# A new proof of Thiant's Lemma

## Modeling the situation as a gift for Eric's 60th birthday

Martín Matamala

Departamento de Ingeniería Matemática
Centro de Modelamiento Matemático
Universidad de Chile, Santiago, Chile.

Valparaíso, 24 November 2011

# A gift for Eric Goles's 60th birthday

## A mathematical Puzzle

# The gift

## As I found in internet

# Inside the gift

## The board

# Inside the gift

## A box of dominoes

# Inside the gift

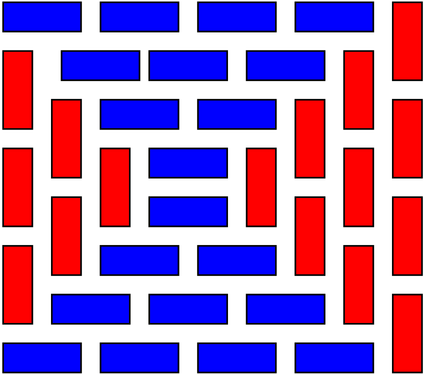
## Instructions

- This is a two players game.
- From the *initial situation* Player 1 removes a domino in *part A* of a *feasible configuration F*.
- Player 2, by using *legal movement*, creates a place in *part B* of $F$ to insert the domino removed by player 1.

# We need an expert!

# More detailed instructions

Initial Situation: A Domino Tiling

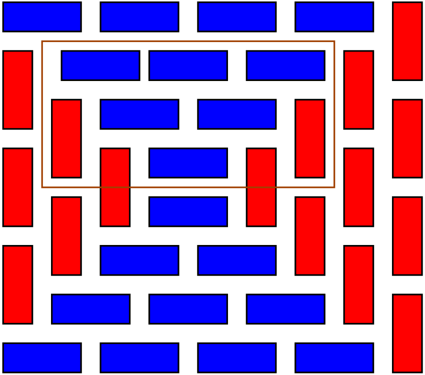


Instructions:

- This is a two players game.
- From the *initial situation* Player 1 removes a domino in *part A* of a *feasible configuration F*.
- Player 2, creates a place in *part B* of *F* by using *legal movement*, to insert the domino removed by player 1.

# More detailed instructions

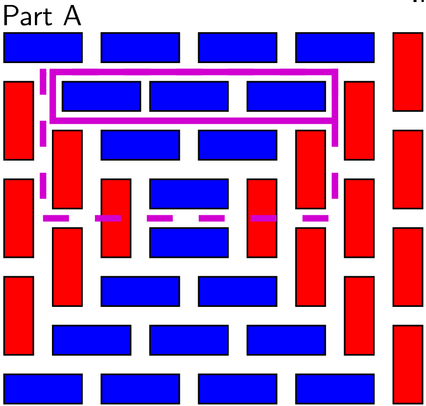
Feasible Configuration



Instructions:

- This is a two players game.
- From the *initial situation* Player 1 removes a domino in *part A* of a *feasible configuration F*.
- Player 2, creates a place in *part B* of *F* by using *legal movement*, to insert the domino removed by player 1.

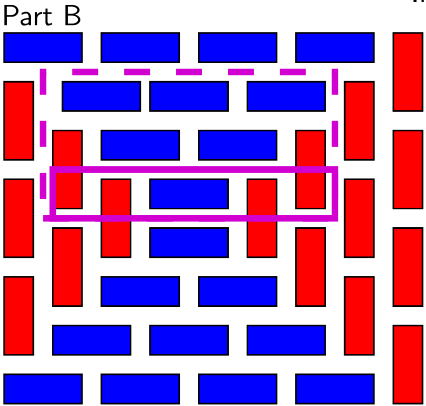
# More detailed instructions




Instructions:

- This is a two players game.
- From the *initial situation* Player 1 removes a domino in *part A* of a *feasible configuration $F$.*
- Player 2, creates a place in *part B* of $F$ by using *legal movement*, to insert the domino removed by player 1.

# More detailed instructions

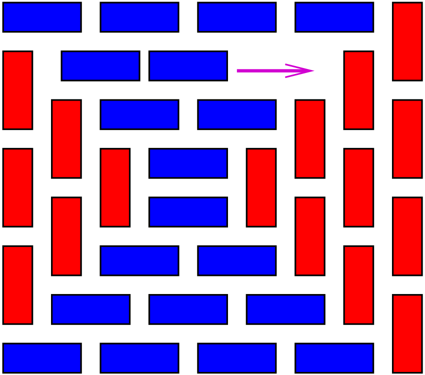



Instructions:

- This is a two players game.
- From the *initial situation* Player 1 removes a domino in *part A* of a *feasible configuration F*.
- Player 2, creates a place in *part B* of *F* by using *legal movement*, to insert the domino removed by player 1.

# More detailed instructions

Legal Movement: dominoes may glide on the board in empty space

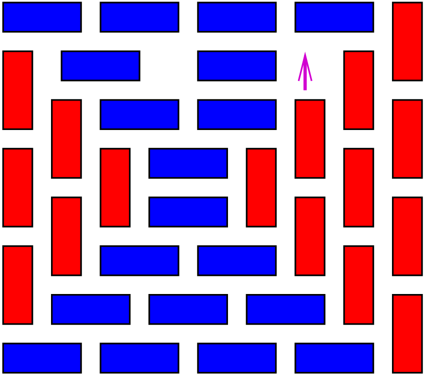


Instructions:

- This is a two players game.
- From the *initial situation* Player 1 removes a domino in *part A* of a *feasible configuration F*.
- Player 2, creates a place in *part B* of *F* by using *legal movement,* to insert the domino removed by player 1.

