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

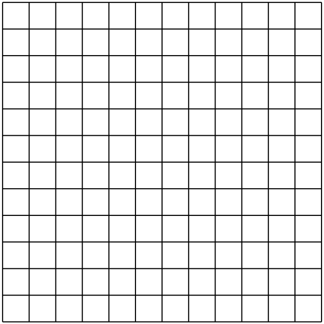
1 The tiling problem

2 Beauquier-Nivat characterization

3 A fast algorithm to detect exact polyominoes

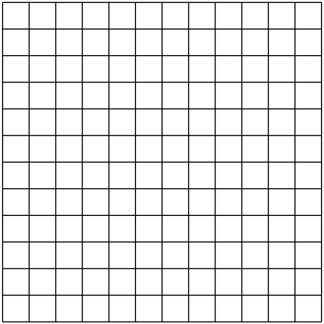
Introduction to polyominoes

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



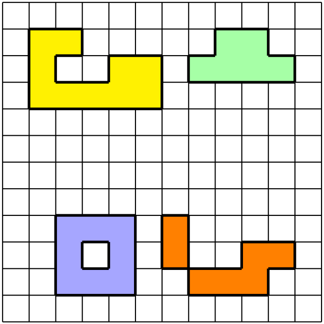
Introduction to polyominoes

- Discrete plane : $\mathbb{Z}^2$
- **Definition** : A *polyomino* is a finite, 4-connected subset of the plane, without holes.



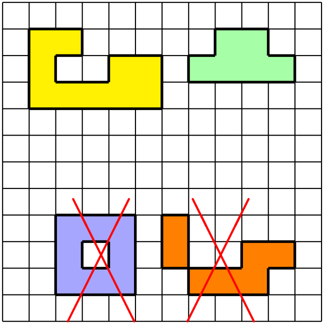
Introduction to polyominoes

- Discrete plane : $\mathbb{Z}^2$
- **Definition** : A *polyomino* is a finite, 4-connected subset of the plane, without holes.



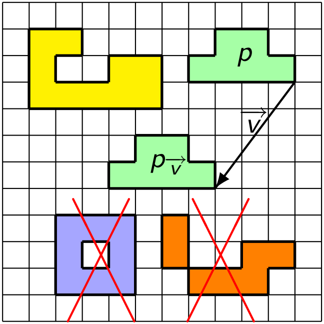
Introduction to polyominoes

- 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 $\overrightarrow{v}$ a vector of $\mathbb{Z}^2$, $p_{\overrightarrow{v}}$ will denote the image of p by de translation $\overrightarrow{v}$.



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 :

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 $p_{\overrightarrow{u}}$.*

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 two distinct* $(p, \overrightarrow{u}), (p', \overrightarrow{v}) \in \mathcal{T}$, $p_{\overrightarrow{u}}$ *and* $p'_{\overrightarrow{v}}$ *are non-overlapping.*

General statement of the tiling problem

Definition (Tiling)

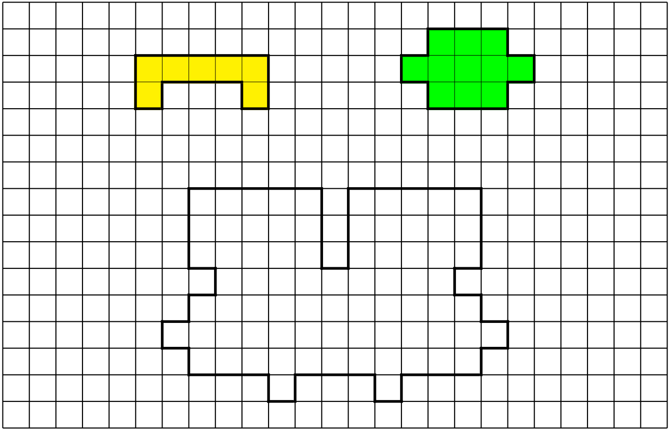
A tiling $\mathcal{T}$ of a subset $D \subset \mathbb{Z}^2$ by a set of polyominoes $\mathcal{P}$ is a set of couples $(p, \overrightarrow{u}) \in \mathcal{P} \times \mathbb{Z}^2$ such that :

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

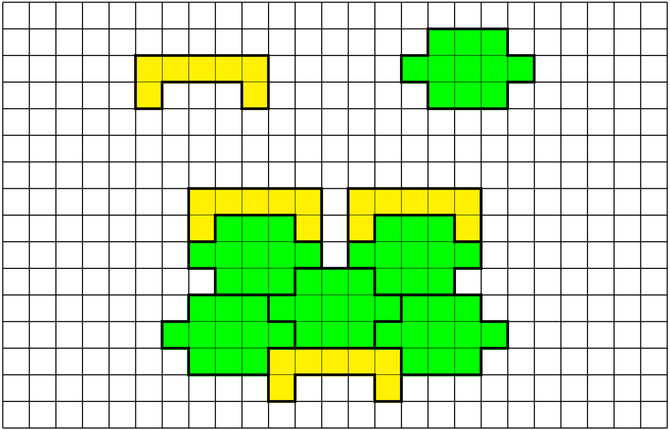
Definition (The Tiling Problem)

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 $\mathcal{P} = \{$  ,  $\}$ *is in P.*

Finite case

Remark

The tiling problem with D finite is in NP.

Remark

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

Theorem (Garey, Johnson and Papadimitriou)

The tiling problem with D finite and $\mathcal{P} = \{$  ,  $\}$ *is NP-Complete.*

Infinite case

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

Infinite case

We concider 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 independant vectors $\overrightarrow{u}$ and $\overrightarrow{v}$ such that $\mathcal{T}$ is not changed by the corresponding translations.

Infinite case

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

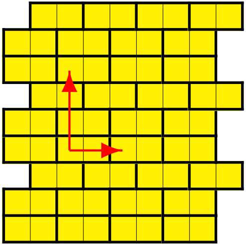
Definition (Periodic Tiling)

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

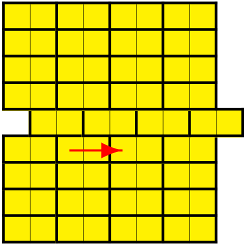
Definition (Half-Periodic Tiling)

A tiling $\mathcal{T}$ is half-periodic if there exists a vectors $\overrightarrow{u}$ such that $\mathcal{T}$ is not changed by the corresponding translation.

Example



Periodic tiling



Half-periodic tiling

Half-Periodic implies periodic

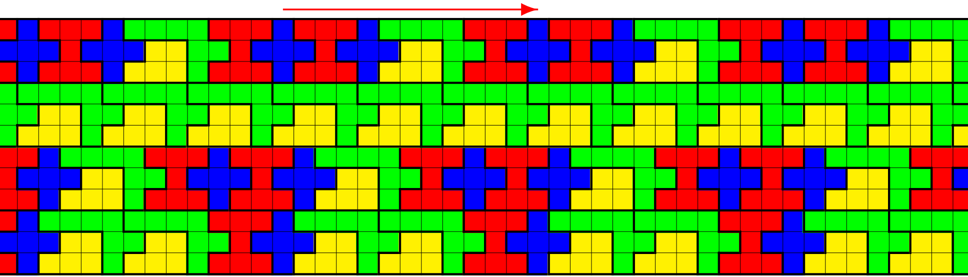
Remark

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

Half-Periodic implies periodic

Remark

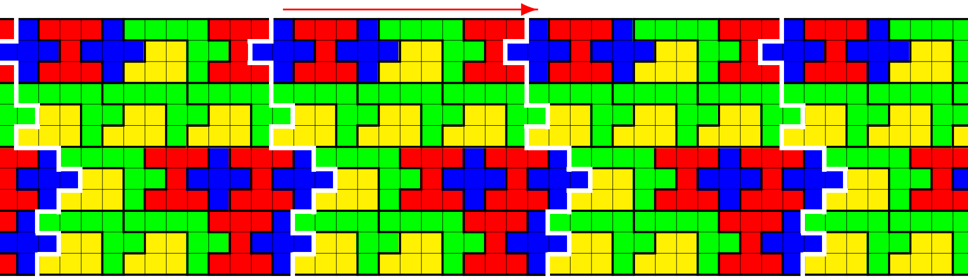
If there is an half-periodic tiling of the plane by $\mathcal{P}$, then there is also a periodic one.



Half-Periodic implies periodic

Remark

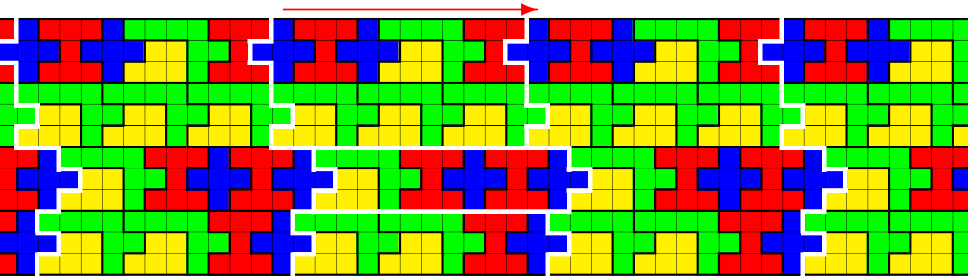
If there is an half-periodic tiling of the plane by $\mathcal{P}$, then there is also a periodic one.



Half-Periodic implies periodic

Remark

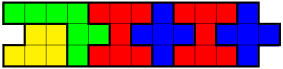
If there is an half-periodic tiling of the plane by $\mathcal{P}$, then there is also a periodic one.



Half-Periodic implies periodic

Remark

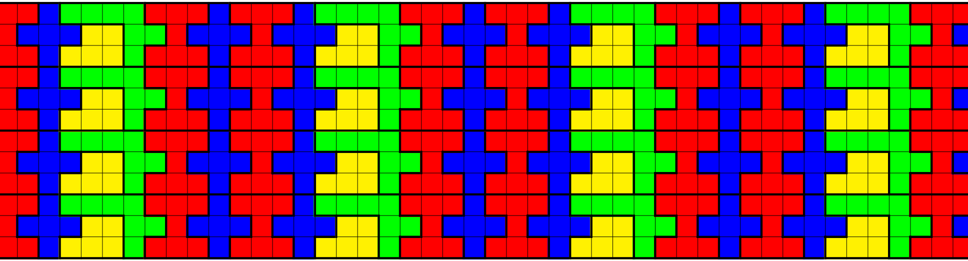
If there is an half-periodic tiling of the plane by $\mathcal{P}$, then there is also a periodic one.



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.



Nonperiodic tilings

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 $\mathcal{P} = \{p\}$ *tiles the plane.*

Tilings with one polyomino

Definition

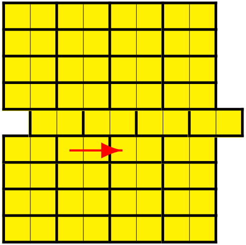
A polyomino p is exact if the set $\mathcal{P} = \{p\}$ *tiles the plane.*

Definition

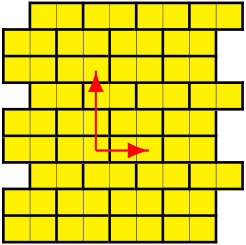
A tiling of the plane $\mathcal{T}$ *by an exact polyomino p is regular if there exist two vectors* $\overrightarrow{u}$ *and* $\overrightarrow{v}$ *such that*

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

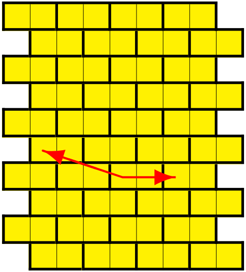
Examples



Half-periodic tiling



Periodic tiling



Regular tiling

Tilings with one polyomino

Theorem (Wijshoff and Van Leeuven, 1984)

If a polyomino p tiles the plane, then there exists a regular tiling of the plane by p.

Tilings with one polyomino

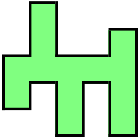
Theorem (Wijshoff and Van Leeuwen, 1984)

If a polyomino p tiles the plane, then there exists a regular tiling of the plane by p.

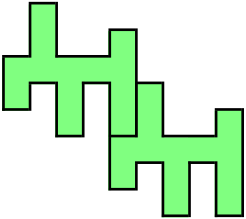
Corollary

The tiling problem with $\mathcal{P} = \{p\}$ and $D = \mathbb{Z}^2$ is decidable in polynomial time.

