On Stable and Unstable Limit Sets of Finite Families of Cellular Automata

Ville Salo, Ilkka Törmä

University of Turku

March 7, 2012

Subshifts and cellular automata

- The *full shift* over the alphabet S is the set $S^{\mathbb{Z}}$ with the product topology.

Subshifts and cellular automata

- The *full shift* over the alphabet S is the set $S^{\mathbb{Z}}$ with the product topology.
- A *subshift* is a closed shift-invariant subset of the full shift – or defined by a set of forbidden words.

Subshifts and cellular automata

- The *full shift* over the alphabet S is the set $S^{\mathbb{Z}}$ with the product topology.
- A *subshift* is a closed shift-invariant subset of the full shift – or defined by a set of forbidden words.
- A subshift X is *transitive* if $u, v \sqsubset X \implies \exists w : uwv \sqsubset X$, and *mixing* if the length of w can be chosen freely, as long as it is long enough.

Subshifts and cellular automata

- The *full shift* over the alphabet S is the set $S^{\mathbb{Z}}$ with the product topology.
- A *subshift* is a closed shift-invariant subset of the full shift – or defined by a set of forbidden words.
- A subshift X is *transitive* if $u, v \sqsubset X \implies \exists w : uwv \sqsubset X$, and *mixing* if the length of w can be chosen freely, as long as it is long enough.
- A *cellular automaton* is a continuous function on a subshift that commutes with the left shift.

SFTs and sofic shifts

- If the set of forbidden words defining a subshift can be taken to be finite, the subshift is said to be of finite type, an *SFT*.

SFTs and sofic shifts

- If the set of forbidden words defining a subshift can be taken to be finite, the subshift is said to be of finite type, an *SFT*.
- Continuous shift-commuting maps (*block codes*) are the natural morphisms of subshifts.

SFTs and sofic shifts

- If the set of forbidden words defining a subshift can be taken to be finite, the subshift is said to be of finite type, an *SFT*.
- Continuous shift-commuting maps (*block codes*) are the natural morphisms of subshifts.
- If there exists a block code from an SFT onto subshift Y, then Y is said to be *sofic*.

SFTs and sofic shifts

- If the set of forbidden words defining a subshift can be taken to be finite, the subshift is said to be of finite type, an *SFT*.
- Continuous shift-commuting maps (*block codes*) are the natural morphisms of subshifts.
- If there exists a block code from an SFT onto subshift Y, then Y is said to be *sofic*.
- We say Y is a factor of X if X maps onto Y by a block code.

SFTs and sofic shifts

- If the set of forbidden words defining a subshift can be taken to be finite, the subshift is said to be of finite type, an *SFT*.
- Continuous shift-commuting maps (*block codes*) are the natural morphisms of subshifts.
- If there exists a block code from an SFT onto subshift Y, then Y is said to be *sofic*.
- We say Y is a factor of X if X maps onto Y by a block code.
- We say X and Y are conjugate if there exists a bijective block code between them (its inverse is then also a block code by compactness of subshifts).

The limit set

- The *limit set* of a cellular automaton $f : X \to X$ is usually defined as
$$\bigcap_n f^n(X)$$
or the points with an infinite chain of preimages. The limit set is always a subshift.

The limit set

- The *limit set* of a cellular automaton $f : X \to X$ is usually defined as
$$\bigcap_n f^n(X)$$
or the points with an infinite chain of preimages. The limit set is always a subshift.
- The idea is that this is that we get closer and closer to the limit set of f as f is iterated.

The limit set

- The *limit set* of a cellular automaton $f : X \to X$ is usually defined as
$$\bigcap_n f^n(X)$$
or the points with an infinite chain of preimages. The limit set is always a subshift.
- The idea is that this is that we get closer and closer to the limit set of f as f is iterated.
- When the system can evolve in multiple ways, and one is chosen at each step, we can model this with a family of CA.

The limit set

- The *limit set* of a cellular automaton $f : X \to X$ is usually defined as
$$\bigcap_n f^n(X)$$
or the points with an infinite chain of preimages. The limit set is always a subshift.
- The idea is that this is that we get closer and closer to the limit set of f as f is iterated.
- When the system can evolve in multiple ways, and one is chosen at each step, we can model this with a family of CA.
- We define the limit set of such a family $\mathcal{F}$ as $\bigcap_n L_n(\mathcal{F})$ where $L_0(\mathcal{F}) = X$ and $L_{i+1}(\mathcal{F}) = \bigcup_{f \in \mathcal{F}} f(L_i(\mathcal{F}))$.

