Machines à signaux :
origine, puissance, retour au « raisonnable »

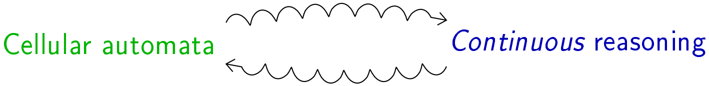
Jérôme Durand-Lose, Vincent Levorato et Maxime Senot

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

LRI, Orsay — 3 mai 2011

Motivation

Discrete space and time

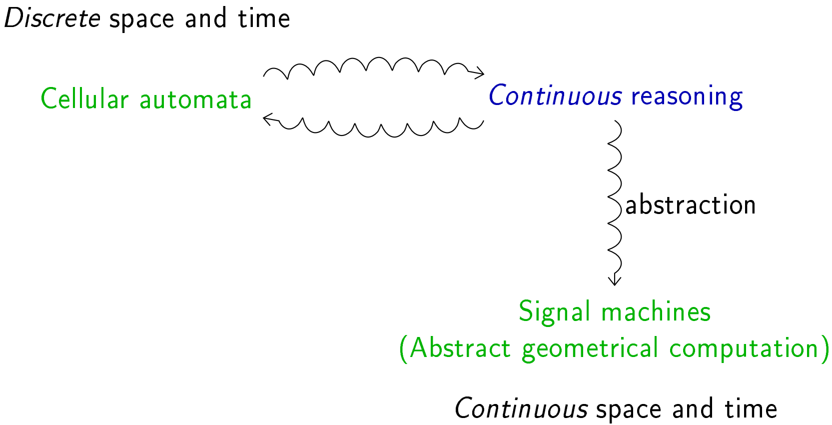


Cellular automata

Continuous reasoning

Continuous space and time

Motivation



Discrete space and time

Cellular automata

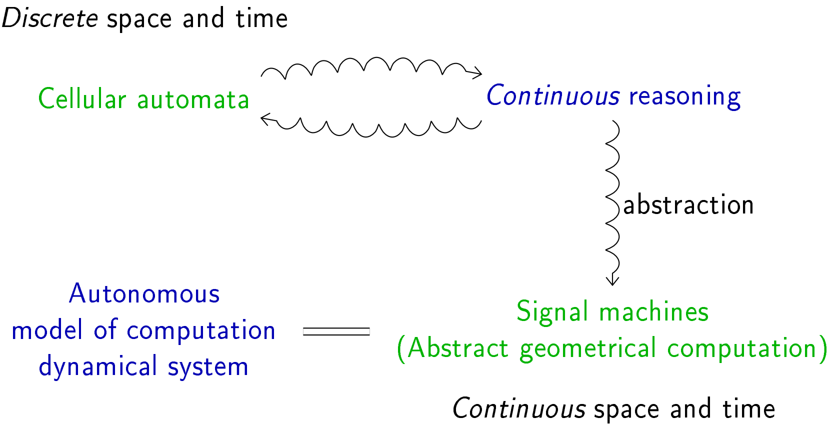
Continuous reasoning

abstraction

Signal machines
(Abstract geometrical computation)

