Abstract geometrical computation for Black hole computation (extended abstract)

Jérôme Durand-Lose

Laboratoire de l'Informatique du Parallélisme

École Normale Supérieure de Lyon
Unité Mixte de Recherche CNRS-INRIA-ENS LYON-UCBL n° 5668

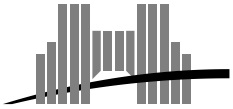
Abstract geometrical computation for Black hole computation (extended abstract)

Jérôme Durand-Lose

Avril 2004

Research Report N° 2004-15

École Normale Supérieure de Lyon
46 Allée d'Italie, 69364 Lyon Cedex 07, France
Téléphone : +33(0)4.72.72.80.37
Télécopieur : +33(0)4.72.72.80.80
Adresse électronique : `lip@ens-lyon.fr`



Abstract geometrical computation for Black hole computation (extended abstract)

Jérôme Durand-Lose

Avril 2004

Abstract

The Black hole model of computation provides a computing power that goes beyond the classical Turing computability since it offers the possibility to decide in finite time any recursively enumerable ($\mathcal{R}.\mathcal{E}.$) problem. In this article, we provide a geometric model of computation, conservative abstract geometrical computation, that has the same property: it can simulate any Turing machine and can decide any $\mathcal{R}.\mathcal{E}.$ problem through the creation of an accumulation. Finitely many signals can leave any accumulation, and it can be known whether anything leaves. This corresponds to a black hole artifact.

Keywords: Abstract geometrical computation, Black hole model, Energy conservation, Malament-Hogarth space-times, Turing universality, Zeno phenomena.

Résumé

Le modèle du calcul avec un trou noir fournit une puissance de calcul supérieure au calcul Turing classique puisqu'on peut y décider tout problème récursivement énumérable (R.E.). Dans cet article, nous proposons un modèle de calcul géométrique, conservative abstract geometrical computation, qui a la même propriété : il peut simuler n'importe quelle machine de Turing et, en créant une accumulation, décider n'importe quel problème R.E. Seulement un nombre fini de signaux peuvent quitter l'accumulation et il est possible de savoir si quoi que ce soit l'a quitté. Ceci correspond à l'artefact du trou noir.

Mots-clés: Abstract geometrical computation, Modèle du trou noir, Conservation de l'énergy, Espace-temps de Malament-Hogarth, Turing universalité, Phénomène de type Zéno.

Abstract geometrical computation for Black hole computation (extended abstract)

Jérôme Durand-Lose

April 2003

Abstract

The Black hole model of computation provides a computing power that goes beyond the classical Turing computability since it offers the possibility to decide in finite time any recursively enumerable ($\mathcal{R}.\mathcal{E}.$) problem. In this article, we provide a geometric model of computation, conservative abstract geometrical computation, that has the same property: it can simulate any Turing machine and can decide any $\mathcal{R}.\mathcal{E}.$ problem through the creation of an accumulation. Finitely many signals can leave any accumulation, and it can be known whether anything leaves. This corresponds to a black hole artifact.

Key-words: Abstract geometrical computation, Black hole model, Energy conservation, Malament-Hogarth space-times, Turing universality, Zeno phenomena.

None of the physicist aspects of this article is to be considered as true. The author, being a computer scientist with little knowledge on the matter, would not feel insulted if one would consider these mere inventions/illusions. However, we do not pretend to explain or describe black holes, but just to provide a computer scientist insight and model mostly directed to the computer science community. This paper could have been presented as a model of computation with special features, but since so much similarities exist, we stress on the correspondence.

1 Introduction

Theoretical physicists address the limits of the Church-Turing thesis as they get insights of possible space-times abiding Einstein's equations but providing more than classical Turing computing power [Hog94]. The idea is to have the possibility to use an infinite amount of time on a separate future endless curve to try solving a recursively enumerable ($\mathcal{R}.\mathcal{E}.$) problem, such that the result, or the absence of any result, can be retrieve in finite time in the main curve. For the theoretical computer scientist, this is related to infinite Turing computation or computation on ordinals [Ham02].

Malament-Hogarth space-times [EN02] provides this. Roughly speaking, the idea is to sent a computer in a black hole and wait, for a finite and known amount of time, for a yes or no answer. The machine sent in the black hole has an infinite amount of time ahead of it; but any signal it returns is received at the border within a bounded local delay. After this finite delay, the observer knows whether the computation ever stops (by noticing whether anything was received) and what the answer is. It is thus possible to decide any $\mathcal{R}.\mathcal{E}.$ problem in finite time.

Abstract geometrical computation [DL04] considers Euclidean lines. The support of space and time is thus $\mathbb{R}$. Computations are produced by *signal machines* which are defined by a

finite set of *meta-signals* and a finite set of *collision rules*. Signals are atomic information, corresponding to meta-signals, moving at constant speed thus generating Euclidean line segments on space-time diagrams. Collision rules are pairs *(incoming meta-signals, outgoing meta-signals)*, that define a mapping (which means determinism) over sets of meta-signals. They define what happens when signals meet, *i.e.* the extremities of the line segments.

