On the problem of tiling the plane with a polyomino

Xavier Provençal

Laboratoire de Combinatoire et d’Informatique Mathématique, Université du Québec à Montréal,

12 mars, 2006
Outline

1. The tiling problem
2. Beauquier-Nivat characterization
3. A fast algorithm to detect exact polyominoes
Introduction to polyominoes

• Discrete plane: $\mathbb{Z}^2$
Introduction to polyominoes

• Discrete plane : \( \mathbb{Z}^2 \)

• **Definition**: A *polyomino* is a finite, 4-connected subset of the plane, without holes.
Introduction to polyominoes

- **Discrete plane**: $\mathbb{Z}^2$

- **Definition**: A *polyomino* is a finite, 4-connected subset of the plane, without holes.
Introduction to polyominoes

• Discrete plane: $\mathbb{Z}^2$

• **Definition**: A *polyomino* is a finite, 4-connected subset of the plane, without holes.
Introduction to polyominoes

• Discrete plane: $\mathbb{Z}^2$

• **Definition**: A *polyomino* is a finite, 4-connected subset of the plane, without holes.

• **Notation**: Let $p$ be a polyomino and $\vec{v}$ a vector of $\mathbb{Z}^2$, $p_{\vec{v}}$ will denote the image of $p$ by de translation $\vec{v}$. 
General statement of the tiling problem

Definition (Tiling)

A tiling $\mathcal{T}$ of a subset $D \subset \mathbb{Z}^2$ by a set of polyominoes $\mathcal{P}$ is a set of couples $(p, \overrightarrow{u}) \in \mathcal{P} \times \mathbb{Z}^2$ such that:

1. $D$ is the union of the polyominoes $p \overrightarrow{u}$.
2. For any distinct pair $(p, \overrightarrow{u}), (p', \overrightarrow{v}) \in \mathcal{T}$, $p \overrightarrow{u}$ and $p' \overrightarrow{v}$ are non-overlapping.
Definition (Tiling)

A tiling $\mathcal{T}$ of a subset $D \subset \mathbb{Z}^2$ by a set of polyominoes $\mathcal{P}$ is a set of couples $(p, \overrightarrow{u}) \in \mathcal{P} \times \mathbb{Z}^2$ such that:

- $D$ is the union of the polyominoes $p_{\overrightarrow{u}}$. 

Xavier Provençal

On the problem of tiling the plane with a polyomino
A tiling $\mathcal{T}$ of a subset $D \subset \mathbb{Z}^2$ by a set of polyominoes $\mathcal{P}$ is a set of couples $(p, \overrightarrow{u}) \in \mathcal{P} \times \mathbb{Z}^2$ such that:

- $D$ is the union of the polyominoes $p\overrightarrow{u}$.
- For any distinct pair $(p, \overrightarrow{u}), (p', \overrightarrow{v}) \in \mathcal{T}$, $p\overrightarrow{u}$ and $p'\overrightarrow{v}$ are non-overlapping.
General statement of the tiling problem

Definition (Tiling)

A tiling $\mathcal{T}$ of a subset $D \subset \mathbb{Z}^2$ by a set of polyominoes $\mathcal{P}$ is a set of couples $(p, \overrightarrow{u}) \in \mathcal{P} \times \mathbb{Z}^2$ such that:

- $D$ is the union of the polyominoes $p_{\overrightarrow{u}}$.
- For any distinct pair $(p, \overrightarrow{u}), (p', \overrightarrow{v}) \in \mathcal{T}$, $p_{\overrightarrow{u}}$ and $p'_{\overrightarrow{v}}$ are non-overlapping.

Definition (The Tiling Problem)

Given a set of polyominoes $\mathcal{P}$ and a subset $D \subset \mathbb{Z}^2$. Does $D$ admits a tiling by $\mathcal{P}$?
Example
Example
Finite case

Remark

The tiling problem with $D$ finite is in NP.
Finite case

Remark

The tiling problem with \( D \) finite is in NP.

Remark

The tiling problem with \( D \) finite and \( \mathcal{P} = \{ \begin{array}{c} \square \\ \square \end{array} \} \) is in P.
Finite case

Remark

The tiling problem with $D$ finite is in NP.

Remark

The tiling problem with $D$ finite and $P = \{\square, \blacksquare\}$ is in $P$.

Theorem (Garey, Johnson and Papadimitriou)

The tiling problem with $D$ finite and $P = \{\square, \blacksquare\}$ is NP-Complete.
We consider the case where \( D = \mathbb{Z}^2 \) and \( \mathcal{P} \) is finite.
Infinite case

We consider the case where $D = \mathbb{Z}^2$ and $\mathcal{P}$ is finite.

**Definition (Periodic Tiling)**

A tiling $\mathcal{T}$ is periodic if there exist two linearly independent vectors $\vec{u}$ and $\vec{v}$ such that $\mathcal{T}$ is not changed by the corresponding translations.
Infinite case

We consider the case where $D = \mathbb{Z}^2$ and $\mathcal{P}$ is finite.

**Definition (Periodic Tiling)**

*A tiling $\mathcal{T}$ is periodic if there exist two linearly independent vectors $\vec{u}$ and $\vec{v}$ such that $\mathcal{T}$ is not changed by the corresponding translations.*

**Definition (Half-Periodic Tiling)**

*A tiling $\mathcal{T}$ is half-periodic if there exists a vector $\vec{u}$ such that $\mathcal{T}$ is not changed by the corresponding translation.*
The tiling problem
Beauquier-Nivat characterization
A fast algorithm to detect exact polyominoes

Example

Periodic tiling
Half-periodic tiling
Remark

*If there is an half-periodic tiling of the plane by \( P \), then there is also a periodic one.*
Remark

If there is an half-periodic tiling of the plane by $P$, then there is also a periodic one.
Remark

*If there is an half-periodic tiling of the plane by \( \mathcal{P} \), then there is also a periodic one.*
Remark

*If there is an half-periodic tiling of the plane by $\mathcal{P}$, then there is also a periodic one.*
Half-Periodic implies periodic

Remark

If there is an half-periodic tiling of the plane by $\mathcal{P}$, then there is also a periodic one.
Remark

If there is an half-periodic tiling of the plane by \( P \), then there is also a periodic one.
Theorem (Berger, 1966)

The tiling problem with $\mathcal{P}$ finite and $D = \mathbb{Z}^2$ is undecidable.
Nonperiodic tilings

Theorem (Berger, 1966)

The tiling problem with $\mathcal{P}$ finite and $D = \mathbb{Z}^2$ is undecidable.

Corollary

There are some finite sets $\mathcal{P}$ such that tilings of the plane by $\mathcal{P}$ do exist and are all nonperiodic.
Tilings with one polyomino

Definition

A polyomino \( p \) is exact if the set \( P = \{p\} \) tiles the plane.
Tilings with one polyomino

Definition

A polyomino $p$ is exact if the set $\mathcal{P} = \{p\}$ tiles the plane.

Definition

A tiling of the plane $\mathcal{T}$ by an exact polyomino $p$ is regular if there exist two vectors $\vec{u}$ and $\vec{v}$ such that

$$\mathcal{T} = \{(p, i\vec{u} + j\vec{v}) | i, j \in \mathbb{Z}^2\}$$
Examples

- Half-periodic tiling
- Periodic tiling
- Regular tiling
Tilings with one polyomino

**Theorem (Wijshoff and Van Leeuven, 1984)**

*If a polyomino $p$ is exact, then there exists a regular tiling of the plane by $p$.*
Theorem (Wijshoff and Van Leeuven, 1984)

If a polyomino \( p \) is exact, then there exists a regular tiling of the plane by \( p \).

Corollary

The tiling problem with \(|P| = 1\) and \( D = \mathbb{Z}^2 \) is decidable in polynomial time.
Example
Example
Example

The tiling problem
Beauquier-Nivat characterization
A fast algorithm to detect exact polyominoes

Definitions
General statement
Finite case
Infinite case

Xavier Provençal
On the problem of tiling the plane with a polyomino
Example
Example
Example
Example
Example
Outline

1. The tiling problem
2. Beauquier-Nivat characterization
3. A fast algorithm to detect exact polyominoes
Coding the boundary of a polyomino

\[ \Sigma = \{ a, \bar{a}, b, \bar{b} \} \]
Coding the boundary of a polyomino

\[ \Sigma = \{ a, \bar{a}, b, \bar{b} \} \]

Notation: \[ w \equiv w' \] notes that \( w \) and \( w' \) are conjugate.

There exist \( u, v \in \Sigma^* \) such that:
\[ w = uv \] and \[ w' = vu \].

\[ w = \]

\[ X \]
Coding the boundary of a polyomino

\[ \Sigma = \{a, \bar{a}, b, \bar{b}\} \]

\[ a \rightarrow b \uparrow \]
\[ \bar{a} \leftarrow \bar{b} \downarrow \]

\[ w = a \]
Coding the boundary of a polyomino

\[ \Sigma = \{ a, \bar{a}, b, \bar{b} \} \]

\[ w = a \ a \ a \]

Notation:
\[ w \equiv w' \] notes that \( w \) and \( w' \) are conjugate.

There exist \( u, v \in \Sigma^* \) such that:
\[ w = uv \quad \text{and} \quad w' = vu. \]
Coding the boundary of a polyomino

\[ \Sigma = \{ a, \bar{a}, b, \bar{b} \} \]

\[ w = a \ a \ a \]

Notation:
\[ w \equiv w' \] notes that \( w \) and \( w' \) are conjugate.

There exist \( u, v \in \Sigma^* \) such that:
\[ w = uv \] and \[ w' = vu \].
Codings the boundary of a polyomino

\[ \Sigma = \{ a, \bar{a}, b, \bar{b} \} \]

\[ a \rightarrow b \]
\[ \bar{a} \leftarrow \bar{b} \]

\[ w = a \ a \ a \ a \]
Coding the boundary of a polyomino

\[ \Sigma = \{ a, \bar{a}, b, \bar{b} \} \]

Notation: \( w \equiv w' \) notes that \( w \) and \( w' \) are conjugate.

There exist \( u, v \in \Sigma^* \) such that:

\[ w = uv \quad \text{and} \quad w' = vu. \]

\[ w = a a a a b \]
Coding the boundary of a polyomino

\[ \Sigma = \{ a, \overline{a}, b, \overline{b} \} \]

\[ w = a \ a \ a \ a \ b \ \overline{a} \]

Notation: \( w \equiv w' \) notes that \( w \) and \( w' \) are conjugate.

