Causal Graph Rewriting

Pablo Arrighi, Marin Costes, Luidnel Maignan, Gilles Dowek

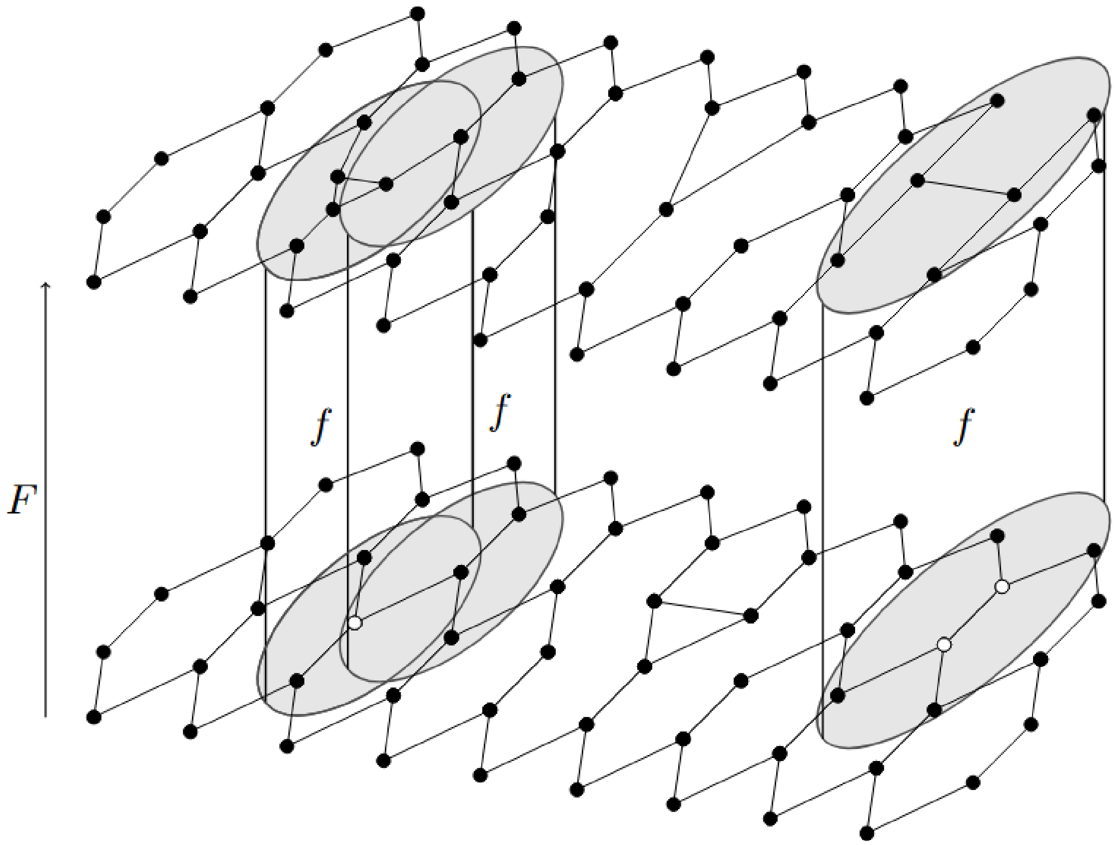
école normale supérieure paris–saclay

université PARIS-SACLAY

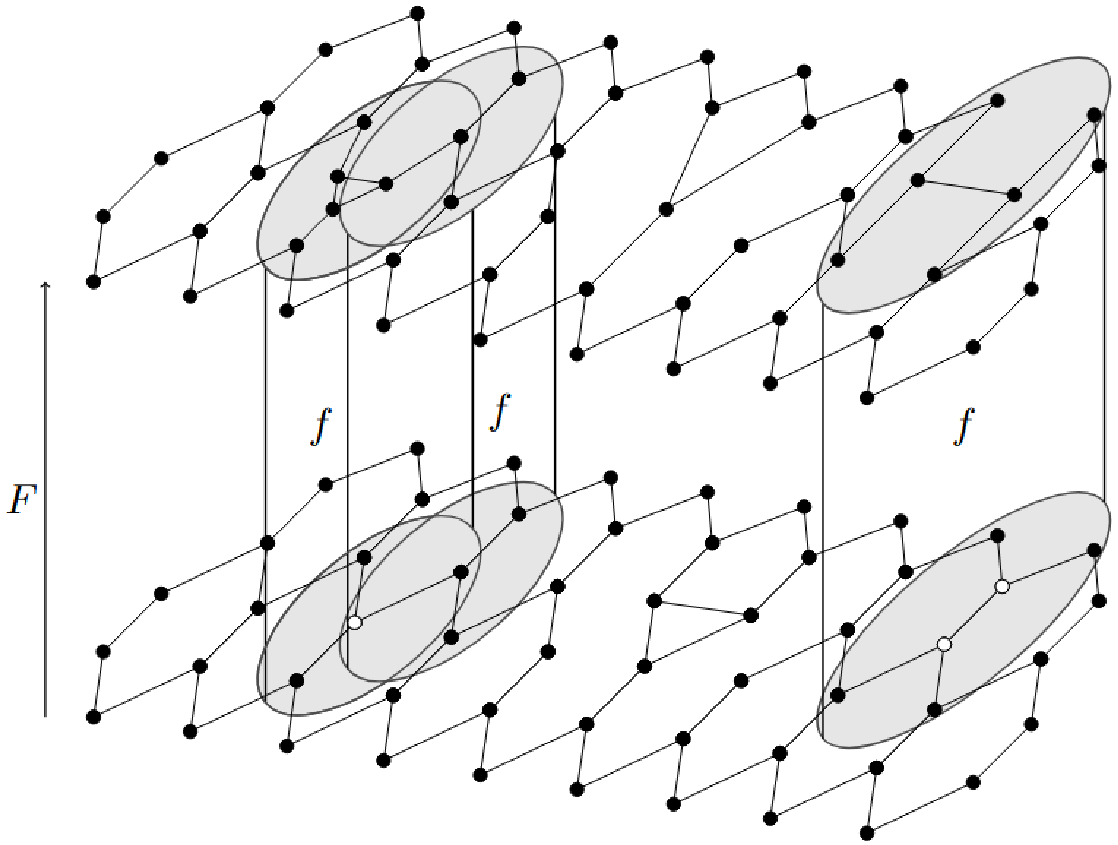
LMF Laboratoire Méthodes Formelles

QuaCS

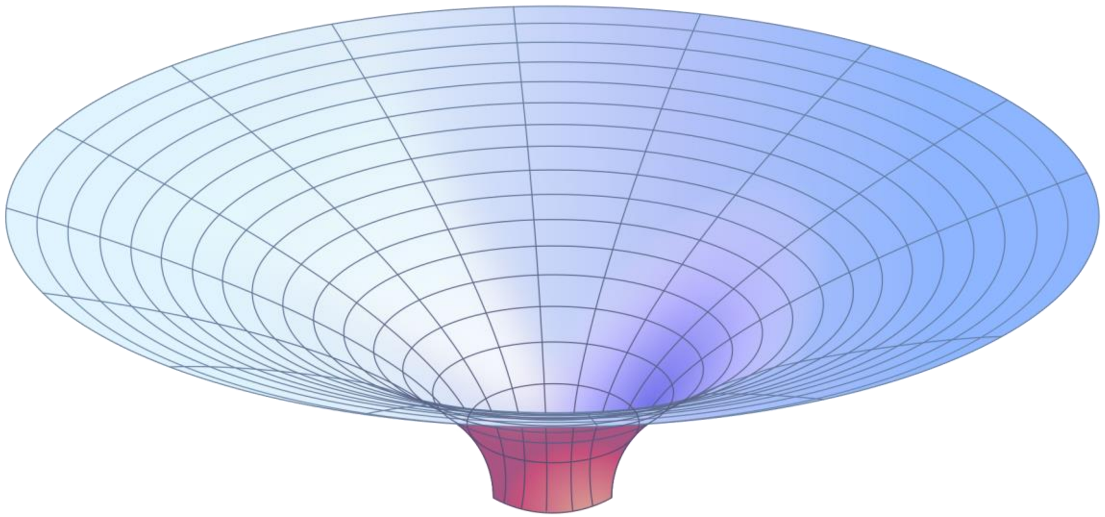
Introduction



Introduction



≠

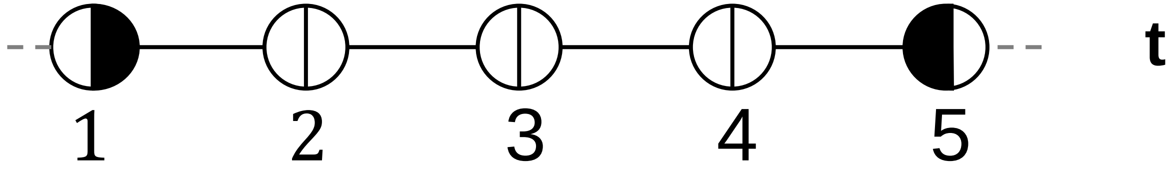


Dynamical systems and rewriting

Dynamical system (Synchronous)

Dynamical systems and rewriting

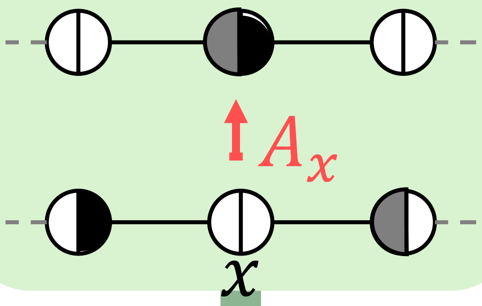
Dynamical system
(Synchronous)

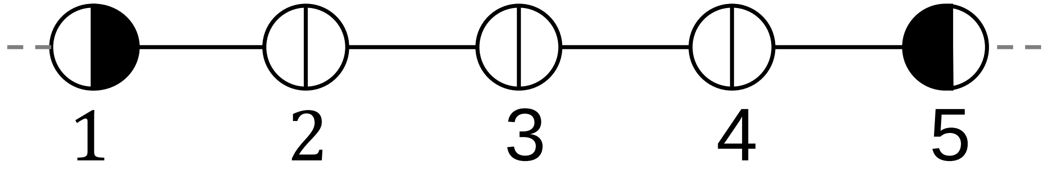


Dynamical systems and rewriting

Dynamical system
(Synchronous)

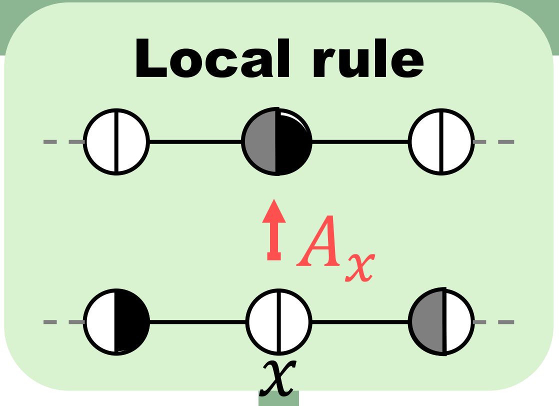
Local rule



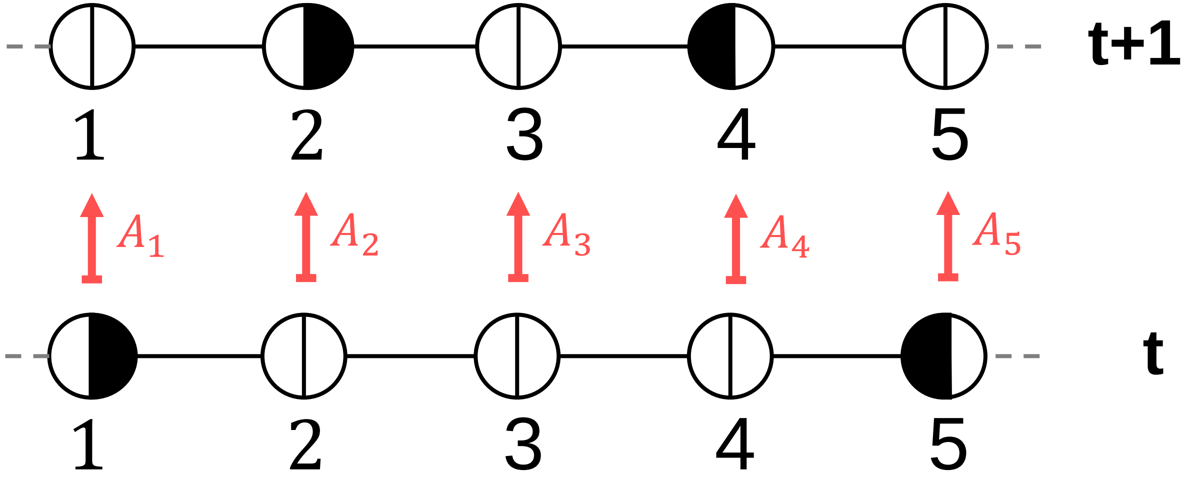


t

Dynamical systems and rewriting

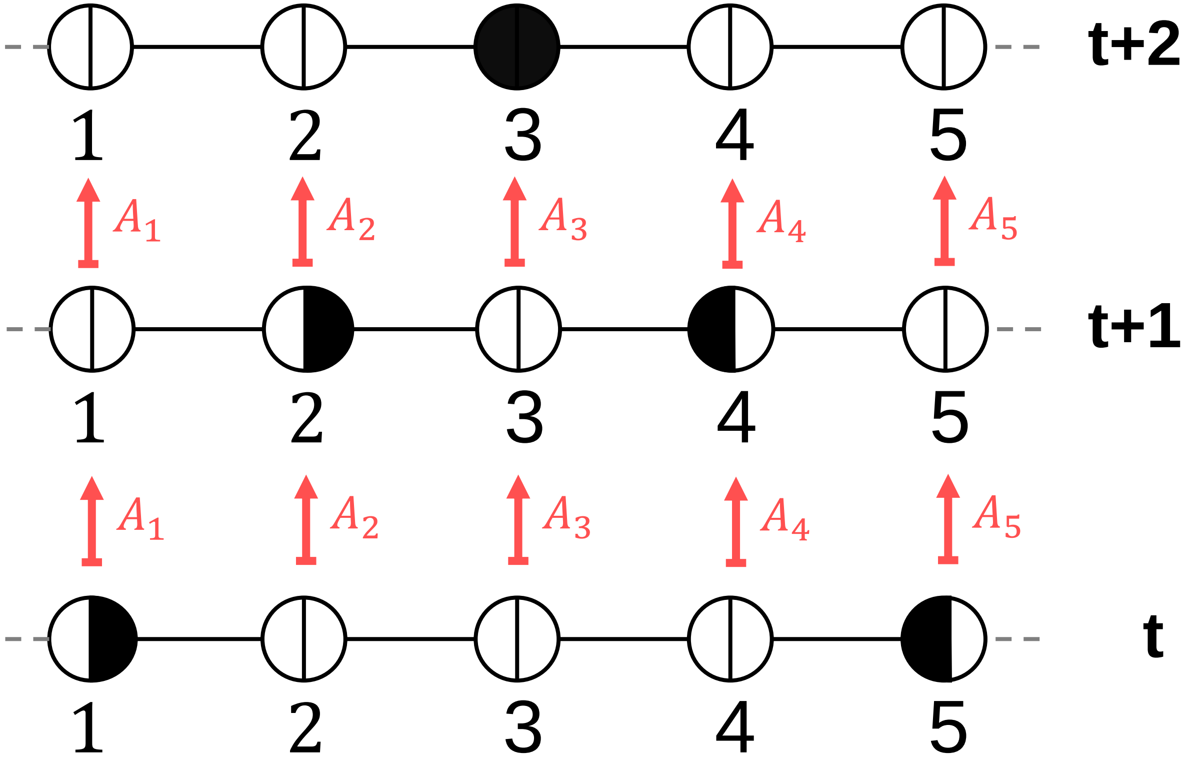


Dynamical system (Synchronous)

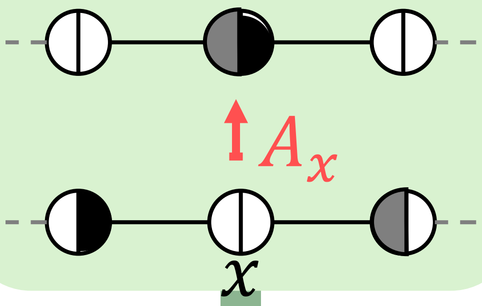


Dynamical systems and rewriting

Dynamical system (Synchronous)

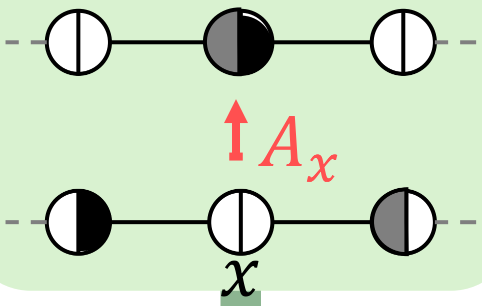


Local rule

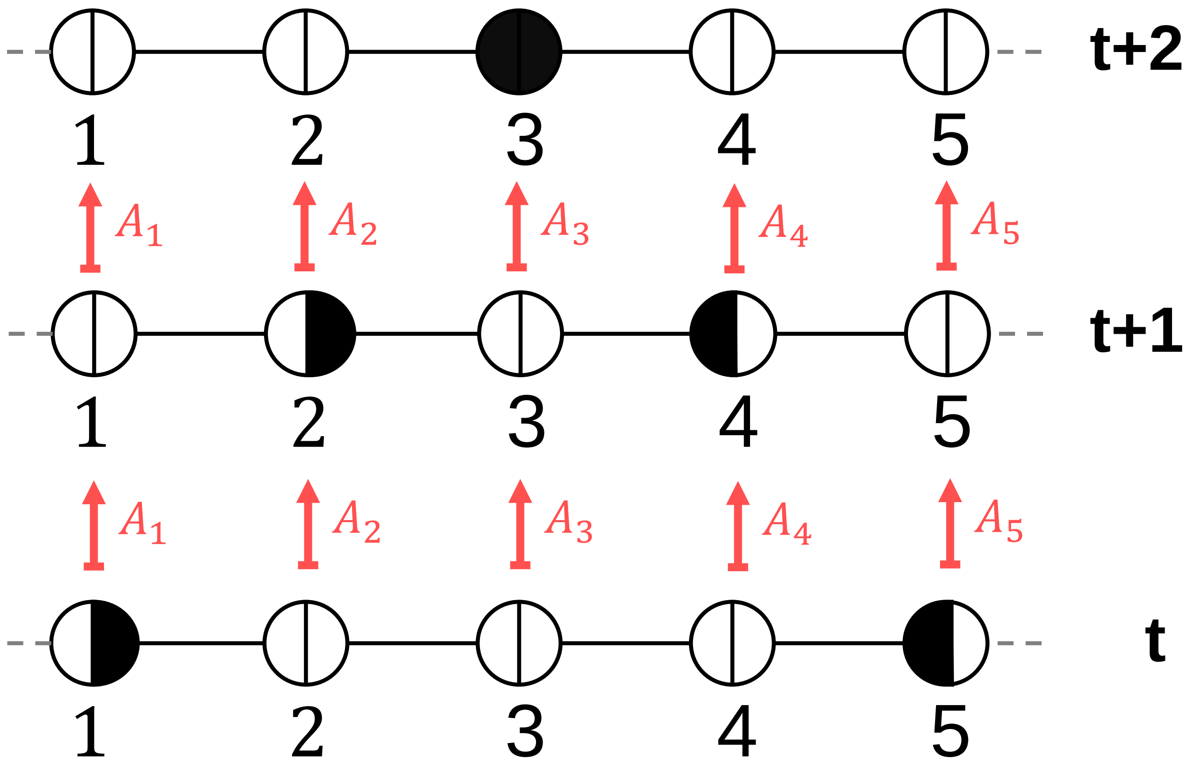


Dynamical systems and rewriting

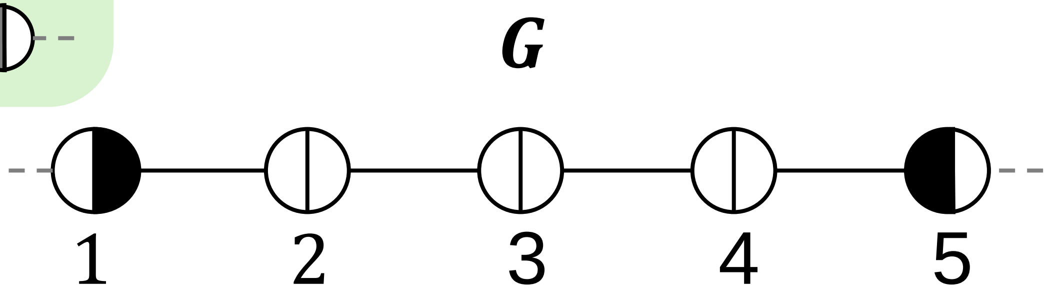
Local rule



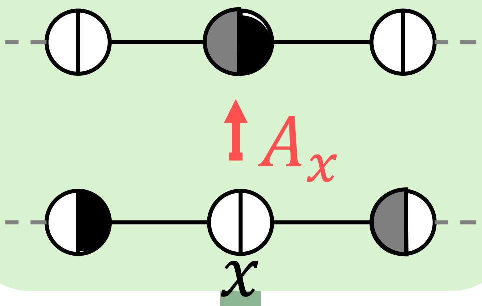
Dynamical system (Synchronous)



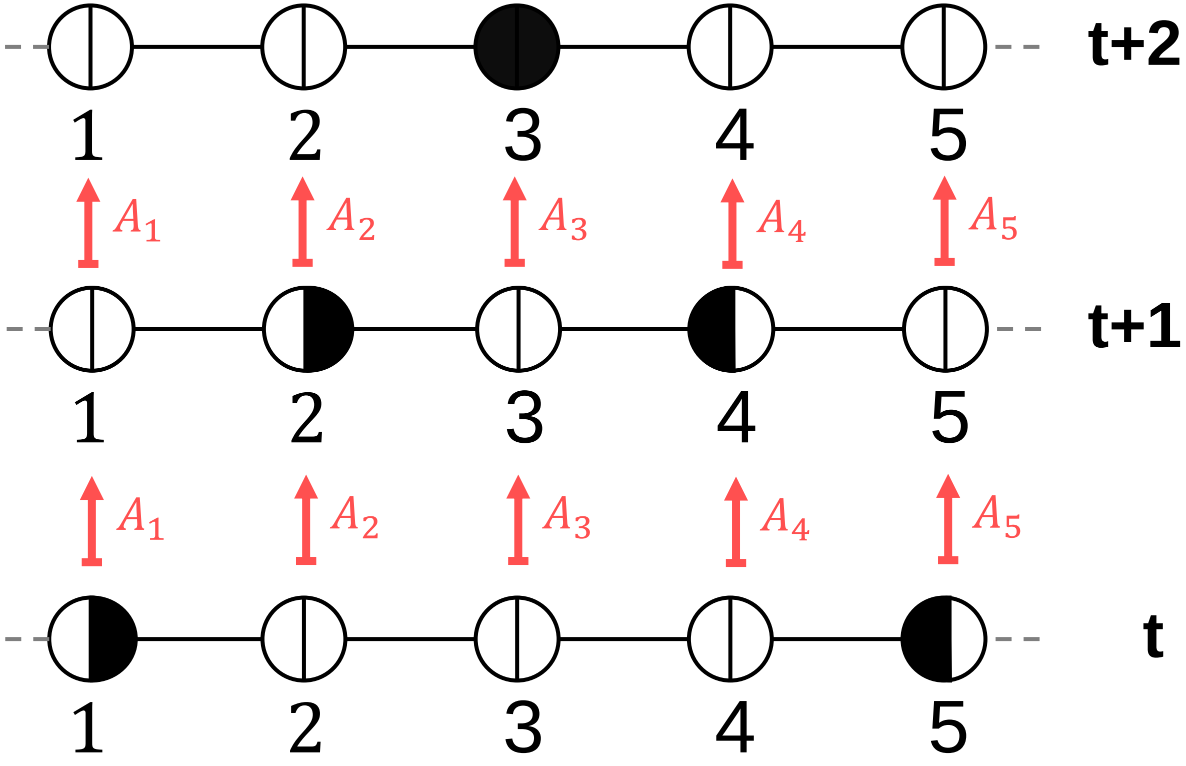
Naive asynchronous counterpart



Dynamical systems and rewriting

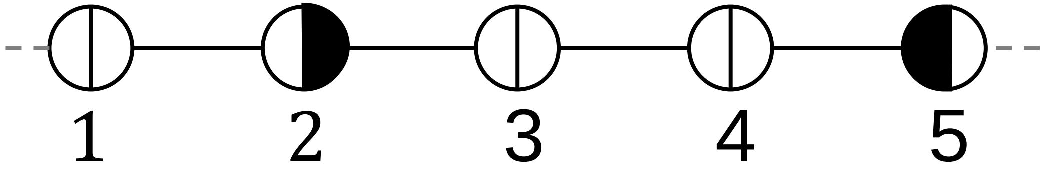


Dynamical system (Synchronous)



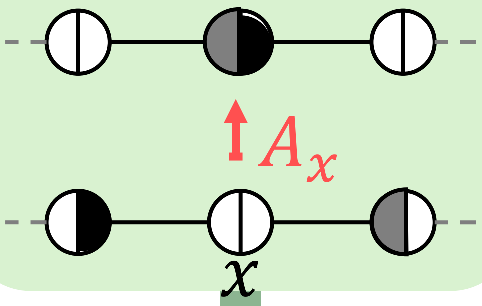
Naive asynchronous counterpart

$A_2 G$

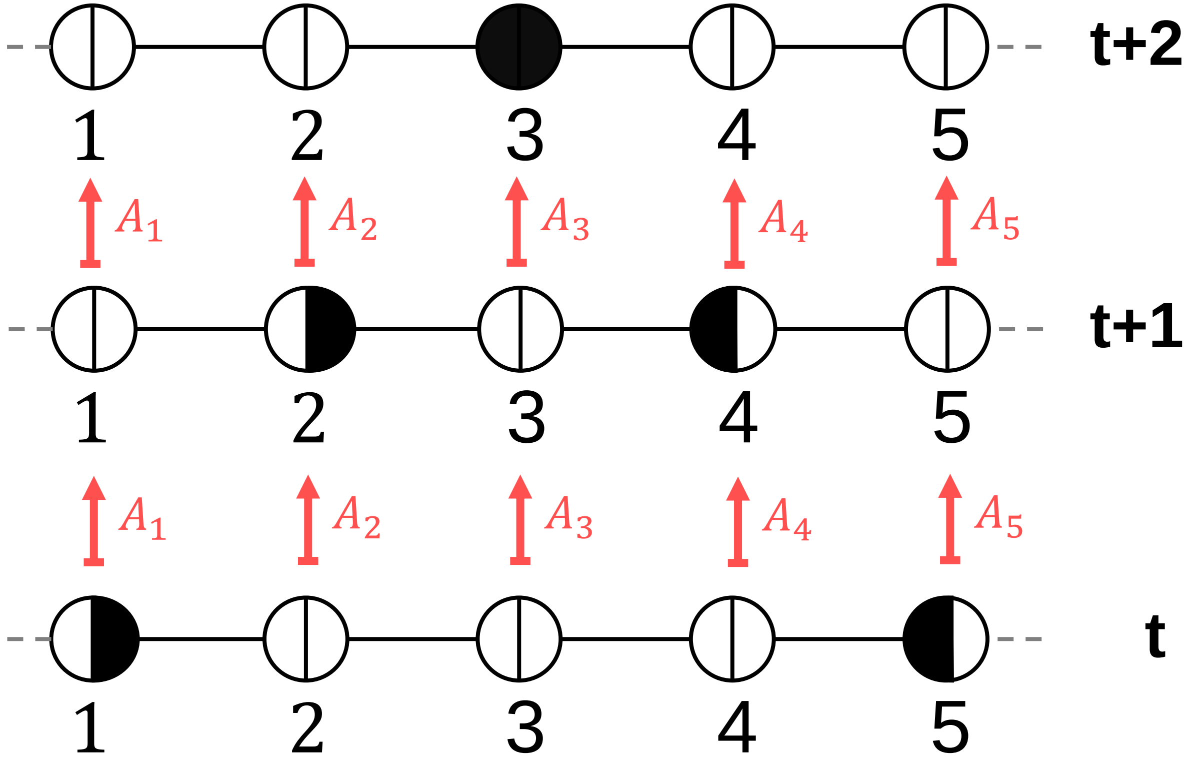


Dynamical systems and rewriting

Local rule

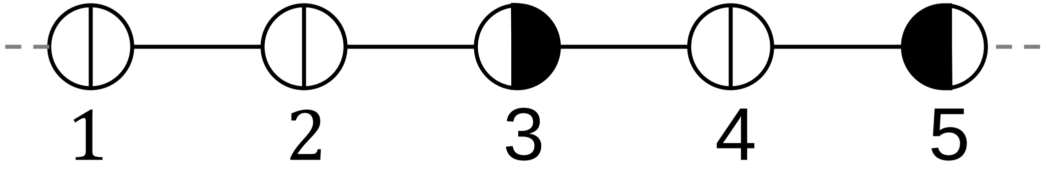


Dynamical system (Synchronous)



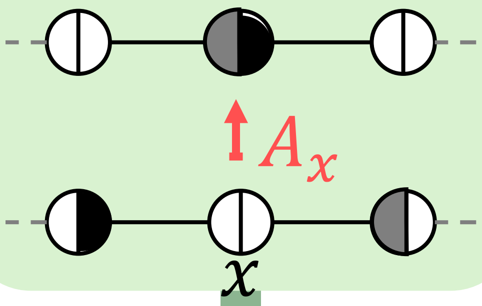
Naive asynchronous counterpart

$A_3 A_2 G$

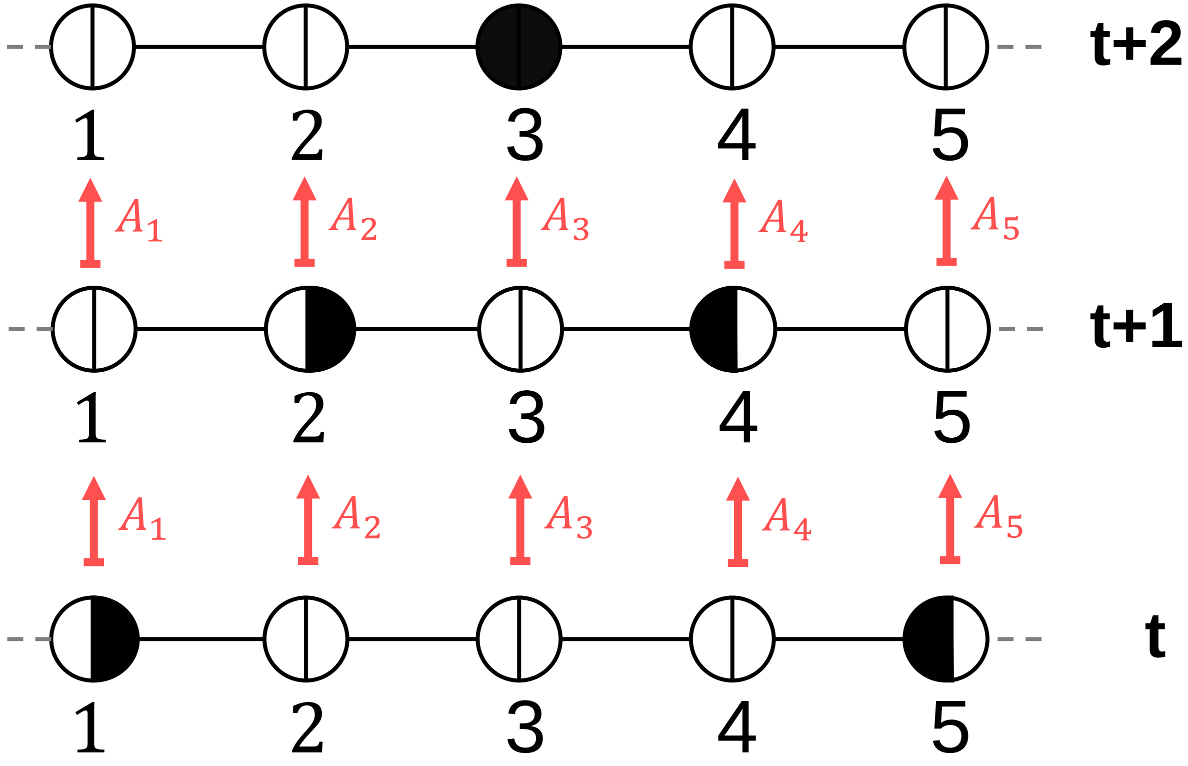


Dynamical systems and rewriting

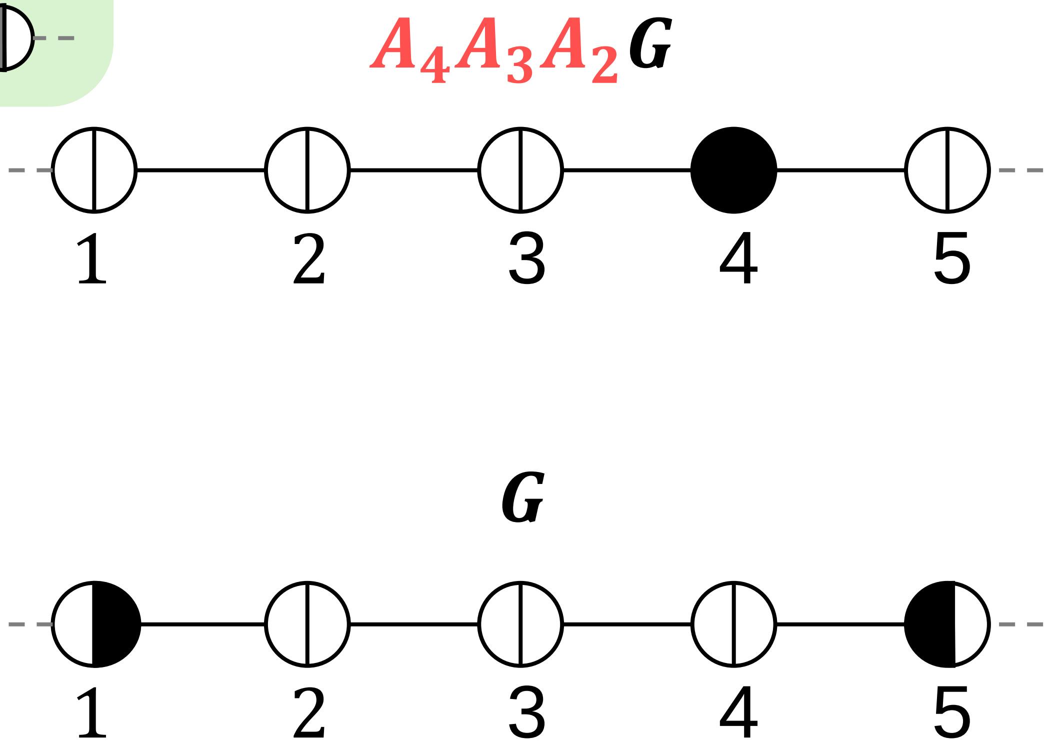
Local rule



Dynamical system (Synchronous)

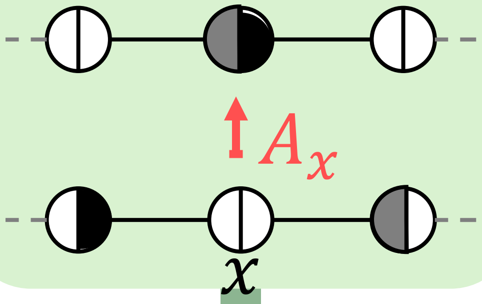


Naive asynchronous counterpart

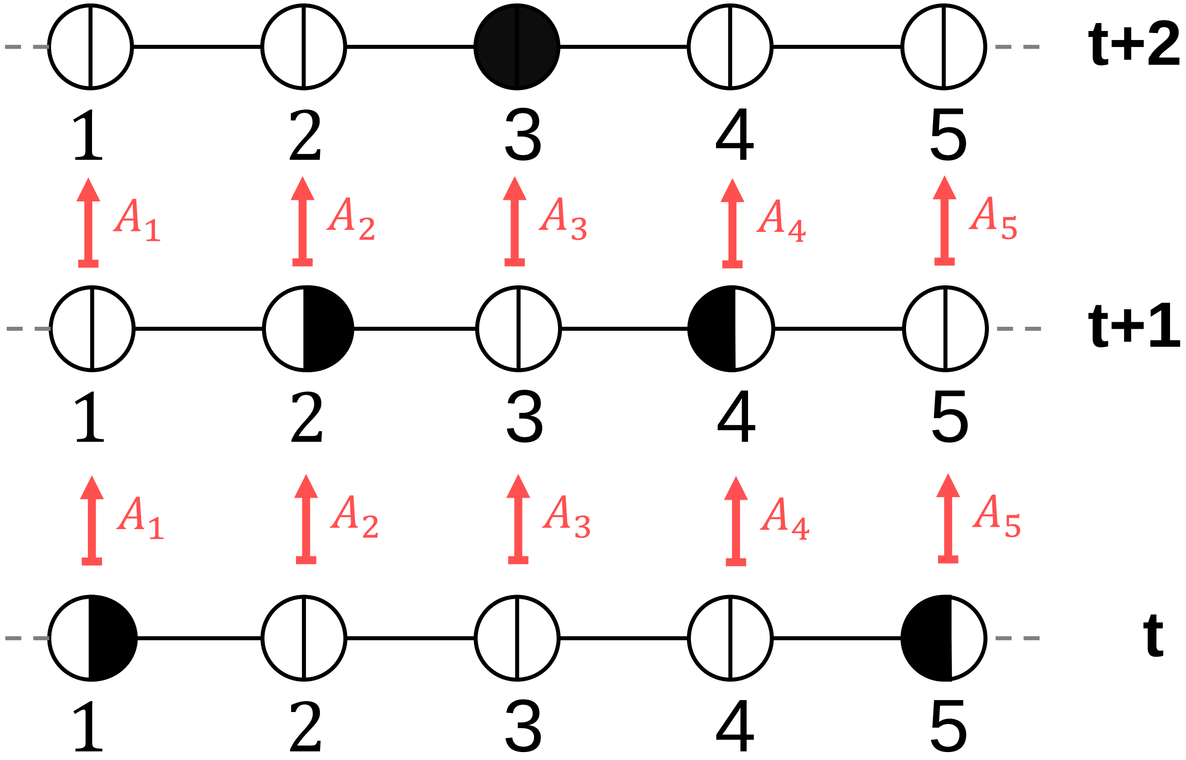


Dynamical systems and rewriting

Local rule

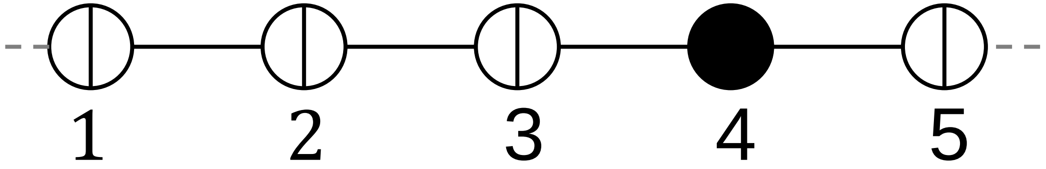


Dynamical system (Synchronous)

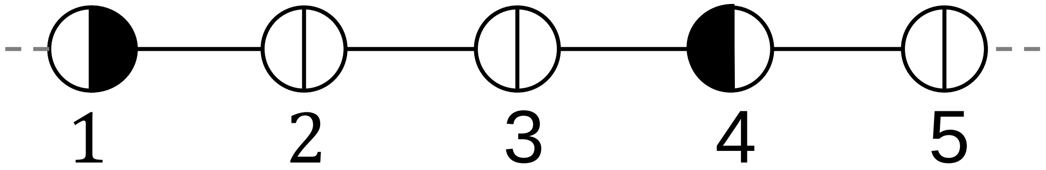


Naive asynchronous counterpart

$A_4 A_3 A_2 G$

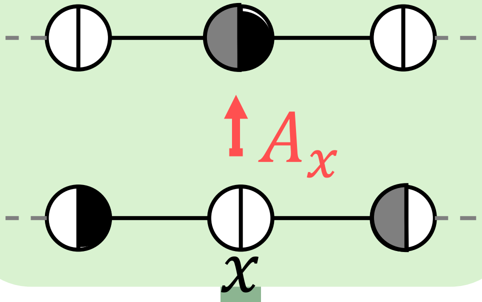


$A_4 G$

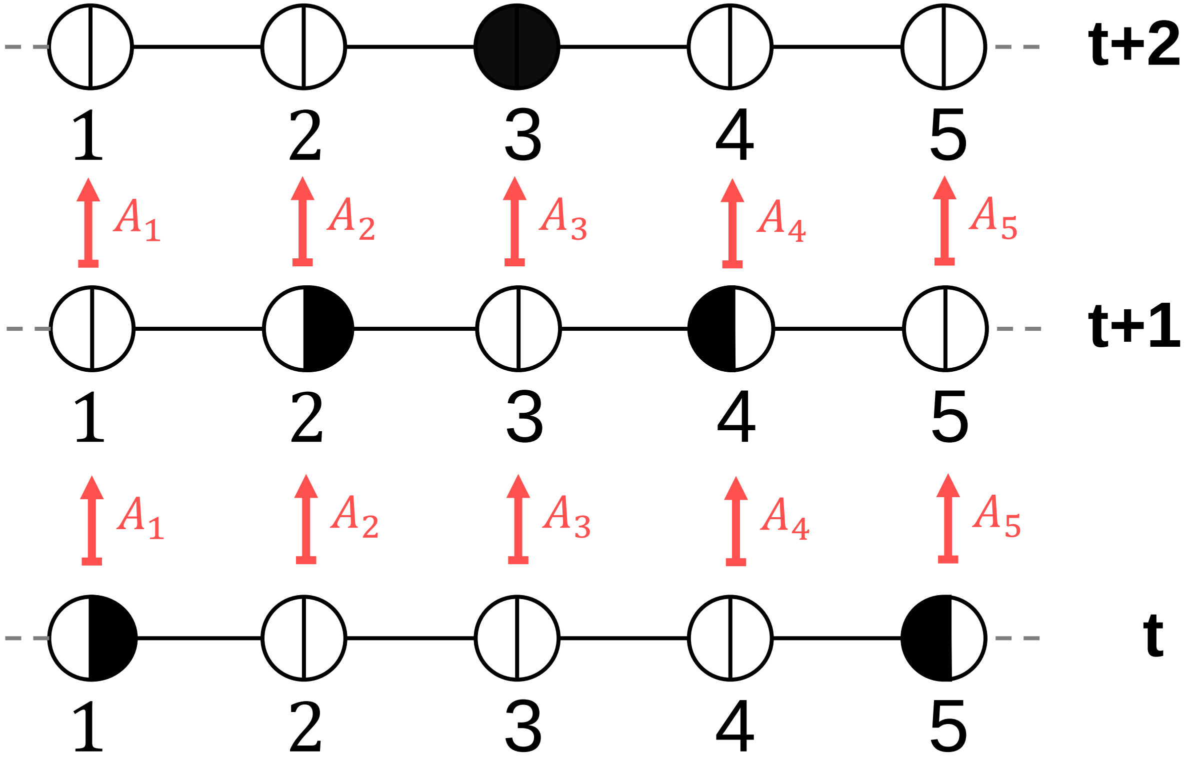


Dynamical systems and rewriting

Local rule

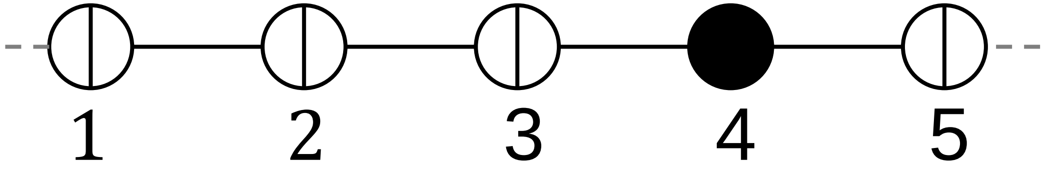


Dynamical system (Synchronous)

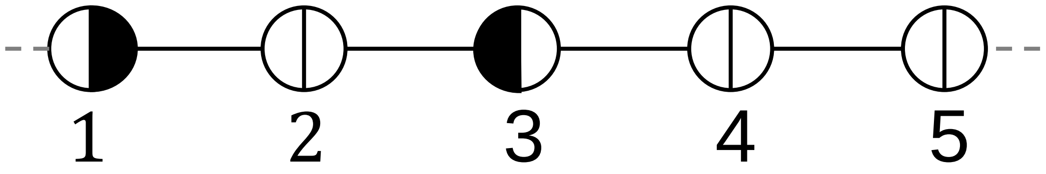


Naive asynchronous counterpart

$A_4 A_3 A_2 G$

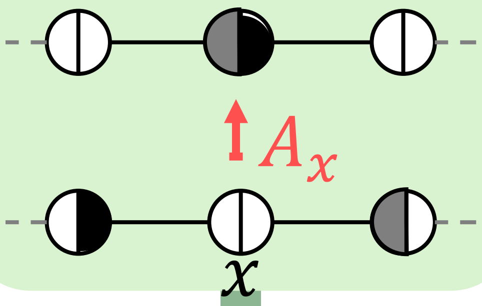


$A_3 A_4 G$

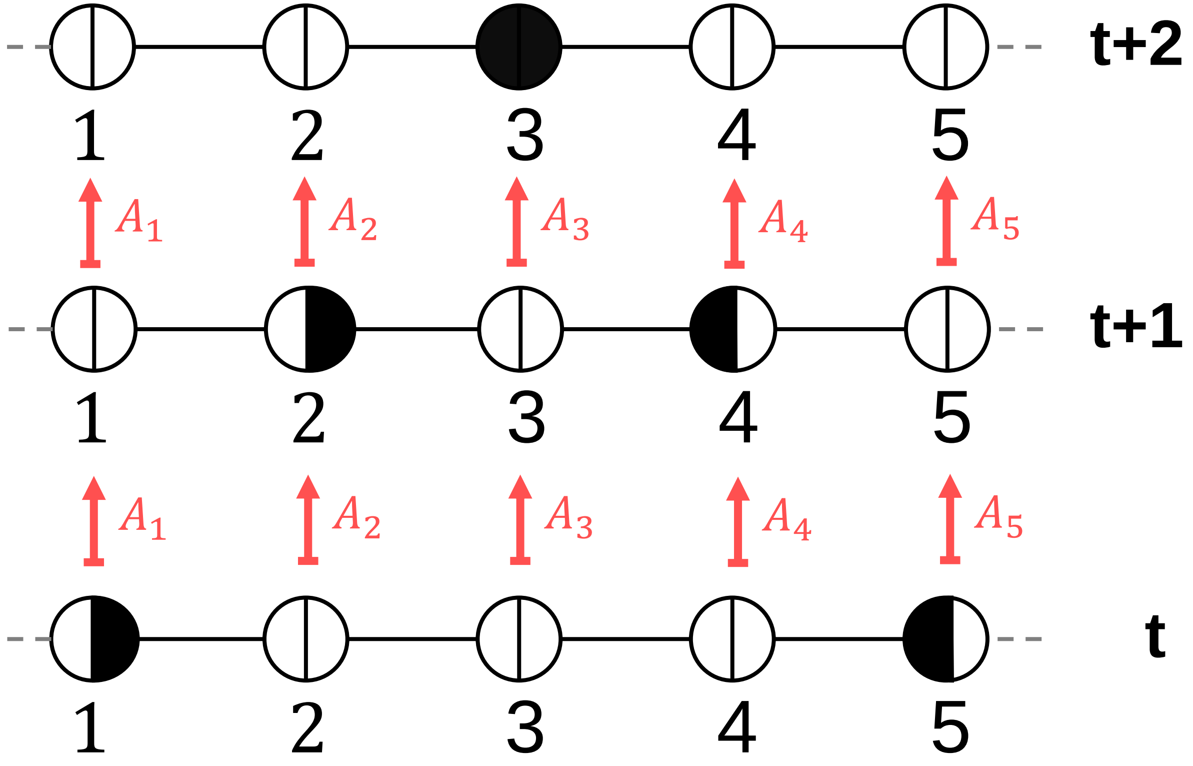


Dynamical systems and rewriting

Local rule

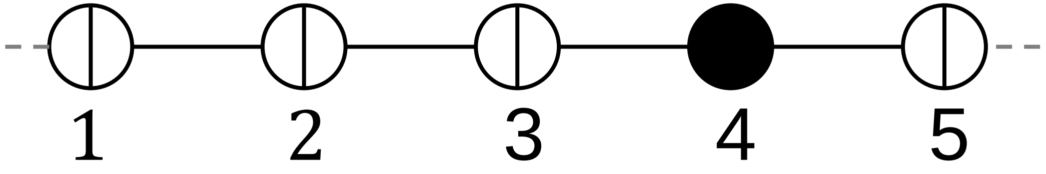


Dynamical system (Synchronous)

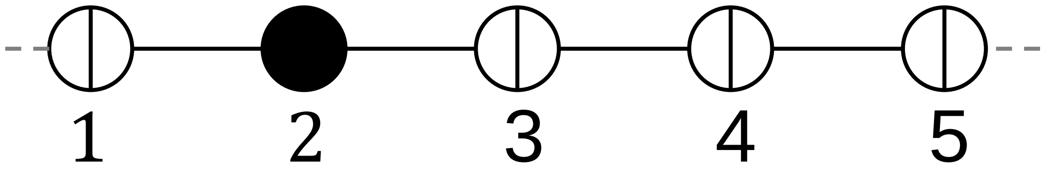


Naive asynchronous counterpart

$A_4 A_3 A_2 G$

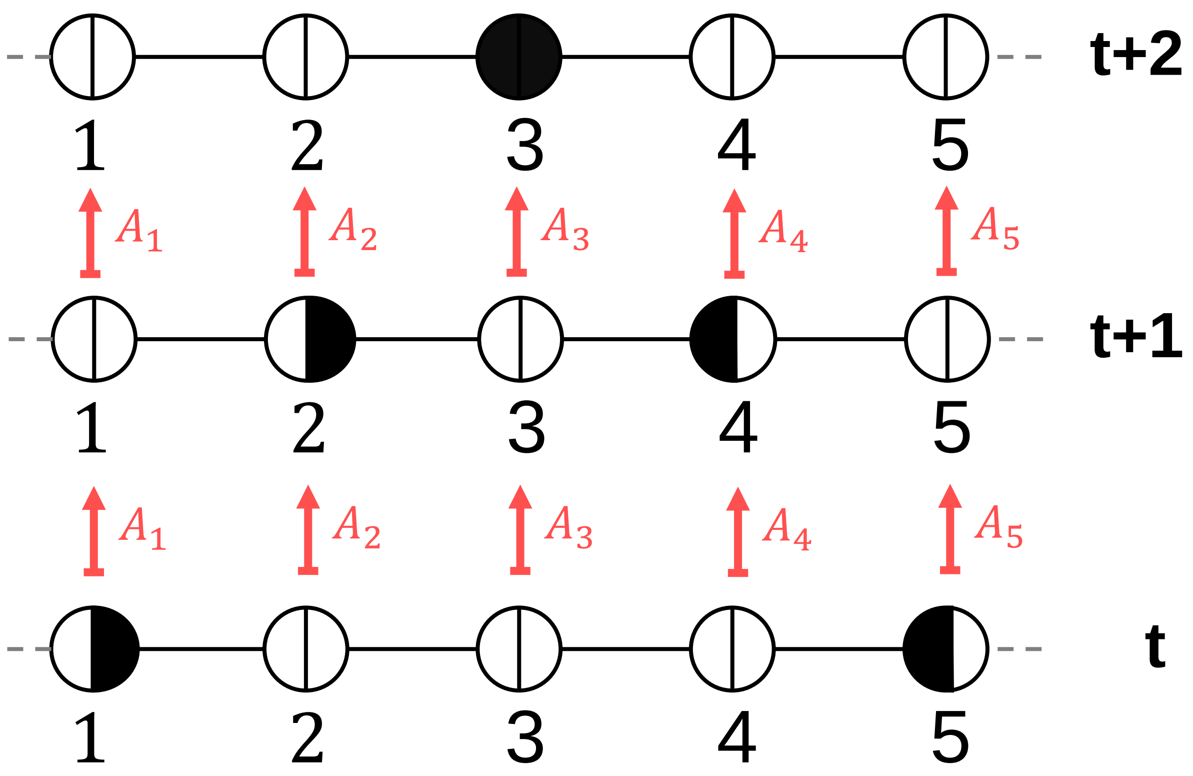


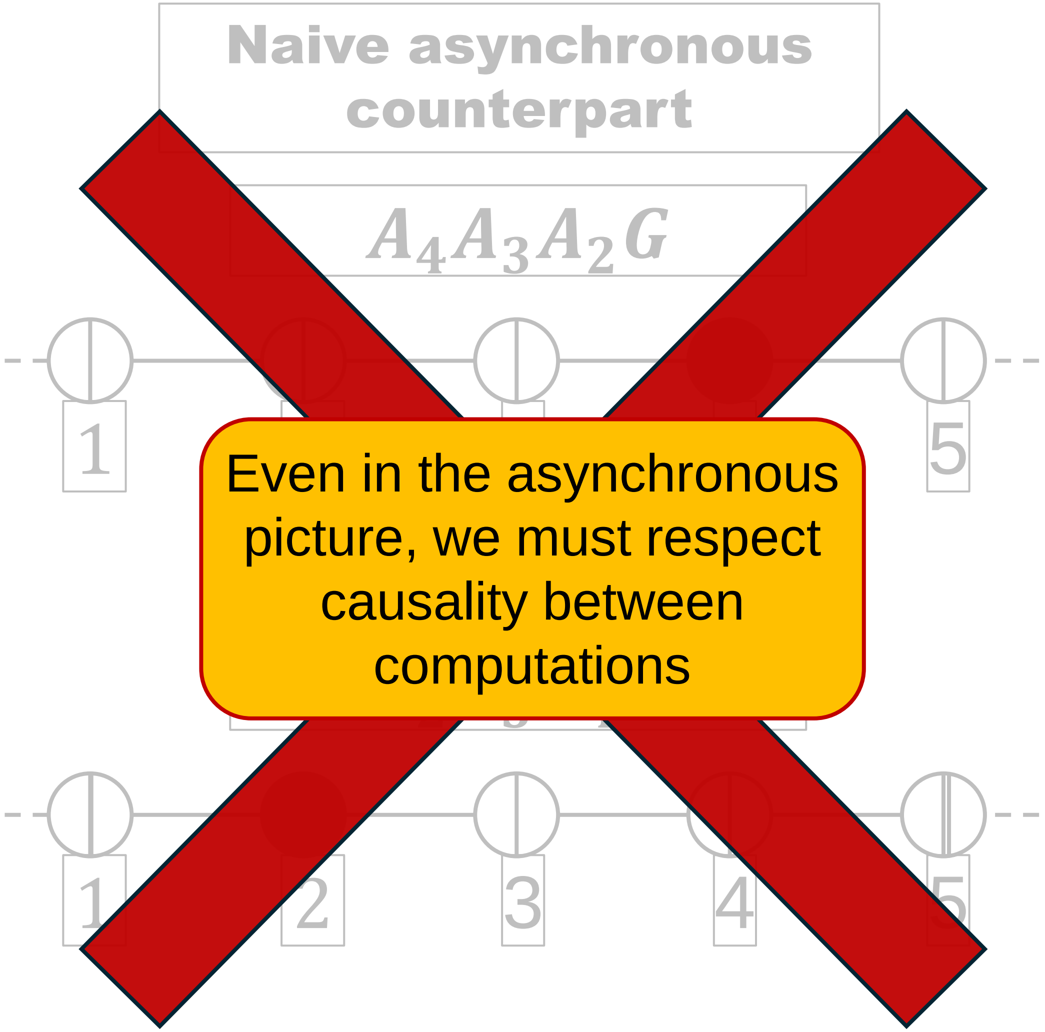
$A_2 A_3 A_4 G$



Dynamical systems and rewriting

Dynamical system (Synchronous)

