Bootstrap Percolation Thresholds in Plane Tilings using Regular Polygons

Daniel W. Cranston
Virginia Commonwealth University
dcranston@vcu.edu

Joint with Neal Bushaw
Slides available on my webpage

VCU Discrete Math
21 February 2018

Two Simple Examples

Two Simple Examples

Bootstrap Percolation: Some faces start infected.

Two Simple Examples

Bootstrap Percolation: Some faces start infected. Infected faces stay infected.

Two Simple Examples

Bootstrap Percolation: Some faces start infected. Infected faces stay infected. Uninfected faces with at least two infected neighbors become infected.

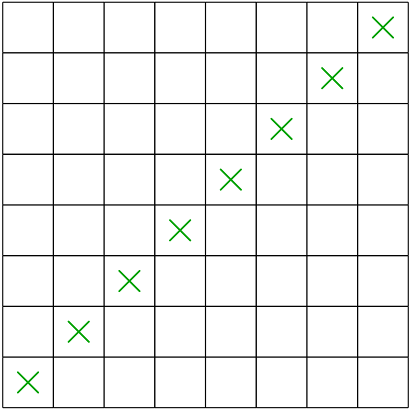
Two Simple Examples

Bootstrap Percolation: Some faces start infected. Infected faces stay infected. Uninfected faces with at least two infected neighbors become infected. *Does the whole graph become infected?*

Two Simple Examples

Bootstrap Percolation: Some faces start infected. Infected faces stay infected. Uninfected faces with at least two infected neighbors become infected. *Does the whole graph become infected?*

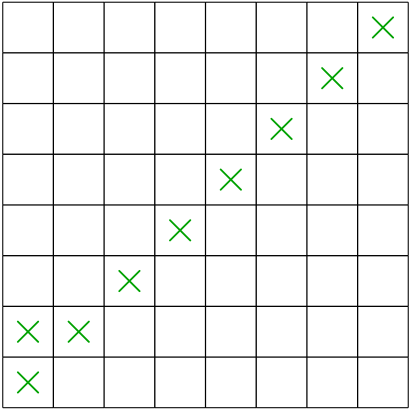
Ex:



Two Simple Examples

Bootstrap Percolation: Some faces start infected. Infected faces stay infected. Uninfected faces with at least two infected neighbors become infected. *Does the whole graph become infected?*

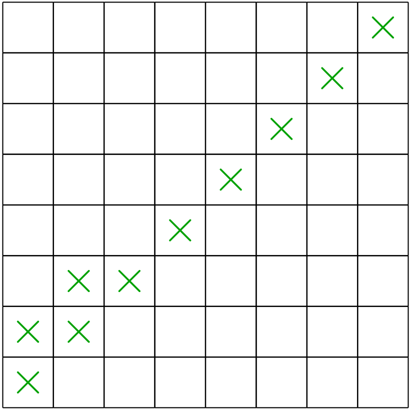
Ex:



Two Simple Examples

Bootstrap Percolation: Some faces start infected. Infected faces stay infected. Uninfected faces with at least two infected neighbors become infected. *Does the whole graph become infected?*

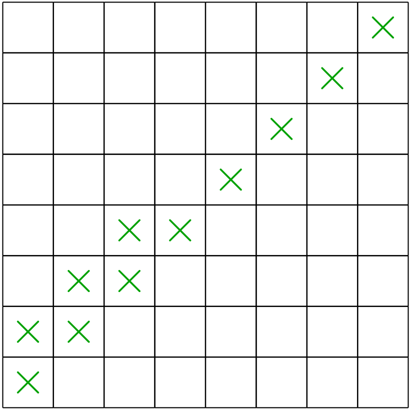
Ex:



Two Simple Examples

Bootstrap Percolation: Some faces start infected. Infected faces stay infected. Uninfected faces with at least two infected neighbors become infected. *Does the whole graph become infected?*

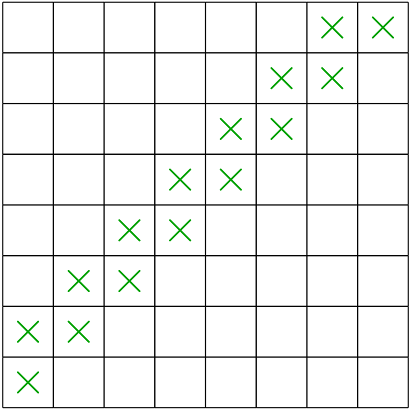
Ex:



Two Simple Examples

Bootstrap Percolation: Some faces start infected. Infected faces stay infected. Uninfected faces with at least two infected neighbors become infected. *Does the whole graph become infected?*

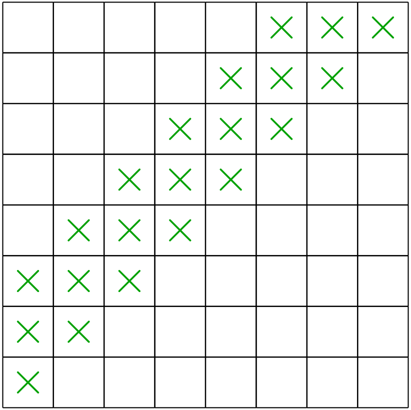
Ex:



Two Simple Examples

Bootstrap Percolation: Some faces start infected. Infected faces stay infected. Uninfected faces with at least two infected neighbors become infected. *Does the whole graph become infected?*

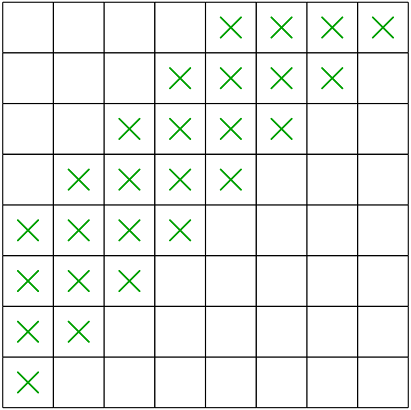
Ex:



Two Simple Examples

Bootstrap Percolation: Some faces start infected. Infected faces stay infected. Uninfected faces with at least two infected neighbors become infected. *Does the whole graph become infected?*

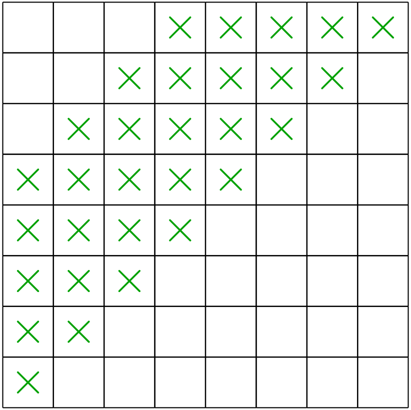
Ex:



Two Simple Examples

Bootstrap Percolation: Some faces start infected. Infected faces stay infected. Uninfected faces with at least two infected neighbors become infected. *Does the whole graph become infected?*

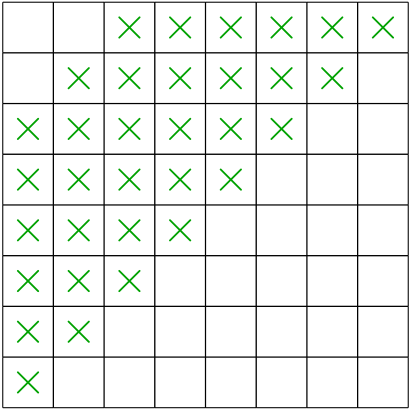
Ex:



Two Simple Examples

Bootstrap Percolation: Some faces start infected. Infected faces stay infected. Uninfected faces with at least two infected neighbors become infected. *Does the whole graph become infected?*

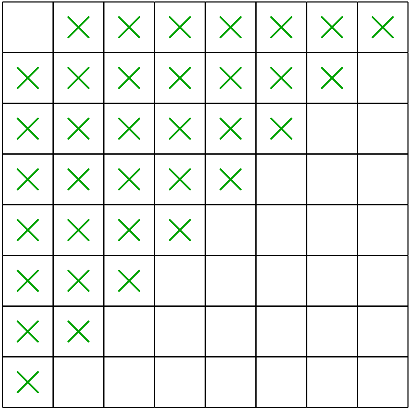
Ex:



