Florent Becker, Tom Besson, Jérôme Durand-Lose, Aurélien Emmanuel, Mohammad-Hadi Foroughmand-Araabi, Sama Goliaei, and Shahrzad Heydarshahi. Abstract geometrical computation 10: An intrinsically universal family of signal machines. *ACM Trans. Comput. Theory*, 13(1):1–31, 2021. doi: 10.1145/3442359. arXiv 1804.09018.

Abstract Geometrical Computation 10: An Intrinsically Universal Family of Signal Machines*

Florent Becker[1] Tom Besson[1] Jérôme Durand-Lose[1,2] Aurélien Emmanuel[1]
Mohammad-Hadi Foroughmand-Araabi [3] Sama Goliaei[4] Shahrzad Heydarshahi[1]

March 23, 2023

[1]Univ. Orléans, INSA Centre Val de Loire, LIFO, France {florent.becker, jerome.durand-lose}@univ-orleans.fr
[2]LIX, CNRS-Inria-École Polytechnique, France
[3]Department of Mathematical Sciences, Sharif University of Technology, Tehran, Iran foroughmand@sharif.ir
[4]Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran sgoliaei@ut.ac.ir

Abstract

Signal machines form an abstract and idealised model of collision computing. Based on dimensionless signals moving on the real line, they model particle/signal dynamics in Cellular Automata. Each particle, or *signal*, moves at constant speed in continuous time and space. When signals meet, they get replaced by other signals. A signal machine defines the types of available signals, their speeds and the rules for replacement in collision.

A signal machine $\mathcal{A}$ simulates another one $\mathcal{B}$ if all the space-time diagrams of $\mathcal{B}$ can be generated from space-time diagrams of $\mathcal{A}$ by removing some signals and renaming other signals according to local information. Given any finite set of speeds $\mathcal{S}$, we construct a signal machine that is able to simulate any signal machine whose speeds belong to $\mathcal{S}$. Each signal is simulated by a *macro-signal*, a ray of parallel signals. Each macro-signal has a main signal located exactly where the simulated signal would be, as well as auxiliary signals which encode its id and the collision rules of the simulated machine.

The simulation of a collision, a *macro-collision*, consists of two phases. In the first phase, macro-signals are shrunk, then the macro-signals involved in the collision are identified and it is ensured that no other macro-signal comes too close. If some do, the process is aborted and the macro-signals are shrunk, so that the correct macro-collision will eventually be restarted and successfully initiated. Otherwise, the second phase starts: the appropriate collision rule is found and new macro-signals are generated accordingly.

Considering all finite set of speeds $\mathcal{S}$ and their corresponding simulators provides an intrinsically universal family of signal machines.

Key-Words

Abstract Geometrical Computation; Collision computing; Intrinsic universality; Signal machine; Simulation

1 Introduction

Signal Machines (SM) arose as a continuous abstraction of Cellular Automata (CA) [Durand-Lose, 2008]. In dimension one, the dynamics of Cellular Automata are often described as signals interacting in collisions resulting in the generation of new signals. Signals store and transmit information to start a process, to

*The authors are thankful to the Franco-Iranian PHC Gundishapur 2017 number 38071PC "Dynamique des machines à signaux" for funding this research.

synchronise, etc. The use of signals in the context of CA is widespread in the literature: collision computing [Adamatzky, 2002], gliders Jin and Chen [2016], solitons [Jakubowski et al., 1996, 2000, 2017, Siwak, 2001], particles [Boccara et al., 1991, Mitchell, 1996, Hordijk et al., 1998], Turing-computation [Lindgren and Nordahl, 1990, Cook, 2004], synchronisation [Varshavsky et al., 1970, Yunès, 2007], geometrical constructions [Cook, 2004], signals [Mazoyer and Terrier, 1999, Delorme and Mazoyer, 2002], etc.

In signal machines, signals are dimensionless points moving on a 1-dimensional Euclidean space in continuous time. They have uniform movement and thus draw line segments on space-time diagrams. Each signal is an instance of a *meta-signal* among a finite given set of meta-signals. As soon as two or more signals meet, a collision happens: incoming signals are instantly replaced by outgoing signals according to *collision rules*, depending on the meta-signals of the incoming signals. In-between collisions, signals propagate at some uniform speed depending on their meta-signal.

A signal machine is defined by a finite set of meta-signals, a function assigning a speed to each meta-signal (negative for leftward), and a set of collision rules. A collision rule associates a set of at least two meta-signals of different speeds (*incoming*) with another set of meta-signals of different speeds (*outgoing*). Collision rules are deterministic: a set appears at most once as the incoming part of a collision rule.

In any configuration, there are finitely many signals and collisions located at distinct places on the real line. The aggregation of the configurations reachable from some (initial) configuration forms a two dimensional space-time diagram like the one in Fig. 1a in which the traces of signals are line segments. Signals corresponding to the same meta-signal have the same speed: their traces are parallel segments (like the dotted μ_2). Collisions provide a discrete time scale and a directed acyclic graph structure inside each space-time diagram. This emphasises the hybrid aspect of SM: continuous steps separated by discrete steps.

Signal machines are known to be able to compute by simulating Turing machines and even to hypercompute [Durand-Lose, 2012]. As an analog model of computation they correspond exactly to the linear BSS model [Blum et al., 1989, Durand-Lose, 2007].

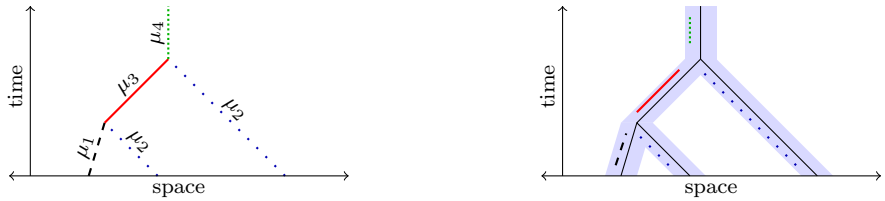
As with any computing dynamical system, it is natural to ask whether there is a signal machine which is able to simulate all signal machines. Intrinsic universality (being able to simulate any device of its own kind) is an important property, since it means to represent all machines and to exhibit all the behaviours available in the class. In computer science, the existence of (intrinsically) universal Turing machines is the cornerstone of computability theory. Many computing systems have intrinsically universal instances: the (full) BSS model, Cellular Automata (CA) [Albert and Čulik II, 1987, Mazoyer and Rapaport, 1998, Ollinger, 2001, 2003, Goles Ch. et al., 2011], reversible CA [Durand-Lose, 1995], quantum CA [Arrigh and Grattage, 2012], some tile assembly models at temperature 2 [Doty et al., 2010, Woods, 2013], etc. Some tile assembly models at temperature 1 [Meunier et al., 2014] or causal graph dynamics [Martiel and Martin, 2015] admit infinite intrinsically universal families but no single intrinsically universal instance.

One key characteristic of intrinsic universality is that it is expected to *simulate according to the model*. Transitive simulations across models are not enough: simulating a TM that can simulate any rational signal machine totally discards relevant aspects of the models such as directed acyclic graph representation, relative location, spatial positioning, energy levels, etc.

It should be noted that although instances of signal-based systems with Turing-computation capability are very common in the literature [Lindgren and Nordahl, 1990, Cook, 2004], to our knowledge, the present paper provides the first result about intrinsic universality in a purely signal-based continuous system.

For Cellular Automata, simulation and intrinsic universality can be defined with an operation of *grouping* on space-time diagrams Mazoyer and Rapaport [1998], Ollinger [2003]. This operation consists of creating a space-time diagram from another one by applying a local function on blocks of the former. Because Cellular Automata are discrete, it is possible to consider the domain of this local function to be finite.

With signal machines, because space is continuous and there is no canonical scale within a diagram, decoding a space-time diagram is done by uniformly applying a *local* decoding function on each point of each configuration of the diagram, rather than having grouping and blocks. The notion of locality for the decoding function of the space-time diagram has to be defined by stating that the decoding function should only look at a uniformly-bounded amount of signals around a collision. Because signal machines lack the discrete time-steps of Cellular Automata, a special handling of the initial configuration is necessary, somewhat like



(a) simulated space-time diagram

(b) simulating space-time diagram

Figure 1: Simulation scheme.

with self-assembling systems Doty et al. [2012].

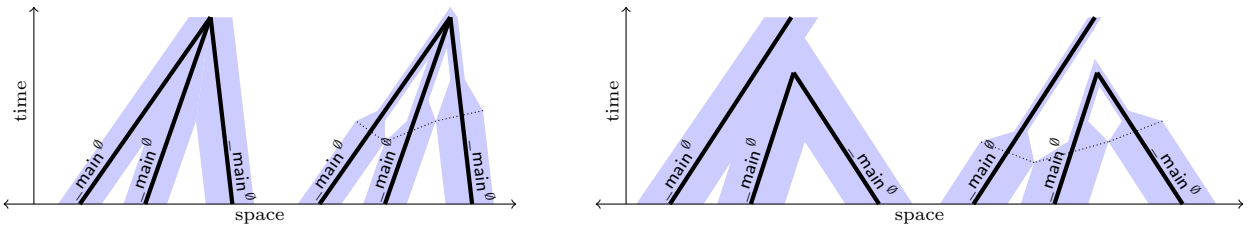
Having defined a fitting concept of simulation, the present paper provides, for any finite set of speeds $\mathcal{S}$, a signal machine capable of simulating all signal machines which only use speeds in $\mathcal{S}$.

In a simulation by one of our universal signal machines, each signal of the simulated SM is replaced by a ray of signals (shaded in Fig. 1b) called a *macro-signal*. Each macro-signal has a non-zero width and contains a $\_\mathsf{main}^{\emptyset}$ signal (black in the middle) which is exactly positioned as the simulated signal. The meta-signal (dot, dash or thick in Fig. 1a) is encoded within the macro-signal (greyed zone), as illustrated in Fig. 1b, which is then used by the decoding function to recover the meta-signal.

Each macro-signal encodes its identity in unary together with the list of all the collision rules of the simulated signal machine. Notice that at any time, the amount of information in the macro-signal is bounded. Macro-collisions are handled locally.

The main challenge is that macro-signals and macro-collisions have non-zero width and might overlap and disturb one another. Figure 2 illustrates this problem. In Fig. 2a all three present macro-signals rightfully interact whereas in Fig. 2b the leftmost one should not participate. In the same spirit, once a collision resolution is started, other signals should be far away enough not to intersect the zone needed for its resolution.

To cope with this, as soon as the borders of two macro-signals touch, both are shrunk in order to "delay" the macro-collision resolution (right part in Figs. 2a and 2b). This delay is to be understood relative to the width of the input macro-signals: the time of the collision is not changed, but after shrinking the input signals, it becomes a larger multiple of the width of each input signal. This delay is used to check which macro-signals exactly enter the macro-collision and to ensure that non-participating macro-signals are far away enough. This checking identifies macro-signals participating in the ongoing collision and ensures that no other signal may collide with control signals, not before the collision nor after a while after collision. This zone in which no other signal might enter is called the *safety zone*.



(a) all in one collision

(b) messing macro-signals get isolated

Figure 2: The effect of shrinking (on right of each case).

If any constrain is not satisfied, the macro-collision aborts; nothing happens but the macro-signals have been shrunk and thus relatively spaced. Later on, testing will be restarted as these thinner macro-signals touch again. Eventually all correct macro-collisions will happen.

If all constraints are satisfied, the macro-collision is resolved. This is done by gathering information of id's

of all participating macro-signals and finding the appropriate collision rule. After actual collision between $\_\mathsf{main}^-$ signals, according to the selected collision rule, macro-signals are replaced by new macro-signals representing output signals of the simulated SM.

The different phases and their relative duration in a successful macro-collision are presented in Fig. 3 where percentages are taken relative to the duration from the collision of ${}_i\mathsf{border\text{-}right}$ and ${}_j\mathsf{border\text{-}left}$ to the exact location of the collision, i.e. the meeting of ${}_i\mathsf{main}^\emptyset$ and ${}_j\mathsf{main}^\emptyset$. These proportions are arbitrary, the only condition is that the macro-collision resolution is started before the signals ${}_i\mathsf{border\text{-}right}$ and ${}_j\mathsf{border\text{-}left}$ met again. The duration of the shrinking phase (10 %) is fixed. The duration of the test and check phase is at most 20 %, for success as well as failure. It is ensured that aborting or disposal is carried out before any two present macro-signals meet again (and initiate a different macro-collision).

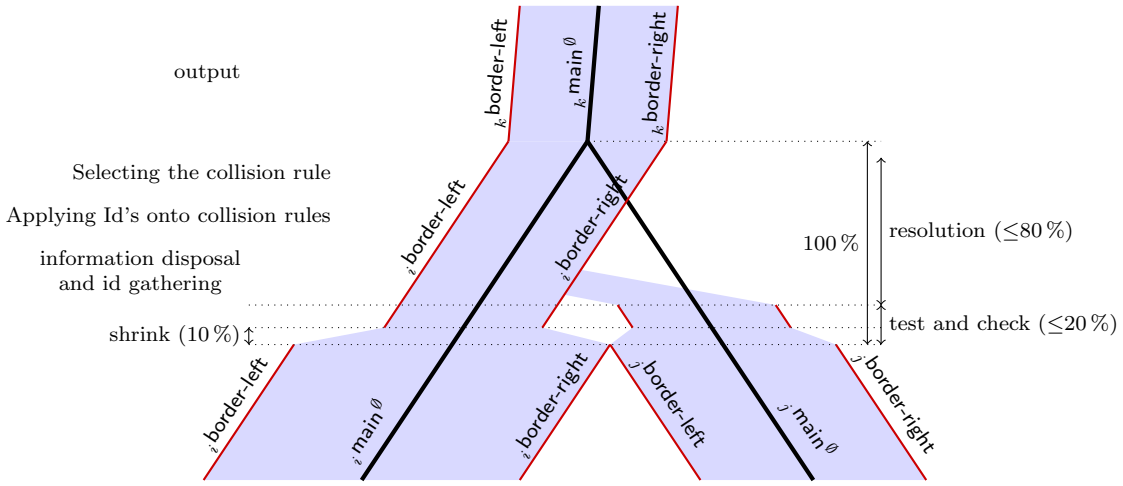


Figure 3: Phases of a successful macro-collision.

This way the constructed signal machine is able to simulate any signal machine with speeds included in a given set. By varying this set, an intrinsically universal family of signal machines is obtained.

All definitions are gathered in Sect. 2. The encoding is detailed in Sect. 3. Macro-collision resolution is explained in Sect. 4; the testing prior to it is found in Sect. 5. Section 6 provides some simulation examples. Conclusion, remarks and perspectives are gathered in Sect. 7.

2 Definitions

A signal machine regroups the definitions of its meta-signals and their dynamics: rewriting rules at collisions and constant speed in-between.

Definition 1. A *signal machine* (SM) $\mathcal{A}$ is a triplet (M, S, R) such that: M is a finite set of *meta-signals*; $S : M \to \mathbb{R}$ is the *speed function* (each meta-signal has a constant speed); and R is a finite set of *collision rules* which are denoted by $\rho^- \to \rho^+$ where ρ^- and ρ^+ are sets of meta-signals of distinct speeds. Each ρ^- must have at least two meta-signals. R is deterministic: all ρ^- are different.

Let V be the set $M \cup R \cup \{\oslash\}$. A *($\mathcal{A}$-)configuration* c, is a map from $\mathbb{R}$ to V, that is from the points of the real line to either a meta-signal, a rule or the value $\oslash$ (indicating that there is nothing there), with only finitely many non-$\oslash$ locations in any configuration.

A signal machine evolution is defined in terms of dynamics. If there is a signal of speed s at x, then after a duration Δt its position is $x + s{\cdot}\Delta t$, unless it enters a collision before. At a collision, all incoming signals are instantly replaced according to rules by outgoing signals in the following configurations. This is formalised as follows.

To simplify notations, the relation *issued from*, $\ltimes \subset M \times V$, is defined to be true only in the following cases:

- $\mu \ltimes \mu$, $\forall \mu \in M$ and
- $\mu \ltimes \rho$, $\forall \rho \in M$ such that $\mu \in \rho^+$.

The relation $\ltimes$ means "is equal to (some meta-signal) or belongs to the output of (a collision)".

Definition 2 (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:

$$\exists x_1, x_2 \in \mathbb{R}, \exists \mu_1, \mu_2 \in M \begin{cases} x_1 + d{\cdot}S(\mu_1) = x_2 + d{\cdot}S(\mu_2) \\ \wedge\, \mu_1 \ltimes c(x_1) \\ \wedge\, \mu_2 \ltimes c(x_2) \end{cases}.$$

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

Let c_t be the configuration at time t.

For t' $(t < t' < t{+}\Delta(c_t))$, the configuration at t' is defined as follows. First, signals are set according to $c_{t'}(x) = \mu$ iff $\mu \ltimes c_t\,(x + (t{-}t'){\cdot}S(\mu))$. There is no collision to set (t' is before the next collision) thus no ambiguity. The rest is $\oslash$.

For the configuration at $t + \Delta(c_t)$, collisions are set first: $c_{t+\Delta(c_t)}(x) = \rho^- \to \rho^+$ where $\rho^- = \{\, \mu \in M \,|\, \mu \ltimes c_t(x - \Delta(c_t){\cdot}S(\mu)) \,\}$. Note that, in a configuration points are assigned to signals, collision rules, or empty set. Here, the collision point is assigned to the collision rule. Then meta-signals are set (with above condition) where there is not already a collision, and finally $\oslash$ everywhere else.

The dynamics are uniform in both space and time. Since configurations are finite, the infimum is non-zero and is reached.

A *space-time diagram* of $\mathcal{A}$ is the aggregation of configurations as times elapses, that is, a function from $\mathbb{R}^+$ into the set of configurations of $\mathcal{A}$. It forms a two dimensional picture (time is always elapsing upwards in the figures). It is denoted $\mathbb{A}$ or $(\mathcal{A}, c)$ to emphasise the signal machine and the initial configuration.

2.1 Simulations among signal machines

In this subsection, we define formally what the sentence "signal machine $\mathcal{A}$ simulates signal machine $\mathcal{B}$" entails.

Local functions on configurations First, let us define how to recover a configuration of a signal machine from a configuration of another one–a putative simulator.

Let $\mathcal{A}$ be a signal machine, and $\mathcal{C}$ be the set of all its configurations. Let $\mathcal{C}^* = \{\, c \in \mathcal{C} \,|\, c(0) \neq \oslash \,\}$. Let $c \in \mathcal{C}^*$, $\mathsf{symb}(c)$ is the bi-infinite word defined by: for all $n \in \mathbb{Z}$, $\mathsf{symb}(c)_n$ is the n-th non-$\oslash$ value in c, counting from position 0 in both directions. For example, $\mathsf{symb}(c)_0 = c(0)$ and $\mathsf{symb}(c)_{-1}$ is the first non-$\oslash$ value leftward from position 0. Pose $\mathsf{symb}(c)_n = \perp$ if c does not have enough non-$\oslash$ values.

Let f be a function from $\mathcal{C}$ to some set F. The function f is *local* if the following hold:

- there is $f_\oslash \in F$ such that when $c(0) = \oslash$, $f(c) = f_\oslash$,
- there is a function $\hat{f}$ on bi-infinite words such that $f(c) = \hat{f}(\mathsf{symb}(c))$ when $c \in \mathcal{C}^*$, and
- there is $n \in \mathbb{N}$ such that $\hat{f}$ only depends on the n symbols around 0 in its input word.

In the analogy with Cellular Automata, local functions will play a role similar to that of grouping functions.

Signal machine simulation Let $\mathcal{A}$ and $\mathcal{B}$ be two signal machines; subscripts are used to identify the machine an element belongs to. Let interp be a local function from the configurations of $\mathcal{B}$ into $V_\mathcal{A}$ such that $\mathsf{interp}_\oslash = \oslash_\mathcal{A}$. For a configuration b of $\mathcal{B}$, we define the configuration $\mathsf{interp}^*(b)$ by $\mathsf{interp}^*(b)(x)$ to be value of interp^* on the configuration b shifted by $-x$, i.e. by centering it at position position x. Schematically, interp interprets sequences of signals centered around position 0 and interp^* uses it (composed with translations) to rebuild whole configurations of the simulated signal machine.

A *representation* function is a function repr from $(M_\mathcal{A} \cup R_\mathcal{A})$ to $\mathcal{C}_\mathcal{B}$, such that the support of $\mathsf{repr}(\cdot)$ is always included in the interval $[-1, 1]$. Let x be any position and d be any positive number, we denote $\mathsf{repr}(\cdot, x, d)$ the configuration $\mathsf{repr}(\cdot)$ scaled by $1/3d$ and shifted by x; its support is included in $[x-d/3, x+d/3]$. One such function will be used for encoding the signals and collisions making up the initial configuration. This function yields a configuration for each signal or collision. For a configuration a, let $d_a(x) = \min\{\, |d| \mid a(x+d) \neq \oslash \wedge d \neq 0 \,\}$. Then, we note $\mathsf{repr}^*(a)$ for the union of the $\mathsf{repr}(a(x), x, d_a(x))$ for $a(x) \neq \oslash$. Note that because of these translations and scalings, there are no collisions between the different $\mathsf{repr}(\cdot)$.

Definition 3 (Simulation between signal machines)**.** Let $\mathcal{A} = (M_\mathcal{A}, S_\mathcal{A}, R_\mathcal{A})$ and $\mathcal{B} = (M_\mathcal{B}, S_\mathcal{B}, R_\mathcal{B})$ be two signal machines. Let interp be a local function from the configurations of $\mathcal{B}$ into $V_\mathcal{A}$. Let repr be a representation function.

For a diagram $\mathbb{A} = (\mathcal{A}, a)$ of $\mathcal{A}$, let $\mathbb{B}_a = (\mathcal{B}, \mathsf{repr}^*(a))$ be the diagram of $\mathcal{B}$ on initial configuration $\mathsf{repr}^*(a)$. Then $\mathcal{B}$ *simulates* $\mathcal{A}$ if:

$$\forall a, \forall t \geq 0, \mathsf{interp}^*(\mathbb{B}_a(t)) = \mathbb{A}(t) \ .$$

The reader familiar with simulations and intrinsic universality in Cellular Automata may wonder what purpose the repr function serves. In Cellular Automata, there is a time grouping as well as a space grouping: when some CA B simulates a CA A, there is a k such that for all i, the configuration of B at time $k \cdot i$ represents the configuration of A at time i, and indeed, any configuration of B seen at a time multiple of k can be used as an initial configuration for simulating the corresponding configuration of A. In a signal machine, having such a periodicity would require the spacing of any auxiliary signals in b to be uniform. This in turn would require that there be a positive lower bound to the distance between two signals, uniformly over configurations of $\mathcal{A}$, which cannot be the case. Instead, a simulator uses repr to get an encoding of each signal and collision at time 0. A similar distinction is necessary in the definition of simulation for self-assembling systems Doty et al. [2012], for the same reason of asynchronicity.

