Verification of Infinite-State Systems

Backward Reachability Analysis for Monotonic Systems

Parosh Aziz Abdulla
Uppsala University
Sweden

Background

Parameterized Systems

Petri Nets

Lossy Channel Systems

Timed Petri Nets

Background

Background

Classical Approach

Finite-State Systems

Model Checking

Model $\models$ (safety) property

Challenge:

Infinite-State Systems

Sources of "Infiniteness":

Unbounded Data Structures

- stacks (recursion)
- queues (protocols)
- counters (programs)
- clocks (time)
- lists, trees, graphs (heaps)

Unbounded Control Structures

- parameterized systems
- multithreaded programs
- concurrent libraries
- Petri nets

Multiple Sources:

- timed Petri nets
- recursive programs with unbounded data
- channels with time stamps
- etc

Infinite-State Systems

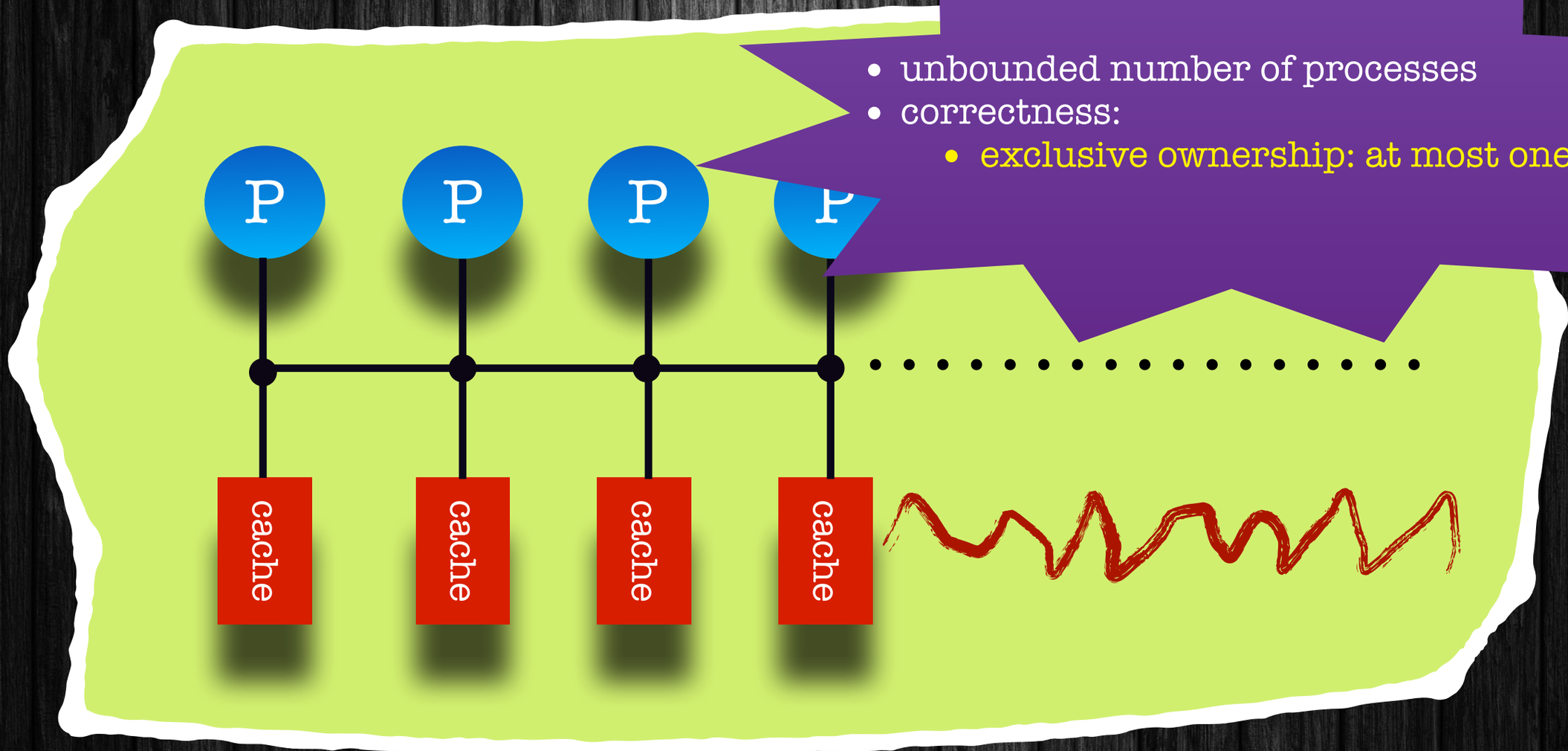
Infinite-State Systems

Unbounded Number of Processes

Infinite-State Systems

Unbounde… Numb… Processes

Cache Coherence Protocol



- unbounded number of processes
- correctness:
  - exclusive ownership: at most one process

Infinite-State Syst

Unbounded Data Structures

Infinite-State Systems

Unbounded Data Structures

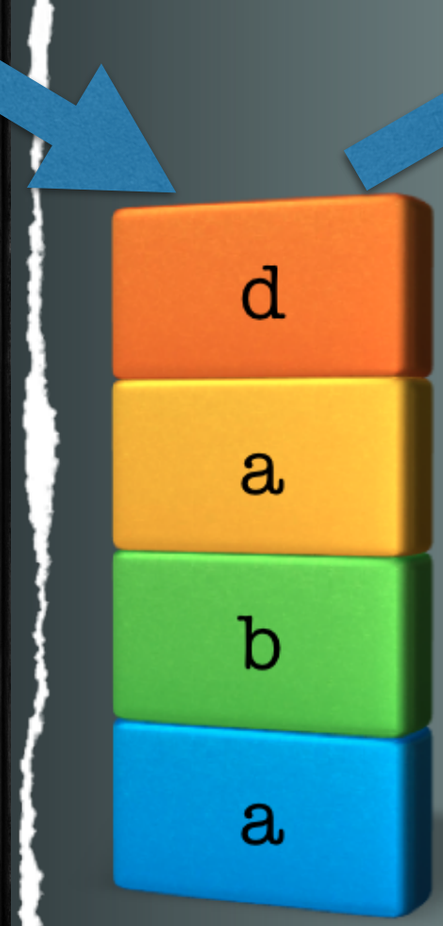
Unbounded Channels

- unbounded FIFO channels
- correctness:
  - regardless of channels size

Infinite-State Systems

Unbounded Data Structures

Unbounded Stack




```
push ( a );
push ( b );
push ( c );
pop ( );
```

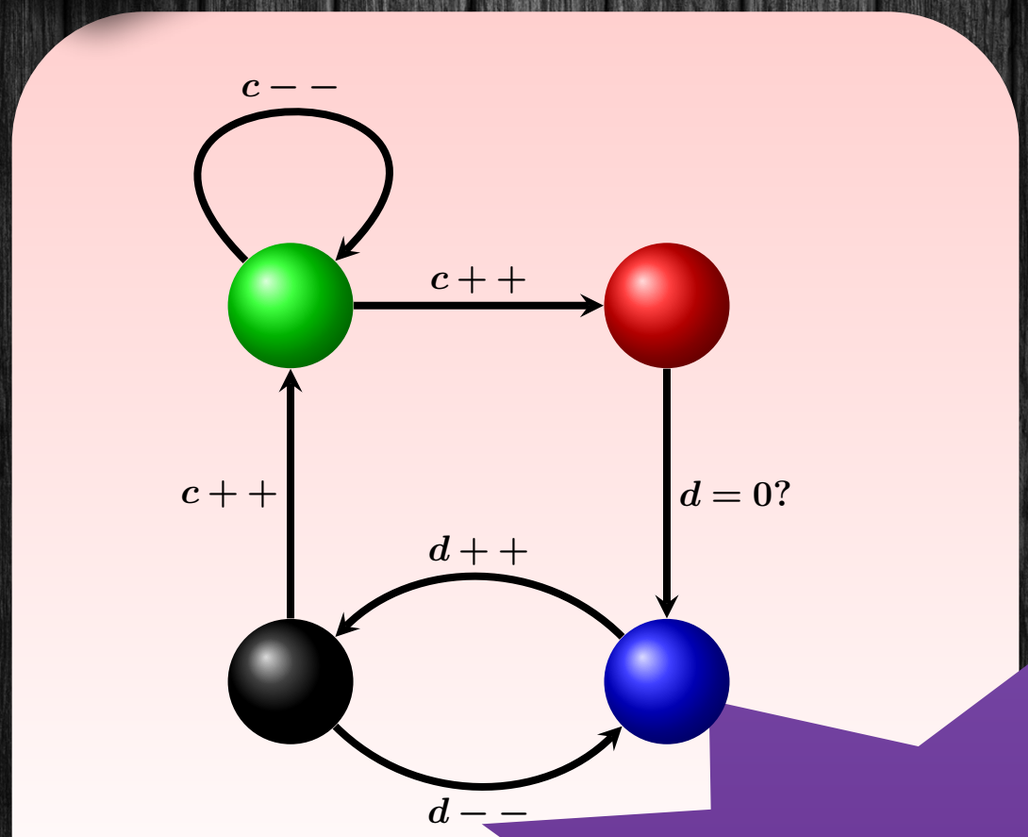
c

- unbounded stack
- correctness:
  - regardless of stack size

Infinite-State Systems

Unbounded Data Structures

Unbounded Counters

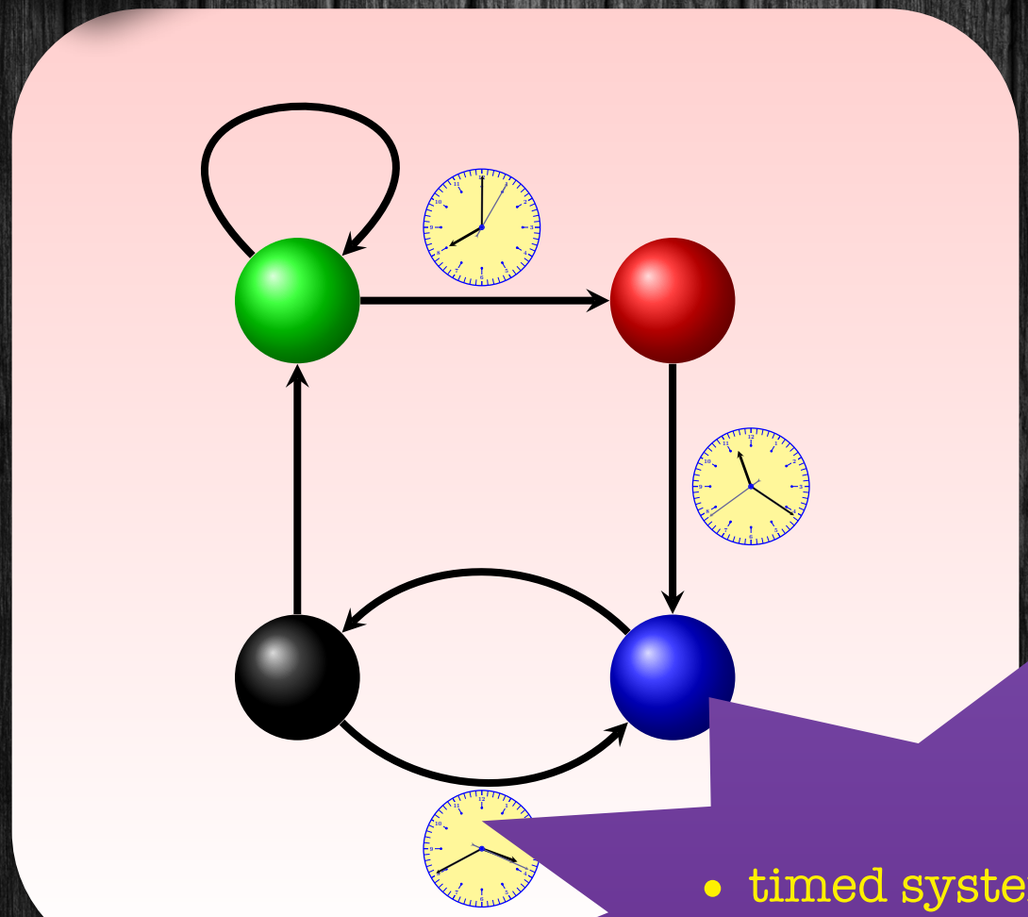


- unbounded counters
- correctness:
  - regardless of counter values

Infinite-State Systems

Unbounded Data Structures

Clocks



- timed systems
- real-value clocks

Background

Classical Approach

Finite-State Systems

Model Checking

Model $\models$ (safety) property

Challenge:

Infinite-State Systems

Sources of "Infiniteness":

Unbounded Data Structures

- stacks (recursion)
- queues (protocols)
- counters (programs)
- clocks (time)
- lists, trees, graphs (heaps)

Unbounded Control Structures

- parameterized systems
- multithreaded programs
- concurrent libraries
- Petri nets

Multiple Sources:

- timed Petri nets
- recursive programs with unbounded data
- queues with time stamps
- etc

Background

Parameterized Systems

Petri Nets

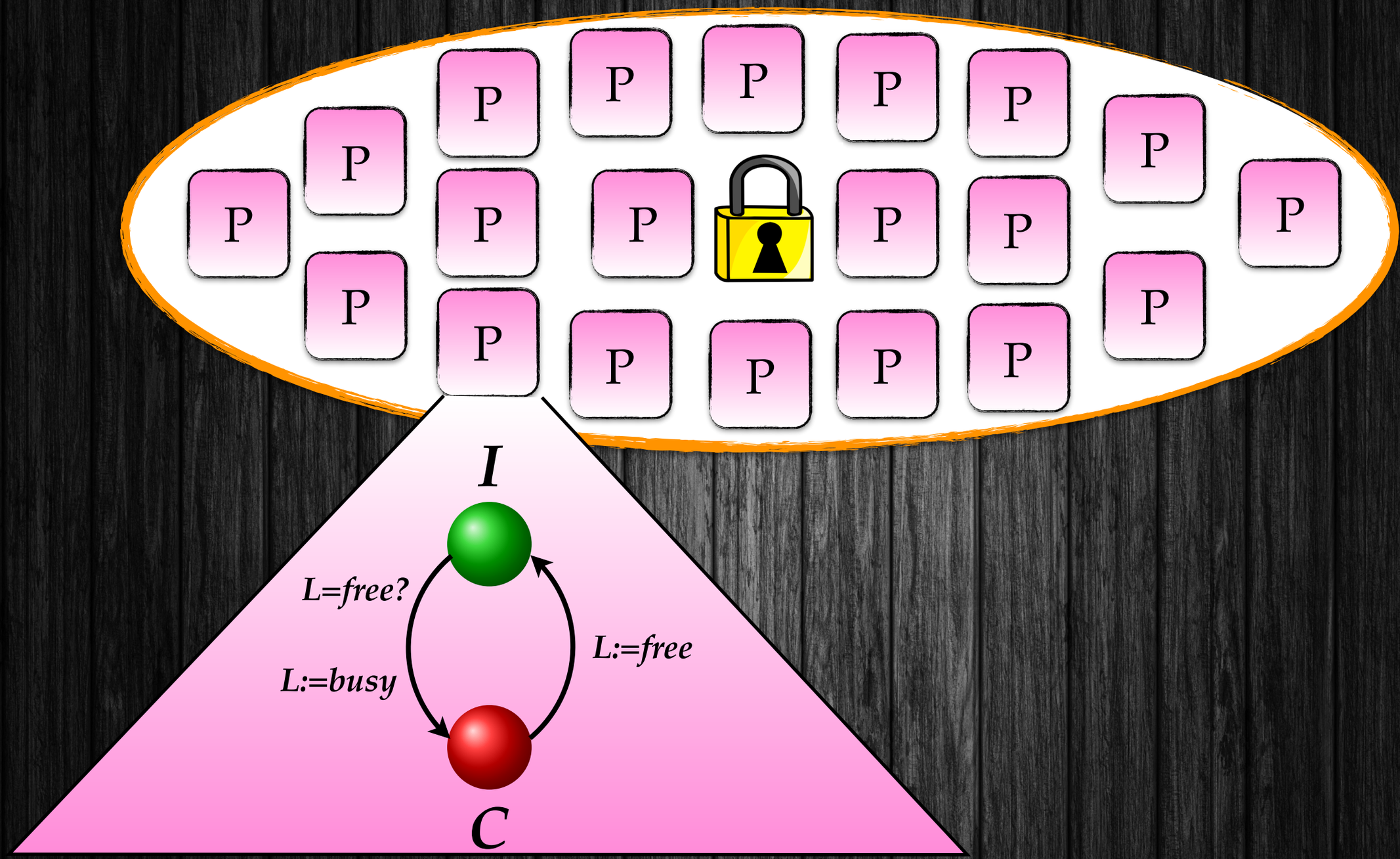
Lossy Channel Systems

Timed Petri Nets

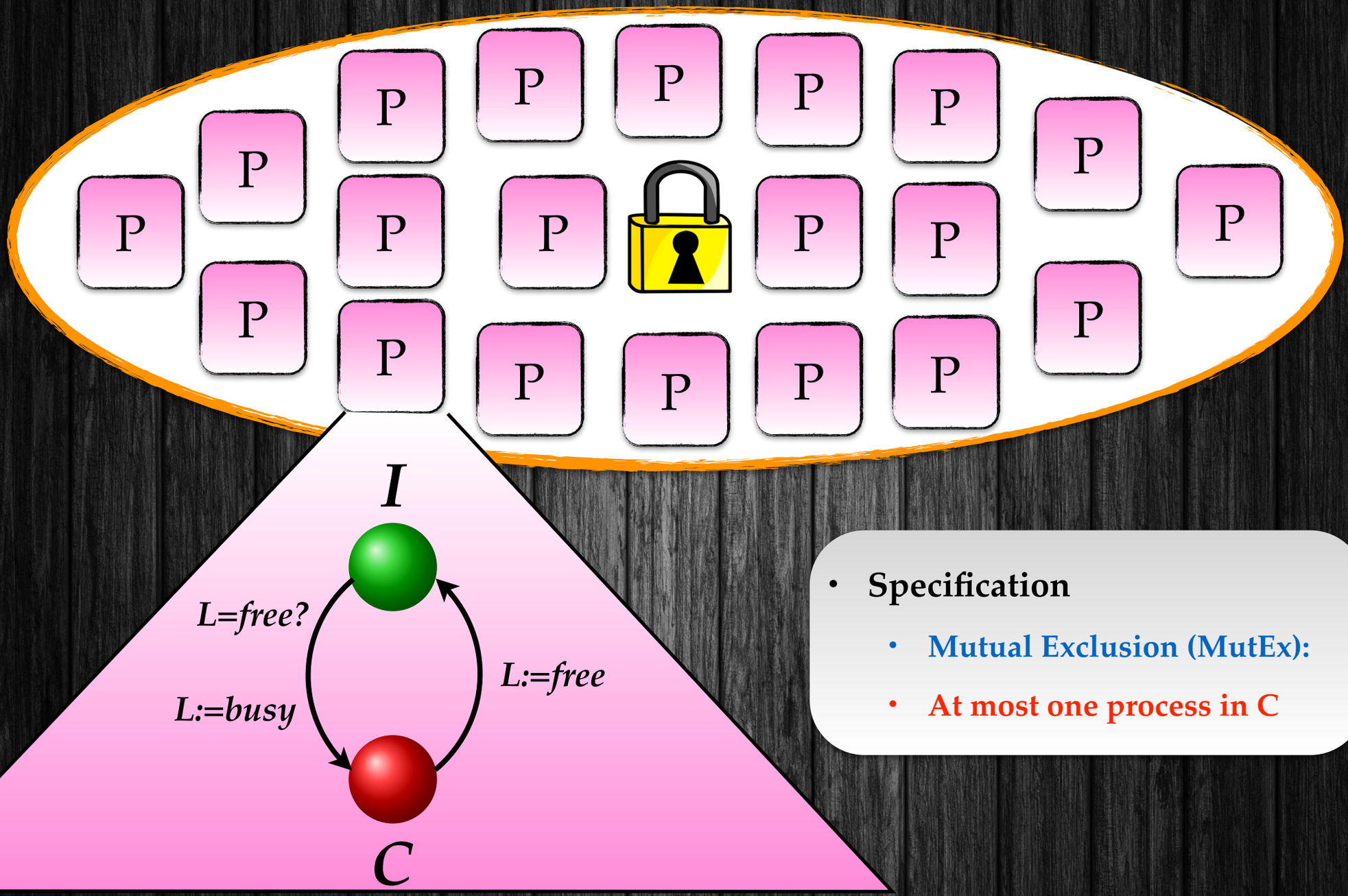
Parameterized Systems

Parameterized Systems

Parameterized Systems



Parameterized Systems



- **Specification**
  - **Mutual Exclusion (MutEx):**
  - **At most one process in C**

Parameterized Systems

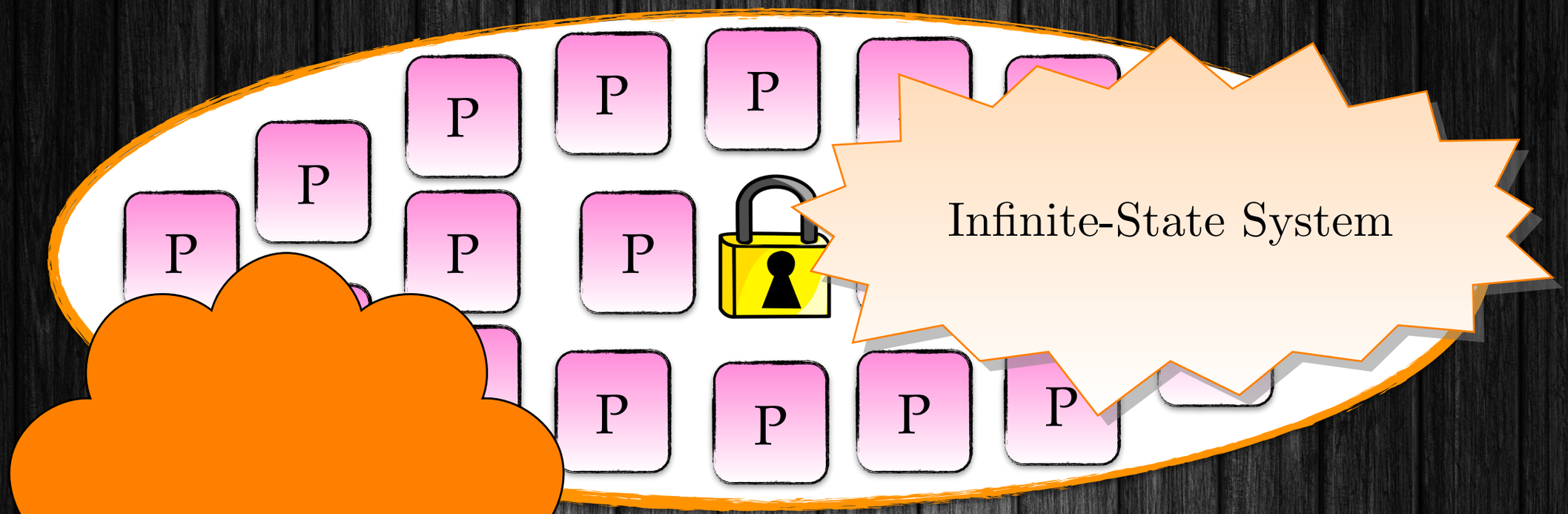
- **Specification**
  - **Mutual Exclusion (MutEx):**
  - **At most one process in C**

Parameterized Systems

- **Specification**
  - **Mutual Exclusion (MutEx):**
  - **At most one process in C**

- **Task = Parameterized Verification**
  - **Verify correctness regardless of the number of processes**
  - $\forall n.\,(P^n \mid L) \models MutEx$

Parameterized Systems



- **Specification**
  - **Mutual Exclusion (MutEx):**
  - **At most one process in C**

- **Task = Parameterized Verification**
  - **Verify correctness regardless of the number of processes**
  - $\forall n.\,(P^n \mid L) \models MutEx$

Background

Parameterized Systems

Petri Nets

Lossy Channel Systems

Timed Petri Nets

Petri Nets

Petri Nets

Model

Configurations

Transitions

Ordering

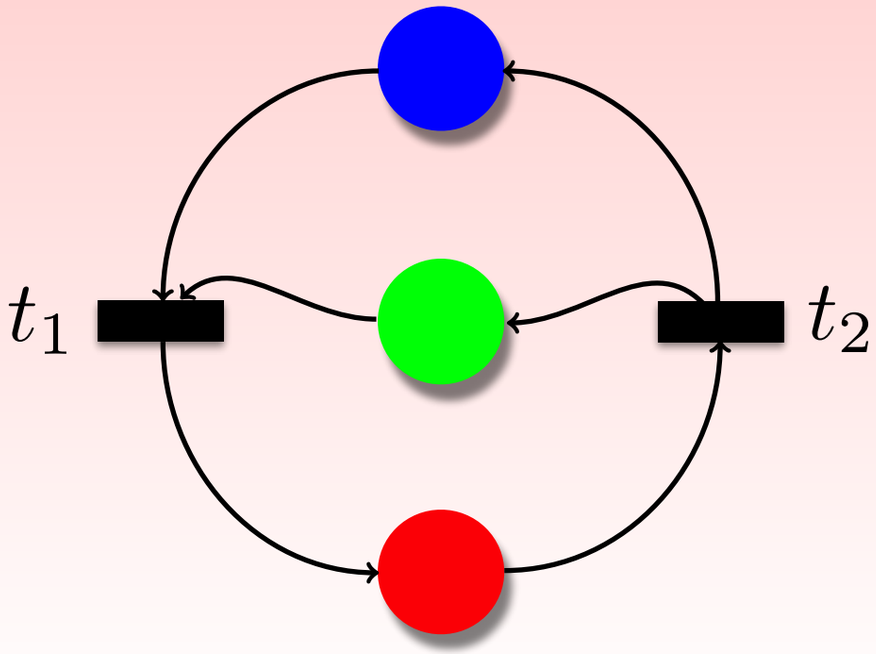
Monotoncity

Upward Closed Sets

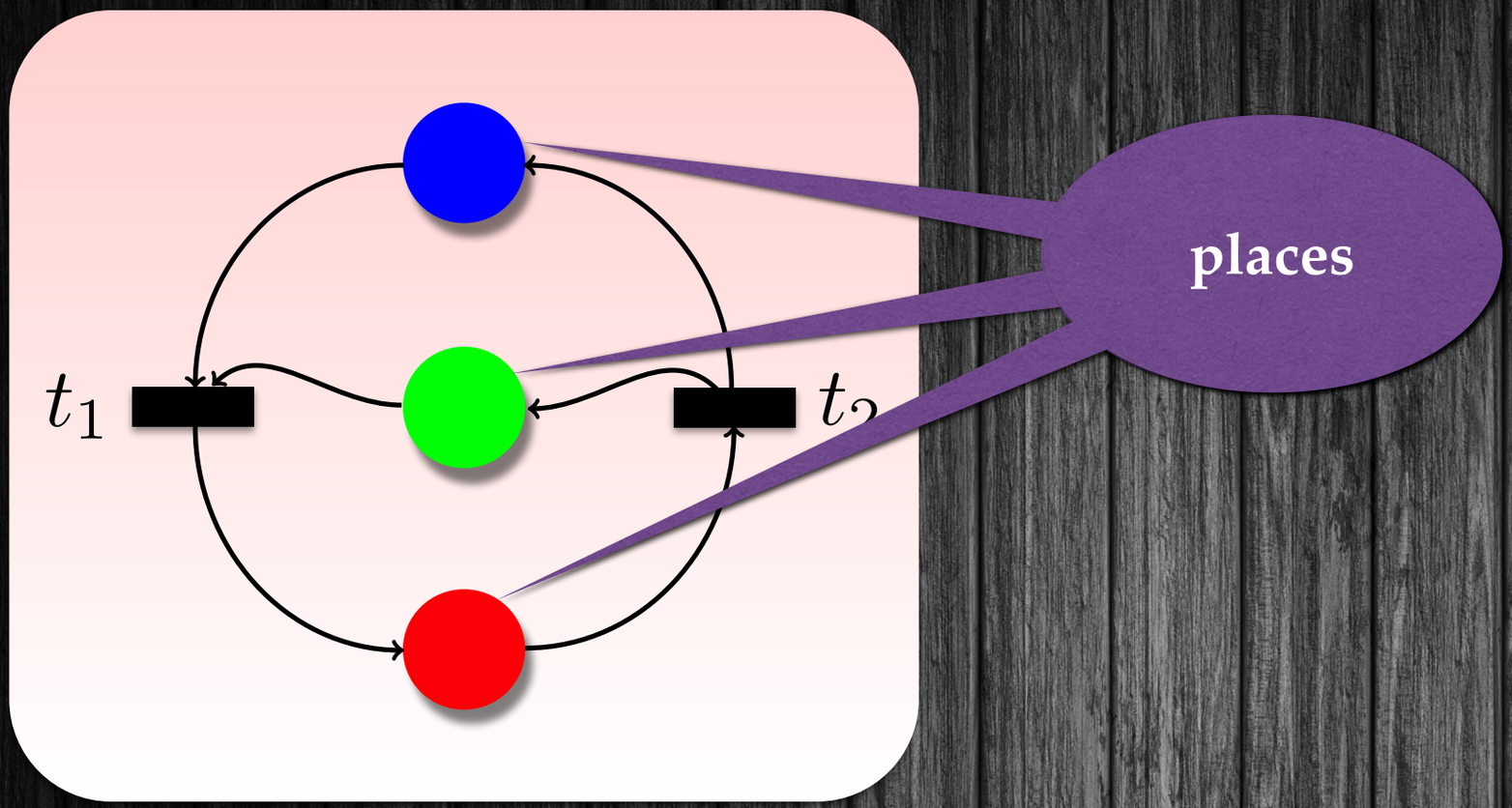
Computing Predecessors

Backward Reachability

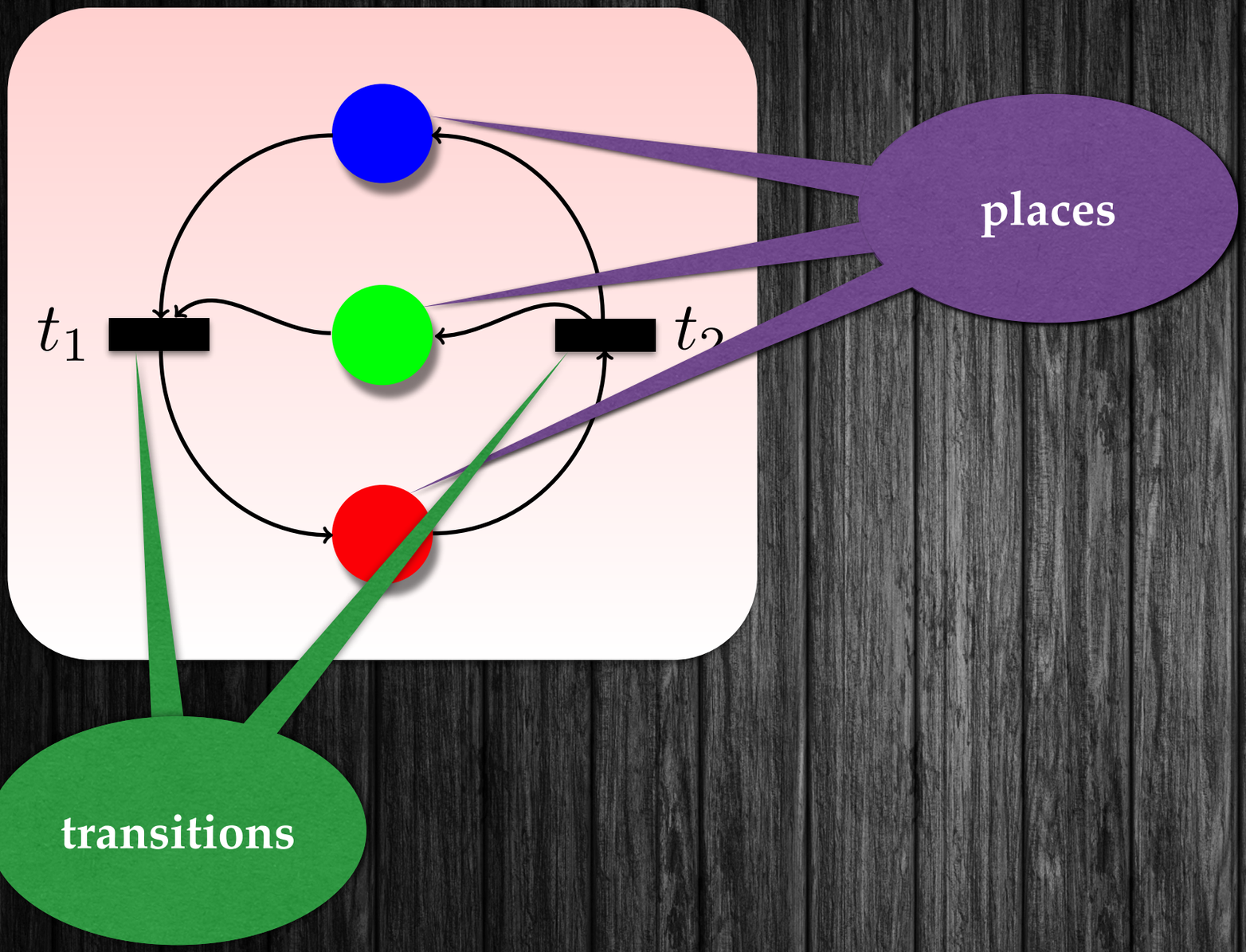
Petri Nets



Petri Nets



Petri Nets



Petri Nets

Model

Configurations

Transitions

Ordering

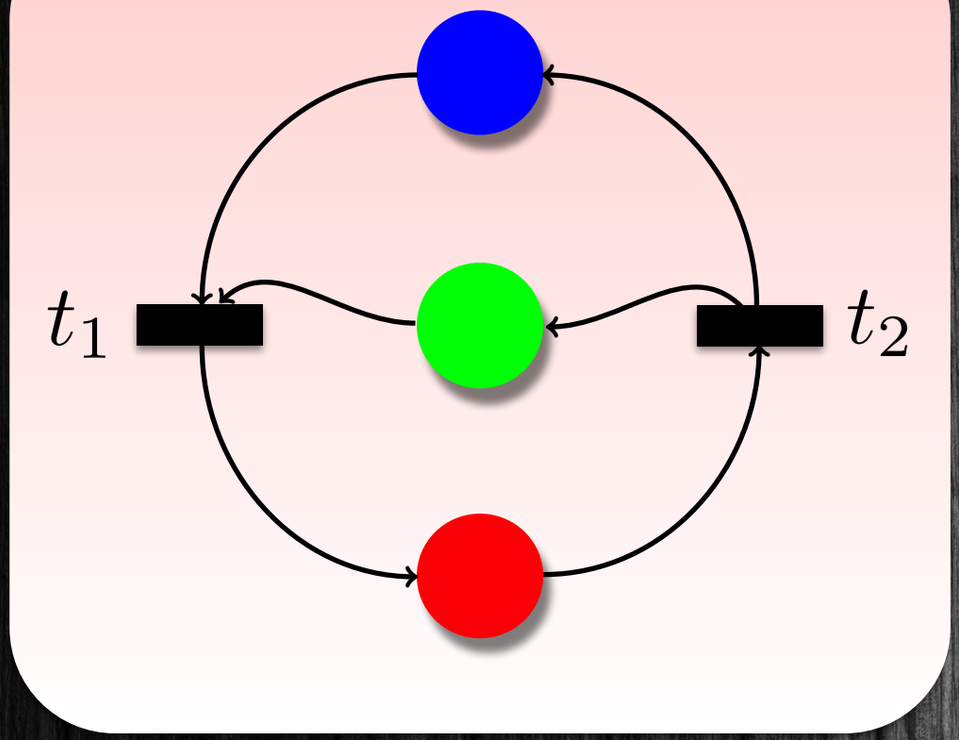
Monotoncity

Upward Closed Sets

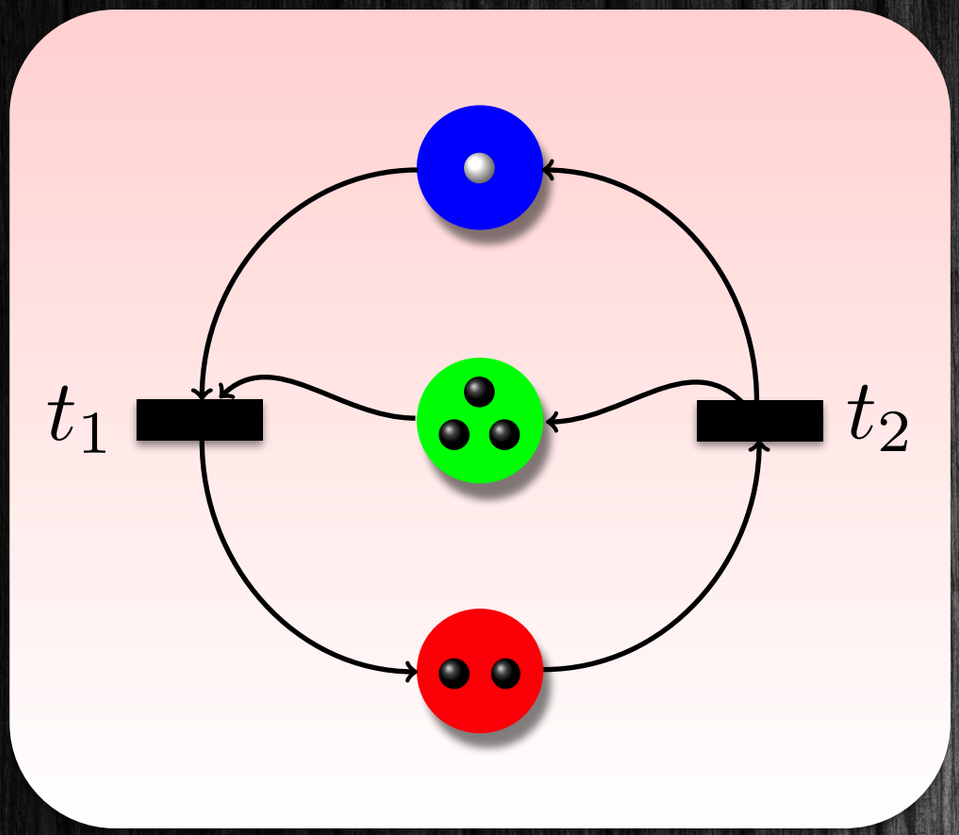
Computing Predecessors

Backward Reachability

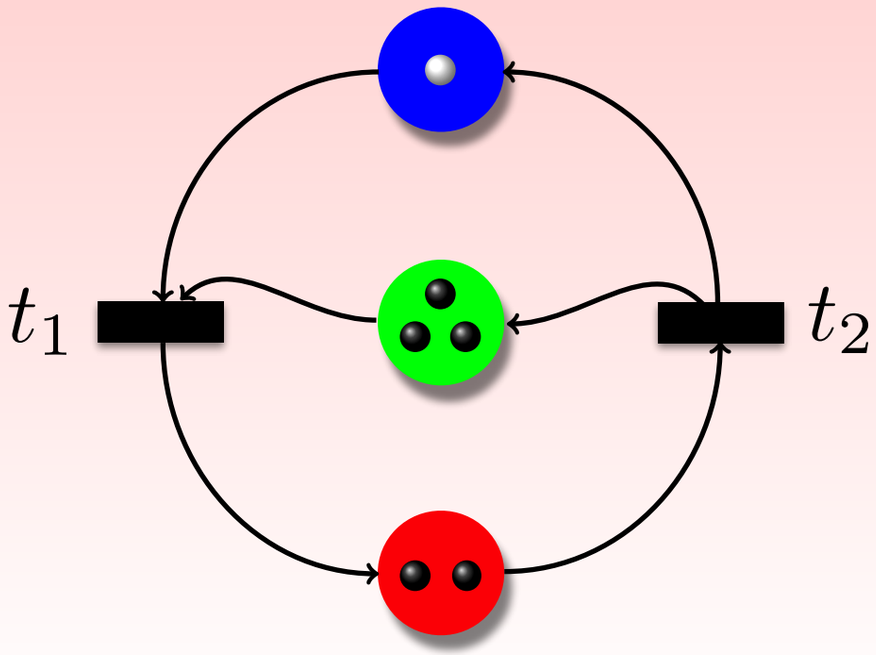
Markings

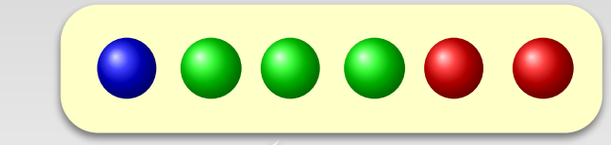


Markings



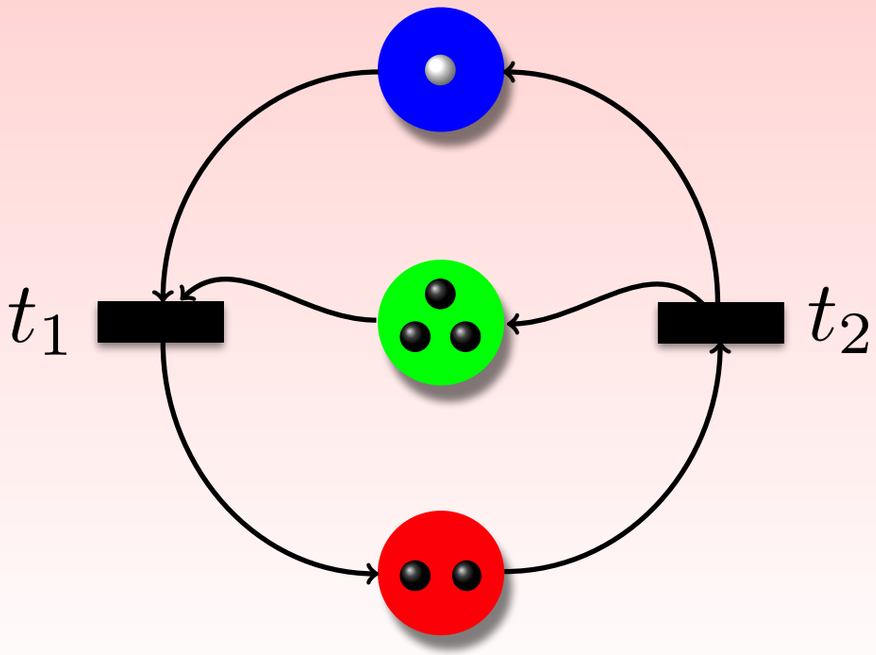
Markings

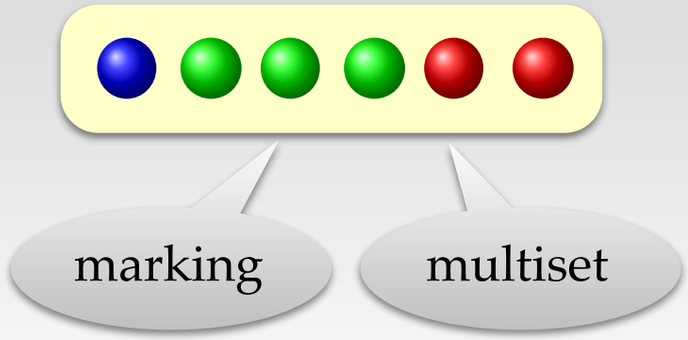




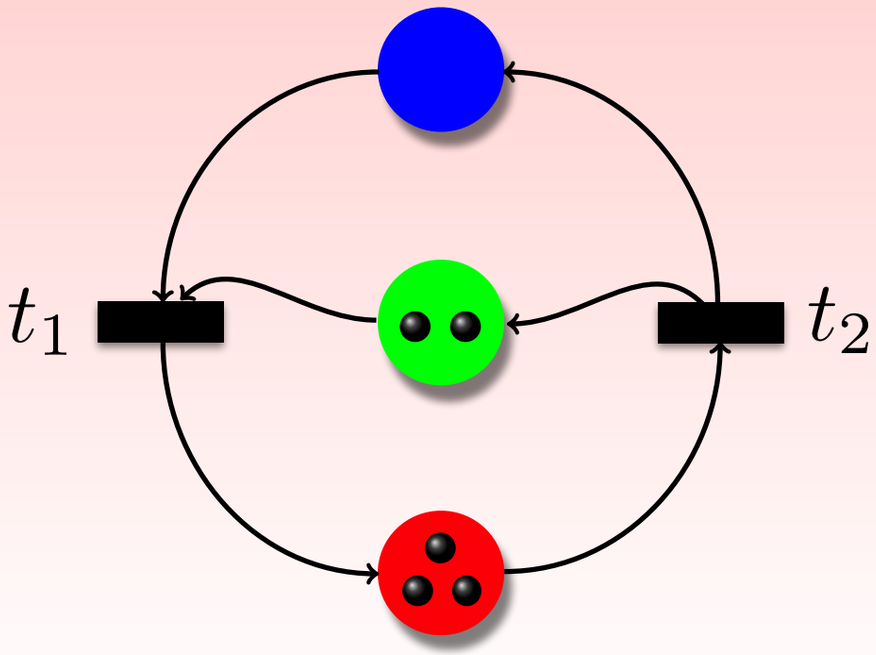
marking

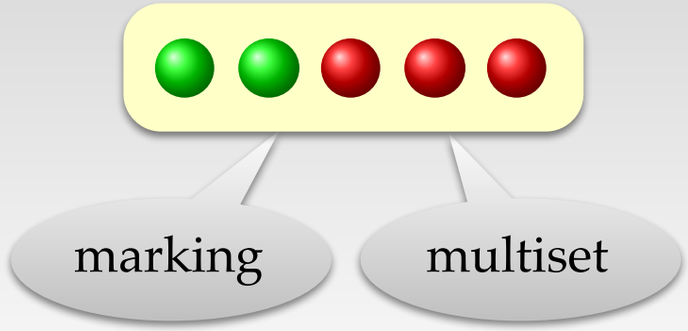
Markings





Markings





Petri Nets

Model

Configurations

Transitions

Ordering

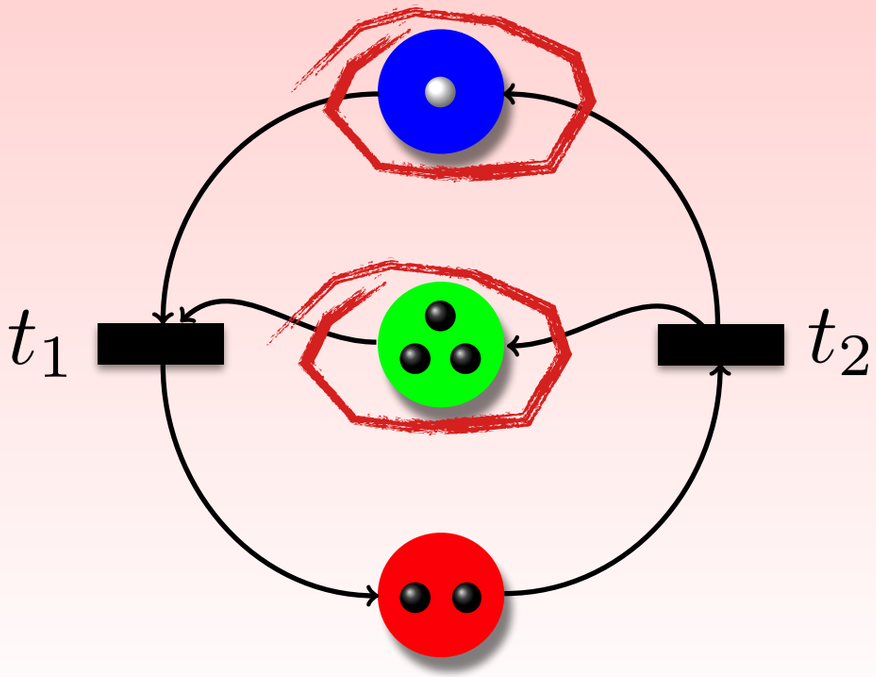
Monotoncity

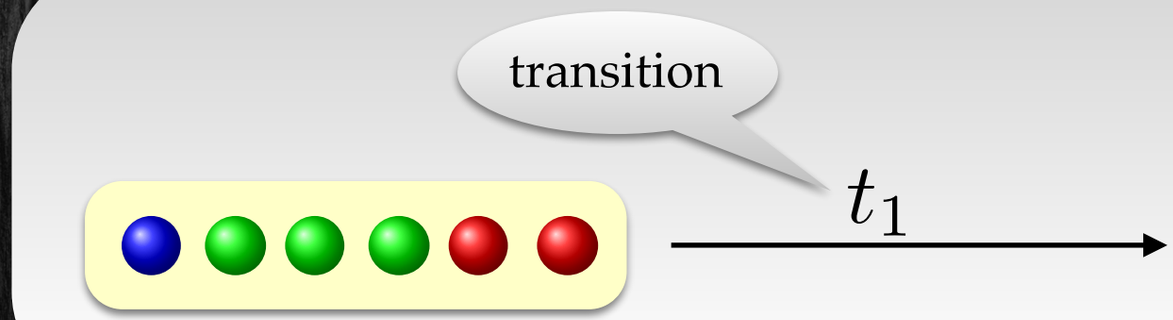
Upward Closed Sets

Computing Predecessors

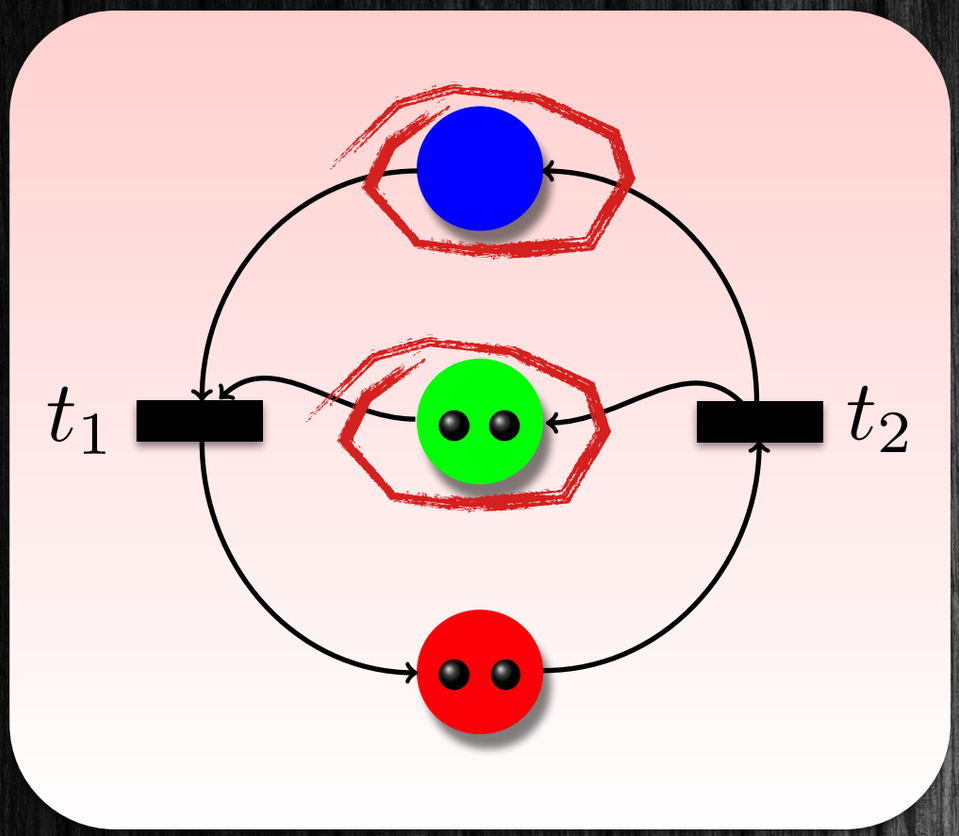
Backward Reachability

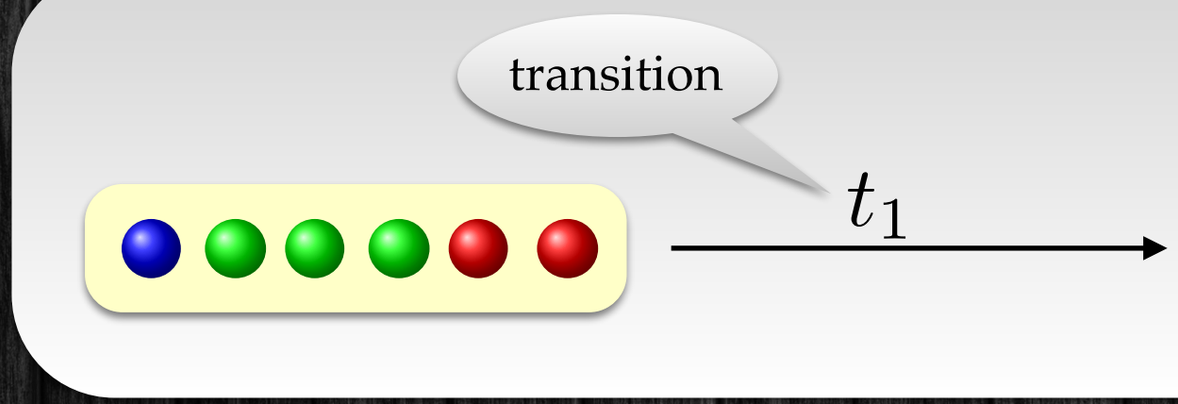
Transitions



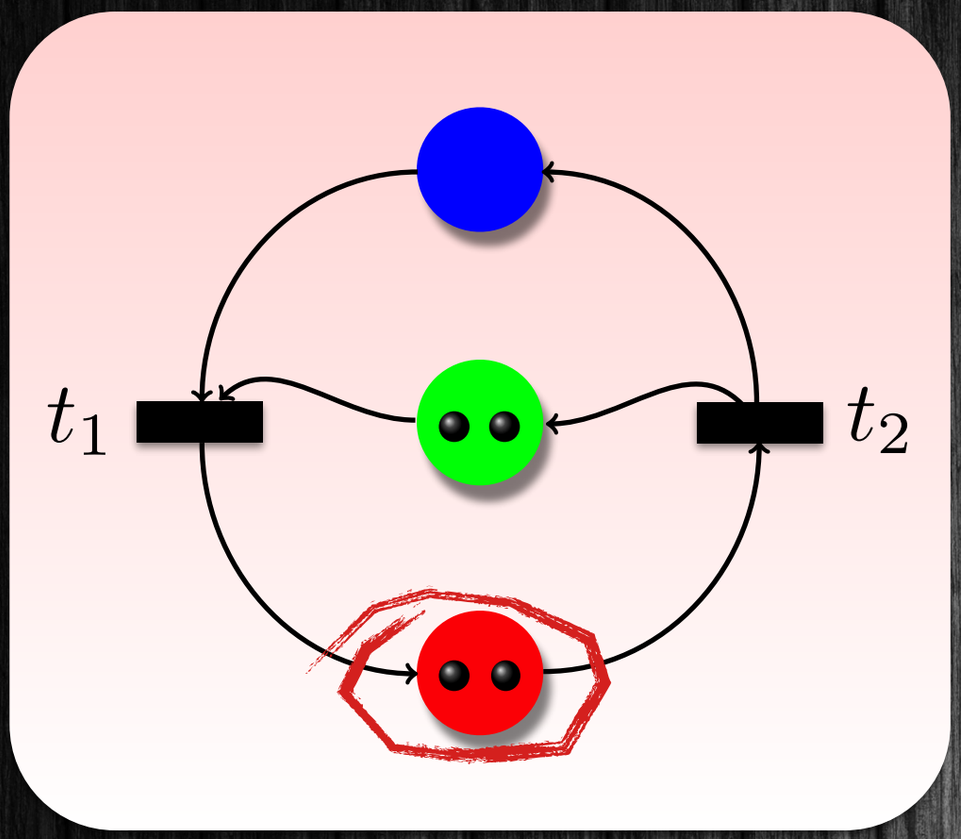


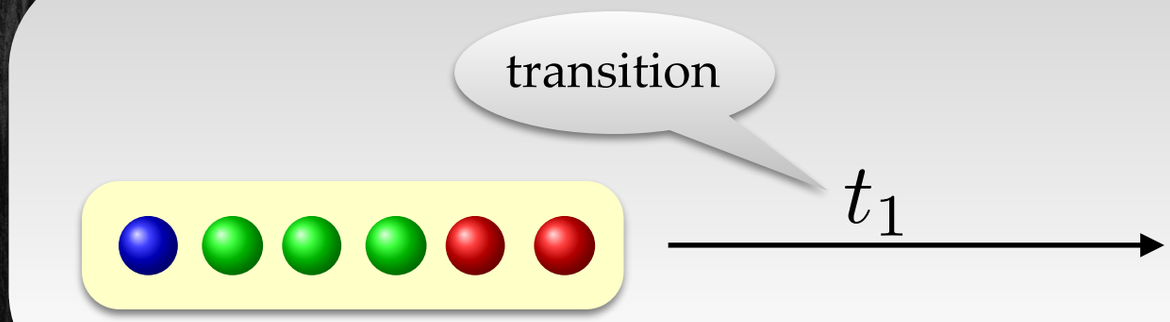
Transitions



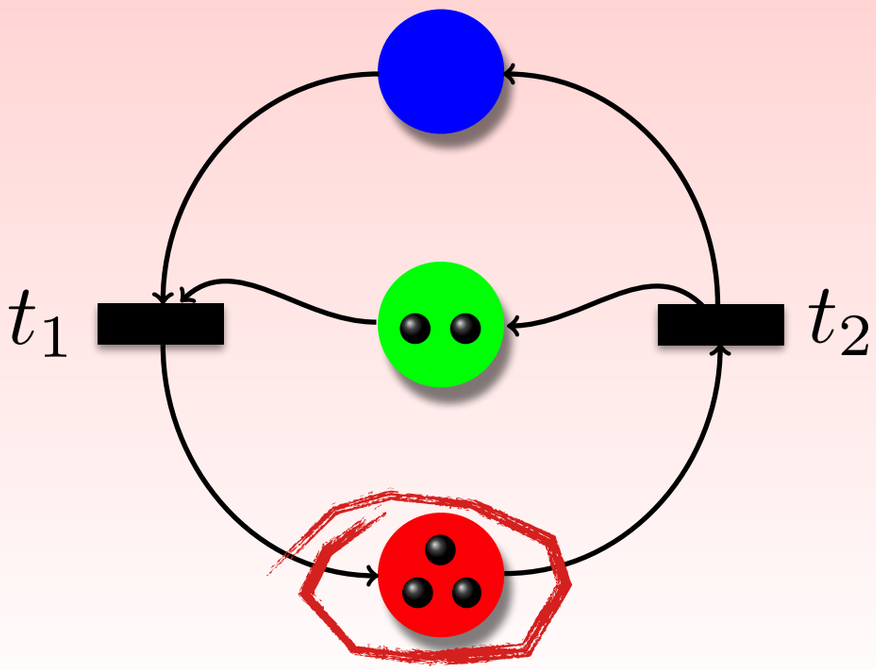


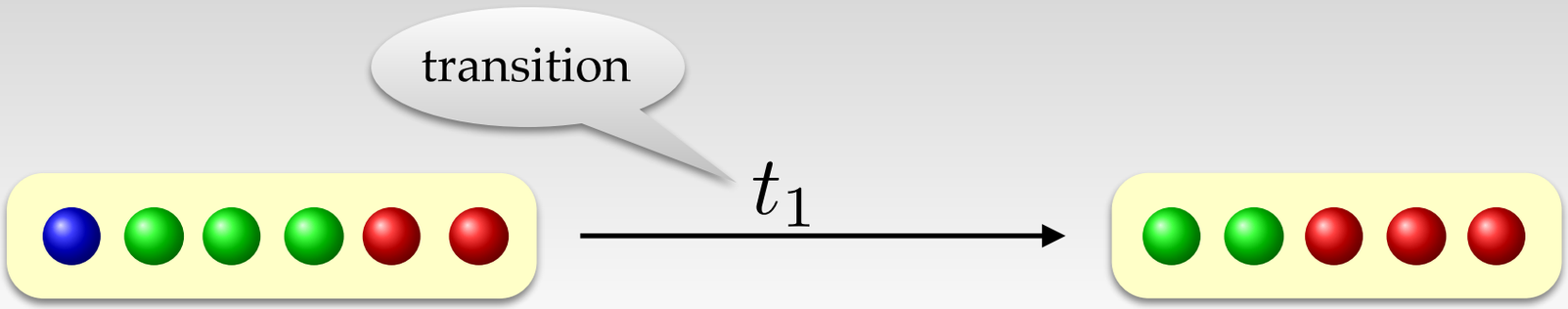
Transitions



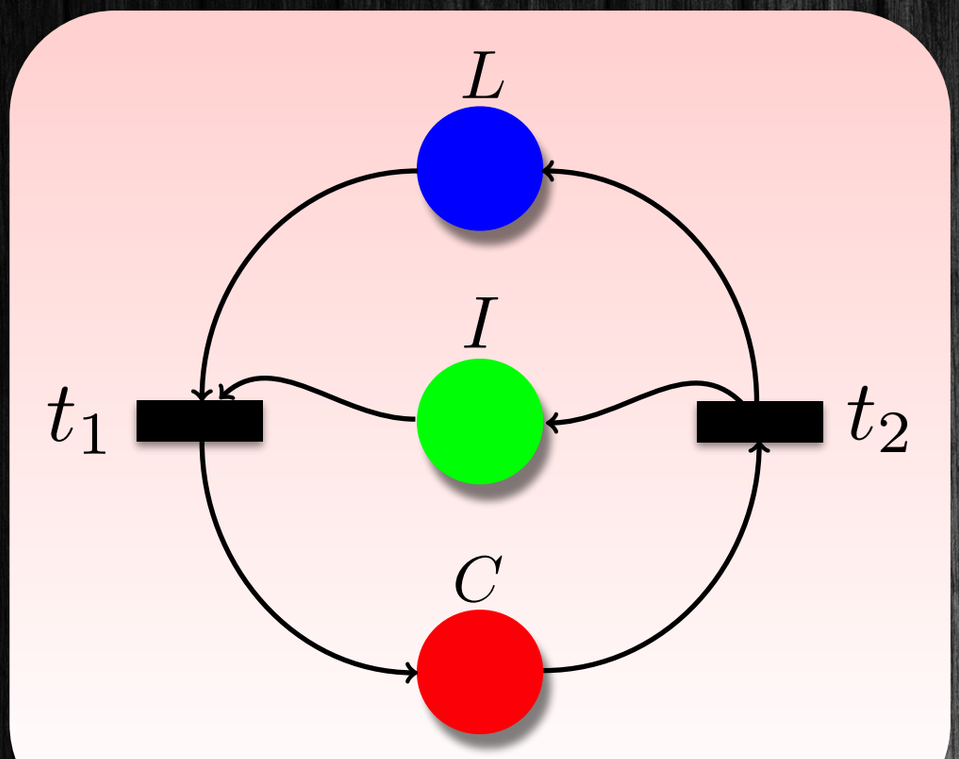


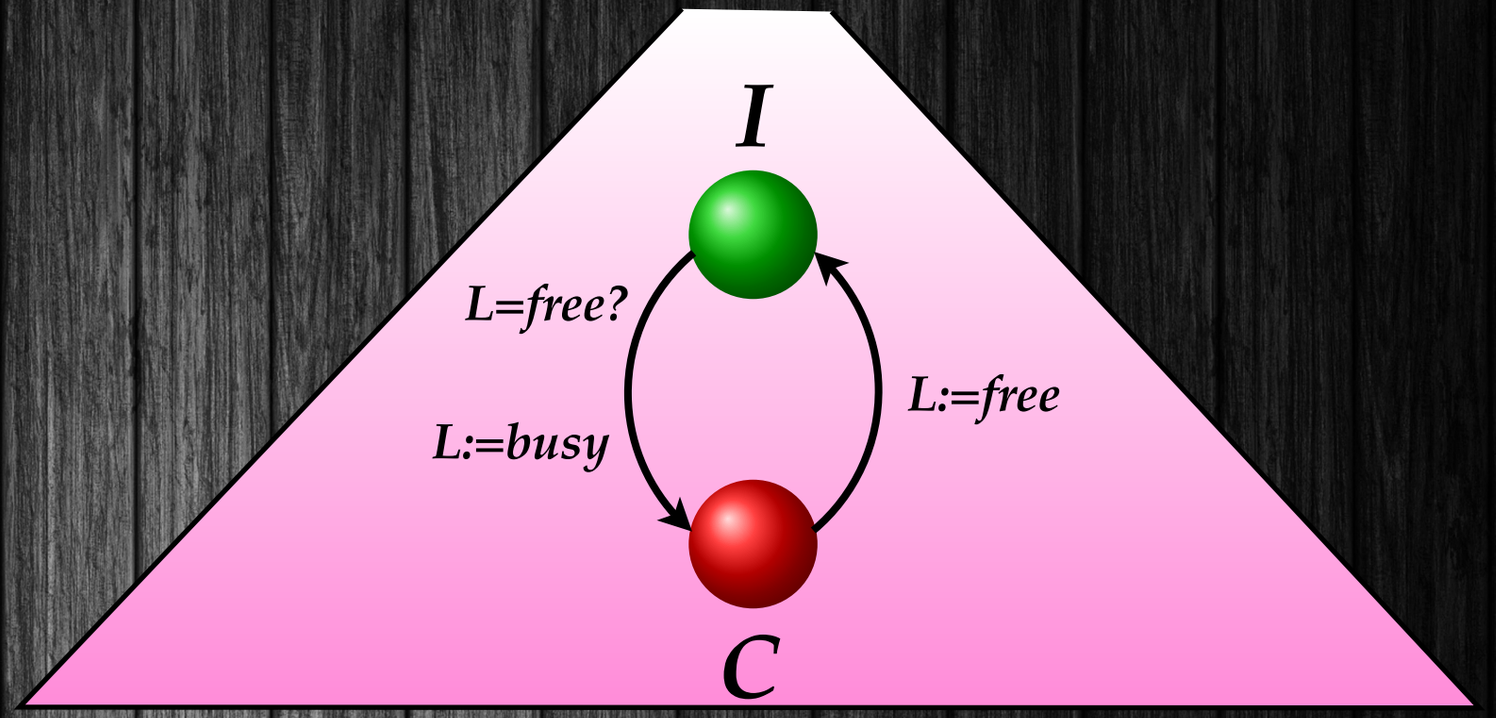
Transitions





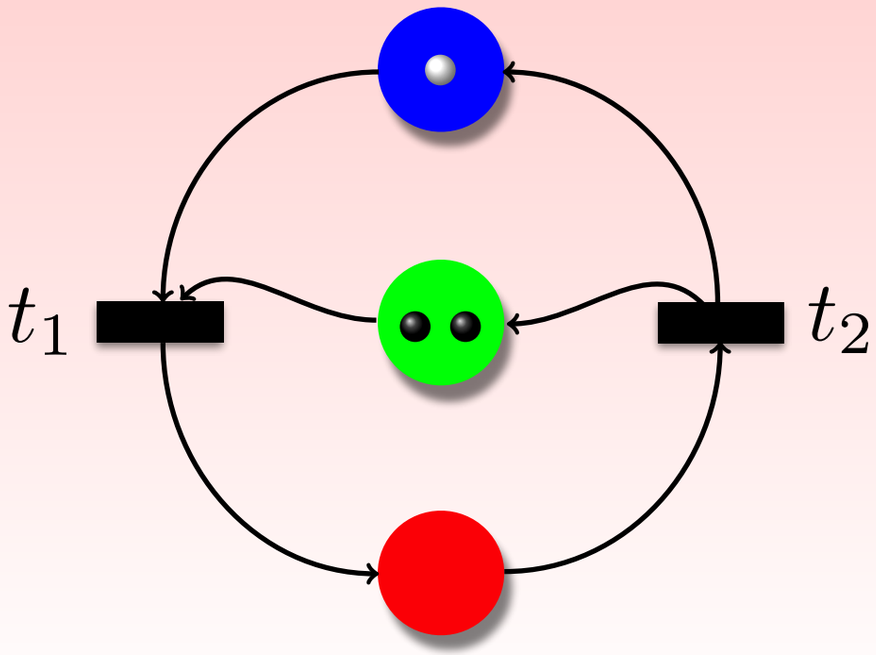
Modeling

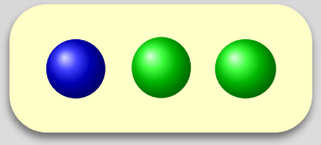




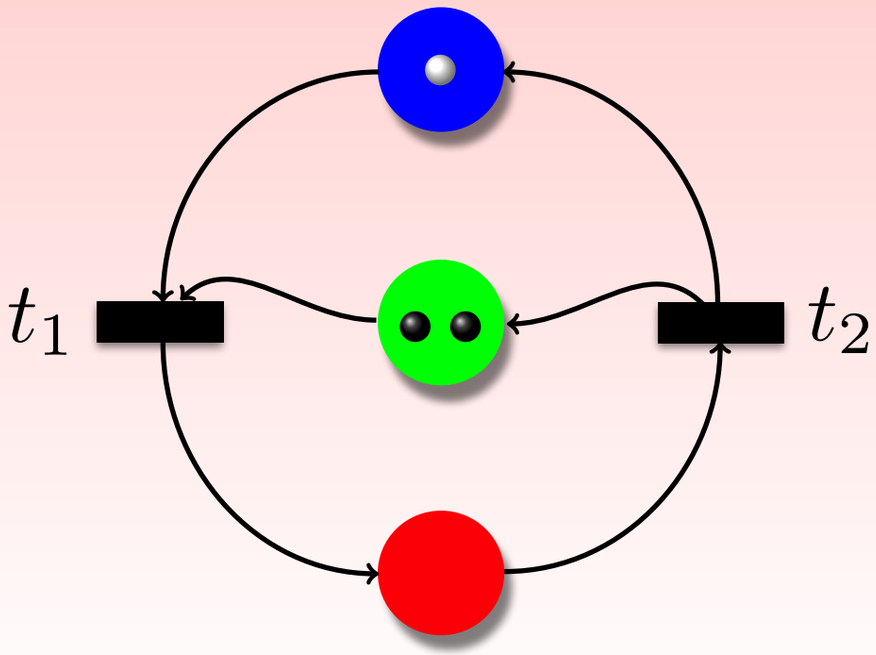
- **Encoding (counter abstraction)**
  - **# tokens in 🔴 = # processes in 🔴**
  - **# tokens in 🟢 = # processes in 🟢**
  - **one/no token in 🔵 = lock free/busy**