The limit set

- The *limit set* of a cellular automaton $f : X \to X$ is usually defined as
$$\bigcap_n f^n(X)$$
or the points with an infinite chain of preimages. The limit set is always a subshift.
- The idea is that this is that we get closer and closer to the limit set of f as f is iterated.
- When the system can evolve in multiple ways, and one is chosen at each step, we can model this with a family of CA.
- We define the limit set of such a family $\mathcal{F}$ as $\bigcap_n L_n(\mathcal{F})$ where $L_0(\mathcal{F}) = X$ and $L_{i+1}(\mathcal{F}) = \bigcup_{f \in \mathcal{F}} f(L_i(\mathcal{F}))$.
- Again, exactly the points that can appear arbitrarily late in a system with these CA as the dynamics.

The limit set

- The *limit set* of a cellular automaton $f : X \to X$ is usually defined as
$$\bigcap_n f^n(X)$$
or the points with an infinite chain of preimages. The limit set is always a subshift.
- The idea is that this is that we get closer and closer to the limit set of f as f is iterated.
- When the system can evolve in multiple ways, and one is chosen at each step, we can model this with a family of CA.
- We define the limit set of such a family $\mathcal{F}$ as $\bigcap_n L_n(\mathcal{F})$ where $L_0(\mathcal{F}) = X$ and $L_{i+1}(\mathcal{F}) = \bigcup_{f \in \mathcal{F}} f(L_i(\mathcal{F}))$.
- Again, exactly the points that can appear arbitrarily late in a system with these CA as the dynamics.
- We will restrict to CA on the full shift.

Stability and the stable and unstable hierarchies

- Just like for the limit set of a single automaton, we say a limit set is *stable* if the limit set is actually reached in a finite amont of steps: $L_n(X) = L_{n+1}(X)$ for some n.

Stability and the stable and unstable hierarchies

- Just like for the limit set of a single automaton, we say a limit set is *stable* if the limit set is actually reached in a finite amont of steps: $L_n(X) = L_{n+1}(X)$ for some n.
- This means that if a point x is mapped with any product of any n cellular automata in the family considered, then the resulting point y also has some infinite chain of preimages.

Stability and the stable and unstable hierarchies

- Just like for the limit set of a single automaton, we say a limit set is *stable* if the limit set is actually reached in a finite amont of steps: $L_n(X) = L_{n+1}(X)$ for some n.
- This means that if a point x is mapped with any product of any n cellular automata in the family considered, then the resulting point y also has some infinite chain of preimages.
- The classes k-LIM$_s$ and k-LIM$_u$ denote the classes of stable and unstable limit sets of k cellular automata, 1-LIM$_s$ and 1-LIM$_u$ are just the usual stable and unstable limit sets. We write k-LIM$_x$ for k-LIM$_s \cup k$-LIM$_u$.

Stability and the stable and unstable hierarchies

- Just like for the limit set of a single automaton, we say a limit set is *stable* if the limit set is actually reached in a finite amont of steps: $L_n(X) = L_{n+1}(X)$ for some n.
- This means that if a point x is mapped with any product of any n cellular automata in the family considered, then the resulting point y also has some infinite chain of preimages.
- The classes k-LIM$_s$ and k-LIM$_u$ denote the classes of stable and unstable limit sets of k cellular automata, 1-LIM$_s$ and 1-LIM$_u$ are just the usual stable and unstable limit sets. We write k-LIM$_x$ for k-LIM$_s$ $\cup$ k-LIM$_u$.
- It is known that 1-LIM$_s$ and 1-LIM$_u$ are incomparable.

An example of a complicated limit set of a CA family

Example

Consider the two automata f_0 and f_1 on the alphabet $\{0, 1, \#\}$ where each f_i has radius $\frac{1}{2}$, and the local rule of f_i is given by the following table:

	0	1	#
0	0	0	#
1	1	1	#
#	i	i	#

Now the limit set $L(\{f_0, f_1\})$ is the subshift defined by the forbidden words $\{\#uv\#w \mid n \in \mathbb{N}, u, w \in \{0,1\}^n, v \in \{0,1,\#\}^*, u \neq w\}$.

