Computability in Multidimensional Symbolic Dynamics

E. Jeandel

LORIA (Nancy, France)

Multidimensional symbolic dynamics

Multidimensional Symbolic Dynamics is the extension to $\mathbb{Z}^2$ and $\mathbb{Z}^n$ of (classical) symbolic dynamics.
We are still interested in shifts of finite type, sofic shifts, and factor maps (among others).

Dimension 1

- ***Subshifts*** are sets of (biinfinite) words over an alphabet Σ that are (topologically) closed and shift-invariant.
- A subshift S can be given by a set $\mathcal{F}$ of forbidden ***patterns*** (words)

$$S = \{\text{words with no cubes}\} = X_{\mathcal{F}}$$

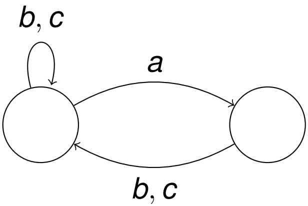
  where $\mathcal{F} = \{uuu, u \in \Sigma^+\}$. S contains e.g. the Thue-Morse word
- Of great interest are ***shifts of finite type*** (SFT), that can be given with a finite $\mathcal{F}$, and *sofic shifts* that are recolorings of SFTs.

In dimension 1

Many answers can be given for subshifts of finite type (SFT) by way of *automata theory*

- (biinfinite) words in a SFT (more generally a sofic shift) are linked to **biinfinite paths** in some finite graph (automaton).

$$\mathcal{F} = \{aa\}$$



Seeing (sofic) subshifts as paths on a automata brings automata theory, graph theory and linear algebra to the rescue.

In higher dimensions

- Subshifts are sets of (biinfinite) pictures over an alphabet Σ that are (topologically) closed and shift-invariant.
- A subshift S can be given by a set $\mathcal{F}$ of forbidden patterns

$$S = \{\text{images where all lignes are identical}\} = X_{\mathcal{F}}$$

where $\mathcal{F} = \left\{ \begin{array}{c} x \\ y \end{array}, x \neq y, x, y \in \Sigma \right\}$.

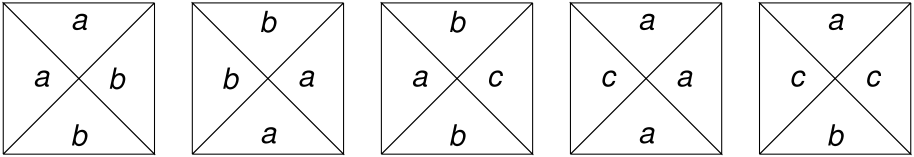
- Of great interest are shifts of finite type (SFT), that can be given with a finite $\mathcal{F}$.

What about automata theory ?

Automata theory in dimension 2

There is no (useful) automata theory in dimension 2.

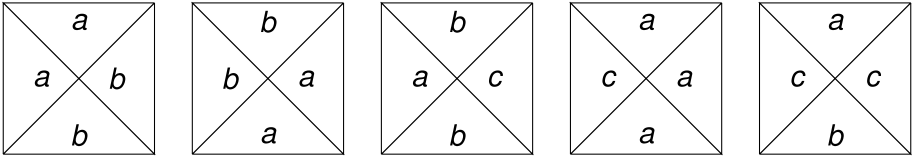
The closest we get is Wang tiles :



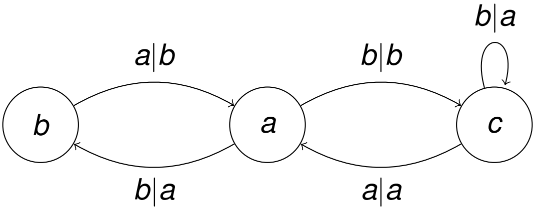
Contiguous tiles must agree on their common edge.
Every SFT is equivalent (upto conjugacy) to some set of Wang tiles.

There is no (useful) automata theory in dimension 2.

The closest we get is Wang tiles :



(letter-to-letter) Transducers :



Automata theory in dimension 2

- SFTs in dimension 2 are equivalent to biinfinite iterations of transducers on biinfinite words
- We can recover a bit of automata theory…
- … of graph theory (Nasu 95) or linear algebra (Markley-Paul 81, Schraudner 08)
- But things remain fundamentally different.

Why ?

