Cut-Offs in Parameterized Verification

Parosh Aziz Abdulla Frédéric Haziza Lukáš Holík

Parameterized System

Analysis / Abstraction

Features

Completeness

Parameterized System

Parameterized Systems:

- **unbounded size**
- **paremeterized verification: verify correctness for all sizes**

Parameterized Systems:

- **unbounded size**
- **paremeterized verification: verify correctness for all sizes**

sources of parameterization

Parameterized Systems:

- **unbounded size**
- **paremeterized verification: verify correctness for all sizes**

sources of parameterization

unbounded numbers of processes:

- **multi-threaded programs**
- **concurrent libraries**
- **Petri nets**

Parameterized Systems:

- **unbounded size**
- **paremeterized verification: verify correctness for all sizes**

sources of parameterization

unbounded numbers of processes:

- **multi-threaded programs**
- **concurrent libraries**
- **Petri nets**

unbounded data structures:

- **queues (protocols)**
- **stacks (recursion)**
- **clocks (timed systems)**
- **lists, trees, graphs (heaps)**

Parameterized Systems:

- **unbounded size**
- **paremeterized verification: verify correctness for all sizes**

sources of parameterization

unbounded numbers of processes:

- **multi-threaded programs**
- **concurrent libraries**
- **Petri nets**

unbounded data structures:

- **queues (protocols)**
- **stacks (recursion)**
- **clocks (timed systems)**
- **lists, trees, graphs (heaps)**

multiple sources

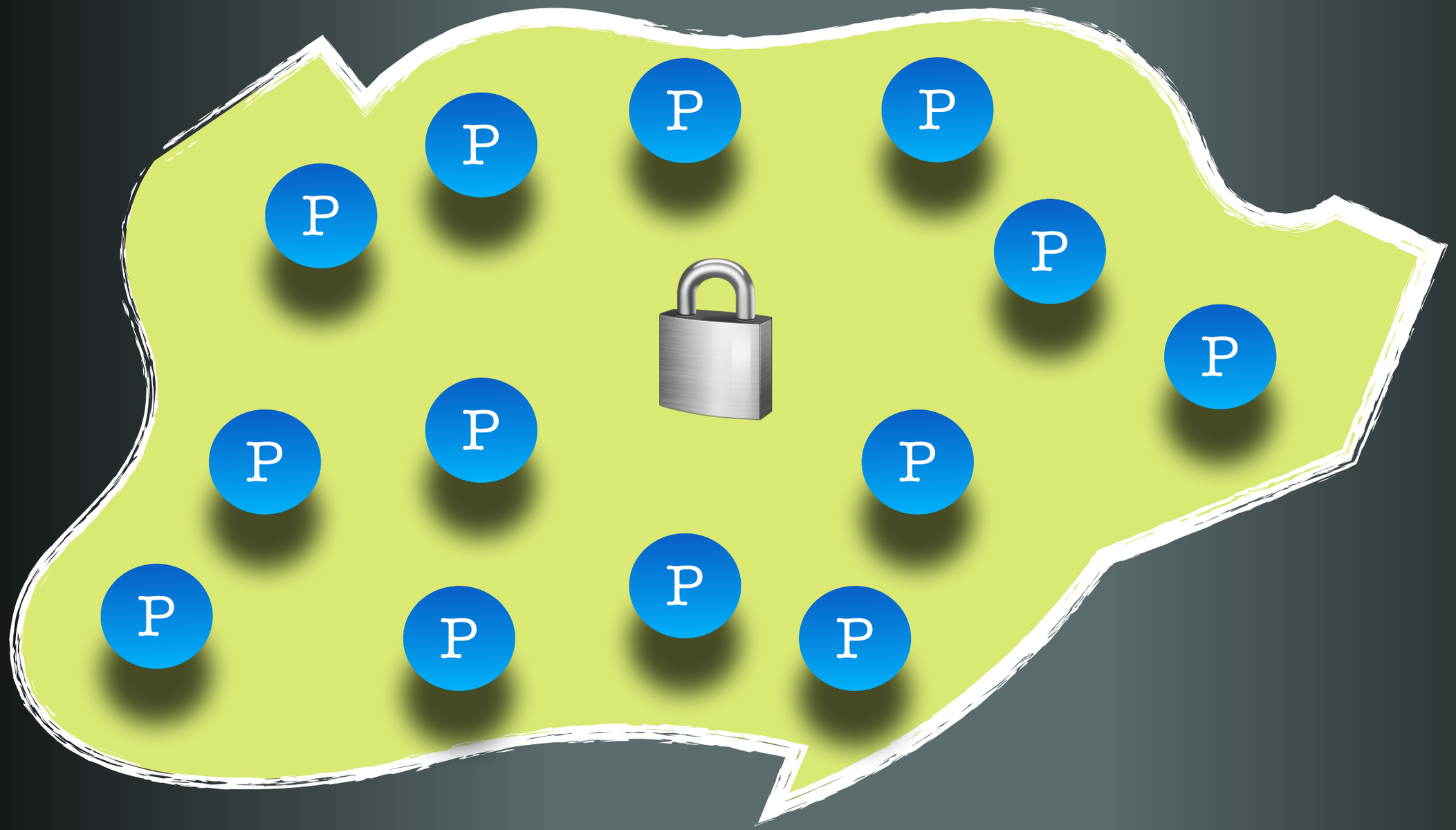
- **timed Petri nets**
- **recursive programs with data**
- **channels with time stamps**
- **etc**

Parameterized Systems

Param Unbounded Number of Processes

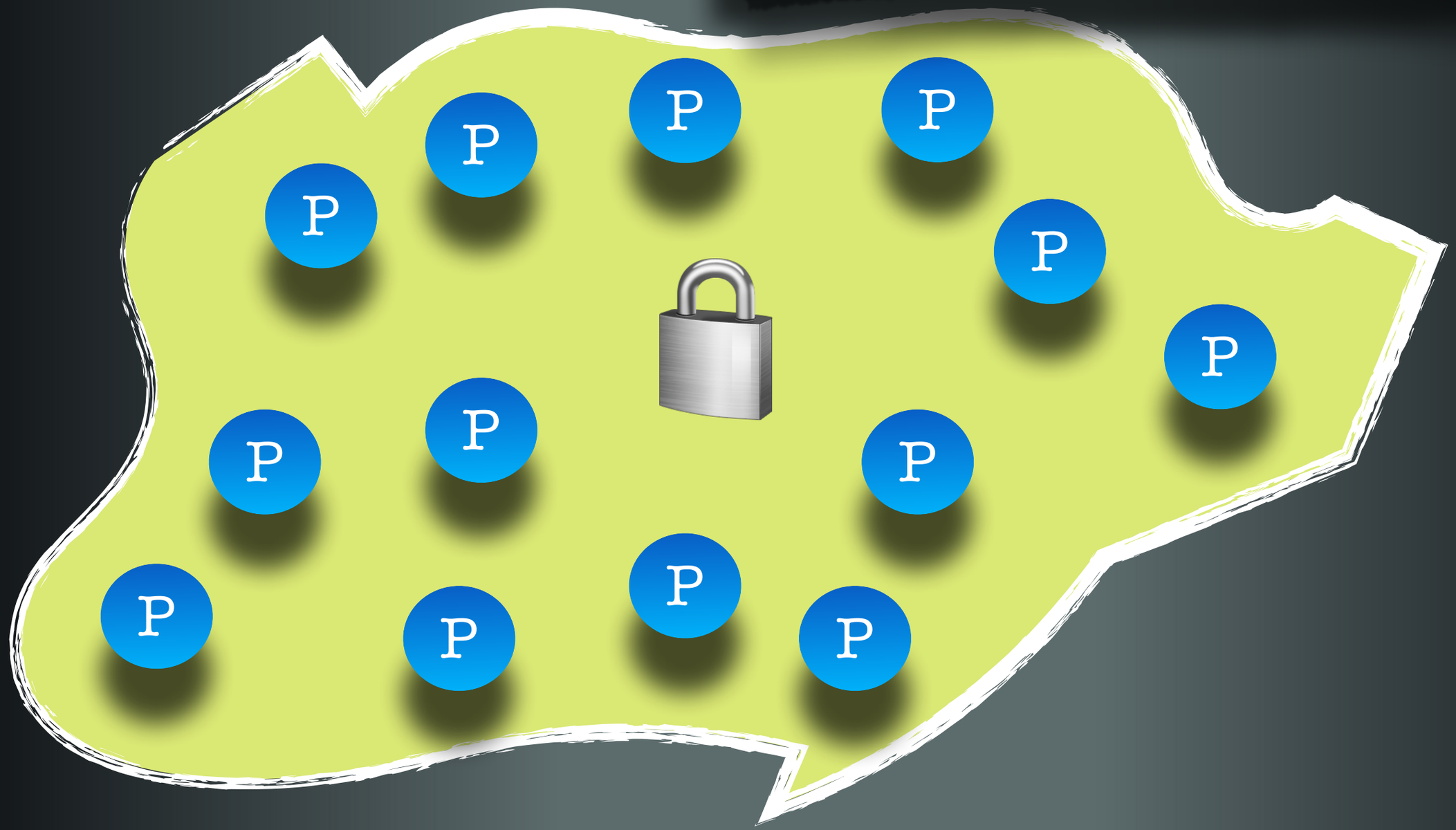
Param Unbou Mutual Exclusion Protocols

Param Unbou Mutual Exclusion Protocols

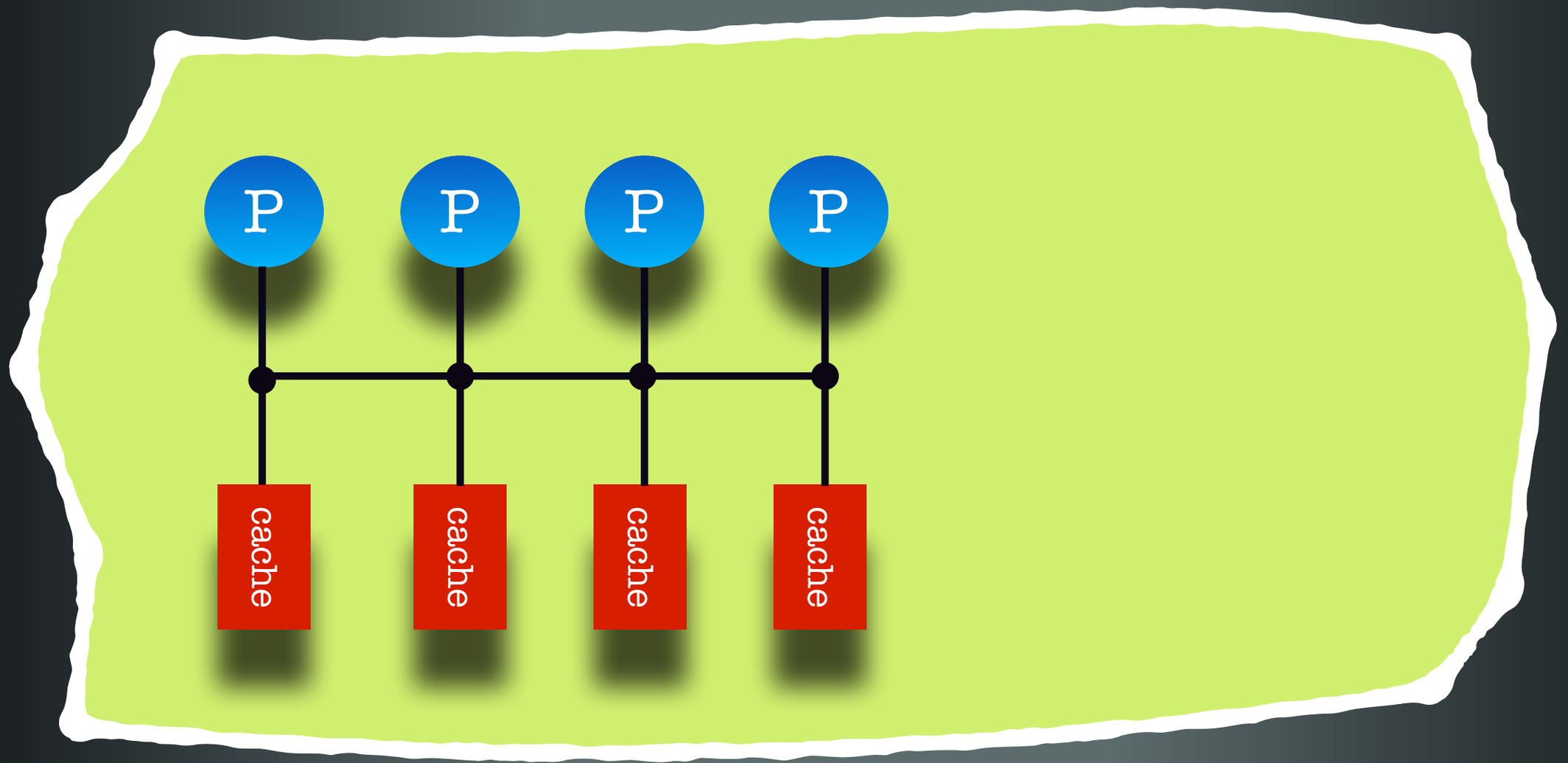


Param Unbou Mutual Exclusion Protocols

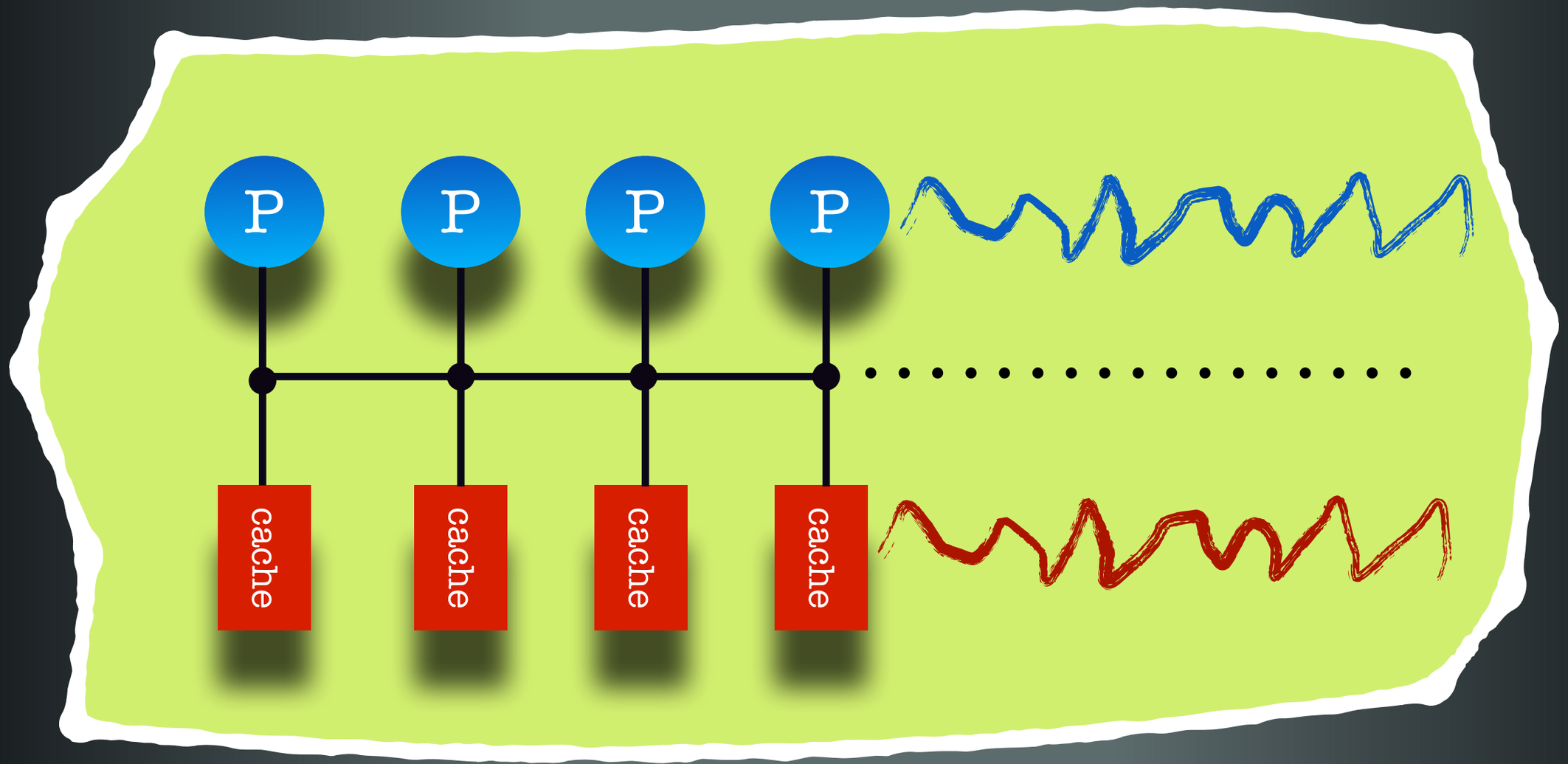
- unbounded number of processes
- correctness:
  - lock taken by at most one process



Param Unbou Cache Coherence Protocols

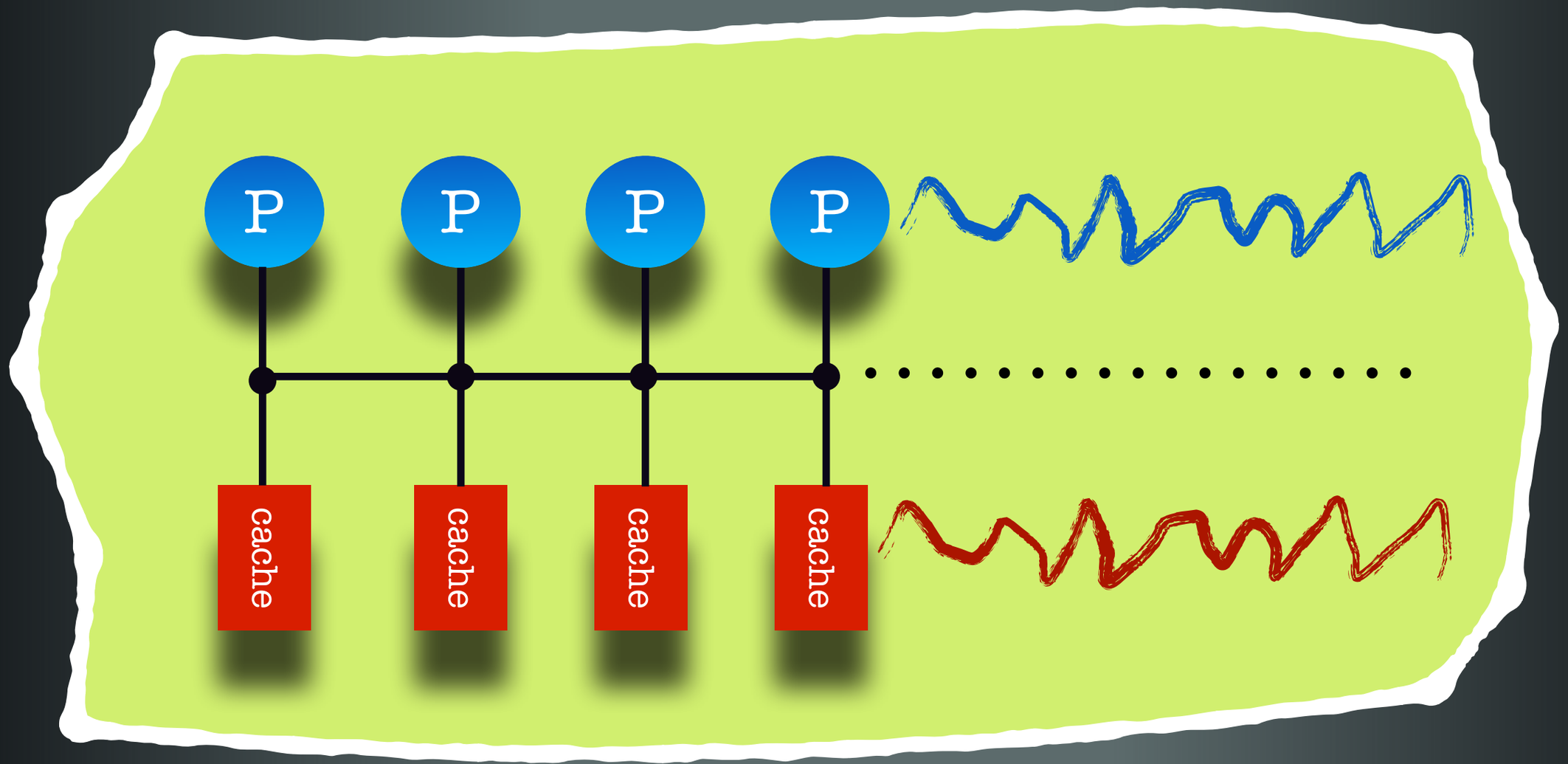


Param Unbou Cache Coherence Protocols



Param Unbou Cache Coherence Protocols

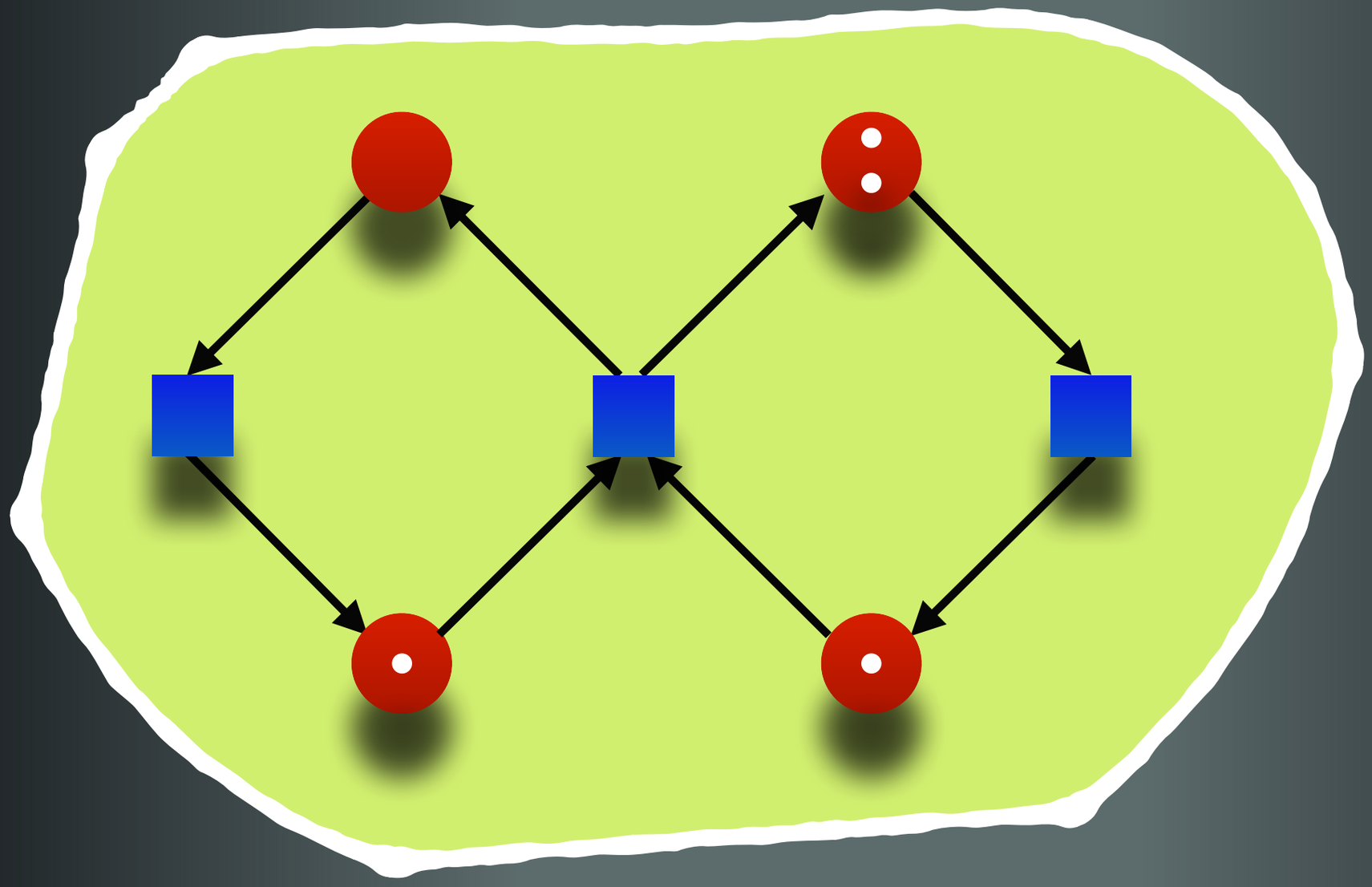
- unbounded number of processes
- correctness:
  - exclusive ownership: at most process



Param Unbounded Number of Processes

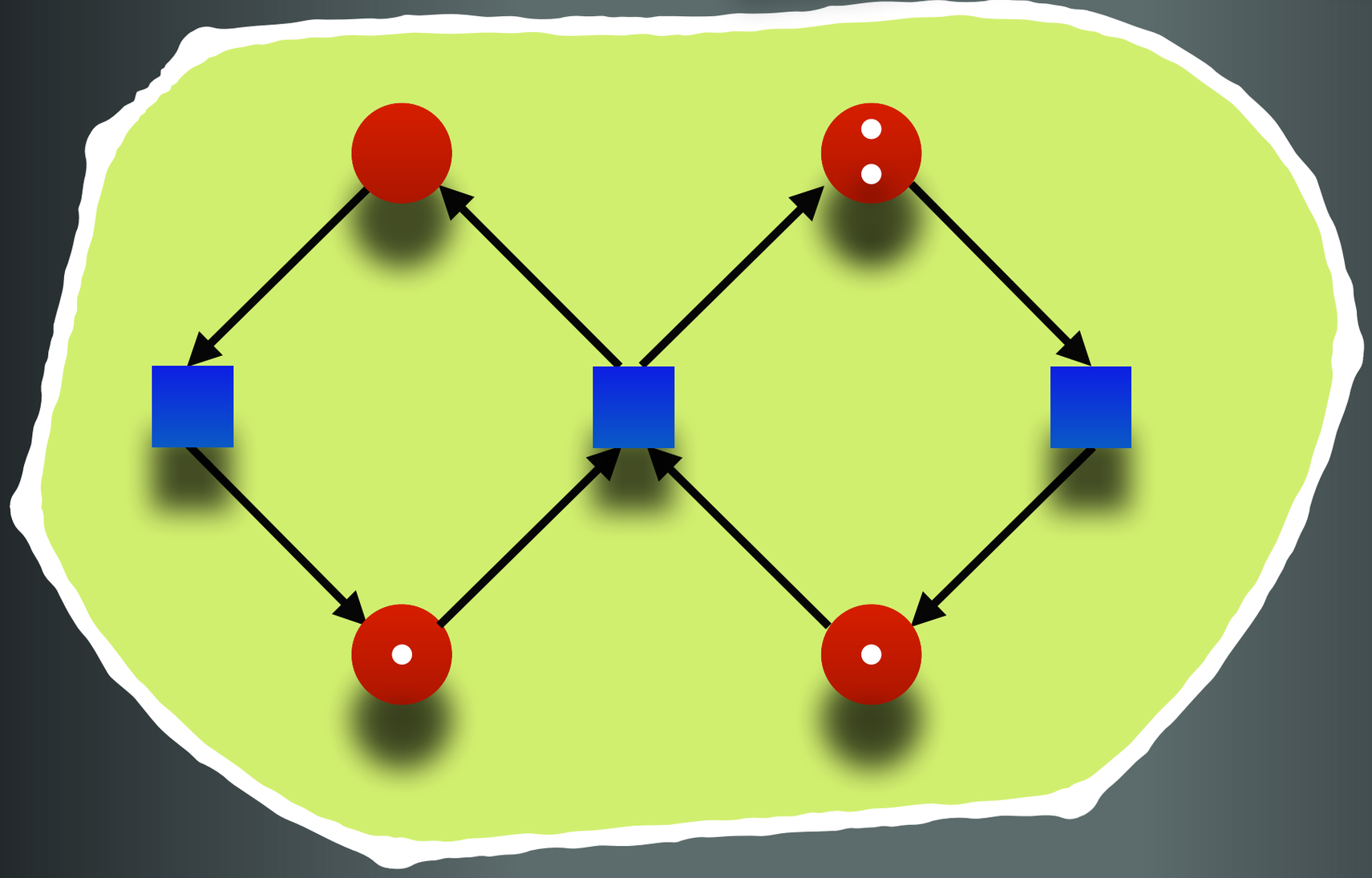


Param Unbounded Nu Petri Nets



Param Unbounded Nu Petri Nets

- unbounded numbers of tokens
- correctness:
  - coverability



Parameterized Systems

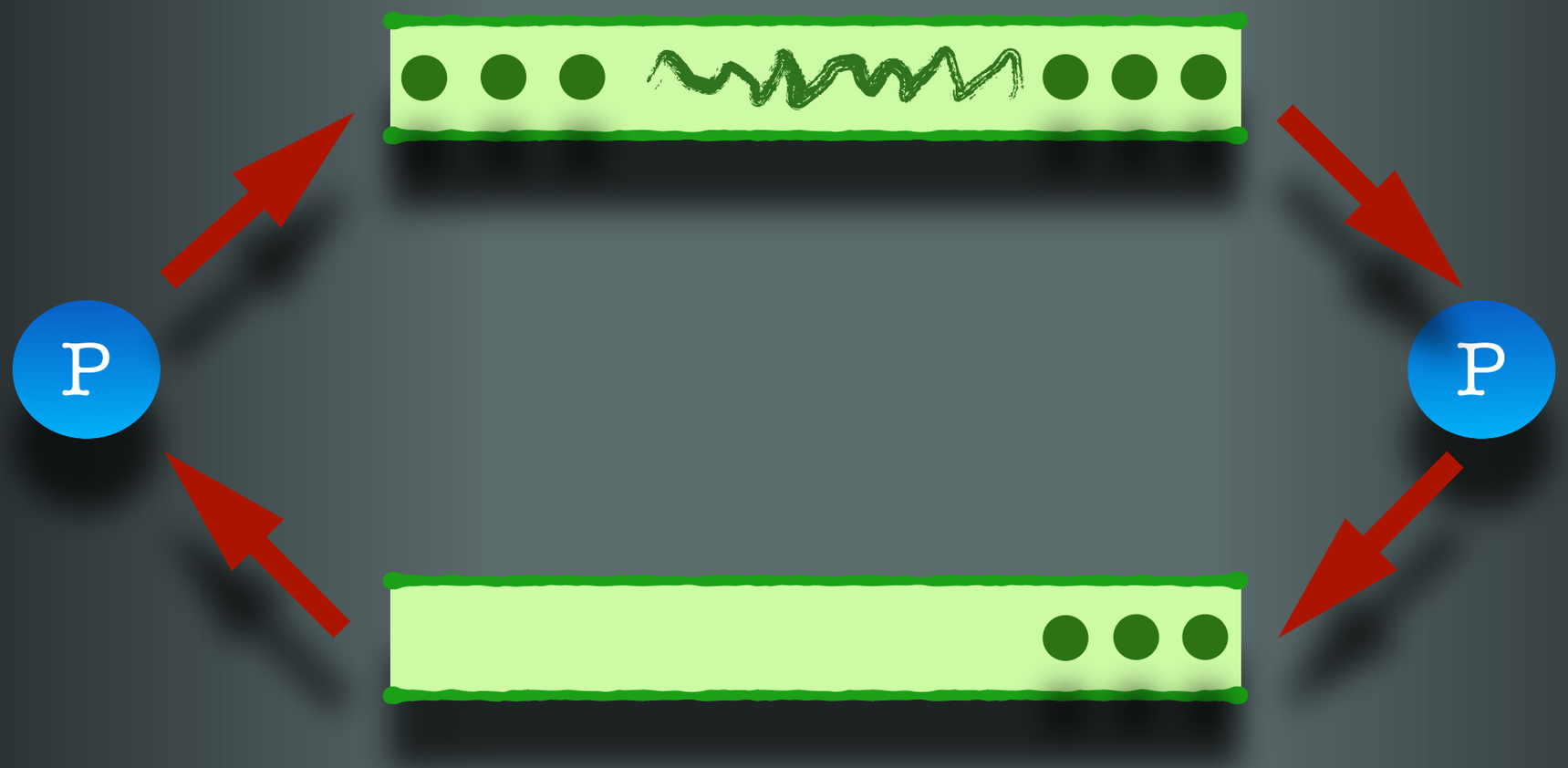
Param Unbounded Data Structures

Param Unbounded Da Unbounded Channels

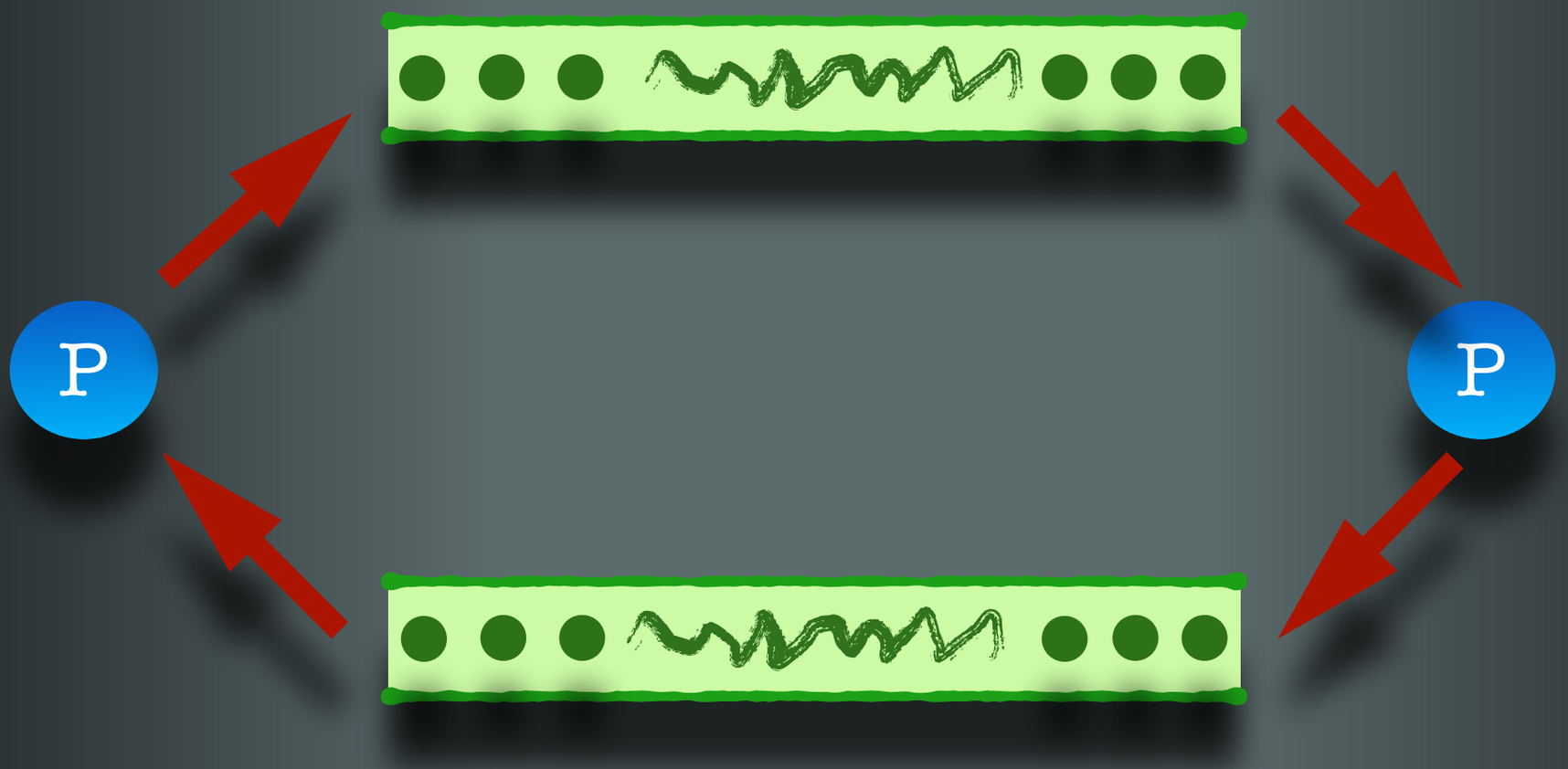
Param Unbounded Da Unbounded Channels



Param Unbounded Da Unbounded Channels

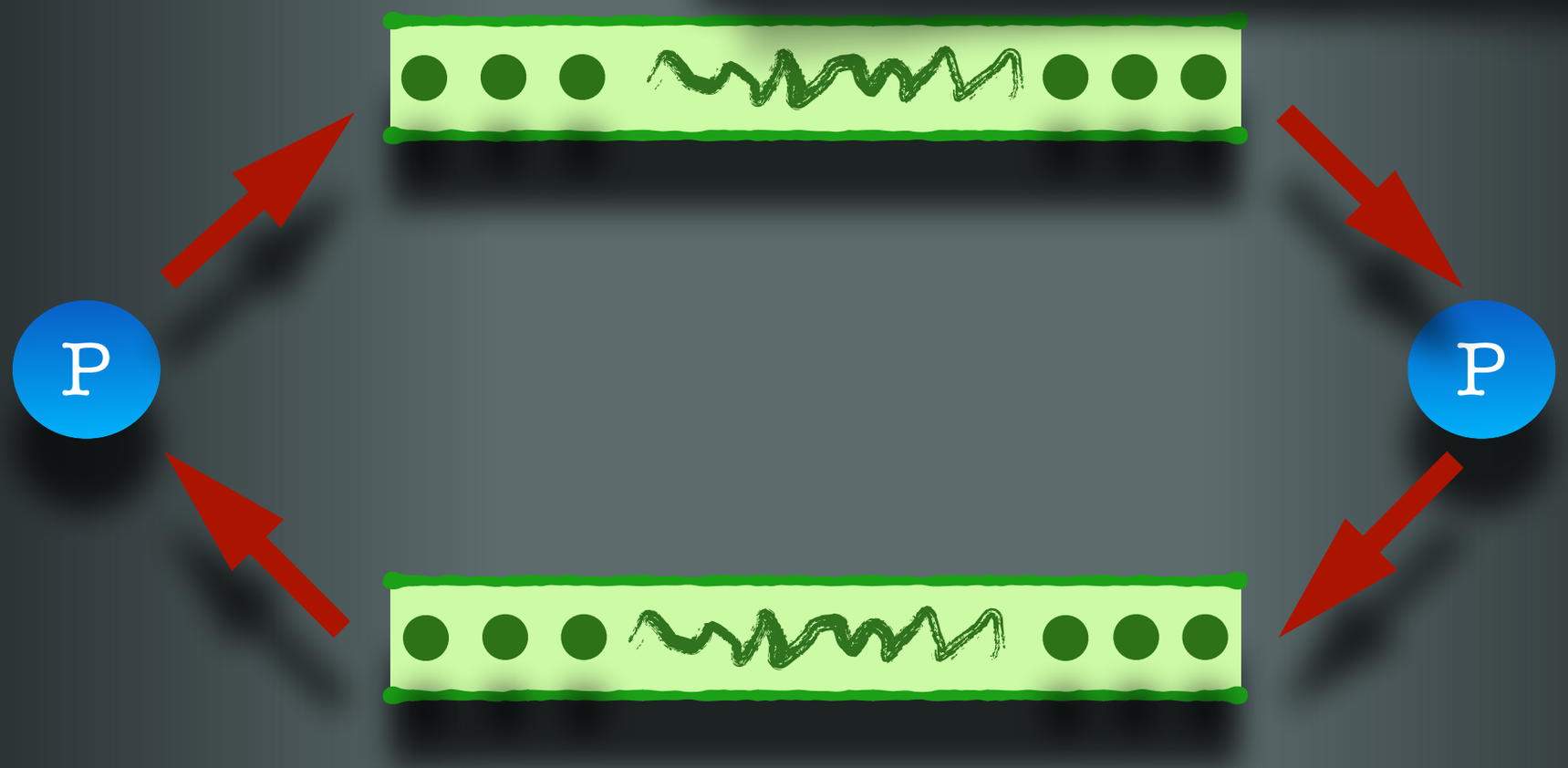


Param Unbounded Da Unbounded Channels



Param Unbounded Da Unbounded Channels

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

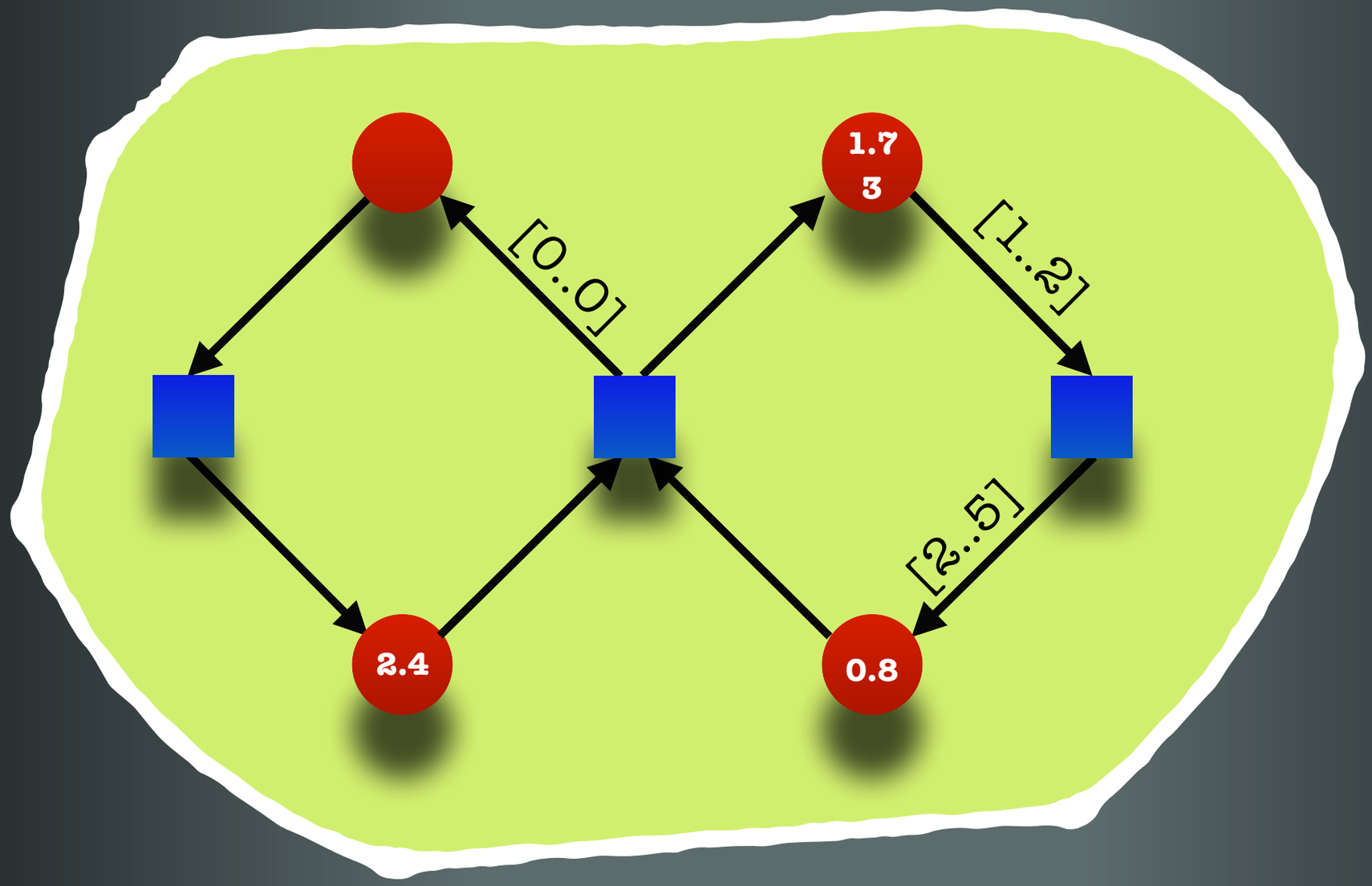


Parameterized Systems

Paran Multiple Sources

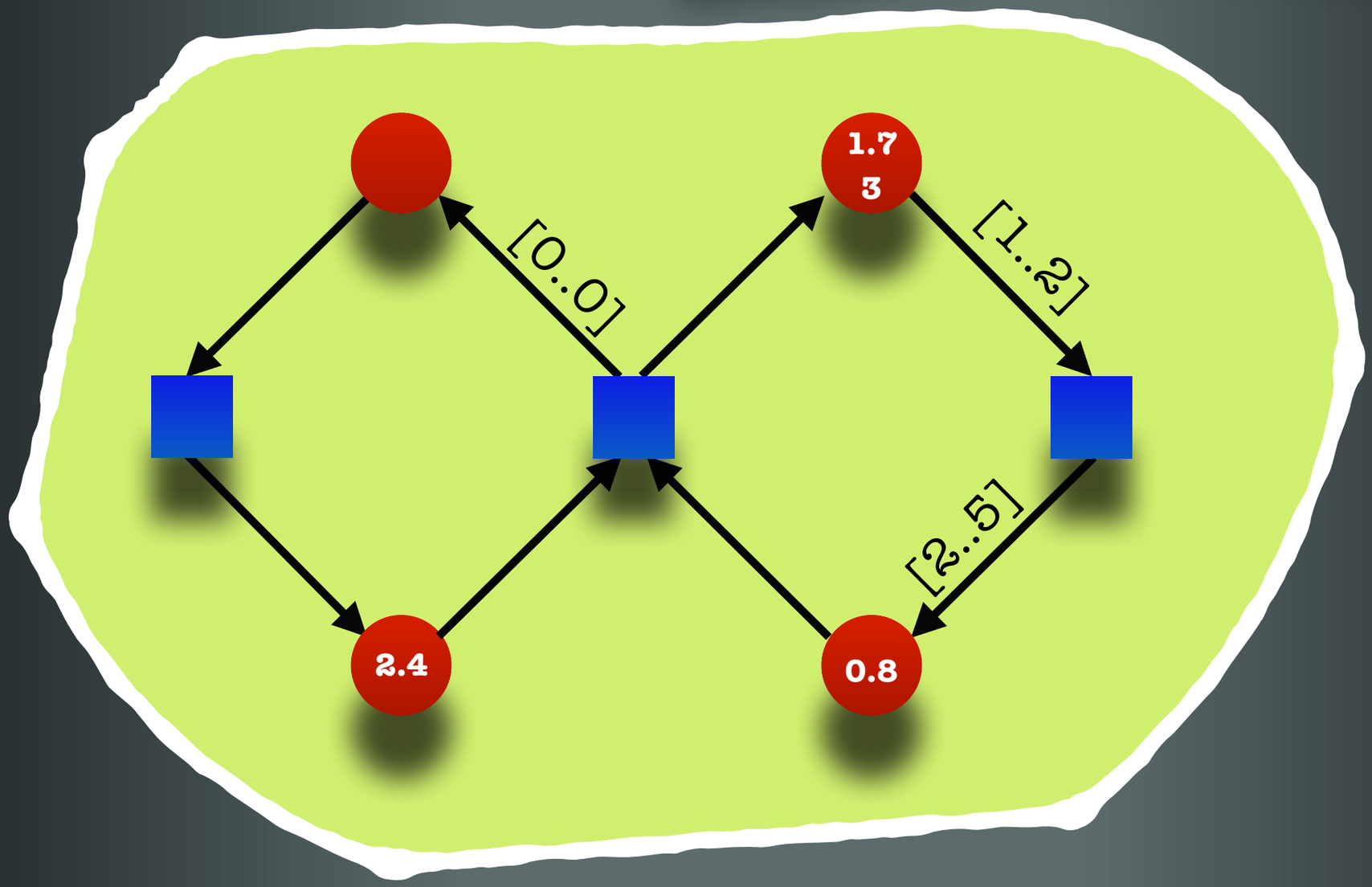
Paran Multiple Sourc Timed Petri nets

Paran Multiple Sourc Timed Petri nets



Paran Multiple Sourc Timed Petri nets

- unbounded number of clocks
- multiply unbounded



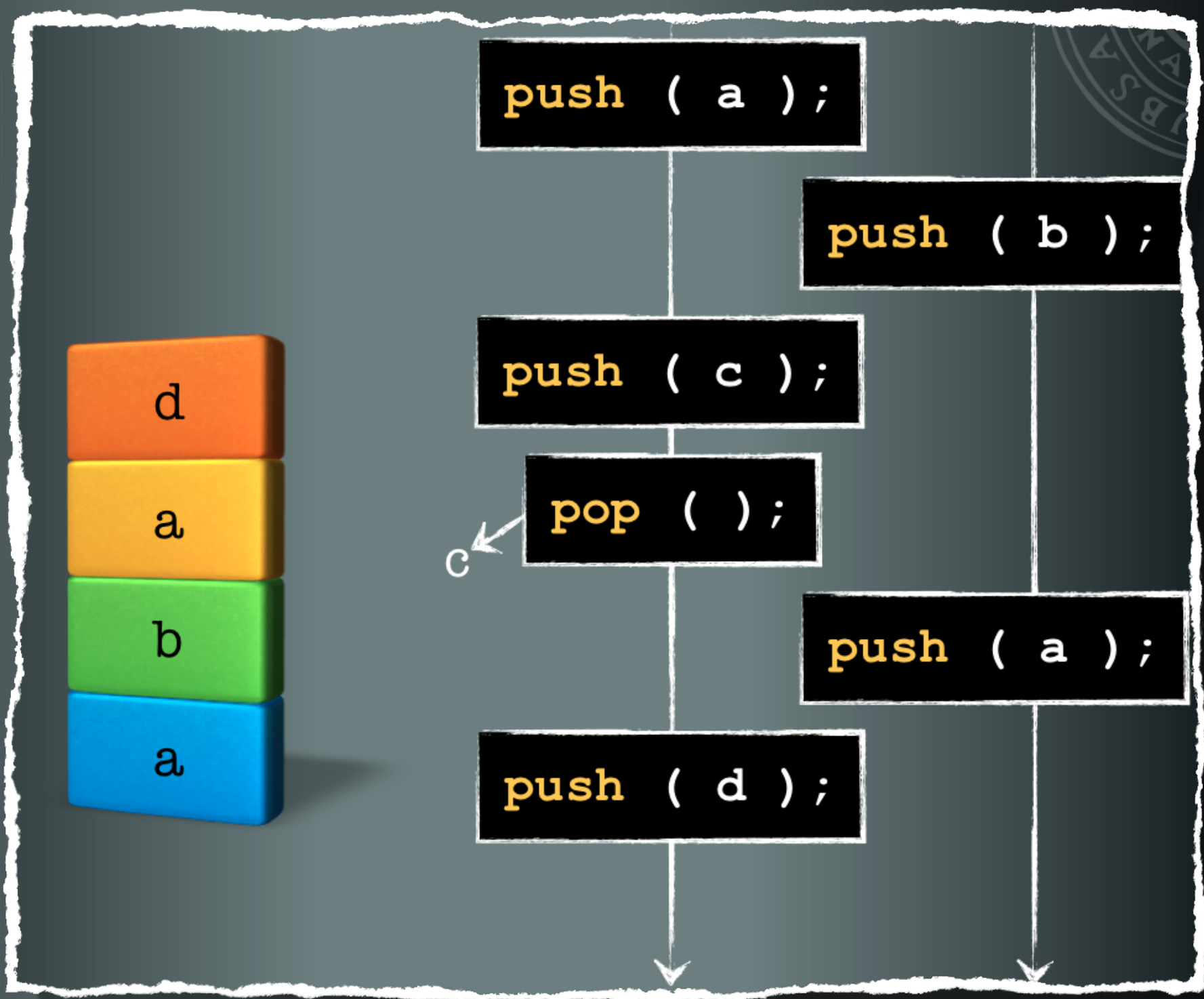
Paran Multiple Sources

Paran

Multiple Sourc

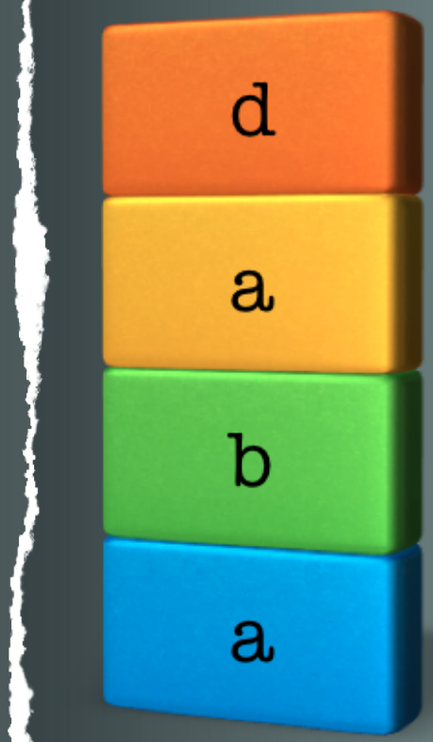
Concurrent Libraries

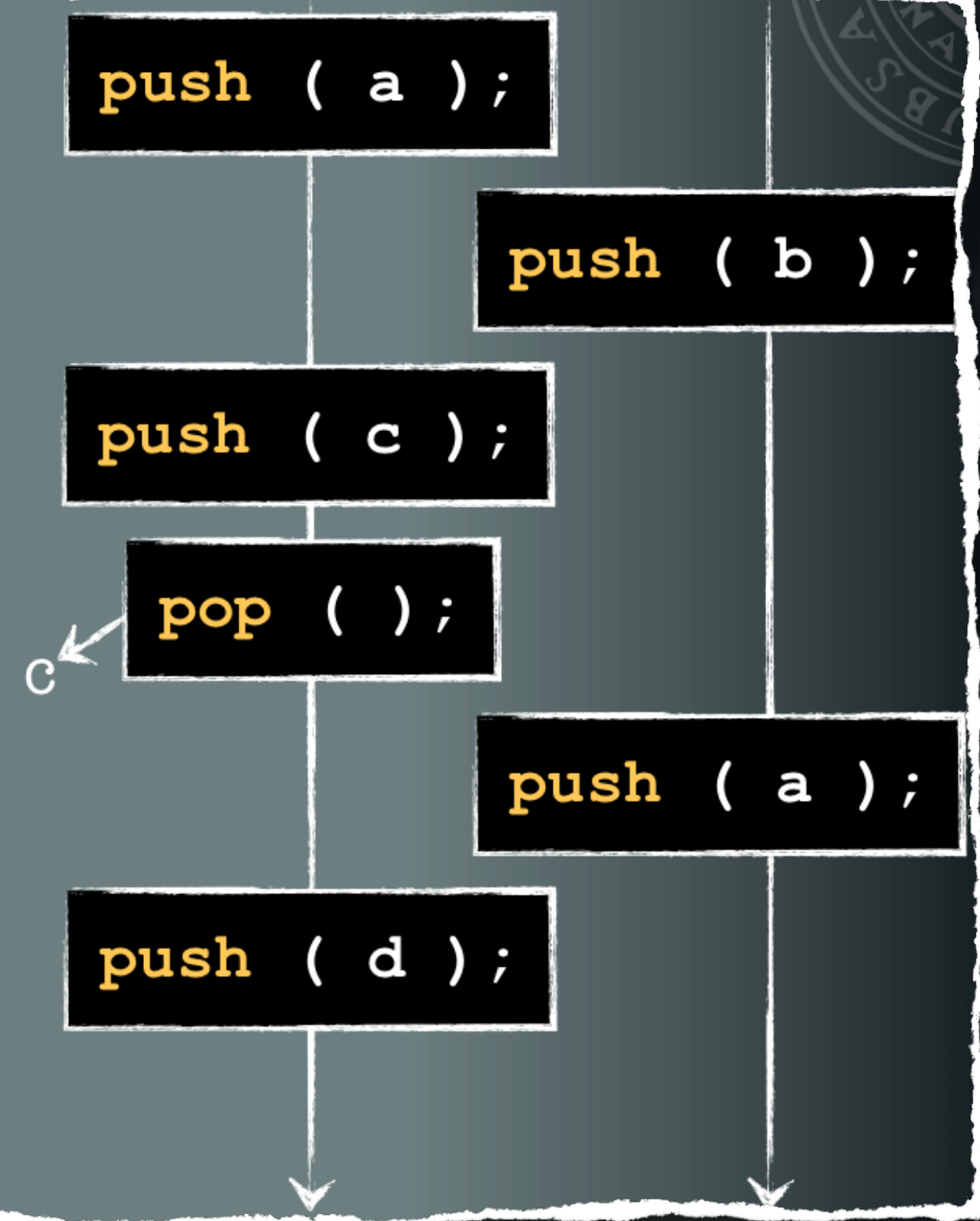
Paran Multiple Sourc Concurrent Libraries



Paran Multiple Sourc Concurrent Libraries

- unbounded heap
- unboundedly many threads
- unbounded data domain





Parameterized Systems

Param Unbounded Number of Processes



Param Unbounded Nu Literature

Param Unbounded Nu Literature

counter abstraction

- German, Sistla, 1992
- Delzanno, 2000
- Pnueli, Xu, Zuck, 2002
- Kelp, Margaria, Mendler, Gsottberger, 1997

Param Unbounded Nu Literature

finite-state abstraction

- Arons, Pnueli, Ruah, Xu, Zuck, 2001
- Bauhaus, Lakhnech, Stahl, 2002
- Lahiri, Bryant, 2001
- Clarke, Talupur, Veith, 2006

Param Unbounded Nu Literature

regular model checking

- Kesten, Maler, Marcus, Pnueli, Shahar, 1997
- Boigelot, Legay, Wolper, 2004
- Bouajjani, Habermehl, Rogalewicz, Vojnar, 2012
- Bouajjani, Jonsson, Nilsson, Touili, 2000
- A., Jonsson, Mahata, d'Orso, 2002

Param Unbounded Nu Literature

induction-based methods

- Brown, Clarke, Grunberg, 1989
- Clarke, Grunberg, Jha, 1997
- Kurshan, MacMillan, 1989
- Shtadler, Grumberg, 1989
- Wolper, Lovinfosse, 1989

Param Unbounded N Literature

symmetry reduction

- Clarke, Jha, Enders, Filkorm, 1996
- German, Sistla, 1996
- Ip, Dill, 1996

Param Unbounded Nu Literature



Petri net coverability

- Kloos, Majumdar, Niksic, Piskac, 2013
- Kaiser, Kroening, Wahl, 2012
- Blondin, Finkel, Hasse, Haddad, 2016
- Ganty, Raskin, Van Begin, 2006

Param Unbounded Nu Literature

cut-off methods

- Emerson, Namjoshi, 1995
- Emerson, Kahlon, 2000

cut-off methods

- Emerson, Namjoshi, 1995
- Emerson, Kahlon, 2000

this tutorial

cut-off methods

- Emerson, Namjoshi, 1995
- Emerson, Kahlon, 2000

Param Unbounded Number of Processes

Param Unbounded Numb Hierarchy

Param Unbounded Num Hierarchy

Processes:
- finite-state
- infinite-state
- dynamic

Topology:
- array
- tree
- graph
- multiset

Communication:
- rendez-vous
- broadcast
- global

Param Unbounded Num Hierarchy

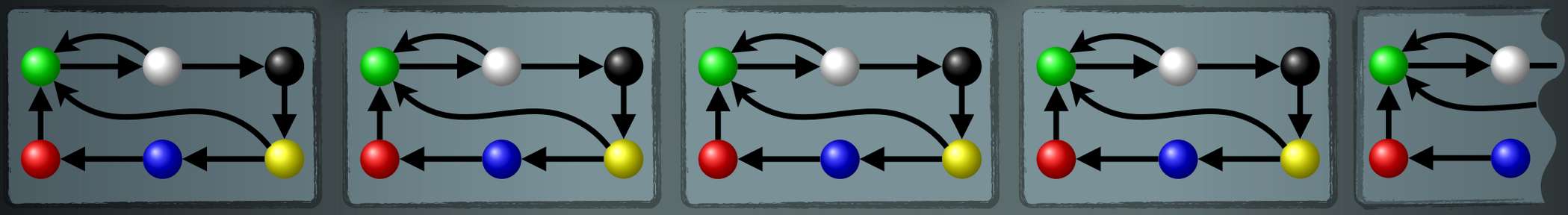
Processes:
- finite-state
- infinite-state
- dynamic

Topology:
- array
- tree
- graph
- multiset

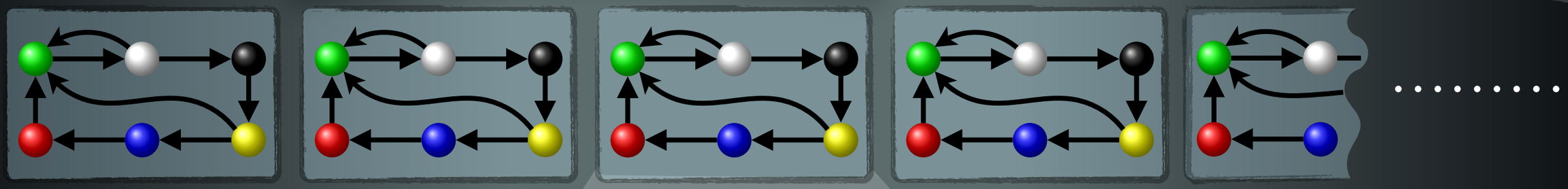
Communication:
- rendez-vous
- broadcast
- global

