Sage Days, Oxford, September 2013

Computational Homology via Discrete Morse Theory

Shaun Harker
Rutgers University

Sage Days, Oxford, September 2013

Introduction

- We can use Discrete Morse theory, originally developed by Forman, to compute homology.
- We generate discrete Morse functions by a generalization of the coreductions approach of Mrozek and Batko.
- We give an efficient way to compute the boundary in the reduced complex.

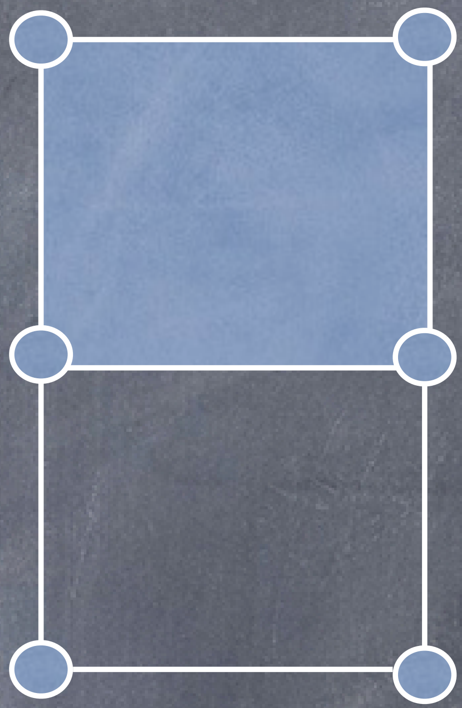
Chain Complex Definition

A principal ideal domain $\mathcal{R}$

Finite sets of *cells* $\mathcal{S}_k, \ k = 0, 1, \cdots, d$

Chain modules $\mathcal{C}_k, \ k = 0, 1, \cdots, d$

Boundary maps $\partial_k, k = 0, 1, \cdots, d$



$$\mathcal{C}_k = \mathcal{R}(\mathcal{S}_k)$$

satisfying: $\partial_{k+1} : \mathcal{C}_{k+1} \to \mathcal{C}_k$

$$\partial^2 = 0$$

Purely algebraic Homology Algorithm

1) Apply Smith Normal Form to each boundary matrix.

2) Read off the invariant factors.

3) Count up # of 0's, 1's, and higher divisors, and number of cells, and do some arithmetic

4) This yields the homology groups in terms of their invariant factor decompositions

Homology Generators Algorithm

And if I want the generators?

We'll need the multipliers from the SNF:

$$A = UDV$$

Essential idea: Compute the invariant factors AND the multipliers for each boundary matrix and then fiddle around with them.

Problem.

It seems that fast SNF algorithms will not give multipliers.

Moreover, we experience cubic run times.

This leaves us looking for ways to somehow reduce the problem before having to do the matrix algebra.

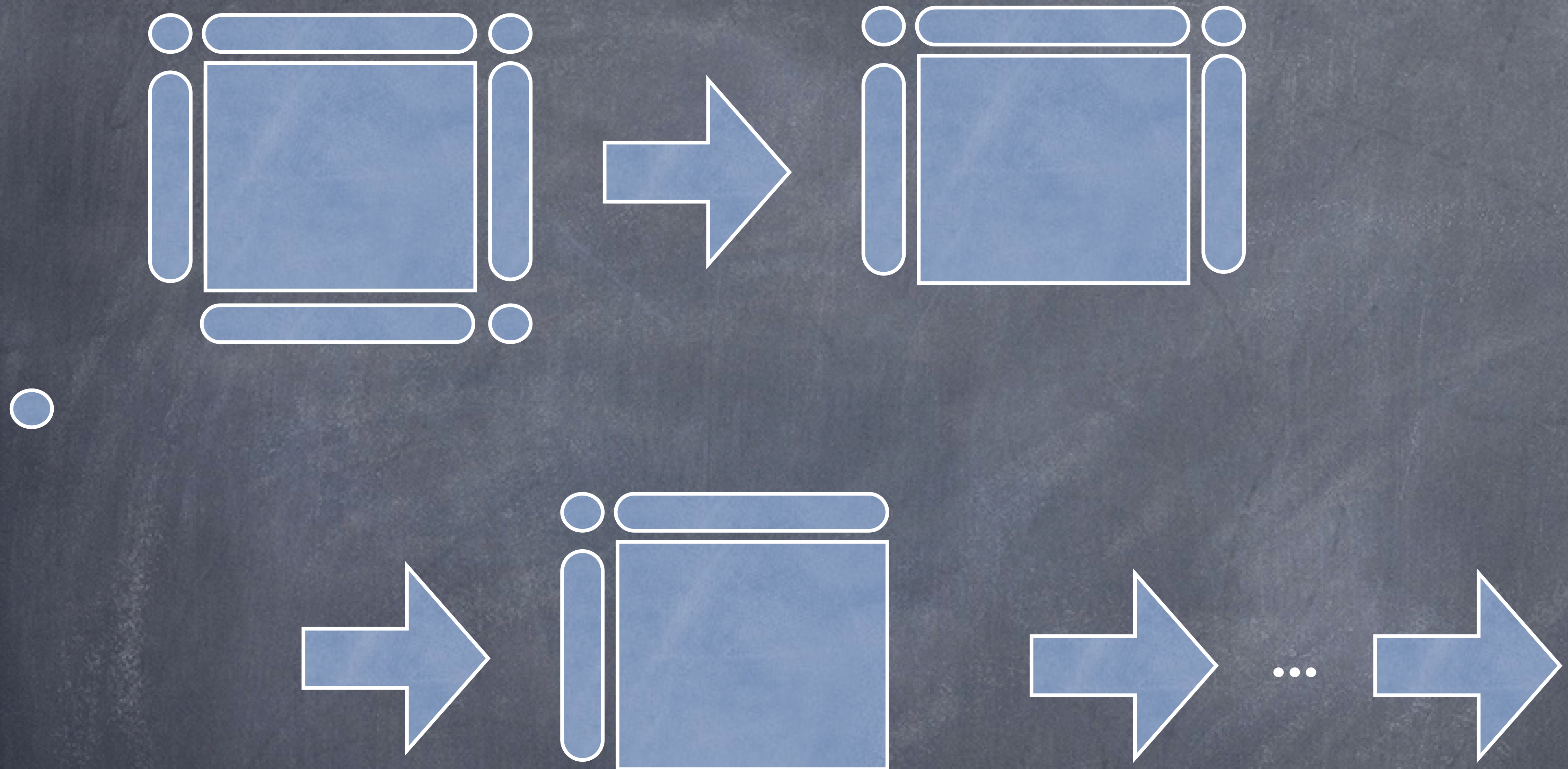
Faster methods

Marian Mrozek and Bogdan Batko. *Coreduction Homology Algorithm.* Discrete and Computational Geometry. **41**(1) (2009) 96-118

Marian Mrozek realized that if you excise a vertex from each connected component, the homology only changes on the 0th level (i.e. the Betti # decreases by the number of connected components)

Additionally, there are simple reductions this enables us to do afterwards.

Free coface collapse:

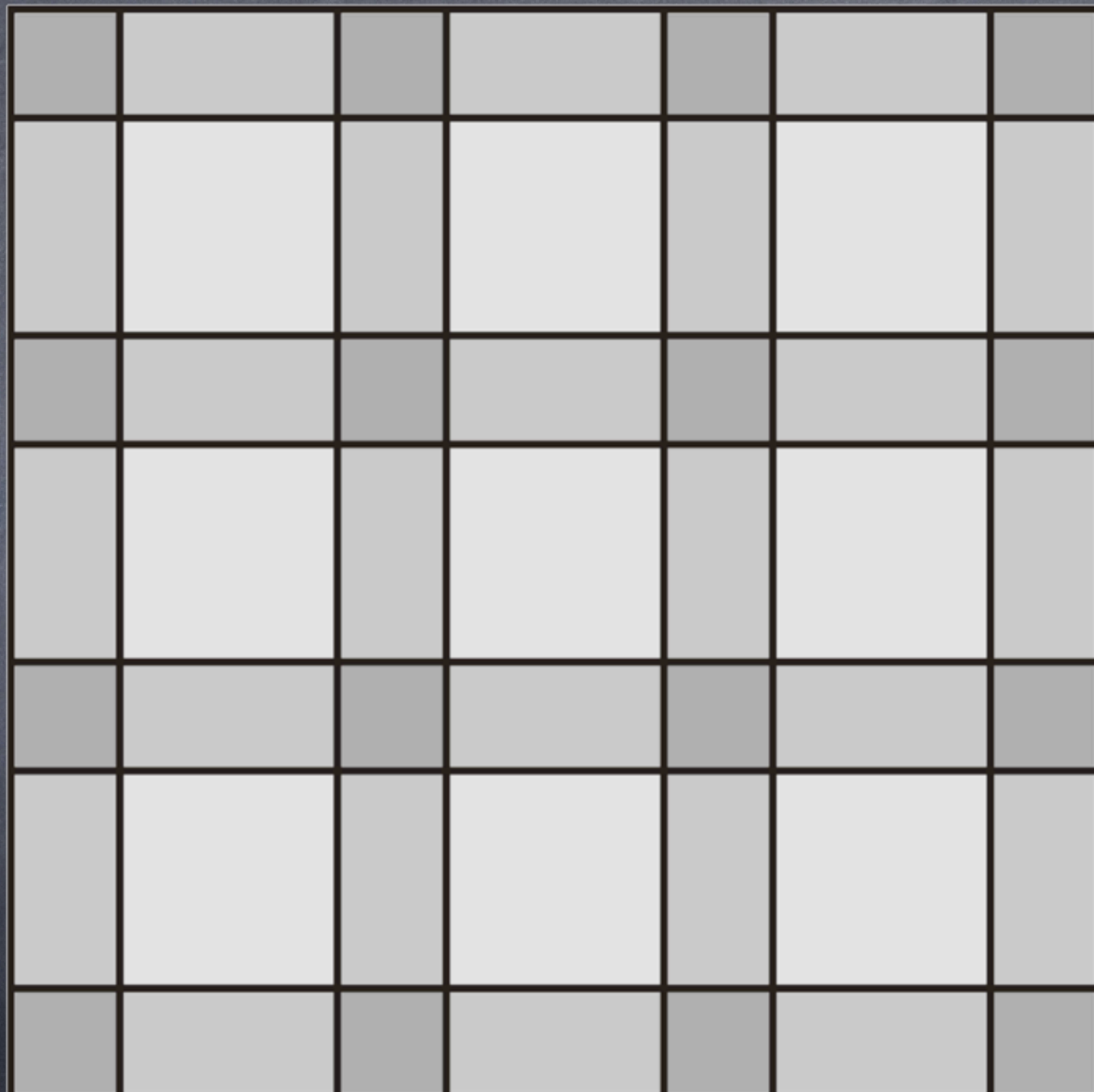


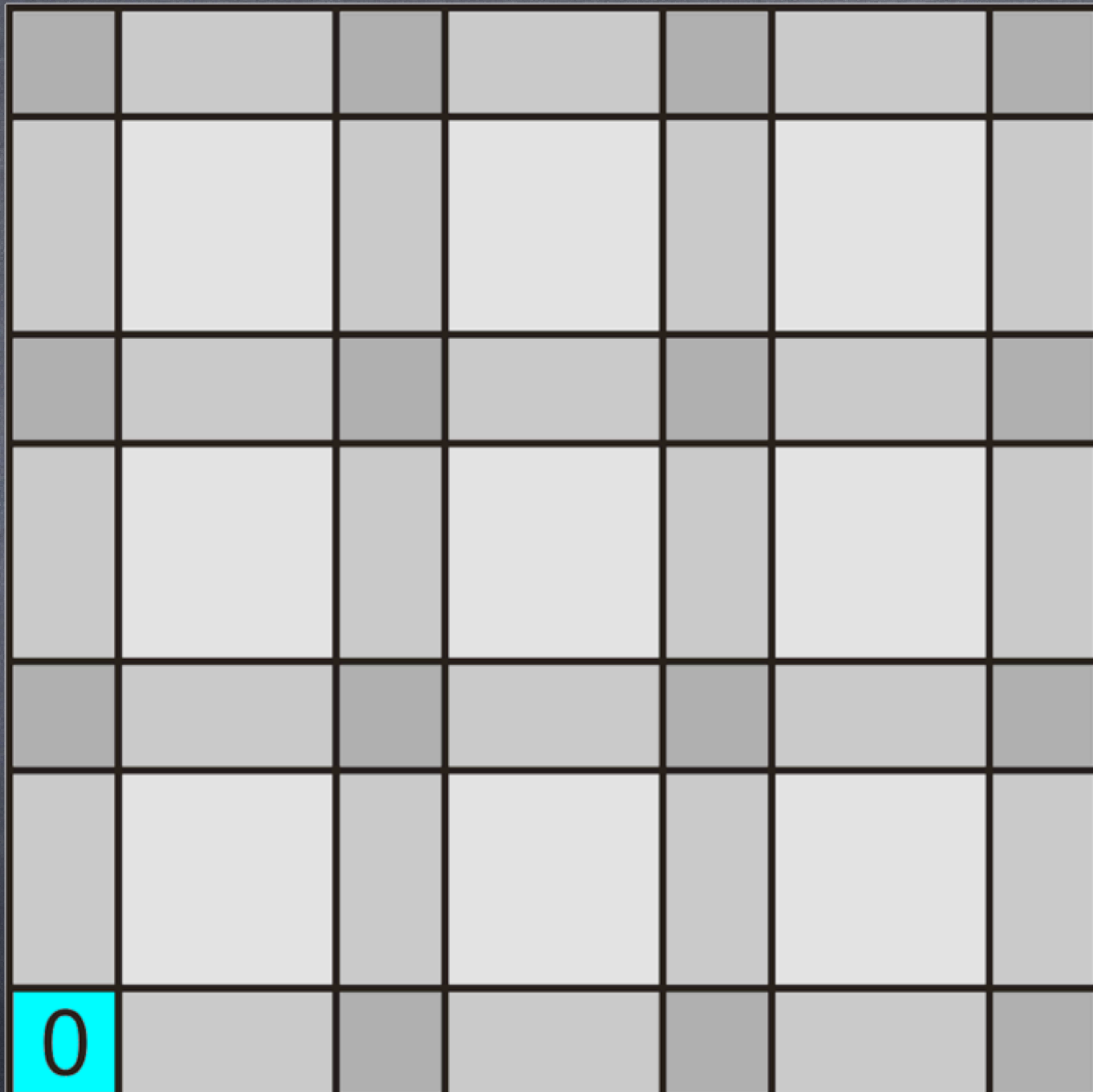
Whenever a cell has precisely one other cell in its boundary, we may excise the pair without altering the homology.

