking step and delays.

Figure 8 presents the outer structure alone with input $\overleftarrow{1} \overleftarrow{1}$. The signal machine (meta-signals and collision rules) is defined on the left. Meta-signals are indicated on the space-time diagram on the right. It can be seen how the input is handled and how $\overrightarrow{o} \text{loop}$ is generated.

The delay corresponds to the time for the signal to go forth and back the configuration. Since the configuration is shrunk by one half each time, the unit delay sequence is geometrical with one half factor. The sum of the delays is z_n for some scaling factor α depending on the initial configuration.

The outer shrinking process alone provides a term to the final accumulation time. This term, g , is the sum of a geometrical sequence of factor one half, which is rational (the initial value depends on the initial configuration which is rational). It is easy to scale down the initial configuration so that g is less than t_0 , the *c.e.* wanted for the time of the accumulation. Since *c.e.* numbers are stable by rational addition, $t_0 - g$ is *c.e.* and the control should output the sequence corresponding to it and α (as given by the scale of the initial configuration). Since the sequences are converging, there is indeed an

Meta-Signal	Speed
$\overleftarrow{o}scale_h$	-1/2
$\overleftarrow{o}scale_l$	-2
$\overrightarrow{o}border_L$	0
$\overrightarrow{o}border_R$	0
$\overleftarrow{o}1$	0
$\overleftarrow{1}$	-1/2
$\overleftarrow{o}loop$	-1
$\overrightarrow{o}loop$	1
$\overrightarrow{o}back$	1

Collision rules

$$\begin{aligned}
\{ \overrightarrow{o}back, \overleftarrow{o}scale_h \} &\rightarrow \{ \overrightarrow{o}border_L, \overrightarrow{o}back \} \\
\{ \overrightarrow{o}border_L, \overleftarrow{o}scale_l \} &\rightarrow \{ \overrightarrow{o}back \} \\
\{ \overrightarrow{o}back, \overrightarrow{o}border_R \} &\rightarrow \{ \overleftarrow{o}scale_l, \overleftarrow{o}scale_h, \overrightarrow{o}border_R \} \\
\{ \overrightarrow{o}back, \overleftarrow{o}1 \} &\rightarrow \{ \overrightarrow{o}loop \} \\
\{ \overrightarrow{o}border_L, \overleftarrow{o}loop \} &\rightarrow \{ \overrightarrow{o}border_L, \overrightarrow{o}back \} \\
\{ \overrightarrow{o}loop, \overrightarrow{o}border_R \} &\rightarrow \{ \overrightarrow{o}loop, \overrightarrow{o}border_R \} \\
\{ \overrightarrow{o}loop, \overleftarrow{1} \} &\rightarrow \{ \overrightarrow{o}1, \overrightarrow{o}loop \} \\
\{ \overrightarrow{o}back, \overleftarrow{1} \} &\rightarrow \{ \overrightarrow{o}loop \}
\end{aligned}$$

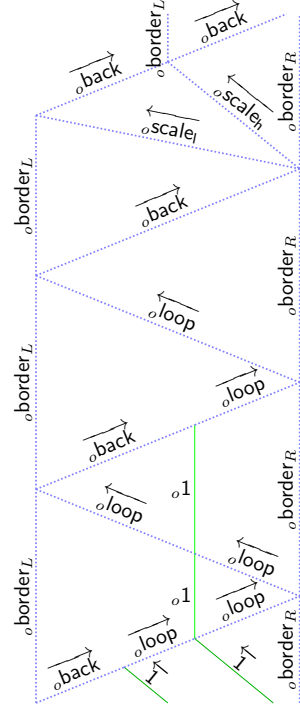


Figure 8: Meta-signals and rules for the outer structure for *c.e.* time.

accumulation.

Theorem 10 *For any c.e. number, there is a signal machine and an initial configuration that has only one accumulation and this accumulation happens exactly at that time.*

Figure 12(a) illustrates a run with a control that always outputs 11.

