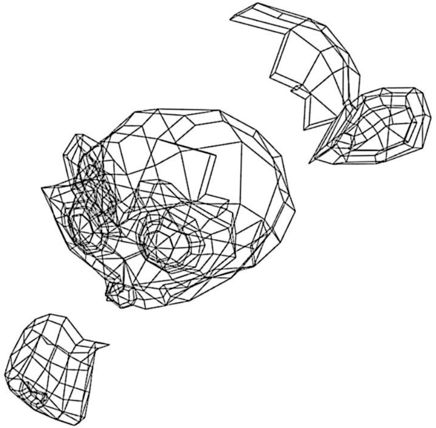
RHARBAOUI Wassim – ALAYRANGUES Sylvie – LIENHARDT Pascal – PELTIER Samuel

LOCAL COMPUTATION OF HOMOLOGY VARIATIONS RELATED TO CELLS MERGING

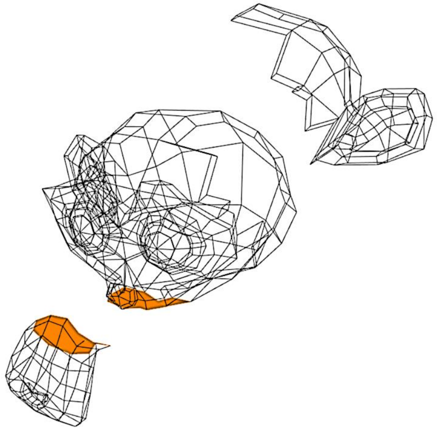
Theoretical results and implementation issues

1: CONTEXT AND OBJECTIVES

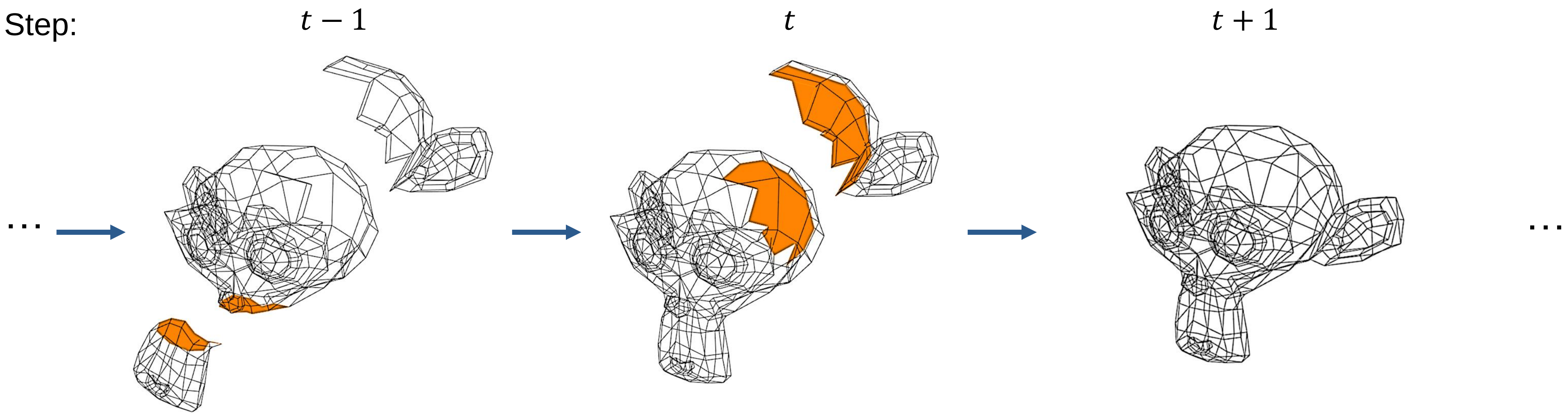
1 CONTEXT AND OBJECTIVES



1 CONTEXT AND OBJECTIVES

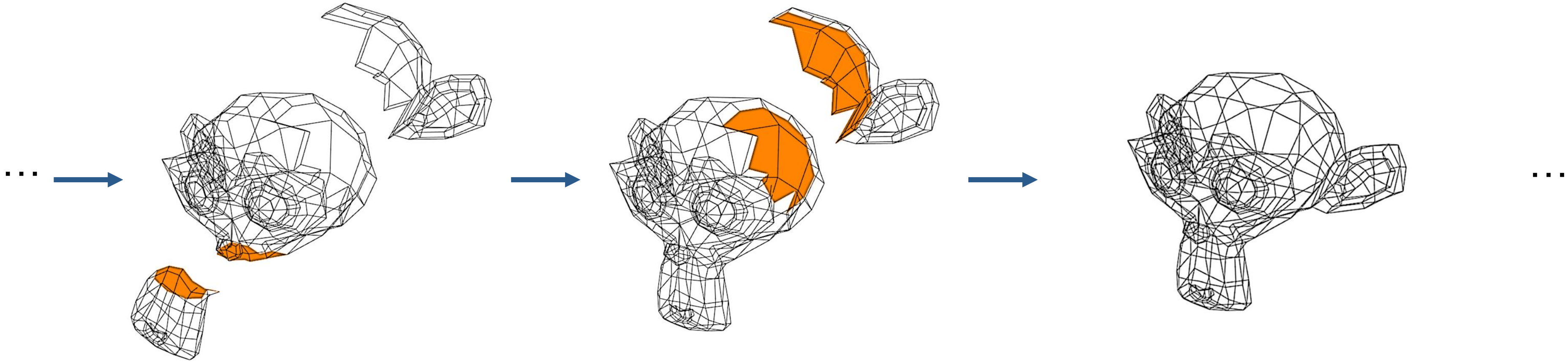


1 CONTEXT AND OBJECTIVES



1 CONTEXT AND OBJECTIVES

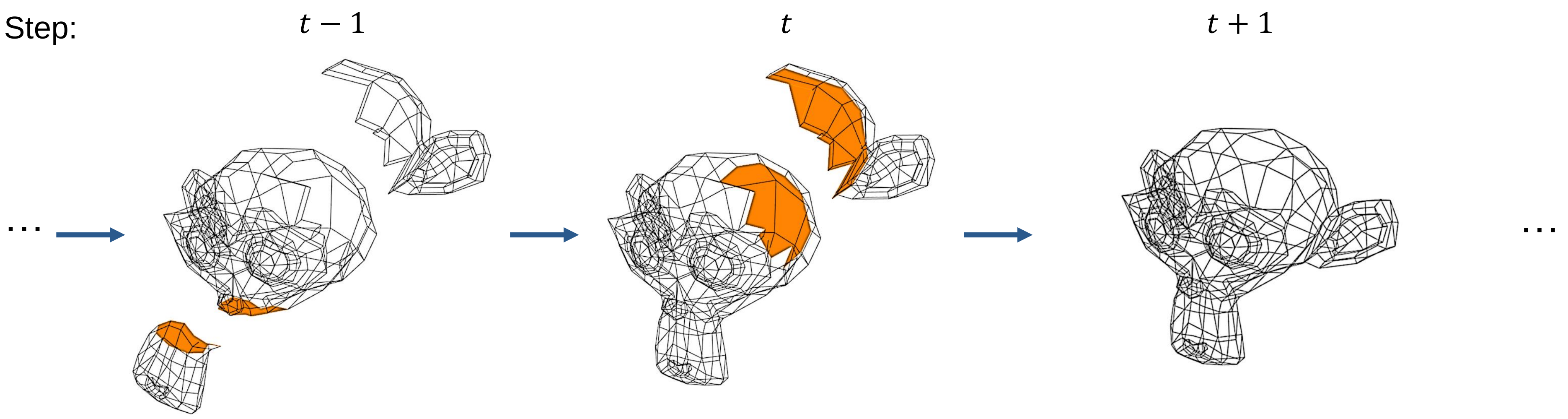
Step: $t-1$ t $t+1$



Operation: merging cells

1

CONTEXT AND OBJECTIVES

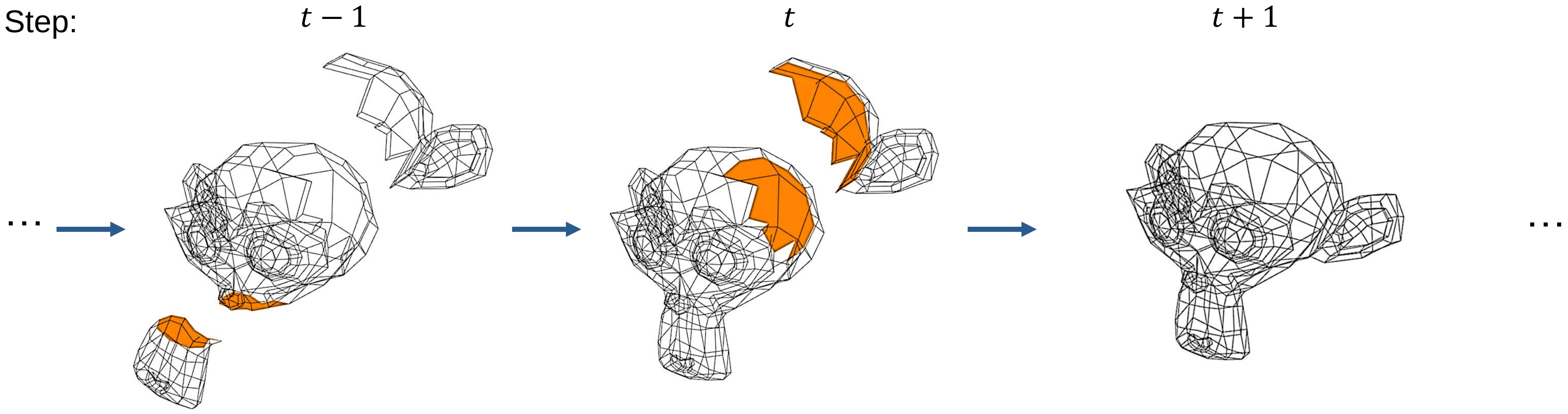


Operation: merging cells

At each step: control of the object construction according to computed information

1

CONTEXT AND OBJECTIVES



Operation: merging cells

At each step: control of the object construction according to computed information

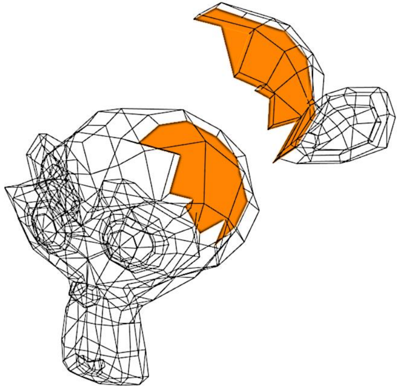
=> Compute homology efficiently

2: STATE OF THE ART

2

SMITH NORMAL FORM (SNF)

t



... → → ...

At each step:

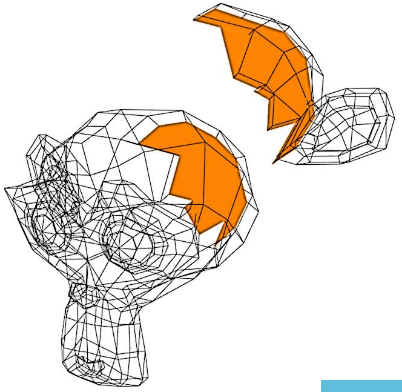
Boundary matrices

- Munkres, James R. (1984). *Elements of algebraic topology*.
- Hatcher, Allen. (2002). *Algebraic topology.*

2

SMITH NORMAL FORM (SNF)

t

… →  → …

At each step:

Boundary matrices → SNF Matrices

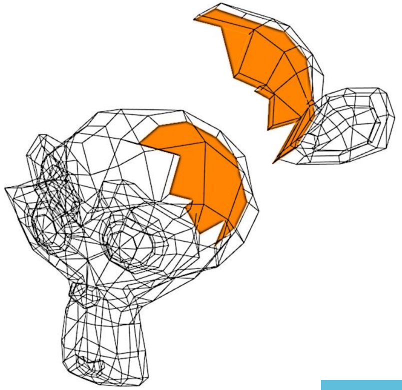
Smith Normal Form

- Munkres, James R. (1984). *Elements of algebraic topology*.
- Hatcher, Allen. (2002). *Algebraic topology.*

2

SMITH NORMAL FORM (SNF)

t

… →  → …

At each step:

Boundary matrices → (Smith Normal Form) → SNF Matrices → (Retrieving homology) → Homology groups + Generators

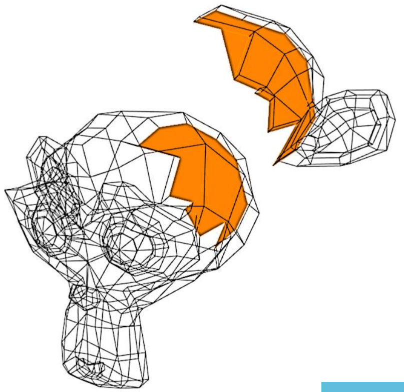
- Munkres, James R. (1984). *Elements of algebraic topology*.
- Hatcher, Allen. (2002). *Algebraic topology.*

2

SMITH NORMAL FORM (SNF)

COMPLEXITY: SIZE OF THE OBJECT

t

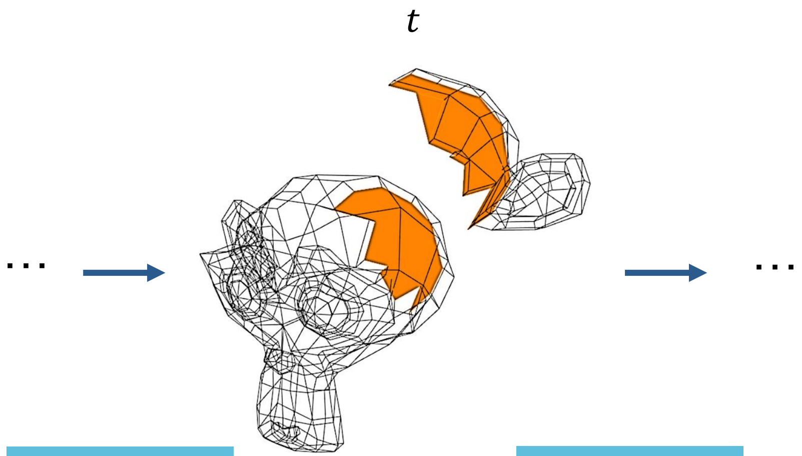
… →  → …

At each step:

Boundary matrices → (Smith Normal Form) → SNF Matrices → (Retrieving homology) → Homology groups + Generators

- Munkres, James R. (1984). *Elements of algebraic topology*.
- Hatcher, Allen. (2002). *Algebraic topology.*

2 REDUCTION



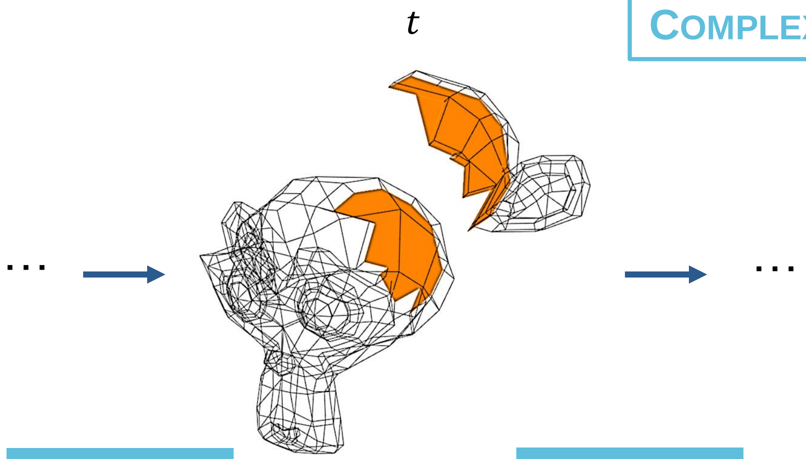
At each step:

Boundary matrices → (Reduction) → Reduced matrices → (Smith Normal Form) → SNF Matrices → (Retrieving homology) → Homology groups + Generators

- Forman, R. (1998). *Morse theory for cell complexes.*
- Gonzalez-Lorenzo, A., Bac, A., Mari, J. L., & Real, P. (2017). *Allowing cycles in discrete Morse theory.*

2 REDUCTION

COMPLEXITY: SIZE OF THE OBJECT



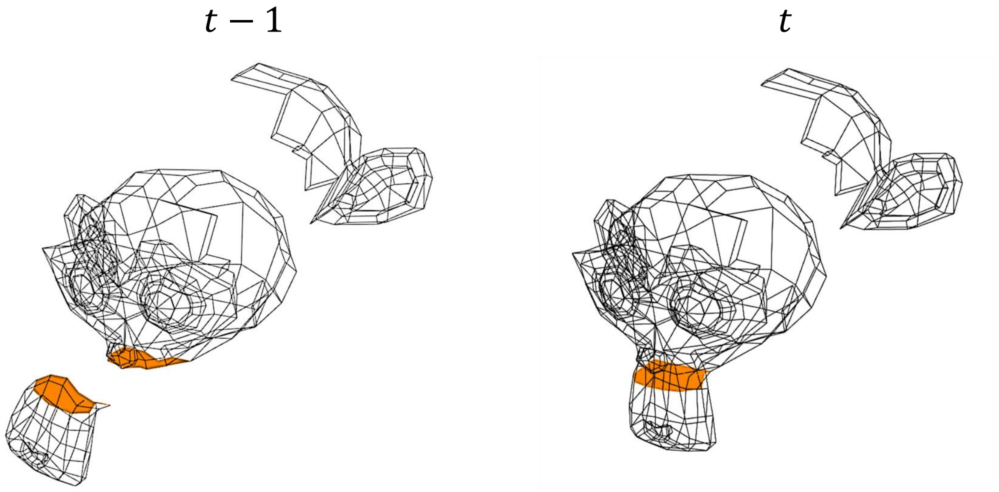
At each step:

Boundary matrices → (Reduction) → Reduced matrices → (Smith Normal Form) → SNF Matrices → (Retrieving homology) → Homology groups + Generators

- Forman, R. (1998). *Morse theory for cell complexes.*
- Gonzalez-Lorenzo, A., Bac, A., Mari, J. L., & Real, P. (2017). *Allowing cycles in discrete Morse theory.*

2

PERSISTENT HOMOLOGY



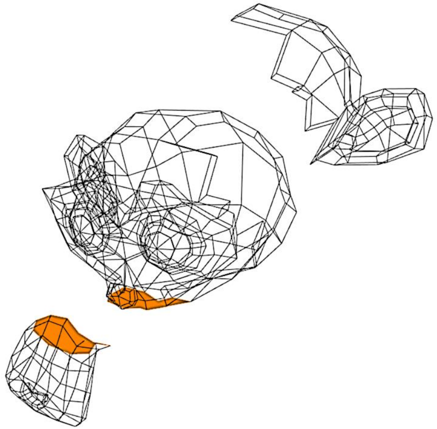
- Zomorodian, A., & Carlsson, G. (2005). *Computing persistent homology.*
- Edelsbrunner, H., Letscher, D., & Zomorodian, A. (2000). *Topological persistence and simplification*.

2

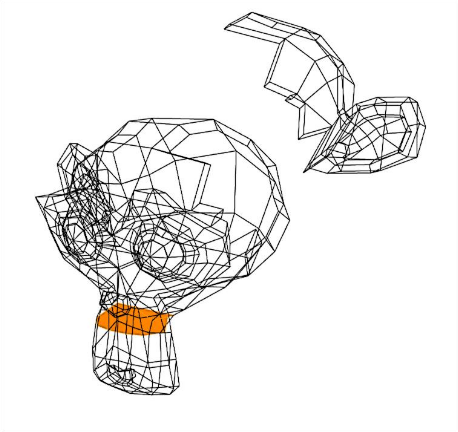
PERSISTENT HOMOLOGY

NO FILTRATION WHEN MERGING CELLS

$t-1$



t

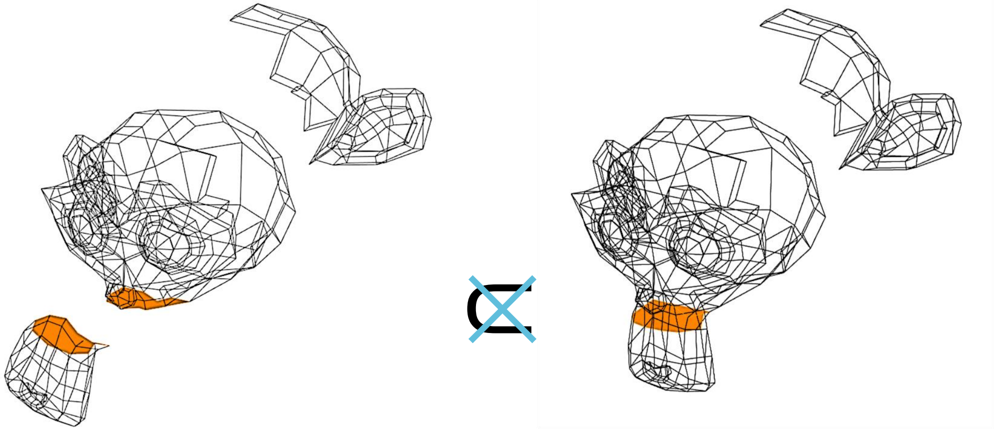


- Zomorodian, A., & Carlsson, G. (2005). *Computing persistent homology.*
- Edelsbrunner, H., Letscher, D., & Zomorodian, A. (2000). *Topological persistence and simplification*.

