COMP0017
Computability and Complexity Theory

Fabio Zanasi
http://www0.cs.ucl.ac.uk/staff/F.Zanasi/

Lecture fifteen

Previously on COMP0017

We thoroughly investigated **unsolvability** in the context of computation via Turing machines.

We collected various examples and techniques to show that a problem is unsolvable.

In this lecture

This last week focusses on unsolvability in contexts **different from computability theory**.

The case study of this lecture is the unsolvability of **the tiling problem.**

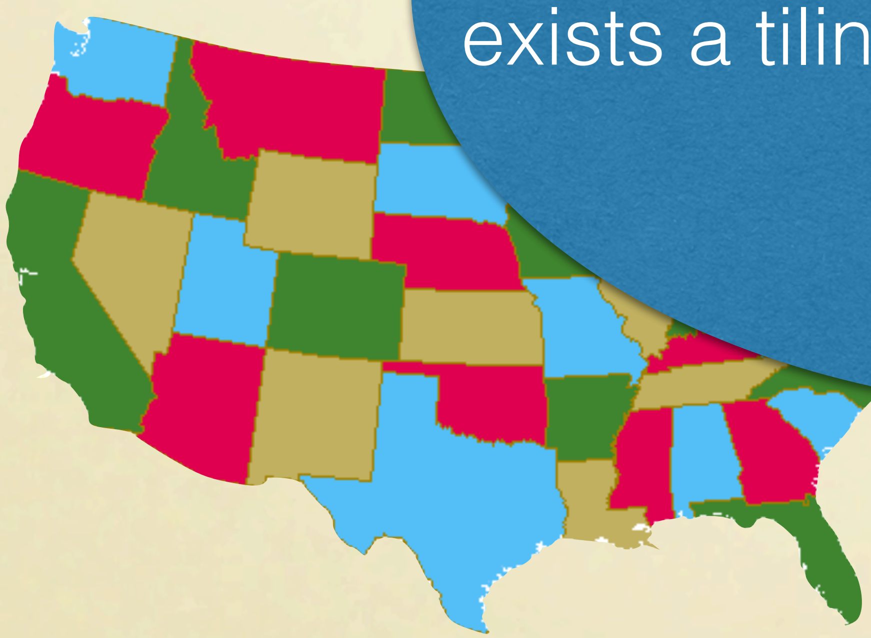
The overall aim is to show that unsolvability is a **pervasive** phenomenon that spreads across different disciplines and problems.

Tiling

Alhambra, Gran

d/fish n.82 (1951)

Given a set of tiles and a set of rules to put them together, does it exists a tiling of the plane respecting the rules?



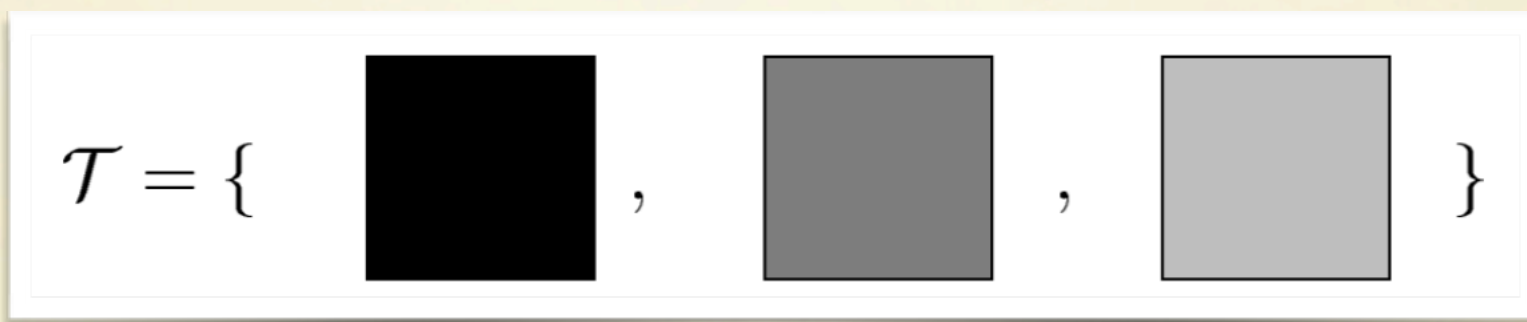
The four colour theorem

A bathroom wall

Tile systems

A **tile system** consists of:

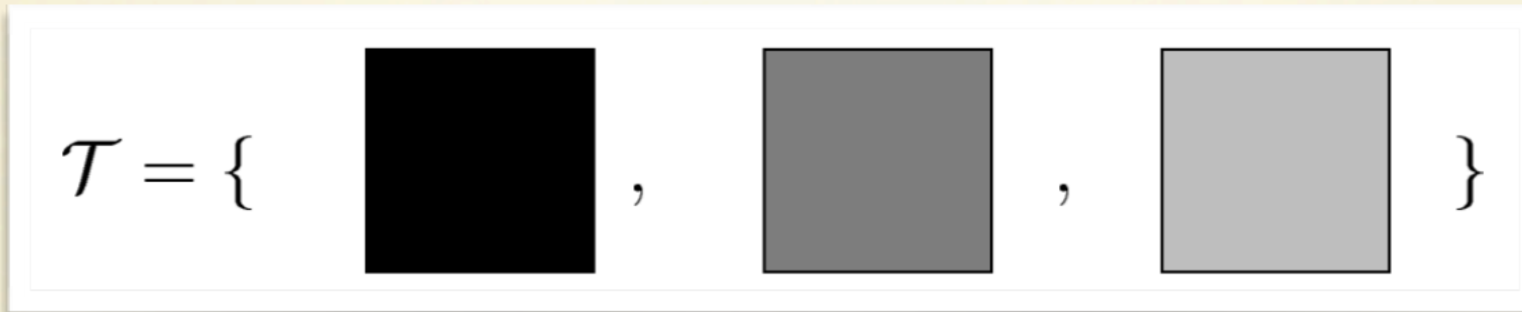
- a set of square tiles, for example



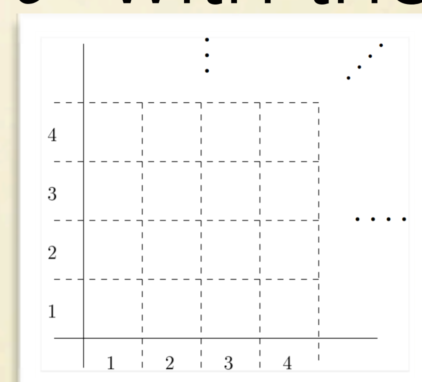
- a chosen element $t_0 \in \mathcal{T}$ called the *origin tile*.

- a set of adjacency rules, which specify which tiles may be placed next to each other.

