Hole-free Partially Directed Animals

Paolo Massazza[1]

[1]Department of Theoretical and Applied Sciences
University of Insubria
Varese Italy

DLT'19 – Warsaw, August 8st 2019

Abstract

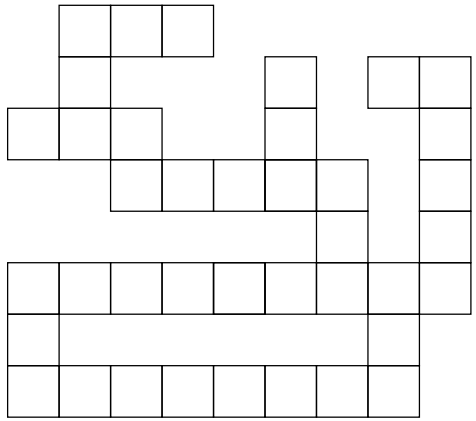
We consider the problem of efficiently generating classes of polyominoes.

We present a Constant Amortized Time algorithm that uses space $O(n)$ to generate all partially directed animals of area n having no holes

The approach can be extended to partially directed animals with a fixed number of holes.

Polyominoes

A polyomino is a finite connected set of edge-to-edge adjacent 1×1 cells in the $2D$ plane



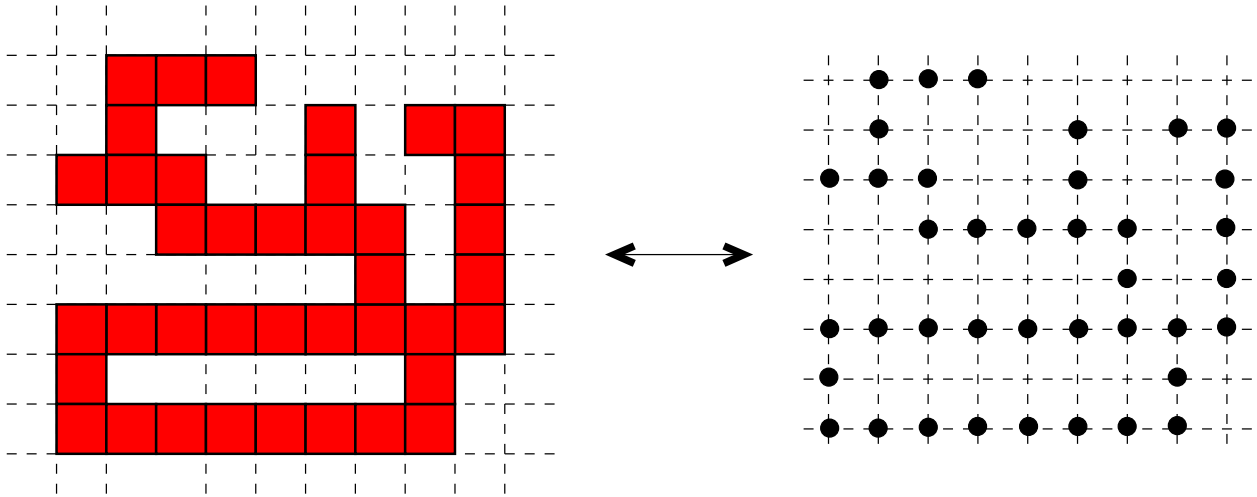
- Introduced by Golomb (1954)
- Widely studied in
  - enumerative combinatorics
  - bijective combinatorics
  - two dimensional language theory
  - tiling theory
  - discrete tomography

Animals

Definition

An animal is a finite connected set of vertices of a lattice

Animals can be seen as duals of polyominoes (each cell replaced by a vertex at its center - square lattice)



Partially Directed Animals

Definition (PDA)

A partially directed animal of area n is a set of n connected elements (points) s.t. any element can be reached from a distinguished element (the root) by a path which never goes opposite to a given direction

- Used to study physical phenomena (e.g. percolation)

Definition (Prefix-Closed Polyominoes PCPol(n))