(We do not know if this is an unstable limit set of a single CA, but it seems unlikely.)

Subshifts in two dimensions

- We can give the same definitions in the space $S^{\mathbb{Z}^2}$, to obtain the two-dimensional subshifts, in particular $\mathbb{Z}^2$ SFTs.

Subshifts in two dimensions

- We can give the same definitions in the space $S^{\mathbb{Z}^2}$, to obtain the two-dimensional subshifts, in particular $\mathbb{Z}^2$ SFTs.
- The possible contents of horizontal lines of a 2D SFT are called its $\mathbb{Z}$-projective subdynamics, and the class of such subshifts is denoted $\mathcal{PRO}$.

Subshifts in two dimensions

- We can give the same definitions in the space $S^{\mathbb{Z}^2}$, to obtain the two-dimensional subshifts, in particular $\mathbb{Z}^2$ SFTs.
- The possible contents of horizontal lines of a 2D SFT are called its $\mathbb{Z}$-projective subdynamics, and the class of such subshifts is denoted $\mathcal{PRO}$.
- Limit sets of finite families of cellular automata can be thought of as a concept between the usual limit sets and projective subdynamics.

- We do not know if ∞-LIM$_x \subset \mathcal{PRO}$.

- We do not know if $\infty\text{-LIM}_x \subset \mathcal{PRO}$.
- We can only prove that for a large class $\mathcal{CLS}$ of subshifts, $\infty\text{-LIM}_x \cap \mathcal{CLS} \subset \mathcal{PRO}$.

- We do not know if ∞-LIM$_x \subset \mathcal{PRO}$.
- We can only prove that for a large class $\mathcal{CLS}$ of subshifts, ∞-LIM$_x \cap \mathcal{CLS} \subset \mathcal{PRO}$.
- Such proofs amount to encoding the cellular automaton used between pairs of rows in some way.

- We do not know if ∞-LIM$_x \subset \mathcal{PRO}$.
- We can only prove that for a large class $\mathcal{CLS}$ of subshifts, ∞-LIM$_x \cap \mathcal{CLS} \subset \mathcal{PRO}$.
- Such proofs amount to encoding the cellular automaton used between pairs of rows in some way.
- At least, $\mathcal{PRO}$ contains $X \times Y$ for any ∞-LIM$_x$ subshift X and some subshift Y with only unary points (where Y chooses the CA used at each step).

A negative result for limit sets of CA families

- A one-dimensional subshift X has *universal period* l if there exists M such that for all $x \in X$ there exists y with $y = \sigma^l(y)$ such that $|\{i \mid x_i \neq y_i\}| \leq M$.

A negative result for limit sets of CA families

- A one-dimensional subshift X has *universal period* l if there exists M such that for all $x \in X$ there exists y with $y = \sigma^l(y)$ such that $|\{i \mid x_i \neq y_i\}| \leq M$.

Lemma

[Ronnie Pavlov, Michael Schraudner] A zero-entropy proper one-dimensional sofic shift X is realizable as the $\mathbb{Z}$-projective subdynamics of a $\mathbb{Z}^2$ SFT if and only if it has no universal period.

A negative result for limit sets of CA families

- A one-dimensional subshift X has *universal period* l if there exists M such that for all $x \in X$ there exists y with $y = \sigma^l(y)$ such that $|\{i \mid x_i \neq y_i\}| \leq M$.

Lemma

[Ronnie Pavlov, Michael Schraudner] A zero-entropy proper one-dimensional sofic shift X is realizable as the $\mathbb{Z}$-projective subdynamics of a $\mathbb{Z}^2$ SFT if and only if it has no universal period.

Corollary

A zero-entropy proper sofic shift with a universal period is not the limit set of any finite family of CA.

Some limit sets which are $\mathbb{Z}$-projective subdynamics

- Another technique is to encode the automaton using periodic points of the limit set X.

Some limit sets which are $\mathbb{Z}$-projective subdynamics

- Another technique is to encode the automaton using periodic points of the limit set X.

Theorem

If X is the limit set of a family of cellular automata containing at least two periodic points, then X is realizable as the $\mathbb{Z}$-projective subdynamics of a $\mathbb{Z}^2$ SFT.