Tiling

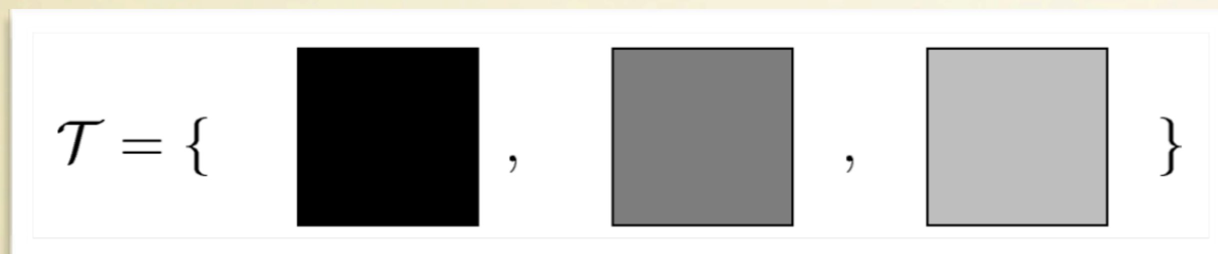


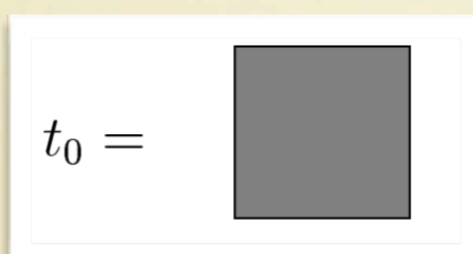
A **tiling** is an arrangement of the tiles in $\mathcal{T}$ with the following properties.



- t_0 is placed in the lower left corner.
- Each tile has another tile arranged on its top and one arranged on its right, with no gaps.
- The adjacency rules are all obeyed.