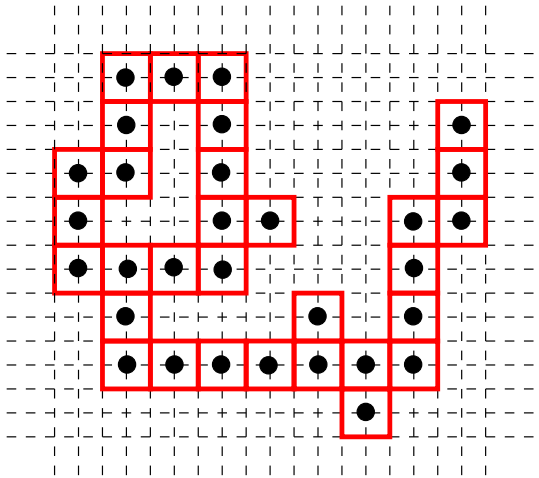
PCPol(n) is the class of polyominoes P of area n s.t. for any $i > 0$, the first i columns of P comprise exactly one polyomino

Fix the Square lattice and the direction East, then

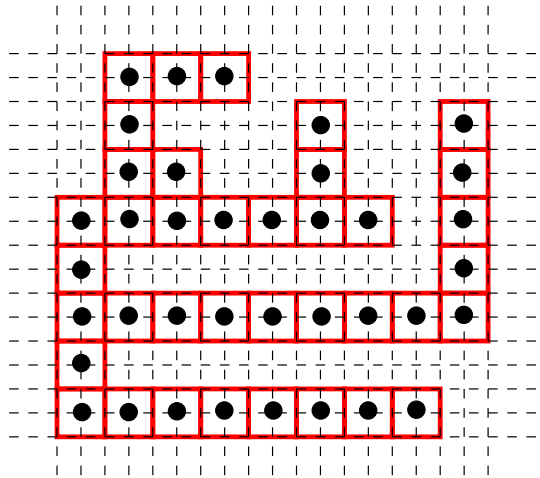
$$\text{PCPol}(n) \equiv \text{PDA}(n)$$

Hole-Free PDAs

We are interested into an efficient algorithm for generating the set HFPDA(n) of Hole-Free Partially Directed Animals (*i.e.* Hole-free Prefix-closed polyominoes)



(a)



(b)

(a) a polyomino not in HFPDA

(b) a polyomino in HFPDA

Known results

There exist efficient (CAT) algorithms for generating (w.r.t. the area)

- polyominoes [Redelmeier 1981]
- parallelogram polyominoes [Mantaci, Massazza 2011]
- L-convex polyominoes [Massazza 2012]
- convex polyominoes [Massazza 2015]
- k-convex polyominoes [Brocchi, Castiglione, Massazza 2017]
- partially directed animals [Formenti, Massazza 2017]

Notation and definitions

- a column is a sequence of segments, $\mathbf{c} = (s_1, s_2, \ldots, s_h)$
- a polyomino is a sequence of columns, $P = (\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k)$
- $\mathrm{Top}(s) = y$-coordinate of the upper cell of s
- $\mathrm{Top}(\mathbf{c}) = \mathrm{Top}(s)$, s upper segment of $\mathbf{c}$
- $s = (\mathrm{Top}(s), \mathrm{A}(s))$
- $\mathbf{c}_{>j}$ = part of $\mathbf{c}$ above position j
- P_i is the i-th column of P
- The i-prefix of P is $P_{\leq i} = (\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_i)$
- $s \in P_i$ is left-adjacent $\implies \exists t \in P_{i-1}$ s.t. s, t have a common edge

Generation: a general method

We generate a family C of polyominoes in three steps:

1. Define C in terms of a property $\mathcal{V}$ satisfied by i-prefixes:

$$P \in C \Leftrightarrow \forall i > 0, \mathcal{V}(P_{\leq i})$$

2. Define a compatibility relation $\square$ on adjacent columns,

$$\mathcal{V}(P_{\leq i}) \wedge \mathbf{c} \square P_i \Leftrightarrow \mathcal{V}(P_{\leq i}|\mathbf{c})$$

