Hard and Easy Instances of L-Tromino Tilings [1]

Javier T. Akagi [1], Carlos F. Gaona [1], Fabricio Mendoza [1],
Manjil P. Saikia [2], Marcos Villagra [1]

[1]Universidad Nacional de Asunción
San Lorenzo, Paraguay

[2]School of Mathematics
Cardiff University, UK

30 June 2020
ICMMAS, Dibrugarh University

[1]Lecture Notes in Comp. Sc. (LNCS), Vol. 11355 (Springer) (2019).
Theoretical Computer Science (Elsevier), Vol. 815 (2020).

Outline

Outline

1. Introduction
   - Polyominoes
   - L-Tromino Tiling Problem
   - Computational Complexity
2. Tiling of the Aztec Rectangles
   - Aztec Rectangle
   - Aztec Rectangle with a single defect
   - Tiling Aztec Rectangle with unbounded number of defects
3. 180-Tromino Tiling
   - A rotation constraint
   - Forbidden Polyominoes
4. Open Problems

Polyominoes

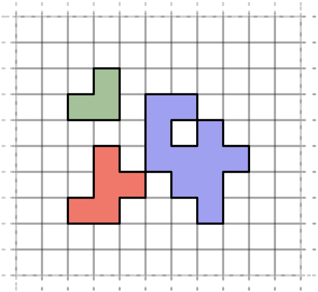
Definition

A polyomino is a planar figure made from one or more equal-sized squares, each joined together along an edge [S. Golomb (1953)].

Polyominoes

Definition

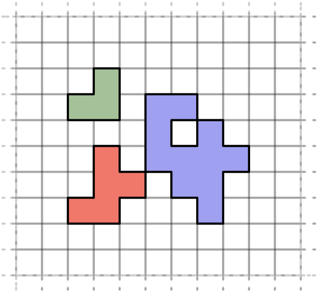
A polyomino is a planar figure made from one or more equal-sized squares, each joined together along an edge [S. Golomb (1953)].



Polyominoes

Definition

A polyomino is a planar figure made from one or more equal-sized squares, each joined together along an edge [S. Golomb (1953)].

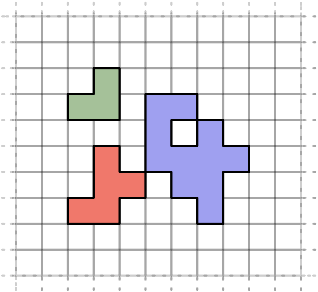


- Every cell (square) is fixed in a square lattice.

Polyominoes

Definition

A polyomino is a planar figure made from one or more equal-sized squares, each joined together along an edge [S. Golomb (1953)].



- Every cell (square) is fixed in a square lattice.
- Two cell are adjacent if the Manhattan distance is 1.

Outline

L-Tromino Tiling Problem

Definition

L-Tromino Tiling Problem

Definition

Given:

- A set of L-trominoes Σ called a **tile set**, $\Sigma = \{$ , , ,  $\}$

L-Tromino Tiling Problem

Definition

Given:

- A set of L-trominoes Σ called a **tile set**, $\Sigma = \{$ , , ,  $\}$
- and a polyomino R called **region**.

L-Tromino Tiling Problem

Definition

Given:

- A set of L-trominoes Σ called a **tile set**, $\Sigma = \{$ , , ,  $\}$
- and a polyomino R called **region**.

Goal: Place tiles from Σ to fill the region R covering every cell without overflowing the perimeter of R and without overlapping between the tiles.

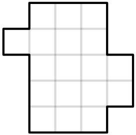
L-Tromino Tiling Problem

Definition

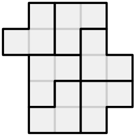
Given:

- A set of L-trominoes Σ called a **tile set**, $\Sigma = \{$ , , ,  $\}$
- and a polyomino R called **region**.

Goal: Place tiles from Σ to fill the region R covering every cell without overflowing the perimeter of R and without overlapping between the tiles.



(a) A region R



(b) A tiling of region R

Outline

Some Concepts

- In theoretical computer science, a computational problem is a mathematical object representing a collection of questions that computers might be able to solve.

Some Concepts

- In theoretical computer science, a computational problem is a mathematical object representing a collection of questions that computers might be able to solve.
- Complexity classes are concerned with the rate of growth of the requirement in resources as the input increases.

Some Concepts

- In theoretical computer science, a computational problem is a mathematical object representing a collection of questions that computers might be able to solve.
- Complexity classes are concerned with the rate of growth of the requirement in resources as the input increases.
- Time complexity is commonly estimated by counting the number of elementary operations performed by the algorithm, supposing that each elementary operation takes a fixed amount of time to perform.

Some Concepts

- In theoretical computer science, a computational problem is a mathematical object representing a collection of questions that computers might be able to solve.
- Complexity classes are concerned with the rate of growth of the requirement in resources as the input increases.
- Time complexity is commonly estimated by counting the number of elementary operations performed by the algorithm, supposing that each elementary operation takes a fixed amount of time to perform.
- If the time complexity is polynomial in the input parameters, then we say that a problem can be solved in Polynomial time.

NP-classes

NP-classes

- NP is the set of decision problems for which the problem instances, where the answer is ”yes”, have proofs verifiable in polynomial time.

NP-classes

- NP is the set of decision problems for which the problem instances, where the answer is "yes", have proofs verifiable in polynomial time.
- A decision problem is a question asked to a computer which gives a yes/no answer.

- NP is the set of decision problems for which the problem instances, where the answer is "yes", have proofs verifiable in polynomial time.
- A decision problem is a question asked to a computer which gives a yes/no answer.
- NP-Hard: H is NP-hard when every problem L in NP can be reduced in polynomial time to H.

NP-classes

- NP is the set of decision problems for which the problem instances, where the answer is "yes", have proofs verifiable in polynomial time.
- A decision problem is a question asked to a computer which gives a yes/no answer.
- NP-Hard: H is NP-hard when every problem L in NP can be reduced in polynomial time to H.
- Example: given a set (or multiset) of integers, is there a non-empty subset whose sum is zero?

NP-classes

- NP is the set of decision problems for which the problem instances, where the answer is "yes", have proofs verifiable in polynomial time.
- A decision problem is a question asked to a computer which gives a yes/no answer.
- NP-Hard: H is NP-hard when every problem L in NP can be reduced in polynomial time to H.
- Example: given a set (or multiset) of integers, is there a non-empty subset whose sum is zero?
- An NP-complete decision problem is one belonging to both the NP and the NP-hard complexity classes.

NP-classes

- NP is the set of decision problems for which the problem instances, where the answer is "yes", have proofs verifiable in polynomial time.
- A decision problem is a question asked to a computer which gives a yes/no answer.
- NP-Hard: H is NP-hard when every problem L in NP can be reduced in polynomial time to H.
- Example: given a set (or multiset) of integers, is there a non-empty subset whose sum is zero?
- An NP-complete decision problem is one belonging to both the NP and the NP-hard complexity classes.
- Example: Subgraph isomorphism problem.

Previous Work

Previous Work

- The problem of tiling with trominoes was first studied by Conway and Lagarias who presented an algebraic necessary condition for a region in order to have a tiling.

Previous Work

- The problem of tiling with trominoes was first studied by Conway and Lagarias who presented an algebraic necessary condition for a region in order to have a tiling.
- C. Moore and J. M. Robson (2000) proved that deciding the existence of a L-tromino tiling in a given region is **NP-complete** with a reduction from Monotone 1-in-3 SAT.

Previous Work

- The problem of tiling with trominoes was first studied by Conway and Lagarias who presented an algebraic necessary condition for a region in order to have a tiling.

- C. Moore and J. M. Robson (2000) proved that deciding the existence of a L-tromino tiling in a given region is **NP-complete** with a reduction from Monotone 1-in-3 SAT.

- T. Horiyama, T. Ito, K. Nakatsuka, A. Suzuki and R. Uehara (2012) constructed a one-one reduction from 1-in-3 SAT.

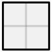
Outline

Aztec Rectangle

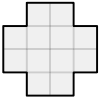
Aztec Rectangle

The **Aztec Diamond** $AD(n)$ is the union of all cell inside the contour $|x| + |y| = n + 1$.

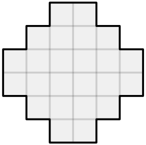
The **Aztec Diamond** $AD(n)$ is the union of all cell inside the contour $|x| + |y| = n + 1$.