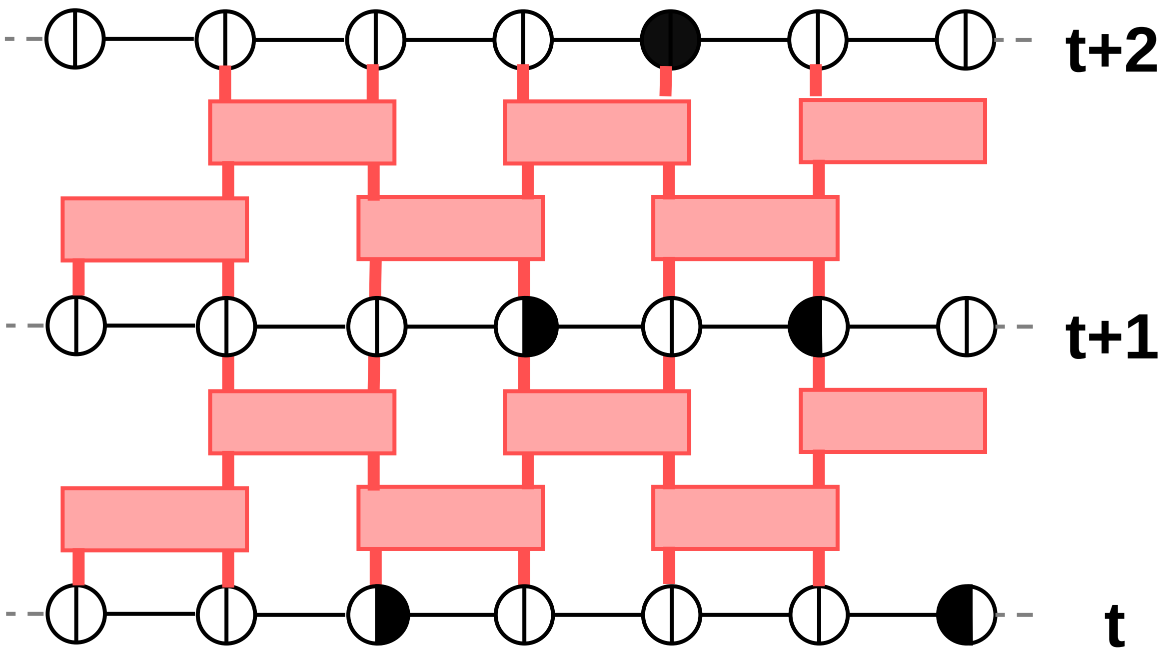


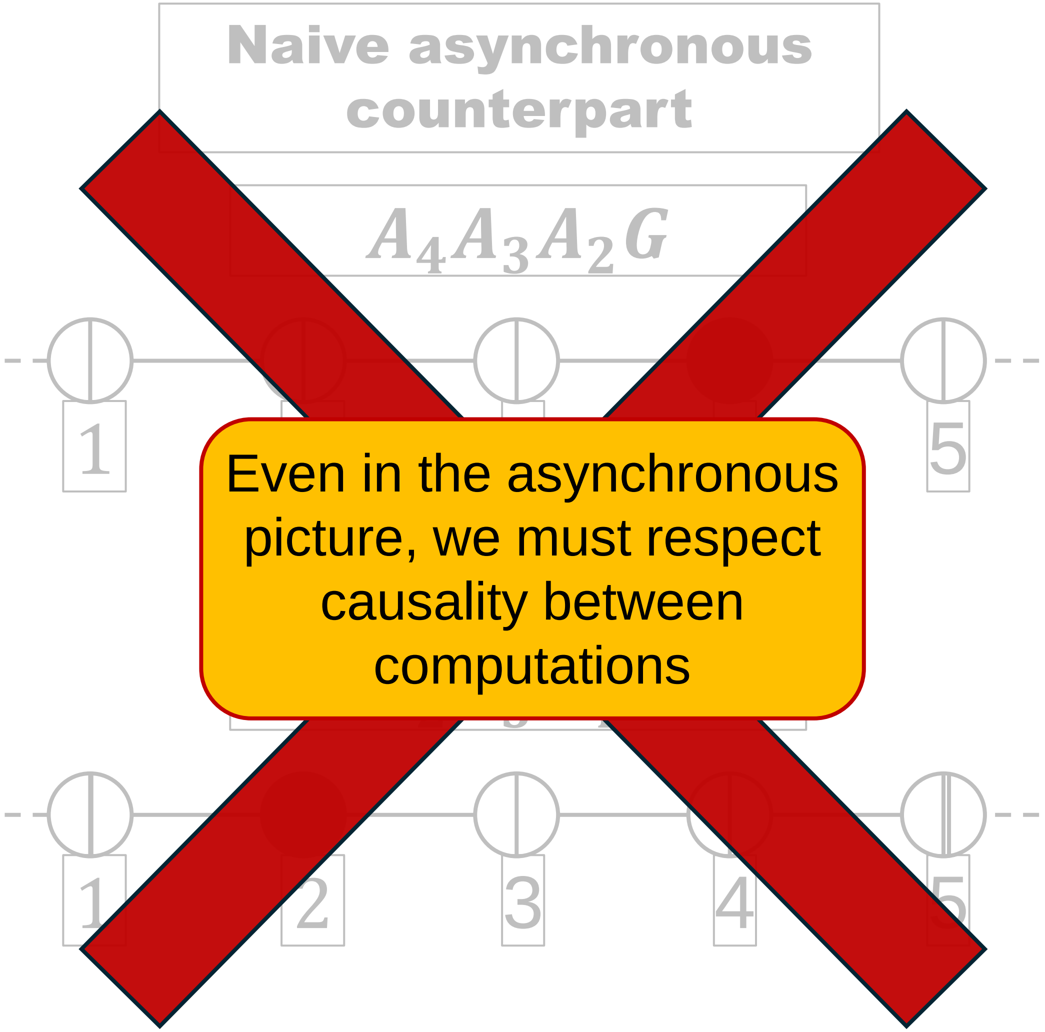


Even in the asynchronous picture, we must respect causality between computations

Dynamical systems and rewriting

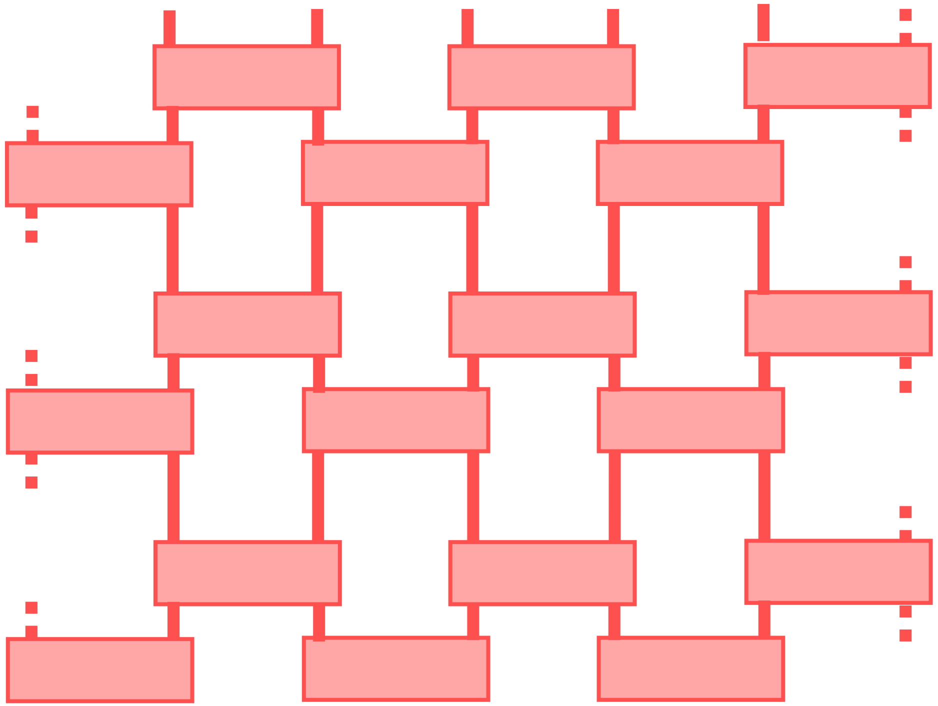
Causal structure

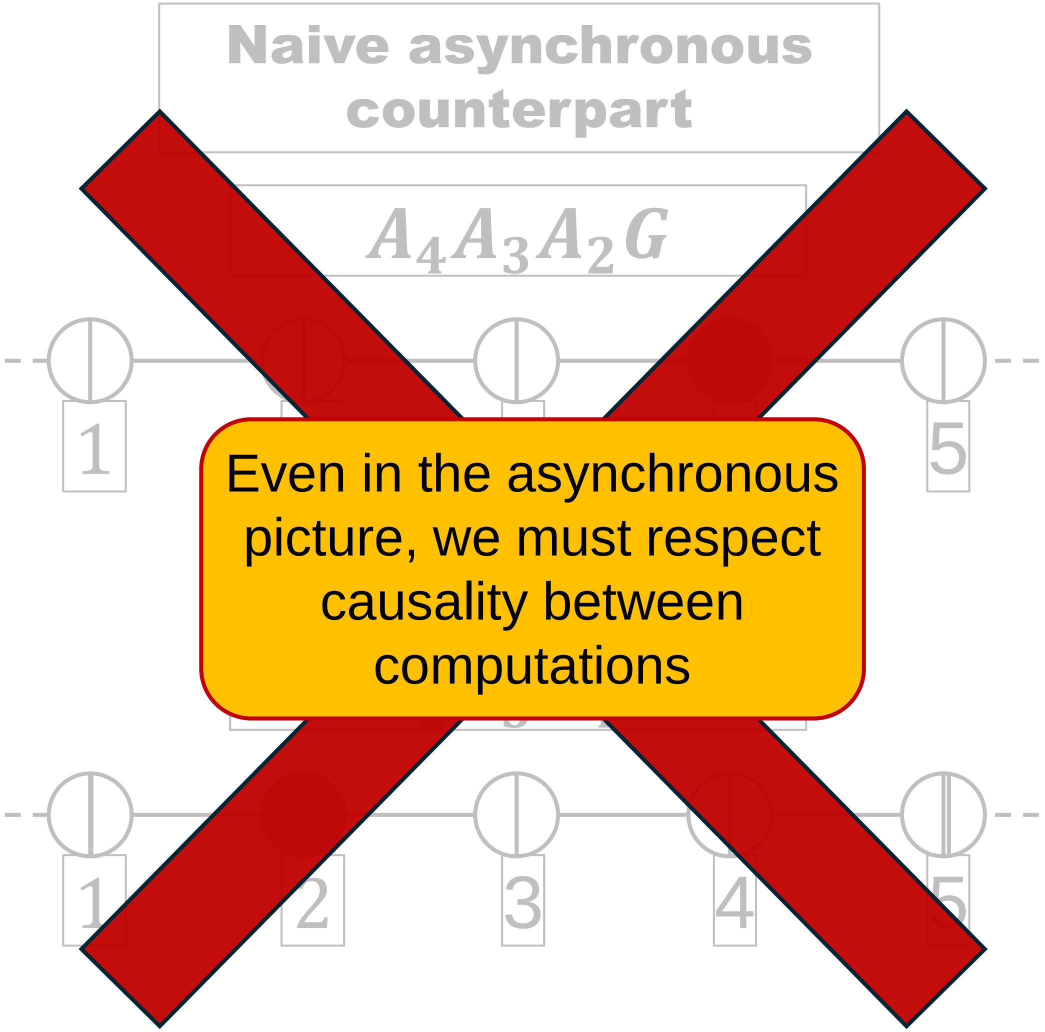




Dynamical systems and rewriting

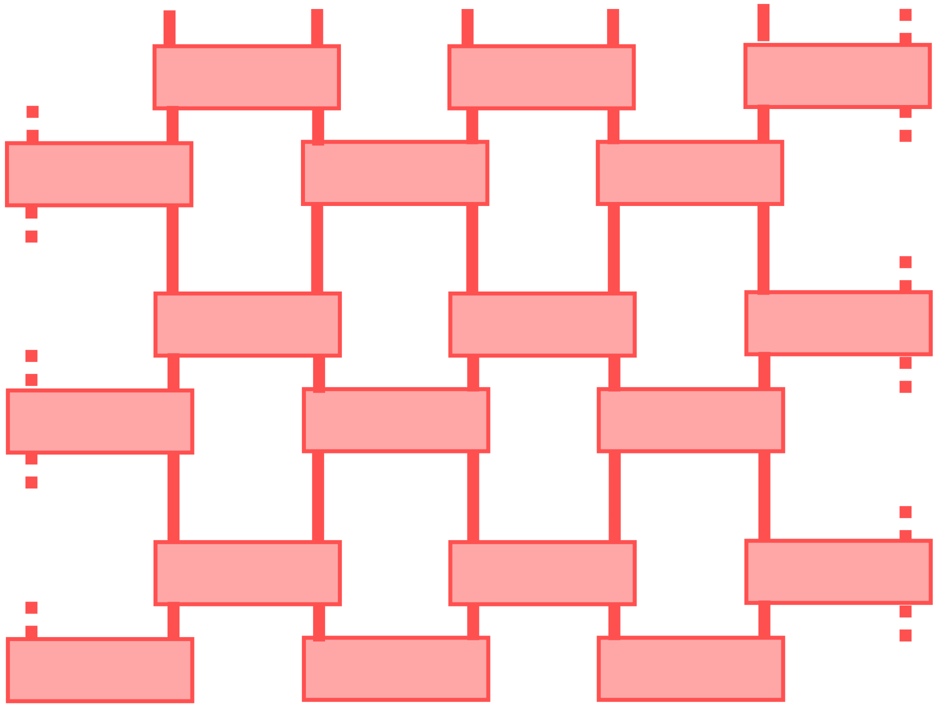
Causal structure



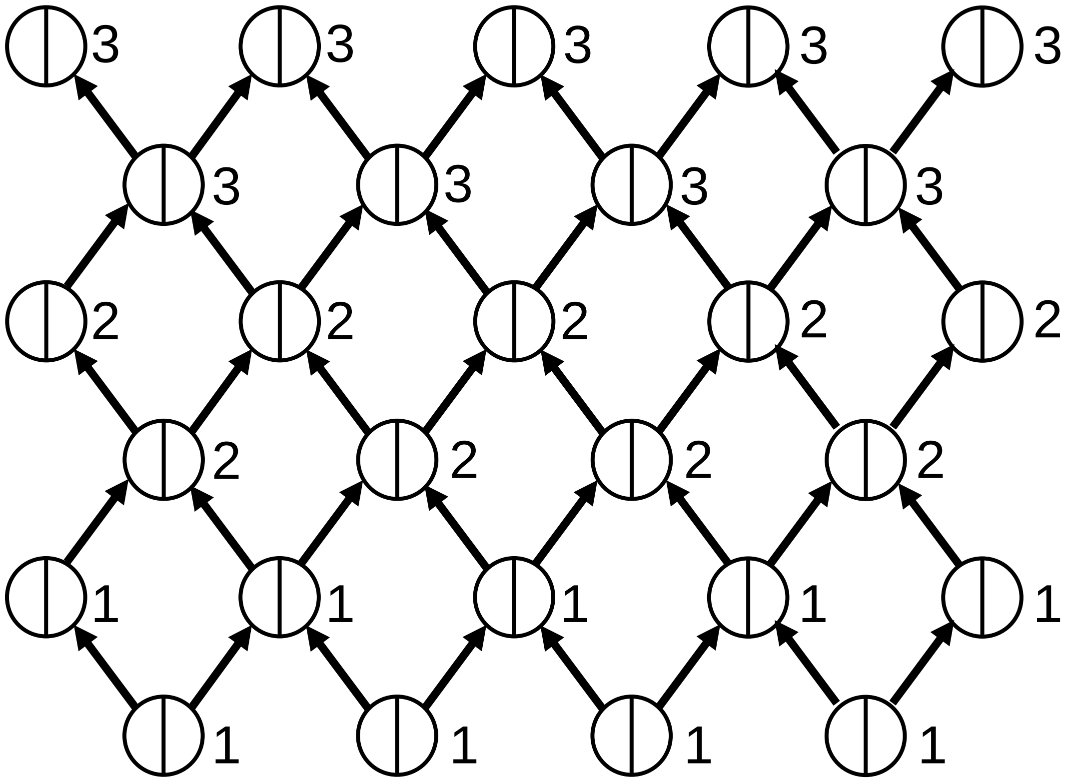


Dynamical systems and rewriting

Causal structure

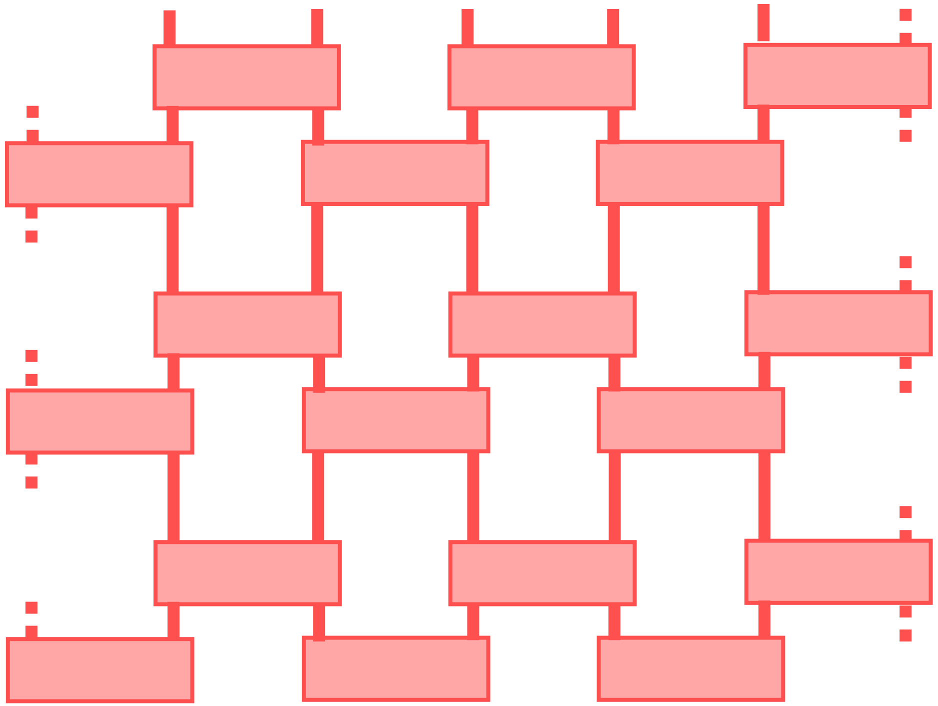


Causal graph rewriting

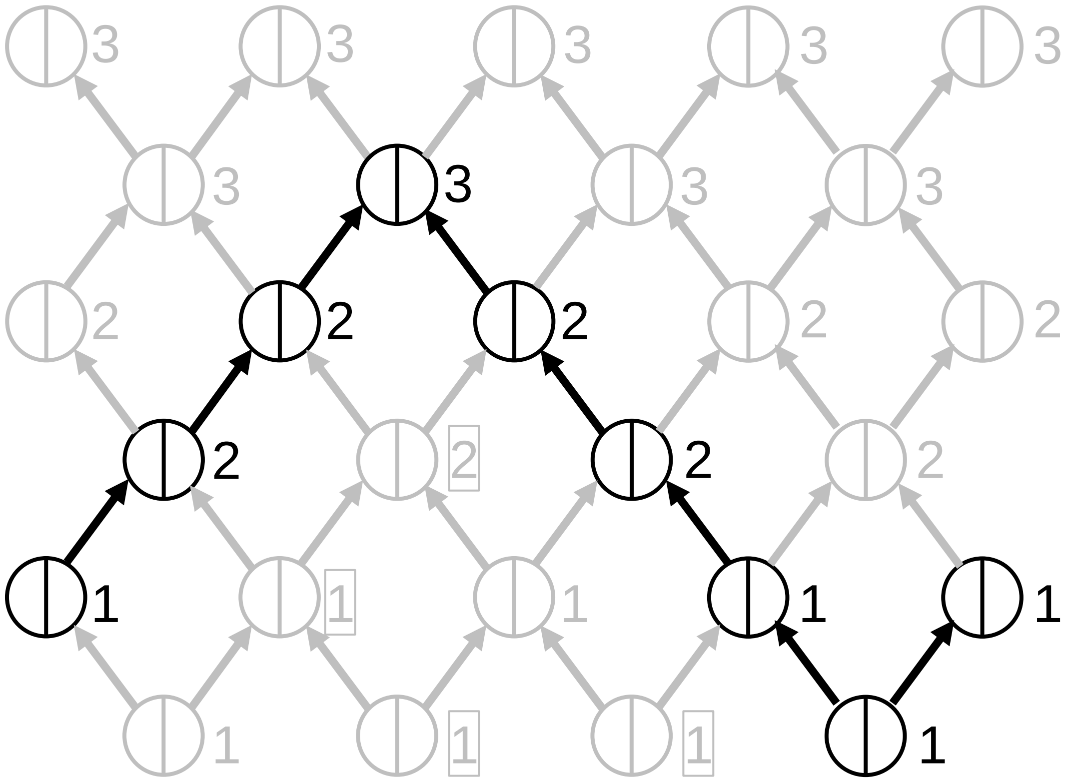


Dynamical systems and rewriting

Causal structure

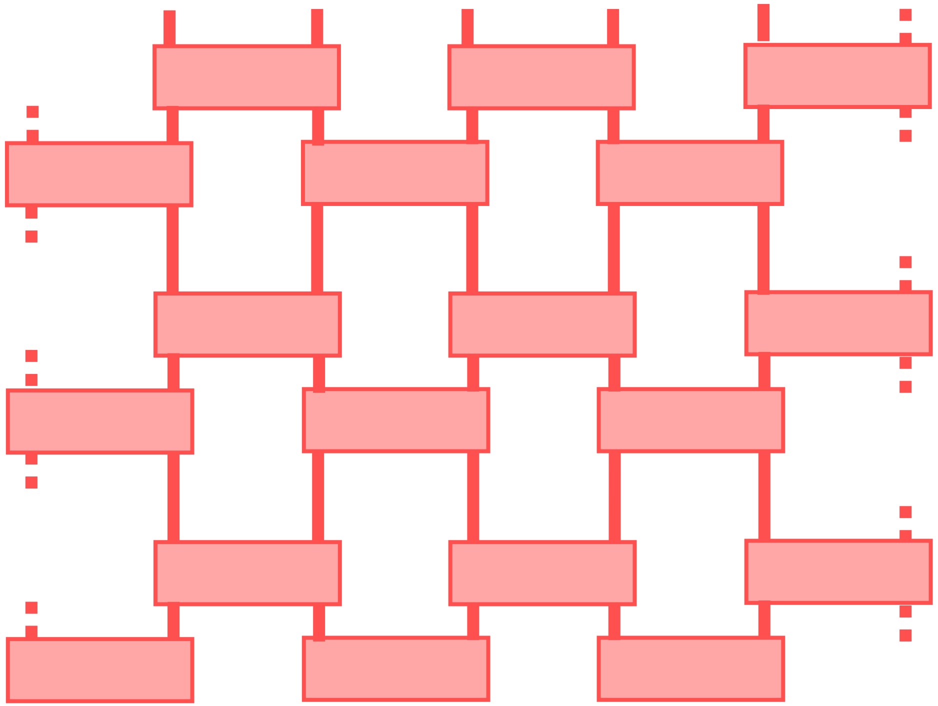


Causal graph rewriting

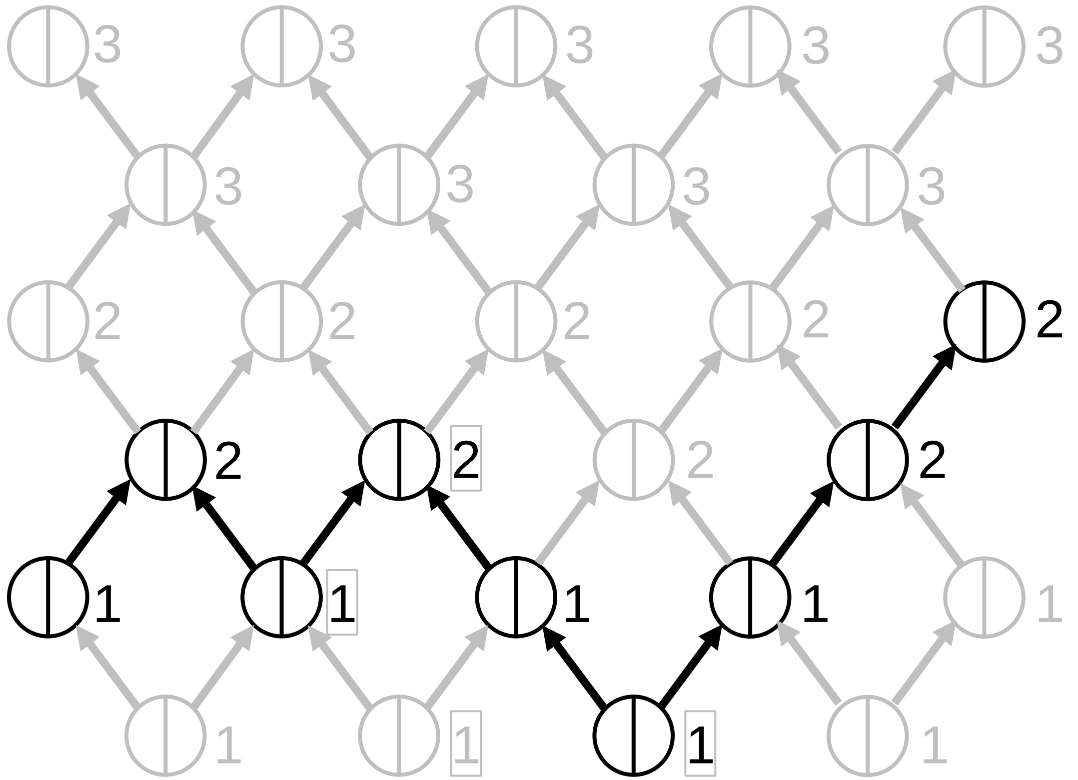


Dynamical systems and rewriting

Causal structure

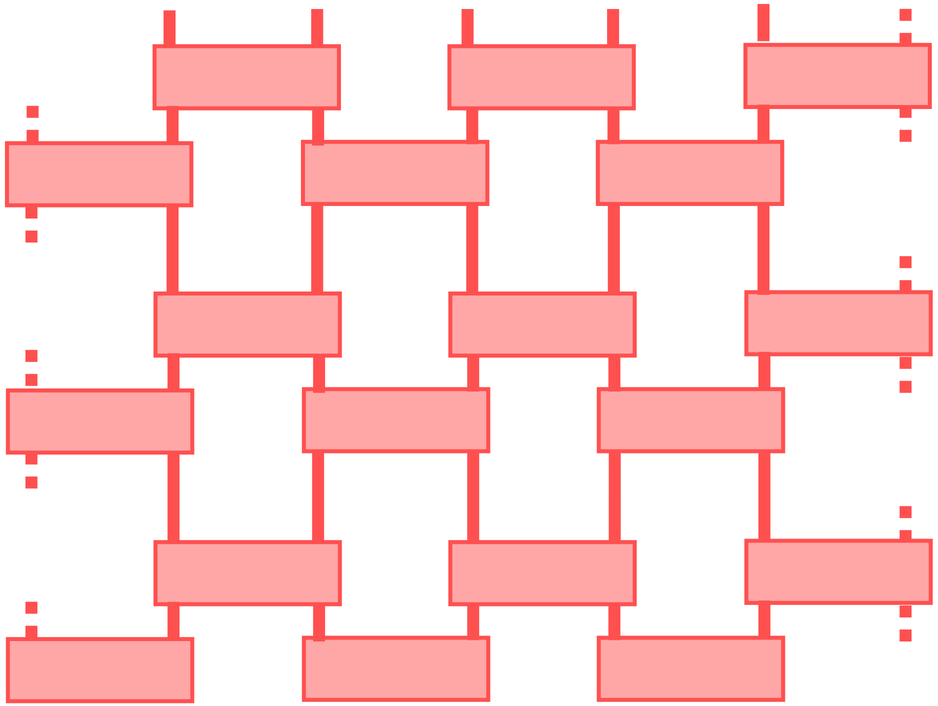


Causal graph rewriting

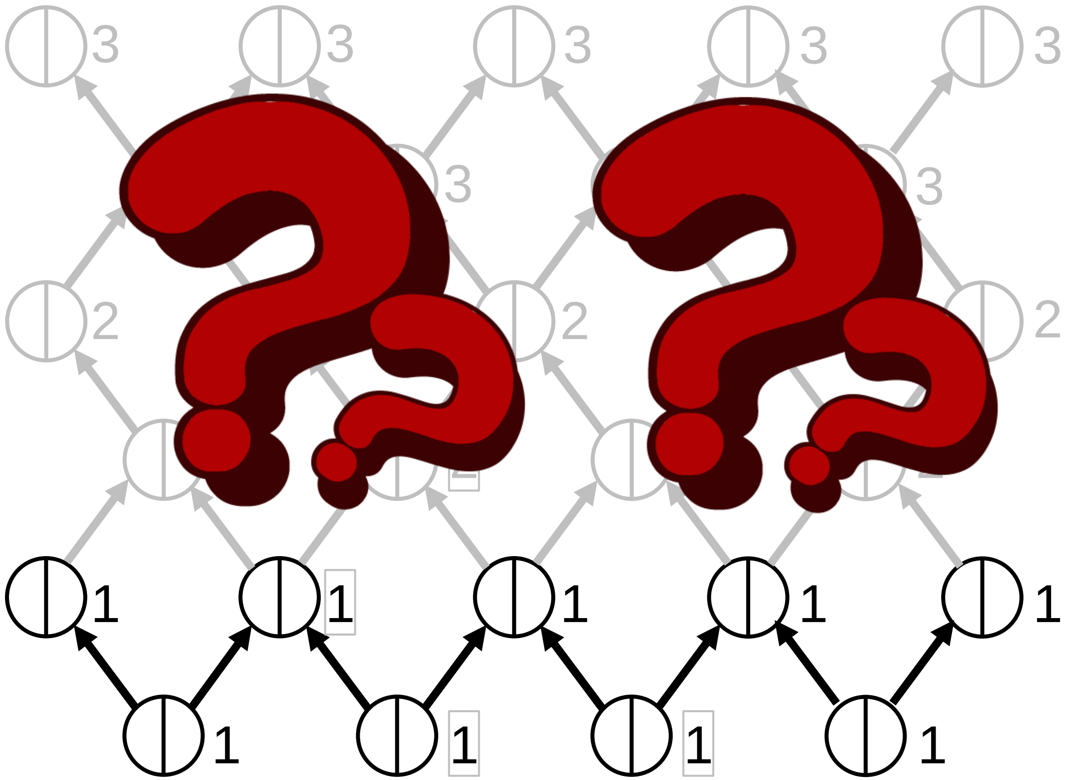


Dynamical systems and rewriting

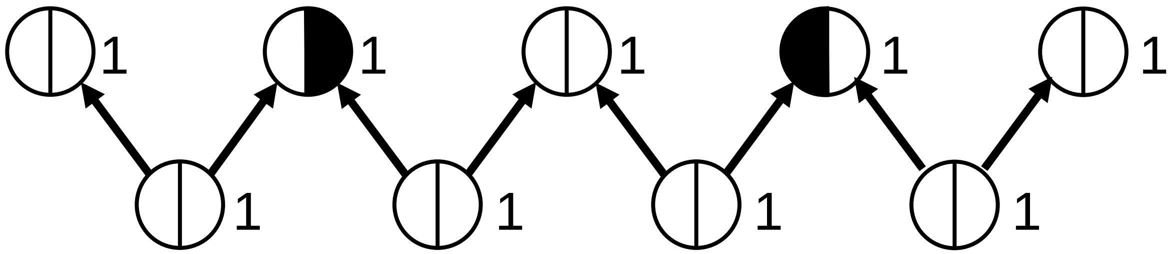
Causal structure



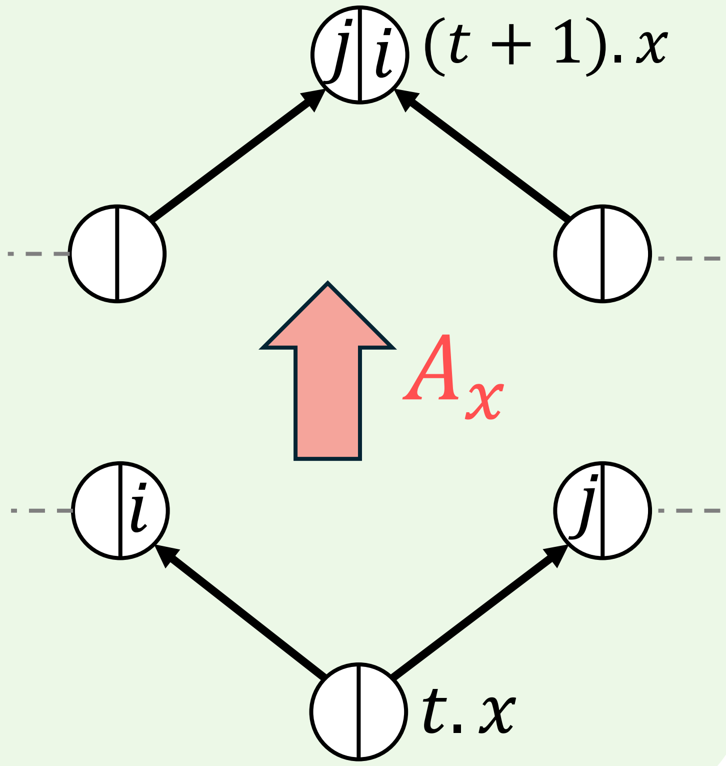
Causal graph rewriting



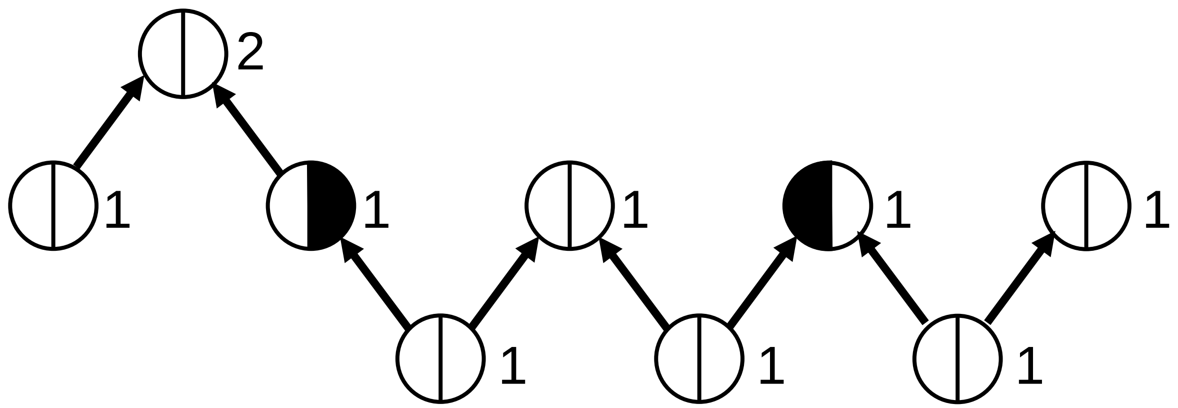
Particle system example



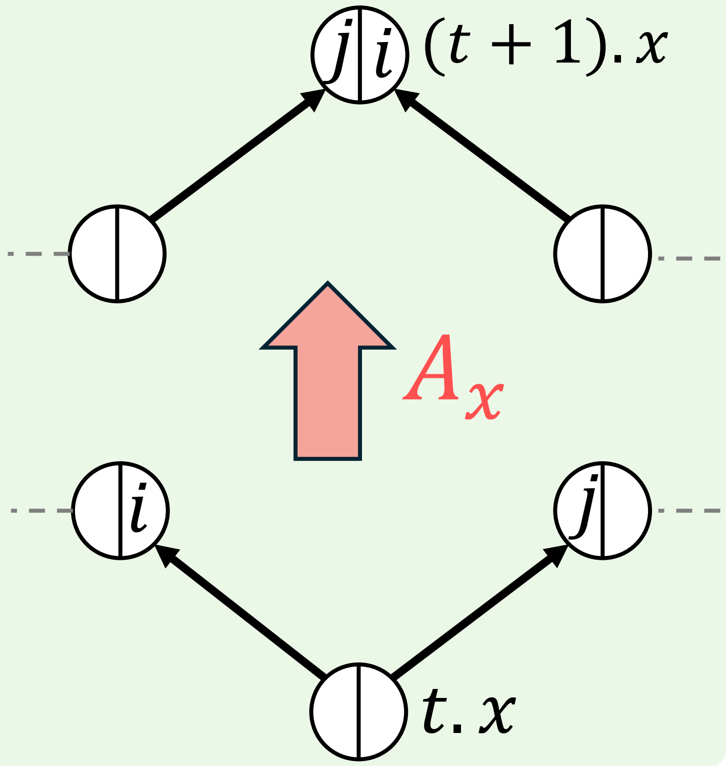
Local rule



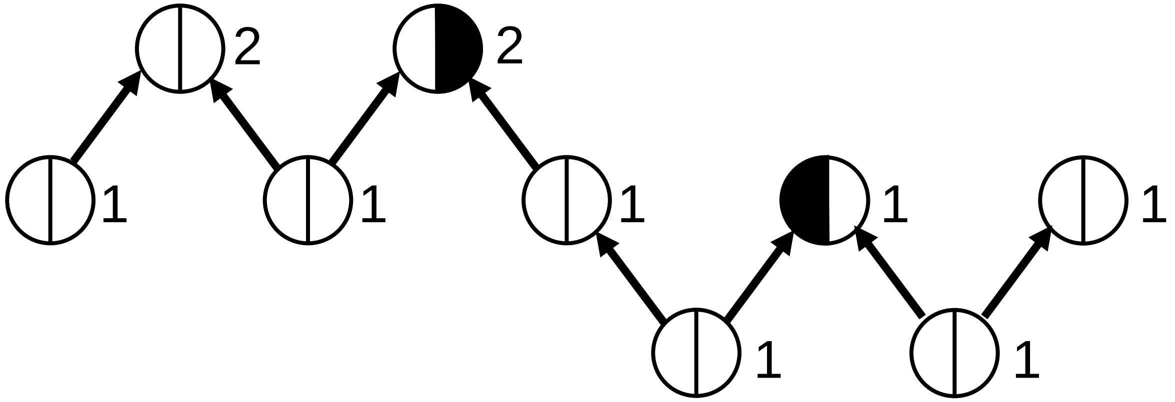
Particle system example

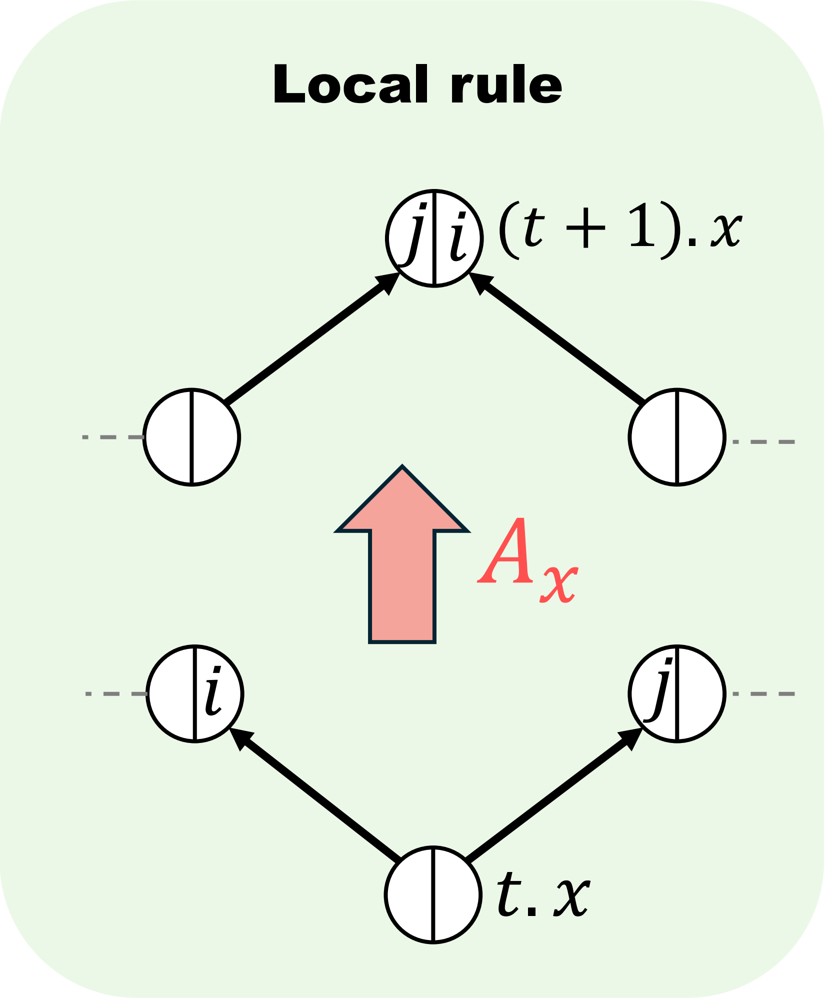


Local rule

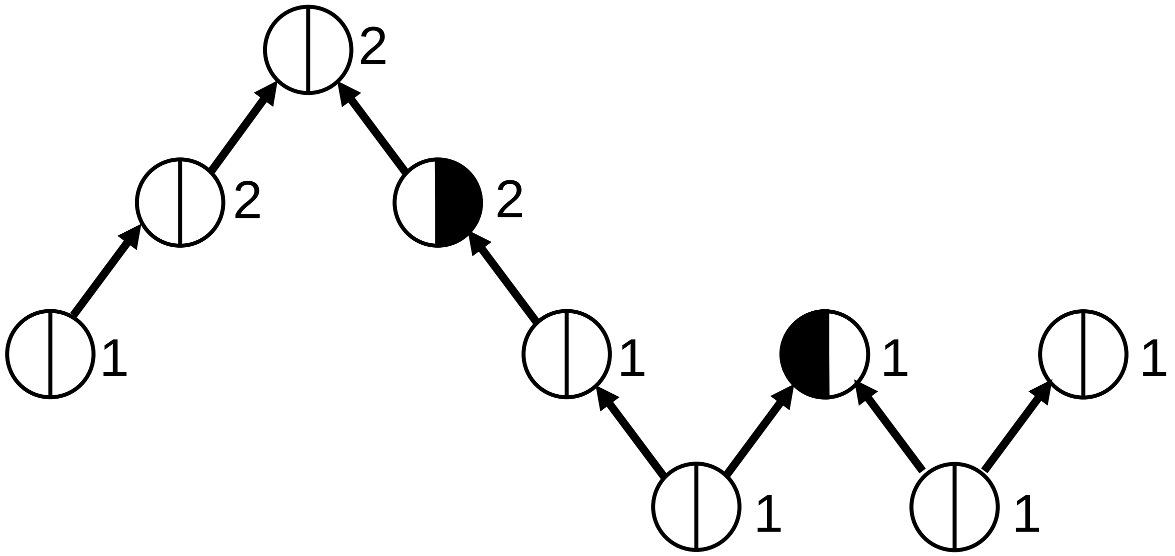


Particle system example

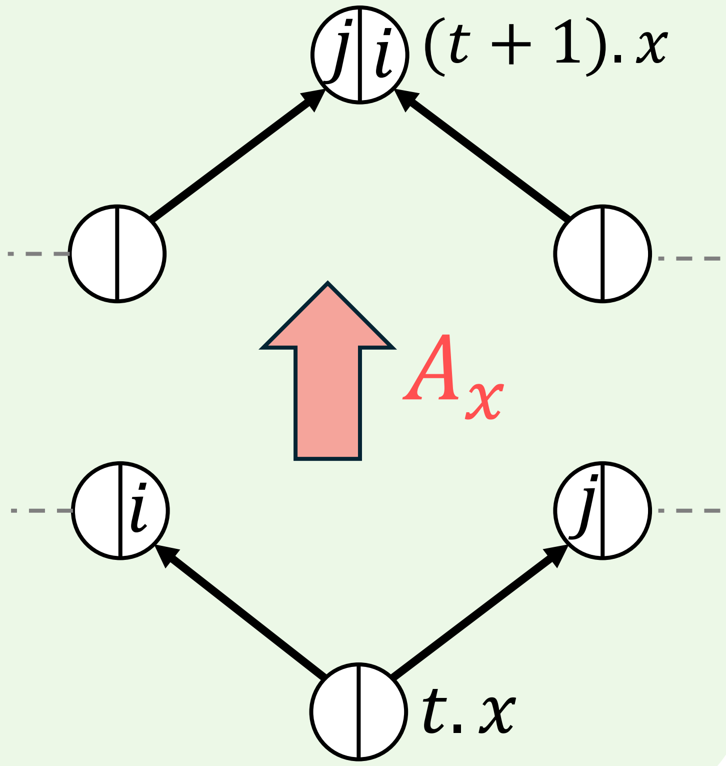




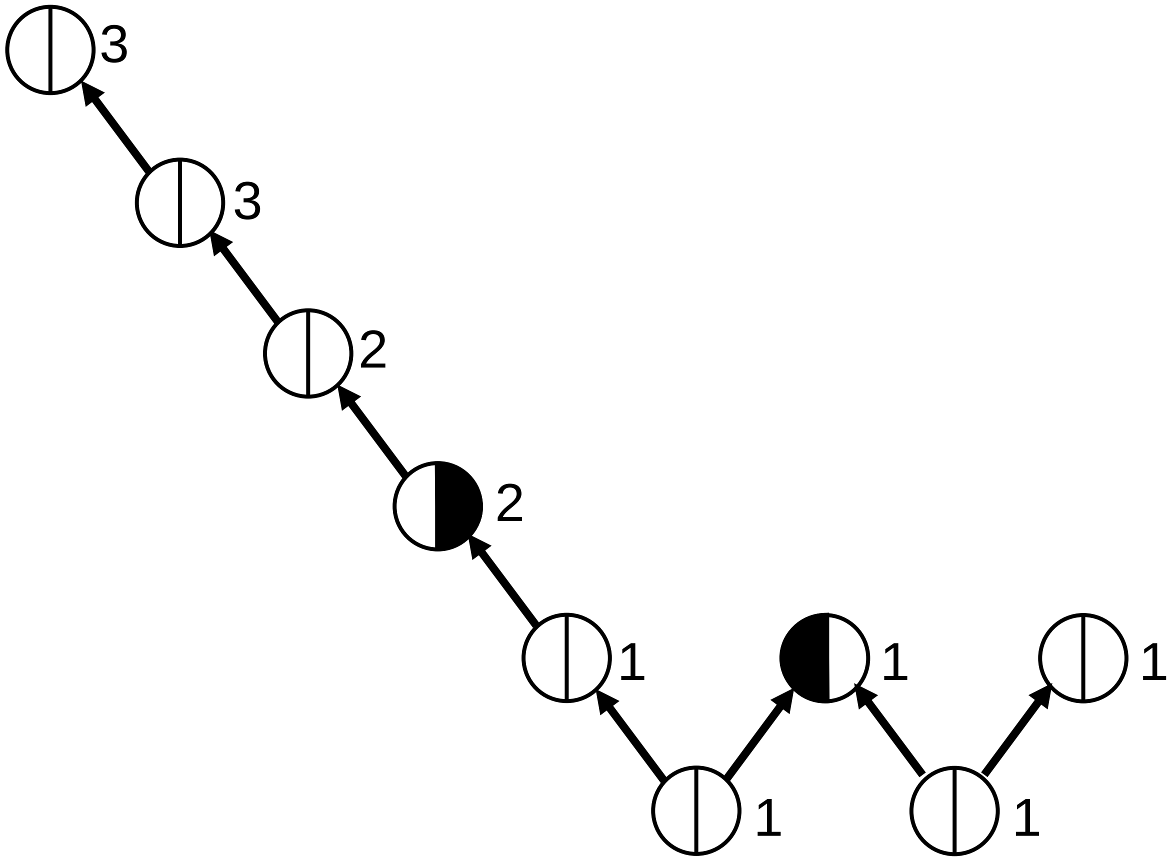
Particle system example



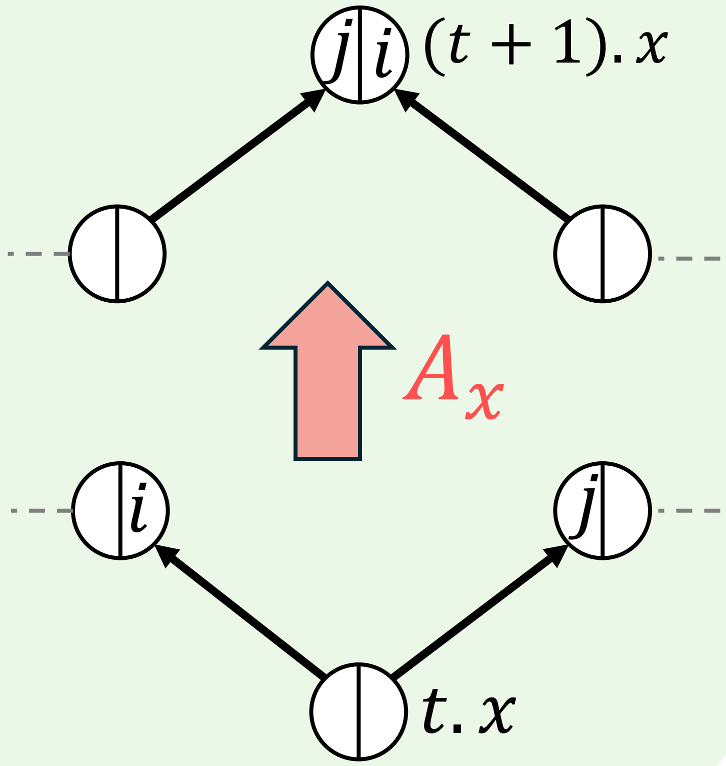
Local rule



