A CAT algorithm for 2-polyominoes

June 21th, 2017 • Journées MDSC • Nice, France.

Authors

Enrico Formenti

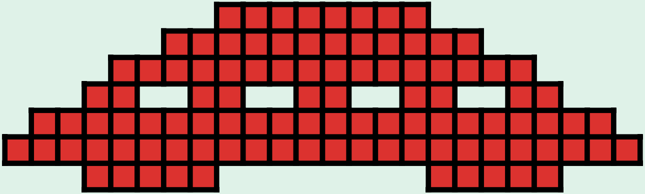
Laboratoire I3S
Université Côte d'Azur

Paolo Massazza

Università dell'Insubria
Varese, Italy.

Polyomino

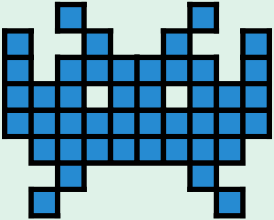
Any finite 4-connected subset of $\mathbb{Z}^2$



Polyominoes

Polyomino

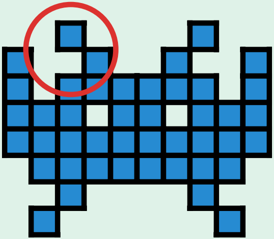
Any finite 4-connected subset of $\mathbb{Z}^2$



Polyominoes

Polyomino

Any finite 4-connected subset of $\mathbb{Z}^2$



Polyominoes: history and motivations

Introduction: Golomb (1954)
First extensive study: ???
Successively: too many!

Polyominoes are widely used in many fields:

- Enumerative combinatorics
- Bijective combinatorics
- Two dimensional language theory
- Tilings
- Discrete tomography

Polyominoes: the open questions

Enumeration

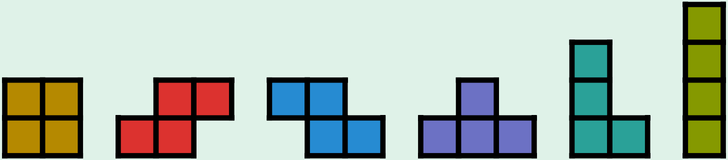
Can polyominoes of a given surface n be efficiently enumerated?

Closed formula

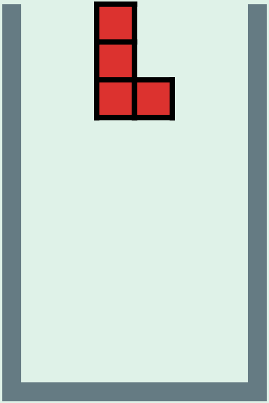
Is there a closed formula for the number of polyominoes of a given surface n?

Tetris

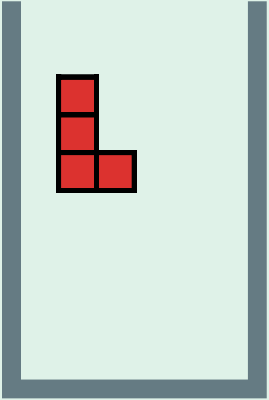
All variants of tetrominos fall one-by-one from top to bottom.



