Counting polyominoes on the twisted cylinder

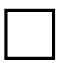
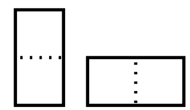
Gill Barequet and Micha Moffie
The Technion, Haifa

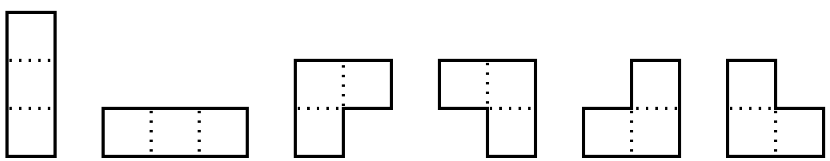
Ares Ribó and Günter Rote
Freie Universität Berlin

Supported by the Deutsche Forschungsgemeinschaft within the European Graduate Program *Combinatorics, Geometry and Computation* (No. GRK 588/2)

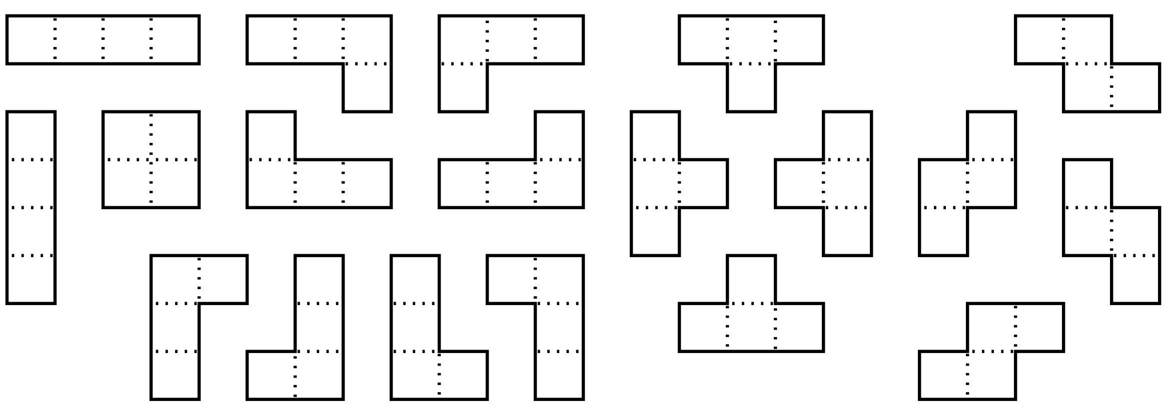
Polyominoes (or lattice animals)

A *polyomino* is a connected subset of squares of the integer grid.

 1 monomino: $A_1 = 1$  2 dominoes: $A_2 = 2$



6 triominoes: $A_3 = 6$



19 tetrominoes:

$A_4 = 19$

$A_n :=$ the number of n-ominoes

n	A_n
1	1
2	2
3	6
4	19
5	63
6	216
7	760
8	2,725
9	9,910
10	36,446
11	135,268
12	505,861
13	1,903,890
14	7,204,874
15	27,394,666
16	104,592,937
17	400,795,844
18	1,540,820,542
19	5,940,738,676
20	22,964,779,660
21	88,983,512,783
22	345,532,572,678
23	1,344,372,335,524
24	5,239,988,770,268
25	20,457,802,016,011
26	79,992,676,367,108
27	313,224,032,098,244
28	1,228,088,671,826,973

n	A_n = polyominoes with n cells
29	4,820,975,409,710,116
30	18,946,775,782,611,174
31	74,541,651,404,935,148
32	293,560,133,910,477,776
33	1,157,186,142,148,293,638
34	4,565,553,929,115,769,162
35	18,027,932,215,016,128,134
36	71,242,712,815,411,950,635
37	281,746,550,485,032,531,911
38	1,115,021,869,572,604,692,100
39	4,415,695,134,978,868,448,596
40	17,498,111,172,838,312,982,542
41	69,381,900,728,932,743,048,483
42	275,265,412,856,343,074,274,146
43	1,092,687,308,874,612,006,972,082
44	4,339,784,013,643,393,384,603,906
45	17,244,800,728,846,724,289,191,074
46	68,557,762,666,345,165,410,168,738
47	272,680,844,424,943,840,614,538,634
48	1,085,035,285,182,087,705,685,323,738
49	4,319,331,509,344,565,487,555,270,660
50	17,201,460,881,287,871,798,942,420,736
51	68,530,413,174,845,561,618,160,604,928
52	273,126,660,016,519,143,293,320,026,256
53	1,088,933,685,559,350,300,820,095,990,030
54	4,342,997,469,623,933,155,942,753,899,000
55	17,326,987,021,737,904,384,935,434,351,490
56	69,150,714,562,532,896,936,574,425,480,218

n	A_n
1	1
2	2
3	6
4	19
5	63
6	216
7	760
8	2,725
9	9,910
10	36,446
11	135,268
12	505,861
13	1,903,890
14	7,204,874
15	27,394,666
16	104,592,937
17	400,795,844
18	1,540,820,542
19	5,940,738,676
20	22,964,779,660
21	88,983,512,783
22	345,532,572,678
23	1,344,372,335,524
24	5,239,988,770,268
25	20,457,802,016,011
26	79,992,676,367,108
27	313,224,032,098,244
28	1,228,088,671,826,973

n	A_n = polyominoes with n cells
29	4,820,975,409,710,116
30	18,946,775,782,611,174
31	74,541,651,404,935,148
32	293,560,133,910,477,776
33	1,157,186,142,148,293,638
34	4,565,553,929,115,769,162
35	18,027,932,215,016,128,134
36	71,242,712,815,411,950,635
37	281,746,550,485,032,531,911
38	1,115,021,869,572,604,692,100
39	4,415,695,134,978,868,448,596
40	17,498,111,172,838,312,982,542
41	69,381,900,728,932,743,048,483
42	275,265,412,856,343,074,274,146
43	1,092,687,308,874,612,006,972,082
44	4,339,784,013,643,393,384,603,906
45	17,244,800,728,846,724,289,191,074
46	68,557,762,666,345,165,410,168,738
47	272,680,844,424,943,840,614,538,634
48	1,085,035,285,182,087,705,685,323,738
49	4,319,331,509,344,565,487,555,270,660
50	17,201,460,881,287,871,798,942,420,736
51	68,530,413,174,845,561,618,160,604,928
52	273,126,660,016,519,143,293,320,026,256
53	1,088,933,685,559,350,300,820,095,990,030
54	4,342,997,469,623,933,155,942,753,899,000
55	17,326,987,021,737,904,384,935,434,351,490
56	69,150,714,562,532,896,936,574,425,480,218

Enumeration and counting of polyominoes

- Ron Read [1962]
- ...
- the transfer matrix method:
  A. R. Conway and A. J. Guttman [1995],
  I. Jensen [2001], D. E. Knuth [2001].
- I. Jensen [2003] computed A_{56}, using parallel computers.

The asymptotic growth of A_n

$$\lim_{n\to\infty} \sqrt[n]{A_n} =: \lambda \text{ exists.} \quad \text{[D. A. Klarner 1967]}$$

$\lambda =$ *Klarner's constant*.

$$3.874 \leq \lambda \leq 4.65$$

Lower bound comes from "extrapolation" from the known values $A_1, \ldots, A_{56}$
("pseudo-renewal sequences" [Rands and Welsh 1981])

λ is estimated to be ≈ 4.06.

Does $\lim_{n\to\infty} \frac{A_{n+1}}{A_n}$ exist? $A_n \sim \text{const} \cdot \lambda^n n^{-1}$?

Best previous lower bound on Klarner's constant:

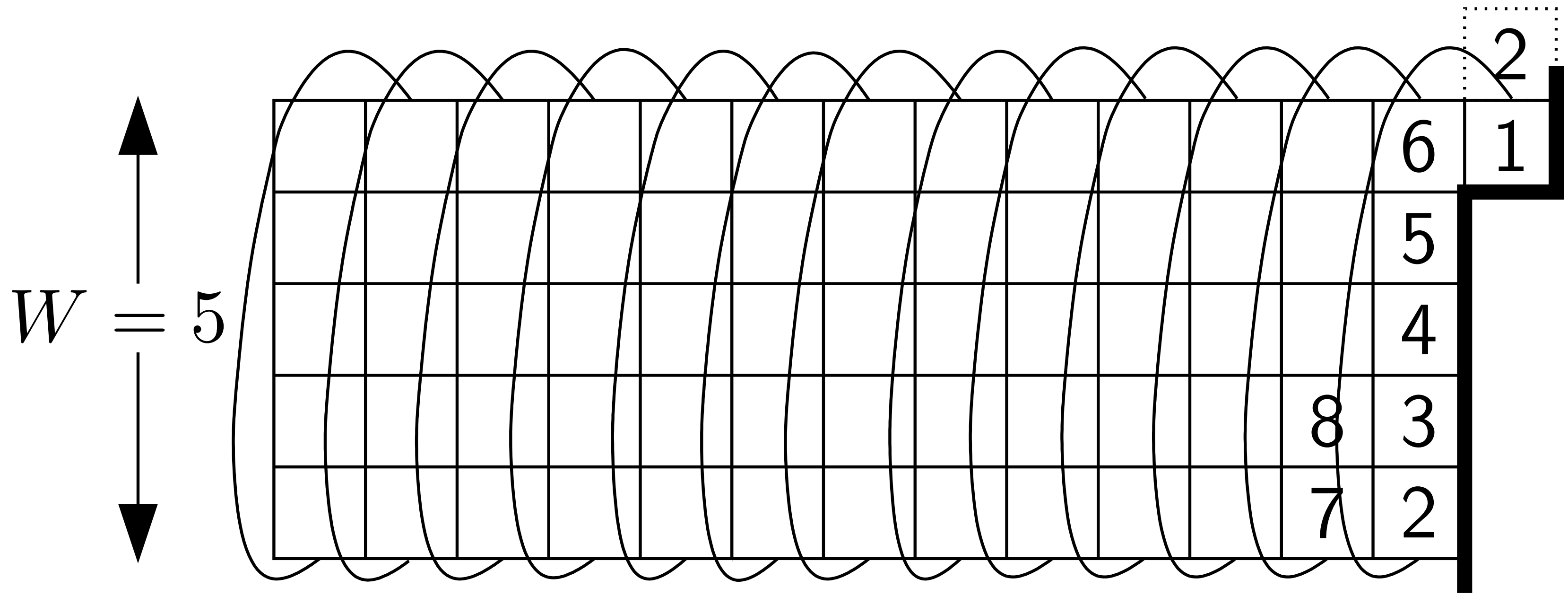
$$3.874 \leq \lambda \leq 4.65$$

We improve the lower bound to

$$\lambda \geq 3.9801$$

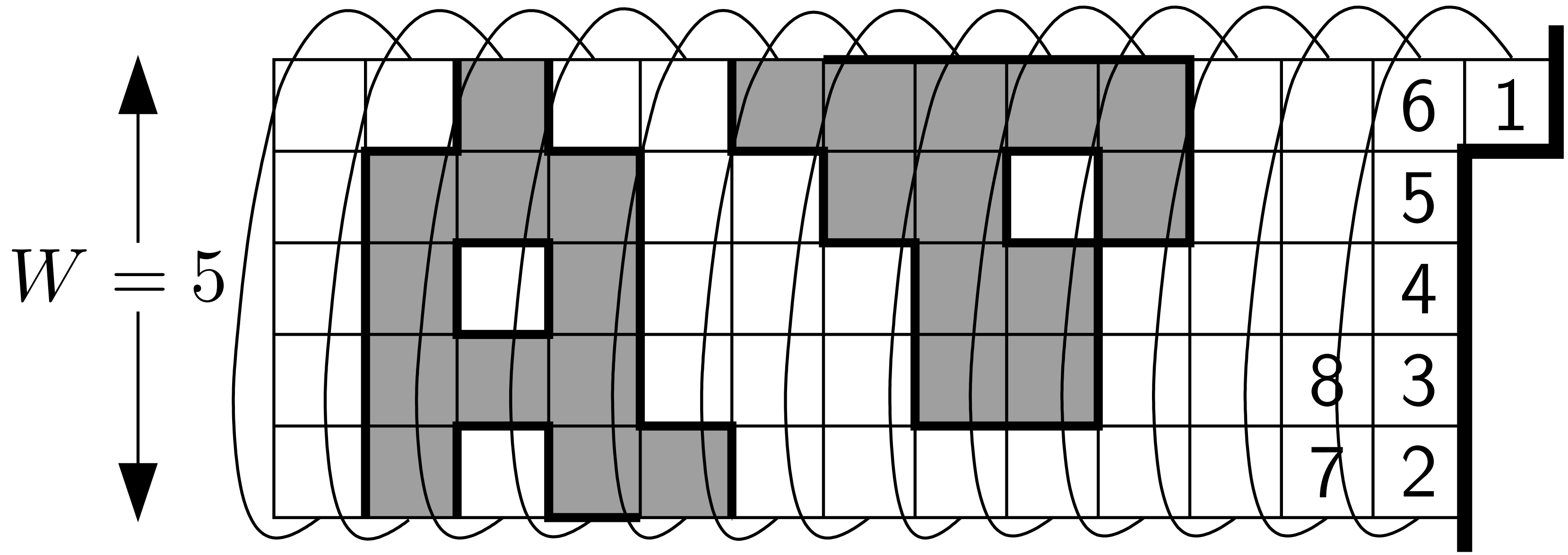
by counting polyominoes on a *twisted cylinder*.