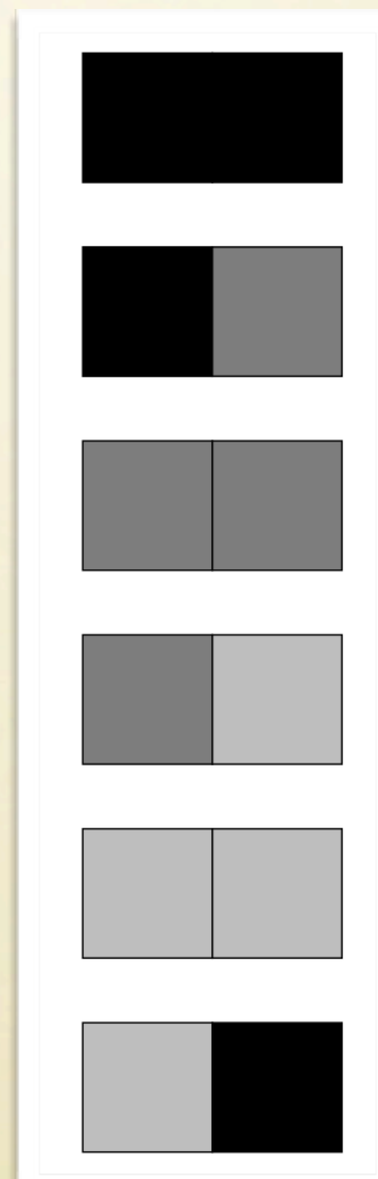
Example



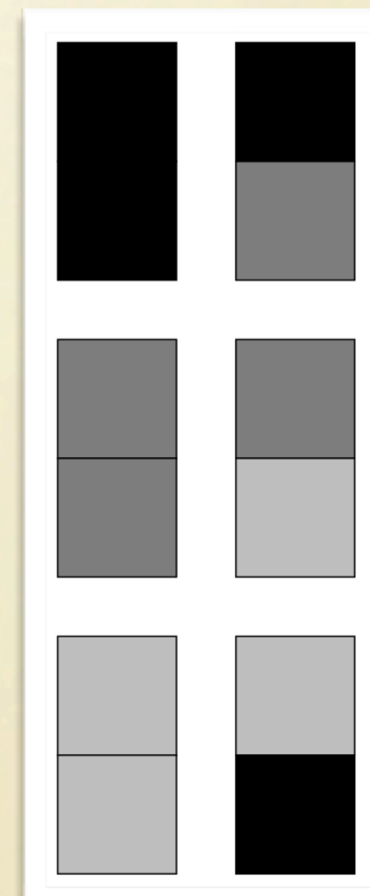


Adjacency rules

Horizontal

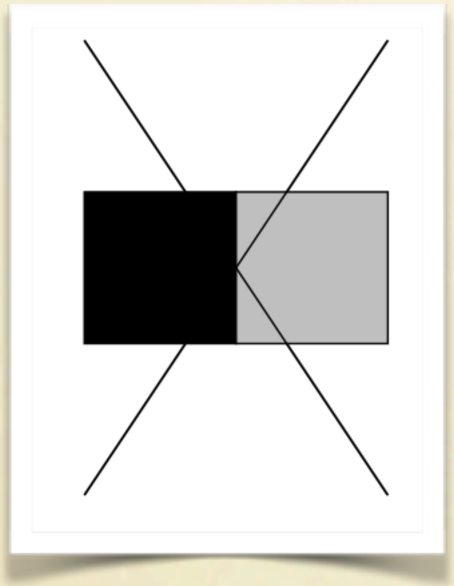


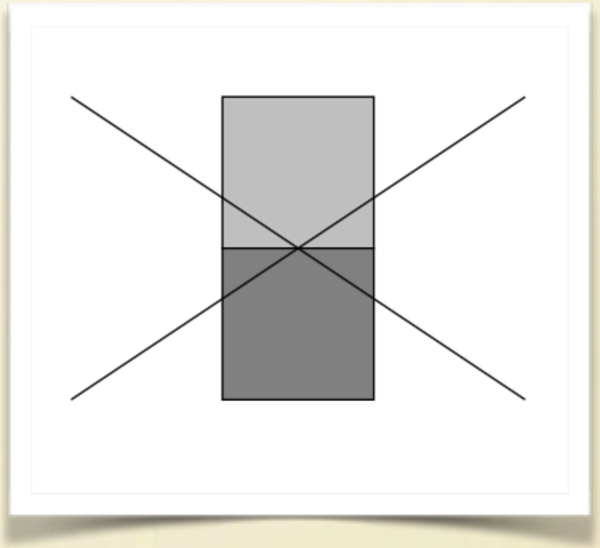
Vertical



Example

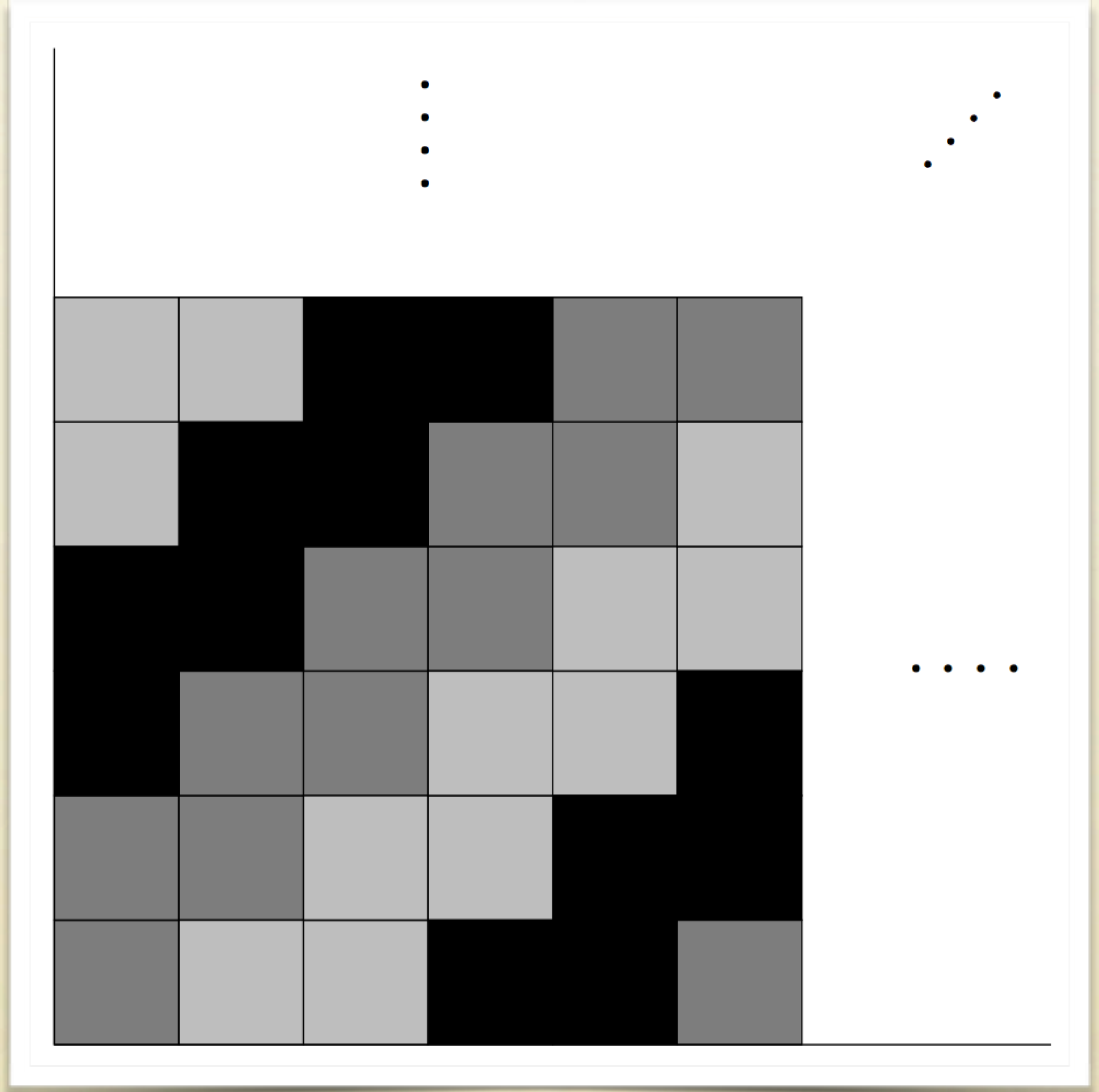
Note that the adjacency rules explicitly do not allow tiles to be placed, for example, as follows:





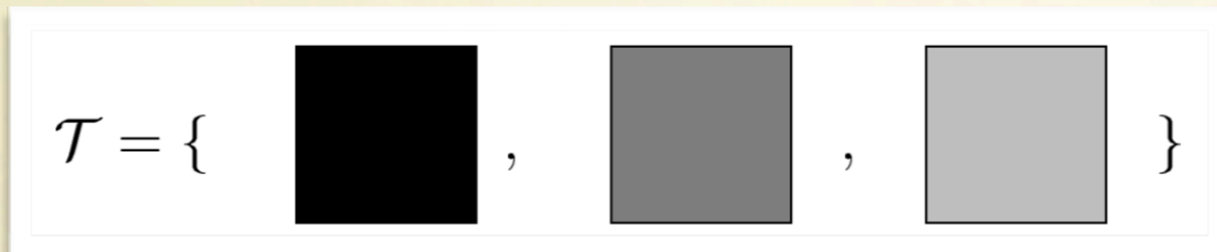
Example

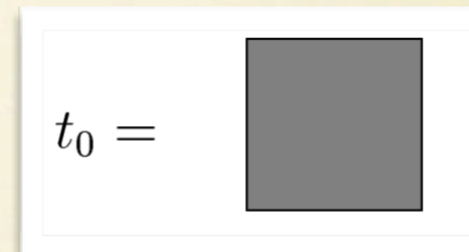
A tiling for this tiling system is for instance



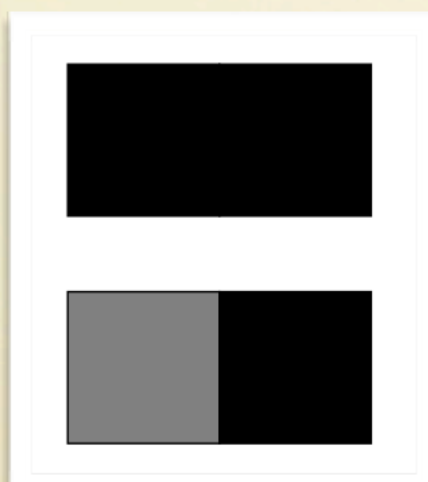
Tiling problems

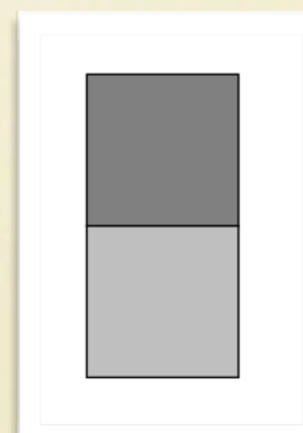
Note that it is not always possible to find a tiling for a given tiling system. For example, with





and adjacency rules





then no tiling exists.

Tiling problems

The problem we want to investigate is the **tiling problem**:

given a tiling system, does a tiling exists?

Hao Wang posed the question (1961) of this problem being solvable by an algorithm.

His PhD student Robert Berger answered negatively (1966), proving that the problem is not only undecidable, but not even recognisable by Turing machines.

In order to prove this, we first need a more formal definition of the data of the problem (tiling systems).

