Journal of Computer and System Sciences 97 (2018) 182–198

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Abstract geometrical computation 8: Small machines, accumulations & rationality ☆

Florent Becker [a], Mathieu Chapelle [a], Jérôme Durand-Lose [a,b,*], Vincent Levorato [a], Maxime Senot [a]

[a] Université d'Orléans, ENSI de Bourges, LIFO ÉA 4022, F-45067 Orléans, France
[b] LIX, UMR 7161, CNRS, Ecole Polytechnique, Palaiseau, France

ARTICLE INFO

Article history:
Received 25 April 2016
Received in revised form 11 June 2018
Accepted 13 June 2018

Keywords:
Abstract geometrical computation
Accumulation
Euclidean geometry
Euclid's algorithm
Signal machine
Unconventional computing

ABSTRACT

In the context of abstract geometrical computation, computing with coloured line segments, we consider the possibility of an accumulation—topological limit point of segment intersections/collisions—with *small* signal machines, i.e. having only a very limited number of distinct slopes/speeds when started with finitely many segments/signals. The cases of 2 and 4 speeds are trivial: no machine can produce an accumulation with only 2 speeds and an accumulation can be generated with 4 speeds. The main result is the twofold 3-speed case. No accumulation can happen when all ratios between speeds and all ratios between initial distances are rational. Accumulation is possible in the case of an irrational ratio between two speeds or of an irrational ratio between two distances in the initial configuration. This dichotomy is explained by the presence of a phenomenon computing Euclid's gcd algorithm: it stops if and only if its input is commensurable, i.e., of rational ratio.

1. Introduction

Imagine a sheet of paper and some colour pencils together with ruler and compass. There is something drawn on an edge of the paper and rules are provided to extend the drawing. Depending on the rules and the initial drawing/configuration one might stop soon, or need to extend the paper indefinitely, or draw forever in a bounded part of the paper as on top of Fig. 1(a). Already in this simple setting emerge the usual questions related to dynamical systems: liveness, unbounded orbit, or convergence/accumulation.

This article concentrates on accumulation in the case of signal machines. In this setting, one direction on the drawing is distinguished and used as *time* axis. Line segments are enlarged synchronously until they intersect/collide with another one. This goes on until no more collisions can happen.

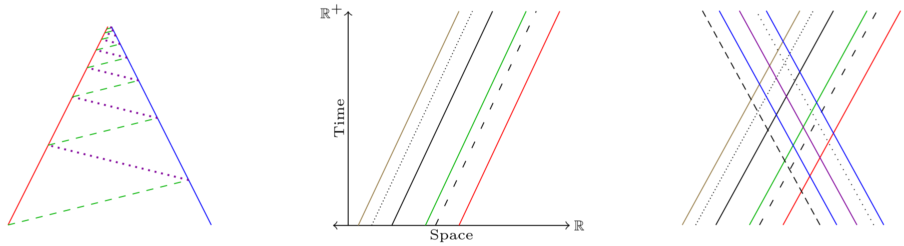
The line segments are the traces of *signals* and their intersections are *collisions* of these signals. Each signal is an instance of some *meta-signal*. At each collision, incoming signals are ended and new ones are started according to some *collision rule* defined on the meta-signals associated with the incoming signals. Signals that correspond to the same meta-signal travel at

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

* Corresponding author at: Laboratoire d'Informatique Fondamentale d'Orléans, Batiment IIIA, Rue Léonard de Vinci, B.P. 6759, F-45067 ORLEANS Cedex 2, France.

E-mail address: jerome.durand-lose@univ-orleans.fr (J. Durand-Lose).

https://doi.org/10.1016/j.jcss.2018.06.001
0022-0000/



(a) accumulation with 4 speeds (b) 1 speed: no collision (c) 2 speeds: no accumulation

Fig. 1. Basic cases.

the same speed, so the resulting traces are parallel. There are finitely many meta-signals so there are finitely many speeds and collision rules.