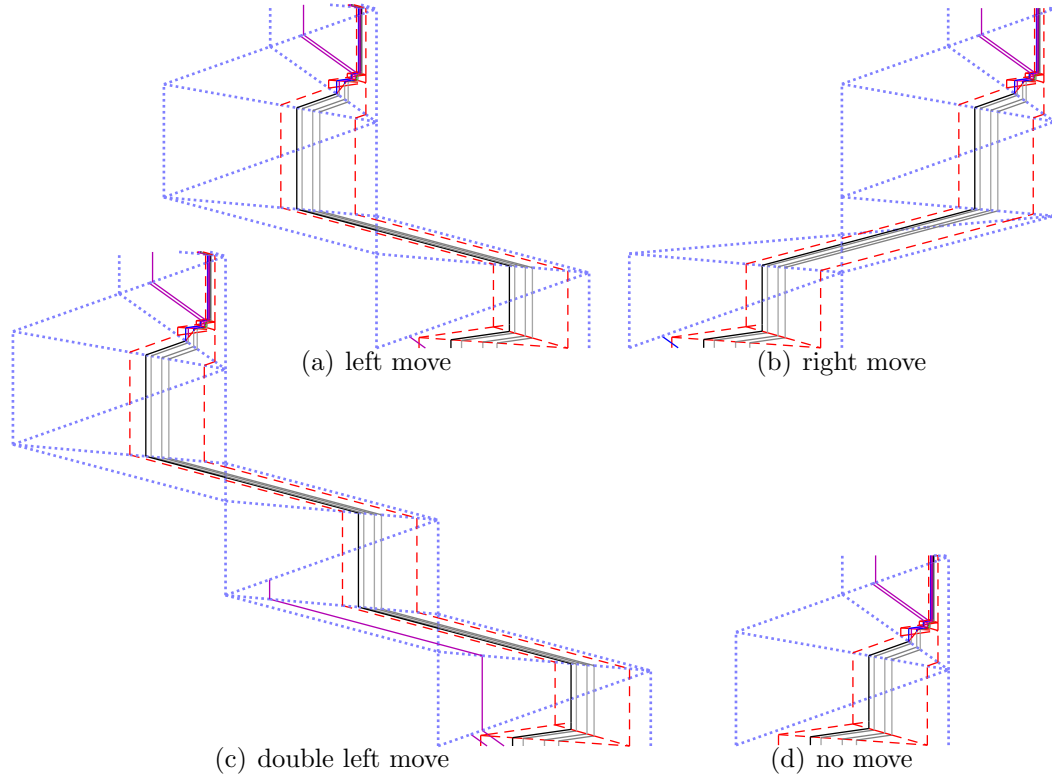
9 Accumulating at a *d-c.e.* spatial position

On the n th iteration, the outer structure receives y_n as a signed unary encoded integer from the control. The whole structure moves left (for negative values) or right (for positive values) the corresponding number of shifts (atomic movements). Each shift is equal to the width of the configuration.

Similarly to the previous case, a $\overrightarrow{o}back$ signal crosses the configuration. If it encounters any (signed) bit input, it records the first one and store the next ones as motionless signals. It processes the shift corresponding to the recorded bit and then starts checking again.

If $\overrightarrow{o}back$ encounters no input, then there is no movement and the shrinking

process is started. Figure 9(d) illustrates this case; it is similar to the one in Fig. 7(a).



(The bottom of each space-time diagram is simplified: the run of the control has been removed and replaced by the output.)

Figure 9: Shifts and shrinking.

If the input is a sequence of $\overleftarrow{1}$ then, before shrinking, for each one, the whole configuration is shifted by its width on the left. This is illustrated in Fig. 9(a) for $\overleftarrow{1}$ and in Fig. 9(c) for $\overleftarrow{1} \overleftarrow{1}$. Figure 10 presents the latter example with the names of the meta-signals and collision rules indicated but without the inner structure. Signal $\overrightarrow{o\text{back}}$ changes to $\overrightarrow{o\text{left}}$ as it collects the first $\overleftarrow{1}$ and then changes all the remaining $\overleftarrow{1}$ into $\overleftarrow{o1}$ so that they become parallel to the signals of the inner structure.

When the outer structure is activated, all signals from the inner structure are parallel so that it is very easy to change their direction and then to restore them. The direction of the signals for the inner structure is changed forth and back by $\overleftarrow{o\text{left}_f}$. When shifting they have the same direction as $\overleftarrow{o\text{left}_s}$ and $\overleftarrow{o1}$.

