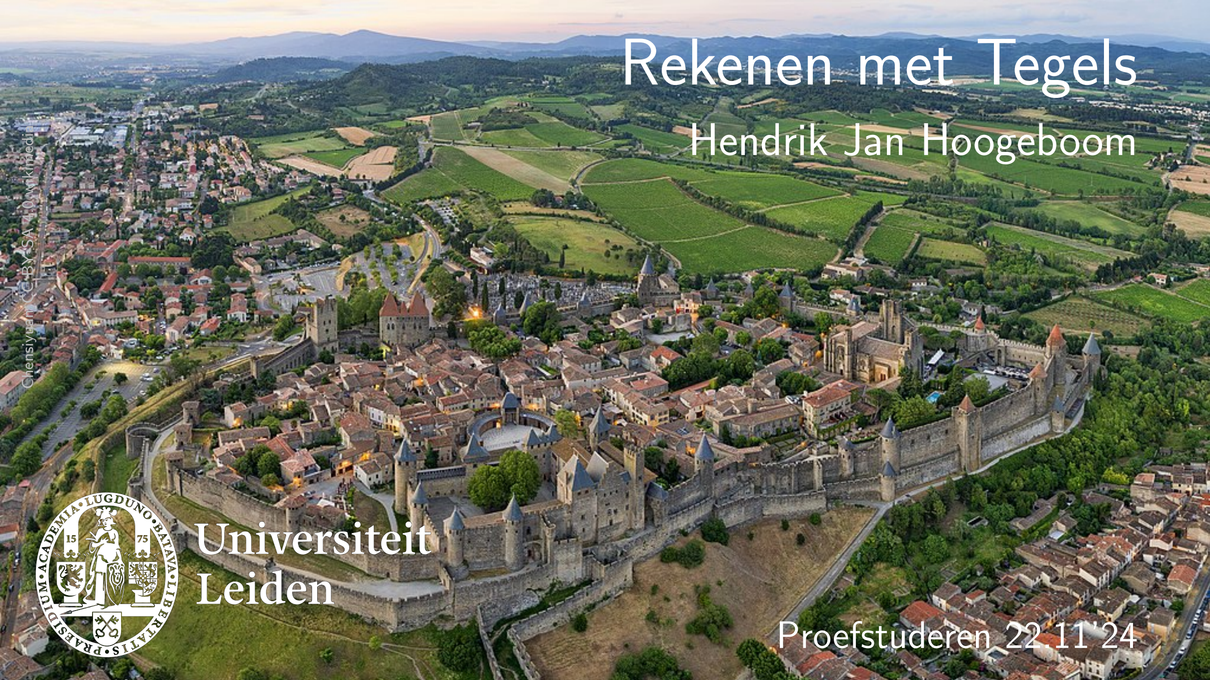
Rekenen met Tegels

Hendrik Jan Hoogeboom



Universiteit Leiden

Proefstuderen 22.11'24

Peter Norvig & Belinda Borman
norvig.com/caracassone.html (2017)

Voorstellen

Bachelor Informatica (Inf) - LIACS - Universiteit Leiden – 2024-2025

liacs Advanced Computer Science | Universiteit Leiden

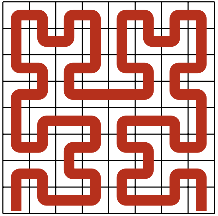
Sklls/Research		Systems/Hardware	Programmlng/Software	Theory	Data/AI	Mathematics
Year 1	Semester 1	**Dlgltal Systems Deslgn**	**Programmeermethoden**	**Foundations of Computer Sclence**	**Studylng and Presentlng** **Orlëntatie Informatica**	**Calculus 1** **Llnear Algebra 1**
	Semester 2	**Programmlng Technlques**	**Algorltmlek**	**Loglc** **Probablllty Theory**	**Databases**	**Llnear Algebra 2** **Calculus 2**
Year 2	Semester 1	**Computerarchltectuur**	**Datastructuren**	**Automata Theory**	**Concepts of Programmlng Languages**	**Statistics**
	Semester 2	**Operating Systems & Networks**	**Securlty**	**Computablllty** **Research Methods In CS**	**Complexlty**	**Artlficlal Intelllgence**
Year 3	Semester 1	Compiler Construction	Human Computer Interaction and Information Visualization		Computer Vision	**30 ECTS Mlnor or 30 ECTS Electives**
		Multiprocessor Programming	Video Game Making	Quantum Computing*	Natural Computing	Natural Language Processing
		Computer Graphics		Data Protection	Internet Governance	
	Semester 2		Requirements Engineering	Program Correctness	Reinforcement Learning	Data Science
		Machlne Learnlng	**Software Englneerlng**	**Bachelorproject**		**AI & Ethlcs**
Extra-curricular		AI & Robotics Challenge	Programmeerwedstrijden*			

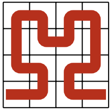
N.B.: Bold courses are mandatory. Courses marked * are not taught every year.
Scheme above is indicative, non-binding and subject to OER changes.

