A Universal Construction for Unique Sink Orientations

EuroCG 2024

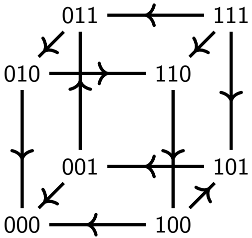
March 14, 2024

Michaela Borzechowski
Freie Universität Berlin

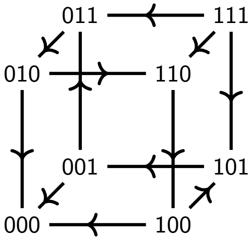
Joseph Doolittle
TU Graz

Simon Weber
ETH Zürich

What are Unique Sink Orientations?

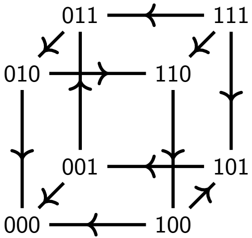


What are Unique Sink Orientations?



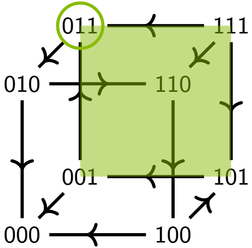
- Orientation of a k-dimensional hypercube

What are Unique Sink Orientations?



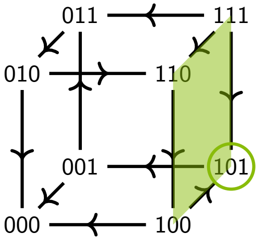
- Orientation of a k-dimensional hypercube
- s.t. each face has a unique sink.

What are Unique Sink Orientations?



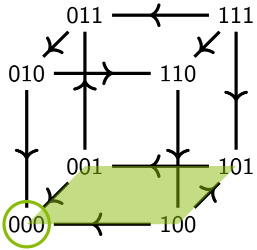
- Orientation of a k-dimensional hypercube
- s.t. each face has a unique sink.

What are Unique Sink Orientations?



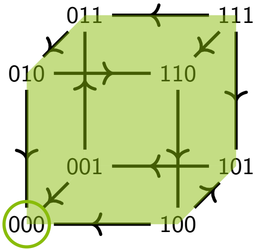
- Orientation of a k-dimensional hypercube
- s.t. each face has a unique sink.

What are Unique Sink Orientations?



- Orientation of a k-dimensional hypercube
- s.t. each face has a unique sink.

What are Unique Sink Orientations?



- Orientation of a k-dimensional hypercube
- s.t. each face has a unique sink.

About USOs

About USOs

- No polynomial time algorithm to find the sink.

About USOs

- No polynomial time algorithm to find the sink.
- Best known runtime: Fibonacci Seesaw with $O(1.61^k)$ [T. Szabó, Welzl 2001].

About USOs

- No polynomial time algorithm to find the sink.
- Best known runtime: Fibonacci Seesaw with $O(1.61^k)$ [T. Szabó, Welzl 2001].
- Deciding if orientation is USO is co-NP complete [Gärtner, Thomas 2015].

About USOs

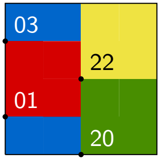
- No polynomial time algorithm to find the sink.
- Best known runtime: Fibonacci Seesaw with $O(1.61^k)$ [T. Szabó, Welzl 2001].
- Deciding if orientation is USO is co-NP complete [Gärtner, Thomas 2015].
- $2^{\Theta(2^k \log k)}$ USOs of the k-cube [Matoušek 2006].

About USOs

- No polynomial time algorithm to find the sink.
- Best known runtime: Fibonacci Seesaw with $O(1.61^k)$ [T. Szabó, Welzl 2001].
- Deciding if orientation is USO is co-NP complete [Gärtner, Thomas 2015].
- $2^{\Theta(2^k \log k)}$ USOs of the k-cube [Matoušek 2006].

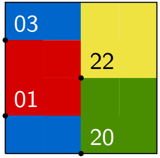
$\Rightarrow$ Construction of USOs.

$4\mathbb{Z}^k$-periodic tilings

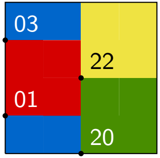


$4\mathbb{Z}^k$-periodic tilings

- Tesselation of k-dimensional space

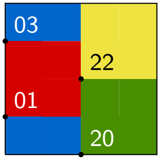


- Tesselation of k-dimensional space
- by periodic repetition of k-cubes of size $4 \times \cdots \times 4$



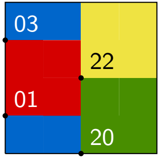
$4\mathbb{Z}^k$-periodic tilings

- Tesselation of k-dimensional space
- by periodic repetition of k-cubes of size $4 \times \cdots \times 4$
- tiled by 2^k tiles size $2 \times \cdots \times 2$



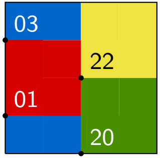
$4\mathbb{Z}^k$-periodic tilings

- Tesselation of k-dimensional space
- by periodic repetition of k-cubes of size $4 \times \cdots \times 4$
- tiled by 2^k tiles size $2 \times \cdots \times 2$
  positioned at integer coordinates $\{0, 1, 2, 3\}^k$



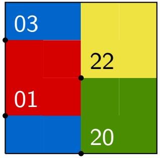
$4\mathbb{Z}^k$-periodic tilings

- Tesselation of k-dimensional space
- by periodic repetition of k-cubes of size $4 \times \cdots \times 4$
- tiled by 2^k tiles size $2 \times \cdots \times 2$
  positioned at integer coordinates $\{0, 1, 2, 3\}^k$
- s.t. each two tiles

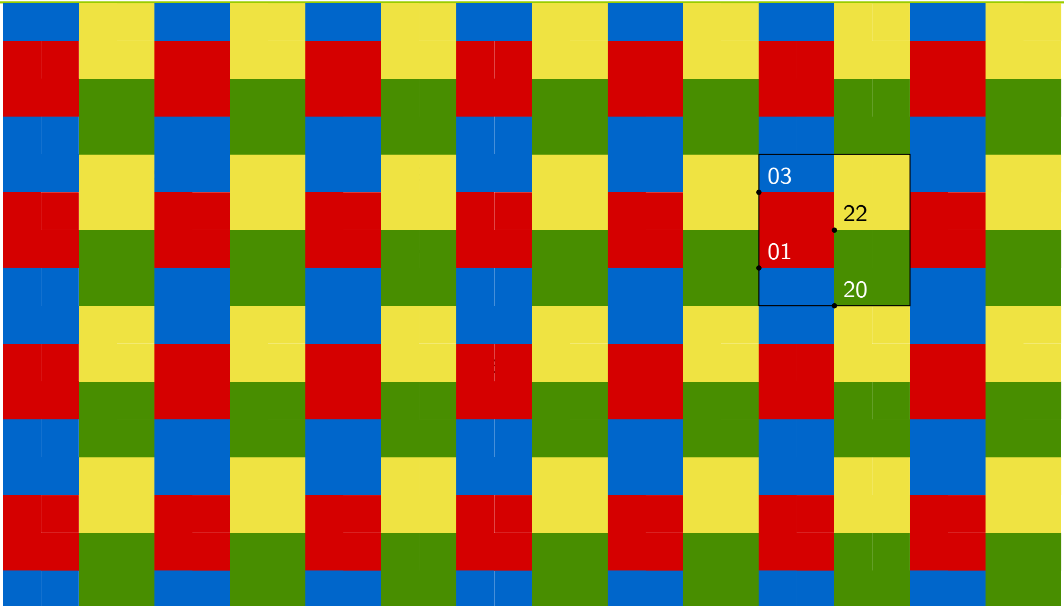


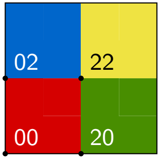
$4\mathbb{Z}^k$-periodic tilings

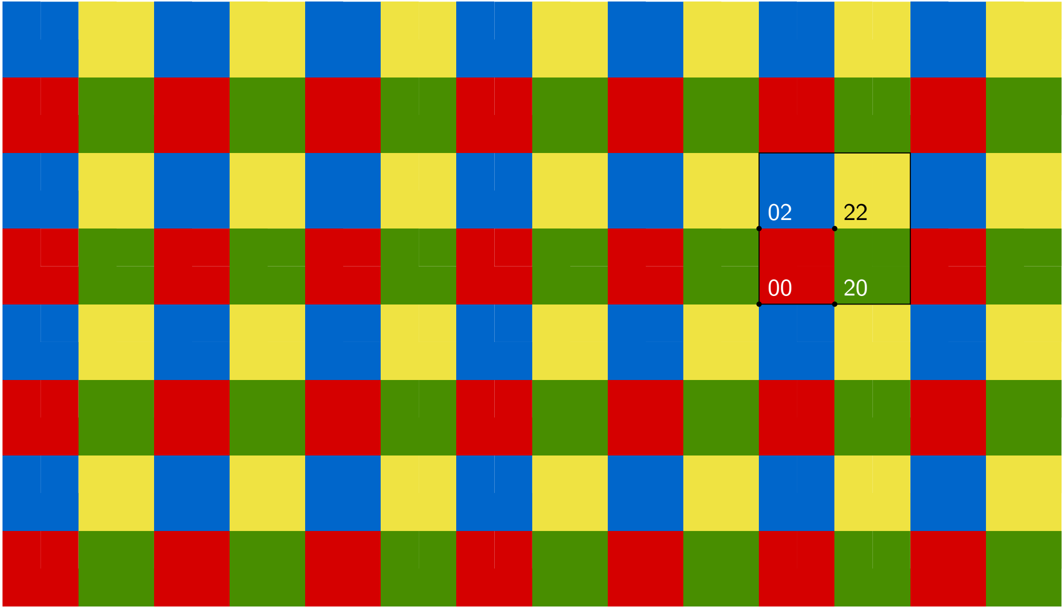
- Tesselation of k-dimensional space
- by periodic repetition of k-cubes of size $4 \times \cdots \times 4$
- tiled by 2^k tiles size $2 \times \cdots \times 2$
  positioned at integer coordinates $\{0, 1, 2, 3\}^k$
- s.t. each two tiles
  have some coordinate with difference 2.