Two Simple Examples

Bootstrap Percolation: Some faces start infected. Infected faces stay infected. Uninfected faces with at least two infected neighbors become infected. *Does the whole graph become infected?*

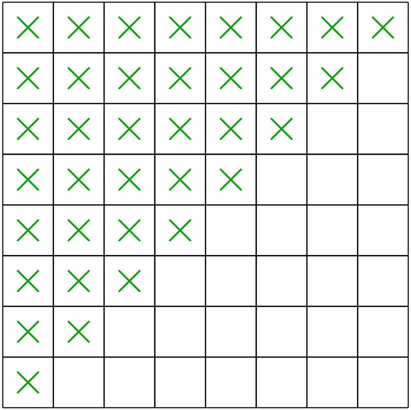
Ex:



Two Simple Examples

Bootstrap Percolation: Some faces start infected. Infected faces stay infected. Uninfected faces with at least two infected neighbors become infected. *Does the whole graph become infected?*

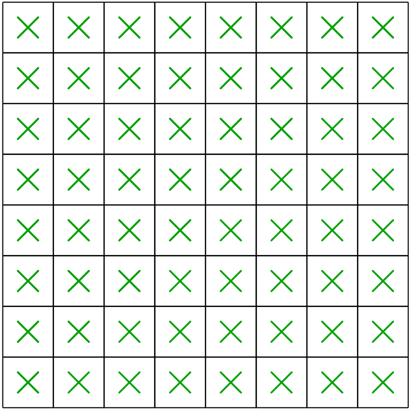
Ex:



Two Simple Examples

Bootstrap Percolation: Some faces start infected. Infected faces stay infected. Uninfected faces with at least two infected neighbors become infected. *Does the whole graph become infected?*

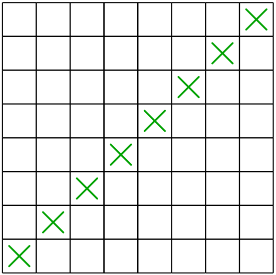
Ex:

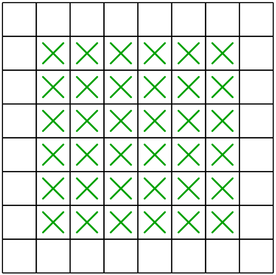


Two Simple Examples

Bootstrap Percolation: Some faces start infected. Infected faces stay infected. Uninfected faces with at least two infected neighbors become infected. *Does the whole graph become infected?*

Ex:

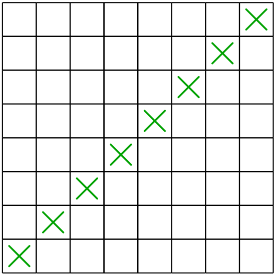




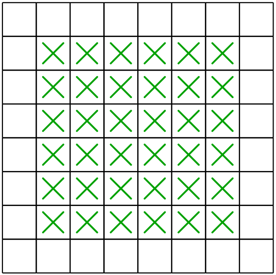
Two Simple Examples

Bootstrap Percolation: Some faces start infected. Infected faces stay infected. Uninfected faces with at least two infected neighbors become infected. *Does the whole graph become infected?*

Ex:



Yes.



No.

The k-bootstrap Model

The k-bootstrap Model

We make a few key changes to our game.

1. The initially infected faces are picked randomly.

The k-bootstrap Model

We make a few key changes to our game.

1. The initially infected faces are picked randomly. Each face is infected with prob. p, independently. This is a p-random set.

The k-bootstrap Model

We make a few key changes to our game.

1. The initially infected faces are picked randomly. Each face is infected with prob. p, independently. This is a p-random set.
2. Number of infected nbrs needed to infect a healthy face is k.

The k-bootstrap Model

We make a few key changes to our game.

1. The initially infected faces are picked randomly. Each face is infected with prob. p, independently. This is a p-random set.
2. Number of infected nbrs needed to infect a healthy face is k.
3. We mainly consider *infinite* graphs.

The k-bootstrap Model

We make a few key changes to our game.

1. The initially infected faces are picked randomly. Each face is infected with prob. p, independently. This is a p-random set.
2. Number of infected nbrs needed to infect a healthy face is k.
3. We mainly consider *infinite* graphs.

The k-bootstrap Model: Fix a plane graph G, a p-random set $\mathcal{I}$ of initially infected faces, and an integer k.

The k-bootstrap Model

We make a few key changes to our game.

1. The initially infected faces are picked randomly. Each face is infected with prob. p, independently. This is a p-random set.
2. Number of infected nbrs needed to infect a healthy face is k.
3. We mainly consider *infinite* graphs.

The k-bootstrap Model: Fix a plane graph G, a p-random set $\mathcal{I}$ of initially infected faces, and an integer k. If a healthy face, f, has at least k infected neighbors, then f becomes infected.

The k-bootstrap Model

We make a few key changes to our game.

1. The initially infected faces are picked randomly. Each face is infected with prob. p, independently. This is a p-random set.
2. Number of infected nbrs needed to infect a healthy face is k.
3. We mainly consider *infinite* graphs.

The k-bootstrap Model: Fix a plane graph G, a p-random set $\mathcal{I}$ of initially infected faces, and an integer k. If a healthy face, f, has at least k infected neighbors, then f becomes infected. The percolation threshold of G is the largest k such that G eventually becomes completely infected with prob $\geq \frac{1}{2}$ (since $\mathcal{I}$ is random).

The k-bootstrap Model

We make a few key changes to our game.

1. The initially infected faces are picked randomly. Each face is infected with prob. p, independently. This is a p-random set.
2. Number of infected nbrs needed to infect a healthy face is k.
3. We mainly consider *infinite* graphs.

The k-bootstrap Model: Fix a plane graph G, a p-random set $\mathcal{I}$ of initially infected faces, and an integer k. If a healthy face, f, has at least k infected neighbors, then f becomes infected. The percolation threshold of G is the largest k such that G eventually becomes completely infected with prob $\geq \frac{1}{2}$ (since $\mathcal{I}$ is random).

Warmup: In the 1-bootstrap model if $\mathcal{I} \neq \emptyset$, then $\mathcal{I}$ percolates.

The k-bootstrap Model

We make a few key changes to our game.

1. The initially infected faces are picked randomly. Each face is infected with prob. p, independently. This is a p-random set.
2. Number of infected nbrs needed to infect a healthy face is k.
3. We mainly consider *infinite* graphs.

The k-bootstrap Model: Fix a plane graph G, a p-random set $\mathcal{I}$ of initially infected faces, and an integer k. If a healthy face, f, has at least k infected neighbors, then f becomes infected. The percolation threshold of G is the largest k such that G eventually becomes completely infected with prob $\geq \frac{1}{2}$ (since $\mathcal{I}$ is random).

Warmup: In the 1-bootstrap model if $\mathcal{I} \neq \emptyset$, then $\mathcal{I}$ percolates.

Pf: Say that $f_0 \in \mathcal{I}$.

The k-bootstrap Model

We make a few key changes to our game.

1. The initially infected faces are picked randomly. Each face is infected with prob. p, independently. This is a p-random set.
2. Number of infected nbrs needed to infect a healthy face is k.
3. We mainly consider *infinite* graphs.

The k-bootstrap Model: Fix a plane graph G, a p-random set $\mathcal{I}$ of initially infected faces, and an integer k. If a healthy face, f, has at least k infected neighbors, then f becomes infected. The percolation threshold of G is the largest k such that G eventually becomes completely infected with prob $\geq \frac{1}{2}$ (since $\mathcal{I}$ is random).

Warmup: In the 1-bootstrap model if $\mathcal{I} \neq \emptyset$, then $\mathcal{I}$ percolates.

Pf: Say that $f_0 \in \mathcal{I}$. By induction, we show that each face within distance t of f_0 becomes infected (for all t).