# More detailed instructions

Legal Movement: dominoes may glide on the board in empty space

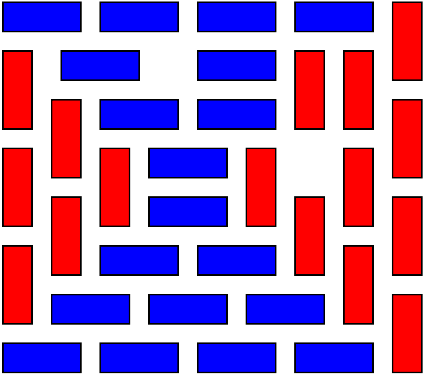


Instructions:

- This is a two players game.
- From the *initial situation* Player 1 removes a domino in *part A* of a *feasible configuration F*.
- Player 2, creates a place in *part B* of *F* by using *legal movement,* to insert the domino removed by player 1.

# More detailed instructions

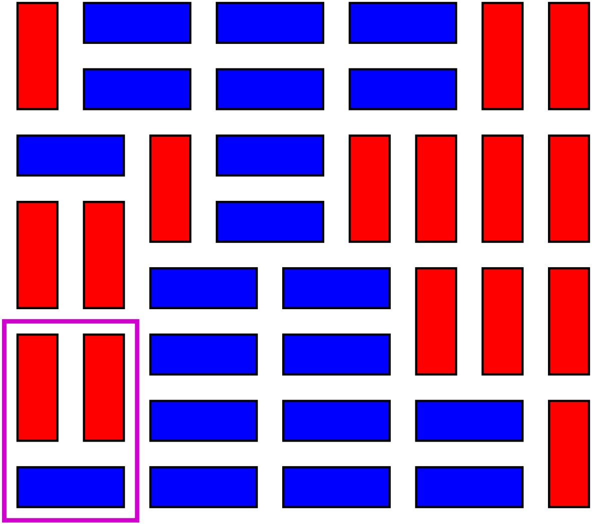
Legal Movement: dominoes may glide on the board in empty space



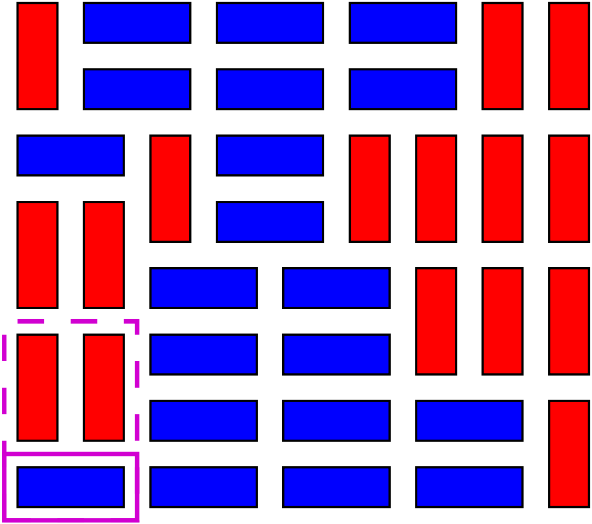
Instructions:

- This is a two players game.
- From the *initial situation* Player 1 removes a domino in *part A* of a *feasible configuration F*.
- Player 2, creates a place in *part B* of *F* by using *legal movement,* to insert the domino removed by player 1.