There exist \( u, v \in \Sigma^* \) such that:

\[ w = uv \quad \text{and} \quad w' = vu. \]
Coding the boundary of a polyomino

\[ \Sigma = \{ a, \bar{a}, b, \bar{b} \} \]

\[ w = a \ a \ a \ a \ b \ \bar{a} \ b \]
Coding the boundary of a polyomino

\[ \Sigma = \{ a, \bar{a}, b, \bar{b} \} \]

\[ w = a \ a \ a \ a \ b \ \bar{a} \ b \ \bar{a} \]
Coding the boundary of a polyomino

\[ \Sigma = \{ a, \bar{a}, b, \bar{b} \} \]

\[ w = a \ a \ a \ a \ b \ \bar{a} \ b \ \bar{a} \ \bar{a} \]
Coded the boundary of a polyomino

\[ \Sigma = \{ a, \overline{a}, b, \overline{b} \} \]

\[ w = a a a a b \overline{a} b \overline{a} \overline{a} \overline{b} \]
Coding the boundary of a polyomino

\[ \Sigma = \{ a, \bar{a}, b, \bar{b} \} \]

\[ a \rightarrow b \uparrow \]
\[ \bar{a} \leftarrow \bar{b} \downarrow \]

\[ w = a \ a \ a \ a \ b \ \bar{a} \ b \ \bar{a} \ \bar{a} \ b \ \bar{a} \]
The tiling problem
Beauquier-Nivat characterization
A fast algorithm to detect exact polyominoes

Polyominoes and words
Definitions
Surroundings and tilings
Surroundings and the factorization

Coding the boundary of a polyomino

\[ \Sigma = \{a, \bar{a}, b, \bar{b}\} \]

\[ w = a \ a \ a \ a \ b \ \bar{a} \ b \ \bar{a} \ \bar{a} \ \bar{b} \ \bar{a} \ \bar{b} \]

Notation :
\[ w \equiv w' \] notes that \( w \) and \( w' \) are conjugate.

There exist \( u, v \in \Sigma^* \) such that :
\[ w = uv \quad \text{and} \quad w' = vu. \]
coding the boundary of a polyomino

\[ \Sigma = \{ a, \overline{a}, b, \overline{b} \} \]

**Notation:** \( w \equiv w' \) notes that \( w \) and \( w' \) are conjugate.

There exist \( u, v \in \Sigma^* \) such that:
\[ w = uv \text{ and } w' = vu. \]
The tiling problem
Beauquier-Nivat characterization
A fast algorithm to detect exact polyominoes
Polyominoes and words
Definitions
Surroundings and tilings
Surroundings and the factorization

Coding the boundary of a polyomino

\( \Sigma = \{ a, \bar{a}, b, \bar{b} \} \)

Notation:
\( w \equiv w' \) notes that \( w \) and \( w' \) are conjugate.

There exist \( u, v \in \Sigma^* \) such that:
\( w = uv \) and \( w' = vu \).
Definition

Let $\hat{\cdot}$ be the involutive antimorphism defined as $\hat{\cdot} = \bar{\cdot} \circ \bar{\cdot}$. 

Theorem (Beauquier and Nivat, 1991)

A polyomino $p$ is exact if and only if its boundary word $w \equiv XYZ \hat{X} \hat{Y} \hat{Z}$ for some $X, Y, Z \in \Sigma^*$. 

On the problem of tiling the plane with a polyomino
Definition

Let $\hat{\cdot}$ be the involutive antimorphism defined as $\hat{\cdot} = - \circ \bar{\cdot}$.

Let $u, v, w \in \Sigma^* = \{a, \bar{a}, b, \bar{b}\}^*$ be such that $w = uv$, $\hat{w} = \hat{uv} = \hat{v}\hat{u}$ and $w = \hat{\hat{w}}$. 
Definition

Let \( \hat{} \) be the involutive antimorphism defined as \( \hat{} = - \circ \sim \).

Let \( u, v, w \in \Sigma^* = \{a, \overline{a}, b, \overline{b}\}^* \) be such that \( w = uv \), \( \hat{w} = \hat{u}v = \hat{v}u \) and \( w = \hat{w} \).

\[
u = a \ a \ b \ a \ \overline{b} \ a \ b
\]
Characterization

Definition

Let \( \hat{\cdot} \) be the involutive antimorphism defined as \( \hat{\cdot} = \overline{\circ \circ} \).

Let \( u, v, w \in \Sigma^* = \{ a, \overline{a}, b, \overline{b} \}^* \) be such that \( w = uv \), \( \hat{w} = \hat{uv} = \hat{v} \hat{u} \) and \( w = \hat{w} \).

\[
 u = a \ a \ b \ a \ \overline{b} \ a \ b
\]
Characterization

**Definition**

Let $\hat{\sim}$ be the involutive antimorphism defined as $\hat{\sim} = - \circ \sim$.

Let $u, v, w \in \Sigma^* = \{a, \bar{a}, b, \bar{b}\}^*$ be such that $w = uv$, $\hat{w} = \hat{uv} = \hat{v} \hat{u}$ and $w = \hat{w}$.

\[
u = a \ a \ b \ a \ \bar{b} \ a \ b\]
Characterization

Definition

Let $\hat{\cdot}$ be the involutive antimorphism defined as $\hat{\cdot} = \overline{\cdot} \circ \overline{\cdot}$.

Let $u, v, w \in \Sigma^* = \{a, \overline{a}, b, \overline{b}\}^*$ be such that $w = uv$, $\hat{w} = \hat{u}v = \hat{v}u$ and $w = \hat{\hat{w}}$.

\[
u = a \ a \ b \ a \ \overline{b} \ a \ b \ \]
Characterization

Definition

Let \( \hat{\ } \) be the involutive antimorphism defined as \( \hat{\ } = - \circ \hat{\ } \).

Let \( u, \nu, w \in \Sigma^* = \{a, \bar{a}, b, \bar{b}\}^* \) be such that \( w = uv \), \( \hat{w} = \hat{uv} = \hat{\nu}u \) and \( w = \hat{\hat{w}} \).

\[
\begin{align*}
\hat{u} &= a \ a \ b \ a \ \bar{b} \ a \ b
\end{align*}
\]
Characterization

Definition

Let \( \hat{\ } \) be the involutive antimorphism defined as \( \hat{\ } = - \circ \tilde{\ } \).

Let \( u, v, w \in \Sigma^* = \{a, \overline{a}, b, \overline{b}\}^* \) be such that \( w = uv \), \( \hat{w} = \hat{uv} = \hat{v}u \) and \( w = \hat{w} \).

\[
\begin{align*}
    u &= a \ a \ b \ a \ \overline{b} \ a \ b \\
    \hat{u} &= a \ \overline{a} \ \overline{b} \ \overline{b} \ \overline{a} \ a
\end{align*}
\]
Definition

Let \( \hat{\ } \) be the involutive antimorphism defined as \( \hat{\ } = \overline{\circ \overline{\ }}. \)

Let \( u, v, w \in \Sigma^* = \{ a, \overline{a}, b, \overline{b} \}^* \) be such that \( w = uv, \hat{w} = \hat{uv} = \hat{v}\hat{u} \) and \( w = \hat{\hat{w}}. \)

\[
\begin{align*}
  u &= a \ a \ b \ a \ \overline{b} \ a \ b \\
\end{align*}
\]
Definition

Let \( \hat{\cdot} \) be the involutive antimorphism defined as \( \hat{\cdot} = \bar{\cdot} \circ \bar{\cdot} \).

Let \( u, v, w \in \Sigma^* = \{a, \bar{a}, b, \bar{b}\}^* \) be such that \( w = uv \), \( \hat{w} = \hat{uv} = \hat{v} \hat{u} \) and \( w = \hat{\hat{w}} \).

\[
\begin{align*}
\hat{u} &= a a b a \bar{b} a b
\end{align*}
\]
The tiling problem

Beauquier-Nivat characterization

A fast algorithm to detect exact polyominoes

Polyominoes and words

Definitions

Surroundings and tilings

Surroundings and the factorization

Characterization

Definition

Let \( \hat{\cdot} \) be the involutive antimorphism defined as \( \hat{\cdot} \circ \cdot \).

Let \( u, v, w \in \Sigma^* = \{a, \bar{a}, b, \bar{b}\}^* \) be such that \( w = uv \), \( \hat{w} = \hat{uv} = \hat{v}u \) and \( w = \hat{w} \).

\[
\begin{align*}
  u & = a \ a \ b \ a \ \bar{b} \ a \ b \\
\end{align*}
\]
Let $\hat{\ast}$ be the involutive antimorphism defined as $\hat{\ast} = \ast \circ \hat{\ast}$.

Let $u, v, w \in \Sigma^* = \{a, \bar{a}, b, \bar{b}\}^*$ be such that $w = uv$, $\hat{w} = \hat{u}v = \hat{v}u$ and $w = \hat{\hat{w}}$.

\[ u = a \ a \ b \ a \ \bar{b} \ a \ b \]

\[ \hat{u} = \bar{b} \ \bar{a} \ b \ \bar{a} \ \bar{b} \ \bar{a} \ \bar{a} \]
**Characterization**

**Definition**

Let \( \hat{\cdot} \) be the involutive antimorphism defined as \( \hat{\cdot} = \overline{\cdot} \circ \overline{\cdot} \).

Let \( u, v, w \in \Sigma^* = \{a, \overline{a}, b, \overline{b}\}^* \) be such that \( w = uv \), \( \hat{w} = \hat{uv} = \hat{v}\hat{u} \) and \( w = \hat{\hat{w}} \).

\[
\begin{align*}
  u &= a\ a\ b\ a\ \overline{b}\ a\ b \\
  \hat{u} &= \overline{b}\ \overline{a}\ b\ \overline{a}\ \overline{b}\ \overline{a}\ \overline{a}
\end{align*}
\]
**Characterization**

**Definition**

Let $\hat{\sim}$ be the involutive antimorphism defined as $\hat{\sim} = \overline{\circ \sim}$.

Let $u, v, w \in \Sigma^* = \{a, \overline{a}, b, \overline{b}\}^*$ be such that $w = uv$, $\hat{w} = \hat{uv} = \hat{v}\hat{u}$ and $w = \hat{\hat{w}}$.

$$u = a \ a \ b \ a \ \overline{b} \ a \ b$$

$$\hat{u} = \overline{b} \ \overline{a} \ b \ \overline{a} \ \overline{b} \ \overline{a} \ a$$
The tiling problem
Beauquier-Nivat characterization
A fast algorithm to detect exact polyominoes

Characterization

Definition

Let \( \hat{\ } \) be the involutive antimorphism defined as \( \hat{\ } = \overline{\circ \circ \circ} \).

Let \( u, v, w \in \Sigma^* = \{a, \overline{a}, b, \overline{b}\}^* \) be such that \( w = uv, \hat{w} = \hat{u}v = \hat{v}u \) and \( w = \hat{w} \).

\[
\begin{align*}
  u &= a \ a \ b \ a \ \overline{b} \ a \ b \\
  \hat{u} &= \overline{b} \ \overline{a} \ b \ \overline{a} \ b \ \overline{a} \ \overline{a} \\
\end{align*}
\]

Theorem (Beauquier and Nivat, 1991)

A polyomino \( p \) is exact if and only if its boundary word \( w \equiv XYZ \hat{X} \hat{Y} \hat{Z} \) for some \( X, Y, Z \in \Sigma^* \).
Characterization

**Definition**

Let \( \hat{\cdot} \) be the involutive antimorphism defined as \( \hat{\cdot} = \hat{-} \circ \hat{\cdot} \).