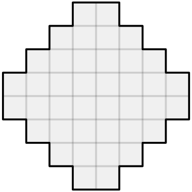
(a) $AD(1)$



(b) $AD(2)$

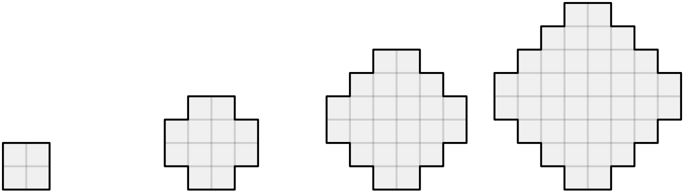


(c) $AD(3)$



(d) $AD(4)$

The **Aztec Diamond** $AD(n)$ is the union of all cell inside the contour $|x| + |y| = n + 1$.

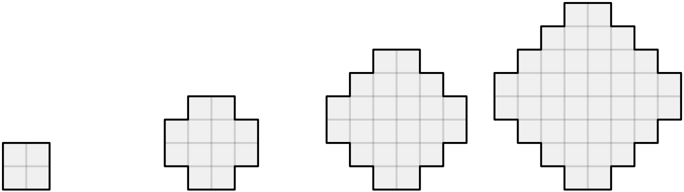


(a) $AD(1)$ (b) $AD(2)$ (c) $AD(3)$ (d) $AD(4)$

The **Aztec Rectangle** $\mathcal{AR}_{a,b}$ is a generalization of an Aztec Diamond.

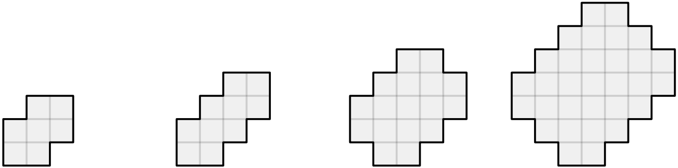
Aztec Rectangle

The **Aztec Diamond** $AD(n)$ is the union of all cell inside the contour $|x| + |y| = n + 1$.



(a) $AD(1)$ (b) $AD(2)$ (c) $AD(3)$ (d) $AD(4)$

The **Aztec Rectangle** $\mathcal{AR}_{a,b}$ is a generalization of an Aztec Diamond.

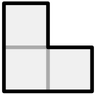


(a) $\mathcal{AR}_{1,2}$ (b) $\mathcal{AR}_{1,3}$ (c) $\mathcal{AR}_{2,3}$ (d) $\mathcal{AR}_{3,4}$

Tiling Aztec Rectangle (cont'd)

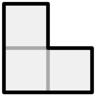
Tiling Aztec Rectangle (cont'd)

Each piece of L-tromino **covers 3 cells**.



Tiling Aztec Rectangle (cont'd)

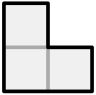
Each piece of L-tromino **covers 3 cells**.



In any L-tromino tiling, the number of covered cells is always **multiple of 3**.

Tiling Aztec Rectangle (cont'd)

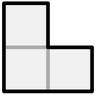
Each piece of L-tromino **covers 3 cells**.



In any L-tromino tiling, the number of covered cells is always **multiple of 3**.

The **number of cells** in an $\mathcal{AR}_{a,b}$ is given by

Each piece of L-tromino **covers 3 cells**.



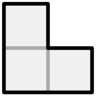
In any L-tromino tiling, the number of covered cells is always **multiple of 3**.

The **number of cells** in an $\mathcal{AR}_{a,b}$ is given by

$$|\mathcal{AR}_{a,b}| = a(b+1) + b(a+1).$$

Tiling Aztec Rectangle (cont'd)

Each piece of L-tromino **covers 3 cells**.



In any L-tromino tiling, the number of covered cells is always **multiple of 3**.

The **number of cells** in an $\mathcal{AR}_{a,b}$ is given by

$$|\mathcal{AR}_{a,b}| = a(b+1) + b(a+1).$$

Theorem

An Aztec rectangle $\mathcal{AR}_{a,b}$ has a tiling with L-trominoes
$\iff |\mathcal{AR}_{a,b}| \equiv 0 \pmod 3$
$\iff (a,b)$ *is equal to* $(3k, 3k')$ *or* $(3k-1, 3k'-1)$ *for some* $k, k' \in \mathbb{N}$.

TROMINO problem

TROMINO problem

Definition

TROMINO is the following problem:

INPUT : a region R with defects.

OUTPUT : "yes" if R has a cover and "no" otherwise.

TROMINO problem

Definition

$\mathrm{TROMINO}$ is the following problem:

INPUT : a region R with defects.
OUTPUT : "yes" if R has a cover and "no" otherwise.

Moore and Robson proved that $\mathrm{TROMINO}$ is NP-complete and Horiyama *et al.* proved that $\#\mathrm{TROMINO}$, the counting version of $\mathrm{TROMINO}$, is $\#$P-complete.

TROMINO problem

Definition

$\mathrm{TROMINO}$ is the following problem:

INPUT : a region R with defects.

OUTPUT : "yes" if R has a cover and "no" otherwise.

Moore and Robson proved that $\mathrm{TROMINO}$ is NP-complete and Horiyama *et al.* proved that $\#\mathrm{TROMINO}$, the counting version of $\mathrm{TROMINO}$, is $\#$P-complete.

A decision problem is P-complete if it is in P and every problem in P can be reduced to it by an appropriate reduction.

Tiling Aztec Rectangle (cont'd)

Tiling Aztec Rectangle (cont'd)

The problem of tiling an Aztec Rectangle can be solved **recursively**.

Tiling Aztec Rectangle (cont'd)

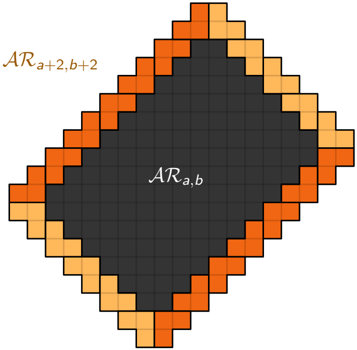
The problem of tiling an Aztec Rectangle can be solved **recursively**.

- If (a, b) equals $(3k, 3k')$, use pattern 1.
- If (a, b) equals $(3k-1, 3k'-1)$, use pattern 2.

Tiling Aztec Rectangle (cont'd)

The problem of tiling an Aztec Rectangle can be solved **recursively**.

- If (a, b) equals $(3k, 3k')$, use pattern 1.
- If (a, b) equals $(3k - 1, 3k' - 1)$, use pattern 2.

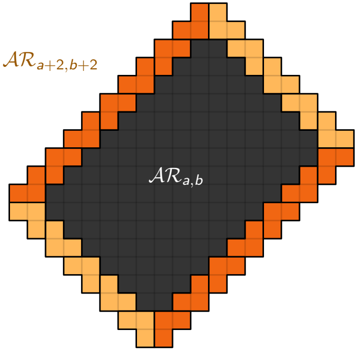


(a) Pattern 1

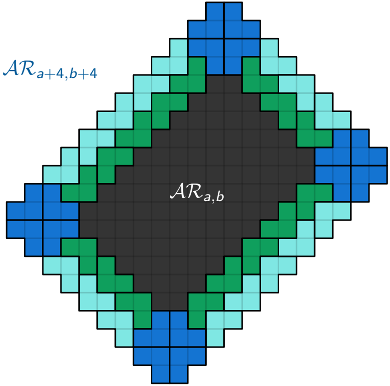
Tiling Aztec Rectangle (cont'd)

The problem of tiling an Aztec Rectangle can be solved **recursively**.

- If (a, b) equals $(3k, 3k')$, use pattern 1.
- If (a, b) equals $(3k-1, 3k'-1)$, use pattern 2.



(a) Pattern 1



(b) Pattern 2

The problem of tiling an Aztec Rectangle can be solved **recursively**.

Tiling Aztec Rectangle (cont'd)

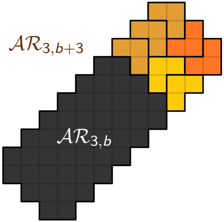
The problem of tiling an Aztec Rectangle can be solved **recursively**.

- If (a, b) equals $(3, 3k')$, use pattern 3.
- If (a, b) equals $(2, 3k' - 1)$, use pattern 4.

Tiling Aztec Rectangle (cont'd)

The problem of tiling an Aztec Rectangle can be solved **recursively**.

- If (a, b) equals $(3, 3k')$, use pattern 3.
- If (a, b) equals $(2, 3k' - 1)$, use pattern 4.

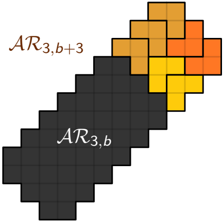


(a) Pattern 3

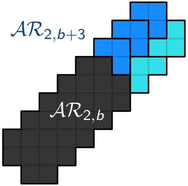
Tiling Aztec Rectangle (cont'd)

The problem of tiling an Aztec Rectangle can be solved **recursively**.