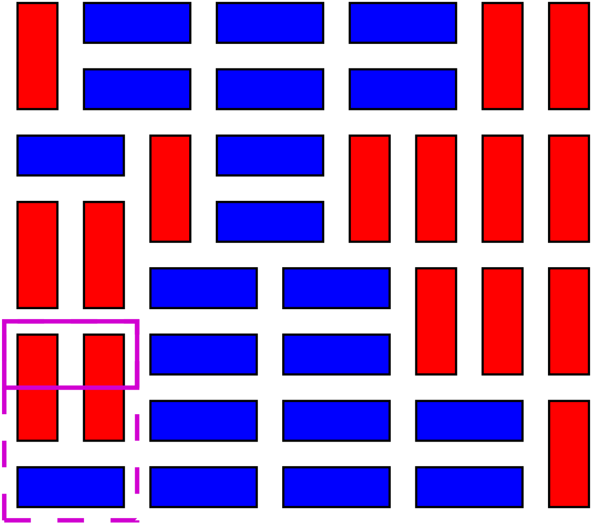
# Easy Example



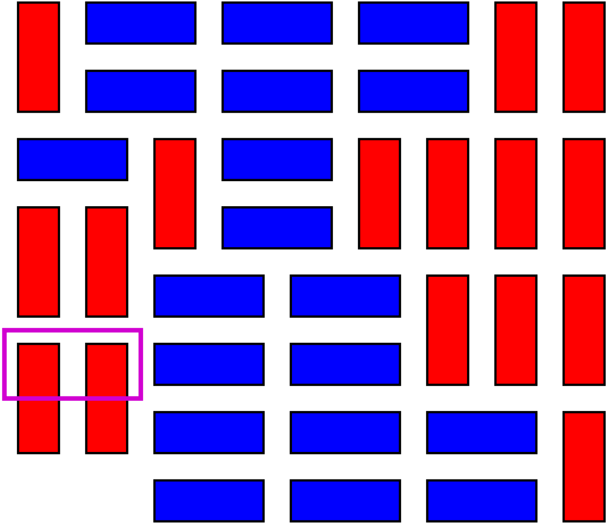
# Easy Example



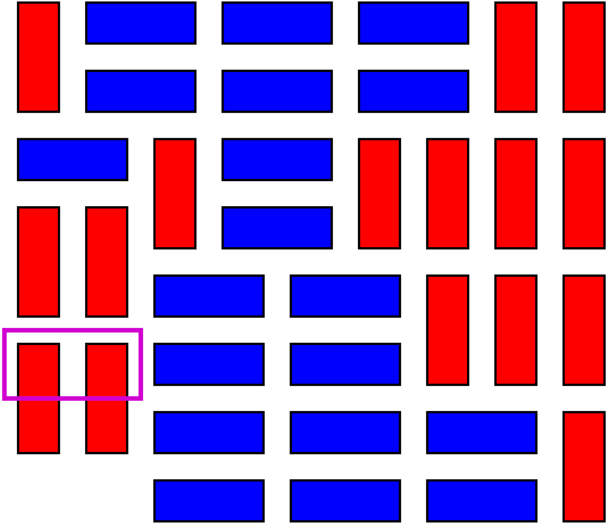
## Easy Example



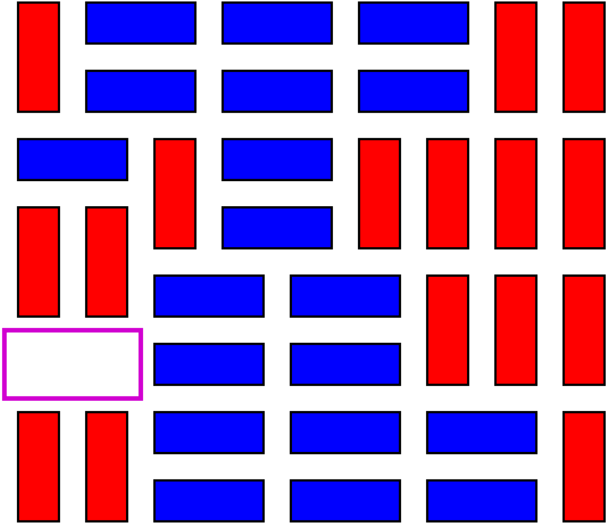
# Easy Example



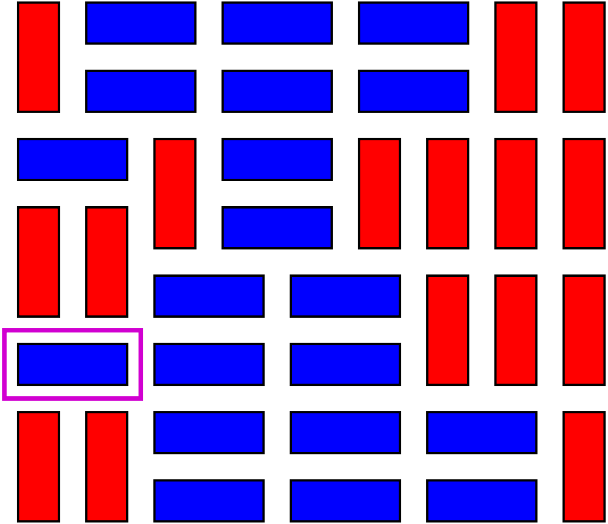
# Easy Example



# Easy Example



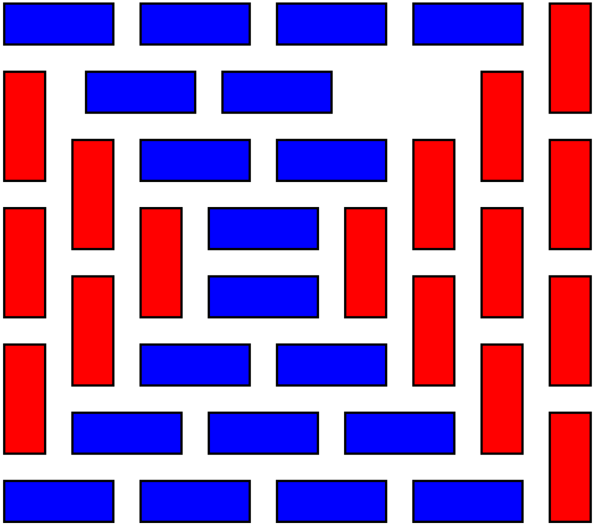
# Easy Example



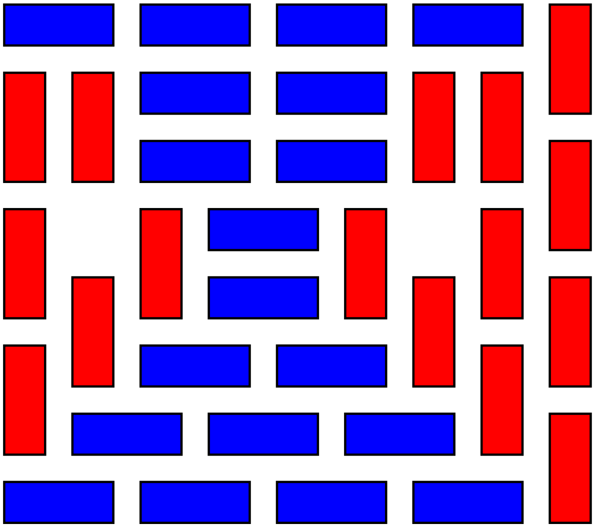
# Last rule

*Player 2 wins if *he can always place the domino. Otherwise, Player 1 wins.*

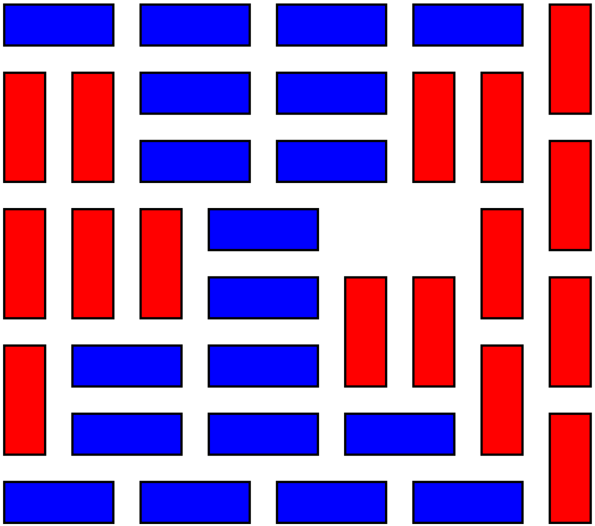
# Last example before to play



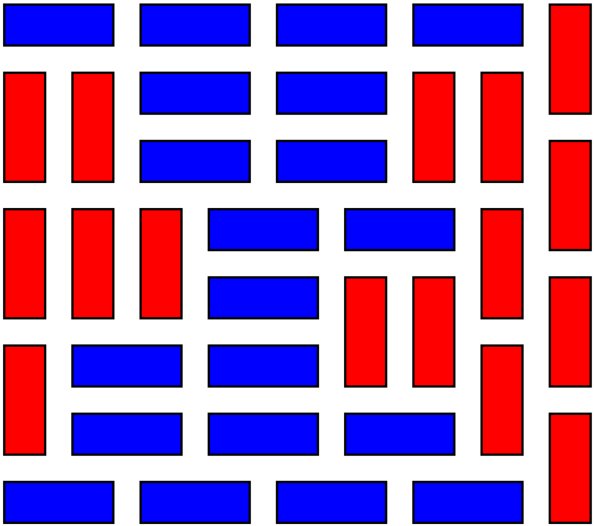
# Last example before to play



# Last example before to play

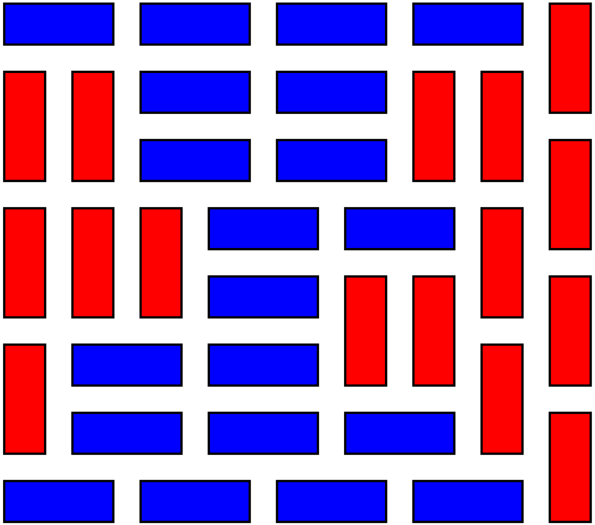


# Last example before to play



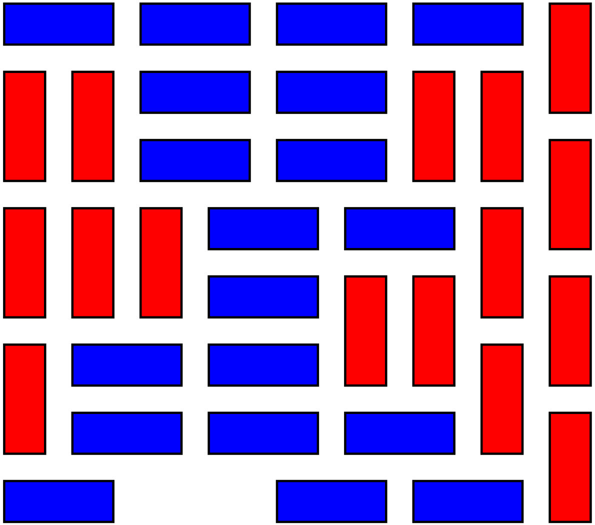
# Playing

I play first,



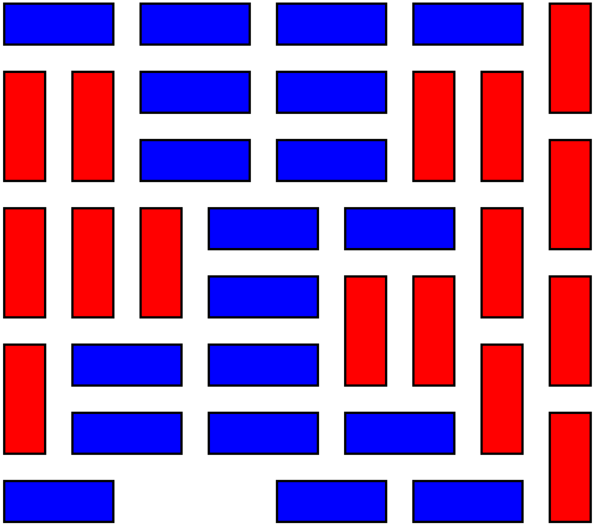