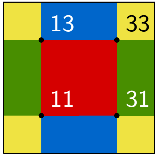


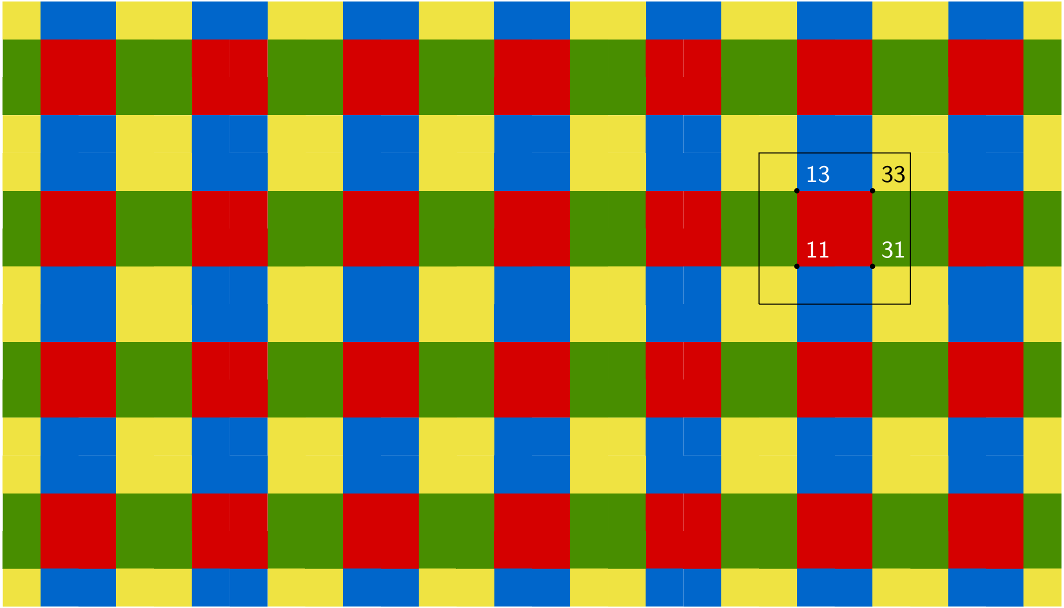
$4\mathbb{Z}^k$-periodic tilings



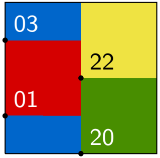

02
22
00
20

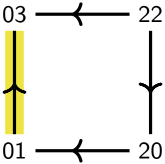

02
22
00
20


13
33
11
31

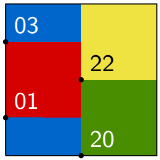

13
33
11
31

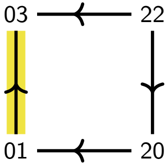
Bijection [Schurr 2004]





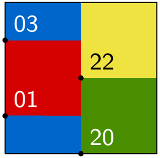
Bijection [Schurr 2004]

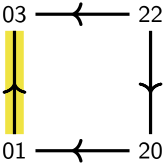




A string $s \in \{0, 1, 2, 3\}^k$ $\leftrightarrow$ A vertex of the k-cube and its orientation.

Bijection [Schurr 2004]

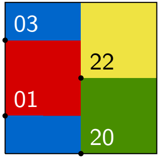


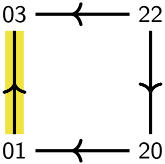


A string $s \in \{0,1,2,3\}^k$ $\leftrightarrow$ A vertex of the k-cube and its orientation.

$s_i = 0$ or $s_i = 1$ $\leftrightarrow$ s is in the *lower* i-facet of the cube.

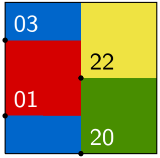
Bijection [Schurr 2004]

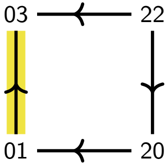




A string $s \in \{0,1,2,3\}^k$	$\leftrightarrow$	A vertex of the k-cube and its orientation.
$s_i = 0$ or $s_i = 1$	$\leftrightarrow$	s is in the *lower* i-facet of the cube.
$s_i = 2$ or $s_i = 3$	$\leftrightarrow$	s is in the *upper* i-facet of the cube.

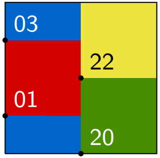
Bijection [Schurr 2004]

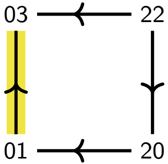




A string $s \in \{0, 1, 2, 3\}^k$	$\leftrightarrow$	A vertex of the k-cube and its orientation.
$s_i = 0$ or $s_i = 1$	$\leftrightarrow$	s is in the *lower* i-facet of the cube.
$s_i = 2$ or $s_i = 3$	$\leftrightarrow$	s is in the *upper* i-facet of the cube.
$s_i = 0$ or $s_i = 2$	$\leftrightarrow$	the edge from s in dimension i is *downwards* oriented.

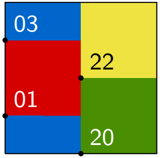
Bijection [Schurr 2004]

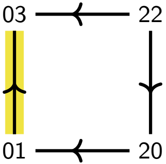




A string $s \in \{0, 1, 2, 3\}^k$	$\leftrightarrow$	A vertex of the k-cube and its orientation.
$s_i = 0$ or $s_i = 1$	$\leftrightarrow$	s is in the *lower* i-facet of the cube.
$s_i = 2$ or $s_i = 3$	$\leftrightarrow$	s is in the *upper* i-facet of the cube.
$s_i = 0$ or $s_i = 2$	$\leftrightarrow$	the edge from s in dimension i is *downwards* oriented.
$s_i = 1$ or $s_i = 3$	$\leftrightarrow$	the edge from s in dimension i is *upwards* oriented.

Bijection [Schurr 2004]

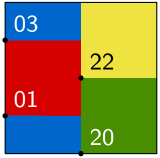


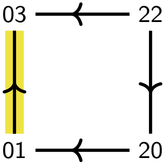


A string $s \in \{0, 1, 2, 3\}^k$	$\leftrightarrow$	A vertex of the k-cube and its orientation.
$s_i = 0$ or $s_i = 1$	$\leftrightarrow$	s is in the *lower* i-facet of the cube.
$s_i = 2$ or $s_i = 3$	$\leftrightarrow$	s is in the *upper* i-facet of the cube.
$s_i = 0$ or $s_i = 2$	$\leftrightarrow$	the edge from s in dimension i is *downwards* oriented.
$s_i = 1$ or $s_i = 3$	$\leftrightarrow$	the edge from s in dimension i is *upwards* oriented.

Downwards edge: 

Bijection [Schurr 2004]





A string $s \in \{0, 1, 2, 3\}^k$	$\leftrightarrow$	A vertex of the k-cube and its orientation.
$s_i = 0$ or $s_i = 1$	$\leftrightarrow$	s is in the *lower* i-facet of the cube.
$s_i = 2$ or $s_i = 3$	$\leftrightarrow$	s is in the *upper* i-facet of the cube.
$s_i = 0$ or $s_i = 2$	$\leftrightarrow$	the edge from s in dimension i is *downwards* oriented.
$s_i = 1$ or $s_i = 3$	$\leftrightarrow$	the edge from s in dimension i is *upwards* oriented.

Downwards edge: 

Upwards edge: 

Now we can apply

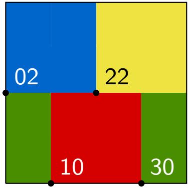
tilings stuff

to

USO's

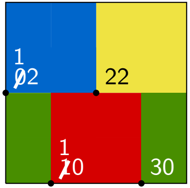
String Rewriting Rules in $4\mathbb{Z}^k$-periodic tilings [Lagarias, Shor]

- Replace in all strings s the digit in dimension h



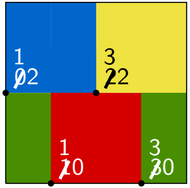
String Rewriting Rules in $4\mathbb{Z}^k$-periodic tilings [Lagarias, Shor]

- Replace in all strings s the digit in dimension h



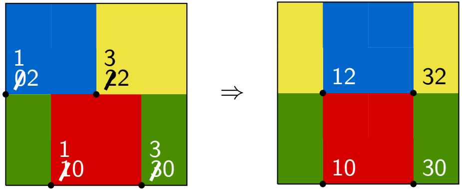
String Rewriting Rules in $4\mathbb{Z}^k$-periodic tilings [Lagarias, Shor]

- Replace in all strings s the digit in dimension h



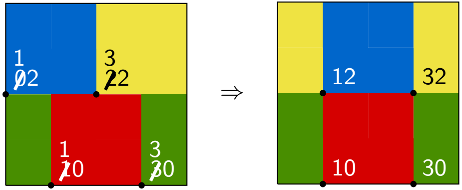
String Rewriting Rules in $4\mathbb{Z}^k$-periodic tilings [Lagarias, Shor]

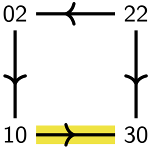
- Replace in all strings s the digit in dimension h



String Rewriting Rules in $4\mathbb{Z}^k$-periodic tilings [Lagarias, Shor]