During the shift, two pairs of parallel signals, $\overleftarrow{o\text{left}_f}$, $\overleftarrow{o\text{left}_s}$, form a parallelogram. This ensures that the distance is the same. All distances between the signals of the inner structure are similarly preserved.

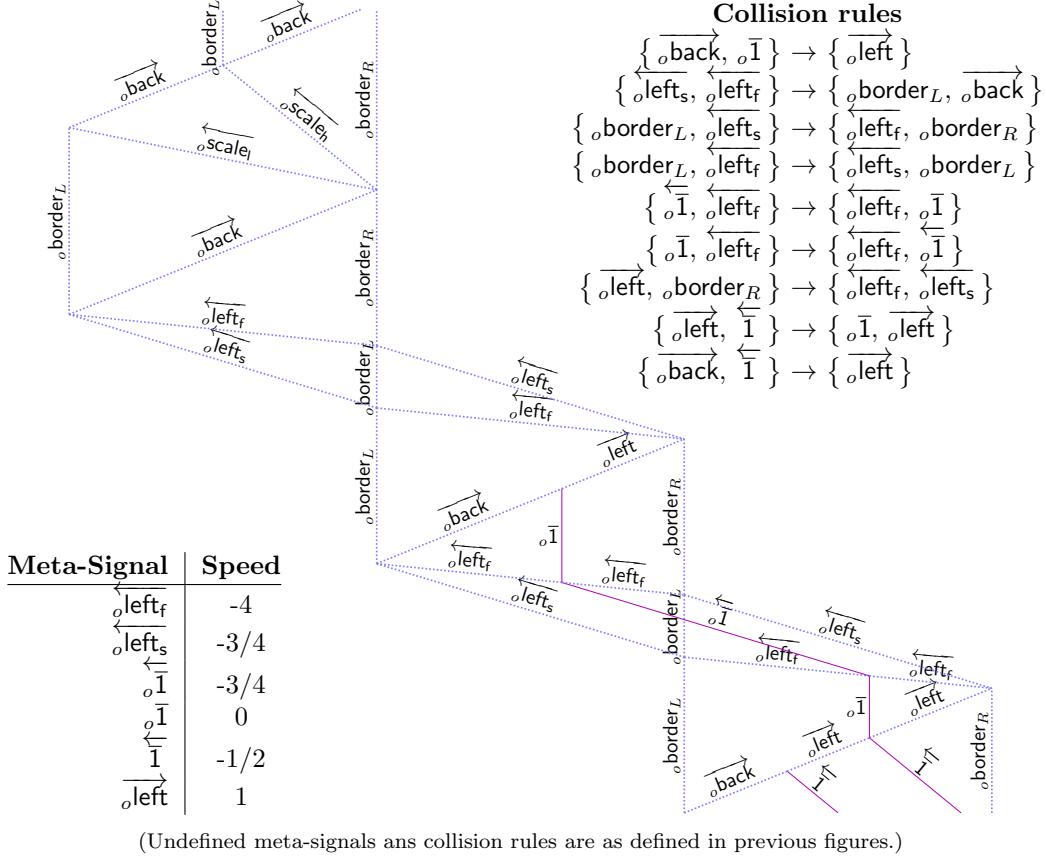


Figure 10: Meta-signals and rules for the outer structure for d -c.e. position, left move.

If the input is a sequence of $\overleftarrow{\text{I}}$'s then, for each one, the whole configuration is shifted by its width on the right before shrinking. This is illustrated in Fig. 9(b) for $\overleftarrow{\text{I}}$. Figure 11 presents the construction with on the input $\overleftarrow{\text{I}} \overleftarrow{\text{I}}$ the names of the meta-signals and collision rules indicated but without the inner structure. Signal $\overrightarrow{o\text{back}}$ changes to $\overrightarrow{o\text{right}}$ as it collects the first $\overleftarrow{\text{I}}$ and then changes all the remaining $\overleftarrow{\text{I}}$ into $\overleftarrow{o\text{I}}$ so that they become parallel to the signals of the inner structure.