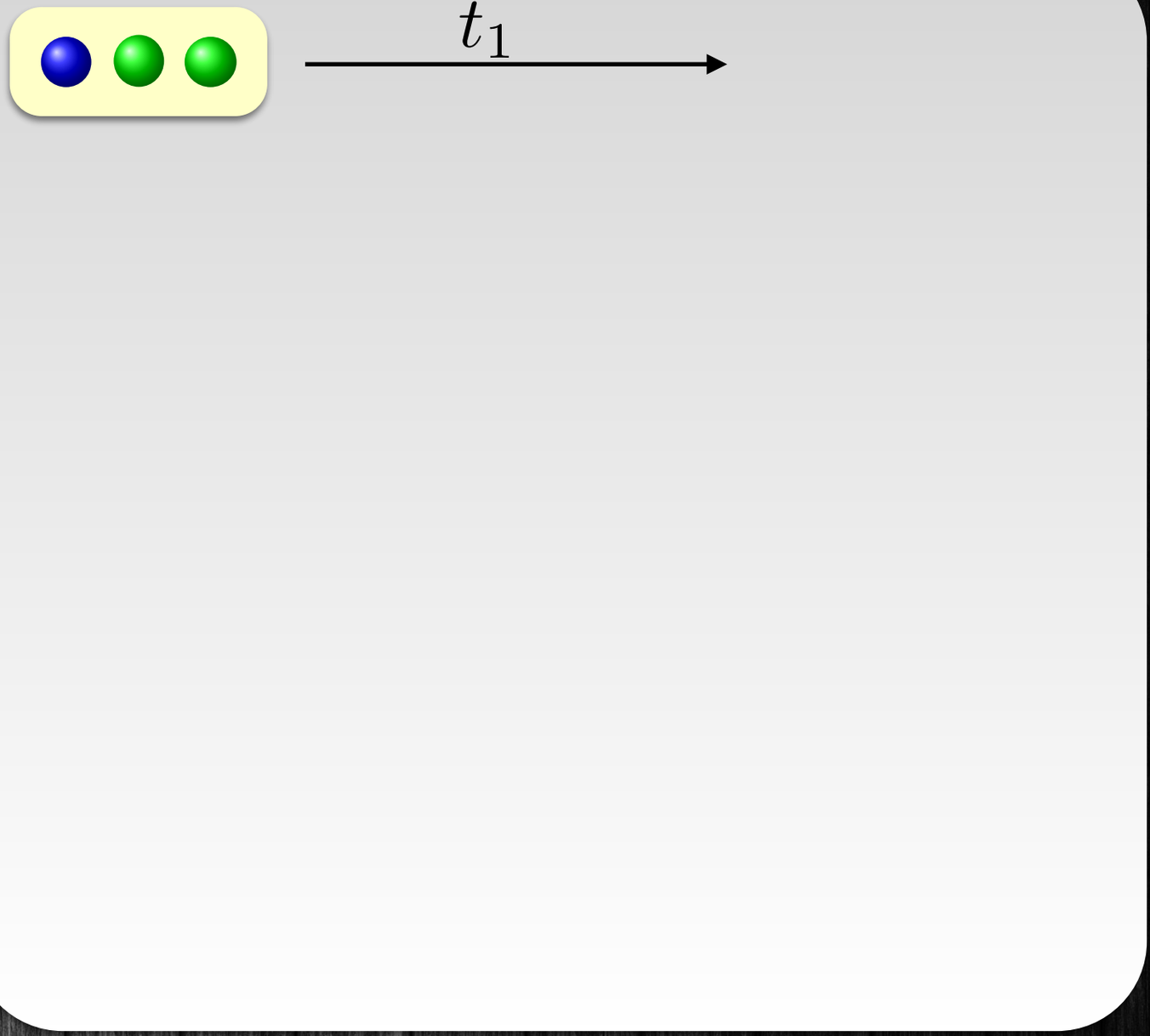
Transitions



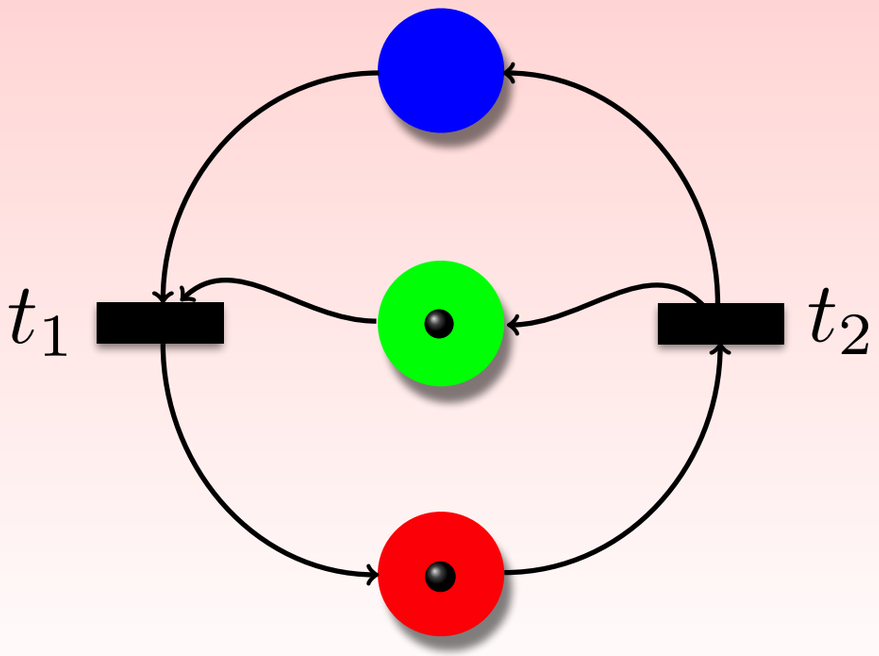


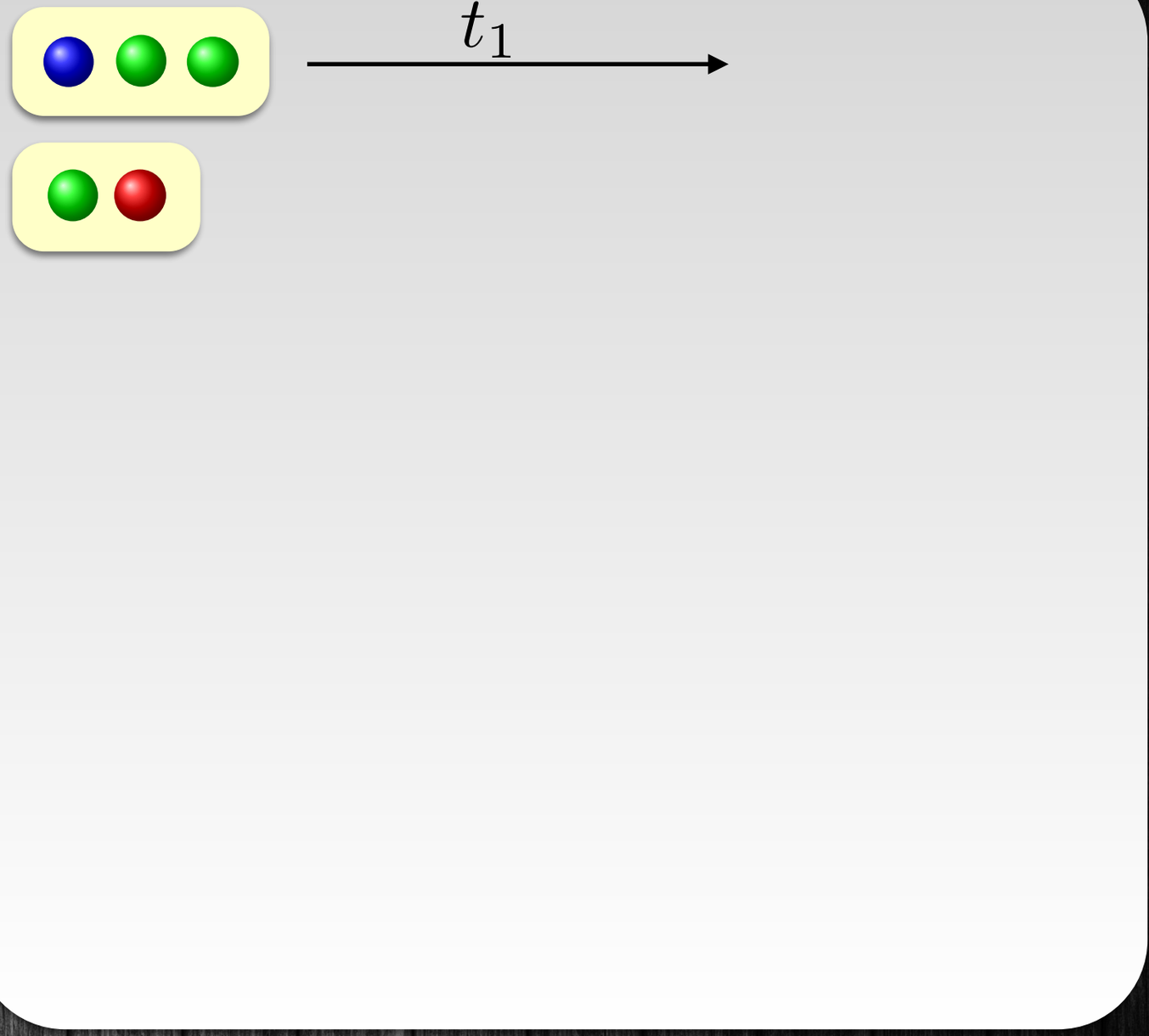
Transitions



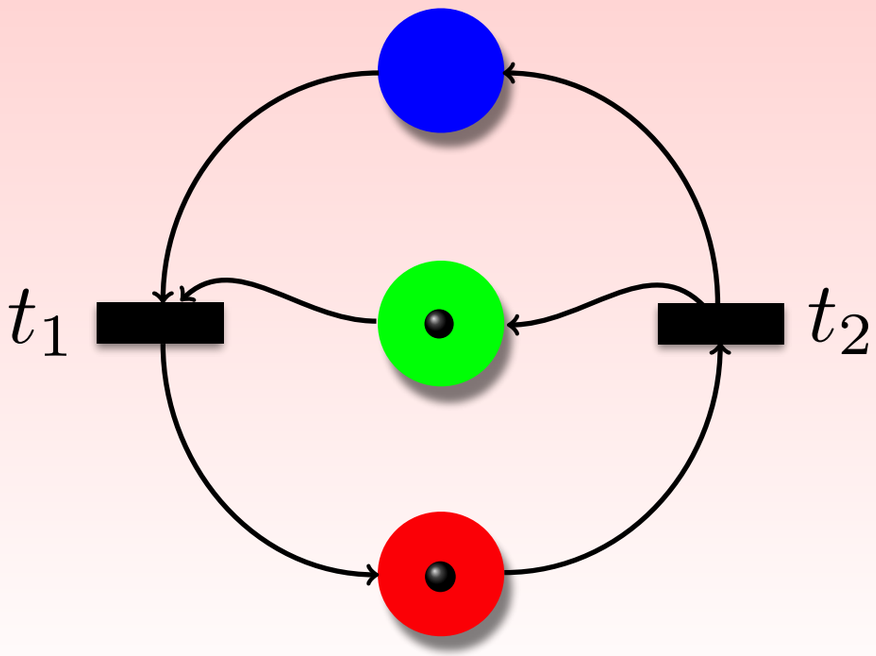


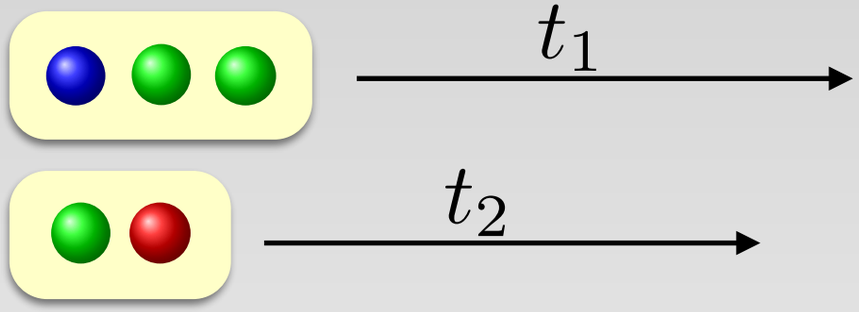
Transitions



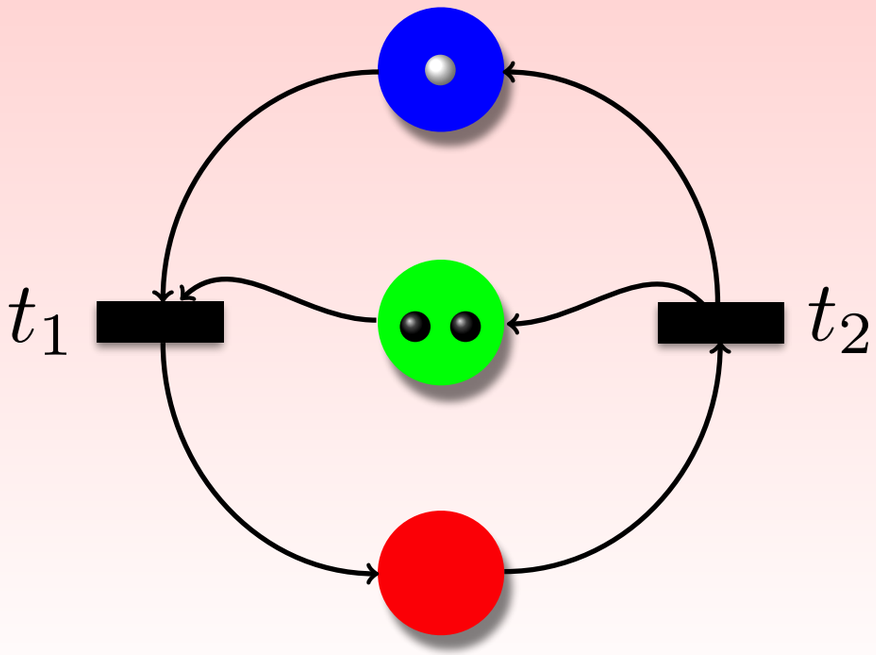


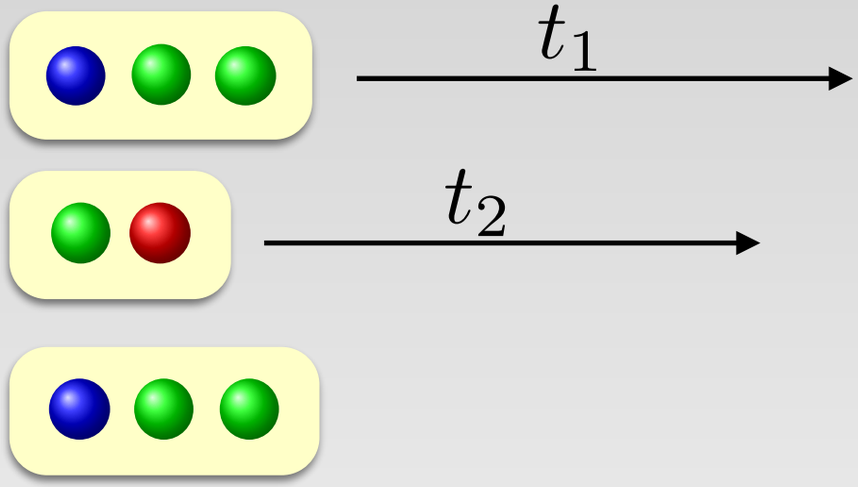
Transitions



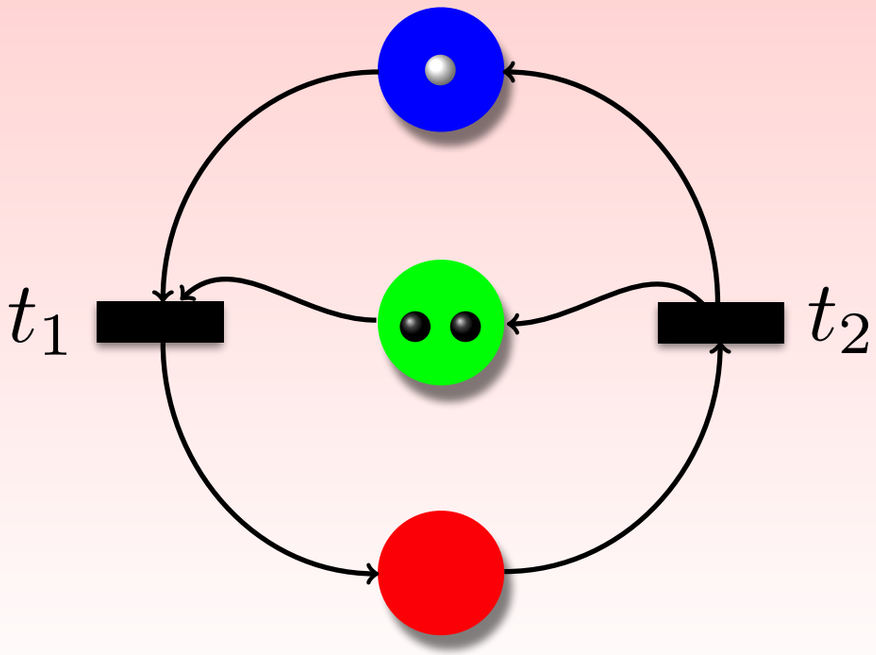


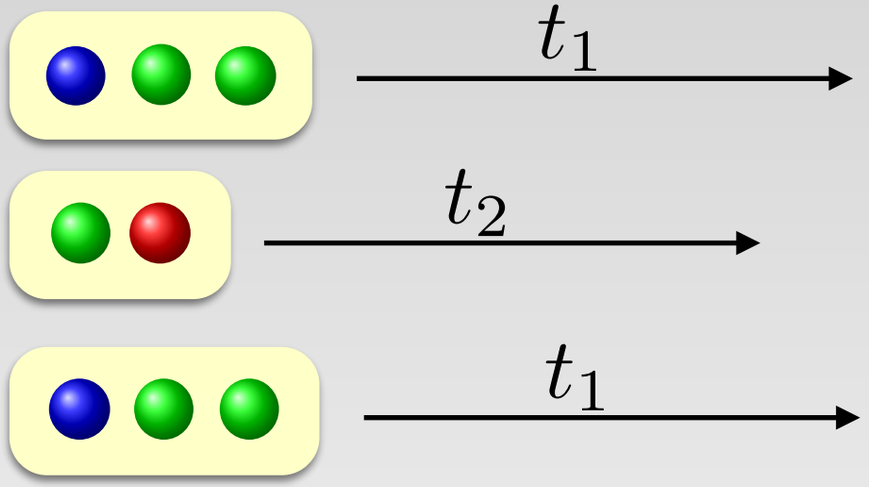
Transitions



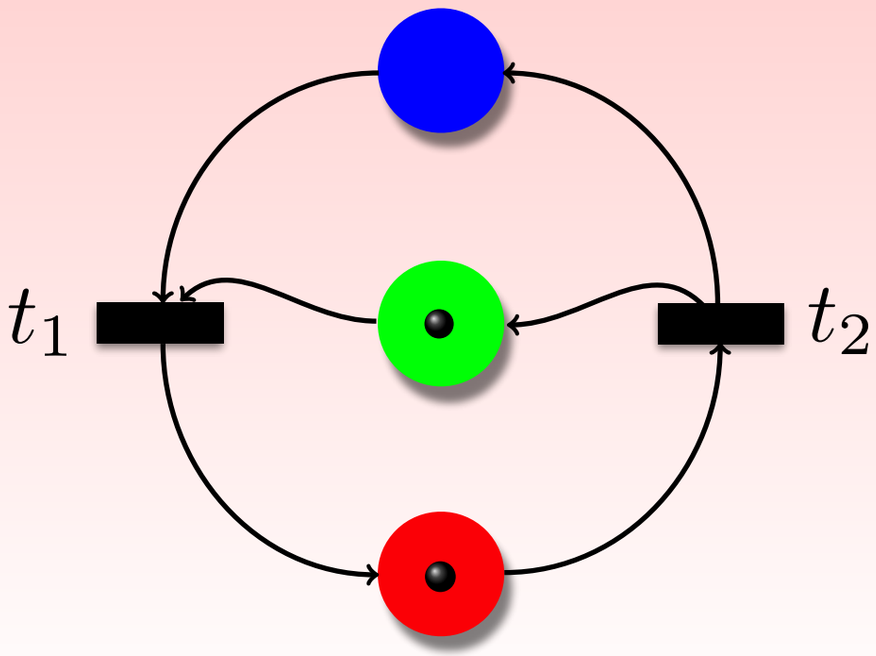


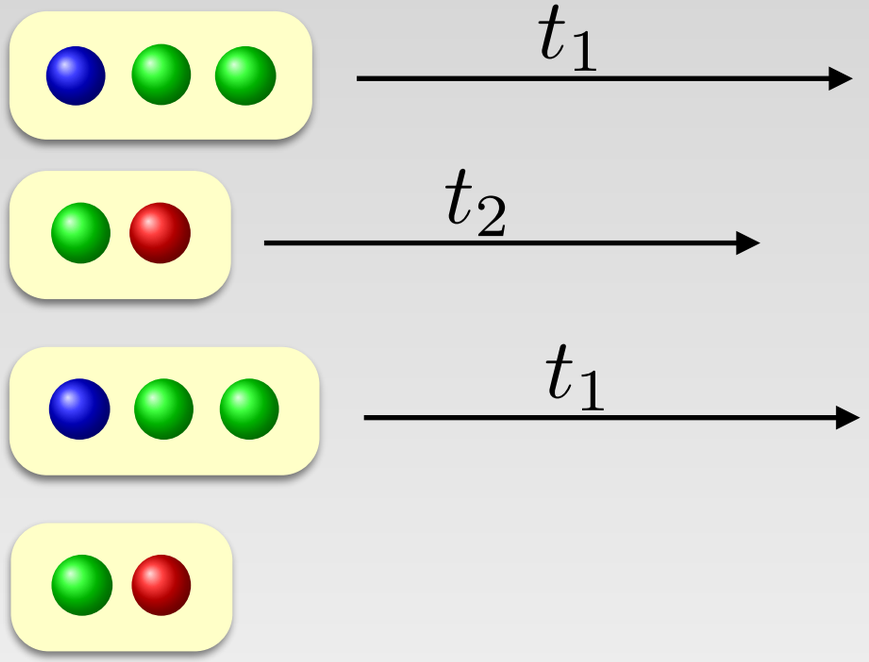
Transitions



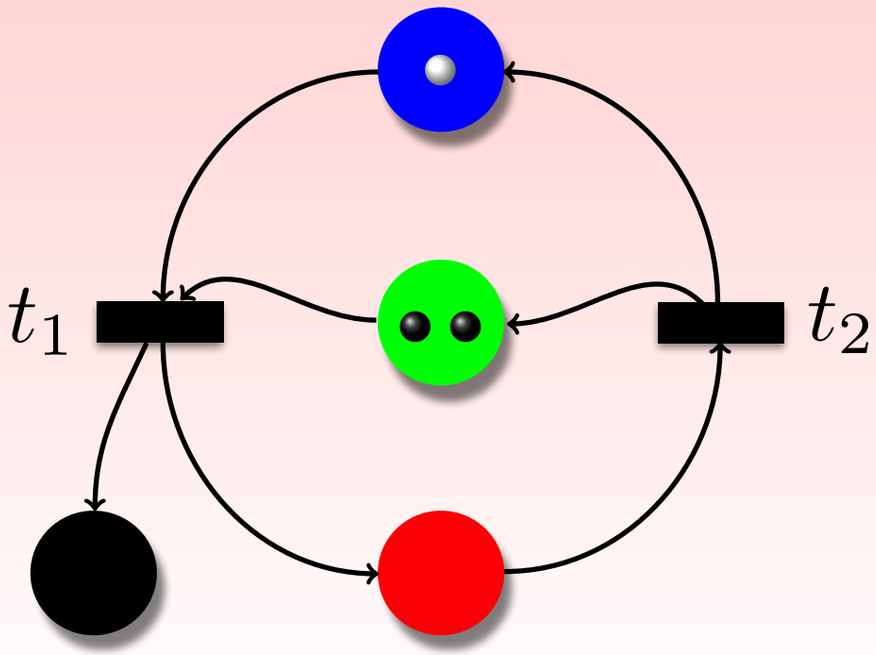


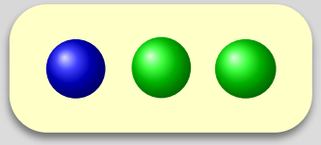
Transitions



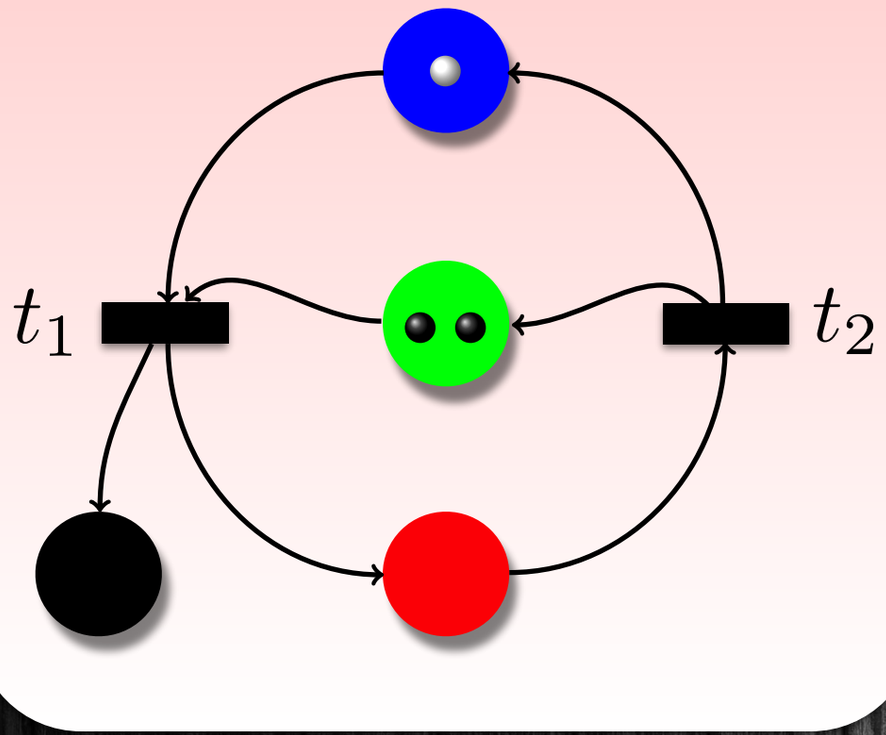


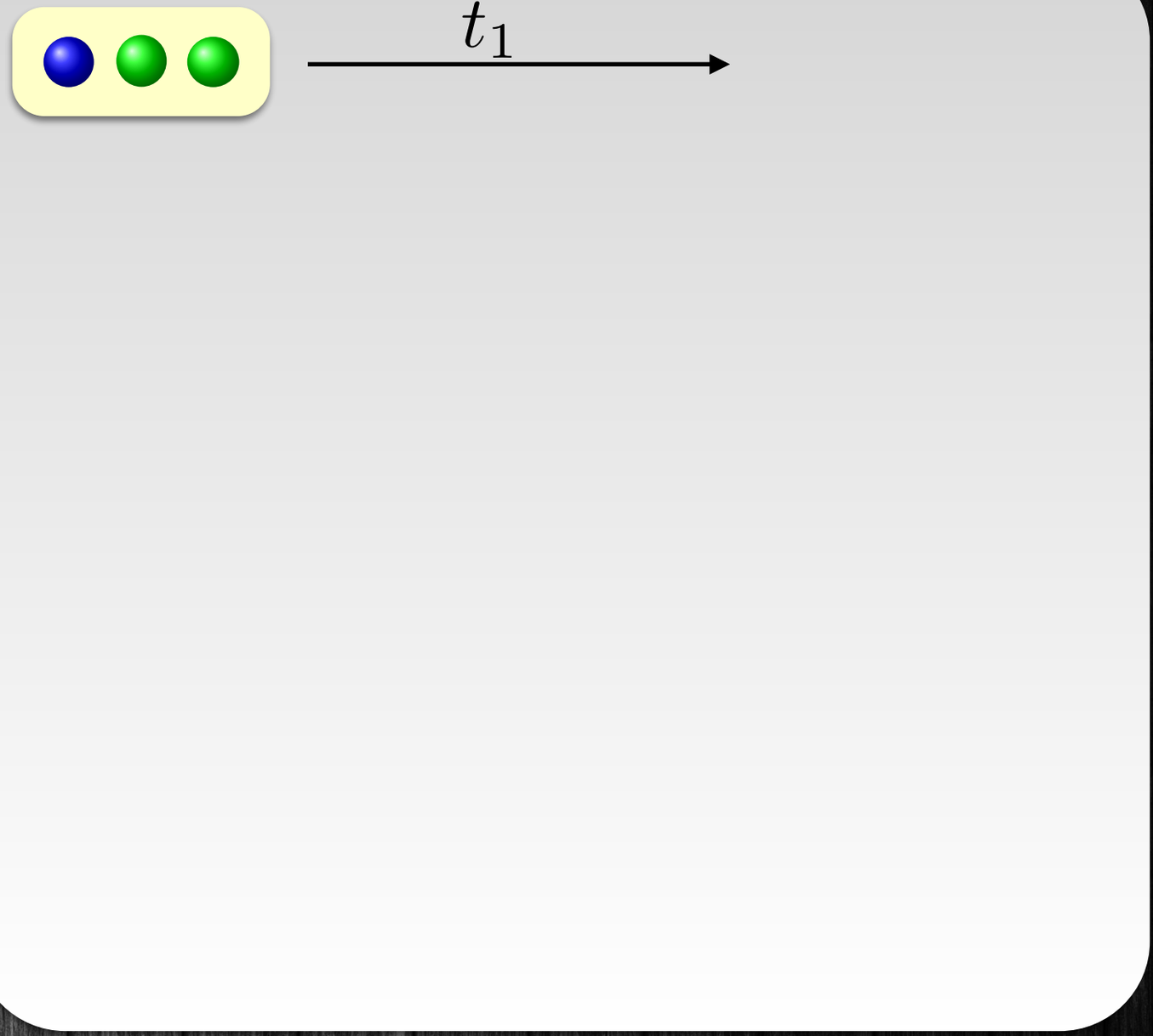
Transitions



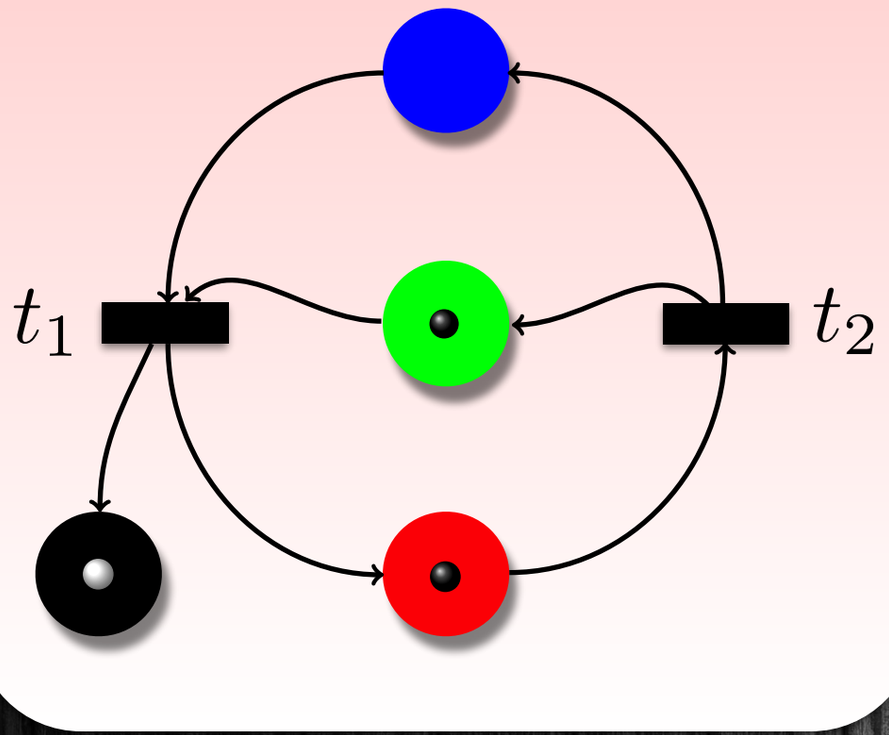


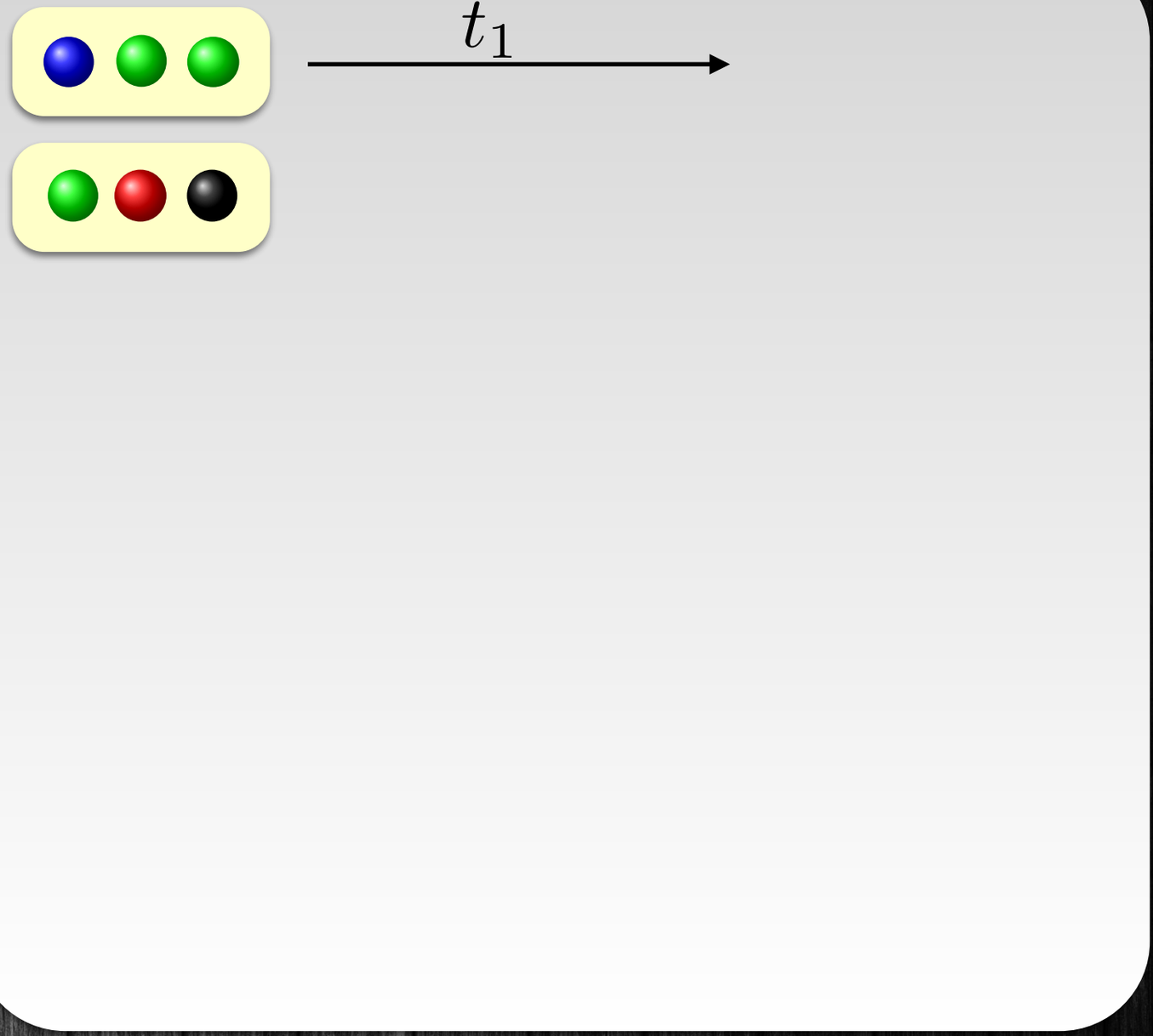
Transitions



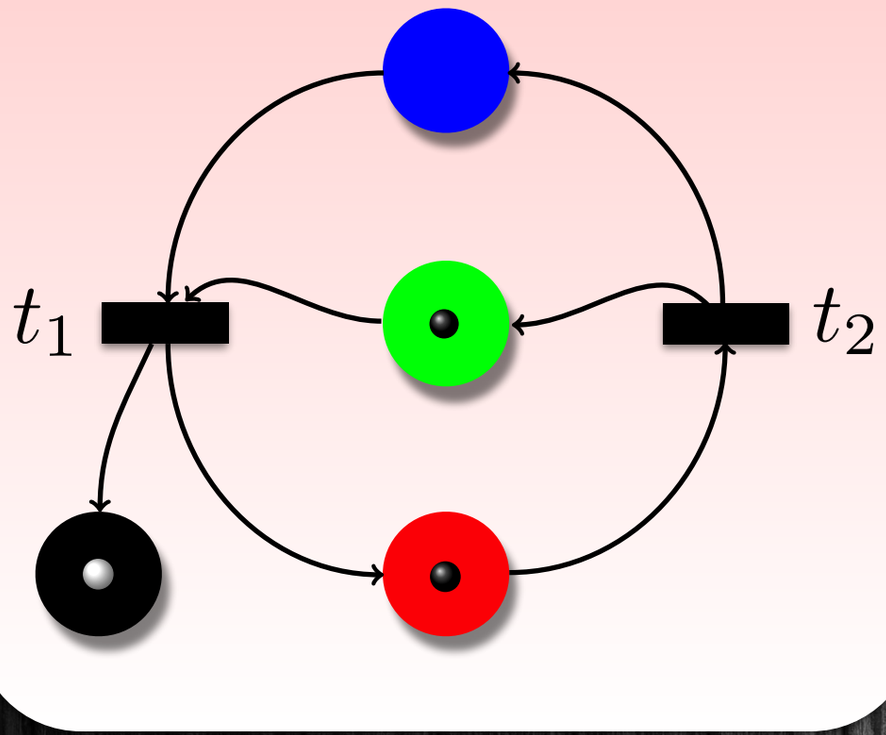


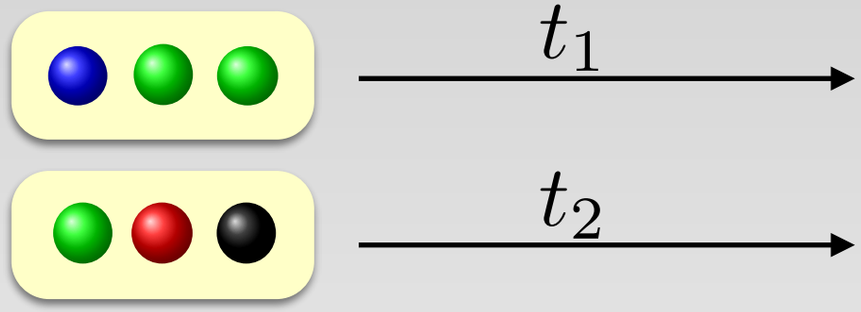
Transitions



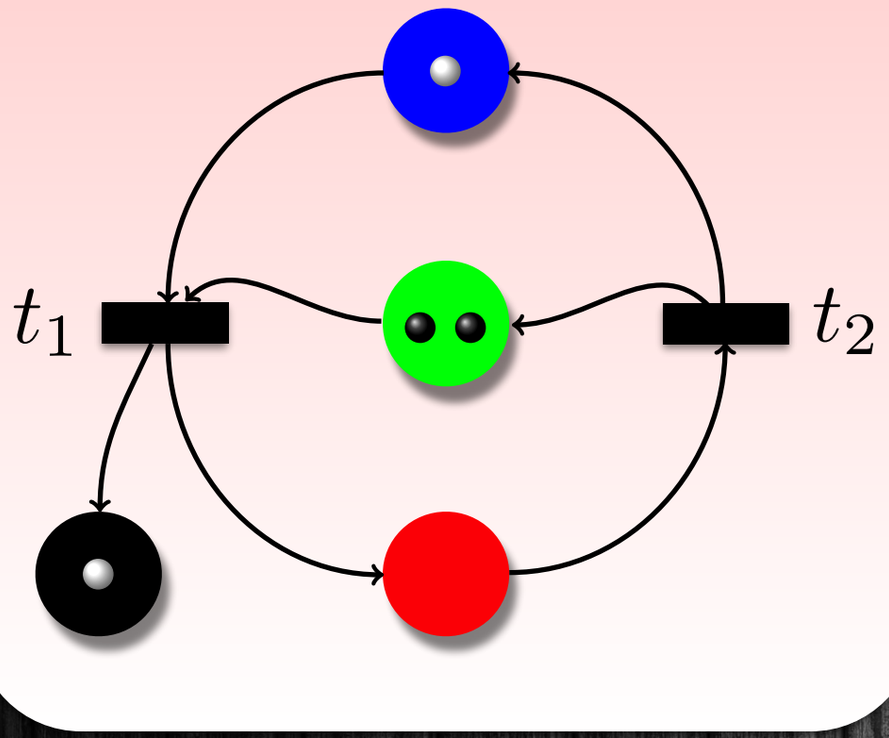


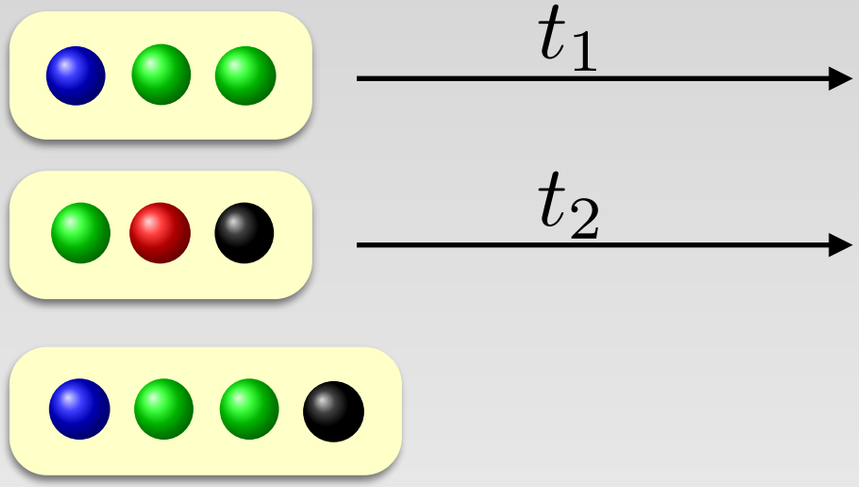
Transitions



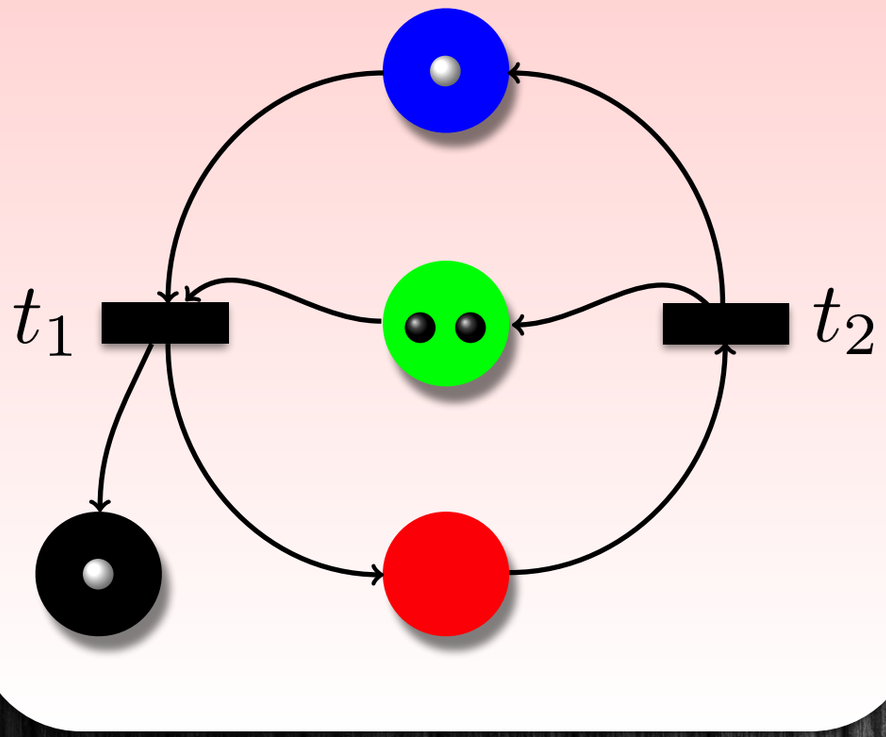


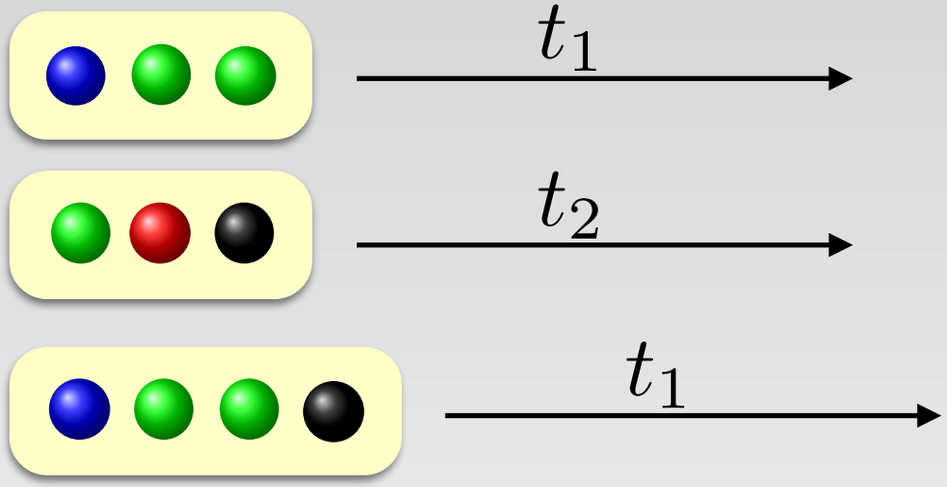
Transitions



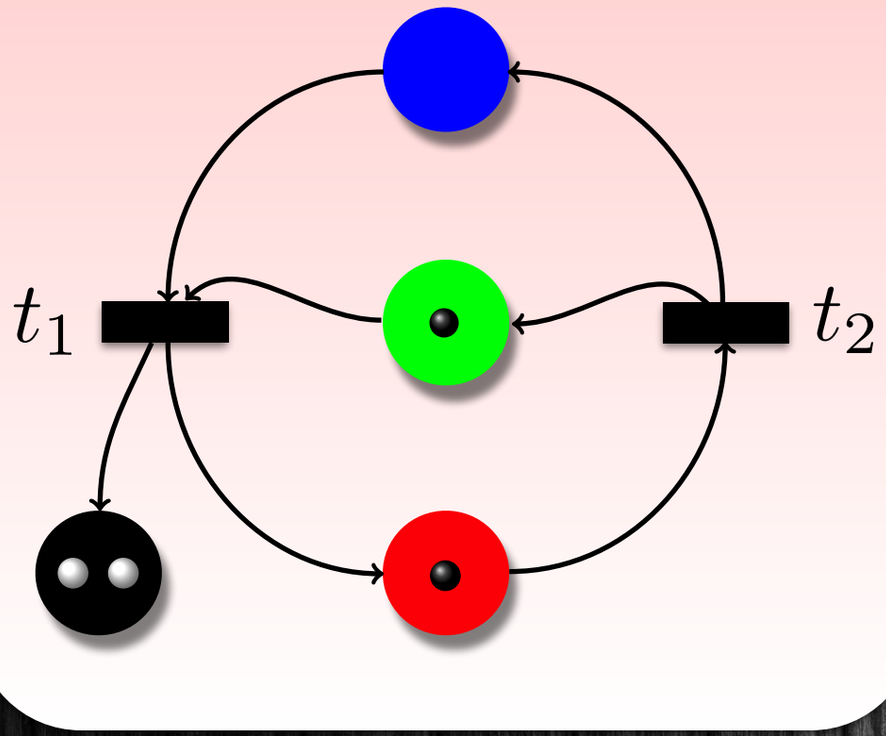


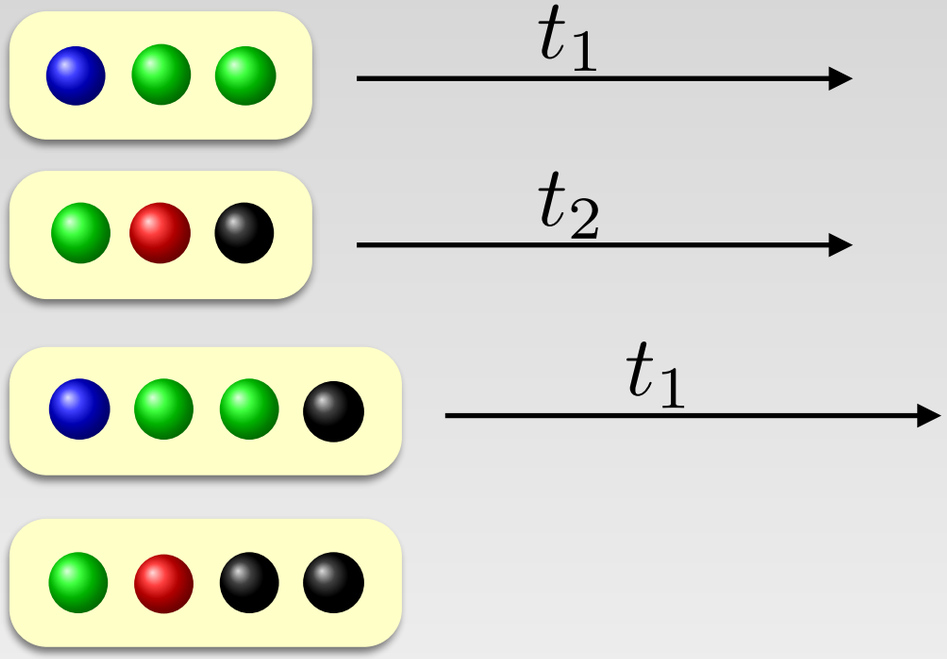
Transitions



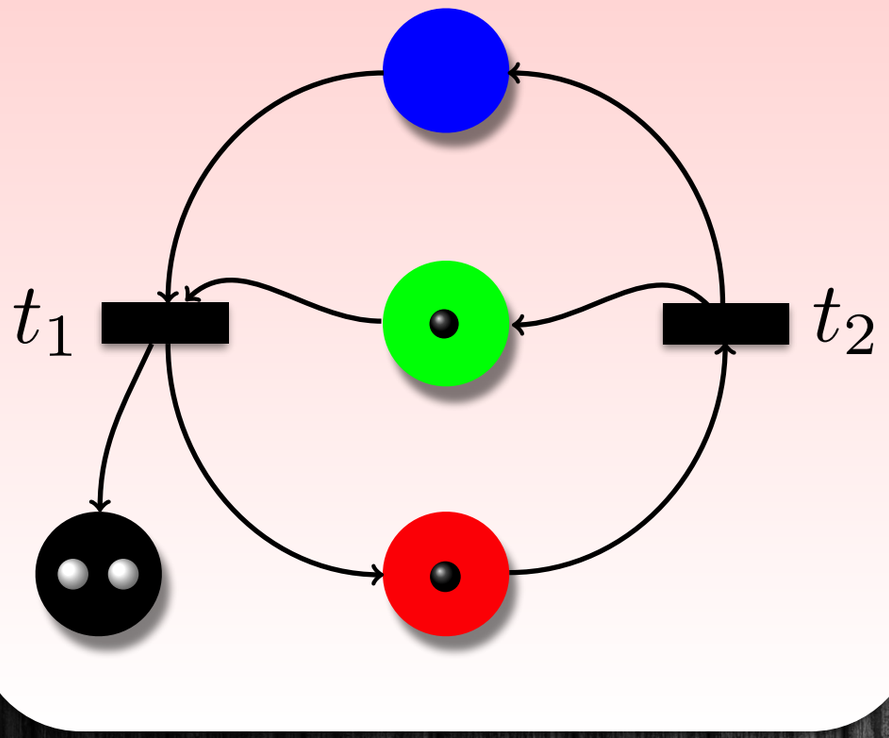


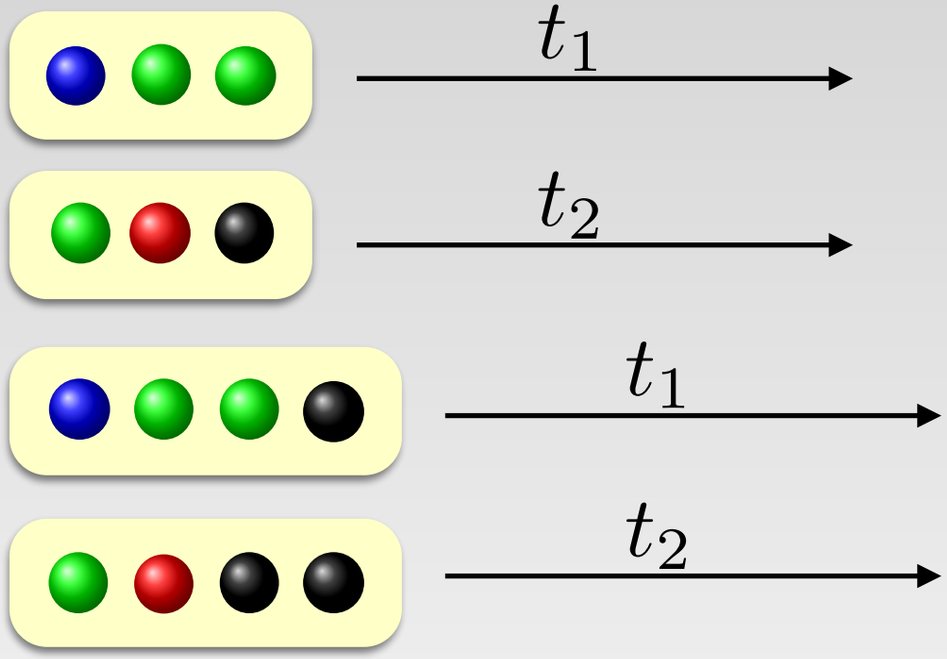
Transitions



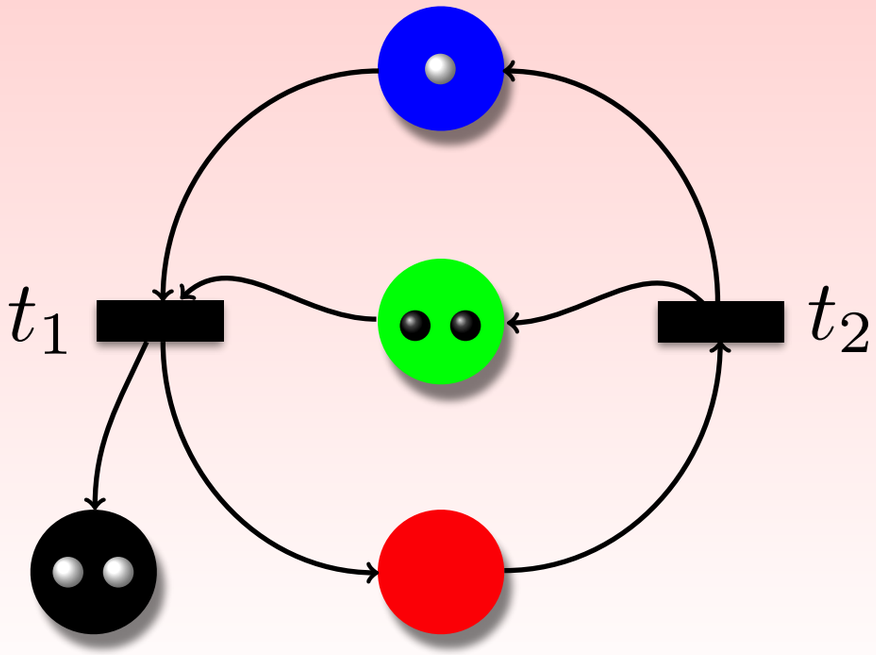


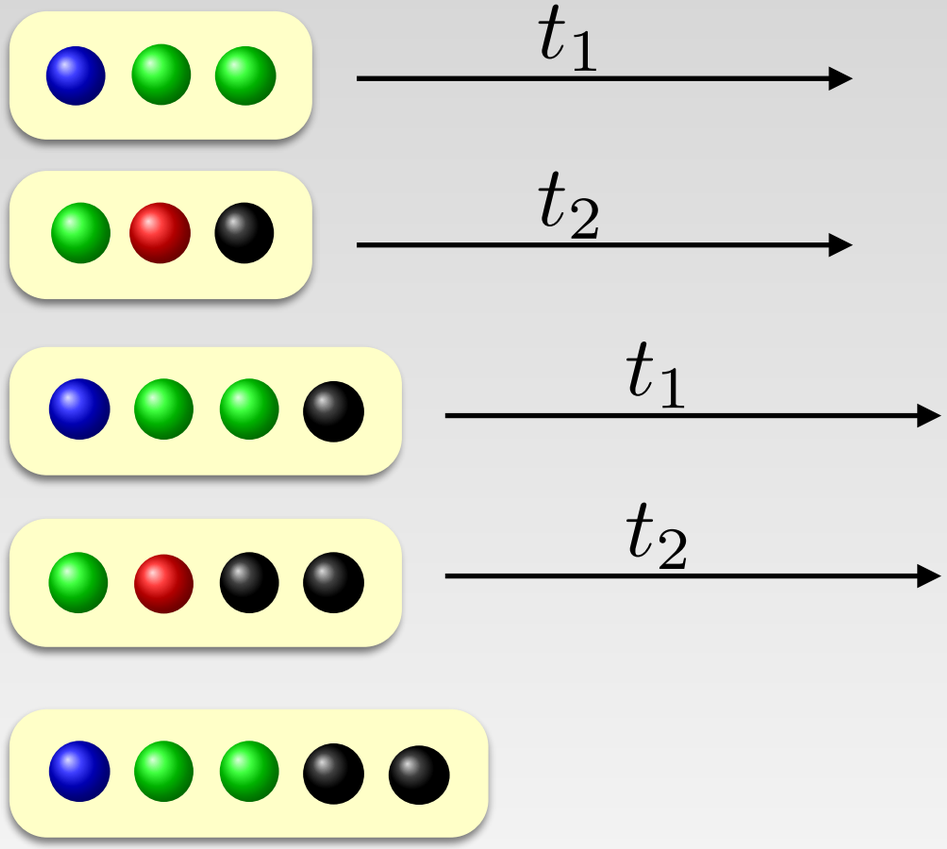
Transitions



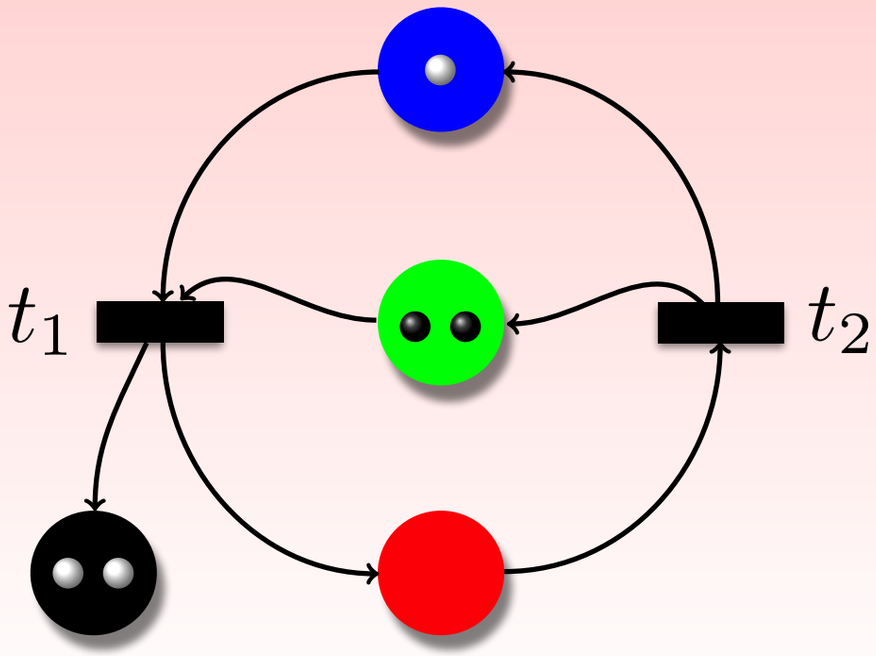


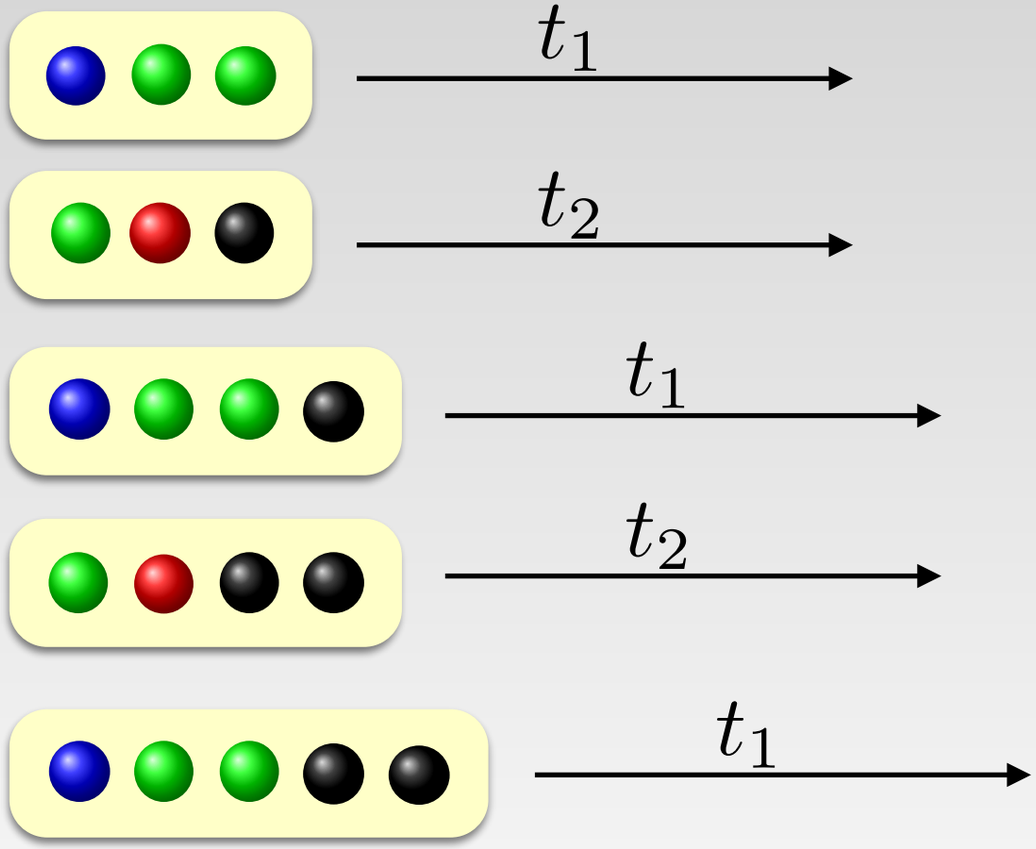
Transitions



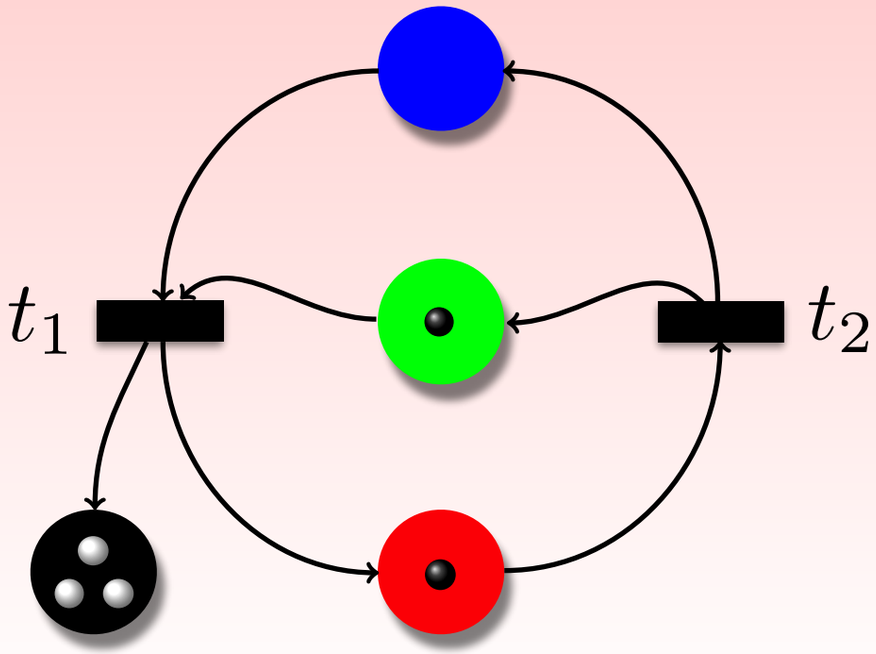


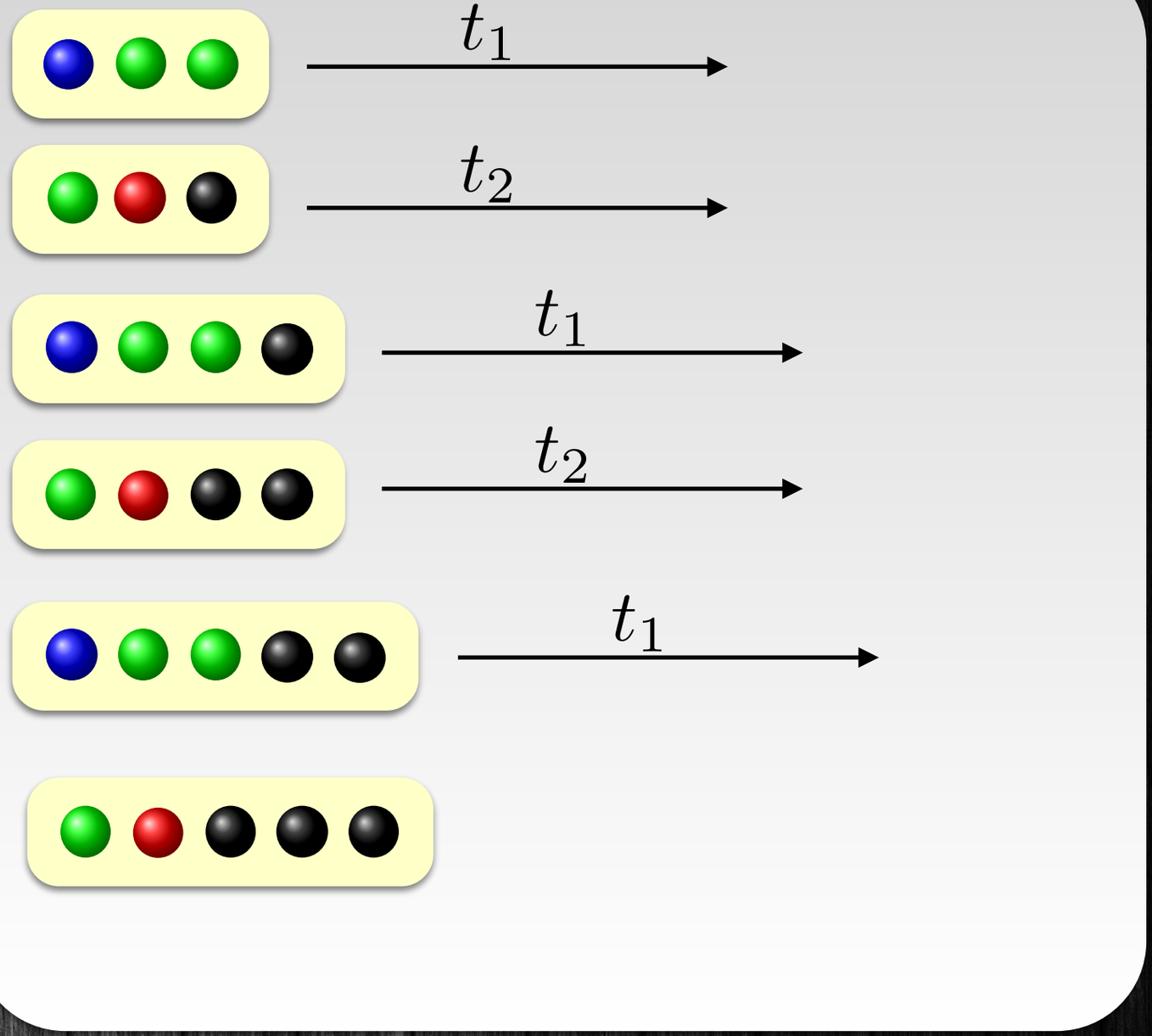
Transitions



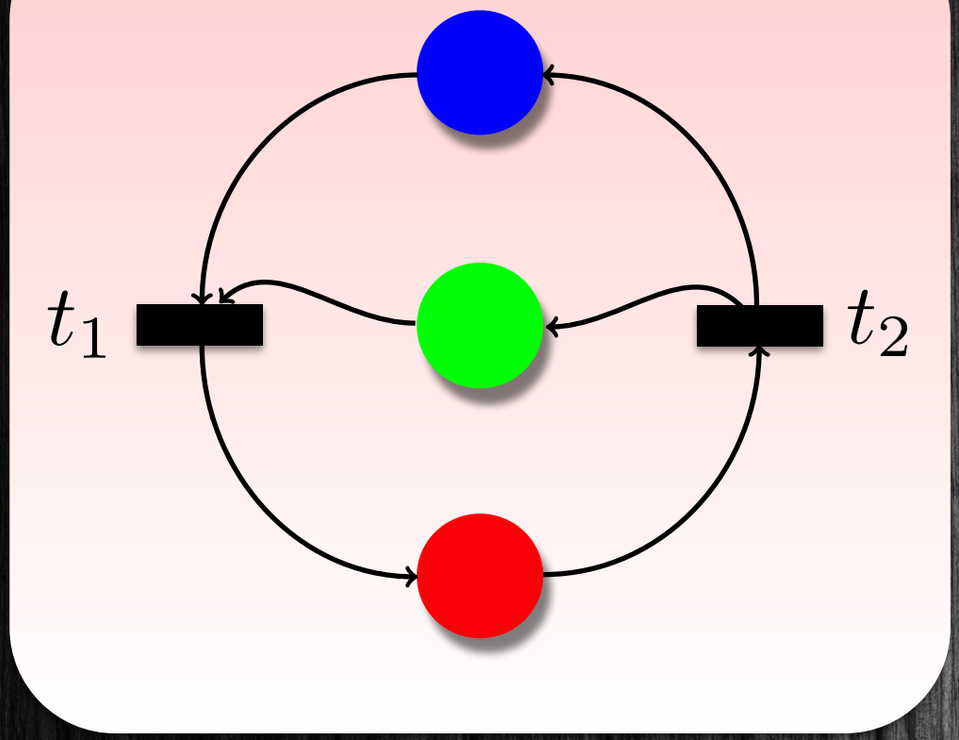


Transitions

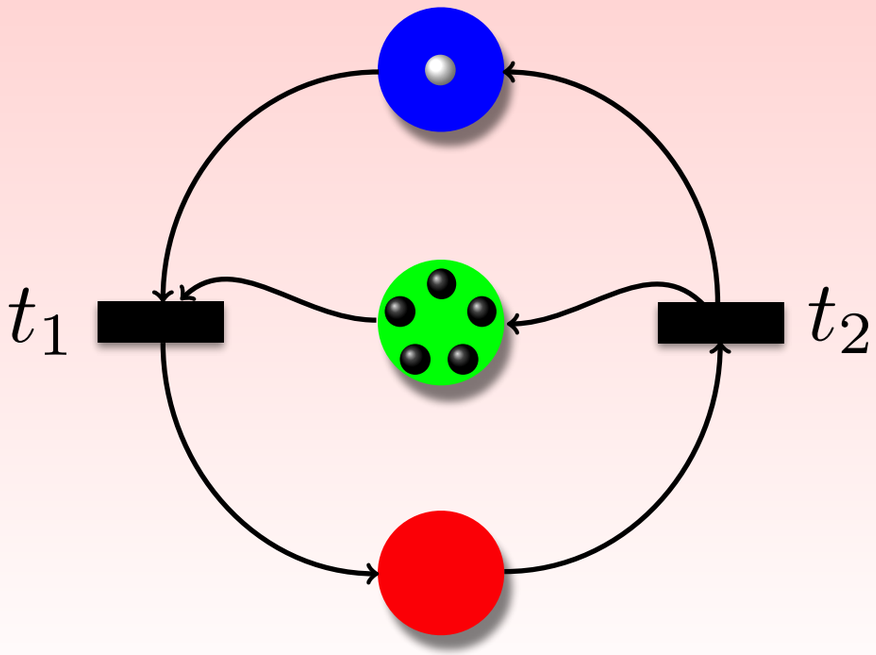




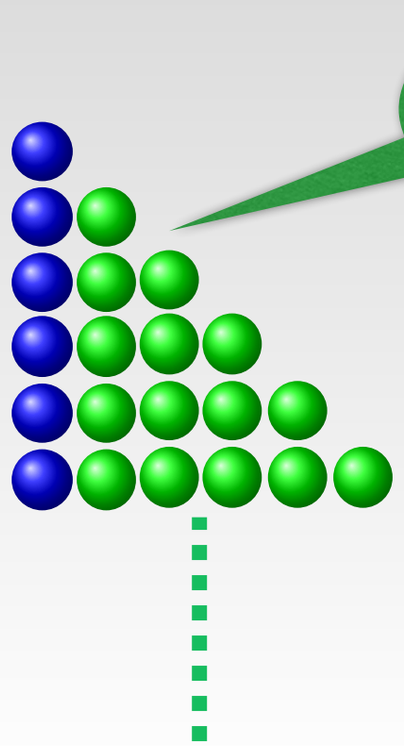
Safety Properties



Safety Properties

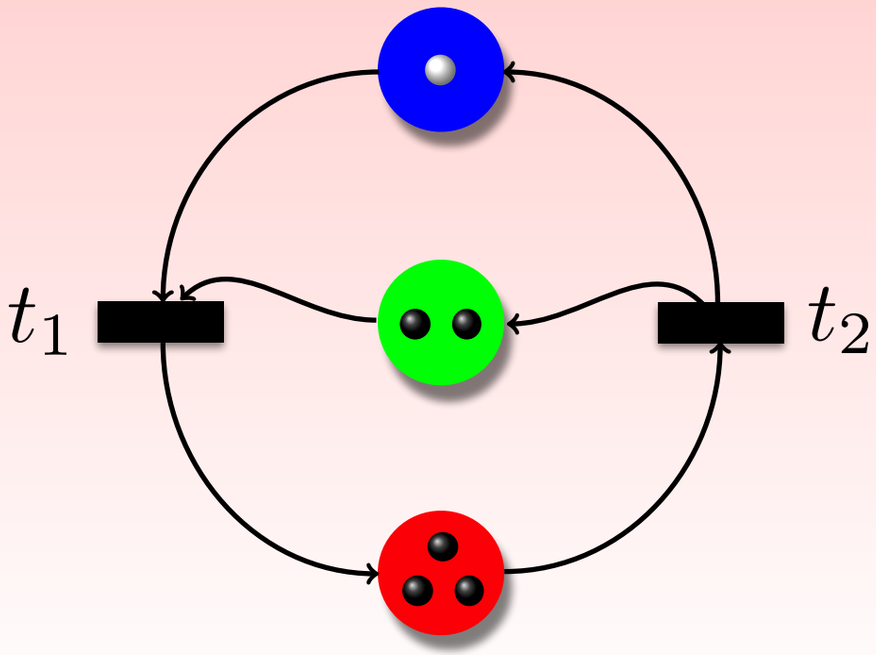


Initial Markings (*Init*)

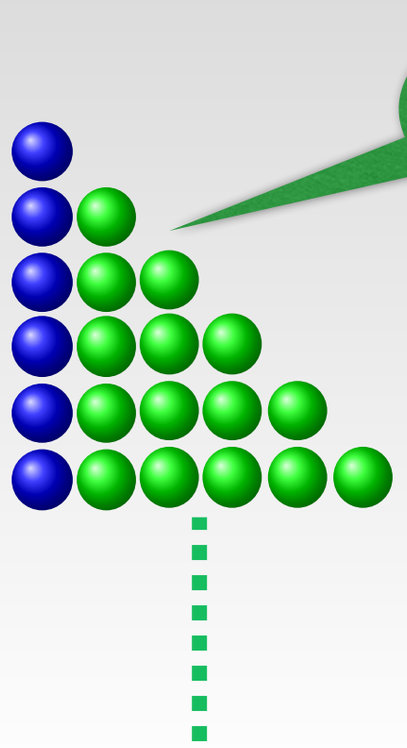


- one
- arbitrarily many

Safety Properties

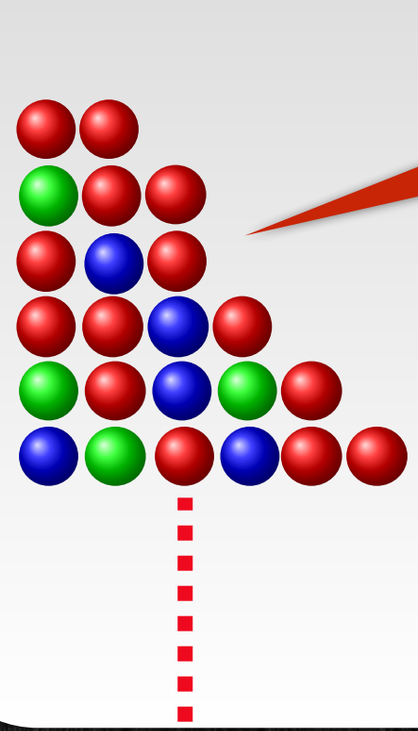


Initial Markings (*Init*)



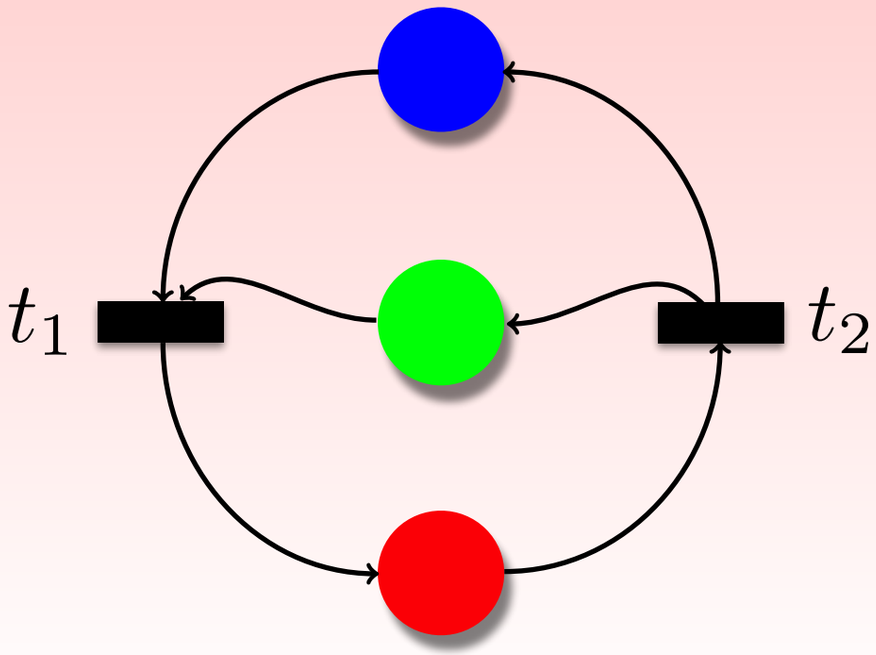
- one
- arbitrarily many

Bad Markings (*Bad*)



- at least two

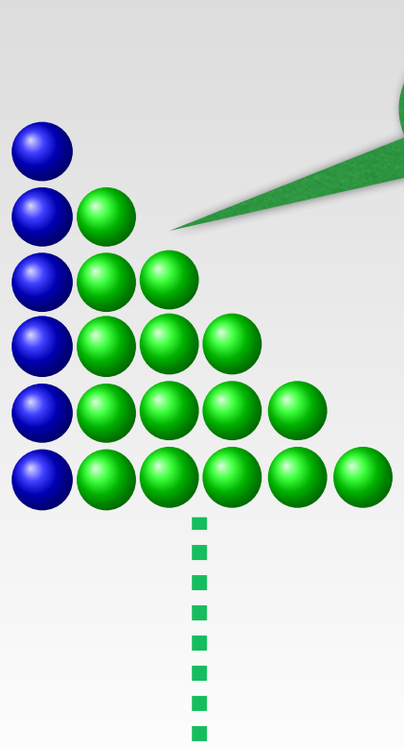
Safety Properties



Safety Property

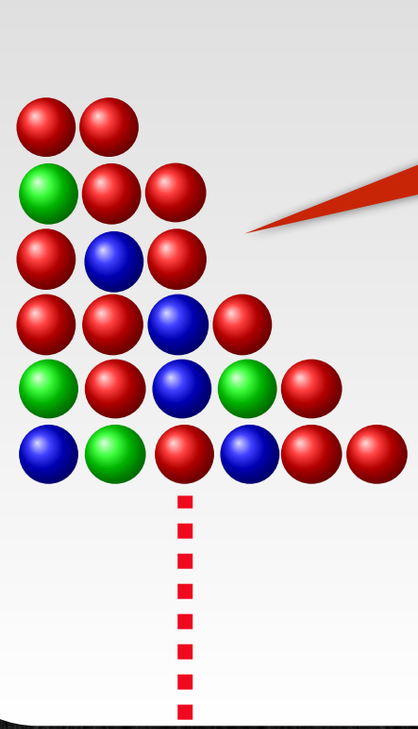
$$\mathit{Init} \xrightarrow{\ *\ } \mathit{Bad}\ ?$$

Initial Markings (*Init*)



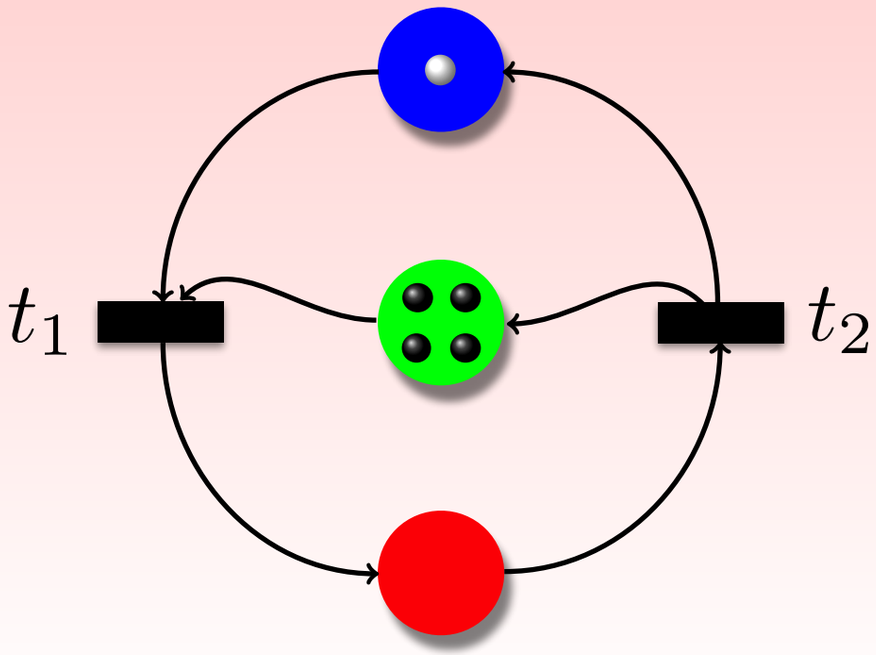
- one
- arbitrarily many

Bad Markings (*Bad*)



- at least two

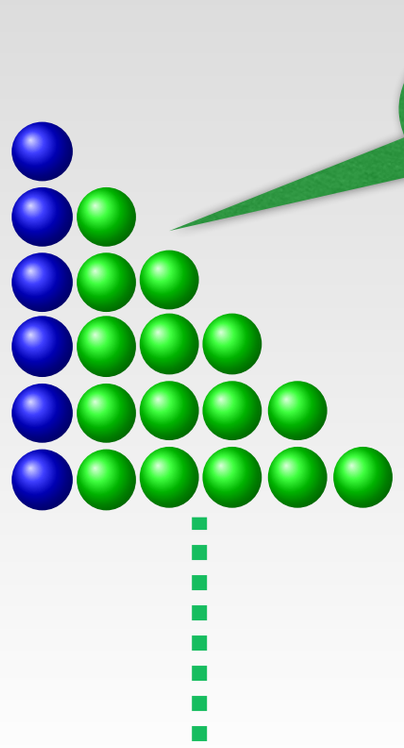
Safety Properties



Safety Property

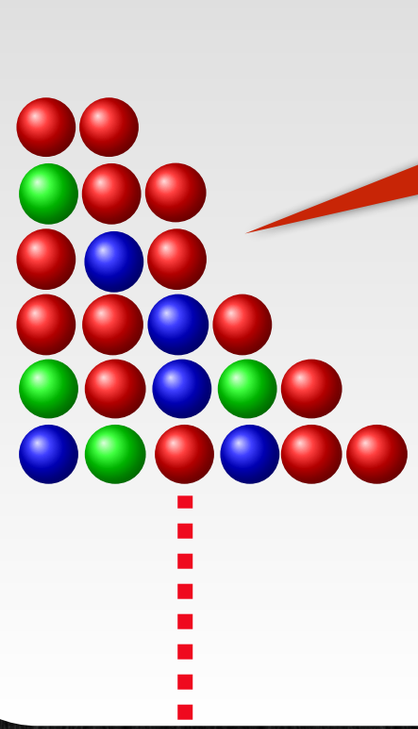
$$\mathit{Init} \xrightarrow{\ *\ } \mathit{Bad}\ ?$$

Initial Markings (*Init*)



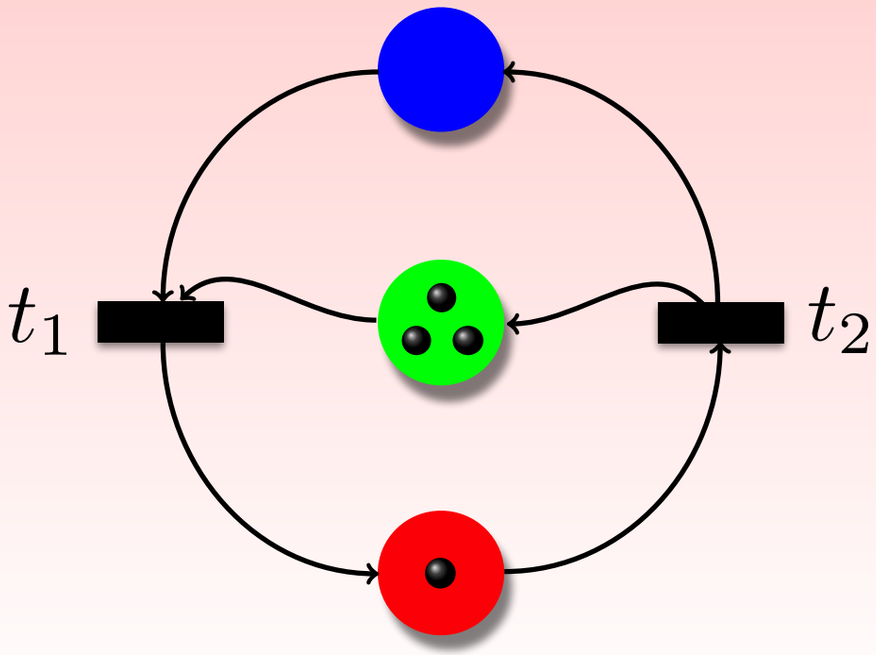
- one
- arbitrarily many

Bad Markings (*Bad*)



- at least two

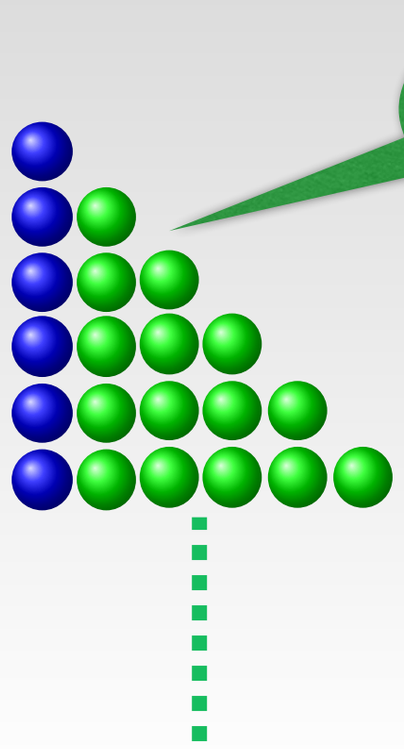
Safety Properties



Safety Property

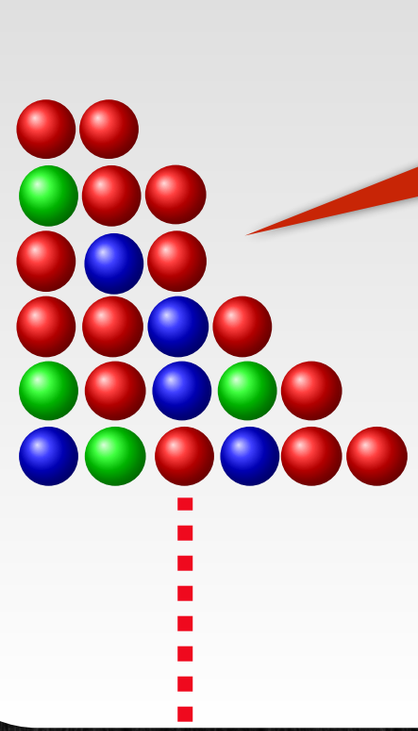
$$Init \xrightarrow{\ *\ } Bad\ ?$$

Initial Markings (*Init*)



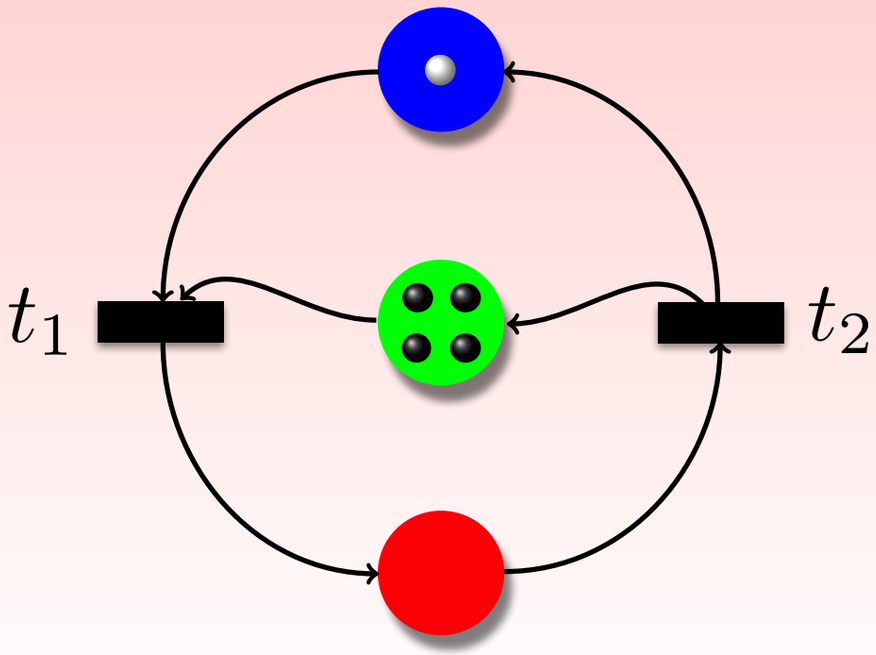
- one
- arbitrarily many

Bad Markings (*Bad*)



- at least two

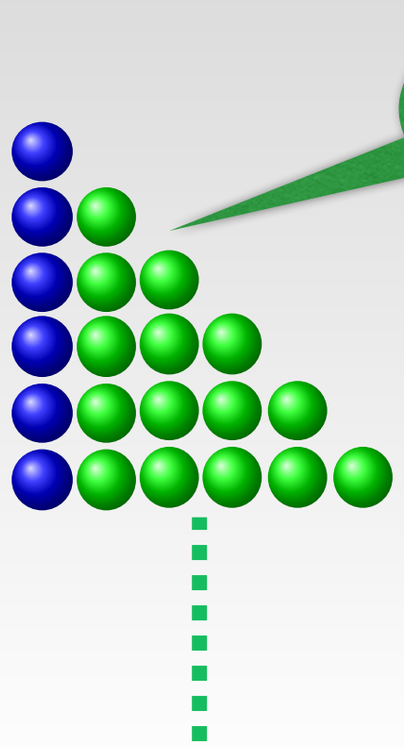
Safety Properties



Safety Property

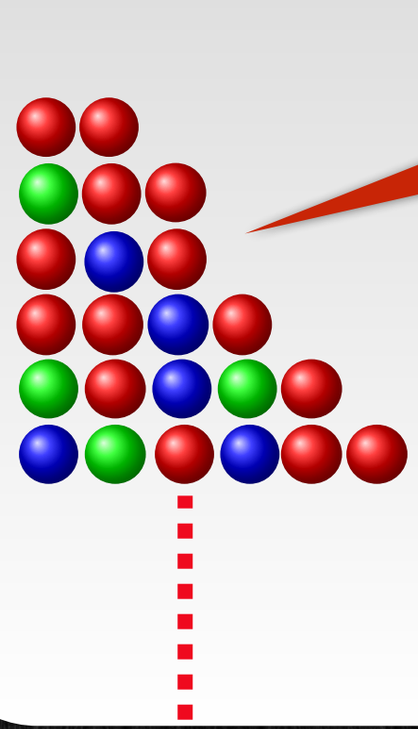
$$\mathit{Init} \xrightarrow{\ *\ } \mathit{Bad}\ ?$$

Initial Markings (*Init*)



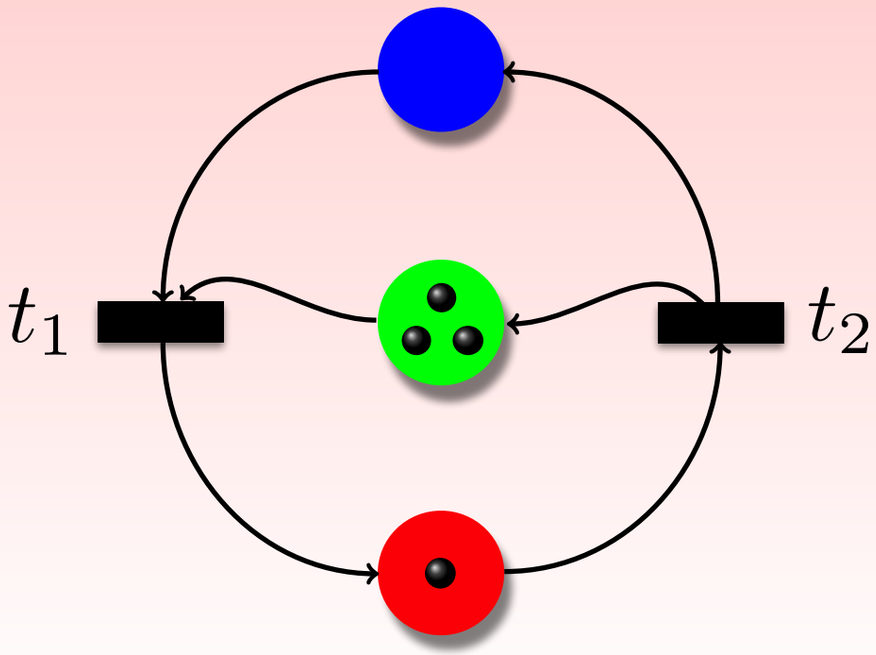
- one
- arbitrarily many

Bad Markings (*Bad*)



- at least two

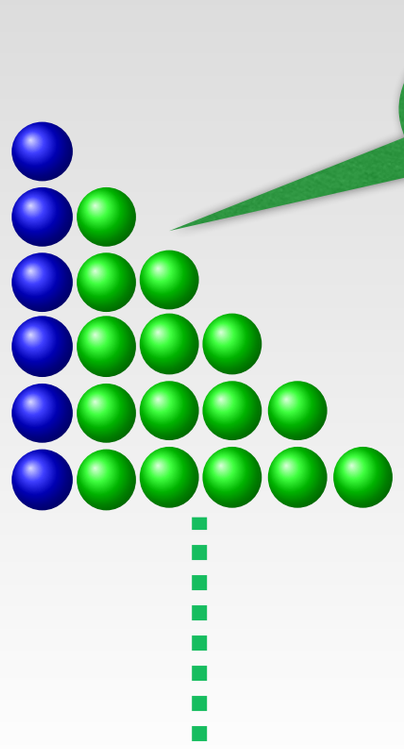
Safety Properties



Safety Property

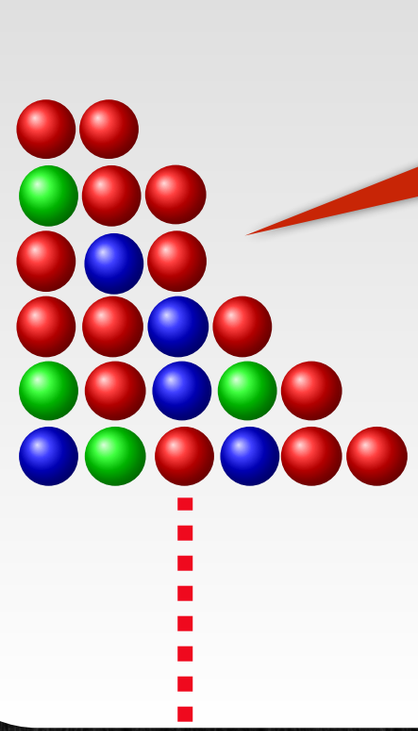
$$Init \xrightarrow{\ *\ } Bad\ ?$$

Initial Markings (*Init*)



- one
- arbitrarily many

Bad Markings (*Bad*)

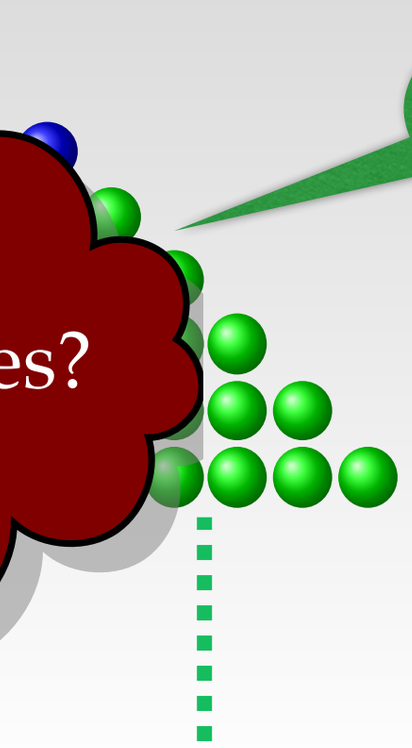


- at least two

Safety Properties

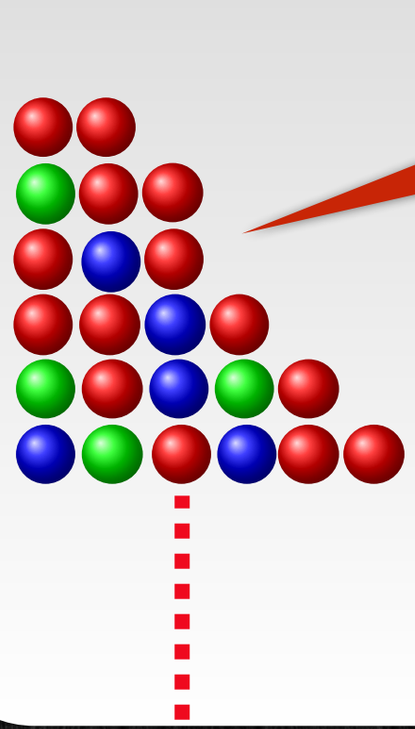
How to check safety properties?

Initial Markings (*Init*)



- one
- arbitrarily many

Bad Markings (*Bad*)



- at least two

Safety Property

$$Init \xrightarrow{*} Bad\,?$$

Safety Properties

How to check safety properties?

Initial Markings (*Init*)

infinitely many

- one
- arbitrarily many

Bad

- Ordering
- Monotonicity
- Upward Closed sets
- Predecessors
- Backward Reachability

Safety Property

$$Init \xrightarrow{*} Bad\,?$$

Petri Nets

Model

Configurations

Transitions

Ordering

Monotoncity

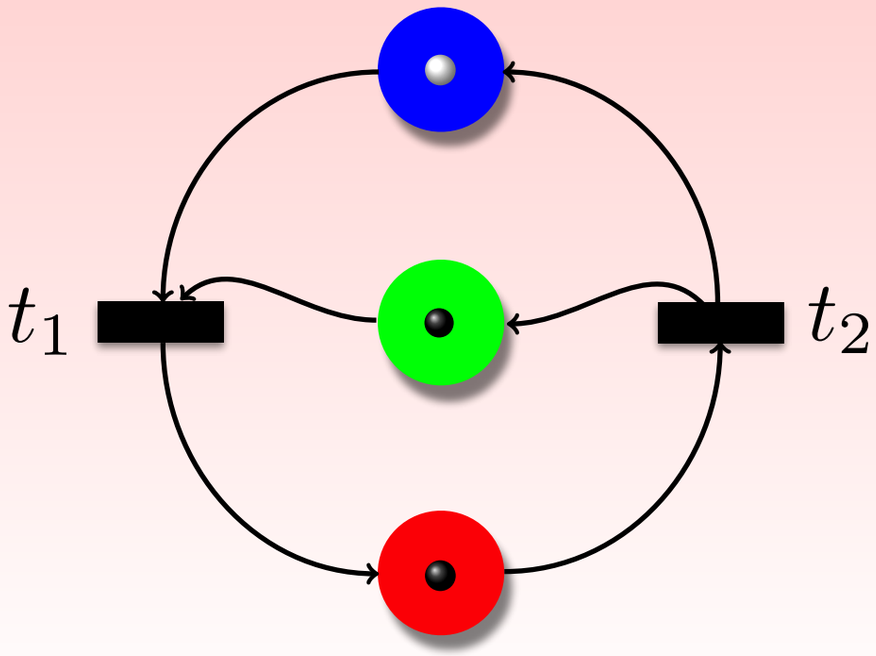
Upward Closed Sets

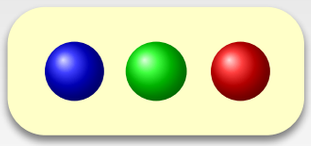
Computing Predecessors

Backward Reachability

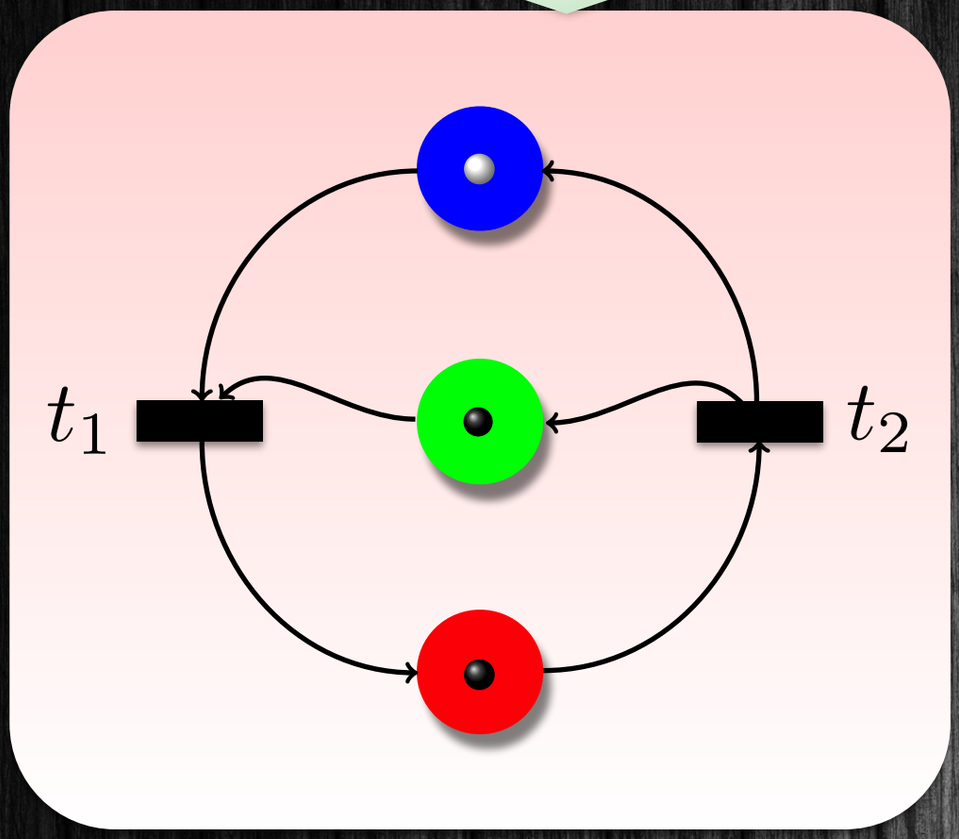
Ordering

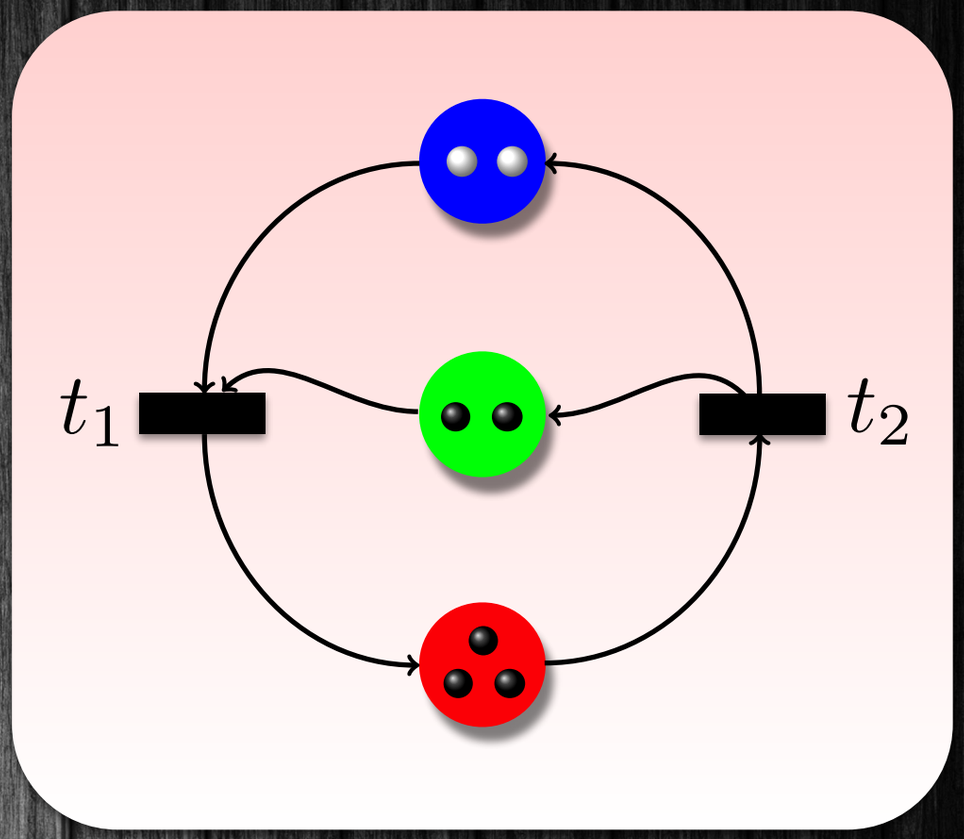
Petri Ordering

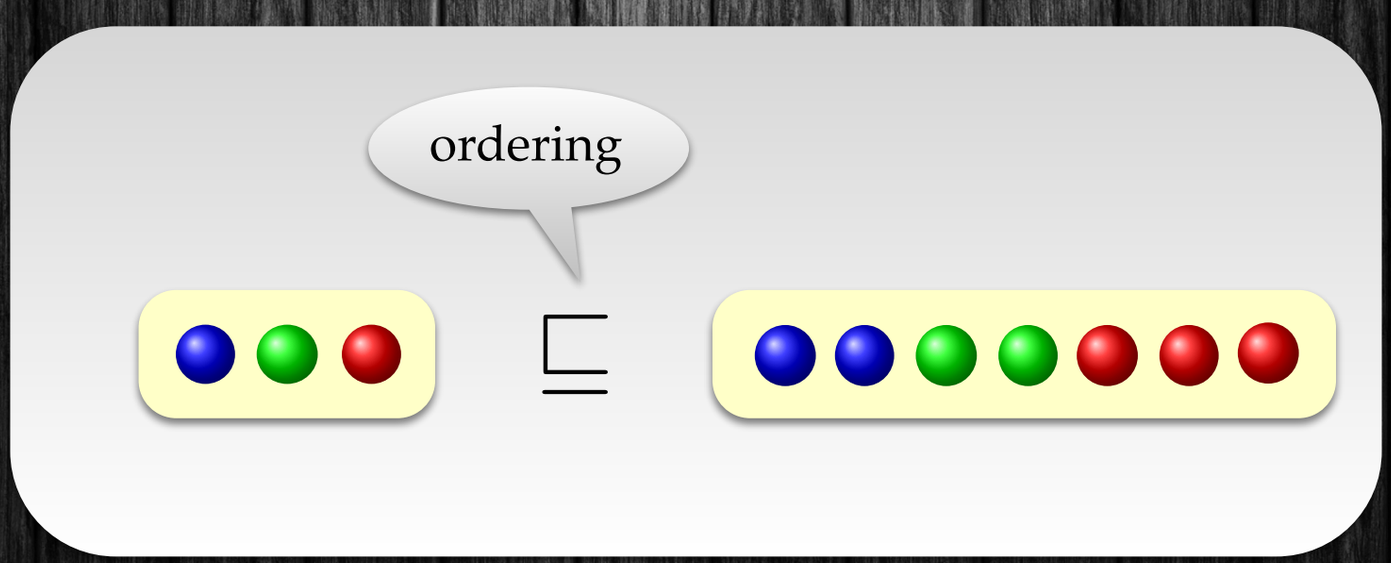




Ordering







Petri Nets

Model

Configurations

Transitions

Ordering

Monotoncity

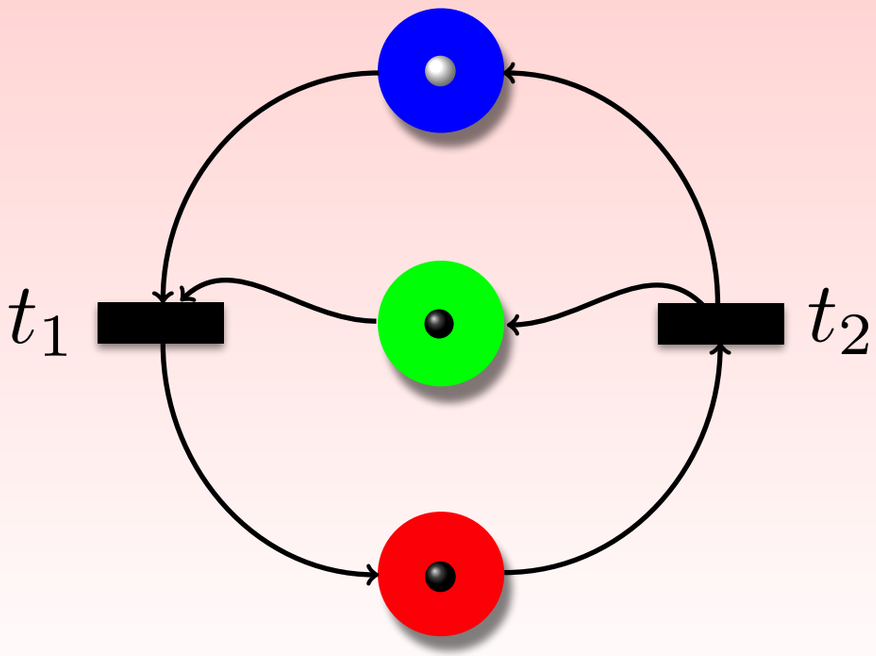
Upward Closed Sets

Computing Predecessors

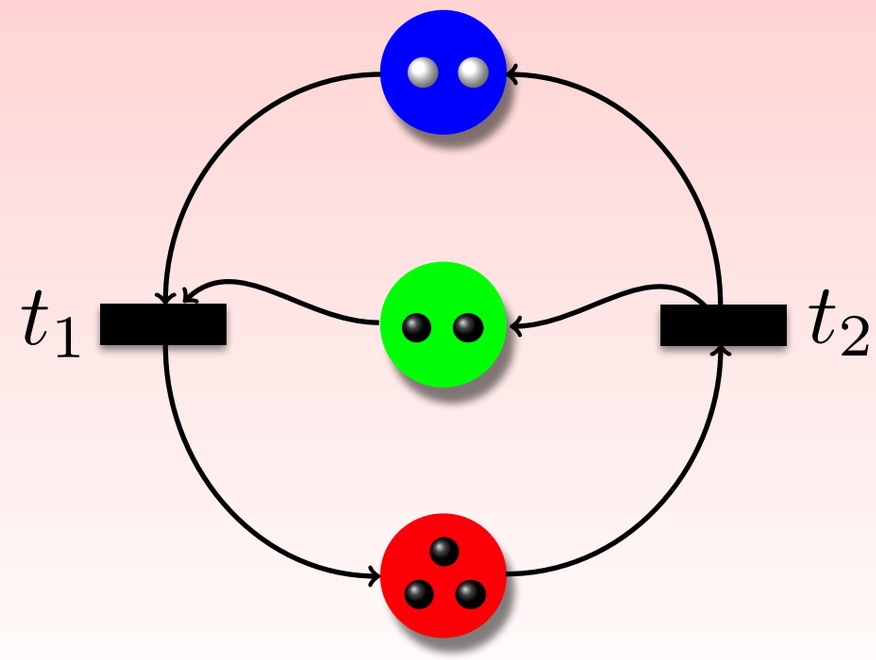
Backward Reachability

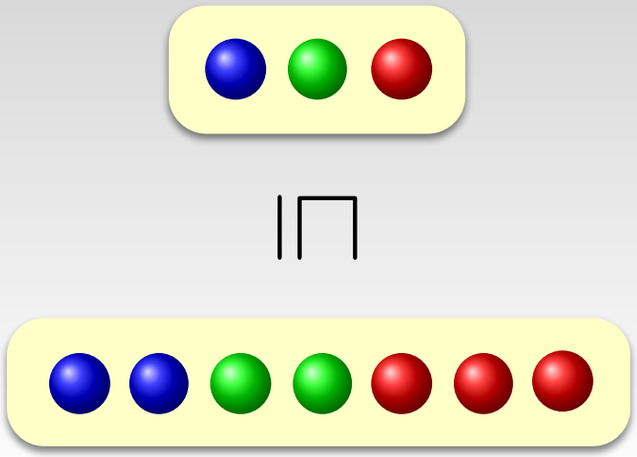
Pet

Monotonicity



t_1



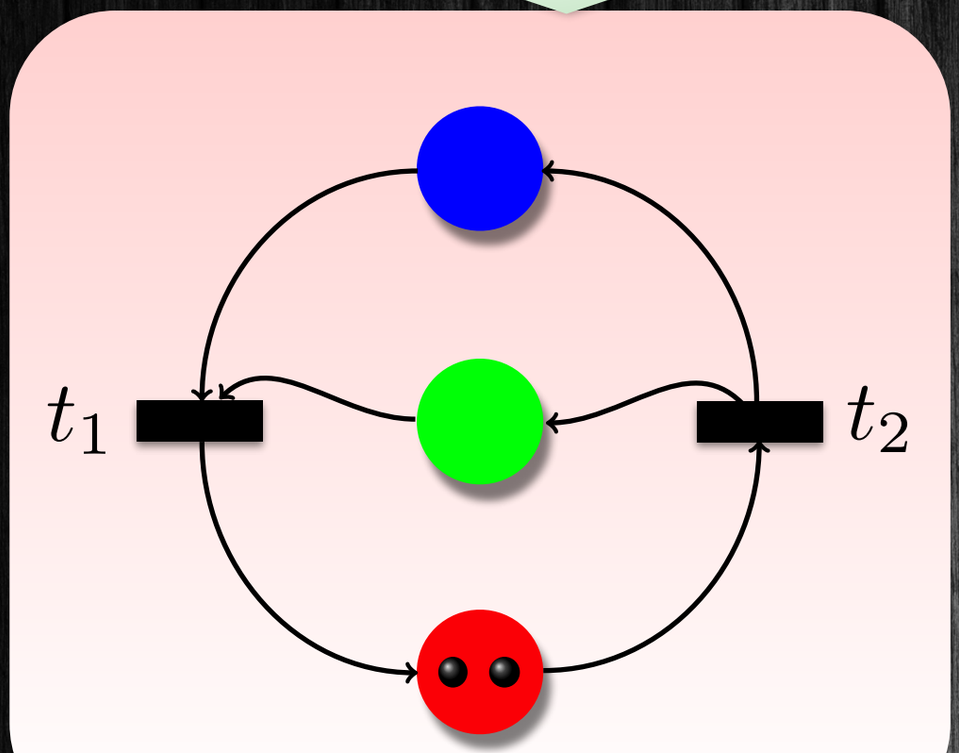


$\sqsubseteq$

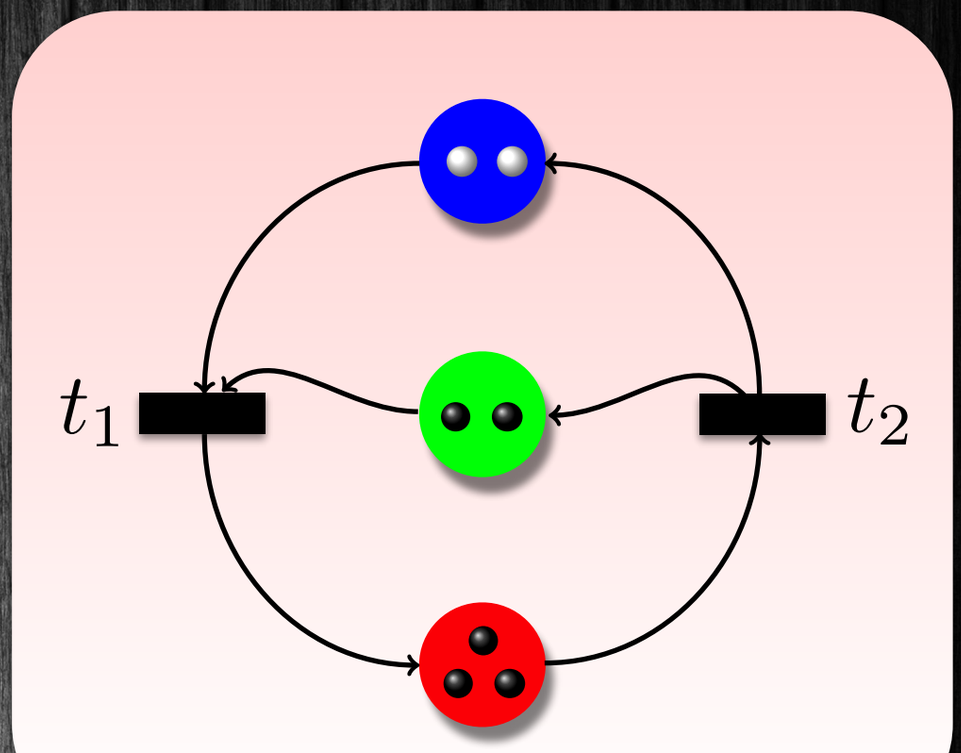
Monotonicity

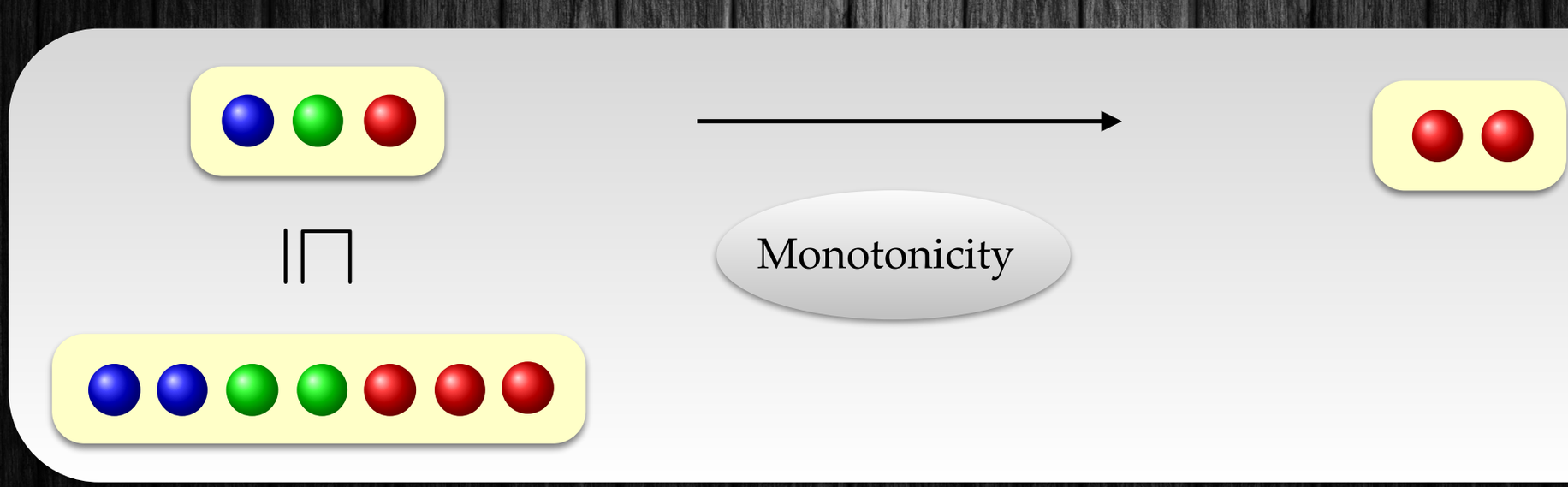
Pet

Monotonicity



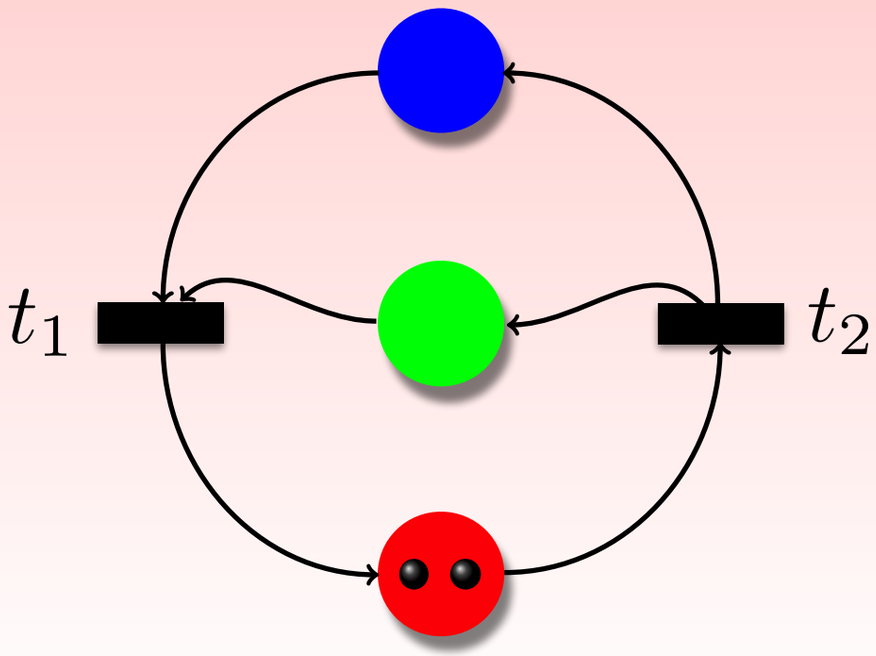
t_1



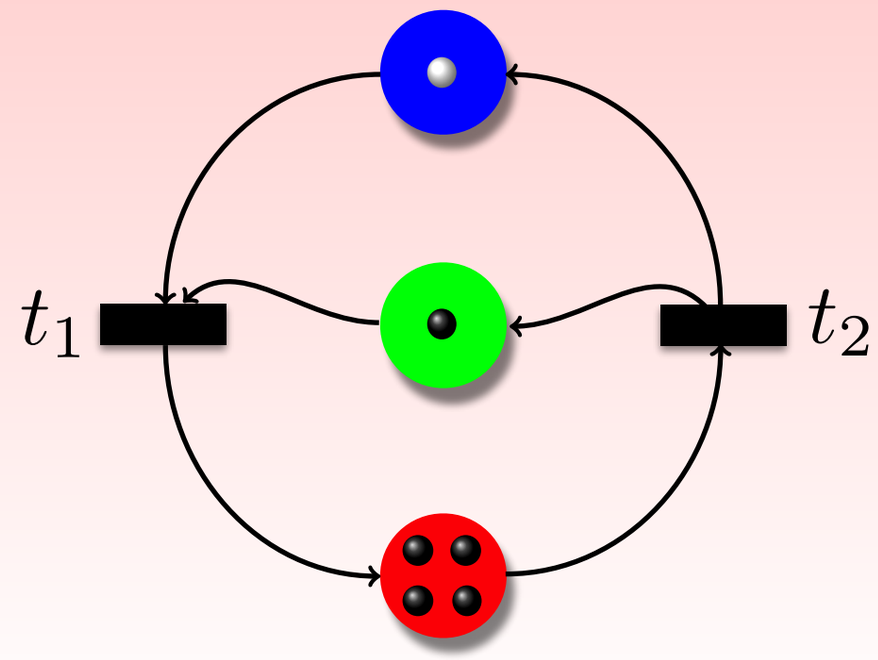


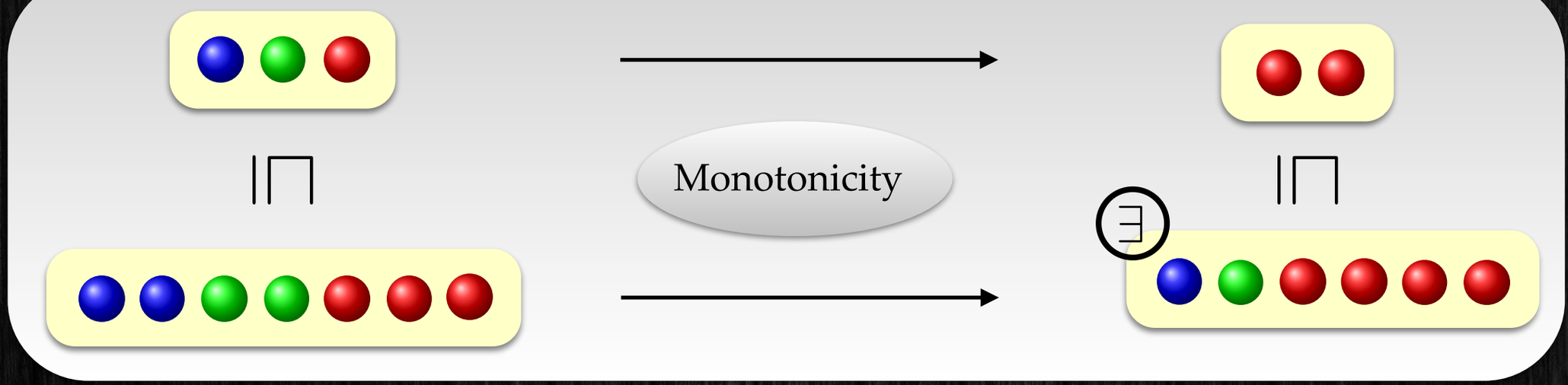
Pet

Monotonicity



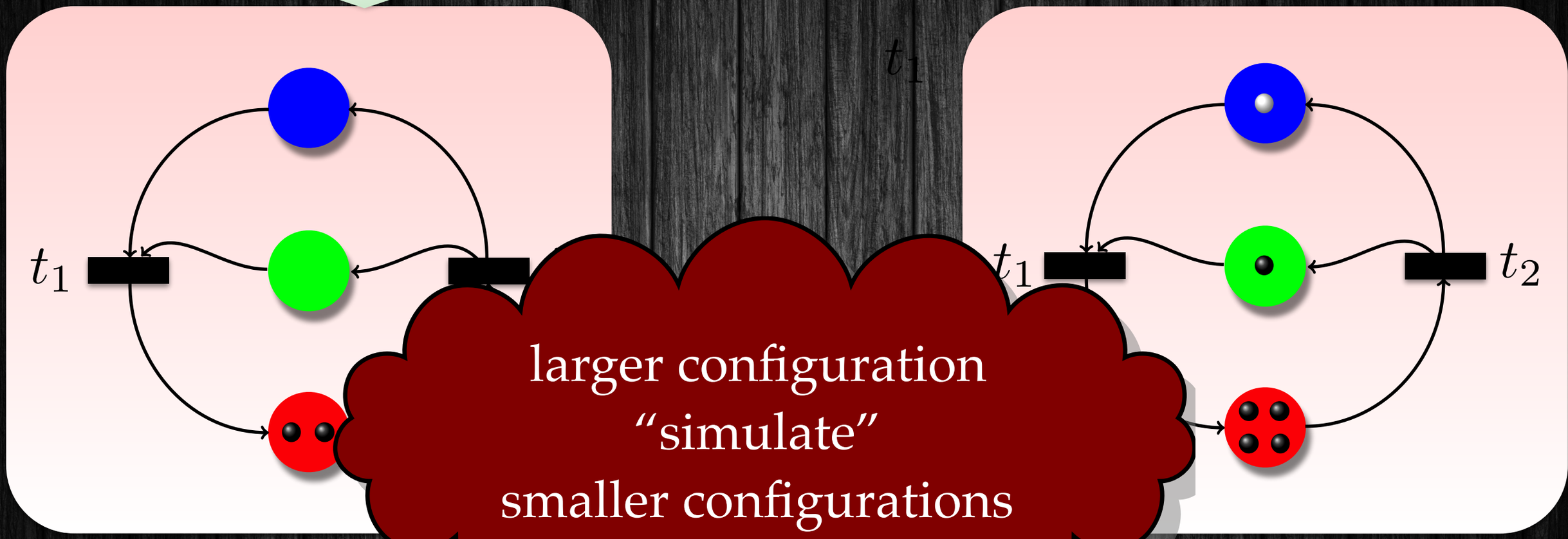
t_1



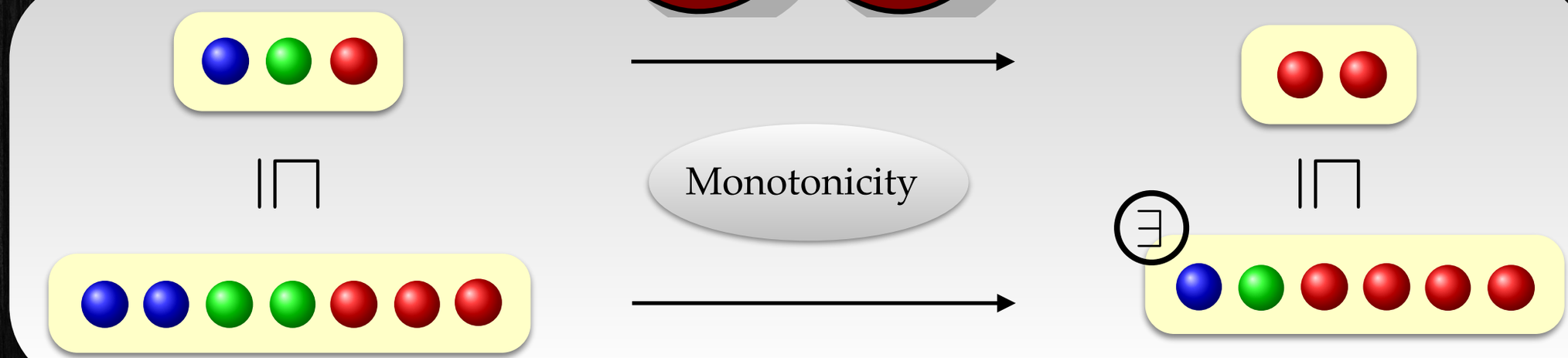


Pet

Monotonicity



larger configuration
"simulate"
smaller configurations



Petri Nets

Model

Configurations

Transitions

Ordering

Monotoncity

Upward Closed Sets

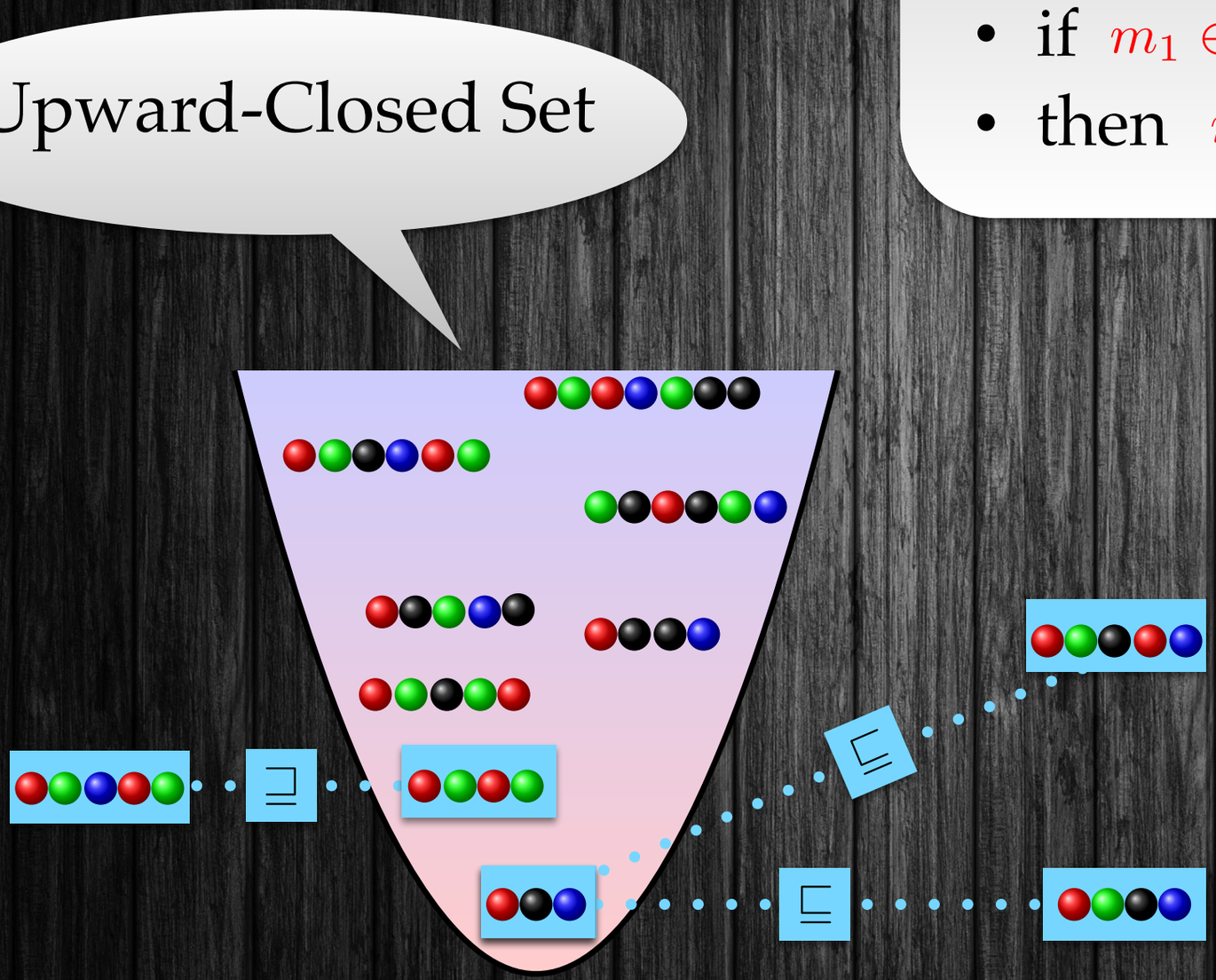
Computing Predecessors

Backward Reachability

Petri

Upward Closed Sets

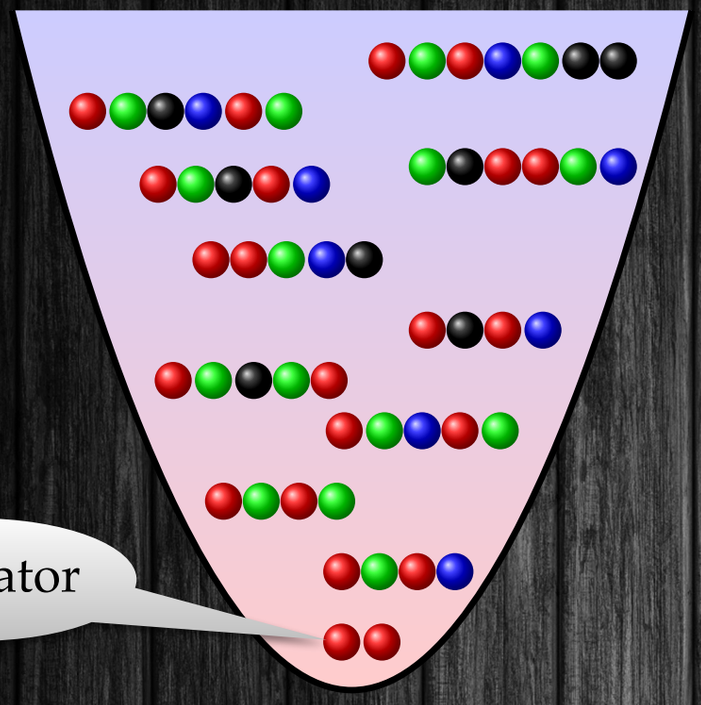
Upward-Closed Set



Upward Closed Set (UC)

- if $m_1 \in U$ and $m_1 \sqsubseteq m_2$
- then $m_2 \in U$

Upward Closed Sets



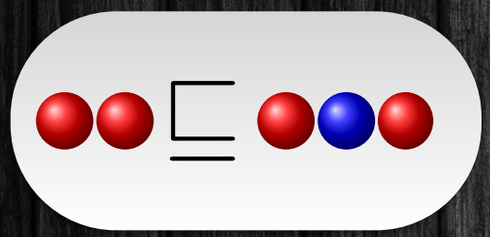
Upward Closed Set (UC)

- if $m_1 \in U$ and $m_1 \sqsubseteq m_2$
- then $m_2 \in U$

Why UC?