A twisted cylinder of width W



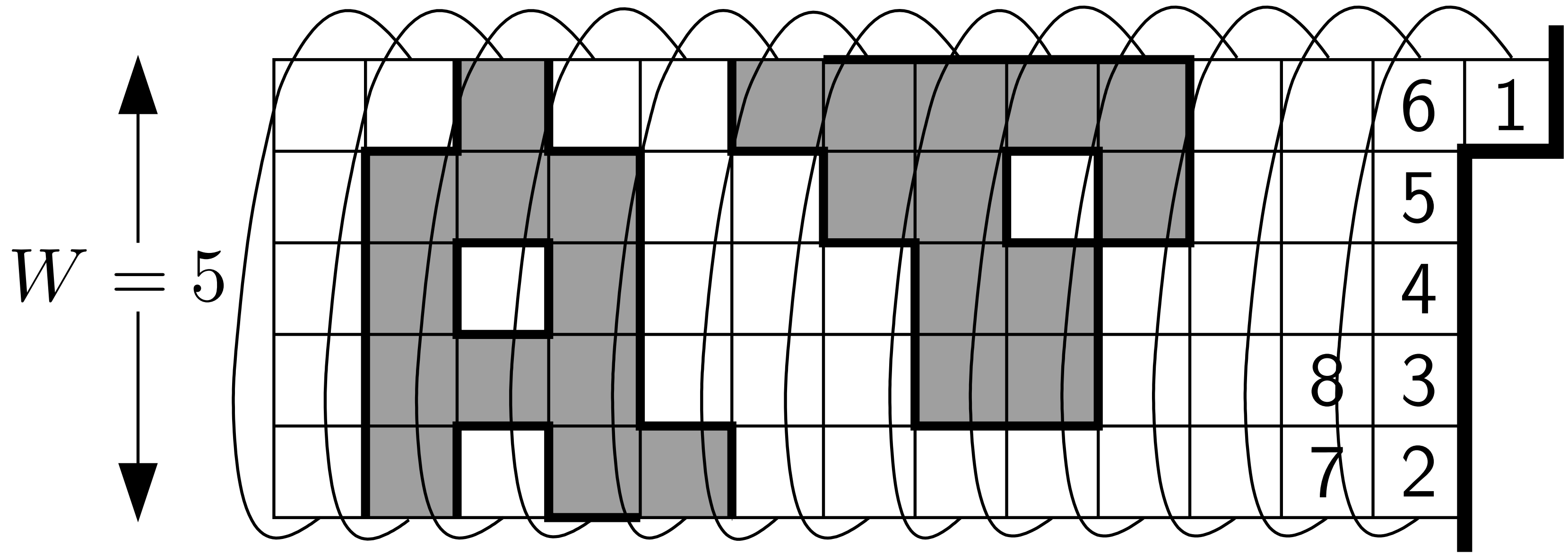
identify point (i, j) with $(i + 1, j + W)$ on the integer grid $\mathbb{Z} \times \mathbb{Z}$

A twisted cylinder of width W



identify point (i, j) with $(i + 1, j + W)$ on the integer grid $\mathbb{Z} \times \mathbb{Z}$

A twisted cylinder of width W



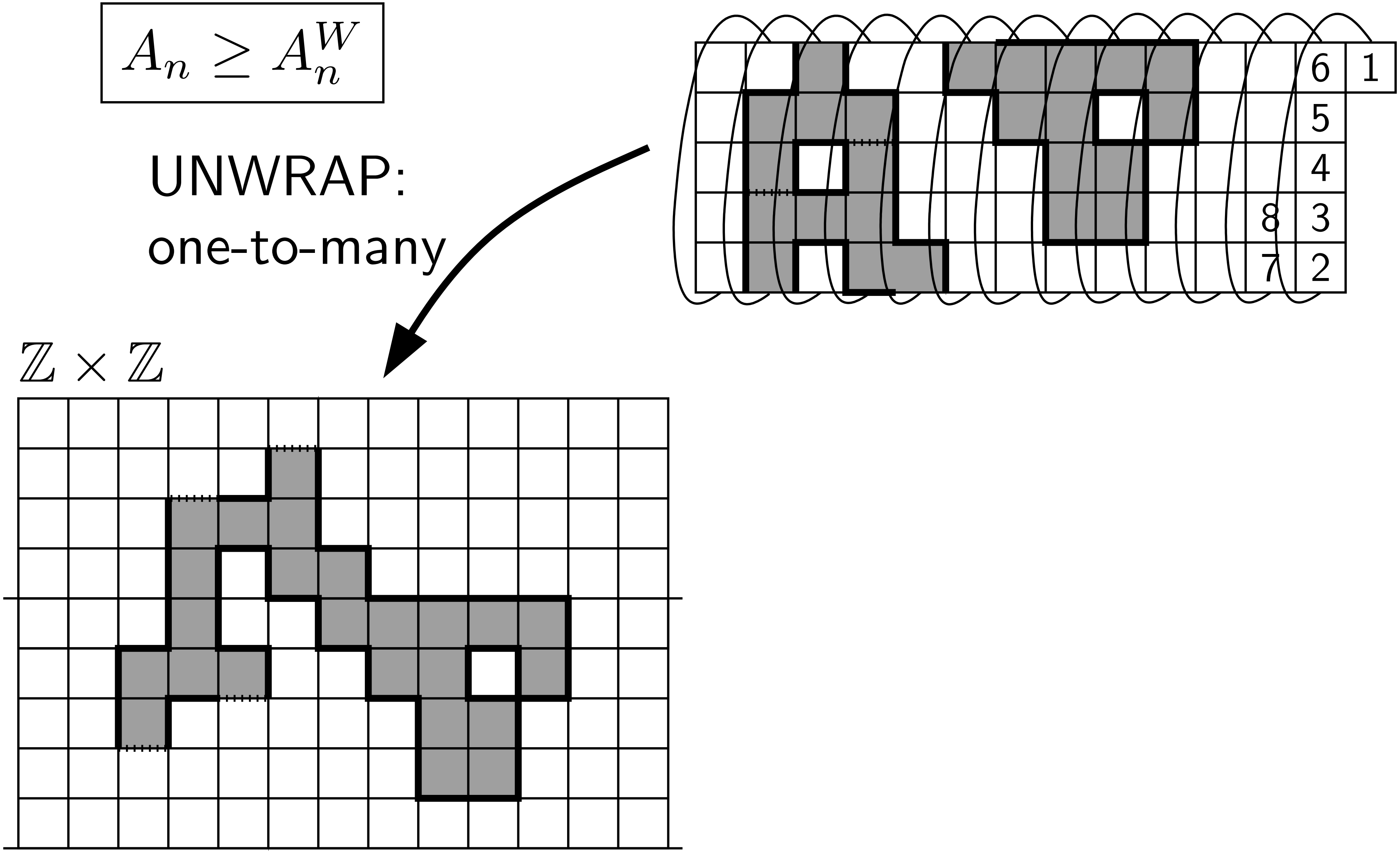
identify point (i, j) with $(i + 1, j + W)$ on the integer grid $\mathbb{Z} \times \mathbb{Z}$

$A_n^W :=$ the number of n-ominoes on a twisted cylinder of width W

The plane contains more polyominoes than the twisted cylinder:

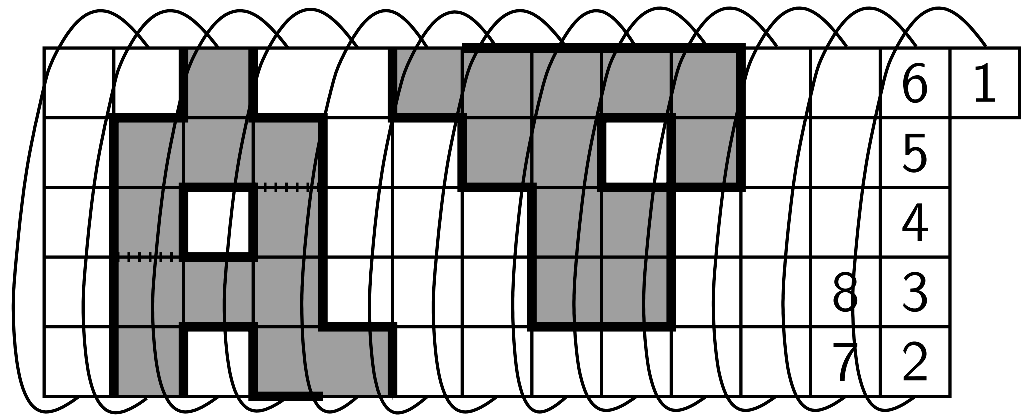
$$\boxed{A_n \geq A_n^W}$$

UNWRAP:
one-to-many



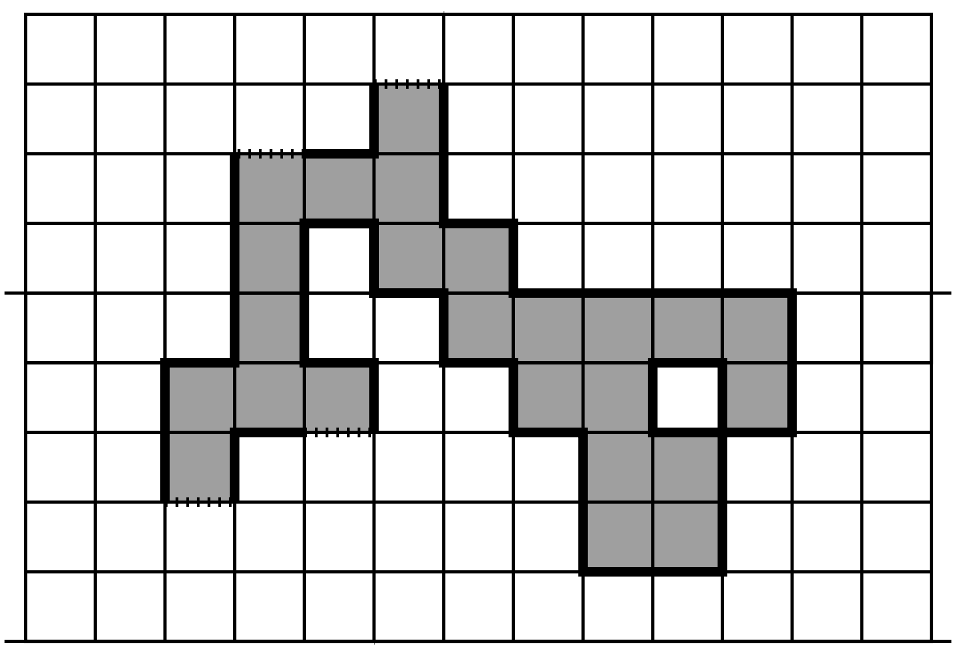
The plane contains more polyominoes than the twisted cylinder:

$$\boxed{A_n \geq A_n^W}$$



UNWRAP:
one-to-many

$\mathbb{Z} \times \mathbb{Z}$



WRAP:
inverse of UNWRAP.

may overlap
$\Longrightarrow$ fewer cells

$$A_n \geq A_n^W$$

$$\lambda^W := \lim_{n \to \infty} \frac{A_{n+1}^W}{A_n^W}$$

λ^W is a lower bound on Klarner's constant λ.