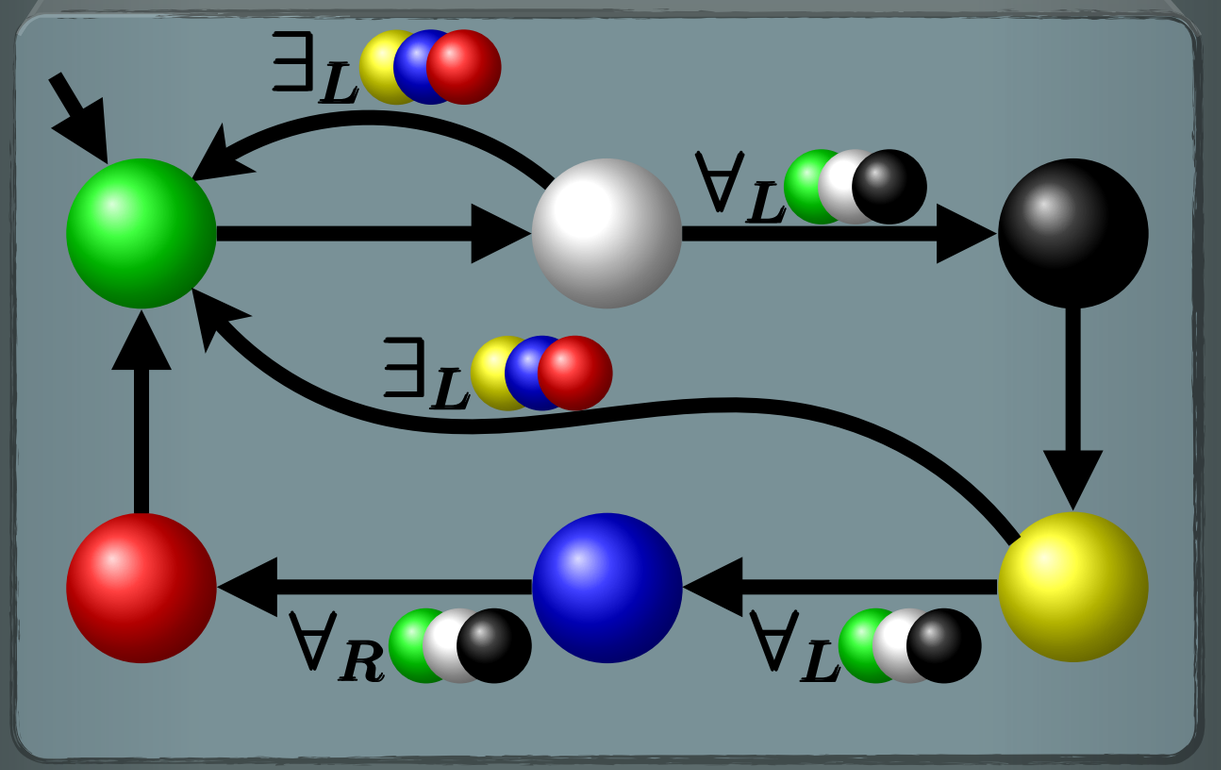
finite-state + array + global

finite-state + array + global

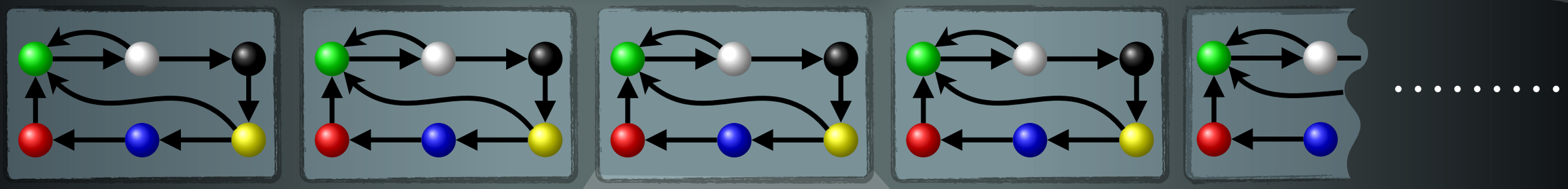


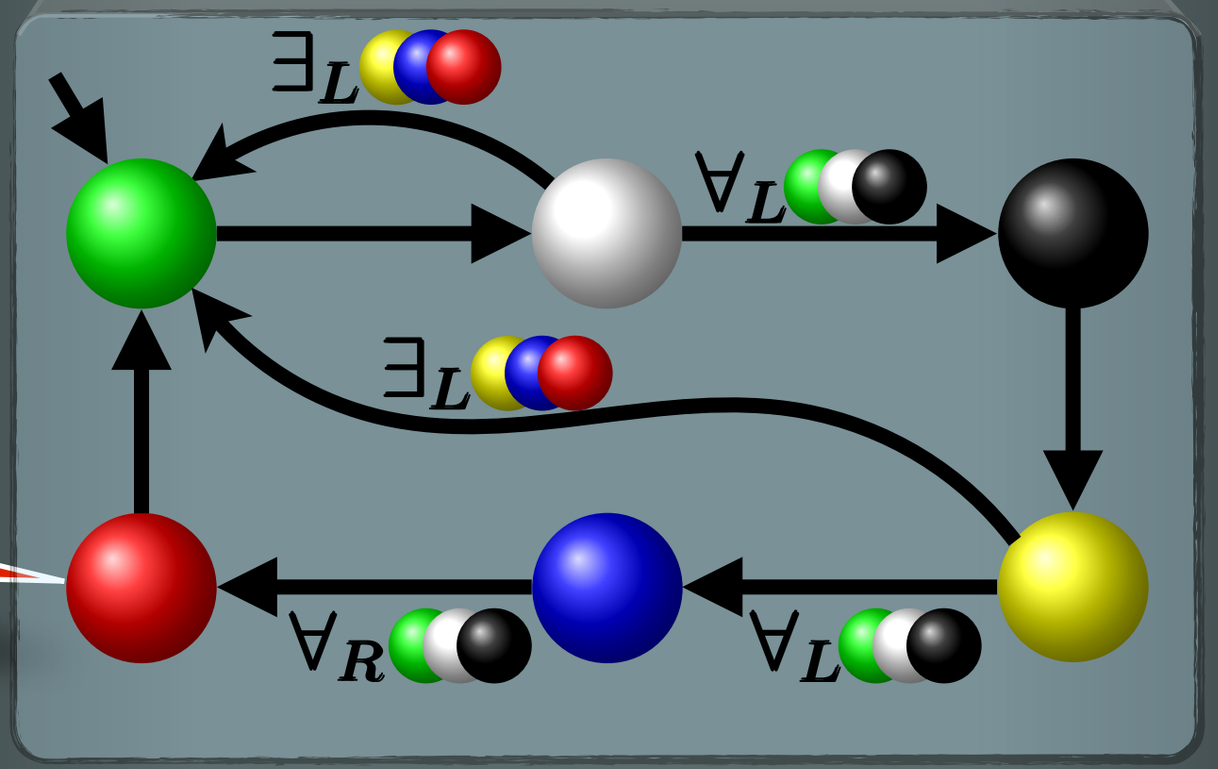
finite-state + array + global





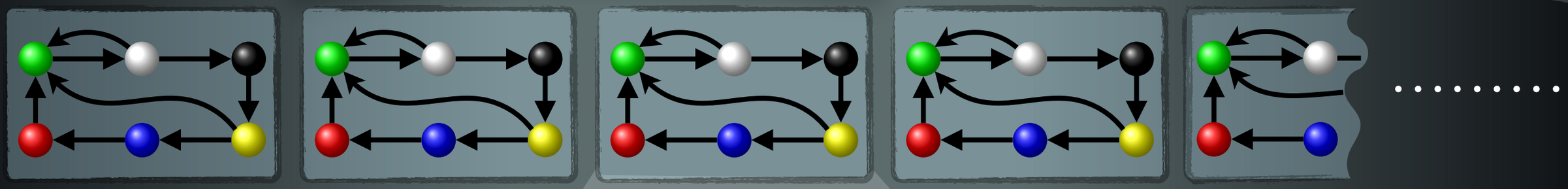
finite-state + array + global

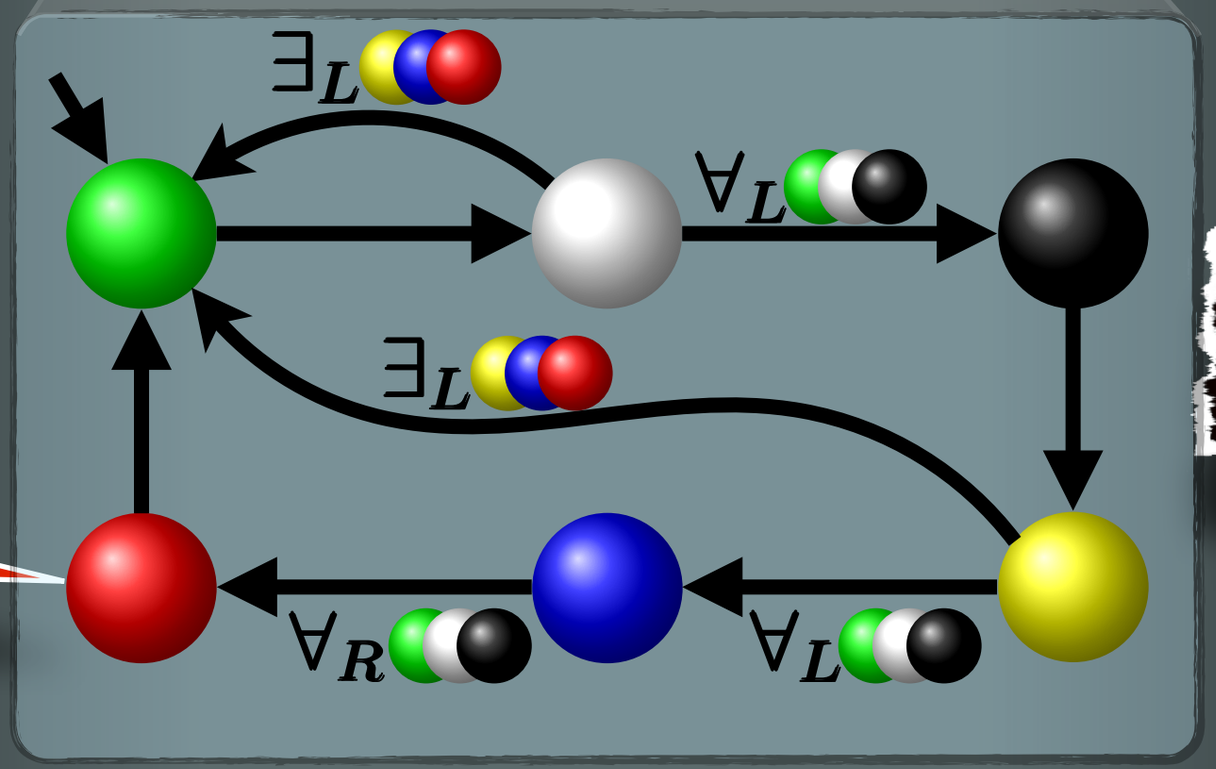




critical section

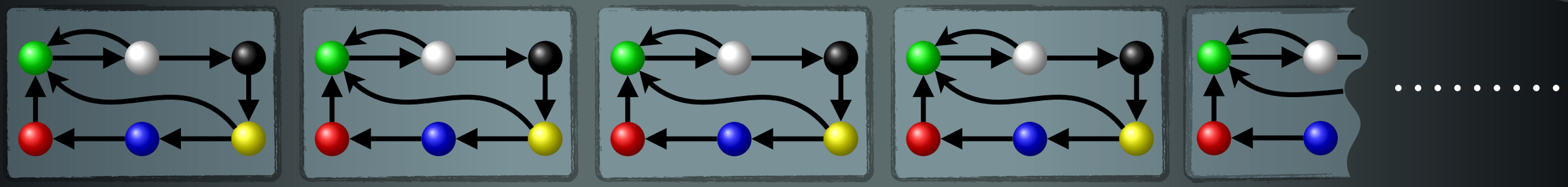
finite-state + array + global

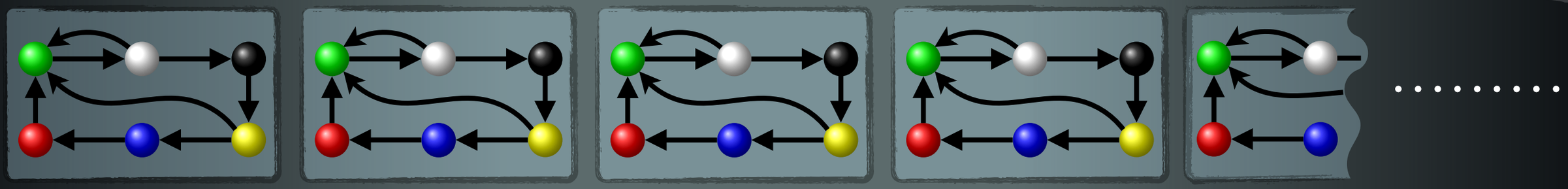


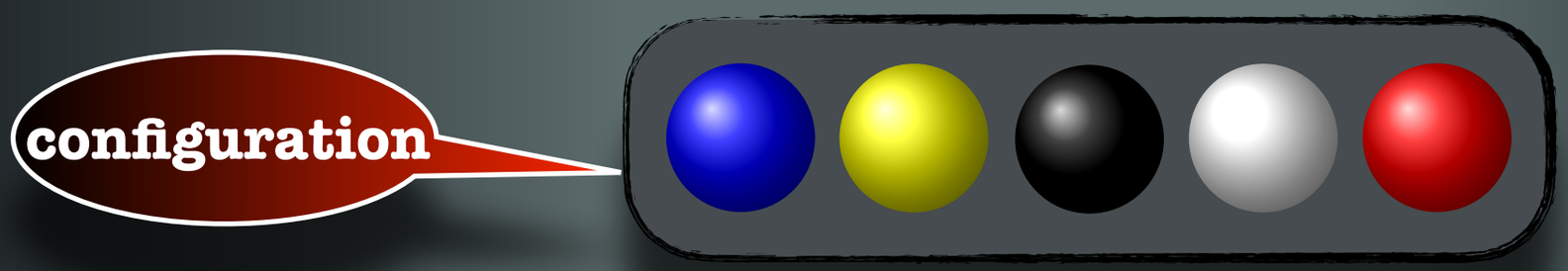


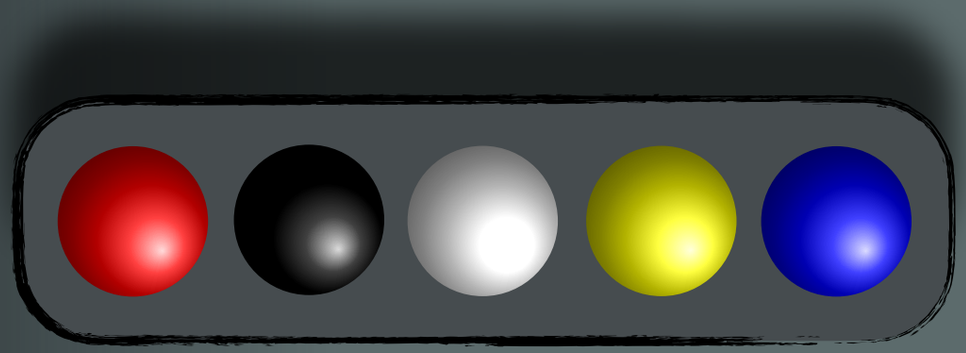
critical section

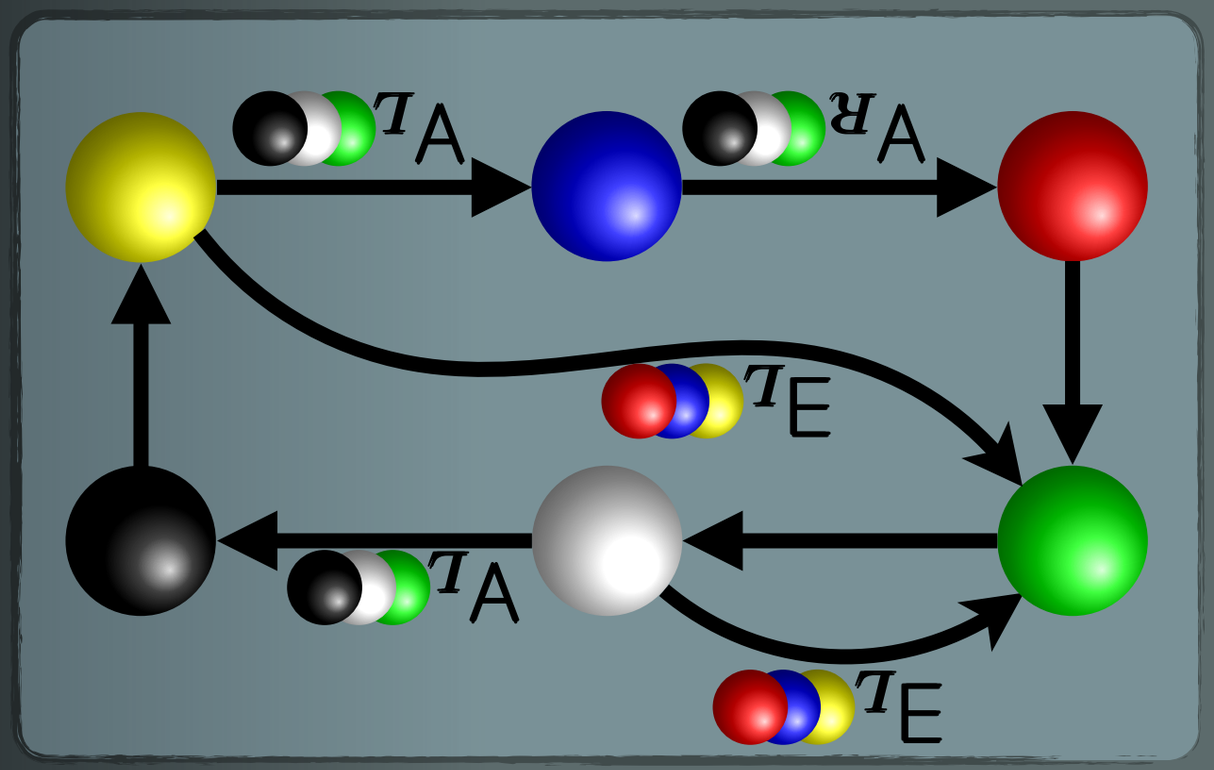
Burns' mutual exclusion protocol





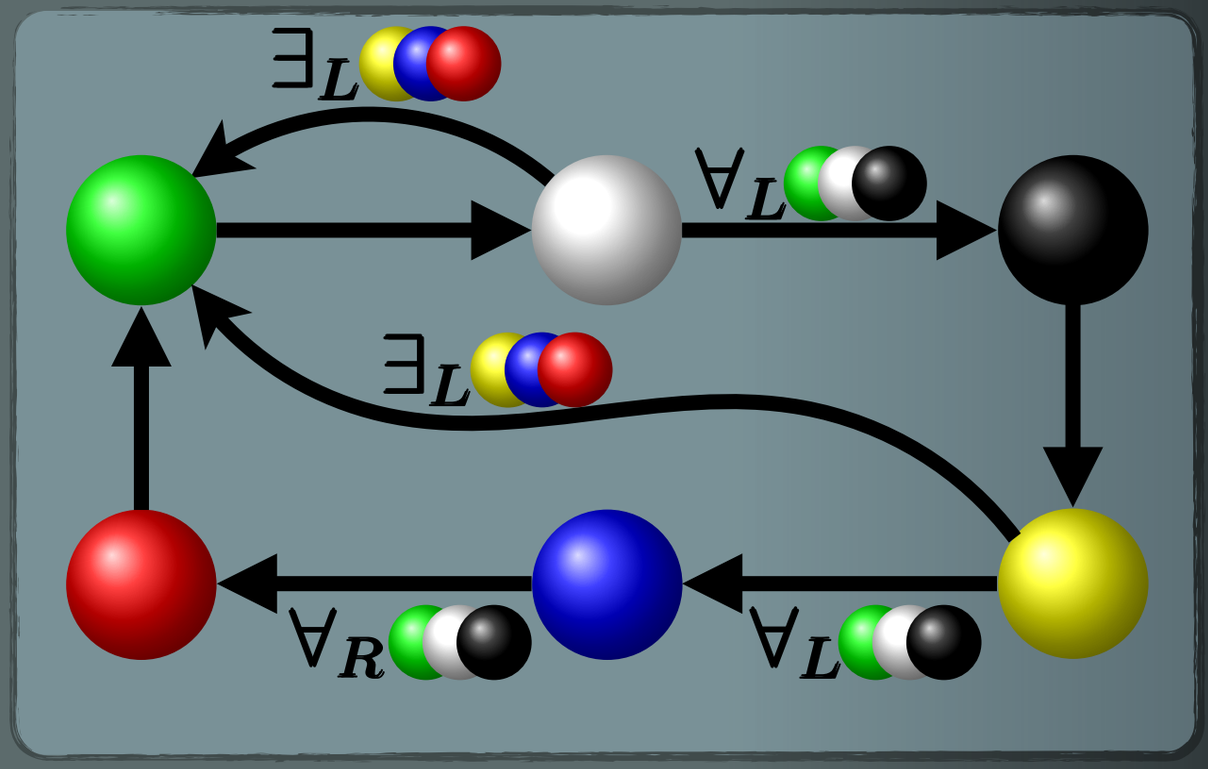

configuration



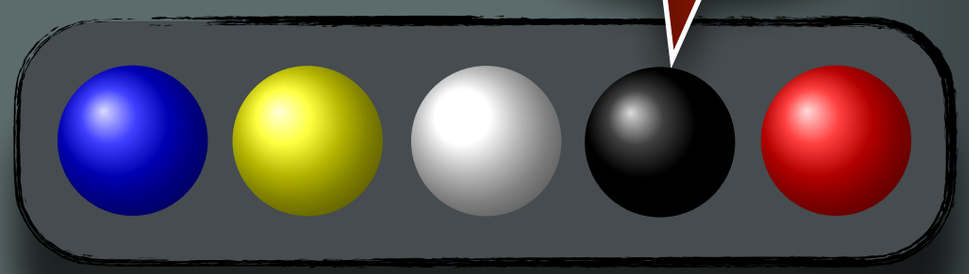


post

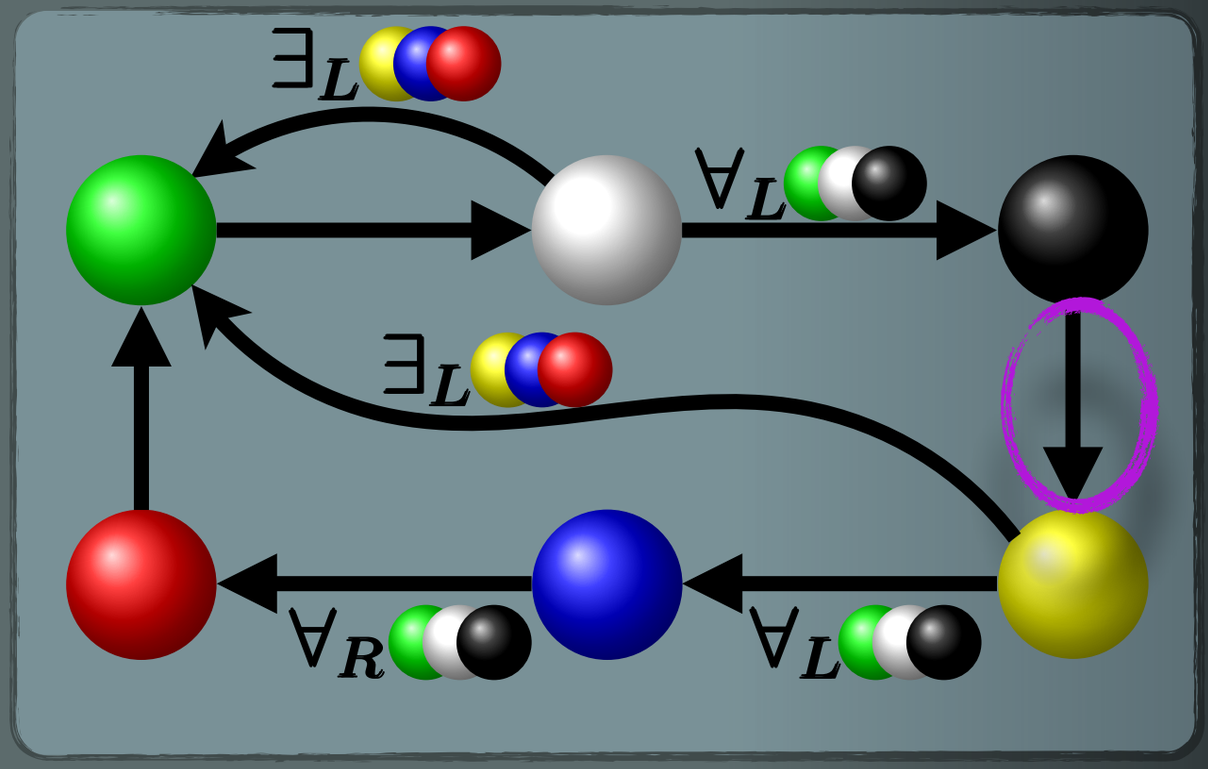
post



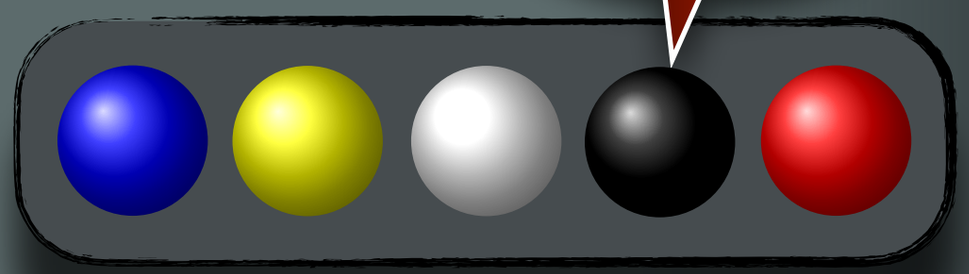
active process



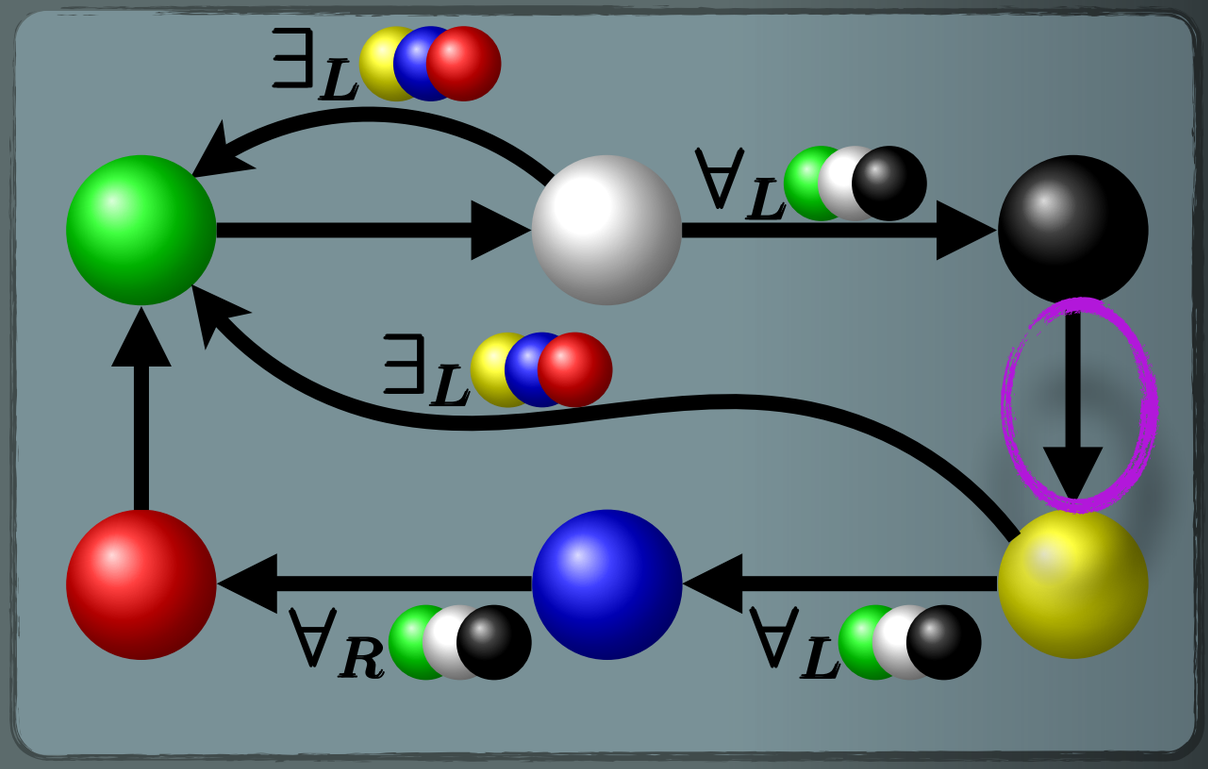
post



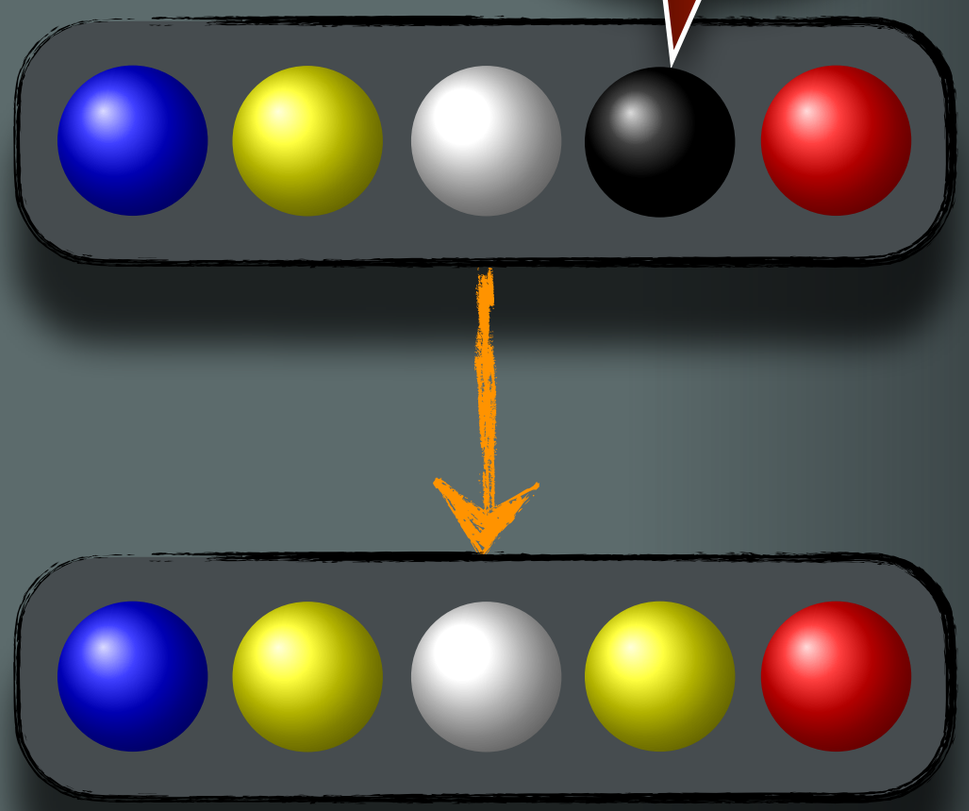
active process



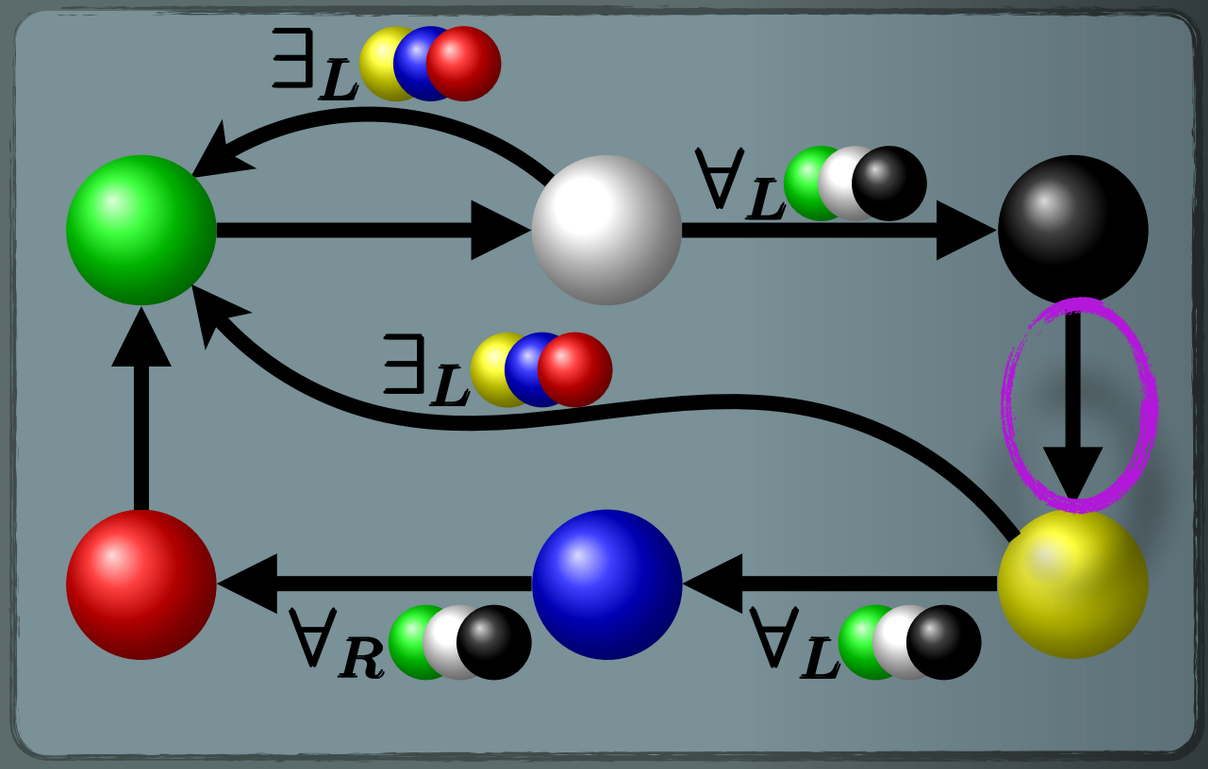
post



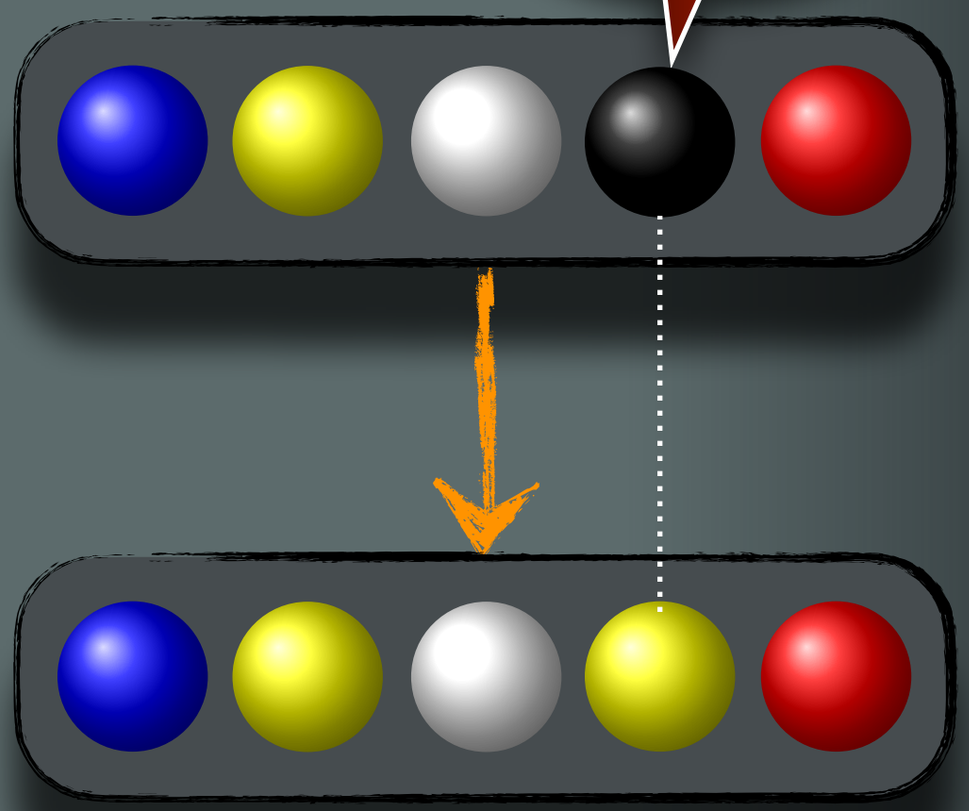
active process



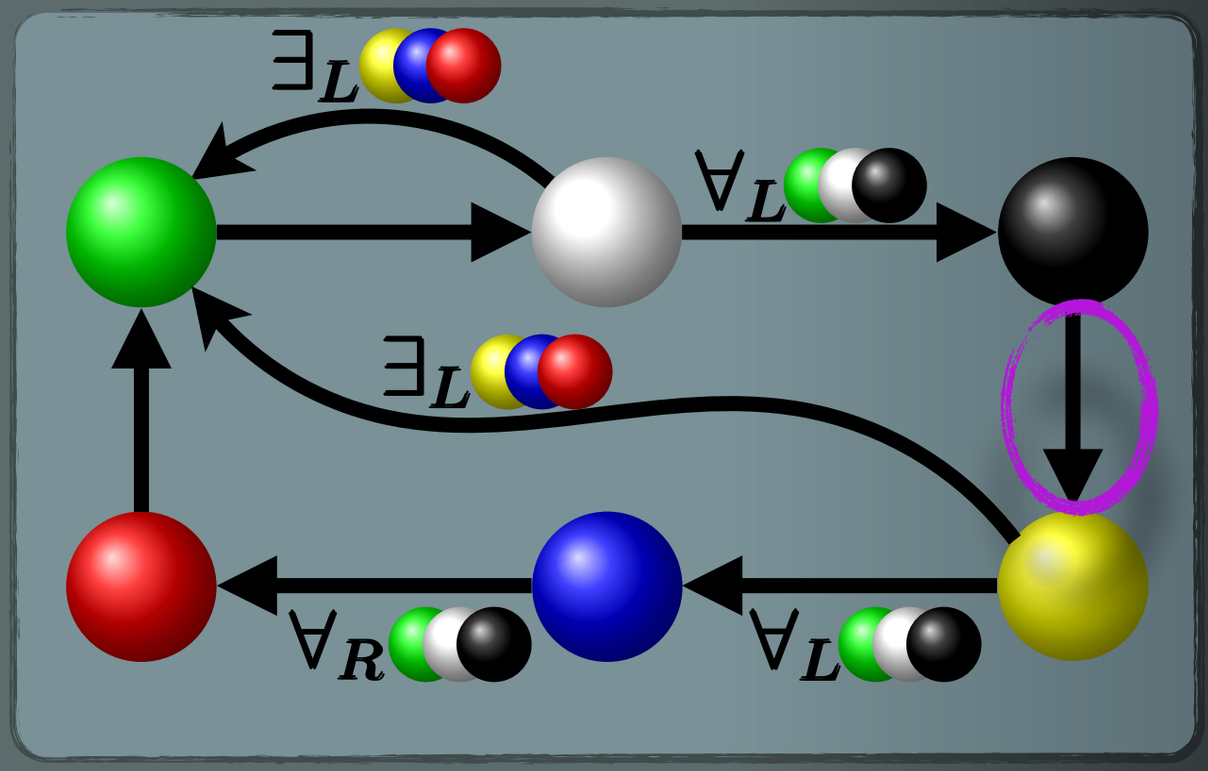
post



active process

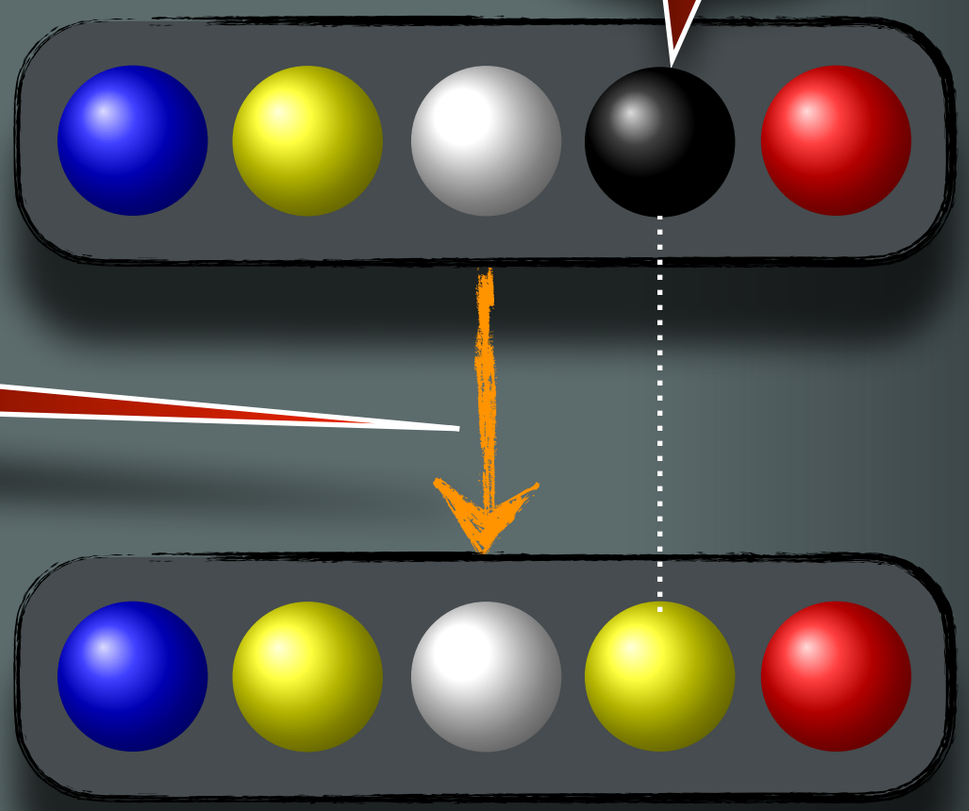


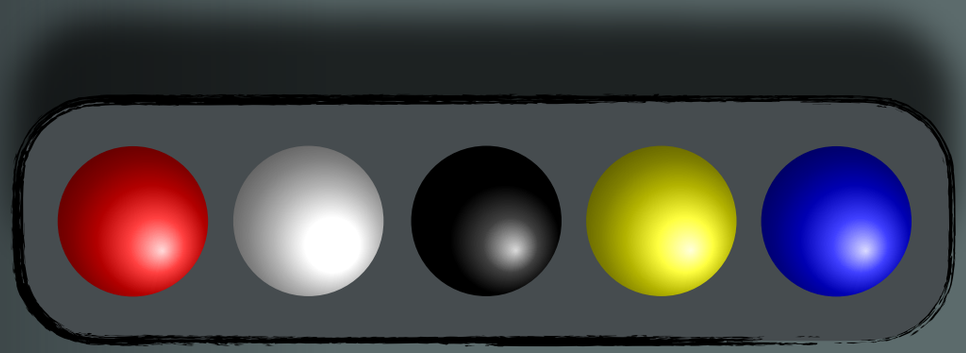
post

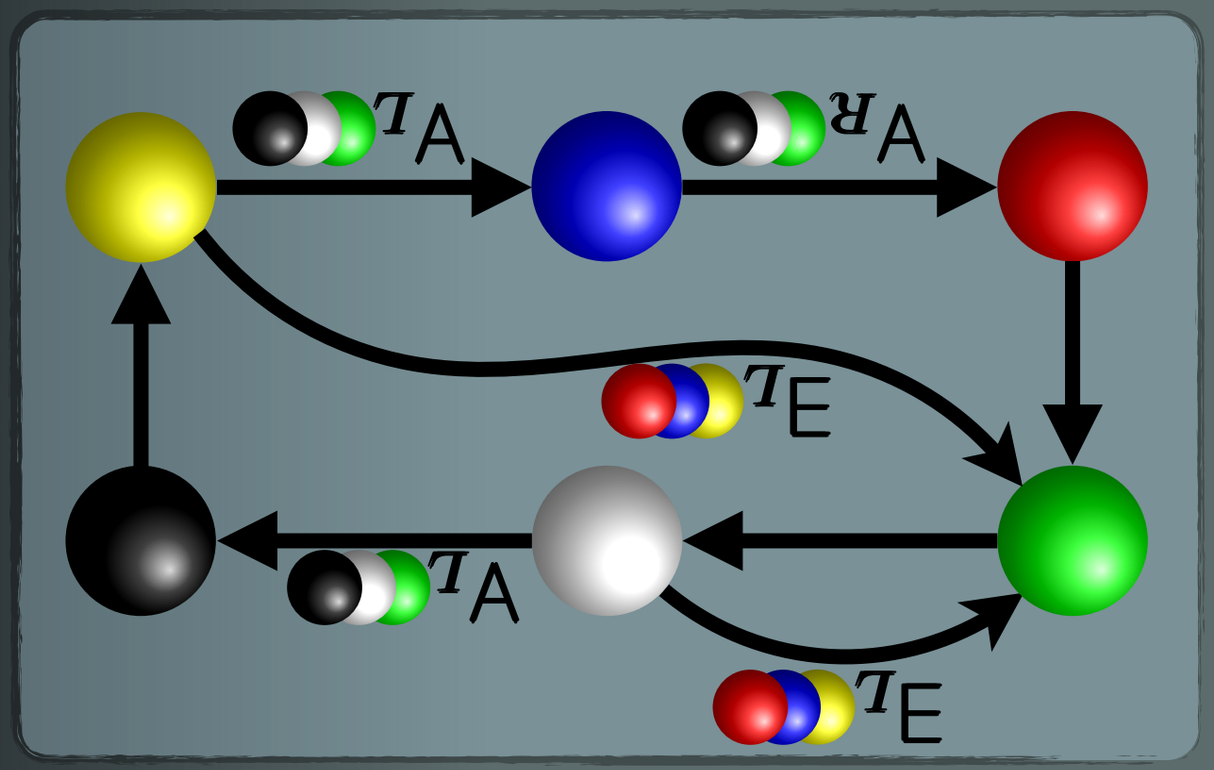


active process

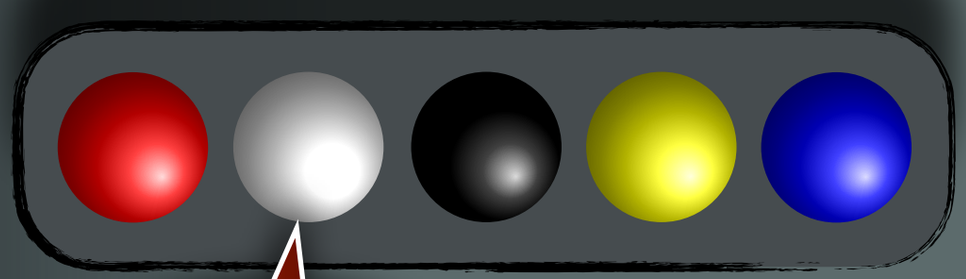
local transition



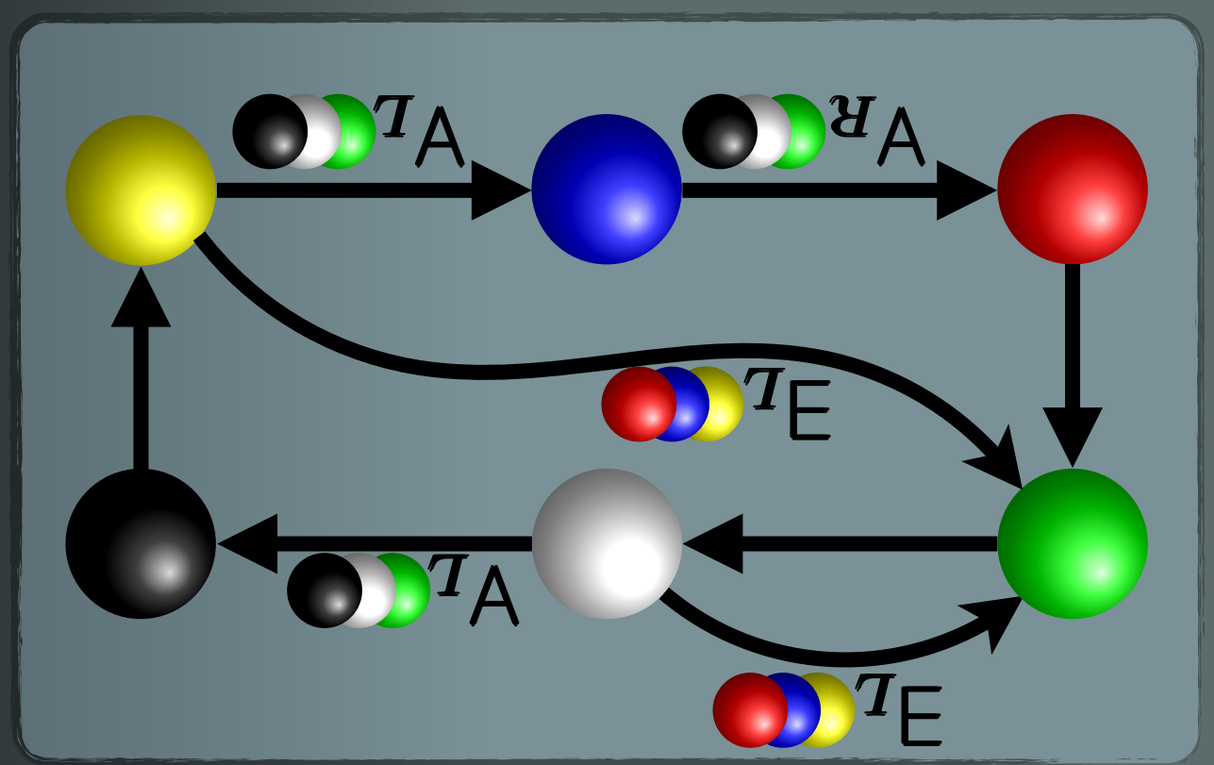




post



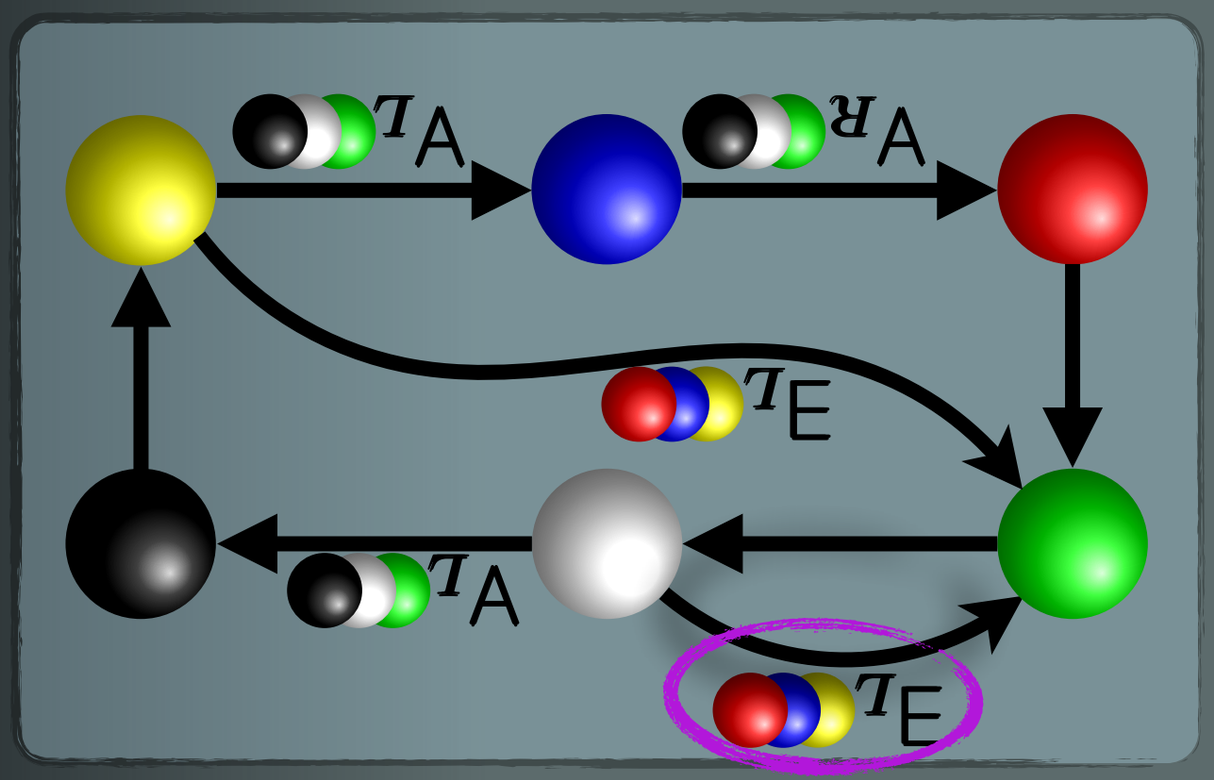
active process



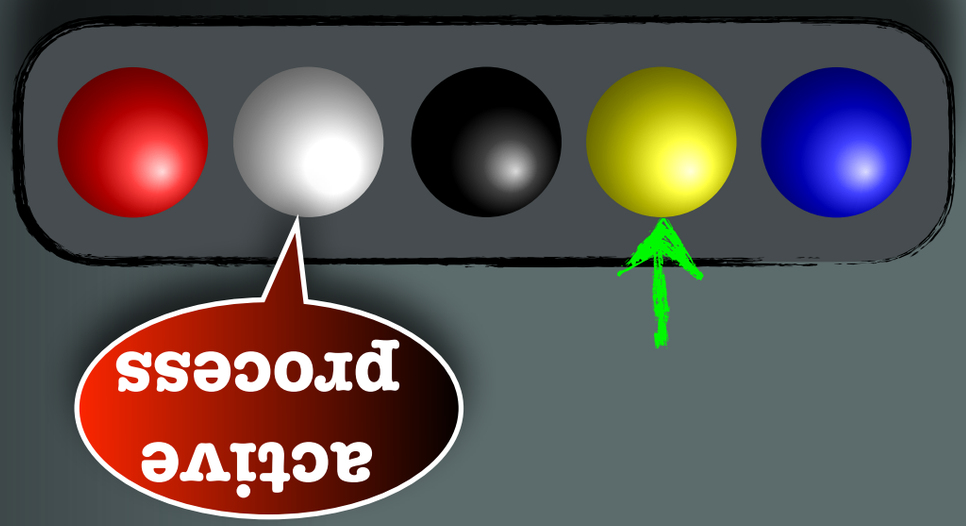
post

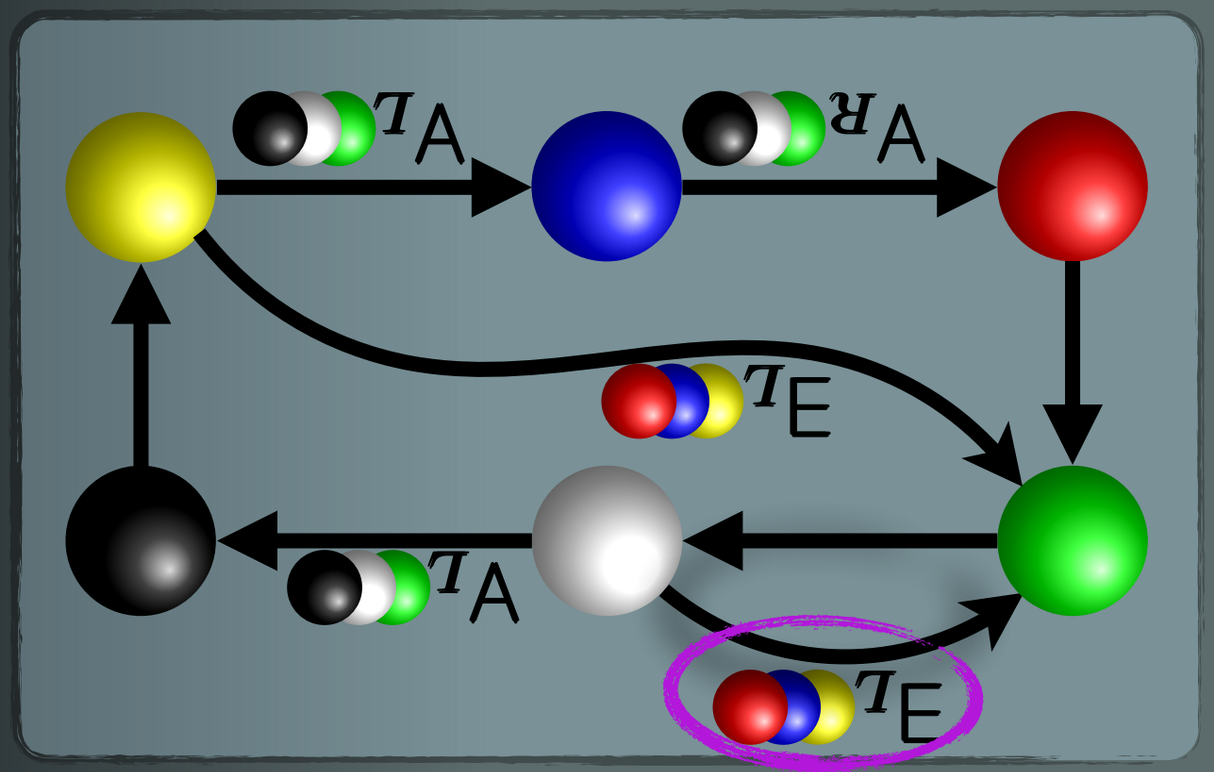


active process



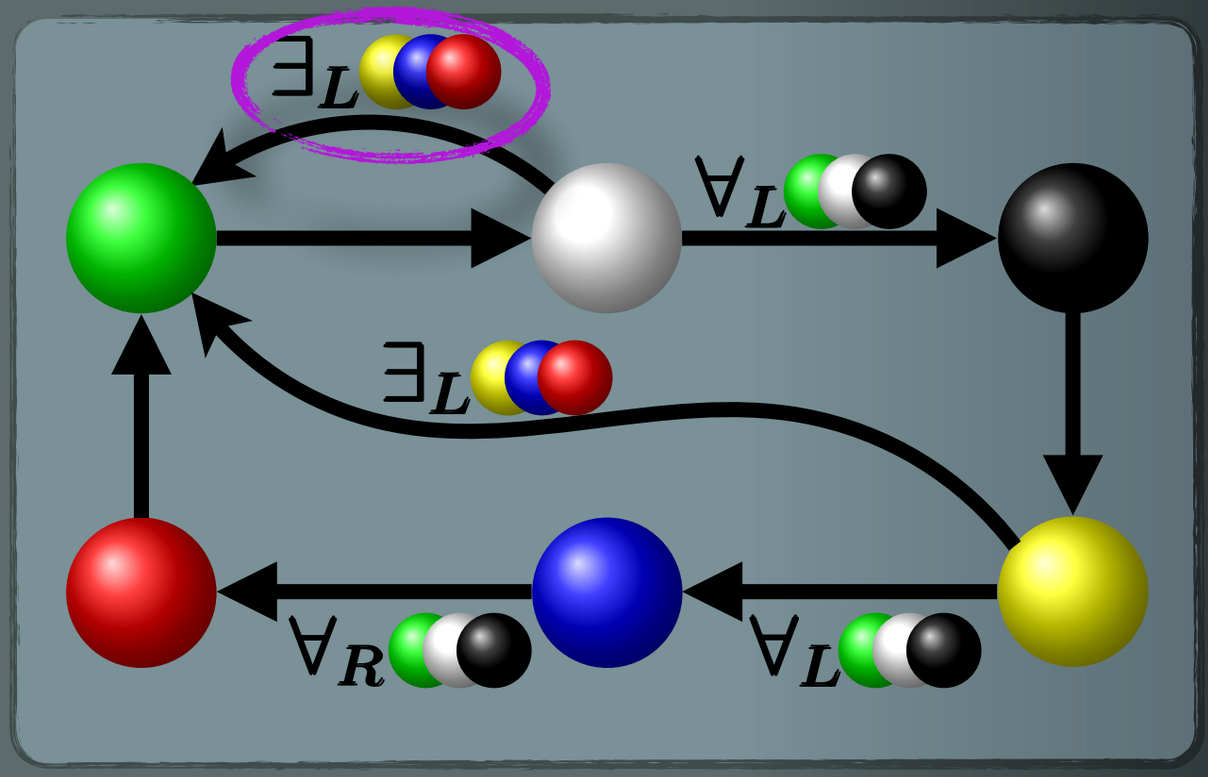
post



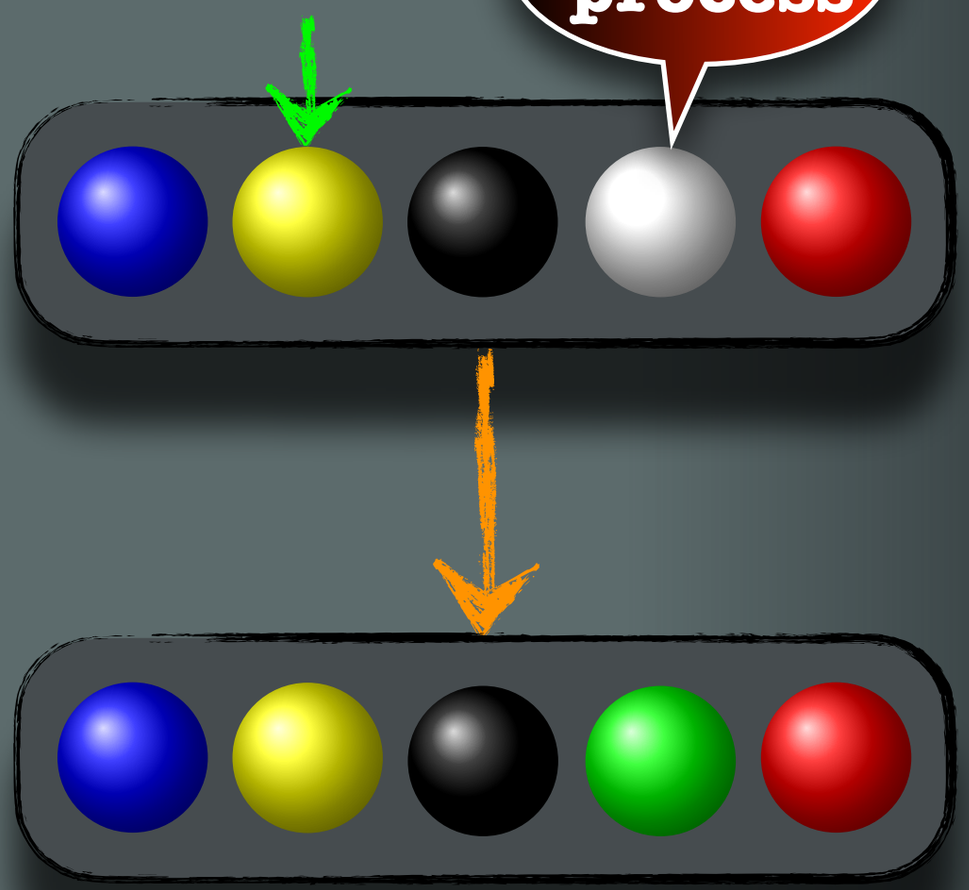


post

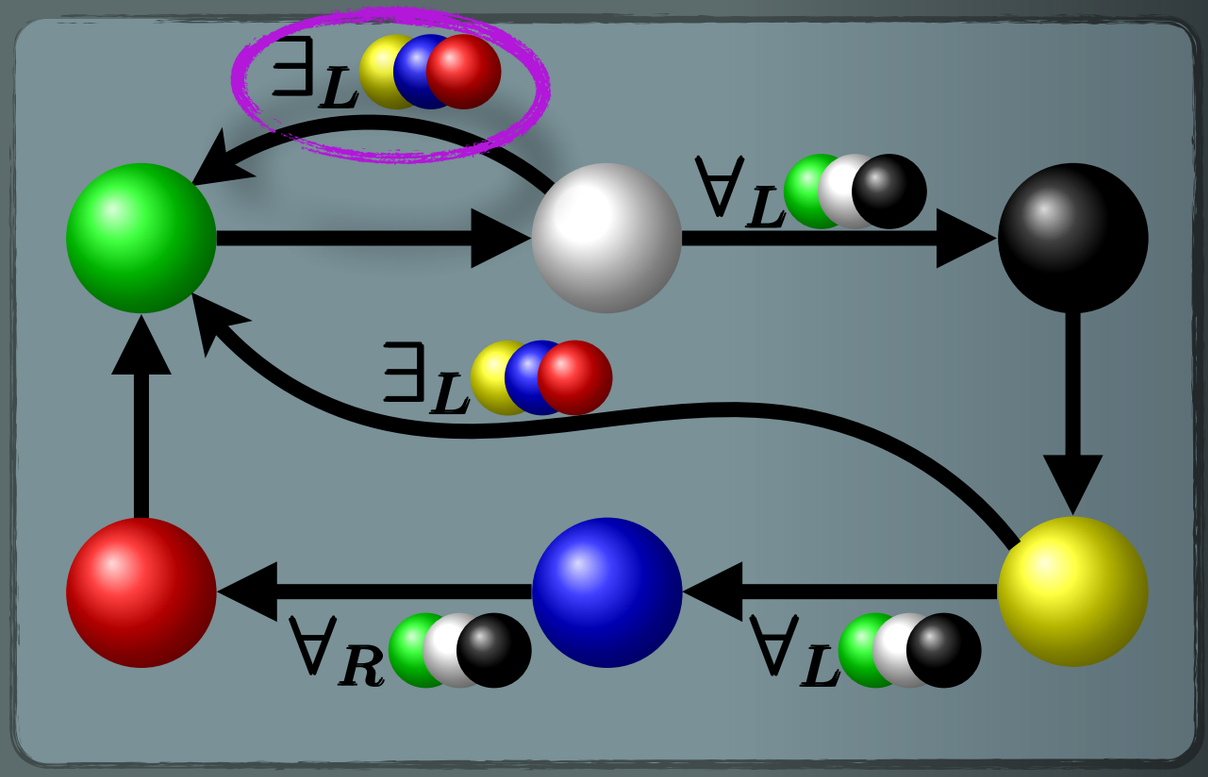
post



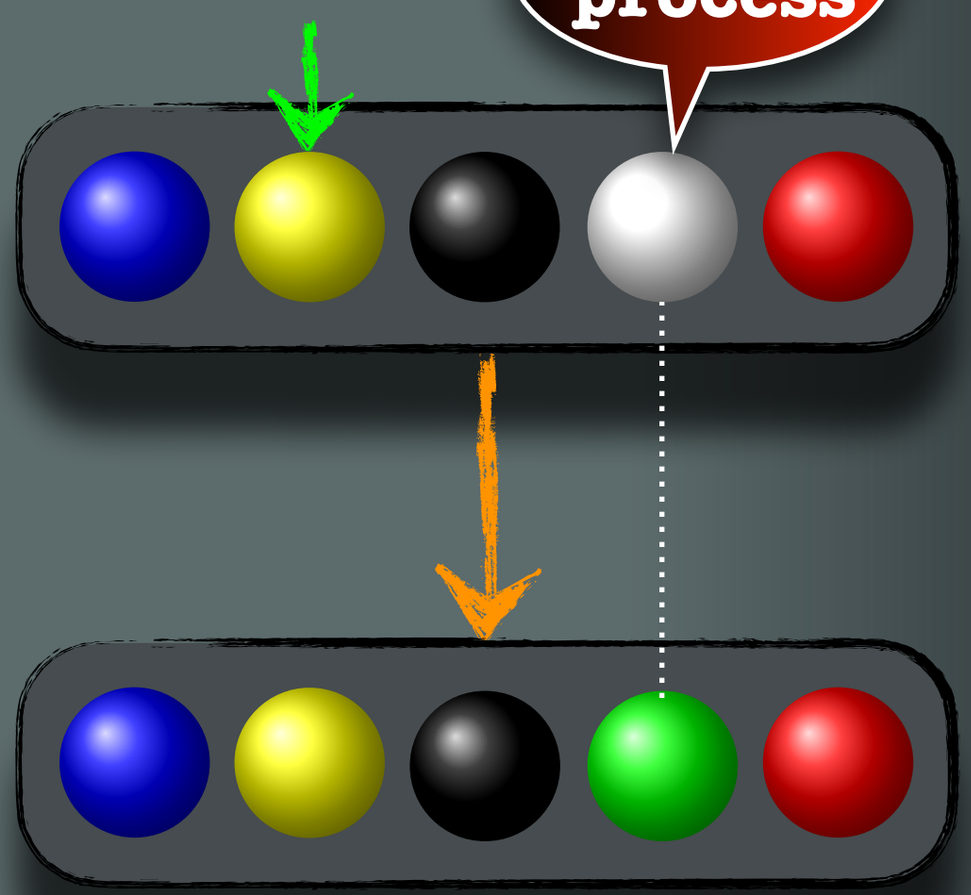
active process



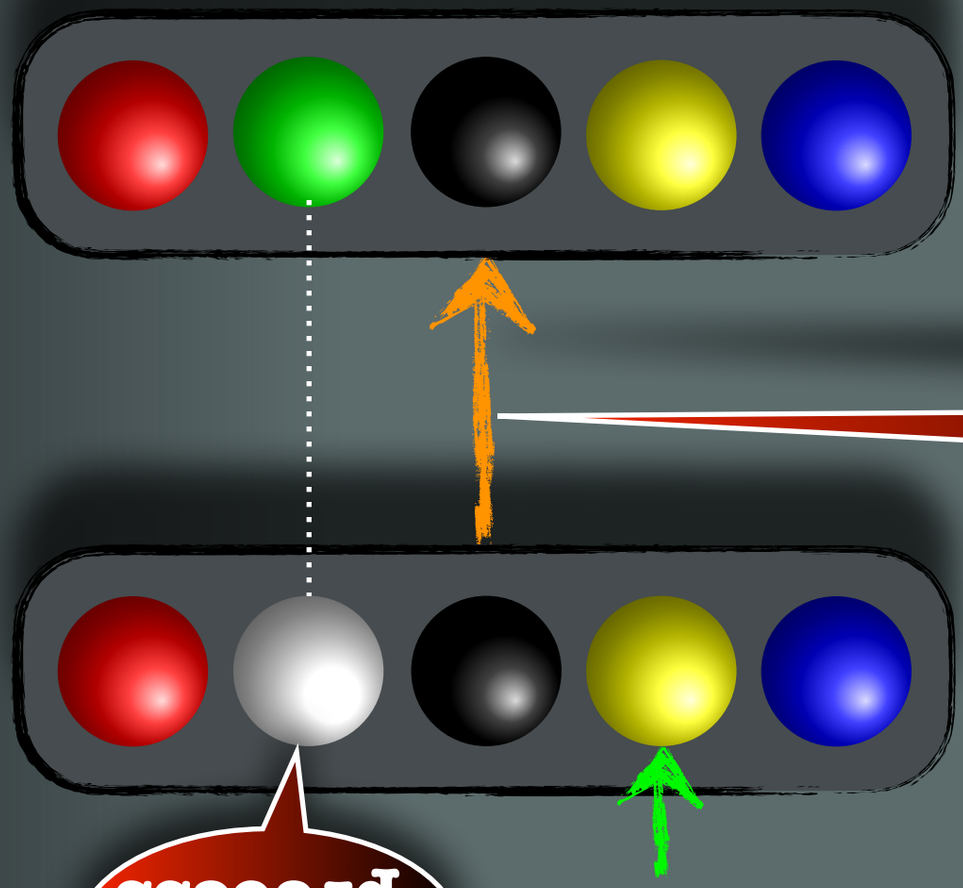
post



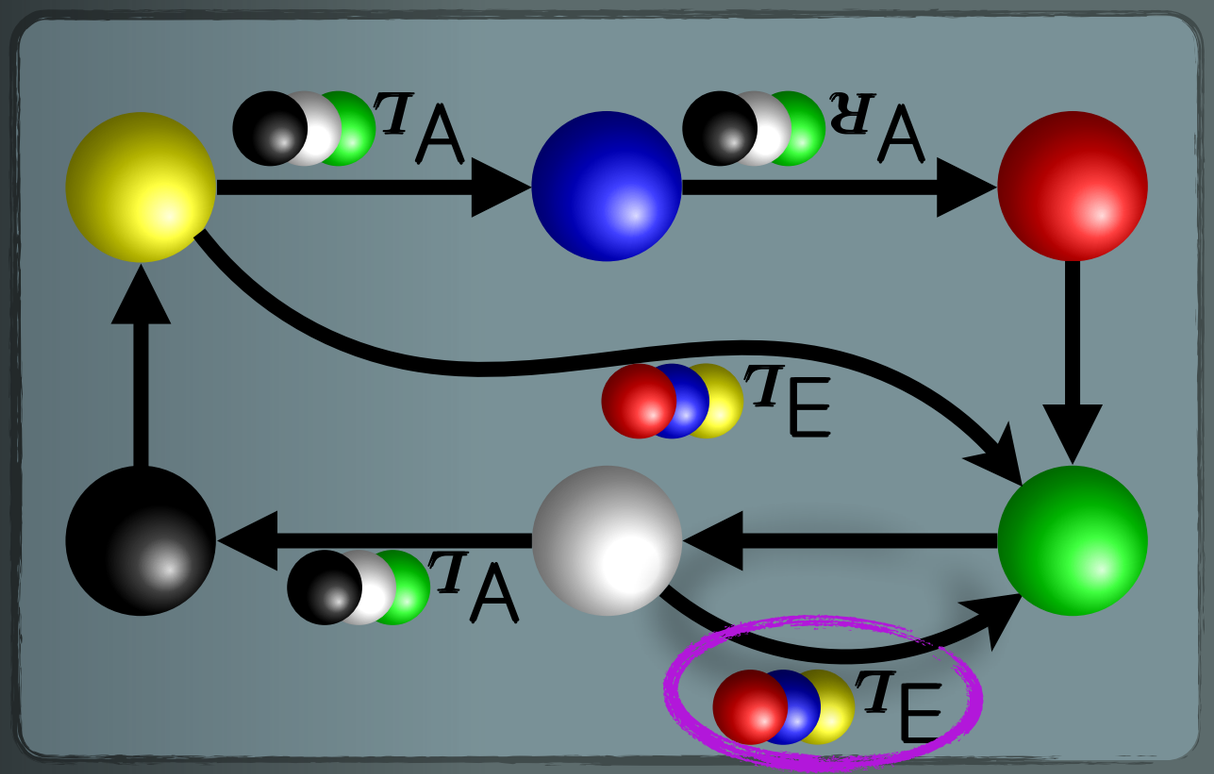
active process



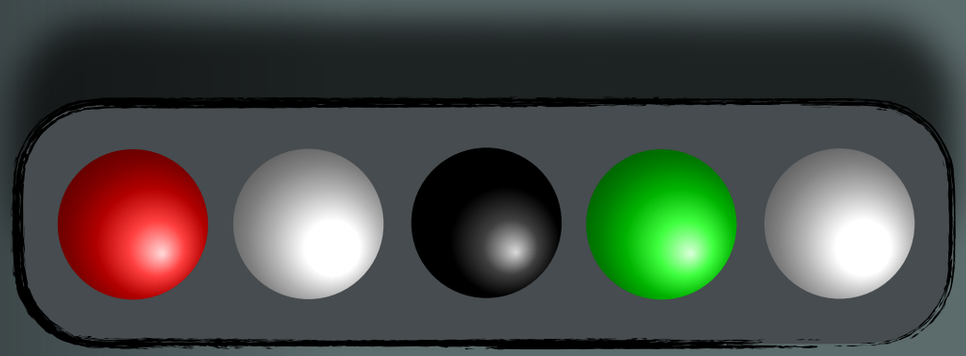
existential
global
transition

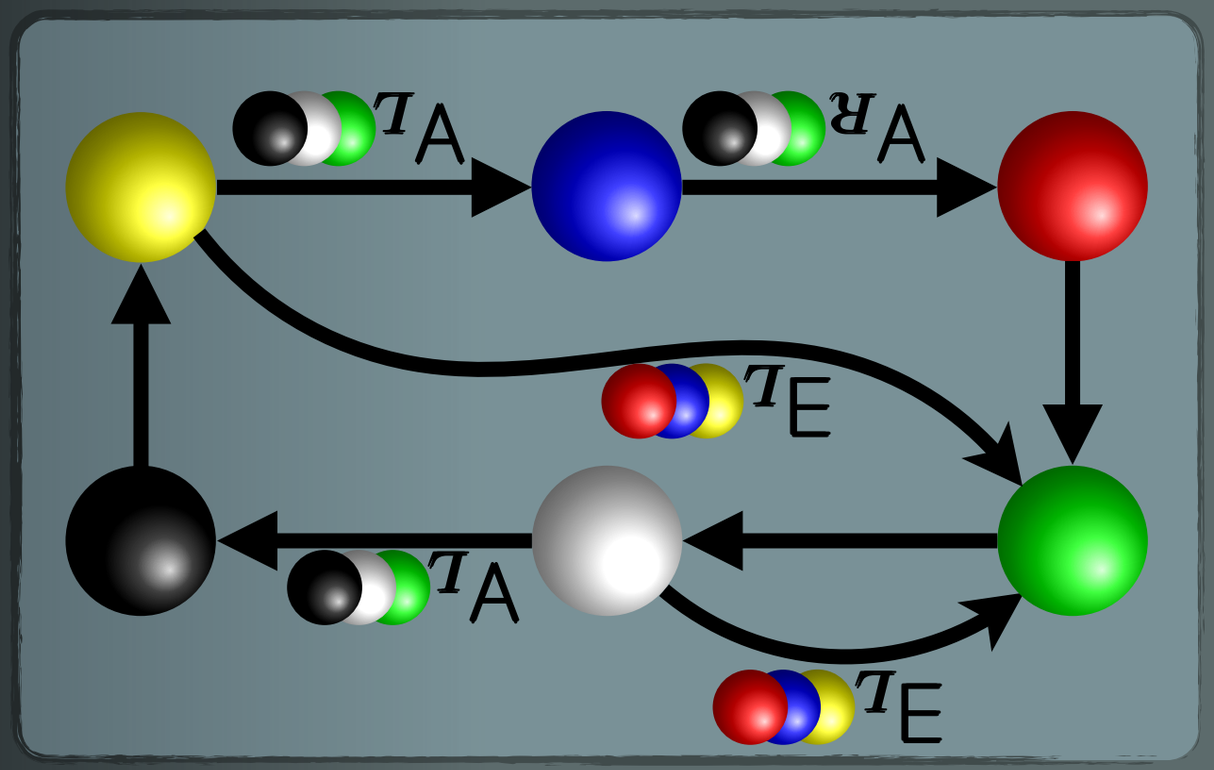


active
process



post

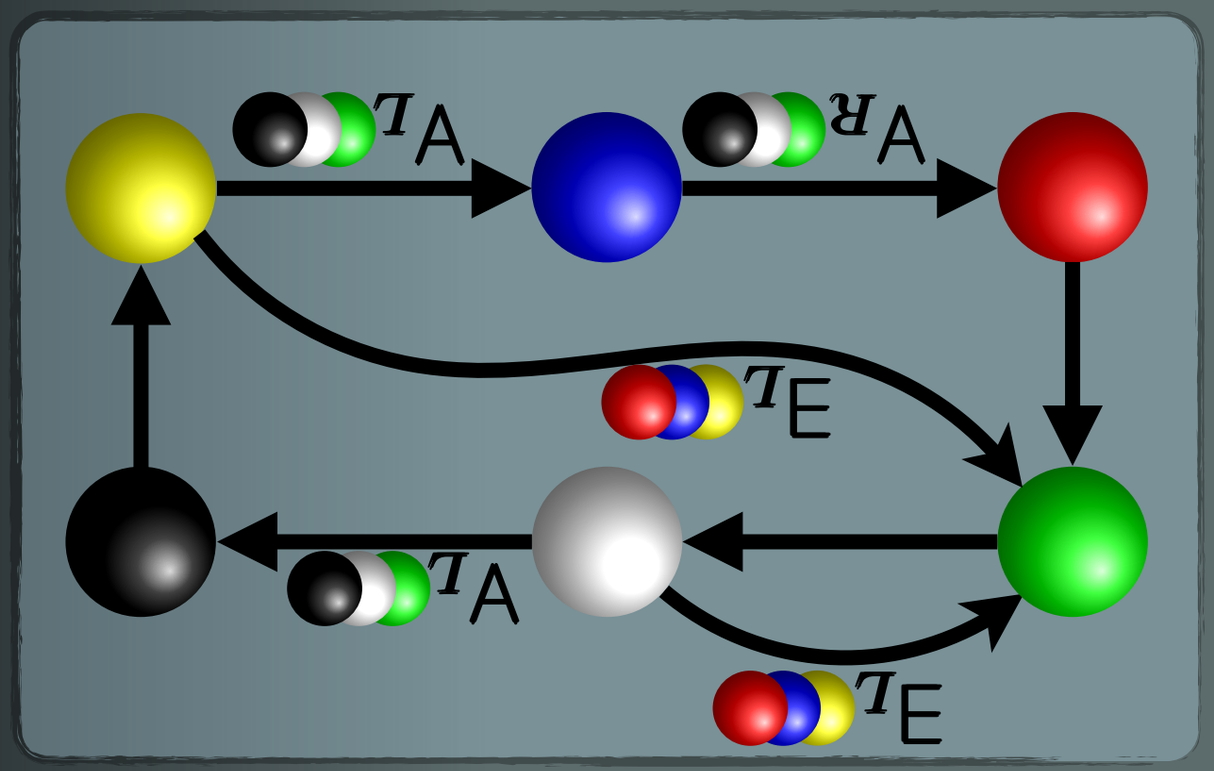




post



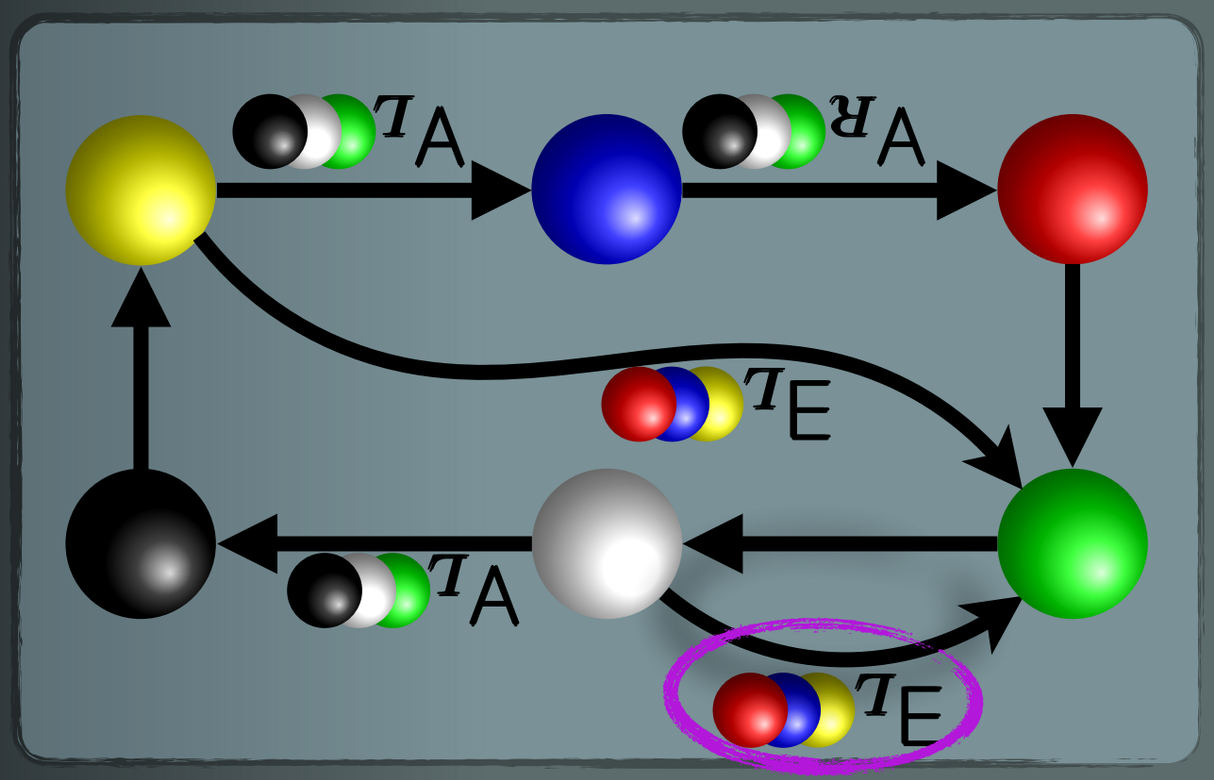
active process



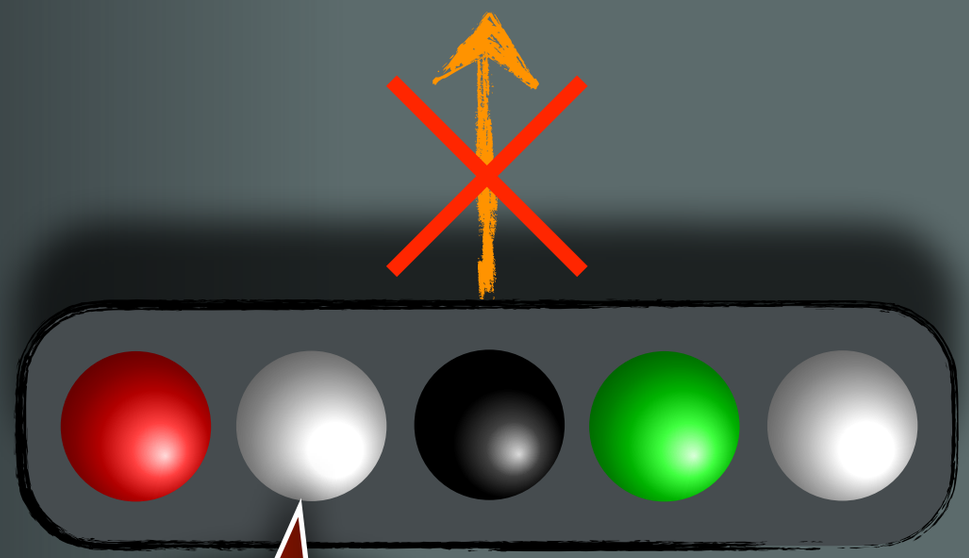
post



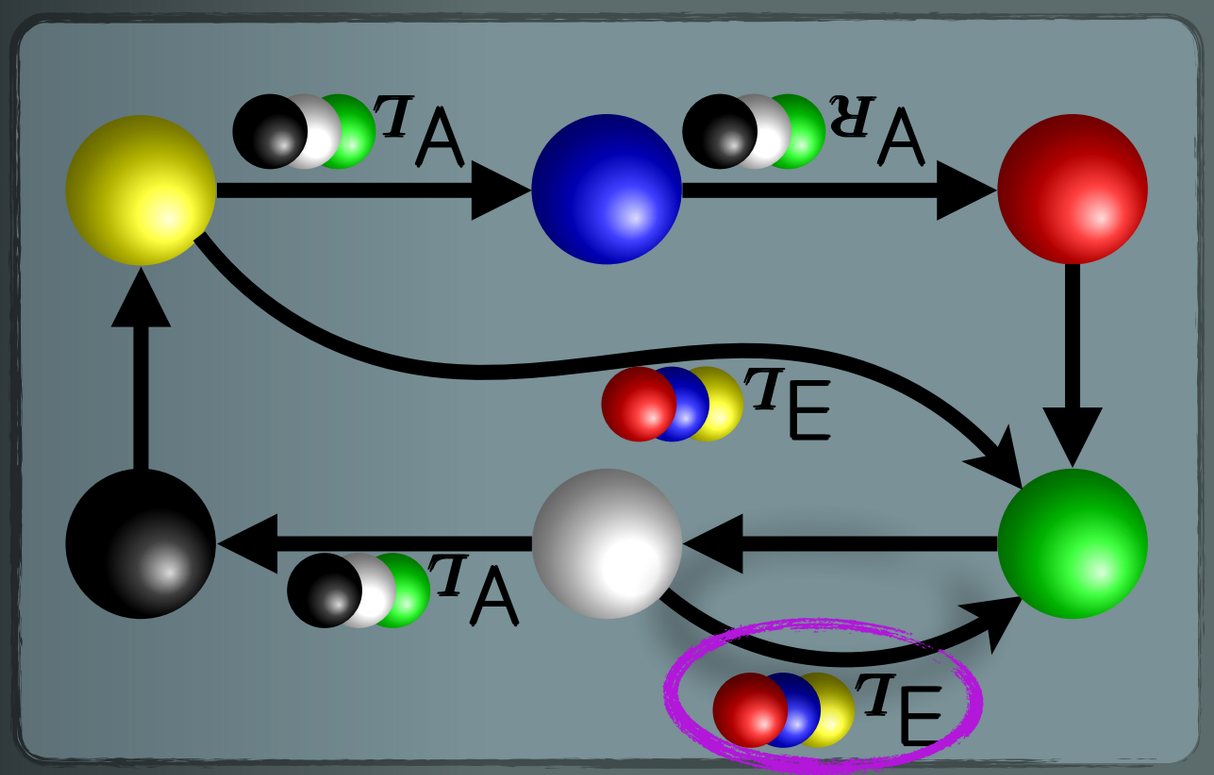
active process



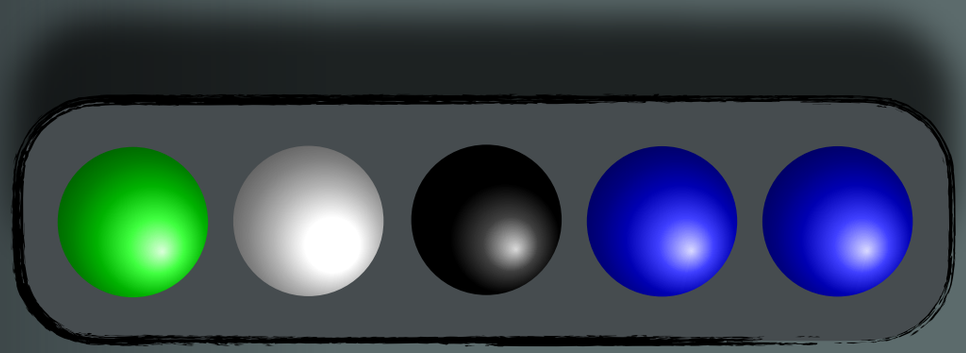
post

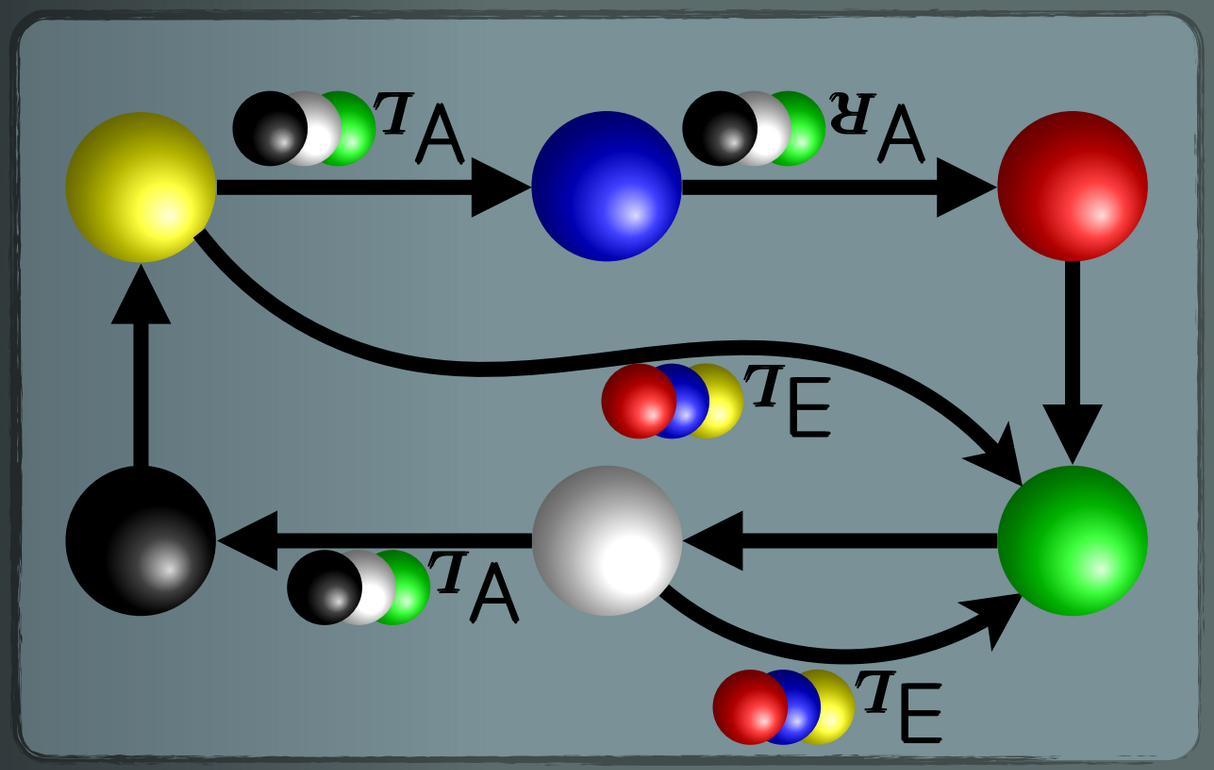


active process

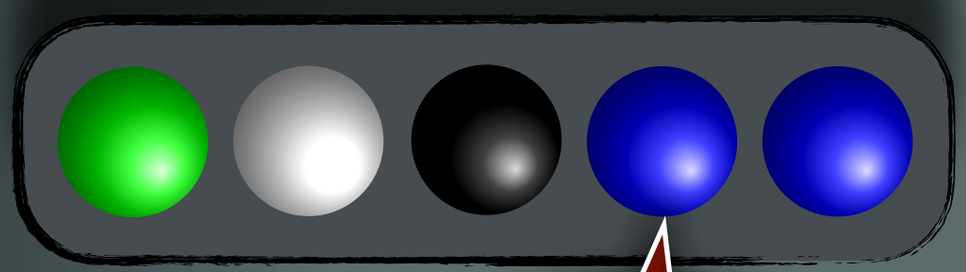


post

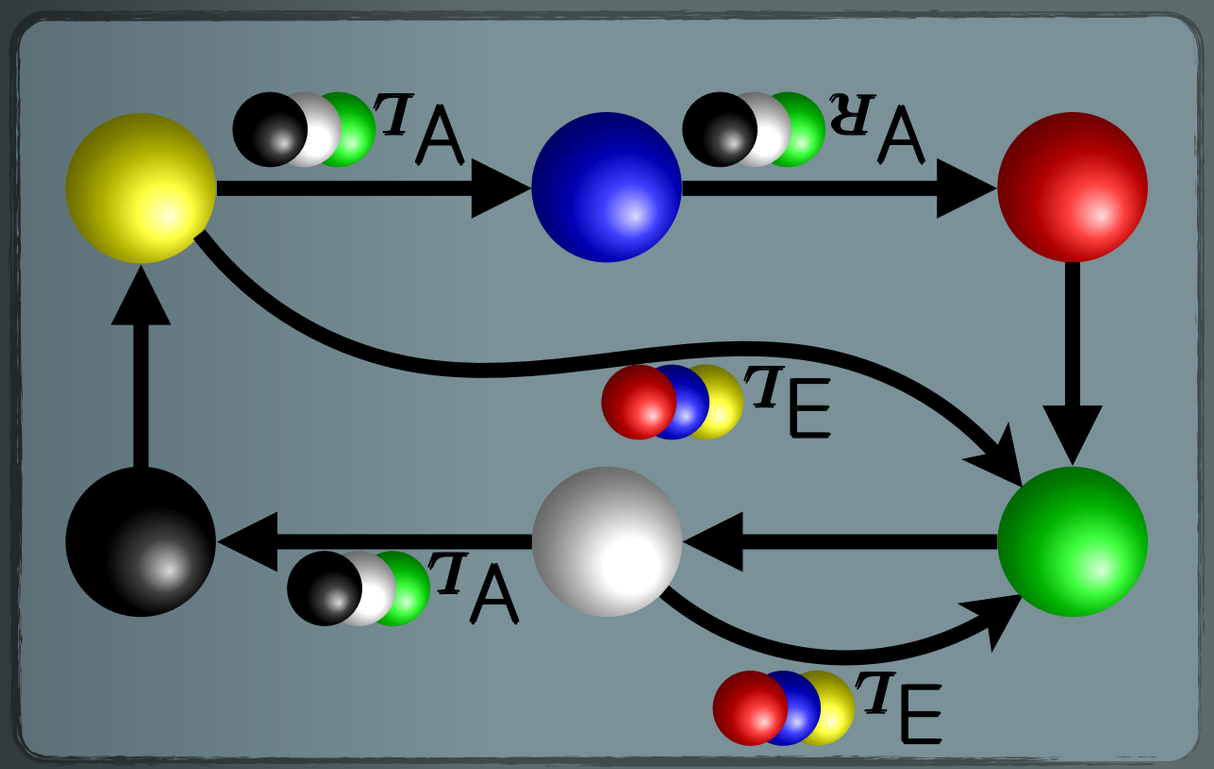




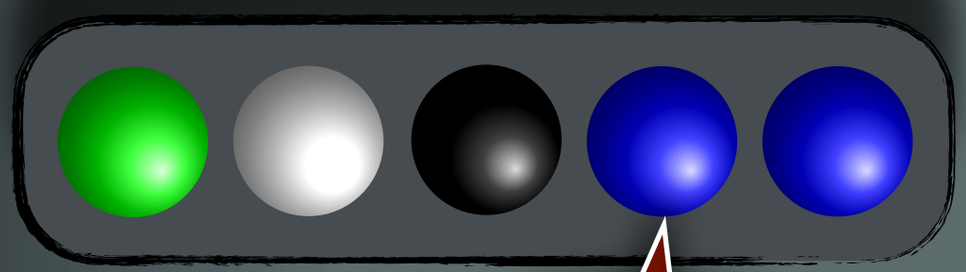
post



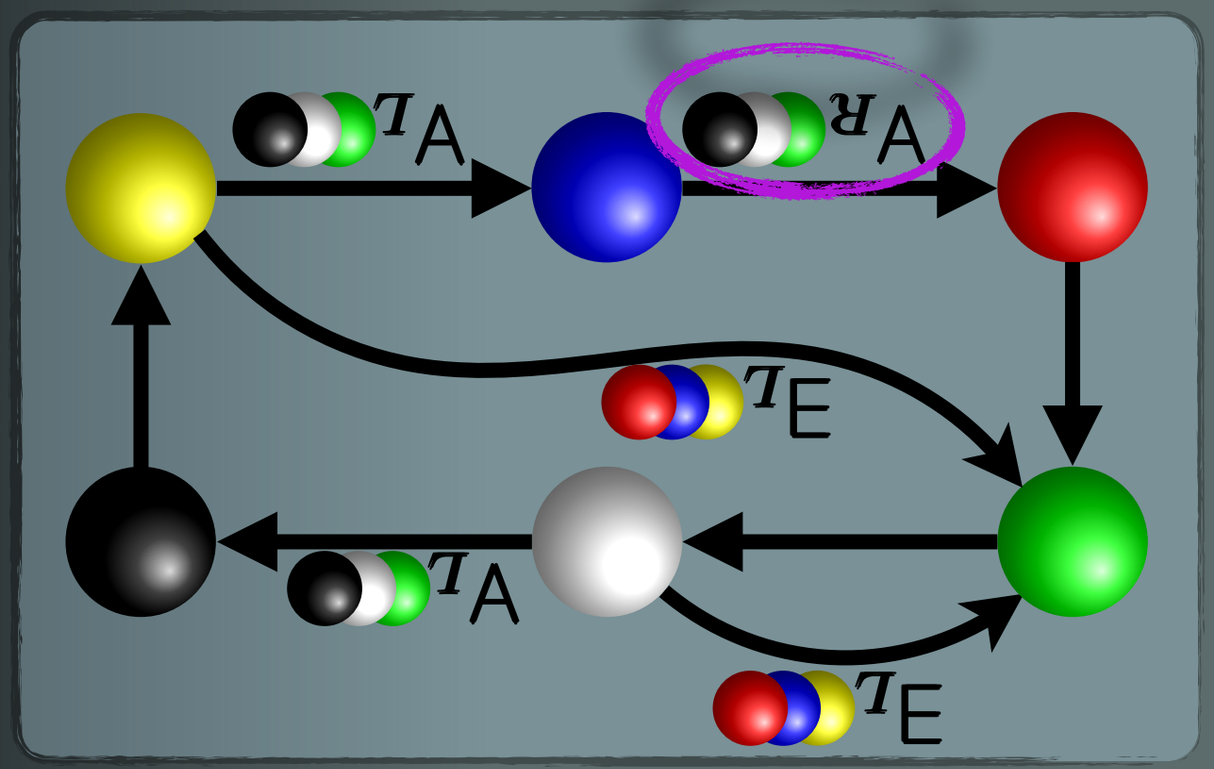
active process



post

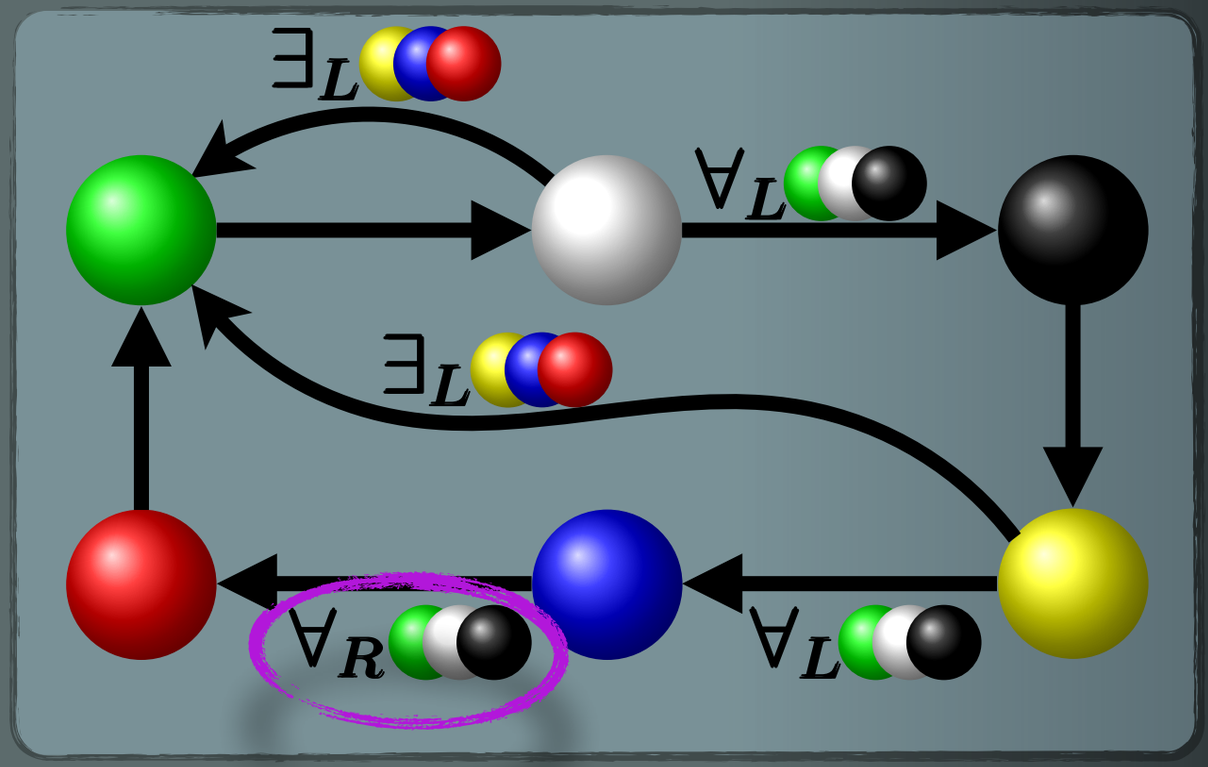


active process

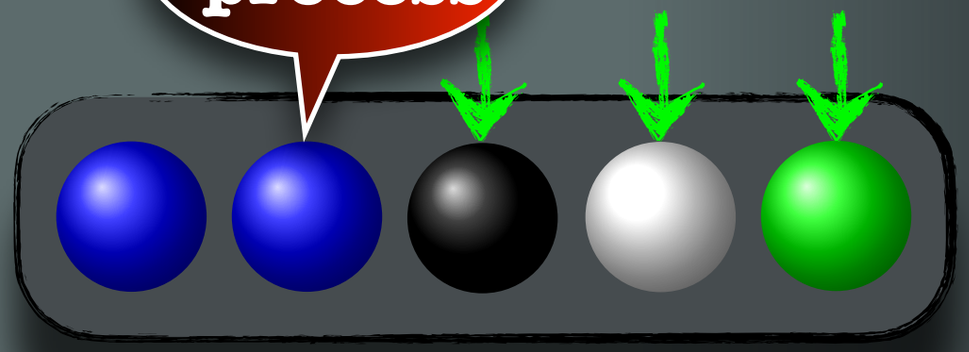


post

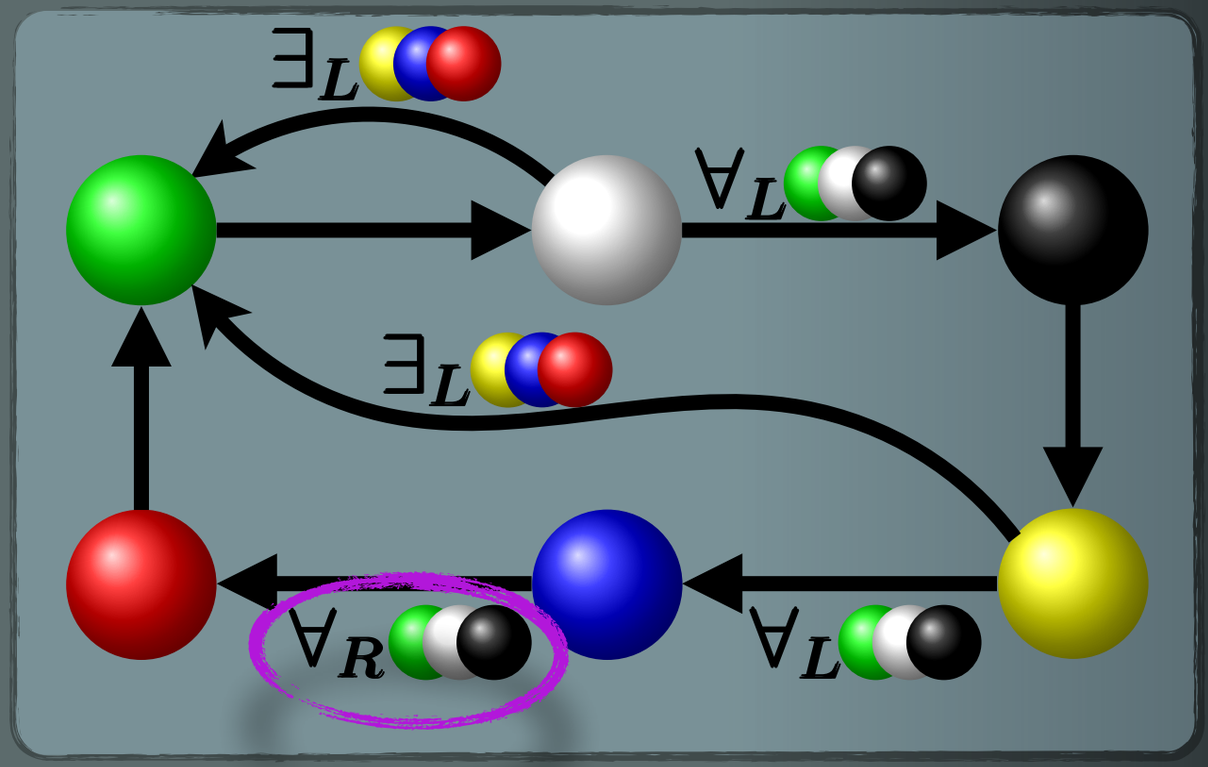
post



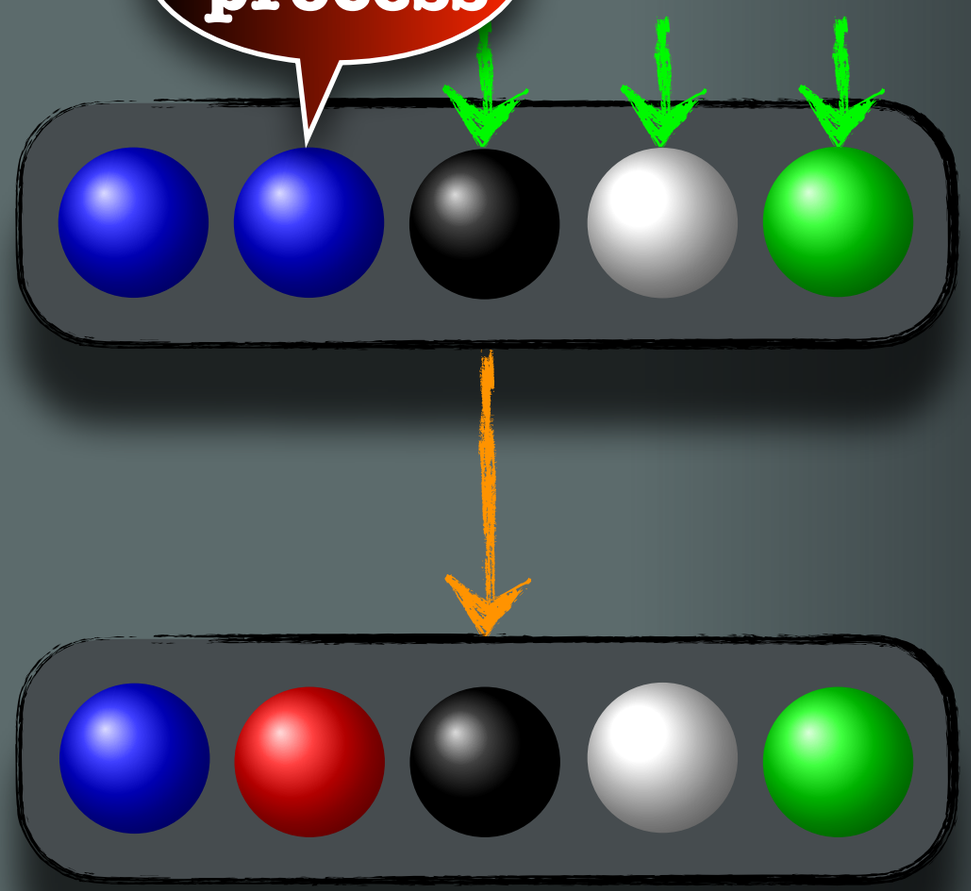
active process



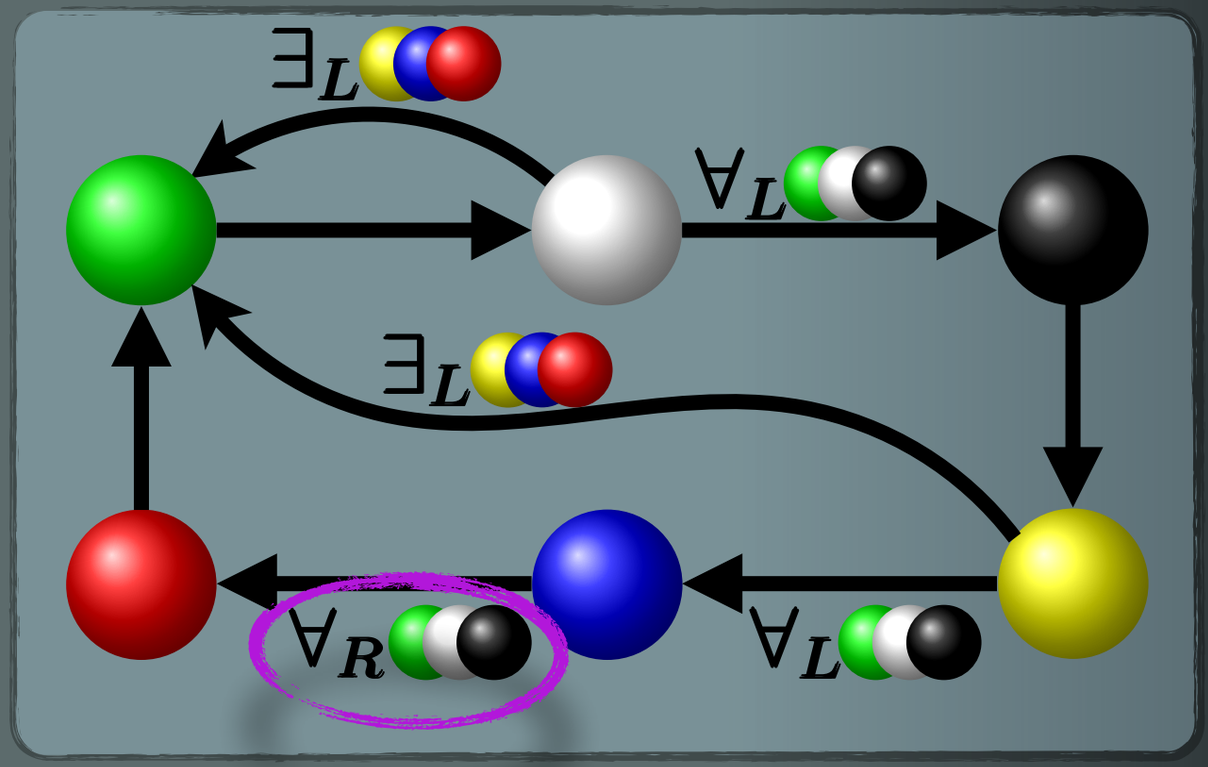
post



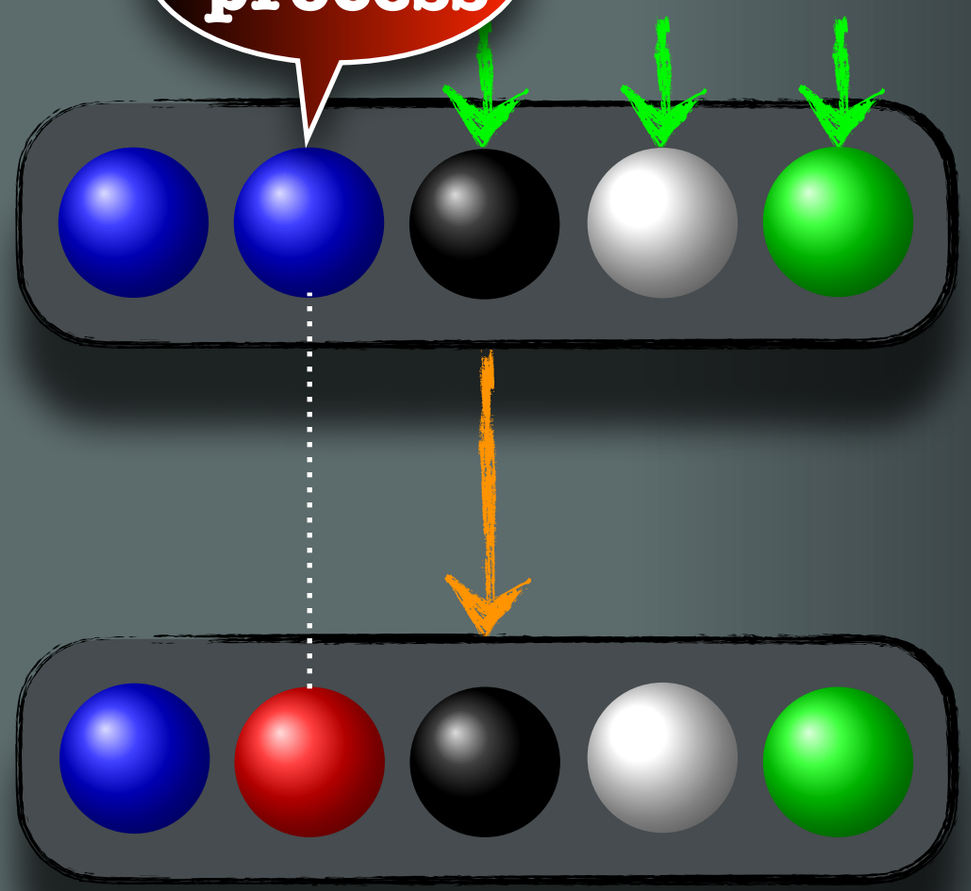
active process



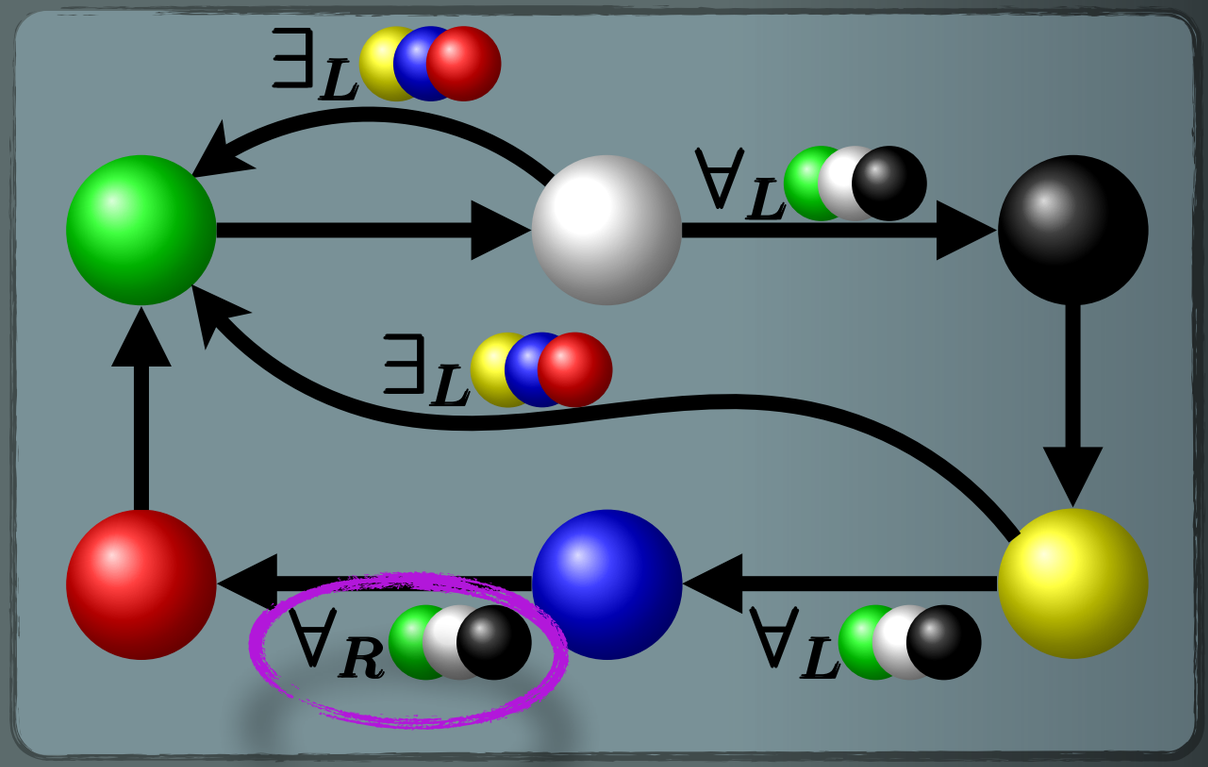
post



active process

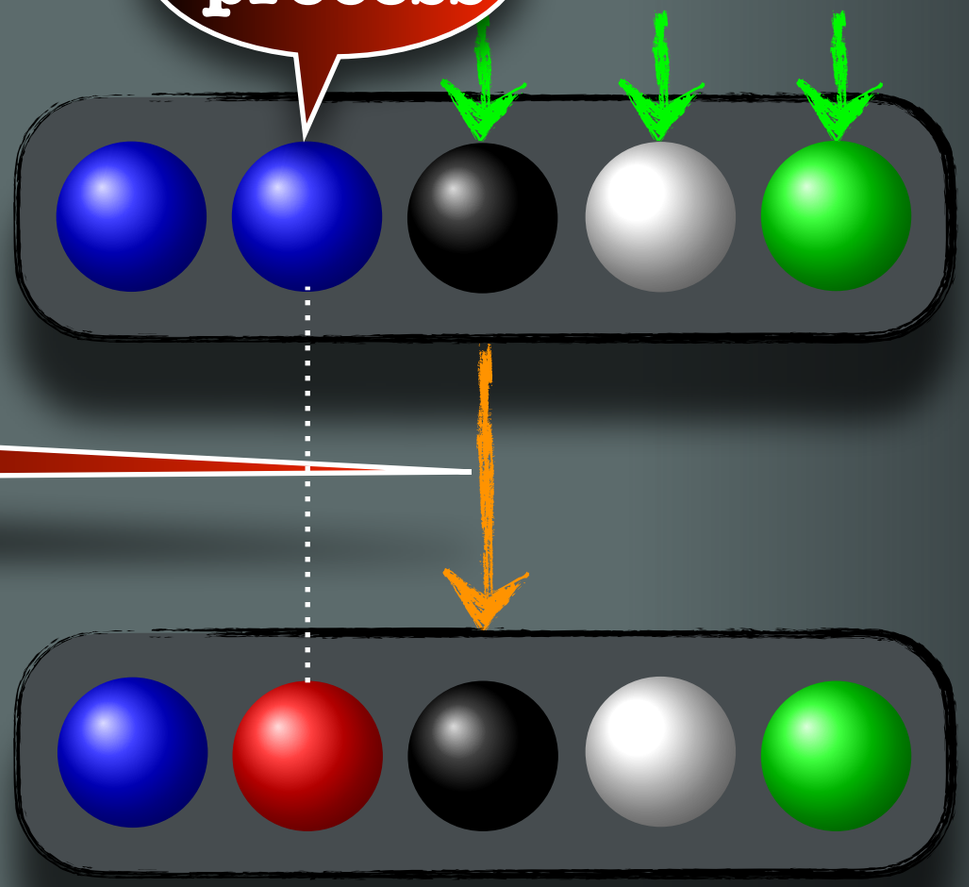


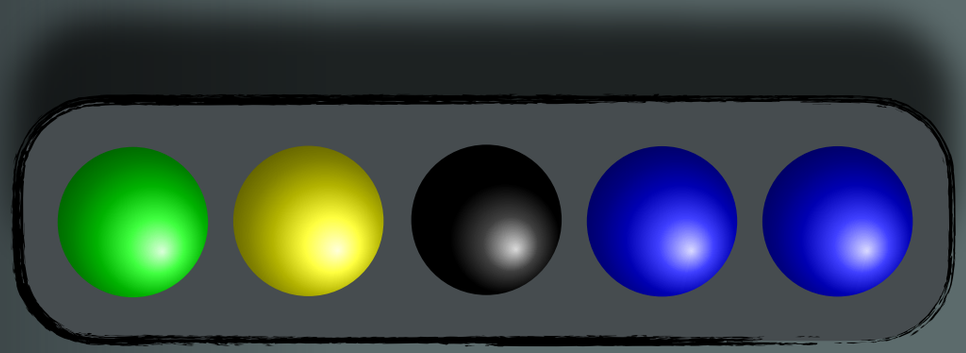
post

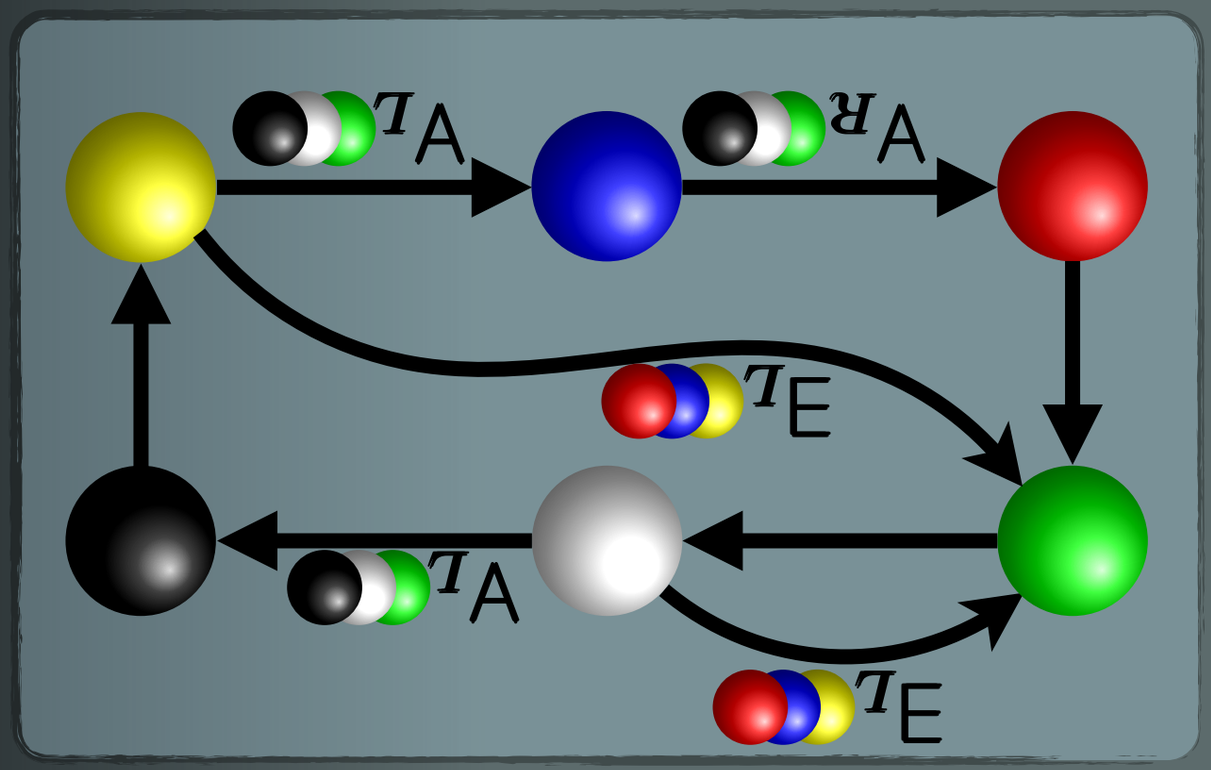


active process

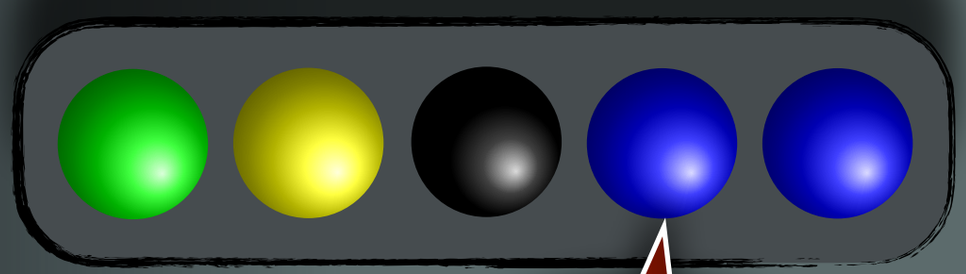
universal global transition



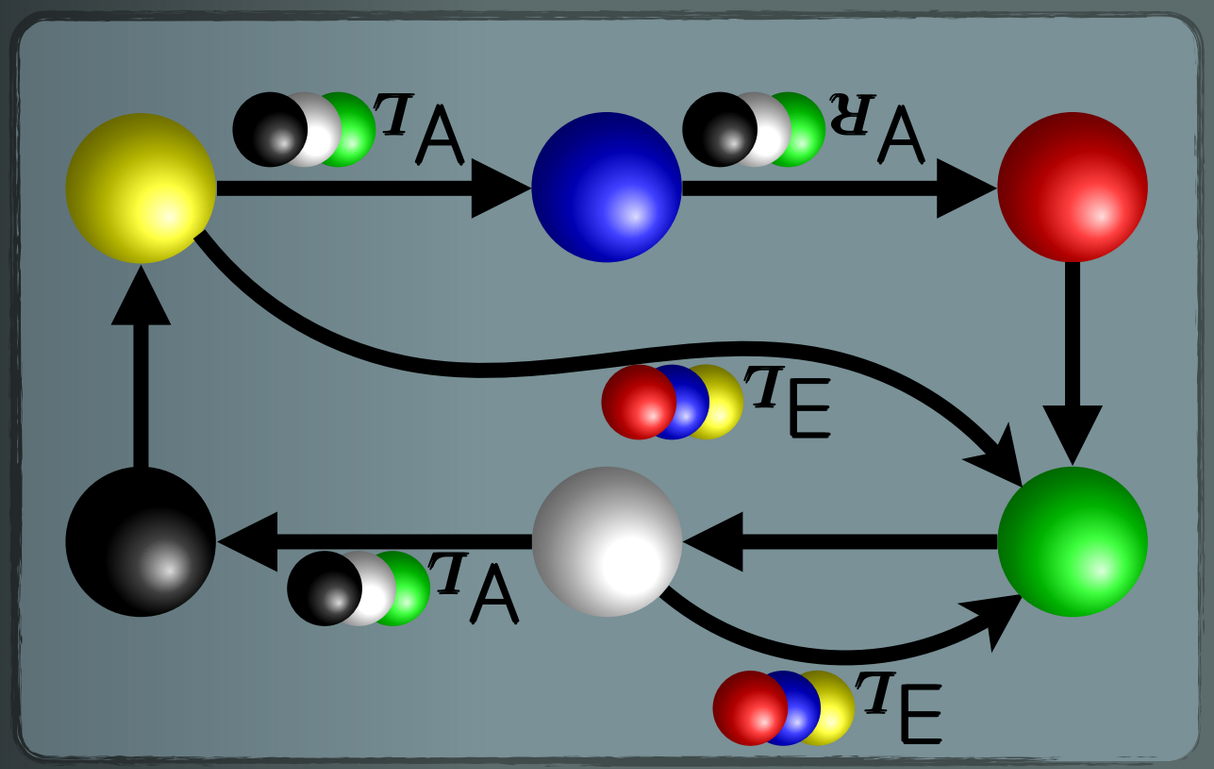




post

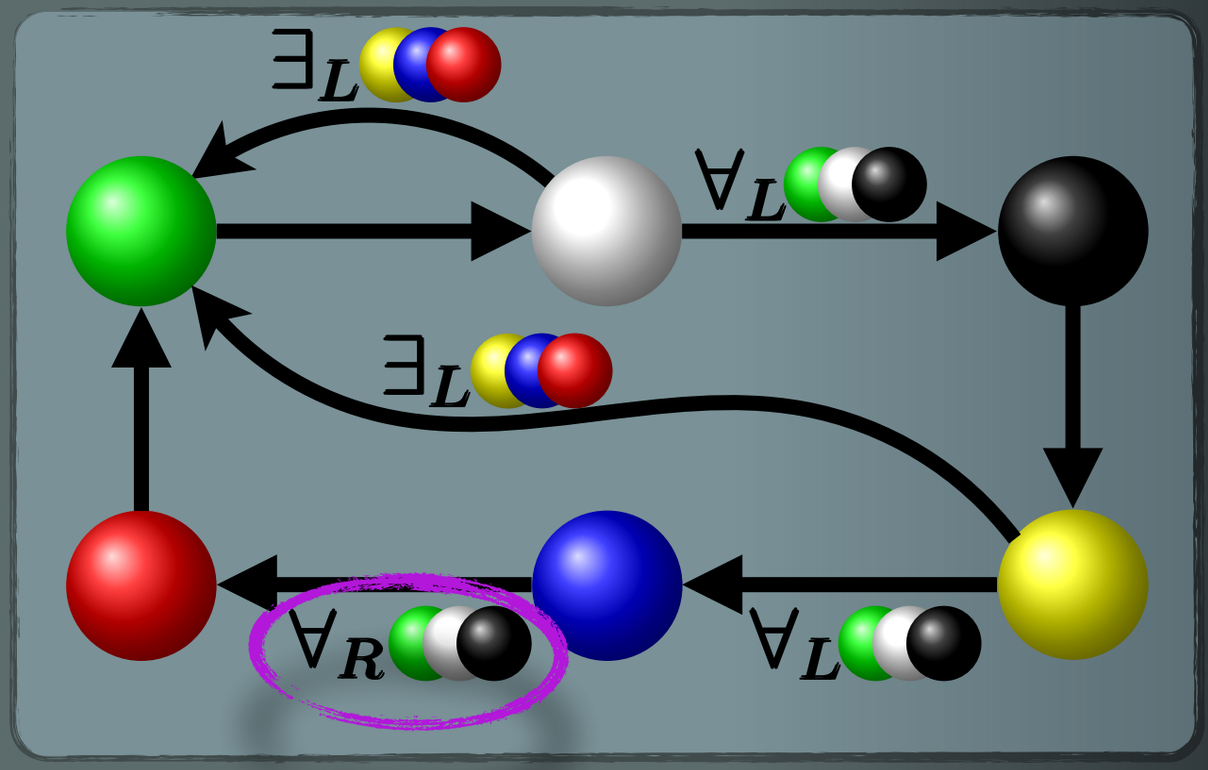


active process

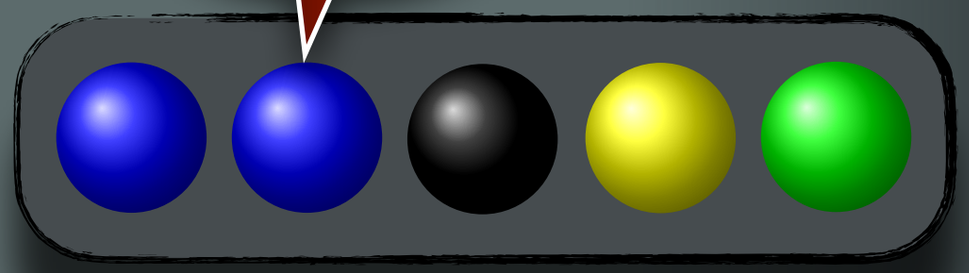


post

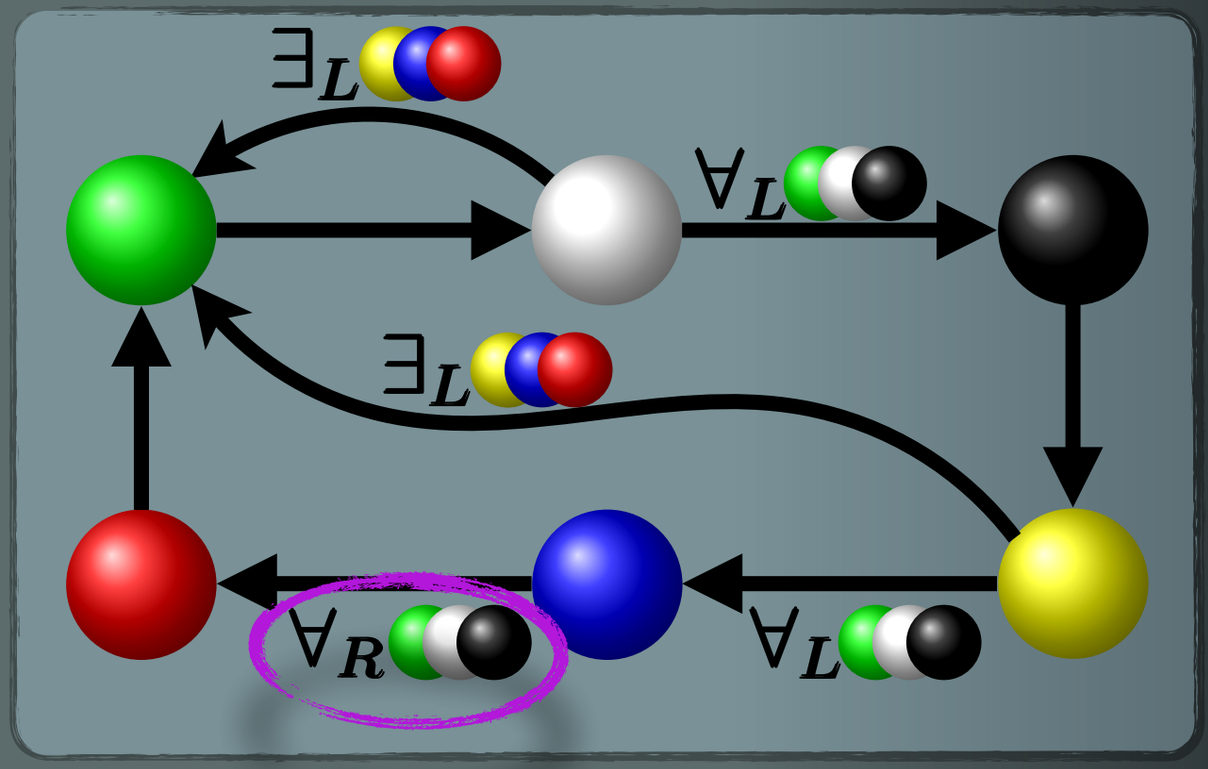
post



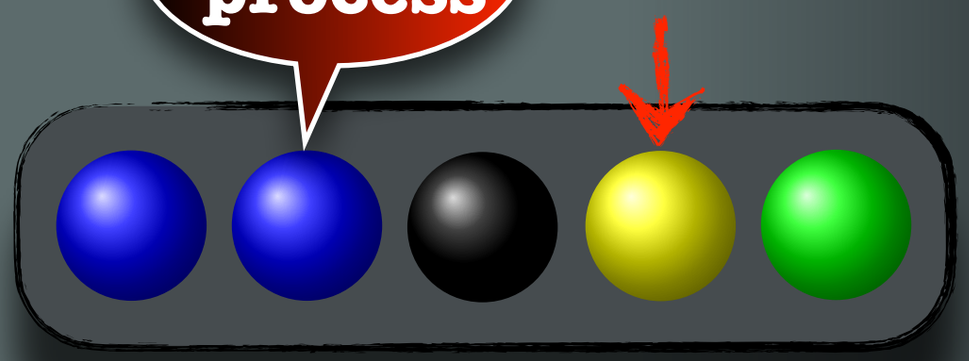
active process



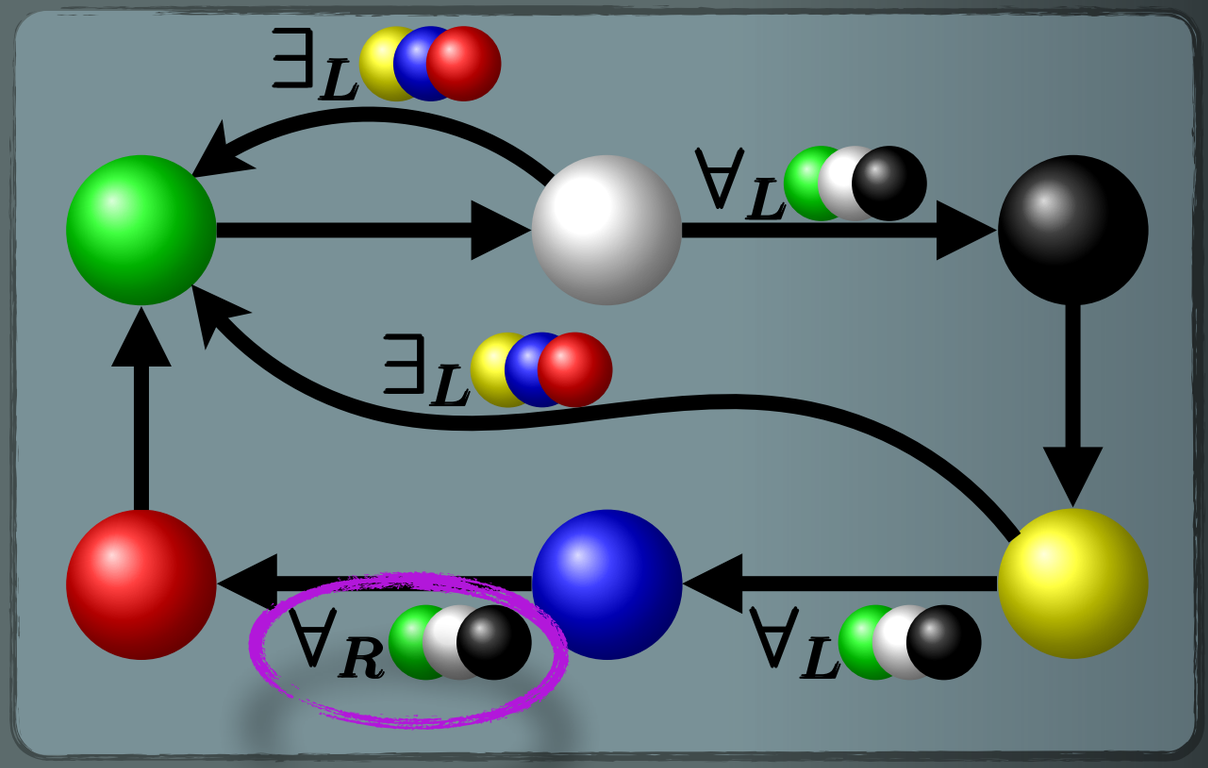
post



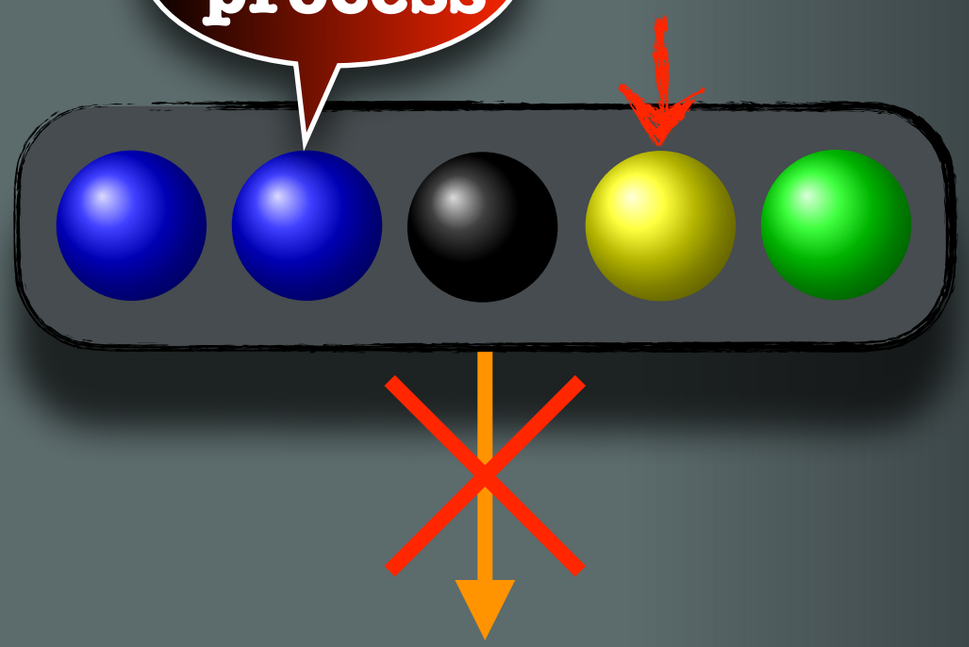
active process

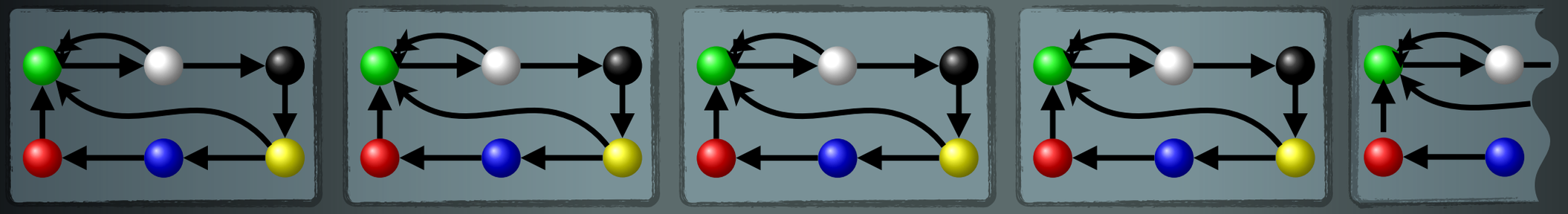


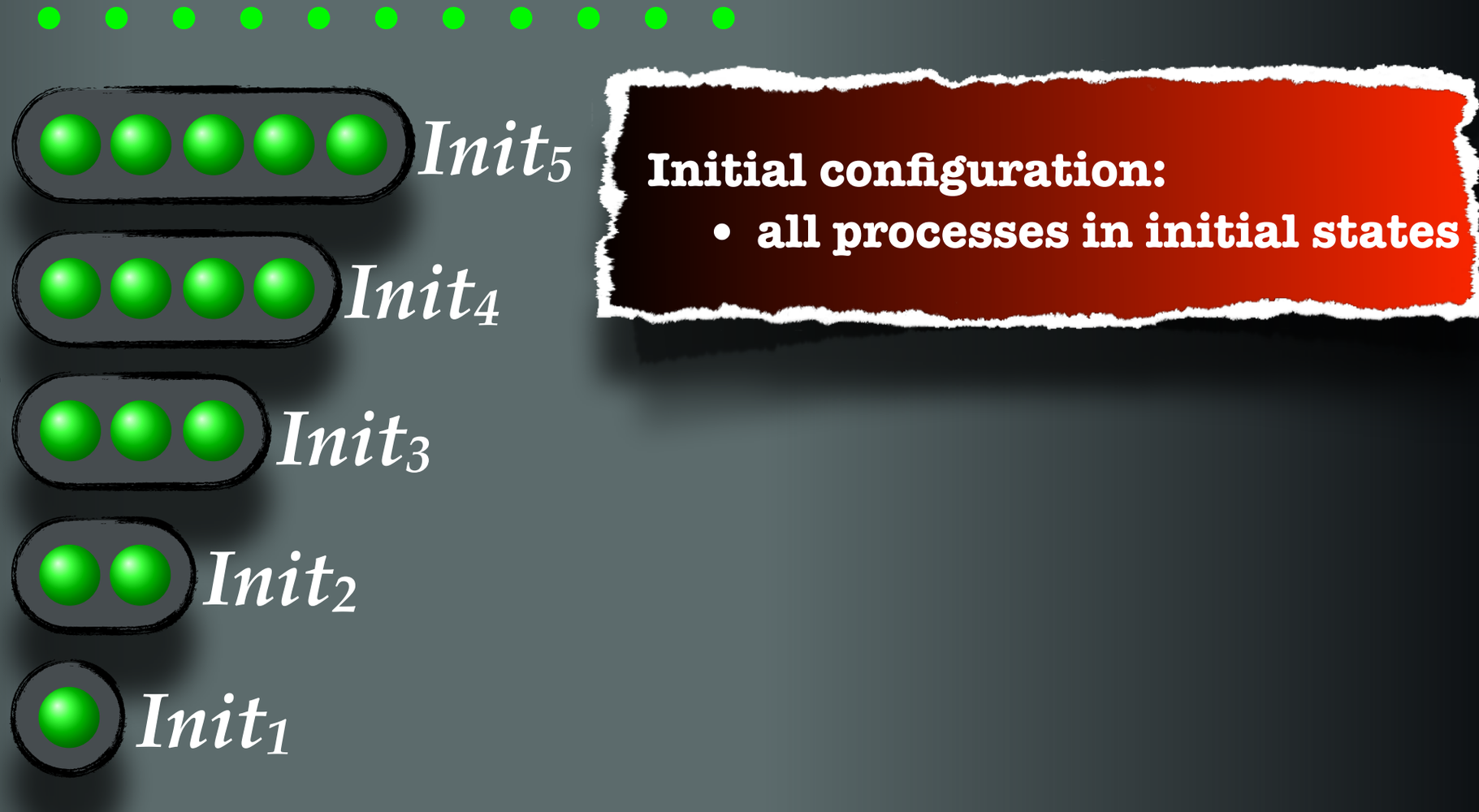
post

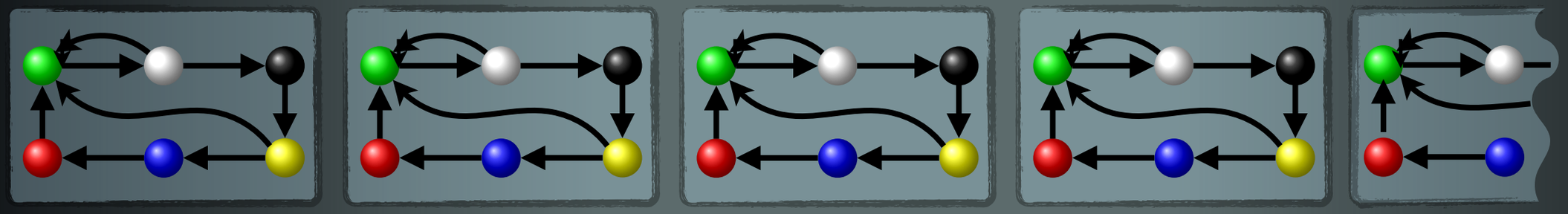


active process

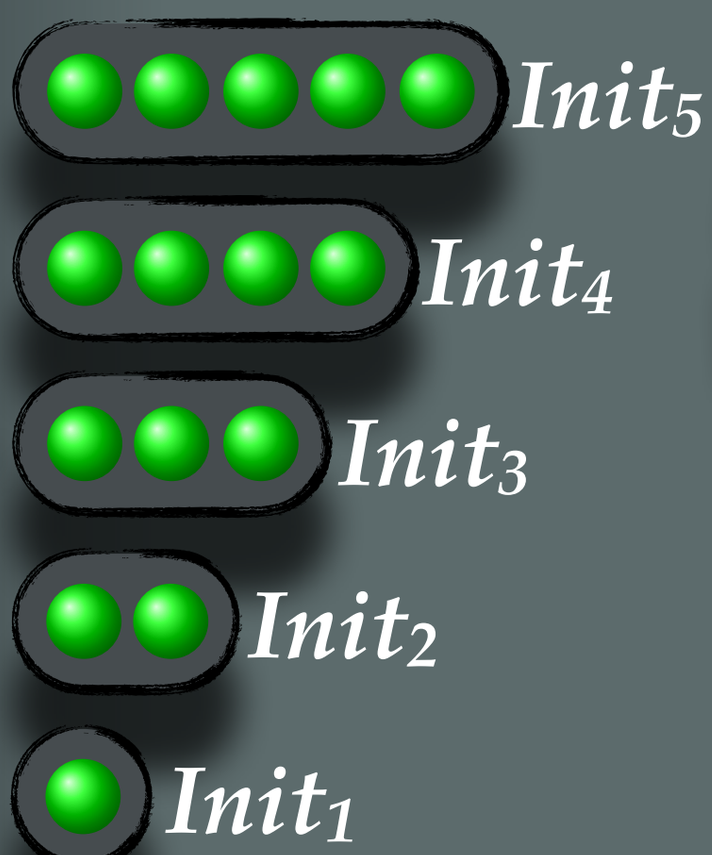





$Init_5$
$Init_4$
$Init_3$
$Init_2$
$Init_1$
Initial Configurations
Initial configuration:
• all processes in initial states



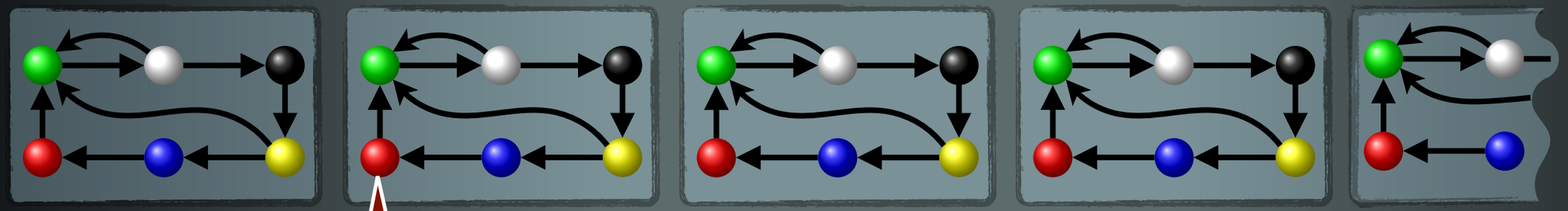




Initial Configurations

Initial configuration:
- **all processes in initial states**

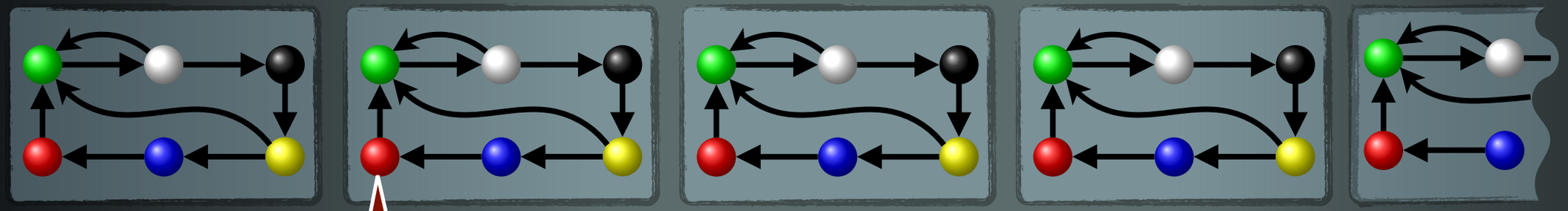
Set of initial configurations:
- **infinite, but**
- **regular**



critical section

Bad configuration:

- **2 or more processes in critical section**

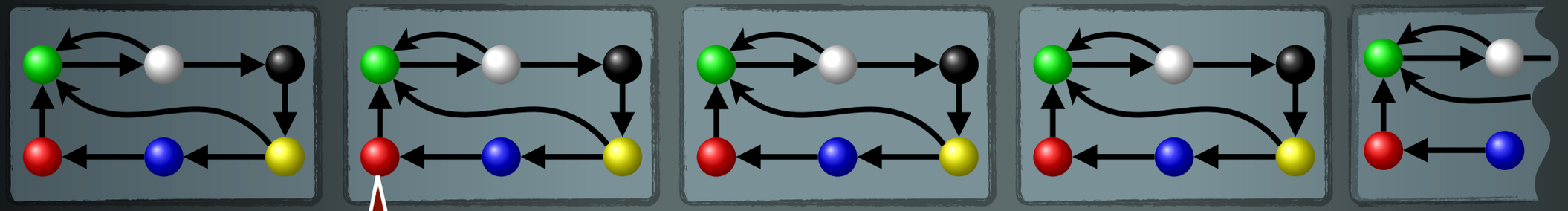


critical section

Bad configuration:

- **2 or more processes in critical section**

Bad Configurations

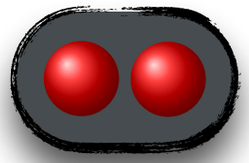


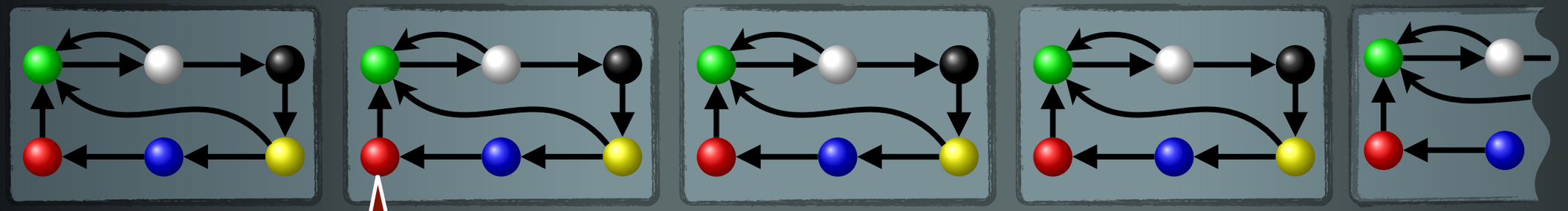
critical section

Bad configuration:

- **2 or more processes in critical section**

Bad Configurations



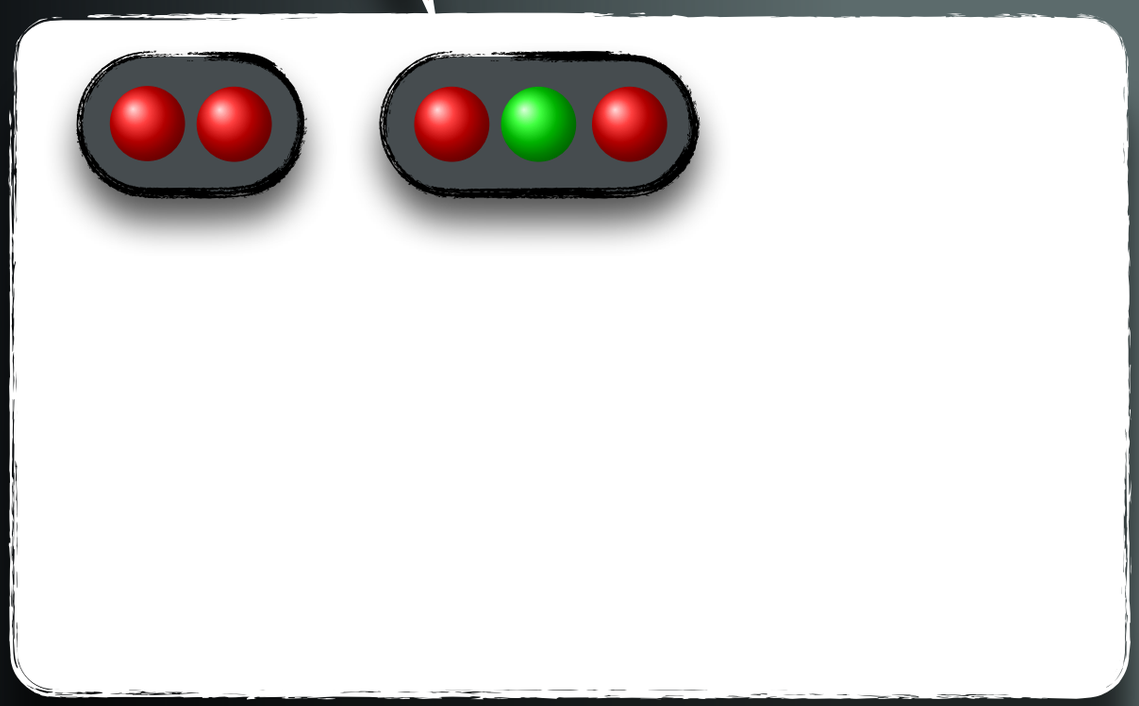


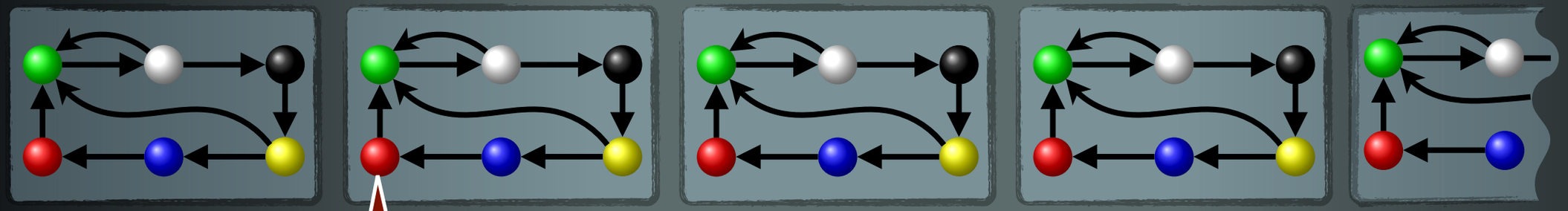
critical section

Bad configuration:

- **2 or more processes in critical section**

Bad Configurations



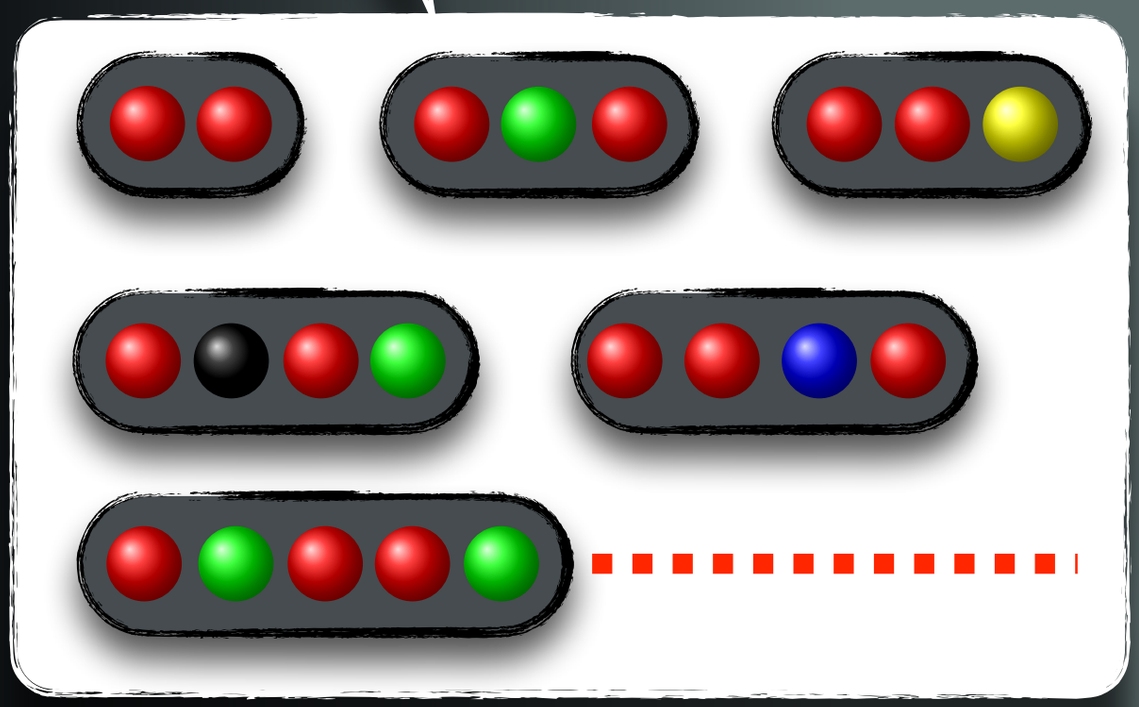


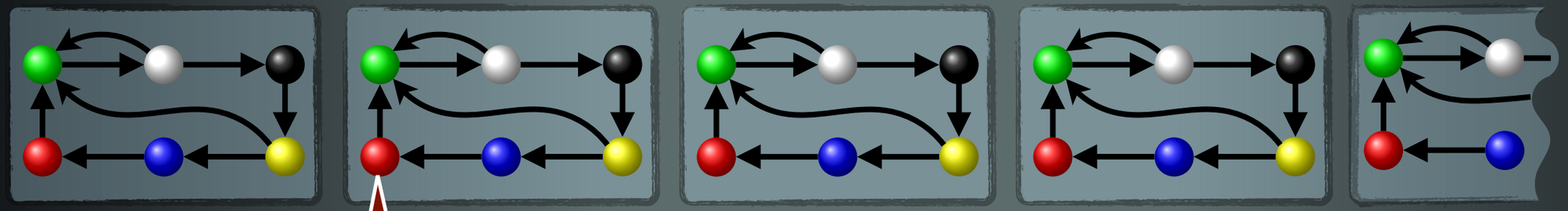
critical section

Bad configuration:

- **2 or more processes in critical section**

Bad Configurations



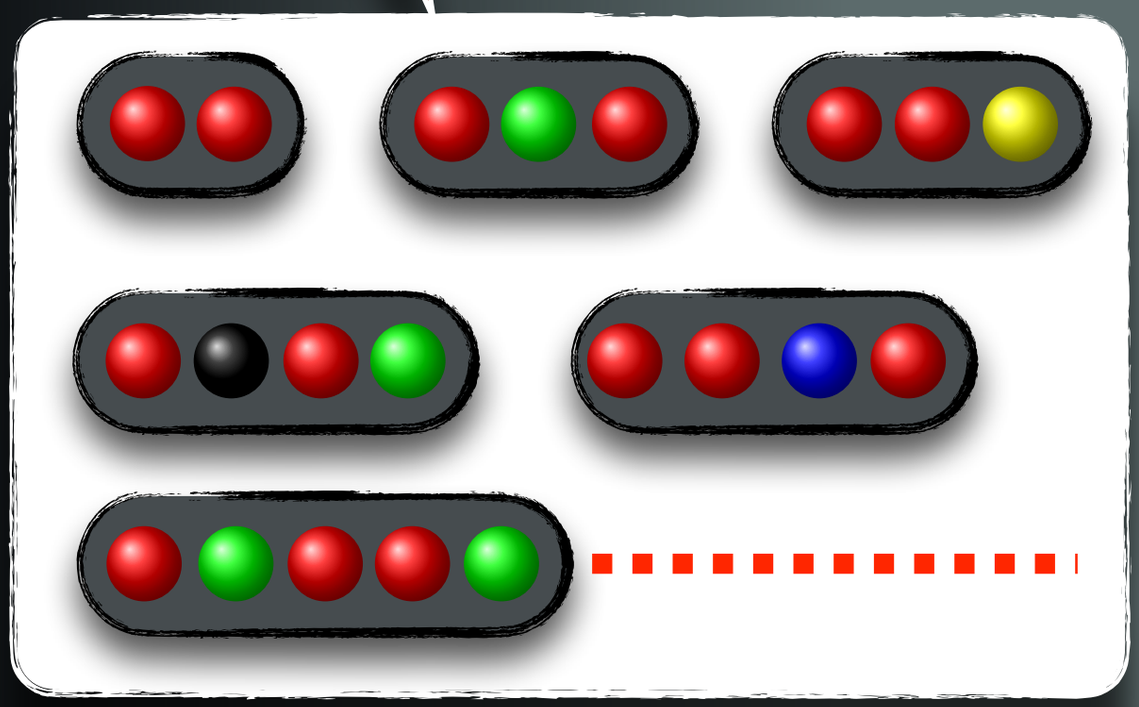


critical section

Bad configuration:

- **2 or more processes in critical section**

Bad Configurations



Set of bad configurations:

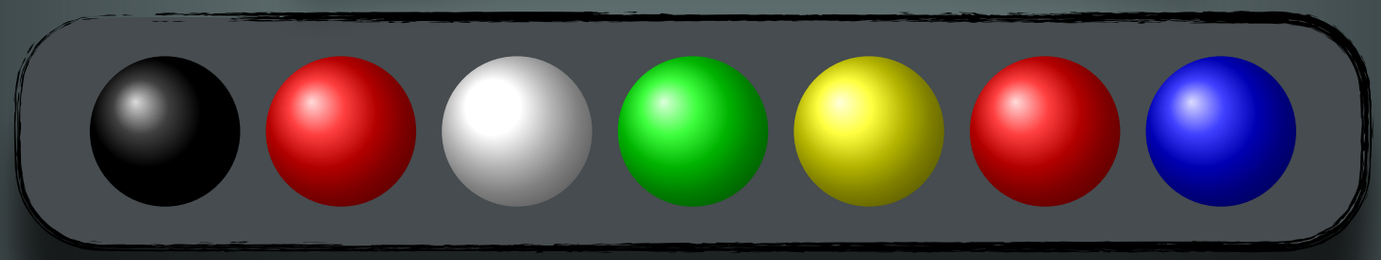
- **infinite, but**
- **“upward closed”**

Safety Property:

- $Init \xrightarrow{*} Bad$?

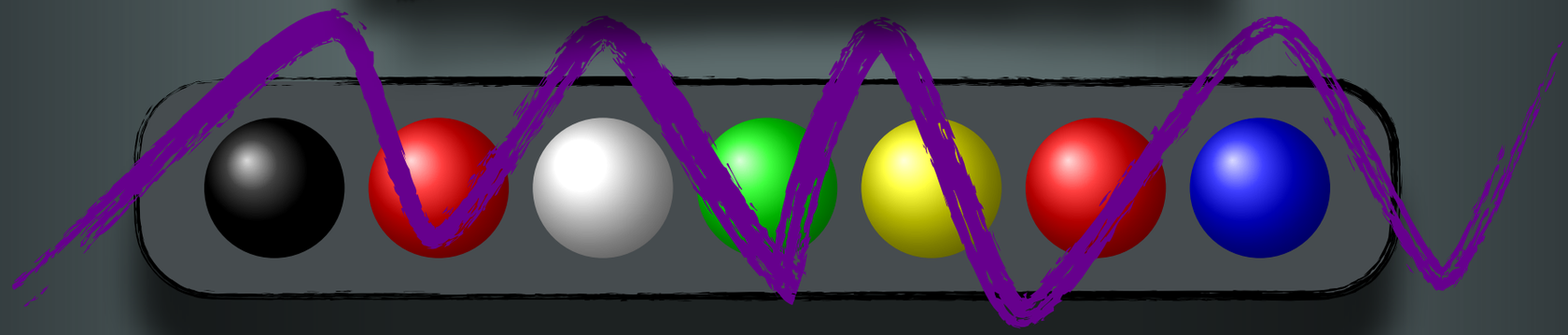
Safety Property:

- $Init \xrightarrow{*} Bad$?



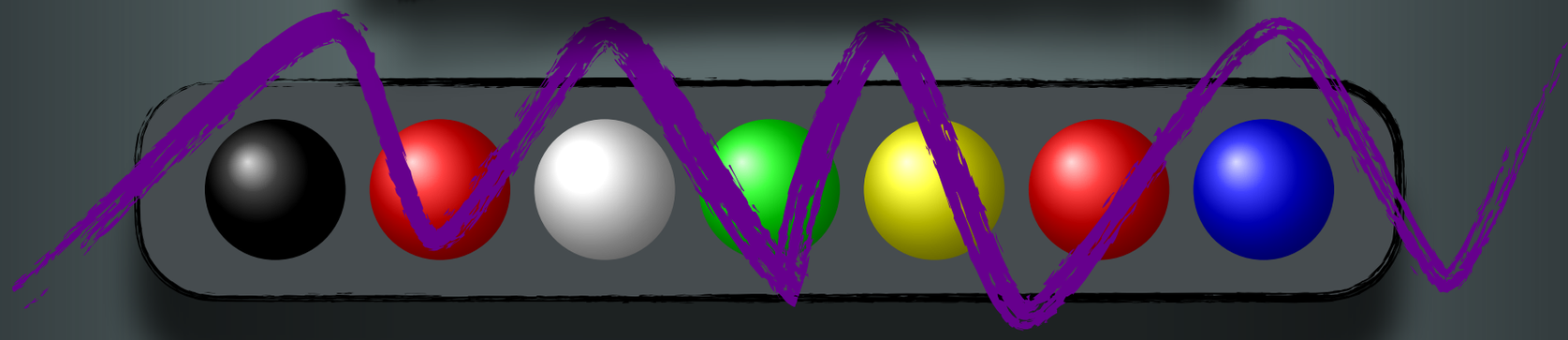
Safety Property:

- $Init \xrightarrow{*} Bad$?



Safety Property:

- $Init \xrightarrow{*} Bad$?

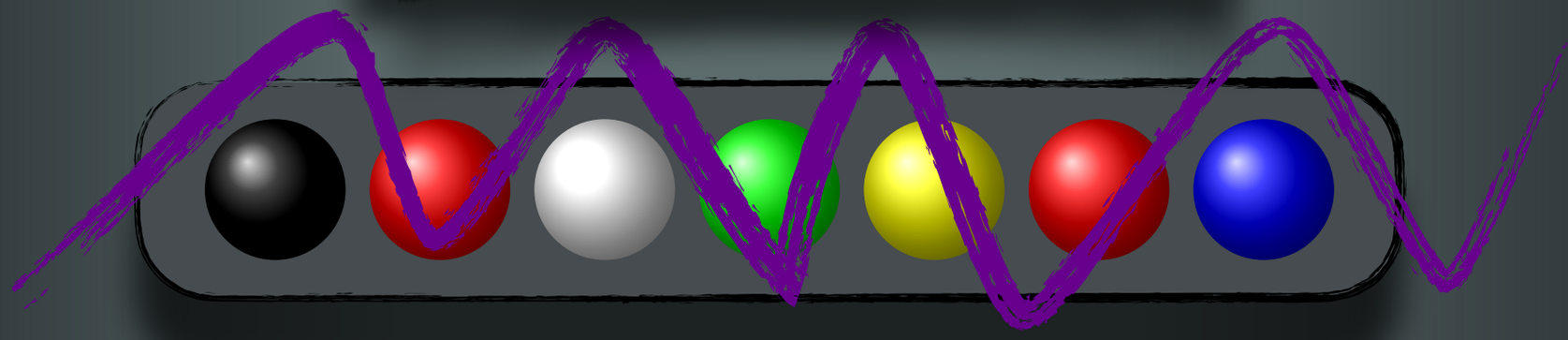


Parameterized Verification:

- **verify correctness regardless of # of processes**
- **# processes: parameter of the system**

Safety Property:

- $Init \xrightarrow{*} Bad$?



Parameterized Verification:

- verify correctness regardless of # of processes
- # processes: parameter of the system

Remarks

- infinite-state system:
  - unbounded number of processes
- Problem undecidable in general
  - challenge: find abstractions that work “often”

Parameterized Systems

- **Mutual Exclusion**
  - Burns
  - Dijkstra
  - Szymanski
- **Cache coherence**
  - MOSI
  - German

Parameterized Systems

- **Mutual Exclusion**
  - Burns
  - Dijkstra
  - Szymanski
- **Cache coherence**
  - MOSI
  - German
- **Petri Nets**
- **Trees**
- **Rings**

Parameterized Systems

Analysis / Abstraction

Features

Completeness

Parameterized Systems ✓

Analysis / Abstraction

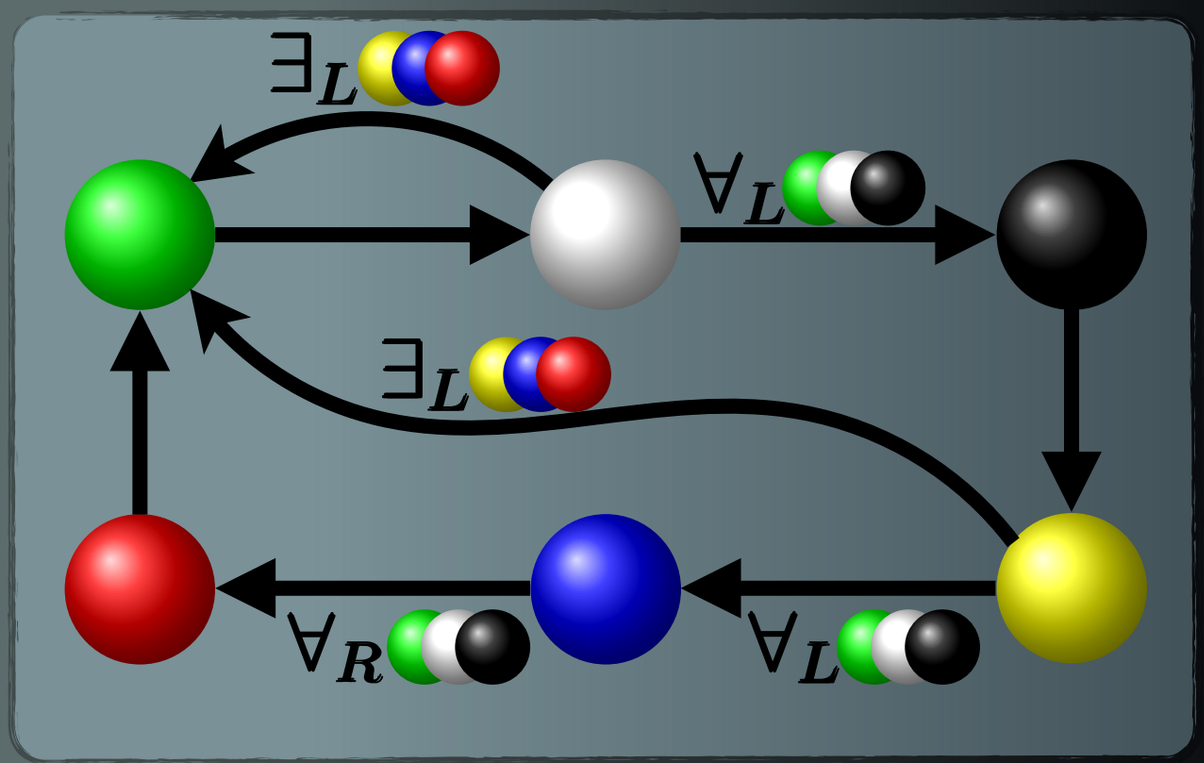
Features

Completeness


Analysis
Abstraction

Cut-off:

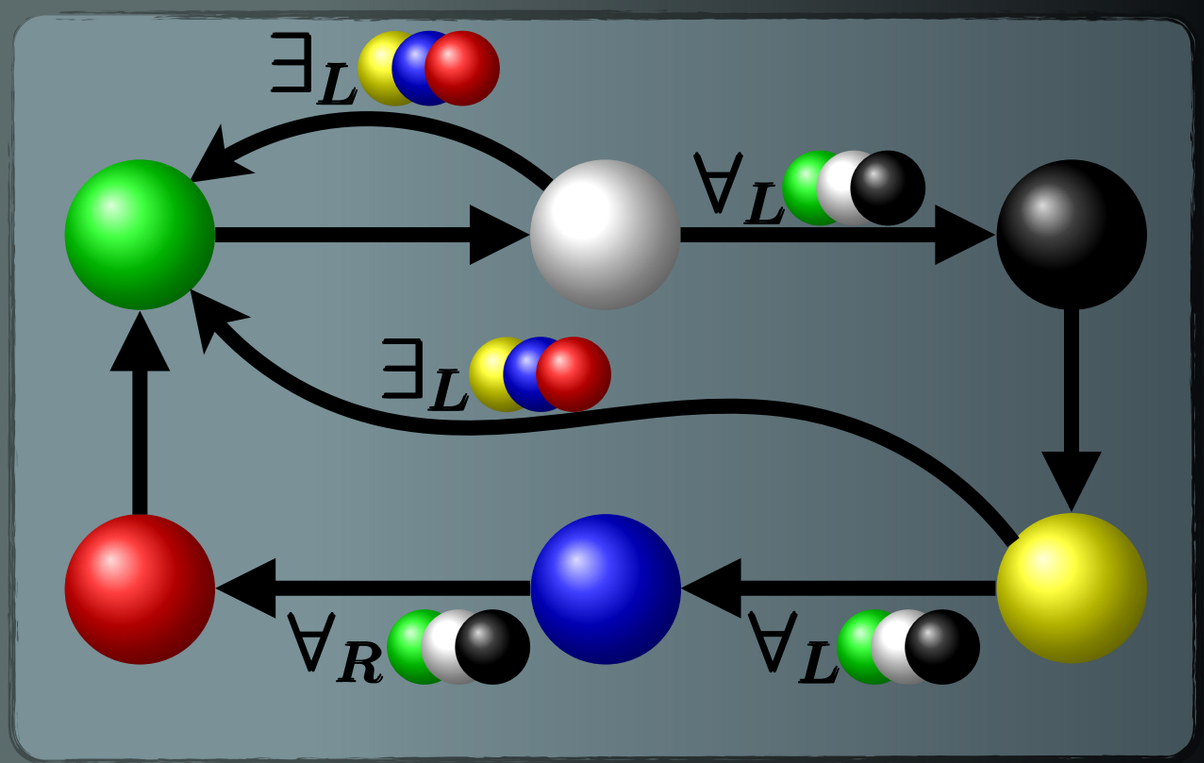
- **inspect small instances of the system**
- **efficient method**



Cut-off:

- **inspect small instances of the system**
- **efficient method**

- **Goal: safety**
- **infinite family**

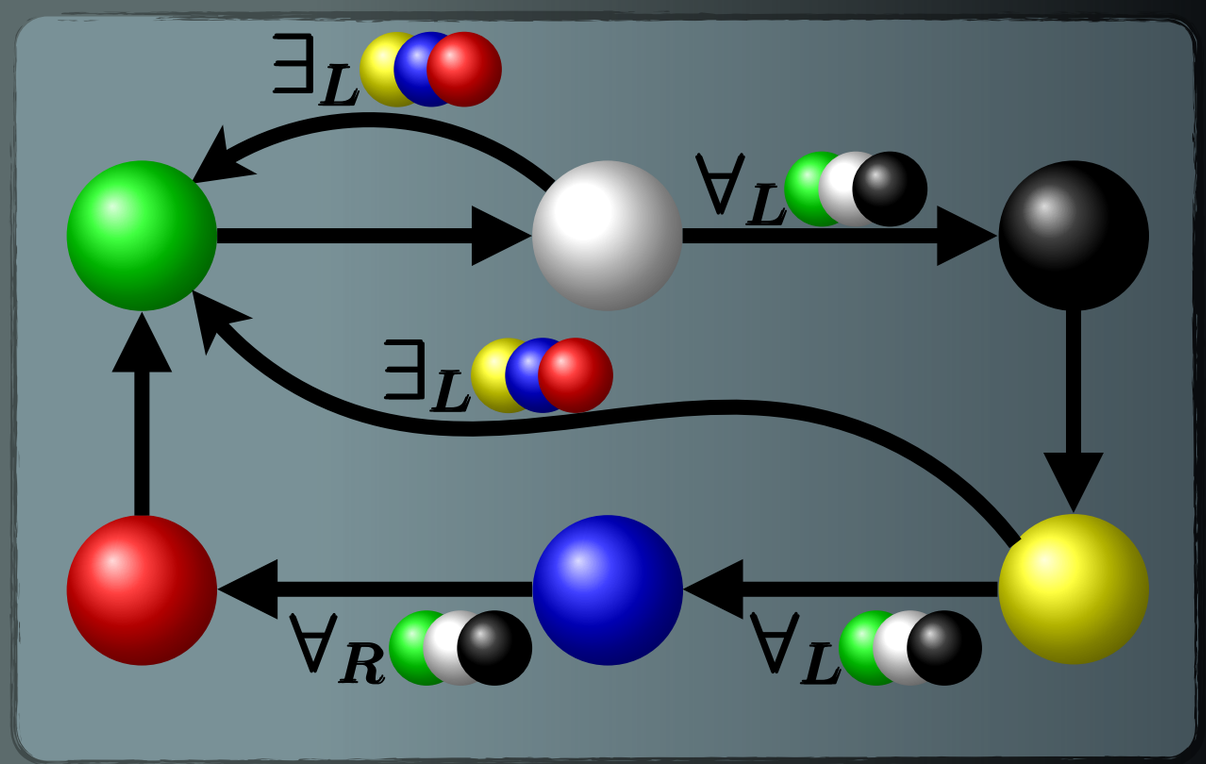


Cut-off:

- **inspect small instances of the system**
- **efficient method**

- **Goal: safety**
- **infinite family**



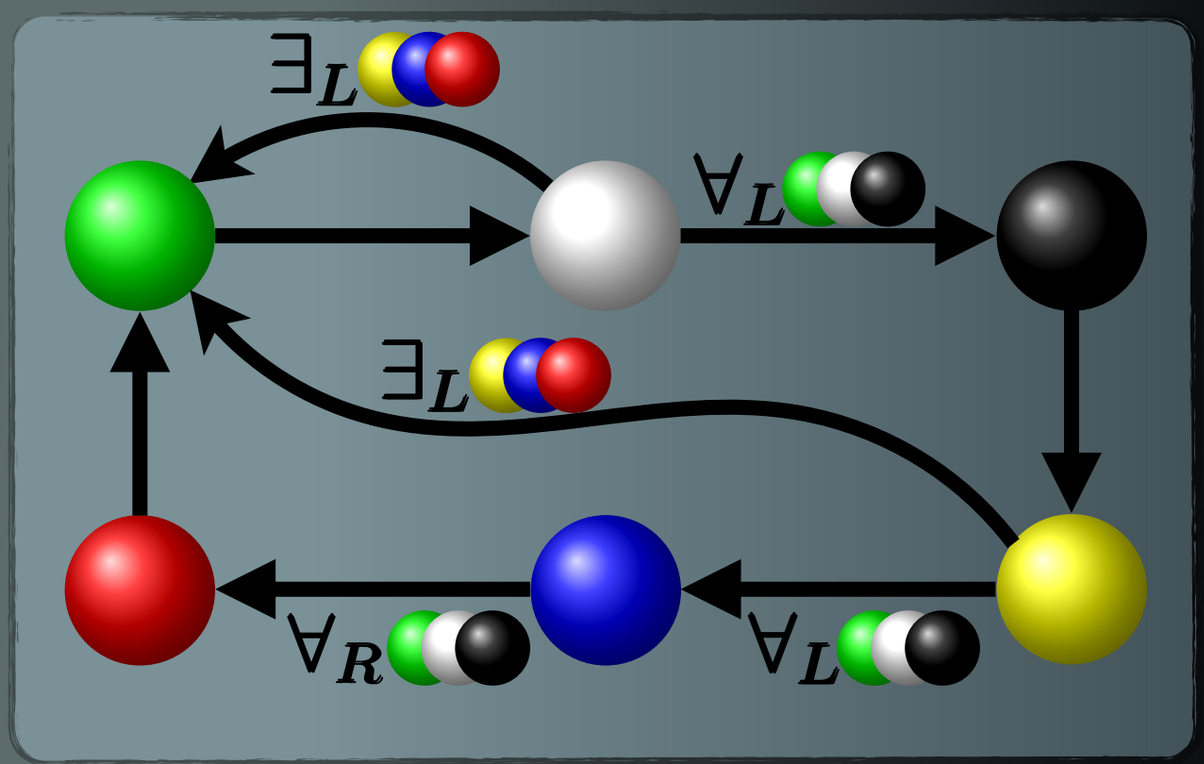


Cut-off:

- **inspect small instances of the system**
- **efficient method**

- **Goal: safety**
- **infinite family**

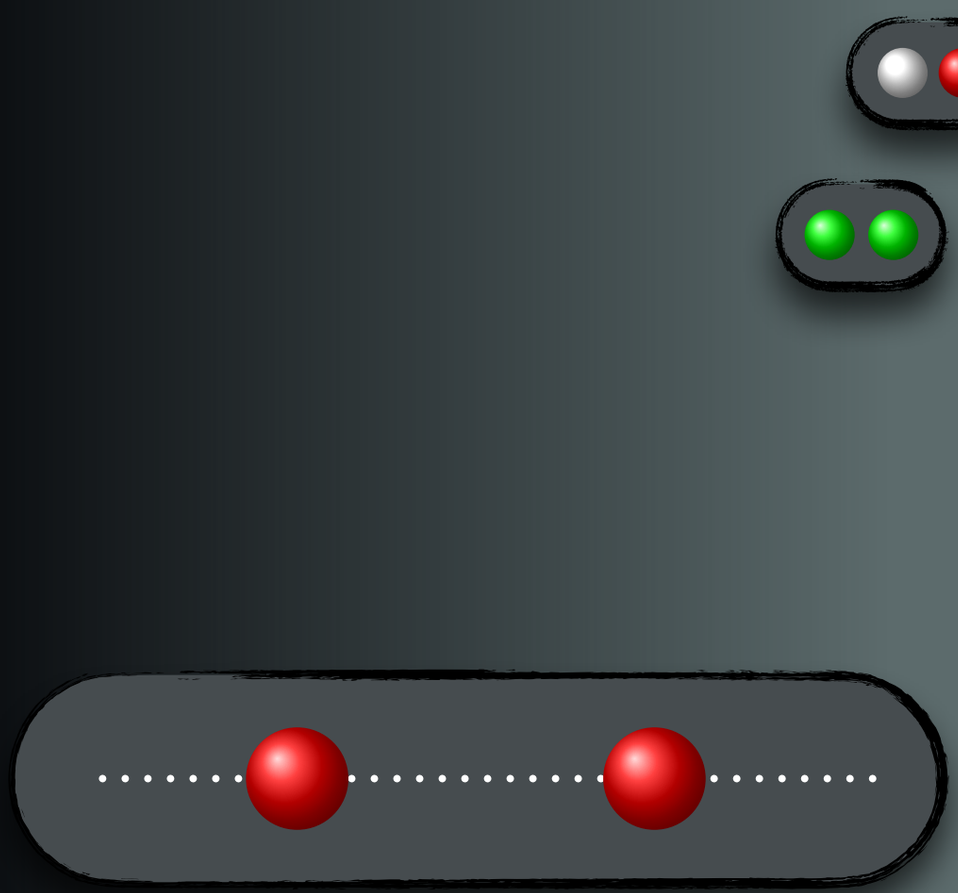


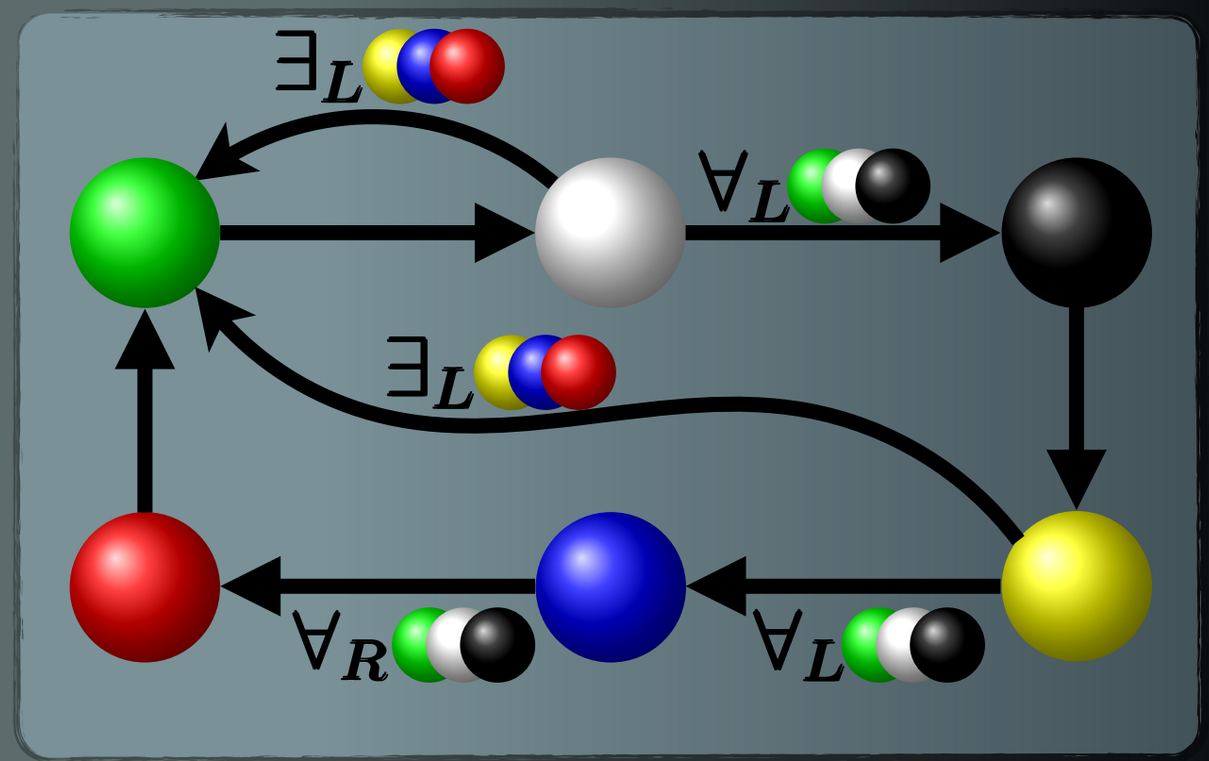


Cut-off:

- **inspect small instances of the system**
- **efficient method**

- **Goal: safety**
- **infinite family**

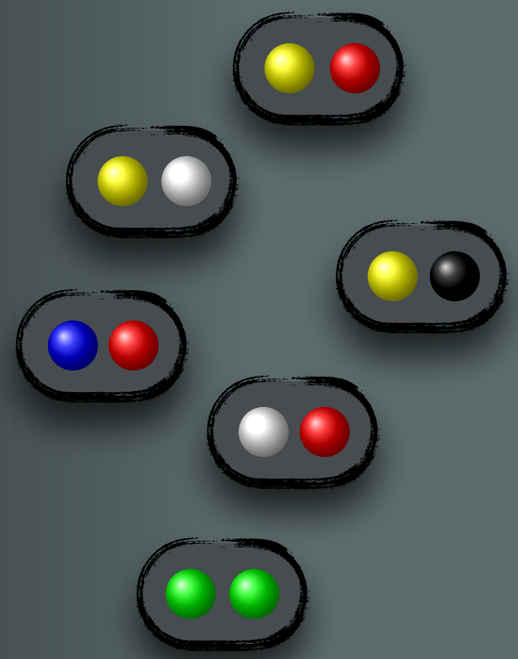




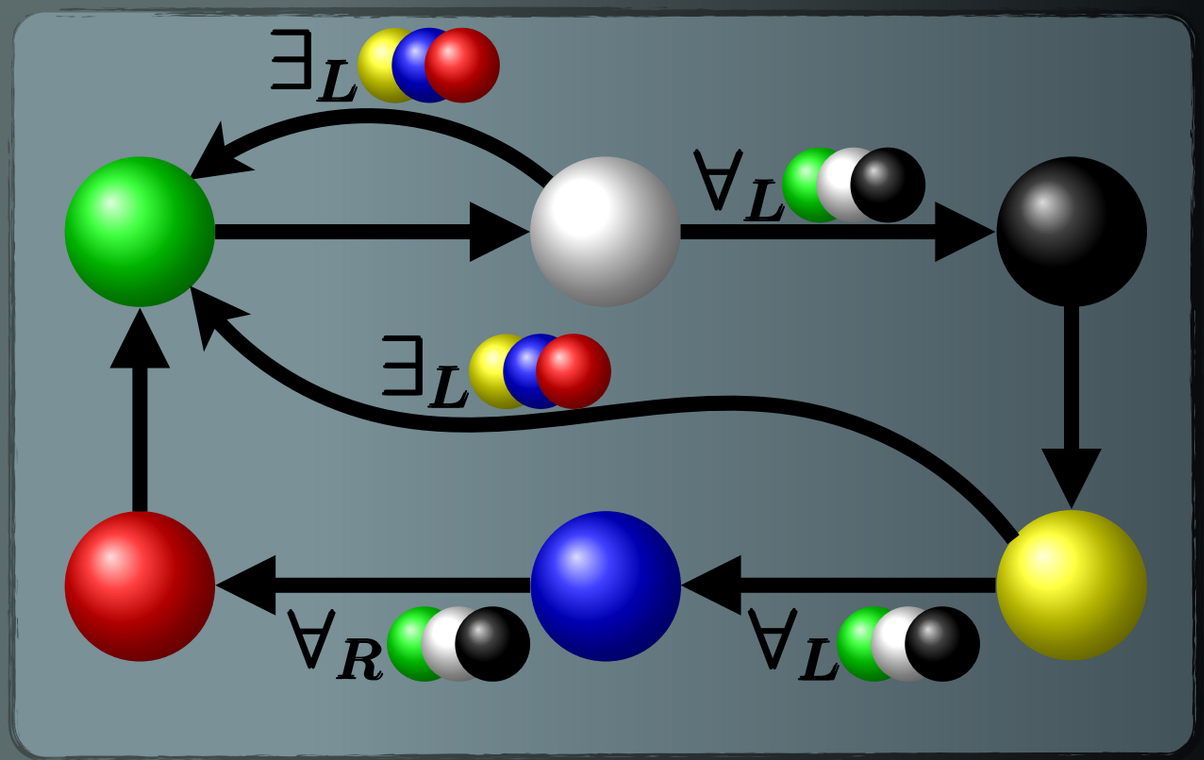
Cut-off:

- **inspect small instances of the system**
- **efficient method**

- **Goal: safety**
- **infinite family**



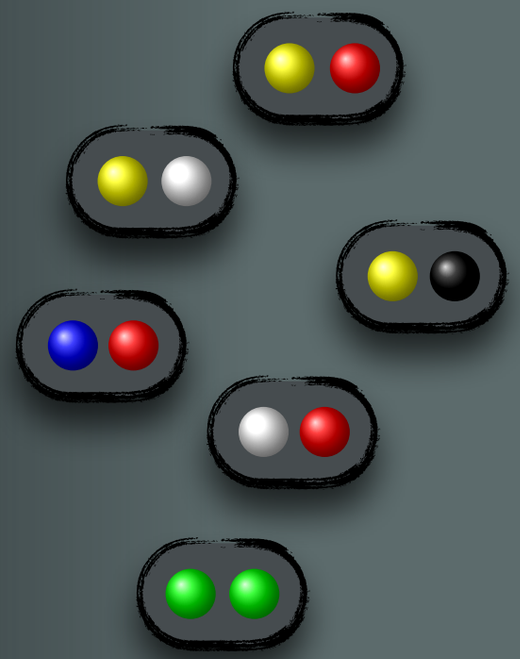




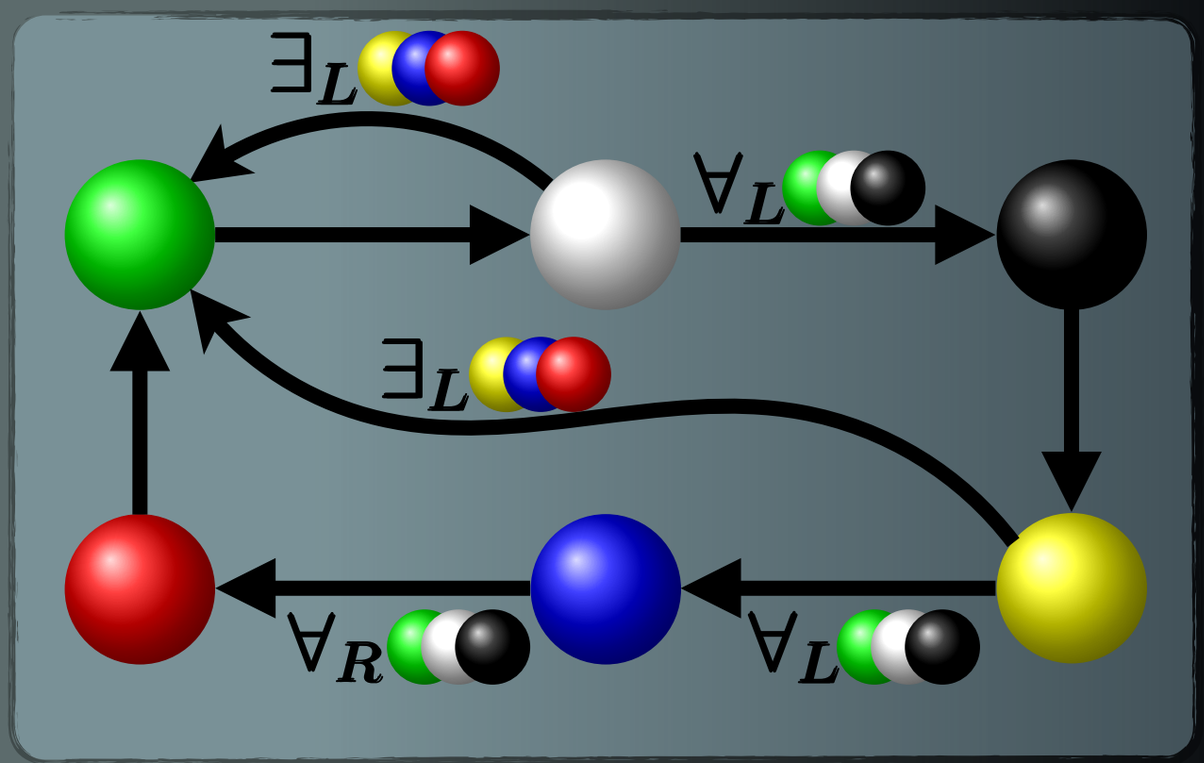
Cut-off:

- **inspect small instances of the system**
- **efficient method**

- **Goal: safety**
- **infinite family**



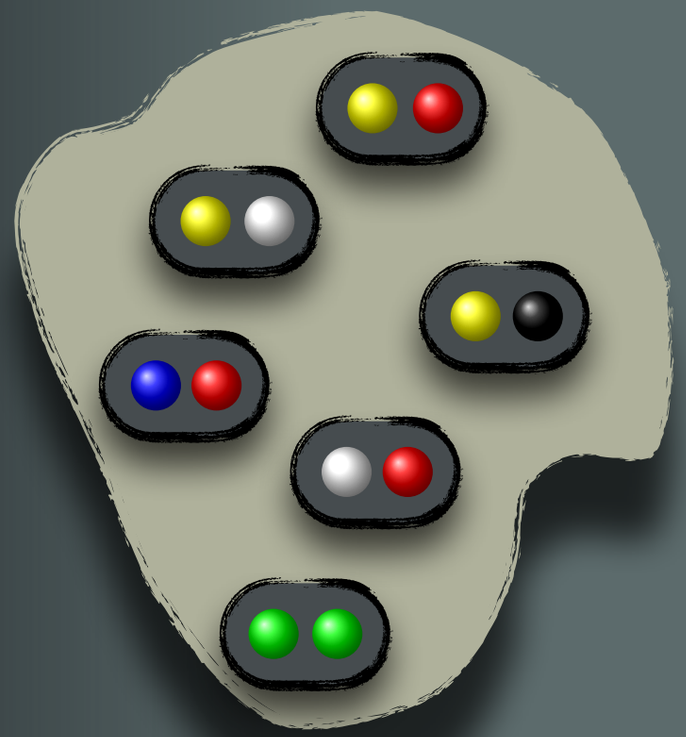




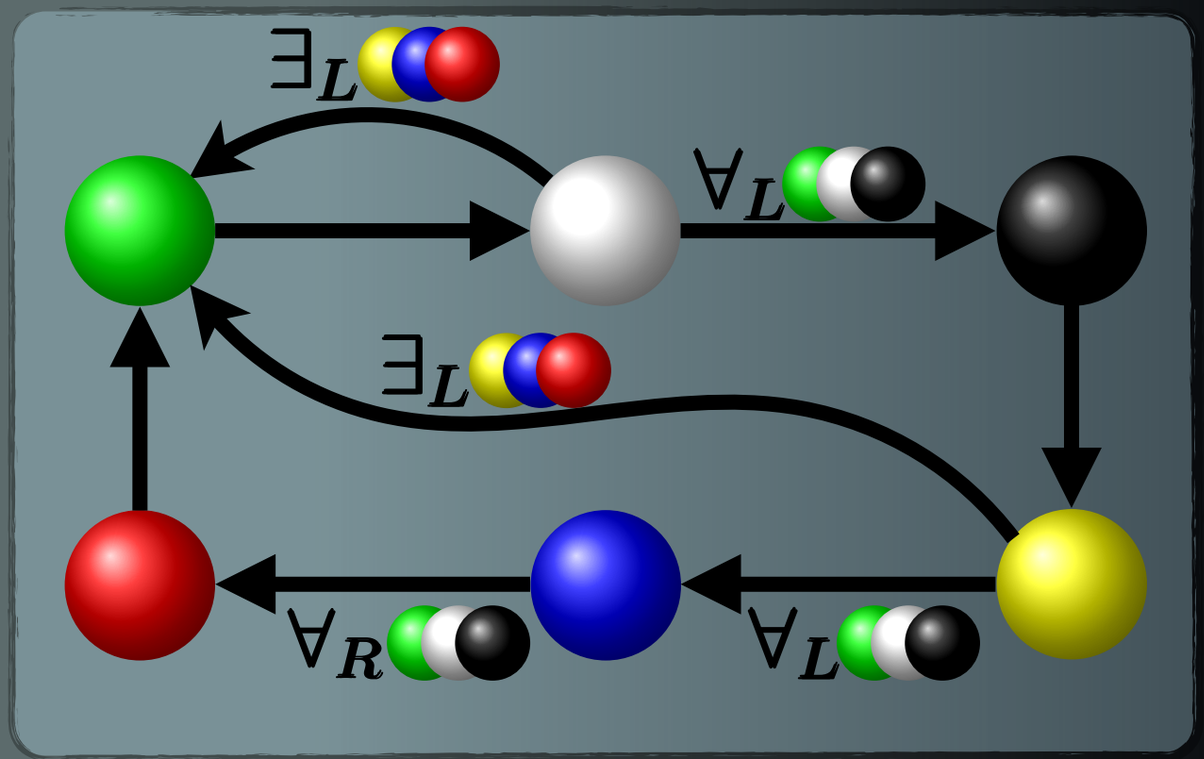
Cut-off:

- **inspect small instances of the system**
- **efficient method**

- **Goal: safety**
- **infinite family**



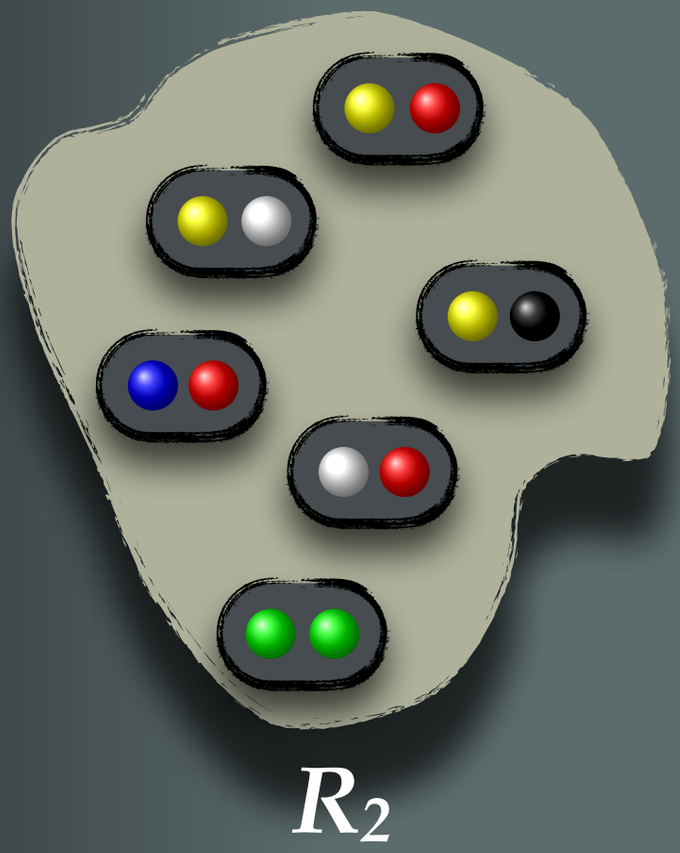




Cut-off:

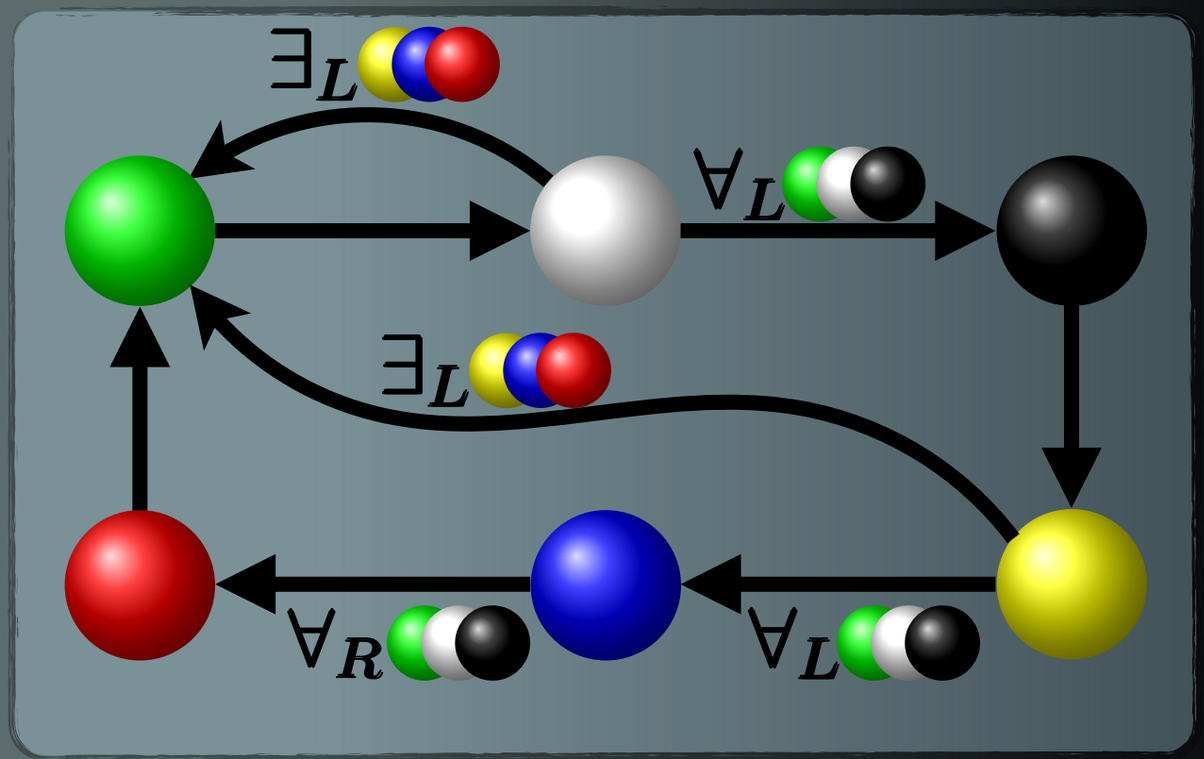
- **inspect small instances of the system**
- **efficient method**

- **Goal: safety**
- **infinite family**



R_2

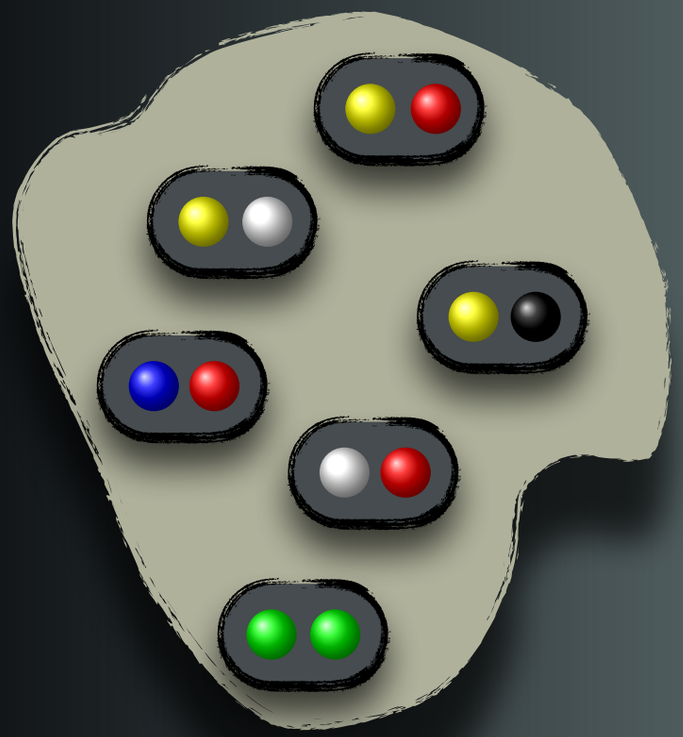




Cut-off:

- **inspect small instances of the system**
- **efficient method**

- **Goal: safety**
- **infinite family**



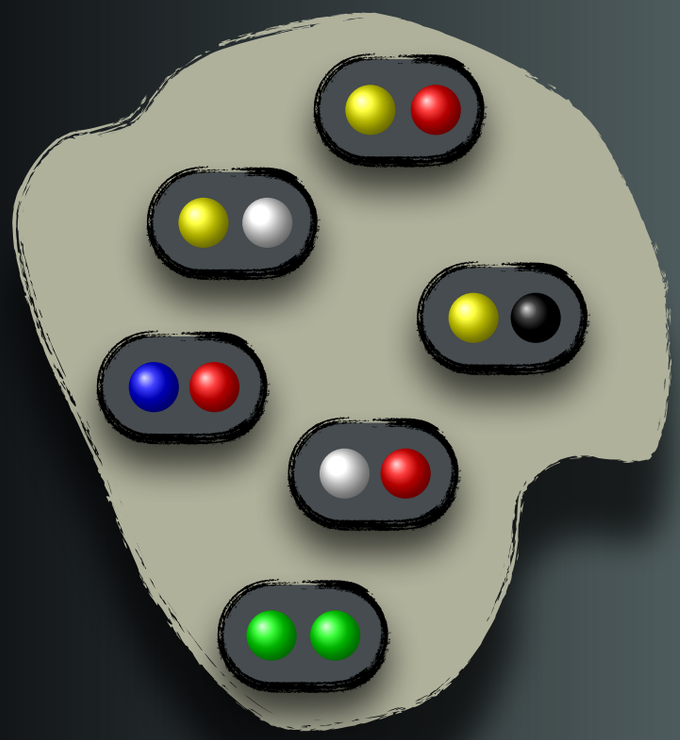
R_2



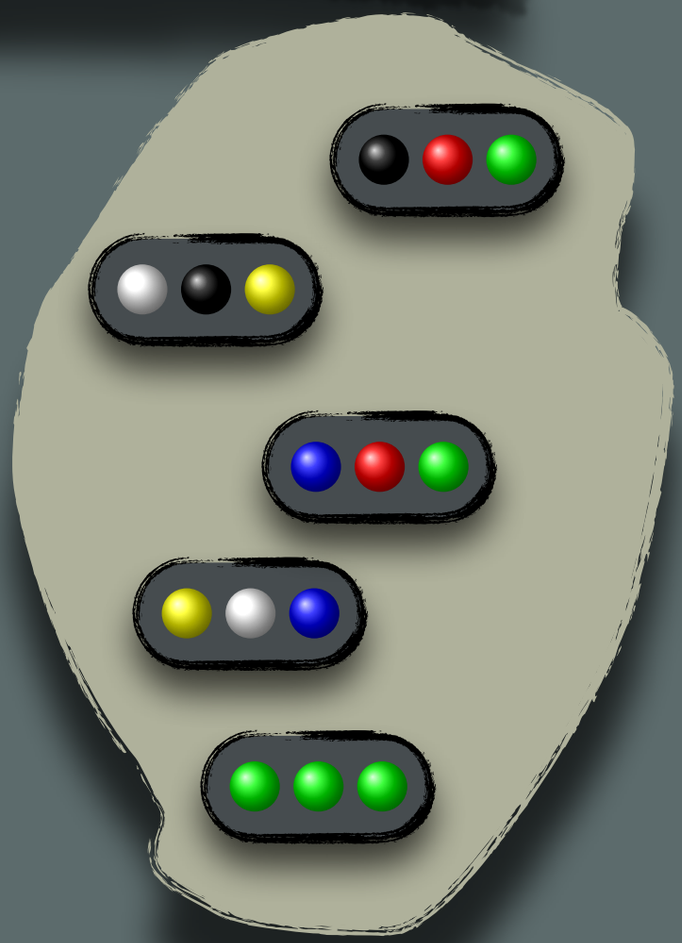
Cut-off:

- **inspect small instances of the system**
- **efficient method**

- **Goal: safety**
- **infinite family**



R_2



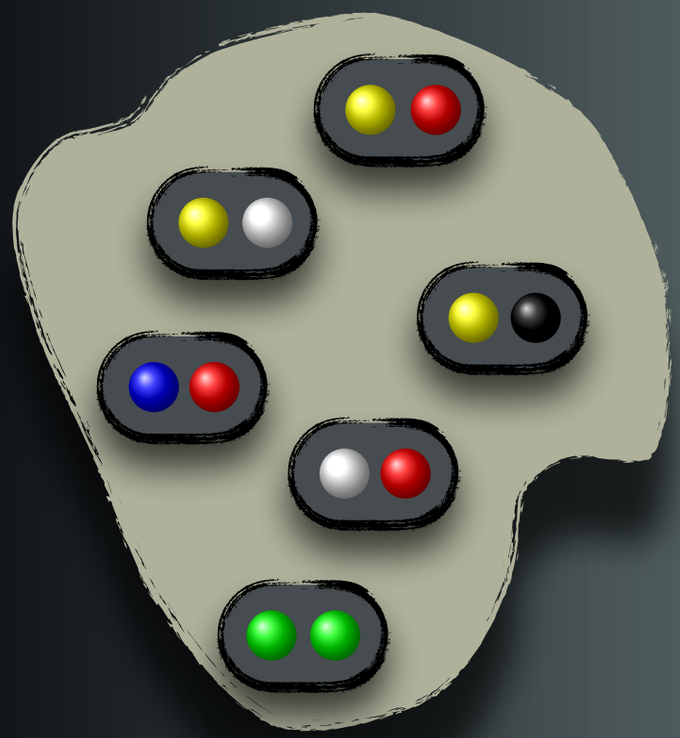
R_3



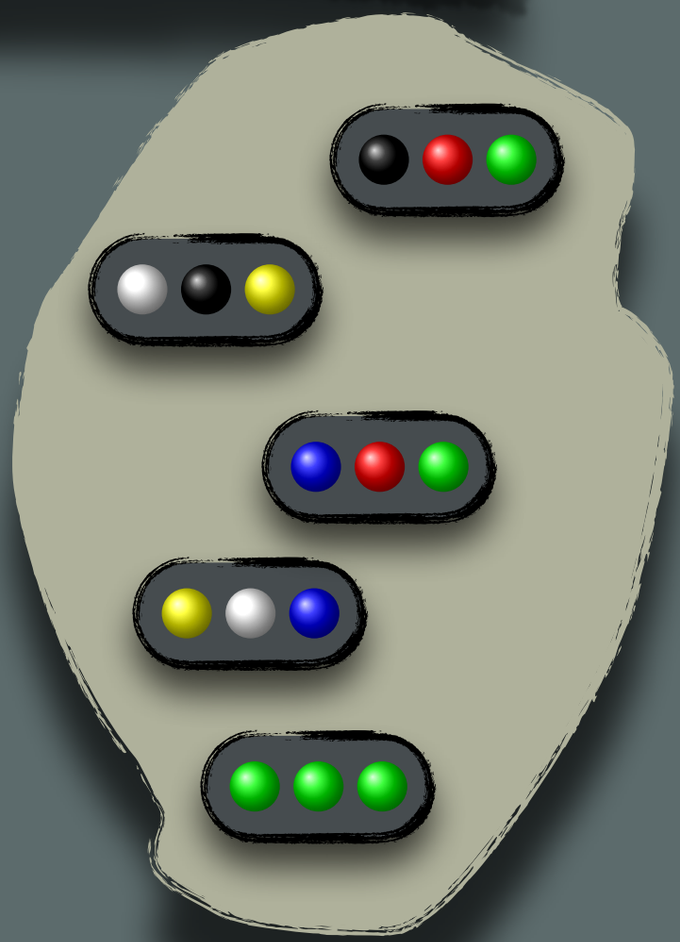
Cut-off:

- **inspect small instances of the system**
- **efficient method**

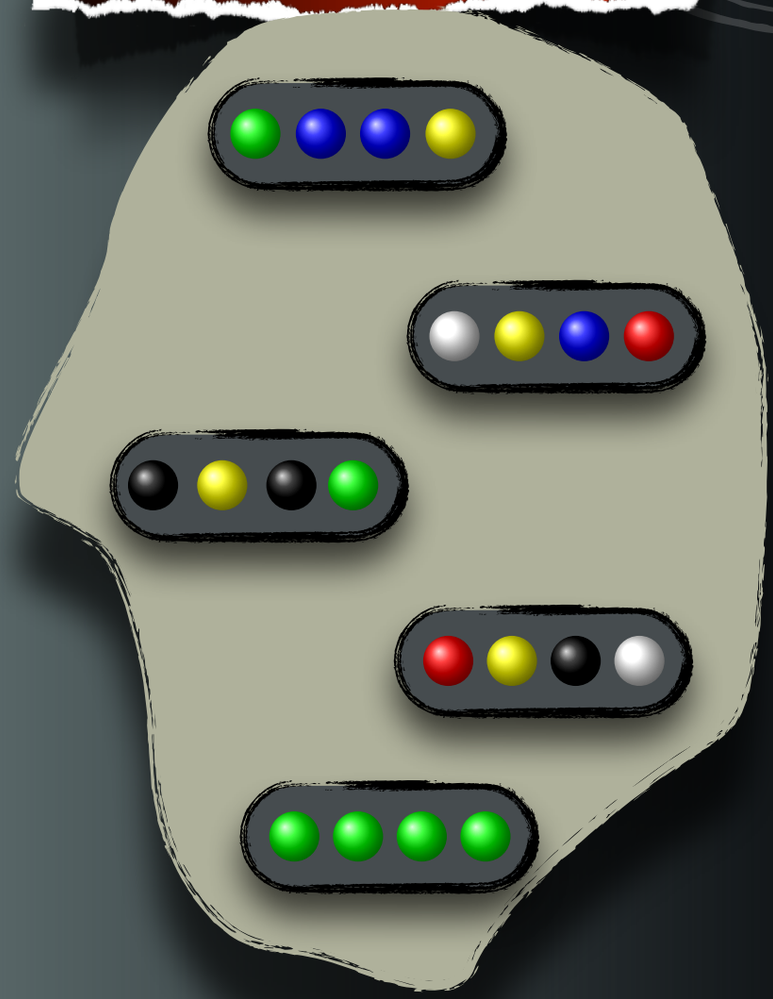
- **Goal: safety**
- **infinite family**



R_2



R_3



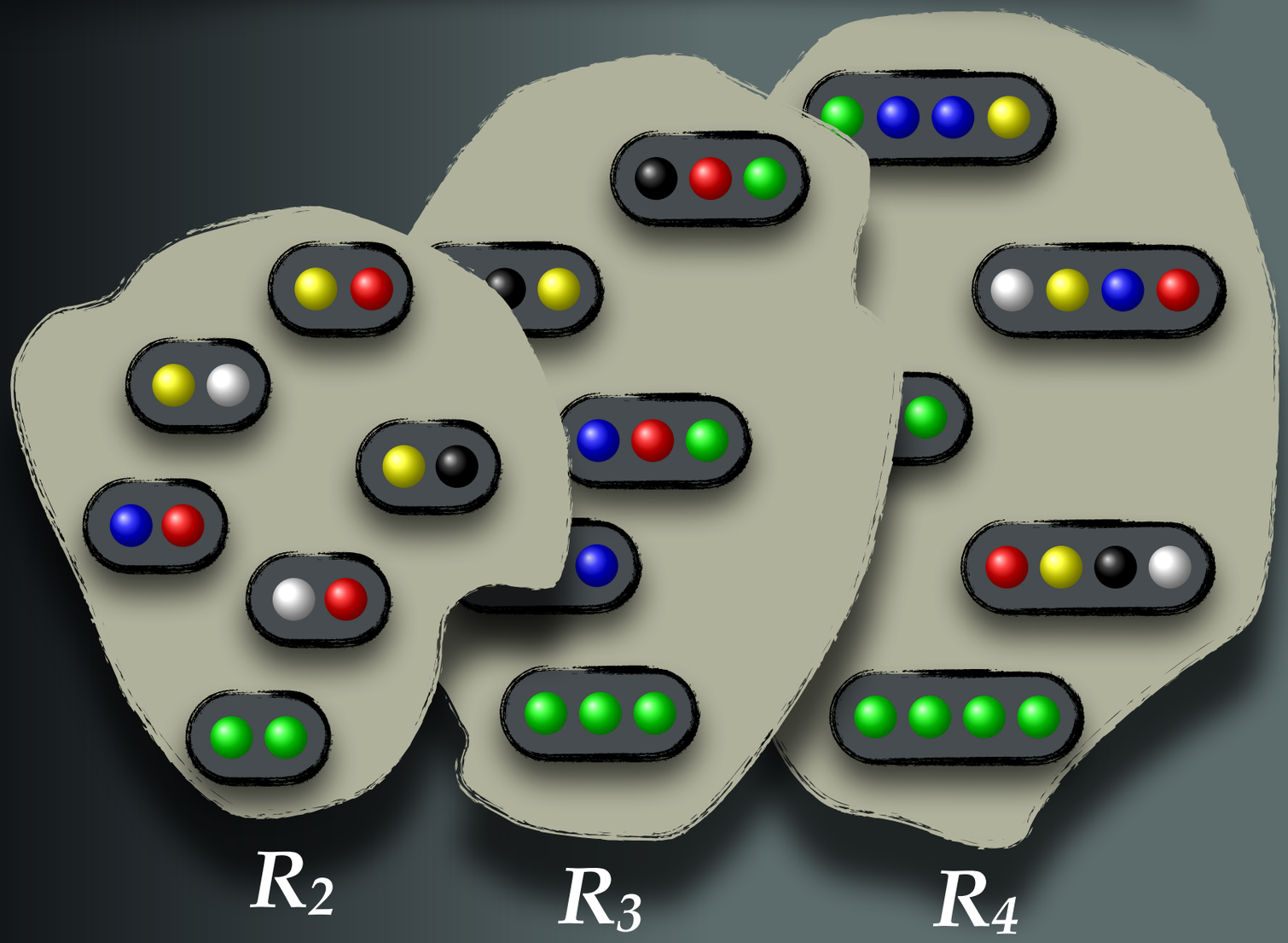
R_4



Cut-off:

- **inspect small instances of the system**
- **efficient method**

- **Goal: safety**
- **infinite family**

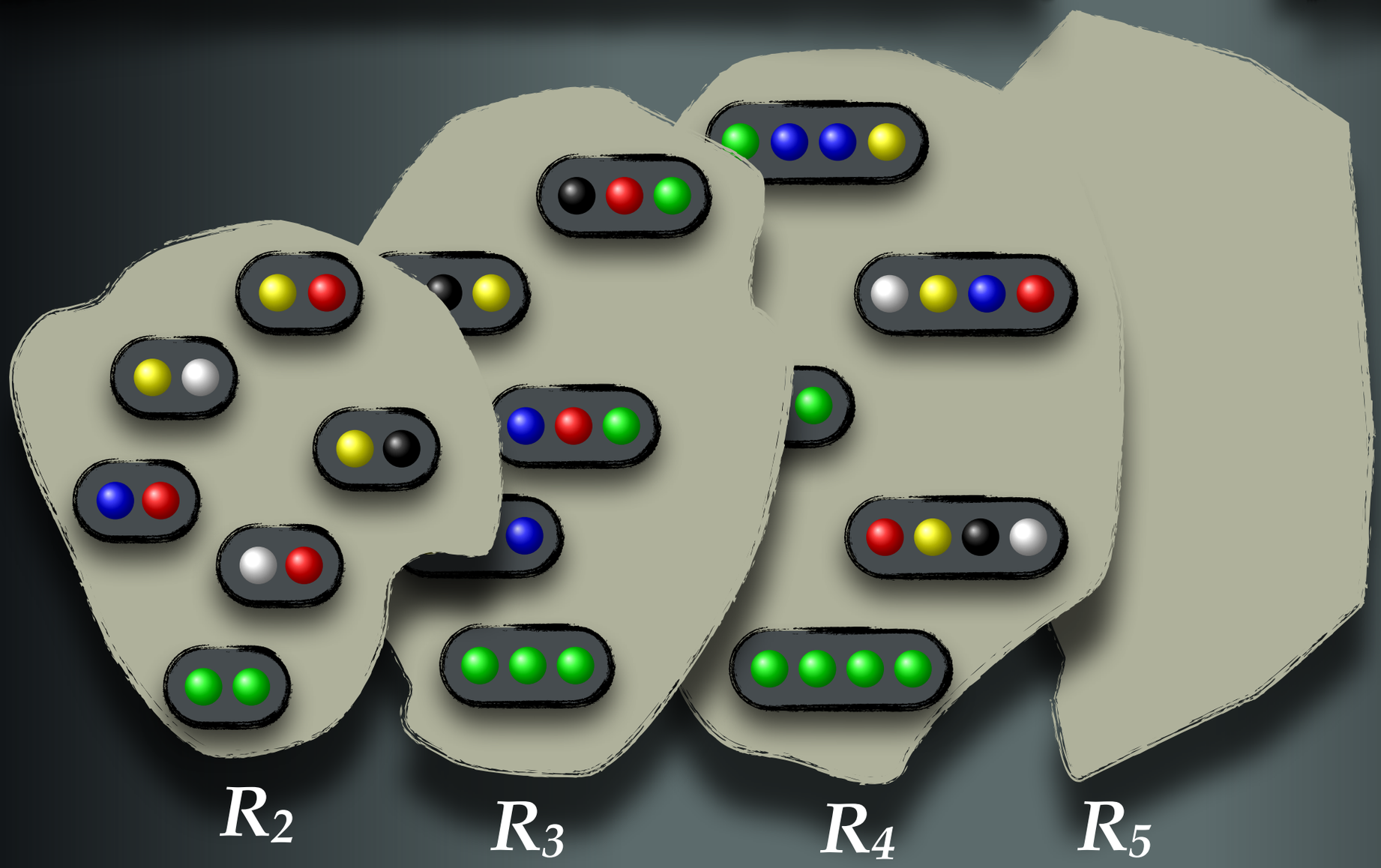




Cut-off:

- **inspect small instances of the system**
- **efficient method**

- **Goal: safety**
- **infinite family**



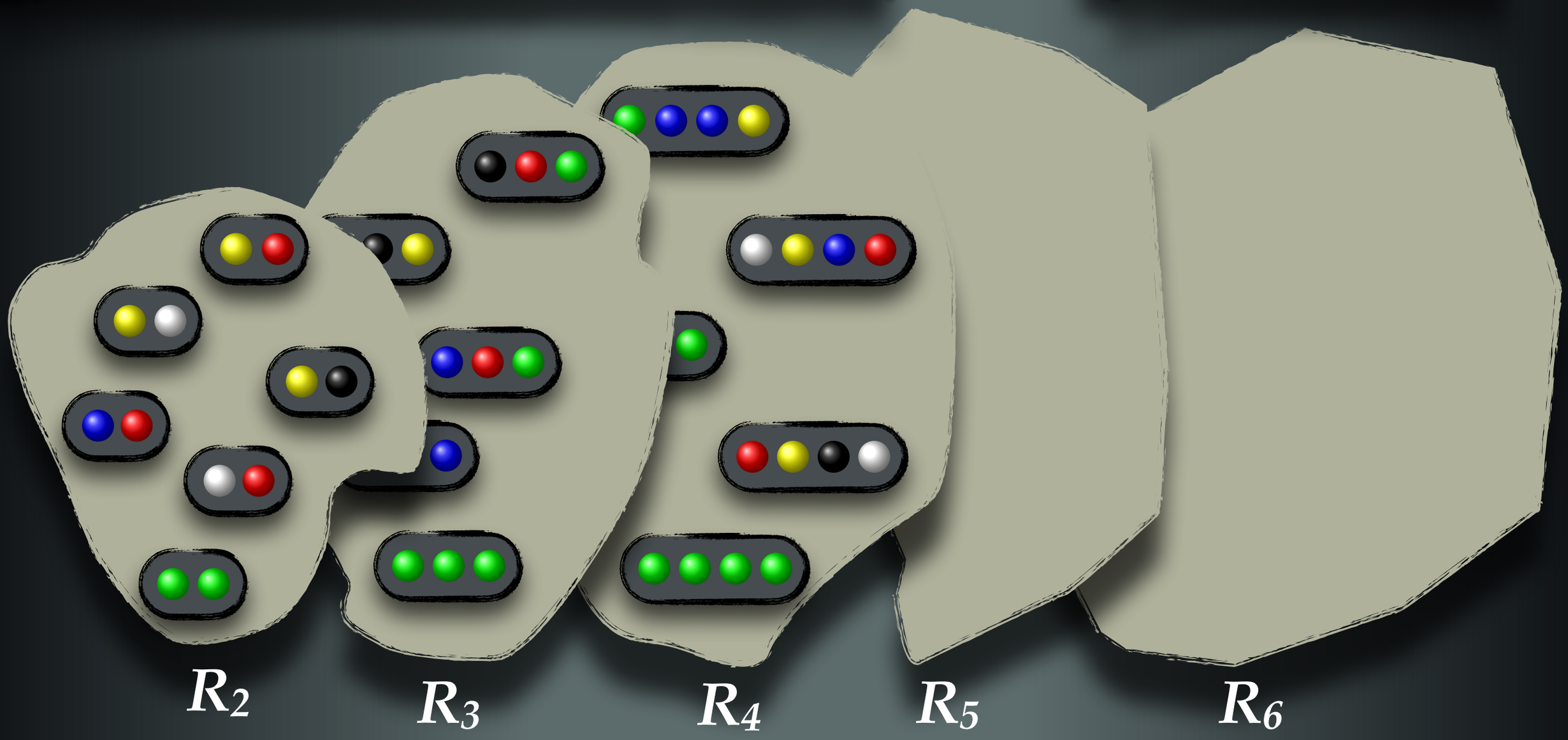
R_2 R_3 R_4 R_5



Cut-off:

- **inspect small instances of the system**
- **efficient method**

- **Goal: safety**
- **infinite family**

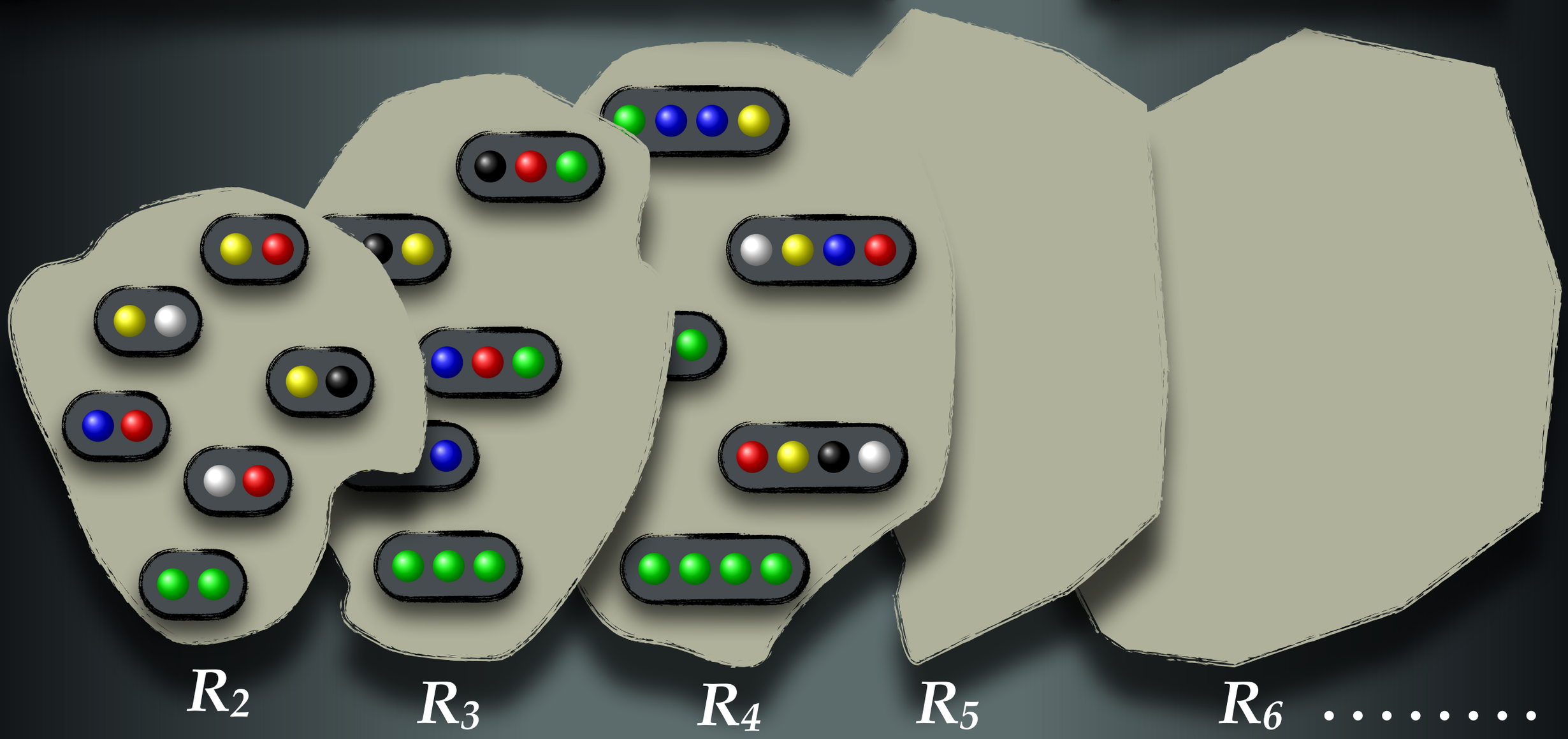




Cut-off:

- **inspect small instances of the system**
- **efficient method**

- **Goal: safety**
- **infinite family**

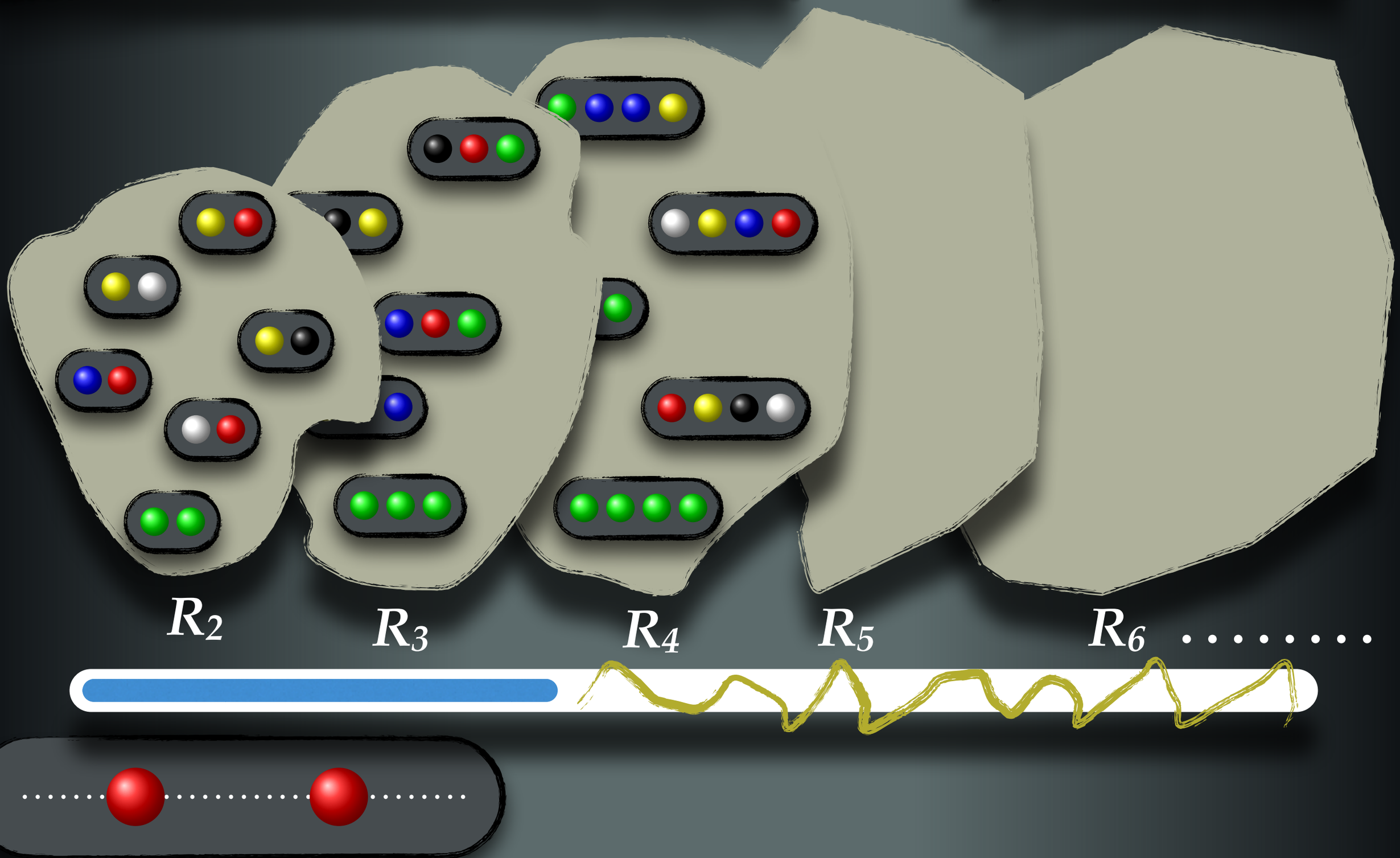




Cut-off:

- **inspect small instances of the system**
- **efficient method**

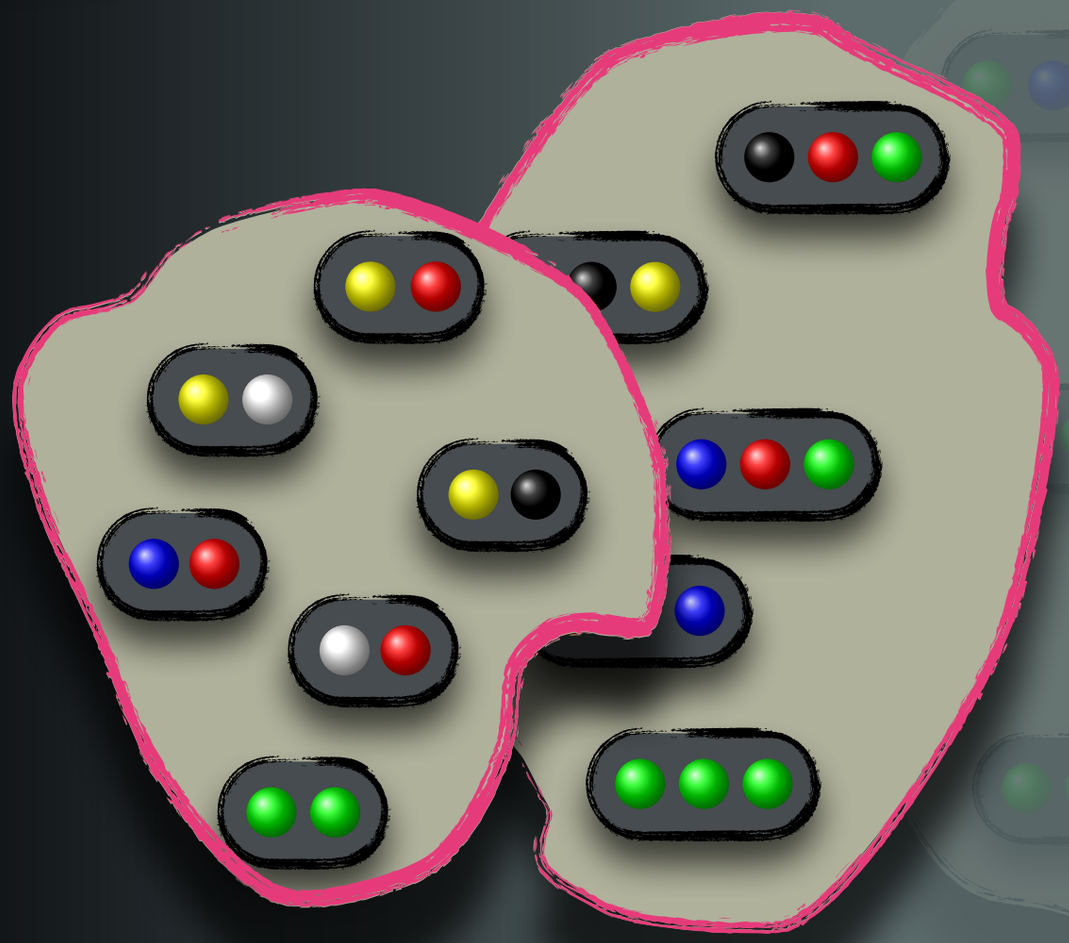
- **Goal: safety**
- **infinite family**



Cut-off:

- **inspect small instances of the system**
- **efficient method**

- **Goal: safety**
- **infinite family**



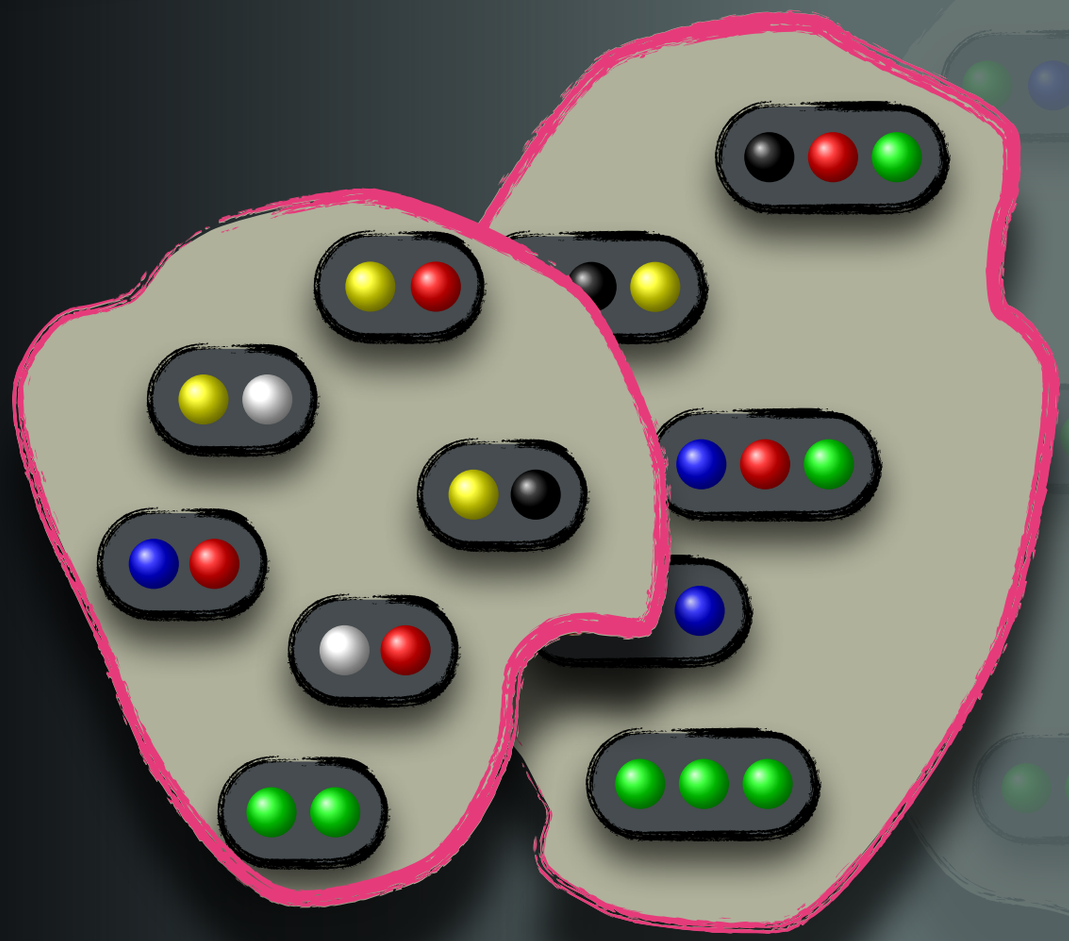
R_2 R_3 R_4 R_5 R_6 . . . . . .