The following theorem comes as a sanity check of the definition of simulation. Its proof, while straightforward will allow us to explain how each part of the definition comes into play in more complex proofs.

Lemma 4. *For any signal machine M, M simulates M.*

Proof. First, the definition of simulation calls for the interp function. This function indicates which signal of M (as the simulatee) is represented by a given local configuration of M (as the simulator). This only depends on the signal at that very point, so define interp by: $\mathsf{interp}(b) = b(0)$.

Consider some position x and a configuration b, and examine $\mathsf{interp}^*(b)(x)$. By definition, $\mathsf{interp}^*(b(x)) = (y \mapsto b(y+x))(0) = b(x)$. Thus, each meta-signal represents itself locally by interp, and this extends to the level of configurations where $\mathsf{interp}^*(b(x)) = b(x)$.

Second, the repr function needs to be defined. For this, set that any meta-signal is represented by a copy of itself in an otherwise empty configuration. That is, for any meta-signal μ, set $\mathsf{repr}(\mu)(0) = \mu$, and $\mathsf{repr}(\mu)(y) = \oslash$ for $y \neq 0$. Then, as above, we then have for any x, $\mathsf{repr}^*(a)(x) = a(x)$.

Finally, let $\mathbb{A}_a = (\mathcal{A}, \mathsf{repr}^*(a))$. Then for each $t \geq 0$ and each a we have $\mathsf{interp}^*(\mathbb{A}_a(t)) = \mathbb{A}_a(t) = (\mathcal{A}, \mathsf{repr}^*(a)) = (\mathcal{A}, a)$. □

Example 5. Let $\mathcal{A} = (M_\mathcal{A}, S_\mathcal{A}, R_\mathcal{A})$ and $\mathcal{B} = (M_\mathcal{B}, S_\mathcal{B}, R_\mathcal{B})$ be two signal machines such that $M_\mathcal{A} = (\{\mu_1\ , \mu_2, \mu_3\}, \{S(\mu_1) = -1, S(\mu_2) = S(\mu_3) = 1, \}, R_\mathcal{A} : \{\rho_4 : \{\mu_1, \mu_2\} \to \{\mu_3\}\}, \{\rho_5 : \{\mu_1, \mu_3\} \to \{\mu_2\}\})$ and $M_\mathcal{B} = (\{\mu_a\ , \mu_b\}, \{S(\mu_a) = -1, S(\mu_b) = 1\}, R_\mathcal{B} : \{\rho_c : \{\mu_a, \mu_b\} \to \{\mu_b\}\})$. We want to show that $\mathcal{B}$ simulates $\mathcal{A}$. Informally, this holds by collapsing μ_2 and μ_3 into one signal, and observing that this yields the meta-signals speeds and collision rules of $\mathcal{B}$.

We define two functions interp and repr to turn this intuition into an instance of the formal definition of simulation.

We want that for all diagrams $\mathbb{A}$ of $\mathcal{A}$, $\mathsf{interp}^*(\mathbb{A})(x,t) = \mu_a$ if $\mathbb{A}(x,t) = \mu_1$, $\mathsf{interp}^*(\mathbb{A})(x,t) = \mu_b$ if $\mathbb{A}(x,t) = \mu_2$ or μ_3, and $\mathsf{interp}^*(\mathbb{A})(x,t) = \rho_c$ if $\mathbb{A}(x,t) \in \{\rho_4, \rho_5\}$.

We take $\mathsf{repr}(\mu_a)(x) = \mu_1$ if $x = 0$, otherwise $\oslash$; $\mathsf{repr}(\mu_b)(x) = \mu_2$ if $x = 0$, otherwise $\oslash$; $\mathsf{repr}(\rho_c)(x) = \rho_4$ if $x = 0$, otherwise $\oslash$.

Also, $\mathsf{interp}(a) = \oslash$ if $a(0) = \oslash$, $\mathsf{interp}(a) = \mu_a$ if $a(0) = \mu_1$, $\mathsf{interp}(a) = \mu_b$ if $a(0) = \mu_2$, $\mathsf{interp}(a) = \mu_b$ if $a(0) = \mu_3$, $\mathsf{interp}(a) = \rho_c$ if $a(0) = \rho_4$ and $\mathsf{interp}(a) = \rho_c$ if $a(0) = \rho_5$.

If $\mathbb{A}$ is a space-time diagram of $\mathcal{A}$, then $\mathsf{interp}^*(\mathbb{A})$ is a space-time diagram of $\mathcal{B}$, because the collisions are preserved by interp^*.

Also, for each s, $\mathsf{interp}^*(\mathbb{A})$ is a space-time diagram of $\mathcal{A}$ and has initial configuration s.

So $\mathsf{interp}^*(\mathbb{A}_s) = \mathbb{B}_s$, and we have our simulation.

2.2 Intrinsically universal machines

Definition 6 ($\mathcal{S}$-(intrinsic) universality)**.** Let $\mathcal{S}$ be a set of speeds. A signal machine $\mathcal{U}_\mathcal{S} = (M_U, \mathcal{S}_\mathcal{U}, R_U)$, is $\mathcal{S}$*-universal* if, for any $\mathcal{A}$ with speeds in $\mathcal{S}$, there is a local function $\mathsf{interp}_\mathcal{A}$ and a representation function $\mathsf{repr}_\mathcal{A}$ such that $\mathcal{U}_\mathcal{S}$ simulates $\mathcal{A}$ through $\mathsf{interp}_\mathcal{A}$ and $\mathsf{repr}_\mathcal{A}$.

For the rest of the paper, we fix $\mathcal{S} = \{s_1, \ldots, s_\mathfrak{n}\} \subset \mathbb{R}$ a set of speeds, with $s_1 < \ldots < s_\mathfrak{n}$ and provide the construction of an $\mathcal{S}$-universal machine $\mathcal{U}_\mathcal{S}$. Its set of speeds, its set of meta-signals and its rules only depend on $\mathcal{S}$. The functions $\mathsf{interp}_\mathcal{A}$ and repr will be defined along with the construction of $\mathcal{U}_\mathcal{S}$. By default, $\mathsf{interp}_\mathcal{A}(x)$ is undefined; in the following, we will list the cases where $\mathsf{interp}_\mathcal{A}(x)$ is defined. The number of signals that $\mathsf{interp}_\mathcal{A}$ actually reads is bounded for each $\mathcal{A}$. The definition of repr will be given in Sect. 3.2.

In the rest of the paper, not all collision rules of intrinsically undefined SM are explicitly defined. They can be found online in the simulation. For a set of meta-signals s which is not explicitly defined as the input of a collision rule, a collision rule ρ_s is implicitly defined thus:

- if $|s| = 2$, then $\rho^-_s = \rho^+_s = s$
- if there is a unique explicit collision rule ρ such that $\rho^- \subset s$, then $\rho^-_s = s$, and $\rho^+_s = \rho^+ \cup (s \setminus \rho^-)$,
- if, for all set of rules $\{\rho_1, \ldots, \rho_k\}$ such that $\bigcup_{1\leq i\leq k} \rho^-_i = s$, $\bigcup_{1\leq i\leq k} \rho^+_i$ is the same set ρ^+, then $\rho^-_s = s$ and $\rho^+_s = \rho^+$,
- otherwise ρ_s is undefined.

In other words, unlisted rules with two inputs are blank, they output the same signals as in input. Unlisted rules can also be the "superposition" of one meta-signal in both input and output of a defined collision rule (this covers the case when a defined collision happens exactly on some irrelevant signal). It might also happen that two or more unrelated collisions happen at the same position and share some input and output signal or that two consecutive collisions become synchronous. The definition of the corresponding collision rules is straightforward from the defined collision rules and can be considered as limit cases, in the sense that a small perturbation of the input configuration gets rid of them. They are not listed to avoid unnecessary listings of collision rules.

3 Encoding of signals and signal machines

3.1 Meta-signal notation

In the rest of the paper the names of $\mathcal{U}_\mathcal{S}$-meta-signals are organised around a base name—in sans-serif font—decorated with parameters:

$$ {}_{\mathsf{a}}\mathsf{base}^{\mathsf{d}}_{\mathsf{c}} \qquad \text{or} \qquad {}^{\mathsf{b}}\mathsf{base}^{\mathsf{d}}_{\mathsf{c}} \ . $$

A signal noted ${}_{\mathsf{a}}\mathsf{base}^{\mathsf{d}}_{\mathsf{c}}$, is instantiated for speed $s_{\mathsf{a}} \in \mathcal{S}$, and its actual speed is s_{a}. A signal noted ${}^{\mathsf{b}}\mathsf{base}^{\mathsf{d}}_{\mathsf{c}}$, is instantiated for speed $s_{\mathsf{b}} \in \mathcal{S}$, but its actual speed is not s_{b}, but some other speed, generally computed from b, c and d. We use $\_\mathsf{base}^{\mathsf{d}}_{\mathsf{c}}$ or ${}^{-}\mathsf{base}^{\mathsf{d}}_{\mathsf{c}}$ when the actual value of the parameter is not relevant. A signal noted $\mathsf{base}^{\mathsf{d}}_{\mathsf{c}}$ belongs to a family that is not parametrized by speeds in $\mathcal{S}$. For example, ${}_1\mathsf{main}^{\emptyset}$ and ${}_2\mathsf{main}^{\emptyset}$ are different meta-signals of respective speeds s_1 and s_2 but with the same meaning, with respect to s_1 and s_2 respectively.

Parameters c and d are used to hold a finite amount of information.

3.2 Encoding of signals and the repr and interp functions

Let ${}_i\mu^{\sigma}$ be any meta-signal of $\mathcal{A}$. The integer i indicates that the speed of ${}_i\mu^{\sigma}$ speed is s_i and σ $(0 < \sigma)$ is its index in some numbering of the meta-signals of $\mathcal{A}$ of speed s_i.

We define $\mathsf{repr}({}_i\mu^{\sigma})$ as a so-called *macro-signal*, i.e. a configuration with finite support, delimited by two parallel signals, here ${}_i\mathsf{border\text{-}left}$ and ${}_i\mathsf{border\text{-}right}$. The space used by a macro-signal is called its *support zone.*

The configuration $\mathsf{repr}({}_i\mu^{\sigma})$ contains the following signals in the following order:

$$ {}_i\mathsf{border\text{-}left} \quad ({}_i\mathsf{id})^{\sigma} \quad {}_i\mathsf{main}^{\emptyset} \quad {}_i\text{<rules encoding>} \quad {}_i\mathsf{border\text{-}right} \ . $$

The left part is thus made of σ parallel signals of speed s_i encoding σ in unary.

The relative position of signals of $\mathsf{repr}({}_i\mu^{\sigma})$ are defined with ${}_i\mathsf{border\text{-}left}$ at $-1 - \frac{s_i + s^{\mathsf{max}}}{s^{\mathsf{rapid}}}$, ${}_i\mathsf{border\text{-}right}$ at $1 - \frac{s_i + s^{\mathsf{max}}}{s^{\mathsf{rapid}}}$, ${}_i\mathsf{main}^{\emptyset}$ at 0, and other signals regularly spaced between them (before the rescaling and translation done by repr^*; the value of δ is given in section 4); s^{max} is the maximum absolute value of speeds of $\mathcal{S}$ and s^{rapid} is a very fast speed defined in section 4.2. The reason for this choice is explained in section 4.4.

The value of repr on collisions (rather than signals) is defined in Sect. 4.4.

All the rules of $\mathcal{A}$ are encoded between ${}_i\mathsf{main}^{\emptyset}$ and ${}_i\mathsf{border\text{-}right}$, one after the other. Each rule (to be read from the right) is encoded as a *then-part* followed by an *if-part*:

$$ {}_i\mathsf{rule\text{-}bound} \quad ({}_i\mathsf{then}_1)^* \ \cdots \ ({}_i\mathsf{then}_{\mathfrak{n}})^* \quad {}_i\mathsf{rule\text{-}middle} \quad ({}_i\mathsf{if}_1)^* \ \cdots \ ({}_i\mathsf{if}_{\mathfrak{n}})^* \quad {}_i\mathsf{rule\text{-}bound} \ . $$

Let ρ_{max} be the collision rule of $\mathcal{A}$ where ρ^{-}_{max} is the set of all meta-signals with maximum id for each speed. The encoding of ρ_{max} has to appear in the rules. This is a technicality to ensure a correct decoding later one.

The number of ${}_i\mathsf{if}_l$ signals between ${}_i\mathsf{rule\text{-}middle}$ and ${}_i\mathsf{rule\text{-}bound}$ corresponds to the index of the $\mathcal{A}$-meta-signal of speed s_l which is expected as input by this rule (or zero for no s_l-speed meta-signal). Then, the number of ${}_i\mathsf{then}_l$ between ${}_i\mathsf{rule\text{-}bound}$ and ${}_i\mathsf{rule\text{-}middle}$ corresponds to the index of the $\mathcal{A}$-meta-signal of speed s_l which is output by this rule. Figure 4 provides an example of a rule encoding.

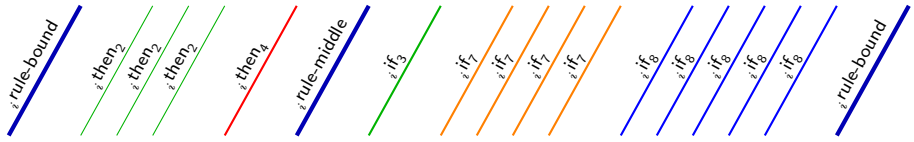


Figure 4: Encoding of the rule $\{\, {}_3\mu^1,\ {}_7\mu^4,\ {}_8\mu^5 \,\} \to \{\, {}_2\mu^3,\ {}_4\mu^1 \,\}$ in the direction i.

All the needed meta-signals are defined in Fig. 5. Later in the construction, the empty set in ${}_i\mathsf{main}^{\emptyset}$ is replaced by a subset of $[\![1, \mathfrak{n}]\!]$ to store the directions to output after a collision.

Let $\mathfrak{w}$ be the maximum number of signals in $\mathsf{repr}({}_i\mu^{\sigma})$ for ${}_i\mu^{\sigma}$ a signal of $\mathcal{A}$. The function interp looks at at most $\mathfrak{n}.(\mathfrak{w}+2)$ non-$\oslash$ values on each side. Except for the symbol at the center of the configuration (i.e. $c(0)$), whenever a collision rule is encountered, interp virtually replaces it by its input signals, in the reverse order of their speeds: it effectively simulates the configuration at time $t - \epsilon$ for small enough ϵ. Thus, from now on, we define interp according to signals only (except for $c(0)$).

Meta-signal	speed	Meta-signal	speed
$\forall i \in [\![1,\mathfrak{n}]\!]$, ${}_i\mathsf{border\text{-}left}$	s_i	$\forall i \in [\![1,\mathfrak{n}]\!]$, ${}_i\mathsf{rule\text{-}bound}$	s_i
$\forall i \in [\![1,\mathfrak{n}]\!]$, ${}_i\mathsf{id}$	s_i	$\forall i,l \in [\![1,\mathfrak{n}]\!]$, ${}_i\mathsf{then}_l$	s_i
$\forall i \in [\![1,\mathfrak{n}]\!]$, $\forall E \subseteq [\![1,\mathfrak{n}]\!]$, ${}_i\mathsf{main}^E$	s_i	$\forall i \in [\![1,\mathfrak{n}]\!]$, ${}_i\mathsf{rule\text{-}middle}$	s_i
$\forall i \in [\![1,\mathfrak{n}]\!]$, ${}_i\mathsf{border\text{-}right}$	s_i	$\forall i,l \in [\![1,\mathfrak{n}]\!]$, ${}_i\mathsf{if}_l$	s_i

Figure 5: Meta-signals for encoding.

We now define its value on some configurations, according to this section. This definition will be completed in later sections, as more meta-signals and collisions of $\mathcal{U}_\mathcal{S}$ are defined. First, $\mathsf{interp}(c)$ is defined to be ${}_i\mu^\sigma$ if $c(0)$ is ${}_i\mathsf{main}^\emptyset$, and the closest signals to 0 are in a configuration that is compatible with the neighbourhood of *main* in $\mathsf{repr}({}_i\mu^\sigma)$. This only depends on the $\mathfrak{w}$ signals closest to 0 on each side. When this is the case, we say that the configuration is *clean* at 0. A configuration c is clean at position x if its translation by $-x$ is clean at 0. For a configuration c which is clean at x, we define $\mathsf{width}(c,x)$ to be width of the macro-signal at x, that is the distance between the signals _border-left ${}_i$border-right- closest to 0.

4 Macro-collision resolution

In the rest of the paper, we define a set of meta-signals and rules to deal with collisions in $\mathcal{A}$. The simulation of a collision has two phases: a check phase, which is presented in Sect. 5, and a resolution phase which is presented in this section. The simplest space-time diagrams with at least one collision are the ones with exactly one collision, and all signals in the starting configuration are inputs of that collision. This section presents the sufficient machinery for dealing with this case, and Sect. 5 completes it in order to be able to deal with a fully general diagram.

The resolution of a macro-collision is done in four consecutive stages as illustrated in Fig. 3:

Information disposal and id gathering (Sect. 4.1). Useless signals are removed (e.g. the encoding of the collision rules is kept in the leftmost macro-signal and deleted in all other incoming macro-signals). The id number of each macro-signal is sent to the copy of the encoded rules in the leftmost macro-signal.

Applying id's onto collision rules (Sect. 4.2). For each speed involved in the collision, the id number signals try to mark the same number in each collision rule *if-part*. In every rule, if the numbers do not match, the rule is marked `failed` or an unmarked signal will remain.

Selecting the collision rule (Sect. 4.3). A signal crosses the rule encoding looking for a rule where all signals are marked and `failed` is not present. When such a rule is found, its *then-part* is sent on the left as the id's to broadcast. It also collects the slopes for the outputs and stores them onto the _$\mathsf{main}^\emptyset$ signal.

Output (Sect. 4.4). Macro-signals are broadcast in each output direction. Each of them has on the left the id numbers and on the right a clean table of collision rules.

Let ρ be a collision rule of $\mathcal{A}$, let $j = i_0 < i_1 < \ldots < i_{|\rho^-|-1} = i$ be the integers such that these are exactly the indices of the speeds in ρ^-. Let a be a configuration of $\mathcal{A}$ whose signals are exactly one of each meta-signal in ρ^- and the positions $x_{i_0} < x_{i_1}, \ldots < x_{i_{|\rho^-|-1}}$ of these signals are such that they all meet at some point (x,t).

Let b' be a configuration which is clean at every x_k, and $\delta = max(\{\mathsf{width}(a,x_k) | k \in \{x_i,\ldots,x_j\}\}))$. Define b to be b', with an additional signal ${}^{i,j}\mathsf{check\text{-}ok}$ at position $x_c = (x_i + x_{i_1})/2$. We say that b is a *δ-width checked configuration* for a. The ${}^{i,j}\mathsf{check\text{-}ok}$ signal acts as a witness that the configuration is locally good. This configuration must coincide with b on a wide enough region around x –and thus, also, for a long enough time. The parameter δ must additionally be small enough with respect to t, as will be defined in Sect. 5.

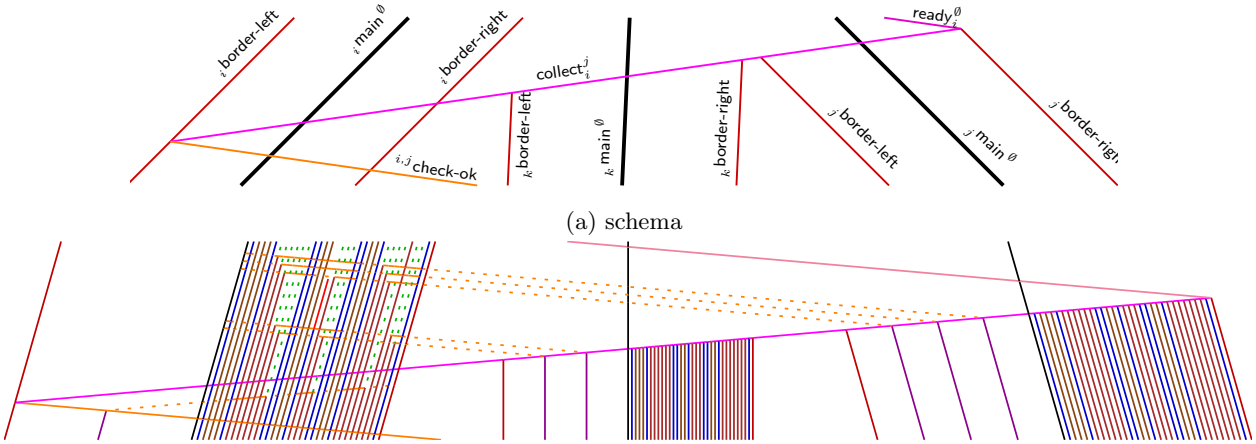
In the rest of this section, we give subsets $M_{\mathcal{U}_\mathcal{S}^{\mathsf{checked}}}$ and $R_{\mathcal{U}_\mathcal{S}^{\mathsf{checked}}}$ of signals and collision rules of $\mathcal{U}_\mathcal{S}$ (depending only on $\mathcal{S}$). The machine $\mathcal{U}_\mathcal{S}^{\mathsf{checked}}$ ensures that, if δ is small enough, there is δ_o with $\delta_o < K.\delta$ for a fixed $0 < K$ and $t + \delta_o < t'$, the configuration $\mathbb{B}(t')$ is such that $\mathsf{interp}^*(\mathbb{B}(t')) = \mathbb{A}(t')$, and for any position x such that $\mathbb{B}(t')(x) \neq \oslash$, $\mathbb{B}(t')$ is clean at x (and has width $\mathsf{width}(x_i)$).