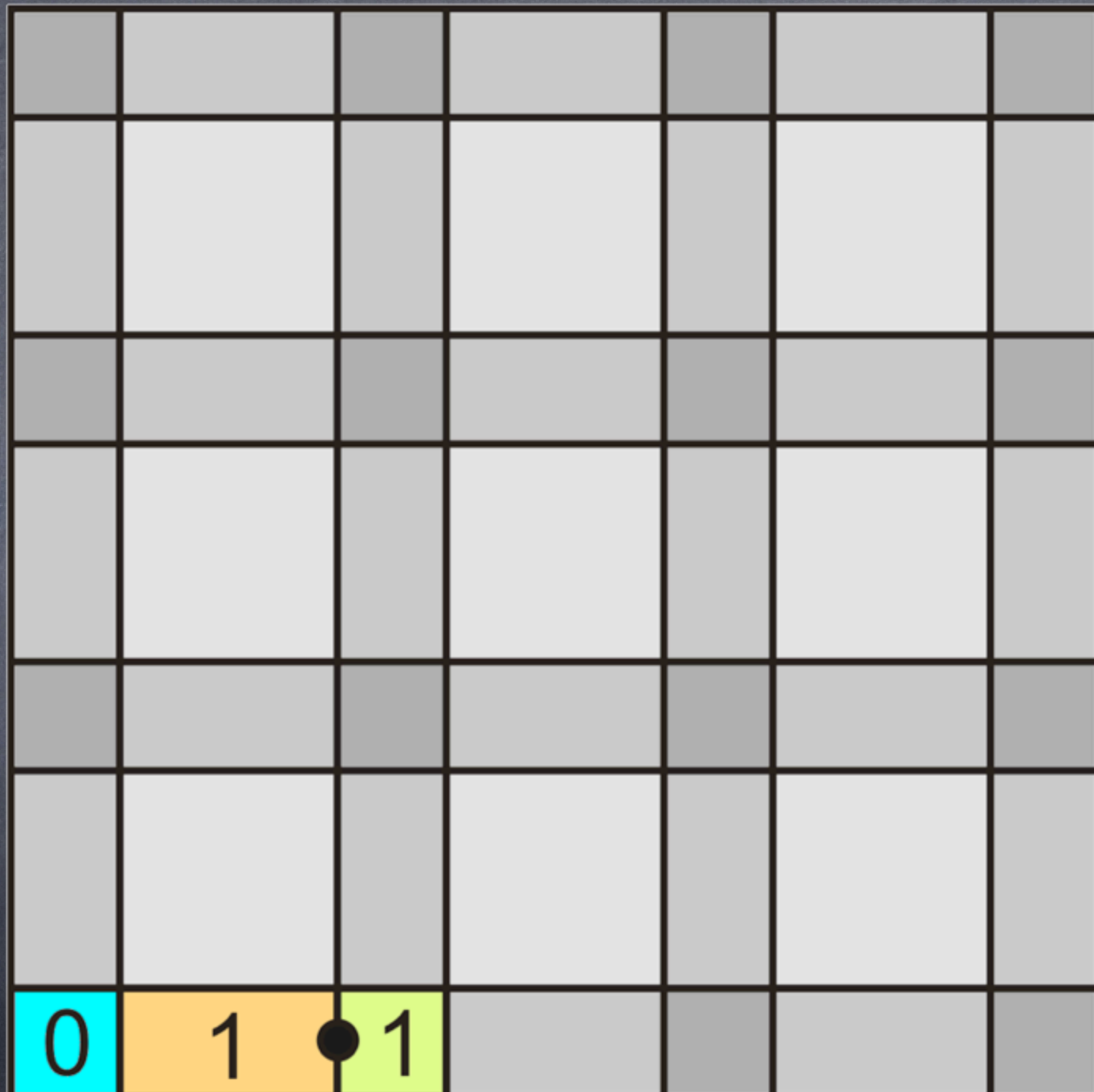
This was very effective, but not perfect. One can make obstructions to this approach.

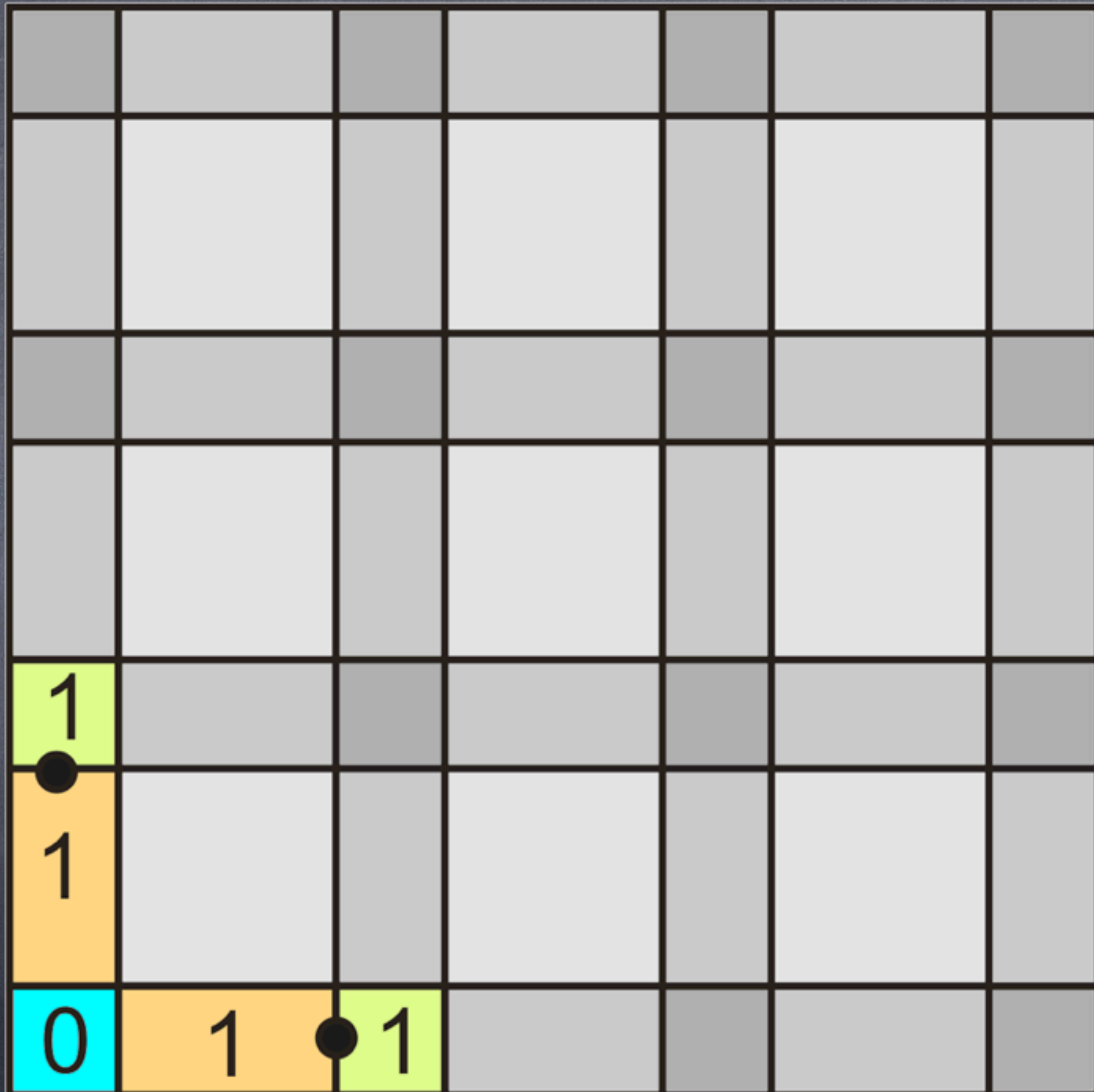
In particular, you are gobbling up an acyclic subcomplex, but if there is deeper homology you must leave things behind.

A simple example is the torus:




0


0
1
1


1
1
0
1
1


1
2
1
2
0
1
1


1 3 2
1 3 2
0 1 1

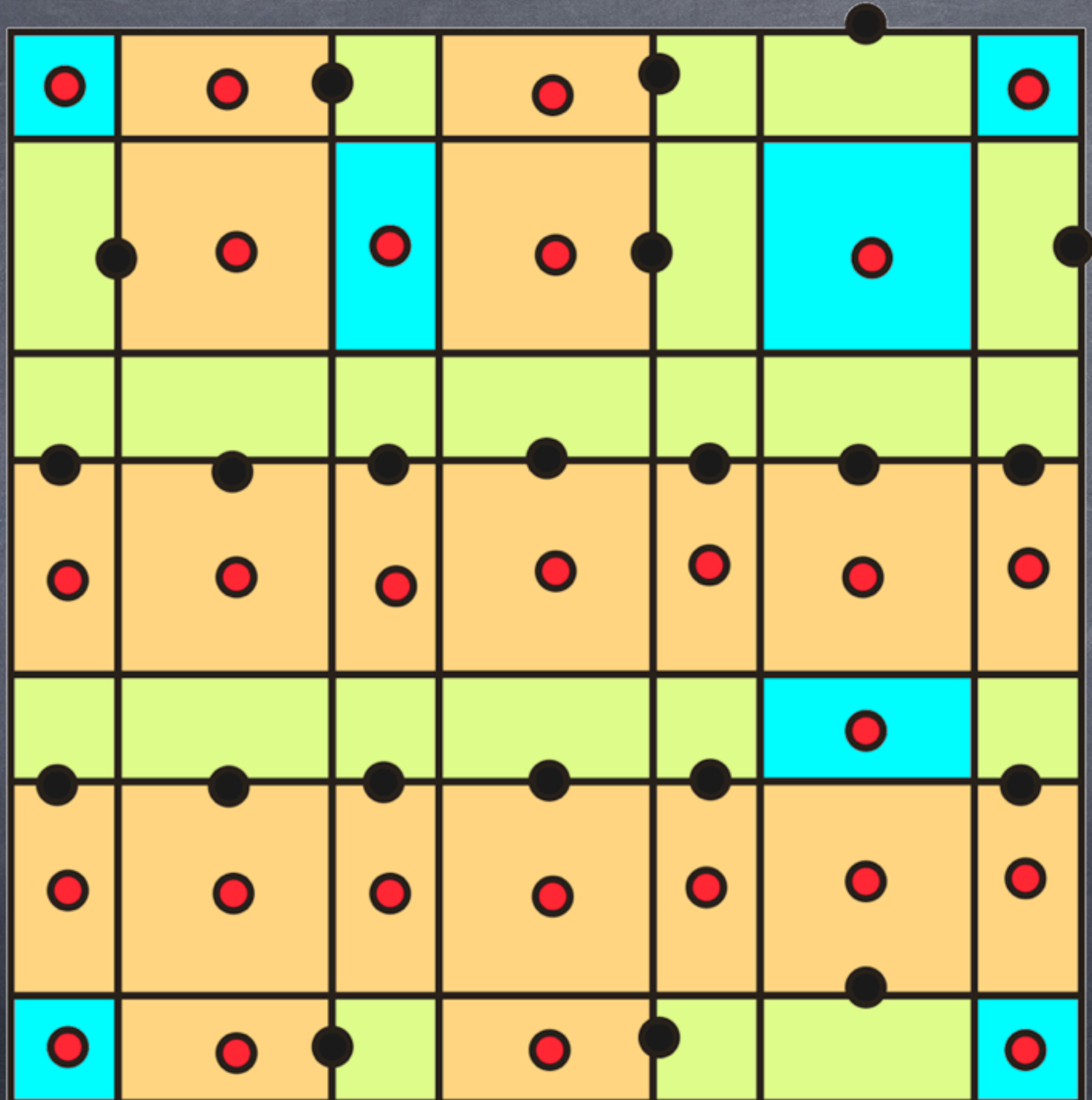

0 1 1 2 2 0
2 4 3 5 4 2
2 4 3 5 4 2
1 3 2 4 3 1
1 3 2 4 3 1
0 1 1 2 2 0

The idea now is: wouldn't it be nice if we didn't have to stop?