Continuous space and time

Motivation



Discrete space and time

Cellular automata

Continuous reasoning

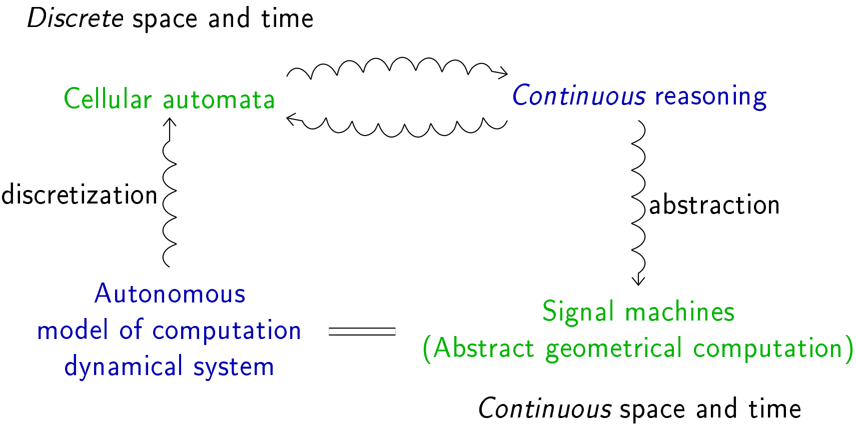
abstraction

Autonomous
model of computation
dynamical system

Signal machines
(Abstract geometrical computation)

Continuous space and time

Motivation



Discrete space and time

Cellular automata

Continuous reasoning

discretization

abstraction

Autonomous
model of computation
dynamical system

Signal machines
(Abstract geometrical computation)

Continuous space and time

1. Cellular automata to signal machines
2. Generic QSat solving in constant space and duration (with D. Duchier)
3. Discretization into CA
4. Conclusion

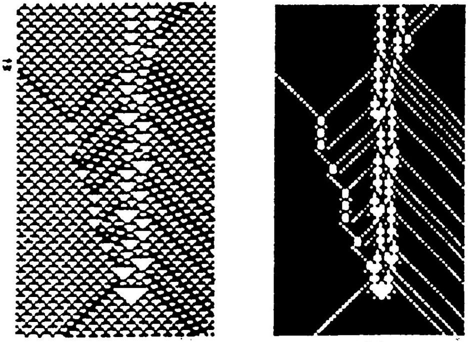
1 Cellular automata to signal machines

2 Generic QSat solving in constant space and duration (with D. Duchier)

3 Discretization into CA

4 Conclusion

Analyzing CA with signals

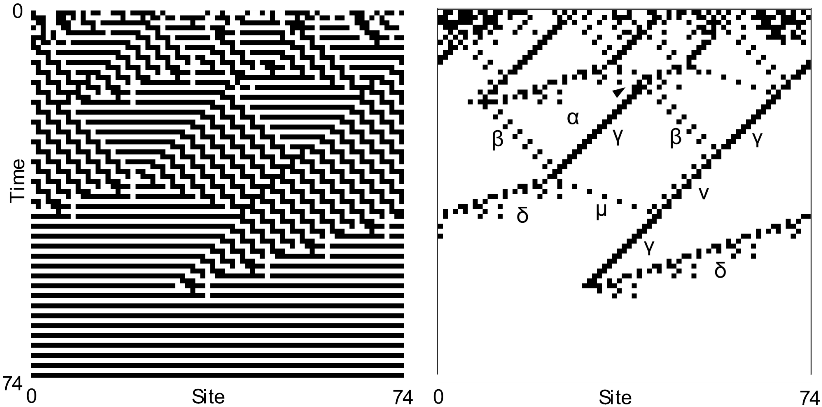


(a) (b)

FIG. 7. Rule 54. (a) Annihilation of the radiating particle. (b) The same as (a) with the mapping defined in Fig. 6.

[Boccara et al., 1991, Fig. 7]

Analyzing CA with signals



(a) Space-time diagram. (b) Filtered space-time diagram.

[Das, Crutchfield, Mitchell 95]

Designing CA with signals

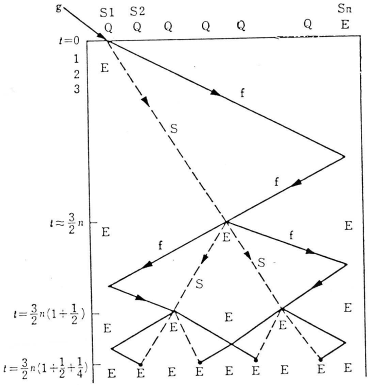
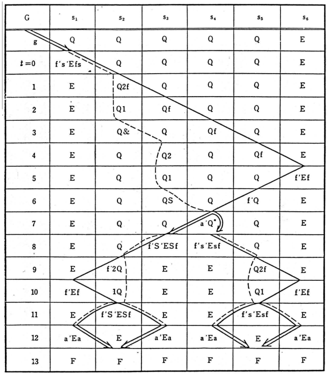