4.1 Useless information disposal and id gathering

We describe the signals and collision rules of $\mathcal{U}_\mathcal{S}^{\mathsf{checked}}$ through its behaviour on δ-width checked configurations. The signals and rules are then listed in full.

The list of rules of the leftmost macro-signal is used to find the corresponding rule to apply. All _id's are sent onto this list to operate the rule selection. The rule lists in other macro-signals are just discarded.

This is done as in Fig. 6. Figure 6a depicts the signals that drive the dynamics (${}^{i,j}\mathsf{check\text{-}ok}$ then $\mathsf{collect}_i^j$ then $\mathsf{ready}_i^\emptyset$) while there is an actual diagram in Fig. 6b.



(a) schema

(b) example

Figure 6: Removing unused lists and sending ids to the decision making area.

Signal ${}^{i,j}\mathsf{check\text{-}ok}$ initiates the process. It first goes on the left to make the id of the leftmost macro-signal act on the rules. It bounces on ${}_i\mathsf{border\text{-}left}$ to become $\mathsf{collect}_i^j$. Signal $\mathsf{collect}_i^j$ crosses the whole configuration and bounces (and erases) the ${}_j\mathsf{border\text{-}right}$ (rightmost) to become $\mathsf{ready}_i^\emptyset$. The latter will select and apply the rule.

Before turning to $\mathsf{collect}_i^j$, signal ${}^{i,j}\mathsf{check\text{-}ok}$ turns each ${}_i\mathsf{id}$ (for i only) into $\mathsf{cross\text{-}back\text{-}ok}_i$ which heads right for the rule list. Together, these $\mathsf{cross\text{-}back\text{-}ok}_i$ signals encode the id of the macro-signal of speed index i. In Fig. 6, this corresponds to the signals on the bottom left that are changed to fast right-bounds signals.

While crossing the configuration, $\mathsf{collect}_i^j$ erases all the surounding signals of collaborating macro-signals. It turns each ${}_k\mathsf{id}$ (for $j \leq k < i$) into $\mathsf{cross\text{-}ok}_k$ which heads left for the rule list. Together, the $\mathsf{cross\text{-}ok}_k$ signals encode the id of the macro-signal of speed index k. In Fig. 6, this corresponds to the remaining signals of each macro-signals, which are changed to fast left-bounds signals.

All the needed meta-signals and collision rules are defined in Fig. 7. The constant 40 is arbitrary. It ensures that the delays in Fig. 3 are respected.

The function interp is refined to take account of these new signals and rules. As before, the value of $\mathsf{interp}(b)$ is $\oslash$ if $b(0)$ is not some ${}_k\mathsf{main}^\emptyset$. It always ignores $\mathsf{collect}_i^j$ and ${}^{i,j}\mathsf{check\text{-}ok}$. If there is a $\mathsf{collect}_i^j$ before the first _border-left to the left, it looks to the left until the first ${}_i\mathsf{main}^\emptyset$ (for the same value of i). If there is a ${}_i\mathsf{border\text{-}right}$ before the ${}_k\mathsf{border\text{-}left}$, then it counts any $\mathsf{cross\text{-}ok}_k$ before that ${}_i\mathsf{border\text{-}right}$ as a ${}_k\mathsf{id}$. Likewise, it also counts any $\mathsf{cross\text{-}back\text{-}ok}_i$ to the right of ${}_i\mathsf{main}^\emptyset$ as a ${}_i\mathsf{id}$.

Parameter	Value
$s^{\max}$	$= \max_{i \in [\![1,\mathfrak{n}]\!]} \lvert s_i \rvert$
s^{rapid}	$= 40.s^{\max}$

	Meta-signal	speed
$\forall i, j \in [\![1,\mathfrak{n}]\!],\ j<i,$	^{i,j}check-ok	$-s^{\text{rapid}}$
$\forall i, j \in [\![1,\mathfrak{n}]\!],\ j<i,$	collect$_i^j$	s^{rapid}
$\forall i \in [\![1,\mathfrak{n}]\!],$	cross-back-ok$_i$	s^{rapid}
$\forall i \in [\![1,\mathfrak{n}]\!],$	cross-ok$_i$	$-s^{\text{rapid}}$

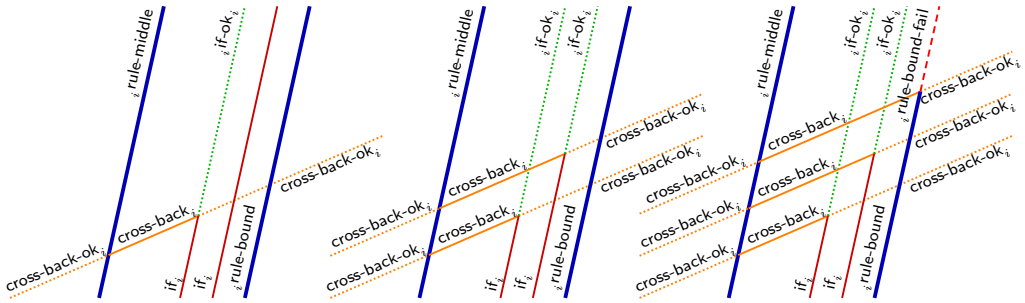
$$
\begin{aligned}
\forall i,j \in [\![1,\mathfrak{n}]\!],\ j<i, &\quad \{ {}_i\text{id},\ {}^{i,j}\text{check-ok} \} \to \{ {}^{i,j}\text{check-ok},\ \text{cross-back-ok}_i \} \\
\forall i,j \in [\![1,\mathfrak{n}]\!],\ j<i, &\quad \{ {}_i\text{border-left},\ {}^{i,j}\text{check-ok} \} \to \{ {}_i\text{border-left},\ \text{collect}_i^j \} \\
\forall i,j,k \in [\![1,\mathfrak{n}]\!],\ j \le k<i, &\quad \{ \text{collect}_i^j,\ {}_k\text{border-left} \} \to \{ \text{collect}_i^j \} \\
\forall i,j,k \in [\![1,\mathfrak{n}]\!],\ j \le k<i, &\quad \{ \text{collect}_i^j,\ {}_k\text{id} \} \to \{ \text{cross-ok}_k,\ \text{collect}_i^j \} \\
\forall i,j,k \in [\![1,\mathfrak{n}]\!],\ j<k<i, &\quad \{ \text{collect}_i^j,\ {}_k\text{border-right} \} \to \{ \text{collect}_i^j \} \\
\forall i,j,k \in [\![1,\mathfrak{n}]\!],\ j \le k<i, &\quad \{ \text{collect}_i^j,\ {}_k\text{rule-bound} \} \to \{ \text{collect}_i^j \} \\
\forall i,j,k \in [\![1,\mathfrak{n}]\!],\ j \le k<i, &\quad \{ \text{collect}_i^j,\ {}_k\text{rule-middle} \} \to \{ \text{collect}_i^j \} \\
\forall i,j,k,l \in [\![1,\mathfrak{n}]\!],\ j \le k<i, &\quad \{ \text{collect}_i^j,\ {}_k\text{if}_l \} \to \{ \text{collect}_i^j \} \\
\forall i,j,k,l \in [\![1,\mathfrak{n}]\!],\ j \le k<i, &\quad \{ \text{collect}_i^j,\ {}_k\text{then}_l \} \to \{ \text{collect}_i^j \} \\
\forall i,j \in [\![1,\mathfrak{n}]\!],\ j<i, &\quad \{ \text{collect}_i^j,\ {}_i\text{border-right} \} \to \{ \text{ready}_i^{\emptyset} \}
\end{aligned}
$$

Figure 7: Meta-signals and collisions rules for disposal.

4.2 Applying id's onto rules

The beam of cross-back-ok$_i$ signals acts on every if-part of the rules and tries to cross-out the same number of if$_i$. Travelling rightward, each meet $_i$rule-middle before the if-part of the rule. It gets *activated* as cross-back$_i$. On meeting if$_i$, they are both *deactivated* and becomes cross-back-ok$_i$ and $_i$if-ok$_i$. This is illustrated on Fig. 8 where dotted lines indicate deactivation and dashed ones indicate failure.

If the numbers do not match, a mark is left on the rule. If the cross-back-ok$_i$ are too few, then at least one (activated) if$_i$ remains as in Fig. 8a. If the cross-back-ok$_i$ are in excess, then at least one (activated) cross-back$_i$ reaches the $_i$rule-bound on the right. It turns it into $_i$rule-bound-fail to indicate failure of the rule as in Fig. 8c. Signal cross-back$_i$ is always deactivated on leaving the rule (on $_i$rule-bound or $_i$rule-bound-fail). Signals cross-back-ok$_i$ are destroyed after the last rule (on $_i$border-right). The left of Fig. 6b displays a real application of cross-back-ok$_i$ on the rules.



(a) too few (b) equal (c) too much

Figure 8: Comparison of id's in the if-part of a rule for cross-back$_i$.

For every other present speed s_k ($j \le k < i$), the beam of cross-ok$_k$ signals acts on every if-part of the rules similarly and tries to cross-out the same number of $_i$if$_k$. The difference is that they enter each rule from the right: cross-ok$_k$ are activated (into cross$_k$) by $_i$rule-bound and mark the excess (of cross$_k$) on $_i$rule-middle (as $_i$rule-middle-fail). Signals cross-ok$_k$/cross$_k$ are destroyed after the last rule (on $_i$main$^{\emptyset}$ after activation by the last closing $_i$rule-bound).

Figure 9 depicts the process with equality on speed number 4, too few on speed 6 and too much on speed 5. Figure 6b displays a real application of id's onto rules.

For every speed index m not involved in the macro-collision, $_i$if$_m$ are unaffected and thus remain active. Altogether, if the if-part of a rule does not match the incoming macro-signals, then at least one $_i$if$_l$ remains or $_i$rule-middle is replaced by $_i$rule-middle-fail or the left $_i$rule-bound is replaced by $_i$rule-bound-fail.

All the meta-signals and collision rules needed for the application are detailed in Fig. 10.

As above, the function interp has to be refined to take these new meta-signal and rules into account. It still yields $\oslash$ if the configuration is not centred on a _main$^{\emptyset}$. If it is centred on $_k$main$^{\emptyset}$, then it needs to recover the identity corresponding to that main signal. This is done by counting any cross-ok$_k$ and cross$_k$ as

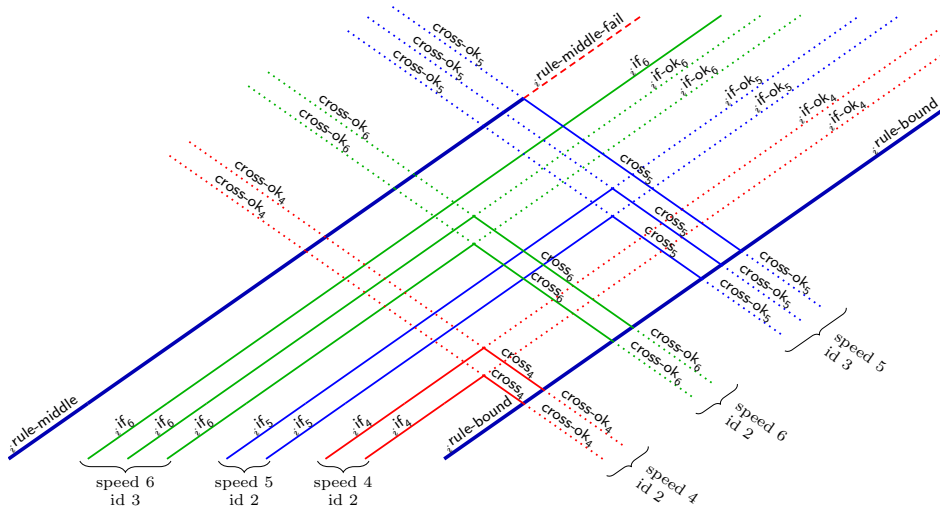


Figure 9: Comparison of id's in the if-part of a rule.

	Meta-signal	speed
$\forall i \in [\![1,\mathfrak{n}]\!]$,	$\mathsf{cross\text{-}back}_i$	s^{rapid}
$\forall i \in [\![1,\mathfrak{n}]\!]$,	cross_i	$-s^{\mathsf{rapid}}$
$\forall i,l \in [\![1,\mathfrak{n}]\!]$,	${}_i\mathsf{if\text{-}ok}_l$	s_i
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}_i\mathsf{rule\text{-}middle\text{-}fail}$	s_i
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}_i\mathsf{rule\text{-}bound\text{-}fail}$	s_i

$$
\begin{array}{rl}
\forall i \in [\![1,\mathfrak{n}]\!], & \{\mathsf{cross\text{-}back\text{-}ok}_i,\ {}_i\mathsf{rule\text{-}middle}\} \to \{\ {}_i\mathsf{rule\text{-}middle},\ \mathsf{cross\text{-}back}_i\} \\
\forall i \in [\![1,\mathfrak{n}]\!], & \{\mathsf{cross\text{-}back}_i,\ \mathsf{if}_i\} \to \{\ {}_i\mathsf{if\text{-}ok}_i,\ \mathsf{cross\text{-}back\text{-}ok}_i\} \\
\forall i \in [\![1,\mathfrak{n}]\!], & \{\mathsf{cross\text{-}back}_i,\ {}_i\mathsf{rule\text{-}bound}\} \to \{\ \mathsf{cross\text{-}back\text{-}ok}_i,\ {}_i\mathsf{rule\text{-}bound\text{-}fail}\} \\
\forall i \in [\![1,\mathfrak{n}]\!], & \{\mathsf{cross\text{-}back}_i,\ {}_i\mathsf{rule\text{-}bound\text{-}fail}\} \to \{\ \mathsf{cross\text{-}back\text{-}ok}_i,\ {}_i\mathsf{rule\text{-}bound\text{-}fail}\} \\
\forall i \in [\![1,\mathfrak{n}]\!], & \{\mathsf{cross\text{-}back\text{-}ok}_i,\ {}_i\mathsf{border\text{-}right}\} \to \{\ {}_i\mathsf{border\text{-}right}\} \\
\forall i \in [\![1,\mathfrak{n}]\!], & \{{}_i\mathsf{rule\text{-}bound},\ \mathsf{cross\text{-}ok}_i\} \to \{\ \mathsf{cross}_i,\ {}_i\mathsf{rule\text{-}bound}\} \\
\forall i,k \in [\![1,\mathfrak{n}]\!],\ k<i, & \{{}_i\mathsf{if}_k,\ \mathsf{cross}_k\} \to \{\ \mathsf{cross\text{-}ok}_k,\ {}_i\mathsf{if\text{-}ok}_k\} \\
\forall i,k \in [\![1,\mathfrak{n}]\!],\ k<i, & \{{}_i\mathsf{rule\text{-}middle},\ \mathsf{cross}_k\} \to \{\ \mathsf{cross\text{-}ok}_k,\ {}_i\mathsf{rule\text{-}middle\text{-}fail}\} \\
\forall i,k \in [\![1,\mathfrak{n}]\!],\ k<i, & \{{}_i\mathsf{rule\text{-}middle\text{-}fail},\ \mathsf{cross}_k\} \to \{\ \mathsf{cross\text{-}ok}_k,\ {}_i\mathsf{rule\text{-}middle\text{-}fail}\} \\
\forall i,k \in [\![1,\mathfrak{n}]\!],\ k<i, & \{{}_i\mathsf{main}^{\emptyset},\ \mathsf{cross}_k\} \to \{\ {}_i\mathsf{main}^{\emptyset}\}
\end{array}
$$

Figure 10: Meta-signals and collisions rules for applying id's to rules.

a ${}_k\mathsf{id}$ (or $\mathsf{cross\text{-}back\text{-}ok}_i$ and $\mathsf{cross\text{-}back}_i$ when k is i). This count is completed by counting the ${}_i\mathsf{if\text{-}ok}_k$ in the section encoding ρ_{max}. This yields the correct value since the ids of the input signals in ρ_{max} are maximal.

4.3 Selecting the rule

After all of the $\mathsf{cross\text{-}back\text{-}ok}_i$ and $\mathsf{cross\text{-}ok}_k$ have operated on the list, since the simulated machine is deterministic at most one of the rules has no ${}_i\mathsf{if}_l$ left and no ${}_i\mathsf{rule\text{-}bound\text{-}fail}$ nor ${}_i\mathsf{rule\text{-}middle\text{-}fail}$. This rule is the one corresponding to the collision that is being simulated. (If there is no rule, then the output is empty: macro-signals just annihilate together.)

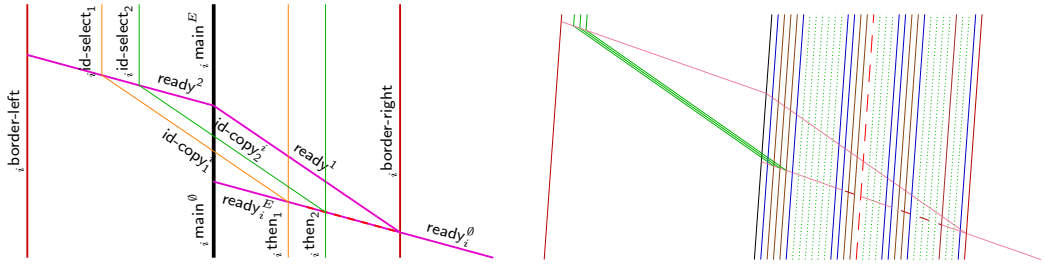
When the rule is found two things have to be done: (a) extracting a copy of the then part of the rule, and (b) recording the output speeds.

This is carried out by the $\mathsf{ready}_i^{\emptyset}$ coming from the right. When it meets some ${}_i\mathsf{if}_l$ or ${}_i\mathsf{rule\text{-}bound\text{-}fail}$ or ${}_i\mathsf{rule\text{-}middle\text{-}fail}$, it becomes $\mathsf{ready\text{-}no}_i^{\emptyset}$. It is reactivated (i.e. turned back to $\mathsf{ready}_i^{\emptyset}$) on meeting ${}_i\mathsf{rule\text{-}bound}$. It is still active only after crossing the correct if-part.

Activated in the correct then-part, $\mathsf{ready}_i^{\emptyset}$ makes a slower copy to be sent on the left and stores the index of the ${}_i\mathsf{then}_l$. The output indices are collected in a set in the exponent part of $\mathsf{ready}_i^{\emptyset}$, becoming ready_i^{E} where E is the subset of $[\![1,\mathfrak{n}]\!]$ collecting the indices of all out-speed. This subset is updated each time a ${}_i\mathsf{then}_l$ is met when active. It is preserved by activation/deactivation and transmitted to ${}_i\mathsf{main}^{\emptyset}$ (that becomes ${}_i\mathsf{main}^{E}$).

Extracting a copy of the then-part is done as it is shown in Figure 11. The copy goes to the left to cross ${}_i\mathsf{main}^{E}$. They are then made parallel to ${}_i\mathsf{main}^{E}$ by a faster signal ready^2. To generate it $\mathsf{ready}_i^{\emptyset}$ emits ready^1 on meeting ${}_i\mathsf{border\text{-}right}$. On meeting ${}_i\mathsf{main}^{E}$, ready^1 is changed to ready^2 that sets on position all _id's and

disappears on meeting ${}_i\mathsf{border\text{-}left}$.



(a) scheme (b) example

Figure 11: Rule selection.

All the needed meta-signals and collision rules are defined in Fig. 12.

	Meta-signal	speed
$\forall i{\in}[\![1,\mathfrak{n}]\!],\ \forall E{\subseteq}[\![1,\mathfrak{n}]\!],$	ready_i^E	$-s^{\mathsf{rapid}}$
$\forall i{\in}[\![1,\mathfrak{n}]\!],\ \forall E{\subseteq}[\![1,\mathfrak{n}]\!],$	$\mathsf{ready\text{-}no}_i^E$	$-s^{\mathsf{rapid}}$
	ready^1	$-s^{\mathsf{rapid}}/2$
	ready^2	$-s^{\mathsf{rapid}}$
$\forall i, l{\in}[\![1,\mathfrak{n}]\!],$	$\mathsf{id\text{-}copy}_l^i$	$-s^{\mathsf{rapid}}$
$\forall i, l{\in}[\![1,\mathfrak{n}]\!],$	${}_i\mathsf{id\text{-}select}_l$	s_i

$$\begin{array}{rl}
\forall i{\in}[\![1,\mathfrak{n}]\!],\ \forall E{\subseteq}[\![1,\mathfrak{n}]\!], & \{\,{}_i\mathsf{rule\text{-}bound\text{-}fail},\ \mathsf{ready}_i^E\,\} \to \{\,\mathsf{ready\text{-}no}_i^E,\ {}_i\mathsf{rule\text{-}bound\text{-}fail}\,\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!],\ \forall E{\subseteq}[\![1,\mathfrak{n}]\!], & \{\,{}_i\mathsf{if}_l,\ \mathsf{ready}_i^E\,\} \to \{\,\mathsf{ready\text{-}no}_i^E,\ {}_i\mathsf{if}_l\,\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!],\ \forall E{\subseteq}[\![1,\mathfrak{n}]\!], & \{\,{}_i\mathsf{rule\text{-}middle\text{-}fail},\ \mathsf{ready}_i^E\,\} \to \{\,\mathsf{ready\text{-}no}_i^E,\ {}_i\mathsf{rule\text{-}middle\text{-}fail}\,\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!],\ \forall E{\subseteq}[\![1,\mathfrak{n}]\!], & \{\,{}_i\mathsf{rule\text{-}bound},\ \mathsf{ready\text{-}no}_i^E\,\} \to \{\,\mathsf{ready}_i^E,\ {}_i\mathsf{rule\text{-}bound}\,\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}_i\mathsf{border\text{-}right},\ \mathsf{ready}_i^\emptyset\,\} \to \{\,\mathsf{ready}_i^\emptyset,\ \mathsf{ready}^1,\ {}_i\mathsf{border\text{-}right}\,\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!],\ \forall E{\subseteq}[\![1,\mathfrak{n}]\!], & \{\,{}_i\mathsf{main}^\emptyset,\ \mathsf{ready}_i^E\,\} \to \{\,{}_i\mathsf{main}^E\,\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!],\ \forall E{\subseteq}[\![1,\mathfrak{n}]\!], & \{\,{}_i\mathsf{main}^E,\ \mathsf{ready}^1\,\} \to \{\,\mathsf{ready}^2,\ {}_i\mathsf{main}^E\,\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}_i\mathsf{border\text{-}left},\ \mathsf{ready}^2\,\} \to \{\,{}_i\mathsf{border\text{-}left}\,\}\\
\forall i, l{\in}[\![1,\mathfrak{n}]\!],\ \forall E{\subseteq}[\![1,\mathfrak{n}]\!], & \{\,{}_i\mathsf{then}_l,\ \mathsf{ready}_i^E\,\} \to \{\,\mathsf{ready}_i^{E\cup\{l\}},\ \mathsf{id\text{-}copy}_l^i,\ {}_i\mathsf{then}_l\,\}\\
\forall i, l{\in}[\![1,\mathfrak{n}]\!],\ \forall E{\subseteq}[\![1,\mathfrak{n}]\!], & \{\mathsf{id\text{-}copy}_l^i,\ \mathsf{ready}^2\,\} \to \{\,\mathsf{ready}^2,\ {}_i\mathsf{id\text{-}select}_l\,\}
\end{array}$$

Figure 12: Meta-signals and collisions rules for selecting the collision rule.