3. For any $r > 1$, define a discrete dynamical system that generates all columns of area r that are compatible with P_i

Generation: a general method

We generate a family C of polyominoes in three steps:

1. Define C in terms of a property $\mathcal{V}$ satisfied by i-prefixes:

$$P \in C \Leftrightarrow \forall i > 0, \mathcal{V}(P_{\leq i})$$

2. Define a compatibility relation $\square$ on adjacent columns,

$$\mathcal{V}(P_{\leq i}) \wedge \mathbf{c} \square P_i \Leftrightarrow \mathcal{V}(P_{\leq i}|\mathbf{c})$$

3. For any $r > 1$, define a discrete dynamical system that generates all columns of area r that are compatible with P_i

Generation: a general method

We generate a family C of polyominoes in three steps:

1. Define C in terms of a property $\mathcal{V}$ satisfied by i-prefixes:

$$P \in C \Leftrightarrow \forall i > 0, \mathcal{V}(P_{\leq i})$$

2. Define a compatibility relation $\square$ on adjacent columns,

$$\mathcal{V}(P_{\leq i}) \wedge \mathbf{c} \square P_i \Leftrightarrow \mathcal{V}(P_{\leq i}|\mathbf{c})$$

3. For any $r > 1$, define a discrete dynamical system that generates all columns of area r that are compatible with P_i

HFPDA generation

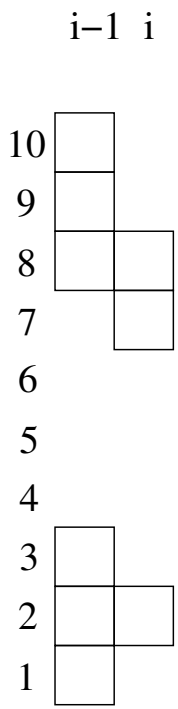
The previous method can be applied to HFPDA

1. **Lemma:** $P \in$ HFPDA iff $\forall i > 1$ any $s \in P_i$ is left-adjacent to exactly one $t \in P_{i-1}$
2. $\mathbf{c} \square \mathbf{b}$ iff $s \in \mathbf{c} \implies \exists! t \in \mathbf{b}$ s.t. s is left-adjacent to t
3. The discrete dynamical system is based on three moves applied to segments in a column: SHIFT, SPLIT and SHIFT&SPLIT

Notation: Comp($\mathbf{a}, r$) = set of columns of area r that are compatible with $\mathbf{a}$