Some limit sets which are $\mathbb{Z}$-projective subdynamics

- Another technique is to encode the automaton using periodic points of the limit set X.

Theorem

If X is the limit set of a family of cellular automata containing at least two periodic points, then X is realizable as the $\mathbb{Z}$-projective subdynamics of a $\mathbb{Z}^2$ SFT.

Corollary

All stable limit sets X of finite families of cellular automata are realizable as the $\mathbb{Z}$-projective subdynamics of a $\mathbb{Z}^2$ SFT.

Proof scetch

Theorem

If $X \in \infty$-LIM$_x$ has at least two periodic points, then $X \in \mathcal{PRO}$.

Proof scetch

Theorem

If $X \in \infty$-LIM$_x$ has at least two periodic points, then $X \in \mathcal{PRO}$.

- The subshift has a 'data row' every k steps, and the rows in between have one of the two periodic points ('control rows'). The SFT rule can locate the data rows if the allowed combinations of periodic points are chosen appropriately.

Proof scetch

Theorem

If $X \in \infty$-LIM_x has at least two periodic points, then $X \in \mathcal{PRO}$.

- The subshift has a 'data row' every k steps, and the rows in between have one of the two periodic points ('control rows'). The SFT rule can locate the data rows if the allowed combinations of periodic points are chosen appropriately.
- Now, the CA that is run from one data run to the next is determined by what is encoded in the k control rows in between.

The results on the hierarchies 1-LIM$_s$, 2-LIM$_s$, . . . and 1-LIM$_u$, 2-LIM$_u$, . . .

- We can prove that each level of each hierarchy is disjoint from the other hierarchy.

The results on the hierarchies 1-LIM$_s$, 2-LIM$_s$, . . . and 1-LIM$_u$, 2-LIM$_u$, . . .

- We can prove that each level of each hierarchy is disjoint from the other hierarchy.
- Using known results in symbolic dynamics and some techniques of our own, we can prove that both hierarchies are proper.

The results on the hierarchies 1-LIM$_s$, 2-LIM$_s$, . . . and 1-LIM$_u$, 2-LIM$_u$, . . .

- We can prove that each level of each hierarchy is disjoint from the other hierarchy.
- Using known results in symbolic dynamics and some techniques of our own, we can prove that both hierarchies are proper.
- However, the properness requires nontransitive subshifts, and for transitive subshifts, we can in fact prove that the stable hierarchy collapses to 1-LIM$_s$, and ∞-LIM$_u \cap \mathcal{TRA} \subset$ 1-LIM$_x$ where $\mathcal{TRA}$ denotes the transitive subshifts.

Properness of the hierarchies

- The idea is to first find, for any k, k SFTs that do not factor onto each other which all contain a unary point.

Properness of the hierarchies

- The idea is to first find, for any k, k SFTs that do not factor onto each other which all contain a unary point.
- For this, in order to easily access tools from symbolic dynamics we need to represent our SFTs with matrices:

Properness of the hierarchies

- The idea is to first find, for any k, k SFTs that do not factor onto each other which all contain a unary point.
- For this, in order to easily access tools from symbolic dynamics we need to represent our SFTs with matrices:
  - To a square $n \times n$ matrix A over the natural numbers we associate a graph with n vertices and A_{ij} edges from vertex i to vertex j.

Properness of the hierarchies

- The idea is to first find, for any k, k SFTs that do not factor onto each other which all contain a unary point.
- For this, in order to easily access tools from symbolic dynamics we need to represent our SFTs with matrices:
  - To a square $n \times n$ matrix A over the natural numbers we associate a graph with n vertices and A_{ij} edges from vertex i to vertex j.
  - To each finite graph we associate its edge shift by taking all the valid bi-infinite paths. This is easily seen to be an SFT, and every SFT can be represented by a graph, up to conjugacy.

Definition

If A is a primitive ('mixing') integral matrix, let λ_A be its greatest eigenvalue with respect to absolute value, and $\mathrm{sp}^\times(A)$ the unordered list (or multiset) of its eigenvalues, called the *nonzero spectrum* of A. We use the notation $\langle \lambda_1, \ldots, \lambda_k \rangle$ for the unordered list containing the elements λ_i.