As above, the function interp has to be refined to take these new meta-signal and rules into account. It suffices to have interp ignore the new signals from this section, since they leave the signals used previously by interp unaffected.

4.4 Setting the output macro-signals

Figure 13 depicts how the output macro-signals are generated. When ${}_i\mathsf{main}^E$ and ${}_j\mathsf{main}^\emptyset$ (and all collaborating main signals) meet, then $\mathsf{fast\text{-}left}^E$ and $\mathsf{fast\text{-}right}^E$ are sent and ${}_l\mathsf{main}^\emptyset$ are generated for every l of E.

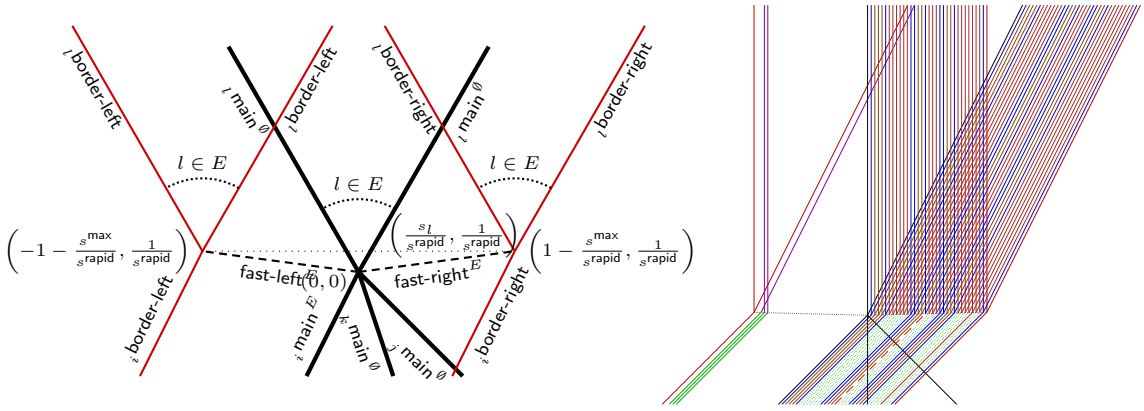
On the left, $\mathsf{fast\text{-}left}^E$ sends each ${}_i\mathsf{id\text{-}select}_l$ signal on the right direction as ${}_l\mathsf{id}$. Then on reaching ${}_i\mathsf{border\text{-}left}$, it emits one ${}_l\mathsf{border\text{-}left}$ for each l of E and disappears. Similarly, on the right, $\mathsf{fast\text{-}right}^E$ sends a clean copy of the rules on each speed l of E. Finally, on reaching ${}_i\mathsf{border\text{-}right}$, it emits one ${}_l\mathsf{border\text{-}right}$ for each l of E and disappears.

If the simulated intersection happens at $(0,0)$ with ${}_i\mathsf{border\text{-}left}$ at $(-1-\frac{s_i+s^{\mathsf{max}}}{s^{\mathsf{rapid}}},0)$ and ${}_i\mathsf{border\text{-}right}$ at $(1-\frac{s_i+s^{\mathsf{max}}}{s^{\mathsf{rapid}}},0)$, $\mathsf{fast\text{-}left}^E$ and ${}_i\mathsf{border\text{-}left}$ intersect at $(-1-\frac{s^{\mathsf{max}}}{s^{\mathsf{rapid}}},\frac{1}{s^{\mathsf{rapid}}})$, while $\mathsf{fast\text{-}right}^E$ and ${}_i\mathsf{border\text{-}right}$ intersect at $(1-\frac{s^{\mathsf{max}}}{s^{\mathsf{rapid}}},\frac{1}{s^{\mathsf{rapid}}})$. At time $\frac{1}{s^{\mathsf{rapid}}}$, each outgoing ${}_l\mathsf{main}^\emptyset$ will be at position $\frac{s_l}{s^{\mathsf{rapid}}}$.

This proves that main signals will remain about in the middle of their borders (specifically, at position $\frac{s_i+s^{\mathsf{max}}}{s^{\mathsf{rapid}}}$ if the borders are at -1 and $+1$), and that the right part of macro-signals remains no bigger than the left part. It ensures all $\mathsf{id\text{-}copy}_l^i$ cross ready^2 before ${}_i\mathsf{border\text{-}left}$.

After a while (given explicitly in Sect. 5.2), all the initiated macro-signals are separated and ready for macro-collision.

Knowing how the output of a macro-collision is set, we are ready to provide the value of repr on a collision



(a) scheme (b) example

Figure 13: Generating the output of a macro-collision.

ρ, as advertised in Sect. 3.2:

$$_i\mathsf{border\text{-}left} \quad {}_i\langle\text{output id encoding}\rangle \quad \rho^{F,E} \quad {}_i\langle\text{rules encoding}\rangle \quad {}_i\mathsf{border\text{-}right} \ .$$

The collision $\rho^{F,E}$ is:

$$\{{}_l\mathsf{main}^\emptyset\}_{l\in F} \cup \{{}_i\mathsf{main}^E\} \to \{{}_l\mathsf{main}^\emptyset\}_{l\in E} \cup \{\mathsf{fast\text{-}left}^E, \mathsf{fast\text{-}right}^E\}$$

With F the set of indices of speeds different from s_l of input meta-signals (of ρ^-). The output ids (of ρ^+) are encoded in unary with ${}_i\mathsf{id\text{-}select}_l$ signals. The rules are tainted with failures marks and ${}_i\mathsf{if\text{-}ok}_k$ signals, as they would be, had the ids of ρ^- been applied to them, so that interp can recognise ρ. The signal ${}_i\mathsf{border\text{-}left}$, ${}_i\mathsf{border\text{-}right}$ and $\rho^{F,E}$ are placed respectively at positions $-1 - \frac{s_i+s^{\max}}{s^{\text{rapid}}}$, $1 - \frac{s_i+s^{\max}}{s^{\text{rapid}}}$, and 0.

All the needed meta-signals and collision rules are defined in Fig. 14.

	Meta-signal	**speed**
$\forall E \subseteq [\![1, \mathfrak{n}]\!]$,	$\mathsf{fast\text{-}left}^E$	$-s^{\text{rapid}} - s^{\max}$
$\forall E \subseteq [\![1, \mathfrak{n}]\!]$,	$\mathsf{fast\text{-}right}^E$	$s^{\text{rapid}} - s^{\max}$

$\forall i \in [\![1, \mathfrak{n}]\!], \forall E \subseteq [\![1, \mathfrak{n}]\!], F \subseteq [\![1, i-1]\!], \lvert F\rvert \neq 0,$	$\{{}_l\mathsf{main}^\emptyset\}_{l\in F} \cup \{{}_i\mathsf{main}^E\} \to \{{}_l\mathsf{main}^\emptyset\}_{l\in E} \cup \{\mathsf{fast\text{-}left}^E, \mathsf{fast\text{-}right}^E\}$
$\forall i \in [\![1, \mathfrak{n}]\!], \forall E \subseteq [\![1, \mathfrak{n}]\!],$	$\{{}_i\mathsf{border\text{-}left}, \mathsf{fast\text{-}left}^E\} \to \{{}_l\mathsf{border\text{-}left}_{l\in E}\}$
$\forall i, l \in [\![1, \mathfrak{n}]\!], \forall E \subseteq [\![1, \mathfrak{n}]\!], l \in E,$	$\{{}_i\mathsf{id\text{-}select}_l, \mathsf{fast\text{-}left}^E\} \to \{\mathsf{fast\text{-}left}^E, {}_i\mathsf{id}\}$
$\forall i \in [\![1, \mathfrak{n}]\!], \forall E \subseteq [\![1, \mathfrak{n}]\!],$	$\{\mathsf{fast\text{-}right}^E, {}_i\mathsf{rule\text{-}bound}\} \to \{{}_l\mathsf{rule\text{-}bound}\}_{l\in E} \cup \{\mathsf{fast\text{-}right}^E\}$
$\forall i \in [\![1, \mathfrak{n}]\!], \forall E \subseteq [\![1, \mathfrak{n}]\!],$	$\{\mathsf{fast\text{-}right}^E, {}_i\mathsf{rule\text{-}bound\text{-}fail}\} \to \{{}_l\mathsf{rule\text{-}bound}\}_{l\in E} \cup \{\mathsf{fast\text{-}right}^E\}$
$\forall i \in [\![1, \mathfrak{n}]\!], \forall E \subseteq [\![1, \mathfrak{n}]\!],$	$\{\mathsf{fast\text{-}right}^E, {}_i\mathsf{rule\text{-}middle}\} \to \{{}_l\mathsf{rule\text{-}middle}\}_{l\in E} \cup \{\mathsf{fast\text{-}right}^E\}$
$\forall i \in [\![1, \mathfrak{n}]\!], \forall E \subseteq [\![1, \mathfrak{n}]\!],$	$\{\mathsf{fast\text{-}right}^E, {}_i\mathsf{rule\text{-}middle\text{-}fail}\} \to \{{}_l\mathsf{rule\text{-}middle}\}_{l\in E} \cup \{\mathsf{fast\text{-}right}^E\}$
$\forall i, m \in [\![1, \mathfrak{n}]\!], \forall E \subseteq [\![1, \mathfrak{n}]\!],$	$\{\mathsf{fast\text{-}right}^E, {}_i\mathsf{if}_m\} \to \{{}_l\mathsf{if}_m\}_{l\in E} \cup \{\mathsf{fast\text{-}right}^E\}$
$\forall i, m \in [\![1, \mathfrak{n}]\!], \forall E \subseteq [\![1, \mathfrak{n}]\!],$	$\{\mathsf{fast\text{-}right}^E, {}_i\mathsf{if\text{-}ok}_m\} \to \{{}_l\mathsf{if}_m\}_{l\in E} \cup \{\mathsf{fast\text{-}right}^E\}$
$\forall i, m \in [\![1, \mathfrak{n}]\!], \forall E \subseteq [\![1, \mathfrak{n}]\!],$	$\{\mathsf{fast\text{-}right}^E, {}_l\mathsf{then}_m\} \to \{{}_l\mathsf{then}_m\}_{l\in E} \cup \{\mathsf{fast\text{-}right}^E\}$
$\forall i \in [\![1, \mathfrak{n}]\!], \forall E \subseteq [\![1, \mathfrak{n}]\!],$	$\{\mathsf{fast\text{-}right}^E, {}_i\mathsf{border\text{-}right}\} \to \{{}_l\mathsf{border\text{-}right}_{l\in E}\}$

Figure 14: Meta-signals and collisions rules for the output.

An exact 3-signal collision simulation is depicted in Figure 15.

Let $\mathcal{U}_{\mathcal{S}}^{\mathsf{checked}}$ be the signal machine defined by the above signals and collision rules, instantiated for all possible values of i and j in $\mathcal{S}$. The above arguments constitute the proof of the following lemma.

Lemma 7. *Let ρ be a collision rule of $\mathcal{A}$, and let a be a configuration of $\mathcal{A}$ whose signals are exactly ρ^- and the positions of these signals are such that they all meet at some point (x, t). Then for small enough δ, let b be a δ-width checked configuration for a. Let $\mathbb{A}$ and $\mathbb{B}$ be the respective space-time diagrams of $(a, \mathcal{A})$ and $(b, \mathcal{U}_{\mathcal{S}}^{\mathsf{checked}})$. There is δ_o (depending on δ) such that for $t' > t + \delta_o$, the configuration $\mathbb{B}(t')$ satisfies $\mathsf{interp}(\mathbb{B}(t')) = \mathbb{A}(t')$, and for any position x such that $\mathbb{B}(t')(x) \neq \oslash$, $\mathbb{B}(t')$ is clean at x.*

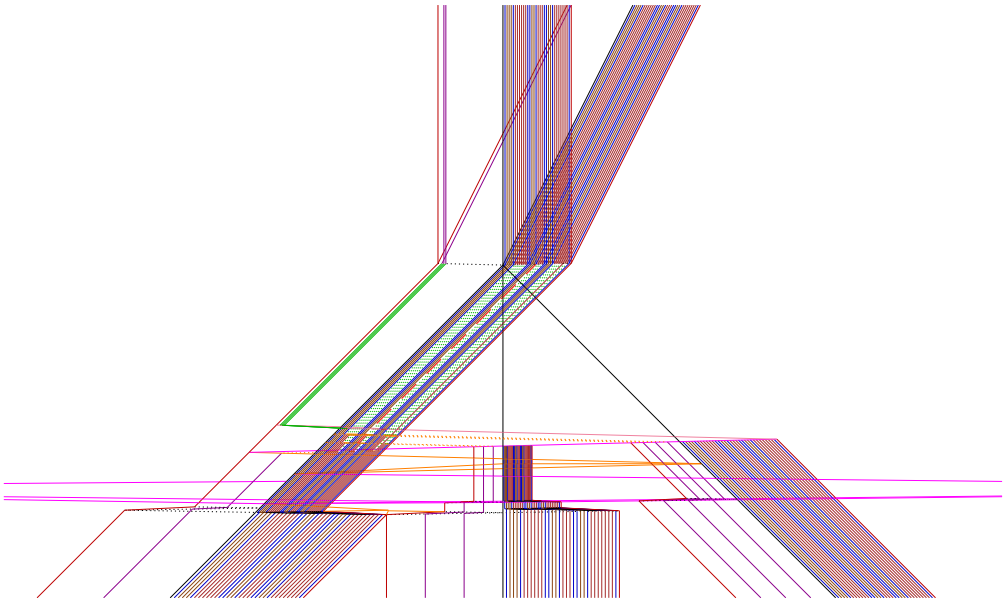


Figure 15: Exact 3-signal collision with whole preparation.

We now refine the interp function in accordance with this section. If there is no fast-left^E or fast-right^E signal, then no change is needed. Otherwise, if the centre of the configuration is a collision between _main$^\emptyset$ signals, both the input and the outputs of that collision have to be determined. This can be done by looking at the signals between $_i$main$^\emptyset$ and $_i$border-right, where one of the rules has been selected. After the collision, the identity of an outgoing $_l$main$^\emptyset$ signal can be gathered from the $_l$id right of fast-left^E, and the $_i$id-select$_l$ left of fast-left^E.

With this refinement of interp*, the above construction gives a "conditional simulation" for configurations with one exact collision, as stated in the following lemma.

Lemma 8. *Let ρ be a collision rule of $\mathcal{A}$, and let a be a configuration of $\mathcal{A}$ whose signals are exactly ρ^- and the positions of these signals are such that they all meet at some point (x,t). Then for small enough δ, let b be a δ-width checked configuration for a. Let $\mathbb{A}$ and $\mathbb{B}$ be the respective space-time diagrams of a and b. We have that for any $t' \geq 0$,*

$$\mathsf{interp}^*(\mathbb{B}(t')) = \mathbb{A}(t') \ .$$

Let ρ be a collision rule of $\mathcal{A}$, and let a be a configuration of $\mathcal{A}$ whose signals are exactly ρ^- and the positions of these signals are such that they all meet at the point $(0,1)$. Then for small enough δ, let b be a δ-width checked configuration for a, and $\mathbb{B}$ the associated space-time diagram. Define $\mathsf{repr}(\rho)$ to be the configuration of $\mathbb{B}$ at time 1, rescaled and translated so that $_i$border-left is at $-1 - \frac{s_i + s^{\mathsf{max}}}{s^{\mathsf{rapid}}}$, $_i$border-right is at $1 - \frac{s_i + s^{\mathsf{max}}}{s^{\mathsf{rapid}}}$, and thus the collision of the _main$^\emptyset$ is at 0.

4.5 Towards simulating a collision in a larger diagram

More generally, this construction works for the simulation of a collision when all the participating signals are identified and no other disturbing macro-signal is near.

Lemma 9. *Let $\mathcal{U_S}' = (M_{\mathcal{U_S}'}, S_{\mathcal{U_S}'}, R_{\mathcal{U_S}'})$ be a signal machine which contains the meta-signals and rules of $\mathcal{U}_{\mathcal{S}}^{\mathsf{checked}}$, where for every rule $\rho_{\mathcal{U_S}'} \in R_{\mathcal{U_S}'}$ with an input signal of speed larger than s^{max}, any input signal belonging to $\mathcal{U}_{\mathcal{S}}^{\mathsf{checked}}$ is also in its output.*

Let ρ be a collision rule of $\mathcal{A}$, and let a be a configuration of $\mathcal{A}$ whose signals are exactly ρ^- and the positions of these signals are such that they all meet at some point (x,t). Let $\mathbb{A}$ be the associated space-time diagram.

Then for small enough δ, let b be a δ-width checked configuration for a. There are W, δ_o and W' such that for any initial configuration $b_{\mathcal{U}_S'}$ of $\mathcal{U}_S'$ which coincides with b on a width W around x, after a time $t+\delta_o$, $b_{\mathcal{U}_S'}$ coincides on a width W' with a configuration b_{after} such that:

$$\mathsf{interp}^*(b_{\text{after}}) = \mathbb{A}(t+\delta_o)$$

b_{after} is clean at every position of a signal in $\mathbb{A}(t+\delta_o)$, and any such position is at distance less than W' from x.

Proof. This follows from Lem. 7. For any value of δ, note $W = s^{\mathsf{max}}.(t+\delta_o) + 2\delta$. Suppose b coincides with a δ-width checked configuration $b_{(x,t)}$, on a width W around x at time 0, then at time $t+\delta_o$, it coincides with b on a width δ around x at time $t+\delta_o$. □

Section 5 will deal with ensuring a locally δ-width checked configuration before each macro-collision, with δ small enough with respect to the delay before the collision. This means the following must be ensured:

1. the width of each macro-signal is small enough with respect to the time remaining before the support zones meet,
2. any macro-signal that is not part of the collision is sufficiently away to not interfere,
3. all $\_\mathsf{main}^{\emptyset}$ signals intersect at the same location, where the simulated collision takes place,
4. a signal ${}^{i,j}\mathsf{check\text{-}ok}$ is arriving on the leftmost macro-signal (index i) and j is the speed index of the rightmost one. It witnesses that the resolution presented in this section is ready to start.

5 Preparing for macro-collision

We now define the rest of the signals and collisions of $\mathcal{U}_S$. Again, we explain first how correct diagrams work, then list explicitly the meta-signals and collision rules.

What is needed from $\mathcal{U}_S$ is to make sure that before every collision of $\mathcal{A}$, there is a δ-width checked configuration for the inputs of that configuration, for small enough δ. This is done through the following phases:

detection of a potential macro-collision when some ${}_i\mathsf{border\text{-}right}$ and ${}_k\mathsf{border\text{-}left}$ meet,

shrinking (Sect. 5.1). This is done by an elementary shrinking gadget, and checks appropriate sizing of width of macro-signals of both sides,

testing around (Sect. 5.2). This which ensures a *safety zone* around the macro-collision. This is done by checking whether any unexpected disturbing signal enters the zone, and

check participating macro-signals (Sect. 5.3). Through checking actual position of signals around, the list of speeds of actual participating signals in the ongoing macro-collision is acquired.

The bottom half of Figure 15 shows the full testing and information gathering before resolving a macro-collision. The last two phases may fail. In such a case, the whole process is cancelled to be eventually restarted later. In the rest of this section, only the positive cases are presented. Failure cases are not fully detailed although diagrams and examples are provided.

Detection phase The detection phase is only one collision: any collision between a ${}_i\mathsf{border\text{-}right}$ and ${}_k\mathsf{border\text{-}left}$ sends a collection of signals: ${}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}}^{\mathsf{both}}$, ${}^i\mathsf{shrink\text{-}top}_{\mathsf{R}}^{\mathsf{test}}$, ${}^{i,k}\mathsf{shrink\text{-}test}$, ${}^k\mathsf{shrink\text{-}top}_{\mathsf{L}}^{\mathsf{test}}$, and ${}^k\mathsf{shrink\text{-}bottom}_{\mathsf{L}}^{\mathsf{both}}$, which initiate the shrinking phase.

Shrinking and width checking Shrinking is done by an elementary and widely used gadget in signal machines. To ensure every controlling signal we will send for future uses passes through all the participating meta-signals and comes back in a reasonable time, we ensure that the width of the left-most participating macro-signal is the larger one, among all participating macro-signals. Comparing width of two meta-signals is done by sending a signal from middle and waiting for the echoes. The echo that arrives first indicates the thinner macro-signal.

Testing for safety zone A zone around any (potential) macro-collision is considered to not contain any disrupting signal. This property helps to ensure that no other signal may have collision with the detected meta-signals during the process of handling an ongoing macro-collision. The zone is surrounded by four borders. The existence of disturbing signals in the safety-zone is checked by sending some signals at two bottom borders and testing for any unexpected collision. Any unexpected collision cancels the process. The two other borders have a high absolute slope, thus, no other signal may cross those.