図 3·5 一斉射撃の問題（連続近似）



G	s_1	s_2	s_3	s_4	s_5	s_6
g	Q	Q	Q	Q	Q	E
$t=0$	f's'Efs	Q	Q	Q	Q	E
1	E	Q2f	Q	Q	Q	E
2	E	Q1	Qf	Q	Q	E
3	E	Q&	Q	Qf	Q	E
4	E	Q	Q2	Q	Qf	E
5	E	Q	Q1	Q	Q	f'Ef
6	E	Q	QS	Q	f'Q	E
7	E	Q	Q	a'Q*	Q	E
8	E	Q	f'S'ESf	f's'Esf	Q	E
9	E	f'2Q	E	E	Q2f	E
10	f'Ef	1Q	E	E	Q1	f'Ef
11	E	f'S'ESf	E	E	f's'Esf	E
12	a'Ea	E	a'Ea	a'Ea	E	a'Ea
13	F	F	F	F	F	F

図 3·6 一斉射撃解（$n=6$）

Goto's solution to the Firing Squad Synchronization Problem [Goto66]

Designing CA with signals

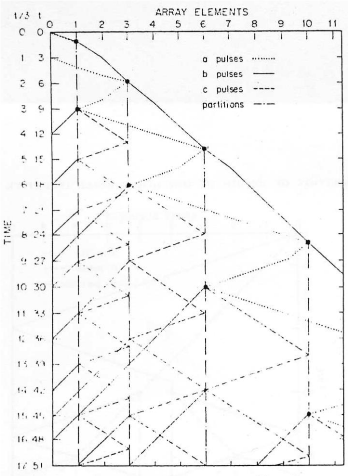


Fig. 2. Solution to the prime problem

Generating primes [Fischer, 1965, Fig. 2]

Designing CA with signals

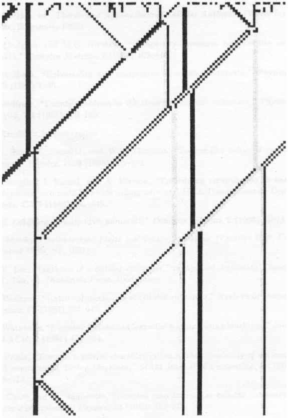


Figure 4: The $k = 4$, $r = 2$ universal cellular automaton of table 4 simulated starting from a random initial state. The symbols 0, 1, ␣, and + are represented by ▨▨□■

Computing by simulating a Turing machine
[Lindgren and Nordahl, 1990, Fig. 4]

A whole programming system with discrete signals

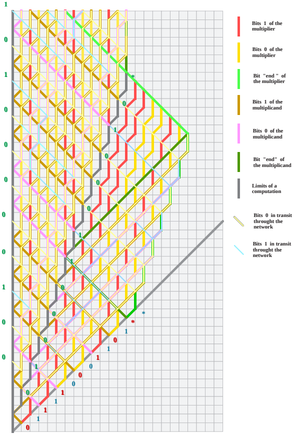


Figure 8: Computing $(ab)^2$.

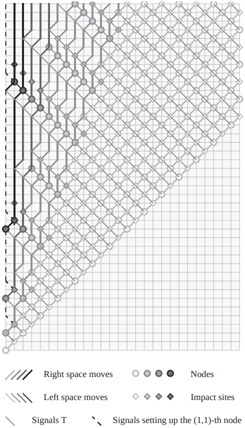


Figure 9: Setting up an infinite family of regular safe grids (the darkness of the grid indicates its rank).