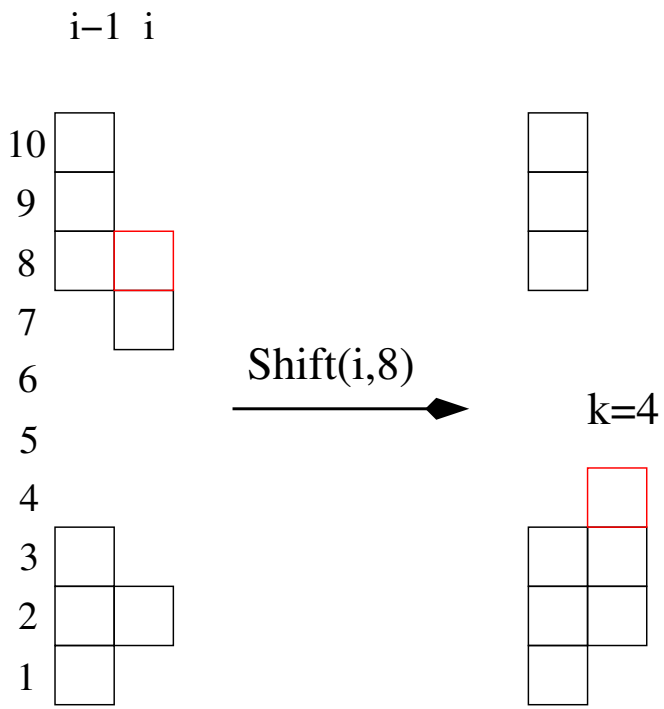
The moves

Shift



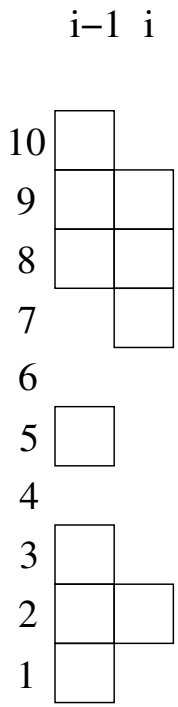
The moves

Shift



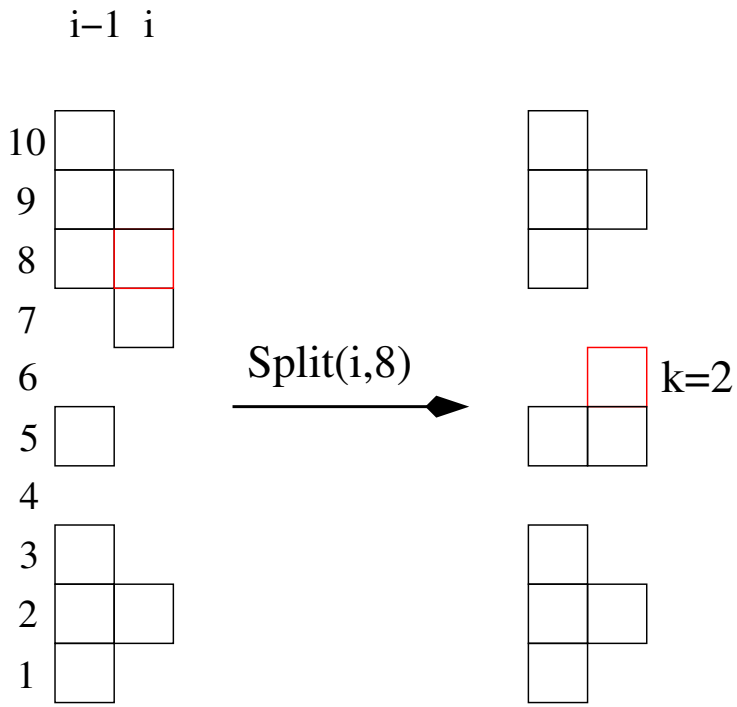
The moves

Split



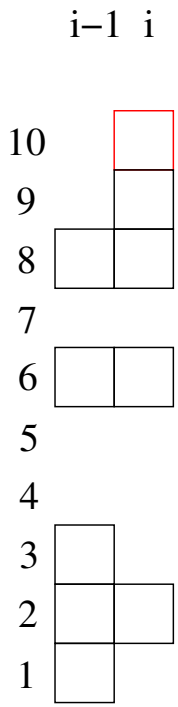
The moves

Split



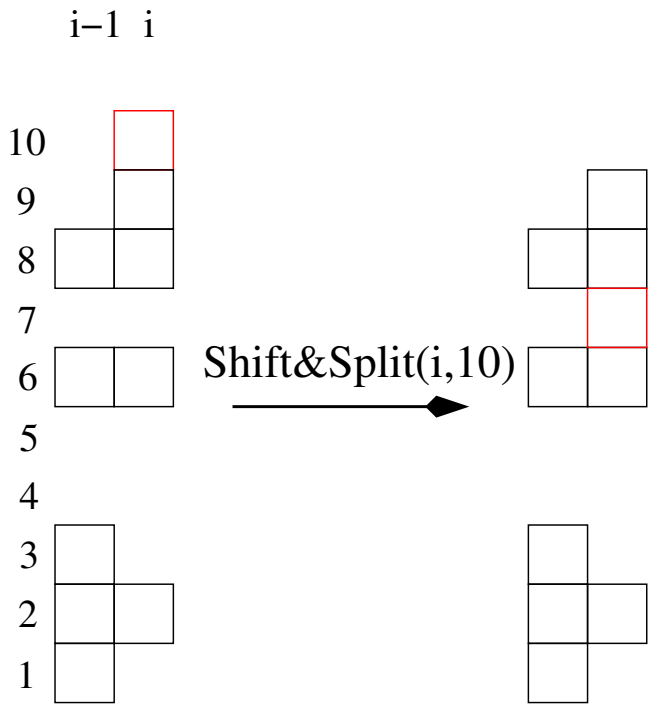
The moves

Shift&Split



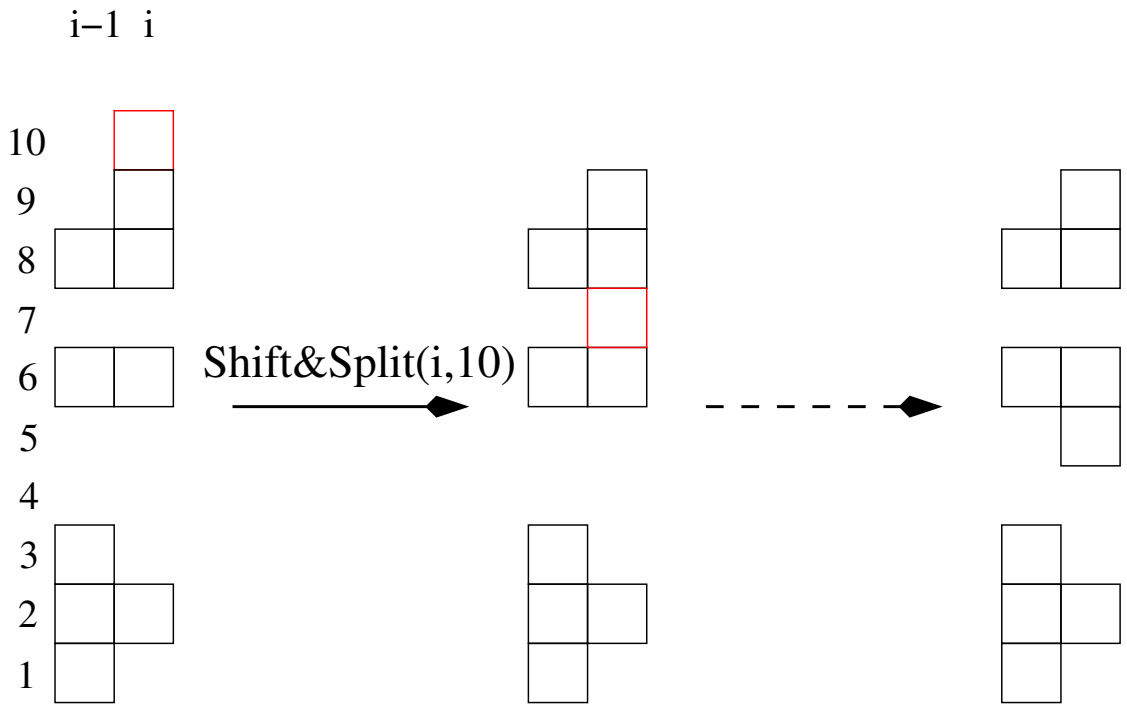
The moves

Shift&Split



The moves

Shift&Split



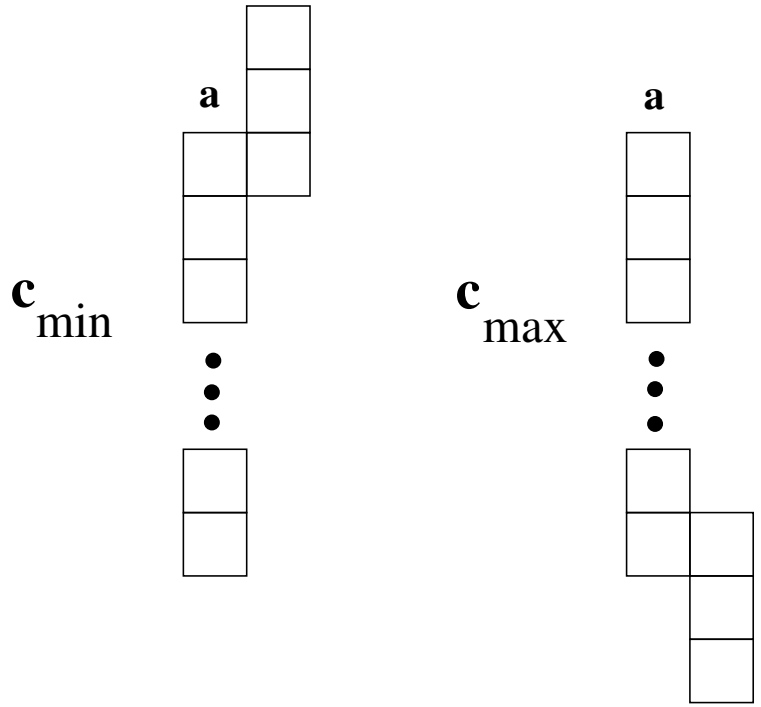
Order on columns

We consider a total order $\prec$ on columns