The k-bootstrap Model

We make a few key changes to our game.

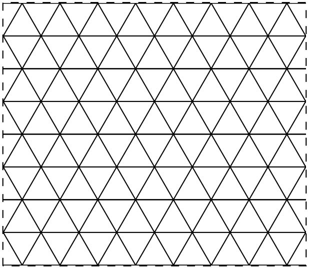
1. The initially infected faces are picked randomly. Each face is infected with prob. p, independently. This is a p-random set.
2. Number of infected nbrs needed to infect a healthy face is k.
3. We mainly consider *infinite* graphs.

The k-bootstrap Model: Fix a plane graph G, a p-random set $\mathcal{I}$ of initially infected faces, and an integer k. If a healthy face, f, has at least k infected neighbors, then f becomes infected. The percolation threshold of G is the largest k such that G eventually becomes completely infected with prob $\geq \frac{1}{2}$ (since $\mathcal{I}$ is random).

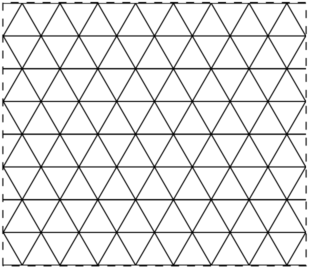
Warmup: In the 1-bootstrap model if $\mathcal{I} \neq \emptyset$, then $\mathcal{I}$ percolates.

Pf: Say that $f_0 \in \mathcal{I}$. By induction, we show that each face within distance t of f_0 becomes infected (for all t). So $\mathcal{I}$ percolates. ■

The Triangular Lattice

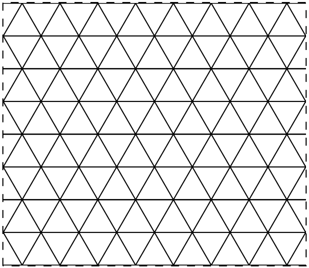


The Triangular Lattice



Lem 1: Let G be the triangular lattice and $\mathcal{I}$ be a p-random set, with $0 < p$. In the 1-bootstrap model, $\mathcal{I}$ percolates with prob. 1.

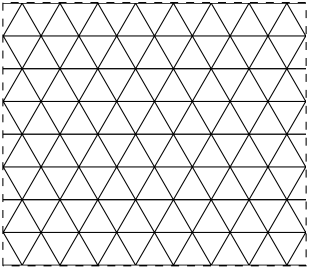
The Triangular Lattice



Lem 1: Let G be the triangular lattice and $\mathcal{I}$ be a p-random set, with $0 < p$. In the 1-bootstrap model, $\mathcal{I}$ percolates with prob. 1.

Pf: By the warmup, we must show that $\Pr[\mathcal{I} \neq \emptyset] = 1$.

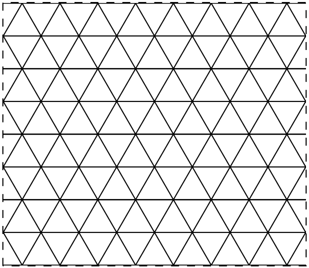
The Triangular Lattice



Lem 1: Let G be the triangular lattice and $\mathcal{I}$ be a p-random set, with $0 < p$. In the 1-bootstrap model, $\mathcal{I}$ percolates with prob. 1.

Pf: By the warmup, we must show that $\Pr[\mathcal{I} \neq \emptyset] = 1$. For $t \geq 0$, let $\mathcal{A}_t$ be a set of t faces.

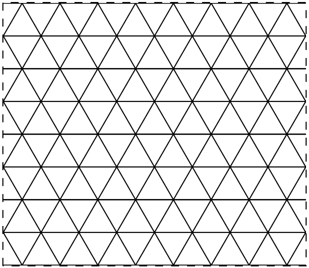
The Triangular Lattice



Lem 1: Let G be the triangular lattice and $\mathcal{I}$ be a p-random set, with $0 < p$. In the 1-bootstrap model, $\mathcal{I}$ percolates with prob. 1.

Pf: By the warmup, we must show that $\Pr[\mathcal{I} \neq \emptyset] = 1$. For $t \geq 0$, let $\mathcal{A}_t$ be a set of t faces. Now $\Pr[\mathcal{I} = \emptyset] \leq \Pr[(I \cap A_t) = \emptyset]$ $= (1 - p)^t$.

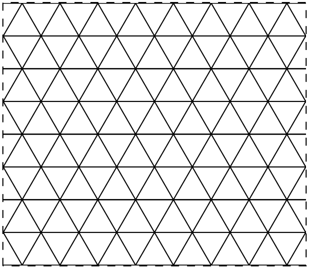
The Triangular Lattice



Lem 1: Let G be the triangular lattice and $\mathcal{I}$ be a p-random set, with $0 < p$. In the 1-bootstrap model, $\mathcal{I}$ percolates with prob. 1.

Pf: By the warmup, we must show that $\Pr[\mathcal{I} \neq \emptyset] = 1$. For $t \geq 0$, let $\mathcal{A}_t$ be a set of t faces. Now $\Pr[\mathcal{I} = \emptyset] \leq \Pr[(I \cap A_t) = \emptyset]$ $= (1-p)^t$. Given $\epsilon > 0$, we pick t big enough s.t. $(1-p)^t < \epsilon$.

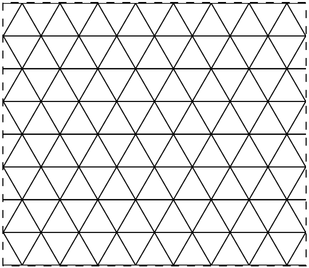
The Triangular Lattice



Lem 1: Let G be the triangular lattice and $\mathcal{I}$ be a p-random set, with $0 < p$. In the 1-bootstrap model, $\mathcal{I}$ percolates with prob. 1.

Pf: By the warmup, we must show that $\Pr[\mathcal{I} \neq \emptyset] = 1$. For $t \geq 0$, let $\mathcal{A}_t$ be a set of t faces. Now $\Pr[\mathcal{I} = \emptyset] \leq \Pr[(I \cap A_t) = \emptyset]$
$= (1-p)^t$. Given $\epsilon > 0$, we pick t big enough s.t. $(1-p)^t < \epsilon$.
So $\Pr[\mathcal{I} = \emptyset] < \epsilon$ for all $\epsilon > 0$.

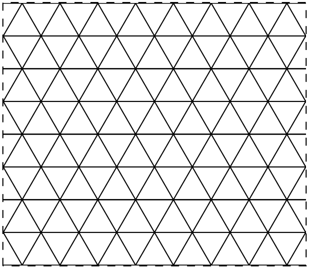
The Triangular Lattice



Lem 1: Let G be the triangular lattice and $\mathcal{I}$ be a p-random set, with $0 < p$. In the 1-bootstrap model, $\mathcal{I}$ percolates with prob. 1.

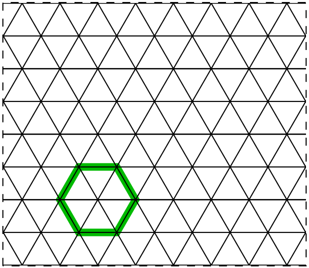
Pf: By the warmup, we must show that $\Pr[\mathcal{I} \neq \emptyset] = 1$. For $t \geq 0$, let $\mathcal{A}_t$ be a set of t faces. Now $\Pr[\mathcal{I} = \emptyset] \leq \Pr[(I \cap A_t) = \emptyset]$ $= (1-p)^t$. Given $\epsilon > 0$, we pick t big enough s.t. $(1-p)^t < \epsilon$. So $\Pr[\mathcal{I} = \emptyset] < \epsilon$ for all $\epsilon > 0$. Thus, $\Pr[\mathcal{I} \neq \emptyset] = 1$. So $\mathcal{I}$ percolates with prob. 1. ■

The Triangular Lattice



Lem 2: Let G be the triangular lattice and $\mathcal{I}$ be a p-random set, with $p < 1$. In the 2-bootstrap model, $\mathcal{I}$ percolates with prob. 0.