A configuration (at a given time or the restriction of the space-time diagram to a given time) is a mapping from $\mathbb{R}$ to meta-signals, collision rules, and two special values: void (*i.e.* nothing there) and black hole (to mark accumulations). There should be finitely many positions not mapped to void. The time scale is $\mathbb{R}^+$, so that there is no such thing as a "next configuration". The following configurations are defined by the uniform movement of each signal, the speed of which is defined by its associated meta-signal. When two or more signals meet, this produces a *collision* defined by a collision rule. In the configurations following a collision, incoming signals are removed and outgoing signals corresponding to the outgoing meta-signals are added.

Zeno like acceleration and accumulation can be constructed as in Fig. 1. This provides the black hole-like artifact for deciding $\mathcal{R}.\mathcal{E}.$ problems. But accumulations can lead to an uncontrolled burst of signals producing infinitely many signals in finite time (as in the right of Fig. 1). In order to avoid this, we impose a *conservativeness* condition on the rules: a positive energy is defined for every meta-signal, the sum of these energies must be conserved by each rule. This means that there is no energy creation possible.

Each signal corresponds to a meta-signal which indicates its slope on the space-time diagram. Since there are finitely many meta-signals, there are finitely many slopes. This limitation may seem restrictive and unrealistic, even awkward as a quantification inside an analog model of computation. Let us notice that, first, it comes from cellular automata (CA) (as explained below): once a discrete line is identified, wherever (and whenever) the same pattern appears, the same line is expected, thus with the same slope. Second, we give two pragmatic arguments: (1) laws to compute new slopes in collisions are not so easy to design and pretty cumbersome to manipulate; (2) there is already much computing power and scheming things.

Abstract geometrical computation comes from the common use in literature of Euclidean lines to model discrete lines in the space-time diagram of CA to access dynamics or to design. Cellular automata form a well known and studied model of computation and simulation. Configurations are $\mathbb{Z}$-arrays of cells, the states of which belong to a finite set. Each cell can only access the states of its neighboring cells. All cells are updated iteratively and simultaneously. The main characteristics of CA, as well as abstract geometrical computation, are: parallelism, synchrony, uniformity and locality of updating. The space-time diagrams are colorings of $\mathbb{Z} \times \mathbb{N}$ with states. Discrete lines are often observed on these diagrams. They can be the keys to understanding the dynamics and correspond to so-called *particles* or *signals* as in, *e.g.*, [Ila01, pp. 87–94] or [BNR91, HSC01]. They can also be the tool to design CA for precise purposes and then named *signals* and used for, *e.g.*, prime number generation [Fis65], firing squad synchronization [VMP70, Maz96] or reversible simulation [DL97]. These discrete line systems have also been studied on their own [DM02, MT99]. All these papers, and many more, implicitly use abstract geometrical computation.

Before presenting our results, we want to convince the reader that it is not just "one more model of computation". First, it does not come "out of the blue" because of its CA origin. Second, to our knowledge[1], it is the only model that is a dynamical system with continuous time and space but finitely many local values. The closest model we know of is the Mondrian automata of Jacopini and Sontacchi [JS90]. They work on space-time diagrams which are mappings from $\mathbb{R}^n$ to a finite set of colors. Their diagrams should be bounded finite polyhedra; we are only addressing lines –faces are not considered– and our diagrams may be unbounded and accumu-

[1] A brief tour of analog/super-Turing models of computation can be found in [DL03, Chap. 2].

lation points are used (they just forbid them). Another close model is the piecewise-constant derivative system [AM95, Bou99]: $\mathbb{R}^n$ is partitioned into finitely many polygonal regions; the trajectory is defined by a constant derivative on each region, thus an orbit is a sequence (possibly over an ordinal) of (Euclidean) line segments. This model is sequential –there is only one "signal"– and the faces that delimit the regions are artifacts that do not exist in our model. Nevertheless, it also uses accumulations to decide $\mathcal{R}.\mathcal{E}.$ problems.

In this paper, space and time are restricted to rationals. This is possible since all the operations used preserve rationality. All intervals should be understood over $\mathbb{Q}$, not $\mathbb{R}$. Extending the definitions to real values is automatic but only the rational case is addressed here.

After formally defining our model in Sect. 2, we rapidly show that any Turing-computation can be carried out through the simulation of 2-counter automata in Sect. 3. The values of the counters are encoded by positions and the instructions are going forth and back between them and fixed signals indicating the scale. The continuous nature of space is used here: all $1/2^n$ positions exist.

In Sect. 4, we show how to bound temporally a computation that is already spatially bounded. This method is constructive and relies on the continuous nature of space and time. The construction generates an accumulation point. We explain how to use these accumulations for deciding $\mathcal{R}.\mathcal{E}.$ problems in Sect. 5.

Conclusion, remarks and perspectives are gathered in Sect. 6.

2 Definitions

Our abstract geometrical computations are defined by the following machines:

Definition 1 A *signal machine* is defined by (M, S, R) where M (*meta-signals*) is a finite set, S (*speeds*) is a mapping from M to $\mathbb{Q}$, and R (*collision rules*) is a subset of $\mathcal{P}(M)\times\mathcal{P}(M)$ that corresponds to a partial mapping of the subsets of M of cardinality at least 2 to the subsets of M (both domain and range are restricted to elements of different speeds).