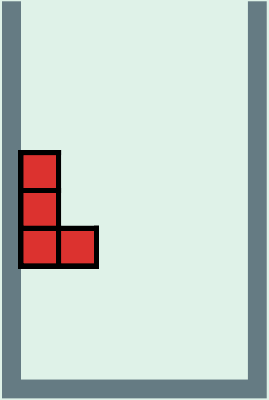
Classic Tetris



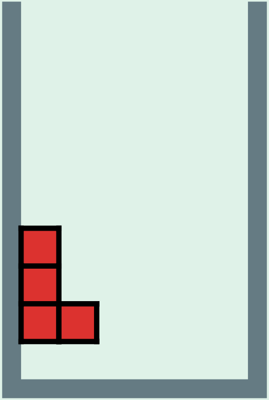
Classic Tetris



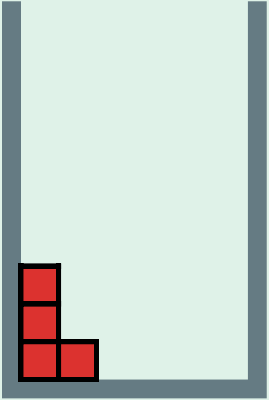
Classic Tetris



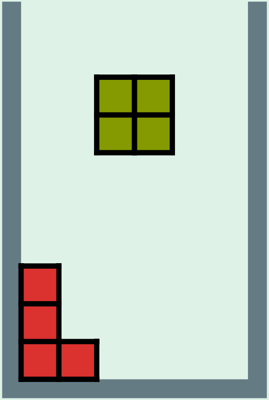
Classic Tetris



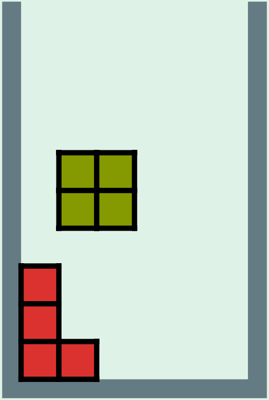
Classic Tetris



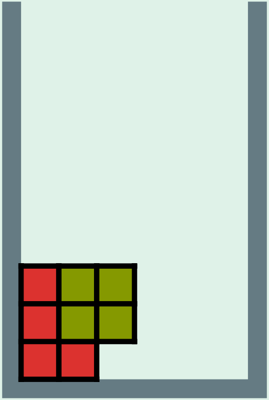
Classic Tetris



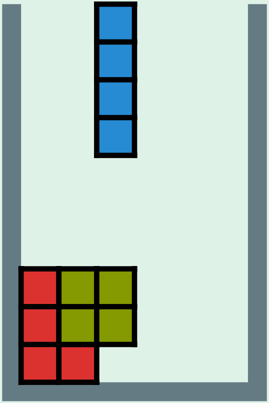
Classic Tetris



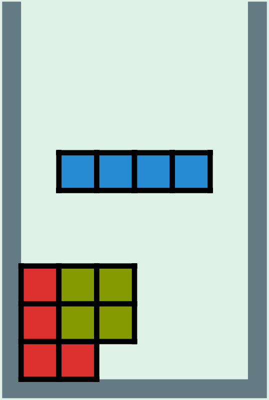
Classic Tetris



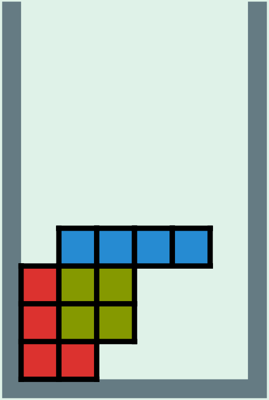
Classic Tetris



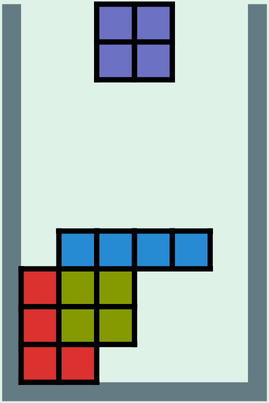
Classic Tetris



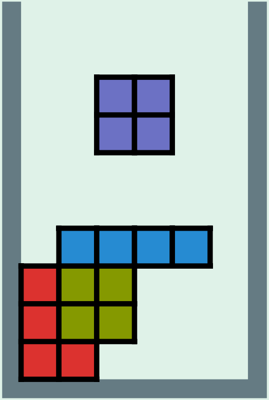
Classic Tetris



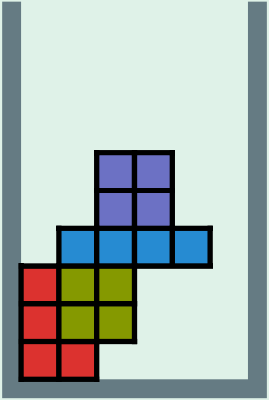