2 PERSISTENT HOMOLOGY

NO FILTRATION WHEN MERGING CELLS

$t-1$ $\qquad\qquad$ t



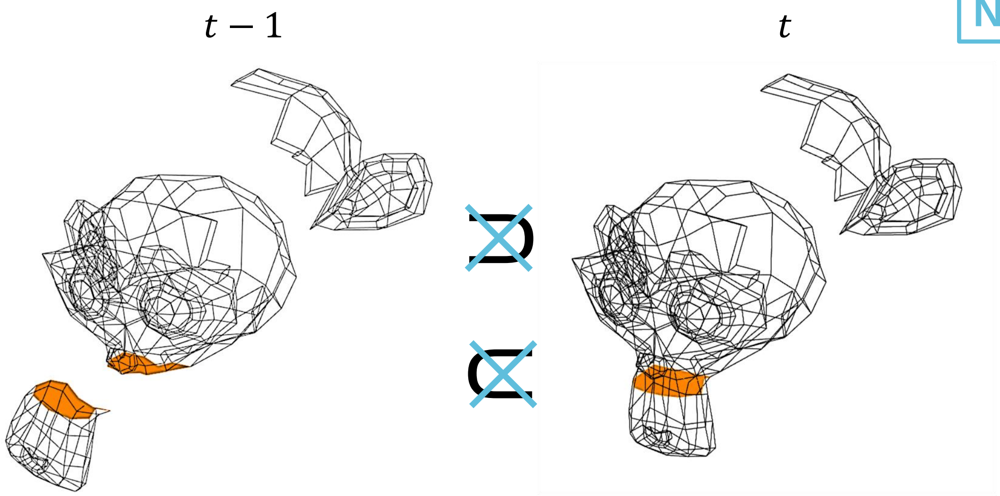
- From $\boldsymbol{t-1}$ to $\boldsymbol{t}$: merged cells

- Zomorodian, A., & Carlsson, G. (2005). *Computing persistent homology.*
- Edelsbrunner, H., Letscher, D., & Zomorodian, A. (2000). *Topological persistence and simplification*.

2

PERSISTENT HOMOLOGY

NO FILTRATION WHEN MERGING CELLS



- From $\boldsymbol{t-1}$ to $\boldsymbol{t}$: merged cells
- From $\boldsymbol{t}$ to $\boldsymbol{t-1}$: modified boundary

- Zomorodian, A., & Carlsson, G. (2005). *Computing persistent homology.*
- Edelsbrunner, H., Letscher, D., & Zomorodian, A. (2000). *Topological persistence and simplification*.

2

SHORT EXACT SEQUENCE BASED METHOD

Goal

- Tracking homology variations induced by a merging operation

- D. Boltcheva, D. Canino, S. Merino Aceituno, J.-C. Leon, L. De Floriani, and F. Hetroy. An iterative algorithm for homology computation on simplical shapes.
- Rubio, J., & Sergeraert, F. (2012). *Constructive homological algebra and applications*.
- Alayrangues, S., Fuchs, L., Lienhardt, P., & Peltier, S. (2015). *Incremental Computation of the Homology of Generalized Maps: An Application of Effective Homology Results.*

2

SHORT EXACT SEQUENCE BASED METHOD

Goal

- Tracking homology variations induced by a merging operation
- Taking advantage of locality => complexity: size of the operated part

- D. Boltcheva, D. Canino, S. Merino Aceituno, J.-C. Leon, L. De Floriani, and F. Hetroy. An iterative algorithm for homology computation on simplical shapes.
- Rubio, J., & Sergeraert, F. (2012). *Constructive homological algebra and applications*.
- Alayrangues, S., Fuchs, L., Lienhardt, P., & Peltier, S. (2015). *Incremental Computation of the Homology of Generalized Maps: An Application of Effective Homology Results.*

2

SHORT EXACT SEQUENCE BASED METHOD

Goal

- Tracking homology variations induced by a merging operation
- Taking advantage of locality => complexity: size of the operated part

Contributions

- Theoretical and experimental complexity in the case of the merging cells operation

- D. Boltcheva, D. Canino, S. Merino Aceituno, J.-C. Leon, L. De Floriani, and F. Hetroy. An iterative algorithm for homology computation on simplical shapes.
- Rubio, J., & Sergeraert, F. (2012). *Constructive homological algebra and applications*.
- Alayrangues, S., Fuchs, L., Lienhardt, P., & Peltier, S. (2015). *Incremental Computation of the Homology of Generalized Maps: An Application of Effective Homology Results.*

2 SHORT EXACT SEQUENCE BASED METHOD

Goal

- Tracking homology variations induced by a merging operation
- Taking advantage of locality => complexity: size of the operated part

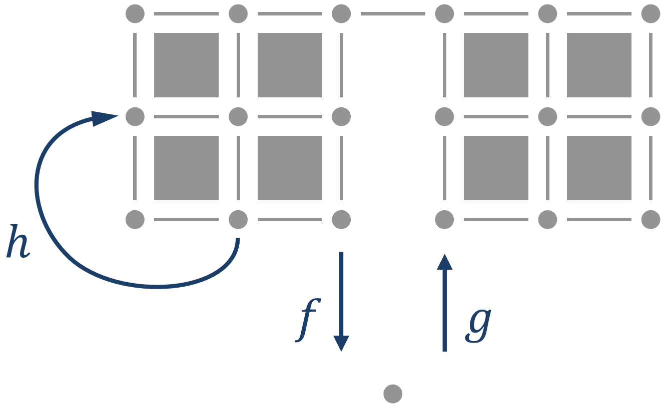
Contributions

- Theoretical and experimental complexity in the case of the merging cells operation
- Highlighting critical cases

- D. Boltcheva, D. Canino, S. Merino Aceituno, J.-C. Leon, L. De Floriani, and F. Hetroy. An iterative algorithm for homology computation on simplical shapes.
- Rubio, J., & Sergeraert, F. (2012). *Constructive homological algebra and applications*.
- Alayrangues, S., Fuchs, L., Lienhardt, P., & Peltier, S. (2015). *Incremental Computation of the Homology of Generalized Maps: An Application of Effective Homology Results.*

3: TOOLS

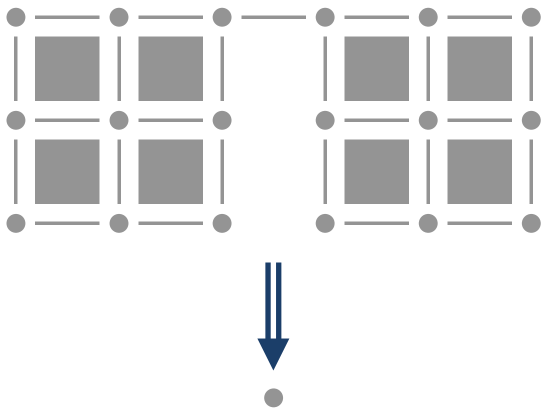
3 REDUCTION



Reducing object while preserving homology

- 3 morphisms h, f, g

REDUCTION



Object

Reduced object

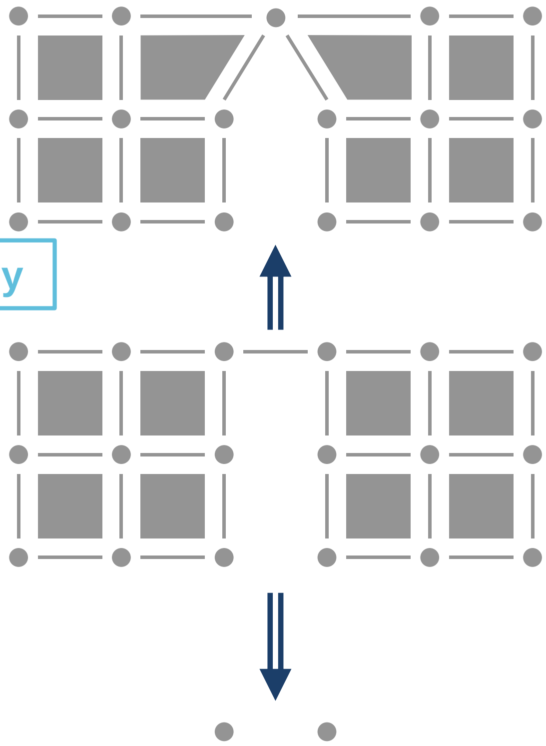
Reducing object while preserving homology

- 3 morphisms h, f, g

- Forman, R. (1998). *Morse theory for cell complexes.*
- Gonzalez-Lorenzo, A., Bac, A., Mari, J. L., & Real, P. (2017). *Allowing cycles in discrete Morse theory.*