The elements of M are called *meta-signals*. Each instance of a meta-signal is a *signal* which corresponds to a line segment in the space-time diagram. The mapping S assigns rational *speeds* to meta-signals, *i.e.* the slopes of the segments. The set R defines the *collision rules*, noted $\rho^- \rightarrow \rho^+$: what happens when two or more signals meet. It also defines the intersections of the segments. The signal machines are deterministic because R must correspond to a mapping.

The *extended value set*, V, is the union M and R plus two symbols: one for void, $\oslash$, and one for a black hole ❋. A *configuration*, c, is a total mapping from $\mathbb{Q}$ to V such that the set $\{ x \in \mathbb{Q} \mid c(x) \neq \oslash \}$ is finite.

A signal corresponding to a meta-signal μ at a position x, *i.e.* $c(x) = \mu$, is moving uniformly with constant speed $S(\mu)$. A signal must start in the initial configuration or be generated by a collision. It must end in a collision or in the last configuration. This corresponds to condition 2. in Def. 2. At a $\rho^- \rightarrow \rho^+$ collision, all, and only, signals corresponding to the meta-signals in ρ^- (resp. ρ^+) must end (resp. start) in this collision. No other signal should be present. This corresponds to 3. in Def. 2. A black hole corresponds to an accumulation of collisions and disappears without a trace. This corresponds to 4. in Def. 2.

Let S_{min} and S_{max} be the minimal and maximal speeds (*i.e.* the extrema of S). The *causal past*, or *light-cone*, arriving at position x and time t, $J^-(x,t)$, is defined by all the positions that might influence through signals, formally:

$$J^-(x,t) = \{ \ (x',t') \mid (0 \leq S_{max}(t-t') - x + x') \wedge (0 \leq x - x' - S_{min}(t-t')) \ \} \ .$$

Before formally defining the dynamics by space-time diagrams, we want to point out the black hole formation example of Fig. 1. This example is so simple (*i.e.* 4 meta-signals and 2 collision rules) that such a situation cannot be excluded.

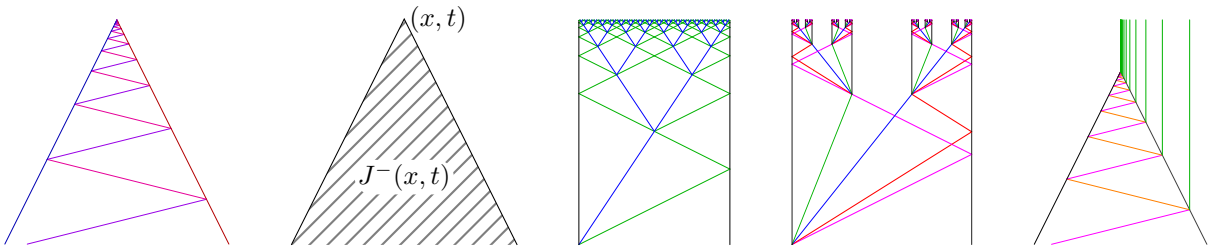


Figure 1: Black hole, light-cone and unwanted phenomena.

Definition 2 The *space-time diagram*, or orbit, issued from an initial configuration c_0 and lasting for T [2], is a mapping c from $[0,T]$ to configurations (*i.e.* a mapping from $\mathbb{Q}\times[0,T]$ to V) such that, $\forall(x,t)\in\mathbb{Q}\times[0,T]$:

1. $\forall t{\in}[0,T]$, $\{\, x\in\mathbb{Q} \mid c_t(x)\neq\oslash \,\}$ is finite,
2. if $c_t(x){=}\mu$ then $\exists t_i,t_f{\in}[0,T]$ with $t_i{<}t{<}t_f$ or $0{=}t_i{=}t{<}t_f$ or $t_i{<}t{=}t_f{=}T$ s.t.:
   - $\forall t'\in(t_i,t_f)$, $c_{t'}(x+S(\mu)(t'-t))=\mu$,
   - $t_i=0$ or $c_{t_i}(x_i)\in R$ and $\mu\in(c_{t_i}(x_i))^+$ where $x_i=x+S(\mu)(t_i-t)$,
   - $t_f=T$ or $c_{t_f}(x_f)\in R$ and $\mu\in(c_{t_f}(x_f))^-$ where $x_f=x+S(\mu)(t_f-t)$;
3. if $c_t(x)=\rho^-{\to}\rho^+\in R$ then $\exists\varepsilon$, $0<\varepsilon$, $\forall t'\in[t-\varepsilon,t+\varepsilon]$, $\forall x'\in[x-\varepsilon,x+\varepsilon]$,
   - $c'_t(x')\in\rho^-{\cup}\rho^+\cup\{\oslash\}$,
   - $\forall\mu\in\rho^-$, ($c'_t(x')=\mu$) $\Leftrightarrow$ ($t'<t$ and $x'=x+S(\mu)(t'-t)$)),
   - $\forall\mu\in\rho^+$, ($c'_t(x')=\mu$) $\Leftrightarrow$ ($t<t'$ and $x'=x+S(\mu)(t'-t)$)),
4. if $c_t(x)=$ ❋ then
   - $\exists\varepsilon>0$, $\forall(x',t')\notin J^-(x,t)$ s.t. $|x{-}x'|{<}\varepsilon$ and $|t{-}t'|{<}\varepsilon$, $c_{t'}(x)=\oslash$,
   - $\forall\varepsilon>0$, $\big|\ \{\ (x',t')\in J^-(x,t)\mid t{-}\varepsilon{<}t'{<}t\wedge c_{t'}(x')\in R\ \}\ \big|=\infty$.