- Bad sets of markings are UC
  - checking safety properties = reachability of bad markings
- Uniquely characterized by generator
  - simple representation= finite multiset

Upward Closed Sets



implies

Petri Nets

Model

Configurations

Transitions

Ordering

Monotoncity

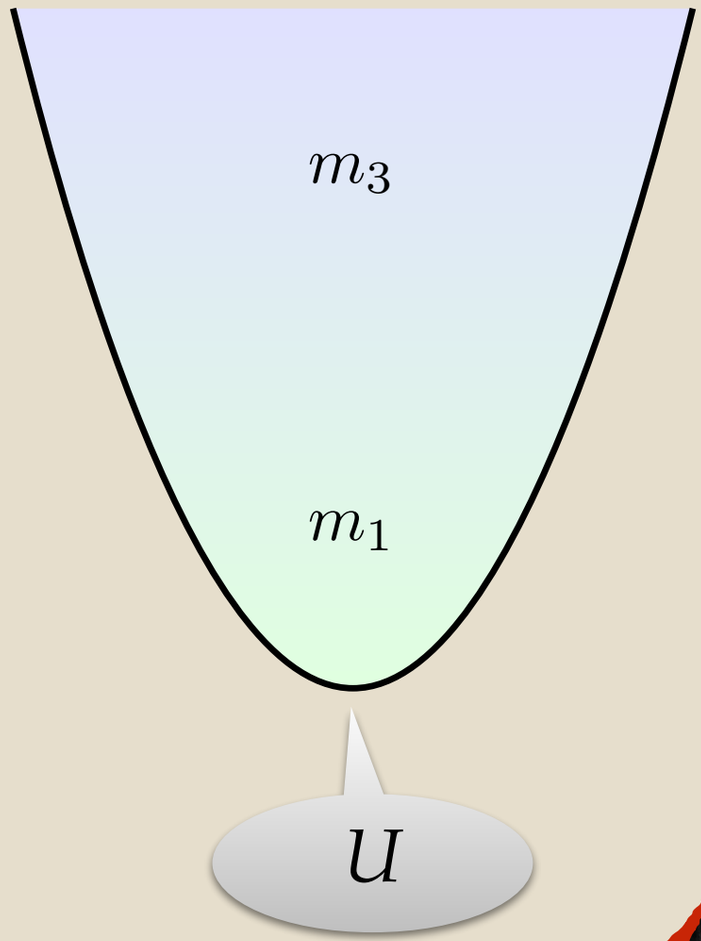
Upward Closed Sets

Computing Predecessors

Backward Reachability

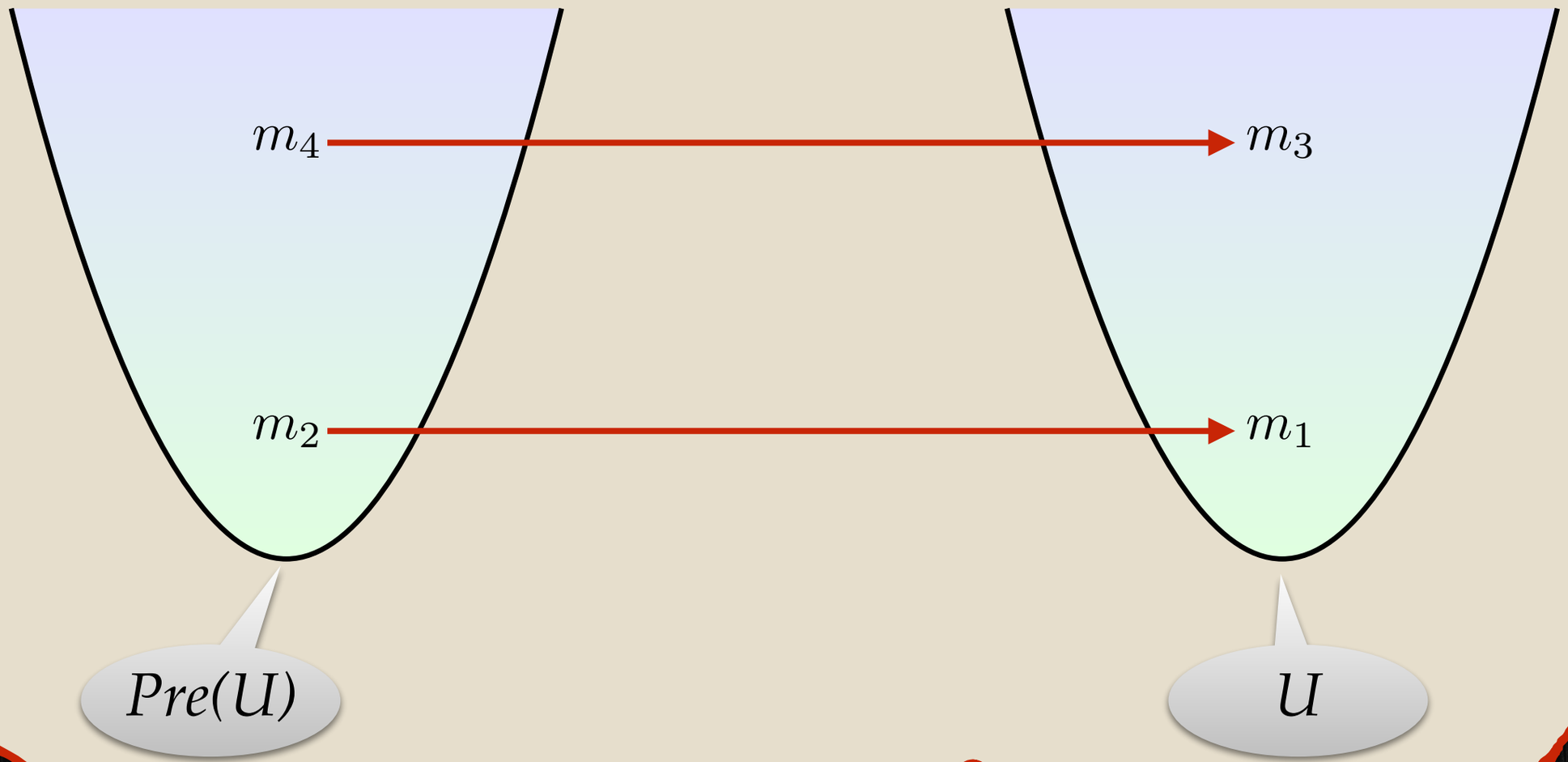
Petri

Predecessors



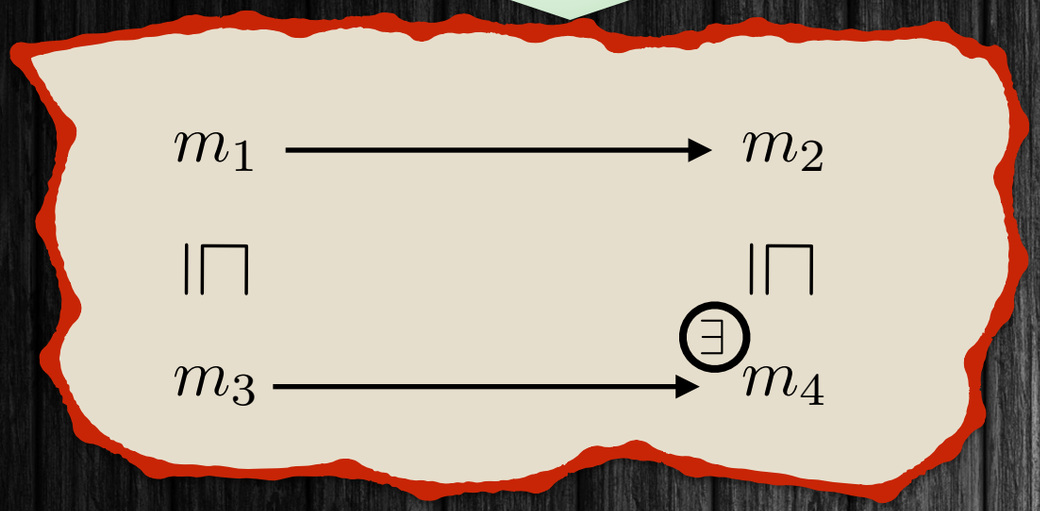
Petri

Predecessors



Petri

Predecessors

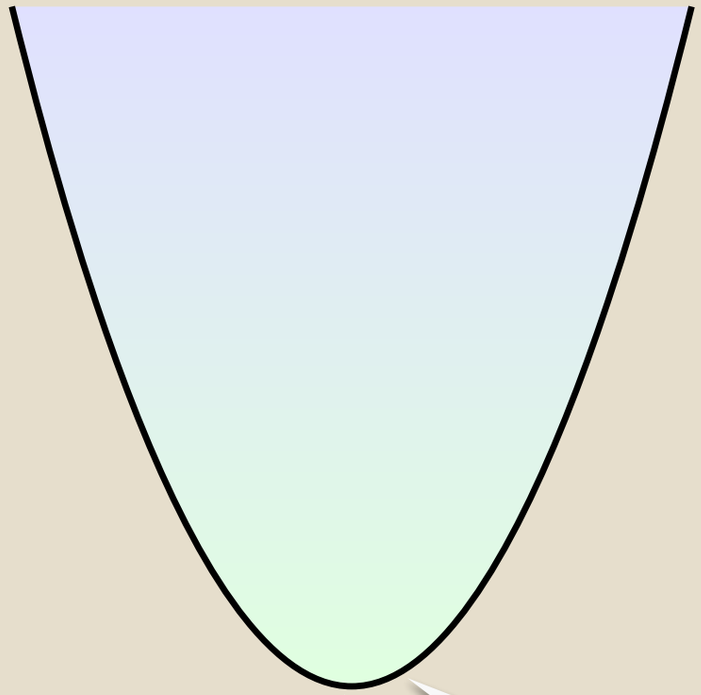


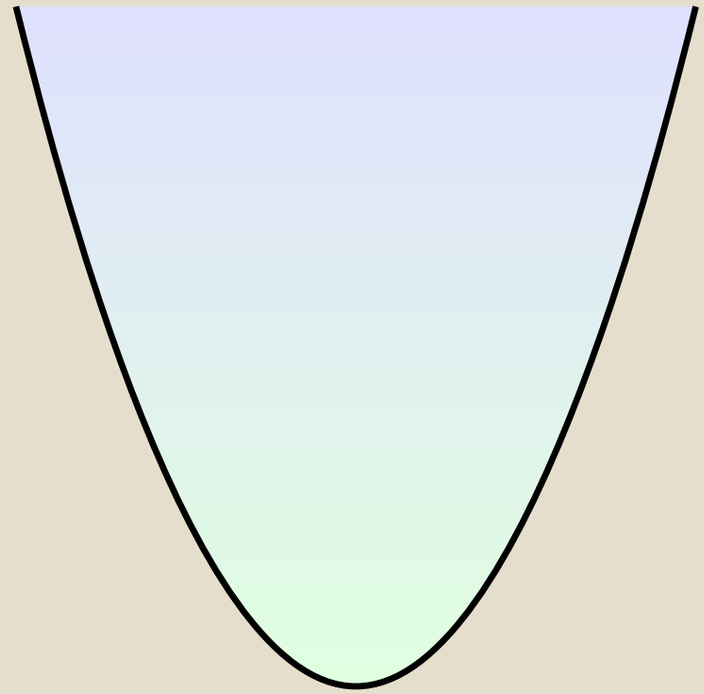
Monotonicity: UC persevered by *Pre*

Petri

Predecessors

Monotonicity: UC persevered by *Pre*





Pre(U)

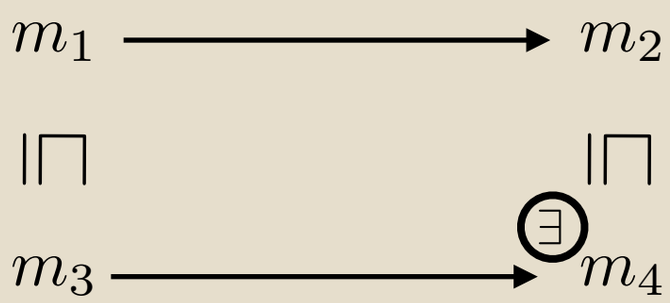
upward closed?

U

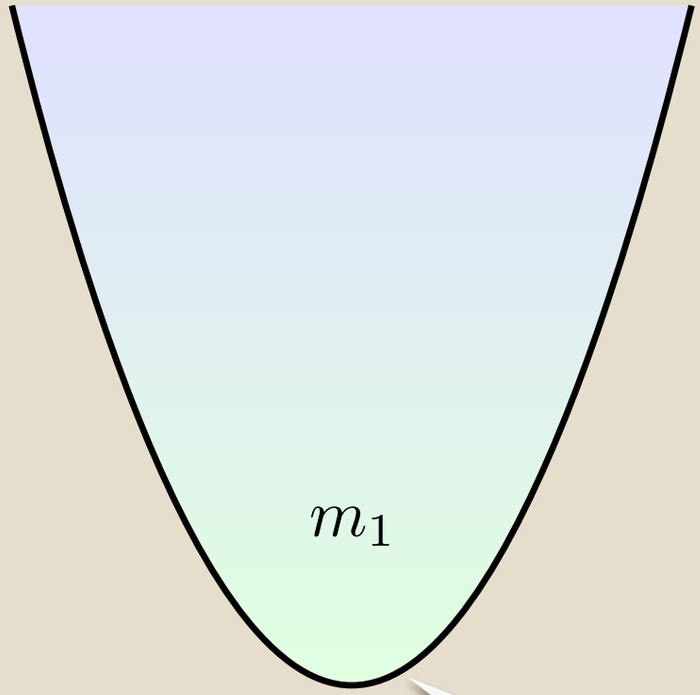
upward closed

Petri

Predecessors

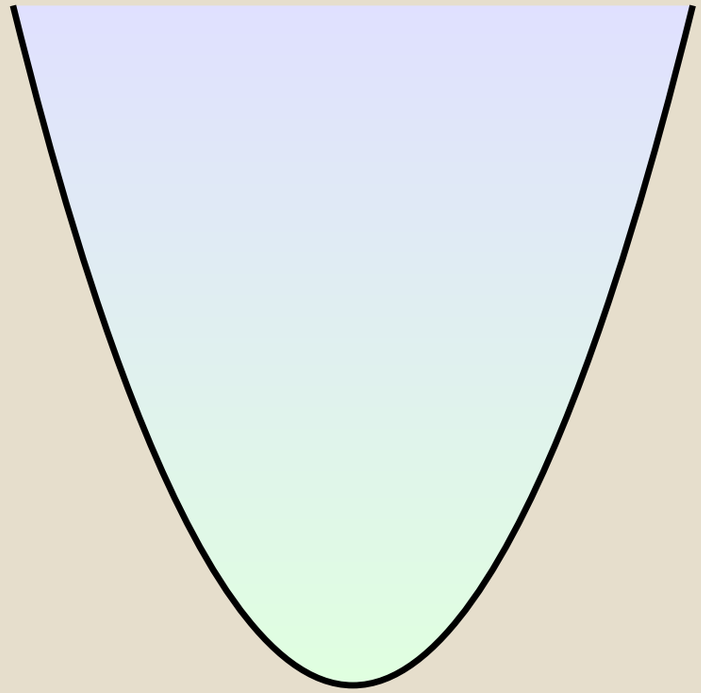


Monotonicity: UC persevered by *Pre*



Pre(U)

upward closed?

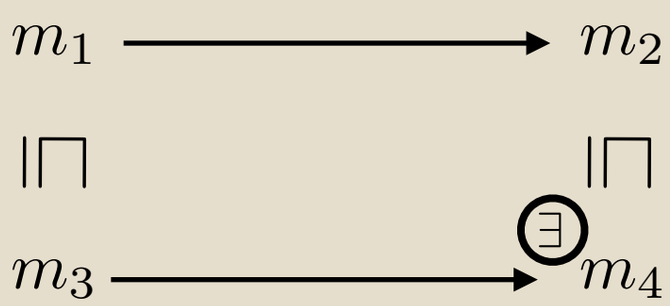


upward closed

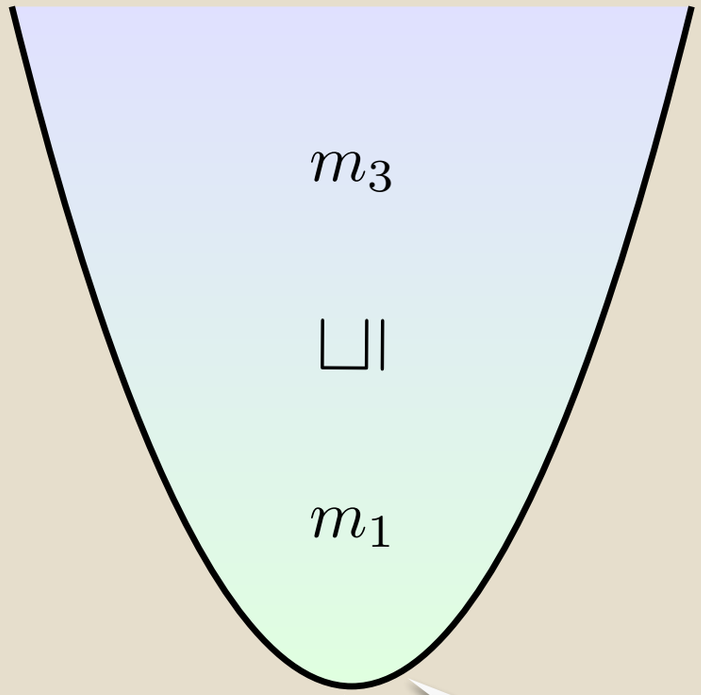
U

Petri

Predecessors

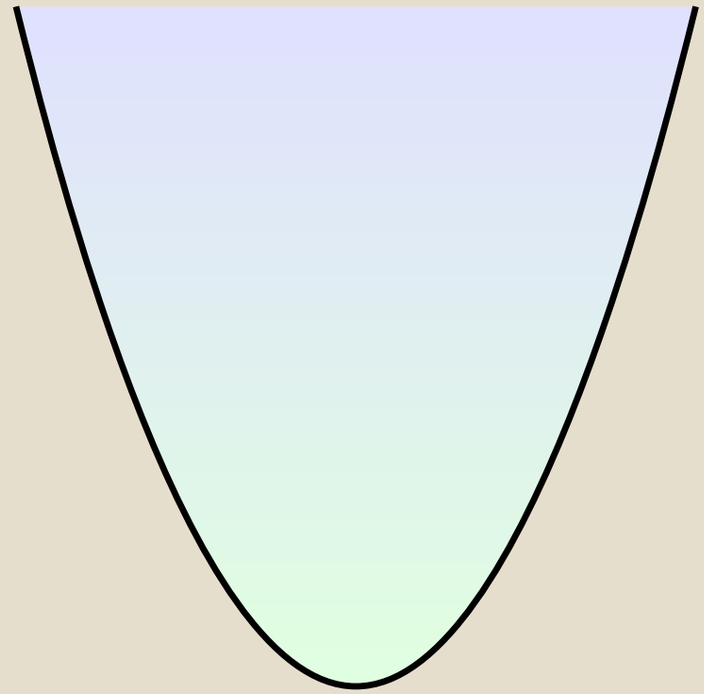


Monotonicity: UC persevered by *Pre*



Pre(U)

upward closed?



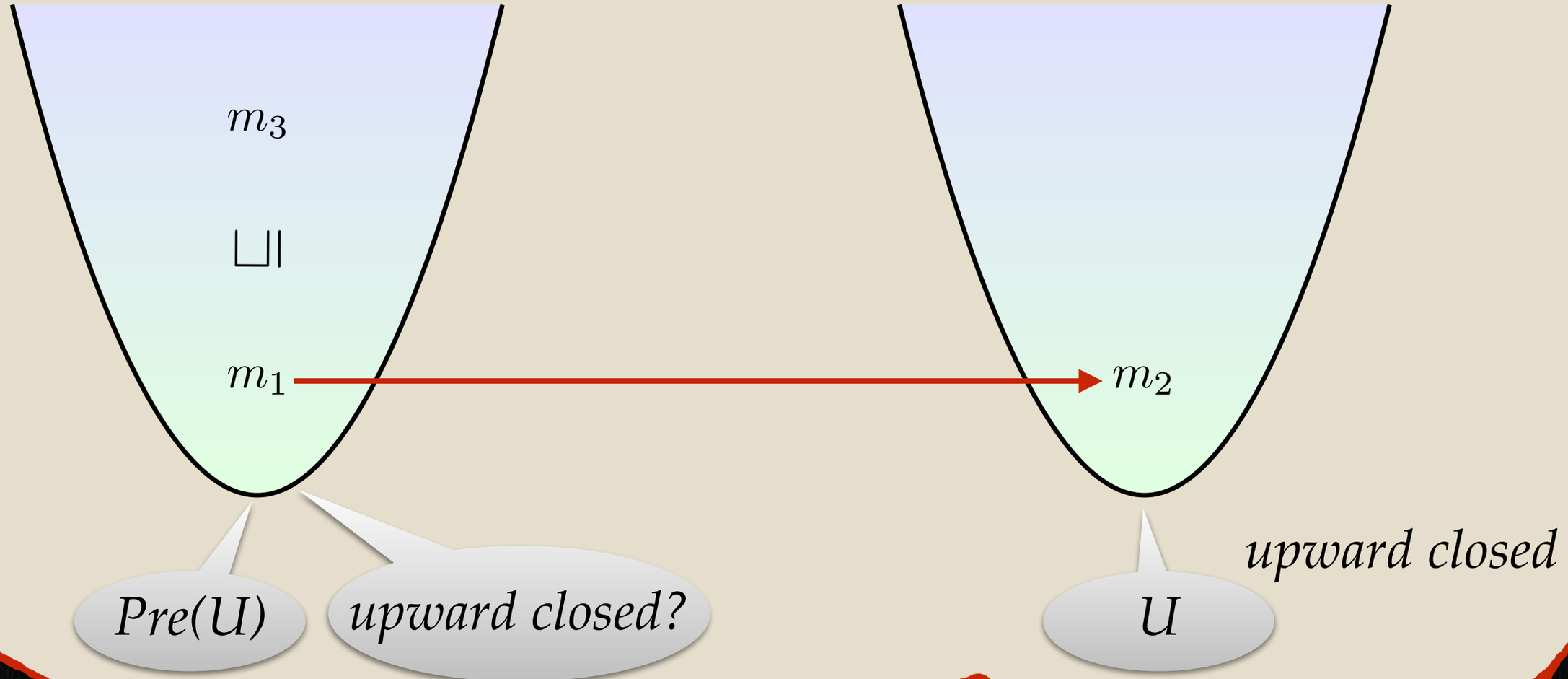
U

upward closed

Petri

Predecessors

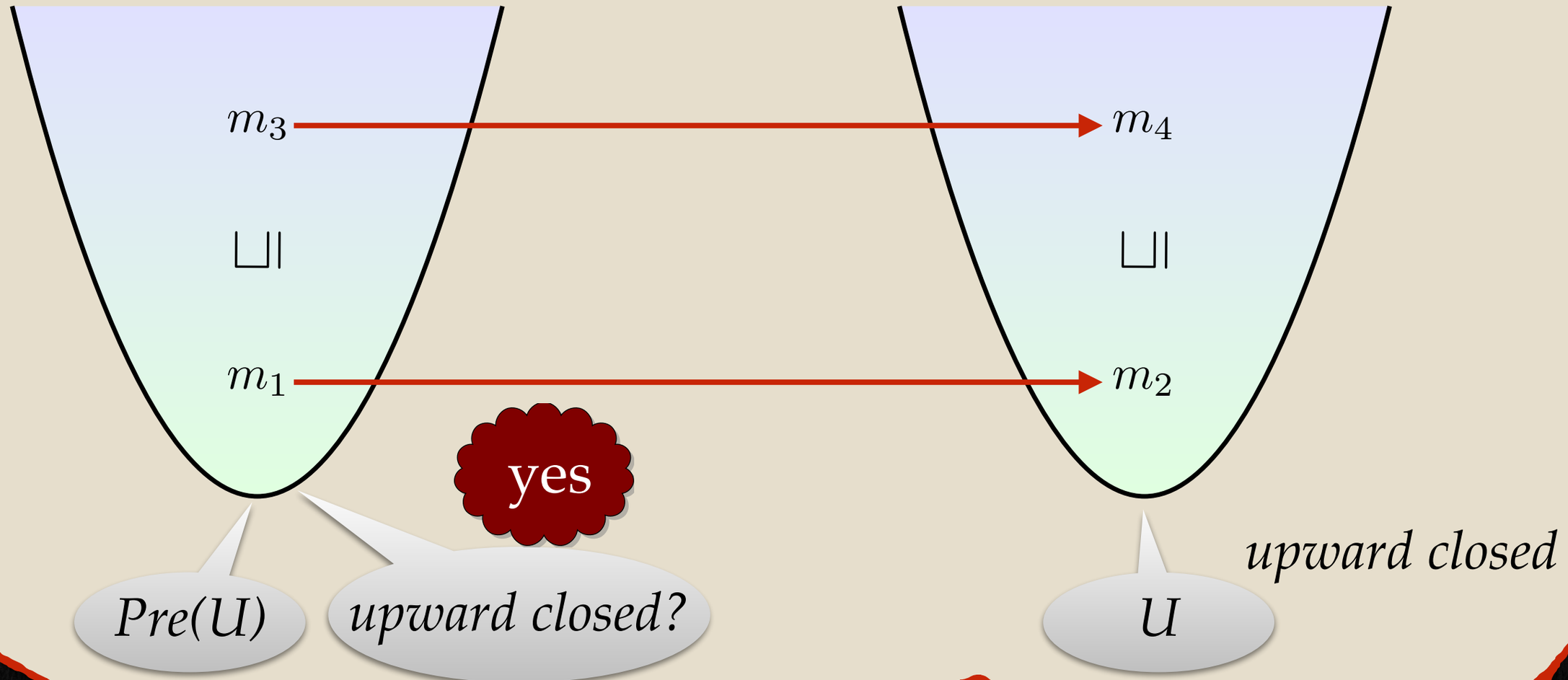
Monotonicity: UC persevered by *Pre*



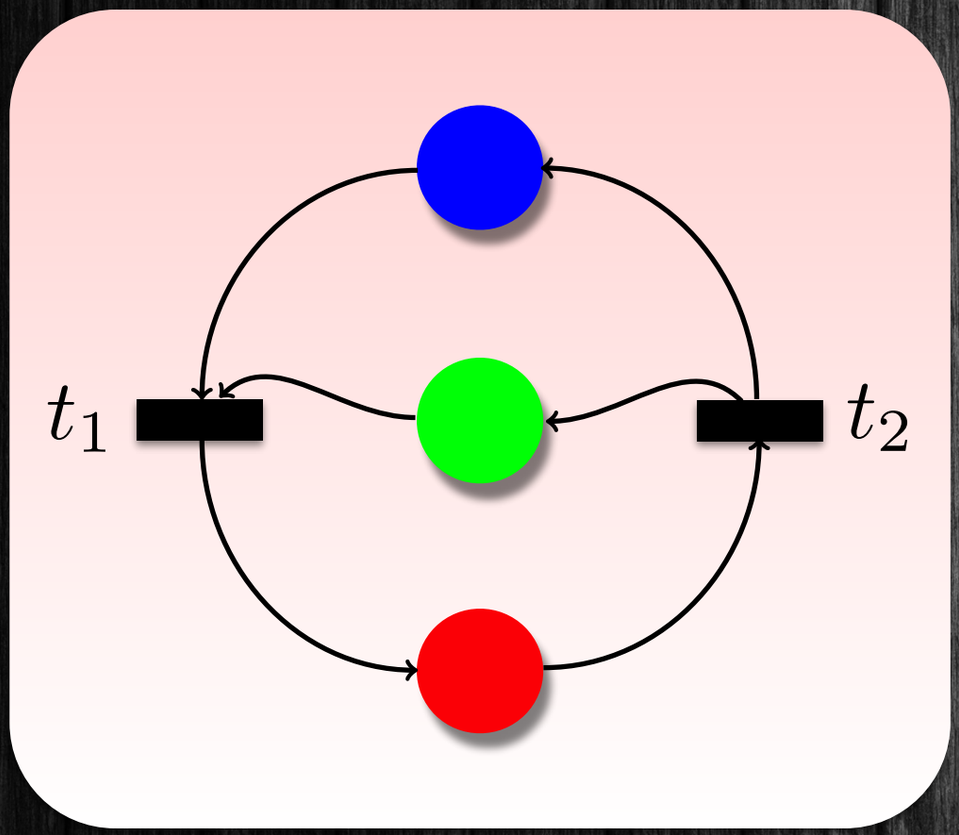
Petri

Predecessors

Monotonicity: UC persevered by *Pre*

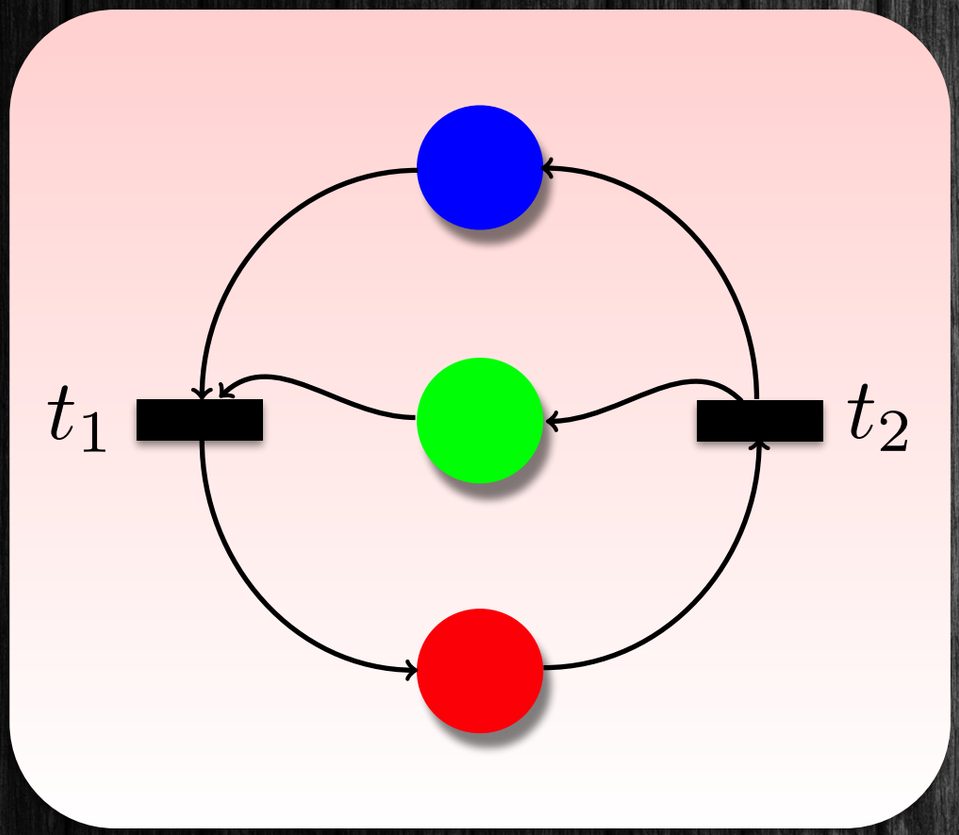


Computing Predecessors



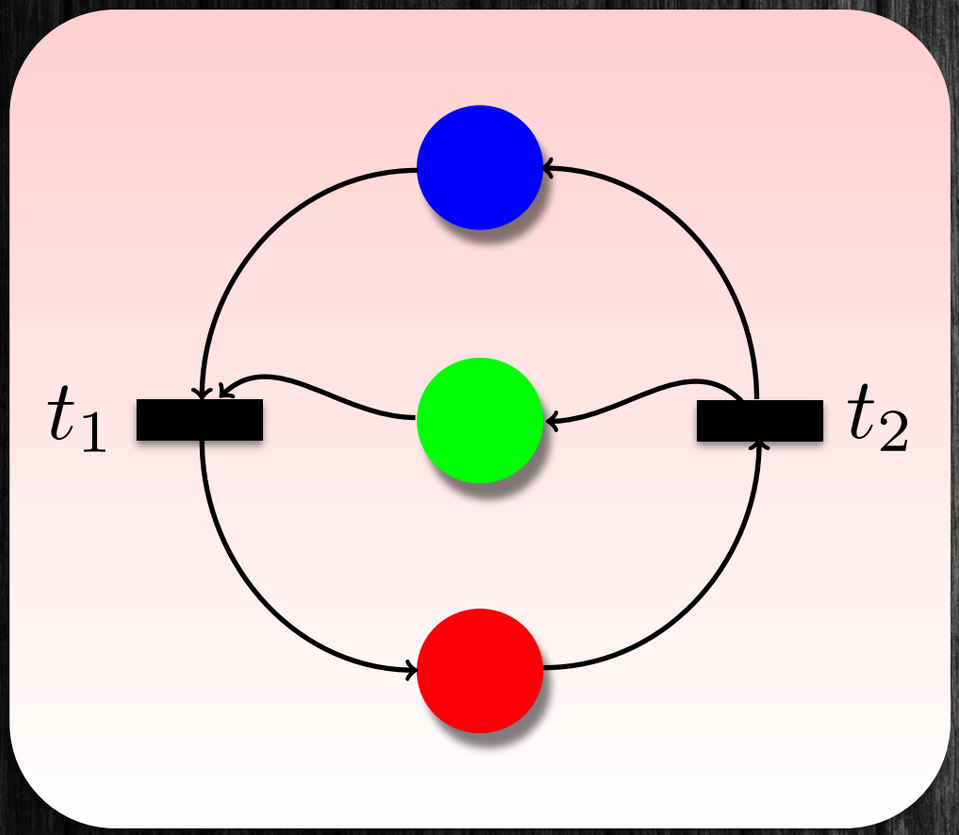
$$Pre_{t_1}\left[\right] =$$

Petri Nets: Computing Predecessors



$$Pre_{t_1}\left[\;\right] = \;$$

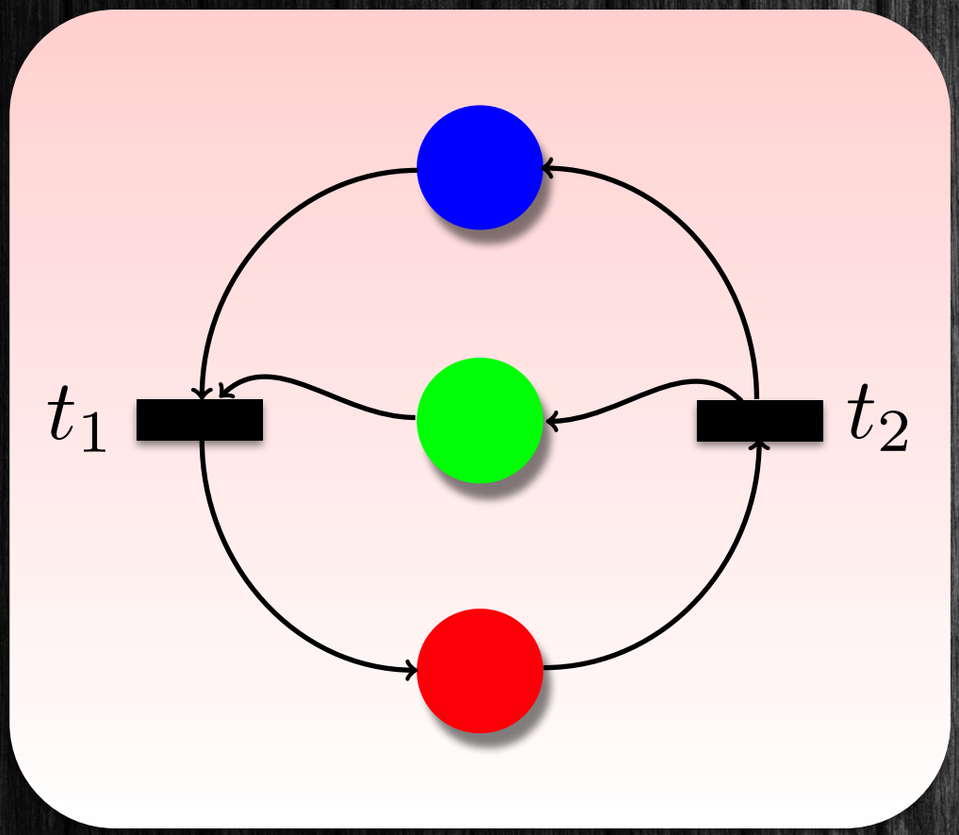
Computing Predecessors



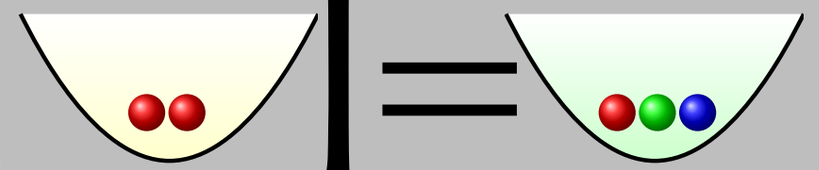
$$Pre_{t_1}\left[\;\right] = \;$$

$$Pre_{t_2}\left[\;\right] =$$

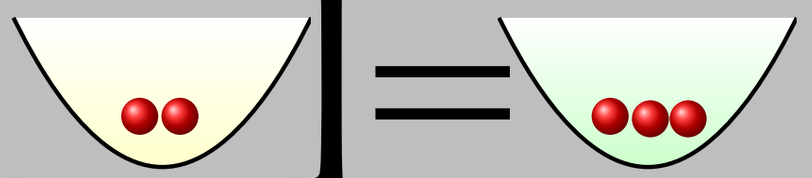
Petri Nets: Computing Predecessors



$$Pre_{t_1}\left[\;\right] = \;$$



$$Pre_{t_2}\left[\;\right] = \;$$



Petri Nets

Model

Configurations

Transitions

Ordering

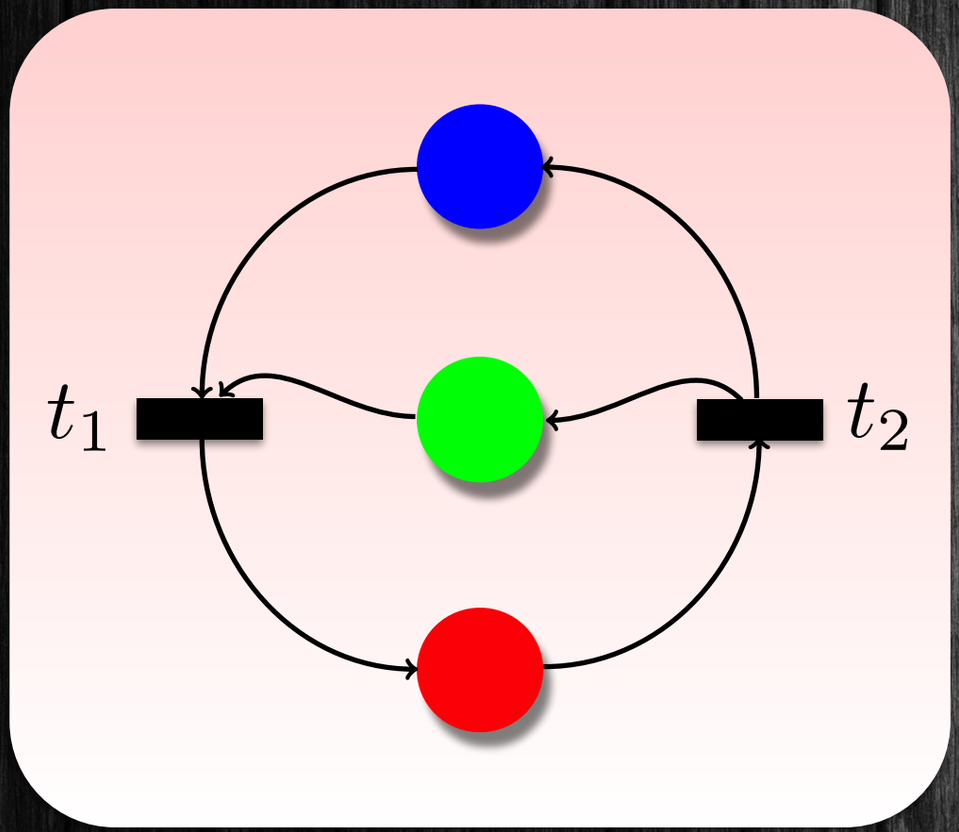
Monotoncity

Upward Closed Sets

Computing Predecessors

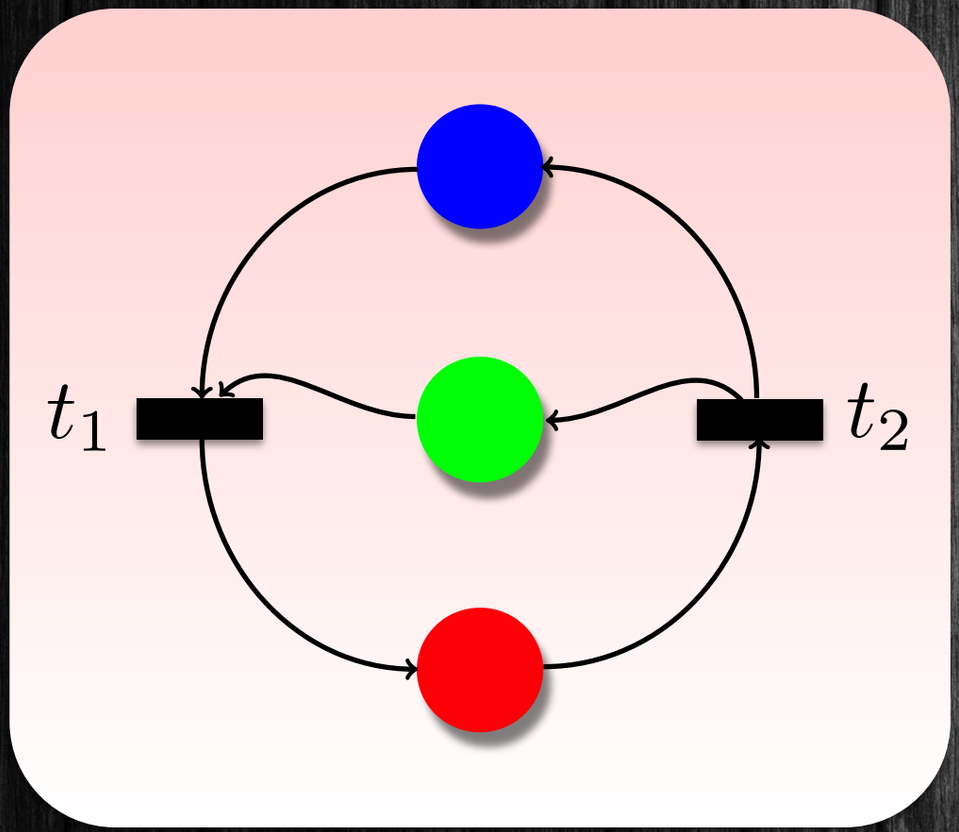
Backward Reachability

Backward Reachability

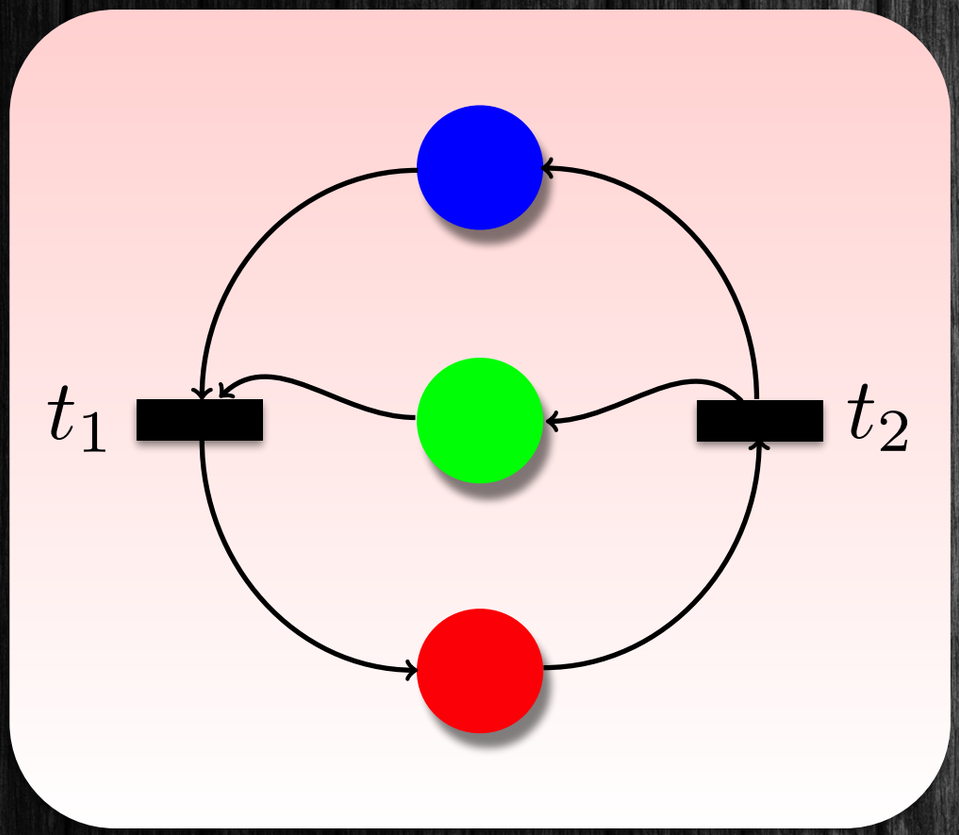


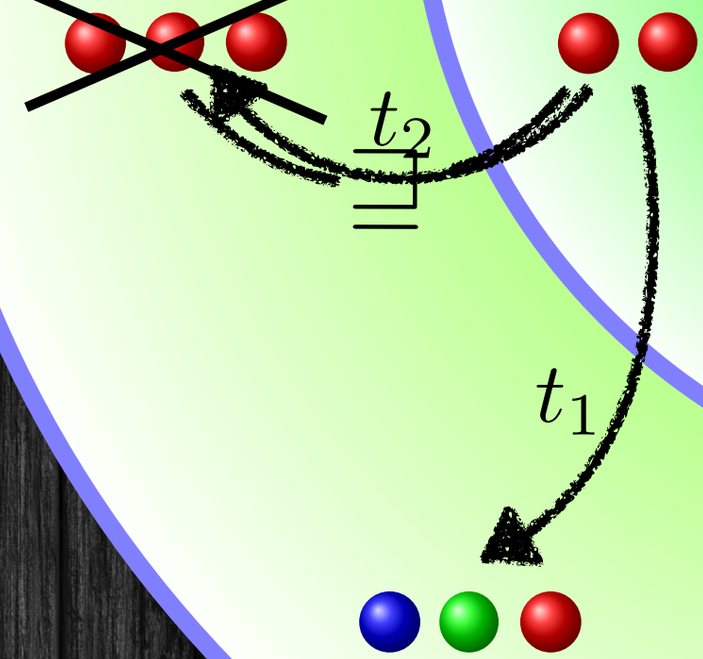
Petri

Backward Reachability



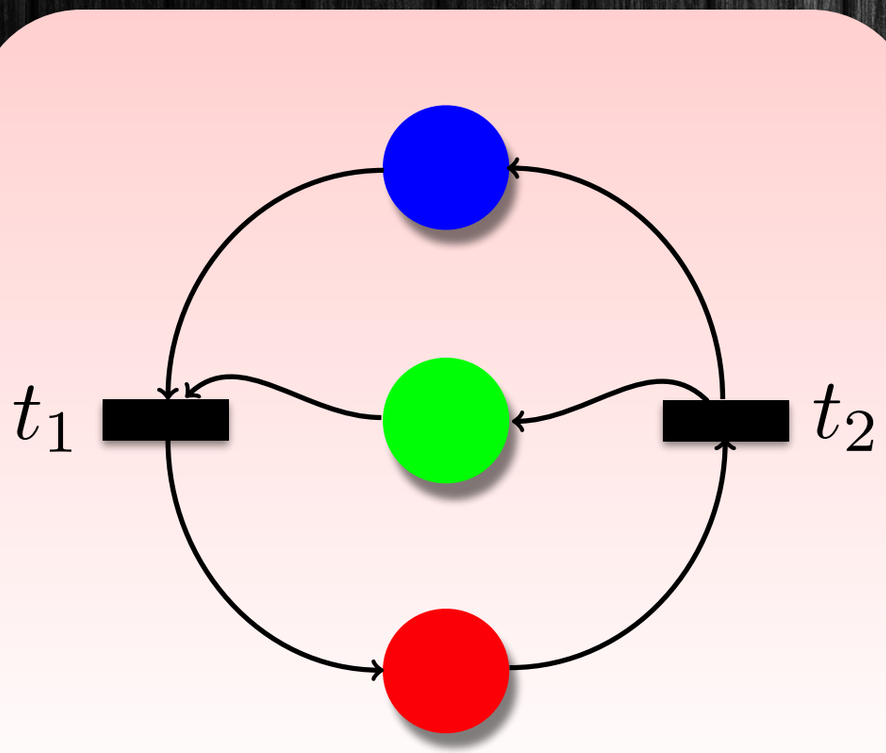
Backward Reachability

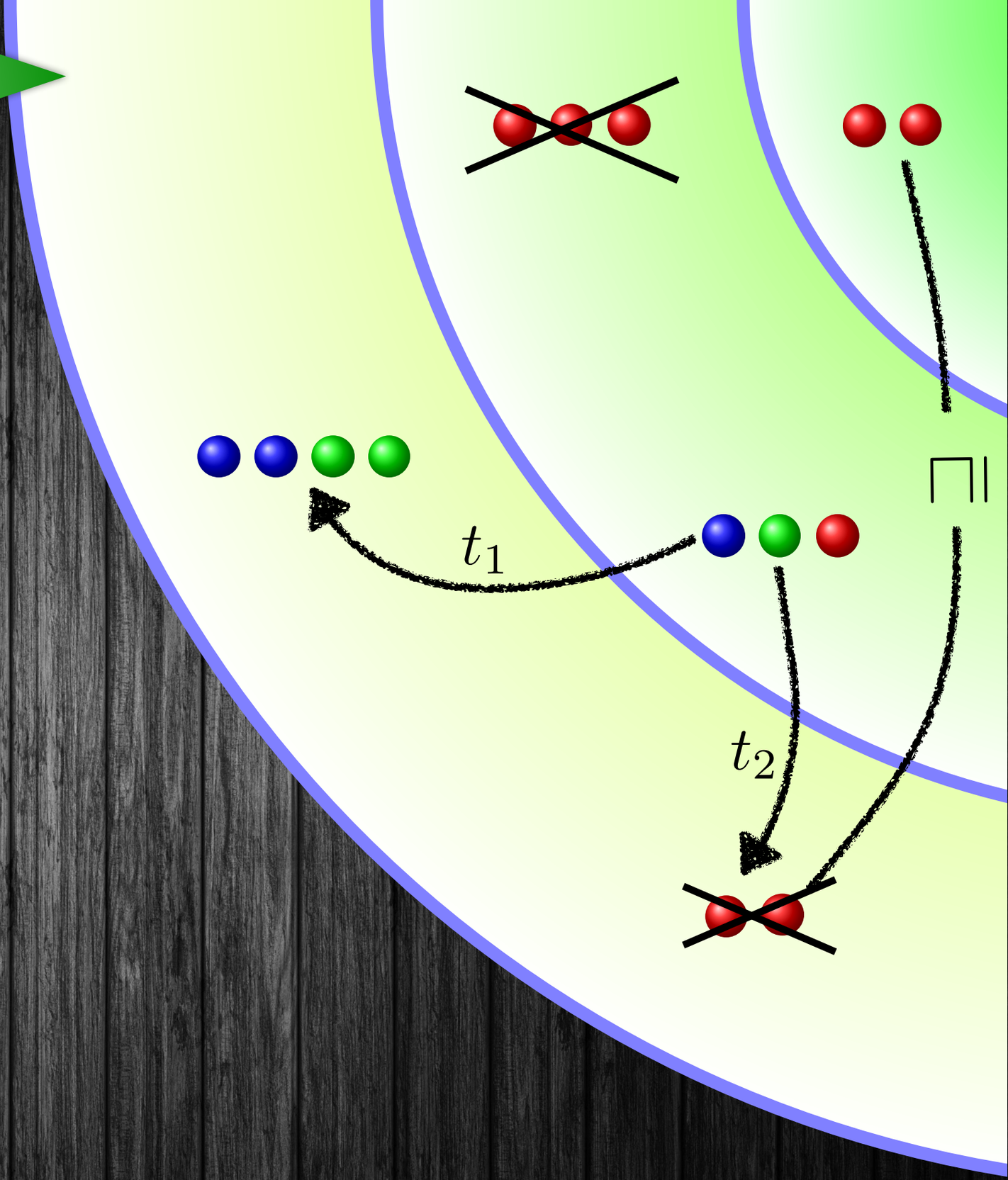




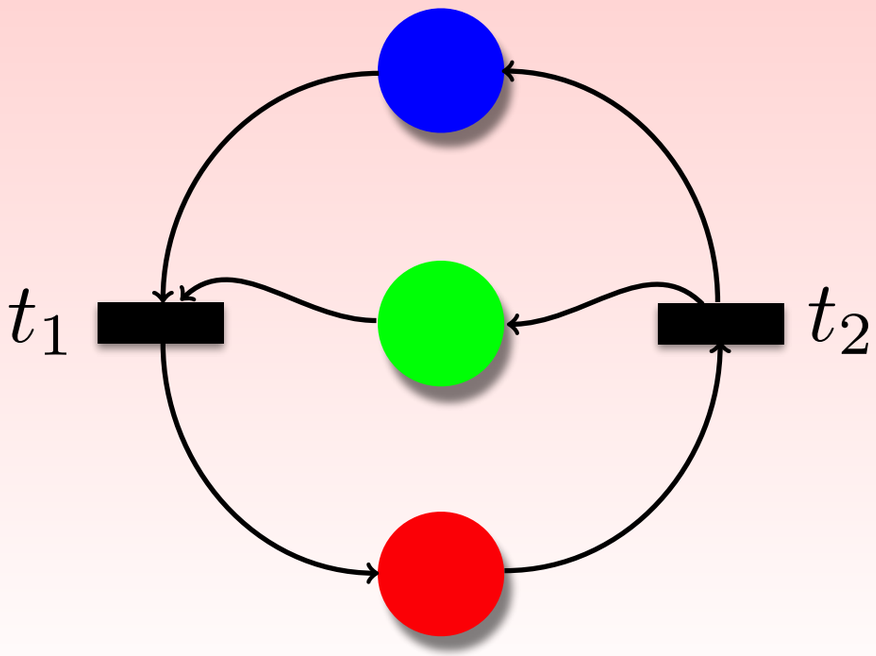
Petri

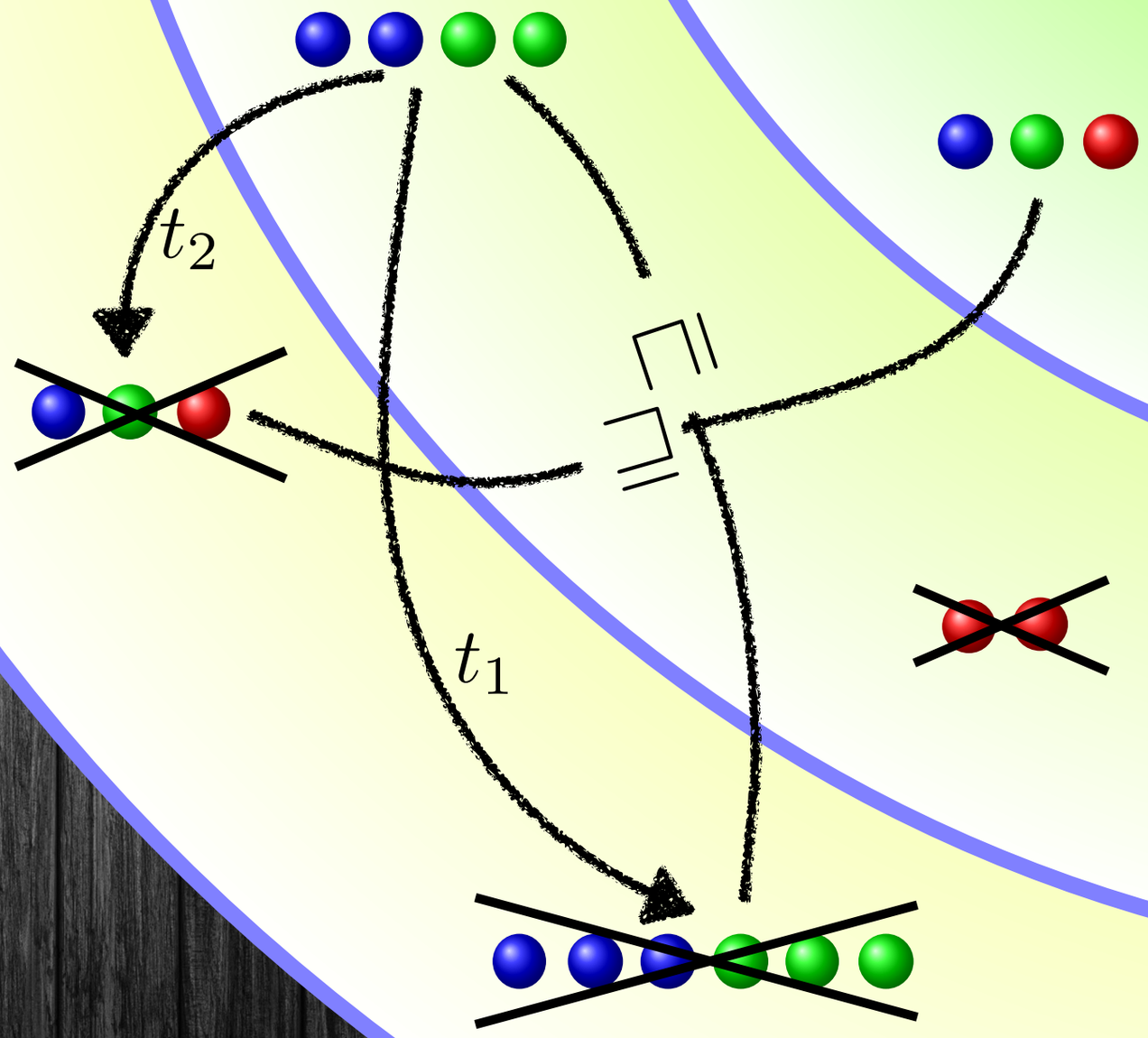
Backward Reachability



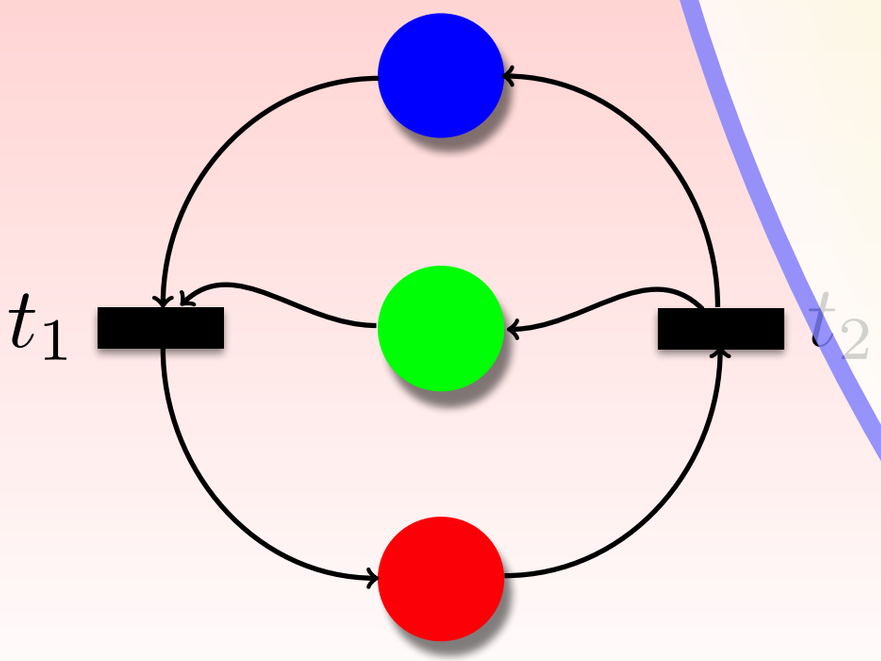


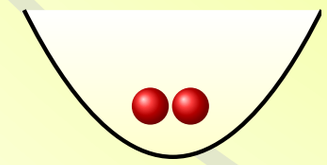
Backward Reachability





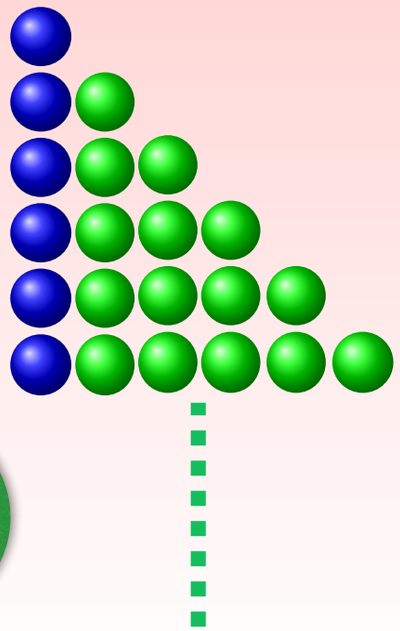
Backward Reachability



$Pre^*($  $)$

Petri

Backward Reachability



initial markings

System Safe !

symbolic representation = finite multisets

Termination: multisets well quasi-ordered

Petri

Well Quasi-Ordering

Well Quasi-Ordering

infinite sequence of markings

$$m_0, m_1, m_2, \ldots, m_i, \ldots m_j \ldots$$

$$\sqsubseteq$$

$$\exists i < j : m_i \sqsubseteq m_j$$

Petri

Backward Reach

Ordering:

- monotonicity
- computing predecessors
- well quasi-ordering

initial markings

sets

ordered

Background

Parameterized Systems

Petri Nets

Lossy Channel Systems

Timed Petri Nets

Lossy Channel Systems

Lossy Channel Systems

Model

Configurations

Transitions

Ordering

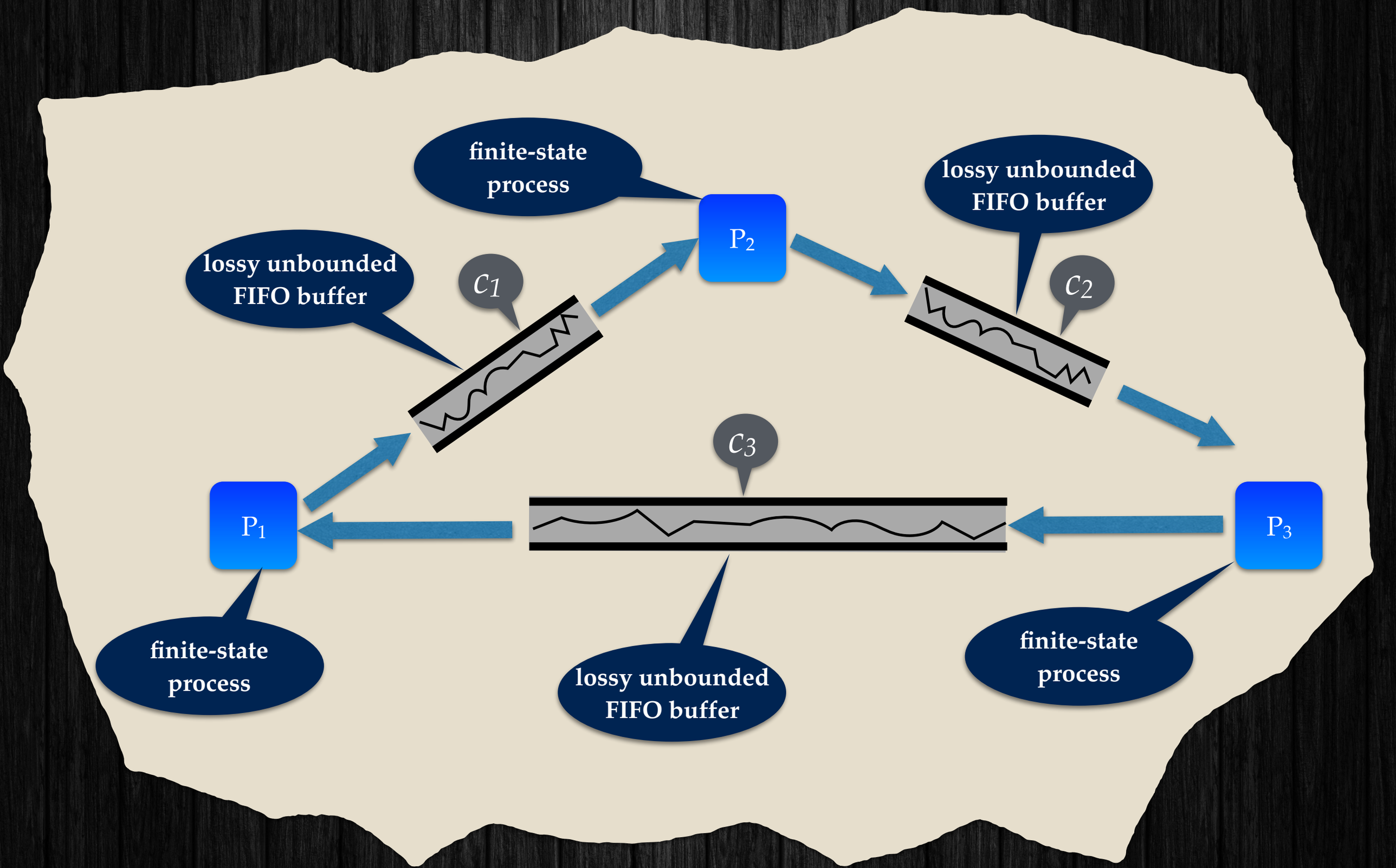
Monotoncity

Upward Closed Sets

Computing Predecessors

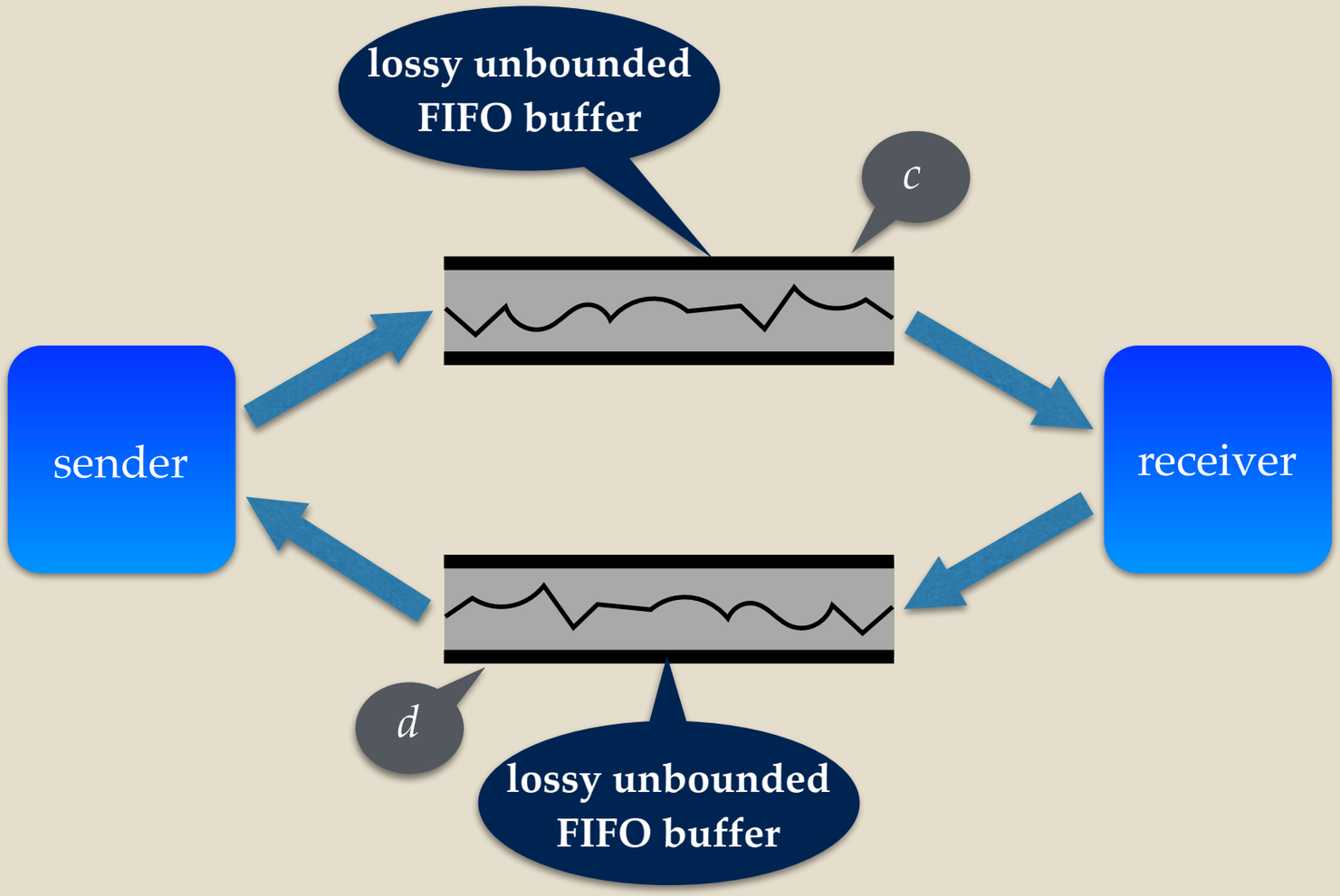
Backward Reachability

Lossy C… Model

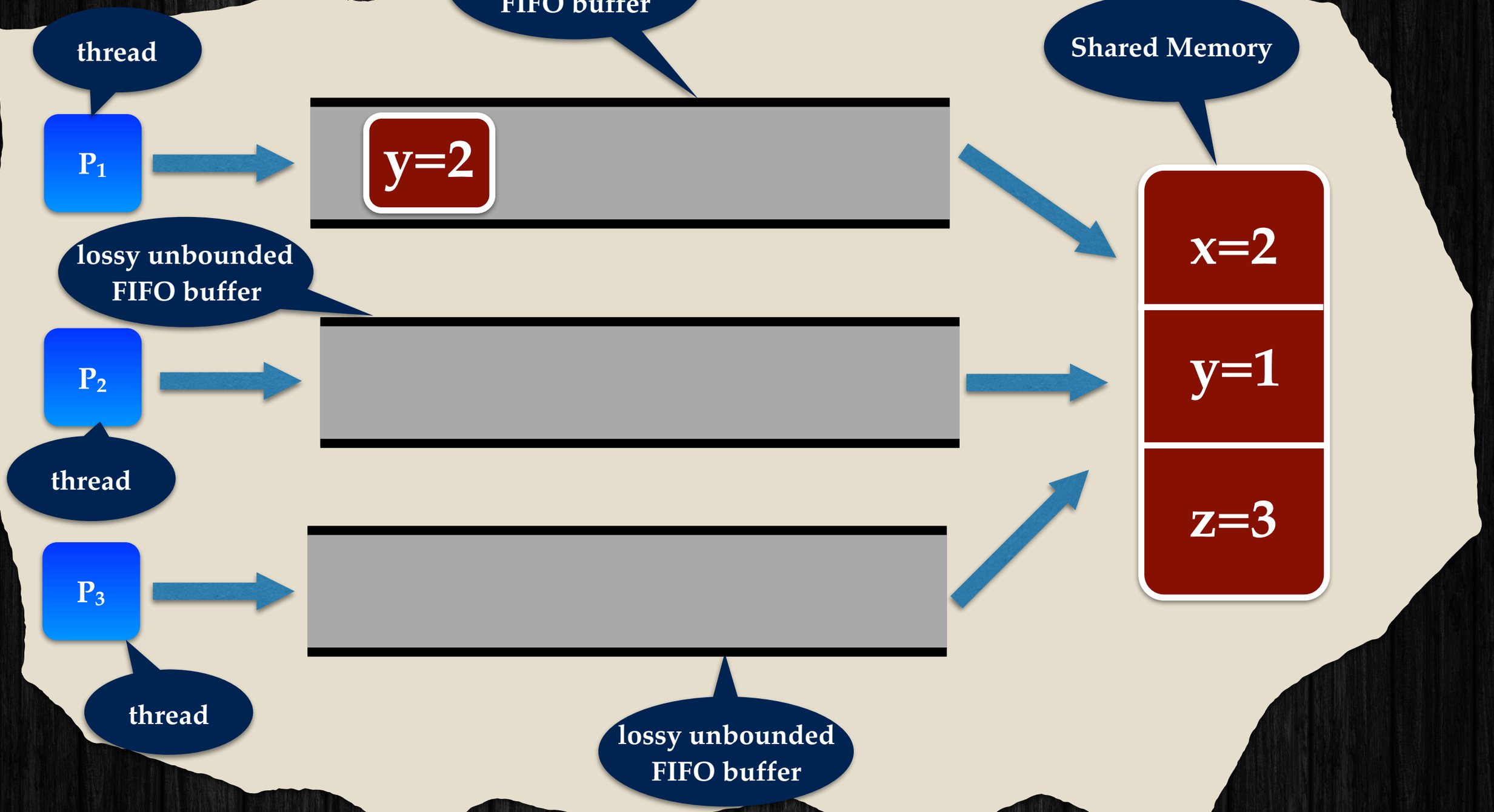


Lossy C… M…

Telecommunication Protocols

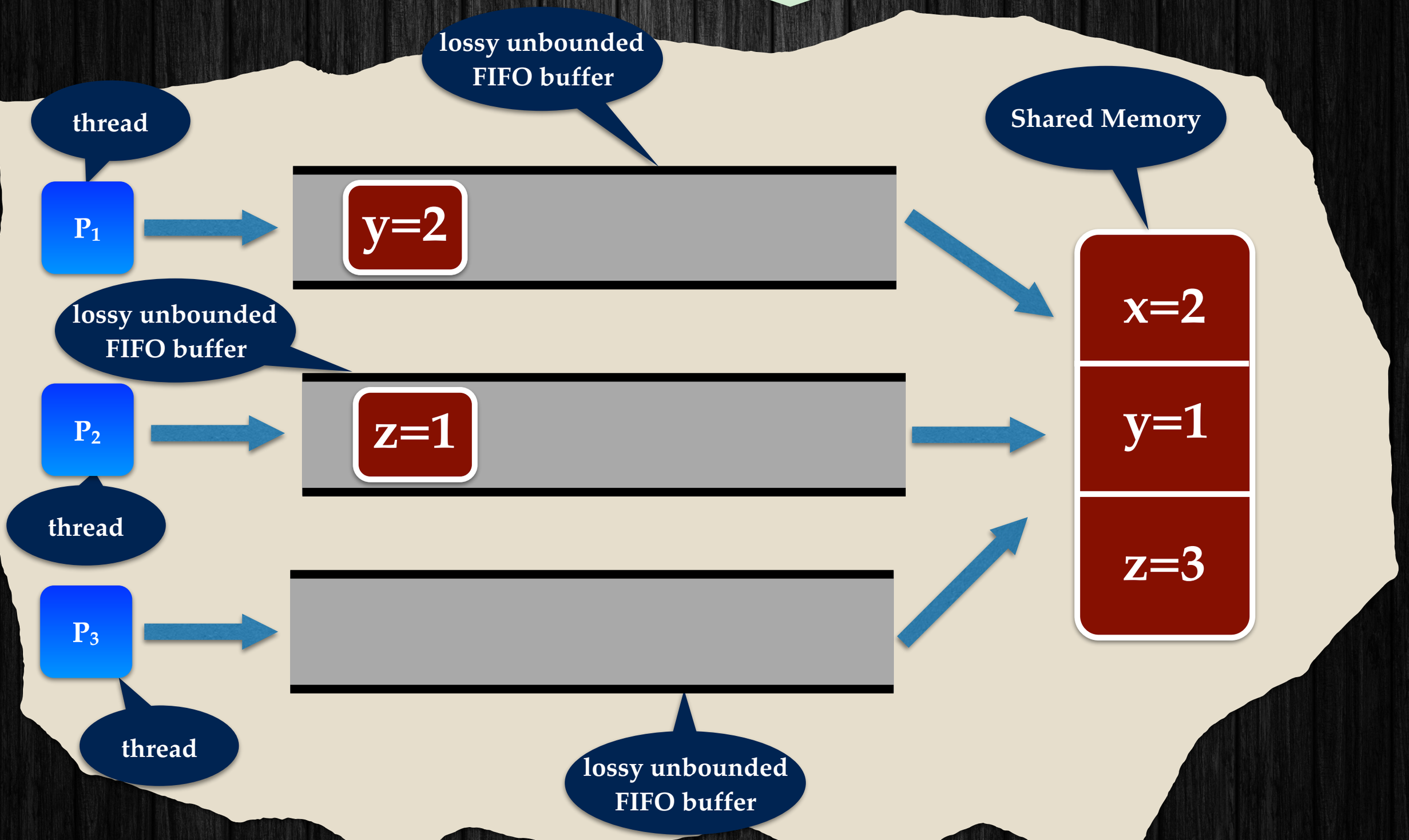


Weak Memory Models



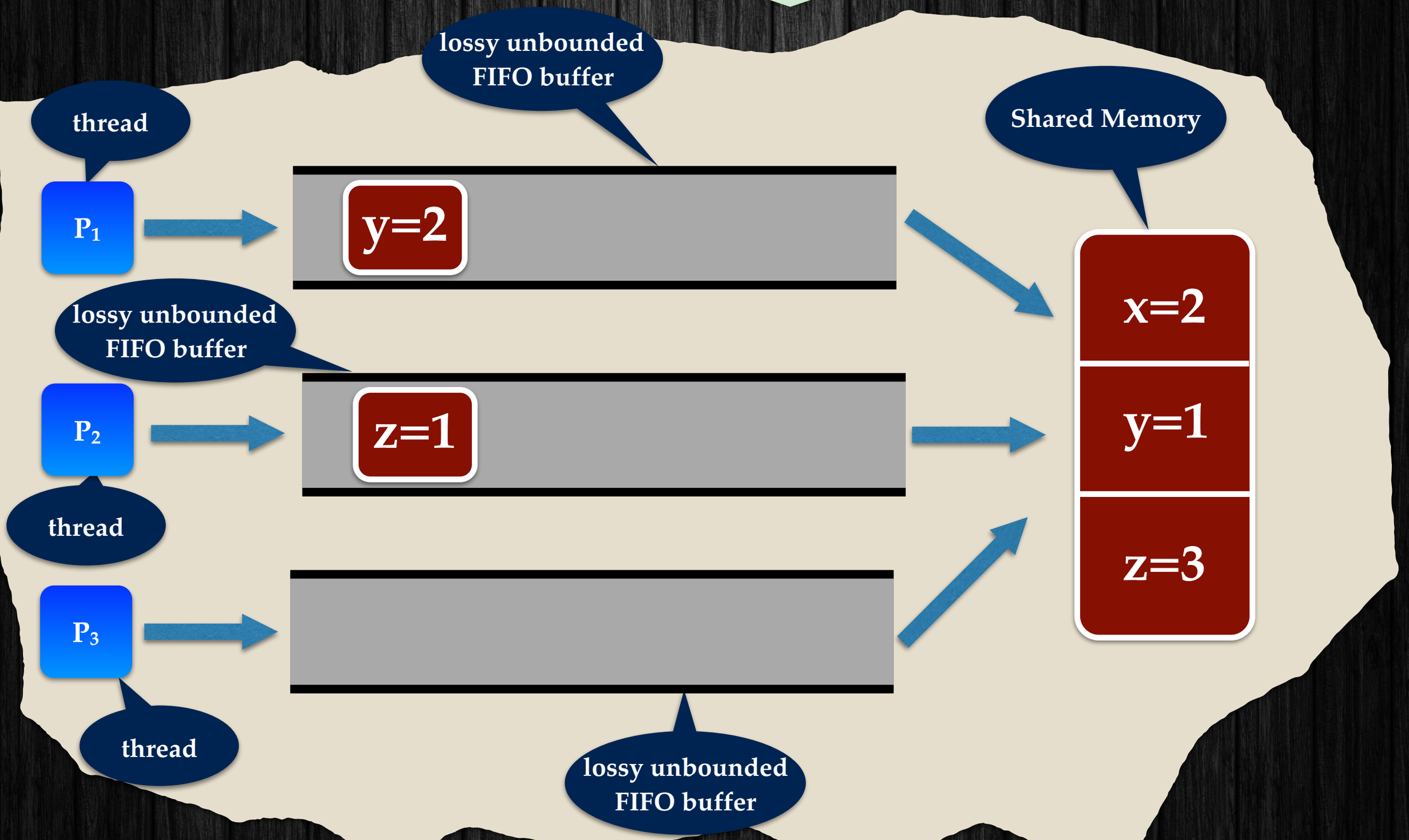
Lossy C… M…

Weak Memory Models



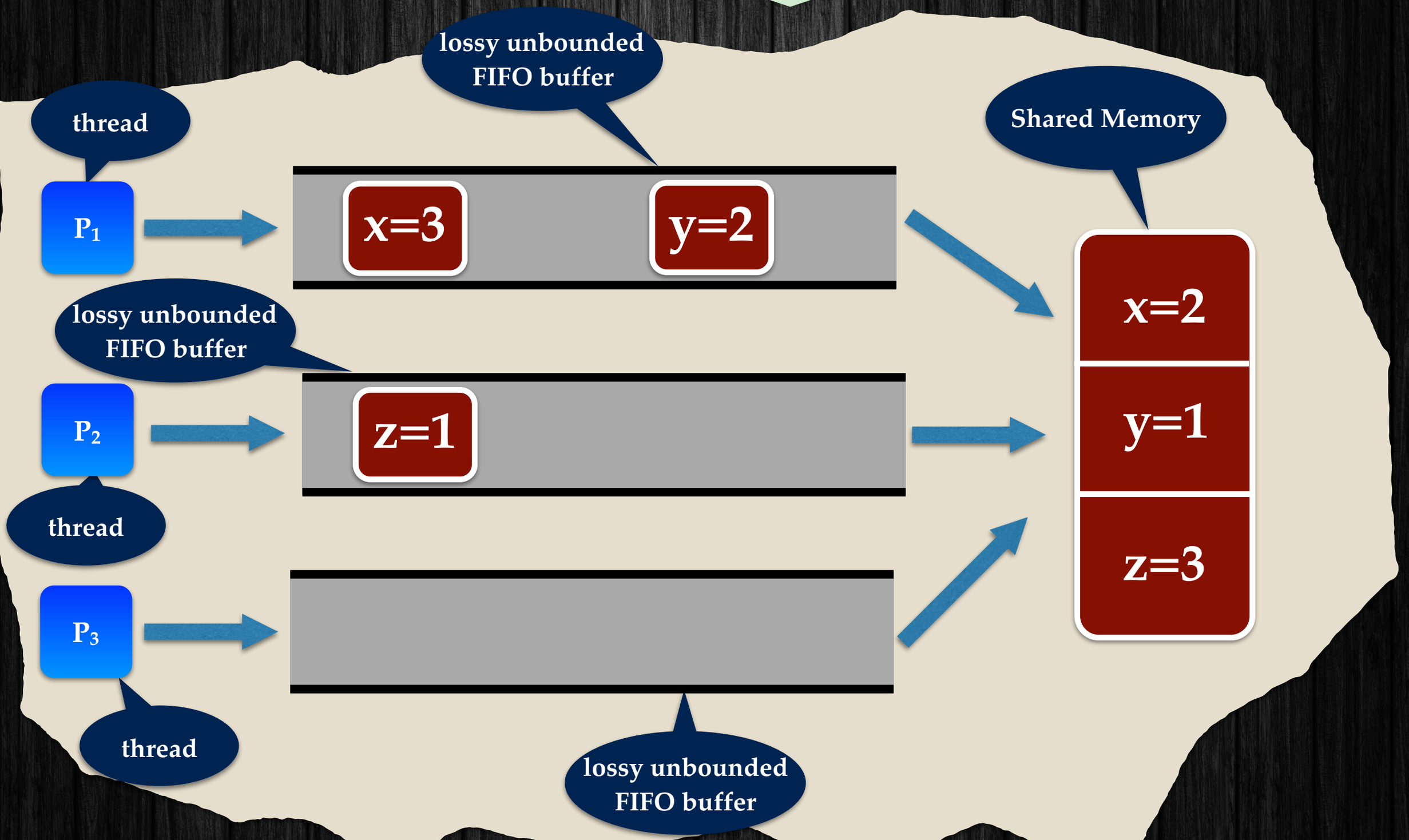
Lossy C… M…

Weak Memory Models



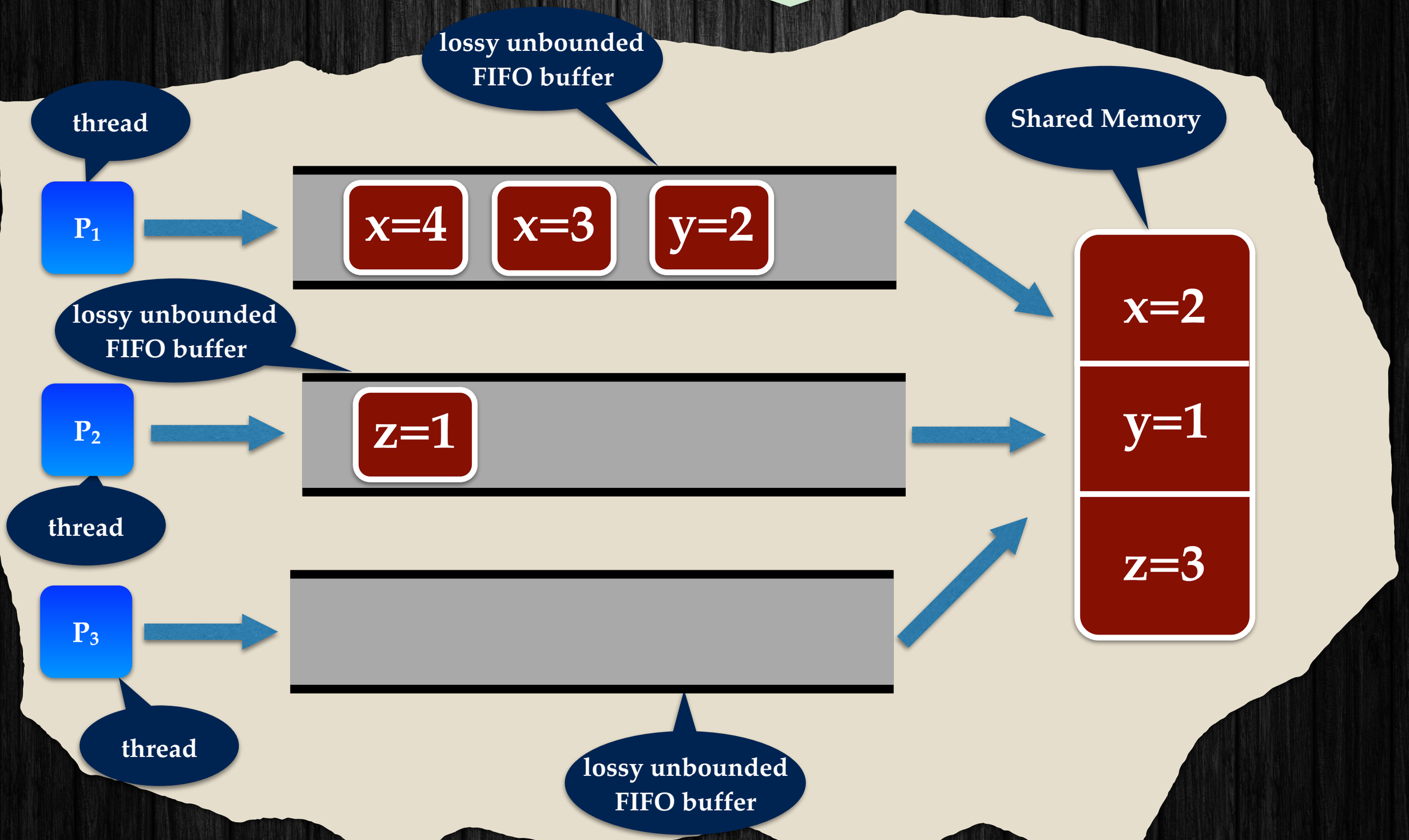
Lossy C… M…

Weak Memory Models



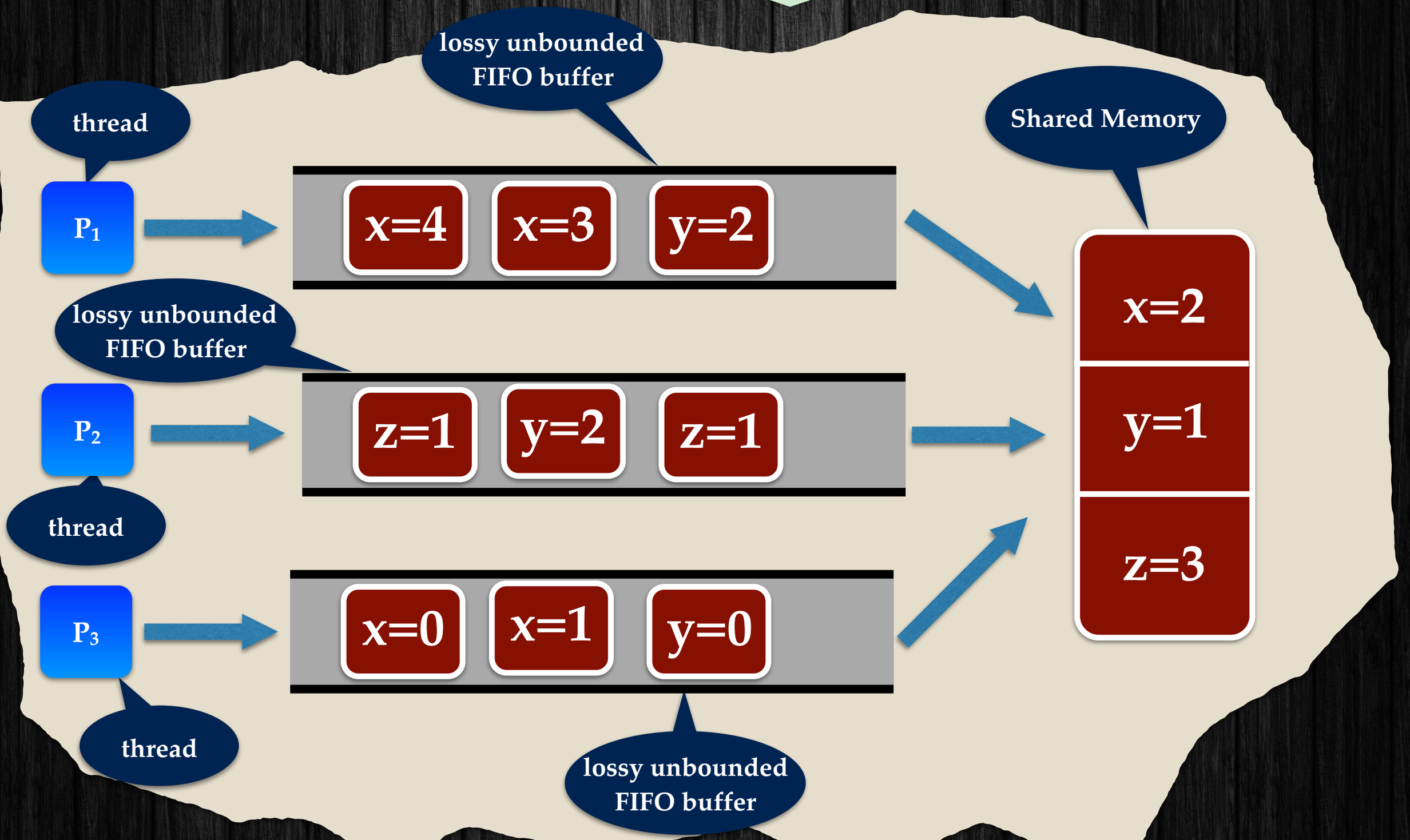
Lossy C… M…

Weak Memory Models



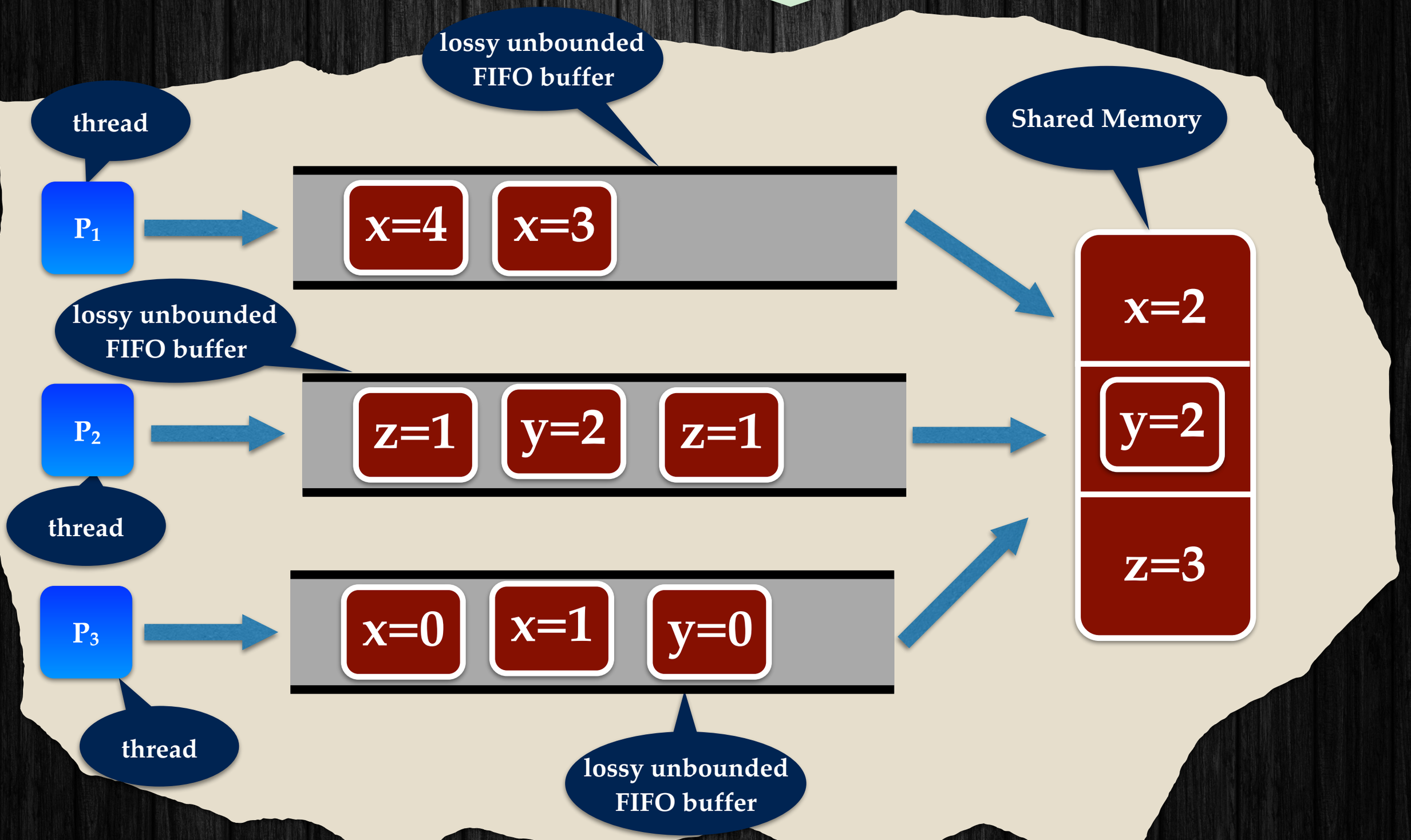
Lossy C… M…

Weak Memory Models



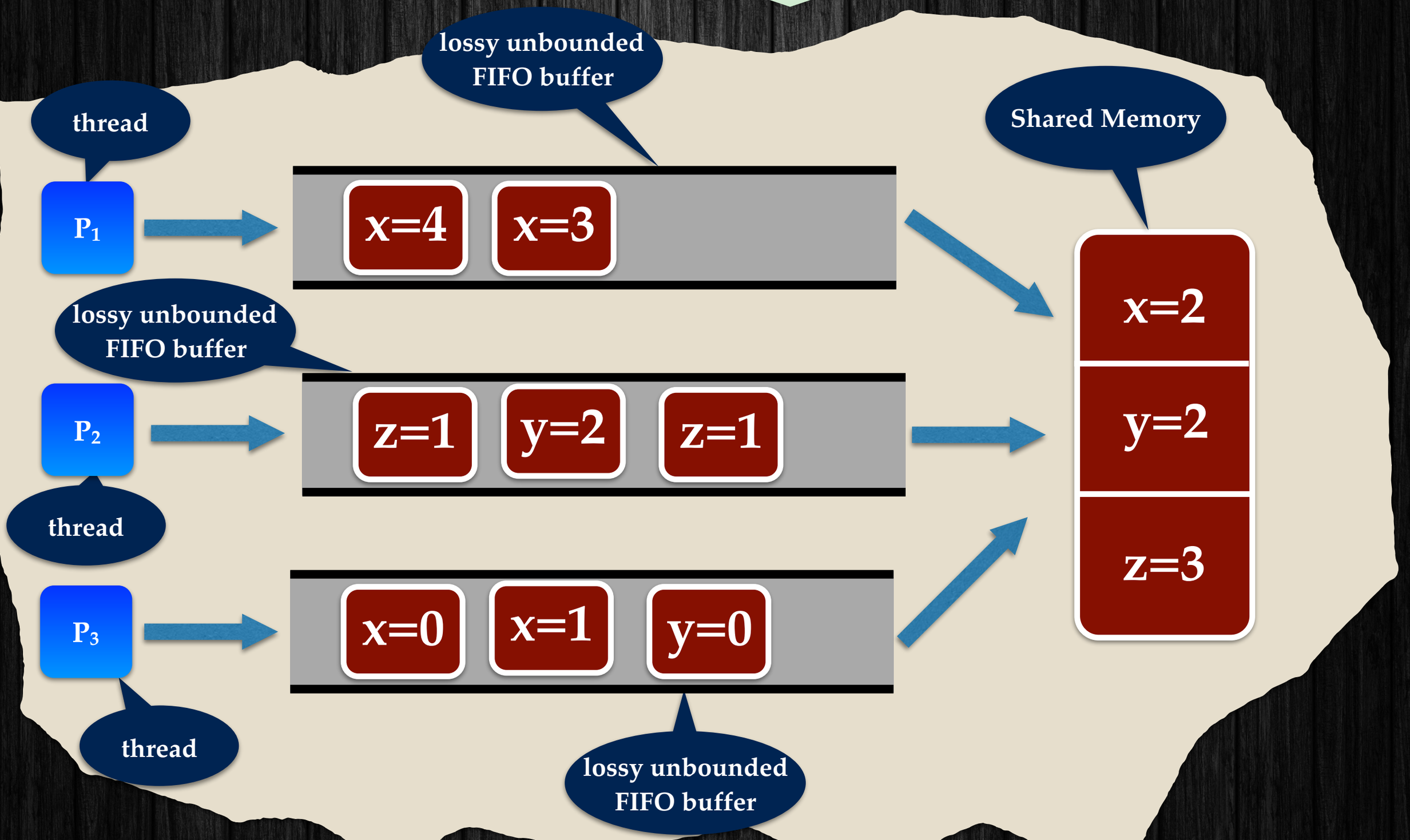
Lossy C… M…

Weak Memory Models



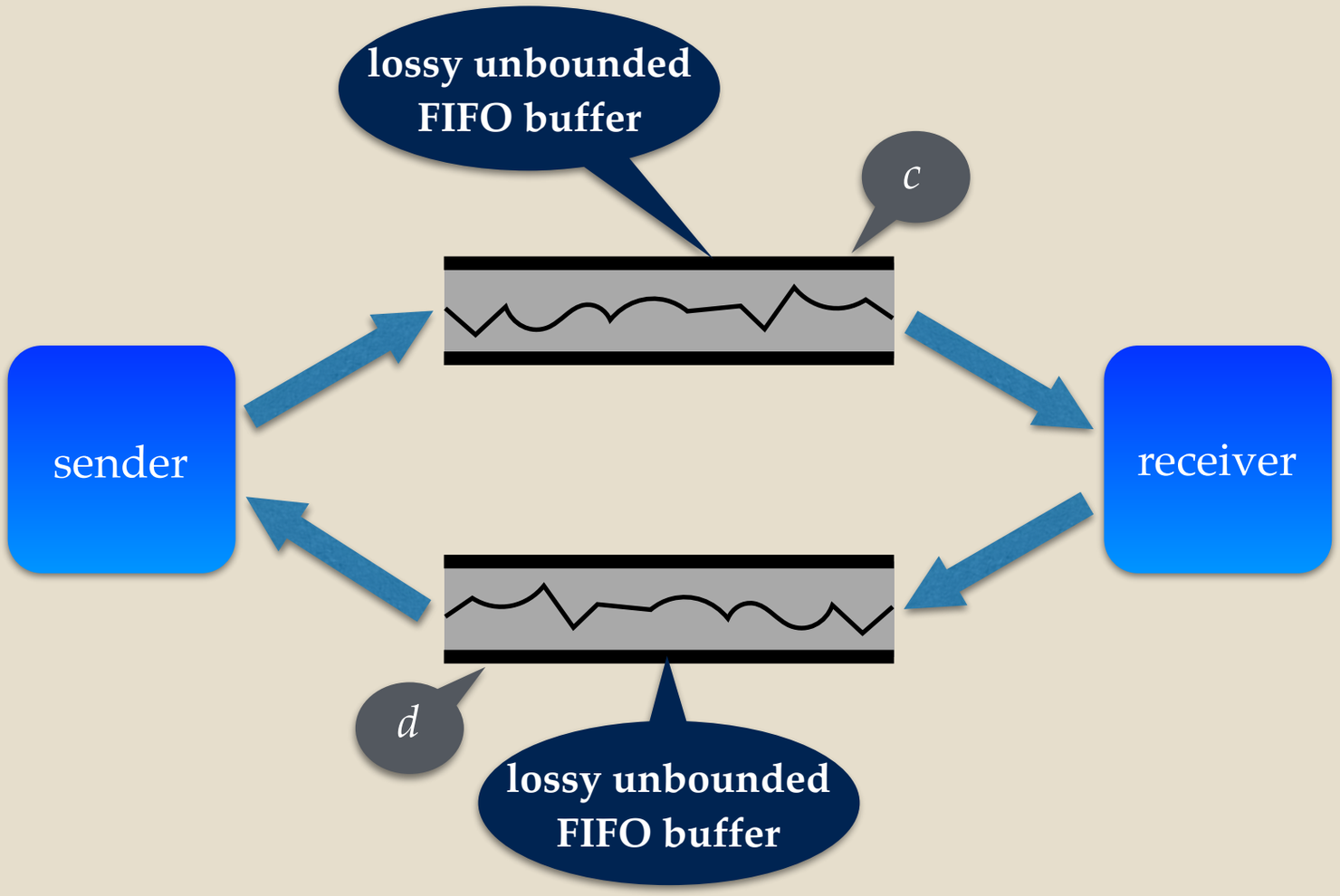
Lossy C… M…

Weak Memory Models

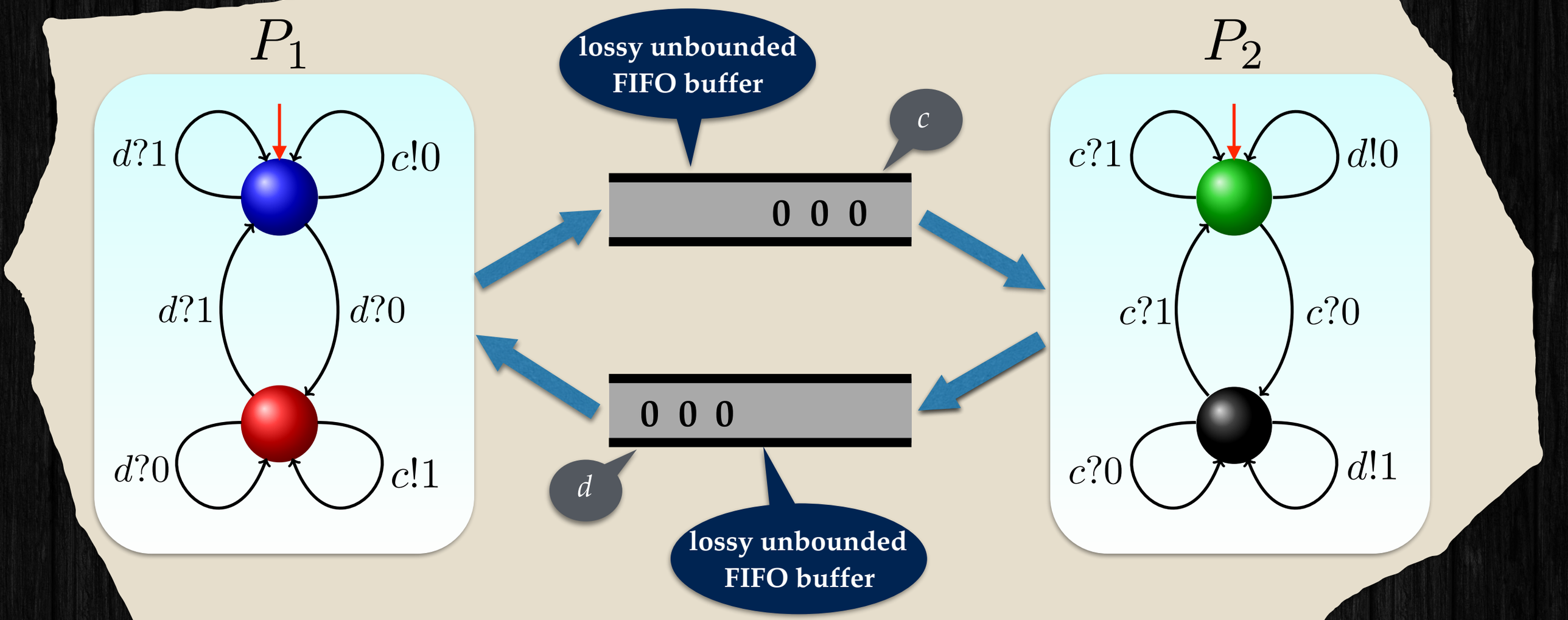


Lossy C… M…

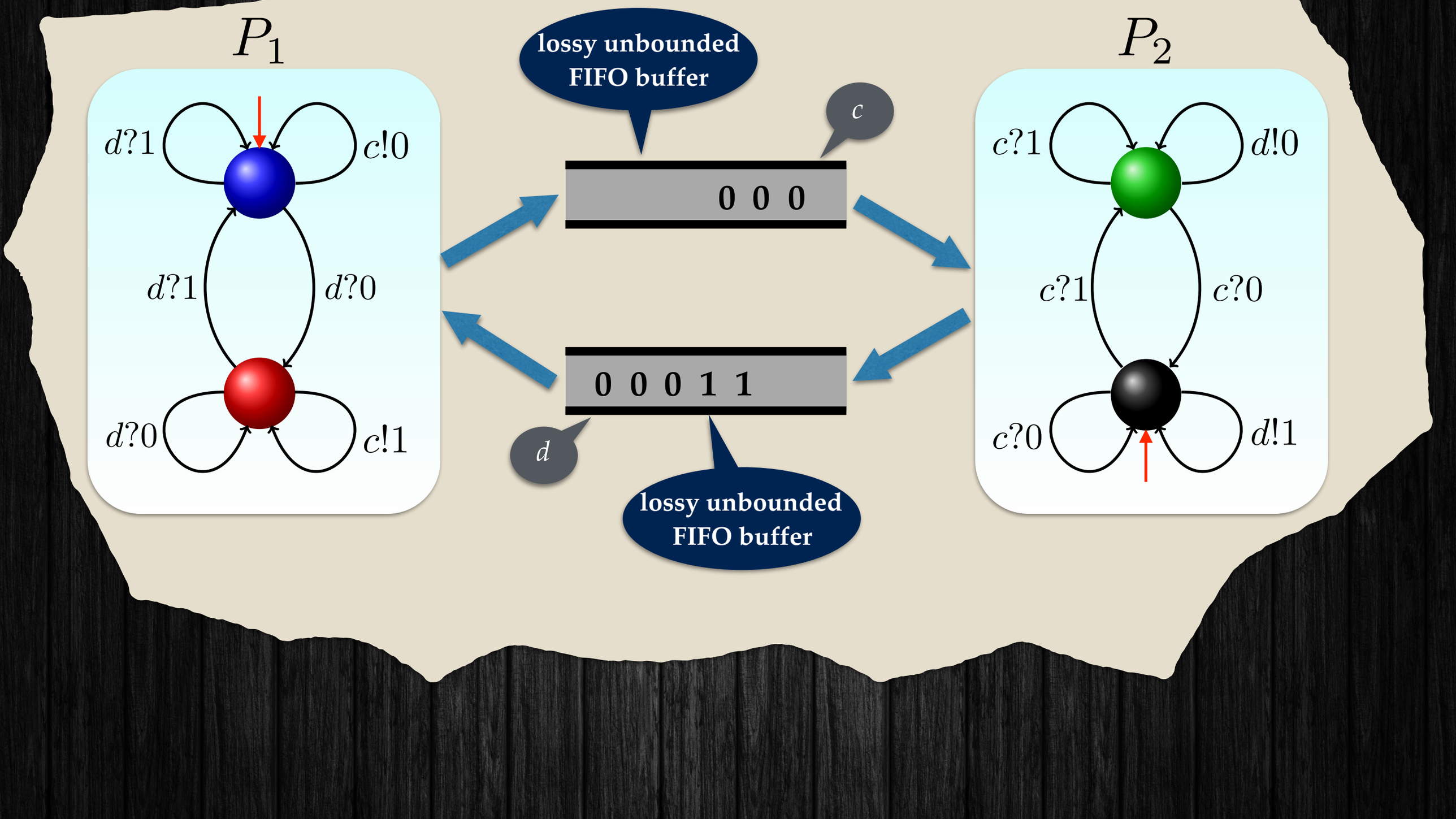
Telecommunication Protocol



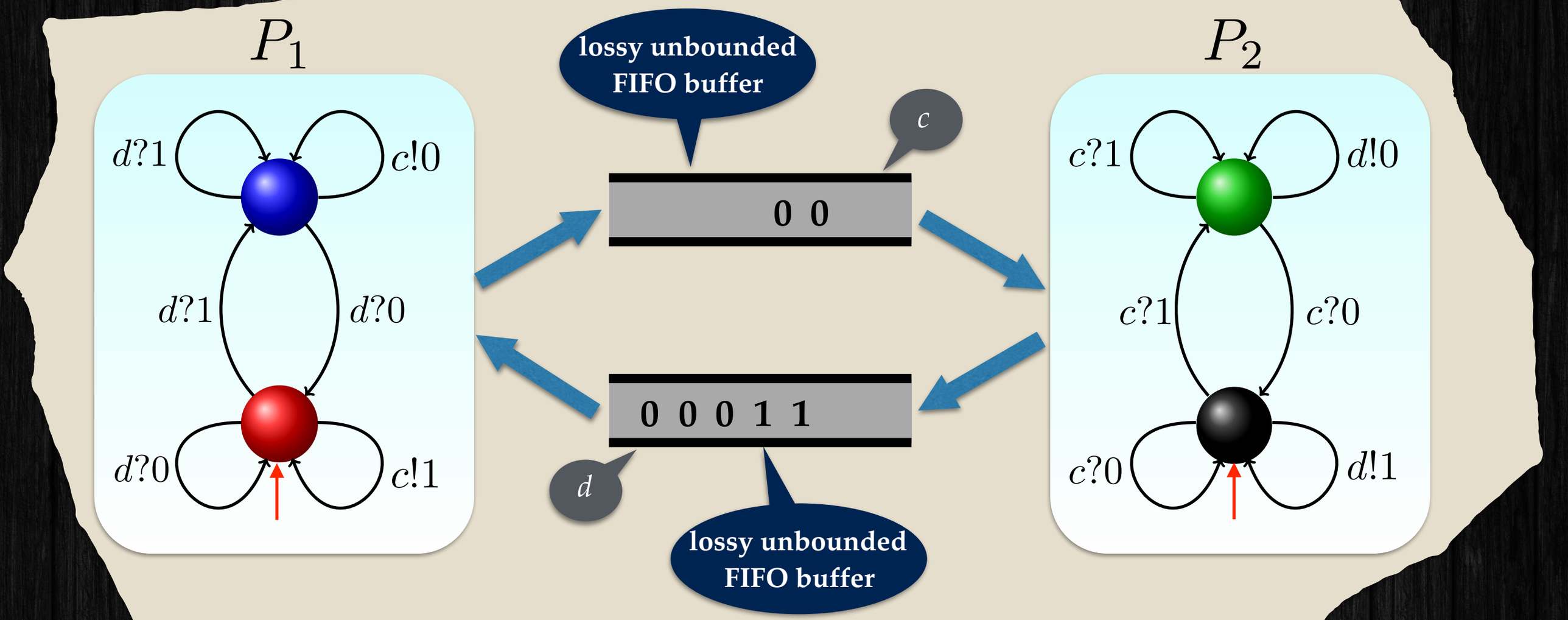
Lossy Channel Systems



Lossy Channel Systems



Lossy Channel Systems



Lossy Channel Systems

Model

Configurations

Transitions

Ordering

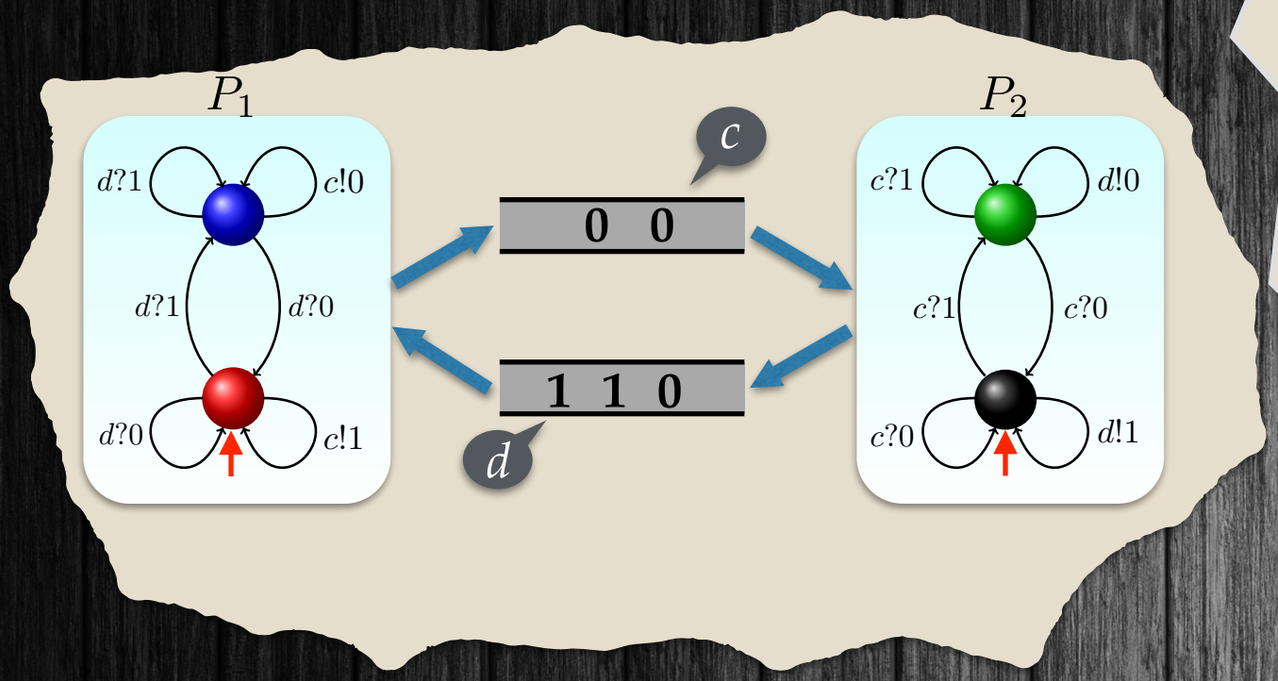
Monotoncity

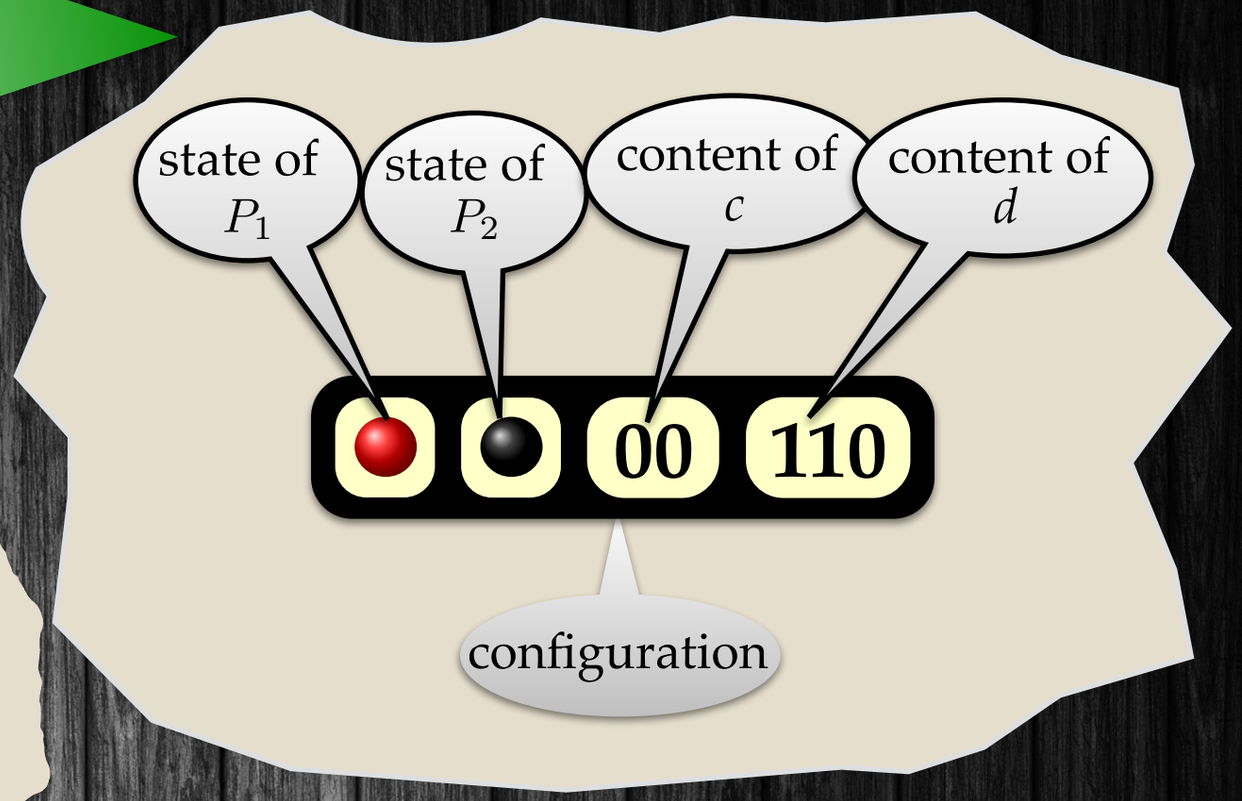
Upward Closed Sets

Computing Predecessors

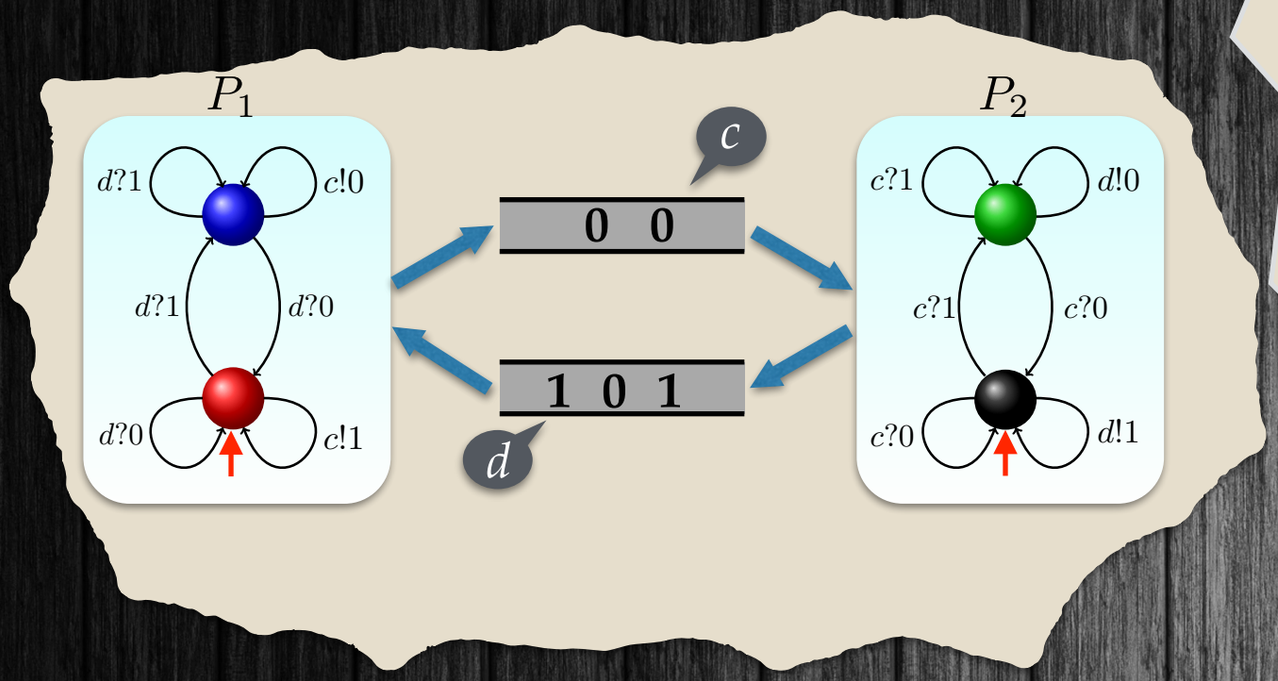
Backward Reachability

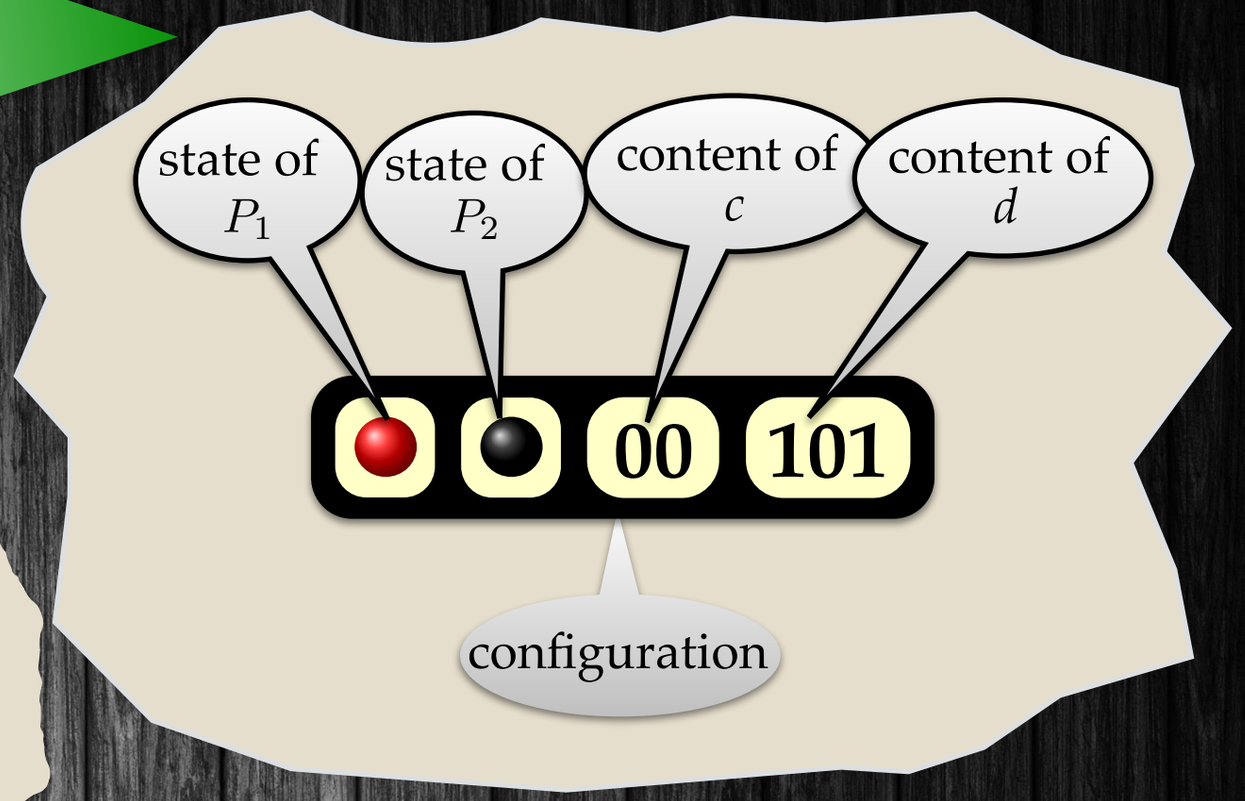
Lossy Configurations



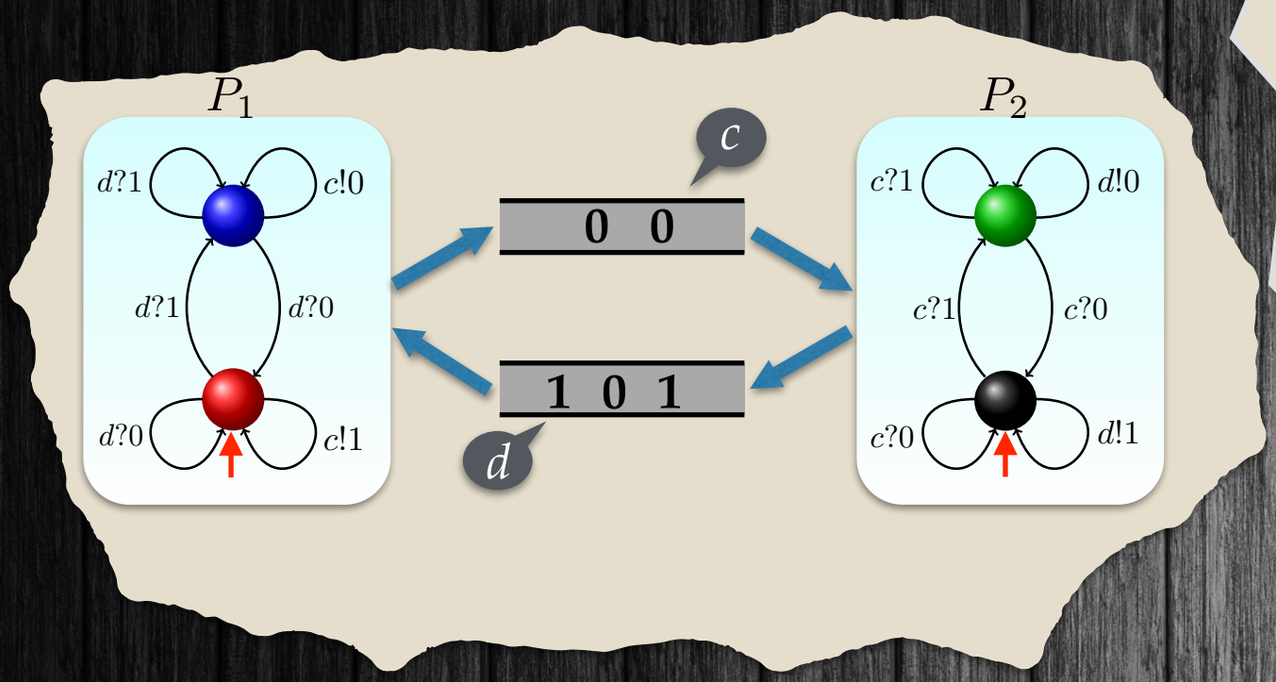


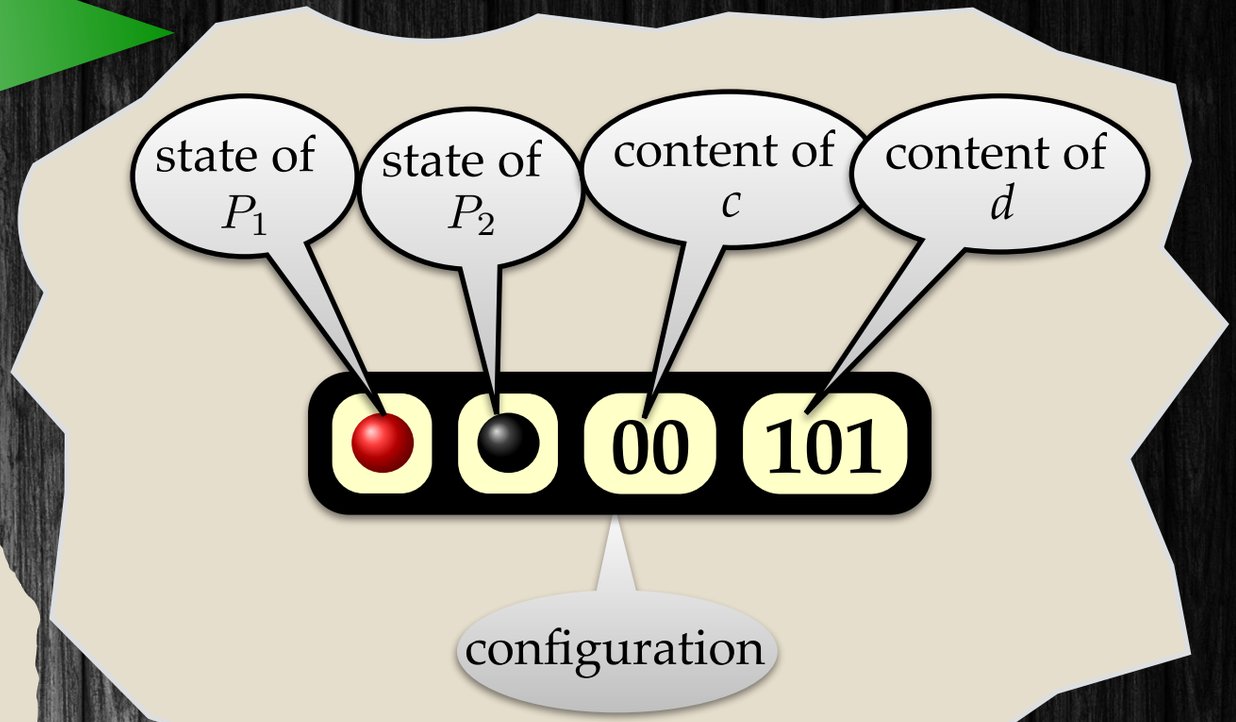
Lossy Configurations

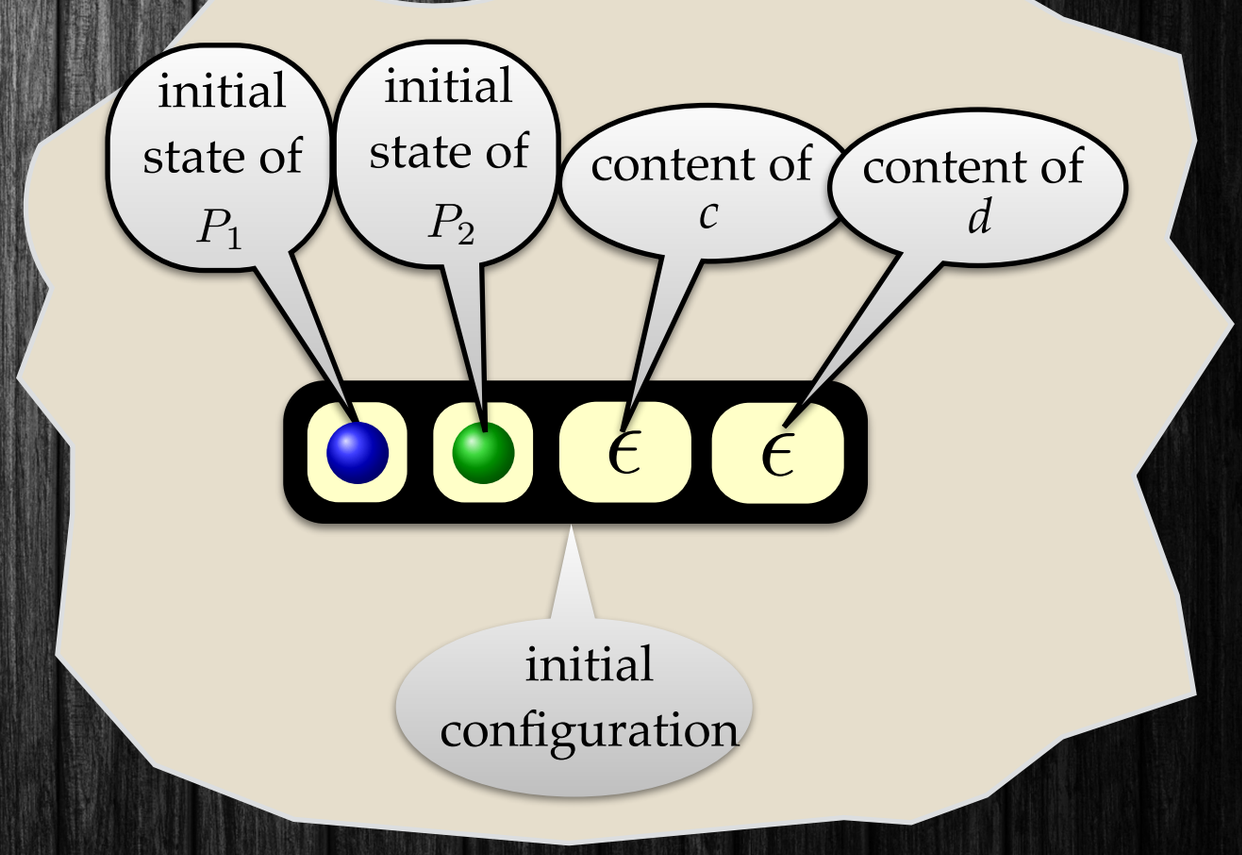




Lossy Configurations







Lossy Channel Systems

Model ✓

Configurations ✓

Transitions

Ordering

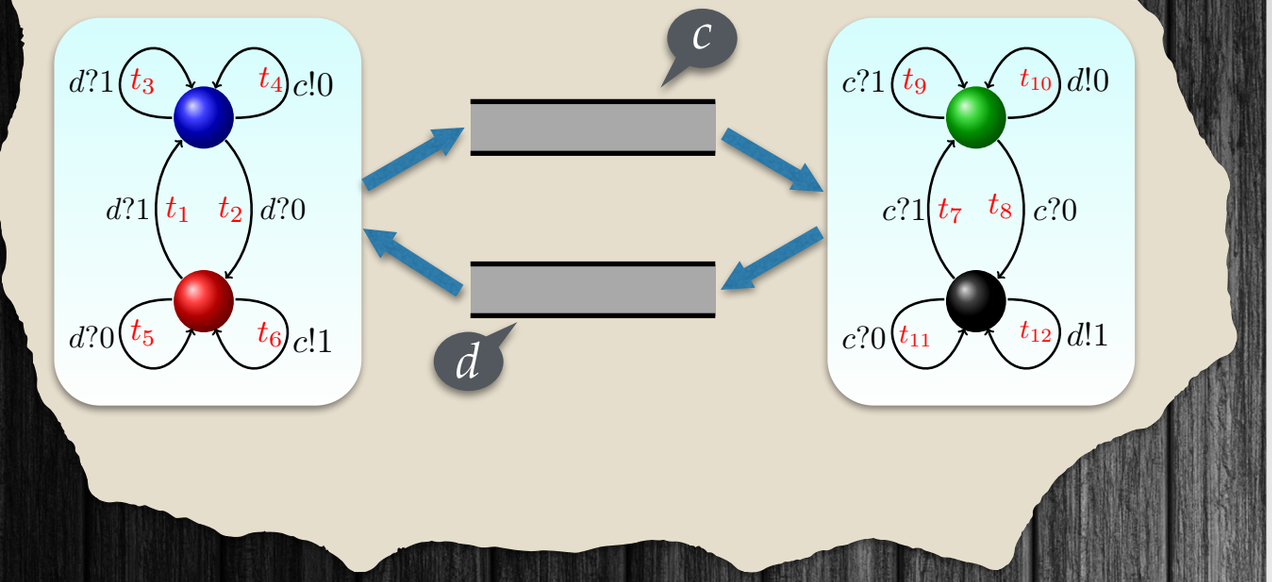
Monotoncity

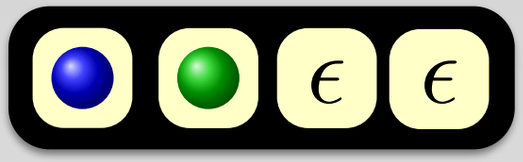
Upward Closed Sets

Computing Predecessors

Backward Reachability

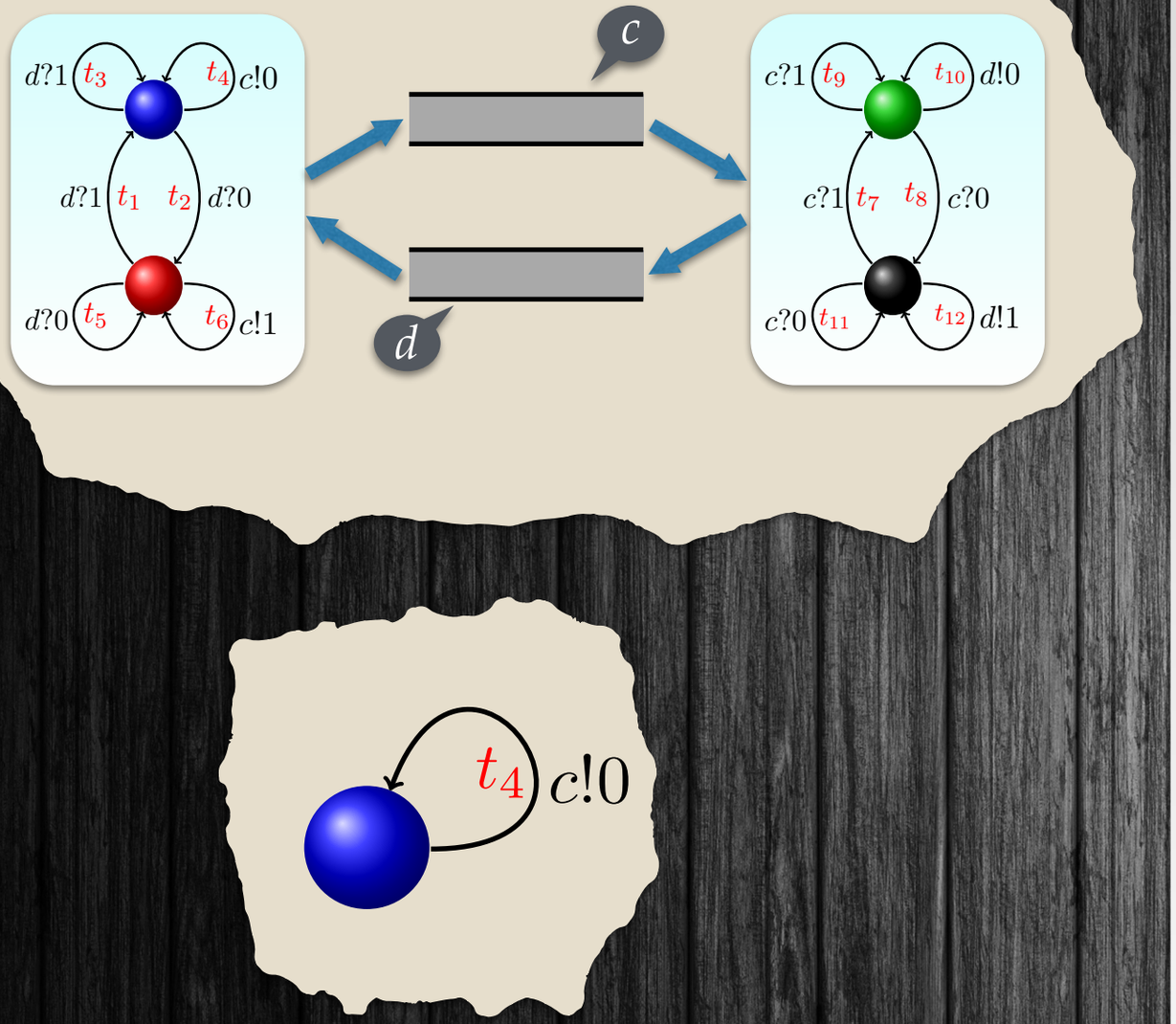
Lossy Transitions

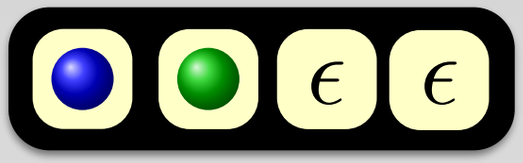




Lossy

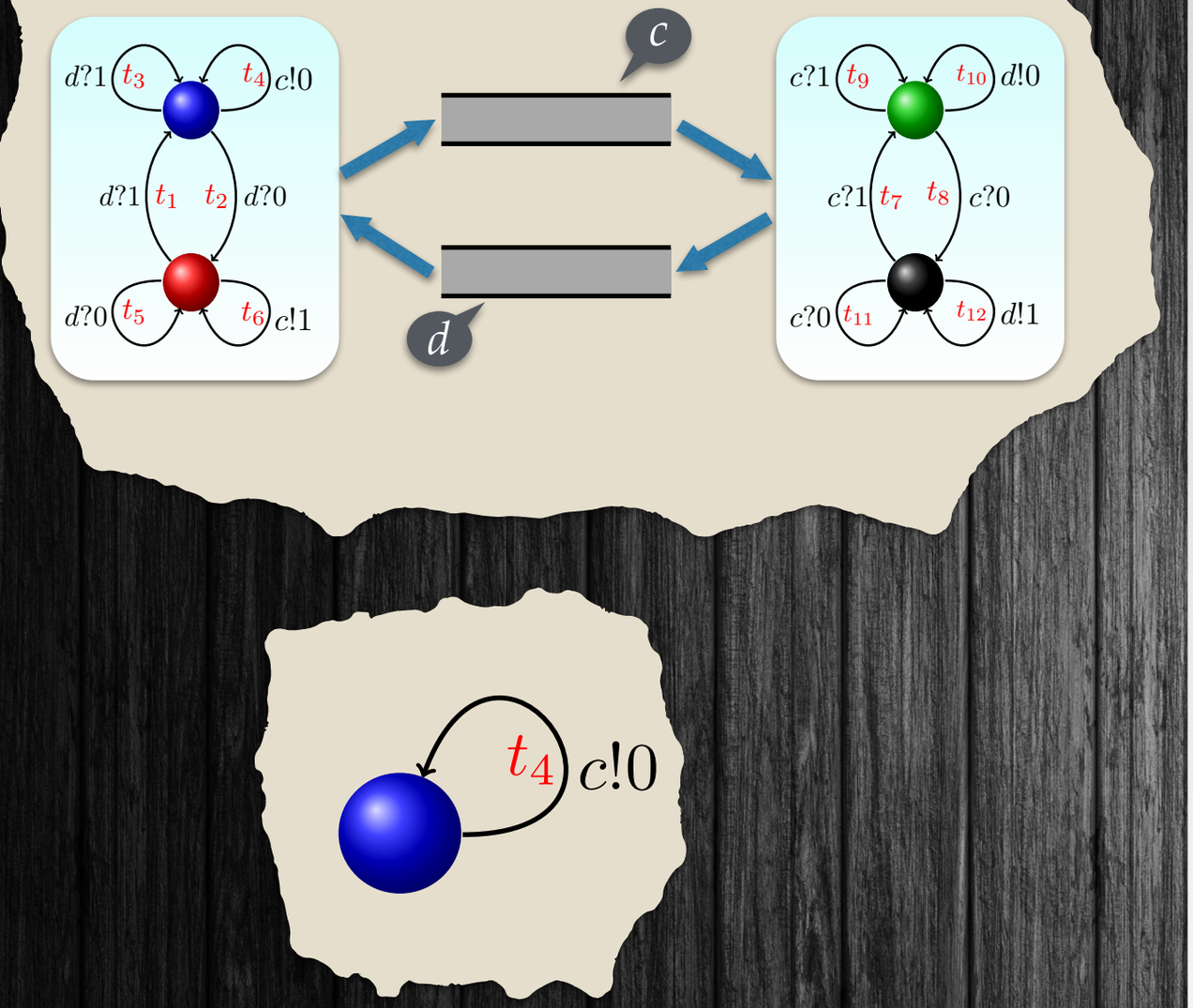
Transitions

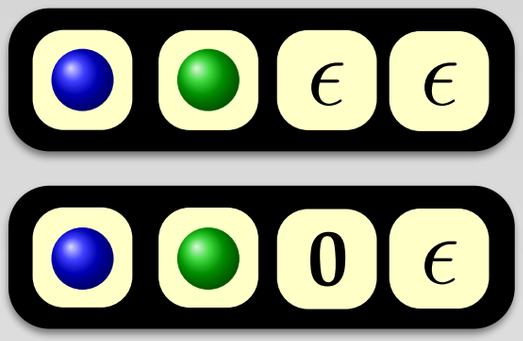


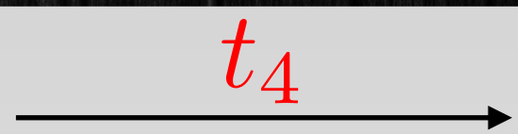


Lossy

Transitions

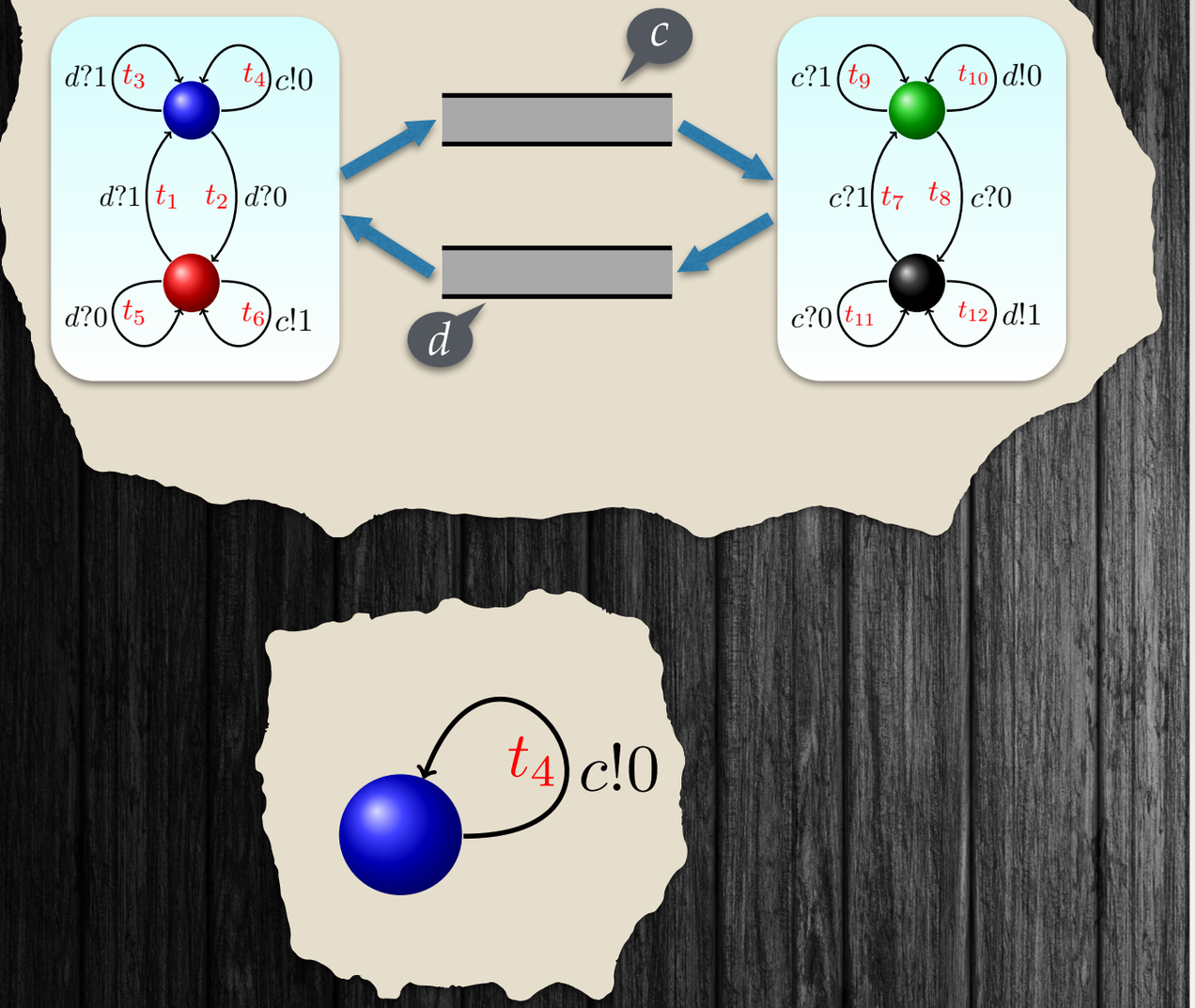


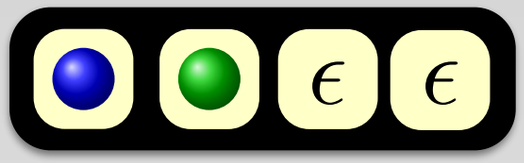




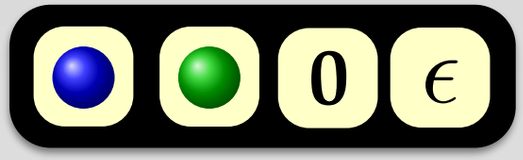
Lossy

Transitions





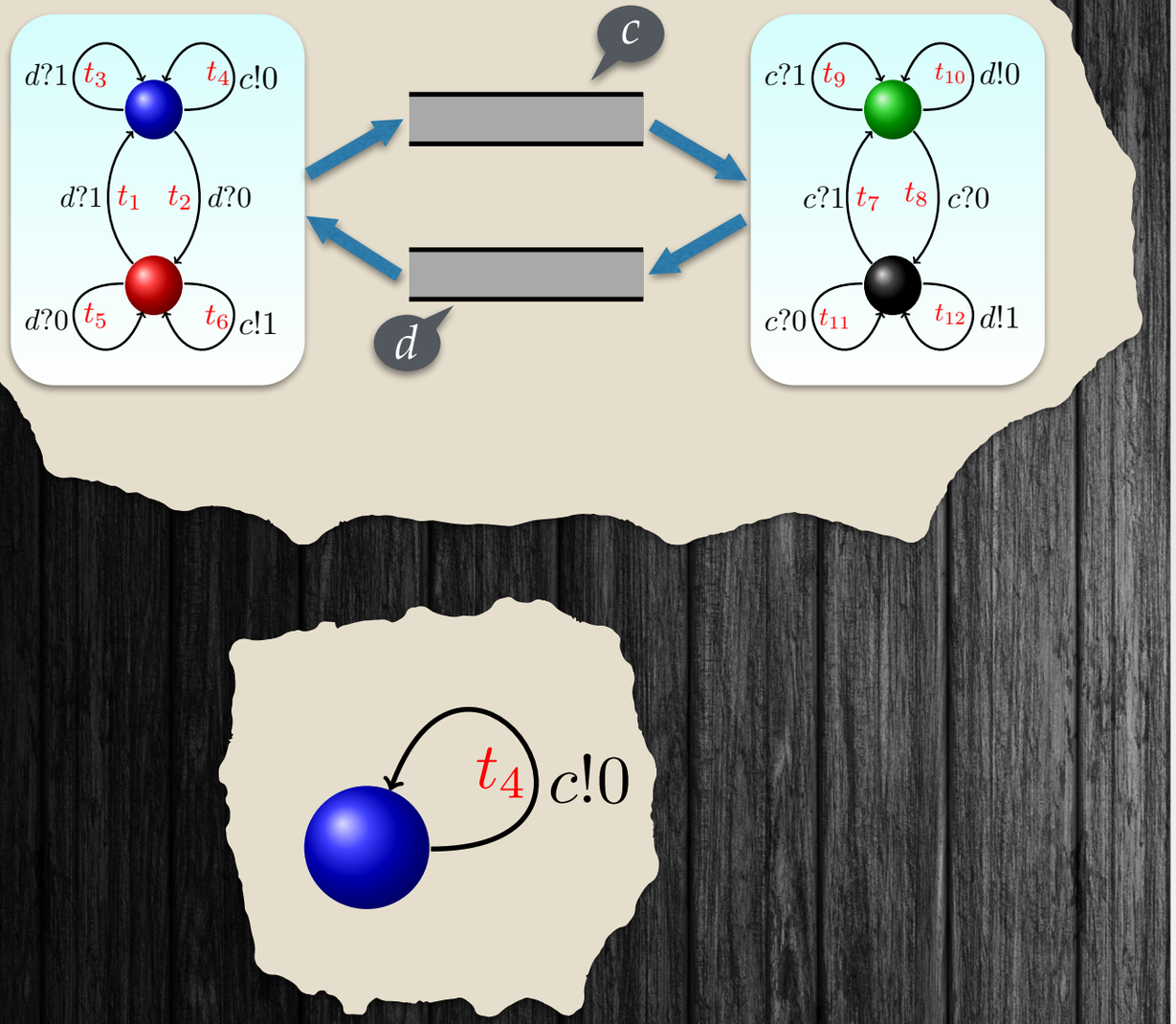
t_4

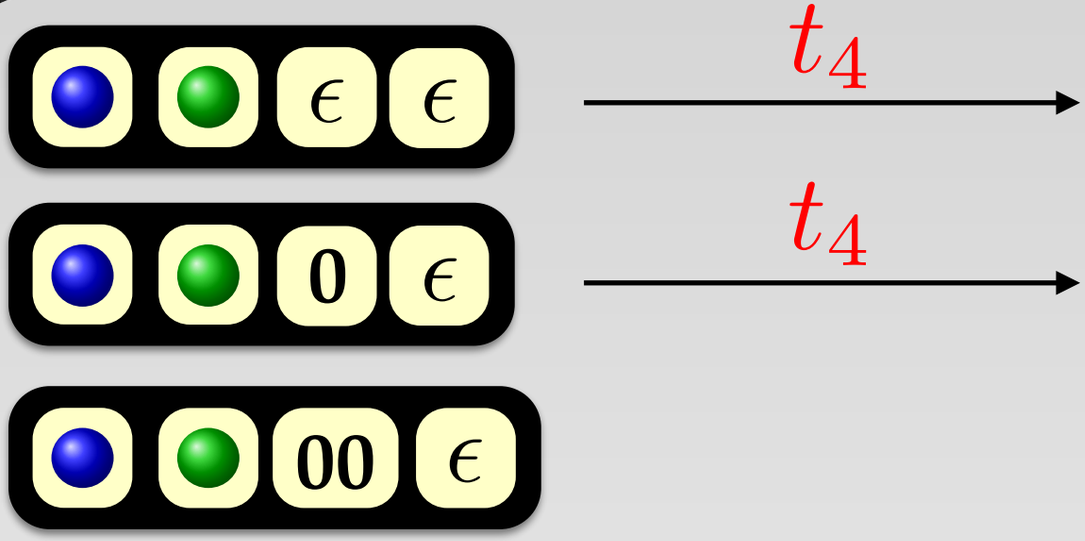


t_4

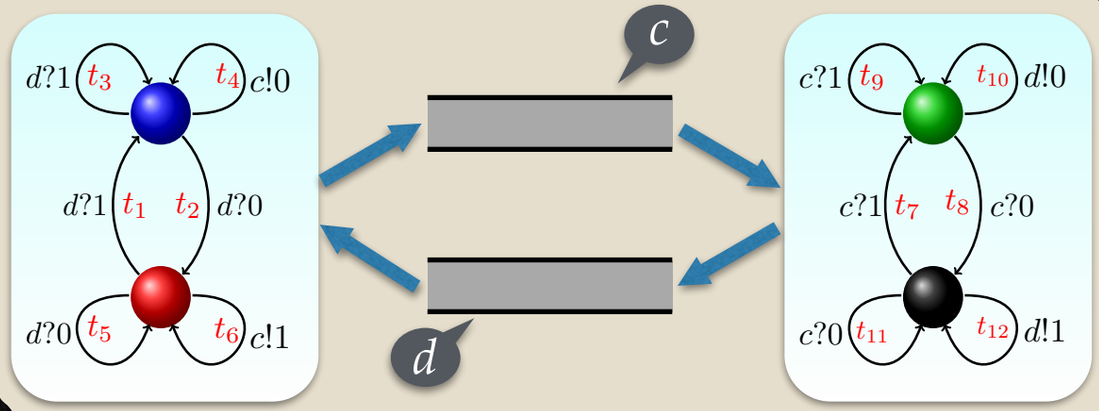
Lossy

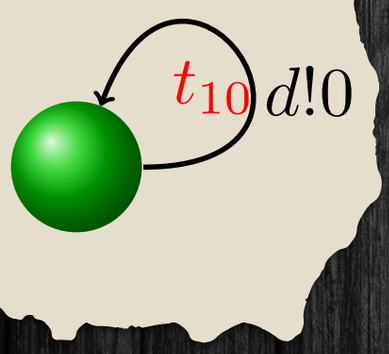
Transitions

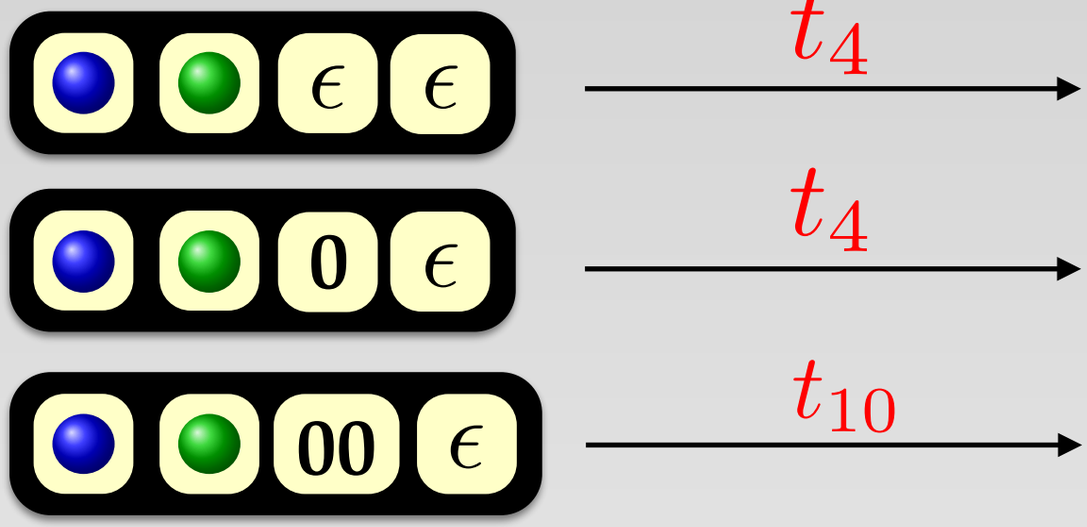




Lossy Transitions

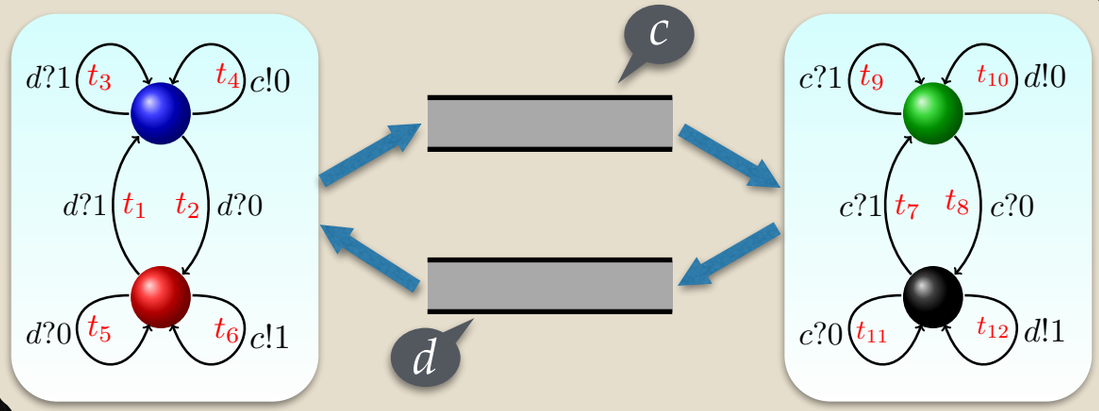


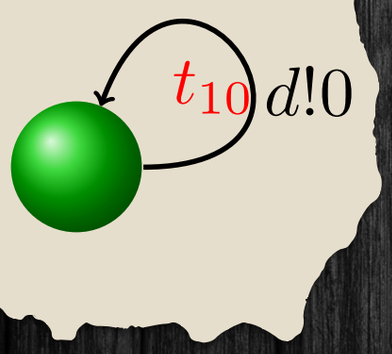


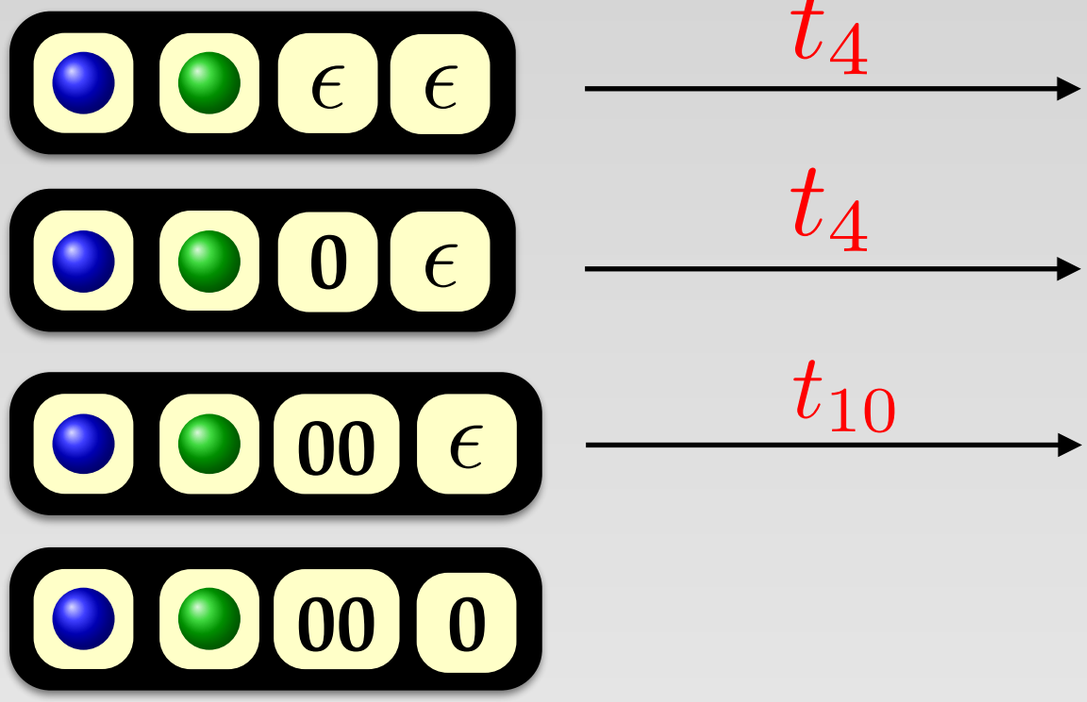


Lossy

Transitions

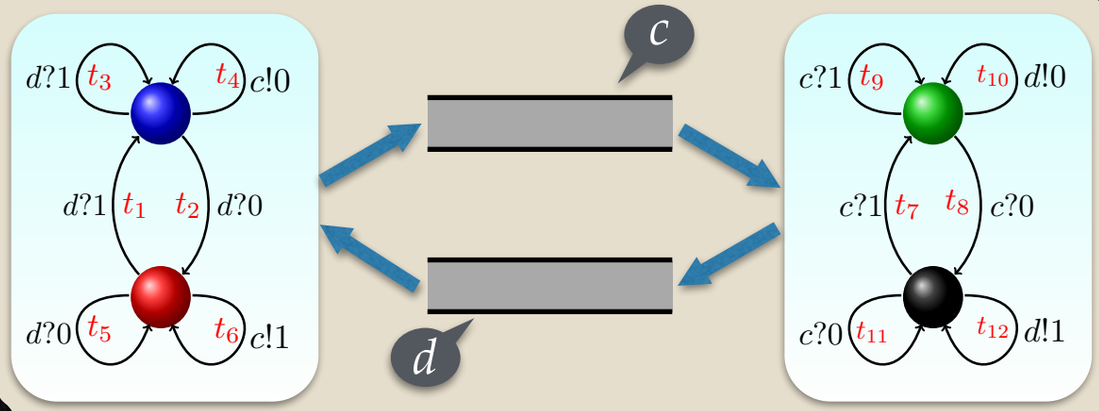


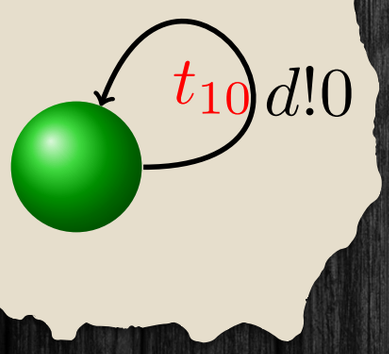