Cut-off:

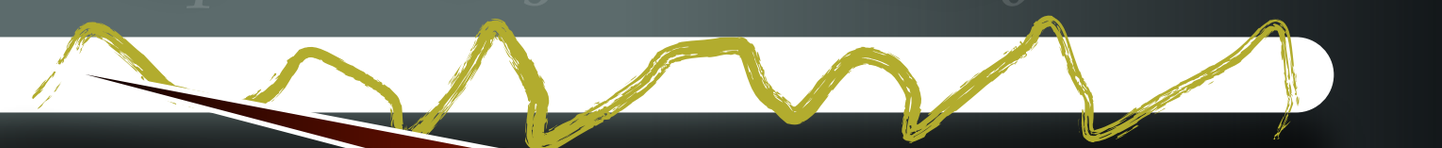
- **inspect small instances of the system**
- **efficient method**

- **Goal: safety**
- **infinite family**



R_2 R_3 R_4 R_5 R_6 . . . . . .



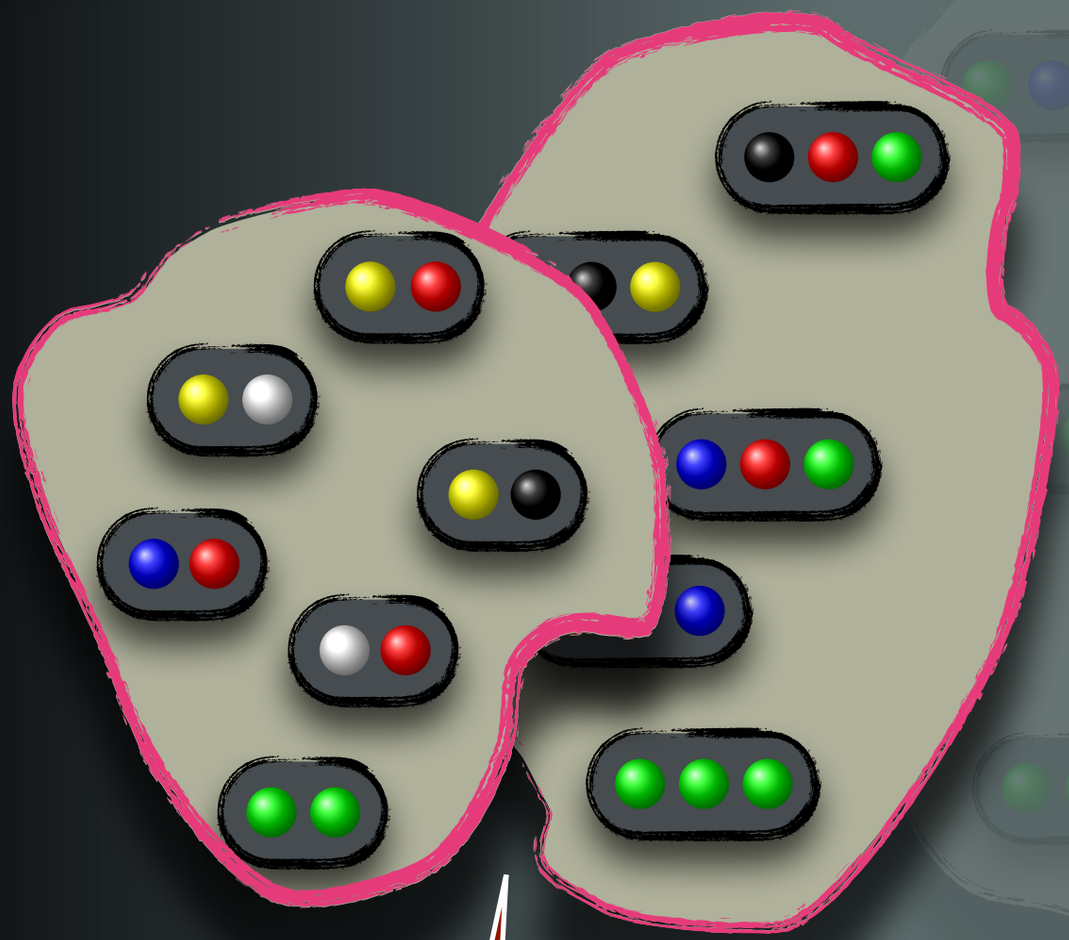


cutoff point

Cut-off:

- **inspect small instances of the system**
- **efficient method**

- **Goal: safety**
- **infinite family**



R_2 R_3 R_4 R_5 R_6 . . . . . .

- **small configurations**
- **efficient method**

cutoff point

Intuition:

- **Bad configurations**
  - **can be characterized by fixed number of witness processes**
- **Good invariants:**
  - **can be demonstrated through small system instances**

- abstraction modulo **k**
- **k:** natural number



- abstraction modulo **k**
- **k:** natural number

$\mathbb{C}$: set of configurations

- abstraction modulo **k**
- **k:** natural number

$\mathbb{C}$: set of configurations

$\mathbb{V}_k := \{v \in \mathbb{C} \mid |v| \leq k\}$

- abstraction modulo **k**
- **k:** natural number

$\mathbb{C}$: set of configurations

$\mathbb{V}_k := \{v \in \mathbb{C} \mid |v| \leq k\}$

$\alpha_k : \mathbb{C} \to 2^{\mathbb{V}_k}$

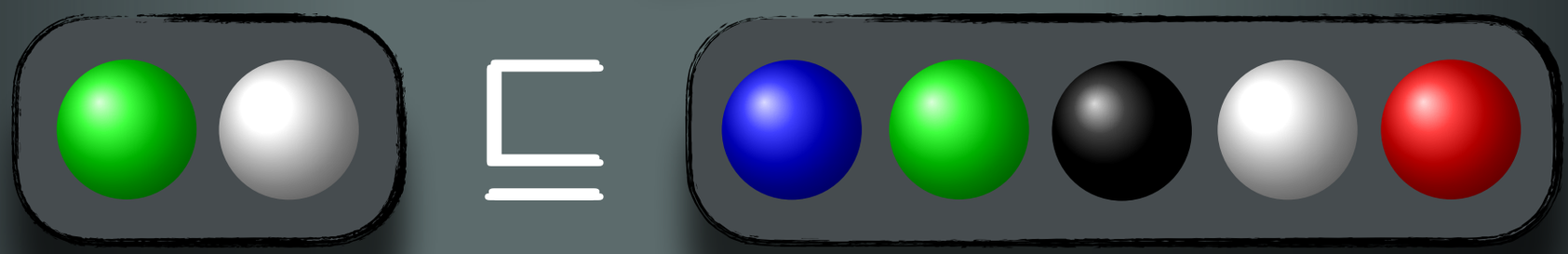
- abstraction modulo k
- k: natural number

$\mathbb{C}$: set of configurations

$\mathbb{V}_k := \{v \in \mathbb{C} \mid |v| \leq k\}$

$\alpha_k : \mathbb{C} \to 2^{\mathbb{V}_k}$

$\sqsubseteq$: subword relation



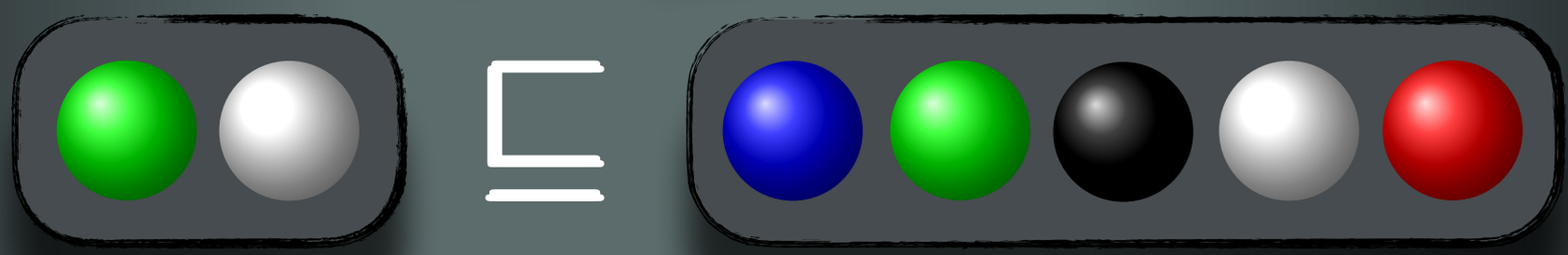
- abstraction modulo **k**
- **k:** natural number

$\mathbb{C}$: set of configurations

$\mathbb{V}_k := \{v \in \mathbb{C} \mid |v| \leq k\}$

$\alpha_k : \mathbb{C} \to 2^{\mathbb{V}_k}$

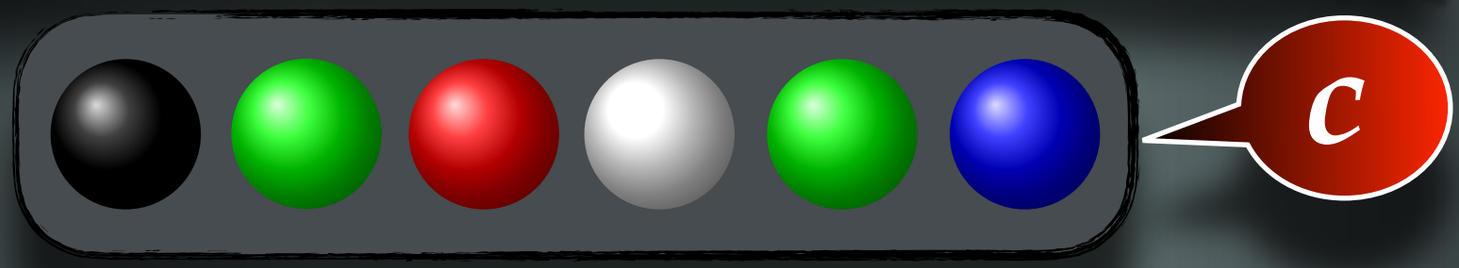
$\sqsubseteq$: subword relation



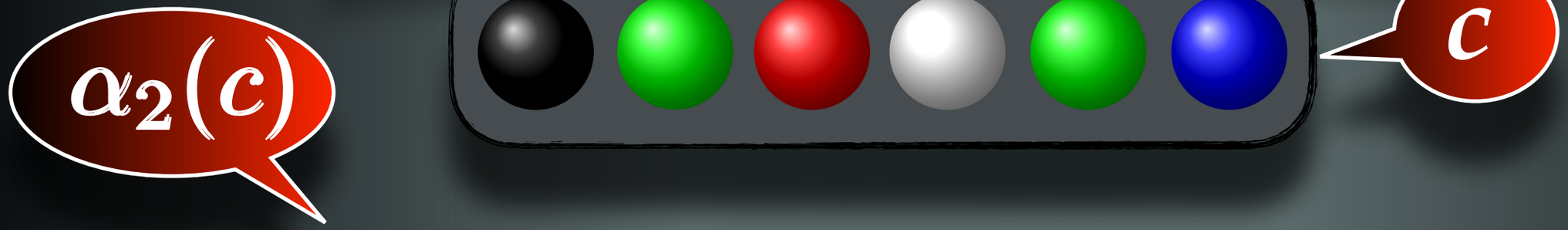
$\alpha_k(c) := \{v \in \mathbb{V}_k \mid v \sqsubseteq c\}$

$$\alpha_k(c) := \{v \in \mathbb{V}_k \mid v \sqsubseteq c\}$$

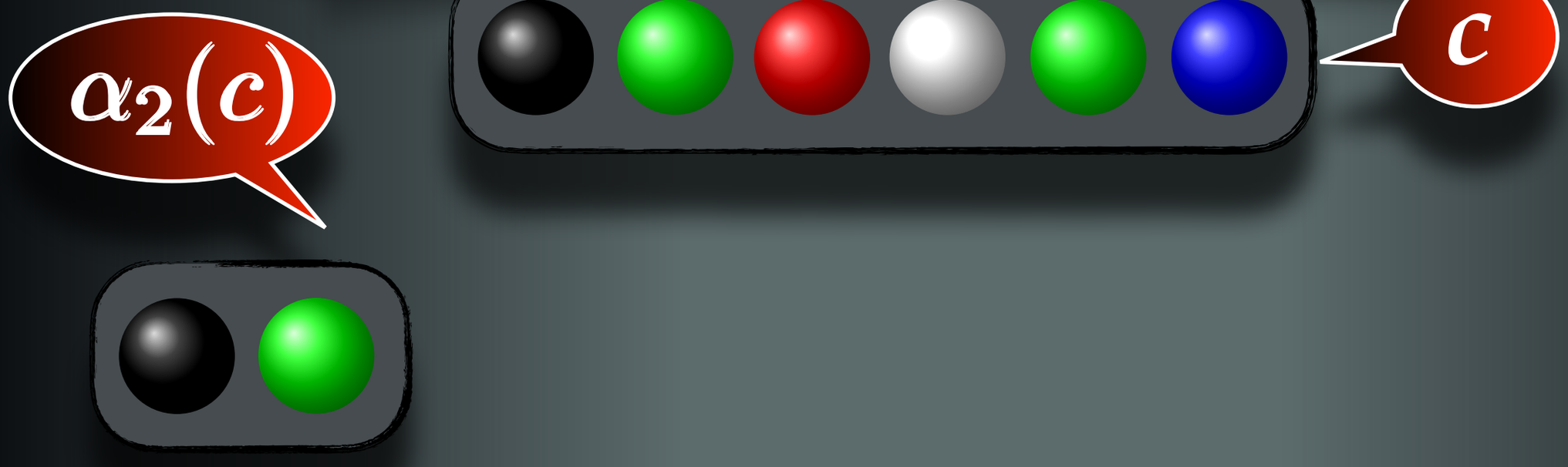
$$\alpha_k(c) := \{v \in \mathbb{V}_k \mid v \sqsubseteq c\}$$



$$\alpha_k(c) := \{v \in \mathbb{V}_k \mid v \sqsubseteq c\}$$



$$\alpha_k(c) := \{v \in \mathbb{V}_k \mid v \sqsubseteq c\}$$



$$\alpha_k(c) := \{v \in \mathbb{V}_k \mid v \sqsubseteq c\}$$



$$\alpha_k(c) := \{v \in \mathbb{V}_k \mid v \sqsubseteq c\}$$



$$\alpha_k(c) := \{v \in \mathbb{V}_k \mid v \sqsubseteq c\}$$



$$\alpha_k(c) := \{v \in \mathbb{V}_k \mid v \sqsubseteq c\}$$



$$\alpha_k(c) := \{v \in \mathbb{V}_k \mid v \sqsubseteq c\}$$



$$\alpha_k(c) := \{v \in \mathbb{V}_k \mid v \sqsubseteq c\}$$



$$\alpha_k(c) := \{v \in \mathbb{V}_k \mid v \sqsubseteq c\}$$

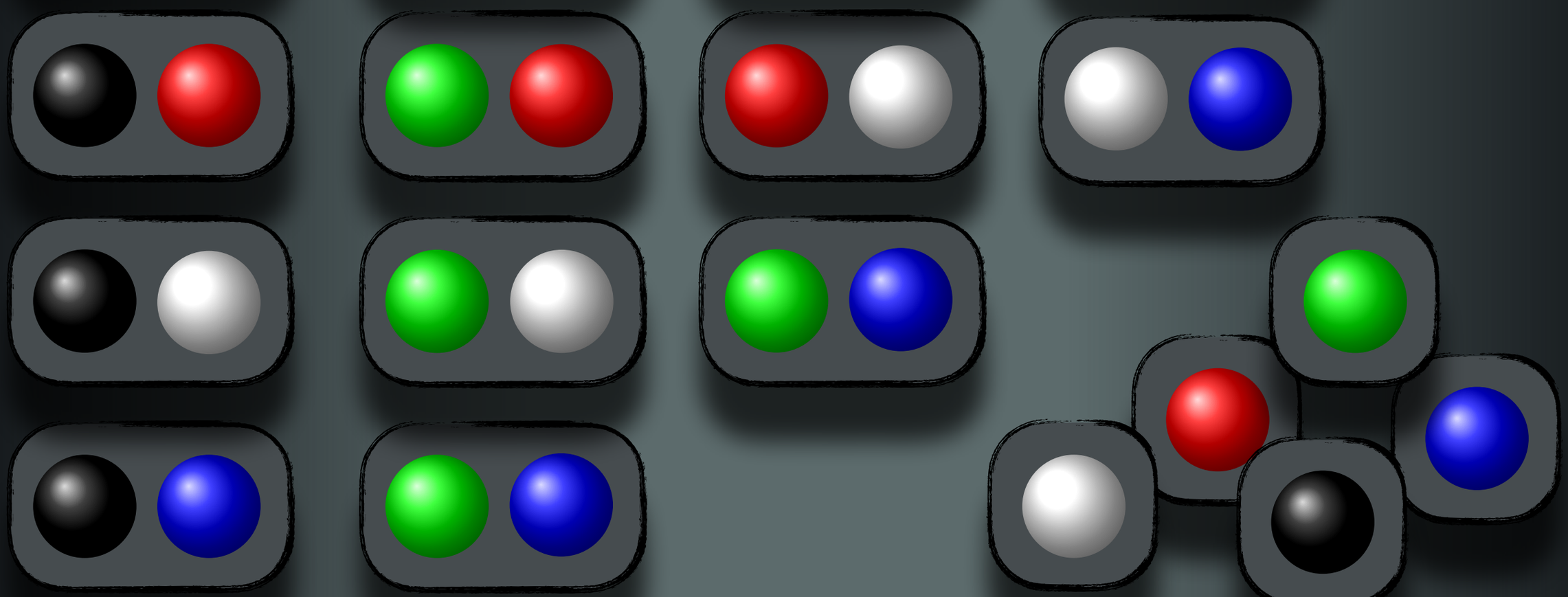


$$\alpha_k(c) := \{v \in \mathbb{V}_k \mid v \sqsubseteq c\}$$

$\alpha_2(c)$

configurations

c



$$\alpha_k(c) := \{v \in \mathbb{V}_k \mid v \sqsubseteq c\}$$

$\alpha_2(c)$

configurations

c

Views

views

For each k : two procedures (in paraellel):

- **Bug detection:**
  - under-approximation
  - concrete domain (configurations of size k)
- **Verification:**
  - over-approximation
  - abstract domain

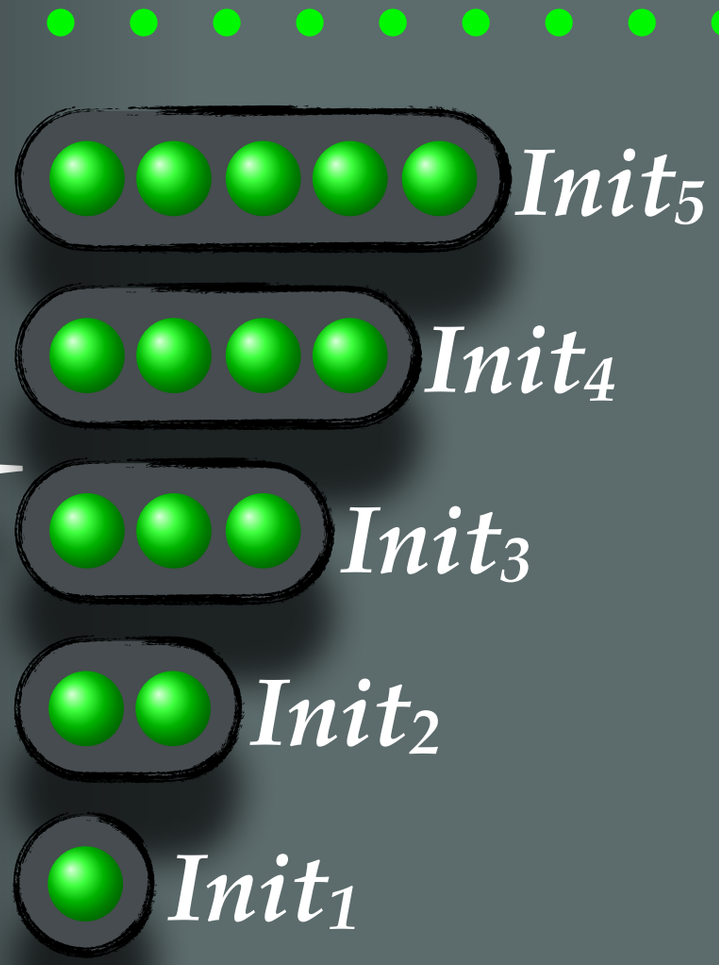
For each k : two procedures (in paraellel):

- **Bug detection:**
  - under-approximation
  - concrete domain (configurations of size k)
- **Verification:**
  - over-approximation
  - abstract domain
  - abstraction of initial configurations
  - abstract post operator

For each **k** : two procedures (in paraellel):

- **Bug detection**:
  - under-approximation
  - concrete domain (configurations of size **k**)
- **Verification**:
  - over-approximation
  - abstract domain

- abstraction of initial configurations
- abstract post operator



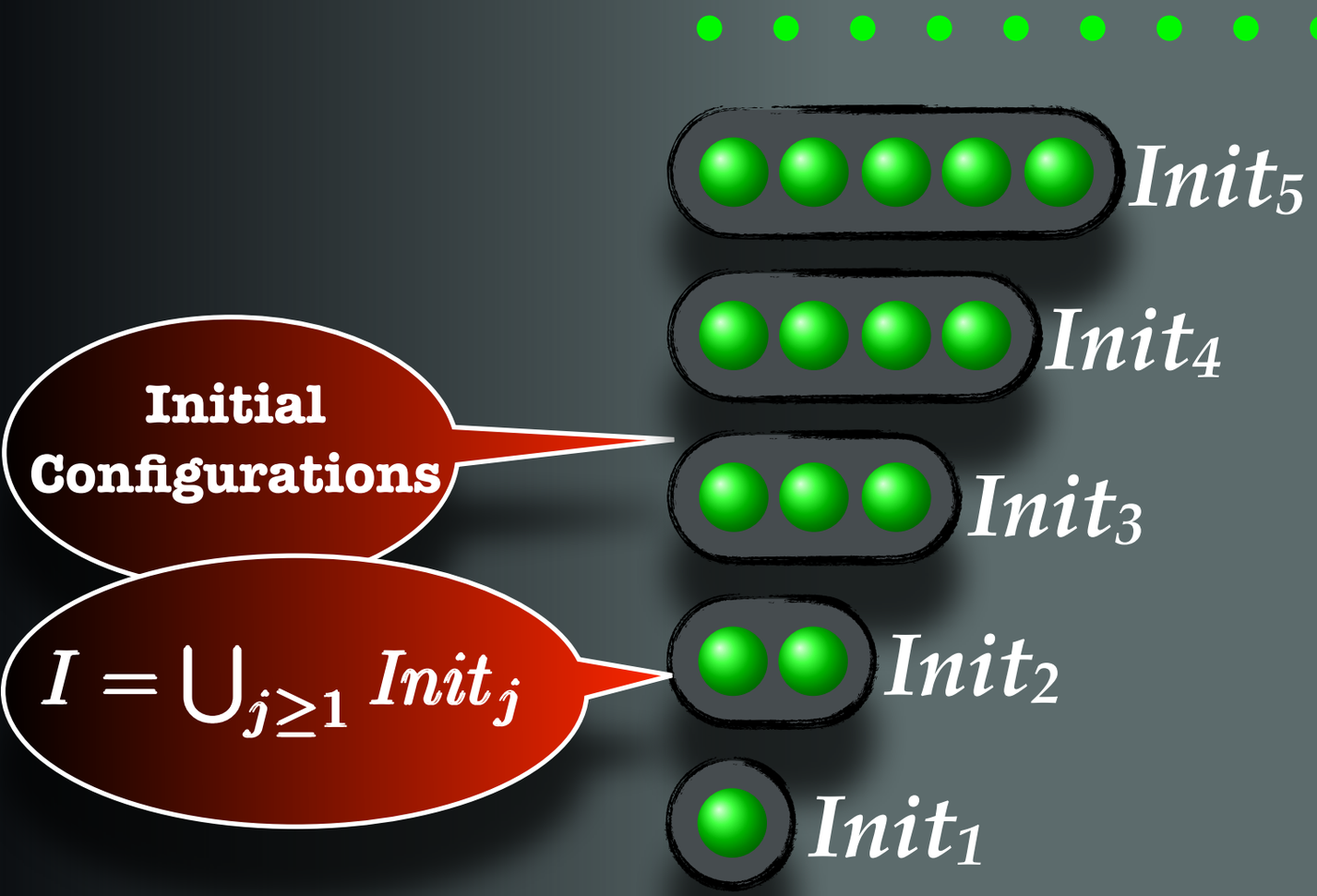
Initial Configurations

Initial configuration:

- **all processes in initial states**

Set of initial configurations:

- **infinite, but**
- **regular**

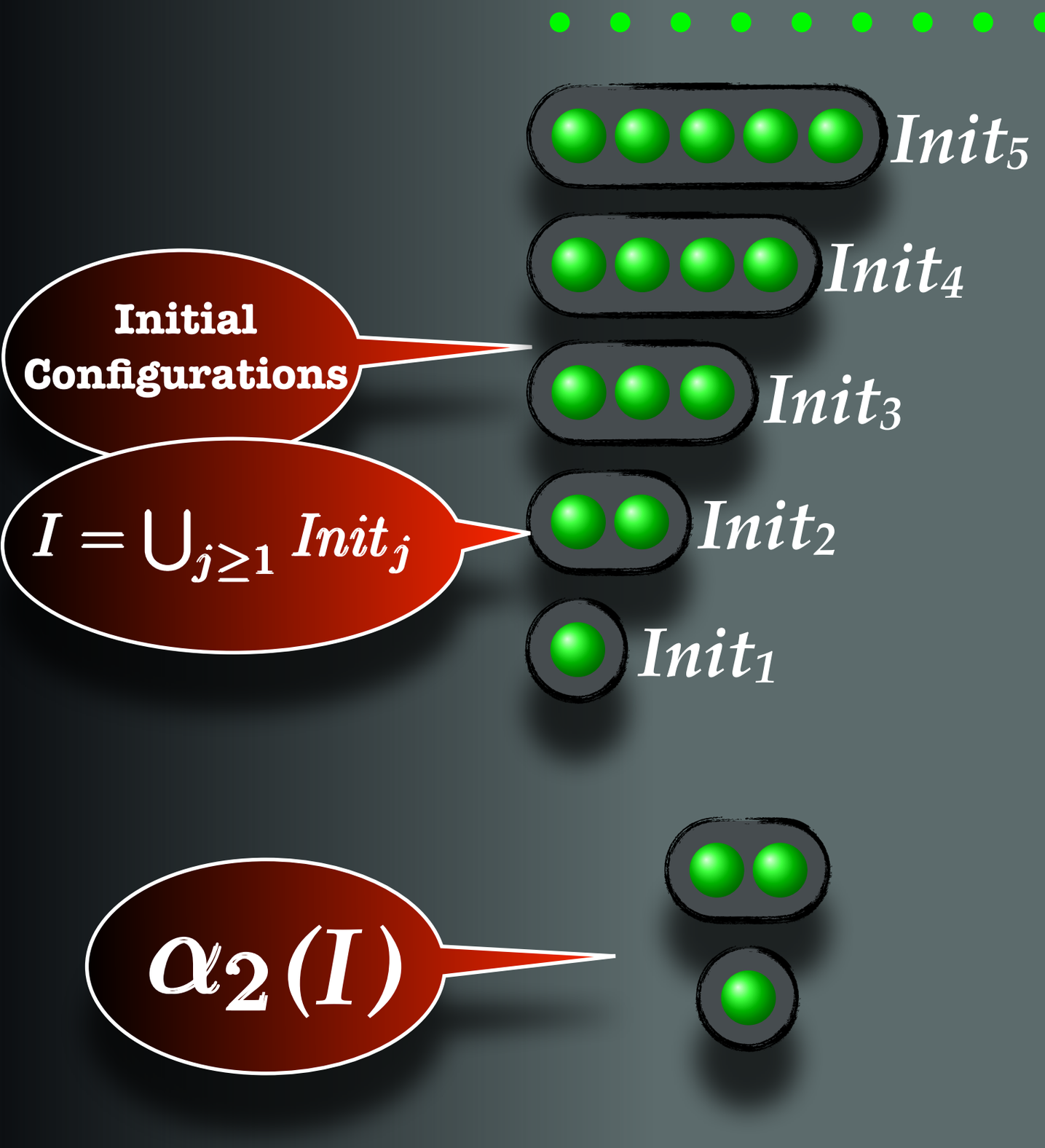


Initial configuration:

- **all processes in initial states**

Set of initial configurations:

- **infinite, but**
- **regular**



Initial configuration:

- **all processes in initial states**

Set of initial configurations:

- **infinite, but**
- **regular**

For each k : two procedures (in paraellel):

- **Bug detection:**
  - under-approximation
  - concrete domain
- **Verification:**
  - over-approximation
  - abstract domain

  - abstraction of initial configurations ✓
  - abstract post operator

For each k : two procedures (in paraellel):

- **Bug detection:**
  - under-approximation
  - concrete domain
- **Verification:**
  - over-approximation
  - abstract domain

- abstraction of initial configurations ✓
- abstract post operator

abstract post operator

Concrete

Confs

post

Confs

abstract post operator

Concrete

Abstract

Confs

post

Confs

Apost$_k$

Views

γ_k

Confs

Views

α_k

Confs

post

$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$

abstract post operator

Concrete

Abstract

Confs

post

Confs

$Apost_k$

Views

γ_k

Confs

post

Views

α_k

Confs

$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$

abstract post operator

Concrete

Abstract

Confs

post

Confs

Apost$_k$

Views

γ_k

Confs

Views

α_k

Confs

post

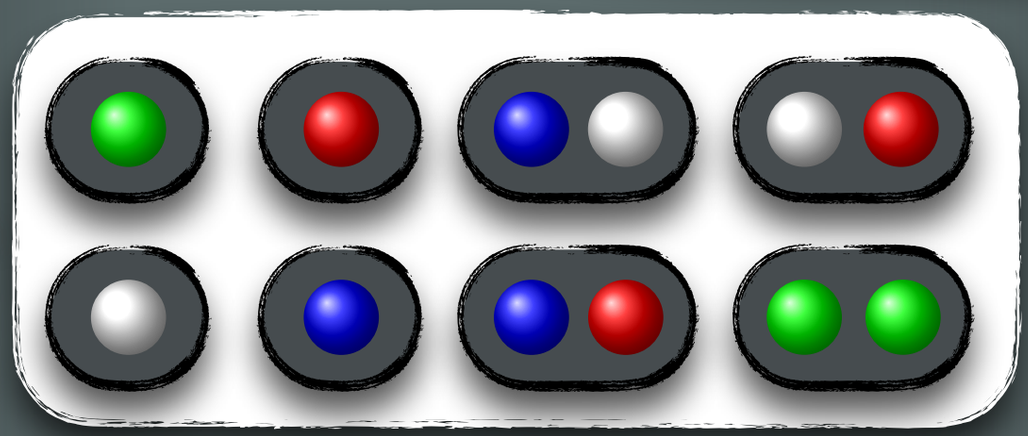
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

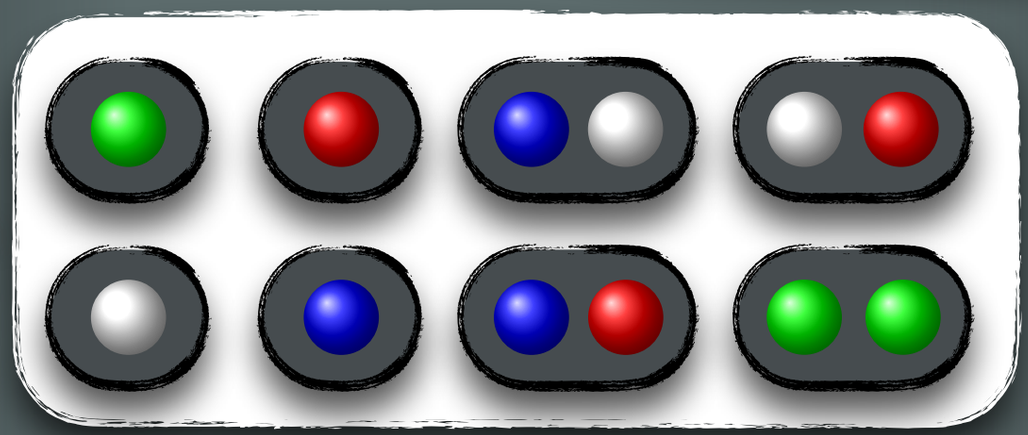
X



$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$

$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$

X

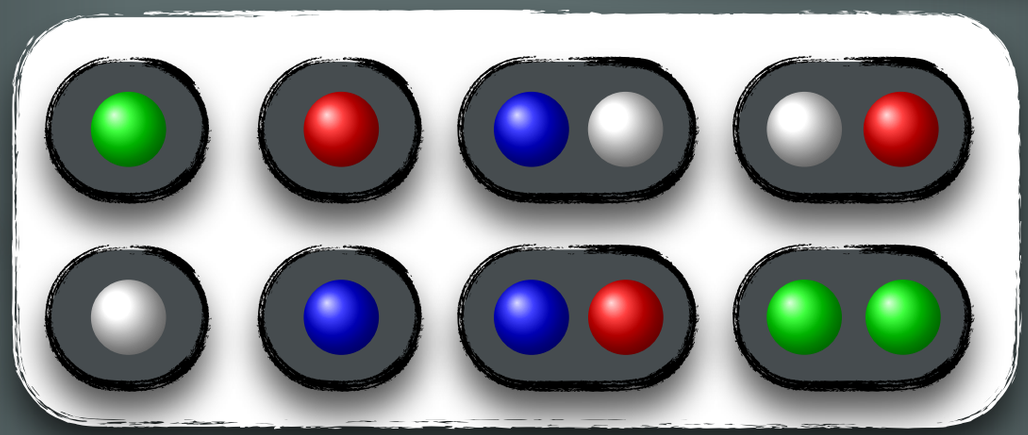


$\gamma_2(X)$

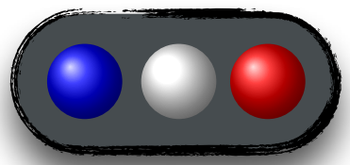
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X



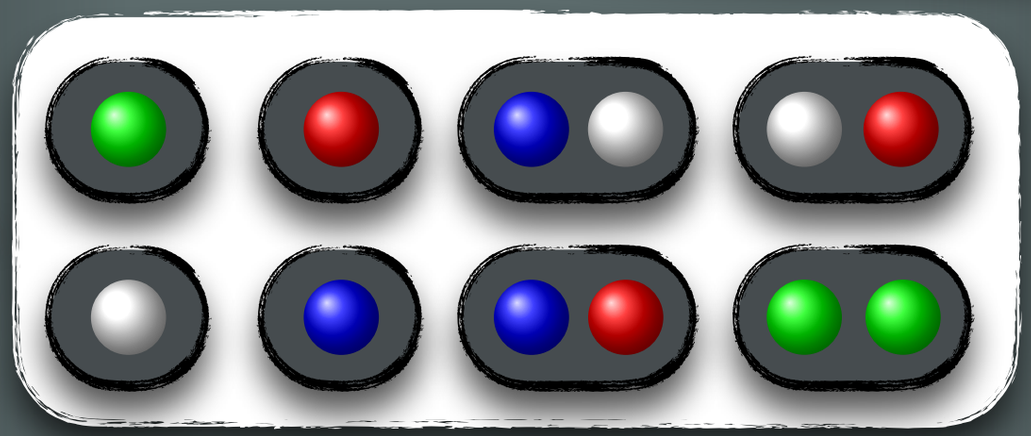
$\gamma_2(X)$



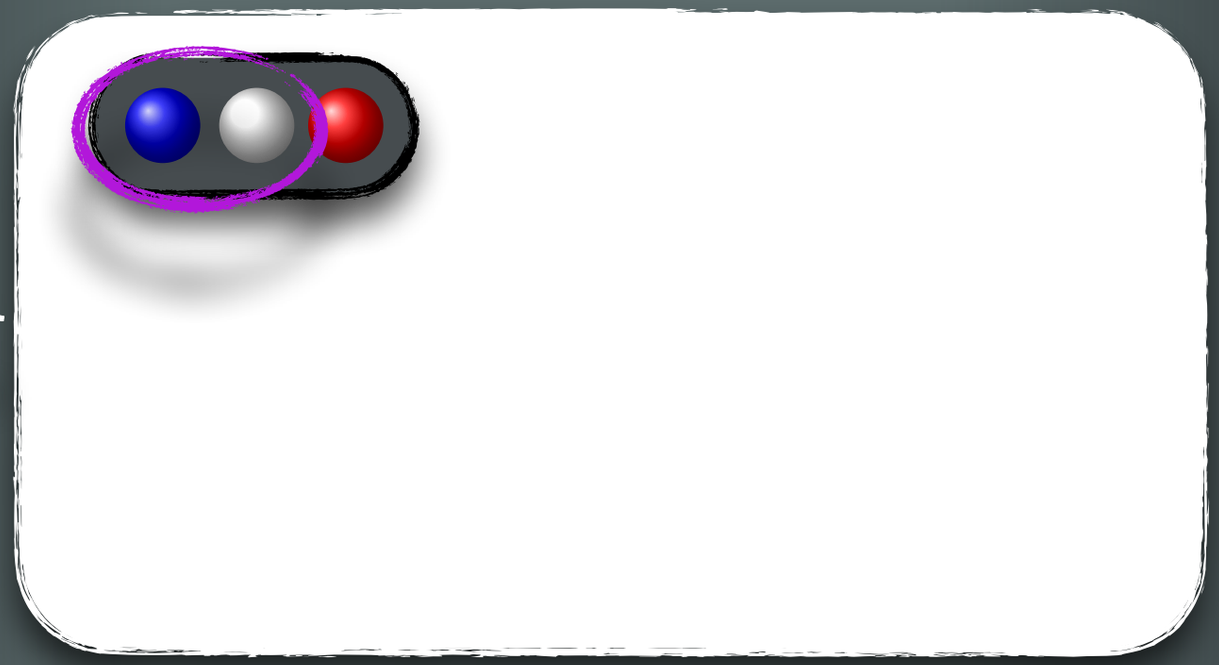
$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$

$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$

X



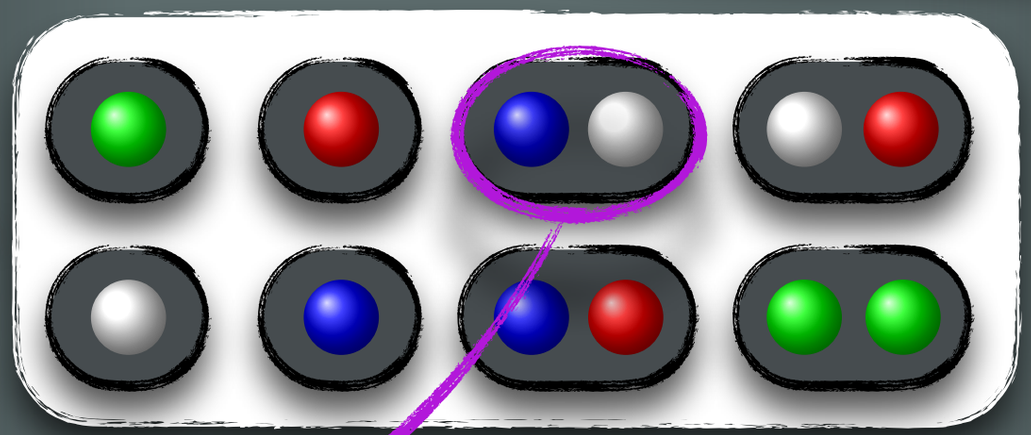
$\gamma_2(X)$

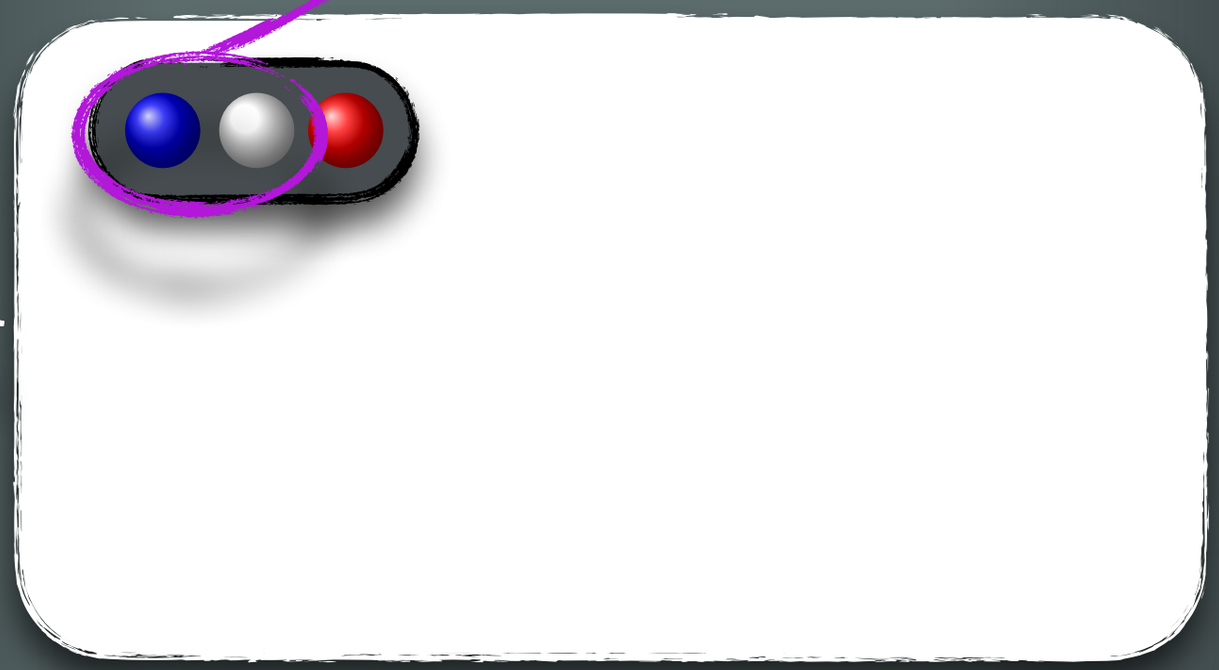


$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X



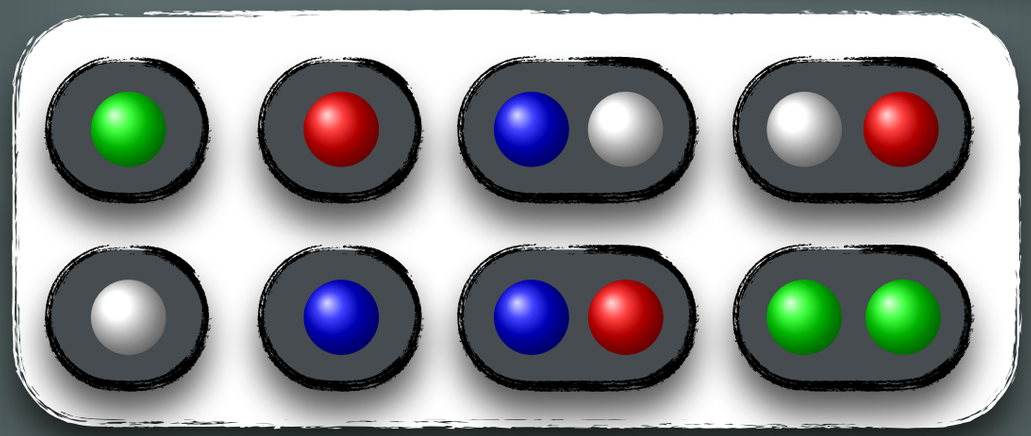


$\gamma_2(X)$

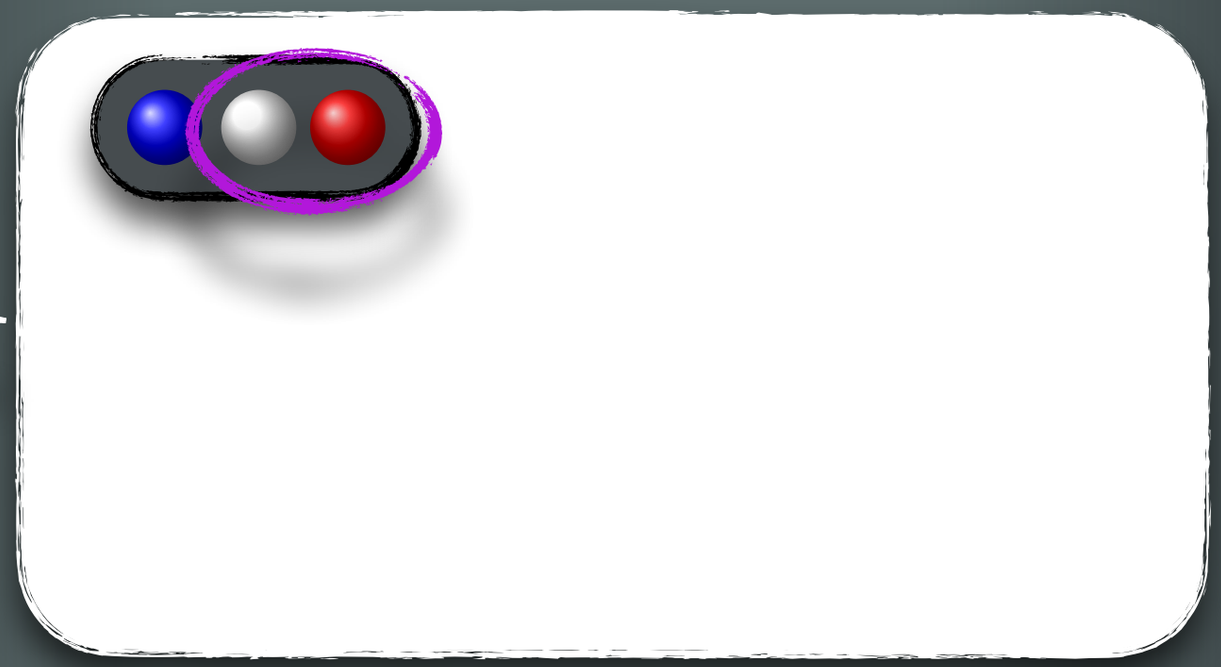
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X



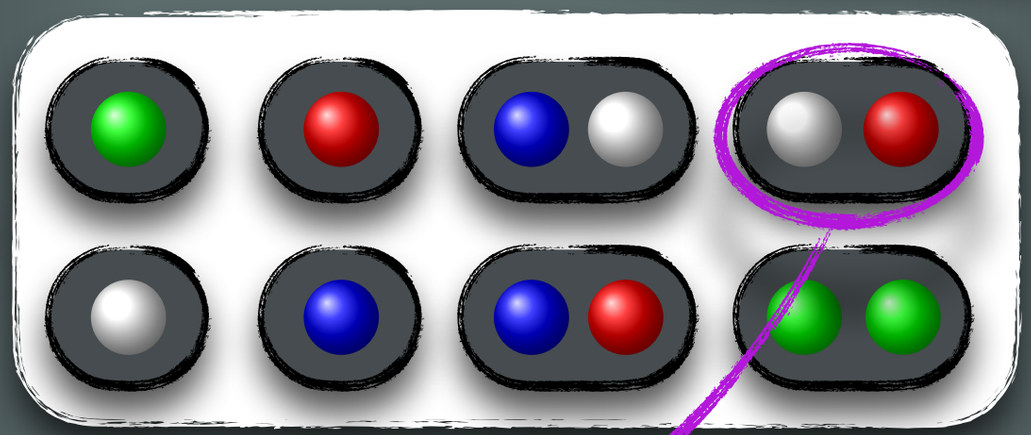
$\gamma_2(X)$



$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X

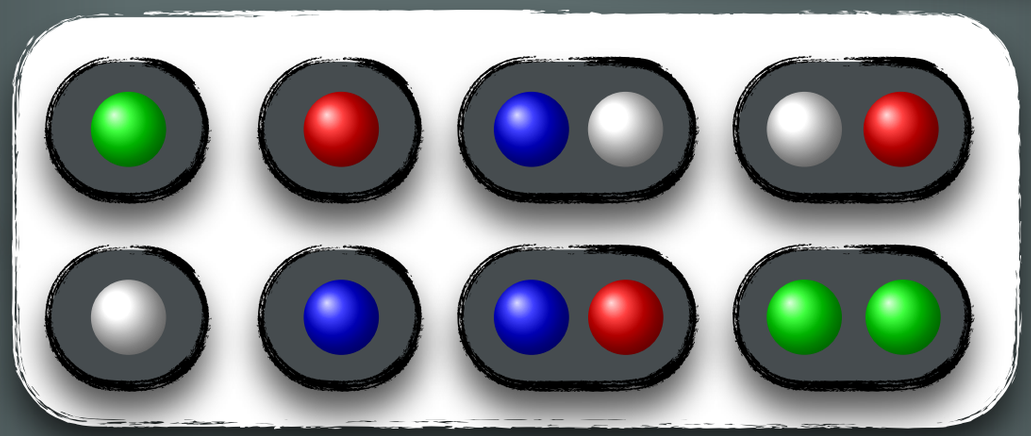


$\gamma_2(X)$

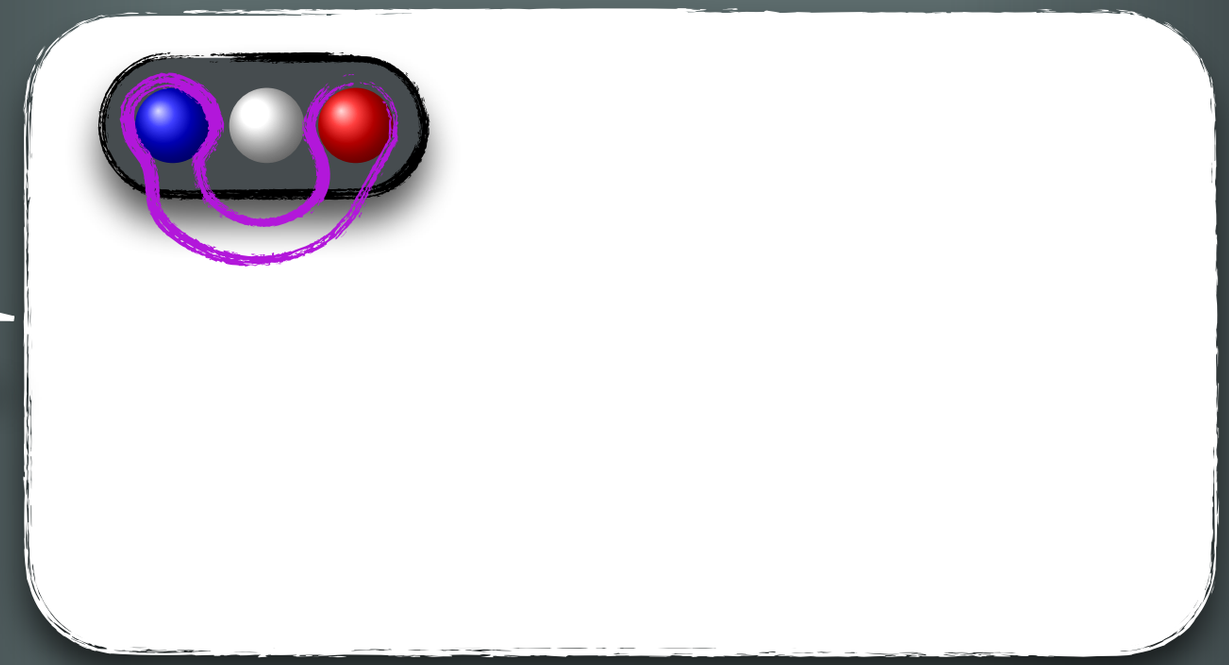
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X



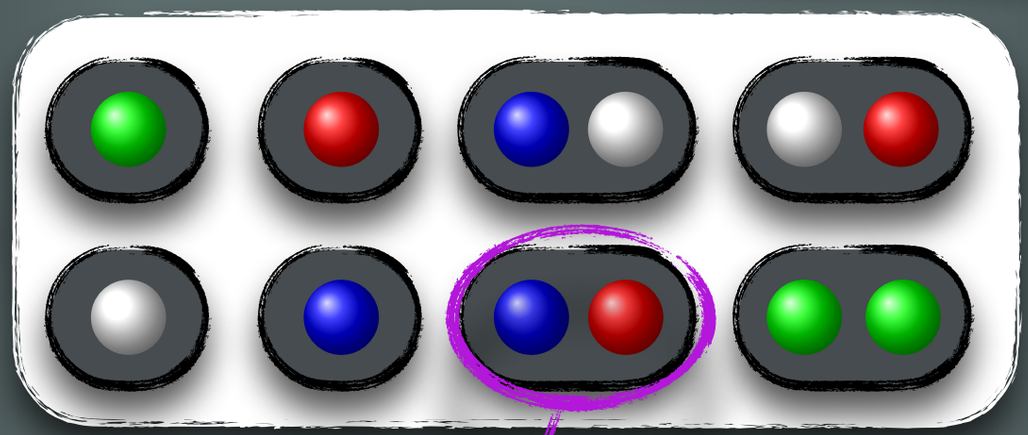
$\gamma_2(X)$

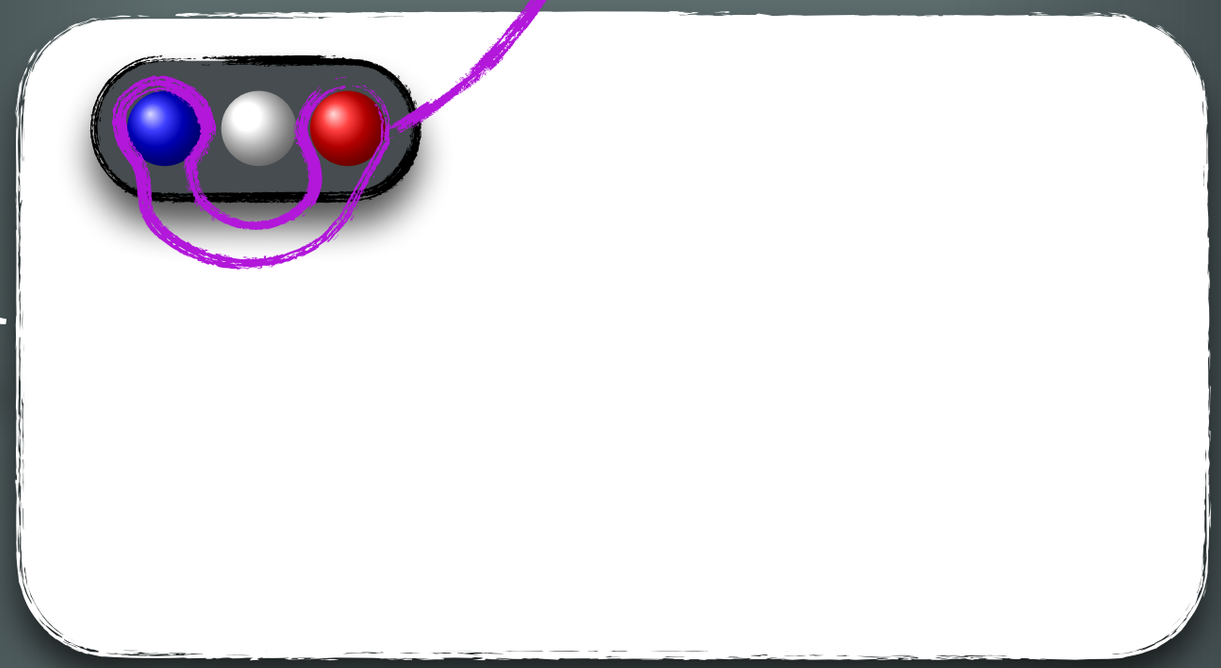


$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X



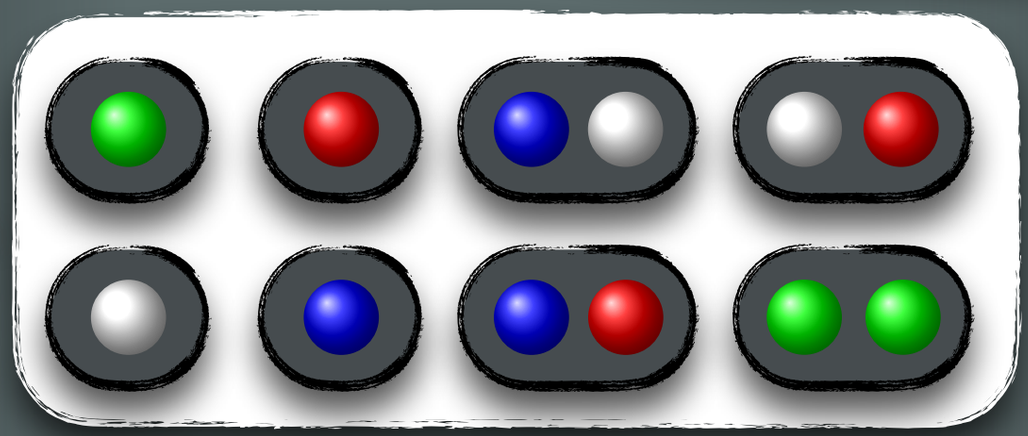


$\gamma_2(X)$

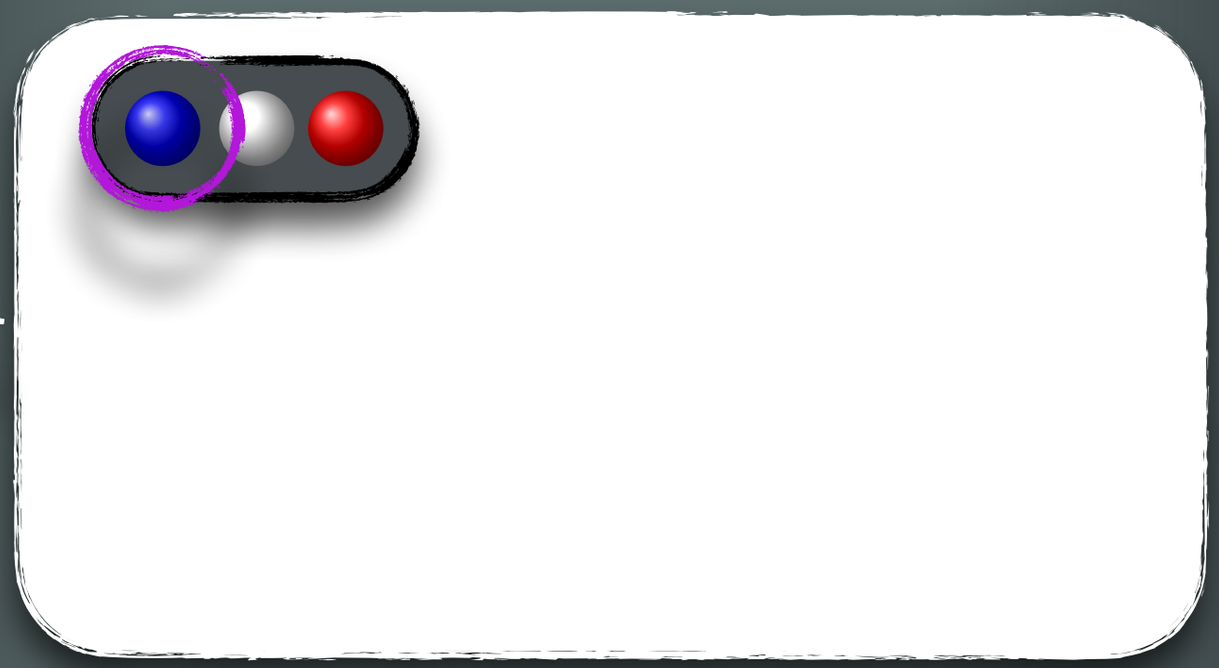
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X

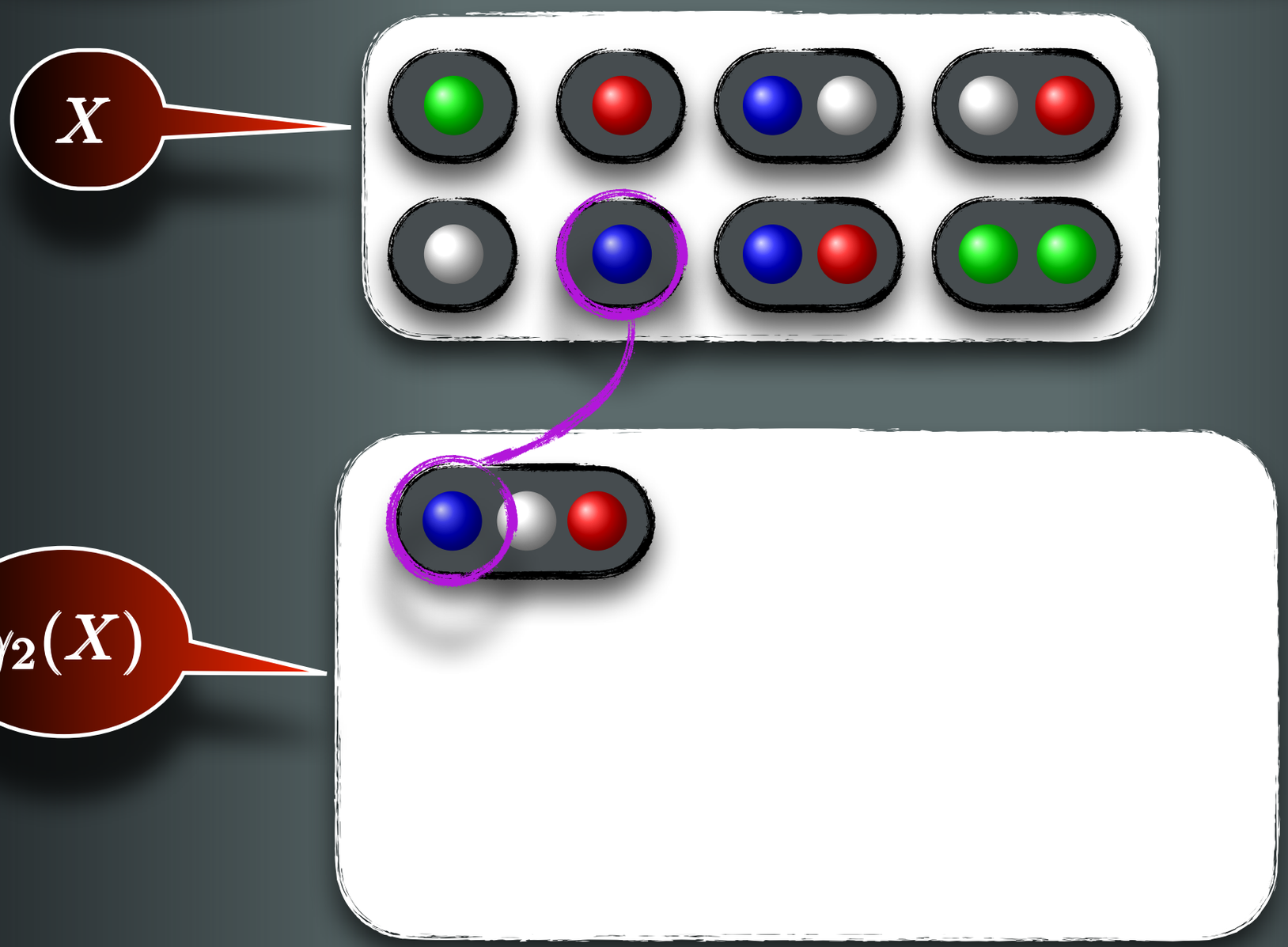


$\gamma_2(X)$



$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

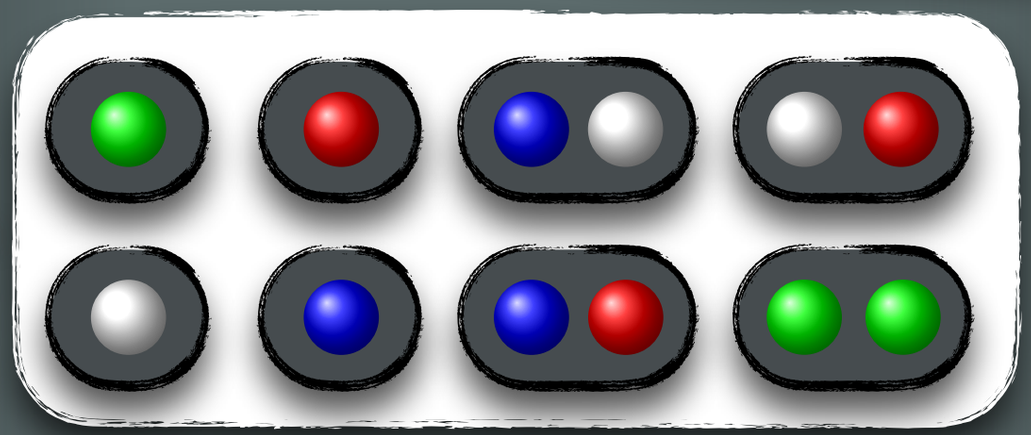
$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$



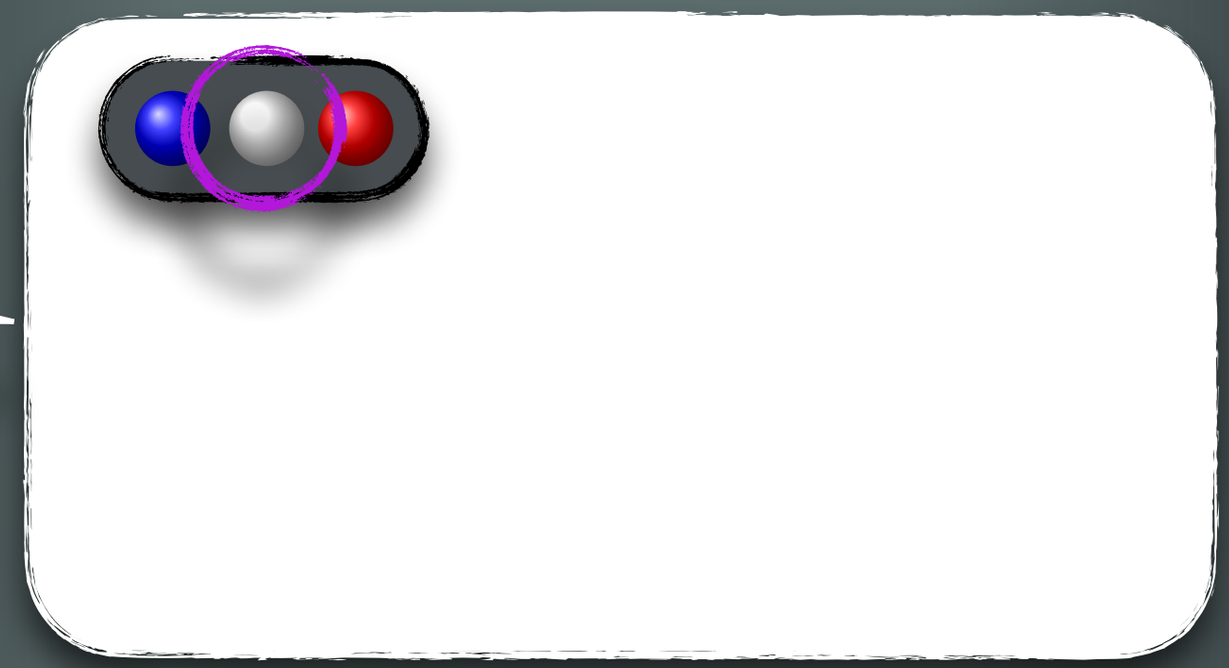
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X

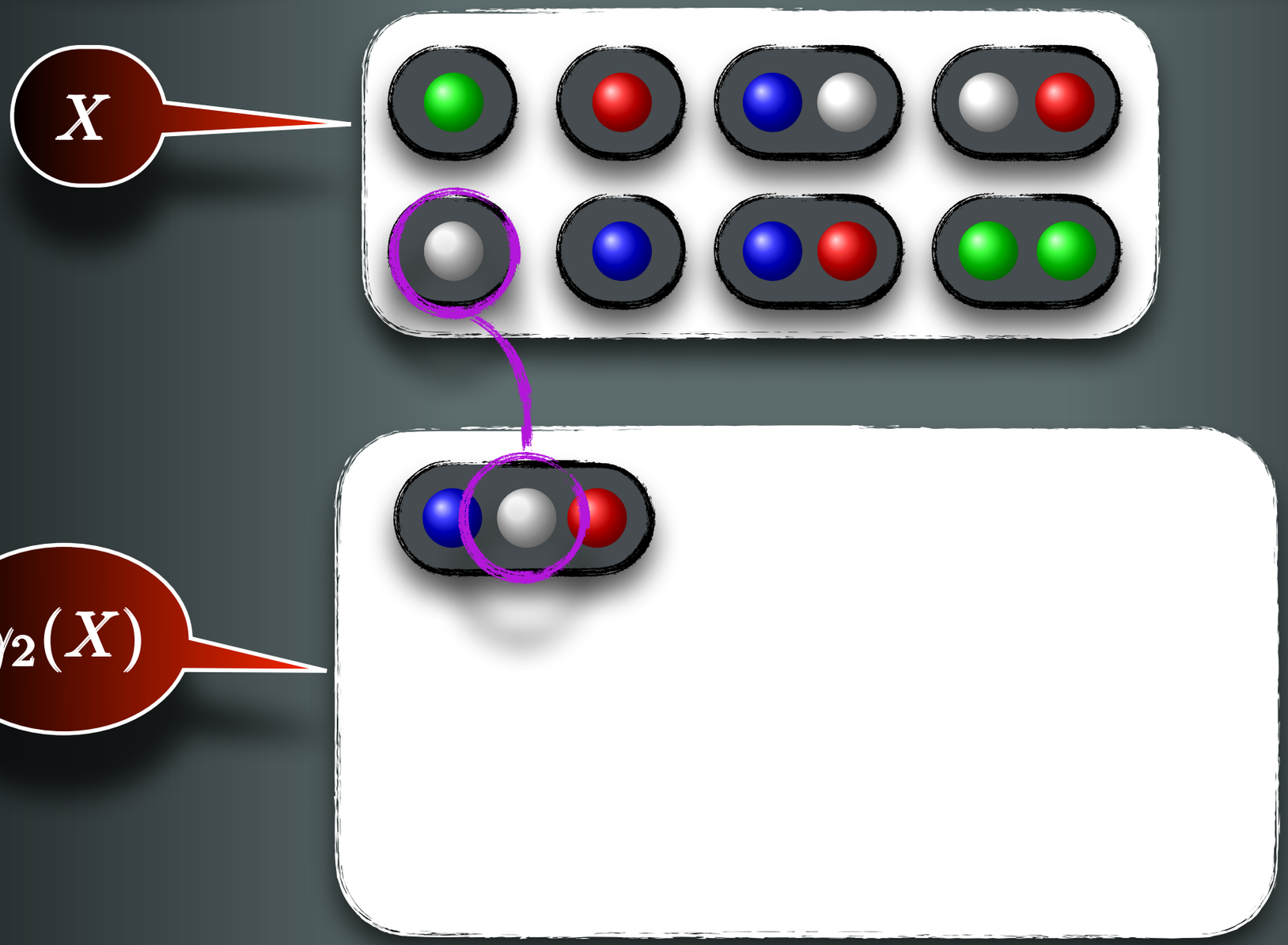


$\gamma_2(X)$



$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$

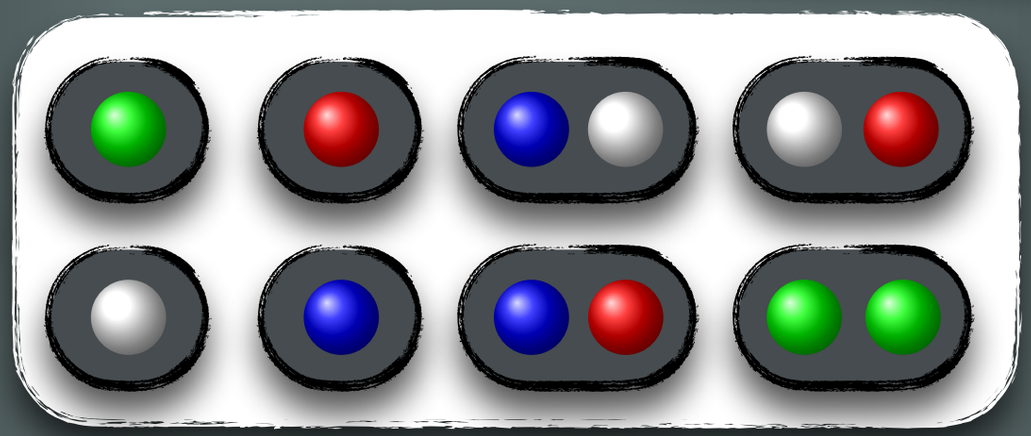
$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$



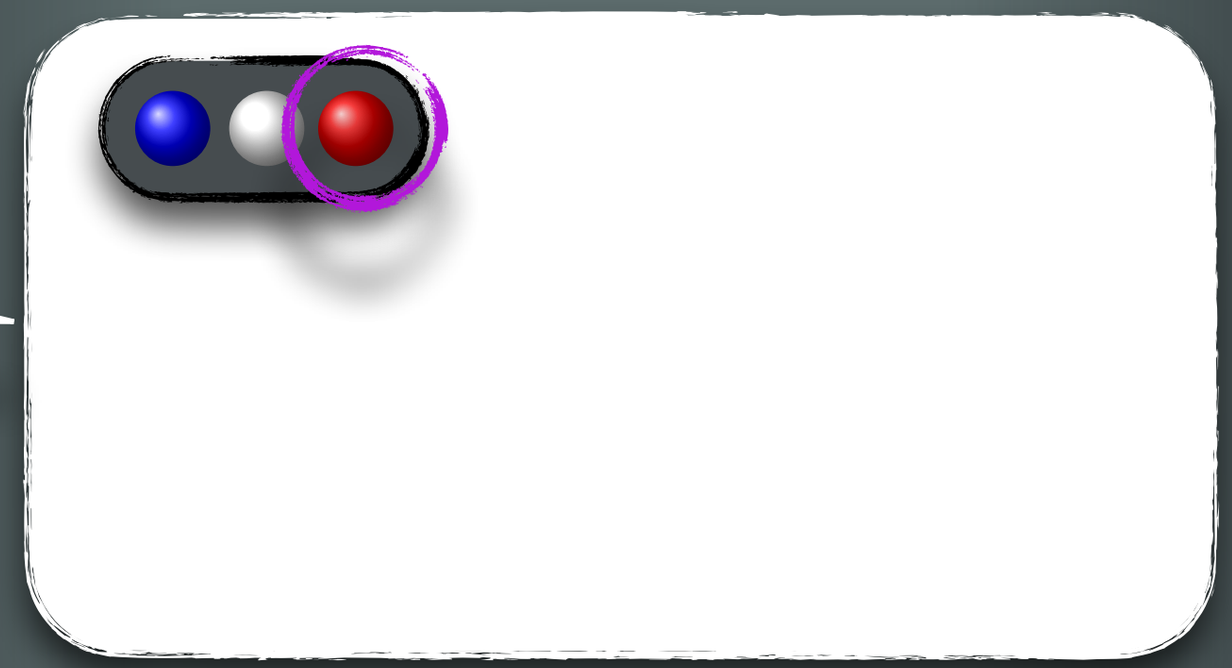
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X



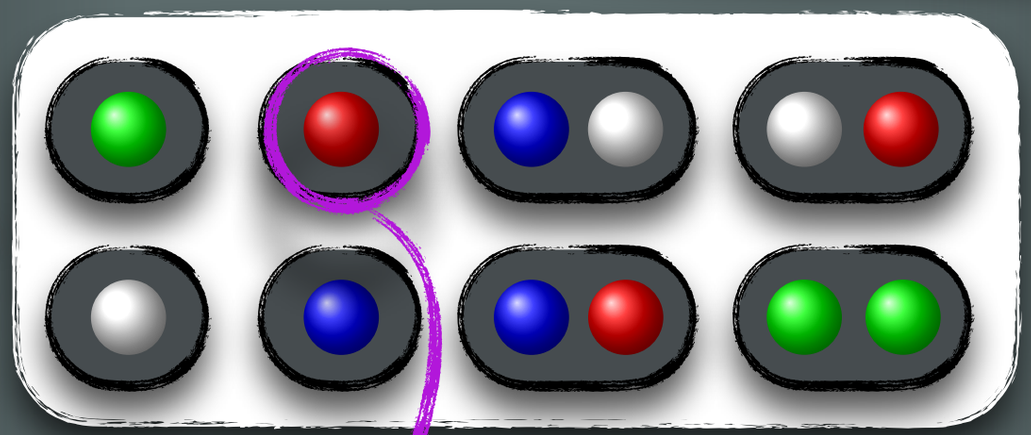
$\gamma_2(X)$

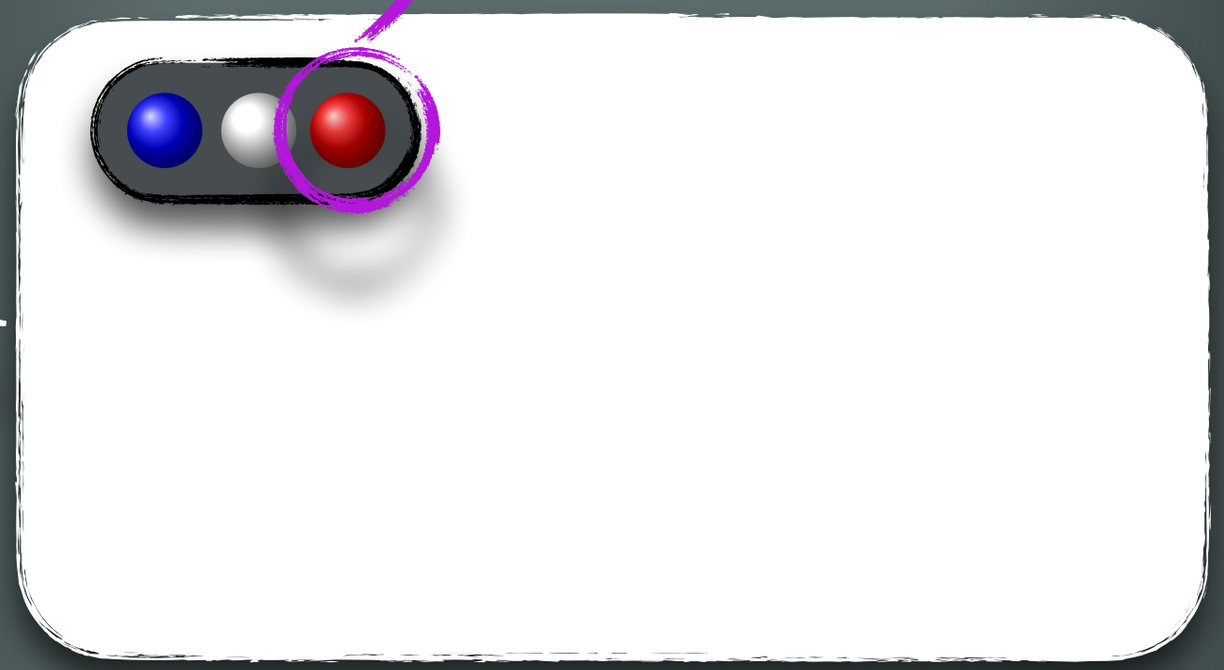


$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X



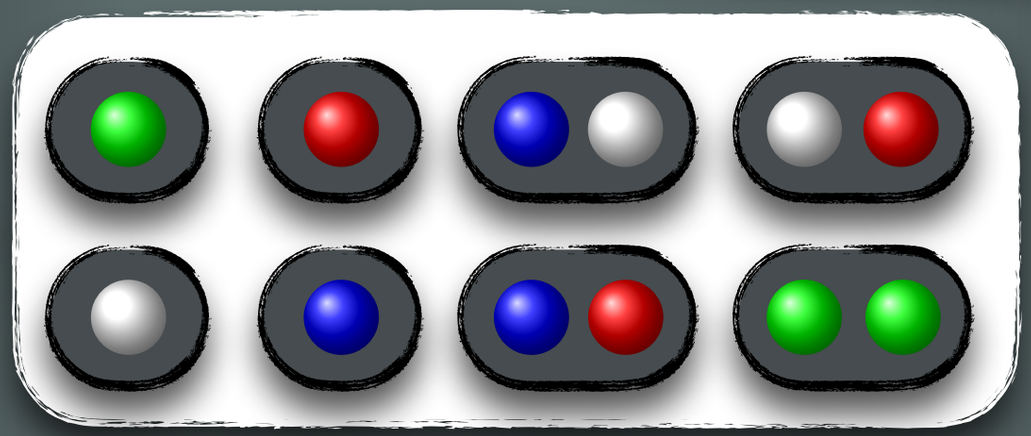


$\gamma_2(X)$

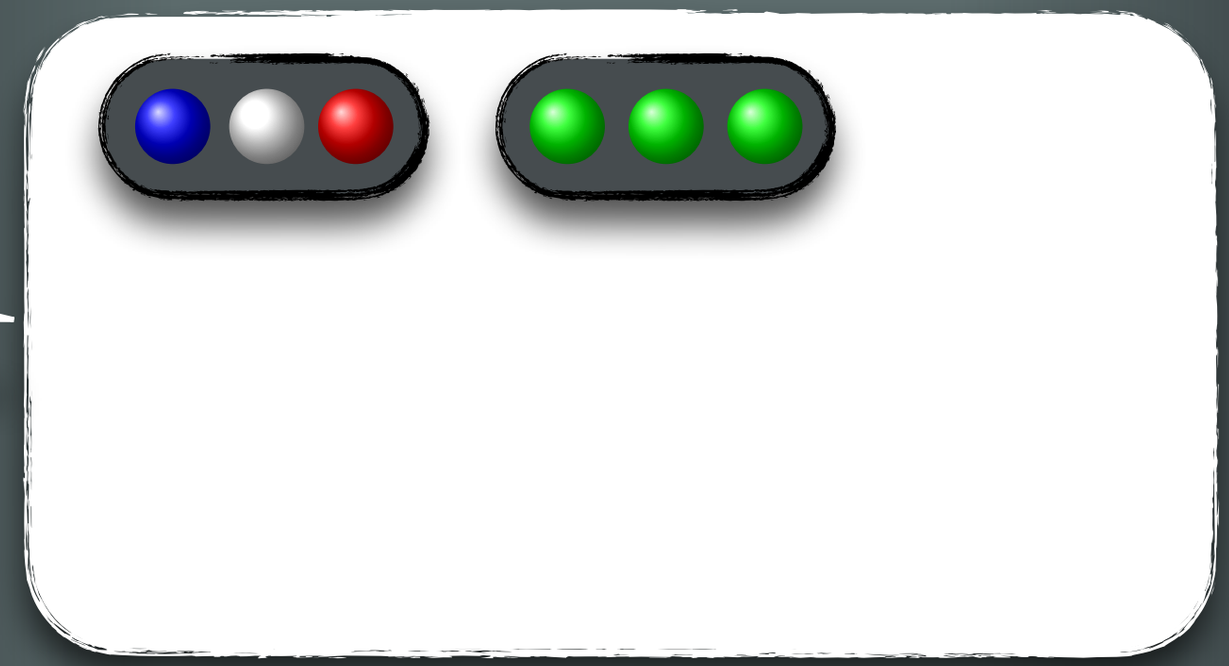
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X



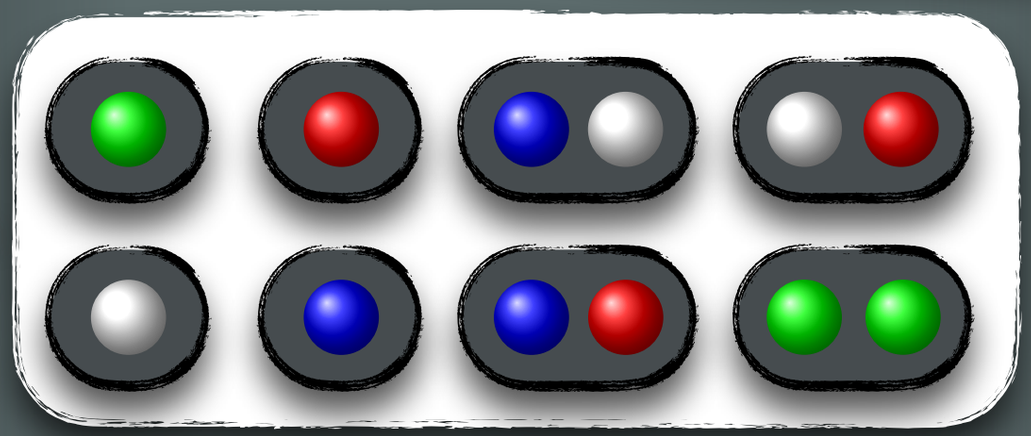
$\gamma_2(X)$



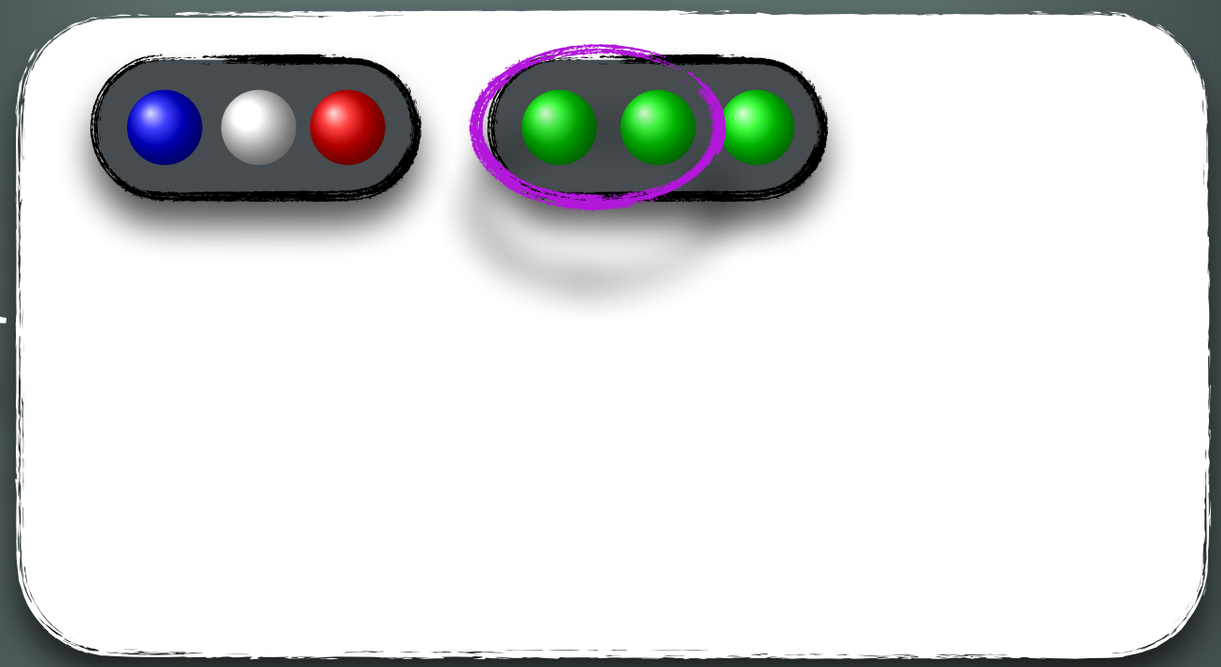
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X



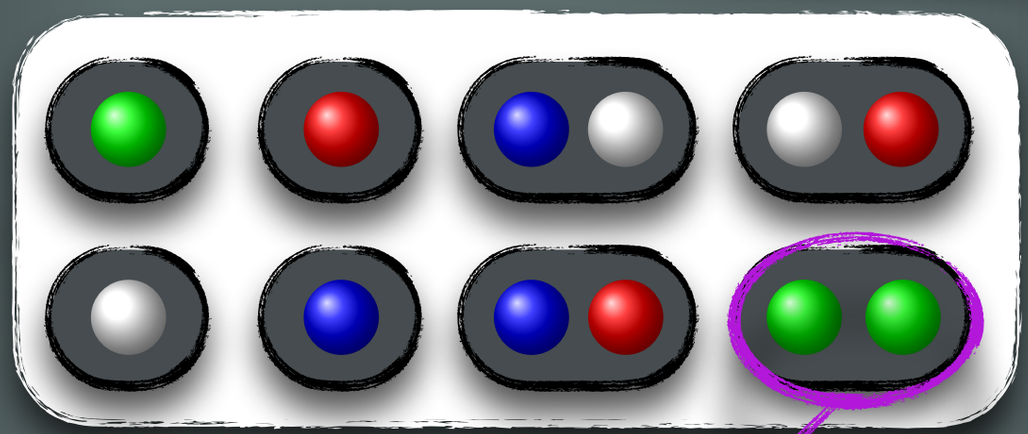
$\gamma_2(X)$

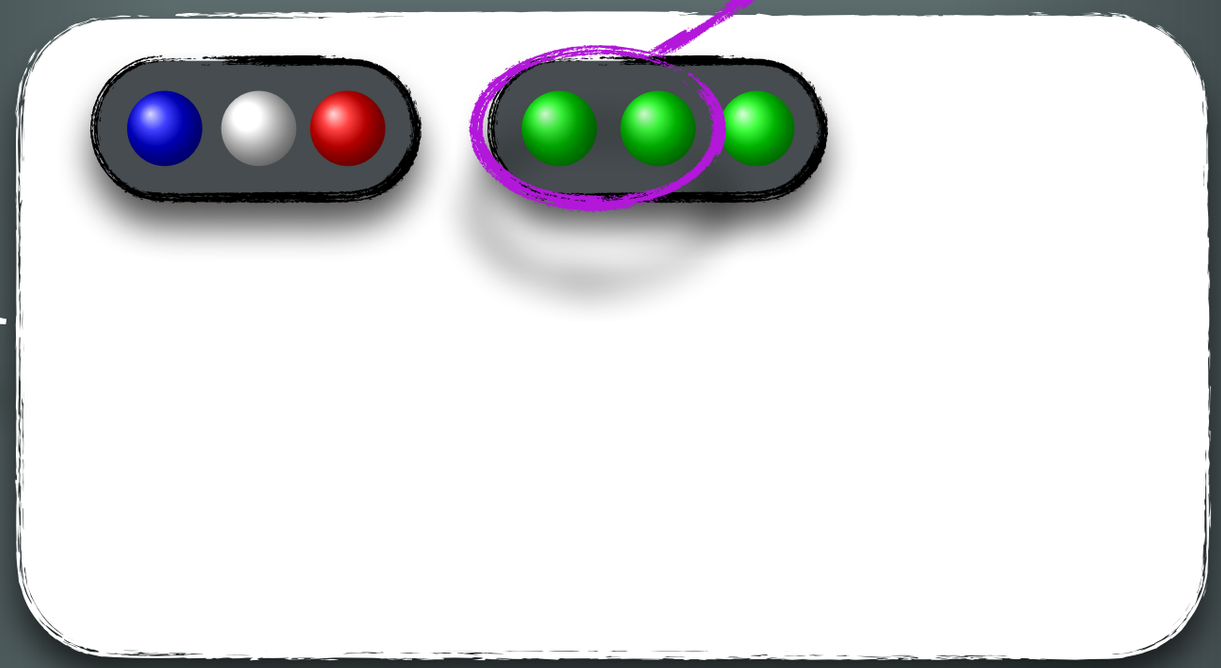


$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X



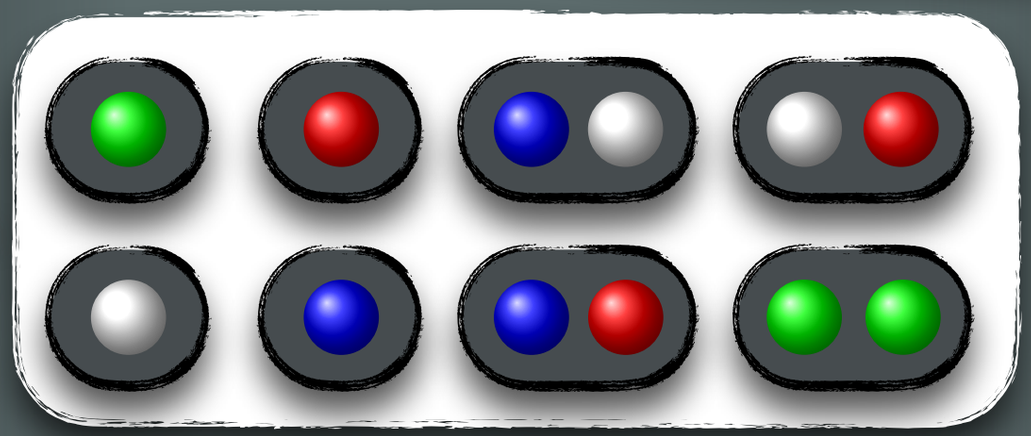


$\gamma_2(X)$

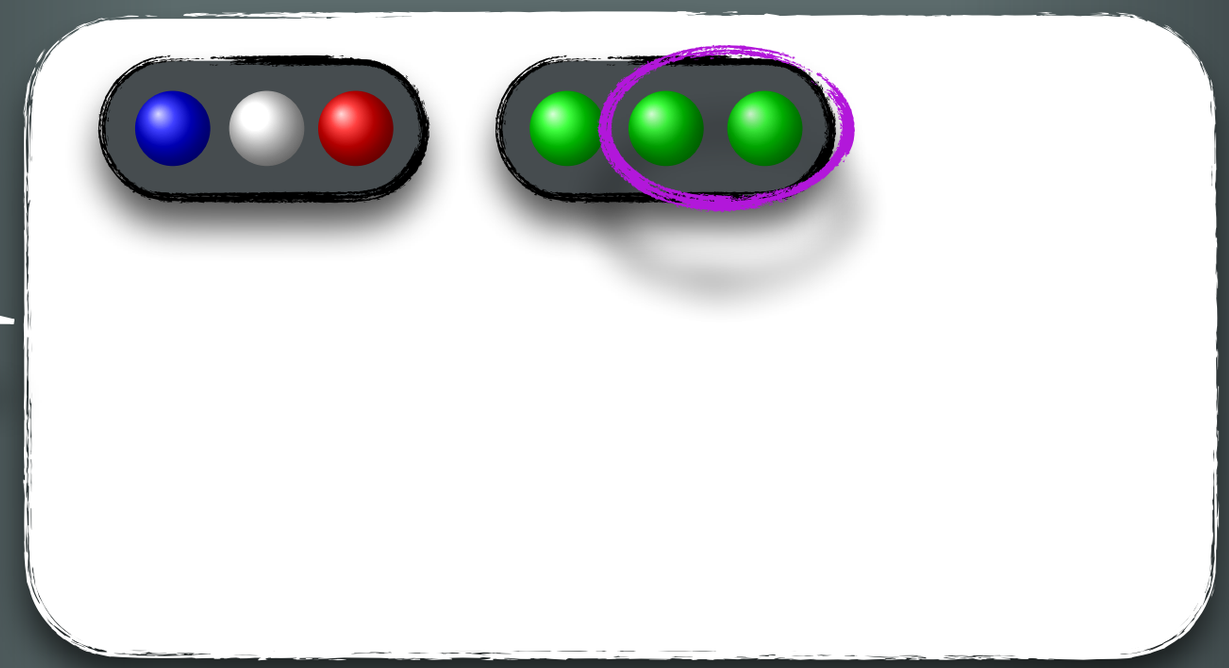
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X



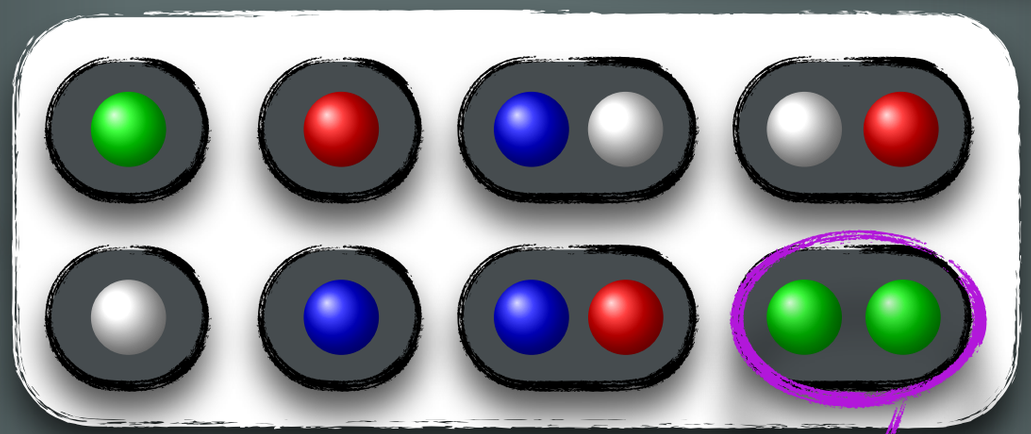
$\gamma_2(X)$

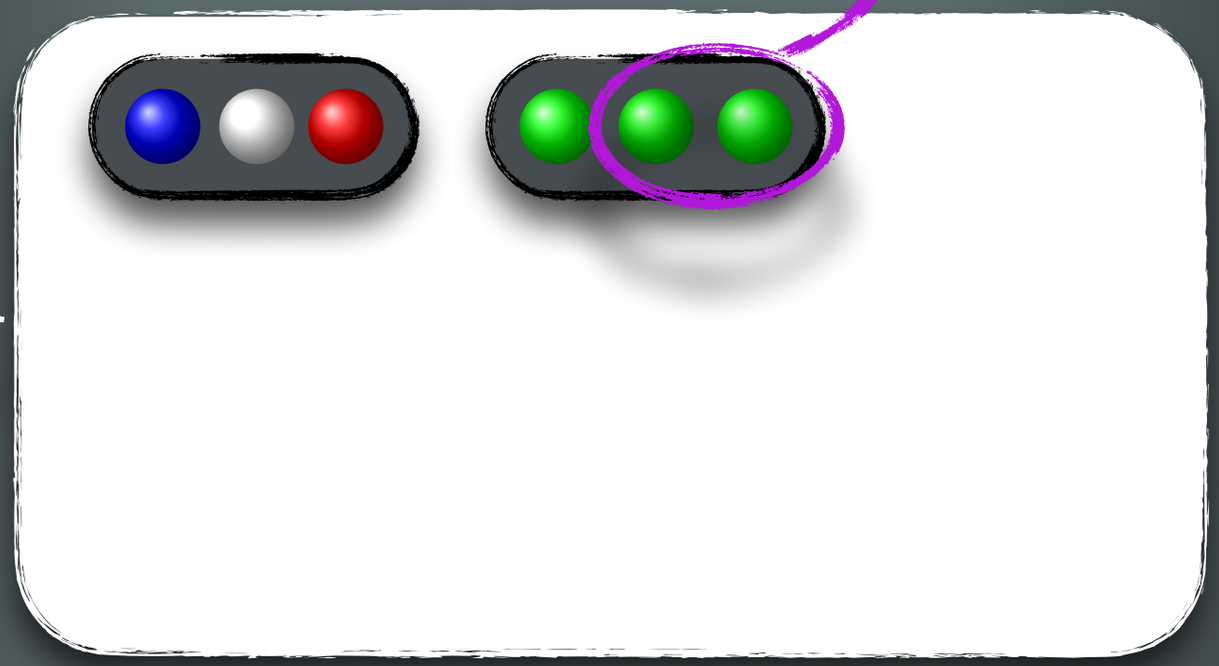


$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X



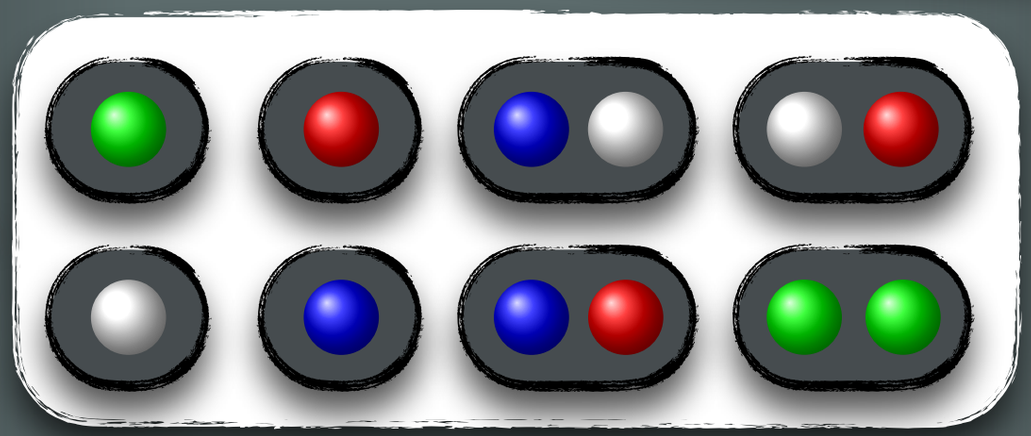


$\gamma_2(X)$

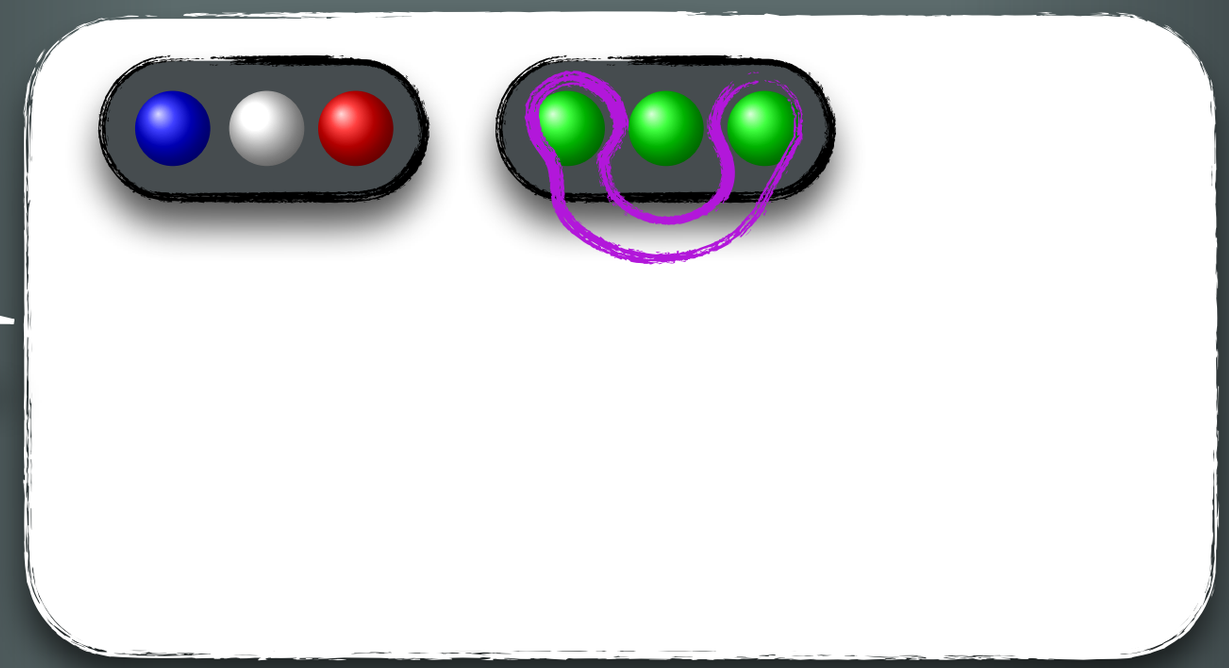
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X

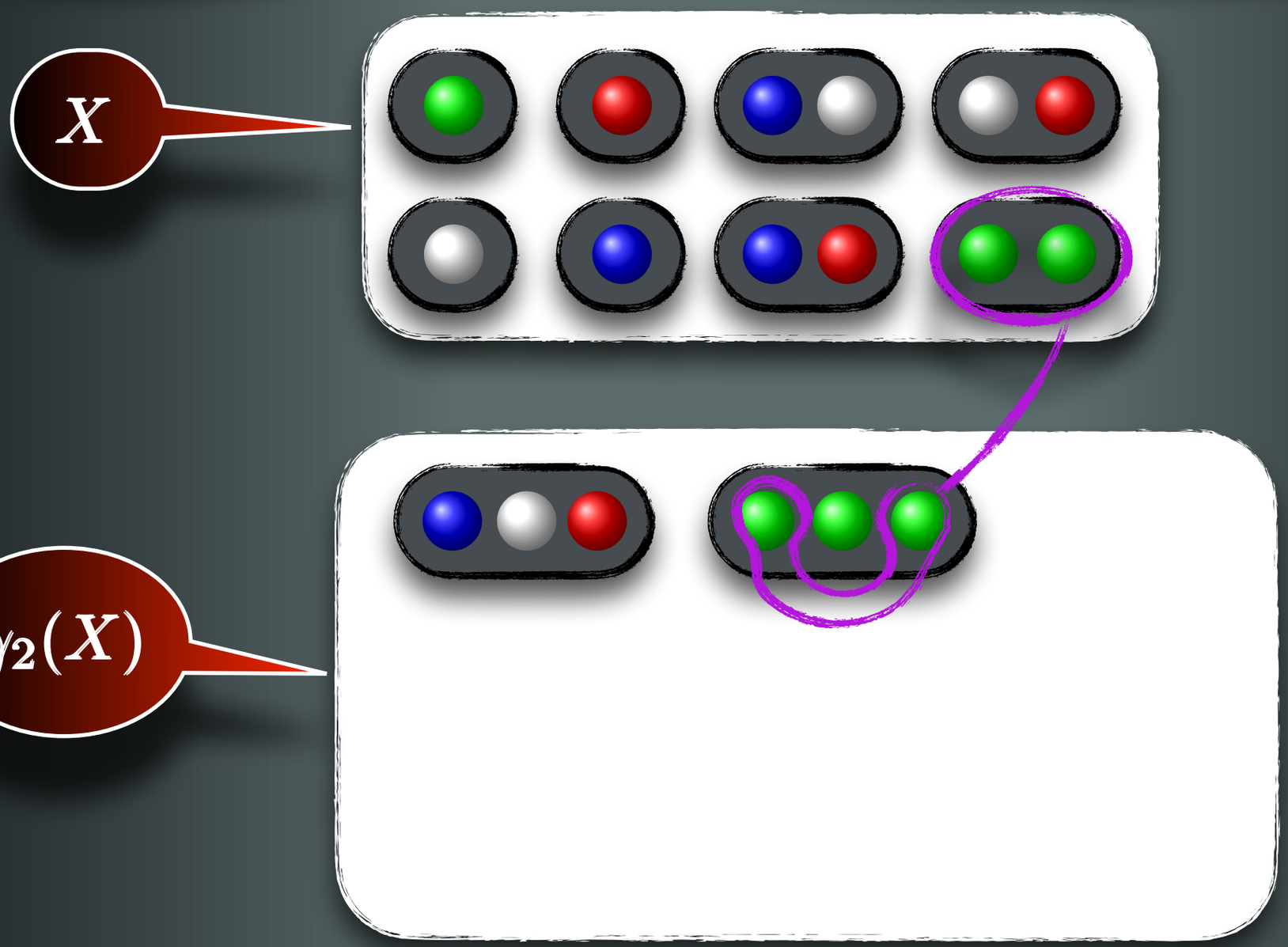


$\gamma_2(X)$



$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

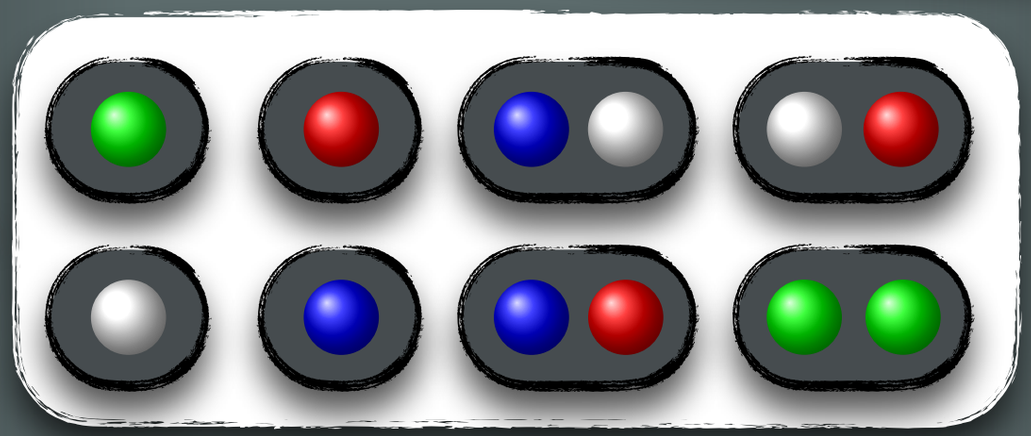
$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$



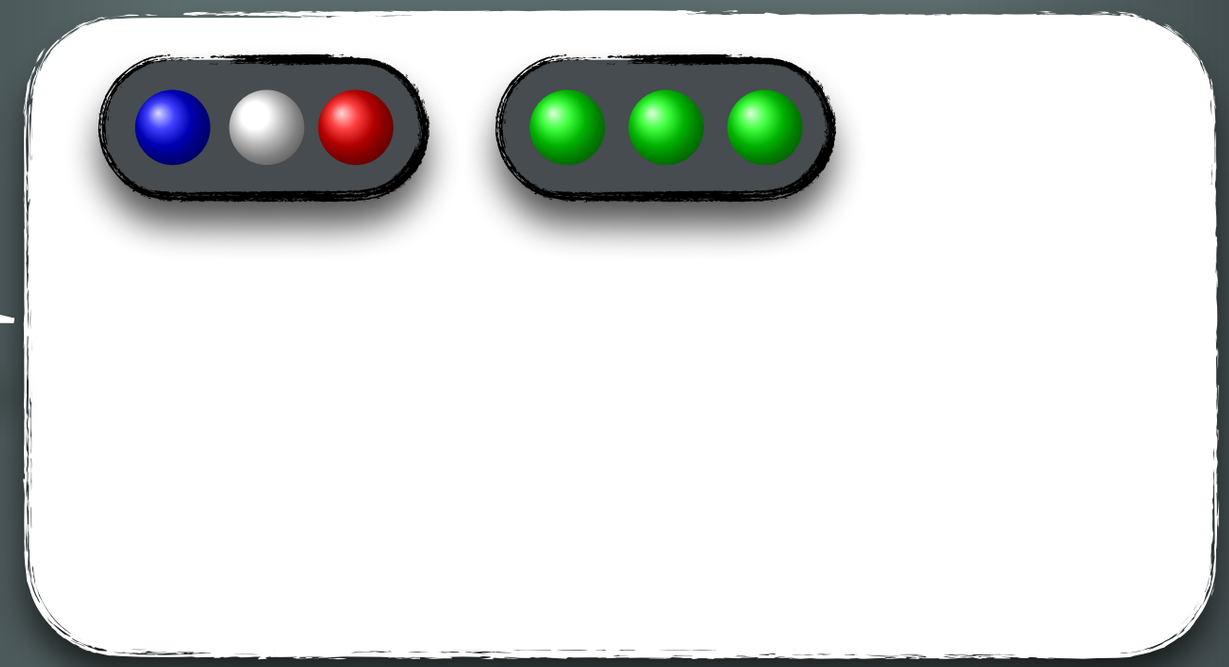
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X



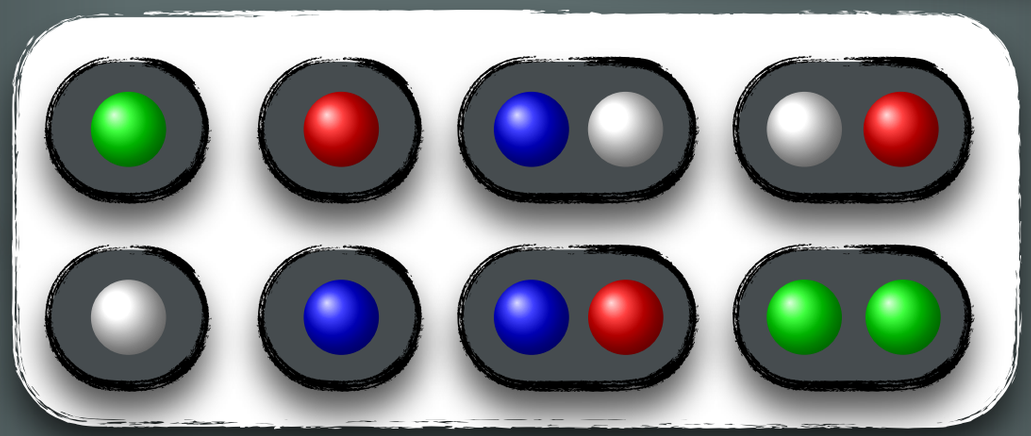
$\gamma_2(X)$



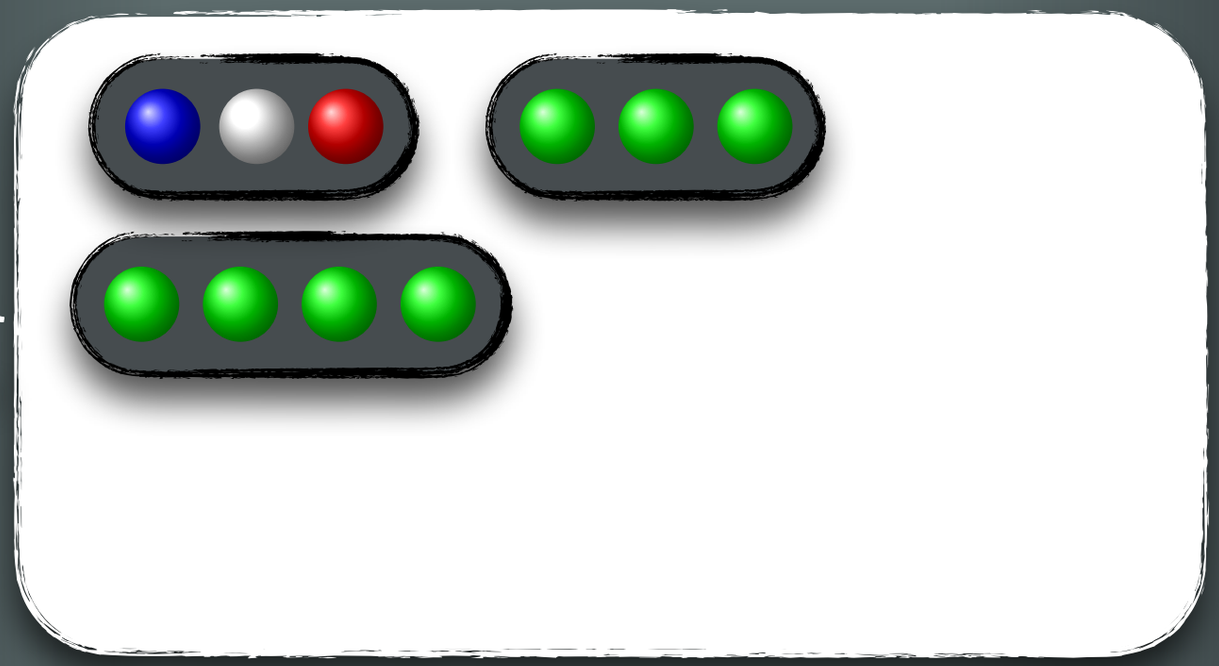
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X



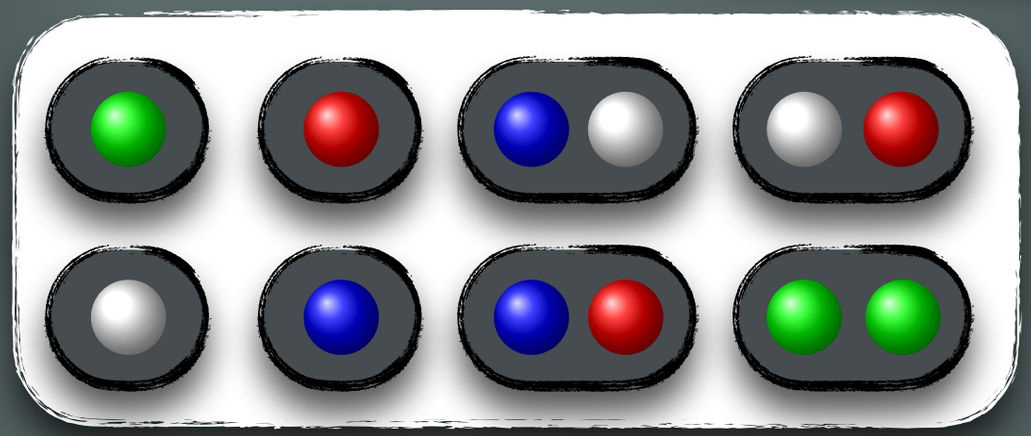
$\gamma_2(X)$



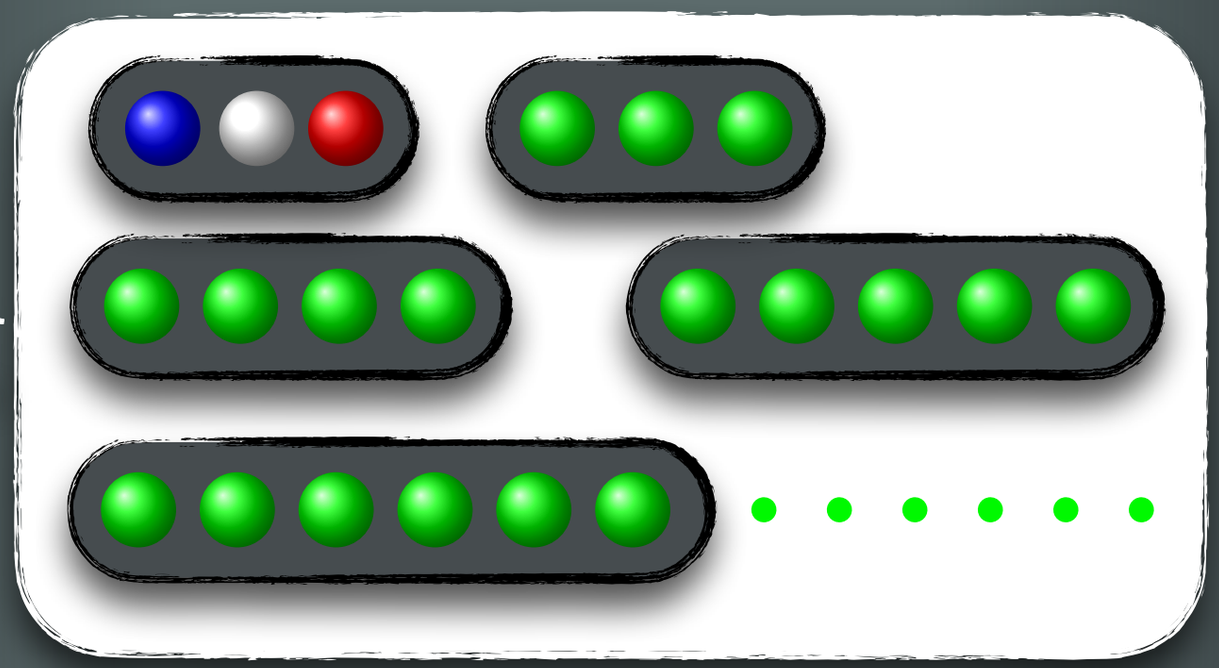
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X



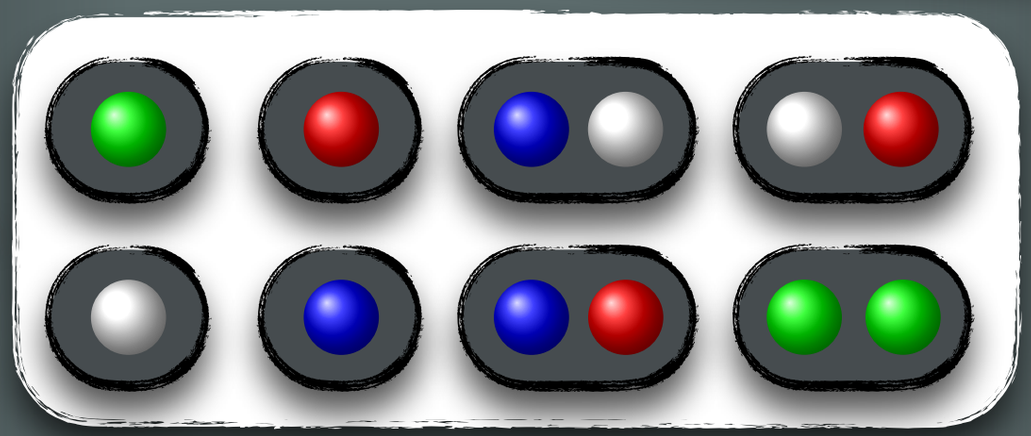
$\gamma_2(X)$



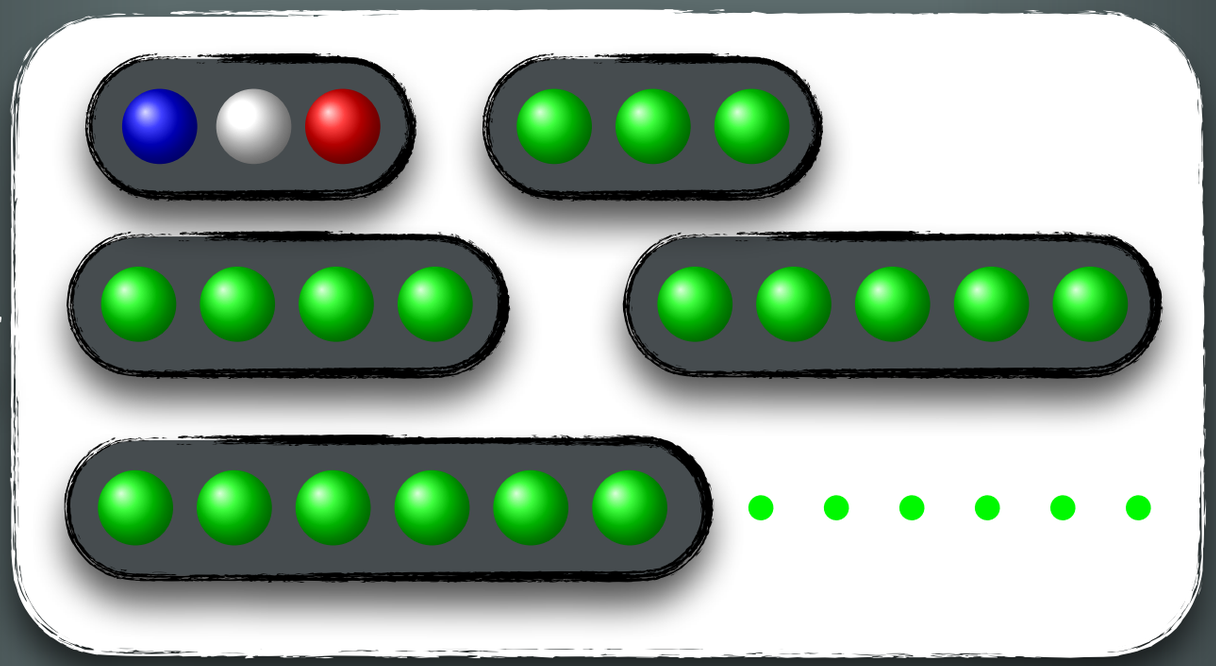
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X



$\gamma_2(X)$

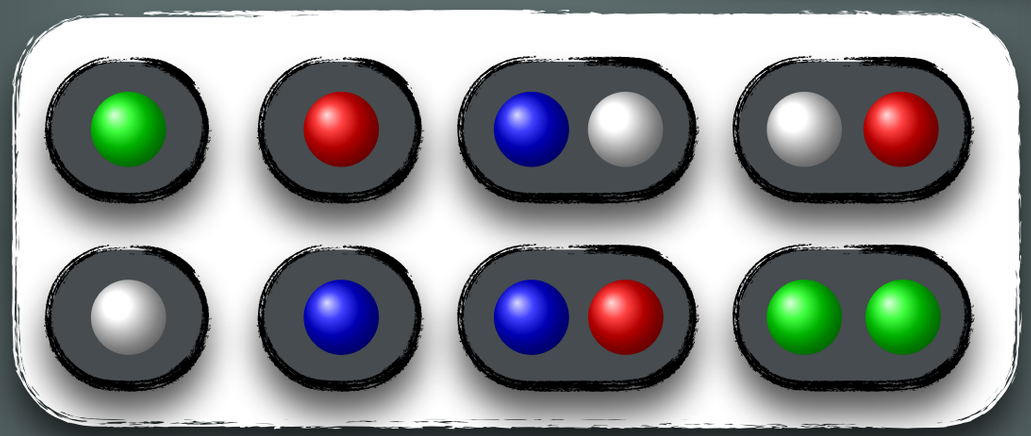


$+ X$

$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$

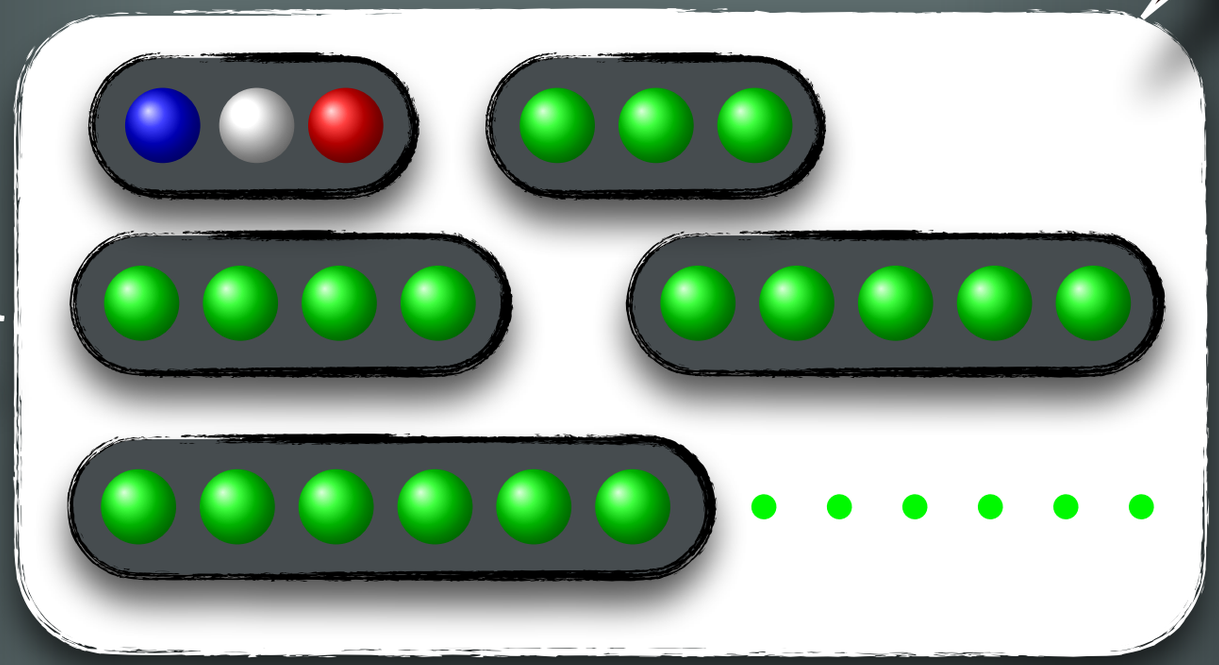
$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$

X



infinite

$\gamma_2(X)$

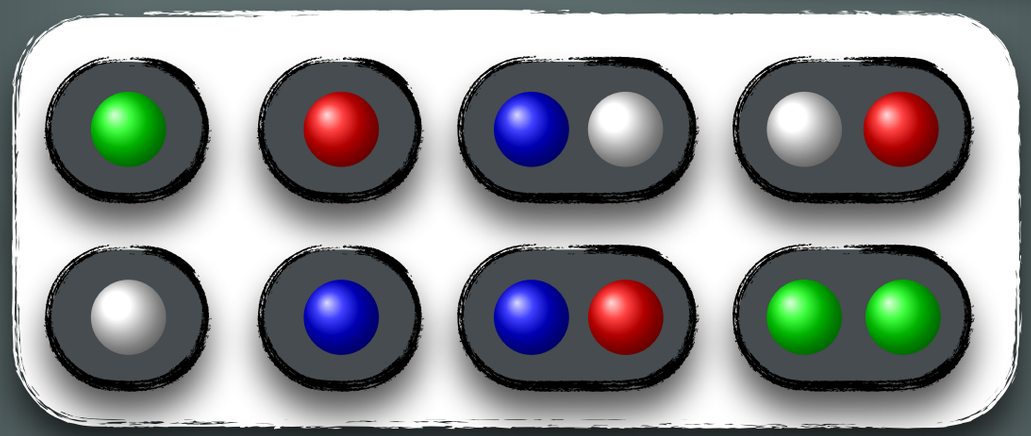


$+ X$

$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$

$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$

X

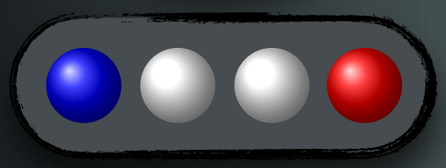


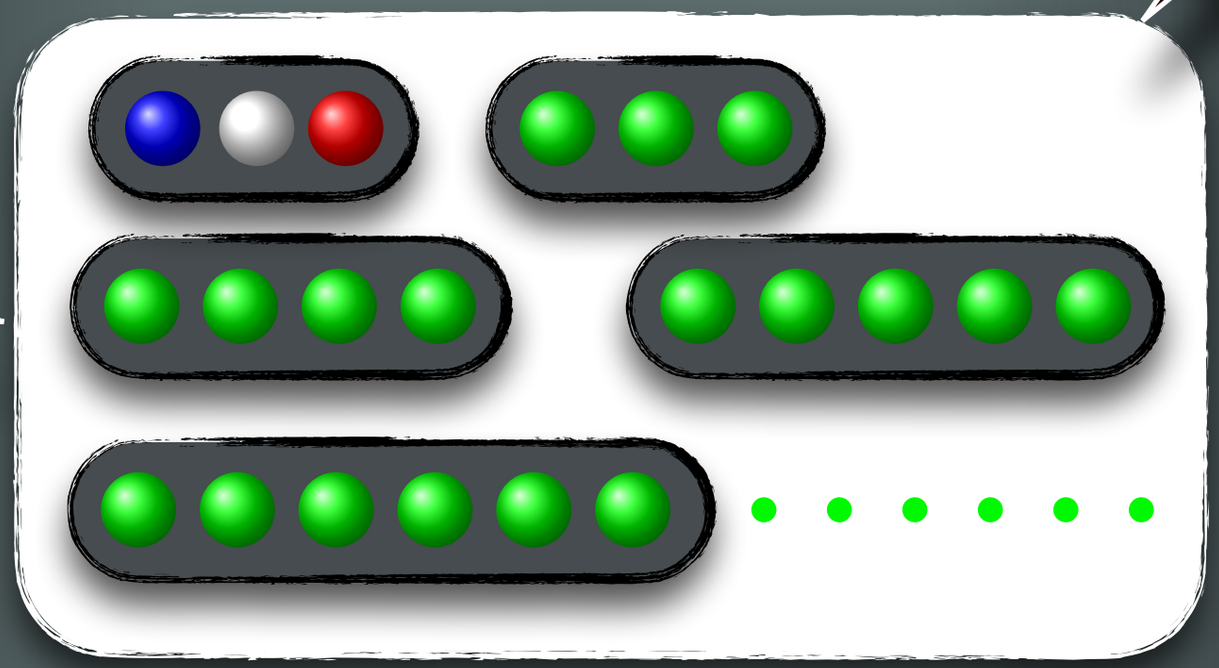
infinite

$\gamma_2(X)$

??