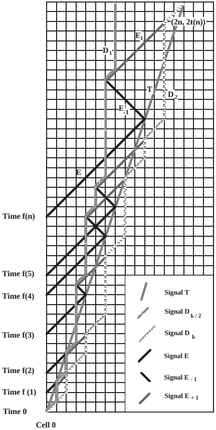
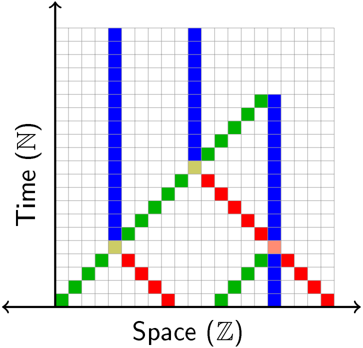
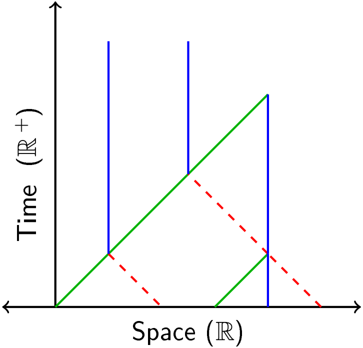


Figure 18: Characterization of the sites $(n, f(n))$.

[Mazoyer, 1996, Fig. 8 and 19] and [Mazoyer and Terrier, 1999, Fig. 18]





Vocabulary

- Signal (meta-signal)
- Collision (rule)

Example: finding the middle

Meta-signal, speed

M, $S(\mathrm{M}) = 0$

Collision rules

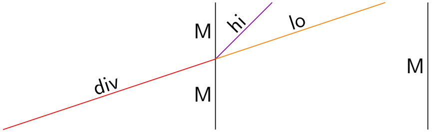
Example: finding the middle

Meta-signal, speed

M, $S(\mathrm{M}) = 0$
div, $S(\mathrm{div}) = 3$

Collision rules

Example: finding the middle



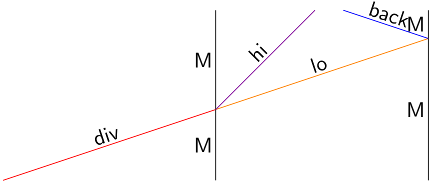
Meta-signal, speed

M, $S(\mathrm{M}) = 0$
div, $S(\mathrm{div}) = 3$
hi, $S(\mathrm{hi}) = 1$
lo, $S(\mathrm{lo}) = 3$

Collision rules

$\{\ \mathrm{div},\ \mathrm{M}\ \} \rightarrow \{\ \mathrm{M},\ \mathrm{hi},\ \mathrm{lo}\ \}$

Example: finding the middle



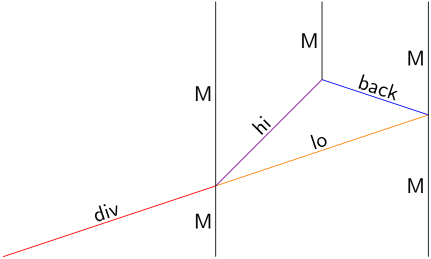
Meta-signal, speed

M, $S(\mathrm{M}) = 0$
div, $S(\mathrm{div}) = 3$
hi, $S(\mathrm{hi}) = 1$
lo, $S(\mathrm{lo}) = 3$
back, $S(\mathrm{back}) = -3$

Collision rules

$\{\ \mathrm{div},\ \mathrm{M}\ \} \rightarrow \{\ \mathrm{M},\ \mathrm{hi},\ \mathrm{lo}\ \}$
$\{\ \mathrm{lo},\ \mathrm{M}\ \} \rightarrow \{\ \mathrm{back},\ \mathrm{M}\ \}$