W	λ^W	W	λ^W
3	2.6590	13	3.8775
4	3.0609	14	3.8973
5	3.3141	15	3.9139
6	3.4809	16	3.9279
7	3.5961	17	3.9399
8	3.6787	18	3.9502
9	3.7402	19	3.9592
10	3.7872	20	3.9671*
11	3.8241	21	3.9740
12	3.8535	22	3.9801

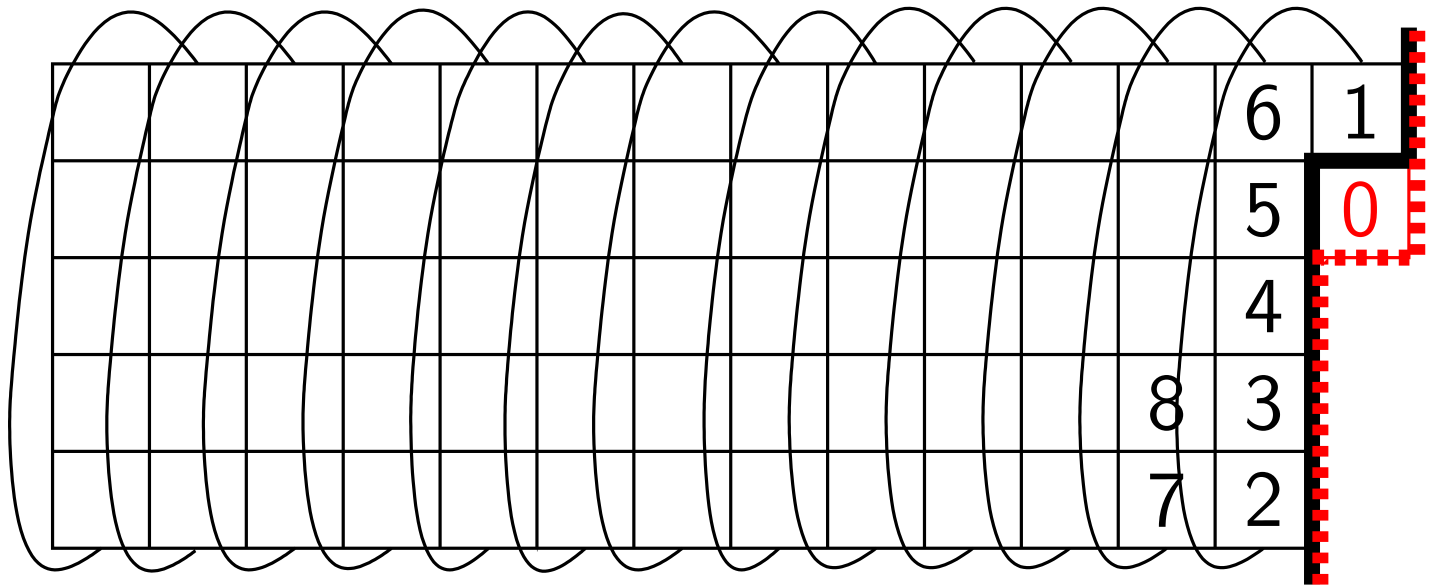
* independently "certified" with MAPLE for $W \leq 20$:

$\lambda^{20} \geq \frac{348\,080}{87\,743} > 3.96704$

Klarner's constant $\lambda \geq \lambda^W > 3.9801$.

The transfer matrix method

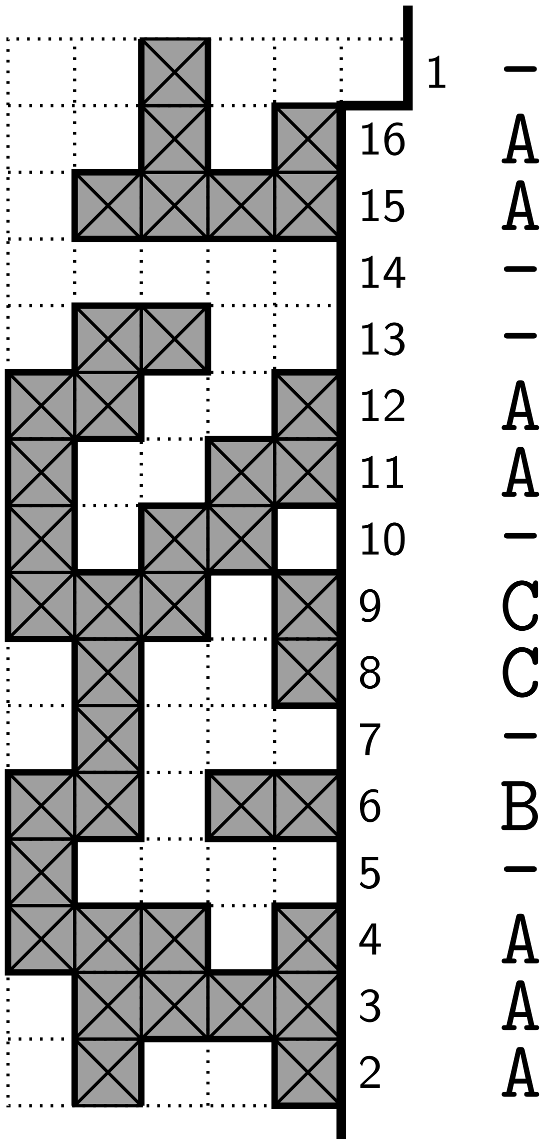
a.k.a. dynamic programming



Build up the cylinder one cell at a time.

Retain information about connectivity of *partial polyomino* = “state” of partial polyomino

The *state* of a partial polyomino



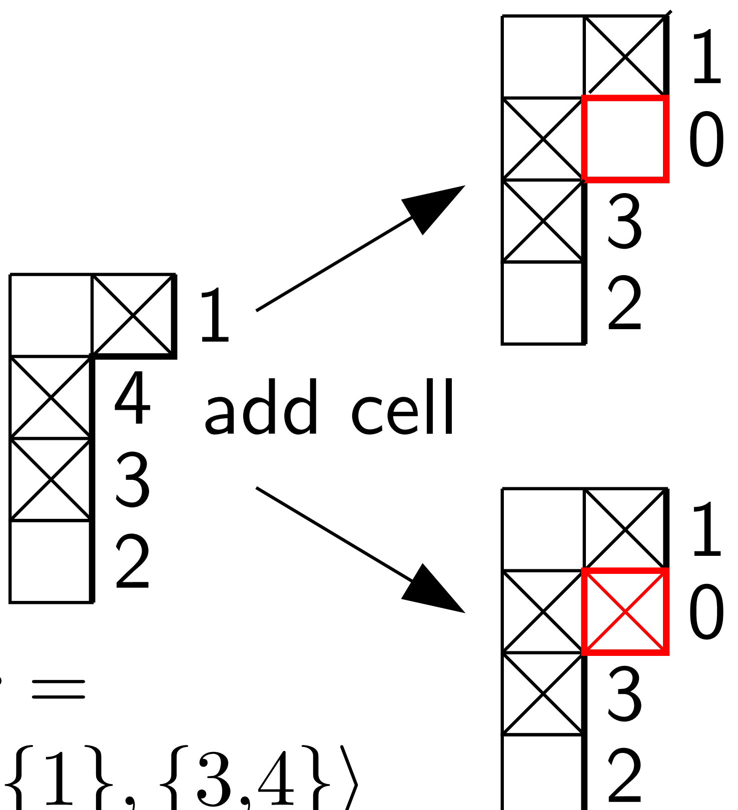
state

$$= \boxed{\texttt{-AAA-B-CC-AA--AA}}$$

$$= \langle \{2,3,4,11,12,15,16\}, \{6\}, \{8,9\} \rangle$$

Adding a cell

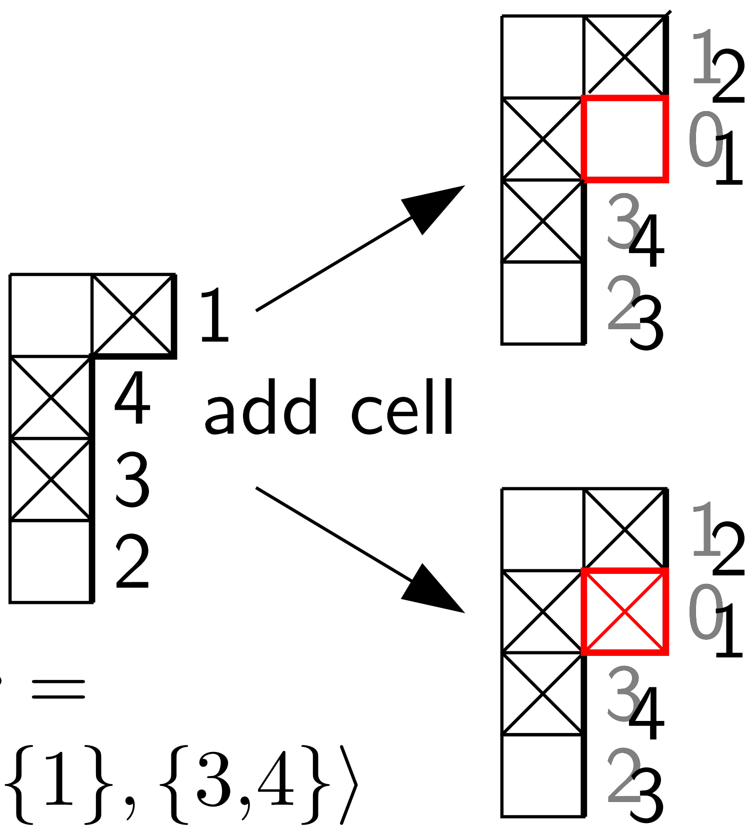
Every state has two *successor states*: $succ_0$ and $succ_1$.



$s =$
$\langle \{1\}, \{3,4\} \rangle$

Adding a cell

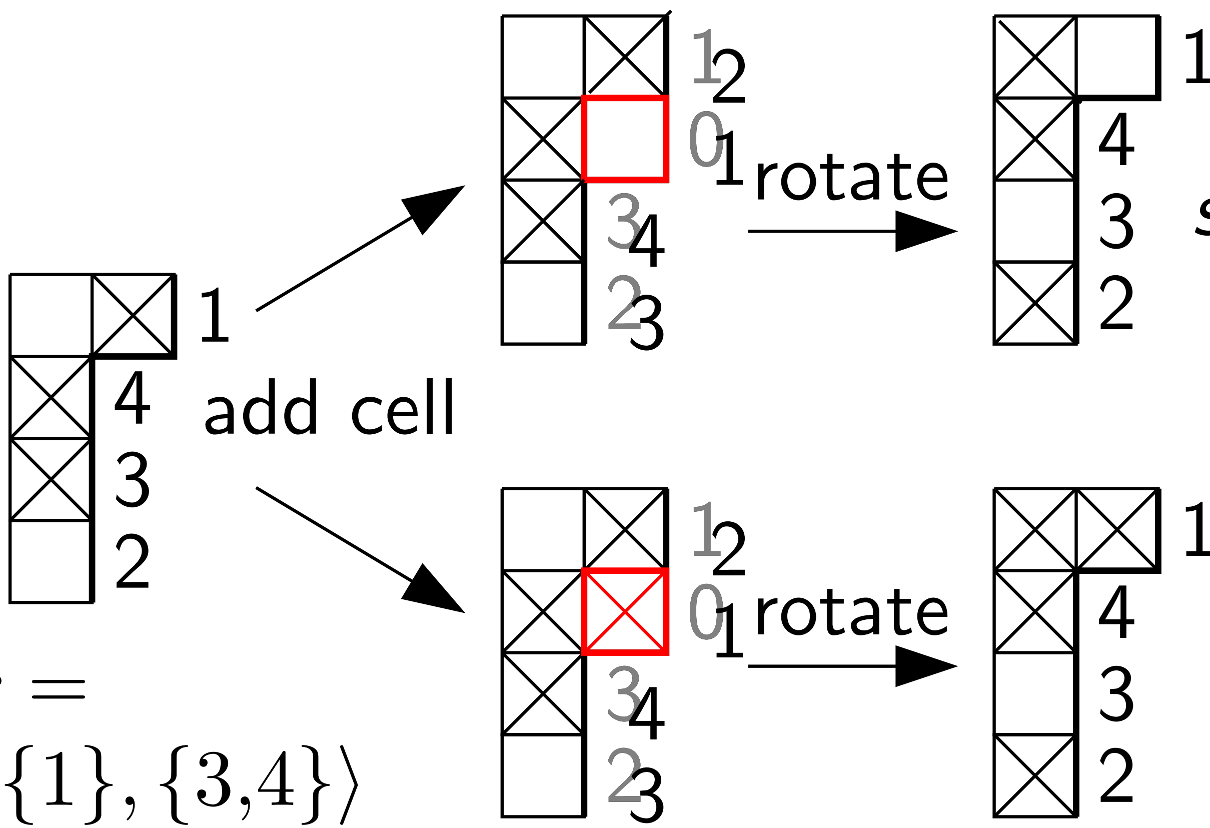
Every state has two *successor states*: $succ_0$ and $succ_1$.



$s = \langle\{1\}, \{3,4\}\rangle$

Adding a cell

Every state has two *successor states*: $succ_0$ and $succ_1$.