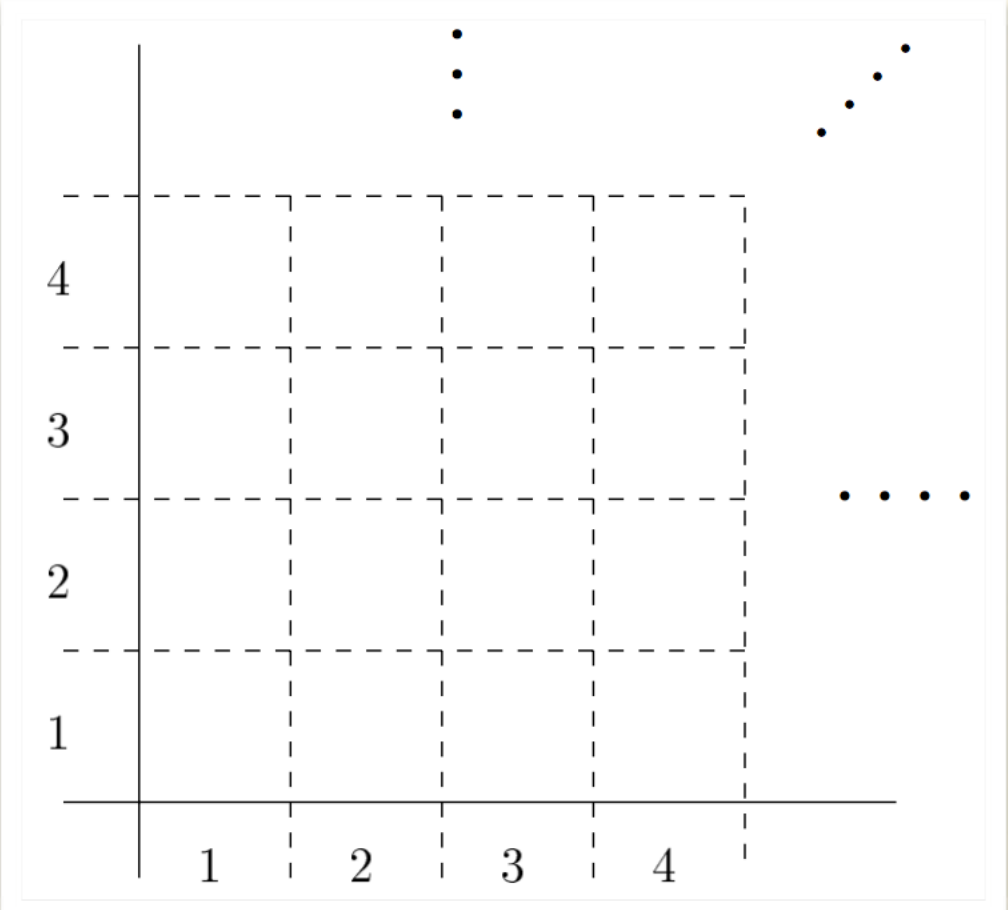
Tiling systems, formally

A **tiling system** can be defined as a tuple $\langle \mathcal{T}, t_0, H, V \rangle$ where

- $\mathcal{T}$ is a set of tiles
- $t_0 \in \mathcal{T}$ is the origin tile
- $H \subseteq \mathcal{T} \times \mathcal{T}$ is a set of horizontal adjacency rules and $V \subseteq \mathcal{T} \times \mathcal{T}$ is a set of vertical adjacency rules.

Tilings, formally

We consider the positive quadrant of the plane to be divided into cells identified by their co-ordinates.



A **tiling** can be seen as a function f : $\mathbb{N} \times \mathbb{N} \to \mathcal{T}$ such that

1. $f(1,1) = t_0$
2. $(f(n,m), f(n,m+1)) \in V$ for all n, m $\in \mathbb{N}$
3. $(f(n,m), f(n+1,m)) \in H$ for all n, m $\in \mathbb{N}$

Tiling problems

We are now back to the **tiling problem**:

given a tiling system, does a tiling exists?

We are going to show that the tiling problem is unrecognisable by showing that the complement of the empty tape halting problem (*ETH*) reduces to it.

$ETH = \{ x \in \Sigma^* \mid x = \text{code}(\mathcal{M}) \text{ and } \mathcal{M} \text{ halts on } \varepsilon. \}$

$ETH^- = \{ x \in \Sigma^* \mid x \neq \text{code}(\mathcal{M}) \text{ for all } \mathcal{M} \text{ or}$

$x = \text{code}(\mathcal{M}) \text{ and } \mathcal{M} \text{ does not halt on } \varepsilon. \}$

Tiling problems

$ETH = \{ x \in \Sigma^* \mid x = \text{code}(\mathcal{M}) \text{ and } \mathcal{M} \text{ halts on } \varepsilon.\}$

$ETH^- = \{ x \in \Sigma^* \mid x \neq \text{code}(\mathcal{M}) \text{ for all } \mathcal{M} \text{ or }$

$x = \text{code}(\mathcal{M}) \text{ and } \mathcal{M} \text{ does not halt on } \varepsilon.\}$

- We saw last week that *ETH* is undecidable but recognisable.
- So, just as with the halting problem and its complement, ETH^- must be not recognisable: otherwise we could devise an algorithm for deciding *ETH*.
- Therefore, reducing ETH^- to the tiling problem implies that the tiling problem is not recognisable by a Turing machine.

Back to the tiling problem

Our strategy in showing the unrecognisability of the tiling problem will be as follows.

1. Show how any Turing machine $\mathcal{M}$ can be transformed into a tiling system $\mathcal{T}_{\mathcal{M}}$.
2. Do so in such a way that

$$\text{code}(\mathcal{M}) \in ETH \quad \leftrightarrow \quad \text{no tiling exists for } \mathcal{T}_{\mathcal{M}}$$

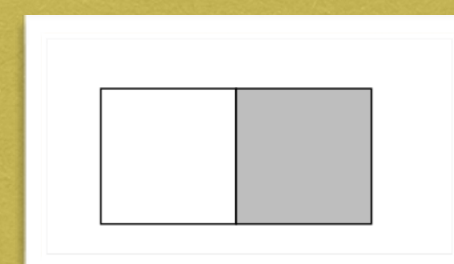
this reduces *ETH* to the complement of the tiling problem, meaning that $\overline{ETH}$ reduces to the tiling problem as wished.

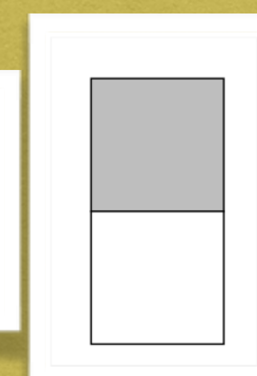
Representing adjacency rules with symbols

Given a tiling system, we can specify its set of tiles and the adjacency rules **simultaneously** by marking the edges of the tiles.