The construction is somehow symmetric to the left construction. For the right shift in Fig. 9(b), the $\overleftarrow{o\text{right}_o\text{right}_s}$ parallelogram is not drawn completely: the upper $\overrightarrow{o\text{right}_s}$ is missing, but the diagonal signal, $\overrightarrow{o\text{right}_f}$, ensures the correct positioning and the preservation of distances. The speed of $\overrightarrow{o\text{right}_f}$

has been computed so that it is indeed a parallelogram. As before if the movement is of more than one shift, all the other signals of the input are stored and treated one after the other as depicted in Fig. 11.

Care has been taken so that the atomic shift is exactly by one width on left and right and the corresponding time delay is the same. (Like many claims in this paper, it can be proven by simple geometrical computations.) A large run is displayed in Fig. 12(b).

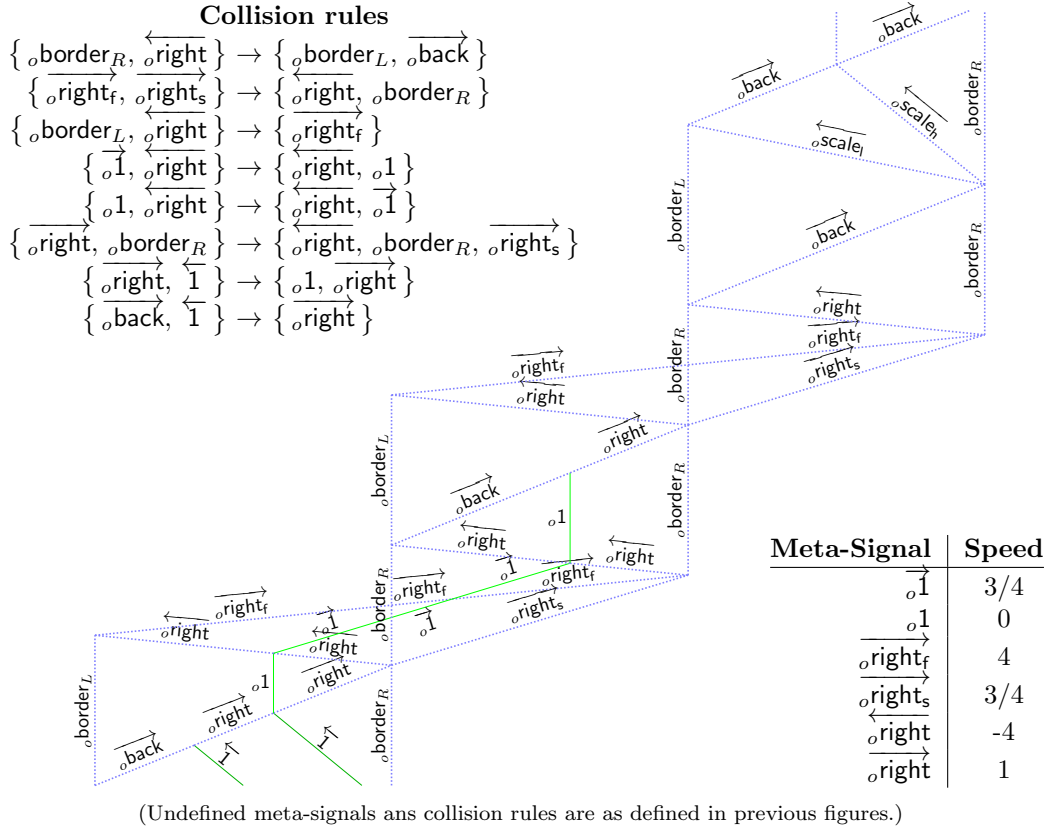


Figure 11: Meta-signals and rules for the outer structure for *d-c.e.* position, right move.

To accumulate on a given *d-c.e.* number, the right extremity of the structure should be set at coordinate 0. The right extremity has to be considered as origin point because the shrinking is always done on the right. The value of α can be chosen to be 1 by scaling and then the control Turing machine can be designed.