- If (a, b) equals $(3, 3k')$, use pattern 3.
- If (a, b) equals $(2, 3k' - 1)$, use pattern 4.



(a) Pattern 3

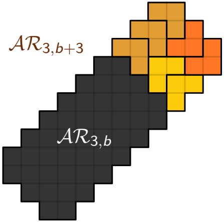


(b) Pattern 4

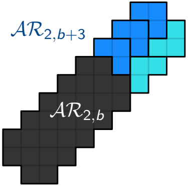
Tiling Aztec Rectangle (cont'd)

The problem of tiling an Aztec Rectangle can be solved **recursively**.

- If (a, b) equals $(3, 3k')$, use pattern 3.
- If (a, b) equals $(2, 3k' - 1)$, use pattern 4.

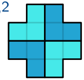


(a) Pattern 3



(b) Pattern 4

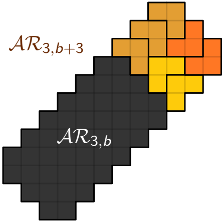
Base case: $\mathcal{AR}_{2,2}$ and $\mathcal{AR}_{3,3}$.



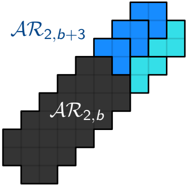
Tiling Aztec Rectangle (cont'd)

The problem of tiling an Aztec Rectangle can be solved **recursively**.

- If (a, b) equals $(3, 3k')$, use pattern 3.
- If (a, b) equals $(2, 3k' - 1)$, use pattern 4.

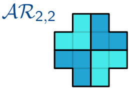


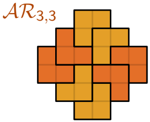
(a) Pattern 3



(b) Pattern 4

Base case: $\mathcal{AR}_{2,2}$ and $\mathcal{AR}_{3,3}$.





Polynomial time algorithm

Polynomial time algorithm

Theorem

A tromino cover for $\mathcal{AR}_{a,b}$ can be found in time $O(b^2)$.

Polynomial time algorithm

Theorem

A tromino cover for $\mathcal{AR}_{a,b}$ can be found in time $O(b^2)$.

Proof.

Given a, b, the following procedure finds a tiling for $\mathcal{AR}_{a,b}$.

Polynomial time algorithm

Theorem

A tromino cover for $\mathcal{AR}_{a,b}$ can be found in time $O(b^2)$.

Proof.

Given a, b, the following procedure finds a tiling for $\mathcal{AR}_{a,b}$.

1. If $a = 2, b = 5$ or $a = 3, b = 6$, return the base cases.

Polynomial time algorithm

Theorem

A tromino cover for $\mathcal{AR}_{a,b}$ can be found in time $O(b^2)$.

Proof.

Given a, b, the following procedure finds a tiling for $\mathcal{AR}_{a,b}$.

1. If $a = 2, b = 5$ or $a = 3, b = 6$, return the base cases.
2. If $a(b+1) + b(a+1) \not\equiv 0 \pmod 3$, then return "no tiling".

Polynomial time algorithm

Theorem

A tromino cover for $\mathcal{AR}_{a,b}$ can be found in time $O(b^2)$.

Proof.

Given a, b, the following procedure finds a tiling for $\mathcal{AR}_{a,b}$.

1. If $a = 2, b = 5$ or $a = 3, b = 6$, return the base cases.
2. If $a(b+1) + b(a+1) \not\equiv 0 \pmod 3$, then return "no tiling".
3. If a, b are multiples of 3, then
    1. $R \leftarrow \texttt{ARTiling}(a-2, b-2)$;
    2. fill the borders of R using stairs.

Polynomial time algorithm

Theorem

A tromino cover for $\mathcal{AR}_{a,b}$ can be found in time $O(b^2)$.

Proof.

Given a, b, the following procedure finds a tiling for $\mathcal{AR}_{a,b}$.

1. If $a = 2, b = 5$ or $a = 3, b = 6$, return the base cases.
2. If $a(b+1) + b(a+1) \not\equiv 0 \pmod 3$, then return "no tiling".
3. If a, b are multiples of 3, then
    1. $R \leftarrow \texttt{ARTiling}(a-2, b-2)$;
    2. fill the borders of R using stairs.
4. If $a+1, b+1$ is a multiple of 3, then
    1. $R \leftarrow \texttt{ARTiling}(a-4, b-4)$;
    2. fill the borders of R using stairs.

Polynomial time algorithm

Theorem

A tromino cover for $\mathcal{AR}_{a,b}$ can be found in time $O(b^2)$.

Proof.

Given a, b, the following procedure finds a tiling for $\mathcal{AR}_{a,b}$.

1. If $a = 2, b = 5$ or $a = 3, b = 6$, return the base cases.
2. If $a(b+1) + b(a+1) \not\equiv 0 \pmod 3$, then return "no tiling".
3. If a, b are multiples of 3, then
    1. $R \leftarrow \texttt{ARTiling}(a-2, b-2)$;
    2. fill the borders of R using stairs.
4. If $a+1, b+1$ is a multiple of 3, then
    1. $R \leftarrow \texttt{ARTiling}(a-4, b-4)$;
    2. fill the borders of R using stairs.
5. Return R.

$\square$

Time Complexity

Time Complexity

- Steps 2 and 3 are done in time $O(\log b)$.

Time Complexity

- Steps 2 and 3 are done in time $O(\log b)$.
- Steps 3.2 and 4.2 can be done in time $O(b)$.

- Steps 2 and 3 are done in time $O(\log b)$.
- Steps 3.2 and 4.2 can be done in time $O(b)$.
- Giving a total time complexity of $O(b^2)$.

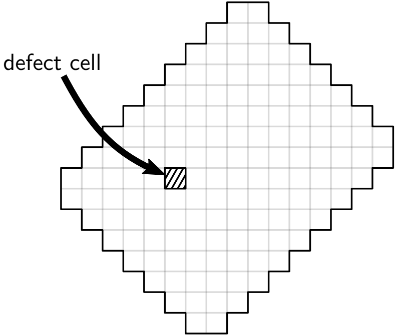
Outline

Tiling Aztec Rectangle with a single defect

A **defect cell** is a cell in which no tromino can be placed on top.

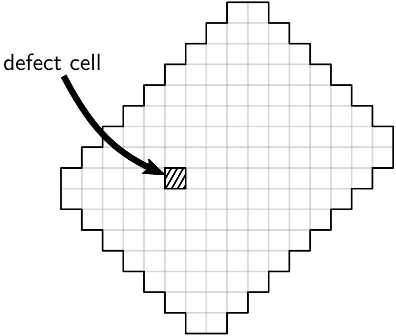
Tiling Aztec Rectangle with a single defect

A **defect cell** is a cell in which no tromino can be placed on top.



Tiling Aztec Rectangle with a single defect

A **defect cell** is a cell in which no tromino can be placed on top.



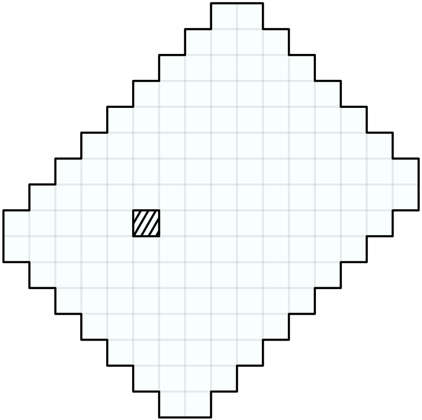
Theorem

An Aztec rectangle $\mathcal{AR}_{a,b}$ with one defect has a tiling with L-trominoes

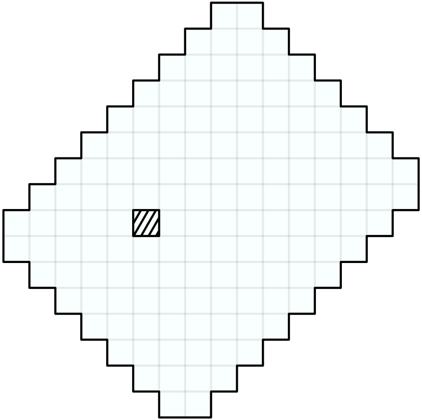
$\iff$ $|\mathcal{AR}_{a,b}| \equiv 1 \pmod 3$

$\iff$ *a or b is equal to $3k - 2$ for some $k \in \mathbb{N}$.*

Tiling Aztec Rectangle with a single defect (cont'd)

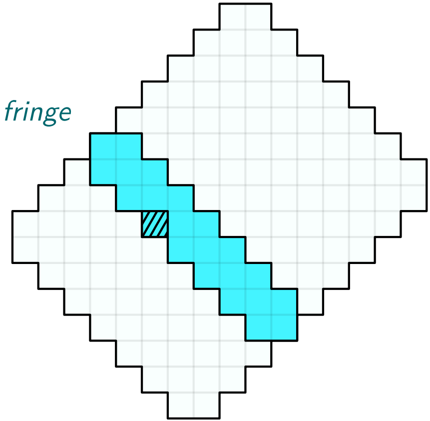


- Place a *fringe* where it covers the defect.