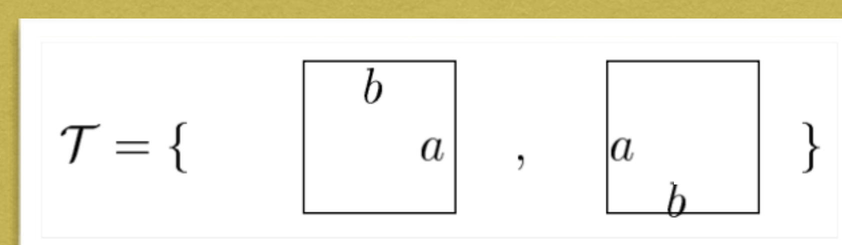
Example

with adjacency rules:





can be represented as

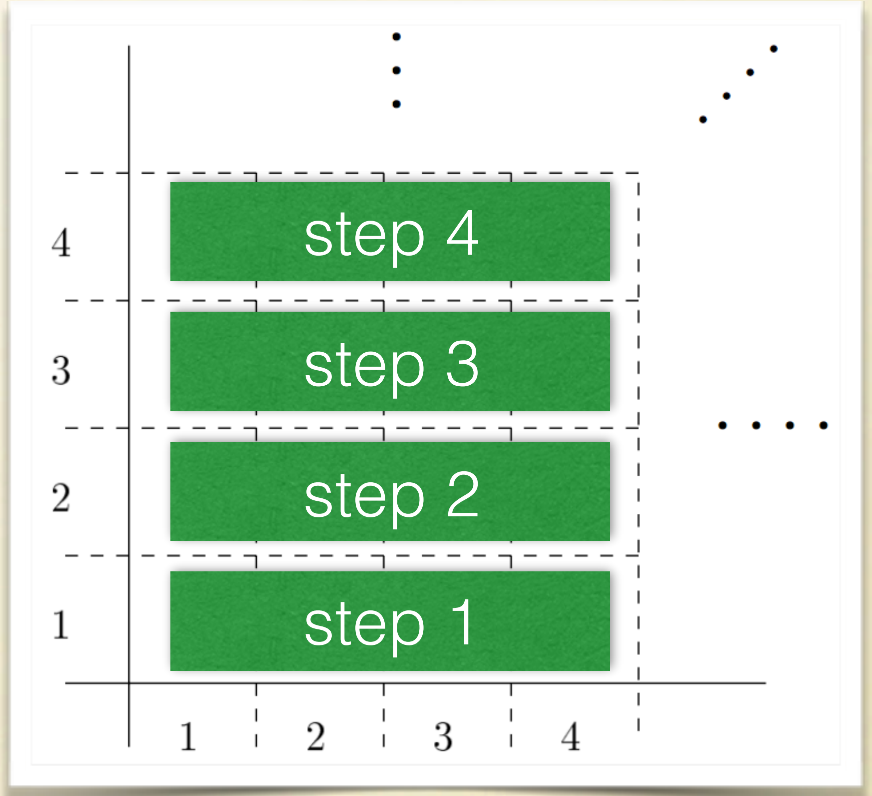


The convention is that tiles can be placed next to each other only if their edges match.

From the TM $\mathcal{M}$ to the tiling system $\mathcal{T}_{\mathcal{M}}$

We now focus on how to transform a Turing machine $\mathcal{M}$ into an appropriate tiling system $\mathcal{T}_{\mathcal{M}}$.

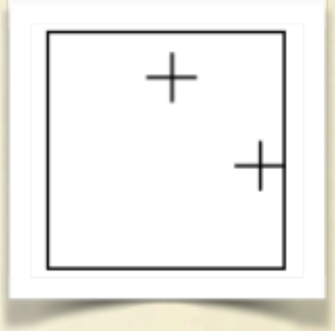
The essential idea is that successive rows of a tiling will represent the tape of at successive steps.



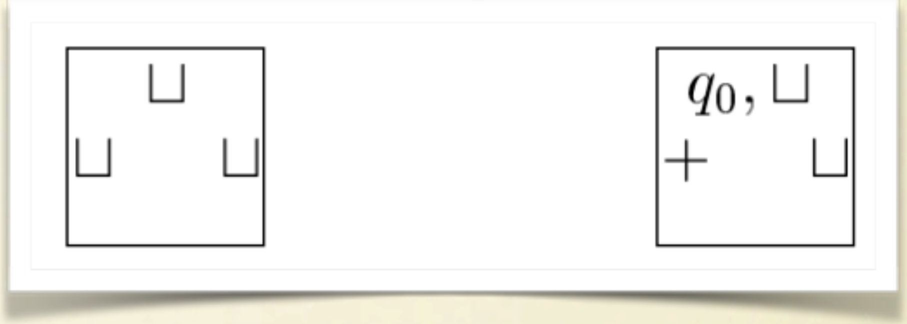
Special tiles will be included to keep track of the current state and current head position.

From $\mathcal{M}$ to $\mathcal{T}_{\mathcal{M}}$: initial tape

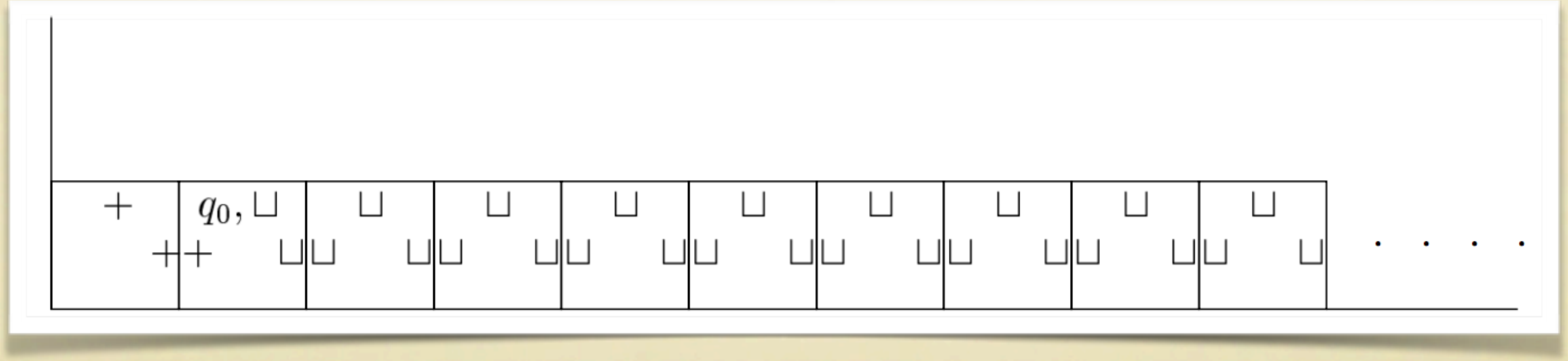
We begin with defining the origin tile of $\mathcal{T}_{\mathcal{M}}$ as



In addition, we include tiles

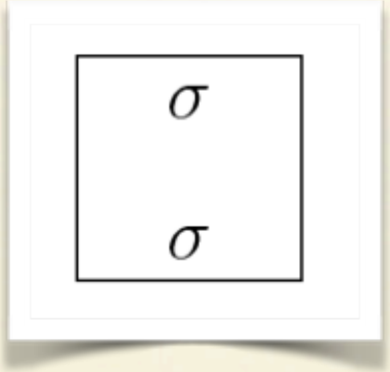


The idea is to force the first row of any tiling to represent the tape at the start of a computation of $\mathcal{M}$ on input ε.