On space-time diagrams, the traces of signals represent line segments whose directions are defined by $(S(.),1)$ (1 is the temporal coordinate). Collisions correspond to the extremities of these segments. Examples of space-time diagrams are provided by the various figures. Time is always increasing upwards.

The middle space-time diagram of Fig. 1 provides an example of a possible but unwanted one. It is not compatible with Def. 2 if times after the accumulation are to be considered. The number of signals is bursting to infinity and there is a black hole segment. This is unwanted because on the one hand it corresponds to the free apparition of energy, and on the other hand we fell that black holes should be dimensionless points. The two remaining space-time diagrams show even more unwanted cases. We thus introduce the following restriction that prevents such cases and corresponds to the energy conservation.

Definition 3 A signal machine is *conservative* when an atomic positive energy is defined for all meta-signals $(E:M\to\mathbb{N}^*)$[3] such that the total energy of the system is preserved, *i.e.* the

[2] This definition can easily be extended to the $T=\infty$ case.

[3] Integer are enough, since there are finitely many meta-signals.

sum of all the energy of existing signals is a constant of the system. This is equivalent to accept only rules that preserve this energy, *i.e.* the sum of the energy of incoming meta-signals equals the sum of outgoing ones.

Conservativeness is straightforward if the condition on rules is satisfied. If it is not satisfied, it is very easy to built a configuration such that only this rule is used and then the energy is not preserved.

Property 4 *Given a conservative signal machine and an initial configuration, the number of signal in any following configuration, as well as the number of accumulations, is bounded (by the total energy divided by the least atomic energy).*

Energy can only be lost in "black hole" formation, *i.e.* accumulation. A sub-case of conservativeness is when all the meta-signals have the same energy and the number of in and out meta-signals are always equal. This is the case in the rest of this article. We chose to present a more complex notion since it is weaker and better suits the physical notion of the energy conservation.

3 Turing-computation capability

We prove the Turing-computation power of our model by simulating any 2-counter automaton (a finite automaton couple with two counters, A and B). The possible actions on any counter are *add/subtract* 1 and *branch if non-zero*. These machines can be described with a six-operations (the three aforementioned ones for each of the two counters) assembly language with branching labels as on the left part of Fig. 6 (see [Min67] for more on 2-counter automata).

The simulation is carried out with both counters encoded by relative positions according to two fixed signals zero and one. These two signals form a scale on the diagram. The counter A (resp. B) is encoded by a single signal a (b) at position $\alpha 2^{-a}$ ($\beta 2^{-b}$) as in Fig. 2. The parameter α and β are rationals such that $1 < \alpha < \beta < 2$; this ensures that the signal a (b) is between zero and one unless its value is zero and in such a case it is on the other side of one. Let us note that the values of α and β prevent the signals from occupying the same place nor to be on the scale signals. As can be easily checked on the construction in the rest of this section, they also prevent that any collision happens on an unconcerned signal.

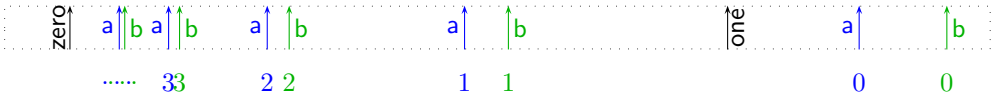


Figure 2: Encoding positions of counters.

The current instruction (*e.g.* n) is encoded as the signal $\overleftarrow{\mathsf{n}}$. It moves back to zero, bounces, carries out the operation and returns as the next operation. The five possible configurations are given in Fig. 3.

The fact that a signal encoding a counter is on the other side of one only for the value 0 provides an easy way to test for non-zero for branching or subtracting 1: going rightward one is encountered first if and only if the value of the counter is 0.

They are two kinds of meta-signals: 8 for the counters and borders, and the ones generated for the code. The meta-signals of the first kind are: zero, one, a and b of speed 0 used to mark the borders and to encode A and B, and $\overleftarrow{\mathsf{a}}$ ($\overleftarrow{\mathsf{b}}$) and $\overrightarrow{\mathsf{a}}$ ($\overrightarrow{\mathsf{b}}$) of speed -1 and 1 used to increment/decrement A (B). For the second kind, each line n of the program is converted into

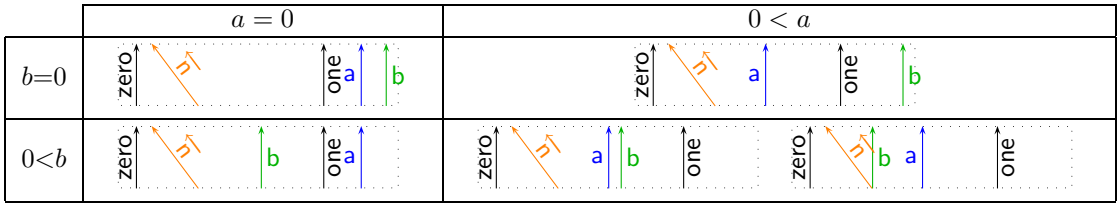


Figure 3: Encoding of configurations.