$\in$



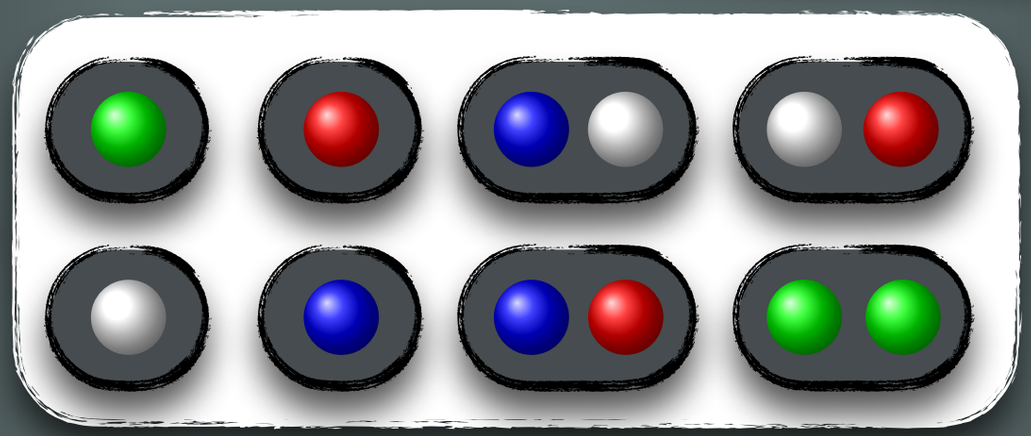


$+ X$

$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$

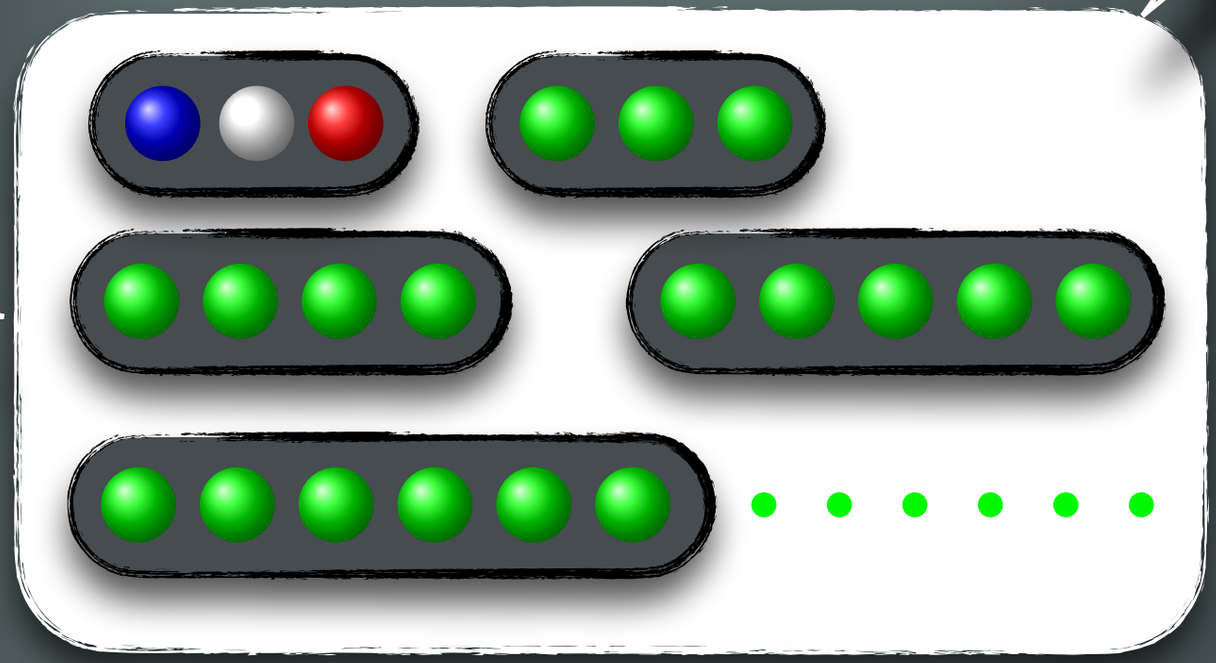
$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$

X



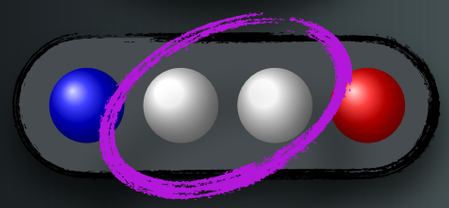
infinite

$\gamma_2(X)$



$??$

$\in$

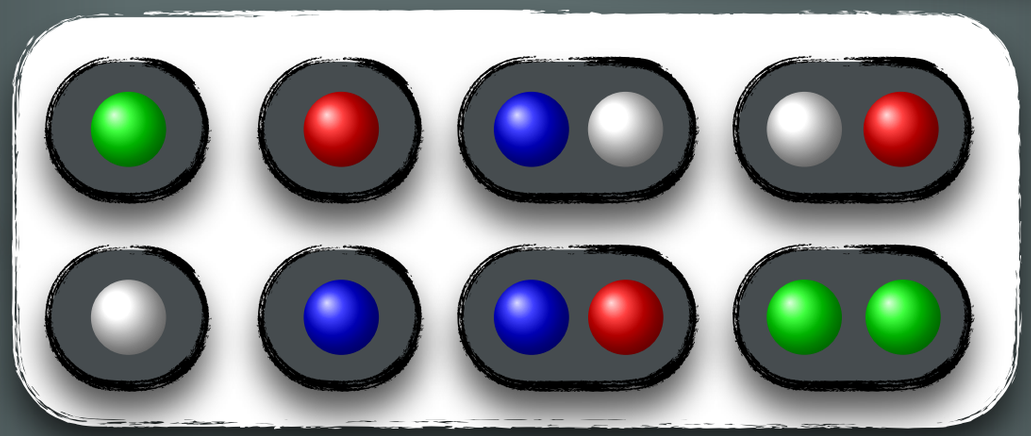


$+ X$

$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$

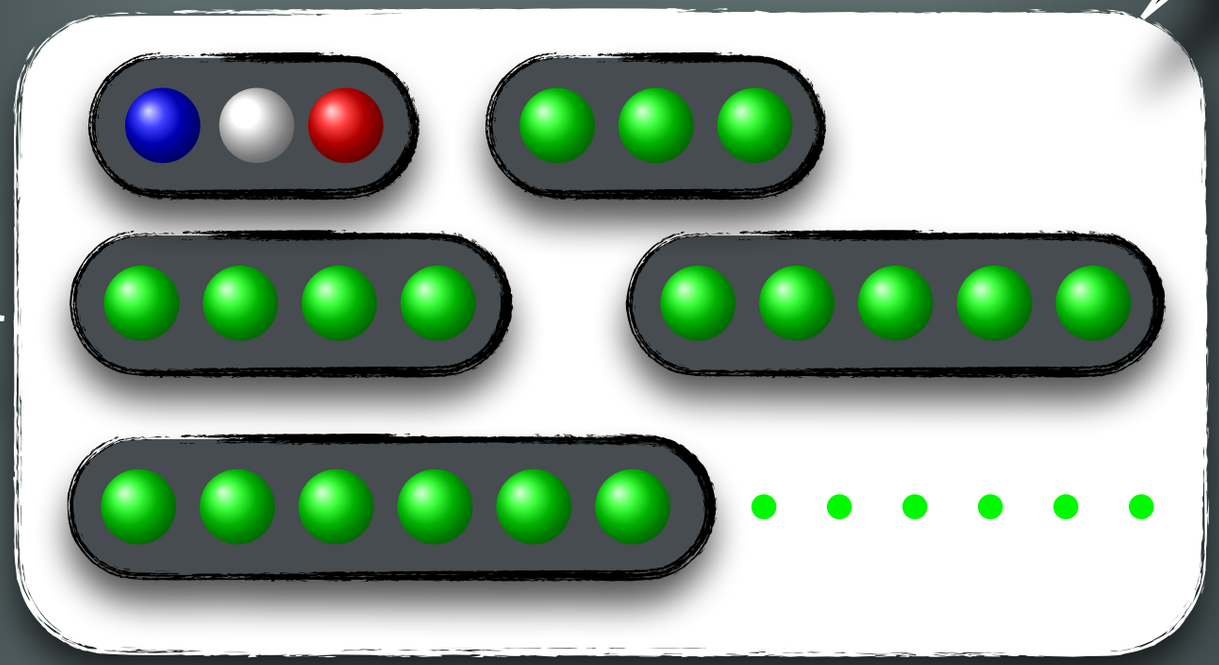
$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$

X



infinite

$\gamma_2(X)$

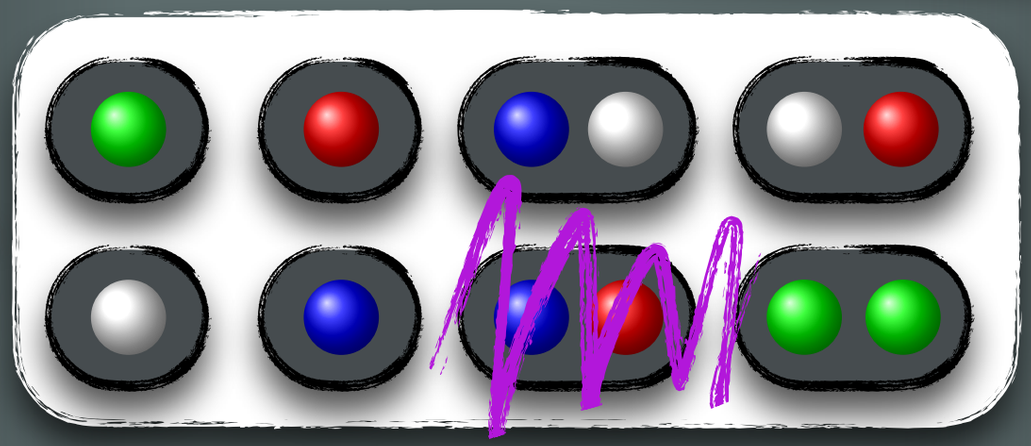


$+ X$

$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$

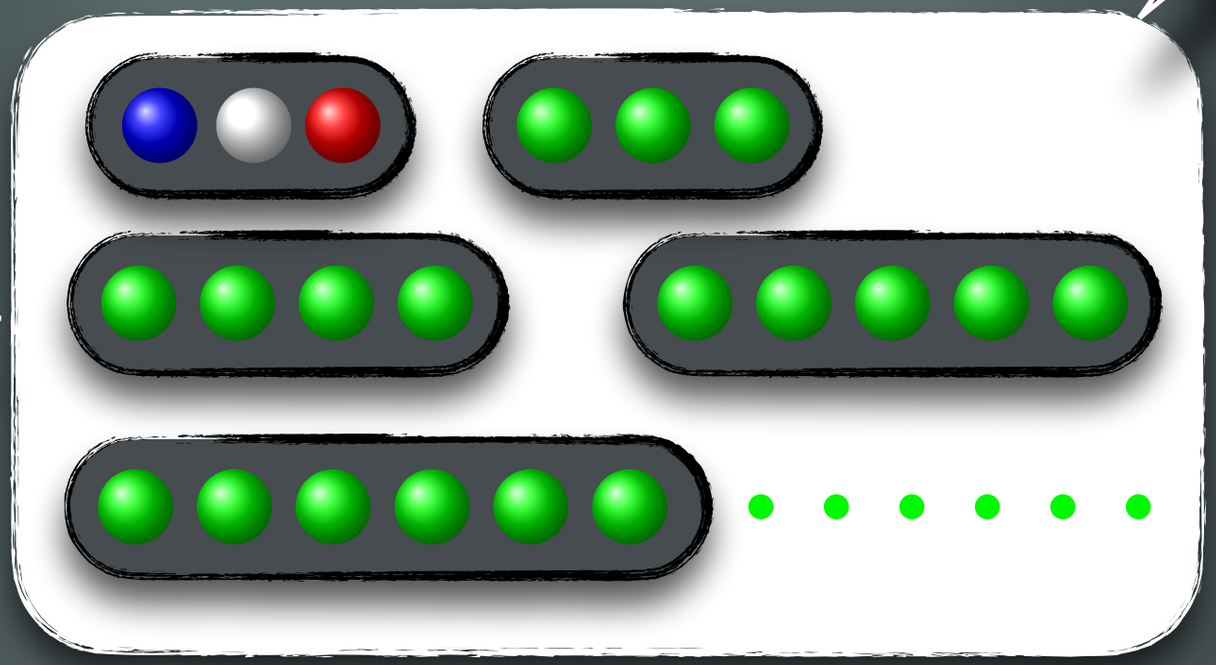
$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$

X



infinite

$\gamma_2(X)$

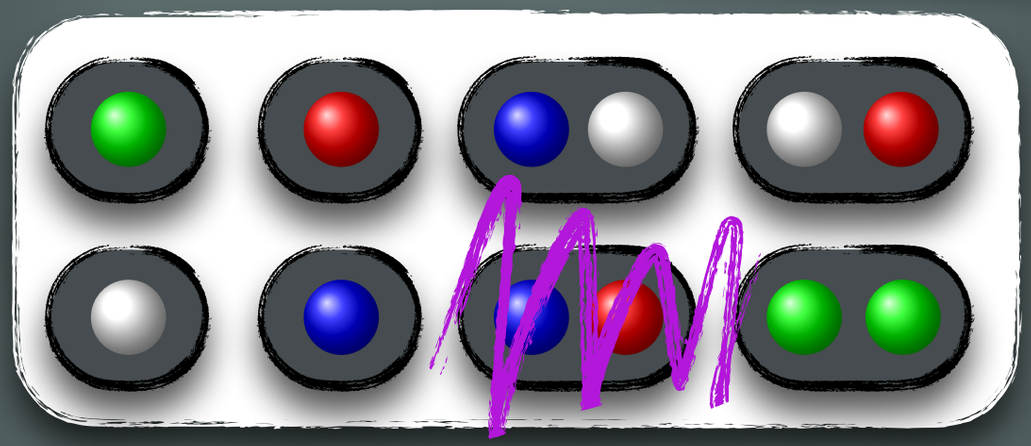


$+ X$

$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$

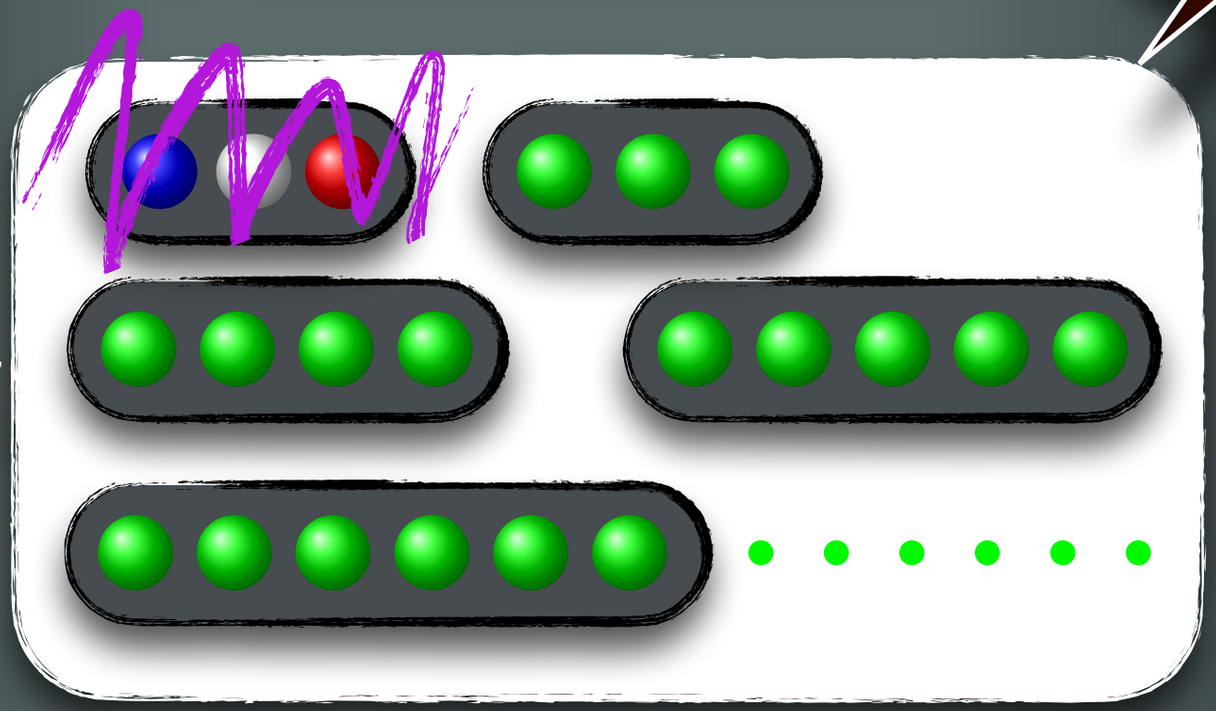
$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$

X



infinite

$\gamma_2(X)$



$+ X$

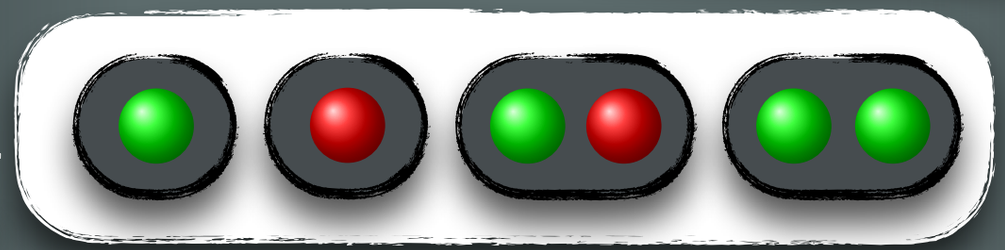
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

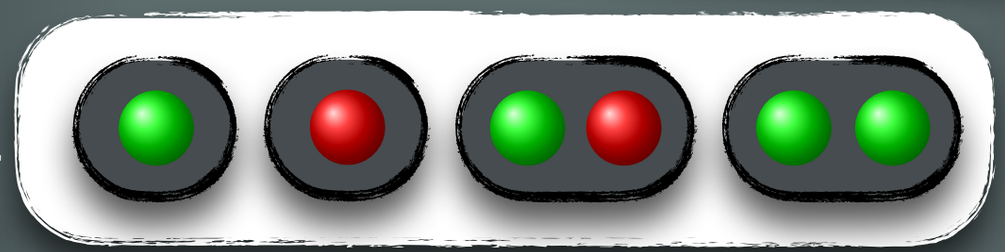
X



$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X

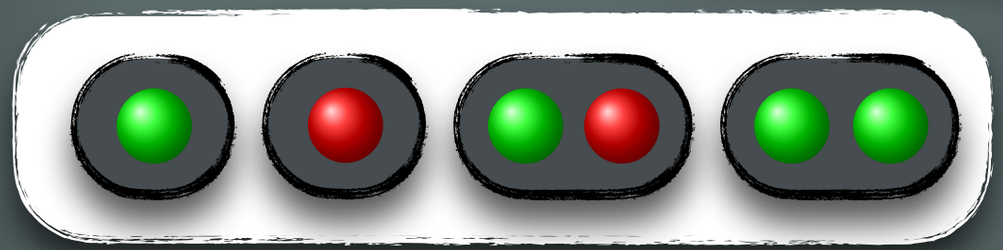


$\gamma_2(X)$

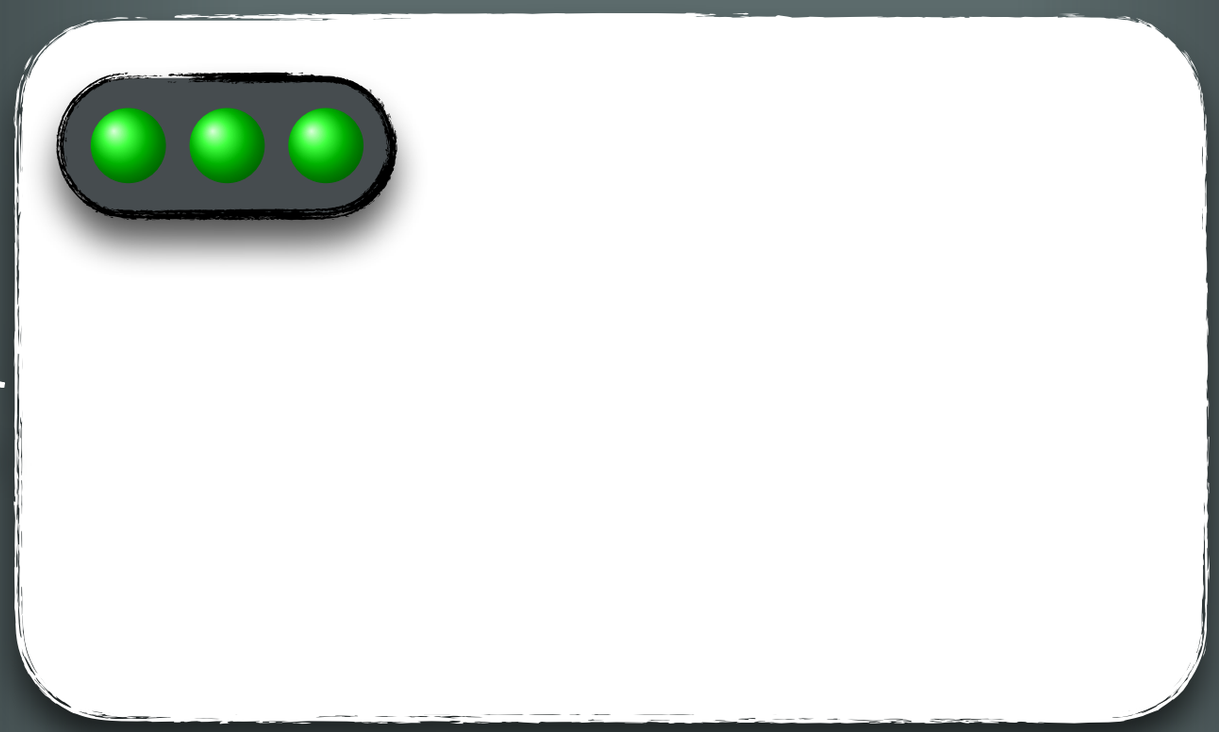
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X



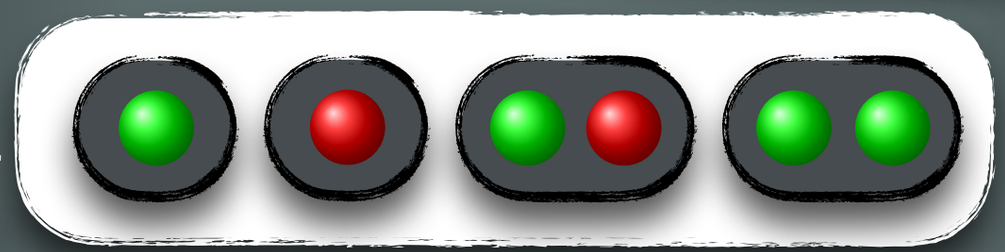
$\gamma_2(X)$



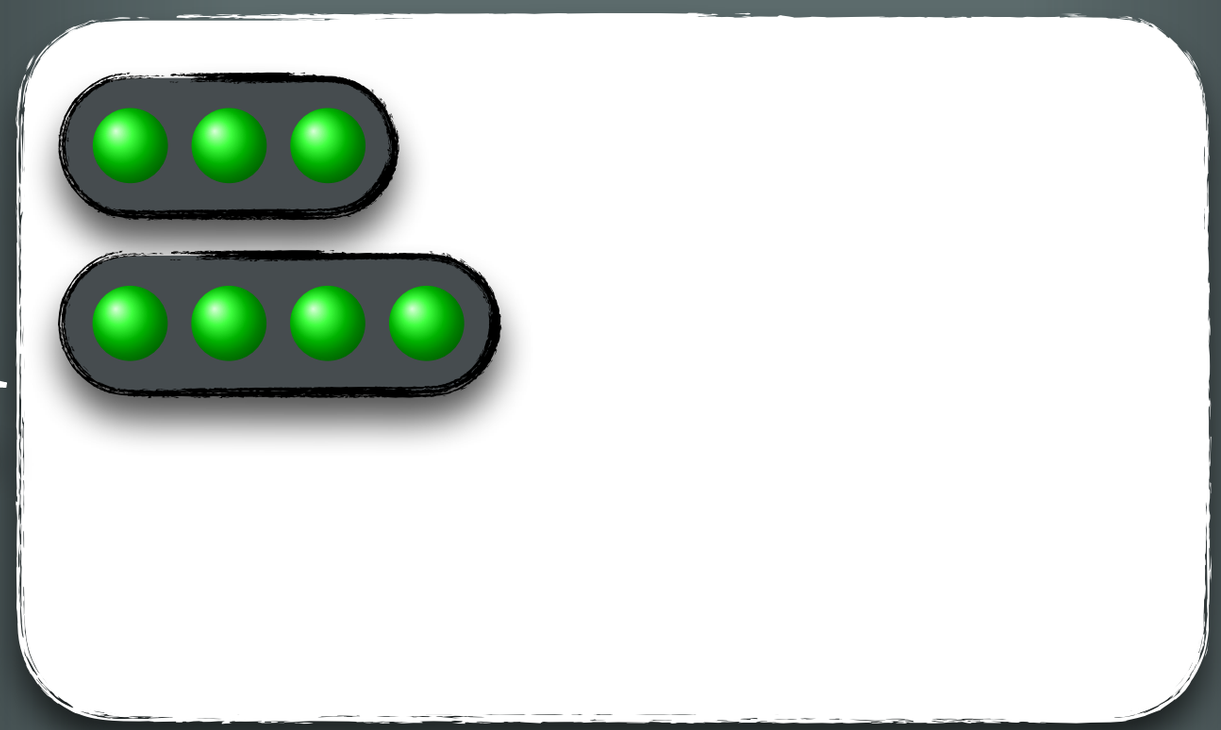
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X

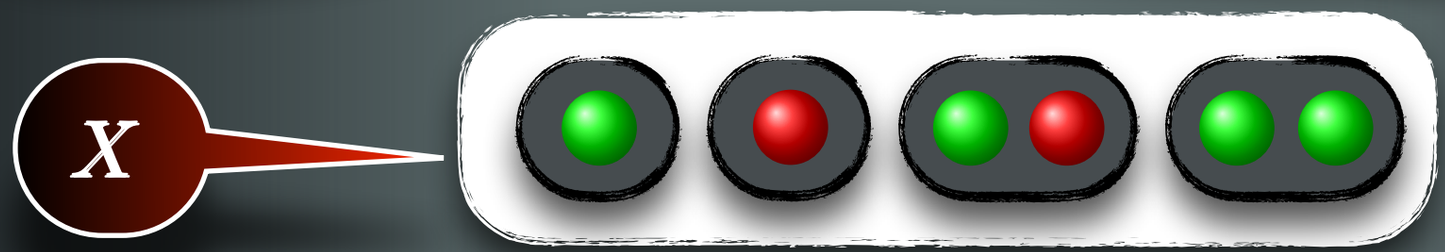


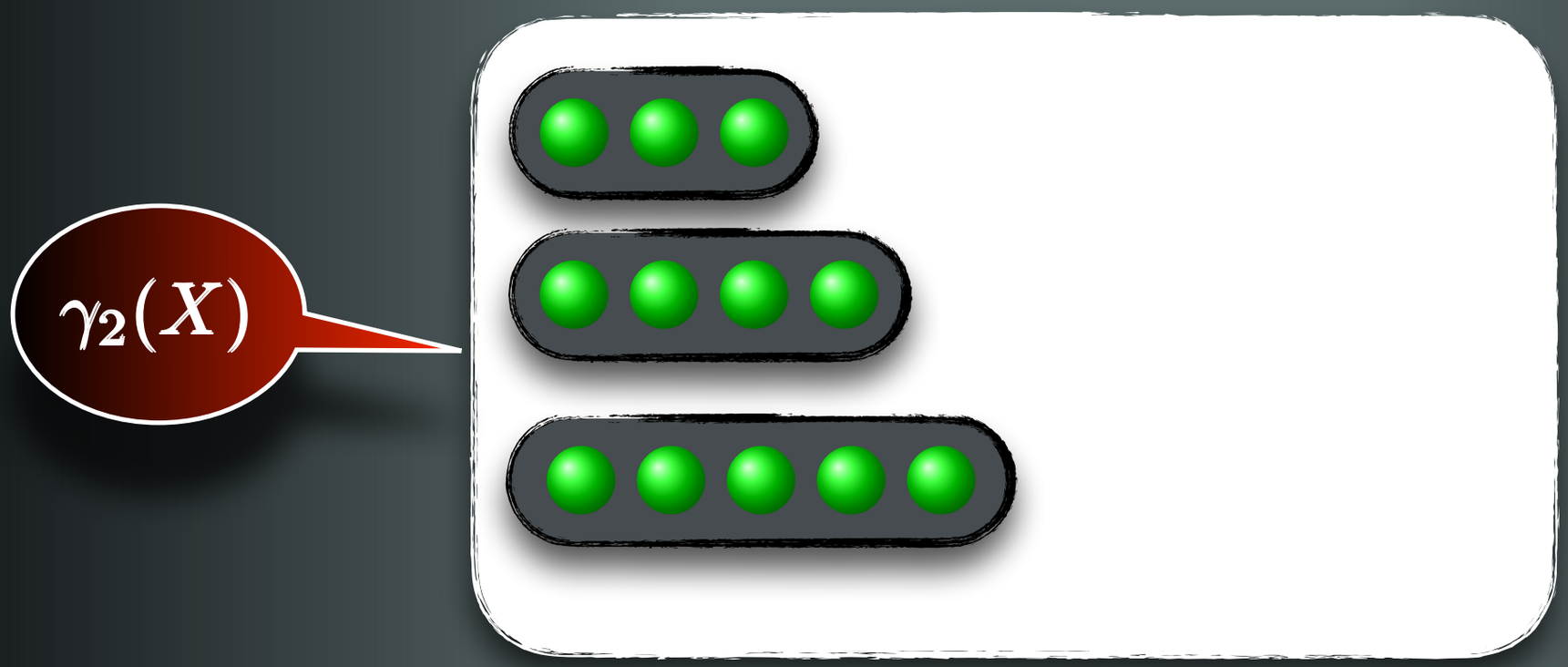
$\gamma_2(X)$



$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

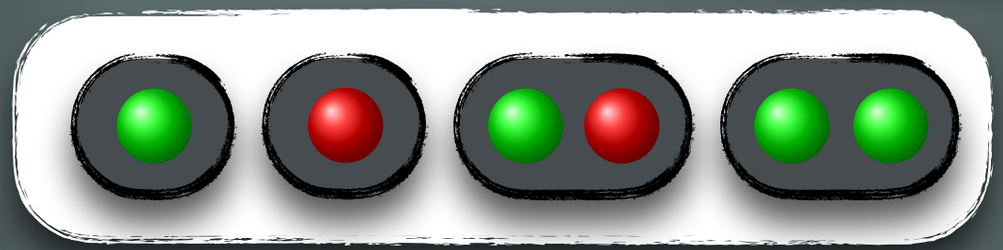




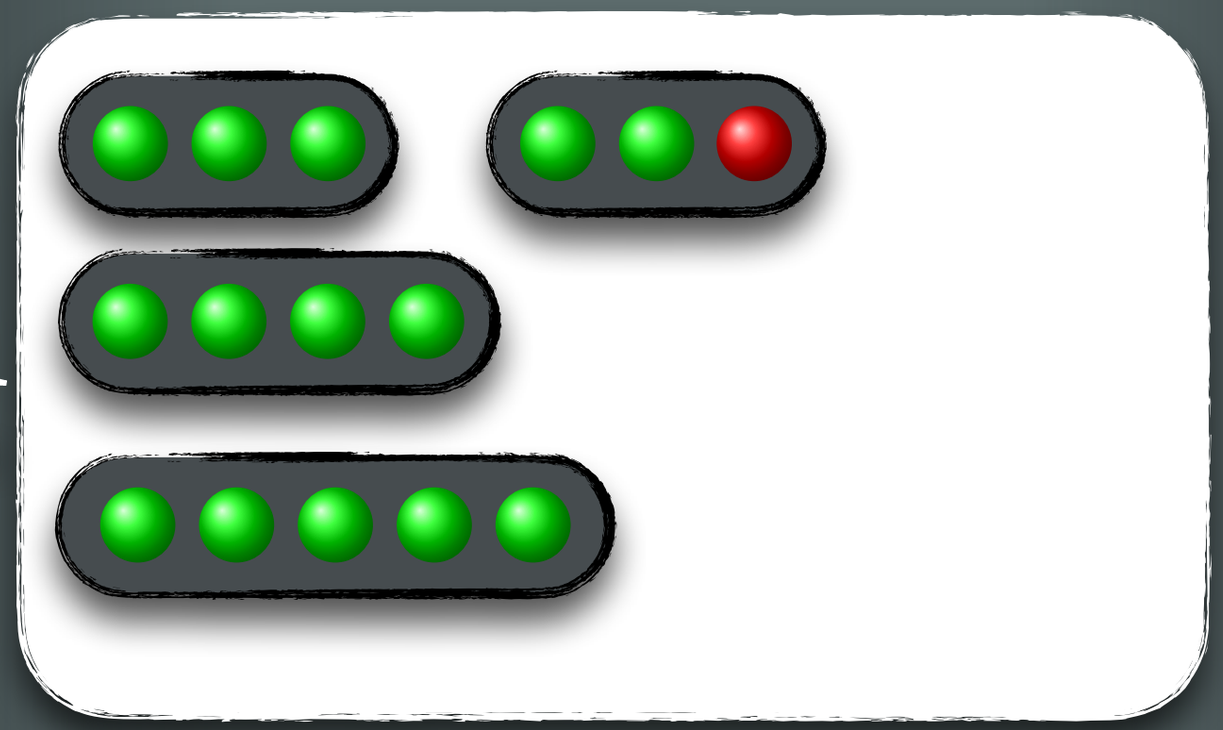
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X



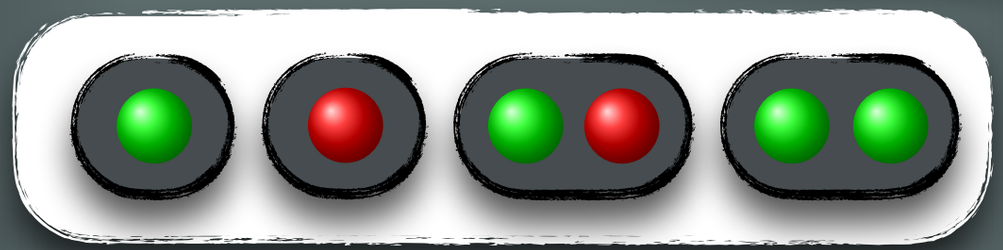
$\gamma_2(X)$



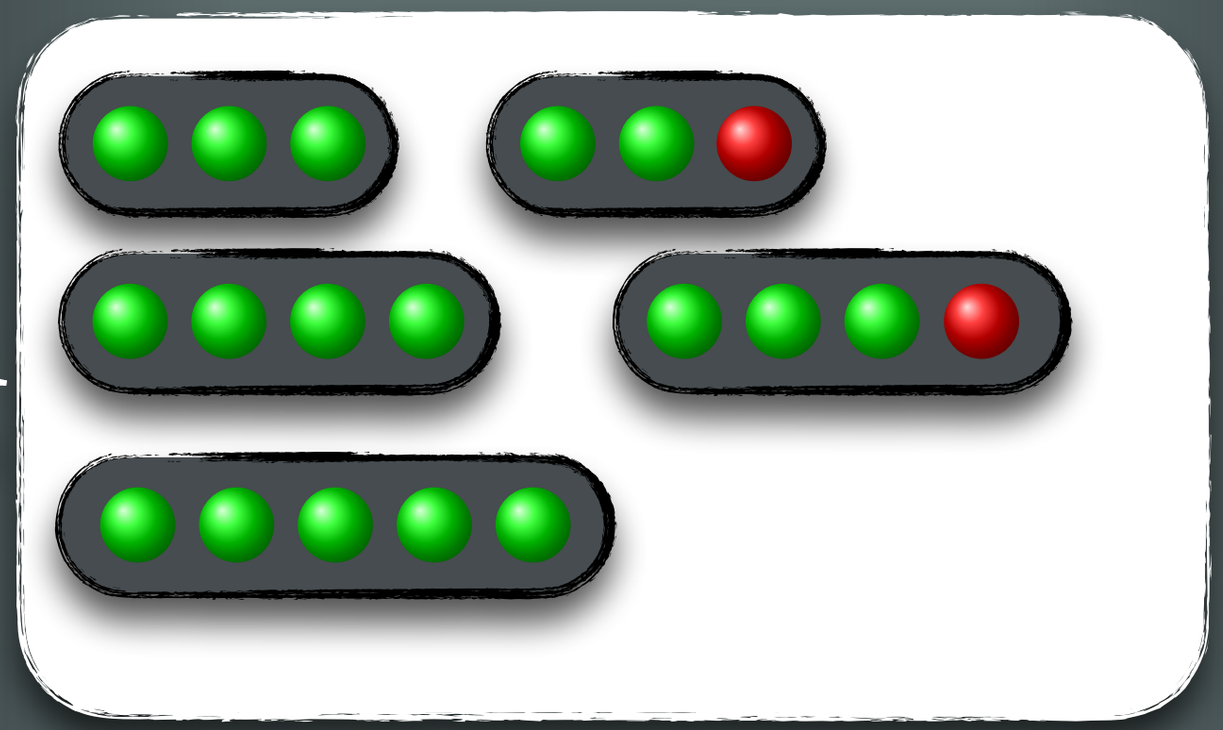
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X



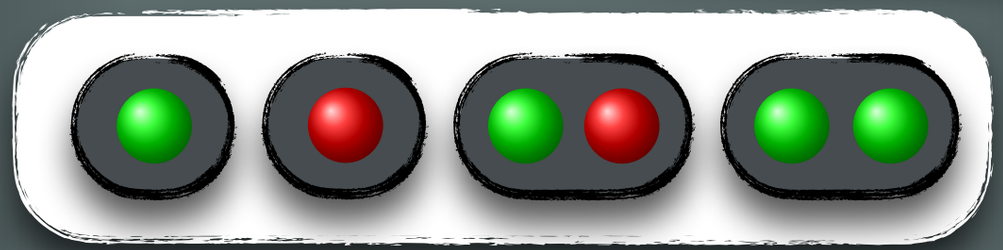
$\gamma_2(X)$



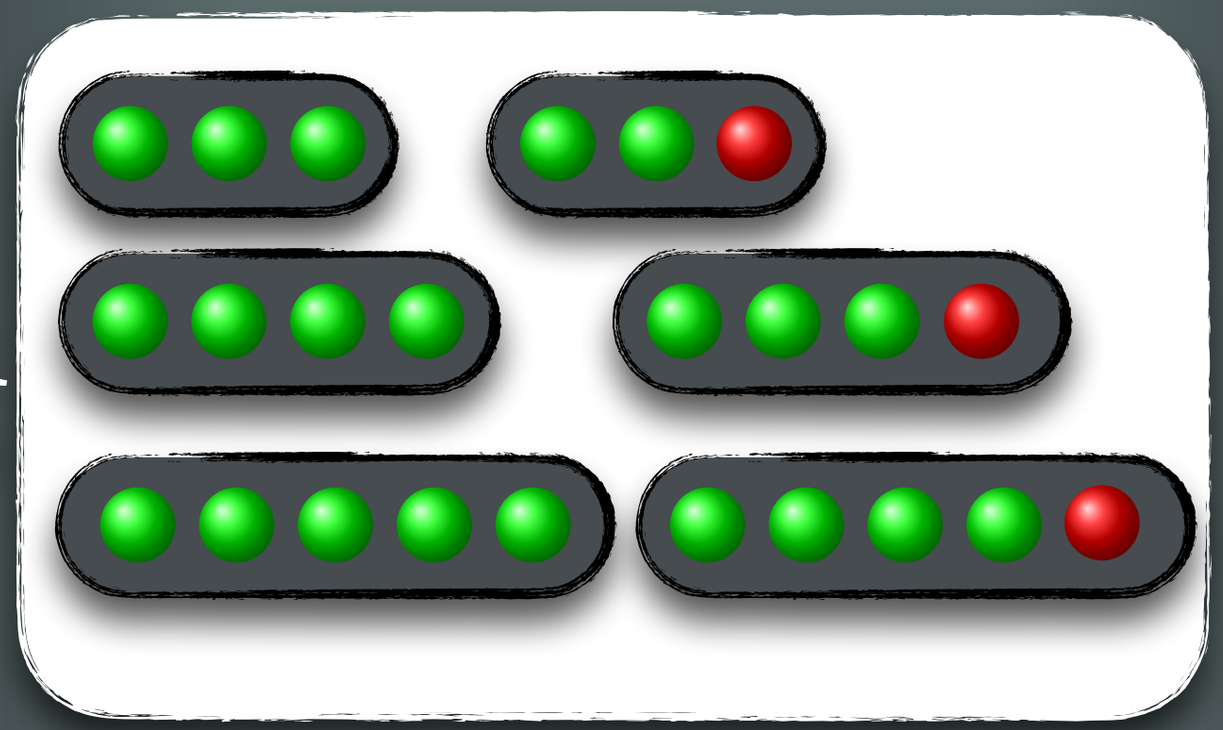
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X



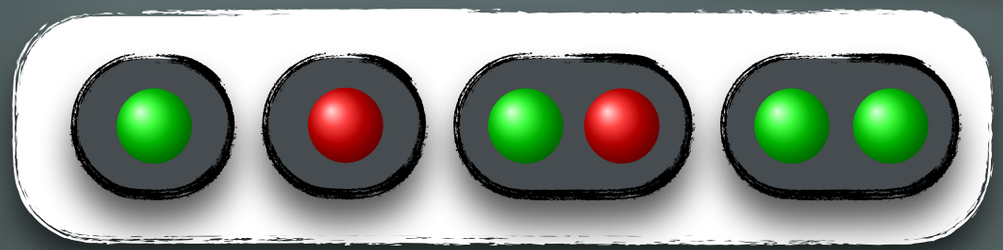
$\gamma_2(X)$



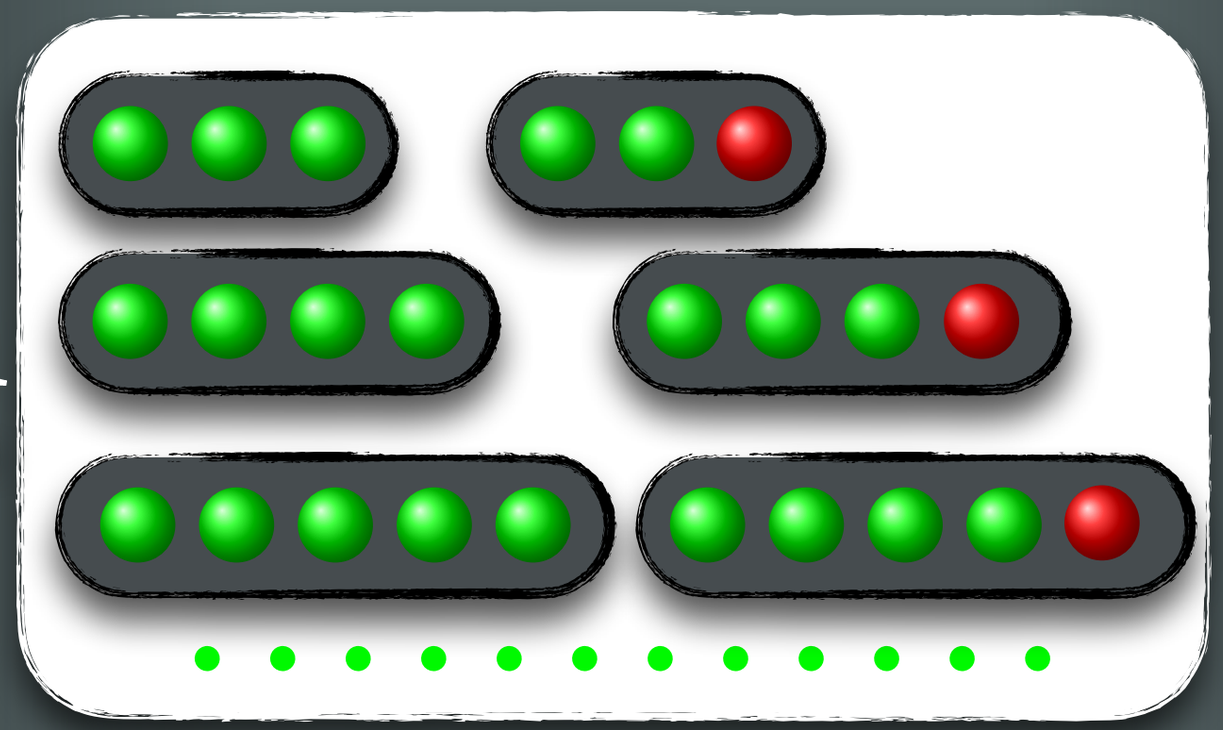
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

X

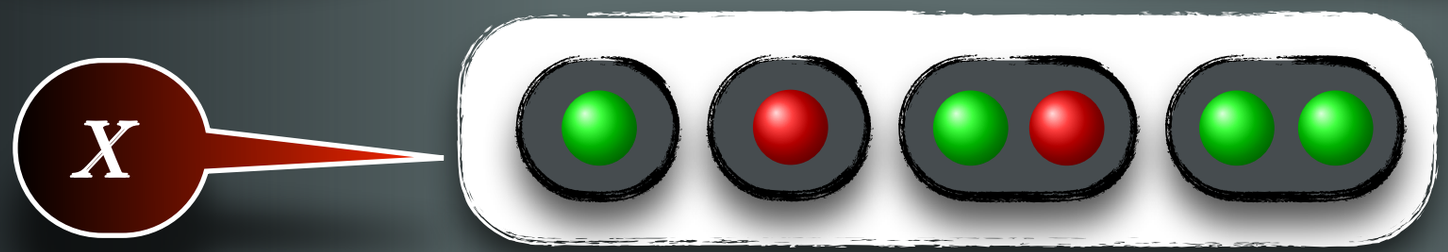


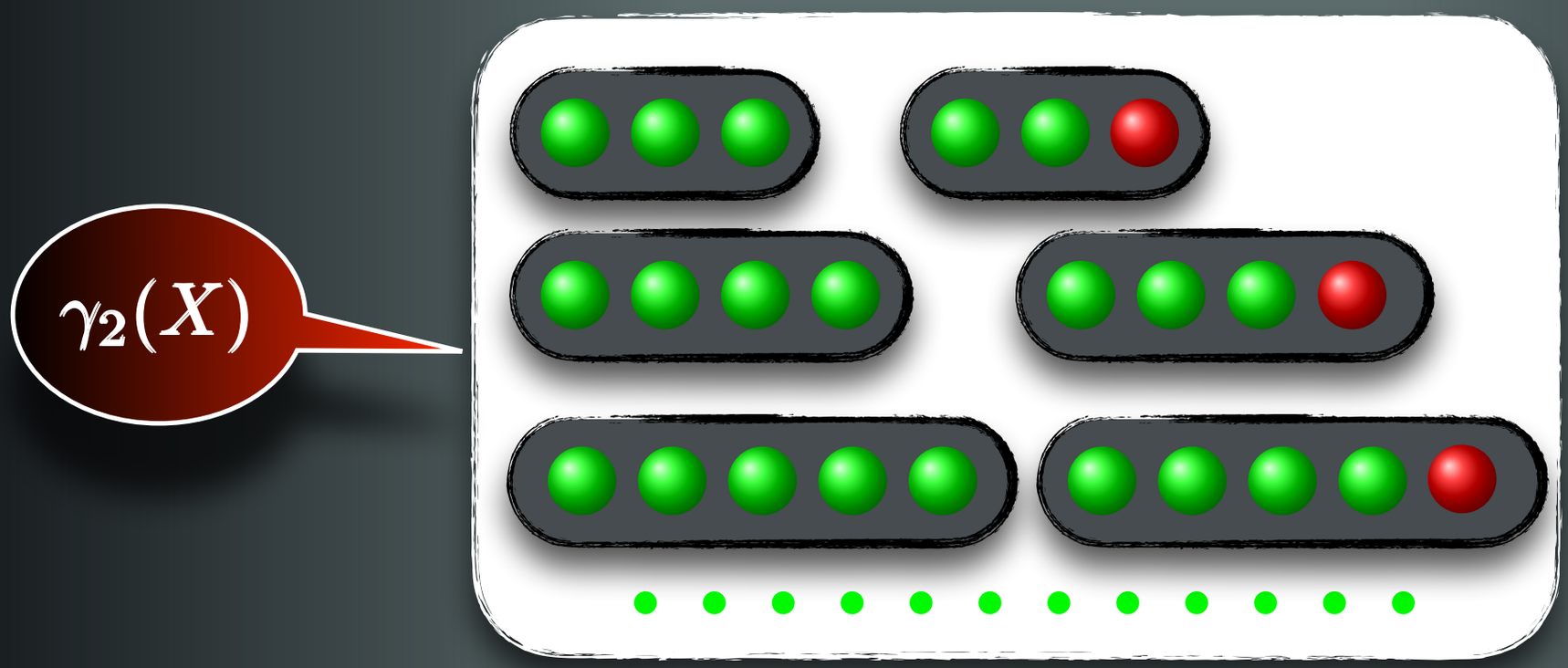
$\gamma_2(X)$



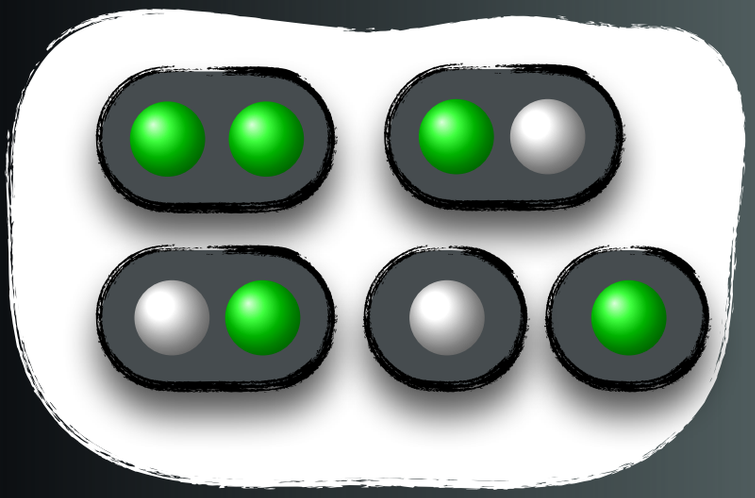
$$\gamma_k : 2^{\mathbb{V}_k} \to 2^{\mathbb{C}}$$

$$\gamma_k(X) = \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

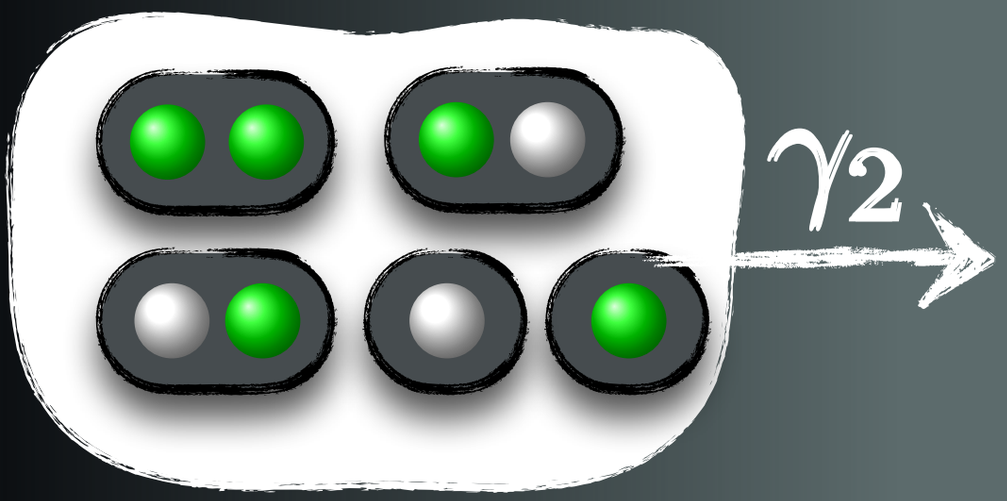




$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$



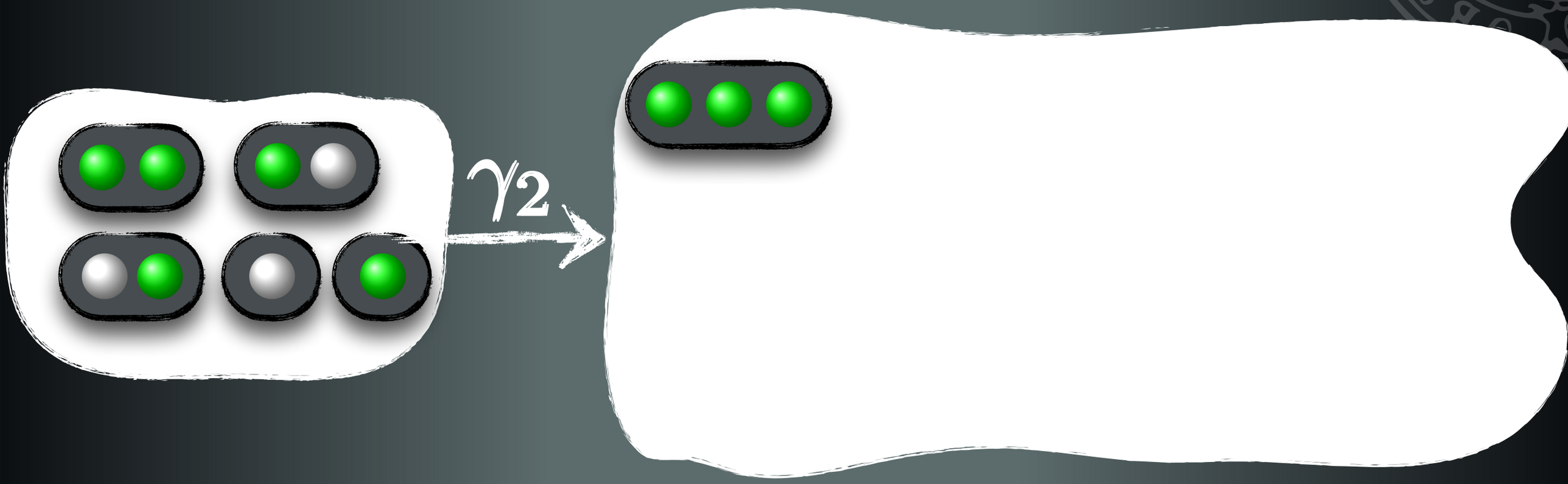
$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$



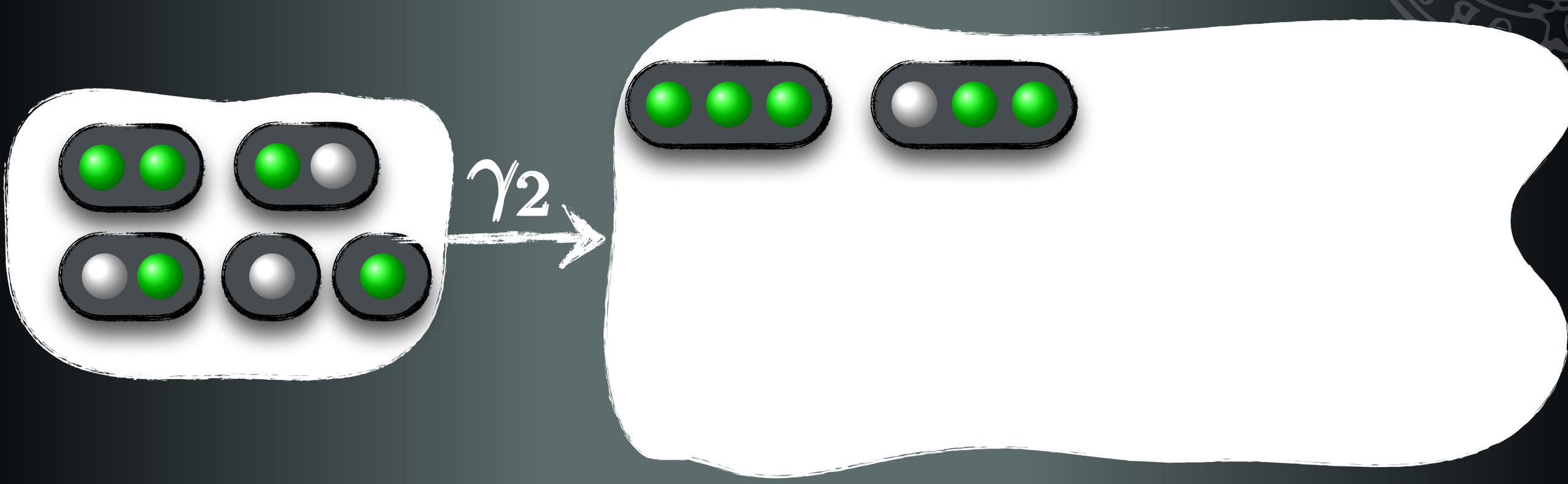
$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$



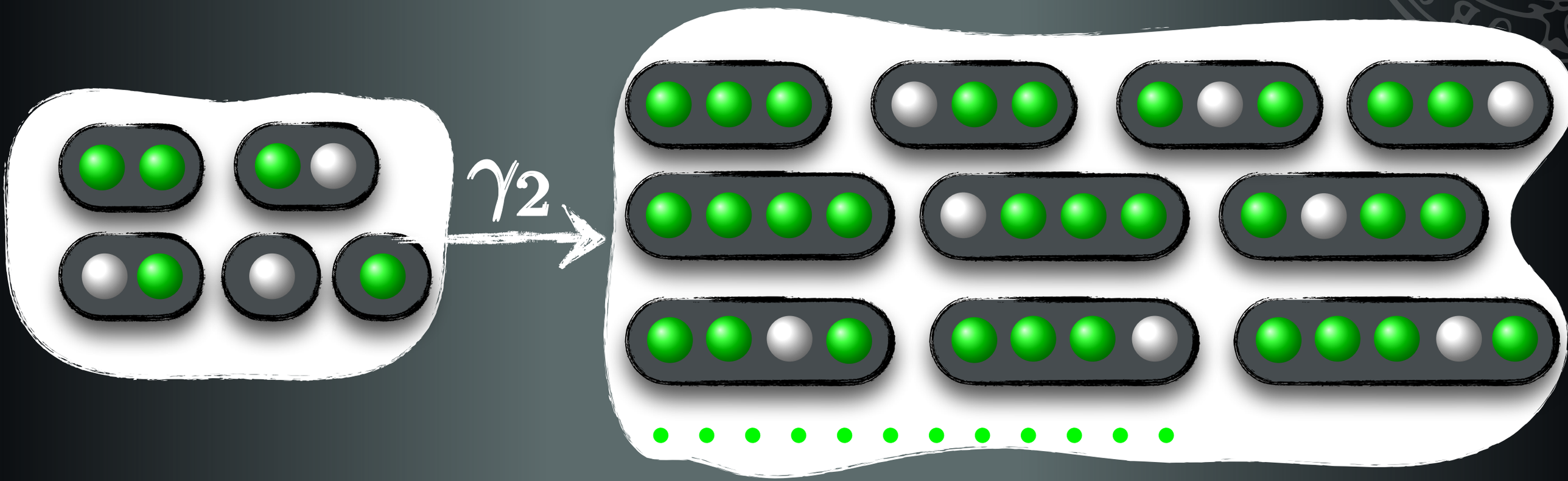
$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$



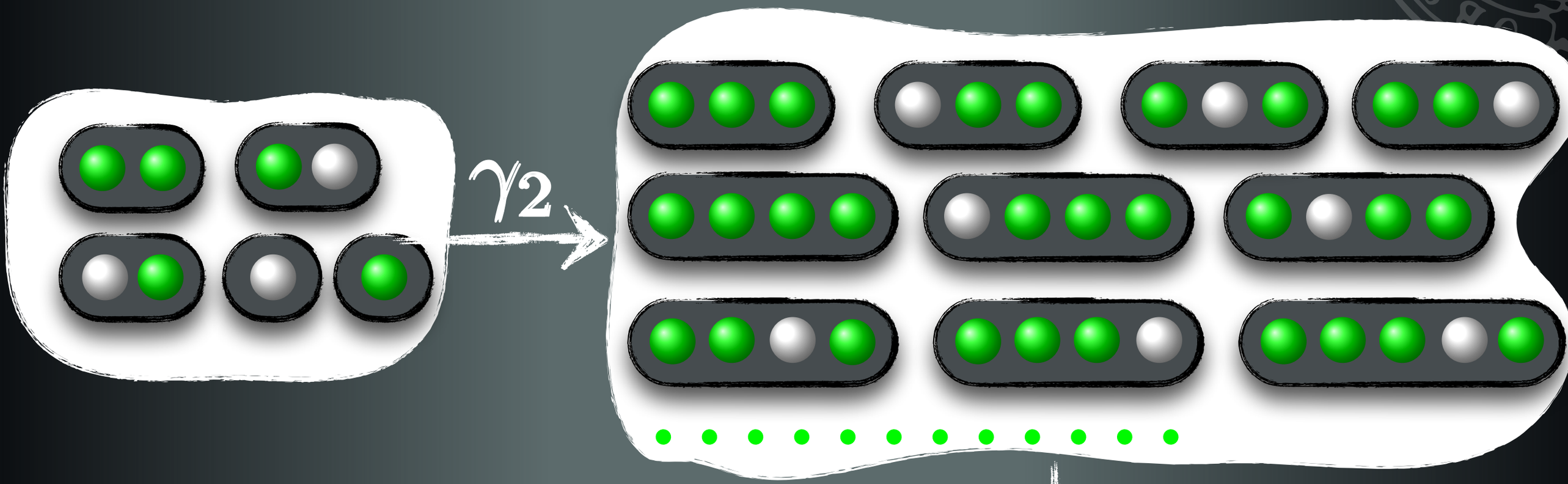
$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$

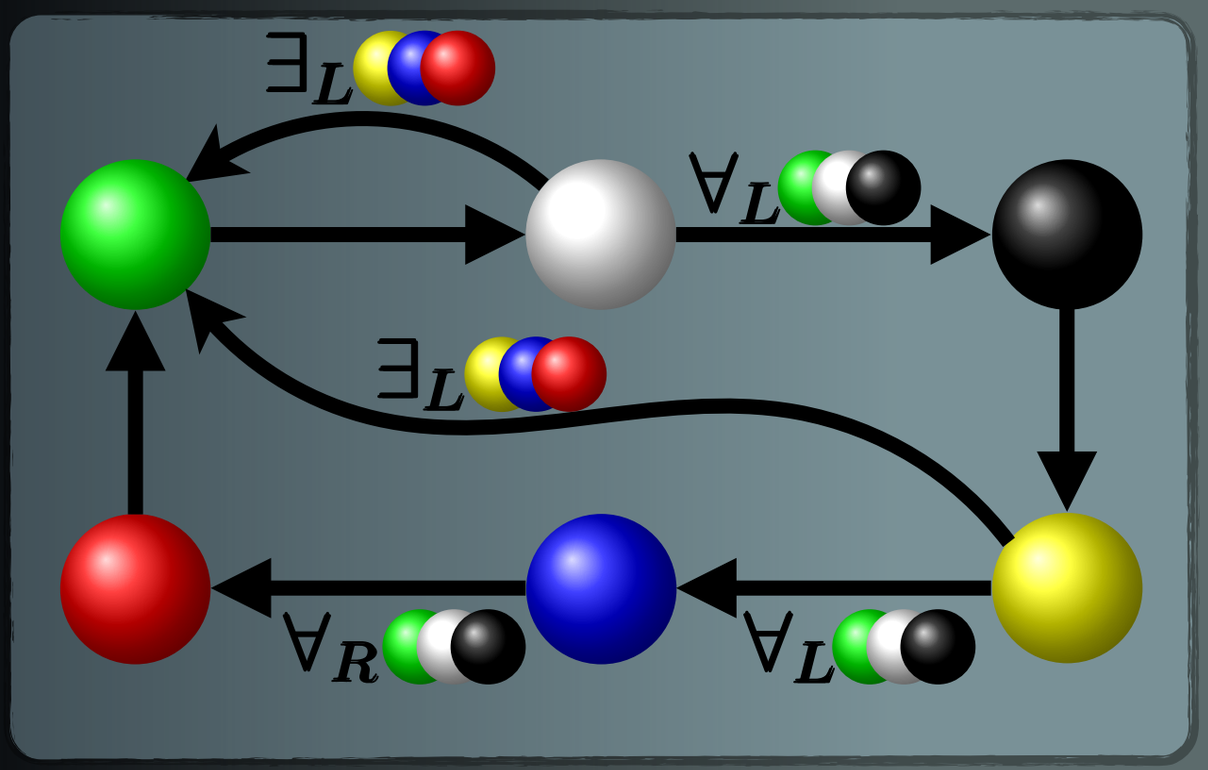


$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$

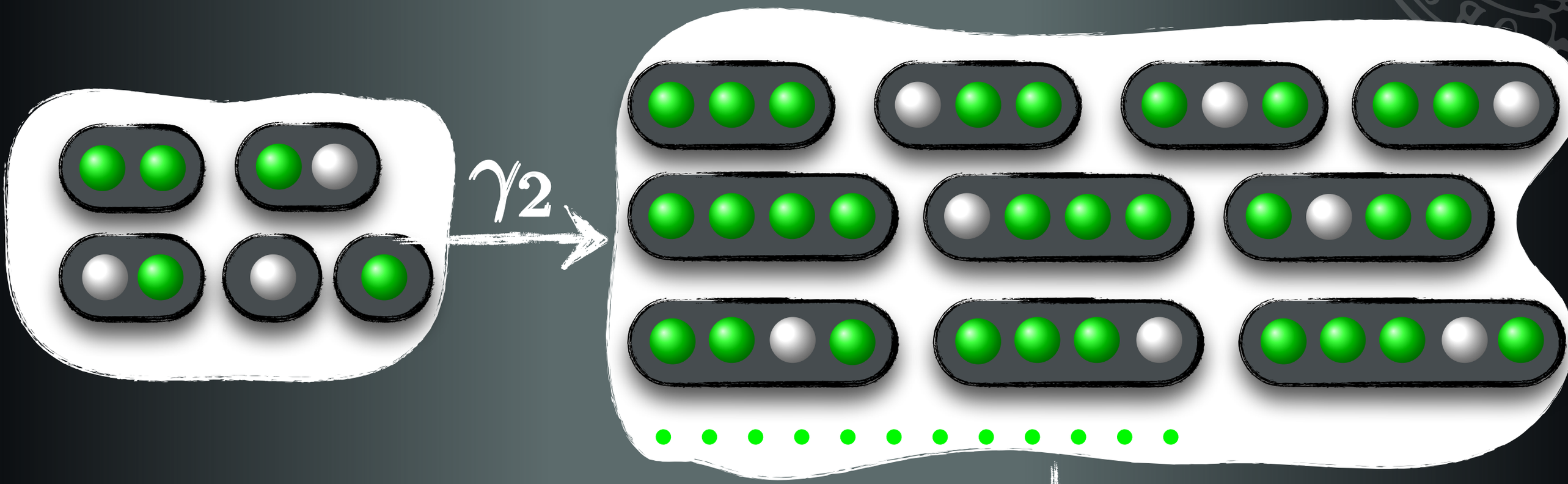


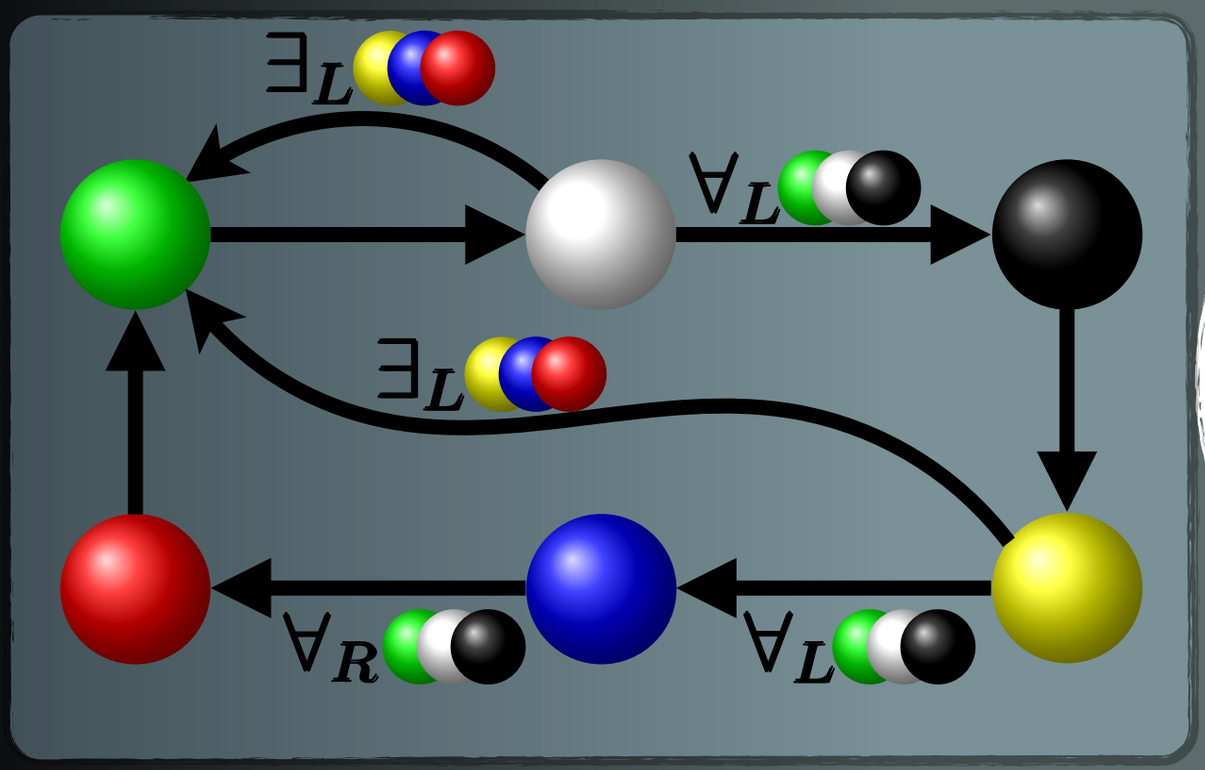
$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$



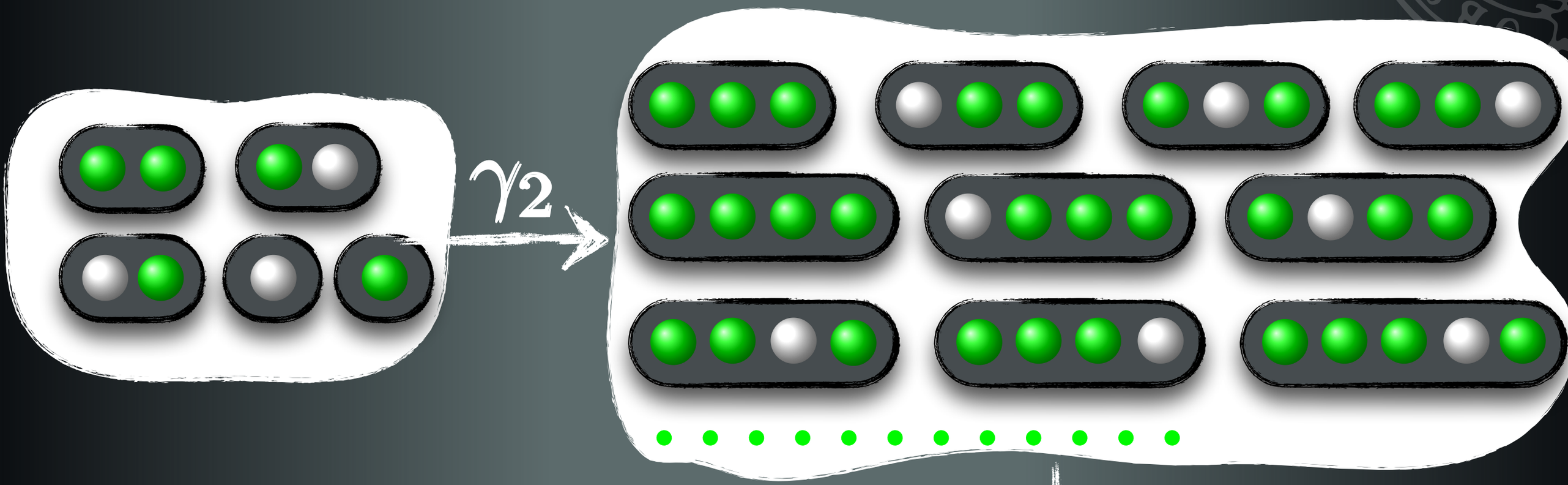


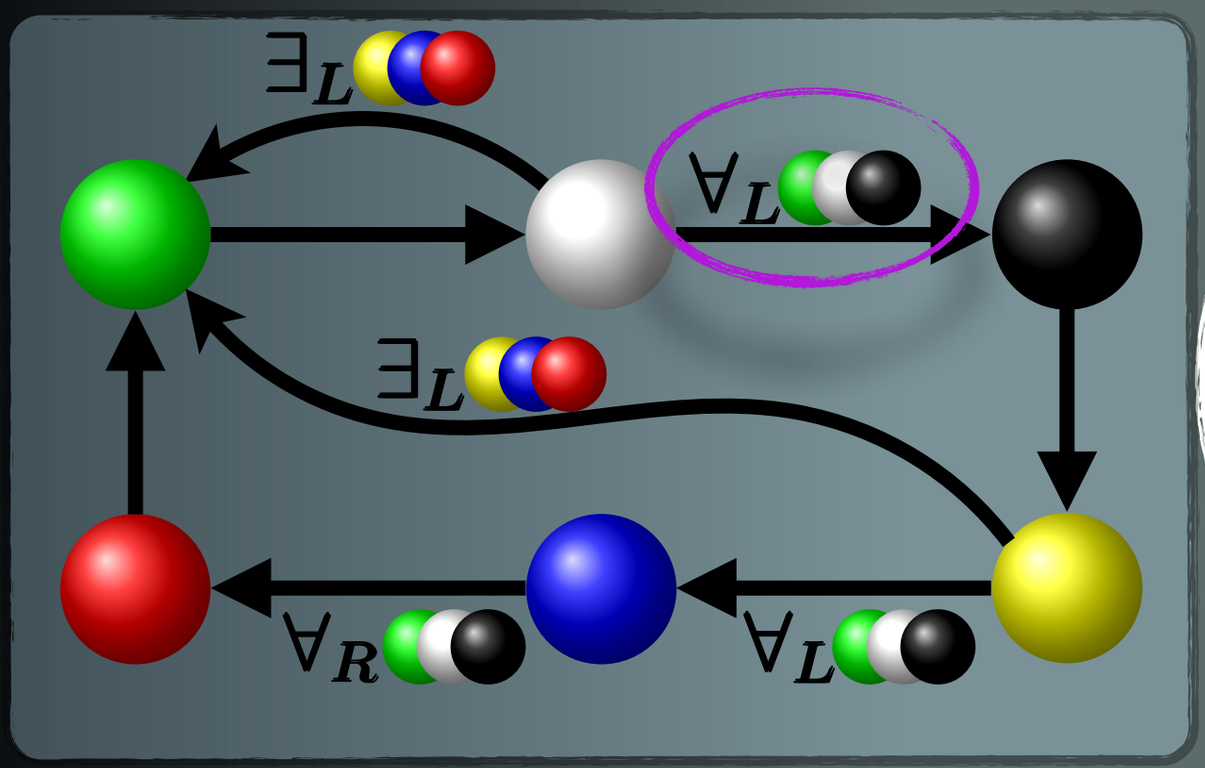
$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$



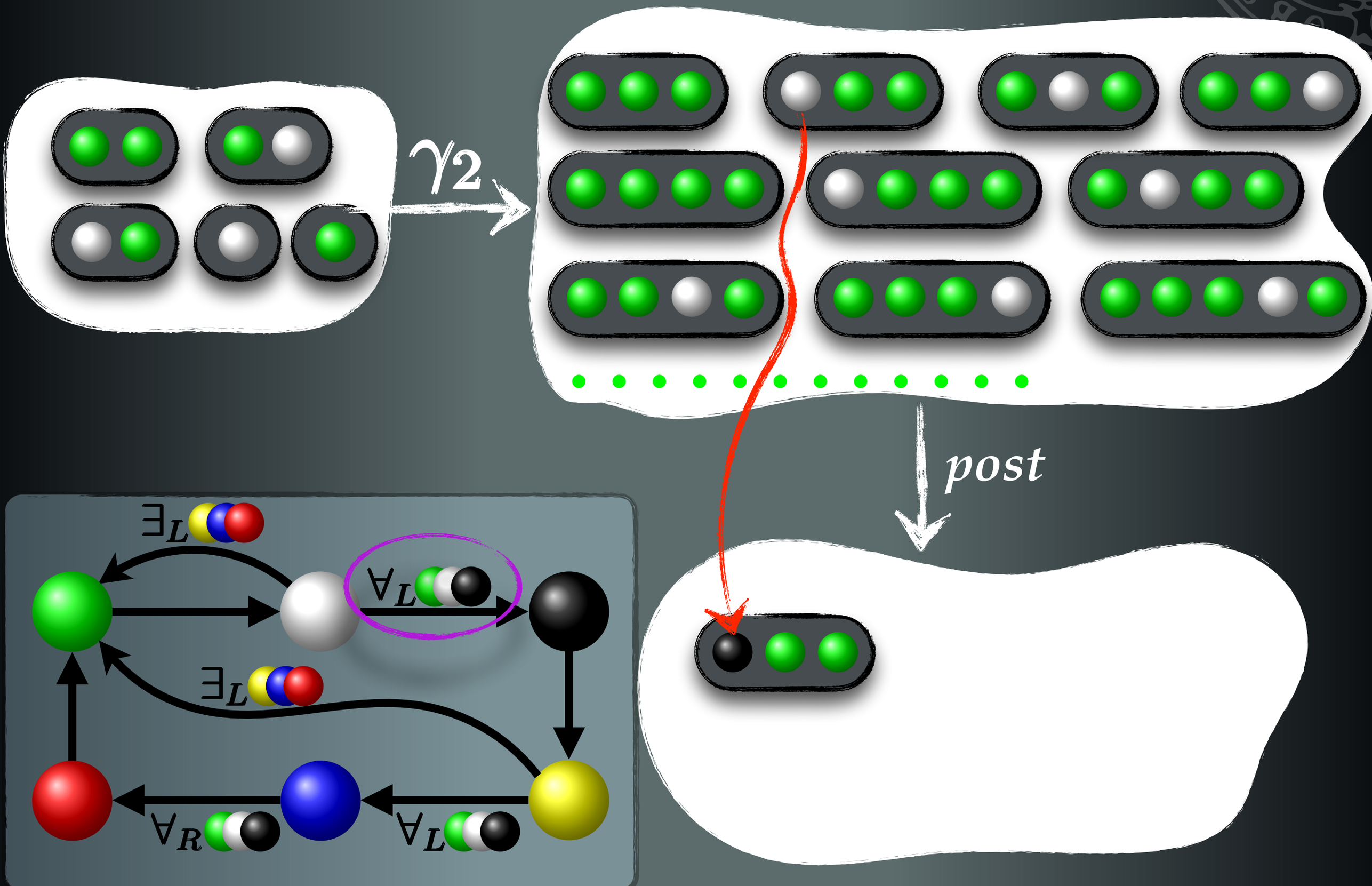


$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$

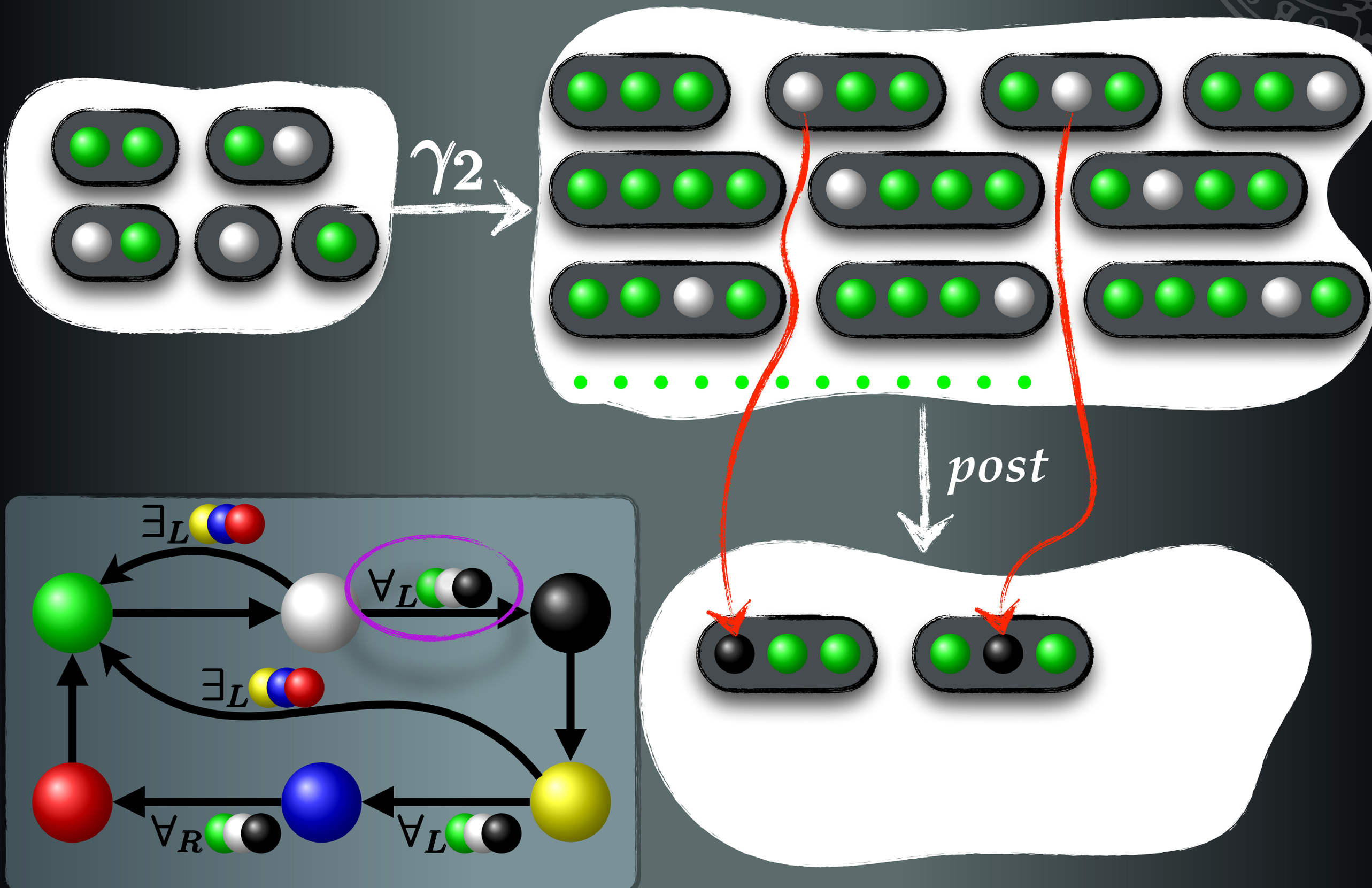




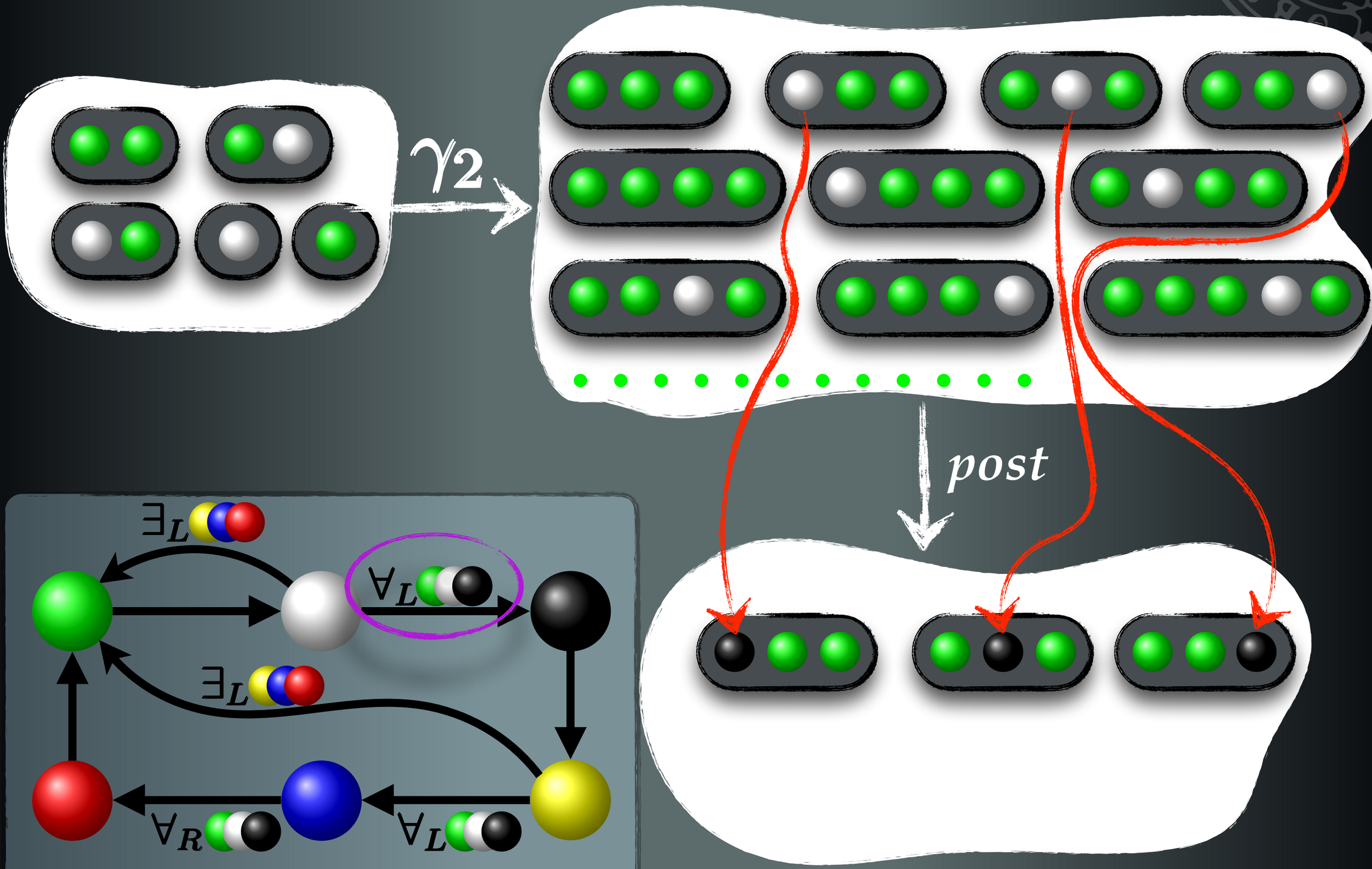
$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$



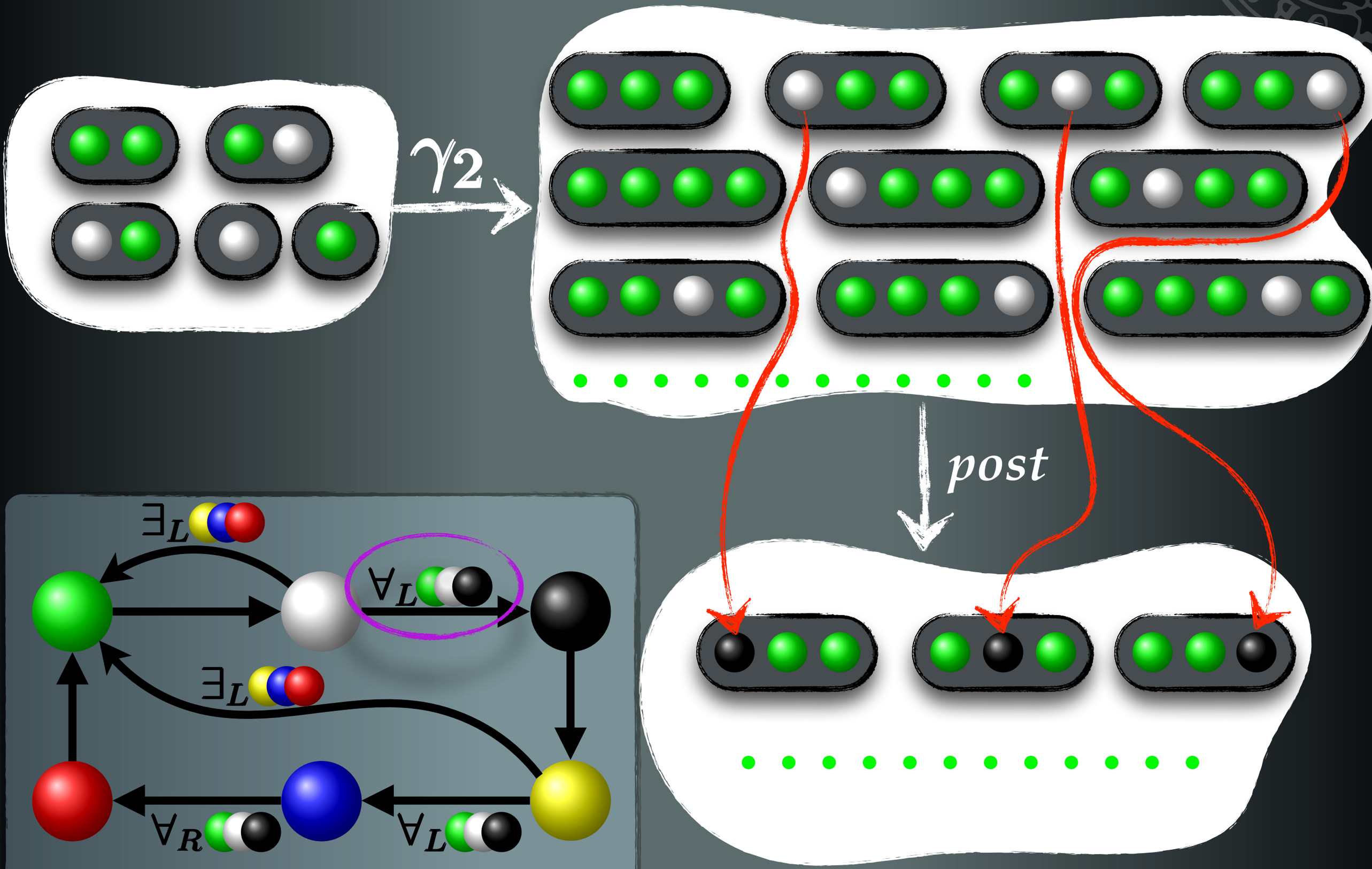
$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$



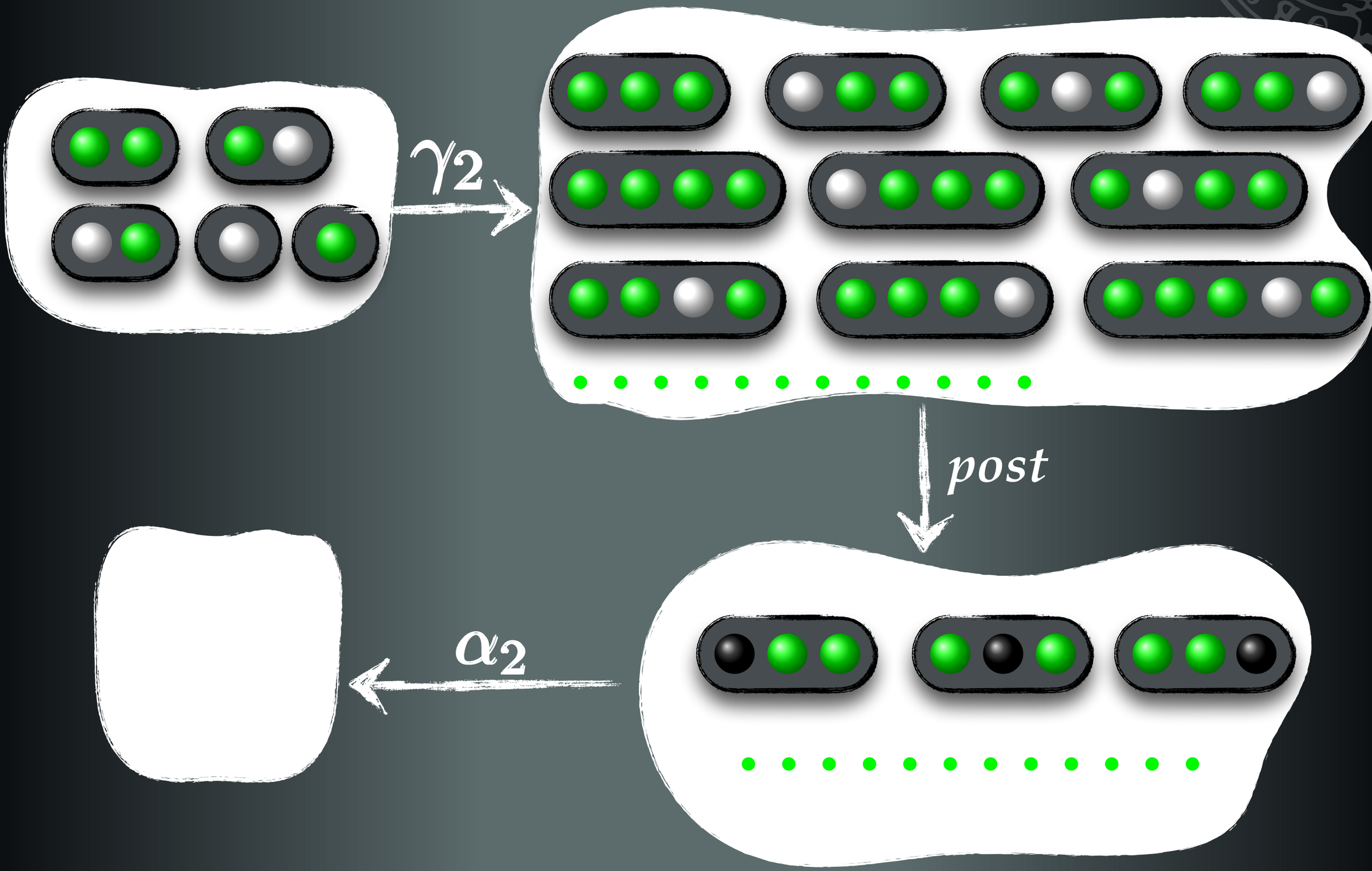
$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$



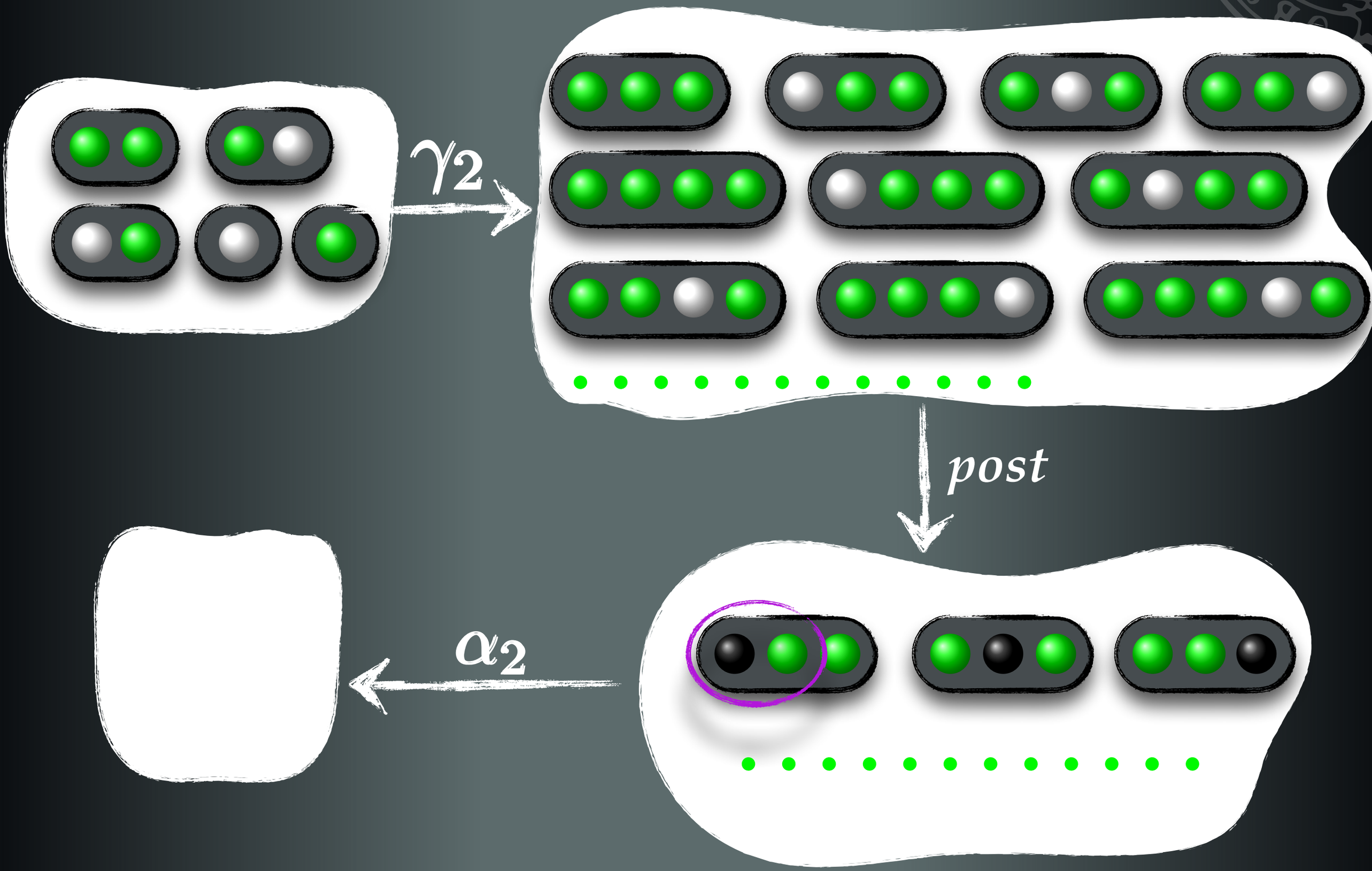
$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$



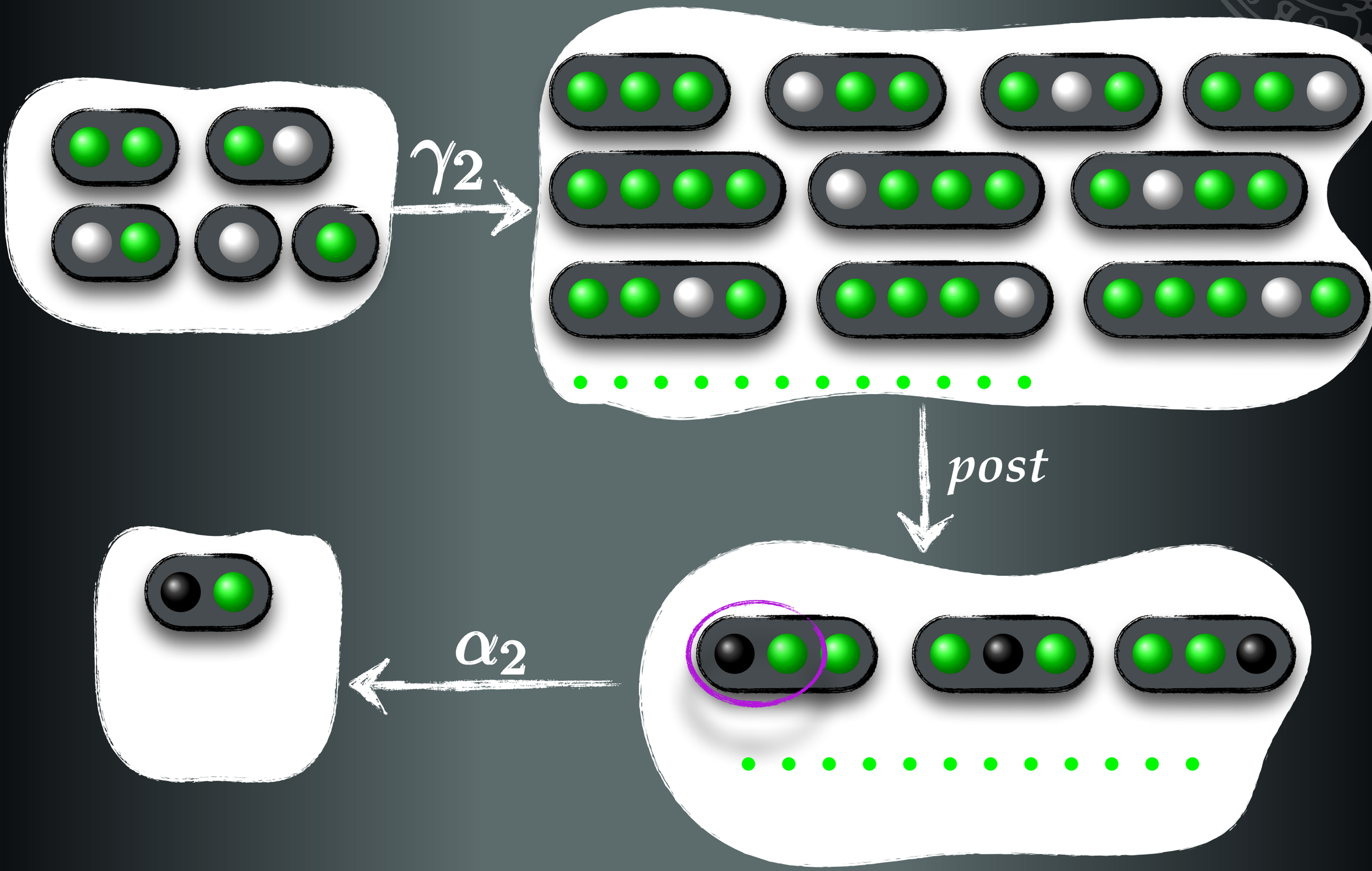
$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$



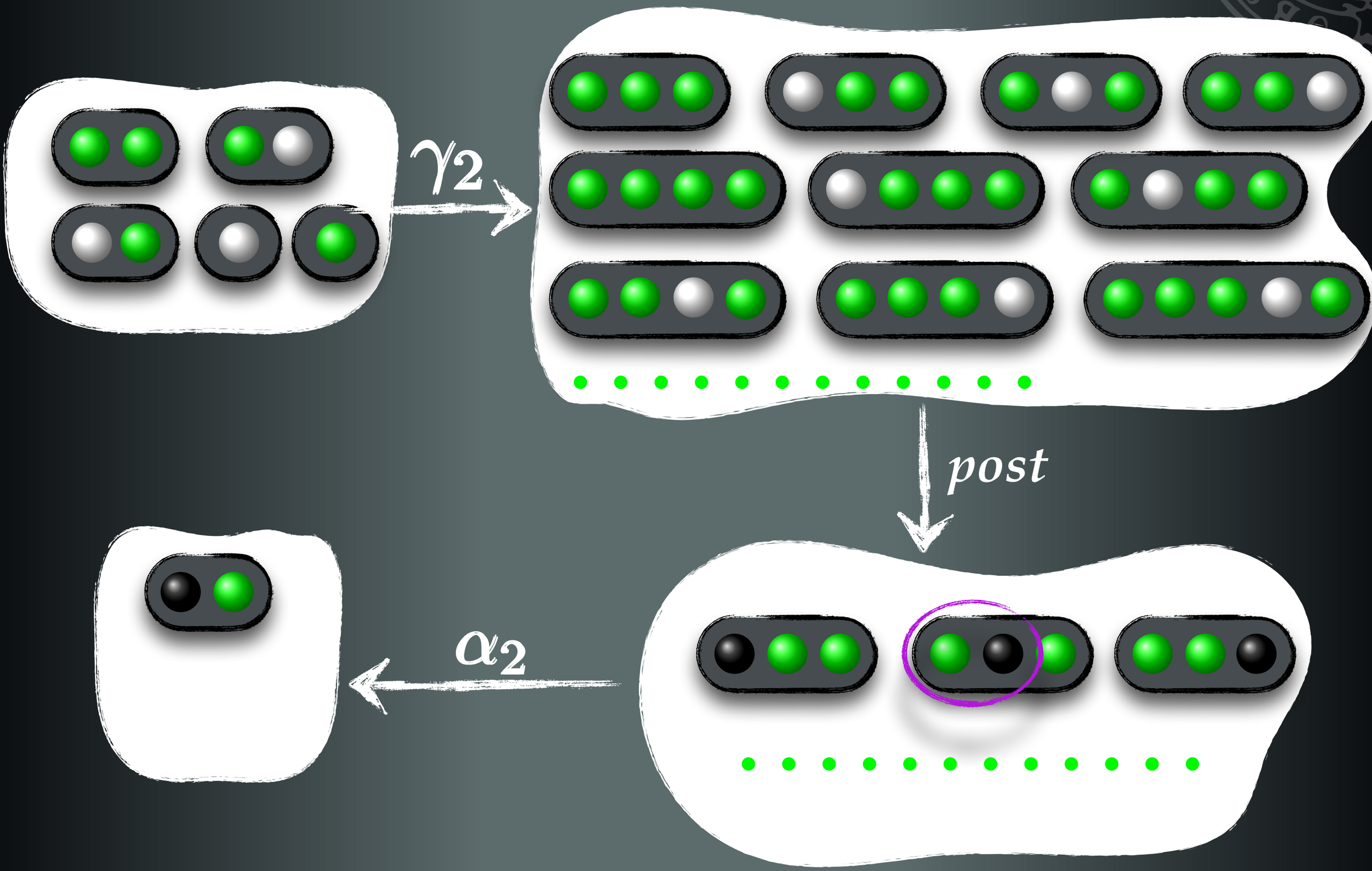
$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$



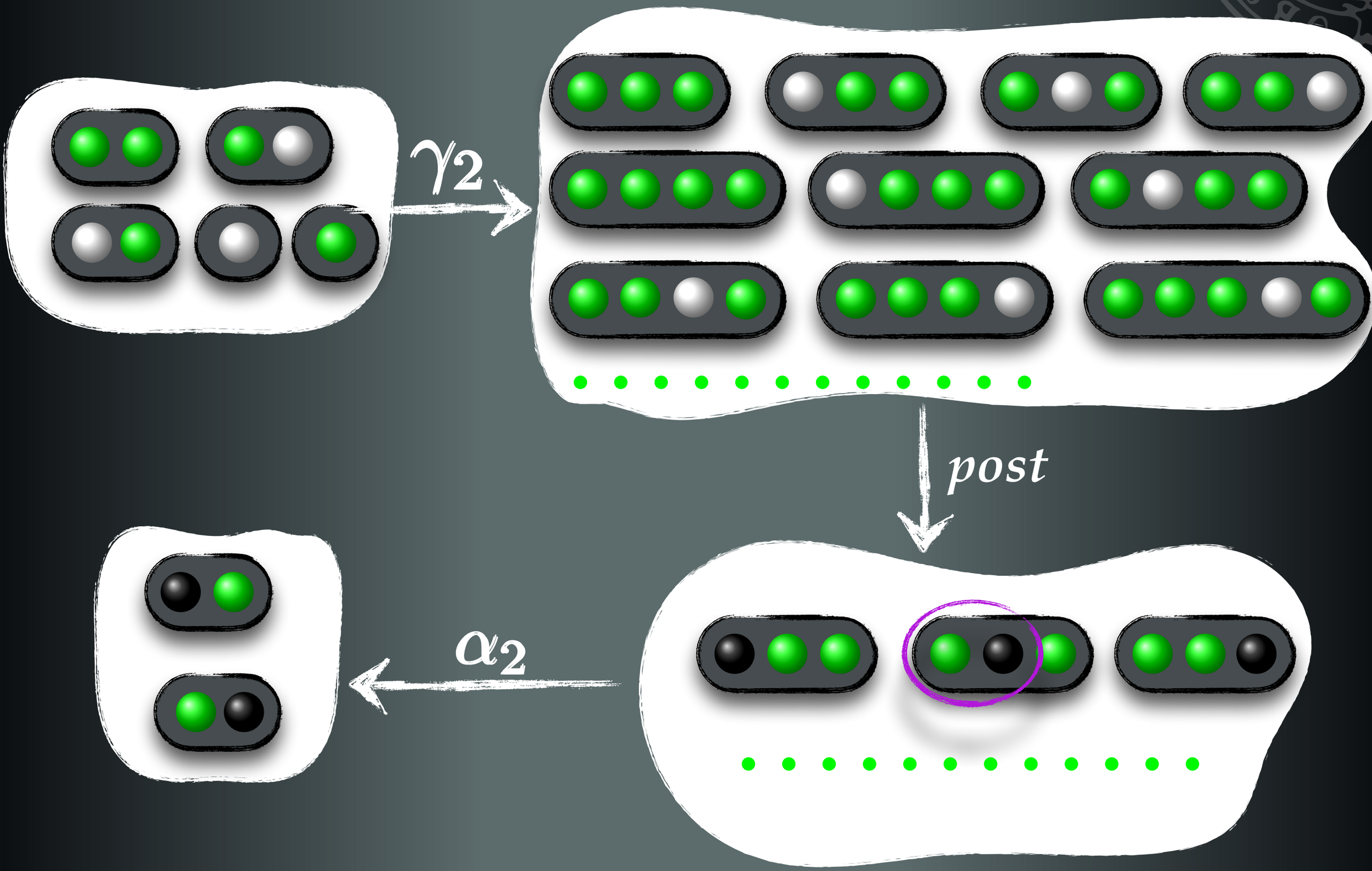
$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$



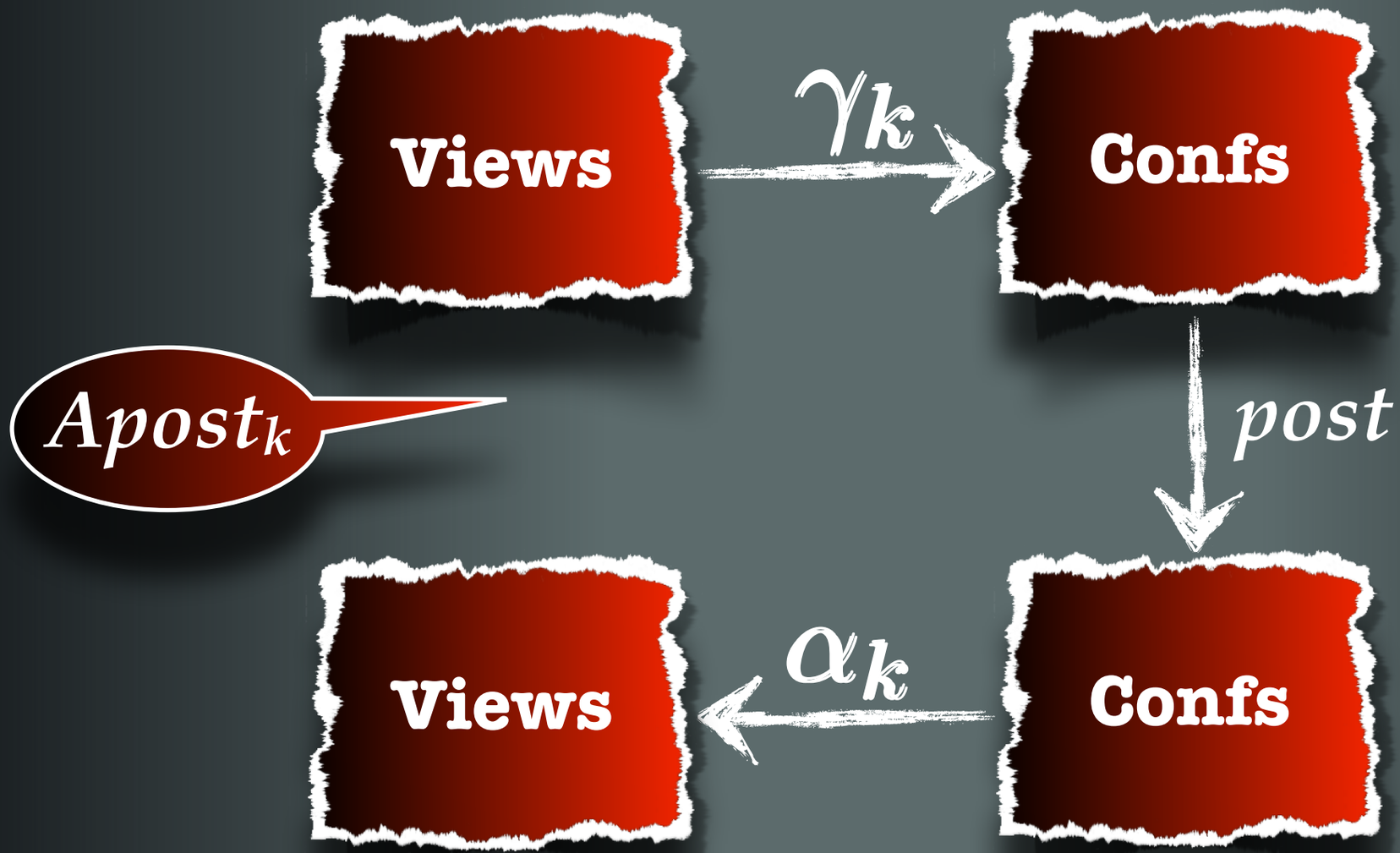
$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$



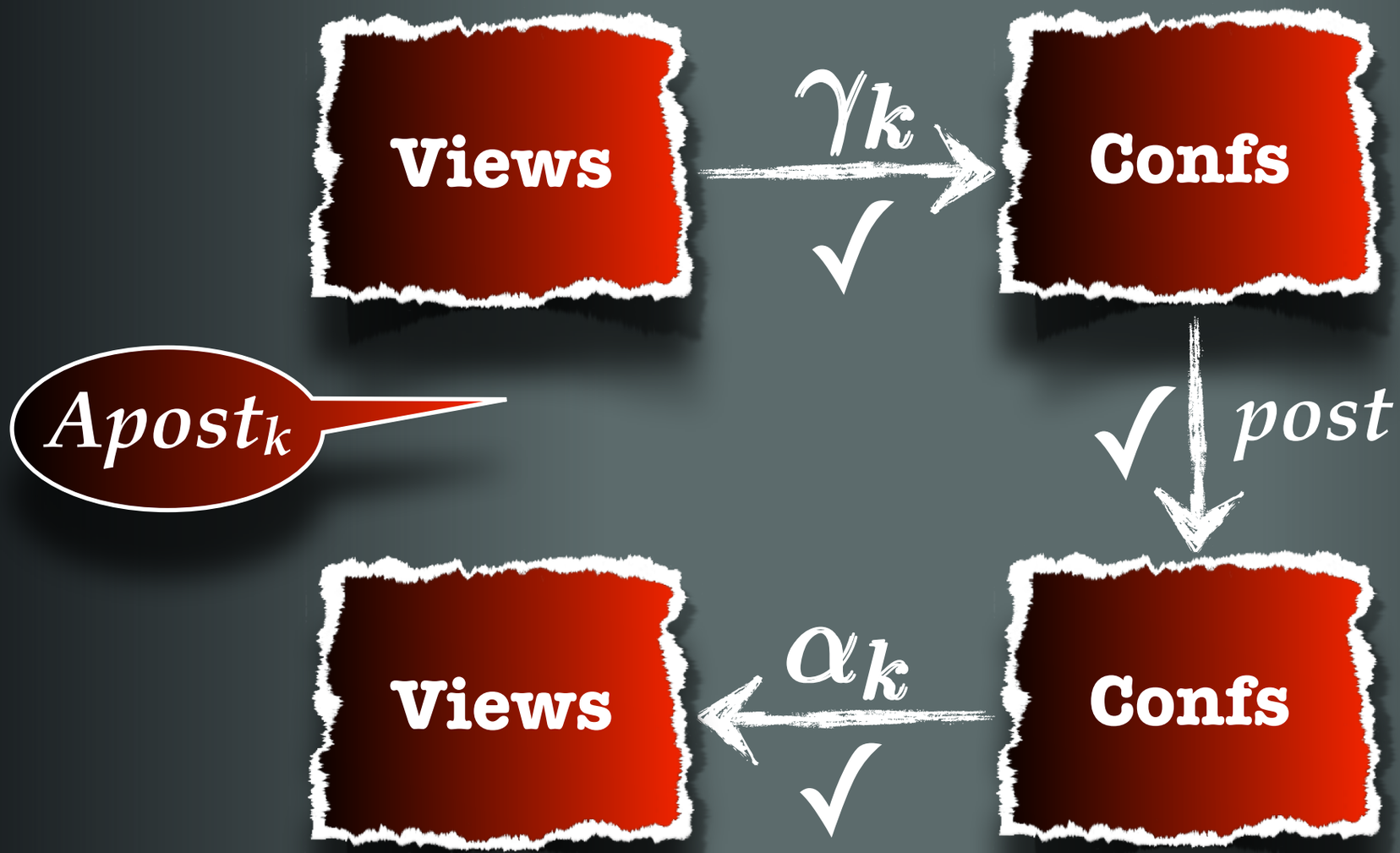
$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$



abstract post operator

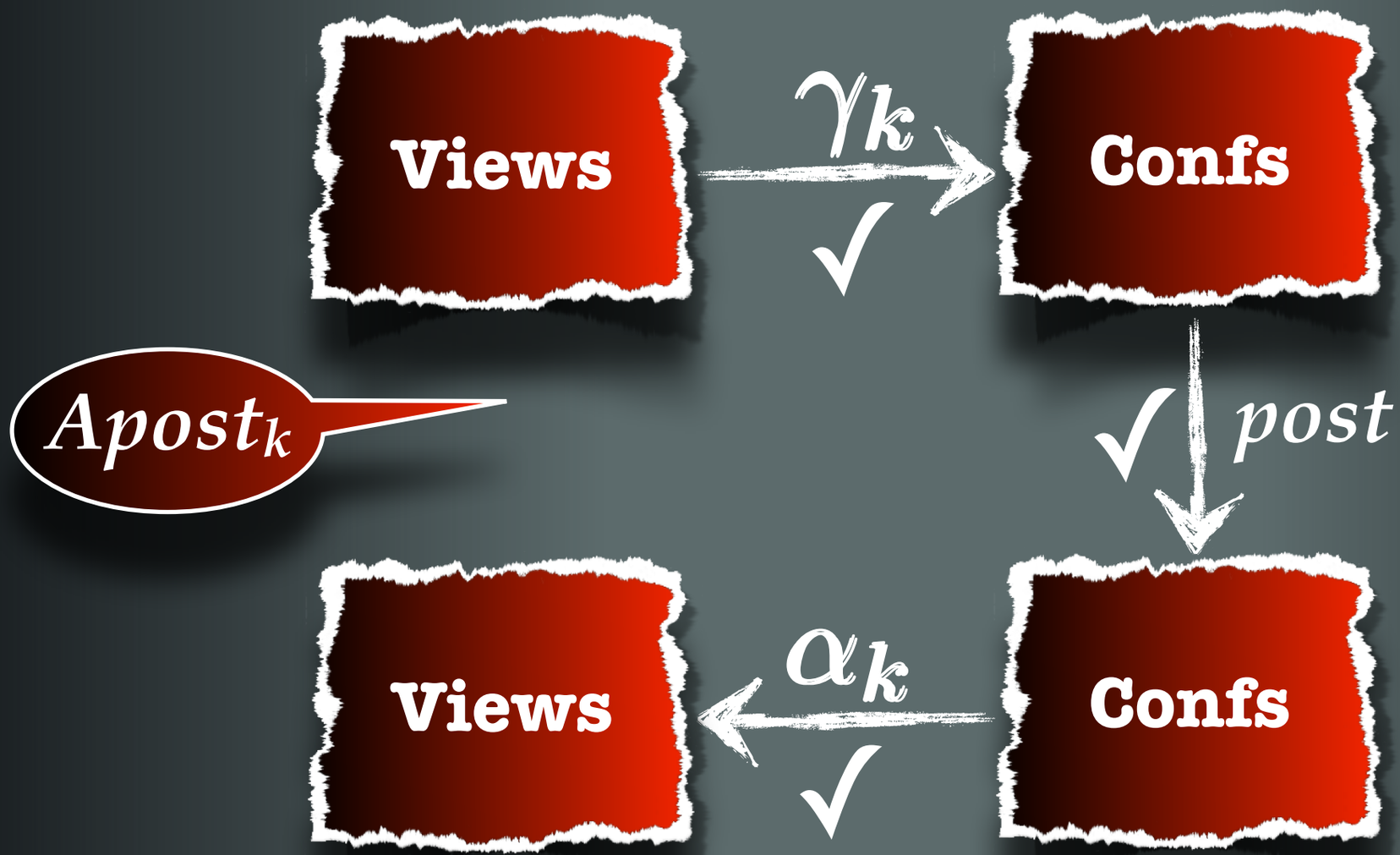


abstract post operator



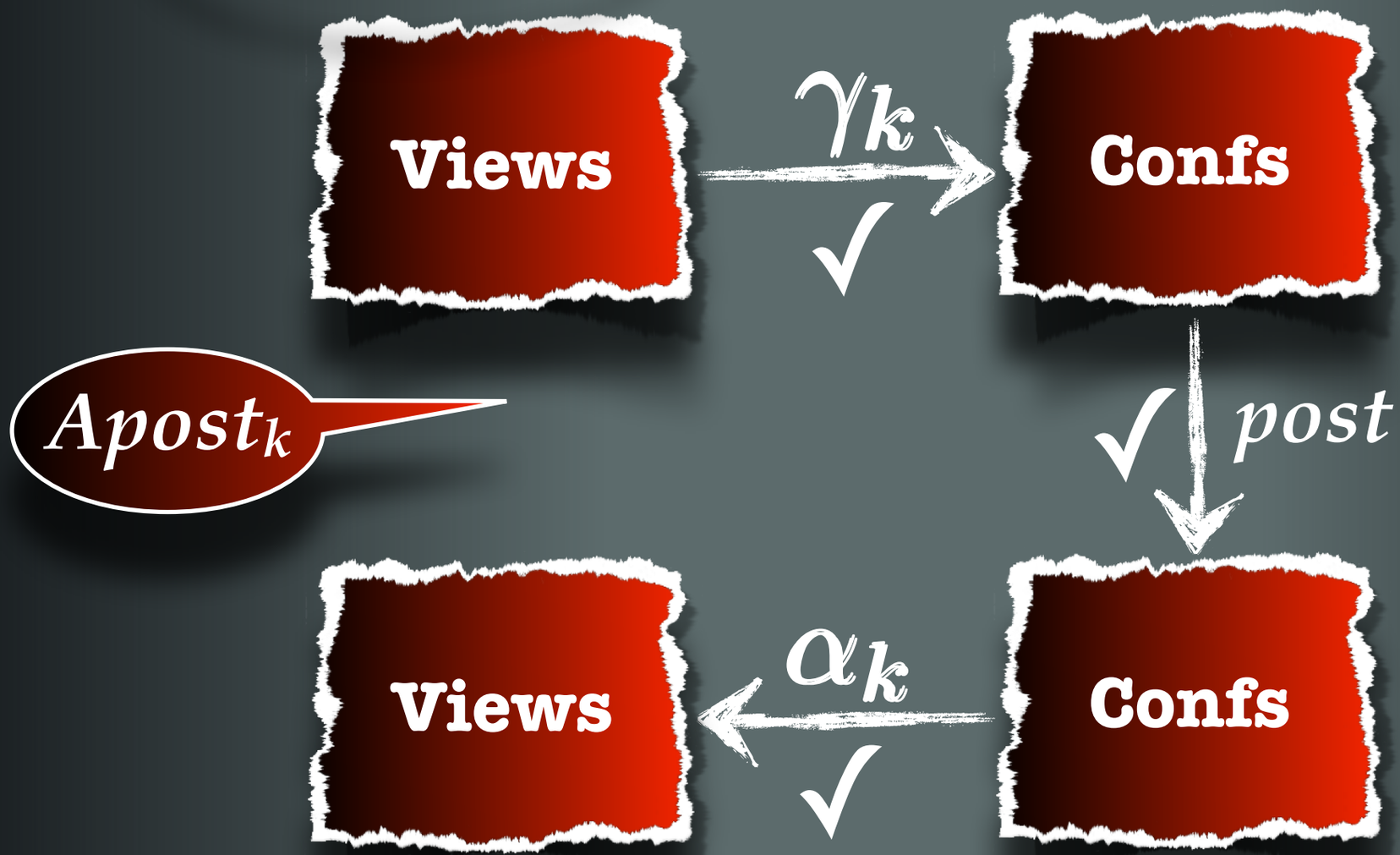
abstract post operator

$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$



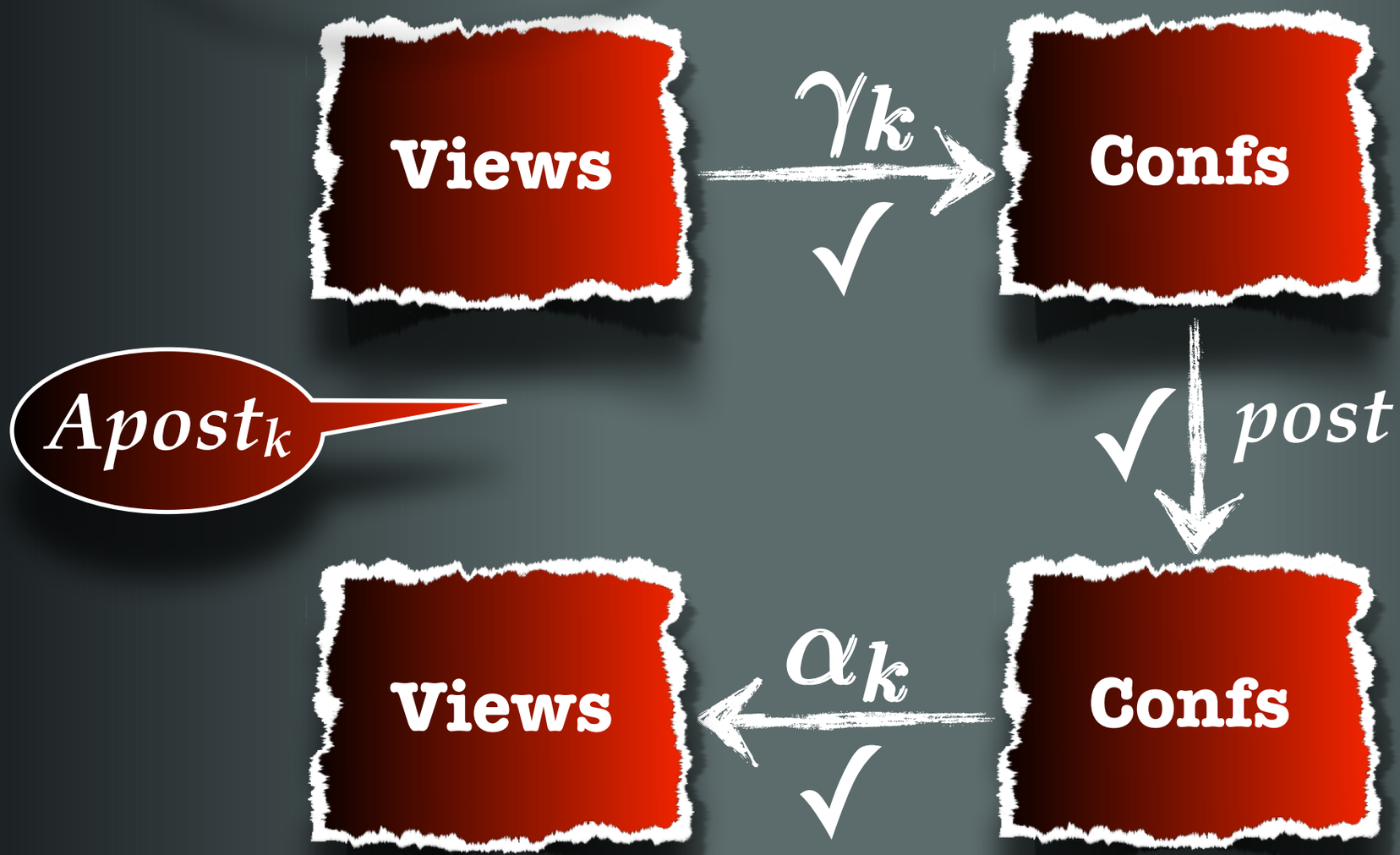
abstract post operator

$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$



abstract post operator

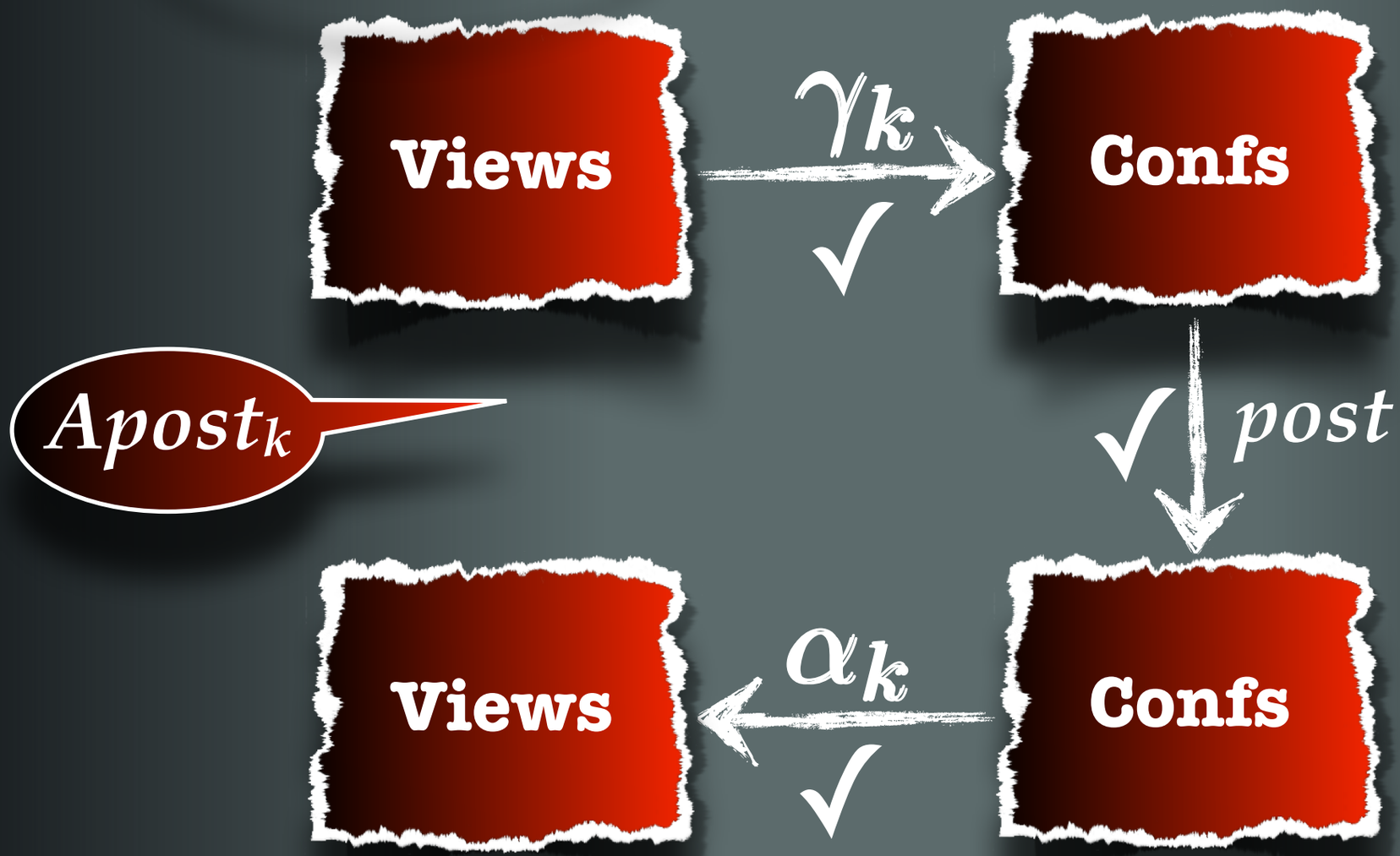
$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$



$$V_k := \mu X.\ \alpha_k(I) \cup Apost_k(X)$$

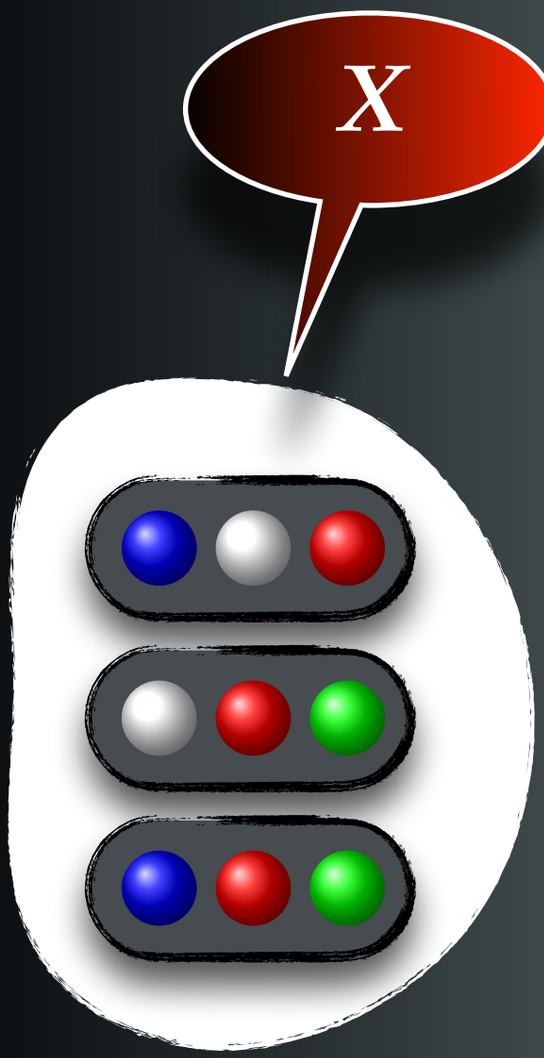
abstract post operator

$$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$$

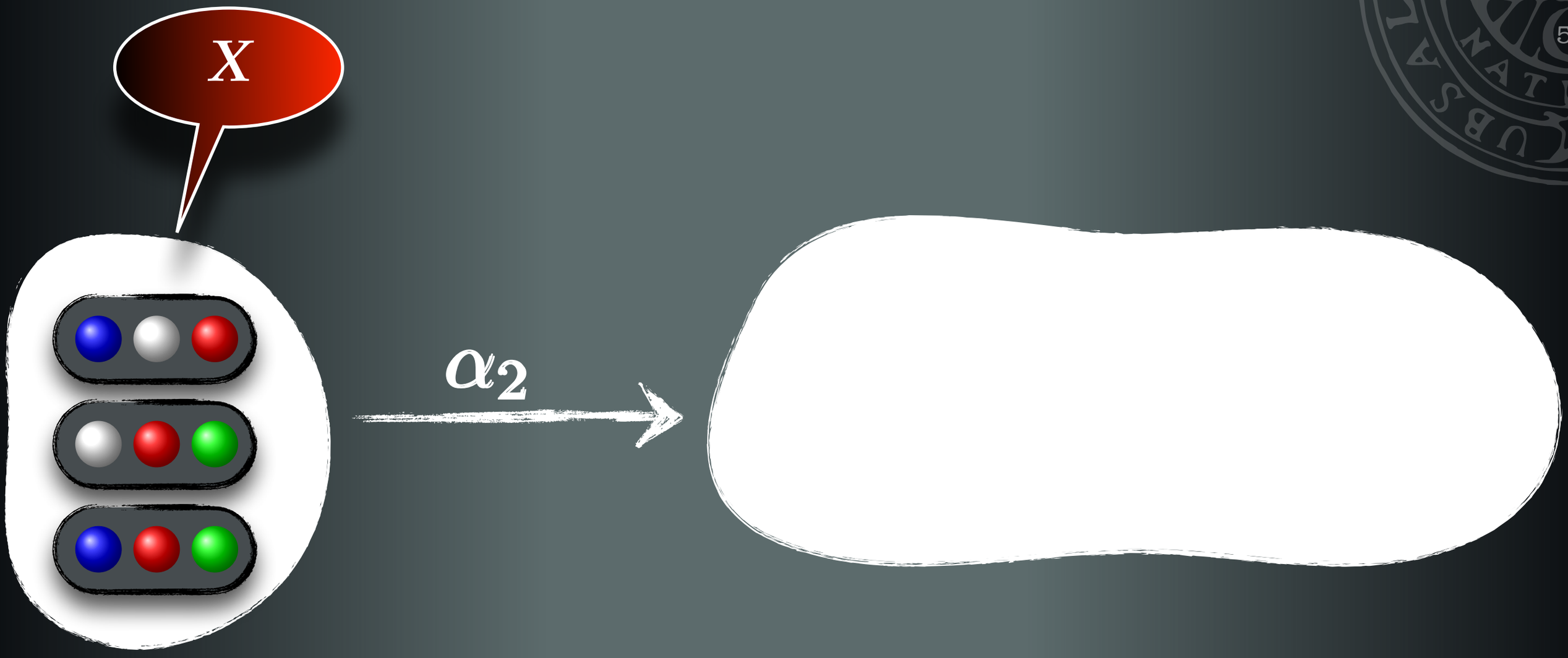


$$V_k := \mu X.\ \alpha_k(I) \cup Apost_k(X)$$

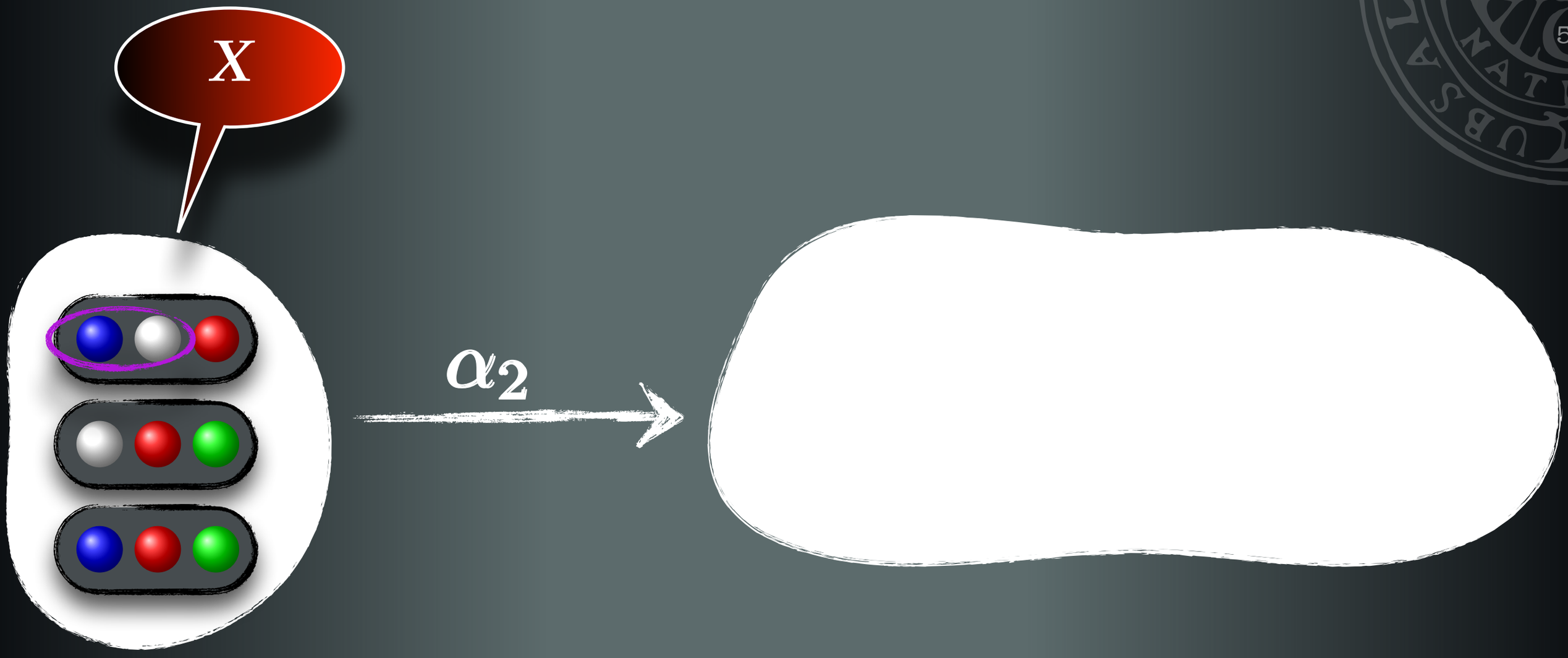
$$\gamma_k(\alpha_k(X)) \supseteq X$$



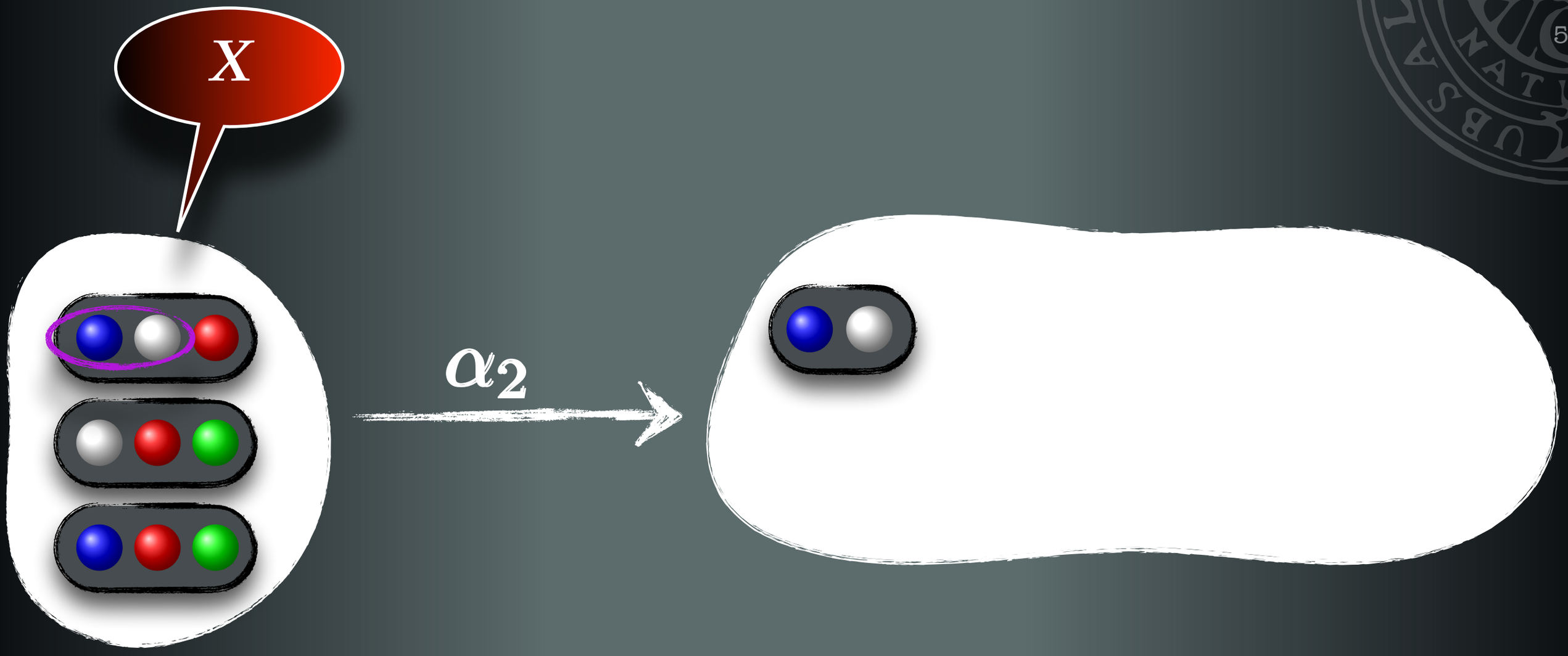
$$\gamma_k(\alpha_k(X)) \supseteq X$$



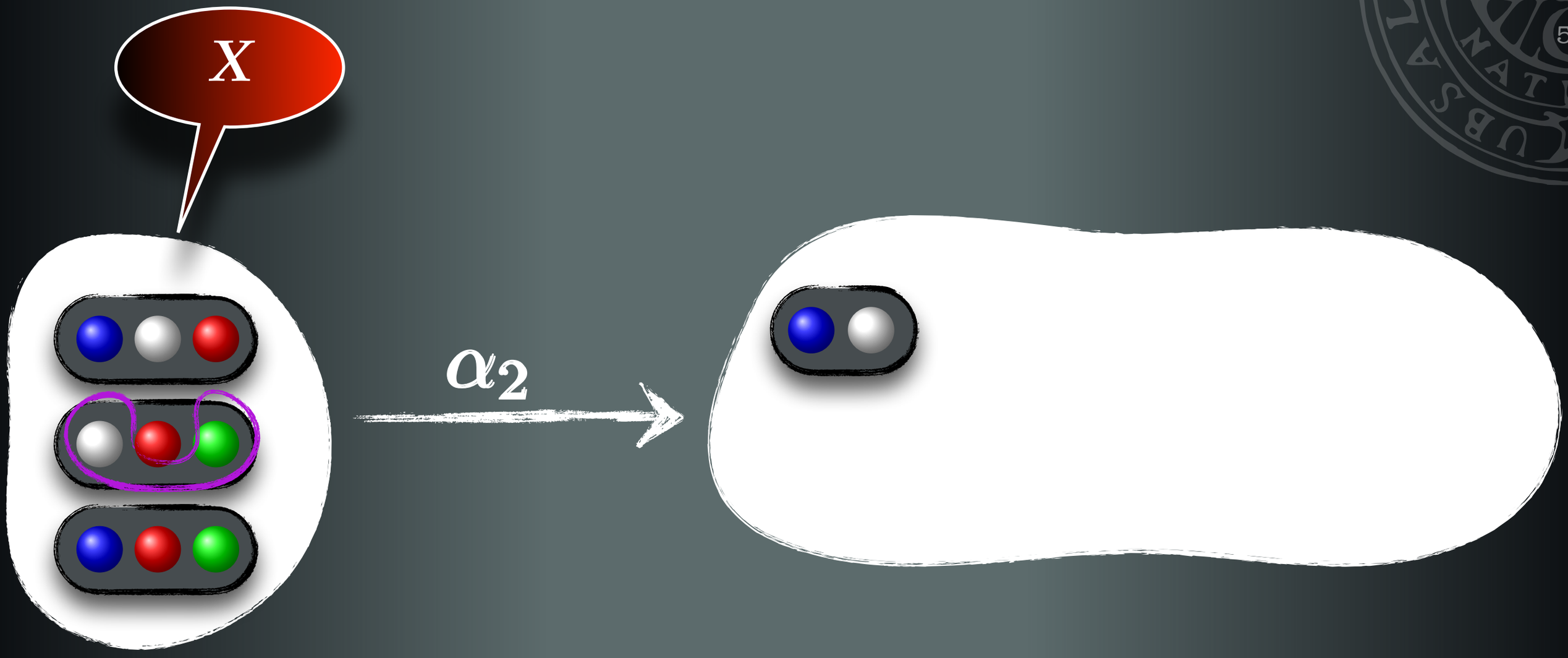
$$\gamma_k(\alpha_k(X)) \supseteq X$$



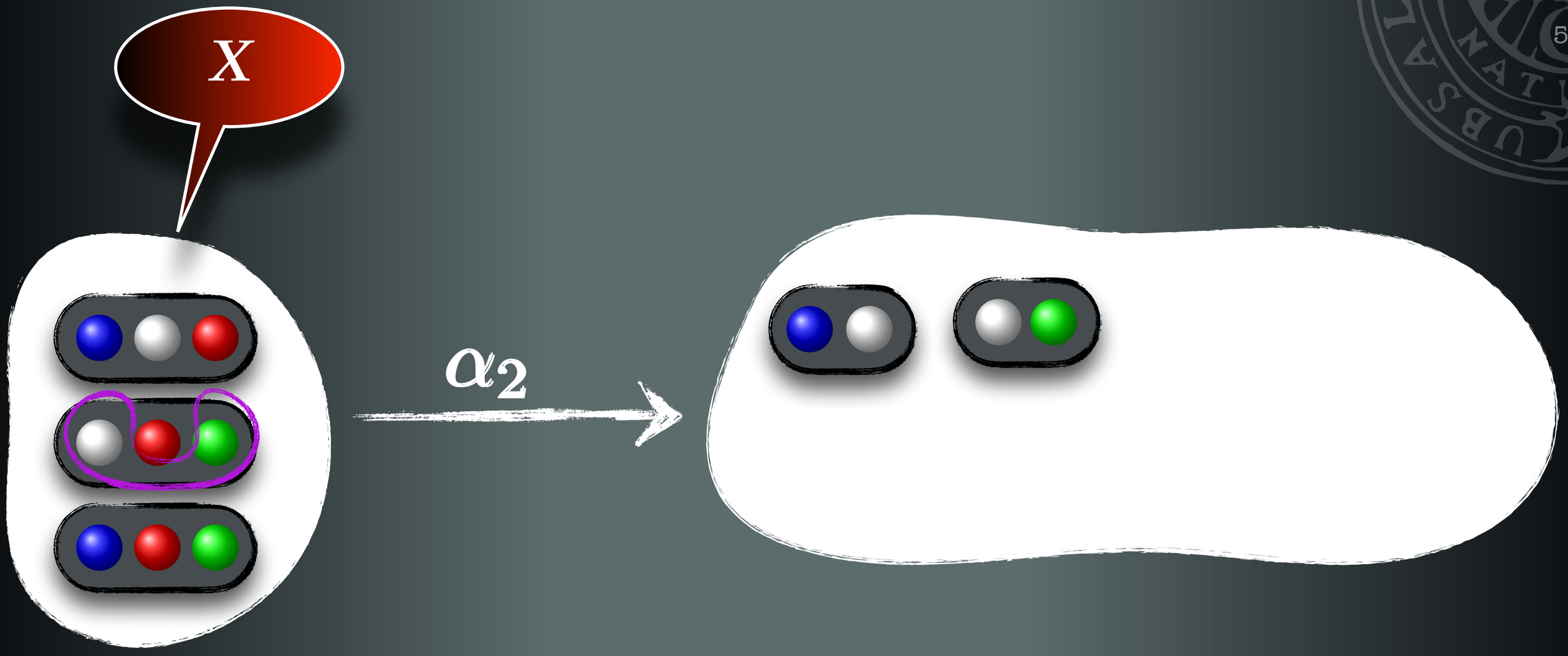
$$\gamma_k(\alpha_k(X)) \supseteq X$$



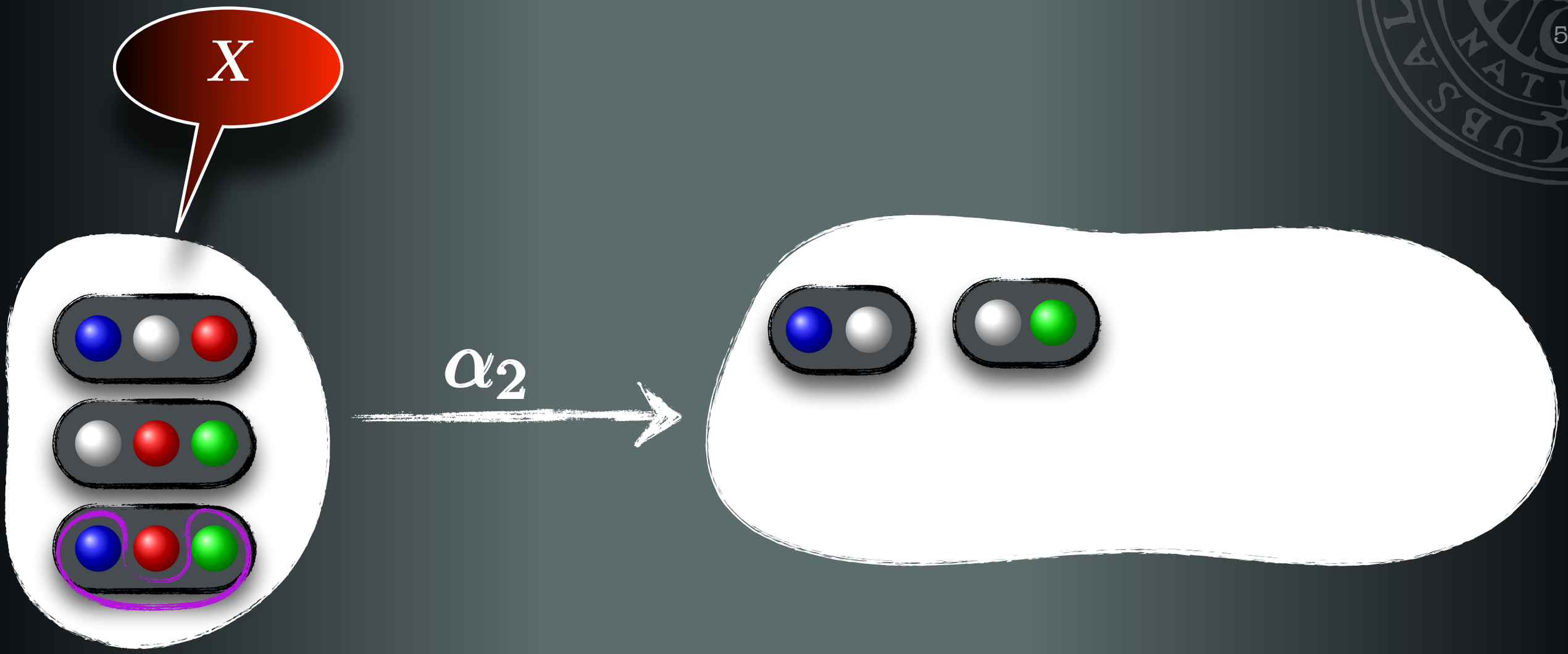
$$\gamma_k(\alpha_k(X)) \supseteq X$$



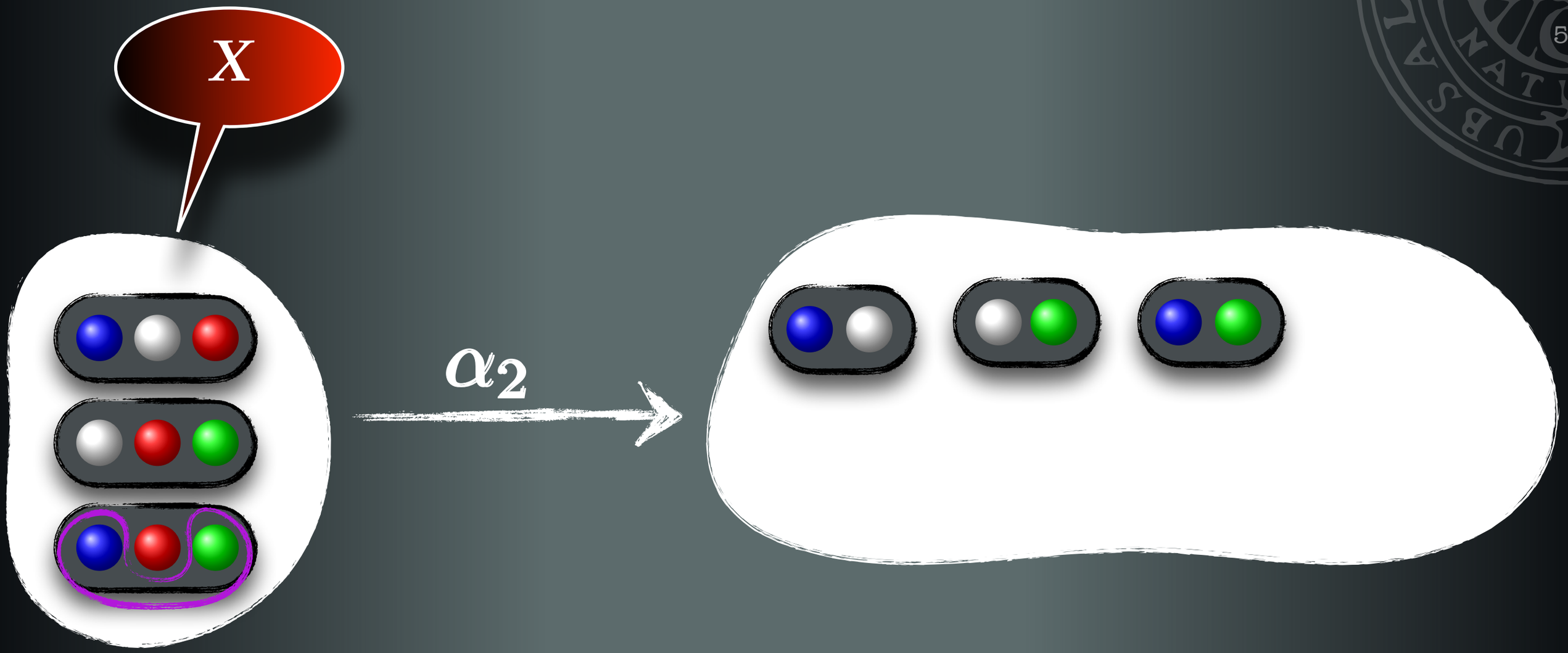
$$\gamma_k(\alpha_k(X)) \supseteq X$$



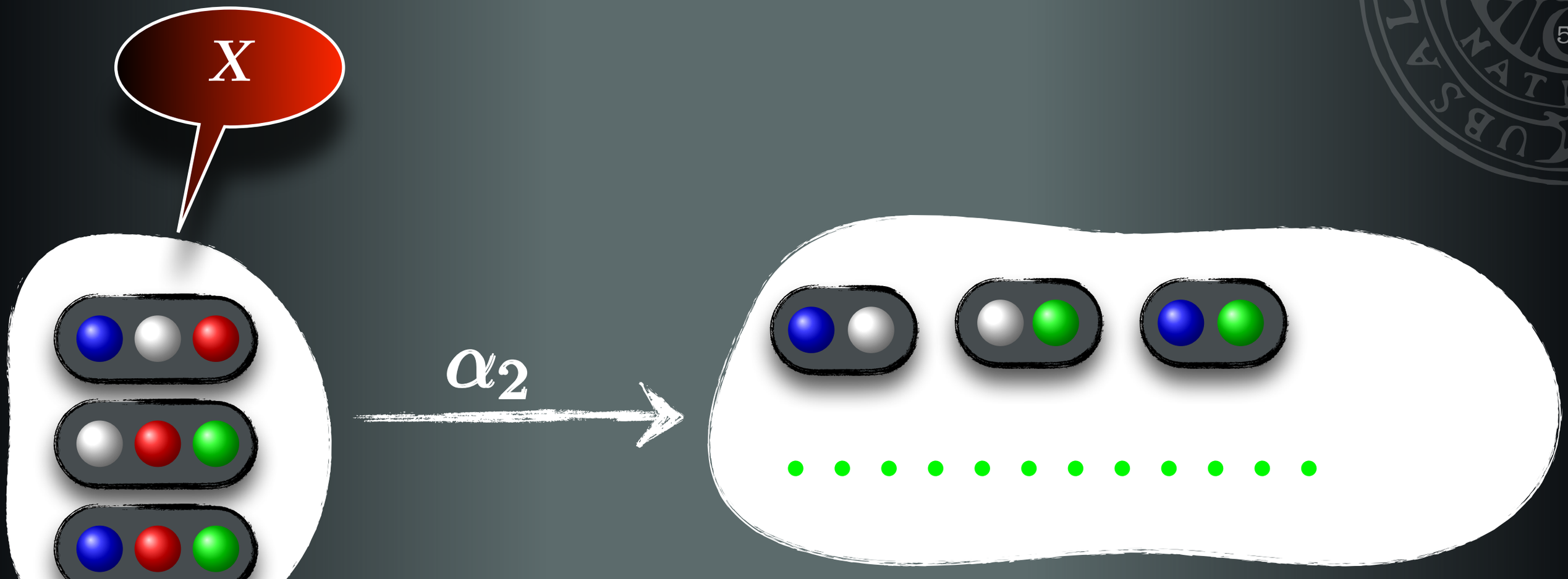
$$\gamma_k(\alpha_k(X)) \supseteq X$$



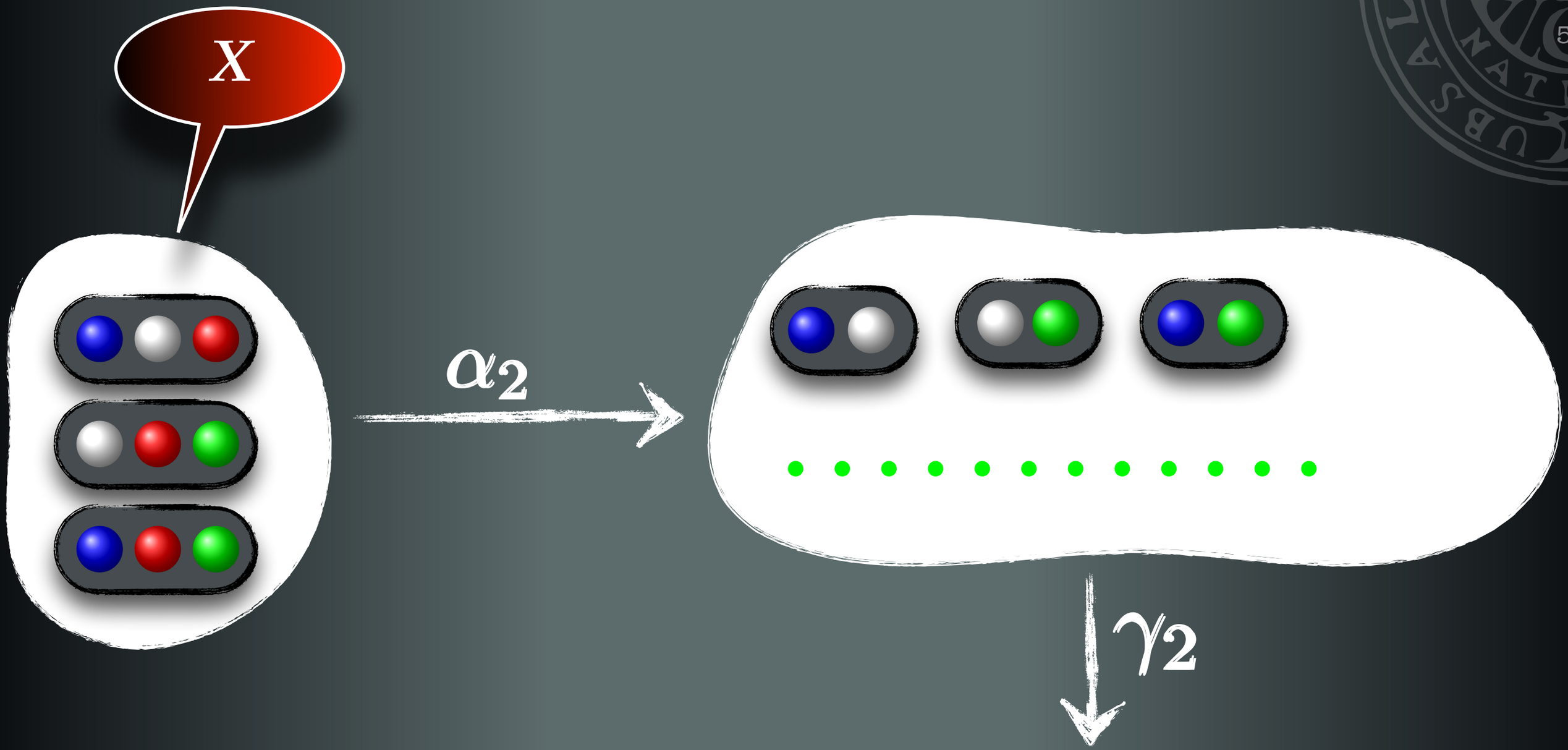
$$\gamma_k(\alpha_k(X)) \supseteq X$$



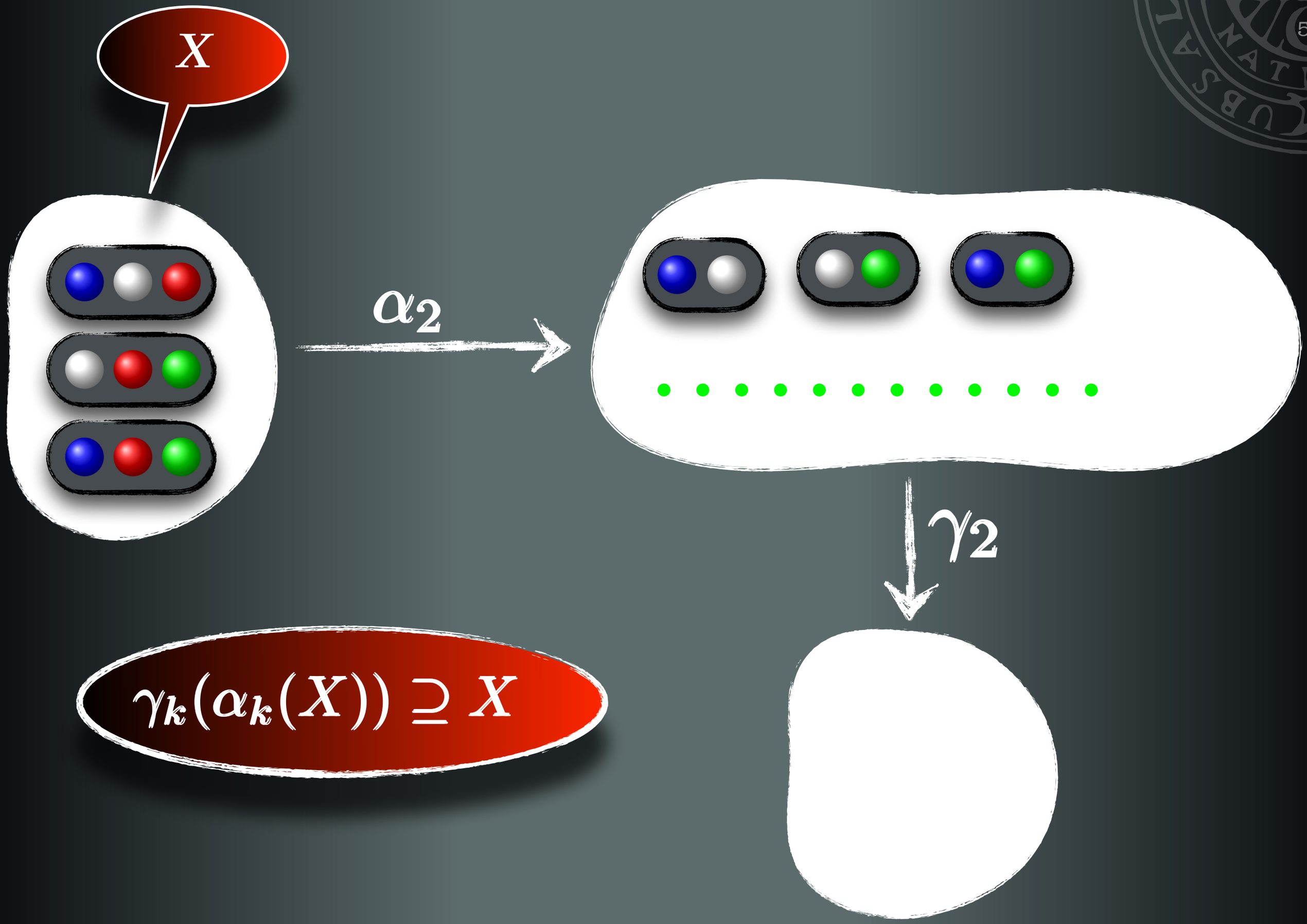
$$\gamma_k(\alpha_k(X)) \supseteq X$$



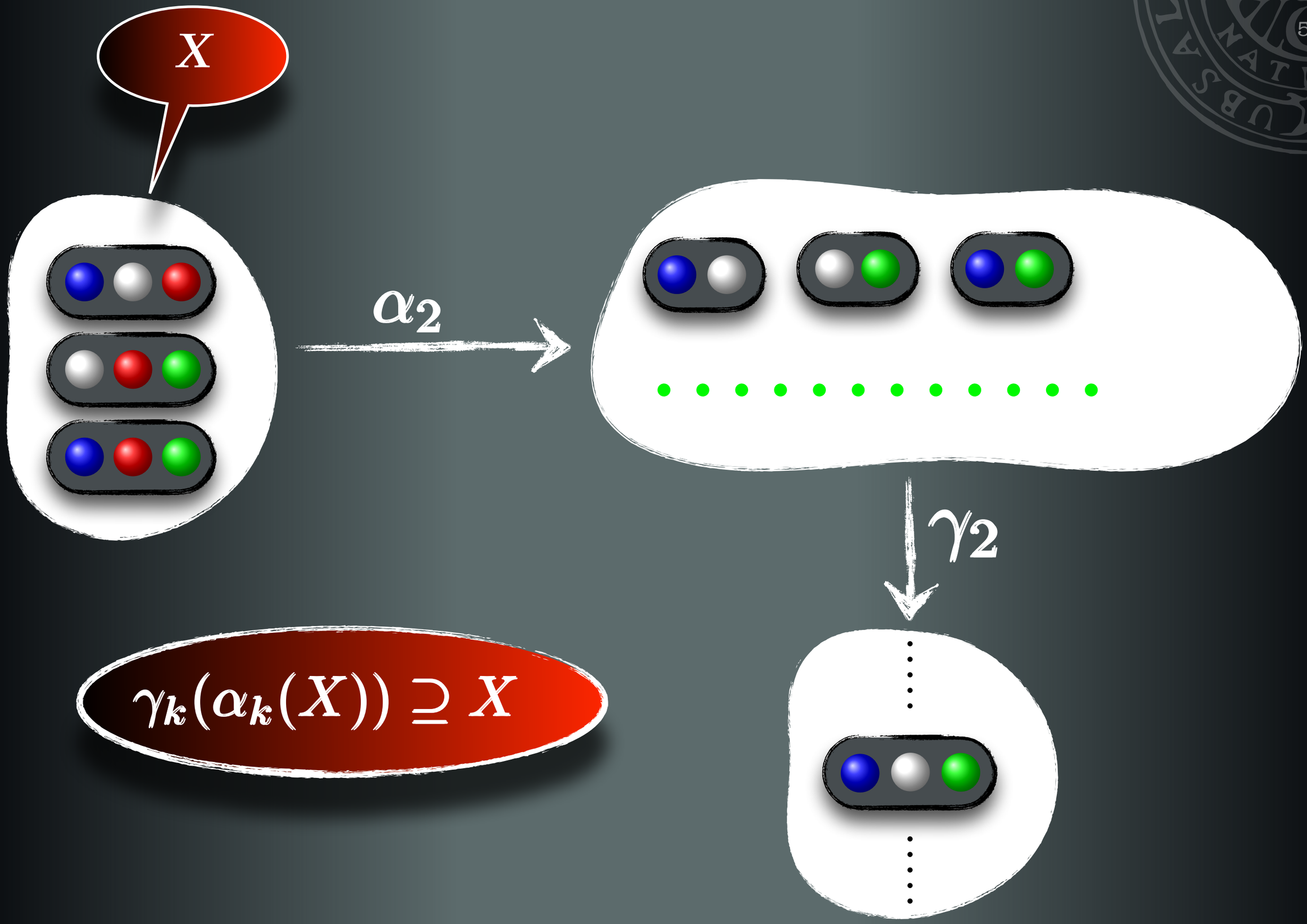
$$\gamma_k(\alpha_k(X)) \supseteq X$$



$$\gamma_k(\alpha_k(X)) \supseteq X$$



$$\gamma_k(\alpha_k(X)) \supseteq X$$


X
α_2
γ_2
$\gamma_k(\alpha_k(X)) \supseteq X$

$$\mathbb{R} \subseteq \gamma_k(V_k)$$ 1

$\gamma_k(V_k) =$

$\gamma_k(\mu X.\ \alpha_k(I) \cup Apost_k(X)) =$

$\gamma_k\ \alpha_k$ ($I \cup$ …)