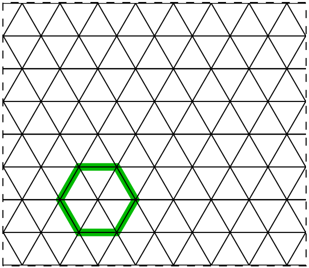
The Triangular Lattice



Lem 2: Let G be the triangular lattice and $\mathcal{I}$ be a p-random set, with $p < 1$. In the 2-bootstrap model, $\mathcal{I}$ percolates with prob. 0.

Pf: Note that if any  has no infected faces, then $\mathcal{I}$ will not percolate.

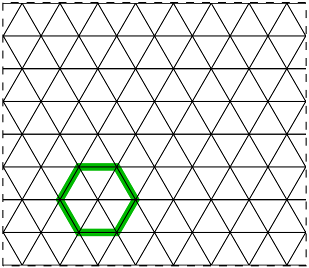
The Triangular Lattice



Lem 2: Let G be the triangular lattice and $\mathcal{I}$ be a p-random set, with $p < 1$. In the 2-bootstrap model, $\mathcal{I}$ percolates with prob. 0.

Pf: Note that if any  has no infected faces, then $\mathcal{I}$ will not percolate. Let $B_1, B_2, \ldots$ denote face disjoint copies of .

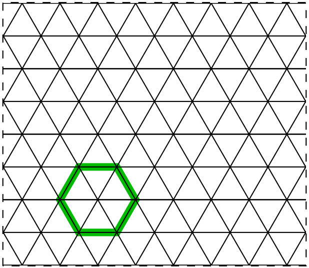
The Triangular Lattice



Lem 2: Let G be the triangular lattice and $\mathcal{I}$ be a p-random set, with $p < 1$. In the 2-bootstrap model, $\mathcal{I}$ percolates with prob. 0.

Pf: Note that if any  has no infected faces, then $\mathcal{I}$ will not percolate. Let $B_1, B_2, \ldots$ denote face disjoint copies of . Let E_j be the event that each of $B_1, \ldots, B_j$ has at least one infected face.

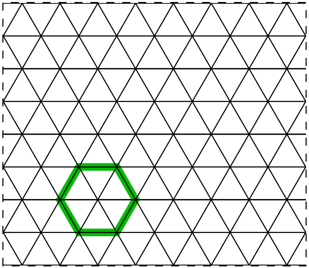
The Triangular Lattice



Lem 2: Let G be the triangular lattice and $\mathcal{I}$ be a p-random set, with $p < 1$. In the 2-bootstrap model, $\mathcal{I}$ percolates with prob. 0.

Pf: Note that if any  has no infected faces, then $\mathcal{I}$ will not percolate. Let $B_1, B_2, \ldots$ denote face disjoint copies of . Let E_j be the event that each of $B_1, \ldots, B_j$ has at least one infected face. Now $\Pr[\mathcal{I} \text{ percolates}] \leq \Pr[E_j] = (1 - (1-p)^6)^j$.

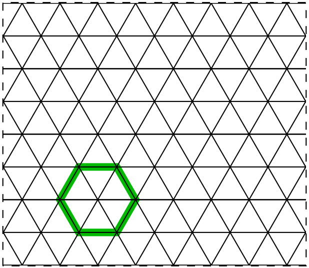
The Triangular Lattice



Lem 2: Let G be the triangular lattice and $\mathcal{I}$ be a p-random set, with $p < 1$. In the 2-bootstrap model, $\mathcal{I}$ percolates with prob. 0.

Pf: Note that if any  has no infected faces, then $\mathcal{I}$ will not percolate. Let $B_1, B_2, \ldots$ denote face disjoint copies of . Let E_j be the event that each of $B_1, \ldots, B_j$ has at least one infected face. Now $\Pr[\mathcal{I} \text{ percolates}] \le \Pr[E_j] = (1-(1-p)^6)^j$. Given any $\epsilon > 0$, j big enough s.t. $(1-(1-p)^6)^j < \epsilon$.

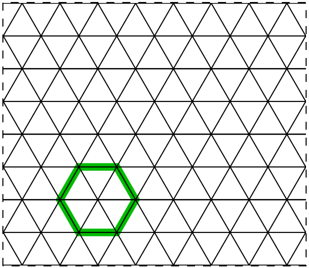
The Triangular Lattice



Lem 2: Let G be the triangular lattice and $\mathcal{I}$ be a p-random set, with $p < 1$. In the 2-bootstrap model, $\mathcal{I}$ percolates with prob. 0.

Pf: Note that if any  has no infected faces, then $\mathcal{I}$ will not percolate. Let $B_1, B_2, \ldots$ denote face disjoint copies of . Let E_j be the event that each of $B_1, \ldots, B_j$ has at least one infected face. Now $\Pr[\mathcal{I} \text{ percolates}] \leq \Pr[E_j] = (1-(1-p)^6)^j$. Given any $\epsilon > 0$, j big enough s.t. $(1-(1-p)^6)^j < \epsilon$. So $\Pr[\mathcal{I} \text{ percolates}] = 0$. ■

The Triangular Lattice

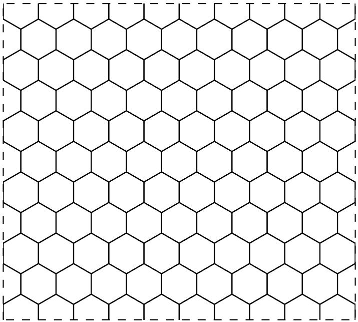


Lem 1: Let G be the triangular lattice and $\mathcal{I}$ be a p-random set, with $0 < p$. In the 1-bootstrap model, $\mathcal{I}$ percolates with prob. 1.

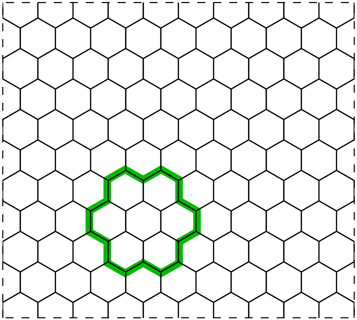
Lem 2: Let G be the triangular lattice and $\mathcal{I}$ be a p-random set, with $p < 1$. In the 2-bootstrap model, $\mathcal{I}$ percolates with prob. 0.

Cor: The triangular lattice has percolation threshold 1, whenever $0 < p < 1$.

The Hex Lattice



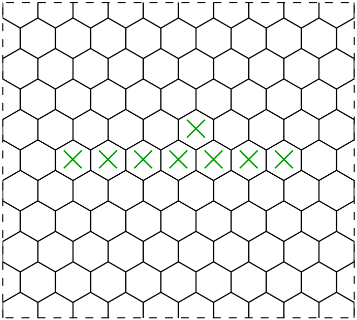
The Hex Lattice



Lem 3: Fix $p < 1$. For the hex lattice, in the 4-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. 0.