Classic Tetris



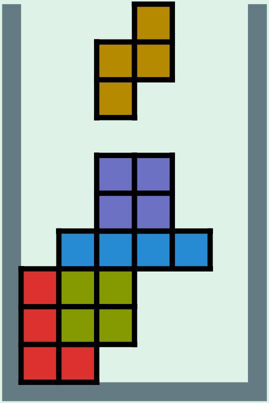
Classic Tetris



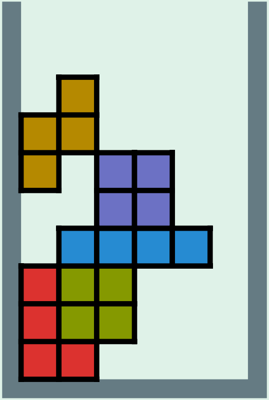
Classic Tetris



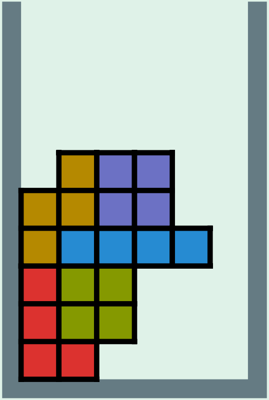
Classic Tetris



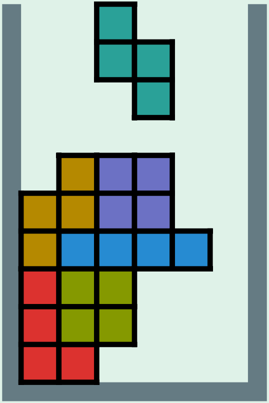
Classic Tetris



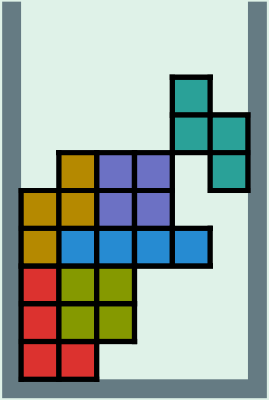
Classic Tetris



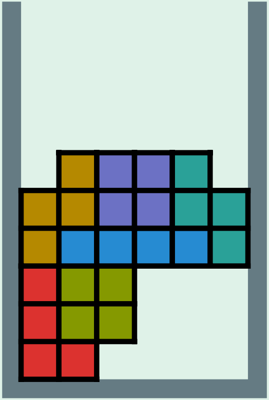
Classic Tetris



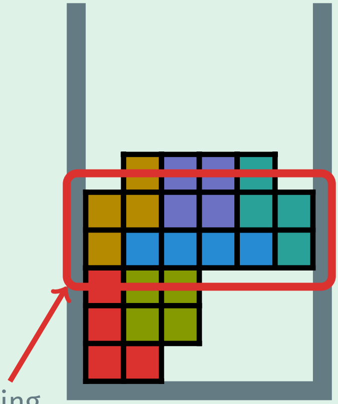
Classic Tetris



Classic Tetris

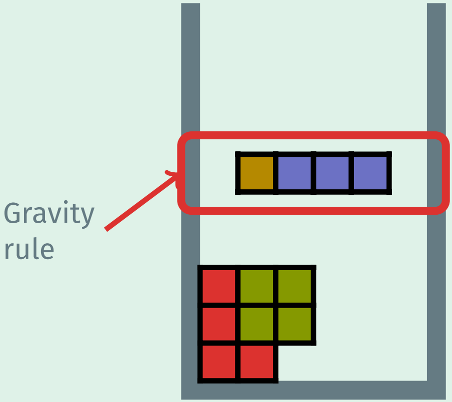


Classic Tetris

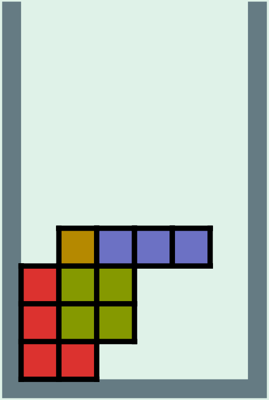


Cancelling rule

Classic Tetris



Classic Tetris



Bad Tetris and bad players

Pieces: any bar of length $\leq k$ (bad Tetris)