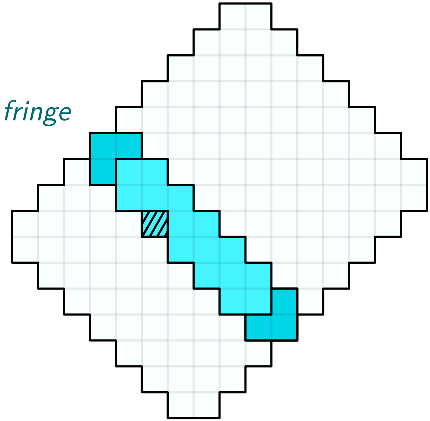
Tiling Aztec Rectangle with a single defect (cont'd)

- Place a *fringe* where it covers the defect.



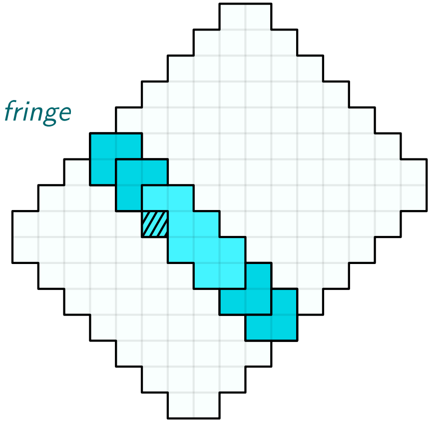
Tiling Aztec Rectangle with a single defect (cont'd)

- Place a *fringe* where it covers the defect.



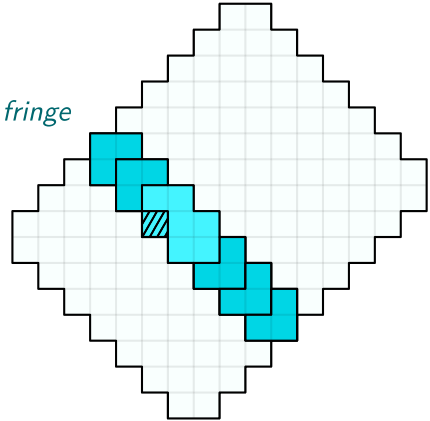
Tiling Aztec Rectangle with a single defect (cont'd)

- Place a *fringe* where it covers the defect.



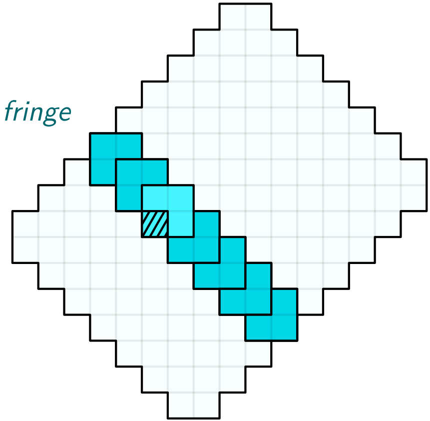
Tiling Aztec Rectangle with a single defect (cont'd)

- Place a *fringe* where it covers the defect.



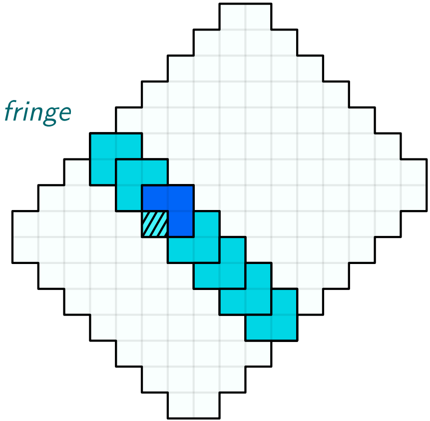
Tiling Aztec Rectangle with a single defect (cont'd)

- Place a *fringe* where it covers the defect.



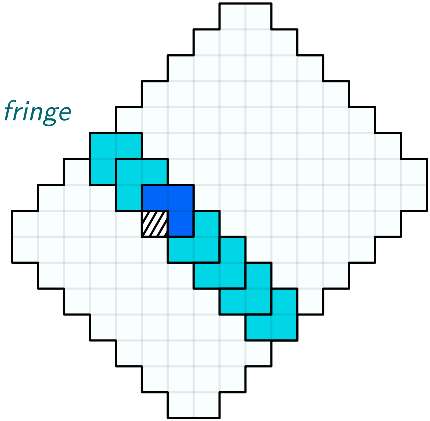
Tiling Aztec Rectangle with a single defect (cont'd)

- Place a *fringe* where it covers the defect.



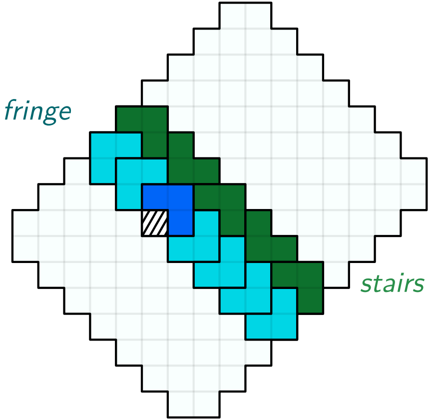
Tiling Aztec Rectangle with a single defect (cont'd)

- Place a *fringe* where it covers the defect.
- Place *stairs* to cover other cells.

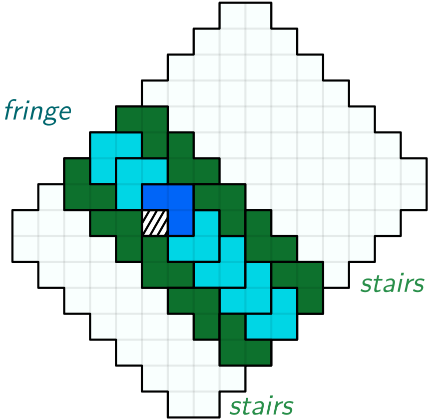


Tiling Aztec Rectangle with a single defect (cont'd)

- Place a *fringe* where it covers the defect.
- Place *stairs* to cover other cells.

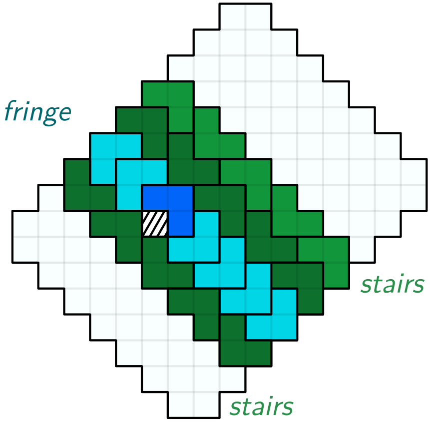


- Place a *fringe* where it covers the defect.
- Place *stairs* to cover other cells.

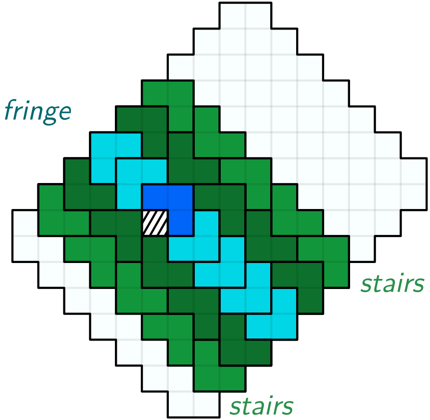


Tiling Aztec Rectangle with a single defect (cont'd)

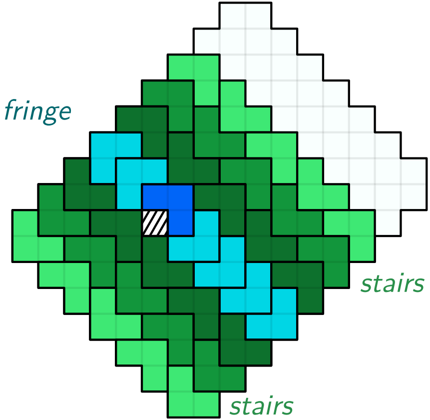
- Place a *fringe* where it covers the defect.
- Place *stairs* to cover other cells.