Definition

If A is a primitive ('mixing') integral matrix, let λ_A be its greatest eigenvalue with respect to absolute value, and $\mathrm{sp}^\times(A)$ the unordered list (or multiset) of its eigenvalues, called the *nonzero spectrum* of A. We use the notation $\langle \lambda_1, \dots, \lambda_k \rangle$ for the unordered list containing the elements λ_i.

Lemma (Lind & Markus, Symbolic Dynamics and Coding)

The entropy of the edge shift X defined by a primitive integral matrix A is $\log \lambda_A$.

Lemma (Lind & Markus, Symbolic Dynamics and Coding)

If the edge shifts X and Y defined by two primitive integral matrices A and B, respectively, have the same entropy and X factors onto Y, then $\mathrm{sp}^\times(B) \subset \mathrm{sp}^\times(A)$.

Lemma (Lind & Markus, Symbolic Dynamics and Coding)

If the edge shifts X and Y defined by two primitive integral matrices A and B, respectively, have the same entropy and X factors onto Y, then $\mathrm{sp}^\times(B) \subset \mathrm{sp}^\times(A)$.

To recapitulate,

Lemma (Lind & Markus, Symbolic Dynamics and Coding)

If the edge shifts X and Y defined by two primitive integral matrices A and B, respectively, have the same entropy and X factors onto Y, then $\mathrm{sp}^\times(B) \subset \mathrm{sp}^\times(A)$.

To recapitulate,

- SFTs are essentially edge shifts defined by matrices.

Lemma (Lind & Markus, Symbolic Dynamics and Coding)

If the edge shifts X and Y defined by two primitive integral matrices A and B, respectively, have the same entropy and X factors onto Y, then $\mathrm{sp}^\times(B) \subset \mathrm{sp}^\times(A)$.

To recapitulate,

- SFTs are essentially edge shifts defined by matrices.
- We call the set of nonzero eigenvalues of a matrix its nonzero spectrum.

Lemma (Lind & Markus, Symbolic Dynamics and Coding)

If the edge shifts X and Y defined by two primitive integral matrices A and B, respectively, have the same entropy and X factors onto Y, then $\mathrm{sp}^\times(B) \subset \mathrm{sp}^\times(A)$.

To recapitulate,

- SFTs are essentially edge shifts defined by matrices.
- We call the set of nonzero eigenvalues of a matrix its nonzero spectrum.
- A factoring relation between mixing edge shifts with the same entropy implies a subset relation between the nonzero spectra of their matrices.

So, it's enough to find k matrices with the same largest eigenvalue, but no subset relations between the nonzero spectra. This can be done using the following lemma:

So, it's enough to find k matrices with the same largest eigenvalue, but no subset relations between the nonzero spectra. This can be done using the following lemma:

Lemma (Lind & Markus, Symbolic Dynamics and Coding)

Let A be a primitive integral matrix and B an integral matrix such that $\lambda_B < \lambda_A$, and

$$\mathrm{tr}_n(\mathrm{sp}^\times(A)) + \mathrm{tr}_n(\mathrm{sp}^\times(B)) \geq 0 \text{ for all } n \geq 1.$$

Then there is a primitive integral matrix C such that $\mathrm{sp}^\times(C) = \mathrm{sp}^\times(A) \cup \mathrm{sp}^\times(B)$.

So, it's enough to find k matrices with the same largest eigenvalue, but no subset relations between the nonzero spectra. This can be done using the following lemma:

Lemma (Lind & Markus, Symbolic Dynamics and Coding)

Let A be a primitive integral matrix and B an integral matrix such that $\lambda_B < \lambda_A$, and

$$\mathrm{tr}_n(\mathrm{sp}^\times(A)) + \mathrm{tr}_n(\mathrm{sp}^\times(B)) \geq 0 \text{ for all } n \geq 1.$$

Then there is a primitive integral matrix C such that $\mathrm{sp}^\times(C) = \mathrm{sp}^\times(A) \cup \ \mathrm{sp}^\times(B)$.

The matrices $A = [\lambda]$ and $B_i = [i]$ satisfy the requirements of the lemma for $i < k$ and large enough $\lambda > k$. Clearly, $\{i, \lambda\}$ have no subset relations, and the matrices with these nonzero spectra have the same largest eigenvalue λ.