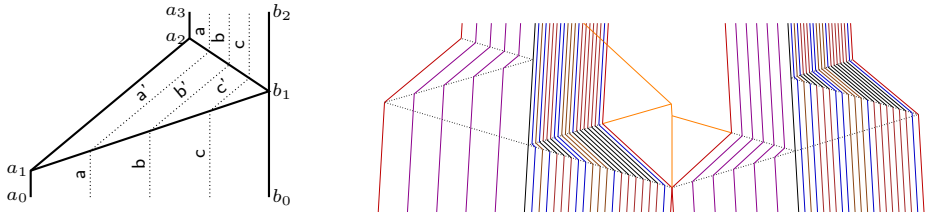
Checking participating meta-signals Once a potential macro-collision is detected, since all speeds of macro-signals are of a fixed set ($\mathcal{S}$), for the ongoing macro-collision, all other potentially present macro-signals could be detected. In order to check their existence, some welcoming signals (${}^{i,k}\mathsf{check\text{-}intersect}^{m}$) are sent to the positions we expect for other meta-signals's ${}_{l}\mathsf{main}^{\emptyset}$ to be present at. Any encounter of ${}_{l}\mathsf{main}^{\emptyset}$ without the corresponding welcoming signal violates the condition of safety-zone. In this case, the process is aborted. Note that, the macro-signals are already shrunk, and next collision would be processed with thinner macro-signals and, thus with a smaller safety-zone.

5.1 Shrinking for delay and separating macro-signals

In order to gain some delay macro-signals are shrunk so that either the macro-collision as presented above is processed or aborted. Aborting is just not to go to the main step; the shrunk macro-signals are operational and can restart a macro-collision.

Figure 16b provides an example of the shrinking process. It is started in the middle where the support zones meet. Shrinking processes are always initiated from the frontier and each half of a macro-signal is shrunk independently; this is done in order to handle concurrent shrinking on the same macro-signal.

Shrinking parallel signals is an application of proportion as illustrated in Fig. 16a. The thick signals control the shrinking while the dotted ones undergo it. Basic geometry shows that the relative position of the intersection of the dotted signals on the segments $[a_0, b_0]$ then $[a_1, b_1]$ then $[a_2, b_1]$ then $[a_3, b_2]$ are identical. Thus they are shrunk with the same relative positions and order.



(a) elementary shrinking (b) shrink and check relative widths

Figure 16: Shrinking.

The elementary shrinking process is quite a usual primitive of signal machines Durand-Lose [2006, 2009], it is not developed more in this article. Figure 17 details the signal scheme to handle the multiple shrinkings.

Process of handling macro-collision requires sending some controlling signals through all the macro-signals and gets back in a reasonable time. By ensuring that the width of the left-most macro-signal is the largest among all participating macro-signals, the width of all potential participating macro-signals could be

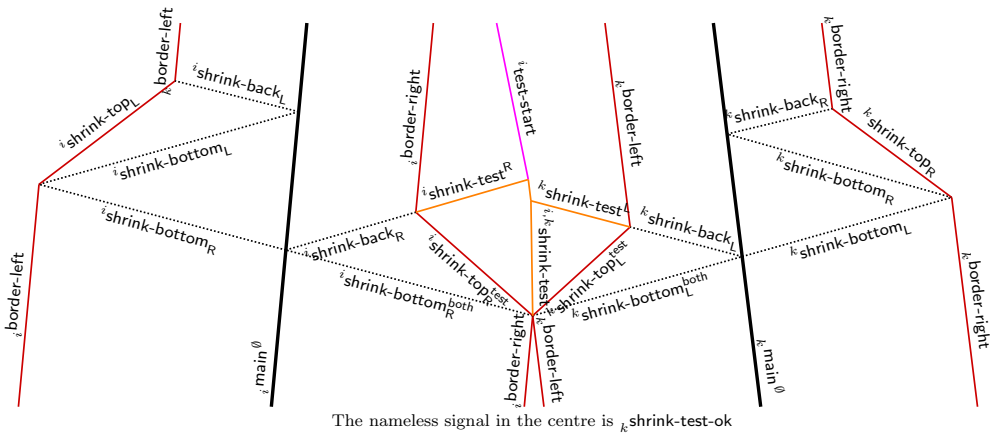


Figure 17: Scheme to shrink and check relative widths.

limited. A gadget in the middle of Figs. 16b and 17 ensures that left macro-signal is wider than the right one. This is done by sending some ^{i,k}shrink-test signal when borders meet. This signal has speed average of $_i$main$^\emptyset$ and $_k$main$^\emptyset$. The ^{i}shrink-bottom$^{\text{both}}_{\text{R}}$ and ^{k}shrink-bottom$^{\text{both}}_{\text{L}}$ bounce on $_i$main$^\emptyset$ and $_k$main$^\emptyset$. If the two macro-collision would have the same width, then the *echos* ^{i}shrink-test$^{\text{R}}$ and ^{k}shrink-test$^{\text{L}}$ would arrive simultaneously. If the left macro is larger, then ^{k}shrink-test$^{\text{L}}$ arrives first.

Otherwise, if ^{i}shrink-test$^{\text{R}}$ arrives first, the right macro-signal is shrunk again and the process is cancelled, to be restarted later when the support zones meet anew, this time with a now-thinner right macro-signal. Figure 18 depicts the case where the right macro-signal is larger than the left one.

In case a new shrink is started when one is already going on, a new signal ^{k}shrink-delayed$_{\text{L}}$ (resp. ^{k}shrink-delayed$_{\text{R}}$) is sent from the collision with ^{k}shrink-top$_{\text{L}}$ (resp. ^{k}shrink-top$_{\text{R}}$) so that it will collide with $_k$border-left (resp. $_k$border-right) to do the shrink after. This is not detailed in the paper, meta-signals and collision rules and the special cases later on are omitted.

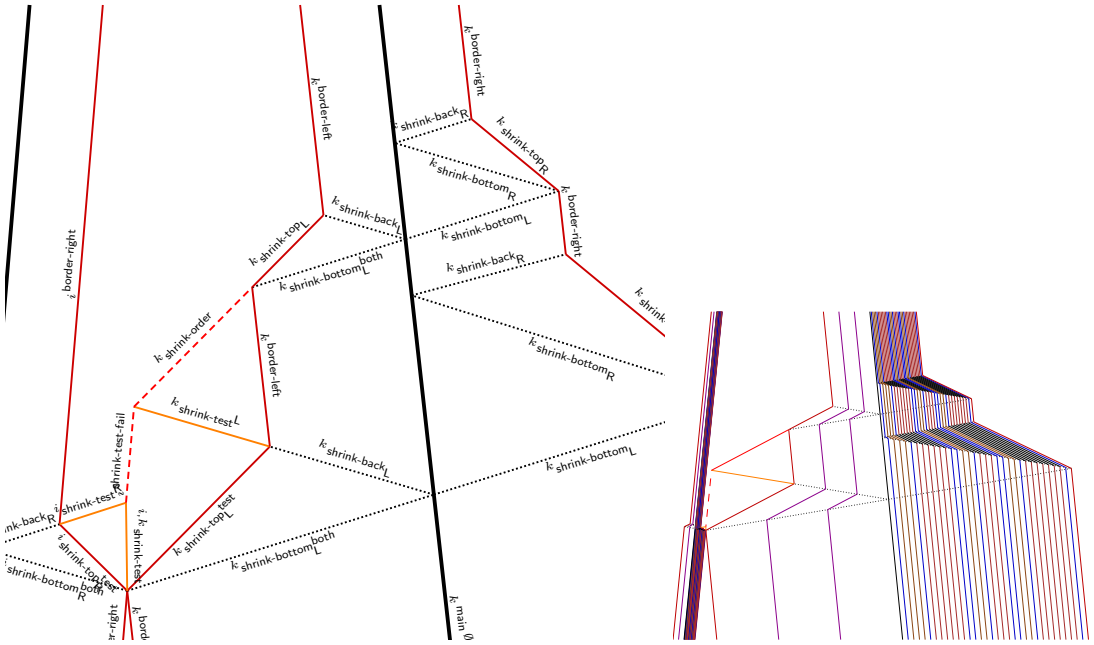
The used meta-signals and collision rules are defined in Figs. 19 and 20.

The interp function has to be refined to take the content of this section in account. For this, it is enough to count any ^{k}shrink-id as if it were a $_k$id, and ignore the other meta-signals of this section. Note that the number of additional signals accounted for by this section for one collision is bounded, therefore interp is indeed defined locally.

5.2 Testing isolation on both sides

It must be ensured that the macro-collision will happen far away enough from any other macro-collision or macro-signals. The outside signals to consider are of two kinds: probe signals (the ones used here and in the next sub-section) and the one that delimits macro-signals and macro-collisions. Probe signals are only testing for the presence of other signals; they are not interacting with any other signals nor collisions, so there is no use to bother with them. The delimiting ones have their speed in $[-s^{\max}, s^{\max}]$ ($s^{\max}$ is the maximum absolute value of any speed in $\mathcal{S}$), so that it is enough to consider only extreme speed on both side.

Figure 21 shows the extent of the safety zone: all the preparation and the resolution is restrained inside it. This large area can be guaranteed from the positions of $_i$main$^\emptyset$ and $_k$main$^\emptyset$ (right next to $_i$main$^\emptyset$), provided the macro-signals have not met yet, so that their width is bounded by the distances between _main$^\emptyset$. The extreme points on top of the collisions are when output macro-signals are separating one from the other as the point C_m for macro-signals of speed s_m and s_{m+1} in Fig. 21. To ensure a large enough safety zone, it will be delimited by four points: Z_T, Z_L, Z_R and Z_B such that:



(a) scheme (b) example

Figure 18: Shrink and check relative width of two neighbouring support zones.

Parameter	Value
s^{shrink}	$= \frac{s^{\mathsf{rapid}}}{2}$

	Meta-signal	speed
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}^{i}\mathsf{shrink\text{-}bottom_L}$	s_i+3s^{shrink}
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}^{i}\mathsf{shrink\text{-}bottom^{both}_L}$	s_i+3s^{shrink}
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}^{i}\mathsf{shrink\text{-}bottom_R}$	s_i-3s^{shrink}
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}^{i}\mathsf{shrink\text{-}bottom^{both}_R}$	s_i-3s^{shrink}
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}^{i}\mathsf{shrink\text{-}top_L}$	s_i+s^{shrink}
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}^{i}\mathsf{shrink\text{-}top^{test}_L}$	s_i+s^{shrink}
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}^{i}\mathsf{shrink\text{-}top_R}$	s_i-s^{shrink}
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}^{i}\mathsf{shrink\text{-}top^{test}_R}$	s_i-s^{shrink}
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}^{i}\mathsf{shrink\text{-}back_L}$	s_i-3s^{shrink}
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}^{i}\mathsf{shrink\text{-}back_R}$	s_i+3s^{shrink}

	Meta-signal	speed
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}^{i}\mathsf{shrink\text{-}id}$	s_i+s^{shrink}
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}^{i}\mathsf{shrink\text{-}rule\text{-}bound}$	s_i-s^{shrink}
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}^{i}\mathsf{shrink\text{-}rule\text{-}middle}$	s_i-s^{shrink}
$\forall i, l \in [\![1,\mathfrak{n}]\!]$,	${}^{i}\mathsf{shrink\text{-}if}^{l}$	s_i-s^{shrink}
$\forall i, l \in [\![1,\mathfrak{n}]\!]$,	${}^{i}\mathsf{shrink\text{-}then}^{l}$	s_i-s^{shrink}
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}^{i}\mathsf{shrink\text{-}test^{L}}$	s_i-3s^{shrink}
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}^{i}\mathsf{shrink\text{-}test^{R}}$	s_i+3s^{shrink}
$\forall i, k \in [\![1,\mathfrak{n}]\!]$, $k<i$,	${}^{i,k}\mathsf{shrink\text{-}test}$	$\frac{s_i+s_k}{2}$
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}_{i}\mathsf{shrink\text{-}test\text{-}ok}$	s_i
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}_{i}\mathsf{shrink\text{-}test\text{-}fail}$	s_i
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}^{i}\mathsf{shrink\text{-}order}$	s_i+s^{shrink}
$\forall i \in [\![1,\mathfrak{n}]\!]$,	${}^{i}\mathsf{test\text{-}start}$	s_i-s^{shrink}

Figure 19: Meta-signals for shrinking.

1. all C_m $(1 \leq m < \mathfrak{n})$ are in the zone,
2. the slope of segment from Z_L to Z_T correspond to the speed s^{max},
3. the slope of segment from Z_R to Z_T correspond to the speed $-s^{\mathsf{max}}$,
4. Z_L and Z_R are low/early enough so that the whole macro-collision is wholly inside the area, and
5. Z_L and Z_R can be reached by signals from Z_B.

First, the position of Z_B is the one where ${}_{i}\mathsf{main}^{\emptyset}$ and ${}^{i}\mathsf{test\text{-}start}$ meet. The position of Z_T is computed from this position and the position of the simulated collision, that is the intersection of ${}_{i}\mathsf{main}^{\emptyset}$ and ${}_{k}\mathsf{main}^{\emptyset}$. To compute the speed to get to the right positions, a coordinate system is introduced where these points have coordinates $(-s_i, -1)$ and $(0, 0)$ respectively. We take $4s^{\mathsf{max}}$ as the width of the left macro-signal as it is an upper bound of it and induces upper bounds on the C_m.

$$
\begin{array}{rr}
\forall i,k{\in}[\![1,\mathfrak{n}]\!],\ k{<}i, & \{\,{}_i\mathsf{border\text{-}right},\ {}_k\mathsf{border\text{-}left}\,\} \to \{\ {}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}}^{\mathsf{both}},\ {}^i\mathsf{shrink\text{-}top}_{\mathsf{R}}^{\mathsf{test}},\ {}^{i,k}\mathsf{shrink\text{-}test},\ {}^k\mathsf{shrink\text{-}top}_{\mathsf{L}}^{\mathsf{test}},\ {}^k\mathsf{shrink\text{-}bottom}_{\mathsf{L}}^{\mathsf{both}}\ \}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{L}}^{\mathsf{both}},\ {}_i\mathsf{id}\,\} \to \{\,{}^i\mathsf{shrink\text{-}id},\ {}^i\mathsf{shrink\text{-}bottom}_{\mathsf{L}}^{\mathsf{both}}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{L}},\ {}_i\mathsf{id}\,\} \to \{\,{}^i\mathsf{shrink\text{-}id},\ {}^i\mathsf{shrink\text{-}bottom}_{\mathsf{L}}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}id},\ {}^i\mathsf{shrink\text{-}back}_{\mathsf{L}}\,\} \to \{\,{}^i\mathsf{shrink\text{-}back}_{\mathsf{L}},\ {}_i\mathsf{id}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}}^{\mathsf{both}},{}_i\mathsf{rule\text{-}bound}\,\} \to \{\,{}^i\mathsf{shrink\text{-}rule\text{-}bound},{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}}^{\mathsf{both}}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}},{}_i\mathsf{rule\text{-}bound}\,\} \to \{\,{}^i\mathsf{shrink\text{-}rule\text{-}bound},{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}rule\text{-}bound},{}^i\mathsf{shrink\text{-}back}_{\mathsf{R}}\,\} \to \{\,{}^i\mathsf{shrink\text{-}back}_{\mathsf{R}},{}_i\mathsf{rule\text{-}bound}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}}^{\mathsf{both}},{}_i\mathsf{rule\text{-}middle}\,\} \to \{\,{}^i\mathsf{shrink\text{-}rule\text{-}middle},{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}}^{\mathsf{both}}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}},{}_i\mathsf{rule\text{-}middle}\,\} \to \{\,{}^i\mathsf{shrink\text{-}rule\text{-}middle},{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}rule\text{-}middle},{}^i\mathsf{shrink\text{-}back}_{\mathsf{R}}\,\} \to \{\,{}^i\mathsf{shrink\text{-}back}_{\mathsf{R}},{}_i\mathsf{rule\text{-}middle}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}}^{\mathsf{both}},{}_i\mathsf{if}_l\,\} \to \{\,{}^i\mathsf{shrink\text{-}if}^l,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}}^{\mathsf{both}}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}},{}_i\mathsf{if}_l\,\} \to \{\,{}^i\mathsf{shrink\text{-}if}^l,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}if}^l,{}^i\mathsf{shrink\text{-}back}_{\mathsf{R}}\,\} \to \{\,{}^i\mathsf{shrink\text{-}back}_{\mathsf{R}},{}_i\mathsf{if}_l\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}}^{\mathsf{both}},{}_i\mathsf{then}_l\,\} \to \{\,{}^i\mathsf{shrink\text{-}then}^l,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}}^{\mathsf{both}}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}},{}_i\mathsf{then}_l\,\} \to \{\,{}^i\mathsf{shrink\text{-}then}^l,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}then}^l,{}^i\mathsf{shrink\text{-}back}_{\mathsf{R}}\,\} \to \{\,{}^i\mathsf{shrink\text{-}back}_{\mathsf{R}},{}_i\mathsf{then}_l\}\\
\hline
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{L}}^{\mathsf{both}},\ {}_i\mathsf{main}^{\emptyset}\,\} \to \{\,{}^i\mathsf{shrink\text{-}back}_{\mathsf{L}},\ {}_i\mathsf{main}^{\emptyset},\ {}^i\mathsf{shrink\text{-}bottom}_{\mathsf{L}}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}_i\mathsf{main}^{\emptyset},\ {}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}}^{\mathsf{both}}\,\} \to \{\,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}},\ {}_i\mathsf{main}^{\emptyset},\ {}^i\mathsf{shrink\text{-}back}_{\mathsf{R}}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{L}}^{\mathsf{both}},\ {}_i\mathsf{main}^{\emptyset},\ {}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}}^{\mathsf{both}}\,\} \to \{\,{}^i\mathsf{shrink\text{-}back}_{\mathsf{L}},\ {}_i\mathsf{main}^{\emptyset},\ {}^i\mathsf{shrink\text{-}back}_{\mathsf{R}}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{L}},\ {}_i\mathsf{main}^{\emptyset}\,\} \to \{\,{}^i\mathsf{shrink\text{-}back}_{\mathsf{L}},\ {}_i\mathsf{main}^{\emptyset}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}_i\mathsf{main}^{\emptyset},\ {}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}}\,\} \to \{\,{}_i\mathsf{main}^{\emptyset},\ {}^i\mathsf{shrink\text{-}back}_{\mathsf{R}}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}_i\mathsf{border\text{-}right},\ {}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}}\,\} \to \{\,{}^i\mathsf{shrink\text{-}top}_{\mathsf{L}},\ {}^i\mathsf{shrink\text{-}bottom}_{\mathsf{L}}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{L}},\ {}_i\mathsf{border\text{-}right}\,\} \to \{\,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{R}},\ {}^i\mathsf{shrink\text{-}top}_{\mathsf{R}}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{L}},\ {}_i\mathsf{main}^{\emptyset}\,\} \to \{\,{}_i\mathsf{main}^{\emptyset},\ {}^i\mathsf{shrink\text{-}back}_{\mathsf{L}}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}top}_{\mathsf{L}},\ {}^i\mathsf{shrink\text{-}back}_{\mathsf{L}}\,\} \to \{\,{}_i\mathsf{border\text{-}left}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}back}_{\mathsf{R}},\ {}^i\mathsf{shrink\text{-}top}_{\mathsf{R}}\,\} \to \{\,{}_i\mathsf{border\text{-}right}\}\\
\hline
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}top}_{\mathsf{L}}^{\mathsf{test}},\ {}^i\mathsf{shrink\text{-}back}_{\mathsf{L}}\,\} \to \{\,{}^i\mathsf{shrink\text{-}test}^{\mathsf{L}},\ {}_i\mathsf{border\text{-}left}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}back}_{\mathsf{R}},\ {}^i\mathsf{shrink\text{-}top}_{\mathsf{R}}^{\mathsf{test}}\,\} \to \{\,{}_i\mathsf{border\text{-}right},\ {}^i\mathsf{shrink\text{-}test}^{\mathsf{R}}\}\\
\forall i,k{\in}[\![1,\mathfrak{n}]\!],\ k{<}i, & \{\,{}^{i,k}\mathsf{shrink\text{-}test},\ {}^k\mathsf{shrink\text{-}test}^{\mathsf{L}}\,\} \to \{\,{}_k\mathsf{shrink\text{-}test\text{-}ok}\}\\
\forall i,k{\in}[\![1,\mathfrak{n}]\!],\ k{<}i, & \{\,{}^i\mathsf{shrink\text{-}test}^{\mathsf{R}},\ {}_k\mathsf{shrink\text{-}test\text{-}ok}\,\} \to \{\,{}^i\mathsf{test\text{-}start}\}\\
\forall i,k{\in}[\![1,\mathfrak{n}]\!],\ k{<}i, & \{\,{}^i\mathsf{shrink\text{-}test}^{\mathsf{R}},\ {}^{i,k}\mathsf{shrink\text{-}test},\ {}^k\mathsf{shrink\text{-}test}^{\mathsf{L}}\,\} \to \{\,{}^i\mathsf{test\text{-}start}\}\\
\hline
\forall i,k{\in}[\![1,\mathfrak{n}]\!],\ k{<}i, & \{\,{}^i\mathsf{shrink\text{-}test}^{\mathsf{R}},\ {}^{i,k}\mathsf{shrink\text{-}test}\,\} \to \{\,{}_i\mathsf{shrink\text{-}test\text{-}fail}\}\\
\forall i,k{\in}[\![1,\mathfrak{n}]\!],\ k{<}i, & \{\,{}_i\mathsf{shrink\text{-}test\text{-}fail},\ {}^k\mathsf{shrink\text{-}test}^{\mathsf{L}}\,\} \to \{\,{}^k\mathsf{shrink\text{-}order}\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{\,{}^i\mathsf{shrink\text{-}order},\ {}_i\mathsf{border\text{-}left}\,\} \to \{\,{}^i\mathsf{shrink\text{-}bottom}_{\mathsf{L}}^{\mathsf{both}},\ {}^i\mathsf{shrink\text{-}top}_{\mathsf{L}}\}
\end{array}
$$

Figure 20: Collision rules for shrinking and test relative width.

Thanks to the choice of relative width of each half of each macro-signal, the point O_L and O_R have coordinates $\left(x_{\mathsf{O}_L} = -2s^{\mathsf{max}} - 2s^{\mathsf{max}}\frac{s^{\mathsf{max}}}{s^{\mathsf{rapid}}}, t_{\mathsf{O}_L} = \frac{2s^{\mathsf{max}}}{s^{\mathsf{rapid}}}\right)$ and $\left(x_{\mathsf{O}_R} = 2s^{\mathsf{max}} - 2s^{\mathsf{max}}\frac{s^{\mathsf{max}}}{s^{\mathsf{rapid}}}, t_{\mathsf{O}_R} = \frac{2s^{\mathsf{max}}}{s^{\mathsf{rapid}}}\right)$. The point C_m has coordinates $\left(x_m = x_{\mathsf{O}_L} + 4s^{\mathsf{max}}\frac{s_{m+1}}{s_{m+1}-s_m}, t_m = t_{\mathsf{O}_L} + \frac{4s^{\mathsf{max}}}{s_{m+1}-s_m}\right)$.