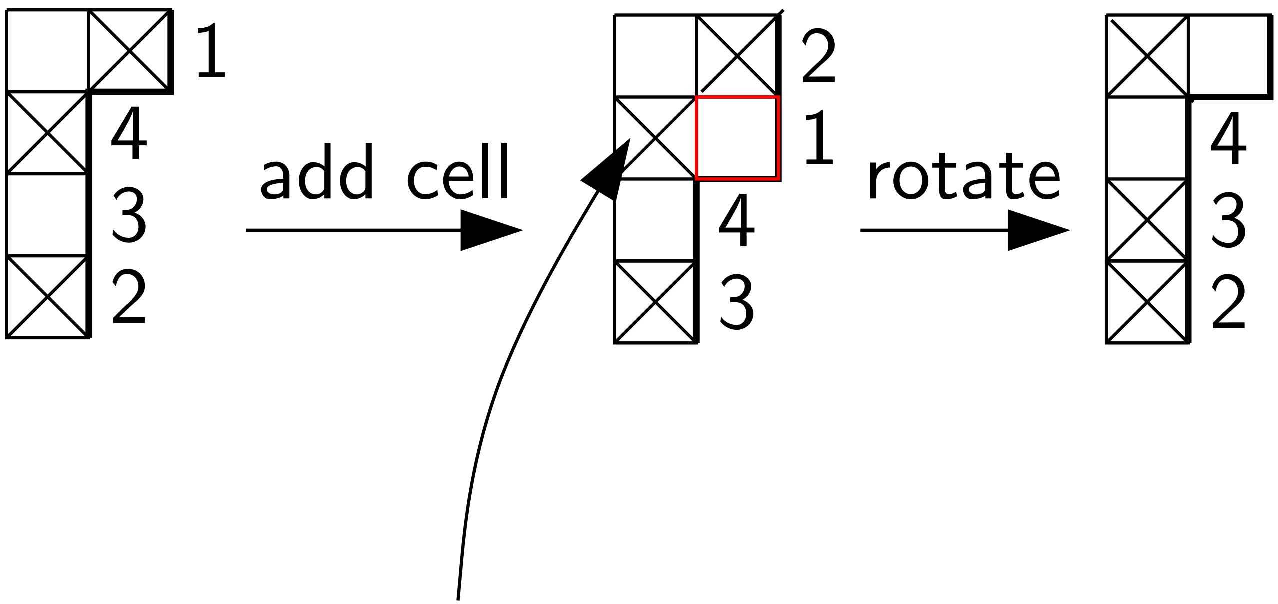
$succ_{\color{red}0}(s) = \langle\{2\}, \{4\}\rangle$

$succ_{\color{red}1}(s) = \langle\{1,2,4\}\rangle$

$s = \langle\{1\}, \{3,4\}\rangle$

$succ_0$ does not always exist.

$s = \langle \{1, 2\}, \{4\} \rangle$



$succ_0(s) = \emptyset$

This cell is disconnected from the boundary.

The transfer equations

$\mathbf{x}_s^{(i)}$:= The number of partial polyominoes

- with i occupied cells
- in state s

$$\mathbf{x}_{\langle\{W\}\rangle}^{(n)} = A_n^W$$

Recursion:

$$\mathbf{x}_s^{(i+1)} = \sum_{s':s=\mathit{succ}_0(s')} \mathbf{x}_{s'}^{(i+1)} + \sum_{s':s=\mathit{succ}_1(s')} \mathbf{x}_{s'}^{(i)} \quad (*)$$

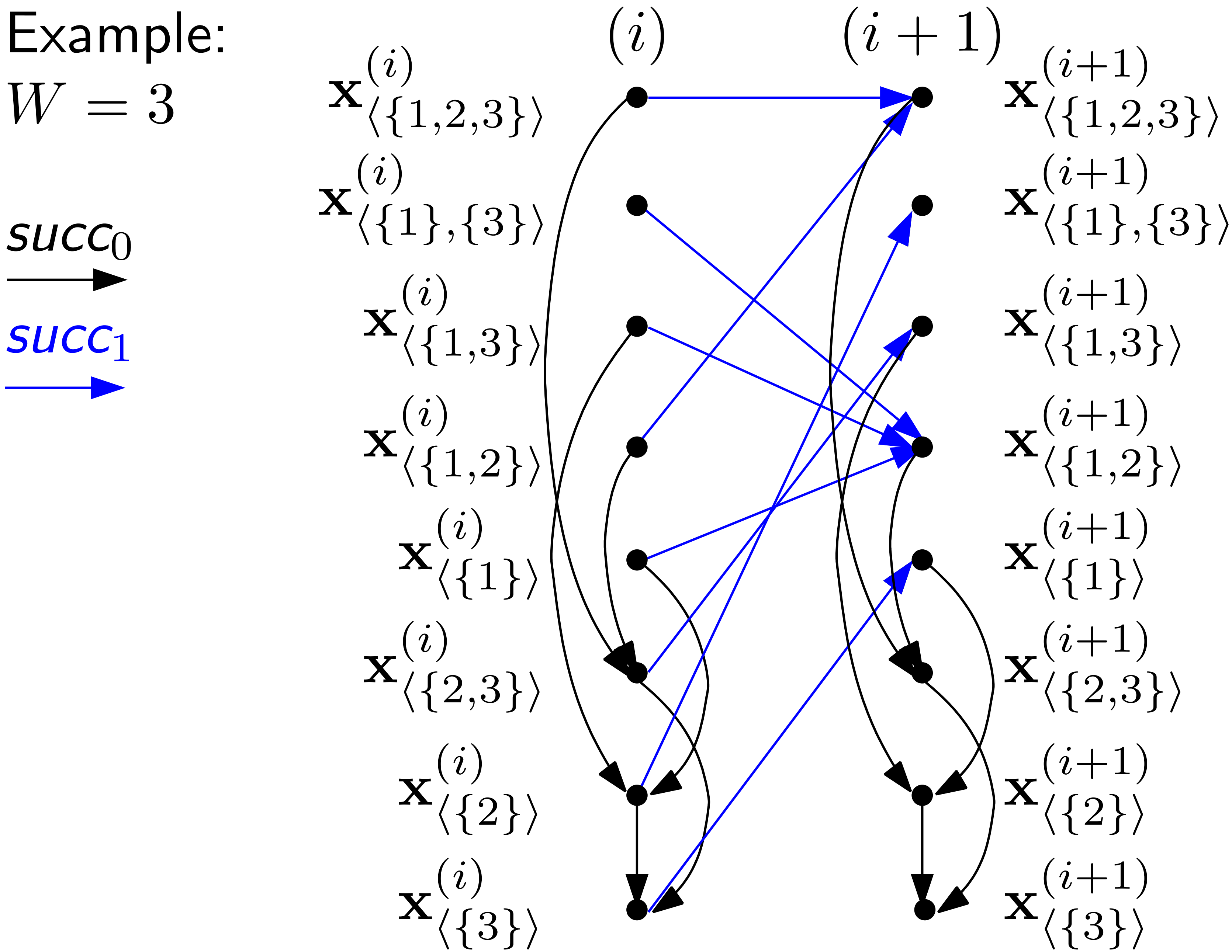
No cyclic dependency

$$\mathbf{x}_s^{(i+1)} = \sum_{s':s=\mathit{succ}_0(s')} \mathbf{x}_{s'}^{(i+1)} + \sum_{s':s=\mathit{succ}_1(s')} \mathbf{x}_{s'}^{(i)} \quad (*)$$

$\mathbf{x}^{(i+1)}$ depends on itself, but there is no cycle in the chain

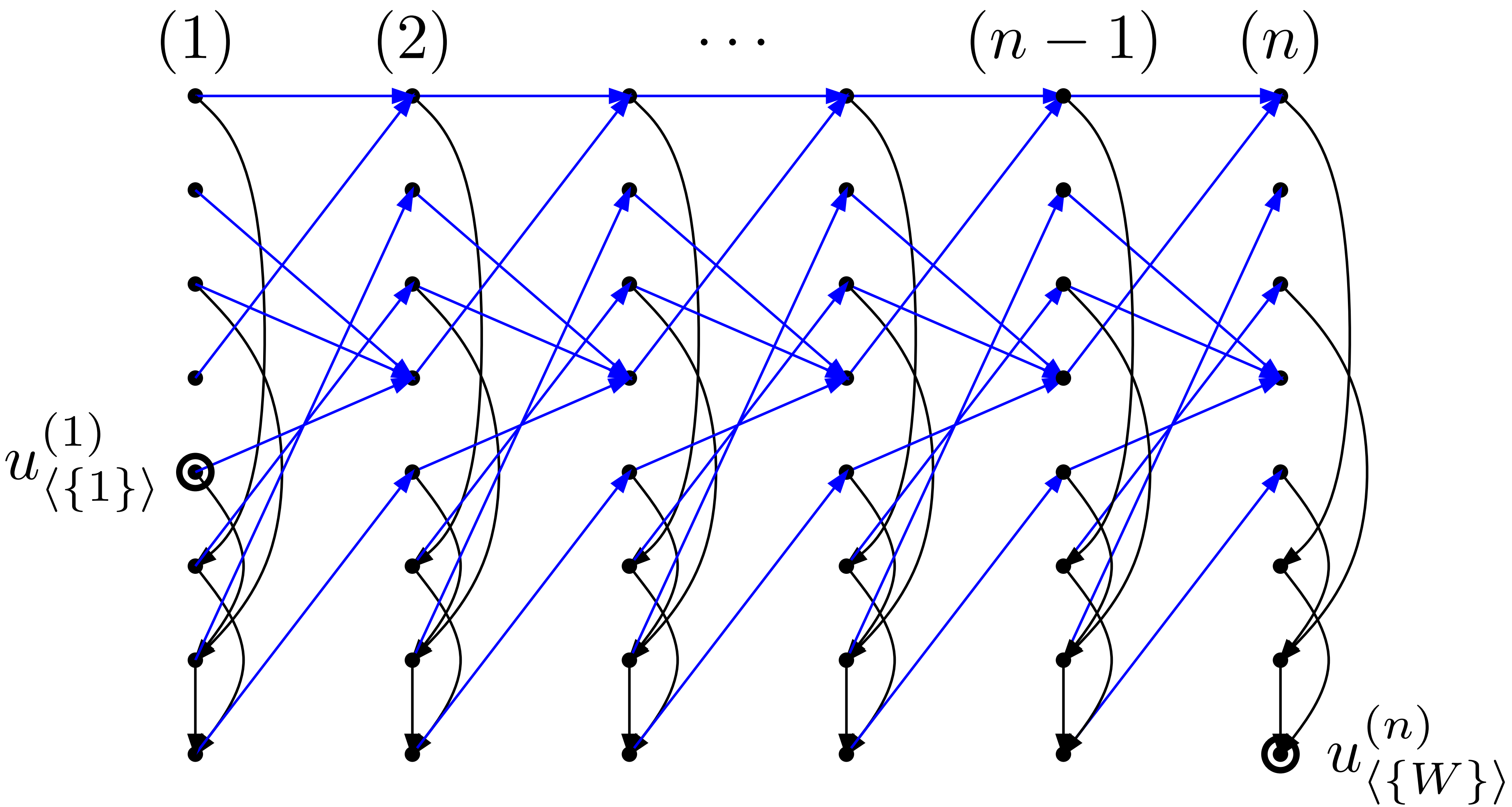
$$s,\ \mathit{succ}_0(s),\ \mathit{succ}_0(\mathit{succ}_0(s)),\ \mathit{succ}_0(\mathit{succ}_0(\mathit{succ}_0(s))),\ \ldots$$

Example:
$W = 3$



$succ_0$
$succ_1$
(i)
$(i+1)$
$\mathbf{x}^{(i)}_{\langle\{1,2,3\}\rangle}$
$\mathbf{x}^{(i)}_{\langle\{1\},\{3\}\rangle}$
$\mathbf{x}^{(i)}_{\langle\{1,3\}\rangle}$
$\mathbf{x}^{(i)}_{\langle\{1,2\}\rangle}$
$\mathbf{x}^{(i)}_{\langle\{1\}\rangle}$
$\mathbf{x}^{(i)}_{\langle\{2,3\}\rangle}$
$\mathbf{x}^{(i)}_{\langle\{2\}\rangle}$
$\mathbf{x}^{(i)}_{\langle\{3\}\rangle}$
$\mathbf{x}^{(i+1)}_{\langle\{1,2,3\}\rangle}$
$\mathbf{x}^{(i+1)}_{\langle\{1\},\{3\}\rangle}$
$\mathbf{x}^{(i+1)}_{\langle\{1,3\}\rangle}$
$\mathbf{x}^{(i+1)}_{\langle\{1,2\}\rangle}$
$\mathbf{x}^{(i+1)}_{\langle\{1\}\rangle}$
$\mathbf{x}^{(i+1)}_{\langle\{2,3\}\rangle}$
$\mathbf{x}^{(i+1)}_{\langle\{2\}\rangle}$
$\mathbf{x}^{(i+1)}_{\langle\{3\}\rangle}$

$$\boxed{\mathbf{x}_s^{(i+1)} := \sum_{s':s=\mathit{succ}_0(s')} \mathbf{x}_{s'}^{(i+1)} + \sum_{s':s=\mathit{succ}_1(s')} \mathbf{x}_{s'}^{(i)} \qquad (*)}$$



$A_n^W = \mathbf{x}^{(n)}_{\langle\{W\}\rangle} =$ the number of paths from $u^{(1)}_{\langle\{1\}\rangle}$ to $u^{(n)}_{\langle\{W\}\rangle}$

Convergence of the iteration

$$\lambda^W = \lim_{n\to\infty} \frac{A^W_{n+1}}{A^W_n} = \lim_{n\to\infty} \frac{\mathbf{x}_s^{(n+1)}}{\mathbf{x}_s^{(n)}},$$

for any state s.