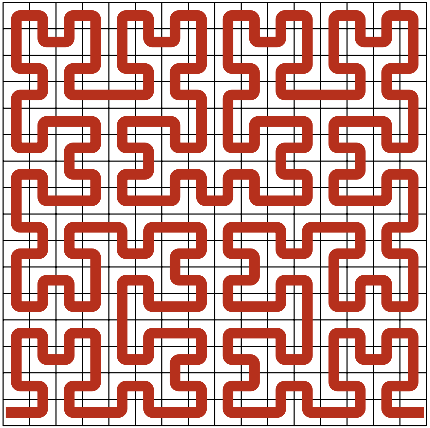
studieprogramma

Motivatie

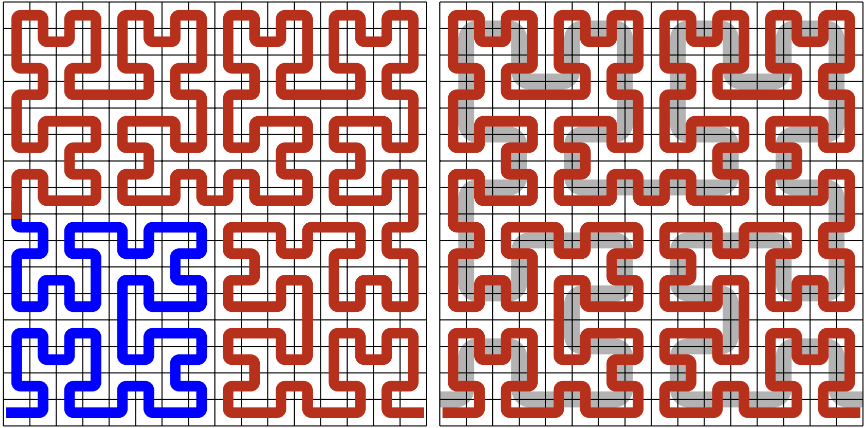
kromme van Hilbert



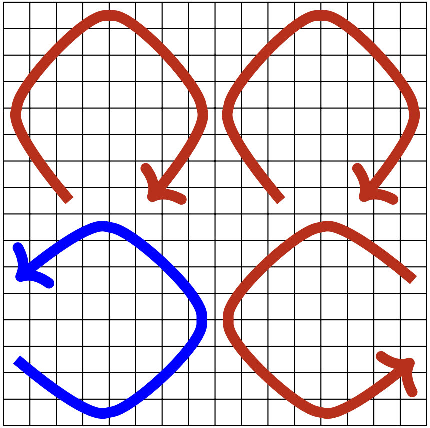




zes ‘tegels’:
twee recht, vier bochten



vier kopieën verfijning

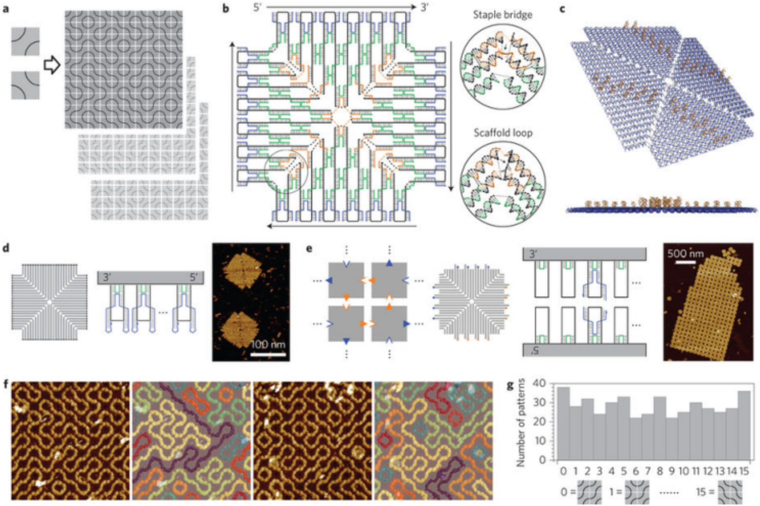



```
to starthilbert :size :level
  hilbert (:size / power 2 (:level-1)) :level 1
end

to hilbert :size :level :parity
  if  :level = 0 [stop]
  right 90 * :parity
  hilbert :size (:level-1) (:parity * -1)
  forward :size
  right -90 * :parity
  hilbert :size (:level-1) :parity
  forward :size
  hilbert :size (:level-1) :parity
  right -90 * :parity
  forward :size
  hilbert :size (:level-1) (:parity * -1)
  right 90 * :parity
end
```

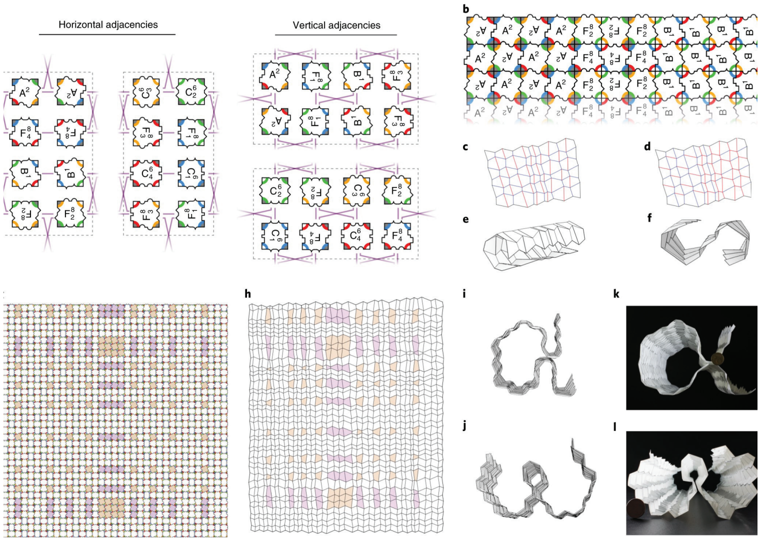
aanroep

```
starthilbert 200 5
```



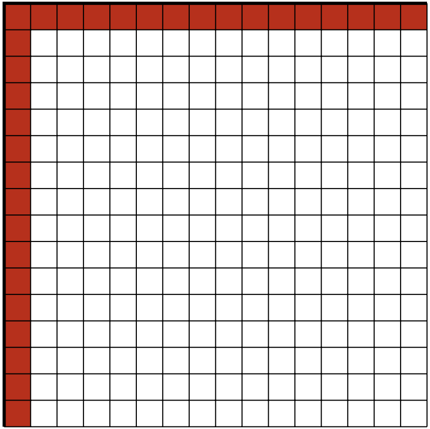
Programmable disorder in random DNA tilings
Tikhomirov, Petersen & Qian – Nature Nanotechnology (2017)

en in Leiden . . .

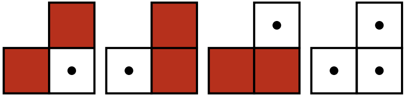


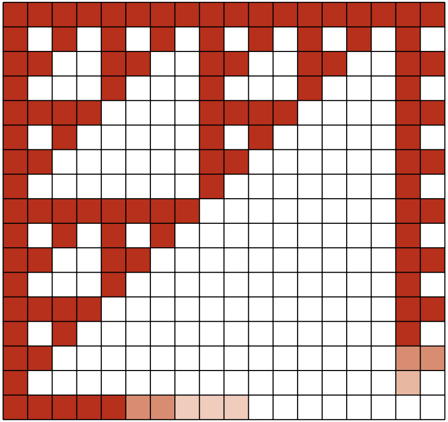
Jigsaw puzzle design of pluripotent origami
Dieleman, Vasmel, Waitukaitis & Van Hecke – Nature Physics (2019)

