Computational Aspects of Symbolic Dynamics

Part I: Effective Dynamics

E. Jeandel

LORIA (Nancy, France)

Introduction

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.

Introduction

Douglas Lind [Lin04] :

> *The fact that none of these three basic questions, (1) the existence of points, (2) the extension of finite configurations, (3) the existence of periodic points, can be decided by a finite procedure is what I call "The Swamp of Undecidability." It's a place you don't want to go.*

Introduction

For many dynamical problems, there is no way to actually use the fact the subshift is of finite type.

In most results, “of finite type” and “with a computable sequence of forbidden patterns” are interchangeable.

There are a few transfer principles that explain that.

Outline

- Part I : Definitions
  - Effective subshifts
- Part II : The transfer principle
  - Effective subshifts can be embedded into SFTs.
- Part III : Recursion-theoretic invariants
  - SFTs and effective subshifts have roughly the same properties
- Part IV : Differences
  - SFTs have nonetheless some specific properties.

Computability

- Classical Computability (or Recursivity) Theory is concerned with integers, finite words, etc.
- A set $S \subseteq \mathbb{N}$ is *computable* (recursive) if there is an algorithm that can decide on input n whether $n \in S$
- A partial map f is recursive if there is an algorithm that compute $f(n)$ given n.

Algorithms are seldom total functions, they might not halt on some inputs.

Enumerability

- S is recursively-enumerable if $S = \{f(n), n \in \mathbb{N}\}$ for some recursive total map f.
- Equivalently, there is an algorithm g that halts on n exactly when $n \in S$.

We can “show” that $n \in S$.

An example

If S is an SFT, then the set L of patterns that are *not* globally admissible is recursively enumerable.

The algorithm, given a pattern w as an input, tries, for all n, to find a $n \times n$ extension of w.

- If it does not succeed for some n, then $w \in L$
- If it never halts, then $w \notin L$ by compactness