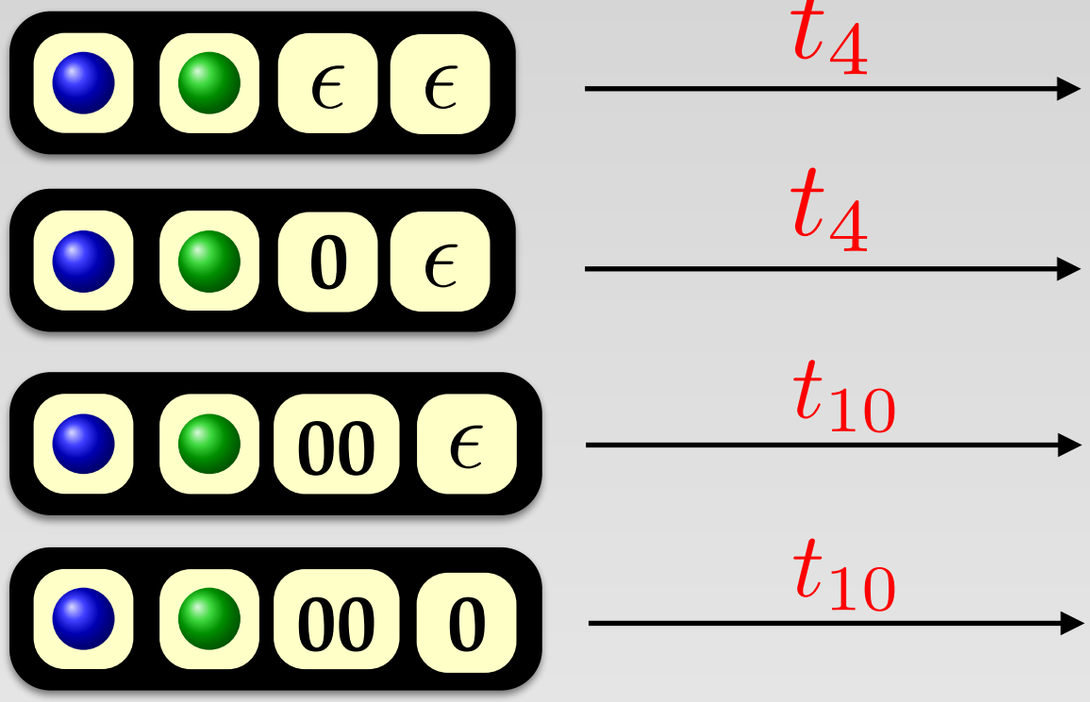


Lossy

Transitions

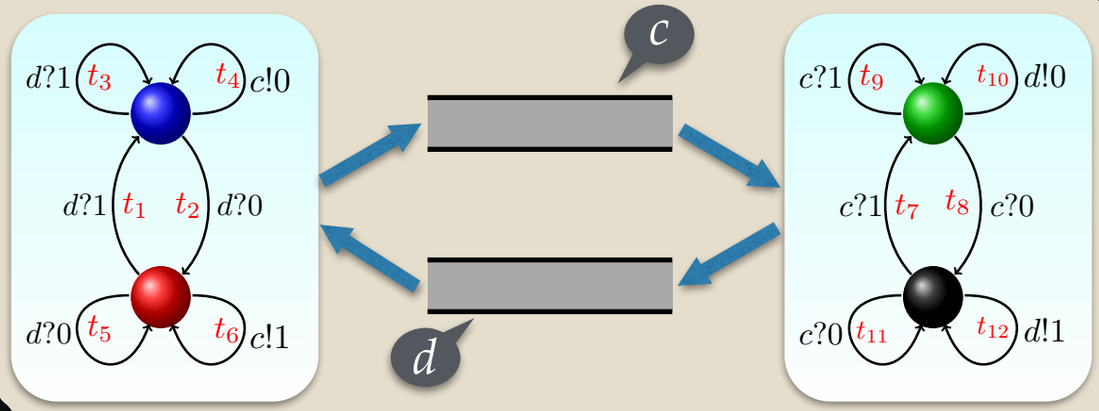


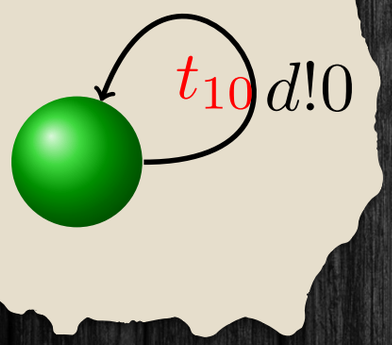


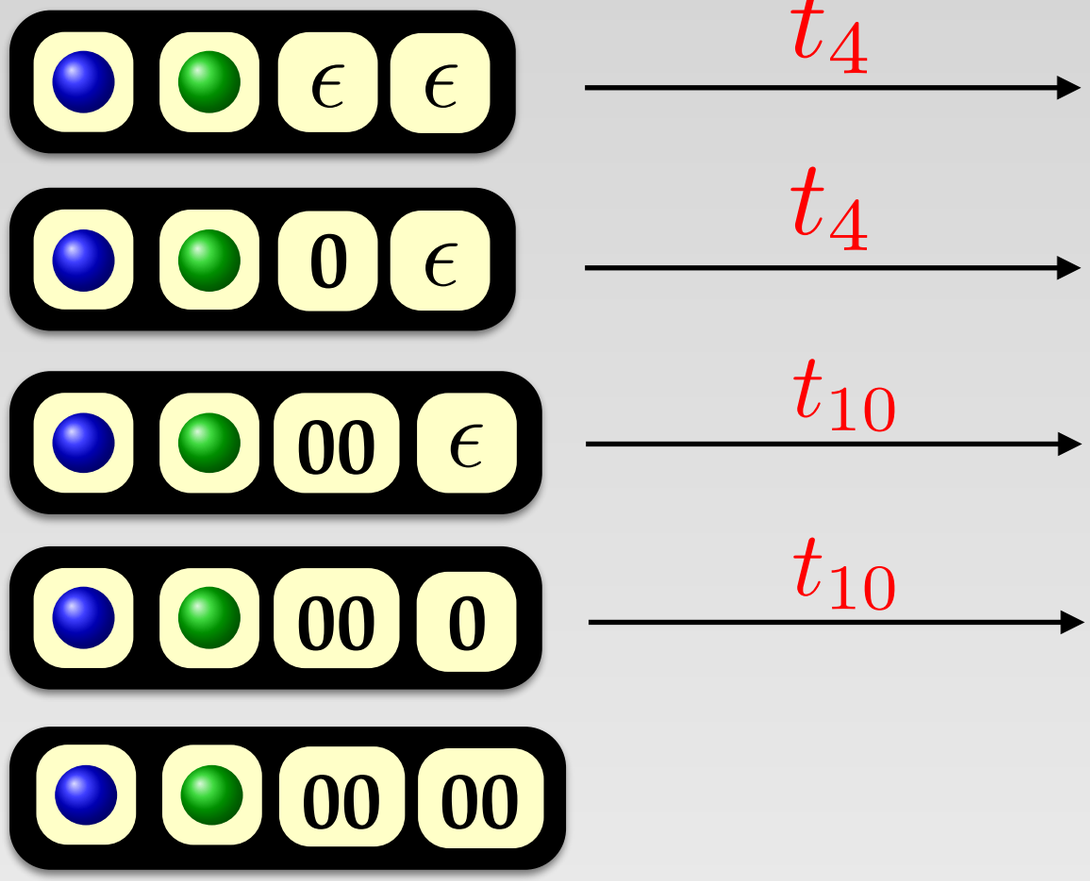


Lossy

Transitions

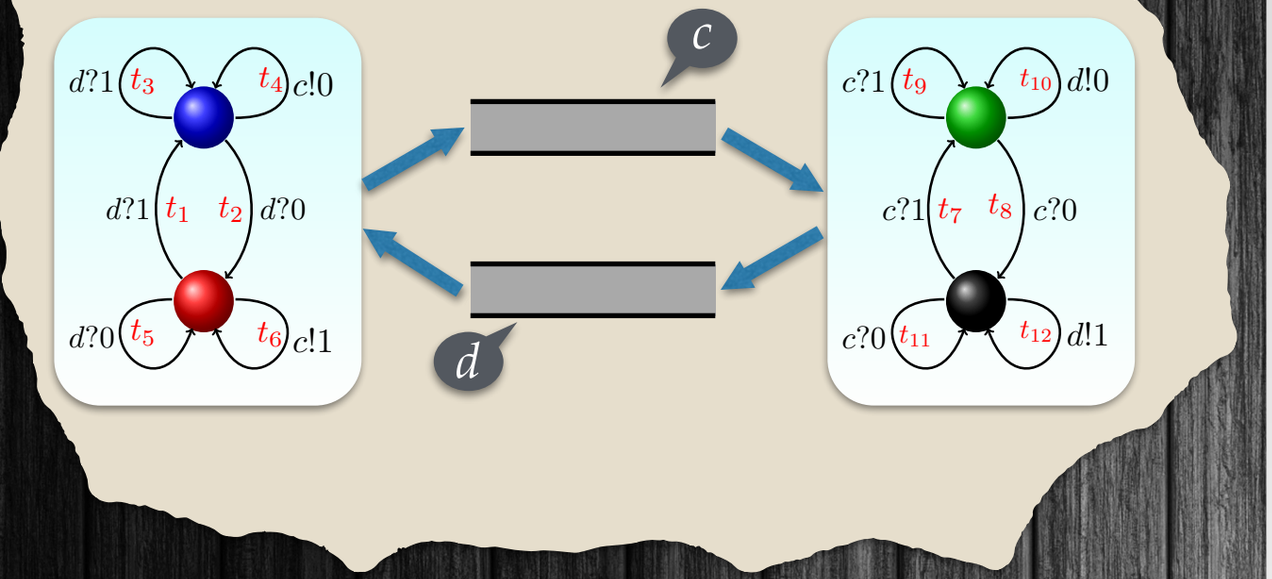


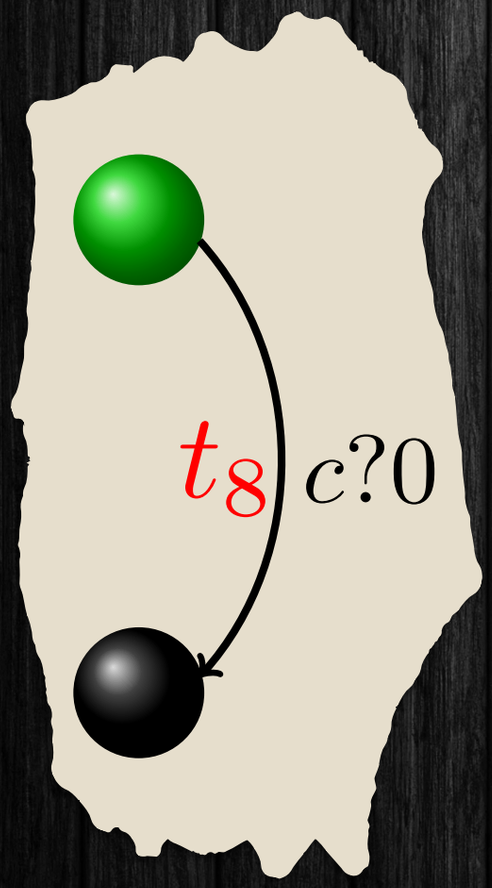


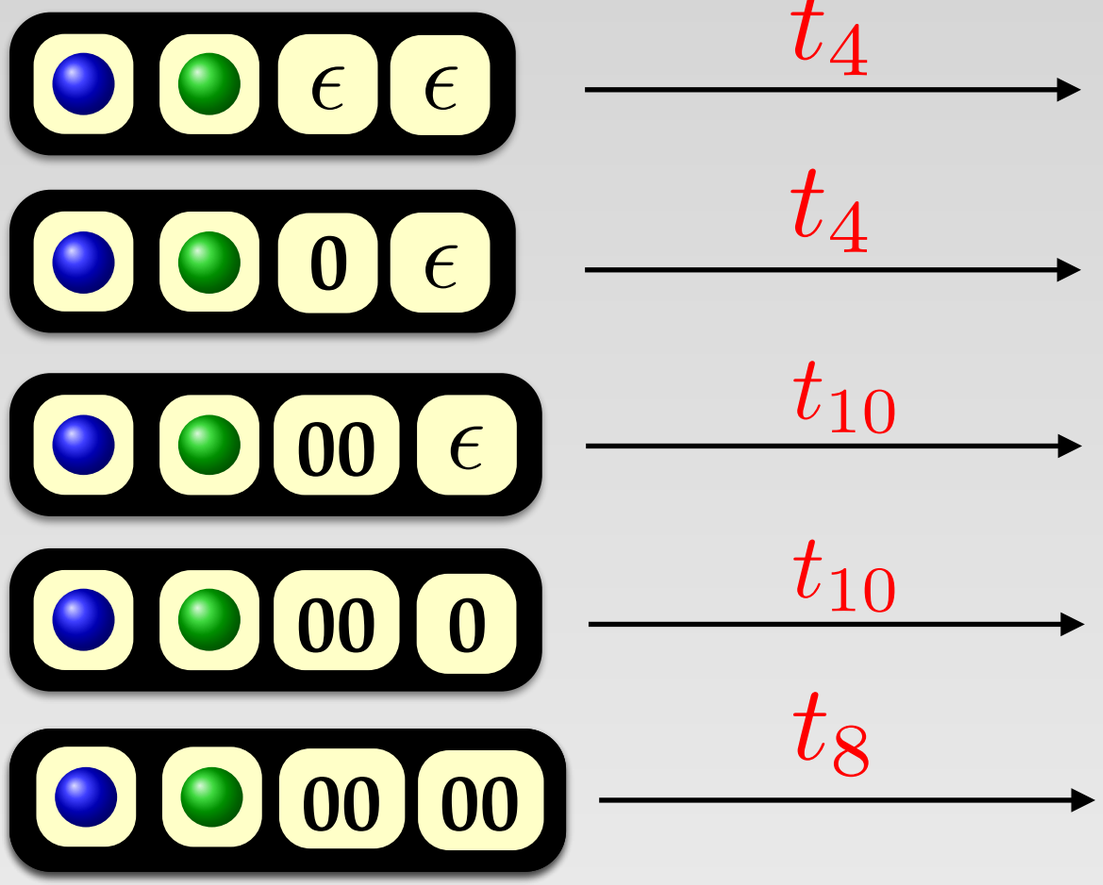


Lossy

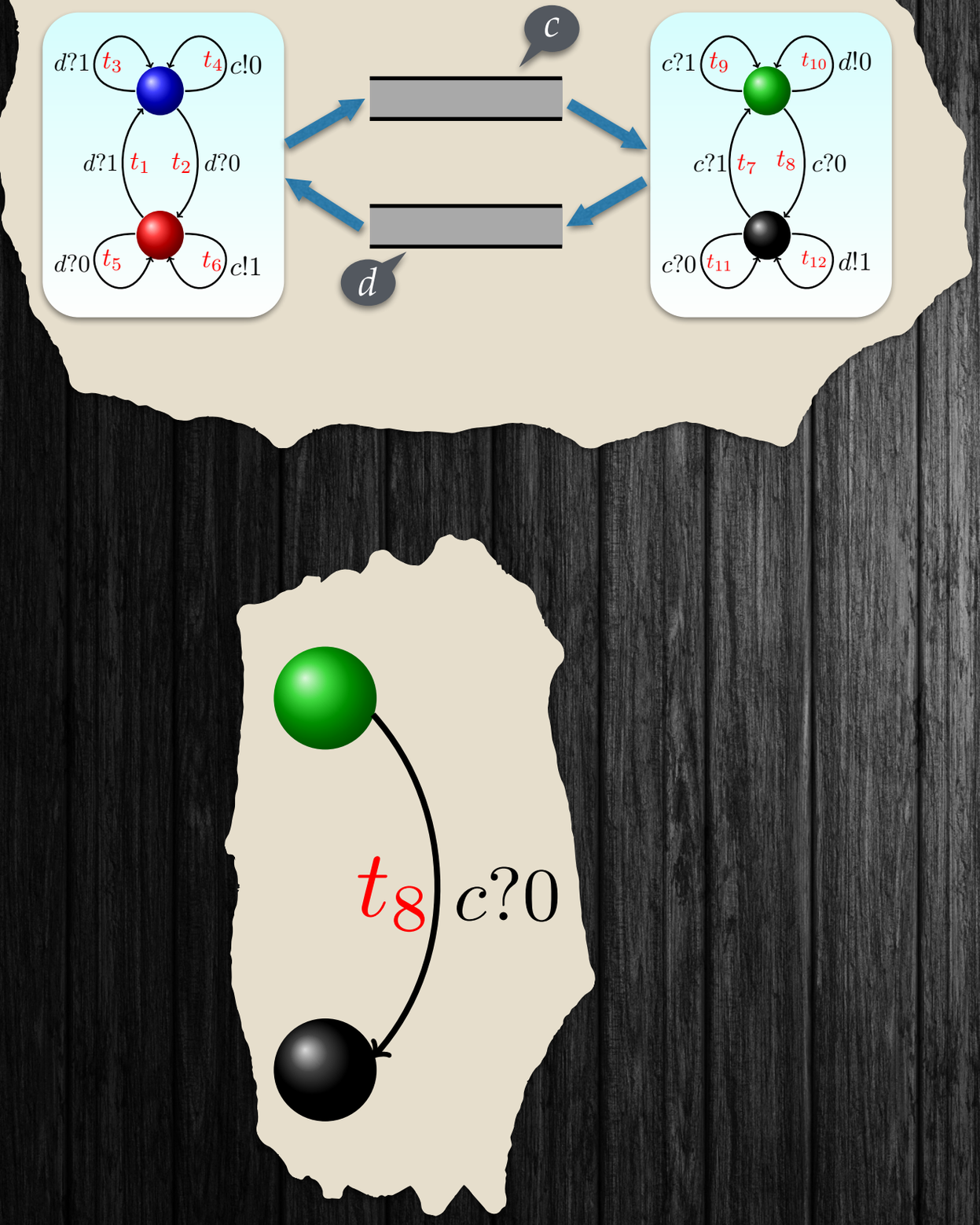
Transitions

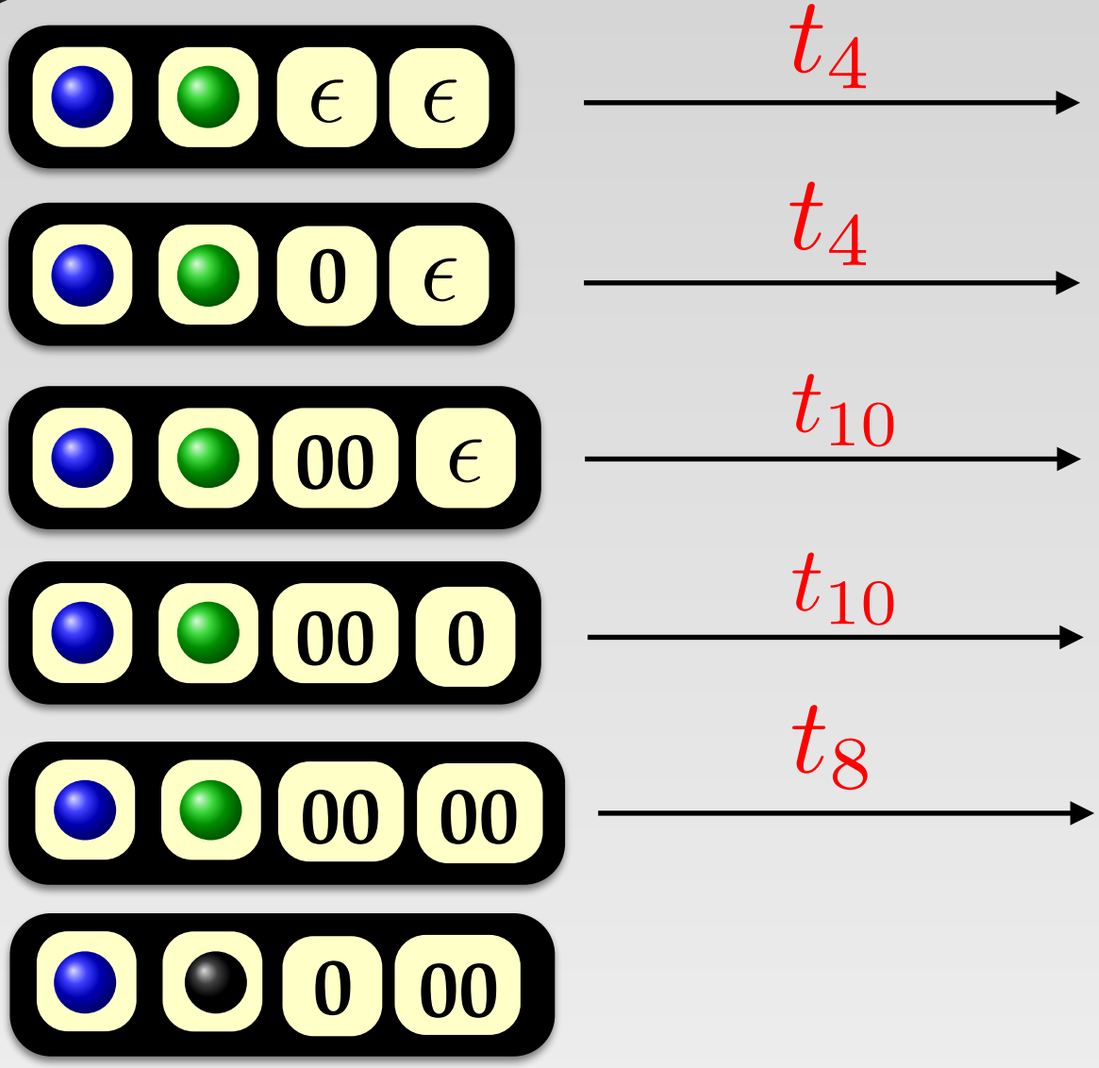




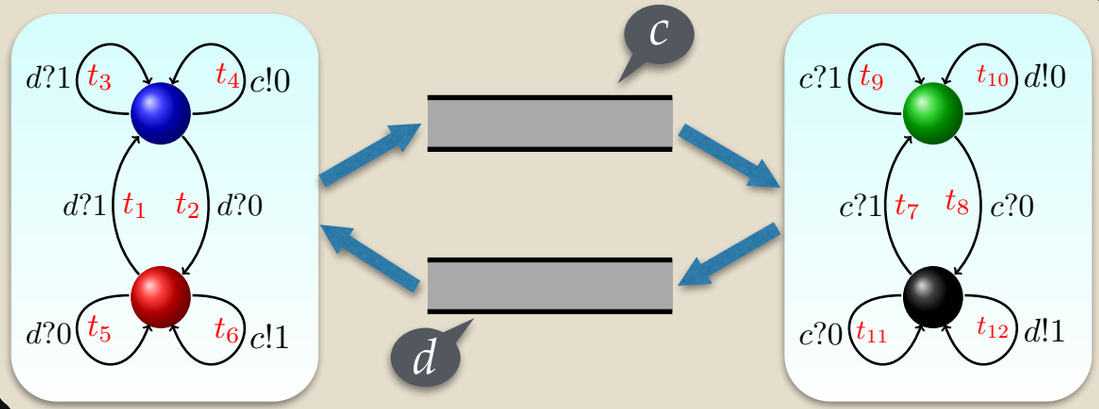


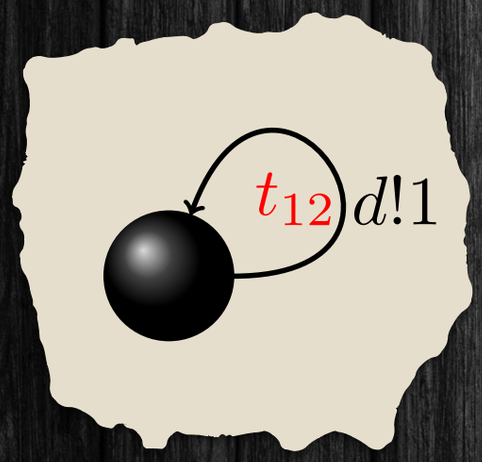
Lossy Transitions

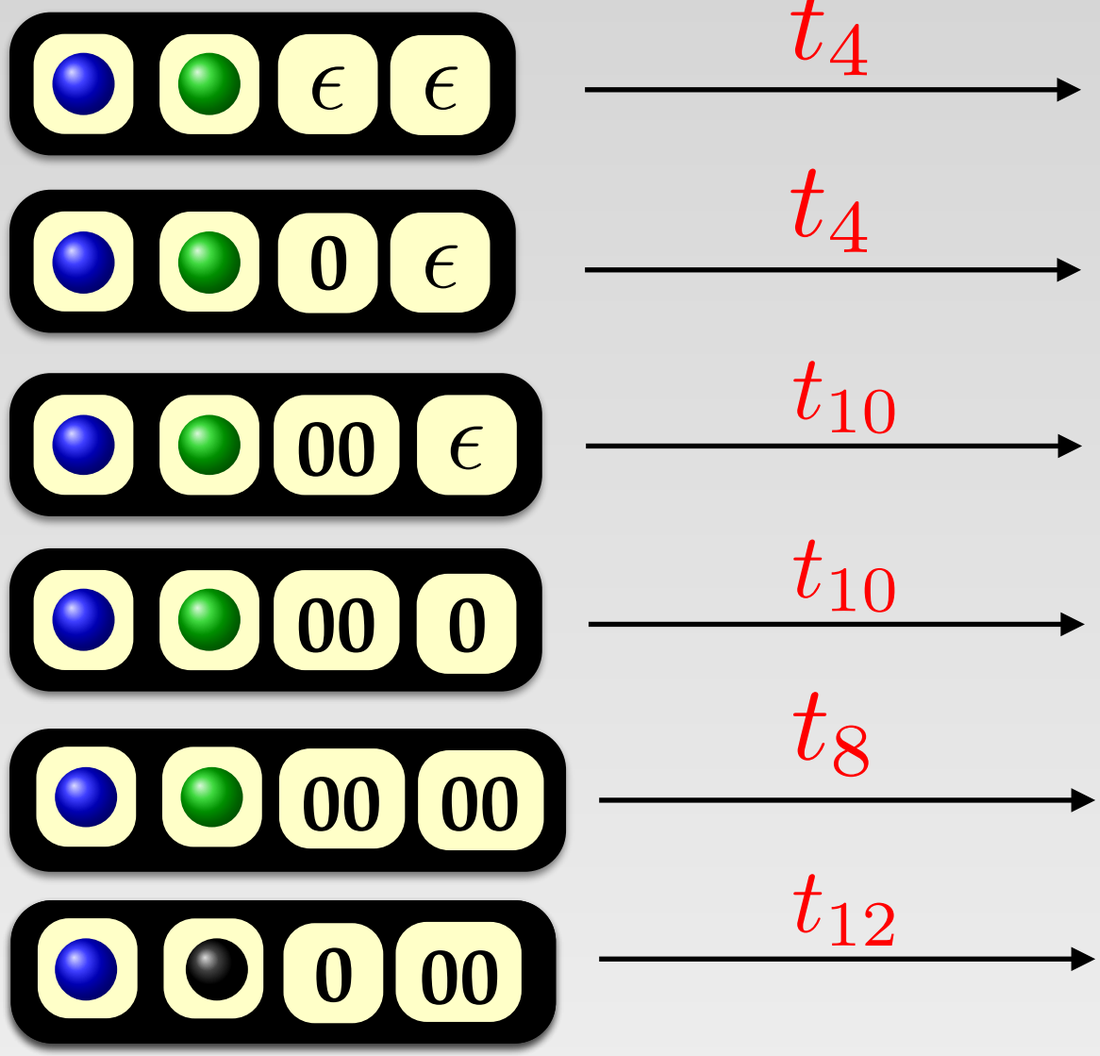




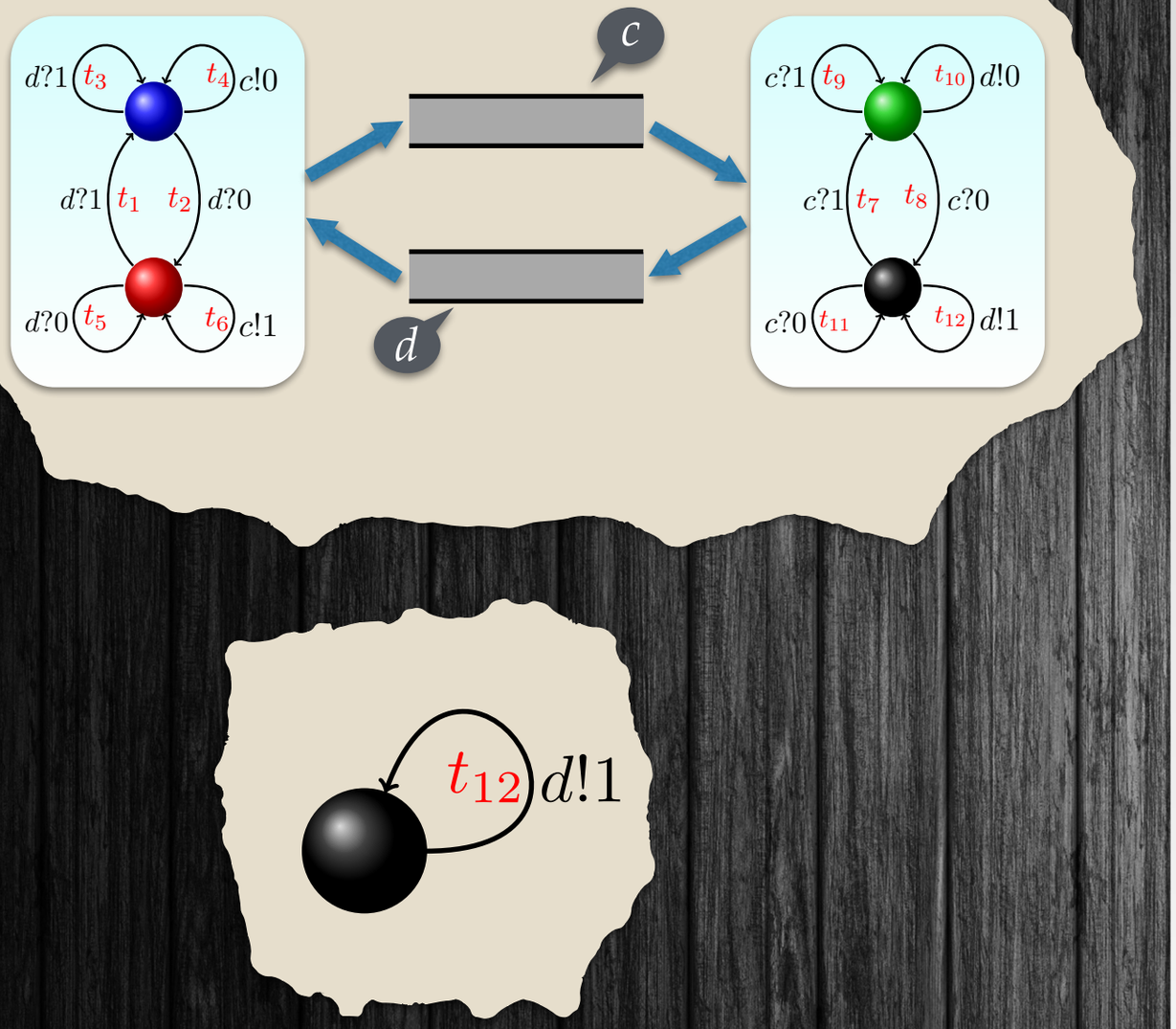
Lossy Transitions

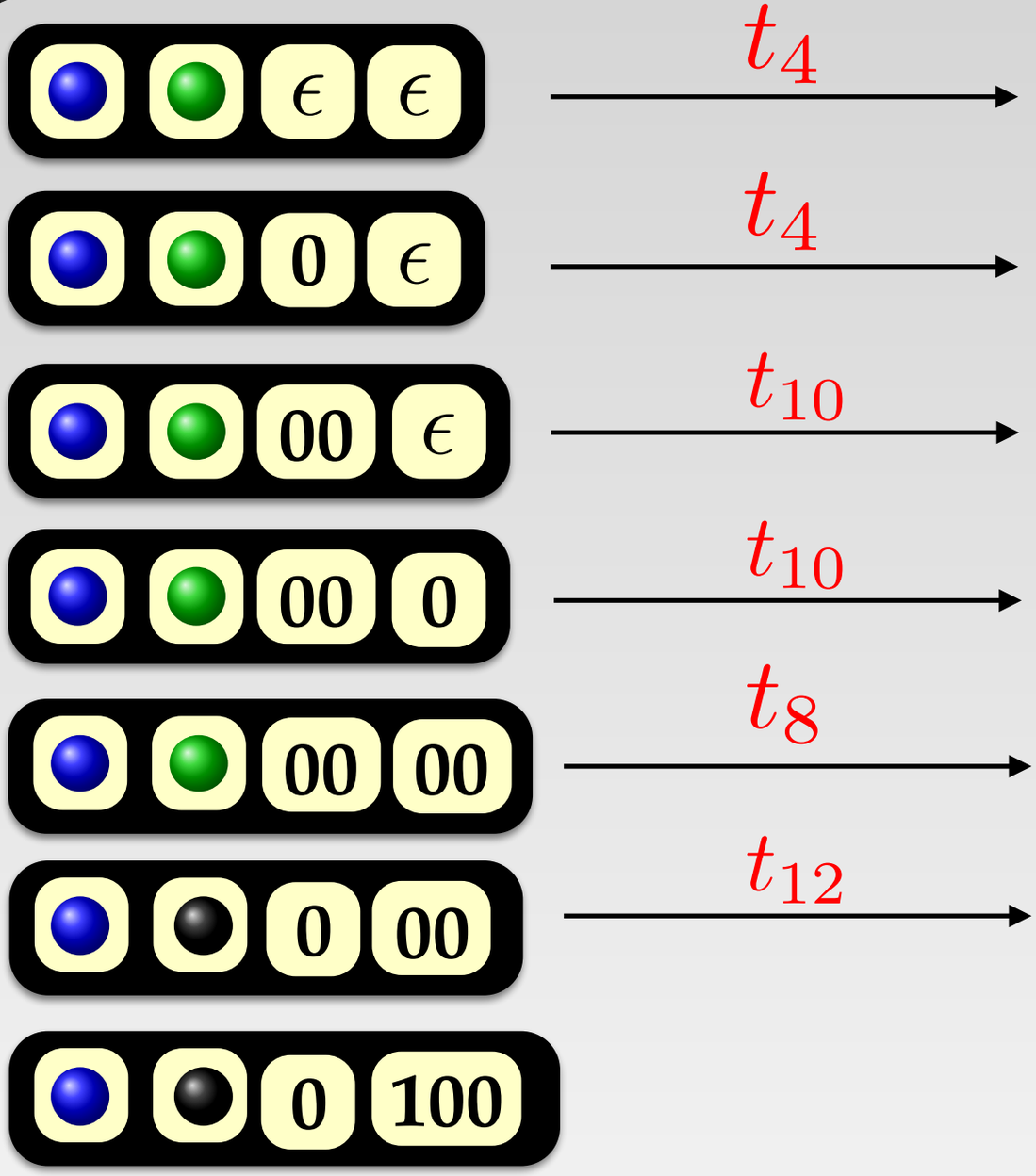






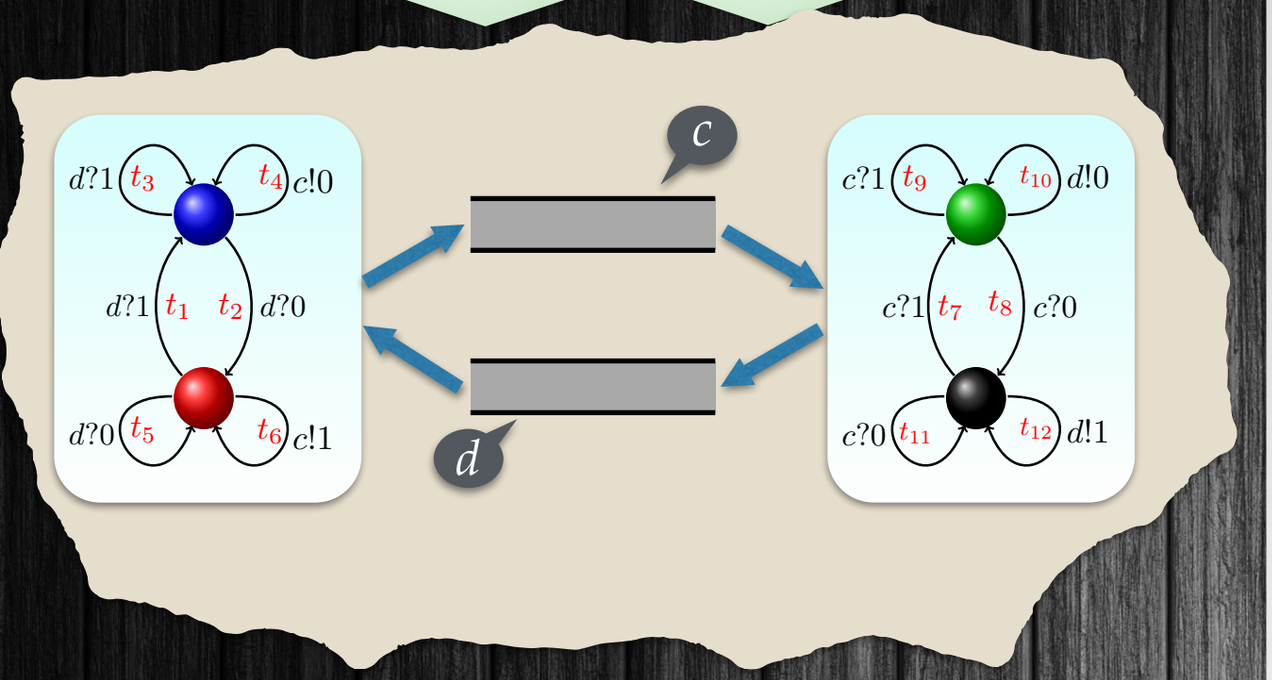
Lossy Transitions

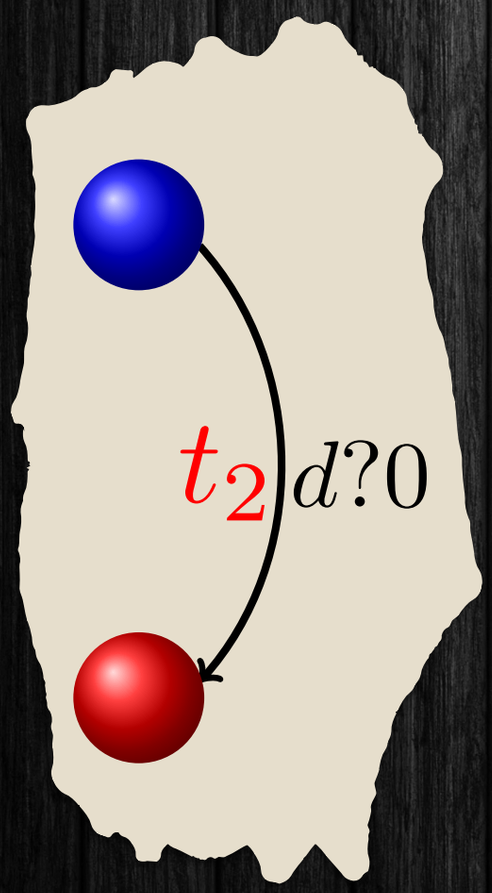


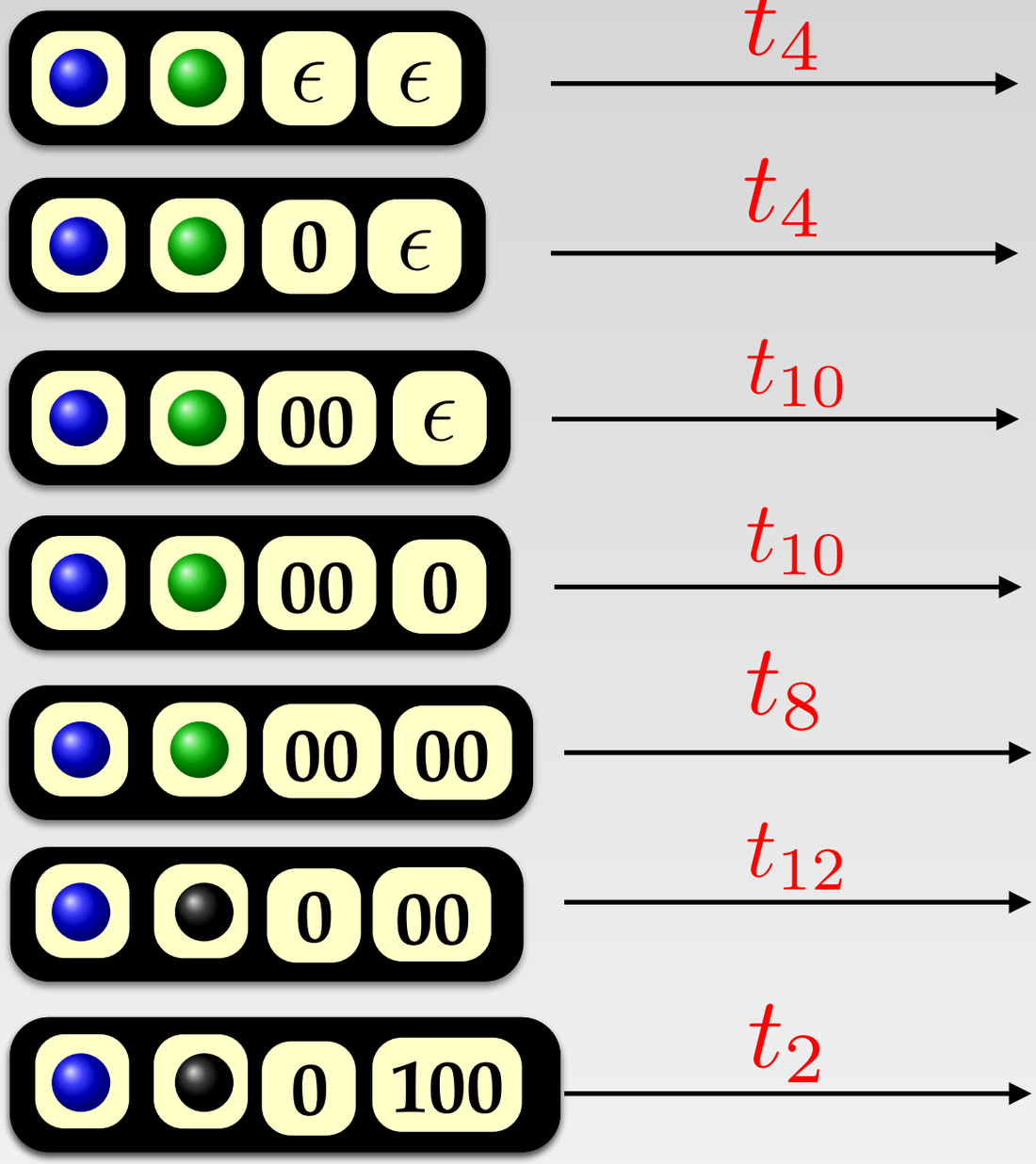


Lossy

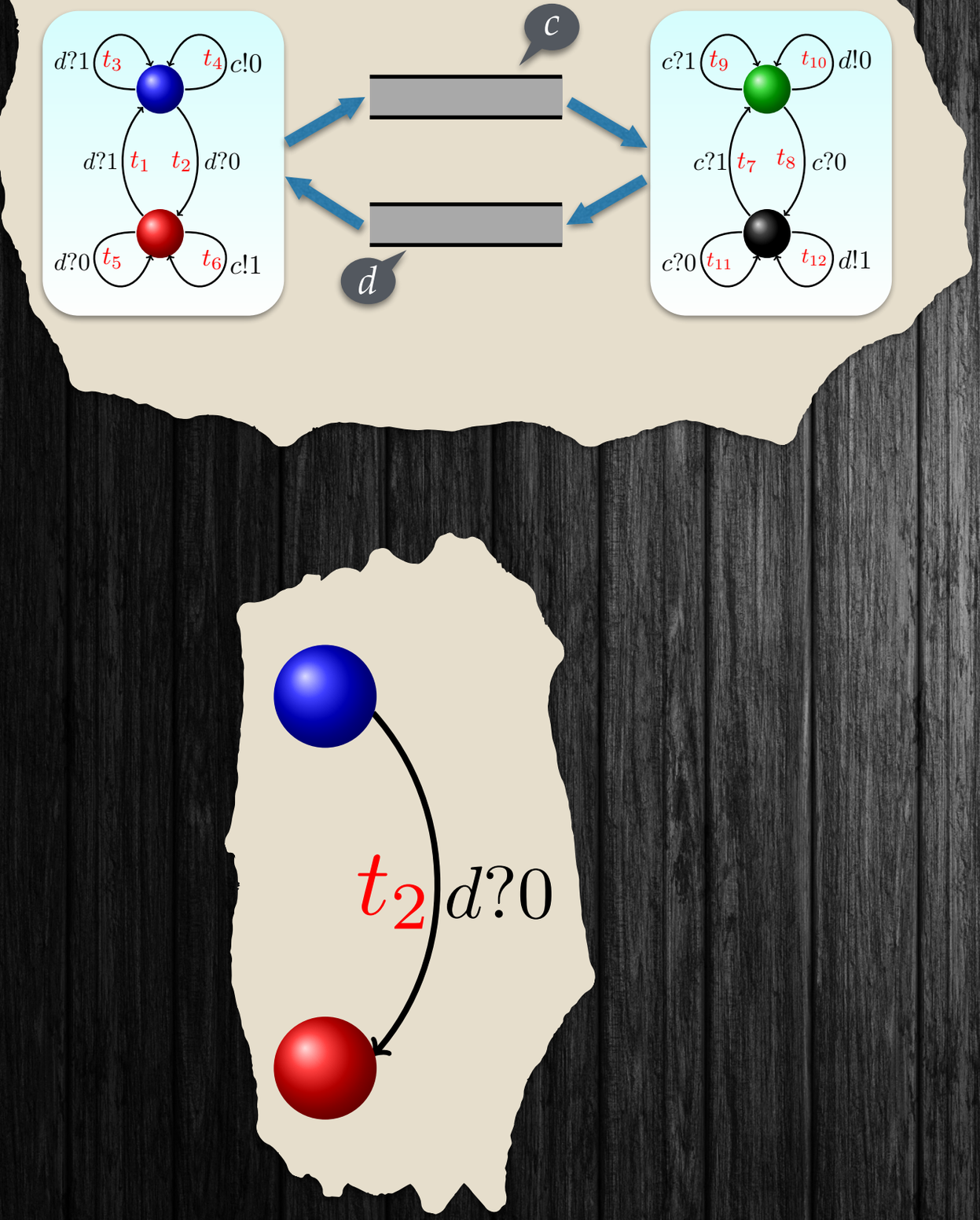
Transitions

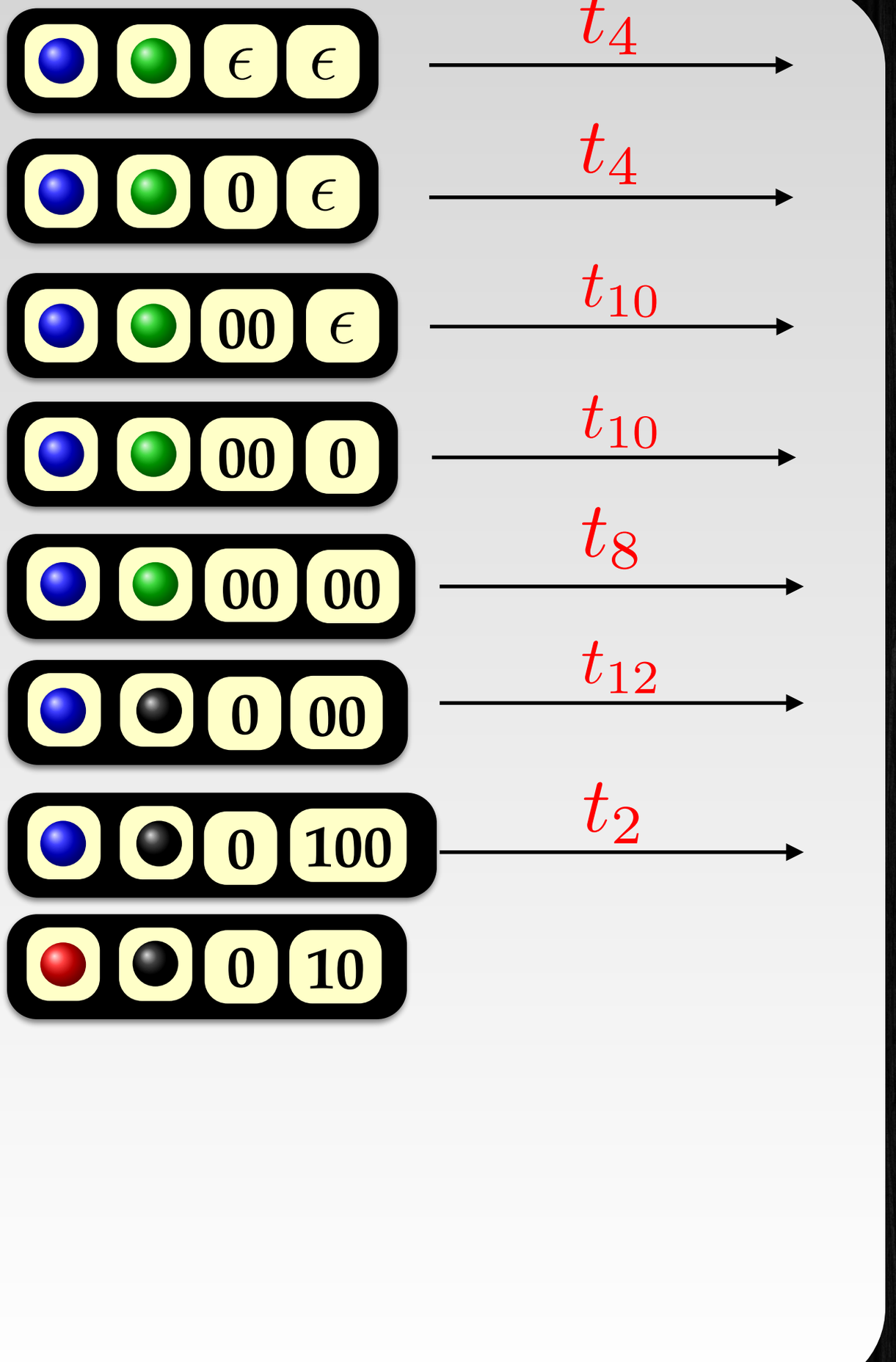




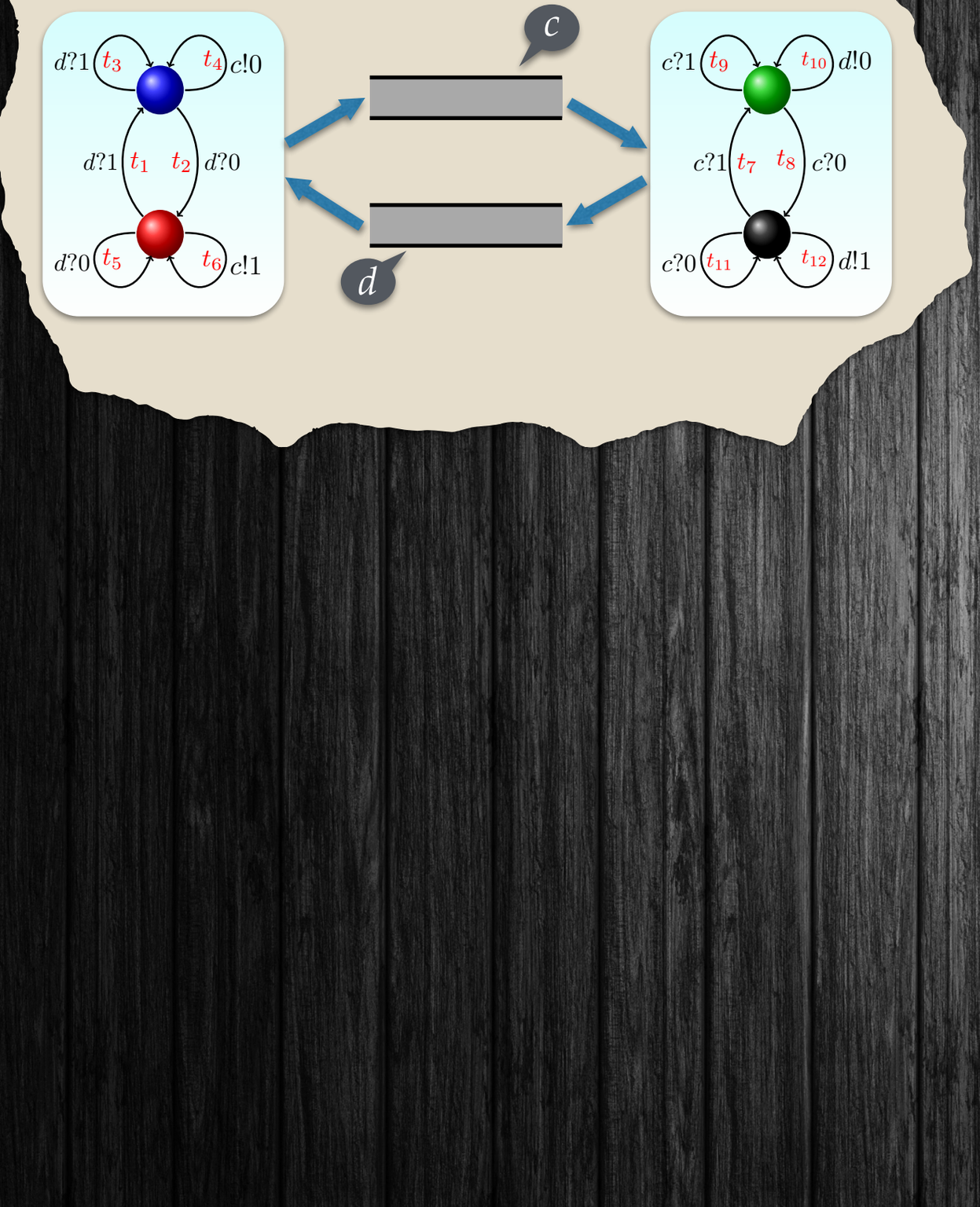


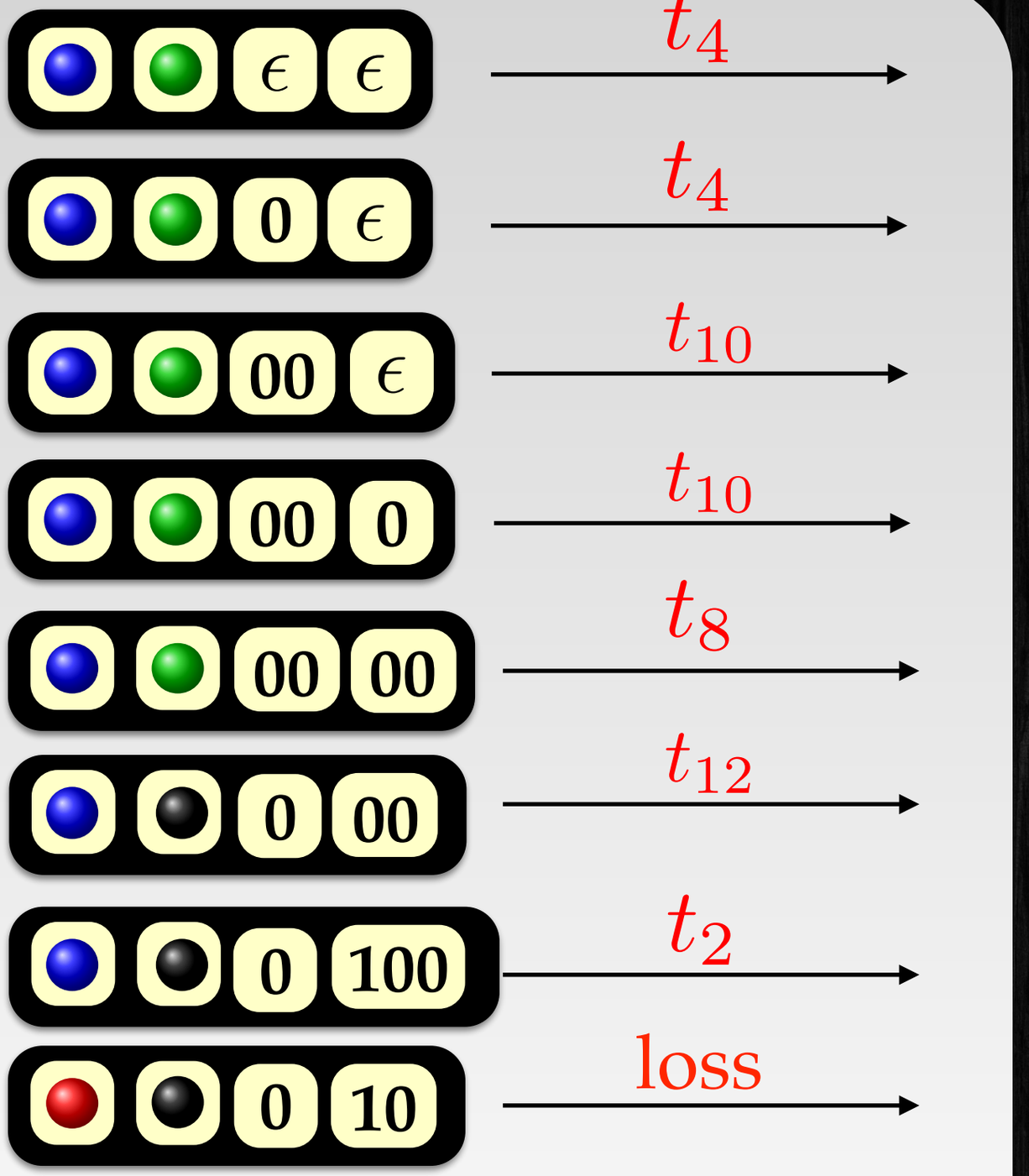
Lossy Transitions



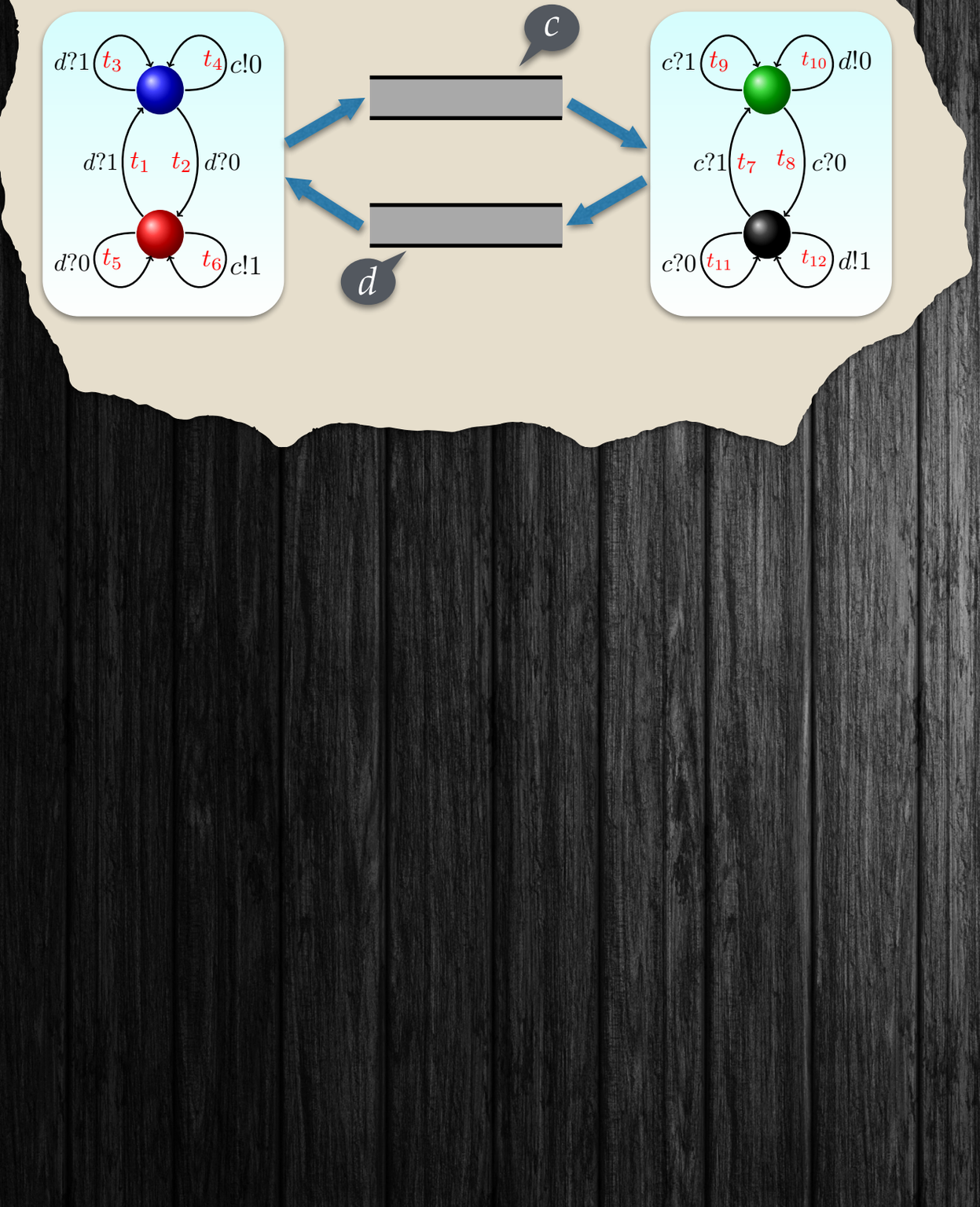


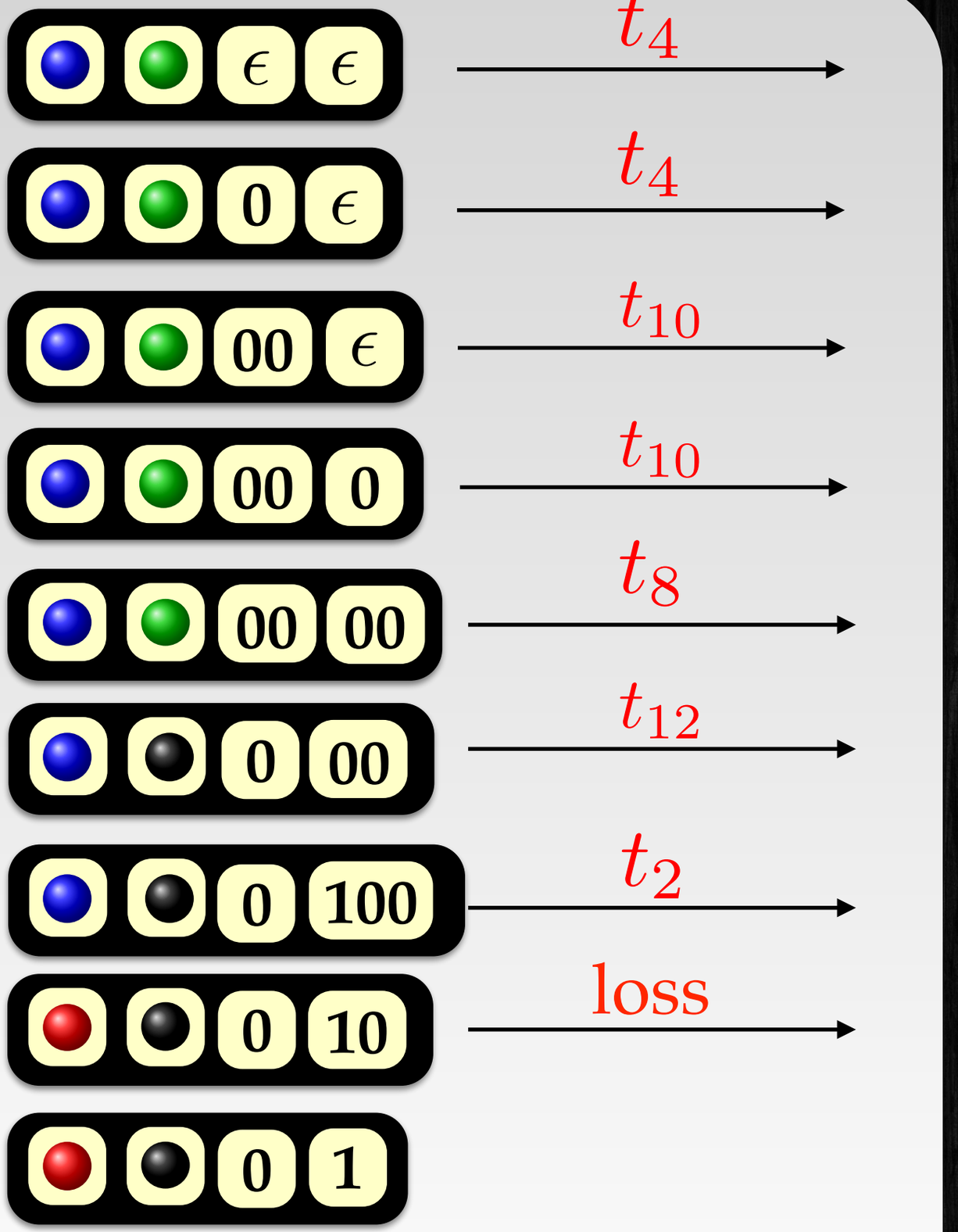
Lossy Transitions





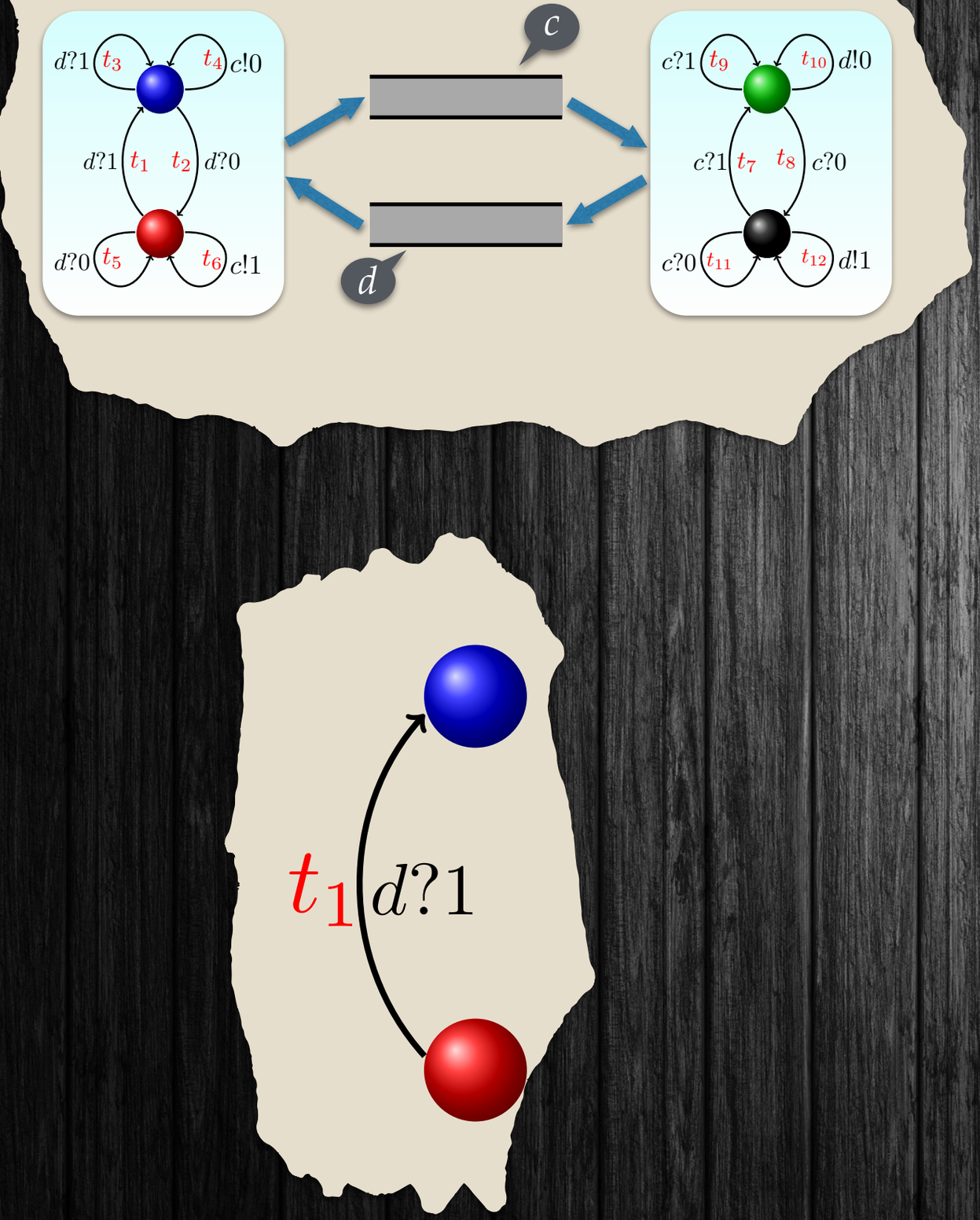
Lossy Transitions

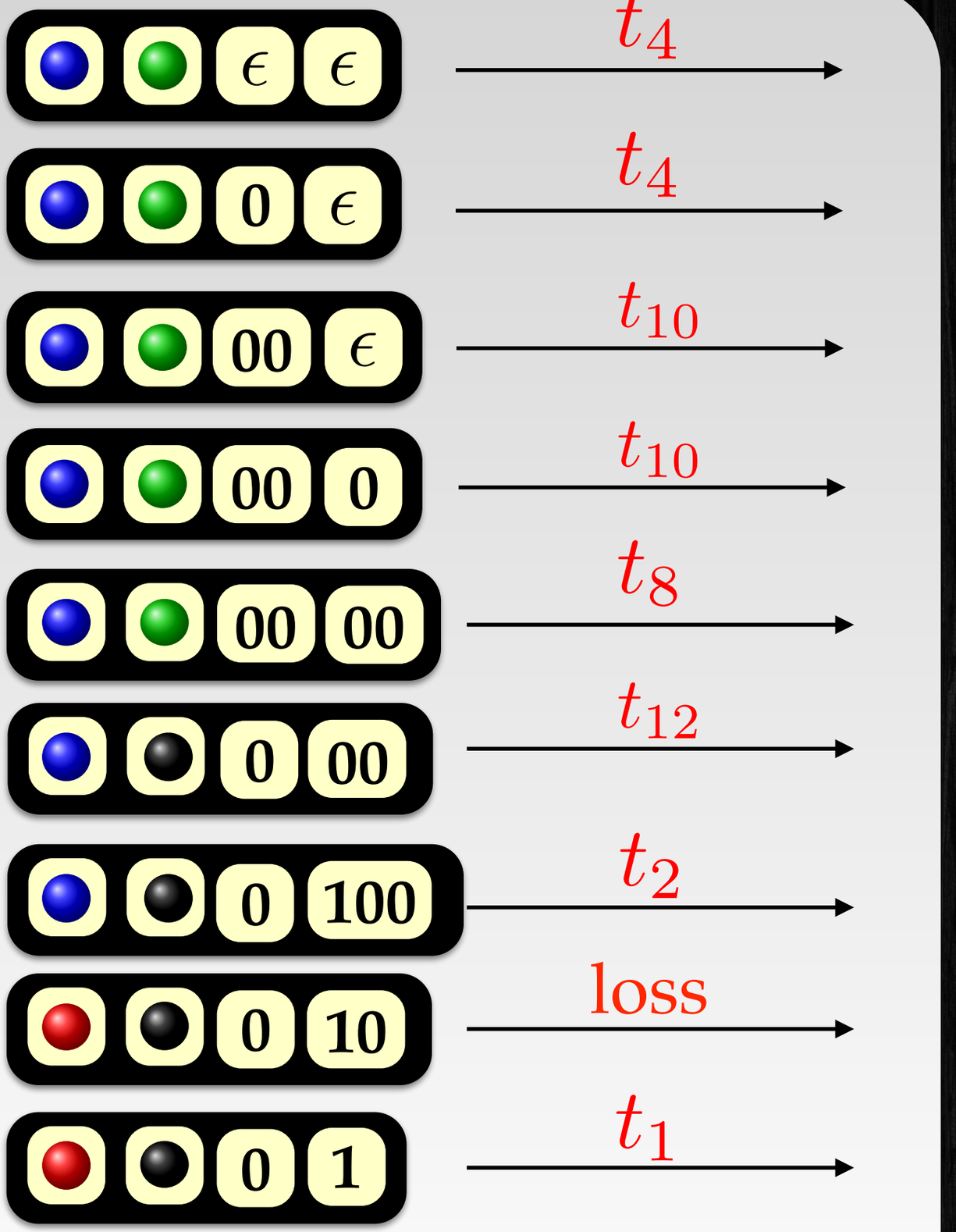




Lossy

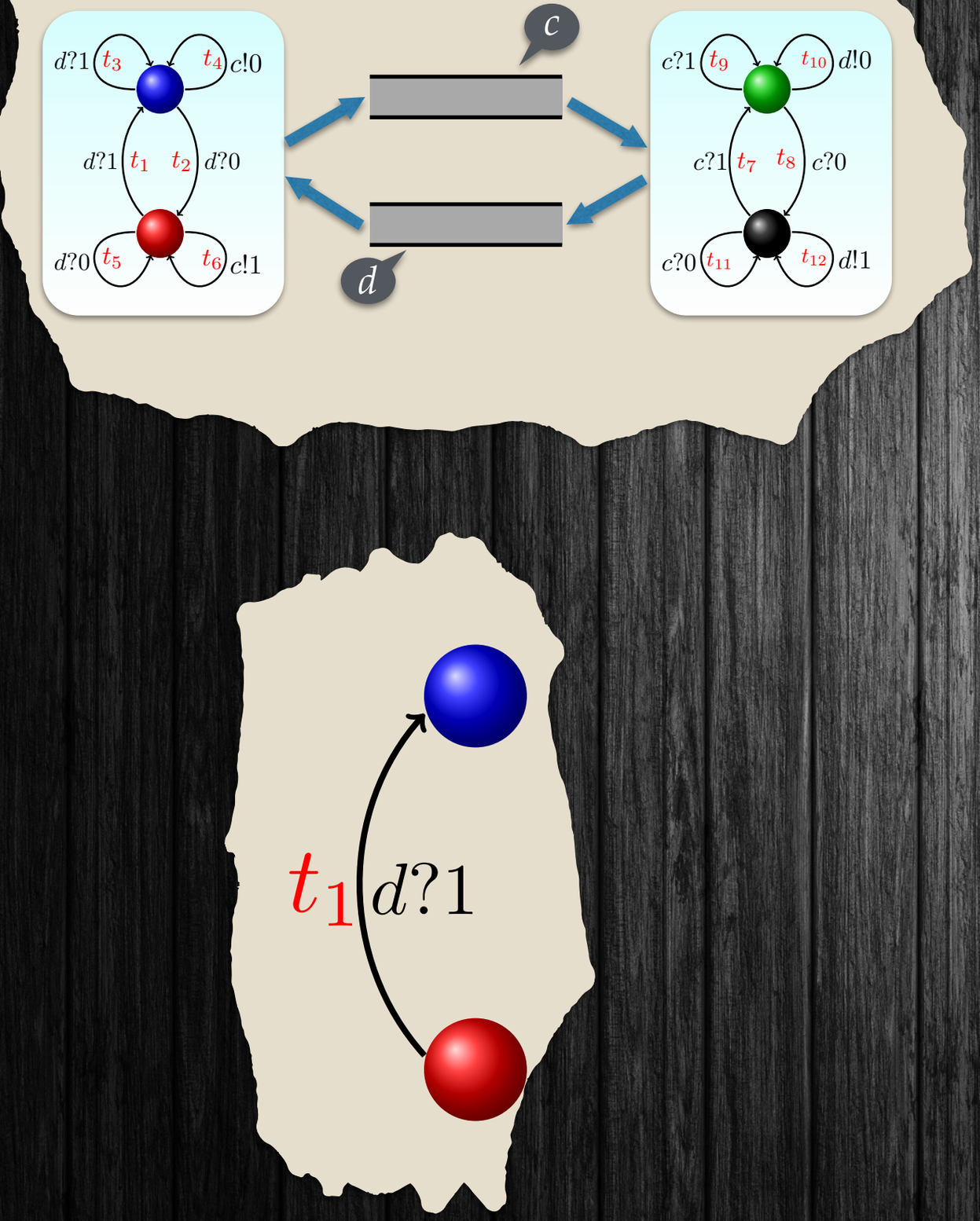
Transitions

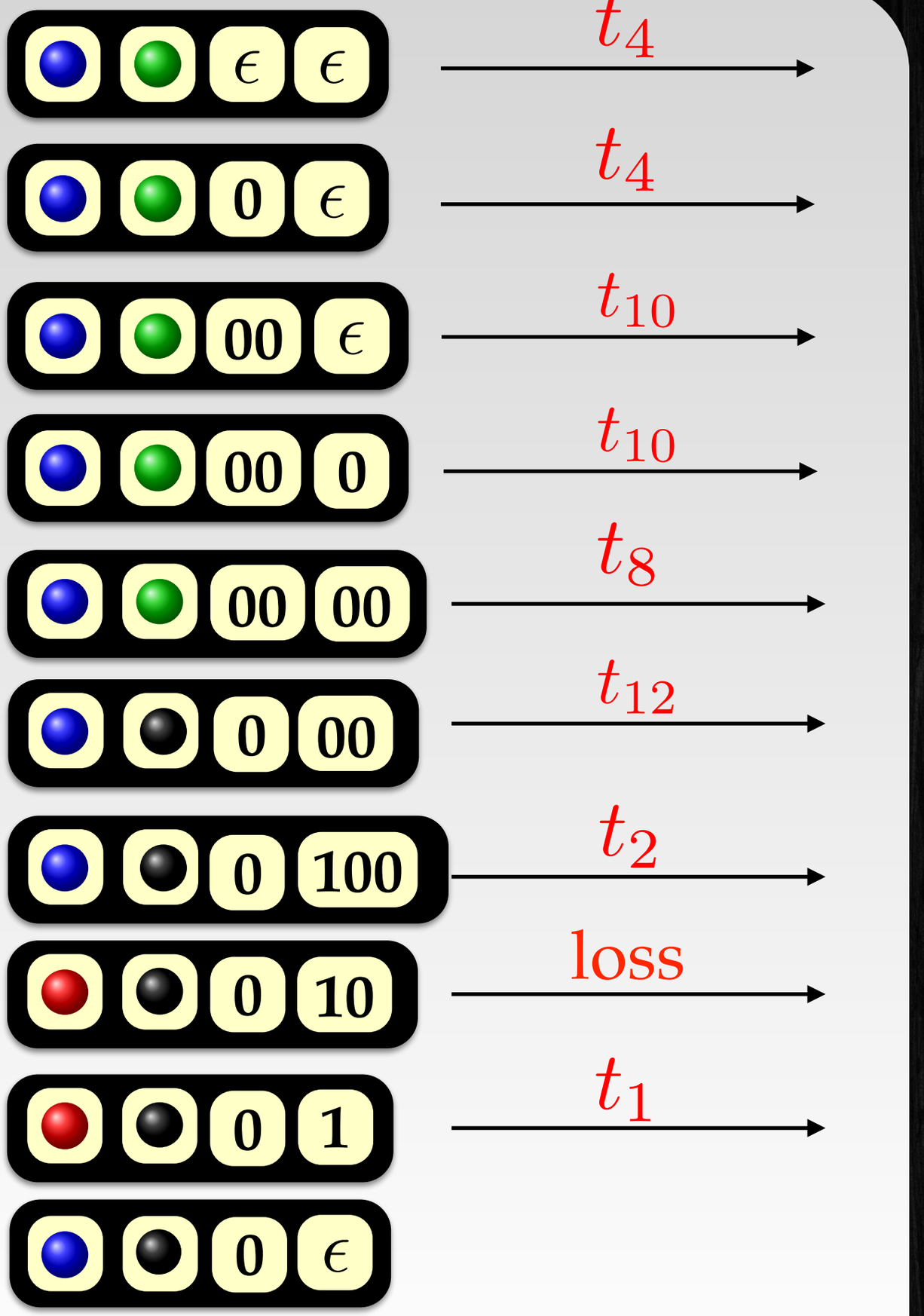




Lossy

Transitions





Lossy Channel Systems

Model ✓

Configurations ✓

Transitions ✓

Ordering

Monotoncity

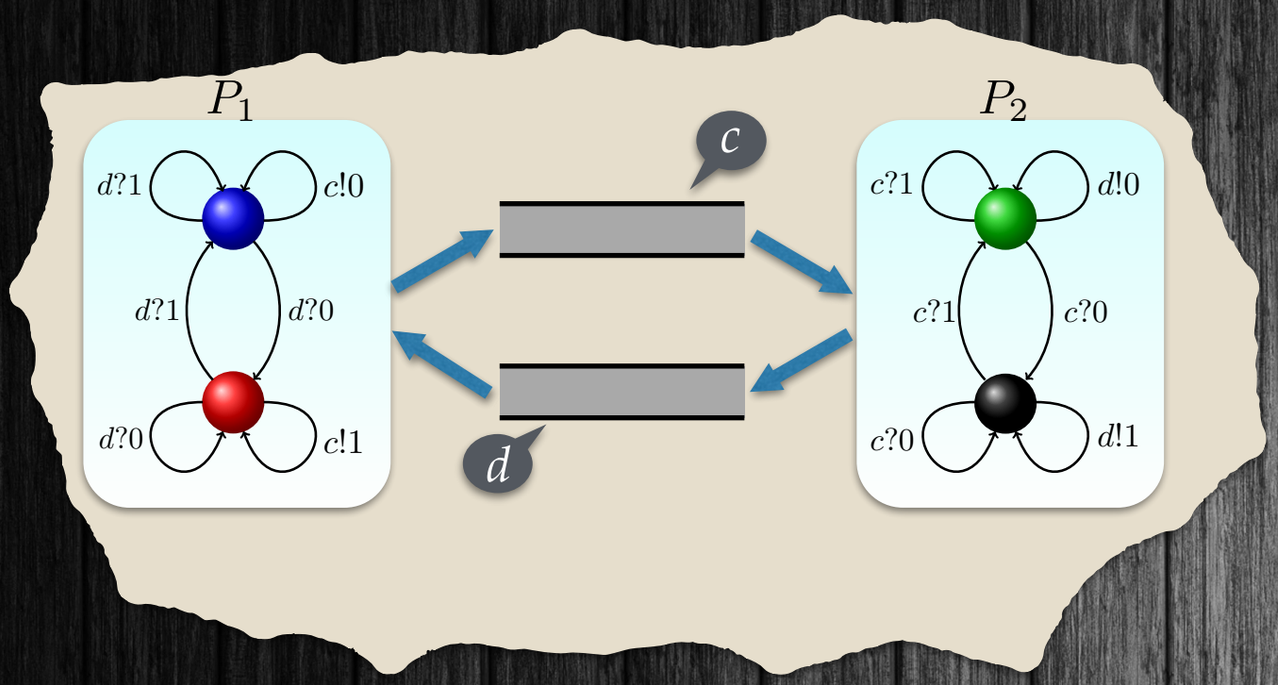
Upward Closed Sets

Computing Predecessors

Backward Reachability

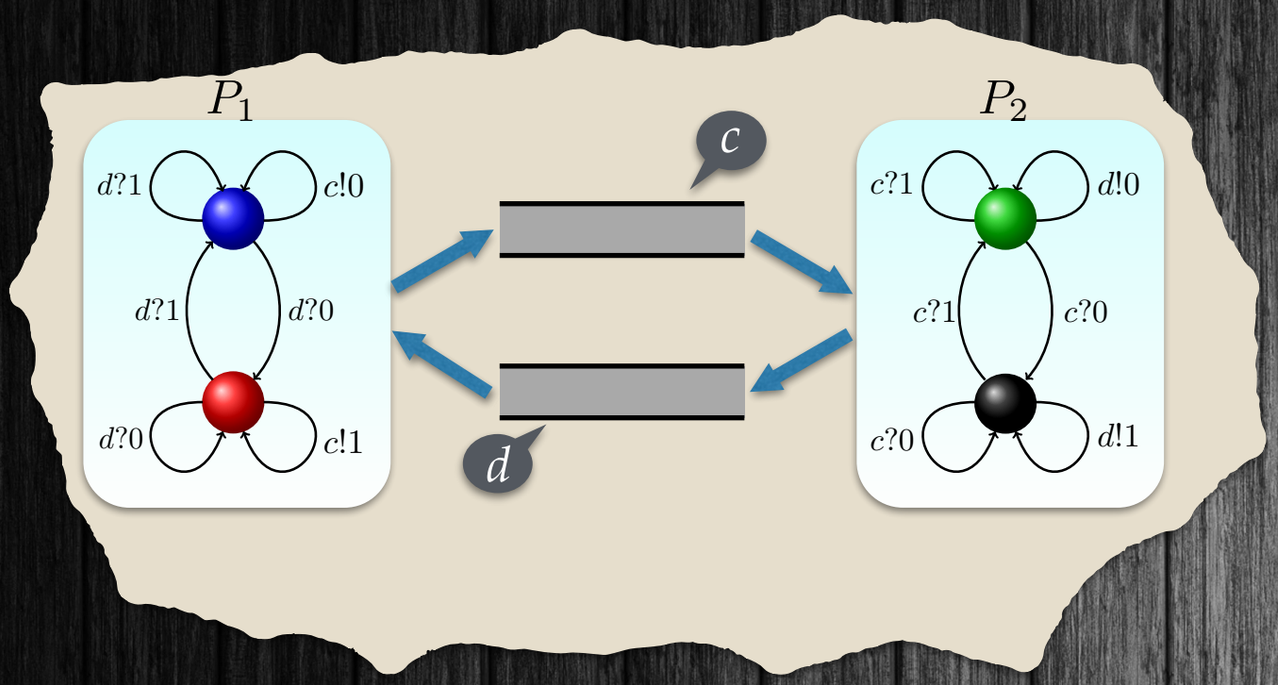
Lossy

Ordering



Lossy

Ordering

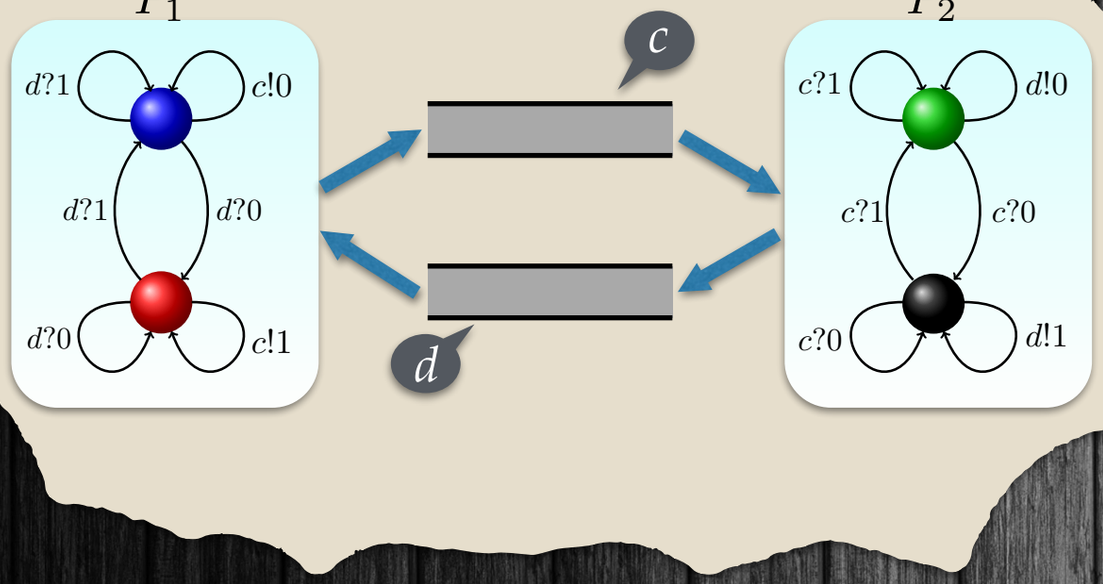


Subword Relation

$ab \sqsubseteq xaycz$

Lossy

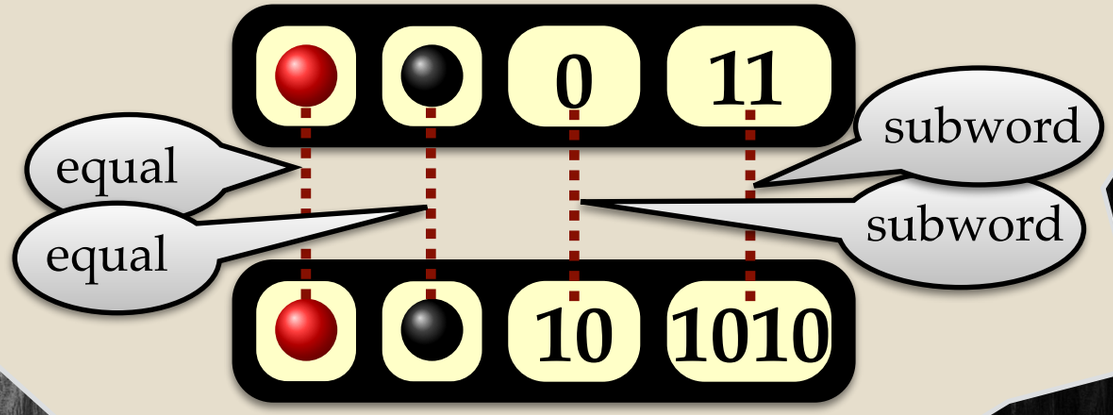
Ordering



Subword Relation

ab $\sqsubseteq$ xaycz

Ordering

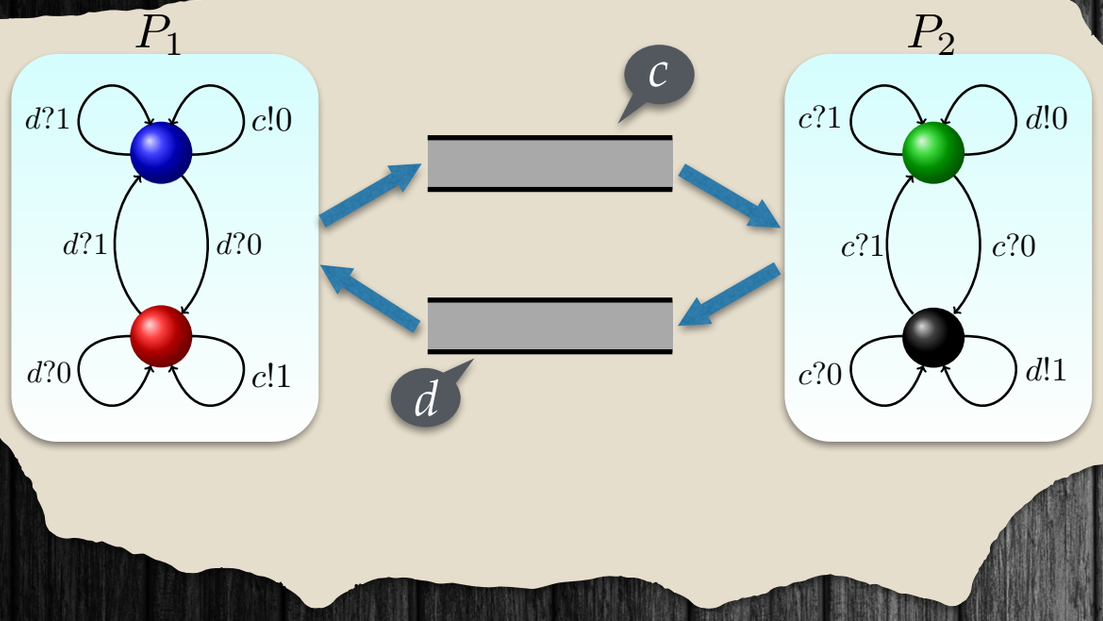


Lossy

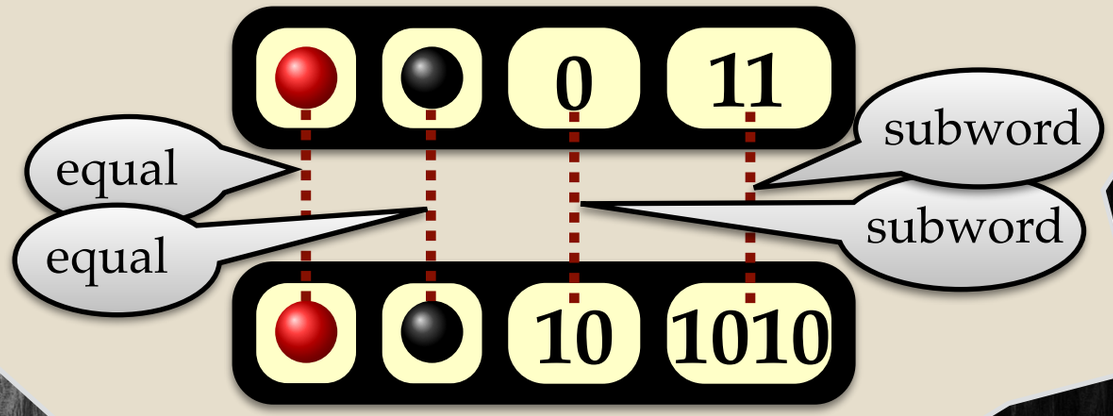
Ordering

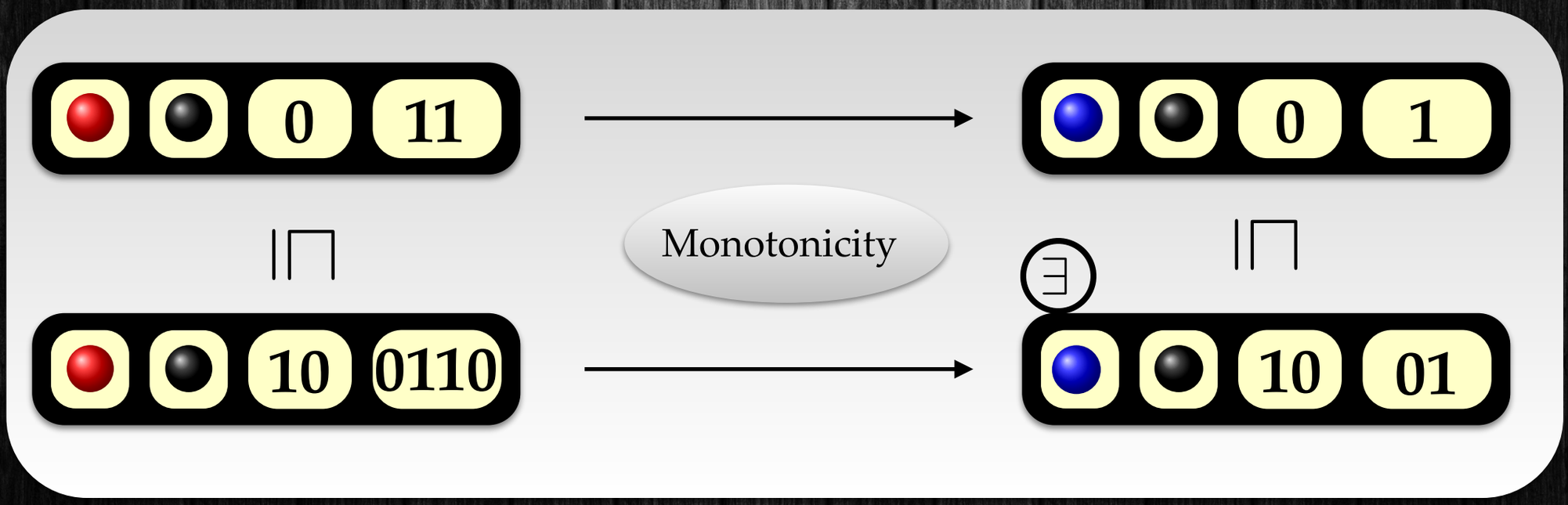
Subword Relation

$ab \sqsubseteq xaycz$



Ordering





Lossy Channel Systems

Model ✓

Configurations ✓

Transitions ✓

Ordering ✓

Monotoncity ✓

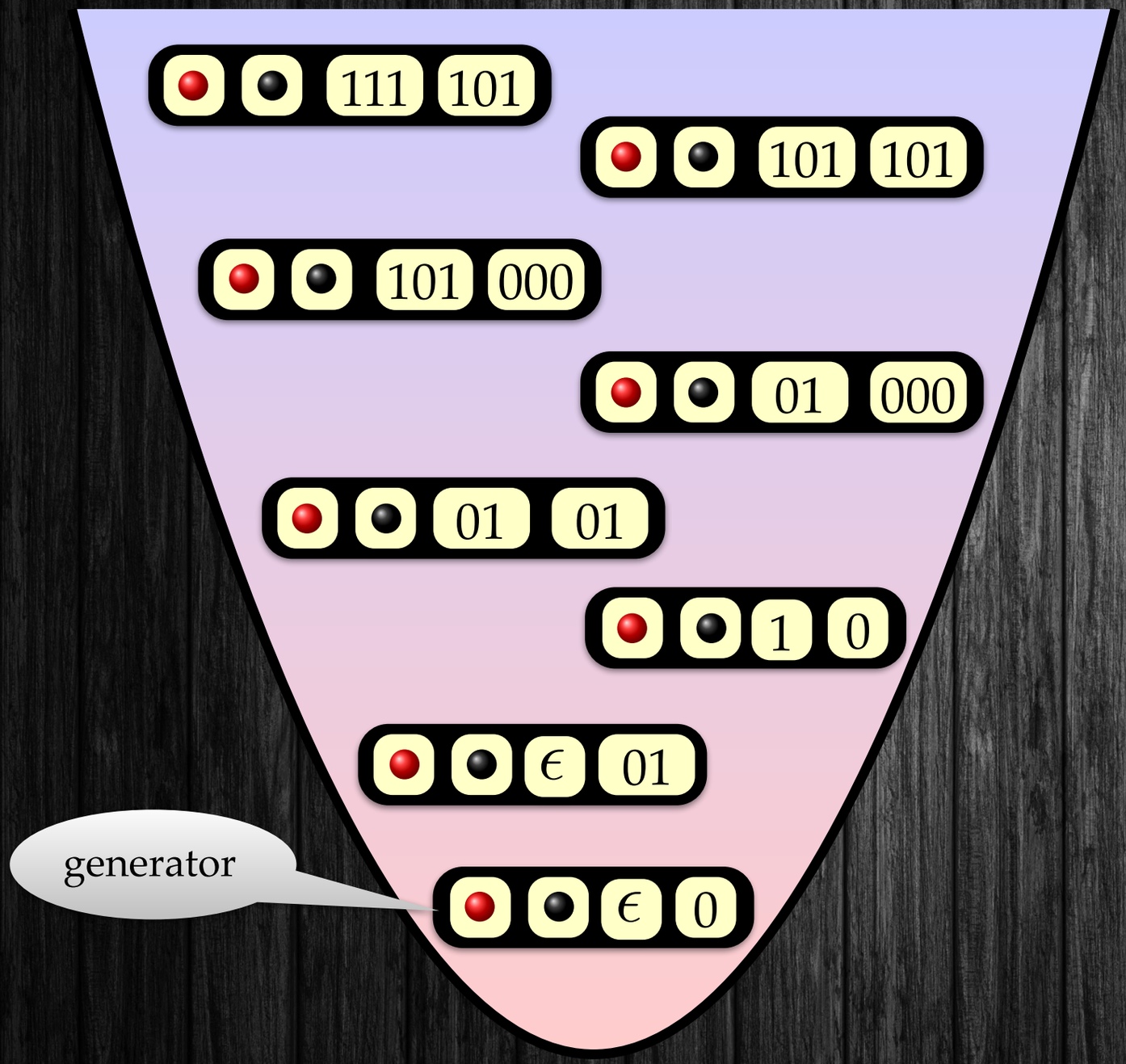
Upward Closed Sets

Computing Predecessors

Backward Reachability

Lossy

Upward-Closed Sets



Lossy Channel Systems

Model

Configurations

Transitions

Ordering

Monotoncity

Upward Closed Sets

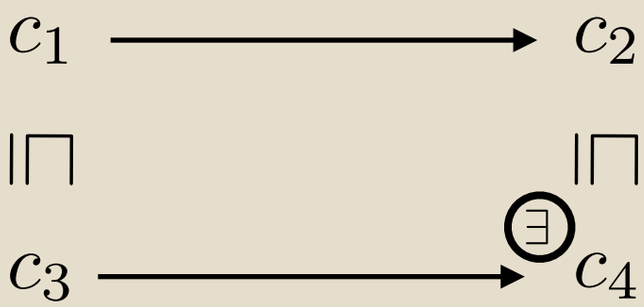
Computing Predecessors

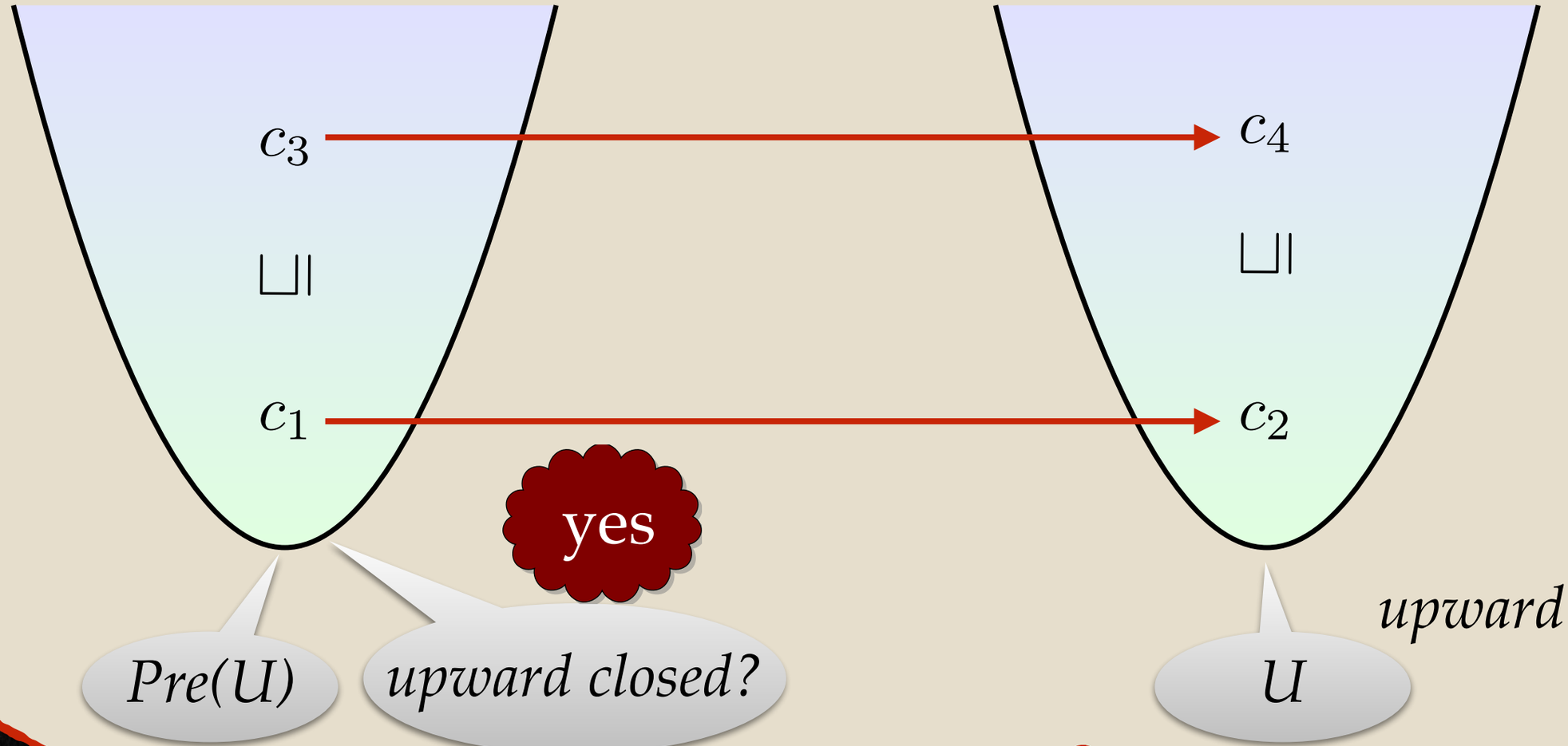
Backward Reachability

Lossy

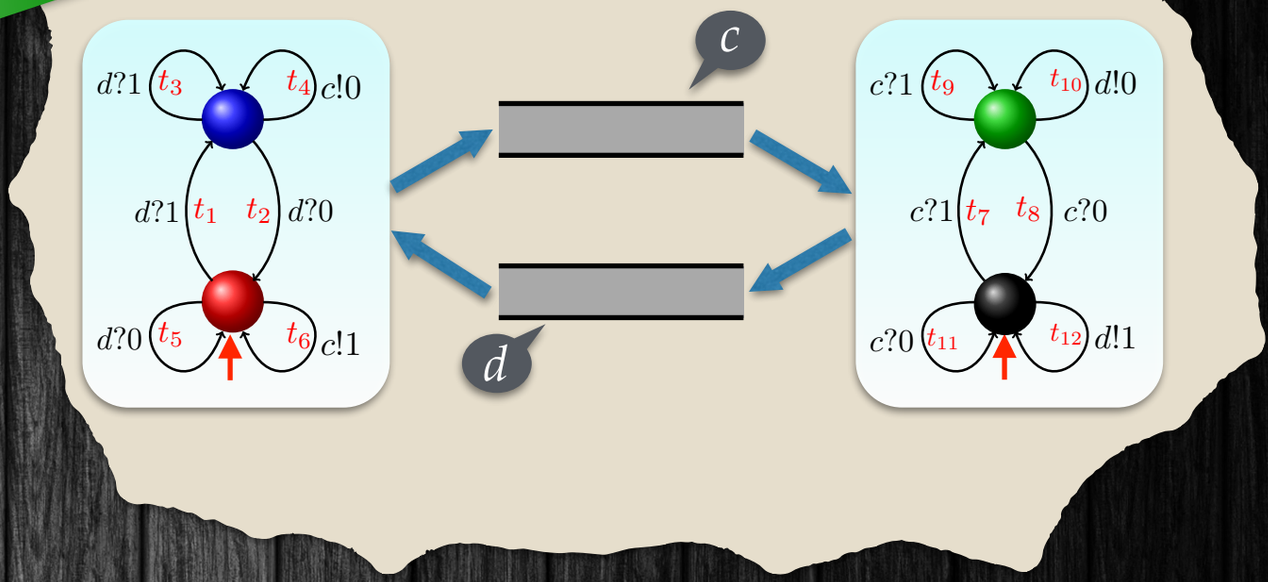
Predecessors

Monotonicity: UC persevered by *Pre*

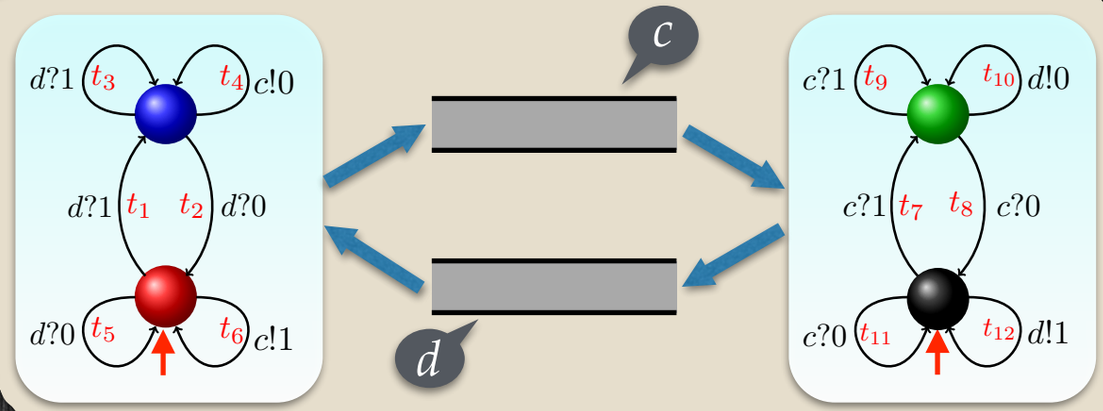




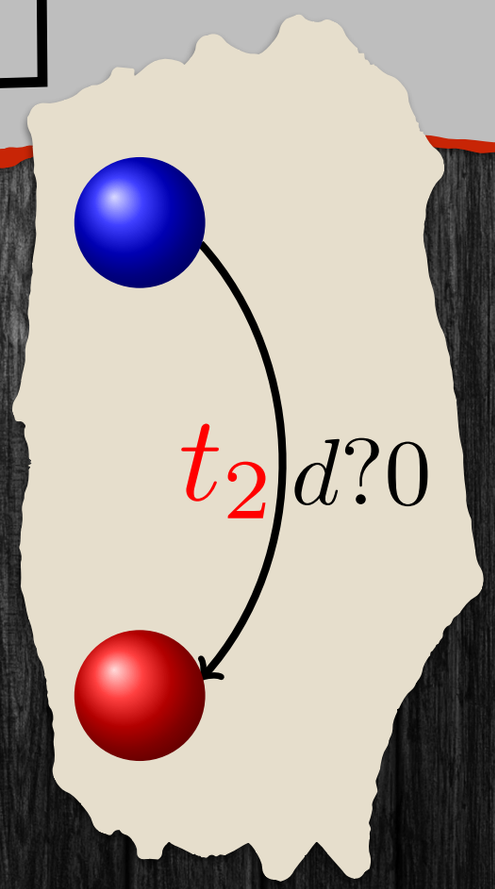
Lossy: Computing Predecessors



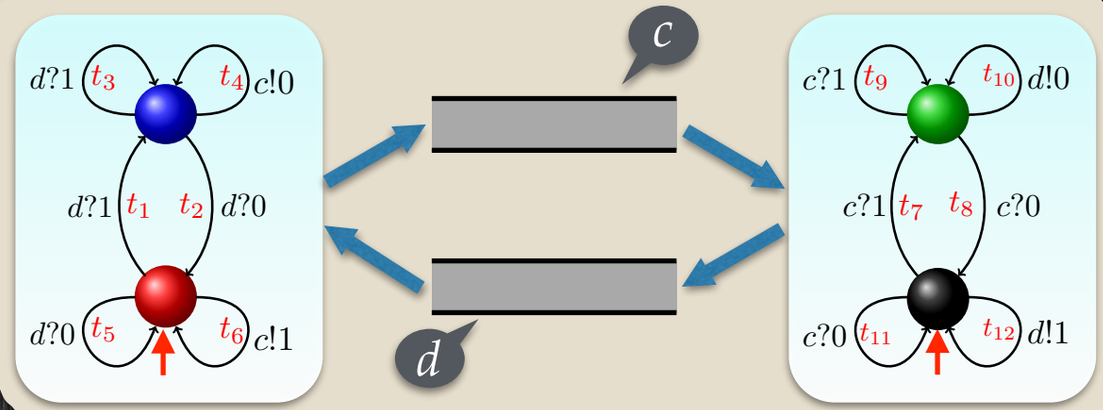
Lossy: Computing Predecessors



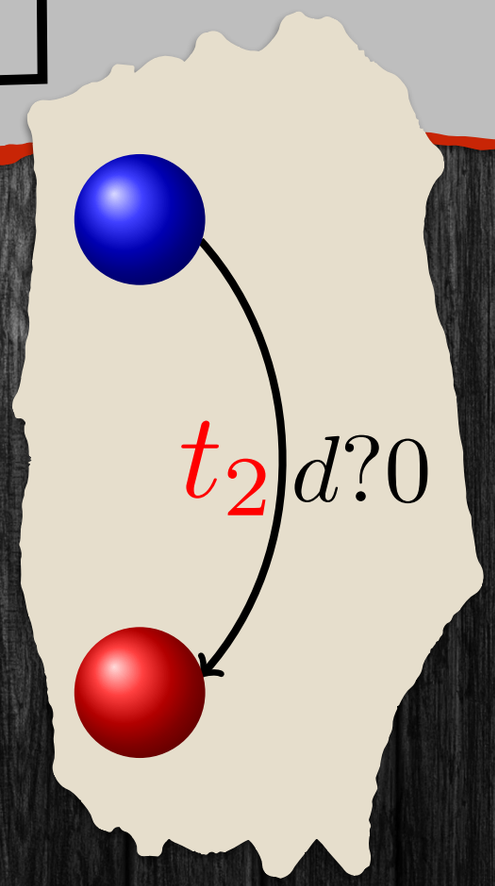
$$Pre_{t_2}\left[\;\epsilon \;\; 01\;\right] =$$



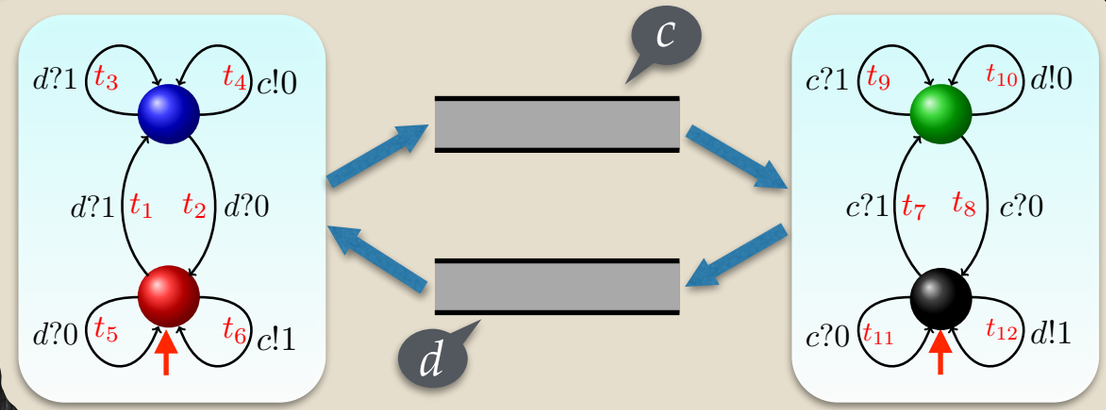
Lossy: Computing Predecessors



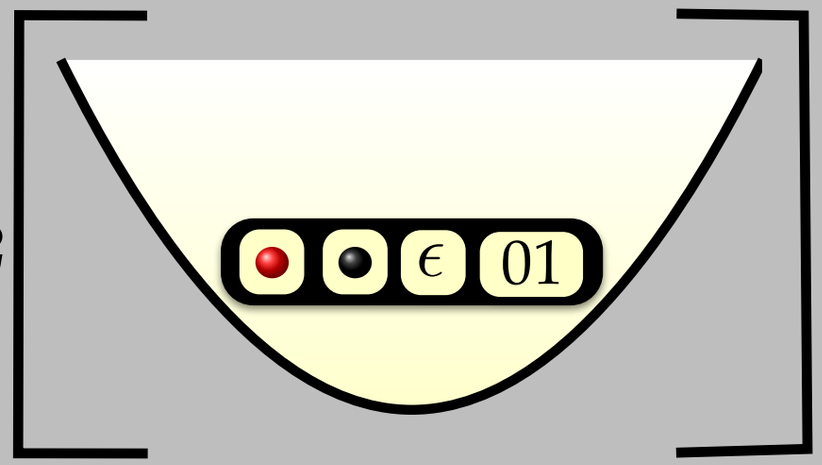
$$Pre_{t_2}\left[\epsilon\ 01\right] = \epsilon\ 010$$