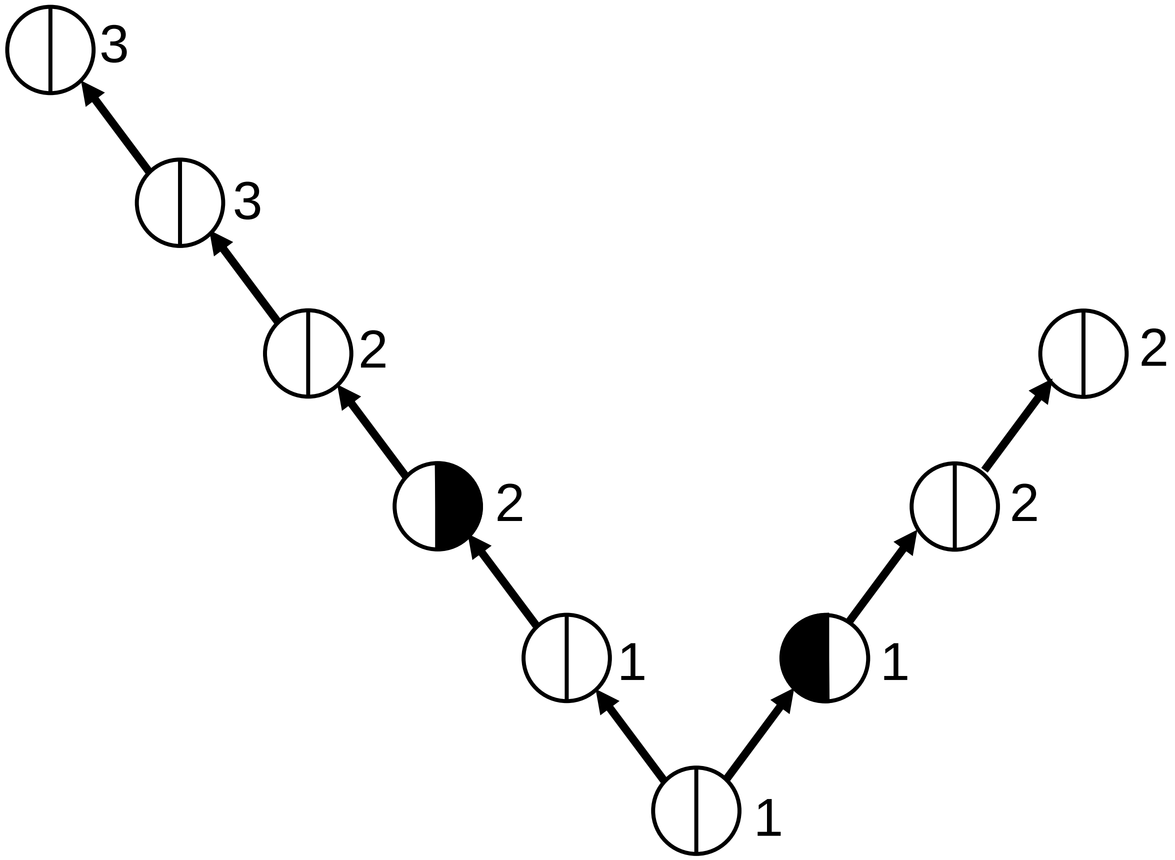
Particle system example



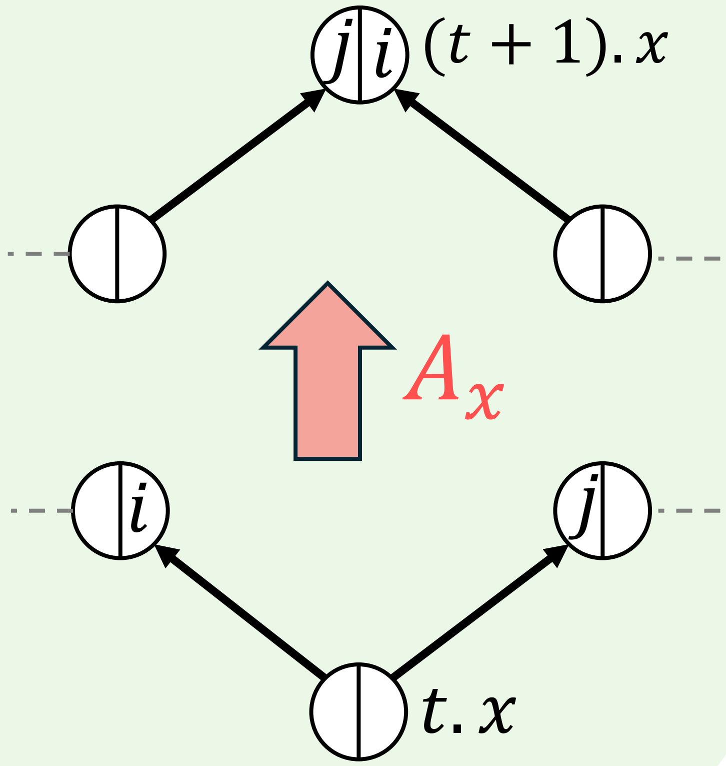
Local rule



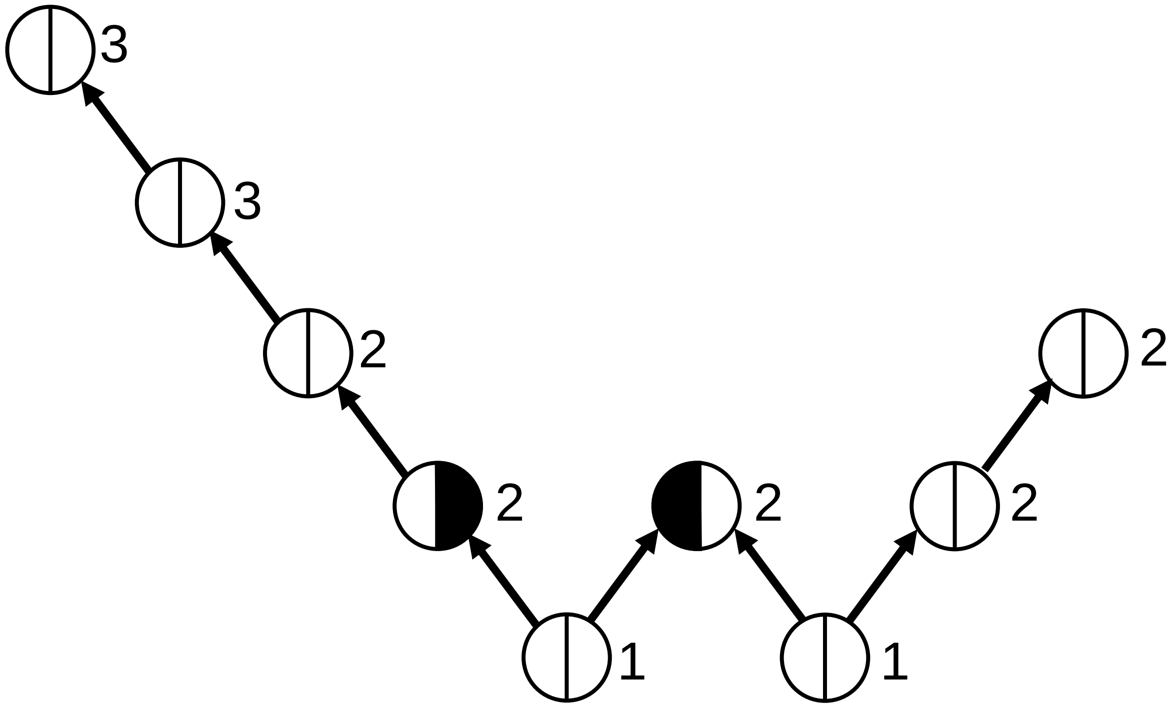
Particle system example

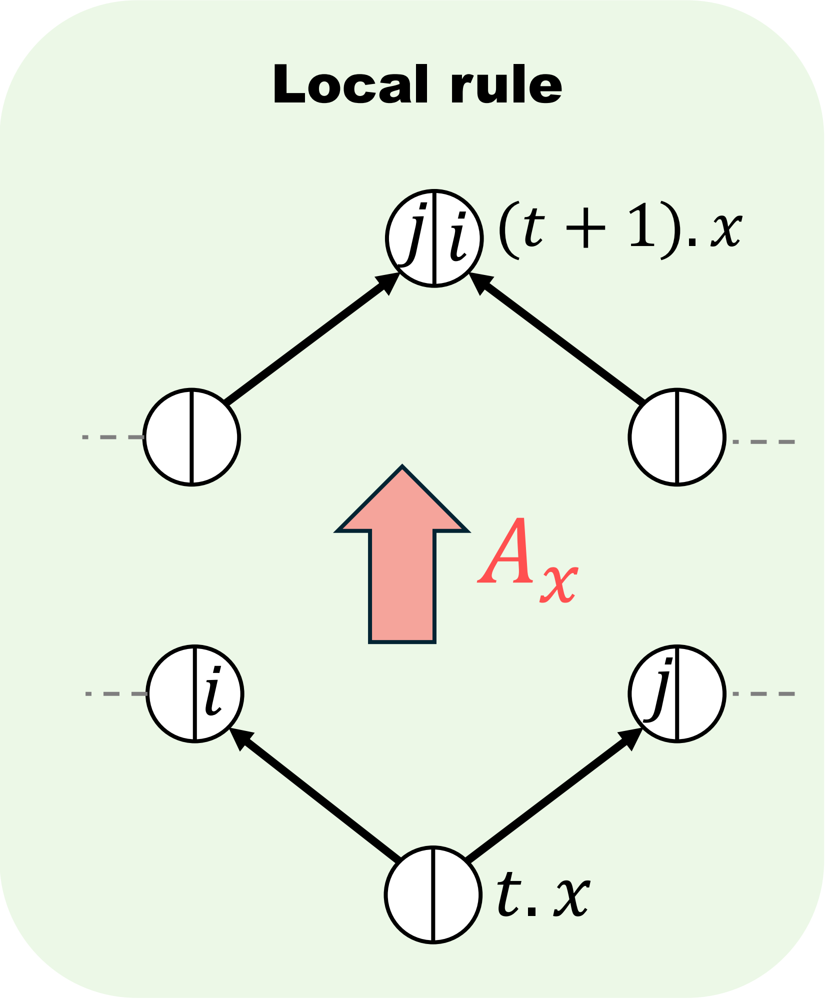


Local rule

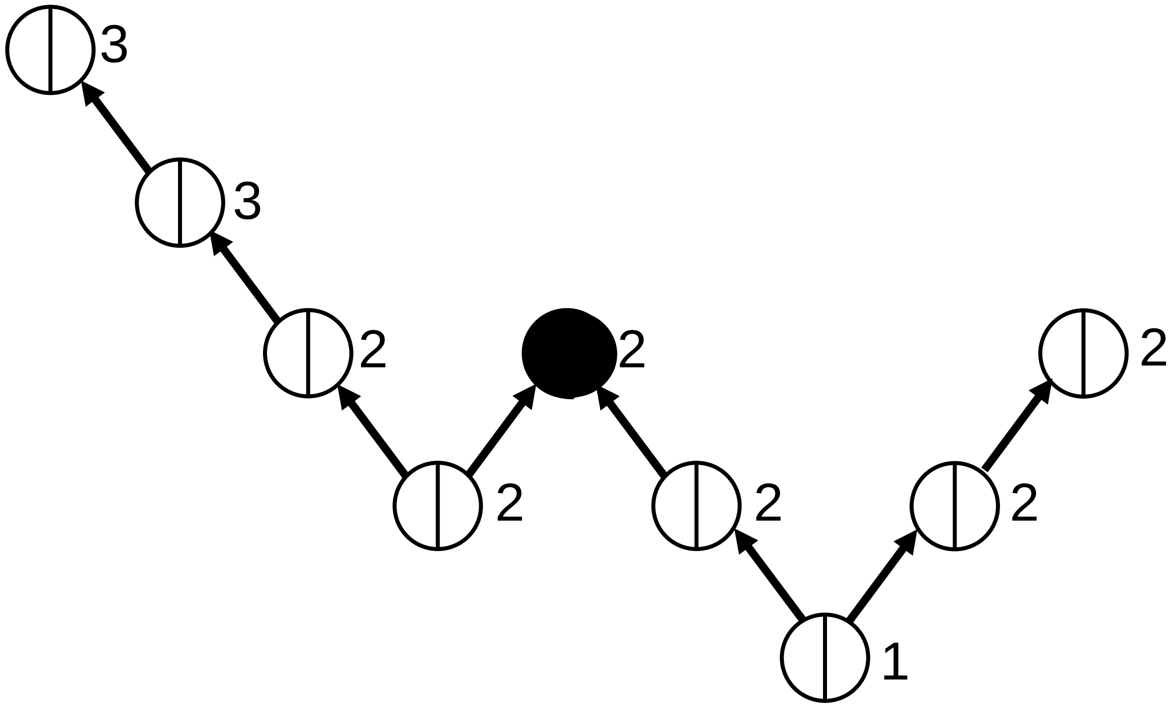


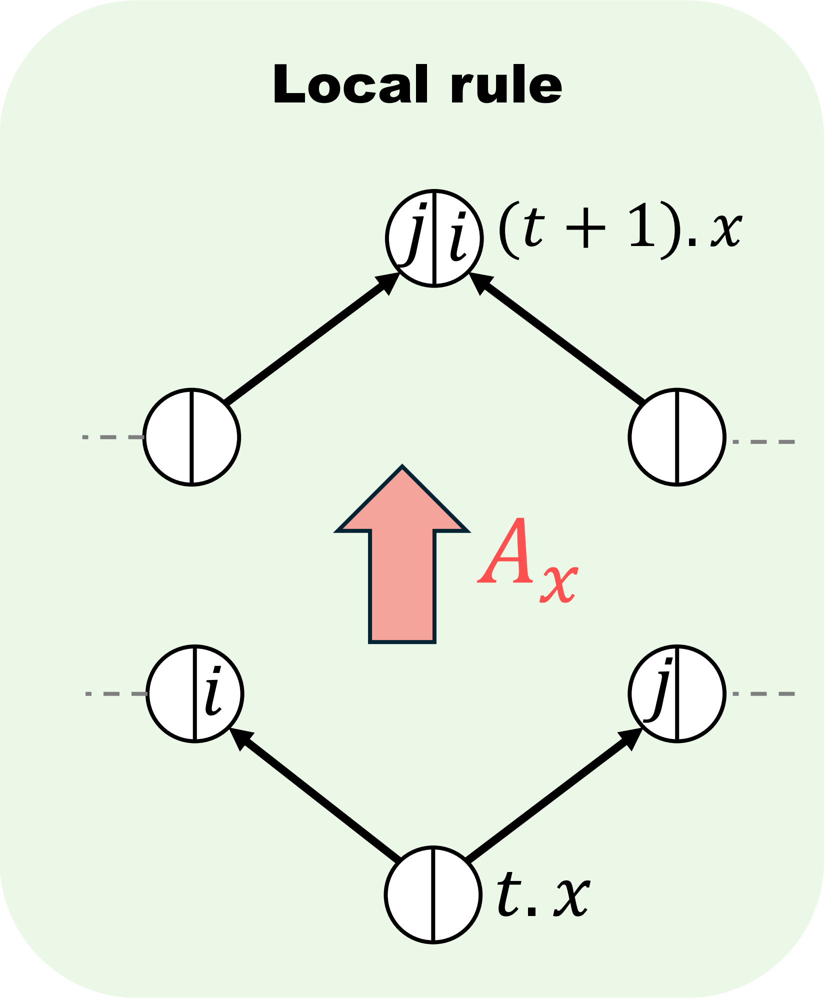
Particle system example



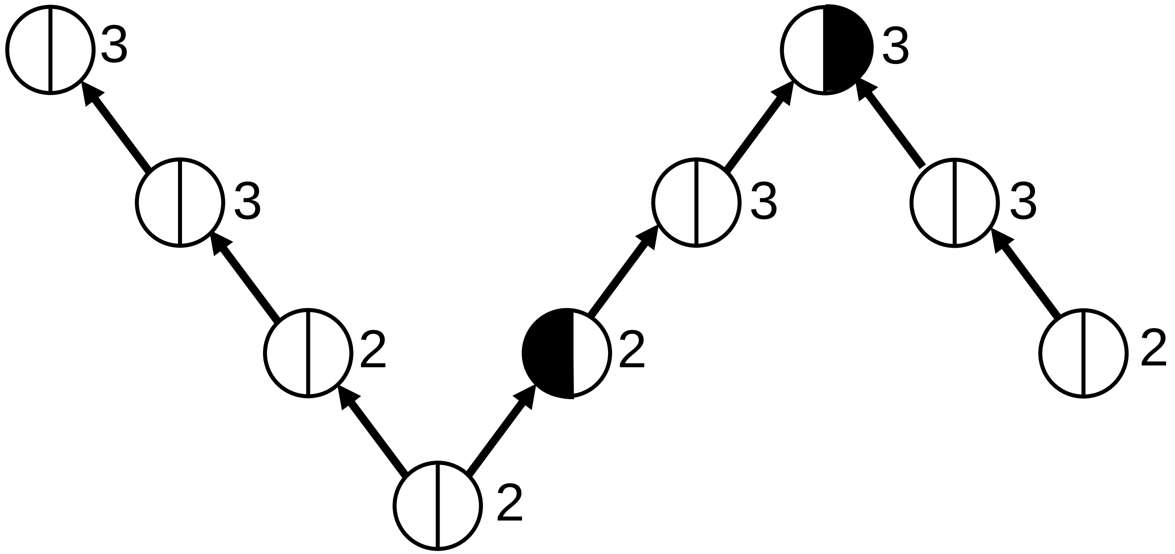


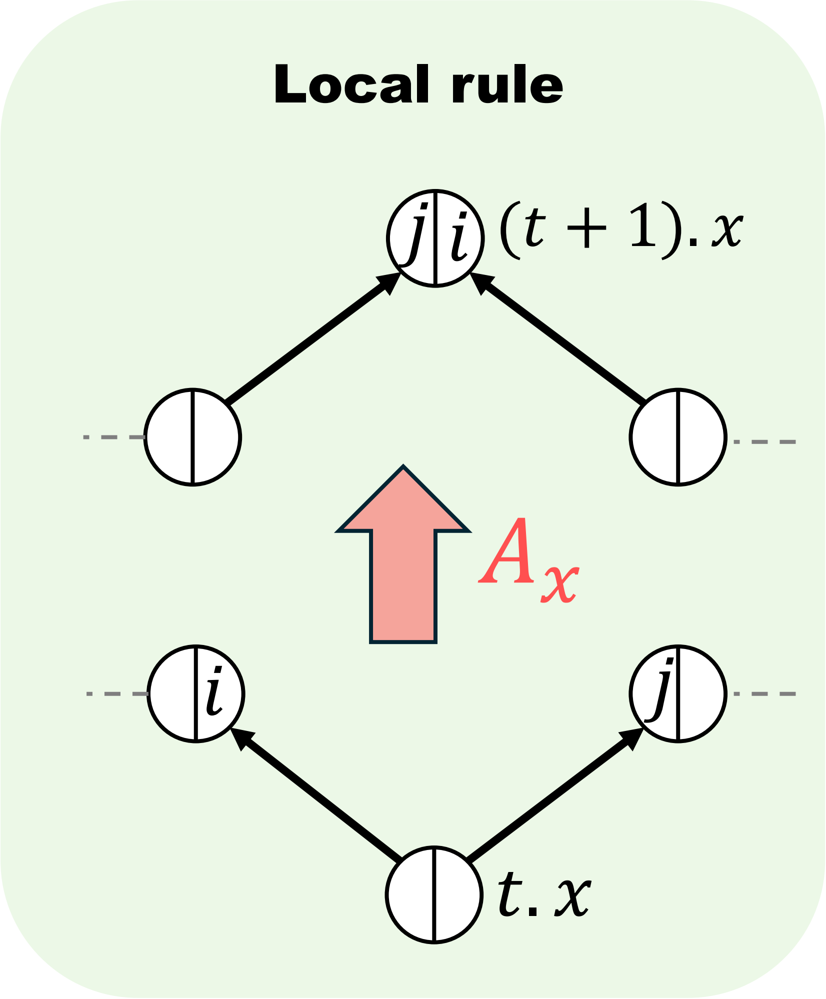
Particle system example



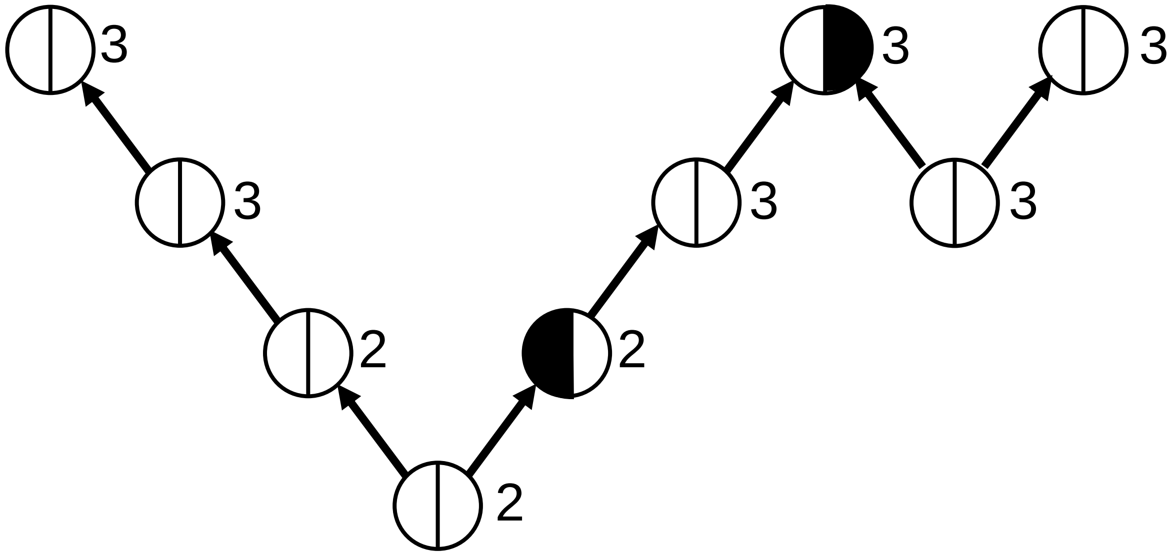


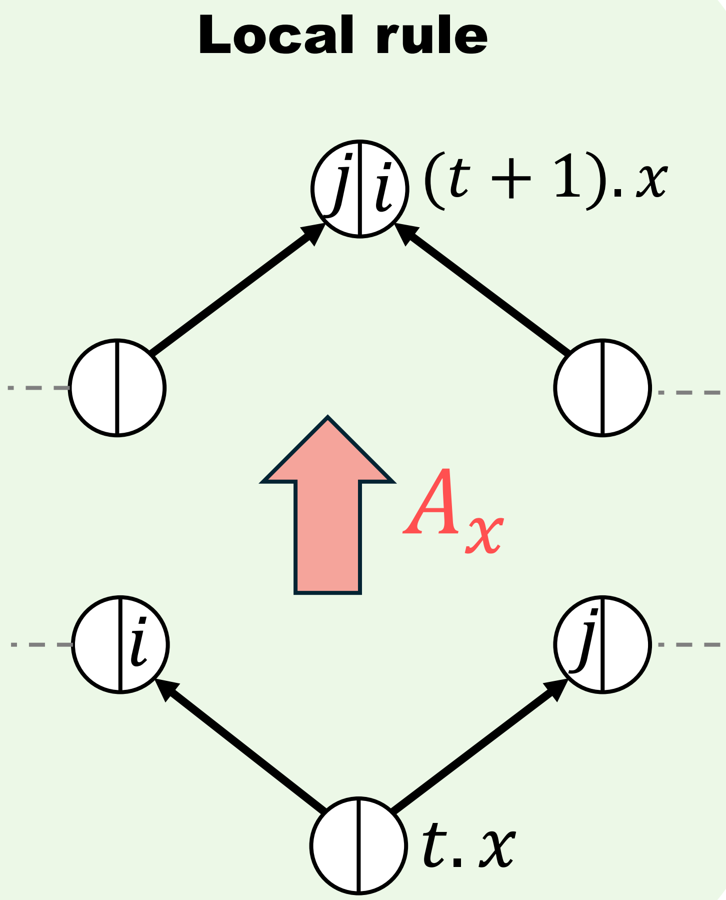
Particle system example



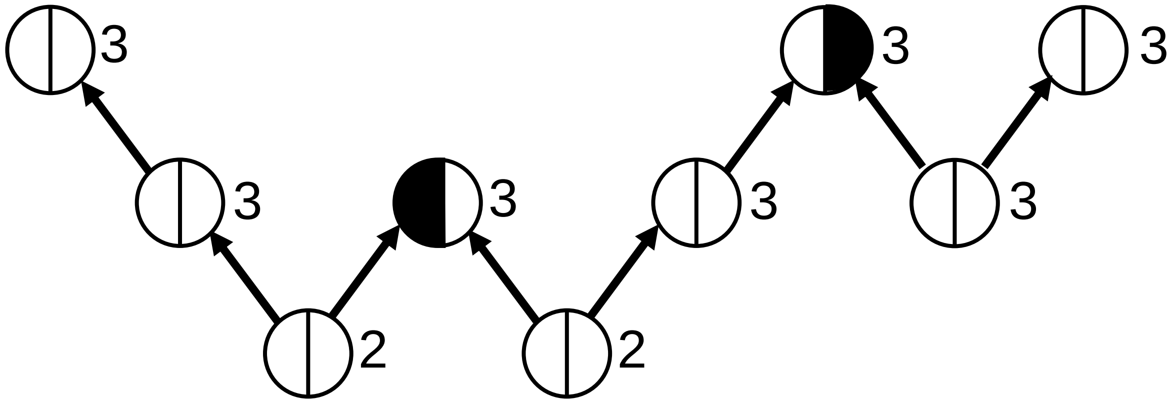


Particle system example

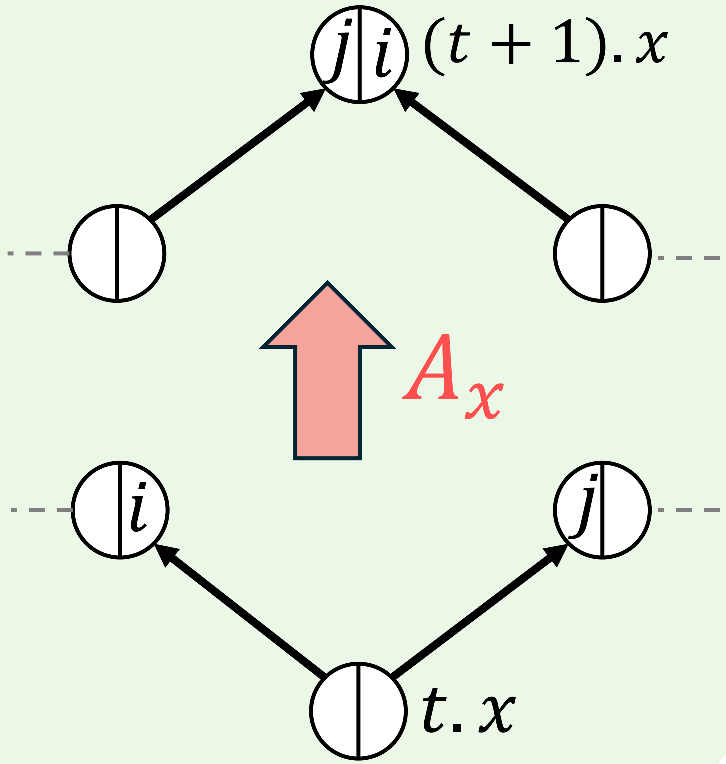




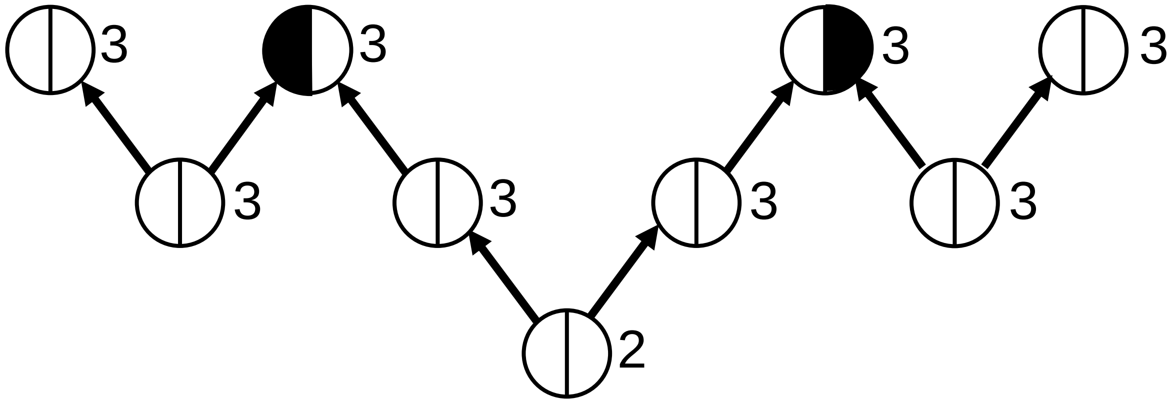
Particle system example

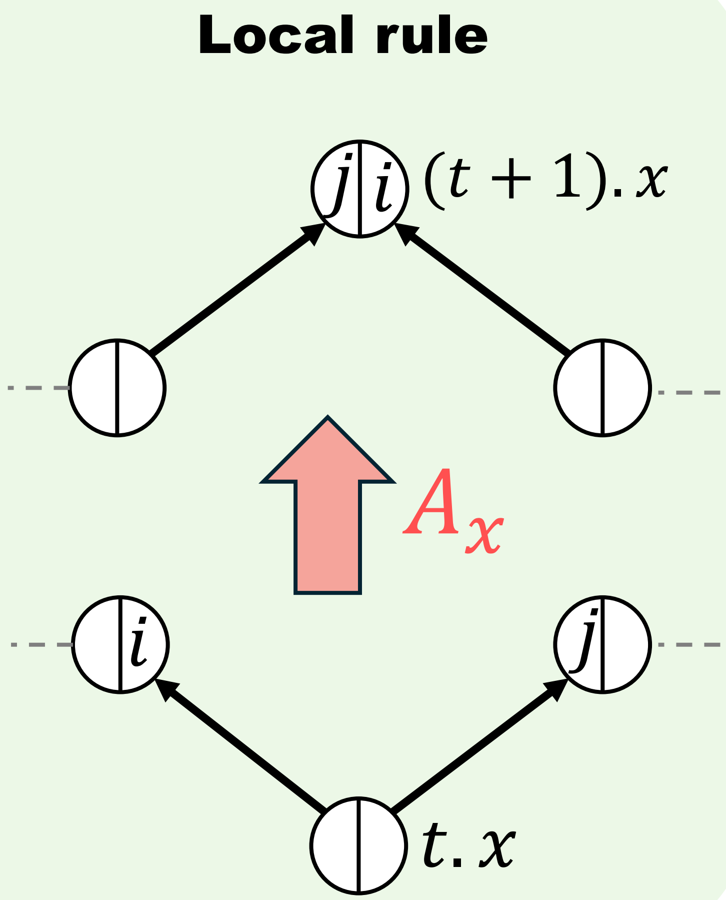


Local rule

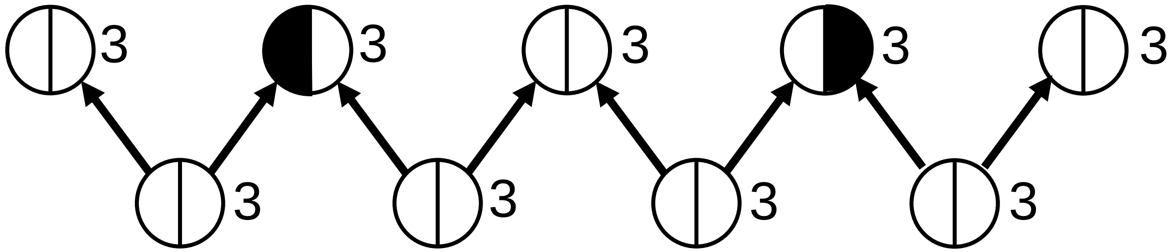


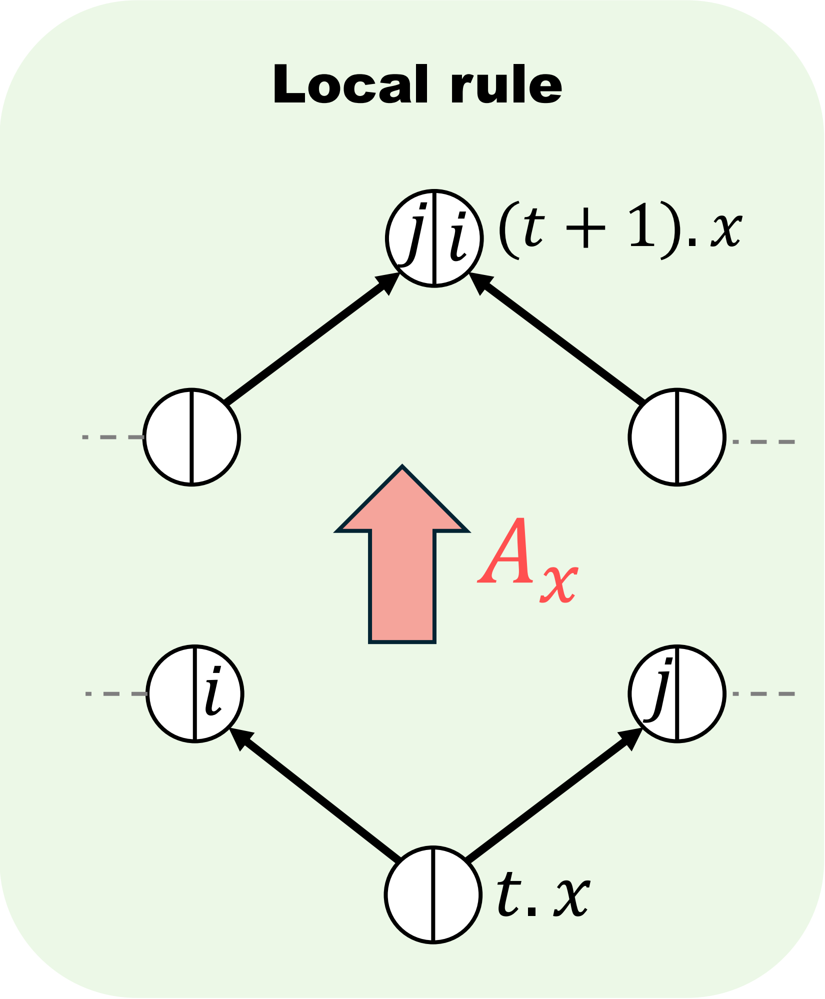
Particle system example



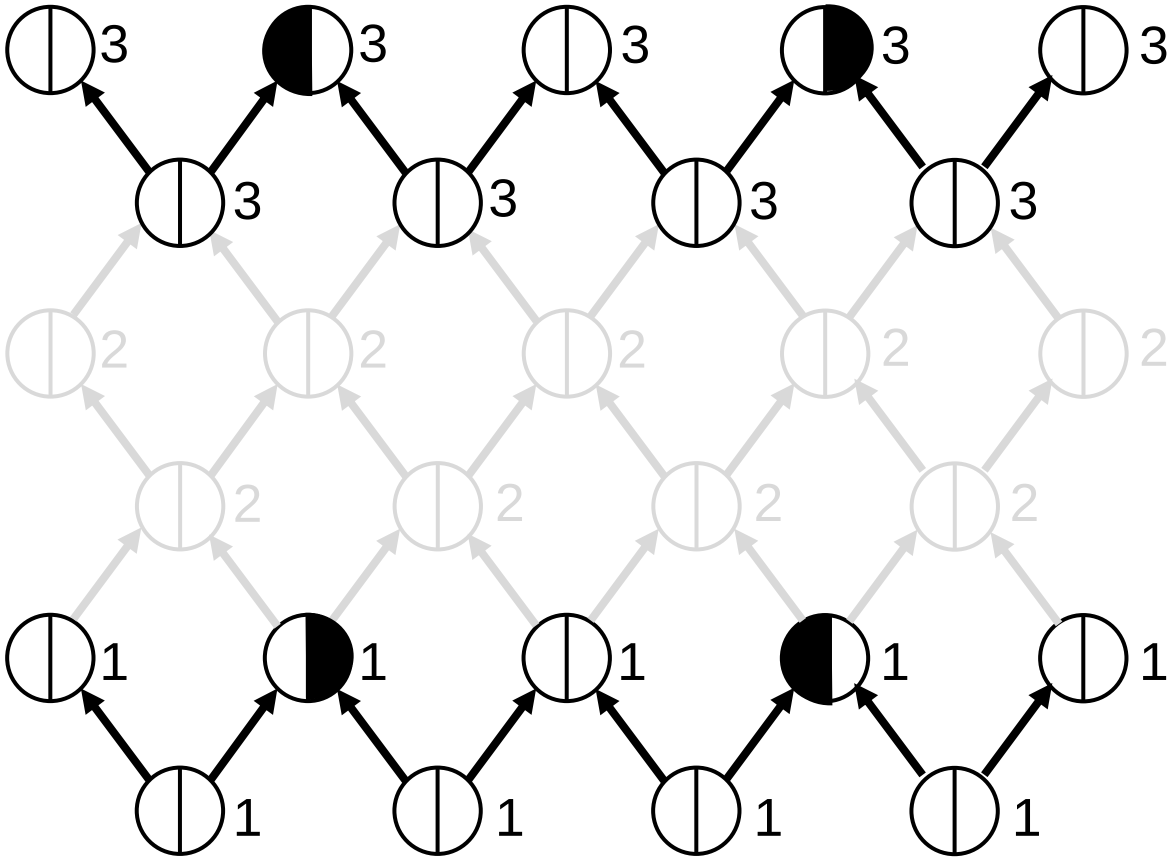


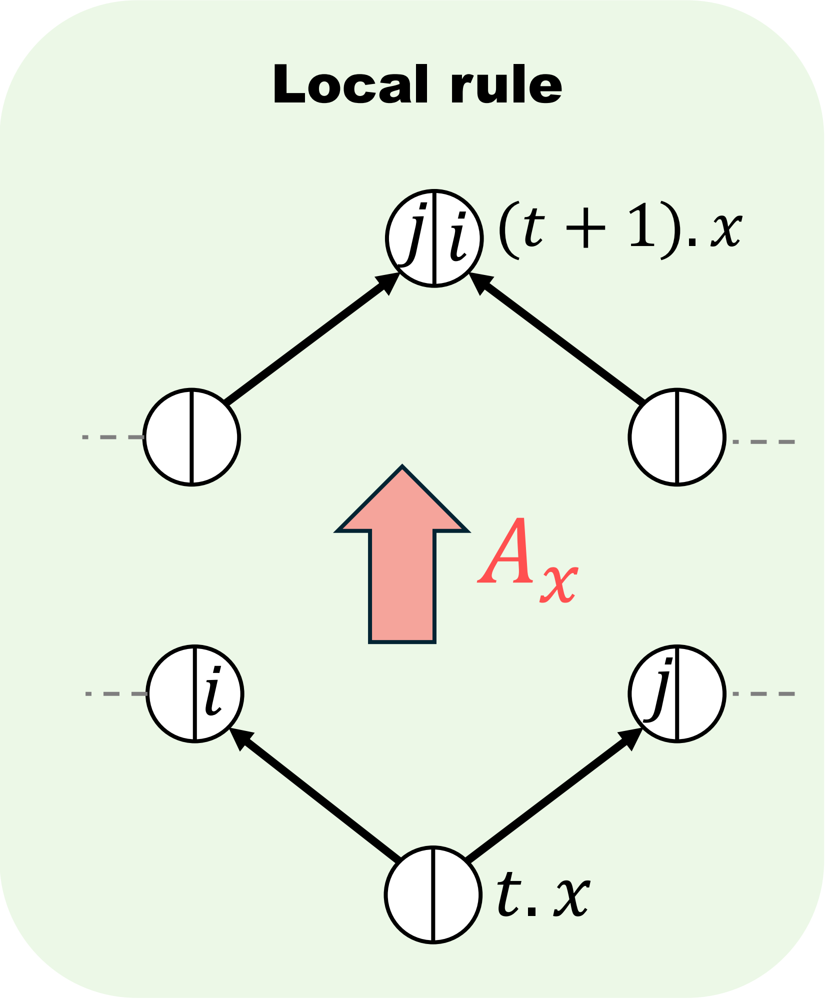
Particle system example



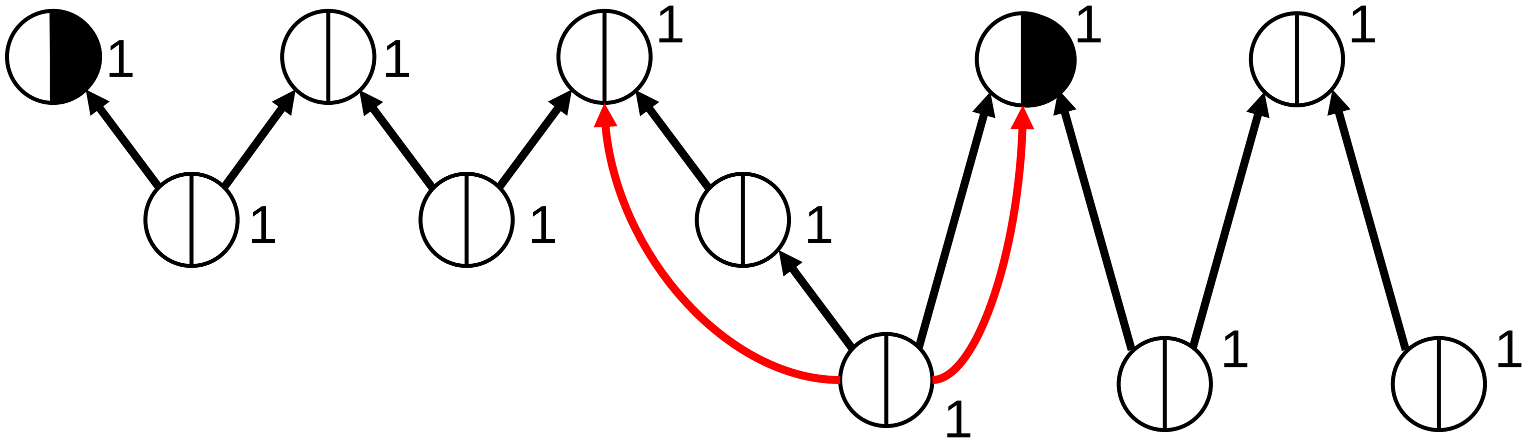


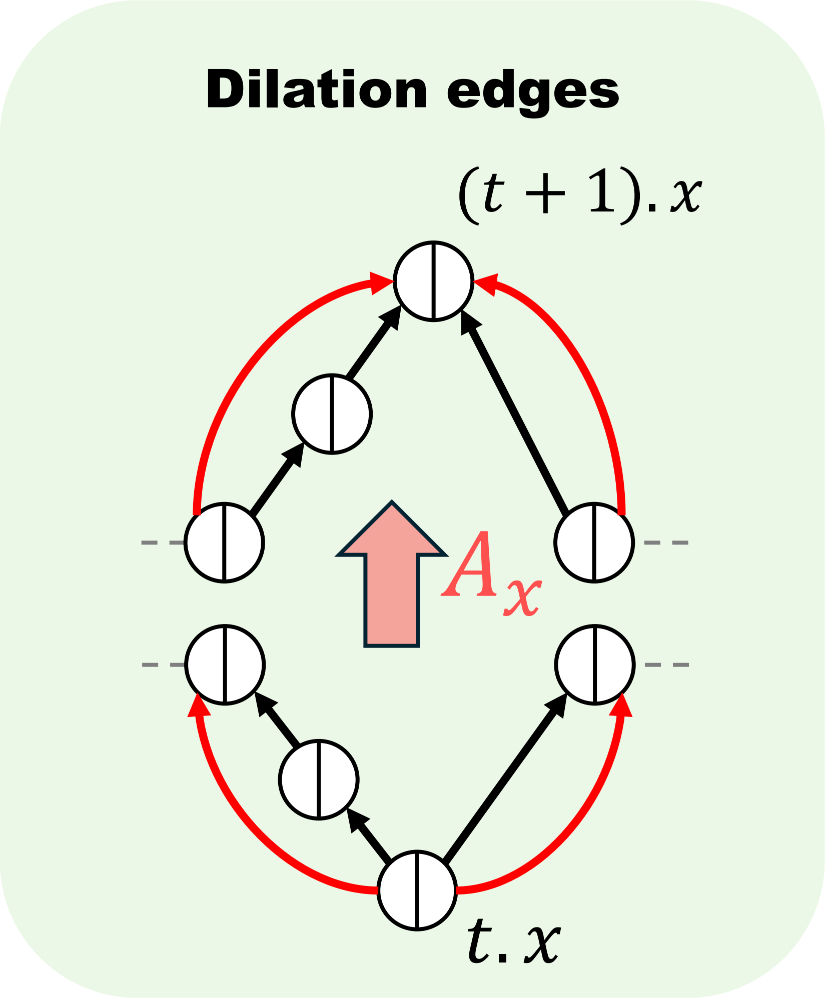
Particle system example



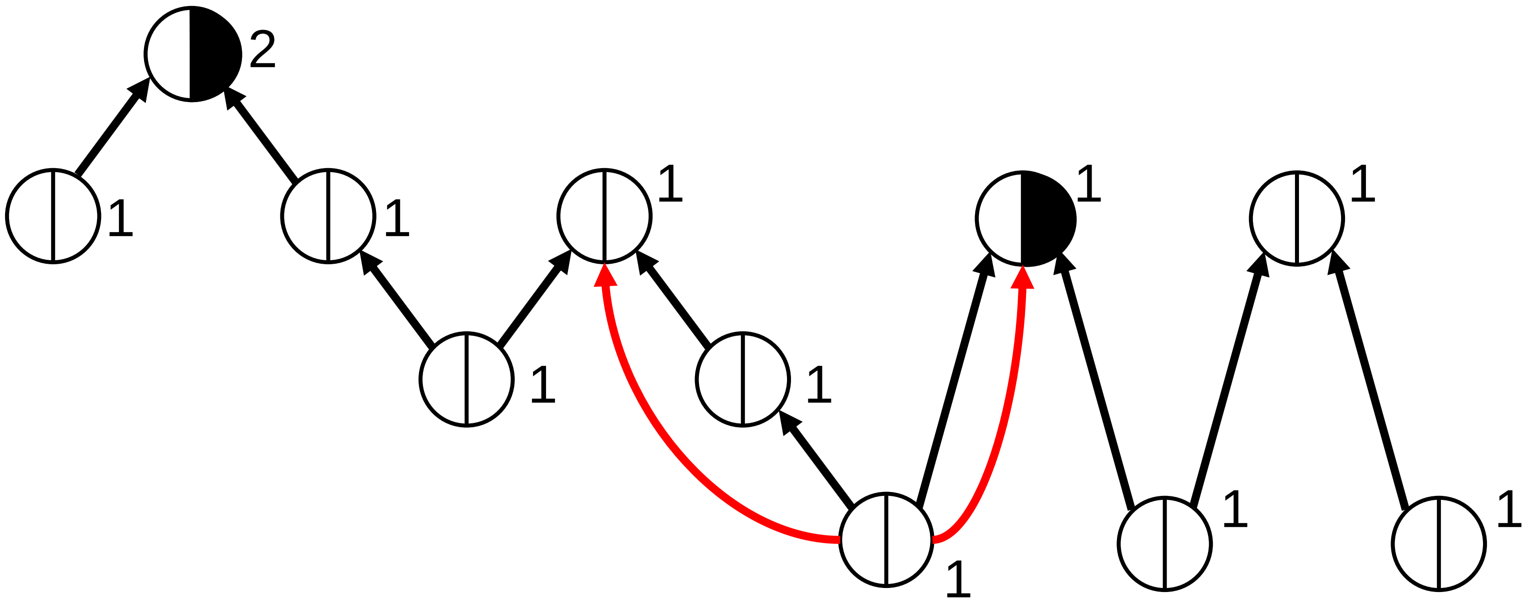


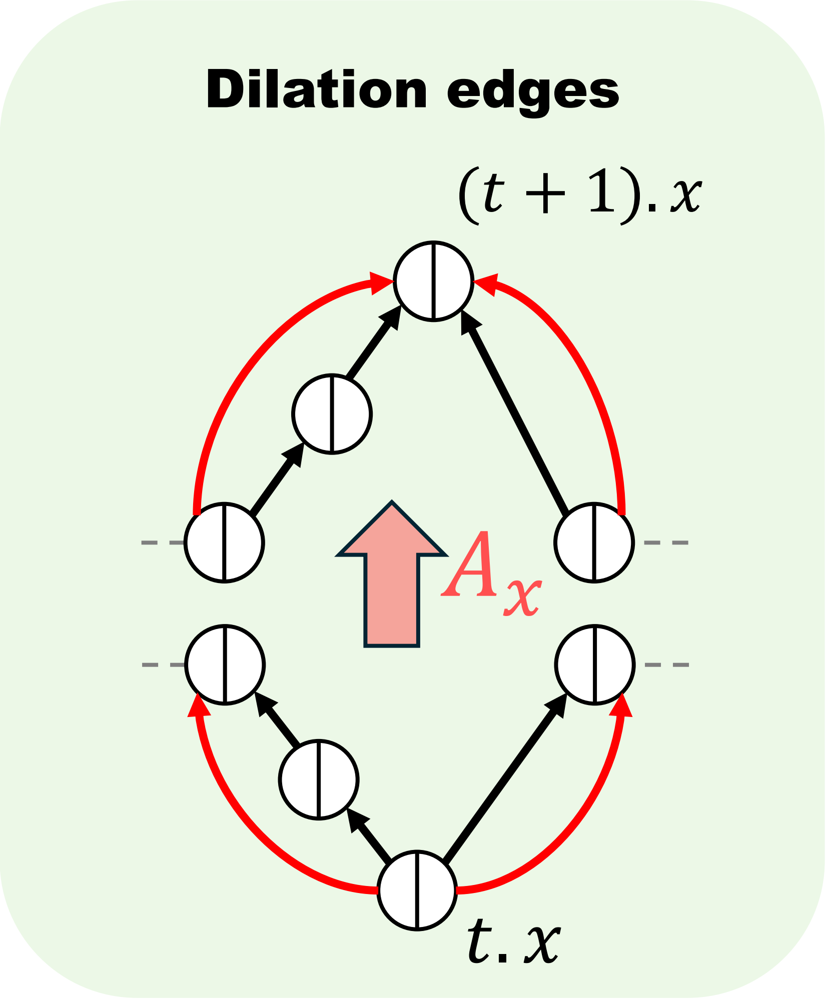
Time dilation example



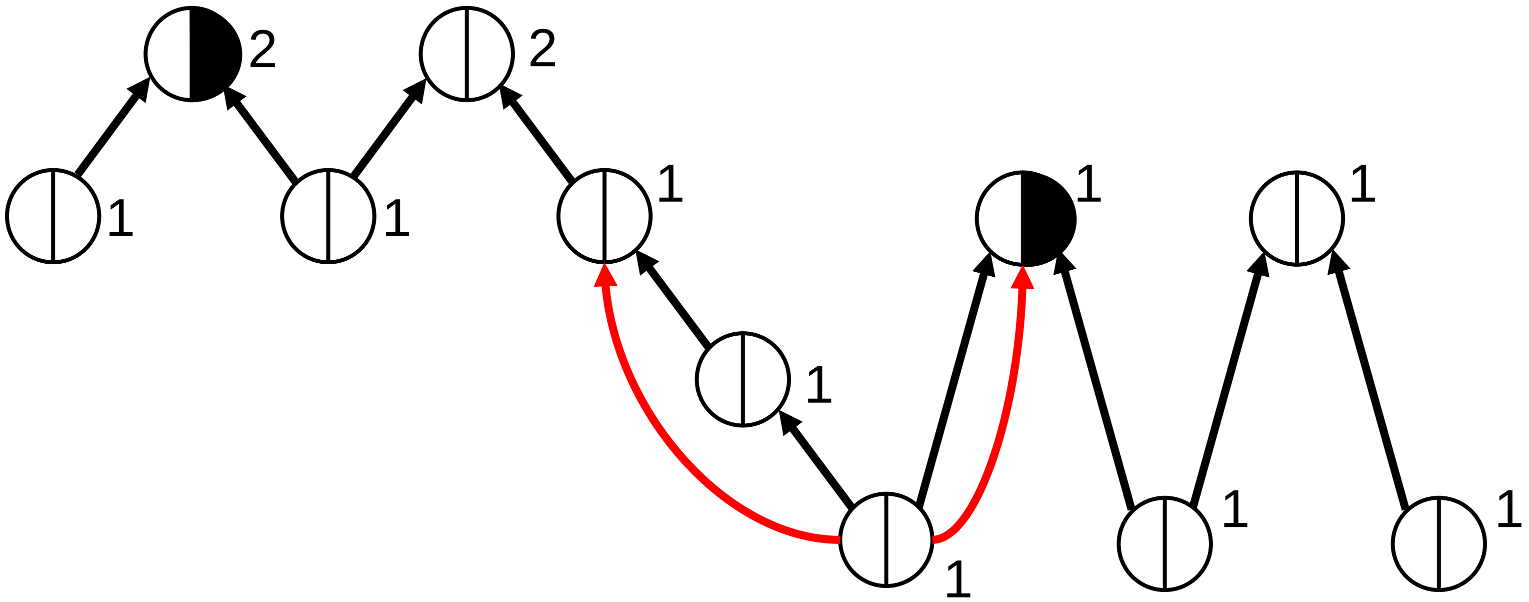


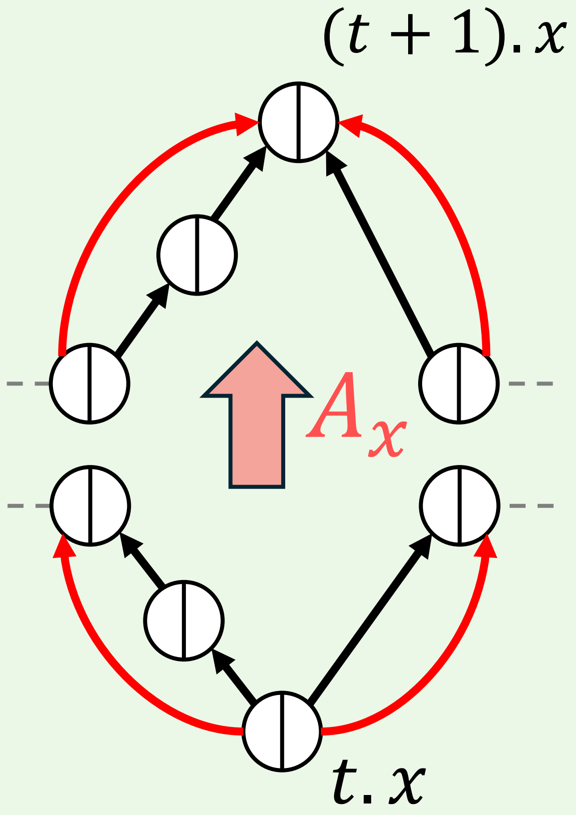
Time dilation example



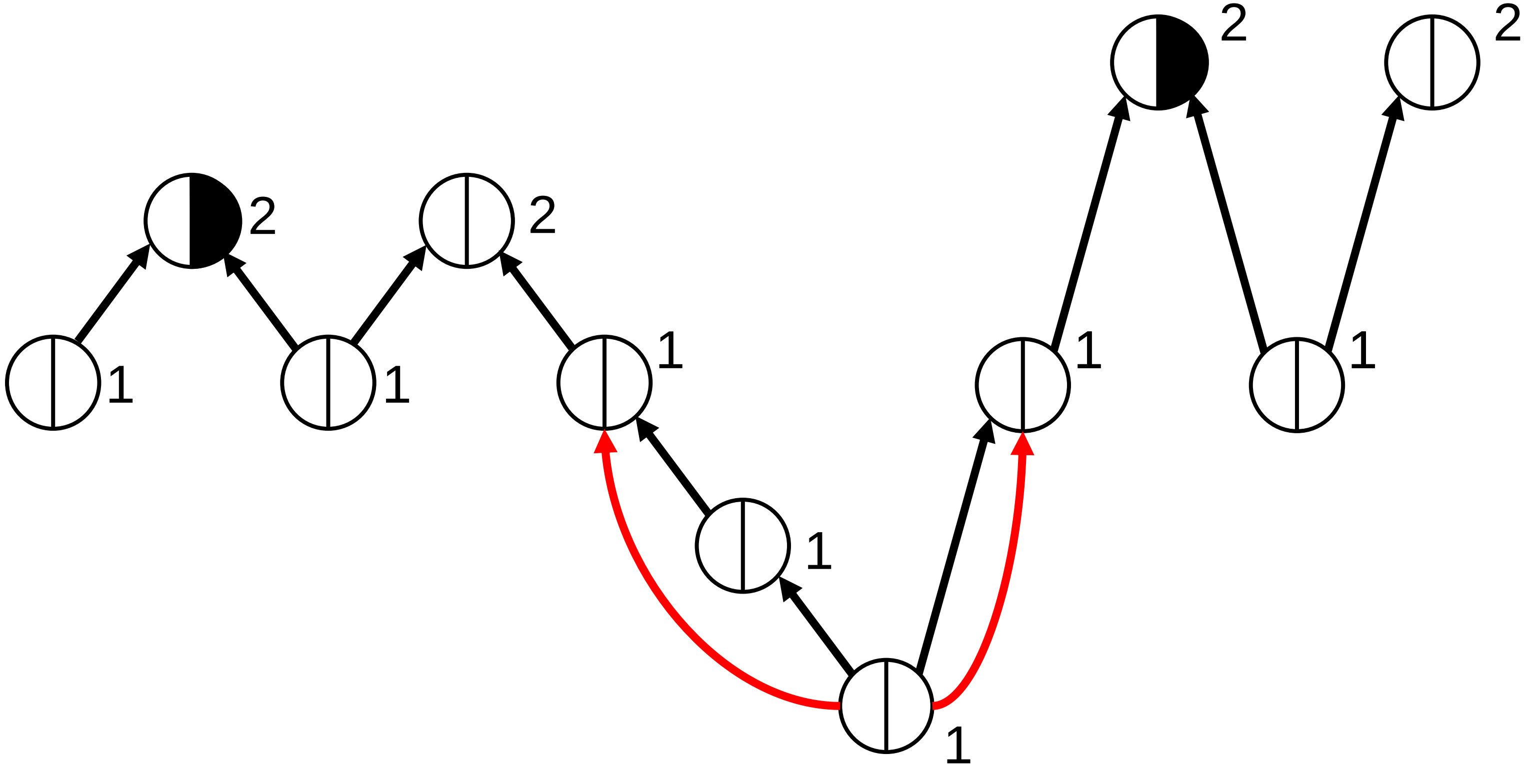


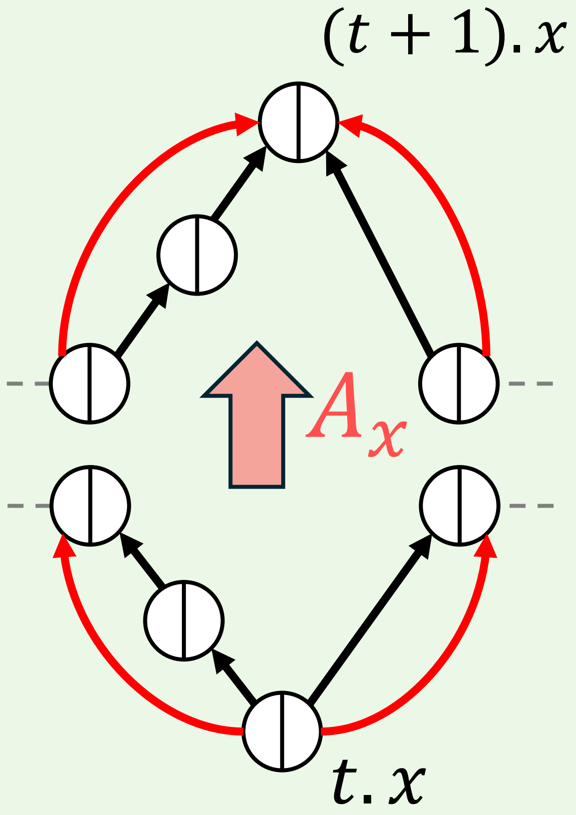
Time dilation example



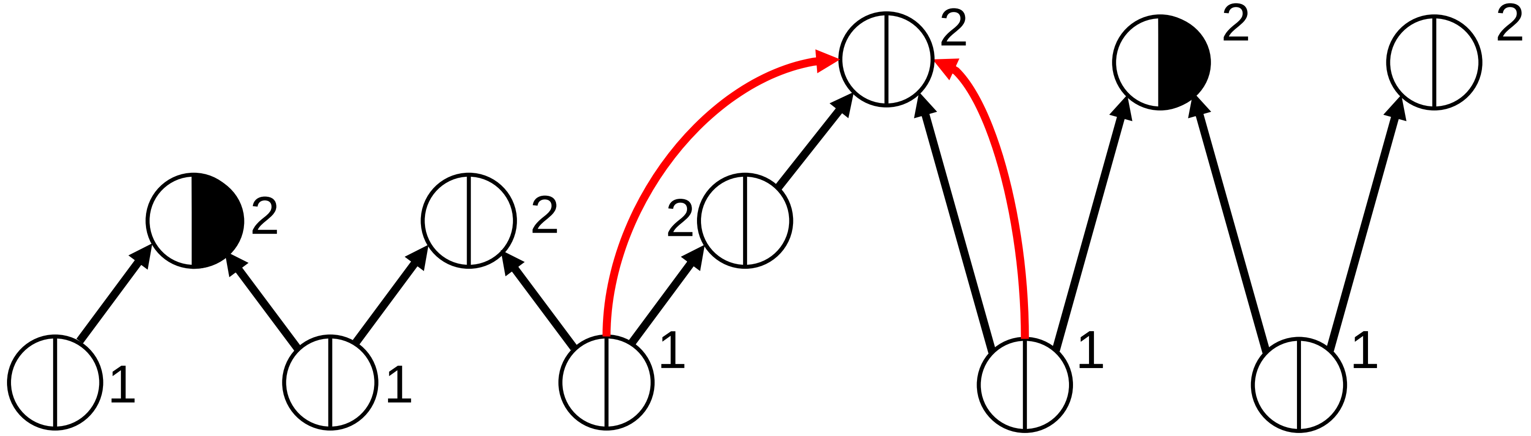


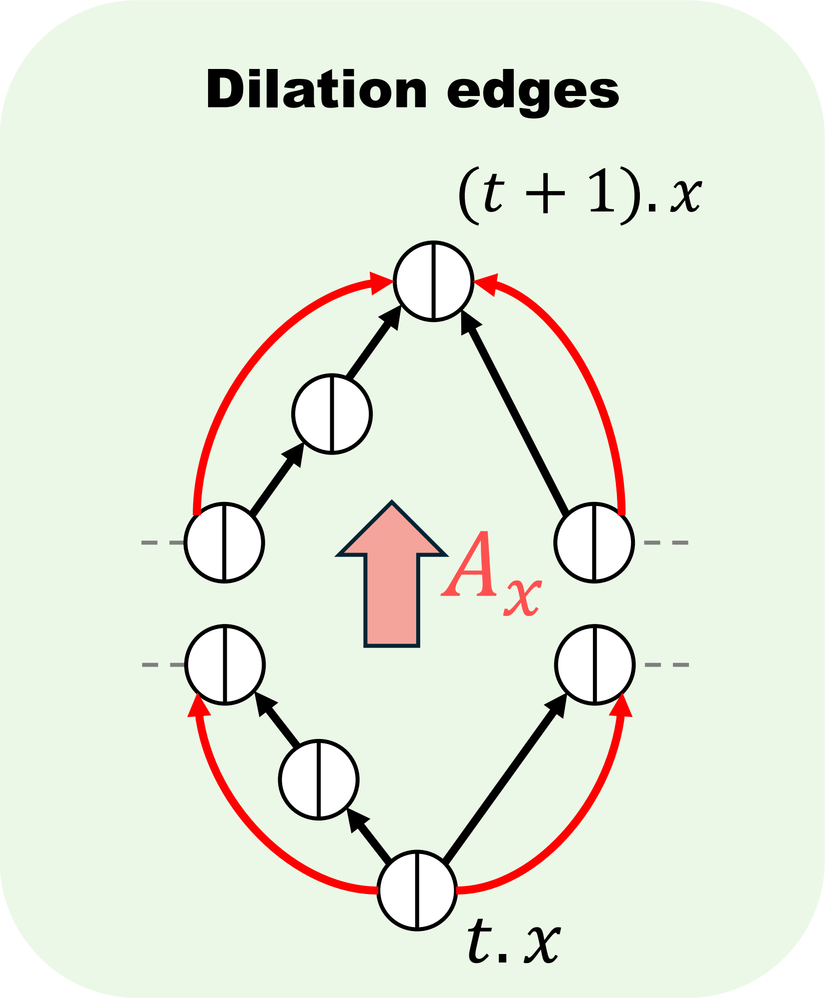