0 1 1 2 2 0
2 4 3 5 4 2
2 4 3 5 4 2
1 3 2 4 3 1
1 3 2 4 3 1
0 1 1 2 2 0


0 1 1 2 2 0
2 4 3 5 4 2
2 4 3 5 4 2
1 3 2 4 3 4 1
1 3 2 4 3 1
0 1 1 2 2 0


0 1 1 2 2 5 0
4
2 4 3 5 4 5 2
2 4 3 5 4 5 2
1 3 2 4 3 4 1
1 3 2 4 3 5 1
0 1 1 2 2 5 0



Homology of the Torus

And we have excised 1 0-cell, 2 1-cells, and 1 2-cell cell.

This is promising, since:

$$H_*(\mathbb{T}^2) \approx (\mathbb{Z}, \mathbb{Z}^2, \mathbb{Z})$$

The Correct Generalization

This doesn't always work, however.

One possible solution seemed to be to put a new boundary operator on the "special" cells we removed.

It turns out the correct generalization is Robin Forman's "discrete Morse theory".

Discrete Morse Function

Forman expressed his theory in terms of a "discrete Morse function" in direct analogy to classical Morse theory.

The discrete Morse function gives rise to a "discrete gradient vector field". It is easier to ignore the discrete Morse function at first pass and axiomatize the discrete gradient vector field directly.

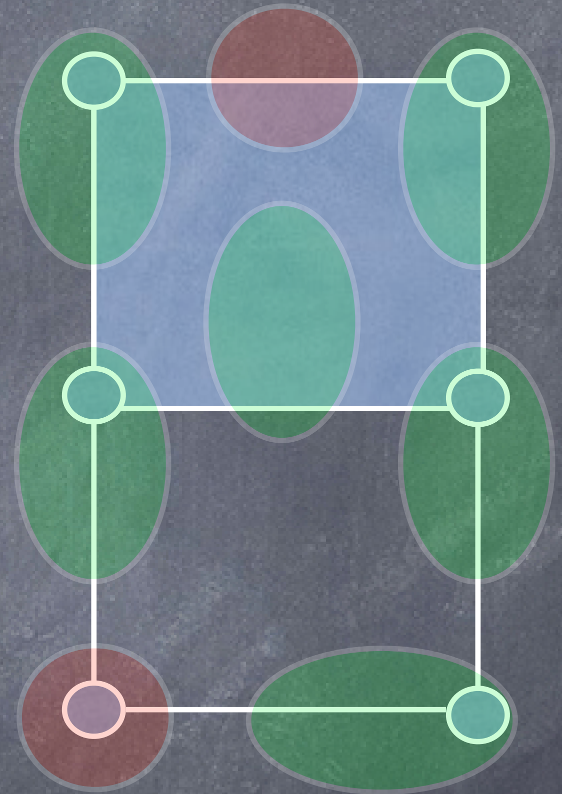
Discrete Gradient Vector Field (AKQ Decomposition)

- Captures information in a discrete Morse function up to equivalence
- Essentially, a partitioning into three classes called Aces, Kings, and Queens.
- Kings and Queens are paired up 1-1

Discrete Gradient Vector Field (AKQ Decomposition)

- Captures information in a discrete Morse function up to equivalence
- Essentially, a partitioning into three classes called Aces, Kings, and Queens.
- Kings and Queens are paired up 1-1

example:



legend:

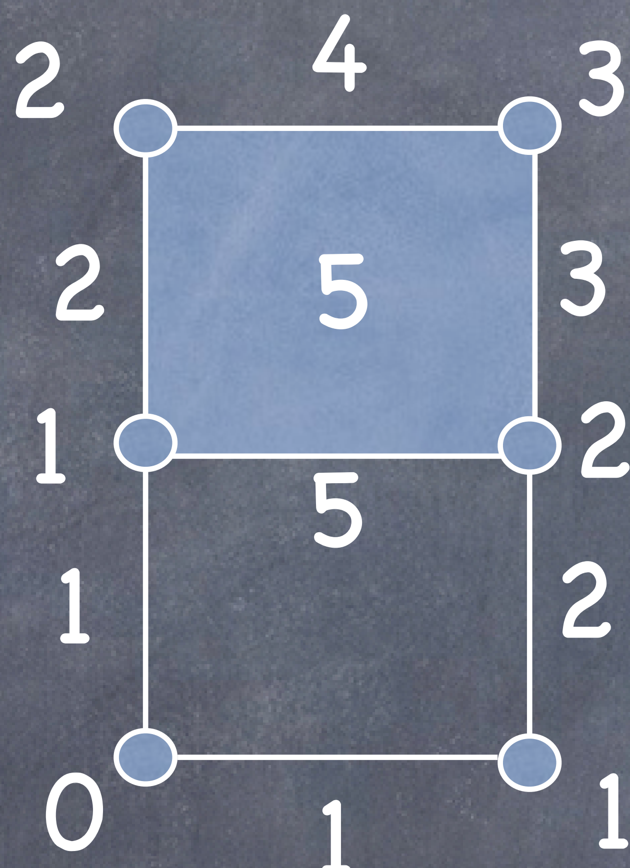
Rules on AKQ Decomposition

- A queen cell must be in the boundary of its associated king cell, so that dim K = dim Q + 1
- The incidence number between King and Queen pairs must be a unit -- that is, multiplicatively invertible in the ring
- Also, there is an acyclicity condition we will discuss.

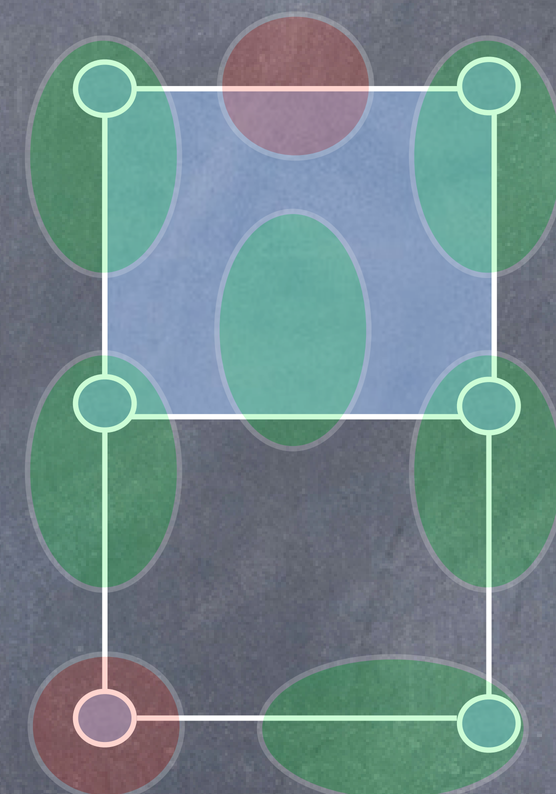
Comments on Notation

- An "Ace-King-Queen Decomposition" is just another way of expressing the discrete gradient vector field
- ``Aces" are Critical Cells
- ``King-Queen" pairs give the ``arrows" of the discrete gradient vector field
- Why make these "Ace/King/Queen" names at all? So we have some short nouns to indicate what role a given cell has with respect to a Discrete Morse Function.

Discrete Morse functions compared to AKQ decomposition



with discrete Morse function, acyclicity is guaranteed by assumptions about how the numbers fit

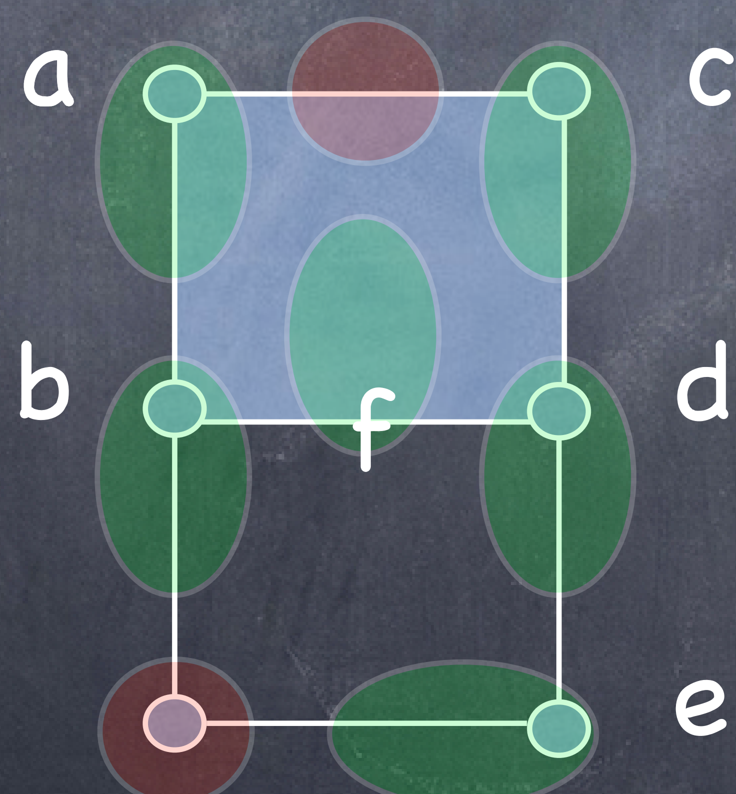


in an AKQ decomposition, we just assume acyclicity directly

Acyclicity Assumption

- There is a partial order induced on the Queens by the generating relationship of

$$\langle Q', \partial K \rangle \neq 0 \text{ and } Q = K^* \text{ implies } Q' \leq Q$$

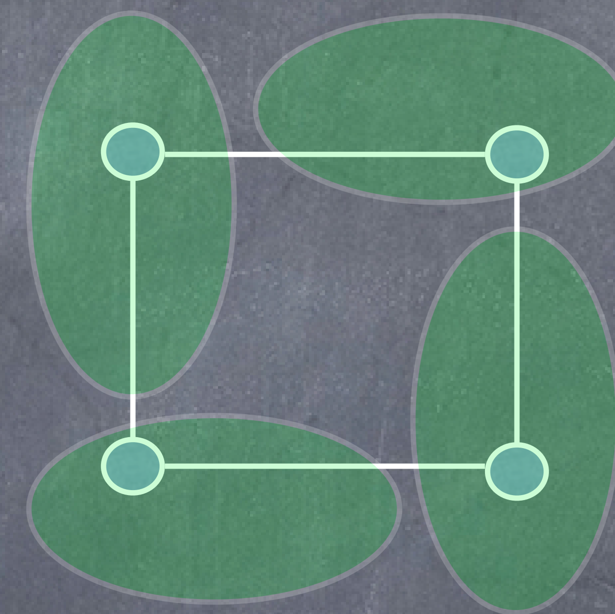


a > b

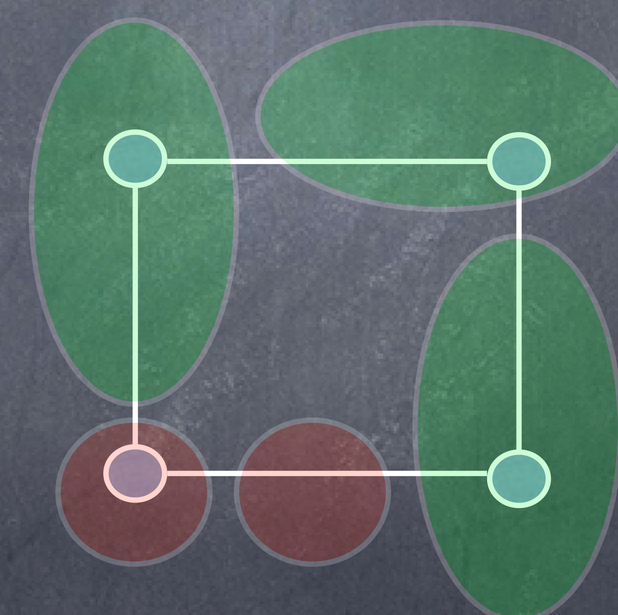
c > d > e

Acyclicity Examples:

Not an AKQ decomposition... violates acyclicity condition!



However, this is okay:



How do we generate AKQ decompositions from scratch?

One method is an extension of the coreduction technique of Mrozek and Batko.

In this method, a reduced complex was found by first excising one vertex from each connected component of the original complex, and then performing free coface collapses until they were exhausted. The vertices are then replaced.

This procedure is linear time.

Coreduction Based AKQ Decomposition Algorithm

n <- 0

Loop until complex is empty

If there is a free coface collapse pair (K, Q)

K is now a King, Q is now a Queen, K* := Q.

v(K) := v(Q) := n, and n <- n+1.

Excise K and Q from the complex.

If there is no free coface collapse,

Choose a cell A such that dA = 0.

A is now an Ace.

v(A) := n, and n <- n+1.

Excise A from the complex.