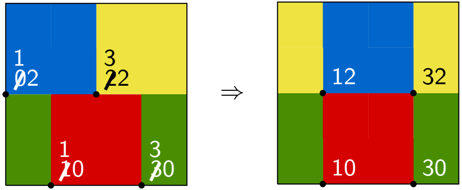
- Replace in all strings s the digit in dimension h

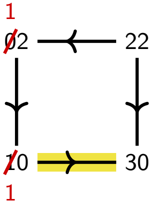




String Rewriting Rules in $4\mathbb{Z}^k$-periodic tilings [Lagarias, Shor]

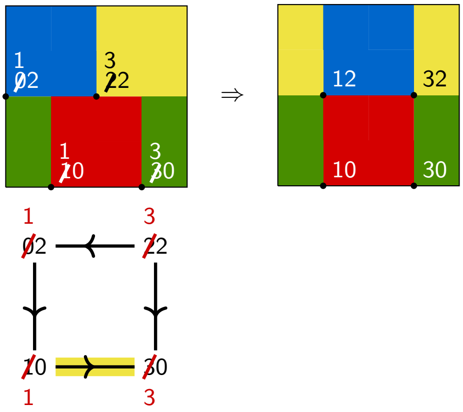
- Replace in all strings s the digit in dimension h





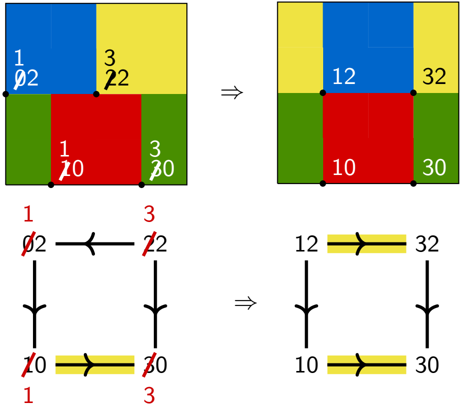
String Rewriting Rules in $4\mathbb{Z}^k$-periodic tilings [Lagarias, Shor]

- Replace in all strings s the digit in dimension h



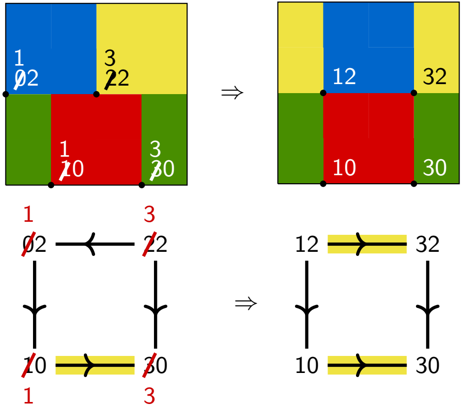
String Rewriting Rules in $4\mathbb{Z}^k$-periodic tilings [Lagarias, Shor]

- Replace in all strings s the digit in dimension h



String Rewriting Rules in $4\mathbb{Z}^k$-periodic tilings [Lagarias, Shor]

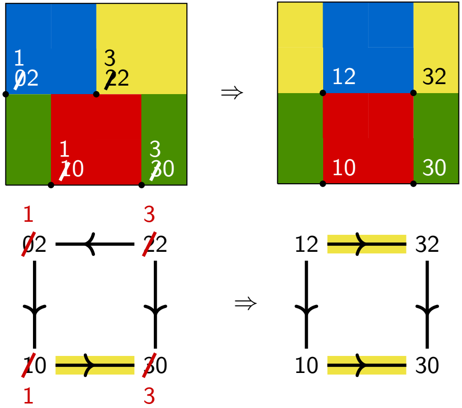
- Replace in all strings s the digit in dimension h



- Given $S^{(0)}, S^{(1)}, S^{(2)}, S^{(3)} \subseteq \{0, 1, 2, 3\}^d$

String Rewriting Rules in $4\mathbb{Z}^k$-periodic tilings [Lagarias, Shor]

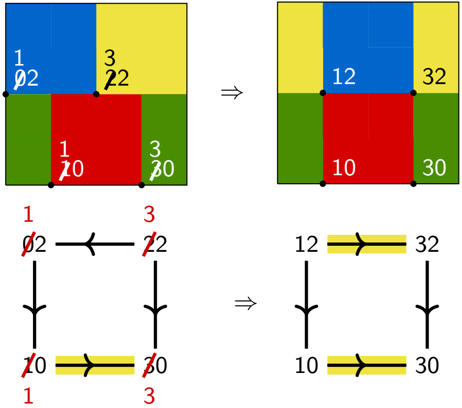
- Replace in all strings s the digit in dimension h



- Given $S^{(0)}, S^{(1)}, S^{(2)}, S^{(3)} \subseteq \{0,1,2,3\}^d$ replace s_h by content of set $S^{(s_h)}$

String Rewriting Rules in $4\mathbb{Z}^k$-periodic tilings [Lagarias, Shor]

- Replace in all strings s the digit in dimension h

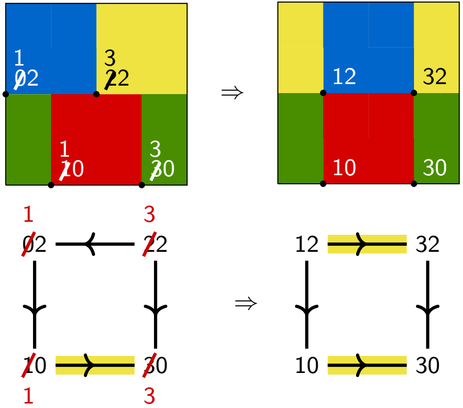


- Given $S^{(0)}, S^{(1)}, S^{(2)}, S^{(3)} \subseteq \{0,1,2,3\}^d$
  replace s_h by content of set $S^{(s_h)}$

$S^{(0)}$
1

String Rewriting Rules in $4\mathbb{Z}^k$-periodic tilings [Lagarias, Shor]

- Replace in all strings s the digit in dimension h

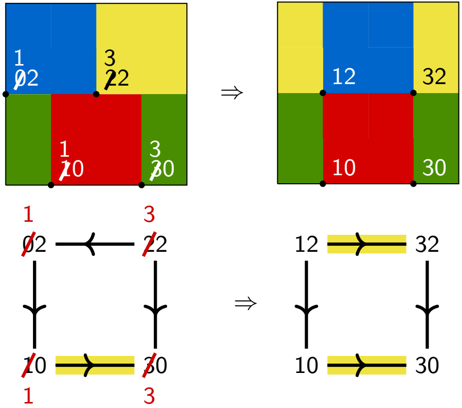


- Given $S^{(0)}, S^{(1)}, S^{(2)}, S^{(3)} \subseteq \{0,1,2,3\}^d$ replace s_h by content of set $S^{(s_h)}$

$S^{(0)}$	$S^{(1)}$
1	1

String Rewriting Rules in $4\mathbb{Z}^k$-periodic tilings [Lagarias, Shor]

- Replace in all strings s the digit in dimension h

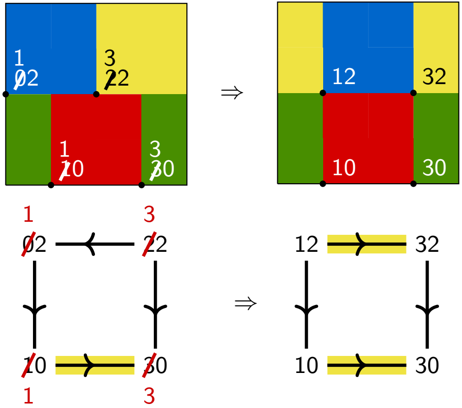


- Given $S^{(0)}, S^{(1)}, S^{(2)}, S^{(3)} \subseteq \{0,1,2,3\}^d$ replace s_h by content of set $S^{(s_h)}$

$S^{(0)}$	$S^{(1)}$	$S^{(2)}$
1	1	3

String Rewriting Rules in $4\mathbb{Z}^k$-periodic tilings [Lagarias, Shor]

- Replace in all strings s the digit in dimension h



- Given $S^{(0)}, S^{(1)}, S^{(2)}, S^{(3)} \subseteq \{0,1,2,3\}^d$ replace s_h by content of set $S^{(s_h)}$

$S^{(0)}$	$S^{(1)}$	$S^{(2)}$	$S^{(3)}$
1	1	3	3

String Rewriting Rules

The sets $S^{(0)}, S^{(1)}, S^{(2)}, S^{(3)} \subseteq \{0, 1, 2, 3\}^d$ are a valid *String Rewriting Rule* if

String Rewriting Rules

The sets $S^{(0)}, S^{(1)}, S^{(2)}, S^{(3)} \subseteq \{0, 1, 2, 3\}^d$ are a valid *String Rewriting Rule* if

(i) $\left(S^{(0)} \cup S^{(2)}\right)$ defines a d-dimensional USO (a $4\mathbb{Z}^d$-periodic tiling) and $S^{(0)} \cap S^{(2)} = \emptyset$, and

String Rewriting Rules

The sets $S^{(0)}, S^{(1)}, S^{(2)}, S^{(3)} \subseteq \{0,1,2,3\}^d$ are a valid *String Rewriting Rule* if

(i) $\left(S^{(0)} \cup S^{(2)}\right)$ defines a d-dimensional USO (a $4\mathbb{Z}^d$-periodic tiling) and $S^{(0)} \cap S^{(2)} = \emptyset$, and

(ii) $\left(S^{(1)} \cup S^{(3)}\right)$ defines a d-dimensional USO (a $4\mathbb{Z}^d$-periodic tiling) and $S^{(1)} \cap S^{(3)} = \emptyset$.