Let \( u, \nu, w \in \Sigma^* = \{a, \bar{a}, b, \bar{b}\}^* \) be such that \( w = uv \), \( \hat{w} = \hat{uv} = \hat{\nu} \hat{u} \) and \( w = \hat{\hat{w}} \).

\[
\begin{align*}
    u &= a \ a \ b \ a \ \bar{b} \ a \ b \\
    \hat{u} &= \bar{b} \ \bar{a} \ b \ \bar{a} \ \bar{b} \ a \ \bar{a}
\end{align*}
\]
Definition

Let \( \hat{\cdot} \) be the involutive antimorphism defined as \( \hat{\cdot} = \overline{\cdot} \circ \overline{\cdot} \).

Let \( u, v, w \in \Sigma^* = \{a, \overline{a}, b, \overline{b}\}^* \) be such that \( w = uv \), \( \hat{w} = \hat{u}v = \hat{v}u \) and \( \hat{w} = \hat{u} \).

\[
\begin{align*}
\hat{u} &= \overline{b} \overline{a} b \overline{a} \overline{b} \overline{a} \overline{a} \\
\hat{u} &= \overline{b} \overline{a} b \overline{a} \overline{b} \overline{a} \overline{a}
\end{align*}
\]
Definition

Let $\hat{\cdot}$ be the involutive antimorphism defined as $\hat{\cdot} = \neg \circ \neg$.

Let $u, v, w \in \Sigma^* = \{a, \bar{a}, b, \bar{b}\}^*$ be such that $w = uv$, $\hat{w} = \hat{u}v = \hat{v}u$ and $w = \hat{\hat{w}}$.

$$u = a \ a \ b \ \bar{a} \ b \ a \ b$$

$$\hat{u} = \bar{b} \ \bar{a} \ b \ \bar{a} \ b \ \bar{a} \ \bar{a}$$
The tiling problem
Beauquier-Nivat characterization
A fast algorithm to detect exact polyominoes

Polyominoes and words
Definitions
Surroundings and tilings
Surroundings and the factorization

Characterization

Definition

Let $\hat{\cdot}$ be the involutive antimorphism defined as $\hat{\cdot} = -\circ \hat{\cdot}$.

Let $u, v, w \in \Sigma^* = \{a, \bar{a}, b, \bar{b}\}^*$ be such that $w = uv$, $\hat{w} = \hat{u}v = \hat{v}u$ and $w = \hat{\hat{w}}$.

\[ u = a \ a \ b \ a \ \bar{b} \ a \ b \]
\[ \hat{u} = \bar{b} \ \bar{a} \ b \ \bar{a} \ \bar{b} \ \bar{a} \ \bar{a} \]

Theorem (Beauquier and Nivat, 1991)

A polyomino $p$ is exact if and only if its boundary word $w \equiv XYZ \hat{X} \hat{Y} \hat{Z}$ for some $X, Y, Z \in \Sigma^*$.
Let $\hat{\cdot}$ be the involutive antimorphism defined as $\hat{\cdot} = \bar{\cdot} \circ \cdot$.

Let $u, v, w \in \Sigma^* = \{a, \bar{a}, b, \bar{b}\}^*$ be such that $w = uv$, $\hat{w} = \hat{uv} = \hat{v} \hat{u}$ and $w = \hat{\hat{w}}$.

$$u = a \ a \ b \ a \ \bar{b} \ a \ b$$

$$\hat{u} = b \ \bar{a} \ b \ \bar{a} \ \bar{b} \ \bar{a} \ \bar{a}$$
Characterization

Definition

Let $\hat{\cdot}$ be the involutive antimorphism defined as $\hat{\cdot} = \circlearrowright \circ \circlearrowright$.

Let $u, v, w \in \Sigma^* = \{a, \overline{a}, b, \overline{b}\}^*$ be such that $w = uv$, $\hat{w} = \hat{uv} = \hat{v}\hat{u}$ and $w = \hat{w}$.

\[
\begin{align*}
  u &= a a b a \overline{b} a b \\
  \hat{u} &= \overline{b} \overline{a} b \overline{a} b \overline{a} \overline{a}
\end{align*}
\]

Theorem (Beauquier and Nivat, 1991)

A polyomino $p$ is exact if and only its boundary word $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$ for some $X, Y, Z \in \Sigma^*$.  

Xavier Provençal
On the problem of tiling the plane with a polyomino
Neighbouring

Definition

Two polyominoes $p$ and $q$ are simply neighbouring if

- They are adjacent.
- They don’t overlap.
- They don’t form a hole.
**Neighbouring**

**Definition**

Two polyominoes $p$ and $q$ are simply neighbouring if

- They are adjacent.
- They don’t overlap.
- They don’t form a hole.
**Definition**

*Three polyominoes p, q and r form a triad if*

- They are two by two simply neighbouring.
- They don’t form a hole.
Three polyominoes $p$, $q$ and $r$ form a triad if

1. They are two by two simply neighbouring.
2. They don’t form a hole.
Definition

A surrounding of the polyomino $p$ is an ordered sequence of translated copies $(p_0, p_1, \ldots, p_{k-1})$ such that for every $i$ from 0 to $k-1$, the polyominoes $p$, $p_i$ and $p_{i+1}$ form a triad.
Surrounding

Definition

A surrounding of the polyomino $p$ is an ordered sequence of translated copies $(p_0, p_1, \ldots, p_{k-1})$ such that for every $i$ from 0 to $k-1$, the polyominoes $p$, $p_i$, and $p_{i+1}$ form a triad.
A surrounding of the polyomino $p$ is an ordered sequence of translated copies $(p_0, p_1, \ldots, p_{k-1})$ such that for every $i$ from 0 to $k - 1$, the polyominoes $p$, $p_i$, and $p_{i+1}$ form a triad.
A surrounding of the polyomino $p$ is an ordered sequence of translated copies $(p_0, p_1, \ldots, p_{k-1})$ such that for every $i$ from 0 to $k-1$, the polyominoes $p$, $p_i$, and $p_{i+1}$ form a triad.
A surrounding of the polyomino $p$ is an ordered sequence of translated copies $(p_0, p_1, \ldots, p_{k-1})$ such that for every $i$ from 0 to $k-1$, the polyominoes $p$, $p_i$, and $p_{i+1}$ form a triad.
Surrounding

Definition

A surrounding of the polyomino $p$ is an ordered sequence of translated copies $(p_0, p_1, \ldots, p_{k-1})$ such that for every $i$ from 0 to $k - 1$, the polyominoes $p$, $p_i$ and $p_{i+1}$ form a triad.
Surrounding

Definition

A surrounding of the polyomino \( p \) is an ordered sequence of translated copies \( (p_0, p_1, \ldots, p_{k-1}) \) such that for every \( i \) from 0 to \( k - 1 \), the polyominoes \( p \), \( p_i \) and \( p_{i+1} \) form a triad.
Surroundings and tilings

Proposition

A polyomino $p$ is exact if and only if it admits a surrounding.
Surroundings and tilings

**Proposition**

*A polyomino $p$ is exact if and only if it admits a surrounding.*
Proposition

A polyomino $p$ is exact if and only if it admits a surrounding.
Surroundings and tilings

**Proposition**

*A polyomino \( p \) is exact if and only if it admits a surrounding.*
Example

The tiling problem
Beauquier-Nivat characterization
A fast algorithm to detect exact polyominoes

Polyominoes and words
Definitions
Surroundings and tilings
Surroundings and the factorization

Example

\[ p \]

On the problem of tiling the plane with a polyomino
Example

The tiling problem
Beauquier-Nivat characterization
A fast algorithm to detect exact polyominoes
Polyominoes and words
Definitions
Surroundings and tilings
Surroundings and the factorization

Example
Example
Example
Example
Example
Example

\[ p \rightarrow u_0 \]
\[ p \rightarrow u_1 \]
\[ p \rightarrow u_2 \]
\[ p \rightarrow u_3 \]
\[ p \rightarrow u_4 \]
\[ p \rightarrow u_5 \]
\[ p \rightarrow u_6 \]
\[ p \rightarrow u_7 \]
\[ p \rightarrow u_8 \]
\[ p \rightarrow u_9 \]
\[ p \rightarrow u_{10} \]
\[ p \rightarrow u_{11} \]
\[ p \rightarrow u_{12} \]
\[ p \rightarrow u_{13} \]
\[ p \rightarrow u_{14} \]
\[ p \rightarrow u_{15} \]
\[ p \rightarrow u_{16} \]
\[ p \rightarrow u_{17} \]

\[ S S \]
\[ S S \ w_1 \]
\[ S S S S \ w_2 \]

\[ p_2 \rightarrow u_1 \]
\[ p_2 \rightarrow u_2 \]
\[ p_2 \rightarrow u_3 \]
\[ p_2 \rightarrow u_4 \]
\[ p_2 \rightarrow u_5 \]
\[ p_2 \rightarrow u_6 \]
\[ p_2 \rightarrow u_7 \]
\[ p_2 \rightarrow u_8 \]
\[ p_2 \rightarrow u_9 \]
\[ p_2 \rightarrow u_{10} \]
\[ p_2 \rightarrow u_{11} \]
\[ p_2 \rightarrow u_{12} \]
\[ p_2 \rightarrow u_{13} \]
\[ p_2 \rightarrow u_{14} \]
\[ p_2 \rightarrow u_{15} \]
\[ p_2 \rightarrow u_{16} \]
\[ p_2 \rightarrow u_{17} \]

\[ p \rightarrow u_1 \]
\[ p \rightarrow u_2 \]
\[ p \rightarrow u_3 \]
\[ p \rightarrow u_4 \]
\[ p \rightarrow u_5 \]
\[ p \rightarrow u_6 \]
\[ p \rightarrow u_7 \]
\[ p \rightarrow u_8 \]
\[ p \rightarrow u_9 \]
\[ p \rightarrow u_{10} \]
\[ p \rightarrow u_{11} \]
\[ p \rightarrow u_{12} \]
\[ p \rightarrow u_{13} \]
\[ p \rightarrow u_{14} \]
\[ p \rightarrow u_{15} \]
\[ p \rightarrow u_{16} \]
\[ p \rightarrow u_{17} \]
Example
Example
Example
Example
Example
Example
Proposition

A polyomino $p$ admits a surrounding if and only if its boundary word $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$ for some $X, Y, Z \in \Sigma^*$. 
Proposition

A polyomino \( p \) admits a surrounding if and only if its boundary word \( w \equiv XYZ\hat{X}\hat{Y}\hat{Z} \) for some \( X, Y, Z \in \Sigma^* \).
Surroundings and the factorization

Proposition

A polyomino $p$ admits a surrounding if and only if its boundary word $w \equiv XYZ \hat{X} \hat{Y} \hat{Z}$ for some $X, Y, Z \in \Sigma^*$. 
Proposition

A polyomino $p$ admits a surrounding if and only if its boundary word $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$ for some $X, Y, Z \in \Sigma^*$. 
Proposition

A polyomino $p$ admits a surrounding if and only if its boundary word $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$ for some $X, Y, Z \in \Sigma^*$. 
Proposition

A polyomino $p$ admits a surrounding if and only if its boundary word $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$ for some $X, Y, Z \in \Sigma^*$. 
Proposition

A polyomino $p$ admits a surrounding if and only if its boundary word $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$ for some $X, Y, Z \in \Sigma^*$. 
Surroundings and the factorization

Proposition

A polyomino \( p \) admits a surrounding if and only if its boundary word \( w \equiv XYZ\hat{X}\hat{Y}\hat{Z} \) for some \( X, Y, Z \in \Sigma^* \).
Surroundings and the factorization

Proposition

A polyomino $p$ admits a surrounding if and only if its boundary word $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$ for some $X, Y, Z \in \Sigma^*$. 
Surroundings and the factorization

**Proposition**

A polyomino $p$ admits a surrounding if and only if its boundary word $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$ for some $X, Y, Z \in \Sigma^*$.