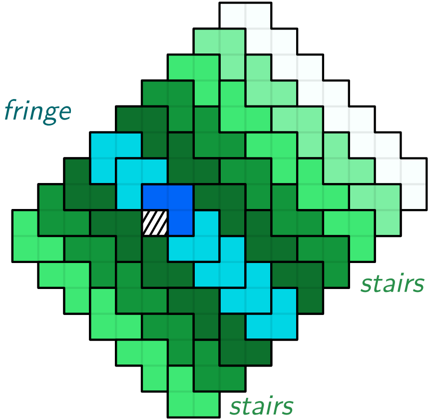
- Place a *fringe* where it covers the defect.
- Place *stairs* to cover other cells.



- Place a *fringe* where it covers the defect.
- Place *stairs* to cover other cells.

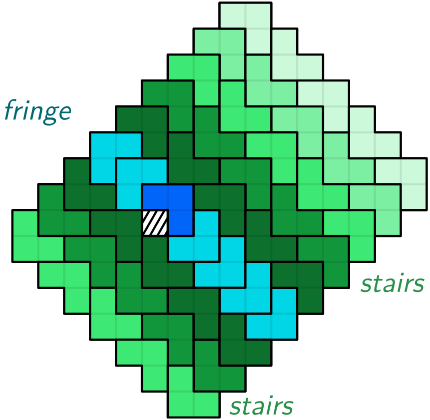


- Place a *fringe* where it covers the defect.
- Place *stairs* to cover other cells.



Tiling Aztec Rectangle with a single defect (cont'd)

- Place a *fringe* where it covers the defect.
- Place *stairs* to cover other cells.



Outline

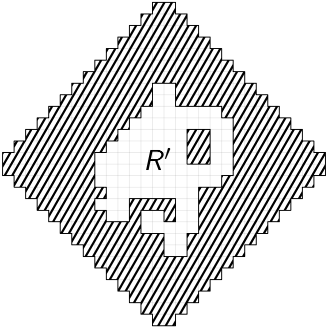
Tiling Aztec Rectangle with an unbounded number of defects

Tiling Aztec Rectangle with an unbounded number of defects

Given a region R', we can embed R' inside a sufficiently large Aztec Rectangle $\mathcal{AR}_{a,b}$.

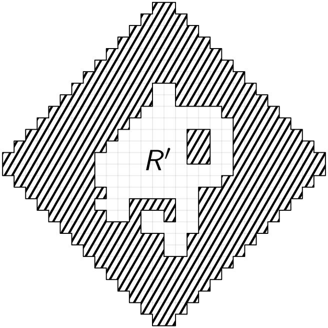
Tiling Aztec Rectangle with an unbounded number of defects

Given a region R', we can embed R' inside a sufficiently large Aztec Rectangle $\mathcal{AR}_{a,b}$.



Tiling Aztec Rectangle with an unbounded number of defects

Given a region R', we can embed R' inside a sufficiently large Aztec Rectangle $\mathcal{AR}_{a,b}$.



Theorem

The problem of tiling Aztec Rectangle $\mathcal{AR}_{a,b}$ with an unbounded number of defects is ***NP-complete****.*

Outline

180°L-Tromino Tiling

180°L-Tromino Tiling

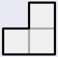
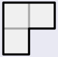
Definition

The 180-tromino tiling problem only allows 180° rotations of L-trominoes, i.e., the tile set can be

180°L-Tromino Tiling

Definition

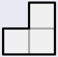
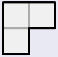
The 180-tromino tiling problem only allows 180° rotations of L-trominoes, i.e., the tile set can be

$\Sigma = \{$ **right-oriented** 180-trominoes $\} = \{$ ,  $\}$

180°L-Tromino Tiling

Definition

The 180-tromino tiling problem only allows 180° rotations of L-trominoes, i.e., the tile set can be

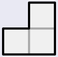
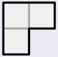
$\Sigma = \{$ **right-oriented** 180-trominoes $\} = \{$ ,  $\}$

or

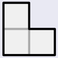
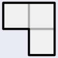
180°L-Tromino Tiling

Definition

The 180-tromino tiling problem only allows 180° rotations of L-trominoes, i.e., the tile set can be

$\Sigma = \{$ **right-oriented** 180-trominoes $\} = \{$ ,  $\}$

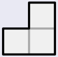
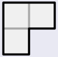
or

$\Sigma = \{$ **left-oriented** 180-trominoes $\} = \{$ ,  $\}$.

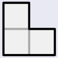
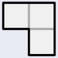
180°L-Tromino Tiling

Definition

The 180-tromino tiling problem only allows 180° rotations of L-trominoes, i.e., the tile set can be

$\Sigma = \{$ **right-oriented** 180-trominoes $\} = \{$ ,  $\}$

or

$\Sigma = \{$ **left-oriented** 180-trominoes $\} = \{$ ,  $\}$.

With no loss of generality, we will only consider **right-oriented 180-trominoes**.

180°L-Tromino Tiling (cont'd)

180°L-Tromino Tiling (cont'd)

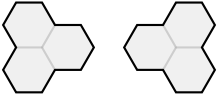
Theorem

There is a one-one correspondence between 180-tromino tiling and the triangular trihex tiling [Conway and Lagarias, (1990)].

180°L-Tromino Tiling (cont'd)

Theorem

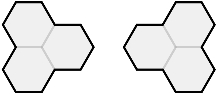
There is a one-one correspondence between 180-tromino tiling and the triangular trihex tiling [Conway and Lagarias, (1990)].



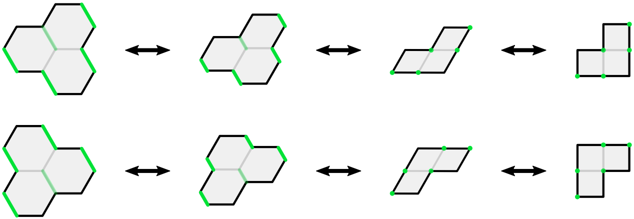
Two **triangular trihex**.

Theorem

There is a one-one correspondence between 180-tromino tiling and the triangular trihex tiling [Conway and Lagarias, (1990)].



Two **triangular trihex**.



Transformation from **triangular trihex** to **180-tromino**

180°L-Tromino Tiling (cont'd)

180°L-Tromino Tiling (cont'd)

Definition

A **cell tetrasection** is a division of a cell into 4 equal size cells.

Definition

A **cell tetrasection** is a division of a cell into 4 equal size cells.

Definition

A **tetrasected polyomino** $P^{\boxplus}$ is obtained by tetrasecting each cell of a poylomino P.

180°L-Tromino Tiling (cont'd)

Definition

A **cell tetrasection** is a division of a cell into 4 equal size cells.

Definition

A **tetrasected polyomino** $P^{\boxplus}$ is obtained by tetrasecting each cell of a poylomino P.

If there is a I-tromino tiling for some R, then there is also a 180-tromino tiling for $R^{\boxplus}$.

180°L-Tromino Tiling (cont'd)

Definition

A **cell tetrasection** is a division of a cell into 4 equal size cells.

Definition

A **tetrasected polyomino** $P^{\boxplus}$ is obtained by tetrasecting each cell of a poylomino P.

If there is a I-tromino tiling for some R, then there is also a 180-tromino tiling for $R^{\boxplus}$.

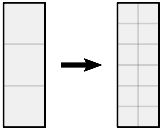
Definition

A **cell tetrasection** is a division of a cell into 4 equal size cells.

Definition

A **tetrasected polyomino** $P^{\boxplus}$ is obtained by tetrasecting each cell of a poylomino P.

If there is a I-tromino tiling for some R, then there is also a 180-tromino tiling for $R^{\boxplus}$.



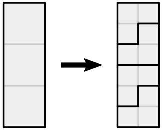
Definition

A **cell tetrasection** is a division of a cell into 4 equal size cells.

Definition

A **tetrasected polyomino** $P^{\boxplus}$ is obtained by tetrasecting each cell of a poylomino P.

If there is a I-tromino tiling for some R, then there is also a 180-tromino tiling for $R^{\boxplus}$.



180°L-Tromino Tiling (cont'd)

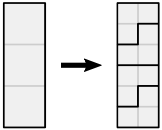
Definition

A **cell tetrasection** is a division of a cell into 4 equal size cells.

Definition

A **tetrasected polyomino** $P^{\boxplus}$ is obtained by tetrasecting each cell of a poylomino P.

If there is a I-tromino tiling for some R, then there is also a 180-tromino tiling for $R^{\boxplus}$.



However, it is not known if the converse statement is true or false.

180°L-Tromino Tiling (cont'd)

Horiyama et al. also proved that the I-tromino tiling problem is **NP-Complete**.

180°L-Tromino Tiling (cont'd)

Horiyama et al. also proved that the I-tromino tiling problem is **NP-Complete**.