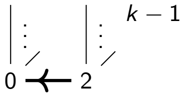
String Rewriting Rules

The sets $S^{(0)}, S^{(1)}, S^{(2)}, S^{(3)} \subseteq \{0, 1, 2, 3\}^d$ are a valid *String Rewriting Rule* if

(i) $\left(S^{(0)} \cup S^{(2)}\right)$ defines a d-dimensional USO (a $4\mathbb{Z}^d$-periodic tiling) and $S^{(0)} \cap S^{(2)} = \emptyset$, and

(ii) $\left(S^{(1)} \cup S^{(3)}\right)$ defines a d-dimensional USO (a $4\mathbb{Z}^d$-periodic tiling) and $S^{(1)} \cap S^{(3)} = \emptyset$.

Applied to a k-dimensional USO:



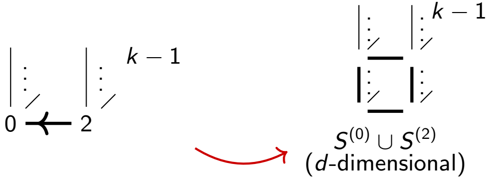
String Rewriting Rules

The sets $S^{(0)}, S^{(1)}, S^{(2)}, S^{(3)} \subseteq \{0,1,2,3\}^d$ are a valid *String Rewriting Rule* if

(i) $\left(S^{(0)} \cup S^{(2)}\right)$ defines a d-dimensional USO (a $4\mathbb{Z}^d$-periodic tiling) and $S^{(0)} \cap S^{(2)} = \emptyset$, and

(ii) $\left(S^{(1)} \cup S^{(3)}\right)$ defines a d-dimensional USO (a $4\mathbb{Z}^d$-periodic tiling) and $S^{(1)} \cap S^{(3)} = \emptyset$.

Applied to a k-dimensional USO:



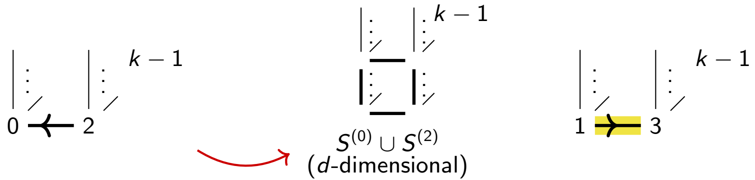
String Rewriting Rules

The sets $S^{(0)}, S^{(1)}, S^{(2)}, S^{(3)} \subseteq \{0,1,2,3\}^d$ are a valid *String Rewriting Rule* if

(i) $\left(S^{(0)} \cup S^{(2)}\right)$ defines a d-dimensional USO (a $4\mathbb{Z}^d$-periodic tiling) and $S^{(0)} \cap S^{(2)} = \emptyset$, and

(ii) $\left(S^{(1)} \cup S^{(3)}\right)$ defines a d-dimensional USO (a $4\mathbb{Z}^d$-periodic tiling) and $S^{(1)} \cap S^{(3)} = \emptyset$.

Applied to a k-dimensional USO:



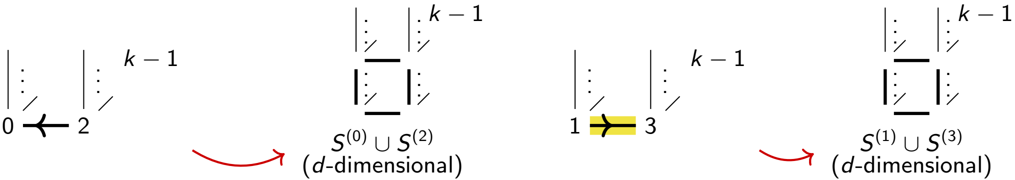
String Rewriting Rules

The sets $S^{(0)}, S^{(1)}, S^{(2)}, S^{(3)} \subseteq \{0,1,2,3\}^d$ are a valid *String Rewriting Rule* if

(i) $\left(S^{(0)} \cup S^{(2)}\right)$ defines a d-dimensional USO (a $4\mathbb{Z}^d$-periodic tiling) and $S^{(0)} \cap S^{(2)} = \emptyset$, and

(ii) $\left(S^{(1)} \cup S^{(3)}\right)$ defines a d-dimensional USO (a $4\mathbb{Z}^d$-periodic tiling) and $S^{(1)} \cap S^{(3)} = \emptyset$.

Applied to a k-dimensional USO:



Example

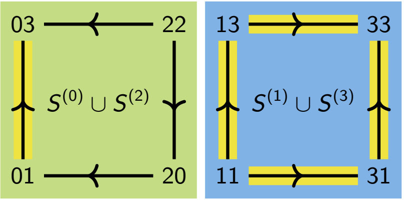
$d = 2$

$S^{(0)}$	$S^{(1)}$	$S^{(2)}$	$S^{(3)}$
01	11	03	33
	31	20	13
		22	

Example

$d = 2$

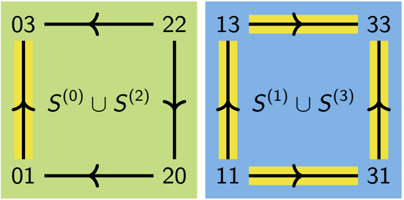
$S^{(0)}$	$S^{(1)}$	$S^{(2)}$	$S^{(3)}$
01	11	03	33
	31	20	13
		22	

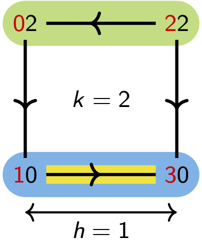


Example

$d = 2$

$S^{(0)}$	$S^{(1)}$	$S^{(2)}$	$S^{(3)}$
01	11	03	33
	31	20	13
		22	

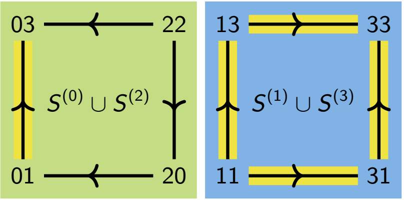


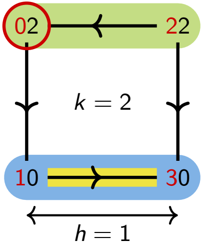


Example

$d = 2$

$S^{(0)}$	$S^{(1)}$	$S^{(2)}$	$S^{(3)}$
01	11	03	33
	31	20	13
		22	





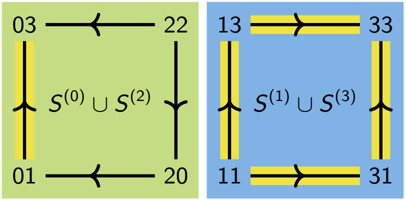
Apply rewriting rule

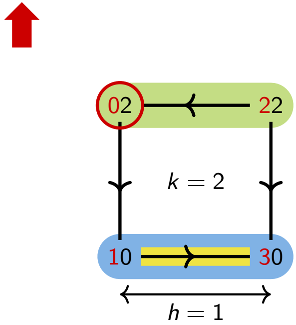
$\Rightarrow$

Example

$d = 2$

$S^{(0)}$	$S^{(1)}$	$S^{(2)}$	$S^{(3)}$
01	11	03	33
	31	20	13
		22	





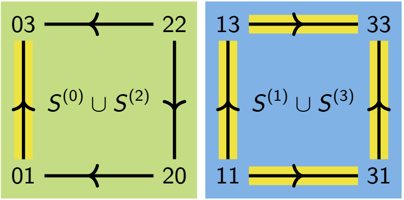
Apply rewriting rule

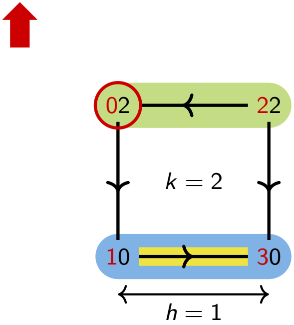
$\Rightarrow$

Example

$d = 2$

$S^{(0)}$	$S^{(1)}$	$S^{(2)}$	$S^{(3)}$
01	11	03	33
	31	20	13
		22	





Apply rewriting rule

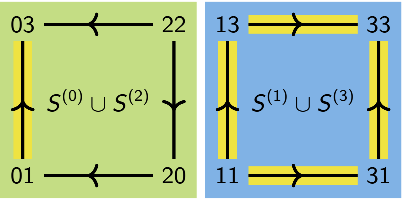
$\Rightarrow$

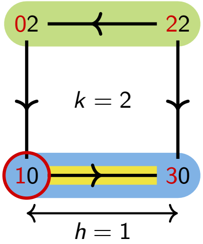
012

Example

$$d = 2$$

$S^{(0)}$	$S^{(1)}$	$S^{(2)}$	$S^{(3)}$
01	11	03	33
	31	20	13
		22	





Apply rewriting rule

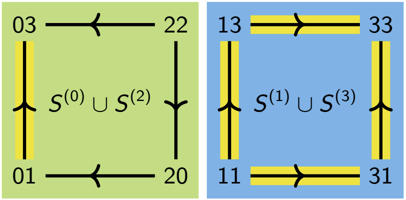
$\Rightarrow$

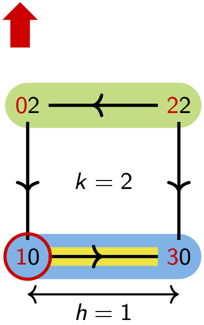
012

Example

$d = 2$

$S^{(0)}$	$S^{(1)}$	$S^{(2)}$	$S^{(3)}$
01	11	03	33
	31	20	13
		22	





Apply rewriting rule

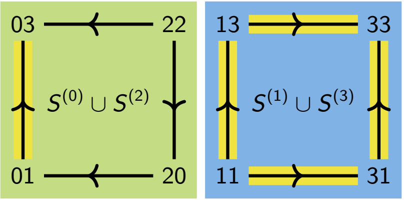
$\Rightarrow$

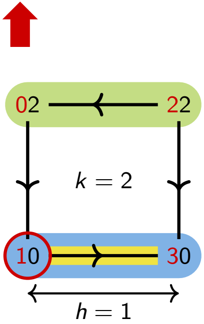
012

Example

$d = 2$

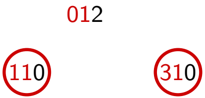
$S^{(0)}$	$S^{(1)}$	$S^{(2)}$	$S^{(3)}$
01	11	03	33
	31	20	13
		22	