λ^W is the Perron-Frobenius eigenvalue of the iteration

$$\mathbf{x}^{(n)} \mapsto \mathbf{x}^{(n+1)},$$

and $\mathbf{x}^{(n)}$ converges to the corresponding eigenvector.

Convergence of the iteration

$$\lambda^W = \lim_{n\to\infty} \frac{A^W_{n+1}}{A^W_n} = \lim_{n\to\infty} \frac{\mathbf{x}_s^{(n+1)}}{\mathbf{x}_s^{(n)}},$$

for any state s.

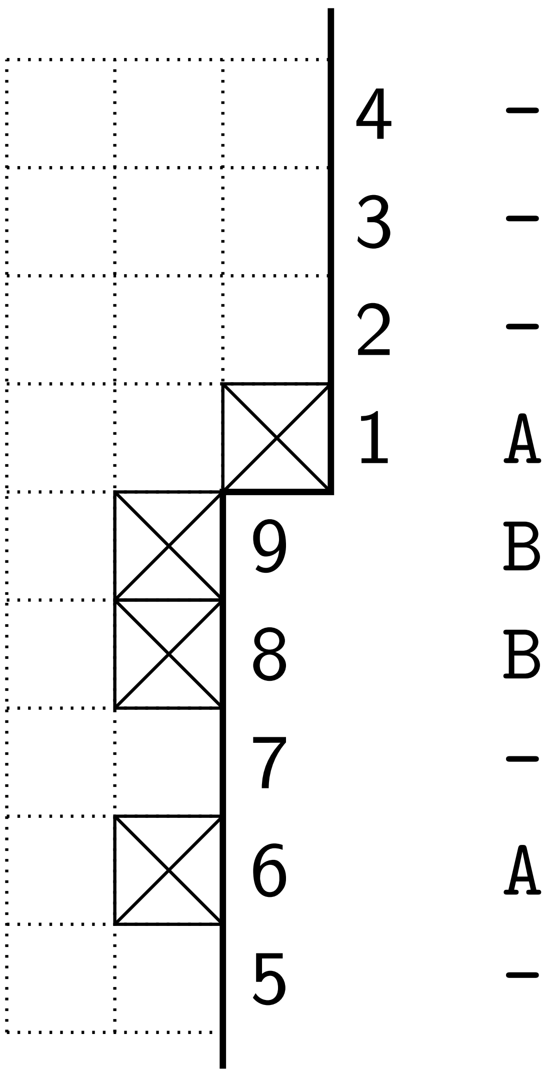
λ^W is the Perron-Frobenius eigenvalue of the iteration

$$\mathbf{x}^{(n)} \mapsto \mathbf{x}^{(n+1)},$$

and $\mathbf{x}^{(n)}$ converges to the corresponding eigenvector.

[Some states are not successors of any state.]

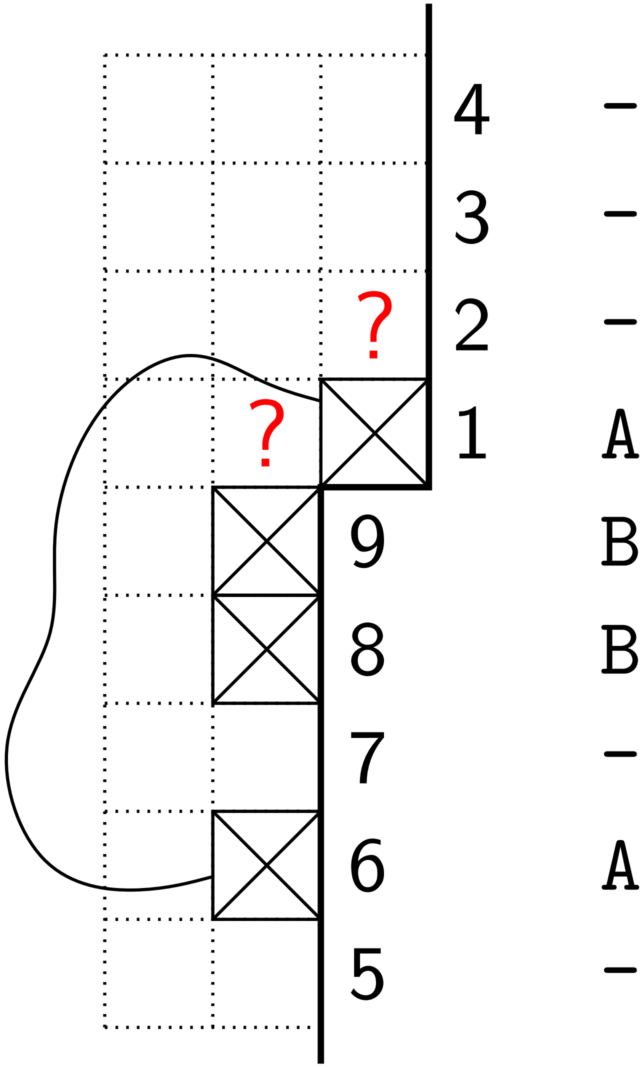
Some states are not successors of any state



$$\boxed{\texttt{A----A-BB}} = \langle \{1,6\}, \{8,9\} \rangle$$

After removing states without predecessor, the “successor graph” is strongly connected, and it is aperiodic.

Some states are not successors of any state



$\boxed{\texttt{A----A-BB}} = \langle\{1,6\},\{8,9\}\rangle$

After removing states without predecessor, the “successor graph” is strongly connected, and it is aperiodic.

Forward and backward recursion

The forward iteration:

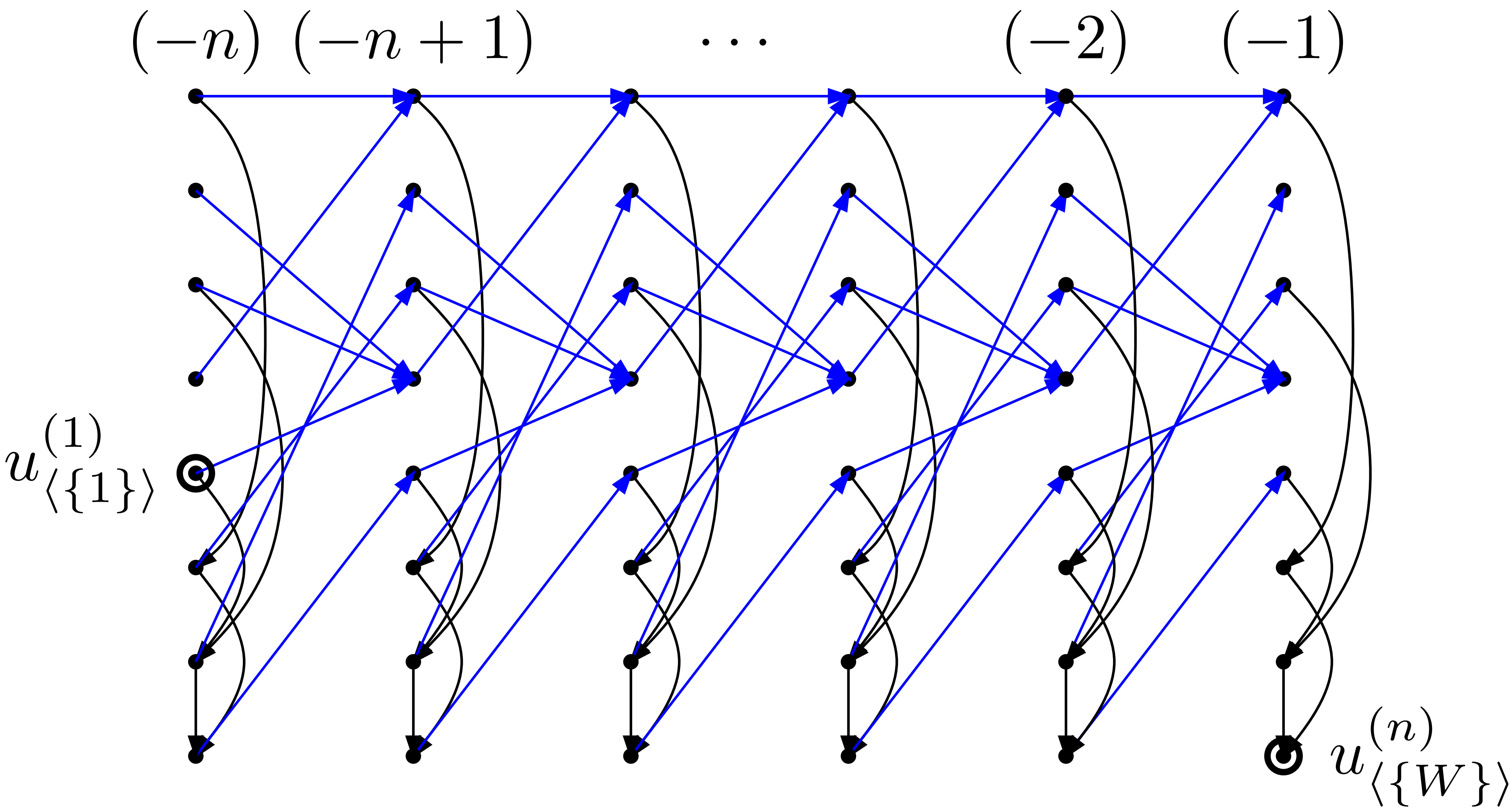
$$\mathbf{x}_s^{(i+1)} := \sum_{s' : s = \mathit{succ}_0(s')} \mathbf{x}_{s'}^{(i+1)} + \sum_{s' : s = \mathit{succ}_1(s')} \mathbf{x}_{s'}^{(i)} \qquad (*)$$

The backward iteration:

$$\mathbf{y}_s^{(i-1)} := \mathbf{y}_{\mathit{succ}_0(s)}^{(i-1)} + \mathbf{y}_{\mathit{succ}_1(s)}^{(i)}$$

(Omit $\mathbf{y}_{\mathit{succ}_0(s)}^{(i-1)}$ if $\mathit{succ}_0(s)$ does not exist.)

$$\boxed{\mathbf{y}_s^{(i-1)} := \mathbf{y}_{\mathit{succ}_0(s)}^{(i-1)} + \mathbf{y}_{\mathit{succ}_1(s)}^{(i)}}$$



$A_n^W = \mathbf{y}^{(-n)}_{\langle\{1\}\rangle} =$ the number of paths from $u^{(1)}_{\langle\{1\}\rangle}$ to $u^{(n)}_{\langle\{W\}\rangle}$

Convergence of the iteration

$$\lambda^W = \lim_{n\to\infty} \frac{\mathbf{y}_s^{(-(n+1))}}{\mathbf{y}_s^{(-n)}}, \text{ for any state } s.$$

λ^W is the Perron-Frobenius eigenvalue of the iteration

$$\mathbf{y}^{(-n)} \mapsto \mathbf{y}^{(-(n+1))},$$

and $\mathbf{y}^{(-n)}$ converges to the corresponding eigenvector.

Convergence of the iteration

$$\lambda^W = \lim_{n\to\infty} \frac{\mathbf{y}_s^{(-(n+1))}}{\mathbf{y}_s^{(-n)}}, \text{ for any state } s.$$

Lemma:

$$\min_s \frac{\mathbf{y}_s^{(-(n+1))}}{\mathbf{y}_s^{(-n)}} \leq \lambda^W \leq \max_s \frac{\mathbf{y}_s^{(-(n+1))}}{\mathbf{y}_s^{(-n)}}$$

$\Longrightarrow$ bounds on λ^W from two successive iterates.