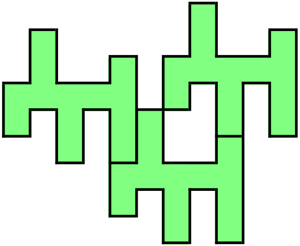
Example



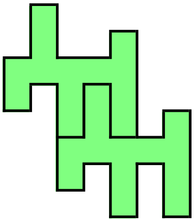
Example



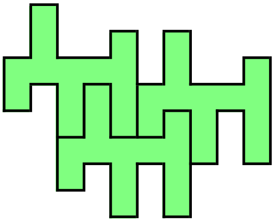
Example



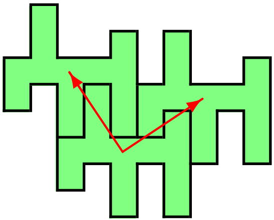
Example



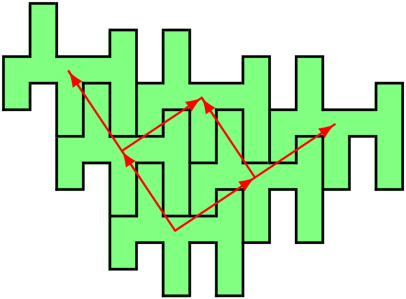
Example



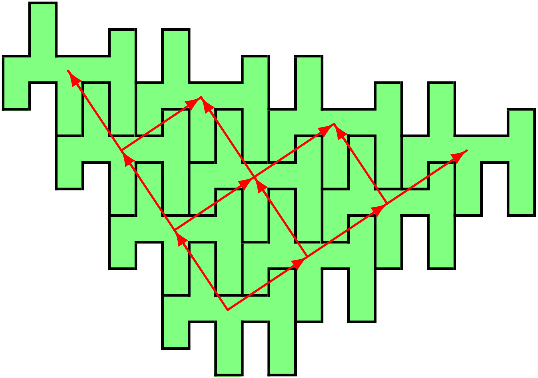
Example



Example



Example

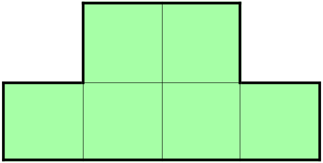


Outline

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

Coding the boundary of a polyomino

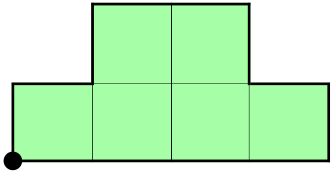
$$\Sigma = \{a, \bar{a}, b, \bar{b}\}$$



Coding the boundary of a polyomino

$$\Sigma = \{a, \bar{a}, b, \bar{b}\}$$

$a \rightarrow$	$b \uparrow$
$\bar{a} \leftarrow$	$\bar{b} \downarrow$

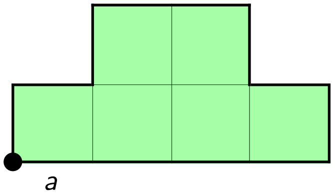


$w =$

Coding the boundary of a polyomino

$$\Sigma = \{a, \overline{a}, b, \overline{b}\}$$

$a \rightarrow$	$b \uparrow$
$\overline{a} \leftarrow$	$\overline{b} \downarrow$

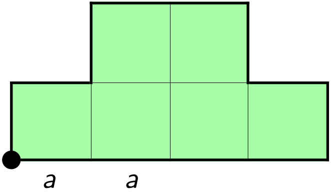


$w = a$

Coding the boundary of a polyomino

$$\Sigma = \{a, \bar{a}, b, \bar{b}\}$$

$a \rightarrow$	$b \uparrow$
$\bar{a} \leftarrow$	$\bar{b} \downarrow$

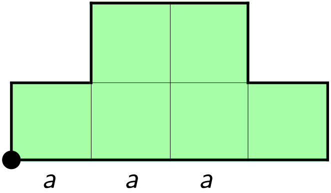


$$w = a\ a$$

Coding the boundary of a polyomino

$$\Sigma = \{a, \overline{a}, b, \overline{b}\}$$

$a \longrightarrow$	$b \uparrow$
$\overline{a} \longleftarrow$	$\overline{b} \downarrow$

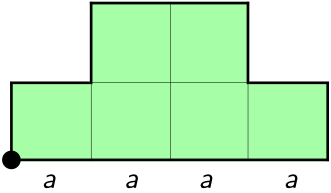


$w = a\ a\ a$

Coding the boundary of a polyomino

$$\Sigma = \{a, \overline{a}, b, \overline{b}\}$$

$a \longrightarrow$	$b \uparrow$
$\overline{a} \longleftarrow$	$\overline{b} \downarrow$

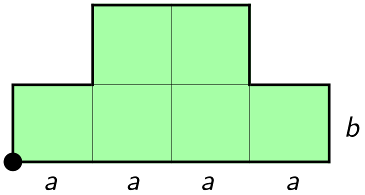


$$w = a\ a\ a\ a$$

Coding the boundary of a polyomino

$$\Sigma = \{a, \overline{a}, b, \overline{b}\}$$

$a \rightarrow$	$b \uparrow$
$\overline{a} \leftarrow$	$\overline{b} \downarrow$

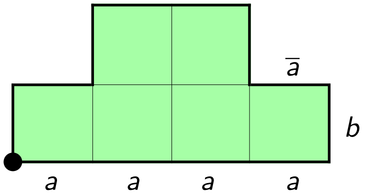


$$w = a\ a\ a\ a\ b$$

Coding the boundary of a polyomino

$$\Sigma = \{a, \overline{a}, b, \overline{b}\}$$

$a \longrightarrow$	$b \uparrow$
$\overline{a} \longleftarrow$	$\overline{b} \downarrow$

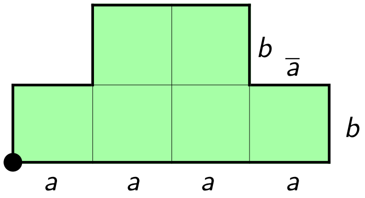


$$w = a\ a\ a\ a\ b\ \overline{a}$$

Coding the boundary of a polyomino

$$\Sigma = \{a, \overline{a}, b, \overline{b}\}$$

$a \longrightarrow$	$b \uparrow$
$\overline{a} \longleftarrow$	$\overline{b} \downarrow$

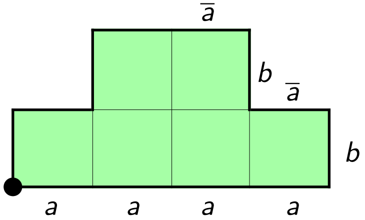


$$w = a\ a\ a\ a\ b\ \overline{a}\ b$$

Coding the boundary of a polyomino

$$\Sigma = \{a, \overline{a}, b, \overline{b}\}$$

$a \rightarrow$	$b \uparrow$
$\overline{a} \leftarrow$	$\overline{b} \downarrow$

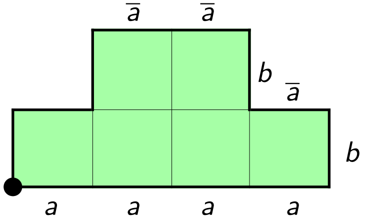


$$w = a\ a\ a\ a\ b\ \overline{a}\ b\ \overline{a}$$

Coding the boundary of a polyomino

$$\Sigma = \{a, \overline{a}, b, \overline{b}\}$$

$a \rightarrow$	$b \uparrow$
$\overline{a} \leftarrow$	$\overline{b} \downarrow$

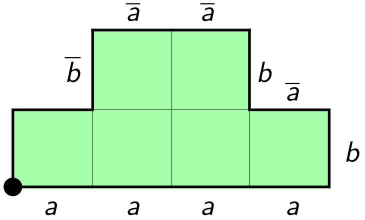


$$w = a\ a\ a\ a\ b\ \overline{a}\ b\ \overline{a}\ \overline{a}$$

Coding the boundary of a polyomino

$$\Sigma = \{a, \overline{a}, b, \overline{b}\}$$

$a \longrightarrow$	$b \uparrow$
$\overline{a} \longleftarrow$	$\overline{b} \downarrow$

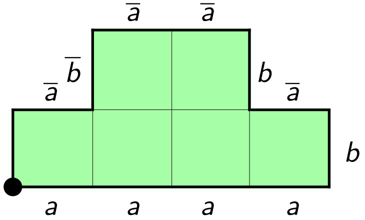


$$w = a\ a\ a\ a\ b\ \overline{a}\ b\ \overline{a}\ \overline{a}\ \overline{b}$$

Coding the boundary of a polyomino

$$\Sigma = \{a, \overline{a}, b, \overline{b}\}$$

$a \rightarrow$	$b \uparrow$
$\overline{a} \leftarrow$	$\overline{b} \downarrow$

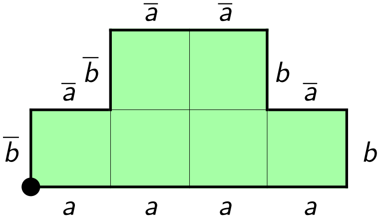


$$w = a\ a\ a\ a\ b\ \overline{a}\ b\ \overline{a}\ \overline{a}\ \overline{b}\ \overline{a}$$

Coding the boundary of a polyomino

$$\Sigma = \{a, \bar{a}, b, \bar{b}\}$$

$a \rightarrow$	$b \uparrow$
$\bar{a} \leftarrow$	$\bar{b} \downarrow$

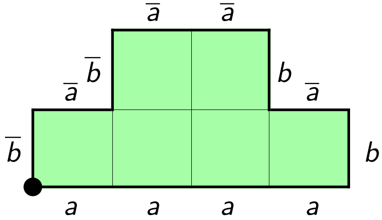


$$w = a\ a\ a\ a\ b\ \bar{a}\ b\ \bar{a}\ \bar{a}\ \bar{b}\ \bar{a}\ \bar{b}.$$

Coding the boundary of a polyomino

$$\Sigma = \{a, \overline{a}, b, \overline{b}\}$$

$a \rightarrow$	$b \uparrow$
$\overline{a} \leftarrow$	$\overline{b} \downarrow$



$$w = a\ a\ a\ a\ b\ \overline{a}\ b\ \overline{a}\ \overline{a}\ \overline{b}\ \overline{a}\ \overline{b}.$$

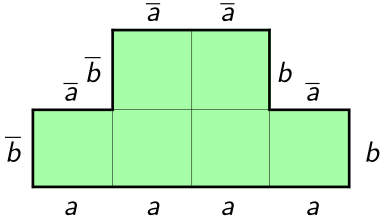
Notation :
$w \equiv w'$ notes that w and w' are conjugate.

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

Coding the boundary of a polyomino

$$\Sigma = \{a, \overline{a}, b, \overline{b}\}$$

$a \rightarrow$	$b \uparrow$
$\overline{a} \leftarrow$	$\overline{b} \downarrow$



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

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

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

Characterization

Definition

Let $\widehat{\ }$ *be the involutive antimorphism defined as* $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

Characterization

Definition

Let $\widehat{\ }$ *be the involutive antimorphism defined as* $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

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

$\widehat{w} = \widehat{uv} = \widehat{v}\widehat{u}$ and $w = \widehat{\widehat{w}}$.

Characterization

Definition

Let $\widehat{\ }$ be the involutive antimorphism defined as $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

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

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

Characterization

Definition

Let $\widehat{\ }$ *be the involutive antimorphism defined as* $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

Let $u, v, w \in \Sigma^* = \left\{a,\ \overline{a},\ b,\ \overline{b}\right\}^*$ such that $w = uv$,

$\widehat{w} = \widehat{uv} = \widehat{v}\,\widehat{u}$ and $w = \widehat{\widehat{w}}$.

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

Characterization

Definition

Let $\widehat{\ }$ be the involutive antimorphism defined as $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

Let $u, v, w \in \Sigma^* = \left\{a,\ \overline{a},\ b,\ \overline{b}\right\}^*$ such that $w = uv$,

$\widehat{w} = \widehat{uv} = \widehat{v}\widehat{u}$ and $w = \widehat{\widehat{w}}$.

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

Characterization

Definition

Let $\widehat{\ }$ be the involutive antimorphism defined as $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

Let $u, v, w \in \Sigma^* = \left\{a, \overline{a}, b, \overline{b}\right\}^*$ such that $w = uv$,

$\widehat{w} = \widehat{uv} = \widehat{v}\widehat{u}$ and $w = \widehat{\widehat{w}}$.

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

Characterization

Definition

Let $\widehat{\ }$ *be the involutive antimorphism defined as* $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

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

$\widehat{w} = \widehat{uv} = \widehat{v}\widehat{u}$ and $w = \widehat{\widehat{w}}$.

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

Characterization

Definition

Let $\widehat{\ }$ be the involutive antimorphism defined as $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

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

$\widehat{w} = \widehat{uv} = \widehat{v}\widehat{u}$ and $w = \widehat{\widehat{w}}$.

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

Characterization

Definition

Let $\widehat{\ }$ *be the involutive antimorphism defined as* $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

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

$\widehat{w} = \widehat{uv} = \widehat{v}\widehat{u}$ and $w = \widehat{\widehat{w}}$.

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

Characterization

Definition

Let $\widehat{\ }$ be the involutive antimorphism defined as $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

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

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

Characterization

Definition

Let $\widehat{\ }$ be the involutive antimorphism defined as $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

Let $u, v, w \in \Sigma^* = \left\{a,\ \overline{a},\ b,\ \overline{b}\right\}^*$ such that $w = uv$,

$\widehat{w} = \widehat{uv} = \widehat{v}\widehat{u}$ and $w = \widehat{\widehat{w}}$.

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

Characterization

Definition

Let $\widehat{\ }$ *be the involutive antimorphism defined as* $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

Let $u, v, w \in \Sigma^* = \left\{a,\ \overline{a},\ b,\ \overline{b}\right\}^*$ such that $w = uv$,

$\widehat{w} = \widehat{uv} = \widehat{v}\,\widehat{u}$ and $w = \widehat{\widehat{w}}$.

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

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

Characterization

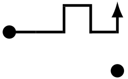
Definition

Let $\widehat{\ }$ *be the involutive antimorphism defined as* $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

Let $u, v, w \in \Sigma^* = \left\{a,\ \overline{a},\ b,\ \overline{b}\right\}^*$ such that $w = uv$,

$\widehat{w} = \widehat{uv} = \widehat{v}\,\widehat{u}$ and $w = \widehat{\widehat{w}}$.

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



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

Characterization

Definition

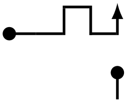
Let $\widehat{\ }$ *be the involutive antimorphism defined as* $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

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

$\widehat{w} = \widehat{uv} = \widehat{v}\,\widehat{u}$ and $w = \widehat{\widehat{w}}$.