Apply rewriting rule

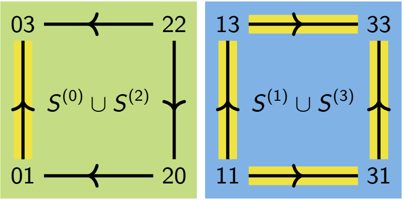
$\Rightarrow$

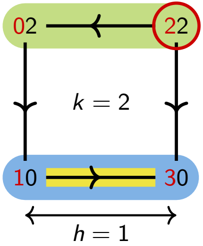


Example

$d = 2$

$S^{(0)}$	$S^{(1)}$	$S^{(2)}$	$S^{(3)}$
01	11	03	33
	31	20	13
		22	





Apply rewriting rule

$\Rightarrow$

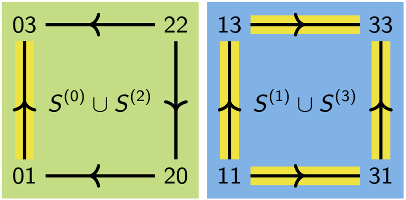
012

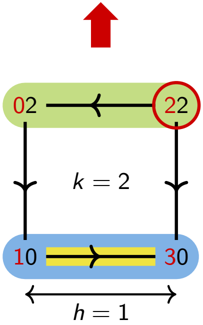
110 310

Example

$d = 2$

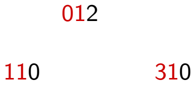
$S^{(0)}$	$S^{(1)}$	$S^{(2)}$	$S^{(3)}$
01	11	03	33
	31	20	13
		22	





Apply rewriting rule

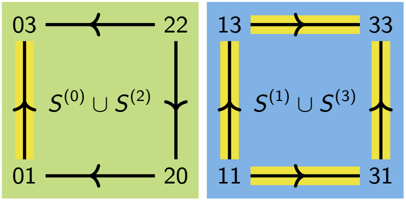
$\Rightarrow$

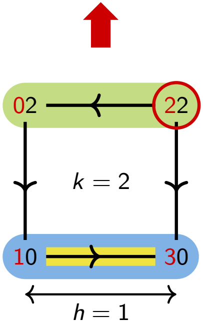


Example

$d = 2$

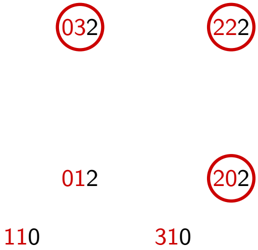
$S^{(0)}$	$S^{(1)}$	$S^{(2)}$	$S^{(3)}$
01	11	03	33
	31	20	13
		22	





Apply rewriting rule

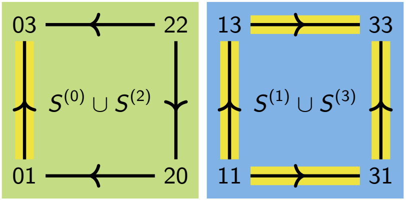
$\Rightarrow$

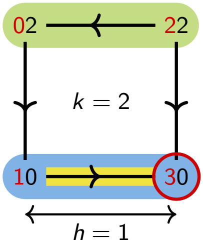


Example

$d = 2$

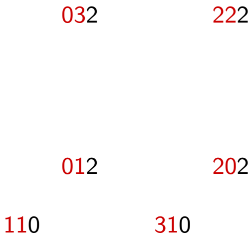
$S^{(0)}$	$S^{(1)}$	$S^{(2)}$	$S^{(3)}$
01	11	03	33
	31	20	13
		22	





Apply rewriting rule

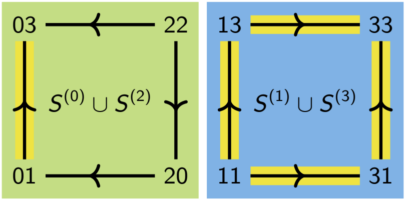
$\Rightarrow$

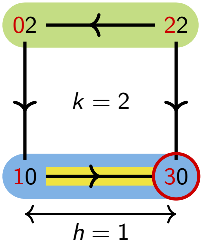


Example

$d = 2$

$S^{(0)}$	$S^{(1)}$	$S^{(2)}$	$S^{(3)}$
01	11	03	33
	31	20	13
		22	





Apply rewriting rule

$\Rightarrow$

032 222

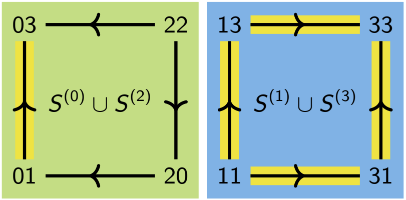
012 202

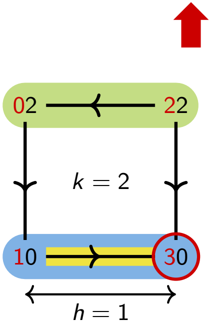
110 310

Example

$d = 2$

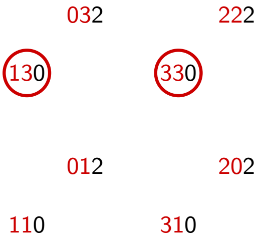
$S^{(0)}$	$S^{(1)}$	$S^{(2)}$	$S^{(3)}$
01	11	03	33
	31	20	13
		22	





Apply rewriting rule

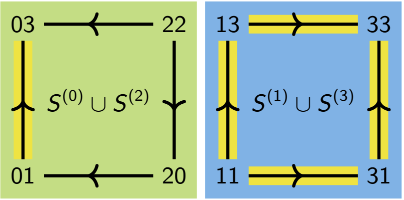
$\Rightarrow$

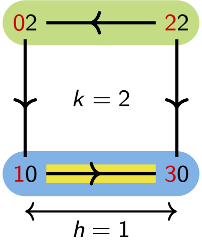


Example

$d = 2$

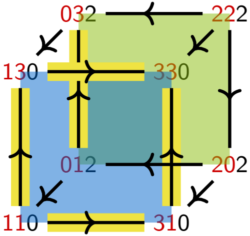
$S^{(0)}$	$S^{(1)}$	$S^{(2)}$	$S^{(3)}$
01	11	03	33
	31	20	13
		22	





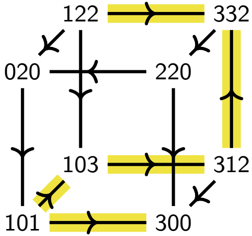
Apply rewriting rule

$\Rightarrow$



But we can't create all USOs from lower dimensional USOs this way :(

But we can't create all USOs from lower dimensional USOs this way :(



String Rewriting Rule

The sets $S^{(0)}, S^{(1)}, S^{(2)}, S^{(3)} \subseteq \{0,1,2,3\}^d$ are a valid *String Rewriting Rule* if

(i) $\left(S^{(0)} \cup S^{(2)}\right)$ defines a d-dimensional USO and $S^{(0)} \cap S^{(2)} = \emptyset$, and

(ii) $\left(S^{(1)} \cup S^{(3)}\right)$ defines a d-dimensional USO and $S^{(1)} \cap S^{(3)} = \emptyset$.

Generalized Rewriting Rule

Assign label $\in [i]$ to each vertex.

The sets $S^{(0)}, S^{(1)}, S^{(2)}, S^{(3)} \subseteq \{0, 1, 2, 3\}^d$ are a valid *String Rewriting Rule* if

(i) $\left(S^{(0)} \cup S^{(2)}\right)$ defines a d-dimensional USO and $S^{(0)} \cap S^{(2)} = \emptyset$, and

(ii) $\left(S^{(1)} \cup S^{(3)}\right)$ defines a d-dimensional USO and $S^{(1)} \cap S^{(3)} = \emptyset$.

Generalized Rewriting Rule

Assign label $\in [i]$ to each vertex.

The sets $S^{(0)}_{1,\dots,i}$, $S^{(1)}_{1,\dots,i}$, $S^{(2)}_{1,\dots,i}$, $S^{(3)}_{1,\dots,i} \subseteq \{0,1,2,3\}^d$ are a valid *String Rewriting Rule* if

(i) $\left(S^{(0)}_j \cup S^{(2)}_{j'}\right)$ defines a d-dimensional USO and $S^{(0)}_j \cap S^{(2)}_{j'} = \emptyset$ for all pairs $j, j' \in [i]$, and

(ii) $\left(S^{(1)}_j \cup S^{(3)}_{j'}\right)$ defines a d-dimensional USO and $S^{(1)}_j \cap S^{(3)}_{j'} = \emptyset$ for all pairs $j, j' \in [i]$.

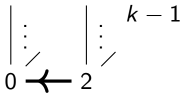
Generalized Rewriting Rule

Assign label $\in [i]$ to each vertex.

The sets $S^{(0)}_{1,\dots,i}$, $S^{(1)}_{1,\dots,i}$, $S^{(2)}_{1,\dots,i}$, $S^{(3)}_{1,\dots,i} \subseteq \{0,1,2,3\}^d$ are a valid *String Rewriting Rule* if

(i) $\left(S^{(0)}_j \cup S^{(2)}_{j'}\right)$ defines a d-dimensional USO and $S^{(0)}_j \cap S^{(2)}_{j'} = \emptyset$ for all pairs $j, j' \in [i]$, and

(ii) $\left(S^{(1)}_j \cup S^{(3)}_{j'}\right)$ defines a d-dimensional USO and $S^{(1)}_j \cap S^{(3)}_{j'} = \emptyset$ for all pairs $j, j' \in [i]$.