Let $\vec{u}, \vec{v} \in \mathbb{Z}^2$ be such that $T = \{(p, i\vec{u} + j\vec{v}) | i, j \in \mathbb{Z}^2\}$ forms a regular tiling and that $p, p\overrightarrow{u}, p\overrightarrow{v}$ form a triad, then $(p\overrightarrow{u}, p\overrightarrow{v}, p\overrightarrow{-u}, p\overrightarrow{-v}, p\overrightarrow{-u}, p\overrightarrow{-v})$ form a surrounding of $p$. 
Proposition

A polyomino $p$ admits a surrounding if and only if its boundary word $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$ for some $X, Y, Z \in \Sigma^*$. 
Proposition

A polyomino $p$ admits a surrounding if and only if its boundary word $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$ for some $X, Y, Z \in \Sigma^*$. 
Proposition

A polyomino $p$ admits a surrounding if and only if its boundary word $w \equiv XYZ \hat{X} \hat{Y} \hat{Z}$ for some $X, Y, Z \in \Sigma^*$. 

\[ p \quad \Rightarrow \quad \hat{X} \quad \Rightarrow \quad \hat{Y} \quad \Rightarrow \quad \hat{Z} \quad \Rightarrow \quad X \quad \Rightarrow \quad Y \quad \Rightarrow \quad Z \quad \Rightarrow \quad p \]
Proposition

A polyomino \( p \) admits a surrounding if and only if its boundary word \( w \equiv XYZ \hat{X} \hat{Y} \hat{Z} \) for some \( X, Y, Z \in \Sigma^* \).
Proposition

A polyomino $p$ admits a surrounding if and only if its boundary word $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$ for some $X, Y, Z \in \Sigma^*$. 
**Proposition**

A polyomino $p$ admits a surrounding if and only if its boundary word $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$ for some $X, Y, Z \in \Sigma^*$. 
Proposition

A polyomino $p$ admits a surrounding if and only if its boundary word $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$ for some $X, Y, Z \in \Sigma^*$. 

![Diagram of polyominoes and surroundings](image-url)
Surroundings and the factorization

Proposition

A polyomino $p$ admits a surrounding if and only if its boundary word $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$ for some $X, Y, Z \in \Sigma^*$. 
Pseudo-square and pseudo-hexagons

Definition

An exact polyomino $p$ with Beauquier-Nivat factorization $X Y Z \hat{X} \hat{Y} \hat{Z}$ is called a pseudo-square if one of the factors $X, Y, Z$ is the empty word. It is called a pseudo-hexagon otherwise.
Pseudo-square and pseudo-hexagons

Definition

An exact polyomino \( p \) with Beauquier-Nivat factorization \( XYZ\hat{X}\hat{Y}\hat{Z} \) is called a pseudo-square if one of the factors \( X, Y, Z \) is the empty word. It is called a pseudo-hexagon otherwise.

Pseudo-hexagon
\[
\hat{w} \equiv XYZ\hat{X}\hat{Y}\hat{Z}.
\]

Pseudo-square
\[
\hat{w} \equiv XY\hat{X}\hat{Y}.
\]
Let $n$ be the length of the word coding the boundary of a polyomino $p$.

Remark

The Beauquier-Nivat characterization provides a naive algorithm to determine if $p$ is exact in $O(n^4)$. 
Let $n$ be the length of the word coding the boundary of a polyomino $p$.

**Remark**

The Beauquier-Nivat characterization provides a naive algorithm to determine if $p$ is exact in $O(n^4)$.

**Remark**

This problem admits $\Omega(n)$ as a lower bound.
Let $n$ be the length of the word coding the boundary of a polyomino $p$.

**Remark**

*The Beauquier-Nivat characterization provides a naive algorithm to determine if $p$ is exact in $O(n^4)$.*

**Remark**

*This problem admits $\Omega(n)$ as a lower bound.*

**Theorem (Gambini and Vuillon, 2003)**

*There is an algorithm to test if a polyomino satisfies the Beauquier-Nivat characterization in $O(n^2)$.*
Outline

1. The tiling problem
2. Beauquier-Nivat characterization
3. A fast algorithm to detect exact polyominoes
Admissible factors

Definition

Let $A$ be a factor of the word $w$ coding a polyomino $p$. $A$ is admissible if

- $w \equiv Ax\hat{A}y$, for $x, y$ such that $|x| = |y|$.
- $A$ is maximal, that is, $\text{first}(x) \neq \text{last}(x)$ and $\text{first}(y) \neq \text{last}(y)$. 


Admissible factors

Proposition

Let \( \mathcal{A} \) be the set of all admissible factors overlapping a position \( \alpha \) in \( w \) and \( \hat{\mathcal{A}} \) be the set of their respective homologous factors. Then, there is at least one position in \( w \) that is not covered by any element of \( \mathcal{A} \cup \hat{\mathcal{A}} \).
Proposition

Let $\mathcal{A}$ be the set of all admissible factors overlapping a position $\alpha$ in $w$ and $\hat{\mathcal{A}}$ be the set of their respective homologous factors. Then, there is at least one position in $w$ that is not covered by any element of $\mathcal{A} \cup \hat{\mathcal{A}}$. 

$w \equiv \alpha$
Admissible factors

**Proposition**

Let $\mathcal{A}$ be the set of all admissible factors overlapping a position $\alpha$ in $w$ and $\hat{\mathcal{A}}$ be the set of their respective homologous factors. Then, there is at least one position in $w$ that is not covered by any element of $\mathcal{A} \cup \hat{\mathcal{A}}$. 
Proposition

Let $A$ be the set of all admissible factors overlapping a position $\alpha$ in $w$ and $\hat{A}$ be the set of their respective homologous factors. Then, there is at least one position in $w$ that is not covered by any element of $A \cup \hat{A}$. 
Admissible factors

Proposition

Let \( A \) be the set of all admissible factors overlapping a position \( \alpha \) in \( w \) and \( \hat{A} \) be the set of their respective homologous factors. Then, there is at least one position in \( w \) that is not covered by any element of \( A \cup \hat{A} \).
Admissible factors

\[ w \equiv \hat{B} \]

\[ \alpha \]

\[ B \]

\[ A \]

\[ \hat{A} \]
Admissible factors

\[ w \equiv \alpha \hat{B}A - x - y \hat{A} \]
1. \(|x| = |y|\)

Lemma (Dorat and Nivat, 2003) (Brlek, Labelle and Lacasse, 2005)

In a non-intersecting closed path on a square lattice,
\[#(\text{left turns}) - #(\text{right turns}) = 4\]
1. $|x| = |y|$

Lemma (Dorat and Nivat, 2003) (Brlek, Labelle and Lacasse, 2005)

In a non-intersecting closed path on a square lattice,

$\#(\text{left turns}) - \#(\text{right turns}) = 4$.
1. $|x| = |y|$

$w \equiv \hat{x} B x A \hat{x} B \hat{x} U x V \hat{x} U \hat{x} V$
1. $|x| = |y|$

$$w \equiv \hat{x} \ U \ x \ V \ \hat{x} \ \hat{U} \ x \ \hat{V}.$$
Admissible factors

1. $|x| = |y|$

$$w \equiv \hat{x} \ U \ x \ V \ \hat{x} \ \hat{U} \ x \ \hat{V}. $$

**Lemma (Dorat and Nivat, 2003) (Brlek, Labelle and Lacasse, 2005)**

*In a non-intersecting closed path on a square lattice,*

$$\#(\text{left turns}) - \#(\text{right turns}) = 4.$$
Admissible factors

1. $|x| = |y|

$w \equiv \hat{x} U x V \hat{x} \hat{U} x \hat{V}.$

**Lemma (Dorat and Nivat, 2003) (Brlek, Labelle and Lacasse, 2005)**

*In a non-intersecting closed path on a square lattice,*

$\#(\text{left turns}) - \#(\text{right turns}) = 4.$
Admissible factors

1. $|x| = |y|$

$w \equiv \hat{x} \ U \ x \ V \ \hat{x} \ \hat{U} \ x \ \hat{V}.$

**Lemma (Dorat and Nivat, 2003) (Brlek, Labelle and Lacasse, 2005)**

*In a non-intersecting closed path on a square lattice,*

#(left turns) $- \#$(right turns) $= 4.$
Admissible factors

1. $|x| = |y|

$w \equiv \hat{B}$

$w \equiv \hat{x} U x V \hat{x} \hat{U} x \hat{V}$.

Lemma (Dorat and Nivat, 2003) (Brlek, Labelle and Lacasse, 2005)

In a non-intersecting closed path on a square lattice,

$\#(\text{left turns}) - \#(\text{right turns}) = 4$. 
1. $|x| = |y|$

Lemma (Dorat and Nivat, 2003) (Brlek, Labelle and Lacasse, 2005)

*In a non-intersecting closed path on a square lattice,*

$\#(\text{left turns}) - \#(\text{right turns}) = 4.$
1. $|x| = |y|$

Lemma (Dorat and Nivat, 2003) (Brlek, Labelle and Lacasse, 2005)

In a non-intersecting closed path on a square lattice,

\[ \#(\text{left turns}) - \#(\text{right turns}) = 4. \]
Admissible factors

1. $|x| = |y|$

$w \equiv \hat{x} B x A \hat{x} U x V \hat{x} \hat{U} x \hat{V}.$

Lemma (Dorat and Nivat, 2003) (Brlek, Labelle and Lacasse, 2005)

*In a non-intersecting closed path on a square lattice,*

$\#(\text{left turns}) - \#(\text{right turns}) = 4.$
2. $|x| \neq |y|$. 

$w \equiv \hat{\beta} \alpha \beta \gamma$, where $\alpha \beta = 0$. 

Xavier Provençal
On the problem of tiling the plane with a polyomino
2. \(|x| \neq |y|\).

\[ w \equiv \alpha \beta \gamma, \text{ where } \vec{\beta} = \vec{0}. \]
Admissible factors

2. \(|x| \neq |y|\).

\[ w \equiv \alpha \beta \gamma, \text{ where } \overrightarrow{\beta} = \overrightarrow{0}. \]
2. $|x| \neq |y|$. 

$$w \equiv \hat{B} - x - y.$$ 

$$w \equiv \alpha \beta \gamma, \text{ where } \vec{\beta} = \vec{0}.$$
2. $|x| \neq |y|$.  

\[ w \equiv \hat{B} \quad \hat{A} \]

\[ w \equiv \alpha \ \beta \ \gamma, \text{ where } \overrightarrow{\beta} = \overrightarrow{0}. \]
2. $|x| \neq |y|$. 