$\overrightarrow{\mathsf{n}}$ and $\overleftarrow{\mathsf{n}}$ of speed 2 and -2 and possibly $\overrightarrow{\mathsf{n'}}$ and $\overleftarrow{\mathsf{n'}}$ of speed 3 and -3 to carry out increment and decrement as explained below.

First any instruction bounces on zero to be on the left of any other signal and thus be in the right position to start carrying out any instruction. This is achieved by the following rule:

$$\{\mathsf{zero}, \overleftarrow{\mathsf{n}}\} \quad \to \quad \{\mathsf{zero}, \overrightarrow{\mathsf{n}}\}.$$

The full transformation of a program into a signal machine is not given. We only detail the collision rules generated for the most complicated case: a `A--` instruction (at line n). The rules are the following:

$$\begin{array}{llll} \{\overrightarrow{\mathsf{n}}, \mathsf{one}\} & \to & \{\overleftarrow{\mathsf{n{+}1}}, \mathsf{one}\}, & \{\overrightarrow{\mathsf{n}}, \mathsf{a}\} \quad \to \quad \{\overleftarrow{\mathsf{n'}}, \overrightarrow{\mathsf{a}}\}, \\ \{\mathsf{zero}, \overleftarrow{\mathsf{n'}}\} & \to & \{\mathsf{zero}, \overrightarrow{\mathsf{n'}}\}, & \{\overrightarrow{\mathsf{n'}}, \overrightarrow{\mathsf{a}}\} \quad \to \quad \{\overleftarrow{\mathsf{n{+}1}}, \mathsf{a}\}. \end{array}$$

All other collisions are blank, *i.e.*, the same signals are regenerated. The effect of these instructions is shown in the space-time diagrams of Fig. 5. The relative position of one and a is very important because a counter already at zero is not decreased. If such is not the case, the distance between zero and a is multiplied by 2 as it can easily be geometrically checked on Fig. 5 where the slopes are indicated with dotted lines.

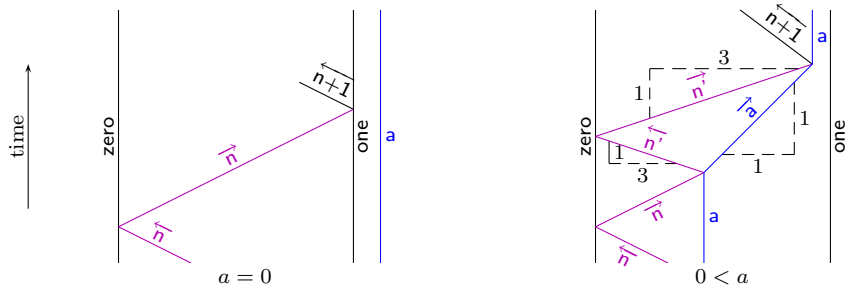


Figure 4: Implementation of `A--`.

The instructions `A++` does exactly the same thing in reverse, but the zero case does not have to be considered. We do not give the rules, they can be recovered from the left diagram space-time of Fig. 5. The non-zero tests are done by simply noticing that one is met before only if it is zero. This is illustrated by the last two space-time diagrams of Fig. 5.

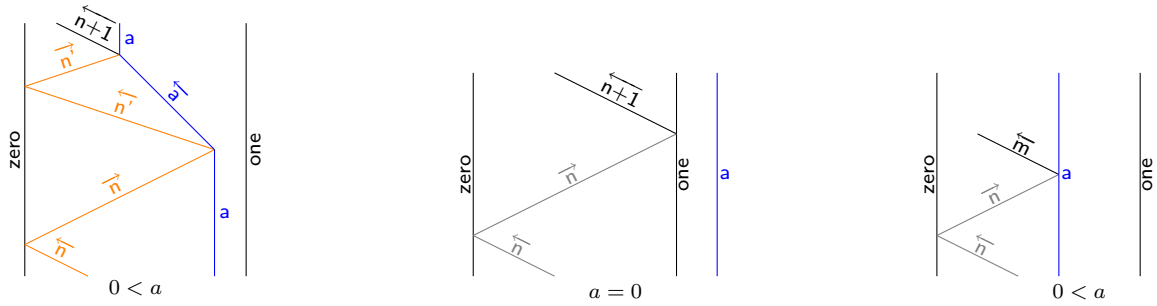


Figure 5: Implementations of `A++` and n : `IF A!=0` m.

All the instructions on B are carried out similarly.

Figure 6 provides three space-time diagrams associated to different initial values. The pictures are strained vertically in order to fit.