$$\mathbf{b} \prec \mathbf{c} \Leftrightarrow \exists j, \mathbf{b}_{>j} = \mathbf{c}_{>j}, \mathbf{b}_{\geq j} \neq \mathbf{c}_{\geq j}, \mathbf{c} \text{ has an empty cell at } j$$

$$\mathbf{c}_{\min} = \min(\mathrm{Comp}(\mathbf{a}, r)) = ((r, \mathrm{Top}(\mathbf{a}) + r - 1))$$

$$\mathbf{c}_{\max} = \max(\mathrm{Comp}(\mathbf{a}, r)) = ((r, \mathrm{Bot}(\mathbf{a}))$$



Remark: no move occurs in $\mathbf{c}_{\max}$

The DDS for Comp($\mathbf{a}, r$)

Definition

Given $\mathbf{a}$ and $r > 0$, define $f_{\mathbf{a},r}$: Comp($\mathbf{a}, r$) $\to$ Comp($\mathbf{a}, r$)

$$f_{\mathbf{a},r}(\mathbf{b}) = \begin{cases} \mathbf{c} & \mathbf{g} \xrightarrow{j} \mathbf{c} \\ \mathbf{c}_{\max} & \text{if } \mathbf{b} = \mathbf{c}_{\max} \end{cases}$$

where

- j is the position of the lowest move in $\mathbf{b}$
- $\mathbf{g}$ is the grand ancestor of $\mathbf{b}$ w.r.t. $\mathbf{a}$, *i.e.* a suitable column that is used to obtain the column following $\mathbf{b}$ in Comp($\mathbf{a}, r$) (w.r.t $\prec$)