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

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



Characterization

Definition

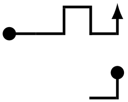
Let $\widehat{\ }$ *be the involutive antimorphism defined as* $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

Let $u, v, w \in \Sigma^* = \left\{a,\ \overline{a},\ b,\ \overline{b}\right\}^*$ such that $w = uv$,

$\widehat{w} = \widehat{uv} = \widehat{v}\,\widehat{u}$ and $w = \widehat{\widehat{w}}$.

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

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



Characterization

Definition

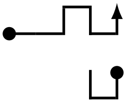
Let $\widehat{\ }$ be the involutive antimorphism defined as $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

Let $u, v, w \in \Sigma^* = \left\{a,\ \overline{a},\ b,\ \overline{b}\right\}^*$ such that $w = uv$,

$\widehat{w} = \widehat{uv} = \widehat{v}\,\widehat{u}$ and $w = \widehat{\widehat{w}}$.

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

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



Characterization

Definition

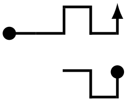
Let $\widehat{\ }$ *be the involutive antimorphism defined as* $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

Let $u, v, w \in \Sigma^* = \left\{a,\ \overline{a},\ b,\ \overline{b}\right\}^*$ such that $w = uv$,

$\widehat{w} = \widehat{uv} = \widehat{v}\widehat{u}$ and $w = \widehat{\widehat{w}}$.

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

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



Characterization

Definition

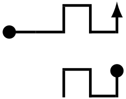
Let $\widehat{\ }$ be the involutive antimorphism defined as $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

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

$\widehat{w} = \widehat{uv} = \widehat{v}\widehat{u}$ and $w = \widehat{\widehat{w}}$.

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

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



Characterization

Definition

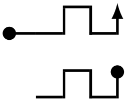
Let $\widehat{\ }$ *be the involutive antimorphism defined as* $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

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

$\widehat{w} = \widehat{uv} = \widehat{v}\widehat{u}$ and $w = \widehat{\widehat{w}}$.

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

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



Characterization

Definition

Let $\widehat{\ }$ *be the involutive antimorphism defined as* $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

Let $u, v, w \in \Sigma^* = \left\{a,\ \overline{a},\ b,\ \overline{b}\right\}^*$ such that $w = uv$,

$\widehat{w} = \widehat{uv} = \widehat{v}\,\widehat{u}$ and $w = \widehat{\widehat{w}}$.

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

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

Characterization

Definition

Let $\widehat{\ }$ be the involutive antimorphism defined as $\widehat{\ } = \overline{\ } \circ \widetilde{\ }$.

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

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

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

Theorem (Beauquier and Nivat, 1991)

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

Neighbouring

Definition

Two polyominoes p and q are simply neighbouring if

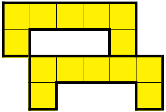
- *They are adjacent.*
- *They don't overlap.*
- *They don't form a hole.*

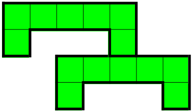
Neighbouring

Definition

Two polyominoes p and q are simply neighbouring if

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





Triad

Definition

Three polyominoes p, q and r form a triad if

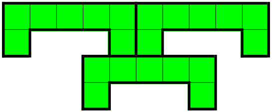
- *They are two by two simply neighbouring.*
- *They don't form a hole.*

Triad

Definition

Three polyominoes p, q and r form a triad if

- *They are two by two simply neighbouring.*
- *They don't form a hole.*



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, the polyominoes p, p_i and p_{i+1} form a triad.

Surrounding

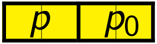
Definition

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

Surrounding

Definition

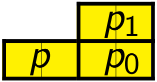
A surrounding of the polyomino p is an ordered sequence of translated copies $(p_0, p_1, \ldots, p_{k-1})$ such that for every i from 0 to k, the polyominoes p, p_i and p_{i+1} form a triad.



Surrounding

Definition

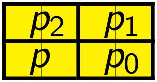
A surrounding of the polyomino p is an ordered sequence of translated copies $(p_0, p_1, \ldots, p_{k-1})$ *such that for every i from* 0 *to* k*, the polyominoes* p*,* p_i *and* p_{i+1} *form a triad.*



Surrounding

Definition

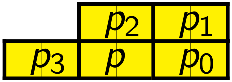
A surrounding of the polyomino p is an ordered sequence of translated copies $(p_0, p_1, \ldots, p_{k-1})$ *such that for every* i *from* 0 *to* k, *the polyominoes* p, p_i *and* p_{i+1} *form a triad.*



Surrounding

Definition

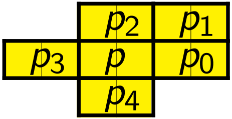
A surrounding of the polyomino p *is an ordered sequence of translated copies* $(p_0, p_1, \ldots, p_{k-1})$ *such that for every* i *from* 0 *to* k*, the polyominoes* p, p_i *and* p_{i+1} *form a triad.*



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, the polyominoes p, p_i and p_{i+1} form a triad.



Surroundings and tilings

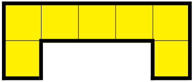
Proposition

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

Surroundings and tilings

Proposition

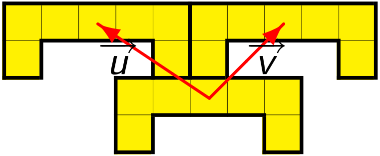
A polyomino p is exact if and only if it admits a surrounding.



Surroundings and tilings

Proposition

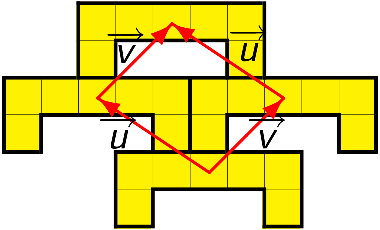
A polyomino p is exact if and only if it admits a surrounding.



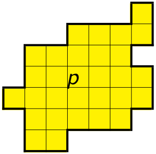
Surroundings and tilings

Proposition

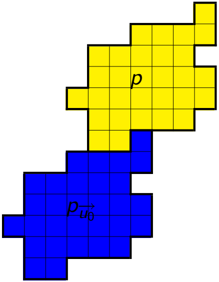
A polyomino p is exact if and only if it admits a surrounding.



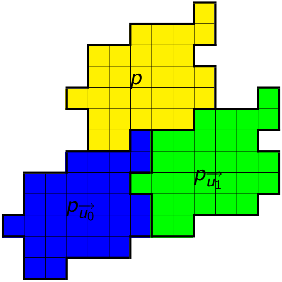
Example



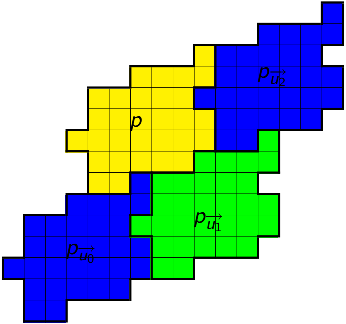
Example



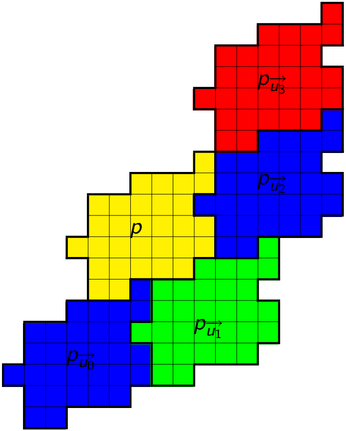
Example



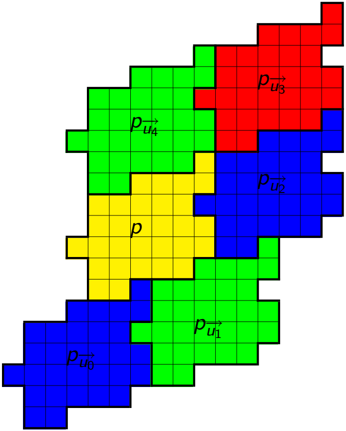
Example


p
$p_{\overrightarrow{u_2}}$
$p_{\overrightarrow{u_1}}$
$p_{\overrightarrow{u_0}}$

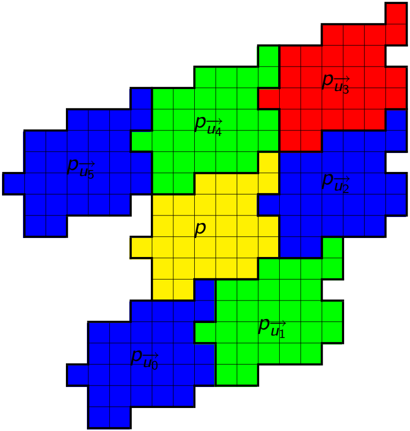
Example



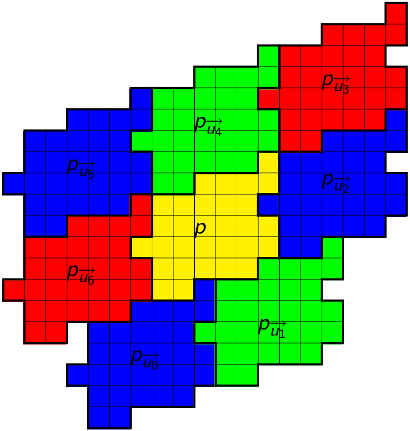
Example



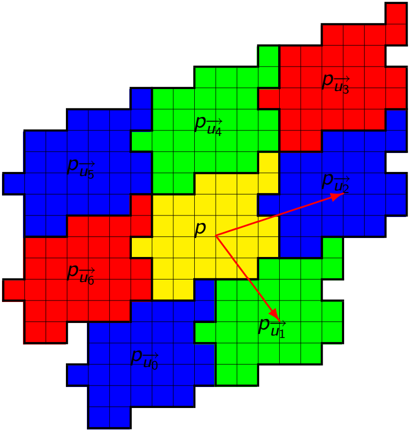
Example


$p_{\overrightarrow{u_3}}$
$p_{\overrightarrow{u_4}}$
$p_{\overrightarrow{u_5}}$
$p_{\overrightarrow{u_2}}$
p
$p_{\overrightarrow{u_1}}$
$p_{\overrightarrow{u_0}}$

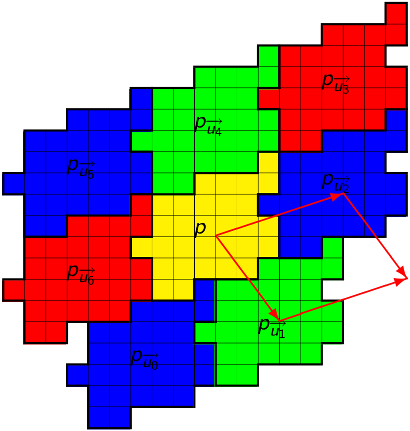
Example


$p_{\overrightarrow{u_3}}$
$p_{\overrightarrow{u_4}}$
$p_{\overrightarrow{u_5}}$
$p_{\overrightarrow{u_2}}$
p
$p_{\overrightarrow{u_6}}$
$p_{\overrightarrow{u_1}}$
$p_{\overrightarrow{u_0}}$

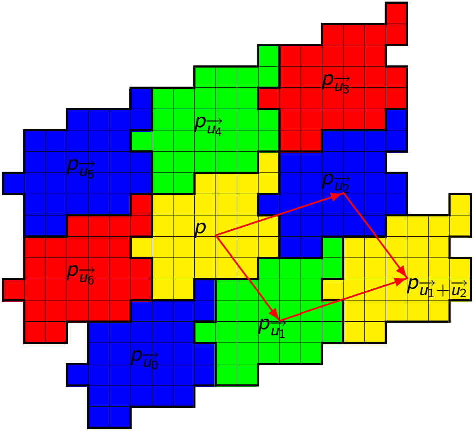
Example



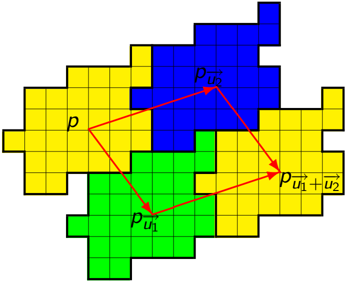
Example


$P_{\overrightarrow{u_3}}$
$P_{\overrightarrow{u_4}}$
$P_{\overrightarrow{u_5}}$
$P_{\overrightarrow{u_2}}$
P
$P_{\overrightarrow{u_6}}$
$P_{\overrightarrow{u_1}}$
$P_{\overrightarrow{u_0}}$

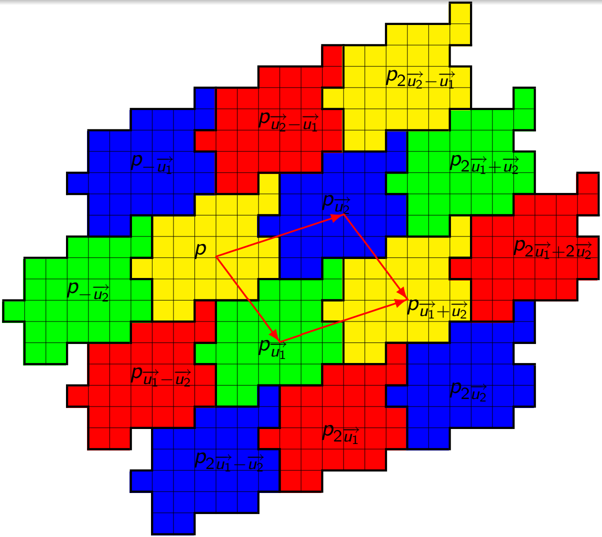
Example


$p_{\vec{u_3}}$
$p_{\vec{u_4}}$
$p_{\vec{u_5}}$
$p_{\vec{u_2}}$
p
$p_{\vec{u_6}}$
$p_{\vec{u_1}+\vec{u_2}}$
$p_{\vec{u_1}}$
$p_{\vec{u_0}}$

Example


p
$p_{\overrightarrow{u_2}}$
$p_{\overrightarrow{u_1}}$
$p_{\overrightarrow{u_1}+\overrightarrow{u_2}}$

Example


$P_{2\vec{u_2}-\vec{u_1}}$
$P_{\vec{u_2}-\vec{u_1}}$
$P_{-\vec{u_1}}$
$P_{2\vec{u_1}+\vec{u_2}}$
$P_{\vec{u_2}}$
P
$P_{2\vec{u_1}+2\vec{u_2}}$
$P_{-\vec{u_2}}$
$P_{\vec{u_1}+\vec{u_2}}$
$P_{\vec{u_1}}$
$P_{\vec{u_1}-\vec{u_2}}$
$P_{2\vec{u_2}}$
$P_{2\vec{u_1}}$
$P_{2\vec{u_1}-\vec{u_2}}$

Surroundings and the factorization

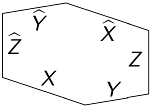
Proposition

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

Surroundings and the factorization

Proposition

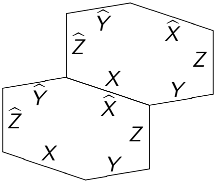
A polyomino p admits a surrounding if and only if its boundary word $w \equiv XYZ\widehat{X}\widehat{Y}\widehat{Z}$ *for some* $X, Y, Z \in \Sigma^*$.