The program

succ$_0$, *succ*$_1$ are stored in two arrays.

initialize $y\_old[s] := 1$ for all s (for example)

`while not` convergence

  `for` $s := 1$ `to` M

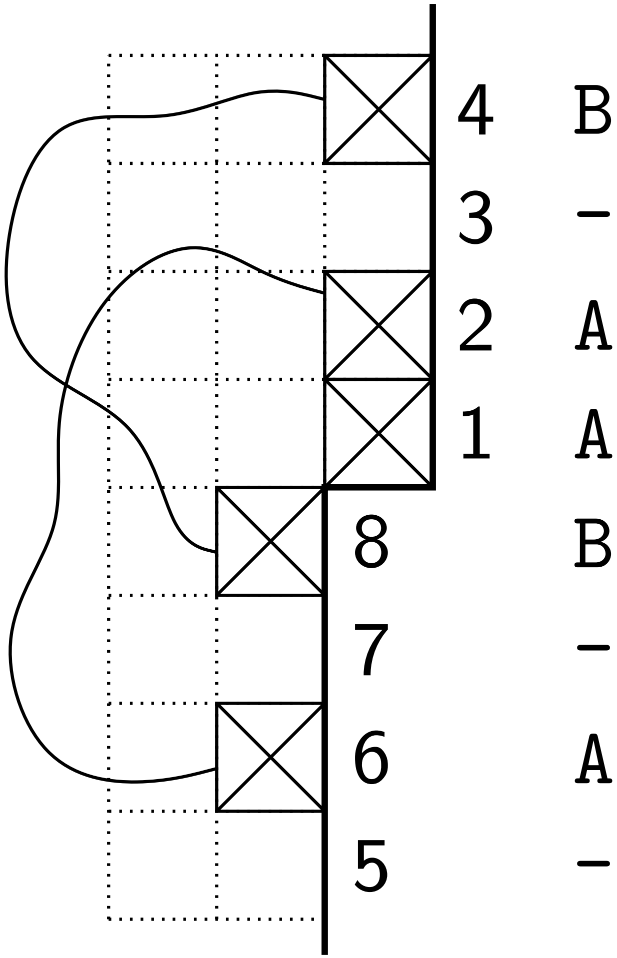
    `if` $succ0[s] \neq 0$
    `then` $y\_new[s] := y\_old[succ1[s]] + y\_new[succ0[s]]$
    `else` $y\_new[s] := y\_old[succ1[s]]$

  $y\_old := y\_new$

"state" s is an integer between 1 and $M = M_{W+1}$

How to represent states

A state is a family of disjoint subsets of $\{1, 2, \ldots, W\}$ with two properties:



- non-crossing:
  The pattern

  $\boxed{\texttt{...A...B...A...B...}}$

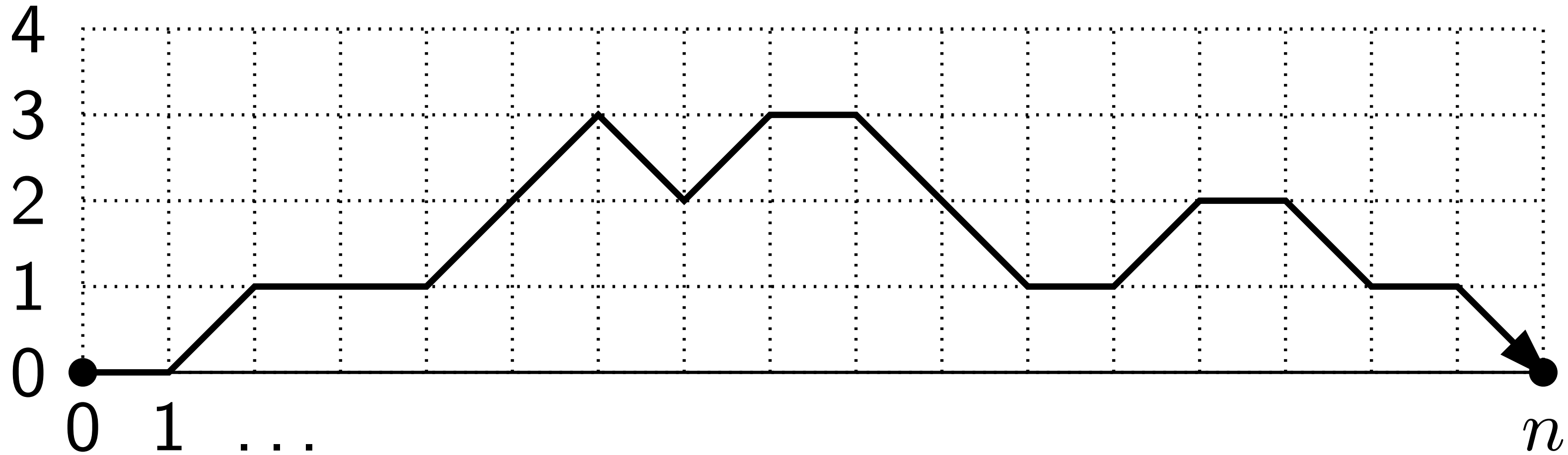
  is forbidden.

- Adjacent occupied cells belong to the same block.

How to represent states

$\mathbf{y}$ is a vector whose entries are indexed by states s.

Use a bijection between states and *Motzkin paths*.



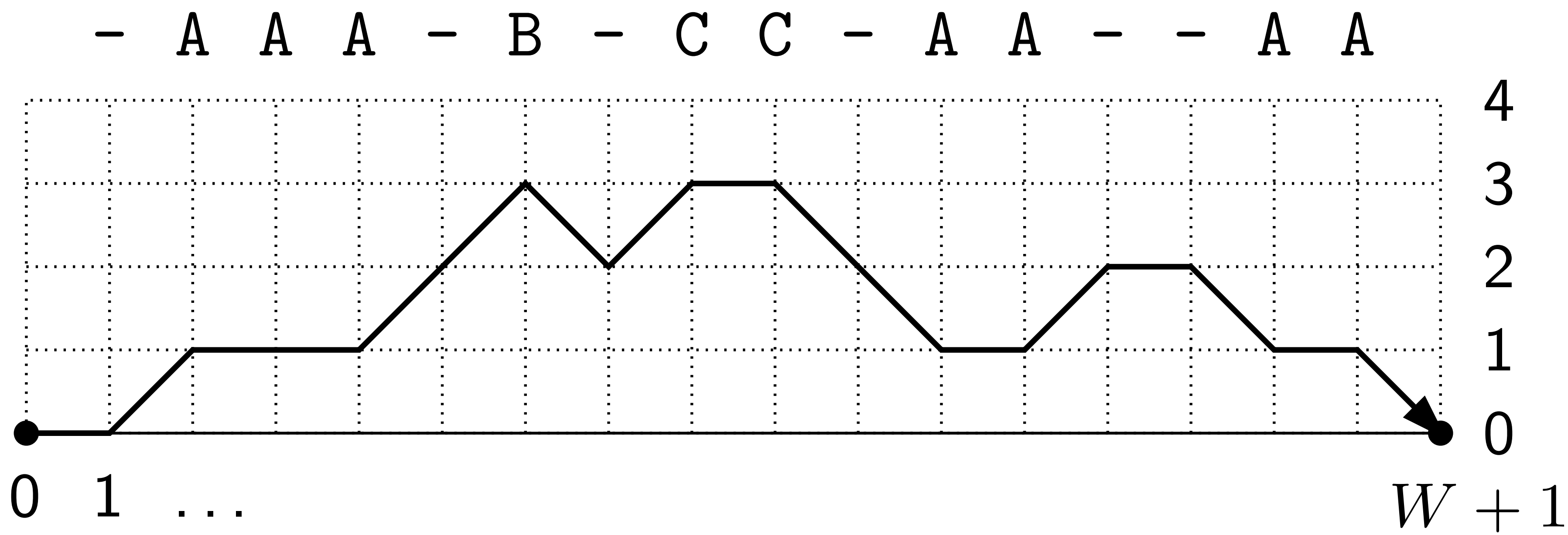
n steps 0, ± 1; nonnegative; start and end at 0.

Motzkin numbers $M_n = \frac{3^n}{n^{3/2}} \cdot \sqrt{\frac{27}{4\pi}} \cdot (1 + O(1/n))$

$$M_1, M_2, \ldots = 1, 2, 4, 9, 21, 51, 127, 323, 835, 2188, \ldots.$$

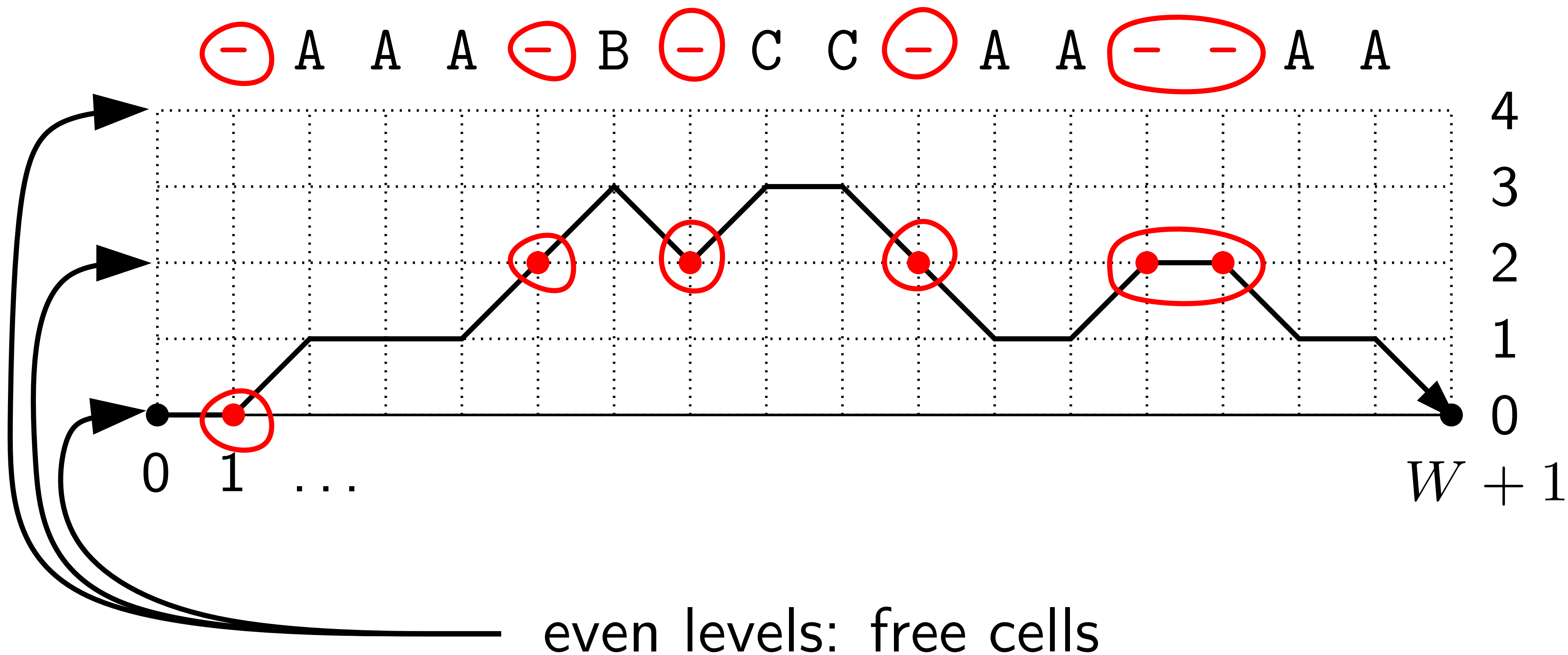
states $\leftrightarrow$ Motzkin paths of length $W+1$

[suggested by Stefan Felsner]



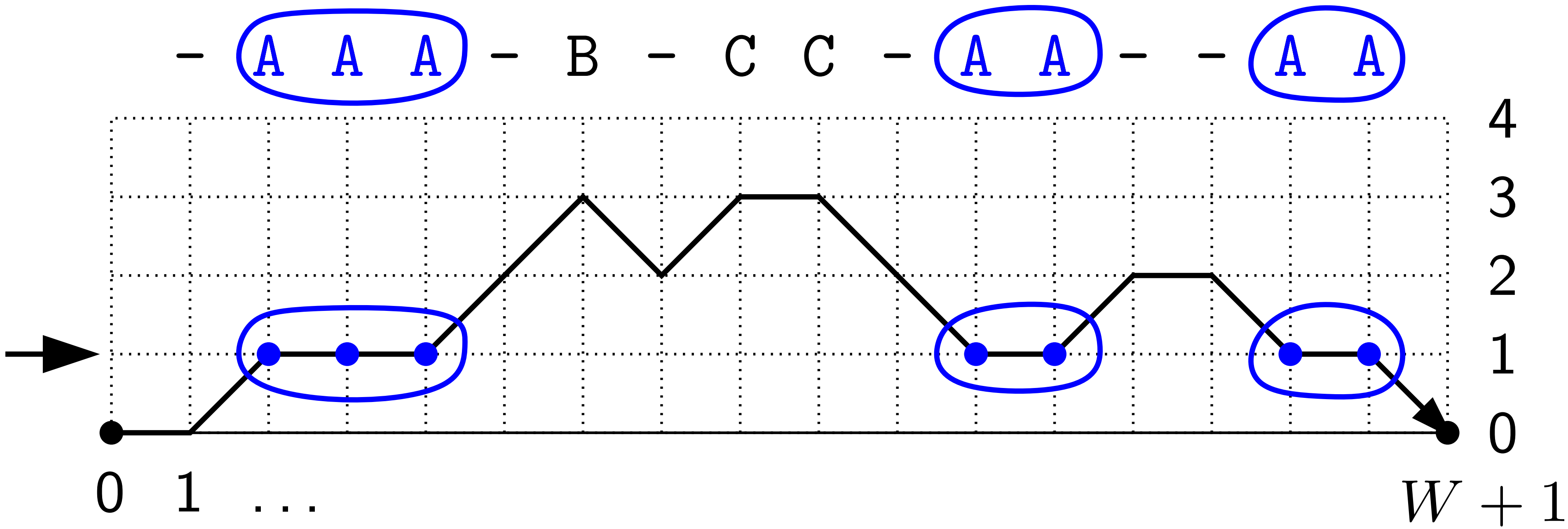
states $\leftrightarrow$ Motzkin paths of length $W+1$