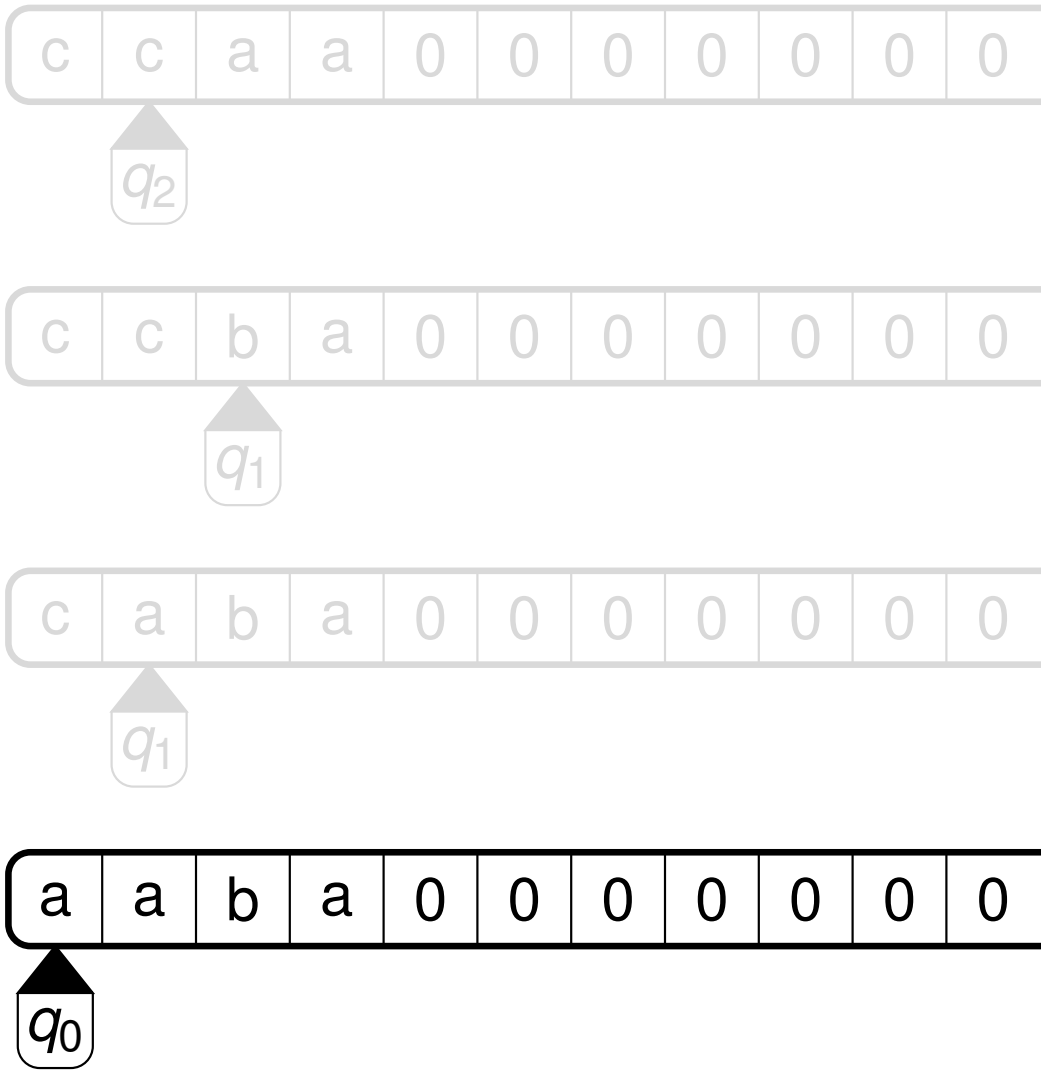
Turing Machines

The theoretical model of computation is *Turing machines*.
In its simplest form, a Turing Machine contains :

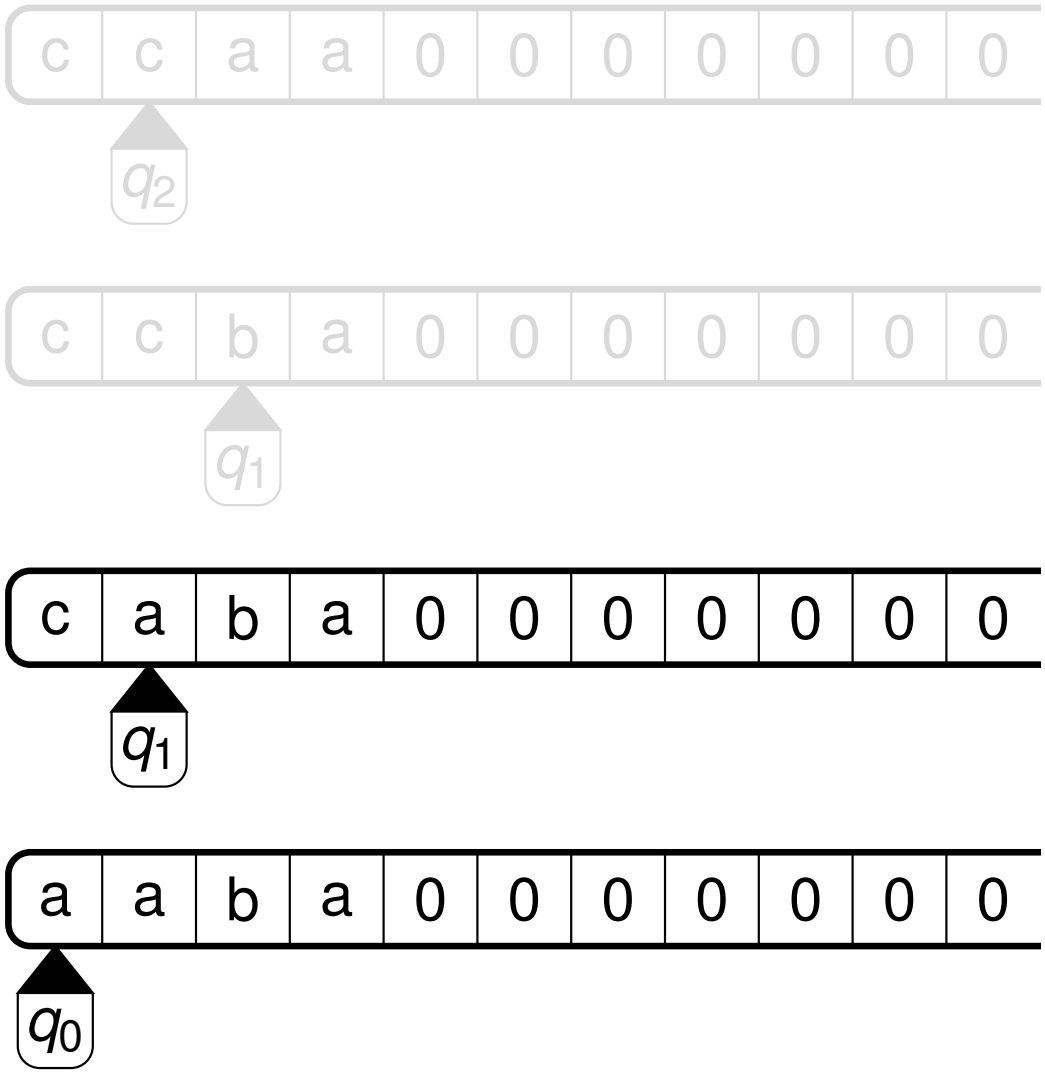
- An infinite tape, that can contain symbols in Σ
- A distinguished position on the tape (the head)
- A state in Q
- An update function $Q \times \Sigma \to Q \times \Sigma \times \{-1, 0, 1\}$

The input is initially written on the tape, and the machine evolves from a specific (initial) state until reaching a specific (halting) state.