$\supseteq$

$\mathbb{R} =$

$\mu X.\ I \cup post(X) =$

($I \cup$ …)

A_0 B_0

$post(\gamma_k(\alpha_k(\ \cdot\ ))) \cup$ $post(\ \cdot\ ) \cup$

A_1 B_1

$post(\gamma_k(\alpha_k(\ \cdot\ ))) \cup$ $post(\ \cdot\ ) \cup$

A_2 B_2

$post(\gamma_k(\alpha_k(\ \cdot\ ))) \cup$ $post(\ \cdot\ ) \cup$

A_3 B_3

$post(\gamma_k(\alpha_k(\ \cdot\ ))) \cup$ $post(\ \cdot\ ) \cup$

$B_i \subseteq A_i$

induction on i

$$B_i \subseteq A_i \subseteq \gamma_k(\alpha_k(A_i))$$

$$B_{i+1} = post(B_i) \subseteq post(\gamma_k(\alpha_k(A_i))) = A_{i+1}$$

$\mathbb{R} \subseteq \gamma_k(V_k)$ 1

$\gamma_k(V_k) =$

$\gamma_k(\mu X.\ \alpha_k(I) \cup Apost_k(X)) =$

$\gamma_k\ \alpha_k$

$I \cup$

A_0

$post(\gamma_k(\alpha_k(\ \cdot\ ))) \cup$

A_1

$post(\gamma_k(\alpha_k(\ \cdot\ ))) \cup$

A_2

$post(\gamma_k(\alpha_k(\ \cdot\ ))) \cup$

A_3

$post(\gamma_k(\alpha_k(\ \cdot\ ))) \cup$

$\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet$

$\supseteq$

$\mathbb{R} =$

$\mu X.\ I \cup post(X) =$

$I \cup$

B_0

$post(\ \cdot\ ) \cup$

B_1

$post(\ \cdot\ ) \cup$

B_2

$post(\ \cdot\ ) \cup$

B_3

$post(\ \cdot\ ) \cup$

$\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet$

$B_i \subseteq A_i$

induction on i

$$B_i \subseteq A_i \subseteq \gamma_k(\alpha_k(A_i))$$

$$B_{i+1} = post(B_i) \subseteq post(\gamma_k(\alpha_k(A_i))) = A_{i+1}$$

$$\mathbb{R} \subseteq \gamma_k(V_k)$$ 1

$\gamma_k(V_k) =$

$\gamma_k(\mu X.\ \alpha_k(I) \cup Apost_k(X)) =$

$\mathbb{R} =$

$\mu X.\ I \cup post(X) =$

$$\gamma_k\,\alpha_k\left(\begin{array}{l} I \cup \\ post(\gamma_k(\alpha_k(\ \cdot\ ))) \cup \\ post(\gamma_k(\alpha_k(\ \cdot\ ))) \cup \\ post(\gamma_k(\alpha_k(\ \cdot\ ))) \cup \\ post(\gamma_k(\alpha_k(\ \cdot\ ))) \cup \\ \bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet \end{array}\right) \supseteq \left(\begin{array}{l} I \cup \\ post(\ \cdot\ ) \cup \\ post(\ \cdot\ ) \cup \\ post(\ \cdot\ ) \cup \\ post(\ \cdot\ ) \cup \\ \bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet \end{array}\right)$$

A_0 A_1 A_2 A_3 — B_0 B_1 B_2 B_3

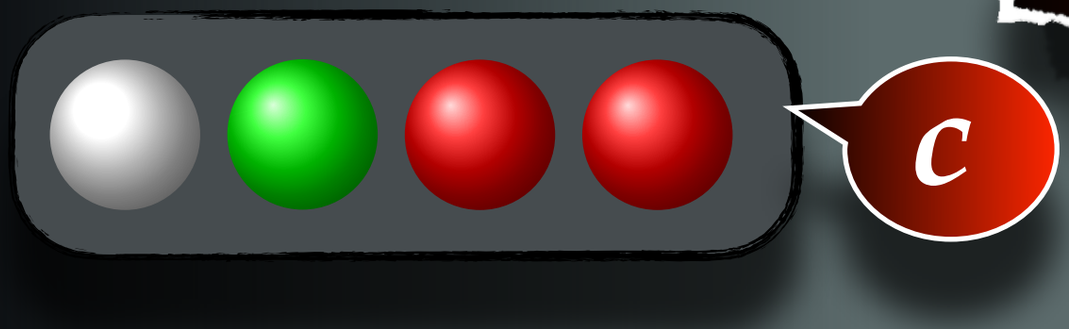
$B_i \subseteq A_i$

induction on i

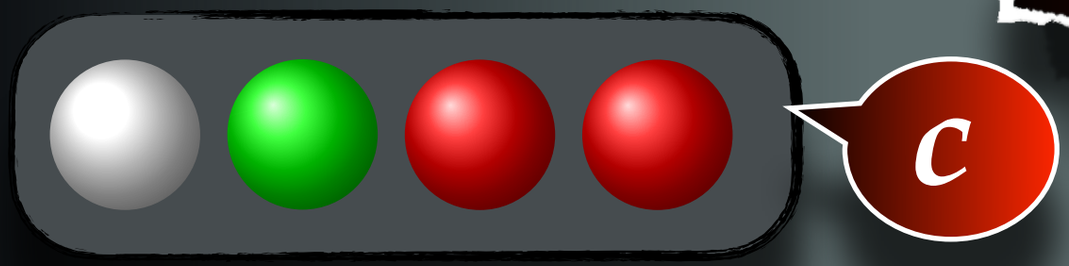
$$B_i \subseteq A_i \subseteq \gamma_k(\alpha_k(A_i))$$

$$B_{i+1} = post(B_i) \subseteq post(\gamma_k(\alpha_k(A_i))) = A_{i+1}$$

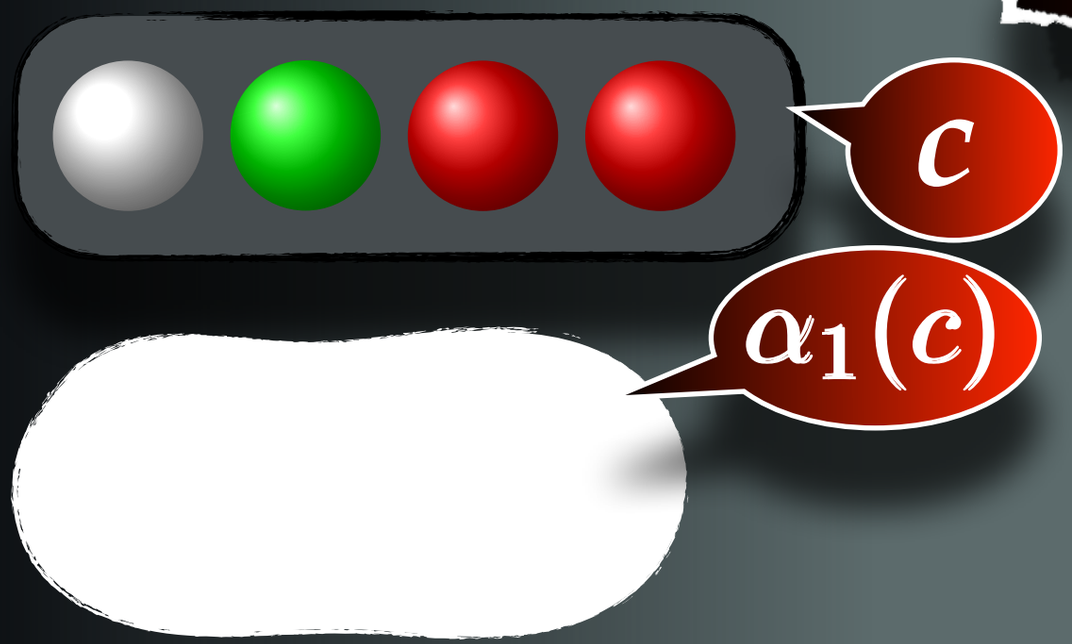
$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$



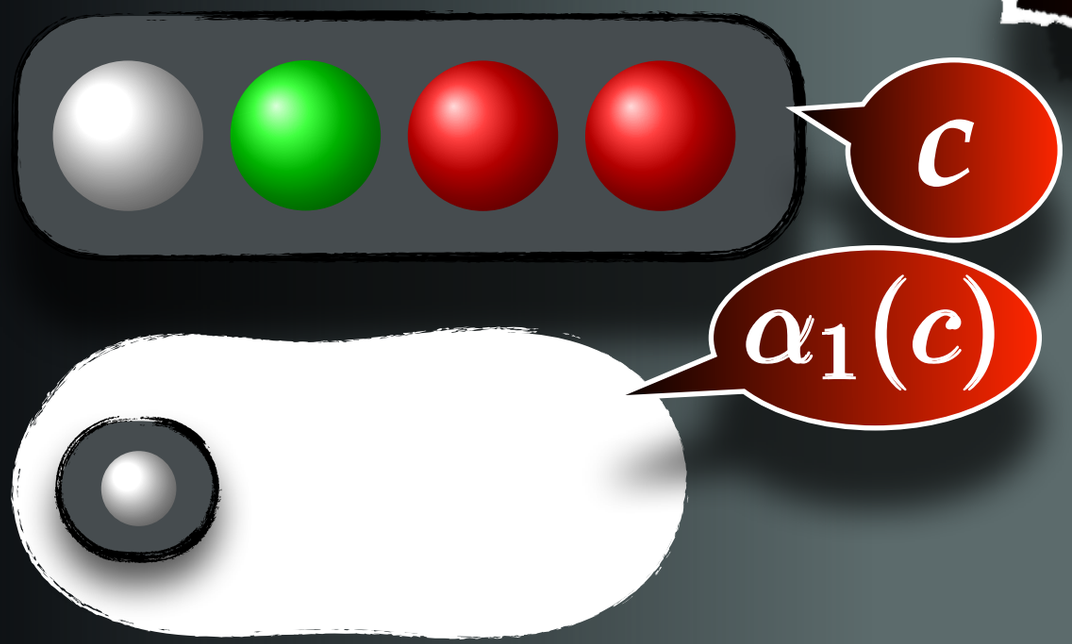
$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$



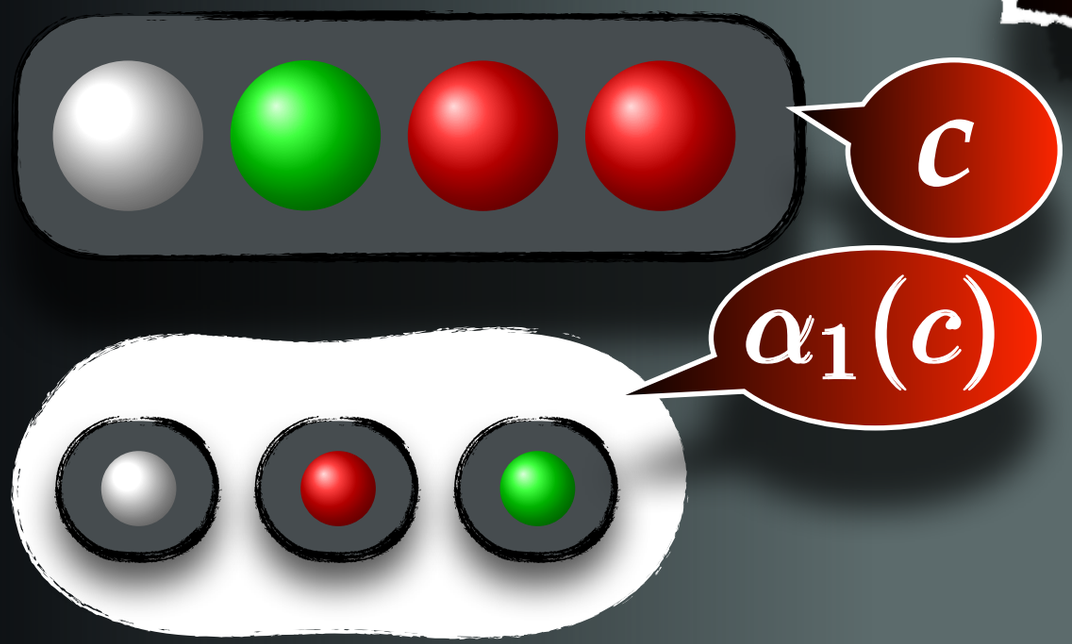
$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$



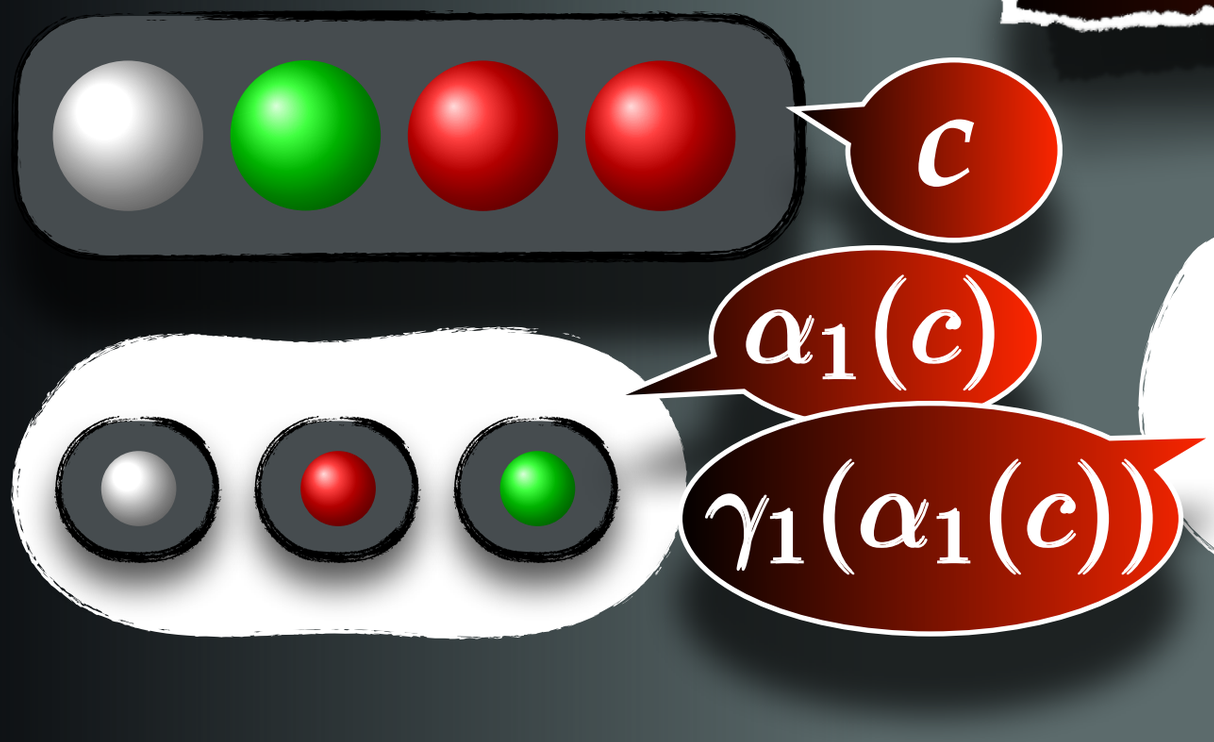
$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$



$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$



$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$

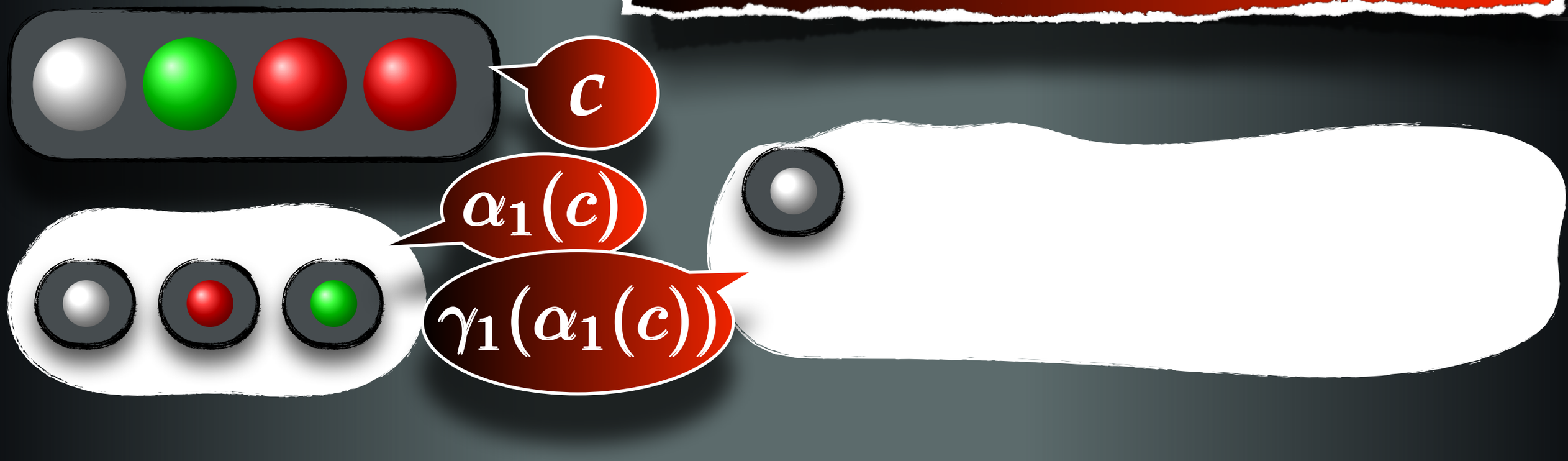


c

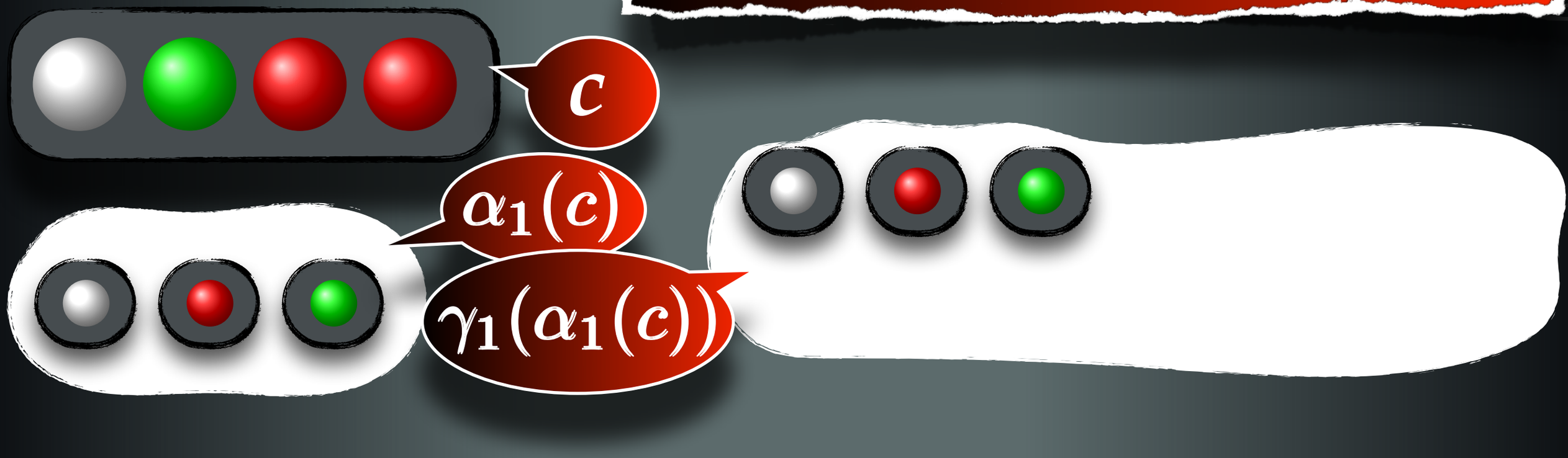
$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

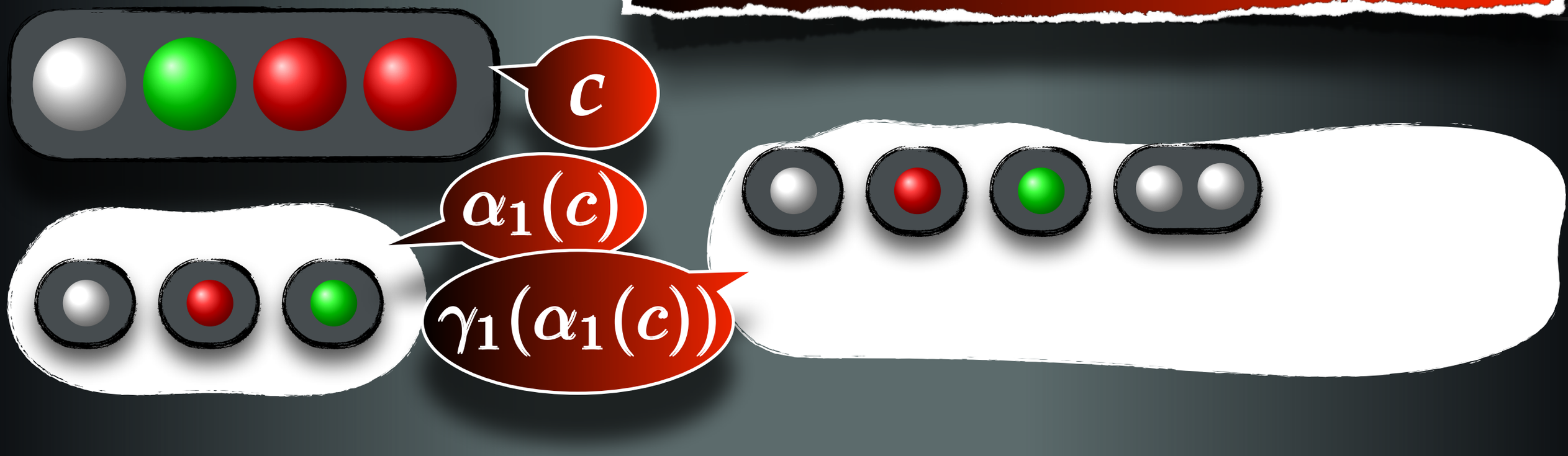
$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$



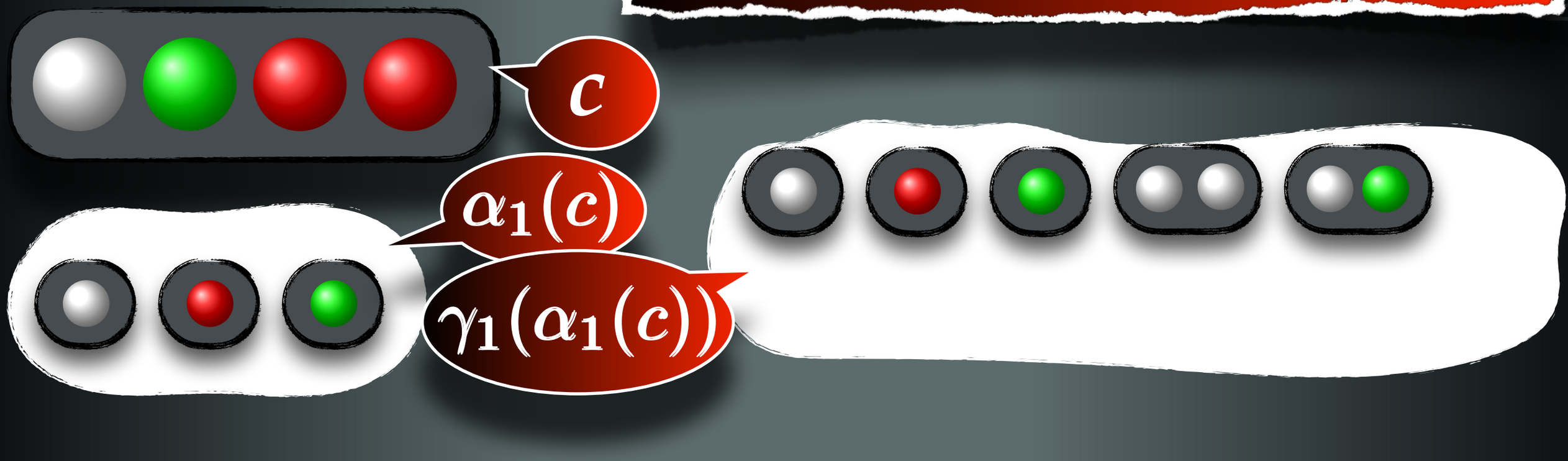
$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$



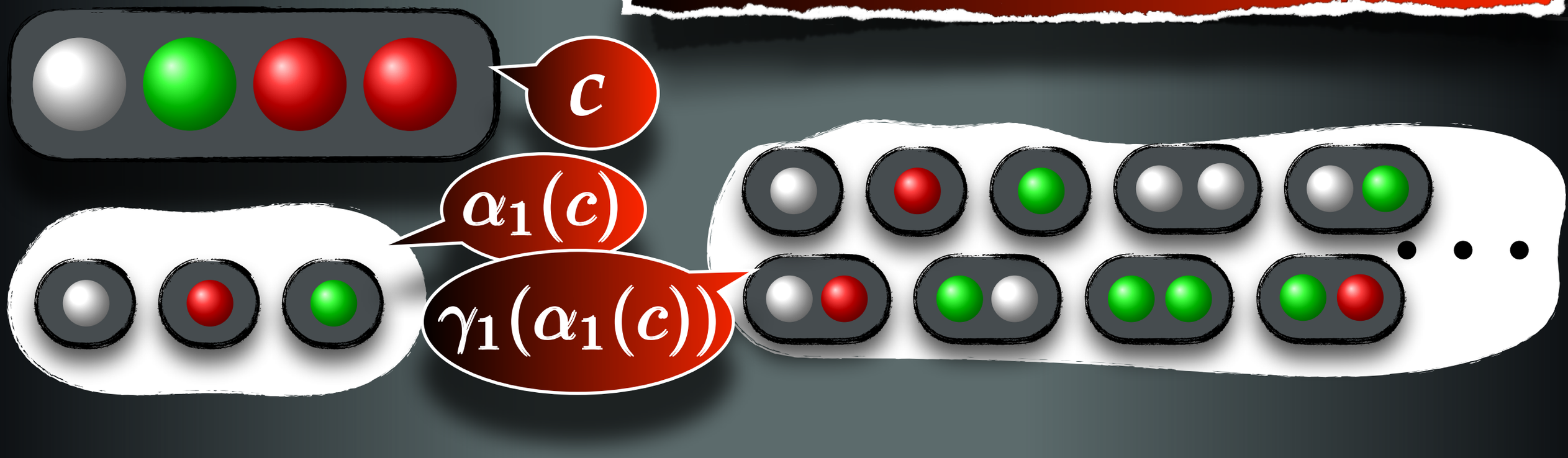
$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$



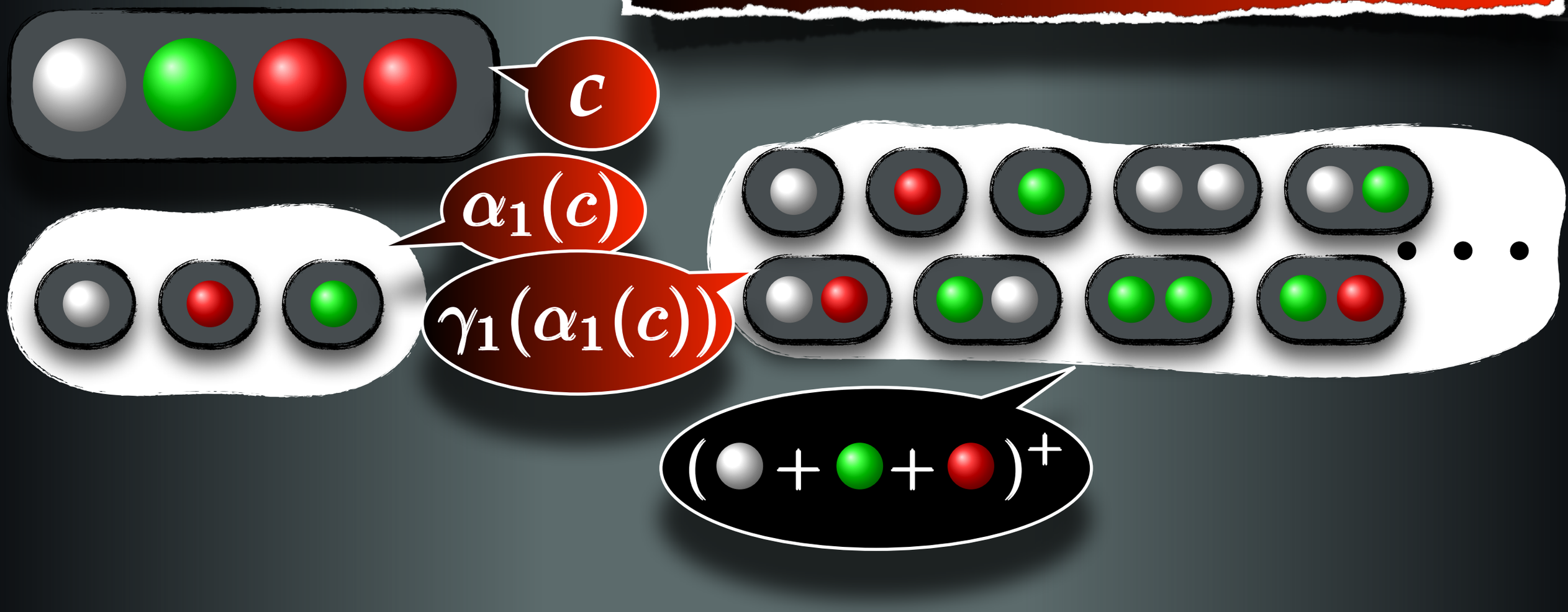
$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$



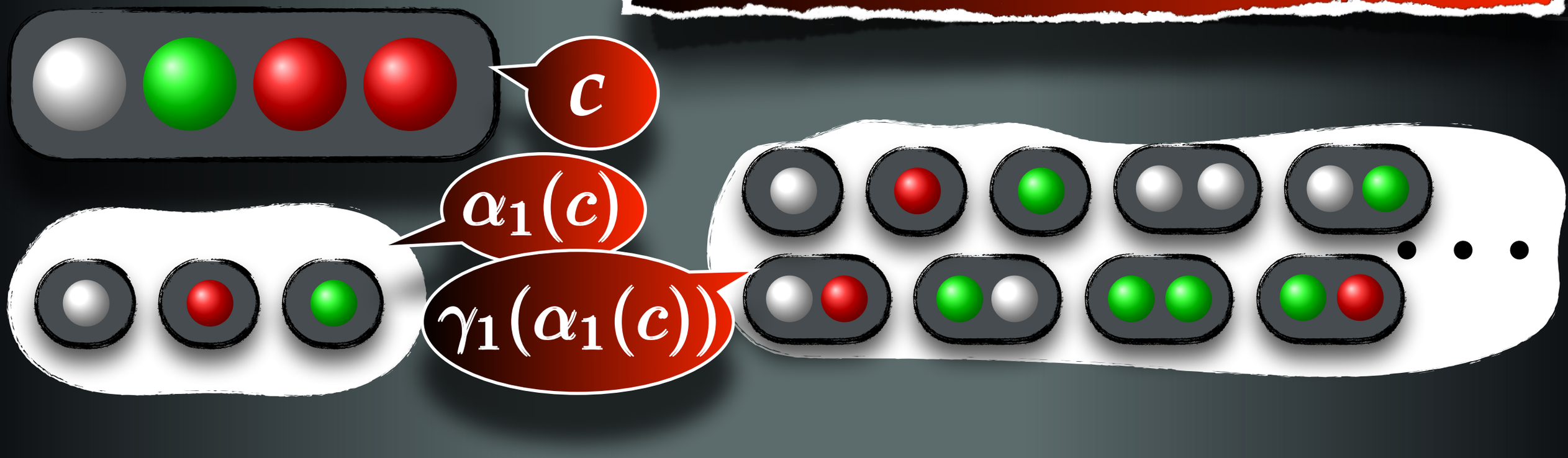
$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$



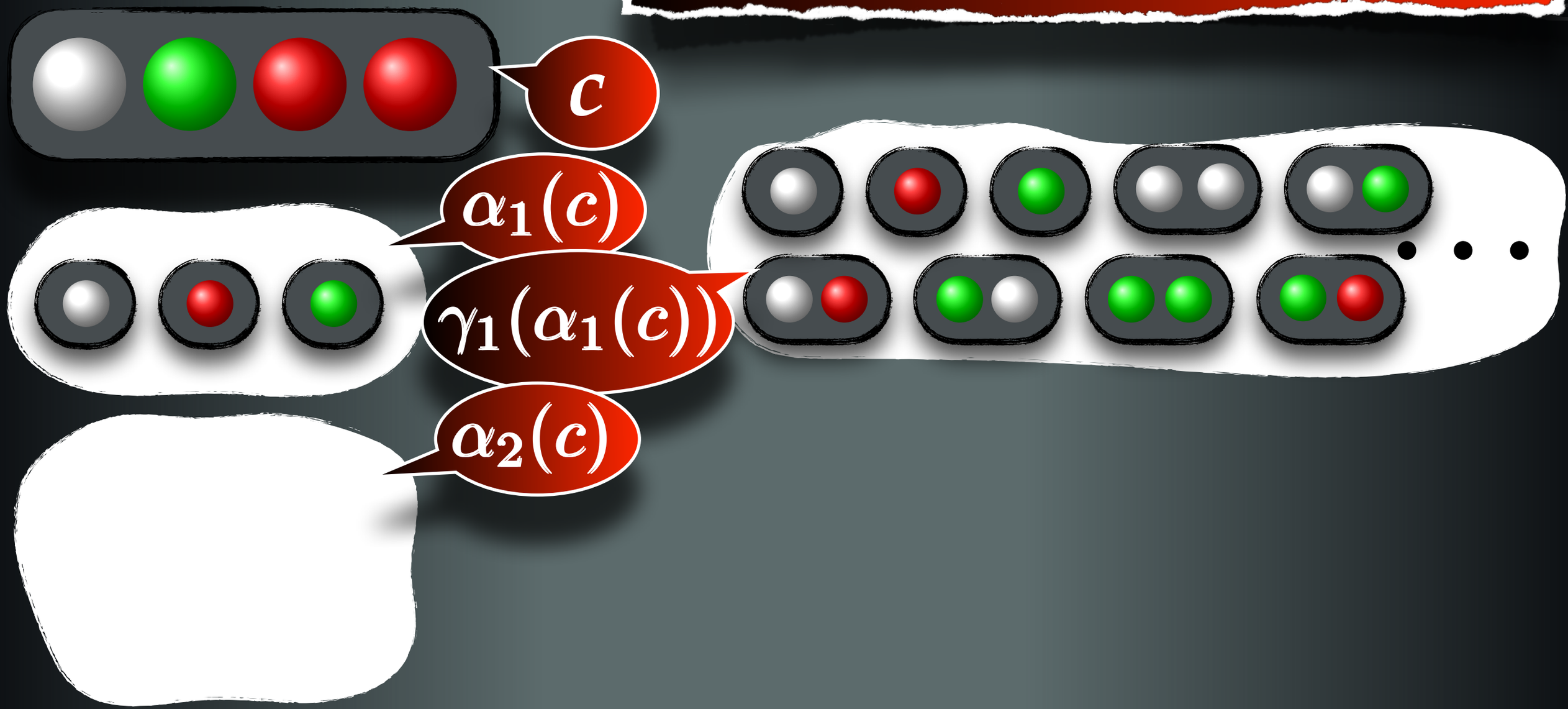
$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$



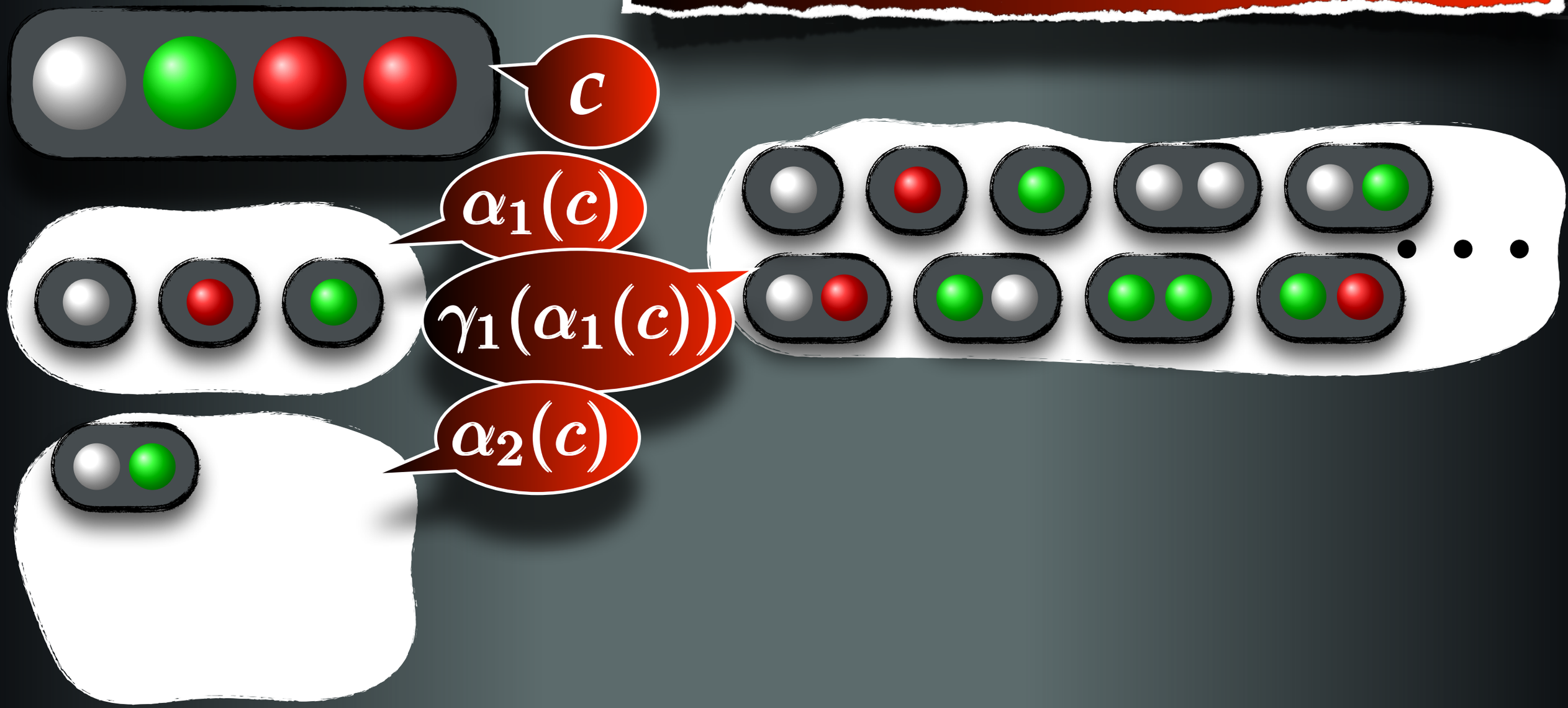
$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$



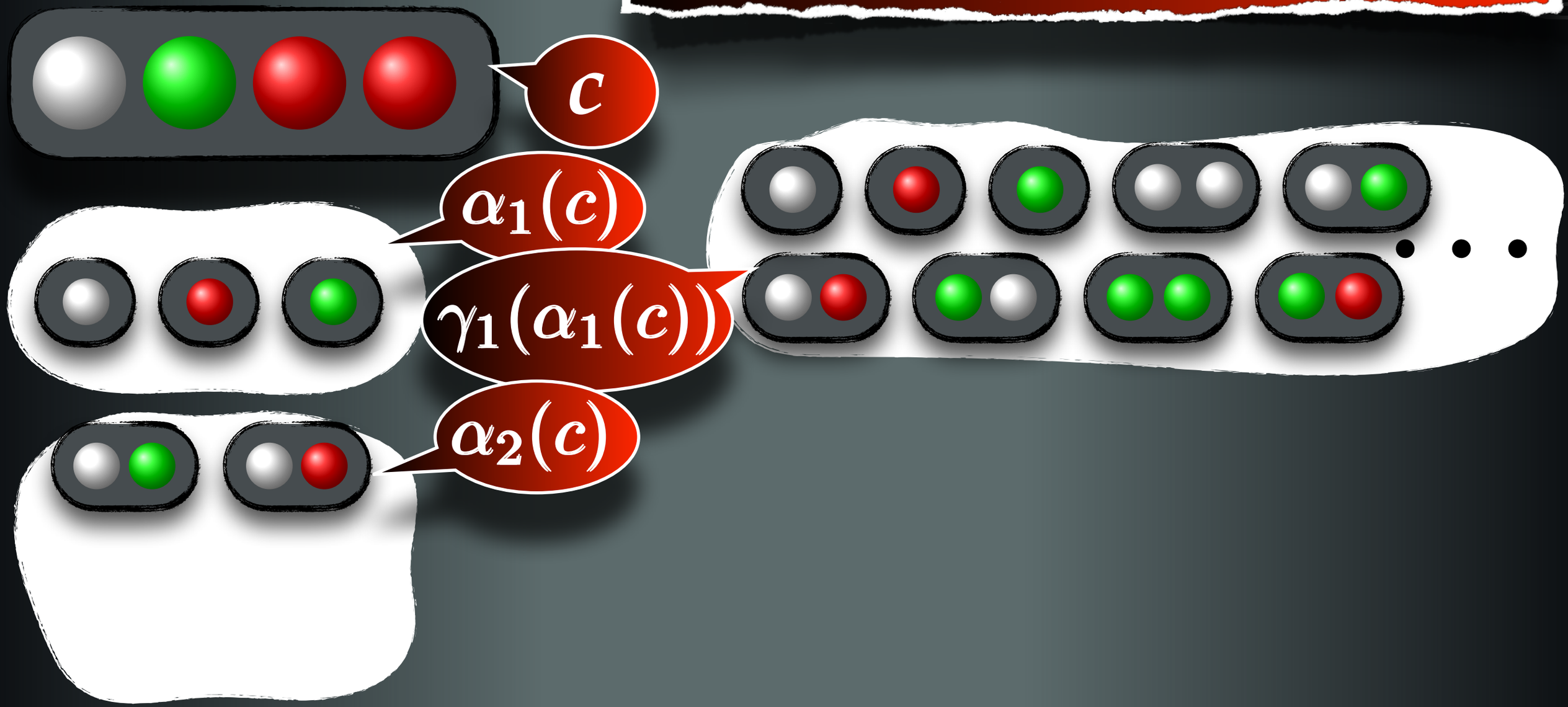
$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$



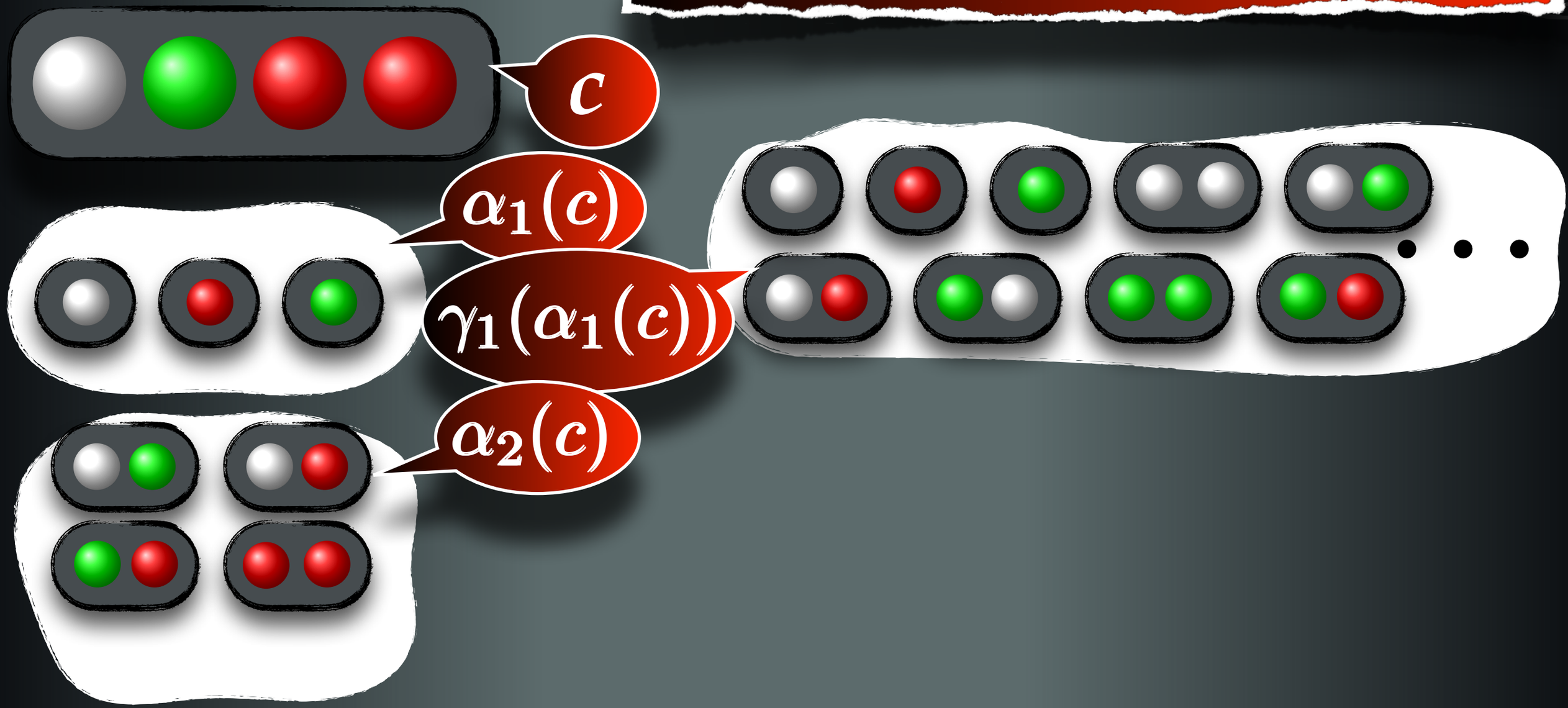
$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$



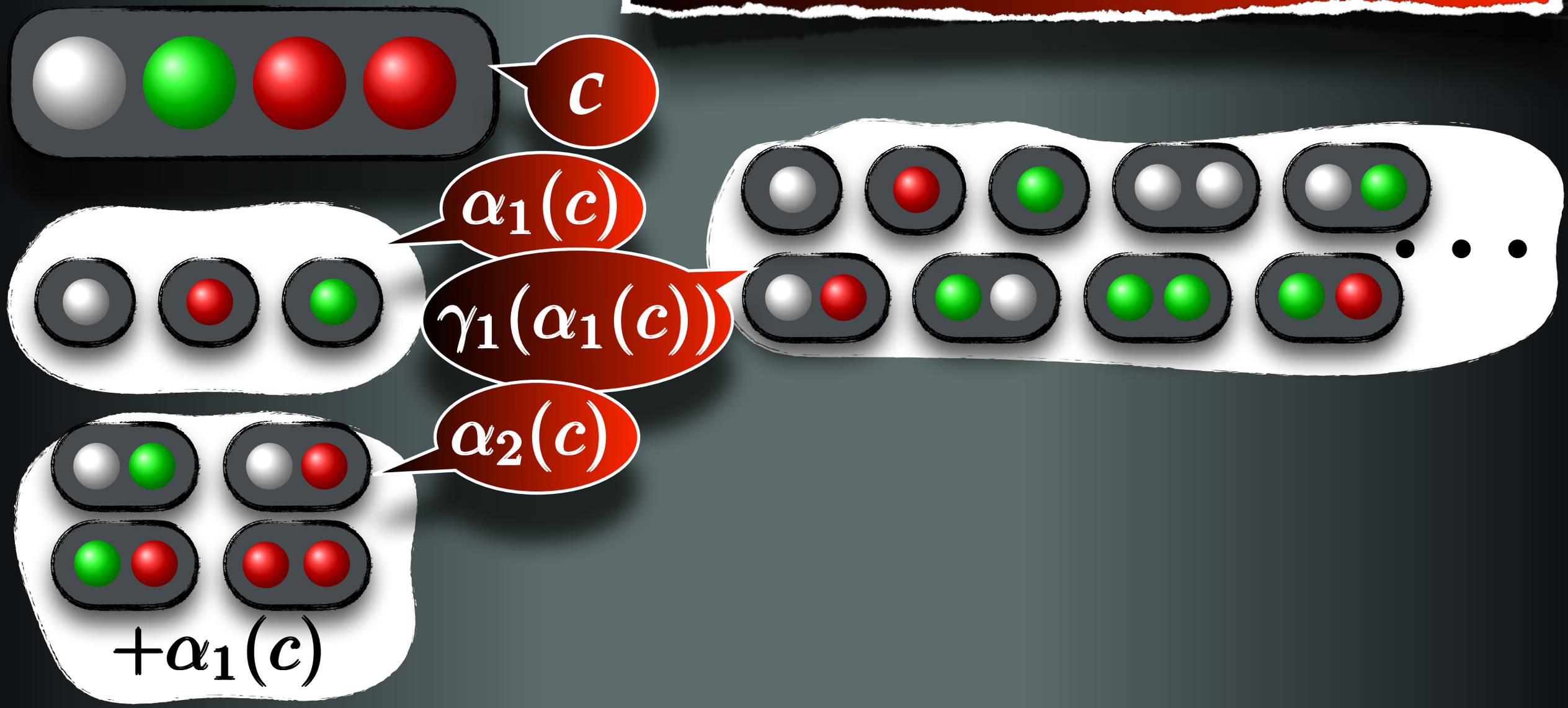
$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$



$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$



$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$



$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$

c

$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

$\alpha_2(c)$

$\gamma_2(\alpha_2(c))$

$+\alpha_1(c)$

$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$

c

$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

$\alpha_2(c)$

$\gamma_2(\alpha_2(c))$

$+\alpha_1(c)$

$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$

c

$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

$\alpha_2(c)$

$\gamma_2(\alpha_2(c))$

$+\alpha_1(c)$

$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$

c

$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

$\alpha_2(c)$

$\gamma_2(\alpha_2(c))$

$+\alpha_1(c)$

$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$

c

$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

$\alpha_2(c)$

$\gamma_2(\alpha_2(c))$

$+\alpha_1(c)$

$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$

c

$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

$\alpha_2(c)$

$\gamma_2(\alpha_2(c))$

$+\alpha_1(c)$

$^*\ +\ ^*\ +\ ^*\ +\ ^+$

$\gamma_{k+1}(\alpha_{k+1}(X)) \subset \gamma_k(\alpha_k(X))$

$\notin \gamma_2(\alpha_2(c))$

c

$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

$\alpha_2(c)$

$\gamma_2(\alpha_2(c))$

$+\alpha_1(c)$

$^* + {}^* + {}^* + {}^+$

$\gamma_{k+1}(\alpha_{k+1}(X)) \subset \gamma_{k+1}(\alpha_k(X))$

$\notin \gamma_2(\alpha_2(c))$

c

$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

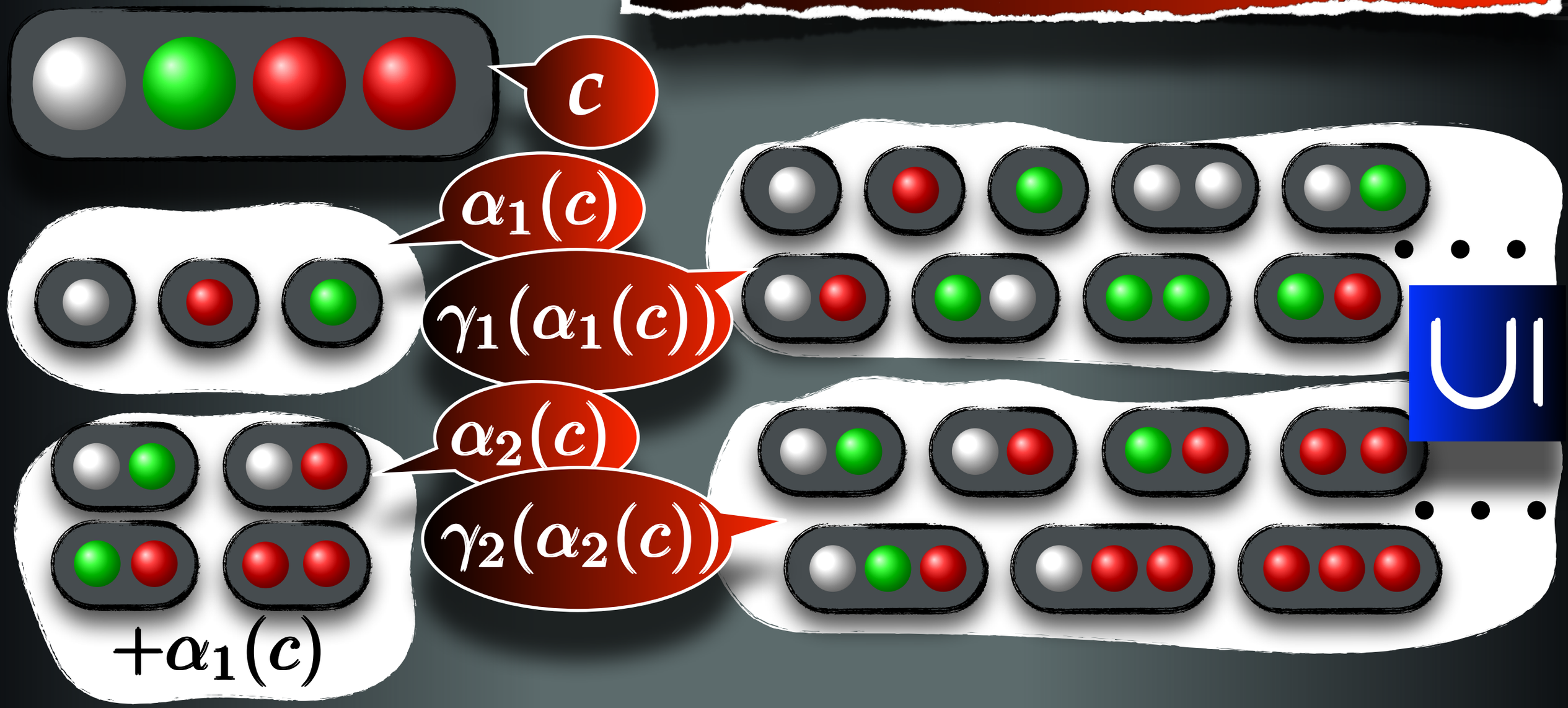
$\alpha_2(c)$

$\gamma_2(\alpha_2(c))$

$+\alpha_1(c)$

* + * + * + $^+$

$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$



$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$

c

$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

$\alpha_2(c)$

$\gamma_2(\alpha_2(c))$

$\alpha_3(c)$

$+\alpha_1(c)$

$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$

c

$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

$\alpha_2(c)$

$\gamma_2(\alpha_2(c))$

$\alpha_3(c)$

$+\alpha_1(c)$



$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$

c

$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

$\alpha_2(c)$

$\gamma_2(\alpha_2(c))$

$\alpha_3(c)$

$+\alpha_1(c)$



$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$

c

$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

$\alpha_2(c)$

$\gamma_2(\alpha_2(c))$

$\alpha_3(c)$

$+\alpha_1(c)$

UI

$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$

c

$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

$\alpha_2(c)$

$\gamma_2(\alpha_2(c))$

$\alpha_3(c)$

$+\alpha_1(c)$

$+\alpha_2(c)$



$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$

c

$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

$\alpha_2(c)$

$\gamma_2(\alpha_2(c))$

$+\alpha_1(c)$

$\alpha_3(c)$

$\gamma_3(\alpha_3(c))$

$+\alpha_2(c)$



$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$

c

$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

$\alpha_2(c)$

$\gamma_2(\alpha_2(c))$

$+\alpha_1(c)$

$\alpha_3(c)$

$\gamma_3(\alpha_3(c))$

$+\alpha_2(c)$



$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$

c

$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

$\alpha_2(c)$

$\gamma_2(\alpha_2(c))$

$+\alpha_1(c)$

$\alpha_3(c)$

$\gamma_3(\alpha_3(c))$

$+\alpha_2(c)$



$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$

c

$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

$\alpha_2(c)$

$\gamma_2(\alpha_2(c))$

$+\alpha_1(c)$

$\alpha_3(c)$

$\gamma_3(\alpha_3(c))$

$+\alpha_2(c)$



$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$

c

$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

$\alpha_2(c)$

$\gamma_2(\alpha_2(c))$

$+\alpha_1(c)$

$\alpha_3(c)$

$\gamma_3(\alpha_3(c))$

$+\alpha_2(c)$



$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$

c

$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

$\alpha_2(c)$

$\gamma_2(\alpha_2(c))$

$+\alpha_1(c)$

$\alpha_3(c)$

$\gamma_3(\alpha_3(c))$

$+\alpha_2(c)$

$\notin \gamma_3(\alpha_3(c))$

$$\gamma_{k+1}(\alpha_{k+1}(X)) \subseteq \gamma_k(\alpha_k(X))$$

c

$\alpha_1(c)$

$\gamma_1(\alpha_1(c))$

$\alpha_2(c)$

$\gamma_2(\alpha_2(c))$

$+\alpha_1(c)$

$\alpha_3(c)$

$\gamma_3(\alpha_3(c))$

$+\alpha_2(c)$

$\notin \gamma_3(\alpha_3(c))$

$\cup$

$\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$

$\gamma_k(V_k) =$

$\gamma_k(\mu X.\ \alpha_k(I) \cup Apost_k(X)) =$

$\gamma_k\ \alpha_k\ (\ I\ \cup\ post(\gamma_k(\alpha_k(\ \cdot\ ))) \cup\ post(\gamma_k(\alpha_k(\ \cdot\ ))) \cup\ post(\gamma_k(\alpha_k(\ \cdot\ ))) \cup \bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\ )$

$\gamma_{k+1}(V_{k+1}) =$

$\gamma_{k+1}(\mu X.\ \alpha_{k+1}(I) \cup Apost_{k+1}(X)) =$

$\gamma_{k+1}\ \alpha_{k+1}\ (\ I\ \cup\ post(\gamma_{k+1}(\alpha_{k+1}(\ \cdot\ ))) \cup\ post(\gamma_{k+1}(\alpha_{k+1}(\ \cdot\ ))) \cup\ post(\gamma_{k+1}(\alpha_{k+1}(\ \cdot\ ))) \cup \bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet\ )$

$A_0 \supseteq B_0$

$A_1 \supseteq B_1$

$A_2 \supseteq B_2$

$A_3 \supseteq B_3$

$B_i \subseteq A_i$

induction on i

$\gamma_{k+1}(\alpha_{k+1}(B_i)) \subseteq \gamma_k(\alpha_k(B_i)) \subseteq \gamma_k(\alpha_k(A_i))$

$B_{i+1} = post(\gamma_{k+1}(\alpha_{k+1}(B_i))) \subseteq post(\gamma_k(\alpha_k(A_i))) = A_{i+1}$

For each k : two procedures (in paraellel):

- **Bug detection:**
  - under-approximation
  - concrete domain
- **Verification:**
  - over-approximation
  - abstract domain

- abstraction of initial configurations ✓
- abstract post operator ✓

For each k : two procedures (in paraellel):

- **Bug detection:**
  - under-approximation
  - concrete domain
- **Verification:**
  - over-approximation
  - abstract domain

- abstraction of initial configurations ✓
- abstract post operator ✓

1

$$\mathbb{R} \subseteq \gamma_k(V_k)$$

2

$$\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$$

1

$$\mathbb{R} \subseteq \gamma_k(V_k)$$

2

$$\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$$



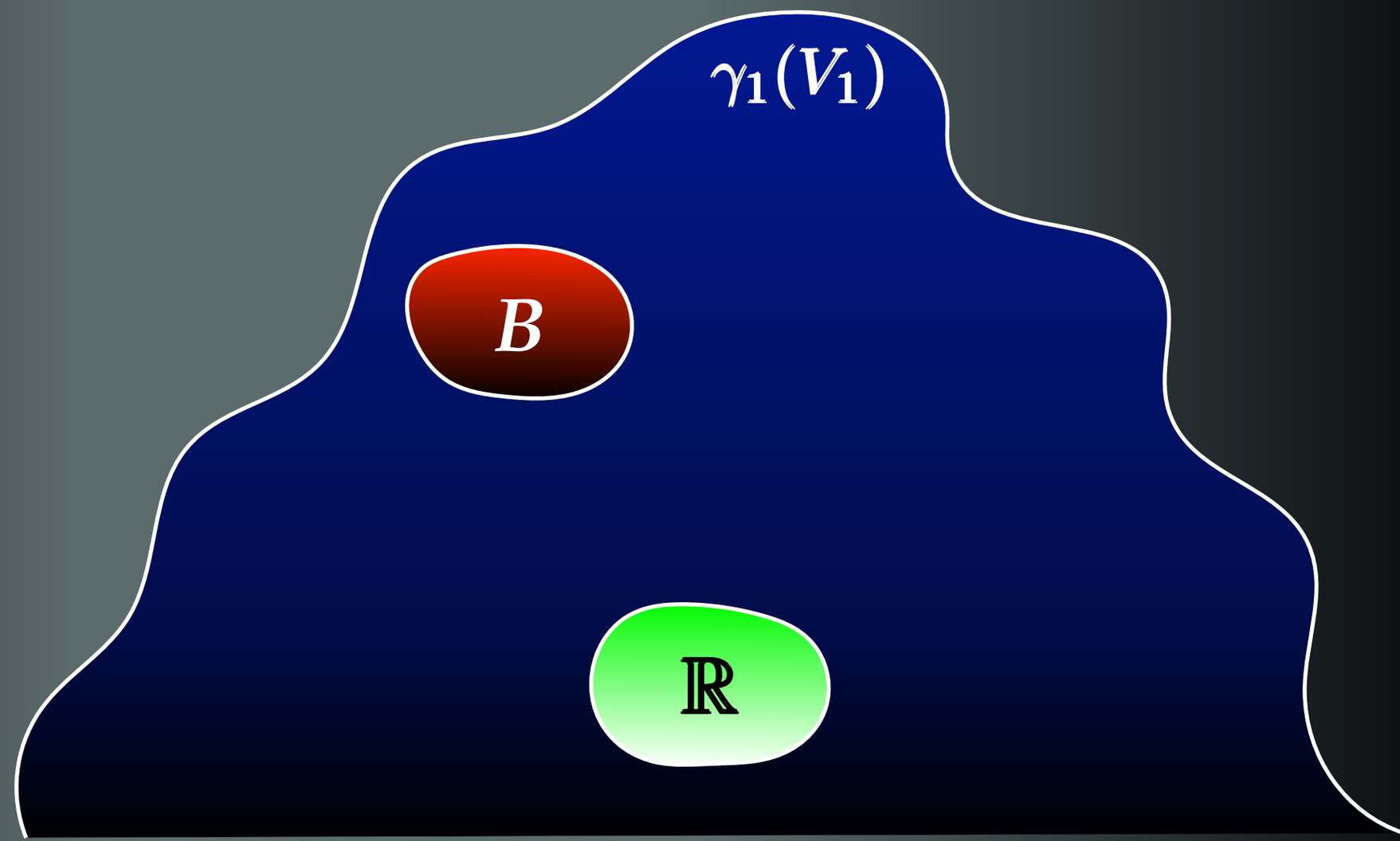


1

$$\mathbb{R} \subseteq \gamma_k(V_k)$$

2

$$\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$$

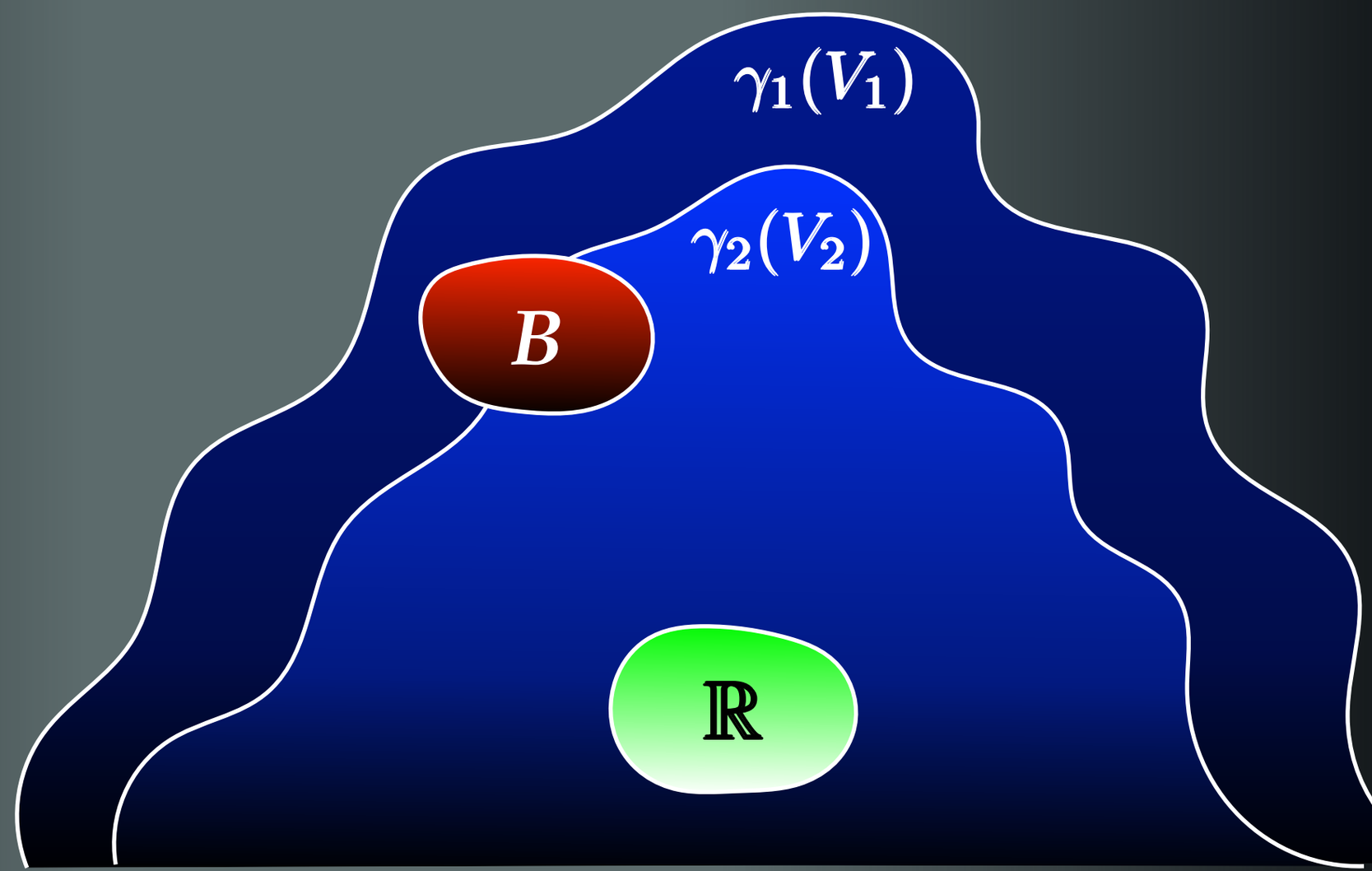


1

$$\mathbb{R} \subseteq \gamma_k(V_k)$$

2

$$\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$$

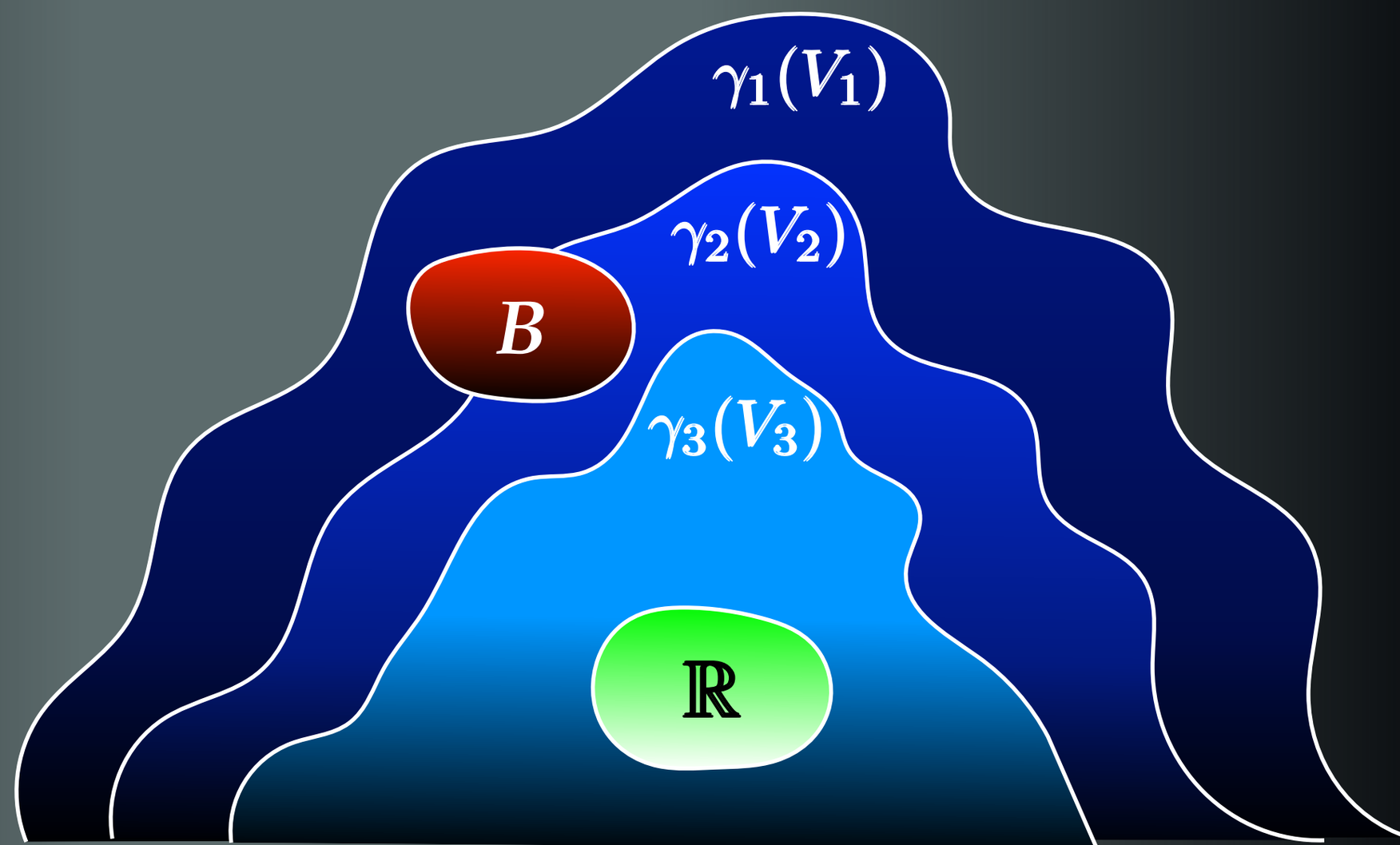


1

$$\mathbb{R} \subseteq \gamma_k(V_k)$$

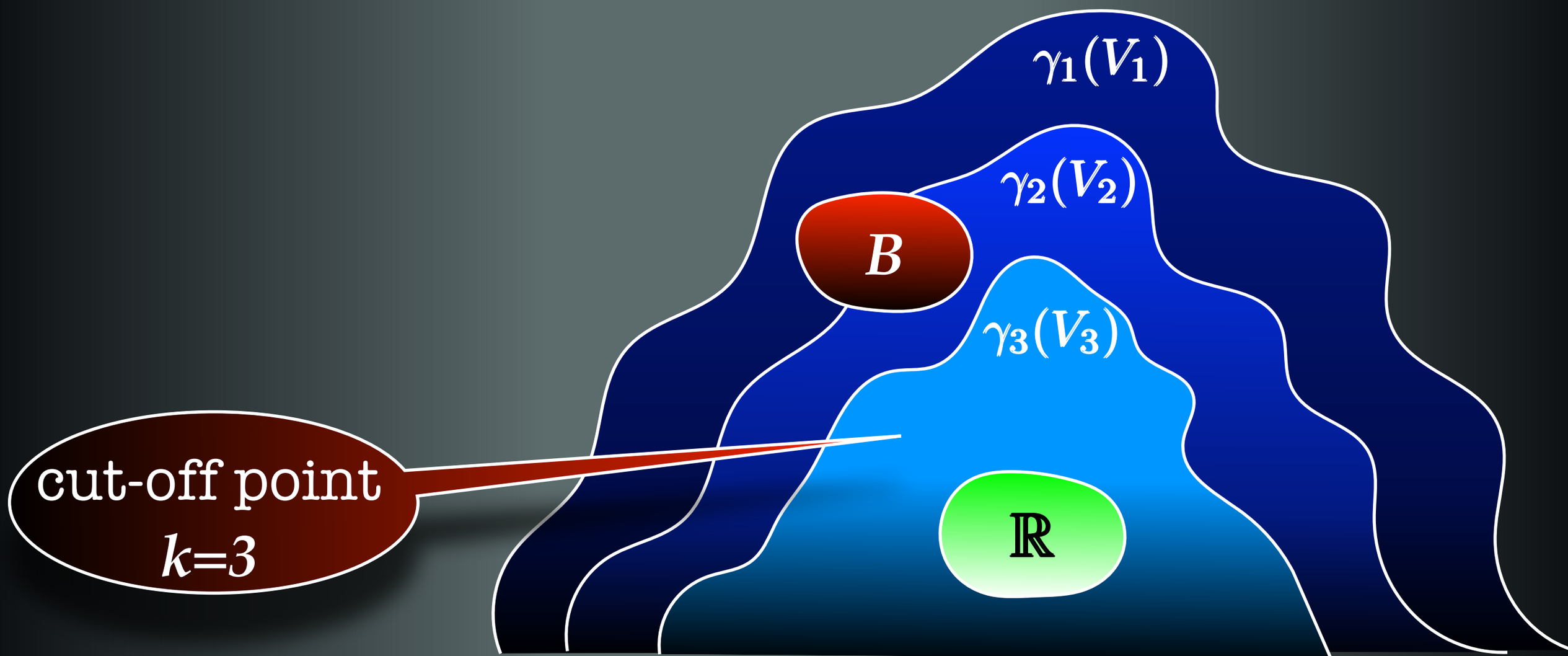
2

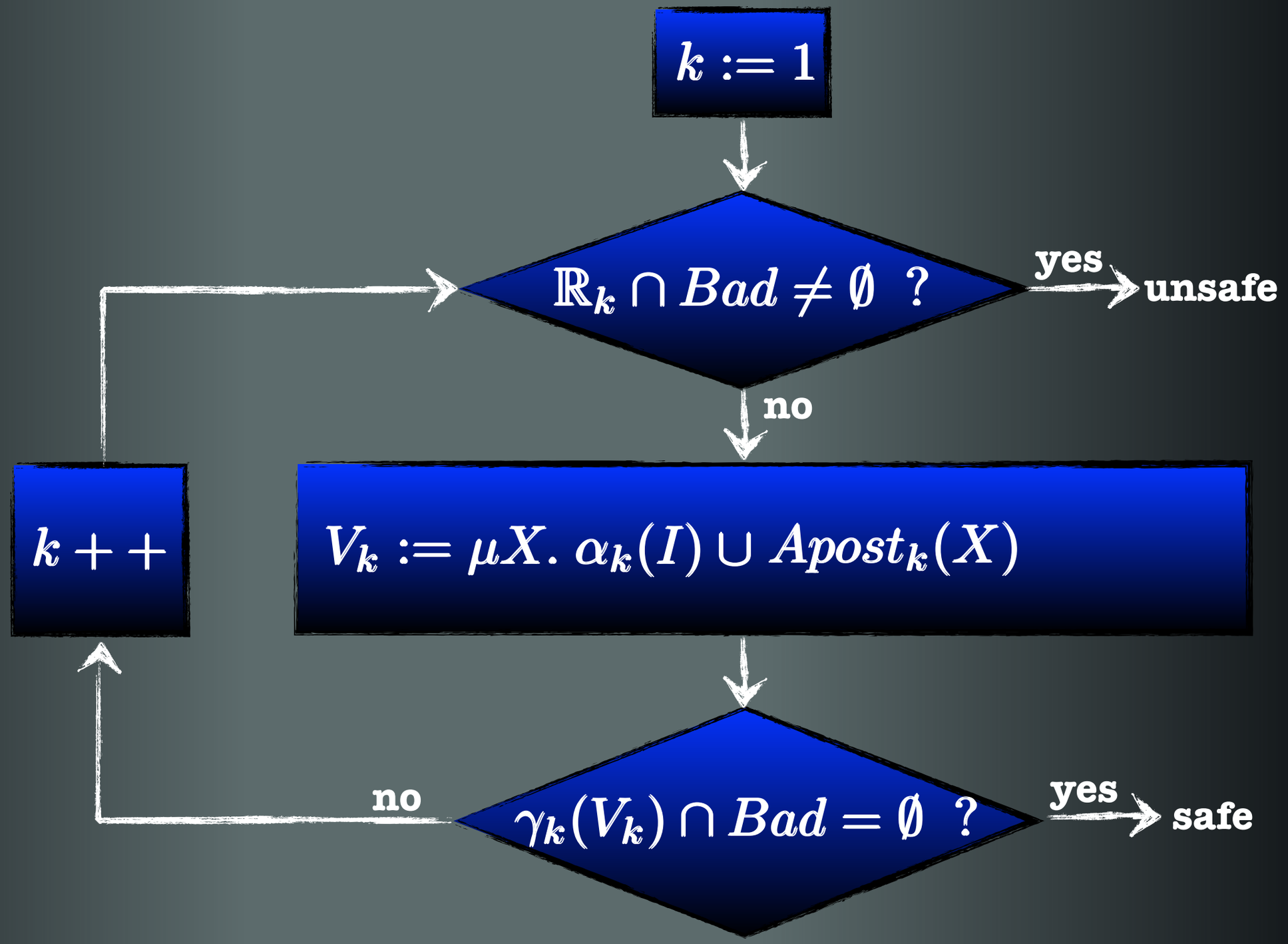
$$\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$$



1 $\mathbb{R} \subseteq \gamma_k(V_k)$

2 $\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$




$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$V_k := \mu X.\ \alpha_k(I) \cup Apost_k(X)$
$k++$
$\gamma_k(V_k) \cap Bad = \emptyset$?
yes
safe
no

$\mathbb{R} \subseteq \gamma_k(V_k)$ 1

$k := 1$

$\mathbb{R}_k \cap Bad \neq \emptyset$?

yes → unsafe

no

$V_k := \mu X.\ \alpha_k(I) \cup Apost_k(X)$

$k++$

$\gamma_k(V_k) \cap Bad = \emptyset$?

no

yes → safe

$\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$ 2

$\mathbb{R} \subseteq \gamma_k(V_k)$ 1

$k := 1$

$\mathbb{R}_k \cap Bad \neq \emptyset$?

yes

unsafe

no

$V_k := \mu X.\ \alpha_k(I) \cup Apost_k(X)$

$k++$

$\gamma_k(V_k) \cap Bad = \emptyset$?

yes

safe

no

$\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$ 2

$\mathbb{R} \subseteq \gamma_k(V_k)$ 1

$k := 1$

$\mathbb{R}_k \cap Bad \neq \emptyset$?

yes

unsafe

no

$V_k := \mu X.\ \alpha_k(I) \cup Apost_k(X)$

$k++$

increase precision

$\gamma_k(V_k) \cap Bad = \emptyset$?

no

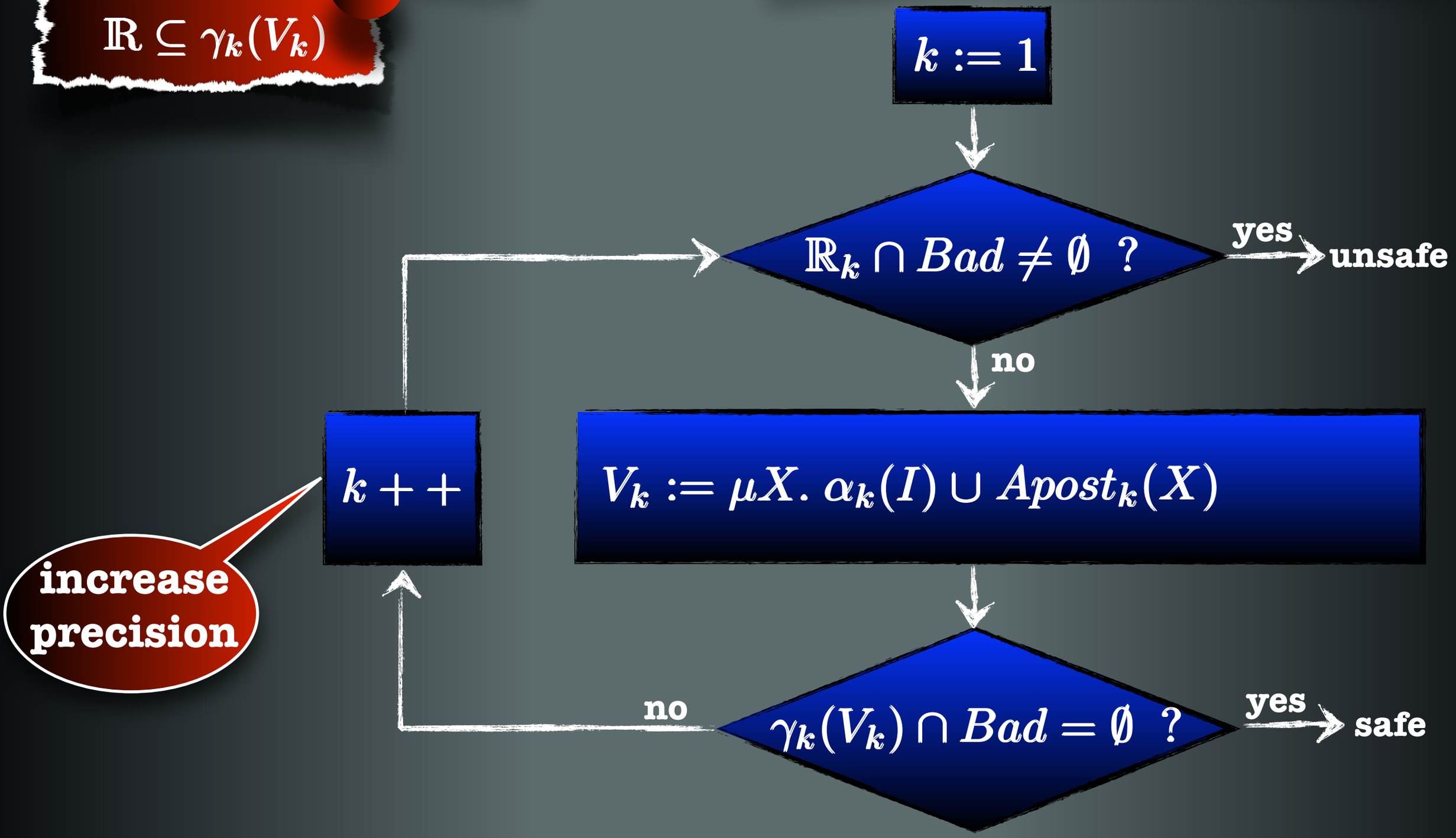
yes

safe

$\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$ 2

$\mathbb{R} \subseteq \gamma_k(V_k)$ 1

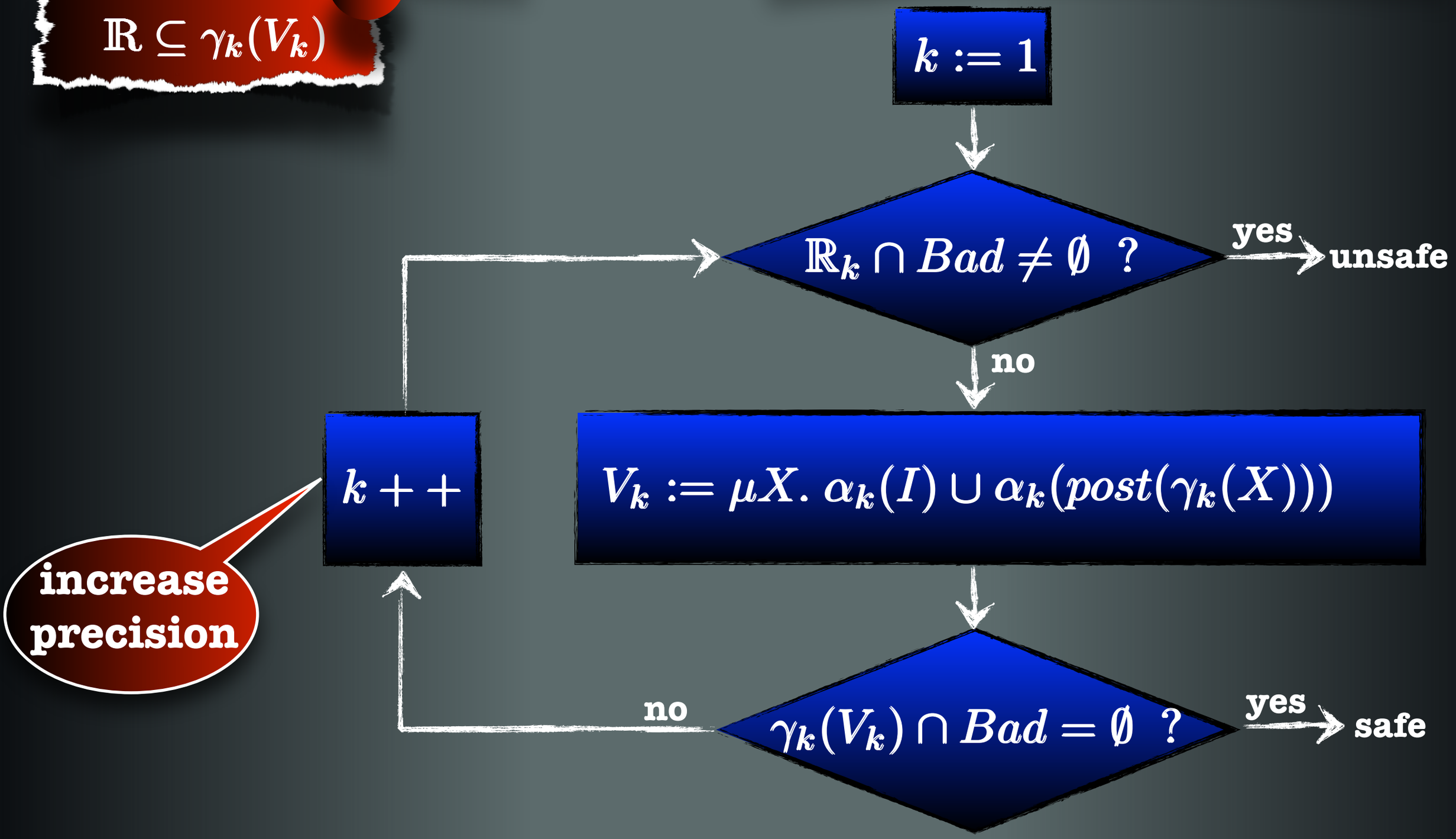
$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$



$\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$ 2

$\mathbb{R} \subseteq \gamma_k(V_k)$ 1

$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$



$\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$ 2

$\mathbb{R} \subseteq \gamma_k(V_k)$ 1

$Apost_k(X) := \alpha_k(post(\gamma_k(X)))$

$k := 1$

$\mathbb{R}_k \cap Bad \neq \emptyset$?

yes → unsafe

no

challenge: infinite set

$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\gamma_k(X)))$

$k++$

increase precision

$\gamma_k(V_k) \cap Bad = \emptyset$?

no

yes → safe

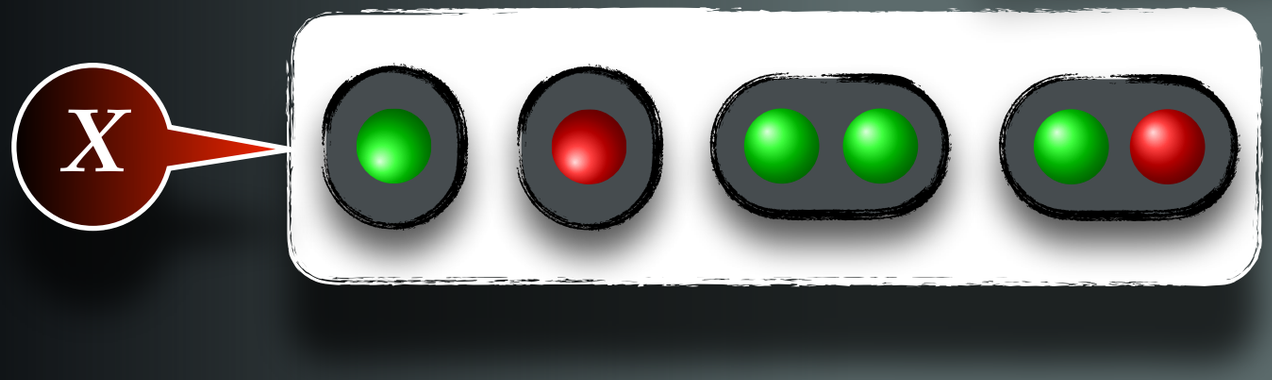
$$\gamma_k(X) := \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

$$\gamma_k(X) := \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

$$\oint_k^m (X) := \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X \ \wedge\ |c| \leq m\}$$

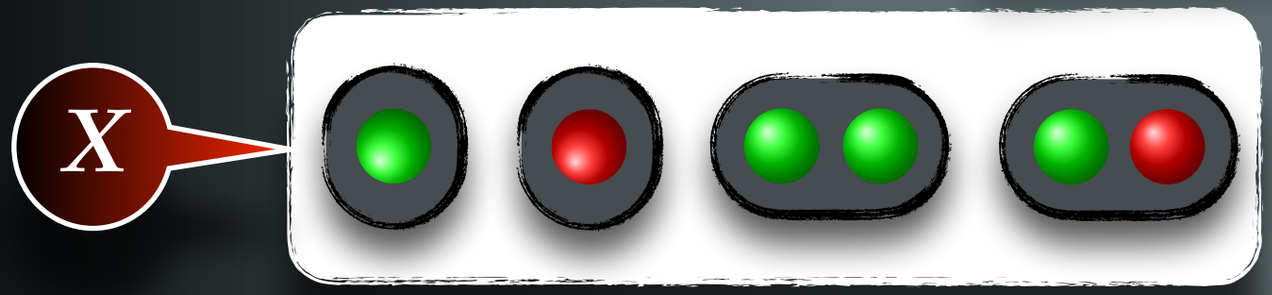
$$\gamma_k(X) := \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

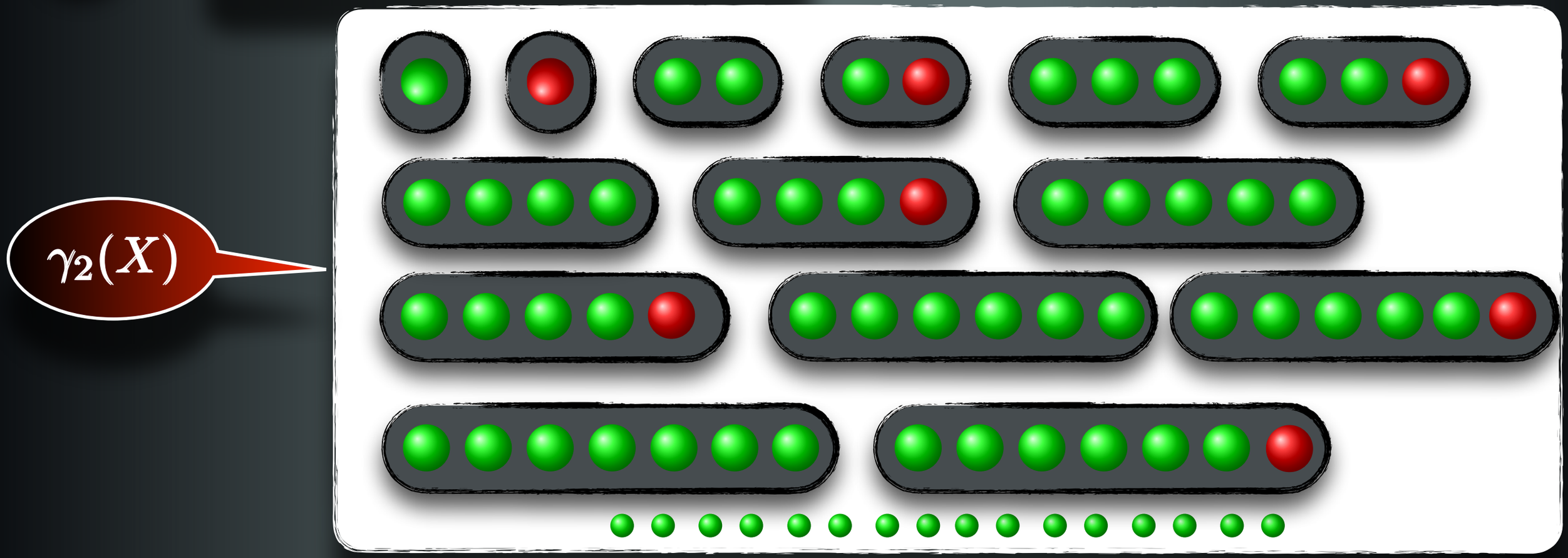
$$\oint_k^m (X) := \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X \ \wedge\ |c| \leq m\}$$



$$\gamma_k(X) := \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

$$\oint_k^m (X) := \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X \ \wedge\ |c| \leq m\}$$

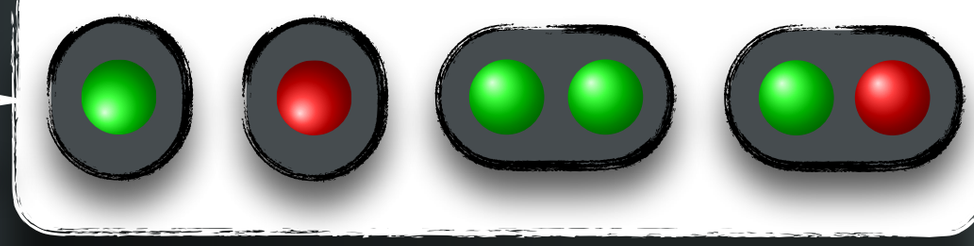




$$\gamma_k(X) := \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

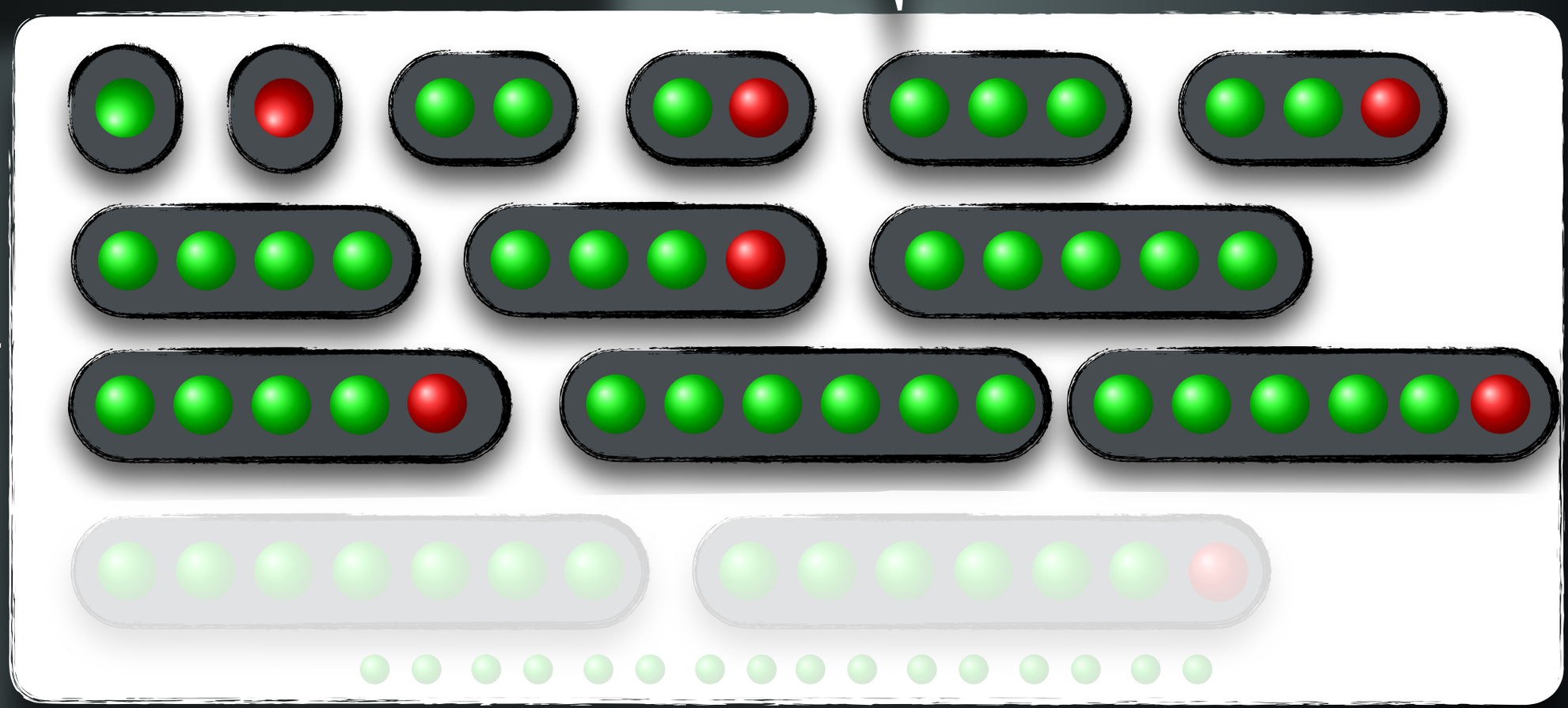
$$\oint_k^m (X) := \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X \ \wedge\ |c| \leq m\}$$

X



$\oint_2^6 (X)$

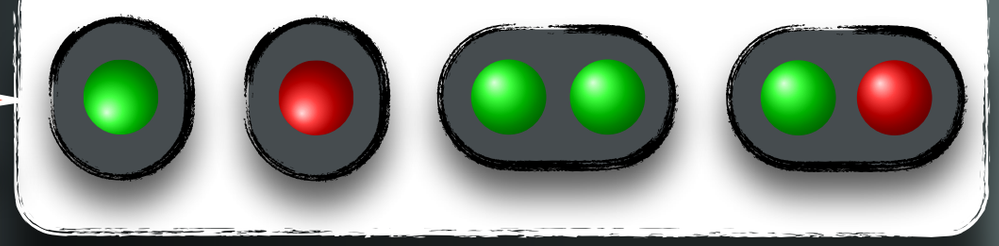
$\gamma_2(X)$



$$\gamma_k(X) := \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

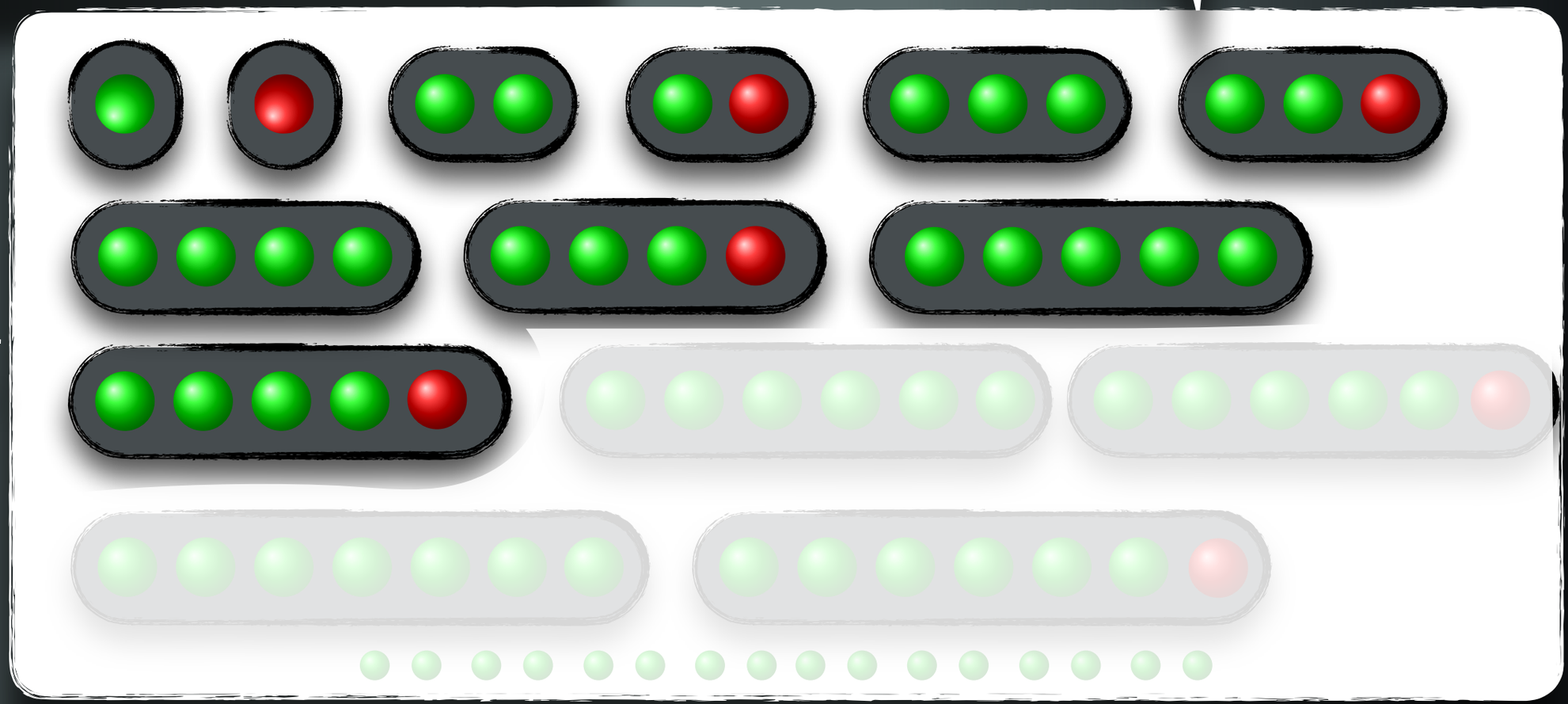
$$\oint_k^m (X) := \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X \ \wedge\ |c| \leq m\}$$

X



$\oint_2^5 (X)$

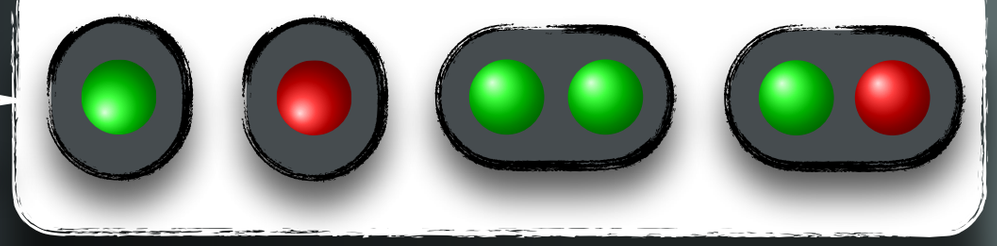
$\gamma_2(X)$



$$\gamma_k(X) := \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

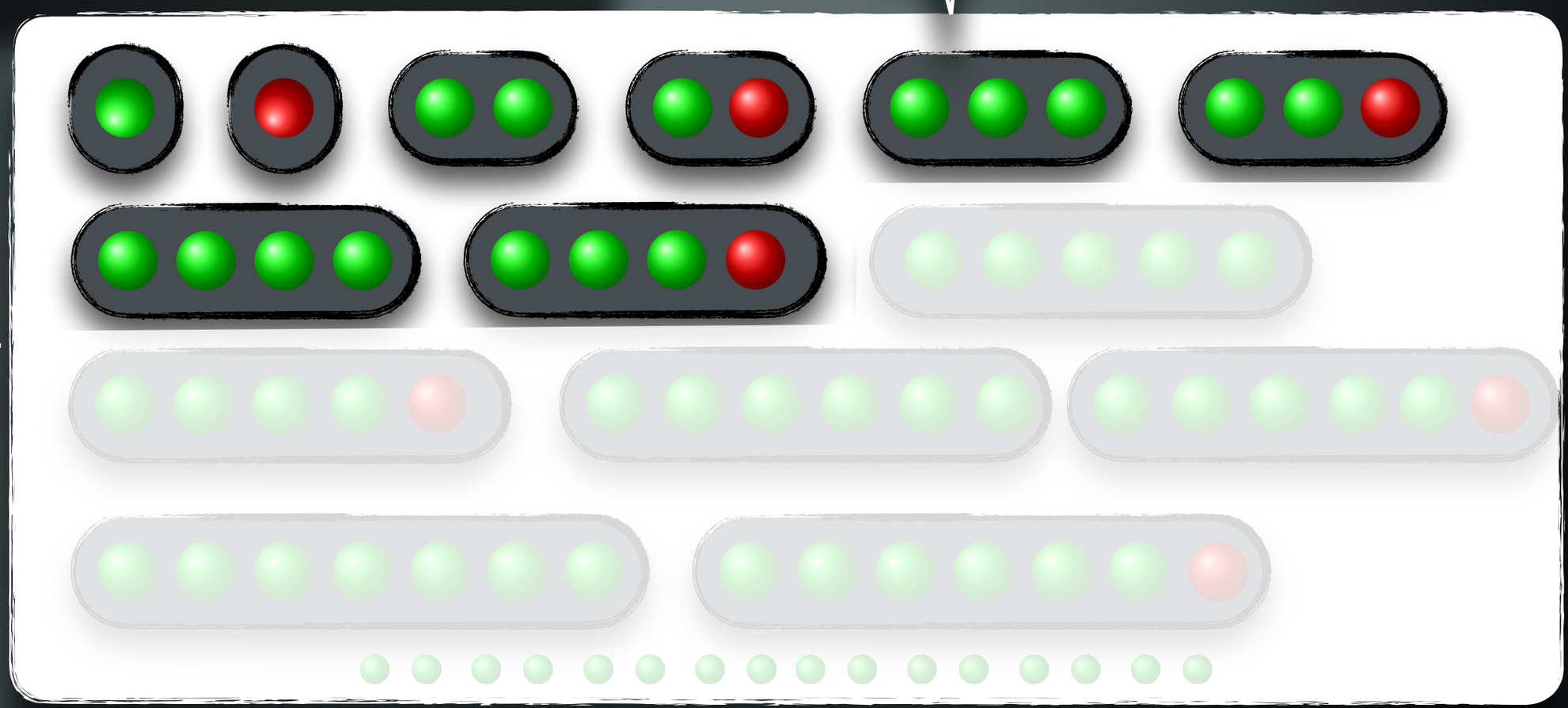
$$\oint_k^m (X) := \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X \ \wedge\ |c| \leq m\}$$

X



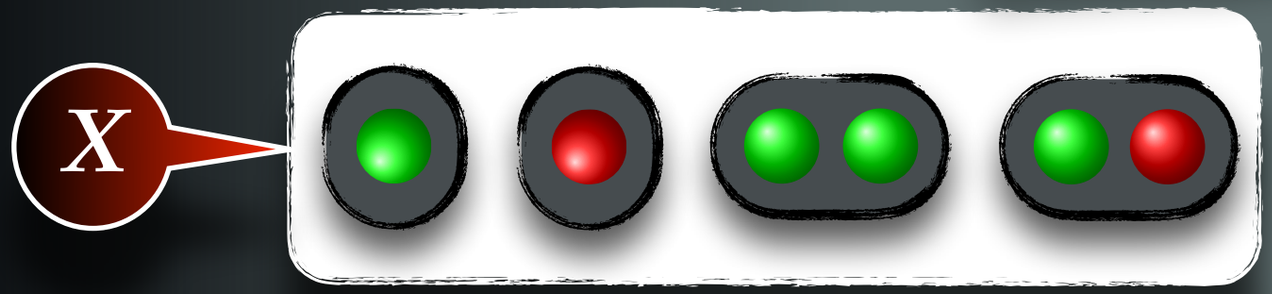
$\oint_2^4 (X)$

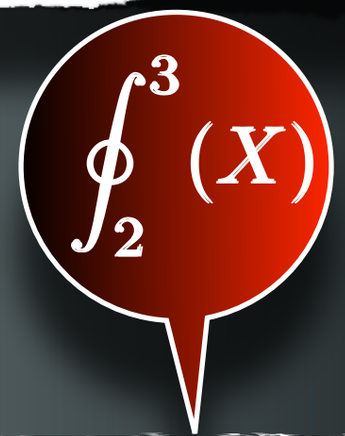
$\gamma_2(X)$

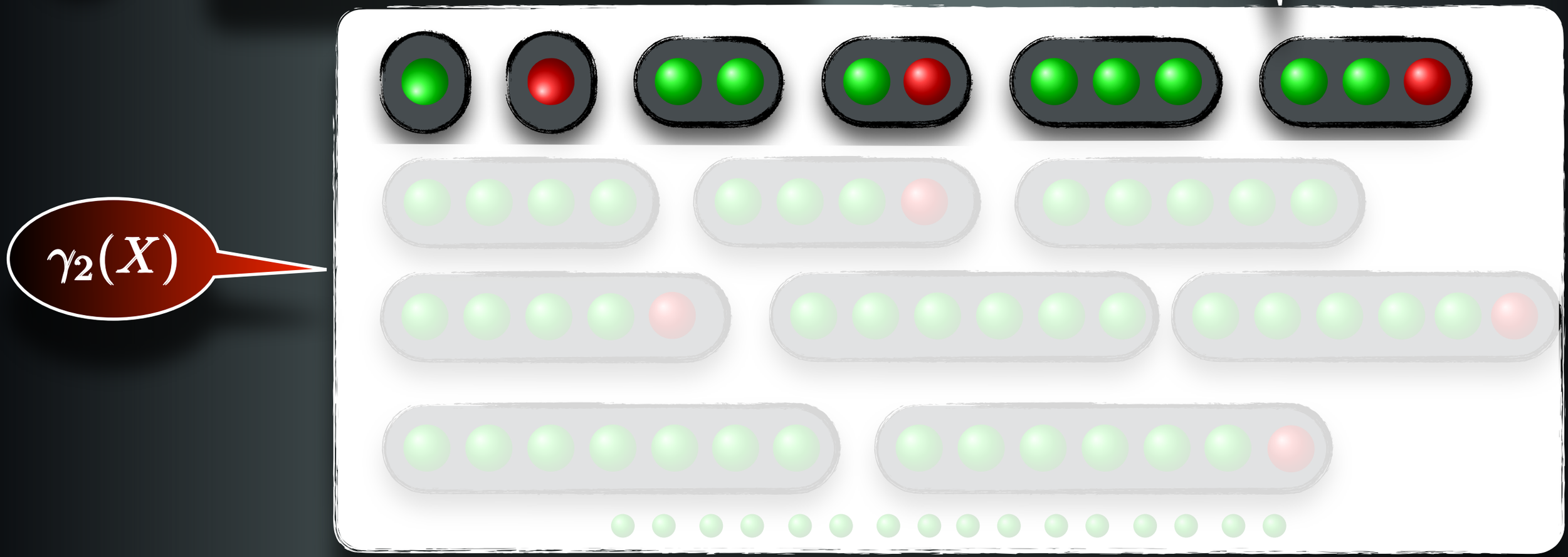


$$\gamma_k(X) := \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X\}$$

$$\oint_k^m (X) := \{c \in \mathbb{C} |\ \alpha_k(c) \subseteq X \ \wedge\ |c| \leq m\}$$

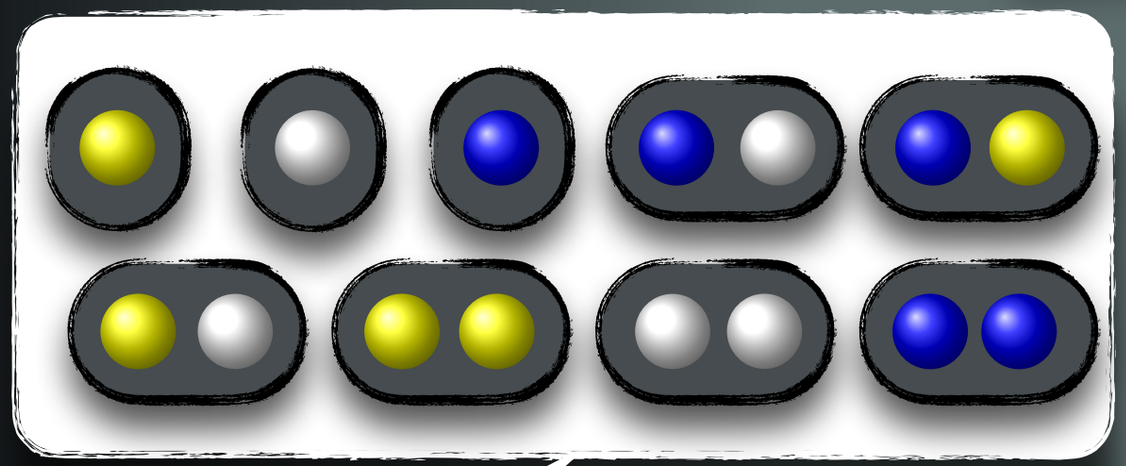




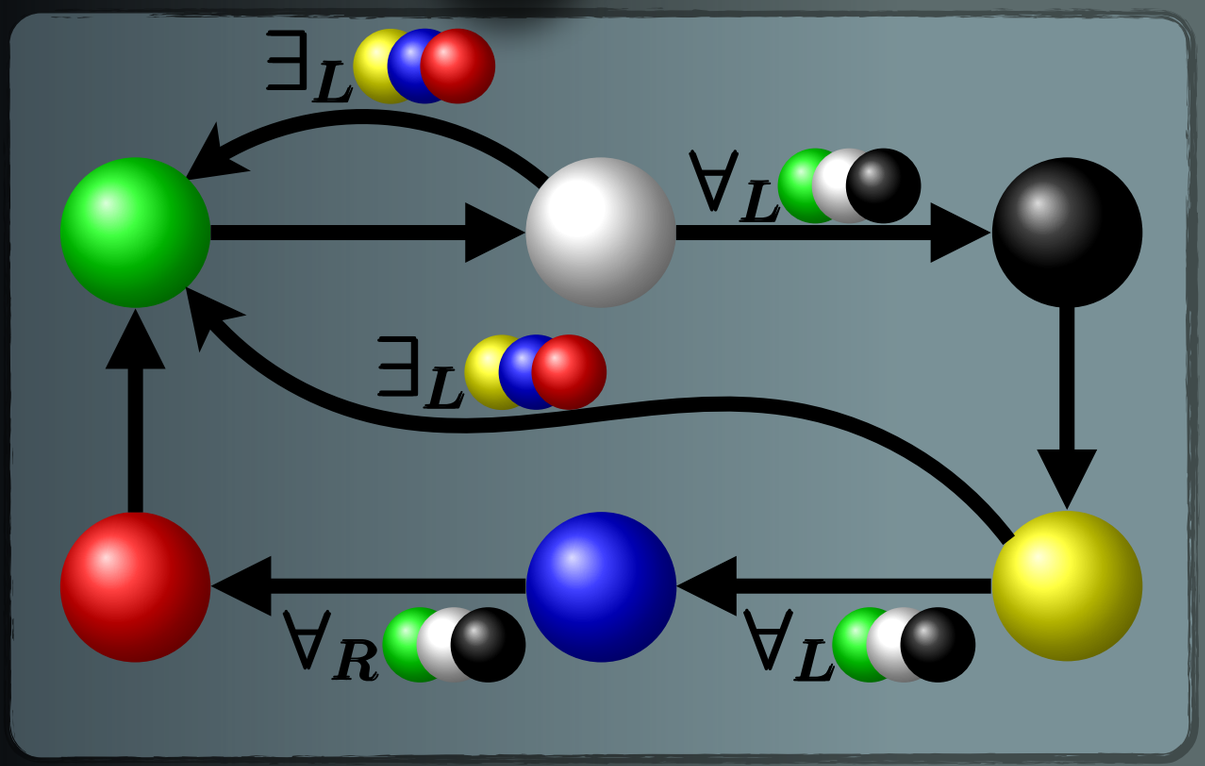


$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$

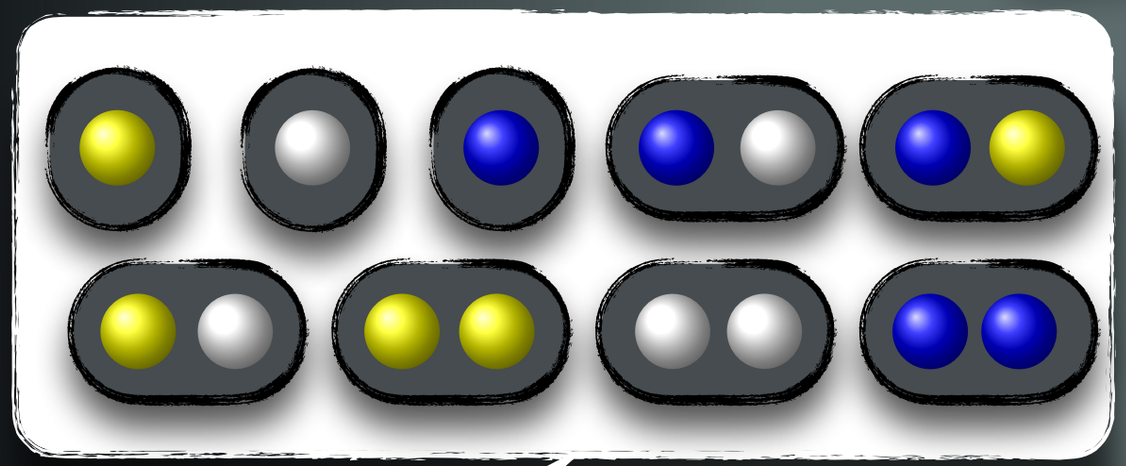
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



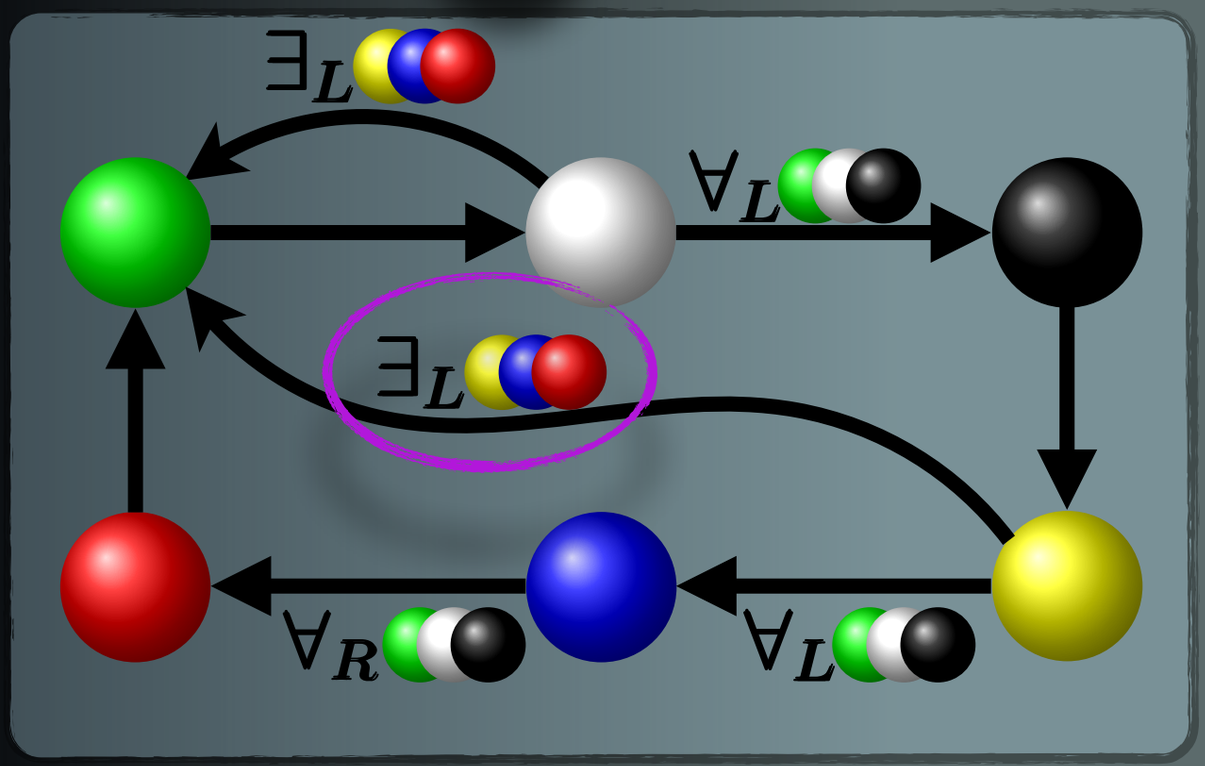
X



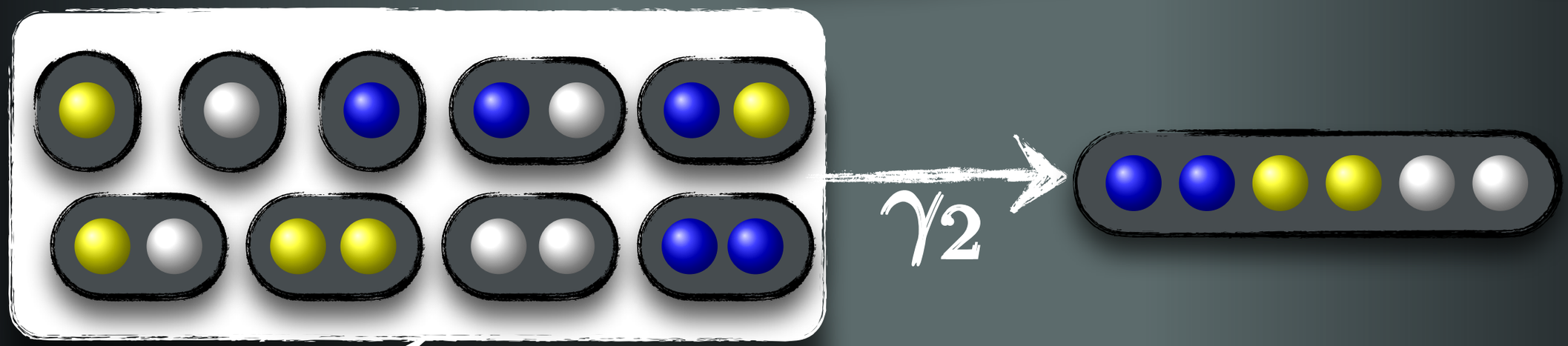
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



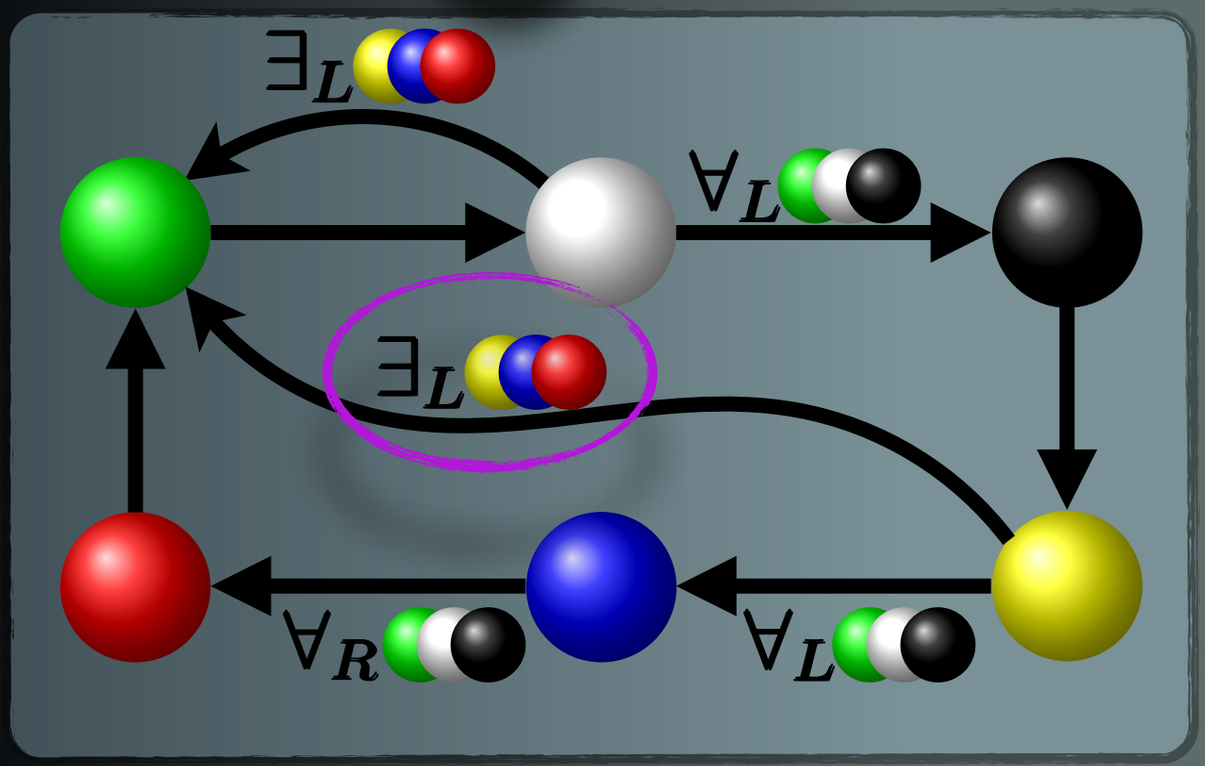
X



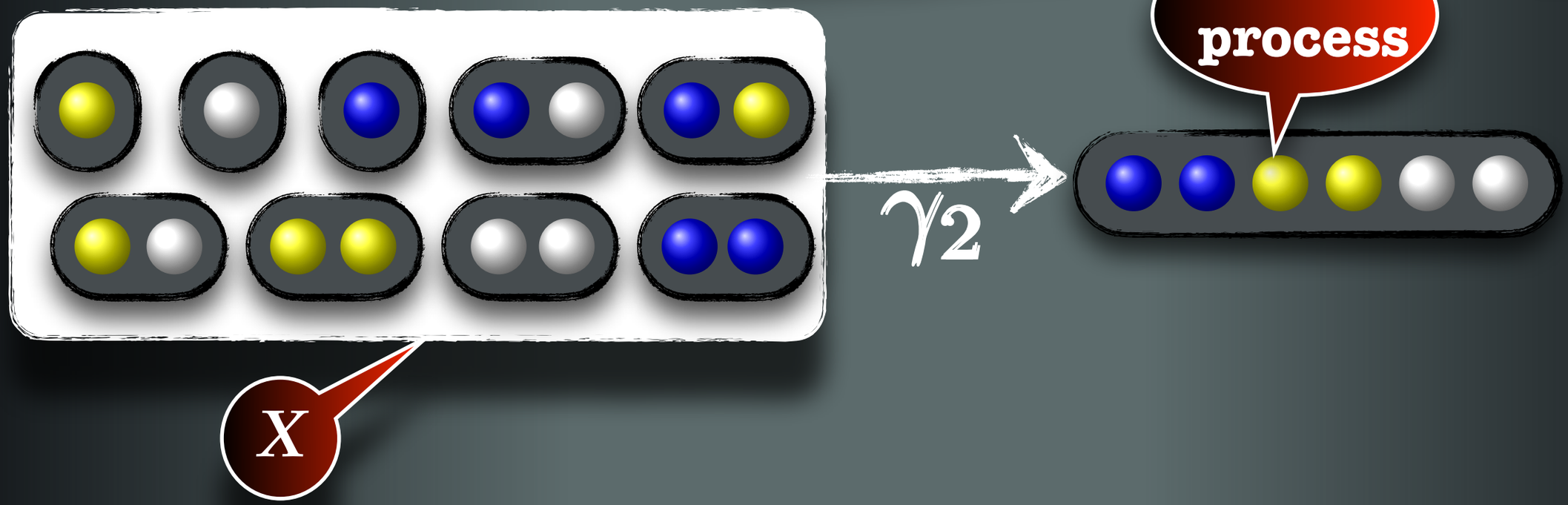
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$

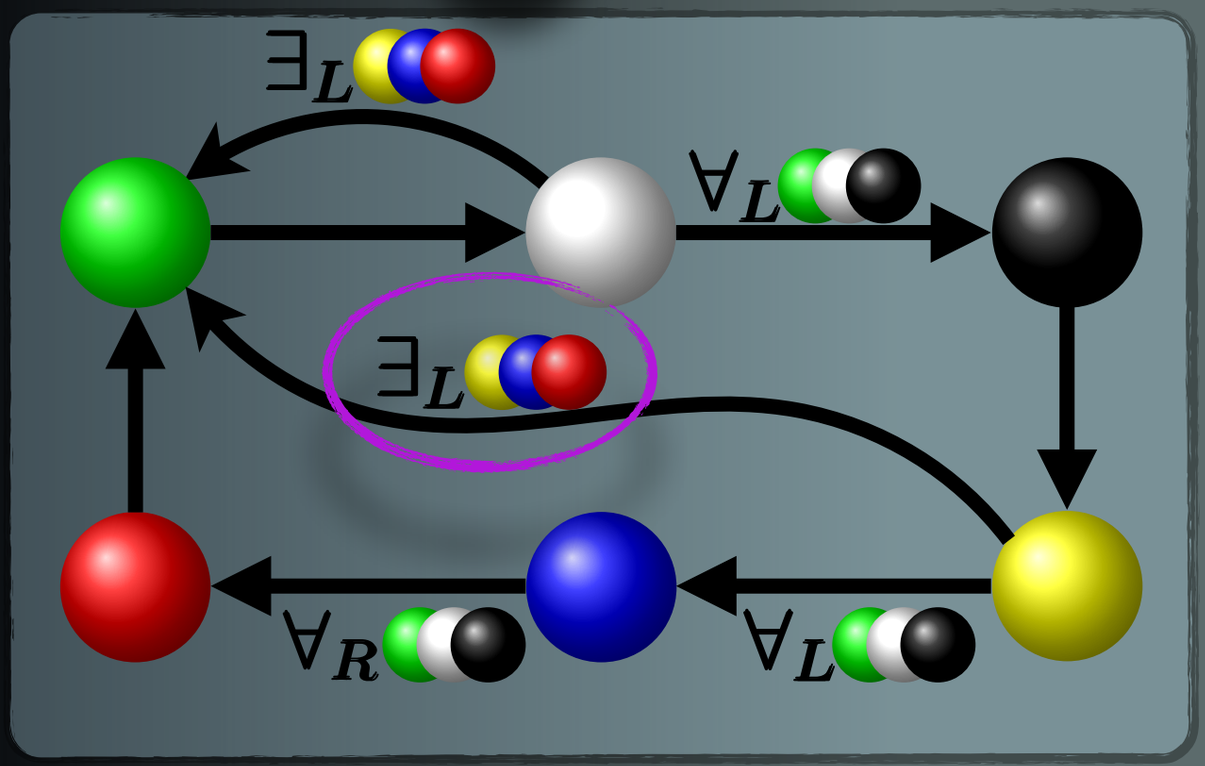


X

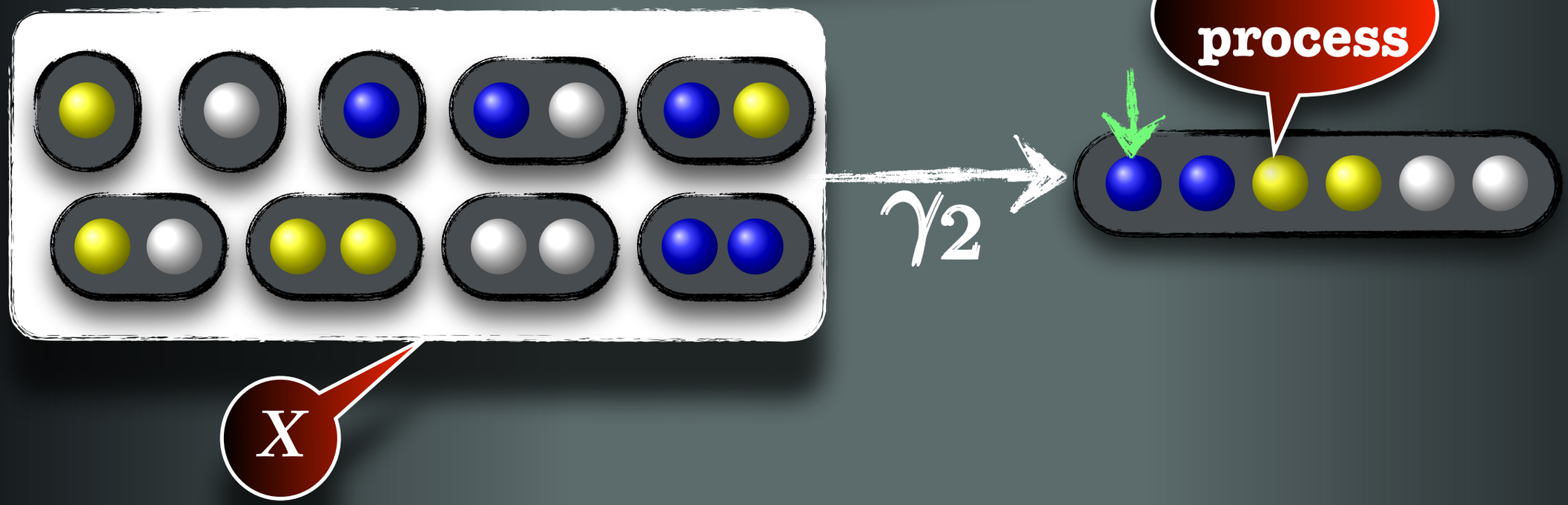


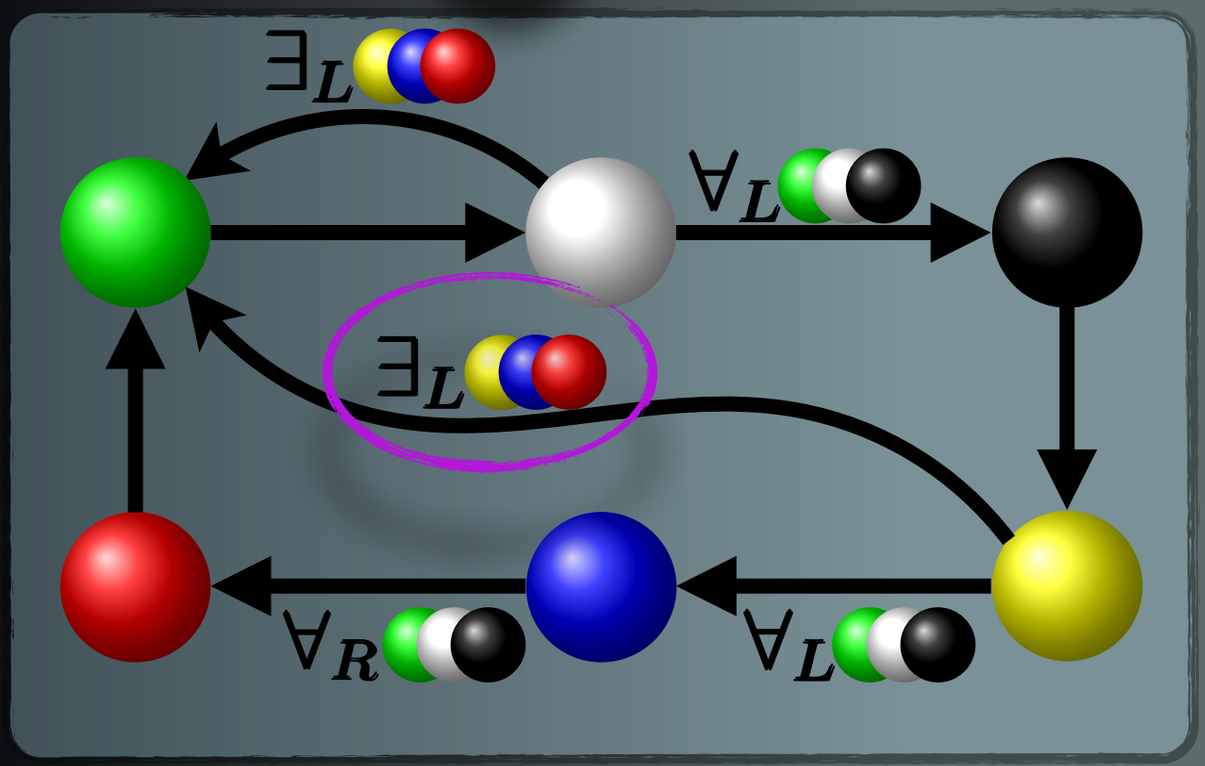
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



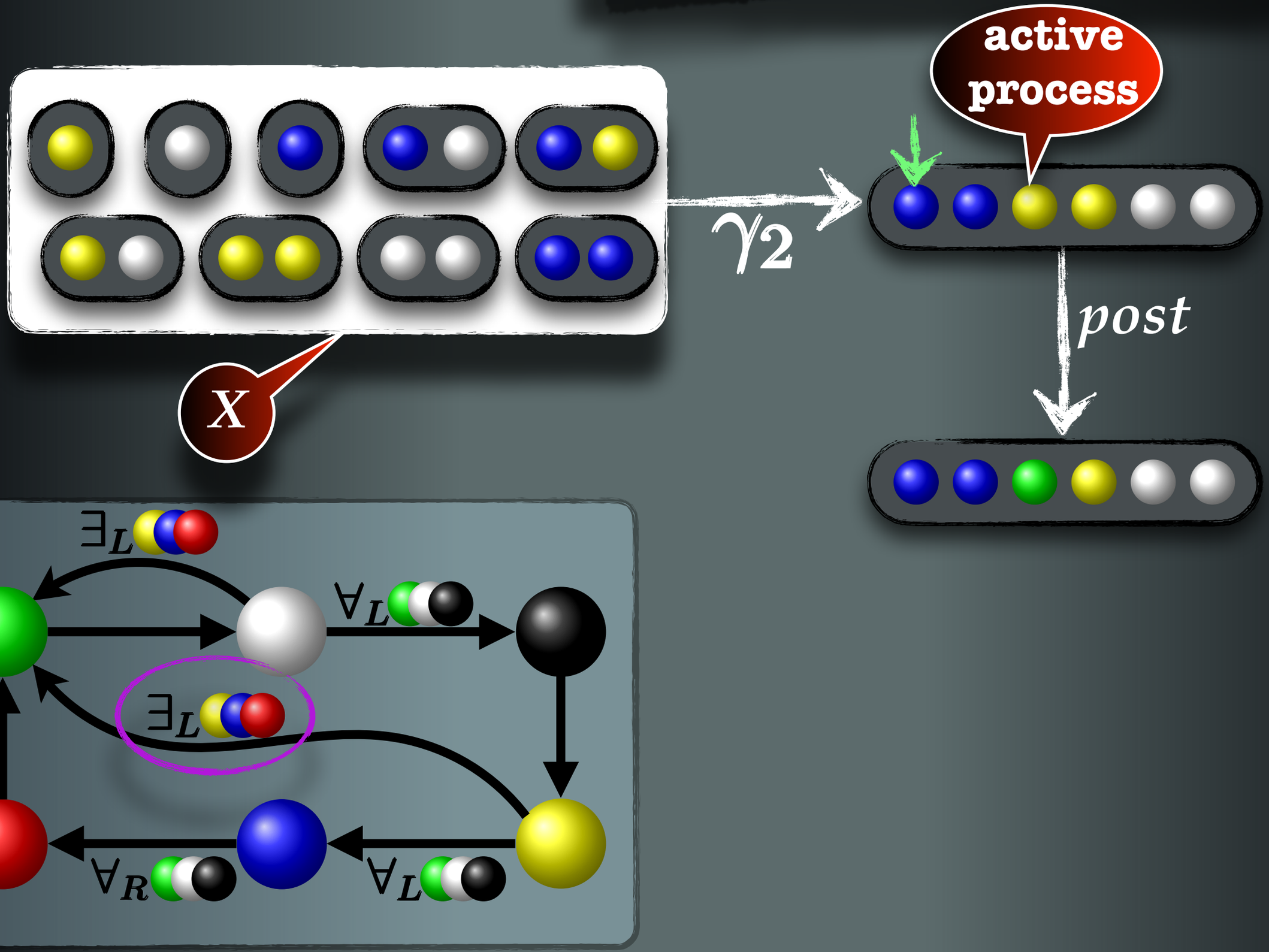


$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$

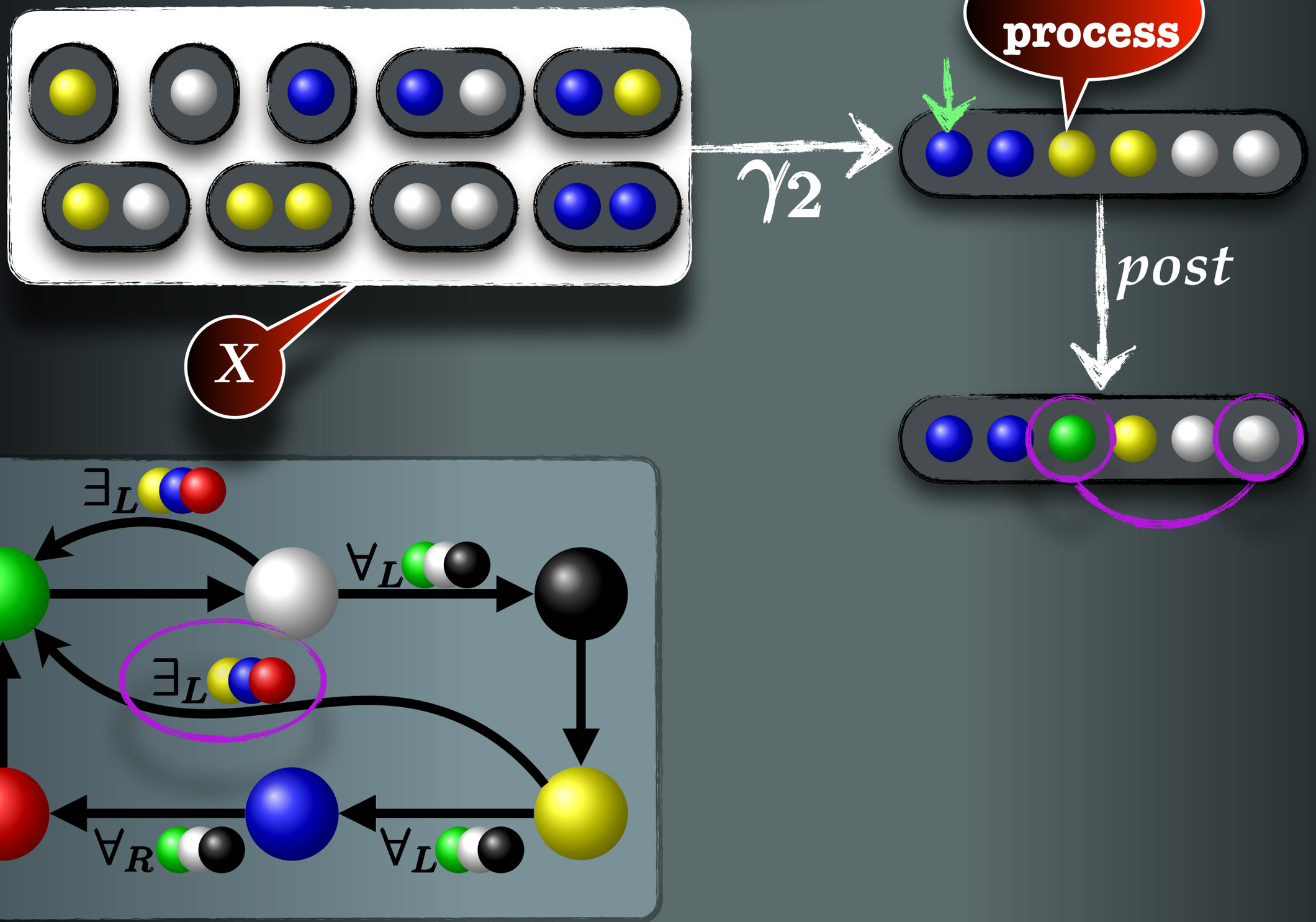




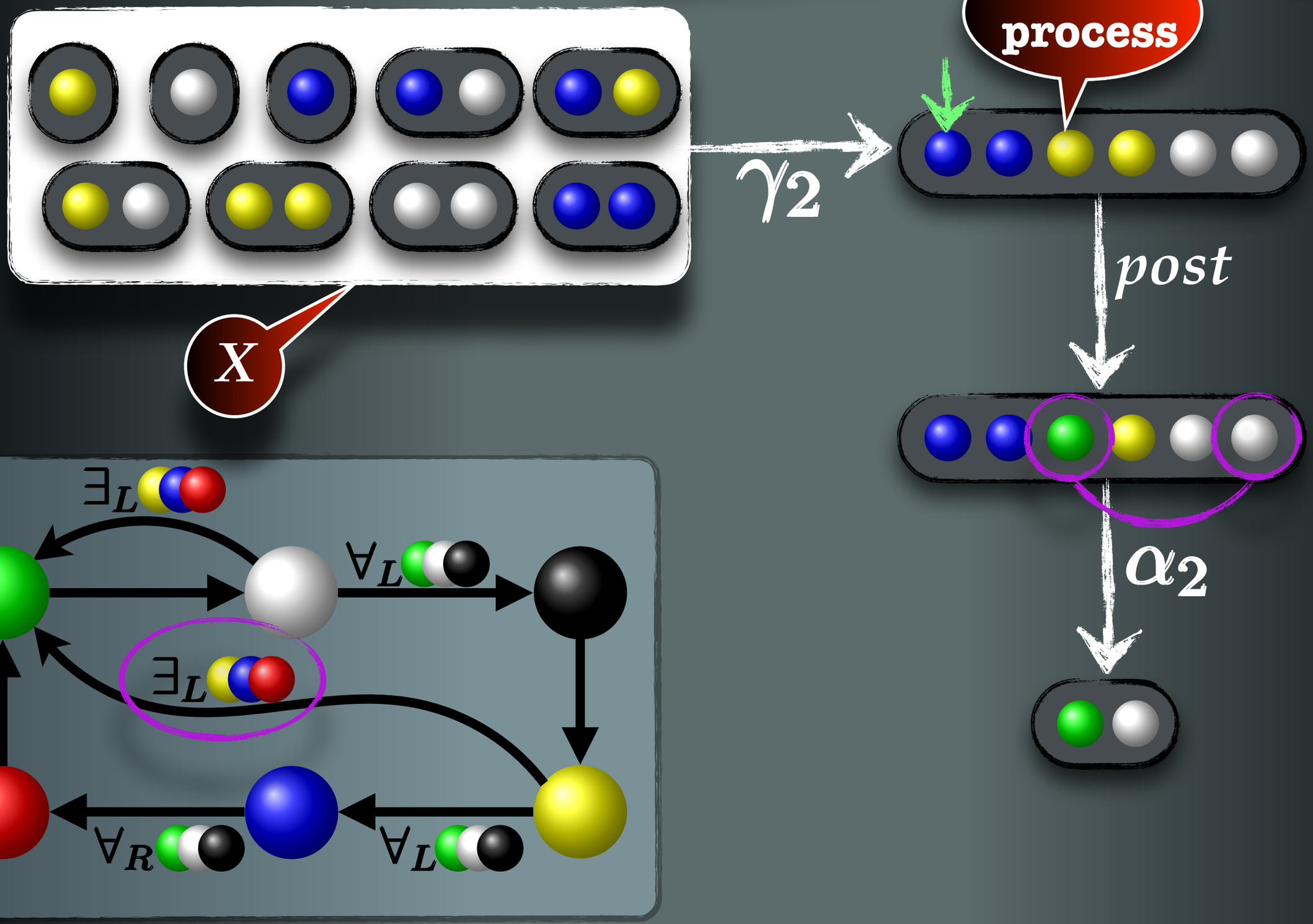
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