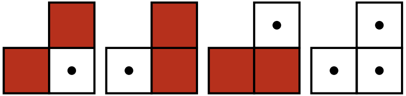
Simpele regels

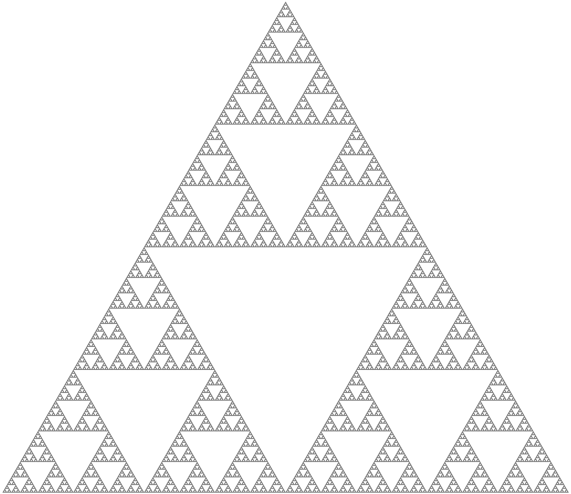


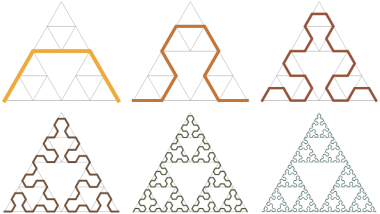
de buren bepalen de kleur











wikipedia

							1							
						1		1						
					1		2		1					
				1		3		3		1				
			1		4		6		4		1			
		1		5		10		10		5		1		
	1		6		15		20		15		6		1	
1		7		21		35		35		21		7		1

$$(x + 1)^4 = 1{\cdot}x^4 + 4{\cdot}x^3 + 6{\cdot}x^2 + 4{\cdot}x + 1$$

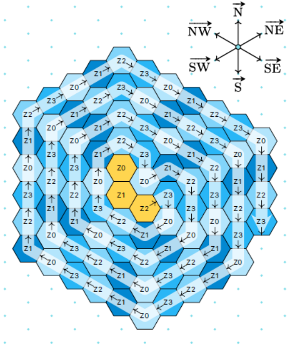
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1

$$(x+1)^4 = 1{\cdot}x^4 + 4{\cdot}x^3 + 6{\cdot}x^2 + 4{\cdot}x + 1$$

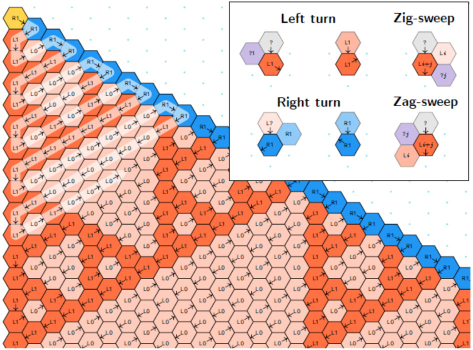
1
1 1
1 0 1
1 1 1 1
1 0 0 0 1
1 1 0 0 1 1
1 0 1 0 1 0 1
1 1 1 1 1 1 1 1

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Oritatami Systems Assemble Shapes No Less Complex Than Tile Assembly Model



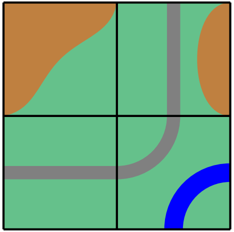
(a) A 4-cyclic clockwise walker turedo.

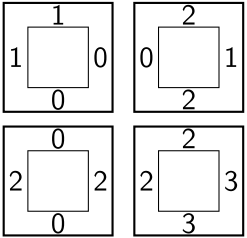


(b) A 3-states Sierpinski triangle. turedo

Oritatami Systems Assemble Shapes No Less Complex Than Tile Assembly Model (ATAM) – STACS (2022)

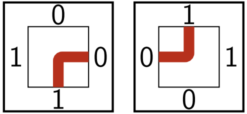
Spelregels



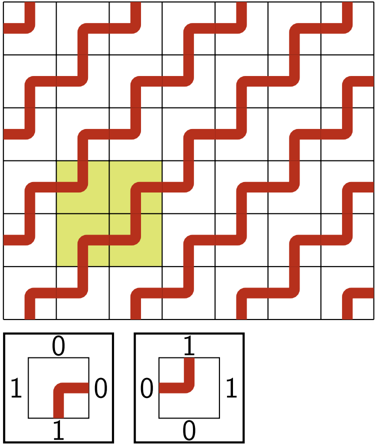


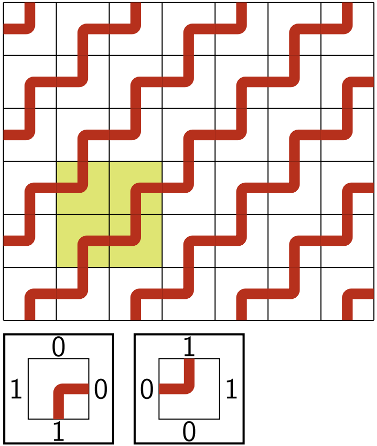
spelregels:

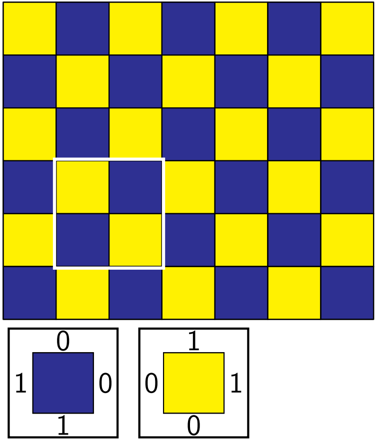
– overeenkomende zijkanten
– eindig veel types
– onbeperkt veel
– niet draaien
– eventueel randen