Undecidability

The theory of multidimensional symbolic dynamics is filled with undecidable problems

- Berger [Ber64] : There is no algorithm to decide if a SFT is empty
- Robinson [Rob71] : For a fixed SFT, there is no algorithm to decide if a pattern is globally admissible (can be extended)
- Gurevich-Koryakov [GK72] : There is no algorithm to decide if a SFT has periodic points.

D. Lind calls it “The swamp of undecidability : It’s a place you don’t want to go.”.

This talk

A recent trend suggest that computability should be seen not as an hindrance, but as the good way to understand multidimensional symbolic dynamics.

In this talk :

- Why computability comes into play
- What computability has to say

Plan

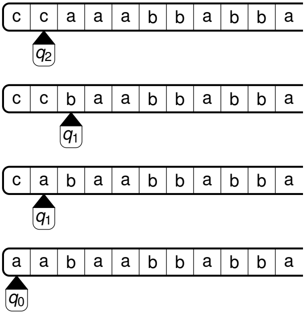
1. Why
2. What

Computations in 2D

It is very easy to encode computations in 2D SFTs

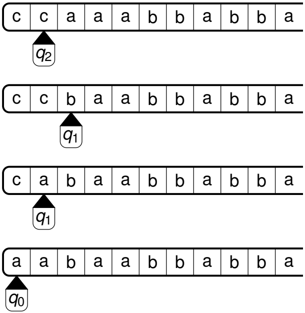
- Clear from the transducer approach
- Or by encoding Turing machines directly.

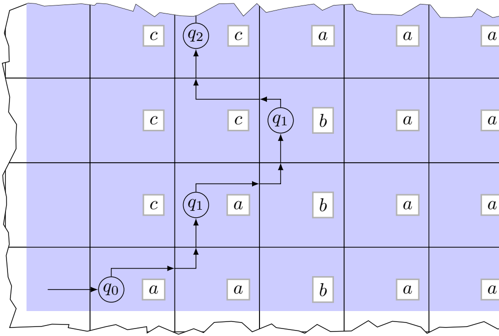
The proof



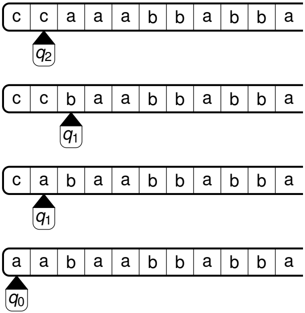
c c a a a b b a b b a
q_2
c c b a a b b a b b a
q_1
c a b a a b b a b b a
q_1
a a b a a b b a b b a
q_0

The proof

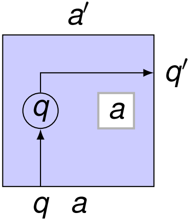

c c a a a b b a b b a
q_2
c c b a a b b a b b a
q_1
c a b a a b b a b b a
q_1
a a b a a b b a b b a
q_0


c q_2 c a a a
c c q_1 b a a
c q_1 a b a a
q_0 a a b a a

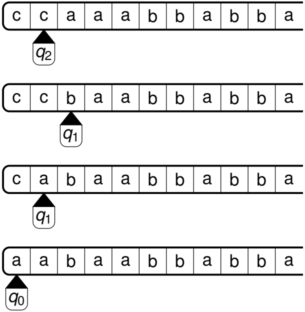
The proof



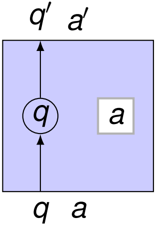
$$(q, a) \longrightarrow (q', a', \rightarrow)$$



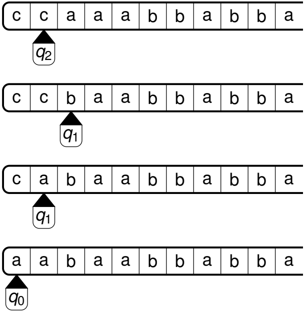
The proof



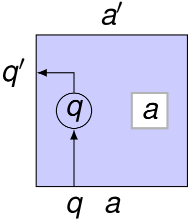
$$(q, a) \longrightarrow (q', a', \uparrow)$$