3

HOMOLOGICAL EQUIVALENCE



3 objects with same homology

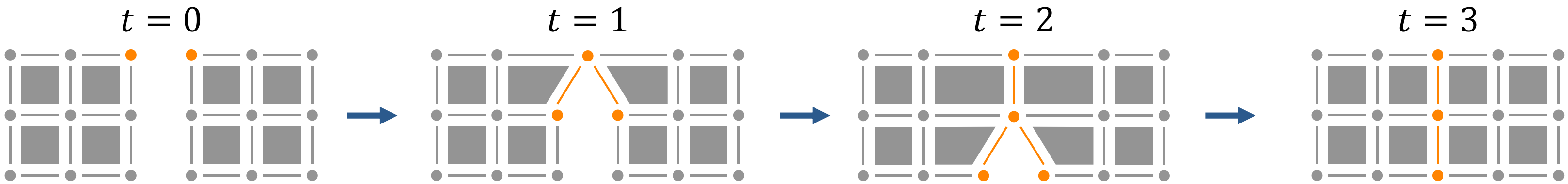
- 2 reductions

4: METHOD

4

METHOD: STUDIED OPERATION SET

3 operations, 4 construction steps

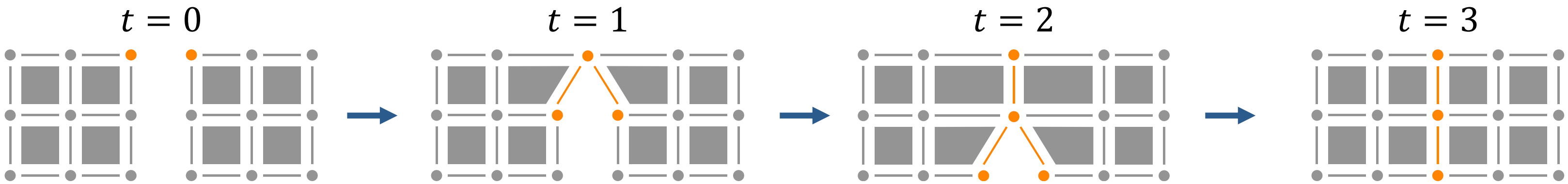


4

METHOD: PRINCIPLE

3 operations, 4 construction steps

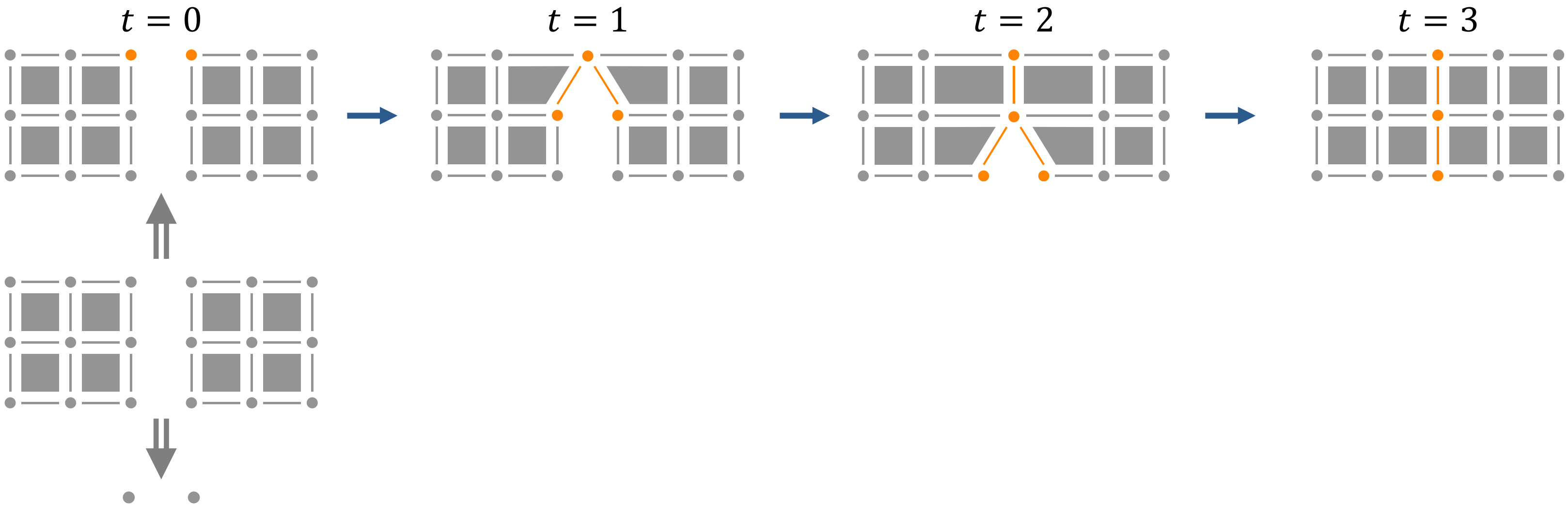
Maintain a homological equivalence



4 METHOD: PRINCIPLE

3 operations, 4 construction steps

Maintain a homological equivalence

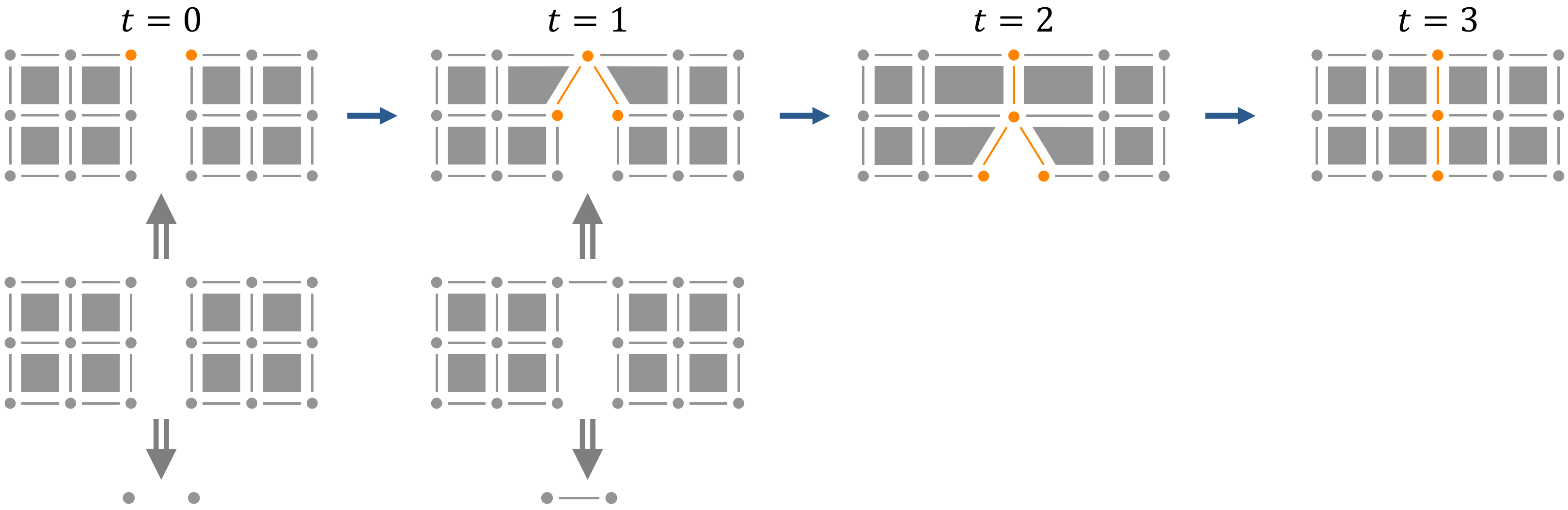


4

METHOD: PRINCIPLE

3 operations, 4 construction steps

Maintain a homological equivalence

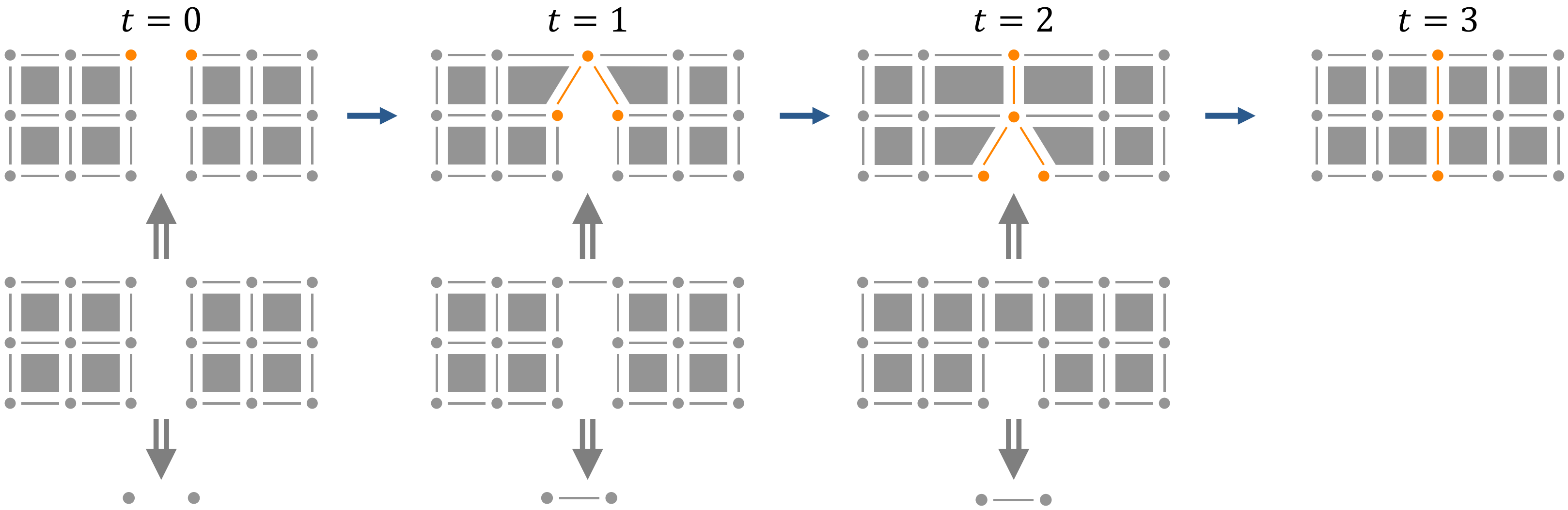


4

METHOD: PRINCIPLE

3 operations, 4 construction steps

Maintain a homological equivalence

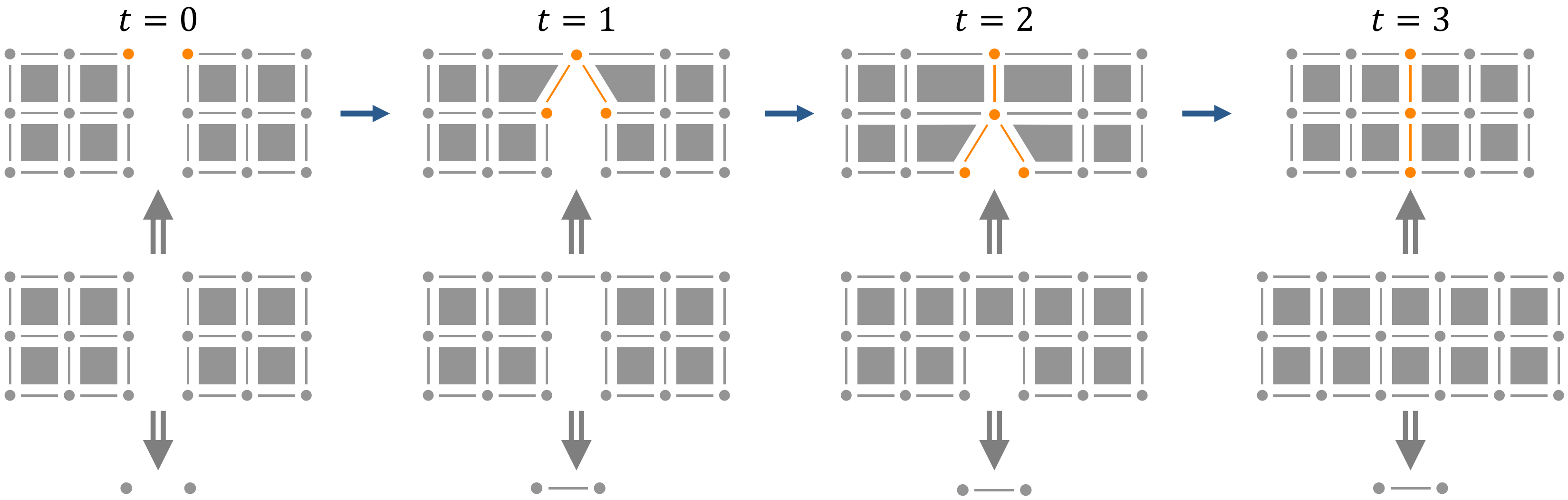


4

METHOD: PRINCIPLE

3 operations, 4 construction steps

Maintain a homological equivalence

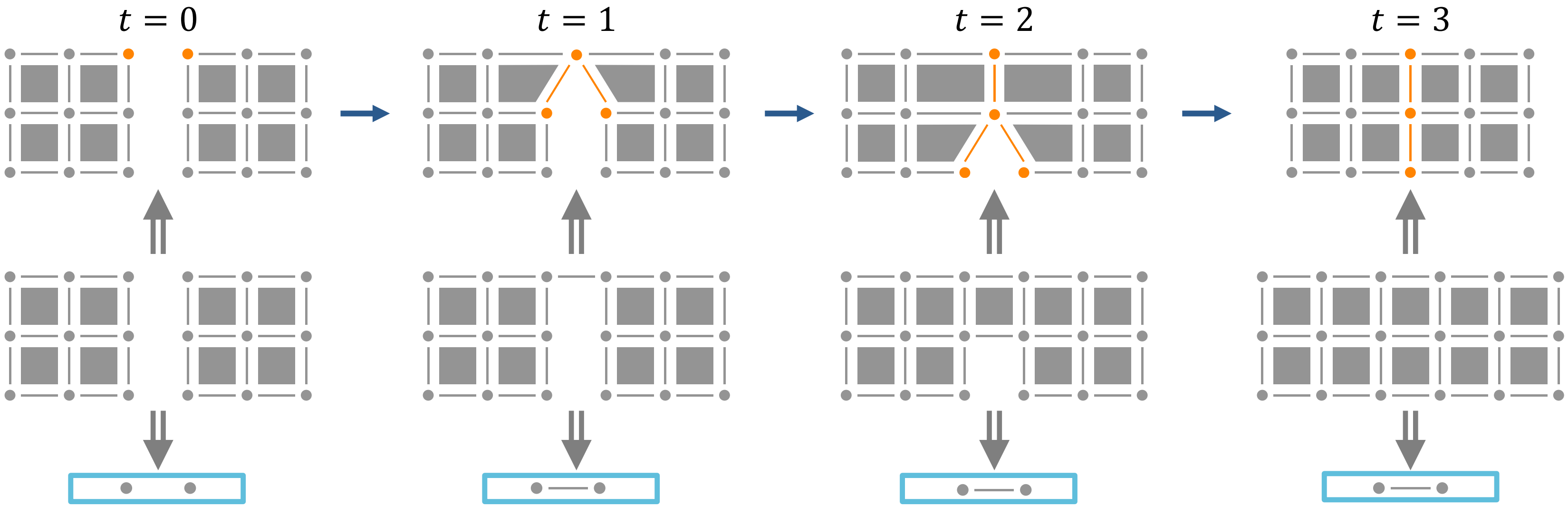


4

METHOD: PRINCIPLE

3 operations, 4 construction steps

Maintain a homological equivalence

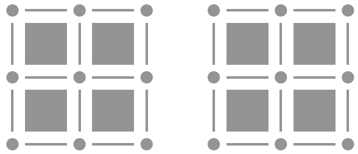


At each step : compute homology on reduced objects

4

METHOD: INITIALIZATION

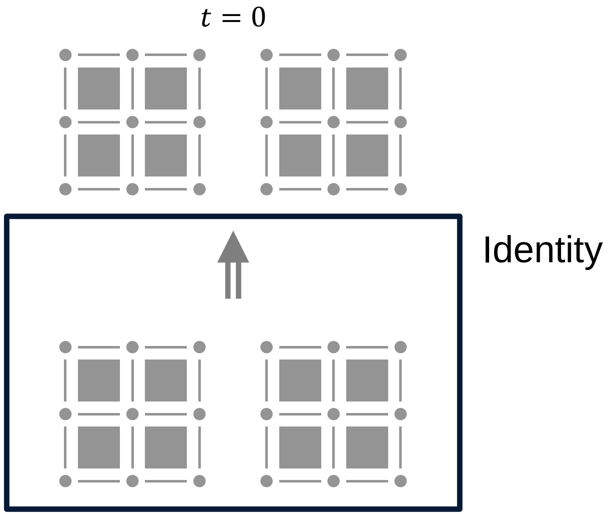
$t = 0$



Initial object

4 METHOD: INITIALIZATION

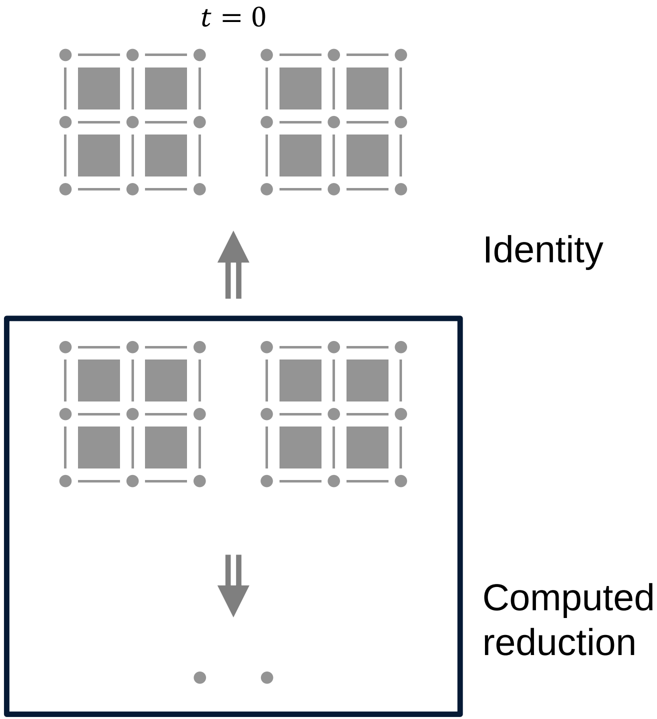
Initial object => build a homological equivalence



4

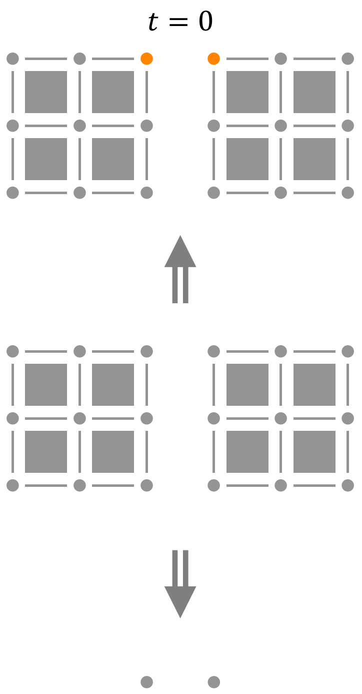
METHOD: INITIALIZATION

Initial object => build a homological equivalence



- Forman, R. (1998). *Morse theory for cell complexes.*
- Gonzalez-Lorenzo, A., Bac, A., Mari, J. L., & Real, P. (2017). *Allowing cycles in discrete Morse theory.*

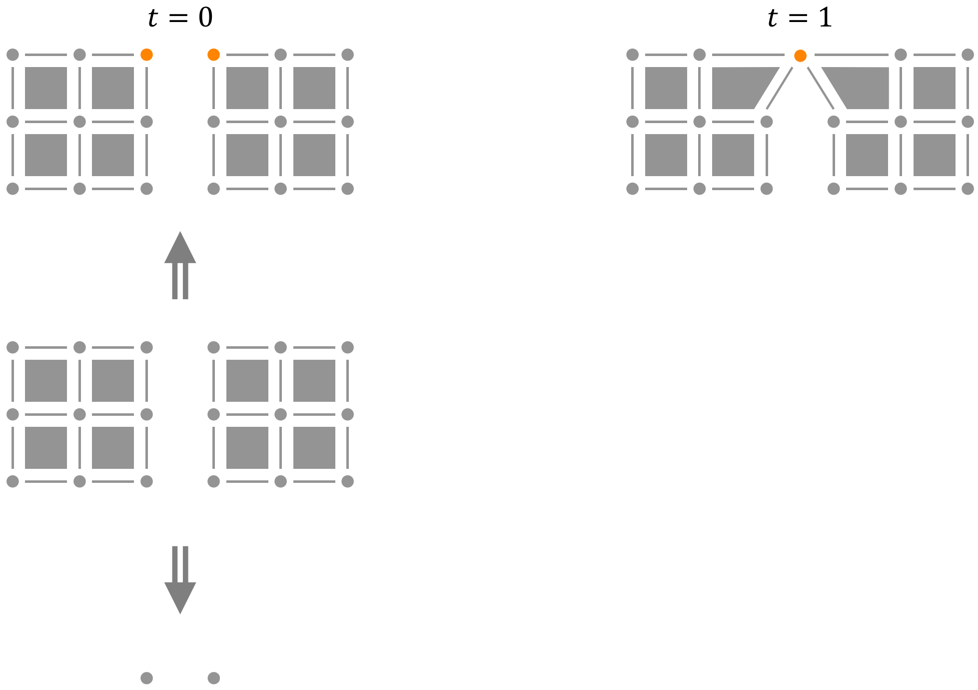
4 METHOD: WHEN MERGING CELLS



- ❑ Operation
- ❑ Initialization

4

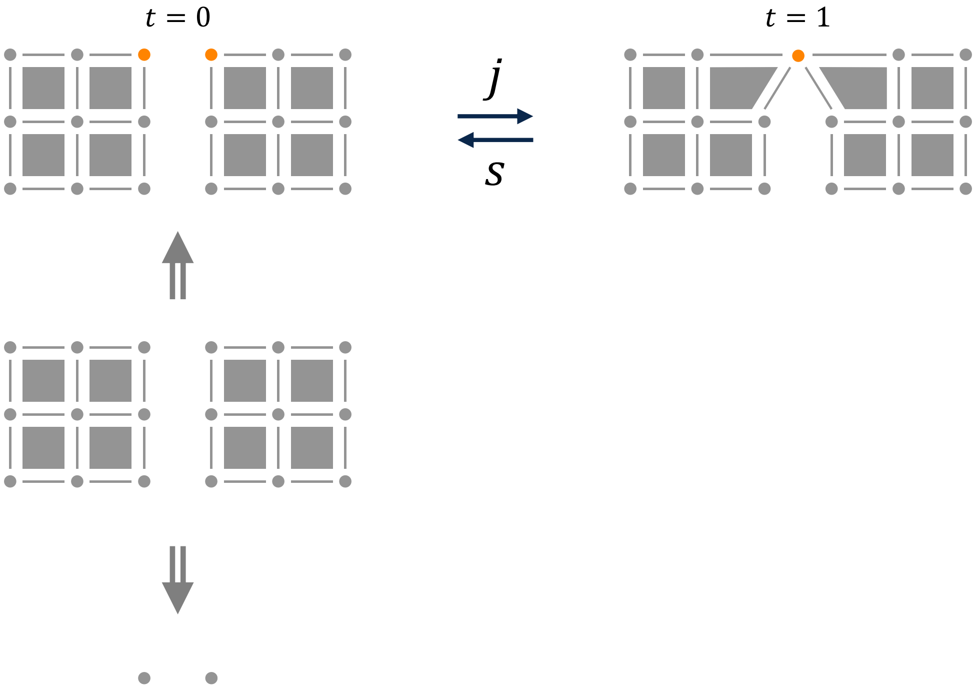
METHOD: WHEN MERGING CELLS



- ❑ Operation
- ❑ Initialization

4

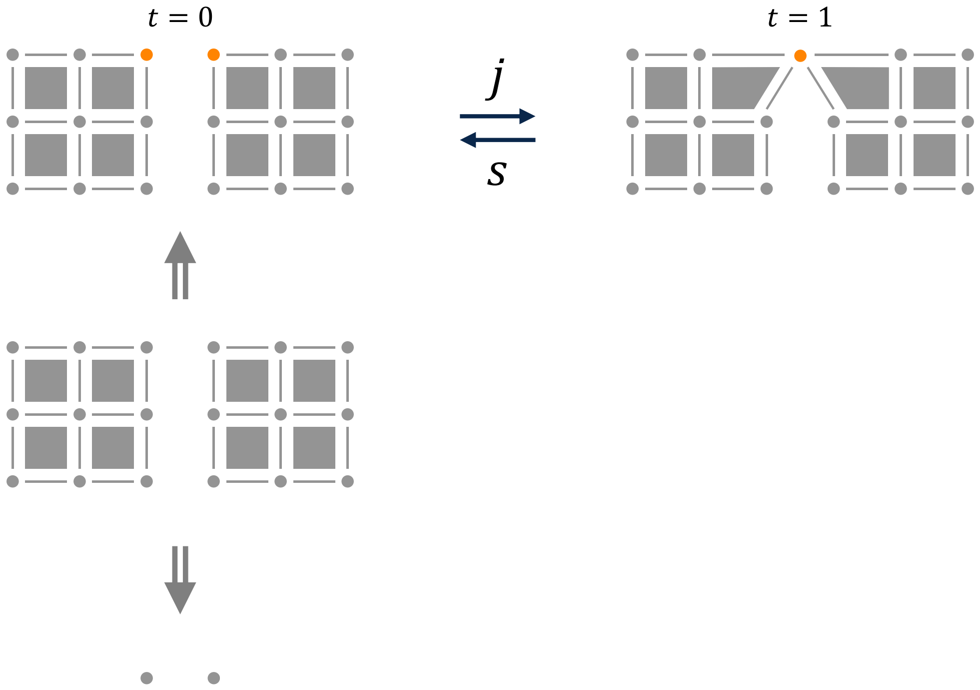
METHOD: WHEN MERGING CELLS



- Operation
- Initialization

4

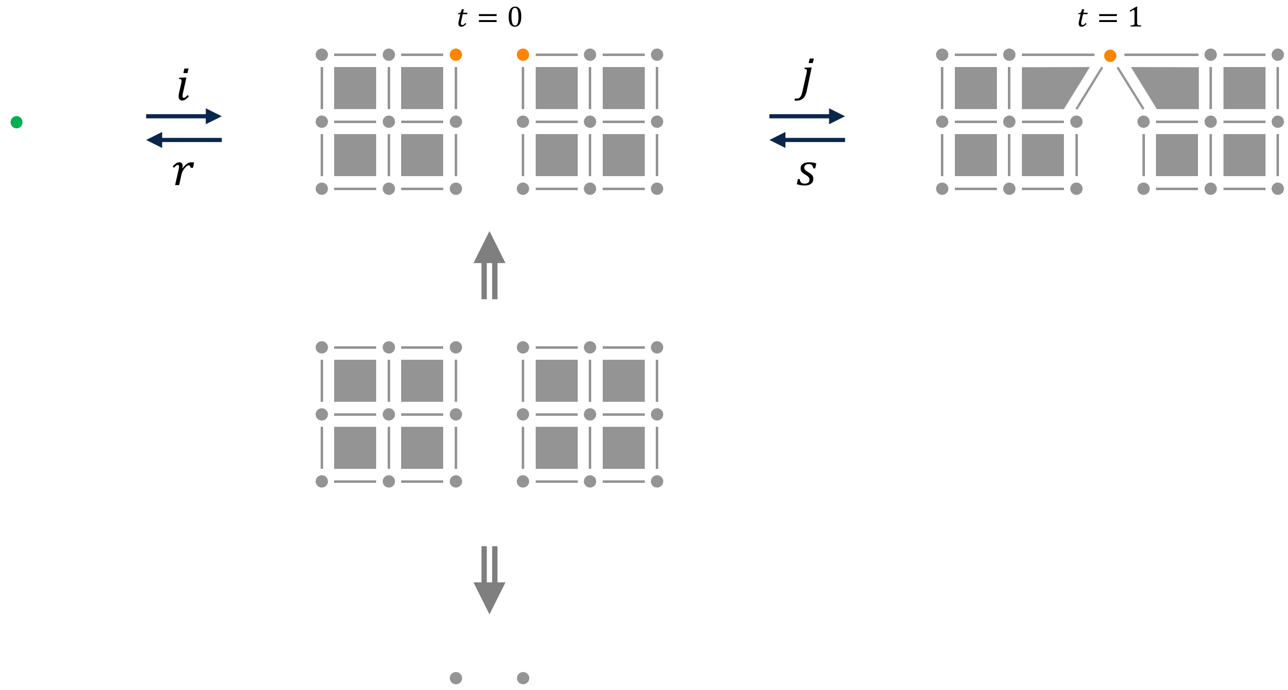
METHOD: WHEN MERGING CELLS



- ❑ Operation
- ❑ Initialization

4

METHOD: WHEN MERGING CELLS

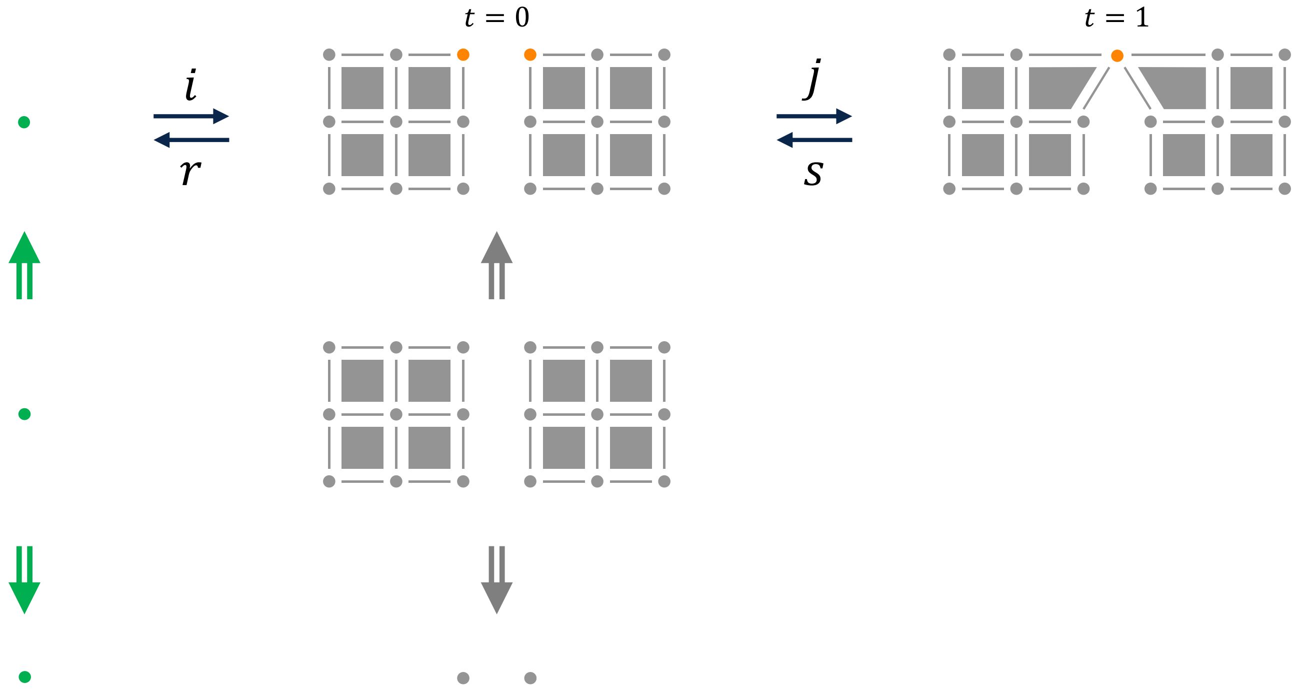


❑ Operation
❑ Initialization

- Rubio, J., & Sergeraert, F. (2012). *Constructive homological algebra and applications*.
- Alayrangues, S., Fuchs, L., Lienhardt, P., & Peltier, S. (2015). *Incremental Computation of the Homology of Generalized Maps: An Application of Effective Homology Results.*