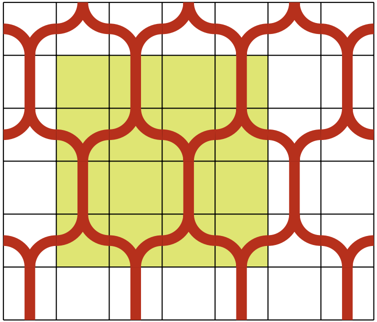


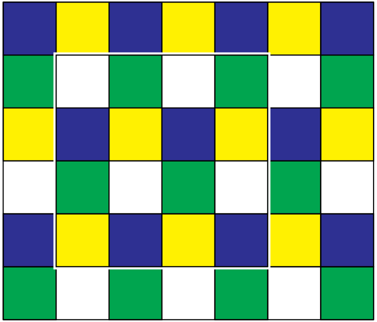
twee tegels

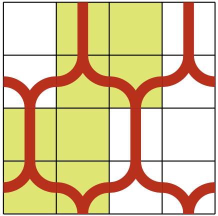

0
1
0
1
1
0
1
0

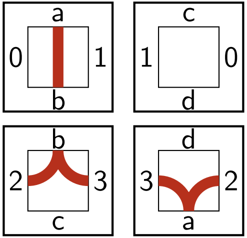

0
1
0
1
1
0
1
0


0
1
0
1
1
0
1
0

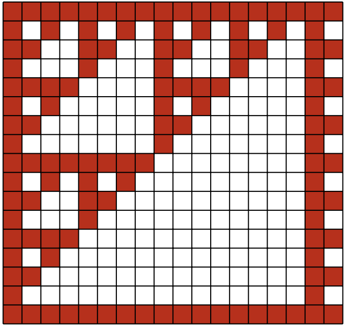


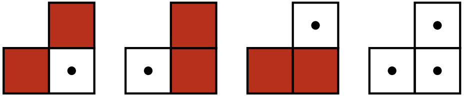


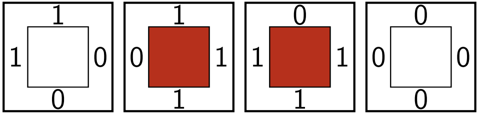


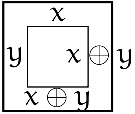

a
0
1
b
c
1
0
d
b
2
3
c
d
3
2
a

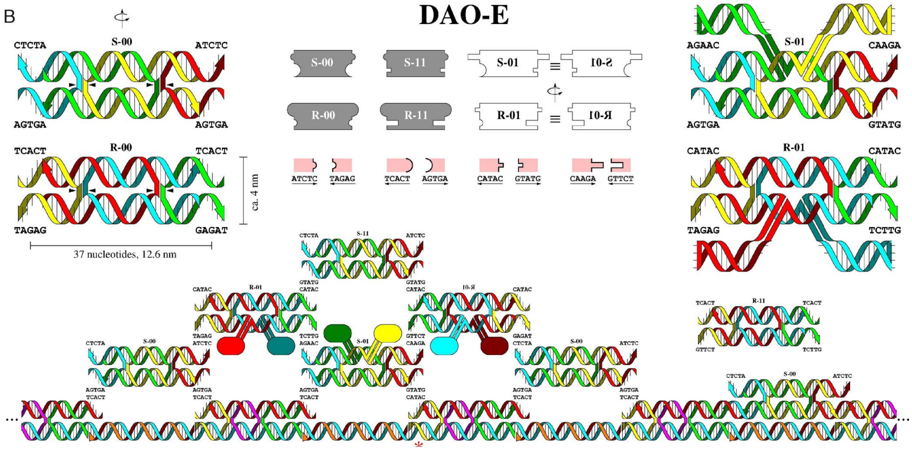
(0) Sierpiński (tegels)







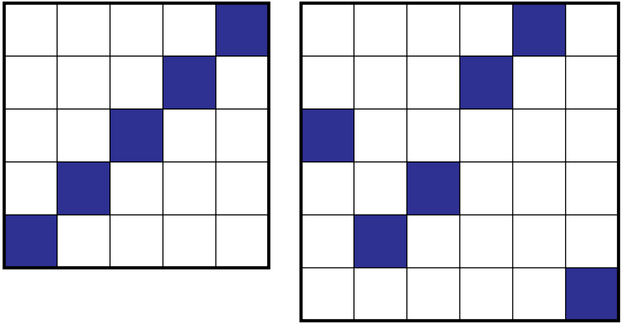


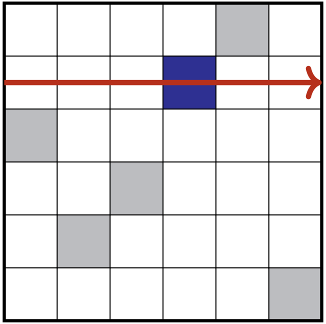


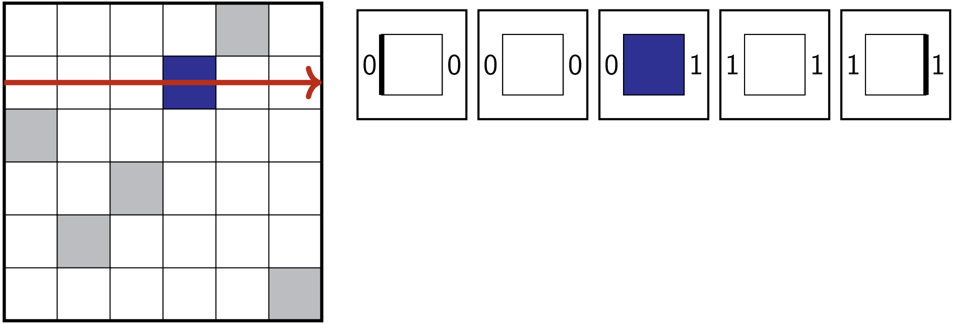
Algorithmic Self-Assembly of DNA Sierpinski Triangles (2004)
Rothemund, Papadakis, Winfree; PLoS Biology