The whole structure accumulates because the total time is: the outer structure shrinking time—which is finite as a geometrical sum—plus the time for the shifts. The unitary shifts, whether on left or right, have the

same duration. This duration is proportional to the shift so we have to ensure that $\sum_n |y_n| 2^{-n}$ converges.

$$\begin{aligned}
|y_{n+1}| &\leq \left| \frac{2^{n+1}}{\alpha} (x_{n+1} - x_n + e_n) \right| + 1 && \text{(from (1))} \\
\frac{\alpha}{2^{n+1}} |y_{n+1}| &\leq |x_{n+1} - x_n| + |e_n| + \frac{\alpha}{2^{n+1}} \\
\frac{\alpha}{2^{n+1}} |y_{n+1}| &\leq |x_{n+1} - x_n| + 3 \frac{\alpha}{2^{n+1}} && \text{(from (3))} \\
\sum_{0 \leq n \leq N} \frac{\alpha}{2^{n+1}} |y_{n+1}| &\leq \sum_{0 \leq n \leq N} |x_{n+1} - x_n| + 3 \sum_{0 \leq n \leq N} \frac{\alpha}{2^{n+1}}
\end{aligned}$$

From Th. 7 the first sum converges (and the last one obviously does) thus there is an accumulation.

Theorem 11 *For any d-c.e., there is a signal machine and an initial configuration that has only one accumulation and this accumulation happens exactly at that spatial position.*

10 Conclusion

By considering a universal Turing machine in the control, Th. 10 can be extended as: there is a signal machine that can accumulate at any *c.e.* time depending only on the initial configuration. Similarly, Th. 11 can be extended as: there is a signal machine that can accumulate at any *d-c.e.* spatial location depending only on the initial configuration. Renaming the meta-signals, it is possible to merge these two machines.

Theorem 12 *There is a rational signal machine that can generate isolated accumulation at any c.e. time or d-c.e. spatial position depending on the initial configuration.*

Following the restriction on the coordinated expressed in Lem. 9, we conjecture that there can be an isolated accumulation at any such coordinates, i.e., time and spatial position simultaneously as long as the condition is satisfied.

When spatial dimension 2 and above is addressed, it seems that each spatial coordinate could be treated independently and the same results would hold with a similar relation with time.