Theorem [Horiyama, Ito, Nakatsuka, Suzuki and Uehara (2012)]

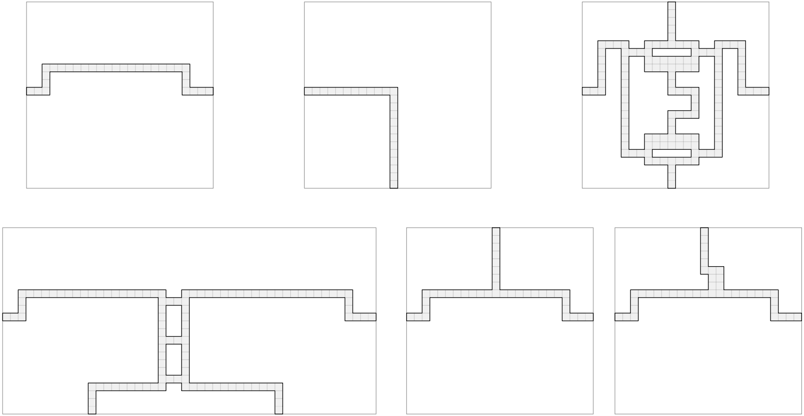
1-in-3 SAT $\leq_P$ I-tromino Tiling

180°L-Tromino Tiling (cont'd)

Horiyama et al. also proved that the I-tromino tiling problem is **NP-Complete**.

Theorem [Horiyama, Ito, Nakatsuka, Suzuki and Uehara (2012)]

1-in-3 SAT $\leq_P$ I-tromino Tiling

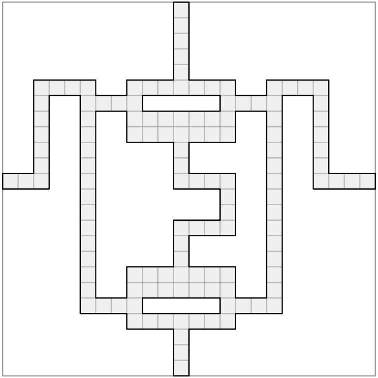


180°L-Tromino Tiling (cont'd)

180°L-Tromino Tiling (cont'd)

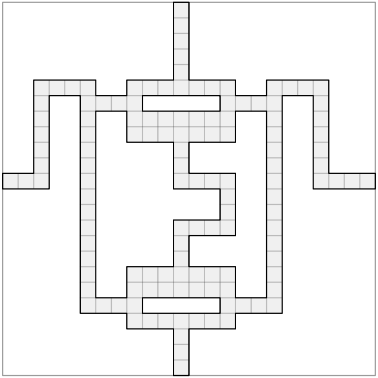
In each gadget G, I-tromino tiling for G can be simulated with 180-tromino tiling for $G^{\boxplus}$.

In each gadget G, I-tromino tiling for G can be simulated with 180-tromino tiling for $G^{\boxplus}$.

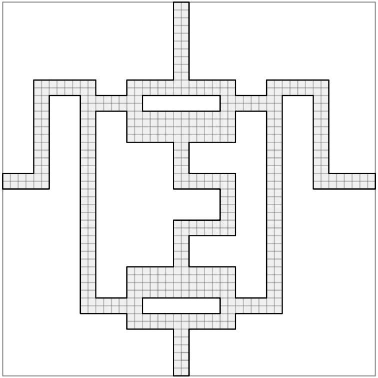


(a) Original gadget G.

In each gadget G, I-tromino tiling for G can be simulated with 180-tromino tiling for $G^{\boxplus}$.



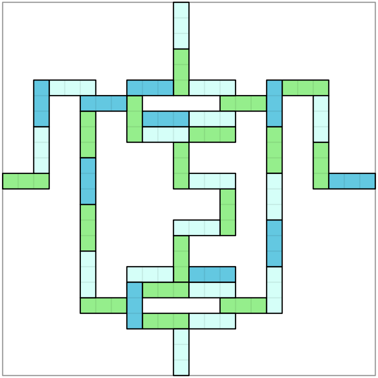
(a) Original gadget G.



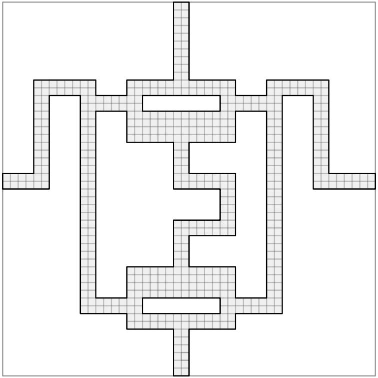
(b) Tetrasected gadget $G^{\boxplus}$.

180°L-Tromino Tiling (cont'd)

In each gadget G, I-tromino tiling for G can be simulated with 180-tromino tiling for $G^{\boxplus}$.

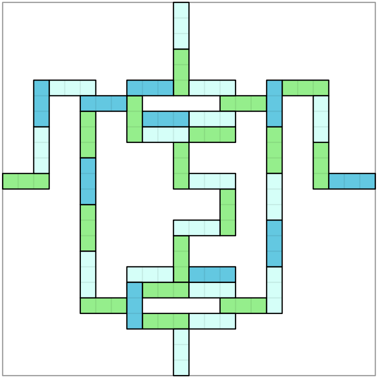


(a) Original gadget G.

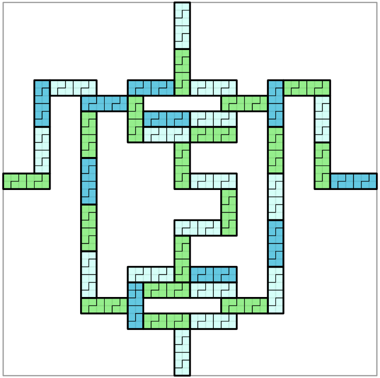


(b) Tetrasected gadget $G^{\boxplus}$.

In each gadget G, I-tromino tiling for G can be simulated with 180-tromino tiling for $G^{\boxplus}$.



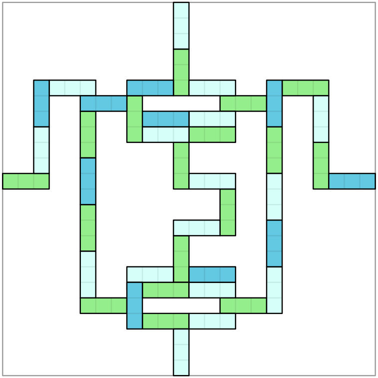
(a) Original gadget G.



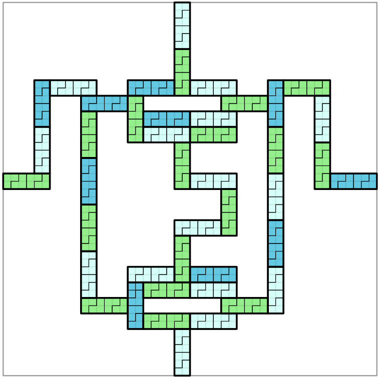
(b) Tetrasected gadget $G^{\boxplus}$.

180°L-Tromino Tiling (cont'd)

In each gadget G, I-tromino tiling for G can be simulated with 180-tromino tiling for $G^{\boxplus}$.



(a) Original gadget G.



(b) Tetrasected gadget $G^{\boxplus}$.

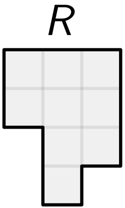
Theorem

180-tromino tiling is ***NP-complete****.*

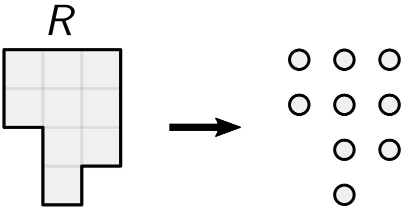
Outline

The 180-tromino tiling can also be reduced to the **Maximum Independent Set** problem.

The 180-tromino tiling can also be reduced to the **Maximum Independent Set** problem.

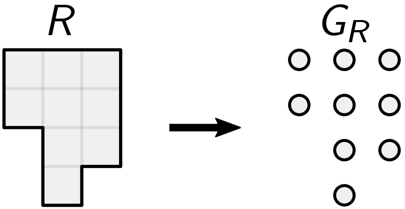


The 180-tromino tiling can also be reduced to the **Maximum Independent Set** problem.



- Transformation from R to G_R:
  - Transform every cell of R to vertices of G_R.
  - Add horizontal, vertical and northeast-diagonal edges.

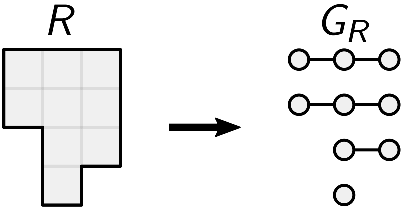
The 180-tromino tiling can also be reduced to the **Maximum Independent Set** problem.



- Transformation from R to G_R:
  - Transform every cell of R to vertices of G_R.
  - Add horizontal, vertical and northeast-diagonal edges.