# Playing

I play first,



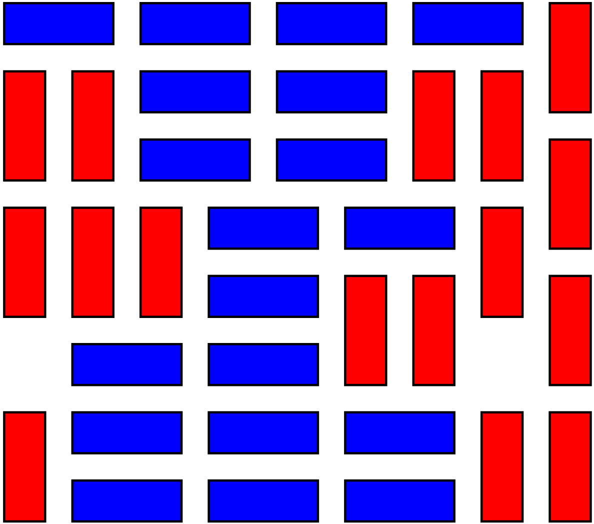
# Playing

I play first, Eric plays



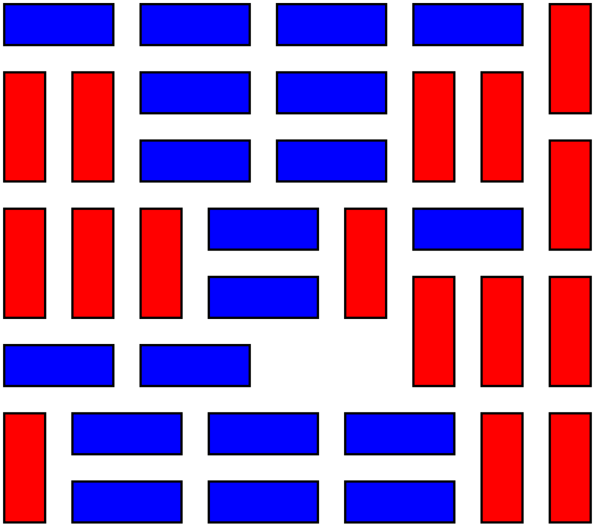
# Playing

I play first, Eric plays a set of movements,



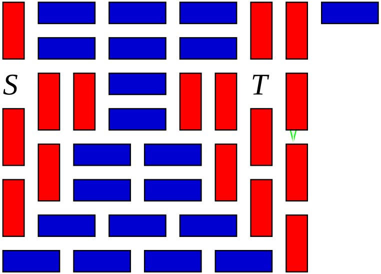
# Playing

I play first, Eric plays  a set of movements, a second set of movements, and he wins.



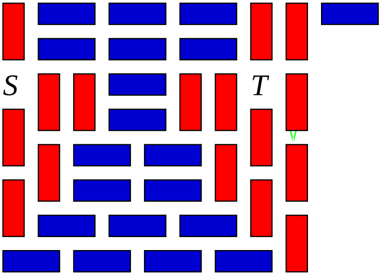
# Eric may always win!

## A proof



# Eric may always win!

## A proof

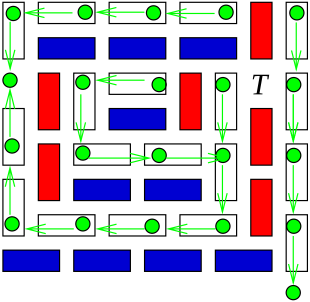



Let $S$ and $T$ be the two empty spaces.

- $G$: graph with vertices the positions at even distance of $S$.
- Arcs in $G$ are $(c, c')$ if one domino covers $c$ and the space between $c$ and $c'$ or $(c, v_\infty)$ if one domino covers $c$ and one space in the border.

# Eric may always win!

## A proof

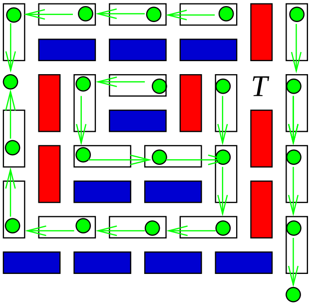



Let $S$ and $T$ be the two empty spaces.