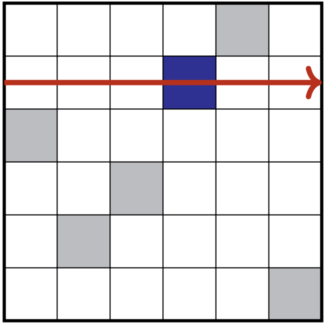
Torentegels

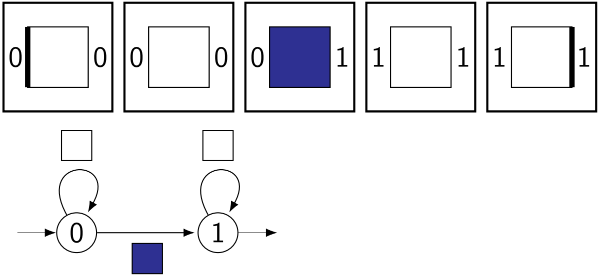
(3) torenvierkant



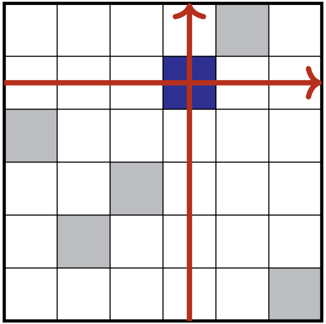


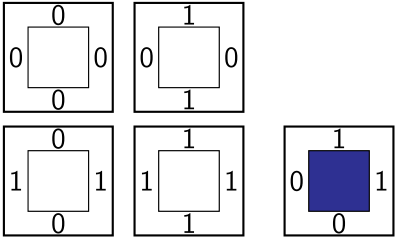

0
0
0
0
0
1
1
1
1
1





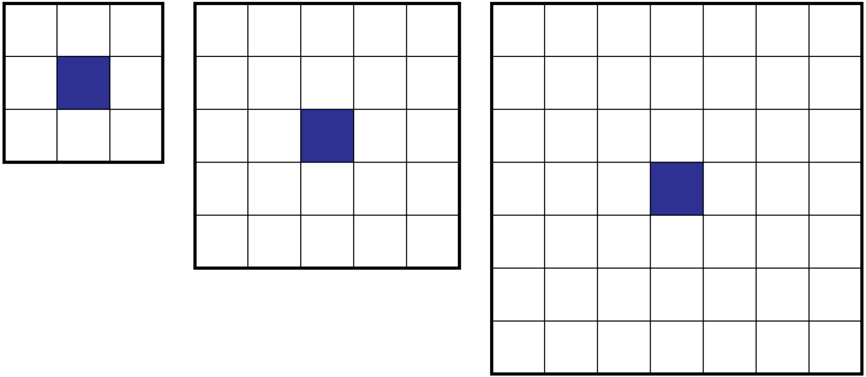
finite state automaton



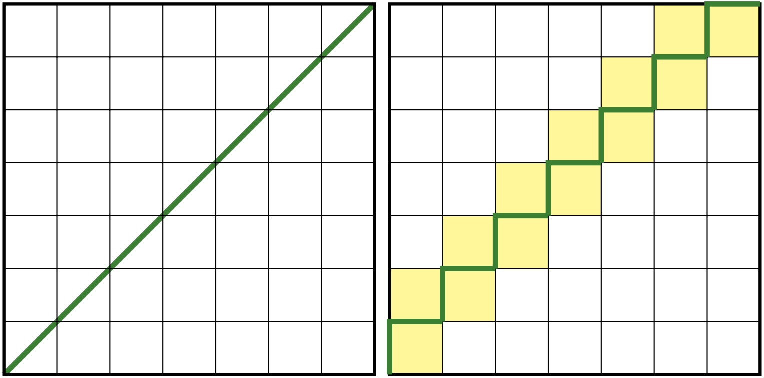


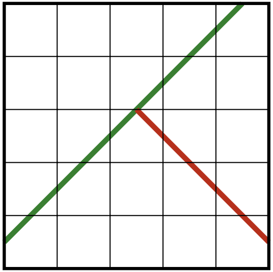
automatisch vierkant

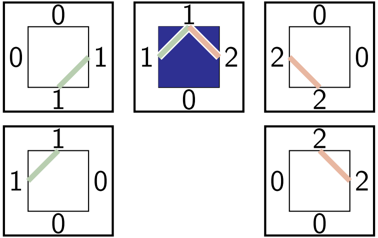
Vierkanten



(4) herken een vierkant!



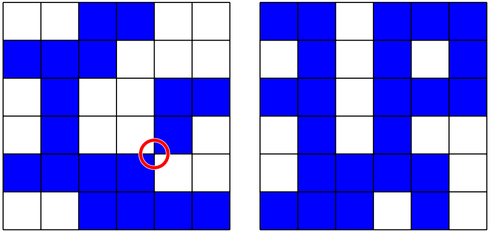




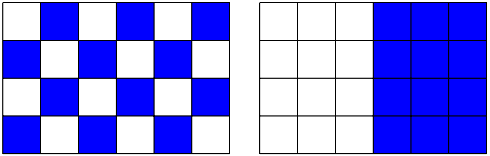
lijnen niet zichtbaar
plus rand- en hoektegels

Verbonden Tegels en Logische Puzzels

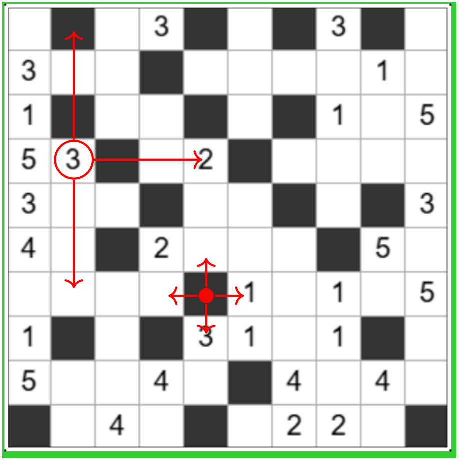
- alle blauwe tegels zijn aaneengesloten (*moeilijk*)

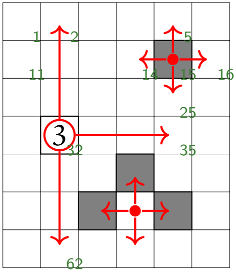


- evenveel van beide kleuren (*heel moeilijk*)



janko.at





x_j vakje j wit $\neg x_j$ zwart

overal: zwart dan alle buren wit

$\neg x_j \to (x_{j+1} \wedge x_{j-10} \wedge x_{j-1} \wedge x_{j+10})$

overal: wit dan tenminste één buur wit

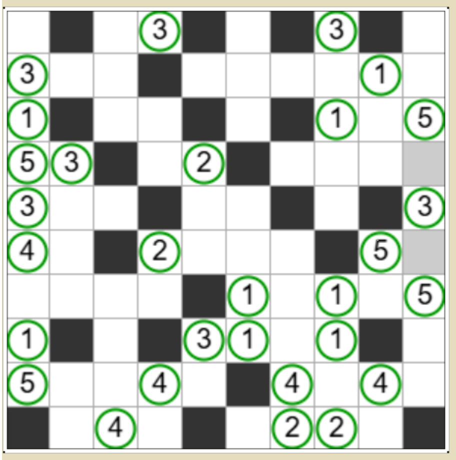
$x_j \to (x_{j+1} \vee x_{j-10} \vee x_{j-1} \vee x_{j+10})$

getal '3': wit en één op afstand zwart

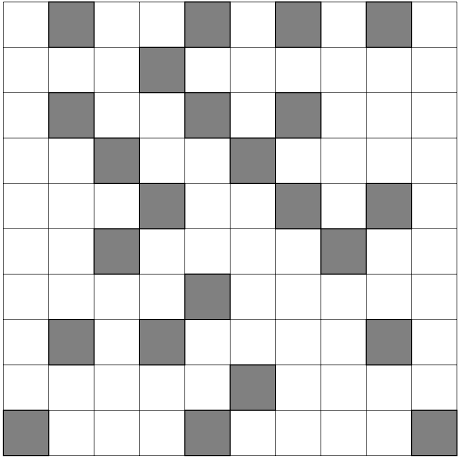
$x_j \wedge (\neg x_{j+3} \vee \neg x_{j-30} \vee \neg x_{j-3} \vee \neg x_{j+30})$

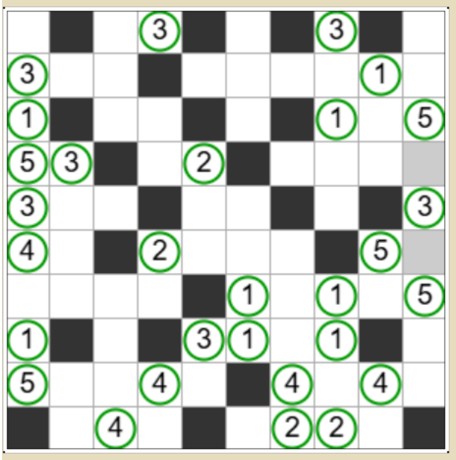
ook: geen tweetal zwart: zes formules ...