Forbidden Polyominoes

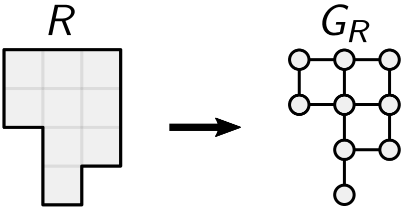
The 180-tromino tiling can also be reduced to the **Maximum Independent Set** problem.



- Transformation from R to G_R:
  - Transform every cell of R to vertices of G_R.
  - Add horizontal, vertical and northeast-diagonal edges.

Forbidden Polyominoes

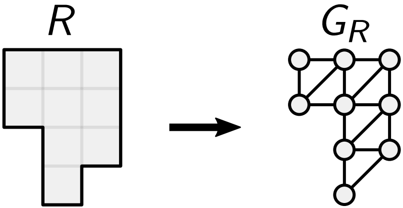
The 180-tromino tiling can also be reduced to the **Maximum Independent Set** problem.



- Transformation from R to G_R:
  - Transform every cell of R to vertices of G_R.
  - Add horizontal, vertical and northeast-diagonal edges.

Forbidden Polyominoes

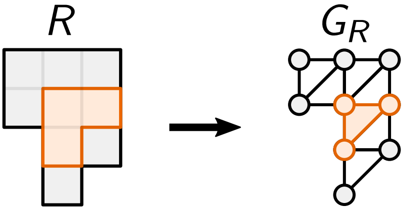
The 180-tromino tiling can also be reduced to the **Maximum Independent Set** problem.



- Transformation from R to G_R:
  - Transform every cell of R to vertices of G_R.
  - Add horizontal, vertical and northeast-diagonal edges.

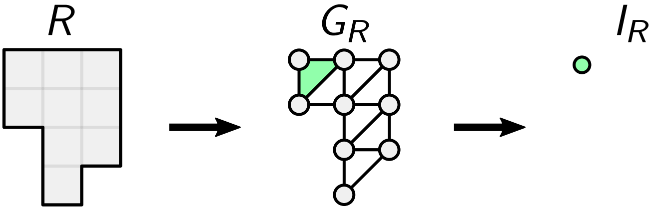
Forbidden Polyominoes

The 180-tromino tiling can also be reduced to the **Maximum Independent Set** problem.



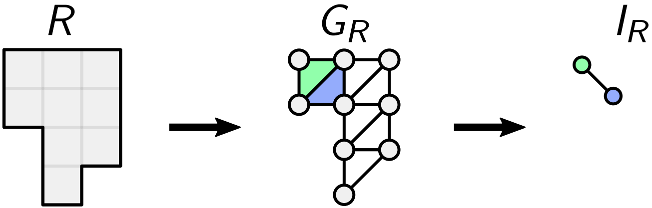
- Transformation from R to G_R:
  - Transform every cell of R to vertices of G_R.
  - Add horizontal, vertical and northeast-diagonal edges.

The 180-tromino tiling can also be reduced to the **Maximum Independent Set** problem.



- Transformation from R to G_R:
  - Transform every cell of R to vertices of G_R.
  - Add horizontal, vertical and northeast-diagonal edges.
- Transformation from G_R to I_R:
  - Transform every 3-cycle of G_R to vertices of I_R.
  - Add an edge where 3-cycles intersects.

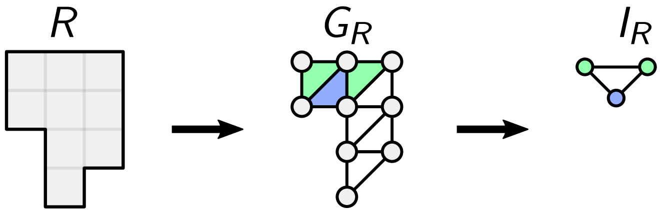
The 180-tromino tiling can also be reduced to the **Maximum Independent Set** problem.



- Transformation from R to G_R:
  - Transform every cell of R to vertices of G_R.
  - Add horizontal, vertical and northeast-diagonal edges.
- Transformation from G_R to I_R:
  - Transform every 3-cycle of G_R to vertices of I_R.
  - Add an edge where 3-cycles intersects.

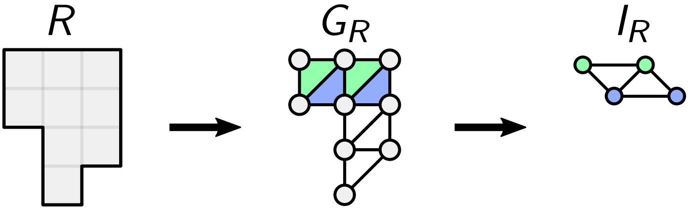
Forbidden Polyominoes

The 180-tromino tiling can also be reduced to the **Maximum Independent Set** problem.



- Transformation from R to G_R:
  - Transform every cell of R to vertices of G_R.
  - Add horizontal, vertical and northeast-diagonal edges.
- Transformation from G_R to I_R:
  - Transform every 3-cycle of G_R to vertices of I_R.
  - Add an edge where 3-cycles intersects.

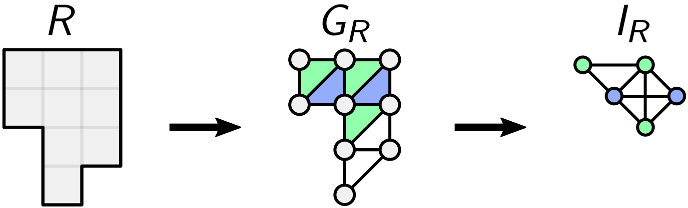
The 180-tromino tiling can also be reduced to the **Maximum Independent Set** problem.



- Transformation from R to G_R:
  - Transform every cell of R to vertices of G_R.
  - Add horizontal, vertical and northeast-diagonal edges.
- Transformation from G_R to I_R:
  - Transform every 3-cycle of G_R to vertices of I_R.
  - Add an edge where 3-cycles intersects.

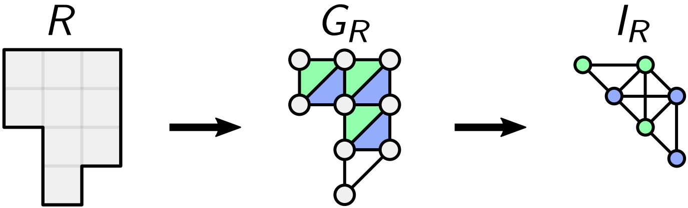
Forbidden Polyominoes

The 180-tromino tiling can also be reduced to the **Maximum Independent Set** problem.



- Transformation from R to G_R:
  - Transform every cell of R to vertices of G_R.
  - Add horizontal, vertical and northeast-diagonal edges.
- Transformation from G_R to I_R:
  - Transform every 3-cycle of G_R to vertices of I_R.
  - Add an edge where 3-cycles intersects.

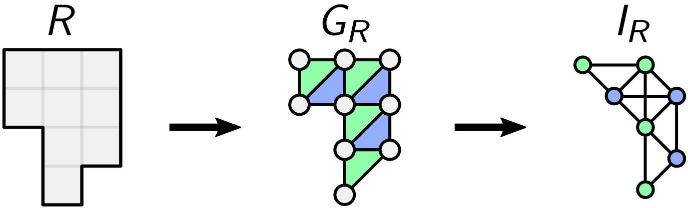
The 180-tromino tiling can also be reduced to the **Maximum Independent Set** problem.



- Transformation from R to G_R:
  - Transform every cell of R to vertices of G_R.
  - Add horizontal, vertical and northeast-diagonal edges.
- Transformation from G_R to I_R:
  - Transform every 3-cycle of G_R to vertices of I_R.
  - Add an edge where 3-cycles intersects.

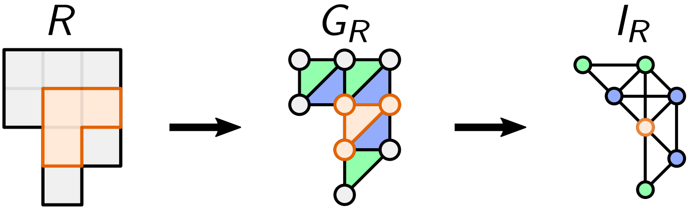
Forbidden Polyominoes

The 180-tromino tiling can also be reduced to the **Maximum Independent Set** problem.