Surroundings and the factorization

Proposition

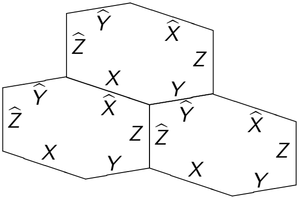
A polyomino p admits a surrounding if and only if its boundary word $w \equiv XYZ\widehat{X}\widehat{Y}\widehat{Z}$ *for some* $X, Y, Z \in \Sigma^*$.



Surroundings and the factorization

Proposition

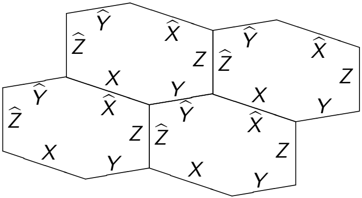
A polyomino p admits a surrounding if and only if its boundary word $w \equiv XYZ\widehat{X}\widehat{Y}\widehat{Z}$ *for some* $X, Y, Z \in \Sigma^*$.



Surroundings and the factorization

Proposition

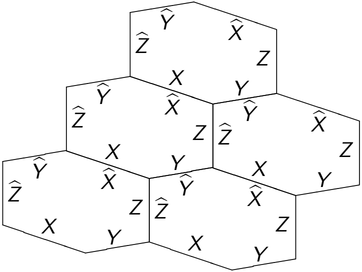
A polyomino p admits a surrounding if and only if its boundary word $w \equiv XYZ\widehat{X}\widehat{Y}\widehat{Z}$ *for some* $X, Y, Z \in \Sigma^*$.



Surroundings and the factorization

Proposition

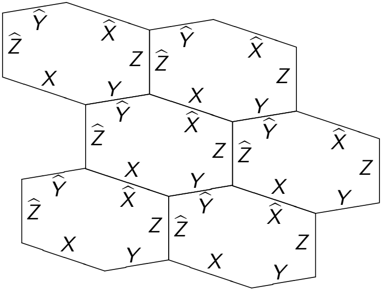
A polyomino p admits a surrounding if and only if its boundary word $w \equiv XYZ\widehat{X}\widehat{Y}\widehat{Z}$ *for some* $X, Y, Z \in \Sigma^*$.



Surroundings and the factorization

Proposition

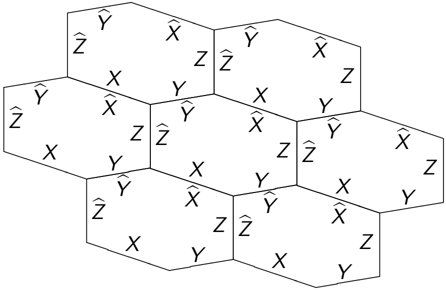
A polyomino p admits a surrounding if and only if its boundary word $w \equiv XYZ\widehat{X}\widehat{Y}\widehat{Z}$ *for some* $X, Y, Z \in \Sigma^*$.



Surroundings and the factorization

Proposition

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



Surroundings and the factorization

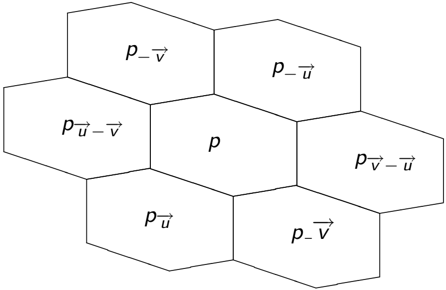
Proposition

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

Surroundings and the factorization

Proposition

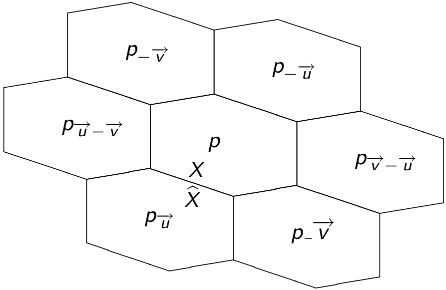
A polyomino p admits a surrounding if and only if its boundary word $w \equiv XYZ\widehat{X}\widehat{Y}\widehat{Z}$ *for some* $X, Y, Z \in \Sigma^*$.



Surroundings and the factorization

Proposition

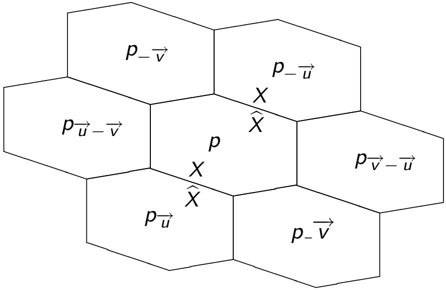
A polyomino p admits a surrounding if and only if its boundary word $w \equiv XYZ\widehat{X}\widehat{Y}\widehat{Z}$ *for some* $X, Y, Z \in \Sigma^*$.



Surroundings and the factorization

Proposition

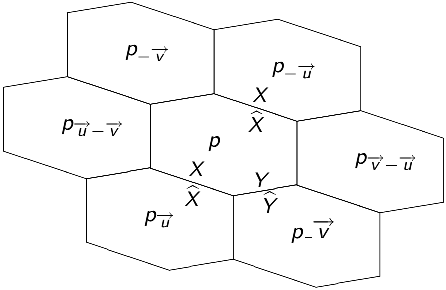
A polyomino p admits a surrounding if and only if its boundary word $w \equiv XYZ\widehat{X}\widehat{Y}\widehat{Z}$ *for some* $X, Y, Z \in \Sigma^*$.



Surroundings and the factorization

Proposition

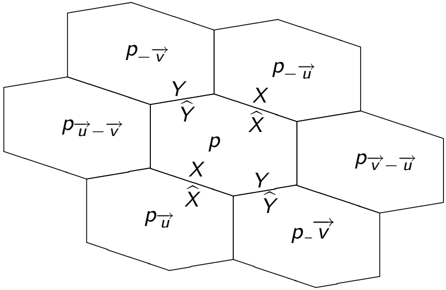
A polyomino p admits a surrounding if and only if its boundary word $w \equiv XYZ\widehat{X}\widehat{Y}\widehat{Z}$ *for some* $X, Y, Z \in \Sigma^*$.



Surroundings and the factorization

Proposition

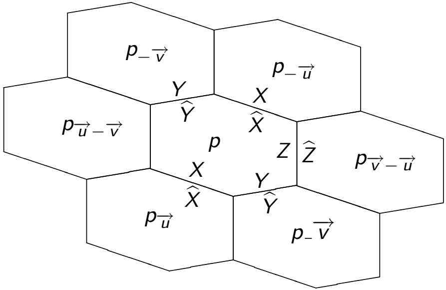
A polyomino p admits a surrounding if and only if its boundary word $w \equiv XYZ\widehat{X}\widehat{Y}\widehat{Z}$ *for some* $X, Y, Z \in \Sigma^*$.



Surroundings and the factorization

Proposition

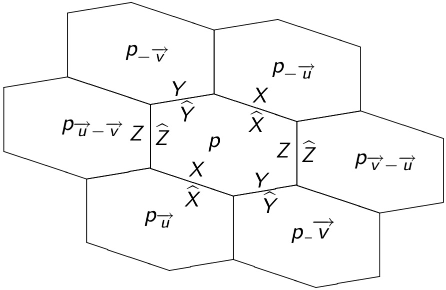
A polyomino p admits a surrounding if and only if its boundary word $w \equiv XYZ\widehat{X}\widehat{Y}\widehat{Z}$ *for some* $X, Y, Z \in \Sigma^*$.



Surroundings and the factorization

Proposition

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



Pseudo-square and pseudo-hexagons

Definition

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

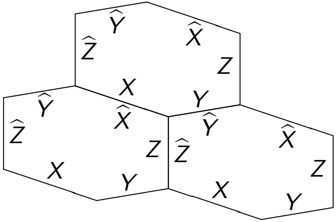
Pseudo-square and pseudo-hexagons

Definition

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

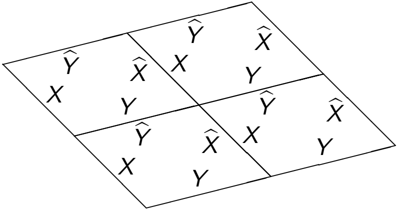
Pseudo-hexagon

$w \equiv XYZ\widehat{X}\widehat{Y}\widehat{Z}$.



Pseudo-square

$w \equiv XY\widehat{X}\widehat{Y}$.



Complexity

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 $\mathcal{O}(n^4)$.

Complexity

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 $\mathcal{O}(n^4)$.

Remark

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

Complexity

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 $\mathcal{O}(n^4)$.

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 $\mathcal{O}(n^2)$.

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

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

Admissible factors

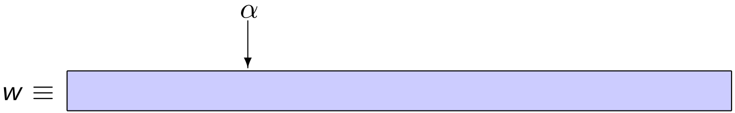
Proposition

Let $\mathcal{A}$ be the set of all admissible factors overlapping a position α in w and $\widehat{\mathcal{A}}$ 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 $\mathcal{A} \cup \widehat{\mathcal{A}}$.

Admissible factors

Proposition

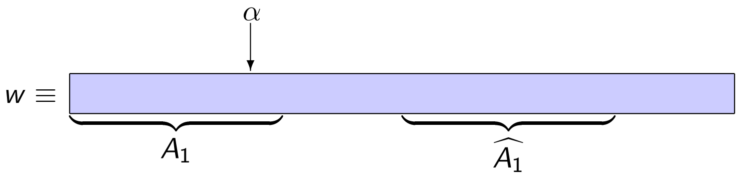
Let $\mathcal{A}$ be the set of all admissible factors overlapping a position α in w and $\widehat{\mathcal{A}}$ be the set of their respective homologous factors. Then, there is at least one position in w that is not covered by any element of $\mathcal{A} \cup \widehat{\mathcal{A}}$.



Admissible factors

Proposition

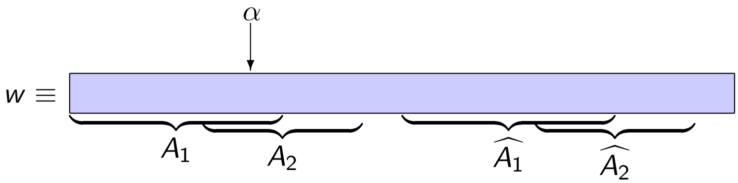
Let $\mathcal{A}$ be the set of all admissible factors overlapping a position α in w and $\widehat{\mathcal{A}}$ be the set of their respective homologous factors. Then, there is at least one position in w that is not covered by any element of $\mathcal{A} \cup \widehat{\mathcal{A}}$.



Admissible factors

Proposition

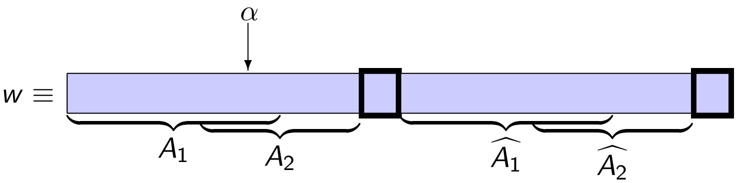
Let $\mathcal{A}$ be the set of all admissible factors overlapping a position α in w and $\widehat{\mathcal{A}}$ be the set of their respective homologous factors. Then, there is at least one position in w that is not covered by any element of $\mathcal{A} \cup \widehat{\mathcal{A}}$.