$(x_{j+3} \vee x_{j-30}) \wedge (x_{j+3} \vee x_{j-3}) \wedge \dots$



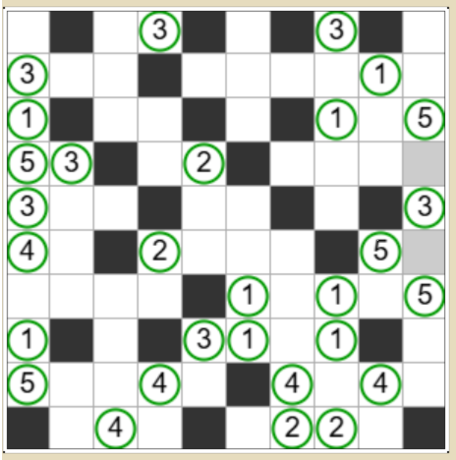
janko.at



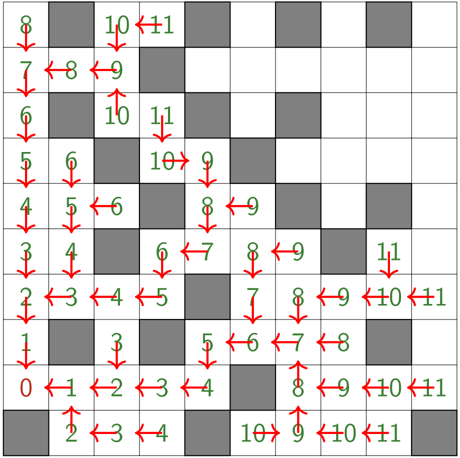


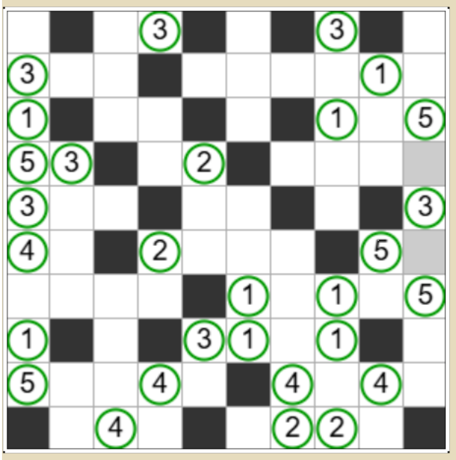
janko.at

8		10	11						
7	8	9							
6		10	11						
5	6		10	9					
4	5	6		8	9				
3	4		6	7	8	9		11	
2	3	4	5		7	8	9	10	11
1		3		5	6	7	8		
0	1	2	3	4		8	9	10	11
	2	3	4		10	9	10	11	

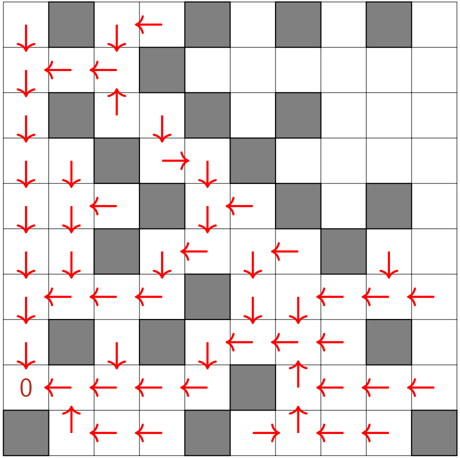


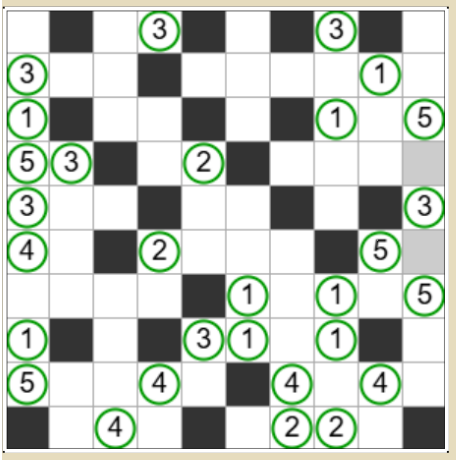
janko.at



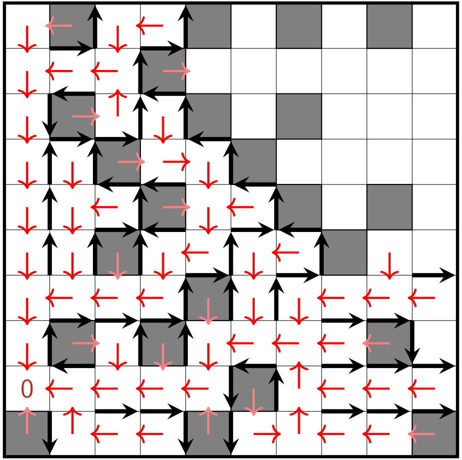


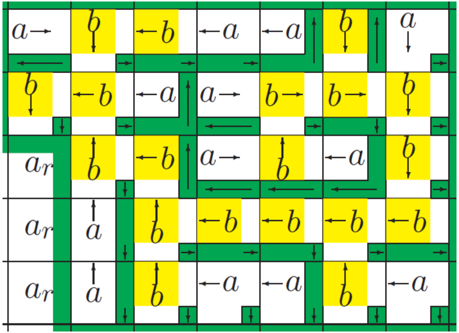
janko.at



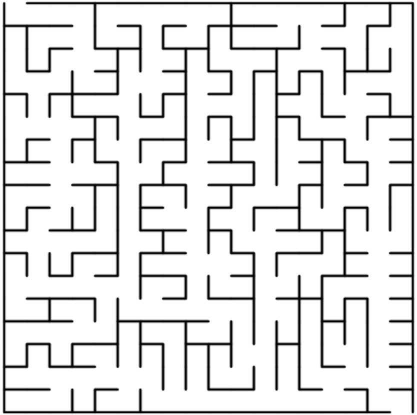


janko.at





Klaus Reinhardt. The $\#a = \#b$ pictures are recognizable. STACS 2001. LNCS 2010 (2001)