$w \equiv \alpha \beta \gamma$, where $\beta = 0$. 

$\alpha \beta \gamma$
2. $|x| \neq |y|$.

\[ w \equiv \alpha \beta \gamma, \text{ where } \beta = \vec{0}. \]
2. $|x| < |y|$, $\hat{y}$ does not overlap $\hat{A}$ in $\hat{B}$. 

$w \equiv \hat{B} - x - \hat{y} - V - \hat{V} - \alpha - \beta - \hat{A}$

$w = -\hat{\alpha} + \hat{\beta}$
2. \(|x| < |y|\), \(\hat{y}\) does not overlap \(\hat{A}\) in \(\hat{B}\).

\[
\begin{align*}
\hat{B} & \quad B \\
\hat{A} & \quad A \\
\end{align*}
\]
2. $|x| < |y|$, $\hat{y}$ does not overlap $\hat{A}$ in $\hat{B}$.
2. $|x| < |y|$, $\hat{y}$ does not overlap $\hat{A}$ in $\hat{B}$.

$$w \equiv \hat{B} - x - y - \hat{y} - V - \hat{V} - \alpha - \beta$$

$$w \equiv A + \hat{V} + \beta + \hat{A} + V + \alpha.$$
2. $|x| < |y|$, $\hat{y}$ does not overlap $\hat{A}$ in $\hat{B}$.
2. $|x| < |y|$, $\hat{y}$ does not overlap $\hat{A}$ in $\hat{B}$.
2. $|x| < |y|$, $\hat{y}$ does not overlap $\hat{A}$ in $\hat{B}$.

$$w \equiv \hat{B} \hat{A}$$

$$w \equiv A \hat{V} \beta \hat{A} V \alpha.$$
2. $|x| < |y|$, $\hat{y}$ does not overlap $\hat{A}$ in $\hat{B}$.

$w \equiv \hat{B} - x - \hat{y} - V - \hat{V} - \alpha - \beta - A - V - \hat{A}$.

$w \equiv A \hat{V} \beta \hat{A} V \alpha$.

$\overrightarrow{w} = \overrightarrow{A} + \overrightarrow{\hat{V}} + \overrightarrow{\beta} + \overrightarrow{\hat{A}} + \overrightarrow{V} + \overrightarrow{\alpha}$.
2. $|x| < |y|$, $\hat{y}$ does not overlap $\hat{A}$ in $\hat{B}$.

$$w \equiv \begin{array}{c}
\hat{B} \\
V \alpha x \\
\hat{y} \\
B \\
\hat{A}
\end{array}$$

$$w \equiv A \hat{V} \beta \hat{A} V \alpha.$$

$$\overrightarrow{w} = \overrightarrow{A} + \overrightarrow{\hat{V}} + \overrightarrow{\beta} + \overrightarrow{\hat{A}} + \overrightarrow{V} + \overrightarrow{\alpha} = \overrightarrow{\alpha} + \overrightarrow{\beta}$$
2. $|x| < |y|$, $\hat{y}$ does not overlap $\hat{A}$ in $\hat{B}$.

$w \equiv \hat{B} \quad \hat{B} \quad \hat{B}$

$\hat{V} \quad \alpha \quad x \quad \hat{A} \quad y \quad \hat{V} \quad \beta \quad \hat{A}$

$w \equiv A \hat{V} \quad \beta \quad \hat{A} \quad V \quad \alpha.$

$\mathbf{w} = \mathbf{A} + \mathbf{V} + \mathbf{\beta} + \mathbf{\hat{A}} + \mathbf{\hat{V}} + \mathbf{\alpha} = \mathbf{\alpha} + \mathbf{\beta} = \mathbf{0}.$
2. $|x| < |y|$, $\hat{y}$ does not overlap $\hat{A}$ in $\hat{B}$.

\[ w \equiv A \hat{V} \beta \hat{A} V \alpha. \]

\[ \overrightarrow{w} = \overrightarrow{A} + \overrightarrow{V} + \overrightarrow{\beta} + \overrightarrow{\hat{A}} + \overrightarrow{V} + \overrightarrow{\alpha} = \overrightarrow{\alpha} + \overrightarrow{\beta} = \overrightarrow{0}. \]
2. $|x| < |y|$, $\hat{y}$ does not overlap $\hat{A}$ in $\hat{B}$.

$$w \equiv \hat{B}$$

$$\begin{array}{cccc}
\hat{V} & \alpha & x & \hat{y} \\
A & y & \hat{V} & \beta & \hat{y} & \hat{A}
\end{array}$$

$$w \equiv A \hat{V} \beta \hat{A} V \alpha.$$
3. $|x| < |y|$, $\hat{y}$ does overlap $\hat{A}$ in $\hat{B}$.

$w \equiv \hat{B} - x - y - \hat{y} - \alpha - \beta - \gamma - \alpha - \beta - x$.

$\Rightarrow w = -\gamma + \alpha = 0$.

$\hat{y} \hat{y} \hat{y} = \alpha \beta x \Rightarrow \gamma \hat{y} \hat{y} = \hat{x} \hat{y} \alpha \beta = \alpha \beta \gamma \alpha \beta$.
3. $|x| < |y|$, $\hat{y}$ does overlap $\hat{A}$ in $\hat{B}$.

$$w \equiv \hat{B} - x - y - \hat{y} - \alpha - \beta - \gamma \Rightarrow w = -\alpha + -\gamma = -0$$

$\hat{y} \hat{y} = \alpha \beta \Rightarrow y \gamma \hat{y} = \hat{x} \hat{\beta} \hat{\alpha} \gamma \alpha \beta x$. 
Admissible factors

3. \(|x| < |y|\), \(\hat{y}\) does overlap \(\hat{A}\) in \(\hat{B}\).

\[ w \equiv \begin{array}{c}
\hat{B} \\
\alpha \beta x \\
\hat{y} \\
A \\
y \\
\hat{A} \\
\end{array} \]

\[ w = -\alpha + \beta + \gamma = 0. \]
Admissible factors

3. $|x| < |y|$, $\hat{y}$ does overlap $\hat{A}$ in $\hat{B}$.

$w \equiv \hat{B} \mathrel{{\sim}} B$
3. $|x| < |y|$, $\hat{y}$ does overlap $\hat{A}$ in $\hat{B}$.

$w \equiv \begin{cases} &\overset{\alpha}{\hat{y}} \overset{\beta}{x} & \overset{\gamma}{\hat{A}} \\ &B & A \\ \end{cases}$

$w \equiv A \gamma \hat{A} \beta.$
3. $|x| < |y|$, $\hat{y}$ does overlap $\hat{A}$ in $\hat{B}$.

$$w \equiv \hat{B}$$

$$w \equiv A \gamma \hat{A} \beta.$$
3. $|x| < |y|$, $\hat{y}$ does overlap $\hat{A}$ in $\hat{B}$.

\[
\begin{align*}
\mathbf{w} &\equiv \hat{B} \\
\alpha & \quad \beta \\
\hat{y} & \\
A & \quad y & \quad \gamma & \quad \hat{y} & \quad \hat{A}
\end{align*}
\]

\[
\begin{align*}
\mathbf{w} &\equiv A \quad \gamma \quad \hat{A} \quad \beta. \\
\mathbf{w} &= \mathbf{\beta} + \mathbf{\gamma} = \mathbf{0}.
\end{align*}
\]
3. $|x| < |y|$, $\hat{y}$ does overlap $\hat{A}$ in $\hat{B}$.

$$w \equiv \begin{array}{ccc} \hat{B} & \hat{y} & \hat{A} \\ A & y & \gamma & \hat{y} \\ \alpha & \beta & \hat{x} \end{array}$$

$$w \equiv A \gamma \hat{A} \beta.$$  

$$\vec{w} = \vec{\beta} + \vec{\gamma} = \vec{0}.$$ 

$$\hat{y} = \alpha \beta x \implies y \gamma \hat{y} = \hat{x} \hat{\beta} \hat{\alpha} \gamma \alpha \beta x.$$
3. $|x| < |y|$, $\hat{y}$ does overlap $\hat{A}$ in $\hat{B}$.

$$w \equiv \hat{B} \hat{A} B \alpha \beta x \gamma \hat{y} \hat{A} \beta.$$  

$$\vec{w} = \vec{\beta} + \vec{\gamma} = \vec{0}.$$  

$$\hat{y} = \alpha \beta x \implies y \gamma \hat{y} = \hat{x} \beta \hat{\alpha} \gamma \alpha \beta x.$$  

$$= \vec{0}$$
Corollary

Let $w$ a word coding a polyomino $p$ with Beauquier-Nivat’s factorization $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$. Then, $X$, $Y$ and $Z$ are admissible.
Corollary

Let \( w \) a word coding a polyomino \( p \) with Beauquier-Nivat’s factorization \( w \equiv XYZ \hat{X} \hat{Y} \hat{Z} \). Then, \( X, Y \) and \( Z \) are admissible.

- \( w \equiv Ax \hat{A} y \), for \( x, y \) such that \( |x| = |y| \).

- \( A \) is maximal, that is, \( \text{first}(x) \neq \text{last}(x) \) and \( \text{first}(y) \neq \text{last}(y) \).
Corollary

Let \( w \) a word coding a polyomino \( p \) with Beauquier-Nivat’s factorization \( w \equiv XYZ\hat{X}\hat{Y}\hat{Z} \). Then, \( X, Y \) and \( Z \) are admissible.

- \( w \equiv Ax\hat{A}y \), for \( x, y \) such that \( |x| = |y| \).
  
  Direct consequence of the fact that \( |u| = |\hat{u}| \) for all \( u \in \Sigma^* \).

- \( A \) is maximal, that is, \( \text{first}(x) \neq \text{last}(x) \) and \( \text{first}(y) \neq \text{last}(y) \).
Corollary

Let \( w \) a word coding a polyomino \( p \) with Beauquier-Nivat’s factorization \( w \equiv XYZ\hat{X}\hat{Y}\hat{Z} \). Then, \( X, Y \) and \( Z \) are admissible.

1. \( w \equiv Ax\hat{A}y \), for \( x, y \) such that \( |x| = |y| \).
   Direct consequence of the fact that \( |u| = |\hat{u}| \) for all \( u \in \Sigma^* \).

2. \( A \) is maximal, that is, \( \text{first}(x) \neq \text{last}(x) \) and \( \text{first}(y) \neq \text{last}(y) \).
   By contradiction, assume that \( X \) is not maximal, then \( \text{first}(YZ) = \text{last}(YZ) \).
Corollary

Let \( w \) a word coding a polyomino \( p \) with Beauquier-Nivat’s factorization \( w \equiv XYZ \hat{X} \hat{Y} \hat{Z} \). Then, \( X, Y \) and \( Z \) are admissible.