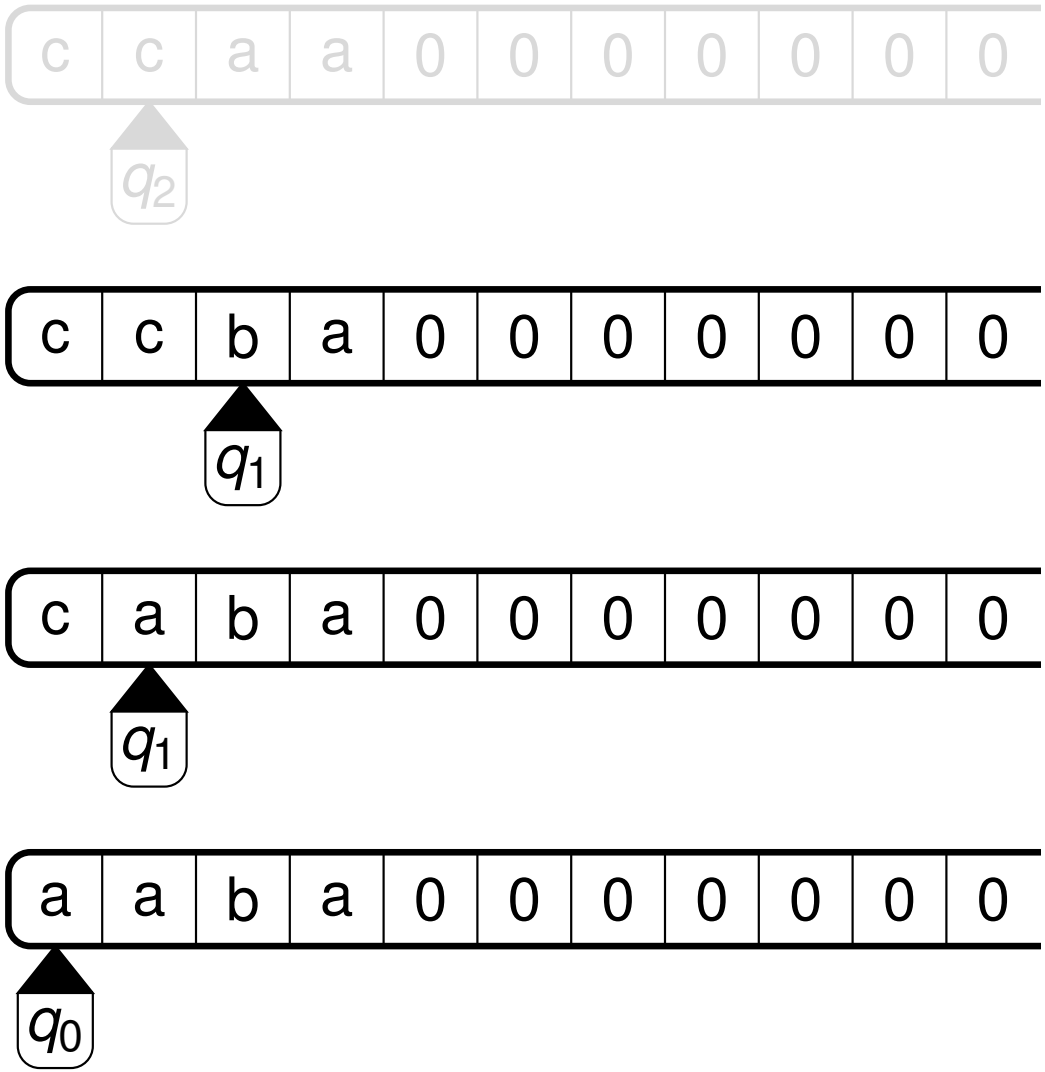
Turing Machines



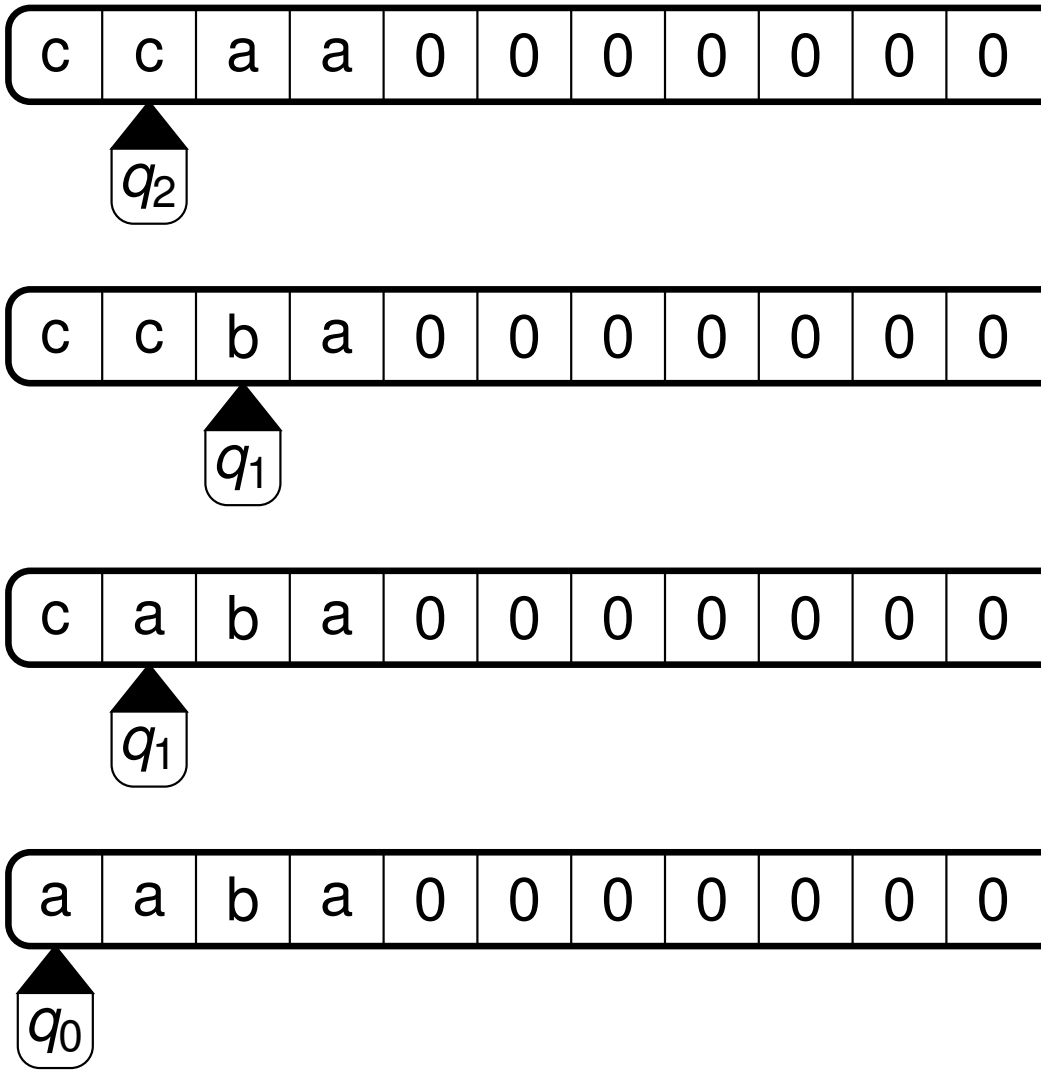
Turing Machines



Turing Machines



Turing Machines



Computability in the Cantor Space

We also need to speak about computability of points in a subshift, i.e. in a Cantor space

- A notion of computability for points of $A^{\mathbb{N}}$, or for infinite sequences ?

Type 2 algorithm

- A Type 2 algorithm takes as input an infinite word $x \in A^{\mathbb{N}}$
- It acts as a regular algorithm, but may at any time ask for the value of x_i for a given $i \in \mathbb{N}$

A map f is computable if there is an algorithm that given x and $n \in \mathbb{N}$ computes $f(x)_n$ (the n-th letter of $f(x)$).

Example

A factor map is computable.

$f : x \mapsto y$ so that

$$y_j = g(x_{j-R}, x_{j-R+1} \ldots, x_0, x_1, \ldots x_R)$$

for some finite map g

Turing Machines

How to do this with a Turing Machine ?

- A specific tape for the output
- The (infinite) input is written on the first tape
- The output tape is *write-once*

The Use principle

If f is computable, it needs only the knowledge of finitely many letters of x to compute the first n letters of $f(x)$.

Computable functions are continuous.

Computable functions are the computability counterpart of continuous functions.

Closed sets

A set S is effectively open if there exists an algorithm f that halts on input x iff $x \in S$.

The equivalent of “recursively enumerable”.

- Effectively open sets are open : if f halts on x, it will have read only finitely many letters of x, hence f halts on a neighbourhood of x.

A set S is effectively closed if its complement is effectively open

Example

Every SFT is effectively closed.

Starting from a point x, the algorithm searches for a forbidden pattern.

Alternative definition

A set S of infinite words is effectively closed if it can be given by a recursively enumerable set of forbidden *prefixes*.

We forbid u whenever f halts on input u without trying to read the rest of the word.

We can replace ”recursively enumerable” by “computable” in the definition. (Not hard, but out of scope for this talk)

Properties

- The intersection of two effectively closed sets is effectively closed
  - Take the union of the two sets of forbidden prefixes
- The union of two effectively closed sets is effectively closed
  - Launch the two algorithms in parallel. Halt only when both halt.
- There is an algorithm that halts iff S is empty
  - For each n, the algorithm tests whether all words of size n contain a forbidden prefix. If they do, it halts.
  - Works by a standard compactness argument.