From $\mathcal{M}$ to $\mathcal{T}_{\mathcal{M}}$: symbols

For each $\sigma \in \Sigma$, $\mathcal{T}_{\mathcal{M}}$ includes a tile

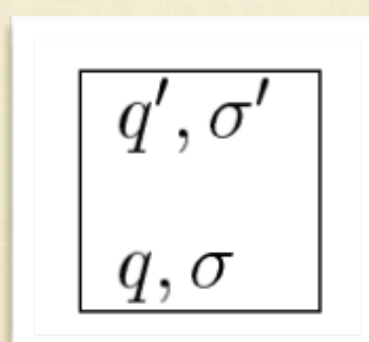


One such tile represents a cell with the symbol σ written in it. Moreover, the disposition of symbols in the tile will ensure that at most one cell is altered at any step in the computation.

From $\mathcal{M}$ to $\mathcal{T}_{\mathcal{M}}$: states

The next set of tiles to be included represents the fact that at each step of the computation $\mathcal{M}$ can change the content of the cell at the current position, and change its state.

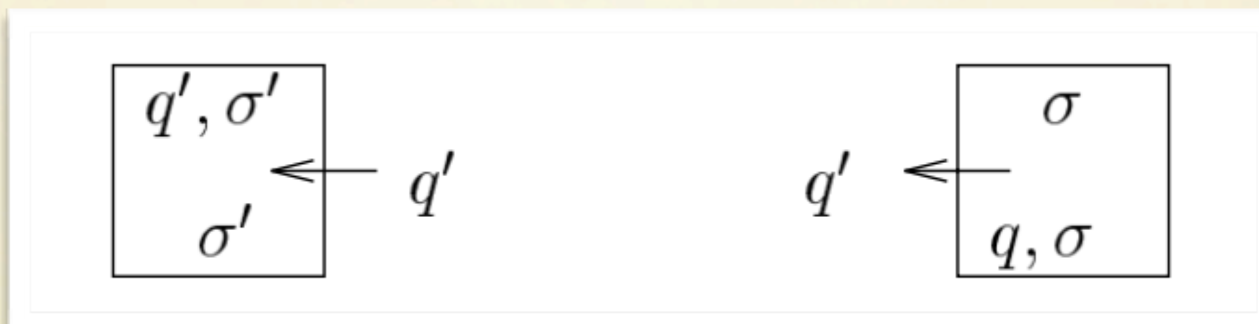
- For each $q \in Q \setminus \{h\}$ and $\sigma \in \Sigma$ for which $\delta(q,\sigma) = (q',\sigma')$ and σ' is not $\leftarrow$ or $\rightarrow$, $\mathcal{T}_{\mathcal{M}}$ includes a tile



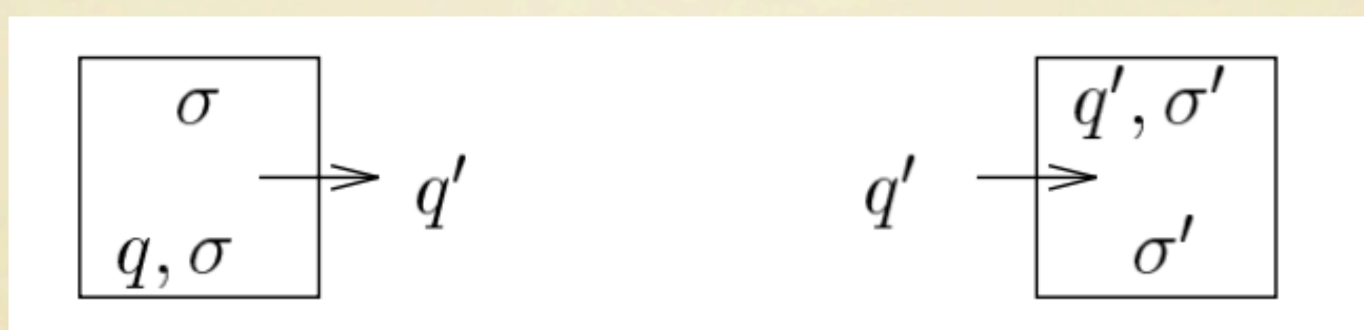
From $\mathcal{M}$ to $\mathcal{T}_{\mathcal{M}}$: head position

We deal with movement of the head.

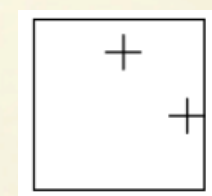
- For each $q \in Q \setminus \{h\}$ and $\sigma \in \Sigma$ for which $\delta(q,\sigma) = (q', \leftarrow)$, and for each $\sigma' \in \Sigma$, $\mathcal{T}_{\mathcal{M}}$ includes tiles

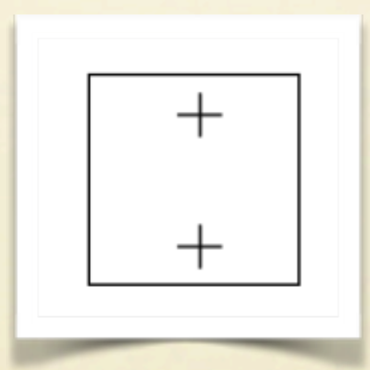


- For each $q \in Q \setminus \{h\}$ and $\sigma \in \Sigma$ for which $\delta(q,\sigma) = (q', \rightarrow)$, and for each $\sigma' \in \Sigma$, $\mathcal{T}_{\mathcal{M}}$ includes tiles



From $\mathcal{M}$ to $\mathcal{T}_{\mathcal{M}}$: placeholder

Finally, $\mathcal{T}_{\mathcal{M}}$ includes a placeholder tile that in tilings of $\mathcal{T}_{\mathcal{M}}$ goes on the top of the origin tile  .