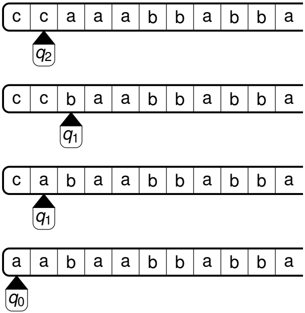
The proof

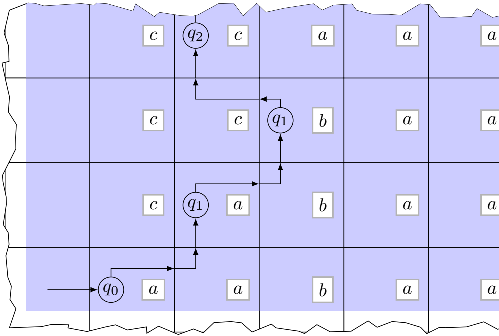


$$(q, a) \longrightarrow (q', a', \leftarrow)$$



The proof


c c a a a b b a b b a
q2
c c b a a b b a b b a
q1
c a b a a b b a b b a
q1
a a b a a b b a b b a
q0


c q2 c a a a
c c q1 b a a
c q1 a b a a
q0 a a b a a

Topological considerations

Computability is nothing more than effective continuity.

- Every computable function on a Cantor space is continuous
- Every continuous function on a Cantor space is computable (relative to some oracle) (with a little help)

Plan

1. Why

2. What

Topological conjugacy and invariants

Recall from symbolic dynamics that two SFTs S_1 and S_2 are conjugate if there are isomorphic, i.e. there exists a reversible block map between S_1 and S_2.

- Conjugacy is of course undecidable in dimension 2 : S is conjugate to $\emptyset$ if and only if S is empty.
  (More on this in Pascal's talk)

It turns out that many conjugacy invariants can be expressed with the vocabulary from computability theory.

Topological entropy

Entropy measures the growth of the number of valid patterns.
Let $p_n(S)$ be the number of $n \times n$ patterns that appear in some point of the subshift S

$$H(S) = \lim_n \frac{\log p_n(S)}{n^2}$$

No easy way to compute $H(S)$ because there is no algorithm to decide if a $n \times n$ pattern appear in some configuration or not.

Topological entropy

$p_n(S)$ is not computable, but we can approximate it.

- Let $q_n(S)$ be the number of $n \times n$ patterns that do not contain any forbidden pattern in $\mathcal{F}$.
- $q_n(S)$ is computable and $q_n(S) \geq p_n(S)$.
- It turns out that

$$H(S) = \lim_n \frac{\log q_n(S)}{n^2} = \inf \frac{\log q_n(S)}{n^2}$$

Topological entropy and Computability Theory

From the point of view of computability, this means $H(S)$ cannot be arbitrary, it must be *computable from above*.

Definition

A real number α is right-c.e. (right-computably enumerable) is $\alpha = \inf a_n$ where a_n is a rational computable given n

Theorem (Hochman-Meyerovitch [HM10])

Entropies of (2D) SFTs are exactly right-c.e. nonnegative reals.

Leitmotif : The computability obstruction is the only obstruction.
(More on this in Ronnie's talk)

Sketch of a Sketch of the Proof

How do you prove these kind of statements ?

- Start from $\alpha = \inf a_n$.
- Build a SFT S of zero entropy where some symbol s appears with frequency proportional to α, say $\alpha / \log 2$.
- Inflate the symbol s into 2 symbols s_1, s_2 to obtain entropy α

Periodic points

Another classical invariant is periodic points.

- Let $pe_n(S)$ be the number of periodic points of size n (points that are periodic of period n in all directions)
- $pe_n(S)$ is now computable given S
- In dimension 1, $pe_n(S)$ is a linear recurrent sequence.
- What does computability says in higher dimensions ?

Periodic points

Given n, we can decide if there exists a point of period n with the following algorithm :

- Choose some $n \times n$ pattern
- Verify that you can tile the plan using this pattern

The second task can be done in polynomial time (roughly n^2)

- The first task is exponential if done sequentially
- but polynomial if done nondeterministically

Periodic points

This algorithm is the typical example of a "NP" algorithm. The counting version is called $\sharp\mathrm{P}$.

Theorem (J.-Vanier 2010)

The sets of periodic points of multidimensional SFTs are exactly the unary sets in NP.
The number of periodic points of multidimensional SFTs are exactly the unary functions in $\sharp\mathrm{P}$.

As always, the computability obstruction is the only obstruction.

Sketch of a sketch of a proof : The space-time diagram of a Turing machine that works in time n^d can be embedded into a hypercube of side n in dimension $2d$.

Constructions in general

Both theorems use some specific constructions.

- What constructions can we carry on ?
- Do we have general theorems ?

No characterizations of sofic shifts. The only way to know if a construction might work is to implement it.

- However, there are a few general constructions. . .