wikipedia

Binaire getallen

tientallig: machten van tien

$3408_{10} =$

$3 \cdot 10^3 + 4 \cdot 10^2 + 0 \cdot 10^1 + 8 \cdot 10^0$

binair: machten van twee

32, 16, 8, 4, 2, 1

13	
6	1
3	0
1	1
0	1

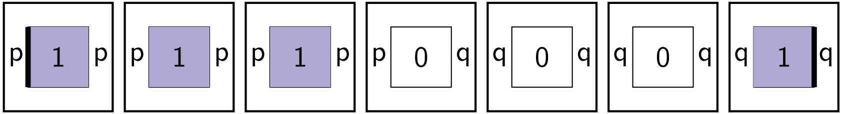
$1101_2 =$

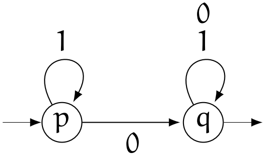
$2^3 + 2^2 + 2^0 = 8 + 4 + 1 = 13$

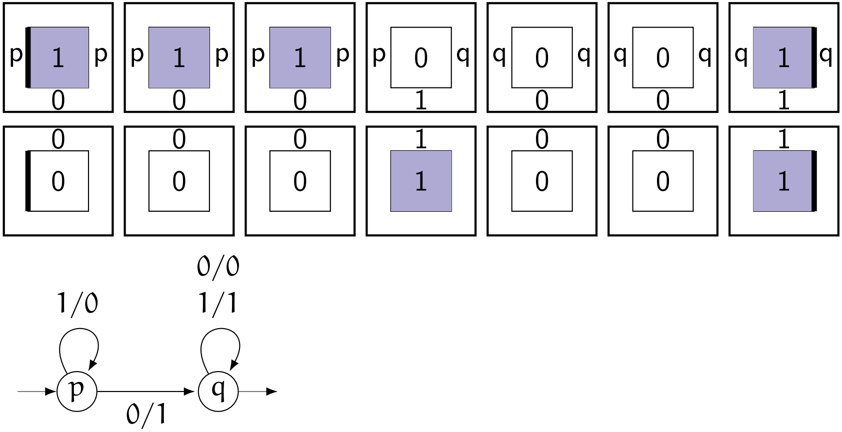
1
10
11
100
101
110
111
1000
1001
1010
1011
1100
1101
1110
1111
10000

1237
1238
1239
1240

1298
1299
1300
1301

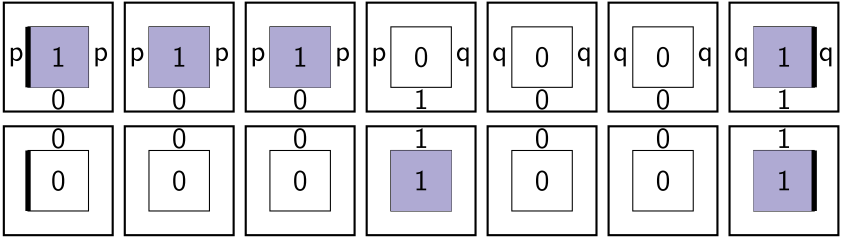

p 1 p
p 1 p
p 1 p
p 0 q
q 0 q
q 0 q
q 1 q

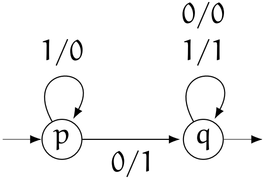

1
0
1
p
0
q

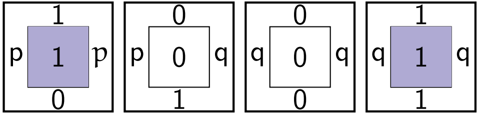


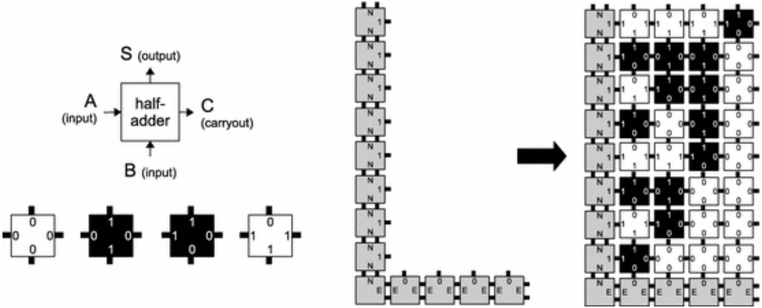
Mealy machine (Digital Systems Design)

plus één (achterstevoren)



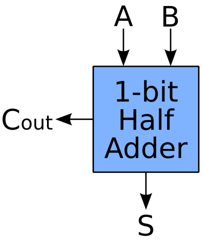


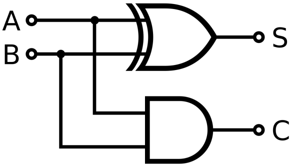




(Left) Four tile types implement the half-adder with two inputs A, B from the west and south, the output S to the north, and the carryout C to the east; (Right) copies of the half-adder tiles turn the L-shape seed into the binary counter pattern

L.Kari, S.Kopecki, P-É.Meunier, et al. Binary Pattern Tile Set Synthesis Is NP-Hard. Algorithmica (2017)

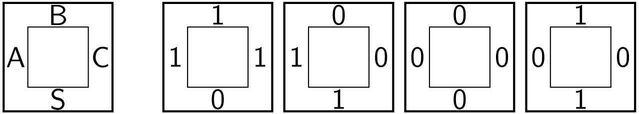


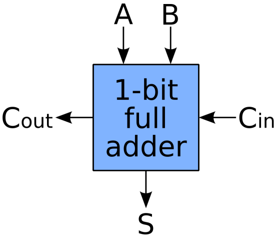


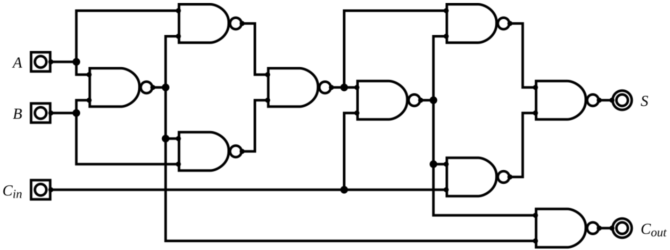
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$S = A \oplus B$$
$$C = A \cdot B$$

https://en.wikipedia.org/wiki/Adder_(electronics)







$$S = A \oplus B \oplus C_{in}$$
$$C_{out} = (A \cdot B) + C_{in} \cdot (A \oplus B)$$

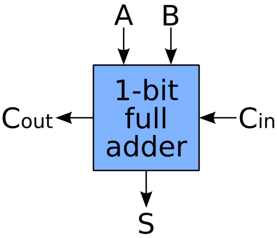
https://en.wikipedia.org/wiki/Adder_(electronics)

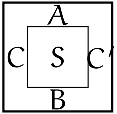
A	0	0	1	1
B	0	1	0	1
S	0	1	1	1 0

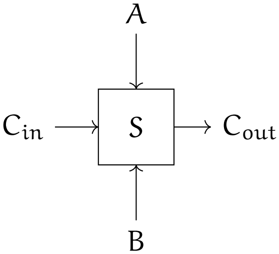
C	0	0	0	0	1	1	1	1
A	0	1	0	1	0	1	0	1
B	0	0	1	1	0	0	1	1
S	0	1	1	0	1	0	0	1
C′	0	0	0	1	0	1	1	1

A	0	0	1	1
B	0	1	0	1
S	0	1	1	1 0

C	0	0	0	0	1	1	1	1
A	0	1	0	1	0	1	0	1
B	0	0	1	1	0	0	1	1
S	0	1	1	0	1	0	0	1
C′	0	0	0	1	0	1	1	1

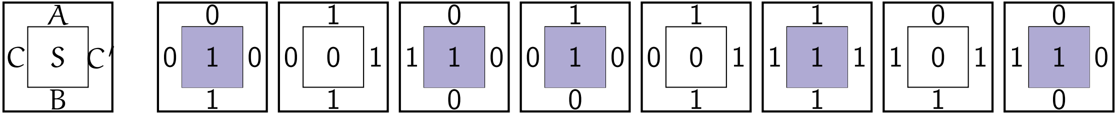






A	0	0	1	1
B	0	1	0	1
S	0	1	1	1 0

C	0	0	0	0	1	1	1	1
A	0	1	0	1	0	1	0	1
B	0	0	1	1	0	0	1	1
S	0	1	1	0	1	0	0	1
C′	0	0	0	1	0	1	1	1



$$0101110 \quad + \quad 1100111 \quad = \quad 10110101$$ (achterstevoren)

$$58 \quad + \quad 115 \quad = \quad 173$$

Ter afsluiting

Alan Turing (1912–1954)

science museum

Turing machine (1936)

computability: wát we kunnen berekenen

complexity: hoe efficient we dat kunnen

A.M. Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. Proc LMS (1936).