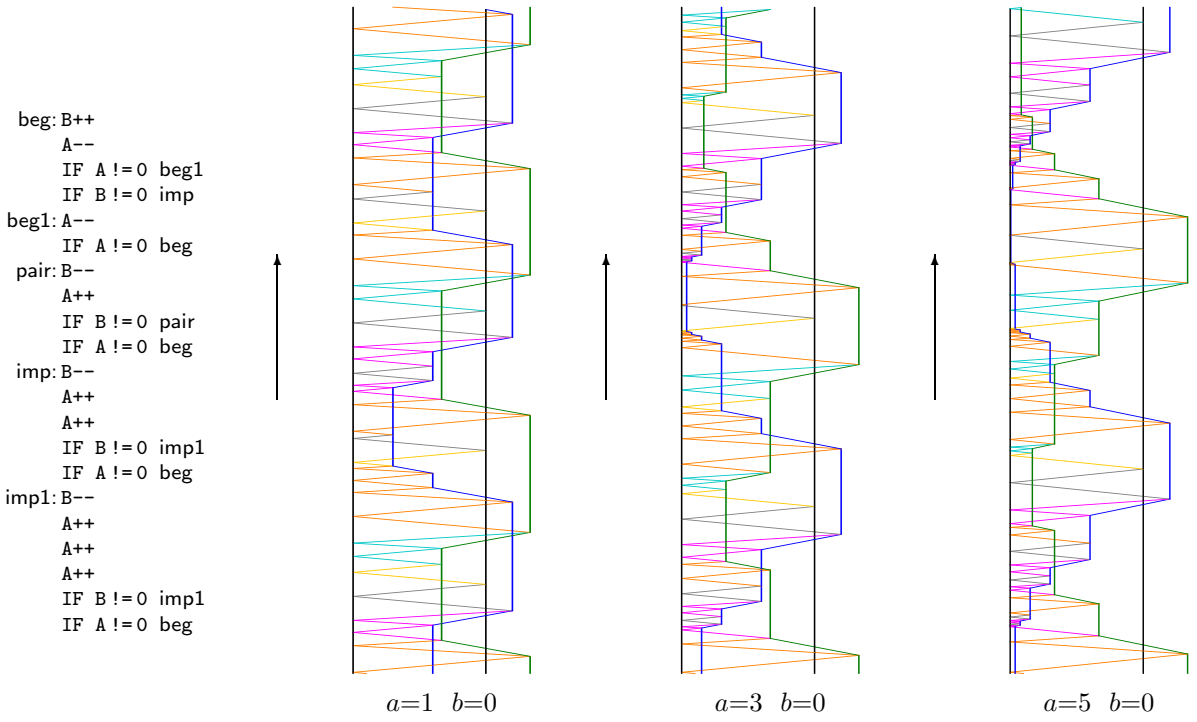


Figure 6: A 2-counter automaton and its simulations for three different initial values.

There is something left to consider: the end of the computation, *i.e.* the treatment of the halt. It is not possible to just make the instruction signal disappears since this would yields a non conservative rule. To cope with this, one can choose to let the instruction signal leaves on the left (but this signal could be damaging for the rest of the computation), or to let it bounce indefinitely between zero and one, thus also preserving the number of signals.

All together, any 2-counter automaton can be simulated by a conservative signal machine. In fact, any finite number of counters can be included and treated similarly. Signal machines thus form a model of computation which has at least Turing-computing capability.

4 Contraction principle

It is possible to partially strain any space-time diagram as schematized on Fig. 7. The idea is to decompose the upper part according to two non-collinear vectors. One vector is used as a frontier (here the one of speed β). A change of scale is done on the second one (here multiplication by 3 on the axis corresponding to speed α). This is a strain of a given ratio about the second axe. On Fig. 7, the dotted lines indicate how the images of two points are computed. The grey parts indicate the ongoing computation.

This geometrical transformation is easily implemented inside our model: by switching to strained signal on the frontier, all following computations mimic the unstrained one. The following meta-signals are added: one for the frontier, and one strained meta-signal for each initial meta-signal[4]. All the collision rules are duplicated so that strained signals behave exactly as unstrained ones. Collisions of the form $\{$*frontier and unstrained*$\}\rightarrow\{$*frontier and strained*$\}$

[4]Its speed is computed by some $(ax+b)/(cx+d)$ formula whose coefficients depend on the parameters which have to verify some easily satisfied conditions (see [DL03, Chap. 7] for details).

are added. New rules are created to account for the possibility of the frontier to pass exactly on a collision. Conservativeness is preserved by setting identical energies to corresponding strained meta-signals.

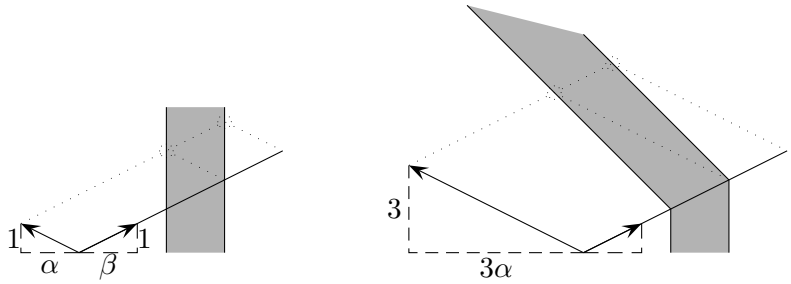


Figure 7: Strain principle.