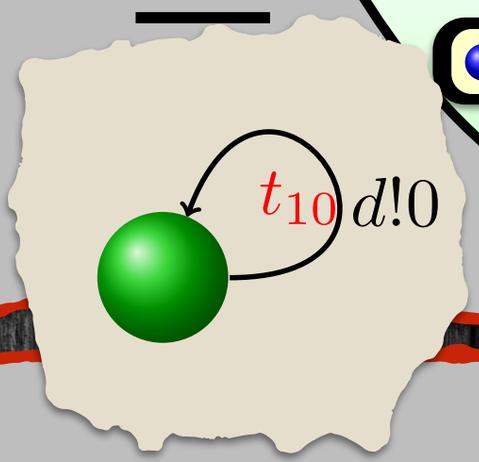


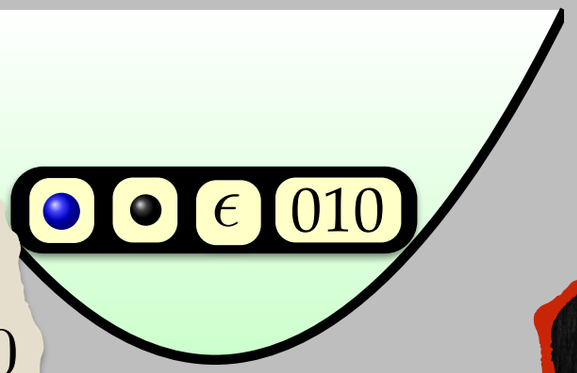
Lossy: Computing Predecessors



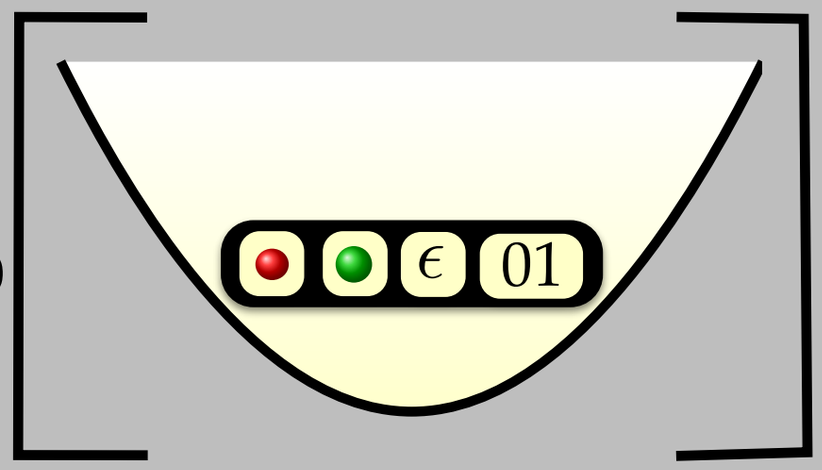
Pre_{t_2}





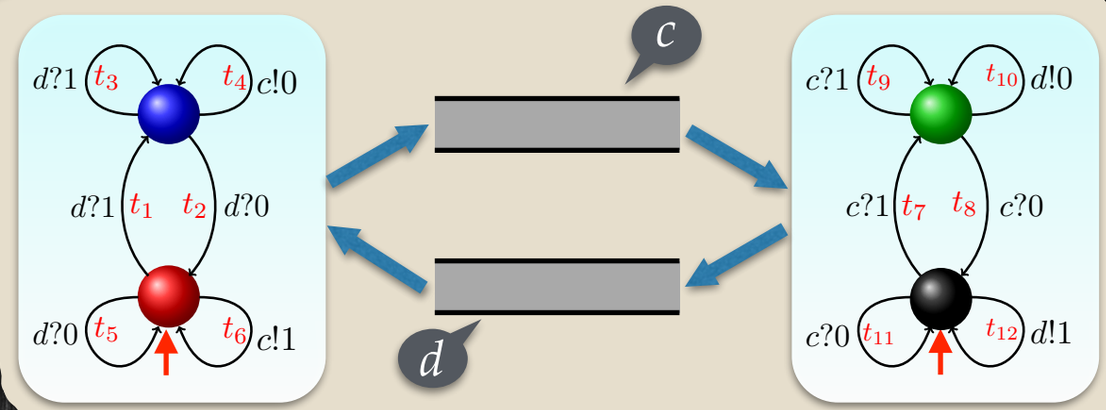


$Pre_{t_{10}}$

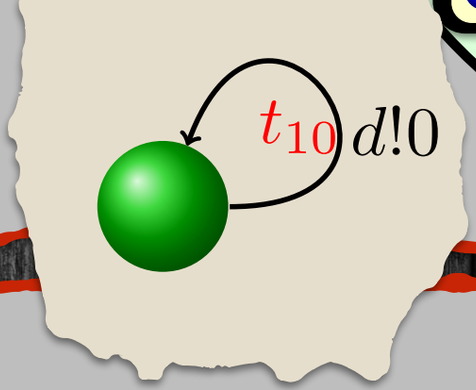


=

Lossy: Computing Predecessors



Pre_{t_2} [ϵ 01] = ϵ 010



$Pre_{t_{10}}$ [ϵ 01] = ϵ 1

Lossy Channel Systems

Model ✓

Configurations ✓

Transitions ✓

Ordering ✓

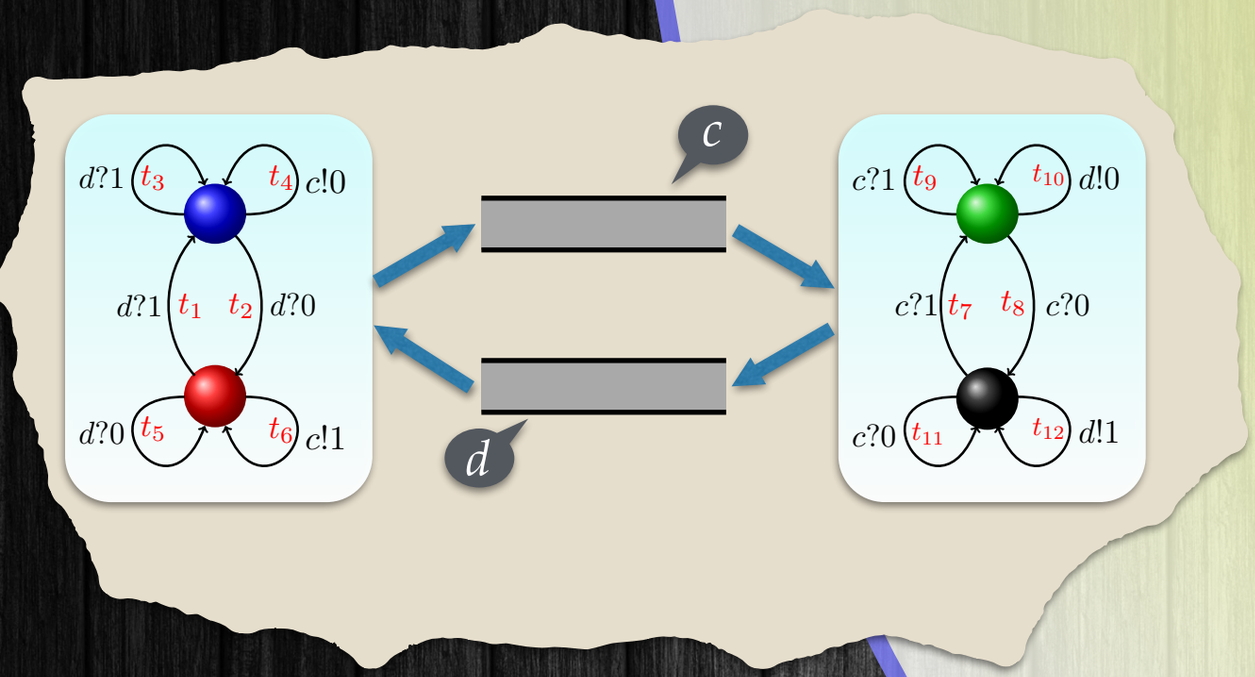
Monotoncity ✓

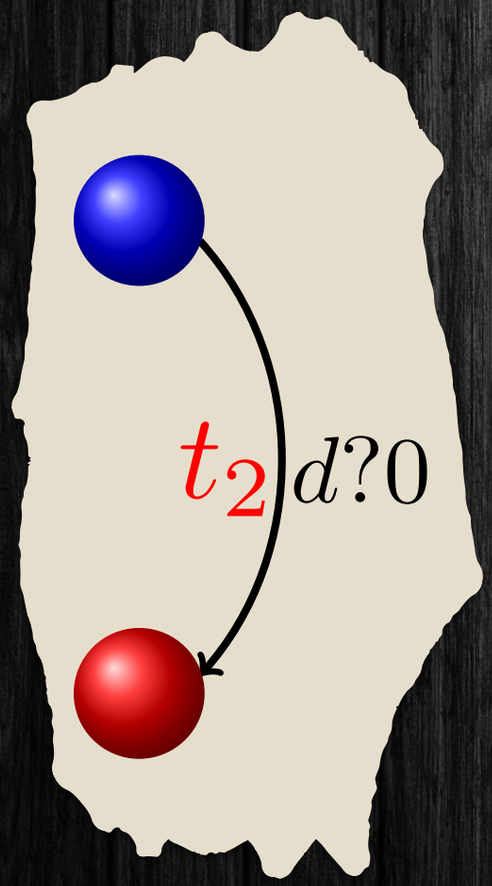
Upward Closed Sets ✓

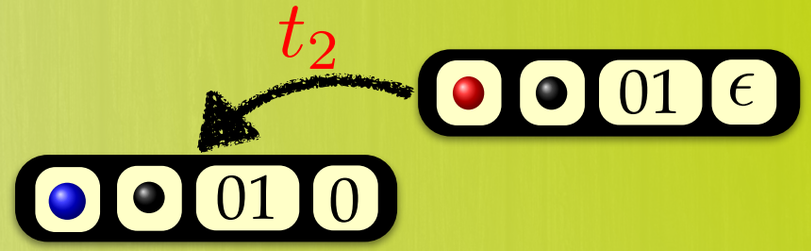
Computing Predecessors ✓

Backward Reachability

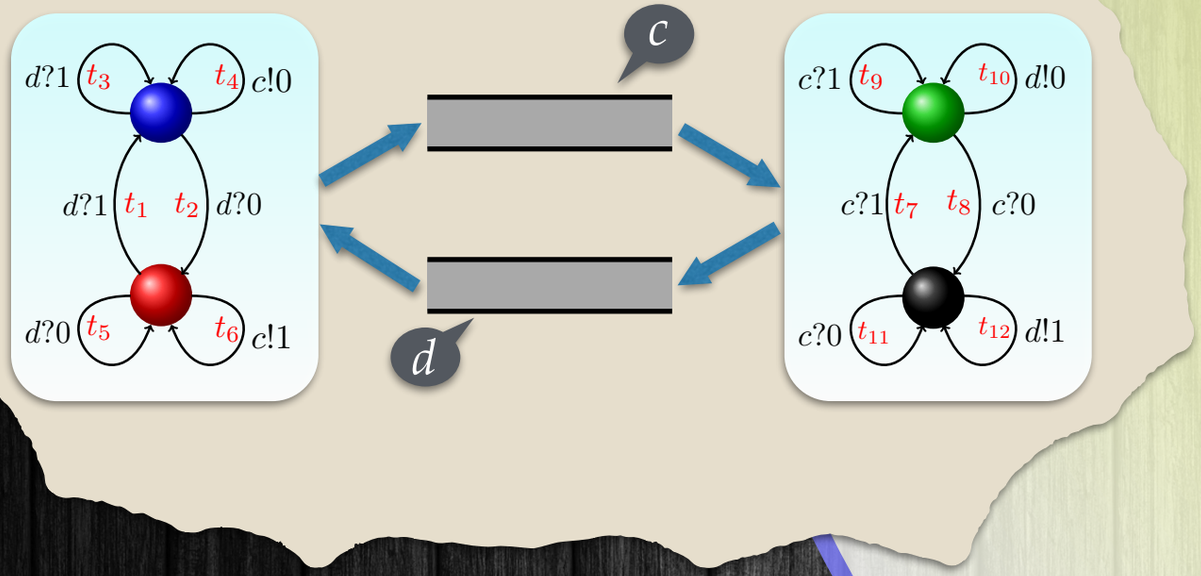
Lossy Backward Reachability

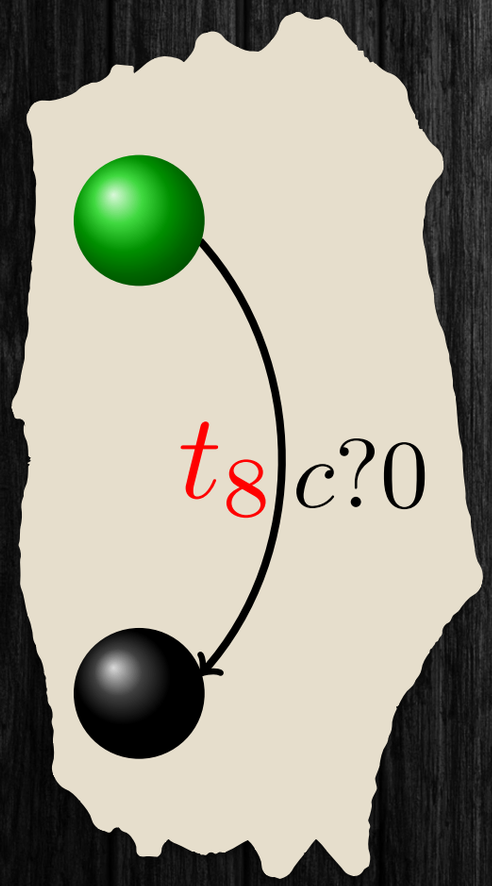


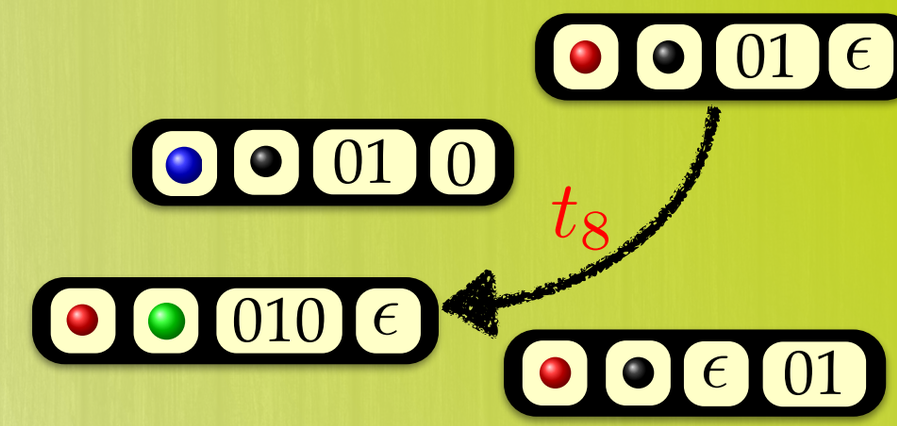




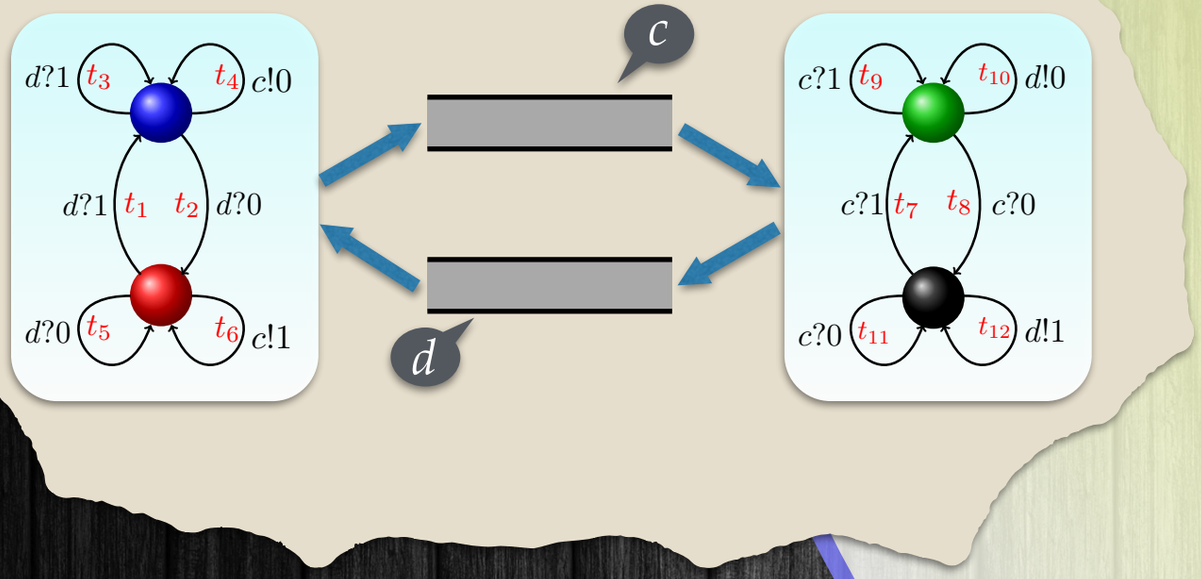
Lossy Backward Reachability

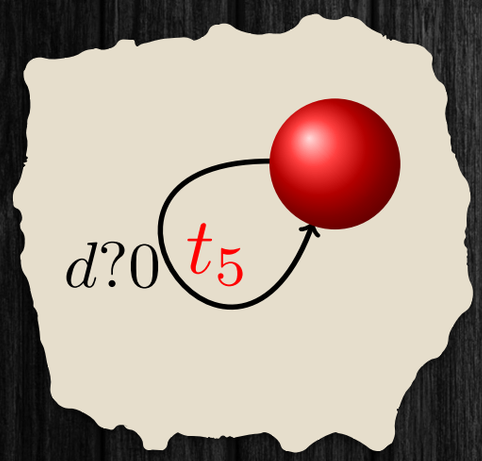


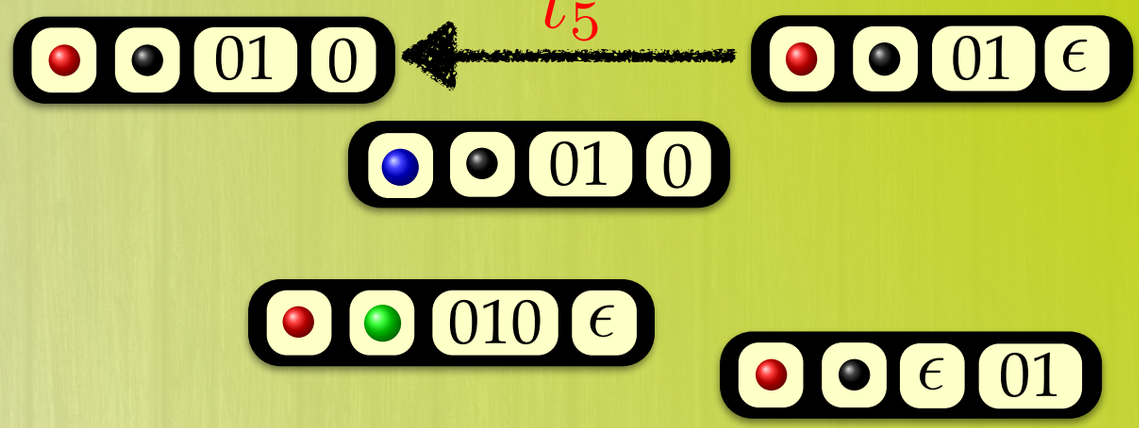




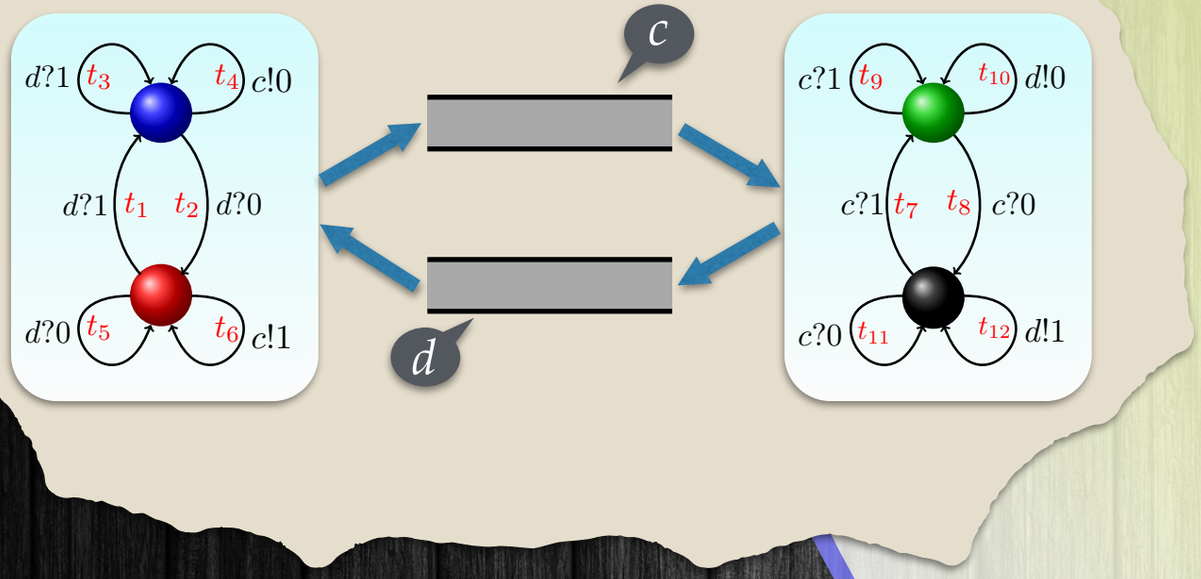
Lossy Backward Reachability

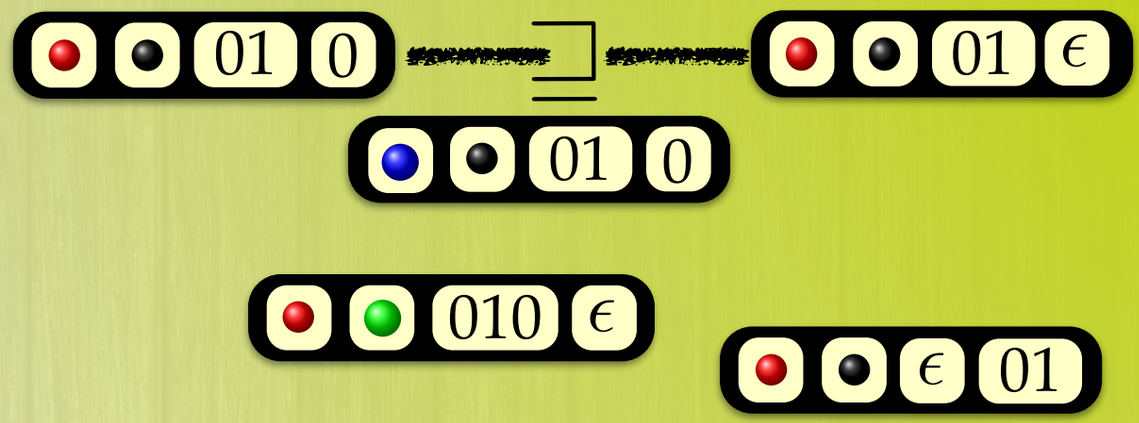




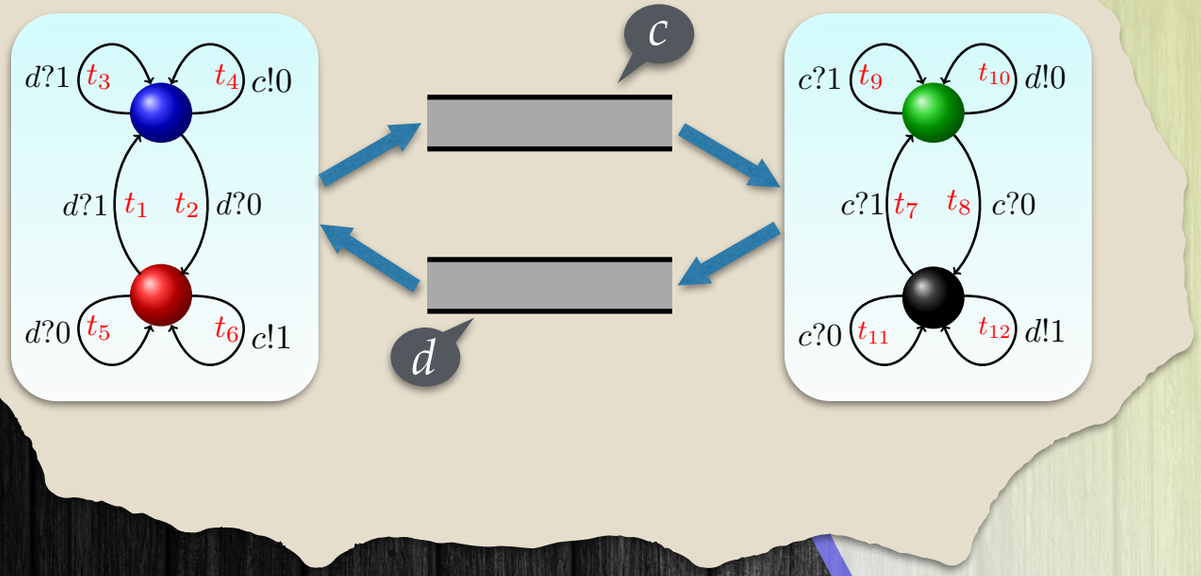


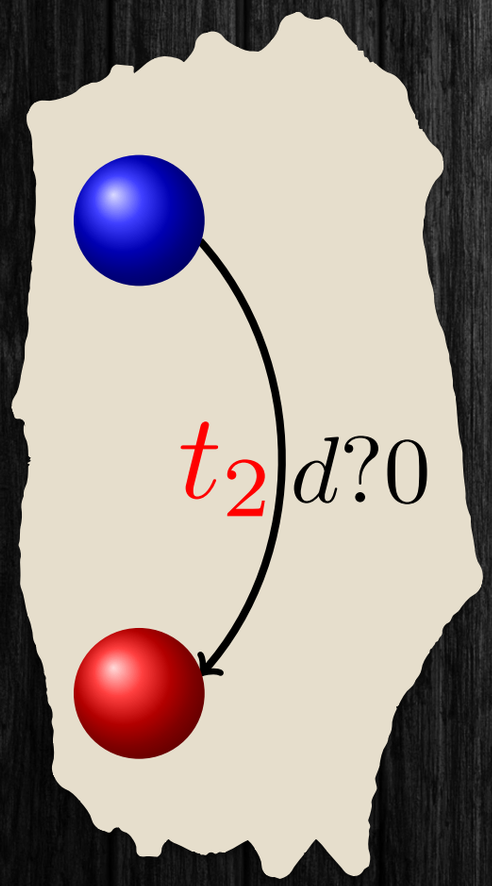
Lossy Backward Reachability

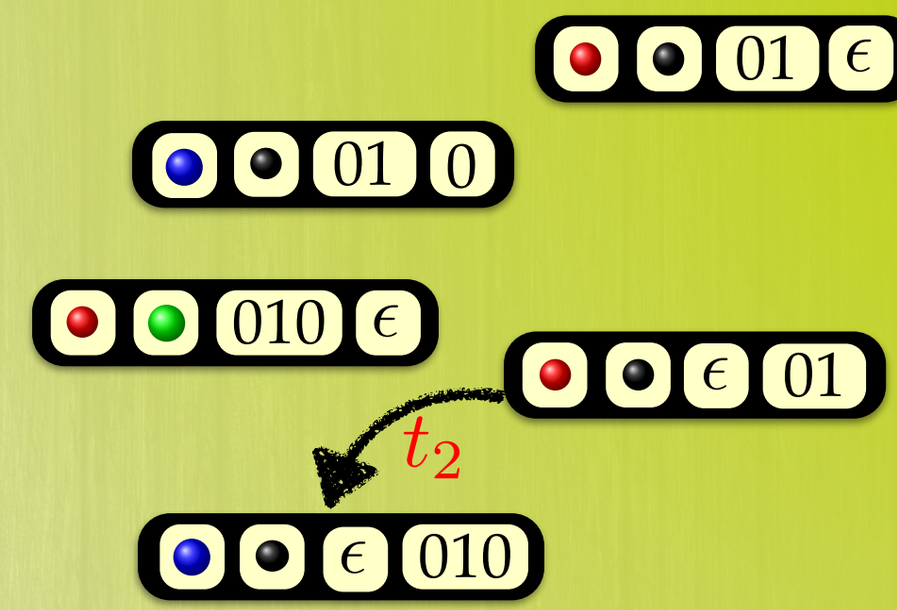




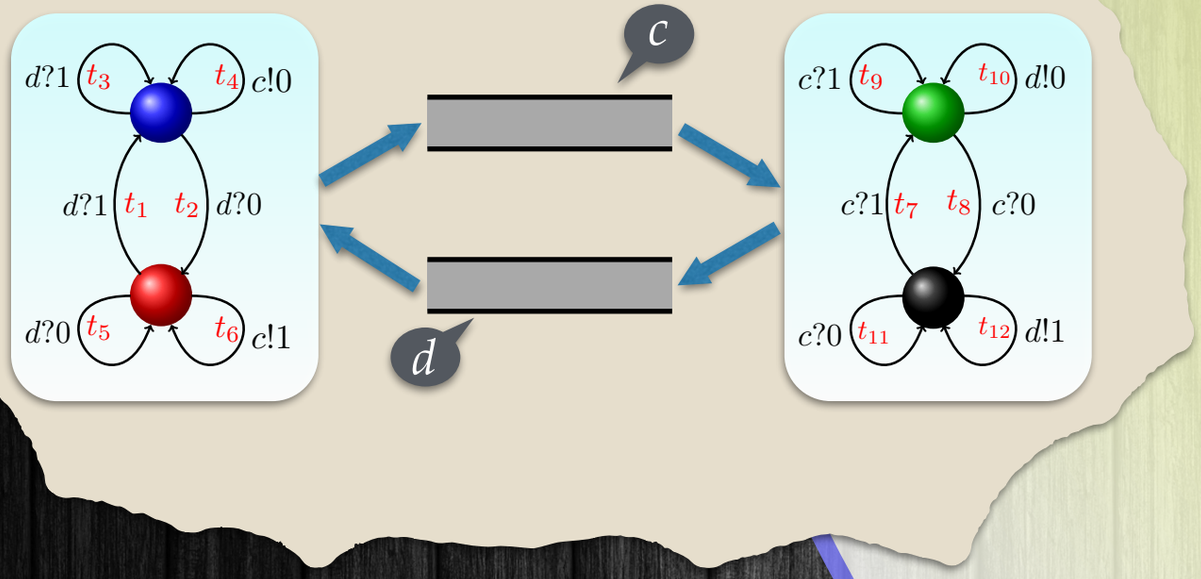
Lossy Backward Reachability

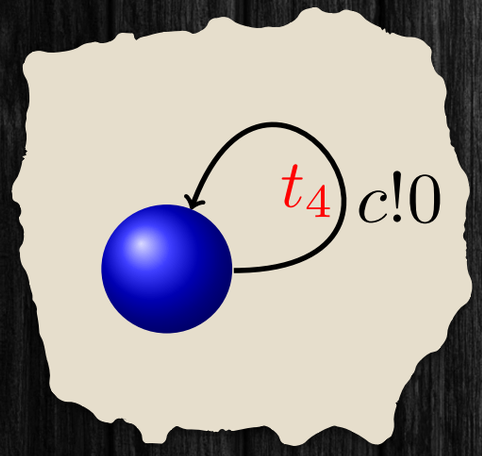




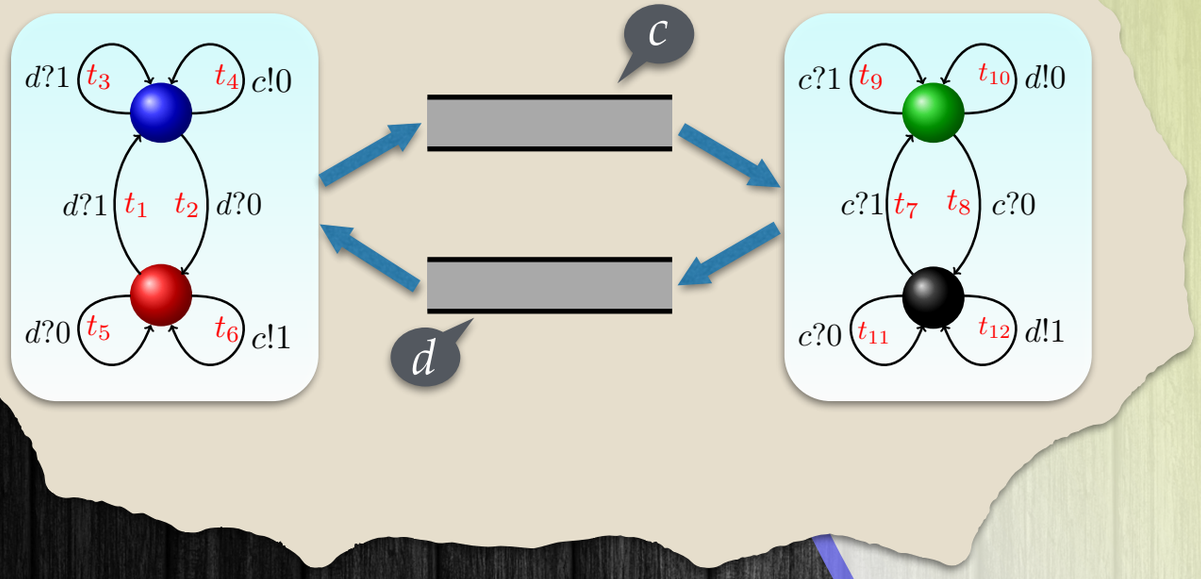


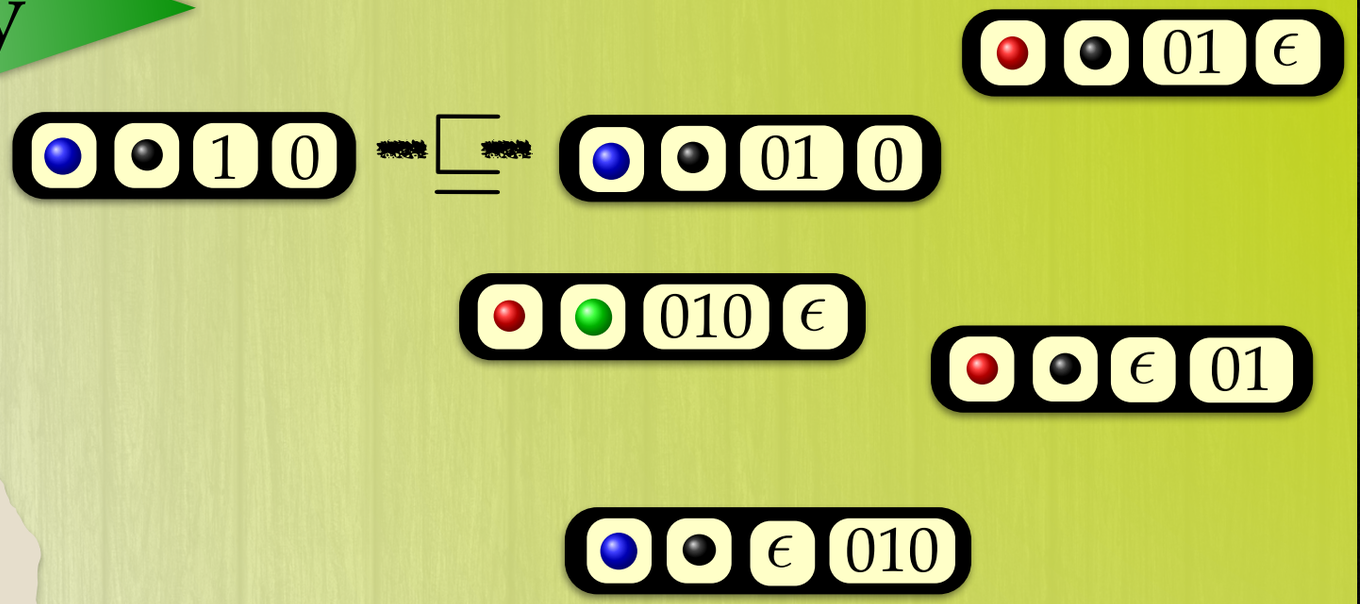
Lossy Backward Reachability



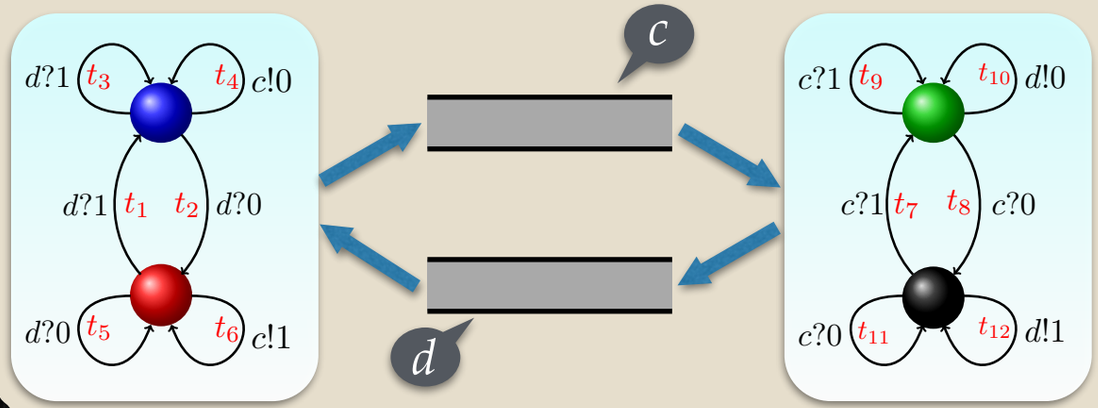


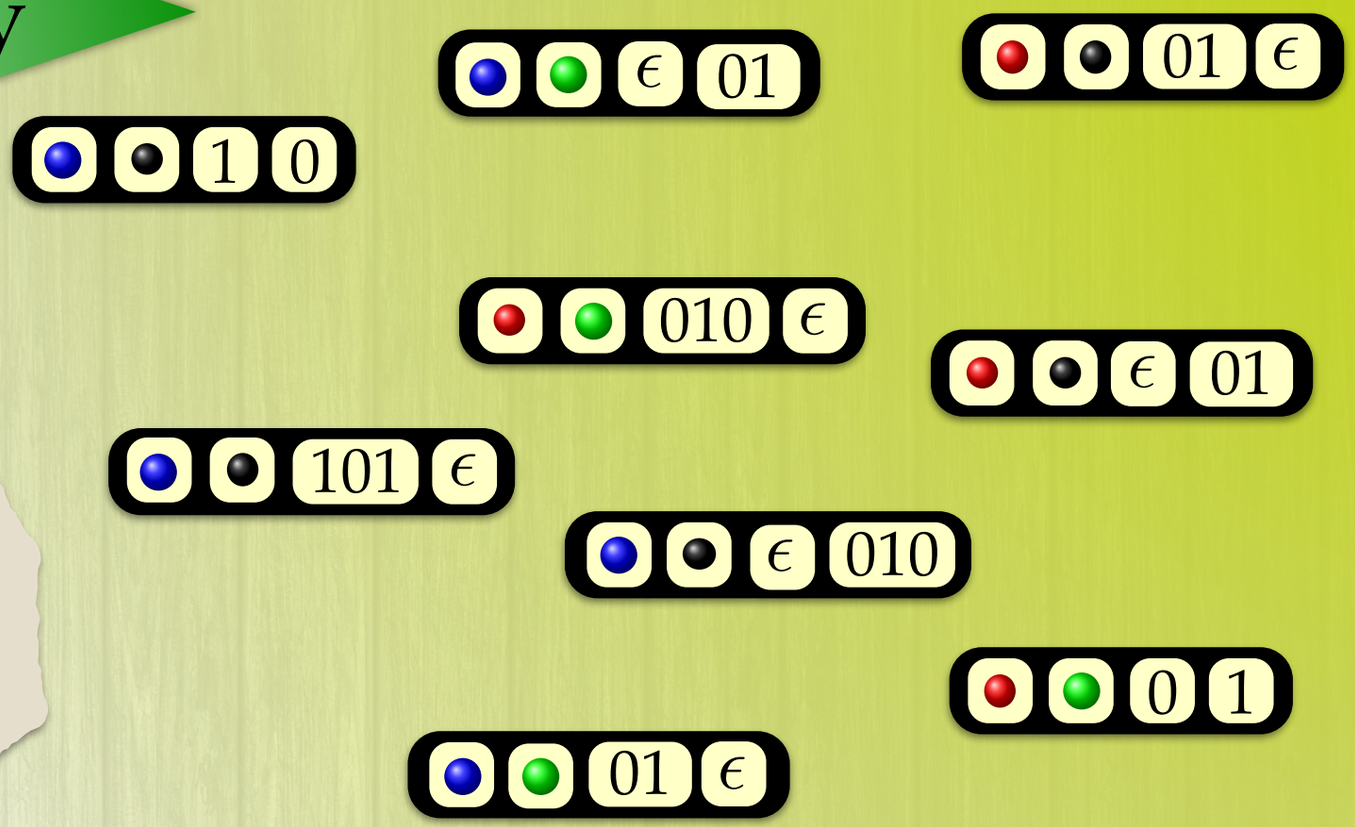
Lossy Backward Reachability





Lossy Backward Reachability





Lossy Backward Reachability

symbolic representation = finite words

Lossy Backward Reachability

symbolic representation = finite words

Termination: words well quasi-ordered

Los Well Quasi-Ordering

Well Quasi-Ordering

infinite sequence of words

$$w_0, w_1, w_2, \ldots, w_i, \ldots, w_j, \ldots$$

$$\sqsubseteq$$

$$\exists i < j : w_i \sqsubseteq w_j$$

Los Well Quasi-Ordering

Well Quasi-Ordering

infinite sequence of words

$$w_0, w_1, w_2, \ldots, w_i, \ldots, w_j, \ldots$$

$$\sqsubseteq$$

$$\exists i < j : w_i \sqsubseteq w_j$$

Well Quasi-Ordering

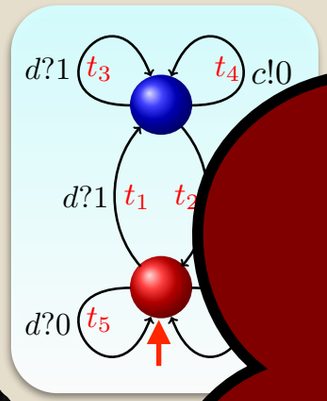
infinite sequence of configurations

$$c_0, c_1, c_2, \ldots, c_i, \ldots, c_j, \ldots$$

$$\sqsubseteq$$

$$\exists i < j : c_i \sqsubseteq c_j$$

Lossy Backward Reachabili



Ordering:

- monotonicity
- computing predecessors
- well quasi-ordering

rds

rdered

Background

Parameterized Systems

Petri Nets

Lossy Channel Systems

Timed Petri Nets

Timed Petri Nets

Timed Petri Nets

Model

Configurations

Transitions

Ordering

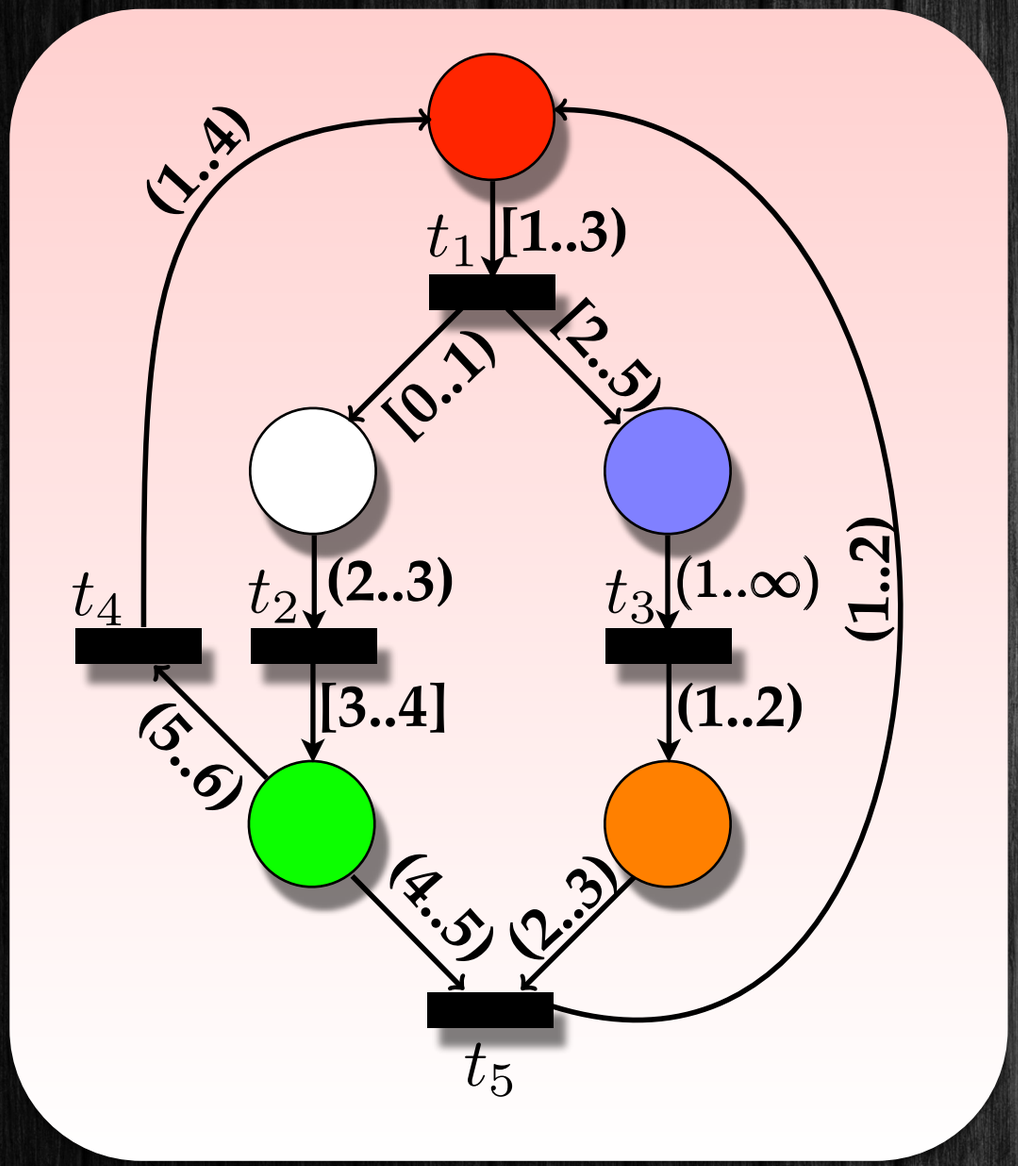
Monotoncity

Upward Closed Sets

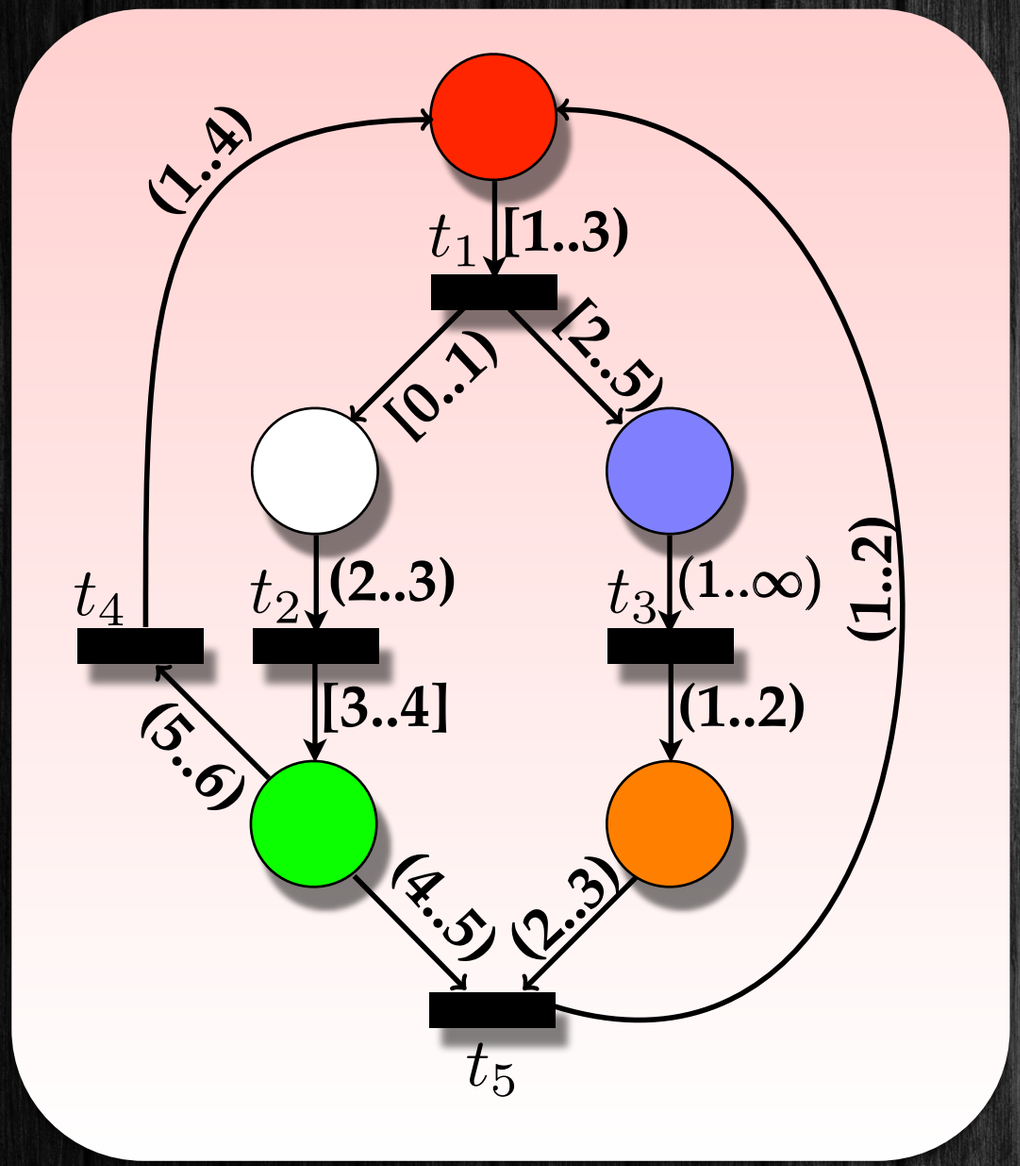
Computing Predecessors

Backward Reachability

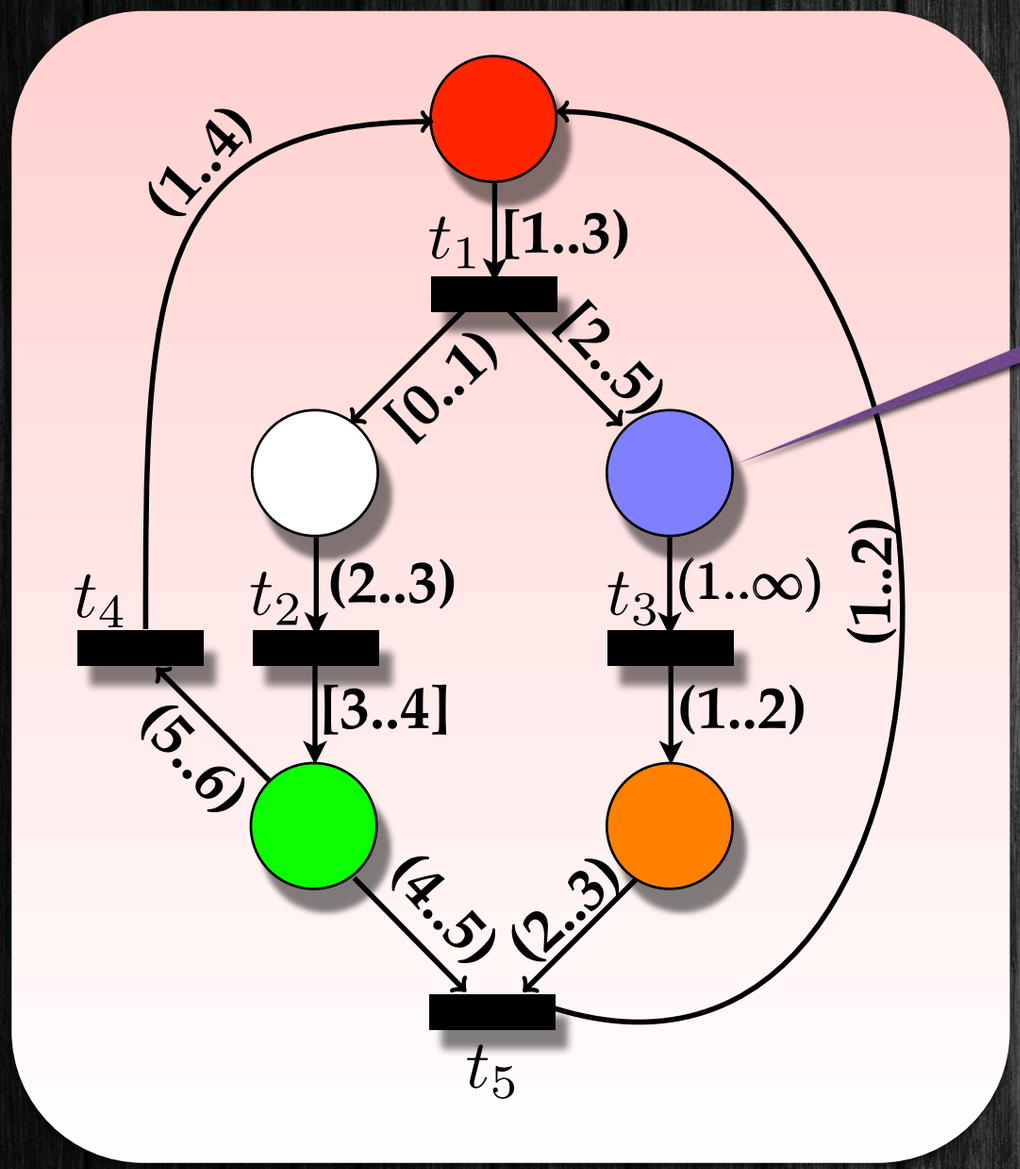
Timed Petri Nets



Timed Petri Nets

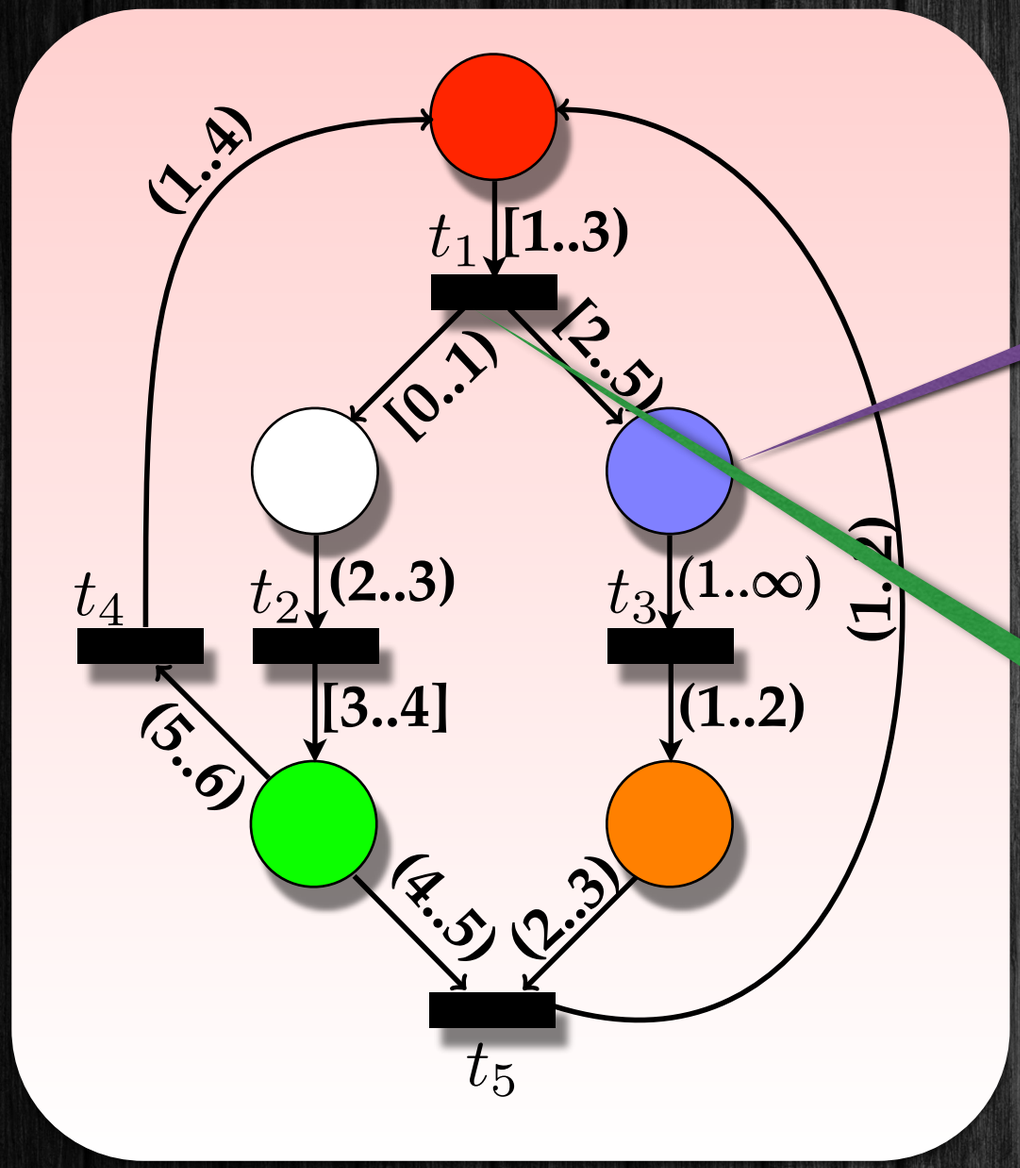


Timed Petri Nets



places

Timed Petri Nets



places

transitions

Timed Petri Nets

Model

Configurations

Transitions

Ordering

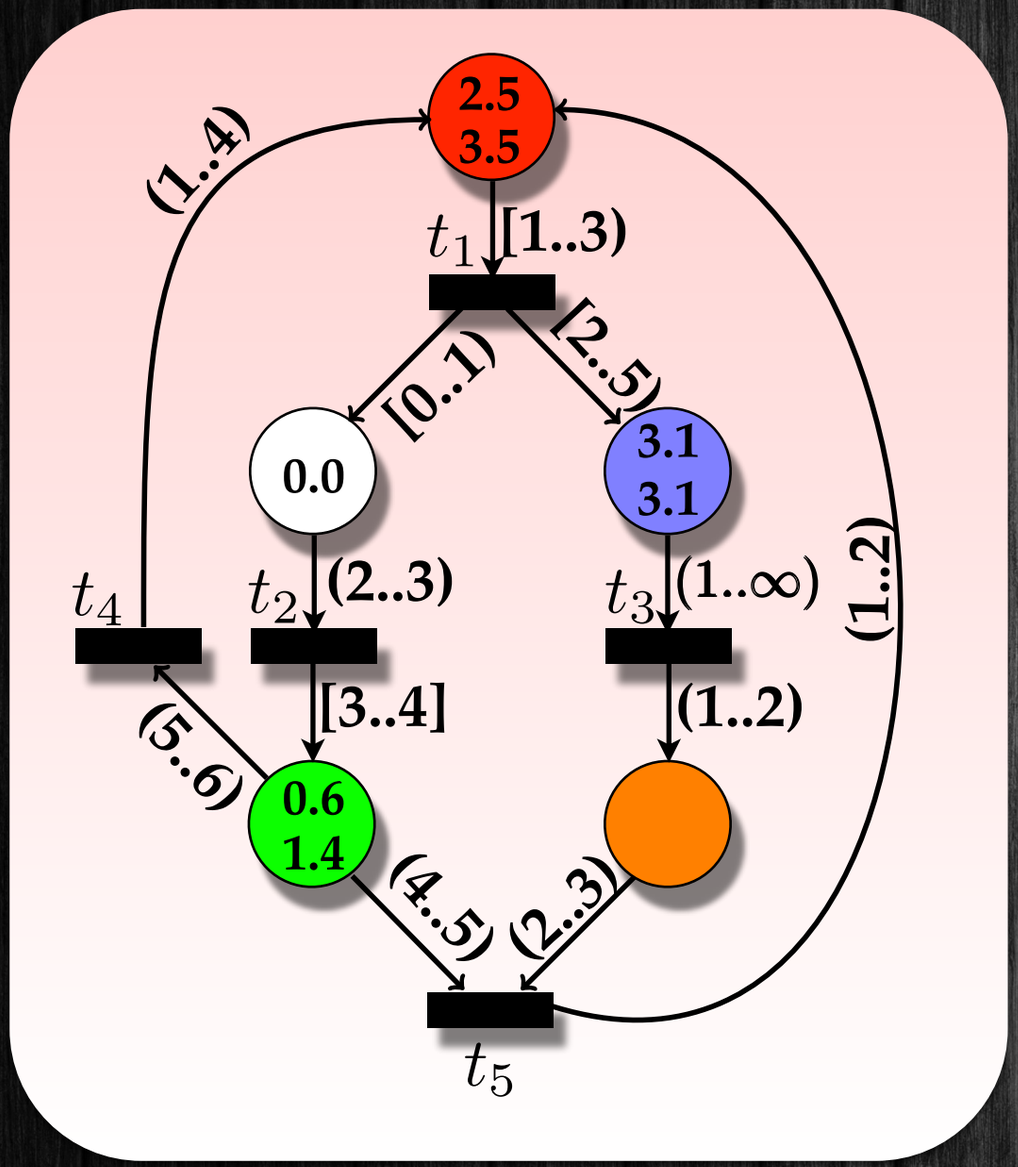
Monotoncity

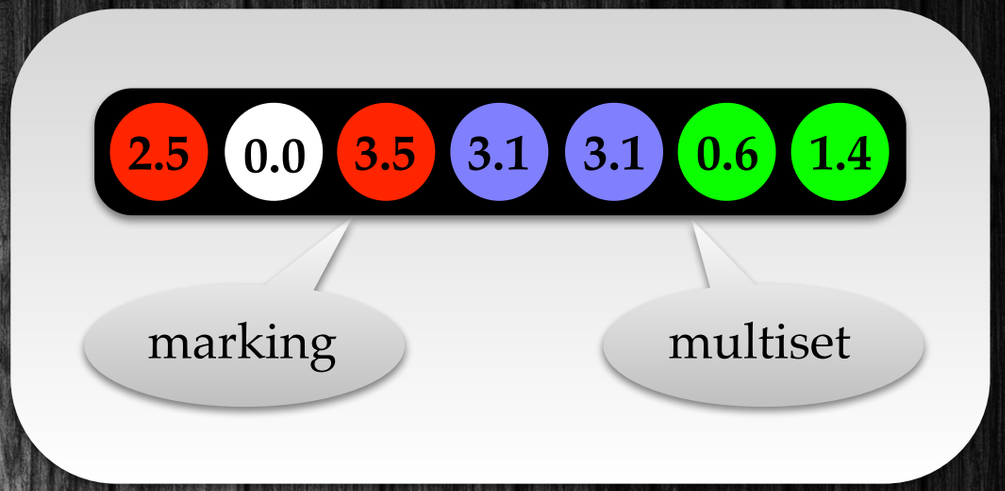
Upward Closed Sets

Computing Predecessors

Backward Reachability

Markings


2.5
3.5
(1..4)
t_1
[1..3)
[0..1)
[2..5)
0.0
3.1
3.1
t_4
t_2
(2..3)
t_3
(1..∞)
(1..2)
[3..4]
(1..2)
(5..6)
0.6
1.4
(4..5)
(2..3)
t_5


2.5
0.0
3.5
3.1
3.1
0.6
1.4
marking
multiset

Timed Petri Nets

Model

Configurations

Transitions

Ordering

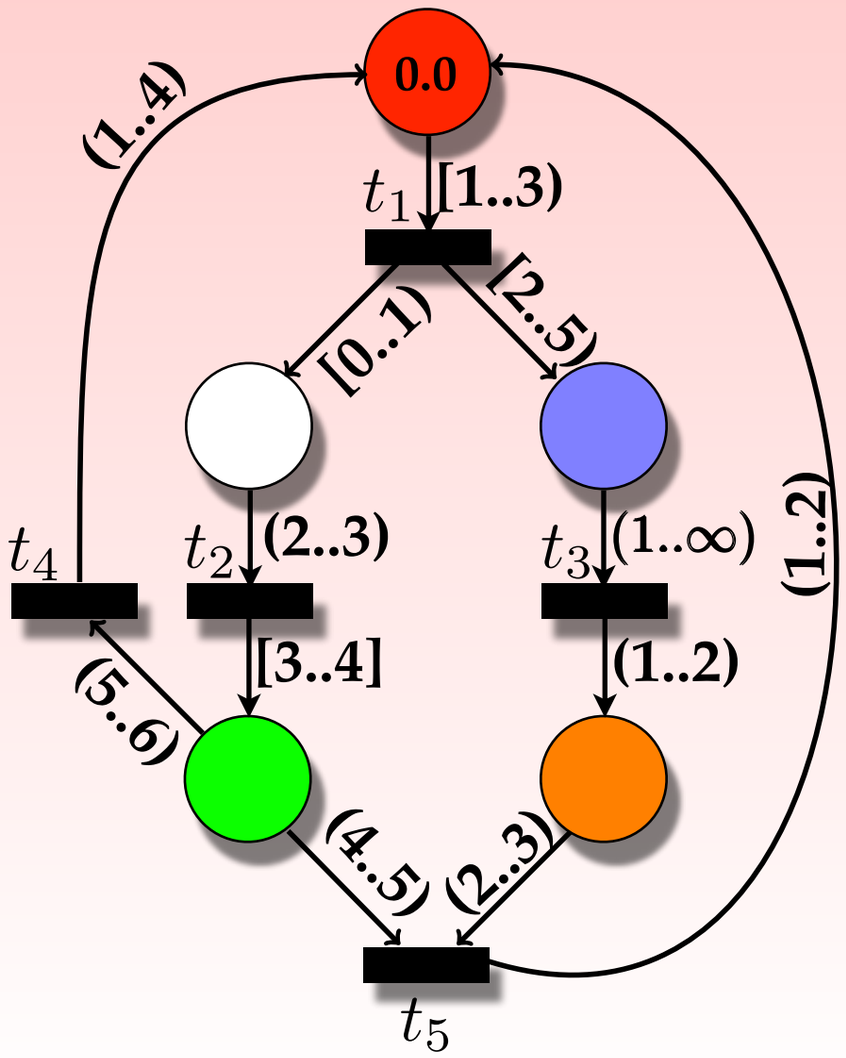
Monotoncity

Upward Closed Sets

Computing Predecessors

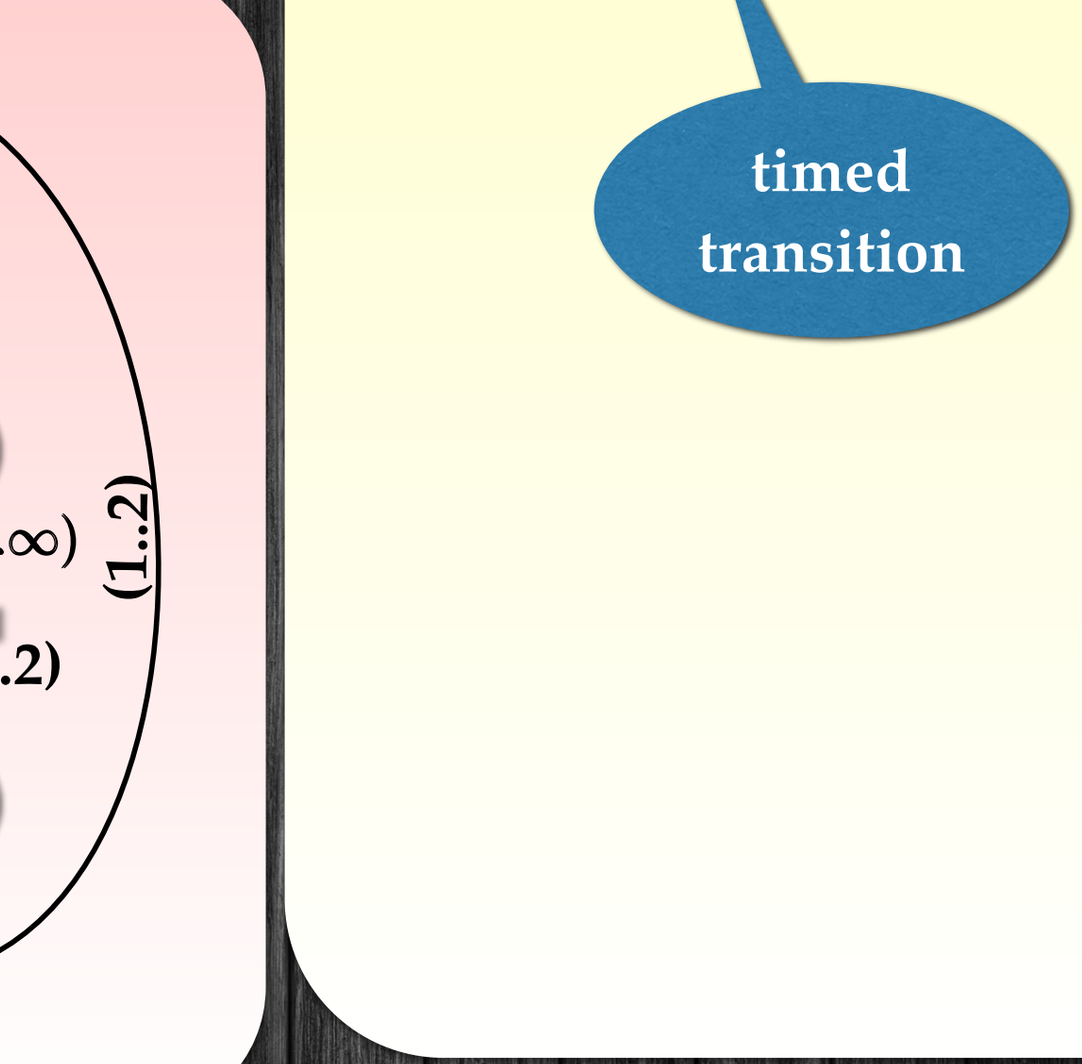
Backward Reachability

Transitions

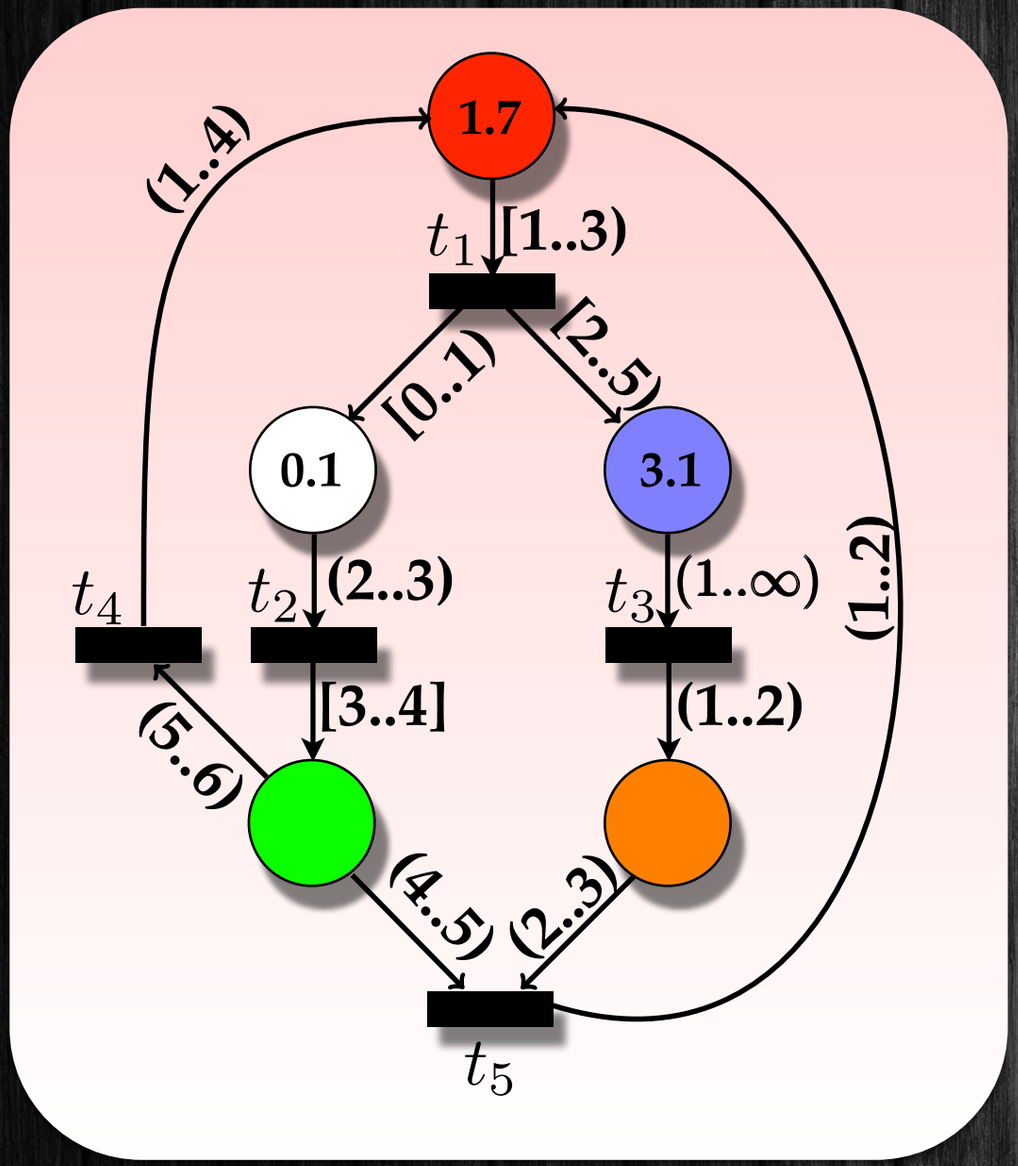


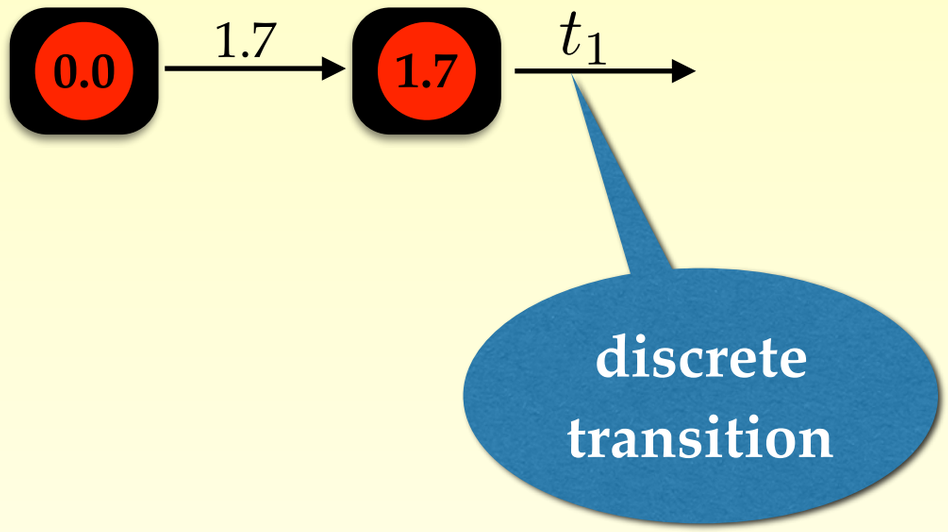
Transitions

Transitions

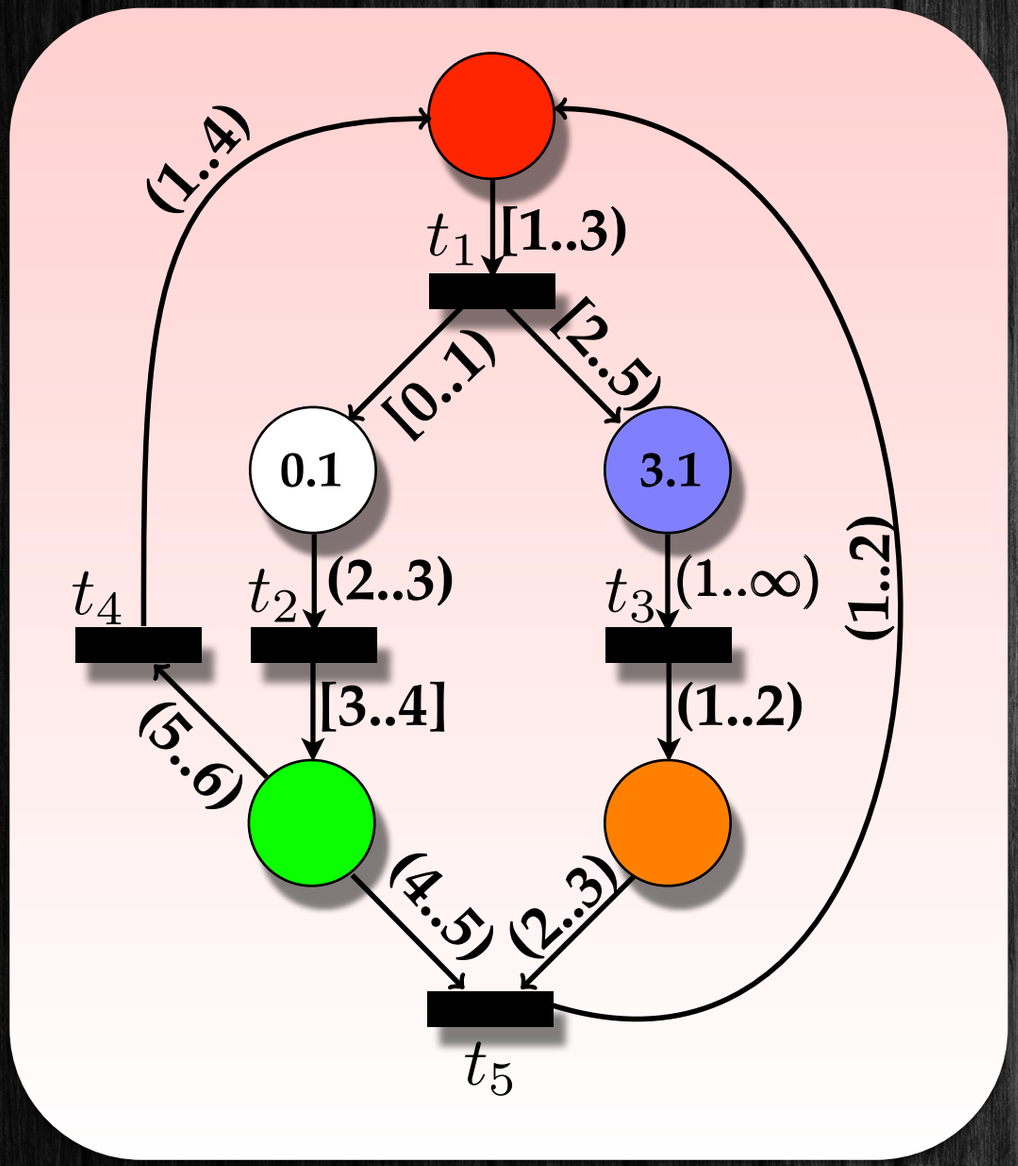


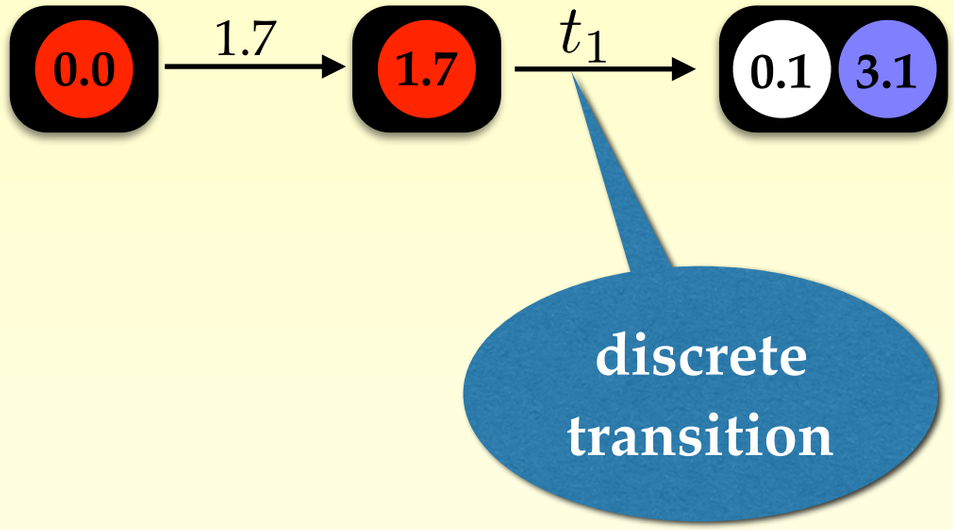
Transitions



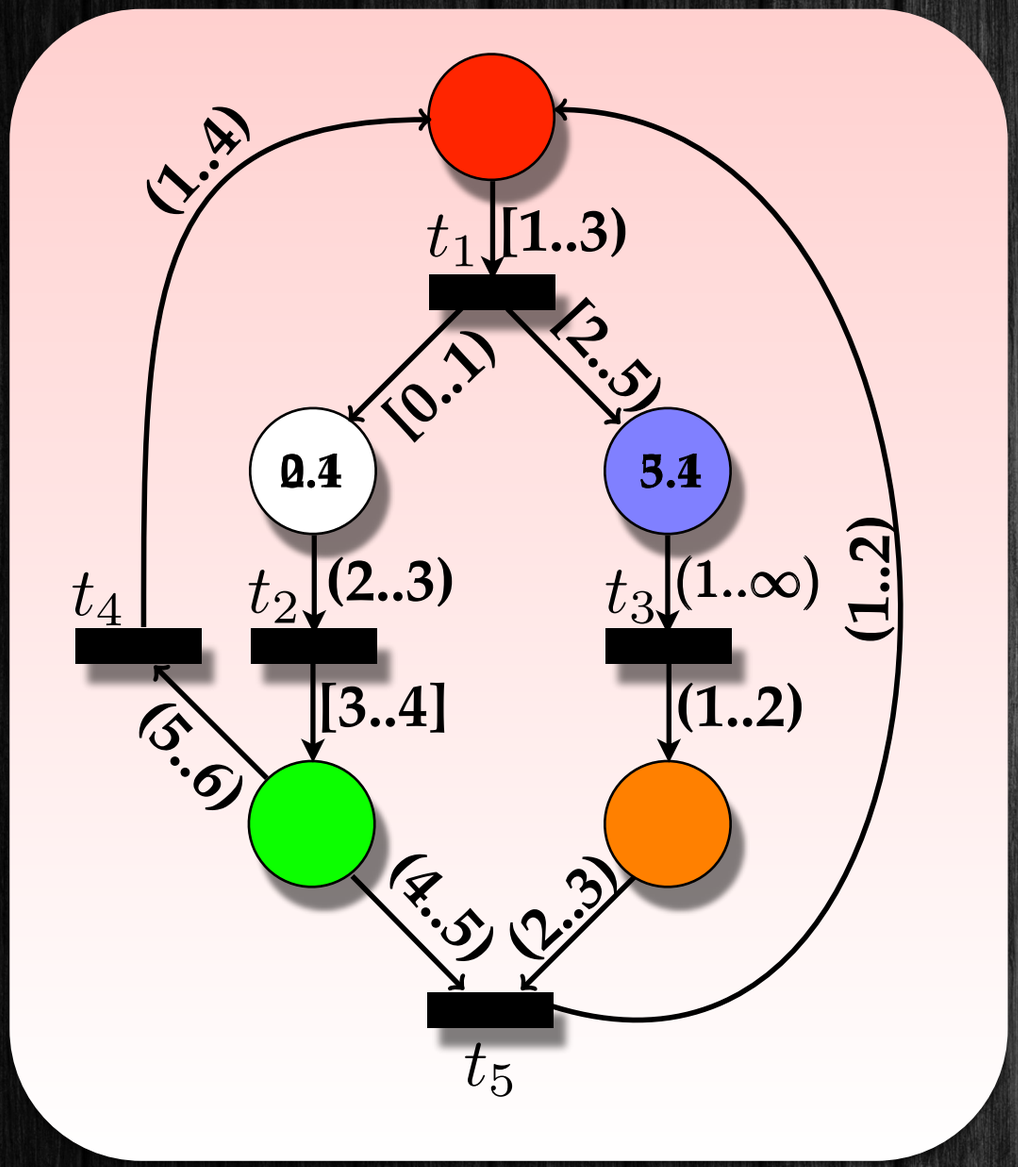


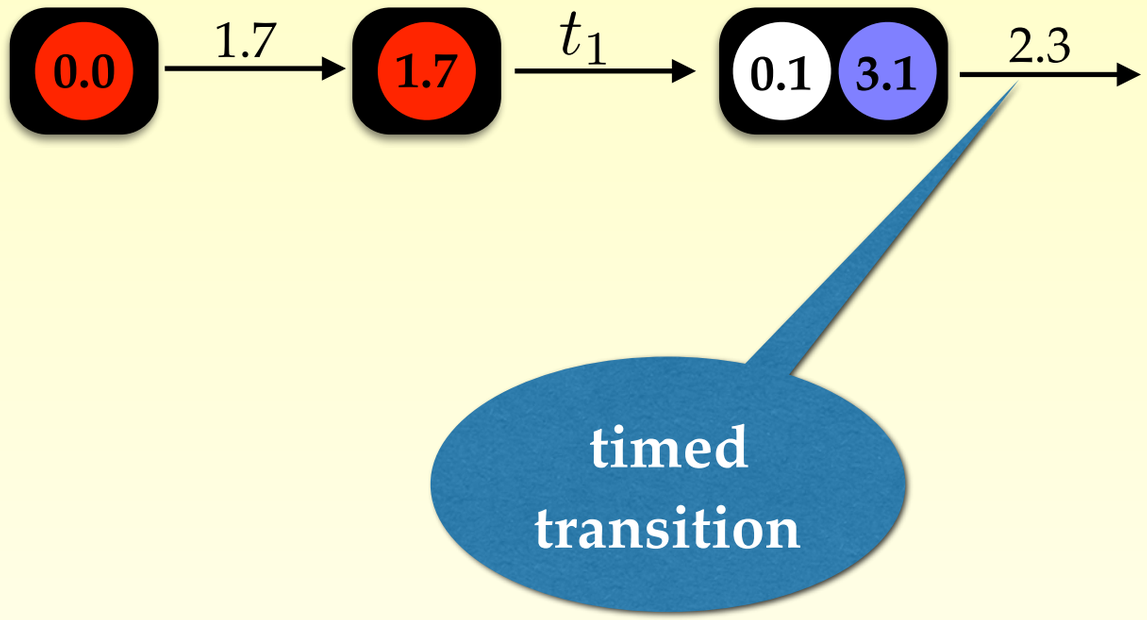
Transitions



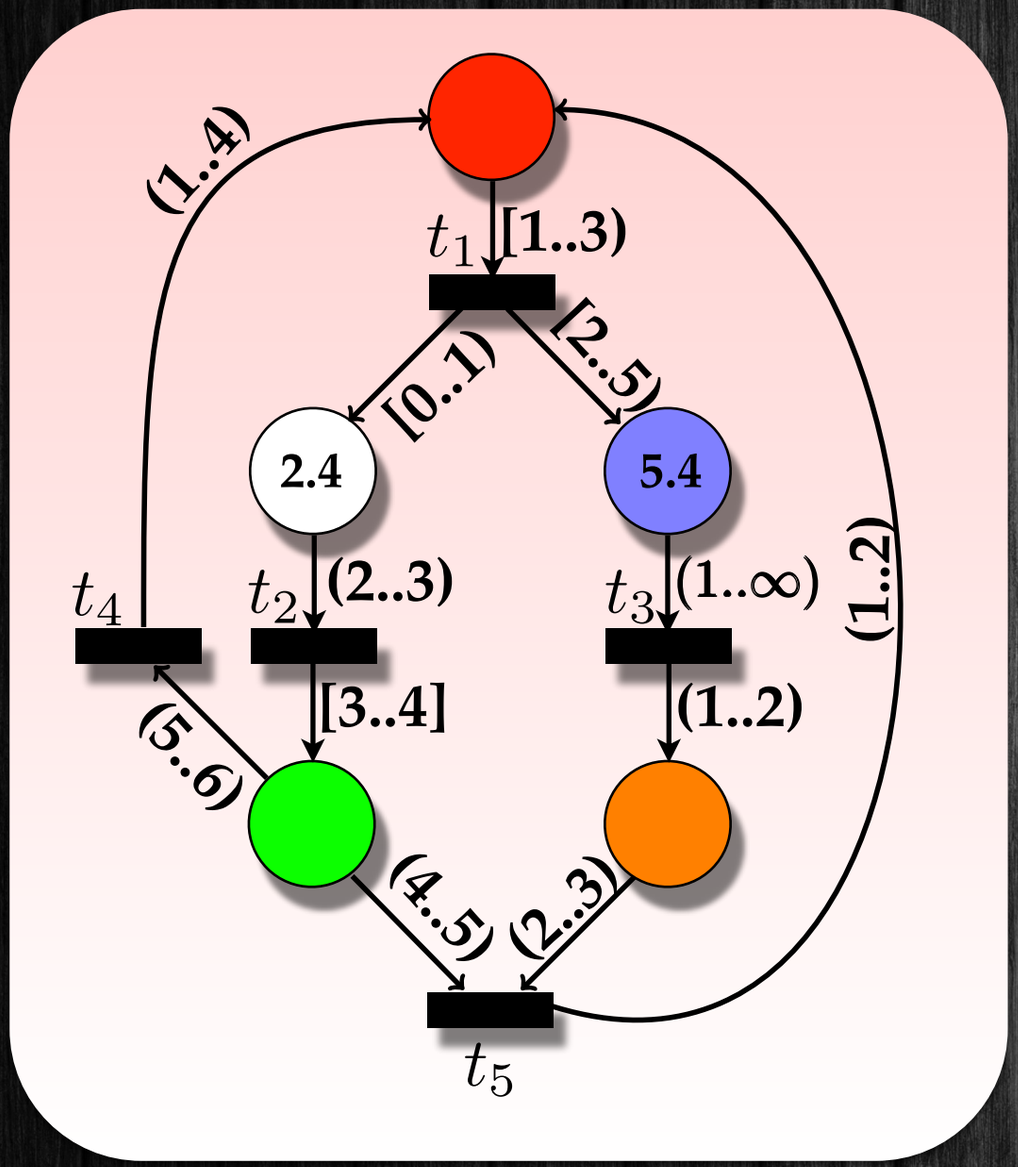


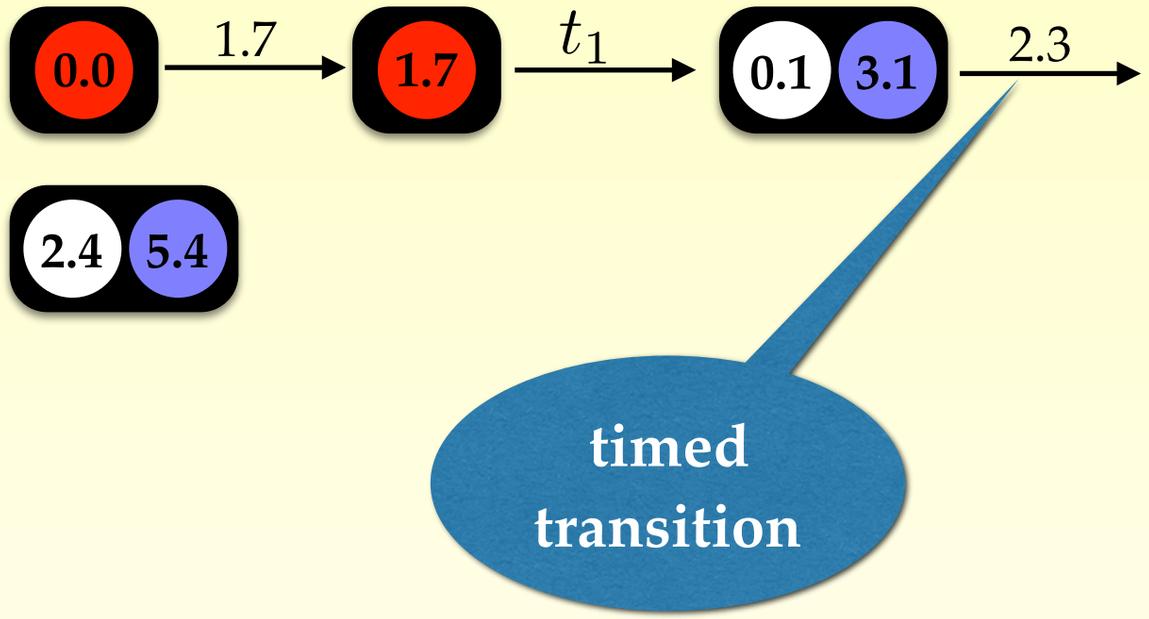
Transitions



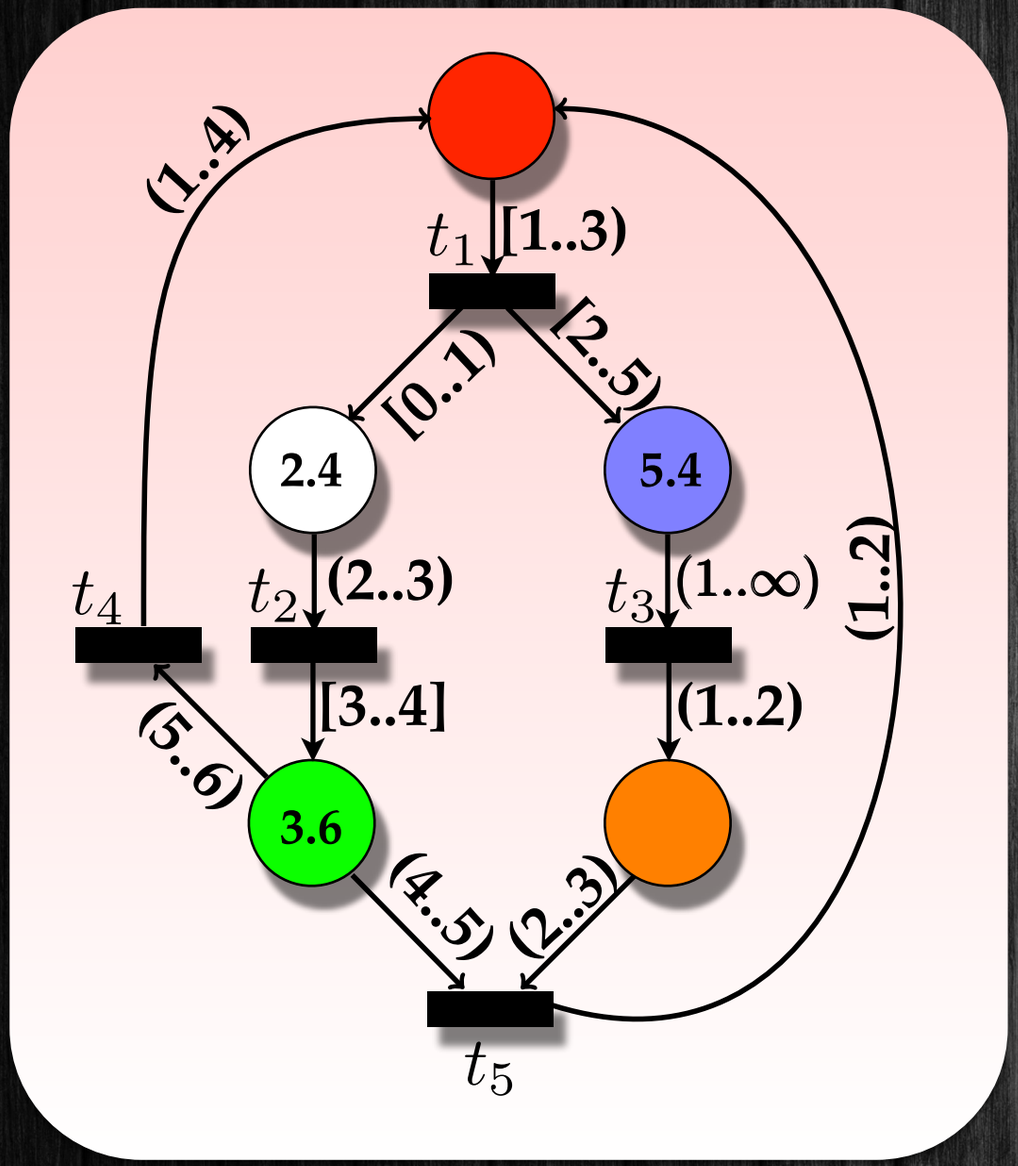


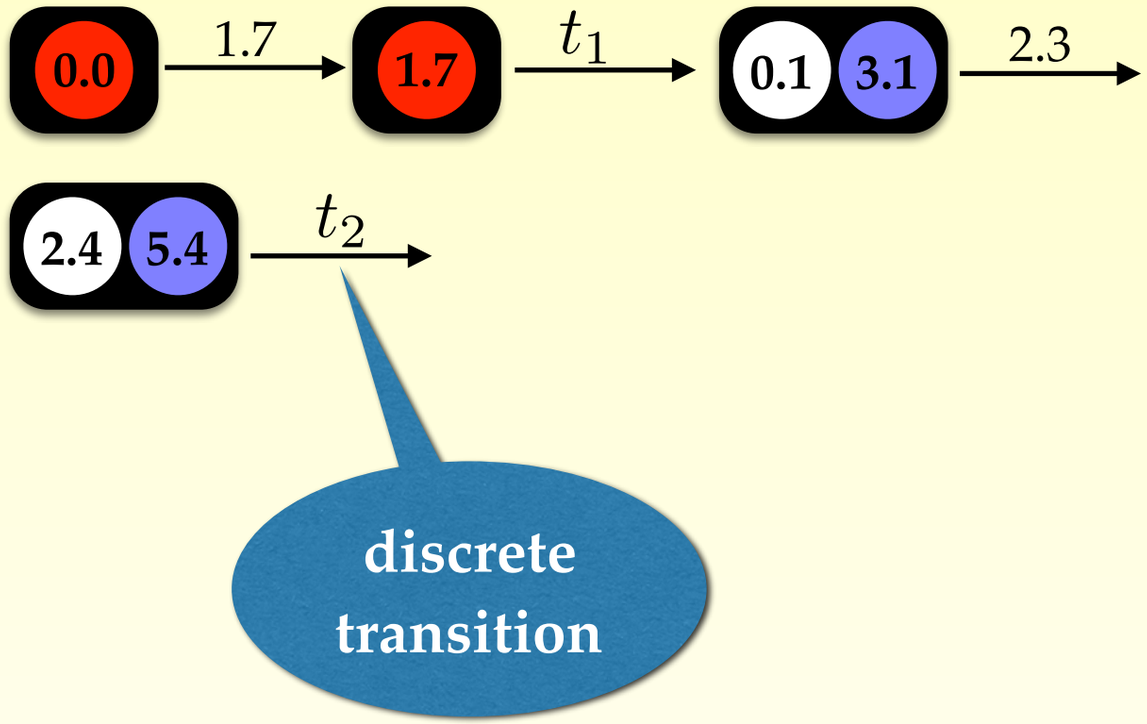
Transitions



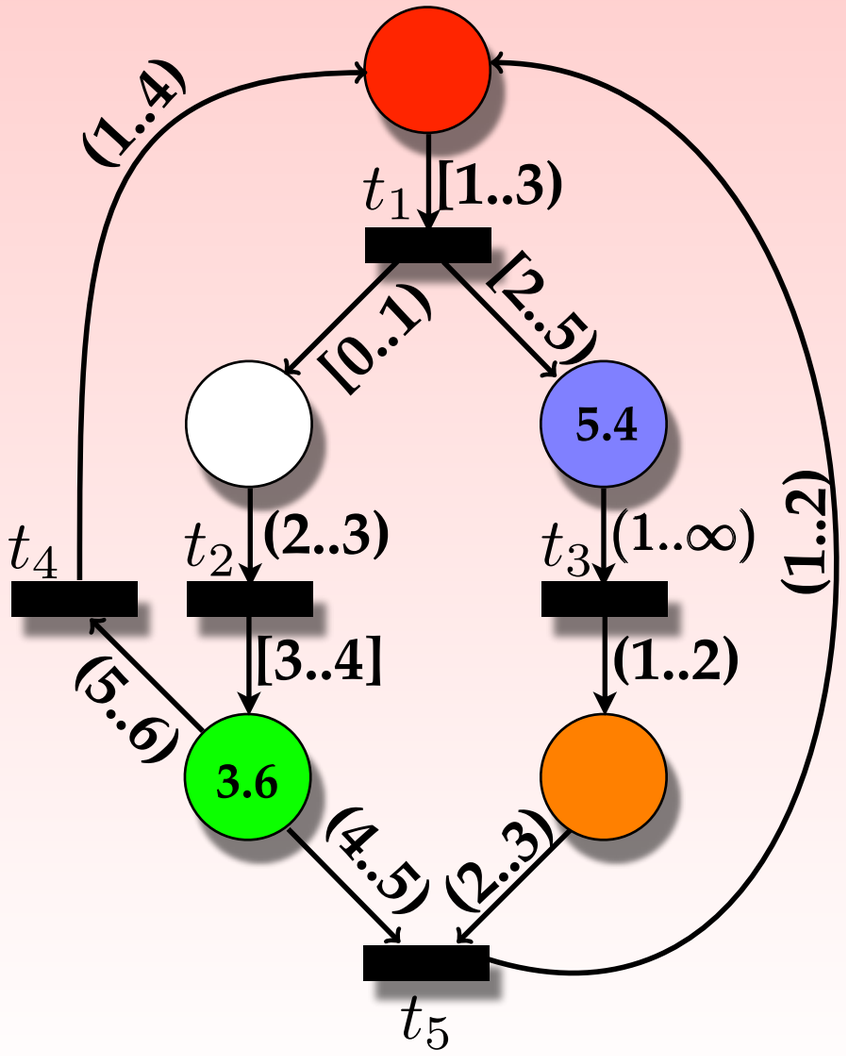


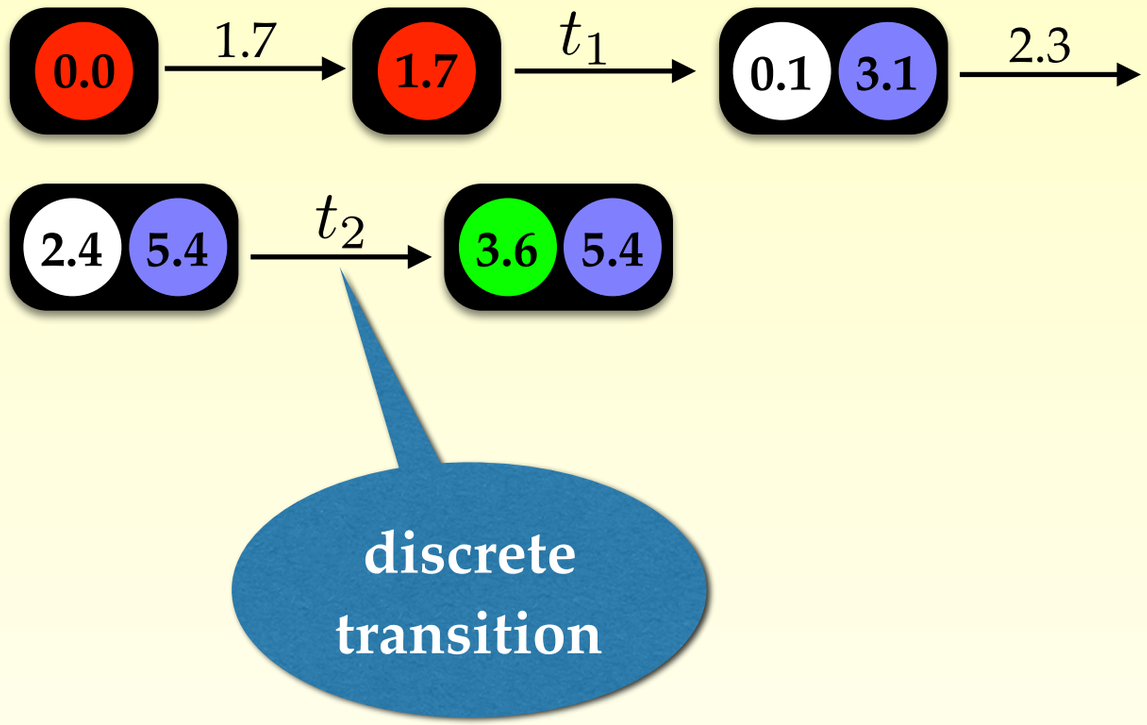
Transitions



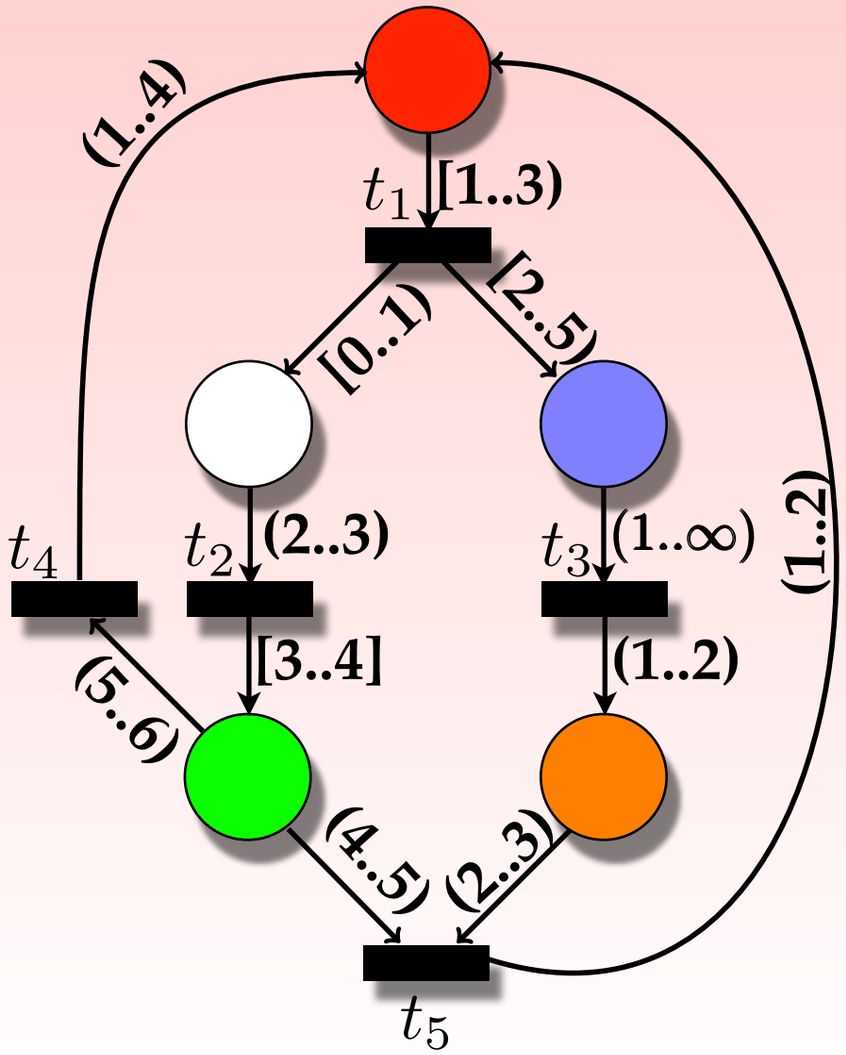


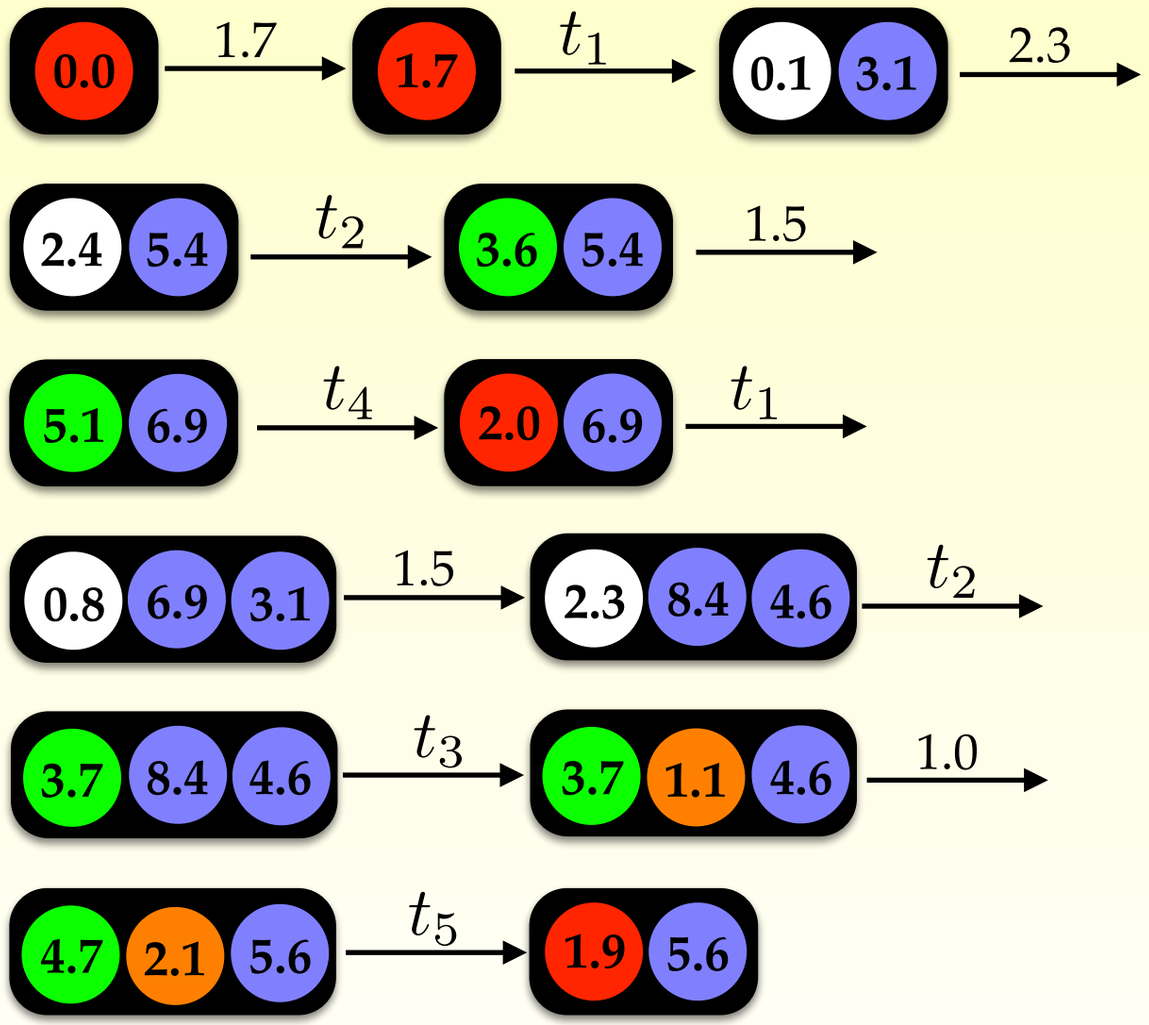
Transitions



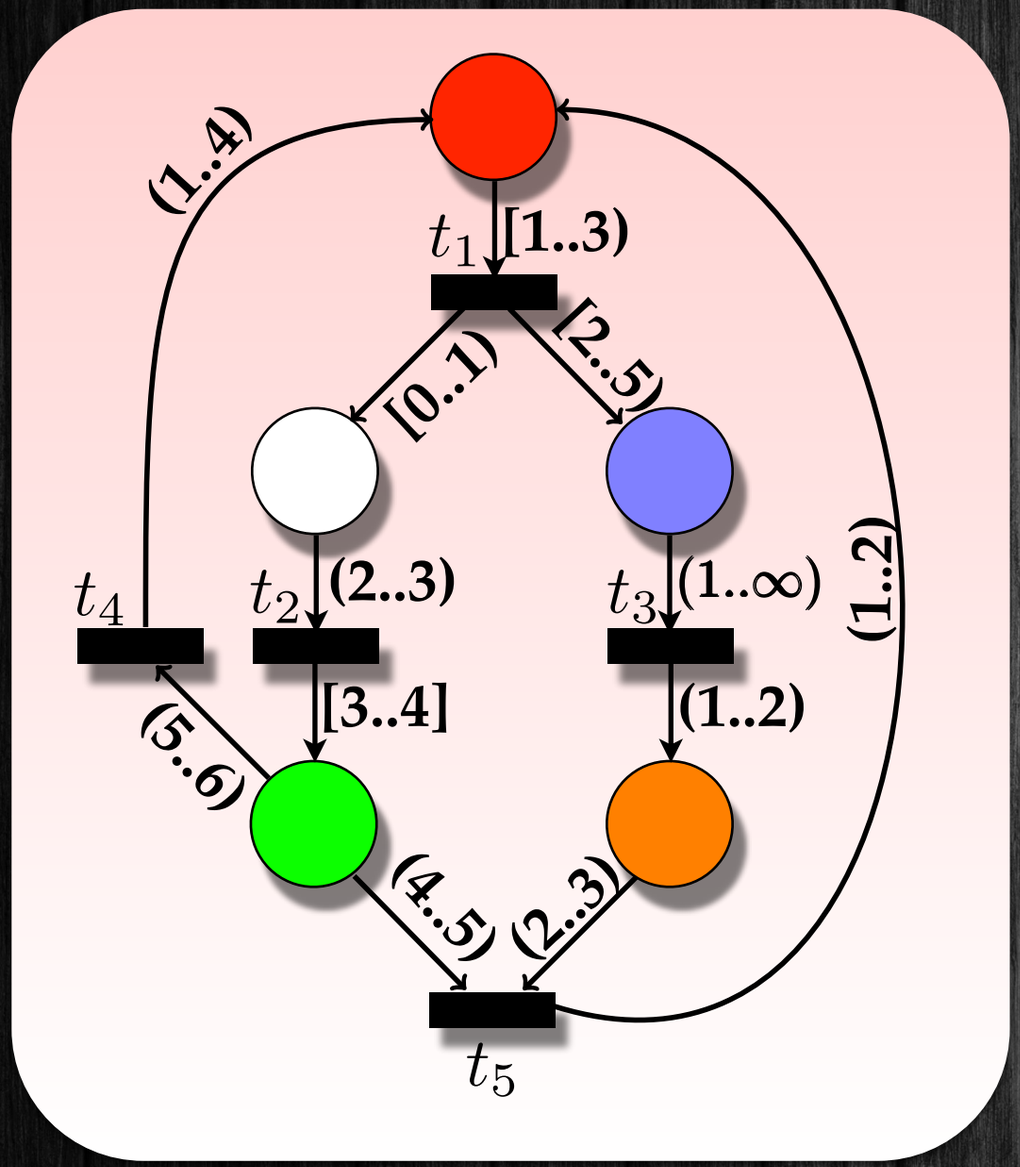


Transitions





Timed Petri Nets



cmax=6

Timed Petri Nets

Model ✓

Configurations ✓

Transitions ✓

signatures

Ordering

Monotoncity

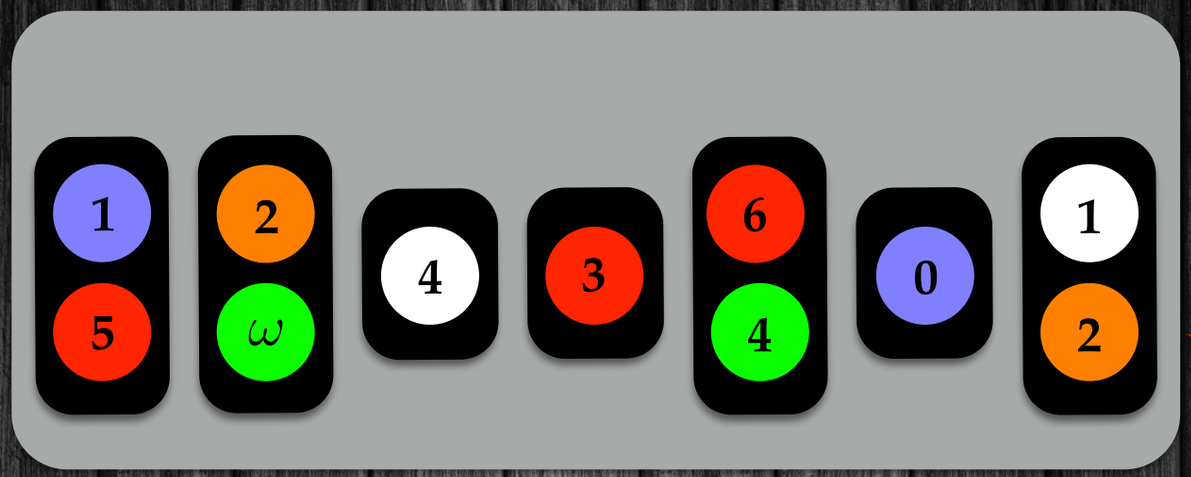
Upward Closed Sets

Computing Predecessors

Backward Reachability

Timed Pet… Signatures

cmax=6



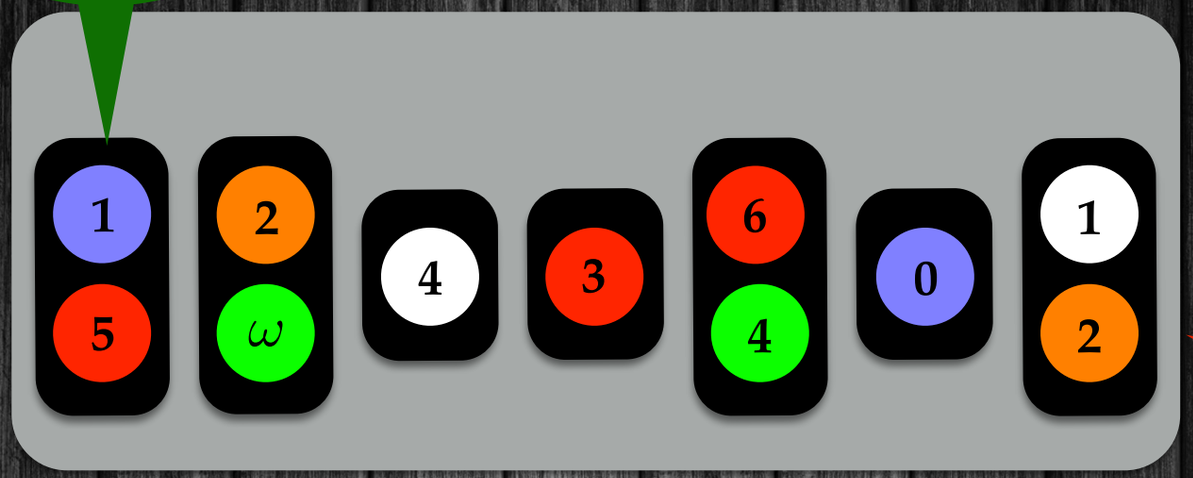
s:
signature

signature:
"sequence of multisets of colored natural numbers"

Timed Pet… Signatures

cmax=6

multiset



s:
signature

signature:
"sequence of multisets of colored natural numbers"

Timed Pet… Signatures

cmax=6

multiset

- integer part = 1
- place:

s: signature

- integer part = 5
- place:

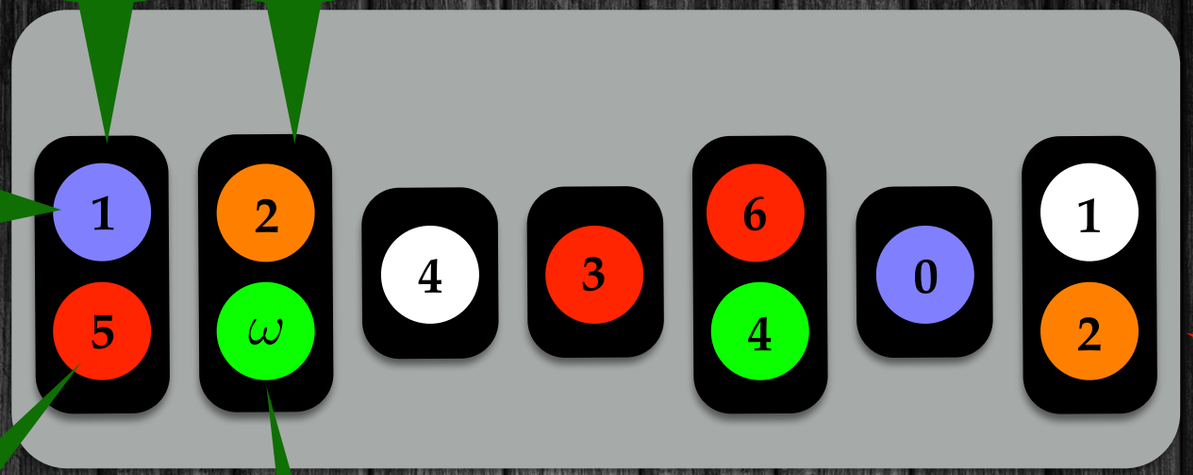
signature:
"sequence of multisets of colored natural numbers"

Timed Pet… Signatures

cmax=6

multis multiset

- integer part = 1
- place:



s: signature

- integer part = 5
- place:

- integer part > cmax
- place:

signature:
"sequence of multisets of colored natural numbers"

Timed Pet… Signatures

cmax=6

multis… mult… multi… multiset multiset multiset multiset

- integer part = 1
- place:

1 5 | 2 ω | 4 | 3 | 6 4 | 0 | 1 2

s: signature

- integer part = 5
- place:

- integer part > cmax
- place:

- integer part = 2
- place:

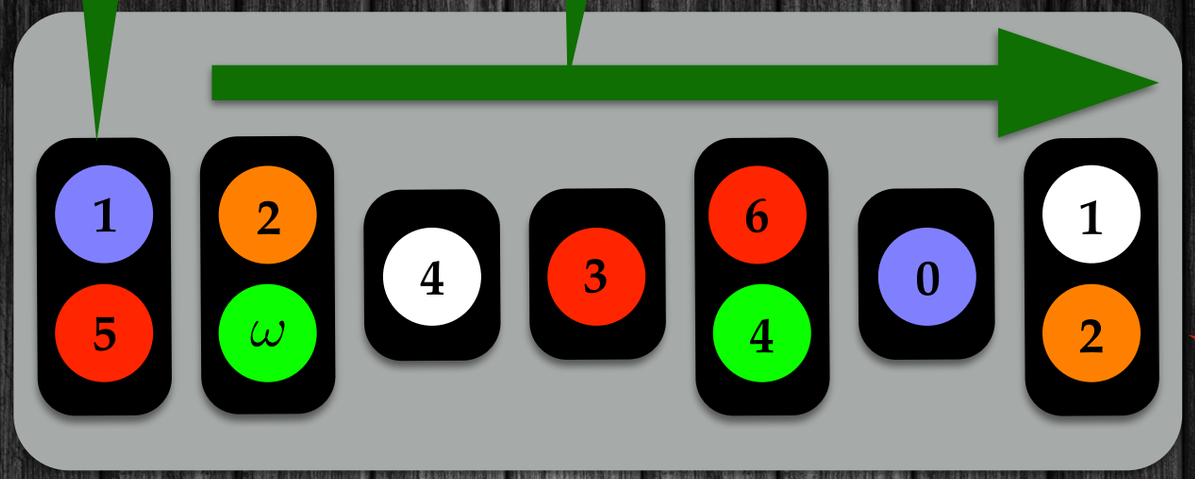
signature:
"sequence of multisets of colored natural numbers"

Timed Pet… Signatures

zero fractional part

increasing fractional parts

cmax=6



s: signature

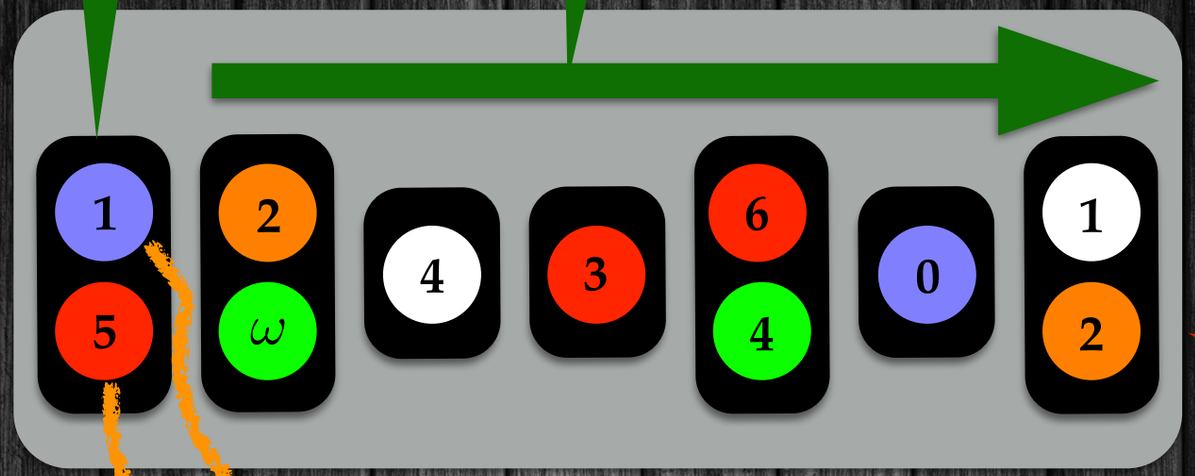
signature:
"sequence of multisets of colored natural numbers"