[suggested by Stefan Felsner]



states $\leftrightarrow$ Motzkin paths of length $W+1$

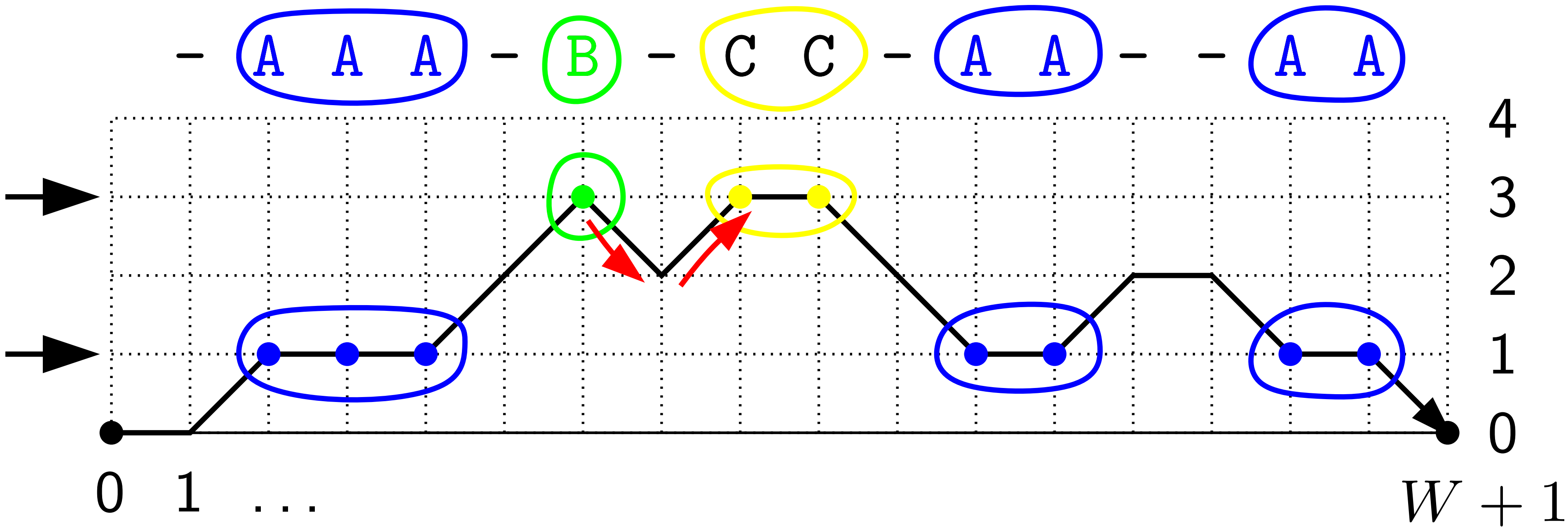
[suggested by Stefan Felsner]



- Cells of one component lie on the same odd level.

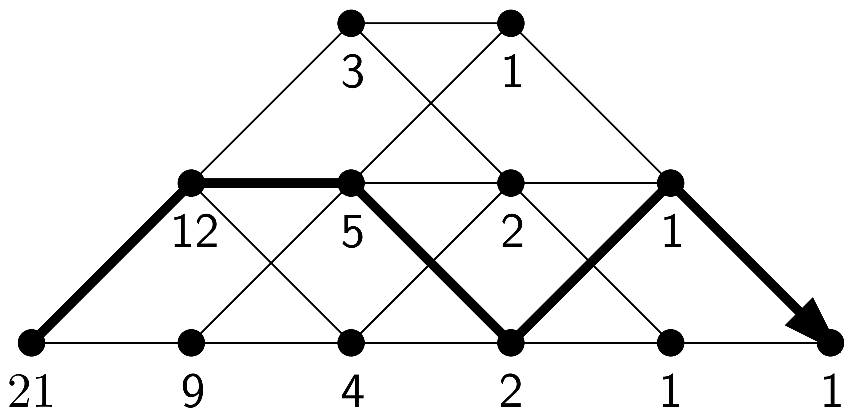
states $\leftrightarrow$ Motzkin paths of length $W + 1$

[suggested by Stefan Felsner]



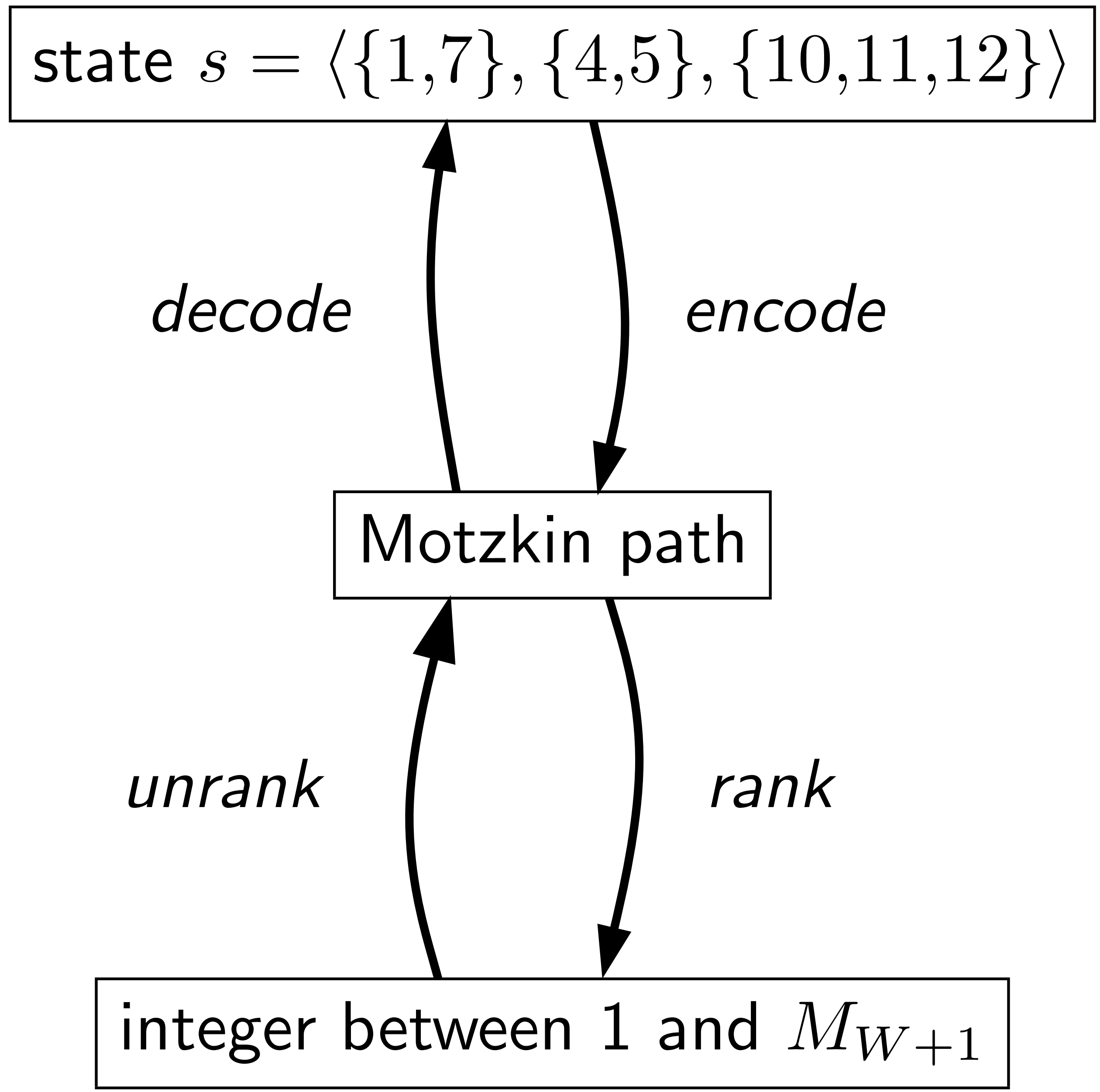
- Cells of one component lie on the same odd level.
- Cells of successive components on the same level are separated by a valley.