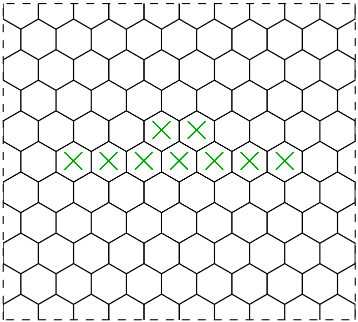
Pf: Same as Lem 2, but with  in place of . ■

The Hex Lattice



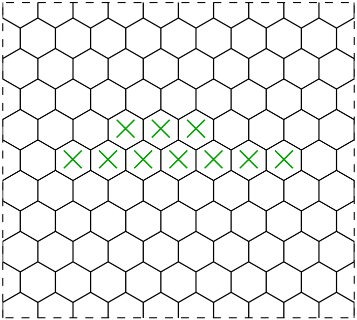
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



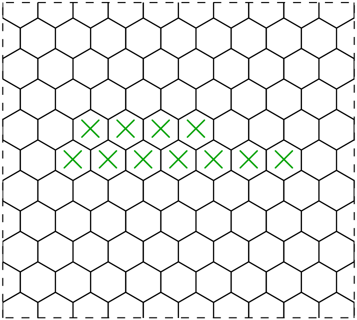
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



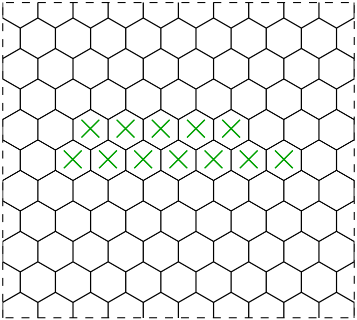
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



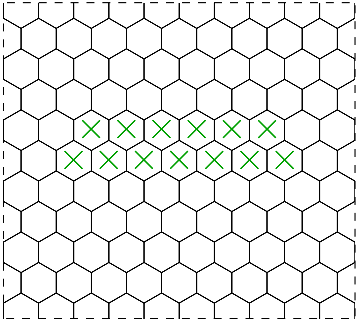
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



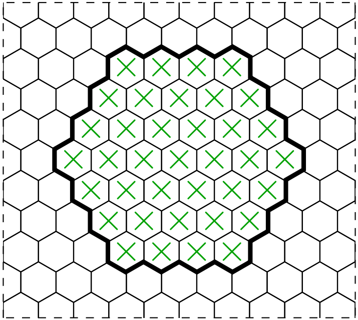
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



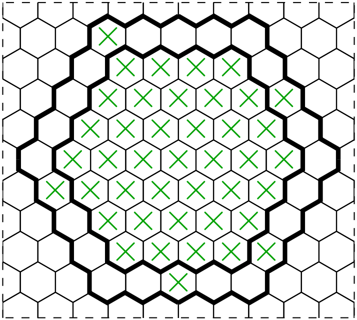
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



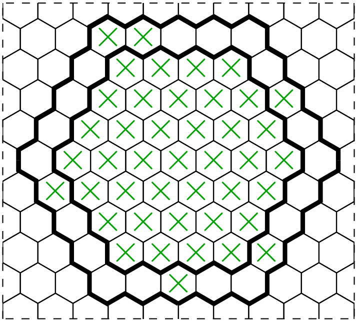
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



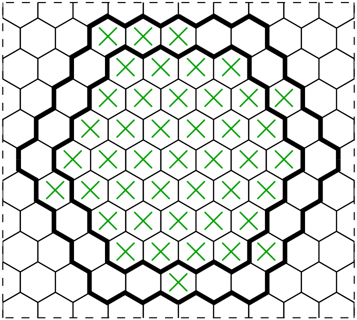
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



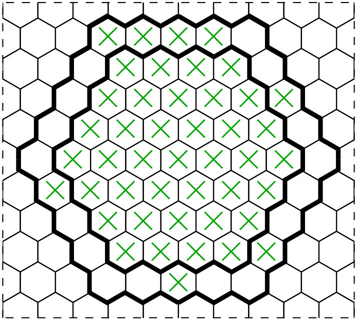
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



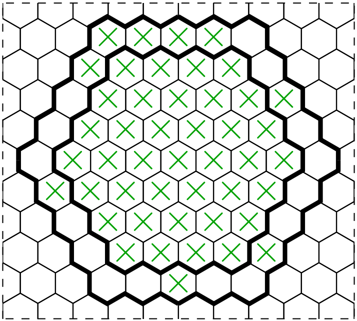
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



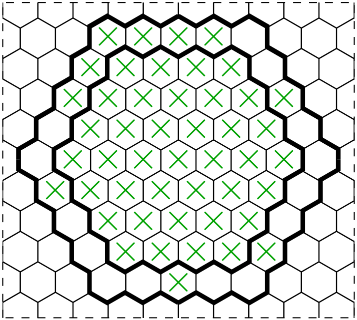
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



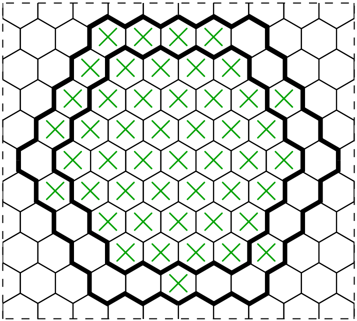
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



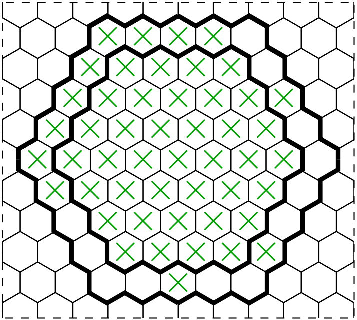
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



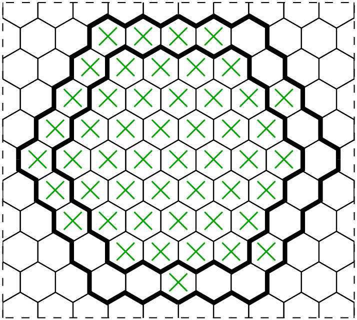
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



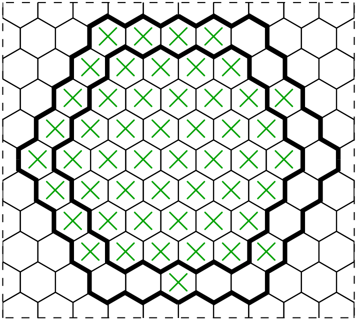
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



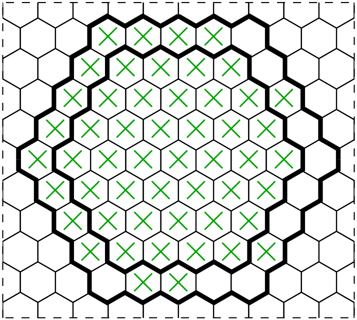
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



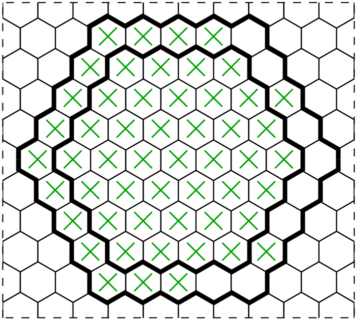
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



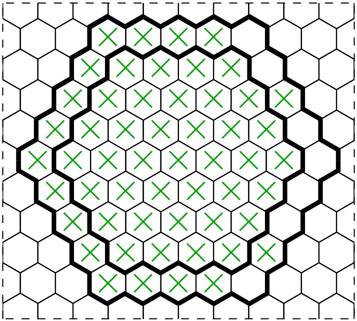
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



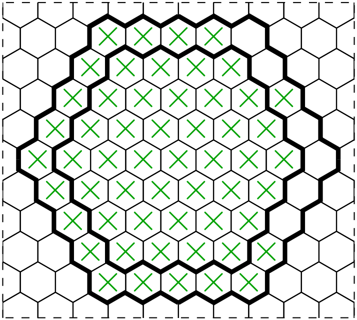
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



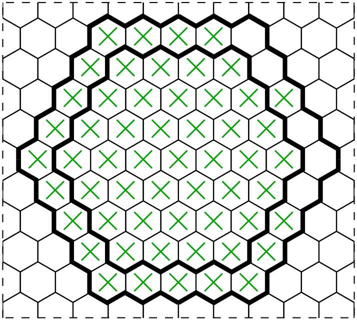
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



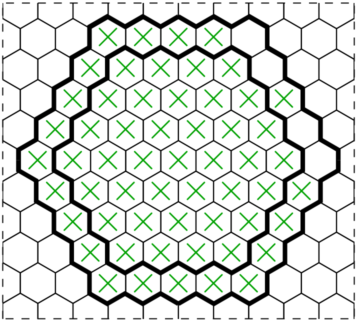
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