We thus have:

Lemma

For all $k \in \mathbb{N}$, there exists a finite alphabet S_k, a symbol $a \in S_k$ and a set $\{X_1, \ldots, X_k\}$ of k mixing edge shifts over S_k such that whenever $i \neq j$, we have that X_i does not factor onto X_j, $X_i \cap X_j = {}^\infty a^\infty$ and $\mathcal{B}_1(X_i) \cap \mathcal{B}_1(X_j) = a$.

We thus have:

Lemma

For all $k \in \mathbb{N}$, there exists a finite alphabet S_k, a symbol $a \in S_k$ and a set $\{X_1, \ldots, X_k\}$ of k mixing edge shifts over S_k such that whenever $i \neq j$, we have that X_i does not factor onto X_j, $X_i \cap X_j = {}^\infty a^\infty$ and $\mathcal{B}_1(X_i) \cap \mathcal{B}_1(X_j) = a$.

From this, we extract the properness of both hierarchies $(k\text{-LIM}_s)_k$ and $(k\text{-LIM}_u)_k$.

Proof scetch of properness of stable hierarchy: realizing the union of the X_i with k automata

- First, we show $X = \bigcup_i X_i$ is in k-LIM$_s$.

Proof scetch of properness of stable hierarchy: realizing the union of the X_i with k automata

- First, we show $X = \bigcup_i X_i$ is in k-LIM$_s$.
- Let f_i have X_i as its (stable) limit set. This is possible since X_i are mixing SFTs and have unary points.

Proof scetch of properness of stable hierarchy: realizing the union of the X_i with k automata

- First, we show $X = \bigcup_i X_i$ is in k-LIM$_s$.
- Let f_i have X_i as its (stable) limit set. This is possible since X_i are mixing SFTs and have unary points.
- We extend f_i to the alphabet S_k by considering symbols in $B_1(X_j)$ as a for $j \neq i$.

Proof scetch of properness of stable hierarchy: realizing the union of the X_i with k automata

- First, we show $X = \bigcup_i X_i$ is in k-LIM$_s$.
- Let f_i have X_i as its (stable) limit set. This is possible since X_i are mixing SFTs and have unary points.
- We extend f_i to the alphabet S_k by considering symbols in $B_1(X_j)$ as a for $j \neq i$.
- Then $\{f_1, \ldots, f_k\}$ has $X = \bigcup_i X_i$ as its limit set.

Proof scetch of properness of stable hierarchy: extracting a factoring relation from any smaller family

- No, assume that $X \in (k-1)$-LIM$_s$ and let $\{f_1, \ldots, f_{k-1}\}$ be a CA family whose limit set it is. Let this limit set be reached in n steps.

Proof scetch of properness of stable hierarchy: extracting a factoring relation from any smaller family

- No, assume that $X \in (k-1)$-LIM$_s$ and let $\{f_1, \ldots, f_{k-1}\}$ be a CA family whose limit set it is. Let this limit set be reached in n steps.
- Take a doubly transitive point in some X_i. It must have a preimage with some f_m, which implies that some X_j must be mapped onto X_i by f_m. In fact, it is easy to see that f_m must also map X_j into X_i, and so $i = j$ by the assumption that there are no factoring relations.

Proof scetch of properness of stable hierarchy: extracting a factoring relation from any smaller family

- No, assume that $X \in (k-1)$-LIM$_s$ and let $\{f_1, \ldots, f_{k-1}\}$ be a CA family whose limit set it is. Let this limit set be reached in n steps.
- Take a doubly transitive point in some X_i. It must have a preimage with some f_m, which implies that some X_j must be mapped onto X_i by f_m. In fact, it is easy to see that f_m must also map X_j into X_i, and so $i = j$ by the assumption that there are no factoring relations.
- Since there are $k-1$ CA and k SFTs, some f_m must map both X_i and X_j onto themselves.

Proof scetch of properness of stable hierarchy: extracting a factoring relation from any smaller family

- No, assume that $X \in (k-1)$-LIM$_s$ and let $\{f_1, \ldots, f_{k-1}\}$ be a CA family whose limit set it is. Let this limit set be reached in n steps.
- Take a doubly transitive point in some X_i. It must have a preimage with some f_m, which implies that some X_j must be mapped onto X_i by f_m. In fact, it is easy to see that f_m must also map X_j into X_i, and so $i = j$ by the assumption that there are no factoring relations.
- Since there are $k-1$ CA and k SFTs, some f_m must map both X_i and X_j onto themselves.
- Now, using the point ${}^\infty a^\infty \subset X_i \cap X_j$ we easily find a point in $S_k^{\mathbb{Z}}$ which is not mapped to X in n steps.