4

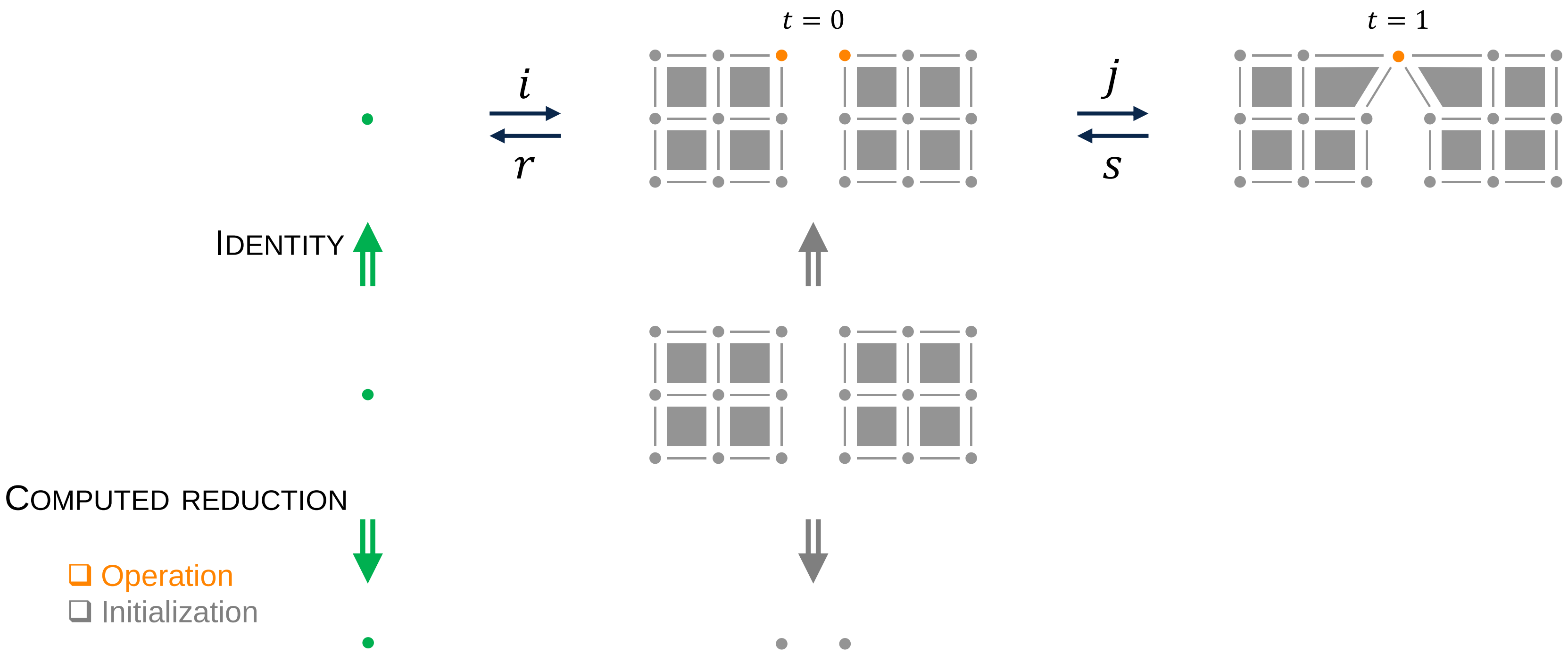
METHOD: WHEN MERGING CELLS



- Operation
- Initialization

4

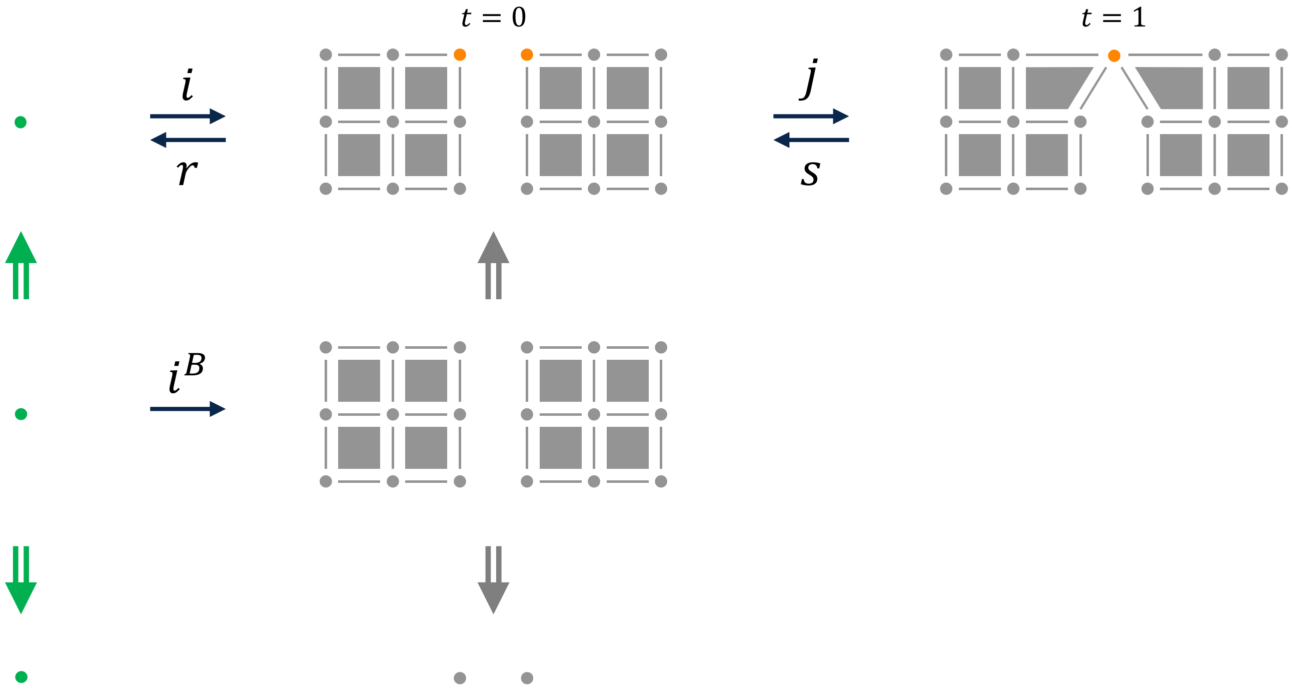
METHOD: WHEN MERGING CELLS



- ❑ Operation
- ❑ Initialization

4

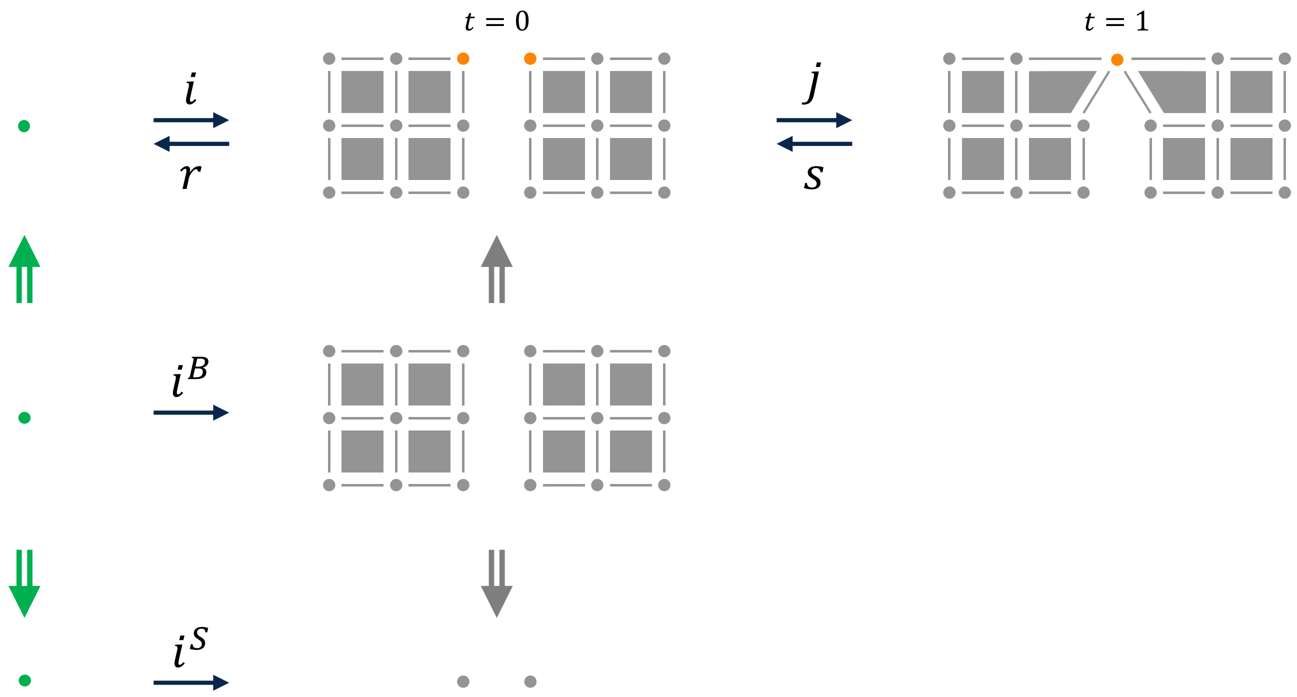
METHOD: WHEN MERGING CELLS



❑ Operation
❑ Initialization

4

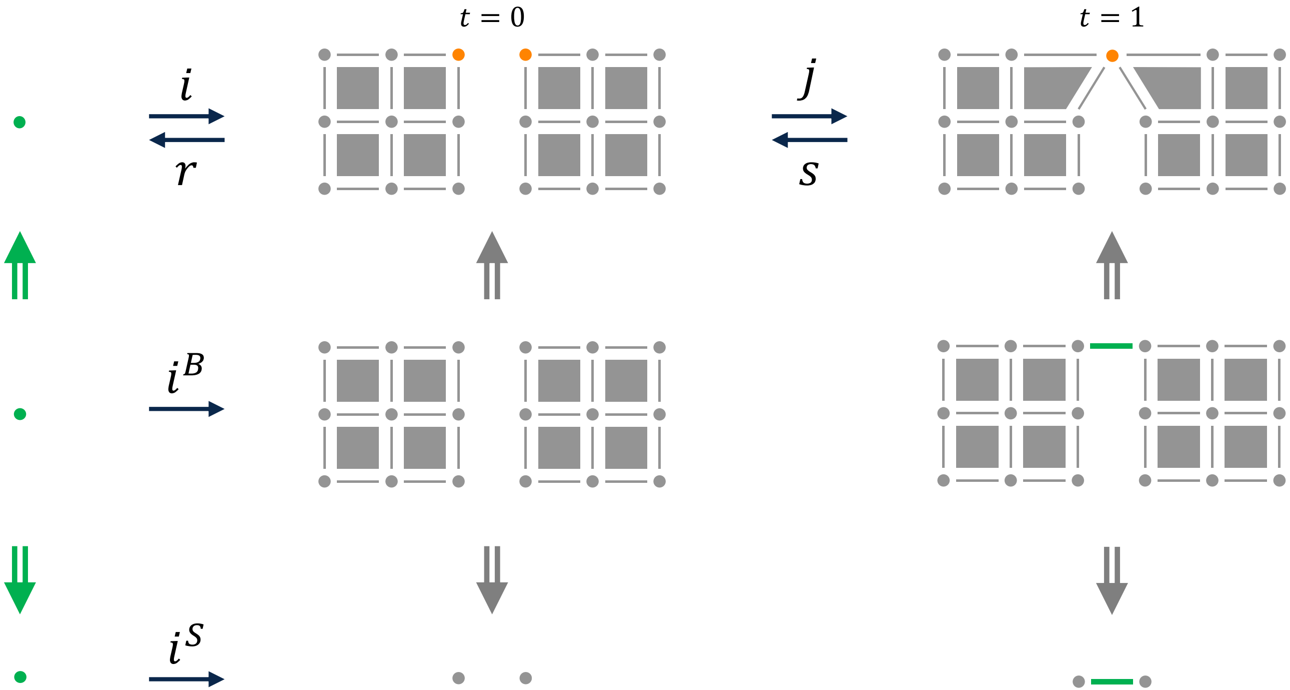
METHOD: WHEN MERGING CELLS



- ❑ Operation
- ❑ Initialization

4

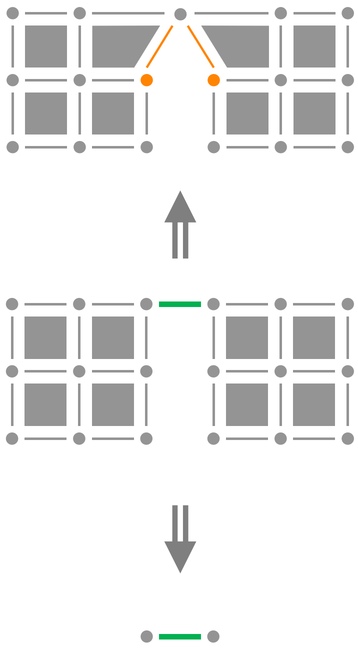
METHOD: WHEN MERGING CELLS



❑ Operation
❑ Initialization

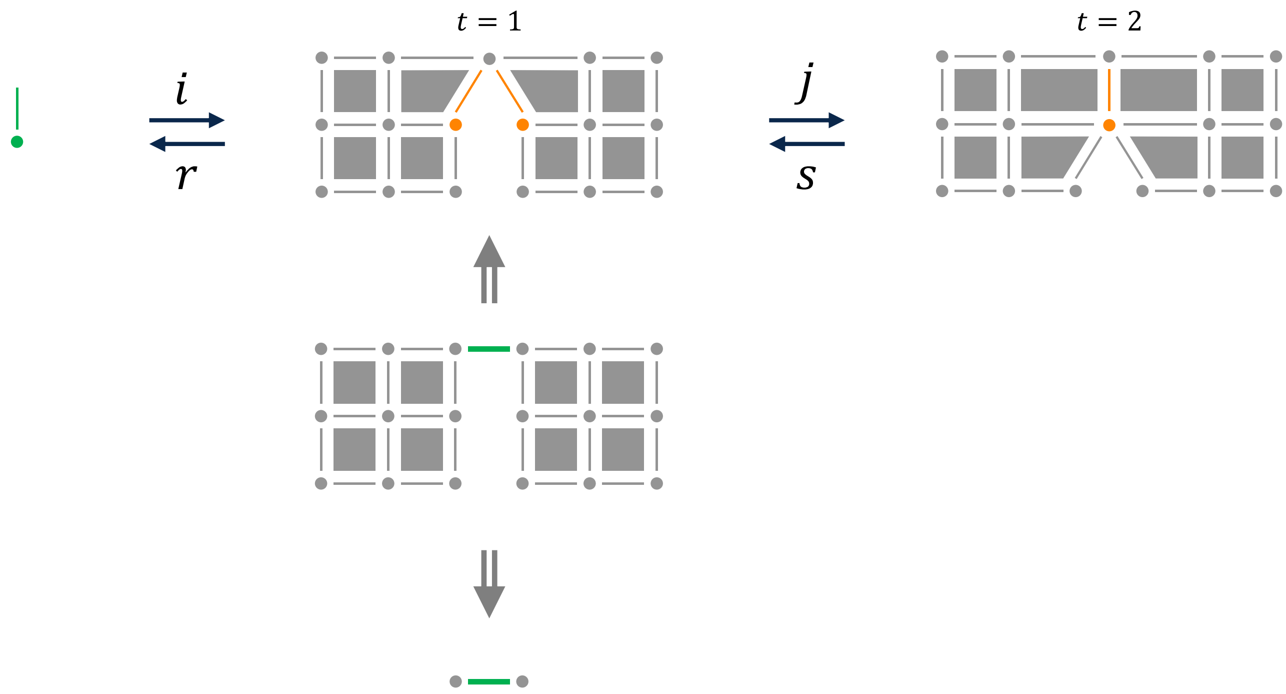
METHOD: WHEN MERGING CELLS

$t = 1$



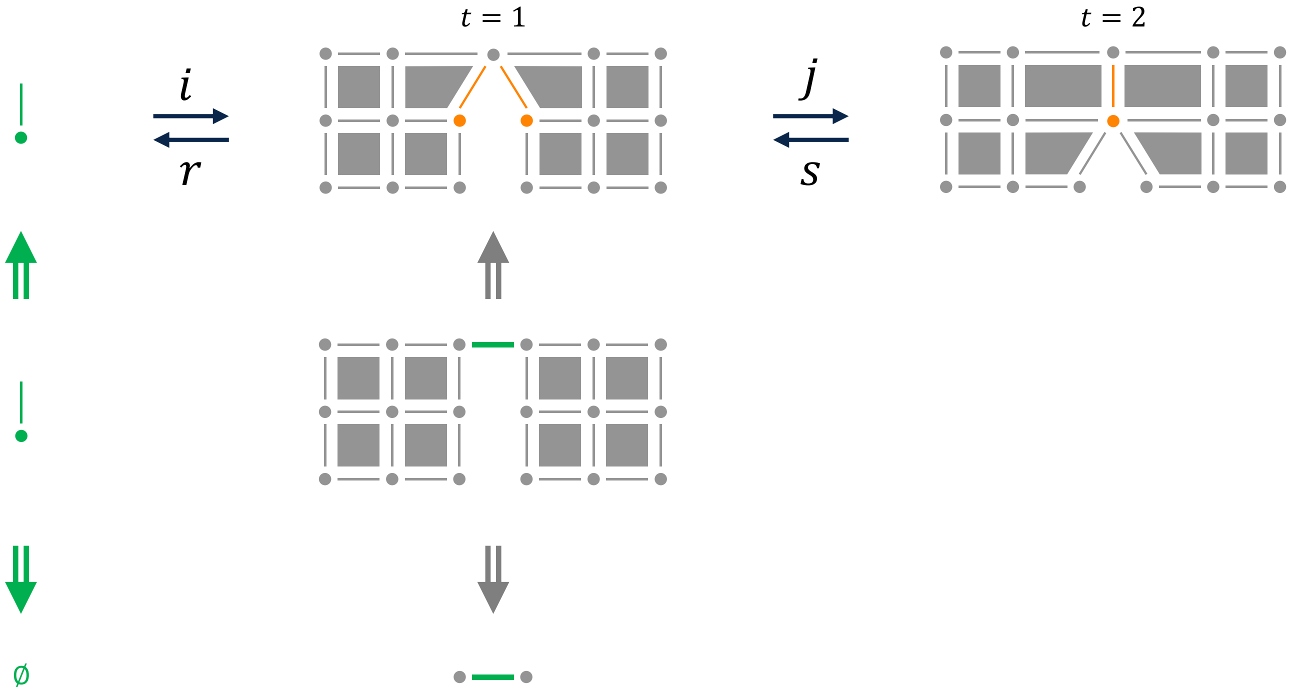
- ❑ Operation
- ❑ Added complexity
- ❑ Initialization