We take, as coordinates of Z_T:

$$\mathrm{Z}_T\,(x_T = 0,\ t_T = 2\max\{\,t_m \mid 1 \le m < \mathfrak{n}\,\})\ .$$

So that, at time $\max\{\,t_m\}$, all the signals are separated and still within the safe zone.

By setting the point Z_L and Z_R to be to be on the line $t = \varepsilon - 1$ for a sufficiently small positive ε, they have coordinates $(x_L = -s^{\mathsf{max}}(t_T - \varepsilon), t_L = \varepsilon - 1)$ and $(x_R = s^{\mathsf{max}}(t_T - \varepsilon), t_R = \varepsilon - 1)$.

It is enough to ensure that no signal (except for probing ones) enters through the bottom of the safety zone since their speeds prevent them from entering from the other two sides. The scheme to send signals to Z_L and Z_R is depicted in Fig. 22.

After the shrinking, pairs of fast enough signals are issued on both side so that they meet on the extreme points: ${}^i\mathsf{test\text{-}left}$ and ${}^{i,k}\mathsf{test\text{-}left\text{-}up}$ on the left and ${}^i\mathsf{test\text{-}right}$ and ${}^i\mathsf{test\text{-}right\text{-}up}$ on the right. The signals ${}^i\mathsf{test\text{-}left}$, ${}^i\mathsf{test\text{-}right}$ and ${}^i\mathsf{test\text{-}right\text{-}up}$ are issued from ${}_i\mathsf{main}^{\emptyset}$ while ${}^{i,k}\mathsf{test\text{-}left\text{-}up}$ and ${}^{i,k}\mathsf{test\text{-}right}^k$ (later to become ${}^{i,k}\mathsf{test\text{-}right}^j$) are issued from the collision between ${}_k\mathsf{main}^{\emptyset}$ and ${}^i\mathsf{test\text{-}right}$ at $U^{i,k}$.

On the left, after crossing ${}_i\mathsf{border\text{-}left}$, if signal ${}^i\mathsf{test\text{-}left}$ meets nothing before ${}^{i,k}\mathsf{test\text{-}left\text{-}up}$ then it returns as ${}^i\mathsf{test\text{-}left\text{-}ok}$ (and ${}^{i,k}\mathsf{test\text{-}left\text{-}up}$ is destroyed). When ${}^i\mathsf{test\text{-}left\text{-}ok}$ meet ${}_i\mathsf{main}^{\emptyset}$, then the later turns into ${}_i\mathsf{main}_{\mathsf{test}}^{\mathsf{ok}}$ to record the success on left. Otherwise, anything met on the left is either too close or participates in the macro-collision (and ${}_i\mathsf{main}^{\emptyset}$ is not the left-most involved). In both cases, the macro-collision should

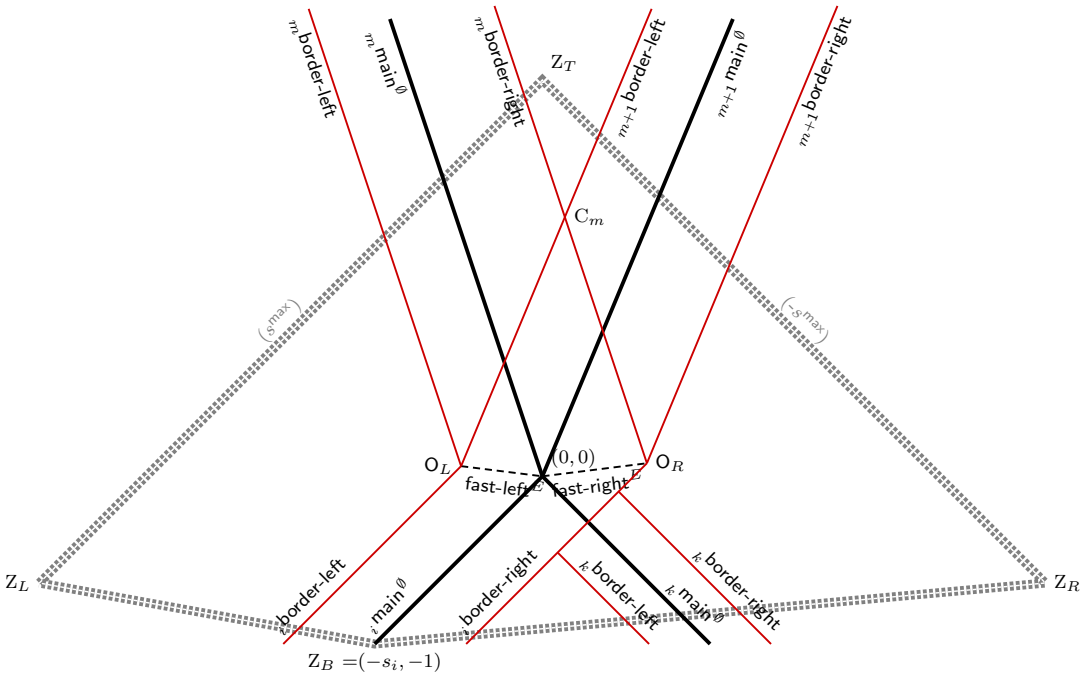
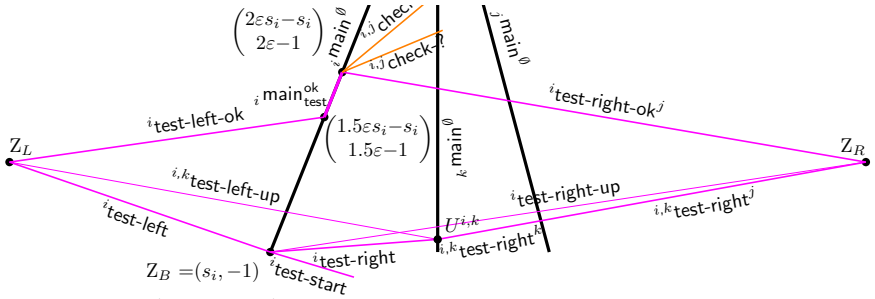


Figure 21: Safety zone (inside the dotted perimeter).



The point where $_{i}$main$^{\emptyset}$ and $_{k}$main$^{\emptyset}$ meet has coordinate $(0, 0)$. For clarity, border signals are not displays.

Figure 22: Testing for the safety zone and identifying the rightmost speed.

be aborted. This is done by coming back as ^{i}test-left-fail. This signal cancels ^{i,k}test-left-up and any signal returning from the right side and disappears thus preventing the macro-collision from being initiated. This is depicted in Fig. 26 where it can be seen that the process is restarted later with success.

On the right, ^{i}test-right tests for obvious non participating macro-signals and collects the index of the rightmost potentially participating macro-signal (next stage checks whether all potentially participating signals are rightly positioned). It verifies that $_{j}$main$^{\emptyset}$ are in strictly decreasing speed order. It also verifies that a $_{j}$main$^{\emptyset}$ is reached for any $_{j}$border-left encountered by turning into ^{i,k}test-right-wait^{j} in between (this is not indicated in Fig. 22). The signal ^{i,k}test-right^{j} also initiate a shrinking process on each macro-signal on the right when it meets a $_{l}$border-left (to avoid useless macro-collision initialisation).

Signal ^{i,k}test-right^{j} updates the least speed index encountered when it meets $_{l}$border-left (becoming ^{i,k}test-right-wait^{l}) with $l < j$ and at crossing $_{l}$main$^{\emptyset}$ becomes ^{i,k}test-right^{l}. When ^{i,k}test-right^{j} and ^{i}test-right-up meet, they comes back as ^{i}test-right-ok^{j}. That way, it brings back the index of the rightmost speed. When ^{i}test-right-ok^{j} meets $_{i}$main$^{\text{ok}}_{\text{test}}$, the next stage of the macro-collision starts.

The signals ^{i}test-left and ^{i}test-right-up head straight to Z_L and Z_R, so that their speed are: $\frac{x_L+s_i}{\varepsilon}$ and $\frac{x_R+s_i}{\varepsilon}$ $(= s_i^{\text{test-right-up}})$ respectively. The speed of ^{i}test-right can be anything greater than the speed of ^{i}test-right-up, double that speed $(= 2s_i^{\text{test-right-up}})$ for example.

To compute the other speed, some coordinates have to be computed (in particular $U^{i,k}$). This is straightforward once an appropriate formula is given. If the speeds (α, β and s) are given as in Fig. 23, then the coordinates of the intersection point, M, are:

$$\left(-\beta\frac{s-\alpha}{s-\beta}, -\frac{s-\alpha}{s-\beta}\right) \quad . \tag{1}$$

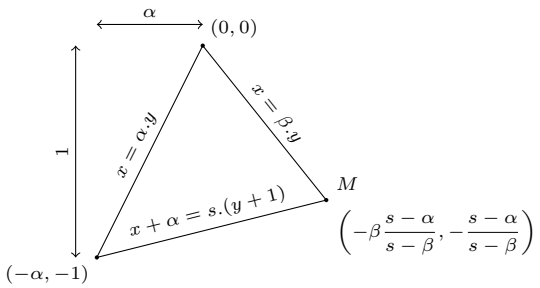


Figure 23: Computing coordinates.

Thus, the coordinates of $U^{i,k}$ are:

$$\left(x_U^{i,k} = -s_k\frac{2s_i^{\text{test-right-up}} - s_i}{2s_i^{\text{test-right-up}} - s_k}, t_U^{i,k} = -\frac{2s_i^{\text{test-right-up}} - s_i}{2s_i^{\text{test-right-up}} - s_k}\right) \quad .$$

The used meta-signals and (success) collision rules are defined in Figs. 24 and 25. To be coherent with Fig. 3, the duration of 2ε with a start at $8/10$ is assured with the value: $\varepsilon = 1/16$.

Parameter	Value
$\varepsilon =$	$\frac{1}{16}$
$\forall i{\in}[\![1,\mathfrak{n}]\!],\ s_i^{\text{test-right-up}} =$	$\frac{x_R + s_i}{\varepsilon}$
$\forall i,k{\in}[\![1,\mathfrak{n}]\!],\ k{<}i,\ t_U^{i,k} =$	$-s_k\frac{2s_i^{\text{test-right-up}} - s_i}{2s_i^{\text{test-right-up}} - s_k}$
$\forall i,k{\in}[\![1,\mathfrak{n}]\!],\ k{<}i,\ x_U^{i,k} =$	$s_k.t_U^{i,k}$
$\forall i{\in}[\![1,\mathfrak{n}]\!],\ s_{i,k}^{\text{test-right}} =$	$\frac{x_R - x_U^{i,k}}{t_R - t_U^{i,k}}$
$\forall i{\in}[\![1,\mathfrak{n}]\!],\ s_i^{\text{test-right-back}} =$	$\frac{2\varepsilon.s_i - s_i - x_R}{\varepsilon}$
$\forall i{\in}[\![1,\mathfrak{n}]\!],\ s_i^{\text{test-left-back}} =$	$\frac{1.5\varepsilon.s_i - s_i - x_L}{0.5\varepsilon}$

	Meta-signal	speed
$\forall i{\in}[\![1,\mathfrak{n}]\!],$	^{i}test-left	$\frac{x_L+s_i}{\varepsilon}$
$\forall i{\in}[\![1,\mathfrak{n}]\!],$	^{i}test-right-up	$s_i^{\text{test-right-up}}$
$\forall i{\in}[\![1,\mathfrak{n}]\!],$	^{i}test-right	$2s_i^{\text{test-right-up}}$
$\forall i,j,k{\in}[\![1,\mathfrak{n}]\!],\ j{\leq}k{<}i,$	^{i,k}test-right^j	$s_{i,k}^{\text{test-right}}$
$\forall i,j,k{\in}[\![1,\mathfrak{n}]\!],\ j{<}k{<}i,$	^{i,k}test-right-wait^j	$s_{i,k}^{\text{test-right}}$
$\forall i,k{\in}[\![1,\mathfrak{n}]\!],\ k{<}i,$	^{i,k}test-left-up	$\frac{x_L - x_U^{i,k}}{t_L - t_U^{i,k}}$
$\forall i,k{\in}[\![1,\mathfrak{n}]\!],\ k{<}i,$	^{i}test-left-ok	$s_i^{\text{test-left-back}}$
$\forall i,j{\in}[\![1,\mathfrak{n}]\!],\ j{<}i,$	^{i}test-right-ok^j	$s_i^{\text{test-right-back}}$
$\forall i{\in}[\![1,\mathfrak{n}]\!],$	${}_i$main$_{\text{test}}^{\text{ok}}$	s_i
$\forall i,k{\in}[\![1,\mathfrak{n}]\!],\ k{<}i,$	^{i}test-left-fail	$s_i^{\text{test-left-back}}$
$\forall i,j{\in}[\![1,\mathfrak{n}]\!],\ j{<}i,$	^{i}test-right-fail	$s_i^{\text{test-right-back}}$
$\forall i{\in}[\![1,\mathfrak{n}]\!],$	${}_i$main$_{\text{test}}^{\text{fail-l}}$	s_i
$\forall i{\in}[\![1,\mathfrak{n}]\!],$	${}_i$main$_{\text{test}}^{\text{fail-r}}$	s_i

Figure 24: Meta-signals for testing.

5.2.1 Test failure

As depicted on Fig. 26, the test can fail because of the presence of unwanted signals on the left or on the right. Both cases are briefly presented.

$$
\begin{aligned}
\forall i{\in}[\![1,\mathfrak{n}]\!], &\quad \{\,{}_i\mathsf{main}^{\emptyset},\ {}^i\mathsf{test\text{-}start}\,\} &&\to \{\,{}^i\mathsf{test\text{-}left}, {}_i\mathsf{main}^{\emptyset}, {}^i\mathsf{test\text{-}right\text{-}up}, {}^i\mathsf{test\text{-}right}\,\}\\
\forall i,k{\in}[\![1,\mathfrak{n}]\!],\ k{<}i, &\quad \{\,{}^i\mathsf{test\text{-}right},\ {}_k\mathsf{main}^{\emptyset}\,\} &&\to \{\,{}^{i,k}\mathsf{test\text{-}left\text{-}up}, {}_k\mathsf{main}^{\emptyset}, {}^{i,k}\mathsf{test\text{-}right}^k\,\}\\
\forall i,j,k,l{\in}[\![1,\mathfrak{n}]\!],\ l{<}j{\le}k{\le}i, &\quad \{\,{}^{i,k}\mathsf{test\text{-}right}^j,\ {}_l\mathsf{border\text{-}left}\,\} &&\to \{\,{}^l\mathsf{shrink\text{-}top}_{\mathsf{L}}, {}^{i,k}\mathsf{test\text{-}right\text{-}wait}^l, {}^l\mathsf{shrink\text{-}bottom}_{\mathsf{L}}^{\mathsf{both}}\,\}\\
\forall i,j,k{\in}[\![1,\mathfrak{n}]\!],\ j{<}k{<}i, &\quad \{\,{}^{i,k}\mathsf{test\text{-}right\text{-}wait}^j,\ {}_j\mathsf{main}^{\emptyset}\,\} &&\to \{\,{}_j\mathsf{main}^{\emptyset}, {}^{i,k}\mathsf{test\text{-}right}^j\,\}\\
\forall i,k{\in}[\![1,\mathfrak{n}]\!],\ k{<}i, &\quad \{\,{}^i\mathsf{test\text{-}left},\ {}^{i,k}\mathsf{test\text{-}left\text{-}up}\,\} &&\to \{\,{}^i\mathsf{test\text{-}left\text{-}ok}\,\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], &\quad \{\,{}^i\mathsf{test\text{-}left\text{-}ok},\ {}_i\mathsf{main}^{\emptyset}\,\} &&\to \{\,{}_i\mathsf{main}^{\mathsf{ok}}_{\mathsf{test}}\,\}\\
\forall i,j,k{\in}[\![1,\mathfrak{n}]\!],\ j{\le}k{<}i, &\quad \{\,{}^i\mathsf{test\text{-}right\text{-}up},\ {}^{i,k}\mathsf{test\text{-}right}^j\,\} &&\to \{\,{}^i\mathsf{test\text{-}right\text{-}ok}^j\,\}\\
\forall i{\in}[\![1,\mathfrak{n}]\!], \{\,{}^i\mathsf{test\text{-}right\text{-}up},\ {}^{i,j}\mathsf{test\text{-}right\text{-}wait}^j,\ {}_j\mathsf{main}^{\emptyset}\,\} &\to \{\,{}^i\mathsf{test\text{-}right\text{-}ok}^j,\ {}_j\mathsf{main}^{\emptyset}\,\}\\
\forall i,j{\in}[\![1,\mathfrak{n}]\!],\ j{<}i, &\quad \{\,{}_i\mathsf{main}^{\mathsf{ok}}_{\mathsf{test}},\ {}^i\mathsf{test\text{-}right\text{-}ok}^j\,\} &&\to \{\,{}_i\mathsf{main}^{\emptyset}, {}^{i,j}\mathsf{check\text{-}up}, {}^{i,j}\mathsf{check\text{-}?}\,\}
\end{aligned}
$$

Figure 25: Collision rules for testing, success case.

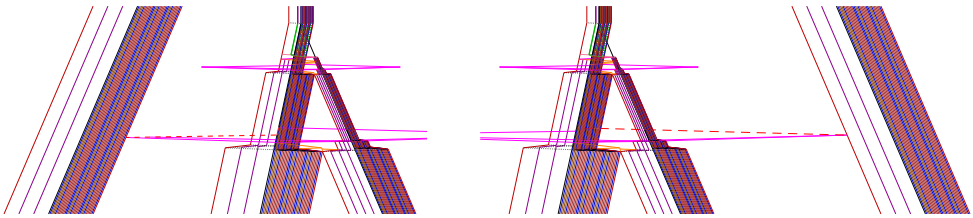


Figure 26: Detection of a signal not participating on the left and on the right.

Figure 27 presents the scheme for failure of test on the left. This happens if ${}^i\mathsf{test\text{-}left}$ encounters anything: it is either something that should not be involved and is too close or just that ${}_i\mathsf{main}^{\emptyset}$ is not the leftmost macro-signal involved, so that macro-collision has to be aborted to be started by the rightful macro-signal. Since probe signals are not concerned, the signals that can be met on the left are: ${}_l\mathsf{main}^{\emptyset}$, $\mathsf{ready}^{\emptyset}_l$, ${}_l\mathsf{border\text{-}right}$ or ${}^l\mathsf{shrink\text{-}top}_{\mathsf{R}}$ (for any l).

On meeting, ${}^i\mathsf{test\text{-}left}$ bounces back as ${}^i\mathsf{test\text{-}left\text{-}fail}$. It arrives back to ${}_i\mathsf{main}^{\emptyset}$ and marks it as ${}_i\mathsf{main}^{\mathsf{fail\text{-}l}}_{\mathsf{test}}$. When ${}^i\mathsf{test\text{-}right\text{-}ok}^j$ meets ${}_i\mathsf{main}^{\mathsf{fail\text{-}l}}_{\mathsf{test}}$, it is destroyed and ${}_i\mathsf{main}^{\emptyset}$ is restored; nothing is emitted so that the macro-collision is aborted. The signal ${}^{i,k}\mathsf{test\text{-}left\text{-}up}$ has to be disposed of; this is done either on ${}^i\mathsf{test\text{-}left\text{-}fail}$ or on ${}_i\mathsf{main}^{\mathsf{fail\text{-}l}}_{\mathsf{test}}$.

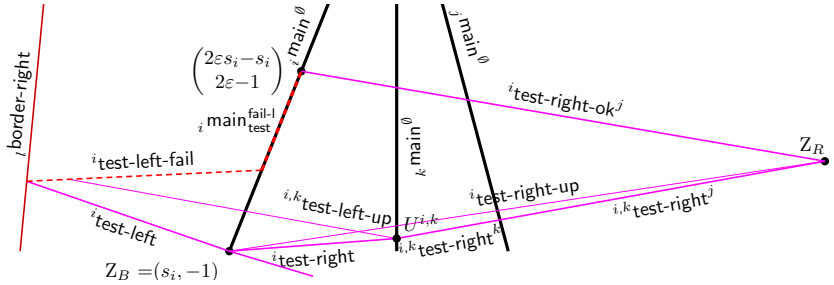


Figure 27: Testing for the safety zone, fail on left.

Figure 28 presents the scheme for failure of test on the right. The alternation of ${}^{i,j}\mathsf{test\text{-}right}^l$ and ${}^{i,j}\mathsf{test\text{-}right\text{-}wait}^l$ is not indicated for clarity although they are used to detect failure. The failure might come from some ${}_l\mathsf{border\text{-}left}$ with l too small or from ${}^i\mathsf{test\text{-}right\text{-}up}$ meeting ${}^{i,j}\mathsf{test\text{-}right\text{-}wait}^l$, i.e. before ${}_l\mathsf{main}^{\emptyset}$ is met. In any failure case, ${}^{i,j}\mathsf{test\text{-}right}^l$ (or ${}^{i,j}\mathsf{test\text{-}right\text{-}wait}^l$) bounces back as ${}^i\mathsf{test\text{-}right\text{-}fail}$. When ${}^i\mathsf{test\text{-}right\text{-}fail}$ meets ${}_i\mathsf{main}^{\mathsf{ok}}_{\mathsf{test}}$, it is destroyed and ${}_i\mathsf{main}^{\emptyset}$ is restored; nothing is emitted so that the macro-collision is aborted. The signal ${}^i\mathsf{test\text{-}right\text{-}up}$ has to be disposed, this happens on meeting ${}^i\mathsf{test\text{-}right\text{-}fail}$.

It might happen that the ${}^i\mathsf{test\text{-}right\text{-}fail}$ arrives before ${}^i\mathsf{test\text{-}left\text{-}ok}$ onto ${}_i\mathsf{main}^{\emptyset}$. It might also happen that there is failure on both side and arrival onto ${}_i\mathsf{main}^{\emptyset}$ can be in any order. The listed rules do take this into account.