- $G$: graph with vertices the positions at even distance of $S$.
- Arcs in $G$ are $(c, c')$ if one domino covers $c$ and the space between $c$ and $c'$ or $(c, v_\infty)$ if one domino covers $c$ and one space in the border.

# Eric may always win!

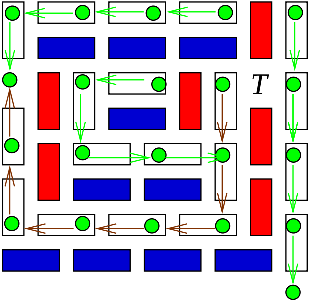
## A proof




- Each vertex of $G$ has exactly one outgoing arc.
- $G$ is acyclic: any cycle contains an odd number of positions. But any cycle either contains both $S$ and $T$, or none.
- A path in $G$ ending in $S$ defines a way to move the empty space by using legal movements.

# Eric may always win!

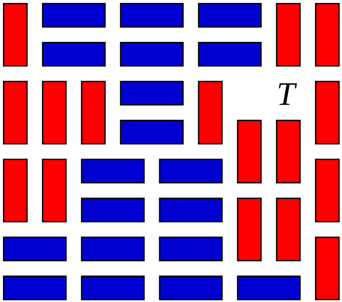
## A proof




- One of the path starting in the horizontal neighbors of $T$ must end in $S$.

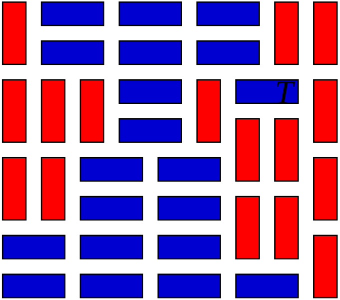
# Eric may always win!

## A proof



# Eric may always win!

## A proof




This finishes the proof.