METHOD: WHEN MERGING CELLS



- Operation
- Added complexity
- Initialization

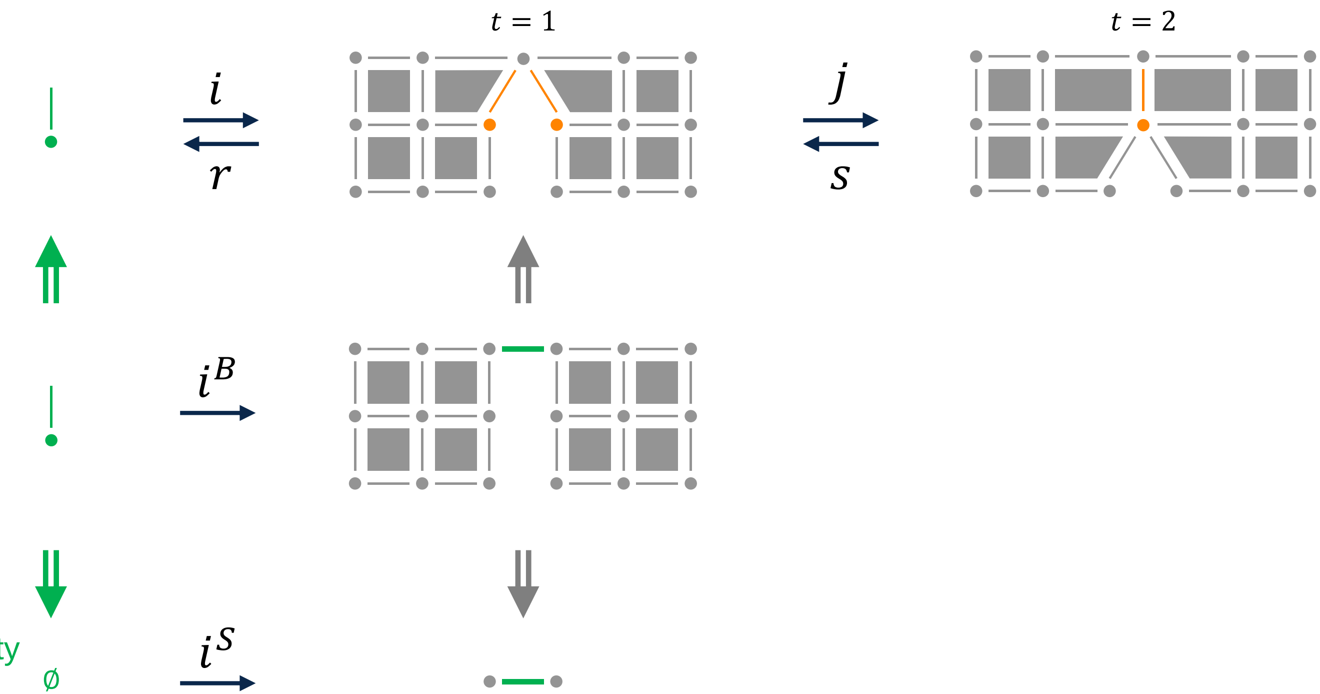
METHOD: WHEN MERGING CELLS



- ❑ Operation
- ❑ Added complexity
- ❑ Initialization

4

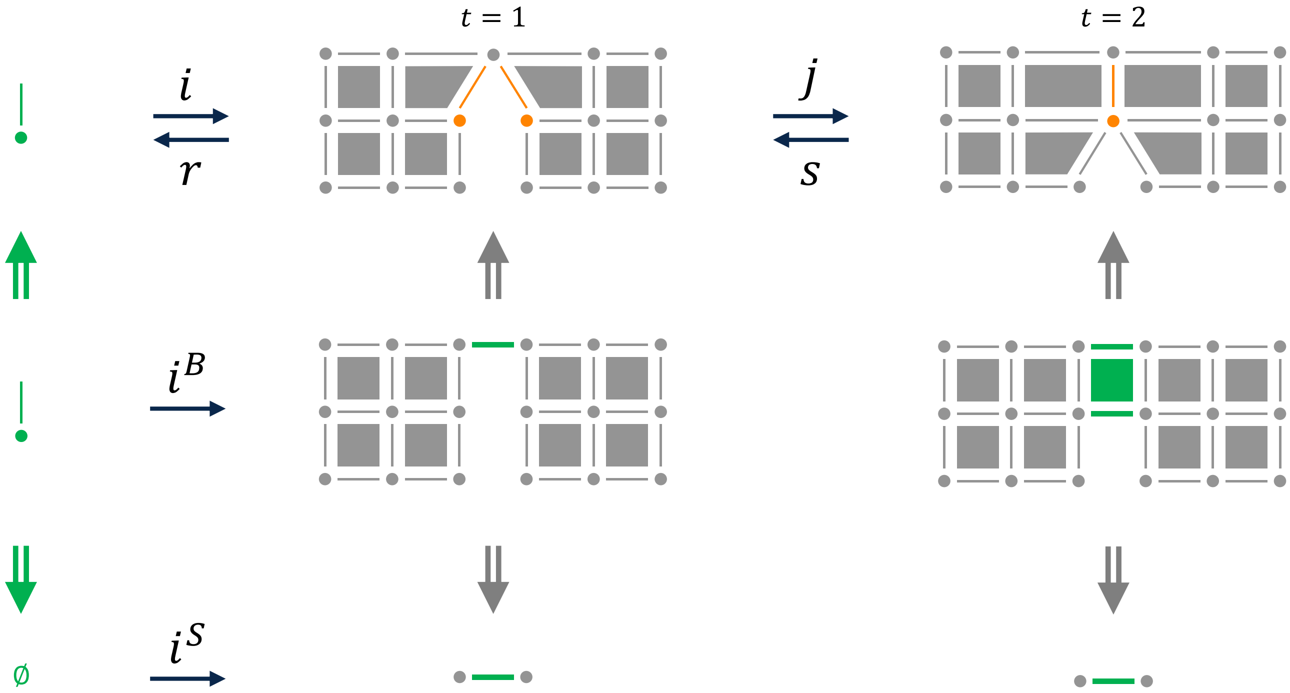
METHOD: WHEN MERGING CELLS



❑ Operation
❑ Added complexity
❑ Initialization

4

METHOD: WHEN MERGING CELLS

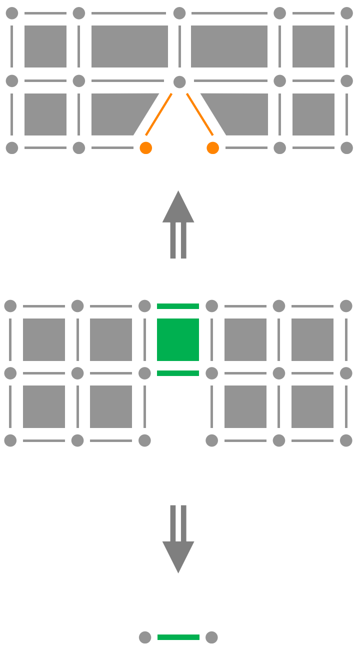


- ❑ Operation
- ❑ Added complexity
- ❑ Initialization

4

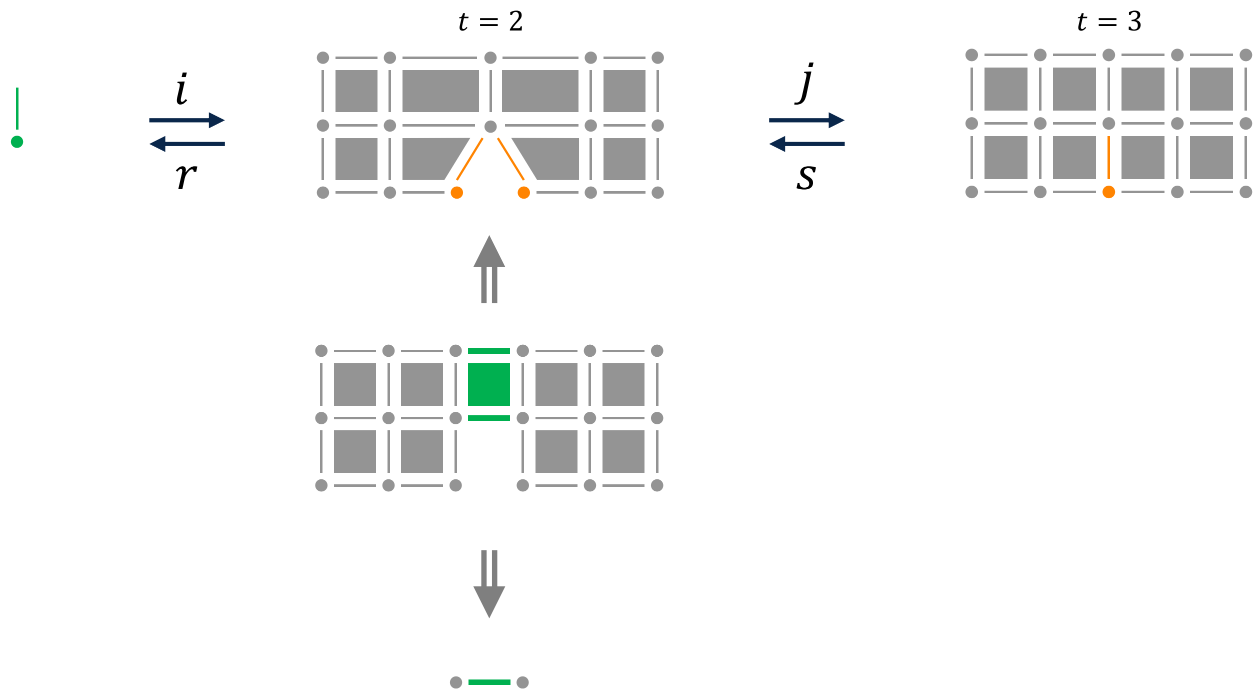
METHOD: WHEN MERGING CELLS

$t = 2$



❑ Operation
❑ Added complexity
❑ Initialization

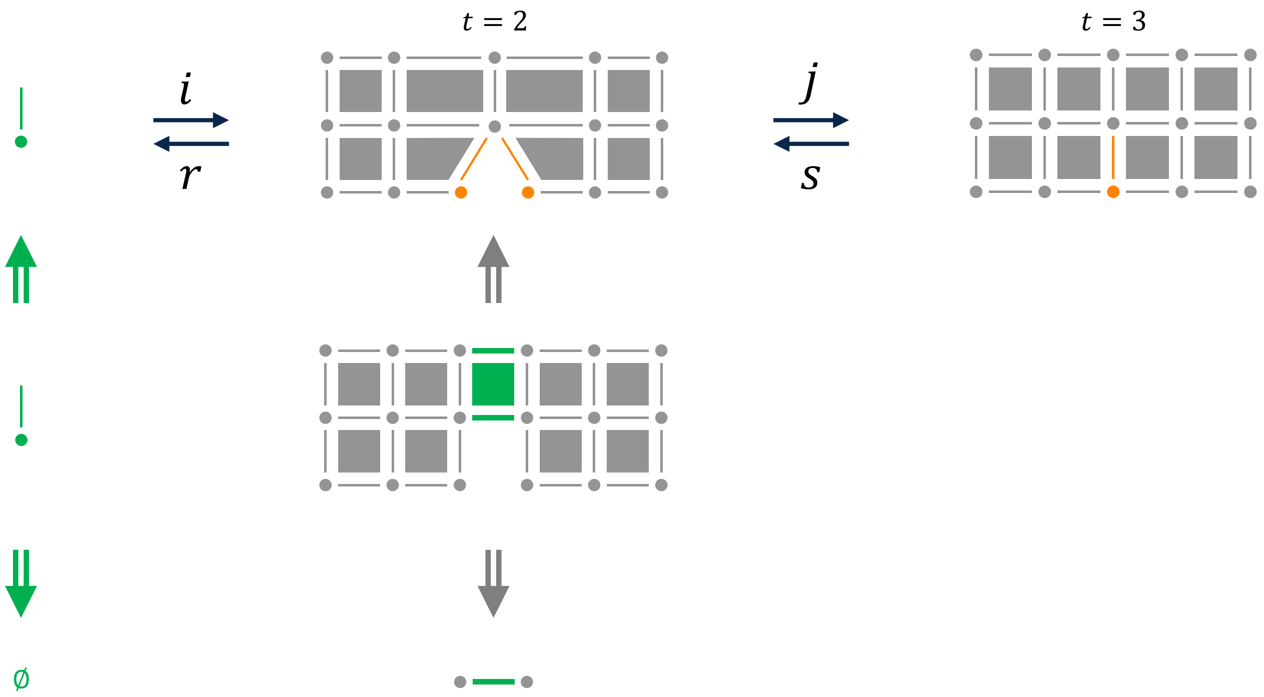
METHOD: WHEN MERGING CELLS



- ❑ Operation
- ❑ Added complexity
- ❑ Initialization

4

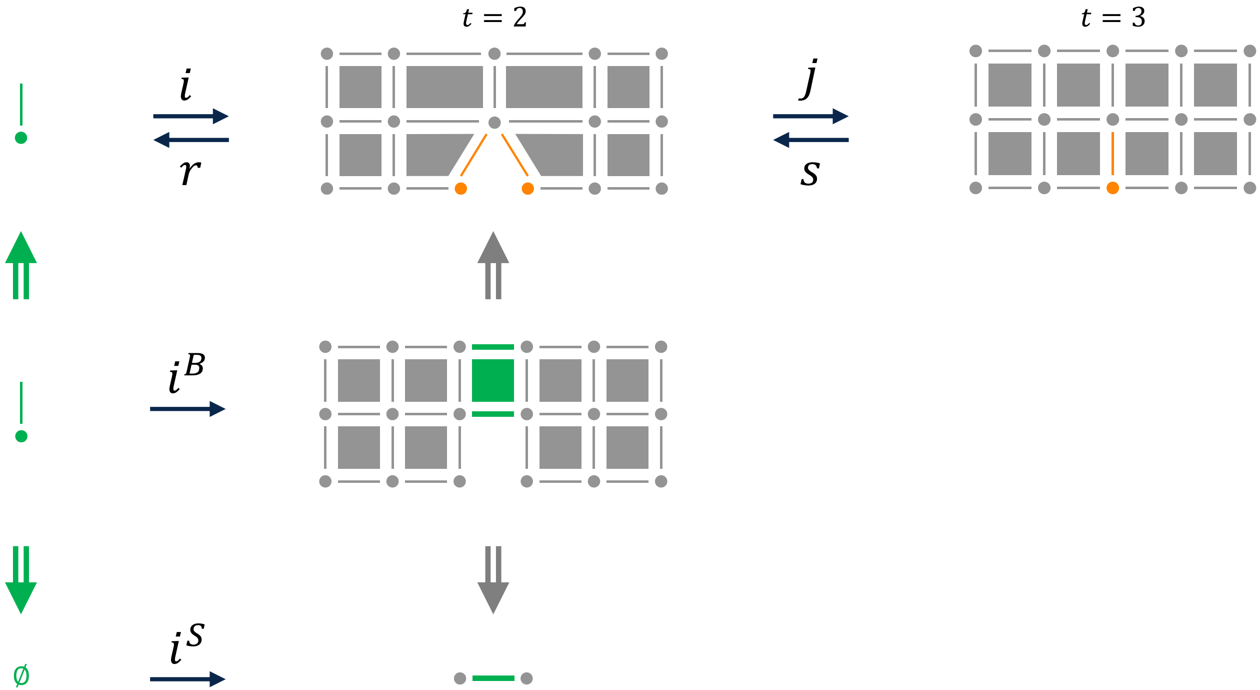
METHOD: WHEN MERGING CELLS

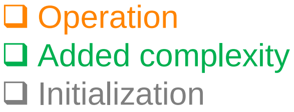


- ❑ Operation
- ❑ Added complexity
- ❑ Initialization

4

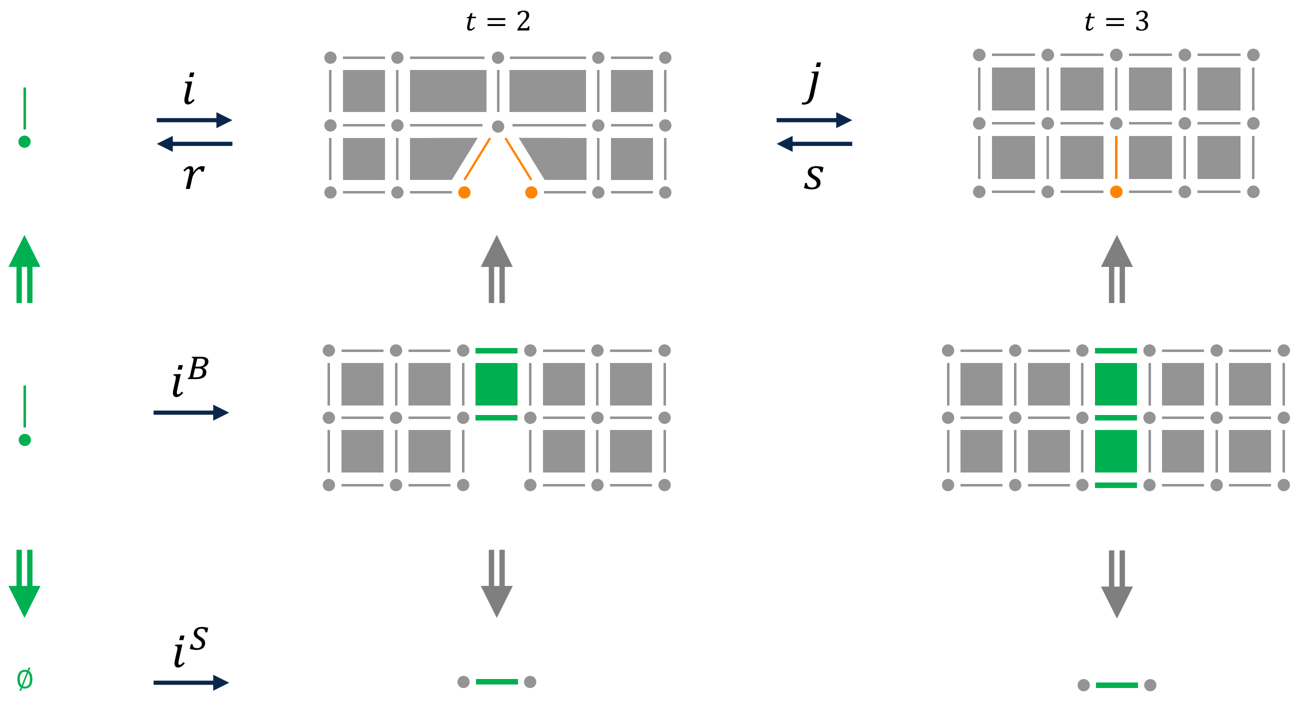
METHOD: WHEN MERGING CELLS





4

METHOD: WHEN MERGING CELLS



❑ Operation
❑ Added complexity
❑ Initialization

5: ANALYSIS

5

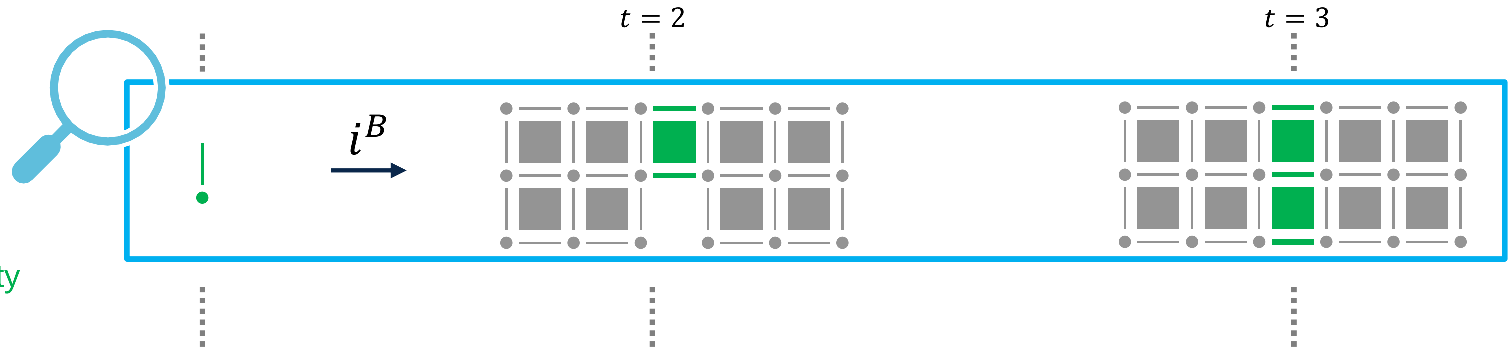
ANALYSIS

Is the computation complexity only related to the complexity of the operated part?