Time dilation example



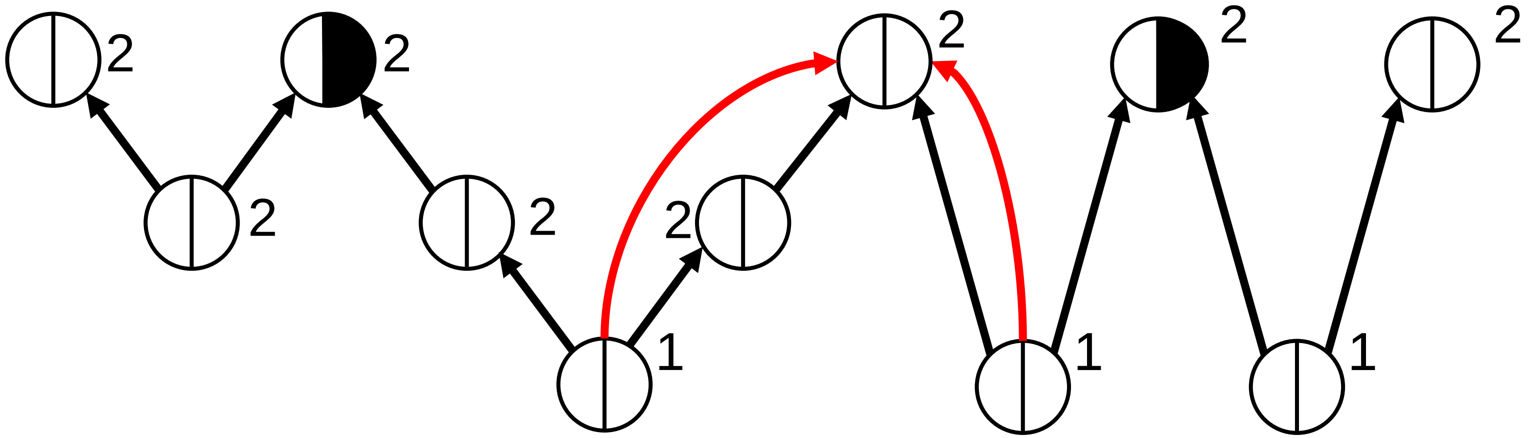


Time dilation example

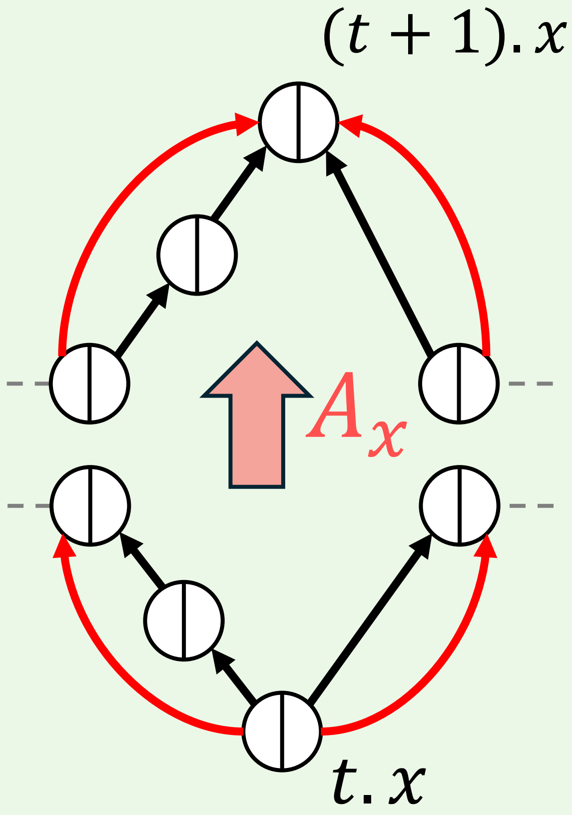




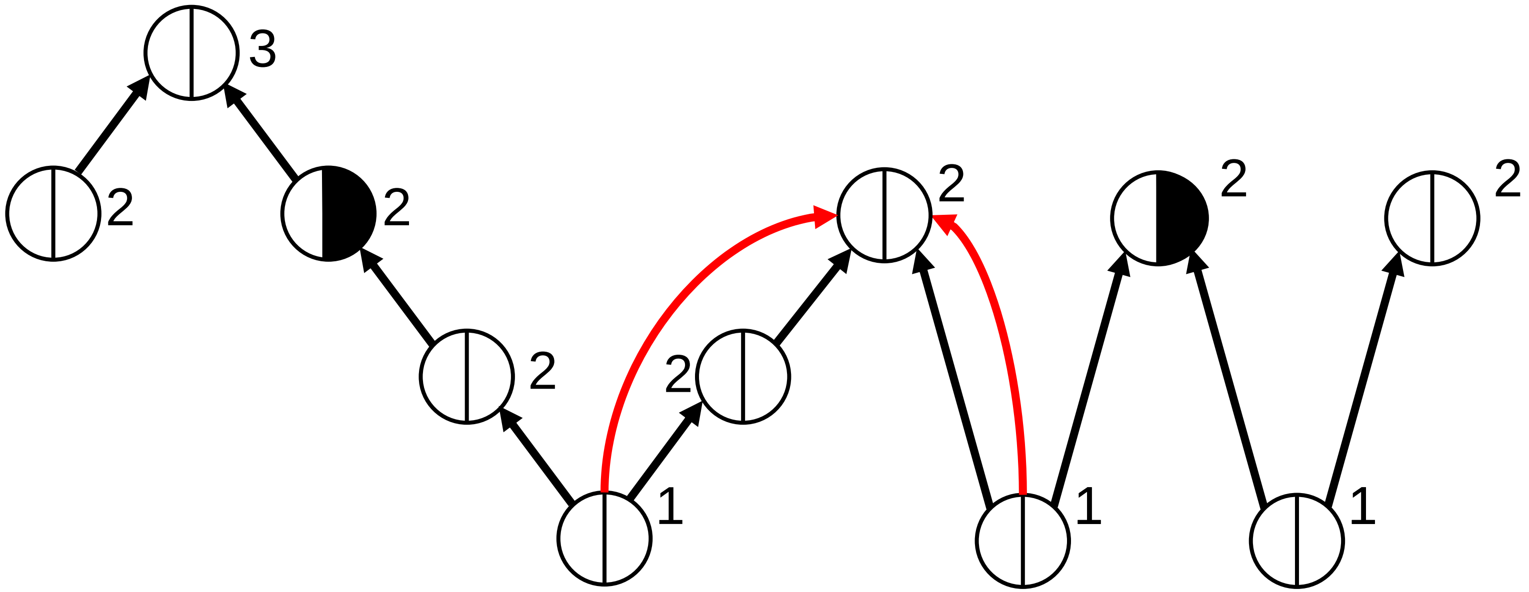
Time dilation example

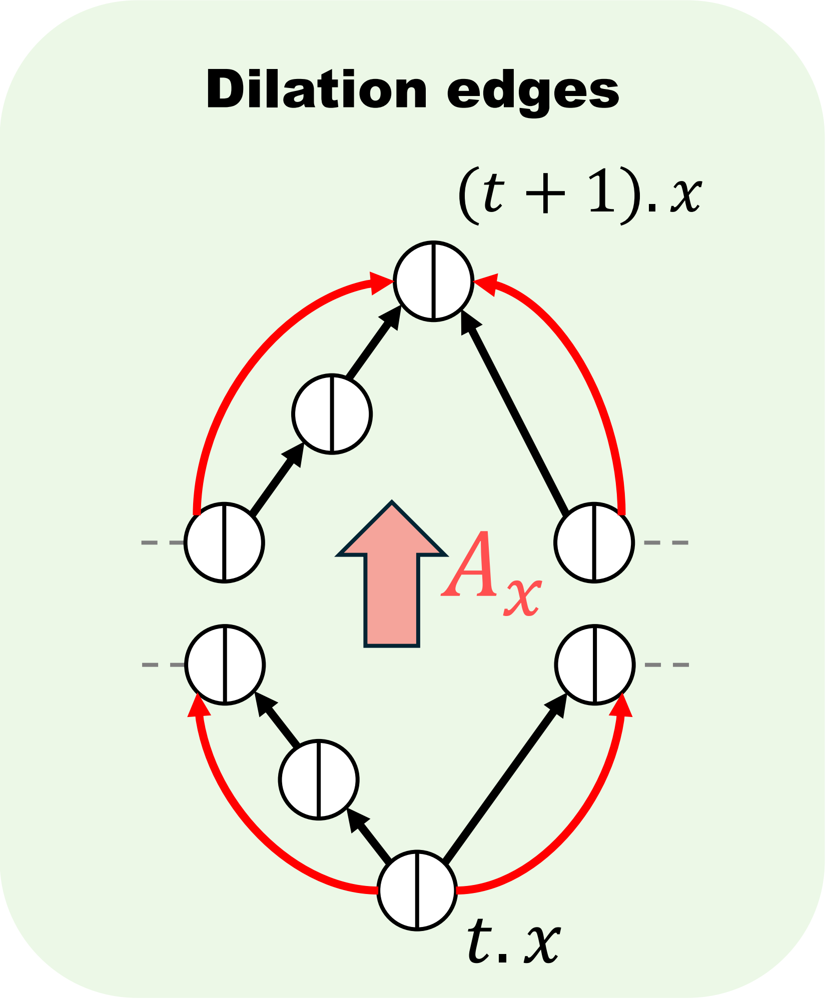


Dilation edges

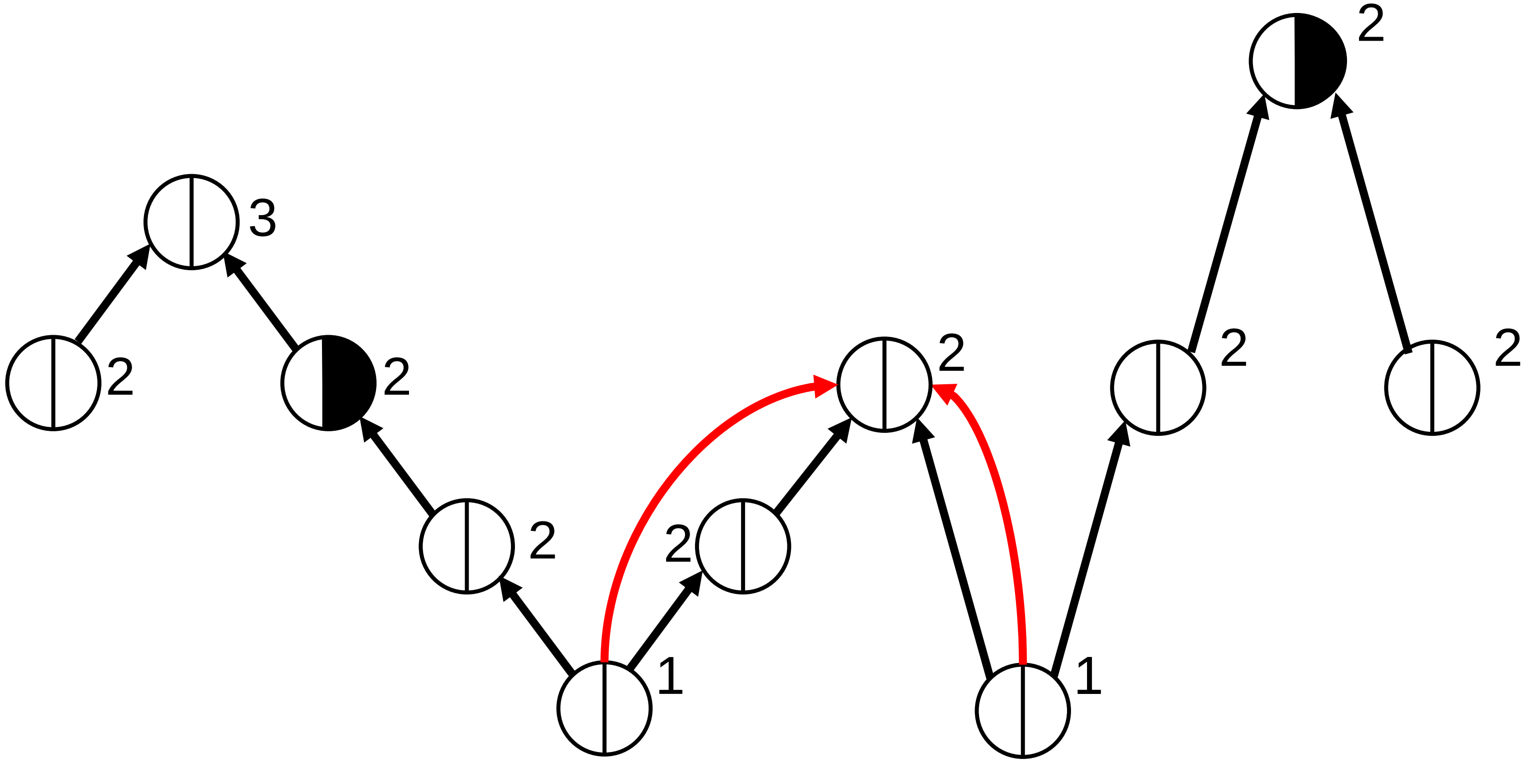


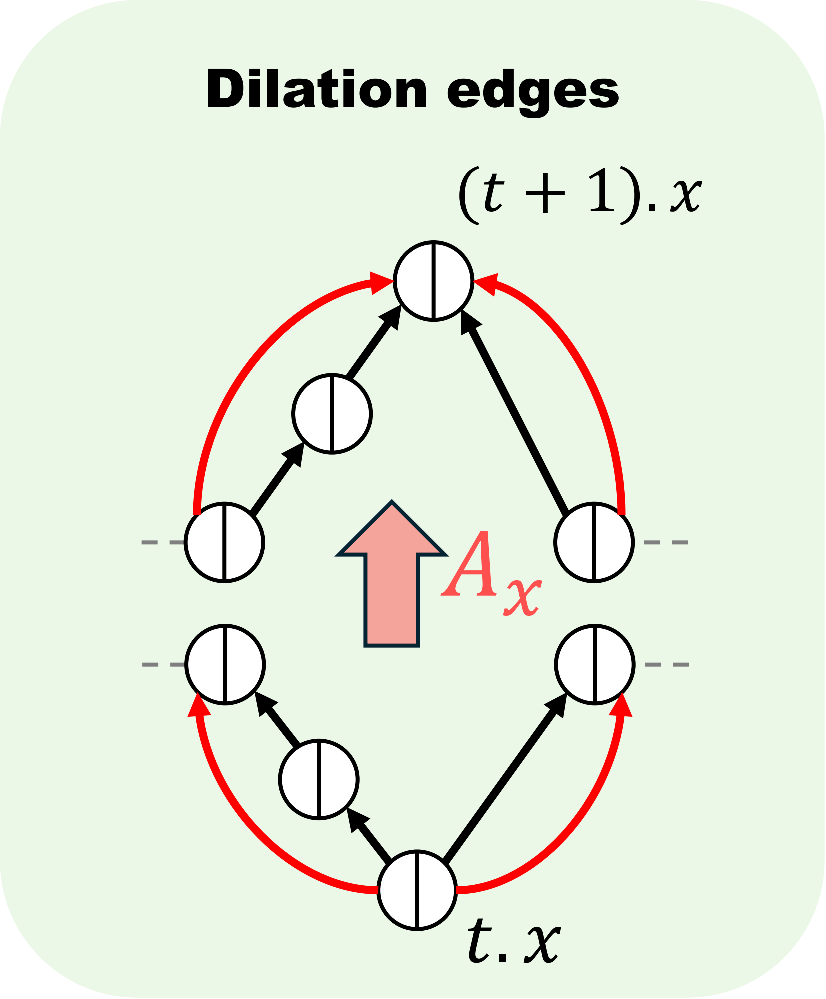
Time dilation example



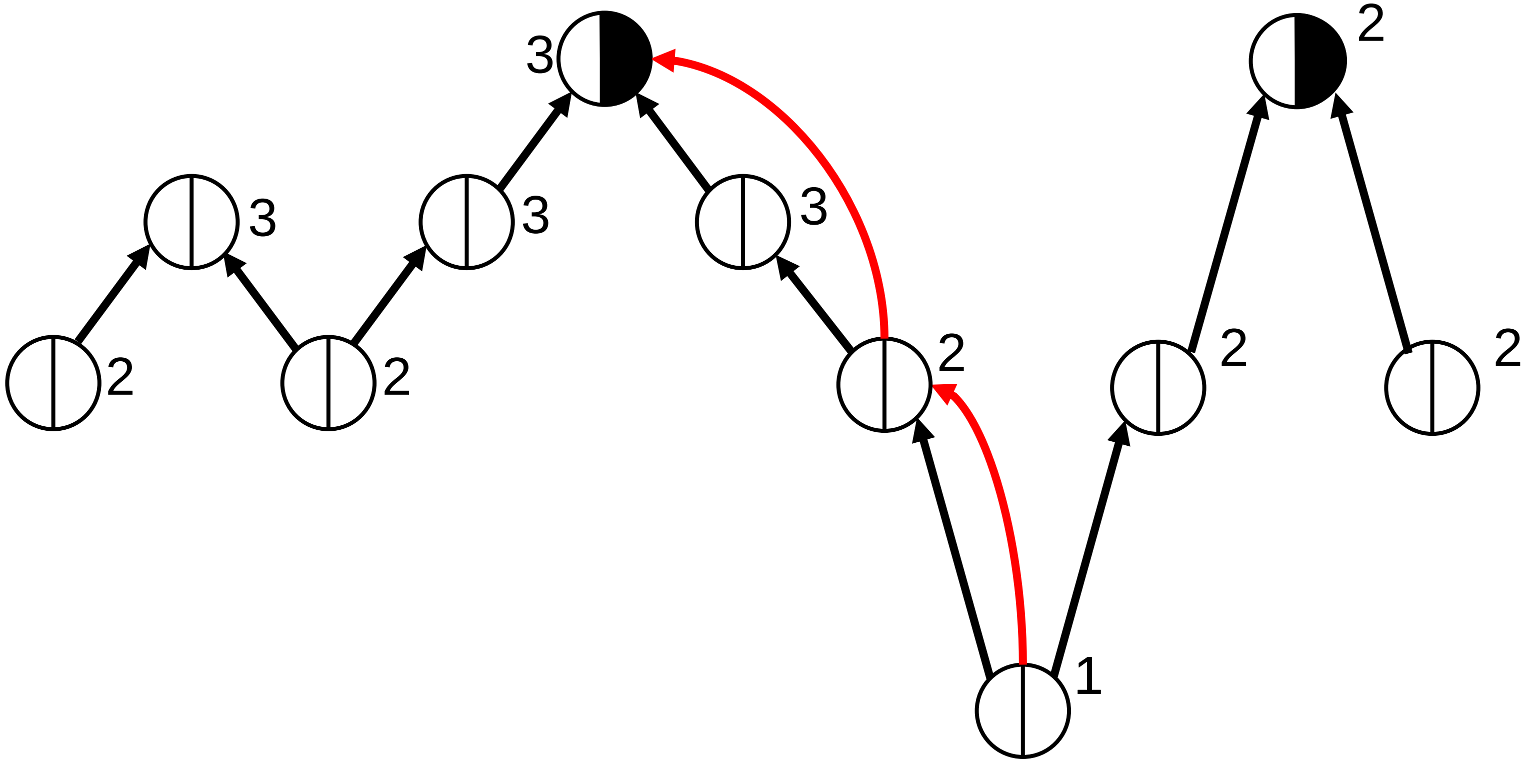


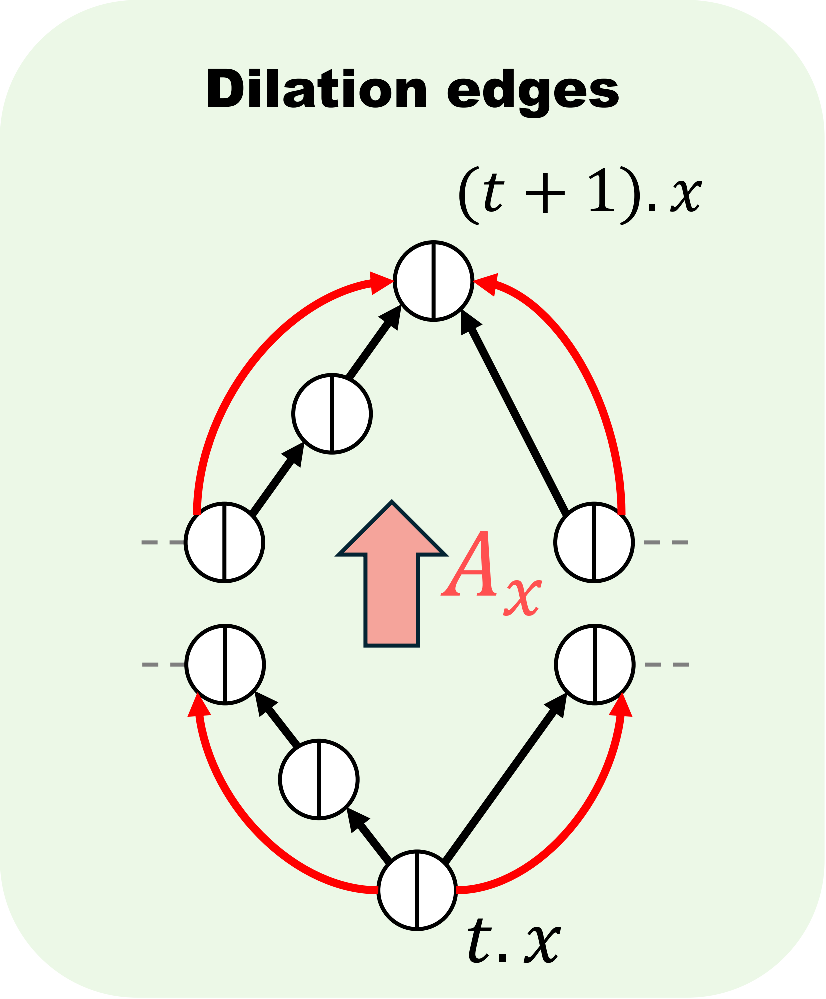
Time dilation example



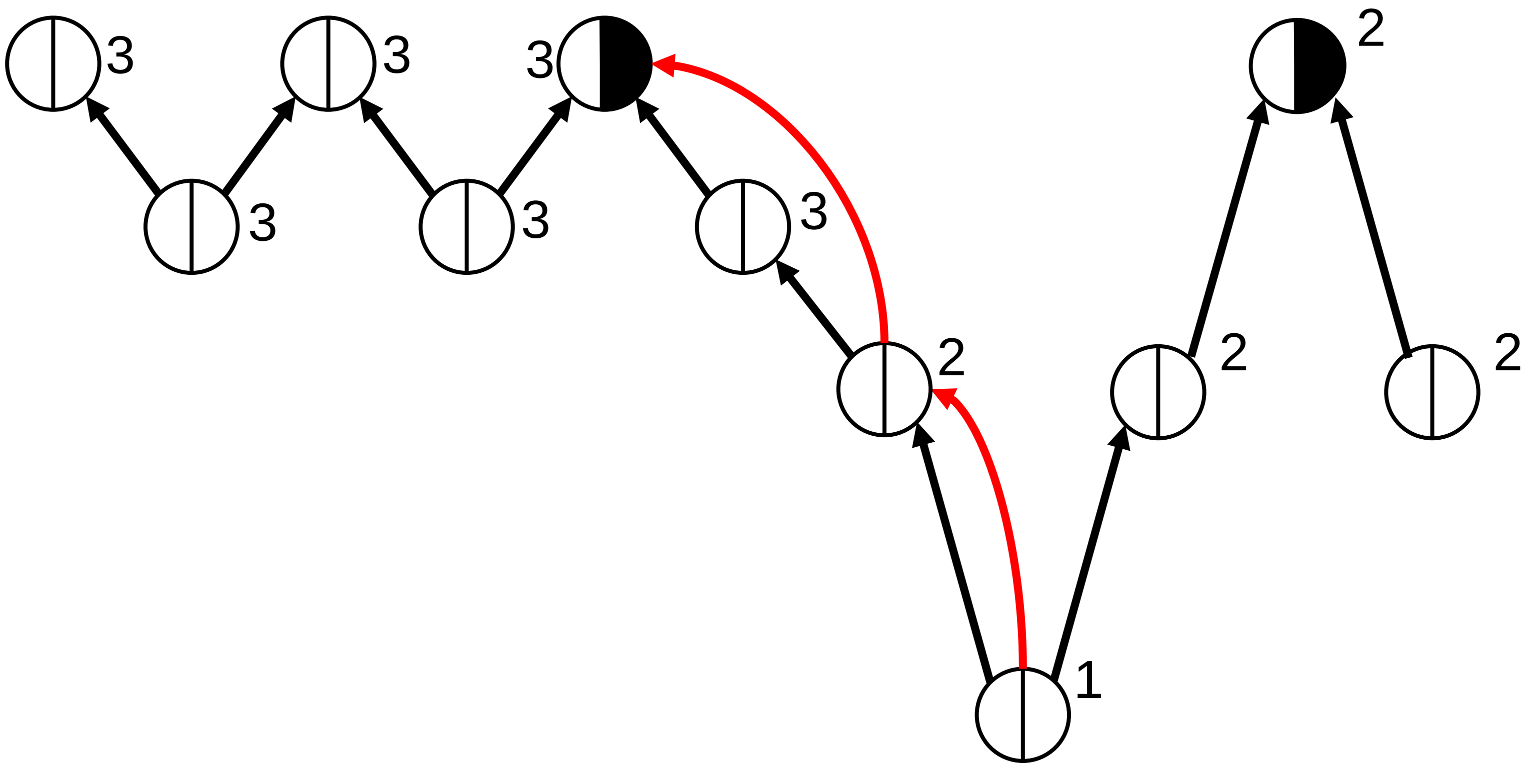


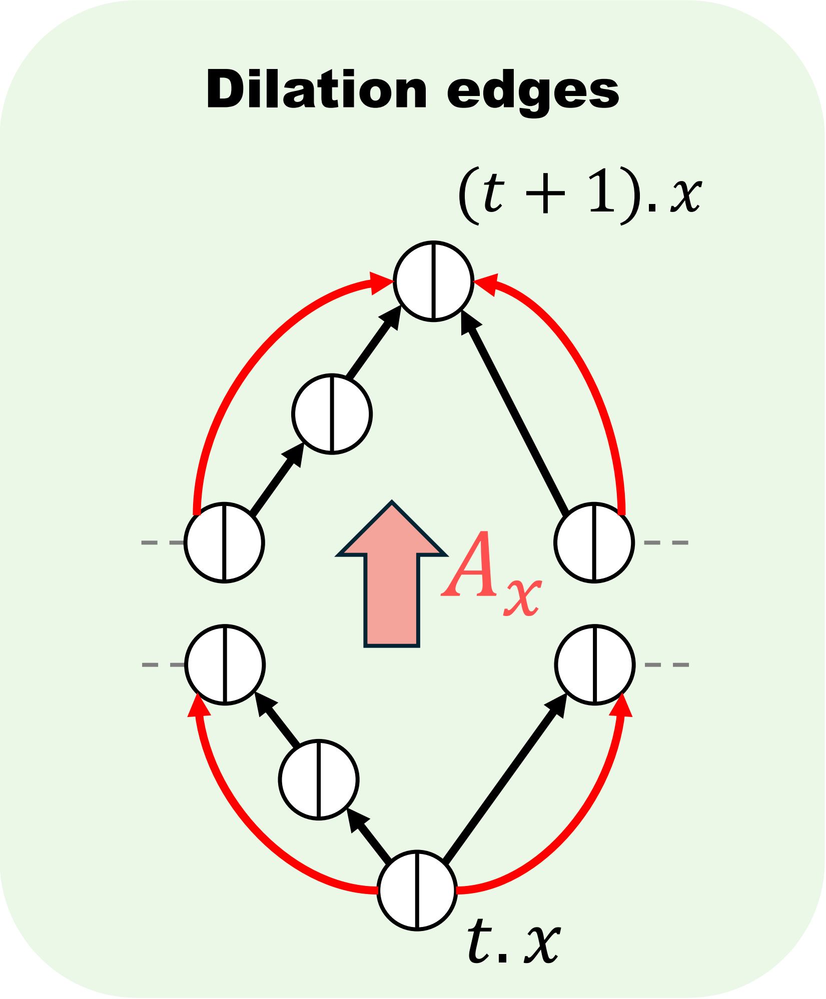
Time dilation example



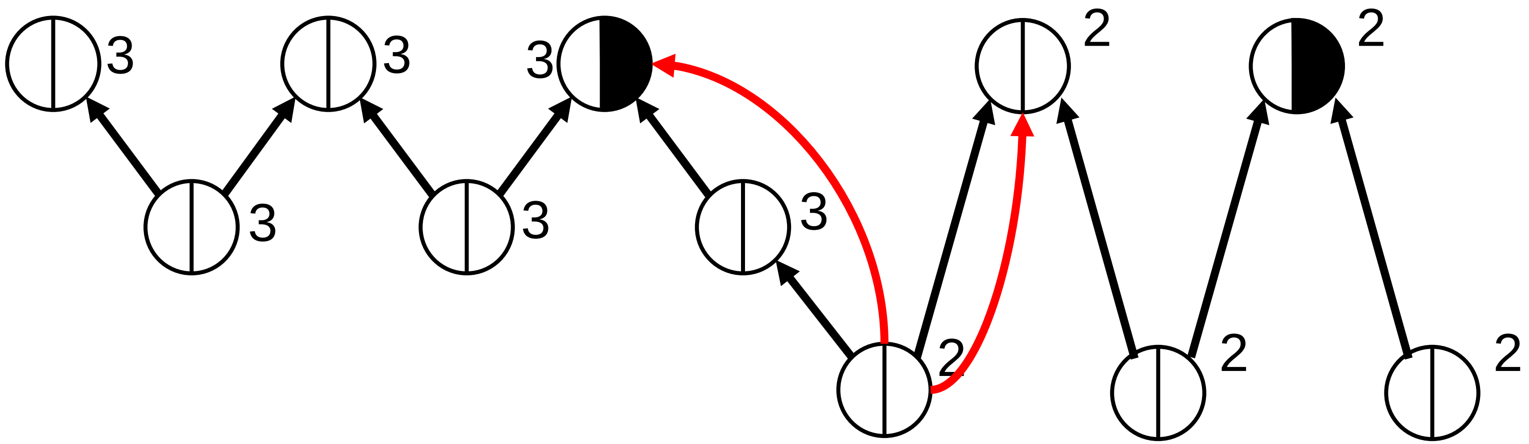


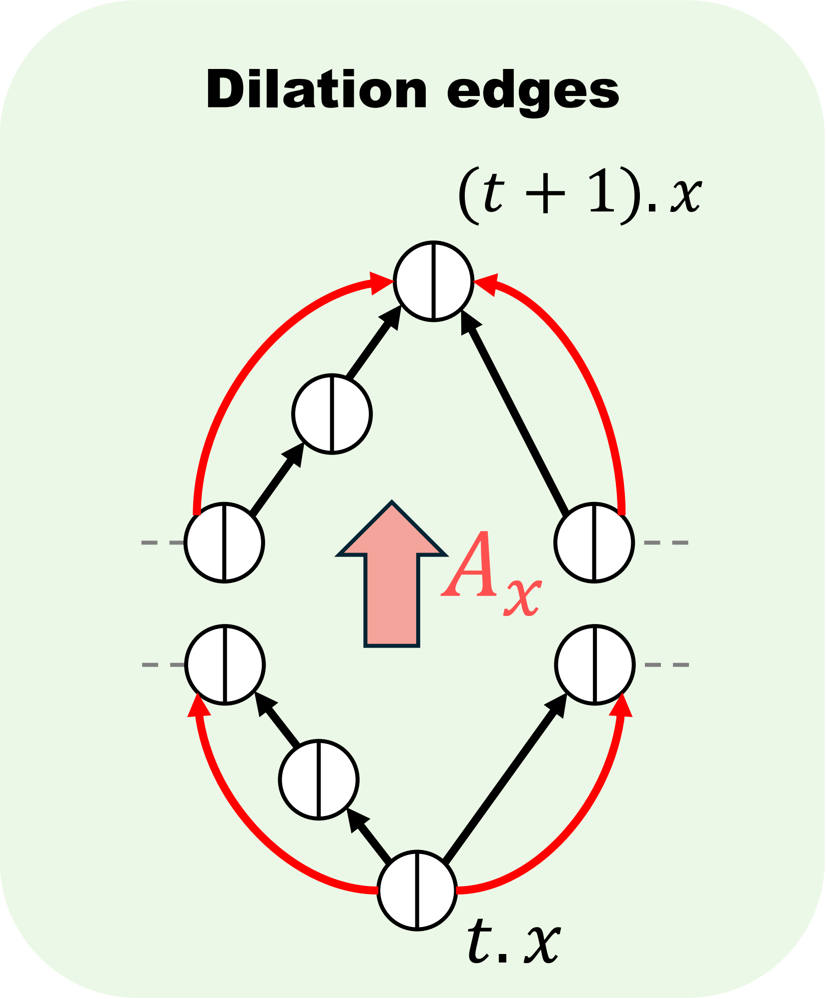
Time dilation example



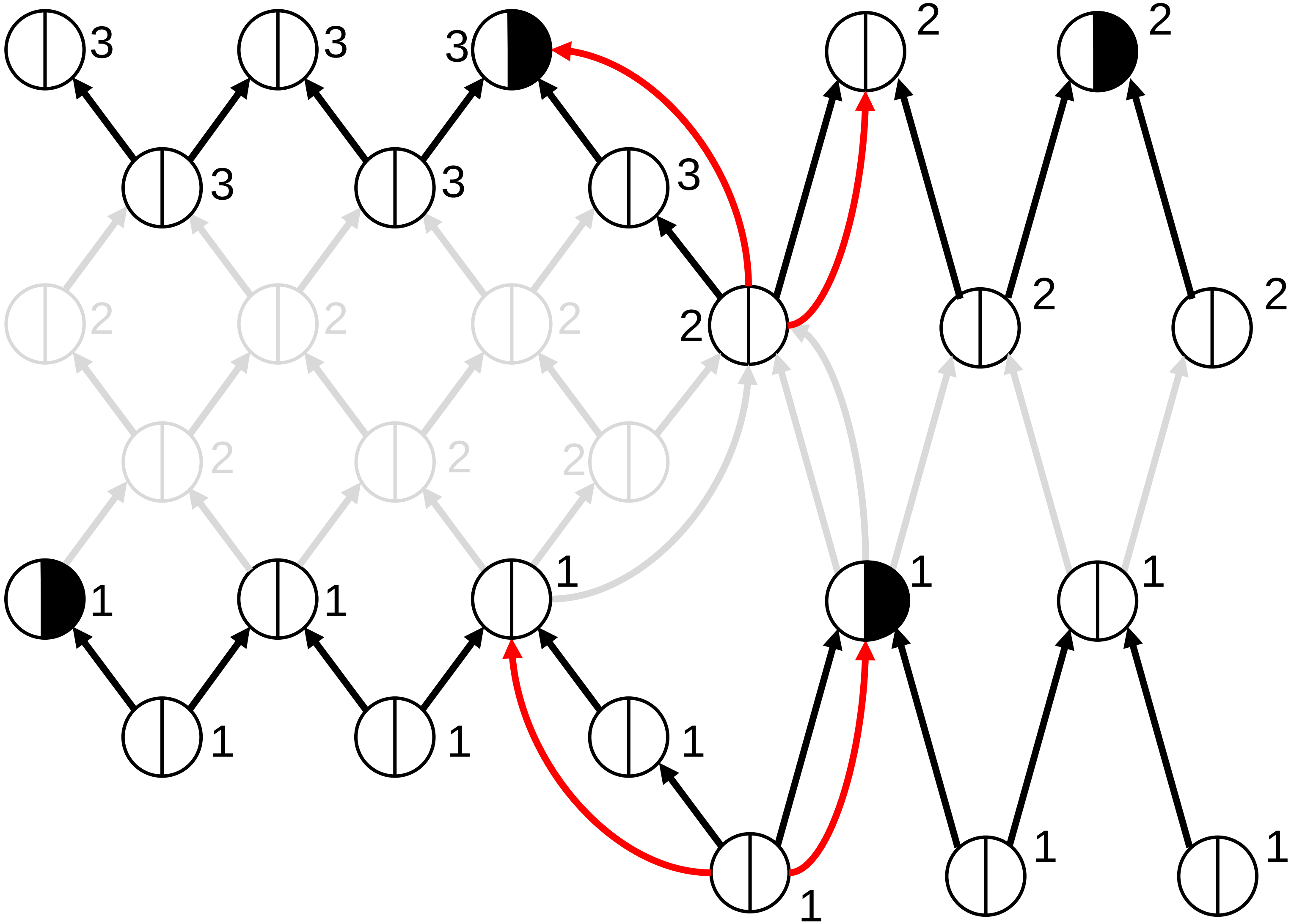


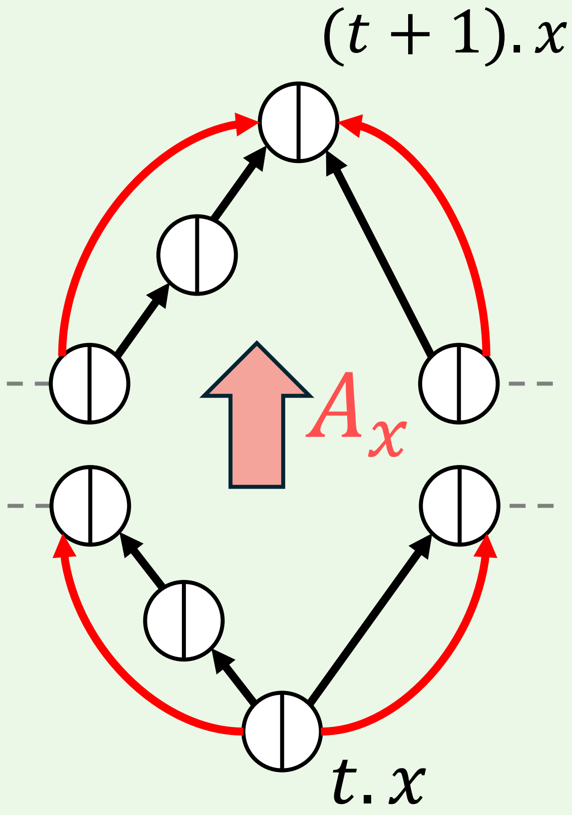
Time dilation example





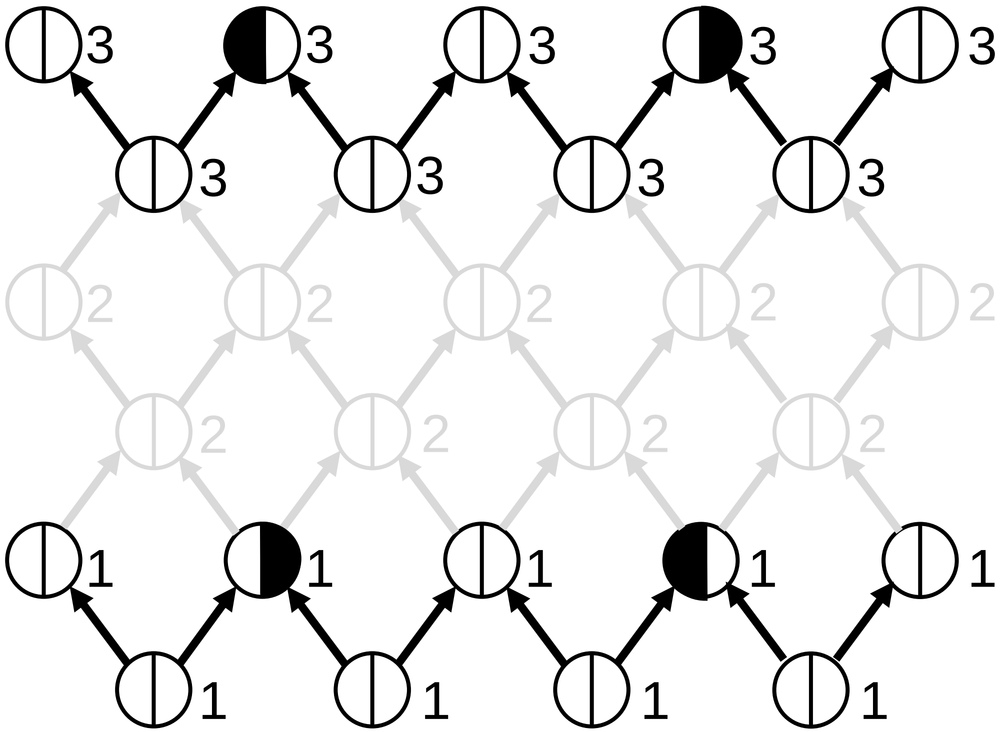
Time dilation example



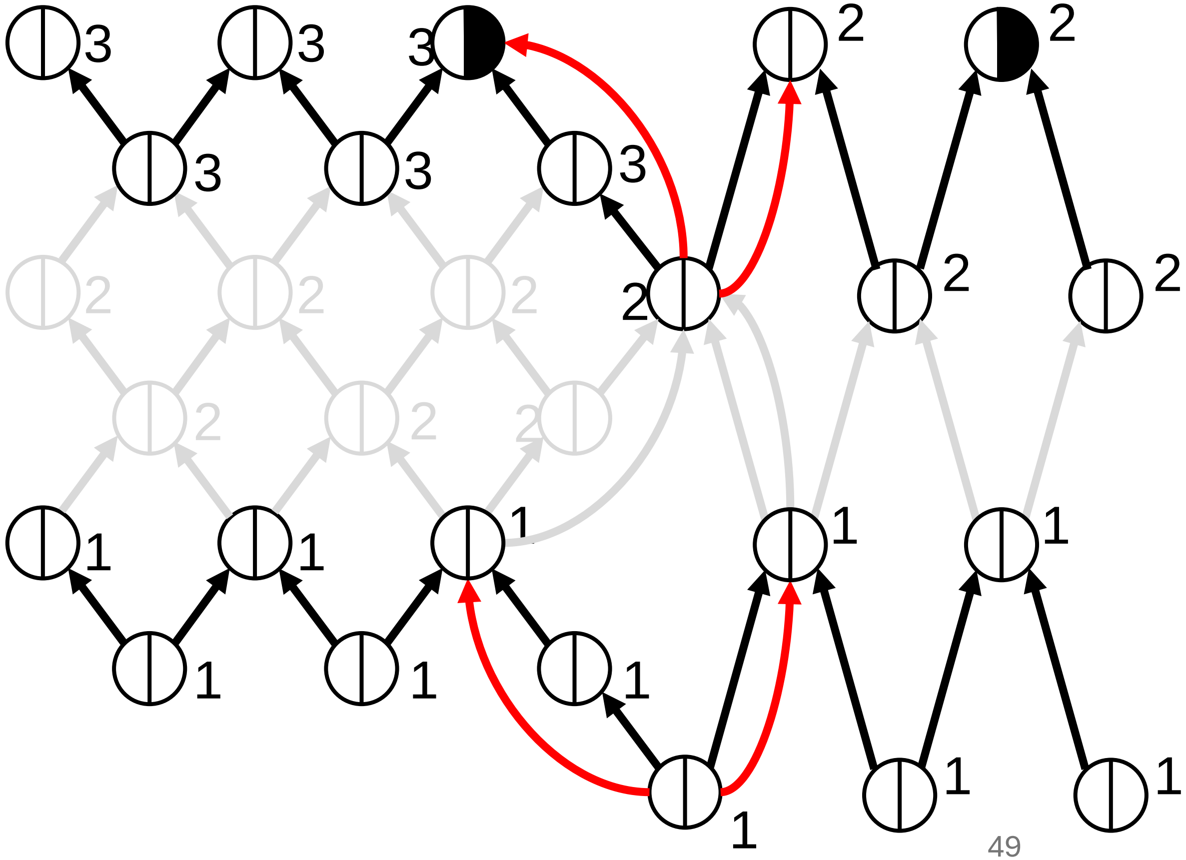


Summary

Using graph rewriting we **can simulate synchronous** dynamical system …

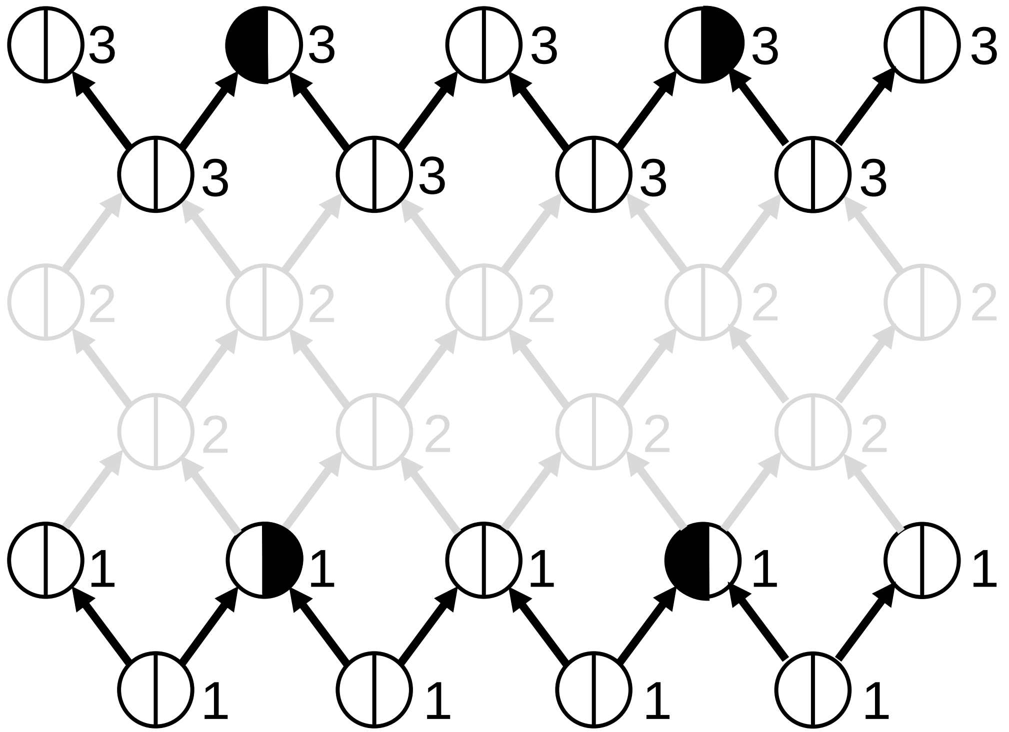


… but also represent some **intrinsically asynchronous** evolution.

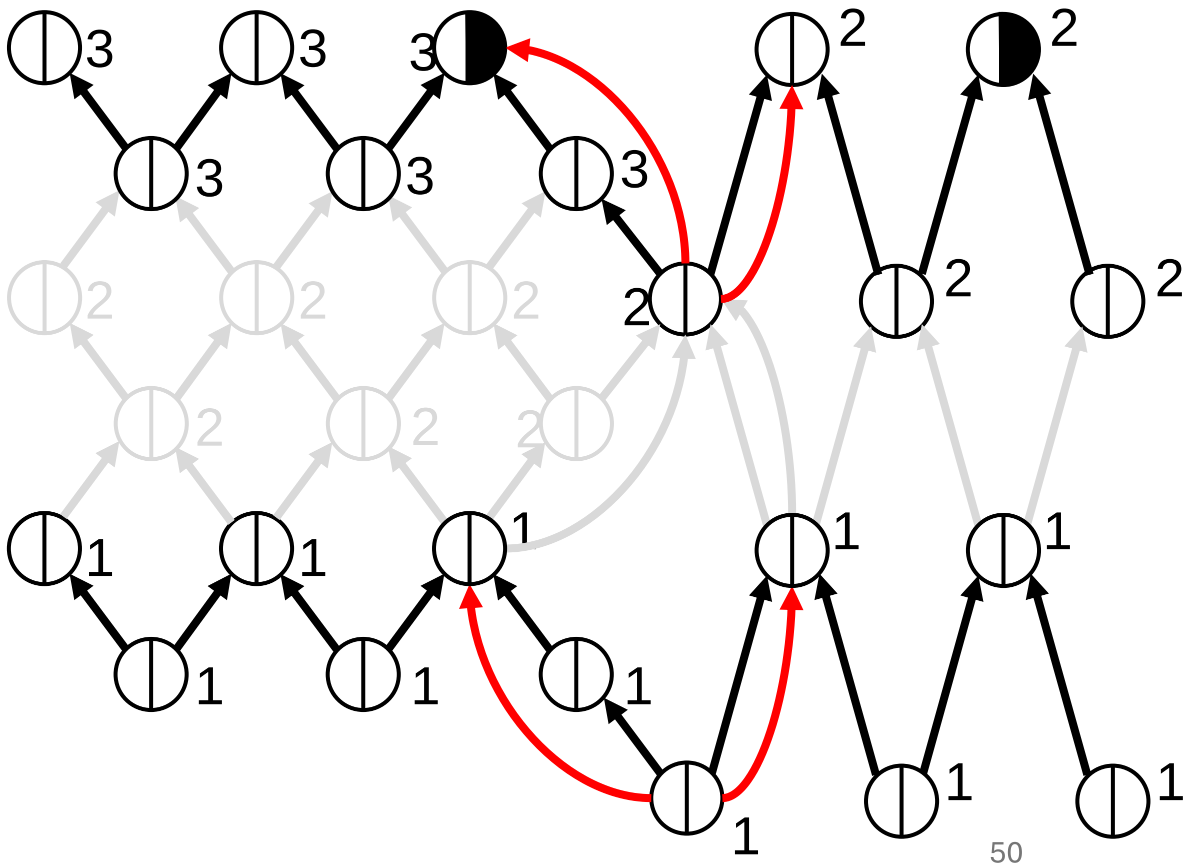


Summary

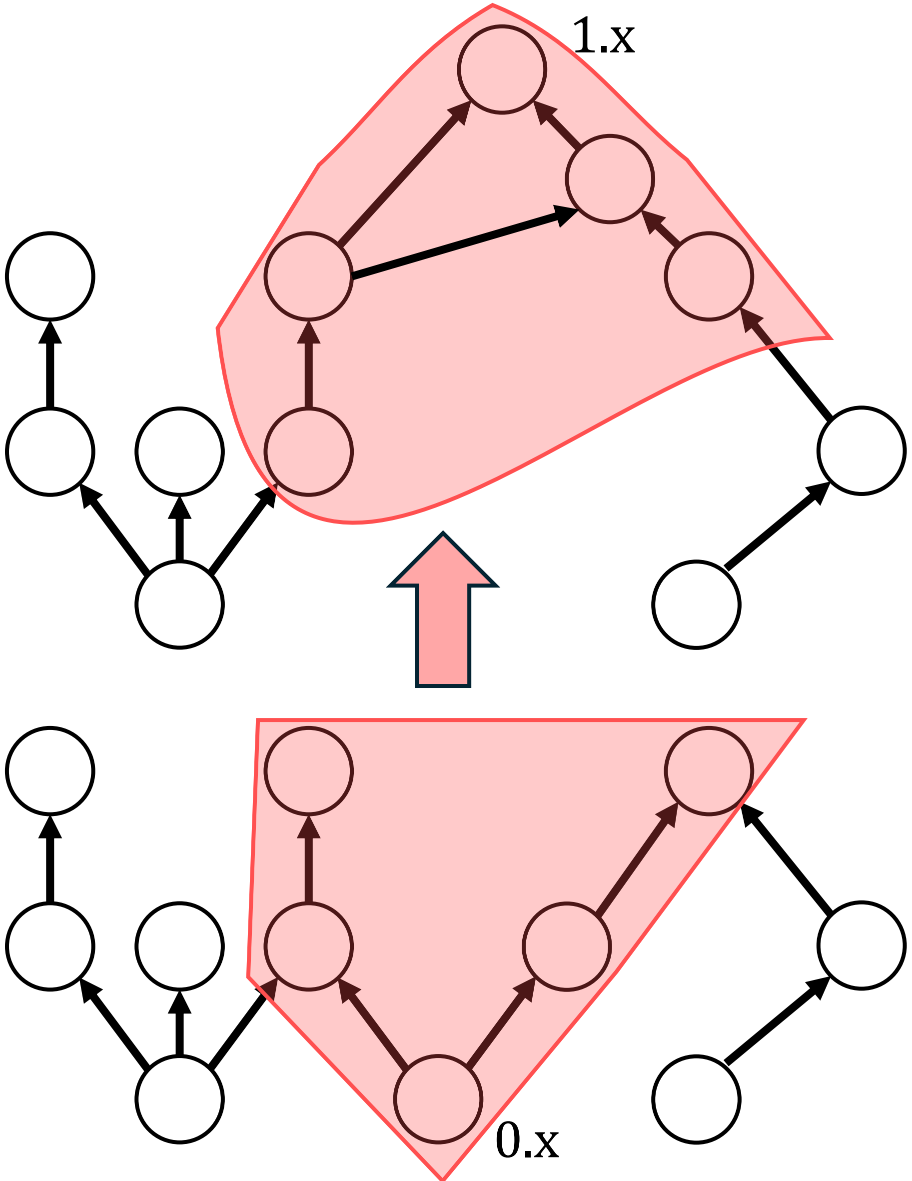
Using graph rewriting we **can simulate synchronous** dynamical system …



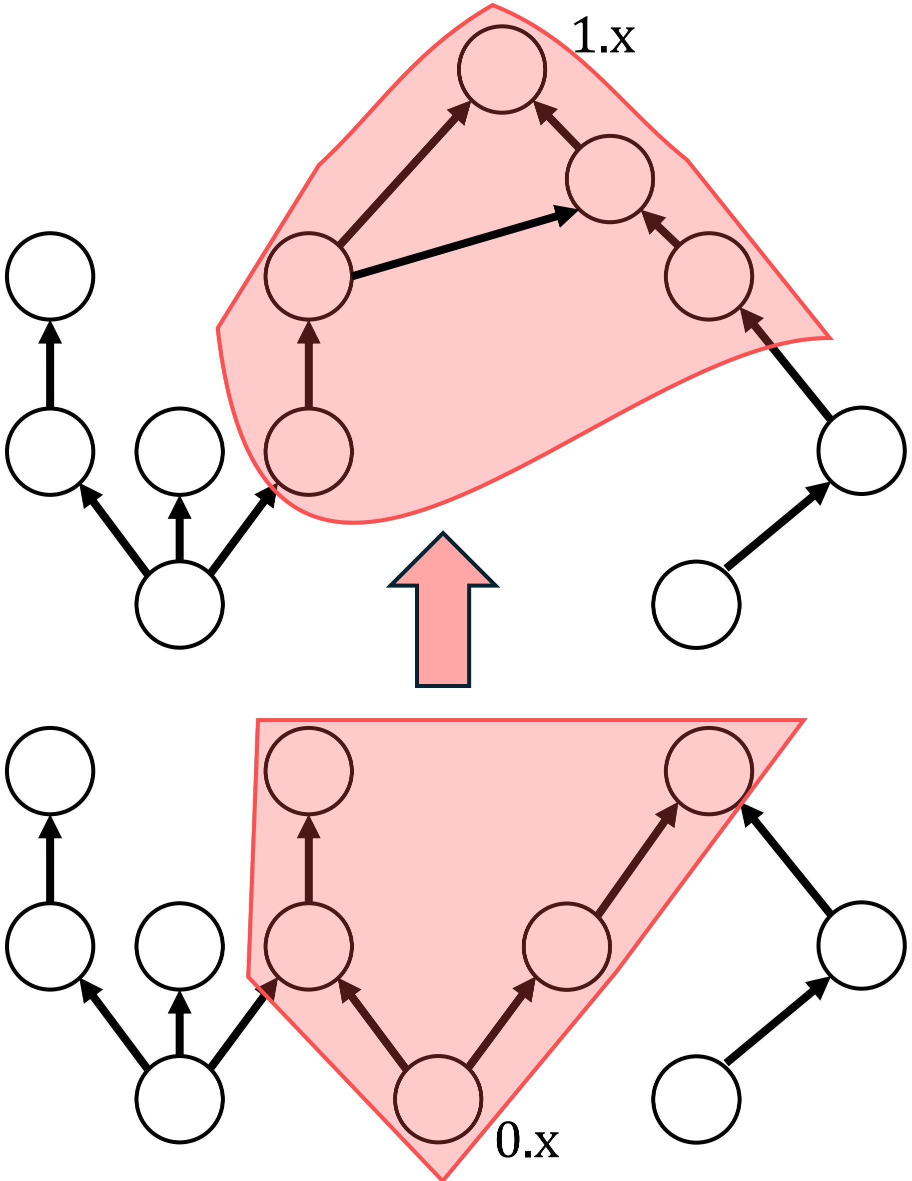
… but also represent some **intrinsically asynchronous** evolution.



General framework



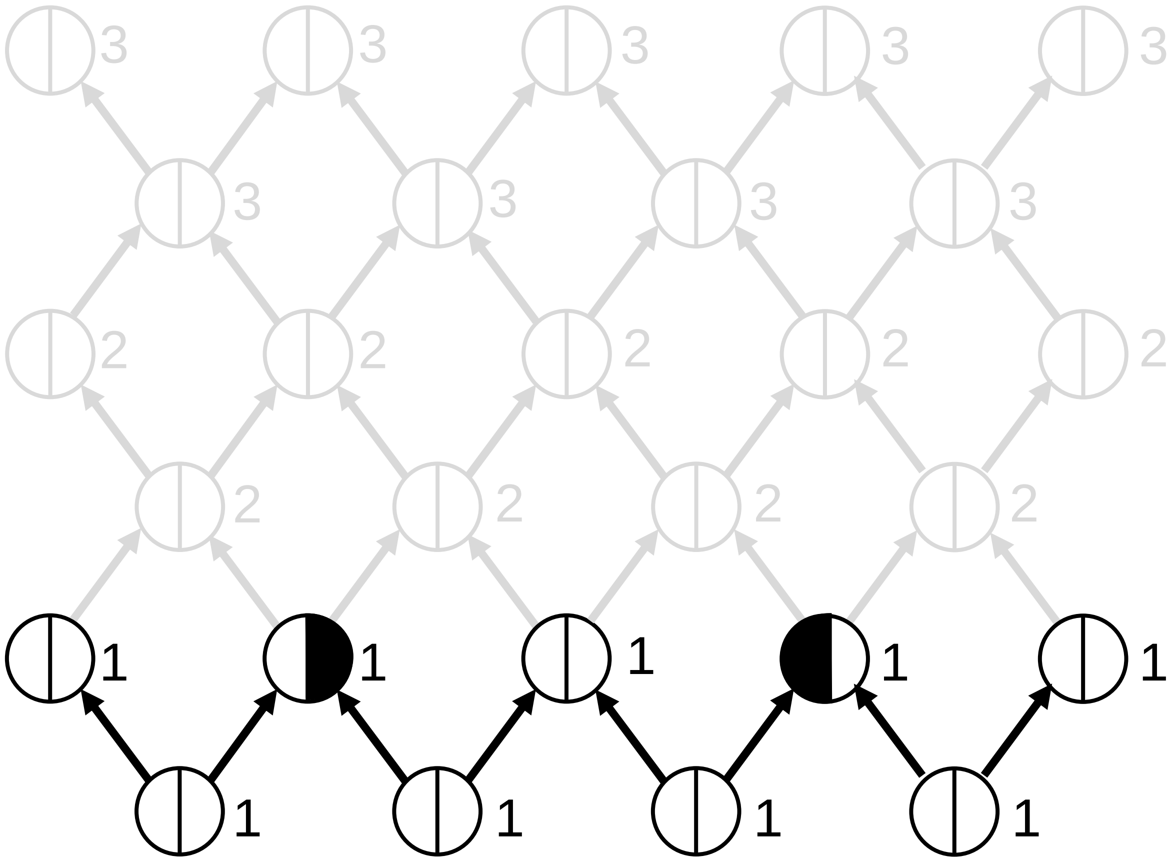
General framework



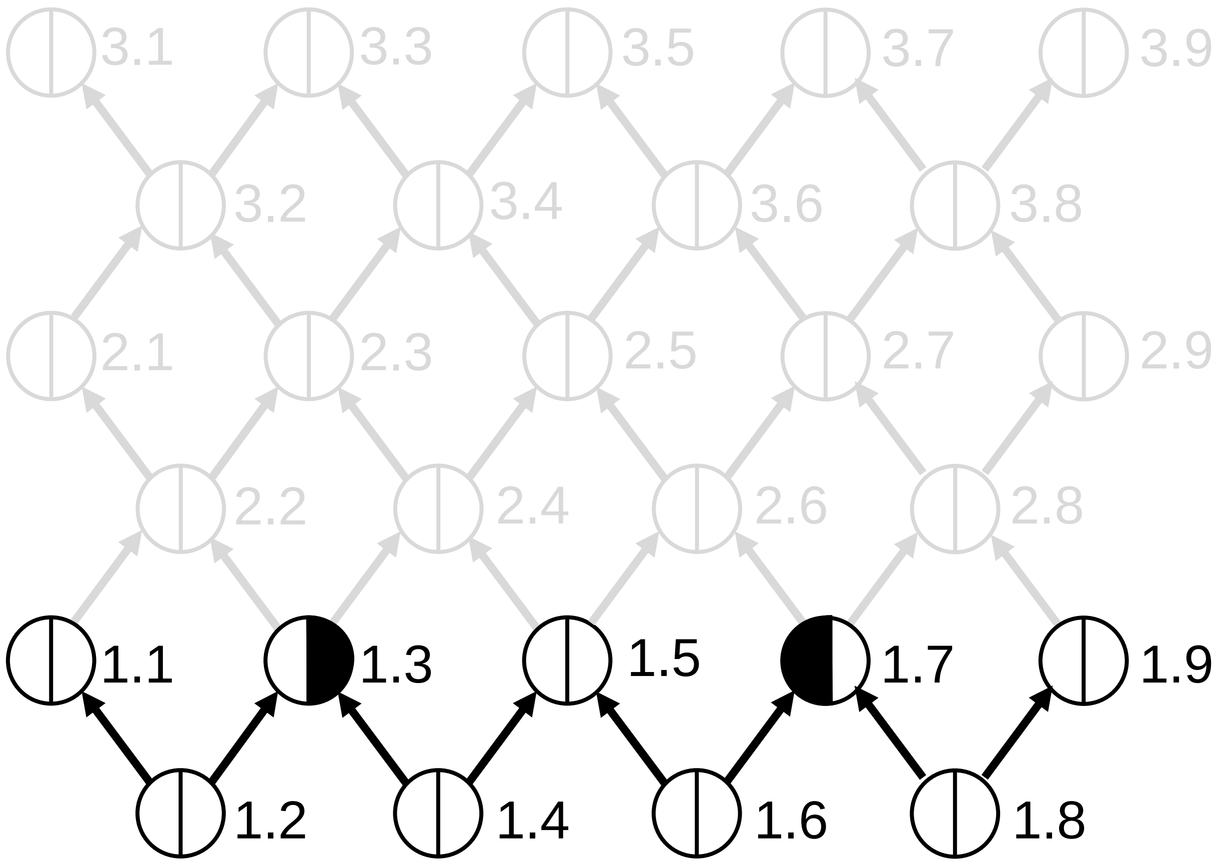
In general which local rewriting rules are physical ?