End Loop

Illustration of the coreduction based decomposition algorithm at work

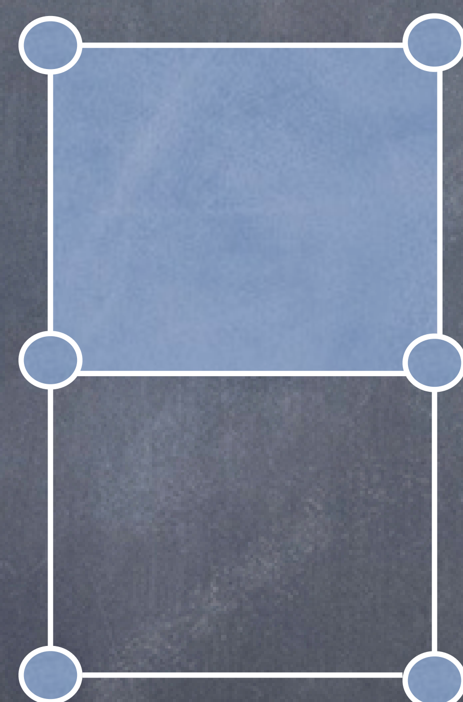


Illustration of the coreduction based decomposition algorithm at work

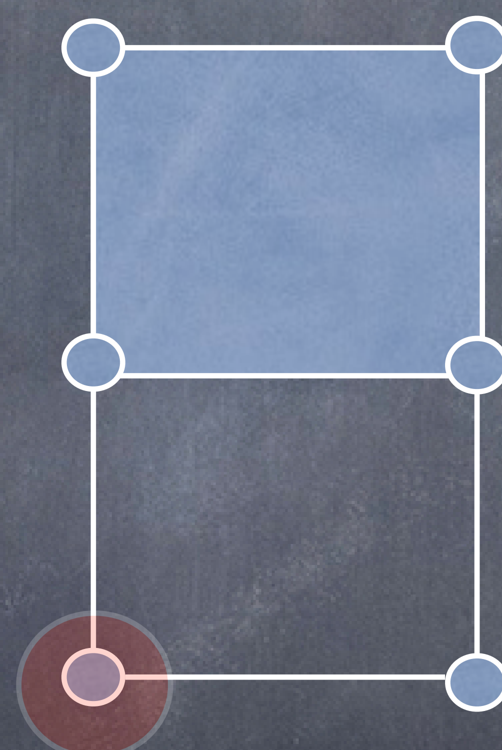


Illustration of the coreduction based decomposition algorithm at work

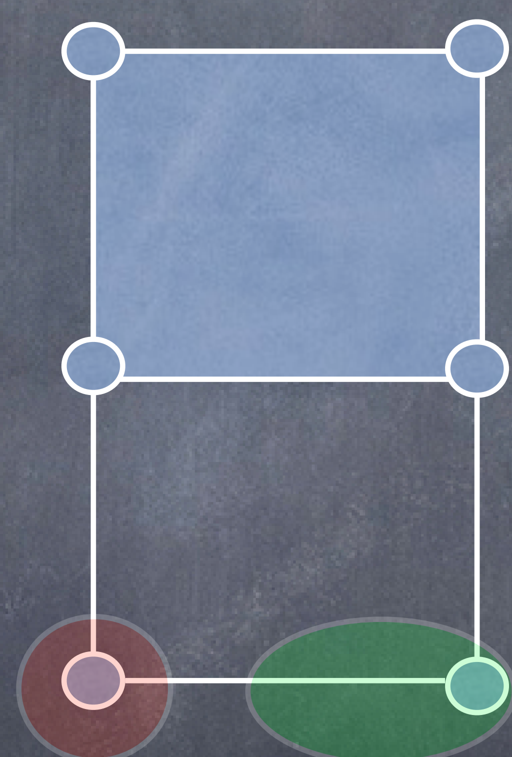


Illustration of the coreduction based decomposition algorithm at work

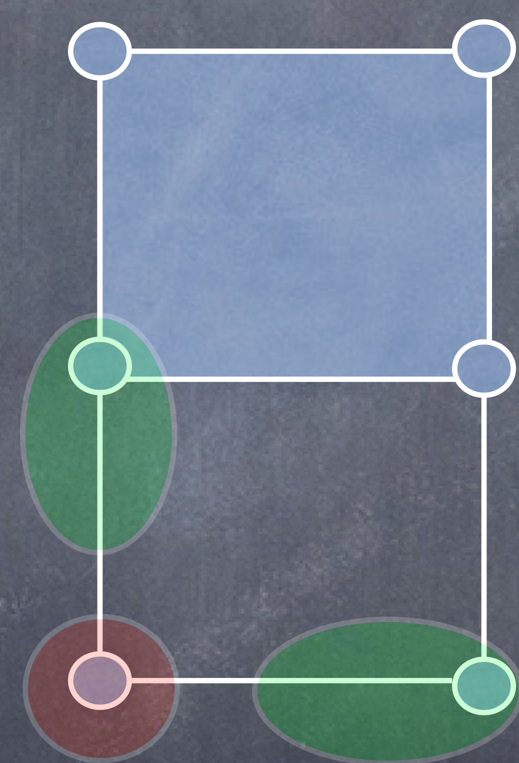


Illustration of the coreduction based decomposition algorithm at work

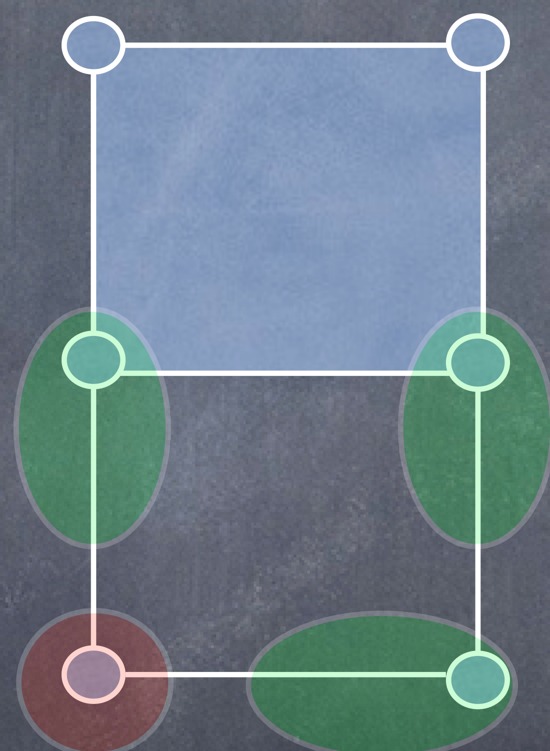