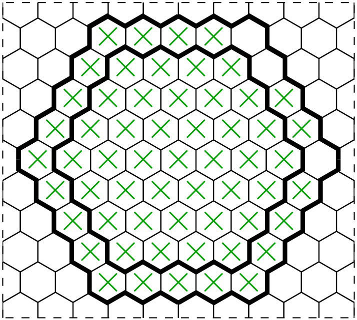
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



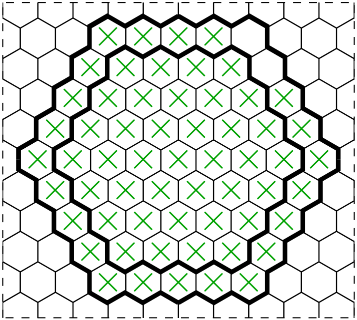
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



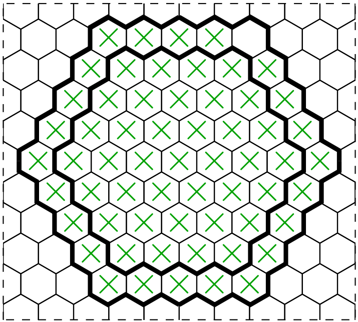
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice



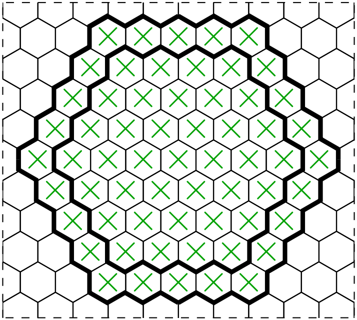
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

The Hex Lattice

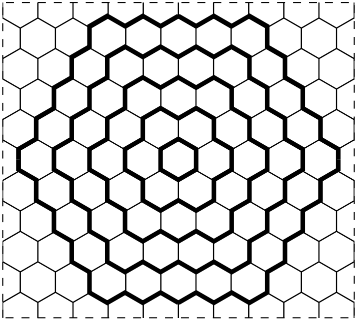


Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

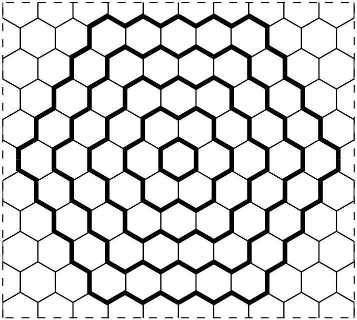
The Hex Lattice



The Hex Lattice

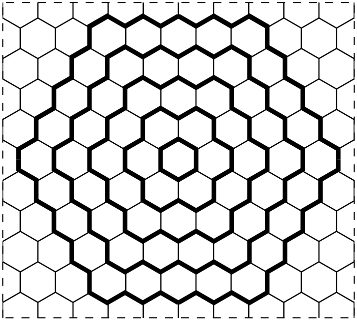


The Hex Lattice



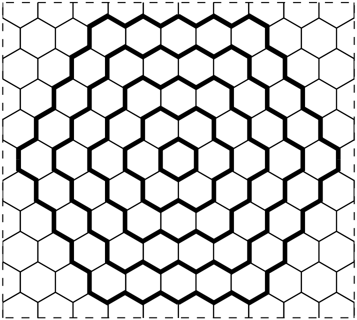
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.
Pf: Now $\Pr[\text{side with } t \text{ hexes is bad}] = (1-p)^t > 0$,

The Hex Lattice



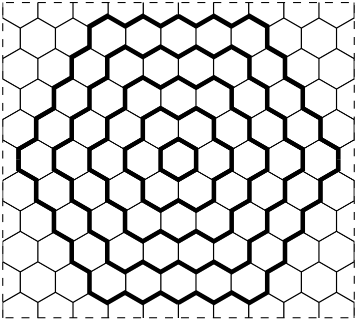
Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.
Pf: Now $\Pr[\text{side with } t \text{ hexes is bad}] = (1-p)^t > 0$,
so $\Pr[\text{ring with } t \text{ hexes per side is bad}] \leq 6(1-p)^t$.

The Hex Lattice



Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.
Pf: Now $\Pr[\text{side with } t \text{ hexes is bad}] = (1-p)^t > 0$,
so $\Pr[\text{ring with } t \text{ hexes per side is bad}] \leq 6(1-p)^t$.
Sum for all rings: $S = \sum_{t=j}^{\infty} 6(1-p)^t = \frac{6(1-p)^j}{p} < 1$ for big j.

The Hex Lattice



Lem 4: Fix $p > 0$. For the hex lattice, in the 3-bootstrap model, a p-random set $\mathcal{I}$ percolates with prob. > 0.

Pf: Now $\Pr[\text{side with } t \text{ hexes is bad}] = (1-p)^t > 0$,
so $\Pr[\text{ring with } t \text{ hexes per side is bad}] \leq 6(1-p)^t$.
Sum for all rings: $S = \sum_{t=j}^{\infty} 6(1-p)^t = \frac{6(1-p)^j}{p} < 1$ for big j.
$\Pr[\mathcal{I} \text{ percolates}] \geq (1-S) \Pr[\text{all small rings good}] > 0$. ■

0–1 Laws

Goal: For hex lattice, in 3-bootstrap model, want to show that p-random set percolates with prob. 1.

0–1 Laws

Goal: For hex lattice, in 3-bootstrap model, want to show that p-random set percolates with prob. 1. Try to copy proof for triangular lattice, repeating Lem 4 proof infinitely often.

0–1 Laws

Goal: For hex lattice, in 3-bootstrap model, want to show that p-random set percolates with prob. 1. Try to copy proof for triangular lattice, repeating Lem 4 proof infinitely often.

Problem: Events "all big rings are good" will never be independent.

0–1 Laws

Goal: For hex lattice, in 3-bootstrap model, want to show that p-random set percolates with prob. 1. Try to copy proof for triangular lattice, repeating Lem 4 proof infinitely often.

Problem: Events "all big rings are good" will never be independent. But can get arbitrarily close...

0–1 Laws

Goal: For hex lattice, in 3-bootstrap model, want to show that p-random set percolates with prob. 1. Try to copy proof for triangular lattice, repeating Lem 4 proof infinitely often.

Problem: Events "all big rings are good" will never be independent. But can get arbitrarily close...
Solution: 0–1 Law If an event E is translation invariant, then $\Pr(E) = 0$ or $\Pr(E) = 1$.

0–1 Laws

Goal: For hex lattice, in 3-bootstrap model, want to show that p-random set percolates with prob. 1. Try to copy proof for triangular lattice, repeating Lem 4 proof infinitely often.

Problem: Events "all big rings are good" will never be independent. But can get arbitrarily close. . .
Solution: 0–1 Law If an event E is translation invariant, then $\Pr(E) = 0$ or $\Pr(E) = 1$.

Thm 5: For the hex lattice, in the 3-bootstrap model, a p-random set percolates with prob. 1.

0–1 Laws

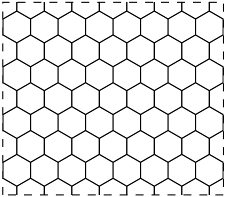
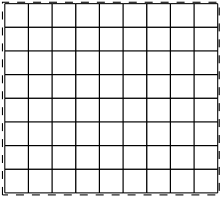
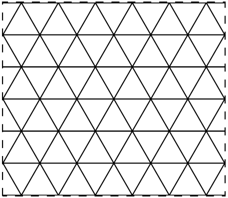
Goal: For hex lattice, in 3-bootstrap model, want to show that p-random set percolates with prob. 1. Try to copy proof for triangular lattice, repeating Lem 4 proof infinitely often.

Problem: Events "all big rings are good" will never be independent. But can get arbitrarily close...
Solution: 0–1 Law If an event E is translation invariant, then $\Pr(E) = 0$ or $\Pr(E) = 1$.

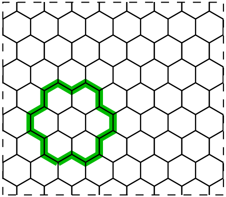
Thm 5: For the hex lattice, in the 3-bootstrap model, a p-random set percolates with prob. 1.
Cor 6: The hex lattice has threshold 3.