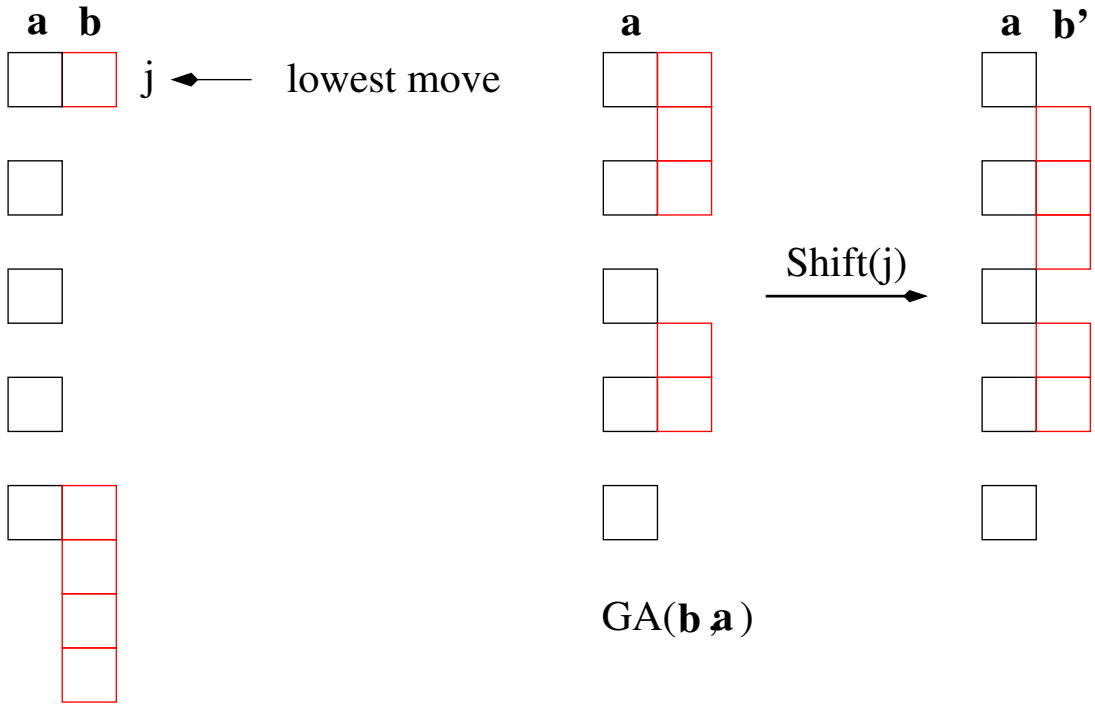
The grand ancestor

Informally, the grand ancestor of **b** w.r.t **a** is a column **g** s.t.

- $A(\mathbf{g}) = A(\mathbf{b})$
- $\mathbf{g}_{\geq j} = \mathbf{b}_{\geq j}$, j is the position of the lowest move in **b**
- **g** admits a move at j
- the segment t of **g** at j is the longest possible s.t. the move at j has the smallest shift
- the segments below t have maximal position and maximal area

The grand ancestor

Example



b′ is the smallest compatible column greater than **b**

DDS: properties

Lemma

Let $f^n_{\mathbf{a},r}$ be the n-fold composition of $f_{\mathbf{a},r}$ with itself. Then

1. $f^n_{\mathbf{a},r}(\mathbf{b}) \prec f^{n+1}_{\mathbf{a},r}(\mathbf{b})$ *for* $\mathbf{b} \neq \mathbf{c}_{\max}$;
2. $\bigcup_{n\in\mathbb{N}} f^n_{\mathbf{a},r}(\mathbf{c}_{\min}) = \mathit{Comp}(\mathbf{a}, r)$.

where $\mathbf{c}_{\max} = ((r, \mathit{Bot}(\mathbf{a})))$ *and* $\mathbf{c}_{\min} = ((r, \mathit{Top}(\mathbf{a}) + r - 1))$