Illustration of the coreduction based decomposition algorithm at work

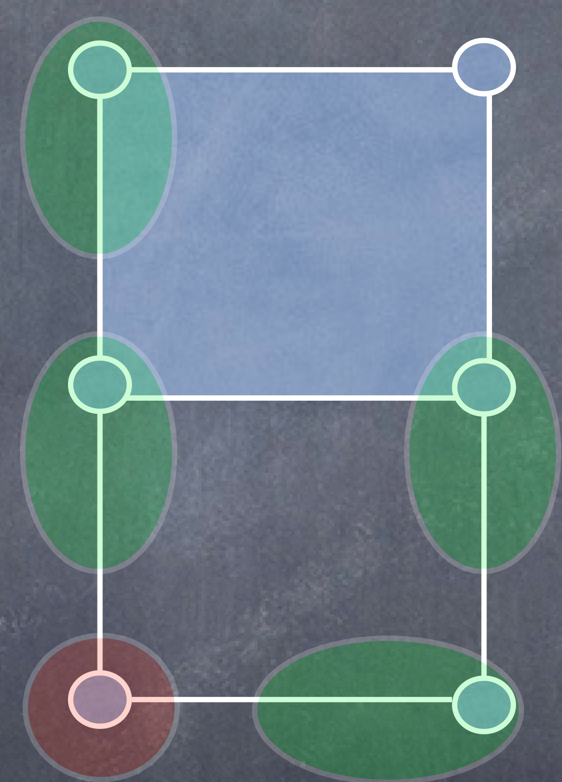


Illustration of the coreduction based decomposition algorithm at work

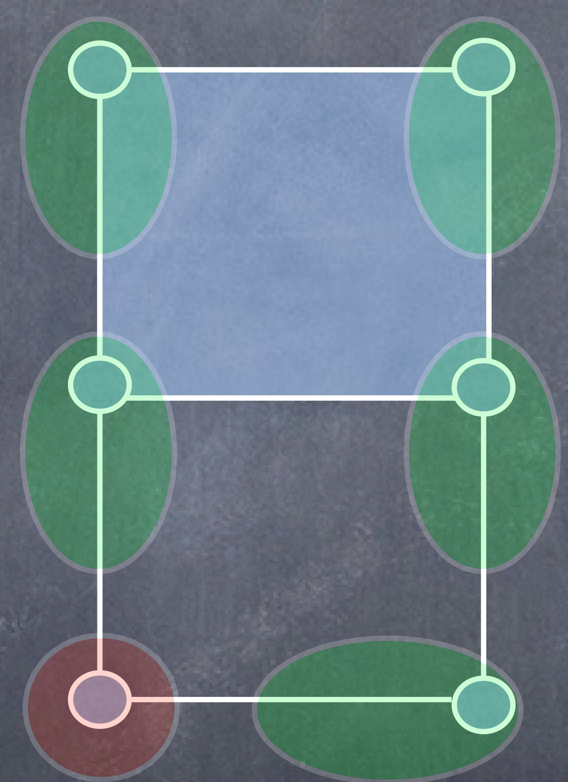


Illustration of the coreduction based decomposition algorithm at work

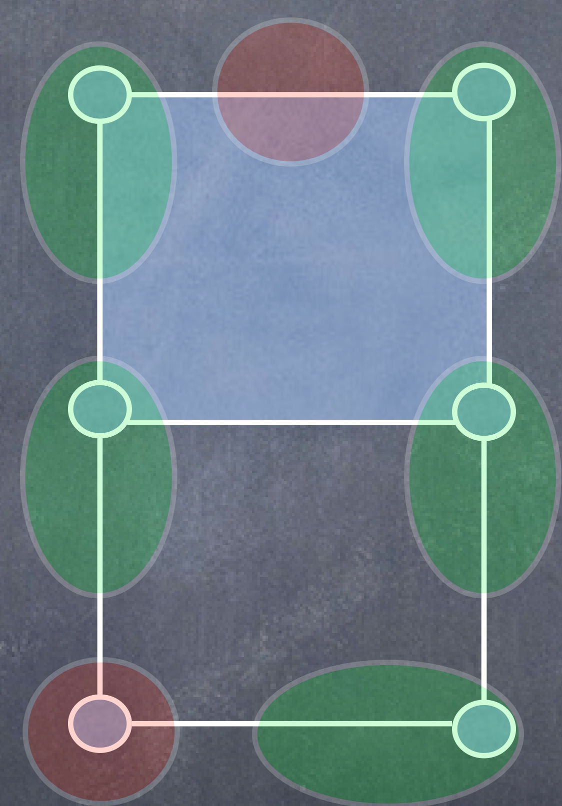
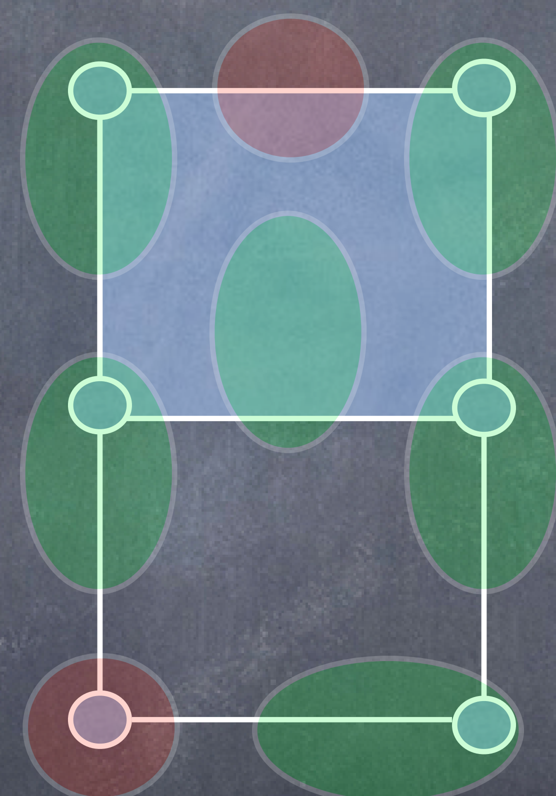
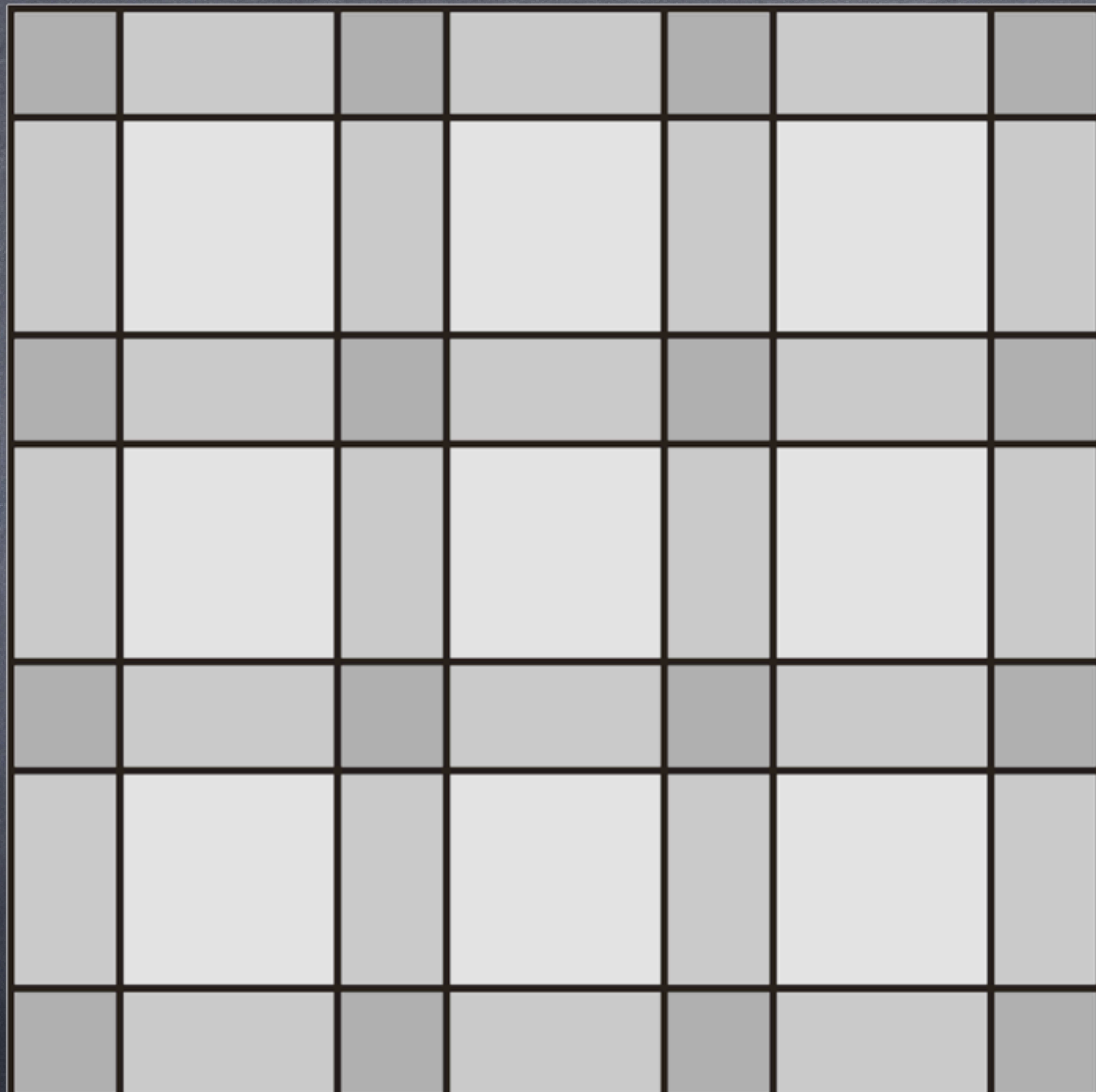
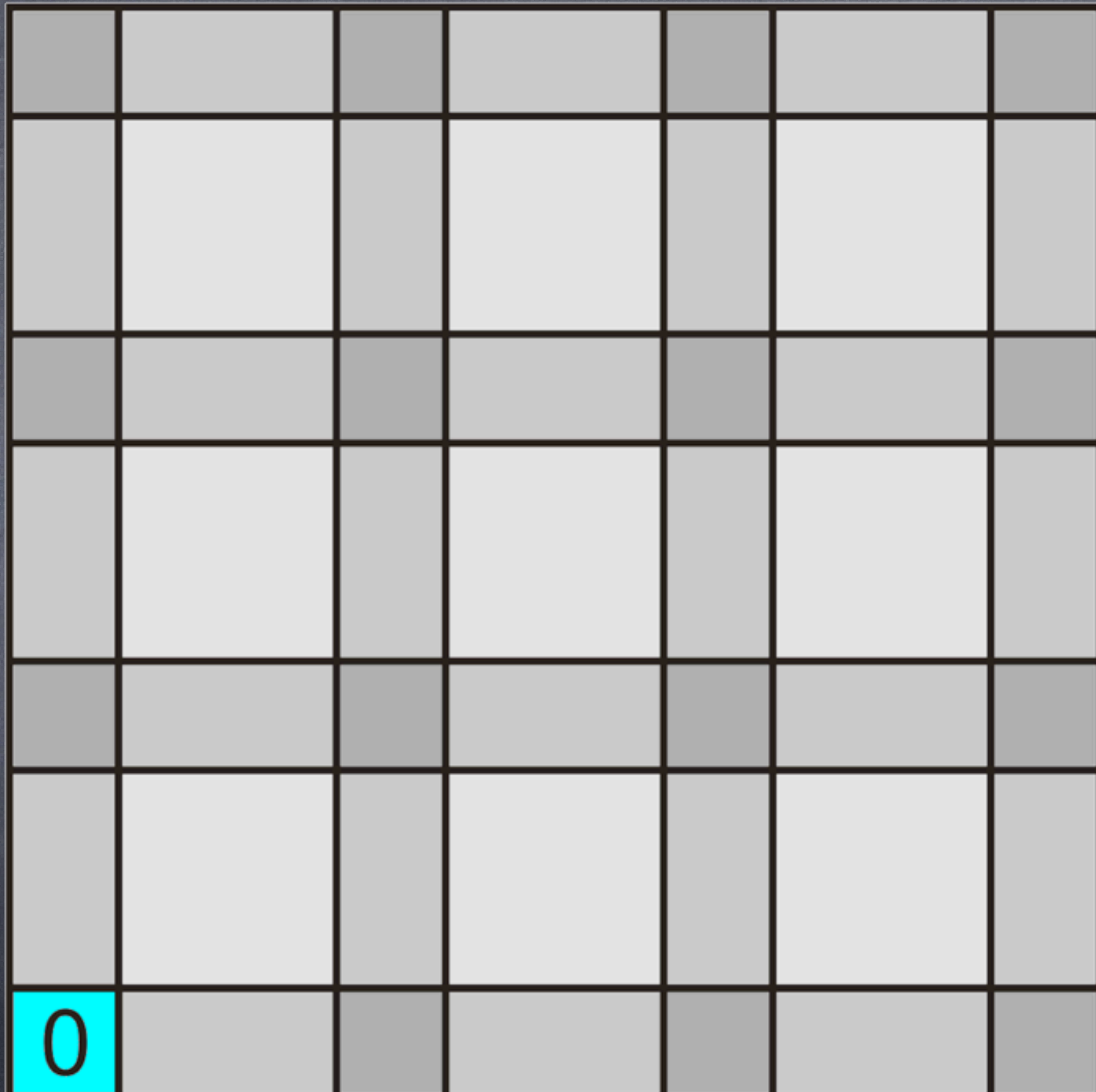
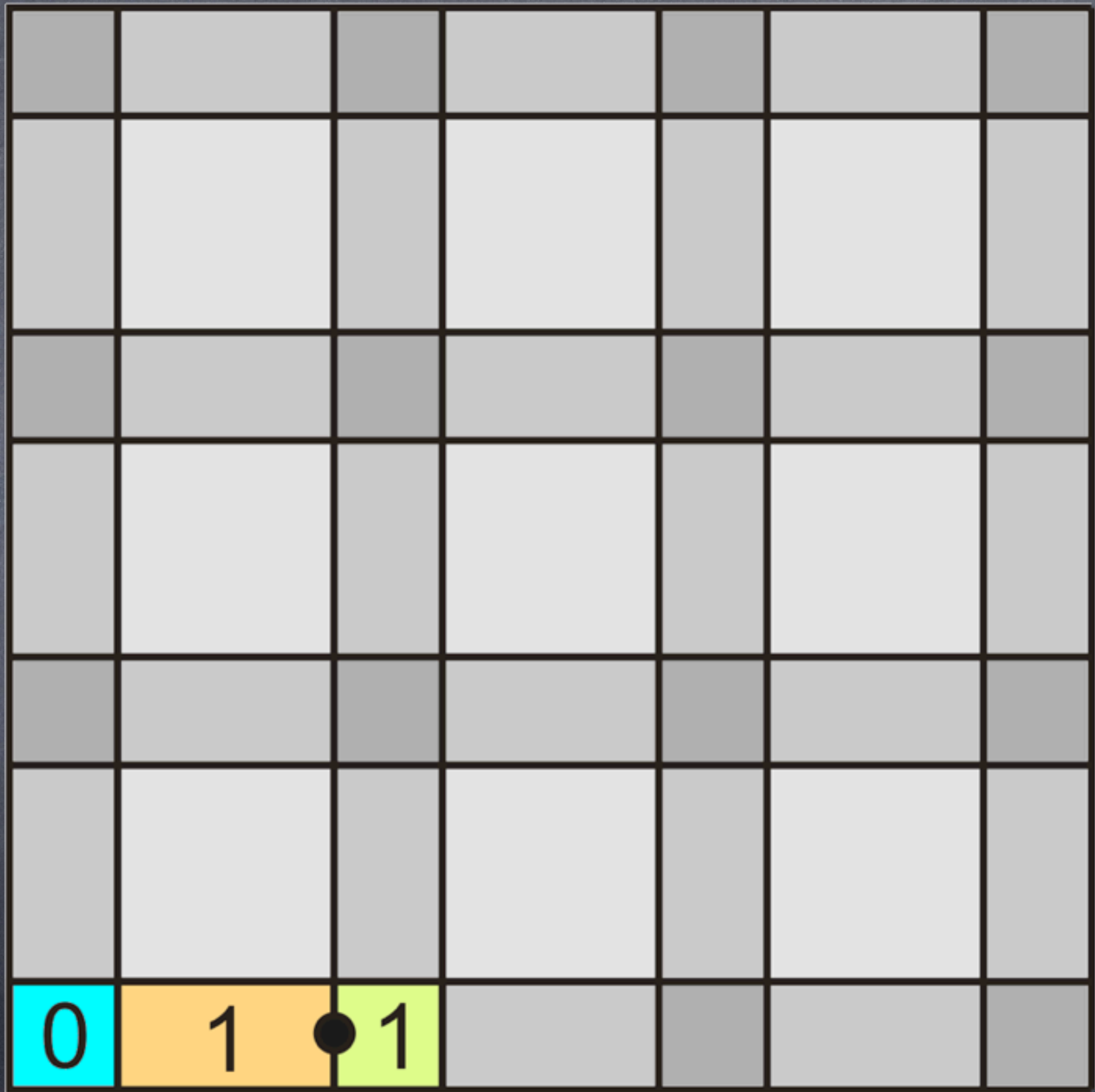


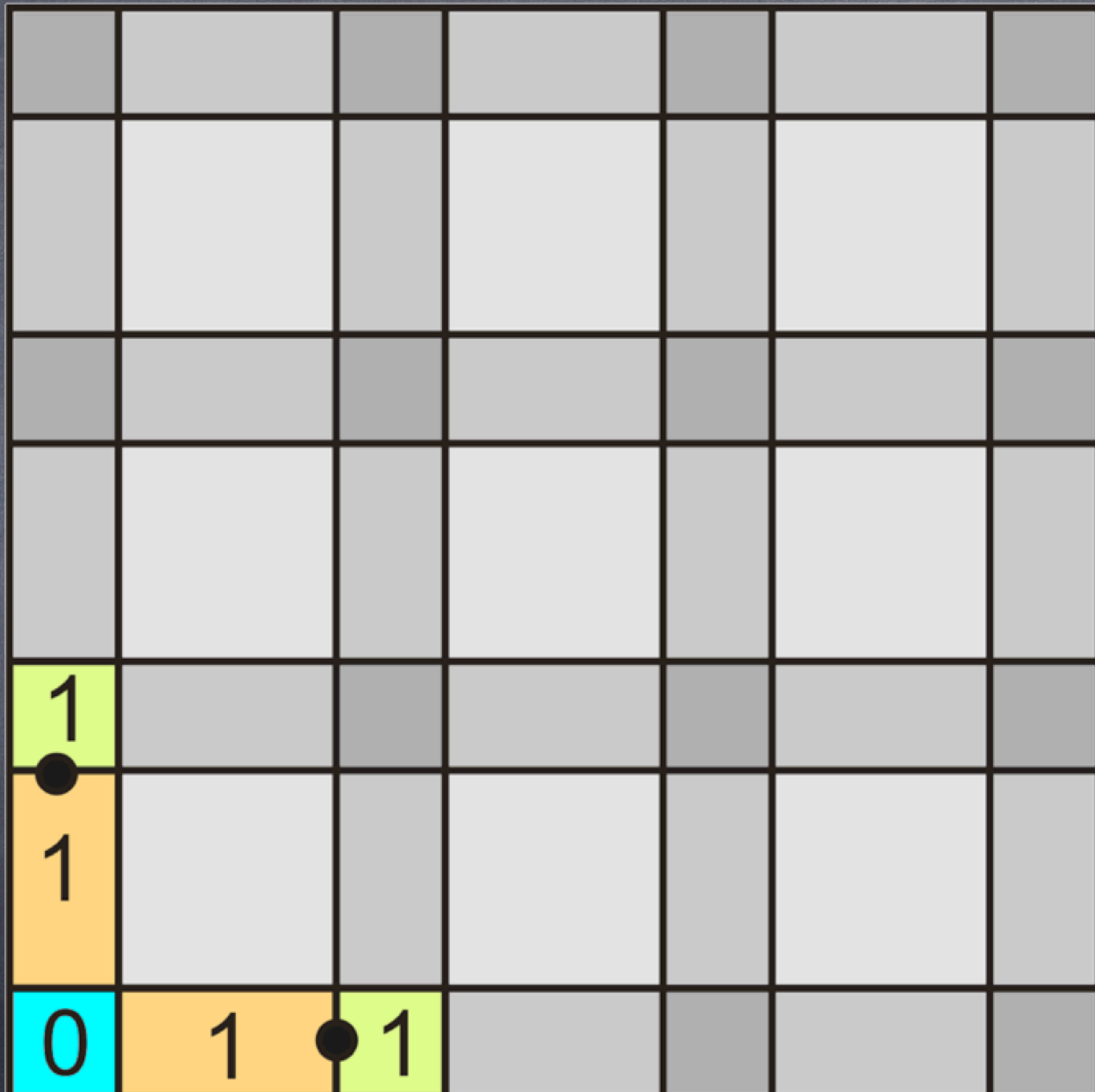
Illustration of the coreduction based decomposition algorithm at work