Recursive Analysis

Computable functions behave nicely with closed sets :

The preimage of an effectively closed set by a computable function is effectively closed

$x \in f^{-1}(S) \iff f(x) \in S$

The image of an effectively closed set by a computable function is effectively closed

Forbid all prefixes w so that $f^{-1}(wA^{\mathbb{N}}) \cap S = \emptyset$

- There is an algorithm that halts whenever w has this property, hence the set of forbidden prefixes is recursively enumerable

The example

Every SFT is effectively closed.

The main reason for all undecidability results on SFTs is that there is a partial converse :

Every effectively closed set of $A^{\mathbb{N}}$ may be embedded into a two-dimensional SFT

The exact statement

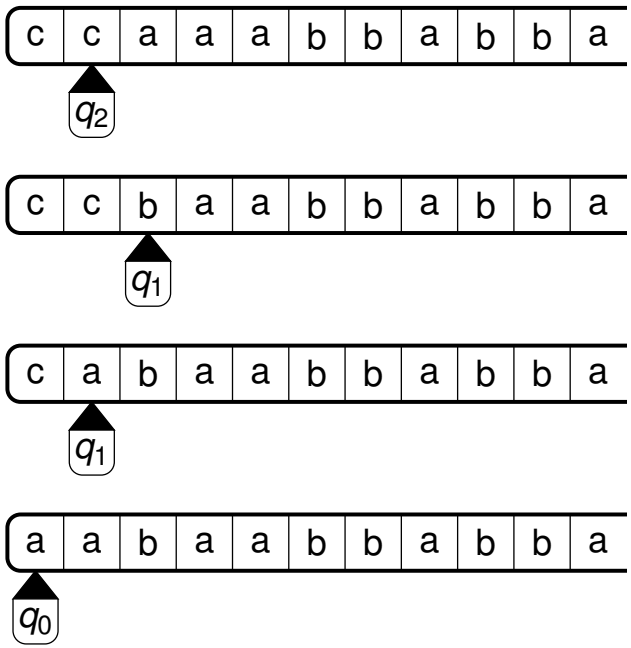
For a SFT S and a letter a, let S_a be the set of all points in S that have the letter a at the center.

Theorem (essentially Hanf [Han74])

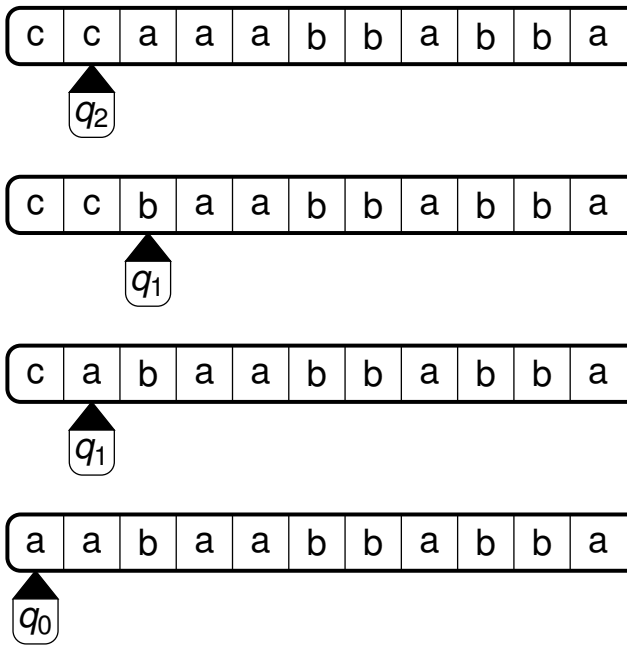
For any effectively closed set $X \subseteq A^{\mathbb{N}}$ there exists a two-dimensional SFT S and a letter a so that S_a is recursively homeomorphic to X.