Wang tiles (Hao Wang, 1961)

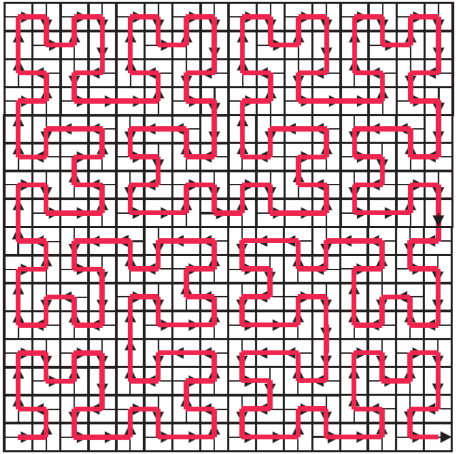


Fig. 1. Plane filling snake forced by D_0.

Jarkko Kari. Infinite Snake Tiling Problems.
DLT 2002. LNCS 2450 (2003)

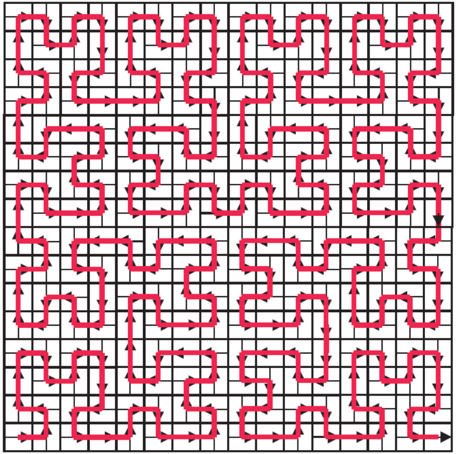
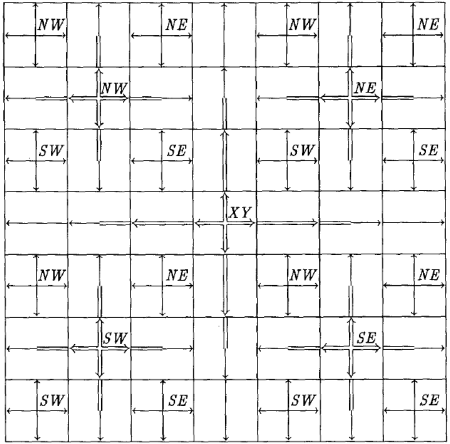


Fig. 1. Plane filling snake forced by D_0.

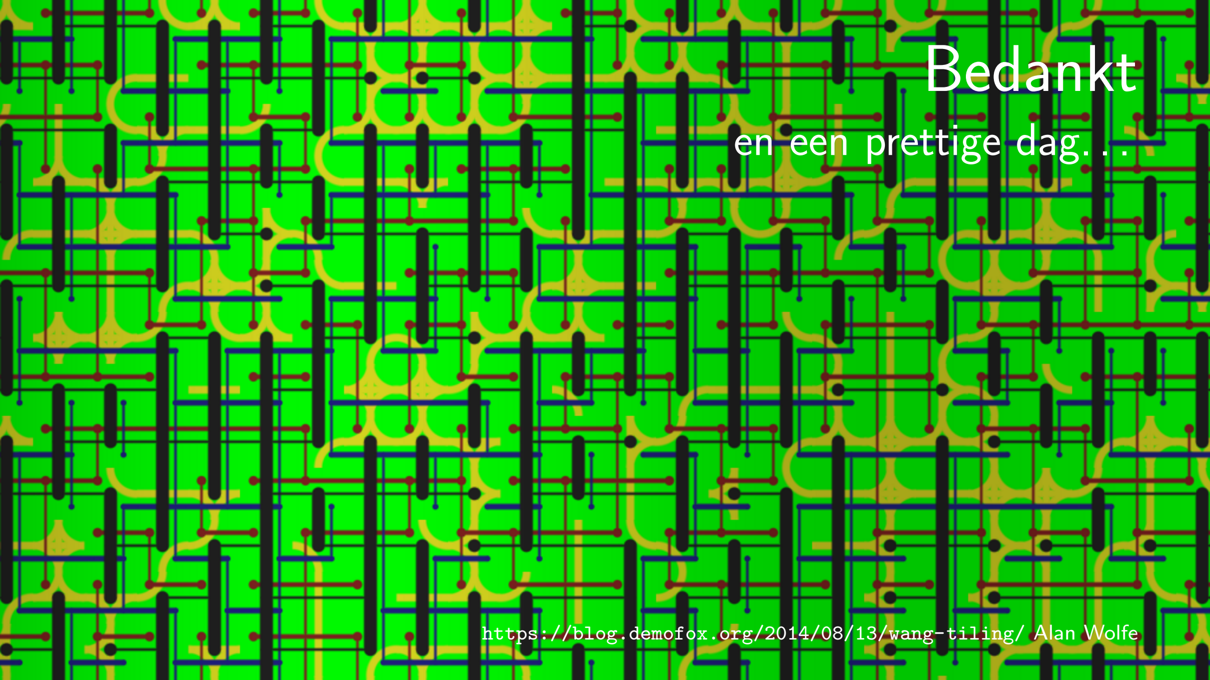
Jarkko Kari. Infinite Snake Tiling Problems. DLT 2002. LNCS 2450 (2003)



Jarkko Kari. Reversibility and surjectivity problems of cellular automata, JCSS 48 (1994)

Bedankt

en een prettige dag. . .



https://blog.demofox.org/2014/08/13/wang-tiling/ Alan Wolfe

There are several experimental assemblies of DNA-based tile arrays that show that aTAM can carry out computation. These include a binary counter using four DX-based tiles [11], binary addition by TX molecules [15], Sierpinski triangle as a pattern on a substrate [25, 12], transducer simulations by TX molecules [6], and the most recent one where different combinations of input tiles achieve a variety of computations [30].