Bad Tetris and bad players

Pieces:	any bar of length $\leq k$	(bad Tetris)
Surface:	fix $s \in \mathbb{N}$	(bad player)

Bad Tetris and bad players

Pieces:	any bar of length $\leq k$	(bad Tetris)
Surface:	fix $s \in \mathbb{N}$	(bad player)
-Rules:	No cancelling	(bad Tetris)

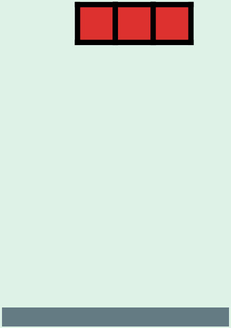
Bad Tetris and bad players

Pieces:	any bar of length $\leq k$	(bad Tetris)
Surface:	fix $s \in \mathbb{N}$	(bad player)
-Rules:	No cancelling	(bad Tetris)
+Rules:	New pieces and old ones have to be 4-connected	(bad player)

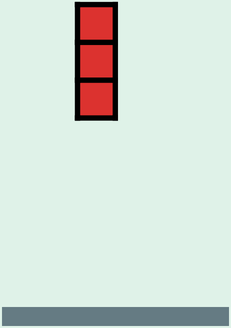
Bad Tetris and bad players

Pieces:	any bar of length $\leq k$	(bad Tetris)
Surface:	fix $s \in \mathbb{N}$	(bad player)
-Rules:	No cancelling	(bad Tetris)
+Rules:	New pieces and old ones have to be 4-connected	(bad player)
-Rules:	No more left and right walls	(bad Tetris)

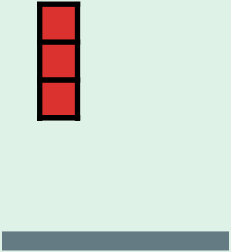
Bad Tetris (and bad player)



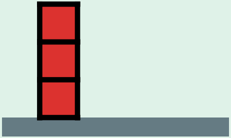
Bad Tetris (and bad player)



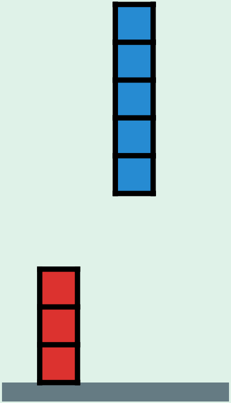
Bad Tetris (and bad player)



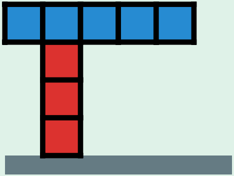
Bad Tetris (and bad player)



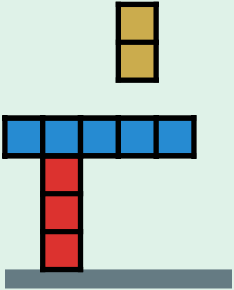
Bad Tetris (and bad player)



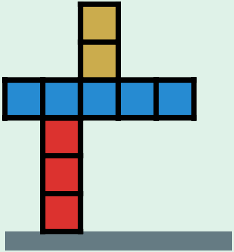
Bad Tetris (and bad player)



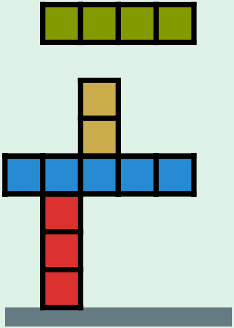
Bad Tetris (and bad player)



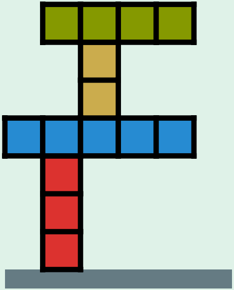
Bad Tetris (and bad player)



Bad Tetris (and bad player)



Bad Tetris (and bad player)



Polyomino representation

Polyomino

Ordered sequence of columns.