Example: finding the middle



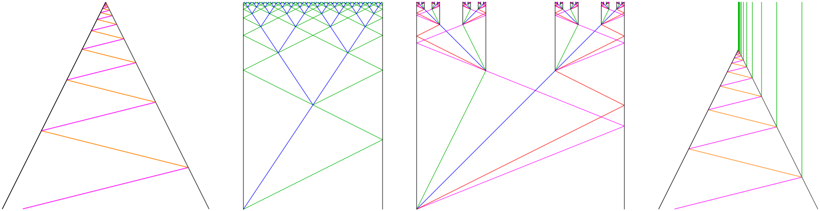
Meta-signal, speed

M, $S(\text{M}) = 0$
div, $S(\text{div}) = 3$
hi, $S(\text{hi}) = 1$
lo, $S(\text{lo}) = 3$
back, $S(\text{back}) = -3$

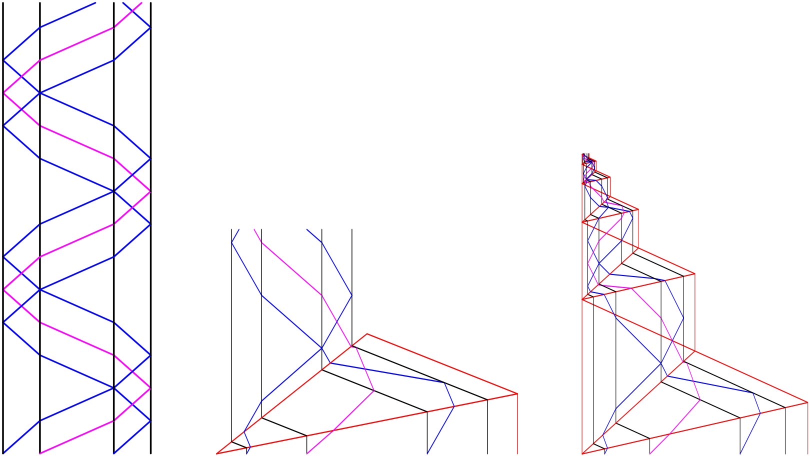
Collision rules

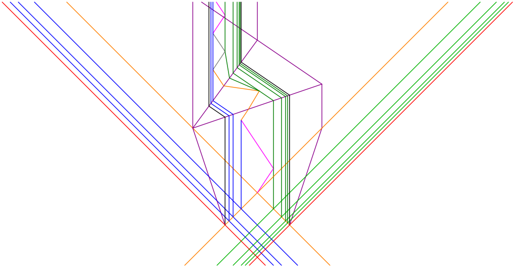
$\{ \text{div, M} \} \rightarrow \{ \text{M, hi, lo} \}$
$\{ \text{lo, M} \} \rightarrow \{ \text{back, M} \}$
$\{ \text{hi, back} \} \rightarrow \{ \text{M} \}$

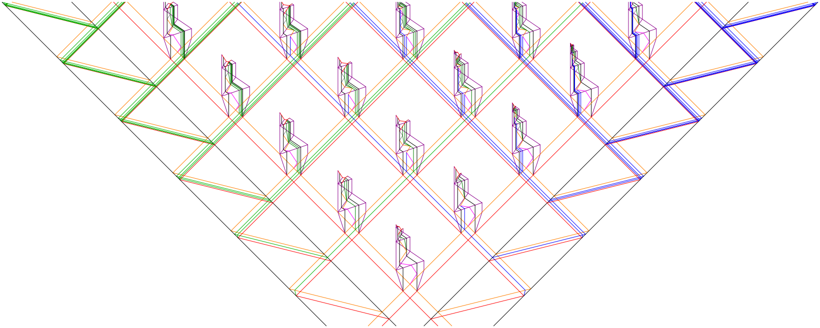
New kinds of *monsters*



Scaling down and bounding the duration







1. Cellular automata to signal machines
2. Generic QSat solving in constant space and duration (with D. Duchier)
3. Discretization into CA
4. Conclusion

QSat: quantified satisfaction problem

- Quantified boolean formula (without free variable)
- Find its logical value
- PSPACE-complete problem