- **Determinism**
- Reversibility

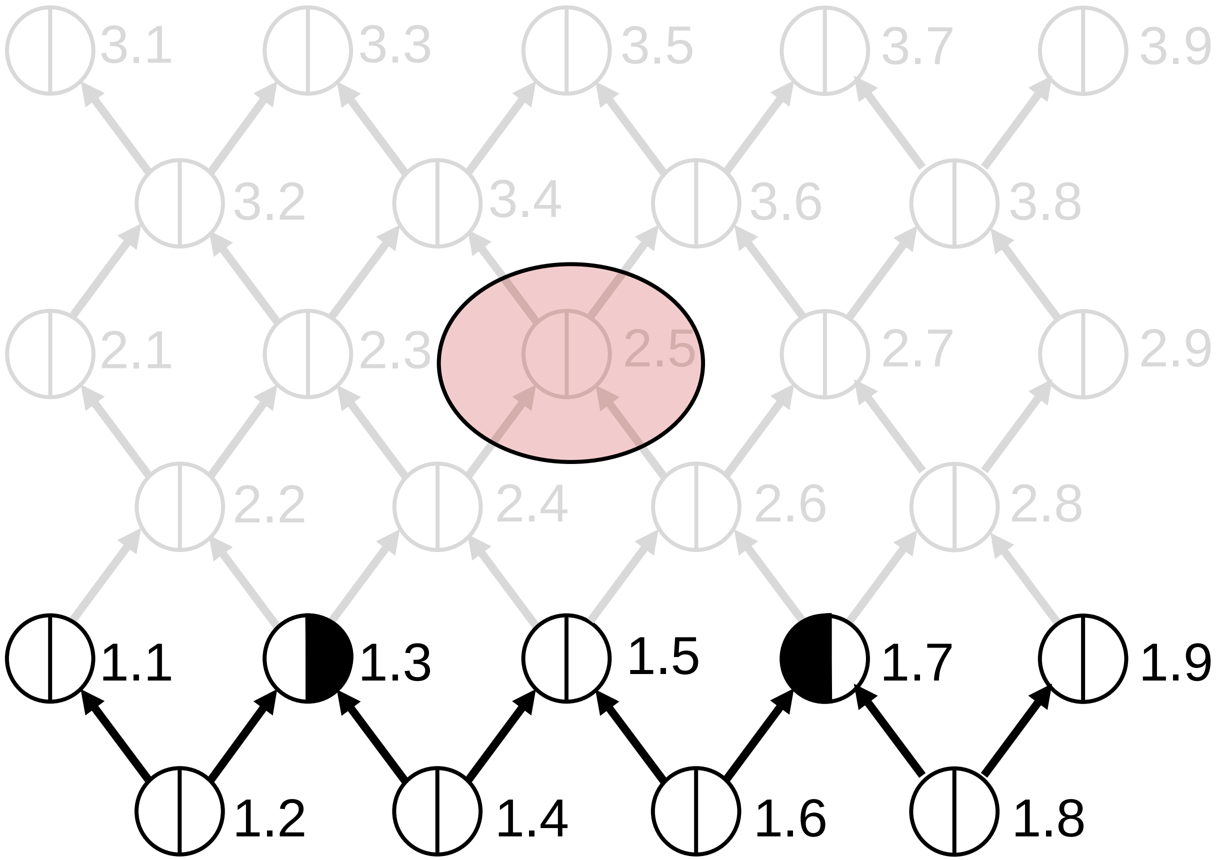
A first attempt at defining determinism



A first attempt at defining determinism



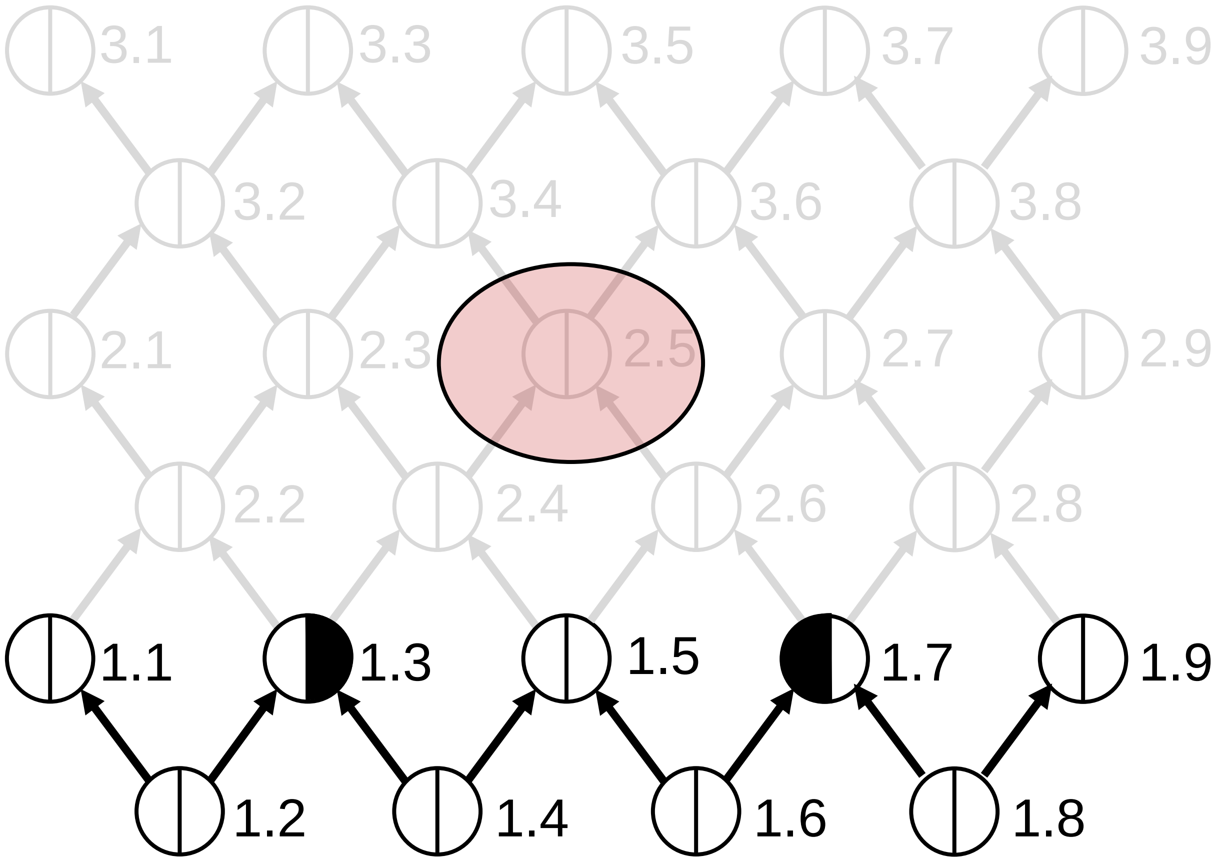
A first attempt at defining determinism



Determinism

The **state** of 2.5 is **always the same**. It does not depend on the rewriting strategy.

A first attempt at defining determinism



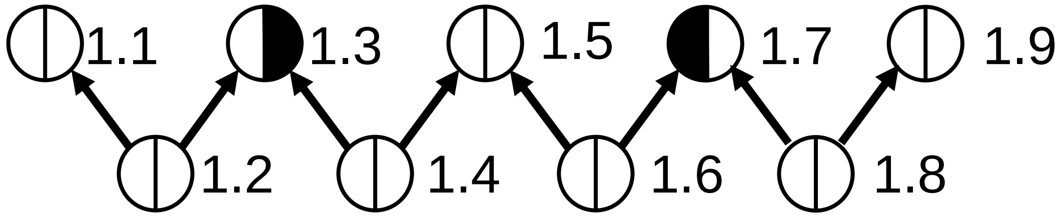
Determinism

The **state** of 2.5 is **always the same**. It does not depend on the rewriting strategy.

It does depend on the local shape of the cut !

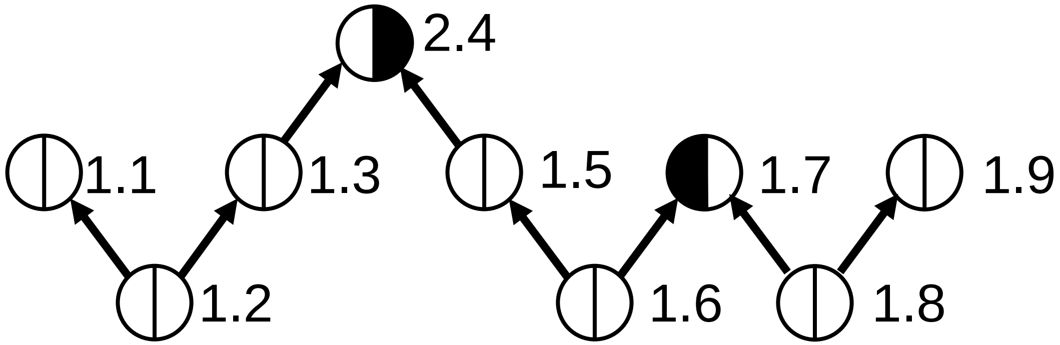
States depend on the cut

G



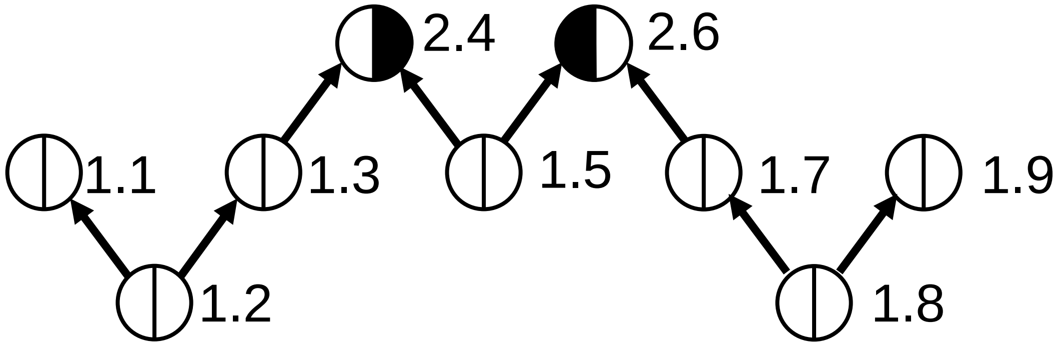
States depend on the cut

$A_4 G$



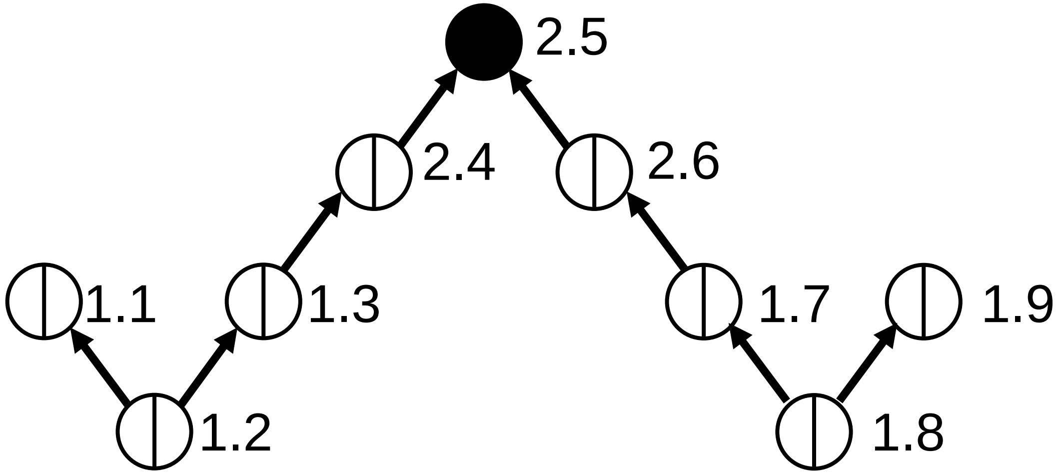
States depend on the cut

$A_6 A_4 G$



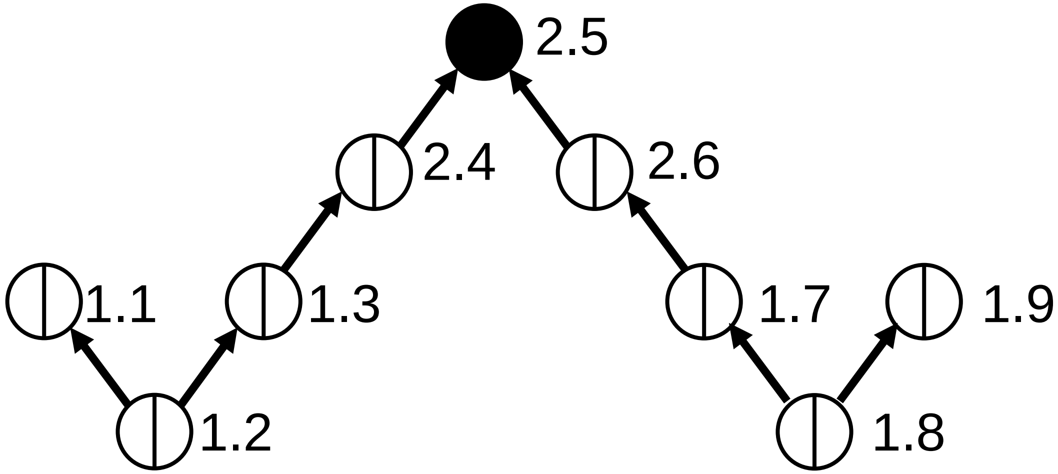
States depend on the cut

$A_5 A_6 A_4 G$

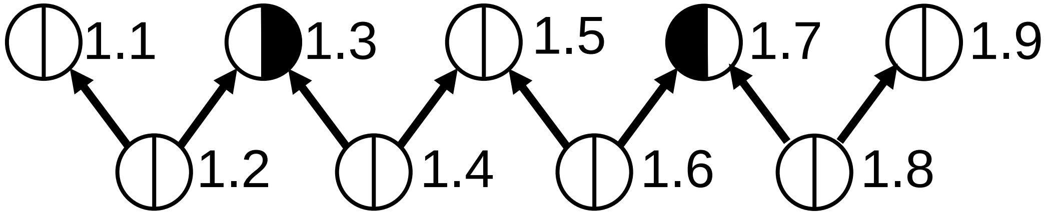


States depend on the cut

$A_5 A_6 A_4 G$

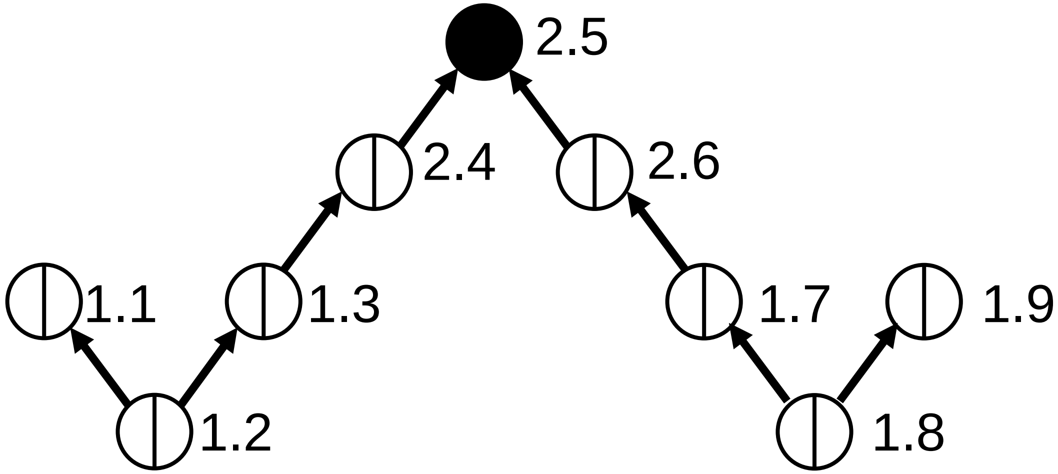


G

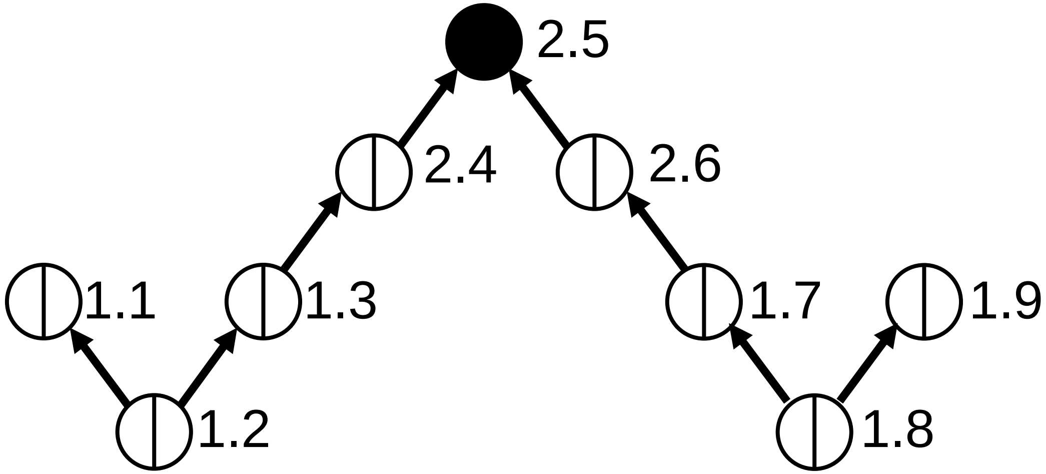


States depend on the cut

$A_5 A_6 A_4 G$

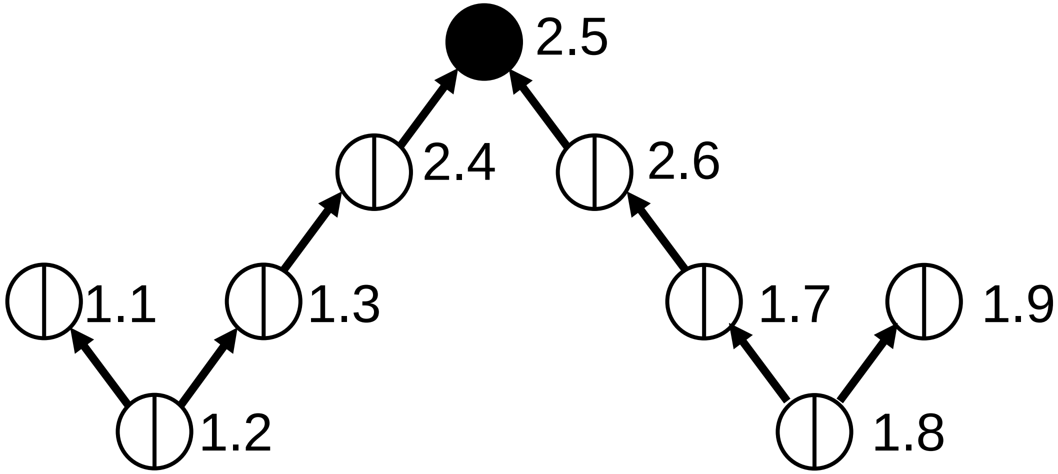


$A_5 A_6 A_4 G$

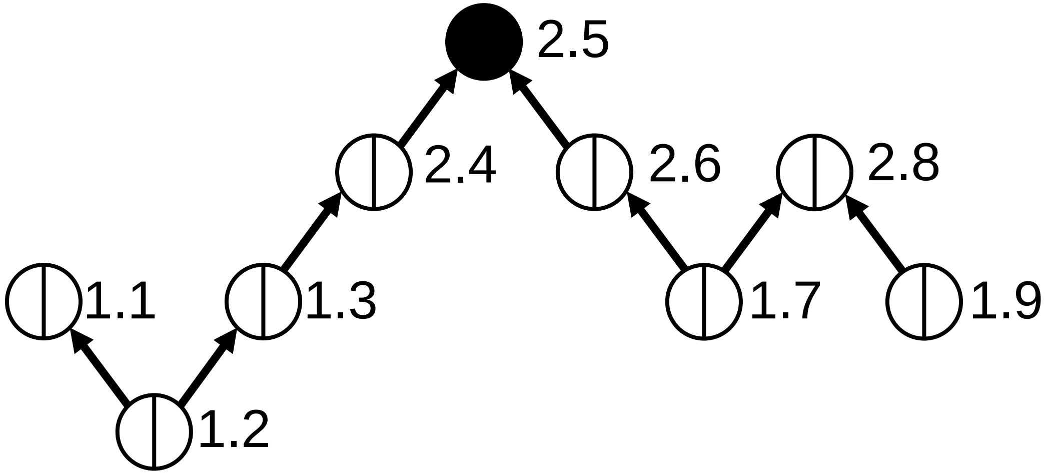


States depend on the cut

$A_5 A_6 A_4 G$



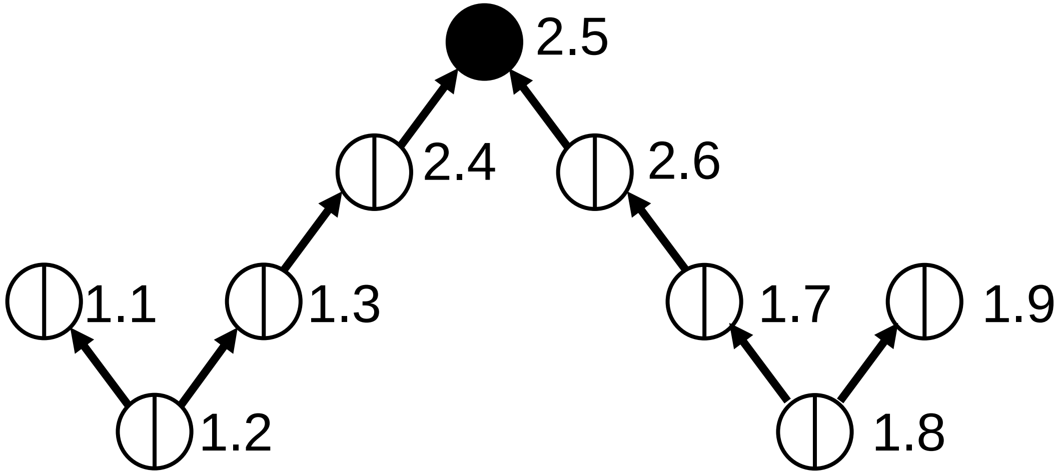
$A_8 A_5 A_6 A_4 G$

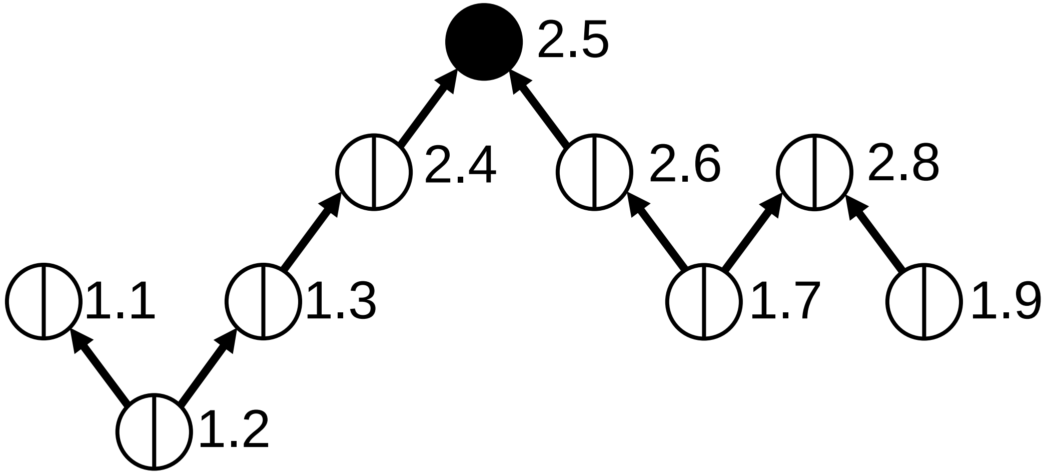


States depend on the cut

$A_5 A_6 A_4 G$

$A_{8564} G$

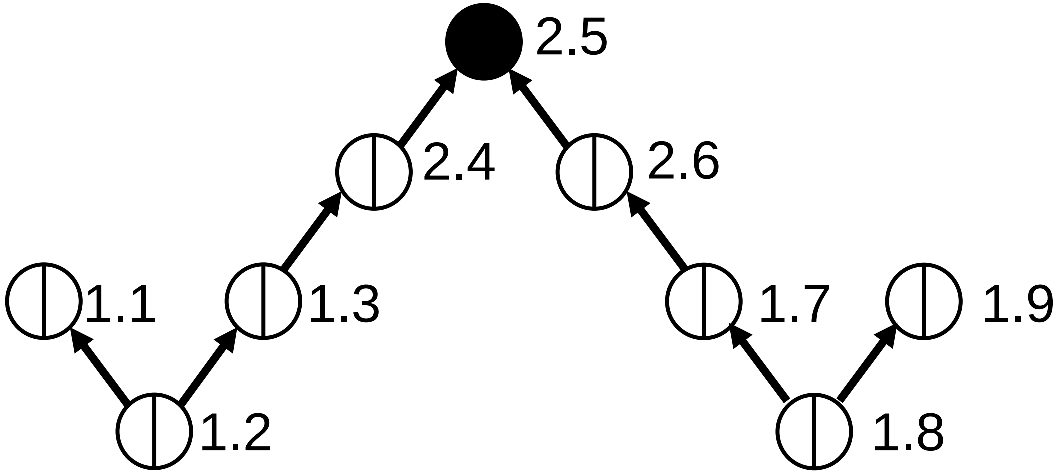


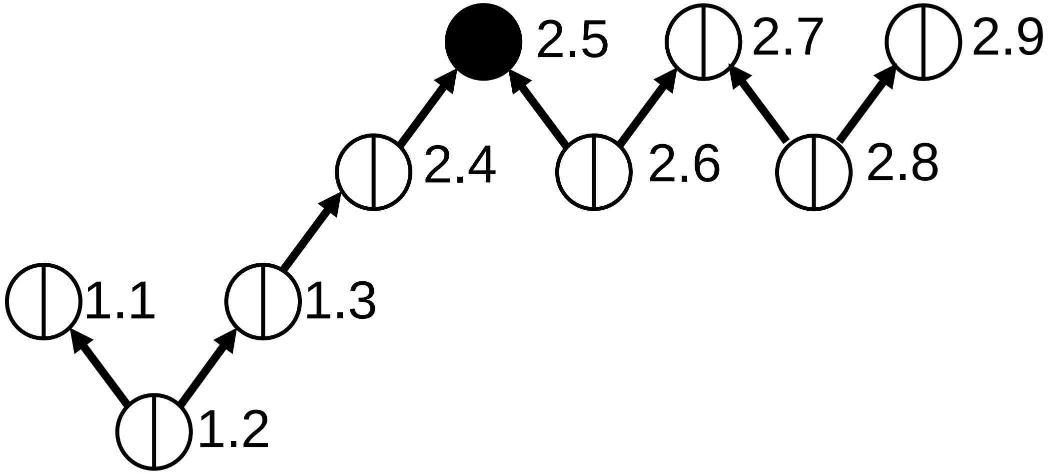


States depend on the cut

$A_5 A_6 A_4 G$

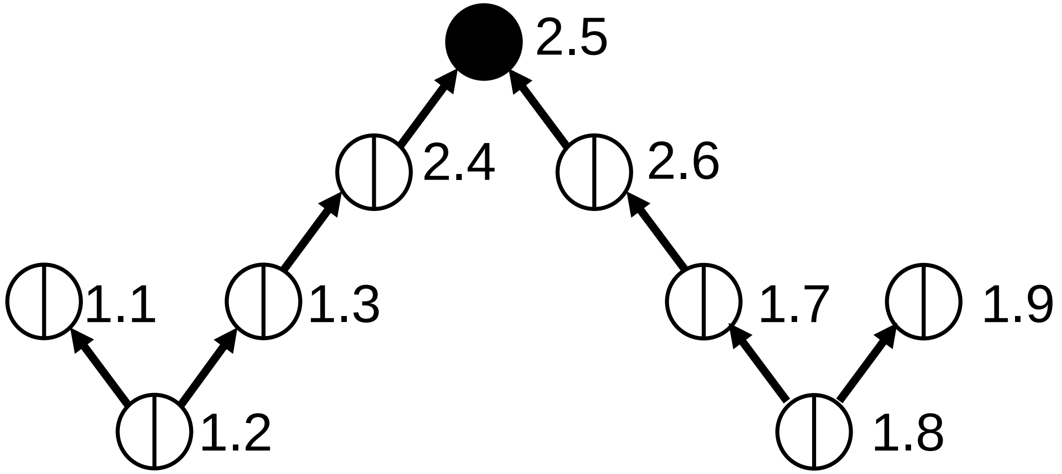
$A_{978564} G$



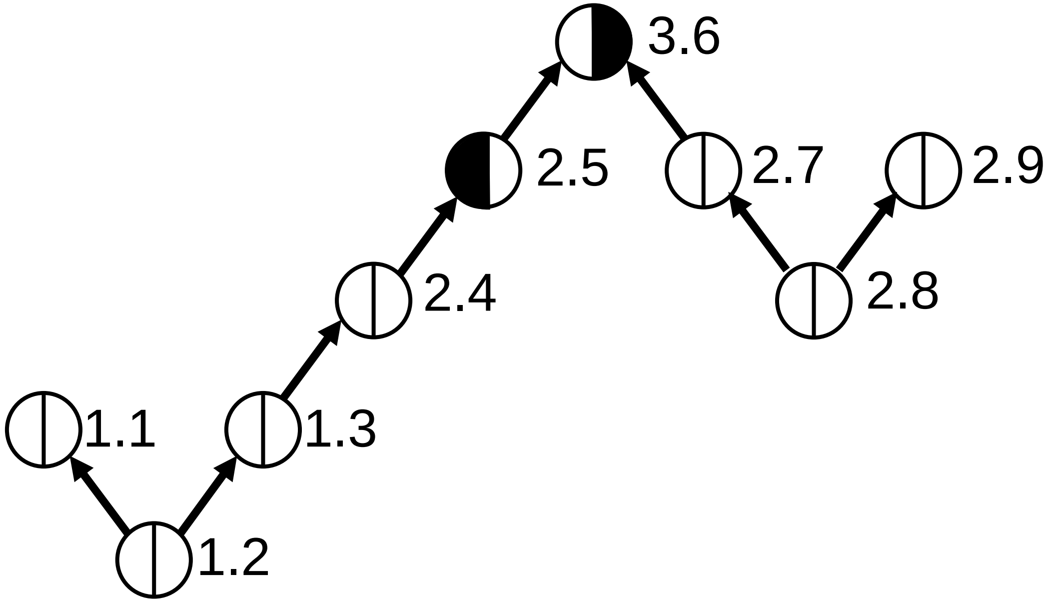


States depend on the cut

$A_5 A_6 A_4 G$

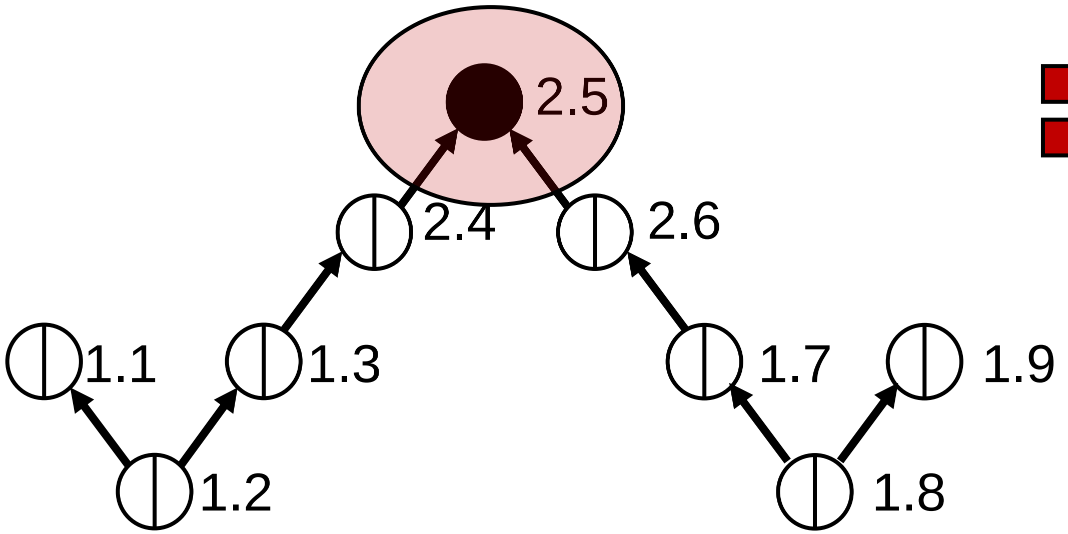


$A_{6978564} G$



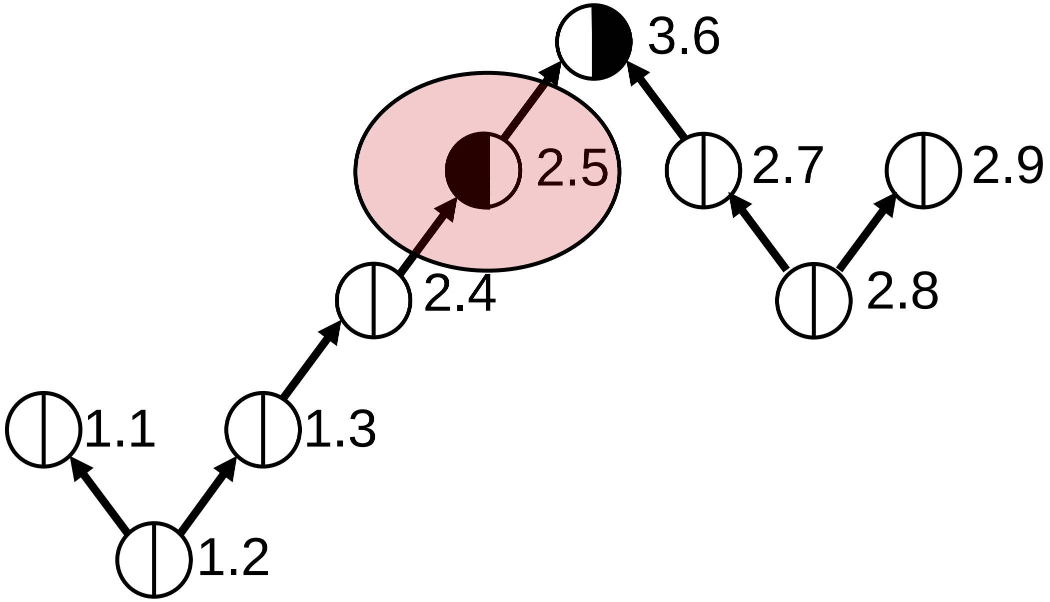
States depend on the cut

$A_5 A_6 A_4 G$

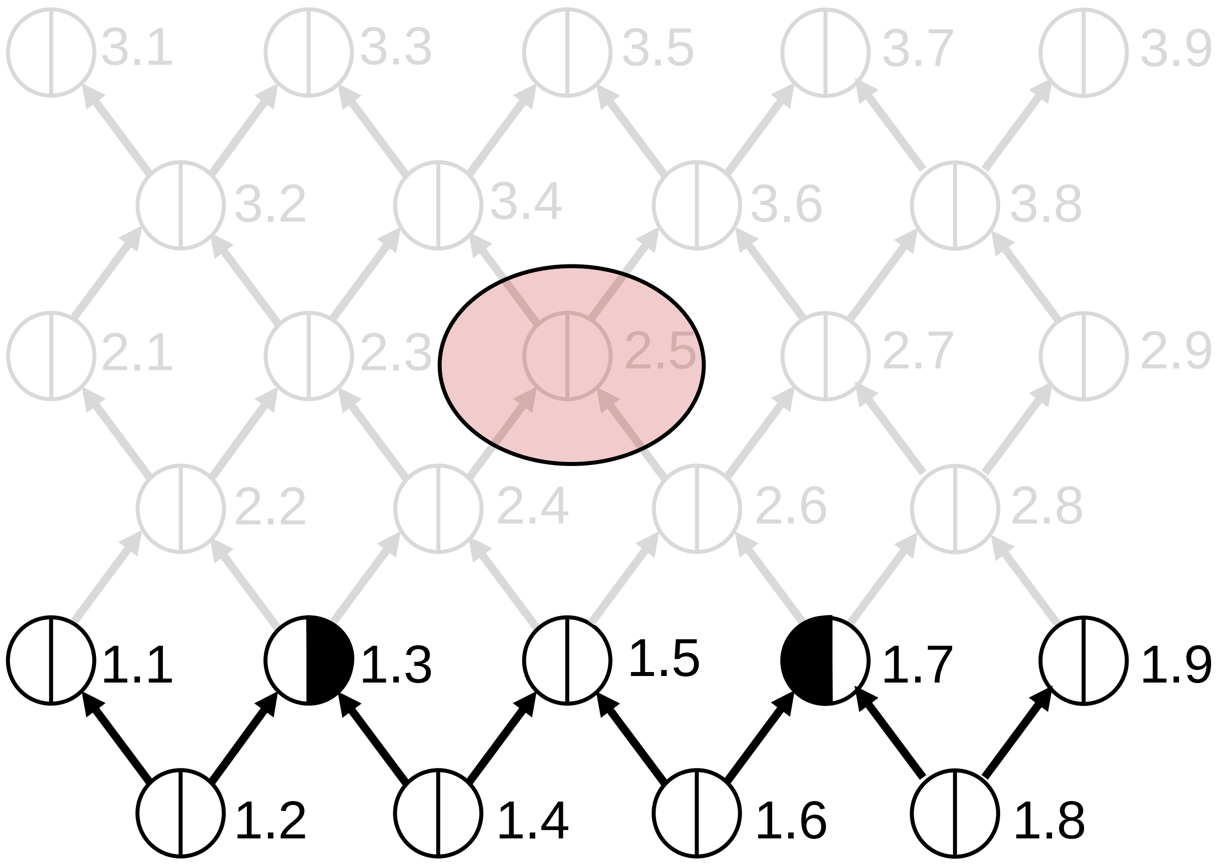


$\neq$

$A_{6978564} G$



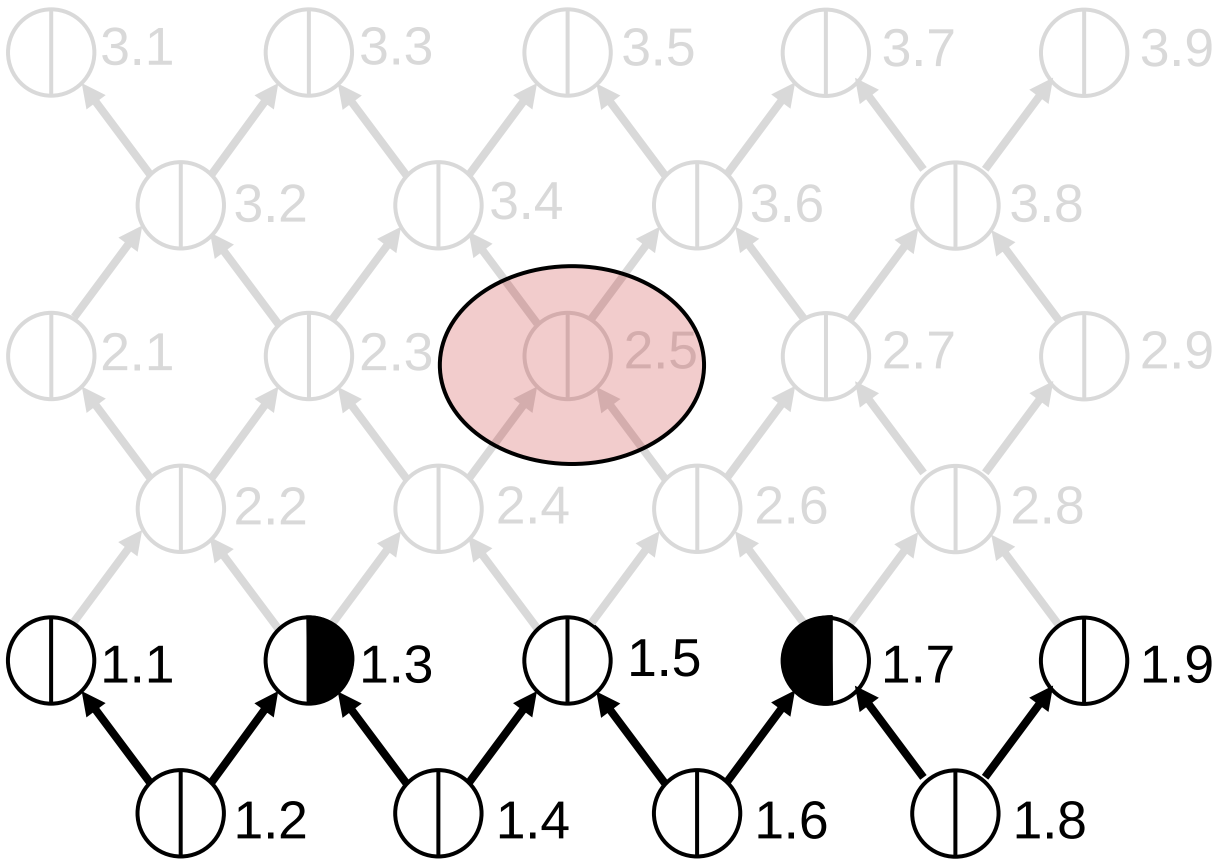
Space-time Determinism



Space-time Determinism

The **state** of 2.5 is **always the same for 2 locally identical cuts**. It does not depend on the rewriting strategy.

Space-time Determinism



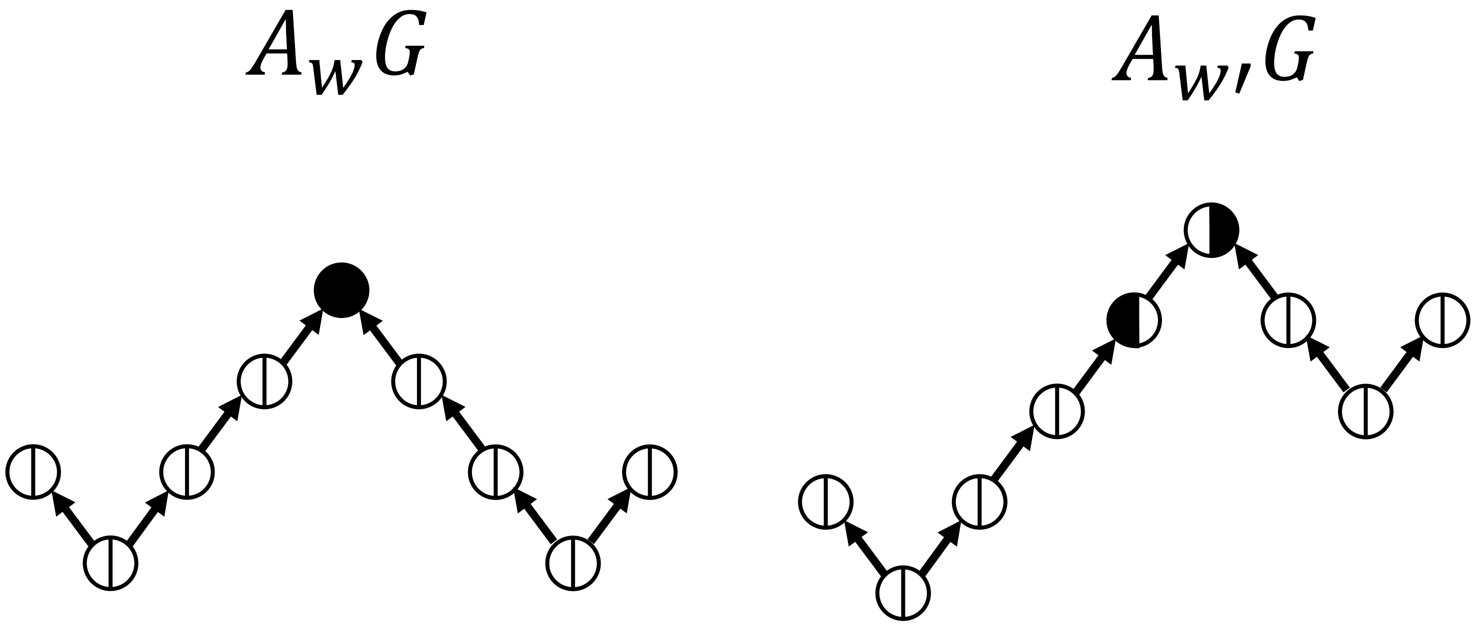
Space-time Determinism

The **state** of 2.5 is **always the same for 2 locally identical cuts**. It does not depend on the rewriting strategy.

A **Deterministic**

Same incoming edges ⇒ Same internal state and outgoing edges

Space-time Determinism



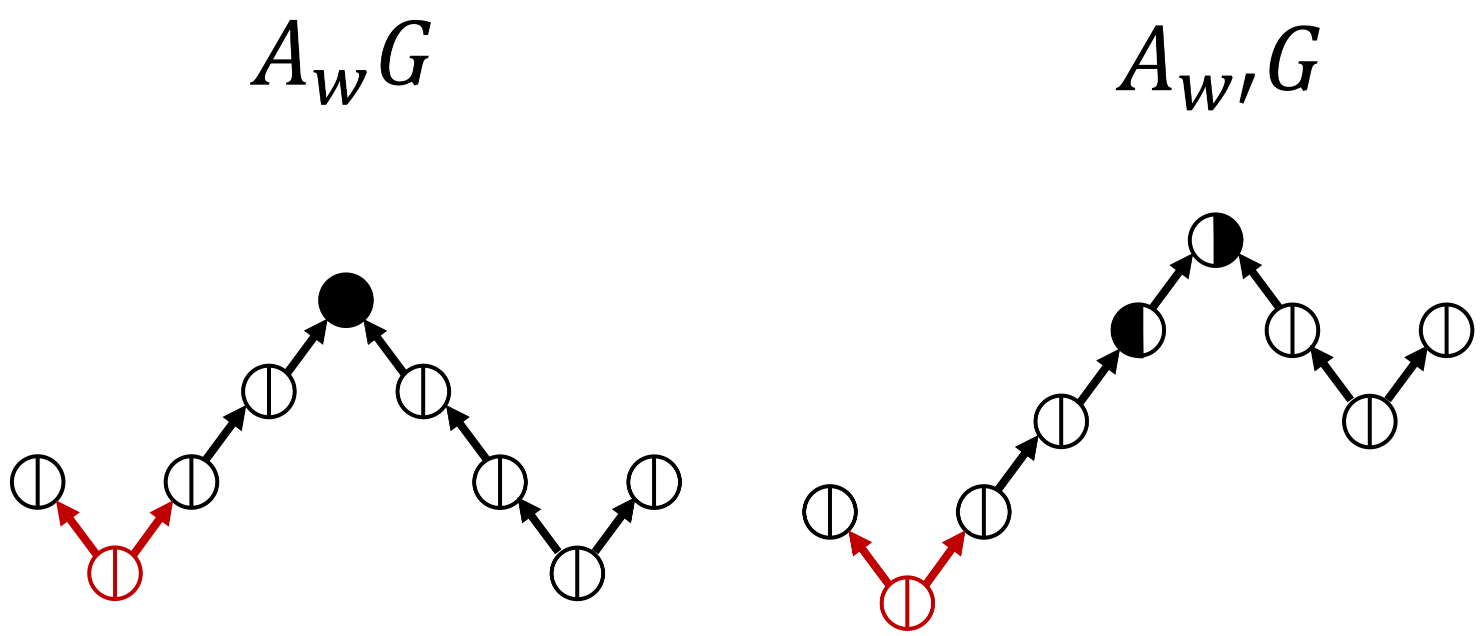
Space-time Determinism

The **state** of 2.5 is **always the same for 2 locally identical cuts**. It does not depend on the rewriting strategy.

***A* Deterministic**

Same incoming edges ⇒ Same internal state and outgoing edges

Space-time Determinism



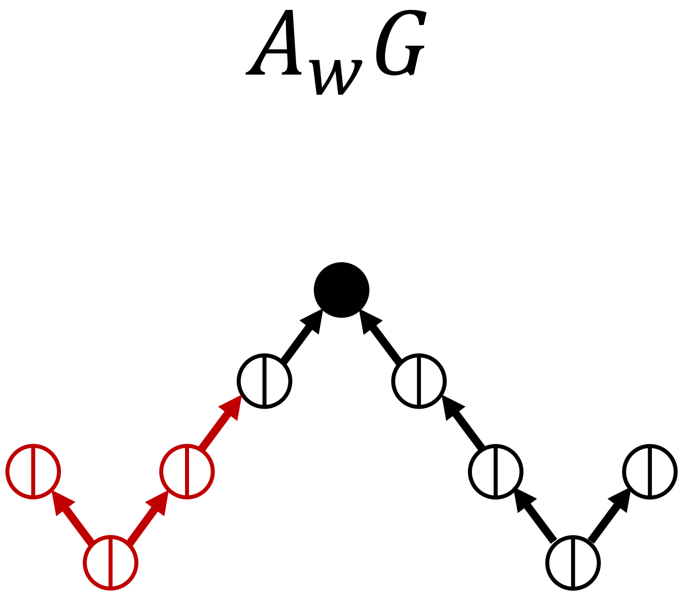
Space-time Determinism

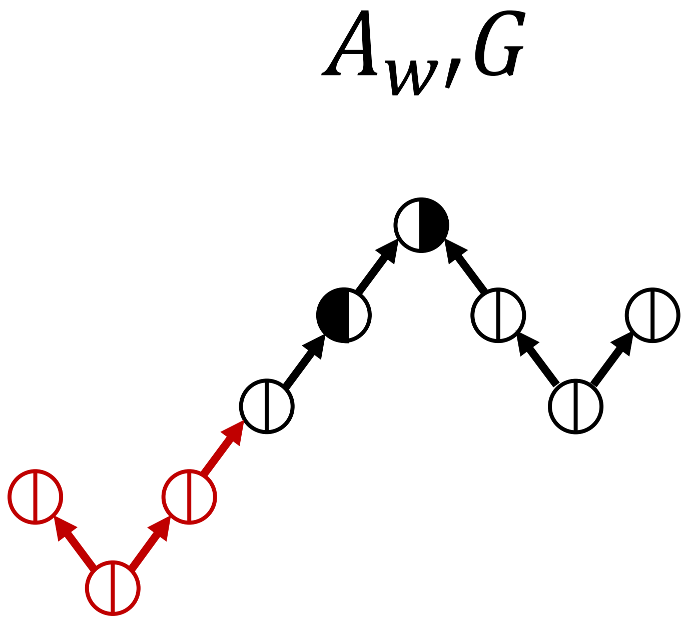
The **state** of 2.5 is **always the same for 2 locally identical cuts**. It does not depend on the rewriting strategy.

***A* Deterministic**

Same incoming edges ⇒ Same internal state and outgoing edges

Space-time Determinism





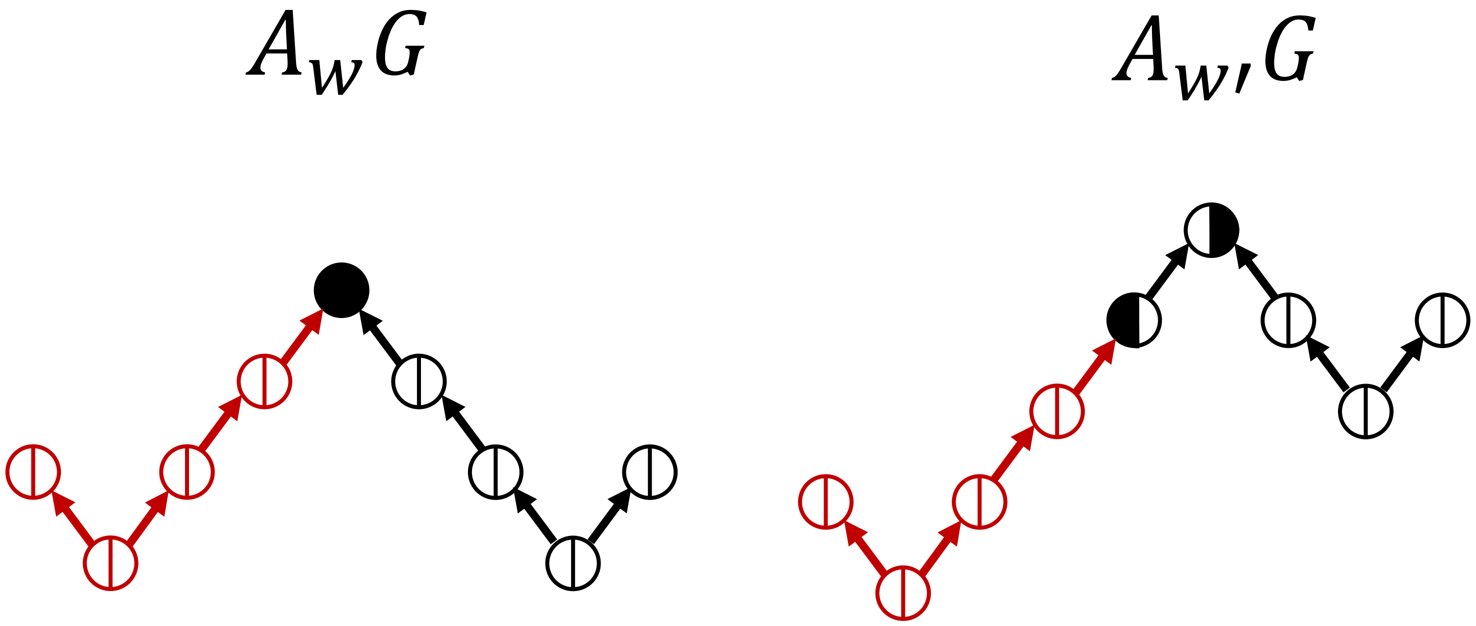
Space-time Determinism

The **state** of 2.5 is **always the same for 2 locally identical cuts**. It does not depend on the rewriting strategy.

A **Deterministic**

Same incoming edges ⇒ Same internal state and outgoing edges

Space-time Determinism



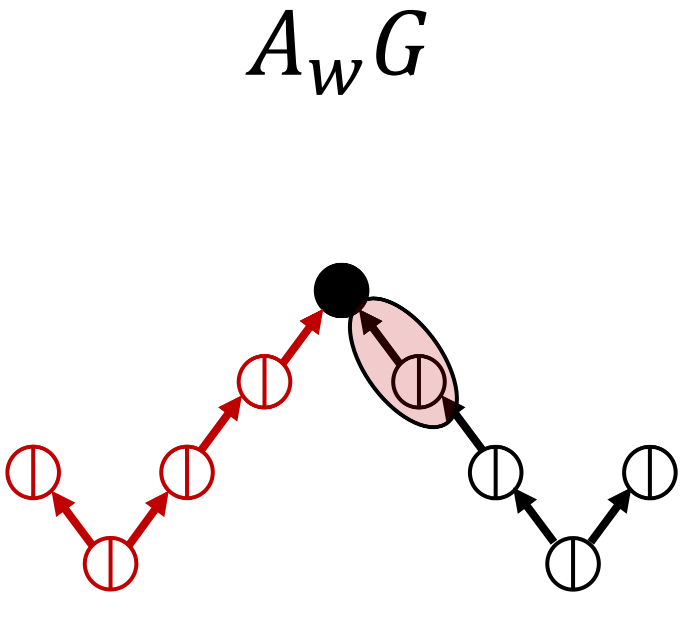
Space-time Determinism

The **state** of 2.5 is **always the same for 2 locally identical cuts**. It does not depend on the rewriting strategy.