Column

Sequence of integers $\{\alpha_1, x_1, \alpha_2, x_2, \ldots, \alpha_k, x_k\}$
st.

α_1	=	vertical displacement wrt previous column
x_i	=	segment length
α_i	=	hole length ($i > 1$)

Some notation

Given $P = \{C_1, C_2, \ldots, C_i, \ldots C_k\}$ with
$C_1 = \{\alpha_1, x_1, \alpha_2, x_2, \ldots, \alpha_k, x_k\}$ and $C_1 = \{\alpha'_1, x'_1, \alpha'_2, x'_2, \ldots, \alpha'_l, x'_l\}$

$P_i = C_i$ (i-th column of P)

$P_{\leq i} = \{C_1, \ldots, C_i\}$

$C_i \Box C_{i-1}$ iff no segment of C_i falls completely in a hole of C_{i-1}

Comparing columns

$C_1 = \{\alpha_1, x_1, \alpha_2, x_2, \ldots, \alpha_k, x_k\}$

$C_2 = \{\alpha'_1, x'_1, \alpha'_2, x'_2, \ldots, \alpha'_l, x'_l\}$

$C_1 \leq C_2$ if one of the following holds

1. $\sum_{i=1}^{k} x_i < \sum_{j=1}^{l} x'_i$
2. $\sum_{i=1}^{k} x_i = \sum_{j=1}^{l} x'_i$ but $\alpha_1 > \alpha'_1$
3. $\sum_{i=1}^{k} x_i = \sum_{j=1}^{l} x'_i$ and $\alpha_1 = \alpha'_1$ but $\exists e$ s.t.
   3.1 $x_i = x'_i$ and $\alpha_i = \alpha'_i$ for $1 < i < e$
   3.2 $x_e > x'_e$ if e is even
   3.3 $\alpha_e < \alpha'_e$ if e is odd

Comparing Polyominoes

$P = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_i, \ldots \mathbf{c}_k\}$

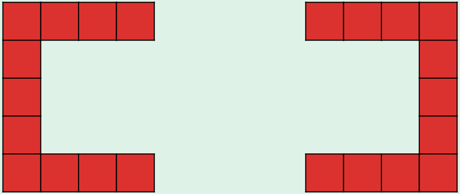
$Q = \{\mathbf{c}'_1, \mathbf{c}'_2, \ldots, \mathbf{c}'_i, \ldots \mathbf{c}'_l\}$

$P \leq Q$ iff $\exists i$ s.t. $\mathbf{c}_i < \mathbf{c}'_i$ and $\mathbf{c}_h = \mathbf{c}'_h$ for $h < i$.

Prefix-closed polyominos

Prefix-closed polyomino

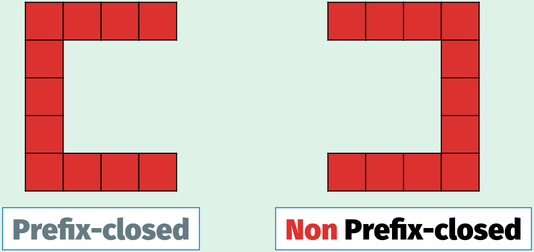
$P = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_i, \ldots \mathbf{c}_k\}$ is prefix-closed iff for all $i \in \{1, \ldots, k\}$, $P_{\leq i}$ is a polyomino.



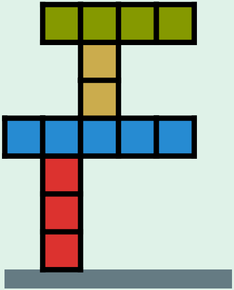
Prefix-closed polyominos

Prefix-closed polyomino

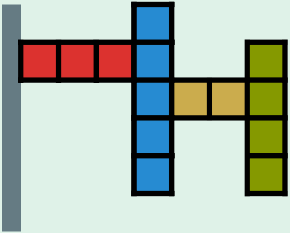
$P = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_i, \ldots \mathbf{c}_k\}$ is prefix-closed iff for all $i \in \{1, \ldots, k\}$, $P_{\leq i}$ is a polyomino.