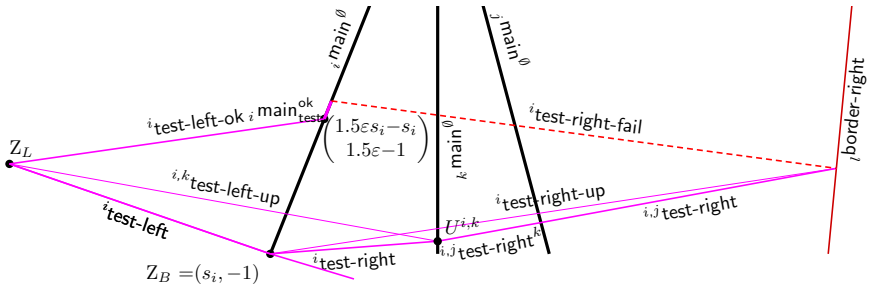


Figure 28: Testing for the safety zone, fail on right.

The used meta-signals and (failure) collision rules are defined in Figs. 24 and 29. The rules in Figure 29 are divided in three part: fail on left only, fail on right only and additional rules in case fail on both left and right.

$$
\begin{array}{rr}
\forall i,l{\in}[\![1,\mathfrak{n}]\!], & \{{}_l\mathsf{border\text{-}right},\ {}^i\mathsf{test\text{-}left}\ \} \to \{\ {}_l\mathsf{border\text{-}right},\ {}^i\mathsf{test\text{-}left\text{-}fail}\} \\
\forall i,l{\in}[\![1,\mathfrak{n}]\!], & \{{}^l\mathsf{shrink\text{-}top}_{\mathsf{R}},\ {}^i\mathsf{test\text{-}left}\ \} \to \{\ {}^l\mathsf{shrink\text{-}top}_{\mathsf{R}},\ {}^i\mathsf{test\text{-}left\text{-}fail}\} \\
\forall i,l{\in}[\![1,\mathfrak{n}]\!], & \{\mathsf{ready}^{\emptyset}_l,\ {}^i\mathsf{test\text{-}left}\ \} \to \{\ \mathsf{ready}^{\emptyset}_l,\ {}^i\mathsf{test\text{-}left\text{-}fail}\} \\
\forall i,l{\in}[\![1,\mathfrak{n}]\!], & \{{}_l\mathsf{main}^{\emptyset},\ {}^i\mathsf{test\text{-}left}\ \} \to \{\ {}_l\mathsf{main}^{\emptyset},\ {}^i\mathsf{test\text{-}left\text{-}fail}\} \\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{{}^i\mathsf{test\text{-}left\text{-}fail},\ {}^{i,k}\mathsf{test\text{-}left\text{-}up}\ \} \to \{\ {}^i\mathsf{test\text{-}left\text{-}fail}\} \\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{{}^i\mathsf{test\text{-}left\text{-}fail},\ {}_i\mathsf{main}^{\emptyset}\ \} \to \{\ {}_i\mathsf{main}^{\mathsf{fail\text{-}l}}_{\mathsf{test}}\} \\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{{}_i\mathsf{main}^{\mathsf{fail\text{-}l}}_{\mathsf{test}},\ {}^{i,k}\mathsf{test\text{-}left\text{-}up}\ \} \to \{\ {}_i\mathsf{main}^{\mathsf{fail\text{-}l}}_{\mathsf{test}}\} \\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{{}_i\mathsf{main}^{\mathsf{fail\text{-}l}}_{\mathsf{test}},\ {}^i\mathsf{test\text{-}right\text{-}ok}^j\ \} \to \{\ {}_i\mathsf{main}^{\emptyset}\} \\
\hline
\forall i,k,l{\in}[\![1,\mathfrak{n}]\!],\ k{<}i,\ k{\le}l, & \{{}^{i,j}\mathsf{test\text{-}right}^k,\ {}_l\mathsf{border\text{-}left}\ \} \to \{\ {}^i\mathsf{test\text{-}right\text{-}fail},\ {}_l\mathsf{border\text{-}left}\} \\
\forall i,k,l{\in}[\![1,\mathfrak{n}]\!],\ k{<}i,\ k{\le}l, & \{{}^{i,j}\mathsf{test\text{-}right}^k,\ {}^l\mathsf{shrink\text{-}top}_{\mathsf{L}}\ \} \to \{\ {}^i\mathsf{test\text{-}right\text{-}fail},\ {}^l\mathsf{shrink\text{-}top}_{\mathsf{L}}\} \\
\forall i,j,k{\in}[\![1,\mathfrak{n}]\!],\ j{<}k{<}i, & \{{}^i\mathsf{test\text{-}right\text{-}up},\ {}^{i,j}\mathsf{test\text{-}right\text{-}wait}^j\ \} \to \{\ {}^i\mathsf{test\text{-}right\text{-}fail}\} \\
\forall i,k{\in}[\![1,\mathfrak{n}]\!],\ k{<}i, & \{{}^i\mathsf{test\text{-}right\text{-}fail},\ {}^i\mathsf{test\text{-}right\text{-}up}\ \} \to \{\ {}^i\mathsf{test\text{-}right\text{-}fail}\} \\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{{}_i\mathsf{main}^{\emptyset},\ {}^i\mathsf{test\text{-}right\text{-}fail}\ \} \to \{\ {}_i\mathsf{main}^{\mathsf{fail\text{-}r}}_{\mathsf{test}}\} \\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{{}^i\mathsf{test\text{-}left\text{-}ok},\ {}_i\mathsf{main}^{\mathsf{fail\text{-}r}}_{\mathsf{test}}\ \} \to \{\ {}_i\mathsf{main}^{\emptyset}\} \\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{{}_i\mathsf{main}^{\mathsf{ok}}_{\mathsf{test}},\ {}^i\mathsf{test\text{-}right\text{-}fail}\ \} \to \{\ {}_i\mathsf{main}^{\emptyset}\} \\
\hline
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{{}^i\mathsf{test\text{-}left\text{-}fail},\ {}_i\mathsf{main}^{\mathsf{fail\text{-}r}}_{\mathsf{test}}\ \} \to \{\ {}_i\mathsf{main}^{\emptyset}\} \\
\forall i{\in}[\![1,\mathfrak{n}]\!], & \{{}_i\mathsf{main}^{\mathsf{fail\text{-}l}}_{\mathsf{test}},\ {}^i\mathsf{test\text{-}right\text{-}fail}\ \} \to \{\ {}_i\mathsf{main}^{\emptyset}\}
\end{array}
$$

Figure 29: Collision rules for testing, failure cases.

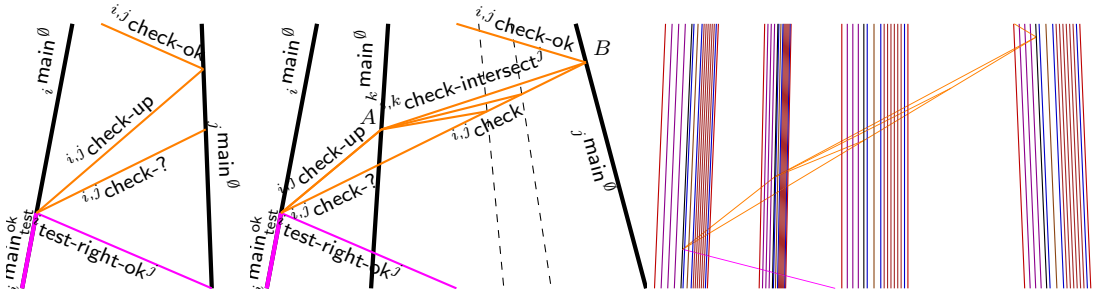
The function interp can be trivially extended for this section by ignoring all of its signals, as they don't affect the identity of macro-signals. Again, the number of additional signals accounted for by this section for one collision is bounded, therefore interp is indeed defined locally.

5.3 Check participating signals

From this point, it is known that the index of involved macro-signals ranges from j to i (included). But it is not known whether they actually participate in one single macro-collision (the situations in Fig. 2 are not yet distinguished).

To check this, the two first $\_\mathsf{main}^{\emptyset}$ ($_i\mathsf{main}^{\emptyset}$ and $_k\mathsf{main}^{\emptyset}$) are used to organise meeting points with the all potential $_l\mathsf{main}^{\emptyset}$ ($l \in [\![j, k-1]\!]$). If any appear anywhere except at their assigned meeting point, then it is known that it will not pass where $_i\mathsf{main}^{\emptyset}$ and $_k\mathsf{main}^{\emptyset}$ intersect (and the macro-collision aborts). The meeting points are computed according to the speeds (like in Fig. 22). This constructions is presented on Fig. 30 with potential $_l\mathsf{main}^{\emptyset}$ dashed. The equation (1) is used again to compute the intersection points and to deduce the speeds.

Signal $^{i,j}\mathsf{check\text{-}?}$ and slower signal $^{i,j}\mathsf{check\text{-}up}$ go on the right. If the first $\_\mathsf{main}^{\emptyset}$ $^{i,j}\mathsf{check\text{-}?}$ meets is $_j\mathsf{main}^{\emptyset}$, then there are only two macro-signals involved, it disappears and lets $^{i,j}\mathsf{check\text{-}up}$ starts the next stage.



(a) scheme (b) example

Figure 30: Testing the other $\_\mathsf{main}^{\emptyset}$ signals.

Otherwise, ${}^{i,j}\mathsf{check\text{-}?}$ turns to ${}^{i,j}\mathsf{check}$ which crosses the configuration until it meets ${}_{j}\mathsf{main}^{\emptyset}$ or a mismatch. ${}^{i,j}\mathsf{check\text{-}up}$ cross ${}_{k}\mathsf{main}^{\emptyset}$ at A and branches into several ${}^{i,k}\mathsf{check\text{-}intersect}^{m}$. Each time ${}^{i,j}\mathsf{check}$ meets some ${}_{l}\mathsf{main}^{\emptyset}$, then there should also be the corresponding ${}^{i,k}\mathsf{check\text{-}intersect}^{l}$. And the resolution is started.

If any ${}_{l}\mathsf{main}^{\emptyset}$ is met with the wrong ${}^{i,k}\mathsf{check\text{-}intersect}^{m}$ ($m \neq l$) or without any, ${}^{i,j}\mathsf{check}$ turns into ${}^{i,j}\mathsf{check\text{-}fail}$. This latter cancels all remaining ${}^{i,k}\mathsf{check\text{-}intersect}^{l}$ and disappears with ${}^{i,k}\mathsf{check\text{-}intersect}^{j}$. The resolution is not started.

The used meta-signals and collision rules are defined in Figs. 31 and 32. The speed of ${}^{i,j}\mathsf{check\text{-}up}$ has to be fast enough so that A occurs before ${}^{i,j}\mathsf{check}$ intersects any ${}_{k}\mathsf{main}^{\emptyset}$. The position of A and of such intersections are again obtained using (1), yielding the necessary speeds of the signals ${}^{i,k}\mathsf{check\text{-}intersect}^{j}$.

Once again, interp simply ignores the new meta-signals and collisions, which are in bounded amount for each collision.

Altogether, the devices presented in this section yield the following lemma, which states that U correctly simulates one collision from clean inputs.

Lemma 10. *Let ρ be a collision rule of $\mathcal{A}$, and let a be a configuration of $\mathcal{A}$ whose signals are exactly ρ^- and the positions of these signals are such that they all meet at some point (x,t). Let $\mathbb{A}$ be the associated space-time diagram.*

Let b be a configuration such that $\mathsf{interp}(b) = a$, and which is clean at every position of a signal in a. Then there is a δ and $t' < t$ such that $\mathbb{B}(t')$ is a δ-checked configuration for a.

Parameter	Value
τ_{check}	$= \frac{101}{100}$
$\forall 1 \leq k \leq i,\ h_{i,k}$	$= \frac{s_i - s^{\mathsf{rapid}}}{s^{\mathsf{rapid}} - s_{k-1}} \tau_{\mathrm{check}}$
$s^{\mathsf{chk\text{-}up}}$	$= \max\left(\frac{s^{\mathsf{rapid}}}{2}, \max_{1\leq k\leq i}\left(\frac{h_{i,k}s_k + s_i}{h_{i,k}+1}\right)\right)$
$\forall i,k \in [\![1,\mathfrak{n}]\!],\ k<i,\ t^{\mathsf{chk\text{-}start}}_{i,k}$	$= \frac{s_i - s^{\mathsf{rapid}}}{s^{\mathsf{rapid}} - s_k}$
$\forall i,k \in [\![1,\mathfrak{n}]\!],\ k<i,\ t^{\mathsf{intersect}}_{i,k}$	$= \frac{s_i - s^{\mathsf{chk\text{-}up}}}{s^{\mathsf{chk\text{-}up}} - s_k}$

	Meta-signal	speed
$\forall i,j \in [\![1,\mathfrak{n}]\!],\ j<i,$	${}^{i,j}\mathsf{check\text{-}?}$	s^{rapid}
$\forall i,j \in [\![1,\mathfrak{n}]\!],\ j<i,$	${}^{i,j}\mathsf{check}$	s^{rapid}
$\forall i,j \in [\![1,\mathfrak{n}]\!],\ j<i,$	${}^{i,j}\mathsf{check\text{-}up}$	$s^{\mathsf{chk\text{-}up}}$
$\forall i,j \in [\![1,\mathfrak{n}]\!],\ j<i,$	${}^{i,k}\mathsf{check\text{-}intersect}^{l}$	$\frac{s_l . t^{\mathsf{intersect}}_{i,k} - s_k . t^{\mathsf{chk\text{-}start}}_{i,k}}{t^{\mathsf{intersect}}_{i,k} - t^{\mathsf{chk\text{-}start}}_{i,k}}$
$\forall i,j \in [\![1,\mathfrak{n}]\!],\ j<i,$	${}^{i,j}\mathsf{check\text{-}fail}$	$-s^{\mathsf{rapid}}$

Figure 31: Meta-signals for checking.

$$
\begin{array}{rr}
\forall i,j{\in}[\![1,\mathfrak{n}]\!],\ j{<}i, & \{{}^{i,j}\mathsf{check\text{-}?},\ {}_{j}\mathsf{main}^{\emptyset}\} \to \{{}_{j}\mathsf{main}^{\emptyset}\} \\
\forall i,j{\in}[\![1,\mathfrak{n}]\!],\ j{<}i, & \{{}^{i,j}\mathsf{check\text{-}up},\ {}_{j}\mathsf{main}^{\emptyset}\} \to \{{}^{i,j}\mathsf{check\text{-}ok},\ {}_{j}\mathsf{main}^{\emptyset}\} \\
\forall i,j,k{\in}[\![1,\mathfrak{n}]\!],\ j{<}k{<}i, & \{{}^{i,j}\mathsf{check\text{-}?},\ {}_{k}\mathsf{main}^{\emptyset}\} \to \{{}_{k}\mathsf{main}^{\emptyset},\ {}^{i,j}\mathsf{check}\} \\
\forall i,j,k{\in}[\![1,\mathfrak{n}]\!],\ j{<}k{<}i, & \{{}^{i,j}\mathsf{check\text{-}up},\ {}_{k}\mathsf{main}^{\emptyset}\} \to \{{}^{i,k}\mathsf{check\text{-}intersect}^{l}\}_{j\le l<k} \cup \{{}_{k}\mathsf{main}^{\emptyset}\} \\
\forall i,j,k,l{\in}[\![1,\mathfrak{n}]\!],\ j{<}l{<}k{<}i, & \{{}^{i,k}\mathsf{check\text{-}intersect}^{l},\ {}^{i,j}\mathsf{check}\} \to \{{}^{i,j}\mathsf{check}\} \\
\forall i,j,k,l{\in}[\![1,\mathfrak{n}]\!],\ j{<}l{<}k{<}i, & \{{}^{i,k}\mathsf{check\text{-}intersect}^{l},\ {}^{i,j}\mathsf{check},\ {}_{l}\mathsf{main}^{\emptyset}\} \to \{{}_{l}\mathsf{main}^{\emptyset},\ {}^{i,j}\mathsf{check}\} \\
\forall i,j,k{\in}[\![1,\mathfrak{n}]\!],\ j{<}k{<}i, & \{{}^{i,k}\mathsf{check\text{-}intersect}^{j},\ {}^{i,j}\mathsf{check},\ {}_{j}\mathsf{main}^{\emptyset}\} \to \{{}^{i,j}\mathsf{check\text{-}ok},\ {}_{j}\mathsf{main}^{\emptyset}\} \\
\hline
\forall i,j,l{\in}[\![1,\mathfrak{n}]\!],\ j{<}l{<}i, & \{{}^{i,j}\mathsf{check},\ {}_{l}\mathsf{main}^{\emptyset}\} \to \{{}^{i,j}\mathsf{check\text{-}fail},\ {}_{l}\mathsf{main}^{\emptyset}\} \\
\forall i,j,k,l,m{\in}[\![1,\mathfrak{n}]\!],\ j{\le}l{<}k{<}i, l\neq m, & \{{}^{i,j}\mathsf{check},\ {}_{l}\mathsf{main}^{\emptyset},\ {}^{i,k}\mathsf{check\text{-}intersect}^{m}\} \to \{{}_{l}\mathsf{main}^{\emptyset},\ {}^{i,j}\mathsf{check\text{-}fail}\} \\
\forall i,j,k,l{\in}[\![1,\mathfrak{n}]\!],\ j{<}l{<}k{<}i, & \{{}^{i,j}\mathsf{check\text{-}fail},\ {}^{i,k}\mathsf{check\text{-}intersect}^{l}\} \to \{{}^{i,j}\mathsf{check\text{-}fail}\} \\
\forall i,j,k,l{\in}[\![1,\mathfrak{n}]\!],\ j{<}l{<}k{<}i, & \{{}^{i,j}\mathsf{check\text{-}fail},\ {}^{i,k}\mathsf{check\text{-}intersect}^{j}\} \to \{\} \\
\forall i,j,k{\in}[\![1,\mathfrak{n}]\!],\ j{<}k{<}i, & \{{}^{i,j}\mathsf{check},\ {}^{i,k}\mathsf{check\text{-}intersect}^{j}\} \to \{{}^{i,j}\mathsf{check\text{-}fail}\}
\end{array}
$$

Figure 32: Collision rules for checking.

6 Simulation examples

The presented construction works and has been implemented. It has been entirely programmed in an ad hoc language for signal machines. Given a signal machine, the library generates the corresponding $\mathcal{U}_{\mathcal{S}}$ together with a function to translate initial configurations. This has been used to generate all the pictures. Figure 33 presents a simulation of the dynamics in Fig. 1a.

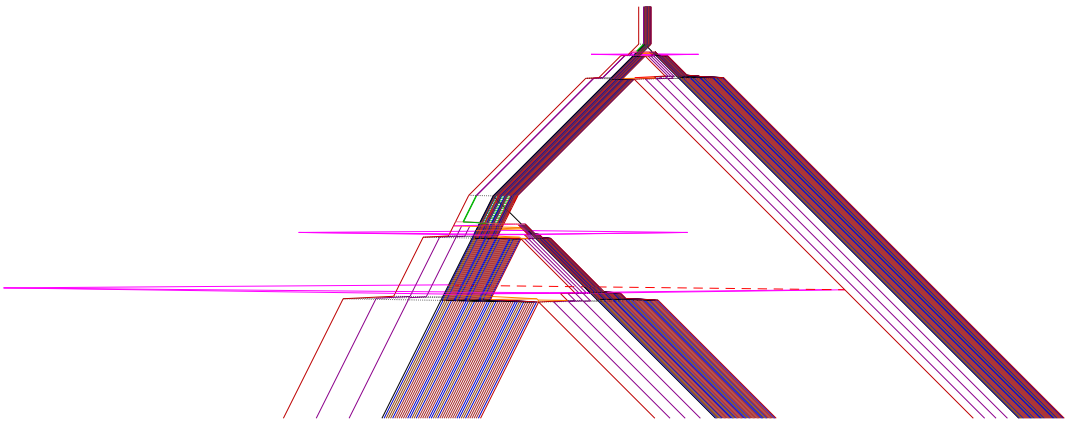


Figure 33: Space-time diagram of simulation of Fig. 1a.

Figure 34 provides different test and check failures before resolving the correct macro-collisions as well as a 3 macro-signal collision.

Figure 35 represents a space-time diagram with finitely many collisions with no special regularity.

Figure 36 represents a space-time diagram with a simple accumulation on top and its simulation.

Figure 36 is the basis for firing squad synchronisation on Cellular Automaton. On signal machines, since space and time are continuous, it generates a fractal. The simulation contains more than 100,000 signals. It has been used as a test for robustness.

7 Conclusion

Altogether, the construction proves the following result.

Theorem 11. *For any finite set of real numbers $\mathcal{S}$, there is an $\mathcal{S}$-universal signal machine. The set of $\mathcal{U}_{\mathcal{S}}$ where $\mathcal{S}$ ranges over finite sets of real numbers is an intrinsically universal family of signal machines.*

With the definition of simulation provided in this paper, there does not exist any intrinsically universal signal machine. Indeed, for a signal machine to be simulated, there should be a simulating signal exactly

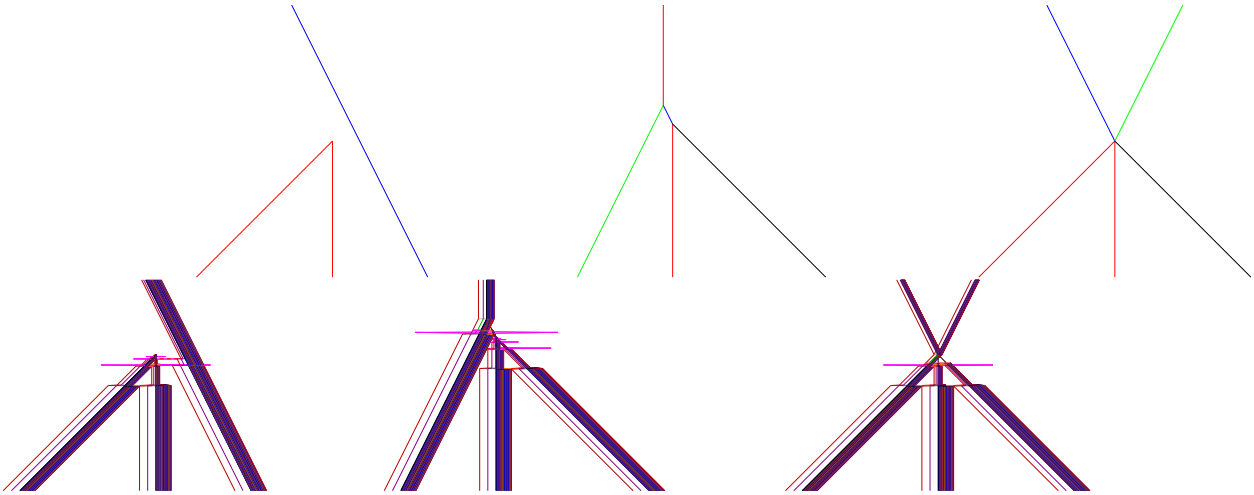


Figure 34: Example 1.