***A* Deterministic**

Same incoming edges ⇒ Same internal state and outgoing edges

Space-time Determinism



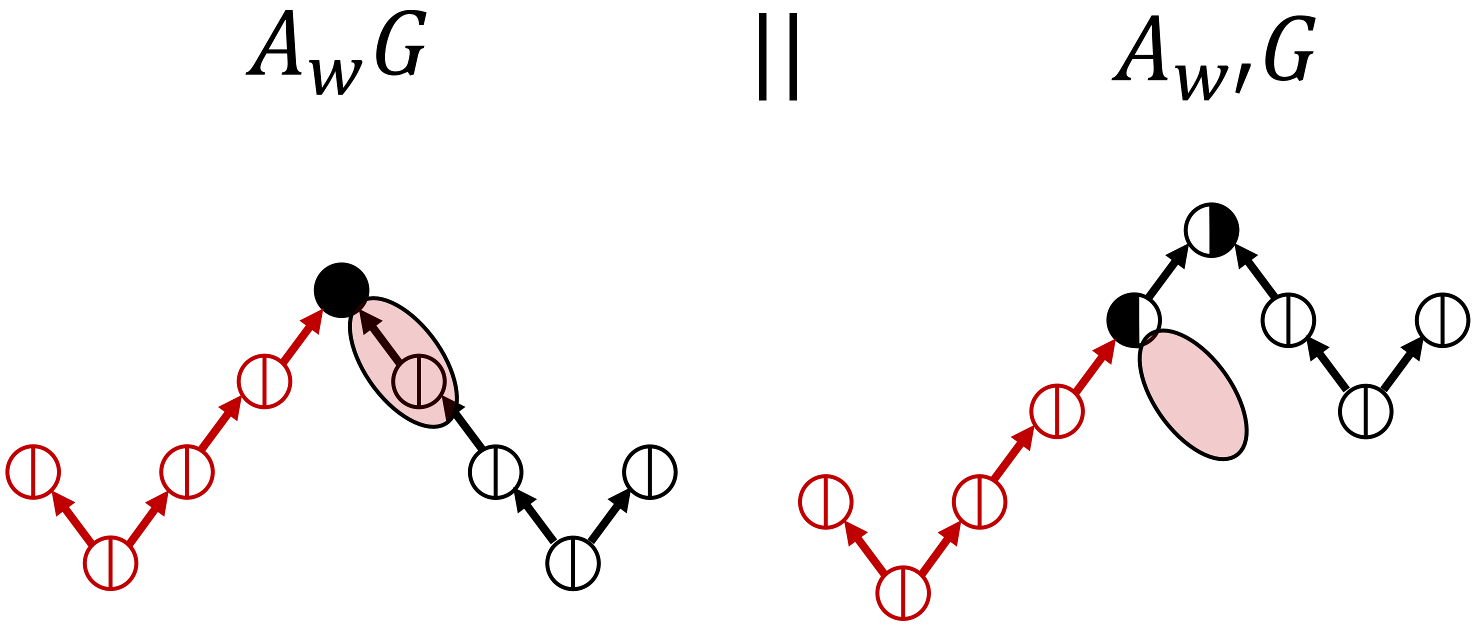
Space-time Determinism

The **state** of 2.5 is **always the same for 2 locally identical cuts**. It does not depend on the rewriting strategy.

***A* Deterministic**

Same incoming edges ⇒ Same internal state and outgoing edges

Space-time Determinism



Space-time Determinism

The **state** of 2.5 is **always the same for 2 locally identical cuts**. It does not depend on the rewriting strategy.

A **Deterministic**

Same incoming edges ⇒ Same internal state and outgoing edges

Ensuring determinism

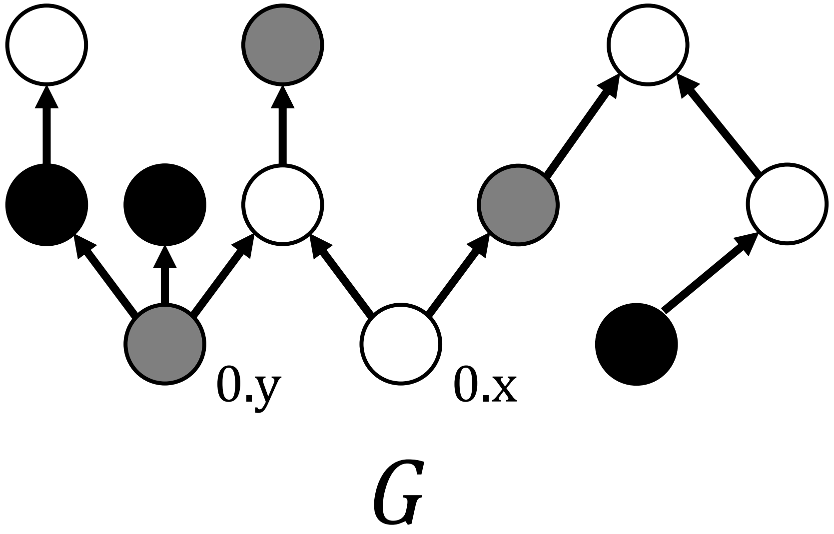
Goal : prove A deterministic, i.e. we always have $A_w || A_w$, which means :

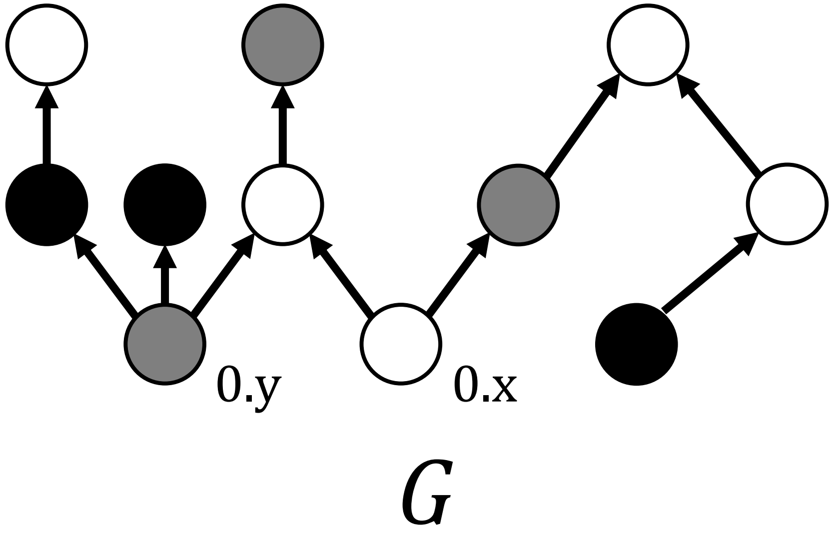
Same incomming edges ⇒ Same internal state and outgoing edges

Ensuring determinism

Goal : prove A deterministic, i.e. we always have $A_w||A_w$, which means :

Same incomming edges ⇒ Same internal state and outgoing edges





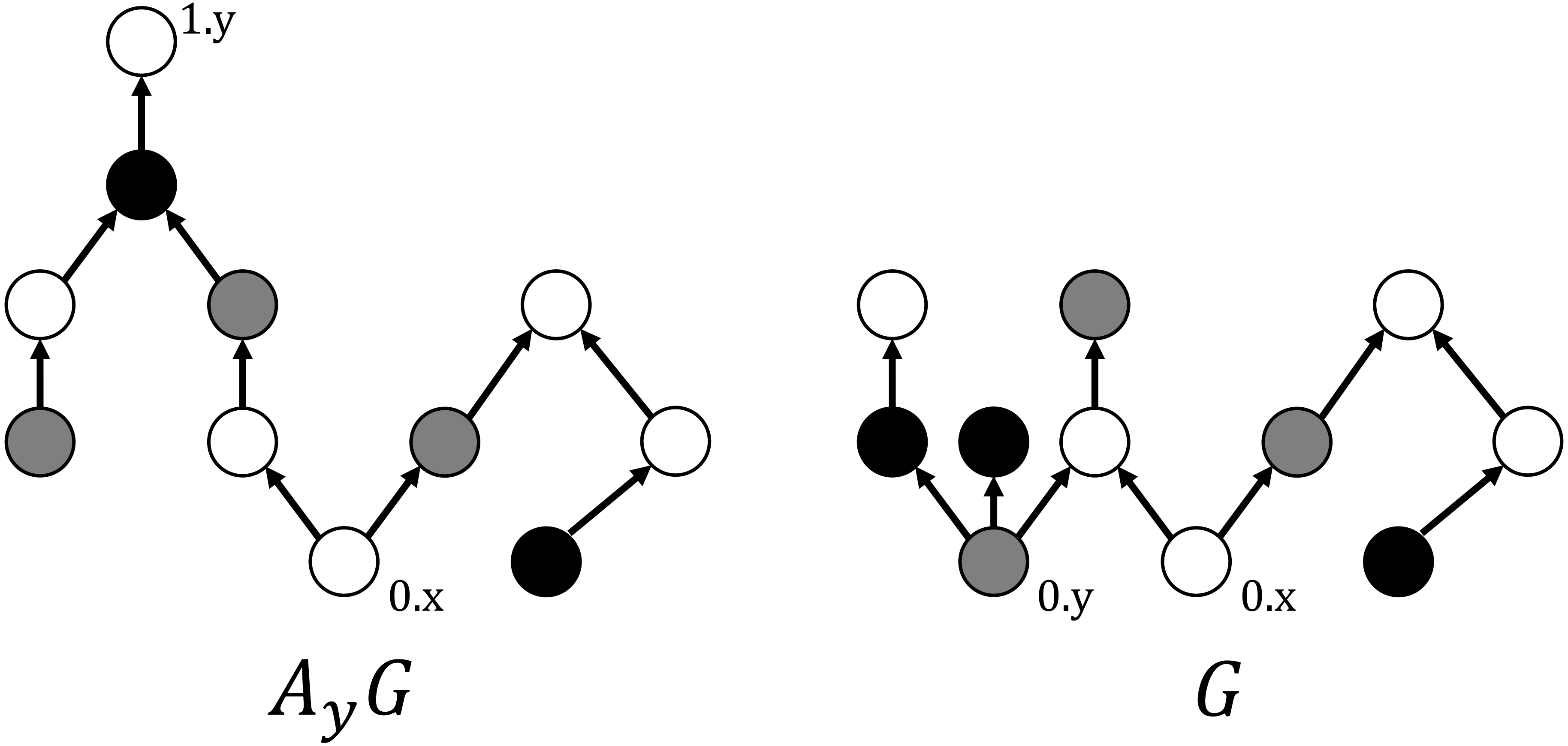
Hypothesis

1. Commutative
   $A_x A_y G = A_y A_x G$

Ensuring determinism

Goal : prove A deterministic, i.e. we always have $A_w || A_w$, which means :

Same incomming edges ⇒ Same internal state and outgoing edges



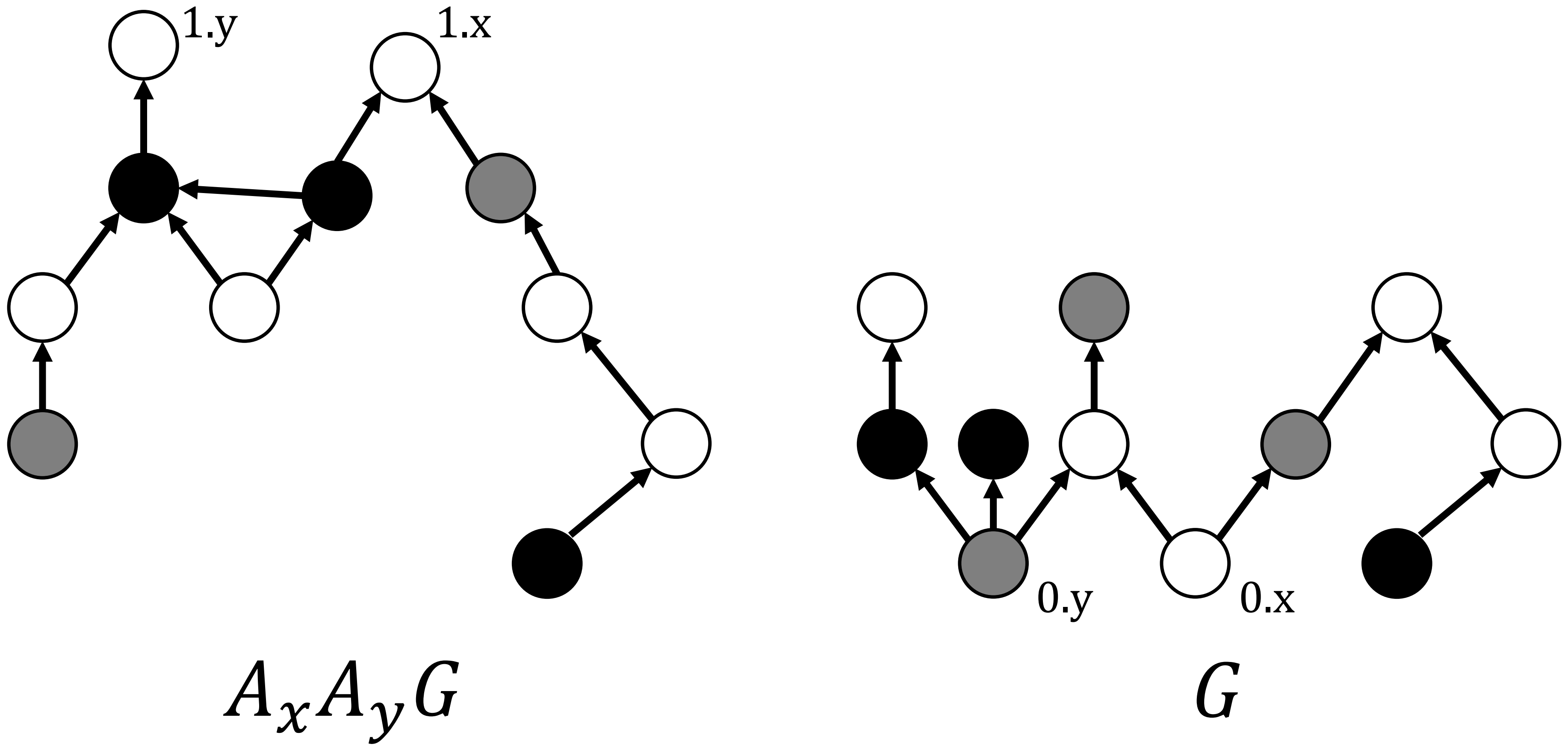
Hypothesis

1. Commutative
   $A_x A_y G = A_y A_x G$

Ensuring determinism

Goal : prove A deterministic, i.e. we always have $A_w||A_w$, which means :

Same incomming edges ⇒ Same internal state and outgoing edges



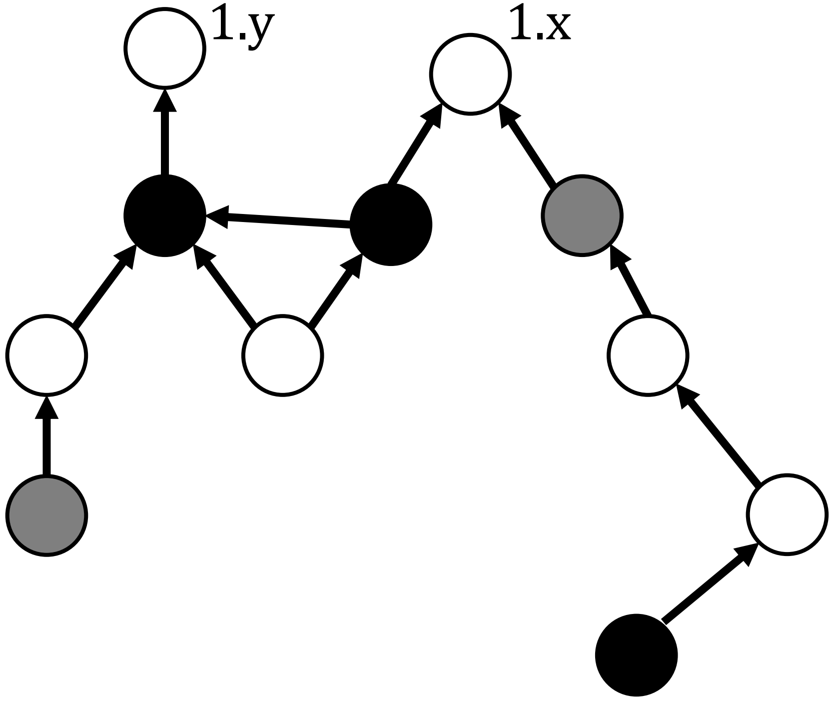
Hypothesis

1. Commutative
   $A_xA_yG = A_yA_xG$

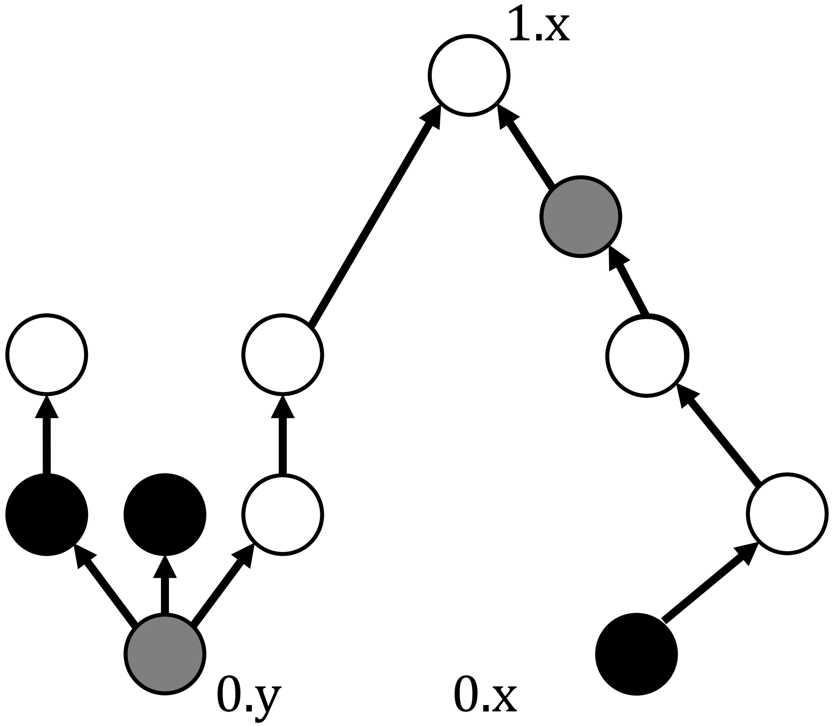
Ensuring determinism

Goal : prove A deterministic, i.e. we always have $A_w || A_w$, which means :

Same incomming edges ⇒ Same internal state and outgoing edges



$A_x A_y G$



$A_x G$

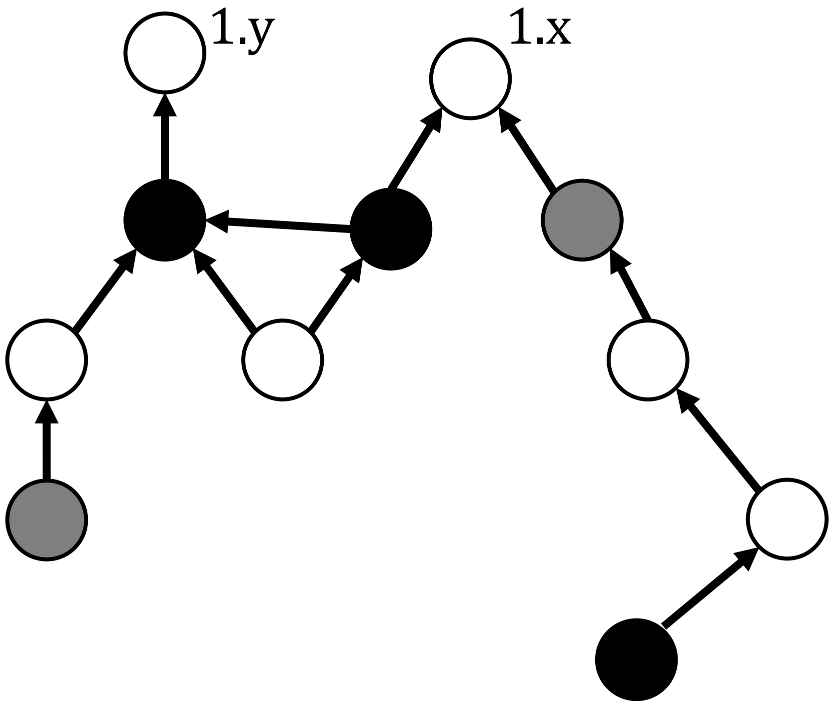
Hypothesis

1. Commutative
   $A_x A_y G = A_y A_x G$

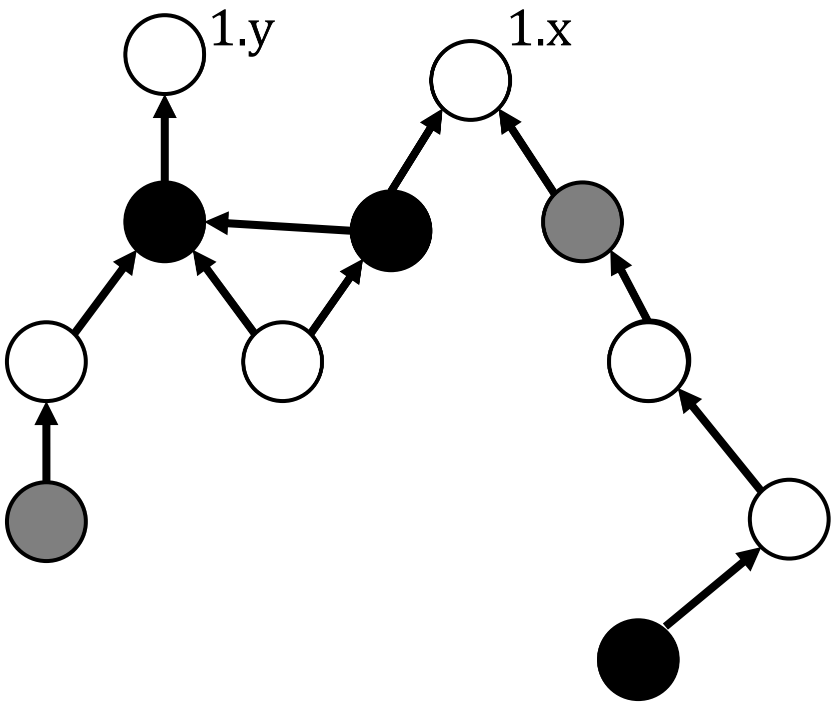
Ensuring determinism

Goal : prove A deterministic, i.e. we always have $A_w || A_w$, which means :

Same incomming edges ⇒ Same internal state and outgoing edges



$A_x A_y G$ = $A_y A_x G$



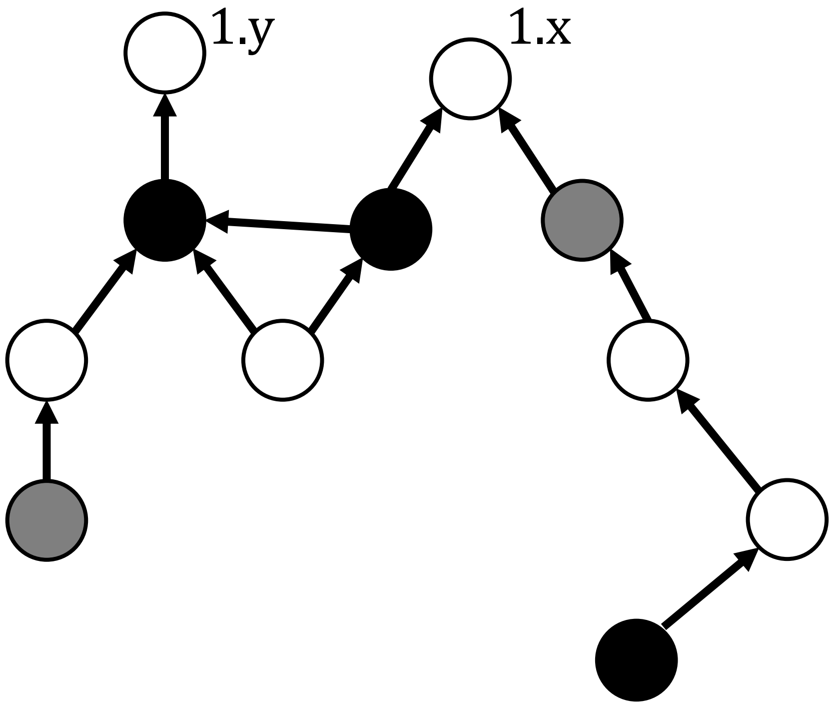
Hypothesis

1. Commutative
   $A_x A_y G = A_y A_x G$

Ensuring determinism

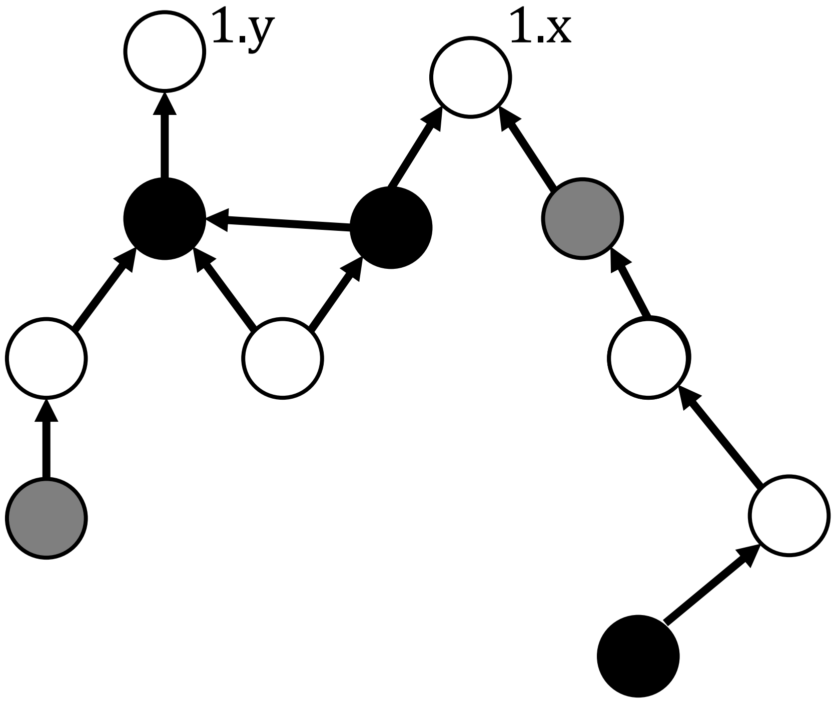
Goal : prove A deterministic, i.e. we always have $A_w || A_w$, which means :

Same incomming edges $\Rightarrow$ Same internal state and outgoing edges



$A_x A_y G$

=



$A_y A_x G$

Hypothesis

1. Commutative
   $A_x A_y G = A_y A_x G$

$\Downarrow$

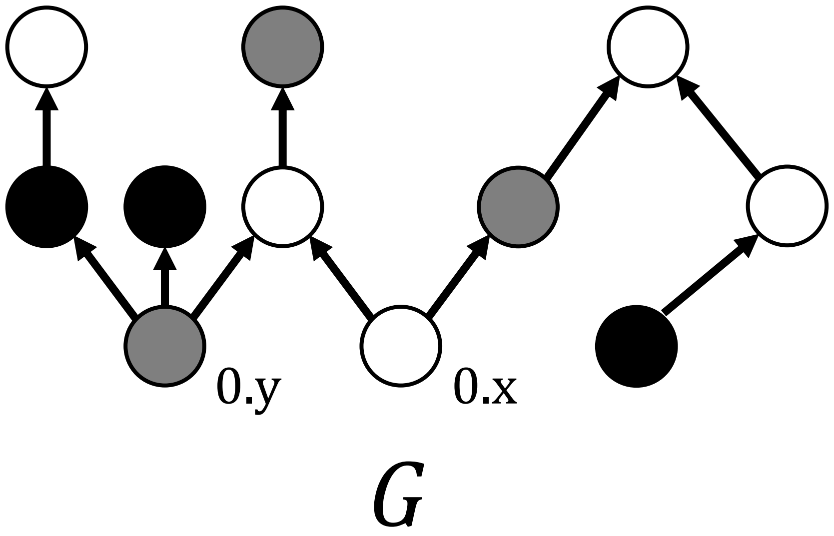
Conclusion

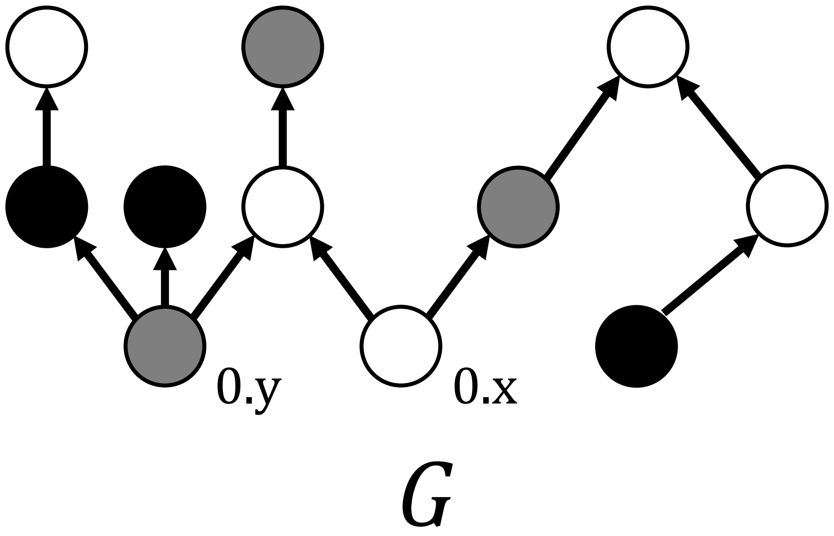
1. $A_x A_y G \,||\, A_y A_x G$

Ensuring determinism

Goal : prove A deterministic, i.e. we always have $A_w || A_w$, which means :

Same incomming edges $\Rightarrow$ Same internal state and outgoing edges





Hypothesis

1. Commutative
   $A_x A_y G = A_y A_x G$

$\Downarrow$

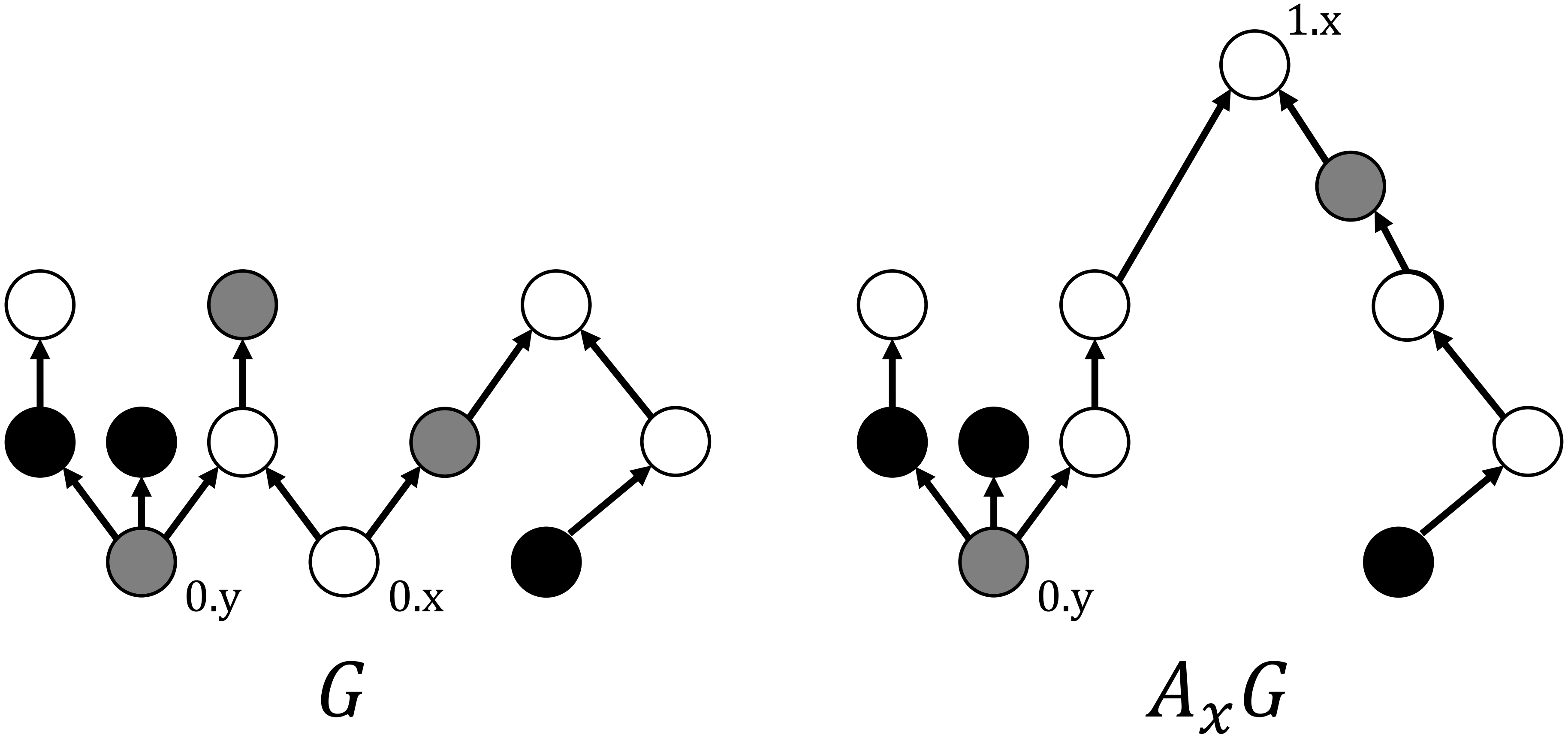
Conclusion

1. $A_x A_y G \mid\mid A_y A_x G$

Ensuring determinism

Goal : prove A deterministic, i.e. we always have $A_w || A_w$, which means :

Same incomming edges ⇒ Same internal state and outgoing edges



Hypothesis

1. Commutative
   $A_x A_y G = A_y A_x G$

⇓

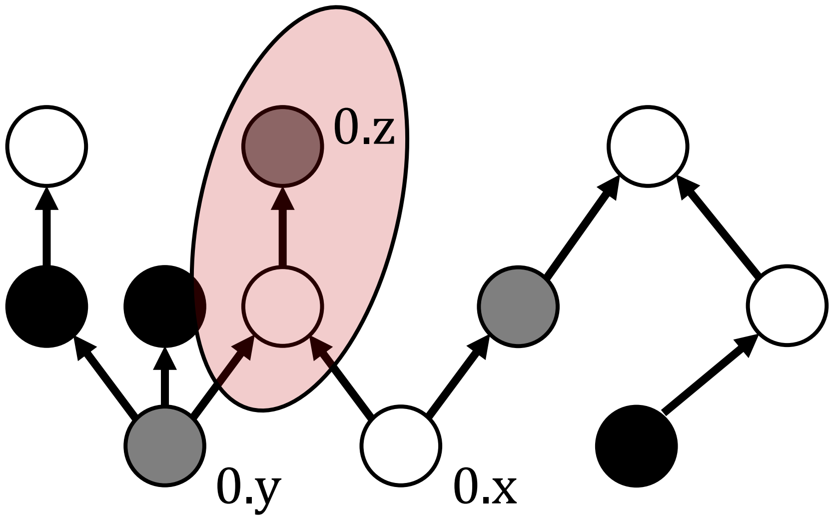
Conclusion

1. $A_x A_y G \,||\, A_y A_x G$

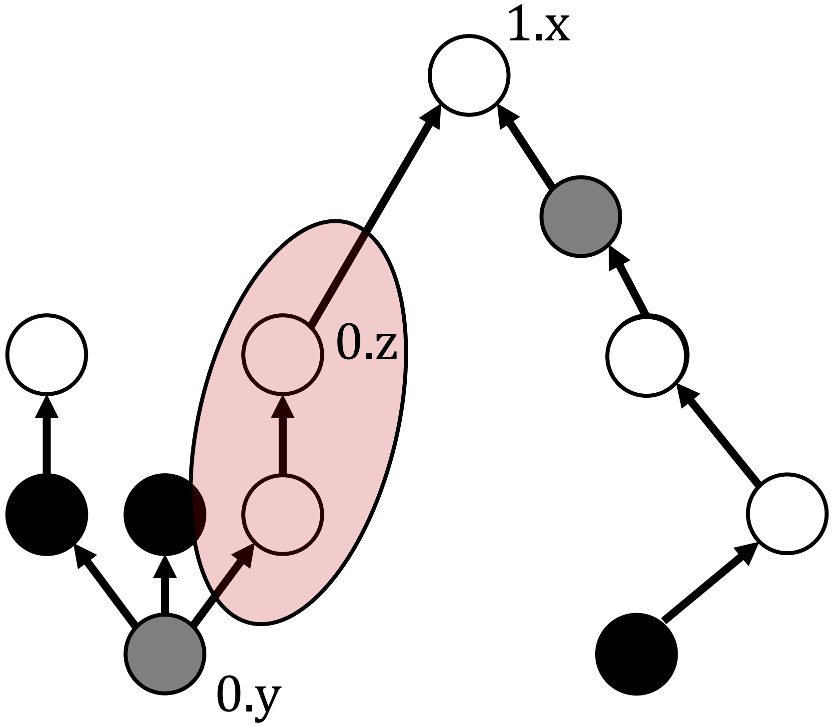
Ensuring determinism

Goal : prove A deterministic, i.e. we always have $A_w || A_w$, which means :

Same incomming edges $\Rightarrow$ Same internal state and outgoing edges



G



$A_x G$

Hypothesis

1. Commutative $A_x A_y G = A_y A_x G$

$\Downarrow$

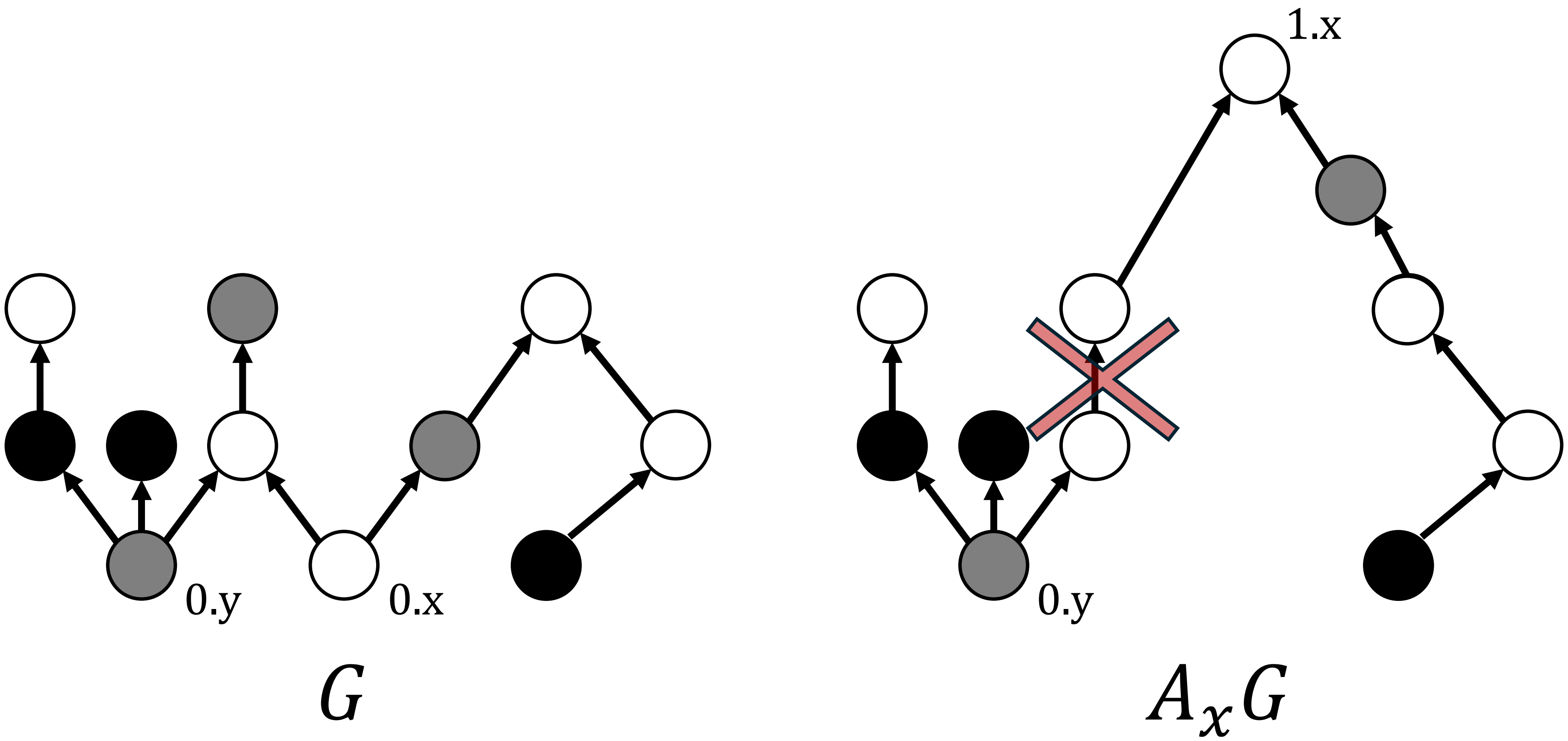
Conclusion

1. $A_x A_y G \mid\mid A_y A_x G$

Ensuring determinism

Goal : prove A deterministic, i.e. we always have $A_w || A_w$, which means :

Same incomming edges $\Rightarrow$ Same internal state and outgoing edges



Hypothesis

1. Commutative
   $A_x A_y G = A_y A_x G$
2. Edge decreasing

$\Downarrow$

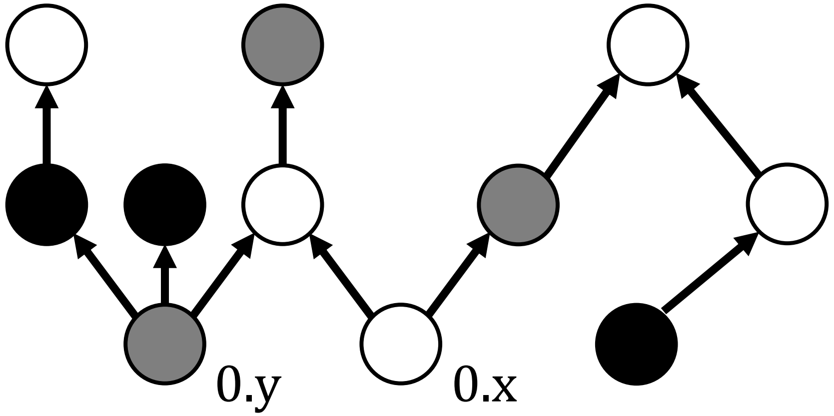
Conclusion

1. $A_x A_y G \,||\, A_y A_x G$

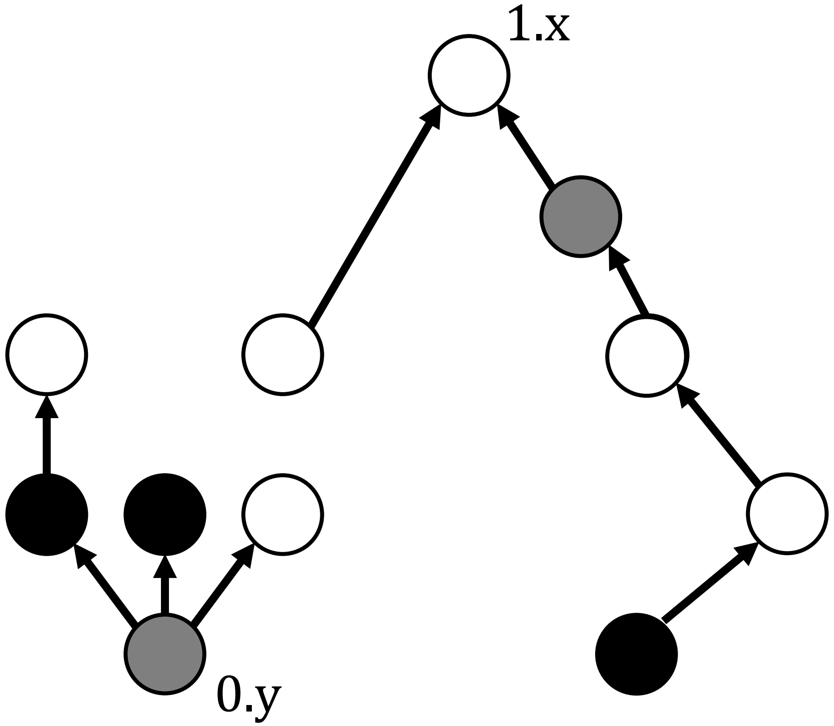
Ensuring determinism

Goal : prove A deterministic, i.e. we always have $A_w || A_{w'}$ which means :

~~Same incomming edges~~ $\Rightarrow$ Same internal state and outgoing edges



G



$A_x G$

Hypothesis

1. Commutative
   $A_x A_y G = A_y A_x G$
2. Edge decreasing

$\Downarrow$

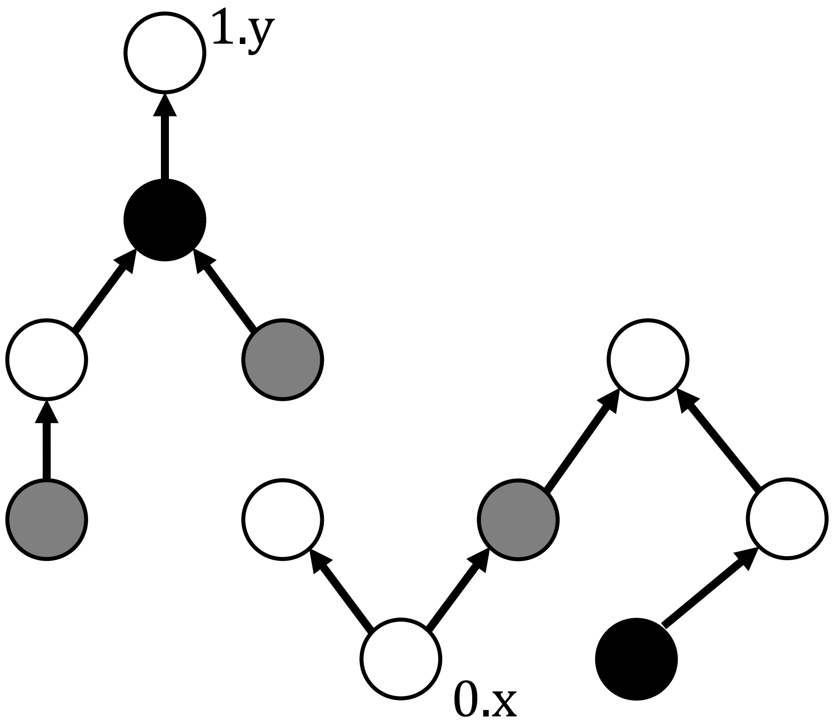
Conclusion

1. $A_x A_y G \,||\, A_y A_x G$
2. $A_x G \,||\, G$

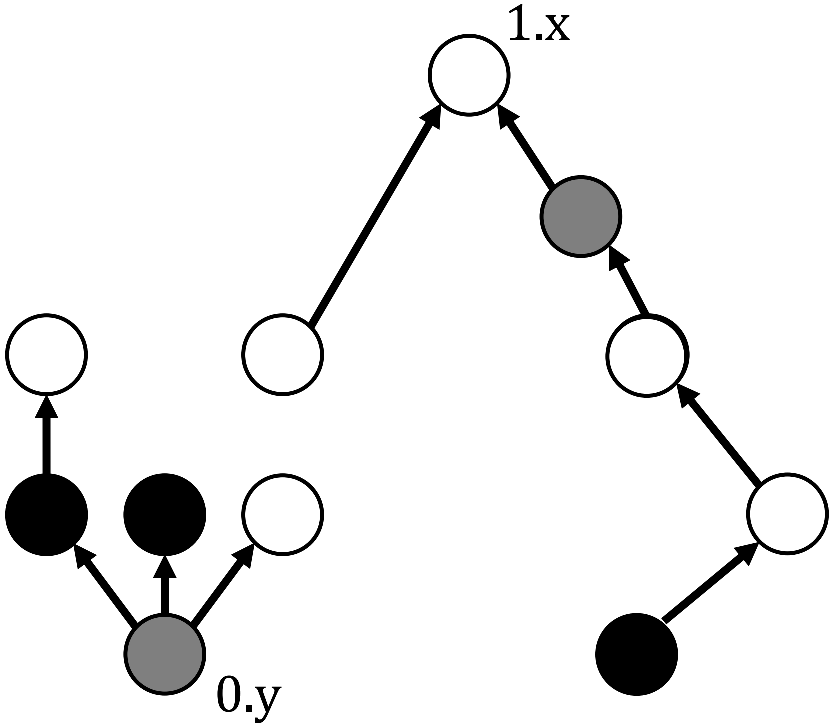
Ensuring determinism

Goal : prove A deterministic, i.e. we always have $A_w || A_w$, which means :

Same incomming edges ⇒ Same internal state and outgoing edges



$A_y G$



$A_x G$

Hypothesis

1. Commutative
   $A_x A_y G = A_y A_x G$
2. Edge decreasing

⇓

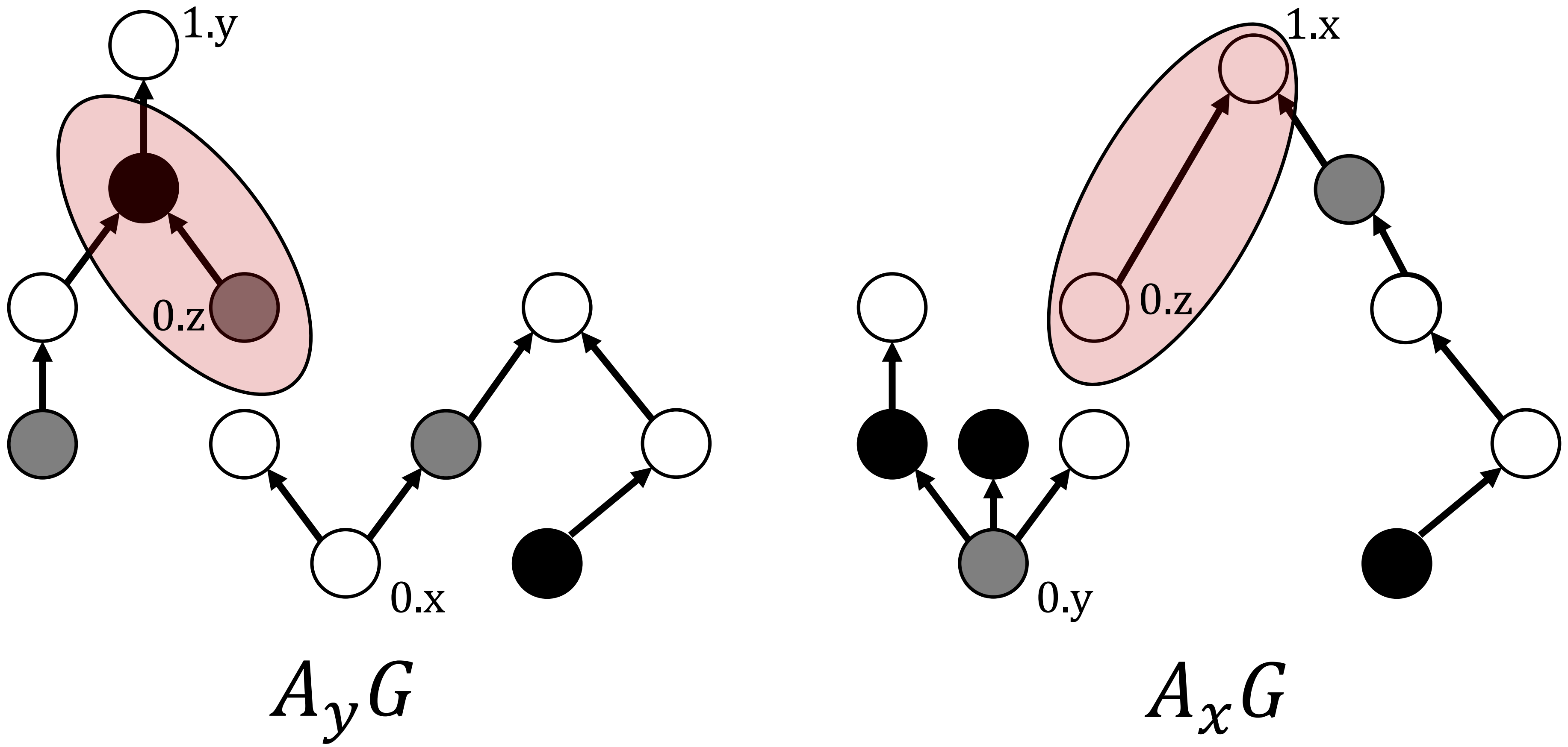
Conclusion

1. $A_x A_y G \; || \; A_y A_x G$
2. $A_x G \; || \; G$

Ensuring determinism

Goal : prove A deterministic, i.e. we always have $A_w || A_w$, which means :

Same incomming edges $\Rightarrow$ Same internal state and outgoing edges



Hypothesis

1. Commutative
   $A_x A_y G = A_y A_x G$
2. Edge decreasing

$\Downarrow$

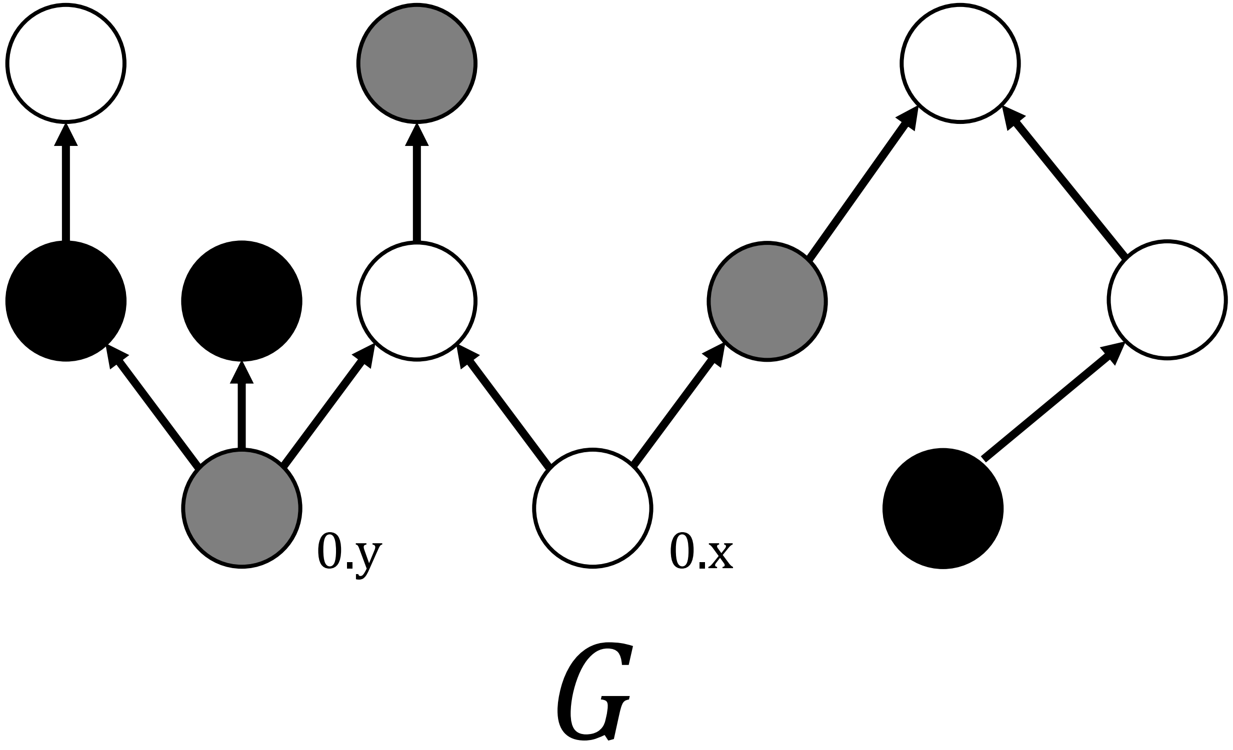
Conclusion

1. $A_x A_y G \,||\, A_y A_x G$
2. $A_x G \,||\, G$

Ensuring determinism

Goal : prove A deterministic, i.e. we always have $A_w || A_w$, which means :