With this construction, it is possible to build a structure that scales by one half the rest of the computation as illustrated on Fig. 8. The two directions used correspond to v_0 and $4v_0$, where v_0 is big enough. In the left picture, nothing happens. In the middle picture, the lower signal is the frontier and a strain of ratio 1/2 is done about to the upper signal. In the right picture, a second strain takes place: the role of the directions are exchanged, and the ratio is still 1/2. After the two strains, the computation is scaled by 1/2 on both directions, thus on any direction. The whole computation is scaled by 1/2 and the original meta-signals can be used again since the computation undergoes no strain after the second one. This makes it possible to iterate the shrinking.

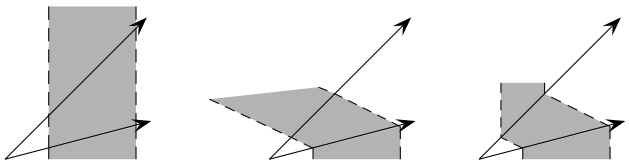


Figure 8: Shrinking principle.

From now on, only spatially bounded space-time diagrams are considered. This is sufficient to ensure that the computation remains inside the structure when shrinking is iterated. It is possible to add some extra signals to restart the shrinking each time as in the left part of Fig. 9. The right picture represents the application of this structure to a simulation of a 2-counter automaton.

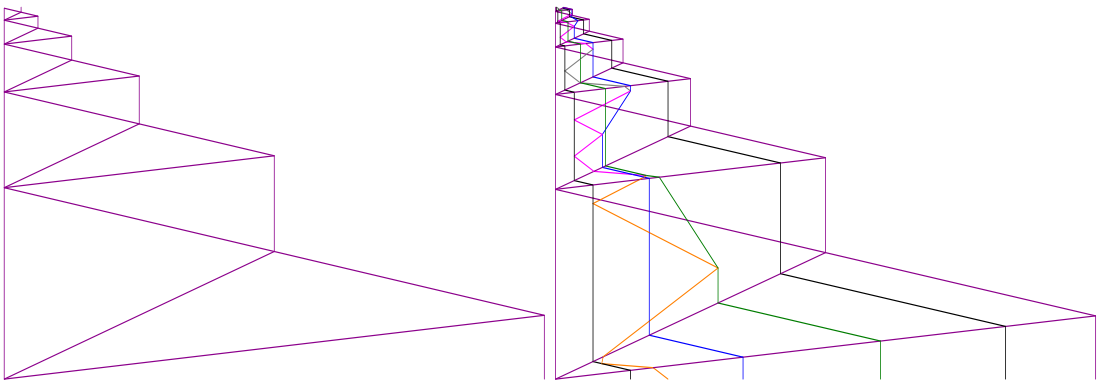


Figure 9: Iterated shrinking: structure and examples.

In each space-time diagrams of Fig. 9, there is an accumulation point: there are infinitely many collisions accumulating to the upper angle of the triangle. This is a "Zeno effect": finite

(continuous) duration but infinitely many (discrete) instants. All collisions are in the light cone ending there (and there is nothing out of it). This corresponds to the black hole of Cond. 4 in Def. 2.

5 Black hole formation

We consider the simulation of a 2-counter automata such that, when the simulation stops, if the configuration corresponds to acceptance, then the instruction signal goes on the left. In the case of rejection, another signal would be issued.

The construction of the iterated shrinking can then be modified in order not to act on this signal (*i.e.* it is always generated unstrained and never strained) so that it leaves the iterated shrinking. The iterated shrinking plays the role of the black hole and these specially treated signals represent the information that "leaves" the black hole before the collapsing.

The only thing left is to get this information or assert that no information left (*i.e.* the computation never stops). This is done by bounding the iterated shrinking by 2 signals that meet after the black hole. If a notification of acceptance or rejection leaves, they grab it before they meet. So that, at the meeting, they know whether the computation finishes. This is illustrated by Fig. 10.

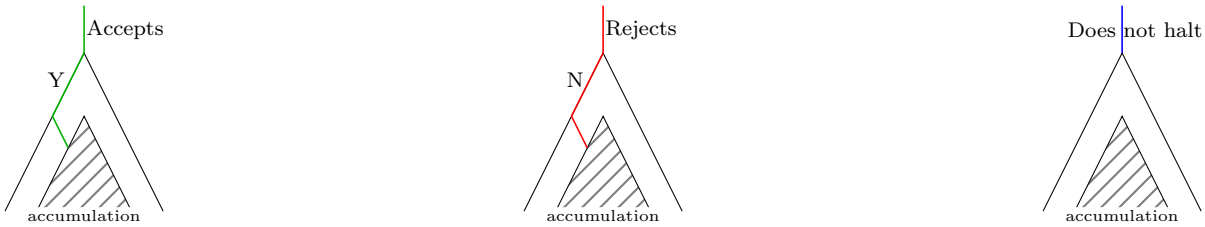


Figure 10: Encapsulation of a black hole.