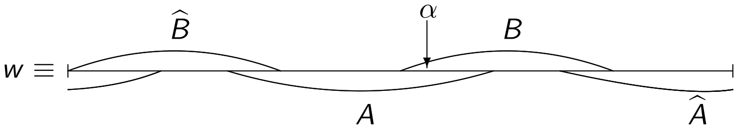
Admissible factors

Proposition

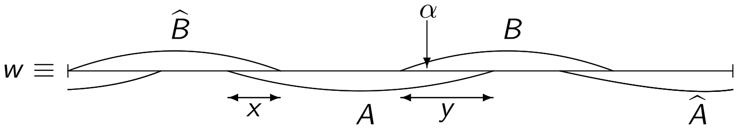
Let $\mathcal{A}$ be the set of all admissible factors overlapping a position α in w and $\widehat{\mathcal{A}}$ be the set of their respective homologous factors. Then, there is at least one position in w that is not covered by any element of $\mathcal{A} \cup \widehat{\mathcal{A}}$.



Admissible factors

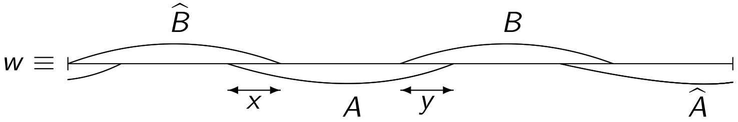


Admissible factors



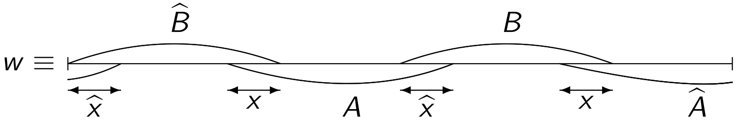
Admissible factors

1. $|x| = |y|$



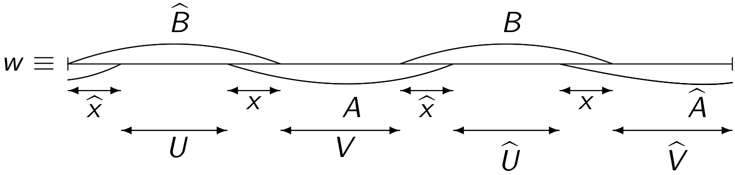
Admissible factors

1. $|x| = |y|$



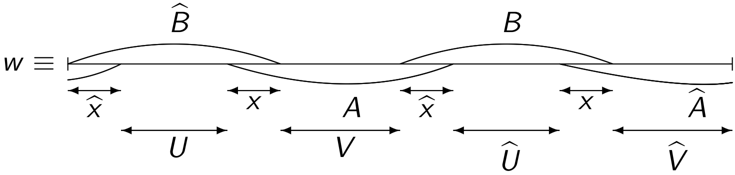
Admissible factors

1. $|x| = |y|$



Admissible factors

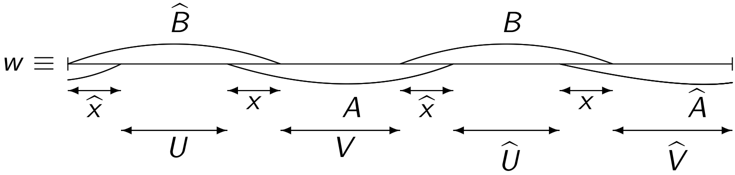
1. $|x| = |y|$



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

Admissible factors

1. $|x| = |y|$



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

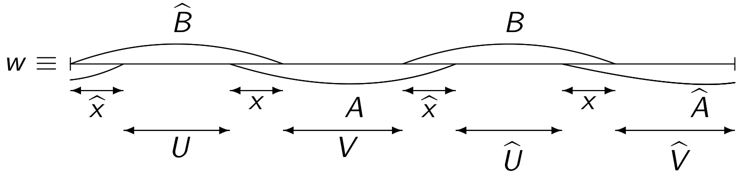
Lemma (Dorat and Nivat, 2003) (Brlek, Labelle and Lacasse, 2005)

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

$$\#(\textit{left turns}) - \#(\textit{right turns}) = 4.$$

Admissible factors

1. $|x| = |y|$



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

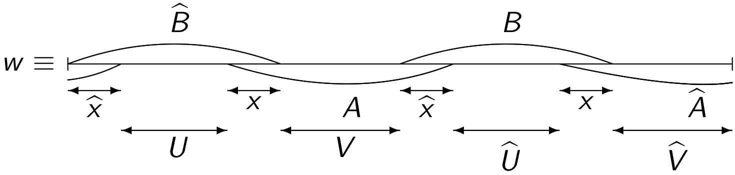
Lemma (Dorat and Nivat, 2003) (Brlek, Labelle and Lacasse, 2005)

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

$$\#(\mathit{left\ turns}) - \#(\mathit{right\ turns}) = 4.$$

Admissible factors

1. $|x| = |y|$



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

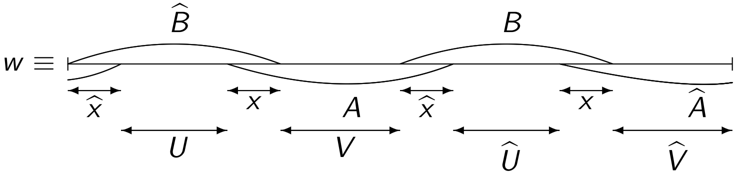
Lemma (Dorat and Nivat, 2003) (Brlek, Labelle and Lacasse, 2005)

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

$$\#(\textit{left turns}) - \#(\textit{right turns}) = 4.$$

Admissible factors

1. $|x| = |y|$



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

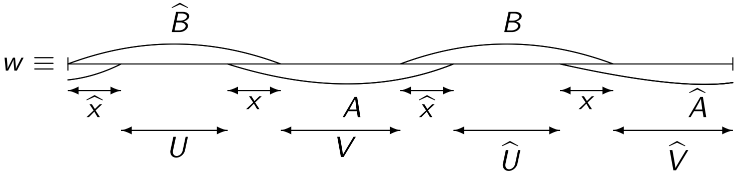
Lemma (Dorat and Nivat, 2003) (Brlek, Labelle and Lacasse, 2005)

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

$$\#(\mathit{left\ turns}) - \#(\mathit{right\ turns}) = 4.$$

Admissible factors

1. $|x| = |y|$



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

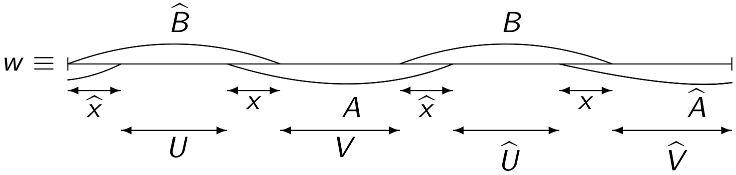
Lemma (Dorat and Nivat, 2003) (Brlek, Labelle and Lacasse, 2005)

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

$$\#(\textit{left turns}) - \#(\textit{right turns}) = 4.$$

Admissible factors

1. $|x| = |y|$



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

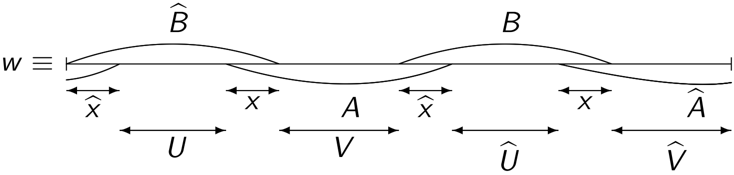
Lemma (Dorat and Nivat, 2003) (Brlek, Labelle and Lacasse, 2005)

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

$$\#(\textit{left turns}) - \#(\textit{right turns}) = 4.$$

Admissible factors

1. $|x| = |y|$



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

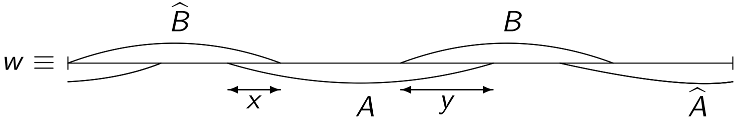
Lemma (Dorat and Nivat, 2003) (Brlek, Labelle and Lacasse, 2005)

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

$$\#(\textit{left turns}) - \#(\textit{right turns}) = 4.$$

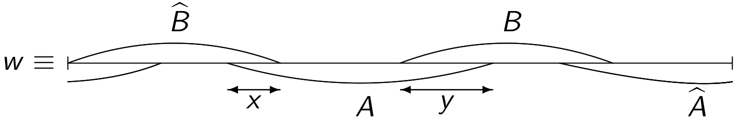
Admissible factors

2. $|x| \neq |y|$.



Admissible factors

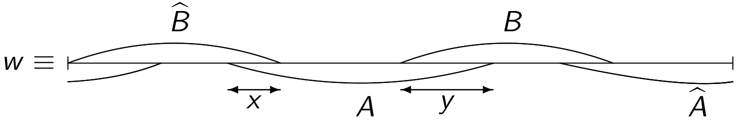
2. $|x| \neq |y|$.



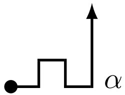
$w \equiv \ \alpha \ \beta \ \gamma$, where $\overrightarrow{\beta} = \overrightarrow{0}$.

Admissible factors

2. $|x| \neq |y|$.

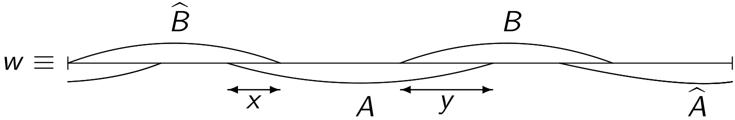


$w \equiv \alpha \ \beta \ \gamma$, where $\overrightarrow{\beta} = \overrightarrow{0}$.

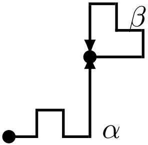


Admissible factors

2. $|x| \neq |y|$.

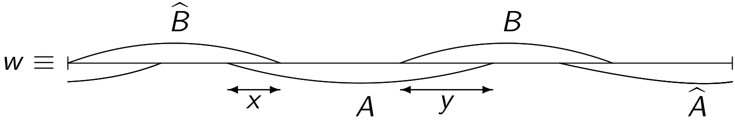


$w \equiv \; \alpha \; \beta \; \gamma$, where $\overrightarrow{\beta} = \overrightarrow{0}$.

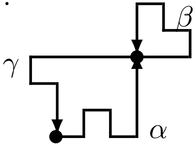


Admissible factors

2. $|x| \neq |y|$.

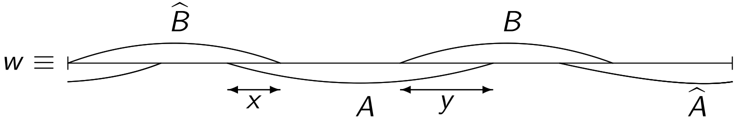


$w \equiv \alpha \ \beta \ \gamma$, where $\overrightarrow{\beta} = \overrightarrow{0}$.

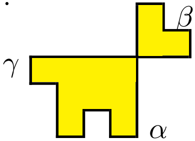


Admissible factors

2. $|x| \neq |y|$.

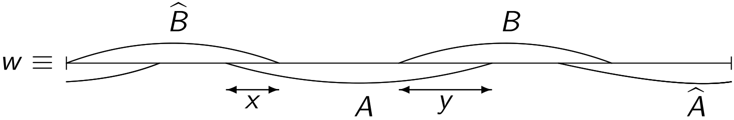


$w \equiv \ \alpha \ \beta \ \gamma$, where $\overrightarrow{\beta} = \overrightarrow{0}$.

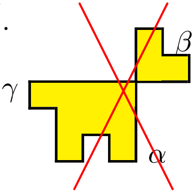


Admissible factors

2. $|x| \neq |y|$.



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



Admissible factors

Lemma

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

Admissible factors

Lemma

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

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

Admissible factors

Lemma

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

- $w \equiv Ax\widehat{A}y$, for x, y such that $|x| = |y|$.
  Direct consequence of the fact that $|u| = |\widehat{u}|$ for all $u \in \Sigma^*$.
- A is *maximal*, that is, $\mathrm{first}(x) \neq \overline{\mathrm{last}(x)}$ and $\mathrm{first}(y) \neq \overline{\mathrm{last}(y)}$.

Admissible factors

Lemma

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

- $w \equiv Ax\widehat{A}y$, for x, y such that $|x| = |y|$.
  Direct consequence of the fact that $|u| = |\widehat{u}|$ for all $u \in \Sigma^*$.
- A is *maximal*, that is, $\mathrm{first}(x) \neq \overline{\mathrm{last}(x)}$ and $\mathrm{first}(y) \neq \overline{\mathrm{last}(y)}$.

  By contradiction, assume that X is not maximal, then $\mathrm{first}(YZ) = \overline{\mathrm{last}(YZ)}$.

Admissible factors

Lemma

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

- $w \equiv Ax\widehat{A}y$, for x, y such that $|x| = |y|$.
  Direct consequence of the fact that $|u| = |\widehat{u}|$ for all $u \in \Sigma^*$.
- A is *maximal*, that is, $\mathrm{first}(x) \neq \overline{\mathrm{last}(x)}$ and $\mathrm{first}(y) \neq \overline{\mathrm{last}(y)}$.

  By contradiction, assume that X is not maximal, then $\mathrm{first}(YZ) = \overline{\mathrm{last}(YZ)}$.

  $YZ = \alpha Y' Z' \overline{\alpha}$

Admissible factors

Lemma

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

- $w \equiv Ax\widehat{A}y$, for x, y such that $|x| = |y|$.
  Direct consequence of the fact that $|u| = |\widehat{u}|$ for all $u \in \Sigma^*$.
- A is *maximal*, that is, $\mathrm{first}(x) \neq \overline{\mathrm{last}(x)}$ and $\mathrm{first}(y) \neq \overline{\mathrm{last}(y)}$.