The signals move on a one dimensional Euclidean space orthogonal to the temporal axis (in the figures, space is horizontal and time elapses upwards). Considering the traces leads to two dimensional drawings, called *space–time diagrams*, as illustrated throughout the article. Space and time are continuous as the real line. Signals as well as collisions are dimensionless points. Computations are exact; there is no noise nor approximation.

Signal machines are computationally powerful complex systems. Accumulations (as the limit points of collisions) are easy to generate and are in fact the cornerstone to hyper-computation in the model [8]. The present article investigates the minimum size of a machine such that an accumulation is possible. Already with four meta-signals of different speeds (or directions on the drawing) an accumulation can happen as depicted on Fig. 1(a). The relevant measure is the number of different speeds and, in the case of three speeds, their values and the initial positions of the signals as explained below. One speed does not even allow any collision (see Fig. 1(b)). With two speeds, the number of collisions is finite and signals follow a grid which has no accumulation (see Fig. 1(c)).

In the three speed case, we consider a signal machine where the output of each collision rule is one signal of each speed. The generated space–time diagrams exhibit an emulation of Euclid's gcd algorithm. If any ratio between speeds or between initial positions is irrational, then that algorithm does not stop and brings forth an accumulation. Alternatively, if each of these ratio is rational, then a global halting gcd computation is generated and some global regular *mesh* emerges. Whatever the number of meta-signals and whatever the collision rules are, there is no way for any signal to escape this mesh. The diagram generated by any other machine exhibit a subset of the positions of signals and collisions of this complete machine. Hence, it cannot have an accumulation either.

State of the art. Signal machines are one of the unconventional models of computation dealing with Euclidean geometry. To name a few: Euclidean abstract machines [20,14], piece-wise constant derivatives systems [2], coloured universes [15]... In physics, [17] presents a system of point particles in a line with continuous speeds capable of displaying an infinite number of collisions in finite time.

Signal machines were originally introduced as a continuous counterpart of cellular automata where "signal" refers to any bounded periodic pattern in a (discrete) cellular automata space–time diagram. Signal machines provide a context for the underlying Euclidean reasoning often found in the cellular automata literature, as well as an abstract formalisation [18,19,7].

Signal machines are able to compute in the classical, Turing sense. This paper is somehow a companion to [10] where a Turing-universal signal machine is presented with only 13 meta-signals and 4 speeds. This research takes place in the quest for minimal computing systems, or minimal systems with undecidable (unpredictable) properties: [22,24] for Turing machines, [3,21] for cellular automata, and [16] for a more general picture.

Being in a continuous setting, signal machines can carry out analog computations in the sense of both the BSS model [1,6] and computable analysis [23,9].

Massive parallelism and the capability to approximate a fractal (with potentially infinitely many accumulations) allows for efficient solutions to hard problems: SAT for the class NP [4] and Q-SAT for PSPACE [5]. From previous work on signal machines, accumulations are known to be easy to generate. They are a powerful tool to accelerate a computation and provide hyper-computation [8]. Recently, it has been proved that, starting from a rational machine and configuration, the locations of isolated accumulations have to be *c.e* (computably enumerable) real numbers in time and *d-c.e* (difference of c.e.) real numbers in space and that any such pair can be reached [11].

While the initial version of this paper was finalised, one of the authors proved that the very same cases arises for Turing computability capability and a conference paper was rapidly published [12]. In this conference paper, the accumulation cases are presented with less formal proofs and the gcd process is only mentioned.

Outline. Formal definitions of signal machines, their dynamics, space–time diagrams, properties and normalisation are given in Sec. 2, as well as two introductory examples: computing the subtraction and the remainder of Euclidean division (later embedded inside the gcd computation). In Sec. 3, the case of two and four speeds are settled. Section 5 studies the case of three speeds, where some irrational ratio enters into play: it is possible to implant a never ending run of Euclid's algorithm.