- \( w \equiv Ax \hat{A} y \), for \( x, y \) such that \( |x| = |y| \).
  Direct consequence of the fact that \( |u| = |\hat{u}| \) for all \( u \in \Sigma^* \).

- \( A \) is maximal, that is, \( \text{first}(x) \neq \text{last}(x) \) and \( \text{first}(y) \neq \text{last}(y) \).
  By contradiction, assume that \( X \) is not maximal, then \( \text{first}(YZ) = \text{last}(YZ) \).
  \[ YZ = \alpha Y'Z'\bar{\alpha} \]
Corollary

Let \( w \) a word coding a polyomino \( p \) with Beauquier-Nivat’s factorization \( w \equiv \text{XYZ} \hat{X} \hat{Y} \hat{Z} \). Then, \( X, Y \) and \( Z \) are admissible.

- \( w \equiv A\hat{x}\hat{y}, \) for \( x, y \) such that \( |x| = |y| \).
  Direct consequence of the fact that \( |u| = |\hat{u}| \) for all \( u \in \Sigma^* \).

- \( A \) is maximal, that is, \( \text{first}(x) \neq \text{last}(x) \) and \( \text{first}(y) \neq \text{last}(y) \).
  By contradiction, assume that \( X \) is not maximal, then \( \text{first}(YZ) = \text{last}(YZ) \).
  \( YZ = \alpha Y'Z'\bar{\alpha} \implies \hat{Y}\hat{Z} = \hat{\alpha}Y'\hat{Z}'\hat{\bar{\alpha}} = \hat{Y}'\hat{\bar{\alpha}}\alpha\hat{Z}' \).
Admissible factors

Corollary

Let $w$ a word coding a polyomino $p$ with Beauquier-Nivat’s factorization $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$. Then, $X$, $Y$ and $Z$ are admissible.

- $w \equiv Ax\hat{A}y$, for $x$, $y$ such that $|x| = |y|$. Direct consequence of the fact that $|u| = |\hat{u}|$ for all $u \in \Sigma^*$.

- $A$ is maximal, that is, $\text{first}(x) \neq \text{last}(x)$ and $\text{first}(y) \neq \text{last}(y)$.

$w \equiv XY\hat{X}\hat{Y}$ with $Y = \alpha Y'\overline{\alpha}$.
Admissible factors

**Corollary**

Let $w$ a word coding a polyomino $p$ with Beauquier-Nivat’s factorization $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$. Then, $X$, $Y$ and $Z$ are admissible.

- $w \equiv Ax\hat{A}y$, for $x, y$ such that $|x| = |y|$. Direct consequence of the fact that $|u| = |\hat{u}|$ for all $u \in \Sigma^*$.

- $A$ is maximal, that is, first$(x) \neq$ last$(x)$ and first$(y) \neq$ last$(y)$.

  $w \equiv XY\hat{X}\hat{Y}$ with $Y = \alpha Y'\overline{\alpha}$.

$w \equiv X\hat{X}Y\hat{Y}$
Admissible factors

Corollary

Let $w$ a word coding a polyomino $p$ with Beauquier-Nivat’s factorization $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$. Then, $X$, $Y$ and $Z$ are admissible.

- $w \equiv Ax\hat{A}y$, for $x$, $y$ such that $|x| = |y|$.
  Direct consequence of the fact that $|u| = |\hat{u}|$ for all $u \in \Sigma^*$.

- $A$ is maximal, that is, $\text{first}(x) \neq \text{last}(x)$ and $\text{first}(y) \neq \text{last}(y)$.

$w \equiv XY\hat{X}\hat{Y}$ with $Y = \alpha Y'\overline{\alpha}$.
Let $w$ a word coding a polyomino $p$ with Beauquier-Nivat’s factorization $w \equiv XYZX\hat{Y}\hat{Z}$. Then, $X$, $Y$ and $Z$ are admissible.

- $w \equiv Ax\hat{A}y$, for $x$, $y$ such that $|x| = |y|$. Direct consequence of the fact that $|u| = |\hat{u}|$ for all $u \in \Sigma^*$.

- $A$ is maximal, that is, $\text{first}(x) \neq \overline{\text{last}(x)}$ and $\text{first}(y) \neq \overline{\text{last}(y)}$.

$w \equiv XY\hat{X}\hat{Y}$ with $Y = \alpha Y'\overline{\alpha}$.

$w \equiv \hat{x}u\hat{u}u\hat{u}$ with $X = \hat{x}$ and $Y = \hat{y}$.
Corollary

Let $w$ a word coding a polyomino $p$ with Beauquier-Nivat’s factorization $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$. Then, $X$, $Y$ and $Z$ are admissible.

- $w \equiv Ax\hat{A}y$, for $x$, $y$ such that $|x| = |y|$. Direct consequence of the fact that $|u| = |\hat{u}|$ for all $u \in \Sigma^*$.

- $A$ is maximal, that is, $\text{first}(x) \neq \text{last}(x)$ and $\text{first}(y) \neq \text{last}(y)$.

$w \equiv XY\hat{X}\hat{Y}$ with $Y = \alpha Y'\bar{\alpha}$.

$w \equiv v\hat{v}u\hat{u}v\hat{u}v\hat{v}u\hat{u}$.
Corollary

Let $w$ a word coding a polyomino $p$ with Beauquier-Nivat's factorization $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$. Then, $X$, $Y$ and $Z$ are admissible.

- $w \equiv Ax\hat{A}y$, for $x, y$ such that $|x| = |y|$. Direct consequence of the fact that $|u| = |\hat{u}|$ for all $u \in \Sigma^*$.

- $A$ is maximal, that is, $\text{first}(x) \neq \text{last}(x)$ and $\text{first}(y) \neq \text{last}(y)$.

- $w \equiv XY\hat{X}\hat{Y}$ with $Y = \alpha Y'\alpha$. 

\[
\begin{array}{ccccccc}
X & Y & \hat{X} & \hat{Y} \\
\downarrow & \downarrow & \downarrow & \downarrow \\
\hat{v} & u & \hat{u} & v & \hat{v} & u & \hat{u} \\
\hat{X} & \hat{Y} & \hat{Y} & \hat{X} & \hat{Y} & \hat{X} & \hat{Y}
\end{array}
\]
Corollary

Let $w$ a word coding a polyomino $p$ with Beauquier-Nivat’s factorization $w \equiv XYZ\hat{X}\hat{Y}\hat{Z}$. Then, $X$, $Y$ and $Z$ are admissible.

- $w \equiv Ax\hat{A}y$, for $x$, $y$ such that $|x| = |y|$.
  
  Direct consequence of the fact that $|u| = |\hat{u}|$ for all $u \in \Sigma^*$.

- $A$ is maximal, that is, $\text{first}(x) \neq \text{last}(x)$ and $\text{first}(y) \neq \text{last}(y)$.

  $w \equiv XY\hat{X}\hat{Y}$ with $Y = \alpha Y'\bar{\alpha}$.
Admissible factors

\[ w \equiv a a a b a b \overline{a} b \overline{a} \overline{a} \overline{a} \overline{b} a b a b \overline{a} b a b \]
Admissible factors

The tiling problem
Beauquier-Nivat characterization
A fast algorithm to detect exact polyominoes

Admissible factors, detection and properties
Detecting pseudo-squares
Detection pseudo-hexagons

Admissible factors

\[ w \equiv a \ a \ a \ b \ a \ b \ \bar{a} \ b \ \bar{a} \ a \ \bar{a} \ b \ \bar{a} \ b \ a \ b \]

\[ A \quad \times \quad \hat{A} \quad y \]

Xavier Provençal
On the problem of tiling the plane with a polyomino
Admissible factors

\[ w \equiv a \ a \ a \ b \ a \ b \ \bar{a} \ b \ \bar{a} \ a \ a \ \bar{b} \ \bar{a} \ b \ a \ b \]

\[
\hat{A} \ \\
\mu \ \\
\nu \ \\
\lambda \ \\
\sigma \ \\
\tau \ \\
\varphi \]

Xavier Provençal
On the problem of tiling the plane with a polyomino
Admissible factors

$w \equiv a a a b a b \overline{a} b \overline{a} a \overline{a} \overline{b} a \overline{a} b a b$

$X Y Z \hat{X} \hat{Y} \hat{Z}$
Admissible factors

The tiling problem
Beauquier-Nivat characterization
A fast algorithm to detect exact polyominoes

Admissible factors, detection and properties
Detecting pseudo-squares
Detection pseudo-hexagons

Admissible factors

On the problem of tiling the plane with a polyomino
Listing admissible factors

Lemma

Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.
Listing admissible factors

**Lemma**

*Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.*

If $w \equiv A \times \hat{A} y$ then $\hat{w} \equiv \hat{y} A \times \hat{A}$.

$w \equiv \alpha$

$\hat{w} \equiv$
Lemma

Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.

If $w \equiv A \times \hat{A} \ y$ then $\hat{w} \equiv \hat{y} \ A \hat{x} \hat{A}$.

\[
\begin{array}{cccccccccccccccc}
\hline
& & & & & & & & & & & & & & & \\
\hline
w \equiv & & & & & & & & & & & & & & & \\
\alpha & & & & & & & & & & & & & & & \\
\hat{w} \equiv & & & & & & & & & & & & & & & \\
\hline
\end{array}
\]
Lemma

Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.

If $w \equiv A \times \hat{A} y$ then $\hat{w} \equiv \hat{y} A \hat{x} \hat{A}$.

\[ w \equiv \begin{array}{cccccccccccccccccccc} & & & & \alpha \end{array} \]

\[ \hat{w} \equiv \begin{array}{cccccccccccccccccccc} & & & & \end{array} \]
Listing admissible factors

Lemma

Given a position \( \alpha \) in the word \( w \) coding a polyomino, all the admissible factors overlapping \( \alpha \) can be listed in linear time.

If \( w \equiv A \times \hat{A} y \) then \( \hat{w} \equiv \hat{y} A \hat{x} \hat{A} \).

\[ w \equiv \]
\[
\alpha
\]
\[ \hat{w} \equiv \]
\[
\hat{x} \hat{A} \hat{x} \hat{A}
\]
Listing admissible factors

Lemma

Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.

If $w \equiv A \times \hat{A} y$ then $\hat{w} \equiv \hat{y} A \hat{x} \hat{A}$.

\[ w \equiv \begin{array}{cccccccccccccccccccccccccc}
\end{array} \]
\[ \alpha \]

\[ \hat{w} \equiv \begin{array}{cccccccccccccccccccccccccc}
\end{array} \]
Listing admissible factors

Lemma

Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.

If $w \equiv A \times \hat{A} y$ then $\hat{w} \equiv \hat{y} A \hat{x} \hat{A}$.

\[
\begin{align*}
\begin{array}{c}
\hat{w} \equiv \\
\end{array}
\end{align*}
\]

\[
\begin{align*}
\begin{array}{c}
w \equiv \\
\end{array}
\end{align*}
\]

Xavier Provençal
On the problem of tiling the plane with a polyomino
Lemma

Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.