Same incomming edges ⇒ Same internal state and outgoing edges



Hypothesis

1. Commutative $A_x A_y G = A_y A_x G$
2. Edge decreasing

⇓

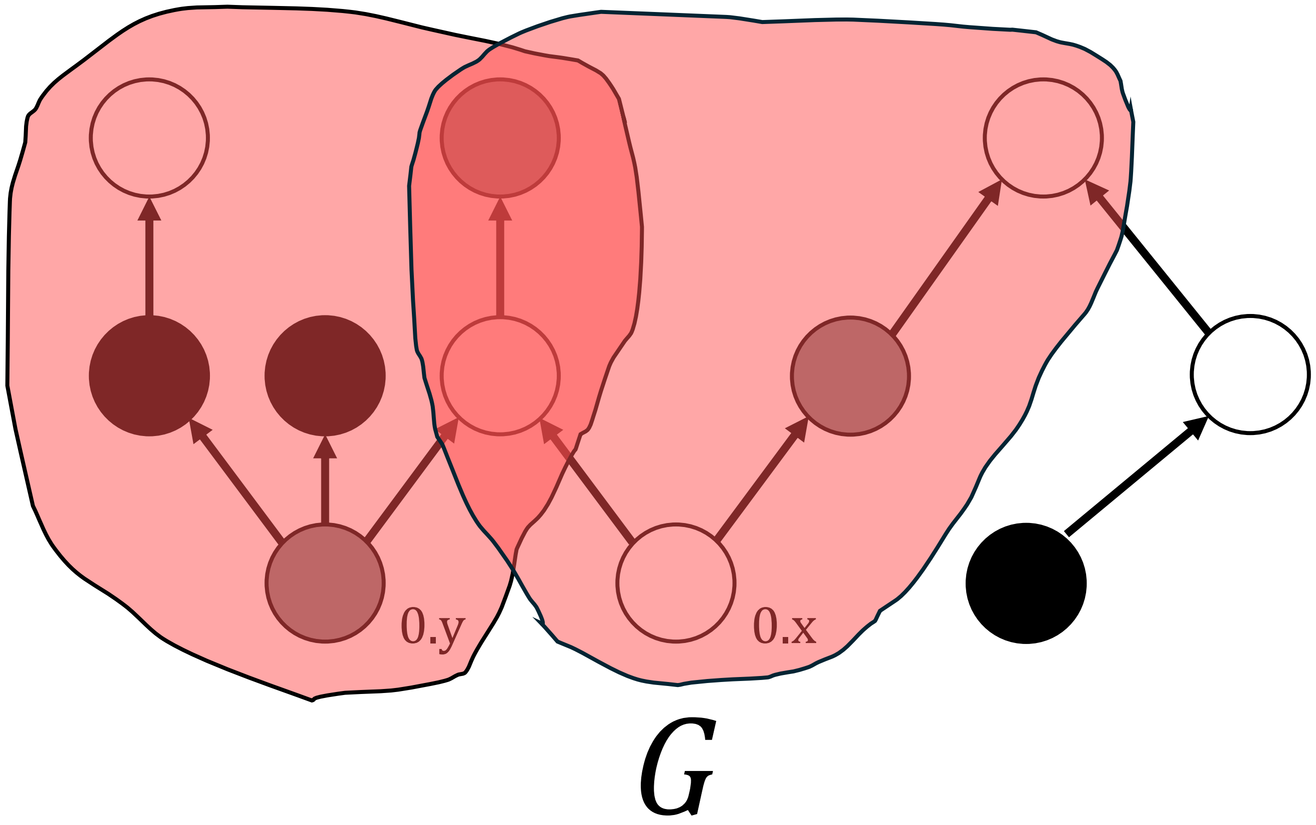
Conclusion

1. $A_x A_y G \,||\, A_y A_x G$
2. $A_x G \,||\, G$

Ensuring determinism

Goal : prove A deterministic, i.e. we always have $A_w || A_w$, which means :

Same incomming edges ⇒ Same internal state and outgoing edges



Hypothesis

1. Commutative $A_x A_y G = A_y A_x G$
2. Edge decreasing

⇓

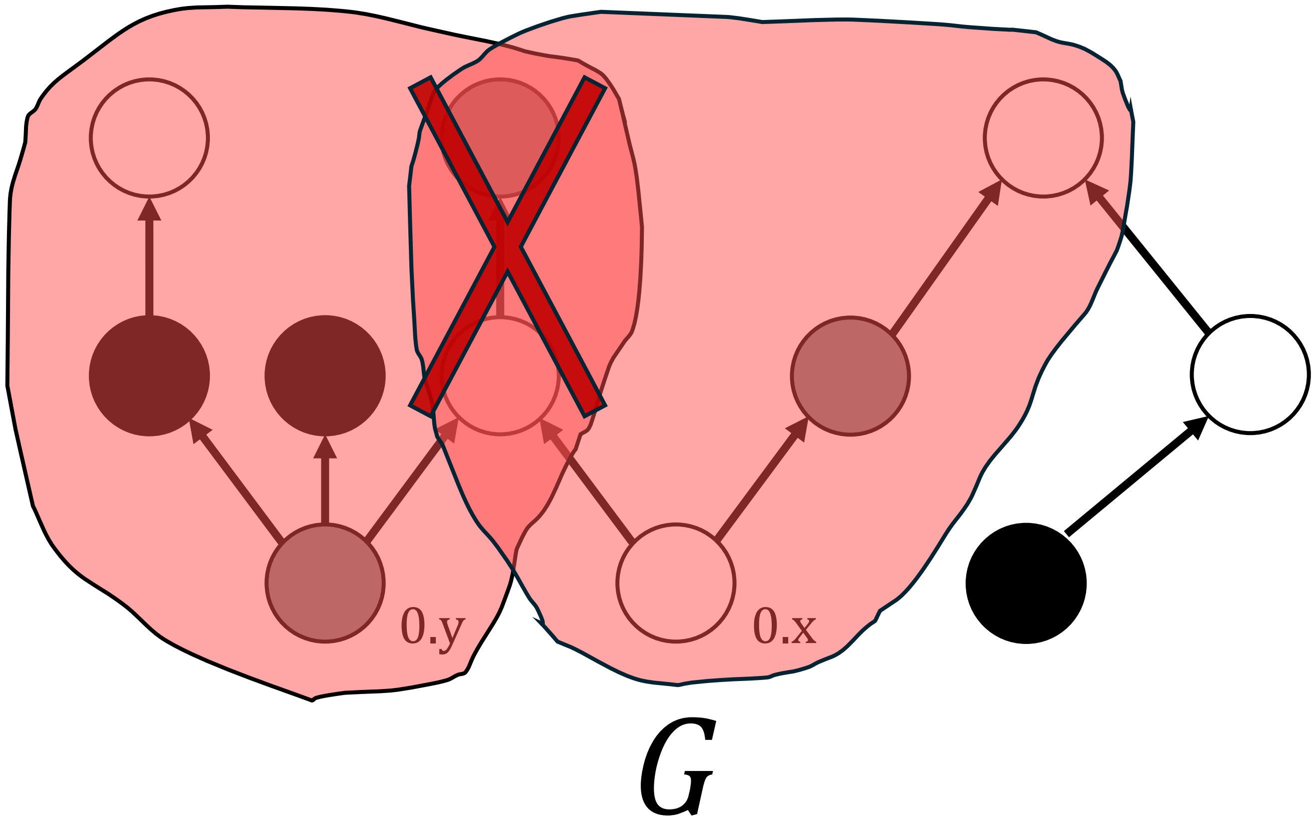
Conclusion

1. $A_x A_y G \,||\, A_y A_x G$
2. $A_x G \,||\, G$

Ensuring determinism

Goal : prove A deterministic, i.e. we always have $A_w || A_w$, which means :

Same incomming edges ⇒ Same internal state and outgoing edges



Hypothesis

1. Commutative $A_x A_y G = A_y A_x G$
2. Edge decreasing
3. Private

⇓

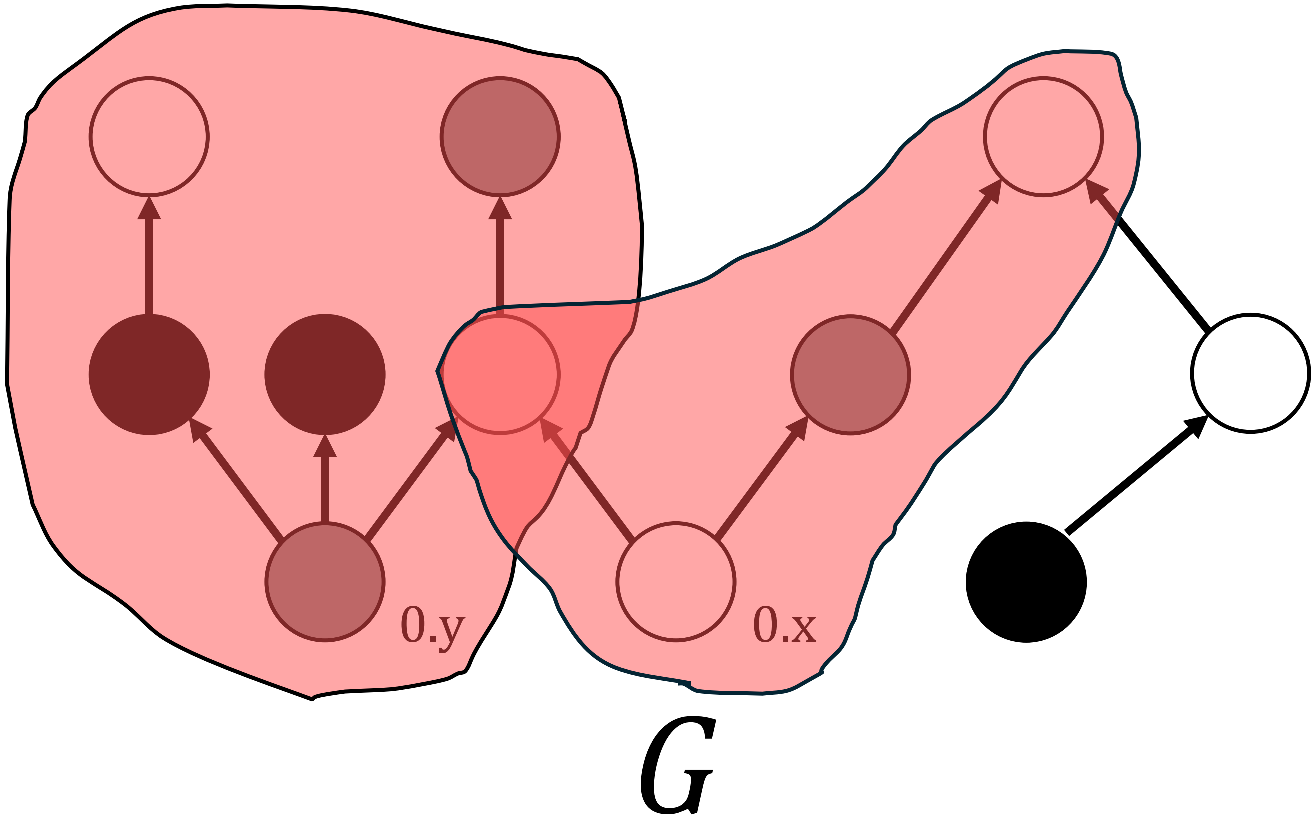
Conclusion

1. $A_x A_y G \,||\, A_y A_x G$
2. $A_x G \,||\, G$

Ensuring determinism

Goal : prove A deterministic, i.e. we always have $A_w || A_w$, which means :

Same incomming edges ⇒ Same internal state and outgoing edges



Hypothesis

1. Commutative
   $A_x A_y G = A_y A_x G$
2. Edge decreasing
3. Private

⇓

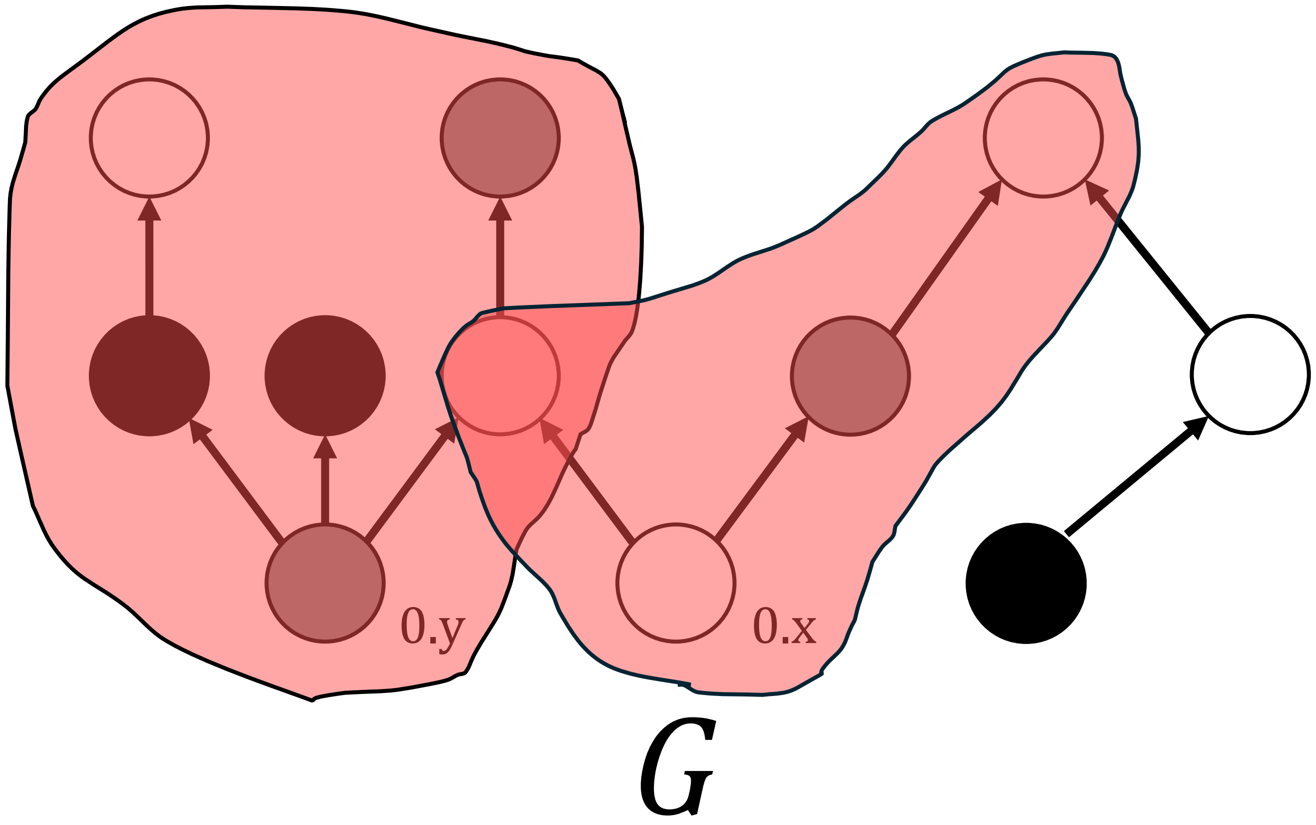
Conclusion

1. $A_x A_y G \, || \, A_y A_x G$
2. $A_x G \, || \, G$
3. $A_x G \, || \, A_y G$

Ensuring determinism

Goal : prove A deterministic, i.e. we always have $A_w || A_{w'}$ which means :

Same incomming edges ⇒ Same internal state and outgoing edges



Hypothesis

1. Commutative
   $A_x A_y G = A_y A_x G$
2. Edge decreasing
3. Private

⇓

Conclusion

1. $A_x A_y G \; || \; A_y A_x G$
2. $A_x G \; || \; G$
3. $A_x G \; || \; A_y G$
4. $A_w G \; || \; A_{w'} G$

Ensuring determinism

Theorem 1

Any commutative, edge decreasing and private local rule is deterministic.

Ensuring determinism

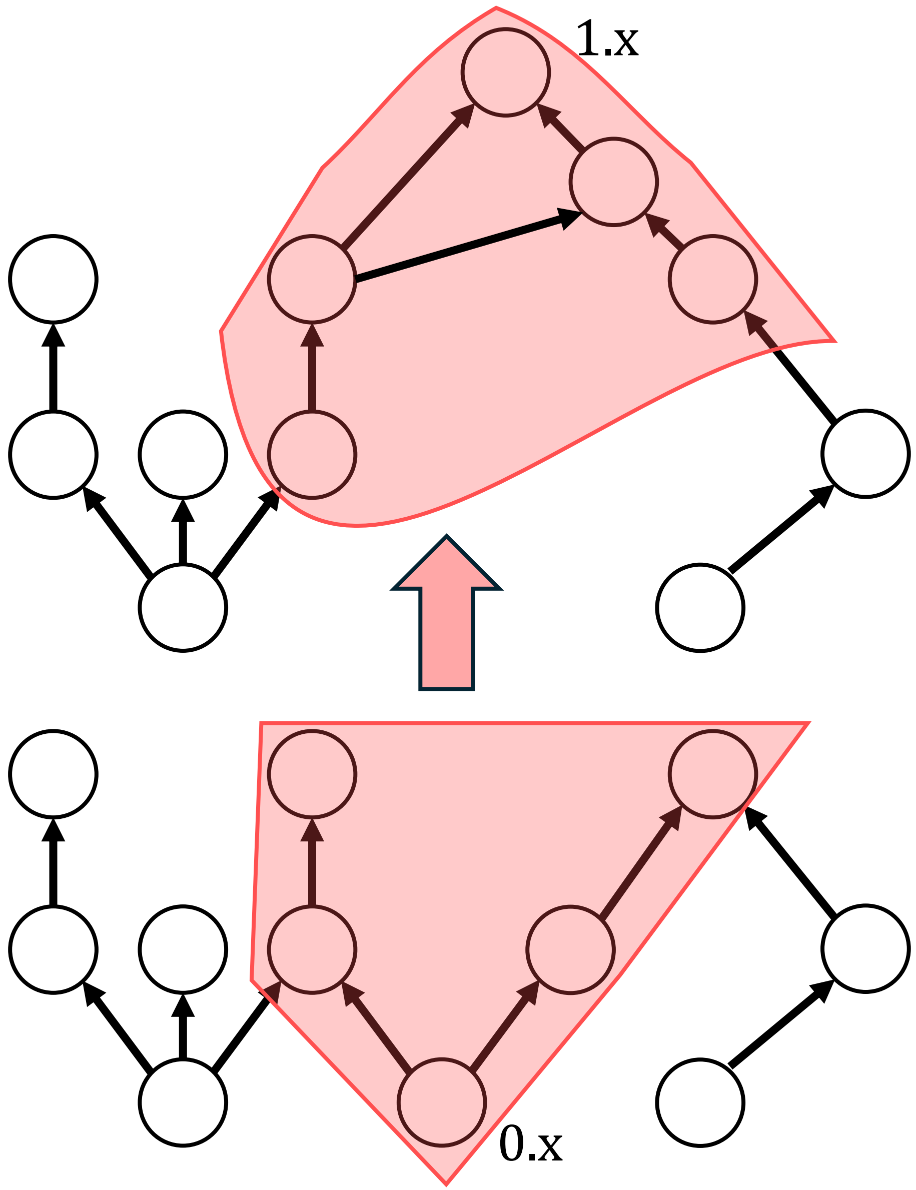
Theorem 1

Any commutative, edge decreasing and private local rule is deterministic.

In general which local rewriting rules are physical ?

- **Determinism** **OK**
- **Reversibility** 

Defining Reversibility

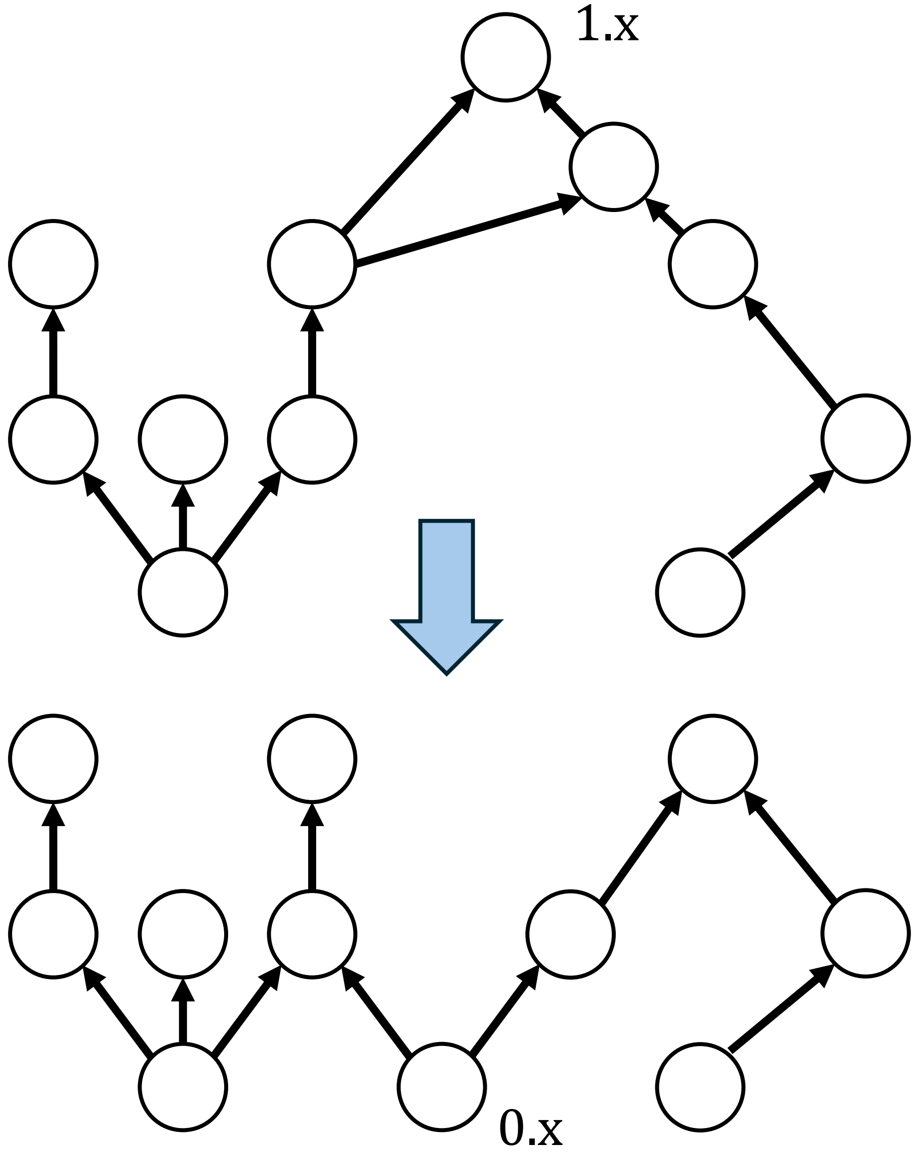


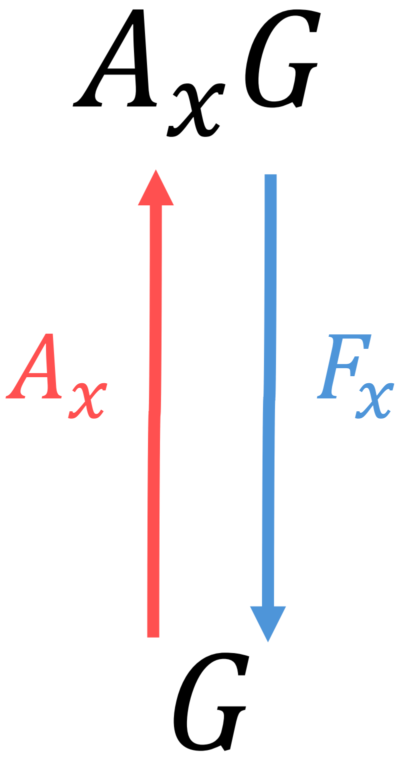
$$A_x G$$

A_x

$$G$$

Defining Reversibility

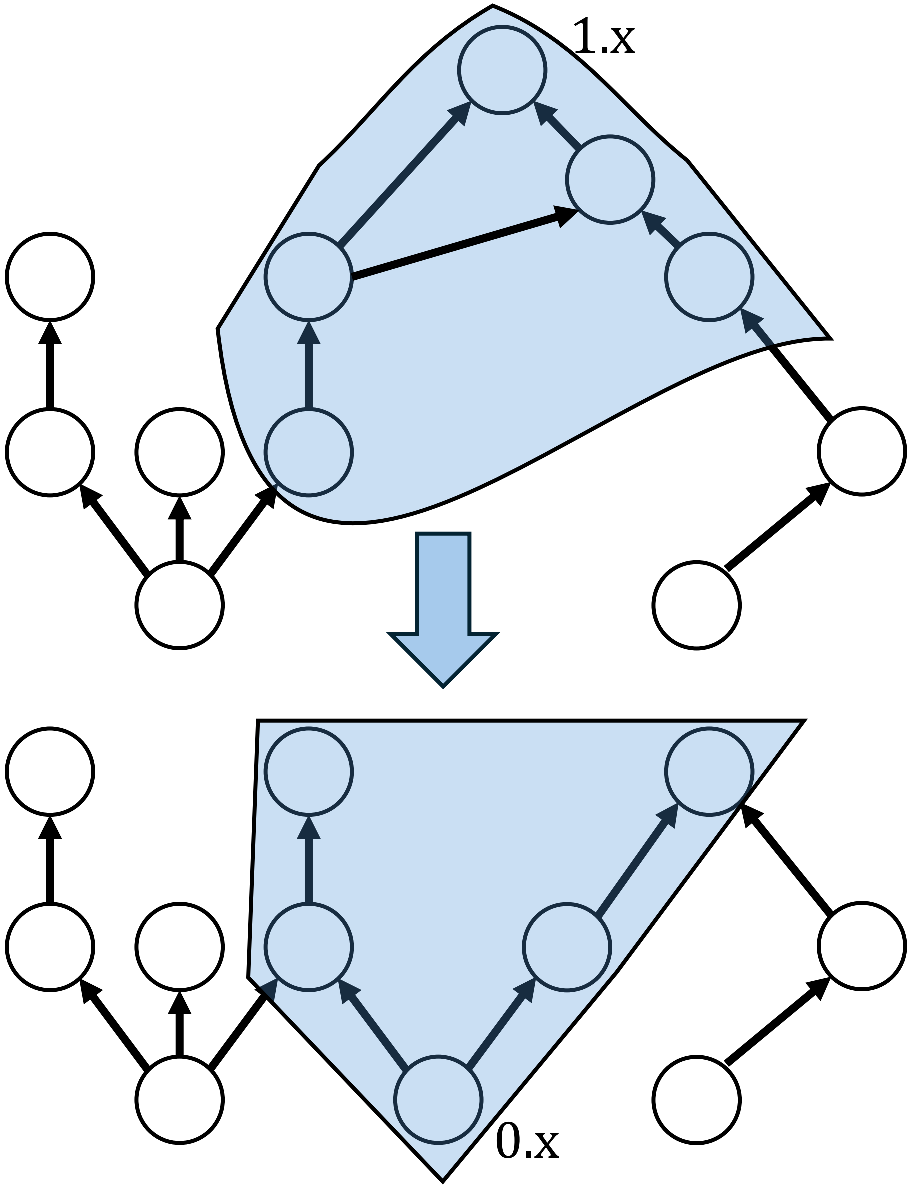


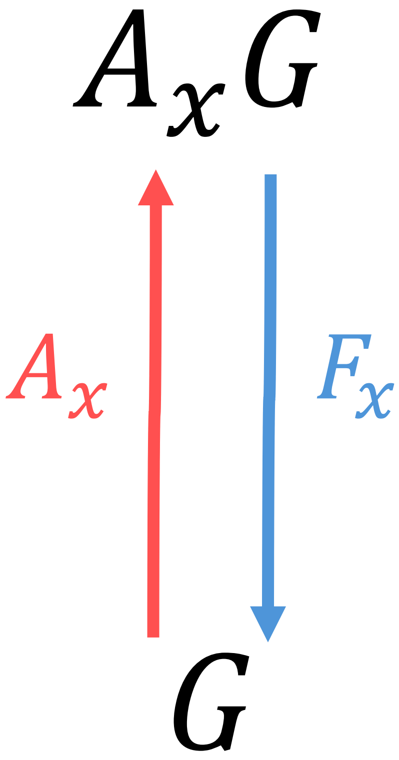


Option 1

- F is a **function s.t.** $F_x A_x G = G$
- **Problem :** non physical, does not match reversibility in CA, …

Defining Reversibility





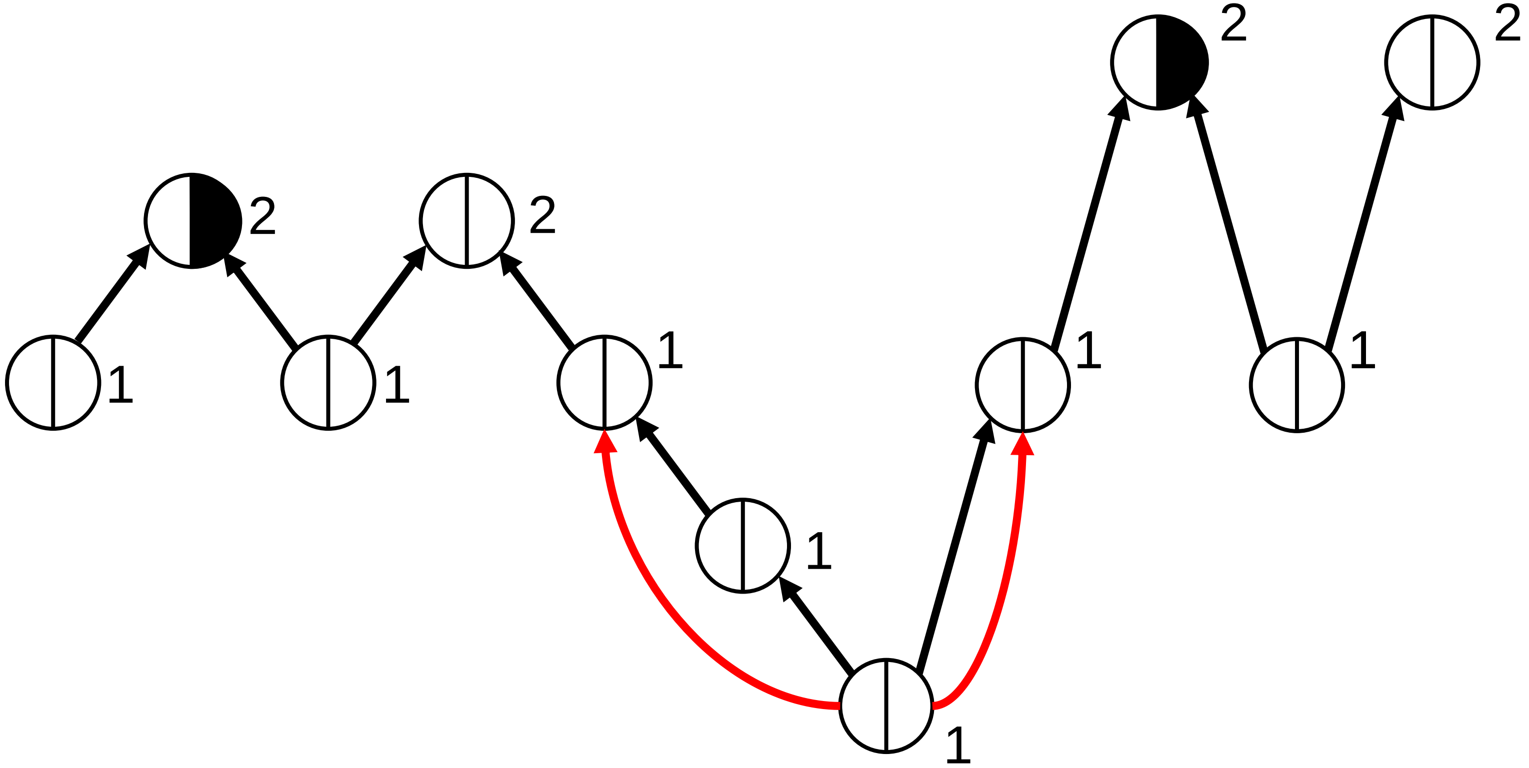
Option 1

- F is a **function s.t.** $F_x A_x G = G$
- **Problem :** non physical, does not match reversibility in CA, …

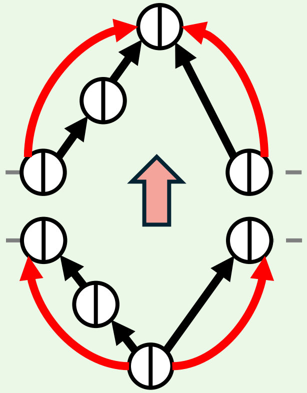
Option 2

- F is a **local rule** s.t. $F_x A_x G = G$
- Then it must be s.t. $A_x F_x G' = G'$