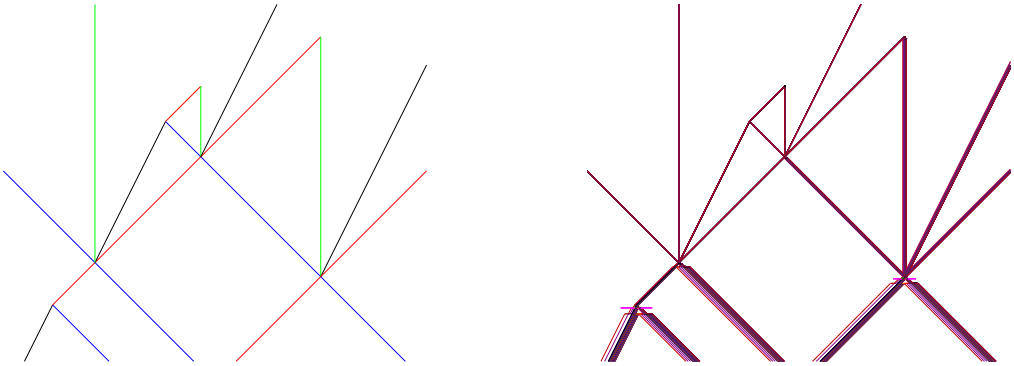


Figure 35: Example 2.

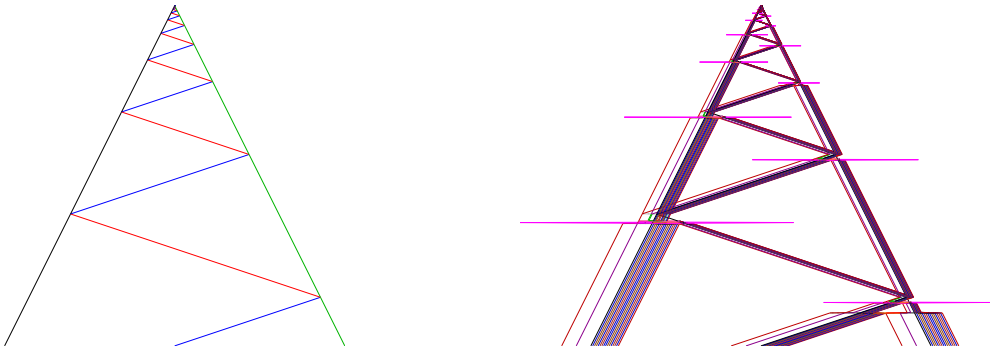


Figure 36: Example 3.

where a simulated one is: it must have the same speed. Thus, to be able to simulate an arbitrary signal machine, a universal signal machine should have signals of each of infinitely many speeds. Since every signal machine has finitely many speeds, there is no intrinsically universal signal machine. On the other hand, it might work with some other reasonable definition of simulation, maybe considering some kind of approximation.

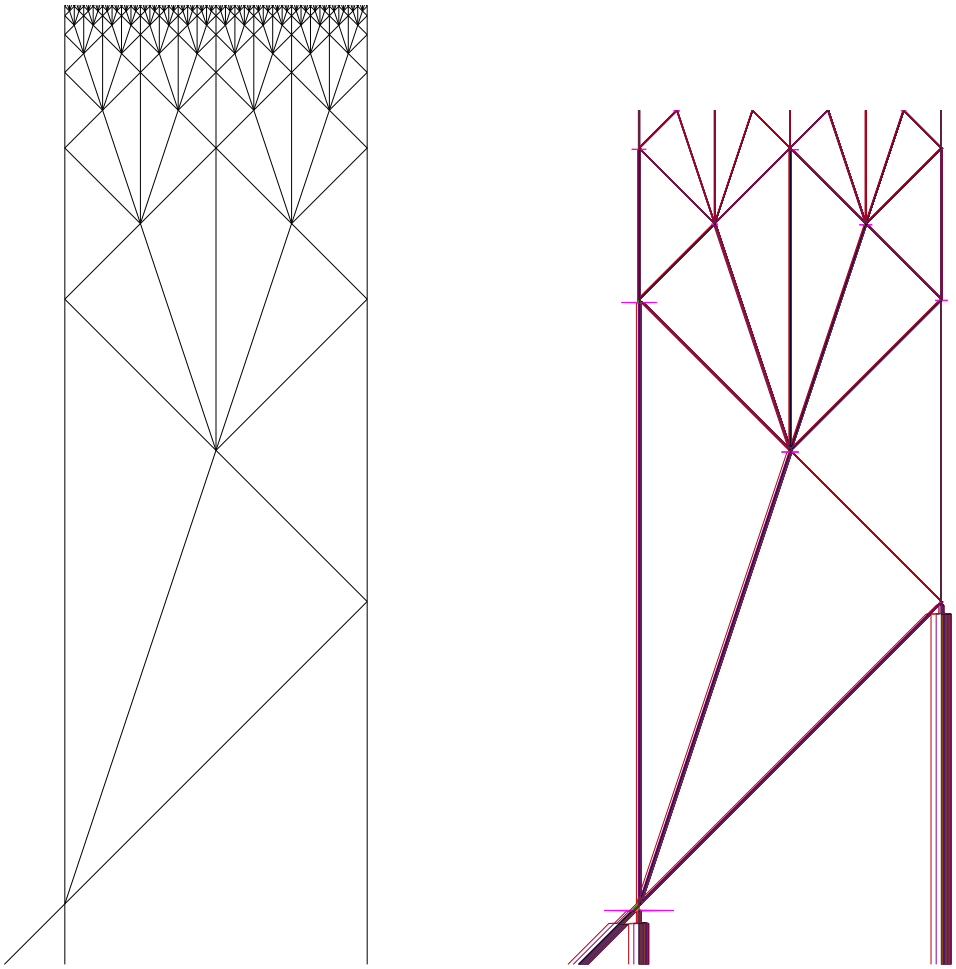


Figure 37: Example 4.

A signal machine may produce accumulations (infinitely many collisions in a bounded part of the space-time). The simulation works up to the first accumulation (excluded) where the configuration of the simulated machine ceases to be defined in Figure 36.

Using macro-signals forces one to deal with width. Hopefully, macro-signals can be made as thin as needed. Nevertheless, as seen through the paper, it requires a lot of technicalities to deal with that.

In the construction, each macro-signals carries the list of rules associates with it, thus its dynamics, like a cell carries its DNA. We wonder what might happen and what kind of artefact could be created if some way to dynamically modify the table were introduced.

References

Andrew Adamatzky, editor. *Collision based computing*. Springer, 2002.

Jürgen Albert and Karel Čulik II. A simple universal cellular automaton and its one-way and totalistic version. *Complex Systems*, 1:1–16, 1987.

Pablo Arrigh and Jonathan Grattage. Partitioned quantum cellular automata are intrinsically universal. *Natural Computing*, 11(1):13–22, 2012. doi: 10.1007/s11047-011-9277-6.

Lenore Blum, Michael Shub, and Steve Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bulletin of the American Mathematical Society*, 21 (1):1–46, 1989.

Nino Boccara, J. Nasser, and Michel Roger. Particle-like structures and interactions in spatio-temporal patterns generated by one-dimensional deterministic cellular automaton rules. *Physical Review A*, 44(2):866–875, 1991.

Matthew Cook. Universality in elementary cellular automata. *Complex Systems*, 15:1–40, 2004.

Marianne Delorme and Jacques Mazoyer. Signals on cellular automata. In Andrew Adamatzky, editor, *Collision-based computing*, pages 234–275. Springer, 2002.

David Doty, Jack H. Lutz, Matthew J. Patitz, Scott M. Summers, and Damien Woods. Intrinsic universality in self-assembly. In Jean-Yves Marion and Thomas Schwentick, editors, *27th Int. Symposium on Theoretical Aspects of Computer Science, (STACS 2010), Nancy, France*, volume 5 of *LIPIcs*, pages 275–286. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2010. doi: 10.4230/LIPIcs.STACS.2010.2461.

David Doty, Jack H. Lutz, Matthew J. Patitz, Robert T. Schweller, Scott M. Summers, and Damien Woods. The tile assembly model is intrinsically universal. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October, 2012*, pages 302–310. IEEE Computer Society, 2012. ISBN 978-1-4673-4383-1. doi: 10.1109/FOCS.2012.76.

Jérôme Durand-Lose. Reversible cellular automaton able to simulate any other reversible one using partitioning automata. In *LATIN 1995*, number 911 in LNCS, pages 230–244. Springer, 1995. doi: 10.1007/3-540-59175-3_92.

Jérôme Durand-Lose. Abstract geometrical computation 1: embedding black hole computations with rational numbers. *Fundamenta Informaticae*, 74(4):491–510, 2006. URL `https://content.iospress.com/articles/fundamenta-informaticae/fi74-4-07`.

Jérôme Durand-Lose. Abstract geometrical computation and the linear Blum, Shub and Smale model. In Barry S. Cooper, Benedikt. Löwe, and Andrea Sorbi, editors, *Computation and Logic in the Real World, 3rd Conf. Computability in Europe (CiE 2007)*, number 4497 in LNCS, pages 238–247. Springer, 2007. doi: 10.1007/978-3-540-73001-9_25.

Jérôme Durand-Lose. The signal point of view: from cellular automata to signal machines. In Bruno Durand, editor, *Journées Automates cellulaires (JAC 2008)*, pages 238–249, 2008.

Jérôme Durand-Lose. Abstract geometrical computation 3: black holes for classical and analog computing. *Natural Computing*, 8(3):455–472, 2009. doi: 10.1007/s11047-009-9117-0.

Jérôme Durand-Lose. Abstract geometrical computation 6: a reversible, conservative and rational based model for black hole computation. *International Journal of Unconventional Computing*, 8 (1):33–46, 2012. URL `http://www.oldcitypublishing.com/journals/ijuc-home/ijuc-issue-contents/ijuc-volume-8-number-1-2012/ijuc-8-1-p-33-46/`.

Eric Goles Ch., Pierre-Etienne Meunier, Ivan Rapaport, and Guillaume Theyssier. Communication complexity and intrinsic universality in cellular automata. *Theoretical Computer Science*, 412(1-2):2–21, 2011. doi: 10.1016/j.tcs.2010.10.005.

Wim Hordijk, James P. Crutchfield, and Melanie Mitchell. Mechanisms of emergent computation in cellular automata. In A. E. Eiben, Thomas Bäck, Marc Schoenauer, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature - PPSN V, 5th Int. Conf., Amsterdam, The Netherlands*, volume 1498 of *LNCS*, pages 613–622. Springer, 1998. doi: 10.1007/BFb0056903.

Mariusz H. Jakubowski, Kenneth Steiglitz, and Richard K. Squier. When can solitons compute? *Complex Systems*, 10(1):1–21, 1996.

Mariusz H. Jakubowski, Kenneth Steiglitz, and Richard K. Squier. Information transfer between solitary waves in the saturable Schrödinger equation. In *Proceedings from the International Conference on Complex Systems on Unifying Themes in Complex Systems*, pages 281–293, Cambridge, MA, USA, 2000. Perseus Books. ISBN 0-7382-0049-2. URL `http://dl.acm.org/citation.cfm?id=331767.331919`.

Mariusz H. Jakubowski, Kenneth Steiglitz, and Richard K. Squier. Computing with classical soliton collisions. In Andrew Adamatzky, editor, *Advances in Unconventional Computing vol. 2: Prototypes, Models and Algorithms*, volume 23 of *Emergence, Complexity and Computation*, pages 261–295. Springer, 2017. doi: 10.1007/978-3-319-33921-4_12.

Weifeng Jin and Fangyue Chen. Symbolic dynamics of glider guns for some one-dimensional cellular automata. *Nonlinear Dynamics*, 86(2):941–952, Oct 2016. ISSN 1573-269X. doi: 10.1007/s11071-016-2935-6.

Kristian Lindgren and Mats G. Nordahl. Universal computation in simple one-dimensional cellular automata. *Complex Systems*, 4:299–318, 1990.

Simon Martiel and Bruno Martin. An intrinsically universal family of causal graph dynamics. In Jérôme Durand-Lose and Benedek Nagy, editors, *Machines, Computations, and Universality - 7th Int. Conf., Famagusta, North Cyprus, (MCU 2015)*, volume 9288 of *LNCS*, pages 129–148. Springer, 2015. doi: 10.1007/978-3-319-23111-2_9.

Jacques Mazoyer and Ivan Rapaport. Inducing an order on cellular automata by a grouping operation. In *15th Annual Symposium on Theoretical Aspects of Computer Science (STACS 1998)*, number 1373 in LNCS, pages 116–127. Springer, 1998.

Jacques Mazoyer and Véronique Terrier. Signals in one-dimensional cellular automata. *Theoretical Computer Science*, 217(1):53–80, 1999. doi: 10.1016/S0304-3975(98)00150-9.

Pierre-Etienne Meunier, Matthew J. Patitz, Scott M. Summers, Guillaume Theyssier, Andrew Winslow, and Damien Woods. Intrinsic universality in tile self-assembly requires cooperation. In Chandra Chekuri, editor, *25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2014), Portland, Oregon, USA*, pages 752–771. SIAM, 2014. doi: 10.1137/1.9781611973402.56.

Melanie Mitchell. Computation in cellular automata: a selected review. In T. Gramss, S. Bornholdt, M. Gross, M. Mitchell, and T. Pellizzari, editors, *Nonstandard Computation*, pages 95–140. Weinheim: VCH Verlagsgesellschaft, 1996.

Nicolas Ollinger. Two-states bilinear intrinsically universal cellular automata. In *Fundamentals of Computation Theory, 13th International Symposium (FCT 2001)*, number 2138 in LNCS, pages 369–399. Springer, 2001.

Nicolas Ollinger. The intrinsic universality problem of one-dimensional cellular automata. In *20th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2003)*, number 2607 in LNCS, pages 632–641. Springer, 2003.

Pawel Siwak. Soliton-like dynamics of filtrons of cycle automata. *Inverse Problems*, 17:897–918, 2001.

Victor I. Varshavsky, Vyacheslav B. Marakhovsky, and V. A. Peschansky. Synchronization of interacting automata. *Mathematical System Theory*, 4(3):212–230, 1970.

Damien Woods. Intrinsic universality and the computational power of self-assembly. In Turlough Neary and Matthew Cook, editors, *Proceedings Machines, Computations and Universality 2013, MCU 2013, Zürich, Switzerland*, volume 128 of *EPTCS*, pages 16–22, 2013. doi: 10.4204/EPTCS.128.5.

Jean-Baptiste Yunès. Simple new algorithms which solve the firing squad synchronization problem: a 7-states 4n-steps solution. In Jérôme Durand-Lose and Maurice Margenstern, editors, *Machine, Computations and Universality (MCU 2007)*, number 4664 in LNCS, pages 316–324. Springer, 2007.

Extra material

An archive is available with examples, simulation source and java programs to run it at
http://www.univ-orleans.fr/lifo/Members/Jerome.Durand-Lose/Recherche/AGC_Intrinsic_Univ_SM__FILES.tgz
The jar file needs java version 11 or higher to run.

Contents

Figure 37 represents the same initial configuration as Fig. 35, but the speed are 5 times faster. This is used to check that large speeds are handled correctly.

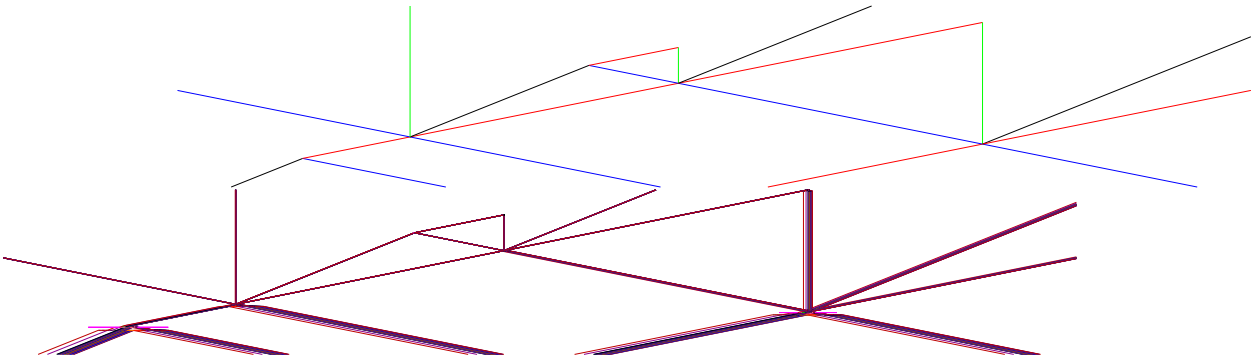


Figure 38: Example 5.

A Table of used symbols

Table of symbols

Symbol	Definition	Page
$\mathcal{A}$	Signal machine	1
$\mathcal{B}$	Signal machine, other	1
$\mathcal{S}$	(finite) Set of speeds	1
μ	Some meta-signal	2
M	Signal machine meta-signal set	4
S	Signal machine speed function	4
R	Set of rules of a signal machine	4
$\mathbb{R}$	Set of all real numbers	4
ρ^-	Incoming meta-signals of a rule	4
ρ	Some rule of a signal machine	4
ρ^+	Outgoing meta-signals of a rule	4
V	AGC set of extended values	4
$\oslash$	Void (AGC spc value)	4
c	Configuration of some signal machine	4
s	Speed (usually with a subscript like s_1)	4
x	Some spatial coordinate	4
Δt	Some duration	4
$\propto$	Meta-signal present or outgoing from a collision/rule	5
Δ	Time to next collision	5
x	Some spatial position	5
t	Some temporal position	5
t'	Some temporal position	5
$\mathbb{A}$	Space-time diagram for machine $\mathcal{A}$	5
$\mathcal{C}$	The set of configurations of some signal machine	5
$\mathcal{C}^*$	The set of all configurations of $\mathcal{A}$ with $c(0) \neq \oslash$	5
symb	Function yielding the symbolic representation of a configuration	5
$\mathbb{Z}$	Set of all integers	5
$\perp$	Symbol used to complete the definition of the bi-infinite word associated to a configuration and a position	5
$\mathbb{N}$	Set of all natural integers	5
interp	Local function used for decoding signals in simulation	6
b	Some configuration of $\mathcal{B}$	6
interp^*	Local function used for decoding whole configurations in simulation	6
repr	Function used to find a representative of a signal, i.e. encoding it	6
a	Some configuration of $\mathcal{A}$	6
repr^*	Extension of repr to whole configurations	6
$\mathbb{B}$	Space-time diagram for machine $\mathcal{B}$	6
$\mathcal{U}_\mathcal{S}$	Signal machine capable of simulating all signal machines using only speed in $\mathcal{S}$	7
M_U	(finite) Set of meta-signals used for simulating all machines with a given speed set	7
$\mathcal{S}_\mathcal{U}$	(finite) Set of speeds used for simulating all machines with a given speed set	7
R_U	(finite) Set of collision-rules used for simulating all machines with a given speed set	7
$\mathfrak{n}$	Number of speeds in $\mathcal{S}$	7
${}_i\mu^\sigma$	Meta-signal σ of speed s_i of $\mathcal{A}$	8
σ	Index for meta-signal of $\mathcal{A}$	8
s^{max}	Maximum absolute value of speed	8
s^{rapid}	Base speed for Check signals	8
δ	Maximum width of macro-signals entering a checked macro-collision	8
ρ_{max}	Collision with an input signal of each speed, each with maximal id.	8
$\mathfrak{w}$	Maximum of signals in any value of $\mathsf{repr}(\cdot)$	8
E	Some subset of $1..\mathfrak{n}$	9

Symbol	Definition	Page
width	Initial width of a macro-signal	9
$\mathcal{U}_S^{\mathsf{checked}}$	Submachine of $\mathcal{U}_S$ for dealing with checked configurations	10
δ_o	Delay between a (macro-)collision and the time when all its outputs are ready and clean	10
m	Index for speeds	11
F	Some (other) subset of $1..\mathtt{n}$	14
$\mathcal{U}_S{}'$	Some variant of $\mathcal{U}_S$	15
C_m	Point of intersection of macro-collision on output	18
Z_T	Point on top of the safety zone	18
Z_L	Point on left of the safety zone	18
Z_R	Point on right of the safety zone	18
Z_B	Point at the bottom of the safety zone	18
s^{shrink}	Base speed for shrinking signals	19
ε	Parameter for ensuring a large enough safety zone	20
$U^{i,k}$	Point used for testing	20
O_L	Left limit of outputting signals	21
O_R	Right limit of outputting signals	21
$s_i^{\mathsf{test\text{-}right\text{-}up}}$	Speed for testing	22
$s_{i,k}^{\mathsf{test\text{-}right}}$	Speed for testing	22
$s_i^{\mathsf{test\text{-}right\text{-}back}}$	Speed for testing	22
$s_i^{\mathsf{test\text{-}left\text{-}back}}$	Speed for testing	22
τ_{check}	[check] Maximal relative height between middle and encounter from start	25
$s^{\mathsf{chk\text{-}up}}$	Speed for checking $\_\mathsf{main}^{\emptyset}$ positions	25

bibtex entry

```
@article{becker+besson+durand-lose+emmanuel+foroughmand-araabi+goliaei+heydarshahi21,
  doi = {10.1145/3442359},
  arxiv = {https://arxiv.org/abs/1804.09018},
  author = {Becker, Florent and
                  Besson, Tom and
                  Durand-Lose, J{\'e}r{\^o}me and
                  Emmanuel, Aur{\'e}lien and
                  Foroughmand-Araabi, Mohammad-Hadi and
                  Goliaei, Sama and
                  Heydarshahi, Shahrzad},
  title = {Abstract Geometrical Computation 10: An Intrinsically Universal Family of Signal Machines},
  journal = {ACM Trans. Comput. Theory},
  year = {2021},
  volume = {13},
  number = {1},
  pages = {1--31},
  note = {arXiv 1804.09018},
  language = {english}
}
```