Running example

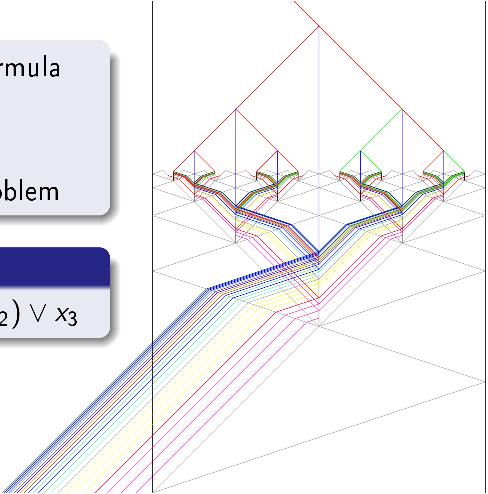
$$\phi = \exists x_1 \forall x_2 \forall x_3 \quad (x_1 \wedge \neg x_2) \vee x_3$$

QSat: quantified satisfaction problem

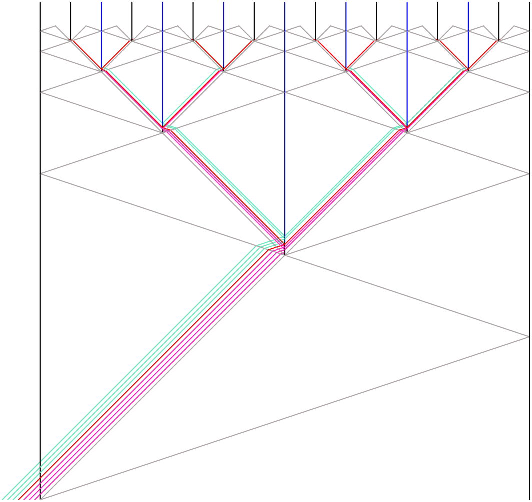
- Quantified boolean formula (without free variable)
- Find its logical value
- PSPACE-complete problem

Running example

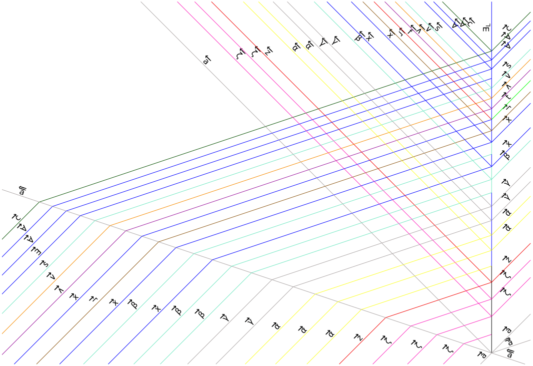
$$\phi = \exists x_1 \forall x_2 \forall x_3 \quad (x_1 \land \neg x_2) \lor x_3$$



Building the tree / combinatorial comb



Lens and variables assignment

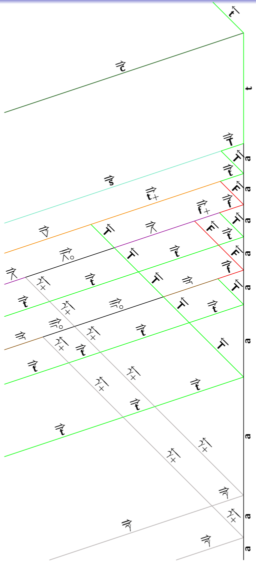


Formula evaluation

$$\phi = \exists x_1 \forall x_2 \forall x_3 \quad (x_1 \land \neg x_2) \lor x_3$$

Case here

(true $\land$ ¬true) $\lor$ true



Complexities

Space

- constant (as a width)
- exponential (as max number of signals)

Time

- constant (as a duration)
- cubic (as max length of collision chain)

Complexities

Space

- constant (as a width)
- exponential (as max number of signals)

Time

- constant (as a duration)
- cubic (as max length of collision chain)

NB: Super-Turing Model with accumulations

- Decide Halt in finite duration and width...