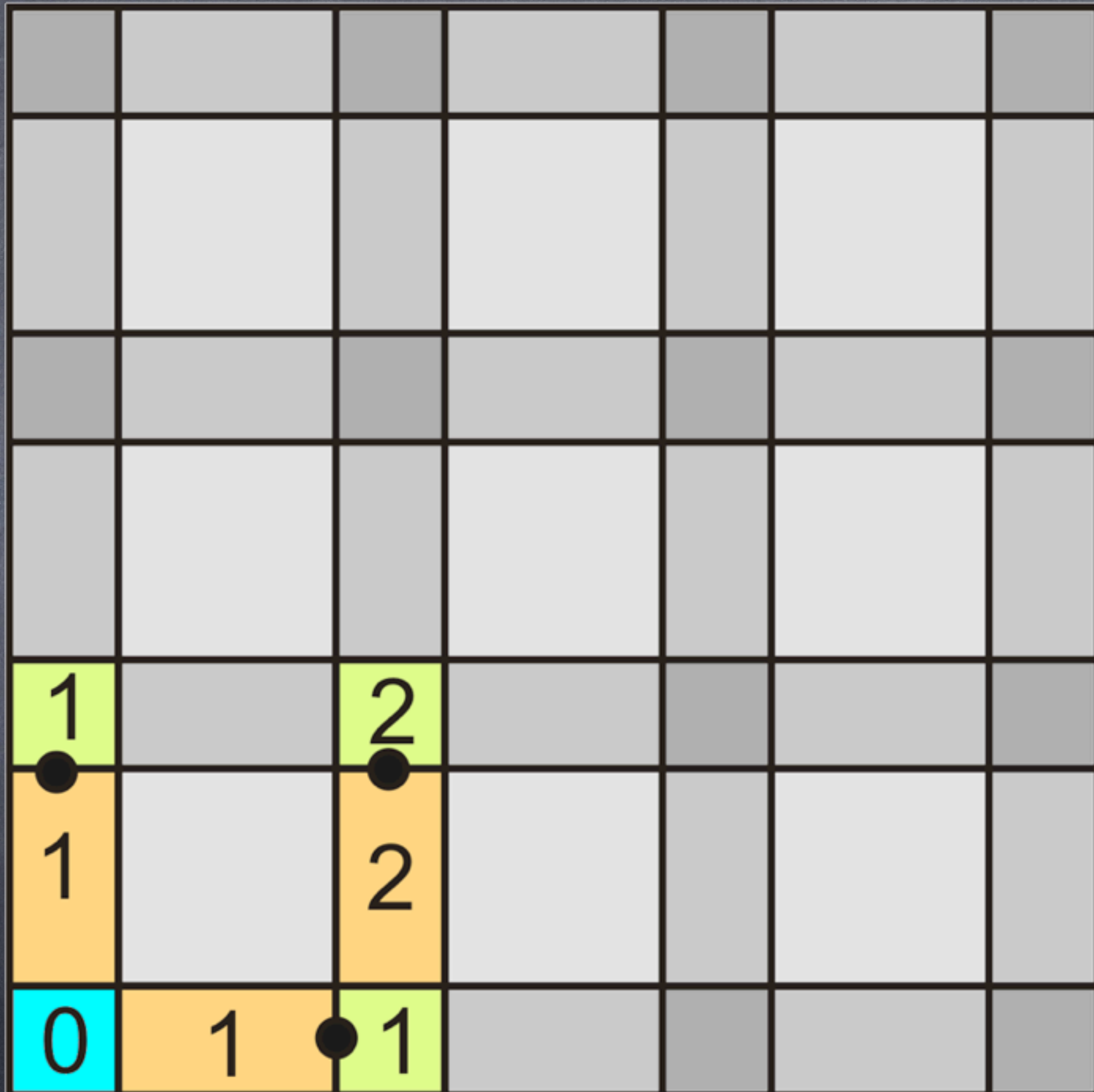





0


0
1
1


1
1
0
1
1

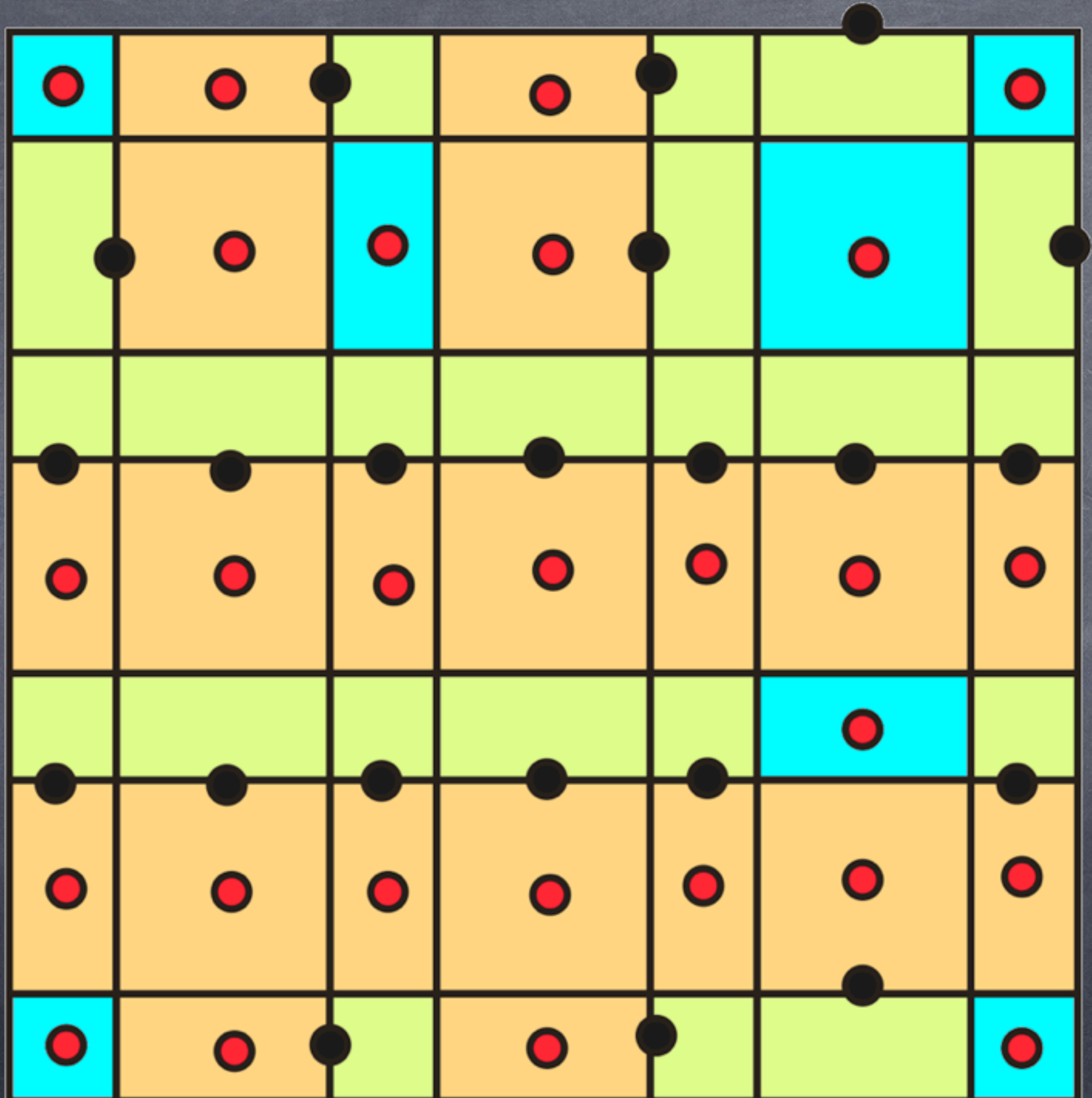

1
2
1
2
0
1
1


1
3
2
1
3
2
0
1
1


0 1 1 2 2 0
2 4 3 5 4 2
2 4 3 5 4 2
1 3 2 4 3 1
1 3 2 4 3 1
0 1 1 2 2 0


0 1 1 2 2 0
2 4 3 5 4 2
2 4 3 5 4 2
1 3 2 4 3 4 1
1 3 2 4 3 1
0 1 1 2 2 0

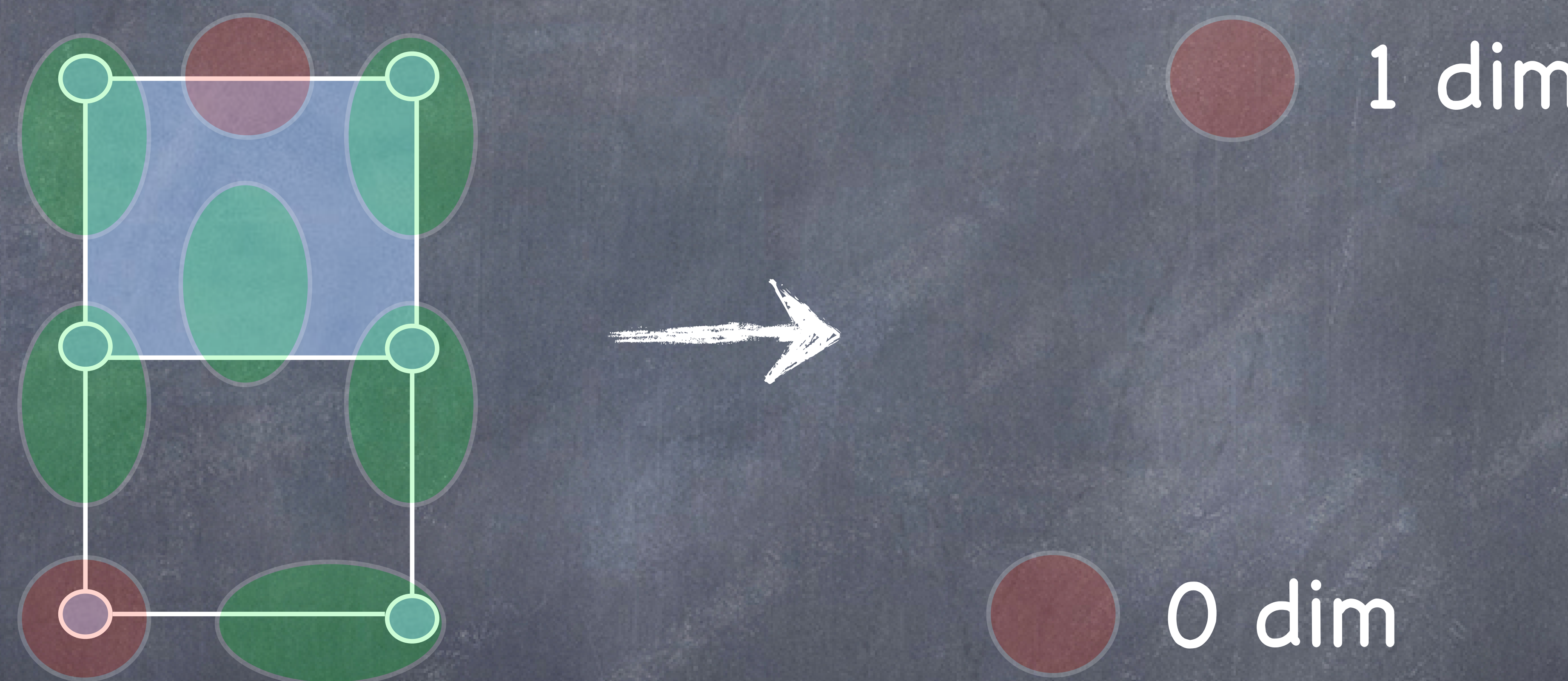

0 1 1 2 2 5 0
4
2 4 3 5 4 5 2
2 4 3 5 4 5 2
1 3 2 4 3 4 1
1 3 2 4 3 5 1
0 1 1 2 2 5 0



The Morse Complex

- The point of making the decomposition is to obtain a reduced, simpler complex with isomorphic homology groups.
- The Morse complex is a chain complex with chain groups $\mathcal{M}_k = \mathcal{R}(\mathcal{A}_k)$
- But what is the boundary operator Δ ?

That is, we keep only the aces:



and we need to know the boundary here

Forman's Formula: Connecting Orbit Viewpoint

- Consider the set of all V-paths from one critical cell to another of one less dimension
- For each path, determine the multiplicity
- Sum the multiplicity over all possible paths
- This yields the incidence numbers
- But... is this a tractable procedure?

Another solution: Deformation Viewpoint

- Deform chains under a flow given by discrete gradient vector field
- More generally, consider equivalence classes of chains modulo the boundaries of ``King" cells.
- These equivalence classes have a canonical representative which can be used to determine the Morse Boundary operator!

The key to the Morse Boundary problem: Canonicalization

Definition: A king chain is formal combination of king cells.

Definition: A canonical chain is a formal combination of ace and king cells.

Lemma. Consider the equivalence classes of chains modulo the boundaries of king chains. In each such class, there is a unique canonical representative. We call this representative the canonicalization of any chain in that class.

How do we canonicalize a chain in practice?

Canonicalization Algorithm

Loop until c is canonical

Choose a maximal queen Q in c

Let K = Q*.

$$c \leftarrow c - \frac{\langle Q, c \rangle}{\langle Q, \partial K \rangle} \partial K.$$

End Loop

Note: By choosing the maximal queen, we never process the same queen twice, via acyclicity assumption!

Notation:

Given a chain c, the unique king chain k such that c + dk is canonical is called $\gamma(c)$

Hence $\alpha := \mathrm{id} + \partial \circ \gamma$ canonicalizes.