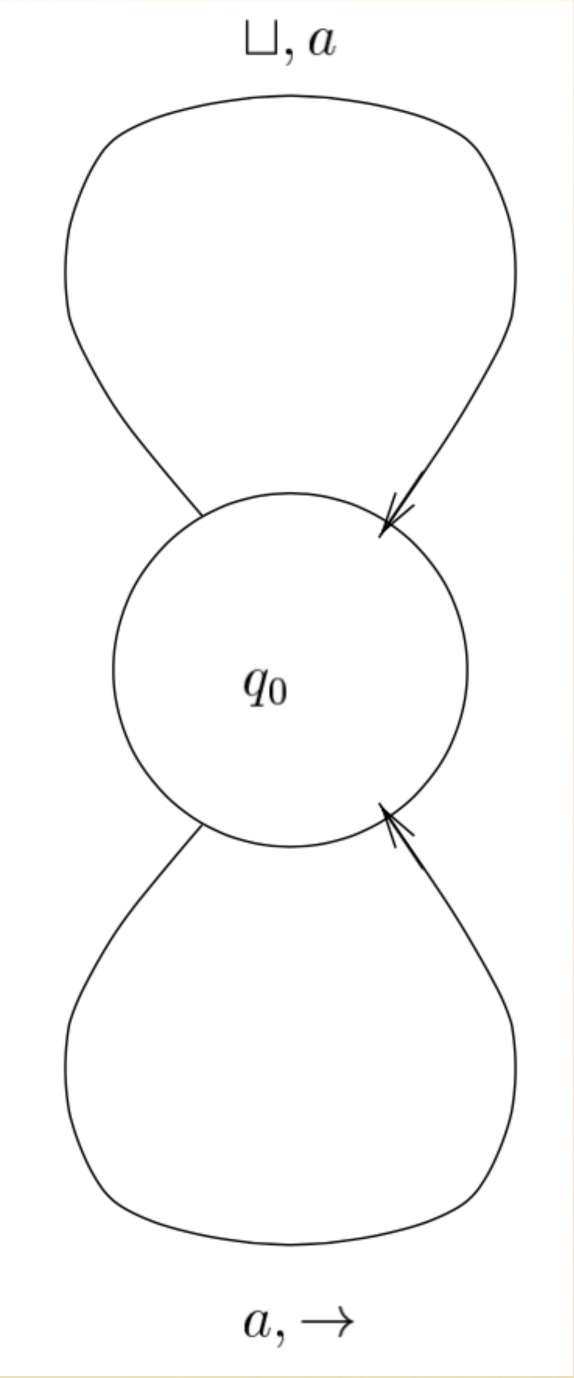


Example

Consider the TM $\mathcal{M}$ with

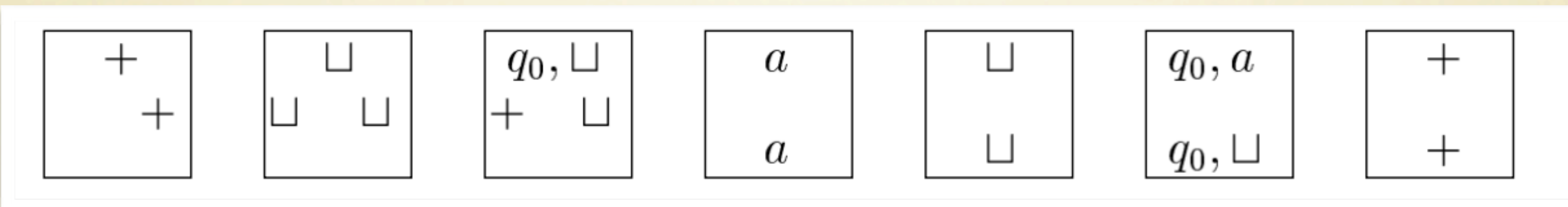
$\Sigma = \{a, \sqcup\}$, $Q = \{q_0, h\}$

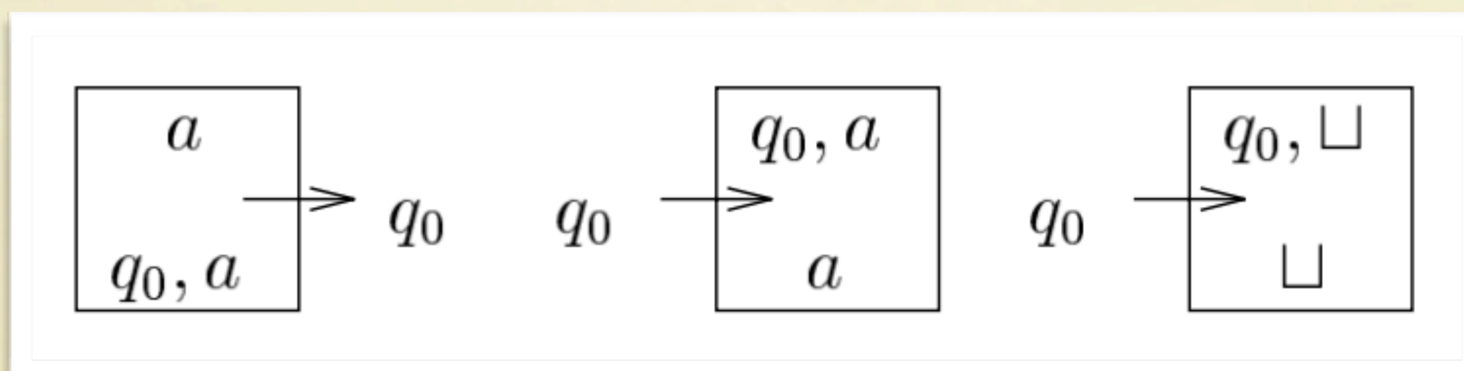
and δ as on the right.