Ranking/unranking of Motzkin paths

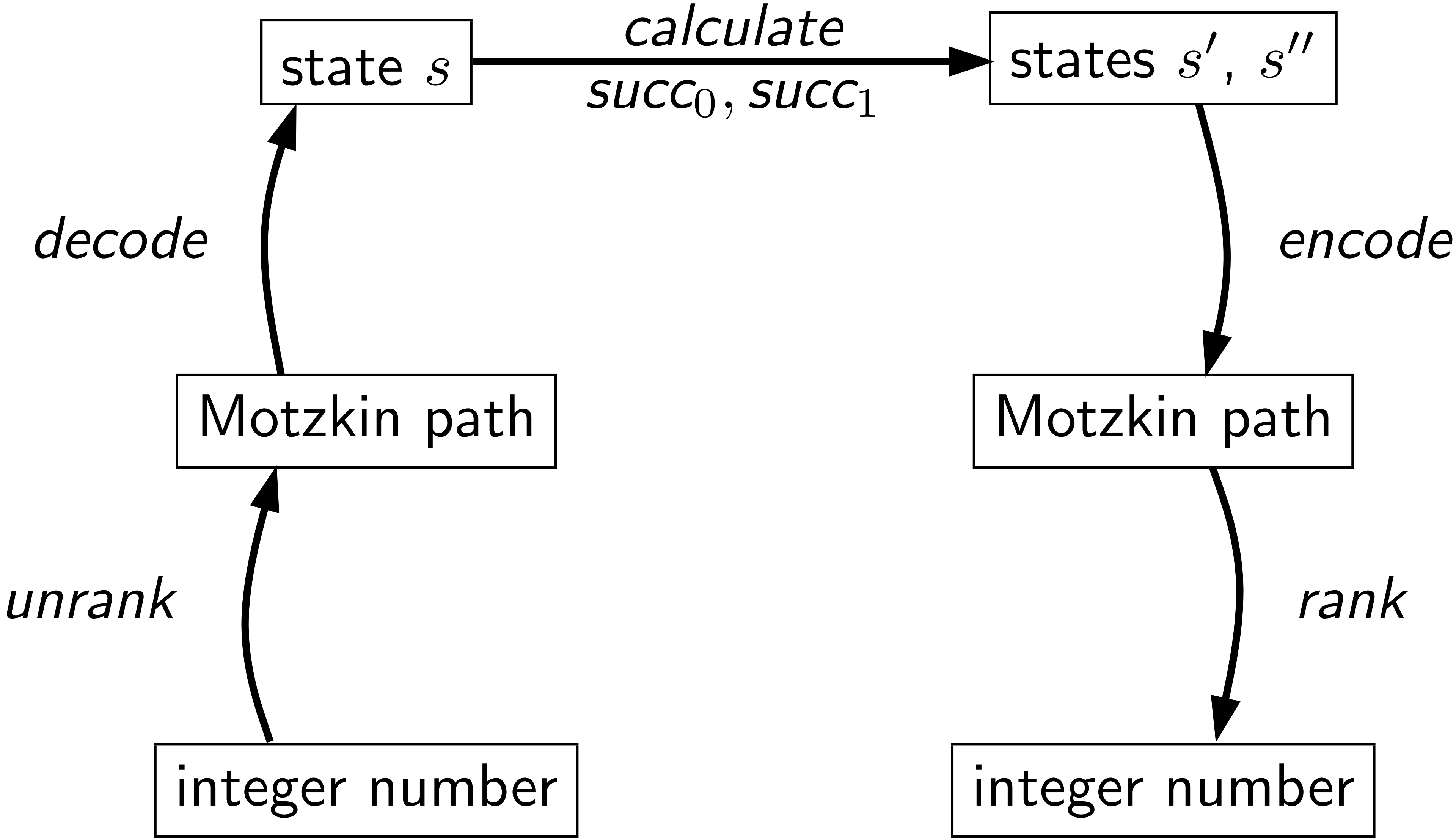


$$+ \quad 0 \quad - \quad + \quad - \qquad \mapsto 9+4+0+1+0=14$$

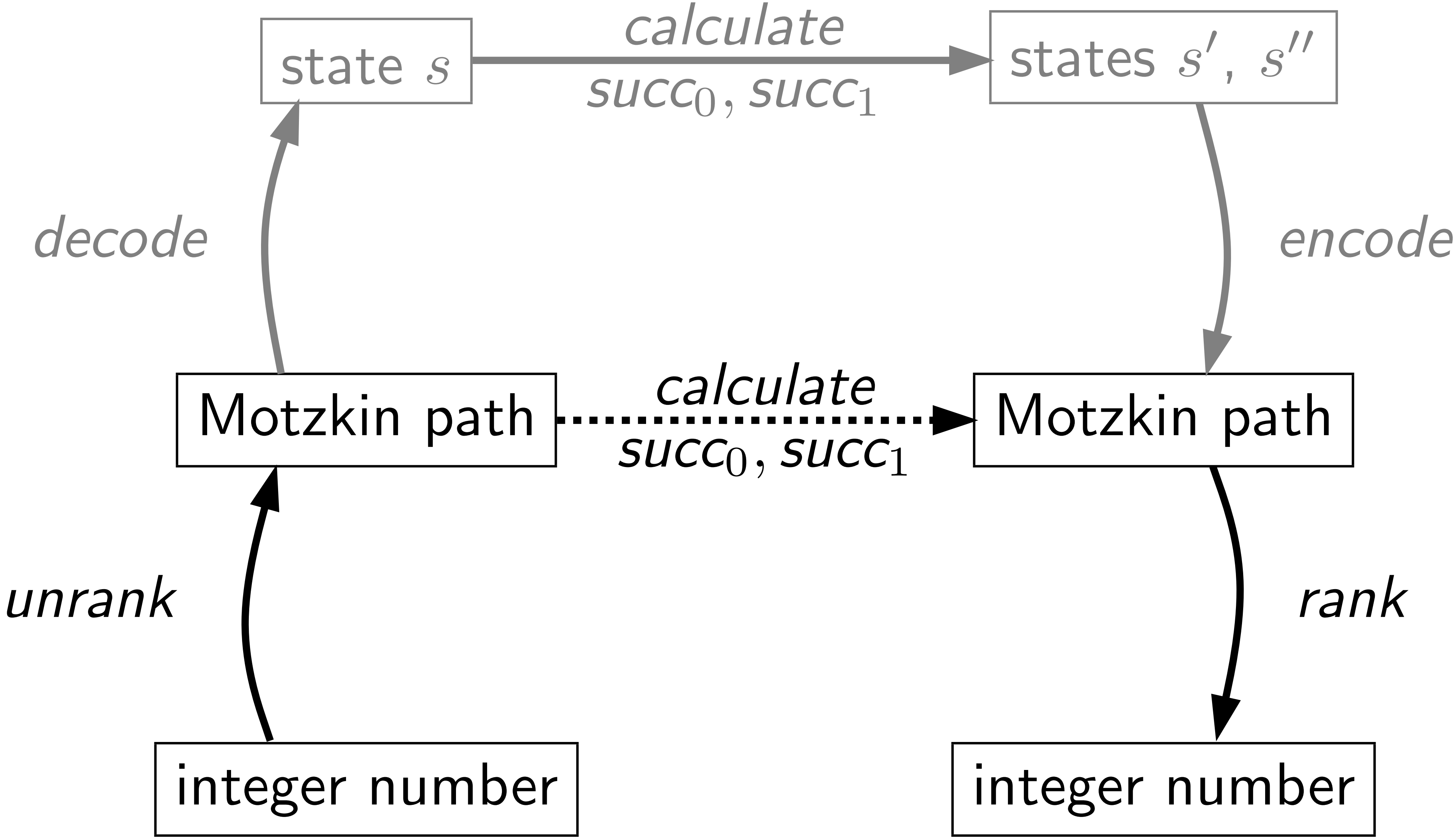
Motzkin path $P \mapsto$ integer between 1 and M_n
$= $ rank of P in lexicographic order


state $s = \langle\{1,7\},\{4,5\},\{10,11,12\}\rangle$
decode
encode
Motzkin path
unrank
rank
integer between 1 and M_{W+1}

Computing successors



Computing successors

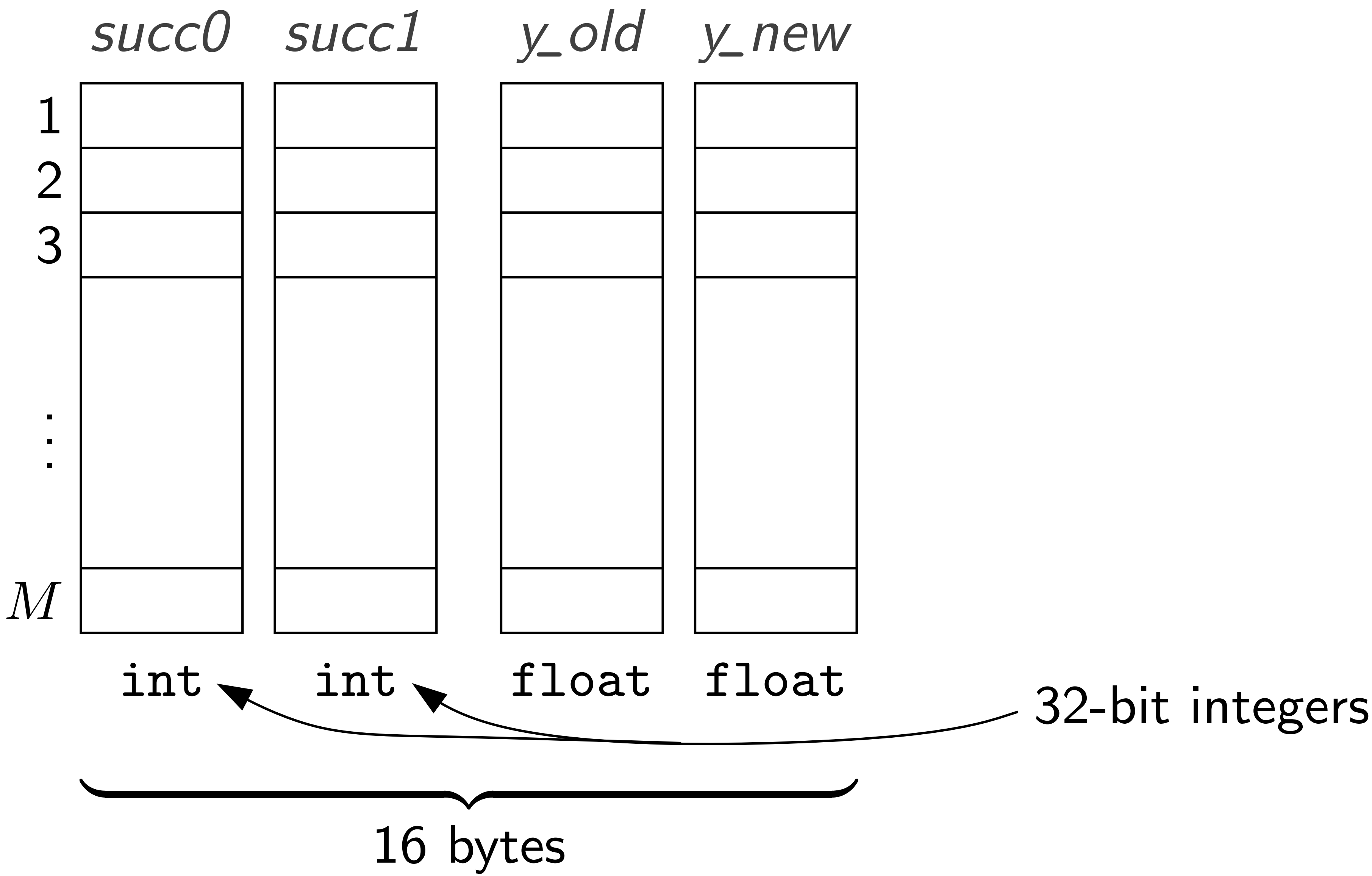


The program

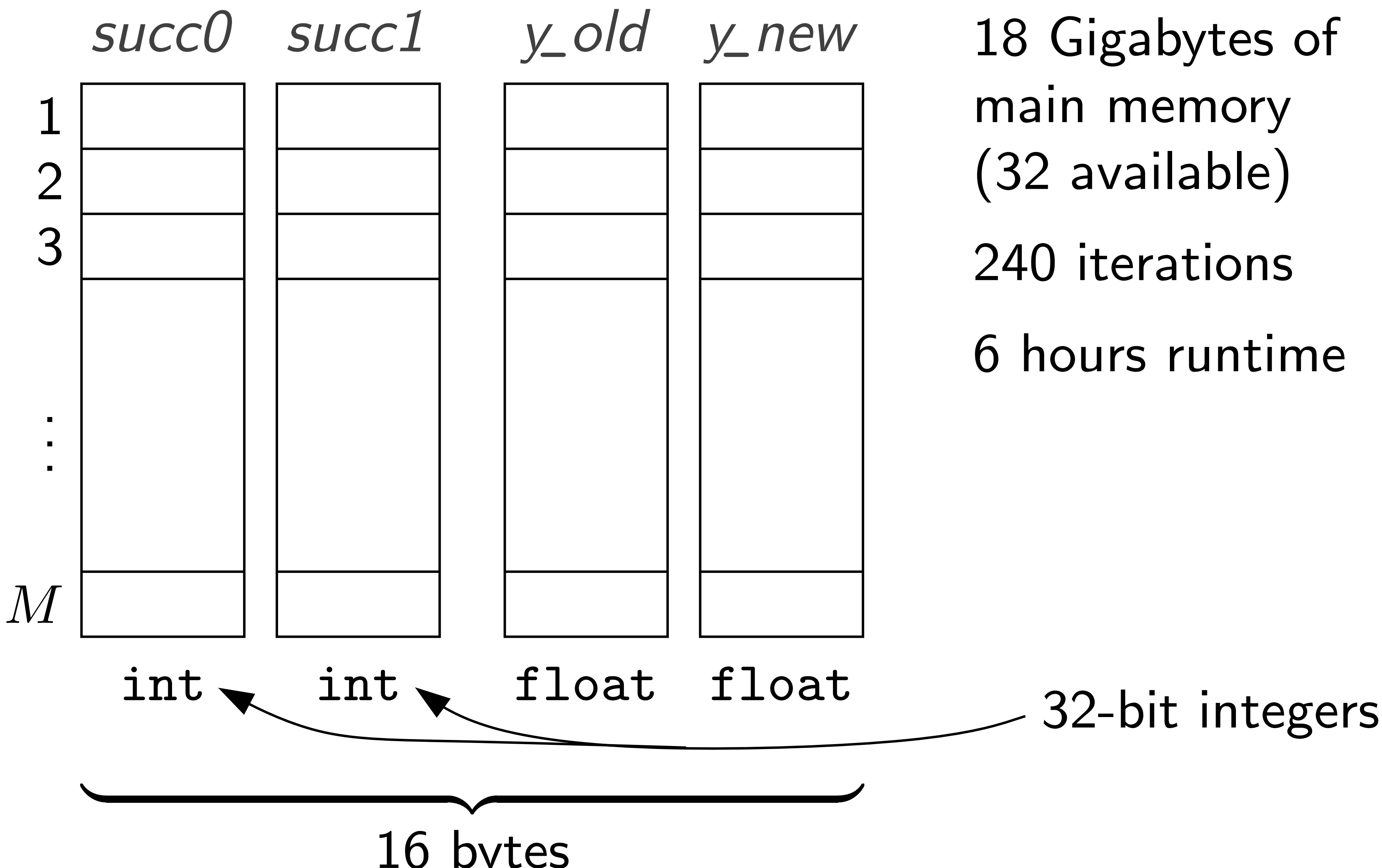
Preprocessing: store $\mathit{succ}_0, \mathit{succ}_1$ in two arrays.

```
initialize y_old[s] := 1 for all s

while not convergence

    for s := 1 to M

        if succ0[s] ≠ 0
        then y_new[s] := y_old[succ1[s]] + y_new[succ0[s]]
        else y_new[s] := y_old[succ1[s]]

    y_old := y_new
```

storage: 4 arrays



storage: 4 arrays



$W = 22$: $M \approx 10^9$

18 Gigabytes of main memory (32 available)

240 iterations

6 hours runtime

Open Questions?

$$\lambda^{W+1} > \lambda^{W}?$$

(may be easy)

$$\lim_{W \to \infty} \lambda^{W} = \lambda \ ?$$