If $w \equiv A \times \hat{A} y$ then $\hat{w} \equiv \hat{y} A \hat{x} \hat{A}$.

\[
\begin{align*}
    w & \equiv \begin{array}{cccccccc}
        & & & & & & & \\
        & & & & & & & \\
        & & & & & & & \\
        & & & & & & & \\
        & & & & & & & \\
        \alpha & & & & & & & \\
    \end{array} \\
    \hat{w} & \equiv \begin{array}{cccccccc}
        & & & & & & & \\
        & & & & & & & \\
        & & & & & & & \\
        & & & & & & & \\
        & & & & & & & \\
        & & & & & & & \\
    \end{array}
\end{align*}
\]
Listing admissible factors

Lemma

Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.

If $w \equiv A \times \hat{A} y$ then $\hat{w} \equiv \hat{y} A \hat{x} \hat{A}$.

\[
\begin{align*}
\hat{w} & \equiv A \\
\w & \equiv A \times \hat{A} y
\end{align*}
\]
Lemma

*Given a position* \( \alpha \) *in the word* \( w \) *coding a polyomino, all the admissible factors overlapping* \( \alpha \) *can be listed in linear time.*

If \( w \equiv A x \hat{A} y \) then \( \hat{w} \equiv \hat{y} A \hat{x} \hat{A} \).

\[
\begin{align*}
\hat{w} & \equiv \\
\hat{w} & \equiv
\end{align*}
\]
Listing admissible factors

Lemma

*Given a position* \( \alpha \) *in the word* \( w \) *coding a polyomino, all the admissible factors overlapping* \( \alpha \) *can be listed in linear time.*

If \( w \equiv A x \hat{A} y \) then \( \hat{w} \equiv \hat{y} A \hat{x} \hat{A} \).

\[
\begin{array}{c}
| & | & | & | & | & | & | & | & | & | & | & | \hline
| & | & | & | & | & | & | & | & | & | & | \hline
\end{array}
\]

\( w \equiv \framebox{\begin{array}{c}
| & | & | & | & | \hline
| & | & | \hline
\end{array}} \alpha \)

\[
\begin{array}{c}
| & | & | & | & | & | & | & | & | & | & | & | \hline
| & | & | & | & | & | & | & | & | & | & | \hline
\end{array}
\]

\( \hat{w} \equiv \framebox{\begin{array}{c}
| & | & | & | & | \hline
| & | & | \hline
\end{array}} \hat{x} \hat{A} \hat{y} \hat{A} \hat{x} \)
Lemma

Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.

If $w \equiv A \times \hat{A} y$ then $\hat{w} \equiv \hat{y} A \hat{x} \hat{A}$.
Lemma

Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.

If $w \equiv A \times \hat{A} y$ then $\hat{w} \equiv \hat{y} A \hat{x} \hat{A}$.

\[ w \equiv \underbrace{\alpha}_{\text{overlap}} \]

\[ \hat{w} \equiv \hat{A} \]
Listing admissible factors

Lemma

*Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.*

If $w \equiv A \times \hat{A} y$ then $\hat{w} \equiv \hat{y} A \hat{x} \hat{A}$.

| $w \equiv$ | [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] |
|  $\alpha$ |

| $\hat{w} \equiv$ | [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] |
|  | [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] |

Xavier Provençal  On the problem of tiling the plane with a polyomino
Listing admissible factors

Lemma

Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.

If $w \equiv A \times \hat{A} y$ then $\hat{w} \equiv \hat{y} A \hat{x} \hat{A}$.

$w \equiv \begin{array}{cccccccccccccccc}
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
\alpha
\end{array}$

$\hat{w} \equiv \begin{array}{cccccccccccccccc}
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & \\
\end{array}$
Listing admissible factors

Lemma

Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.

If $w \equiv A \times \hat{A} \ y$ then $\hat{w} \equiv \hat{y} \ A \ \hat{x} \ \hat{A}$.

\[ w \equiv \begin{array}{ccccccccccccccccc} & & & & & & & & & & & & & & & & \\ \end{array} \]
\[ \alpha \]

\[ \hat{w} \equiv \begin{array}{ccccccccccccccccc} & & & & & & & & & & & & & & & & \\ \end{array} \]
Listing admissible factors

Lemma

Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.

If $w \equiv A \times \hat{A} \ y$ then $\hat{w} \equiv \hat{y} \ A \ \hat{x} \ \hat{A}$.

$w \equiv \begin{array}{cccccccccccccccc}
\vspace{0.5cm}
\end{array}$

$\alpha$

$\hat{w} \equiv \begin{array}{cccccccccccccccc}
\vspace{0.5cm}
\end{array}$

Xavier Provençal

On the problem of tiling the plane with a polyomino
Listing admissible factors

Lemma

Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.

If $w \equiv A \times \hat{A} y$ then $\hat{w} \equiv \hat{y} A \hat{x} \hat{A}$.

\[
\begin{array}{cccccccccc}
  & & & & & & & & & \\
\end{array}
\]

$w \equiv \alpha$

\[
\begin{array}{cccccccccc}
  & & & & & & & & & \\
\end{array}
\]

$\hat{w} \equiv \alpha$

\[
\begin{array}{cccccccccc}
  & & & & & & & & & \\
\end{array}
\]

Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.

If $w \equiv A \times \hat{A} y$ then $\hat{w} \equiv \hat{y} A \hat{x} \hat{A}$.
Listing admissible factors

Lemma

Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.

If $w \equiv A \times \hat{A} \ y$ then $\hat{w} \equiv \hat{y} \ A \hat{x} \hat{A}$.

\[
\begin{array}{ccccccccccccccc}
\text{\hat{w}} & \equiv & & & & & & & & & & & & \\
\text{\hat{y} A \hat{x} \hat{A}} & \equiv & & & & & & & & & & & & \\
\text{\hat{y} A \hat{x} \hat{A}} & \equiv & & & & & & & & & & & & \\
\end{array}
\]

\[
\begin{array}{ccccccccccccccc}
w & \equiv & & & & & & & & & & & & \\
\text{A \times \hat{A} \ y} & \equiv & & & & & & & & & & & & \\
\text{A \times \hat{A} \ y} & \equiv & & & & & & & & & & & & \\
\end{array}
\]

\[
\begin{array}{ccccccccccccccc}
\alpha & \equiv & & & & & & & & & & & & \\
\end{array}
\]
Lemma

Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.

If $w \equiv A \times \hat{A} y$ then $\hat{w} \equiv \hat{y} A \hat{x} \hat{A}$.

$w \equiv \begin{array}{cccccccccccccccccccc}
\text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } \\
\alpha & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } \\
\end{array}$

$\hat{w} \equiv \begin{array}{cccccccccccccccccccc}
\text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } & \text{          } \\
\end{array}$
Lemma

Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.

If $w \equiv A \times \hat{A} \ y$ then $\hat{w} \equiv \hat{y} \ A \times \hat{A}$.
Lemma

Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.

If $w \equiv A \times \hat{A} y$ then $\hat{w} \equiv \hat{y} A \hat{x} \hat{A}$.

\[
\begin{array}{cccccccccccccccc}
\, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, \\
\, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, \\
\, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, \\
\, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, \\
\, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, \\
\, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, \\
\, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, \\
\, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, \\
\, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, \\
\alpha
\end{array}
\]

\[
\begin{array}{cccccccccccccccc}
\, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, \\
\, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, \\
\, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, \\
\, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, \\
\, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, \\
\, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, \\
\, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, \\
\, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, \\
\, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, & \, \\
\hat{\alpha}
\end{array}
\]
Listing admissible factors

Lemma

Given a position $\alpha$ in the word $w$ coding a polyomino, all the admissible factors overlapping $\alpha$ can be listed in linear time.

If $w \equiv A \times \hat{A} \ y$ then $\hat{w} \equiv \hat{y} \ A \hat{x} \ \hat{A}$.

$w \equiv \begin{array}{cccccccccccccccccccc}
\alpha \\
\end{array}$

$\hat{w} \equiv \begin{array}{cccccccccccccccccccc}
\end{array}$

Xavier Provençal

On the problem of tiling the plane with a polyomino
Listing admissible factors

Lemma

*Given a position* $\alpha$ *in the word* $w$ *coding a polyomino, all the admissible factors overlapping* $\alpha$ *can be listed in linear time.*

If $w \equiv A \times \hat{A} y$ then $\hat{w} \equiv \hat{y} A \times \hat{A}$.

\[
\begin{align*}
\hat{w} & \equiv \\
\end{align*}
\]
Detecting pseudo-squares

Theorem

\textit{Let }w\textit{ be the boundary of }p. \textit{Determining if }w\textit{ codes a pseudo-square \textit{is decidable in linear time.}
Theorem

Let $w$ be the boundary of $p$. Determining if $w$ codes a pseudo-square is decidable in linear time.
Detecting pseudo-squares

Theorem

Let \( w \) be the boundary of \( p \). Determining if \( w \) codes a pseudo-square is decidable in linear time.

If \( x = \hat{y} \) then \( w \equiv XY\hat{X}\hat{Y} \).
Theorem

Let \( w \) be the boundary of \( p \). Determining if \( w \) codes a pseudo-square is decidable in linear time.

If \( x = \hat{y} \) then \( w \equiv XY\hat{X}\hat{Y} \).
Since \( w \equiv A x \hat{A} y \) then \( \hat{w} \equiv \hat{y} A \hat{x} \hat{A} \).
Let $w$ be the boundary of $p$. Determining if $w$ codes a pseudo-square is decidable in linear time.

If $x = \hat{y}$ then $w \equiv XY\hat{X}\hat{Y}$.
Since $w \equiv A x \hat{A} y$ then $\hat{w} \equiv \hat{y} A \hat{x} \hat{A}$. 
Detecting pseudo-squares

**Theorem**

Let $w$ be the boundary of $p$. Determining if $w$ codes a pseudo-square *is decidable in linear time.*

If $x = \hat{y}$ then $w \equiv XY\hat{X}\hat{Y}$.
Since $w \equiv A x \hat{A} y$ then $\hat{w} \equiv \hat{y} A \hat{x} \hat{A}$. 
**k-square-free words**

**Definition**

A word \( w \) is \( k \)-square-free if

\[
\max \{|f| : f \in Squares(w)\} < k.
\]
**Definition**

A word $w$ is $k$-square-free if

$$\max \{|f| : f \in \text{Squares}(w)\} < k.$$  

**Exemple** : $w = a a b a b b a$ is $k$-square-free for $k \geq 5$. 

---

**Exemple** : $w = a a b a b b a$ is $k$-square-free for $k \geq 5$. 

---

**Lemma**

Let $w$ be a $k$-square-free word coding a polyomino, and let $\alpha$ be a position in $w$. The number of admissible factors overlapping $\alpha$ in $w$ is bounded by $4k + 2 \log(n)$. 