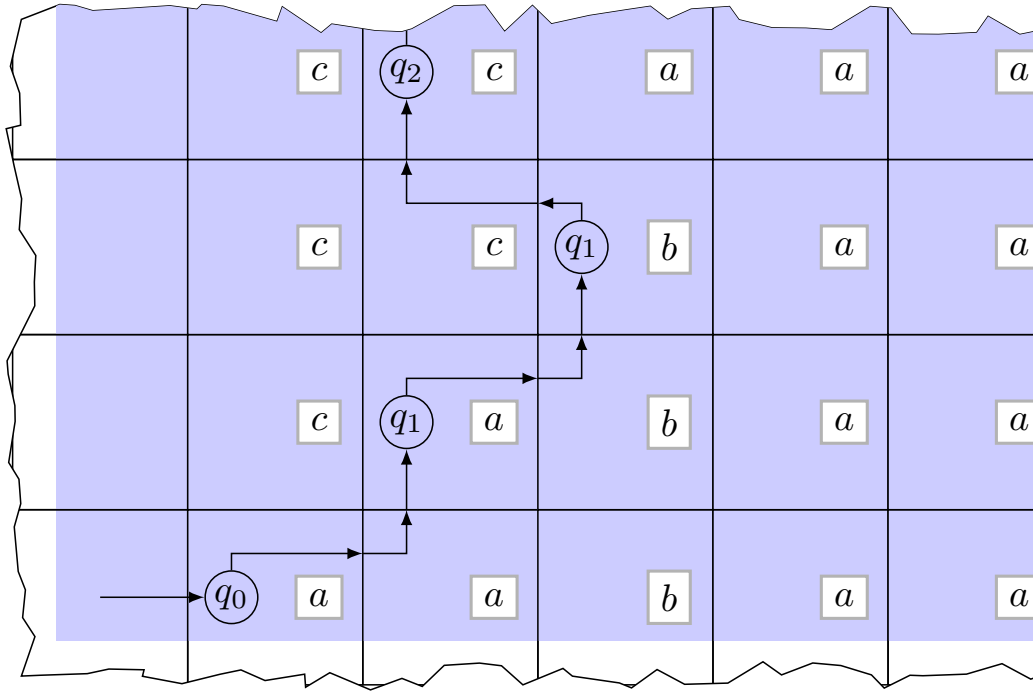
A recursive homeomorphism f is a homeomorphism that is computable with a computable inverse.

The proof

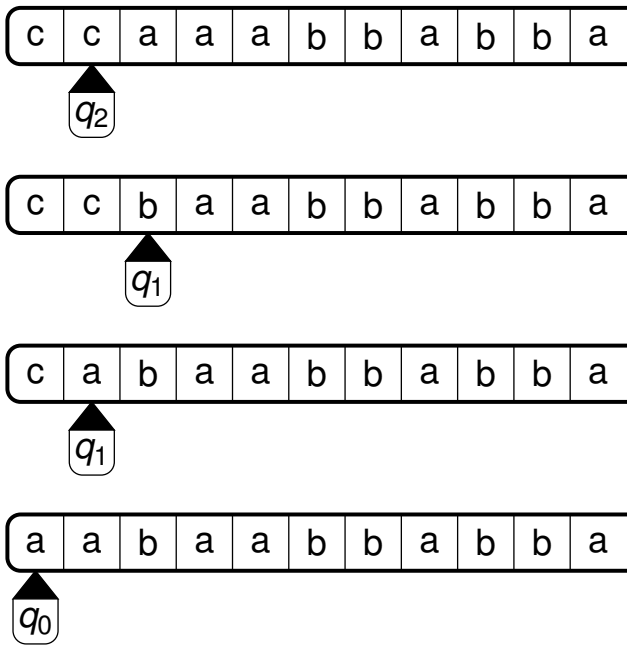

c c a a a b b a b b a
q_2
c c b a a b b a b b a
q_1
c a b a a b b a b b a
q_1
a a b a a b b a b b a
q_0

The proof

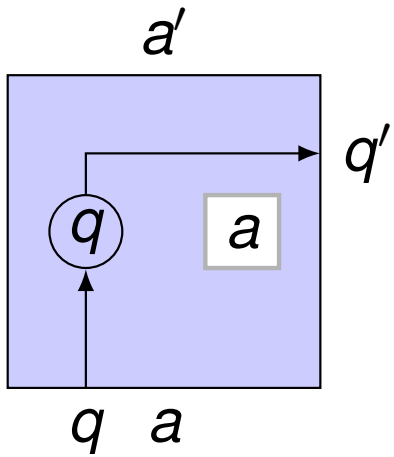

c c a a a b b a b b a
q_2
c c b a a b b a b b a
q_1
c a b a a b b a b b a
q_1
a a b a a b b a b b a
q_0