1. Cellular automata to signal machines
2. Generic QSat solving in constant space and duration (with D. Duchier)
3. Discretization into CA
4. Conclusion

Returning to a “reasonnable” model of computation

Natural candidate

Cellular automata

Limits

No more than a *crude approximation*

- No unbounded computation in finite space and time
- No accumulation

Goal

- Keep as much as possible the signal geometry
- Transfer of properties
- (finding the middle is already a problem)

Considering pretopology

- Based on pseudo-closure
  ($A \subseteq a(A)$ and no idempotency requirement)
- Use to relate geometrical properties between continuous and discrete images
- A signal is considered as the "trace" of the dilatation of a set (orle: $o(A) = a(A) - A$)

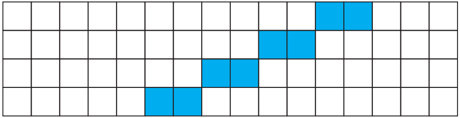
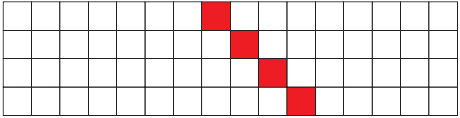
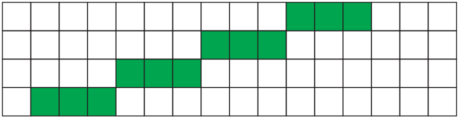
Goal

Generate a *discrete signal machine*
which is nothing but a special case of CA

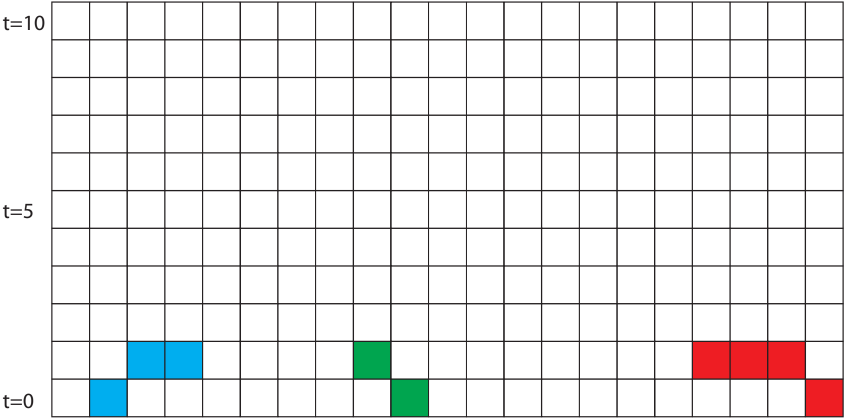
Experimental prototype

Simulation sotware in JAVA with the
PretopoLib [LevoratoBui08] library

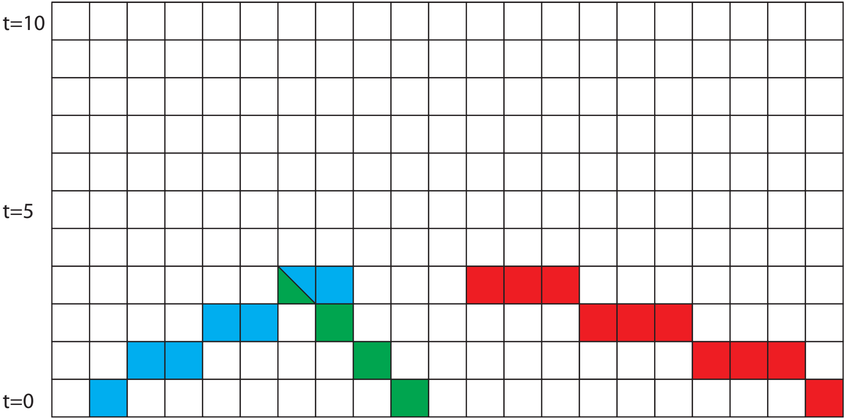
Illustration of meta-signals

Meta-Signal	Base of neighborhood	Meta-Signal movement (successive orles)
(blue, 2)	B(x)= {☐☐▨} or B(x)={-2, -1, 0}	
(red, -1)	B(x)= {▨☐} or B(x)={0,1}	
(green, 3)	B(x)= {☐☐☐▨} or B(x)={-3, -2, -1, 0}	