# The title makes sense

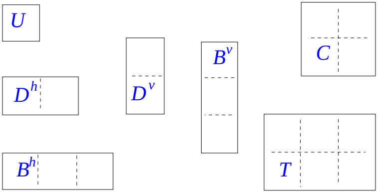
## A new proof of Thiant's Lemma

[Thiant 2006] If $(r, s)$ are the projections of a tiling with dominoes of a rectangular region $R$, and $r_i < r_{i-1}$, then $(r', s)$ are also the proyections of a tiling with dominoes of $R$, where $r'_j = r_j$, for all $j \neq i-1, i+1$, $r'_{i-1} = r_{i-1} - 1$ and $r'_{i+1} = r_{i+1} + 1$

# Tilings

## Rectangular tiles: a taxonomy

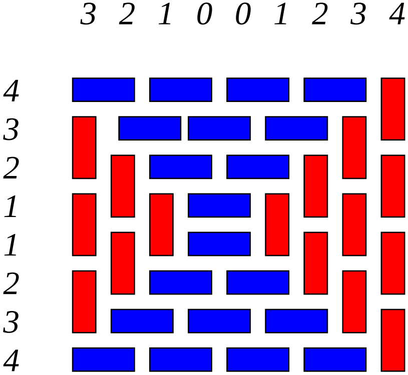
- $U$: unitary.
- *Unidimensionals:* one side's size $= 1$ and the other $\geq 2$.
  $D^h$ and $D^v$: horizontal dominoes and vertical dominoes.
  $B^h$ and $B^v$: horizontal bars and vertical bars.
- *Bi-dimensionals:* both sides of size at least 2.
  $C$ squares.



# Reconstruction of Domino tilings

## Polynomial time algorithms

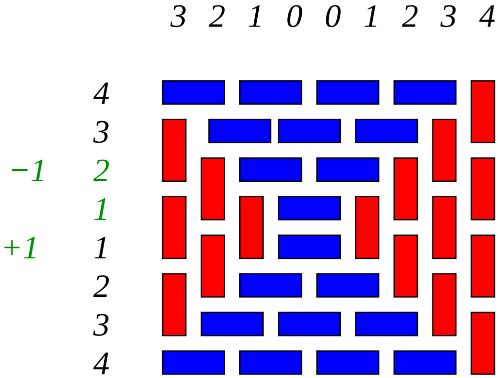
[Thiant, 2006.] The domino reconstruction problem can be reduced to the reconstruction problem with tiles $U$ and $D^h$ in polynomial time.



# Reconstruction of Domino tilings

## Polynomial time algorithms

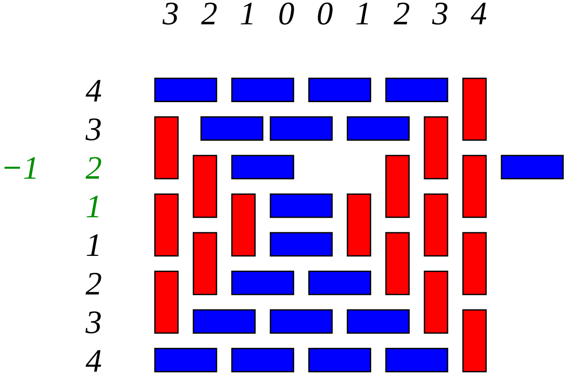
[Thiant, 2006.] The domino reconstruction problem can be reduced to the reconstruction problem with tiles $U$ and $D^h$ in polynomial time.



# Reconstruction of Domino tilings

## Polynomial time algorithms

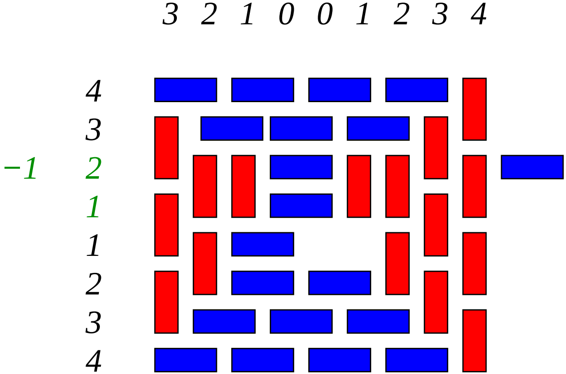
[Thiant, 2006.] The domino reconstruction problem can be reduced to the reconstruction problem with tiles $U$ and $D^h$ in polynomial time.



# Reconstruction of Domino tilings

## Polynomial time algorithms

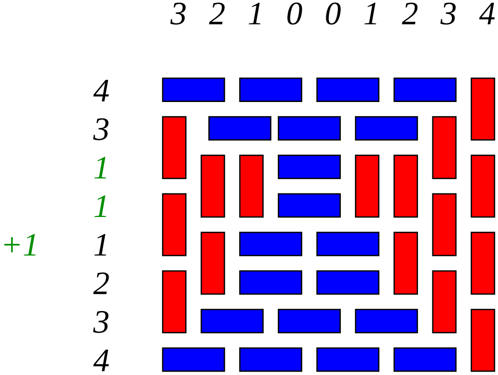
[Thiant, 2006.] The domino reconstruction problem can be reduced to the reconstruction problem with tiles $U$ and $D^h$ in polynomial time.



# Reconstruction of Domino tilings

## Polynomial time algorithms

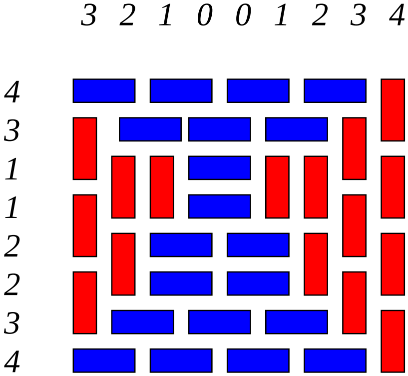
[Thiant, 2006.] The domino reconstruction problem can be reduced to the reconstruction problem with tiles $U$ and $D^h$ in polynomial time.