Properness of unstable hierarchy and collapse in the transitive case

- The unstable hierarchy is handled similarly to the stable one, although we need some further tricks:

Properness of unstable hierarchy and collapse in the transitive case

- The unstable hierarchy is handled similarly to the stable one, although we need some further tricks:
  - We make the automata f_i unstable by taking a cartesian product with a sofic shift that can be implemented in an unstable way, but which is trivial enough not to break the argumentation.

Properness of unstable hierarchy and collapse in the transitive case

- The unstable hierarchy is handled similarly to the stable one, although we need some further tricks:
  - We make the automata f_i unstable by taking a cartesian product with a sofic shift that can be implemented in an unstable way, but which is trivial enough not to break the argumentation.
  - We use a slightly more complicated argument to find a point in the full shift that can never go to one of the X_i.

Properness of unstable hierarchy and collapse in the transitive case

- The unstable hierarchy is handled similarly to the stable one, although we need some further tricks:
  - We make the automata f_i unstable by taking a cartesian product with a sofic shift that can be implemented in an unstable way, but which is trivial enough not to break the argumentation.
  - We use a slightly more complicated argument to find a point in the full shift that can never go to one of the X_i.
- The collapse of the transitive hierarchies follows from doubly transitive points: such a point has a preimage, so one of the automata f_i is in fact surjective on the limit set X. This in fact means f_i has X as its limit set.

Properness of unstable hierarchy and collapse in the transitive case

- The unstable hierarchy is handled similarly to the stable one, although we need some further tricks:
  - We make the automata f_i unstable by taking a cartesian product with a sofic shift that can be implemented in an unstable way, but which is trivial enough not to break the argumentation.
  - We use a slightly more complicated argument to find a point in the full shift that can never go to one of the X_i.
- The collapse of the transitive hierarchies follows from doubly transitive points: such a point has a preimage, so one of the automata f_i is in fact surjective on the limit set X. This in fact means f_i has X as its limit set.
- This collapses ∞-LIM$_s$ into 1-LIM$_s$ and ∞-LIM$_u$ into 1-LIM$_x$.

- ∞-LIM$_s$ is closed under union.

- ∞-LIM_s is closed under union.
- If $X \in \infty$-LIM_s, then X is a finite union of subshifts in 2-LIM_s.

Open problems

- Are all limit sets of CA families in $\mathcal{PRO}$?

Open problems

- Are all limit sets of CA families in $\mathcal{PRO}$?
- Is the unstable hierarchy proper when restricted to transitive subshifts? Note that the growing part of the hierarchy would have to grow within 1-LIM$_s$, crazy right? However, the intersection 1-LIM$_s \cap$ 1-LIM$_u$ is not well-understood, and for a long time, it was thought to be empty. See [Limit sets of stable and unstable cellular automata] for an example.

Open problems

- Are all limit sets of CA families in $\mathcal{PRO}$?
- Is the unstable hierarchy proper when restricted to transitive subshifts? Note that the growing part of the hierarchy would have to grow within 1-LIM_s, crazy right? However, the intersection $1\text{-LIM}_s \cap 1\text{-LIM}_u$ is not well-understood, and for a long time, it was thought to be empty. See [Limit sets of stable and unstable cellular automata] for an example.
- Is $\infty\text{-LIM}_u$ closed under union?

Open problems

- Are all limit sets of CA families in $\mathcal{PRO}$?
- Is the unstable hierarchy proper when restricted to transitive subshifts? Note that the growing part of the hierarchy would have to grow within 1-LIM$_s$, crazy right? However, the intersection 1-LIM$_s$ $\cap$ 1-LIM$_u$ is not well-understood, and for a long time, it was thought to be empty. See [Limit sets of stable and unstable cellular automata] for an example.
- Is ∞-LIM$_u$ closed under union?
- If $X \in$ 2-LIM$_s$, is X a finite union of subshifts in 1-LIM$_s$?