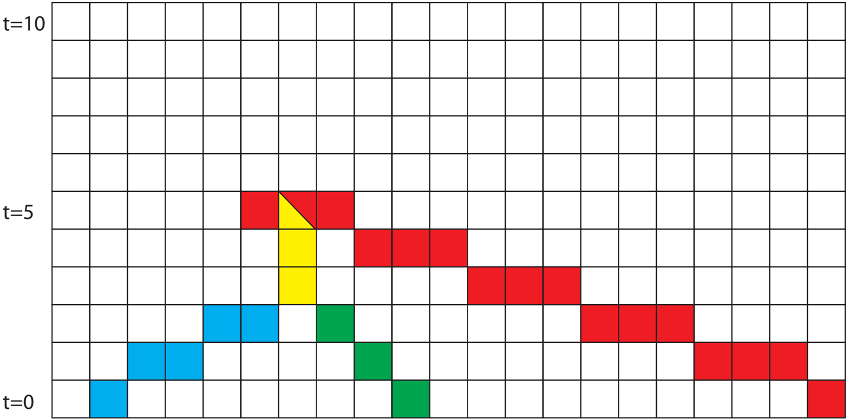
Example



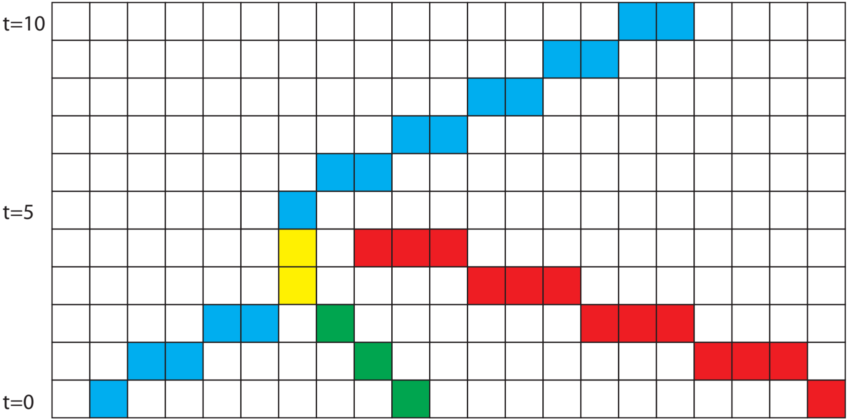
Example



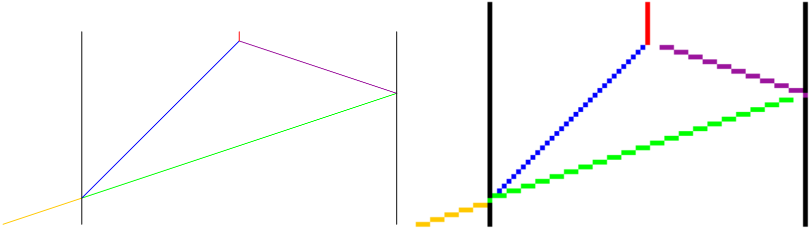
Example



Example



Test: middle geometrical algorithm



(a) Continuous Signal Machine (b) Discrete Signal Machine

1. Cellular automata to signal machines
2. Generic QSat solving in constant space and duration (with D. Duchier)
3. Discretization into CA
4. Conclusion

Future work

Fractal computation

- modularity
- fractal characterization
- higher classes of complexity

Automatic discretization

- simple translation of discrete signal machine as CA
- accuracy / approximation characterization
- robustness
- develop a logic to express preserved properties