# Reconstruction of Domino tilings

## Polynomial time algorithms

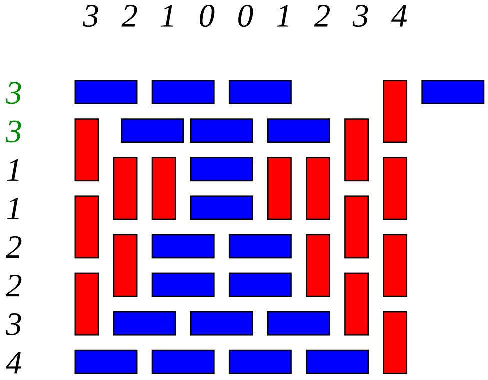
[Thiant, 2006.] The domino reconstruction problem can be reduced to the reconstruction problem with tiles $U$ and $D^h$ in polynomial time.



# Reconstruction of Domino tilings

## Polynomial time algorithms

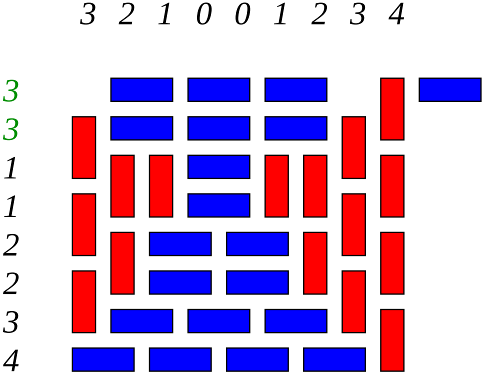
[Thiant, 2006.] The domino reconstruction problem can be reduced to the reconstruction problem with tiles $U$ and $D^h$ in polynomial time.



# Reconstruction of Domino tilings

## Polynomial time algorithms

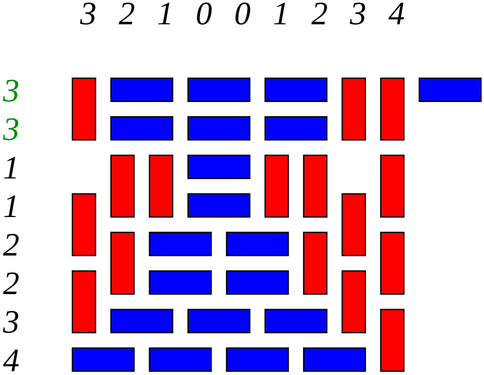
[Thiant, 2006.] The domino reconstruction problem can be reduced to the reconstruction problem with tiles $U$ and $D^h$ in polynomial time.



# Reconstruction of Domino tilings

## Polynomial time algorithms

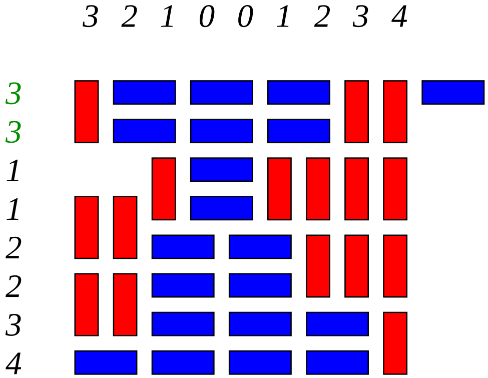
[Thiant, 2006.] The domino reconstruction problem can be reduced to the reconstruction problem with tiles $U$ and $D^h$ in polynomial time.



# Reconstruction of Domino tilings

## Polynomial time algorithms

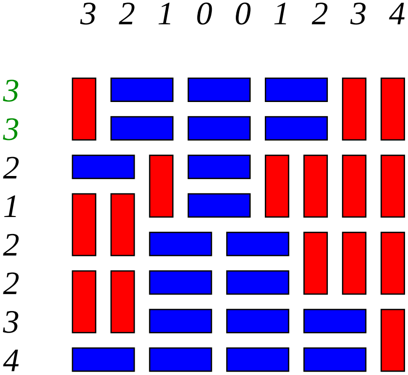
[Thiant, 2006.] The domino reconstruction problem can be reduced to the reconstruction problem with tiles $U$ and $D^h$ in polynomial time.



# Reconstruction of Domino tilings

## Polynomial time algorithms

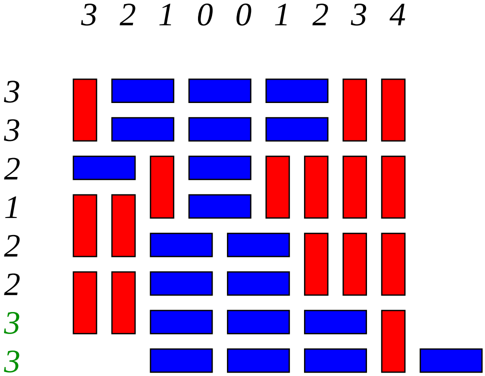
[Thiant, 2006.] The domino reconstruction problem can be reduced to the reconstruction problem with tiles $U$ and $D^h$ in polynomial time.



# Domino tilings

## Polynomial time algorithms

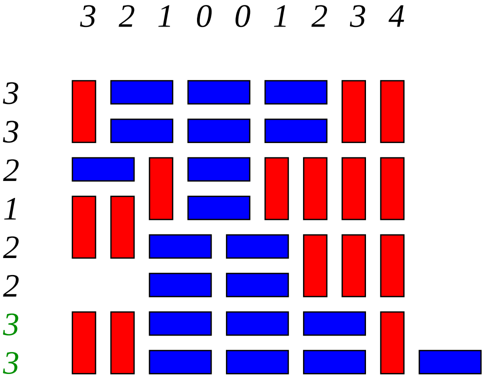
[Thiant, 2006.] The domino reconstruction problem can be reduced to the reconstruction problem with tiles $U$ and $B^h$ in polynomial time.