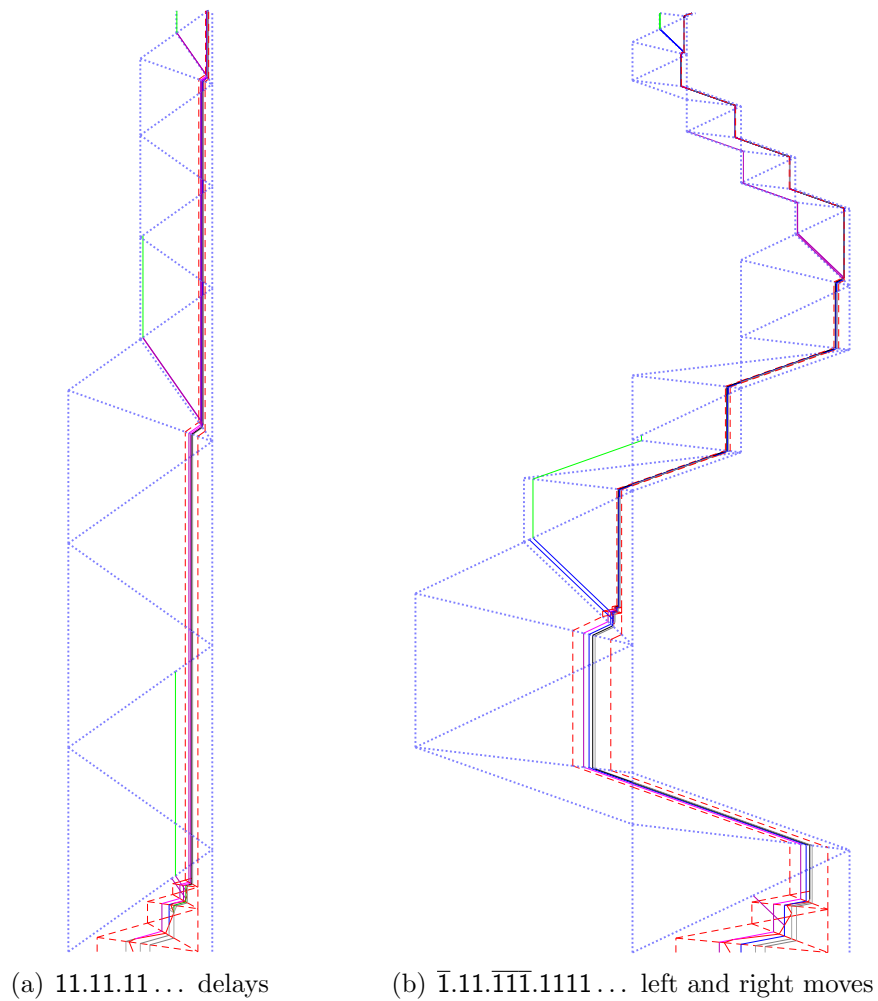


Figure 12: Longer runs.

Acknowledgement

The author would like to thank Laurent Bienvenu for inspiring discussions on *d-c.e.* numbers.

References

- A. Adamatzky, editor. *Collision based computing*. Springer London, 2002.
- A. Adamatzky and J. Durand-Lose. Collision computing. In D. Corne, editor, *Handbook of Natural Computing: Theory, Experiments, and Applications, Part II*. Springer, 2012.

- K. Ambos-Spies, K. Weihrauch, and X. Zheng. Weakly computable real numbers. *J. Complexity*, 16(4):676–690, 2000. doi: 10.1006/jcom.2000.0561.
- H. Andréka, I. Németi, and P. Németi. General relativistic hypercomputing and foundation of mathematics. *Nat. Comput.*, 8(3):499–516, 2009. doi: 10.1007/s11047-009-9114-3.
- L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bull. Amer. Math. Soc.*, 21(1):1–46, 1989.
- D. Duchier, J. Durand-Lose, and M. Senot. Fractal parallelism: Solving SAT in bounded space and time. In C. Otfried, K.-Y. Chwa, and K. Park, editors, *Int. Symp. on Algorithms and Computation (ISAAC '10)*, number 6506 in LNCS, pages 279–290. Springer, 2010. doi: 10.1007/978-3-642-17517-6_26.
- D. Duchier, J. Durand-Lose, and M. Senot. Computing in the fractal cloud: modular generic solvers for sat and q-sat variants. In M. Agrawal, S. B. Cooper, and A. Li, editors, *Theory and Applications of Models of Computations (TAMC '12)*, number 7287 in LNCS, pages 435–447. Springer, 2012. doi: 10.1007/978-3-642-29952-0_42. URL <http://arxiv.org/abs/1105.3454>.
- J. Durand-Lose. Abstract geometrical computation: Turing computing ability and undecidability. In B. S. Cooper, B. Löwe, and L. Torenvliet, editors, *New Computational Paradigms, 1st Conf. Computability in Europe (CiE '05)*, number 3526 in LNCS, pages 106–116. Springer, 2005. doi: 10.1007/11494645_14.
- J. Durand-Lose. Abstract geometrical computation 1: Embedding black hole computations with rational numbers. *Fund. Inf.*, 74(4):491–510, 2006a.
- J. Durand-Lose. Forecasting black holes in abstract geometrical computation is highly unpredictable. In J.-Y. Cai, S. B. Cooper, and A. Li, editors, *Theory and Applications of Models of Computations (TAMC '06)*, number 3959 in LNCS, pages 644–653. Springer, 2006b. doi: 10.1007/11750321_61.
- J. Durand-Lose. Abstract geometrical computation and the linear Blum, Shub and Smale model. In S. B. Cooper, B. Löwe, and A. Sorbi, editors, *Computation and Logic in the Real World, 3rd Conf. Computability in Europe (CiE '07)*, number 4497 in LNCS, pages 238–247. Springer, 2007. doi: 10.1007/978-3-540-73001-9_25.
- J. Durand-Lose. Abstract geometrical computation with accumulations: Beyond the Blum, Shub and Smale model. In A. Beckmann, C. Dimitracopoulos, and B. Löwe, editors, *Logic and Theory of Algorithms, 4th Conf. Computability in Europe (CiE '08) (abstracts and extended abstracts of unpublished papers)*, pages 107–116. University of Athens, 2008a.