  By contradiction, assume that X is not maximal, then $\mathrm{first}(YZ) = \overline{\mathrm{last}(YZ)}$.

  $$YZ = \alpha Y'Z'\overline{\alpha} \implies \widehat{Y}\widehat{Z} = \widehat{\alpha Y'}\widehat{Z'\overline{\alpha}} = \widehat{Y'}\overline{\alpha}\alpha\widehat{Z'}.$$

Admissible factors

Lemma

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

- $w \equiv Ax\widehat{A}y$, for x, y such that $|x| = |y|$.
  Direct consequence of the fact that $|u| = |\widehat{u}|$ for all $u \in \Sigma^*$.
- A is *maximal*, that is, $\mathrm{first}(x) \neq \overline{\mathrm{last}(x)}$ and $\mathrm{first}(y) \neq \overline{\mathrm{last}(y)}$.

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

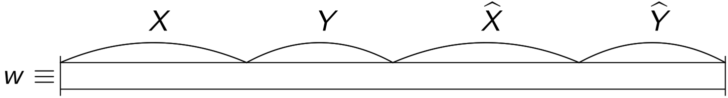
Admissible factors

Lemma

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

- $w \equiv Ax\widehat{A}y$, for x, y such that $|x| = |y|$.
  Direct consequence of the fact that $|u| = |\widehat{u}|$ for all $u \in \Sigma^*$.
- A is *maximal*, that is, $\mathrm{first}(x) \neq \overline{\mathrm{last}(x)}$ and $\mathrm{first}(y) \neq \overline{\mathrm{last}(y)}$.

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



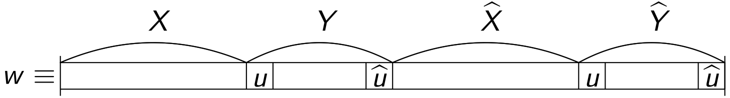
Admissible factors

Lemma

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

- $w \equiv Ax\widehat{A}y$, for x, y such that $|x| = |y|$.
  Direct consequence of the fact that $|u| = |\widehat{u}|$ for all $u \in \Sigma^*$.
- A is *maximal*, that is, $\mathrm{first}(x) \neq \overline{\mathrm{last}(x)}$ and $\mathrm{first}(y) \neq \overline{\mathrm{last}(y)}$.

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



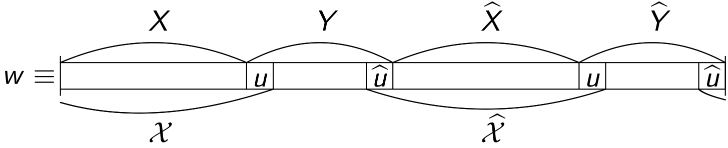
Admissible factors

Lemma

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

- $w \equiv Ax\widehat{A}y$, for x, y such that $|x| = |y|$.
  Direct consequence of the fact that $|u| = |\widehat{u}|$ for all $u \in \Sigma^*$.
- A is *maximal*, that is, $\mathrm{first}(x) \neq \overline{\mathrm{last}(x)}$ and $\mathrm{first}(y) \neq \overline{\mathrm{last}(y)}$.

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



Admissible factors

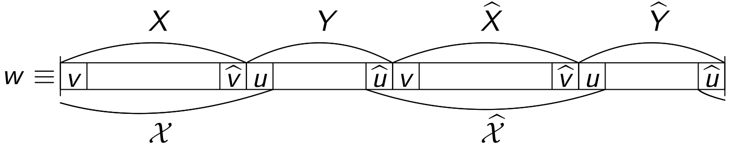
Lemma

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

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

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

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



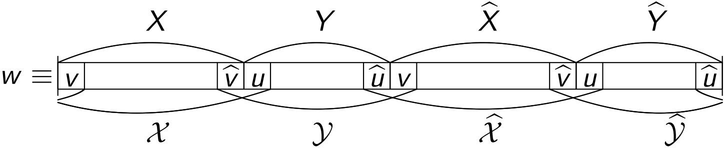
Admissible factors

Lemma

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

- $w \equiv Ax\widehat{A}y$, for x, y such that $|x| = |y|$.
  Direct consequence of the fact that $|u| = |\widehat{u}|$ for all $u \in \Sigma^*$.
- A is *maximal*, that is, $\mathrm{first}(x) \neq \overline{\mathrm{last}(x)}$ and $\mathrm{first}(y) \neq \overline{\mathrm{last}(y)}$.

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



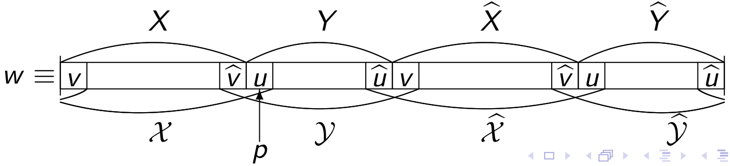
Admissible factors

Lemma

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

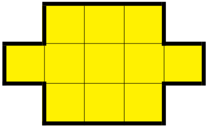
- $w \equiv Ax\widehat{A}y$, for x, y such that $|x| = |y|$.
  Direct consequence of the fact that $|u| = |\widehat{u}|$ for all $u \in \Sigma^*$.
- A is *maximal*, that is, $\mathrm{first}(x) \neq \overline{\mathrm{last}(x)}$ and $\mathrm{first}(y) \neq \overline{\mathrm{last}(y)}$.

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



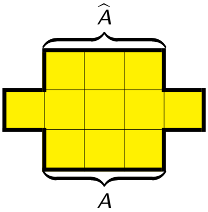
Admissible factors

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



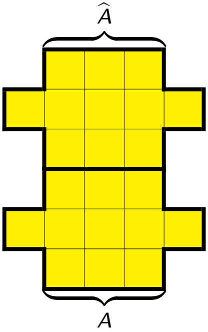
Admissible factors

$$w \equiv \underbrace{a\,a\,a}_{A}\,\underbrace{b\,a\,b\,\overline{a}\,b}_{x}\,\underbrace{\overline{a}\,\overline{a}\,\overline{a}}_{\widehat{A}}\,\underbrace{\overline{b}\,\overline{a}\,\overline{b}\,a\,\overline{b}}_{y}$$



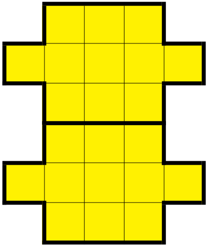
Admissible factors

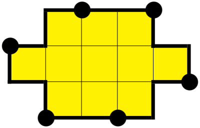
$$w \equiv \underbrace{a\,a\,a}_{A}\,\underbrace{b\,a\,b\,\bar{a}\,b}_{x}\,\underbrace{\bar{a}\,\bar{a}\,\bar{a}}_{\widehat{A}}\,\underbrace{\bar{b}\,\bar{a}\,\bar{b}\,a\,\bar{b}}_{y}$$



Admissible factors

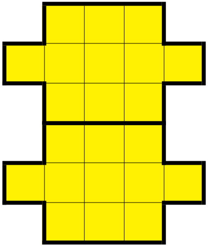
$$w \equiv \underbrace{a\ a}_{X}\ \underbrace{a\ b\ a}_{Y}\ \underbrace{b\ \overline{a}\ b}_{Z}\ \underbrace{\overline{a}\ \overline{a}}_{\widehat{X}}\ \underbrace{\overline{a}\ \overline{b}\ \overline{a}}_{\widehat{Y}}\ \underbrace{\overline{b}\ a\ \overline{b}}_{\widehat{Z}}$$

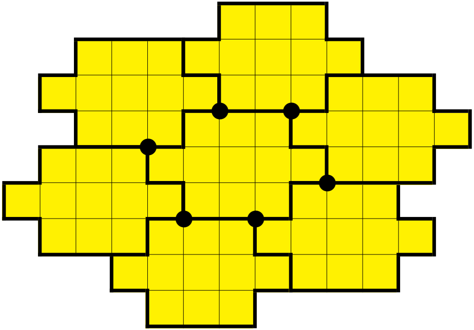




Admissible factors

$$w \equiv \underbrace{a\,a}_{X}\,\underbrace{a\,b\,a}_{Y}\,\underbrace{b\,\overline{a}\,b}_{Z}\,\underbrace{\overline{a}\,\overline{a}}_{\widehat{X}}\,\underbrace{\overline{a}\,\overline{b}\,\overline{a}}_{\widehat{Y}}\,\underbrace{\overline{b}\,a\,\overline{b}}_{\widehat{Z}}$$





Listing admissible factors

Lemma

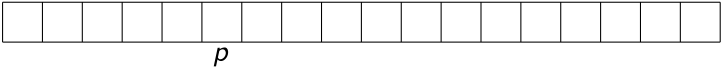
Given a position p in the word w coding a polyomino, all the admissible factors overlapping p can be listed in linear time.

Listing admissible factors

Lemma

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

If $w \equiv A\, x\, \widehat{A}\, y$ then $\widehat{w} \equiv \widehat{y}\, A\, \widehat{x}\, \widehat{A}$.

$w \equiv$ 

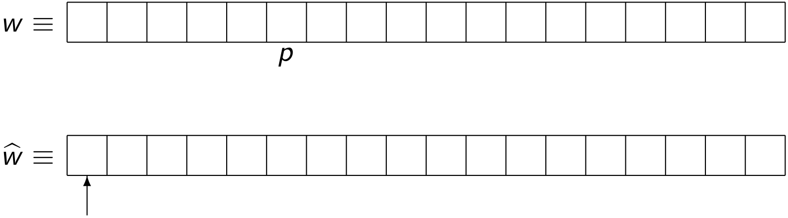
$\widehat{w} \equiv$ 

Listing admissible factors

Lemma

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

If $w \equiv A\, x\, \widehat{A}\, y$ then $\widehat{w} \equiv \widehat{y}\, A\, \widehat{x}\, \widehat{A}$.

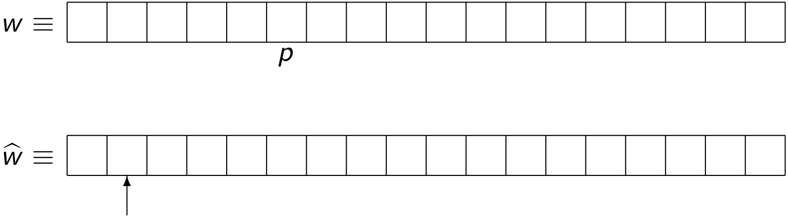


Listing admissible factors

Lemma

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

If $w \equiv A\,x\,\widehat{A}\,y$ then $\widehat{w} \equiv \widehat{y}\,A\,\widehat{x}\,\widehat{A}$.

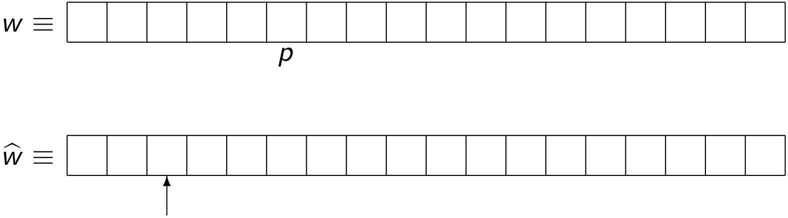


Listing admissible factors

Lemma

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

If $w \equiv A\,x\,\widehat{A}\,y$ then $\widehat{w} \equiv \widehat{y}\,A\,\widehat{x}\,\widehat{A}$.

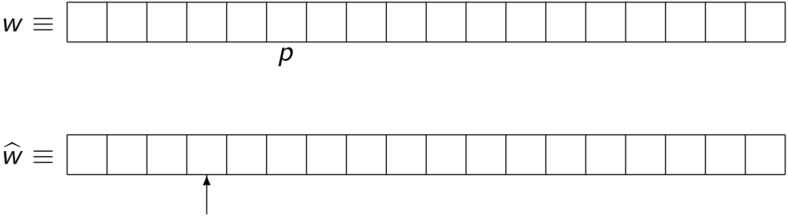


Listing admissible factors

Lemma

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

If $w \equiv A\, x\, \widehat{A}\, y$ then $\widehat{w} \equiv \widehat{y}\, A\, \widehat{x}\, \widehat{A}$.

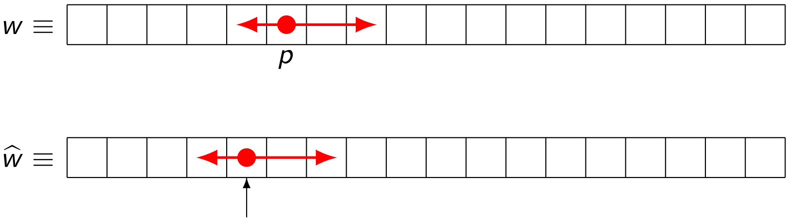


Listing admissible factors

Lemma

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

If $w \equiv A\, x\, \widehat{A}\, y$ then $\widehat{w} \equiv \widehat{y}\, A\, \widehat{x}\, \widehat{A}$.