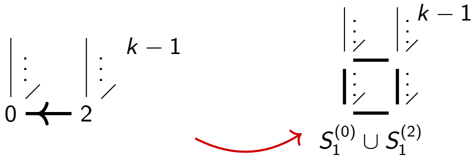
Generalized Rewriting Rule

Assign label $\in [i]$ to each vertex.

The sets $S^{(0)}_{1,\ldots,i}$, $S^{(1)}_{1,\ldots,i}$, $S^{(2)}_{1,\ldots,i}$, $S^{(3)}_{1,\ldots,i} \subseteq \{0,1,2,3\}^d$ are a valid *String Rewriting Rule* if

(i) $\left(S^{(0)}_j \cup S^{(2)}_{j'}\right)$ defines a d-dimensional USO and $S^{(0)}_j \cap S^{(2)}_{j'} = \emptyset$ for all pairs $j, j' \in [i]$, and

(ii) $\left(S^{(1)}_j \cup S^{(3)}_{j'}\right)$ defines a d-dimensional USO and $S^{(1)}_j \cap S^{(3)}_{j'} = \emptyset$ for all pairs $j, j' \in [i]$.



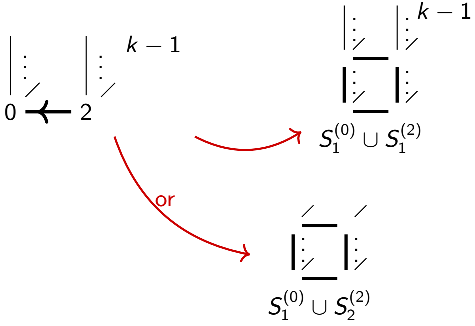
Generalized Rewriting Rule

Assign label $\in [i]$ to each vertex.

The sets $S^{(0)}_{1,\ldots,i}$, $S^{(1)}_{1,\ldots,i}$, $S^{(2)}_{1,\ldots,i}$, $S^{(3)}_{1,\ldots,i} \subseteq \{0,1,2,3\}^d$ are a valid *String Rewriting Rule* if

(i) $\left(S^{(0)}_j \cup S^{(2)}_{j'}\right)$ defines a d-dimensional USO and $S^{(0)}_j \cap S^{(2)}_{j'} = \emptyset$ for all pairs $j, j' \in [i]$, and

(ii) $\left(S^{(1)}_j \cup S^{(3)}_{j'}\right)$ defines a d-dimensional USO and $S^{(1)}_j \cap S^{(3)}_{j'} = \emptyset$ for all pairs $j, j' \in [i]$.



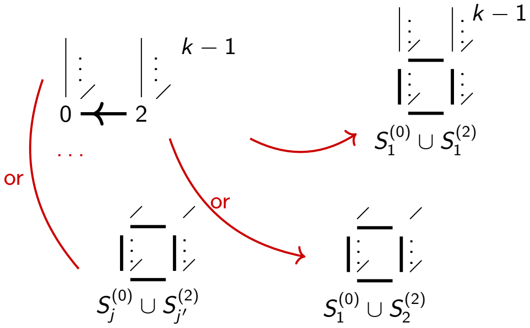
Generalized Rewriting Rule

Assign label $\in [i]$ to each vertex.

The sets $S^{(0)}_{1,\dots,i}$, $S^{(1)}_{1,\dots,i}$, $S^{(2)}_{1,\dots,i}$, $S^{(3)}_{1,\dots,i} \subseteq \{0,1,2,3\}^d$ are a valid *String Rewriting Rule* if

(i) $\left(S^{(0)}_j \cup S^{(2)}_{j'}\right)$ defines a d-dimensional USO and $S^{(0)}_j \cap S^{(2)}_{j'} = \emptyset$ for all pairs $j, j' \in [i]$, and

(ii) $\left(S^{(1)}_j \cup S^{(3)}_{j'}\right)$ defines a d-dimensional USO and $S^{(1)}_j \cap S^{(3)}_{j'} = \emptyset$ for all pairs $j, j' \in [i]$.



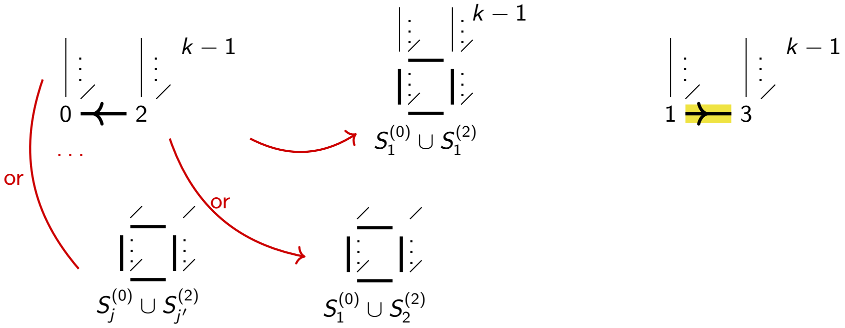
Generalized Rewriting Rule

Assign label $\in [i]$ to each vertex.

The sets $S^{(0)}_{1,\ldots,i}$, $S^{(1)}_{1,\ldots,i}$, $S^{(2)}_{1,\ldots,i}$, $S^{(3)}_{1,\ldots,i} \subseteq \{0,1,2,3\}^d$ are a valid *String Rewriting Rule* if

(i) $\left(S^{(0)}_j \cup S^{(2)}_{j'}\right)$ defines a d-dimensional USO and $S^{(0)}_j \cap S^{(2)}_{j'} = \emptyset$ for all pairs $j, j' \in [i]$, and

(ii) $\left(S^{(1)}_j \cup S^{(3)}_{j'}\right)$ defines a d-dimensional USO and $S^{(1)}_j \cap S^{(3)}_{j'} = \emptyset$ for all pairs $j, j' \in [i]$.



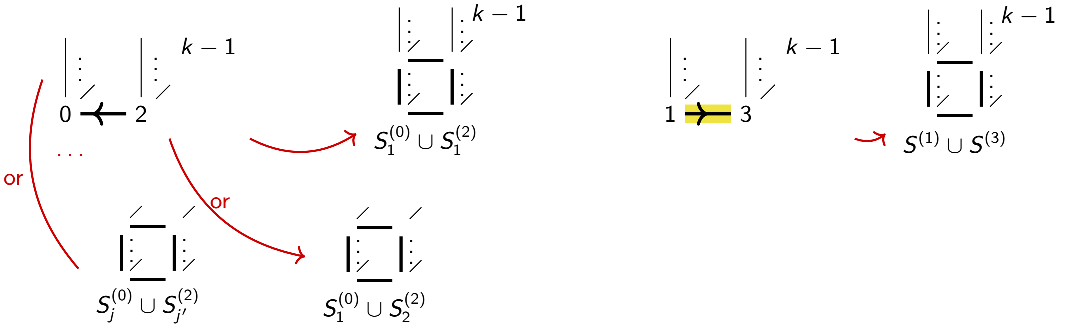
Generalized Rewriting Rule

Assign label $\in [i]$ to each vertex.

The sets $S^{(0)}_{1,\ldots,i}$, $S^{(1)}_{1,\ldots,i}$, $S^{(2)}_{1,\ldots,i}$, $S^{(3)}_{1,\ldots,i} \subseteq \{0,1,2,3\}^d$ are a valid *String Rewriting Rule* if

(i) $\left(S^{(0)}_j \cup S^{(2)}_{j'}\right)$ defines a d-dimensional USO and $S^{(0)}_j \cap S^{(2)}_{j'} = \emptyset$ for all pairs $j, j' \in [i]$, and

(ii) $\left(S^{(1)}_j \cup S^{(3)}_{j'}\right)$ defines a d-dimensional USO and $S^{(1)}_j \cap S^{(3)}_{j'} = \emptyset$ for all pairs $j, j' \in [i]$.



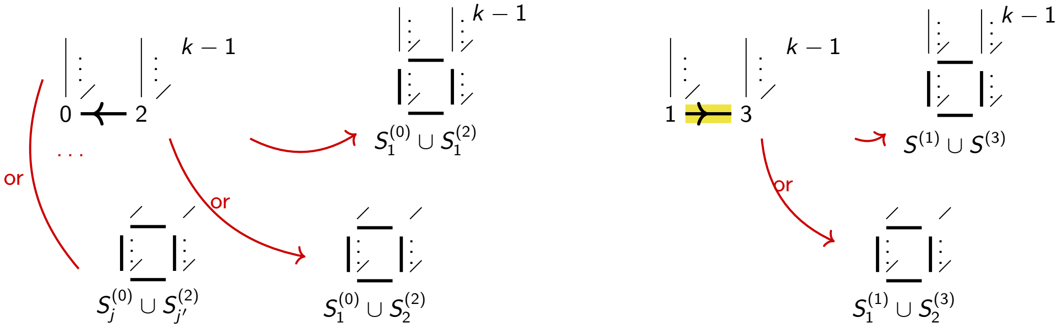
Generalized Rewriting Rule

Assign label $\in [i]$ to each vertex.