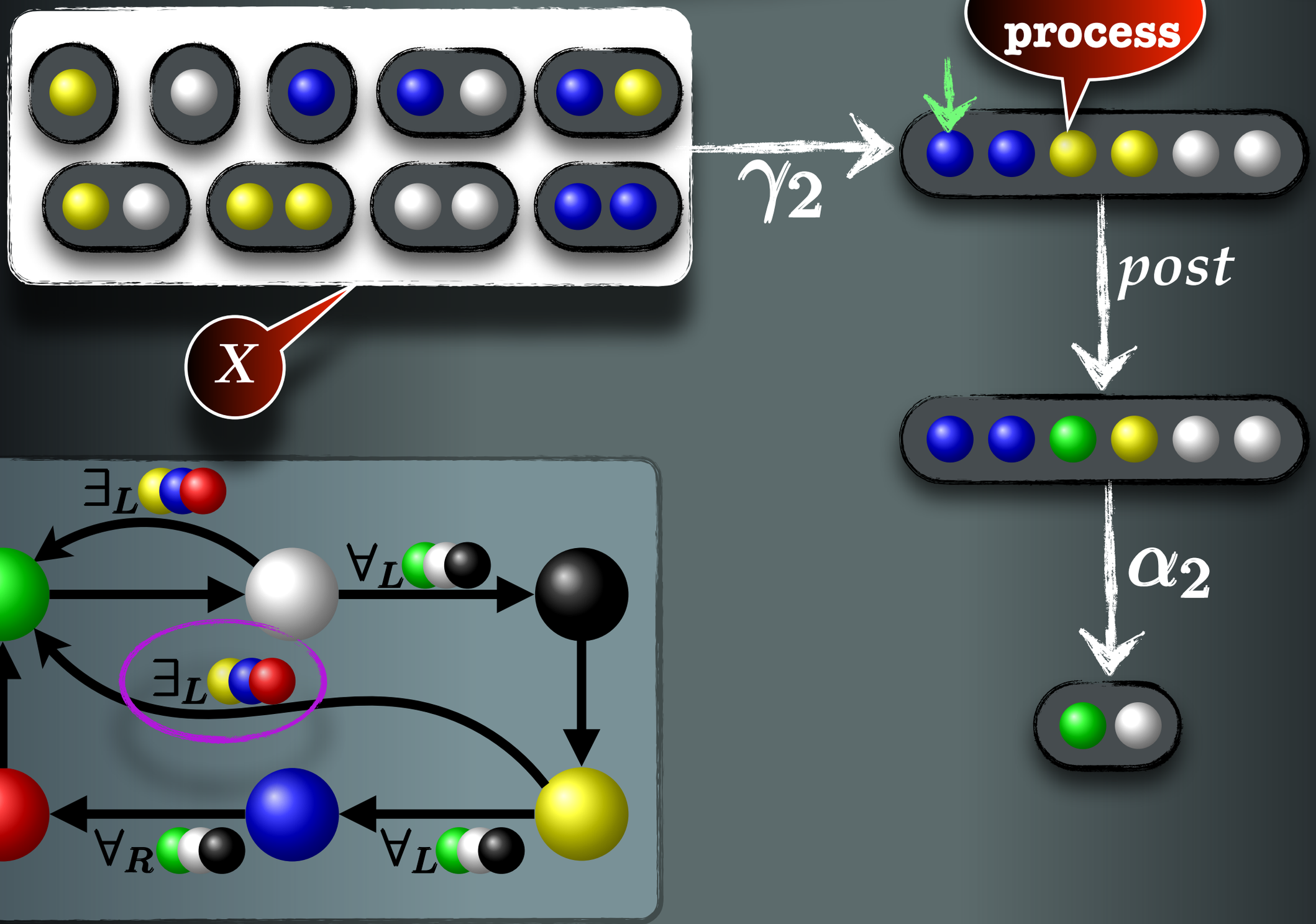
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



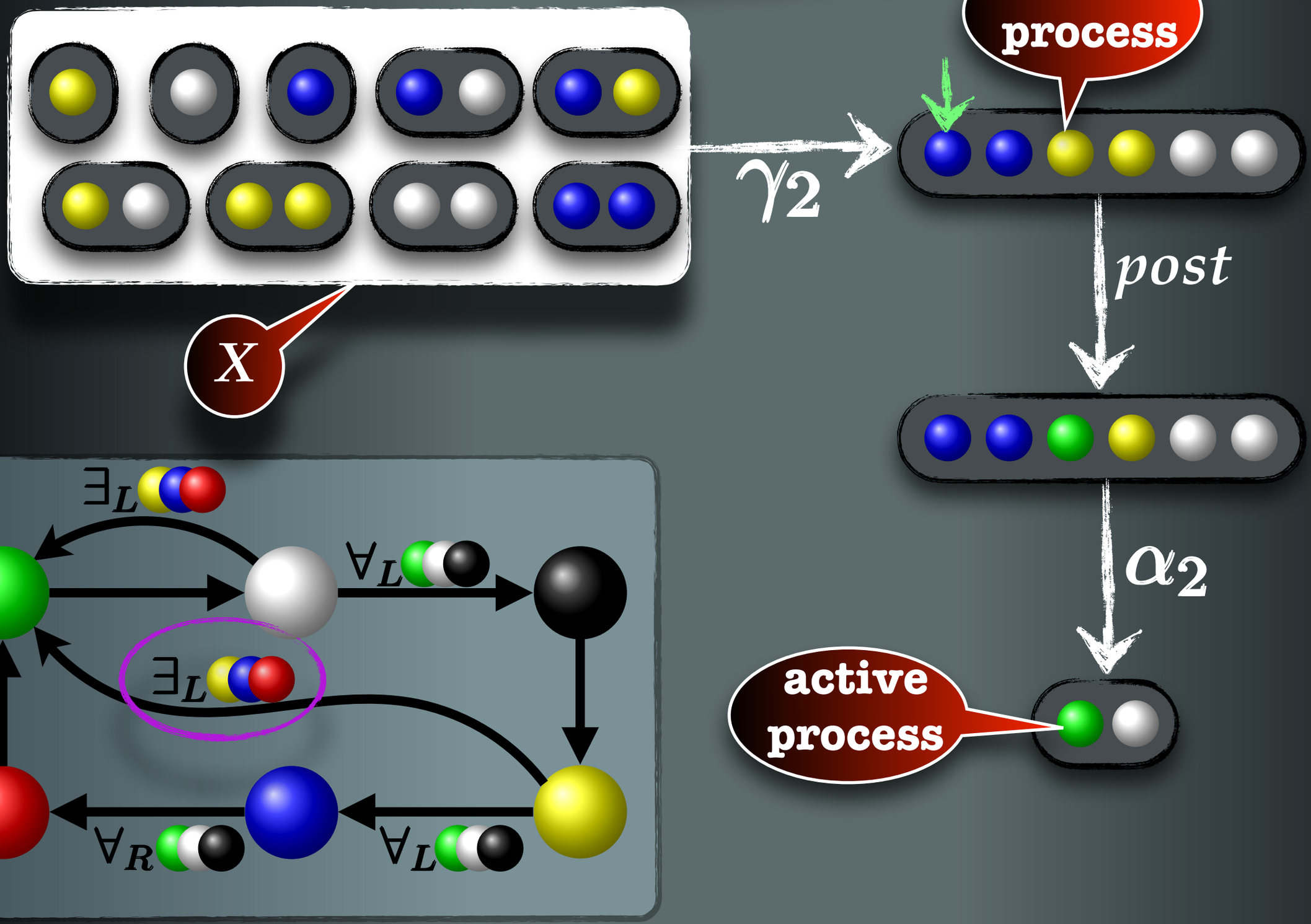
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



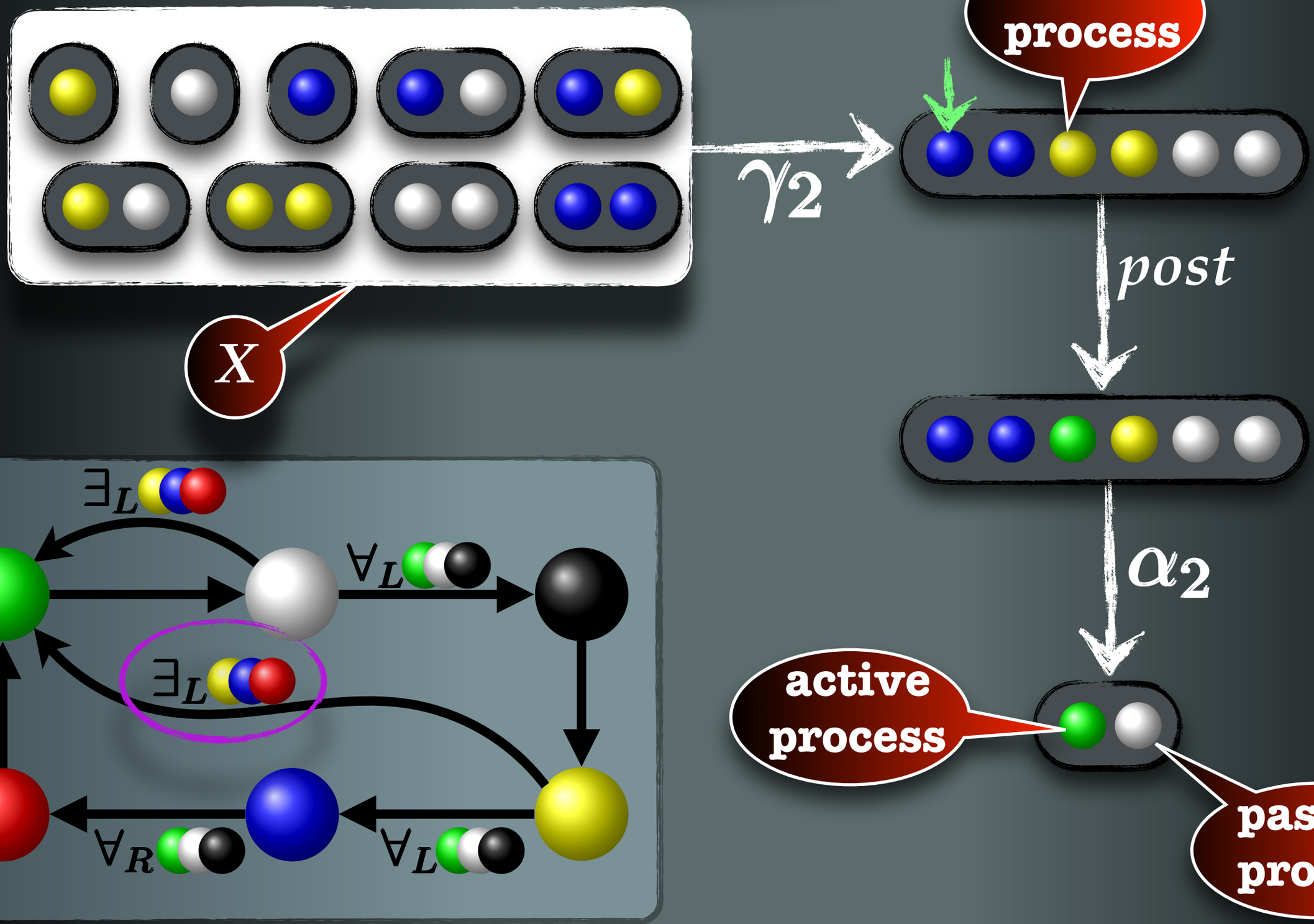
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



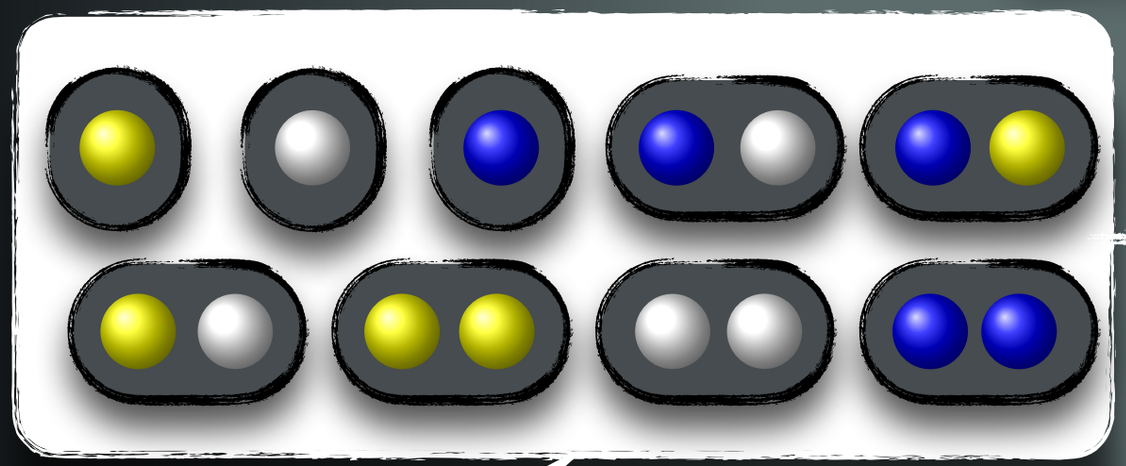
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



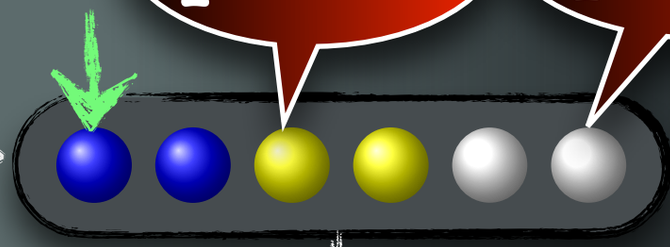
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



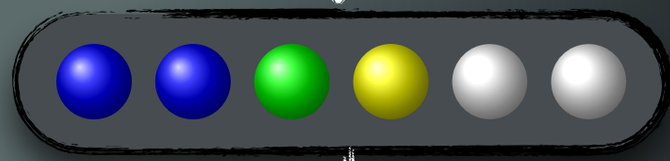
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



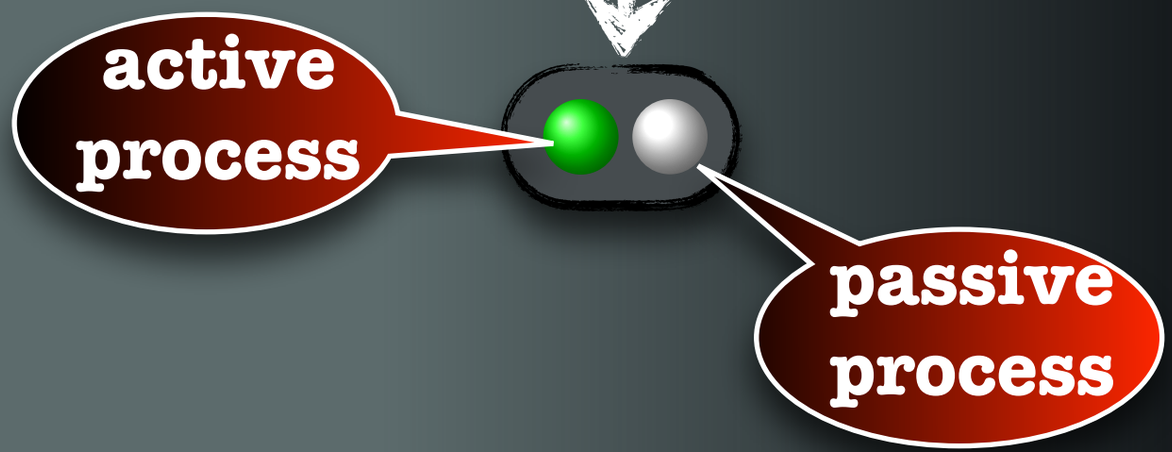
γ_2



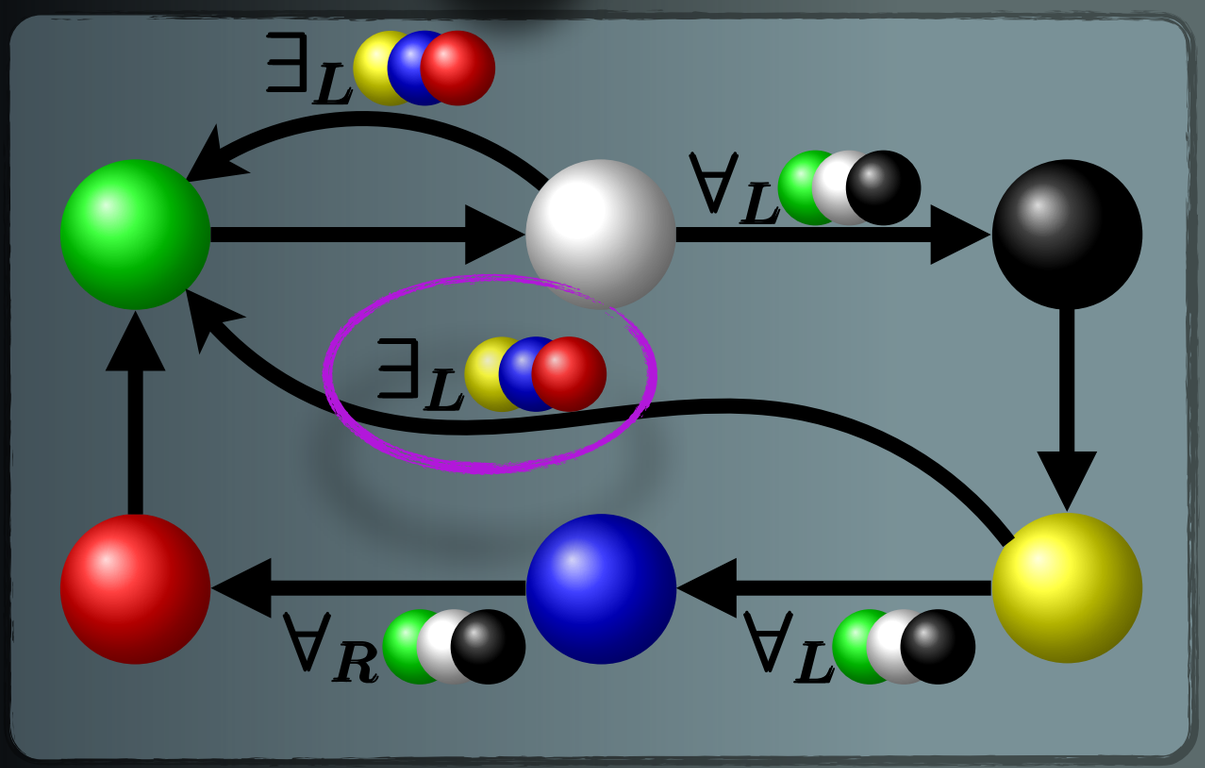
$post$



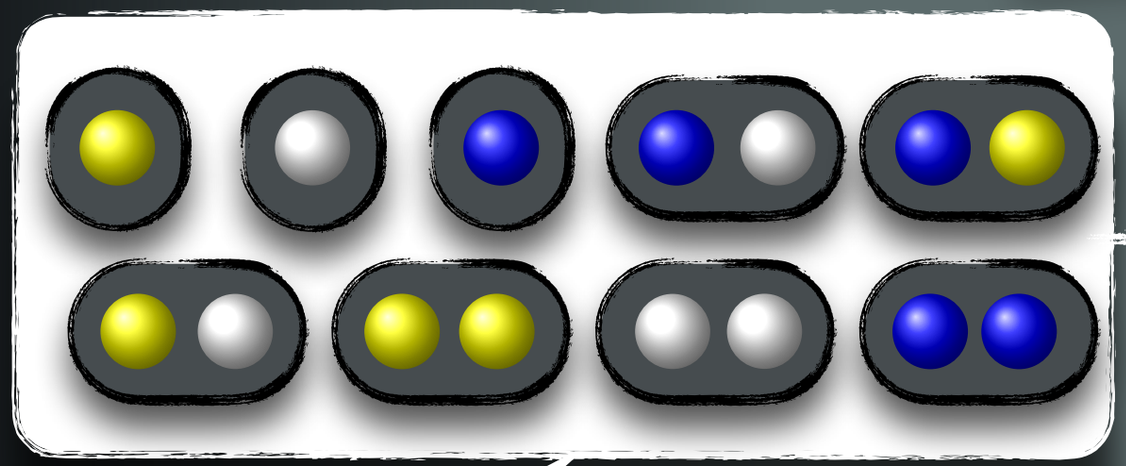
α_2



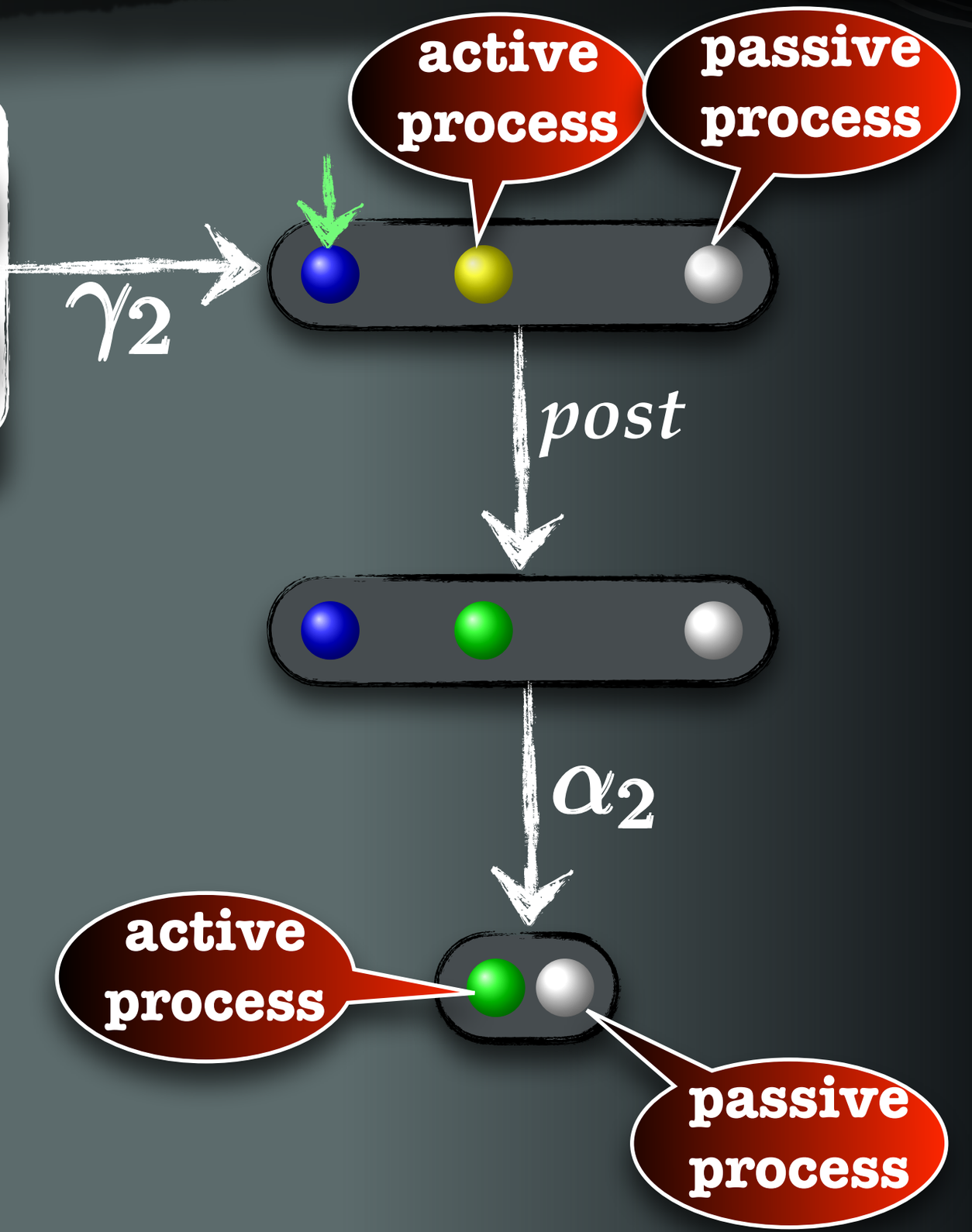
X

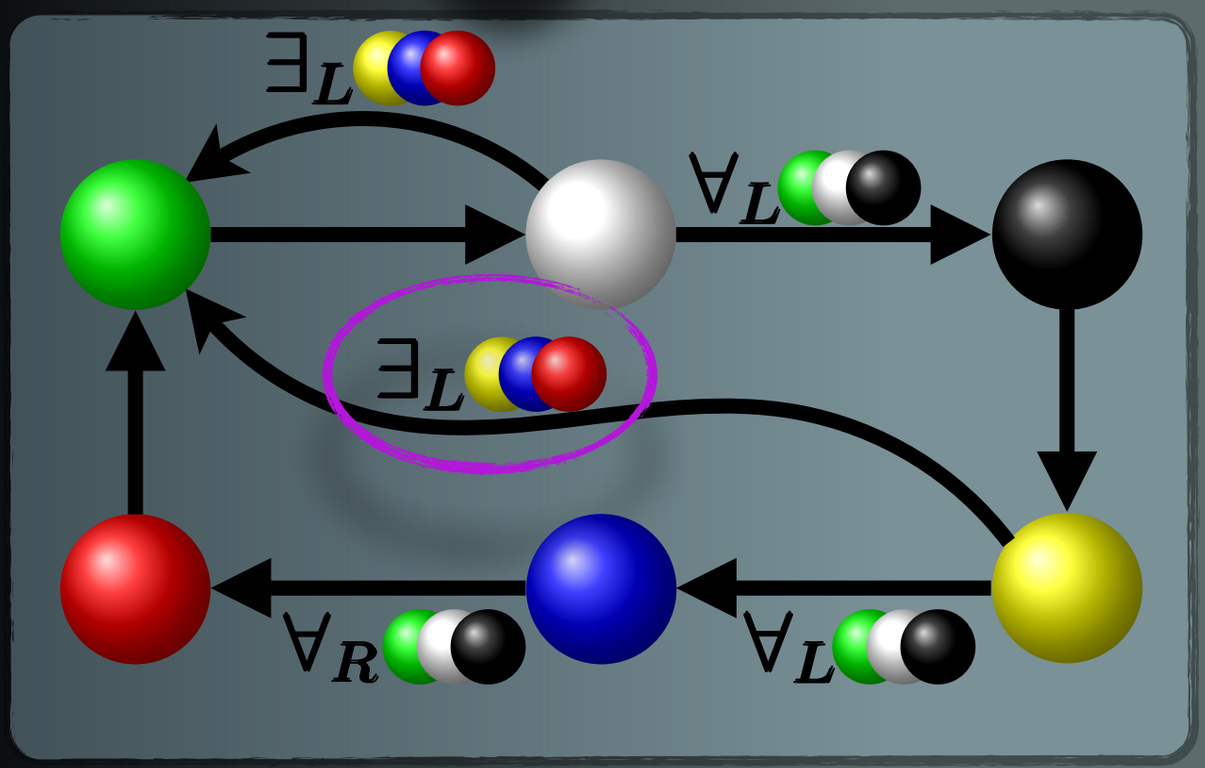


$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$

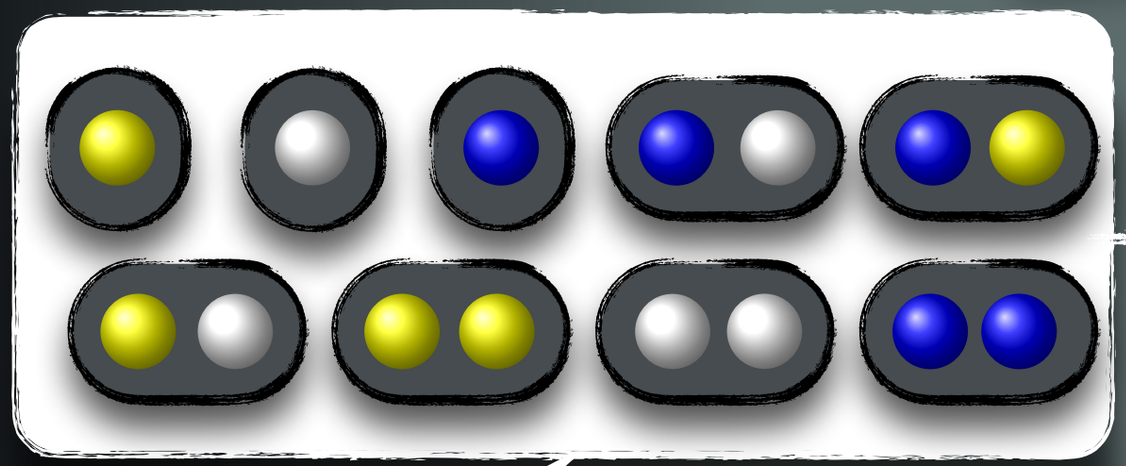


X

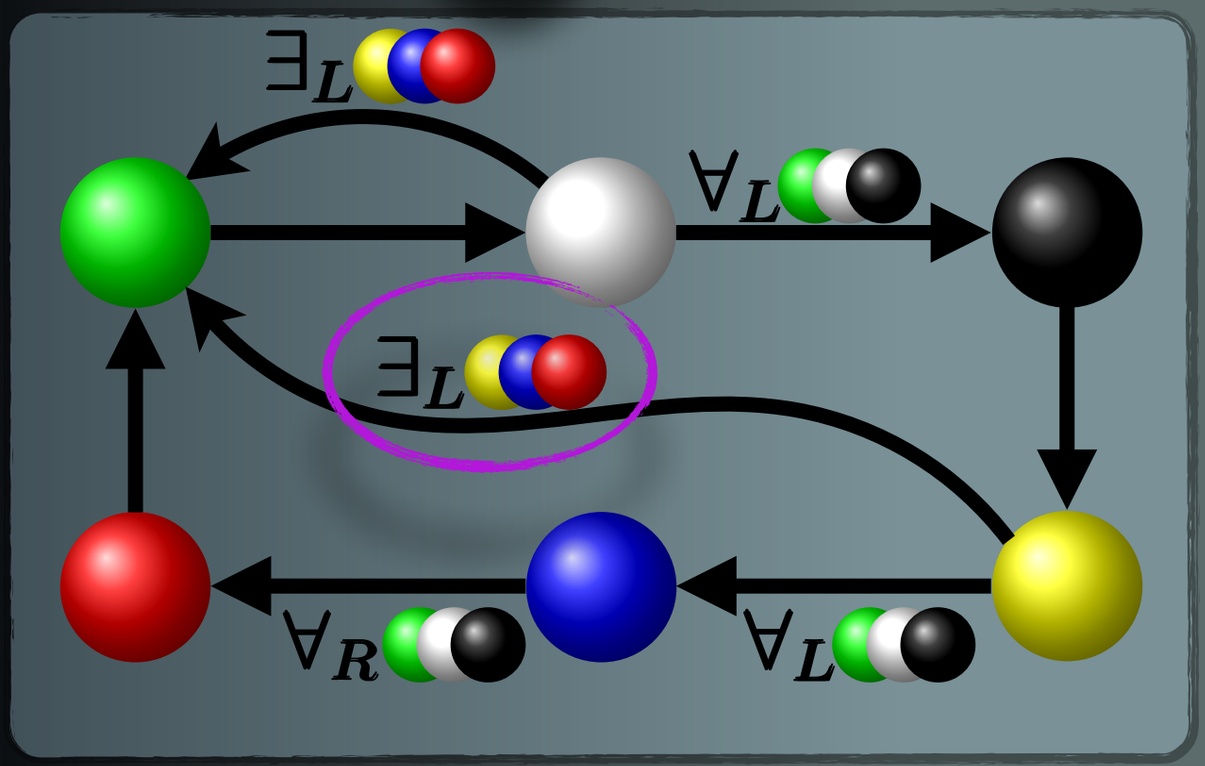




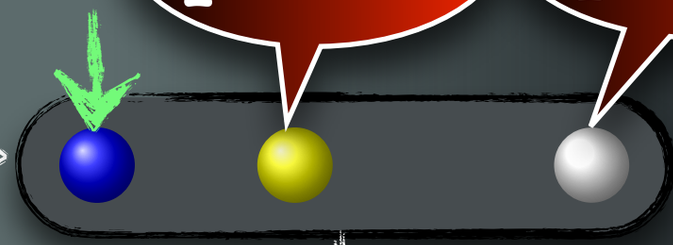
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



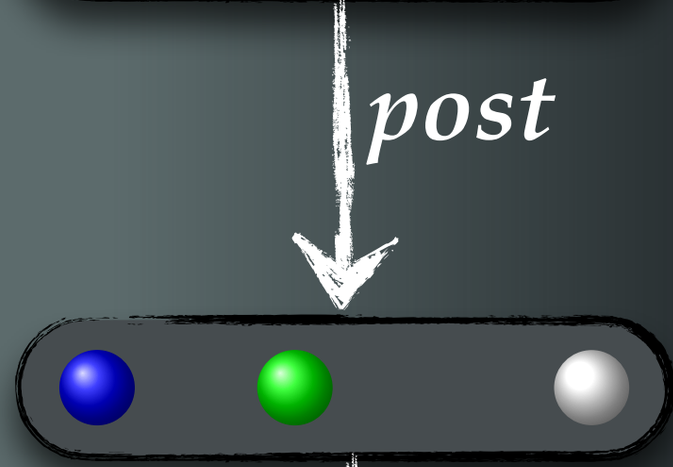
X



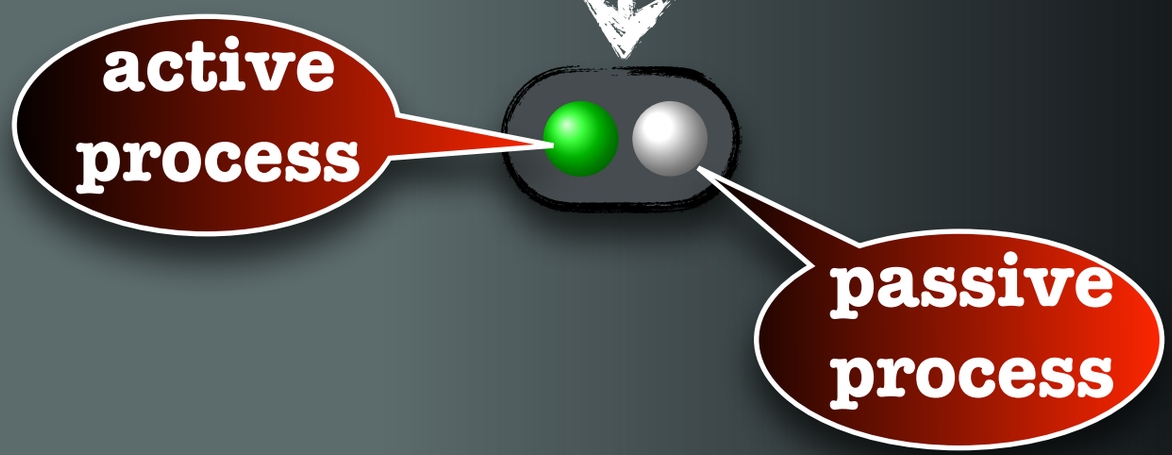
$\oint_2^3$



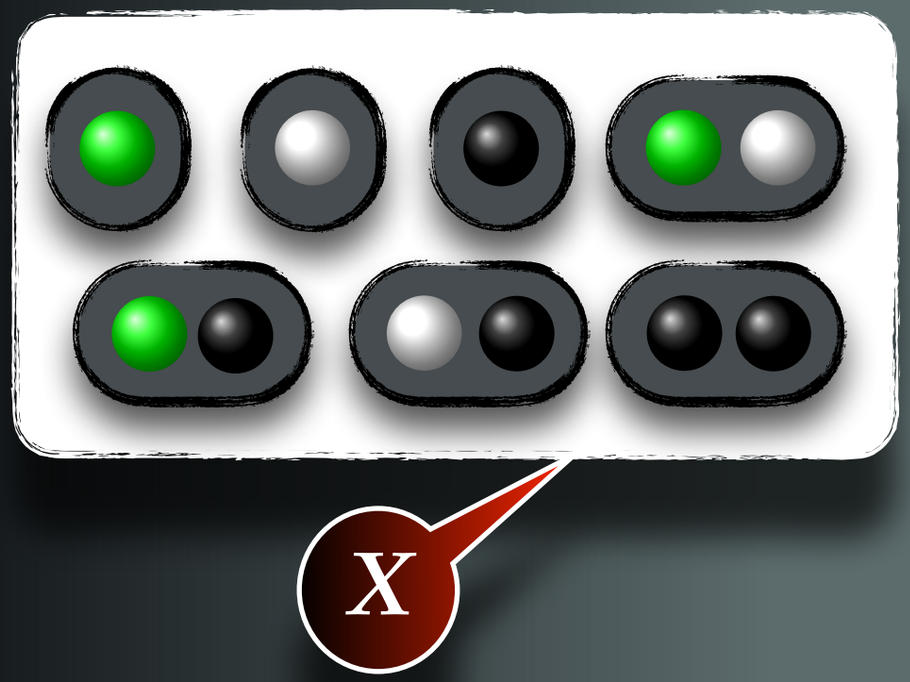
$post$



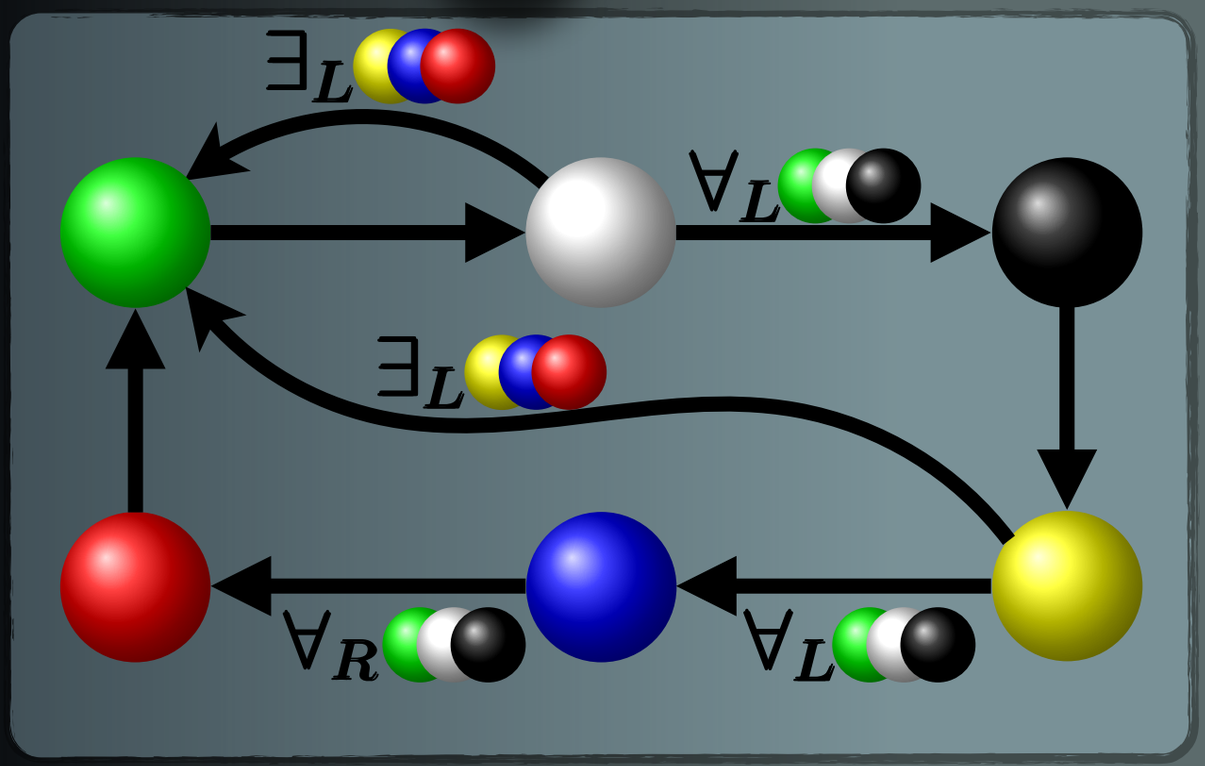
α_2



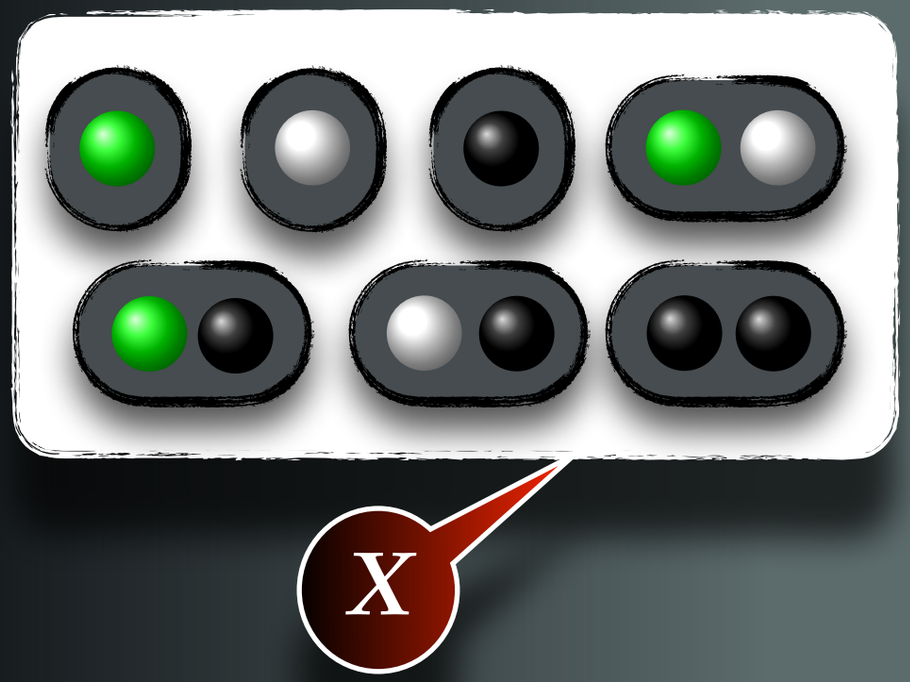
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



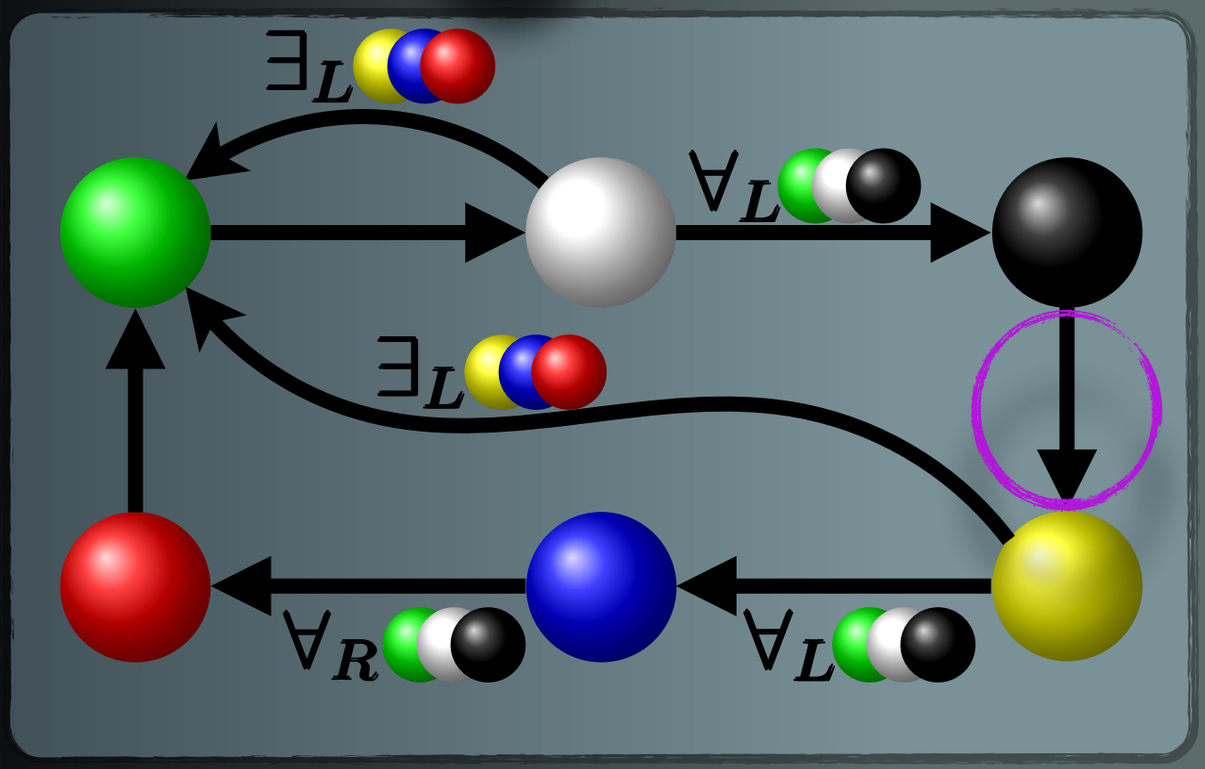
X



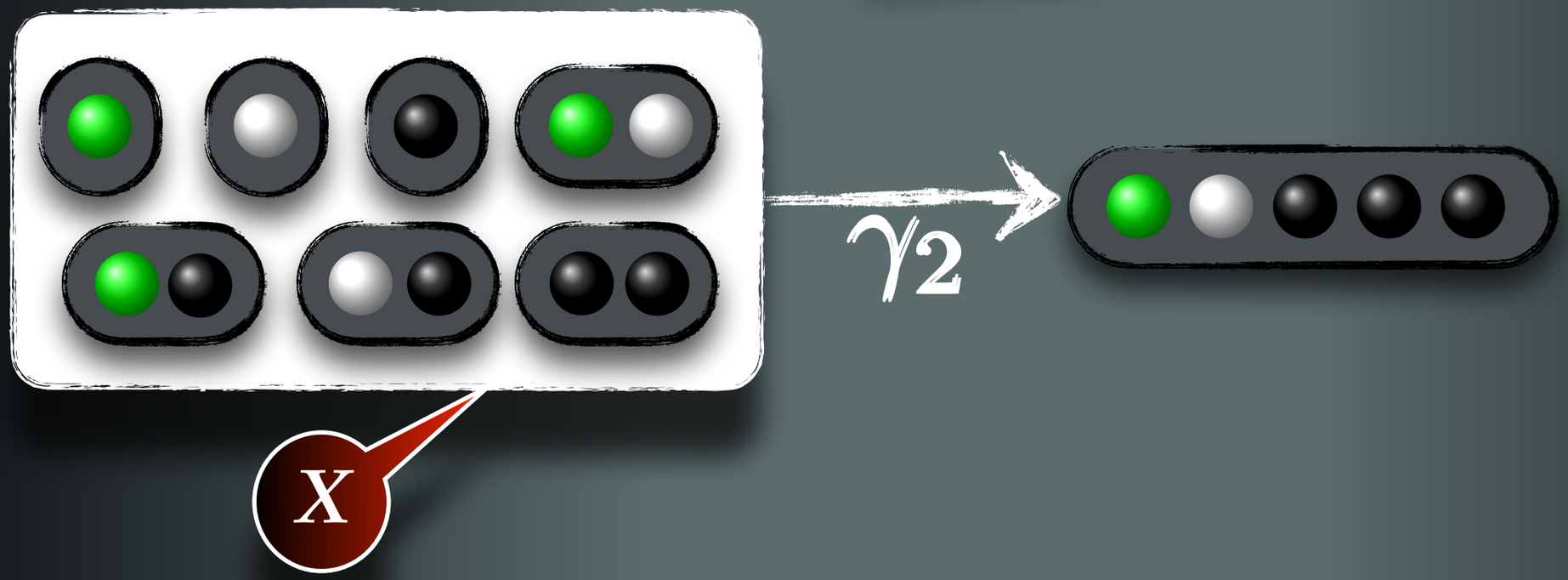
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$

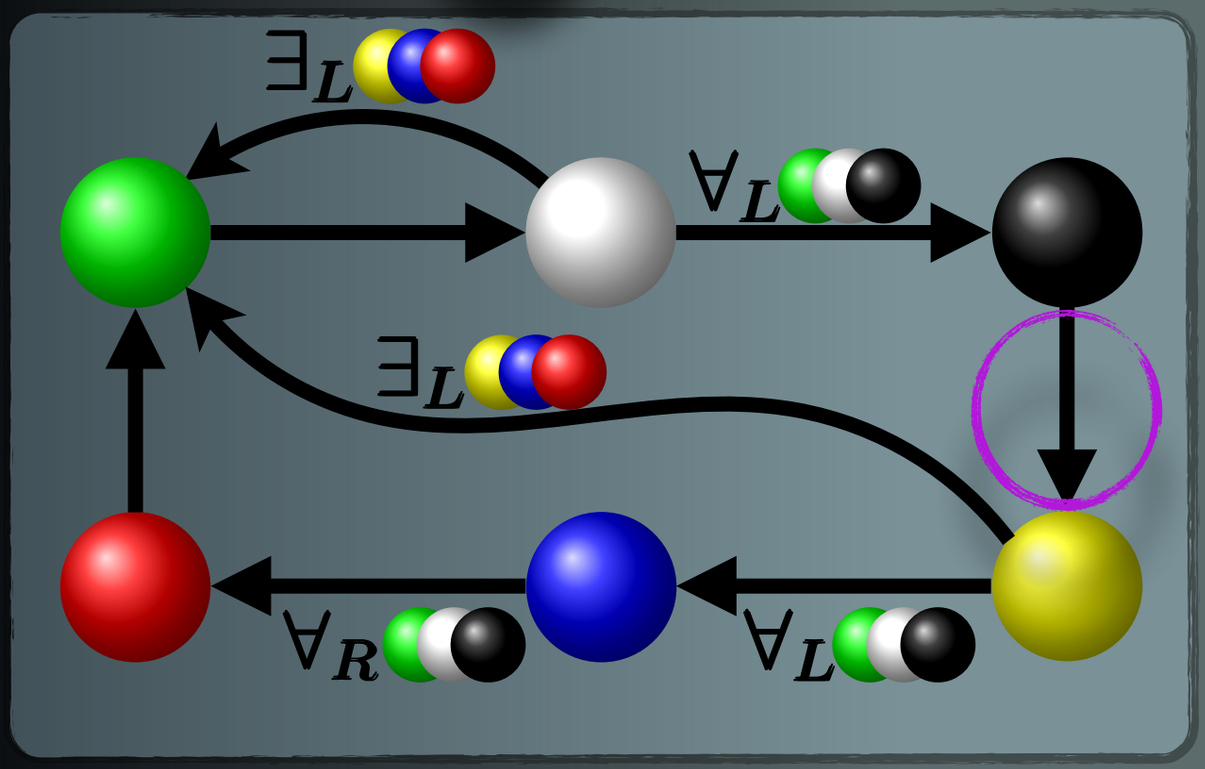


X

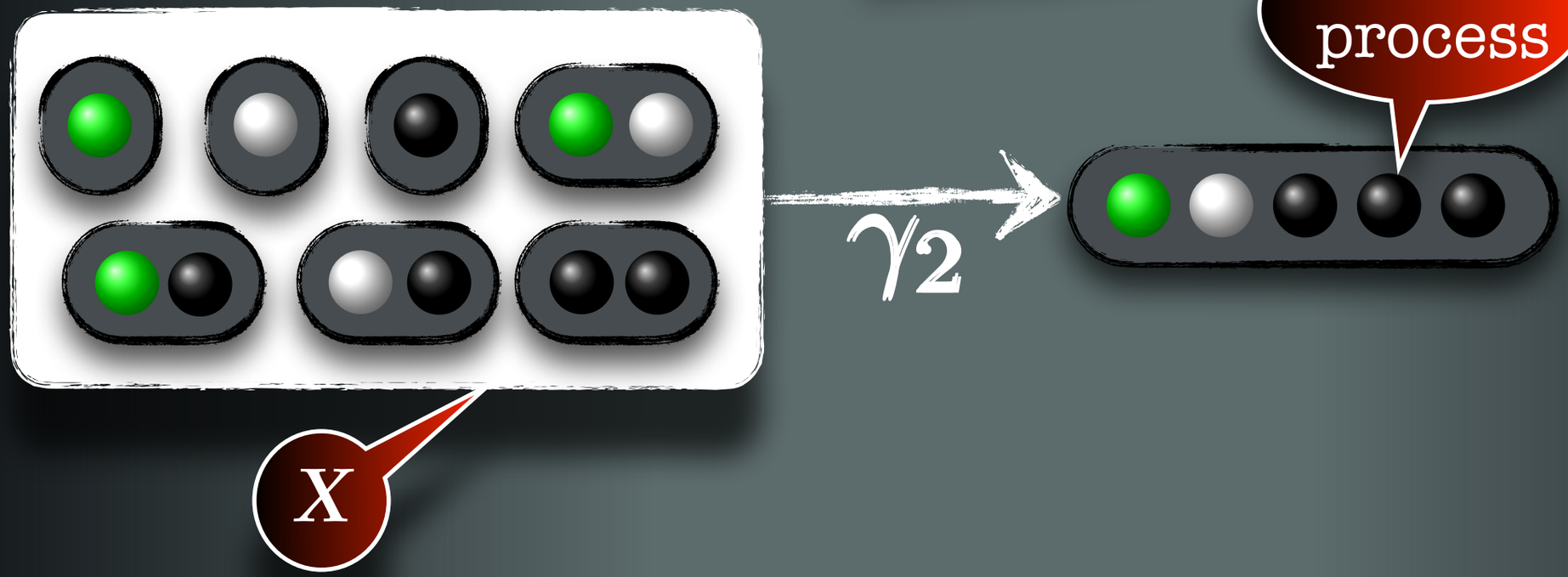


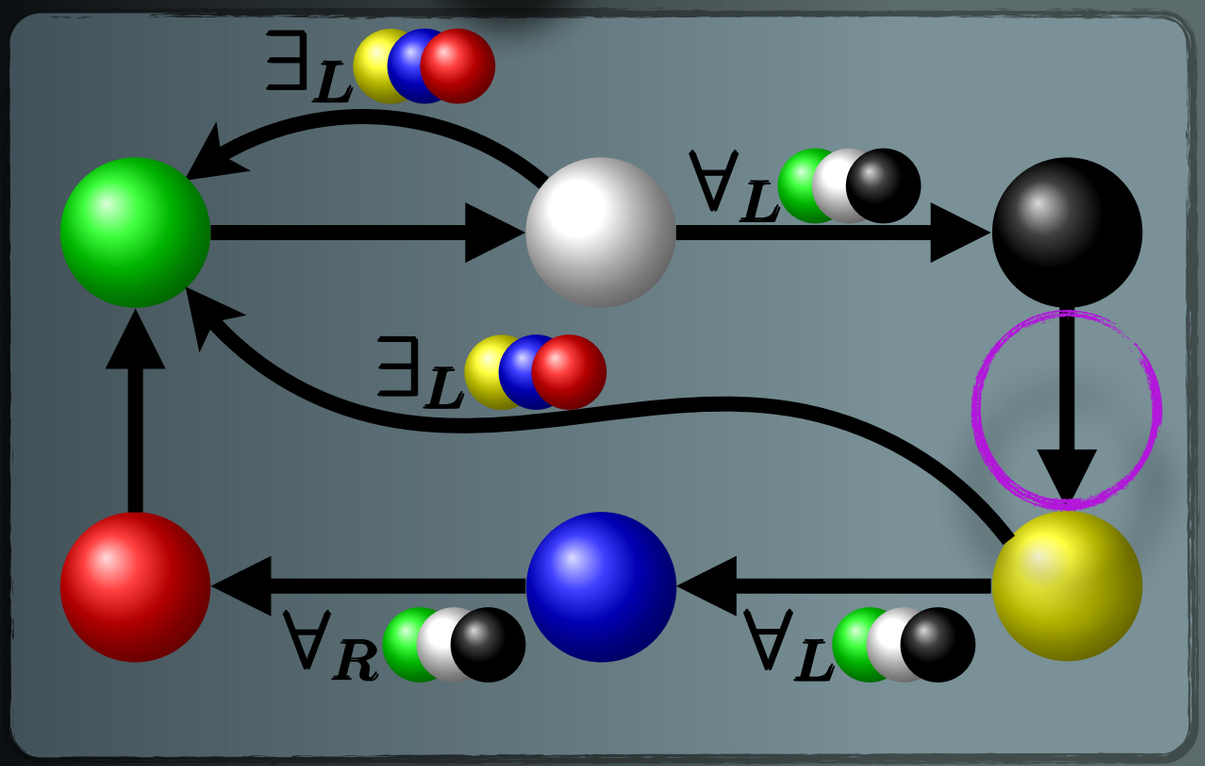
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



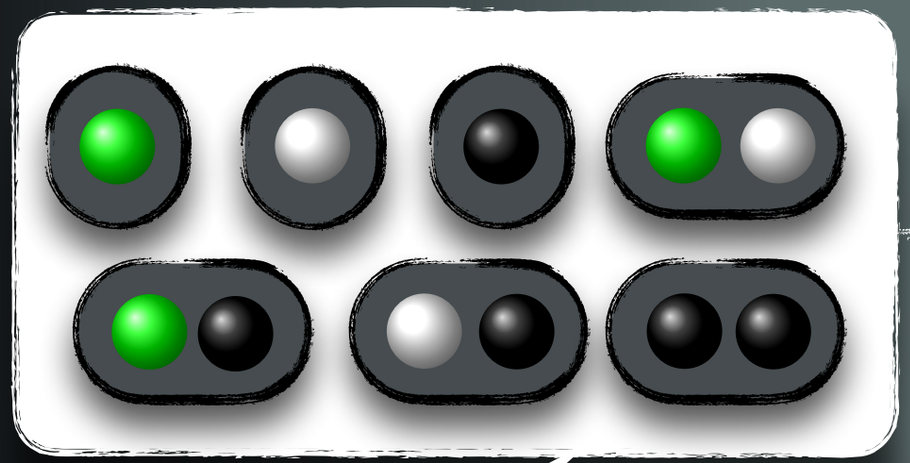


$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$

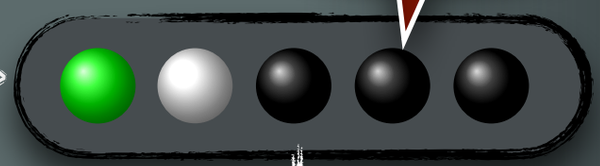




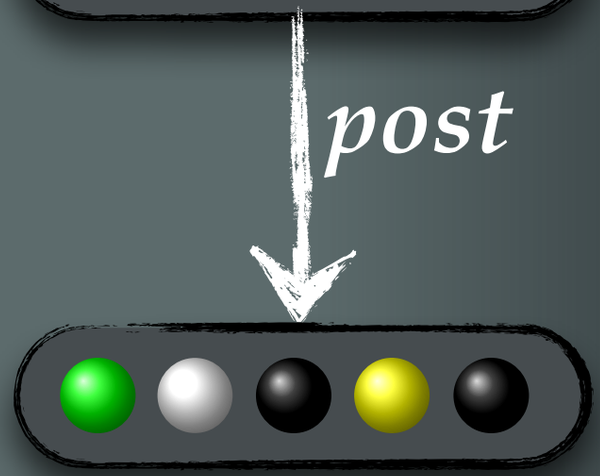
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



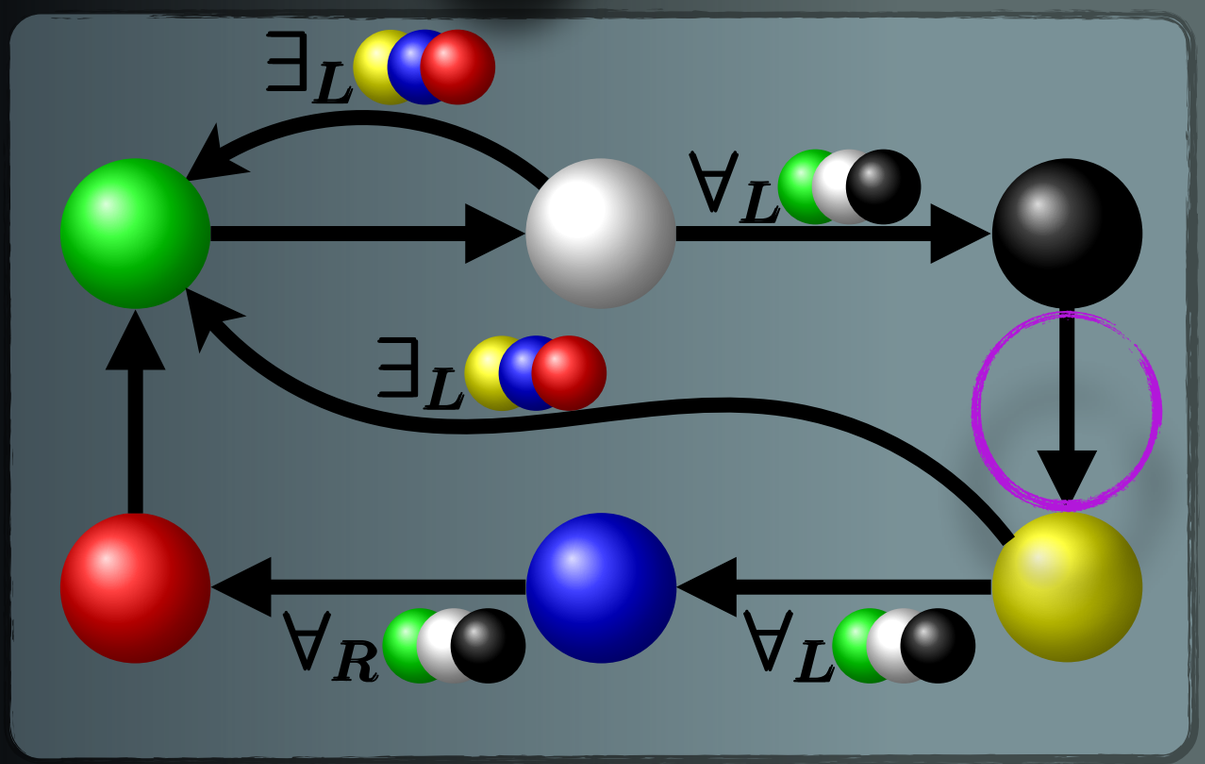
γ_2



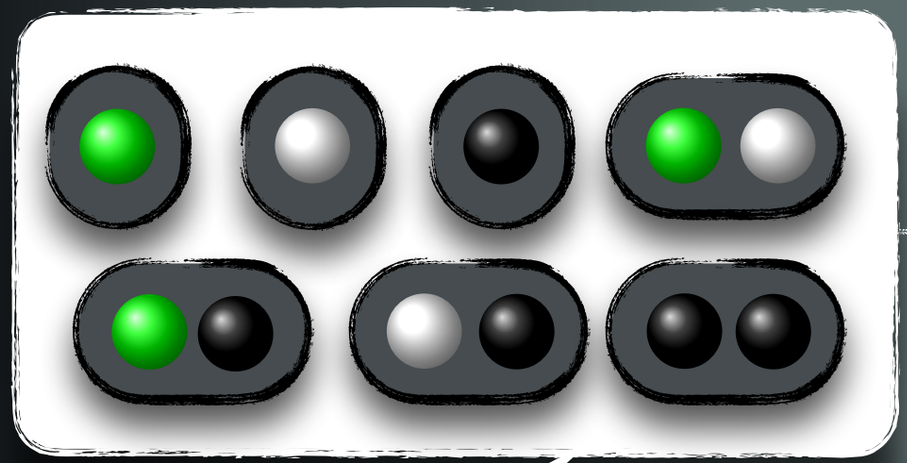
post



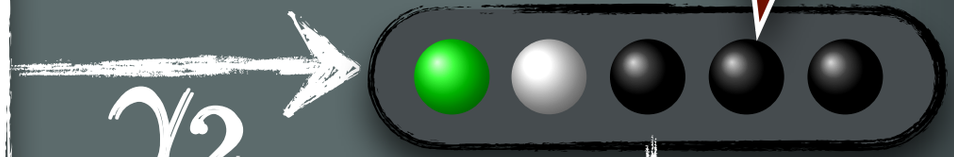
X



$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



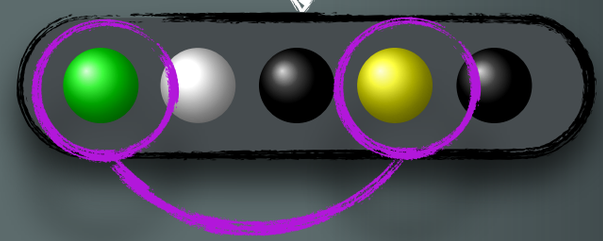
X

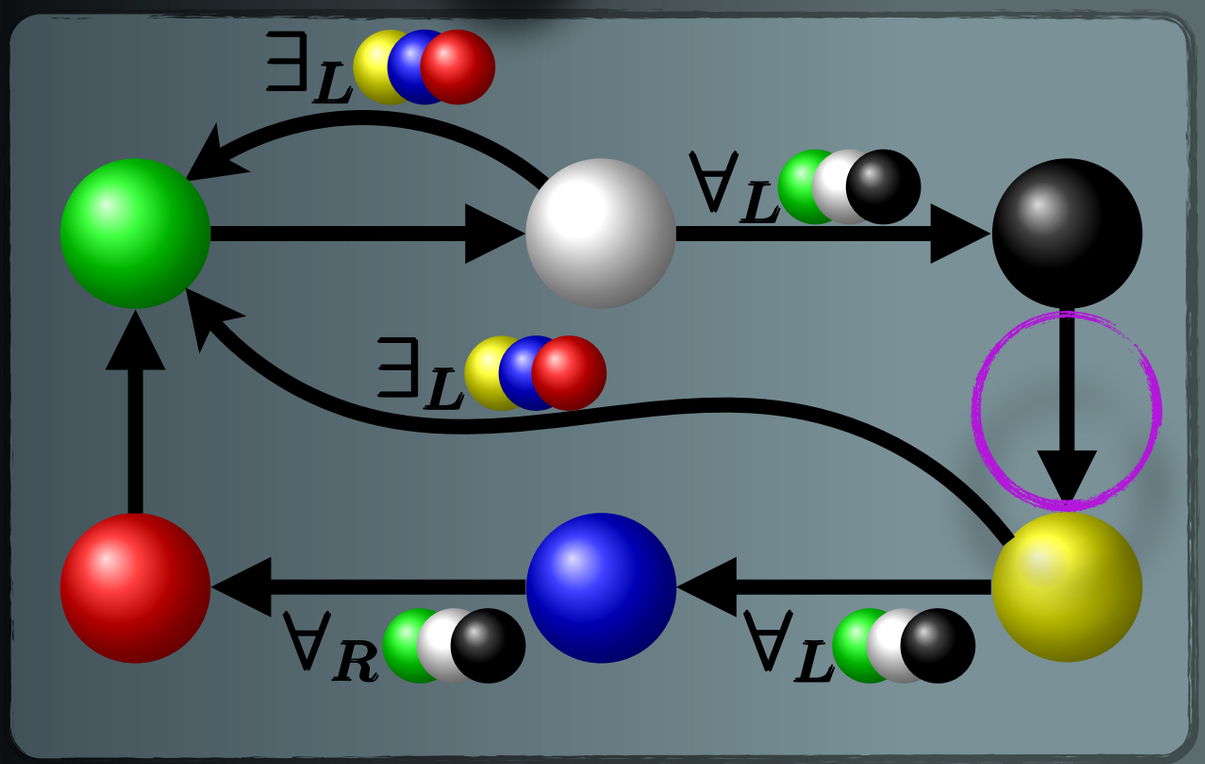


γ_2

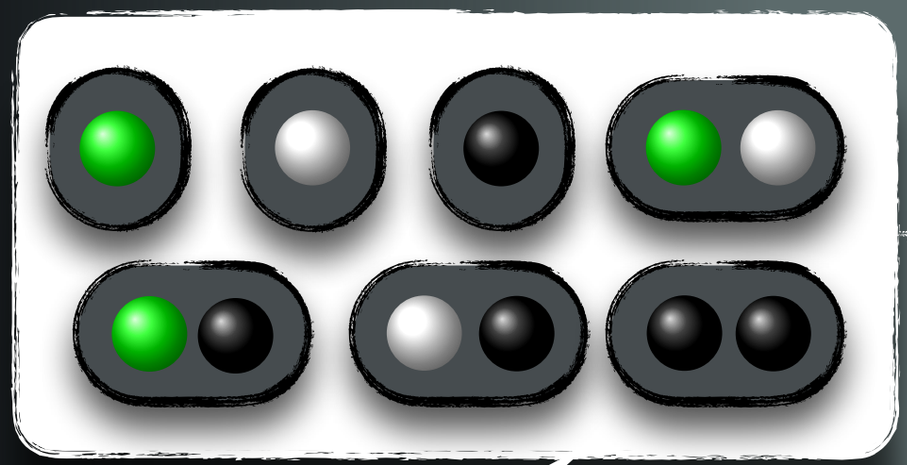
active process

$post$



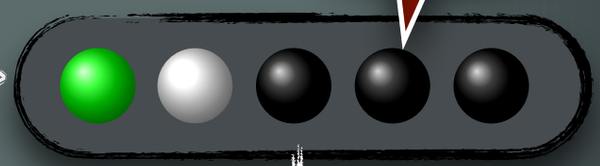


$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$

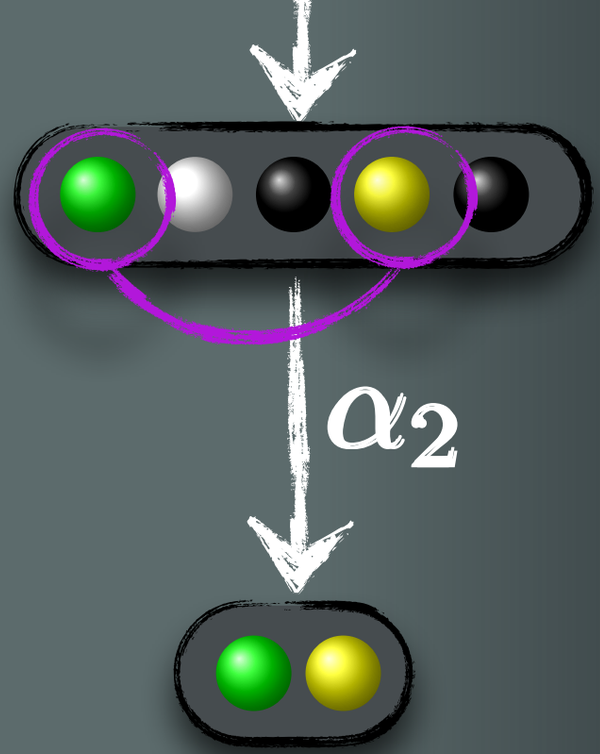


X

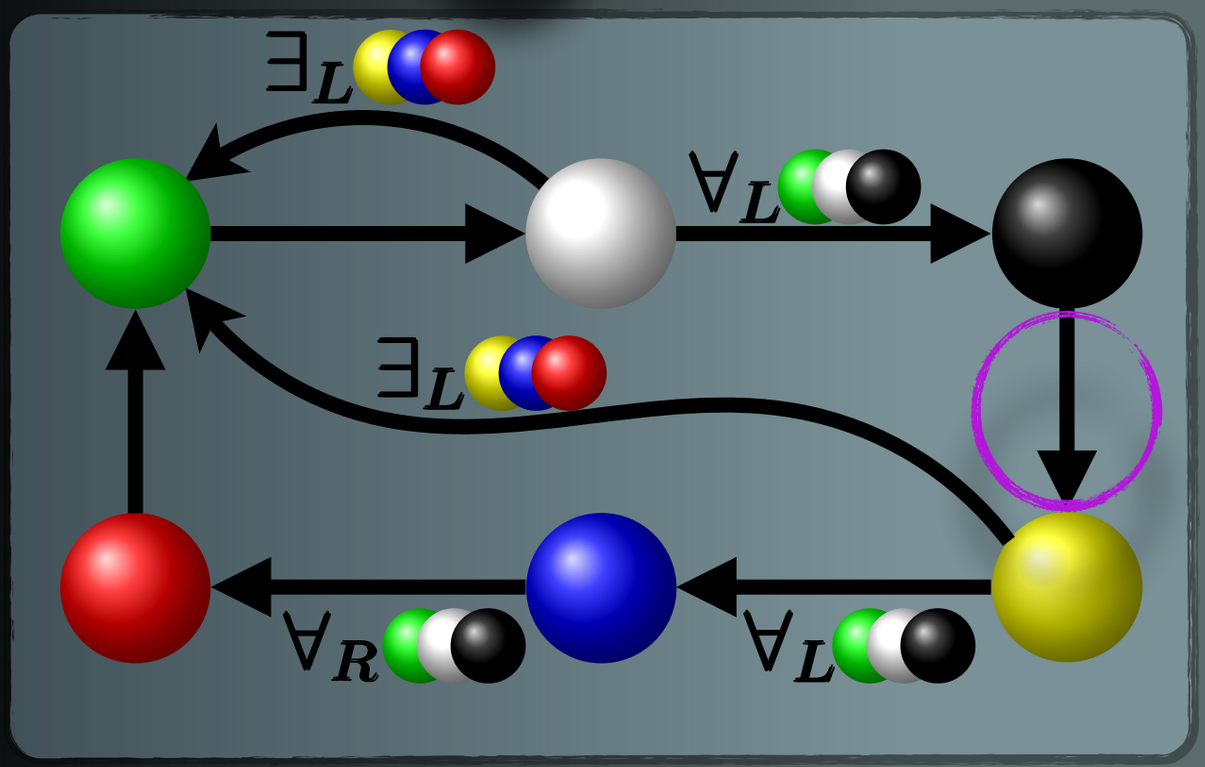
γ_2



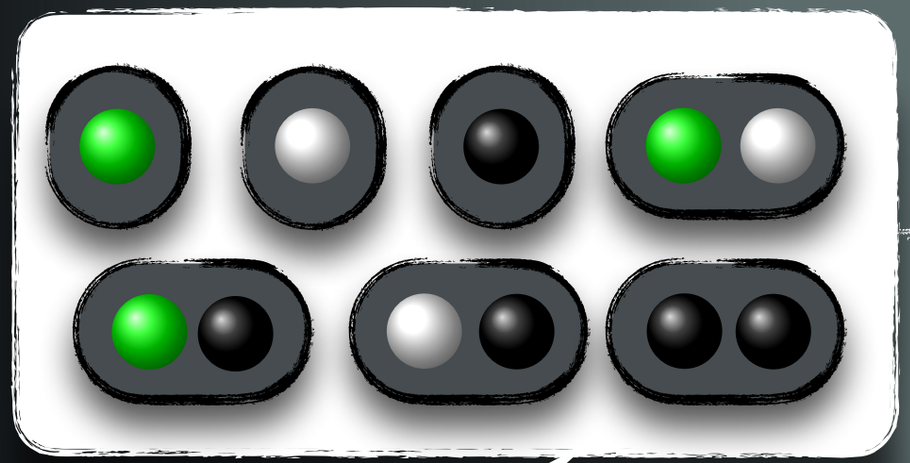
$post$



α_2

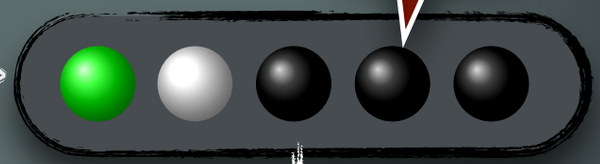


$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$

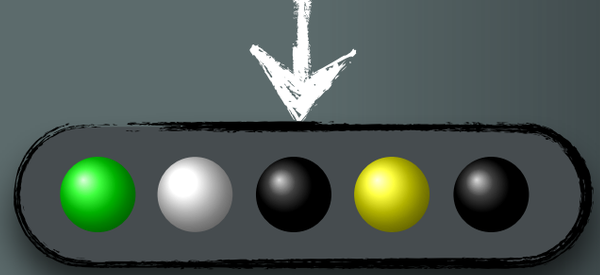


X

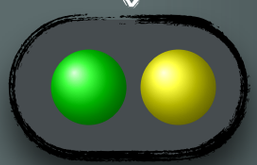
γ_2

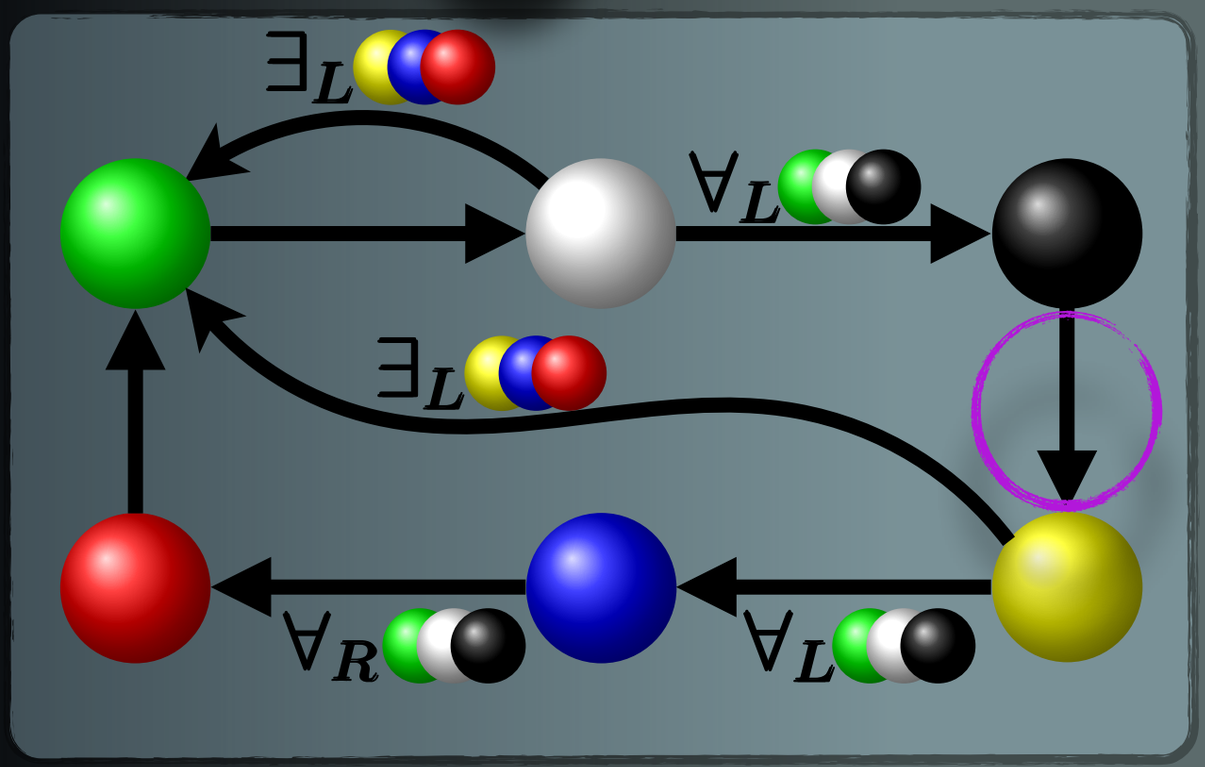


$post$

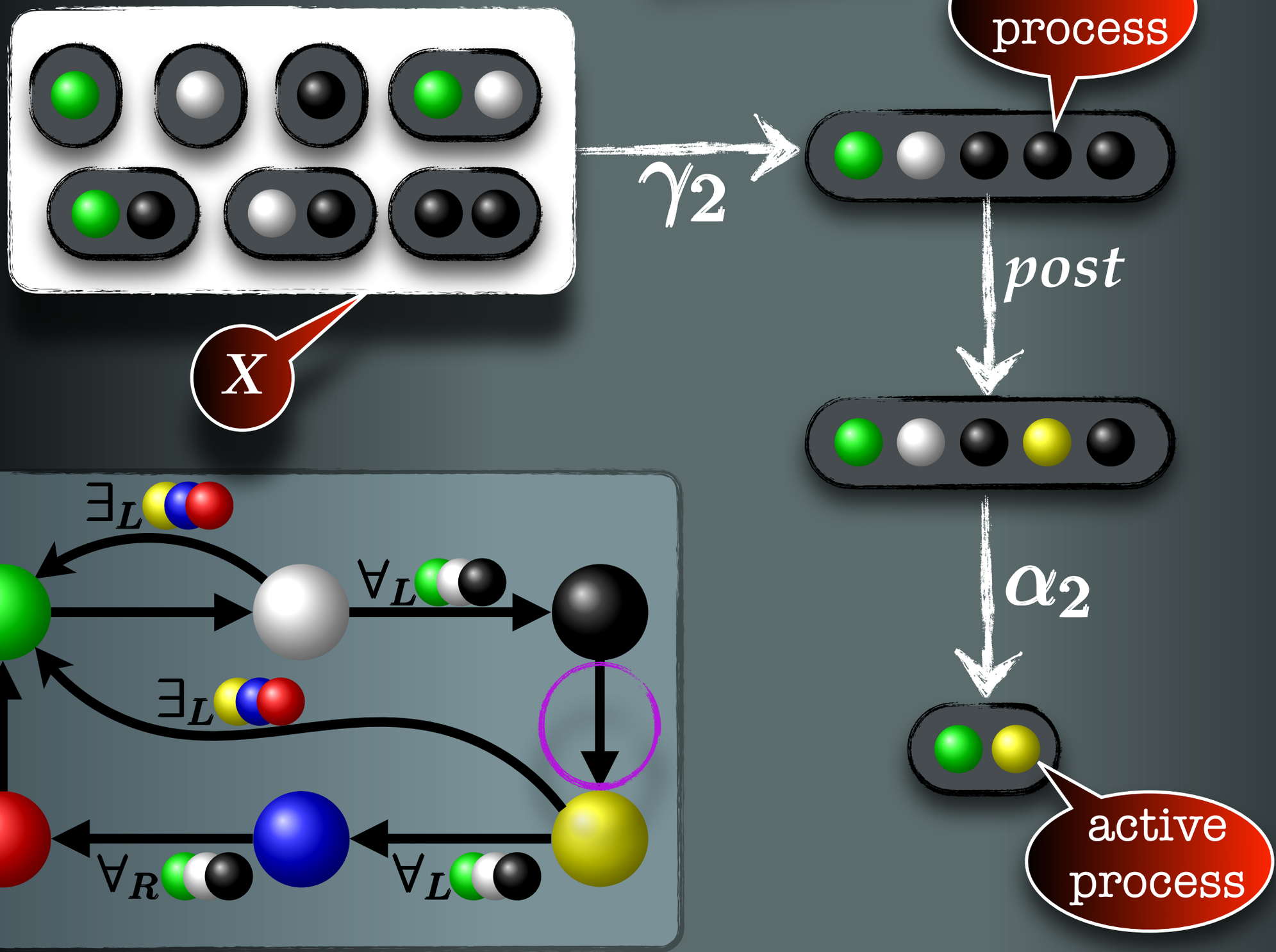


α_2

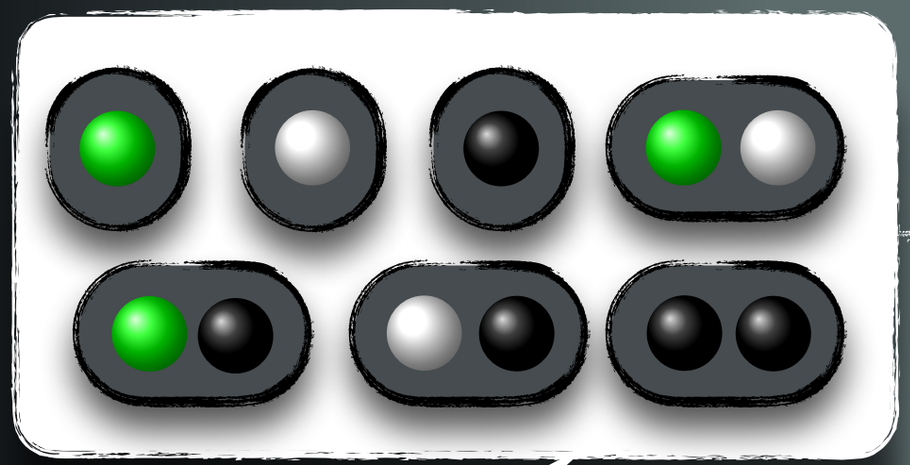




$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$

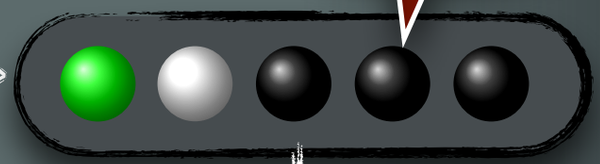


$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



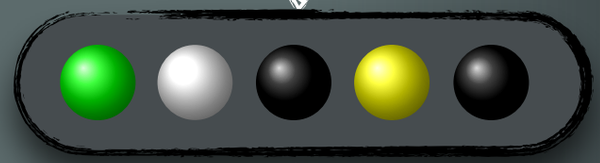
X

γ_2

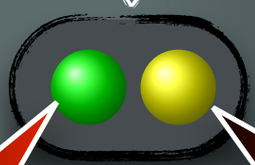


active process

$post$

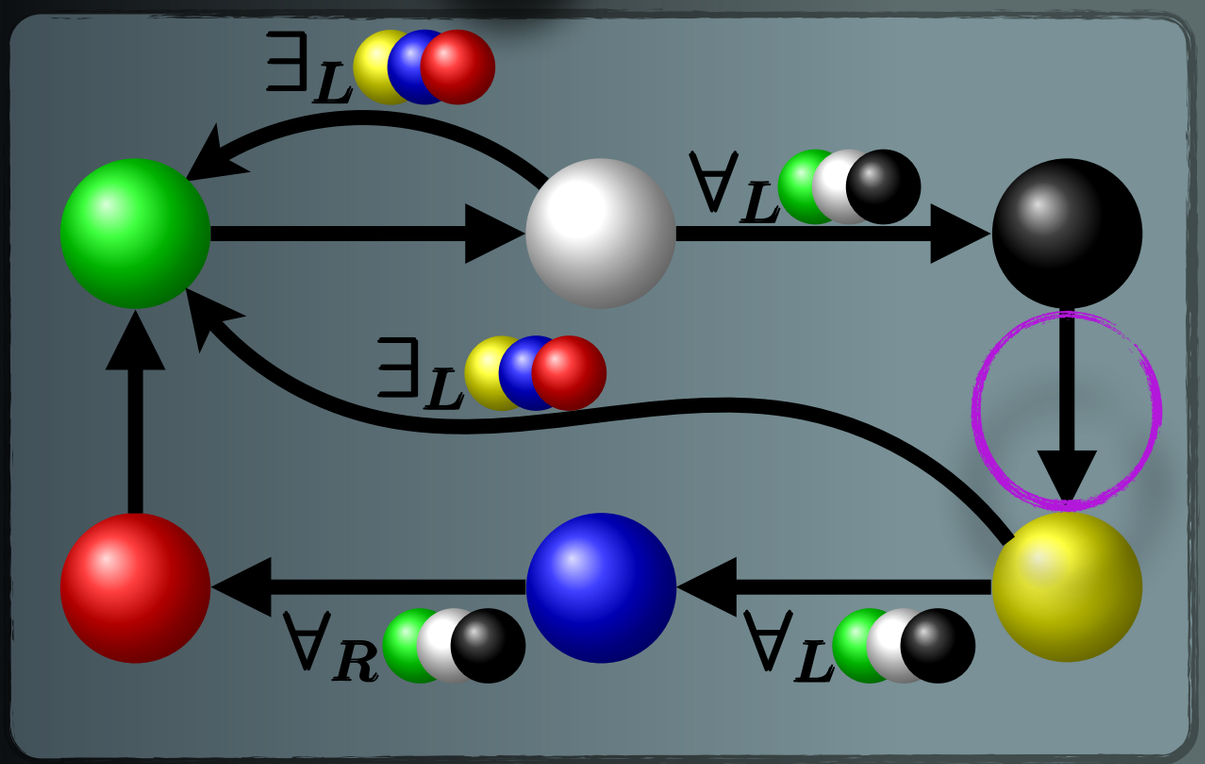


α_2

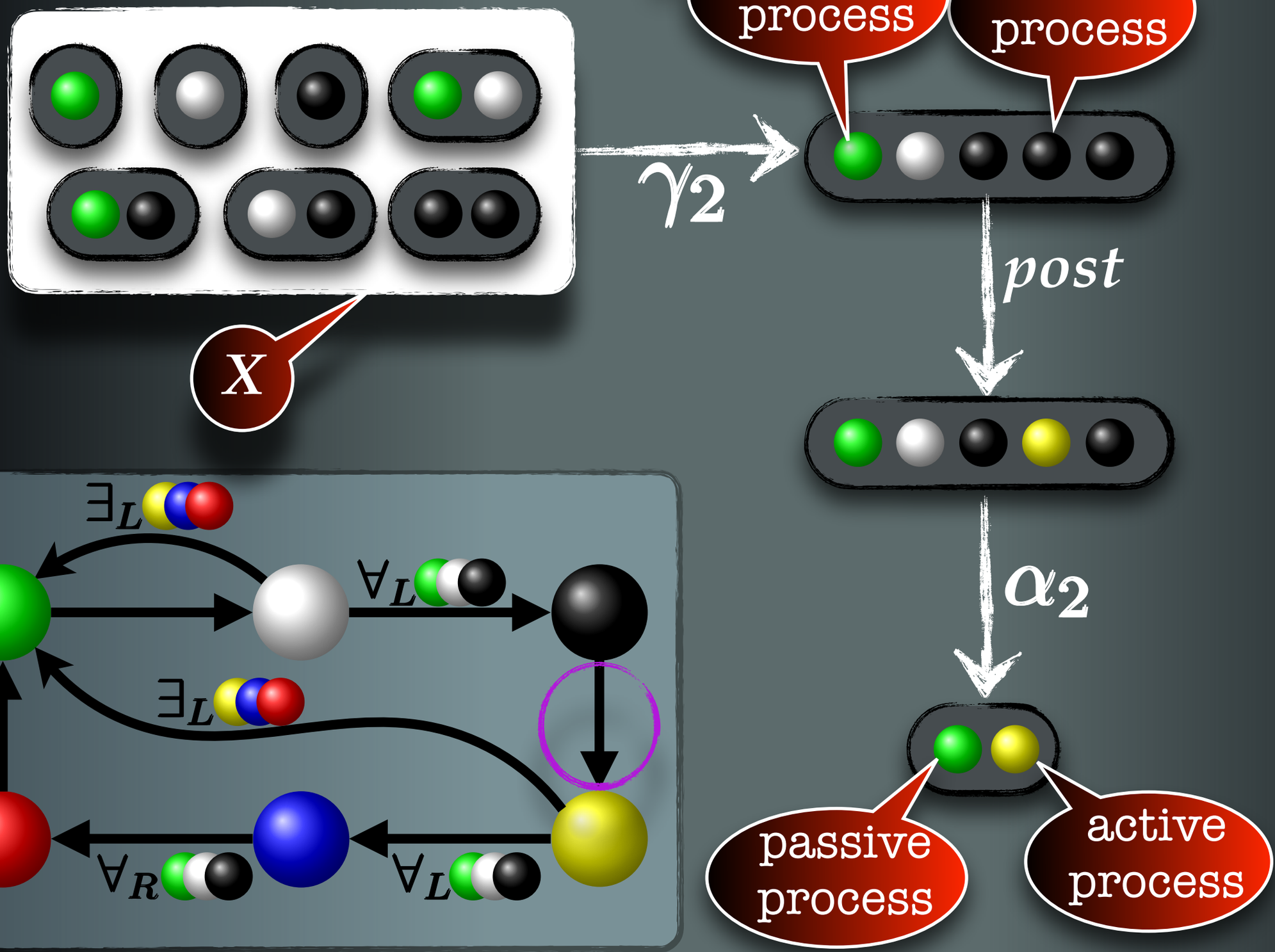


passive process

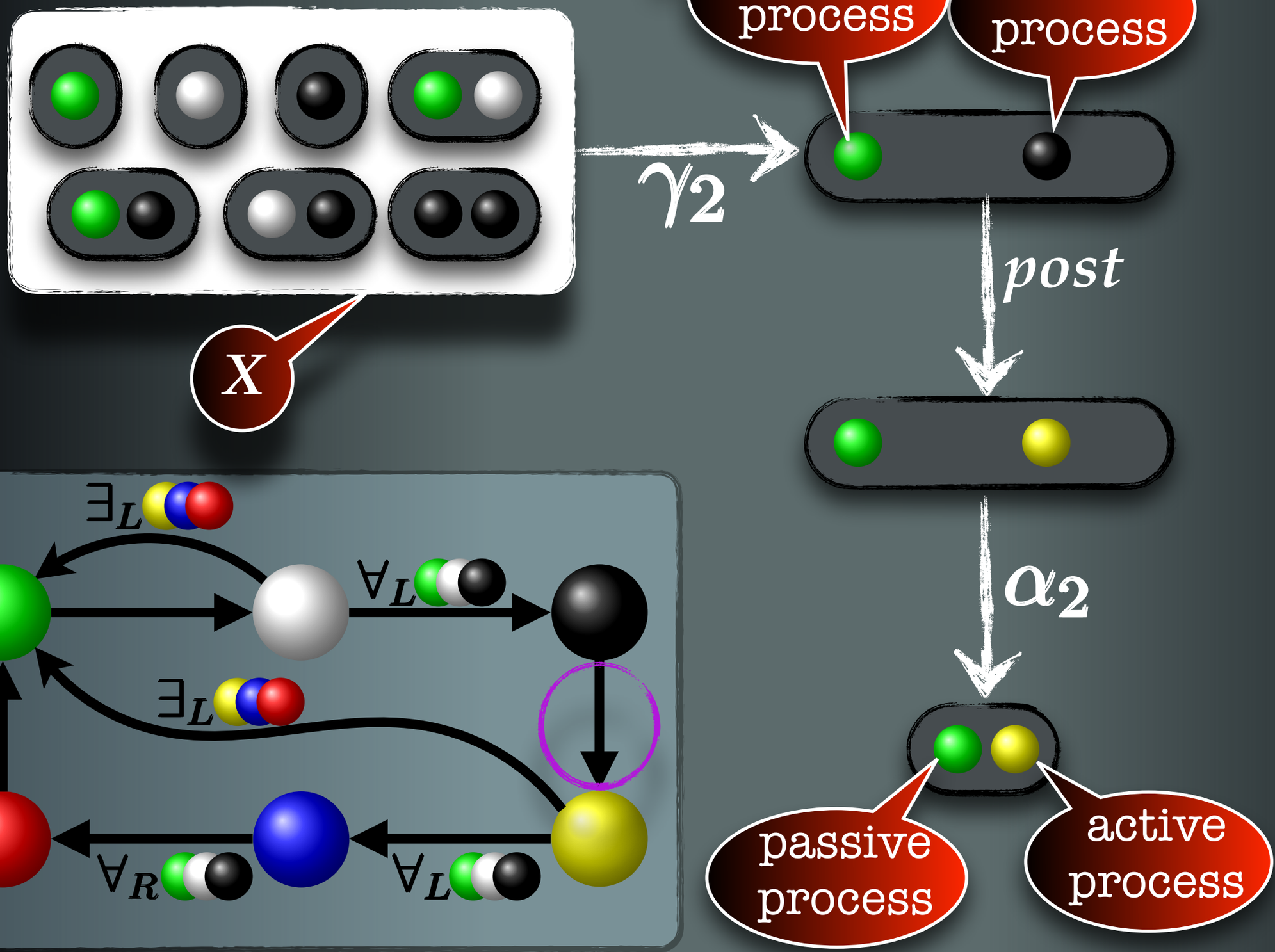
active process



$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$

passive process

active process

$\oint_2^2$

$post$

X

$\exists_L$

$\forall_L$

$\exists_L$

$\forall_R$

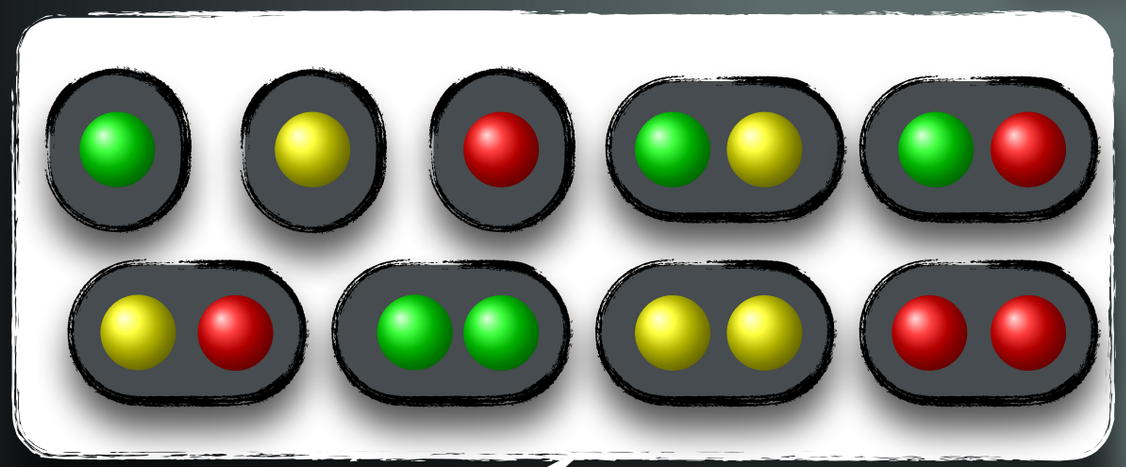
$\forall_L$

α_2

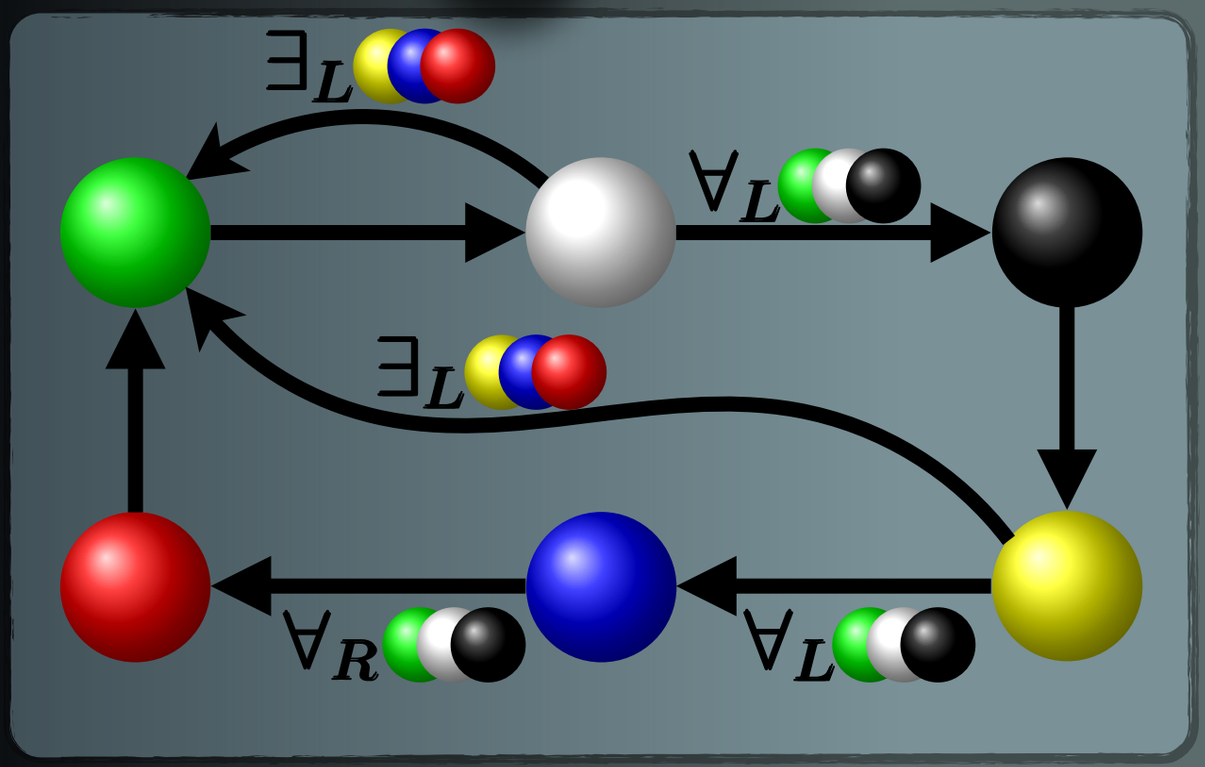
passive process

active process

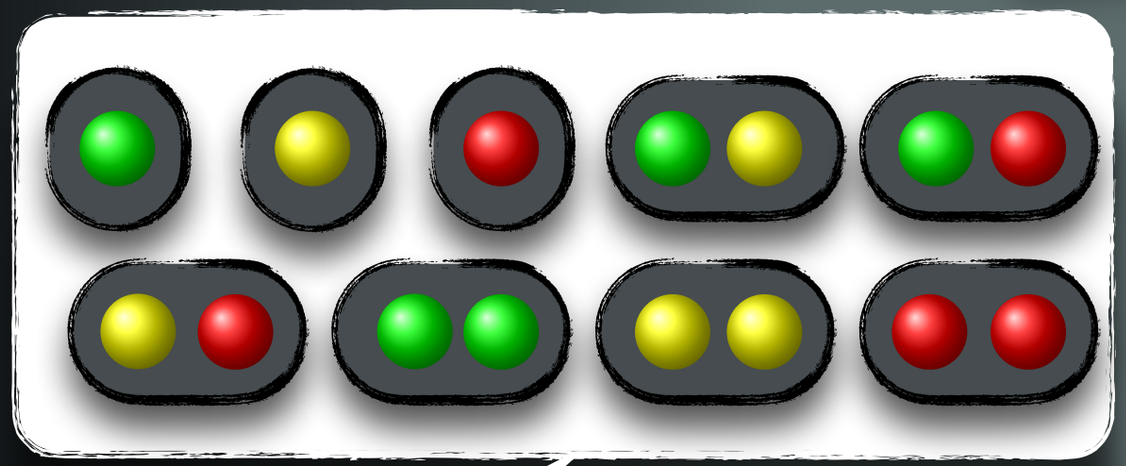
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



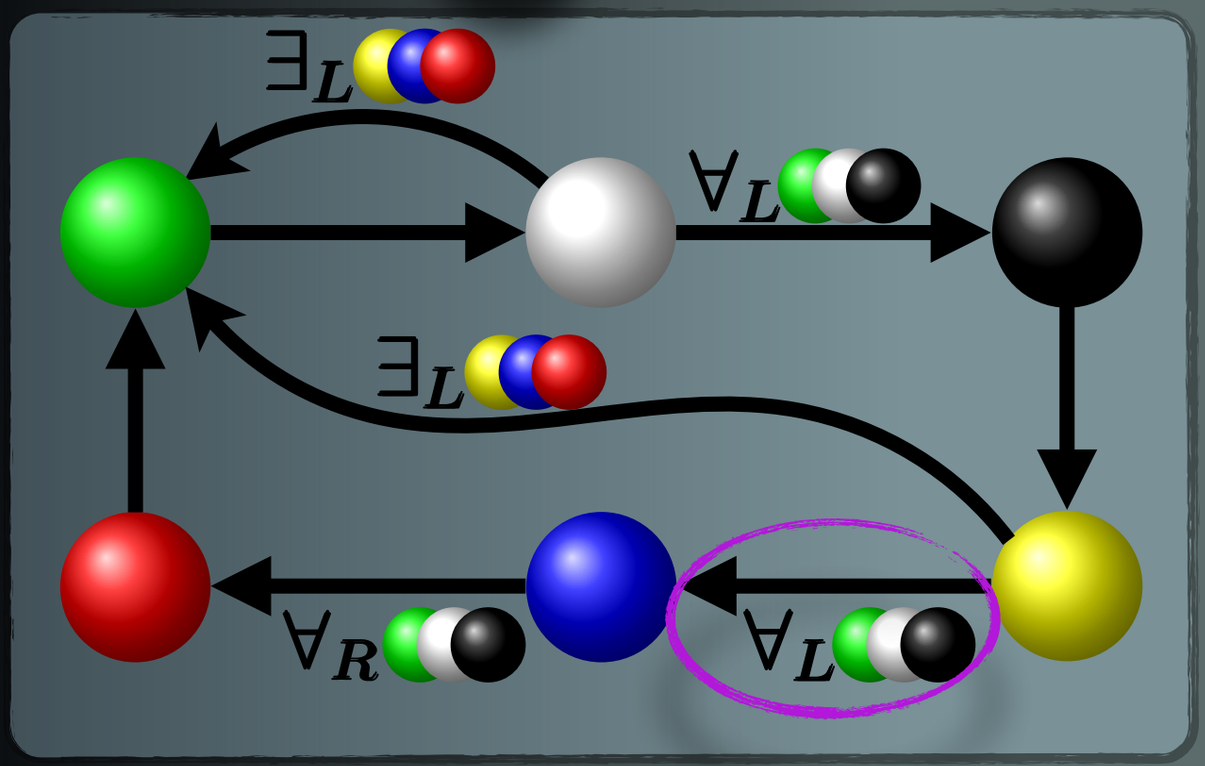
X



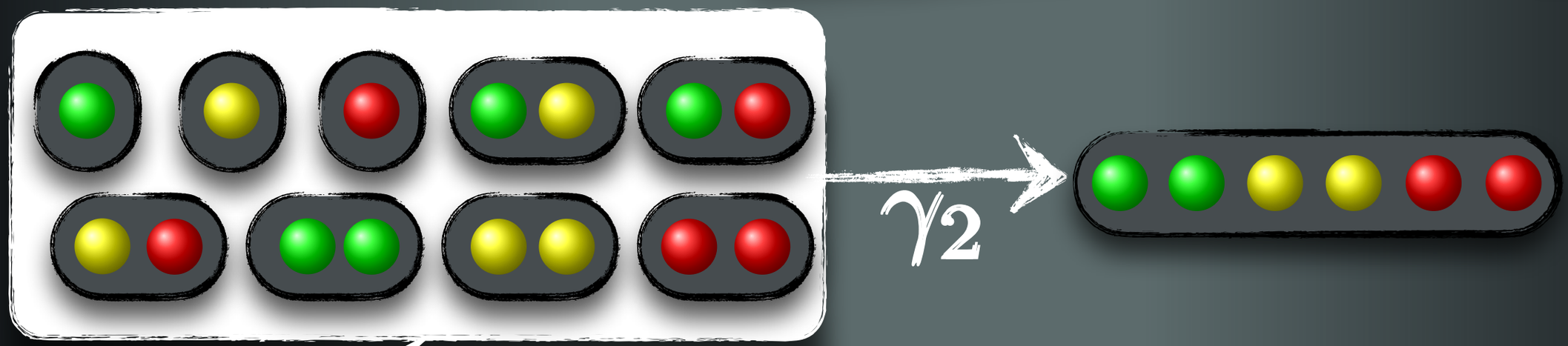
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



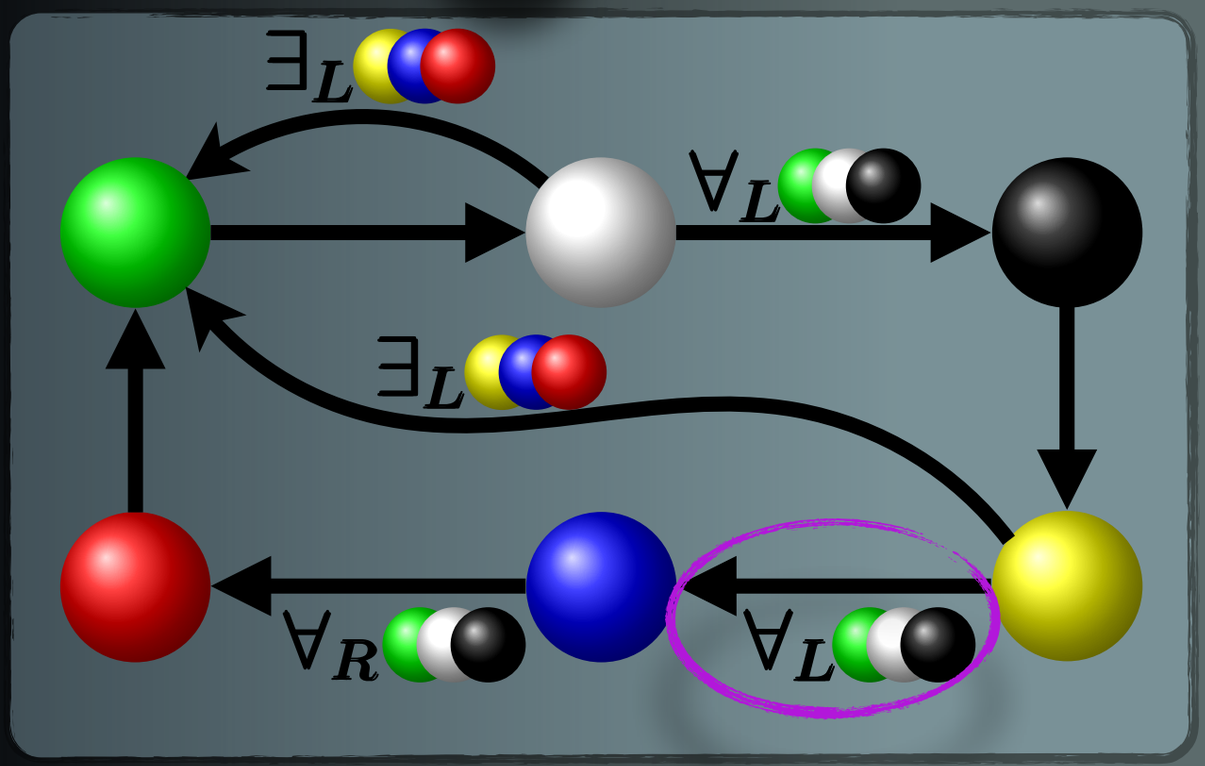
X



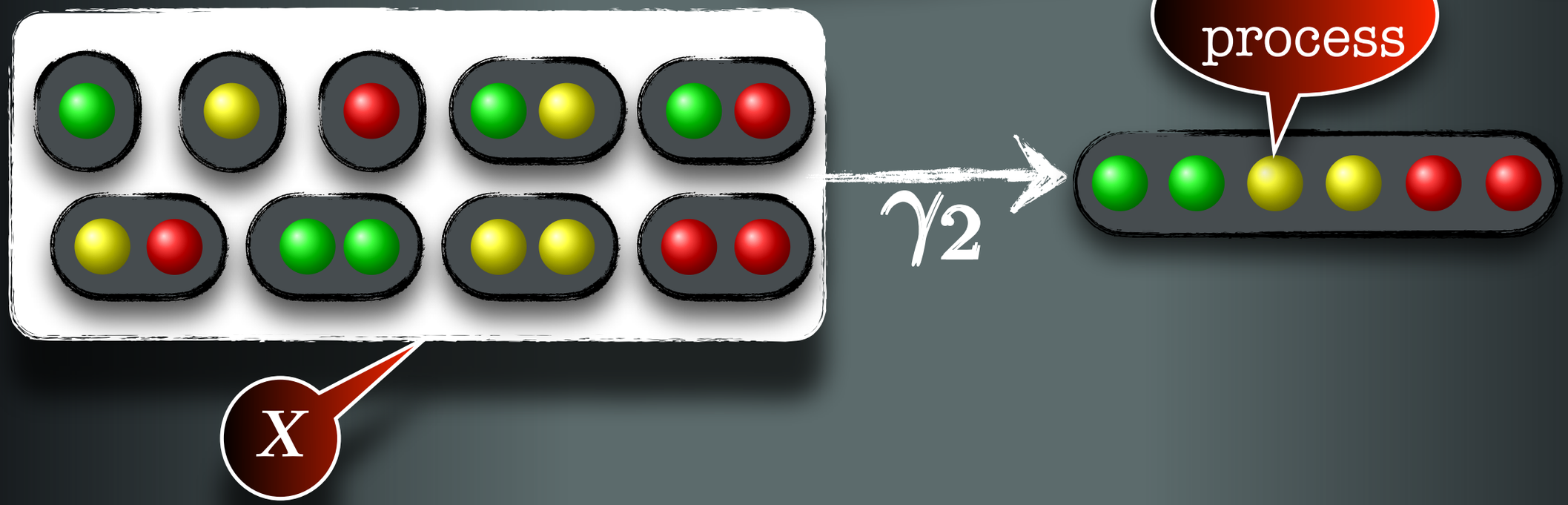
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$

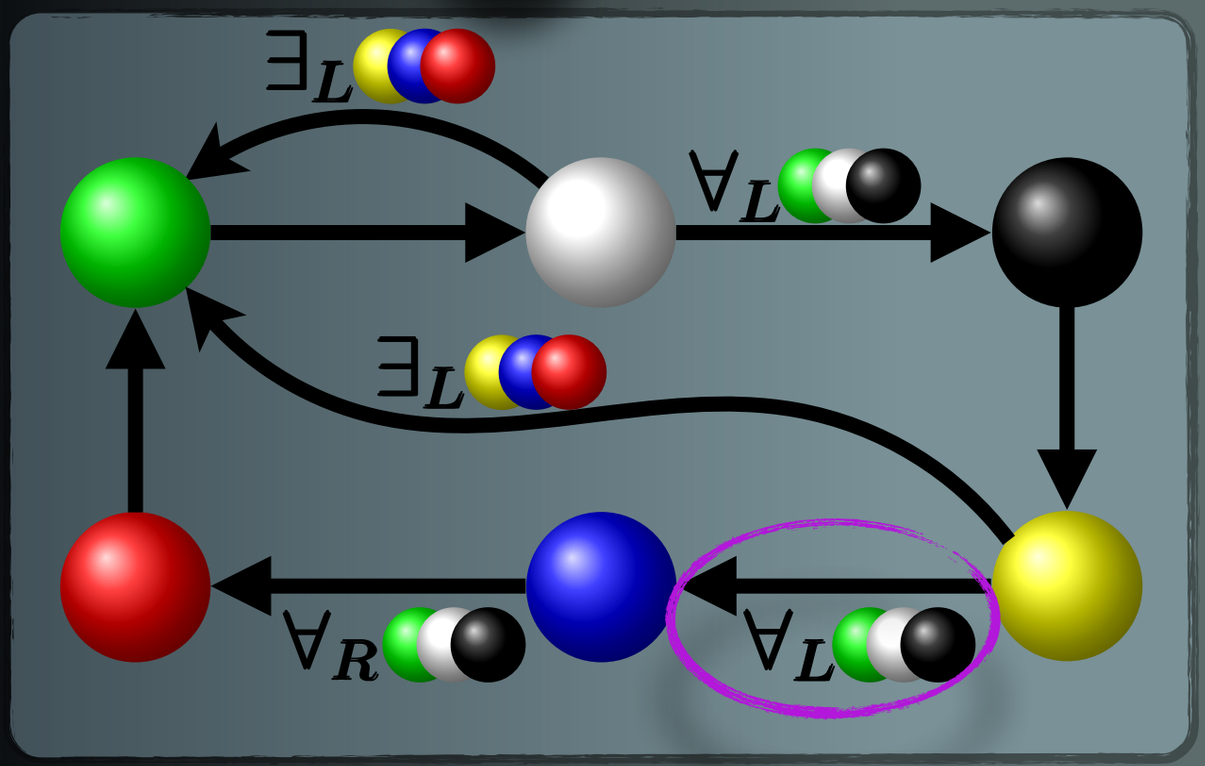


X

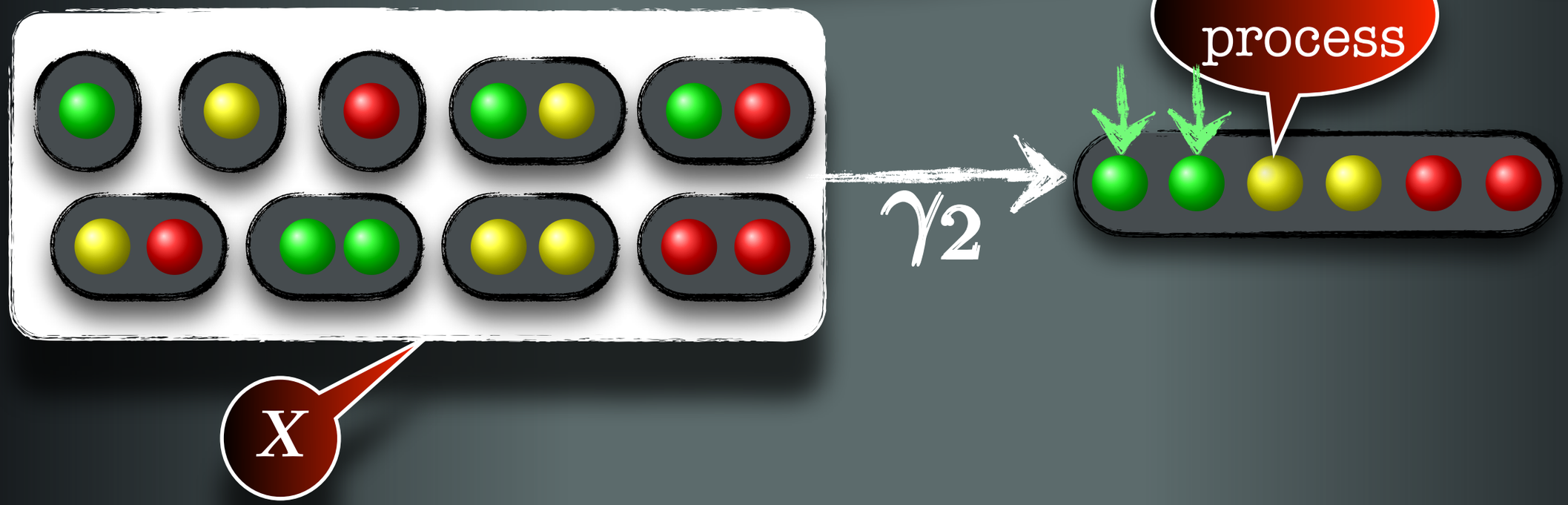


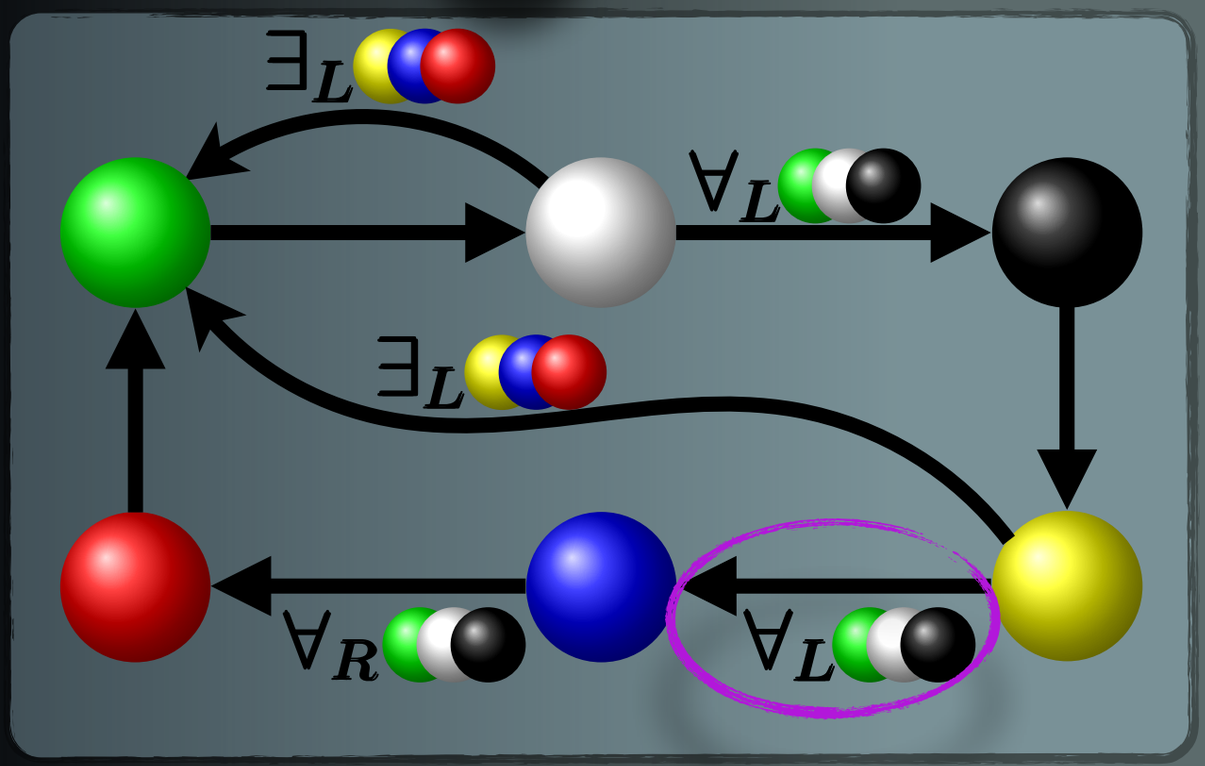
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



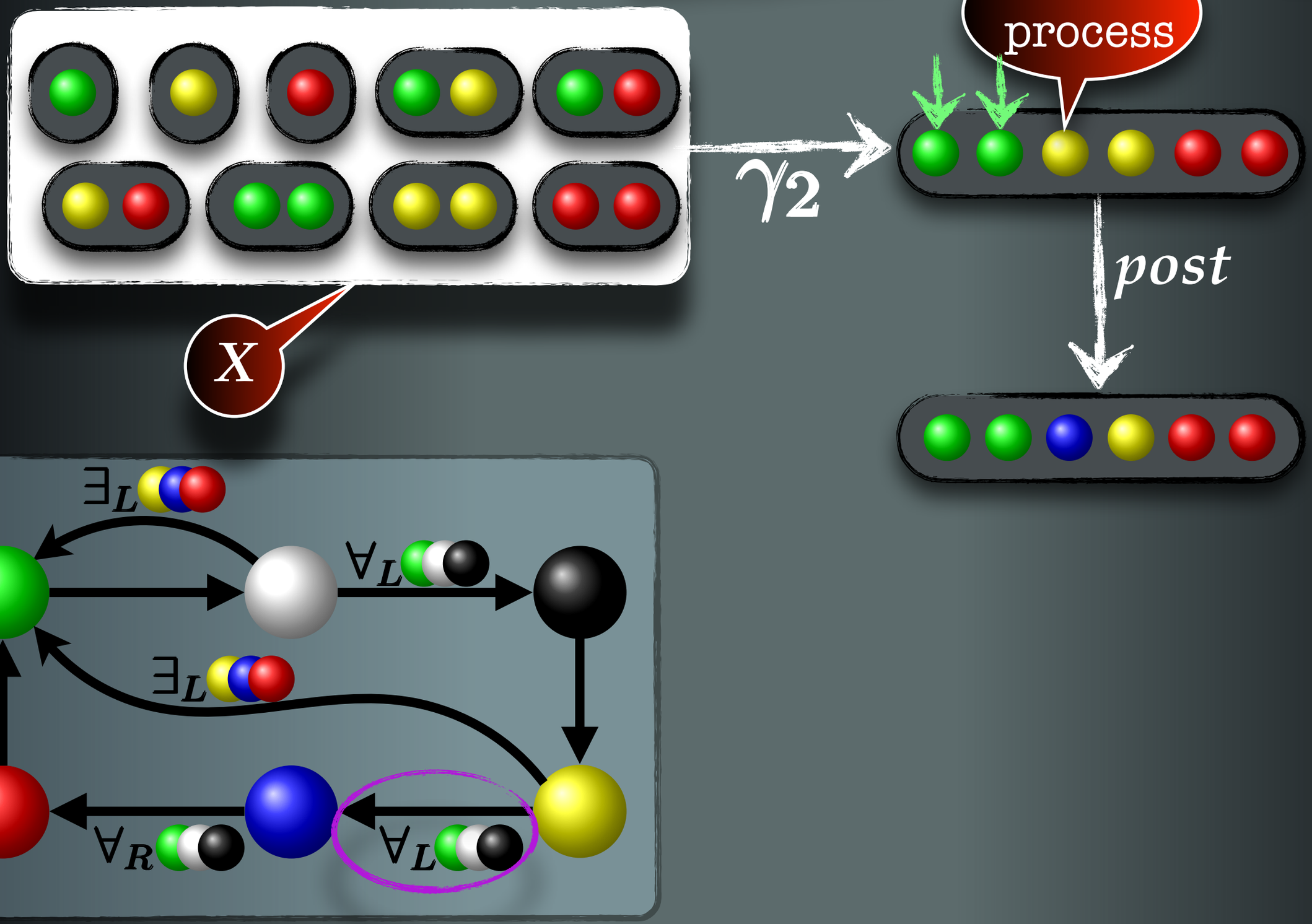


$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$

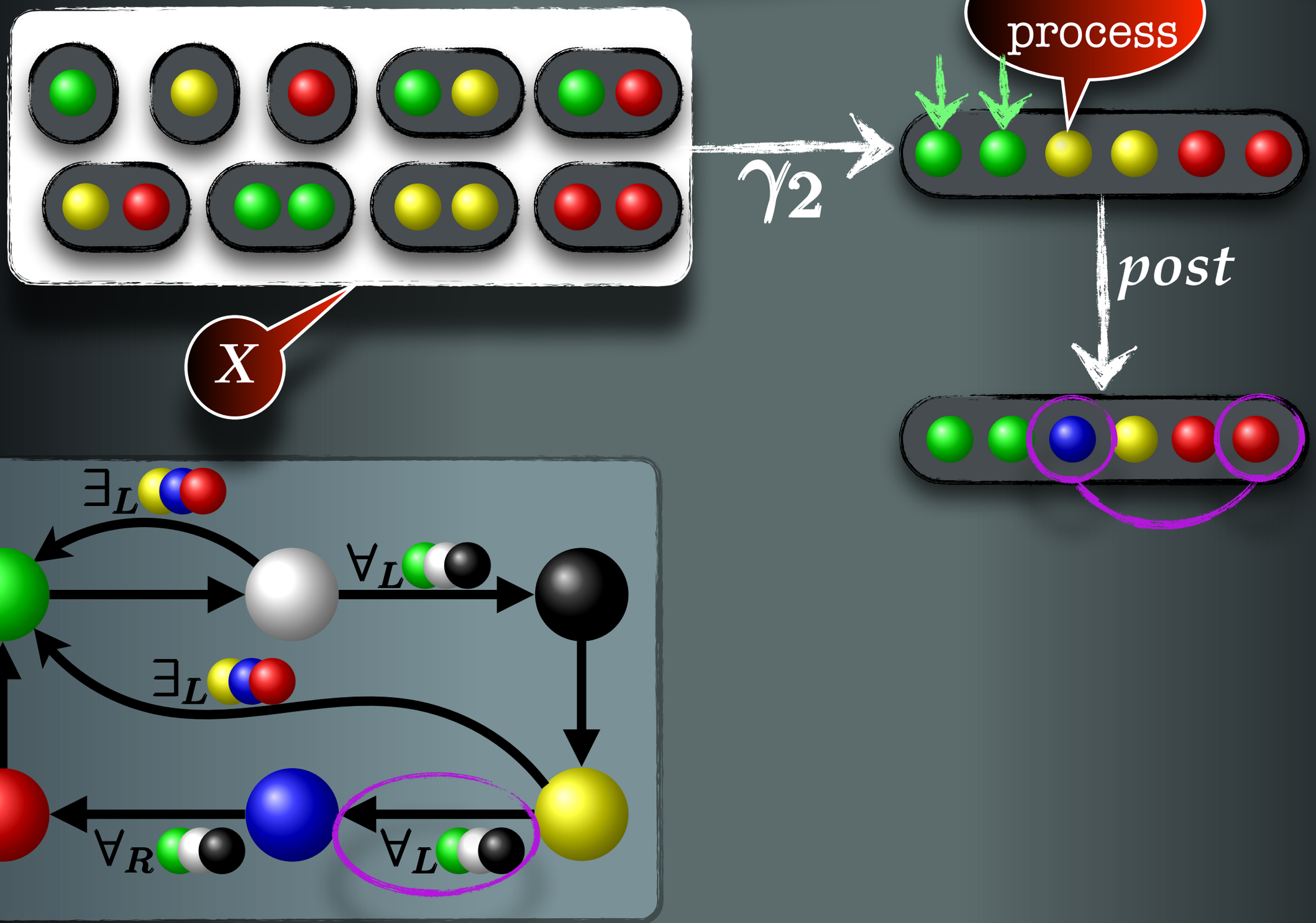




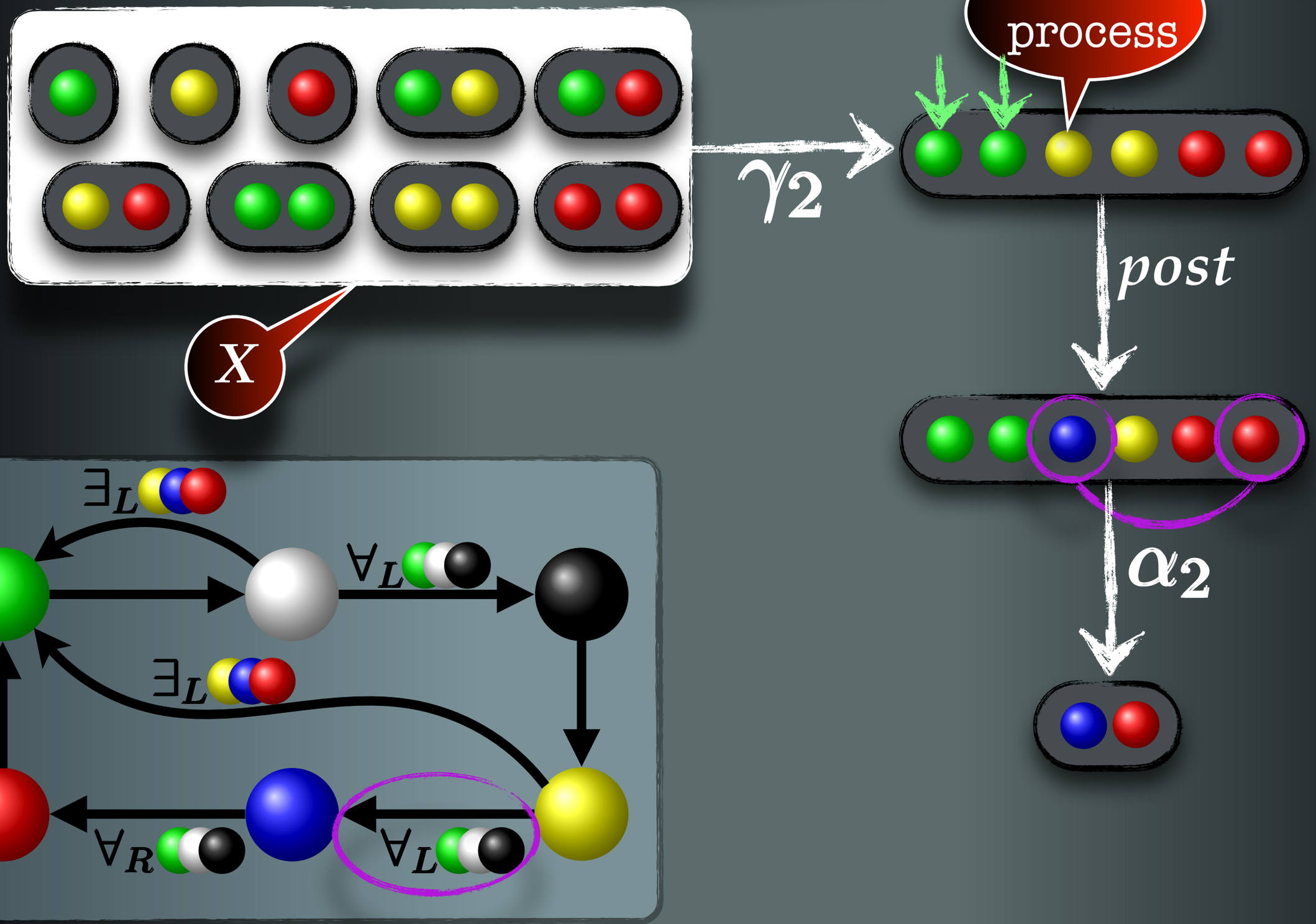
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



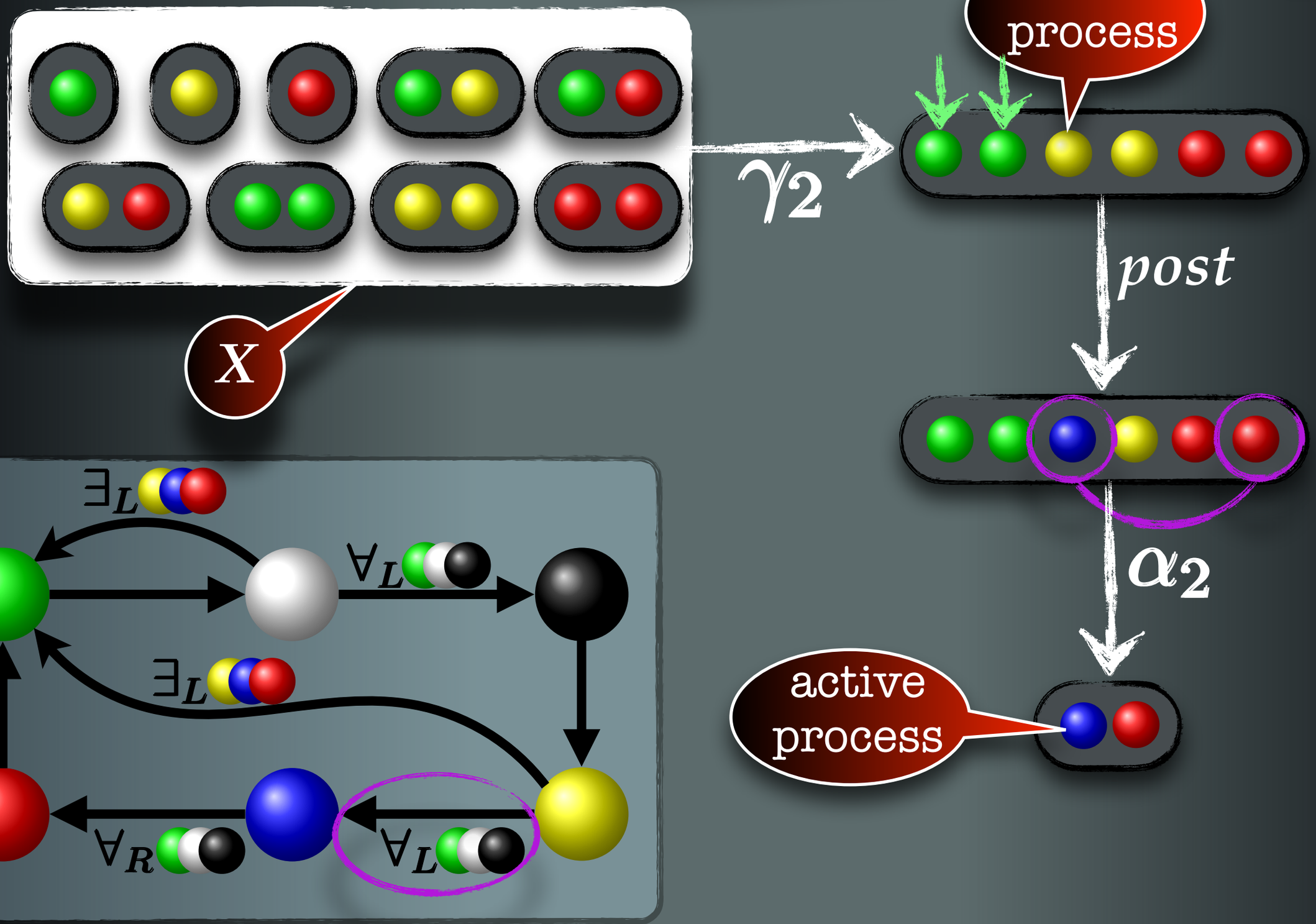
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



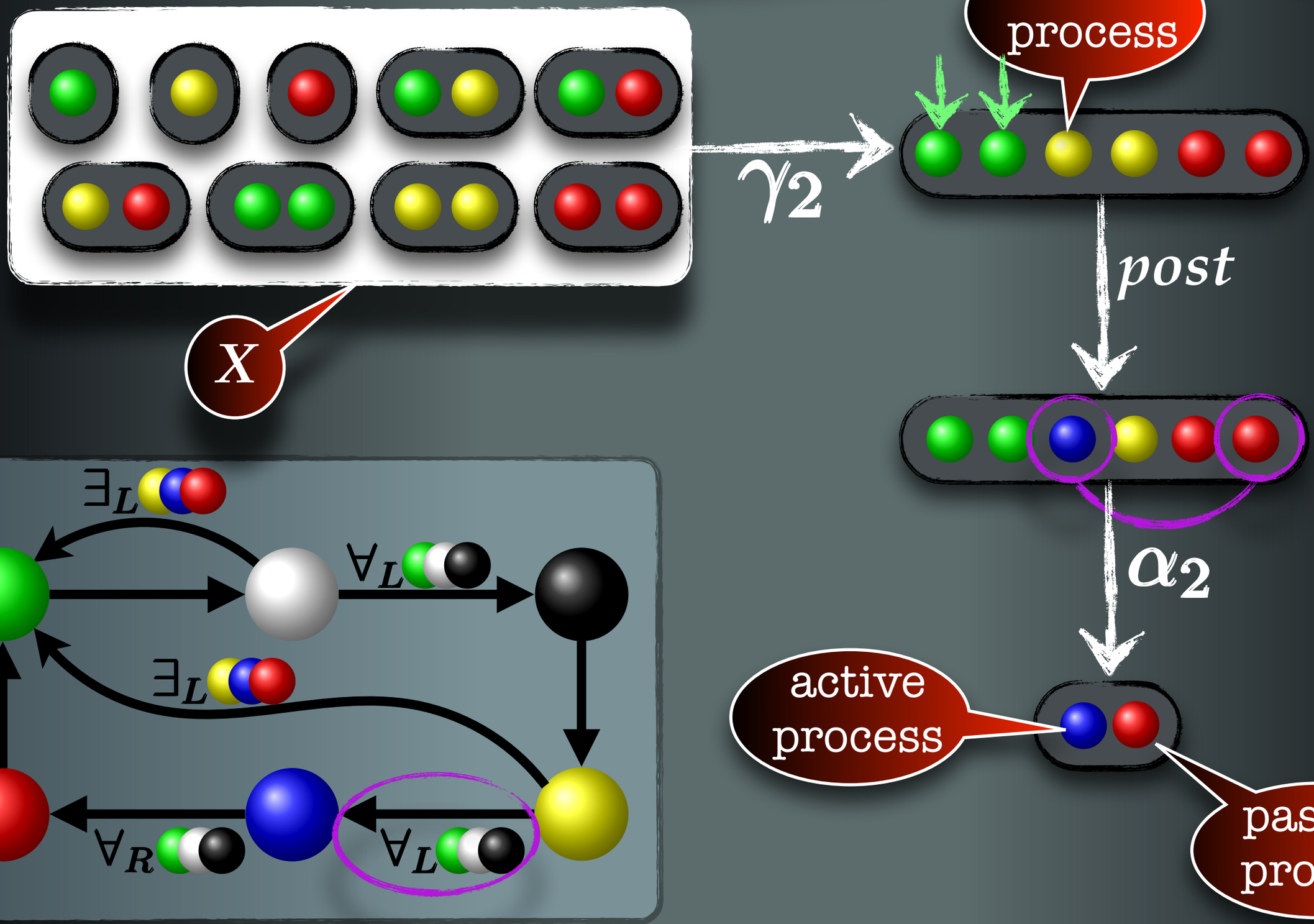
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



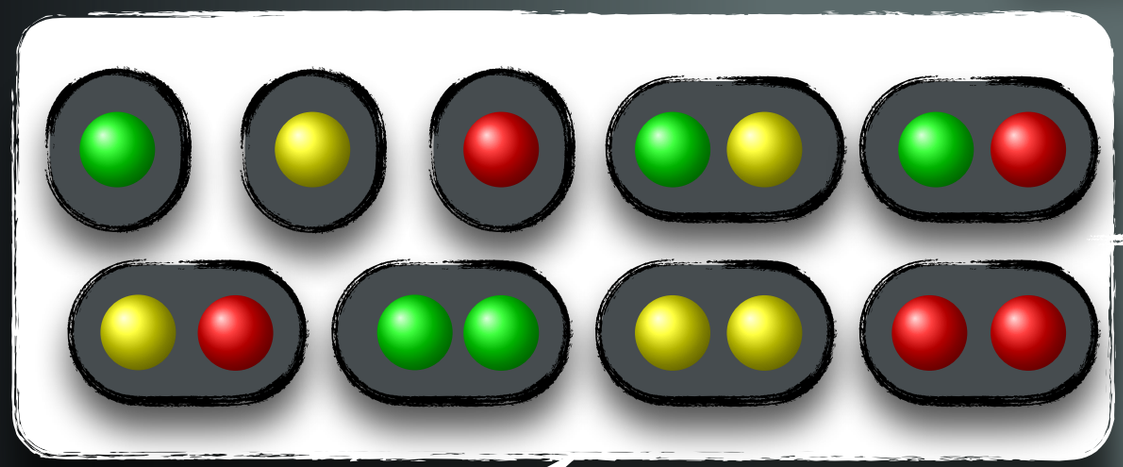
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$