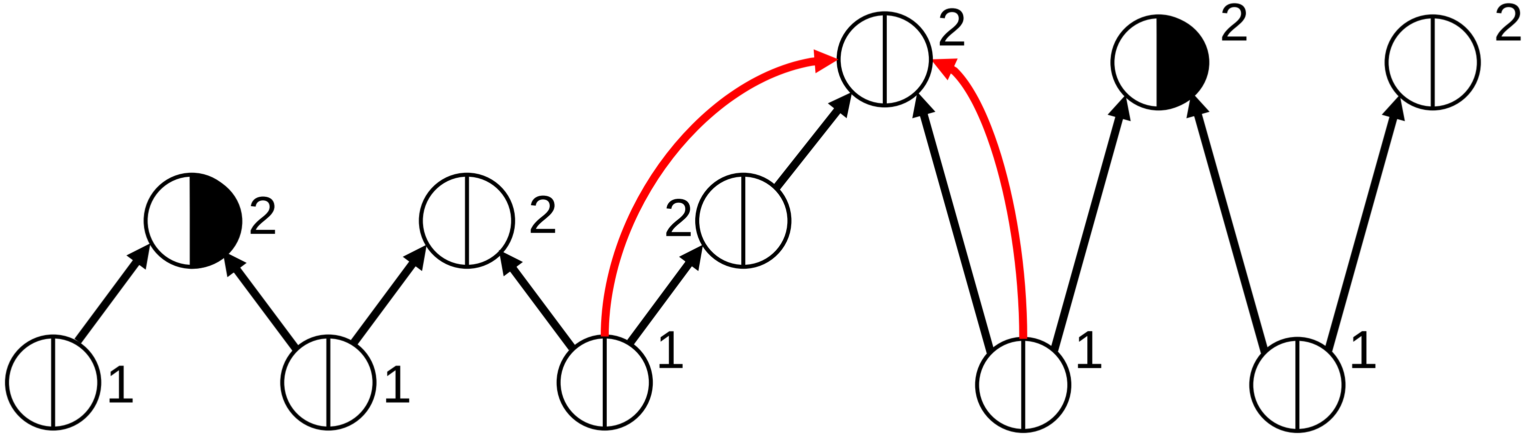
Reversible time dilation



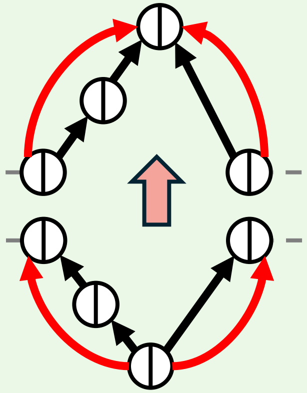
Local rule



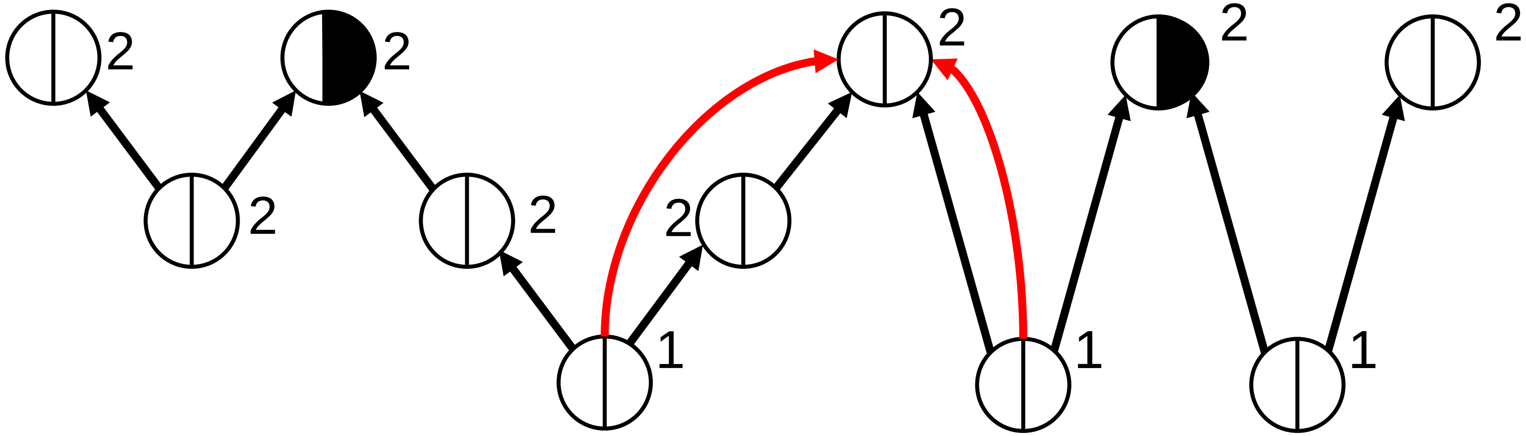
Reversible time dilation



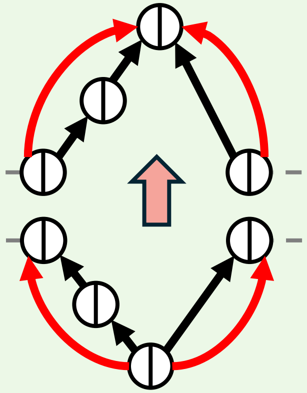
Local rule



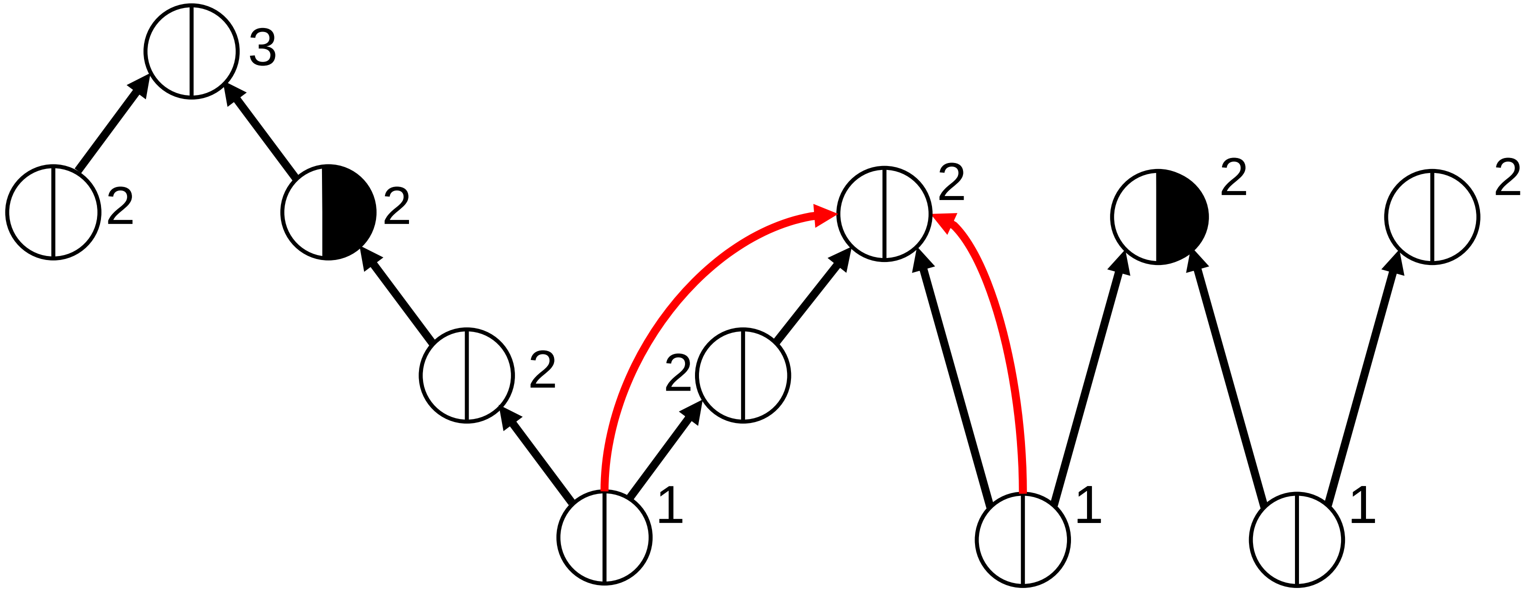
Reversible time dilation



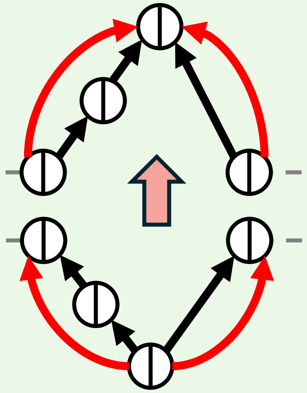
Local rule



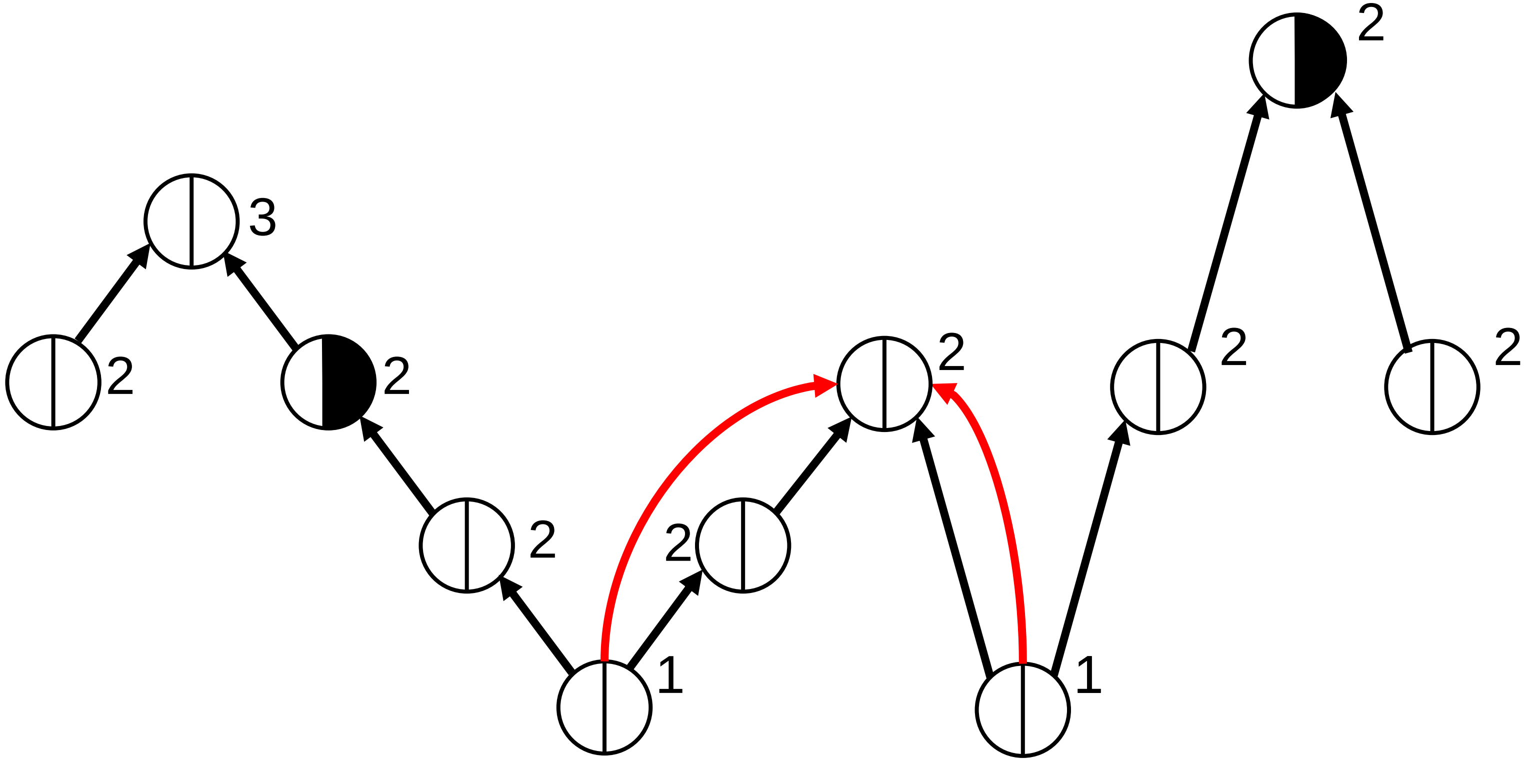
Reversible time dilation



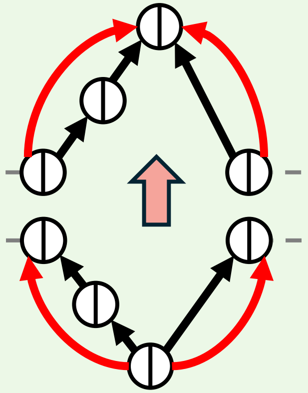
Local rule



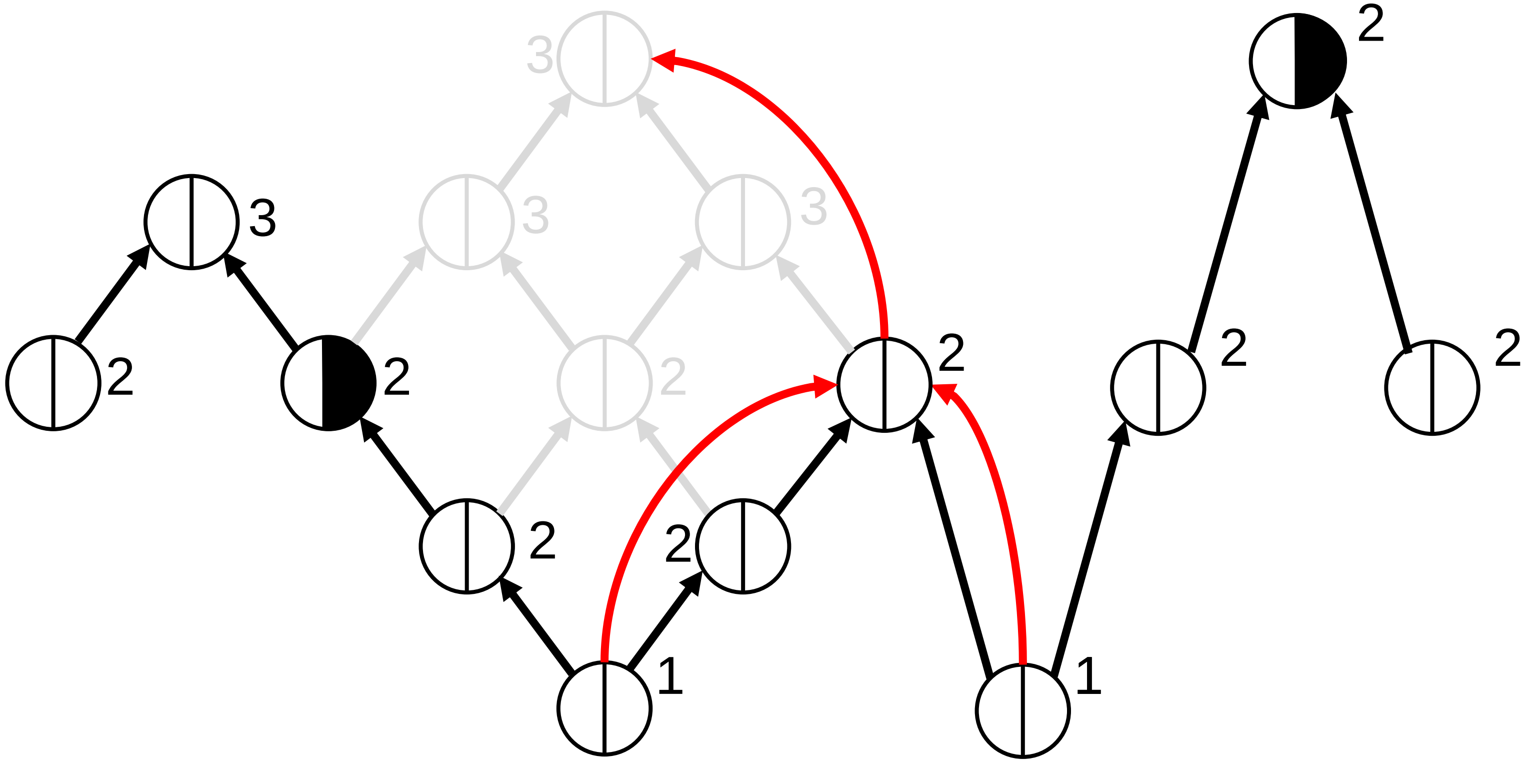
Reversible time dilation



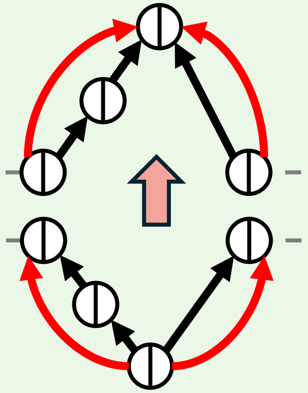
Local rule



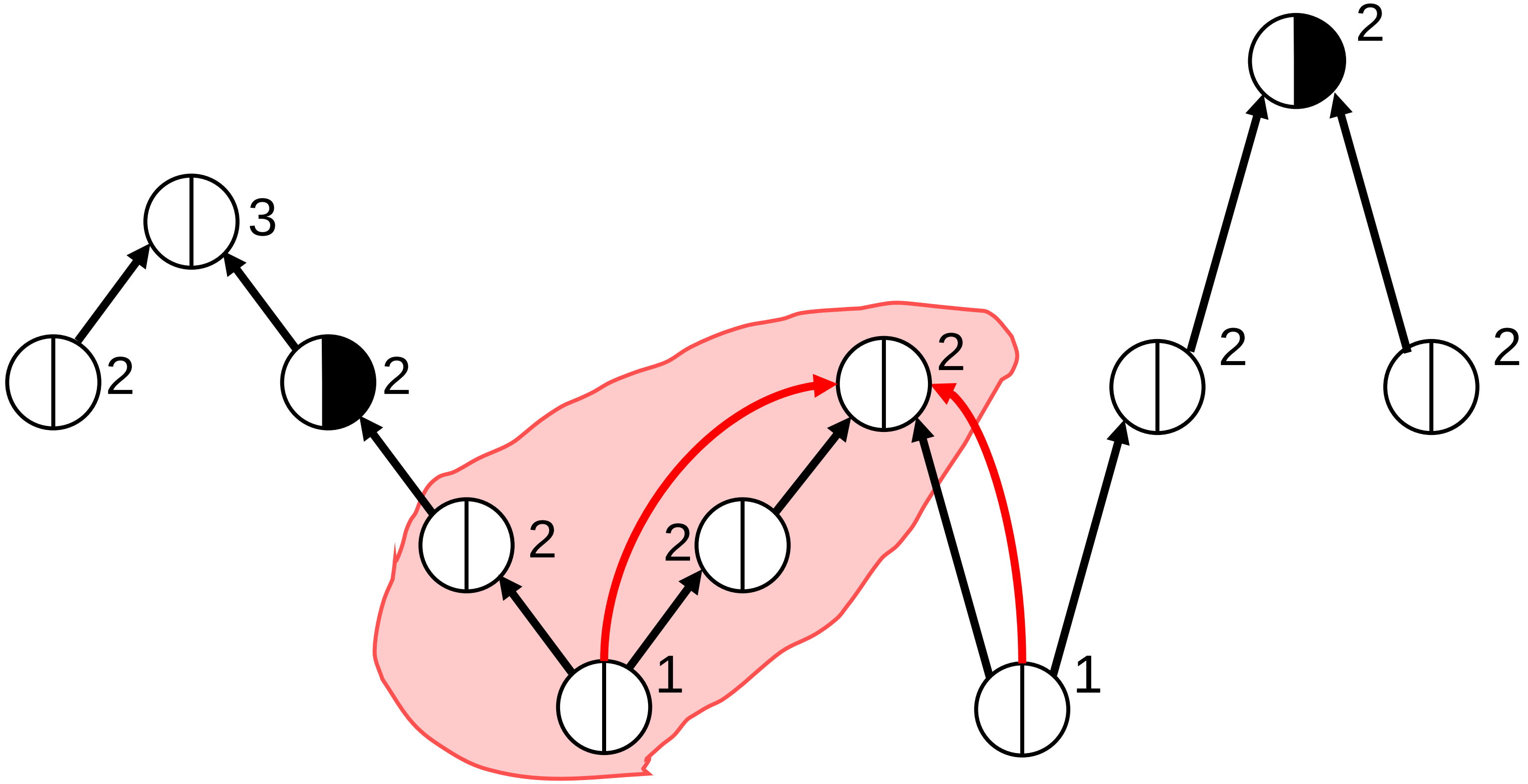
Reversible time dilation



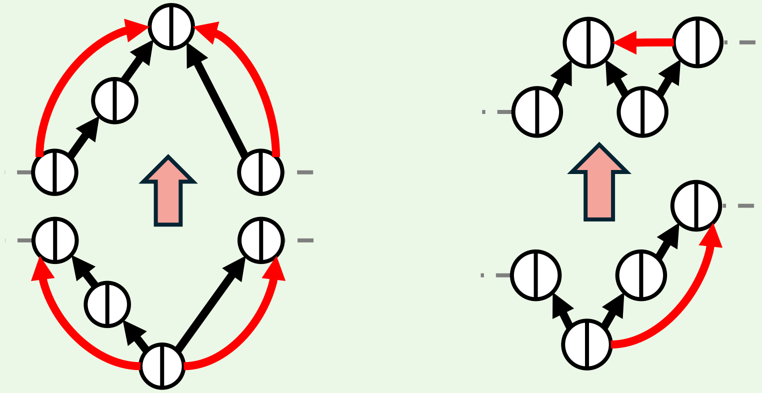
Local rule



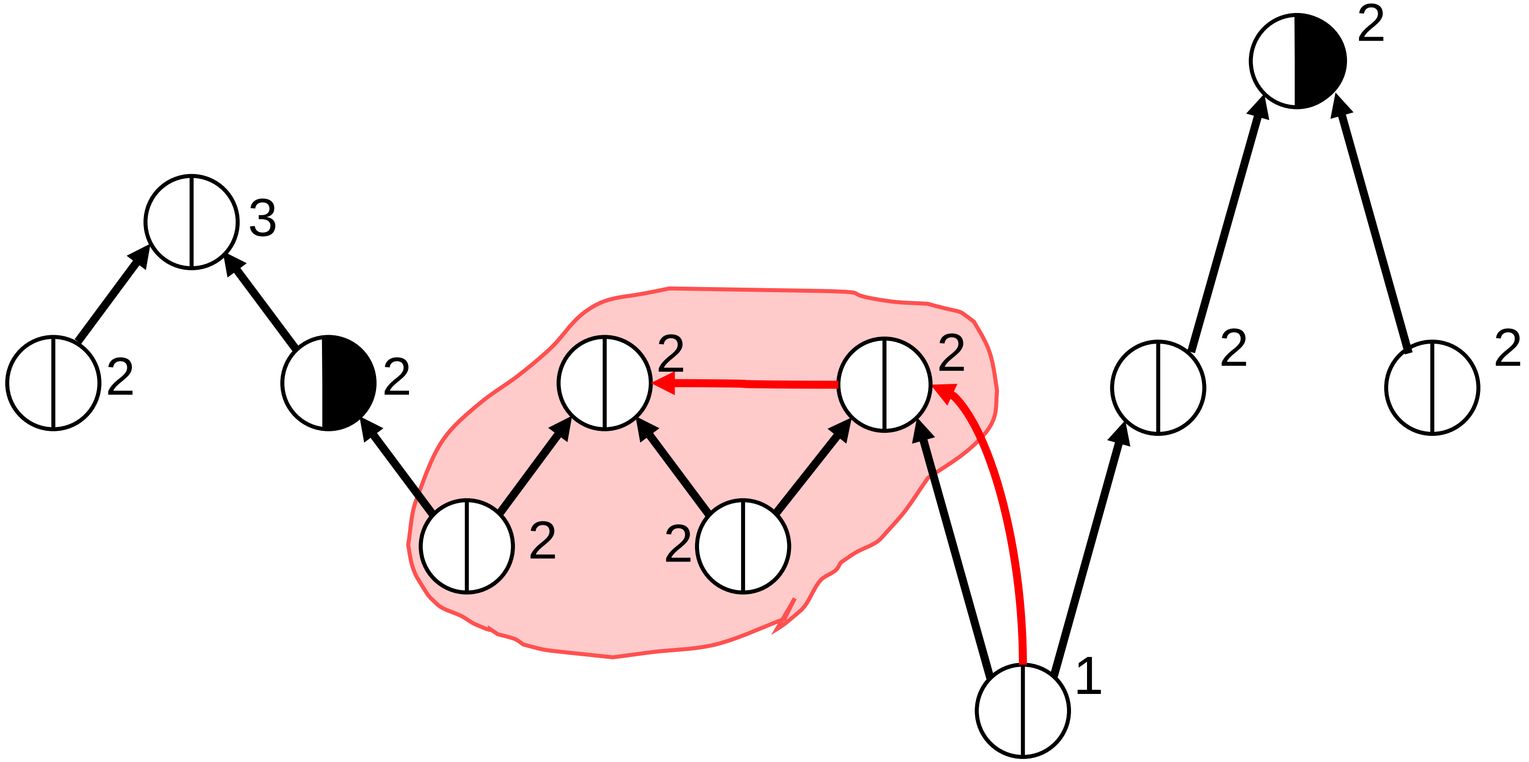
Reversible time dilation



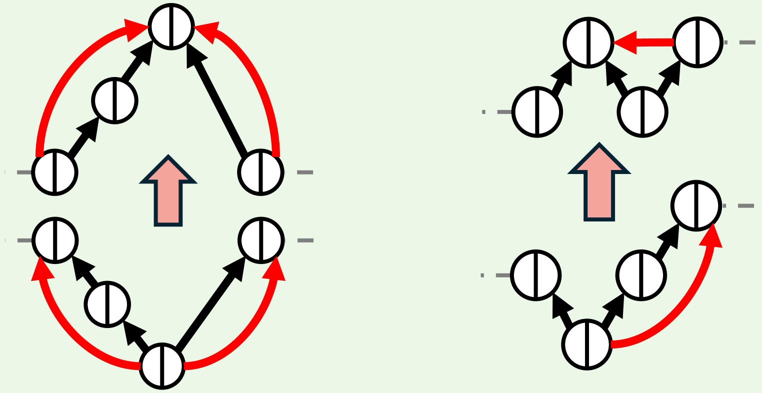
Local rule



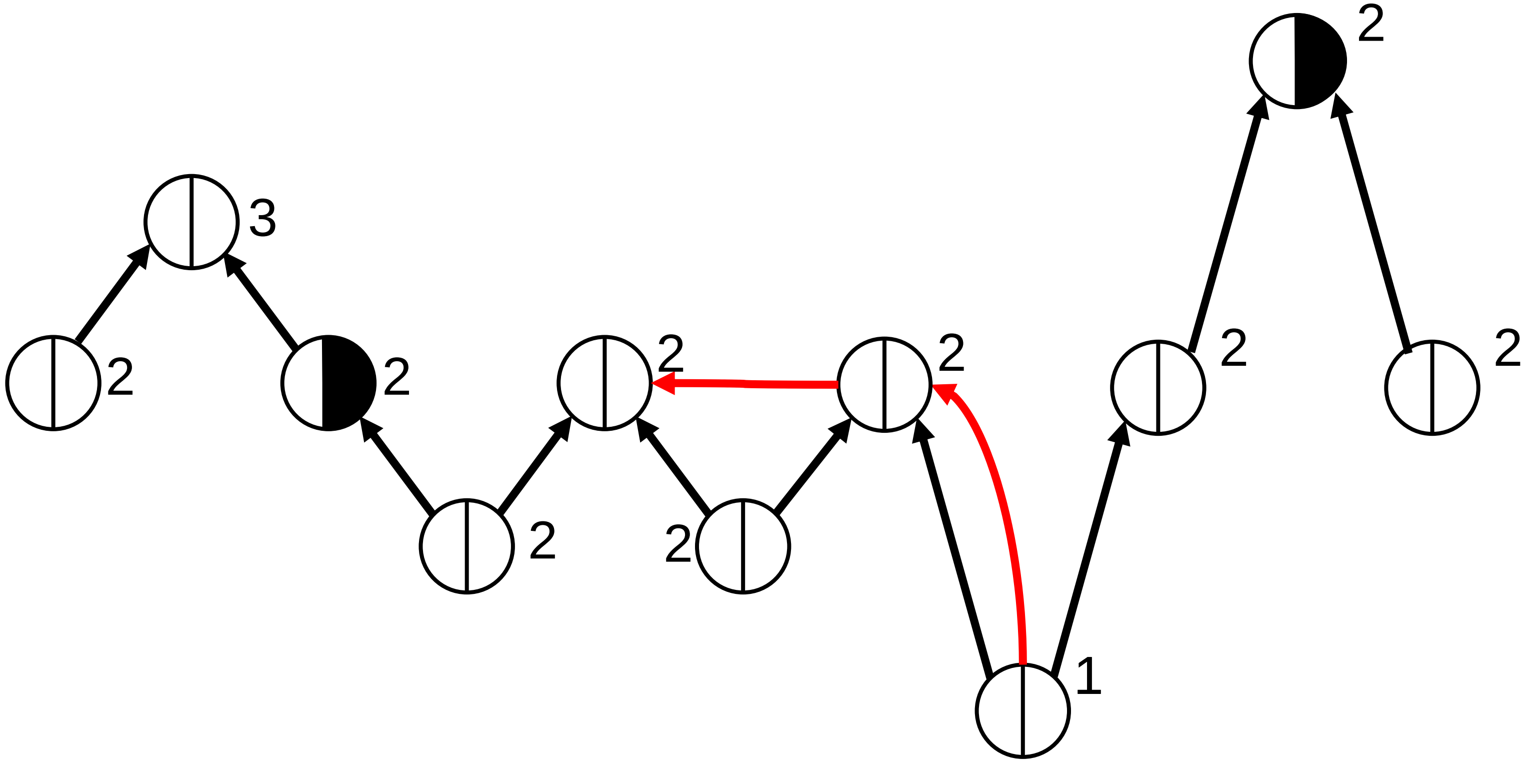
Reversible time dilation



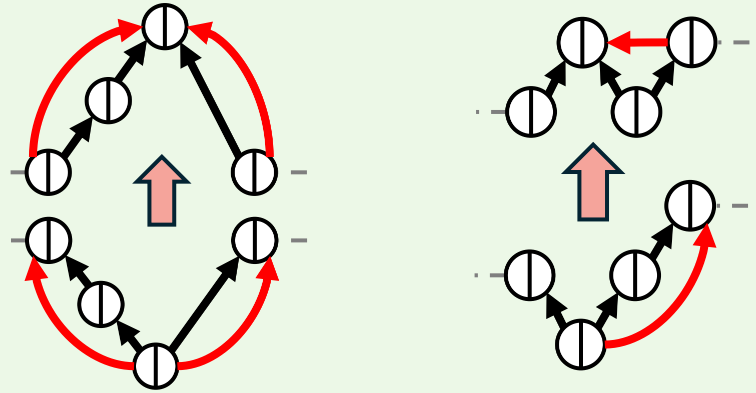
Local rule



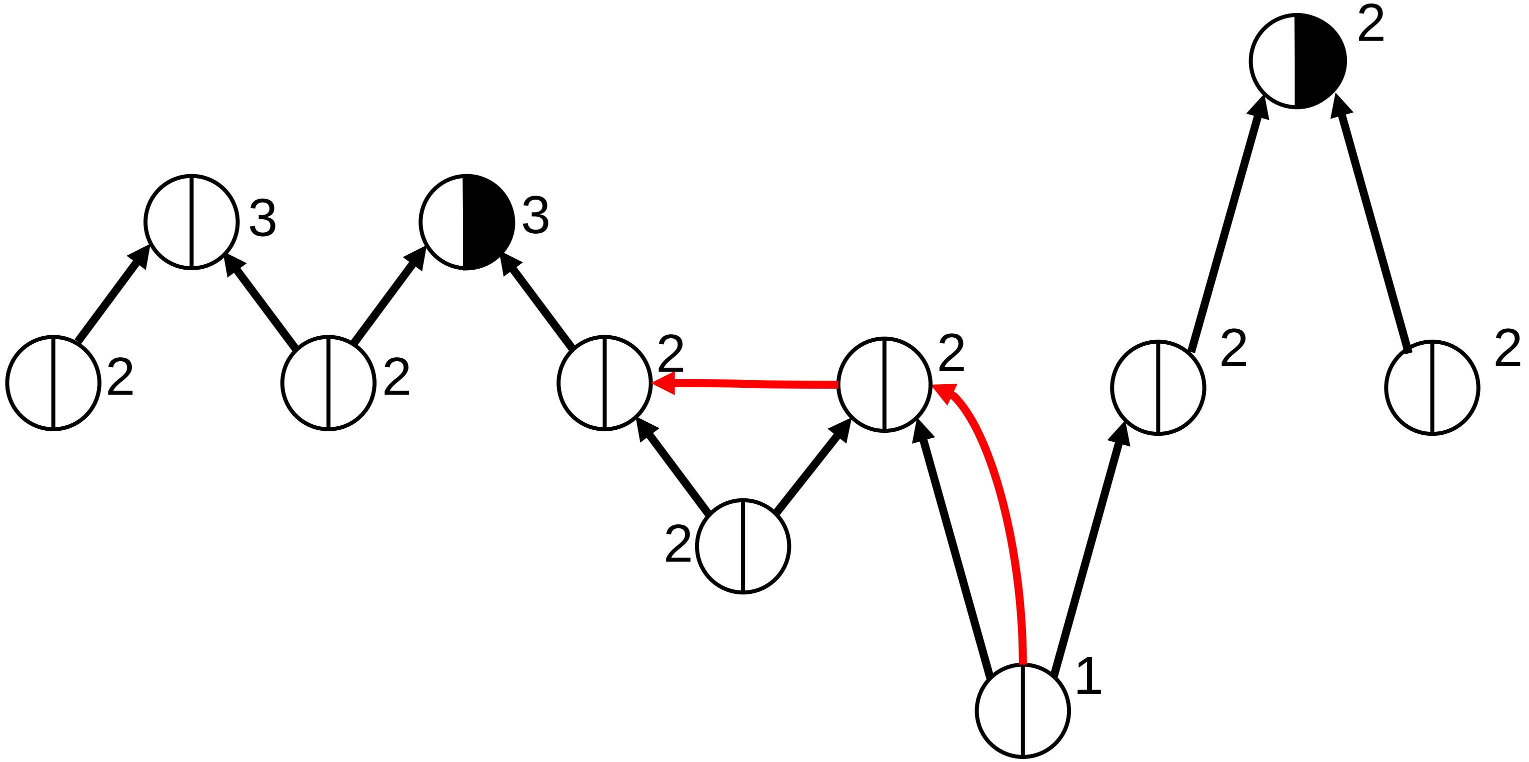
Reversible time dilation



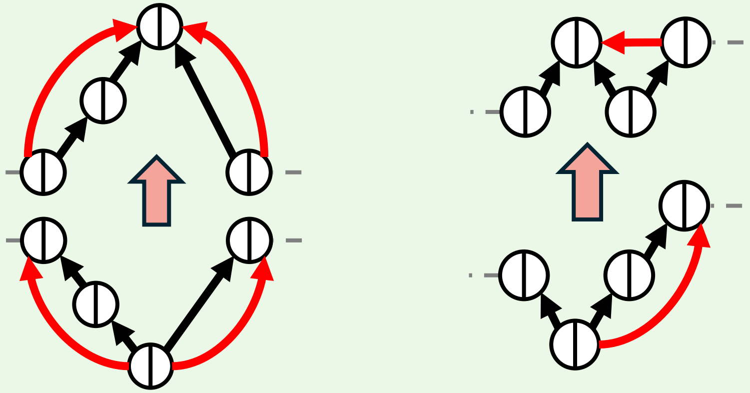
Local rule



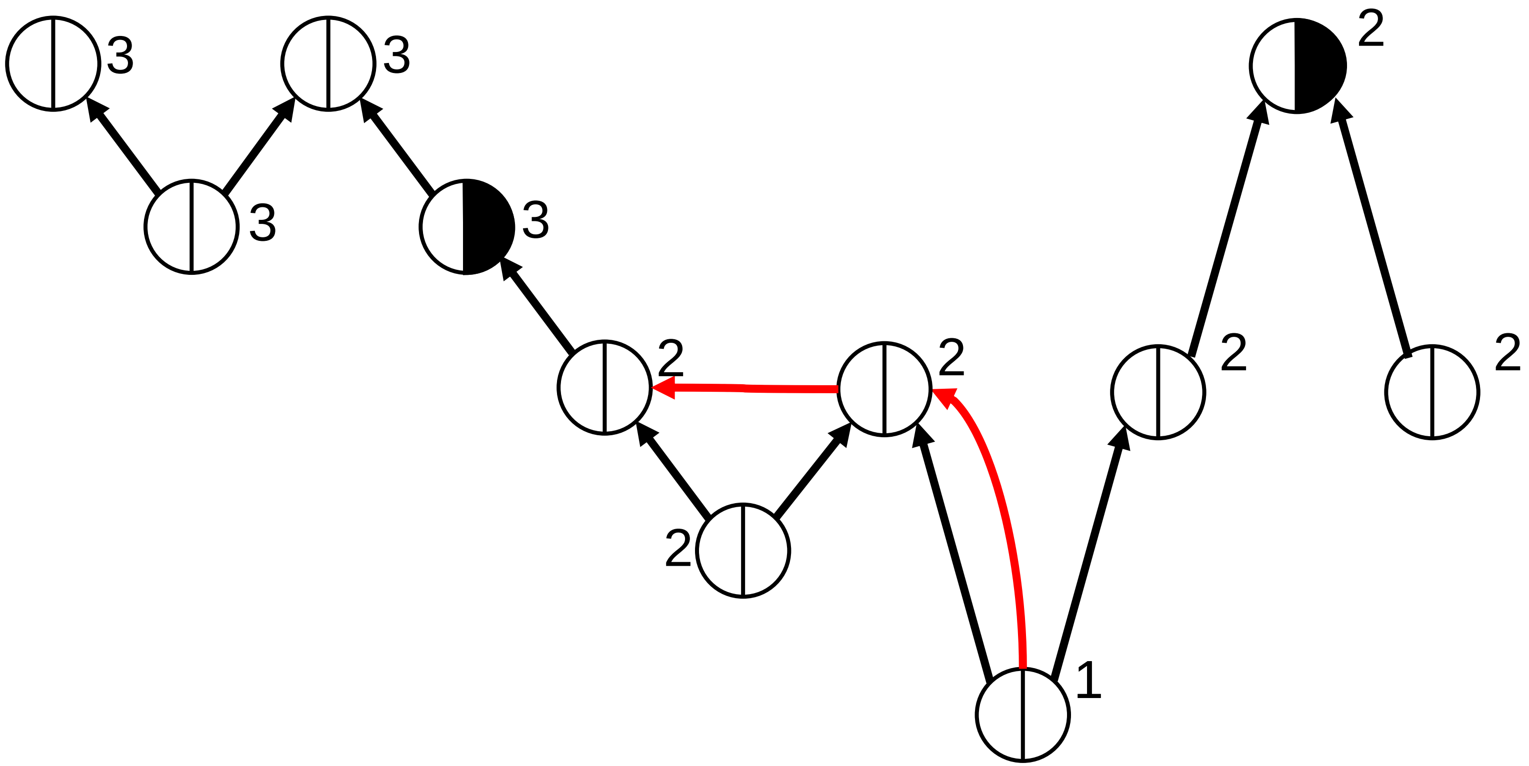
Reversible time dilation



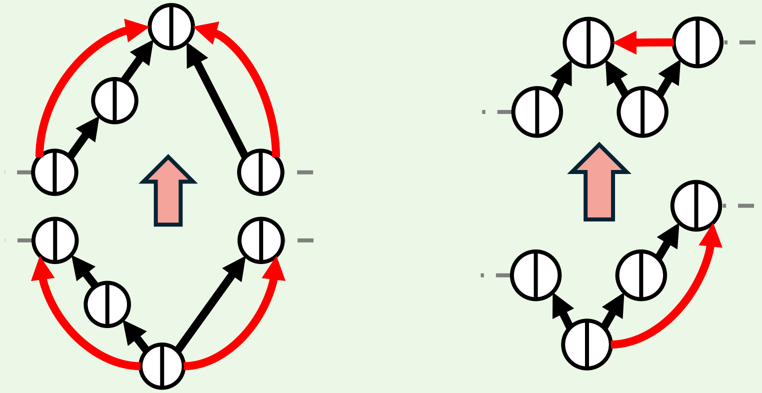
Local rule



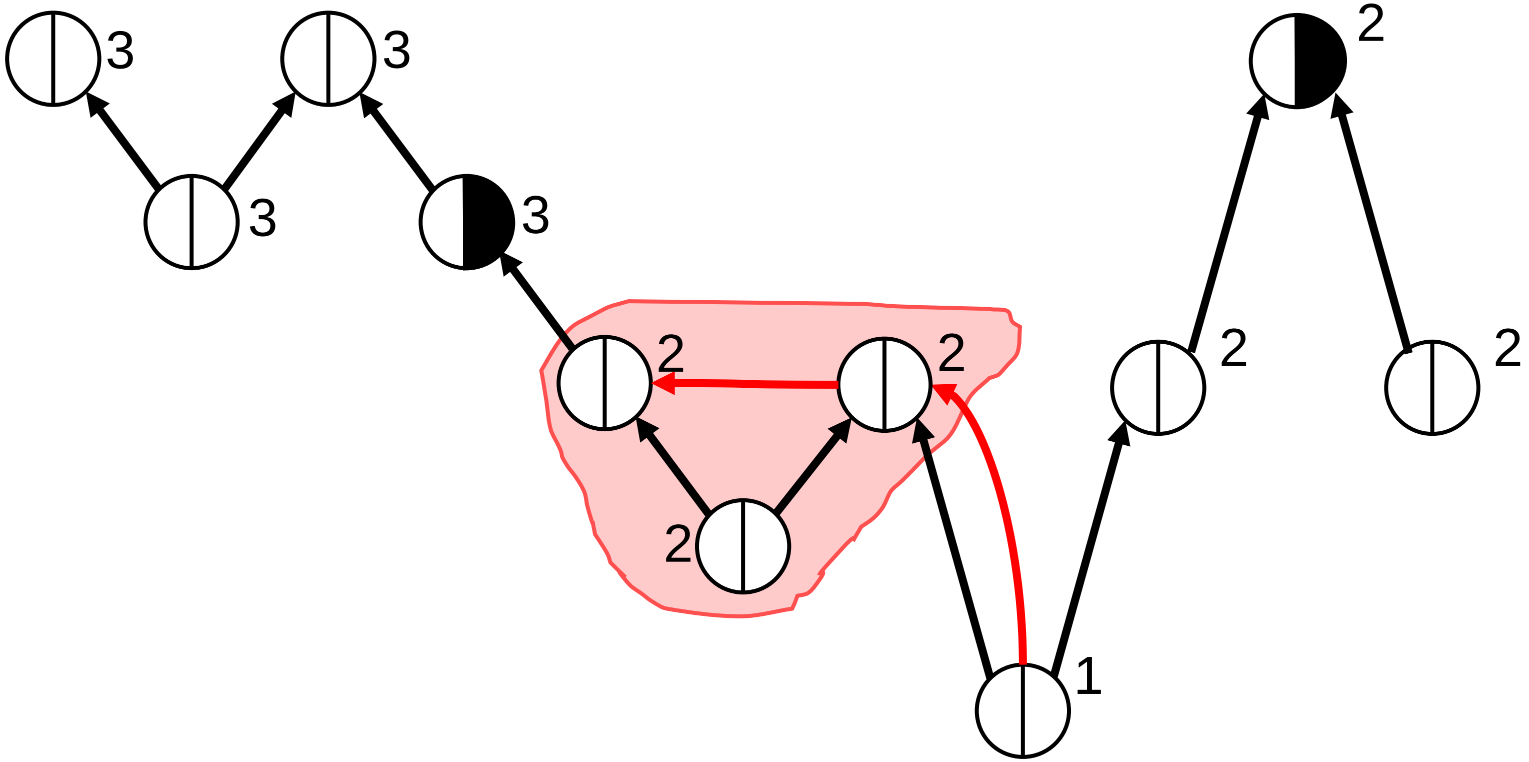
Reversible time dilation



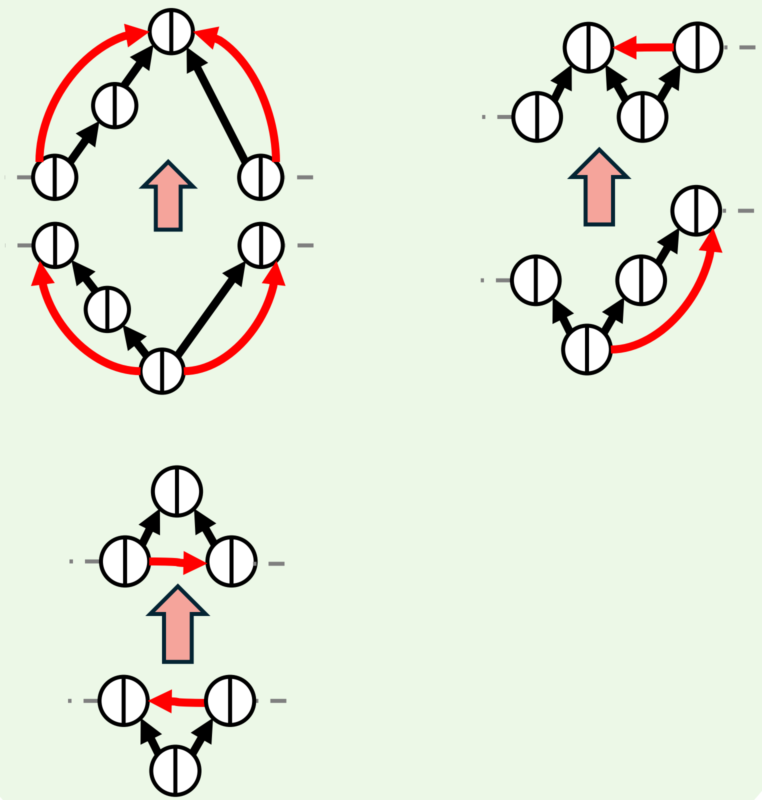
Local rule



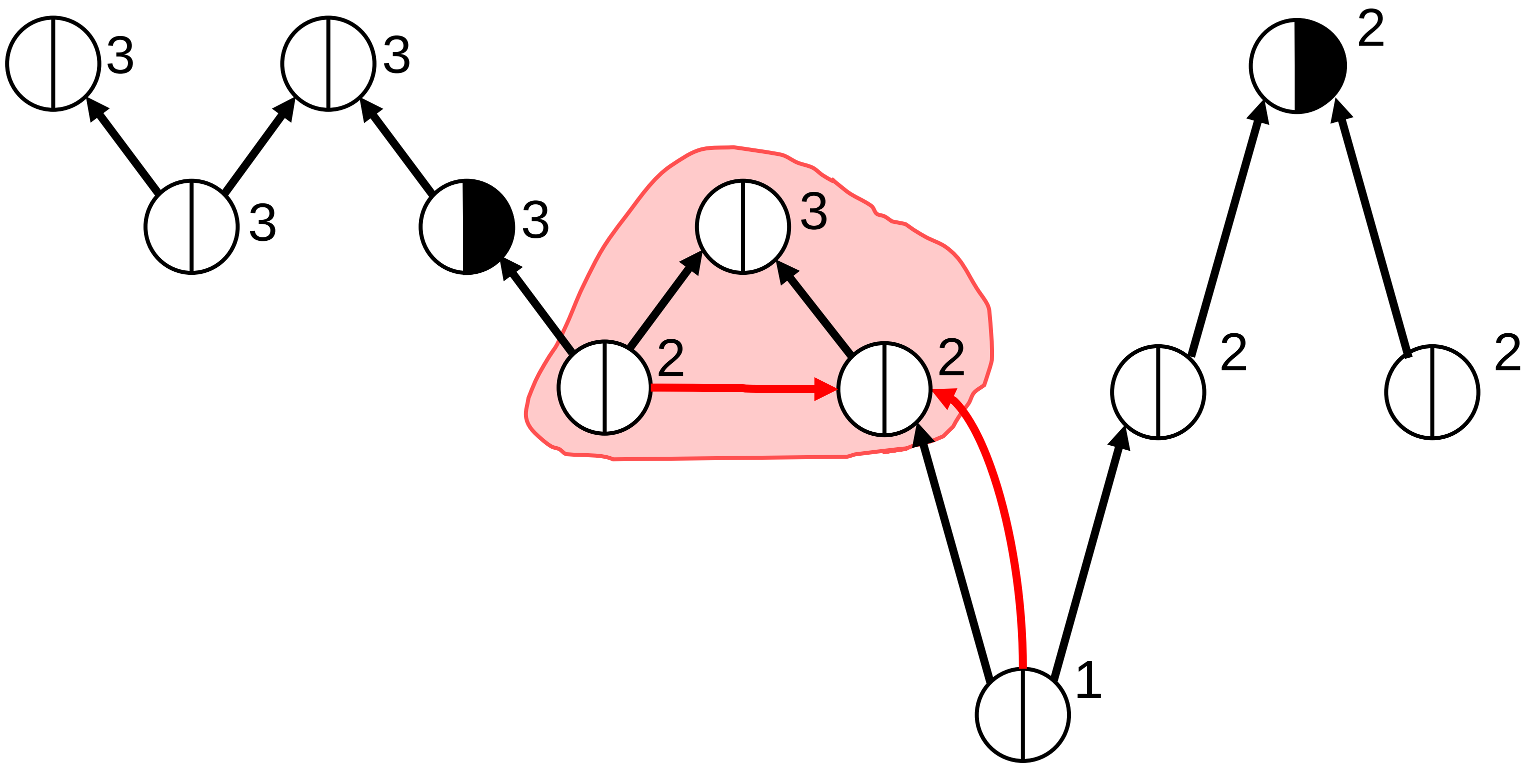
Reversible time dilation



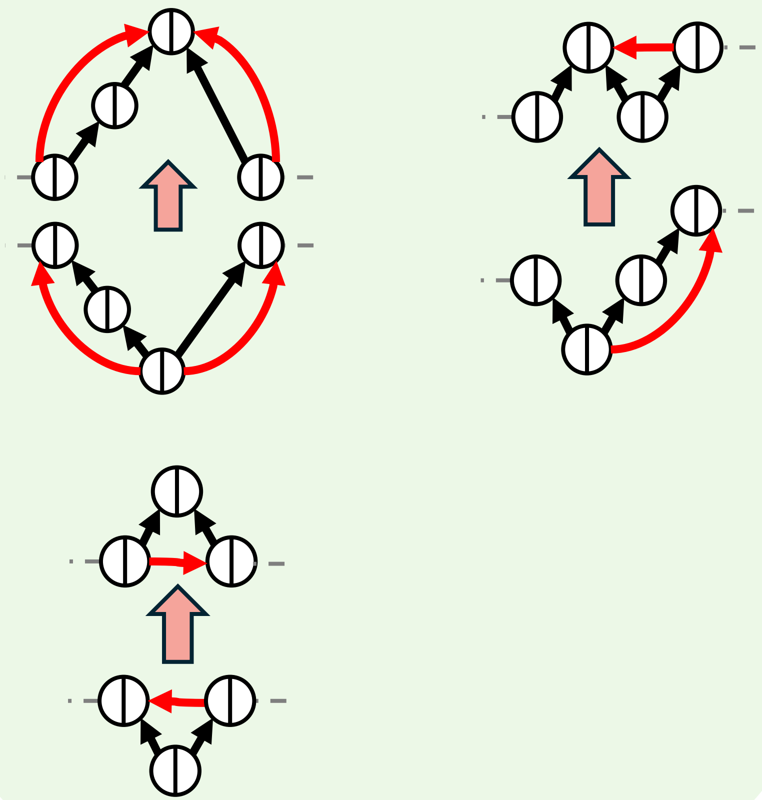
Local rule



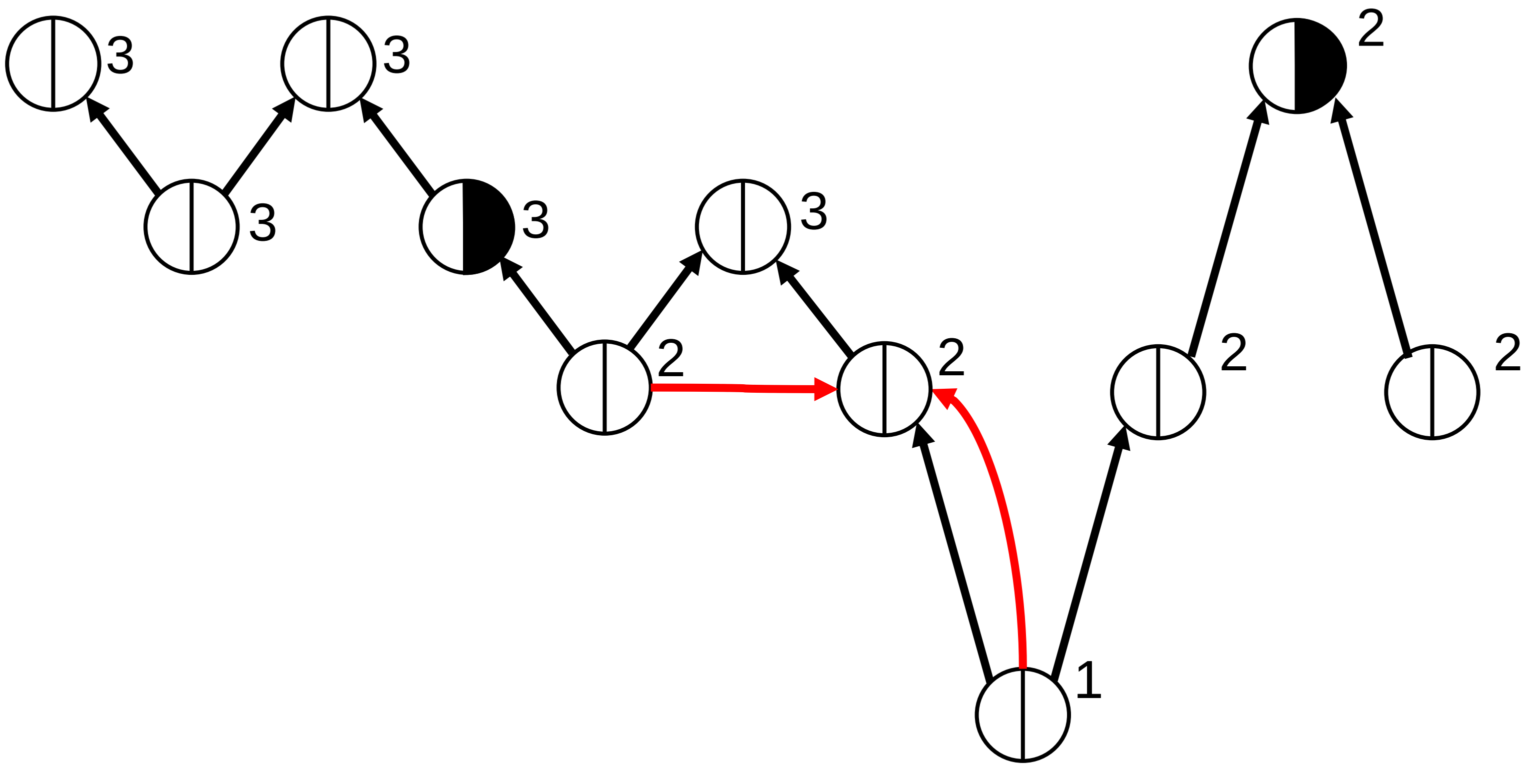
Reversible time dilation



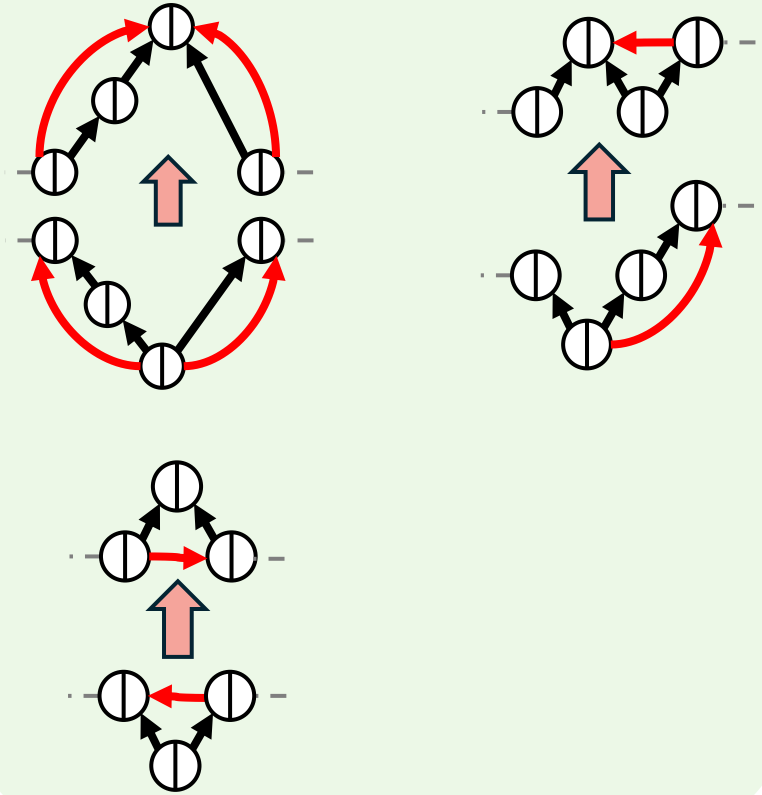
Local rule



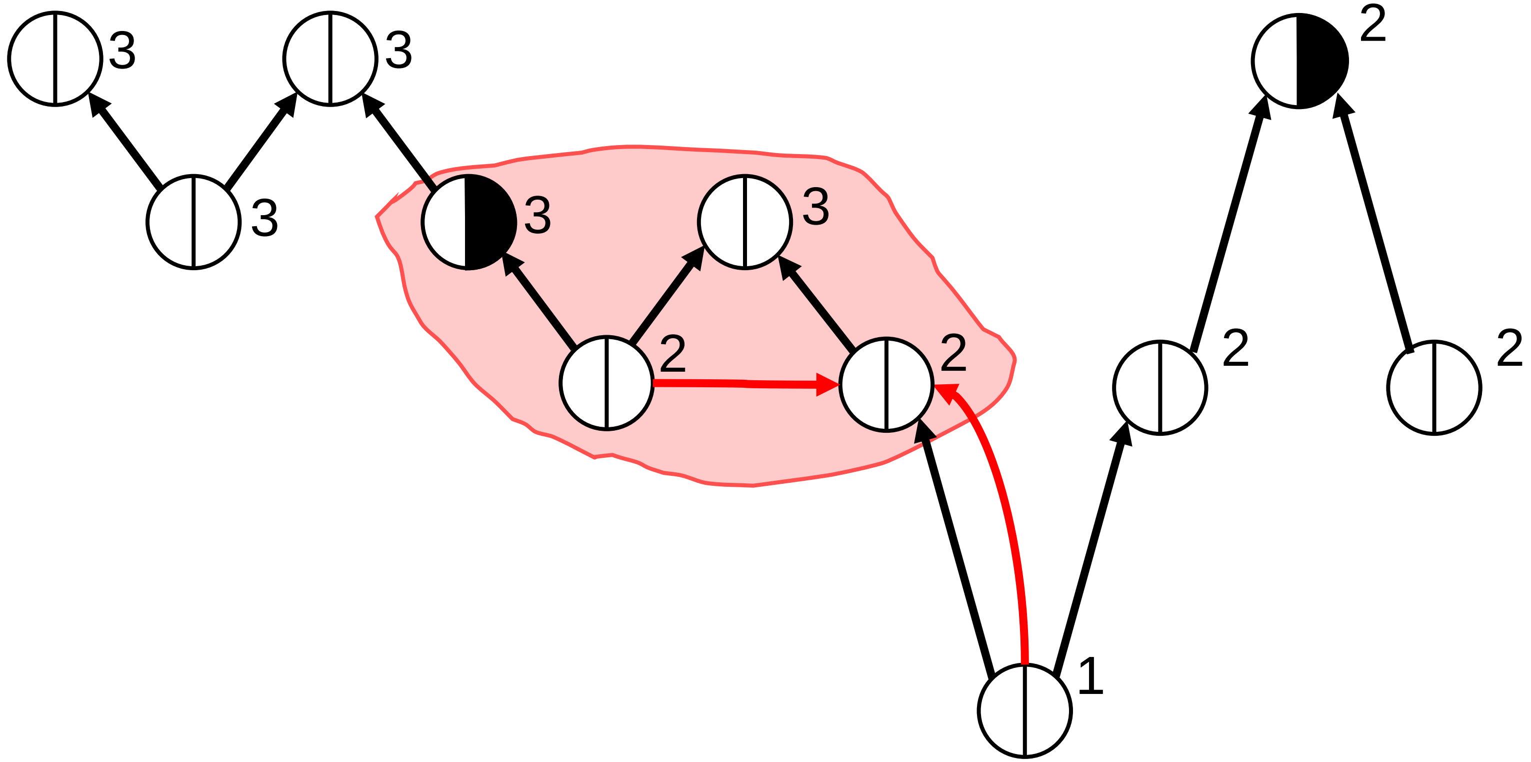
Reversible time dilation

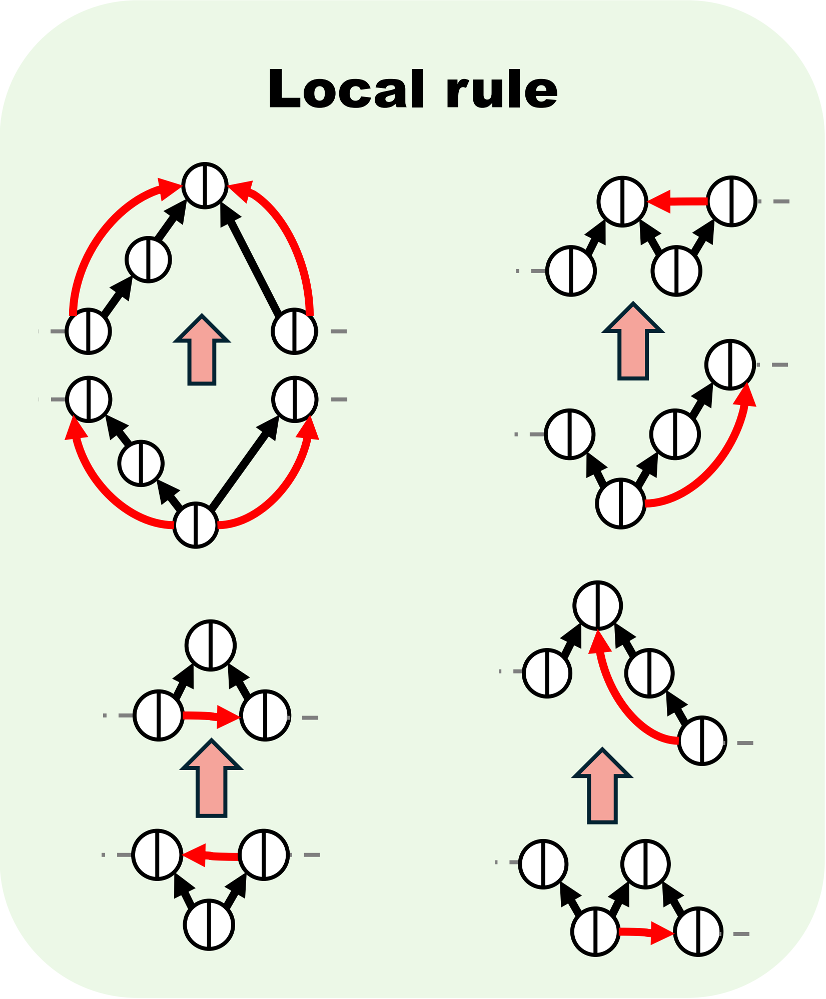


Local rule

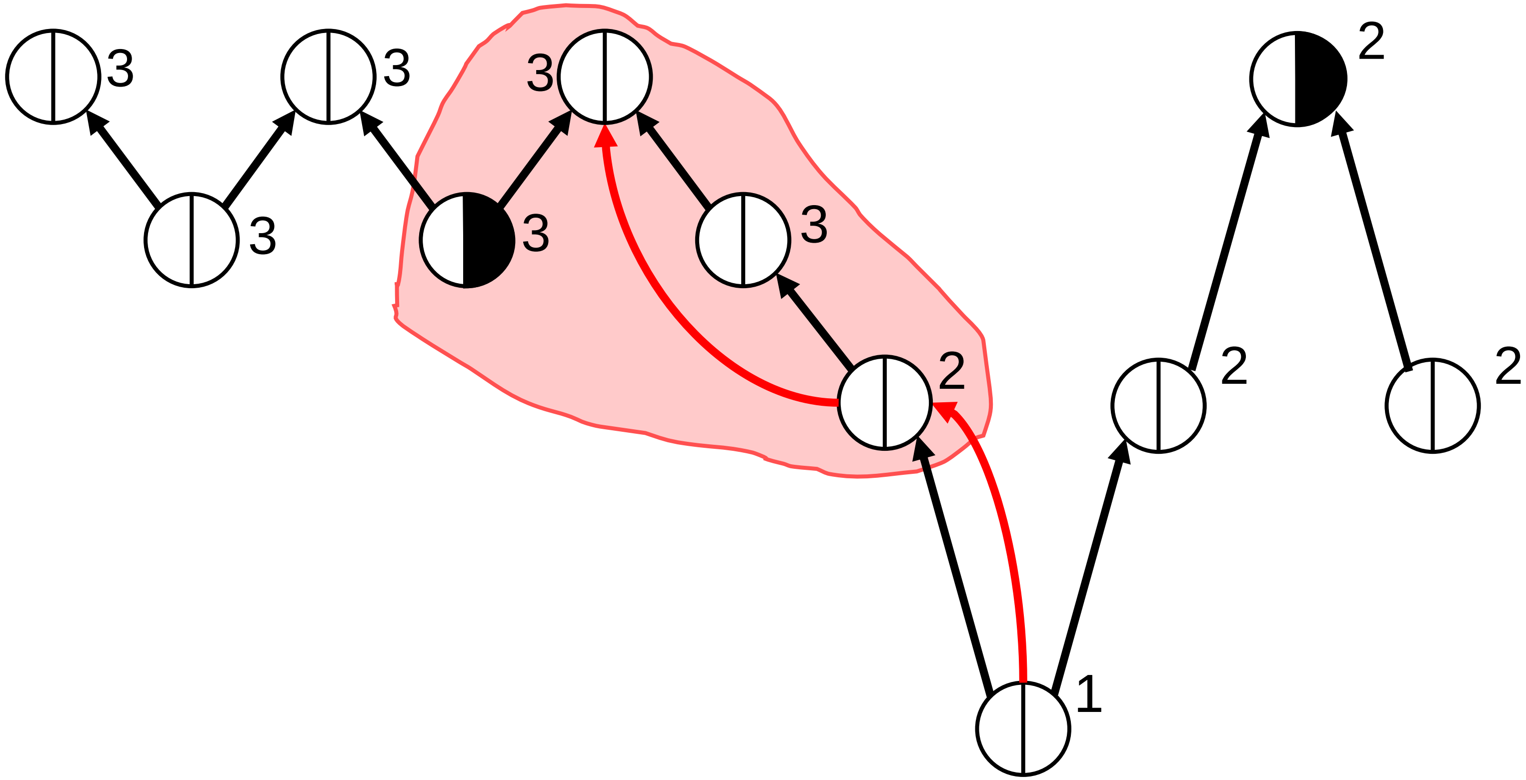


Reversible time dilation

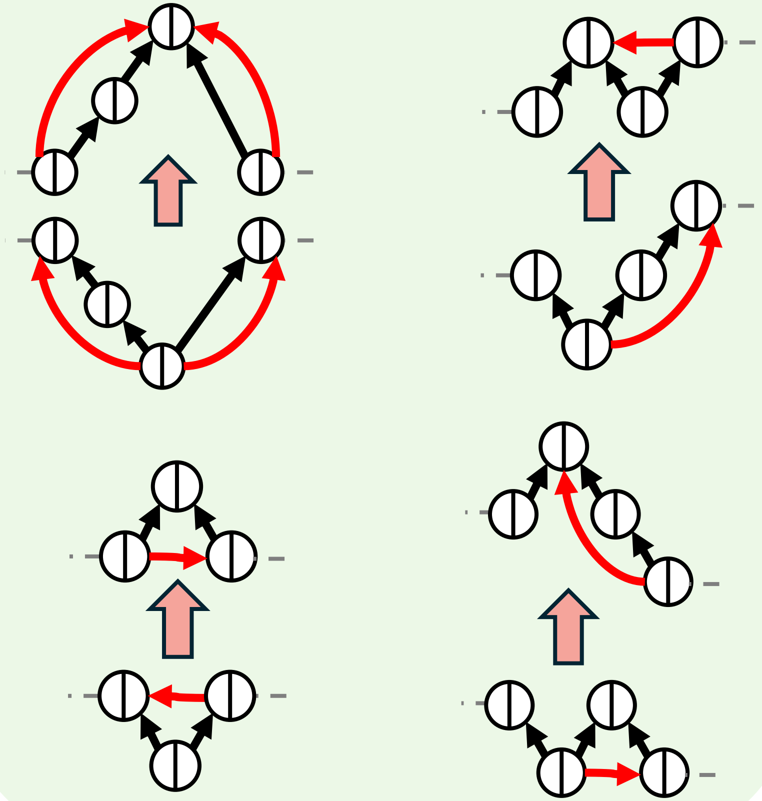




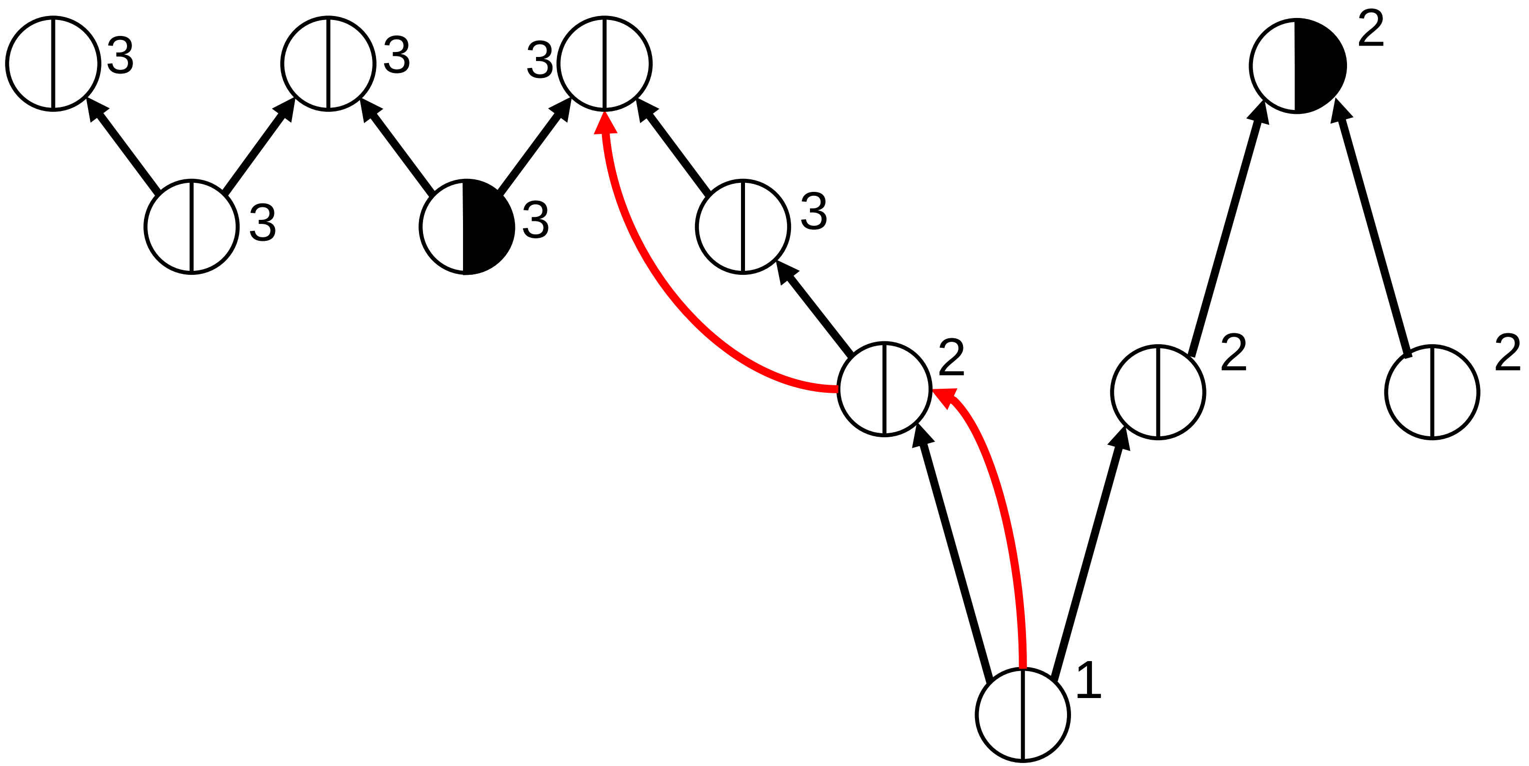
Reversible time dilation

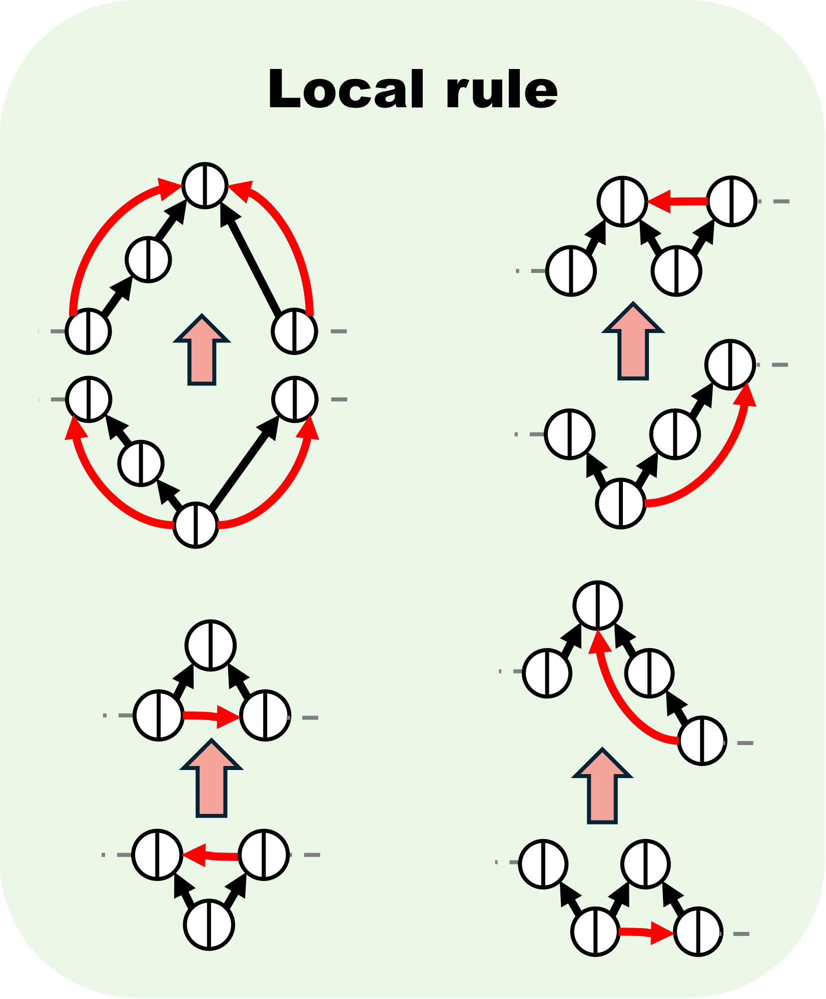


Local rule

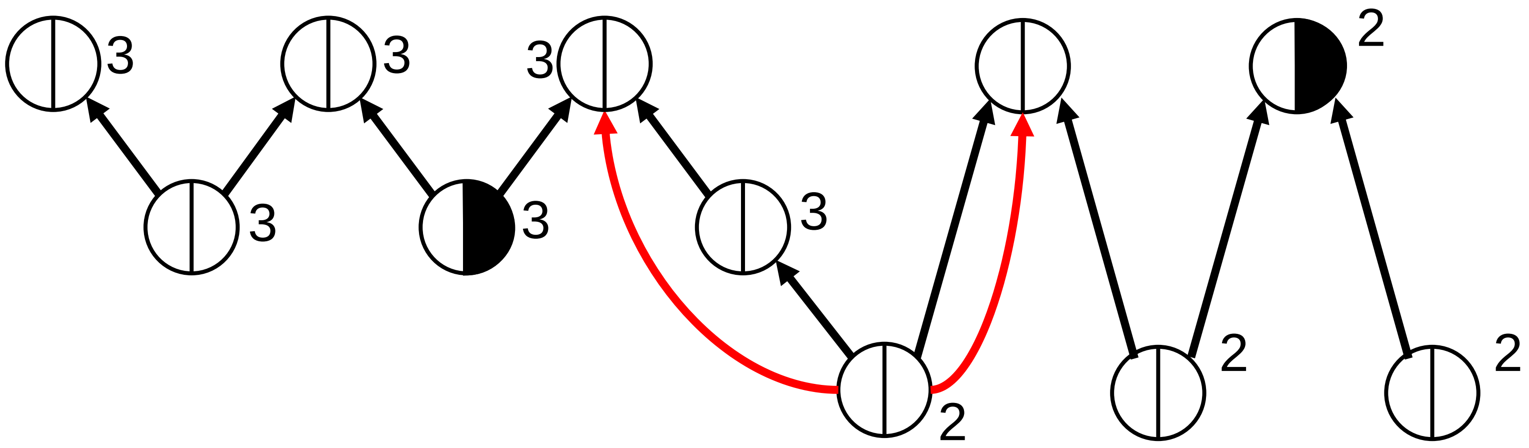


Reversible time dilation

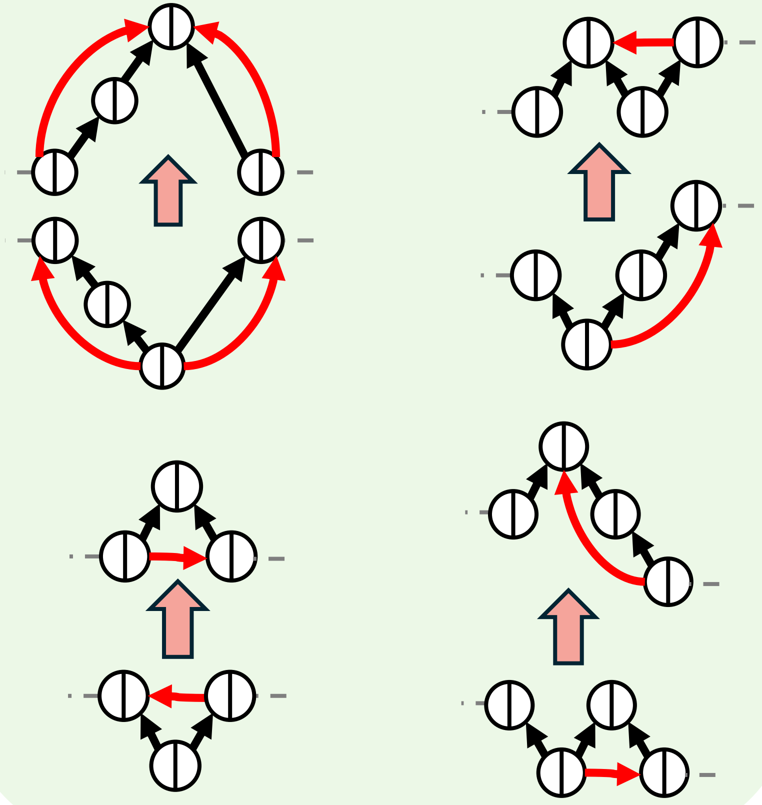




Reversible time dilation

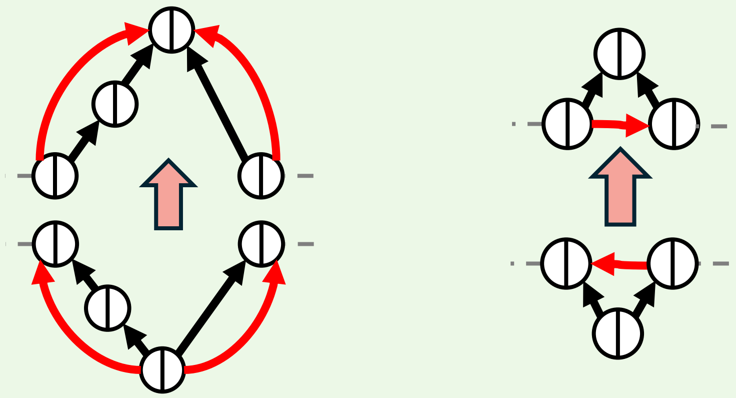


Local rule

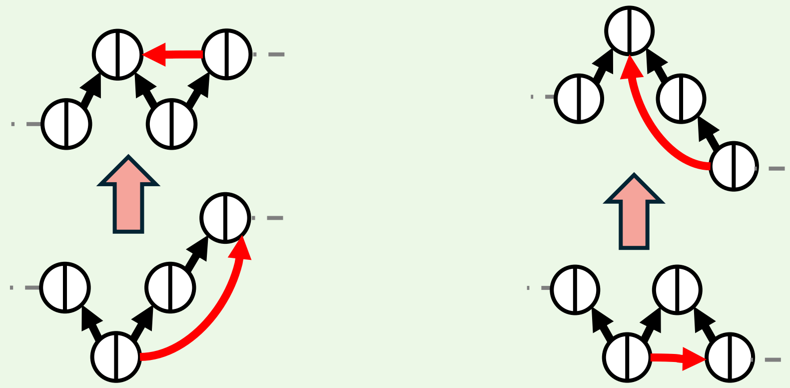


Is this really reversible ?

Is this really reversible ?



Local rule

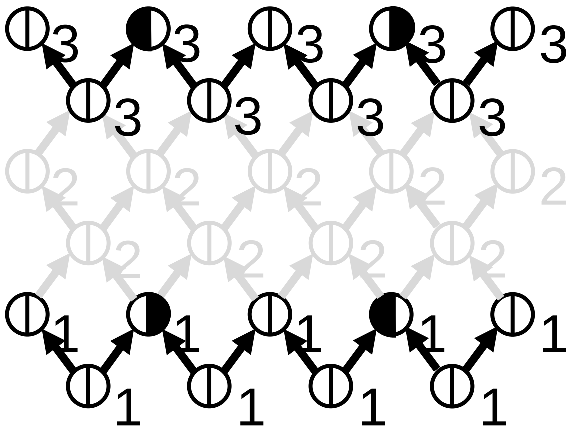


Theorem 2

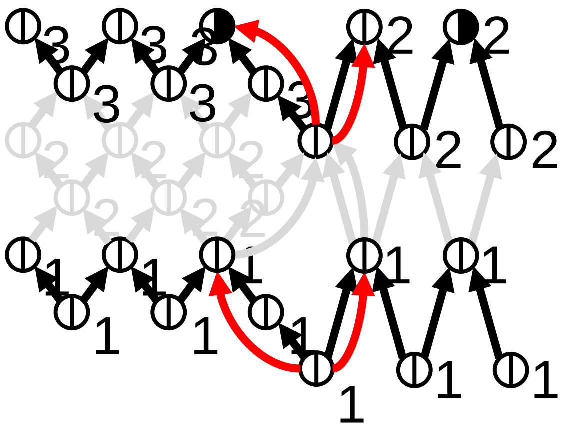
Any disk symmetric local rule is reversible.

Conclusion

Using graph rewriting we **can simulate synchronous** dynamical system …

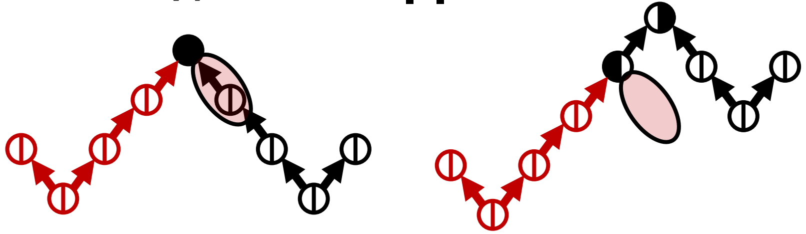


… but also represent some **intrinsically asynchronous** evolution.



While preserving important physical properties such as **determinism**…

$$A_w G \quad || \quad A_{w'} G$$



… and **reversibility.**