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

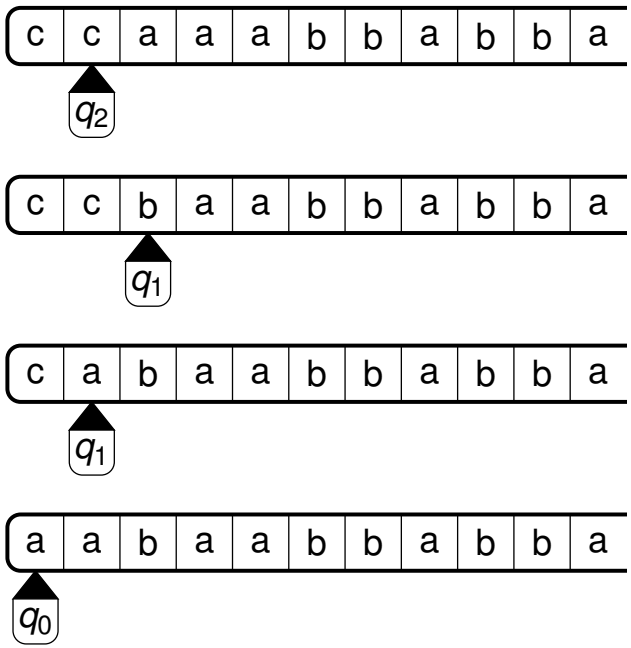
The proof



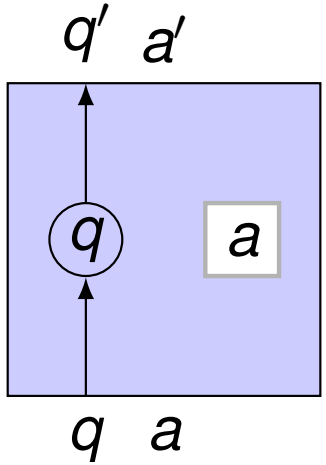
$$(q, a) \longrightarrow (q', a', \rightarrow)$$



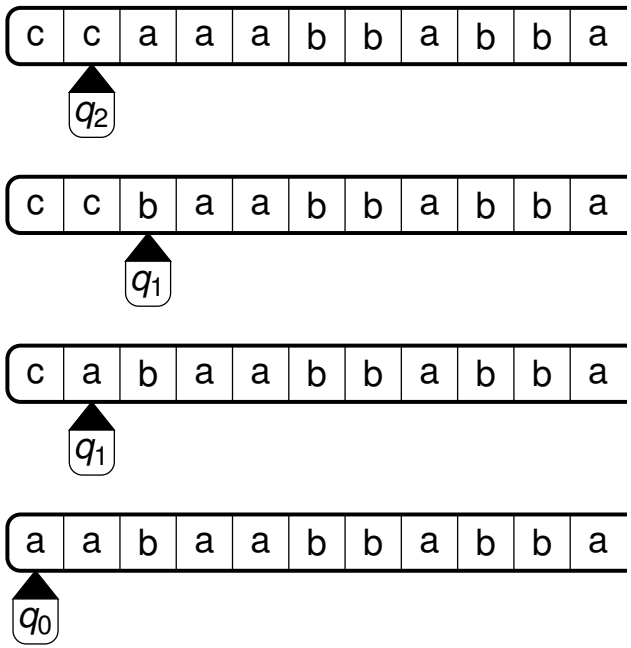
The proof



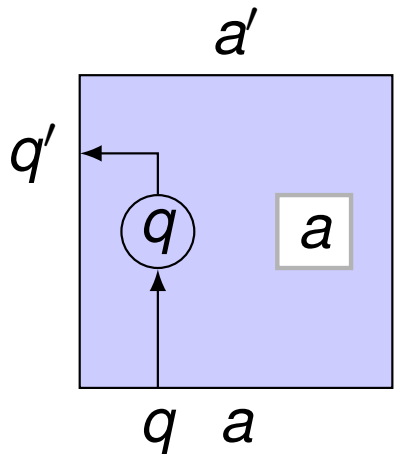
$$(q, a) \longrightarrow (q', a', \uparrow)$$