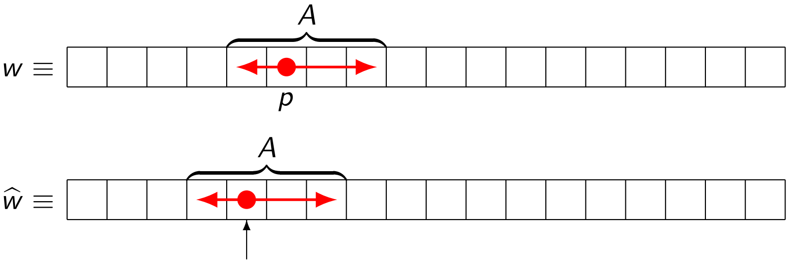


Listing admissible factors

Lemma

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

If $w \equiv A\,x\,\widehat{A}\,y$ then $\widehat{w} \equiv \widehat{y}\,A\,\widehat{x}\,\widehat{A}$.

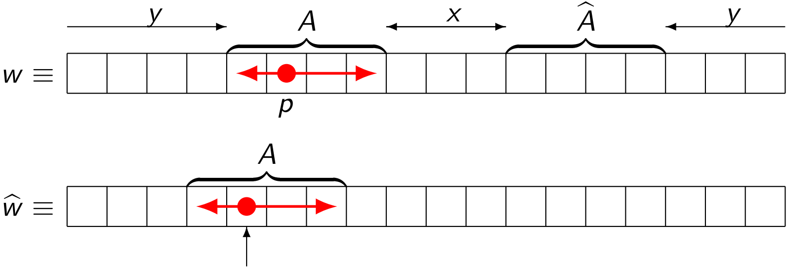


Listing admissible factors

Lemma

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

If $w \equiv A\, x\, \widehat{A}\, y$ then $\widehat{w} \equiv \widehat{y}\, A\, \widehat{x}\, \widehat{A}$.

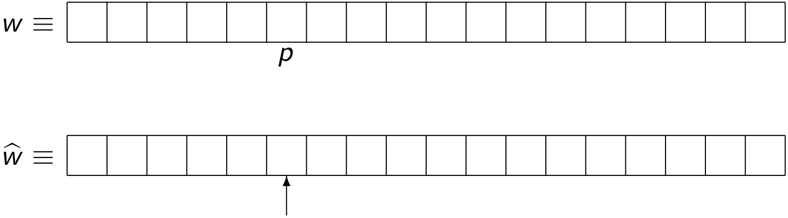


Listing admissible factors

Lemma

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

If $w \equiv A\,x\,\widehat{A}\,y$ then $\widehat{w} \equiv \widehat{y}\,A\,\widehat{x}\,\widehat{A}$.

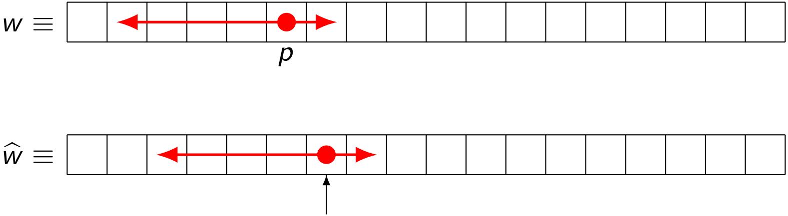


Listing admissible factors

Lemma

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

If $w \equiv A\, x\, \widehat{A}\, y$ then $\widehat{w} \equiv \widehat{y}\, A\, \widehat{x}\, \widehat{A}$.

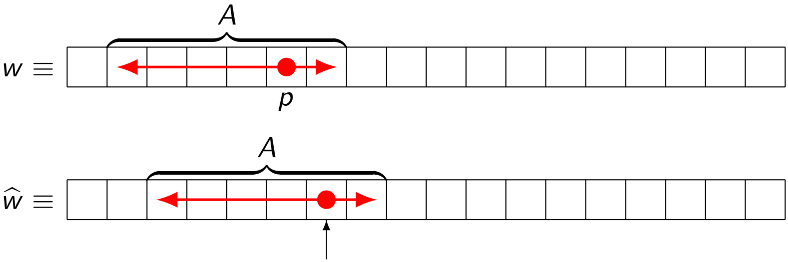


Listing admissible factors

Lemma

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

If $w \equiv A\,x\,\widehat{A}\,y$ then $\widehat{w} \equiv \widehat{y}\,A\,\widehat{x}\,\widehat{A}$.

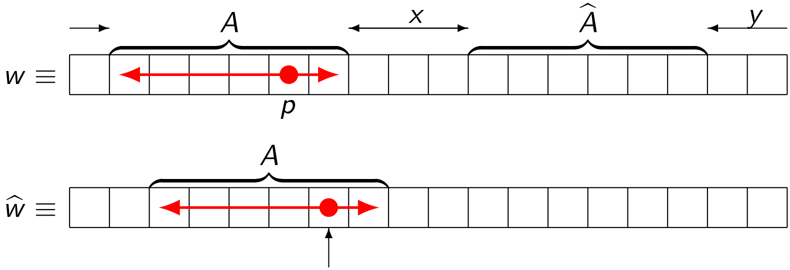


Listing admissible factors

Lemma

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

If $w \equiv A\, x\, \widehat{A}\, y$ then $\widehat{w} \equiv \widehat{y}\, A\, \widehat{x}\, \widehat{A}$.

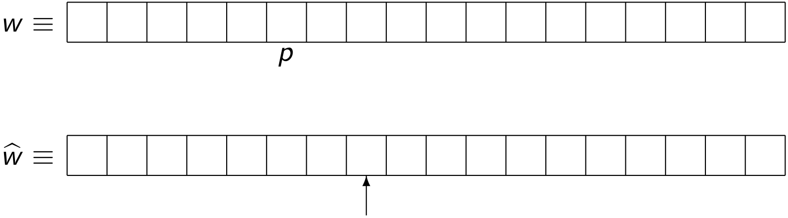


Listing admissible factors

Lemma

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

If $w \equiv A\, x\, \widehat{A}\, y$ then $\widehat{w} \equiv \widehat{y}\, A\, \widehat{x}\, \widehat{A}$.

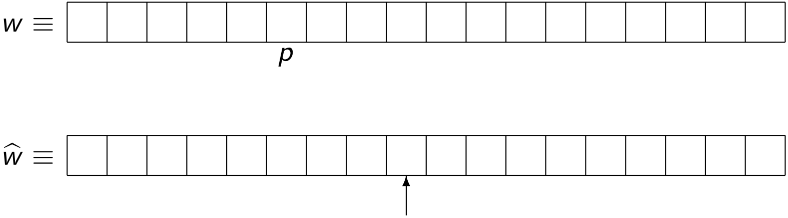


Listing admissible factors

Lemma

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

If $w \equiv A\,x\,\widehat{A}\,y$ then $\widehat{w} \equiv \widehat{y}\,A\,\widehat{x}\,\widehat{A}$.

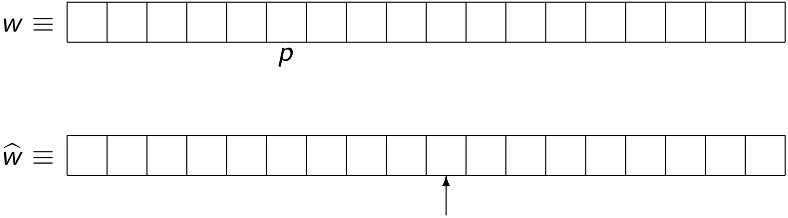


Listing admissible factors

Lemma

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

If $w \equiv A\, x\, \widehat{A}\, y$ then $\widehat{w} \equiv \widehat{y}\, A\, \widehat{x}\, \widehat{A}$.

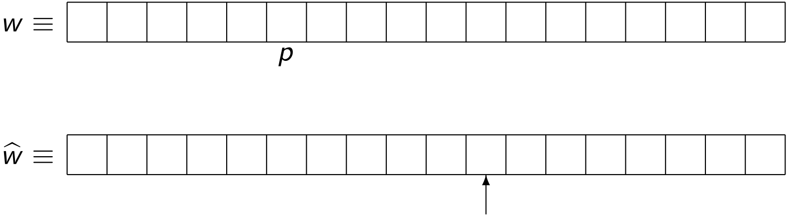


Listing admissible factors

Lemma

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

If $w \equiv A\,x\,\widehat{A}\,y$ then $\widehat{w} \equiv \widehat{y}\,A\,\widehat{x}\,\widehat{A}$.

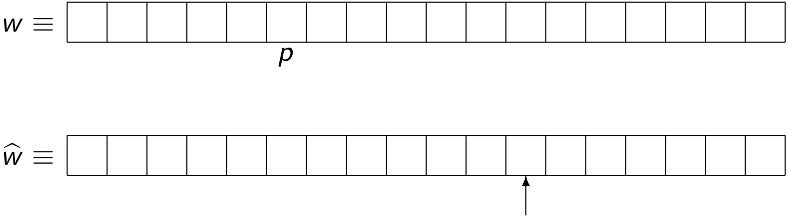


Listing admissible factors

Lemma

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

If $w \equiv A\, x\, \widehat{A}\, y$ then $\widehat{w} \equiv \widehat{y}\, A\, \widehat{x}\, \widehat{A}$.

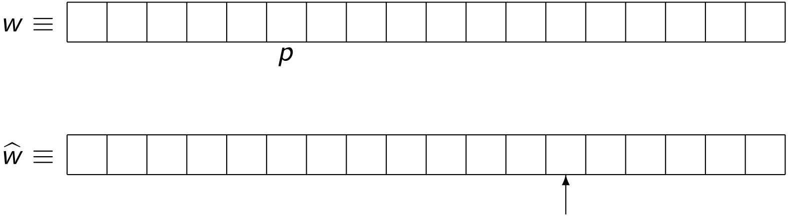


Listing admissible factors

Lemma

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

If $w \equiv A\, x\, \widehat{A}\, y$ then $\widehat{w} \equiv \widehat{y}\, A\, \widehat{x}\, \widehat{A}$.

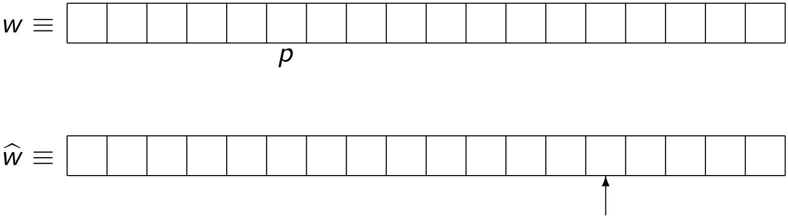


Listing admissible factors

Lemma

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

If $w \equiv A\,x\,\widehat{A}\,y$ then $\widehat{w} \equiv \widehat{y}\,A\,\widehat{x}\,\widehat{A}$.

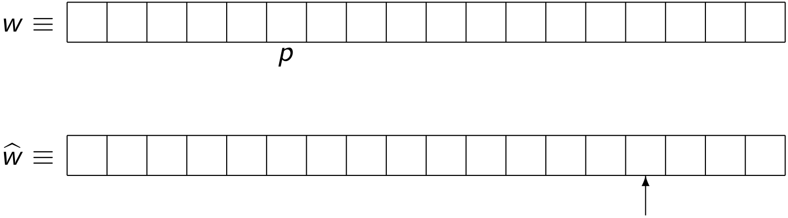


Listing admissible factors

Lemma

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

If $w \equiv A\, x\, \widehat{A}\, y$ then $\widehat{w} \equiv \widehat{y}\, A\, \widehat{x}\, \widehat{A}$.

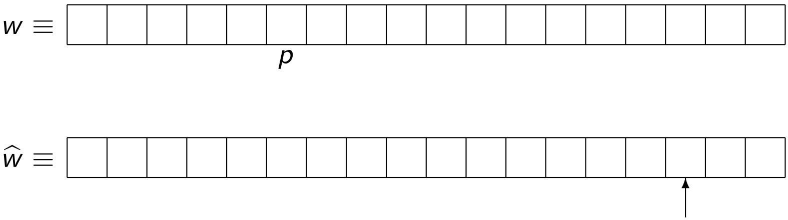


Listing admissible factors

Lemma

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

If $w \equiv A\, x\, \widehat{A}\, y$ then $\widehat{w} \equiv \widehat{y}\, A\, \widehat{x}\, \widehat{A}$.

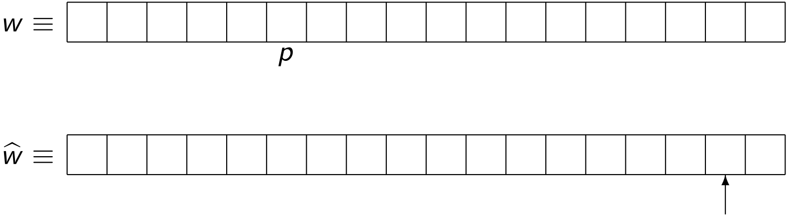


Listing admissible factors

Lemma

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

If $w \equiv A\,x\,\widehat{A}\,y$ then $\widehat{w} \equiv \widehat{y}\,A\,\widehat{x}\,\widehat{A}$.

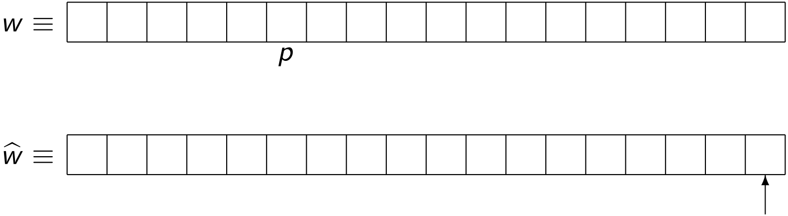


Listing admissible factors

Lemma

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

If $w \equiv A\, x\, \widehat{A}\, y$ then $\widehat{w} \equiv \widehat{y}\, A\, \widehat{x}\, \widehat{A}$.