5

FOCUS ON INTERMEDIATE OBJECTS

At step t :



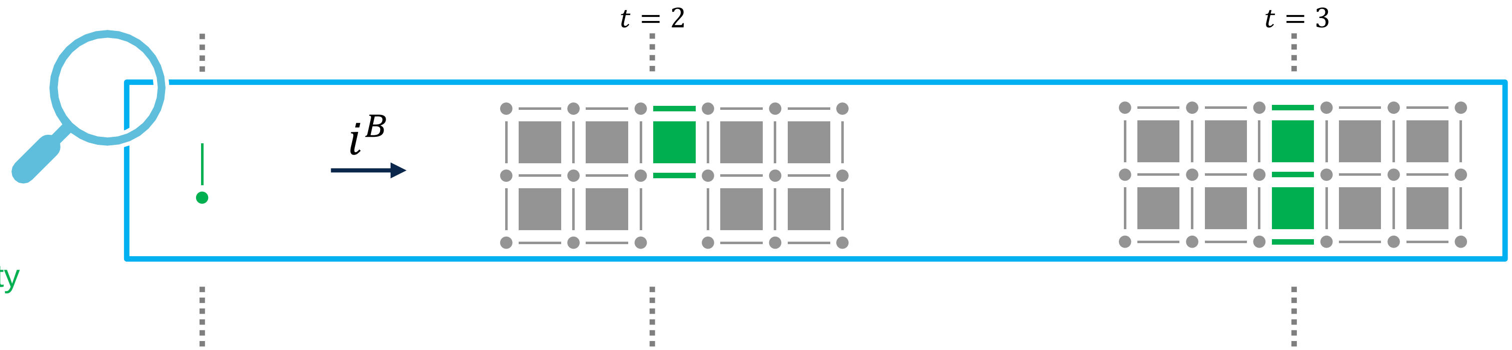
- ❑ Added complexity
- ❑ Initialization

5

FOCUS ON INTERMEDIATE OBJECTS

At step t :

- Space: object at step t = initial object + $\sum_{n=1}^{t}$ operations representations



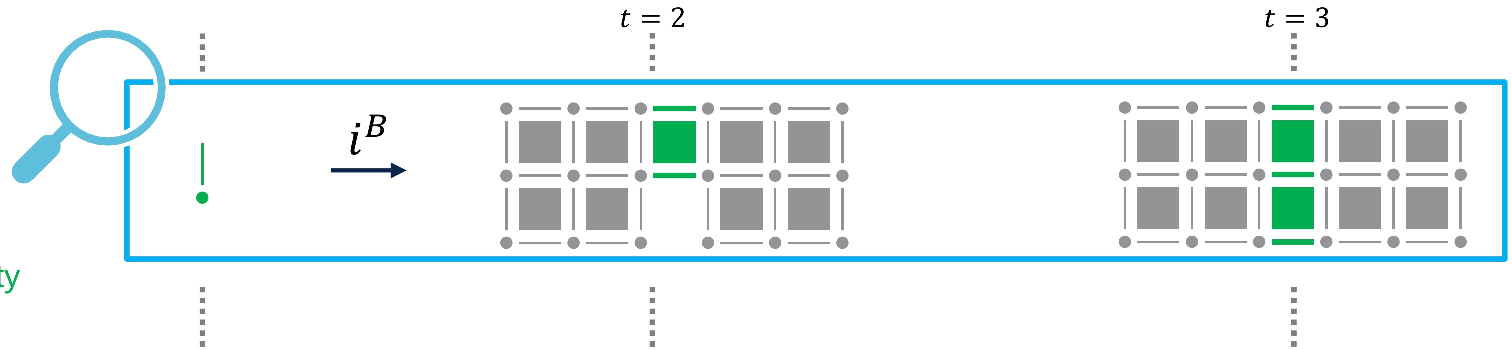
❑ Added complexity
❑ Initialization

5

FOCUS ON INTERMEDIATE OBJECTS

At step t :

- Space: object at step t = initial object + $\sum_{n=1}^{t}$ operations representations
- Time: depends only on current step operation : $\partial_{t+1} = \begin{pmatrix} \partial_{op.rep.} & i^B \\ 0 & \partial_t \end{pmatrix}$



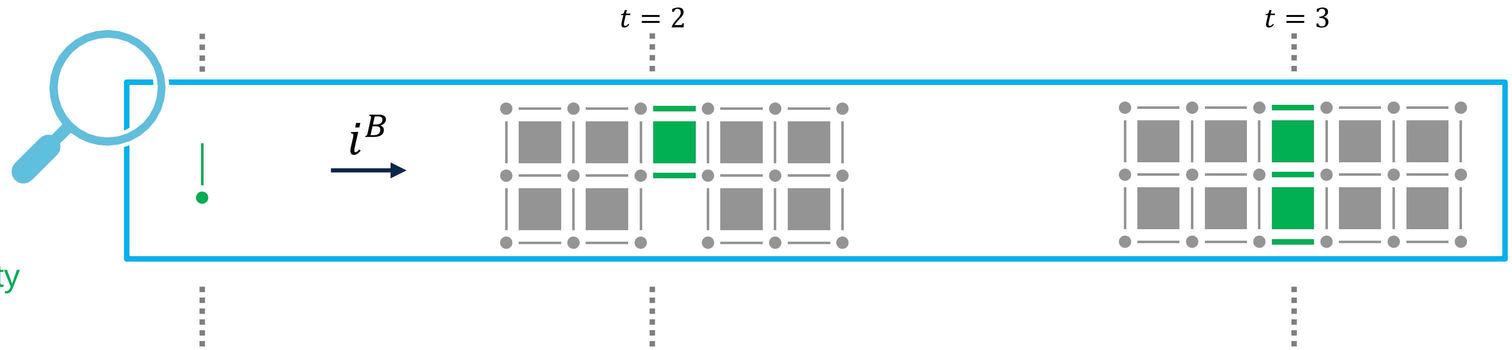
- ❑ Added complexity
- ❑ Initialization

5

FOCUS ON INTERMEDIATE OBJECTS

At step t :

- Space: object at step t = initial object + $\sum_{n=1}^{t}$ operations representations
- Time: depends only on current step operation : $\partial_{t+1} = \begin{pmatrix} \partial_{op.rep.} & i^B \\ 0 & \partial_t \end{pmatrix}$
- Possible loss of locality browsing ∂_t to construct ∂_{t+1} : => **Construct $\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}$ with a complexity depending only on A,B,C**

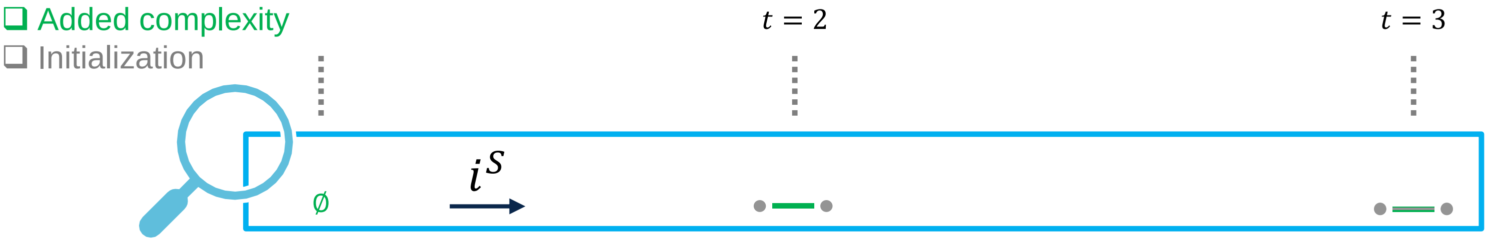


❑ Added complexity
❑ Initialization

5

FOCUS ON REDUCED OBJECTS

EXACTLY THE SAME BEHAVIOR than intermediate objects.



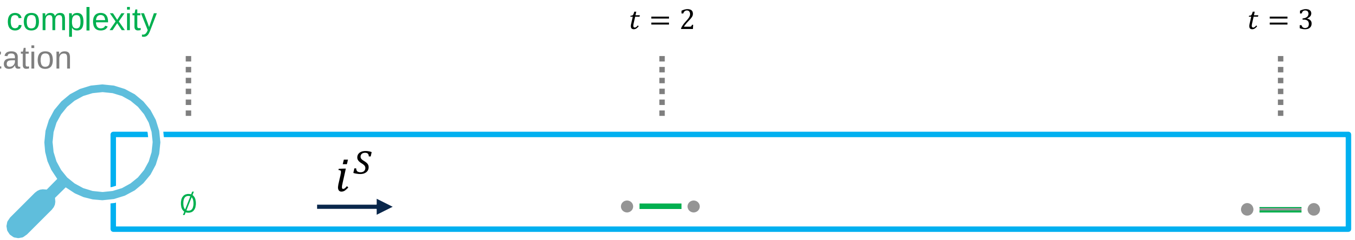
5

FOCUS ON REDUCED OBJECTS

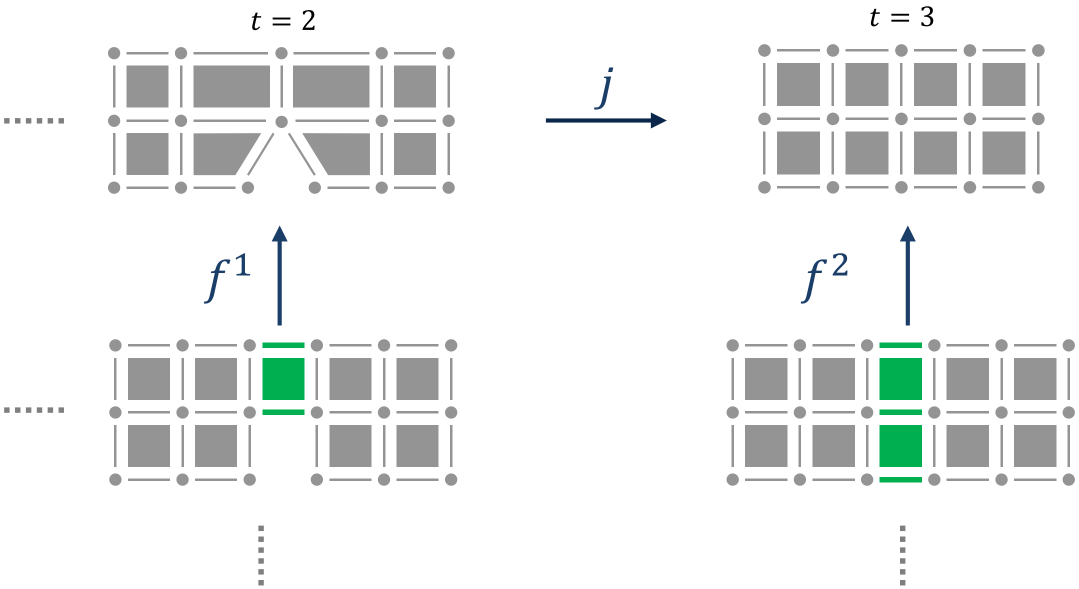
EXACTLY THE SAME BEHAVIOR than intermediate objects. At step t :

- Space: object at step t = initial object + $\sum_{n=1}^{t}$ operations representations
- Time: depends only on current step operation : $\partial_{t+1} = \begin{pmatrix} \partial_{op.rep.} & i^S \\ 0 & \partial_t \end{pmatrix}$
- Possible loss of locality browsing ∂_t to construct ∂_{t+1} : => Construct $\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}$ with a complexity depending only on A,B,C

❑ Added complexity
❑ Initialization



5 FOCUS ON f COMPLEXITY

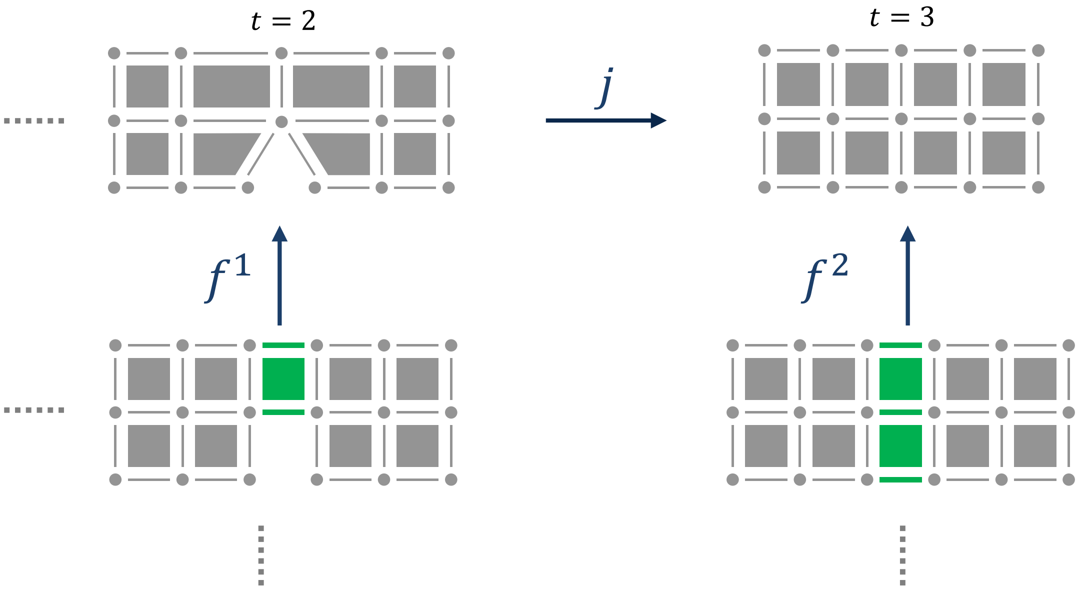


- Added complexity
- Initialization complexity

5

FOCUS ON f COMPLEXITY

$$f^2 = \binom{0}{f^1 j}$$

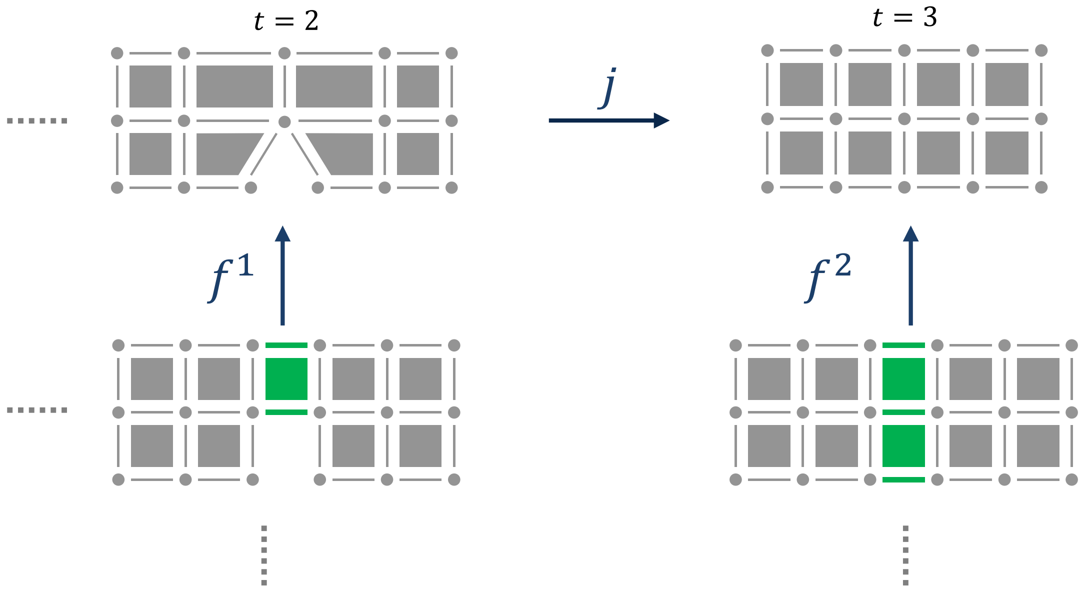


- ❑ Added complexity
- ❑ Initialization complexity

5 FOCUS ON f COMPLEXITY

$$f^2 = \begin{pmatrix} 0 \\ f^1 j \end{pmatrix}$$

Possible loss of locality: $f^1 j$



- ❑ Added complexity
- ❑ Initialization complexity

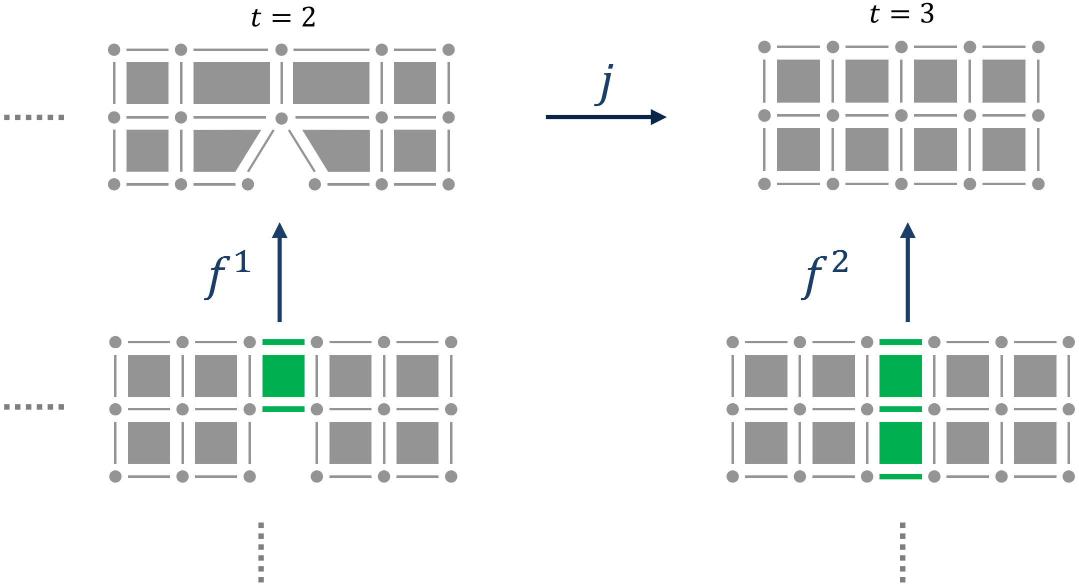
5

FOCUS ON f COMPLEXITY

$$f^2 = \binom{0}{f^1 j}$$

Possible loss of locality: $f^1 j$

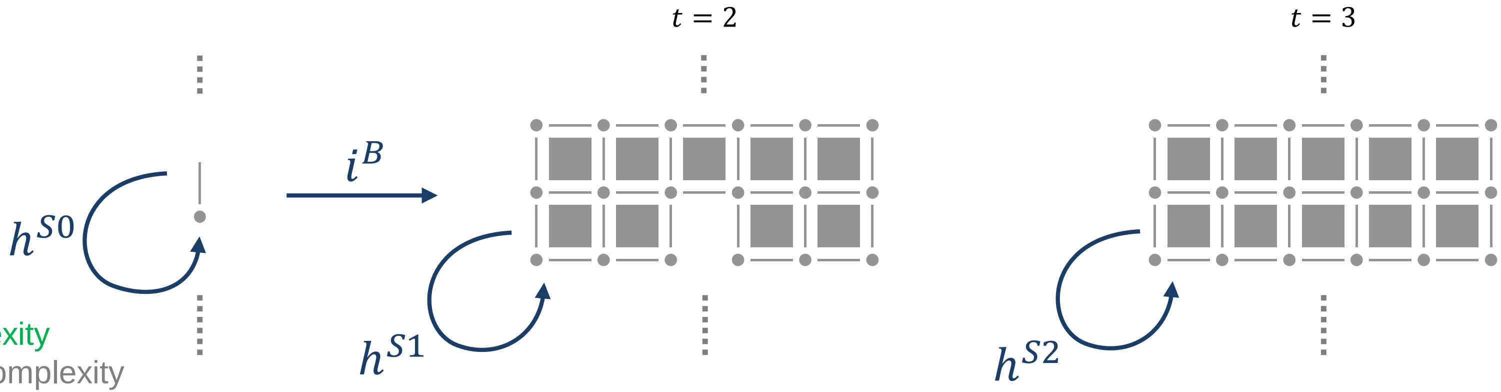
- Identity + Variations



❑ Added complexity
❑ Initialization complexity

5

FOCUS ON h^S COMPLEXITY

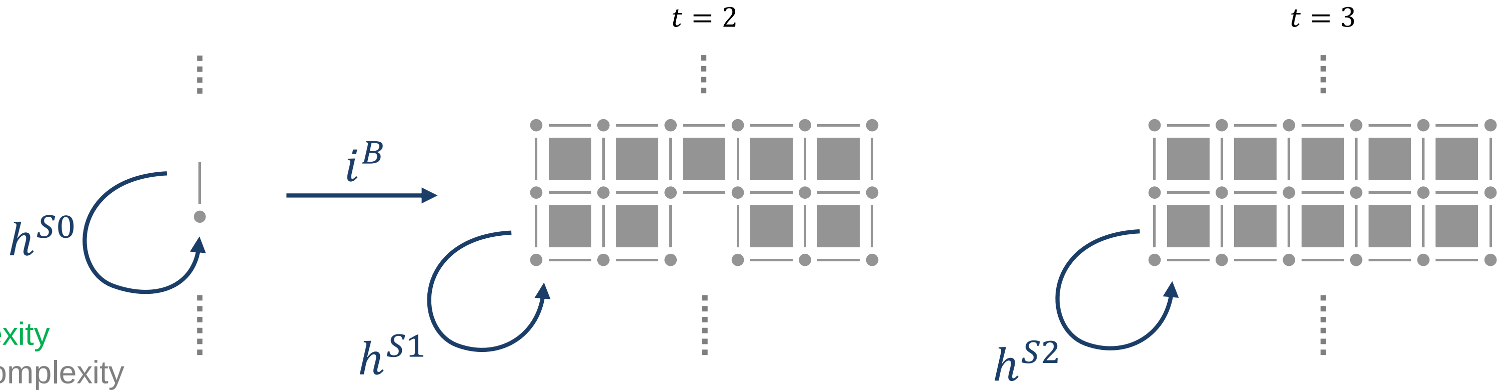


- ❑ Added complexity
- ❑ Initialization complexity

5

FOCUS ON h^S COMPLEXITY

$$h^{S2} = \begin{pmatrix} -h^{S0} & h^{S0} i^B h^{S1} \\ 0 & h^{S1} \end{pmatrix}$$