---

Xavier Provençal
On the problem of tiling the plane with a polyomino
**k-square-free words**

**Definition**

A word \(w\) is \(k\)-square-free if

\[
\max \{|f| : f \in \text{Squares}(w)\} < k.
\]

**Exemple** : \(w = a \ a \ b \ a \ b \ b \ a\) is \(k\)-square-free for \(k \geq 5\).

**Lemma**

Let \(w\) be a \(k\)-square-free word coding a polyomino, and let \(\alpha\) be a position in \(w\). The number of admissible factors overlapping \(\alpha\) in \(w\) is bounded by \(4k + 2 \log(n)\).
Detecting pseudo-hexagons

Theorem

Let $w$ be a $k$-square-free word coding a polyomino, with $k \in O(\sqrt{n})$. Determining if $w$ codes a pseudo-hexagon is decidable in linear time.
Detecting pseudo-hexagons

Input : $w \in \Sigma^*$ coding a polyomino $p$.

Build $L_1$ the list of all admissible factors that overlap the position $\alpha$.

$\beta :=$ (the position of the rightmost letter of $w$ include in a factor of $L_1$) + 1.

Build $L_2$ the list of all admissible factors that overlap the position $\beta$.

For all $X \in L_1$ do
  For all $Y \in L_2$ do
    If $w \equiv XYx\hat{X}\hat{Y}y$ then
      Compute $i$ : the position of $x$ in $w$.
      Compute $j$ : the position of $\hat{y}$ in $\hat{w}$.
      If longest common extention($w, \hat{w}, i, j$) = $|x|$ then
        $p$ is a pseudo-hexagon.
    End if
  End if
End for

End for
Detecting pseudo-hexagons

Input: $w \in \Sigma^*$ coding a polyomino $p$.

**Build** $L_1$ the list of all admissible factors that overlap the position $\alpha$.

$\beta := \text{(the position of the rightmost letter of } w \text{ include in a factor of } L_1) + 1$.

**Build** $L_2$ the list of all admissible factors that overlap the position $\beta$.

For all $X \in L_1$ do

* For all $Y \in L_2$ do

* If $w \equiv XYx\hat{X}\hat{Y}y$ then

  * Compute $i$: the position of $x$ in $w$.
  * Compute $j$: the position of $\hat{y}$ in $\hat{w}$.
  * If $\text{longest common extension}(w, \hat{w}, i, j) = |x|$ then

    * $p$ is a pseudo-hexagon.

End if

End if

End for

End for
Detecting pseudo-hexagons

**Input**: \( w \in \Sigma^* \) coding a polyomino \( p \).

**Build** \( L_1 \) the list of all admissible factors that overlap the position \( \alpha \).

\[ \beta := (\text{the position of the rightmost letter of } w \text{ include in a factor of } L_1) + 1. \]

**Build** \( L_2 \) the list of all admissible factors that overlap the position \( \beta \).

**For** all \( X \in L_1 \) **do**

**For** all \( Y \in L_2 \) **do**

**If** \( w \equiv XYX\hat{X}Y \) **then**

- **Compute** \( i \) : the position of \( x \) in \( w \).
- **Compute** \( j \) : the position of \( \hat{y} \) in \( \hat{w} \).
- **If** longest common extention\((w, \hat{w}, i, j) = |x| \) **then**
  - \( p \) is a pseudo-hexagon.

**End if**

**End if**

**End for**

**End for**
Detecting pseudo-hexagons

Input : \( w \in \Sigma^* \) coding a polyomino \( p \).

Build \( L_1 \) the list of all admissible factors that overlap the position \( \alpha \).

\[ \beta := (\text{the position of the rightmost letter of } w \text{ include in a factor of } L_1) + 1. \]

Build \( L_2 \) the list of all admissible factors that overlap the position \( \beta \).

For all \( X \in L_1 \) do
  For all \( Y \in L_2 \) do
    If \( w \equiv XYx\hat{X}\hat{Y}y \) then
      Compute \( i \) : the position of \( x \) in \( w \).
      Compute \( j \) : the position of \( \hat{y} \) in \( \hat{w} \).
      If longest common extention\((w, \hat{w}, i, j) = |x| \) then
        \( p \) is a pseudo-hexagon.
    End if
  End if
End for
End for

\( w \equiv X \hat{X} \hat{Y} \)
Detecting pseudo-hexagons

Input : \( w \in \Sigma^* \) coding a polyomino \( p \).

Build \( L_1 \) the list of all admissible factors that overlap the position \( \alpha \).
\( \beta := (\text{the position of the rightmost letter of} \ w \ \text{include in a factor of} \ L_1) + 1 \).
Build \( L_2 \) the list of all admissible factors that overlap the position \( \beta \).

For all \( X \in L_1 \) do

For all \( Y \in L_2 \) do

If \( w \equiv XY\hat{X}\hat{Y}y \) then

Compute \( i : \) the position of \( x \) in \( w \).
Compute \( j : \) the position of \( \hat{y} \) in \( \hat{w} \).
If longest common extention(\( w, \hat{w}, i, j \) = |\( x | \) then

\( p \) is a pseudo-hexagon.
End if
End if
End if
End for

End for
Detecting pseudo-hexagons

**Input:** \( w \in \Sigma^* \) coding a polyomino \( p \).

**Build** \( L_1 \) the list of all admissible factors that overlap the position \( \alpha \).

\[ \beta := (\text{the position of the rightmost letter of } w \text{ include in a factor of } L_1) + 1. \]

**Build** \( L_2 \) the list of all admissible factors that overlap the position \( \beta \).

**For** all \( X \in L_1 \) **do**

**For** all \( Y \in L_2 \) **do**

**If** \( w \equiv XYx\hat{X}\hat{Y}y \) **then**

**Compute** \( i \) : the position of \( x \) in \( w \).

**Compute** \( j \) : the position of \( \hat{y} \) in \( \hat{w} \).

**If** longest common extension\((w, \hat{w}, i, j) = |x| \) **then**

\( p \) is a pseudo-hexagon.

**End if**

**End if**

**End for**

**End for**
Detecting pseudo-hexagons

Input: $w \in \Sigma^*$ coding a polyomino $p$.

Build $L_1$ the list of all admissible factors that overlap the position $\alpha$. 
\[ \beta := \text{(the position of the rightmost letter of } w \text{ include in a factor of } L_1) + 1. \]

Build $L_2$ the list of all admissible factors that overlap the position $\beta$.

For all $X \in L_1$ do
  For all $Y \in L_2$ do
    If $w \equiv XYx\hat{X}\hat{Y}y$ then
      Compute $i$ : the position of $x$ in $w$.
      Compute $j$ : the position of $\hat{y}$ in $\hat{w}$.
      If longest common extention($w, \hat{w}, i, j$) = $|x|$ then
        $p$ is a pseudo-hexagon.
    End if
  End if
End for

End if

End for

$w \equiv \begin{bmatrix} y & X & Y & x & \hat{X} & \hat{Y} \end{bmatrix}$

$\hat{w} \equiv \begin{bmatrix} Y & X & \hat{x} & \hat{y} & \hat{X} & \hat{Y} \end{bmatrix}$
Detecting pseudo-hexagons

Input: \( w \in \Sigma^* \) coding a polyomino \( p \).

Build \( L_1 \) the list of all admissible factors that overlap the position \( \alpha \).

\( \beta := (\text{the position of the rightmost letter of } w \text{ include in a factor of } L_1) + 1 \).

Build \( L_2 \) the list of all admissible factors that overlap the position \( \beta \).

For all \( X \in L_1 \) do

For all \( Y \in L_2 \) do

If \( w \equiv XYx\hat{X}\hat{Y}_y \) then

Compute \( i \): the position of \( x \) in \( w \).

Compute \( j \): the position of \( \hat{y} \) in \( \hat{w} \).

If longest common extension(\( w, \hat{w}, i, j \) = \( |x| \) then

\( p \) is a pseudo-hexagon.

End if

End if

End for

End for
Detecting pseudo-hexagons

**Input**: \( w \in \Sigma^* \) coding a polyomino \( p \).

**Build** \( L_1 \) the list of all admissible factors that overlap the position \( \alpha \).

\[ \beta := ( \text{the position of the rightmost letter of } w \text{ include in a factor of } L_1 ) + 1. \]

**Build** \( L_2 \) the list of all admissible factors that overlap the position \( \beta \).

**For** all \( X \in L_1 \) **do**

**For** all \( Y \in L_2 \) **do**

**If** \( w \equiv XYx\hat{X}\hat{Y}y \) **then**

**Compute** \( i \) : the position of \( x \) in \( w \).

**Compute** \( j \) : the position of \( \hat{y} \) in \( \hat{w} \).

**If** longest common extension \( (w, \hat{w}, i, j) = |x| \) **then**

\( p \) is a pseudo-hexagon.

**End if**

**End if**

**End for**

**End for**

\( w \equiv yXyX\hat{X}\hat{Y}y \)

\( \hat{w} \equiv YX\hat{x}\hat{y}\hat{X}\hat{Y} \)
Detecting pseudo-hexagons

**Input:** \( w \in \Sigma^* \) coding a polyomino \( p \).

**Build** \( L_1 \) the list of all admissible factors that overlap the position \( \alpha \).

\( \beta := \) (the position of the rightmost letter of \( w \) include in a factor of \( L_1 \)) + 1.

**Build** \( L_2 \) the list of all admissible factors that overlap the position \( \beta \).

**For all** \( X \in L_1 \) **do**

**For all** \( Y \in L_2 \) **do**

**If** \( w \equiv XYx\hat{X}\hat{Y}y \) **then**

**Compute** \( i \) : the position of \( x \) in \( w \).

**Compute** \( j \) : the position of \( \hat{y} \) in \( \hat{w} \).

**If** longest common extension(\( w, \hat{w}, i, j \) = \( |x| \) **then**

\( p \) is a pseudo-hexagon.

**End if**

**End if**

**End for**

**End for**

\( w \equiv \hat{w} \equiv \)

\( \begin{array}{c}
\hat{Z} \\
X \\
Y \\
Z \\
\hat{X} \\
\hat{Y} \\
\end{array} \)

\( \begin{array}{c}
\hat{Y} \\
X \\
\hat{x} \\
\hat{Y} \\
\hat{X} \\
\end{array} \)
Detecting pseudo-hexagons

Input: \( w \in \Sigma^* \) coding a polyomino \( p \).

Build \( L_1 \) the list of all admissible factors that overlap the position \( \alpha \).
\[ \beta := (\text{the position of the rightmost letter of } w \text{ include in a factor of } L_1) + 1. \]

Build \( L_2 \) the list of all admissible factors that overlap the position \( \beta \).

For all \( X \in L_1 \) do

For all \( Y \in L_2 \) do

If \( w \equiv XYx\hat{X}\hat{Y}y \) then

Compute \( i \) : the position of \( x \) in \( w \).

Compute \( j \) : the position of \( \hat{y} \) in \( \hat{w} \).

If longest common extention(\( w, \hat{w}, i, j \) = \( |x| \) then

\( p \) is a pseudo-hexagon.

End if

End if

End for

End for

\[ O \left(n + (k + \log n)^2\right) = O(n) \]
THANK YOU!