# Domino tilings

## Polynomial time algorithms

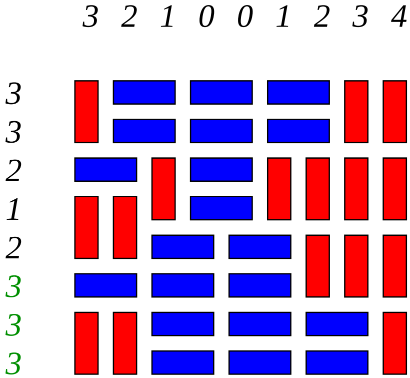
[Thiant, 2006.] The domino reconstruction problem can be reduced to the reconstruction problem with tiles $U$ and $B^h$ in polynomial time.



# Domino tilings

## Polynomial time algorithms

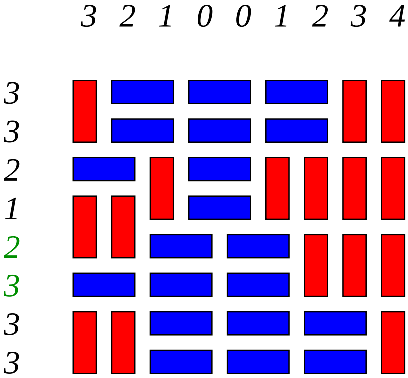
[Thiant, 2006.] The domino reconstruction problem can be reduced to the reconstruction problem with tiles $U$ and $B^h$ in polynomial time.



# Domino tilings

## Polynomial time algorithms

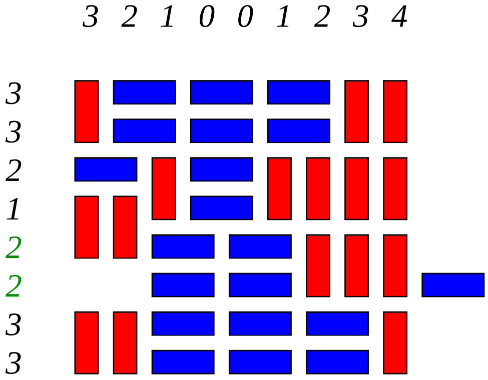
[Thiant, 2006.] The domino reconstruction problem can be reduced to the reconstruction problem with tiles $U$ and $B^h$ in polynomial time.



# Domino tilings

## Polynomial time algorithms

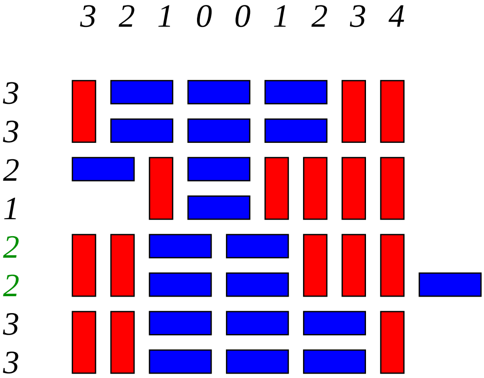
[Thiant, 2006.] The domino reconstruction problem can be reduced to the reconstruction problem with tiles $U$ and $B^h$ in polynomial time.



# Domino tilings

## Polynomial time algorithms

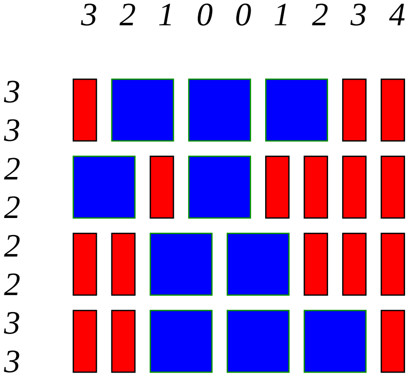
[Thiant, 2006.] The domino reconstruction problem can be reduced to the reconstruction problem with tiles $U$ and $B^h$ in polynomial time.



# Domino tilings

## Polynomial time algorithms

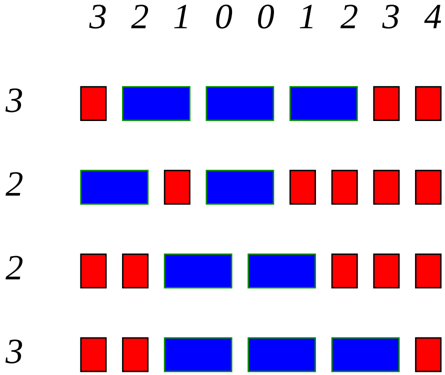
[Thiant, 2006.] The domino reconstruction problem can be reduced to the reconstruction problem with tiles $U$ and $B^h$ in polynomial time.



# Domino tilings

## Polynomial time algorithms

[Thiant, 2006.] The domino reconstruction problem can be reduced to the reconstruction problem with tiles $U$ and $B^h$ in polynomial time.



# Tiling reconstructions

with dominoes and bars

[Dürr, Goles, Rapaport, Rémila, 2003] The reconstruction problem with tiles $U$ and $D^h$ can be solved in polynomial time.

[Thiant, 2006.] The reconstruction problem with tiles $D^v$ and $D^h$ can be solved in polynomial time.

[Dürr, Guiñez, M., 2009.] The reconstruction problem with tiles $B^v$ of two different lengths can be solved in polynomial time.

[Open] Complexity of the reconstruction problem with tiles $D^h$ and $B^v$.