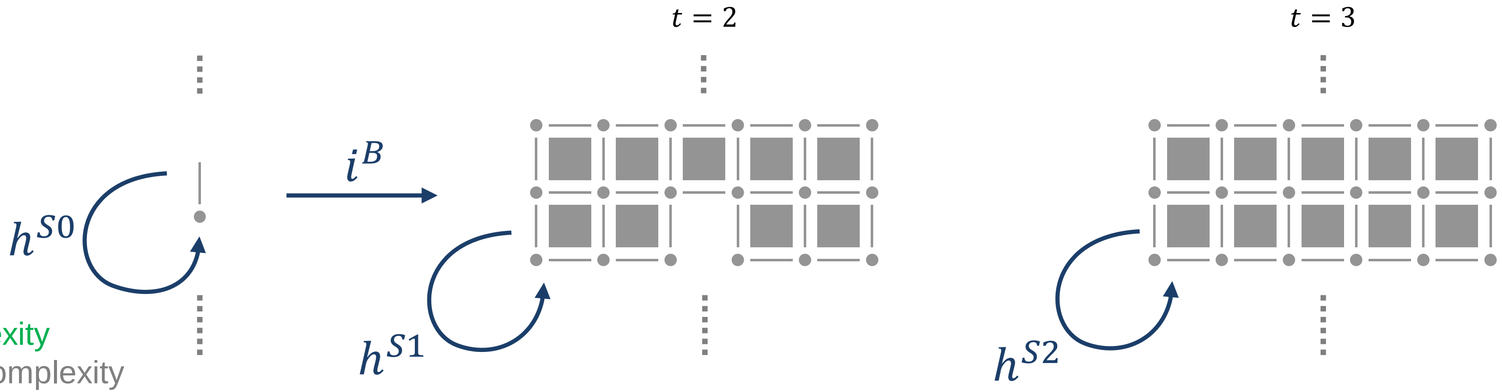
- ❑ Added complexity
- ❑ Initialization complexity

5

FOCUS ON h^S COMPLEXITY

Possible loss of locality: browsing h^{S1} for h^{S2} construction

$$h^{S2} = \begin{pmatrix} -h^{S0} & h^{S0} i^B h^{S1} \\ 0 & h^{S1} \end{pmatrix}$$



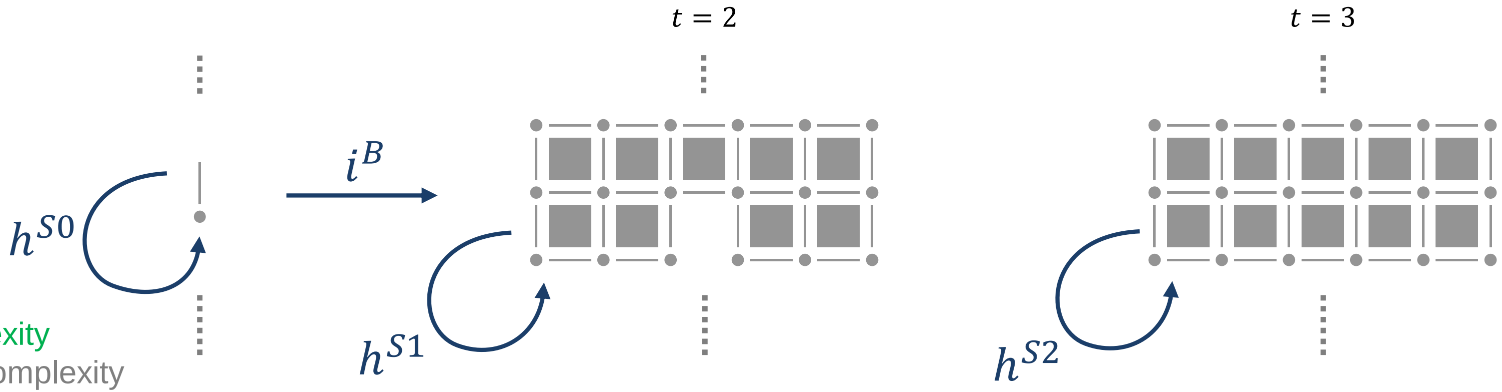
- ❑ Added complexity
- ❑ Initialization complexity

5

FOCUS ON h^S COMPLEXITY

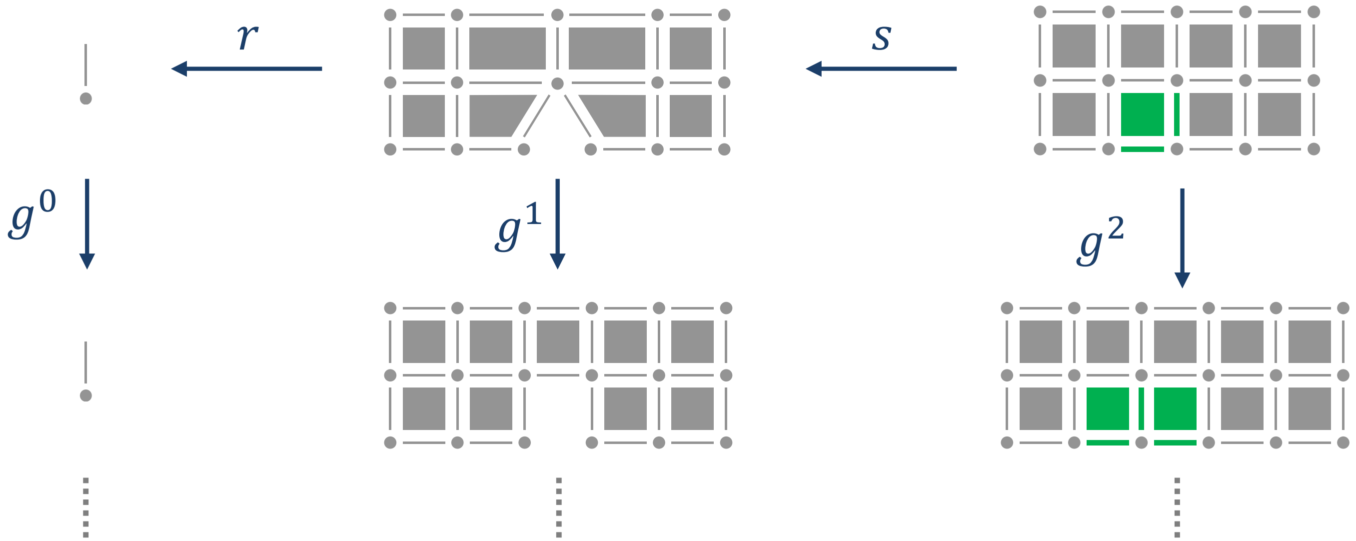
Possible loss of locality: browsing h^{S1} for h^{S2} construction

$h^{S2} = \begin{pmatrix} -h^{S0} & h^{S0} i^B h^{S1} \\ 0 & h^{S1} \end{pmatrix}$ => Construct $\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}$ with a complexity depending only on A,B,C



- ❑ Added complexity
- ❑ Initialization complexity

5 FOCUS ON g COMPLEXITY

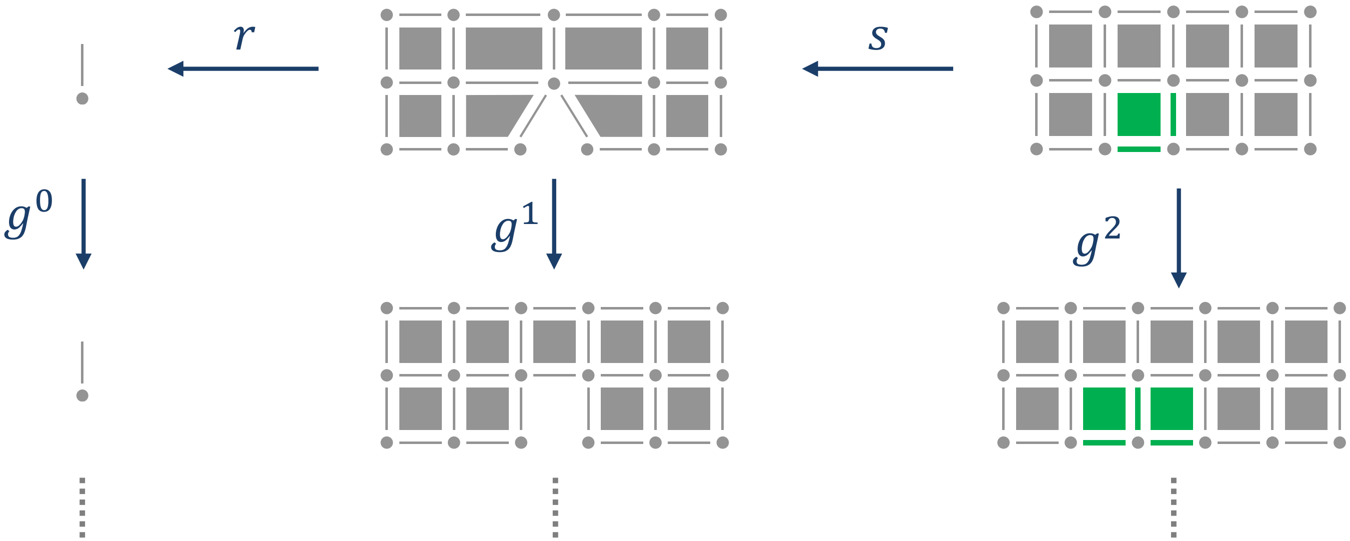


❑ Added complexity
❑ Initialization complexity

5

FOCUS ON g COMPLEXITY

$$g^2 = (-s\partial^1 r g^0 \quad s g^1)$$



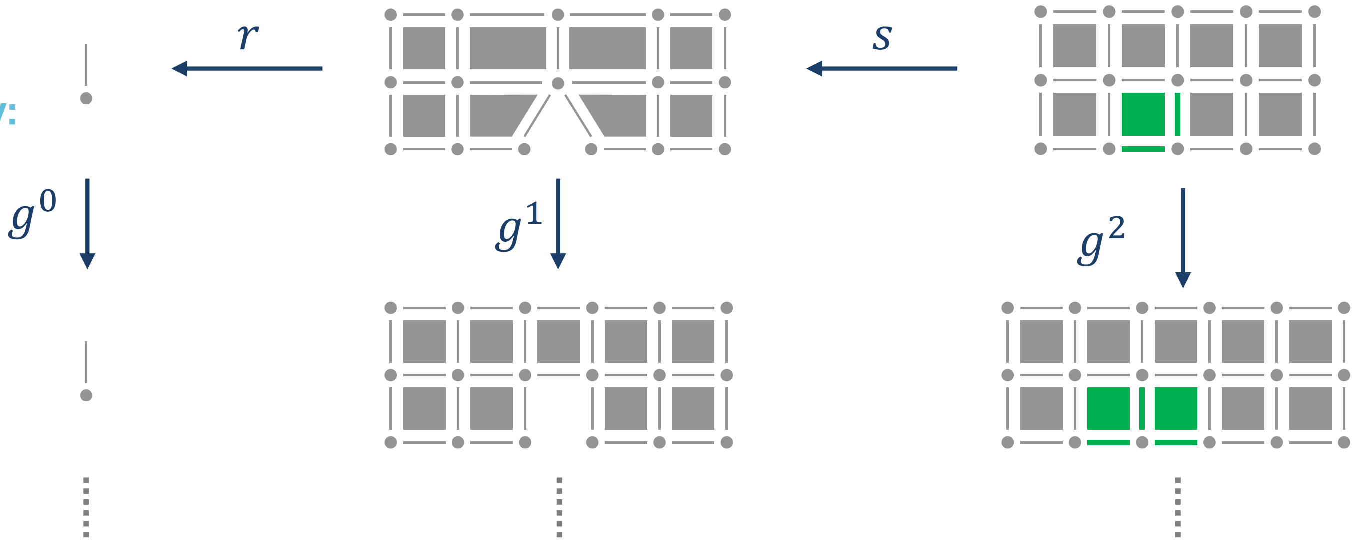
- ❑ Added complexity
- ❑ Initialization complexity

5

FOCUS ON g COMPLEXITY

$$g^2 = (-s\partial^1 r g^0 \quad s g^1)$$

Possible loss of locality:
sg^1



- ❑ Added complexity
- ❑ Initialization complexity

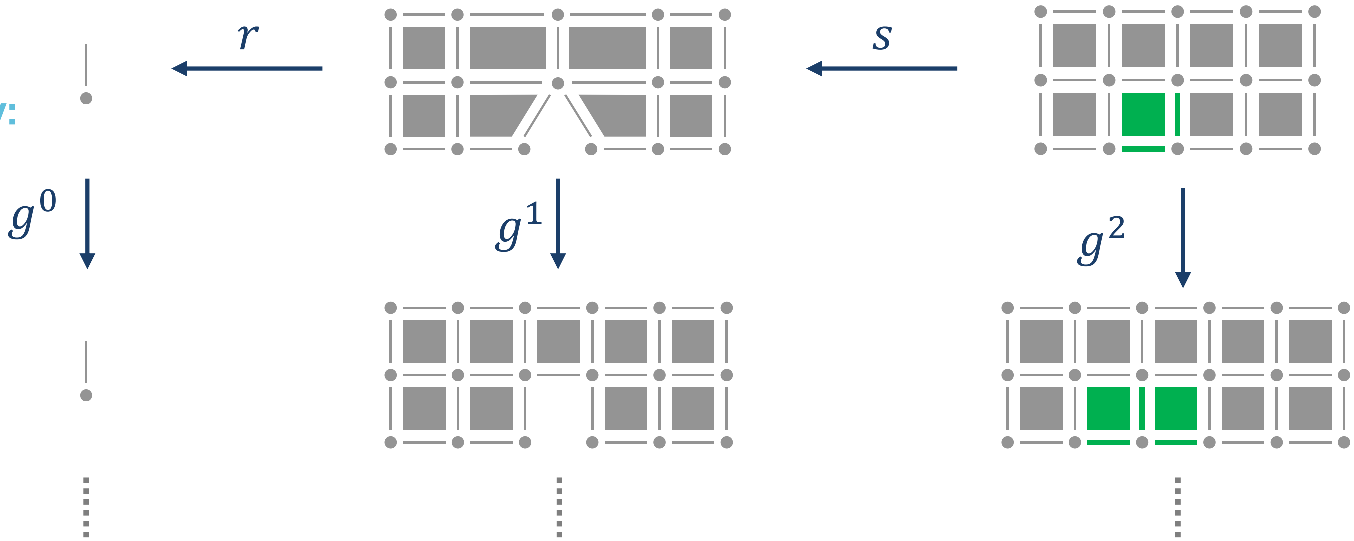
5

FOCUS ON g COMPLEXITY

$g^2 = (-s\partial^1 r g^0 \quad s g^1)$

Possible loss of locality:
sg^1

- Identity + Variations



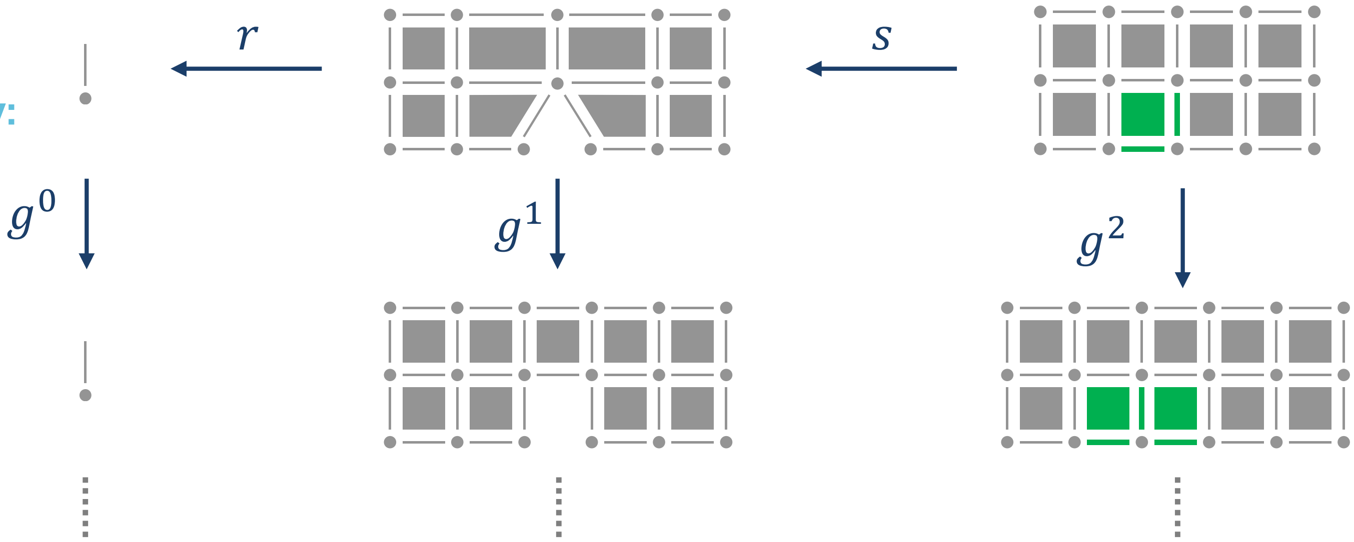
❑ Added complexity
❑ Initialization complexity

5 FOCUS ON g COMPLEXITY

$$g^2 = \begin{pmatrix} -s\partial^1 r g^0 & s g^1 \end{pmatrix}$$

Possible loss of locality: sg^1

- Identity + Variations
- $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ with a complexity depending only on A,B,C



- ❑ Added complexity
- ❑ Initialization complexity

5

REQUIREMENTS

Is the computation complexity only related to the complexity of the operated part?

5 REQUIREMENTS

Is the computation complexity only related to the complexity of the operated part?