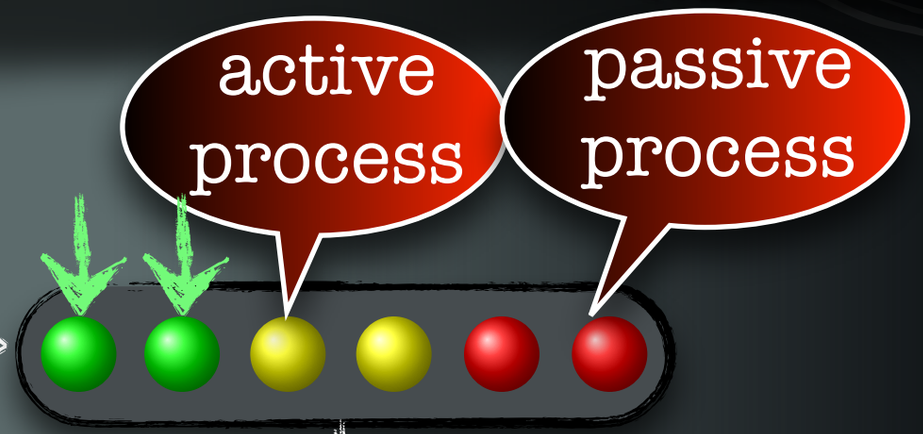


$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$

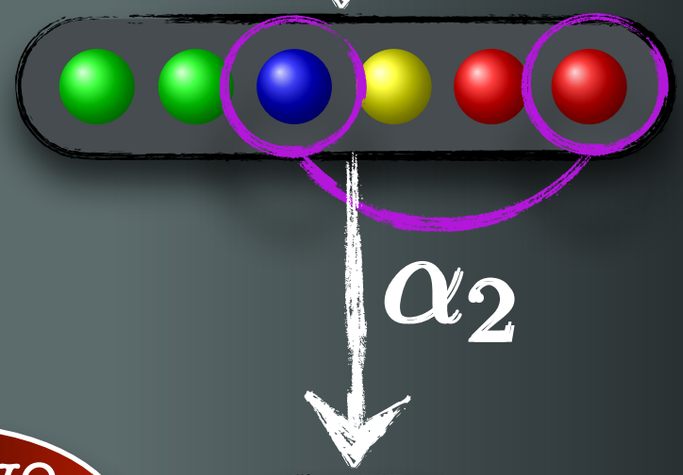


X

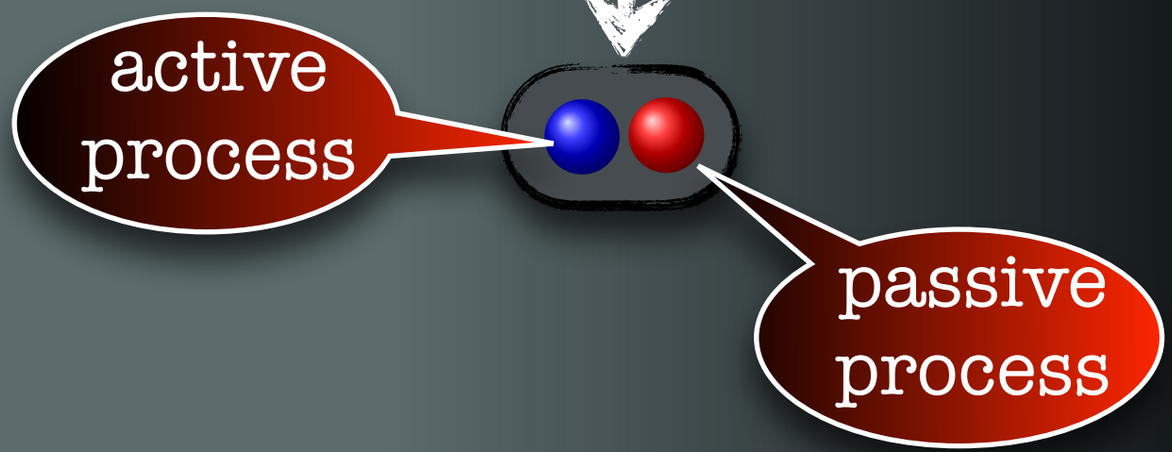
γ_2

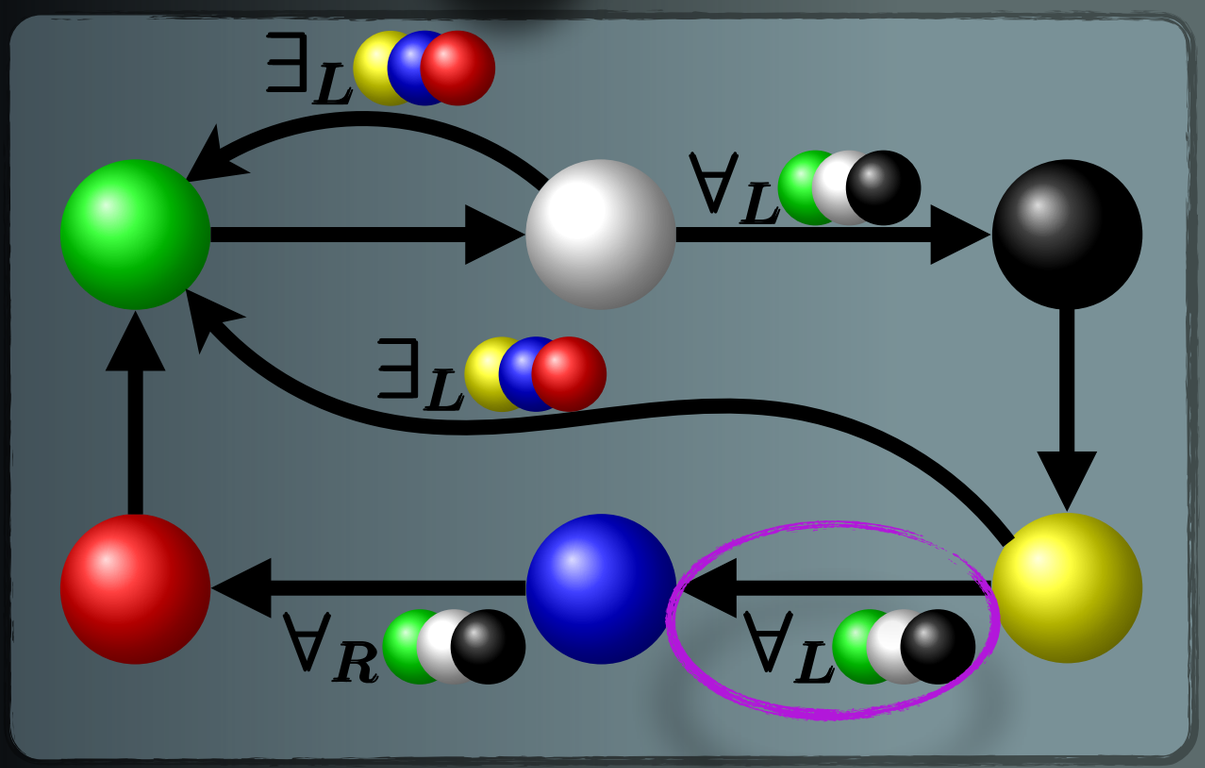


$post$

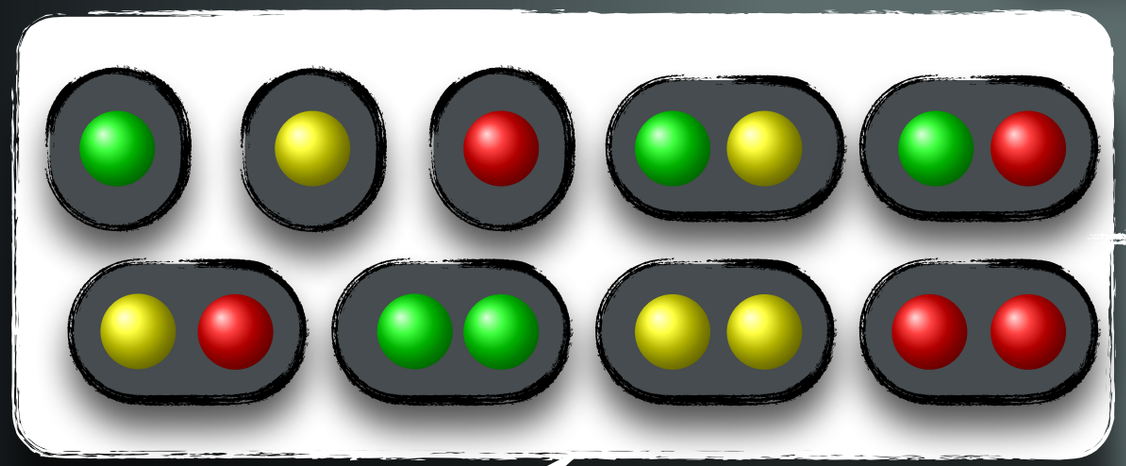


α_2





$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



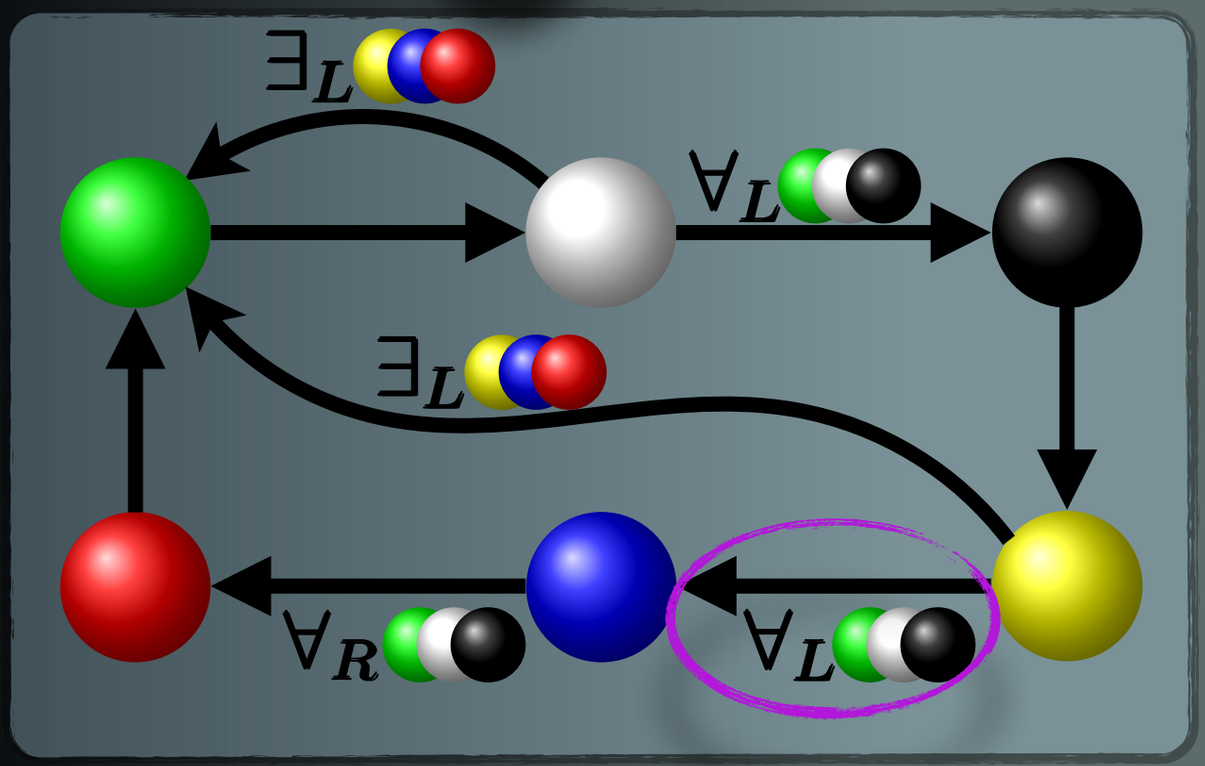
γ_2



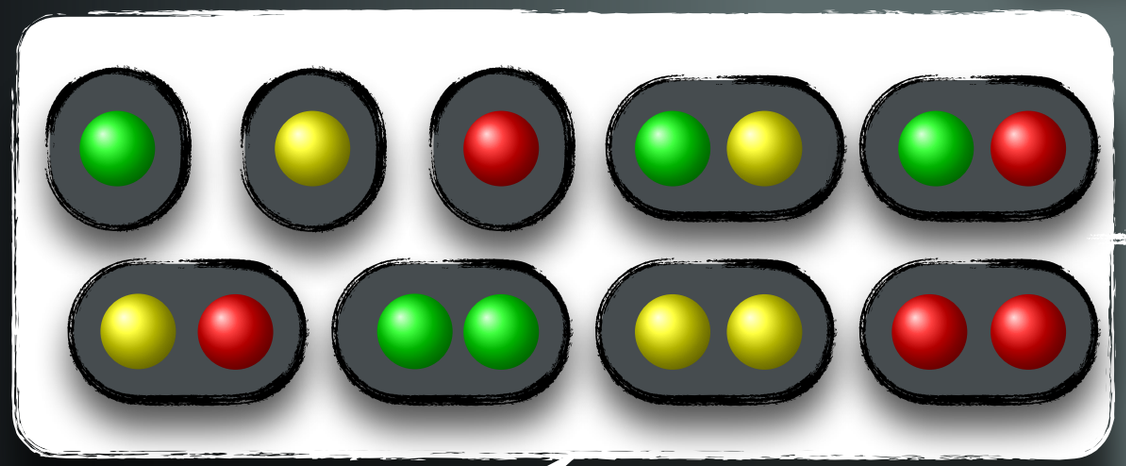
$post$

α_2

X

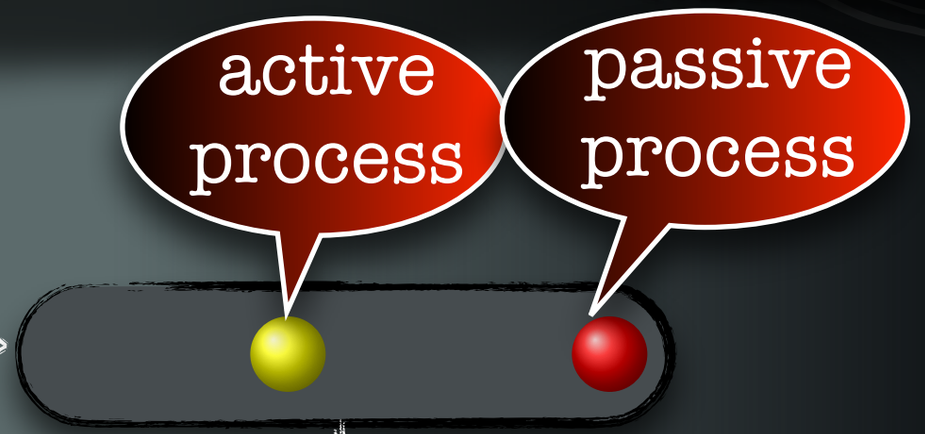


$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$

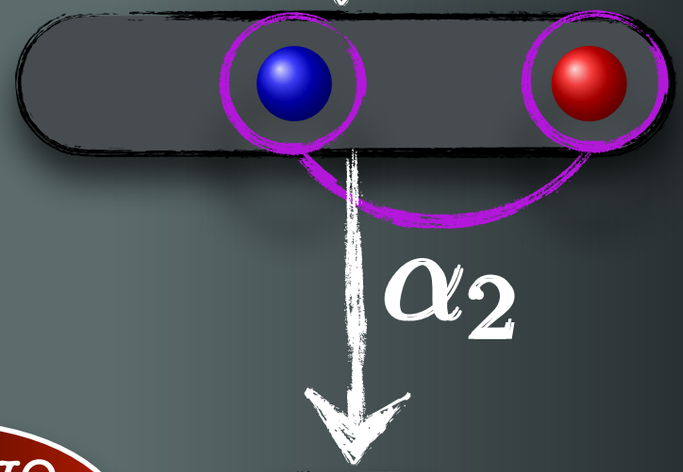


X

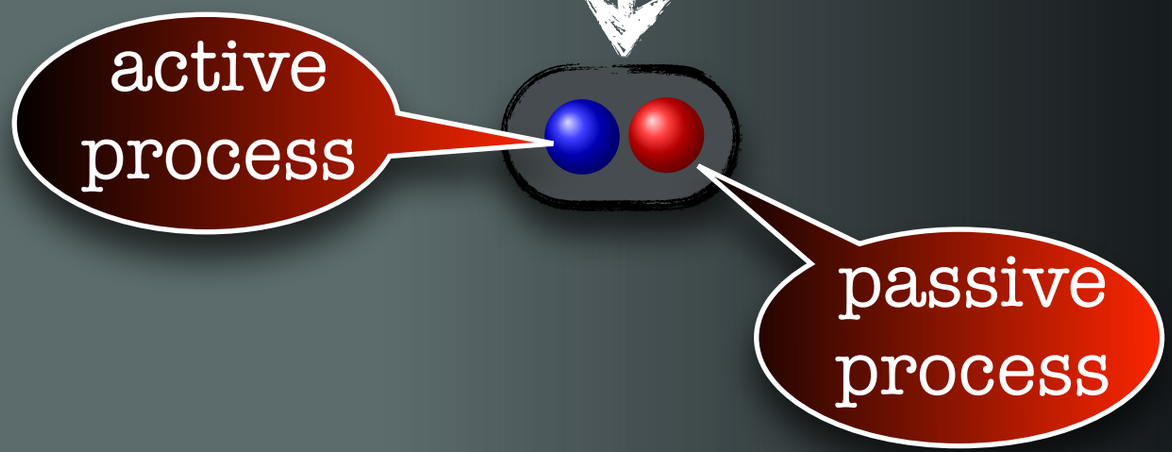
$\oint_2^2$

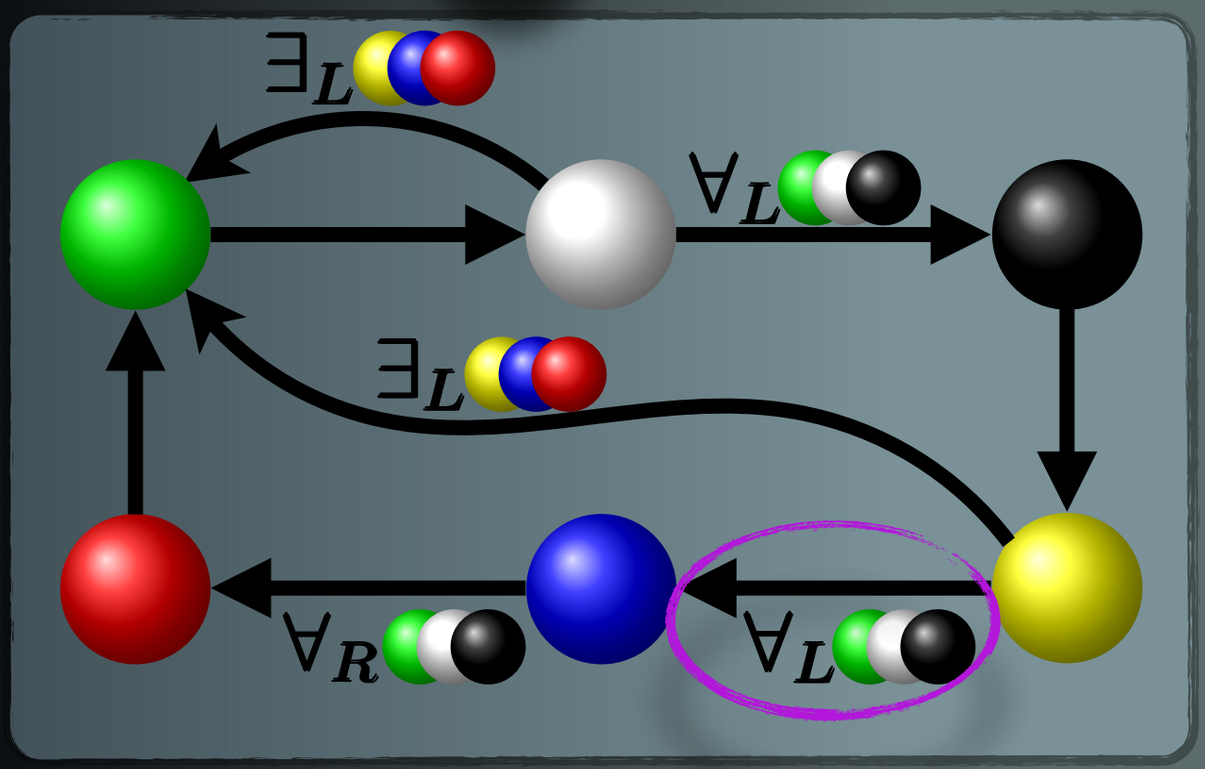


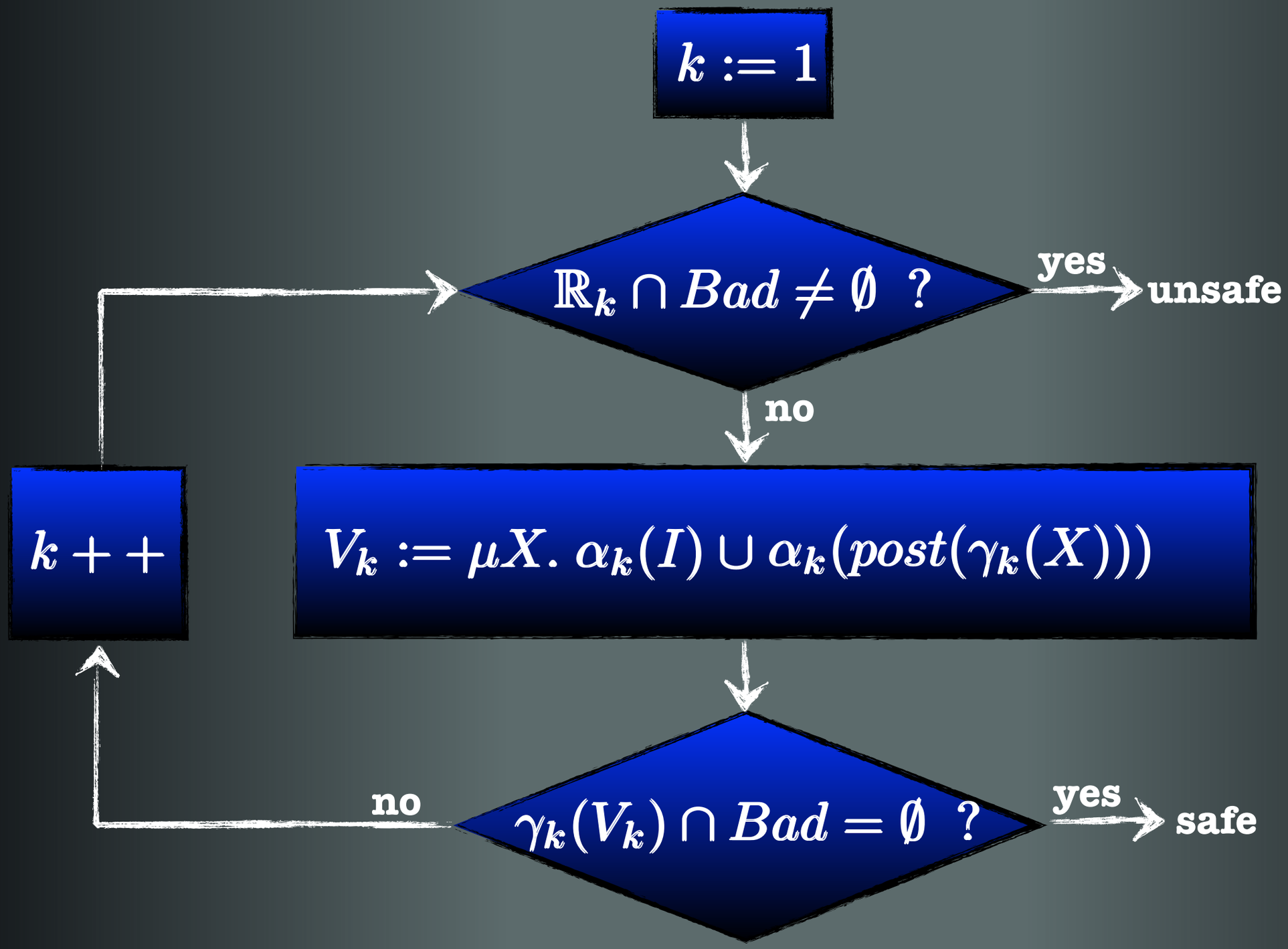
$post$



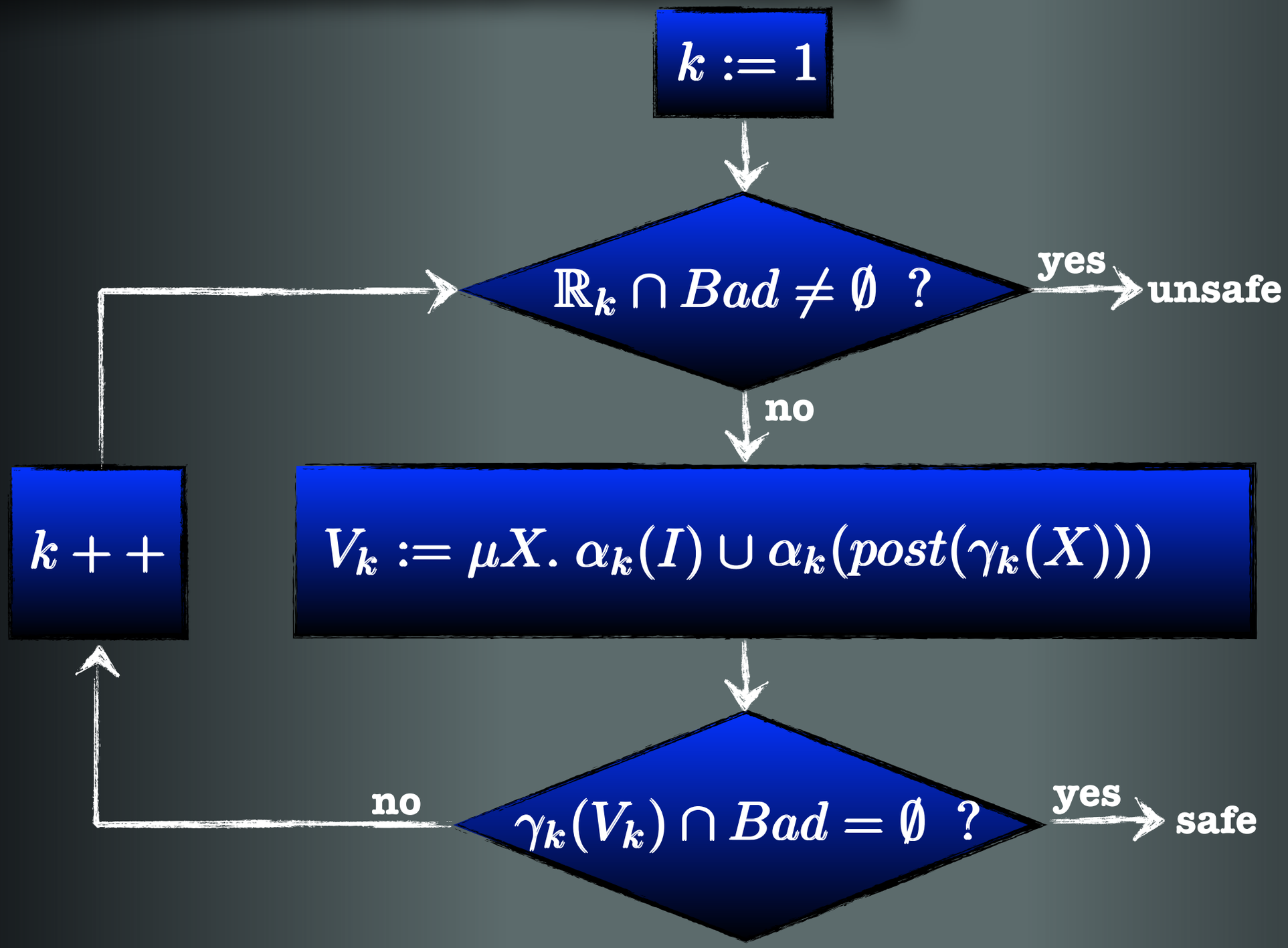
α_2



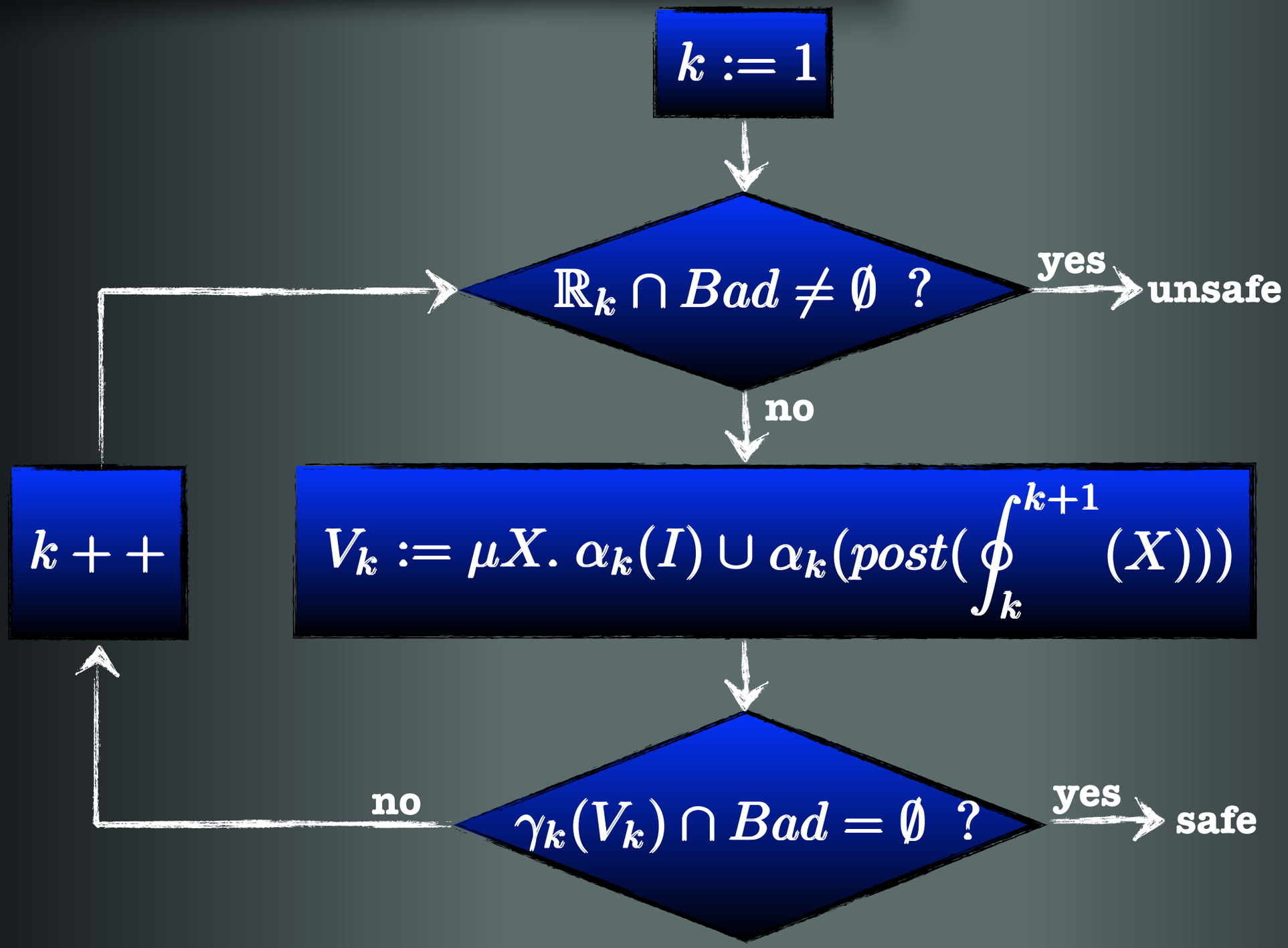



$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$k++$
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\gamma_k(X)))$
no
$\gamma_k(V_k) \cap Bad = \emptyset$?
yes
safe

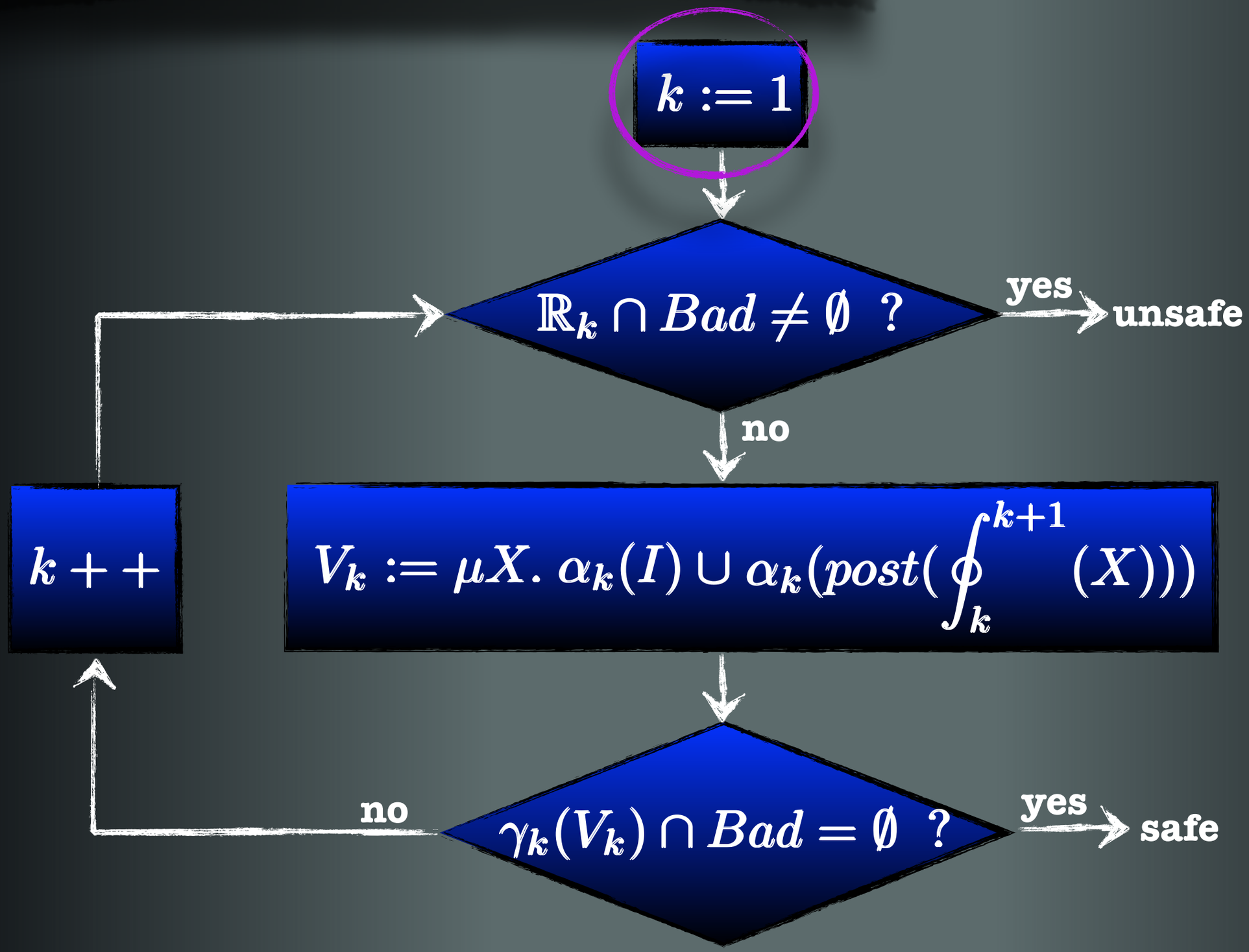
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$

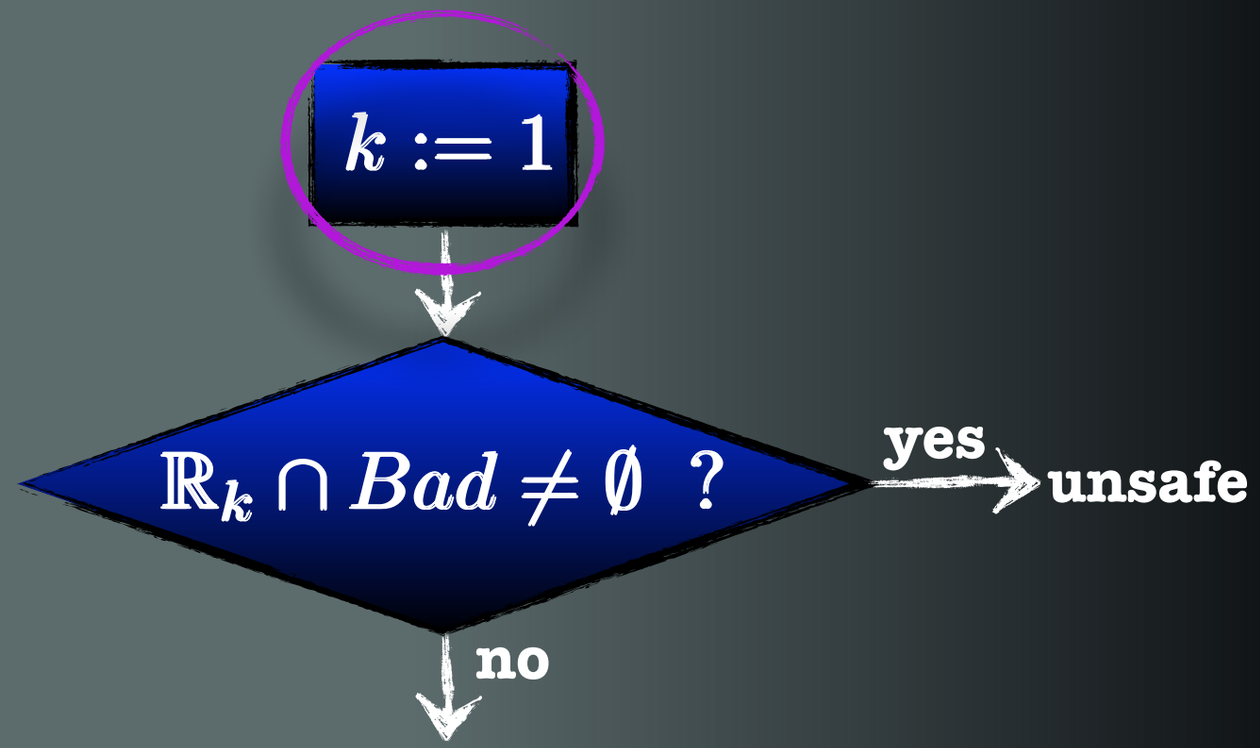


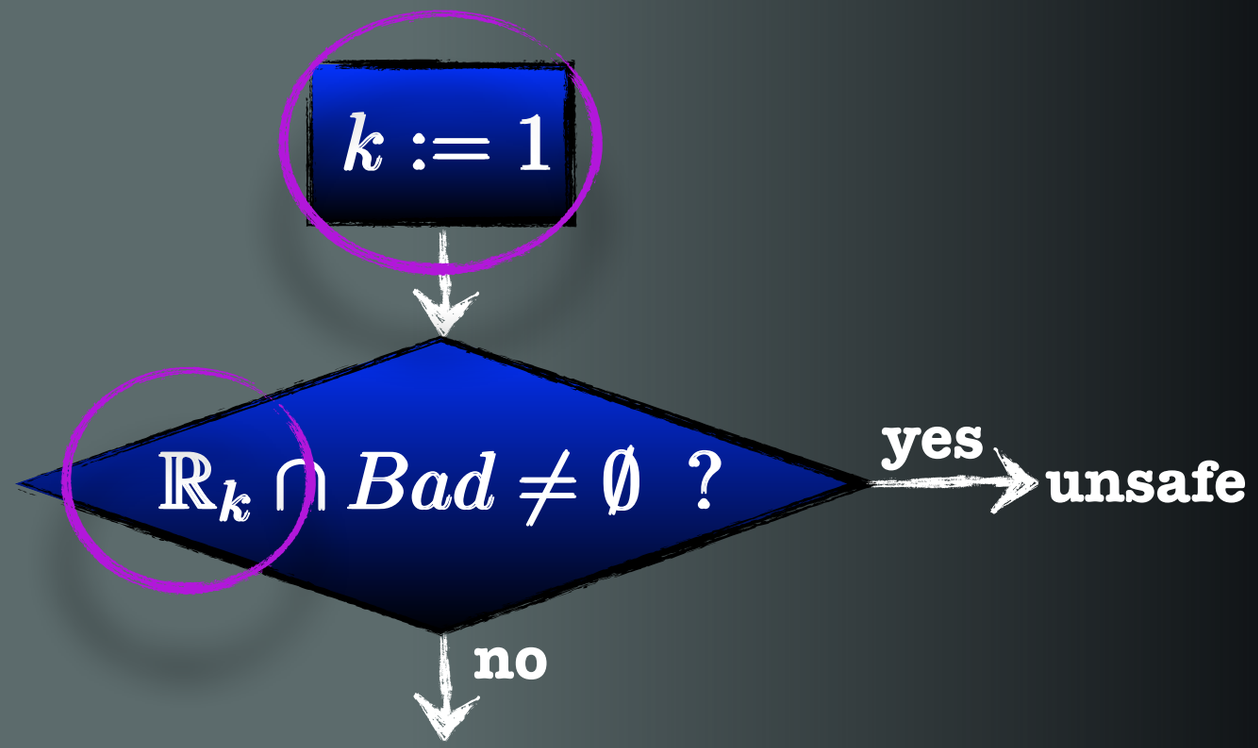
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



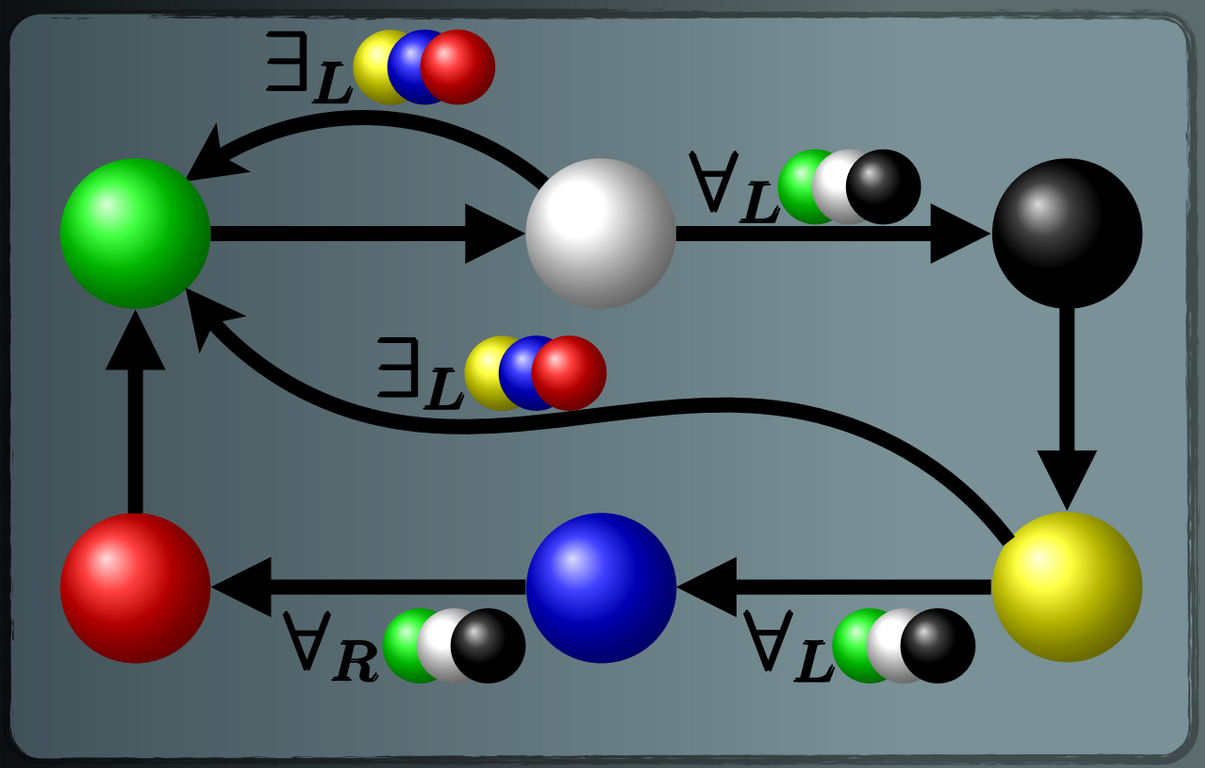
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$

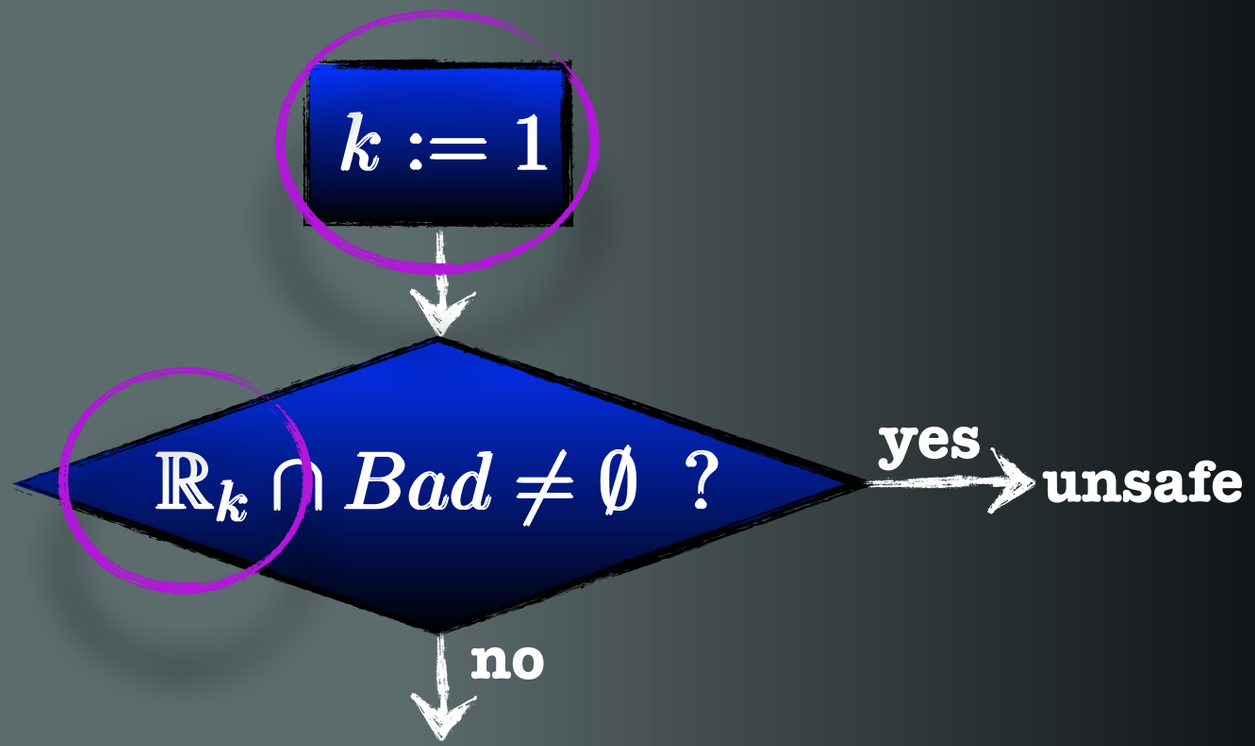



$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

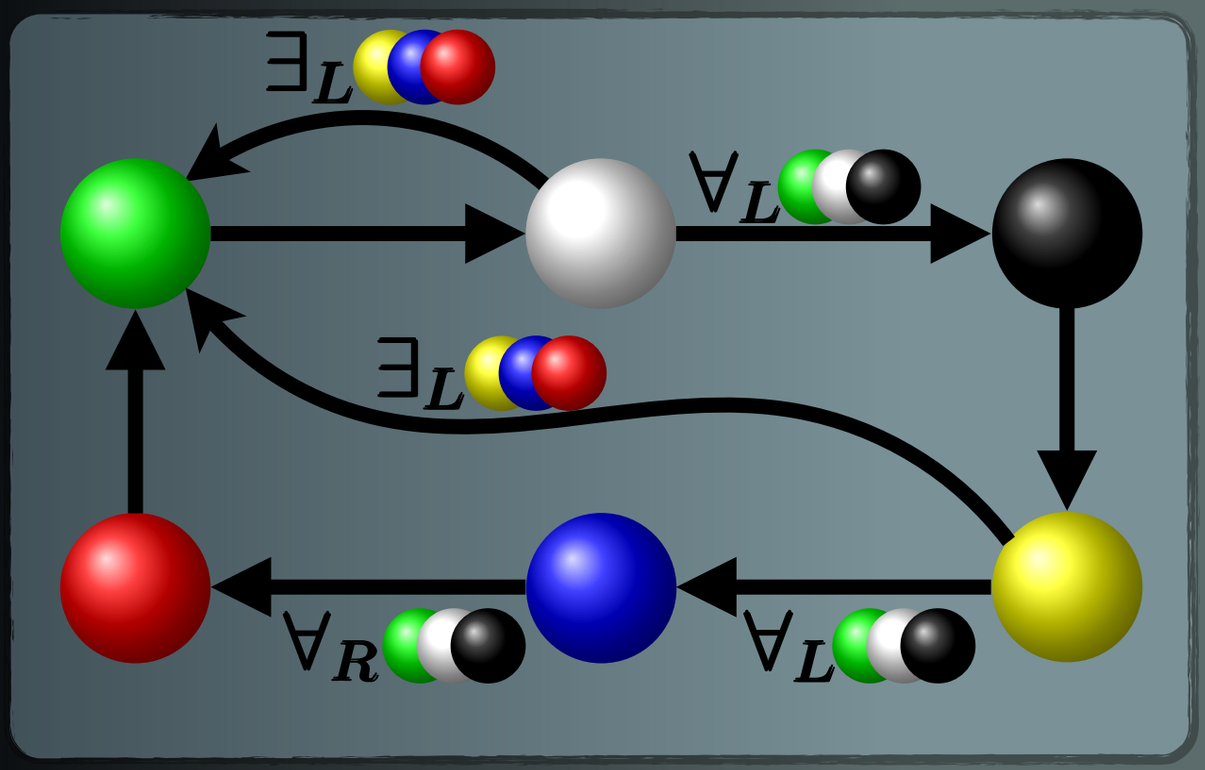

$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

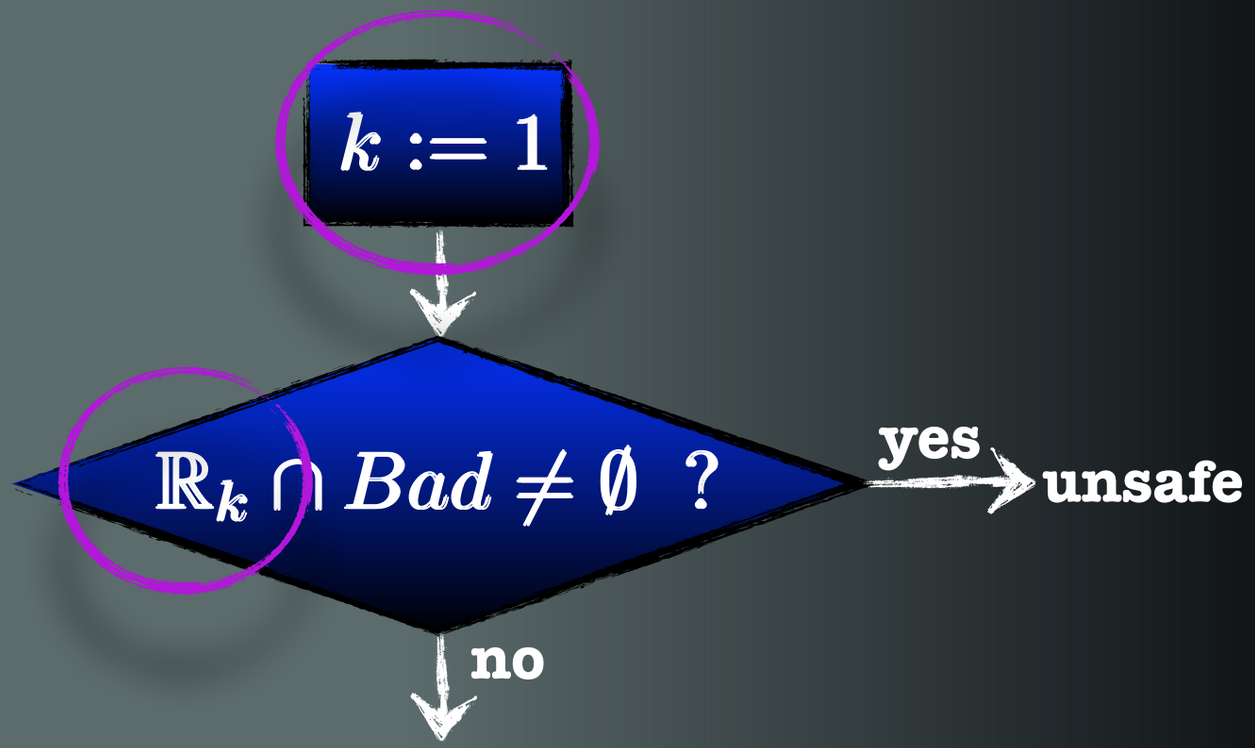
$\mathbb{R}_1$


$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$

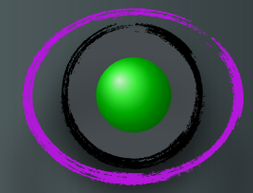

$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

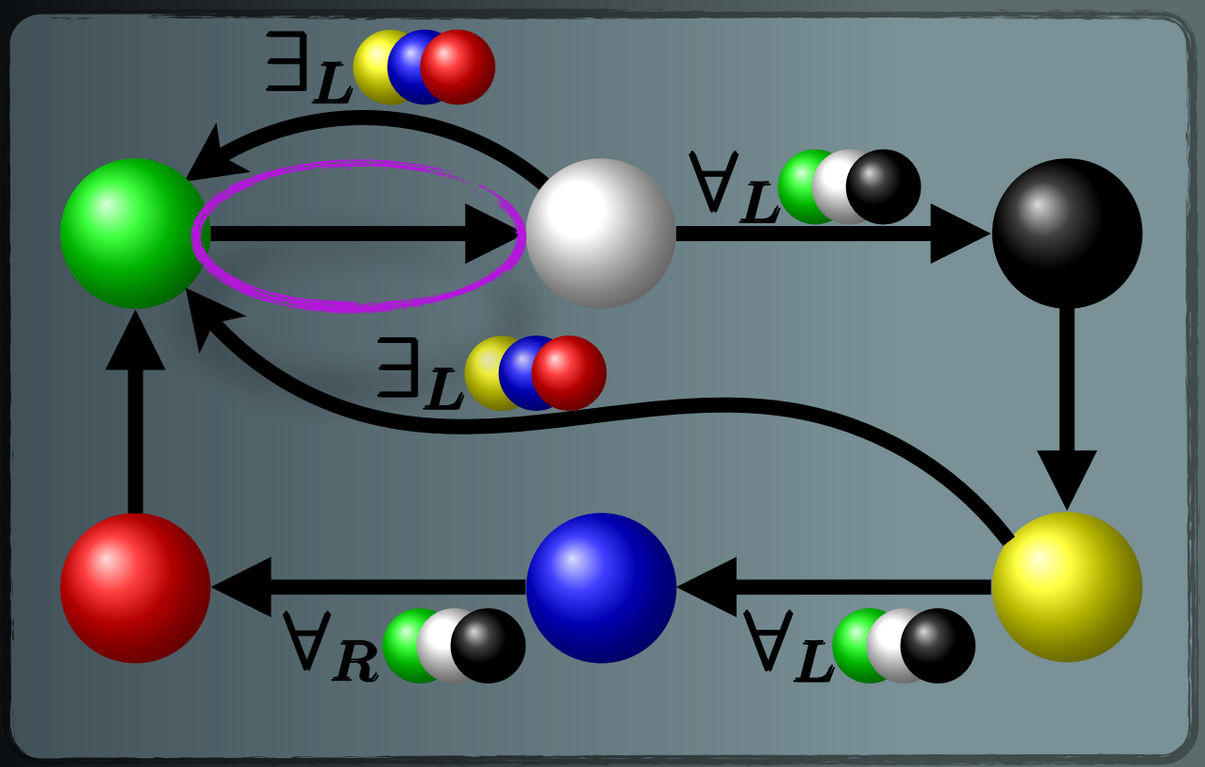
$\mathbb{R}_1$ $Init_1$

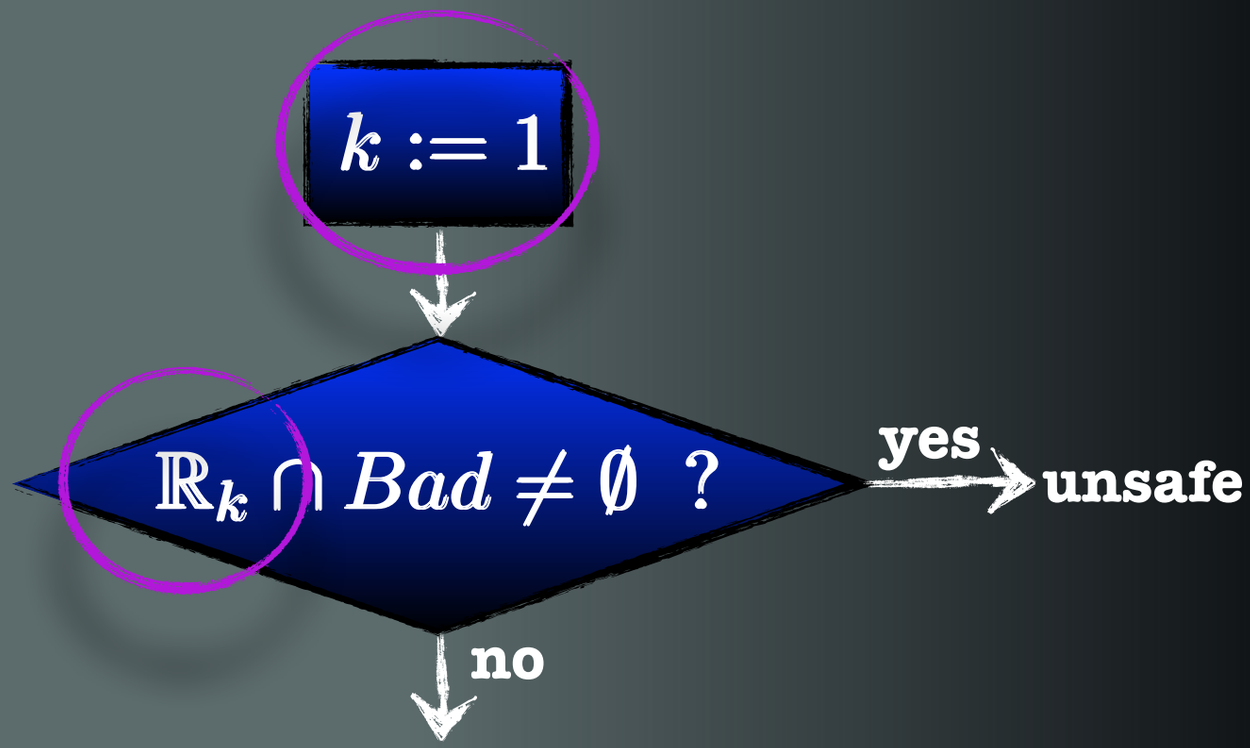




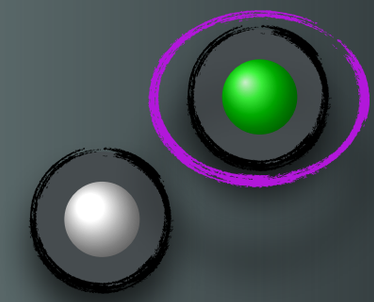
$\mathbb{R}_1$

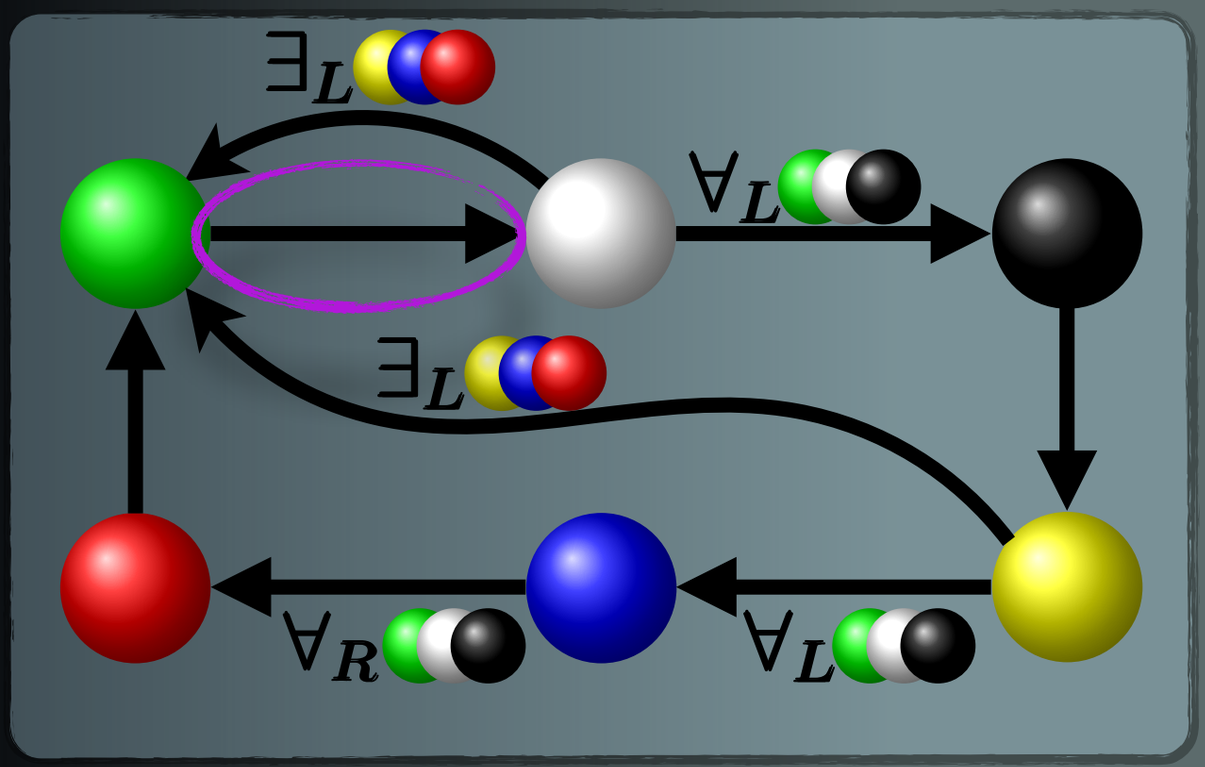


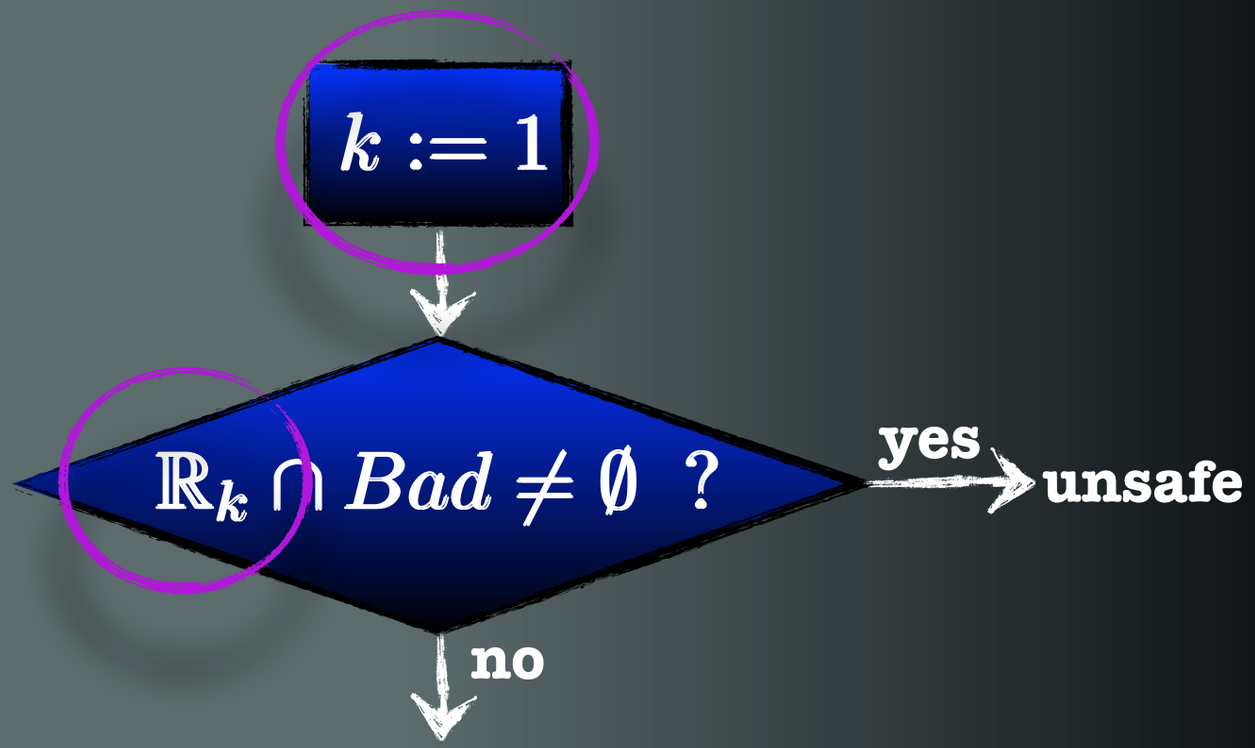

$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

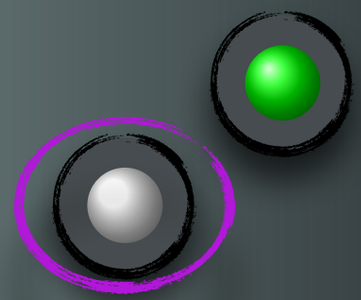
$\mathbb{R}_1$

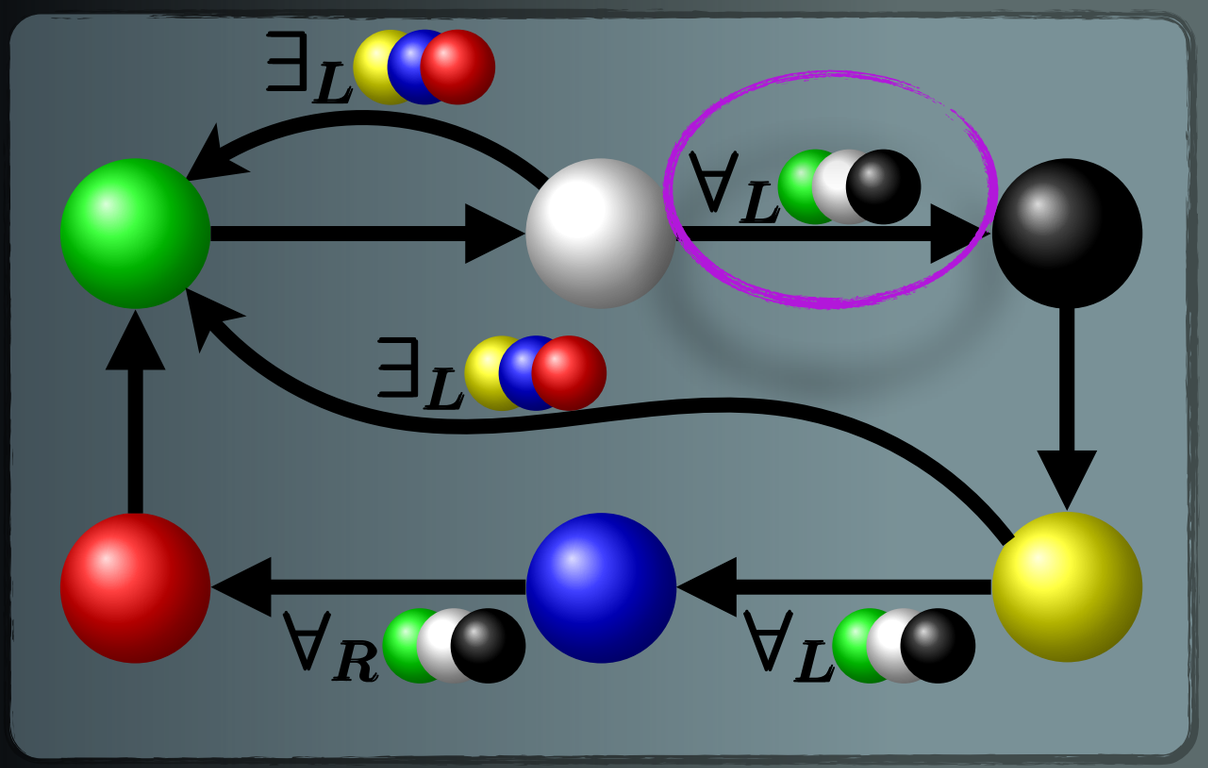


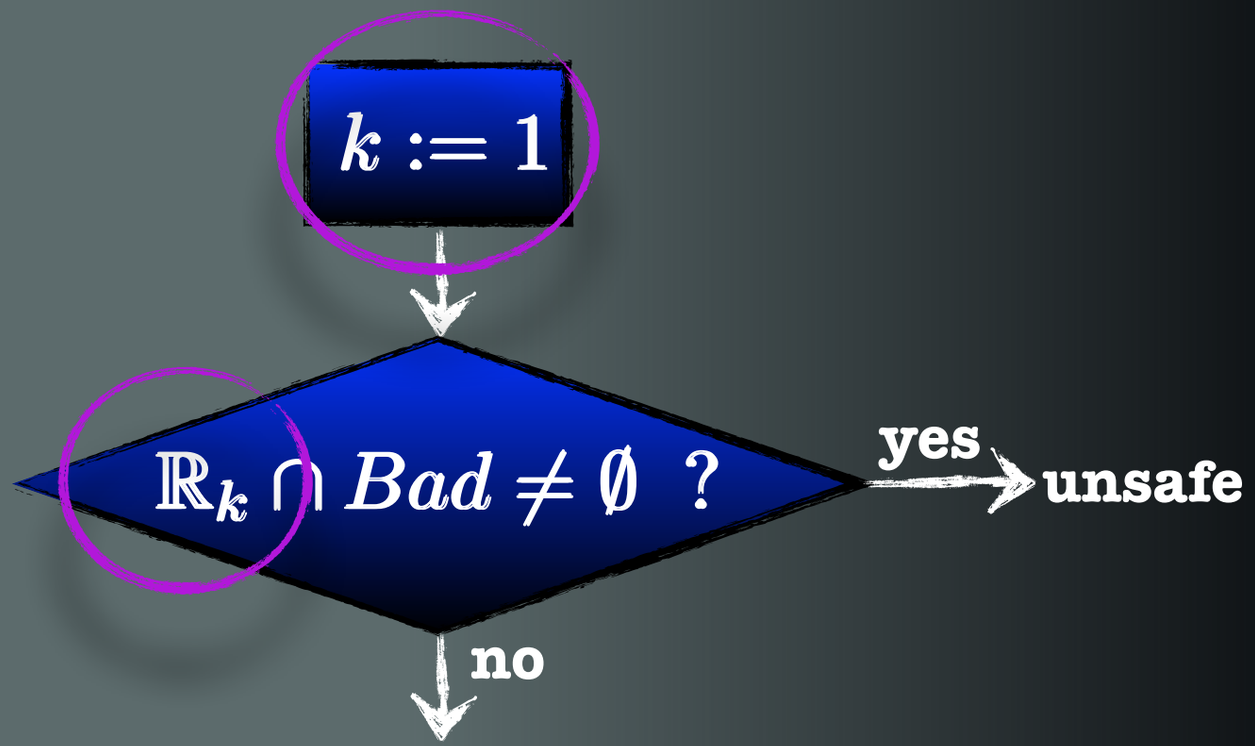

$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

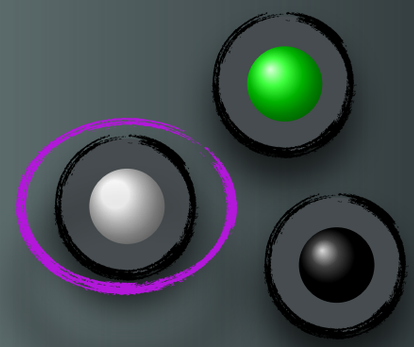
$\mathbb{R}_1$

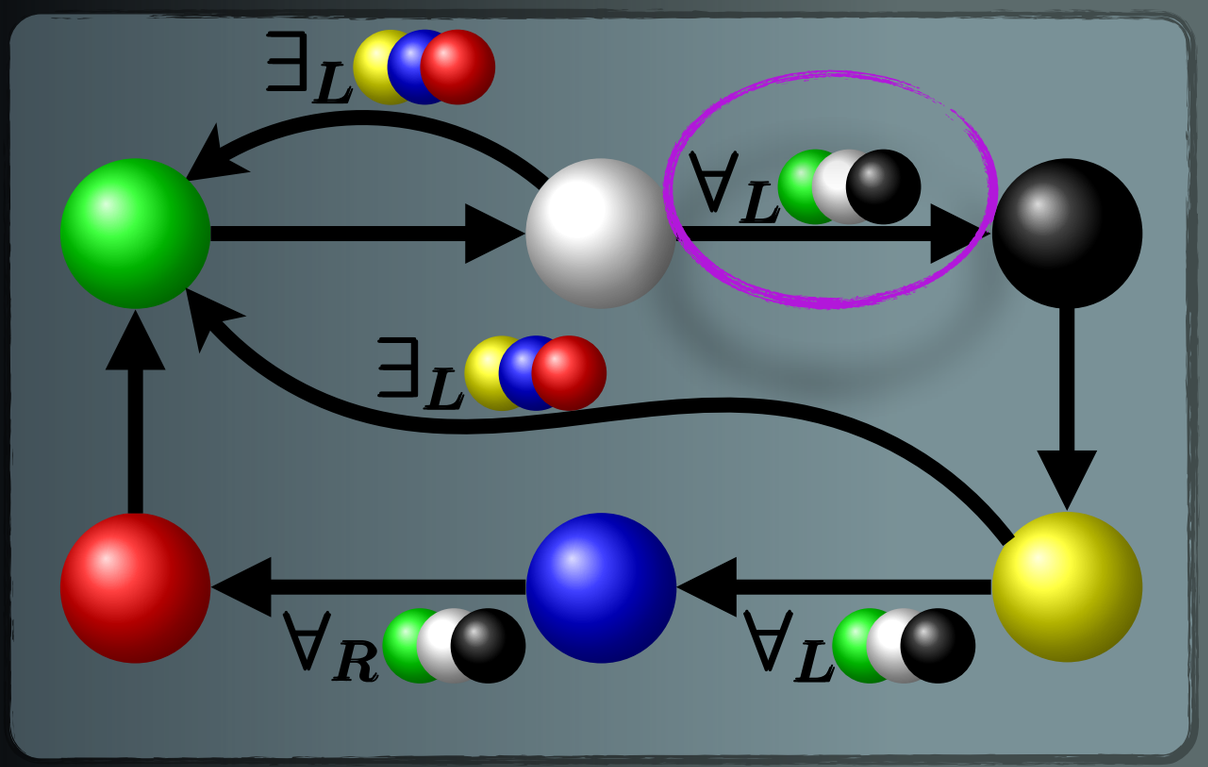


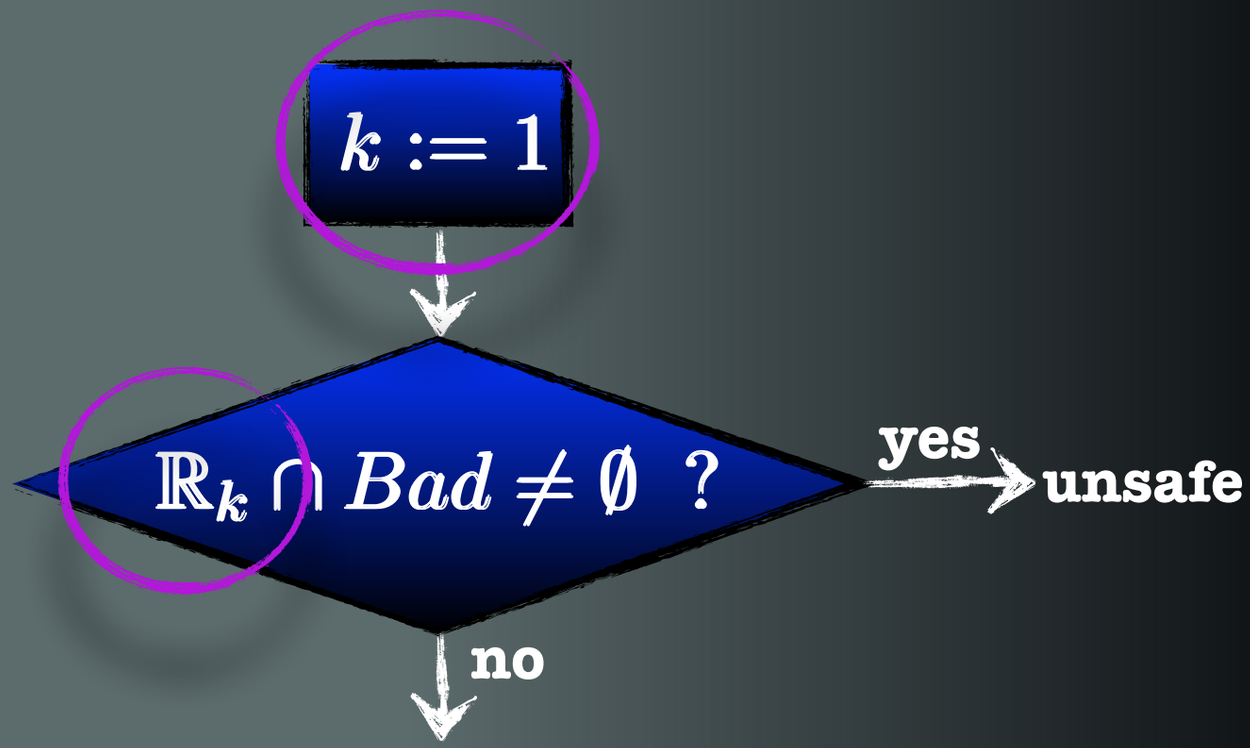

$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

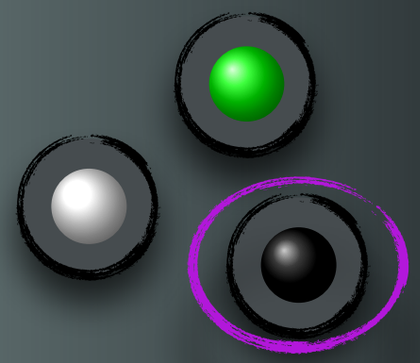
$\mathbb{R}_1$

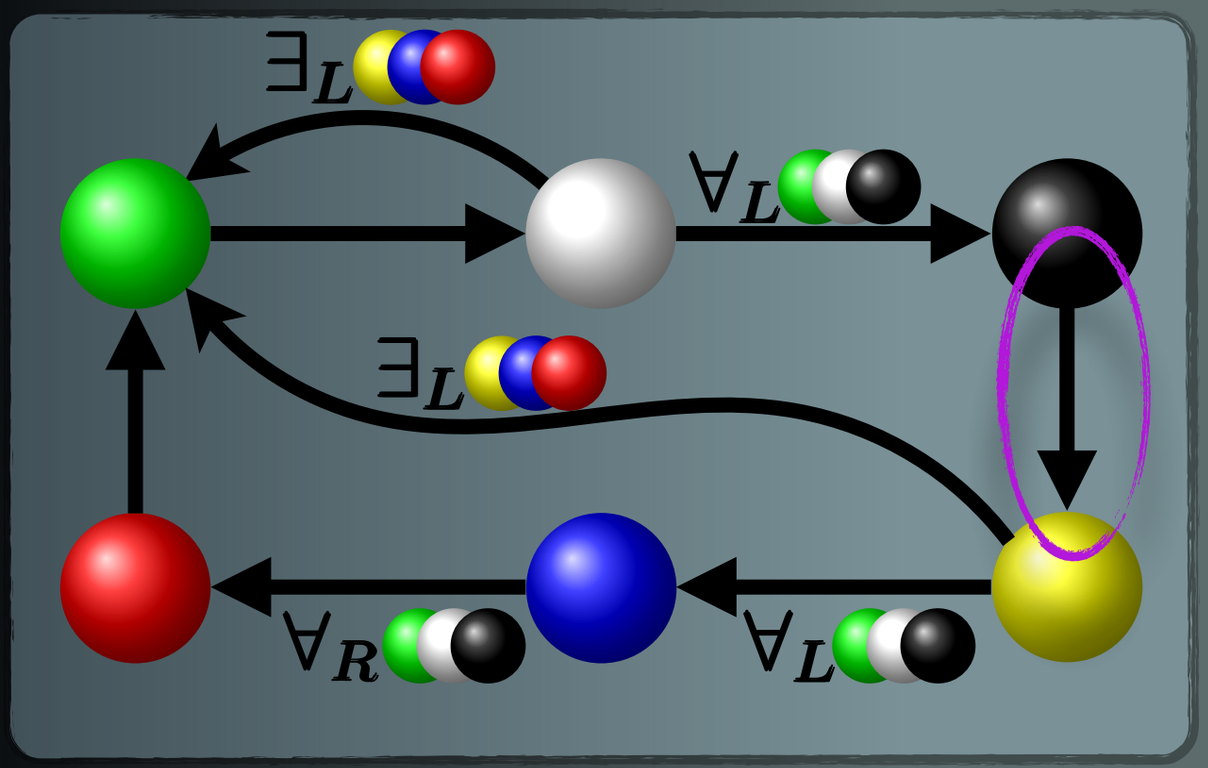


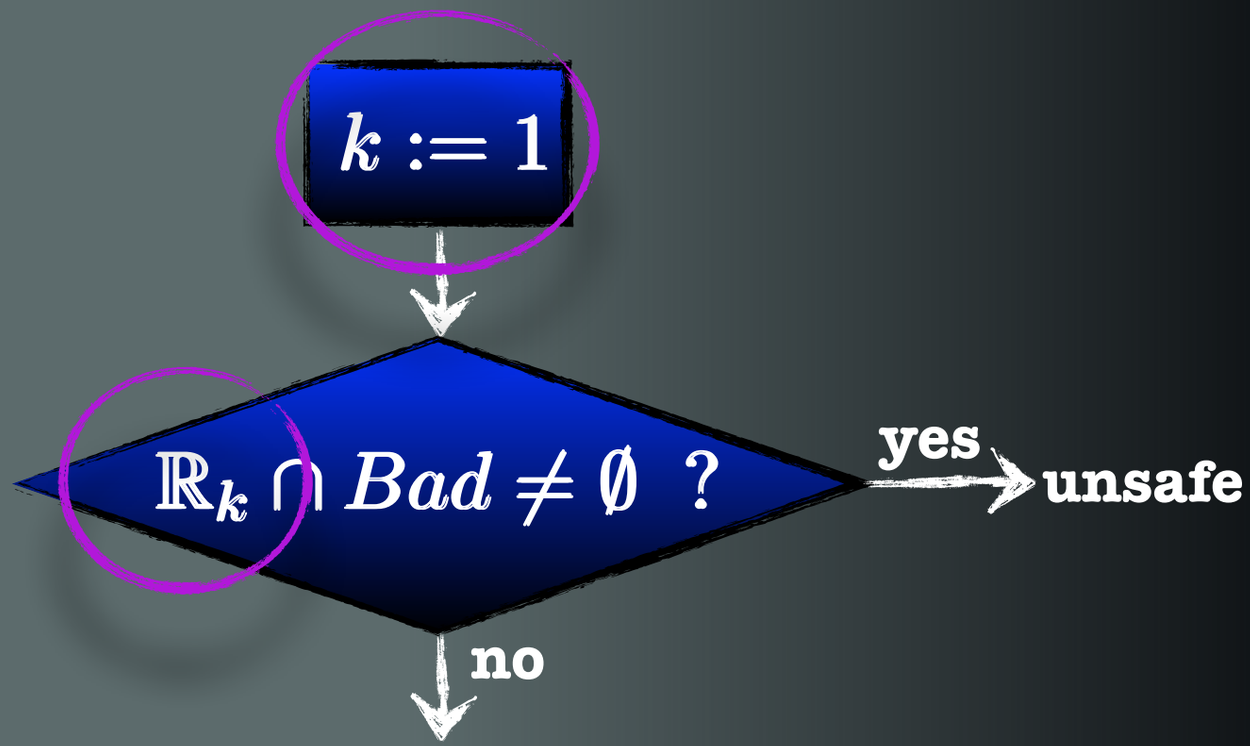

$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

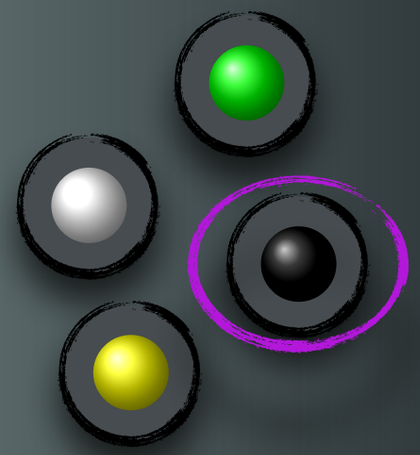
$\mathbb{R}_1$

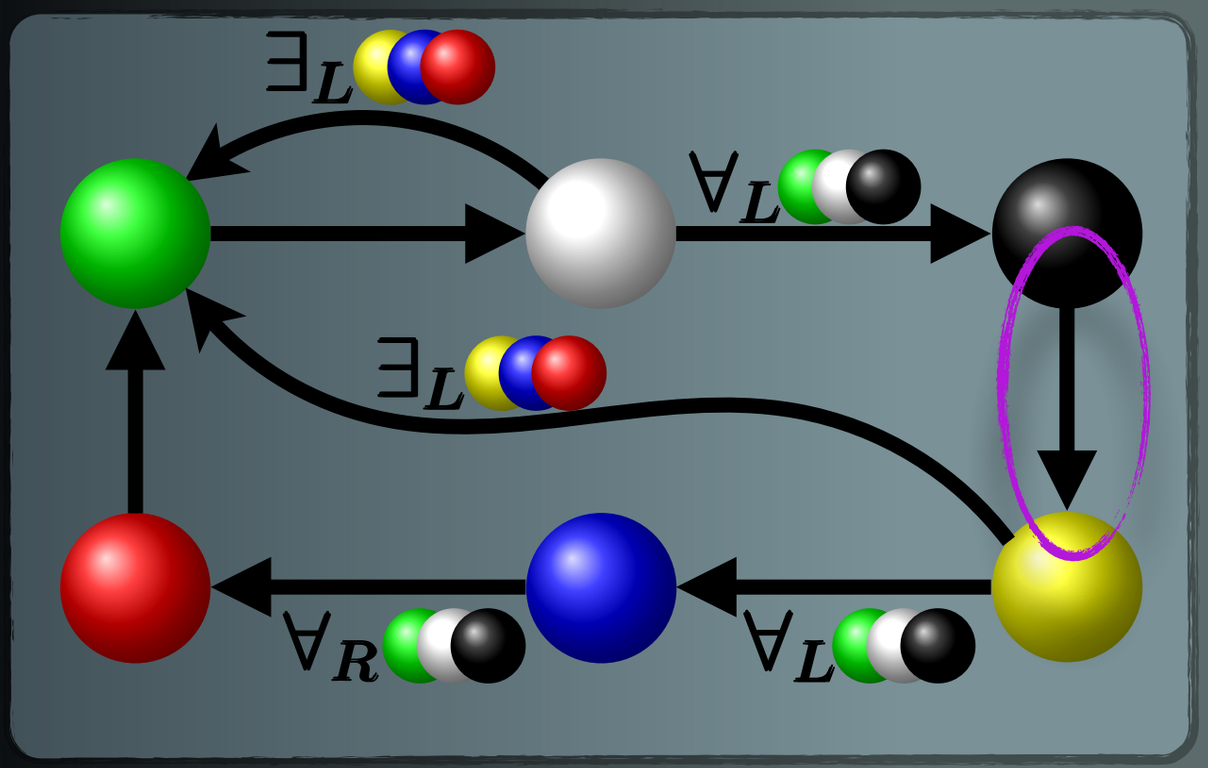


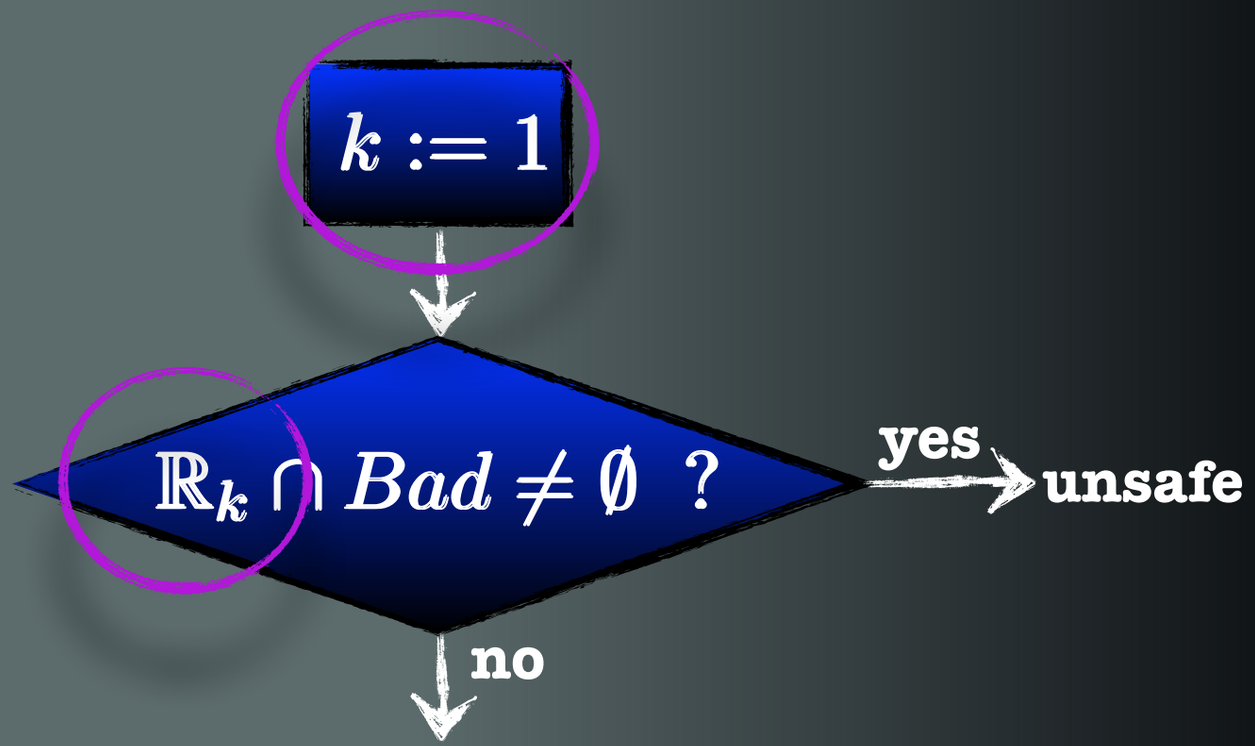

$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

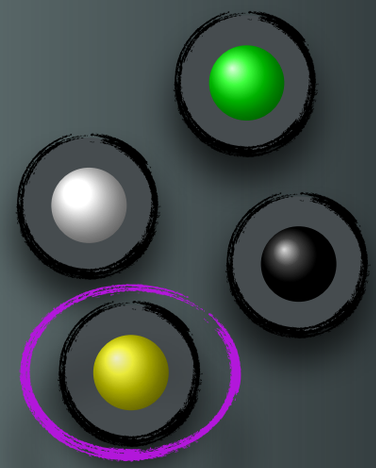
$\mathbb{R}_1$

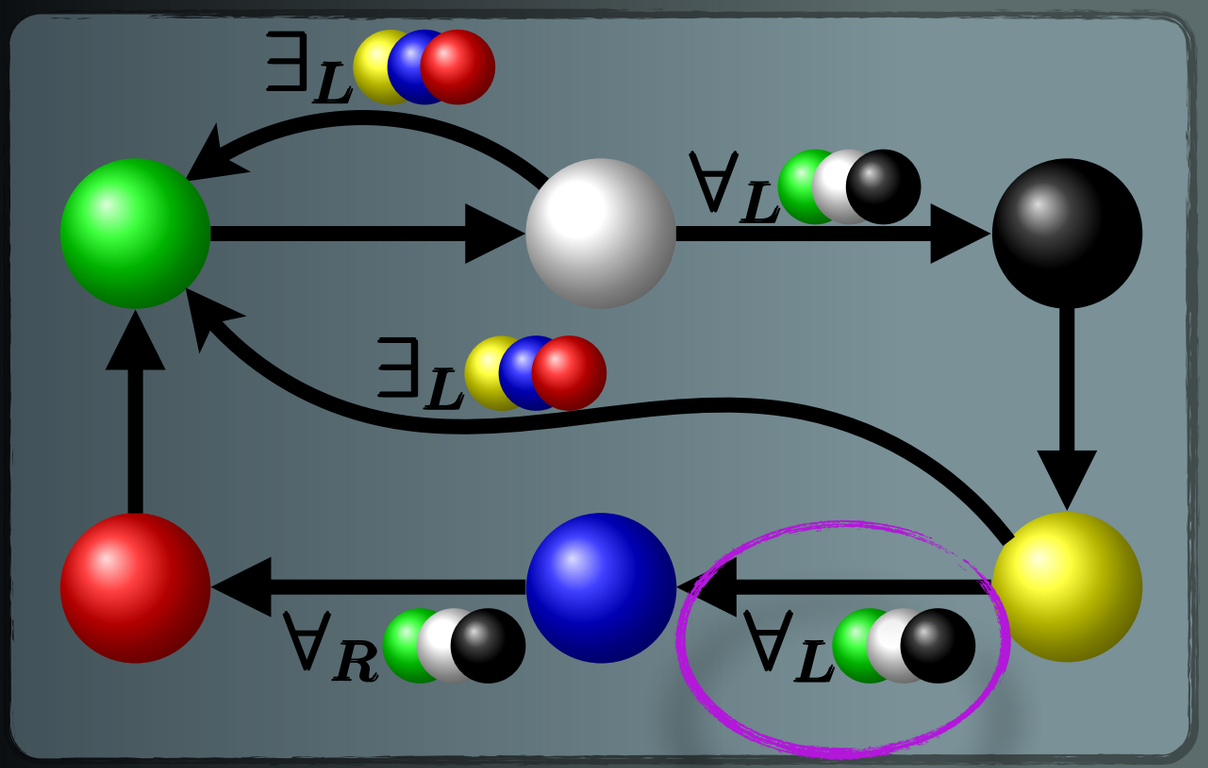


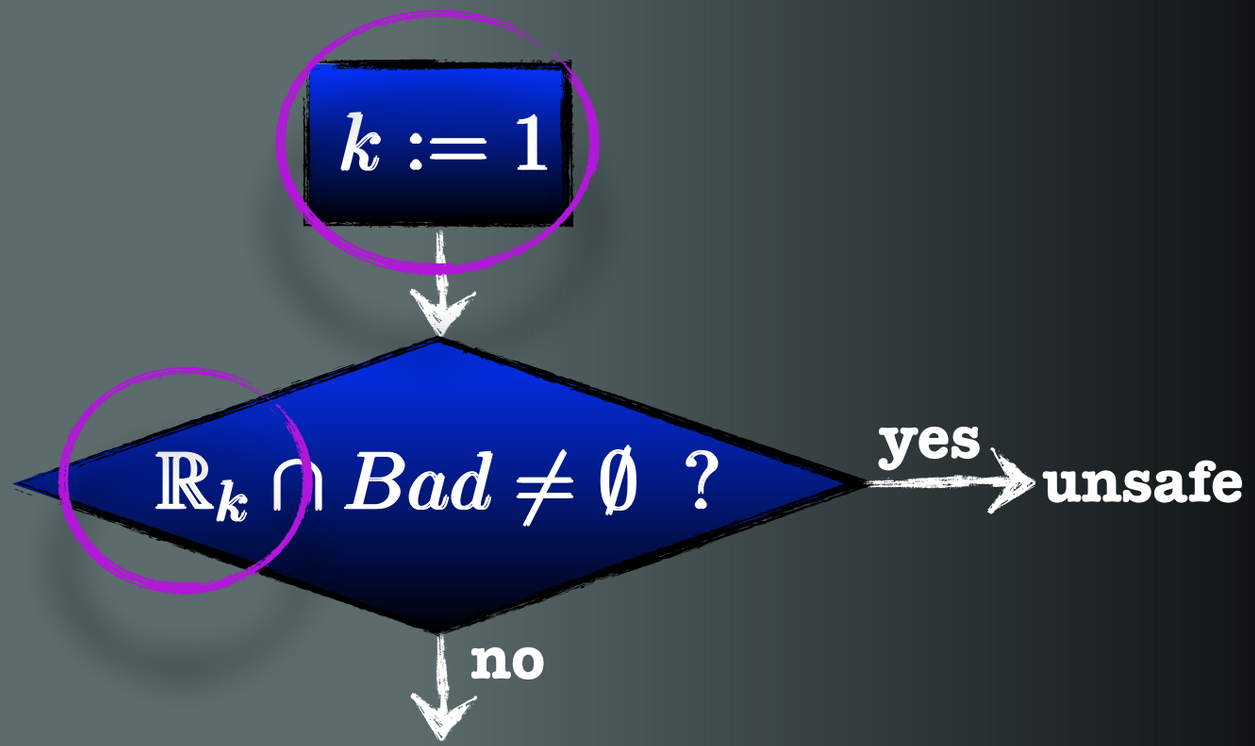

$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

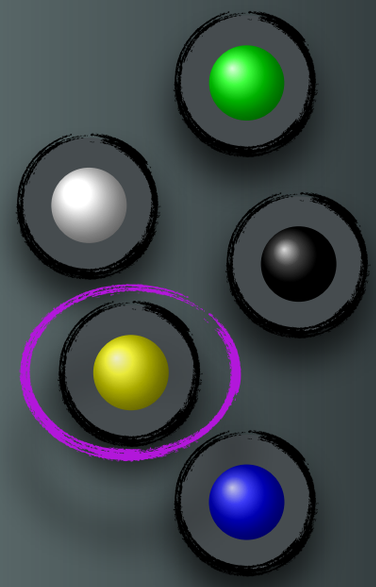
$\mathbb{R}_1$

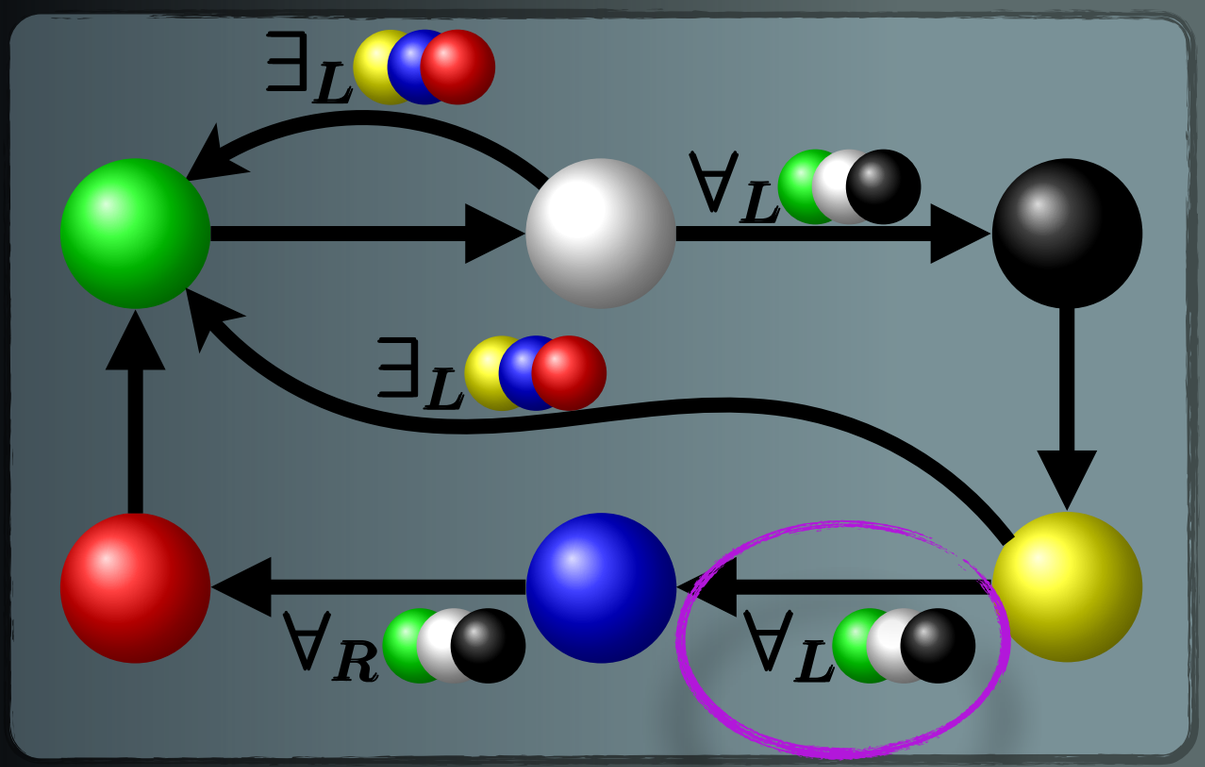


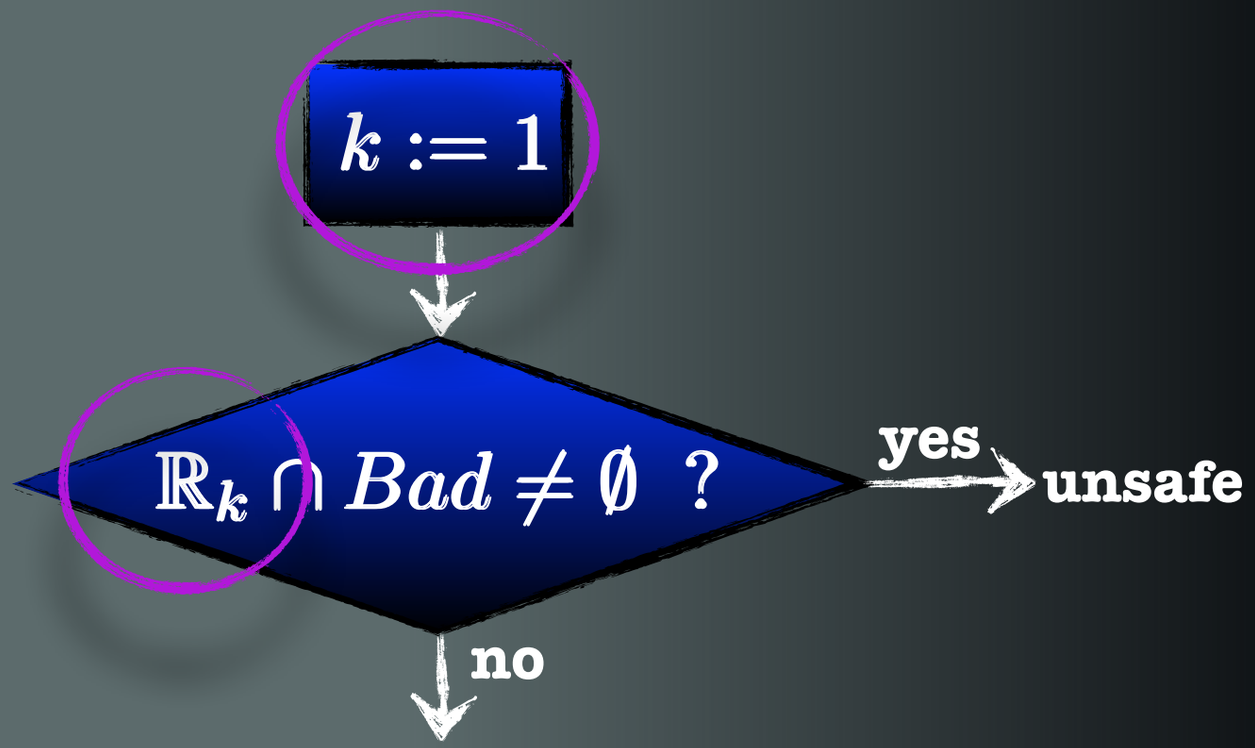

$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

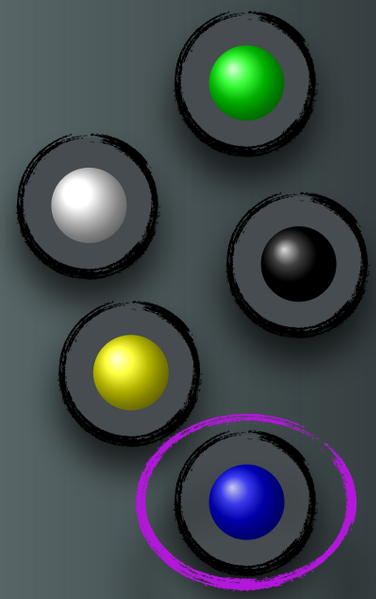
$\mathbb{R}_1$

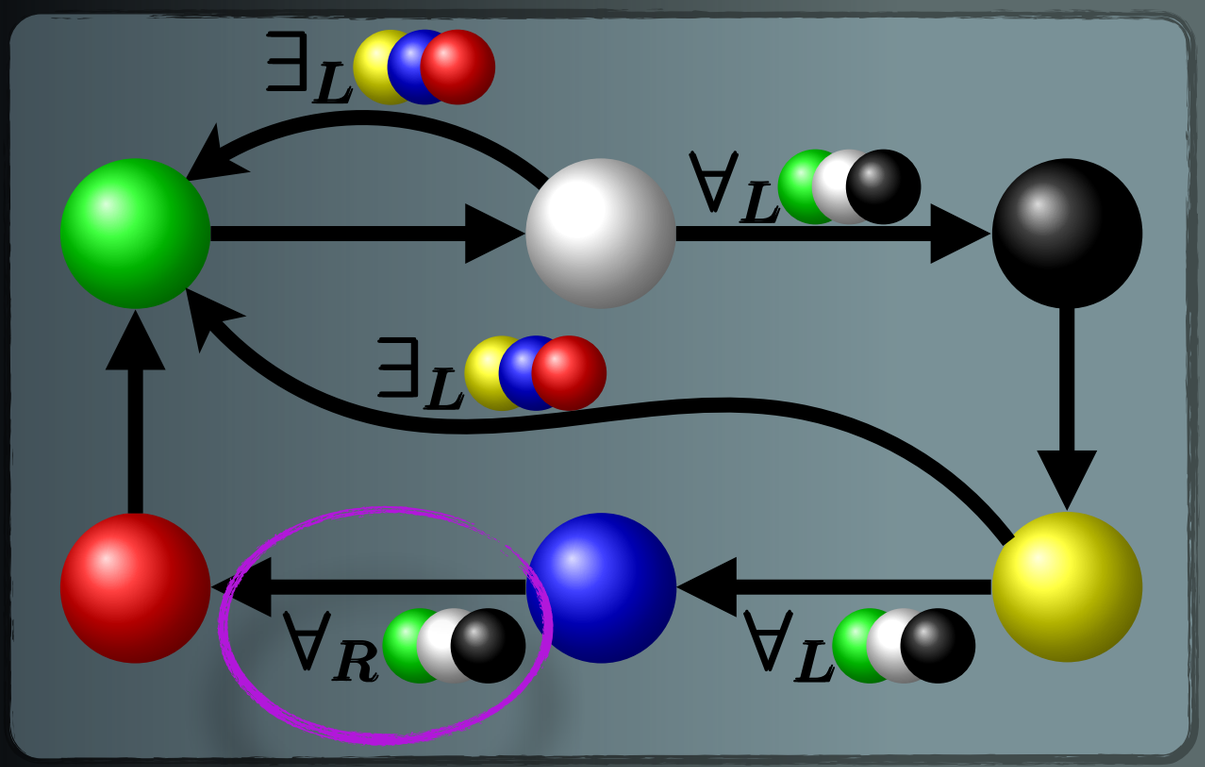


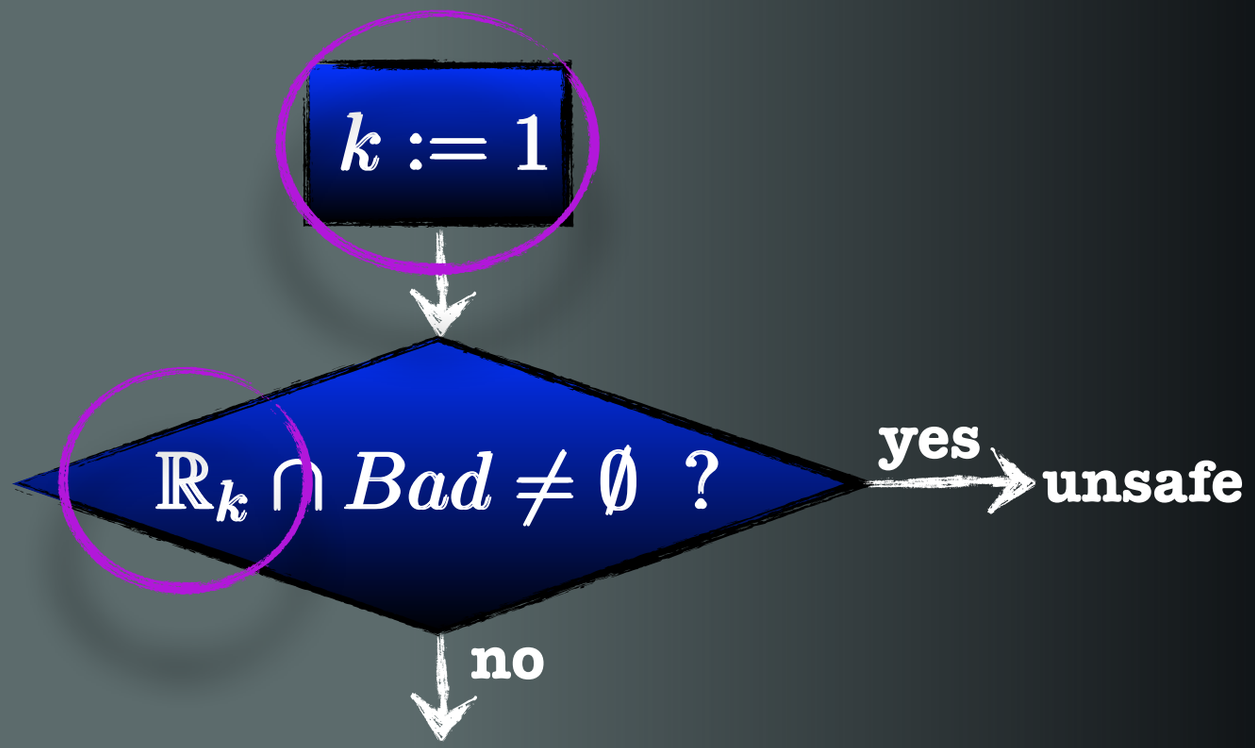

$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

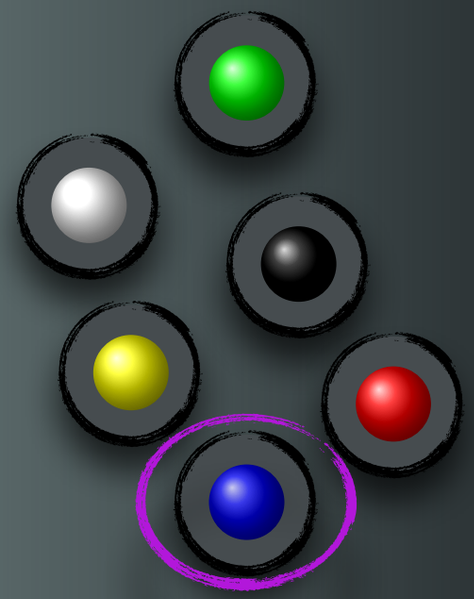
$\mathbb{R}_1$

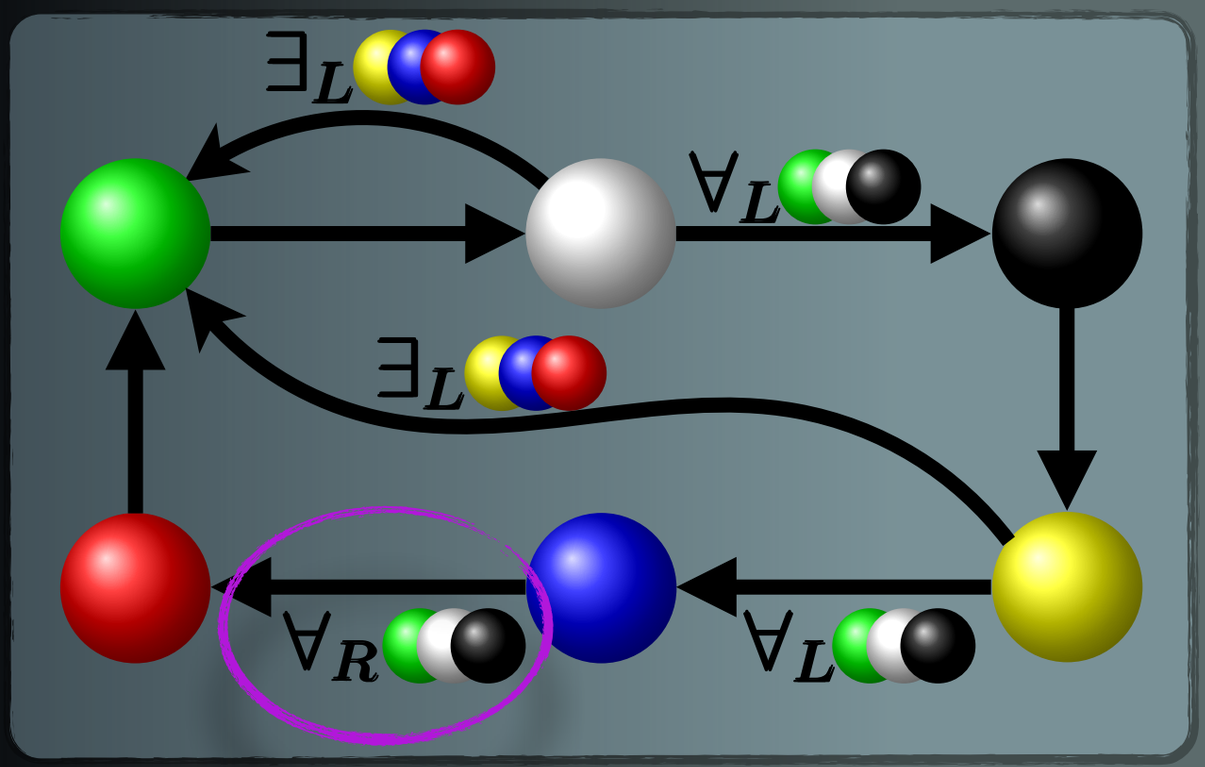


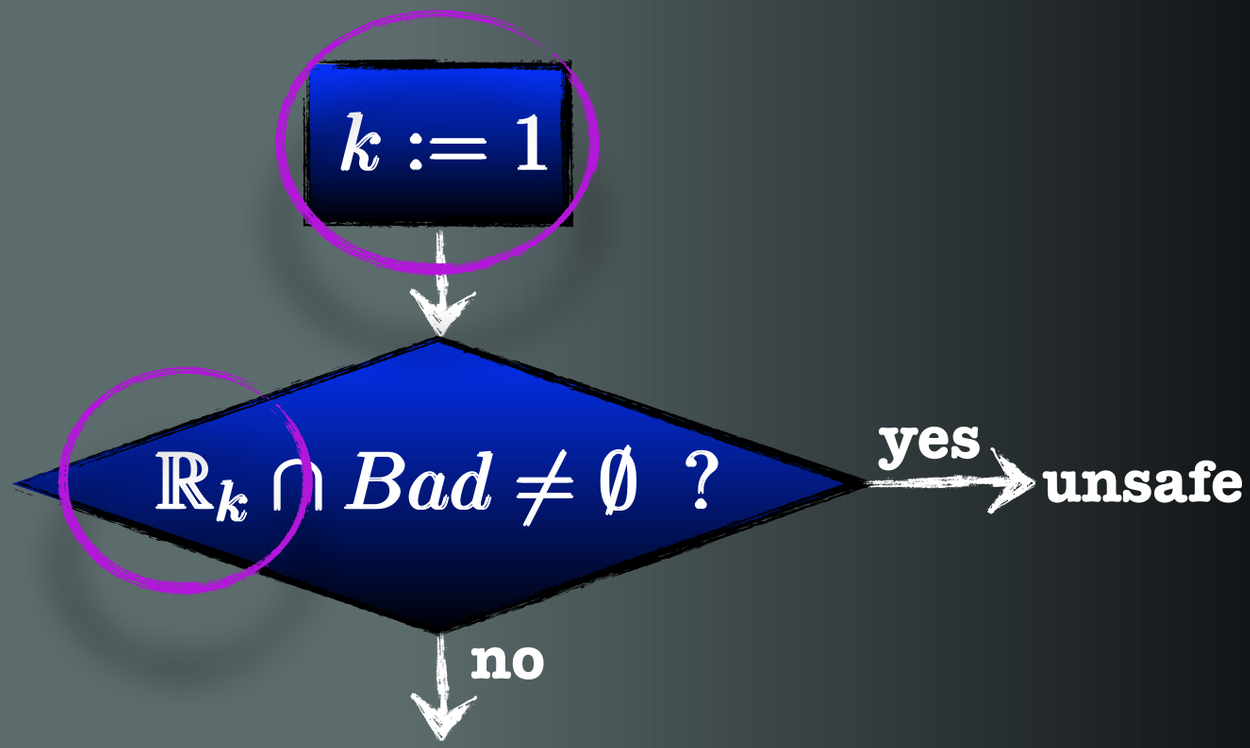

$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

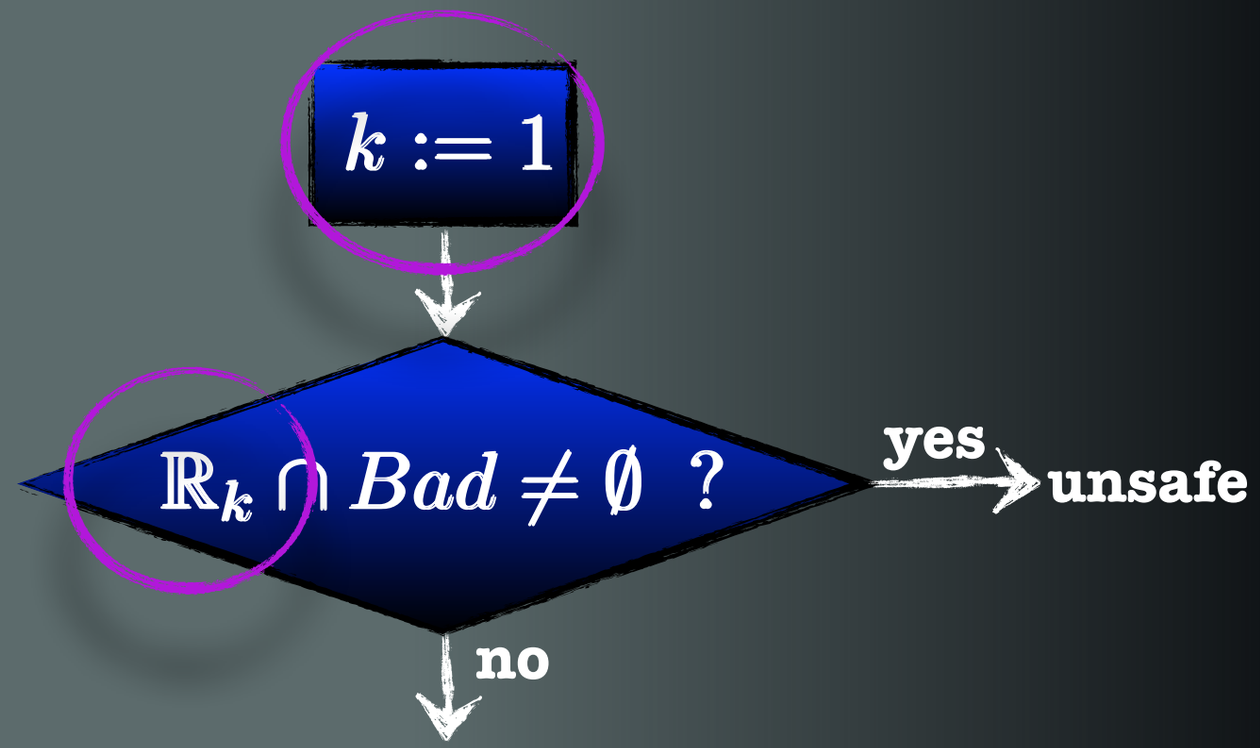
$\mathbb{R}_1$



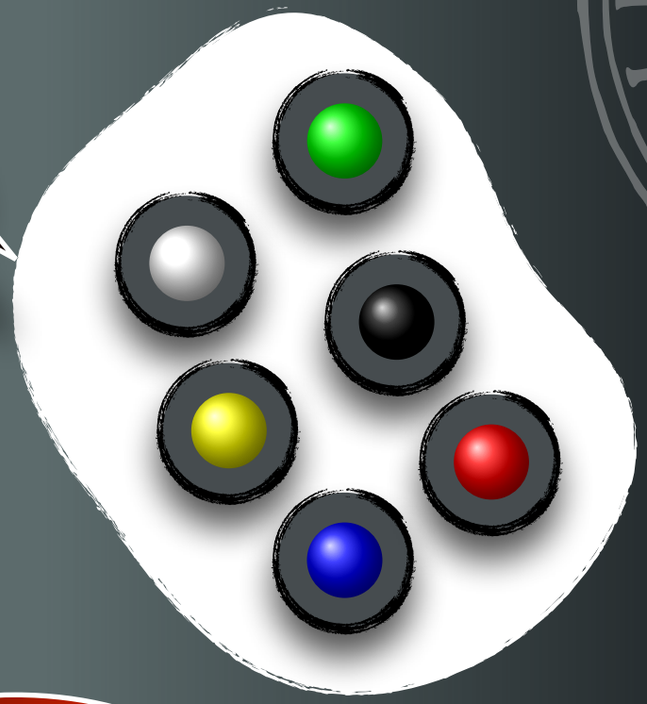

$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$

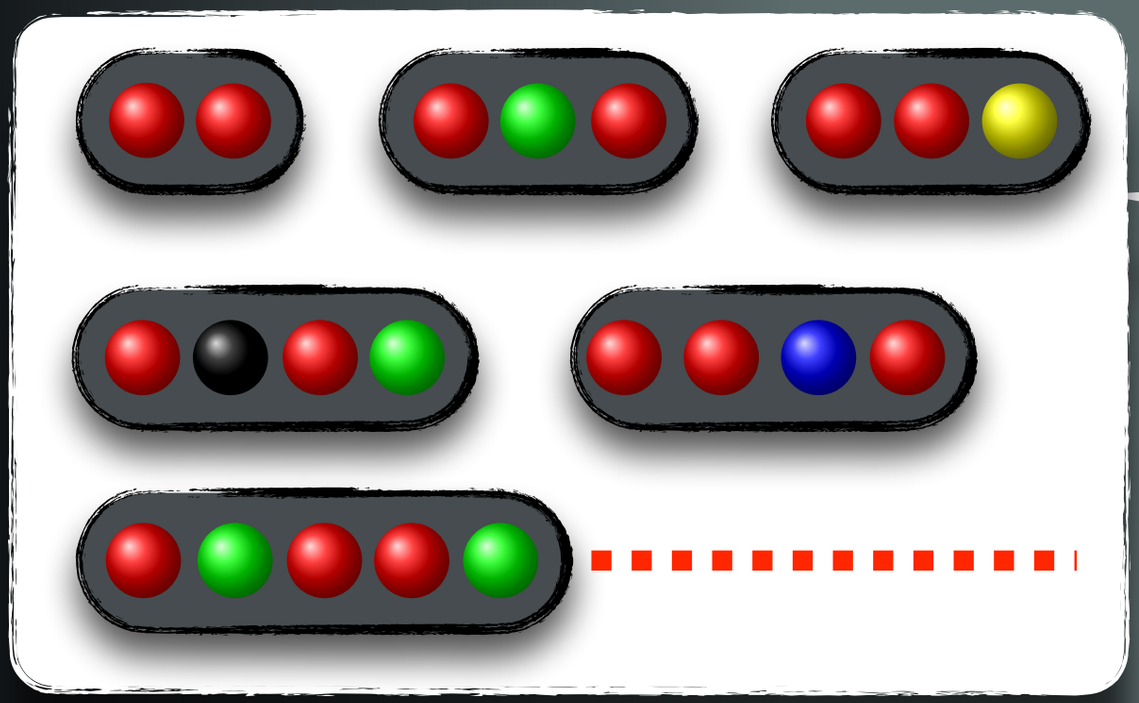

$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no


$\mathbb{R}_1$


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

$\mathbb{R}_1$





Bad Configurations

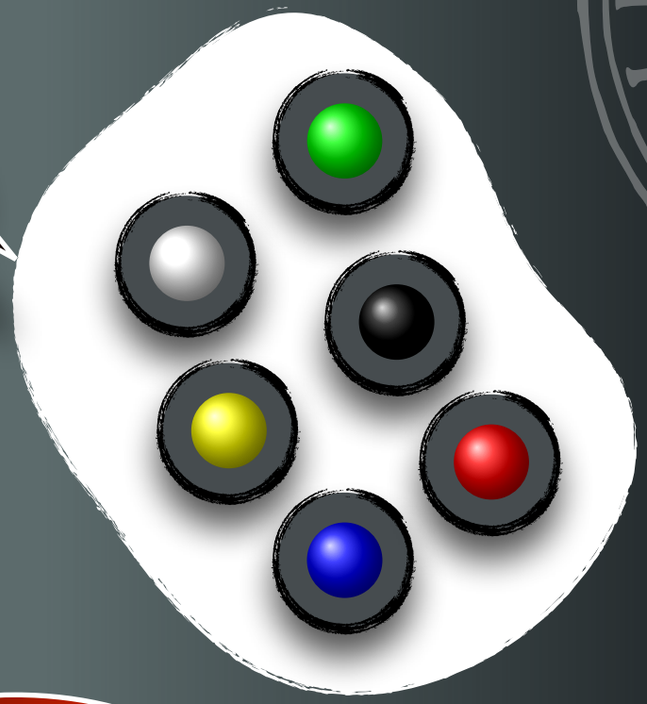
$k := 1$

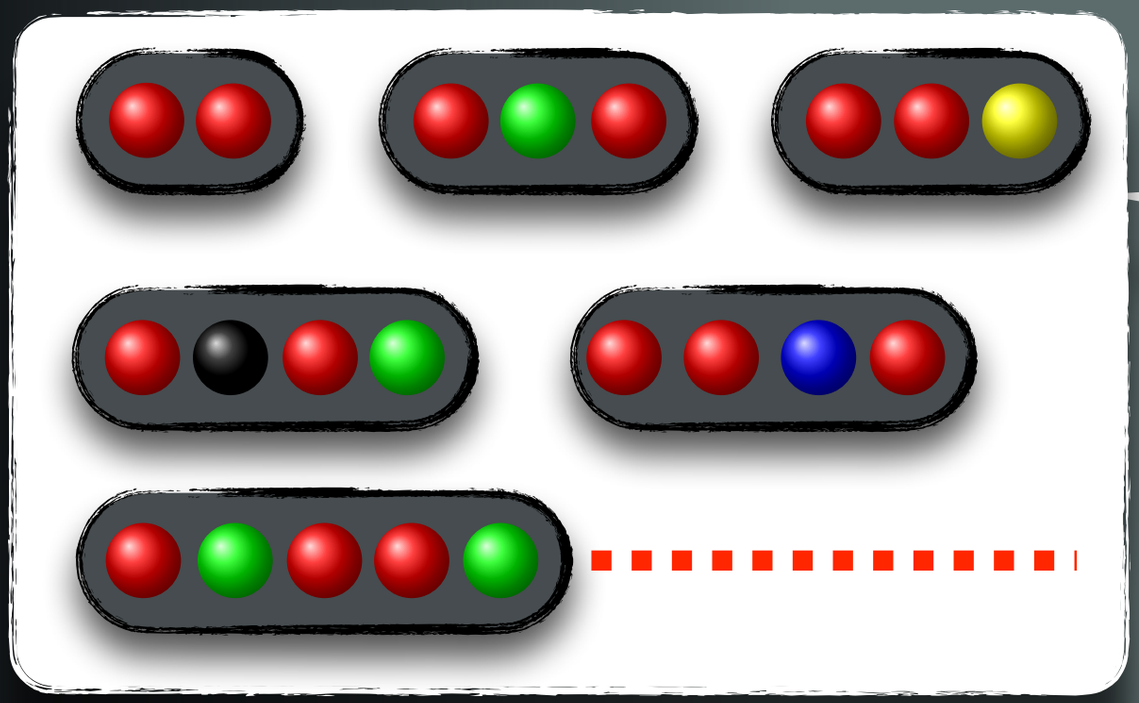
$\mathbb{R}_k \cap Bad \neq \emptyset$?

yes → **unsafe**

no

$\mathbb{R}_1$





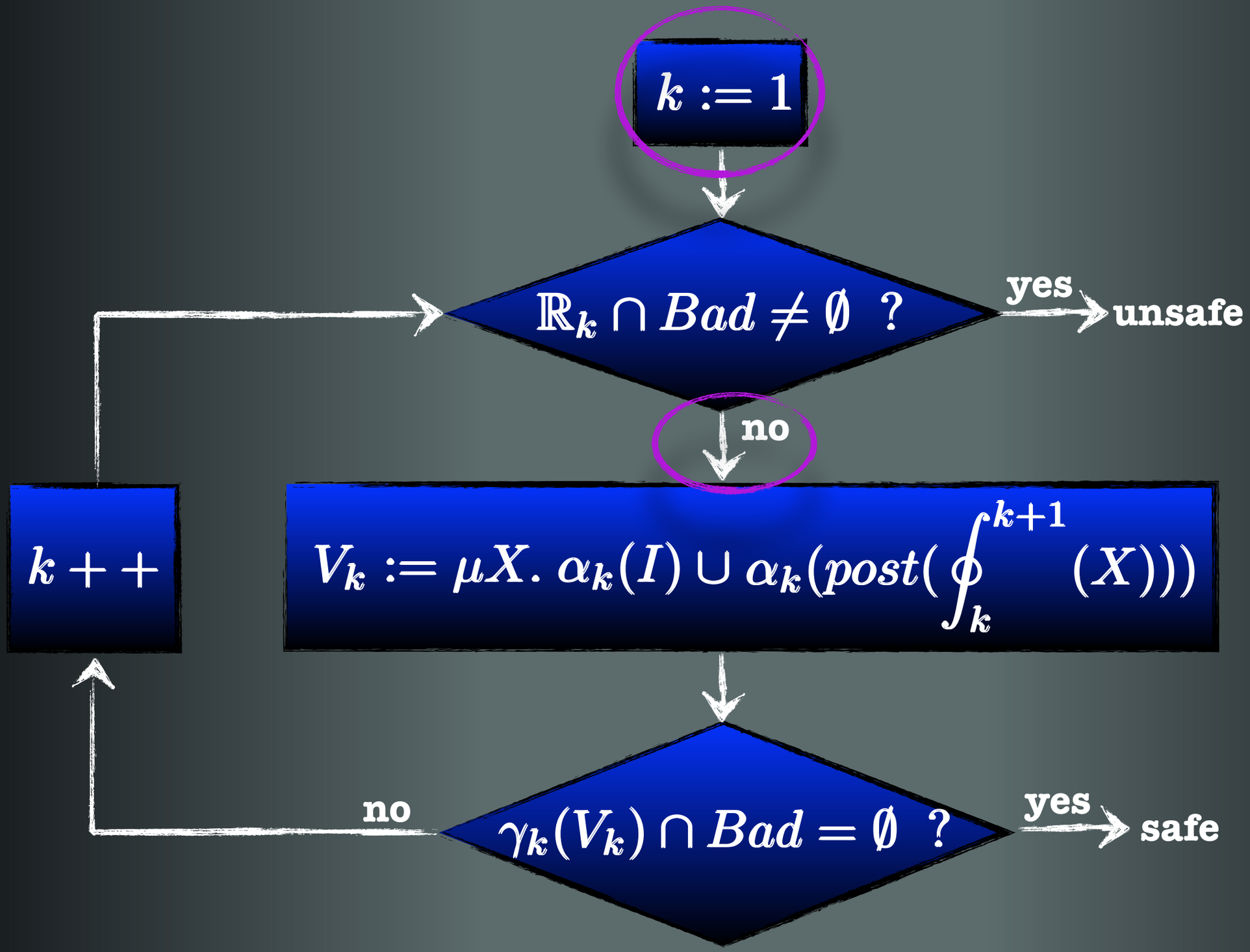
Bad Configurations

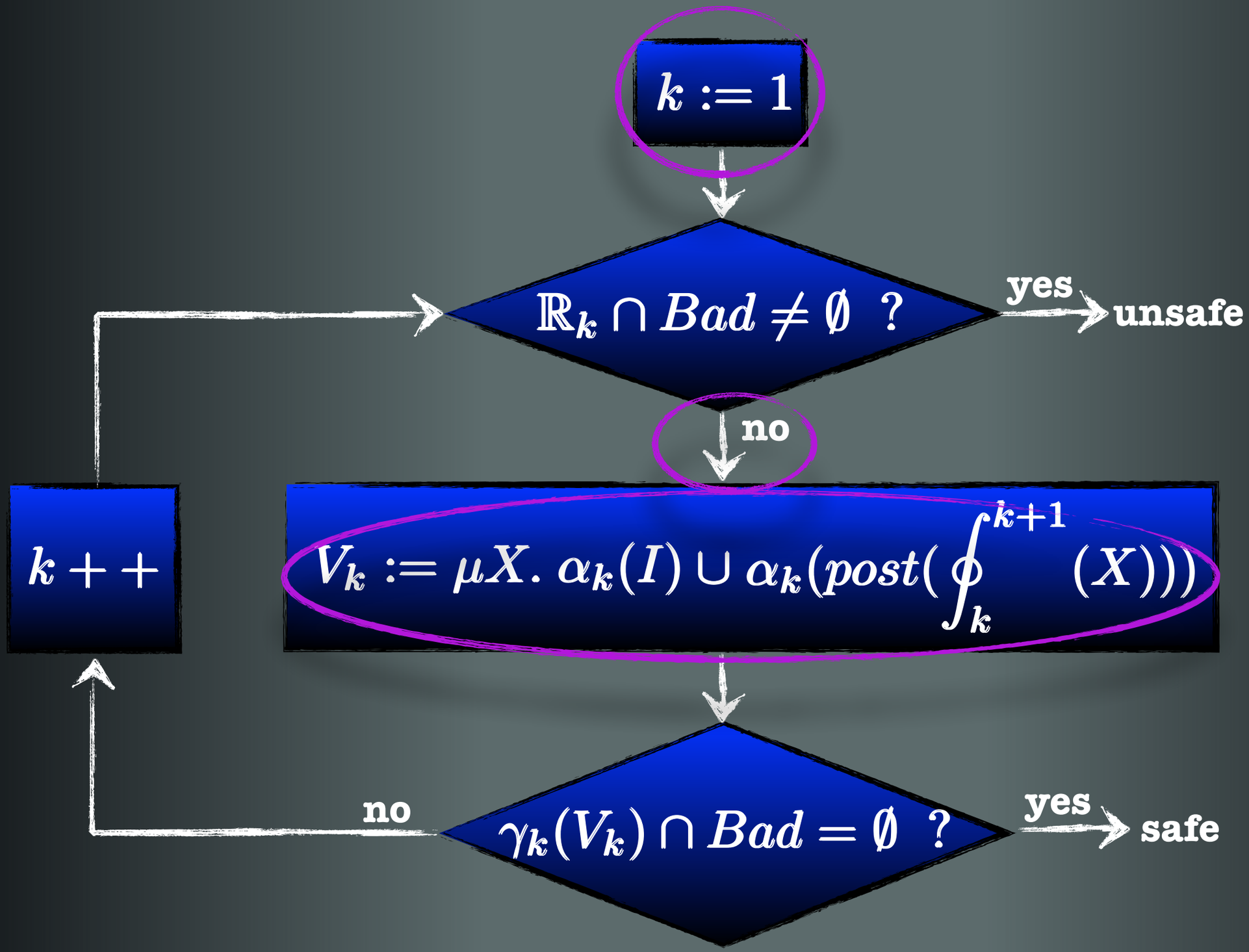
$k := 1$

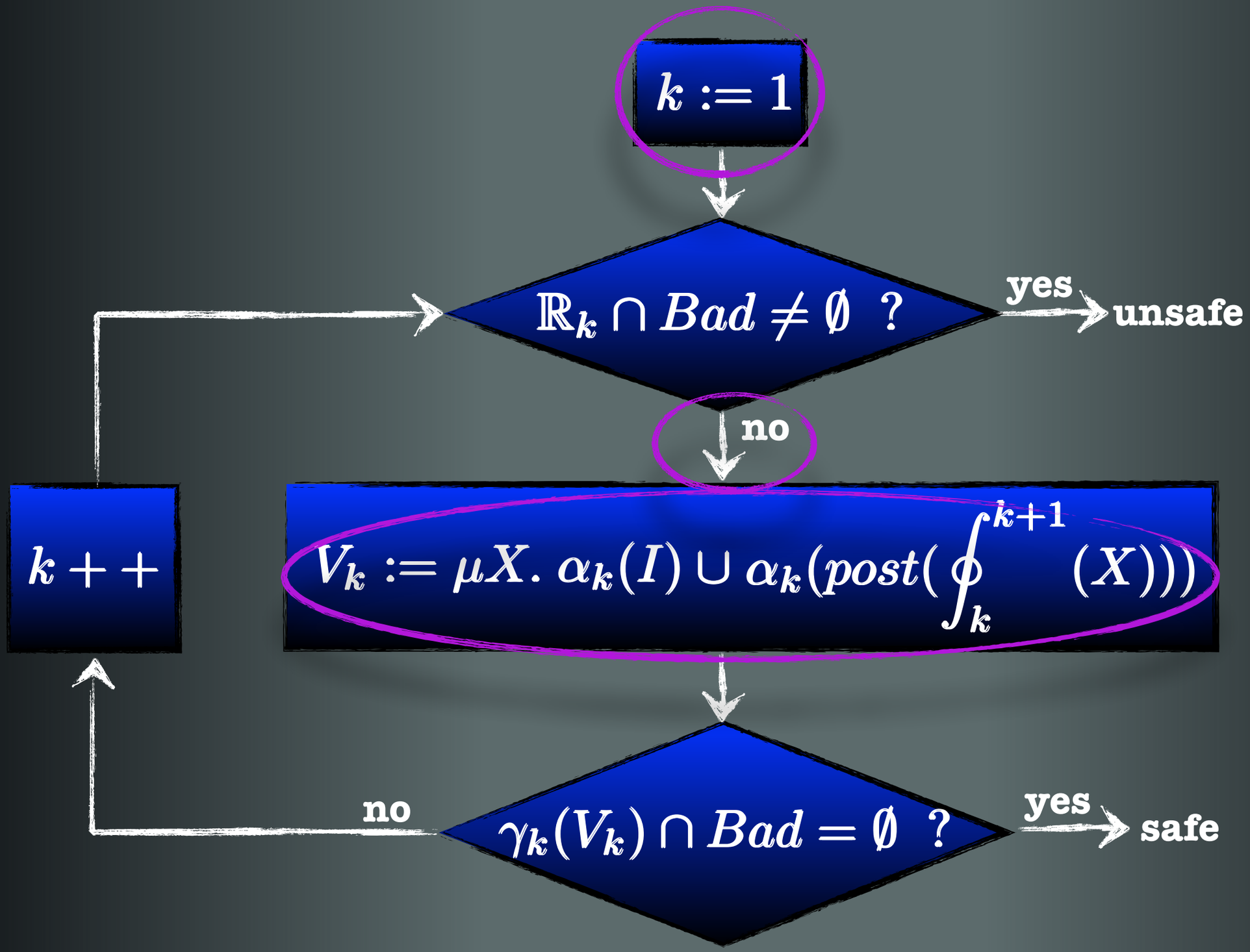
$\mathbb{R}_k \cap Bad \neq \emptyset$?

yes → unsafe

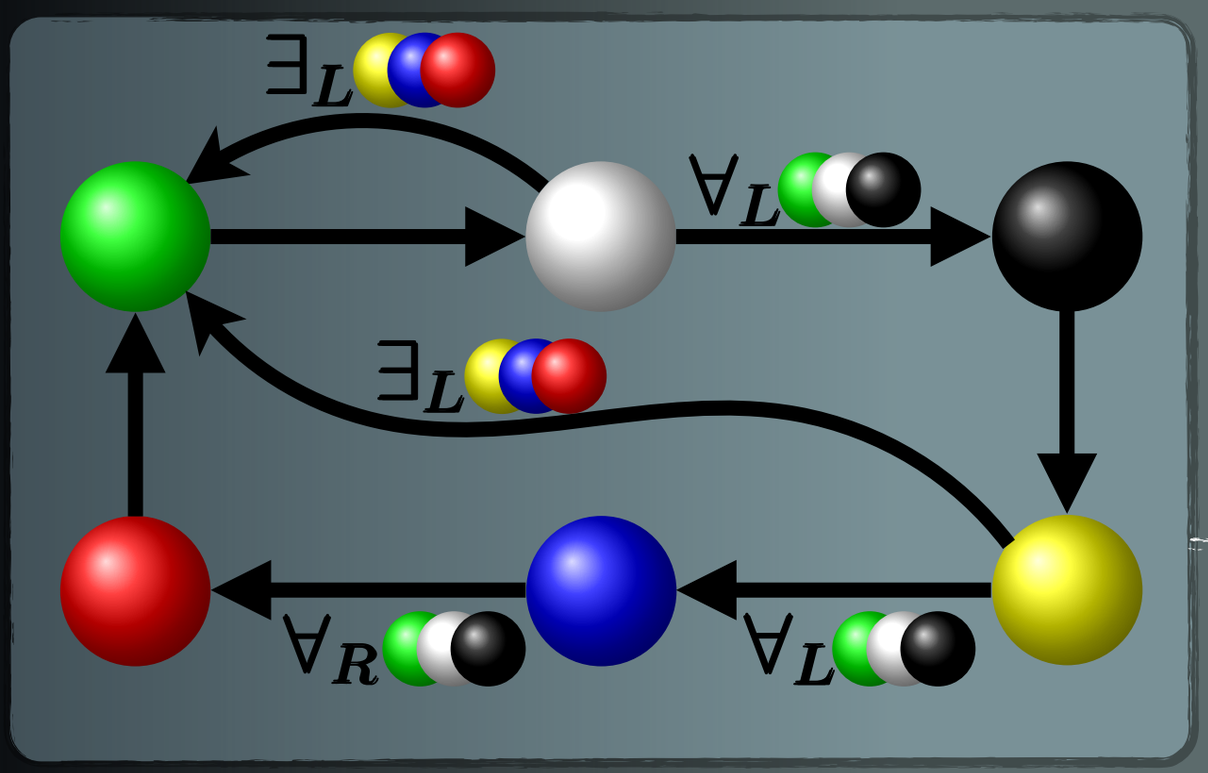
no


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$k++$
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$
$\gamma_k(V_k) \cap Bad = \emptyset$?
no
yes
safe


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$k++$
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$
$\gamma_k(V_k) \cap Bad = \emptyset$?
no
yes
safe

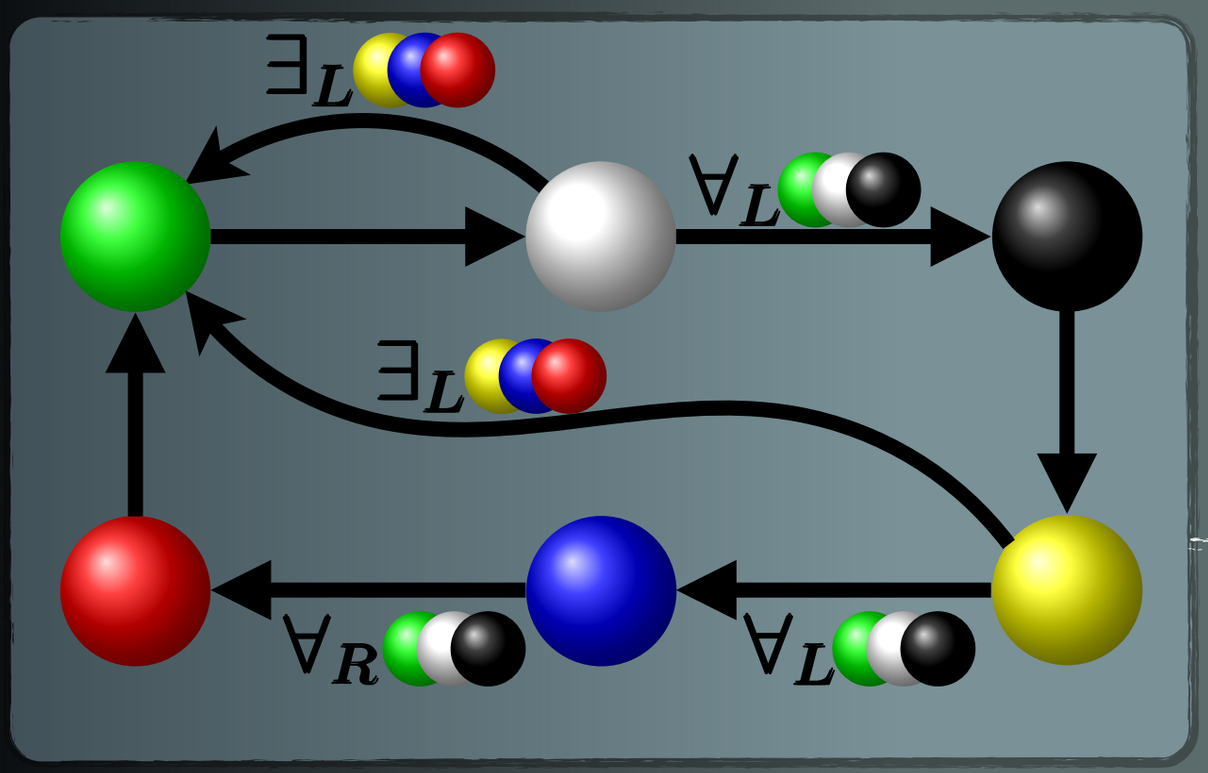


$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^{2}(X)))$$



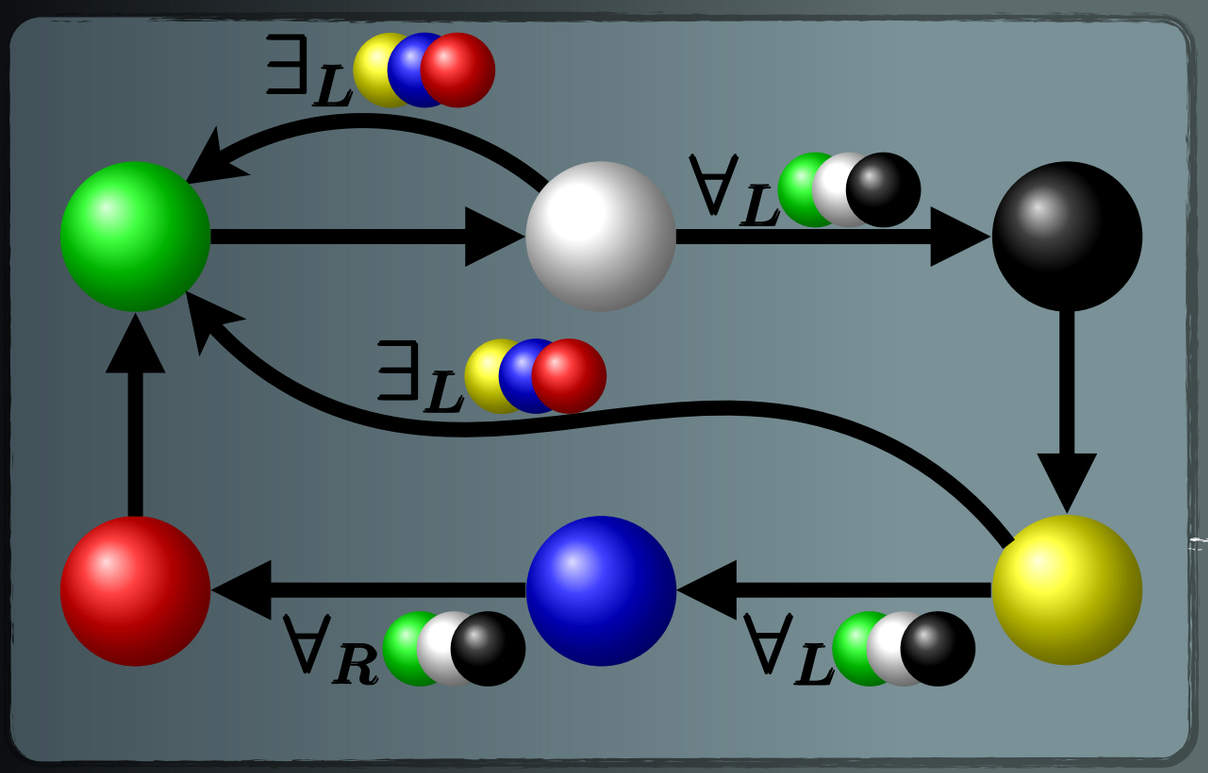
$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$

X



$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$

X



$$V_1 := \mu X.\, \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$

X

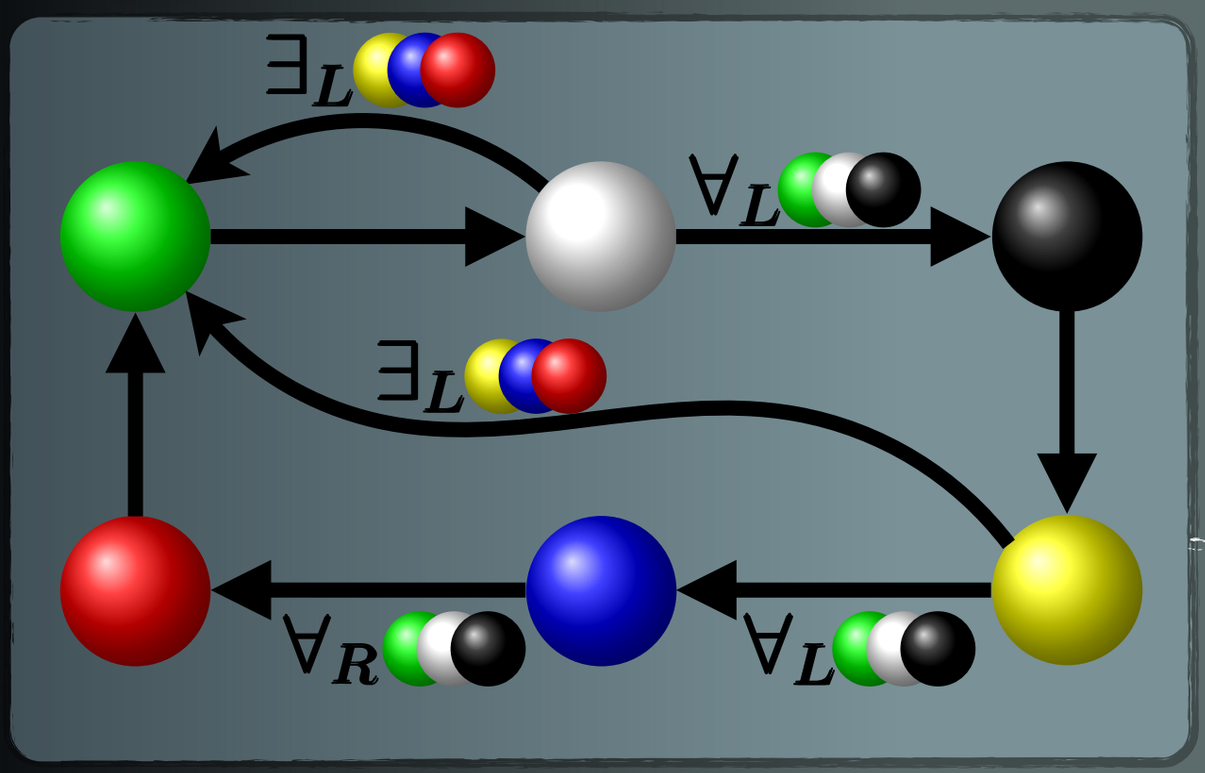
$Init_5$

$Init_4$

$Init_3$

$Init_2$

$Init_1$



$$V_1 := \mu X.\, \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$

X

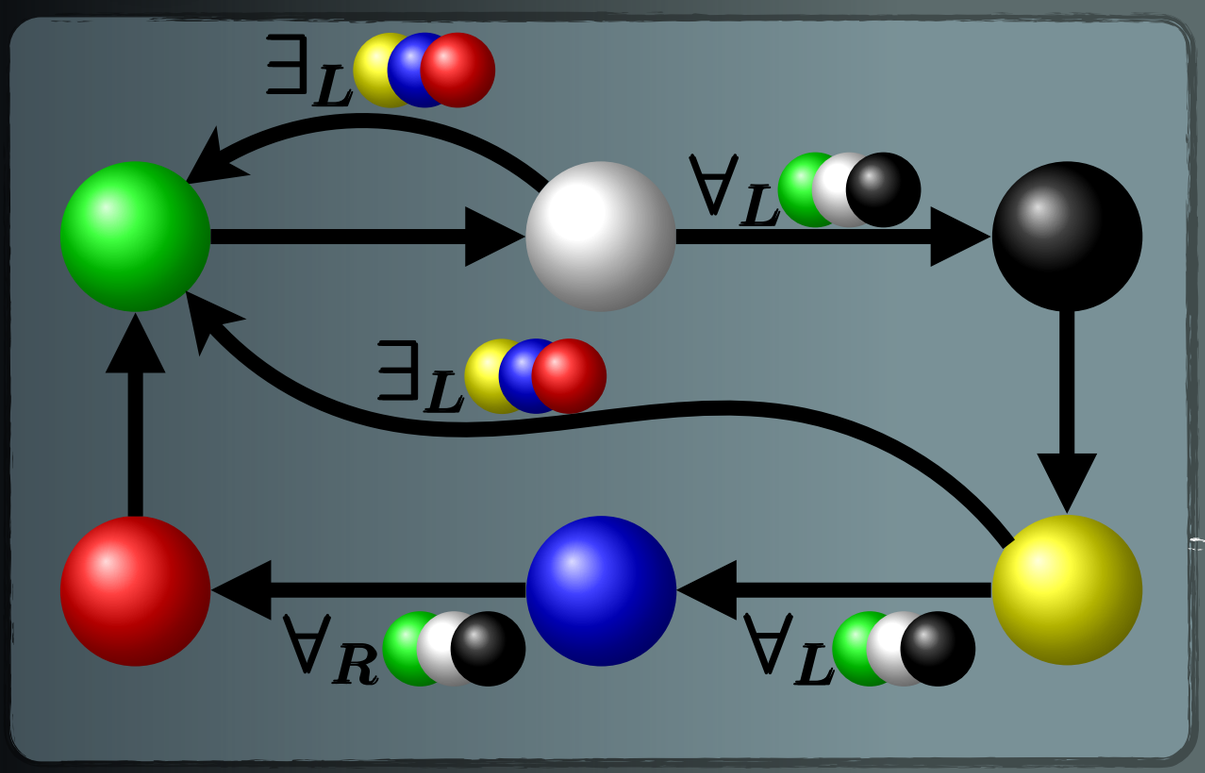
$Init_5$

$Init_4$

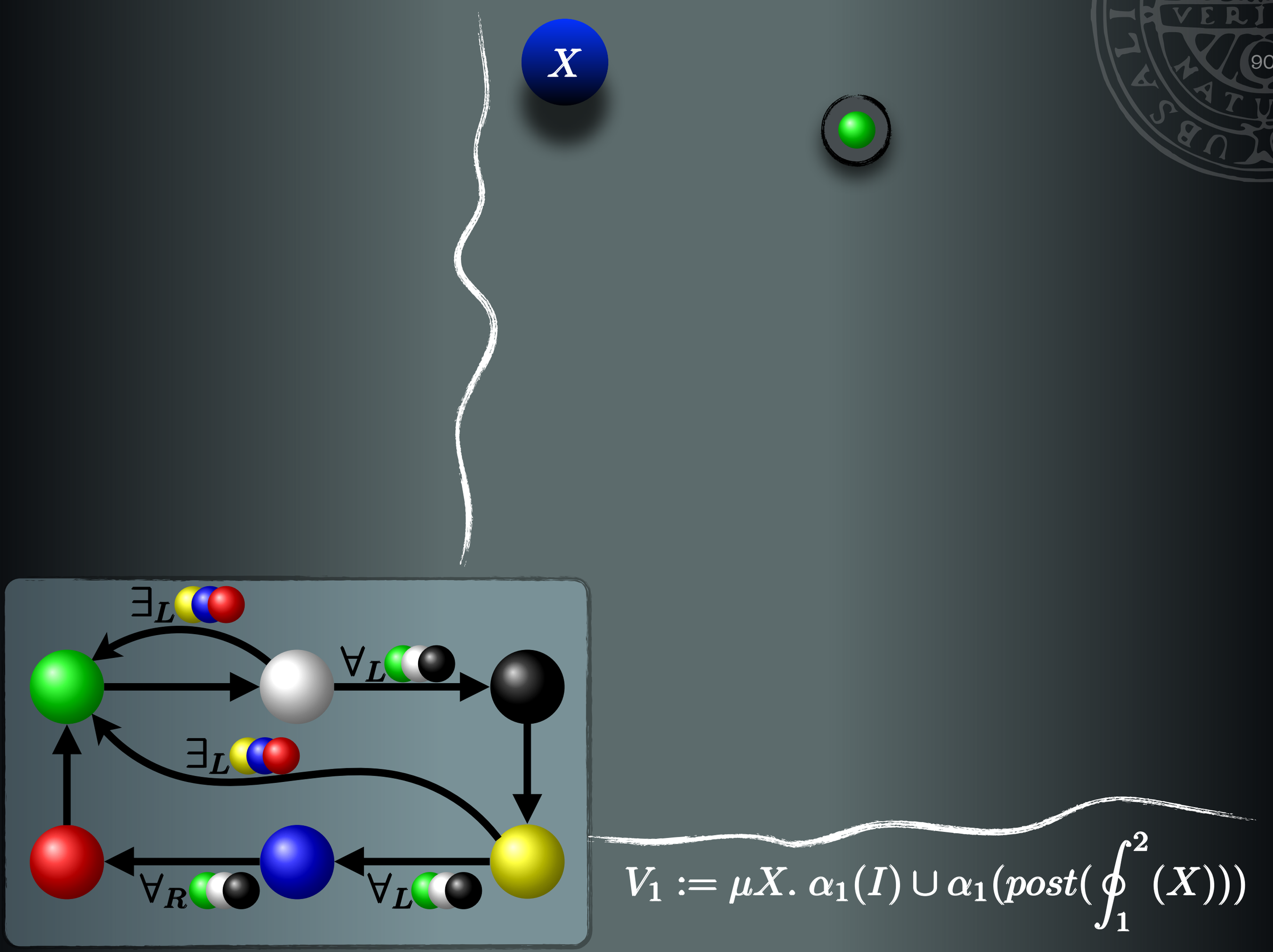
$Init_3$

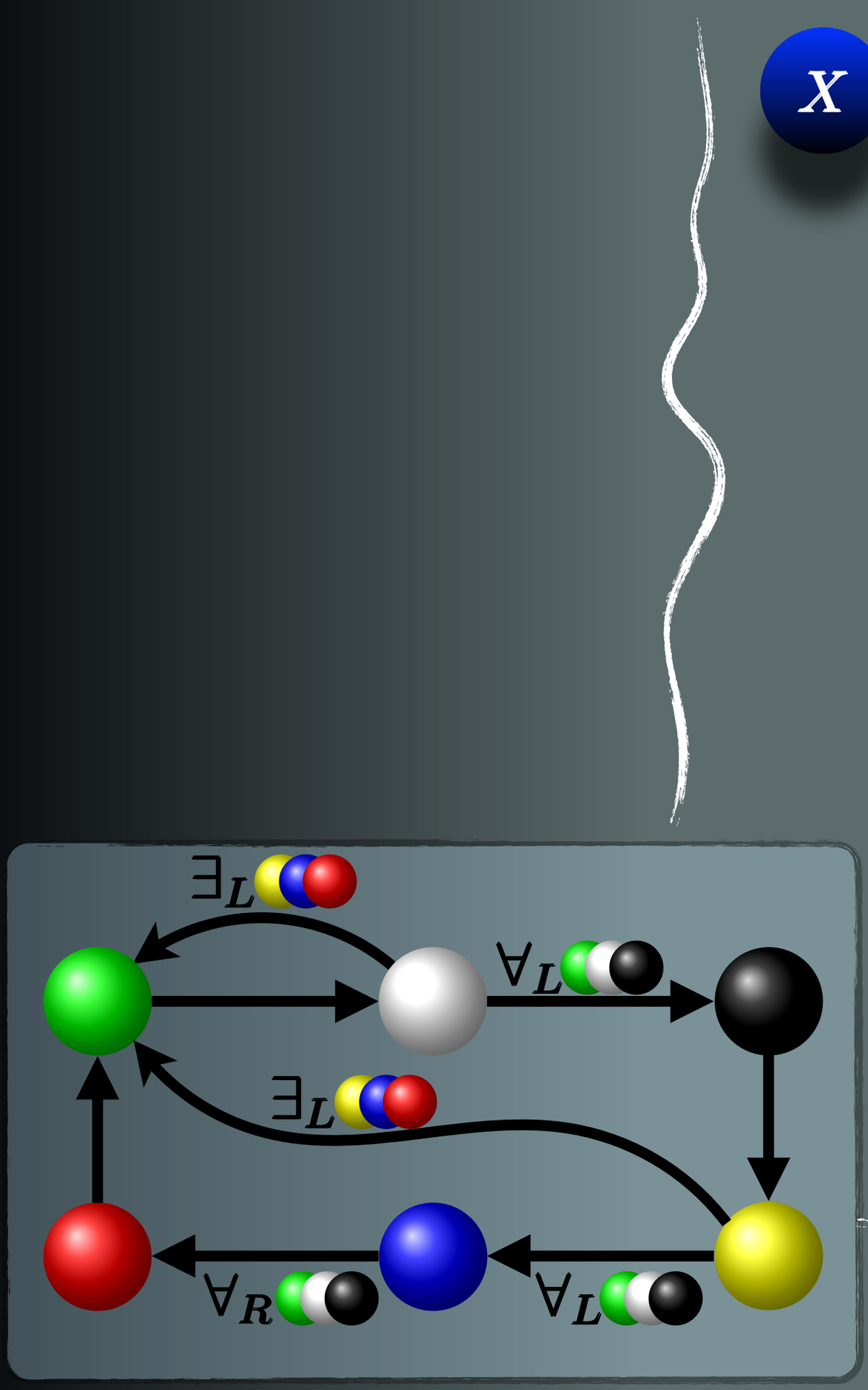
$Init_2$

$Init_1$



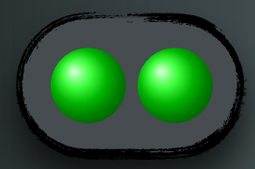
$$V_1 := \mu X.\, \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$


X
$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$
$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$