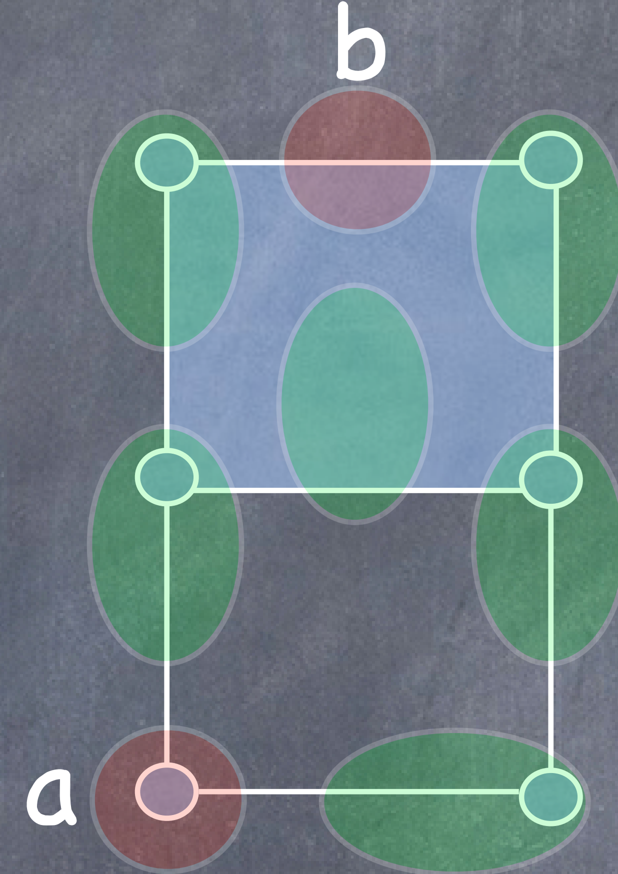
We also define $\beta := \mathrm{id} + \gamma \circ \partial$ which we call completion (it canonicalizes the boundary of a chain).

There are chain maps i and j between the original complex and the Morse complex naturally induced from inclusion and projection.

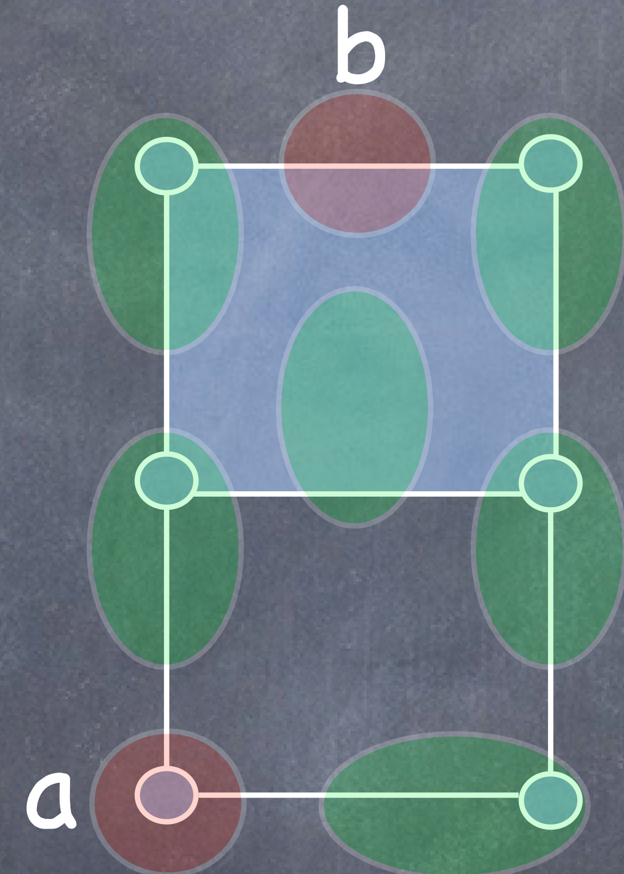
And now the Morse boundary:

$$\Delta = j \circ \alpha \circ \partial \circ i$$

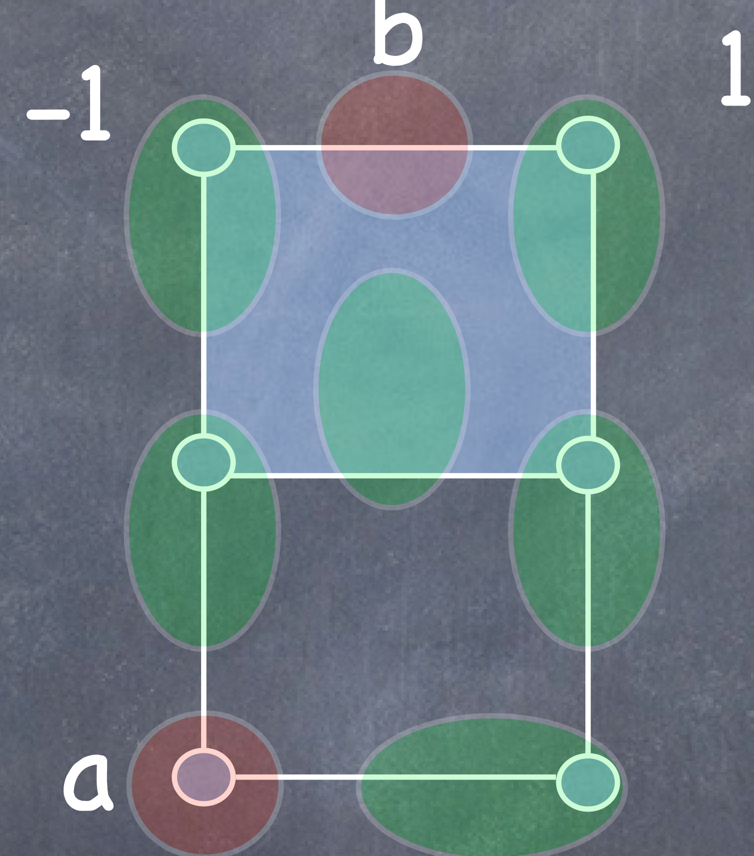
Let's use this formula to compute Δb



Let's use this formula to compute Δb

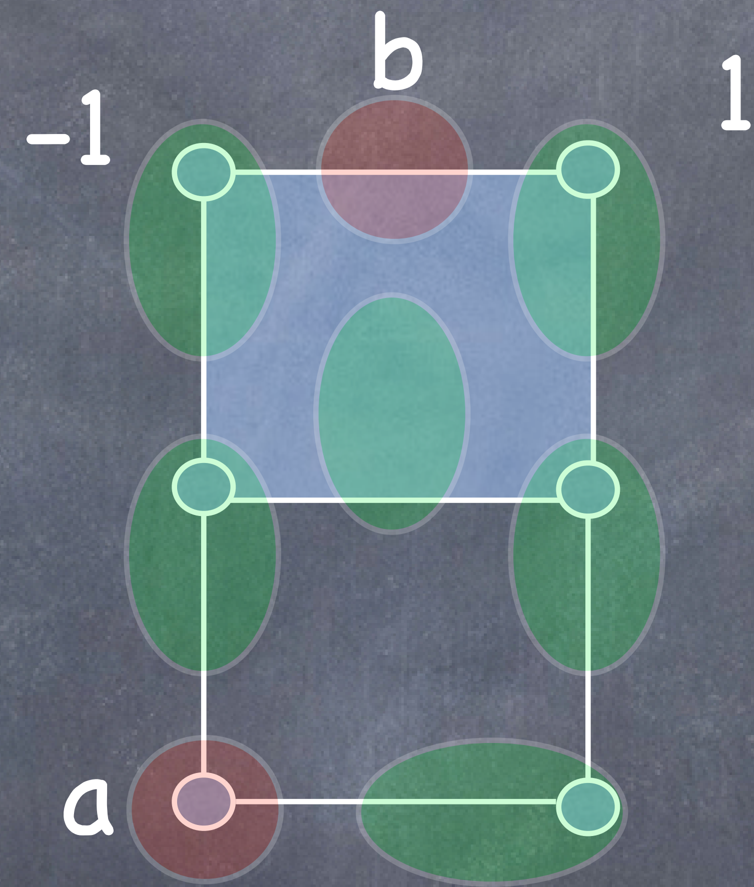


Let's use this formula to compute Δb



First compute ∂b

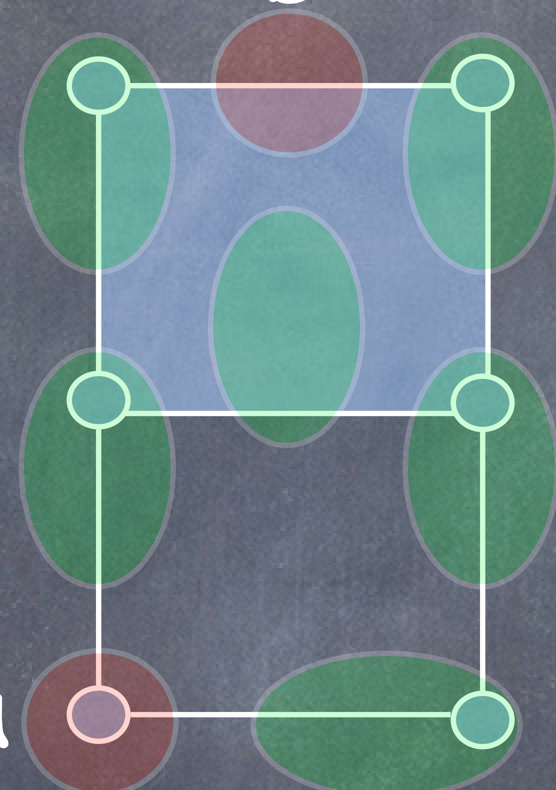
Let's use this formula to compute Δb



First compute ∂b

Now canonicalize.

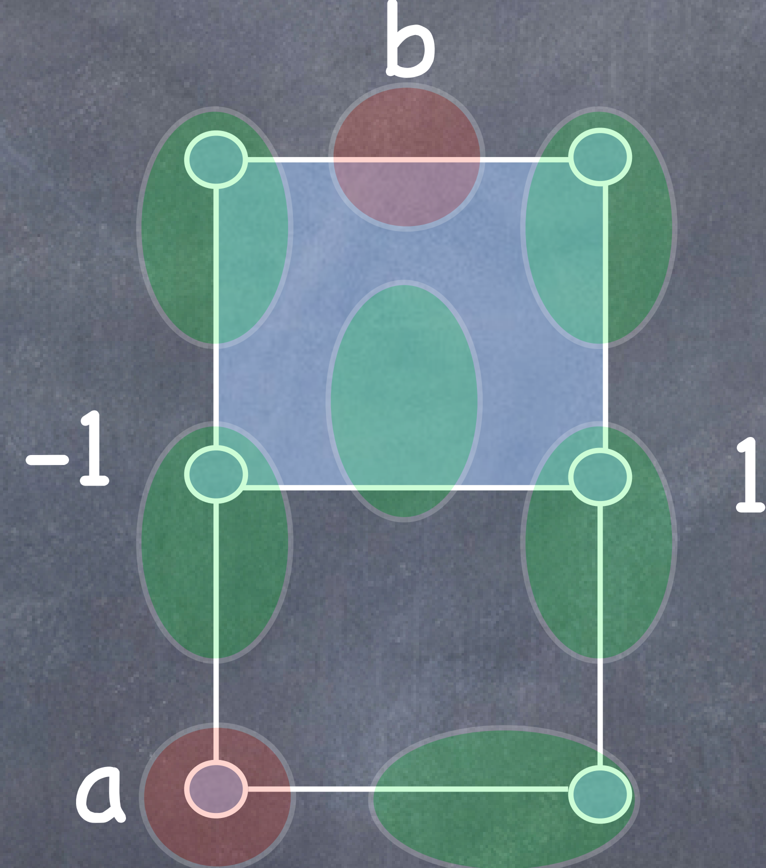
Let's use this formula to compute Δb



First compute ∂b

Now canonicalize.

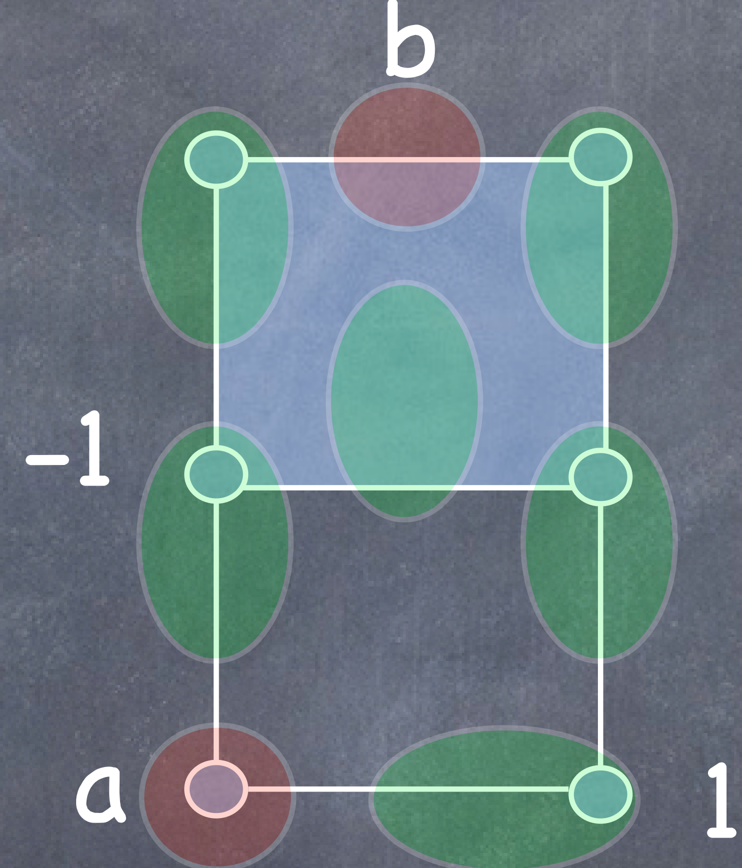
Let's use this formula to compute Δb



First compute ∂b

Now canonicalize.