Regular Lattices and Beyond

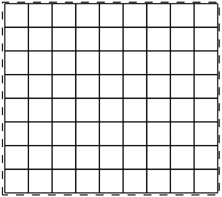
Regular Lattices and Beyond

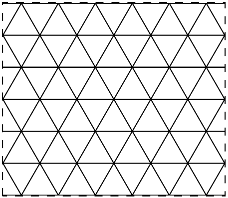




Regular Lattices and Beyond

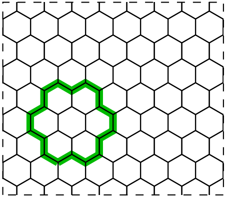


3

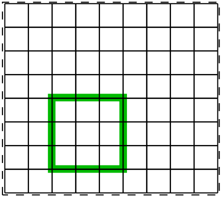




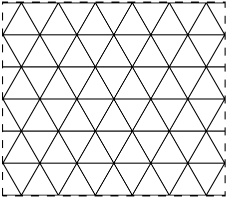
Regular Lattices and Beyond



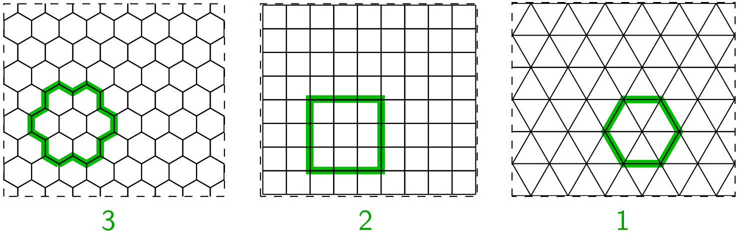
3



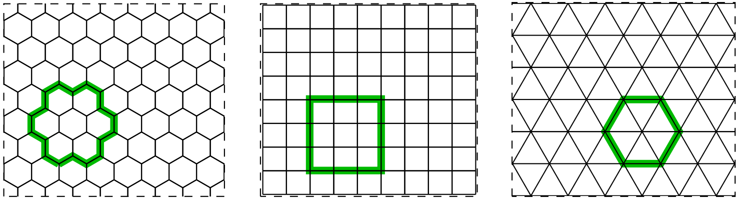
2



Regular Lattices and Beyond



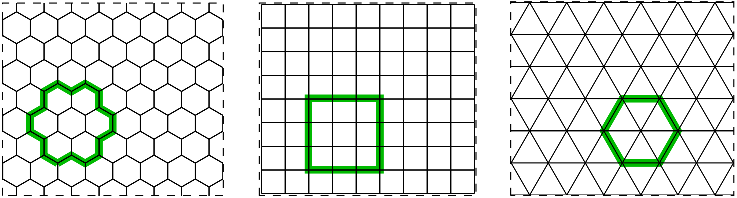
Regular Lattices and Beyond



3 2 1

What other graphs to consider?

Regular Lattices and Beyond

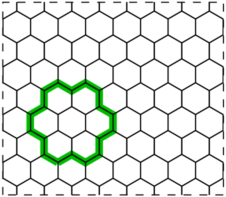


3 2 1

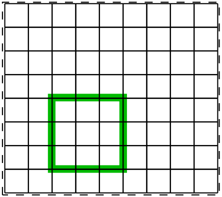
What other graphs to consider?

- Allow more face lengths in same graph

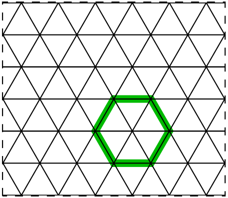
Regular Lattices and Beyond



3



2



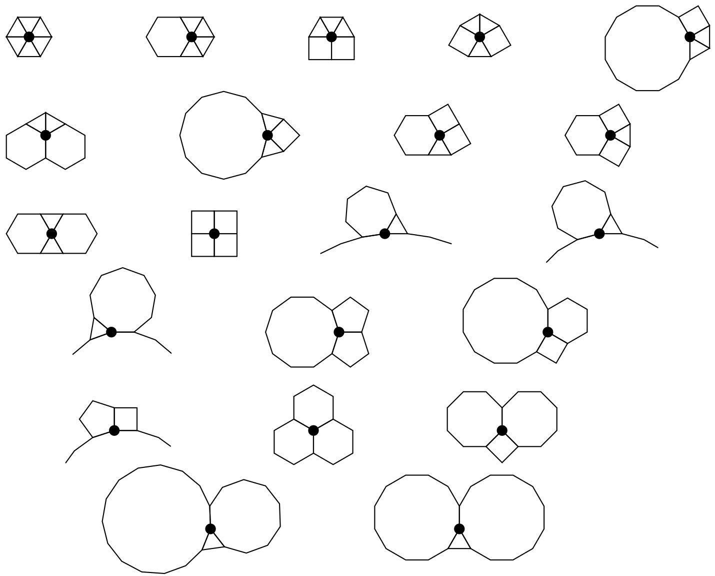
1

What other graphs to consider?

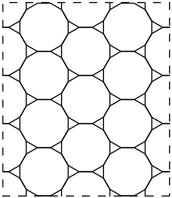
- Allow more face lengths in same graph
- All faces are still regular polygons

How Could Vertices Look?

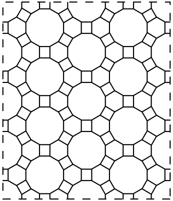
How Could Vertices Look?



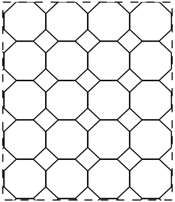
Archimedean Lattices



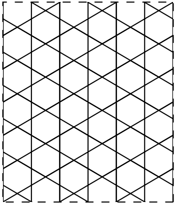
(3.12.12)



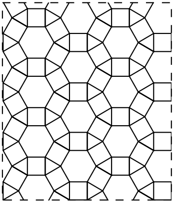
(4.6.12)



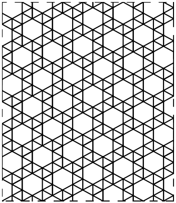
(4.8.8)



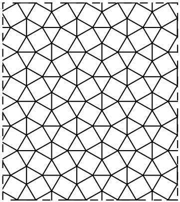
(3.6.3.6)



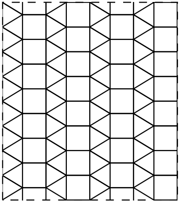
(3.4.6.4)



(3.3.3.3.6)

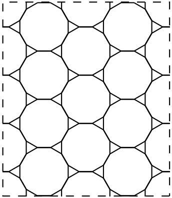


(3.3.4.3.4)

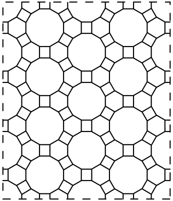


(3.3.3.4.4)