The sets $S^{(0)}_{1,\ldots,i}$, $S^{(1)}_{1,\ldots,i}$, $S^{(2)}_{1,\ldots,i}$, $S^{(3)}_{1,\ldots,i} \subseteq \{0,1,2,3\}^d$ are a valid *String Rewriting Rule* if

(i) $\left(S^{(0)}_j \cup S^{(2)}_{j'}\right)$ defines a d-dimensional USO and $S^{(0)}_j \cap S^{(2)}_{j'} = \emptyset$ for all pairs $j, j' \in [i]$, and

(ii) $\left(S^{(1)}_j \cup S^{(3)}_{j'}\right)$ defines a d-dimensional USO and $S^{(1)}_j \cap S^{(3)}_{j'} = \emptyset$ for all pairs $j, j' \in [i]$.



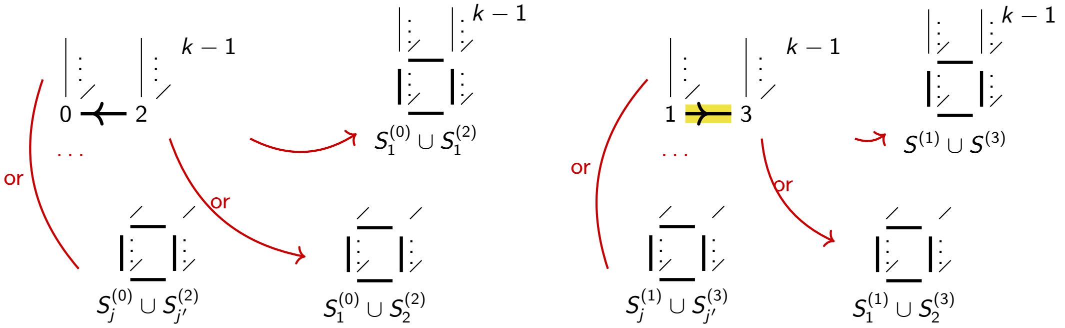
Generalized Rewriting Rule

Assign label $\in [i]$ to each vertex.

The sets $S^{(0)}_{1,\ldots,i}$, $S^{(1)}_{1,\ldots,i}$, $S^{(2)}_{1,\ldots,i}$, $S^{(3)}_{1,\ldots,i} \subseteq \{0,1,2,3\}^d$ are a valid *String Rewriting Rule* if

(i) $\left(S^{(0)}_j \cup S^{(2)}_{j'}\right)$ defines a d-dimensional USO and $S^{(0)}_j \cap S^{(2)}_{j'} = \emptyset$ for all pairs $j, j' \in [i]$, and

(ii) $\left(S^{(1)}_j \cup S^{(3)}_{j'}\right)$ defines a d-dimensional USO and $S^{(1)}_j \cap S^{(3)}_{j'} = \emptyset$ for all pairs $j, j' \in [i]$.



Universality

Theorem

For each k-dimensional USO K,

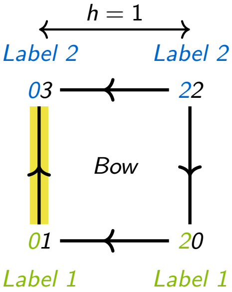
Universality

Theorem

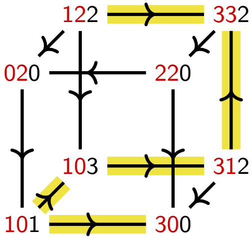
For each k-dimensional USO K,
there exist a generalized rewriting rule $S_1^{(0)}, S_2^{(0)}, S_1^{(2)}, S_2^{(2)}$ *of* $(k-1)$*-dimensional USOs*

Theorem

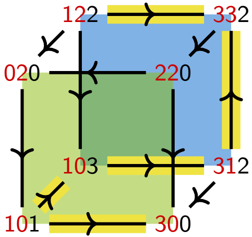
For each k-dimensional USO K,
there exist a generalized rewriting rule $S_1^{(0)}, S_2^{(0)}, S_1^{(2)}, S_2^{(2)}$ of $(k-1)$-dimensional USOs s.t. K is equal to this rewriting rule applied to the bow in dimension 1:



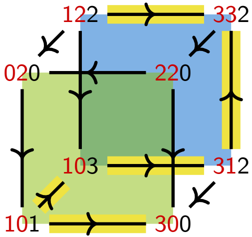
Proof Sketch



Proof Sketch

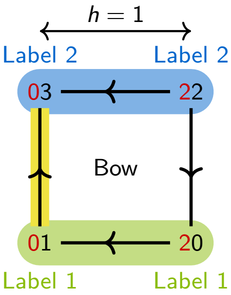


Proof Sketch

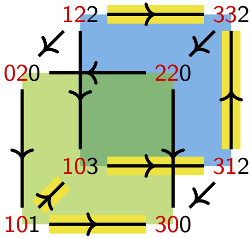


Apply rewriting rule to dimension h

$\Leftarrow$



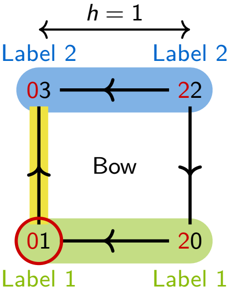
Proof Sketch



Apply rewriting rule

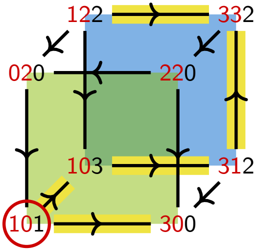
to dimension h

$\Leftarrow$



$S_1^{(0)}$: Green facet, *upwards* edge to blue

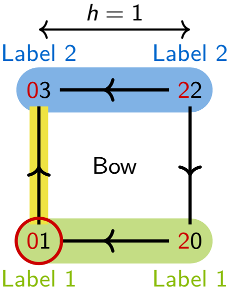
Proof Sketch



Apply rewriting rule

to dimension h

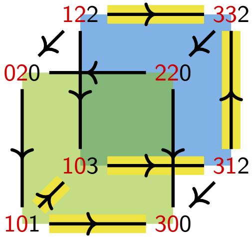
$\Leftarrow$



$S_1^{(0)}$	
10	

$S_1^{(0)}$: Green facet, *upwards* edge to blue

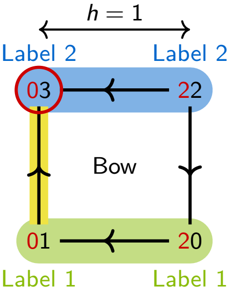
Proof Sketch



Apply rewriting rule

to dimension h

$\Leftarrow$

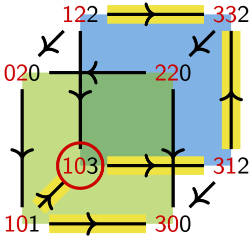


$S_1^{(0)}$	
10	

$S_1^{(0)}$: Green facet, *upwards* edge to blue

$S_2^{(0)}$: Blue facet, *upwards* edge to green

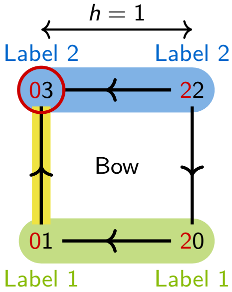
Proof Sketch



Apply rewriting rule

to dimension h

$\Leftarrow$

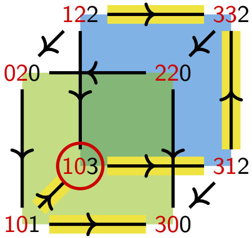


$S_1^{(0)}$	
10	

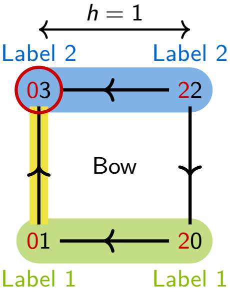
$S_1^{(0)}$: Green facet, *upwards* edge to blue

$S_2^{(0)}$: Blue facet, *upwards* edge to green

Proof Sketch



Apply rewriting rule
to dimension h
$\Leftarrow$

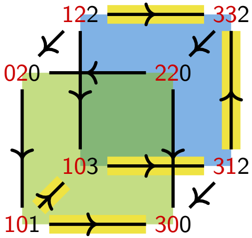


$S_1^{(0)}$	$S_2^{(0)}$
10	10

$S_1^{(0)}$: Green facet, *upwards* edge to blue

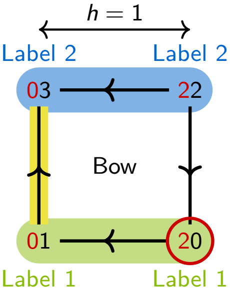
$S_2^{(0)}$: Blue facet, *upwards* edge to green

Proof Sketch



Apply rewriting rule to dimension h

$\Leftarrow$



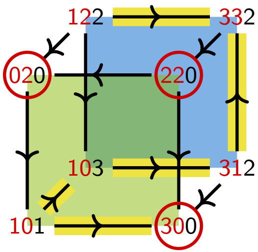
$S_1^{(0)}$	$S_2^{(0)}$
10	10

$S_1^{(0)}$: Green facet, *upwards* edge to blue

$S_2^{(0)}$: Blue facet, *upwards* edge to green

$S_1^{(2)}$: Green facet, *downwards* edge to blue

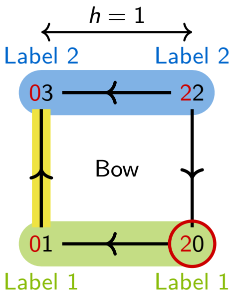
Proof Sketch



Apply rewriting rule

to dimension h

$\Leftarrow$



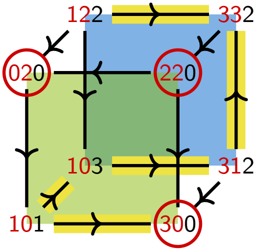
$S_1^{(0)}$	$S_2^{(0)}$
10	10

$S_1^{(0)}$: Green facet, *upwards* edge to blue

$S_2^{(0)}$: Blue facet, *upwards* edge to green

$S_1^{(2)}$: Green facet, *downwards* edge to blue

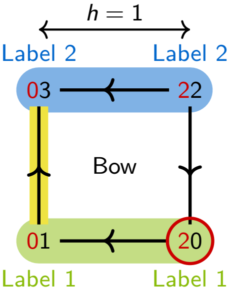
Proof Sketch



Apply rewriting rule

to dimension h

$\Leftarrow$



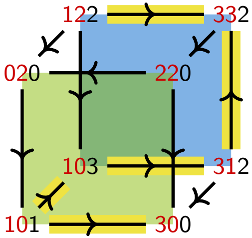
$S_1^{(0)}$	$S_2^{(0)}$	$S_1^{(2)}$
10	10	02
		22
		30