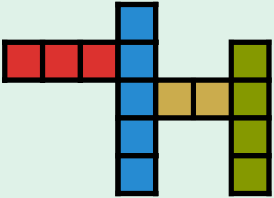
Bad Tetris + bad player = Prefix-closed Polyominoes



Bad Tetris + bad player = Prefix-closed Polyominoes



Bad Tetris + bad player = Prefix-closed Polyominoes

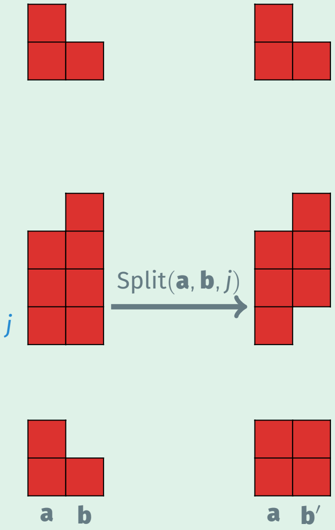


The result

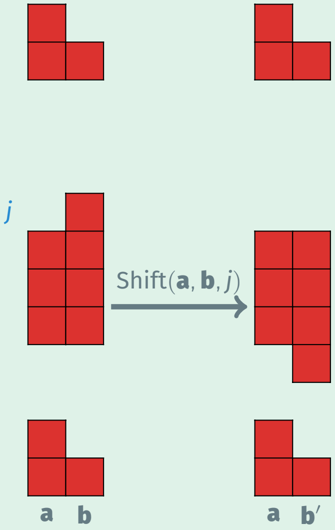
Theorem

There exist a CAT algorithm for exhaustive generation of PCPol(n).

Generating columns: split move



Generating columns: shift move



Generating columns: the grand ancestor

Grand ancestor

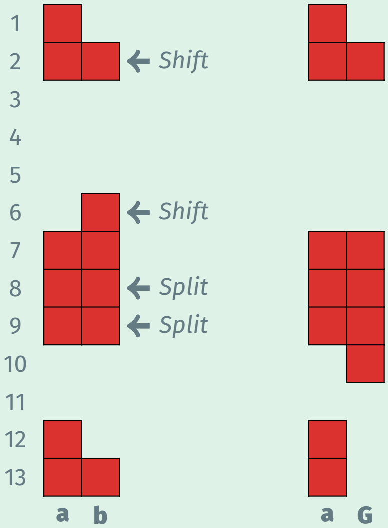
$C_1 = \{\alpha_1, x_1, \alpha_2, x_2, \dots, \alpha_k, x_k\}$

$C_2 = \{\alpha'_1, x'_1, \alpha'_2, x'_2, \dots, \alpha'_l, x'_l\}$

$G(C_2, C_1) =$ column which is

1. the smallest $\square$-compat. with C_1
2. identical to C_2 up to $j = \max\{M(C_2)\}$
3. admits a move at j
4. has same surface as C_2

Generating columns: the grand ancestor



$$M(b) = \{2, 6, 8, 9\}$$

Generating columns: the dynamical system

$C(\mathbf{a}, r)$ = set of all columns of area r and $\square$-compatible with $\mathbf{a}$

$\mathbf{b}' \xrightarrow{j} \mathbf{b}$ if $\mathbf{b}' = \mathrm{Split}(\mathbf{a}, \mathbf{b}, j)$ or $\mathbf{b}' = \mathrm{Shift}(\mathbf{a}, \mathbf{b}, j)$

$$f_{\mathbf{a},r}(\mathbf{b}) = \begin{cases} \mathbf{b}' & \exists \mathbf{b}', \mathrm{GA}(\mathbf{b}, \mathbf{a}) \xrightarrow{j} \mathbf{b}',\ j = \max M(\mathbf{b}) \\ (\alpha_1 - h(\mathbf{a}) + 1, r) & \text{otherwise} \end{cases}$$

Generating columns: the dynamical system (cont.)