Timed Pet… Signatures

zero fractional part

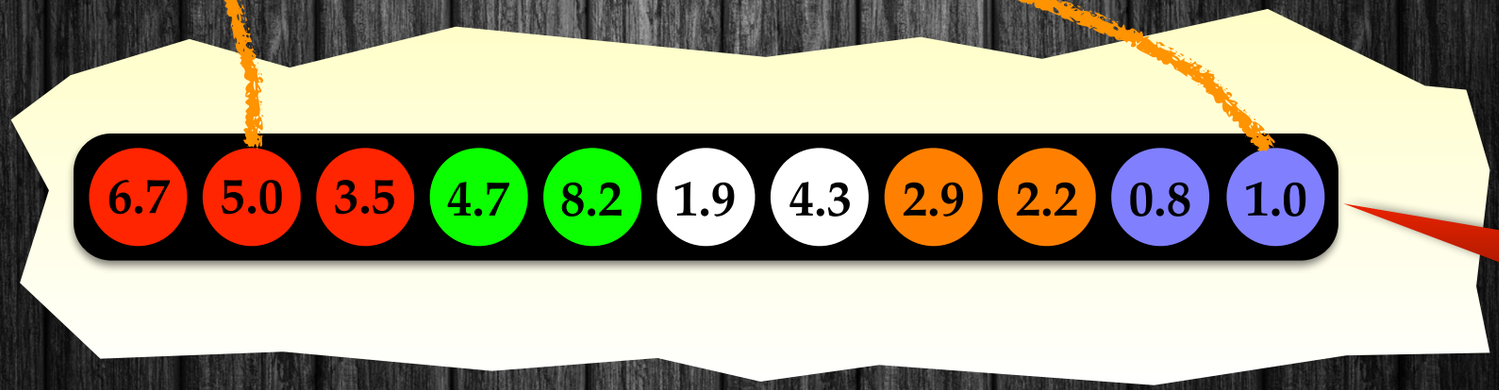
increasing fractional parts

cmax=6



s: signature

$sig(c) = s$



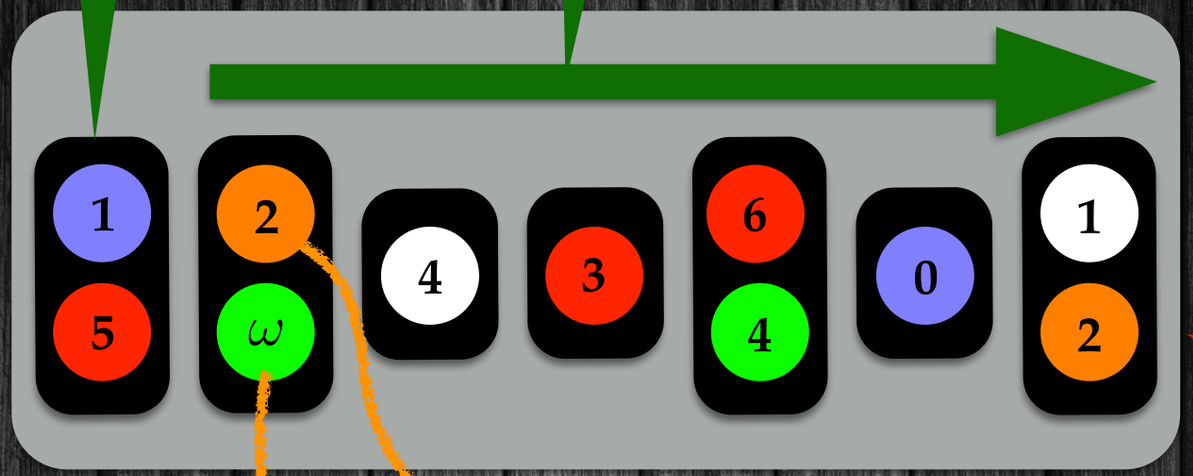
c: configuration

Timed Pet… Signatures

zero fractional part

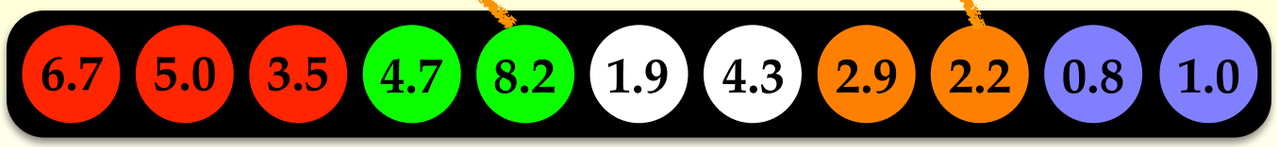
increasing fractional parts

cmax=6



s: signature

$sig(c) = s$



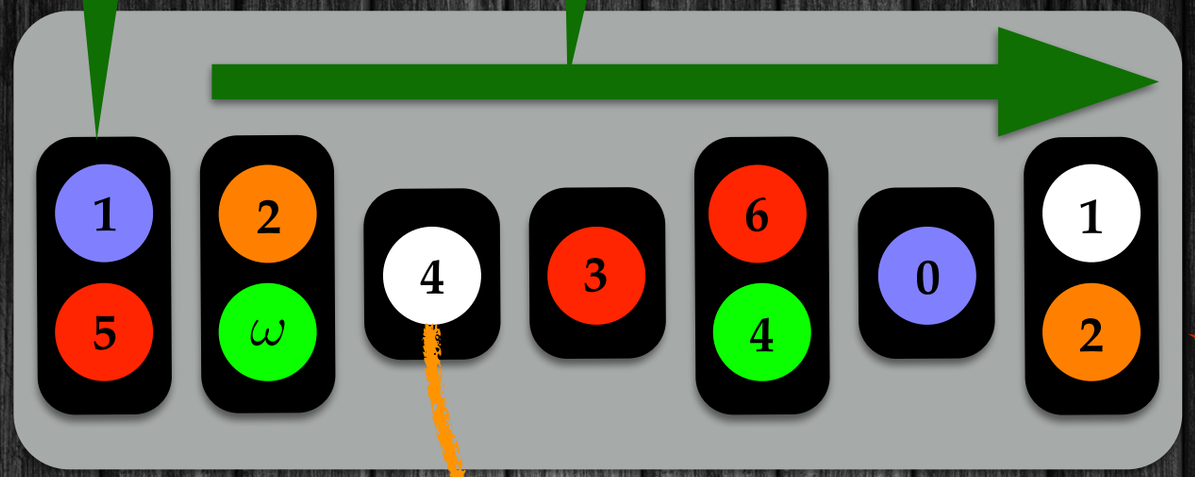
c: configuration

Timed Pet… Signatures

zero fractional part

increasing fractional parts

cmax=6



s: signature

$sig(c) = s$

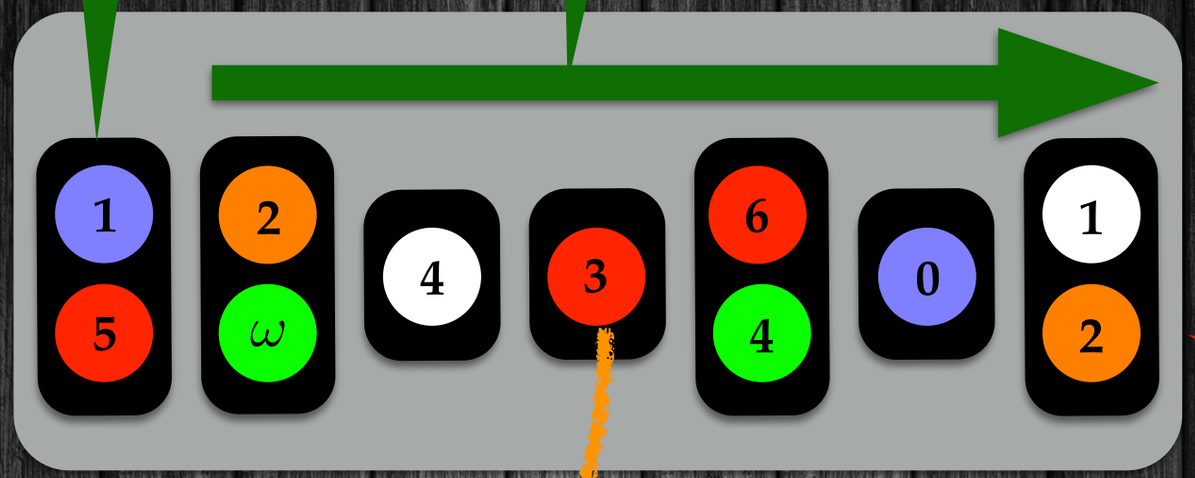
c: configuration

Timed Pet… Signatures

zero fractional part

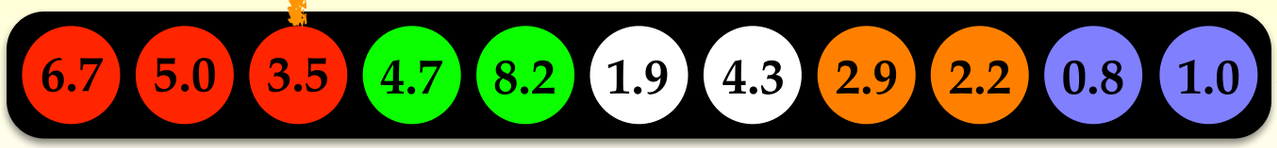
increasing fractional parts

cmax=6



s: signature

$sig(c) = s$



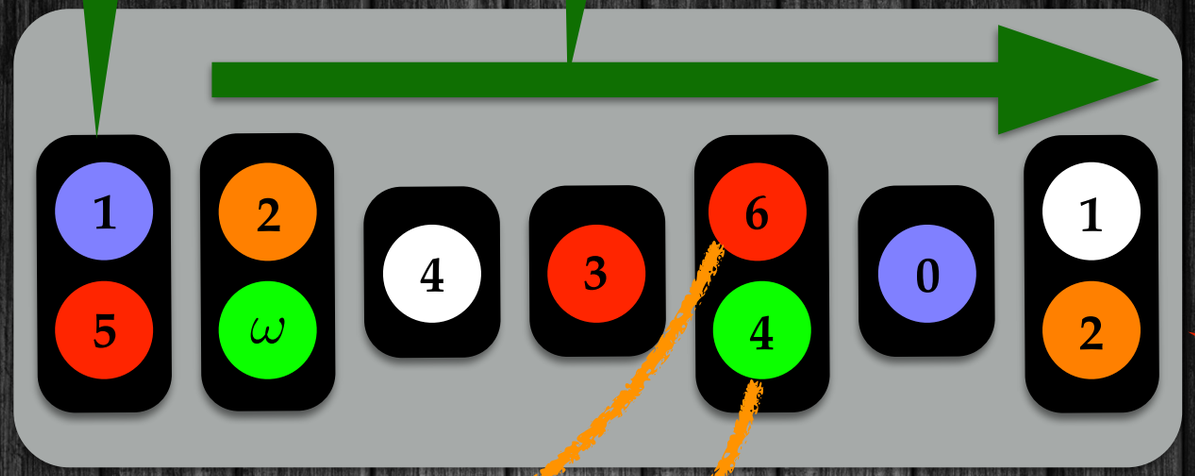
c: configuration

Timed Pet… Signatures

zero fractional part

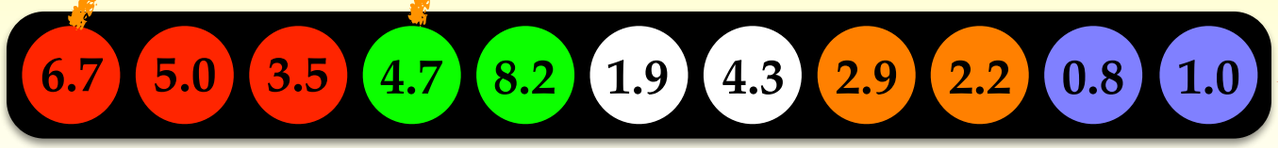
increasing fractional parts

cmax=6



s: signature

$sig(c) = s$



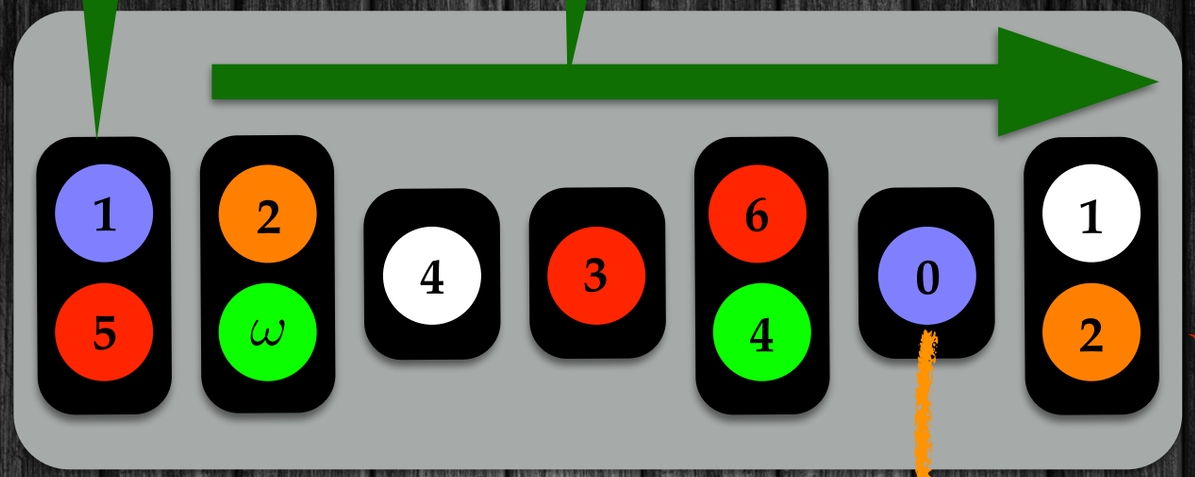
c: configuration

Timed Pet… Signatures

zero fractional part

increasing fractional parts

cmax=6



s: signature

$sig(c) = s$

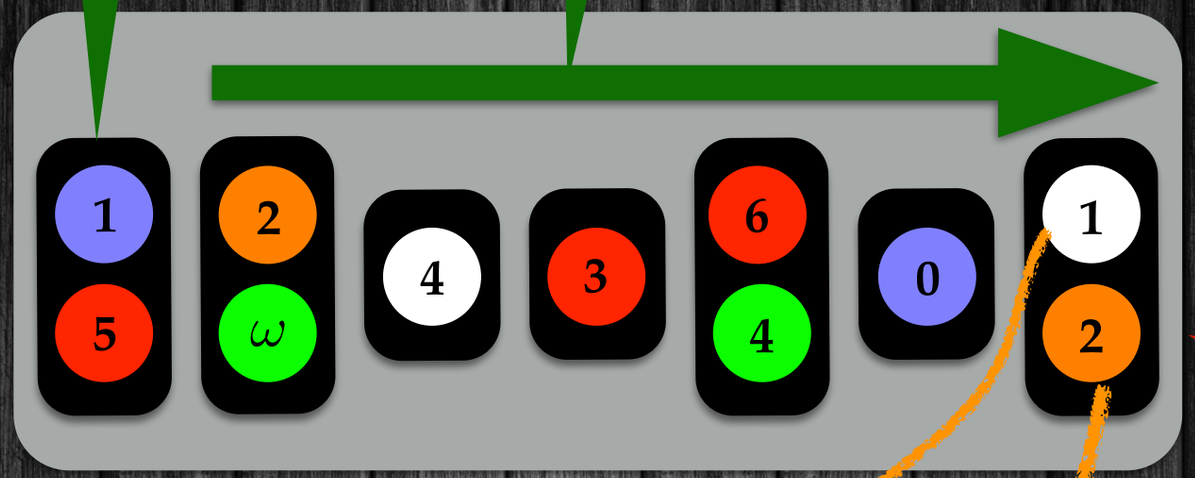
c: configuration

Timed Pet… Signatures

zero fractional part

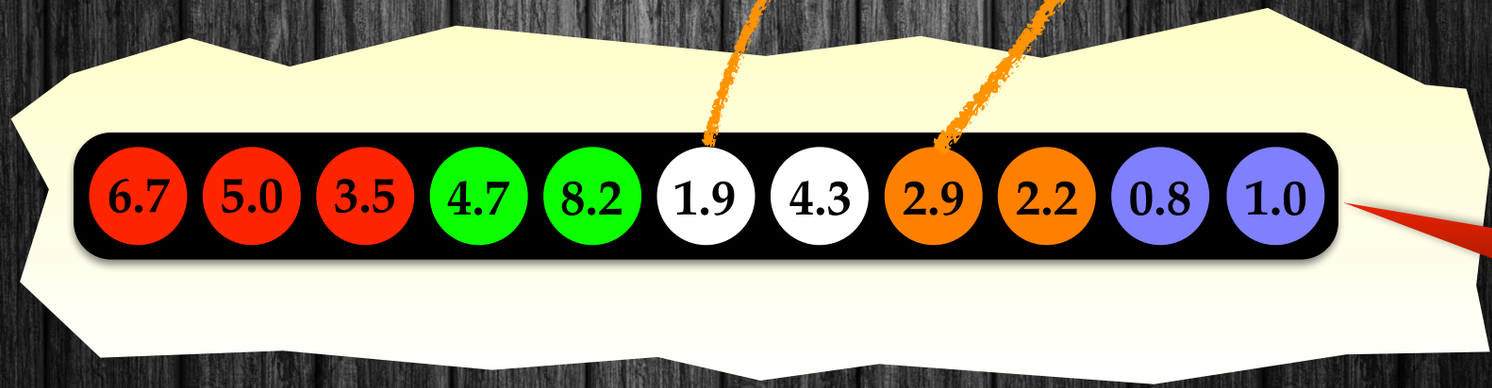
increasing fractional parts

cmax=6



s: signature

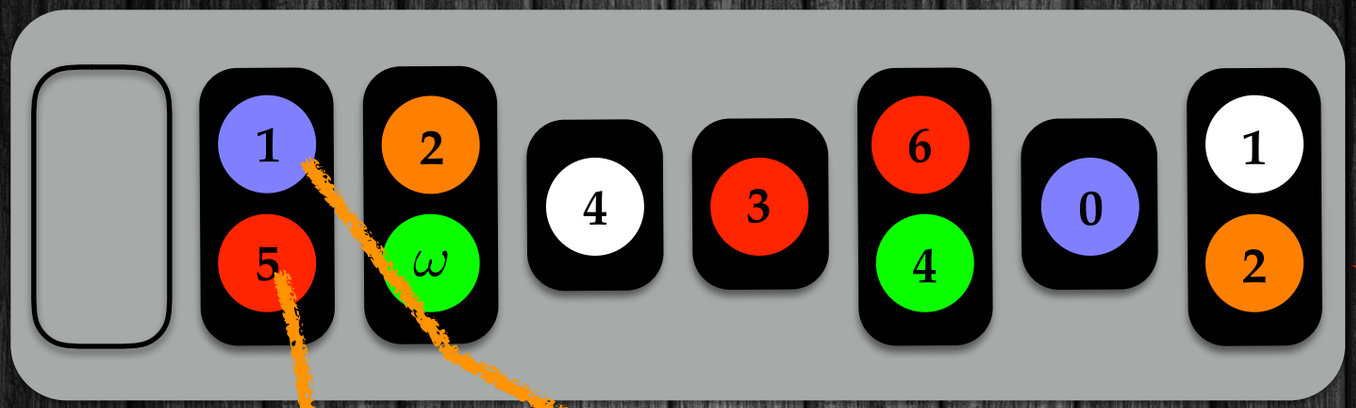
$sig(c) = s$



c: configuration

Timed Pe… Signatures

cmax=6



s: signature

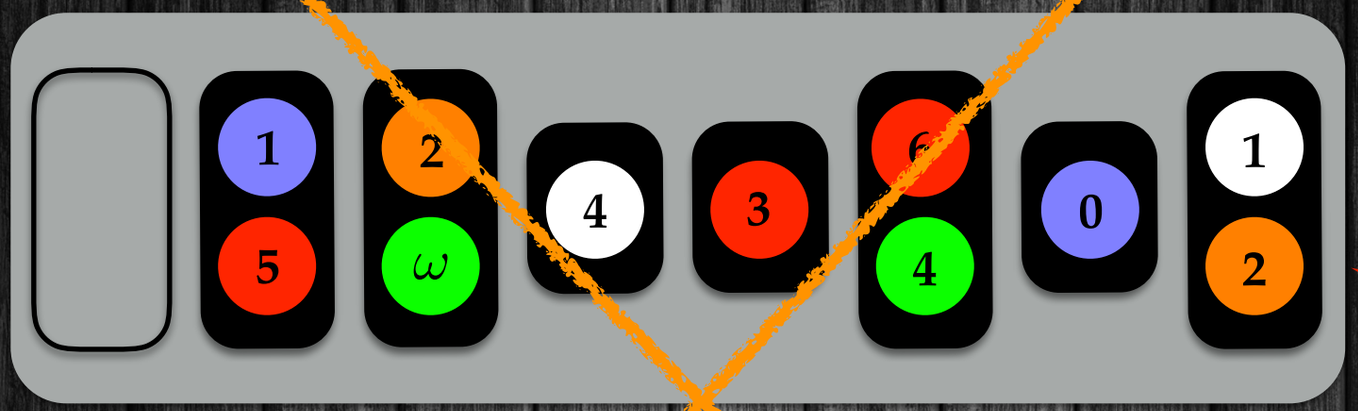
$sig(c) = s$

c: configuration

Timed Pe

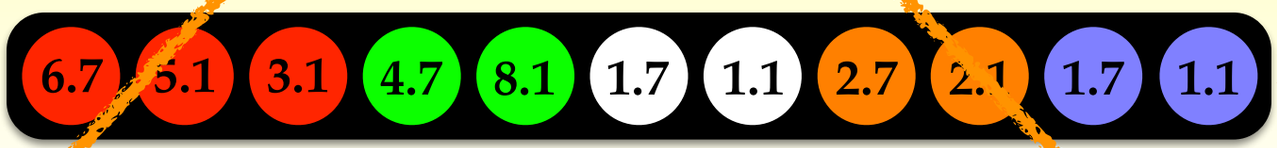
Signatures

cmax=6



s:
signature

$sig(c) \neq s$

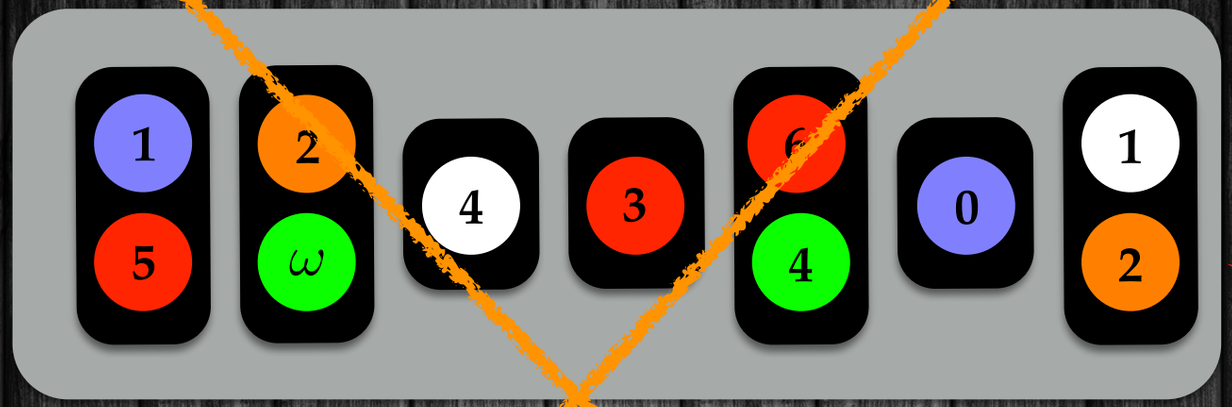


c:
configuration

Timed Pet

Signatures

cmax=6



s:
signature

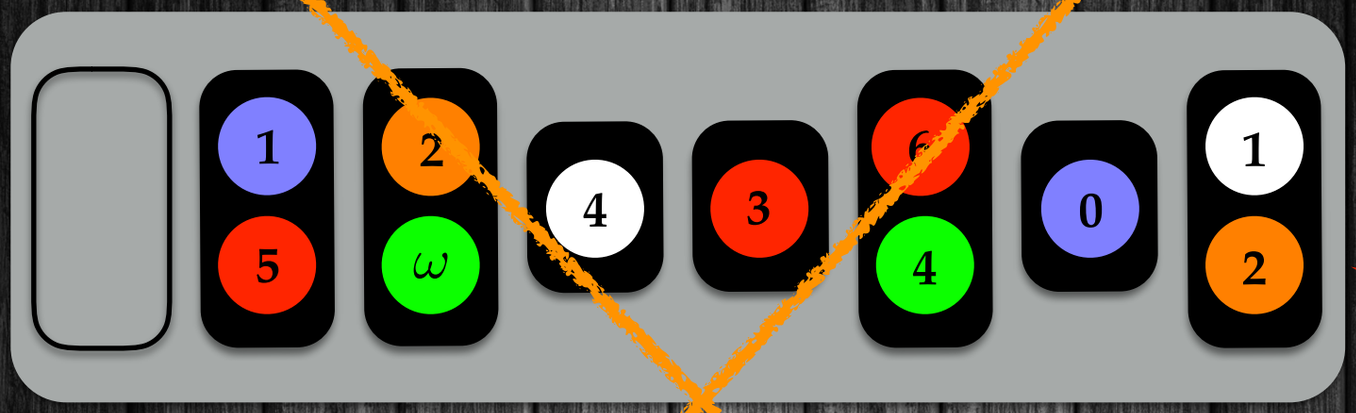
$sig(c) \neq s$

c:
configuration

Timed Pe

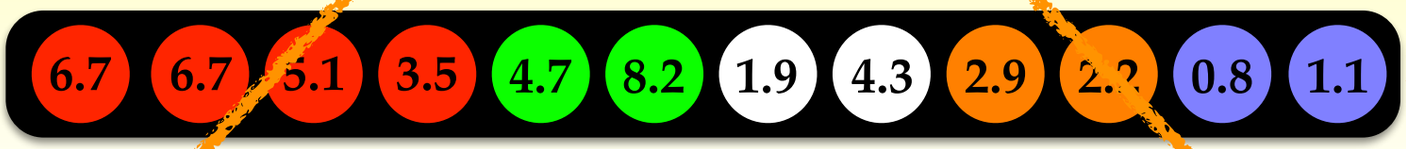
Signatures

cmax=6



s:
signature

$sig(c) \neq s$

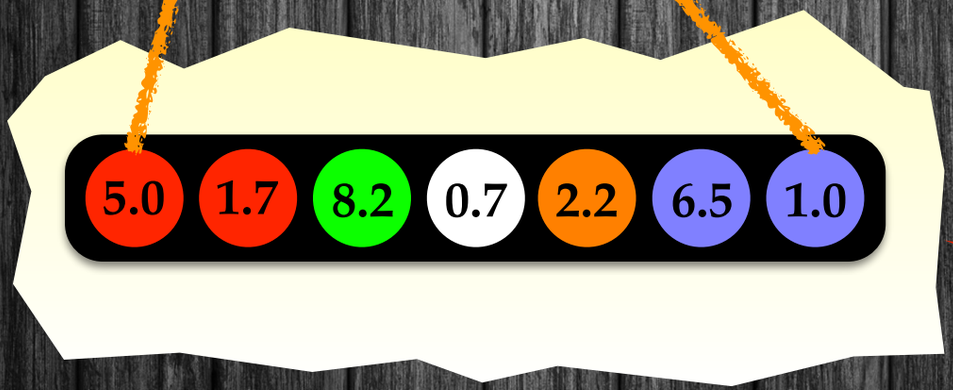


c:
configuration

Timed Pet… Signatures

s: signature

$sig(c) = s$



c: configuration

Timed Pet… Signatures

s:
signature

$sig(c) = s$

c:
configuration

Timed Pet… Signatures

s:
signature

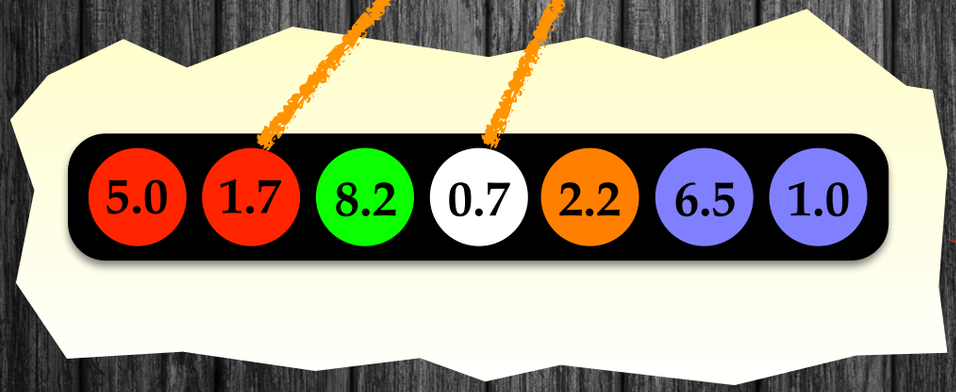
$sig(c) = s$

c:
configuration

Timed Pet… Signatures

s: signature

$sig(c) = s$

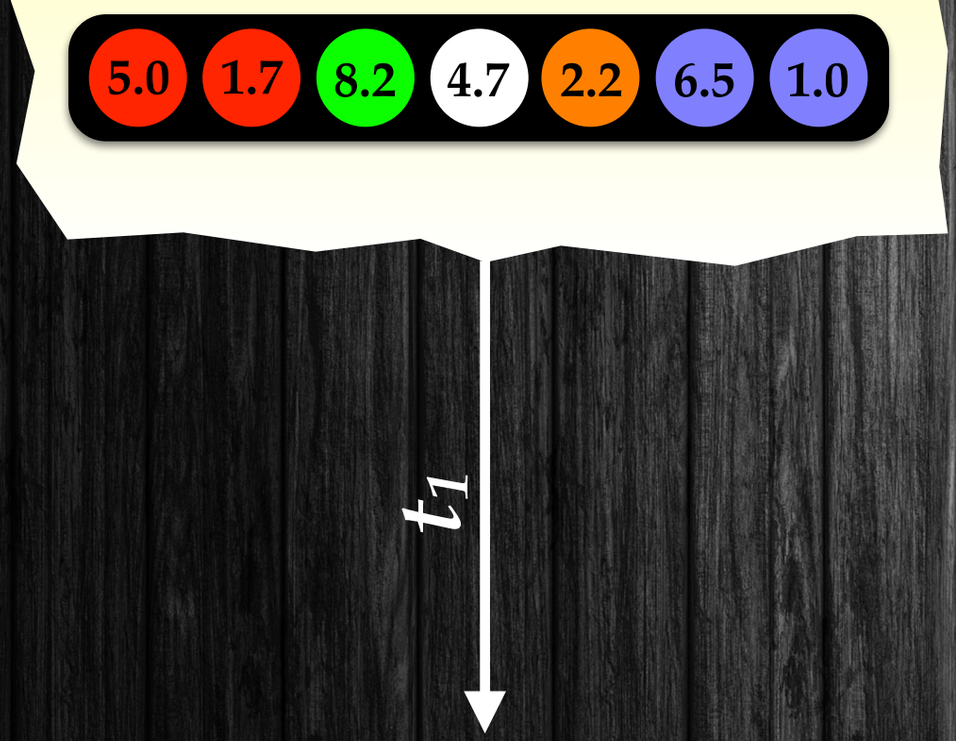


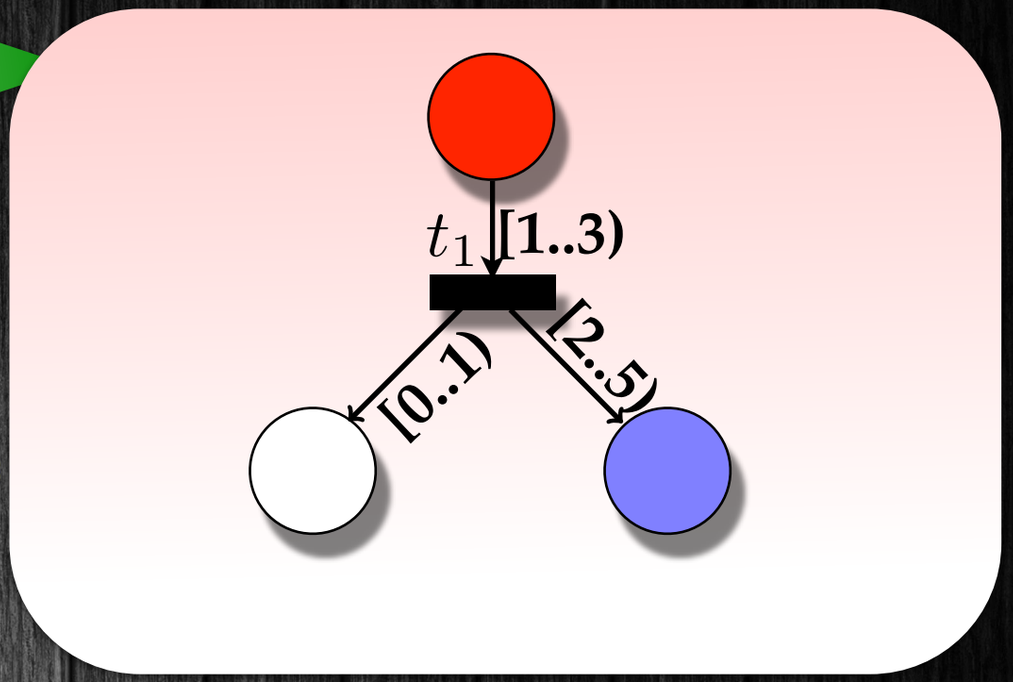
c: configuration

Timed Pet… Signatures

sig(c)=s:

c and s are "bisimilar"

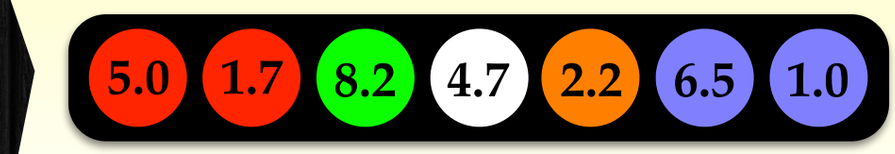




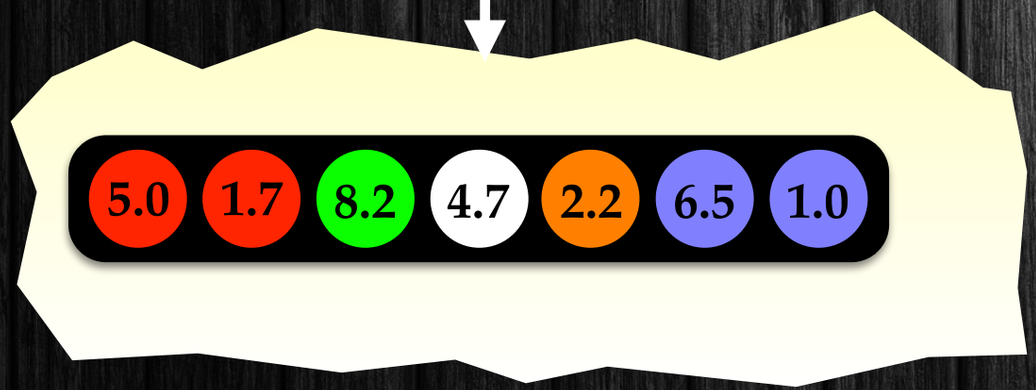
Timed Pe… Signatures

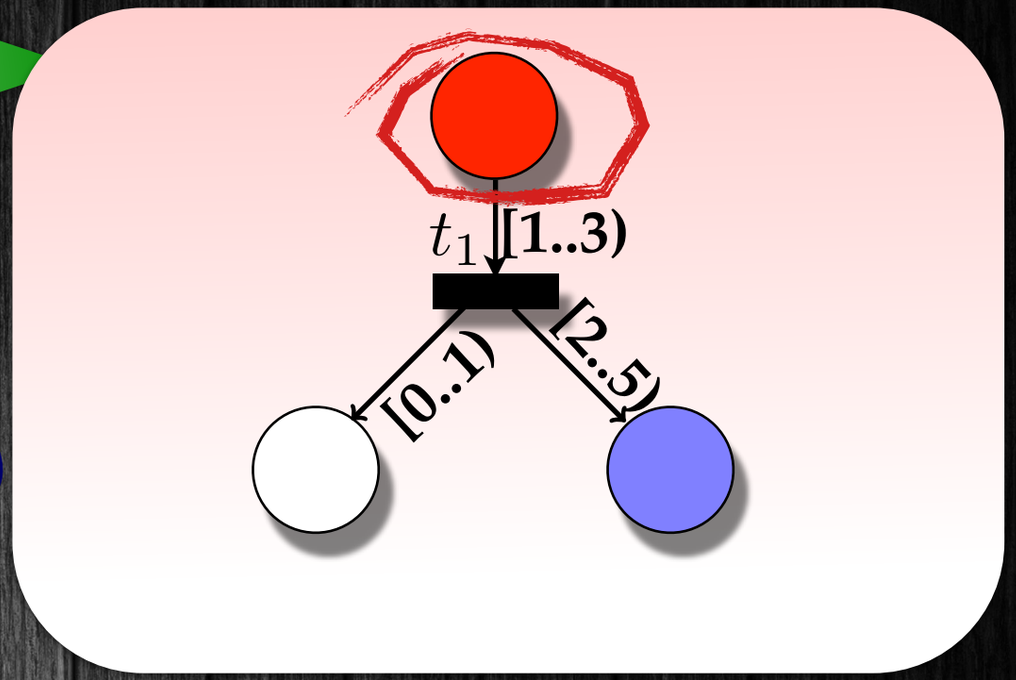
sig(c)=s:

c and s are "bisimilar"



t_1



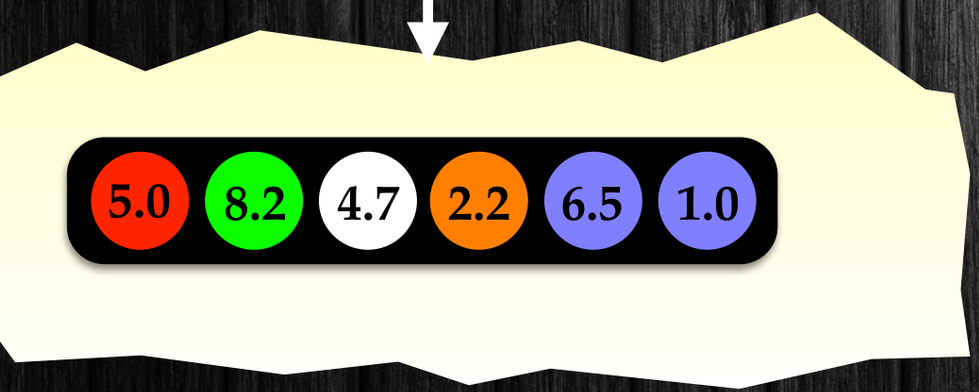


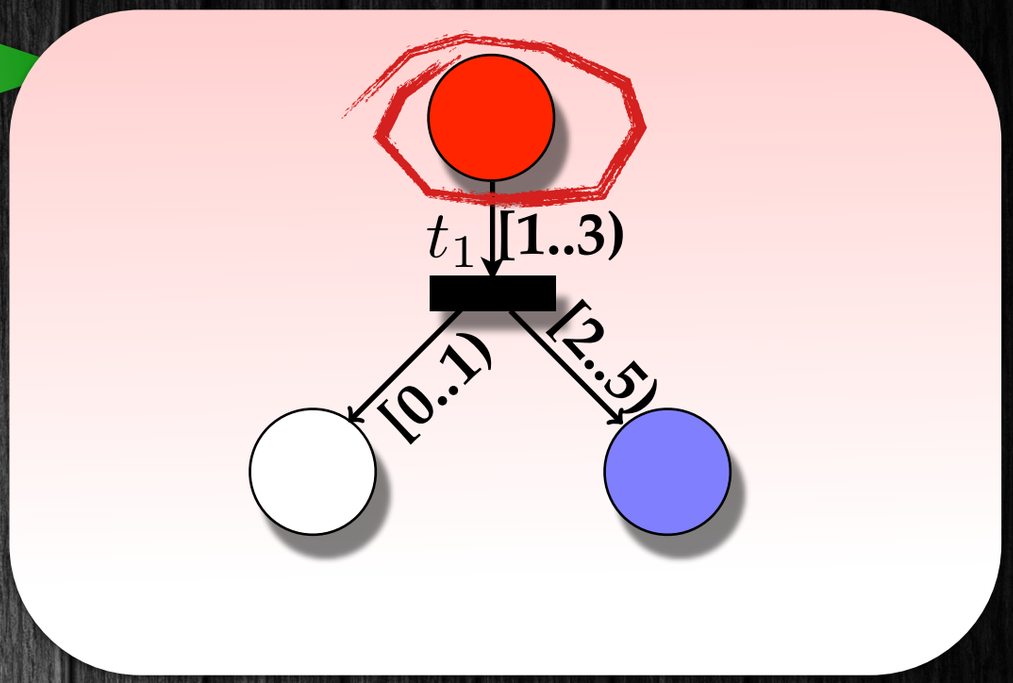
Timed Pe… Signatures

sig(c)=s:

c and s are "bisimilar"

t_1

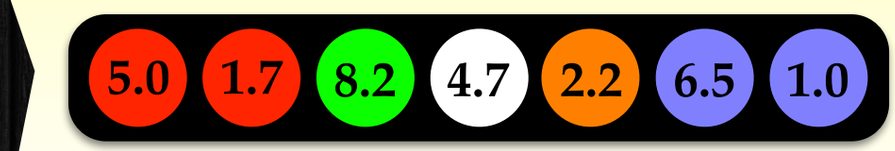




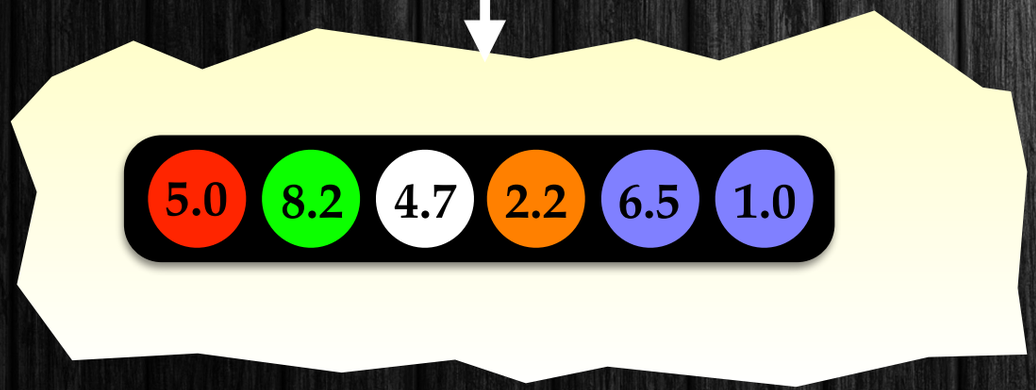
Timed Pe… Signatures

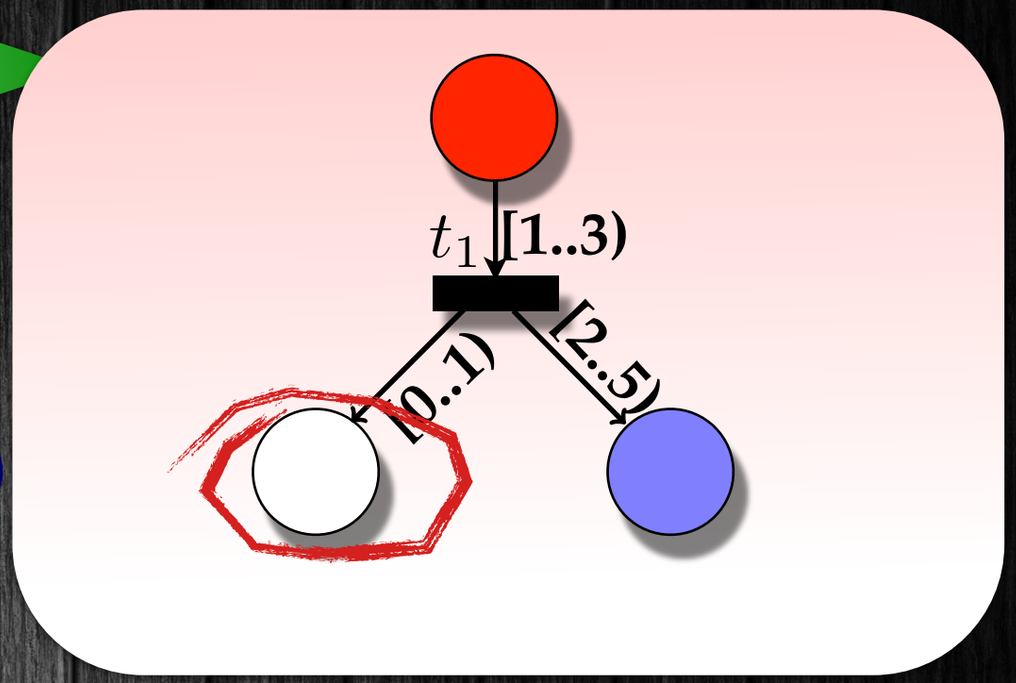
sig(c)=s:

c and s are "bisimilar"



t_1

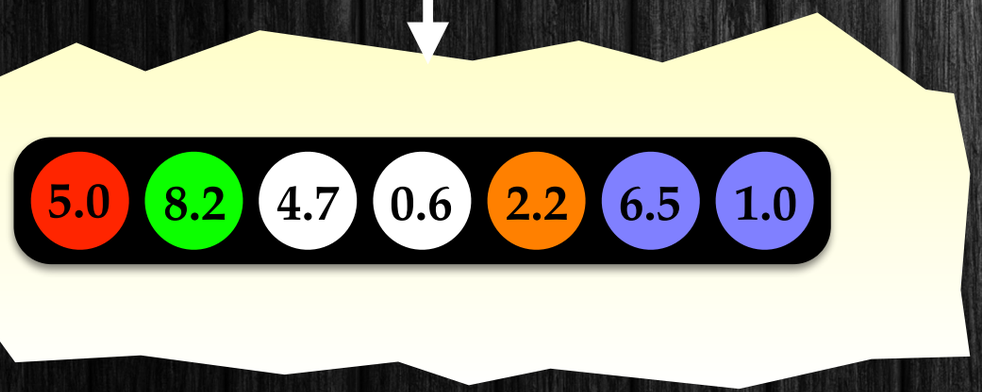


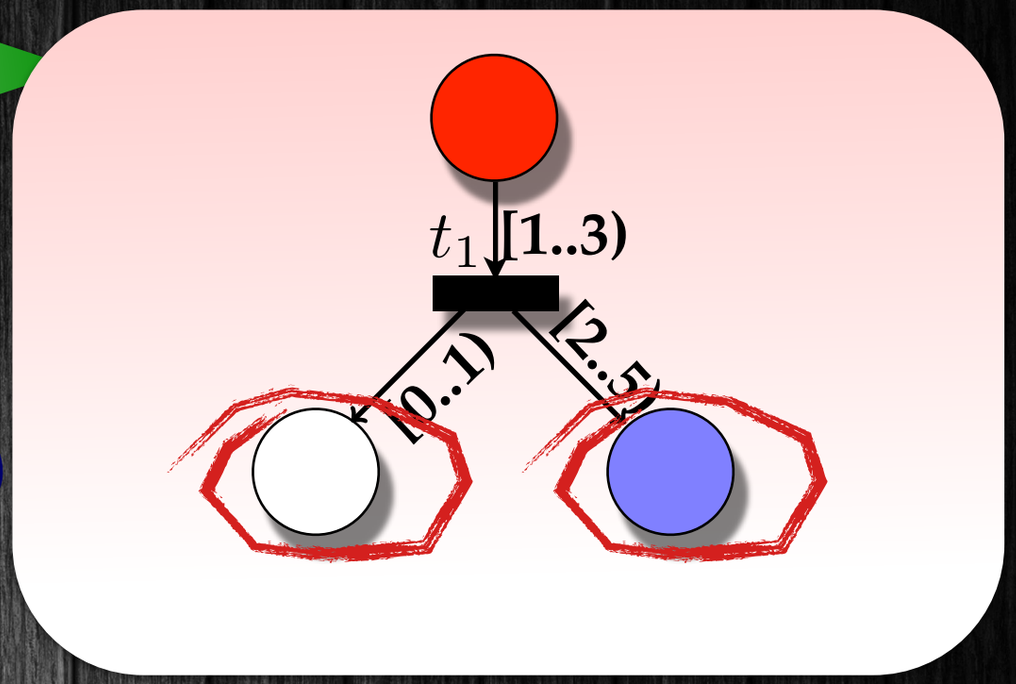


Timed Pet… Signatures

sig(c)=s:
c and s are "bisimilar"

t_1

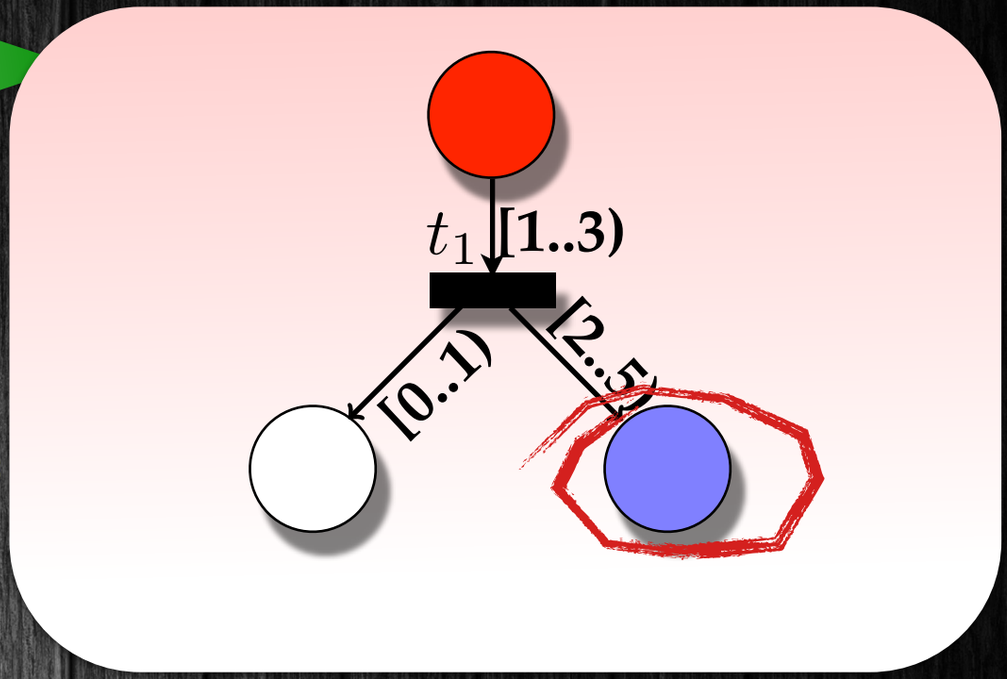


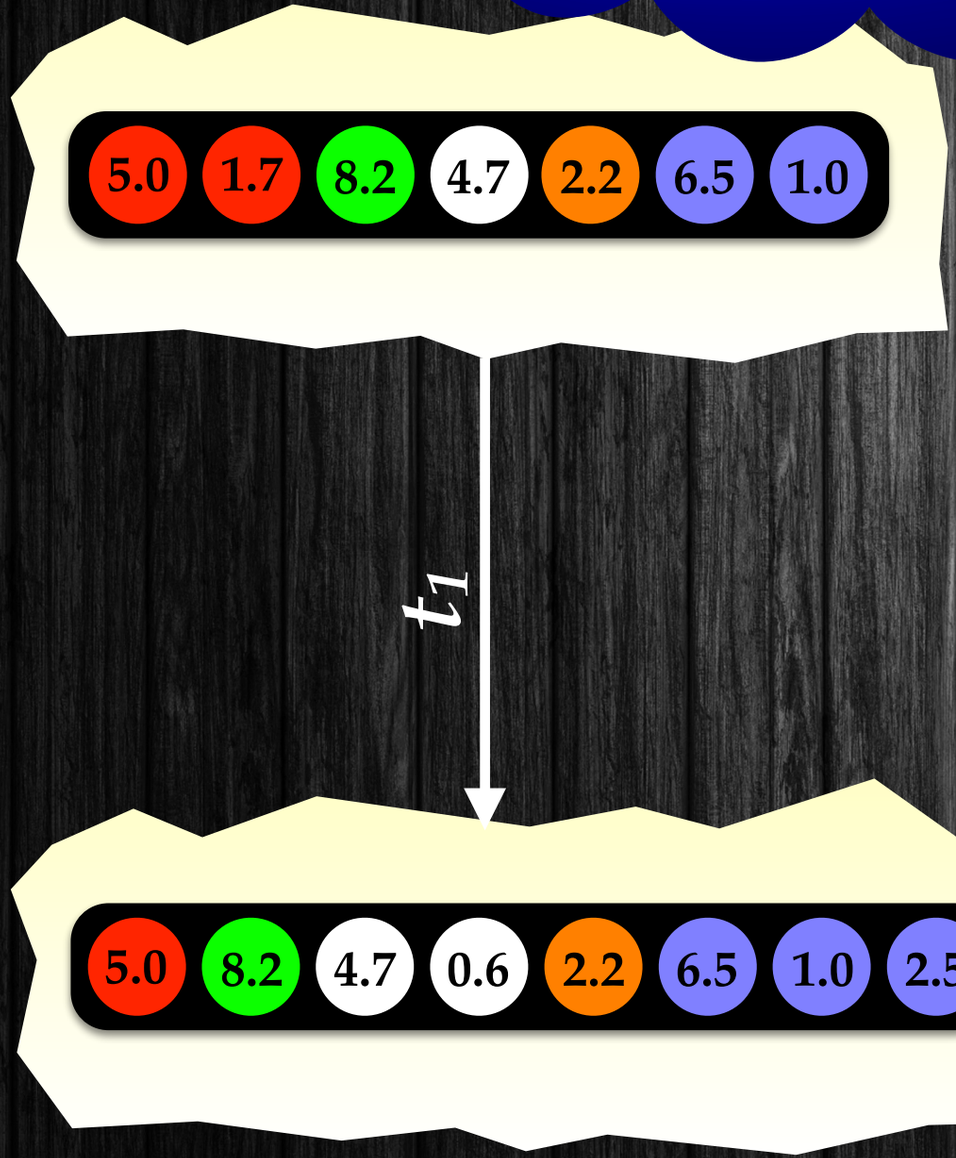


Timed Pe… Signatures

sig(c)=s:

c and s are "bisimilar"

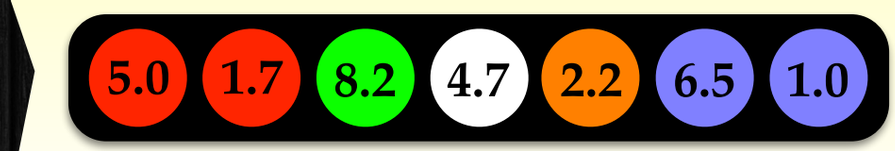




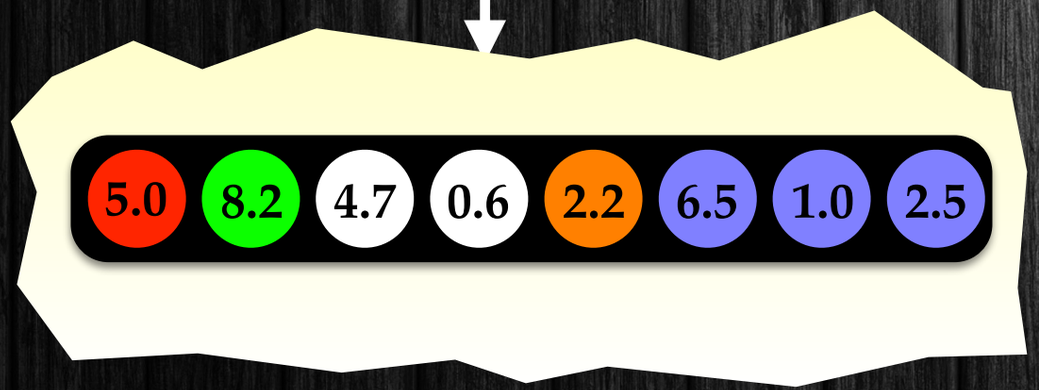
Timed Pe… Signatures

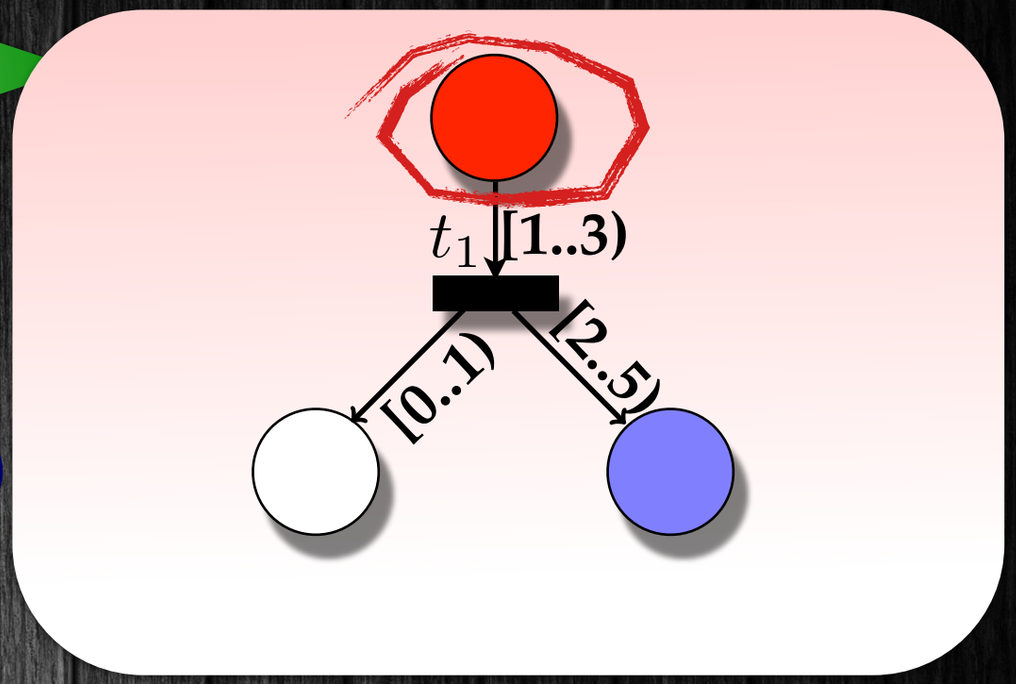
sig(c)=s:

c and s are "bisimilar"

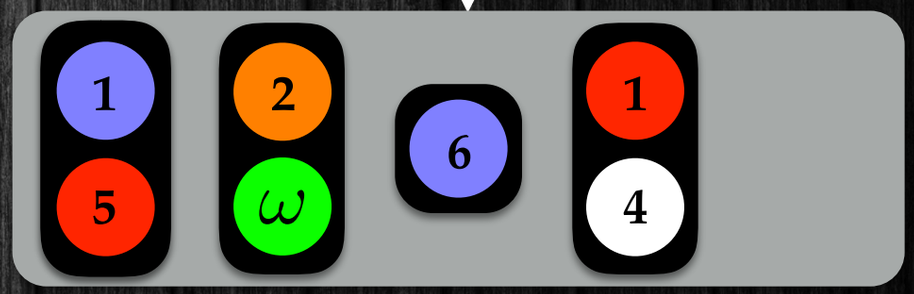


t_1





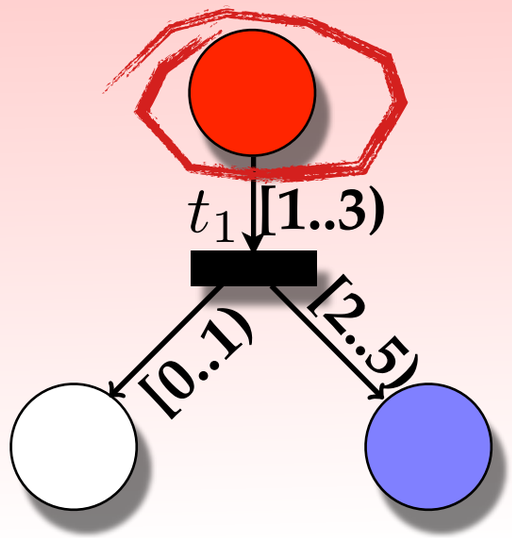
t_1

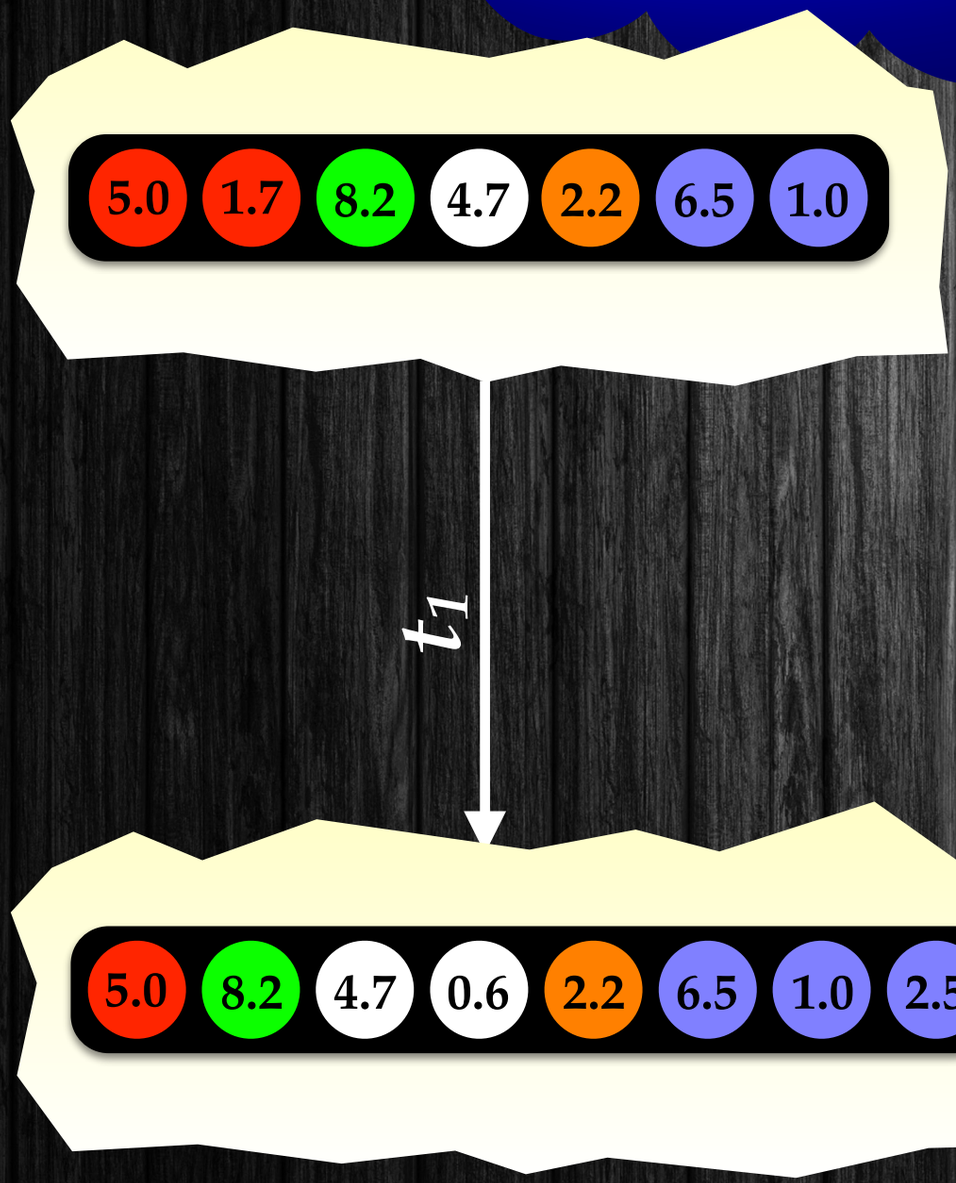


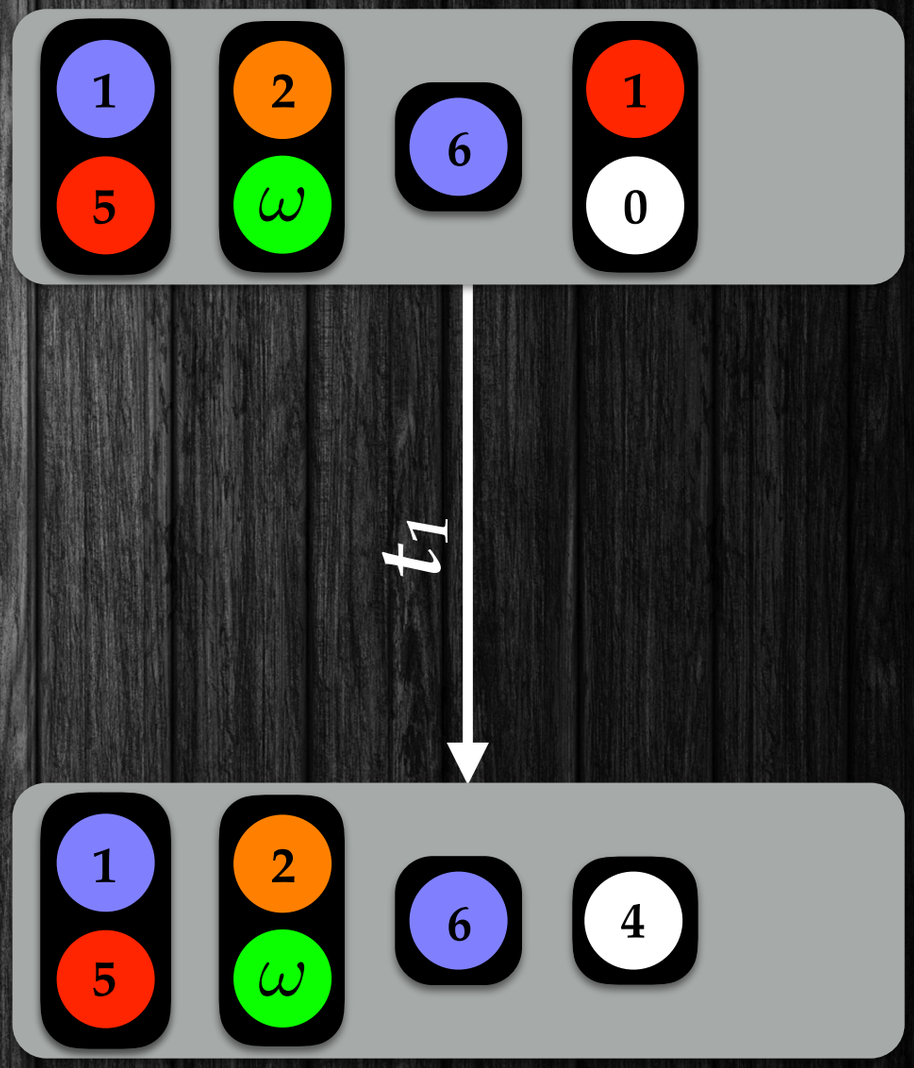
Timed Pe… Signatures

sig(c)=s:

c and s are "bisimilar"



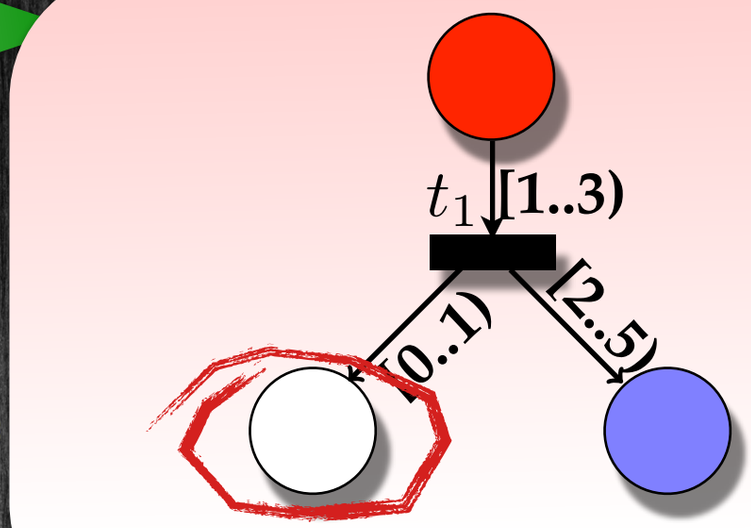


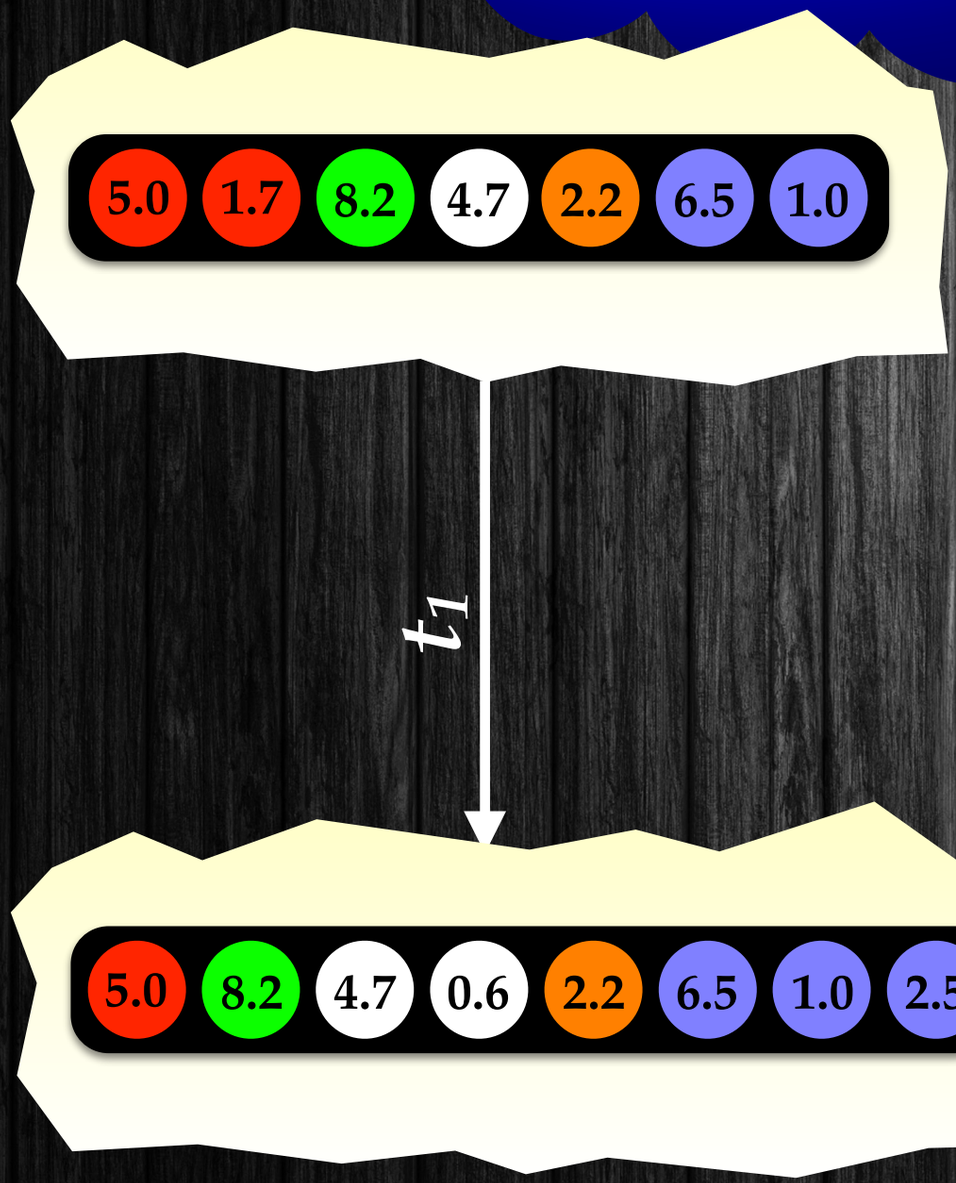


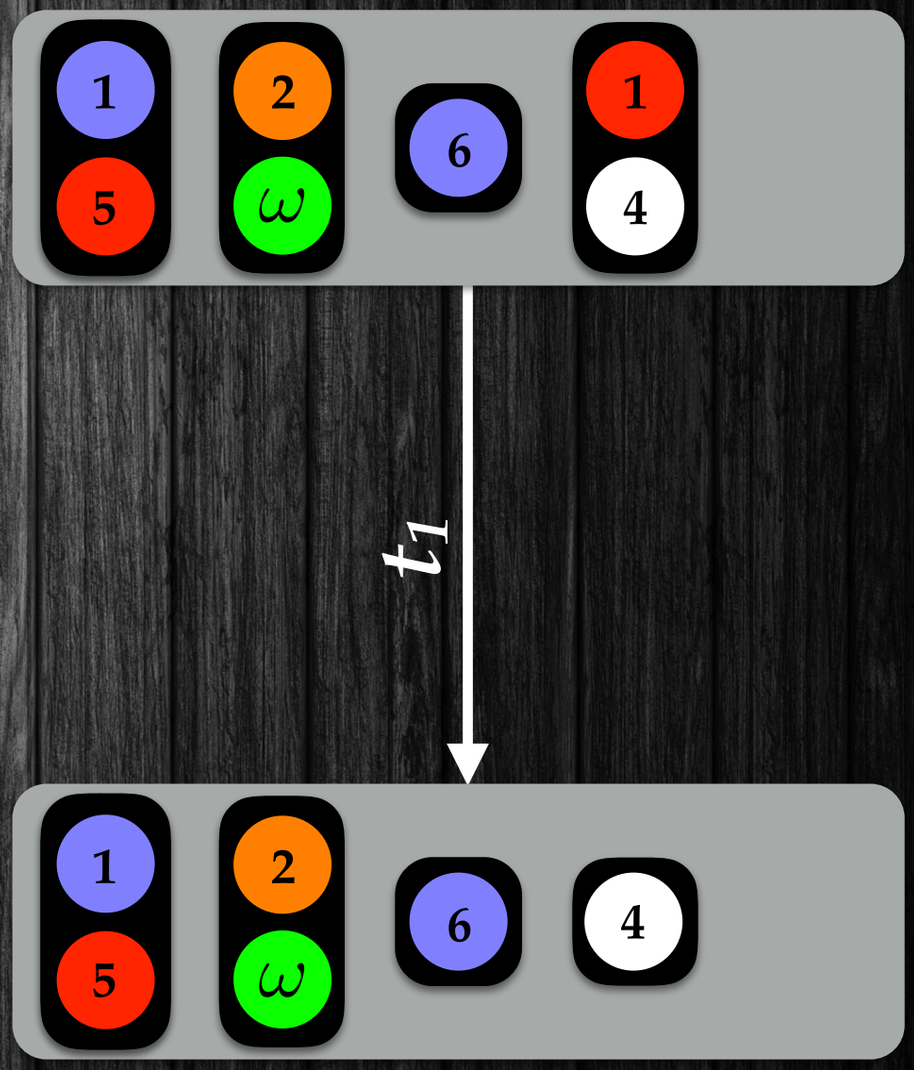
Timed Pe… Signatures

sig(c)=s:

c and s are "bisimilar"



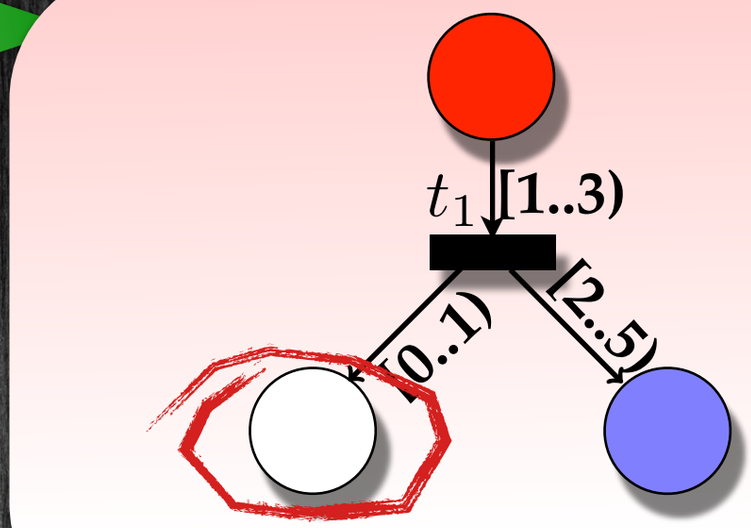


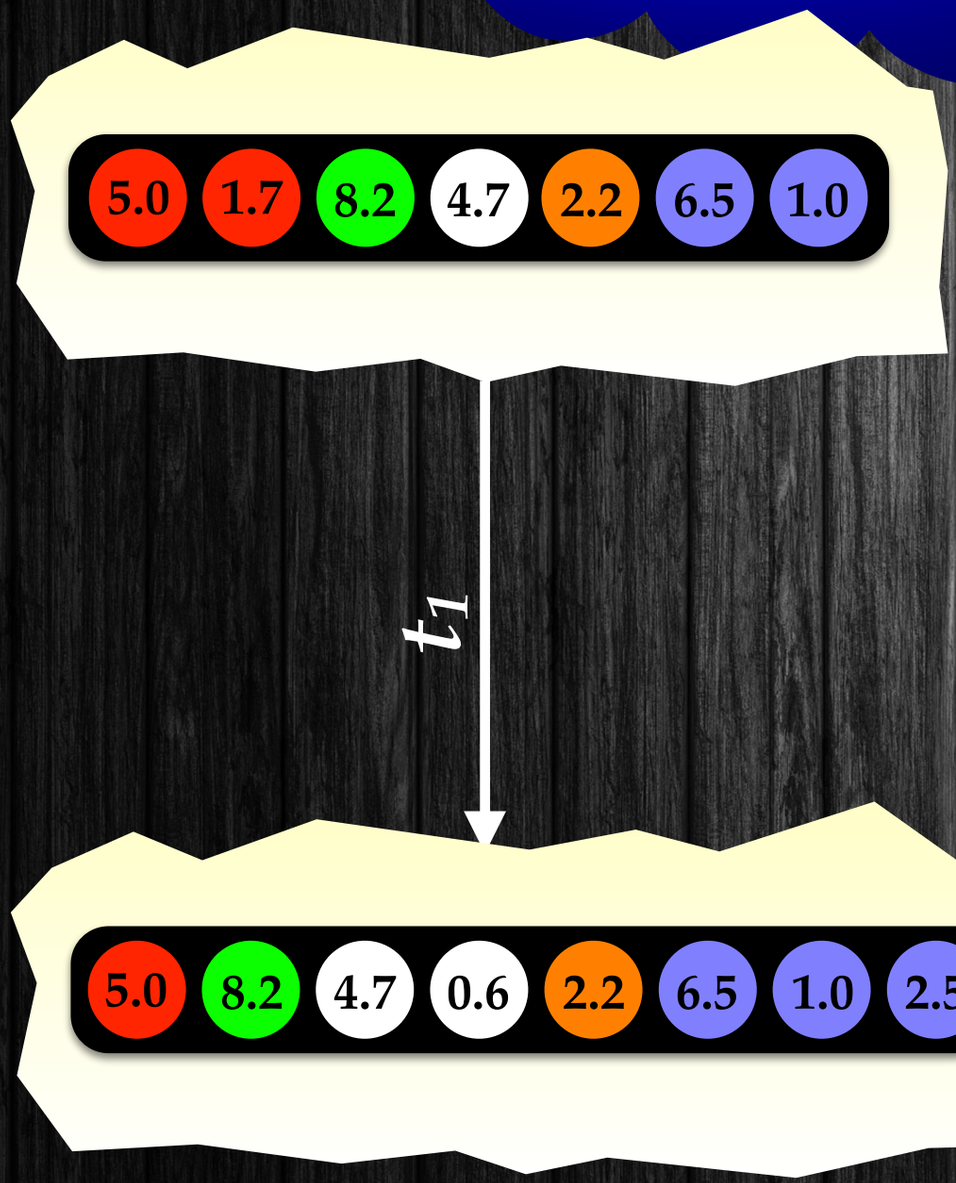


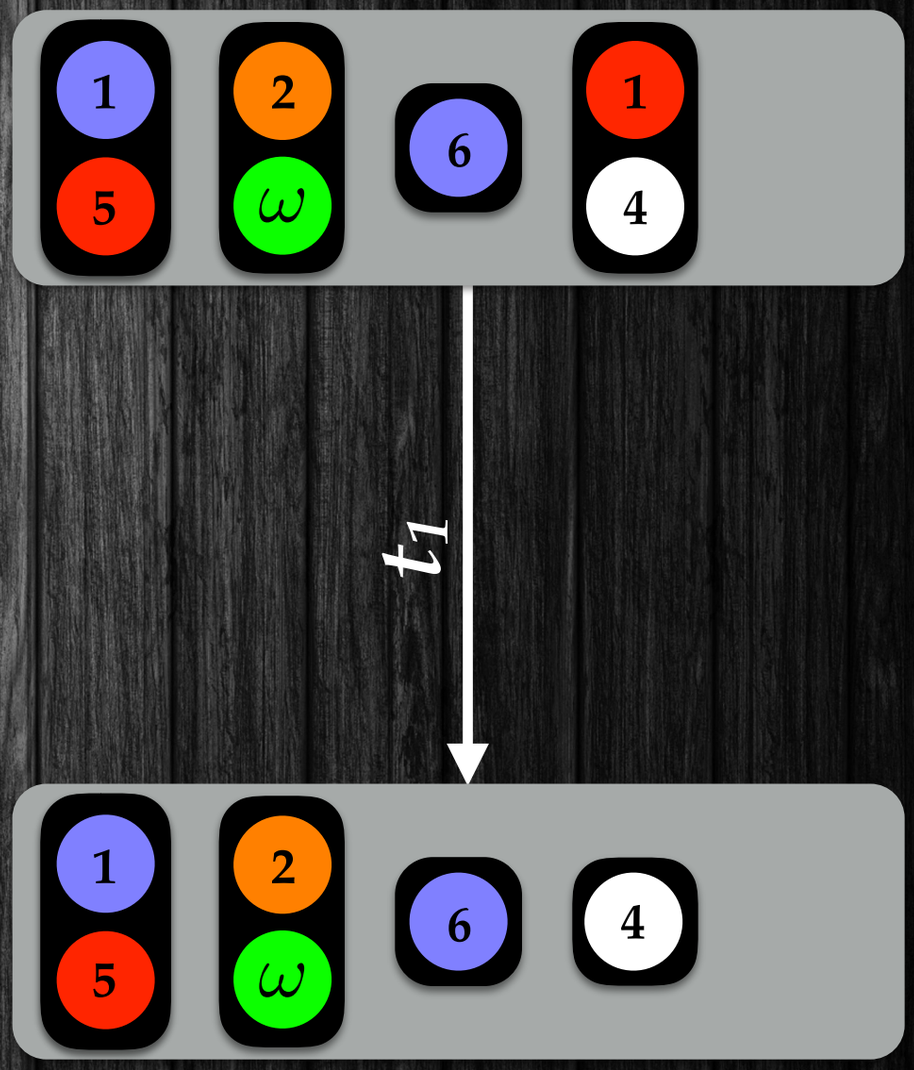
Timed Pe… Signatures

sig(c)=s:

c and s are "bisimilar"



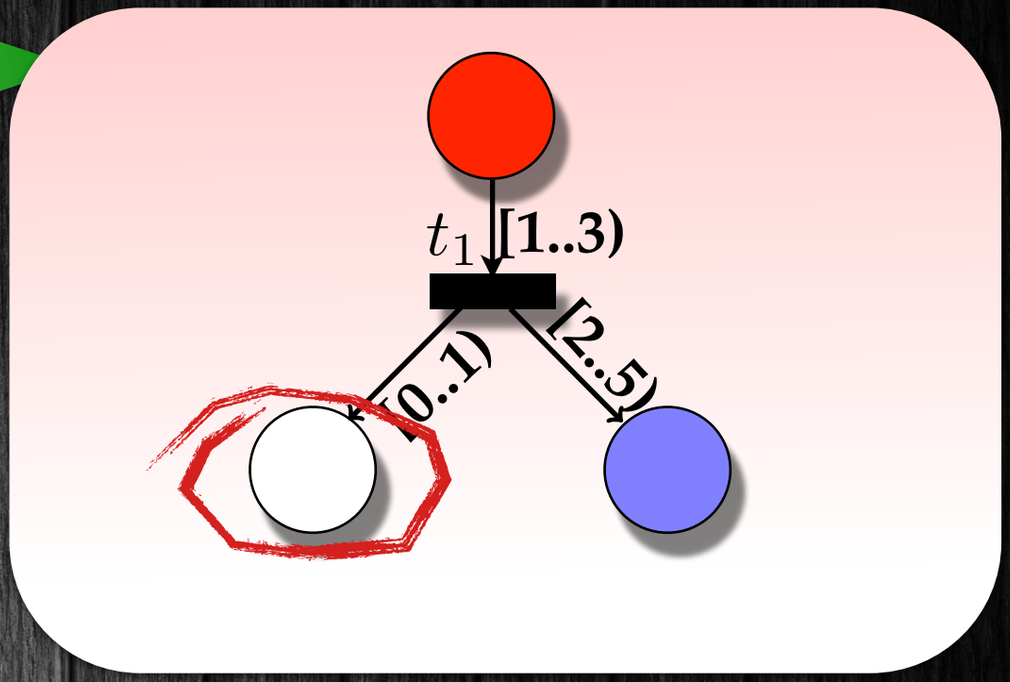


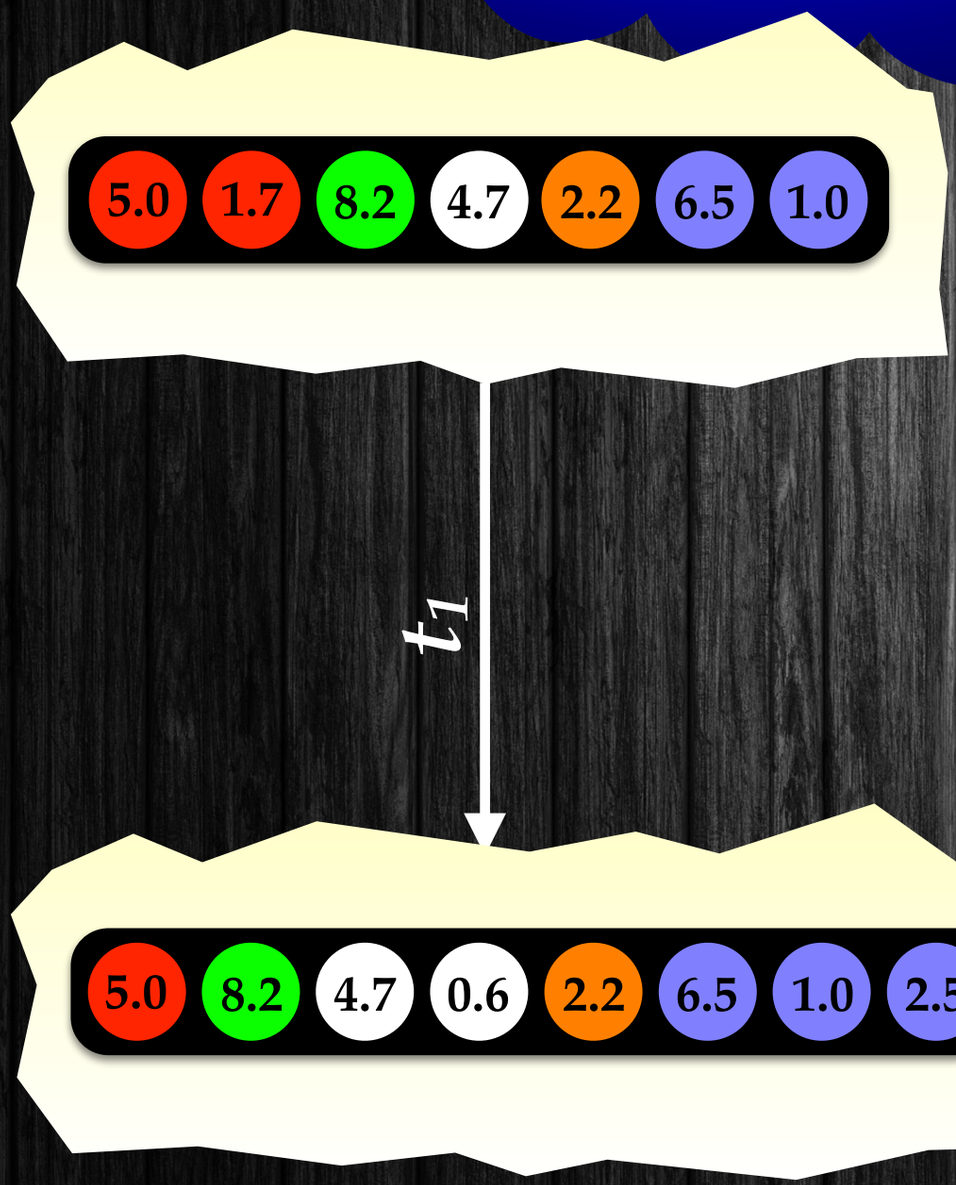


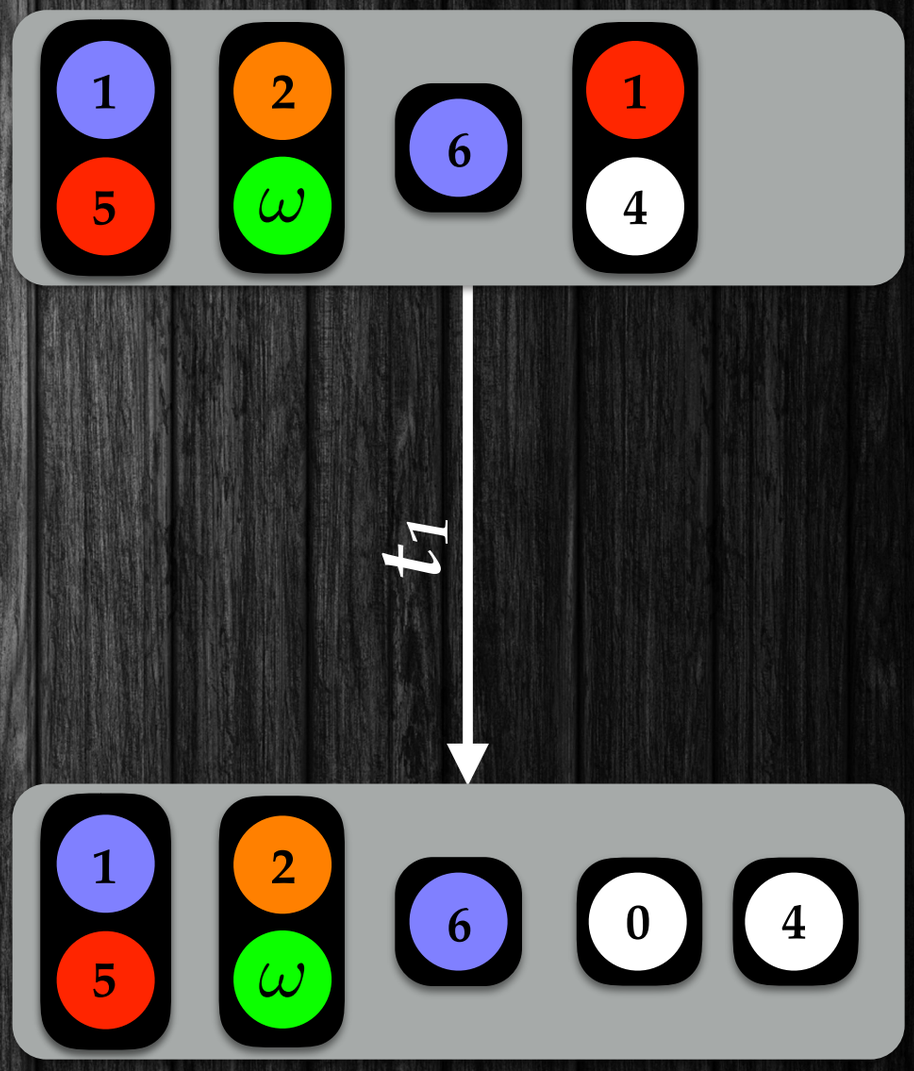
Timed Pe… Signatures

sig(c)=s:

c and s are "bisimilar"



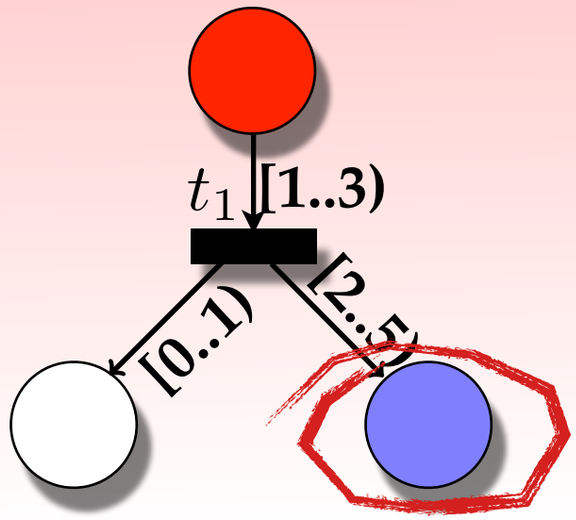




Timed Pe… Signatures

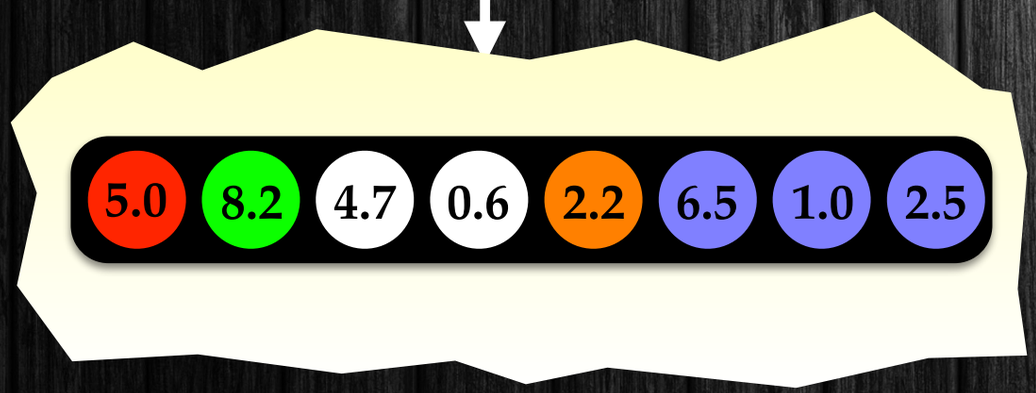
sig(c)=s:

c and s are "bisimilar"



t_1

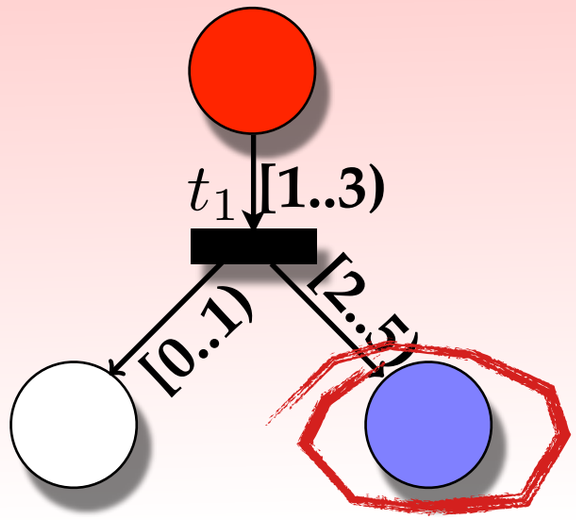
t_1

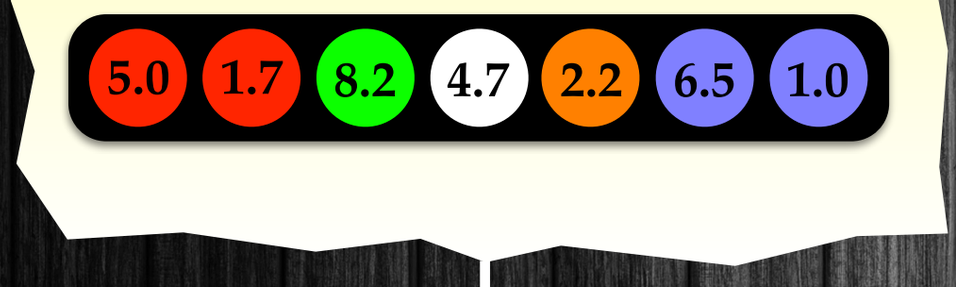


Timed Pe… Signatures

sig(c)=s:

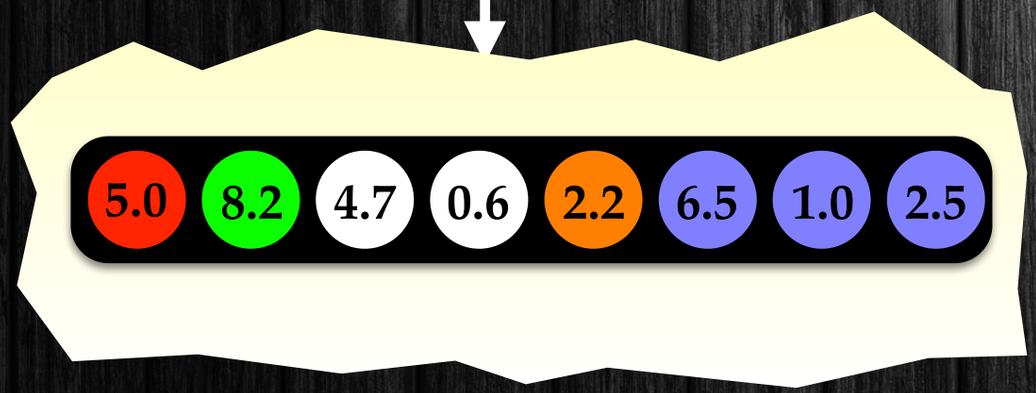
c and s are "bisimilar"





t_1

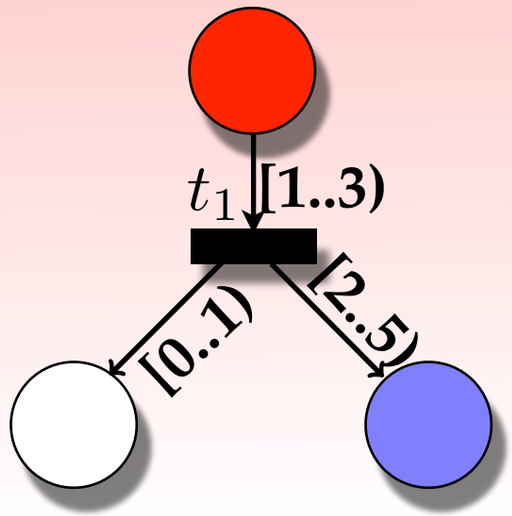
t_1

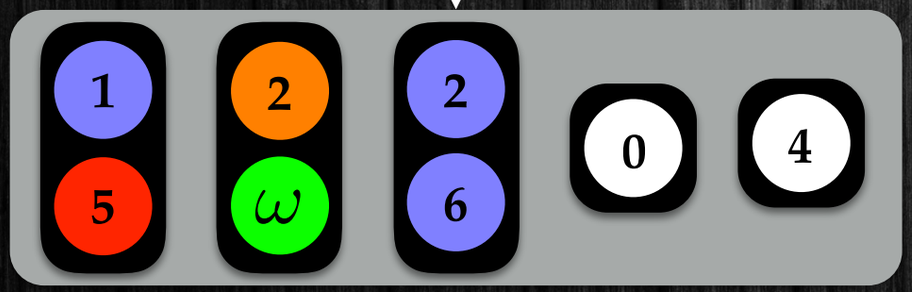


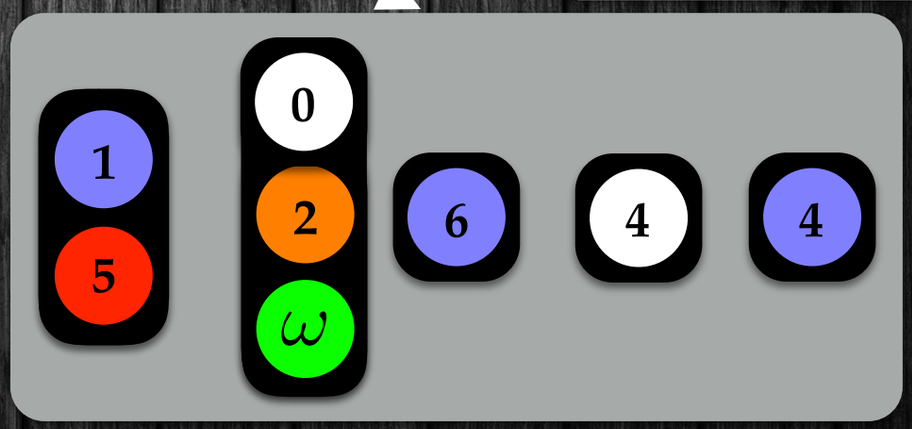
Timed Pe… Signatures

sig(c)=s:

c and s are "bisimilar"







t_1

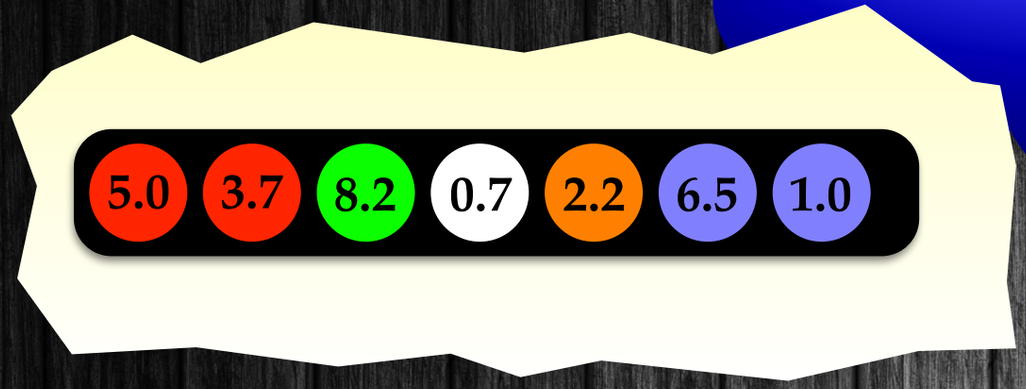
t_1

t_1

Timed Pe… Signa…

sig(c)=s:

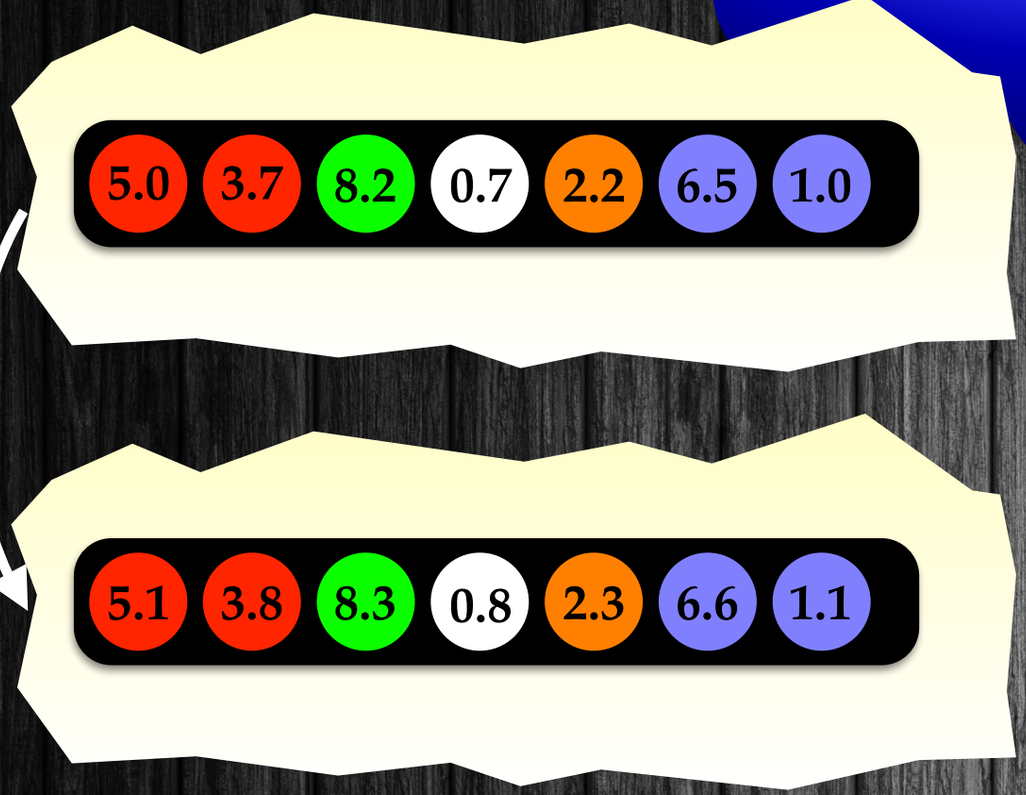
c and s are “bisimilar”



Timed Pe… Signa…

sig(c)=s:

c and s are "bisimilar"



Timed Pe… Signa…

sig(c)=s:

c and s are "bisimilar"