Example

The resulting tiling system $\mathcal{T}_{\mathcal{M}}$ is defined as



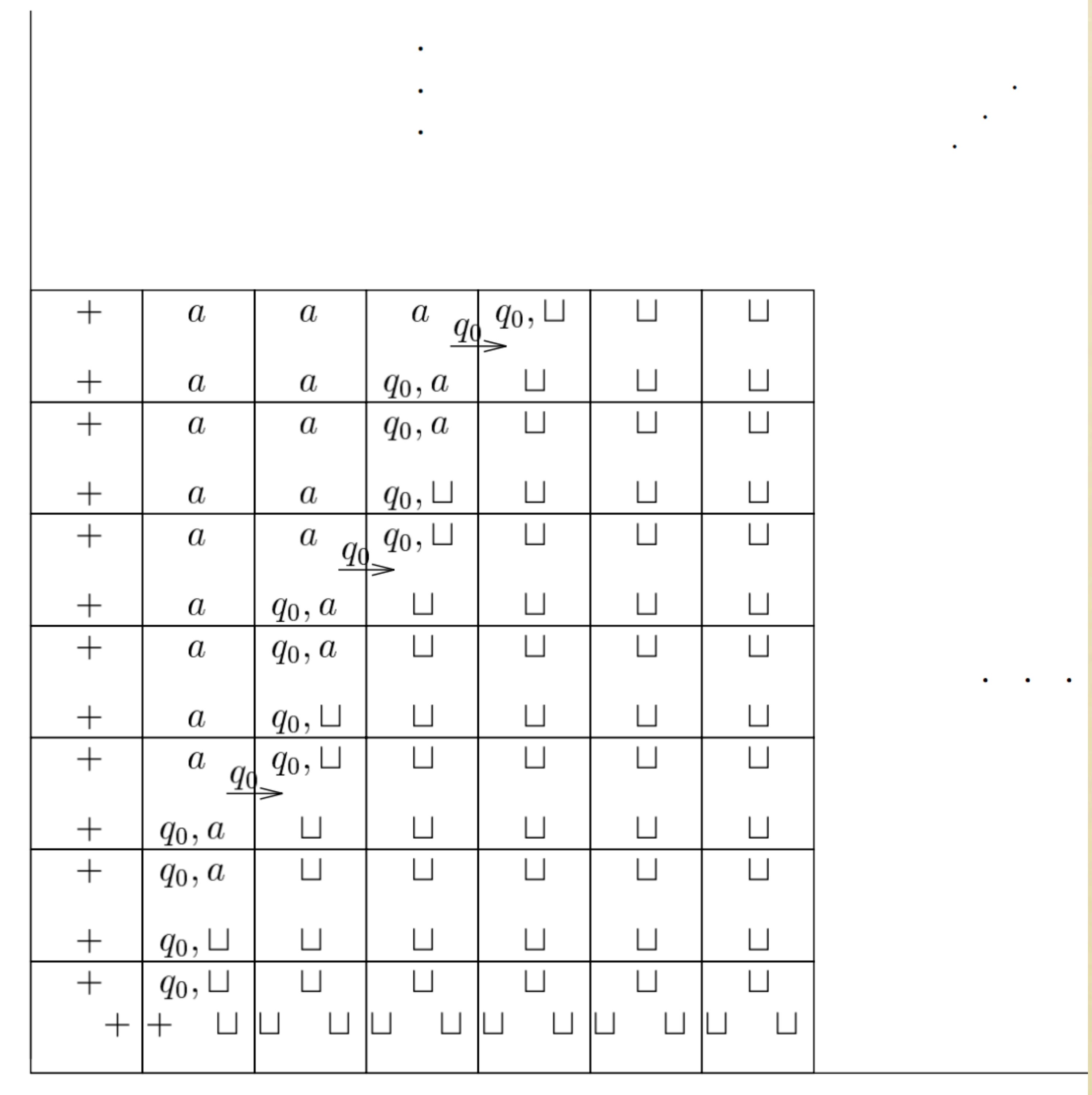


Example

$\mathcal{T}_{\mathcal{M}}$ has one tiling, whose initial part looks as on the right.

The tiling describes the computation of $\mathcal{M}$ on ε.

Observe that such a tiling exists $\mathcal{M}$ because does *not* halt.



The tiling problem is undecidable

We now have all the ingredients to return on our claim.

$$\text{code}(\mathcal{M}) \in \mathit{ETH} \quad \leftrightarrow \quad \text{no tiling exists for } \mathcal{T}_{\mathcal{M}}$$

That means,

$$\mathcal{M} \text{ halts on } \varepsilon \quad \leftrightarrow \quad \text{no tiling exists for } \mathcal{T}_{\mathcal{M}}$$

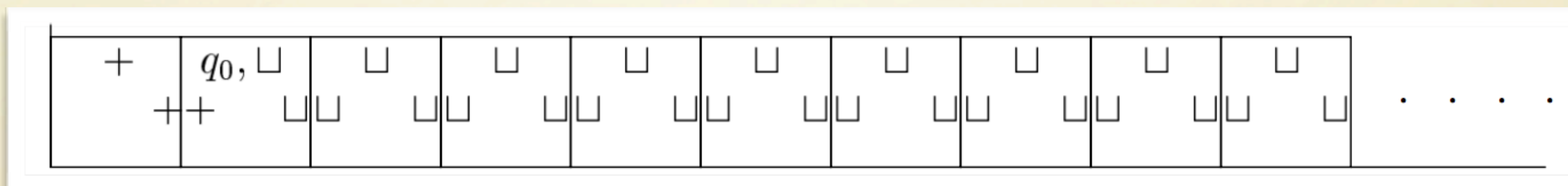
From this it follows that if the tiling problem was recognisable, then ETH^{-} would also be recognisable.

Therefore, the tiling problem is not recognisable.

The tiling problem is undecidable

So let's prove $\mathcal{M}$ halts on ε $\leftrightarrow$ no tiling exists for $\mathcal{T}_{\mathcal{M}}$

- First, suppose that $\mathcal{M}$ halts on ε, say in *n* steps.
- A tiling for $\mathcal{T}_{\mathcal{M}}$ is forced by definition to have a first row

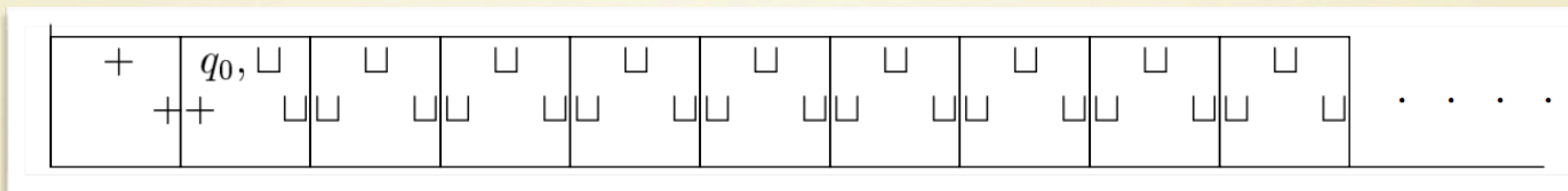


- In general, row *i* will describe the *i*th computation step of $\mathcal{M}$ on ε.
- This until row *n*. As $\mathcal{M}$ reaches an halting state h, by definition of $\mathcal{T}_{\mathcal{M}}$, row *n+1* cannot be constructed.
- So a tiling for $\mathcal{T}_{\mathcal{M}}$ does not exists.

The tiling problem is undecidable

So let's prove $\mathcal{M}$ halts on ε $\leftrightarrow$ no tiling exists for $\mathcal{T}_{\mathcal{M}}$

- Viceversa, suppose that $\mathcal{M}$ loops on ε.
- A tiling for $\mathcal{T}_{\mathcal{M}}$ is forced by definition to have a first row



- As before, the i^{th} computation step of $\mathcal{M}$ on ε will uniquely fix what is row i.

- So, if the computation never halts, a tiling exists.