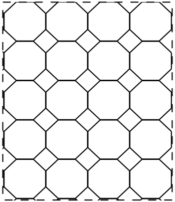
Archimedean Lattices



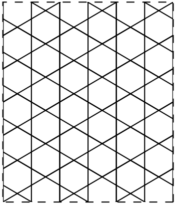
(3.12.12) 3



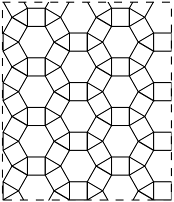
(4.6.12) 3



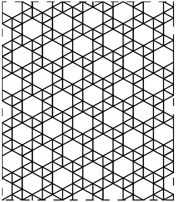
(4.8.8) 3



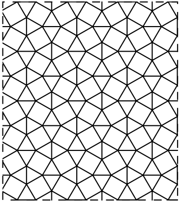
(3.6.3.6) 2



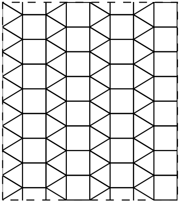
(3.4.6.4) 1



(3.3.3.3.6) 1

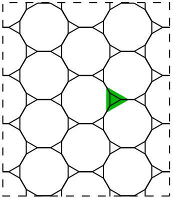


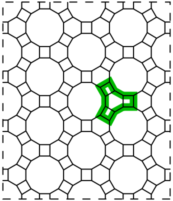
(3.3.4.3.4) 1

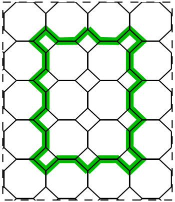


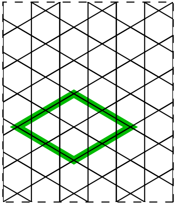
(3.3.3.4.4) 1

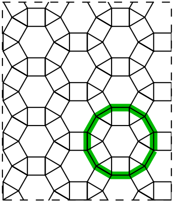
Archimedean Lattices

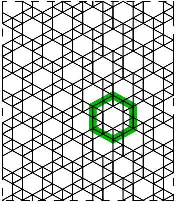

(3.12.12) 3

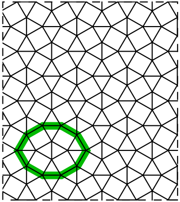

(4.6.12) 3

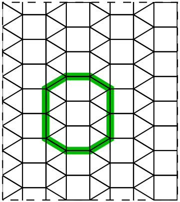

(4.8.8) 3


(3.6.3.6) 2

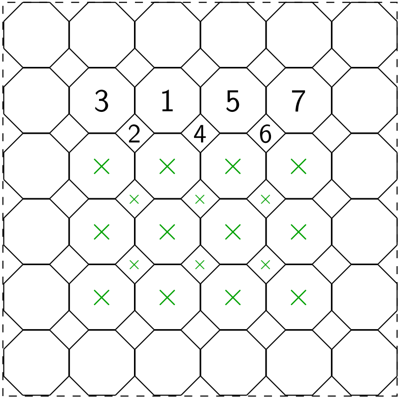

(3.4.6.4) 1


(3.3.3.3.6) 1

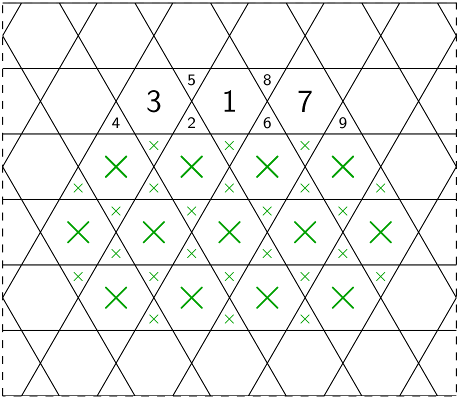

(3.3.4.3.4) 1


(3.3.3.4.4) 1

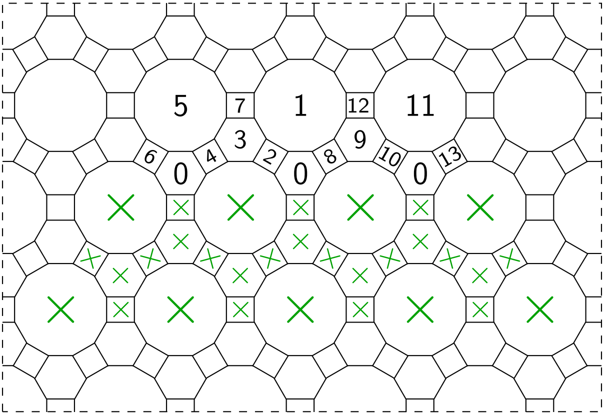
Archimedean Lattices: Lower Bounds



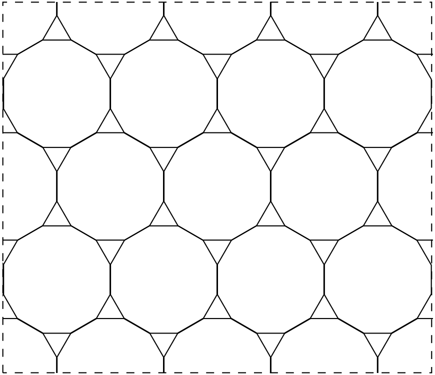
Archimedean Lattices: Lower Bounds



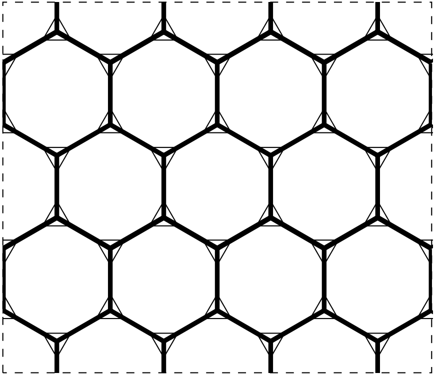
Archimedean Lattices: Lower Bounds



Archimedean Lattices: Lower Bounds



Archimedean Lattices: Lower Bounds



More General Tilings

Defn: Let $\mathcal{T}$ be set of all plane tilings such that if $T \in \mathcal{T}$ and T has one copy of some vertex type, then T has infinitely many copies of that type.

More General Tilings

Defn: Let $\mathcal{T}$ be set of all plane tilings such that if $T \in \mathcal{T}$ and T has one copy of some vertex type, then T has infinitely many copies of that type.

Main Theorem: Every $T \in \mathcal{T}$ has threshold at most 3.

More General Tilings

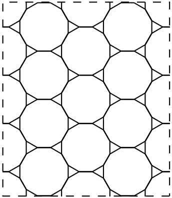
Defn: Let $\mathcal{T}$ be set of all plane tilings such that if $T \in \mathcal{T}$ and T has one copy of some vertex type, then T has infinitely many copies of that type.

Main Theorem: Every $T \in \mathcal{T}$ has threshold at most 3. The only tilings in $\mathcal{T}$ with threshold 3 are the Archimedean Lattices (3.12.12), (4.6.12), (4.8.8), and (6.6.6).

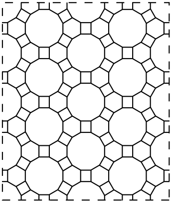
More General Tilings

Defn: Let $\mathcal{T}$ be set of all plane tilings such that if $T \in \mathcal{T}$ and T has one copy of some vertex type, then T has infinitely many copies of that type.

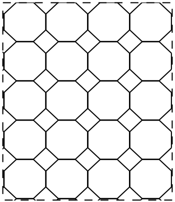
Main Theorem: Every $T \in \mathcal{T}$ has threshold at most 3. The only tilings in $\mathcal{T}$ with threshold 3 are the Archimedean Lattices (3.12.12), (4.6.12), (4.8.8), and (6.6.6).



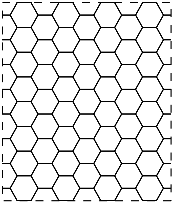
(3.12.12)



(4.6.12)



(4.8.8)

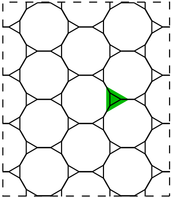


(6.6.6)

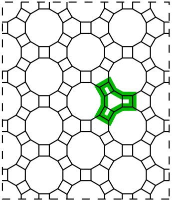
More General Tilings

Defn: Let $\mathcal{T}$ be set of all plane tilings such that if $T \in \mathcal{T}$ and T has one copy of some vertex type, then T has infinitely many copies of that type.

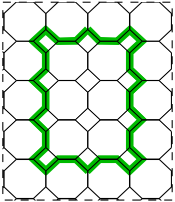
Main Theorem: Every $T \in \mathcal{T}$ has threshold at most 3. The only tilings in $\mathcal{T}$ with threshold 3 are the Archimedean Lattices (3.12.12), (4.6.12), (4.8.8), and (6.6.6).



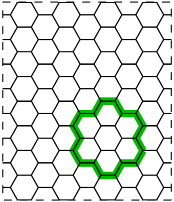
(3.12.12) 3



(4.6.12) 3



(4.8.8) 3



(6.6.6) 3