$S_1^{(0)}$: Green facet, *upwards* edge to blue

$S_2^{(0)}$: Blue facet, *upwards* edge to green

$S_1^{(2)}$: Green facet, *downwards* edge to blue

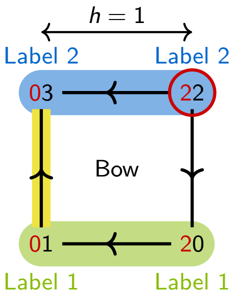
Proof Sketch



Apply rewriting rule

to dimension h

$\Leftarrow$



$S_1^{(0)}$	$S_2^{(0)}$	$S_1^{(2)}$
10	10	02
		22
		30

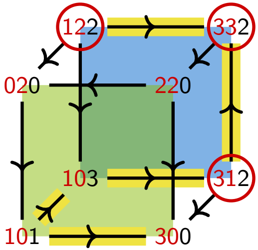
$S_1^{(0)}$: Green facet, *upwards* edge to blue

$S_2^{(0)}$: Blue facet, *upwards* edge to green

$S_1^{(2)}$: Green facet, *downwards* edge to blue

$S_2^{(2)}$: Blue facet, *downwards* edge to blue

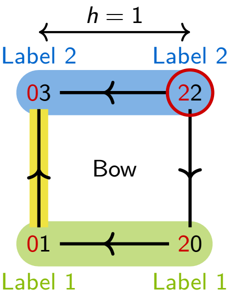
Proof Sketch



Apply rewriting rule

to dimension h

$\Leftarrow$



$S_1^{(0)}$	$S_2^{(0)}$	$S_1^{(2)}$
10	10	02
		22
		30

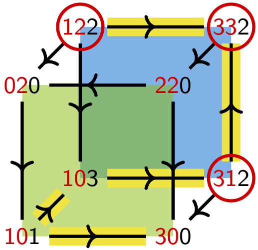
$S_1^{(0)}$: Green facet, *upwards* edge to blue

$S_2^{(0)}$: Blue facet, *upwards* edge to green

$S_1^{(2)}$: Green facet, *downwards* edge to blue

$S_2^{(2)}$: Blue facet, *downwards* edge to blue

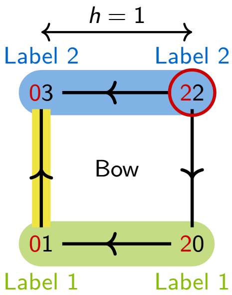
Proof Sketch



Apply rewriting rule

to dimension h

$\Leftarrow$



$S_1^{(0)}$	$S_2^{(0)}$	$S_1^{(2)}$	$S_2^{(2)}$
10	10	02	12
		22	33
		30	31

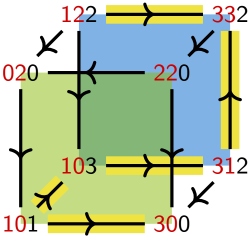
$S_1^{(0)}$: Green facet, *upwards* edge to blue

$S_2^{(0)}$: Blue facet, *upwards* edge to green

$S_1^{(2)}$: Green facet, *downwards* edge to blue

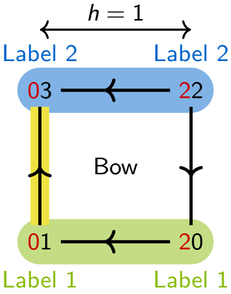
$S_2^{(2)}$: Blue facet, *downwards* edge to blue

Proof Sketch



Apply rewriting rule to dimension h

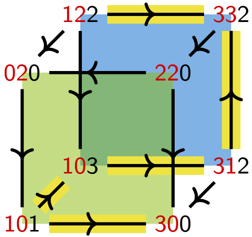
$\Leftarrow$



$S_1^{(0)}$	$S_2^{(0)}$	$S_1^{(2)}$	$S_2^{(2)}$
10	10	02	12
		22	33
		30	31

- $S_1^{(0)}$: Green facet, *upwards* edge to blue
- $S_2^{(0)}$: Blue facet, *upwards* edge to green
- $S_1^{(2)}$: Green facet, *downwards* edge to blue
- $S_2^{(2)}$: Blue facet, *downwards* edge to blue

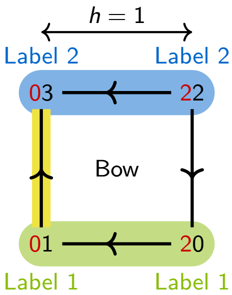
Proof Sketch



Apply rewriting rule

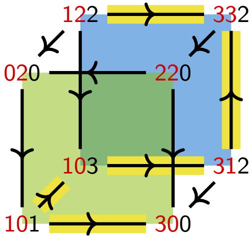
to dimension h

$\Leftarrow$



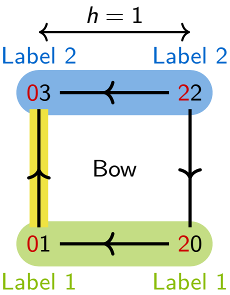
$S_1^{(0)}$	$S_2^{(0)}$	$S_1^{(2)}$	$S_2^{(2)}$	$S_1^{(1)}$	$S_1^{(3)}$	$S_2^{(1)}$	$S_2^{(3)}$
10	10	02	12				
		22	33				
		30	31				

Proof Sketch

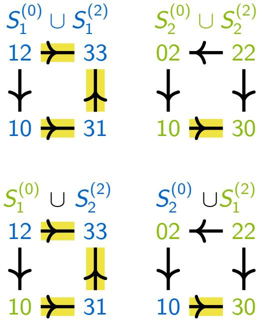


Apply rewriting rule to dimension h

$\Leftarrow$



Check sets:



$S_1^{(0)}$	$S_2^{(0)}$	$S_1^{(2)}$	$S_2^{(2)}$	$S_1^{(1)}$	$S_1^{(3)}$	$S_2^{(1)}$	$S_2^{(3)}$
10	10	02	12				
		22	33				
		30	31				

Conclusion

Conclusion

- Universal construction

Conclusion

- Universal construction
- Unfortunately, we have to check all pairs for sets for USOness.

Conclusion

- Universal construction
- Unfortunately, we have to check all pairs for sets for USOness.

$\Rightarrow$ This construction method is not very efficient.

Conclusion

- Universal construction
- Unfortunately, we have to check all pairs for sets for USOness.

$\Rightarrow$ This construction method is not very efficient.

- We cannot yet efficiently enumerate all USOs.

Conclusion

- Universal construction
- Unfortunately, we have to check all pairs for sets for USOness.

$\Rightarrow$ This construction method is not very efficient.

- We cannot yet efficiently enumerate all USOs.

But . . .

Conclusion

- Universal construction
- Unfortunately, we have to check all pairs for sets for USOness.

$\Rightarrow$ This construction method is not very efficient.

- We cannot yet efficiently enumerate all USOs.

But . . .

- Each rewriting rule is one valid operation on USOs.

Conclusion

- Universal construction
- Unfortunately, we have to check all pairs for sets for USOness.

$\Rightarrow$ This construction method is not very efficient.

- We cannot yet efficiently enumerate all USOs.

But . . .

- Each rewriting rule is one valid operation on USOs.

$\Rightarrow$ Other interesting operations?

Conclusion

- Universal construction
- Unfortunately, we have to check all pairs for sets for USOness.

$\Rightarrow$ This construction method is not very efficient.

- We cannot yet efficiently enumerate all USOs.

But . . .

- Each rewriting rule is one valid operation on USOs.

$\Rightarrow$ Other interesting operations?

- Multidimensional Rewriting? [Mackey]

Conclusion

- Universal construction
- Unfortunately, we have to check all pairs for sets for USOness.

$\Rightarrow$ This construction method is not very efficient.

- We cannot yet efficiently enumerate all USOs.

But ...

- Each rewriting rule is one valid operation on USOs.

$\Rightarrow$ Other interesting operations?

- Multidimensional Rewriting? [Mackey]
  - Replace pairs of coordinates together.

Conclusion

- Universal construction
- Unfortunately, we have to check all pairs for sets for USOness.

$\Rightarrow$ This construction method is not very efficient.

- We cannot yet efficiently enumerate all USOs.

But ...

- Each rewriting rule is one valid operation on USOs.

$\Rightarrow$ Other interesting operations?

- Multidimensional Rewriting? [Mackey]
  - Replace pairs of coordinates together.
  - Requires $4^2 = 16$ lists $S^{(00)}, S^{(01)}, \ldots, S^{(32)}, S^{(33)}$.

Conclusion

- Universal construction
- Unfortunately, we have to check all pairs for sets for USOness.

$\Rightarrow$ This construction method is not very efficient.

- We cannot yet efficiently enumerate all USOs.

But . . .

- Each rewriting rule is one valid operation on USOs.

$\Rightarrow$ Other interesting operations?

- Multidimensional Rewriting? [Mackey]
  - Replace pairs of coordinates together.
  - Requires $4^2 = 16$ lists $S^{(00)}, S^{(01)}, \ldots, S^{(32)}, S^{(33)}$.