The computability obstruction

Given a SFT, we can decide whether a point *x* is valid (whether it contains no forbidden patterns) with the following algorithm :

- Start from $(0, 0)$ and look if some forbidden pattern appear.
- If some does, then halt.
- Otherwise, go to $(1, 0)$.
- Rinse, repeat until the whole of $\mathbb{Z}^2$ is processed.

This algorithm halts if *x* contains a forbidden pattern, and does not halt otherwise.

The computability obstruction

Definition

A set is effective if it is the sets of points on which a Turing machine does not halt.

Definition

S is effective if there is an algorithm that halts outside S

Some non trivial example :

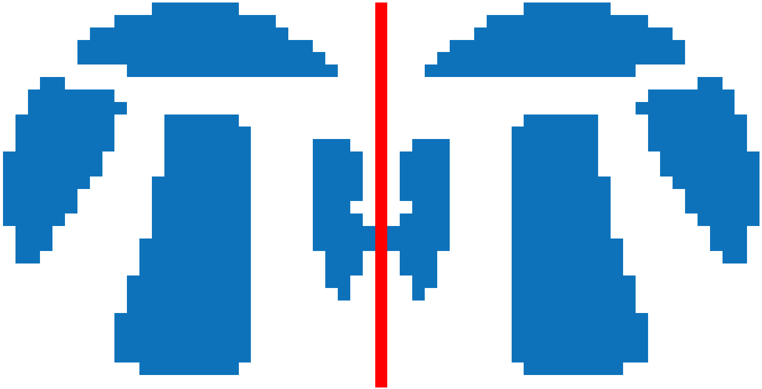
$$\{x \in \{0,1\}^{\mathbb{N}} | \text{the set of } i \text{ with } x_i = 1 \text{ is a subsemigroup of } \mathbb{N}\}$$

SFTs are effective

- SFTs are effective
- Sofic shifts are effective
- The Thue Morse shift is effective
- β-shifts are effective iff β is computable from above

Is effectivity the only obstruction ?
For technical reasons, all effective sets cannot be transformed into SFTs, but maybe all effective shifts ?

Some effective sets are not sofic...



Each line must be symmetric around the red cell, and all red cells must be aligned vertically.
This cannot be done with a SFT or a sofic shift : too much “information” should transit through the red line.

. . . All effective sets are sofic

Theorem (Aubrun-Sablik [AS13], Durand-Romashchenko-Shen [DRS10])

Every n-dimensional effective subshift S can be implemented by a $n+1$-dimensional sofic subshift $S^{\mathbb{Z}}$.
A point in $S^{\mathbb{Z}}$ consists of $\mathbb{Z}$ copies of the same $x \in S$.

(More on this in Nathalie's talk)

Some notes

- Every *n*-dimensional sofic shift is a *n*-dimensional effective shift
- Every *n*-dimensional effective shift is a $n+1$-dimensional sofic shift

Gives a framework for computability results :

- Computability Obstructions on SFTs are usually also obstructions for effective shifts
- Prove the obstruction is the only obstruction for effective shifts
- Use the previous theorem to go back to SFTs.

Does not always work (periodic points)

Conclusion

- Interplay between computability and symbolic dynamics
- Answers are very precise, but involve computability.

Every behaviour is possible, as long as it is computationally possible

Interesting new direction : Is everything still working with SFTs that are expansive in some directions (e.g. cellular automata) ?

Bibliography I

Nathalie Aubrun and Mathieu Sablik, *Simulation of effective subshifts by two-dimensional subshifts of finite type*, Acta Applicandae Mathematicae (2013).

Robert Berger, *The Undecidability of the Domino Problem*, Ph.D. thesis, Harvard University, 1964.

Bruno Durand, Andrei Romashchenko, and Alexander Shen, *Effective Closed Subshifts in 1D Can Be Implemented in 2D*, Fields of Logic and Computation, Lecture Notes in Computer Science, no. 6300, Springer, 2010, pp. 208–226.

Yuri Gurevich and I Koryakov, *Remarks on Berger's paper on the domino problem*, Siberian Math. Journal (1972), 319–320.

Bibliography II

- Michael Hochman and Tom Meyerovitch, *A characterization of the entropies of multidimensional shifts of finite type*, Annals of Mathematics **171** (2010), no. 3, 2011–2038.
- Raphael M. Robinson, *Undecidability and Nonperiodicity for Tilings of the Plane*, Inventiones Mathematicae **12** (1971), no. 3, 177–209.