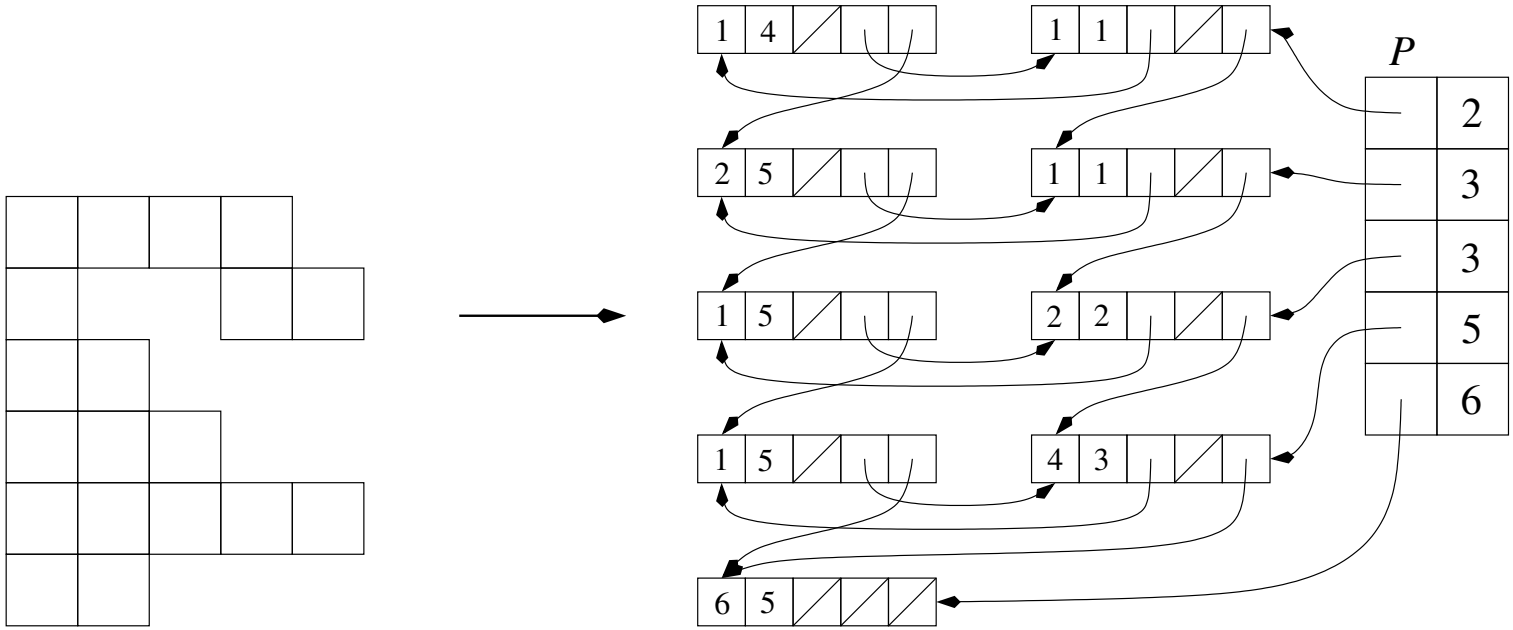
Generation of HFPDA

We use an inductive approach (remark: P_1 consists of exactly one segment)
Given $P_{<i}$ (already generated) call ColGen(i, r)

```
Procedure ColGen(i,r)
for d:=r downto 1
 P[i]:=((d,Top(P[i-1])+d-1));\\smallest column
 if d<r then ColGen(i+1,r-d);else Output(P);
 while NOT IsFixedPoint(P[i])
  j:=GrAn(P[i],P[i-1]);\\restore grandancestor
  Move(P[i],j);
  if d<r then ColGen() else Output(P)
 endwhile
endfor
```

Complexity

A simple data structure allows to execute moves in time $O(1)$



All the functions within ColGen (apart GrAn) have cost $O(1)$

Complexity

In general, GRAN(P_i, P_{i-1}) runs in time $O(\mathrm{A}(P_{i-1}))$

Lemma

*For each call G=*GRAN*(P_i, P_{i-1}) with cost $\Theta(p)$ there exists a set T_G with $|T_G| = \Omega(p)$ s.t.*

- *each column in T_G is generated in time $O(1)$*
- *$T_{G_1} \cap T_{G_2} = \emptyset$ for G_1, G_2 with cost $\Theta(p_1), \Theta(p_2)$*

$$\Downarrow$$

Lemma

COLGEN*(i, r) runs in time $O(t)$ where $t = |Comp(P_{i-1}, r)|$*

$$\Downarrow$$

Theorem

there exists a CAT algorithm for generating HFPDA(n) *(using space $O(n)$)*

Conclusions

We have developed a CAT algorithm to generate HFPDA(n)

Work in progress

- Definition of similar DDSs to obtain CAT algorithms for
  - Partially directed animals with at most k holes
  - Partially directed animals with exactly k holes
- Implementation of the algorithms to obtain counting sequences