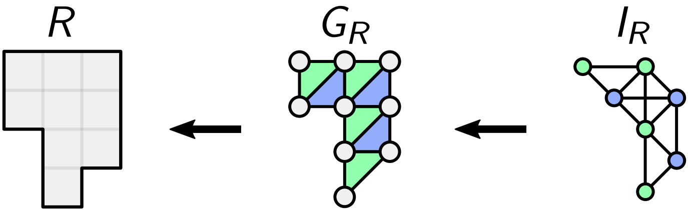
- Transformation from R to G_R:
  - Transform every cell of R to vertices of G_R.
  - Add horizontal, vertical and northeast-diagonal edges.
- Transformation from G_R to I_R:
  - Transform every 3-cycle of G_R to vertices of I_R.
  - Add an edge where 3-cycles intersects.

The 180-tromino tiling can also be reduced to the **Maximum Independent Set** problem.



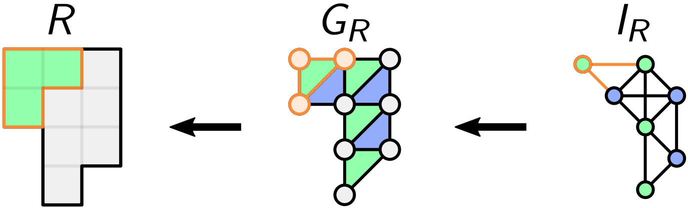
- Transformation from R to G_R:
    - Transform every cell of R to vertices of G_R.
    - Add horizontal, vertical and northeast-diagonal edges.
- Transformation from G_R to I_R:
    - Transform every 3-cycle of G_R to vertices of I_R.
    - Add an edge where 3-cycles intersects.

Forbidden Polyominoes (cont'd)

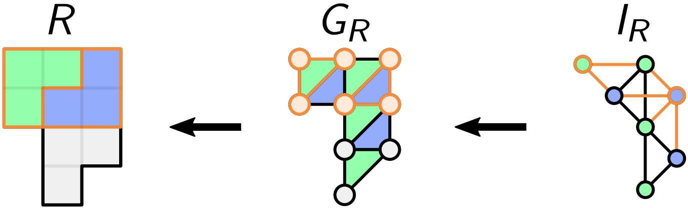


R
G_R
I_R

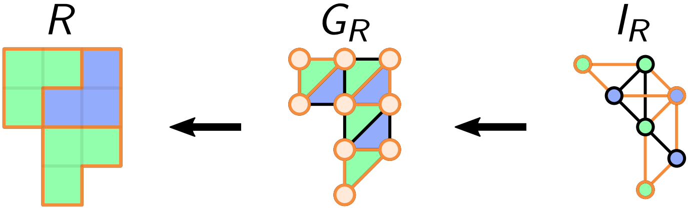
Forbidden Polyominoes (cont'd)

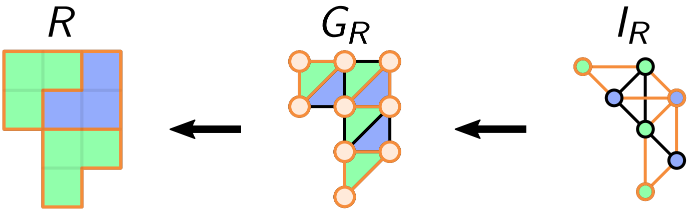


Forbidden Polyominoes (cont'd)



Forbidden Polyominoes (cont'd)





Theorem

Maximum Independent Set of I_R is equal to $\frac{|R|}{3}$
$\iff$ *R has a 180-tromino tiling .*

where $|R|$ the number of cells in a region R.

Forbidden Polyominoes (cont'd)

Forbidden Polyominoes (cont'd)

If I_G is **claw-free**, i.e., does not contain a claw as induced graph, then computing Maximum Independent Set can be computed in polynomial time.

If I_G is **claw-free**, i.e., does not contain a claw as induced graph, then computing Maximum Independent Set can be computed in polynomial time.

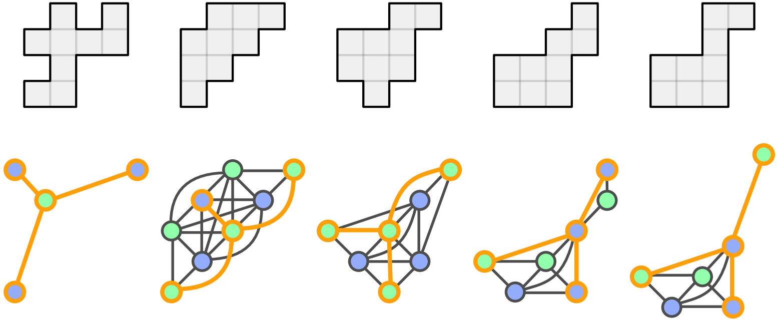
Forbidden Polyominoes (cont'd)

If I_G is **claw-free**, i.e., does not contain a claw as induced graph, then computing Maximum Independent Set can be computed in polynomial time.

The following five polyominoes generates a distinct I_G with a **claw** in it.

If I_G is **claw-free**, i.e., does not contain a claw as induced graph, then computing Maximum Independent Set can be computed in polynomial time.

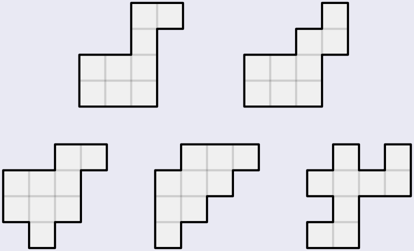
The following five polyominoes generates a distinct I_G with a **claw** in it.



Forbidden Polyominoes (cont'd)

Theorem

*If a region R **doesn't** contains a rotated, reflected or sheared **forbidden polyomino**, then 180-tromino tiling can be computed in a polynomial time.*



Open Problems

1. *Hardness of tiling the Aztec rectangle with a given number of defects.* We saw that an Aztec rectangle with 0 or 1 defects can be covered with L-trominoes in polynomial time, whereas in general the problem is NP-complete when the Aztec rectangle has an unknown number of defects. It is open if there exists a polynomial time algorithm for deciding a tiling for an Aztec rectangle with a given number of defects.

Open Problems

1. *Hardness of tiling the Aztec rectangle with a given number of defects.* We saw that an Aztec rectangle with 0 or 1 defects can be covered with L-trominoes in polynomial time, whereas in general the problem is NP-complete when the Aztec rectangle has an unknown number of defects. It is open if there exists a polynomial time algorithm for deciding a tiling for an Aztec rectangle with a given number of defects.
2. *Tiling of orthogonally-convex regions.* In this work we showed several instances where a tiling can be found in polynomial time. In general, it is open if an orthogonally-convex region with no defects can be covered in polynomial time or if it is NP-complete to decide if a tiling exists.

Enumeration of tilings

- We have not considered the problem of enumerating tromino tilings of the regions described in this talk. In general, there are no such formulas known in the literature for the shapes studied so far.

Enumeration of tilings

- We have not considered the problem of enumerating tromino tilings of the regions described in this talk. In general, there are no such formulas known in the literature for the shapes studied so far.
- Some bounds can be given for the number ($T(n)$) of L-tromino tilings of Aztec diamonds.

Enumeration of tilings

- We have not considered the problem of enumerating tromino tilings of the regions described in this talk. In general, there are no such formulas known in the literature for the shapes studied so far.
- Some bounds can be given for the number ($T(n)$) of L-tromino tilings of Aztec diamonds.

Theorem

If $n = 3k$ for some $k > 0$, then we have

$$T(n) \geq 4T(n-1) + 4\sum_{l=1}^{k-1}(l-1)T(3l) + \sum_{l=1}^{k-1}(l-4)T(3l-1).$$

Enumeration of tilings

- We have not considered the problem of enumerating tromino tilings of the regions described in this talk. In general, there are no such formulas known in the literature for the shapes studied so far.
- Some bounds can be given for the number ($T(n)$) of L-tromino tilings of Aztec diamonds.

Theorem

If $n = 3k$ for some $k > 0$, then we have

$$T(n) \geq 4T(n-1) + 4\sum_{l=1}^{k-1}(l-1)T(3l) + \sum_{l=1}^{k-1}(l-4)T(3l-1).$$

- A similar result will also hold for Aztec Rectangles, but with more parameters as well as increased complexity.

Enumeration of tilings

- We have not considered the problem of enumerating tromino tilings of the regions described in this talk. In general, there are no such formulas known in the literature for the shapes studied so far.
- Some bounds can be given for the number ($T(n)$) of L-tromino tilings of Aztec diamonds.

Theorem

If $n = 3k$ for some $k > 0$, then we have

$$T(n) \geq 4T(n-1) + 4\sum_{l=1}^{k-1}(l-1)T(3l) + \sum_{l=1}^{k-1}(l-4)T(3l-1).$$

- A similar result will also hold for Aztec Rectangles, but with more parameters as well as increased complexity.

It appears that these bound can be improved substantially.

Thank you!

Thank you!

You can try the tetrasected cell tiling program in your phone browser: http://bit.ly/TetrasectedTiling