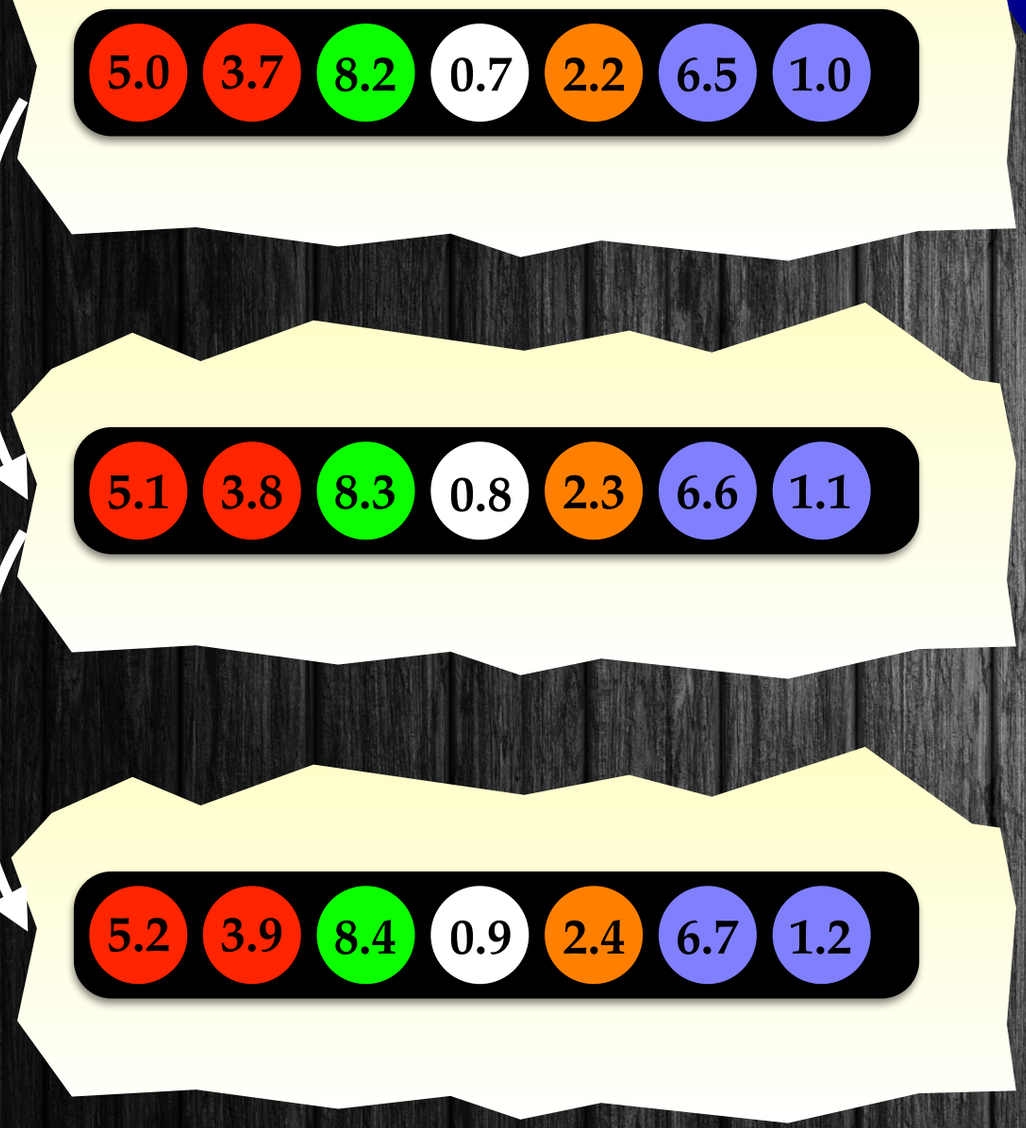
time=0.1

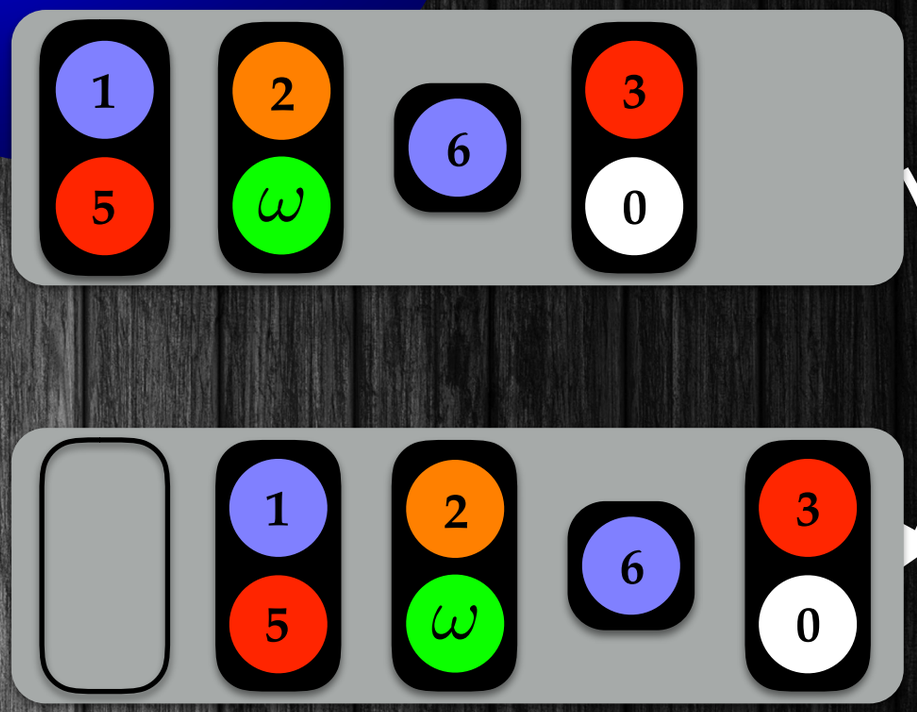
time

Timed Pet… Signal…

sig(c)=s:

c and s are "bisimilar"





Timed Pe… Signa…

sig(c)=s:
c and s are "bisimilar"

time=0.1

time=0.1

5.0 3.7 8.2 0.7 2.2 6.5 1.0

5.1 3.8 8.3 0.8 2.3 6.6 1.1

5.2 3.9 8.4 0.9 2.4 6.7 1.2

time

time

1 5 2 ω 6 3 0

1 5 2 ω 6 3 0

1 5 2 ω 6 3 0

Timed Pe… Signa…

sig(c)=s:

c and s are "bisimilar"

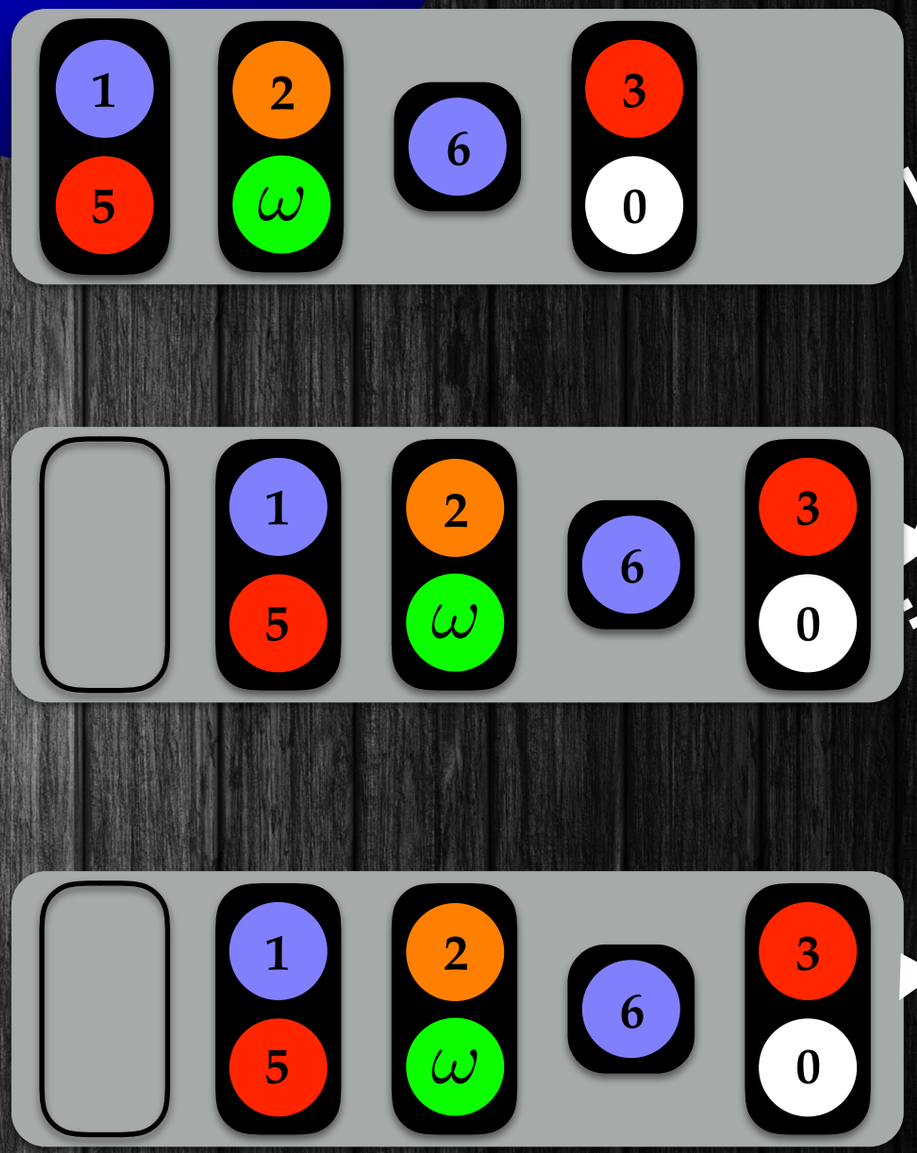
time=0.1 time=0.1 time=0.1 …

5.0 3.7 8.2 0.7 2.2 6.5 1.0

5.1 3.8 8.3 0.8 2.3 6.6 1.1

5.2 3.9 8.4 0.9 2.4 6.7 1.2

5.3 4.0 8.5 1.0 2.5 6.8 1.3



Timed Pe… Signa…

sig(c)=s:
c and g are "bisimilar"

5.0 3.7 8.2 0.7 2.2 6.5 1.0

time=0.1

5.1 3.8 8.3 0.8 2.3 6.6 1.1

time=0.1

5.2 3.9 8.4 0.9 2.4 6.7 1.2

time=0.1

5.3 4.0 8.5 1.0 2.5 6.8 1.3

time

time

time

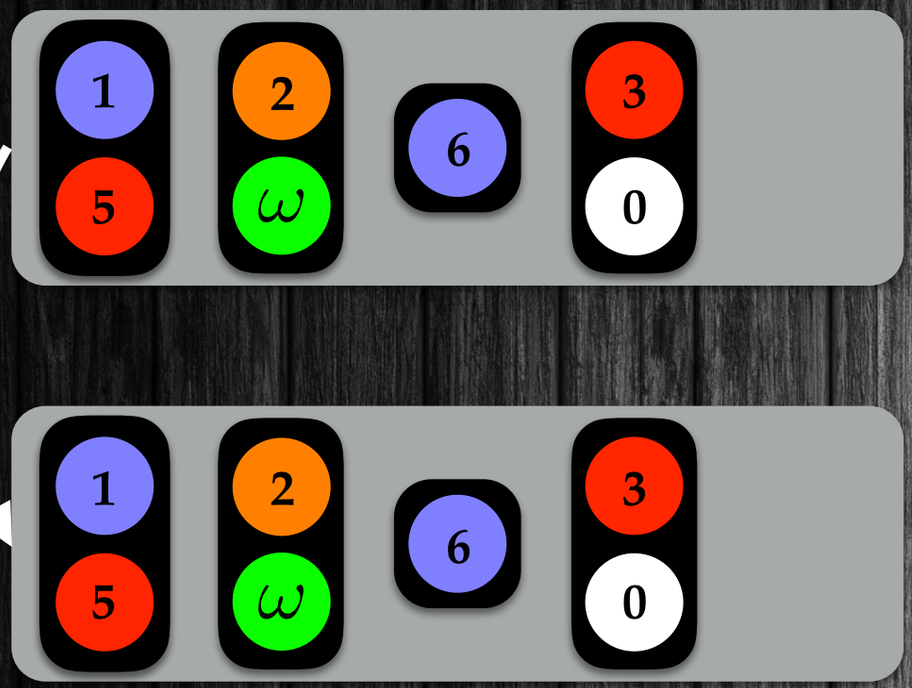
Timed Pe

Signatures

Timed Pe

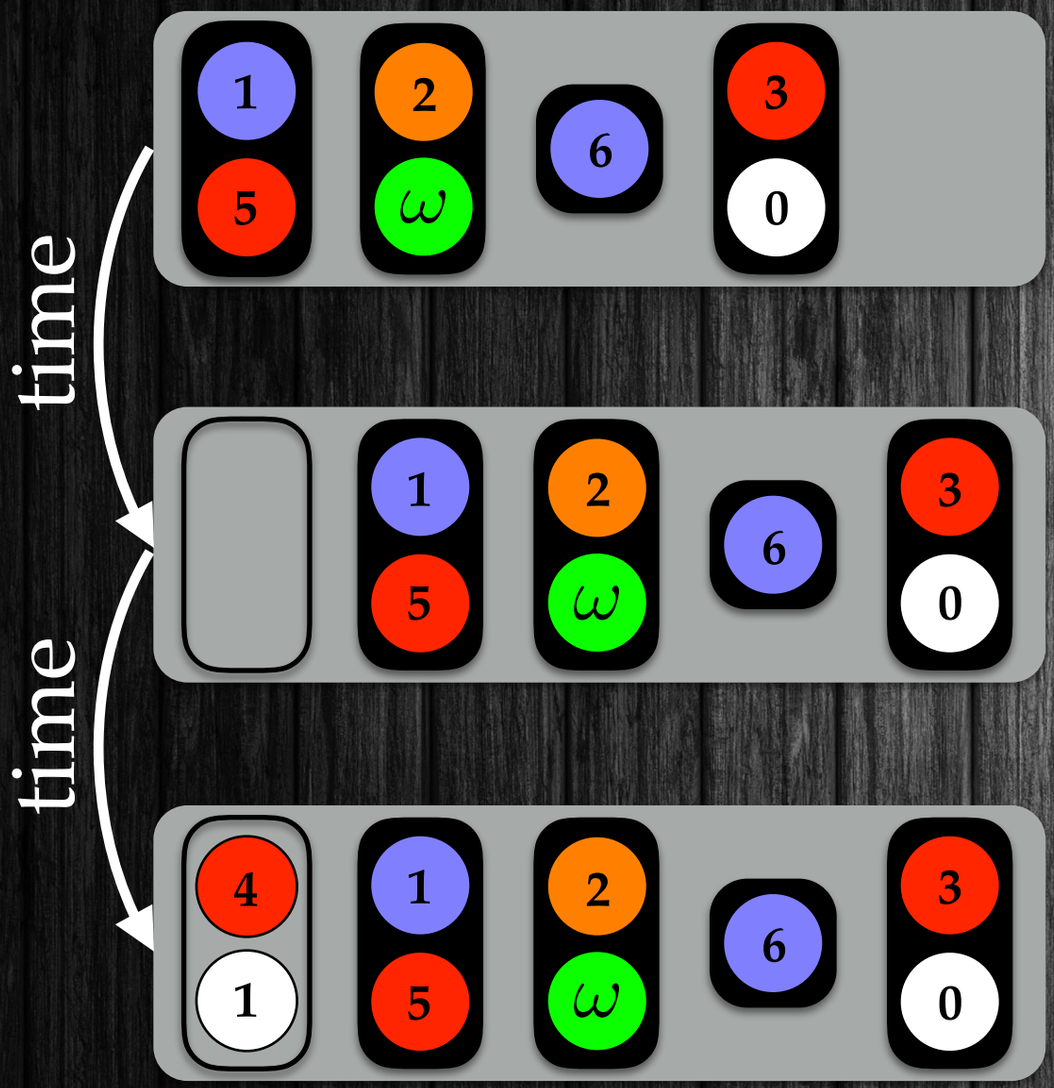
Signatures

time



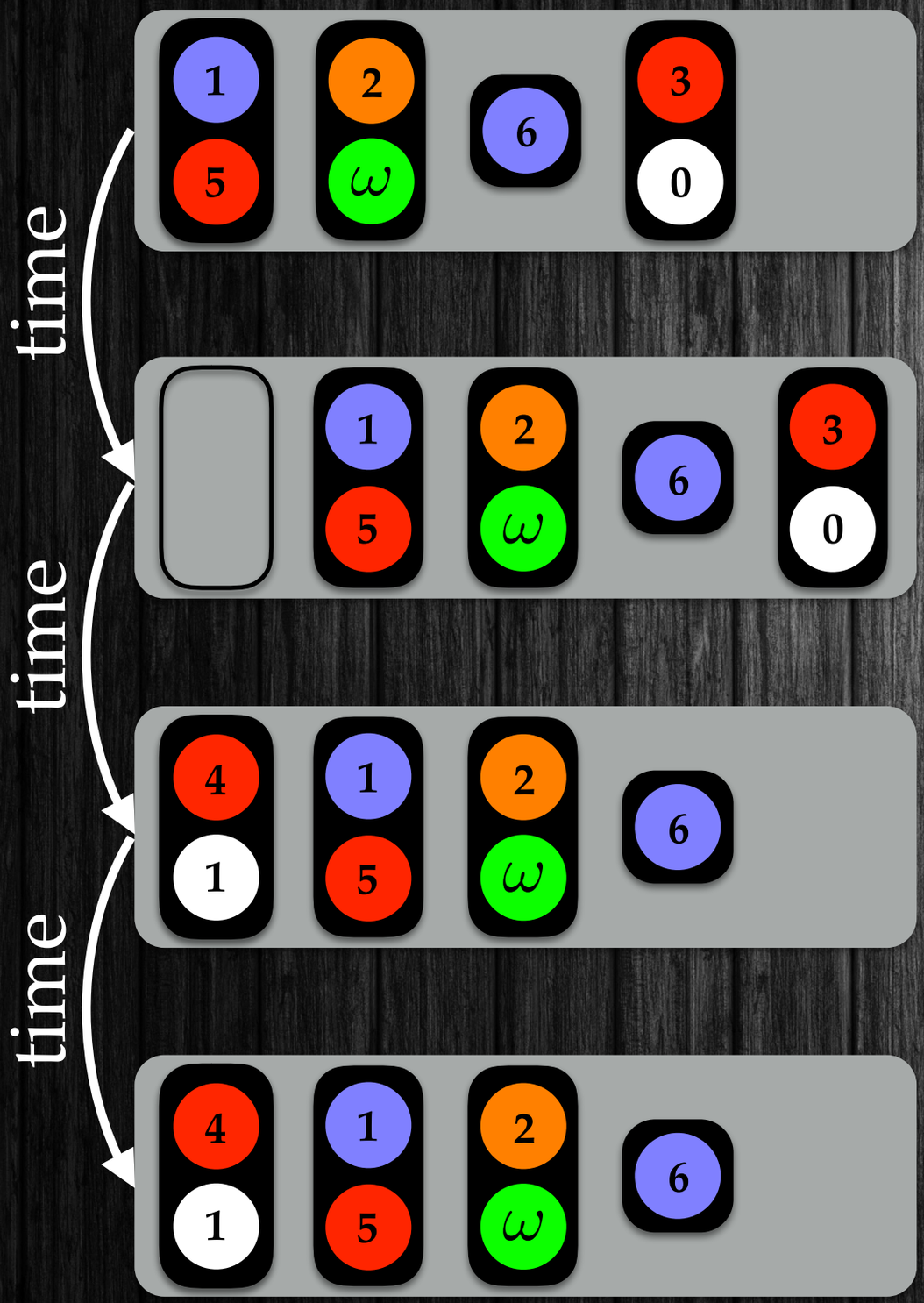
Timed Pe

Signatures

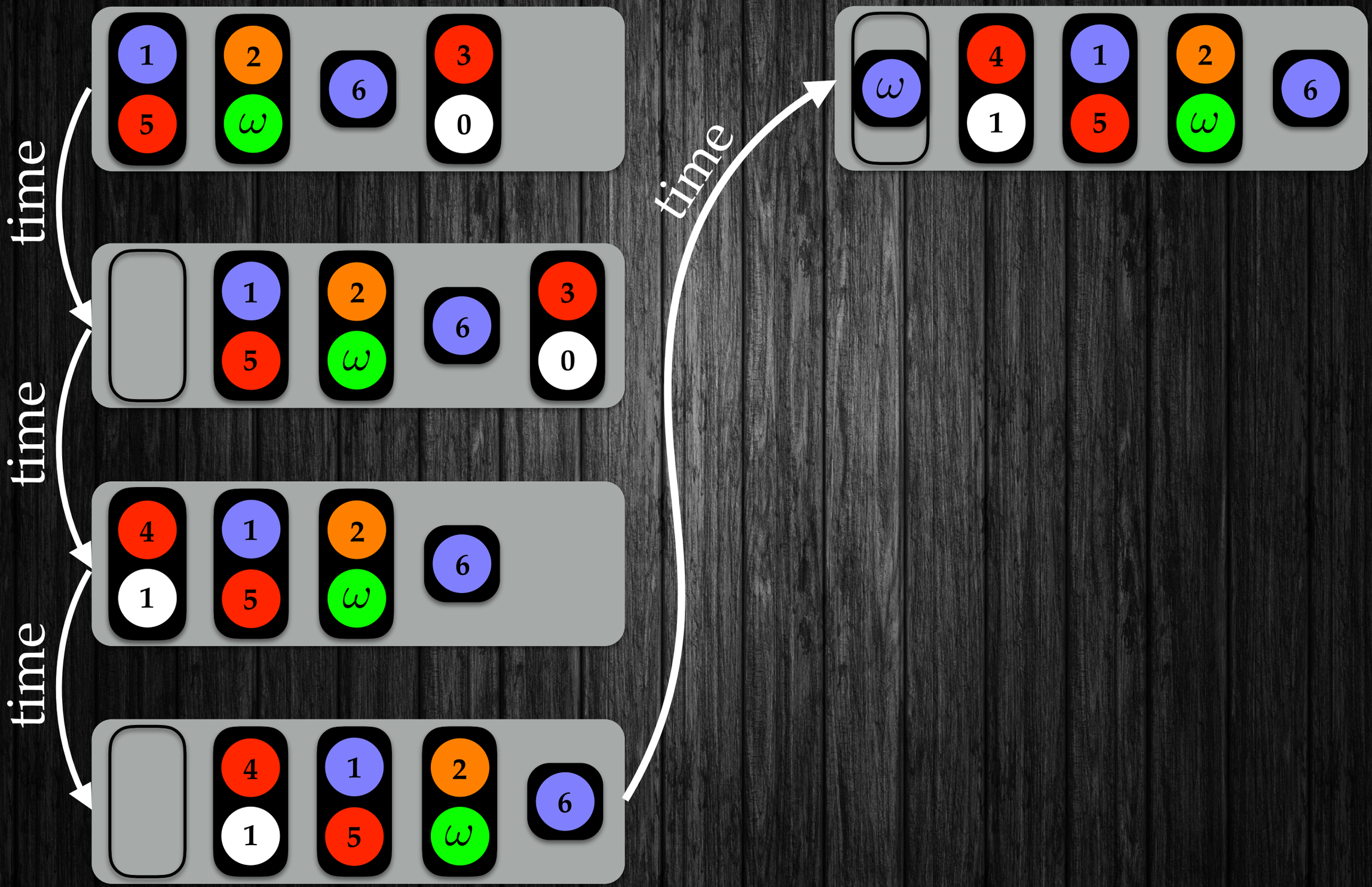


Timed Pe

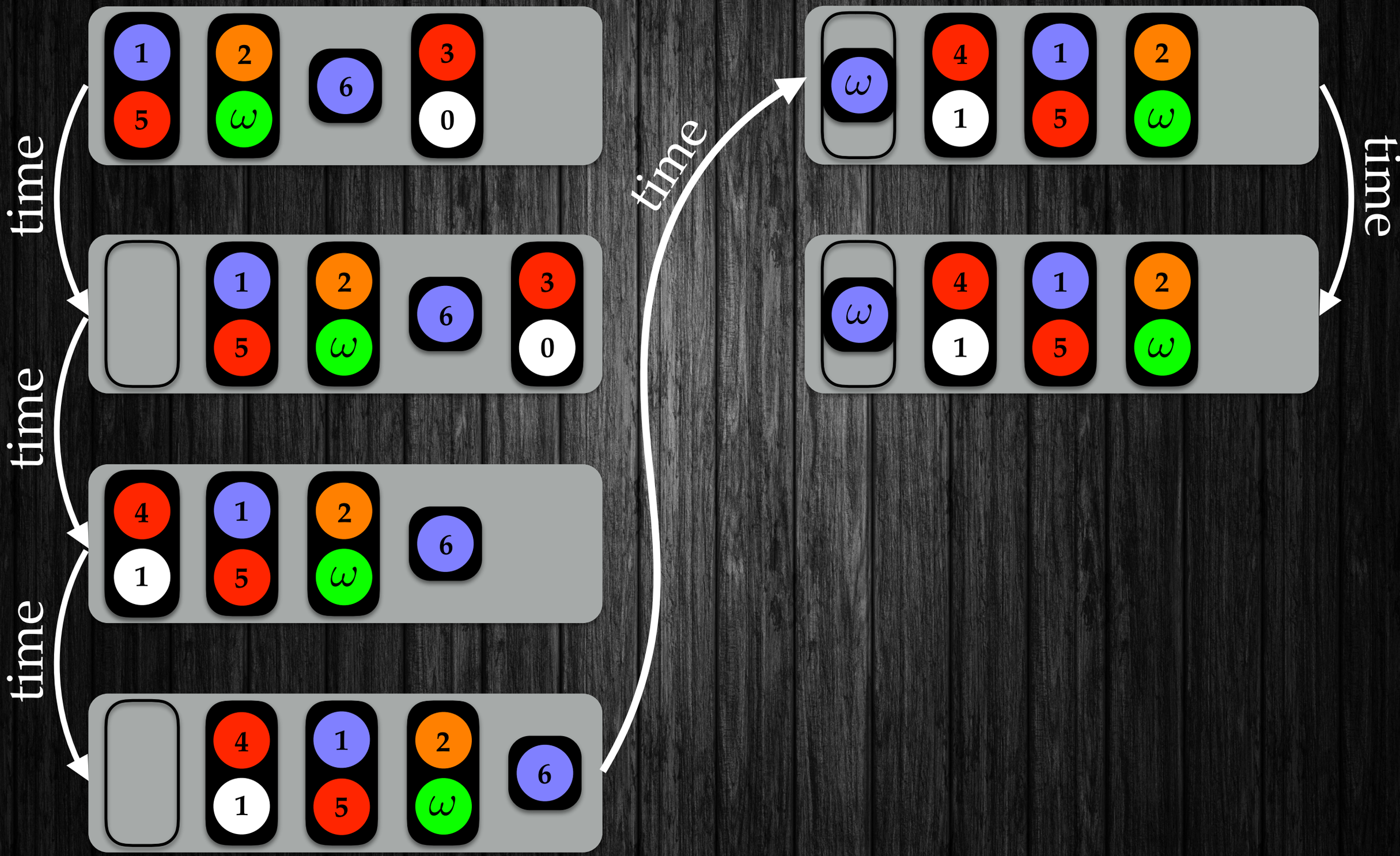
Signatures



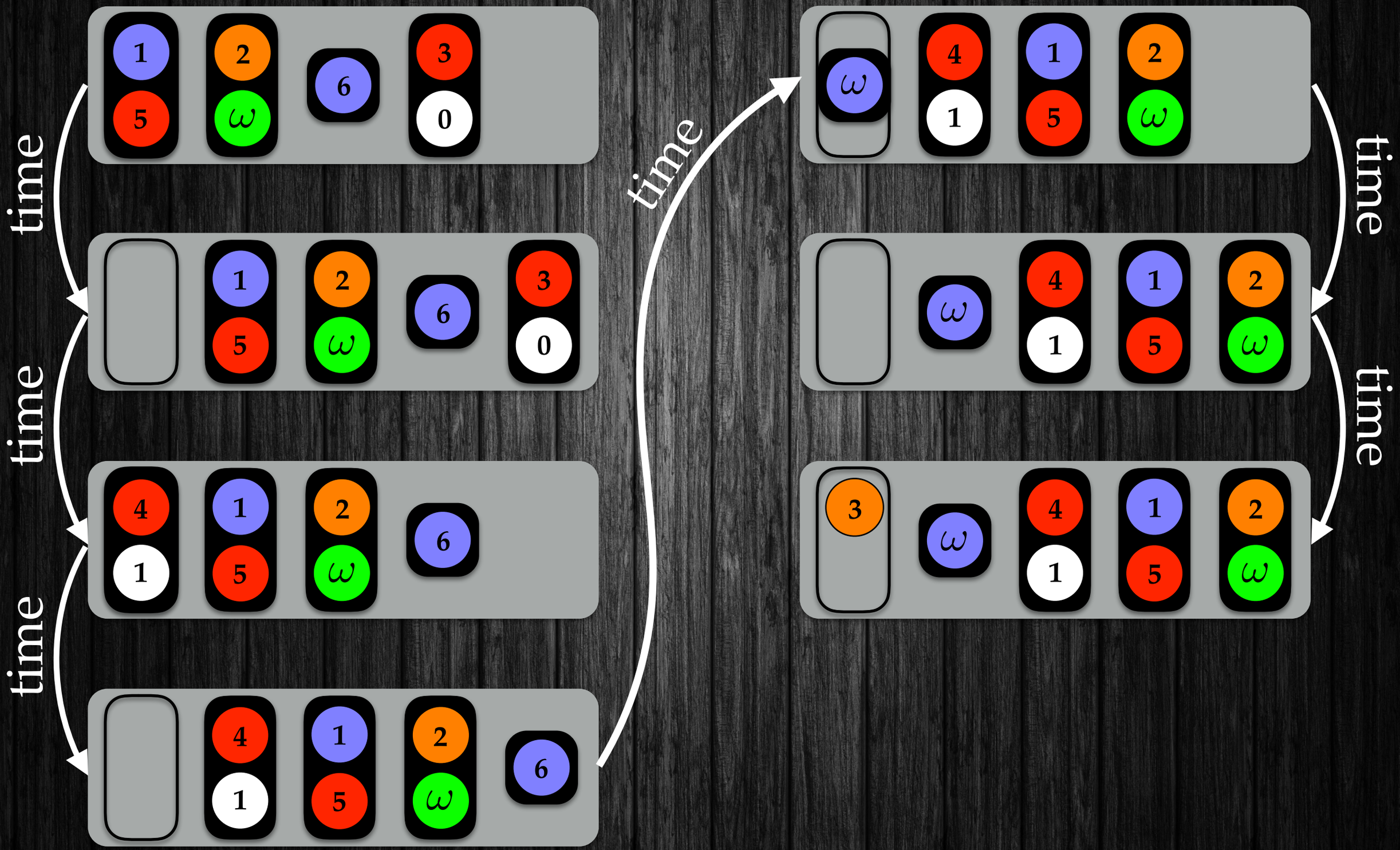
Timed Pe… Signatures



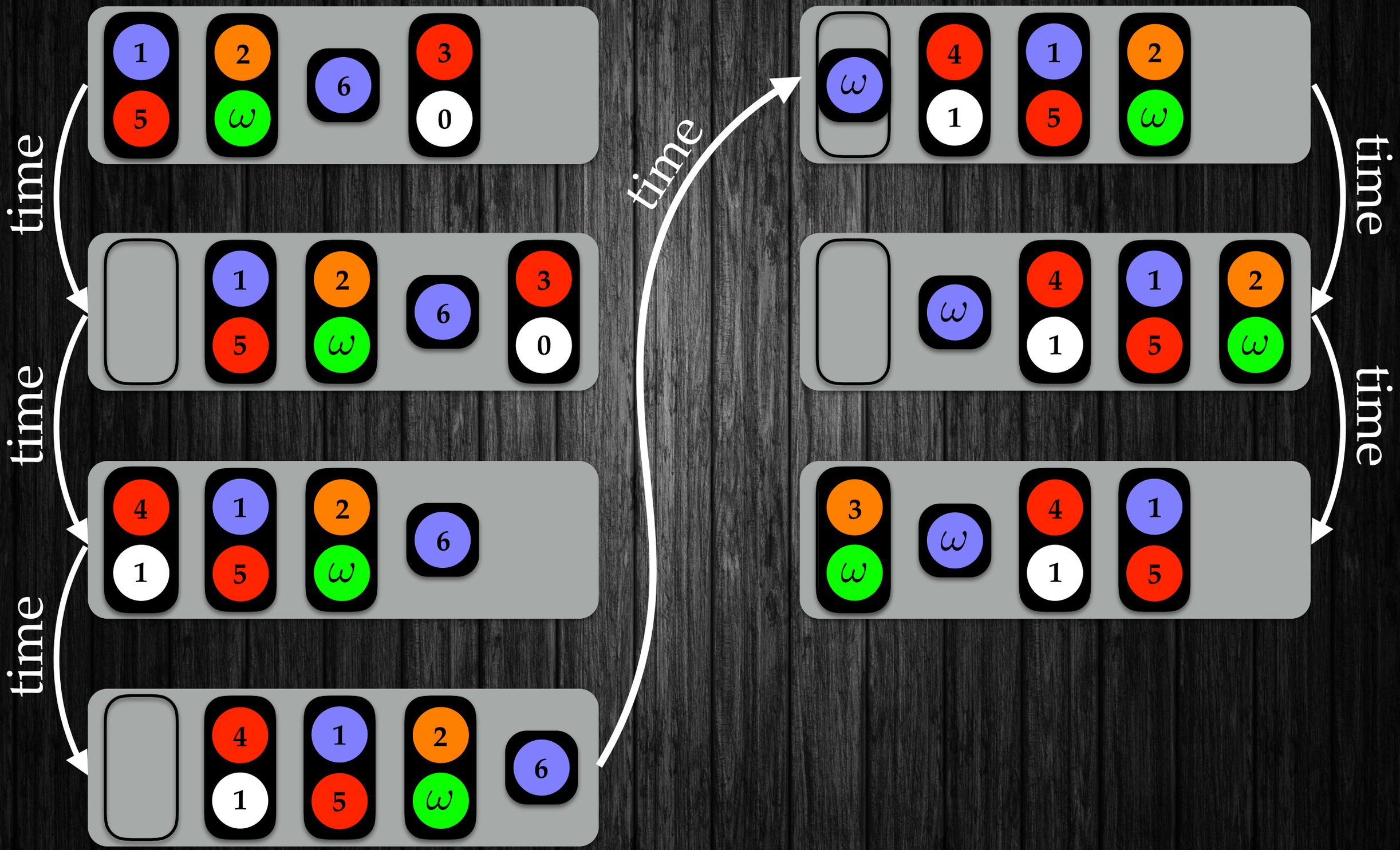
Timed Pet… Signatures



Timed Pe… Signatures



Timed Pet… Signatures



Timed Petri Nets

Model ✓

Configurations ✓

Transitions ✓

signatures

Ordering

Monotoncity

Upward Closed Sets

Computing Predecessors

Backward Reachability

Timed Pe

Equivalence

$c_1 \equiv c_2$:

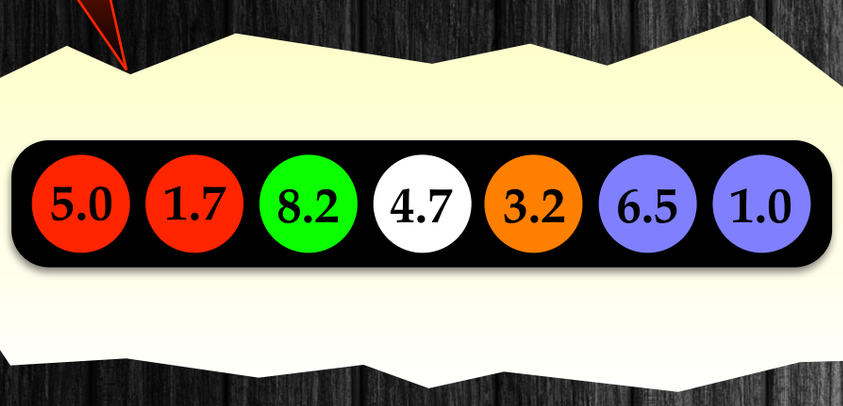
$sig(c_1) = sig(c_2)$

Timed Pet… Equivalence

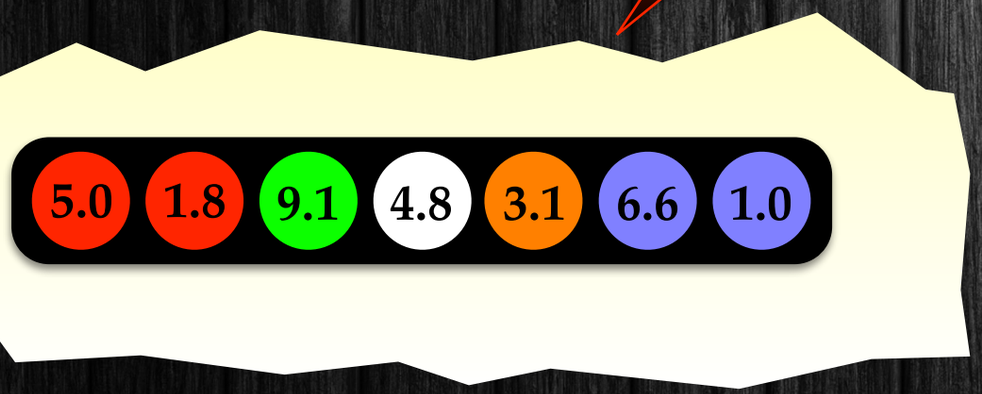
$c_1 \equiv c_2$:

$sig(c_1) = sig(c_2)$

c_1



c_2



Timed Pe… Equivalence

$c_1 \equiv c_2$:

$sig(c_1) = sig(c_2)$

c_1

5.0 1.7 8.2 4.7 3.2 6.5 1.0

$sig(c_1)$

1 5 3 ω 6 1 4

c_2

5.0 1.8 9.1 4.8 3.1 6.6 1.0

$sig(c_2)$

Timed Pe

Ordering

$c_1 \sqsubseteq c_2$:

$\exists c_3.\ (c_1 \equiv c_3) \wedge (c_3 \subseteq c_2)$

Timed Pe… Ordering

$c_1 \sqsubseteq c_2$:

$$\exists c_3.\ (c_1 \equiv c_3) \wedge (c_3 \subseteq c_2)$$

c_1

5.0 1.7 8.2 4.7 3.2 6.5 1.0

$\sqsubseteq$

c_2

5.0 3.2 2.5 4.8 3.1 6.6 1.0

1.8 9.1 9.1 1.1 9.9 6.6

Timed Pet… Ordering

$c_1 \sqsubseteq c_2$:

$\exists c_3.\ (c_1 \equiv c_3) \wedge (c_3 \subseteq c_2)$

c_1

5.0 1.7 8.2 4.7 3.2 6.5 1.0

c_3

5.0 3.2 2.5 4.8 3.1 6.6 1.0
1.8 9.1 9.1 1.1 9.9 6.6

c_2

5.0 3.2 2.5 4.8 3.1 6.6 1.0
1.8 9.1 9.1 1.1 9.9 6.6

$\mathrm{sig}(c_1)$

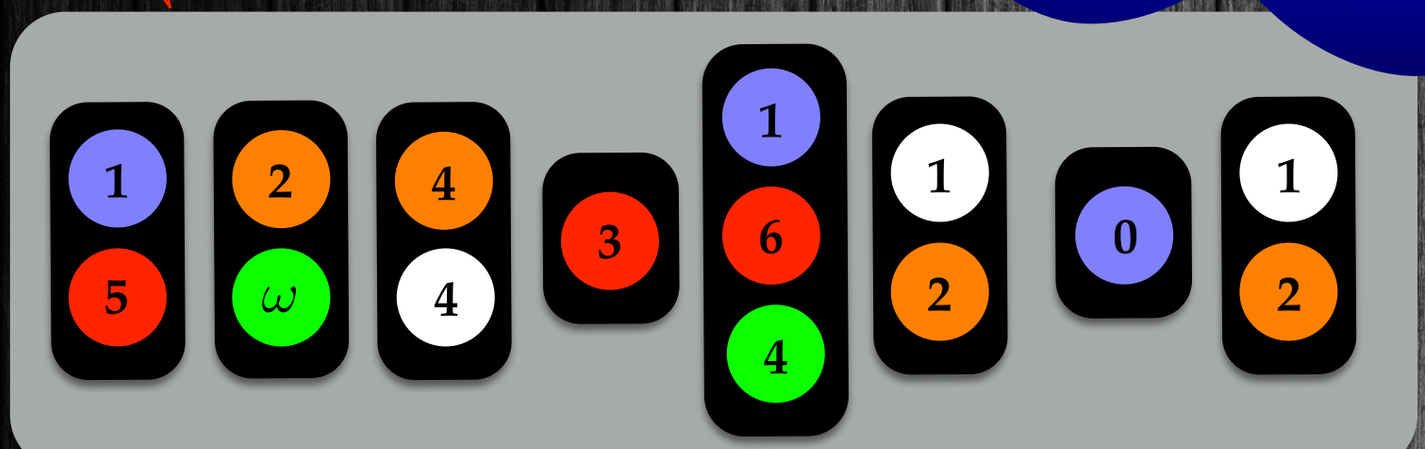
$\mathrm{sig}(c_3)$

Timed Pet… Ordering

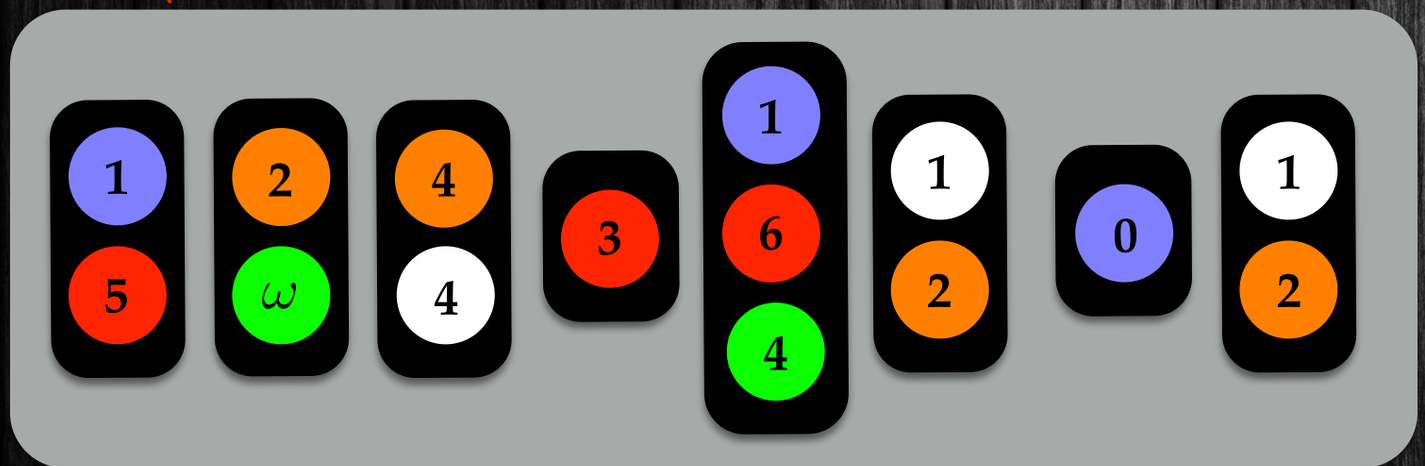
$s_1 \sqsubseteq s_2$: Derive s_1 from s_2 by:

- removing elements from multisets
- removing multisets

s_1



s_2

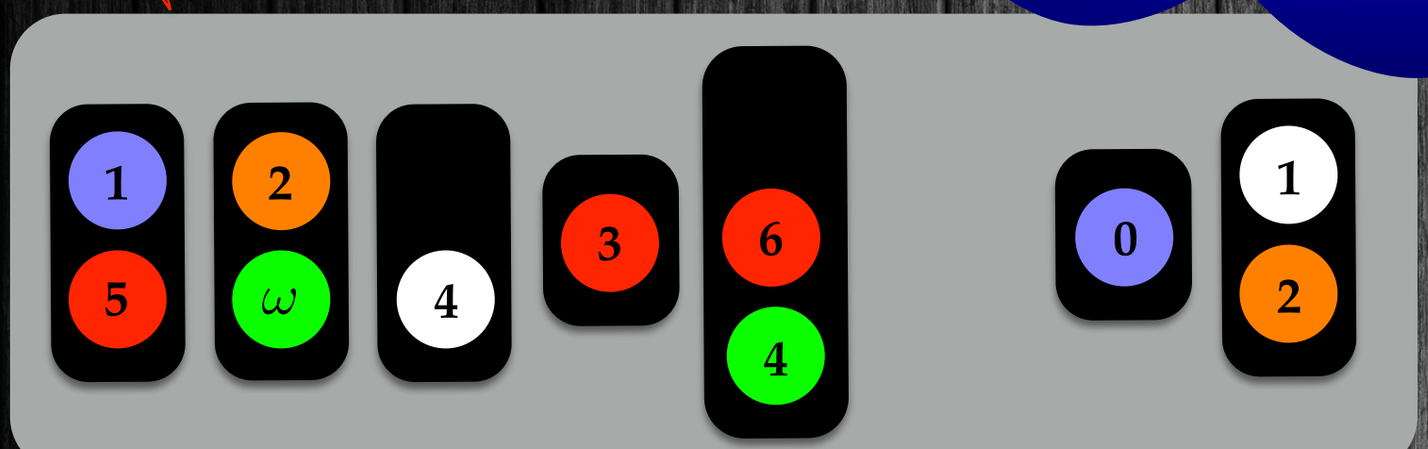


Timed Pet… Ordering

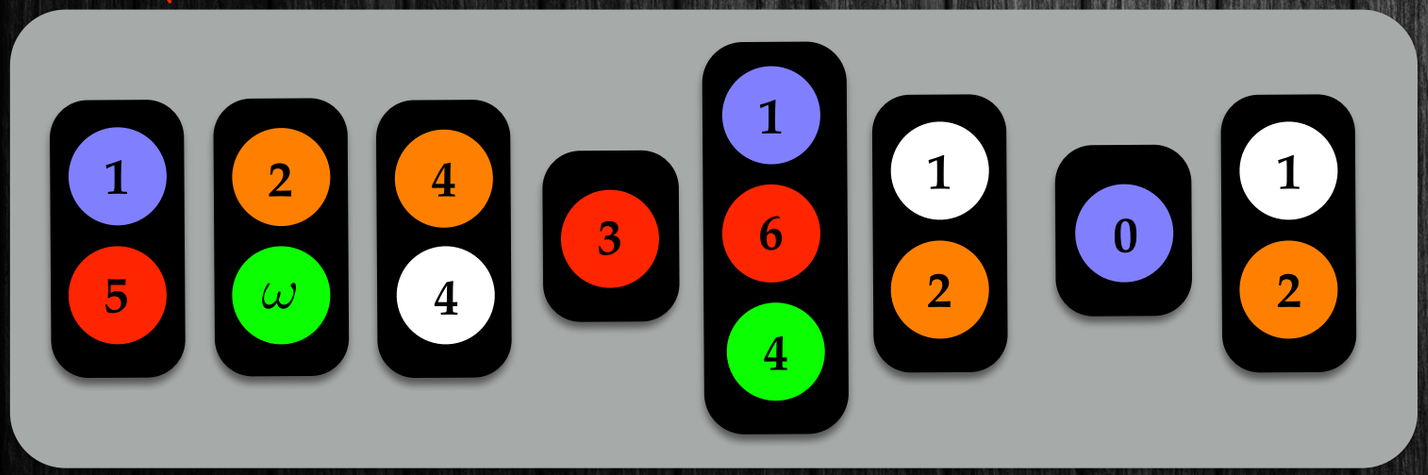
$s_1 \sqsubseteq s_2$: Derive s_1 from s_2 by:

- removing elements from multisets
- removing multisets

s_1



s_2

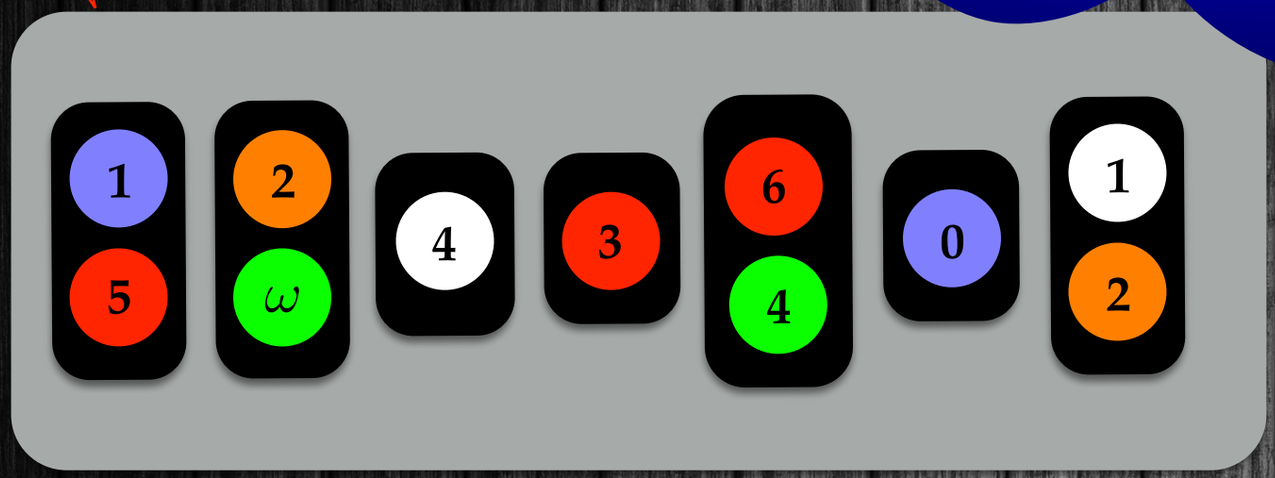


Timed Pe… Ordering

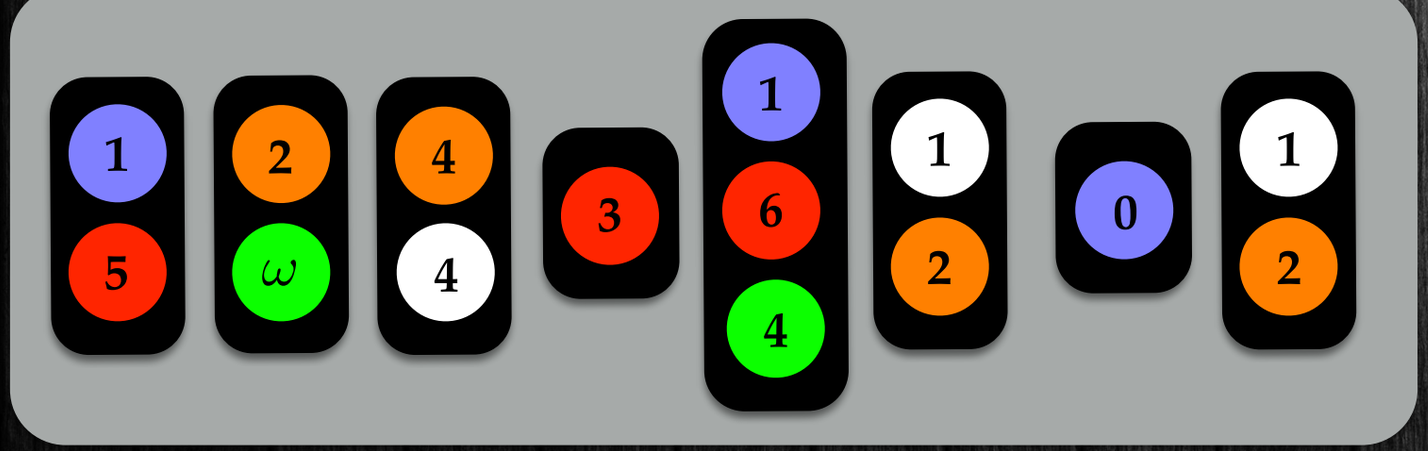
$s_1 \sqsubseteq s_2$: Derive s_1 from s_2 by:

- removing elements from multisets
- removing multisets

s_1



s_2



Timed Pe

Ordering

$$c \models s \ :$$

$$\exists c' . (c' \sqsubseteq c) \wedge (sig(c') = s)$$

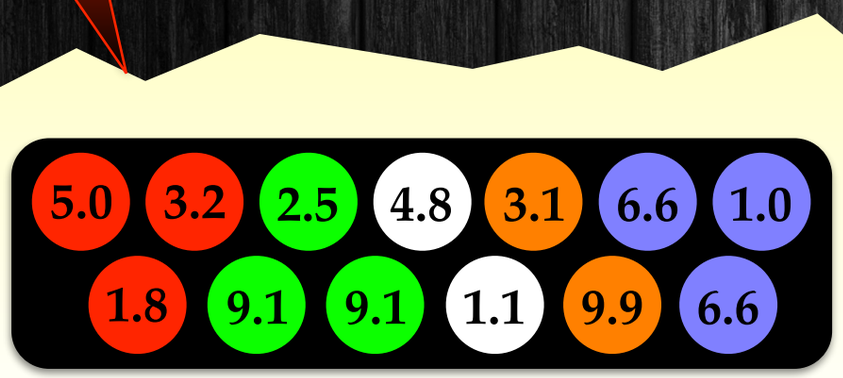
Timed Pet

Ordering

$$c \models s \ :$$

$$\exists c' . (c' \sqsubseteq c) \wedge (sig(c') = s)$$

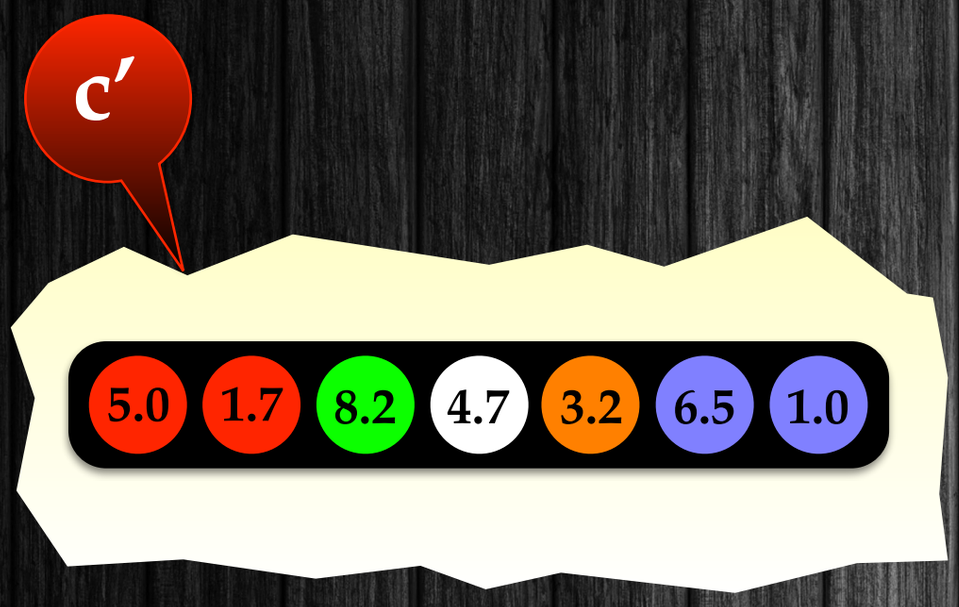
c



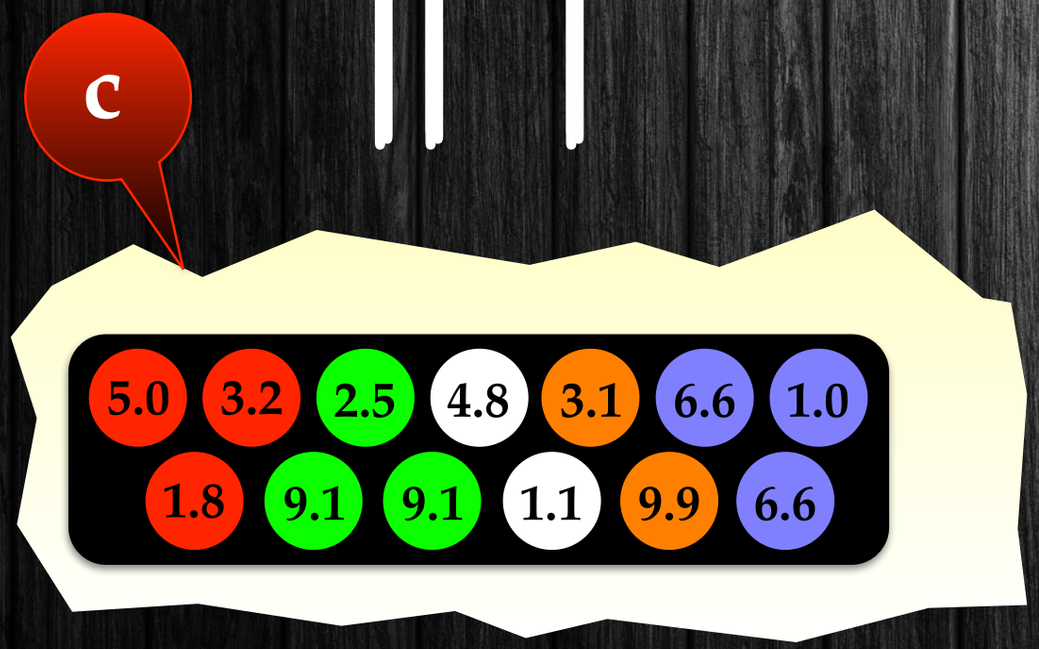
s

Timed Pet… Ordering

$$c \models s \;:\; \exists c'.\,(c' \sqsubseteq c) \wedge (sig(c') = s)$$



$\sqsubseteq$



Timed Pet… Ordering

$c \models s$:

$\exists c'.\,(c' \sqsubseteq c) \wedge (sig(c') = s)$

c′

5.0 1.7 8.2 4.7 3.2 6.5 1.0

$\sqsubseteq$

c

5.0 3.2 2.5 4.8 3.1 6.6 1.0
1.8 9.1 9.1 1.1 9.9 6.6

$\models$

sig(c′)

1 5 | 3 ω | 6 | 1 4

s

Timed Petri Nets

Model

Configurations

Transitions

Ordering

Monotoncity

denotation

Upward Closed Sets

Computing Predecessors

Backward Reachability

Timed Pe

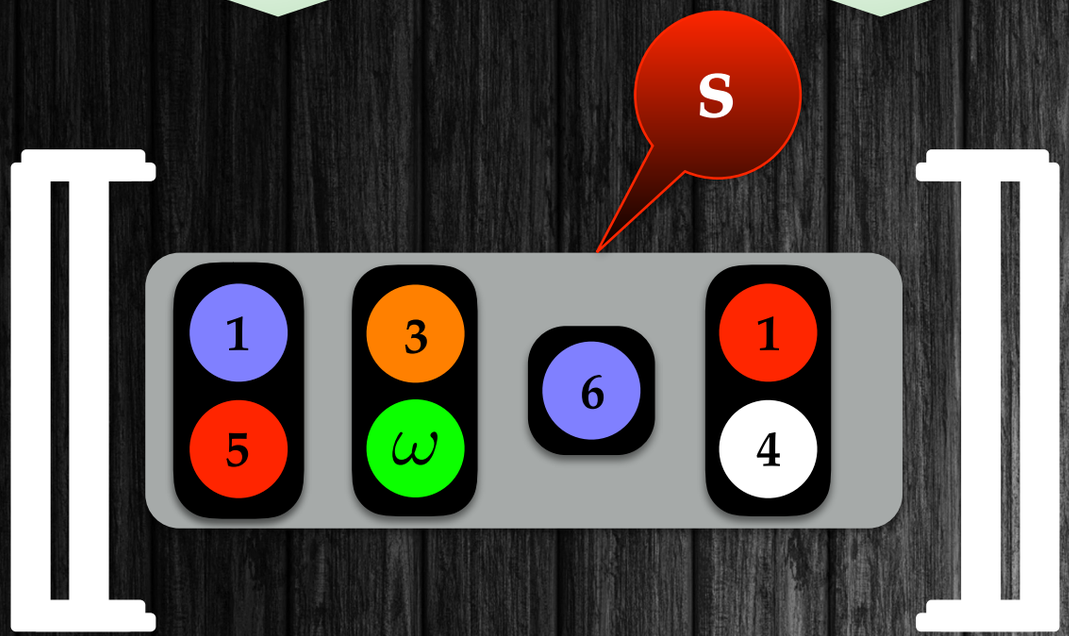
Denotation

$$[\![s]\!] = \{c \mid c \models s\}$$

Timed Pe

Denotation

$$[\![s]\!] = \{c \mid c \models s\}$$

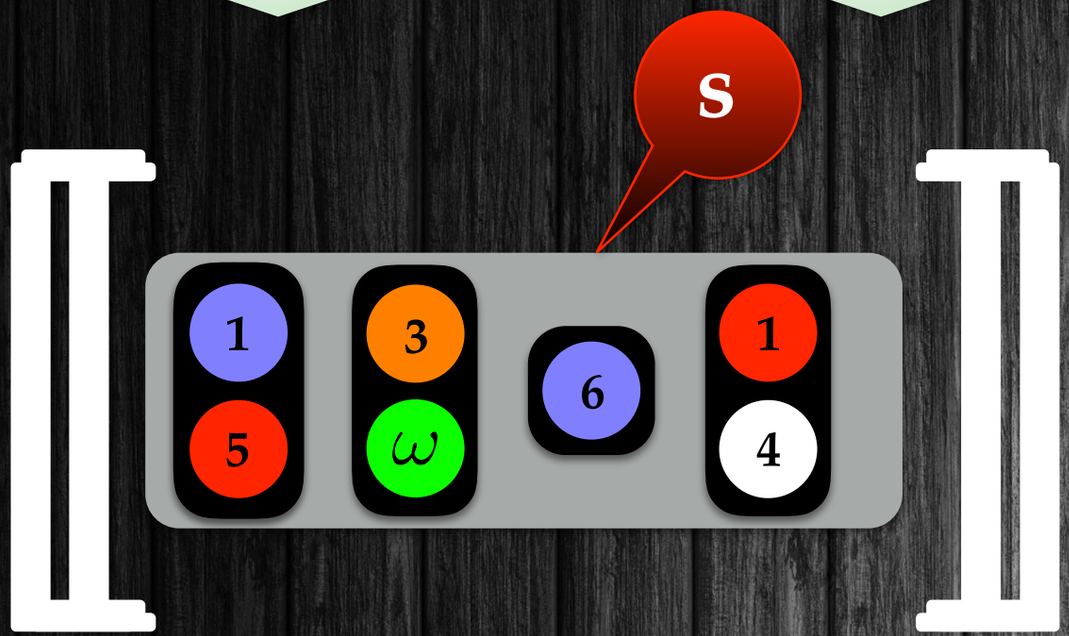


=

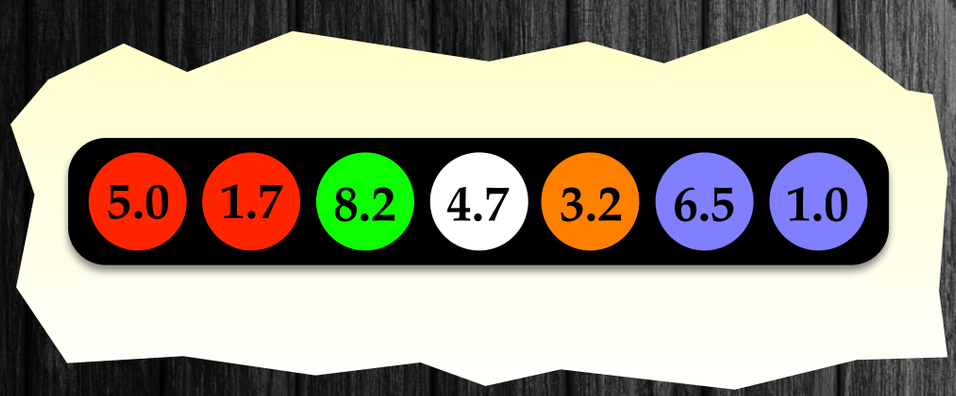
Timed Pe

Denotation

$$[\![s]\!] = \{c \mid c \models s\}$$



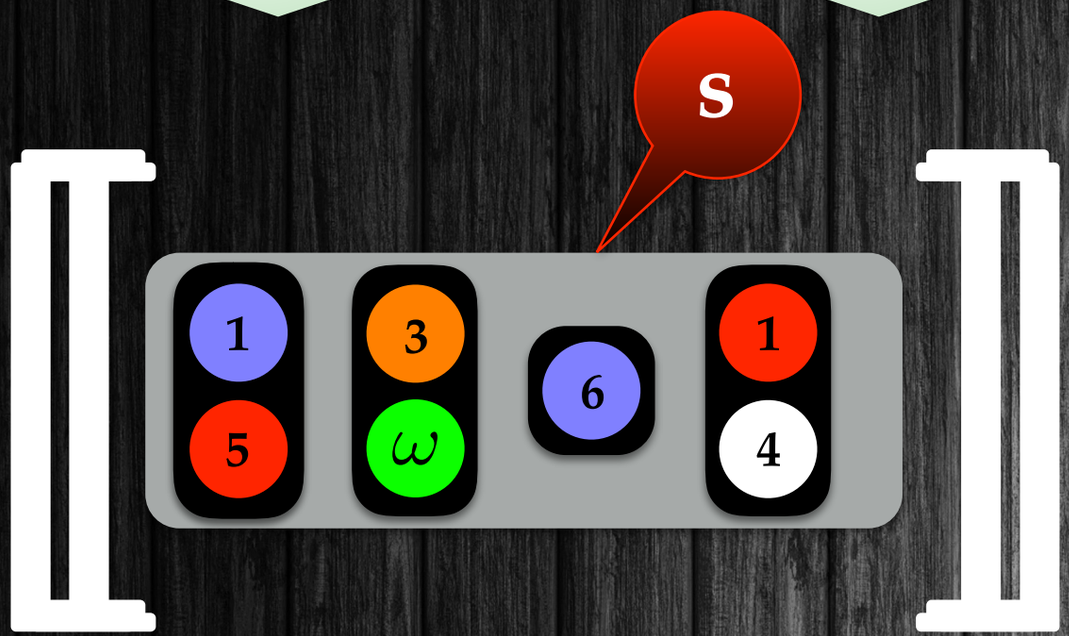
=



Timed Pe

Denotation

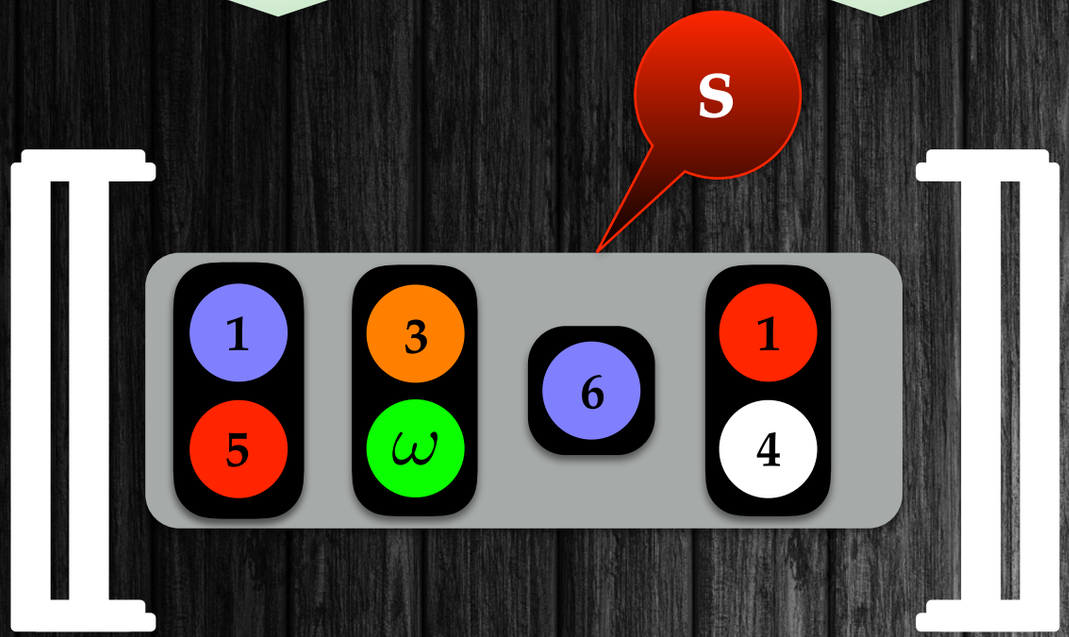
$$[\![s]\!] = \{c \mid c \models s\}$$



=

Timed Pe… Denotation

$$[\![s]\!] = \{c \mid c \models s\}$$



=

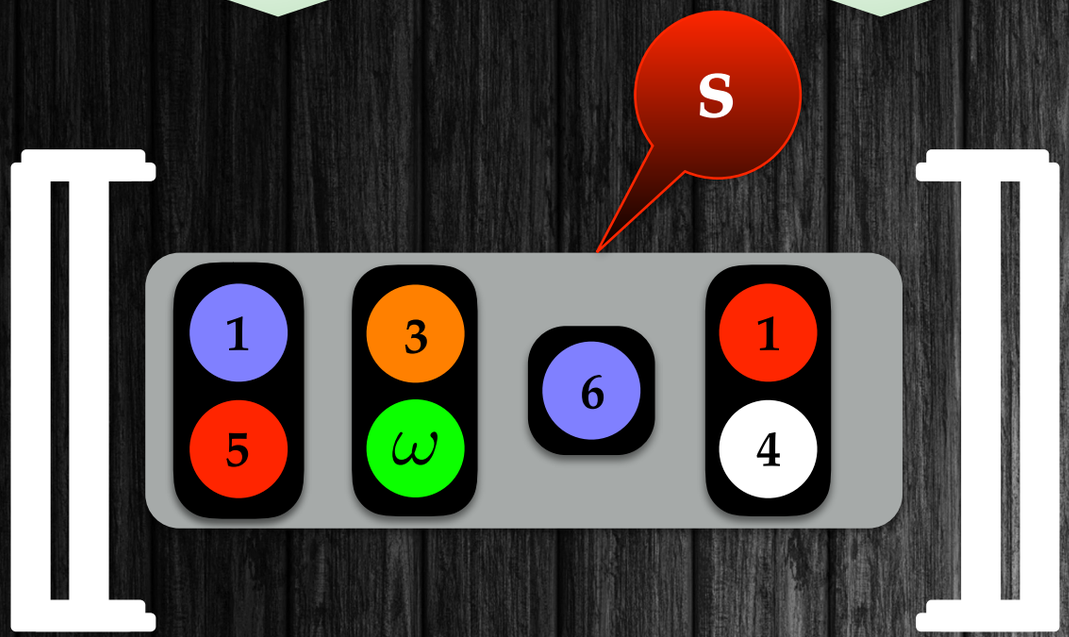
5.0 1.7 8.2 4.7 3.2 6.5 1.0 ,

5.0 1.8 9.1 4.8 3.1 6.6 1.0 ,

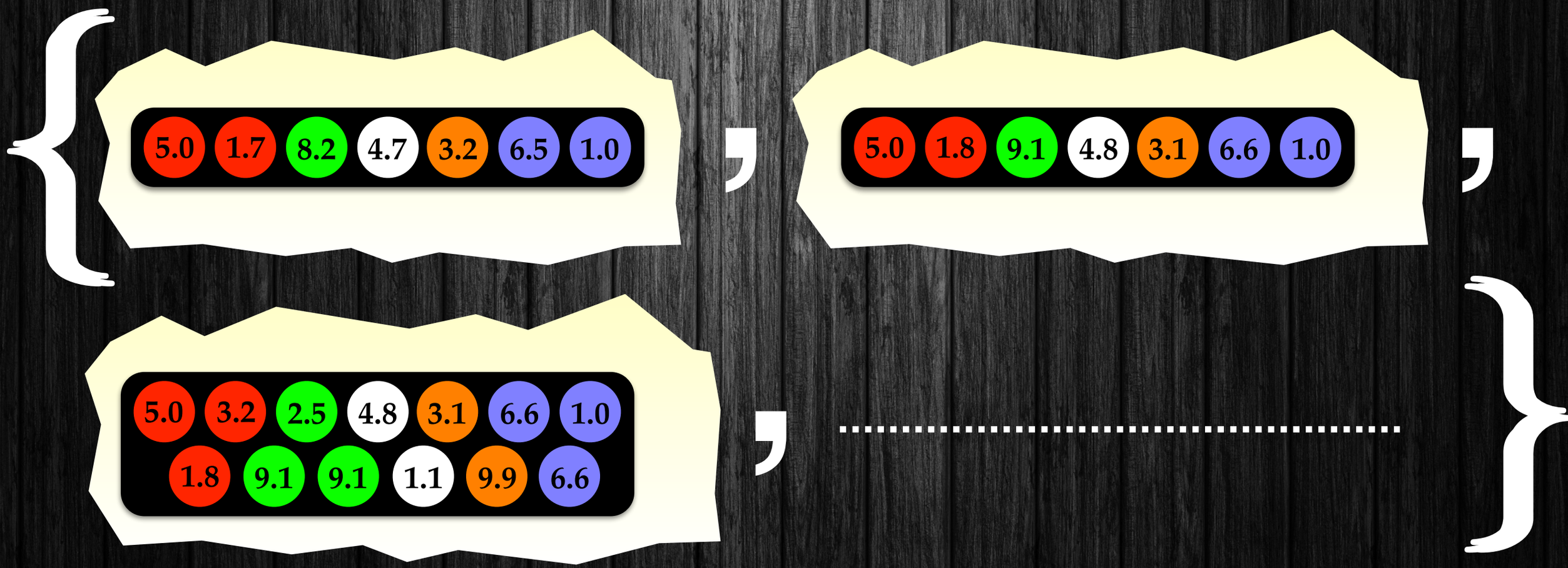
5.0 3.2 2.5 4.8 3.1 6.6 1.0
1.8 9.1 9.1 1.1 9.9 6.6

Timed Pe… Denotation

$$[\![s]\!] = \{c \mid c \models s\}$$



=



Timed Pe… Denotation

$[\![s]\!] = \{c \mid c \models s\}$

infinite

upward closed wrt. $\sqsubseteq$

=

{ 5.0 1.7 8.2 4.7 3.2 6.5 1.0 , 5.0 1.8 9.1 4.8 3.1 6.6 1.0 , 5.0 3.2 2.5 4.8 3.1 6.6 1.0 1.8 9.1 9.1 1.1 9.9 6.6 , ... }

Timed Pe… Denotation

$[\![s]\!] = \{c \mid c \models s\}$

s

infinite

upward closed wrt. $\sqsubseteq$

$[\![$ 1 5 | 3 ω | 6 | 1 4 $]\!]$ =

$s_1 \sqsubseteq s_2$

implies

$[\![s_2]\!] \subseteq [\![s_1]\!]$

{ 5.0 1.7 8.2 4.7 … 1.8 9.1 4.8 3.1 6.6 1.0 ,

{ 5.0 3.2 2.5 4.8 3.1 6.6 1.0
1.8 9.1 9.1 1.1 9.9 6.6 , ... }

Timed Petri Nets

Model ✓

Configurations ✓

Transitions ✓

Ordering ✓

Monotoncity

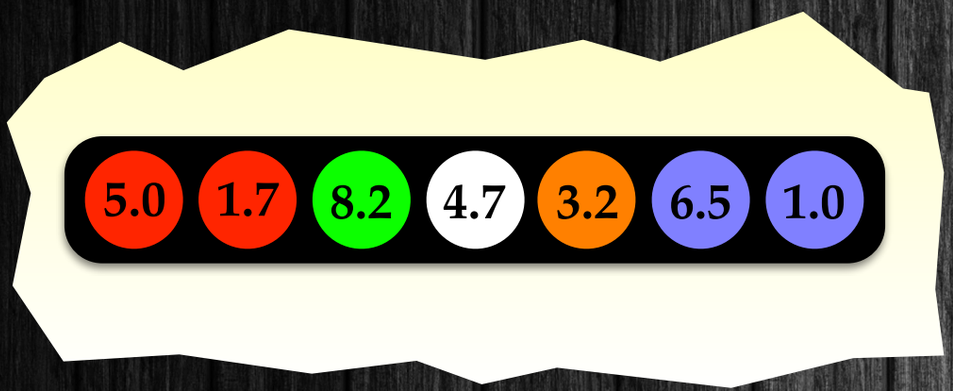
Upward Closed Sets ✓

Computing Predecessors

Backward Reachability

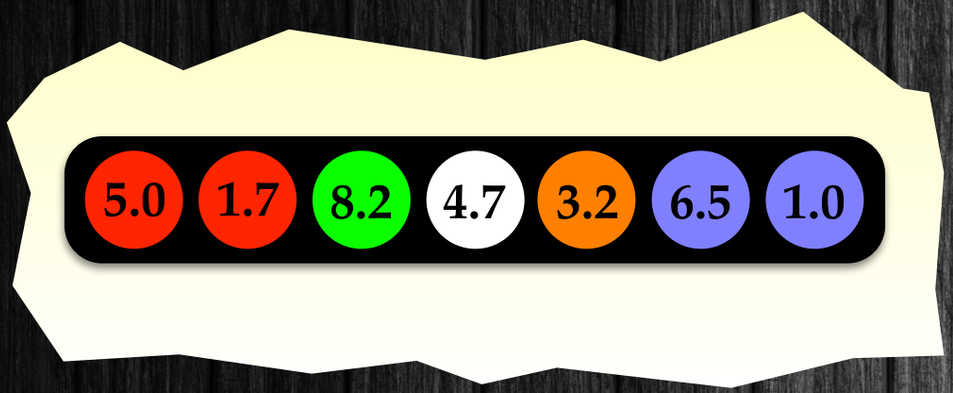
Timed

Monotonicity

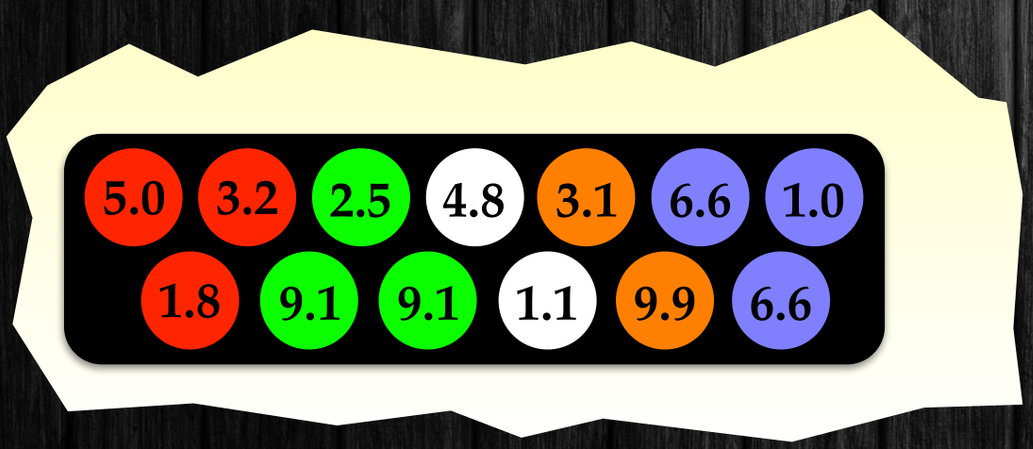


Timed

Monotonicity



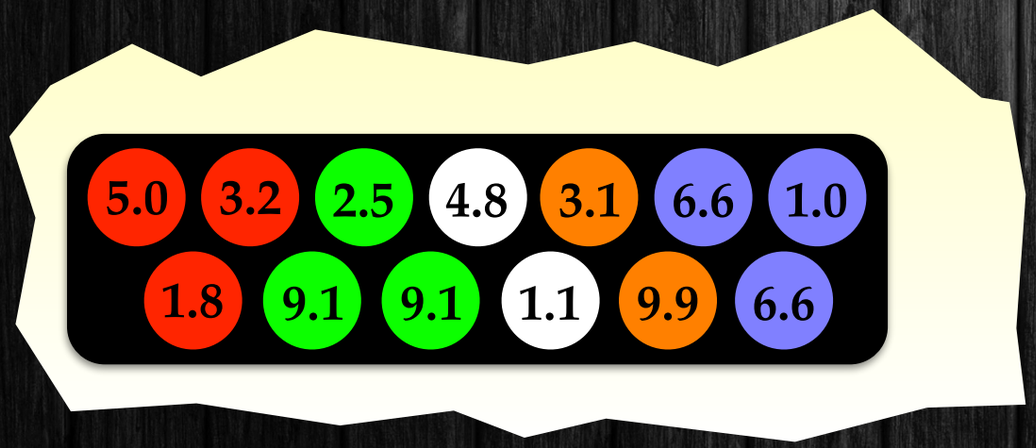
IN



Timed

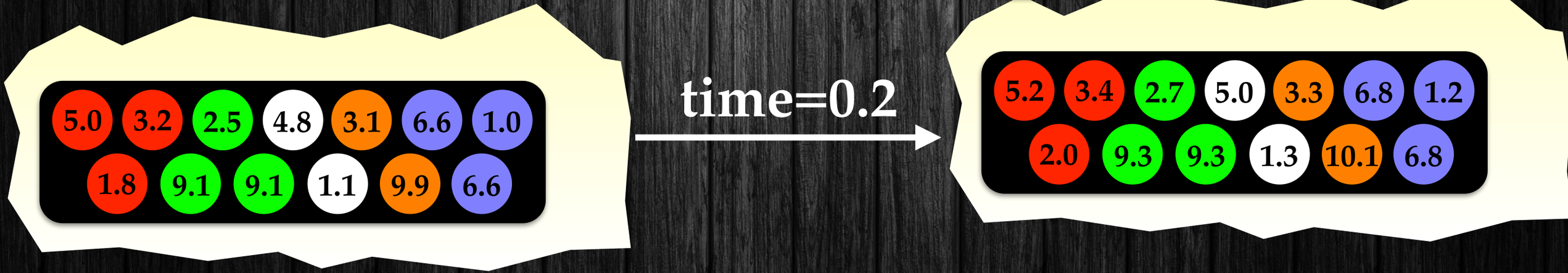
Monotonicity

IN



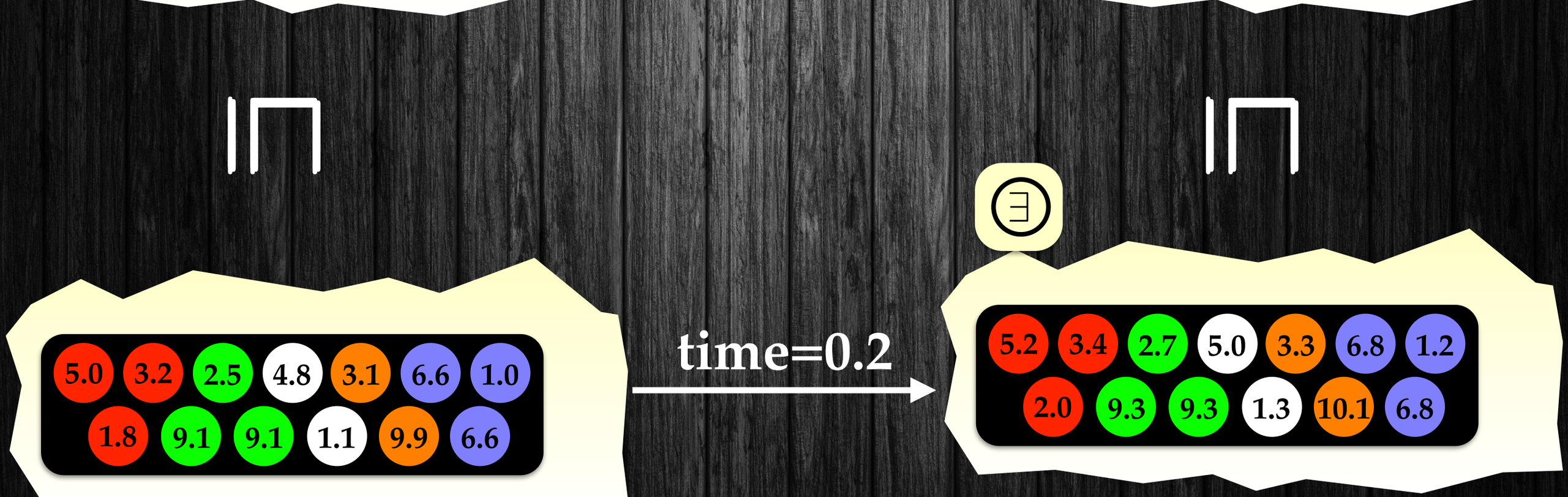
Timed [illegible] Monotonicity

IN



IN

Timed … Computing Predecessors



Timed Petri Nets

Model

Configurations

Transitions

Ordering

Monotoncity

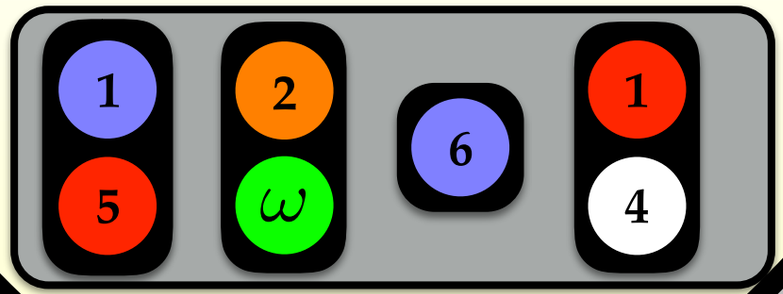
Upward Closed Sets

Computing Predecessors

Backward Reachability

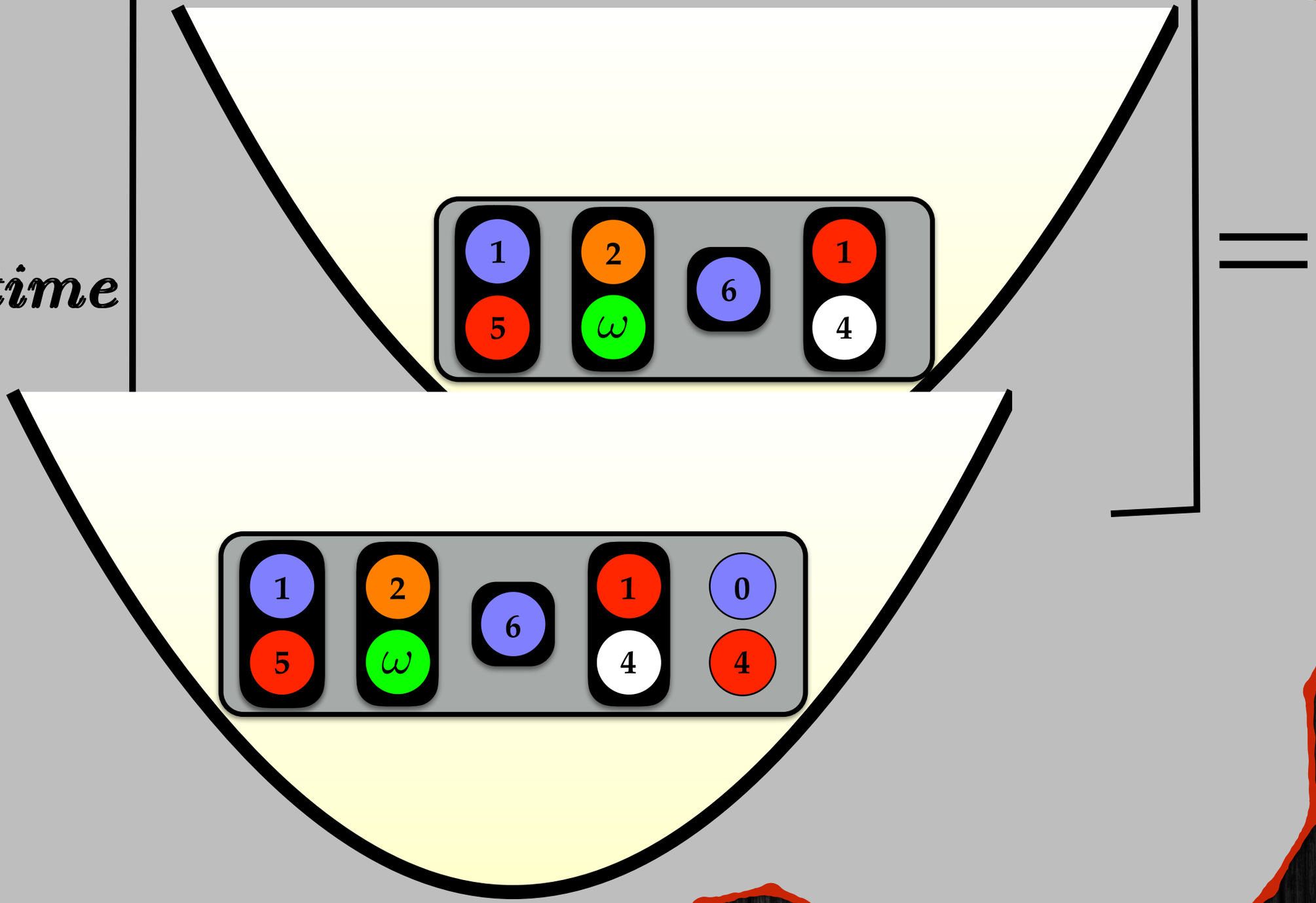
Timed … Computing Predecessors

$$Pre_{time}\left(\right) =$$



Timed … Computing Predecessors

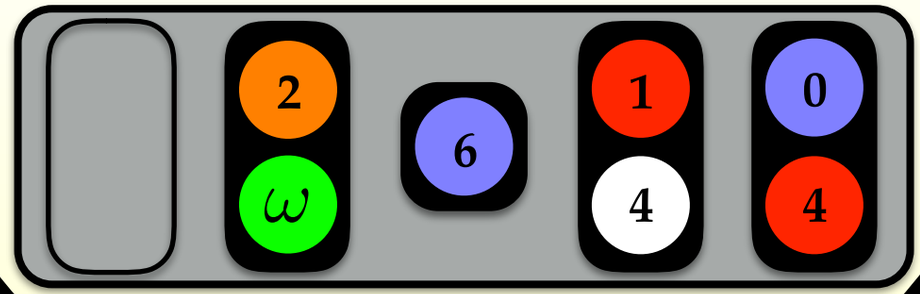
Pre_{time}



=

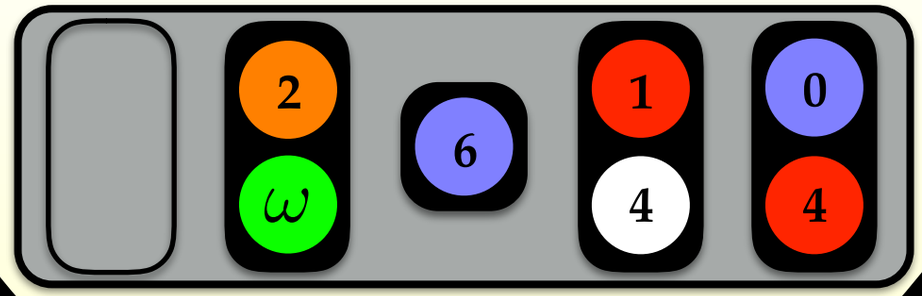
Timed … Computing Predecessors

$$Pre_{time}\left(\right) =$$

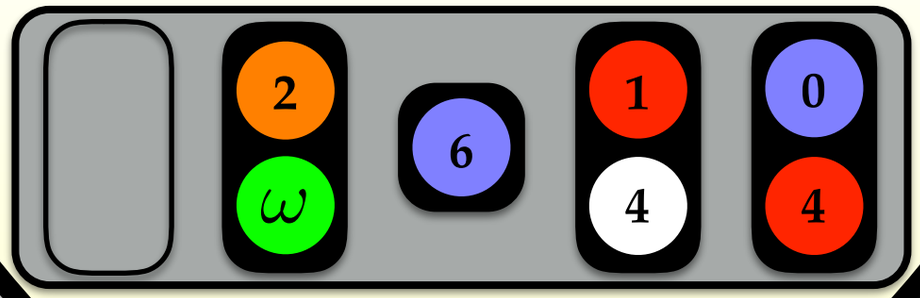


Timed … Computing Predecessors

Pre_{time}

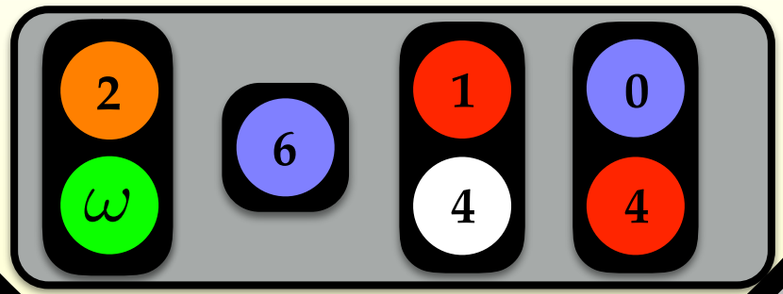


=



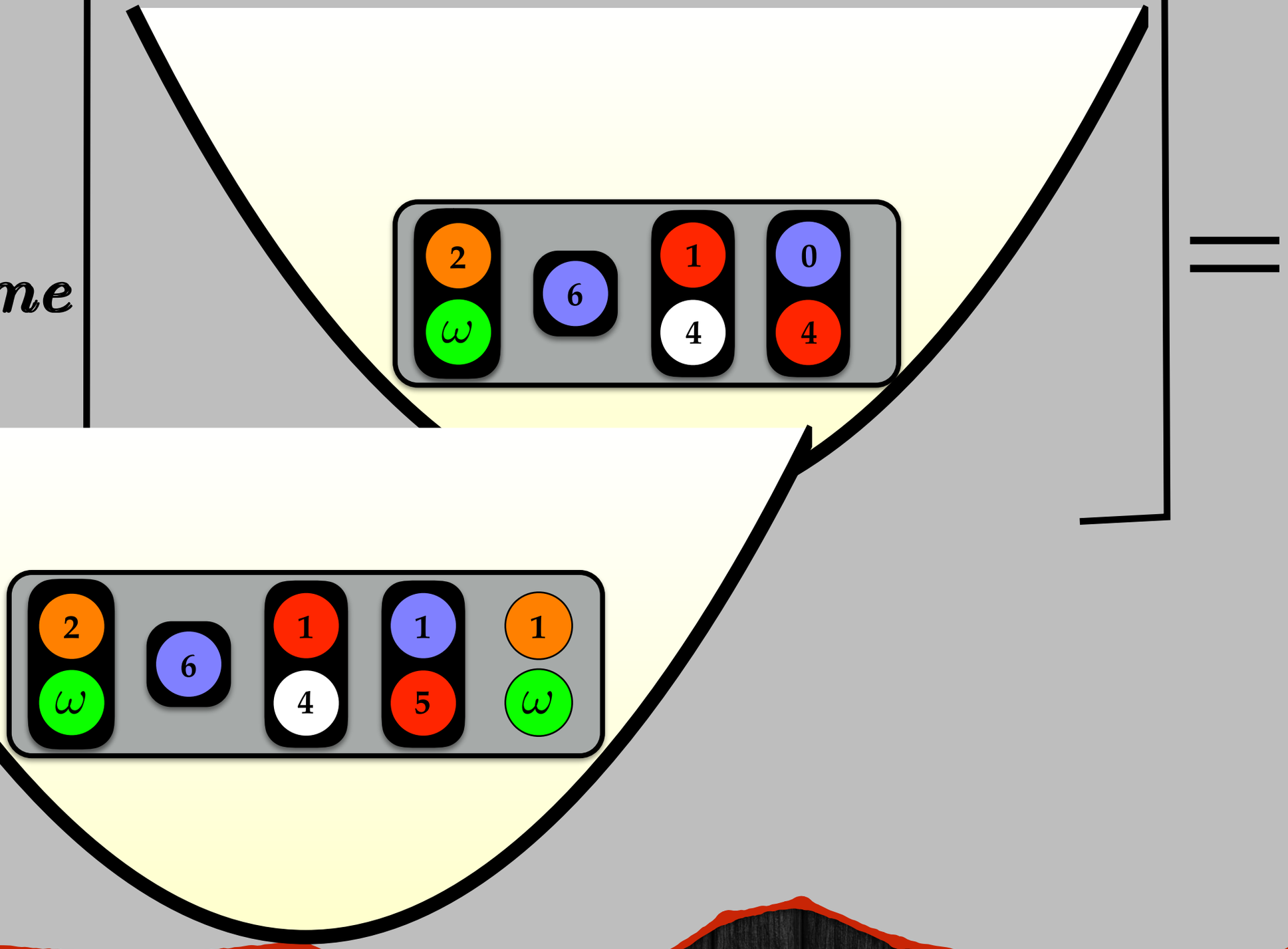
Timed Computing Predecessors

$$Pre_{time}\left(\begin{array}{c} \\ \\ \end{array}\right) =$$



Timed … Computing Predecessors

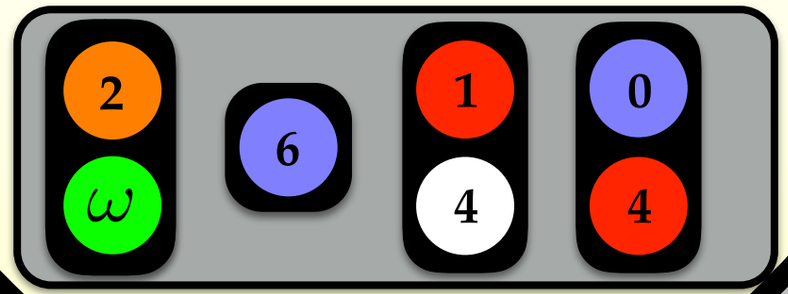
Pre_{time}

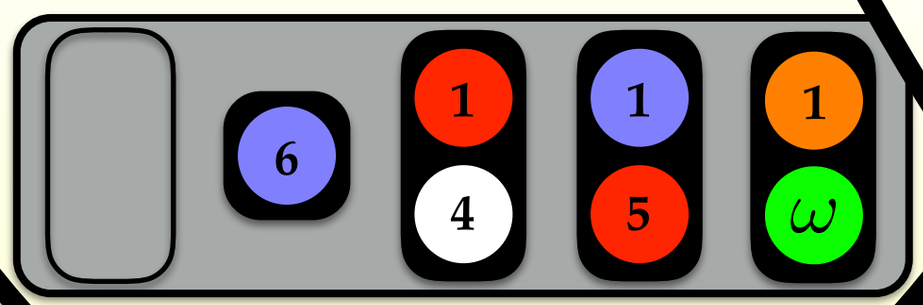


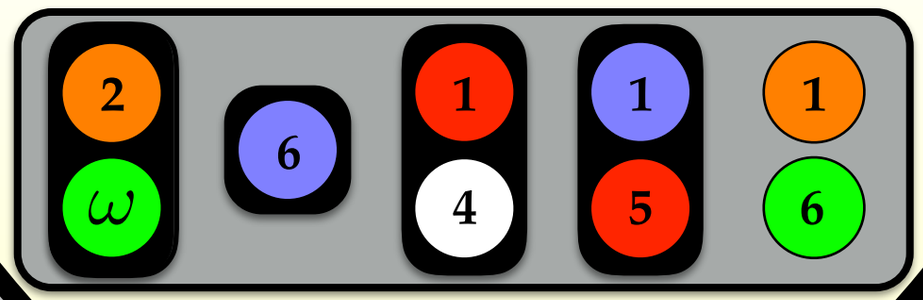
=

Timed … Computing Predecessors

Pre_{time}







Timed Petri Nets

Model

Configurations

Transitions

Ordering

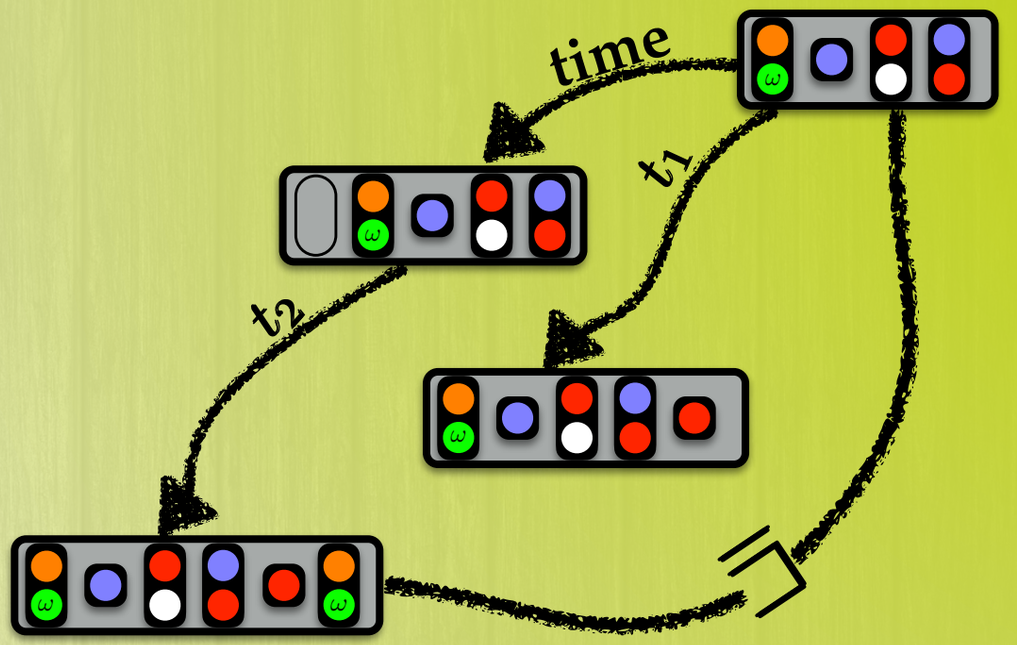
Monotoncity

Upward Closed Sets

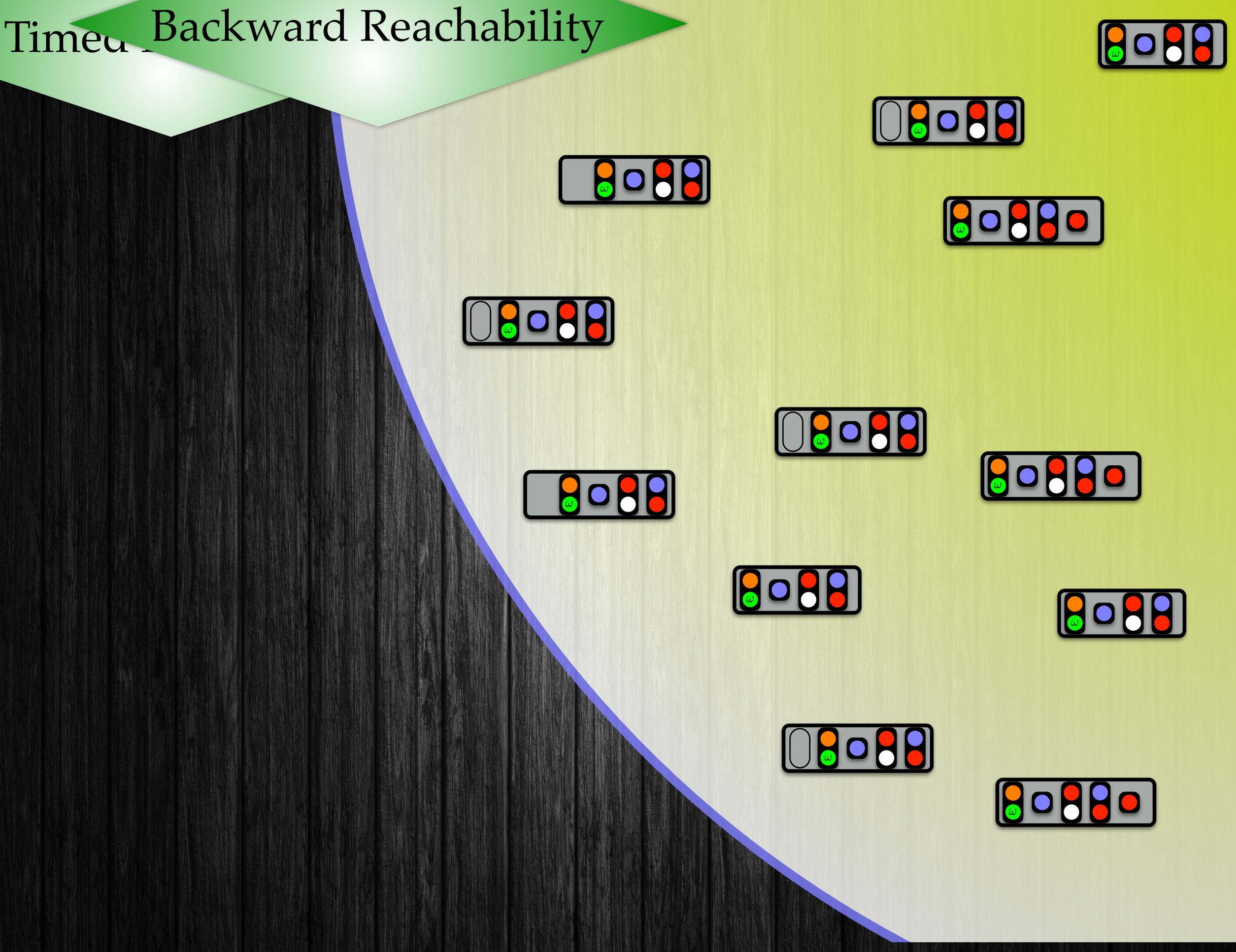
Computing Predecessors

Backward Reachability

Timed … Backward Reachability



Timed … Backward Reachability



Timed … Backward Reachability

symbolic representation =
finite words over finite multisets

Termination:
finite words over finite multisets
well quasi-ordered