Yes, with the following conditions :

5

REQUIREMENTS

Is the computation complexity only related to the complexity of the operated part?

Yes, with the following conditions :

- Being able to construct $\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}$ with an algorithm browsing A, B, C only

5 REQUIREMENTS

Is the computation complexity only related to the complexity of the operated part?

Yes, with the following conditions :

- Being able to construct $\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}$ with an algorithm browsing A, B, C only
- Having a matrix product algorithm using an implicit representation of identity

5

REQUIREMENTS

Is the computation complexity only related to the complexity of the operated part?

Yes, with the following conditions :

- Being able to construct $\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}$ with an algorithm browsing A, B, C only
- Having a matrix product algorithm using an implicit representation of identity
- Handling sparsity

5

TRIED SOLUTIONS

SOFTWARE	IMPLICIT IDENTITY
EIGEN	
INTELMKL	
NUMPY	
BOOST	

5 TRIED SOLUTIONS

SOFTWARE	IMPLICIT IDENTITY	$\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ BROWSING ONLY A, B, C
EIGEN		
INTELMKL		
NUMPY		
BOOST		

Essentially due to sparsity representation schemes

5 TRIED SOLUTIONS

Software	Implicit identity	$\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ browsing only A, B, C	Sparse matrices
Eigen			
IntelMKL			
Numpy			
Boost			

5

PROPOSED SOLUTION

Doubly linked matrices + (i, j, v)

$$\begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 4 & 0 & 3 & 4 \end{pmatrix}$$

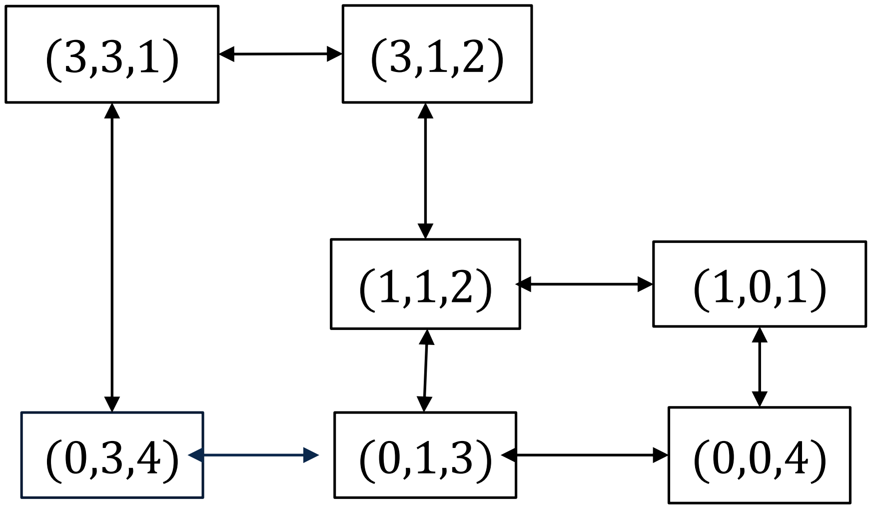
5

PROPOSED SOLUTION

Doubly linked matrices + (i, j, v)

$$\begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 4 & 0 & 3 & 4 \end{pmatrix}$$

i (↑), j (←)

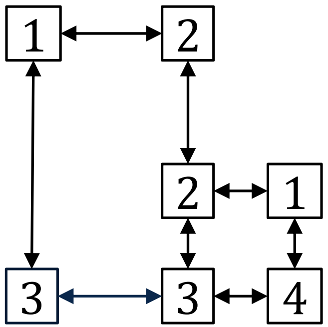


5

PROPOSED SOLUTION

Doubly linked matrices + (i, j, v)

$$\begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 4 & 0 & 3 & 4 \end{pmatrix}$$



5

PROPOSED SOLUTION

Doubly linked matrices : construct $\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}$ with an algorithm browsing A, B, C only

$$\begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 4 & 0 & 3 & 4 \end{pmatrix}$$

5

PROPOSED SOLUTION

Doubly linked matrices : construct $\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}$ with an algorithm browsing A, B, C only

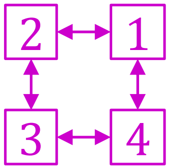
$$\begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 4 & 0 & 3 & 4 \end{pmatrix} \rightarrow \begin{matrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 \\ 4 & 0 \end{pmatrix} & \begin{pmatrix} 2 & 1 \\ 3 & 4 \end{pmatrix} \end{matrix}$$

5

PROPOSED SOLUTION

Doubly linked matrices : construct $\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}$ with an algorithm browsing A, B, C only

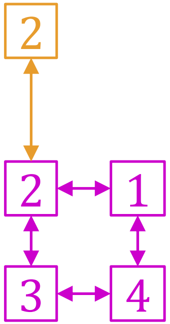
$$\begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 4 & 0 & 3 & 4 \end{pmatrix} \longrightarrow \begin{matrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 \\ 4 & 0 \end{pmatrix} & \begin{pmatrix} 2 & 1 \\ 3 & 4 \end{pmatrix} \end{matrix}$$



5 PROPOSED SOLUTION

Doubly linked matrices : construct $\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}$ with an algorithm browsing A, B, C only

$$\begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 4 & 0 & 3 & 4 \end{pmatrix} \longrightarrow \begin{matrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 \\ 4 & 0 \end{pmatrix} & \begin{pmatrix} 2 & 1 \\ 3 & 4 \end{pmatrix} \end{matrix}$$

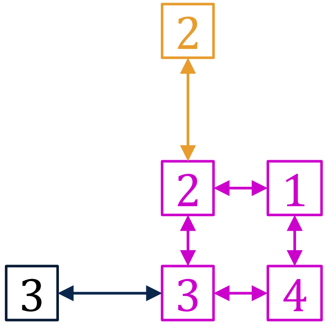


5

PROPOSED SOLUTION

Doubly linked matrices : construct $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ with an algorithm browsing A, B, C only

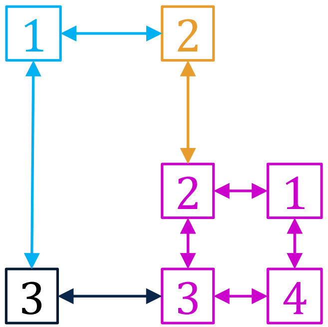
$$\begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 4 & 0 & 3 & 4 \end{pmatrix} \longrightarrow \begin{matrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 \\ 4 & 0 \end{pmatrix} & \begin{pmatrix} 2 & 1 \\ 3 & 4 \end{pmatrix} \end{matrix}$$



5 PROPOSED SOLUTION

Doubly linked matrices : construct $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ with an algorithm browsing A, B, C only

$$\begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 4 & 0 & 3 & 4 \end{pmatrix} \longrightarrow \begin{matrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 \\ 4 & 0 \end{pmatrix} & \begin{pmatrix} 2 & 1 \\ 3 & 4 \end{pmatrix} \end{matrix}$$



5

PROPOSED SOLUTION

Doubly linked matrices : construct $\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}$ with an algorithm browsing A, B, C only

Implicit representation of identity

$$\begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix}$$

5 PROPOSED SOLUTION

Doubly linked matrices : construct $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ with an algorithm browsing A, B, C only

Implicit representation of identity

$$\begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

5 PROPOSED SOLUTION

Doubly linked matrices : construct $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ with an algorithm browsing A, B, C only

Implicit representation of identity

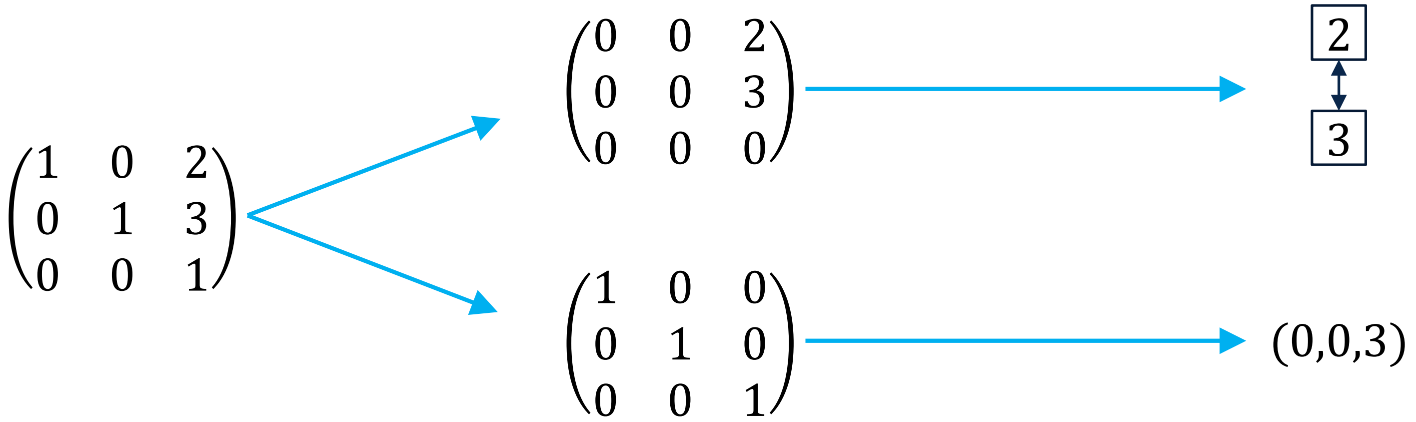
$$\begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \rightarrow (0,0,3)$$

5

PROPOSED SOLUTION

Doubly linked matrices : construct $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ with an algorithm browsing A, B, C only

Implicit representation of identity

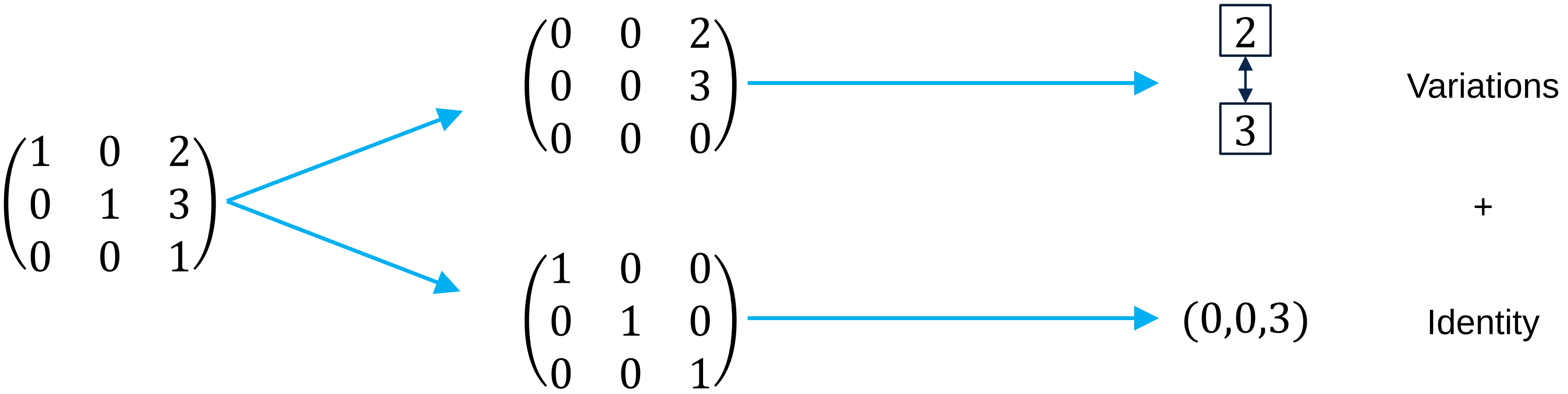


5

PROPOSED SOLUTION

Doubly linked matrices : construct $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ with an algorithm browsing A, B, C only

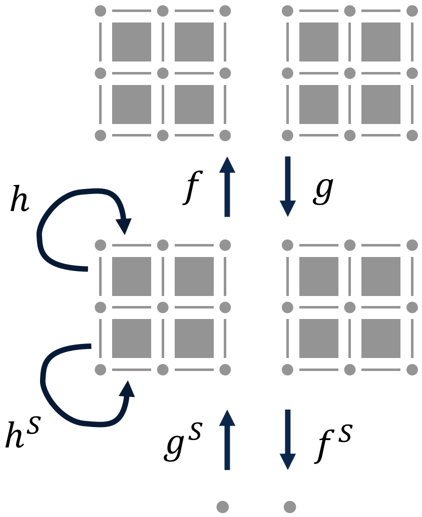
Implicit representation of identity



6: CONCLUSION AND PERSPECTIVES

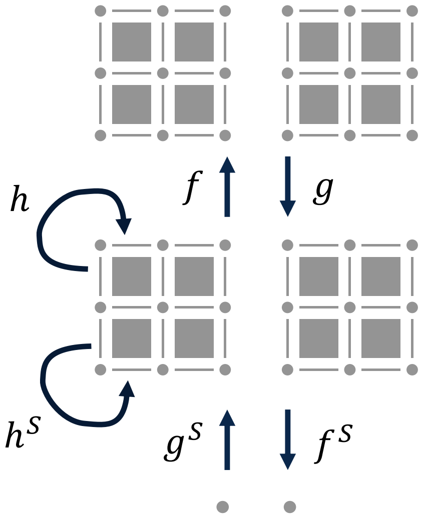
6

MORE EFFICIENT WITHOUT GENERATORS



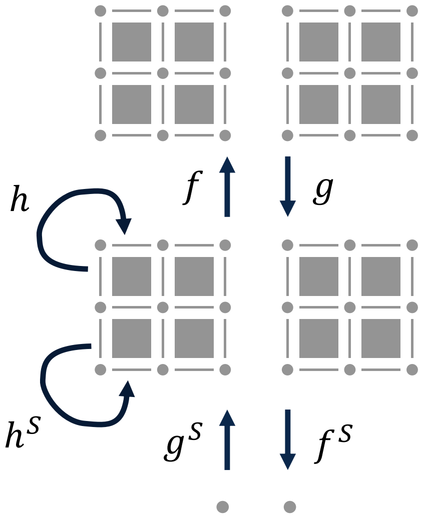
MORE EFFICIENT WITHOUT GENERATORS

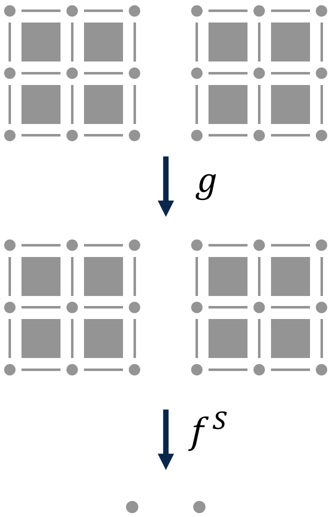
Some morphisms can be ignored



6 MORE EFFICIENT WITHOUT GENERATORS

Some morphisms can be ignored





6 CONCLUSION AND PERSPECTIVES

At each step: control of the object construction according to computed information

- Short exact sequence based method => tracking homology variations

6 CONCLUSION AND PERSPECTIVES

At each step: control of the object construction according to computed information

- Short exact sequence based method => tracking homology variations
- Theoretical and experimental study => complexity depending on the operated part

6

CONCLUSION AND PERSPECTIVES

At each step: control of the object construction according to computed information

- Short exact sequence based method => tracking homology variations
- Theoretical and experimental study => complexity depending on the operated part
- Cautions have to be taken regarding the implementation to preserve locality

6 CONCLUSION AND PERSPECTIVES

At each step: control of the object construction according to computed information

- Short exact sequence based method => tracking homology variations
- Theoretical and experimental study => complexity depending on the operated part
- Cautions have to be taken regarding the implementation to preserve locality

Perspectives

- Apply the method to the inverse operation (splitting cells)

6 CONCLUSION AND PERSPECTIVES

At each step: control of the object construction according to computed information

- Short exact sequence based method => tracking homology variations
- Theoretical and experimental study => complexity depending on the operated part
- Cautions have to be taken regarding the implementation to preserve locality

Perspectives

- Apply the method to the inverse operation (splitting cells)
- Apply in an application case: animation

6 CONCLUSION AND PERSPECTIVES

At each step: control of the object construction according to computed information

- Short exact sequence based method => tracking homology variations
- Theoretical and experimental study => complexity depending on the operated part
- Cautions have to be taken regarding the implementation to preserve locality

Perspectives

- Apply the method to the inverse operation (splitting cells)
- Apply in an application case: animation
- Parallelization

GARBAGE

7

ATTACHMENTS

Sparse matrices representation schemes

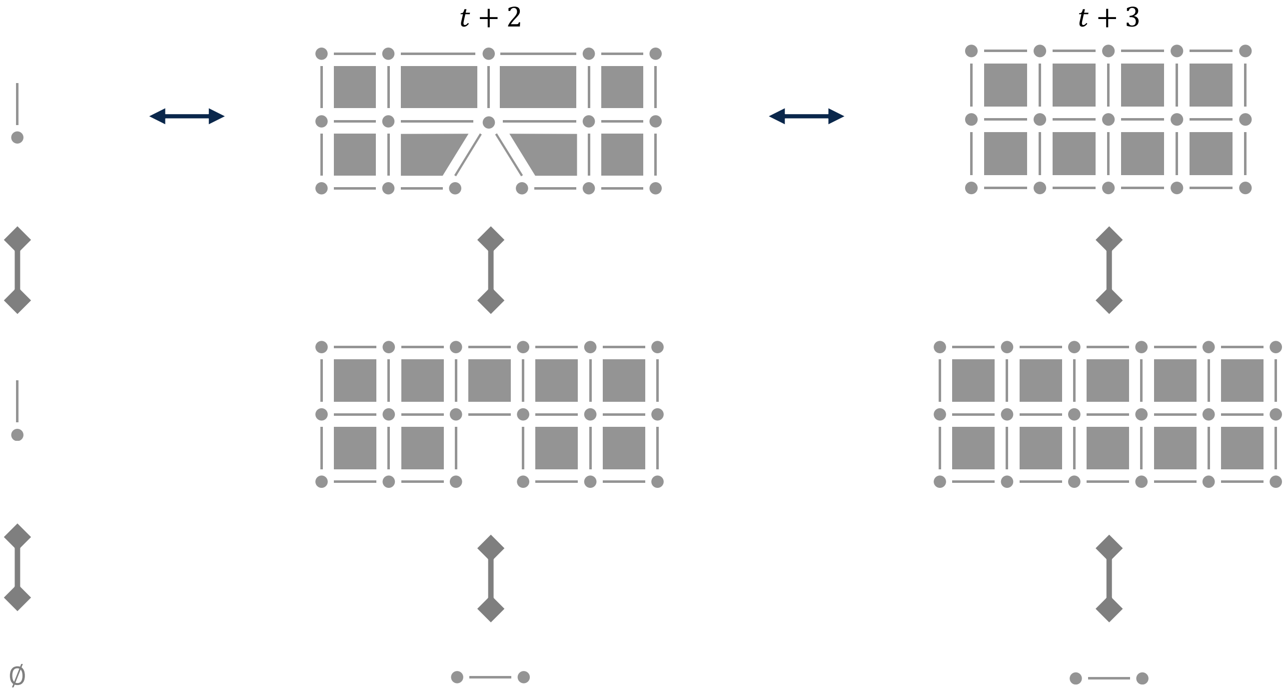
$$\begin{pmatrix} 0 & 4 & 1 \\ 2 & 0 & 5 \\ 0 & 3 & 0 \end{pmatrix}$$

CCS :

Values	2	4	3	1	5
Row Index	1	0	2	0	1
Column start index	0	1	3	6	

5

ANALYSIS : SPATIAL COMPLEXITY EVOLUTION ON COMPLEXES



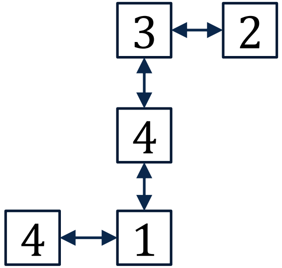
- Operation
- Added part
- Initialization

5 PROPOSED SOLUTION

Implicit identity

$$\begin{pmatrix} 1 & 3 & 2 \\ 0 & 5 & 0 \\ 4 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 3 & 2 \\ 0 & 4 & 0 \\ 4 & 1 & 0 \end{pmatrix}$$



$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \rightarrow (0,0,2)$$

- ❑ Identity block