Detecting pseudo-squares

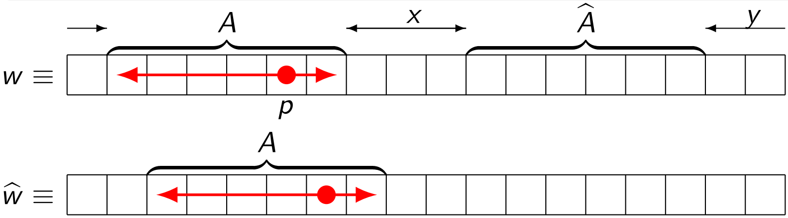
Theorem

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

Detecting pseudo-squares

Theorem

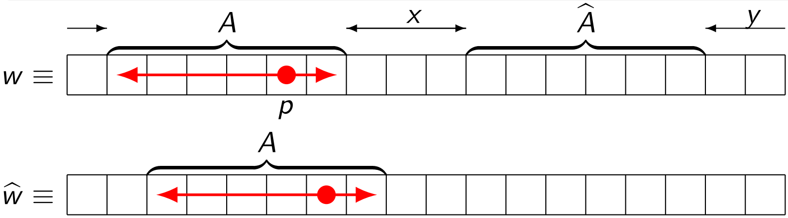
Let w be the boundary of p. Determining if w codes a pseudo-square is decidable in linear time.



Detecting pseudo-squares

Theorem

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

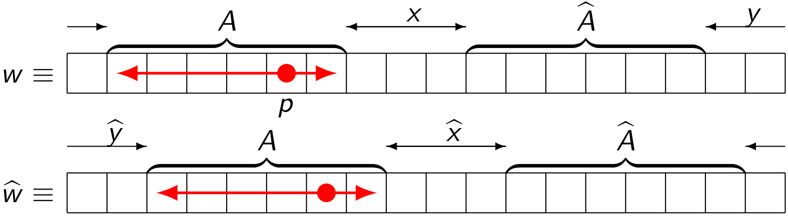


If $x = \widehat{y}$ then $w \equiv XY\widehat{X}\widehat{Y}$.

Detecting pseudo-squares

Theorem

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

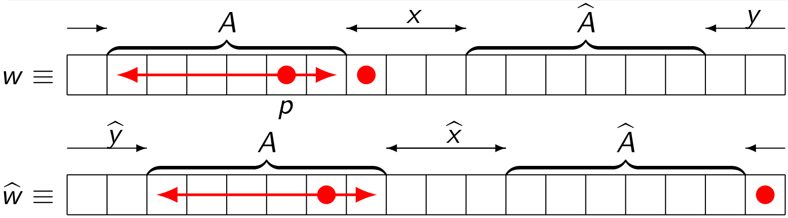


If $x = \widehat{y}$ then $w \equiv XY\widehat{X}\widehat{Y}$.
Since $w \equiv A\,x\,\widehat{A}\,y$ then $\widehat{w} \equiv \widehat{y}\,A\,\widehat{x}\,\widehat{A}$.

Detecting pseudo-squares

Theorem

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



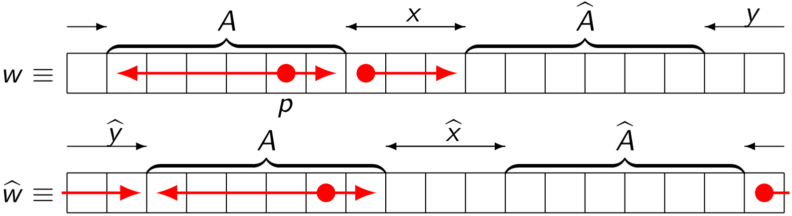
If $x = \widehat{y}$ then $w \equiv XY\widehat{X}\widehat{Y}$.

Since $w \equiv A\,x\,\widehat{A}\,y$ then $\widehat{w} \equiv \widehat{y}\,A\,\widehat{x}\,\widehat{A}$.

Detecting pseudo-squares

Theorem

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



If $x = \widehat{y}$ then $w \equiv XY\widehat{X}\widehat{Y}$.
Since $w \equiv A\,x\,\widehat{A}\,y$ then $\widehat{w} \equiv \widehat{y}\,A\,\widehat{x}\,\widehat{A}$.

k-square-free words

Definition

A word w is k-square-free if

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

k-square-free words

Definition

A word w is k-square-free if

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

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

k-square-free words

Definition

A word w is k-square-free if

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

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

Lemma

Let w be a k-square-free word coding a polyomino, and let α be a position in w. the number of admissible factors overlapping α 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 $k \in \mathcal{O}(\sqrt{n})$. Determining if w codes a pseudo-hexagon is decidable in linear time.

Detecting pseudo-hexagons

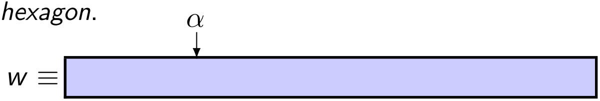
Input : $w \in \Sigma^*$ coding a polyomino p.

Build L_1 the list of all admissible factors that overlap the position α.
$\beta :=$ (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 β.
For all $X \in L_1$ **do**
  For all $Y \in L_2$ **do**
    If $w \equiv XYx\widehat{X}\widehat{Y}y$ **then**
      Compute i : the position of x in w.
      Compute j : the position of $\widehat{y}$ in $\widehat{w}$.
      If longest common extention$(w, \widehat{w}, i, j) = |x|$ **then**
        p is a *speudo-hexagon*.
      End if
    End if
  End for
End for

Detecting pseudo-hexagons

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

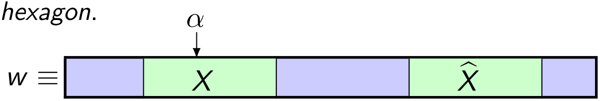
Build L_1 the list of all admissible factors that overlap the position α.
$\beta :=$ (the position of the rightmost letter of w include in a factor of L_1) $+ 1$.
Build L_2 the list of all admissible factors that overlap the position β.
For all $X \in L_1$ **do**
  For all $Y \in L_2$ **do**
    If $w \equiv XYx\widehat{X}\widehat{Y}y$ **then**
      Compute i : the position of x in w.
      Compute j : the position of $\widehat{y}$ in $\widehat{w}$.
      If longest common extention$(w, \widehat{w}, i, j) = |x|$ **then**
        p is a *speudo-hexagon*.
      End if
    End if
  End for
End for



Detecting pseudo-hexagons

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

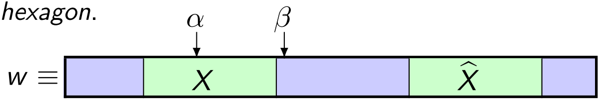
Build L_1 the list of all admissible factors that overlap the position α.
$\beta :=$ (the position of the rightmost letter of w include in a factor of L_1) $+ 1$.
Build L_2 the list of all admissible factors that overlap the position β.
For all $X \in L_1$ **do**
  For all $Y \in L_2$ **do**
    If $w \equiv XYx\widehat{X}\widehat{Y}y$ **then**
      Compute i : the position of x in w.
      Compute j : the position of $\widehat{y}$ in $\widehat{w}$.
      If longest common extention$(w, \widehat{w}, i, j) = |x|$ **then**
        p is a *speudo-hexagon*.
      End if
    End if
  End for
End for



Detecting pseudo-hexagons

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

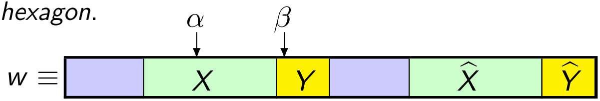
Build L_1 the list of all admissible factors that overlap the position α.
$\beta :=$ (the position of the rightmost letter of w include in a factor of L_1) $+ 1$.
Build L_2 the list of all admissible factors that overlap the position β.
For all $X \in L_1$ **do**
  For all $Y \in L_2$ **do**
    If $w \equiv XYx\widehat{X}\widehat{Y}y$ **then**
      Compute i : the position of x in w.
      Compute j : the position of $\widehat{y}$ in $\widehat{w}$.
      If longest common extention$(w, \widehat{w}, i, j) = |x|$ **then**
        p is a *speudo-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 α.
$\beta :=$ (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 β.
For all $X \in L_1$ **do**
  For all $Y \in L_2$ **do**
    If $w \equiv XYx\widehat{X}\widehat{Y}y$ **then**
      Compute i : the position of x in w.
      Compute j : the position of $\widehat{y}$ in $\widehat{w}$.
      If longest common extention$(w, \widehat{w}, i, j) = |x|$ **then**
        p is a *speudo-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 α.

$\beta :=$ (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 β.

For all $X \in L_1$ **do**

  For all $Y \in L_2$ **do**

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

      Compute i : the position of x in w.

      Compute j : the position of $\widehat{y}$ in $\widehat{w}$.

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

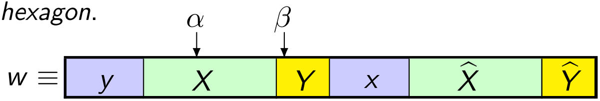
        p is a *speudo-hexagon*.

      End if

    End if

  End for

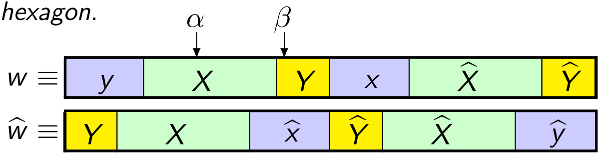
End for



Detecting pseudo-hexagons

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

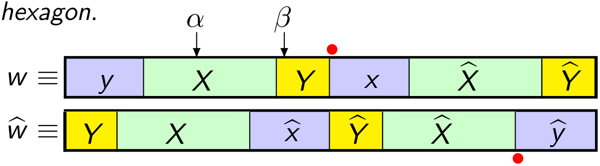
Build L_1 the list of all admissible factors that overlap the position α.
$\beta :=$ (the position of the rightmost letter of w include in a factor of L_1) $+ 1$.
Build L_2 the list of all admissible factors that overlap the position β.
For all $X \in L_1$ **do**
  For all $Y \in L_2$ **do**
    If $w \equiv XYx\widehat{X}\widehat{Y}y$ **then**
      Compute i : the position of x in w.
      Compute j : the position of $\widehat{y}$ in $\widehat{w}$.
      If longest common extention$(w, \widehat{w}, i, j) = |x|$ **then**
        p is a *speudo-hexagon*.
      End if
    End if
  End for
End for



Detecting pseudo-hexagons

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

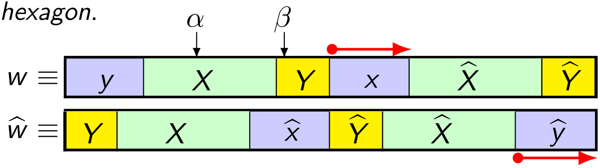
Build L_1 the list of all admissible factors that overlap the position α.
$\beta :=$ (the position of the rightmost letter of w include in a factor of L_1) $+ 1$.
Build L_2 the list of all admissible factors that overlap the position β.
For all $X \in L_1$ **do**
  For all $Y \in L_2$ **do**
    If $w \equiv XYx\widehat{X}\widehat{Y}y$ **then**
      Compute i : the position of x in w.
      Compute j : the position of $\widehat{y}$ in $\widehat{w}$.
      If longest common extention$(w, \widehat{w}, i, j) = |x|$ **then**
        p is a *speudo-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 α.
$\beta :=$ (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 β.
For all $X \in L_1$ **do**
  For all $Y \in L_2$ **do**
    If $w \equiv XYx\widehat{X}\widehat{Y}y$ **then**
      Compute i : the position of x in w.
      Compute j : the position of $\widehat{y}$ in $\widehat{w}$.
      If longest common extention$(w, \widehat{w}, i, j) = |x|$ **then**
        p is a *speudo-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 α.

$\beta :=$ (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 β.

For all $X \in L_1$ **do**

  For all $Y \in L_2$ **do**

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

      Compute i : the position of x in w.

      Compute j : the position of $\widehat{y}$ in $\widehat{w}$.

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

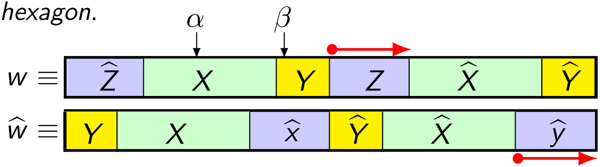
        p is a *speudo-hexagon*.

      End if

    End if

  End for

End for

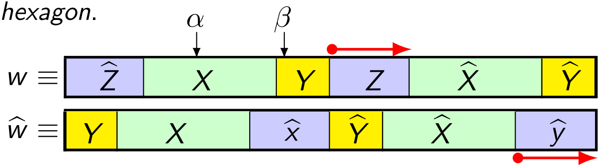


Detecting pseudo-hexagons

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

Build L_1 the list of all admissible factors that overlap the position α.
$\beta :=$ (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 β.
For all $X \in L_1$ **do**
  For all $Y \in L_2$ **do**
    If $w \equiv XYx\widehat{X}\widehat{Y}y$ **then**
      Compute i : the position of x in w.
      Compute j : the position of $\widehat{y}$ in $\widehat{w}$.
      If longest common extention$(w, \widehat{w}, i, j) = |x|$ **then**
        p is a *speudo-hexagon*.
      End if
    End if
  End for
End for

$$\mathcal{O}\left(n + (k + \log n)^2\right) = \mathcal{O}(n)$$



THANK YOU!