- J. Durand-Lose. The signal point of view: from cellular automata to signal machines. In B. Durand, editor, *Journées Automates cellulaires (JAC '08)*, pages 238–249, 2008b.
- J. Durand-Lose. Abstract geometrical computation 3: Black holes for classical and analog computing. *Nat. Comput.*, 8(3):455–472, 2009a. doi: 10.1007/s11047-009-9117-0.
- J. Durand-Lose. Abstract geometrical computation and computable analysis. In J. F. Costa and N. Dershowitz, editors, *Int. Conf. on Unconventional Computation 2009 (UC '09)*, number 5715 in LNCS, pages 158–167. Springer, 2009b. doi: 10.1007/978-3-642-03745-0\20.
- J. Durand-Lose. The coordinates of isolated accumulations [includes] computable real numbers. In F. Ferreira, H. Guerra, E. Mayordomo, and J. Rasga, editors, *Programs, Proofs, Processes, 6th Int. Conf. Computability in Europe (CiE '10) (abstracts and extended abstracts of unpublished papers)*, pages 158–167. CMATI, U. Azores, 2010.
- J. Durand-Lose. Abstract geometrical computation 5: embedding computable analysis. *Nat. Comput.*, 10(4):1261–1273, 2011a. doi: 10.1007/s11047-010-9229-6. Special issue on Unconv. Comp. '09.
- J. Durand-Lose. Geometrical accumulations and computably enumerable real numbers (extended abstract). In C. S. Calude, J. Kari, I. Petre, and G. Rozenberg, editors, *Int. Conf. Unconventional Computation 2011 (UC '11)*, number 6714 in LNCS, pages 101–112. Springer, 2011b. doi: 10.1007/978-3-642-21341-0.
- G. Etesi and I. Németi. Non-Turing computations via Malament-Hogarth spacetimes. *Int. J. Theor. Phys.*, 41(2):341–370, 2002. URL <http://arxiv.org/abs/gr-qc/0104023>.
- M. Hagiya. Discrete state transition systems on continuous space-time: A theoretical model for amorphous computing. In C. Calude, M. J. Dinneen, G. Paun, M. J. Pérez-Jiménez, and G. Rozenberg, editors, *Unconventional Computation, 4th Int. Conf., UC '05, Sevilla, Spain, October 3-7, 2005, Proceedings*, volume 3699 of LNCS, pages 117–129. Springer, 2005. doi: 10.1007/11560319\12.
- M. L. Hogarth. Deciding arithmetic using SAD computers. *Brit. J. Philos. Sci.*, 55:681–691, 2004.
- U. Hickenbeck. Euclidian geometry in terms of automata theory. *Theoret. Comp. Sci.*, 68(1):71–87, 1989. doi: 10.1016/0304-3975(89)90120-5.
- U. Hickenbeck. A result about the power of geometric oracle machines. *Theoret. Comp. Sci.*, 88(2):231–251, 1991. doi: 10.1016/0304-3975(91)90375-C.

- G. Jacopini and G. Sontacchi. Reversible parallel computation: an evolving space-model. *Theoret. Comp. Sci.*, 73(1):1–46, 1990. doi: 10.1016/0304-3975(90)90160-J.
- S. Lloyd and J. Ng, Y. Black hole computers. *Scientific American*, 291(5):31–39, November 2004.
- I. Takeuti. Transition systems over continuous time-space. *Electr. Notes Theor. Comput. Sci.*, 120:173–186, 2005. doi: 10.1016/j.entcs.2004.06.043.
- K. Weihrauch. *Introduction to computable analysis*. Texts in Theoretical computer science. Springer, Berlin, 2000.
- X. Zheng. A computability theory of real numbers. In A. Beckmann, U. Berger, B. Löwe, and J. V. Tucker, editors, *Logical Approaches to Computational Barriers, Second Int. Conf. on Computability in Europe, CiE '06, Swansea, UK, June 30-July 5, 2006, Proceedings*, volume 3988 of *LNCS*, pages 584–594. Springer, 2006. ISBN 3-540-35466-2. doi: 10.1007/11780342_60.