Lemma

Fix an integer r and a column

$$\mathbf{a} = (\alpha_1, x_1, \alpha_2, x_2, \dots, \alpha_k, x_k)$$

then for all $\mathbf{b}, \mathbf{d} \in C(\mathbf{a}, r)$ it holds:

1. $f_{\mathbf{a},r}(\mathbf{b}) \square \mathbf{a}$
2. $\mathbf{b} < \mathbf{d}$ implies $f_{\mathbf{a},r}(\mathbf{b}) < f_{\mathbf{a},r}(\mathbf{d})$;
3. $f^n_{\mathbf{a},r}(\mathbf{b}) < f^{n+1}_{\mathbf{a},r}(\mathbf{b})$ for $\mathbf{b} \neq (\alpha_1 - h(\mathbf{a}) + 1, r)$;
4. $\bigcup_{n\in\mathbb{N}} f^n_{\mathbf{a},r}((r-1, r)) = C(\mathbf{a}, r)$.

Generating PCPol(n): the algorithm

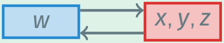
```
PROCEDURE PCPOLGEN(n)
for r := 1 to n − 1 do
    P_1 := (0, r); COLGEN(2, n − r);
end for
P := (0, n); OUTPUT(P);
```

Generating PCPol(n): the algorithm (columns)

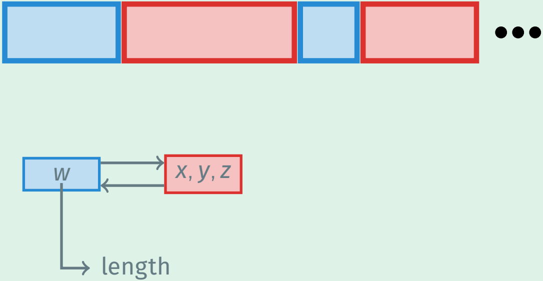
PROCEDURE COLGEN(i, r)
for d := 1 to r **do**
    INITCOLUMN(i); {i.e. P_i:= $((d), d-1)$}
    INIT(S, i); {init stack}
    if $d < r$ **then** COLGEN($i+1, r-d$); **else** OUTPUT(P); **endif**
    while not ISEMPTY(S) **do**
        j:=GRAN(i, S); {restore the grand ancestor}
        MOVE(j, S); {execute a move at j and update S}
        if $d < r$ **then** COLGEN($i+1, r-d$); **else** OUTPUT(P);
        endif
    end while
end for

Generating PCPol(n): the algorithm (columns)

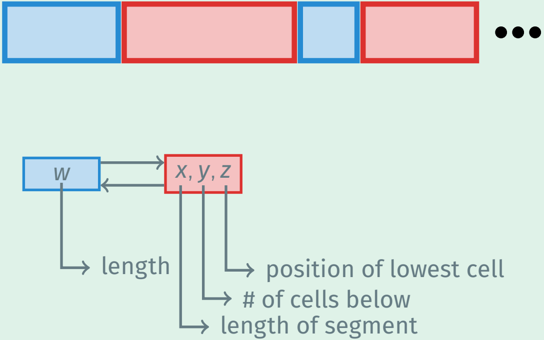
Generating PCPol(n): the algorithm (columns)



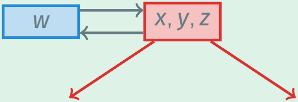
Generating PCPol(n): the algorithm (columns)



Generating PCPol(n): the algorithm (columns)



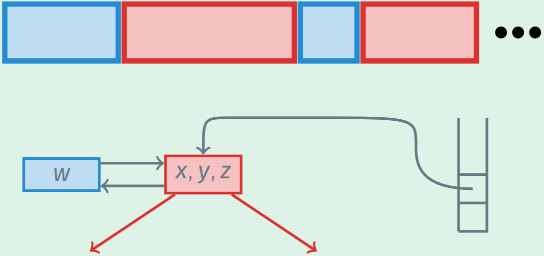
Generating PCPol(n): the algorithm (columns)



ptr to smallest compat. segment of prev. column

ptr to largest compat. segment of prev. column

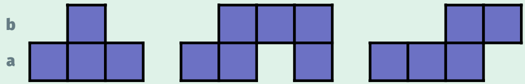
Generating PCPol(n): the algorithm (columns)