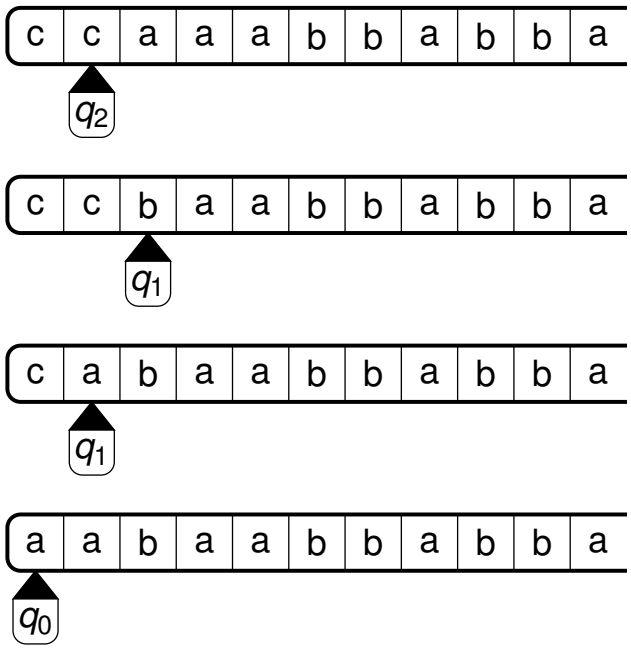
The proof

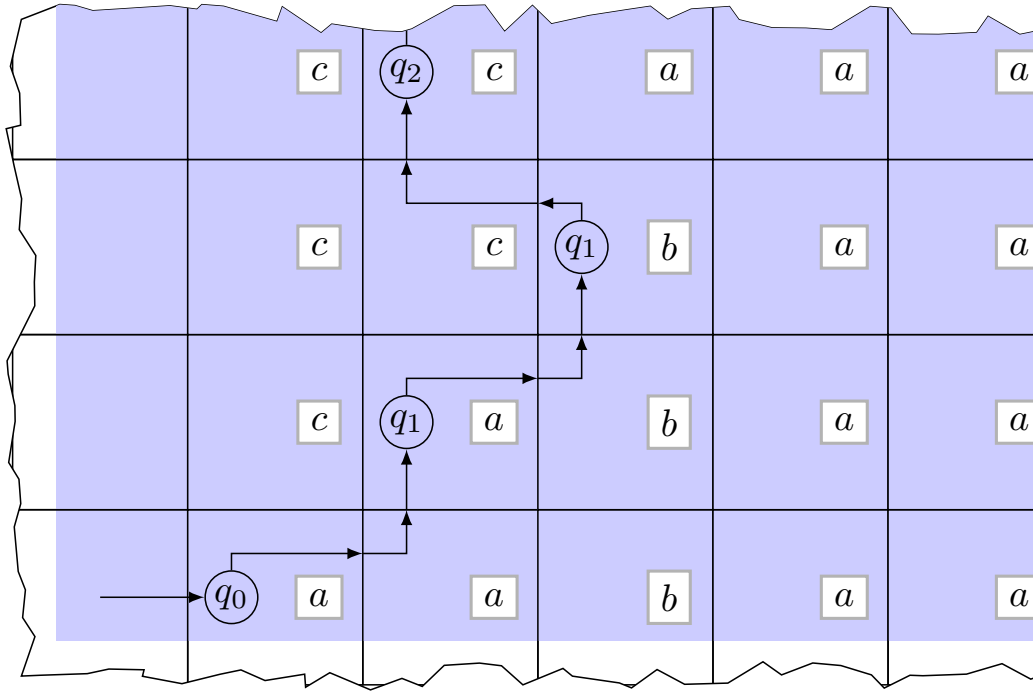


$$(q, a) \longrightarrow (q', a', \leftarrow)$$



The proof


c c a a a b b a b b a
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

On the result

Can we do better ?

- It's not really possible to have a recursive homeomorphism between X and the whole of S
  - S is shift-invariant. This would imply the existence of a non trivial automorphism of X.
  - More about this on part *III* of the talk
- What about embedding a *shift invariant* effectively closed set ?

Effective subshifts

An effective subshift is a subshift which is effectively closed.

Alternate (more useful) definition. :

- A subshift is effective if it can be given by a computable list of forbidden patterns.

SFTs are effective subshifts.
Sofic shifts are effective subshifts.

Effective subshifts

Effective subshifts are more general than SFT/sofic shifts.

Theorem (Cenzer-Dashti-King [CDK08])

If $f : X \to X$ is computable and X effectively closed, then $It(f)$ is an effective shift.

$$It(f) = \{x_0 f(x)_0 f(f(x))_0 \dots\}$$

Conversely, for every effective shift S, there exists f so that $S = It(f)$.

The theorem

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

For every n-dimensional effective subshift S, the $n+1$-dimensional subshift :

$$S^{\mathbb{Z}} = \{y | \exists x \in S, \forall i, j, y_{ij} = x_i\}$$

$$S^{\mathbb{Z}} = \{y | \text{all lines are equal to the same } x \in S\}$$

is sofic.
(That is, there exists a SFT X and an onto factor map $f : X \to S^{\mathbb{Z}}$)

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

This theorem explains a lot of the similarities between SFTs and effective subshifts.

- A proof by Hochman [Hoc09] with $n \mapsto n+2$.
- Extended to $n \mapsto n+1$ by Aubrun-Sablik and Durand-Romashchenko-Shen independently

Next time : the proof by Aubrun-Sablik.

Bibliography I

- Nathalie Aubrun and Mathieu Sablik, *Simulation of effective subshifts by two-dimensional SFT and a generalization*, preprint.
- Robert Berger, *The Undecidability of the Domino Problem*, Ph.D. thesis, Harvard University, 1964.
- Douglas Cenzer, Ali Dashti, and Jonathan L. F. King, *Computable symbolic dynamics*, Mathematical Logic Quarterly **54** (2008), no. 5, 460–469.
- 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

- William Hanf, *Non Recursive Tilings of the Plane I*, Journal of Symbolic Logic **39** (1974), no. 2, 283–285.
- Michael Hochman, *On the dynamics and recursive properties of multidimensional symbolic systems*, Inventiones Mathematicae **176** (2009), no. 1, 2009.
- Douglas A. Lind, *Multi-Dimensional Symbolic Dynamics*, Symbolic Dynamics and its Applications (Susan G. Williams, ed.), Proceedings of Symposia in Applied Mathematics, no. 60, American Mathematical Society, 2004, pp. 61–79.
- Raphael M. Robinson, *Undecidability and Nonperiodicity for Tilings of the Plane*, Inventiones Mathematicae **12** (1971), no. 3, 177–209.