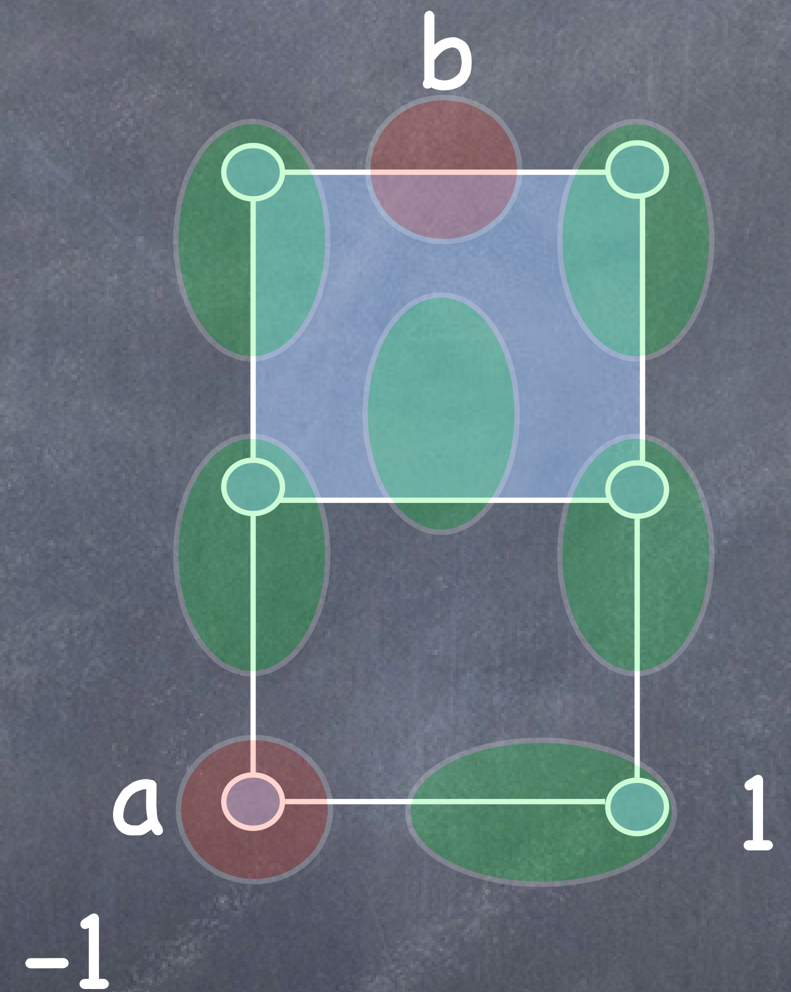
Let's use this formula to compute Δb



First compute ∂b

Now canonicalize.

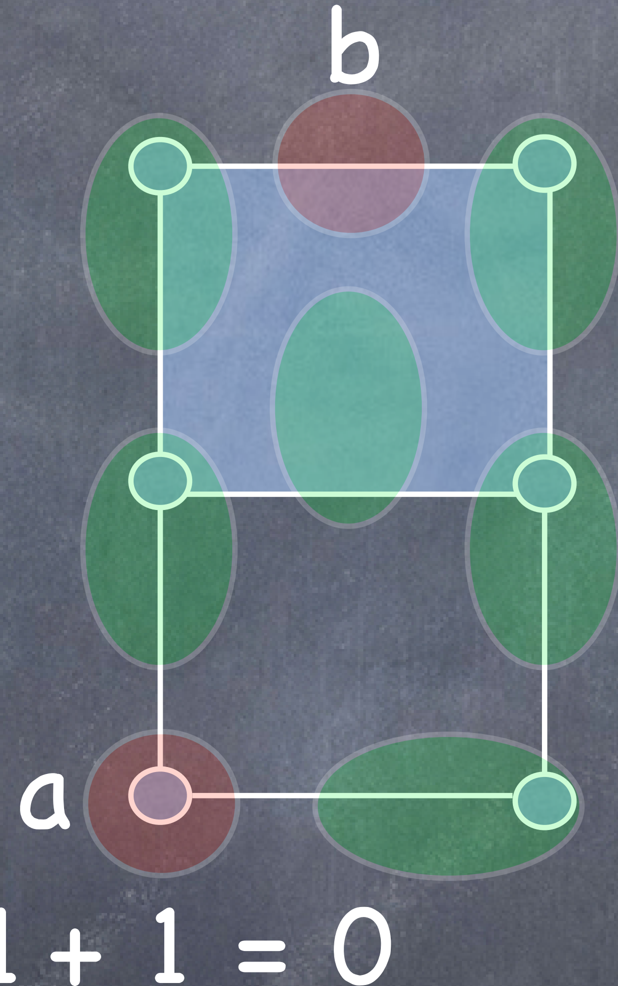
Let's use this formula to compute Δb



First compute ∂b

Now canonicalize.

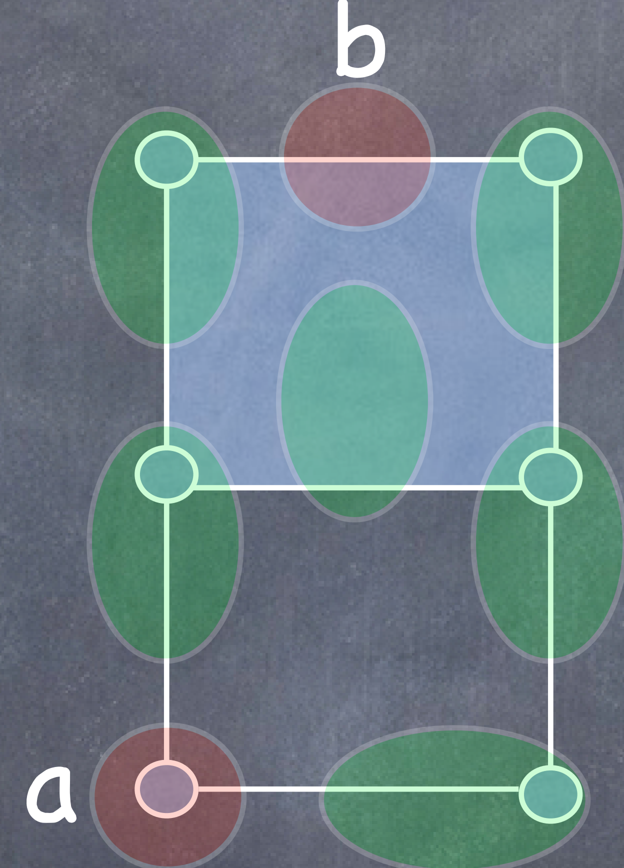
Let's use this formula to compute Δb



First compute ∂b

Now canonicalize.

Let's use this formula to compute Δb



First compute ∂b

Now canonicalize.

$$\Delta b = 0$$

Chain Equivalences

- We know the Morse complex has isomorphic homology groups because there are chain equivalences to and fro:

$$\phi : (\mathcal{C}, \partial) \to (\mathcal{M}, \Delta)$$

$$\psi : (\mathcal{M}, \Delta) \to (\mathcal{C}, \partial)$$

- We can directly exhibit these chain equivalences using our language of canonicalization and completion:

$$\phi := j \circ \alpha$$

where $$\psi := \beta \circ i$$

$$\Delta := \phi \circ \partial \circ \psi$$

Practical uses of chain equivalences:

- We can not only compute homology using the reduced Morse complex, but we can lift homology generators via a chain equivalence.
- It is possible to solve the "preboundary" equation $\partial c = b$ for c given b via an iterative approach:

$$P_{\mathcal{C}} = \gamma + \psi \circ P_{\mathcal{M}} \circ \phi$$

Experimental Results

- Several Implementations:
- CHOMP-CR
- CHOMP-DMT
- CHOMP-CR+DMT
- REDHOM-CR
- REDHOM-CR+DMT

Computer Experiments with CHOMP version

Dimension	Size	CR Time	DMT Time
4	8.5 x 10^6	9.82 sec	2.54 sec
4	40.8 x 10^6	560 sec	13.59 sec
5	3.8 x 10^6	91.6 sec	1.98 sec
5	5.8 x 10^6	458 sec	4.11 sec
6	3.0 x 10^6	42.5 sec	3.26 sec
6	7.2 x 10^6	2346 sec	8.51 sec

Randomly Generated Cubical Complexes

Cubical Complexes for Selected Spaces

	$T \times S^1$	$(S^1)^3$	$S^1 \times K$	$T \times T$
dim	5	6	6	6
size in millions	0.07	0.10	0.40	2.36
H_0	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$
H_1	$\mathbb{Z}^3$	0	$\mathbb{Z}^2 + \mathbb{Z}_2$	$\mathbb{Z}^4$
H_2	$\mathbb{Z}^3$	0	$\mathbb{Z} + \mathbb{Z}_2$	$\mathbb{Z}^6$
H_3	$\mathbb{Z}$	$\mathbb{Z}$		$\mathbb{Z}^4$
H_4				$\mathbb{Z}$
Linbox::Smith	130	350	> 600	> 600
RedHom::Shave+**Linbox**::Smith	0.5	0.1	2.2	> 600
ChomP	1.3	1.7	10	56
RedHom::CR	0.03	0.04	0.26	2.5
ChomP::DMT	0.06	0.15	1.6	5.9
ChomP::CR+DMT	0.04	0.16	1.7	3
RedHom::CR+DMT	0.02	0.08	0.5	1.1

Cahn-Hillard Equation

	P0001	P0050	P0100
dim	3	3	3
size in millions	75.56	73.36	71.64
H_0	$\mathbb{Z}^7$	$\mathbb{Z}^2$	$\mathbb{Z}$
H_1	$\mathbb{Z}^{6554}$	$\mathbb{Z}^{2962}$	$\mathbb{Z}^{1057}$
H_2	$\mathbb{Z}^2$		
Linbox::Smith	> 600	> 600	> 600
RedHom::Shave+**Linbox**::Smith	> 600	> 600	> 600
ChomP	400	360	310
RedHom::CR	36	34	33
ChomP::DMT	110	110	100
ChomP::CR+DMT	45	43	42
RedHom::CR+DMT	26	25	24

Random Cubical Sets

	d4s8f50	d4s12f50	d4s16f50	d4s20f50
dim	4	4	4	4
size in millions	0.07	0.34	1.04	2.48
H_0	$\mathbb{Z}^2$	$\mathbb{Z}^2$	$\mathbb{Z}^2$	$\mathbb{Z}^2$
H_1	$\mathbb{Z}^2$	$\mathbb{Z}^{17}$	$\mathbb{Z}^{30}$	$\mathbb{Z}^{51}$
H_2	$\mathbb{Z}^{174}$	$\mathbb{Z}^{1389}$	$\mathbb{Z}^{5510}$	$\mathbb{Z}^{15401}$
H_3	$\mathbb{Z}^2$	$\mathbb{Z}^{15}$	$\mathbb{Z}^{71}$	$\mathbb{Z}^{179}$
Linbox::Smith	120	> 600	> 600	> 600
RedHom::Shave+**Linbox**::Smith	4	> 600	> 600	> 600
ChomP	1	8.3	41	170
RedHom::CR	0.08	1.4	15	140
ChomP::DMT	0.05	0.38	1.8	5.3
ChomP::CR+DMT	0.03	0.16	0.56	1.4
RedHom::CR+DMT	0.03	0.16	0.58	2.9

Simplicial Complexes

	random set	Björner set	S^5
dim	4	2	5
size in millions	4.8	1.9	4.3
H_0	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$
H_1	$\mathbb{Z}^{39}$	0	0
H_2	$\mathbb{Z}^{84}$	$\mathbb{Z}$	0
H_3			0
H_4			$\mathbb{Z}$
ChomP	830	310	2100
RedHom::CR+DMT	65	11	100

Acknowledgements

Coauthors: Konstantin Mischaikow,
Marian Mrozek, Vidit Nanda

CHOMP-DMT Implementation: Shaun Harker
http://code.google.com/p/chomp-rutgers/
sharker@math.rutgers.edu

REDHOM Implementation: Hubert Wagner, Mateusz Juda, Pawel Dlotko, Marian Mrozek
http://capd.ii.uj.edu.pl/

References

1. Marian Mrozek and Bogdan Batko. *Coreduction Homology Algorithm*. Discrete and Computational Geometry. **41**(1) (2009) 96-118
2. Robin Forman. *Morse Theory for Cell Complexes.* Advances in Mathematics. **134** (1998) 90-145
3. Tomasz Kaczyński, Konstantin Mischaikow, Marian Mrozek. *Computational Homology*, Applied Mathematical Sciences, **157**, Springer 2004
4. Dmitry Kozlov. *Combinatorial Algebraic Topology.* Algorithms and Computation in Mathematics, **21**, Springer 2008
5. Shaun Harker, Konstantin Mischaikow, Marian Mrozek, and Vidit Nanda. "Discrete Morse theoretic algorithms for computing homology of complexes and maps." *Foundations of Computational Mathematics* (2013): 1-34.

Thanks!