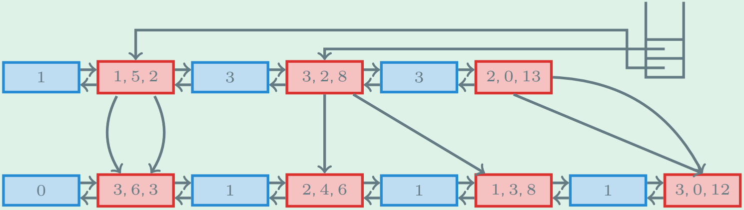


ptr to smallest compat. segment of prev. column

ptr to largest compat. segment of prev. column

Generating PCPol(n): an example





Conclusions and perspectives

- CAT algorithm for $PCPol(n)$

Conclusions and perspectives

- CAT algorithm for $PCPol(n)$ $\Leftarrow$ Done

Conclusions and perspectives

- CAT algorithm for $PCPol(n)$ $\Leftarrow$ Done
- Closed formula for $|PCPol(n)|$

Conclusions and perspectives

- CAT algorithm for $PCPol(n)$ $\Leftarrow$ Done
- Closed formula for $|PCPol(n)|$ To do!

Conclusions and perspectives

- CAT algorithm for $PCPol(n)$ $\Leftarrow$ Done
- Closed formula for $|PCPol(n)|$ To do!
- Extension to k-$PCPol(n)$ ($k \in \mathbb{N}$ fixed)

Conclusions and perspectives

- CAT algorithm for $PCPol(n)$ $\Leftarrow$ Done
- Closed formula for $|PCPol(n)|$ To do!
- Extension to k-$PCPol(n)$ ($k \in \mathbb{N}$ fixed) $\Leftarrow$ Done (?)

Conclusions and perspectives

- CAT algorithm for $\mathit{PCPol}(n)$ $\Leftarrow$ Done
- Closed formula for $|\mathit{PCPol}(n)|$ To do!
- Extension to k-$\mathit{PCPol}(n)$ ($k \in \mathbb{N}$ fixed) $\Leftarrow$ Done (?)
- Closed formula for $|k\text{-}\mathit{PCPol}(n)|$

Conclusions and perspectives

- CAT algorithm for $PCPol(n)$ $\Leftarrow$ Done
- Closed formula for $|PCPol(n)|$ To do!
- Extension to k-$PCPol(n)$ ($k \in \mathbb{N}$ fixed) $\Leftarrow$ Done (?)
- Closed formula for $|k\text{-}PCPol(n)|$ To do!!

Conclusions and perspectives

- CAT algorithm for $PCPol(n)$ $\Leftarrow$ Done
- Closed formula for $|PCPol(n)|$ To do!
- Extension to k-$PCPol(n)$ ($k \in \mathbb{N}$ fixed) $\Leftarrow$ Done (?)
- Closed formula for $|k\text{-}PCPol(n)|$ To do!!
- Extension to $Pol(n)$

Conclusions and perspectives

- CAT algorithm for $PCPol(n)$ $\Leftarrow$ Done
- Closed formula for $|PCPol(n)|$ To do!
- Extension to k-$PCPol(n)$ ($k \in \mathbb{N}$ fixed) $\Leftarrow$ Done (?)
- Closed formula for $|k\text{-}PCPol(n)|$ To do!!
- Extension to $Pol(n)$ To do!!!

Conclusions and perspectives

- CAT algorithm for $PCPol(n)$ $\Leftarrow$ Done
- Closed formula for $|PCPol(n)|$ To do!
- Extension to k-$PCPol(n)$ ($k \in \mathbb{N}$ fixed) $\Leftarrow$ Done (?)
- Closed formula for $|k\text{-}PCPol(n)|$ To do!!
- Extension to $Pol(n)$ To do!!!
- Please, go meta!!!

Thank you.