6 Conclusion

We have provided a geometrical model of computation that is Turing-computation universal and has the special features of the Black hole model. We are not using already existing black holes, but rather creating them on demand (a Malament-Hogarth space-time is implicitly built). It is not so strange that computation forms the black hole since they come from the same matter as the machine sent into. One can also consider that some signals fix black hole formation, while others carry out the computation using the black hole.

One may object that black holes disappearance is not acceptable. The underlying space is one-dimensional, any remaining black hole would form a barrier preventing information to cross from one side to the other; in two and more dimension, it is alway possible to go around it. Another argument is to imagine that signals are drifting in a higher dimensional space, so that the black holes remain, but its orbit is not in the plane of the space-time diagram.

Reversibility is an important issue in theoretical physics. One can easily check that reversibility corresponds to R being one-to-one. At the expense of more complex constructions, universality can be achieved as well as the use of accumulations as black holes. But the final collapsing is not reversible.

The number of possible black holes / $\mathcal{R}.\mathcal{E}.$ queries is bounded from the start (each needs a minimal amount of energy). Unless the black hole returns the energy in some form, which is clearly forbidden here, there is no way to address second order accumulations (*i.e.* ω^2 or second

order space-time arithmetic deciding or Σ_2^0 in the arithmetical hierarchy), unless infinitely many signals exist at start, apart one from another. This way it would be reasonably possible to hope to climb the arithmetical hierarchy as in [AM95, Bou99].

As long as the model is restricted to rationals, there are finitely many signals present at any instant and there is no accumulation, the model is Turing-universal and can be simulated by any Turing machine and is thus Turing-equivalent. Real values for speeds and/or positions can be used as oracles and thus provide computing ability that goes beyond Turing-computation.

References

[Ada02] A. Adamatzky, editor. *Collision based computing*. Springer, 2002.

[AM95] E. Asarin and O. Maler. Achilles and the Tortoise climbing up the arithmetical hierarchy. In *FSTTCS '95*, number 1026 in LNCS, pp. 471–483, 1995.

[BNR91] N. Boccara, J. Nasser, and M. Roger. Particle-like structures and interactions in spatio-temporal patterns generated by one-dimensional deterministic cellular automaton rules. *Phys. Rev. A*, 44(2):866–875, 1991.

[Bou99] O. Bournez. Achilles and the Tortoise climbing up the hyper-arithmetical hierarchy. *Theoret. Comp. Sci.*, 210(1):21–71, 1999.

[DL97] J. Durand-Lose. Intrinsic universality of a 1-dimensional reversible cellular automaton. In *STACS '97*, number 1200 in LNCS, pp. 439–450. Springer, 1997.

[DL03] J. Durand-Lose. *Calculer géométriquement sur le plan – machines à signaux –*. Habilitation à diriger des recherches, École Doctorale STIC, Université de Nice-Sophia Antipolis, 2003. In French, http://perso.ens-lyon.fr/jerome.durand-lose/Hdr.

[DL04] J. Durand-Lose. Abstract geometrical computation: Turing-computing ability and unpredictable accumulations (extended abstract). Technical Report 2004–09, LIP, ÉNS Lyon, 46 allée d'Italie, 69 364 Lyon 7, 2004.

[DM02] M. Delorme and J. Mazoyer. Signals on cellular automata. in [Ada02], pp. 234–275, 2002.

[EN02] G. Etesi and I. Nemeti. Non-Turing computations via Malament-Hogarth space-times. *Int. J. Theor. Phys.*, 41(2):341–370, 2002. gr-qc/0104023.

[Fis65] P. C. Fischer. Generation of primes by a one-dimensional real-time iterative array. *J. ACM*, 12(3):388–394, 1965.

[Ham02] J. D. Hamkins. Infinite time Turing machines: Supertask computation. *Minds and Machines*, 12(4):521–539, 2002. math.LO/0212047.

[Hog94] M. Hogarth. Non-Turing computers and non-Turing computability. In *Biennial Meeting of the Philosophy of Science Association*, number 1, pp. 126–138, 1994.

[HSC01] W. Hordijk, C. R. Shalizi, and J. P. Crutchfield. An upper bound on the products of particle interactions in cellular automata. *Phys. D*, 154:240–258, 2001.

[Ila01] A. Ilachinski. *Cellular Automata –A Discrete Universe–*. World Scientific, 2001.

[JS90] G. Jacopini and G. Sontacchi. Reversible parallel computation: an evolving space-model. *Theoret. Comp. Sci.*, 73(1):1–46, 1990.

[Maz96] J. Mazoyer. On optimal solutions to the Firing squad synchronization problem. *Theoret. Comp. Sci.*, 168(2):367–404, 1996.

[Min67] M. Minsky. *Finite and Infinite Machines*. Prentice Hall, 1967.

[MT99] J. Mazoyer and V. Terrier. Signals in one-dimensional cellular automata. *Theoret. Comp. Sci.*, 217(1):53–80, 1999.

[VMP70] V. I. Varshavsky, V. B. Marakhovsky, and V. A. Peschansky. Synchronization of interacting automata. *Math. System Theory*, 4(3):212–230, 1970.