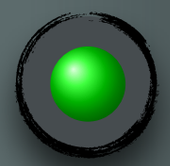


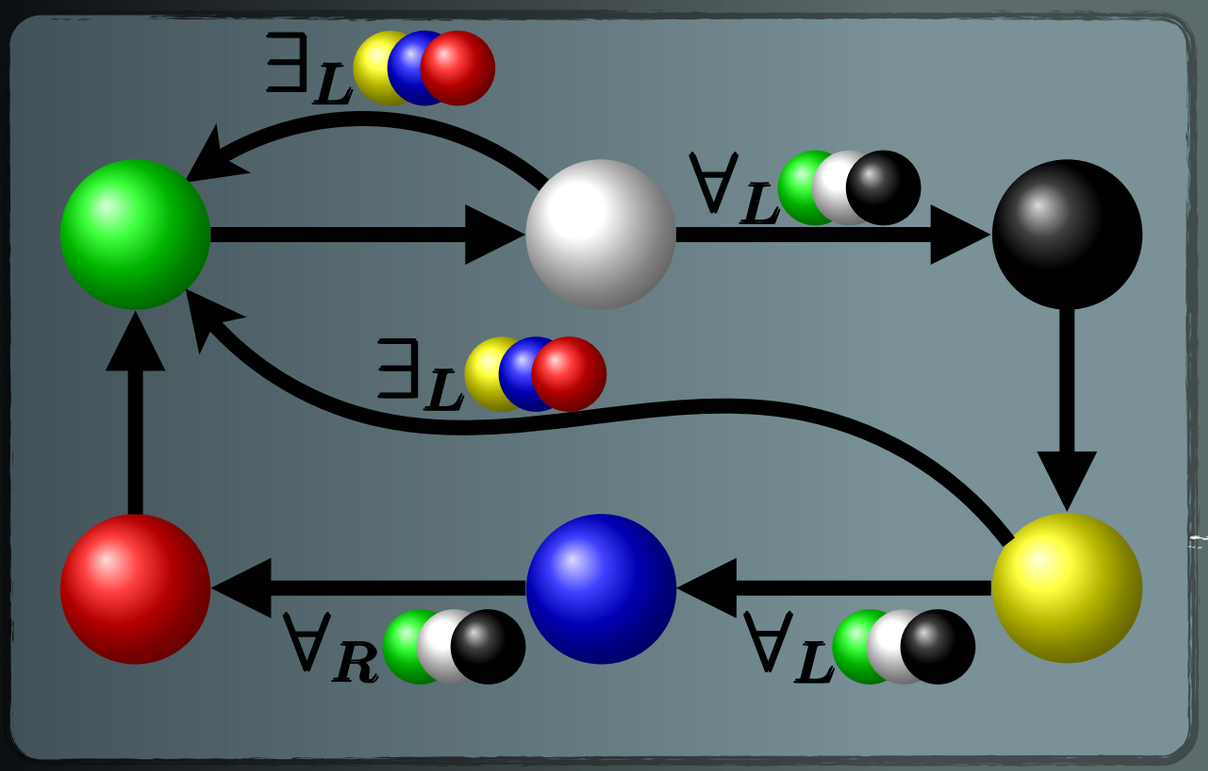
X

$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$

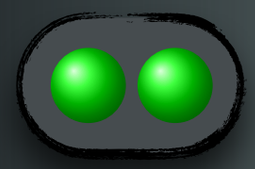




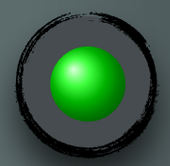


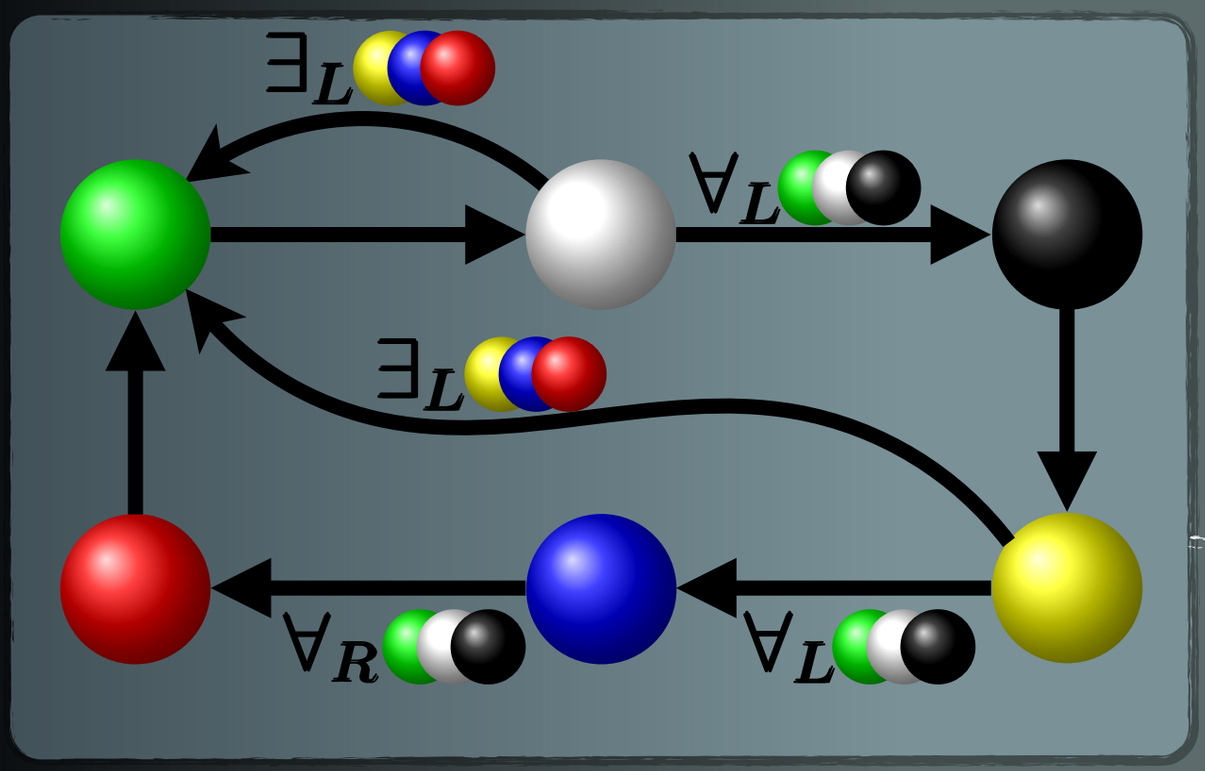


$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$

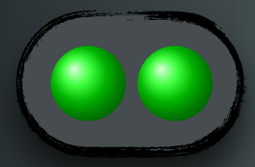




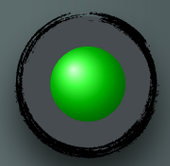


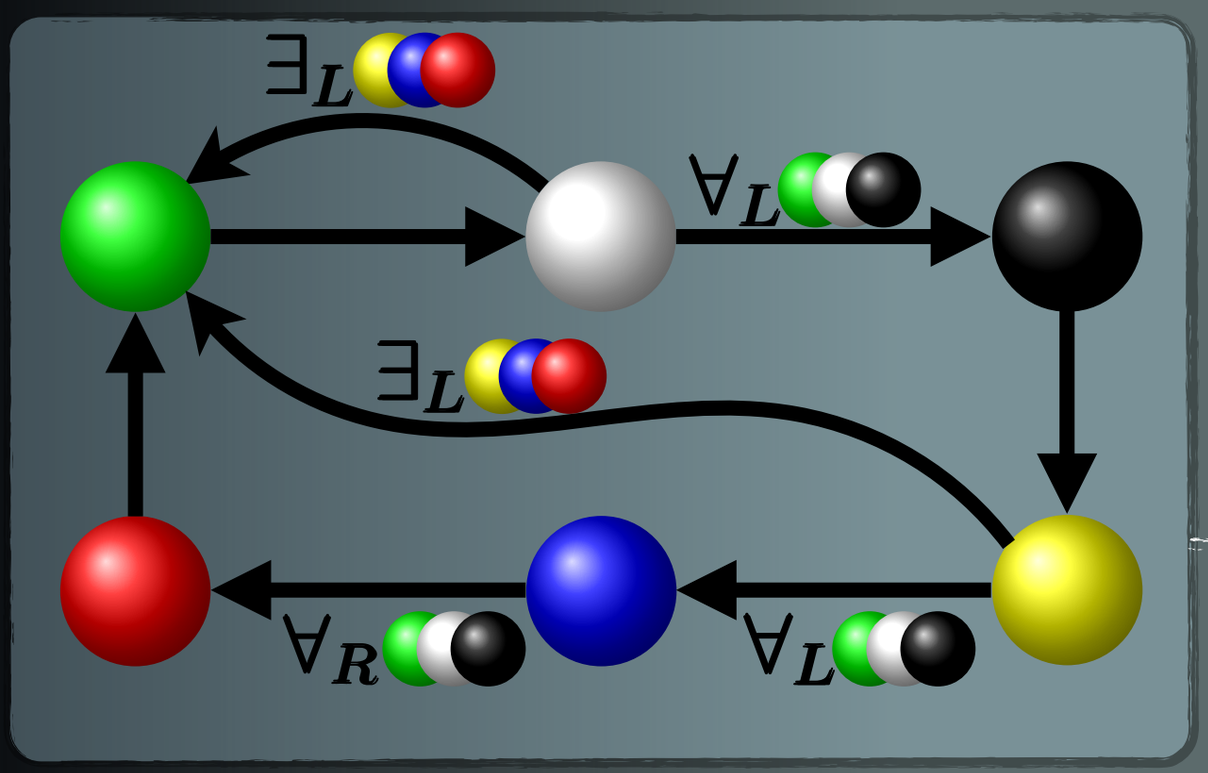


$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$

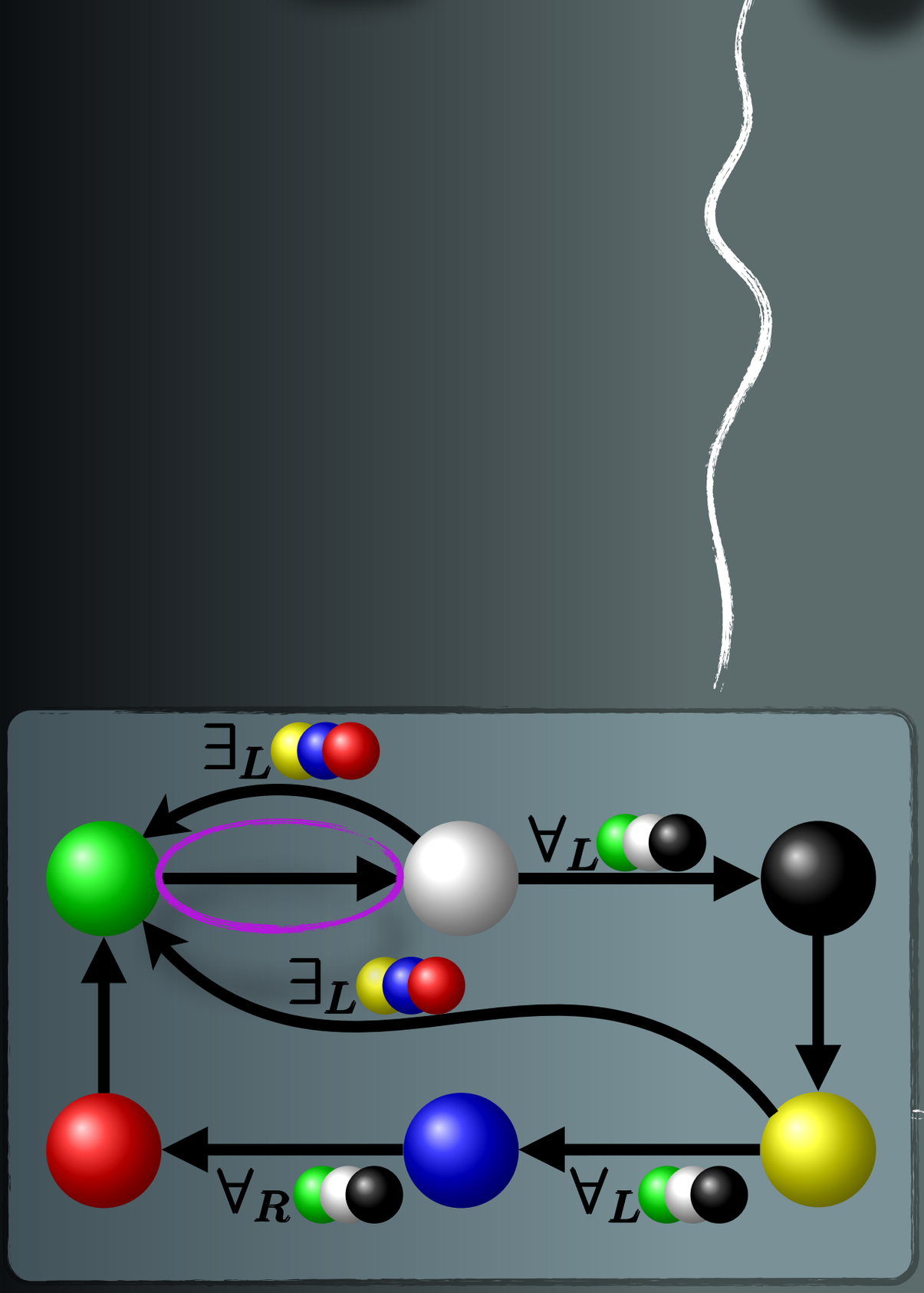








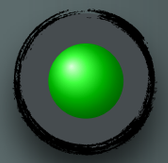
$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2(X)))$$

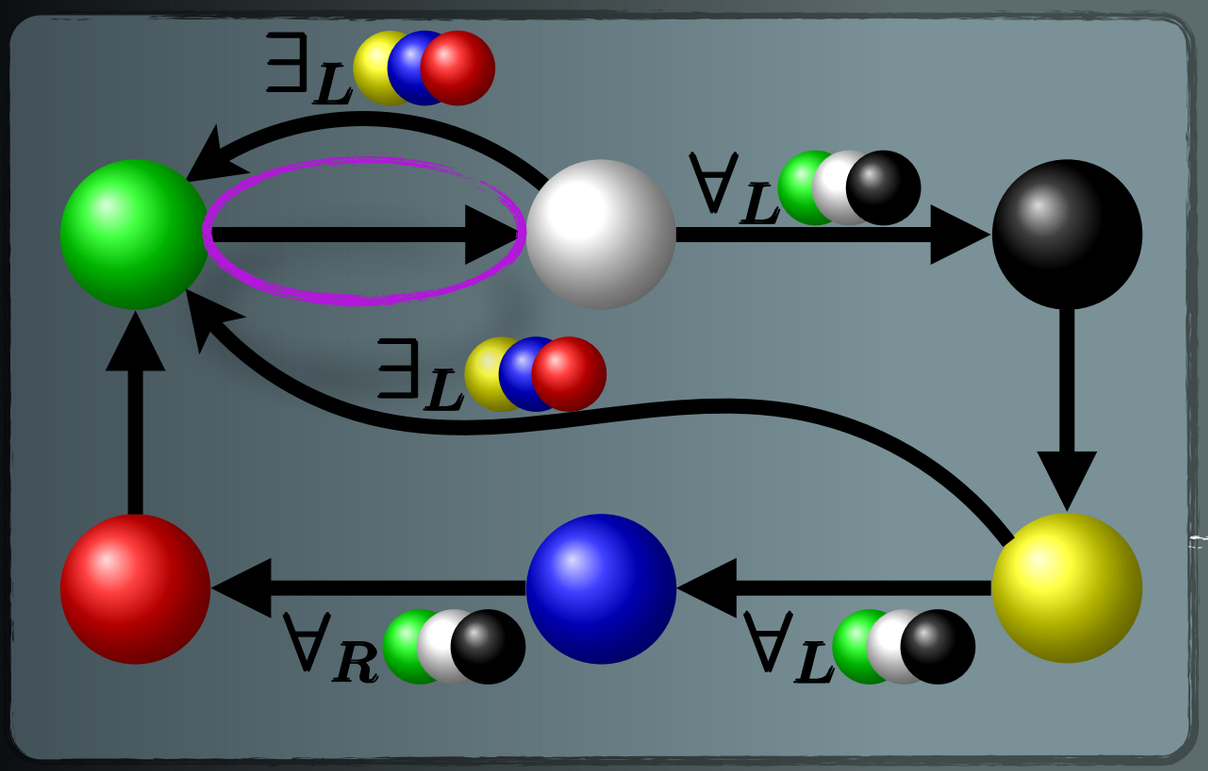


$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$





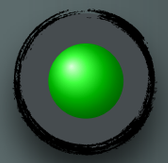


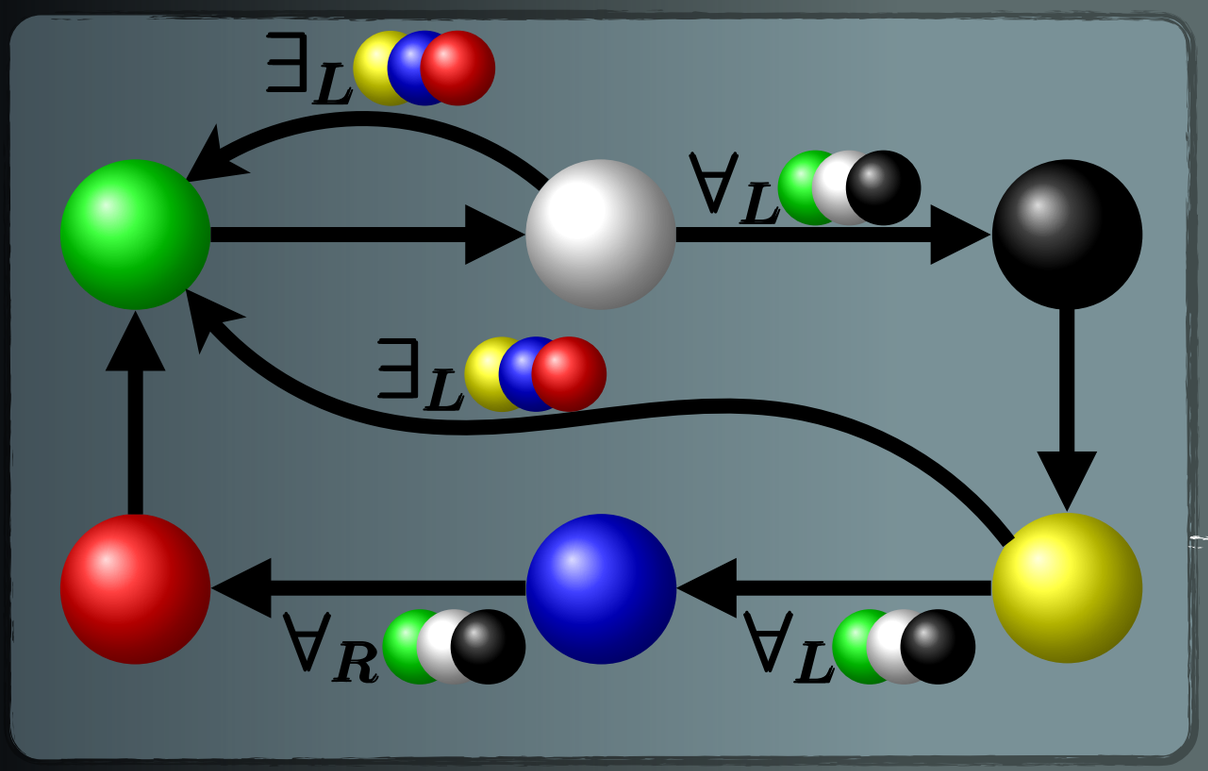


$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2(X)))$$





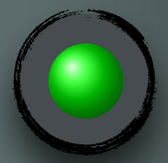


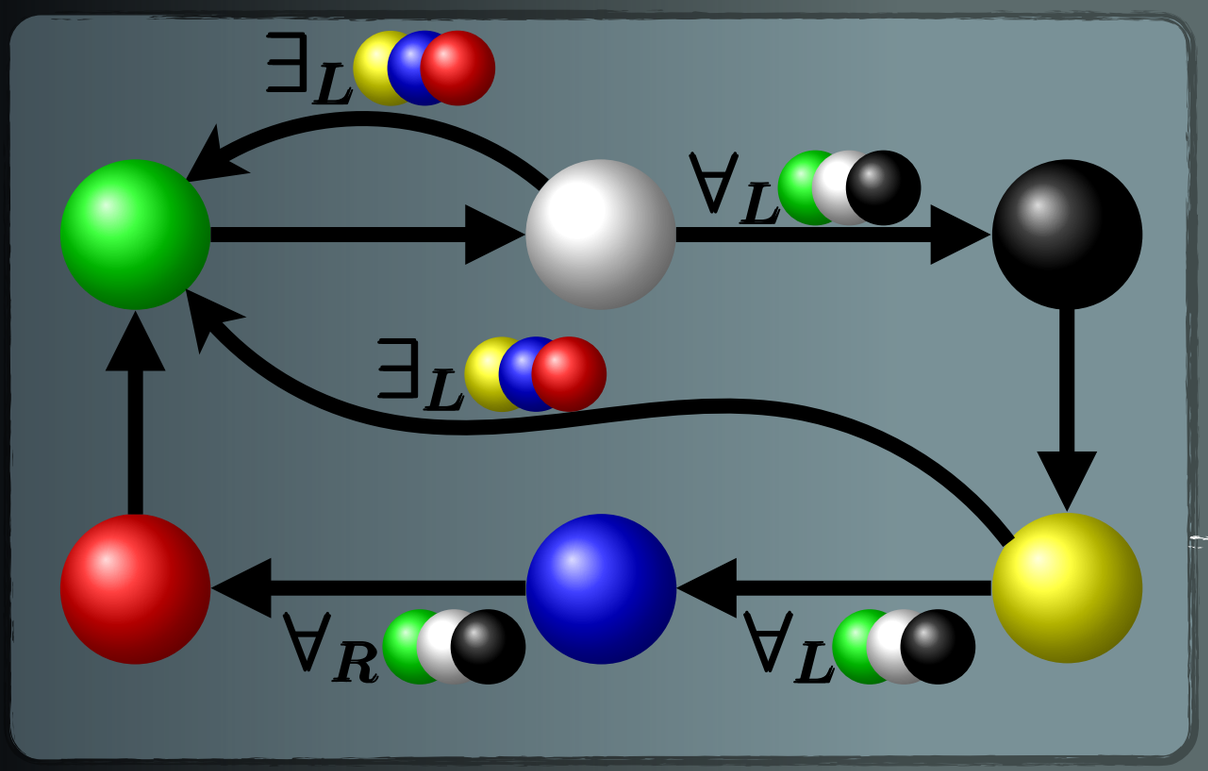


$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$





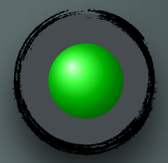


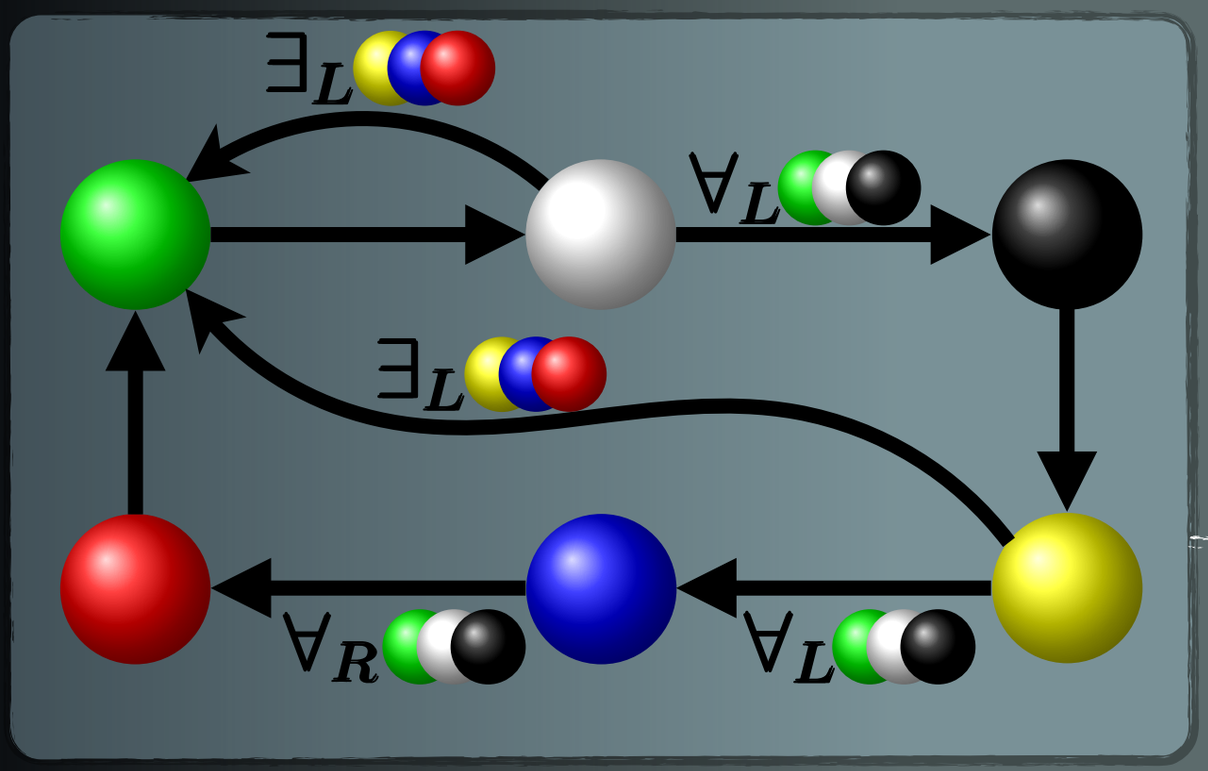


$$V_1 := \mu X.\ \alpha_1(I) \sqcup \alpha_1(post(\oint_1^2 (X)))$$





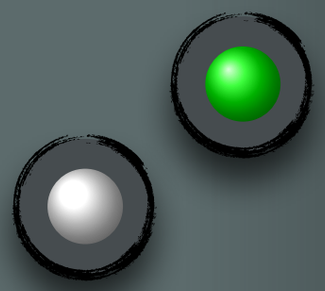


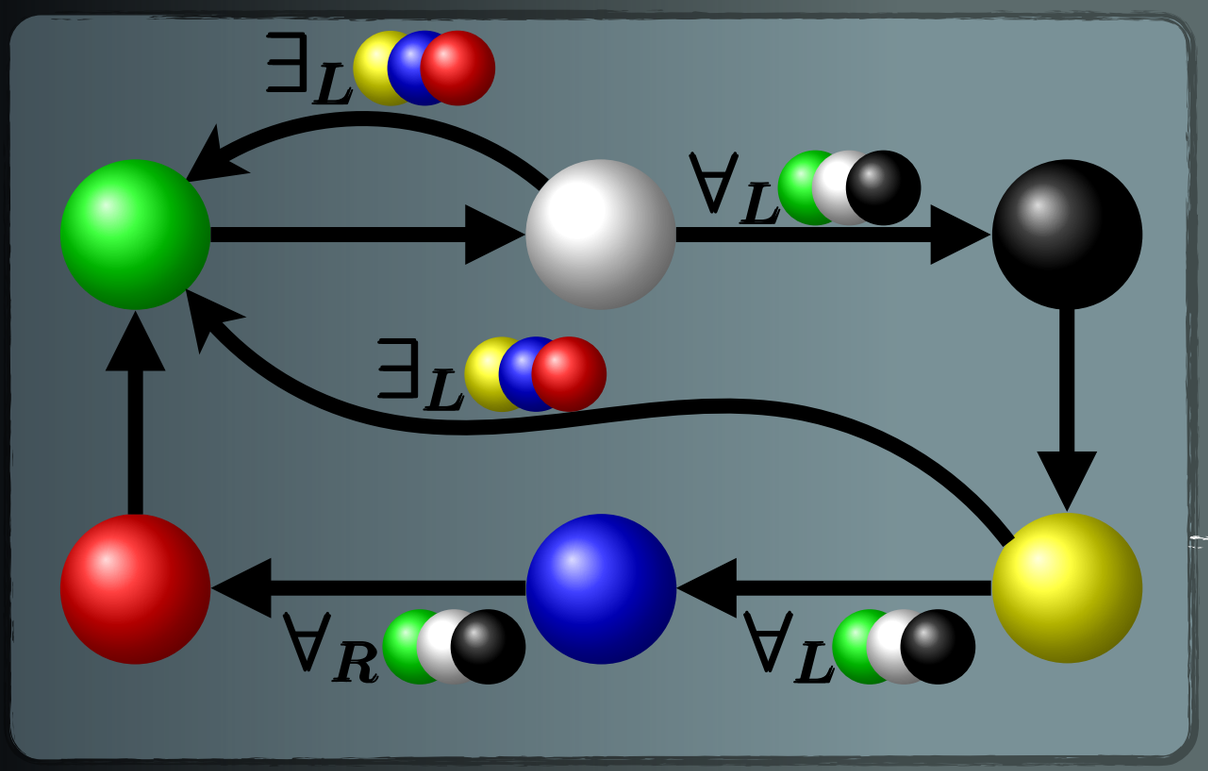


$$V_1 := \mu X.\ \alpha_1(I) \sqcup \alpha_1(post(\oint_1^2 (X)))$$

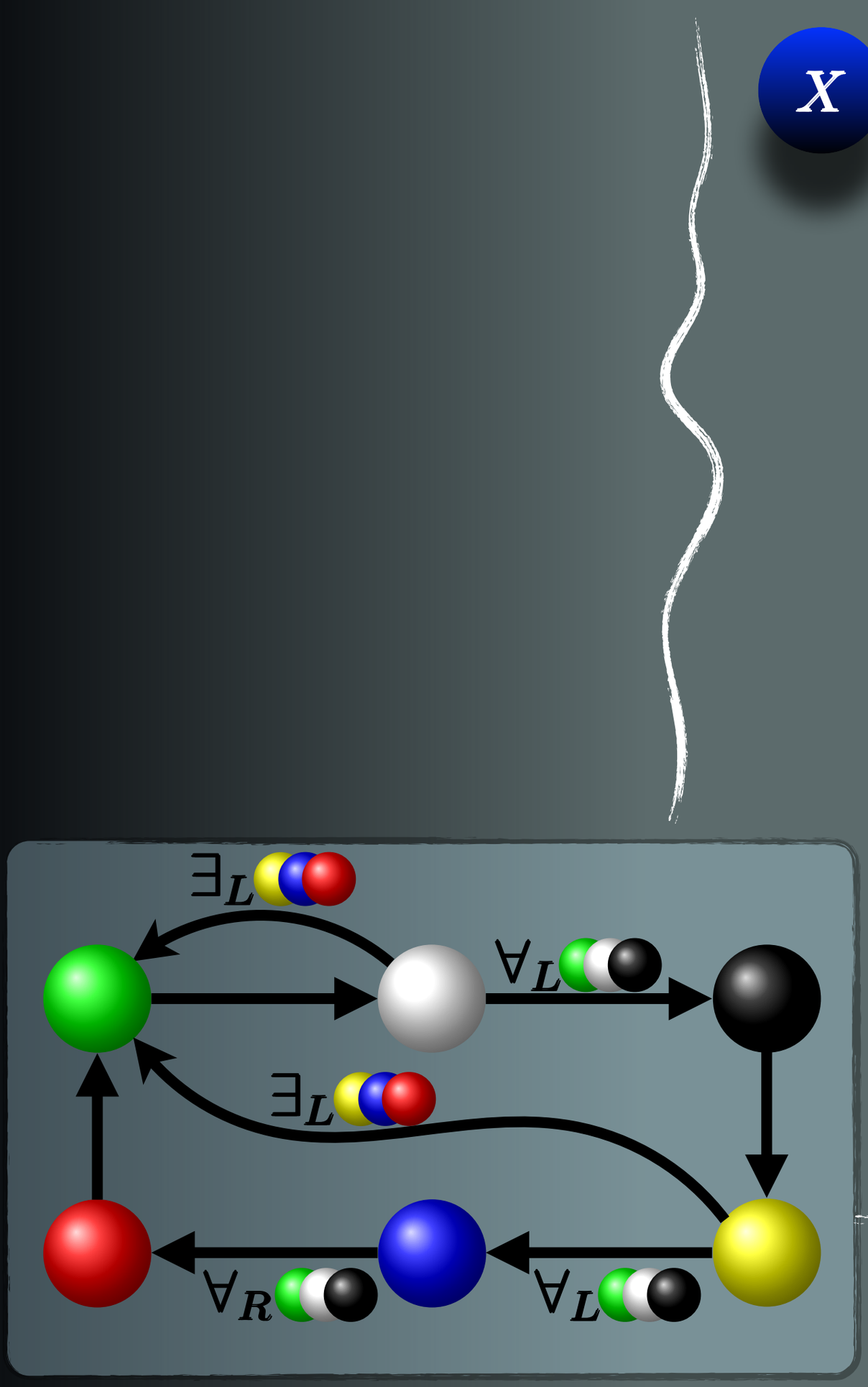




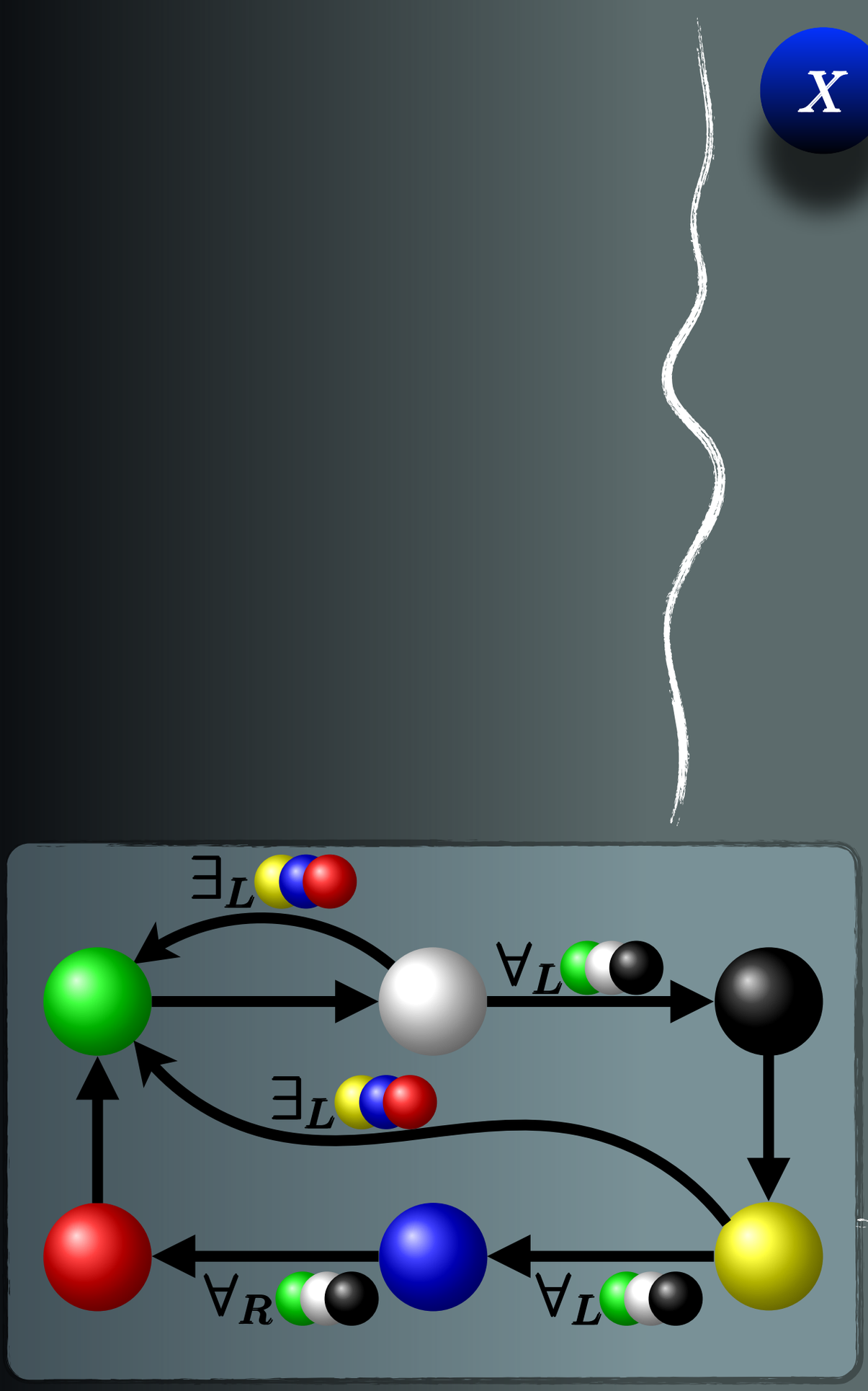




$$V_1 := \mu X.\ \alpha_1(I) \sqcup \alpha_1(post(\oint_1^2 (X)))$$

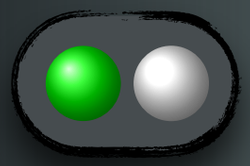


$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$

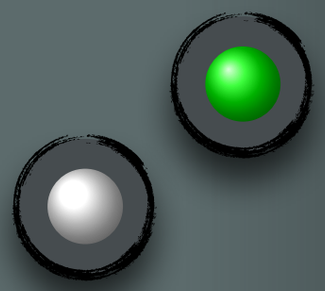


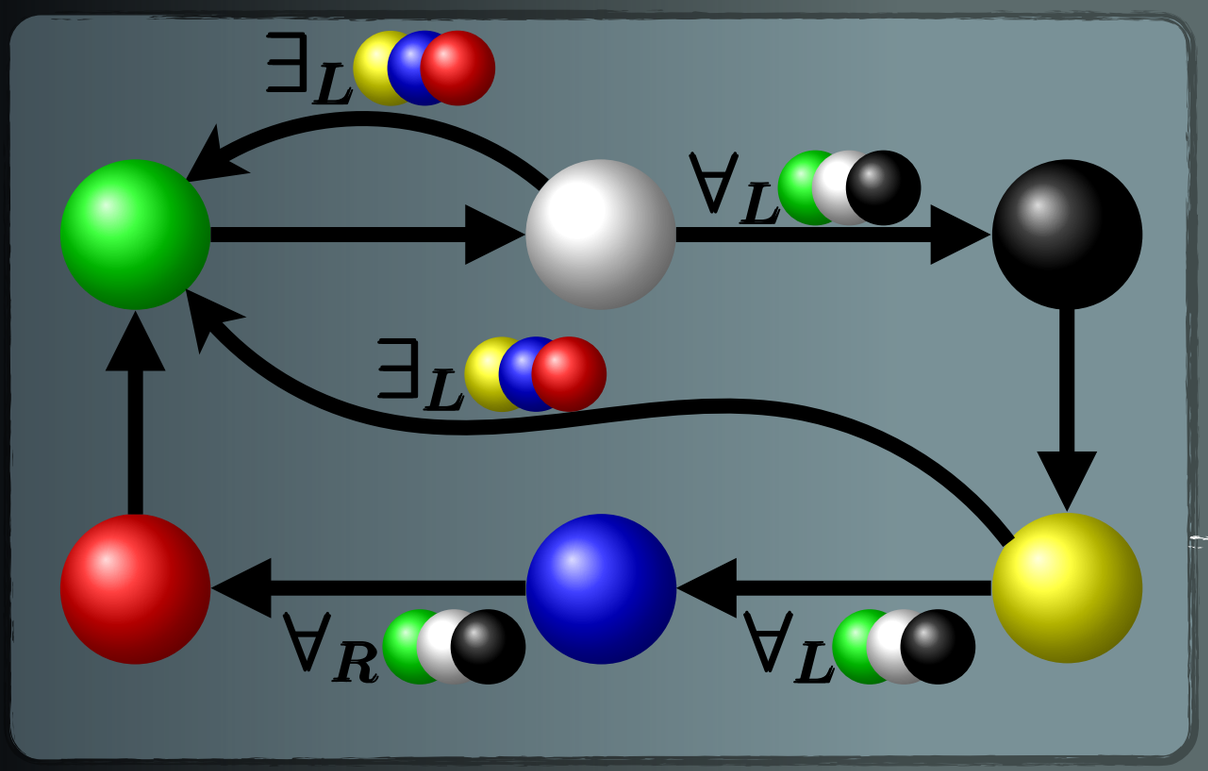
X

$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$

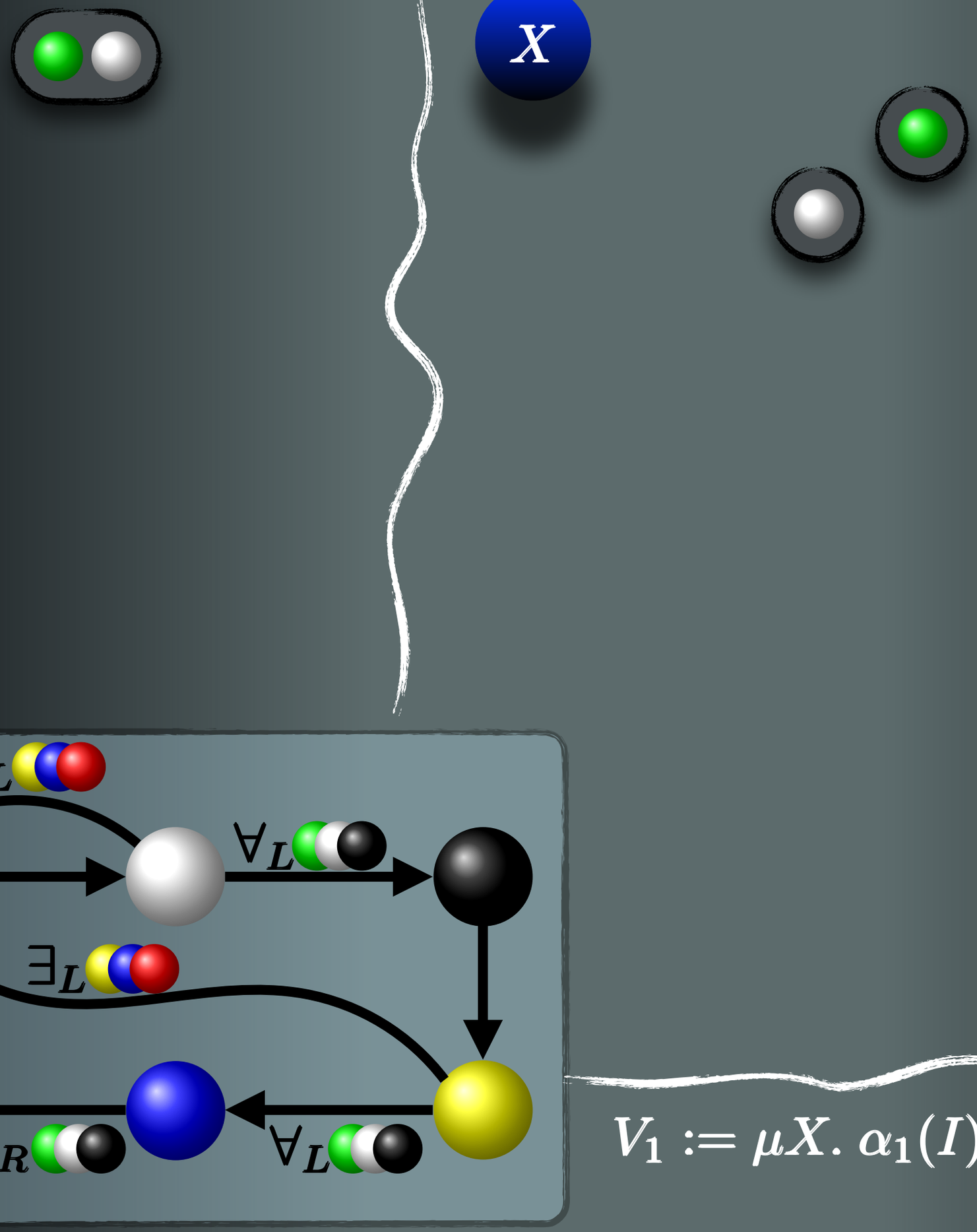




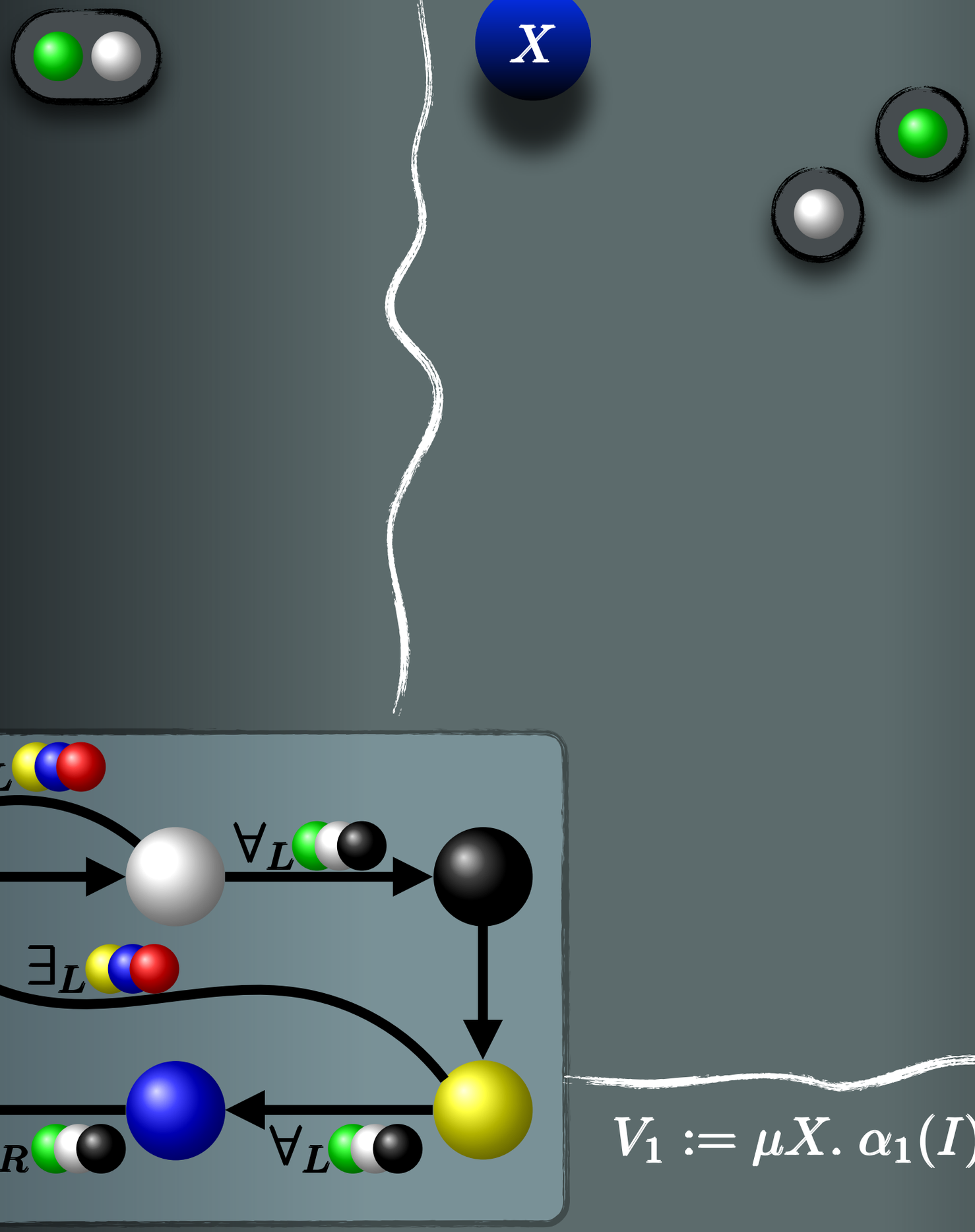




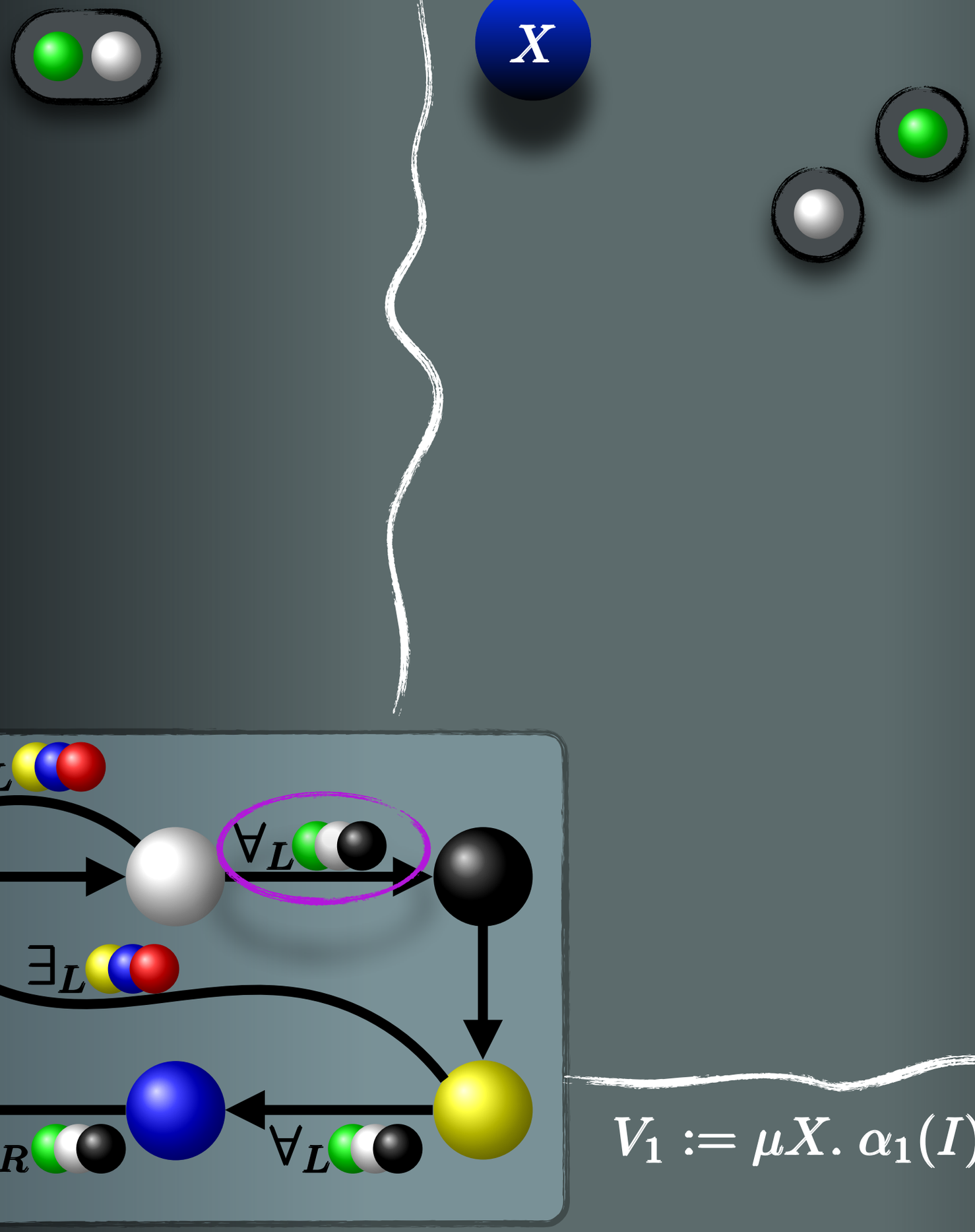
$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$



$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$



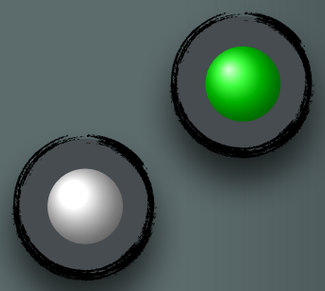
$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$

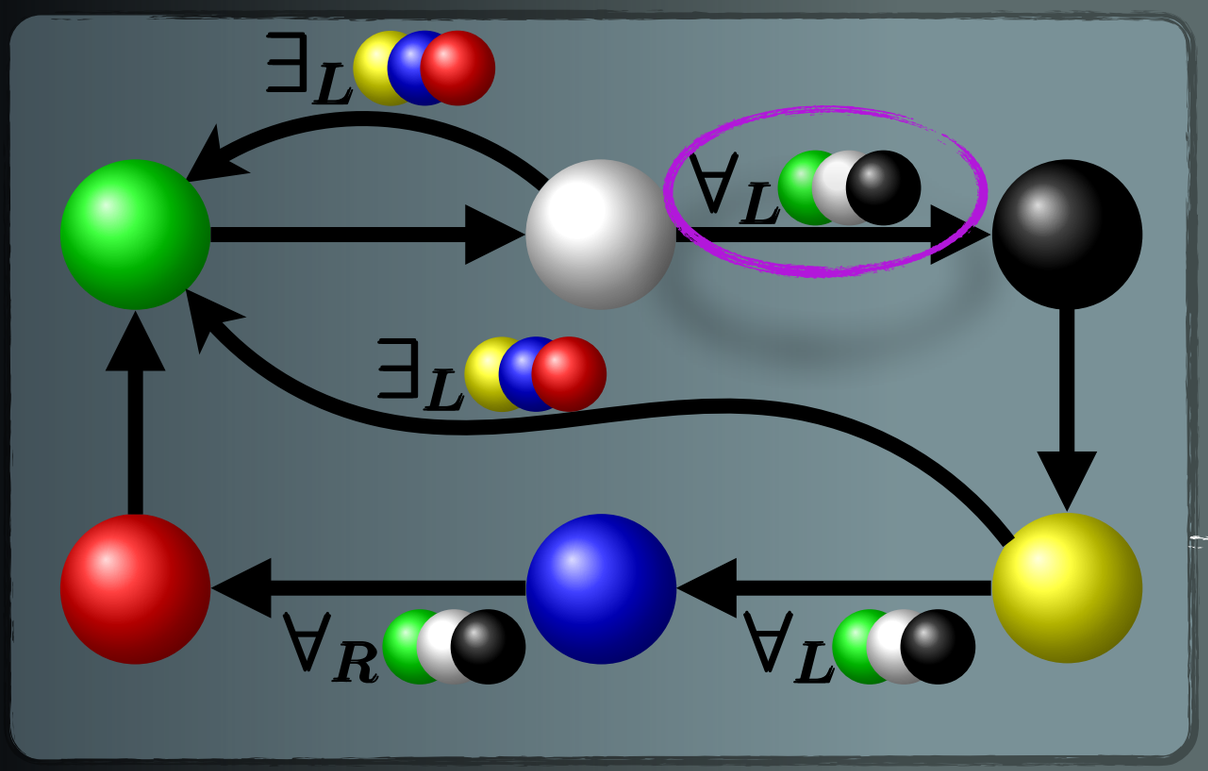


$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$





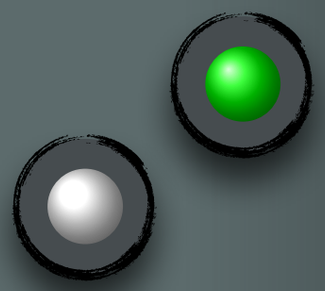


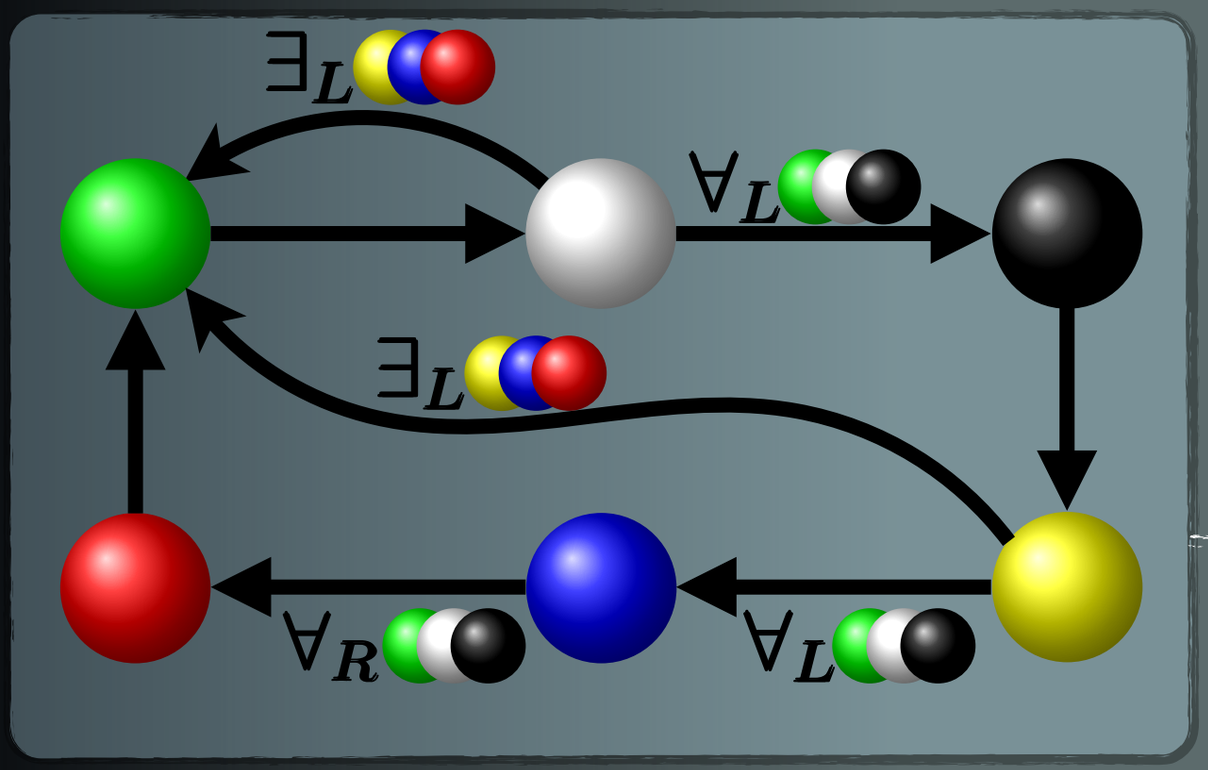


$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$





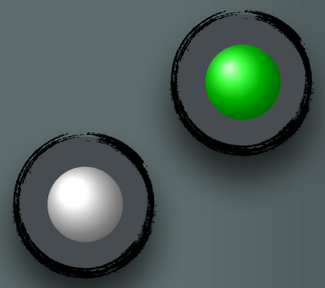


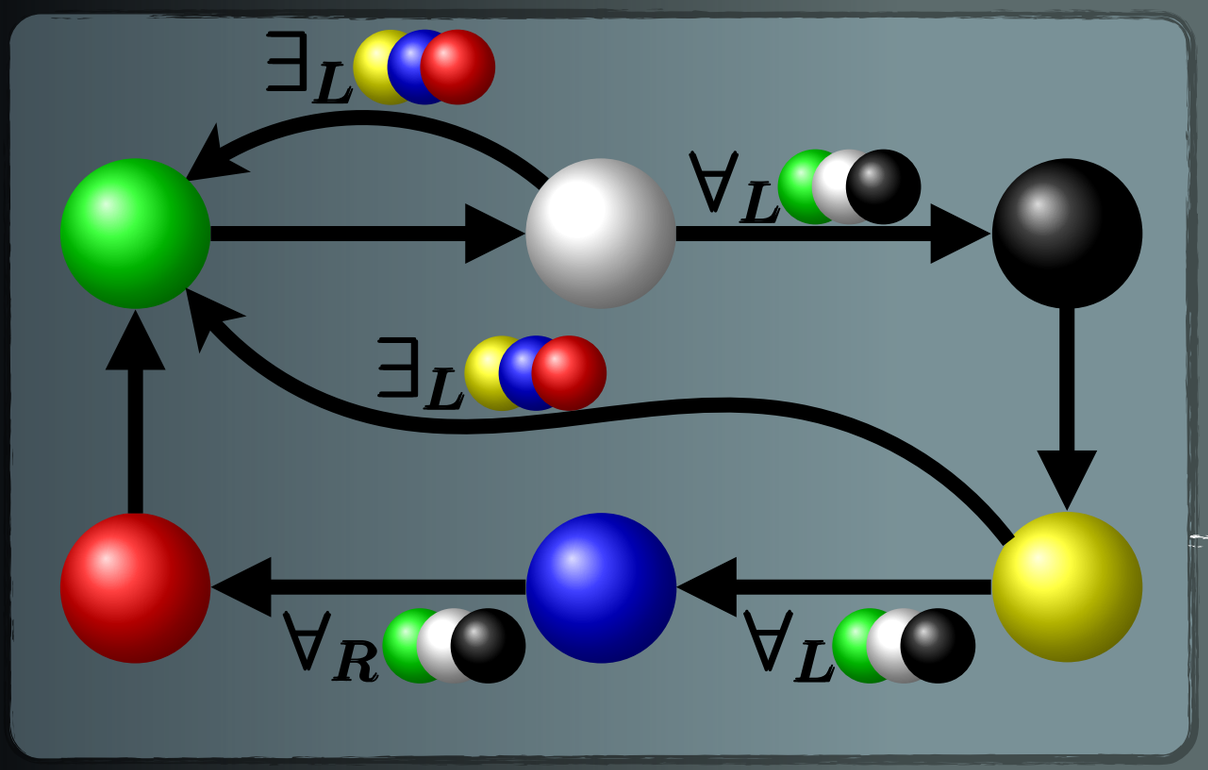


$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$





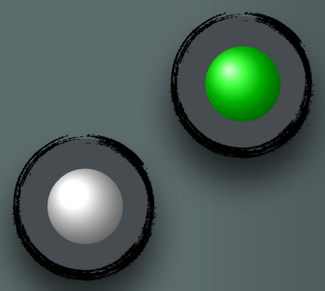


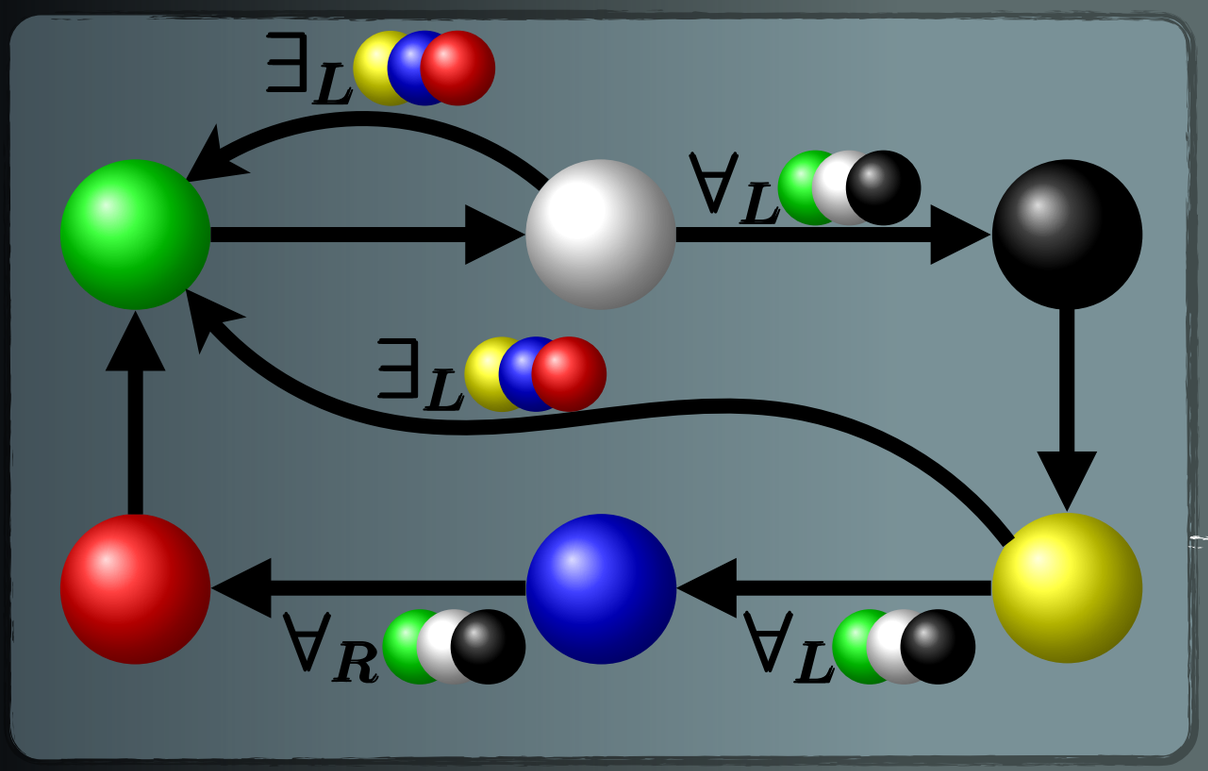


$$V_1 := \mu X.\ \alpha_1(I) \sqcup \alpha_1(post(\oint_1^2 (X)))$$





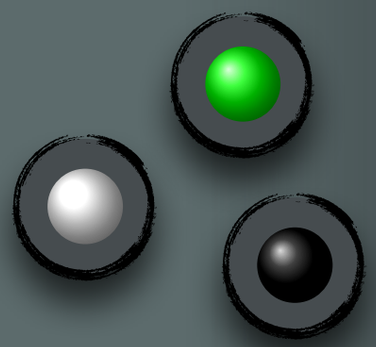


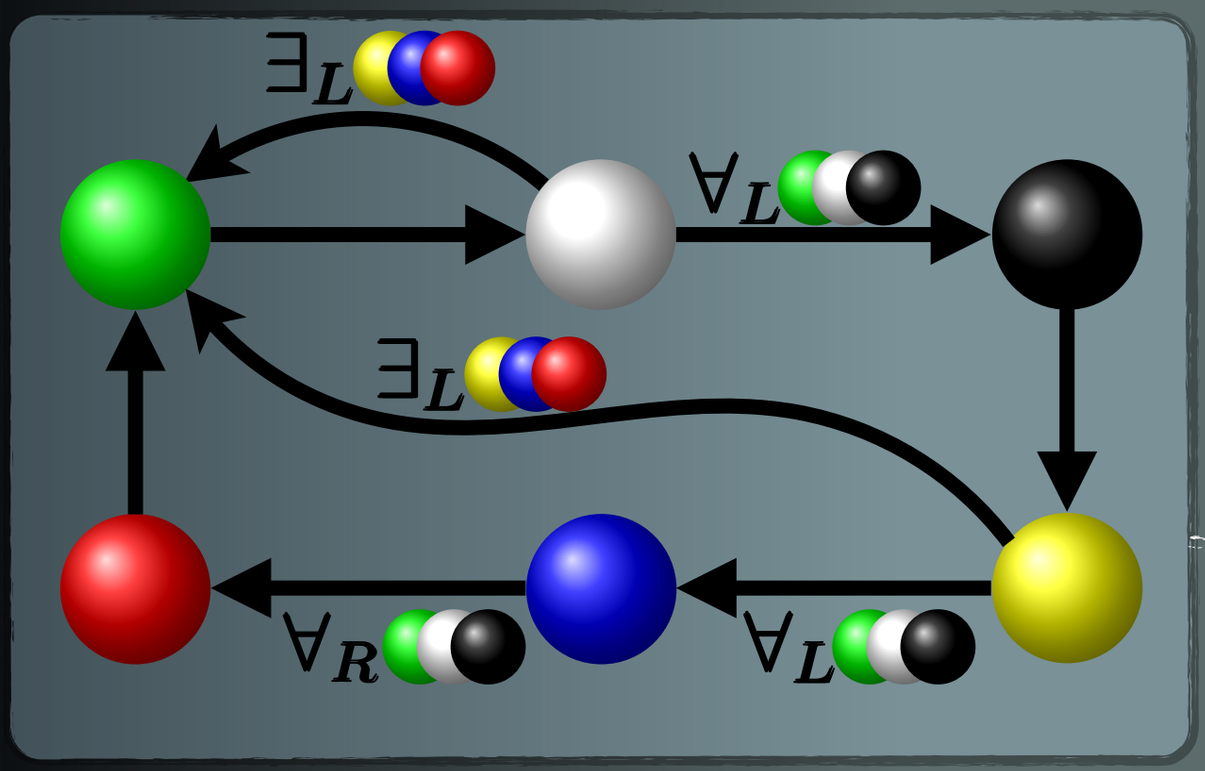


$$V_1 := \mu X.\ \alpha_1(I) \sqcup \alpha_1(post(\oint_1^2 (X)))$$





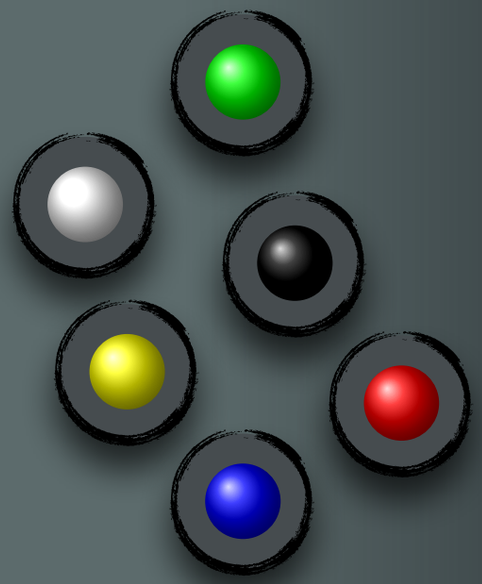


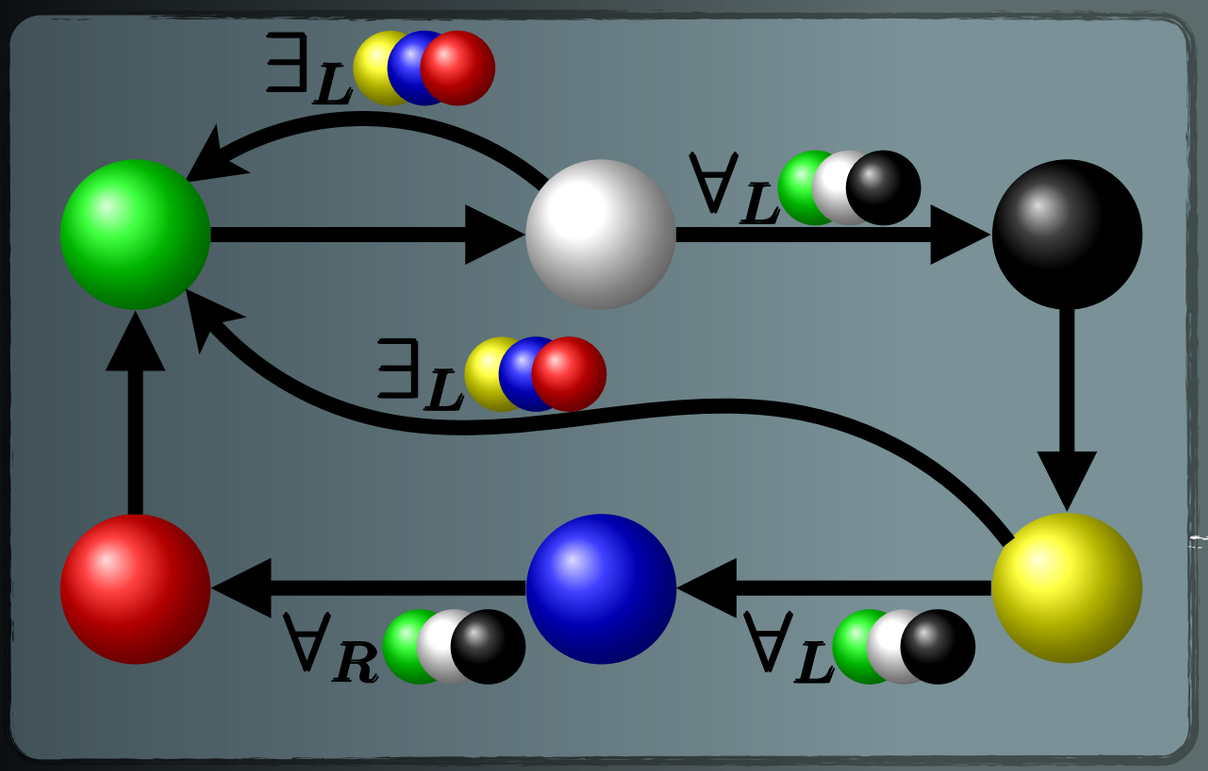


$$V_1 := \mu X.\ \alpha_1(I) \sqcup \alpha_1(post(\oint_1^2 (X)))$$



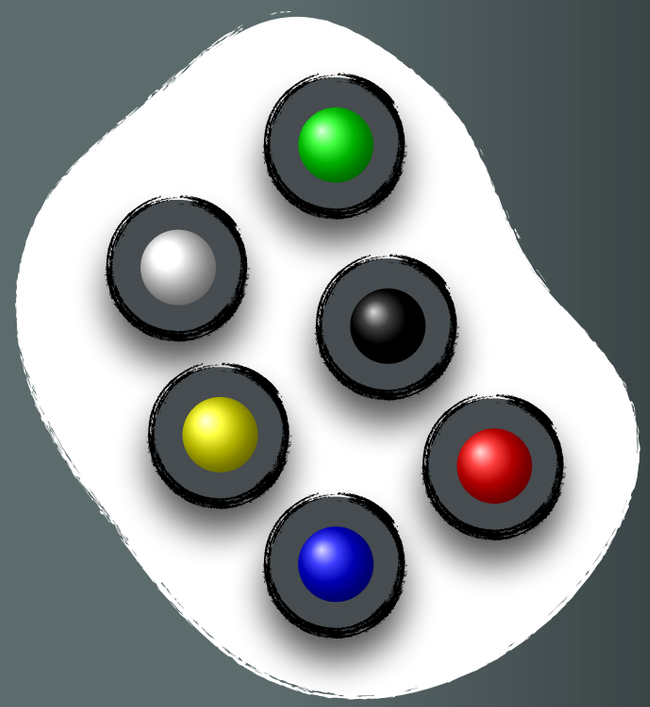


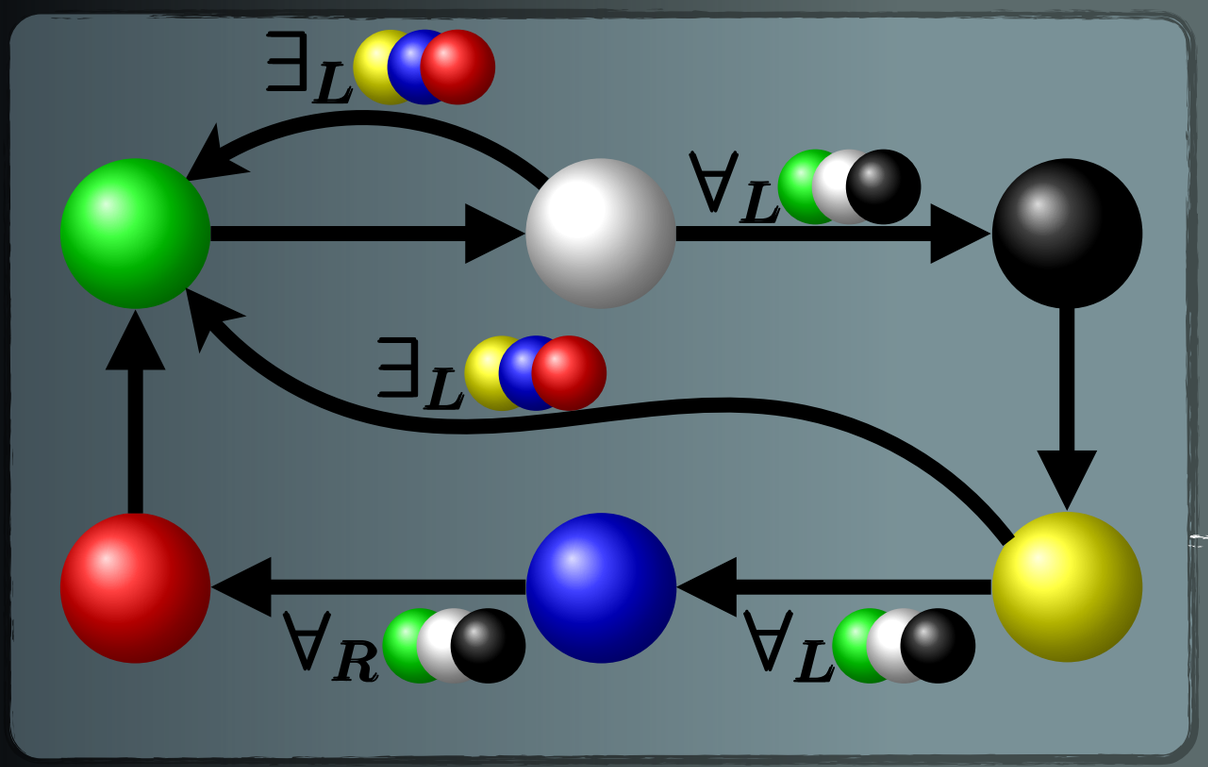




$$V_1 := \mu X.\ \alpha_1(I) \sqcup \alpha_1(post(\oint_1^2 (X)))$$

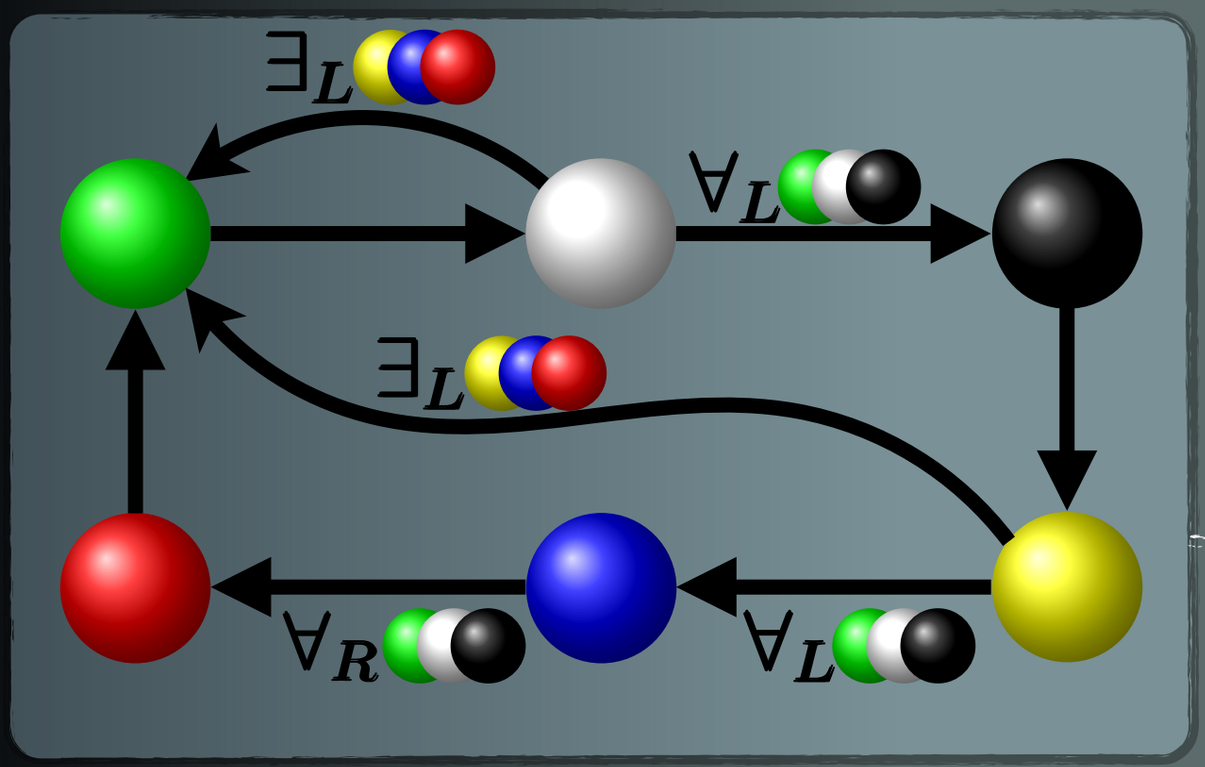






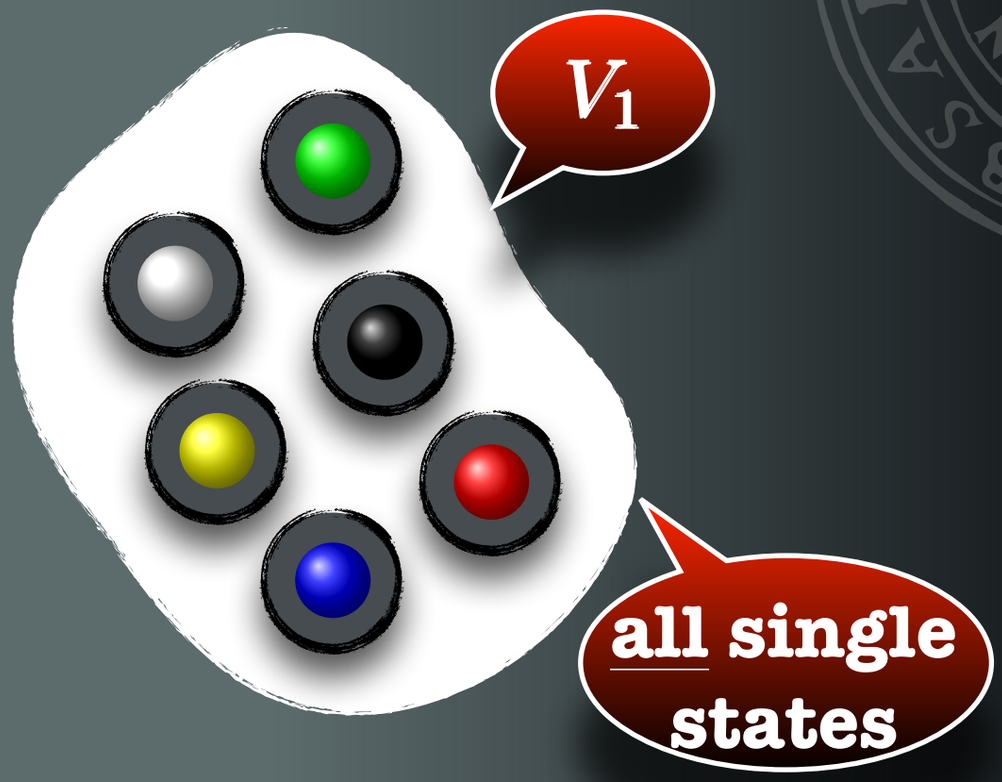
$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$

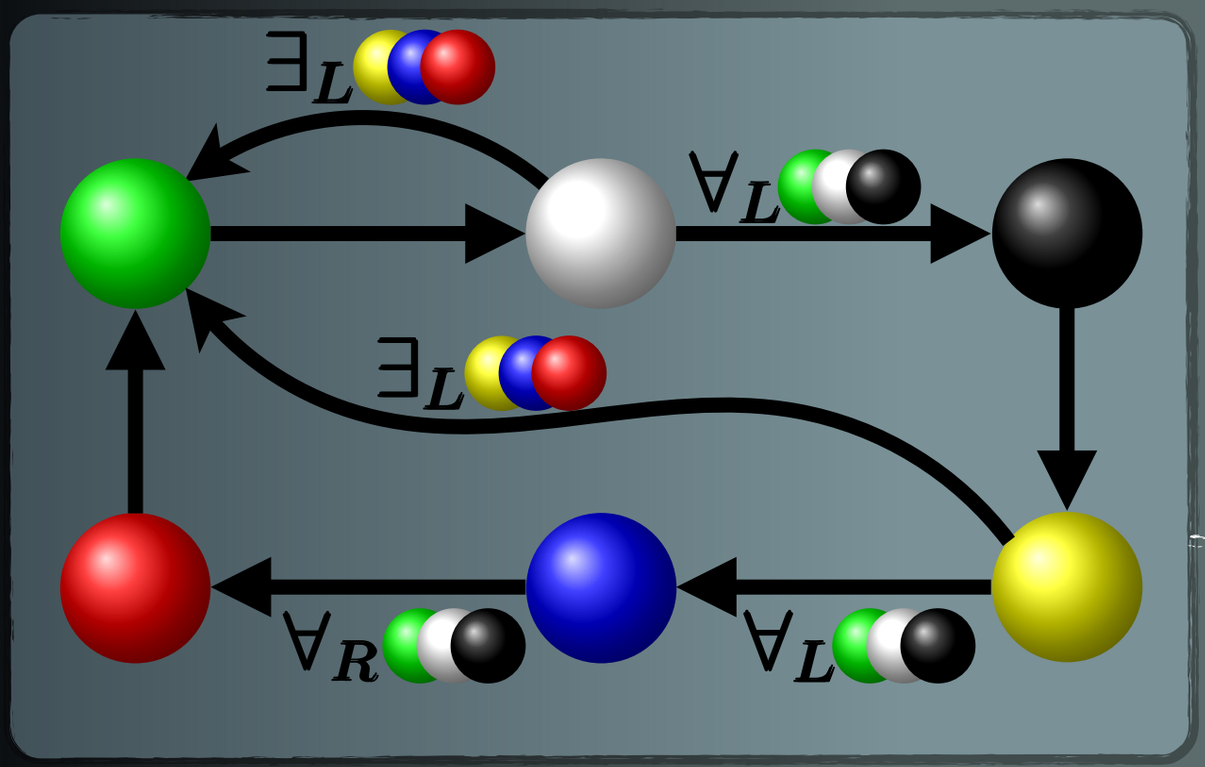




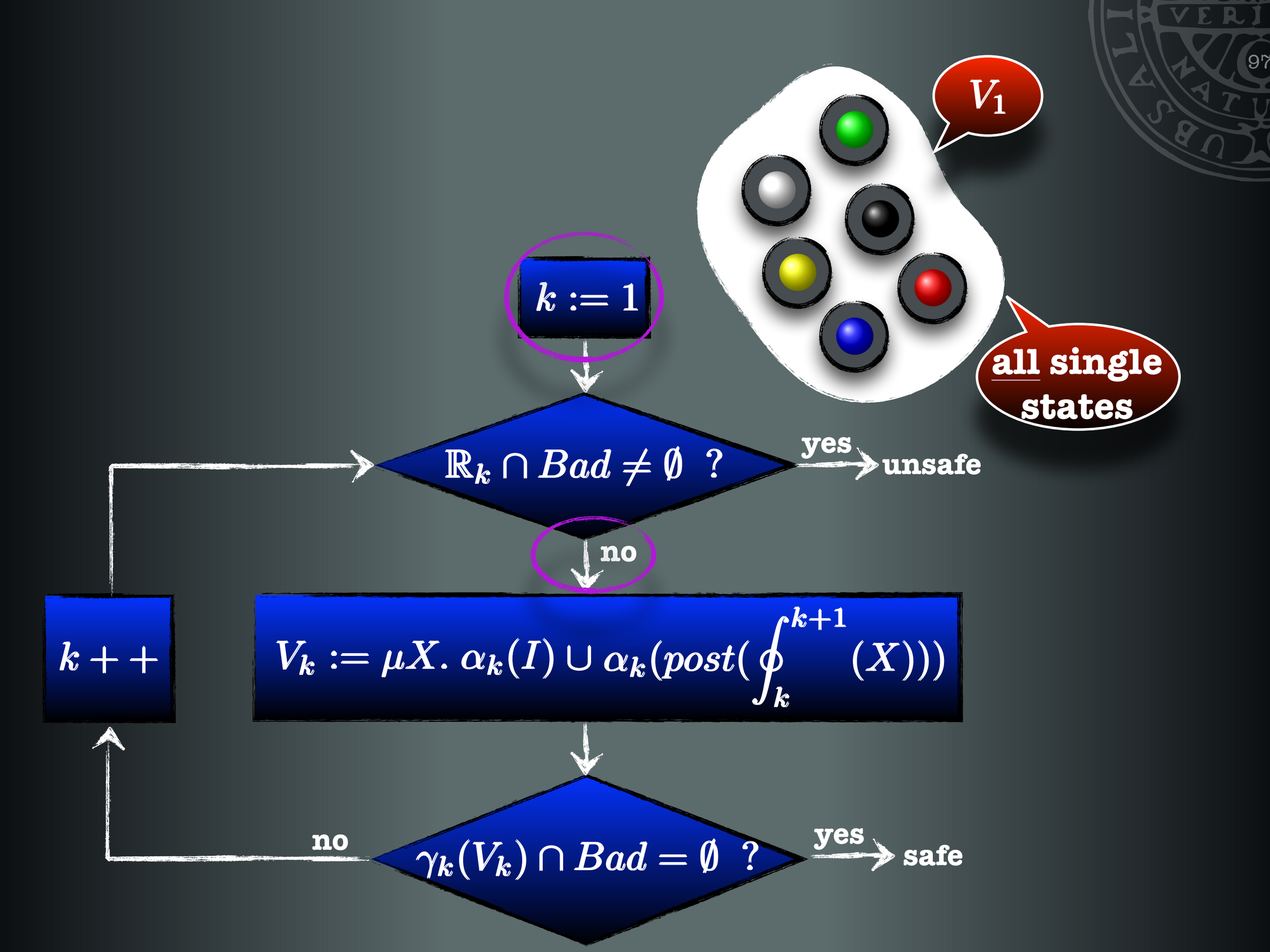
$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$

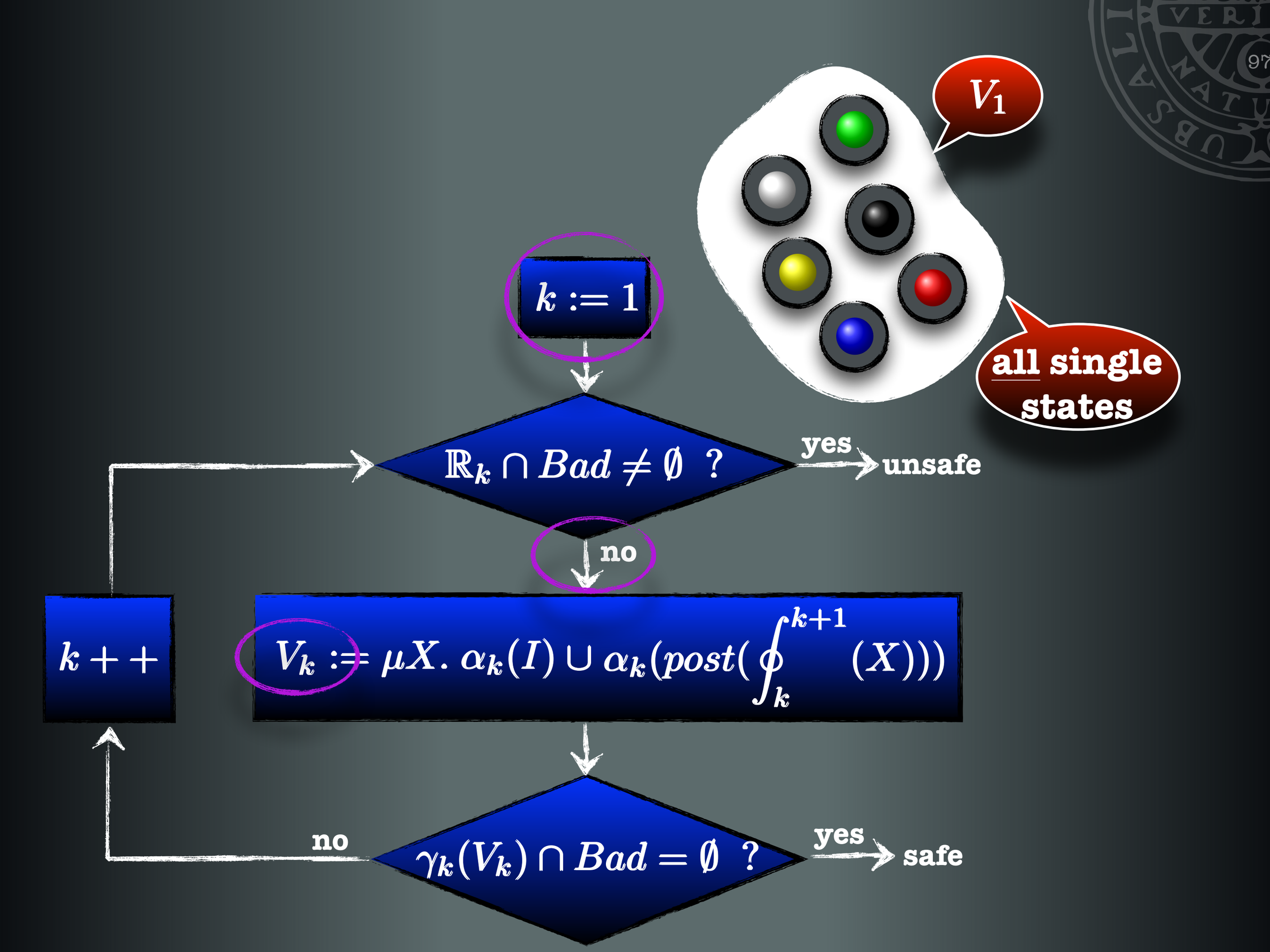
X

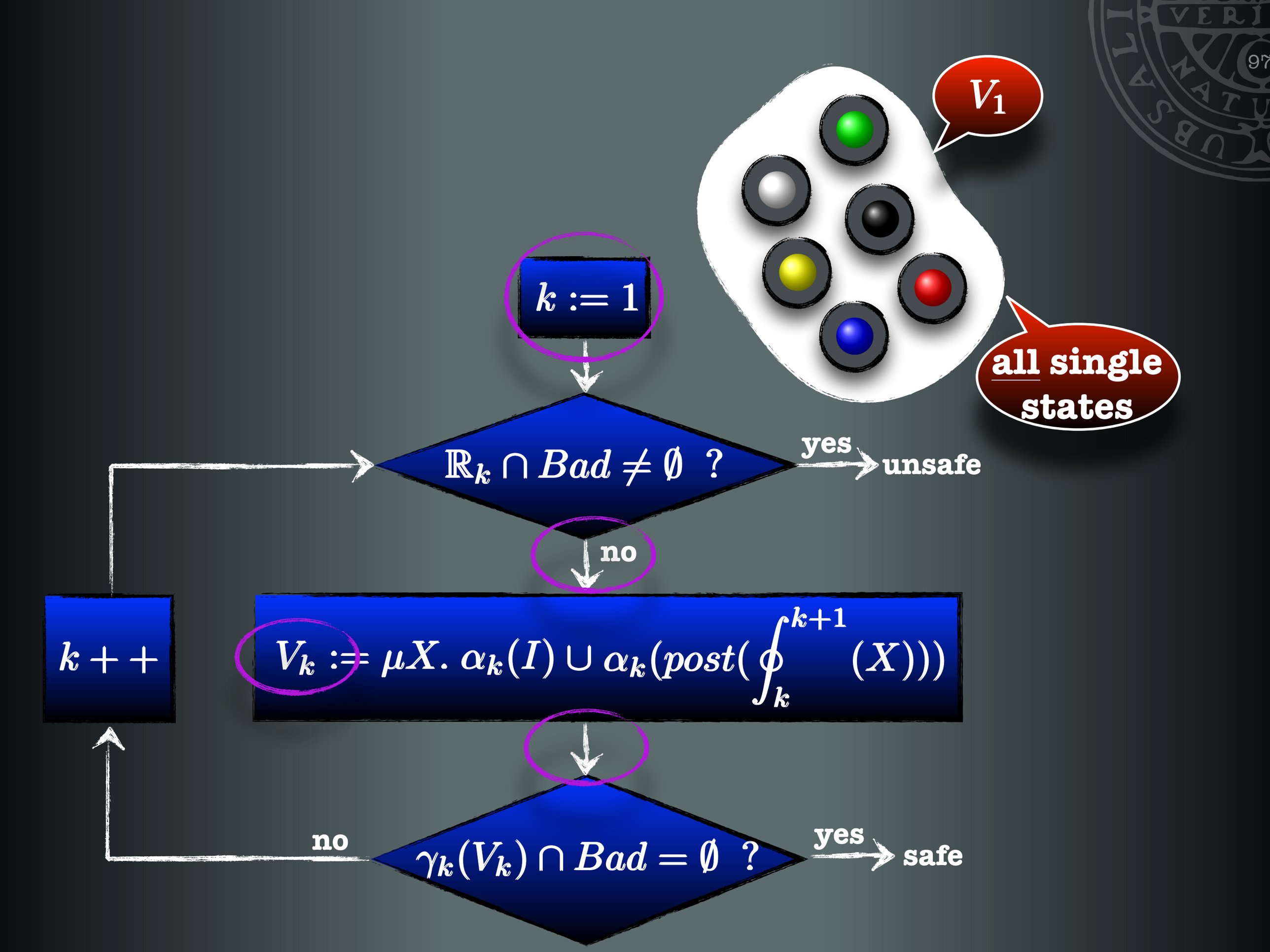


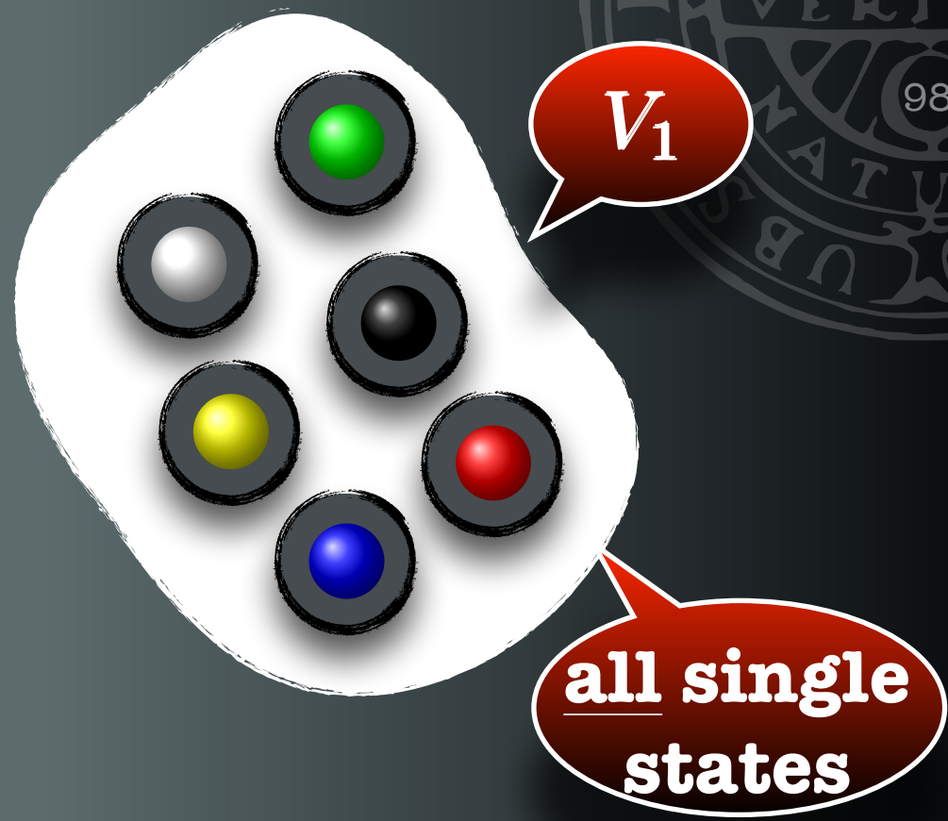


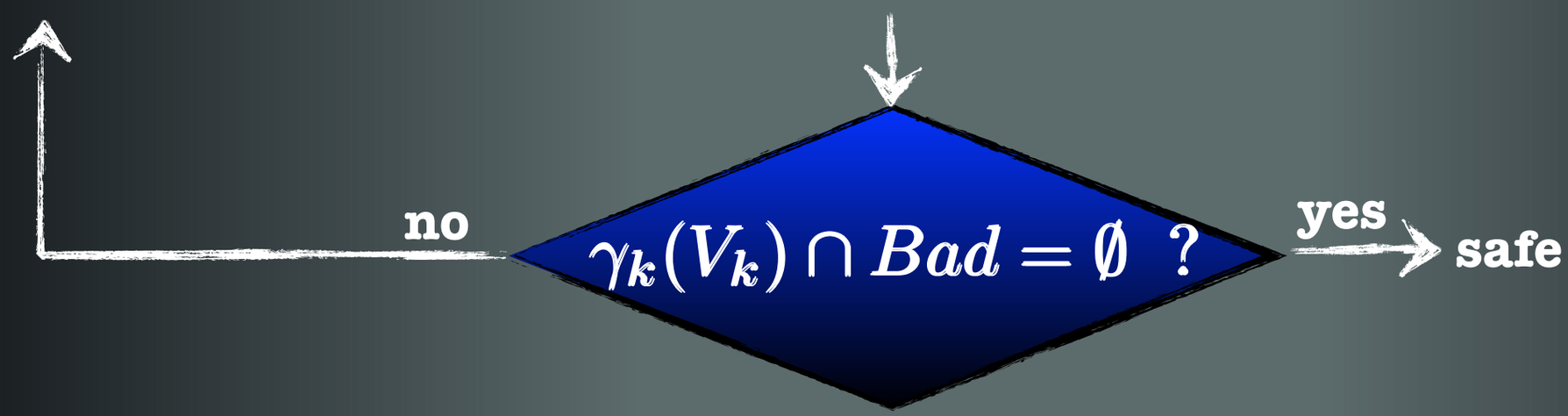
$$V_1 := \mu X.\ \alpha_1(I) \cup \alpha_1(post(\oint_1^2 (X)))$$

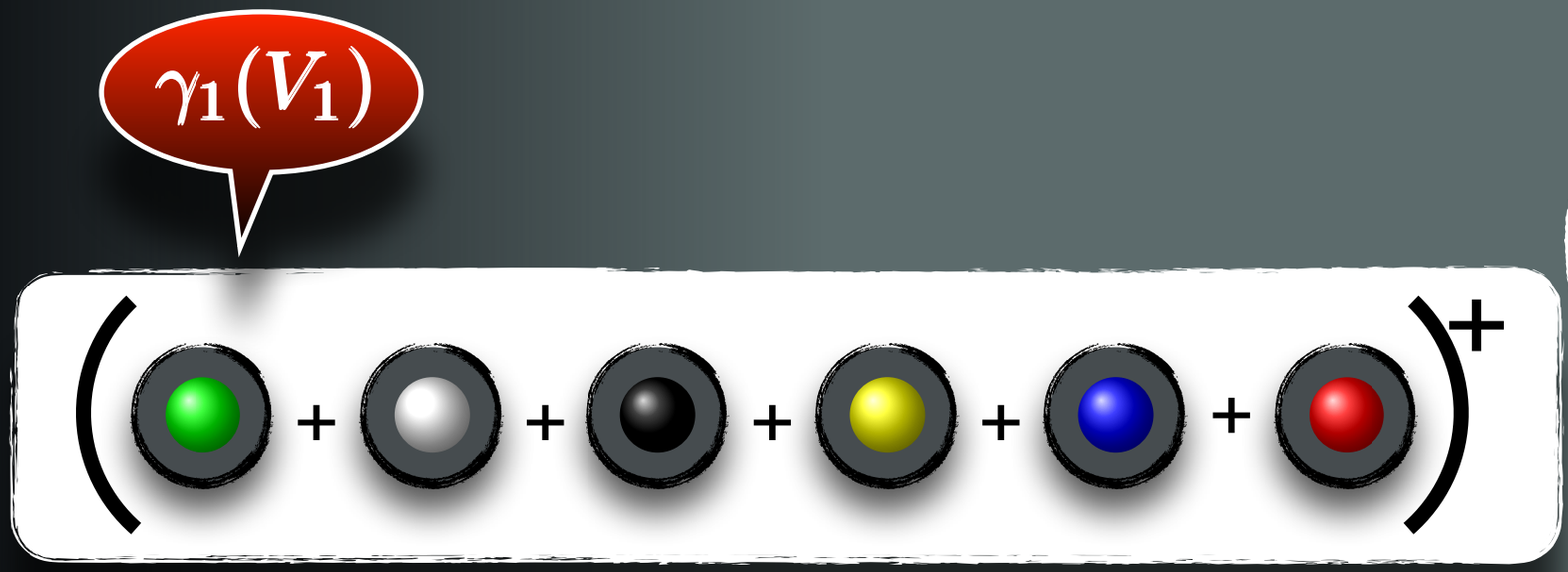

V_1
all single states
$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$k++$
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$
no
$\gamma_k(V_k) \cap Bad = \emptyset$?
yes
safe

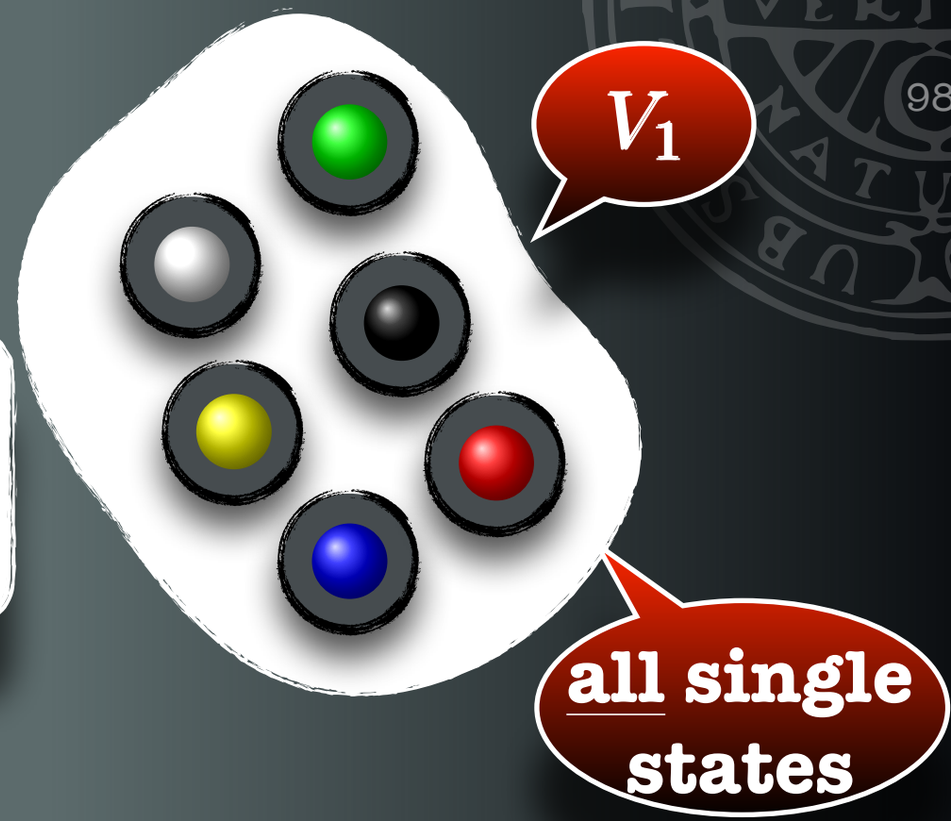

V_1
all single states
$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$
$k++$
$\gamma_k(V_k) \cap Bad = \emptyset$?
no
yes
safe

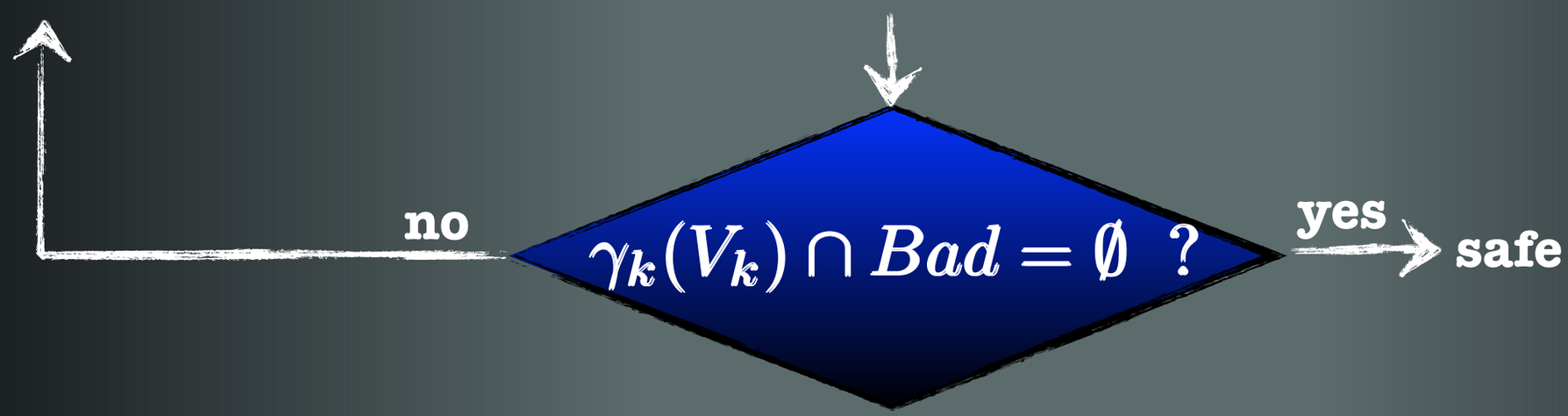

V_1
all single states
$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$k++$
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$
no
$\gamma_k(V_k) \cap Bad = \emptyset$?
yes
safe

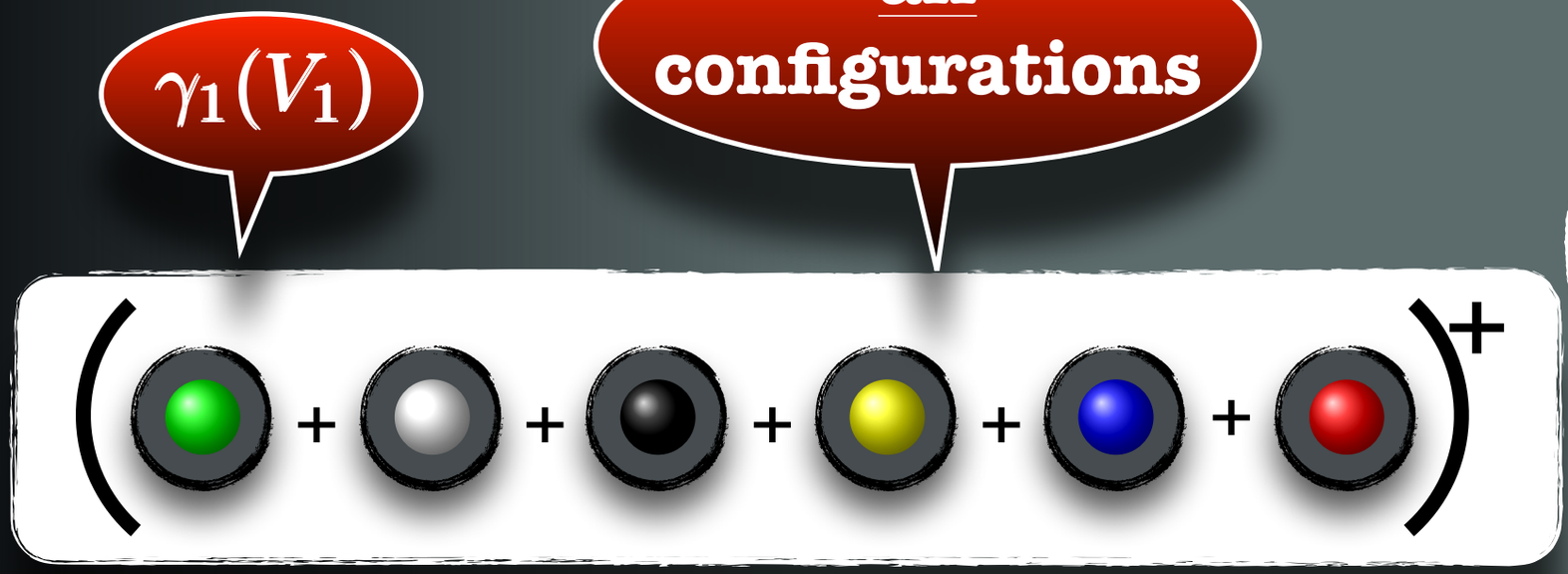

V_1
all single states

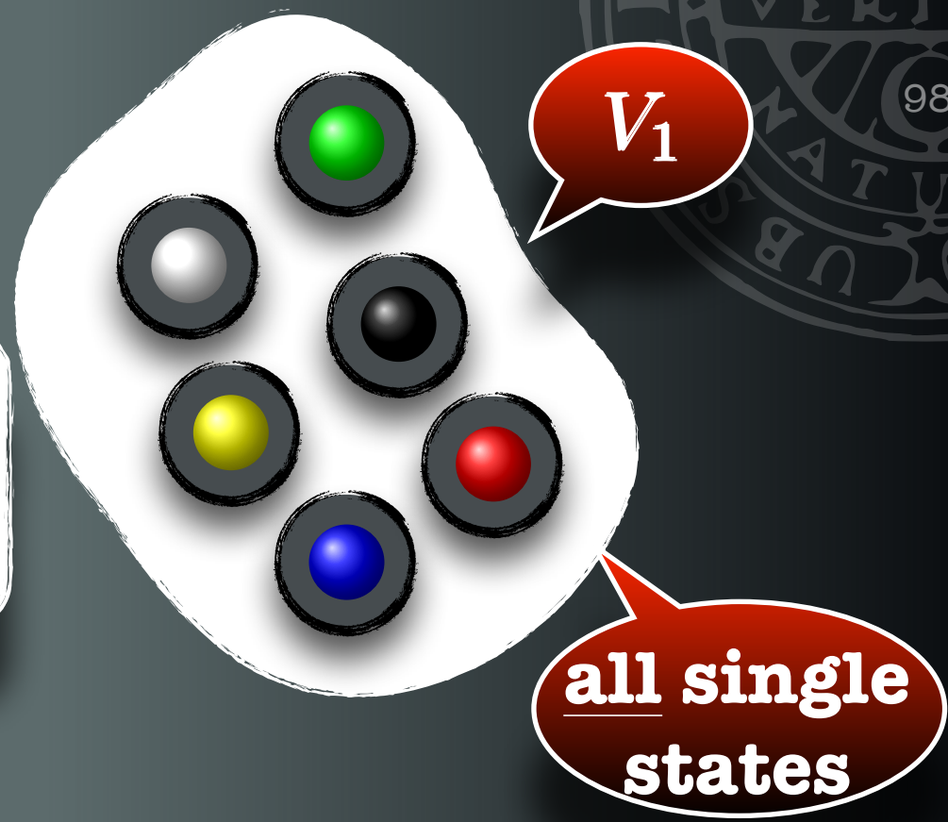

no
$\gamma_k(V_k) \cap Bad = \emptyset$?
yes
safe

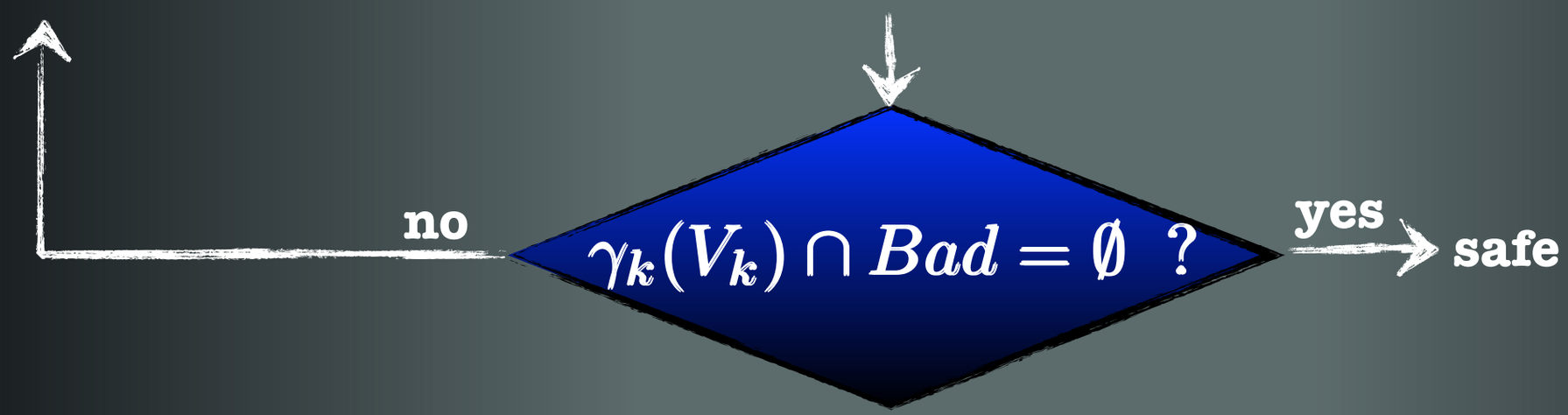

$\gamma_1(V_1)$
+
+
+
+
+
+


V_1
all single states


no
$\gamma_k(V_k) \cap Bad = \emptyset$?
yes
safe


$\gamma_1(V_1)$
all configurations
+
+
+
+
+
+


V_1
all single states


no
$\gamma_k(V_k) \cap Bad = \emptyset$?
yes
safe

$\gamma_1(V_1)$

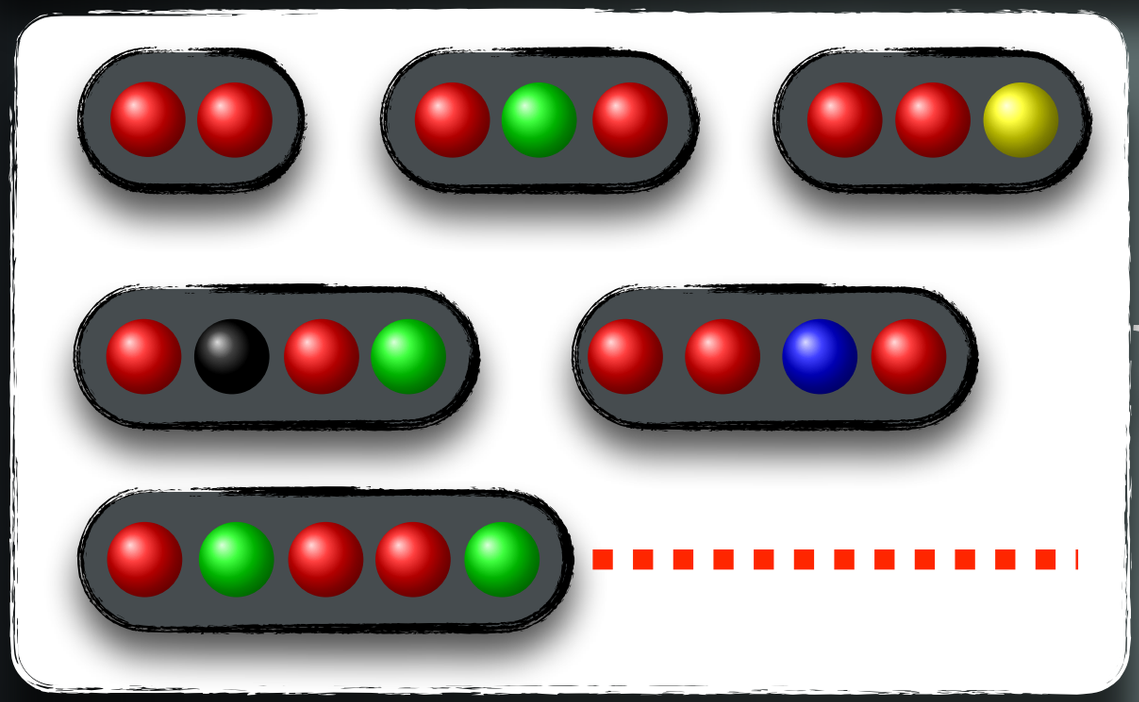
all configurations

$$\left(\bigcirc + \bigcirc + \bigcirc + \bigcirc + \bigcirc + \bigcirc \right)^{+}$$

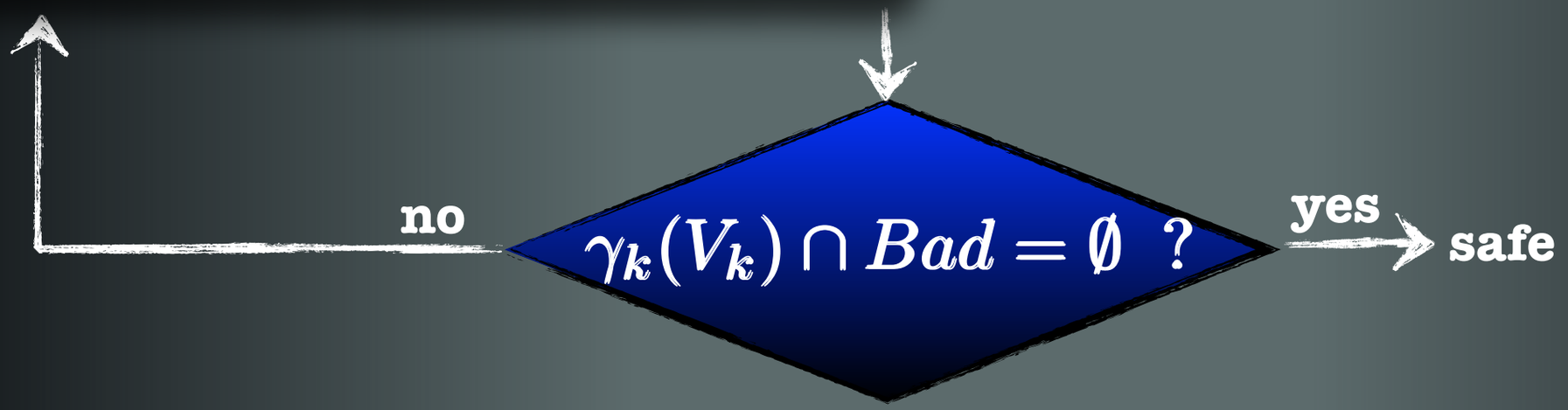


V_1

all single states



Bad Configurations



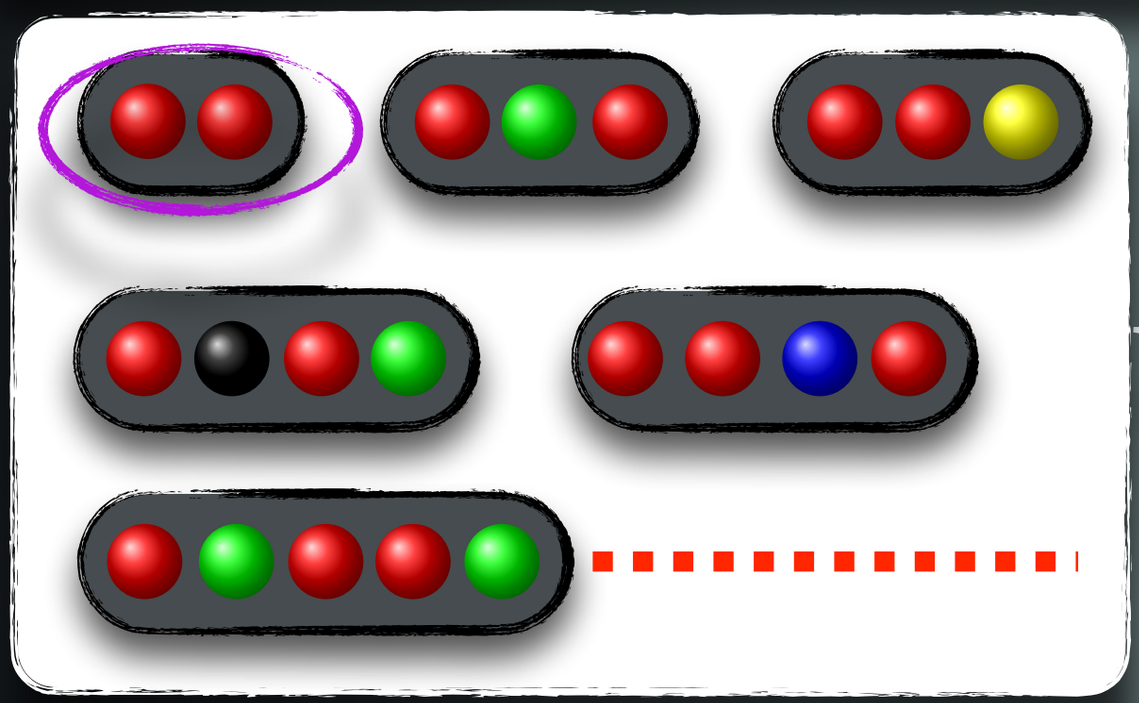
$\gamma_1(V_1)$

all configurations

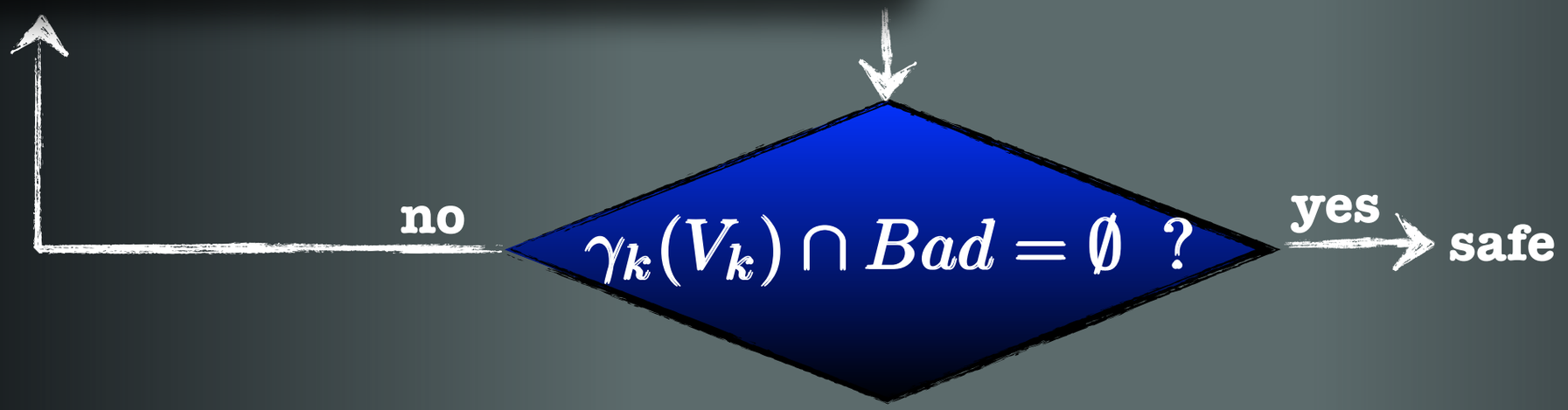
$$\left(\bigcirc + \bigcirc + \bigcirc + \bigcirc + \bigcirc + \bigcirc \right)^{+}$$

V_1

all single states



Bad Configurations



$\gamma_1(V_1)$

all configurations

$$\left(\quad + \quad + \quad + \quad + \quad + \quad \right)^{+}$$

V_1

all single states

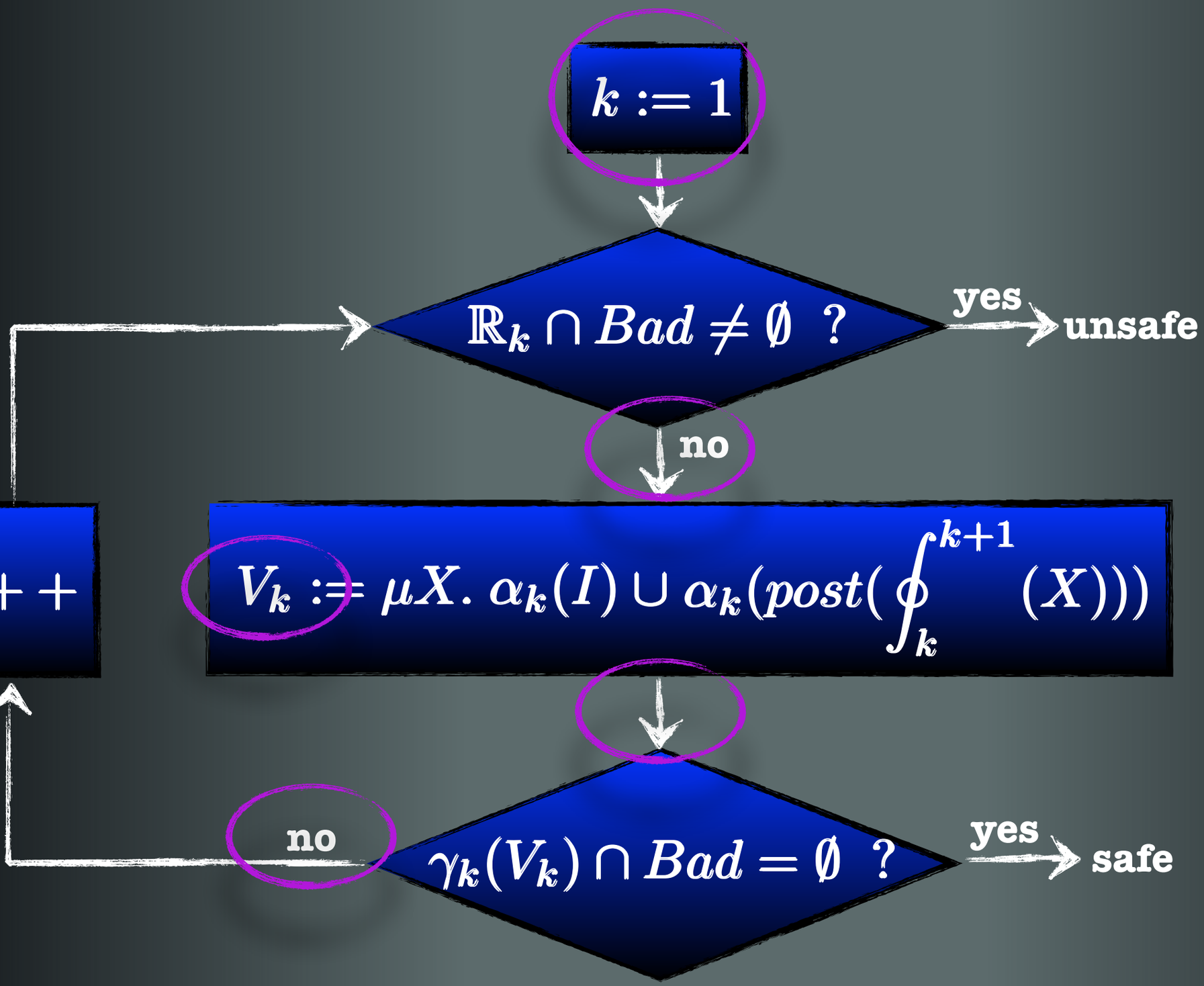
Bad Configurations

no

$\gamma_k(V_k) \cap Bad = \emptyset \ ?$

yes

safe


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$k++$
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$
no
$\gamma_k(V_k) \cap Bad = \emptyset$?
yes
safe

increase precision

$k := 1$

$\mathbb{R}_k \cap Bad \neq \emptyset$?

yes

unsafe

no

$k++$

$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$

no

$\gamma_k(V_k) \cap Bad = \emptyset$?

yes

safe

increase precision

$k := 1$

$k = 2$

$\mathbb{R}_k \cap Bad \neq \emptyset$?

yes

unsafe

no

$k++$

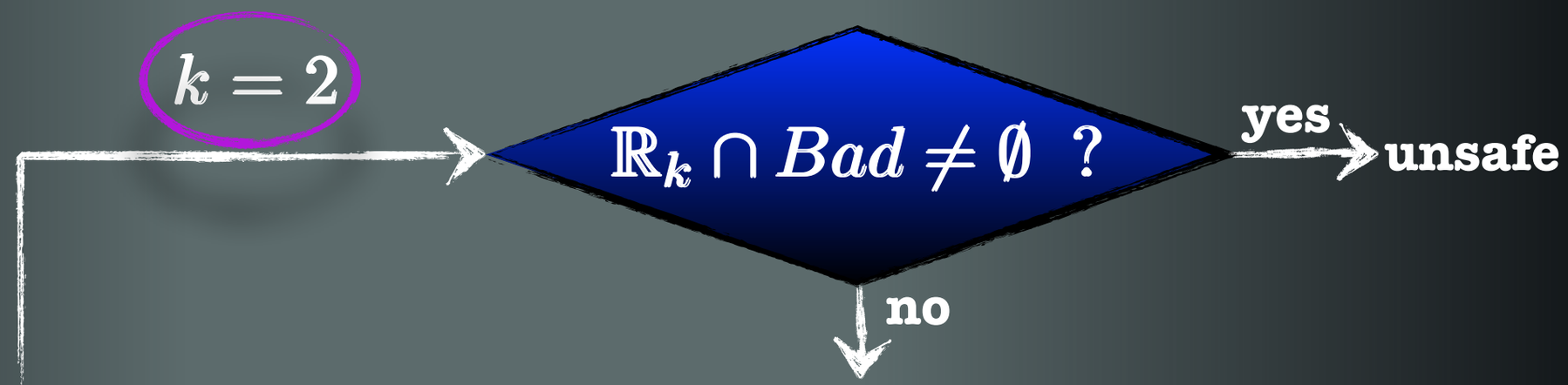
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$

no

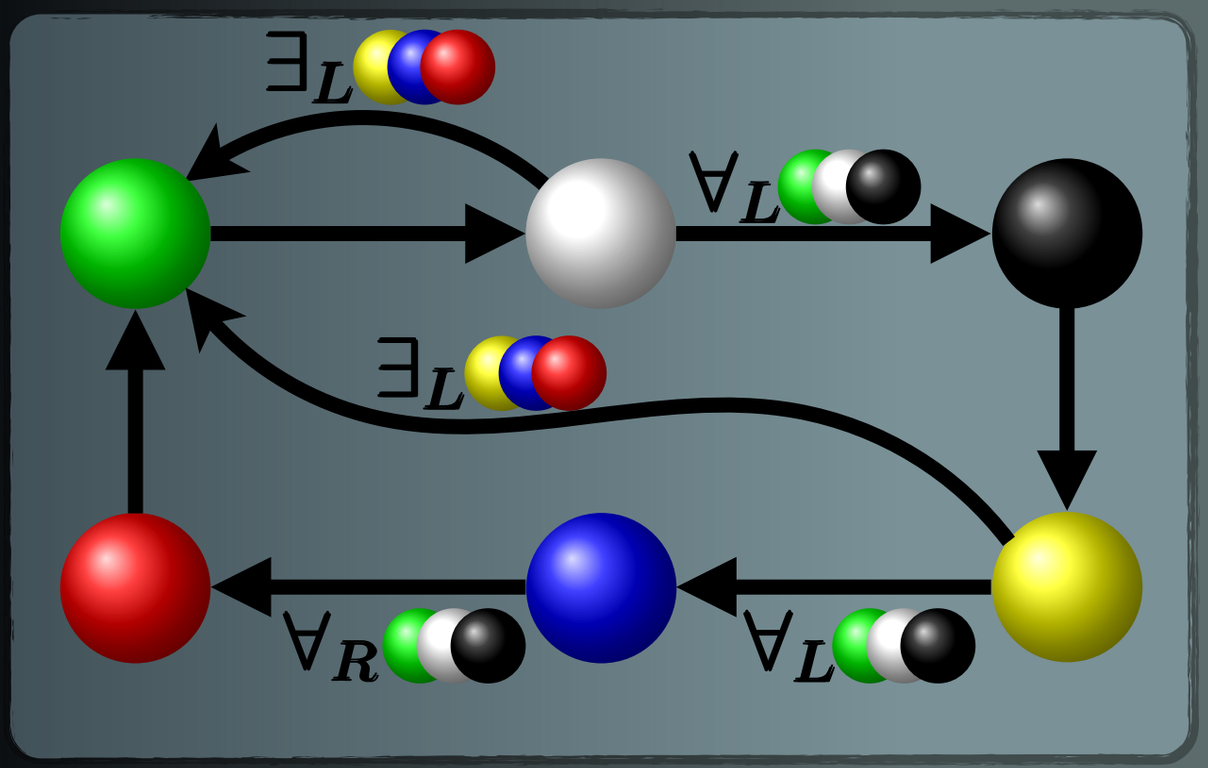
$\gamma_k(V_k) \cap Bad = \emptyset$?

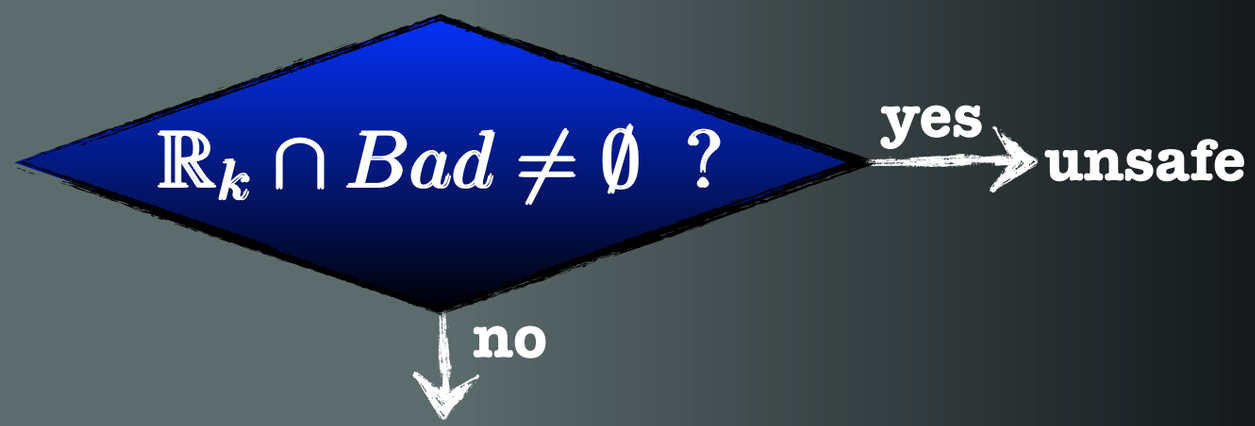
yes

safe


$k = 2$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

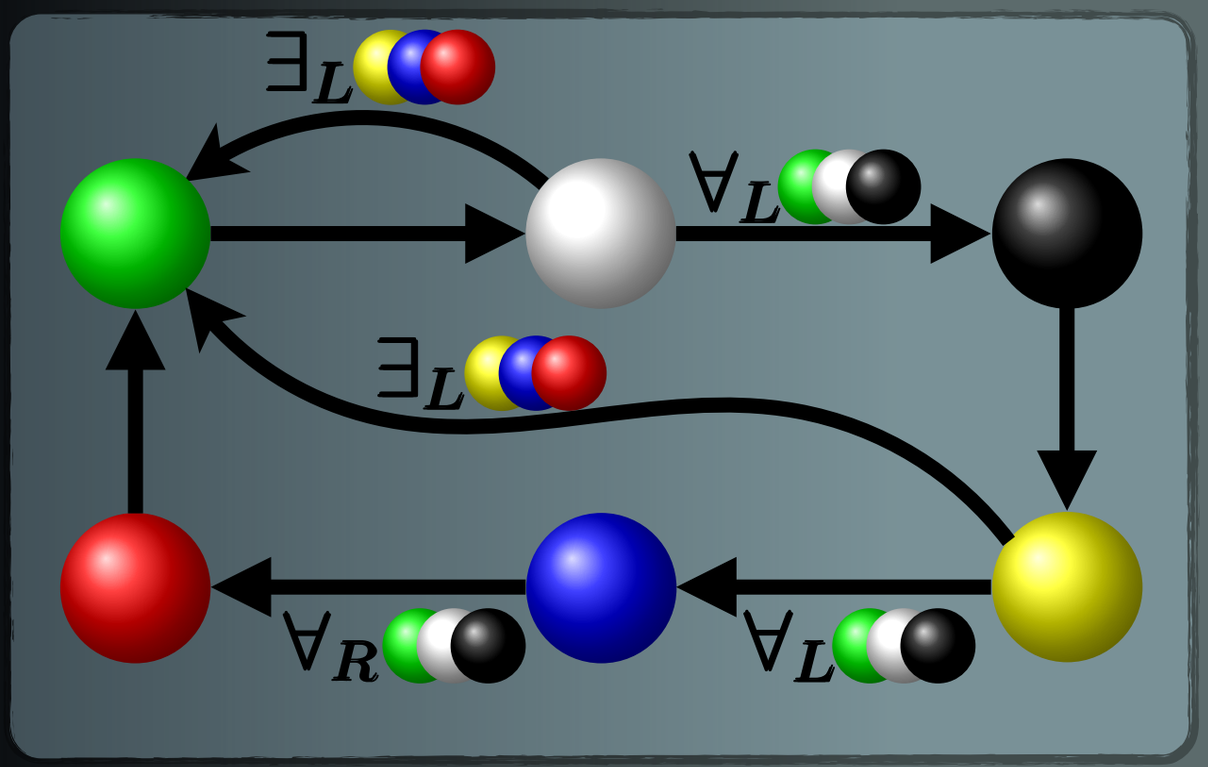
$\mathbb{R}_2$

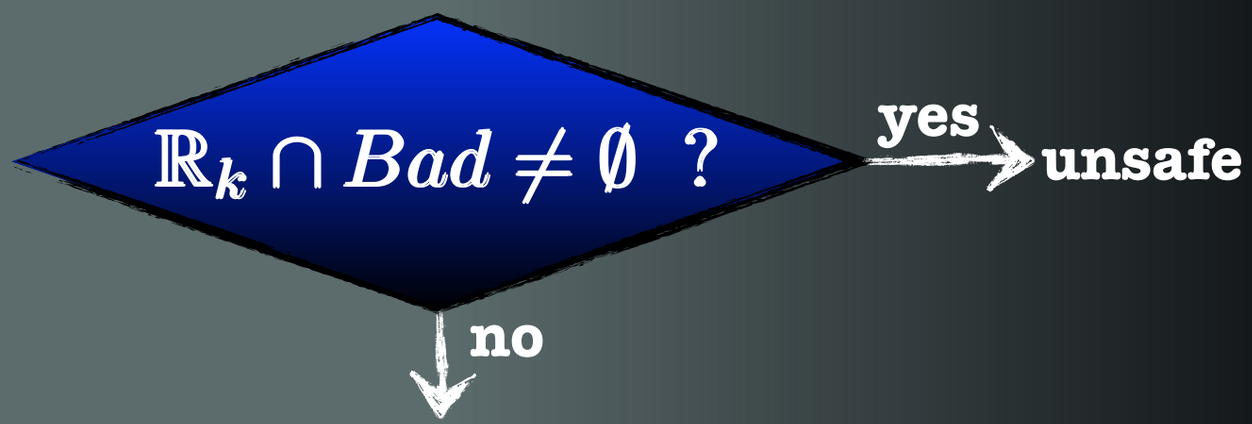

$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$


$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

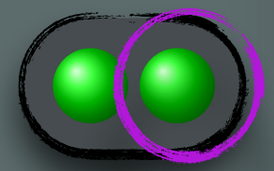
$\mathbb{R}_2$

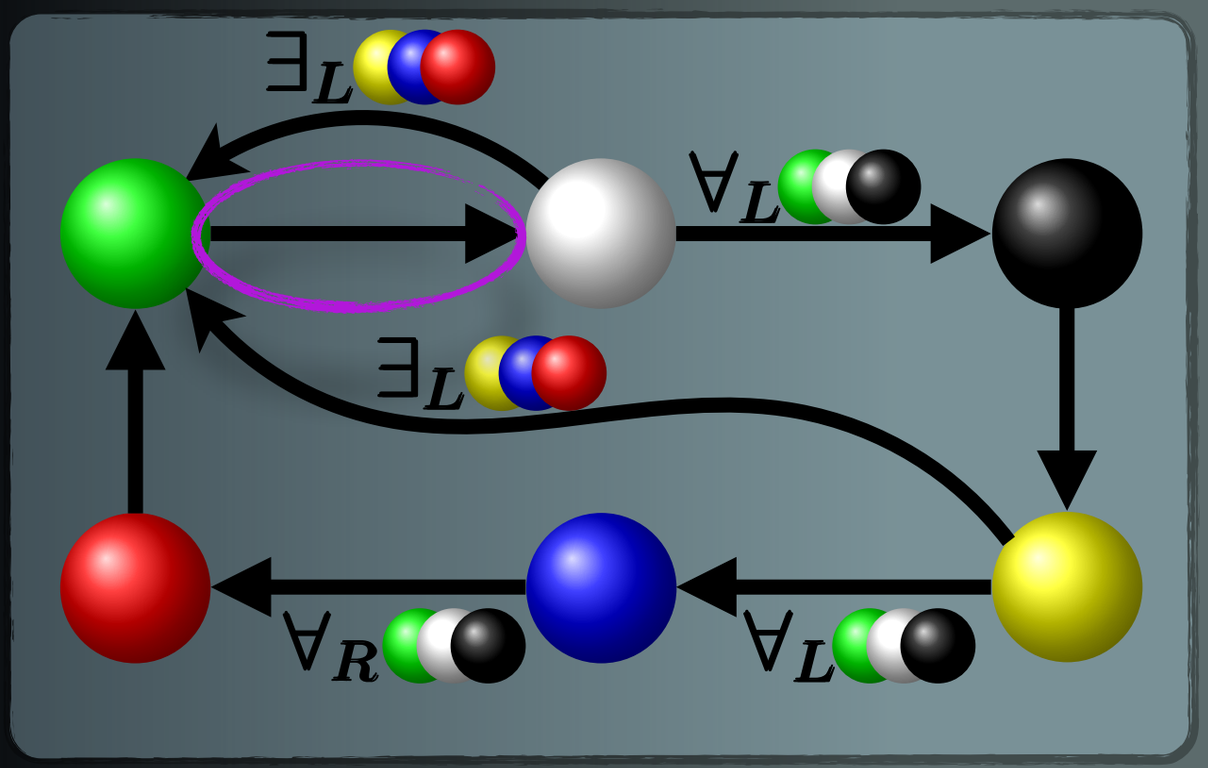

$Init_2$

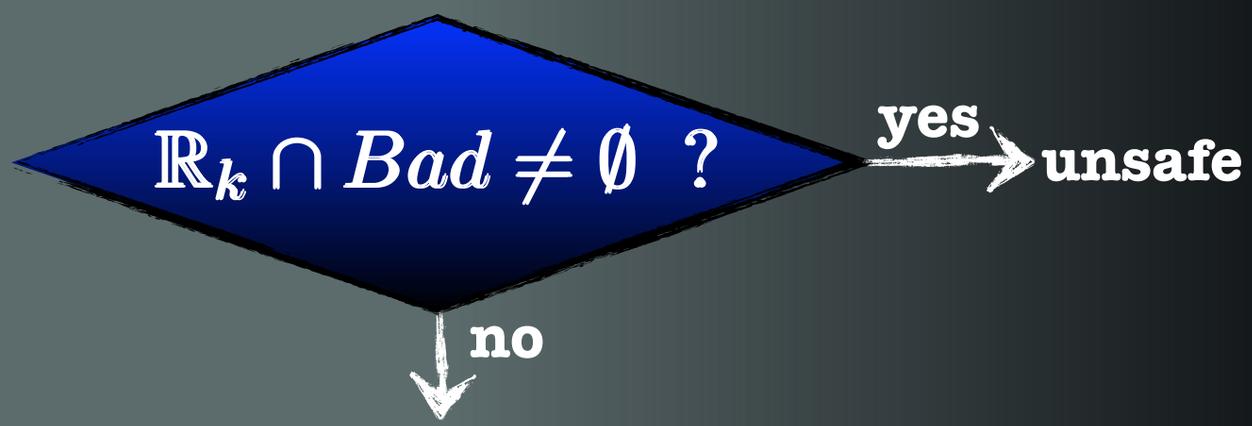

$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$


$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

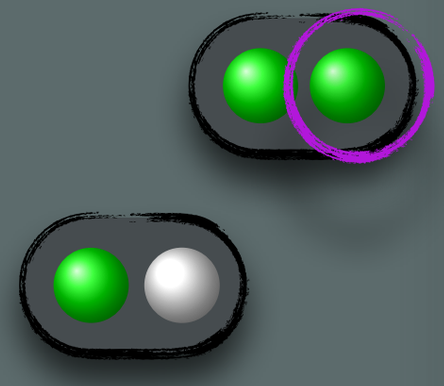
$\mathbb{R}_2$

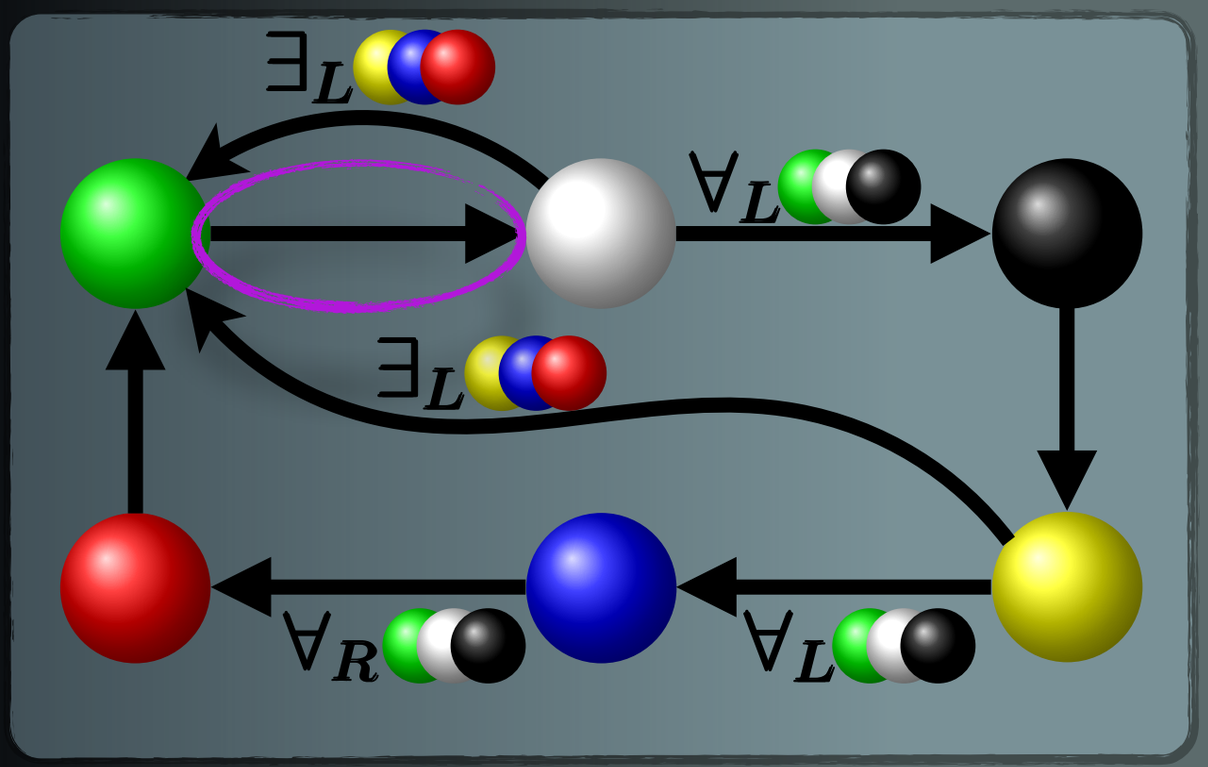


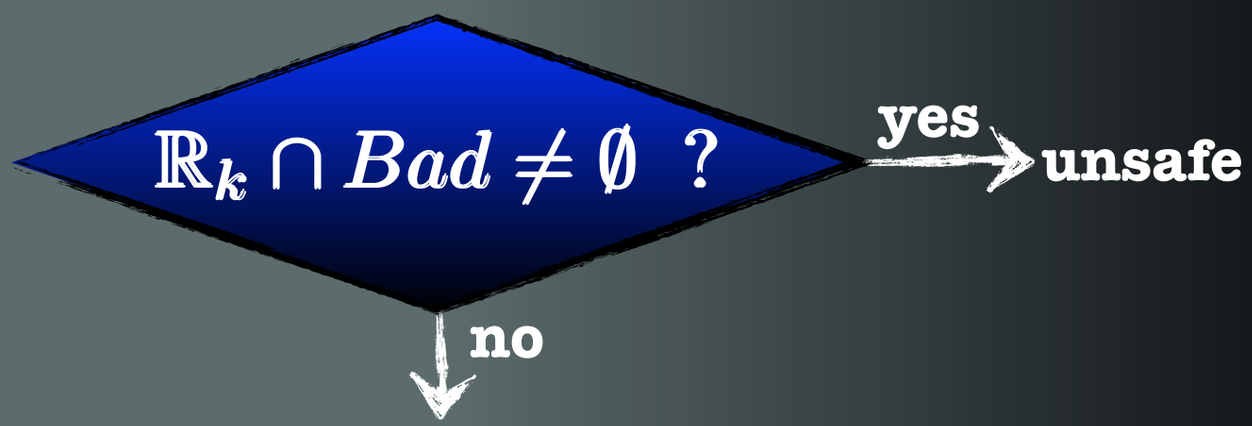

$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$


$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

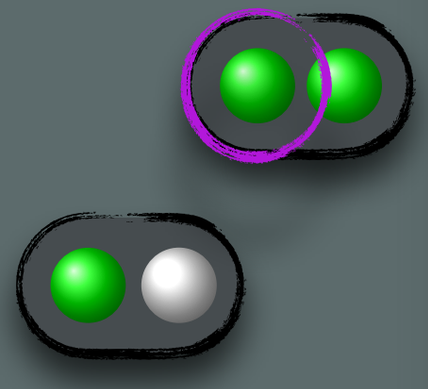
$\mathbb{R}_2$

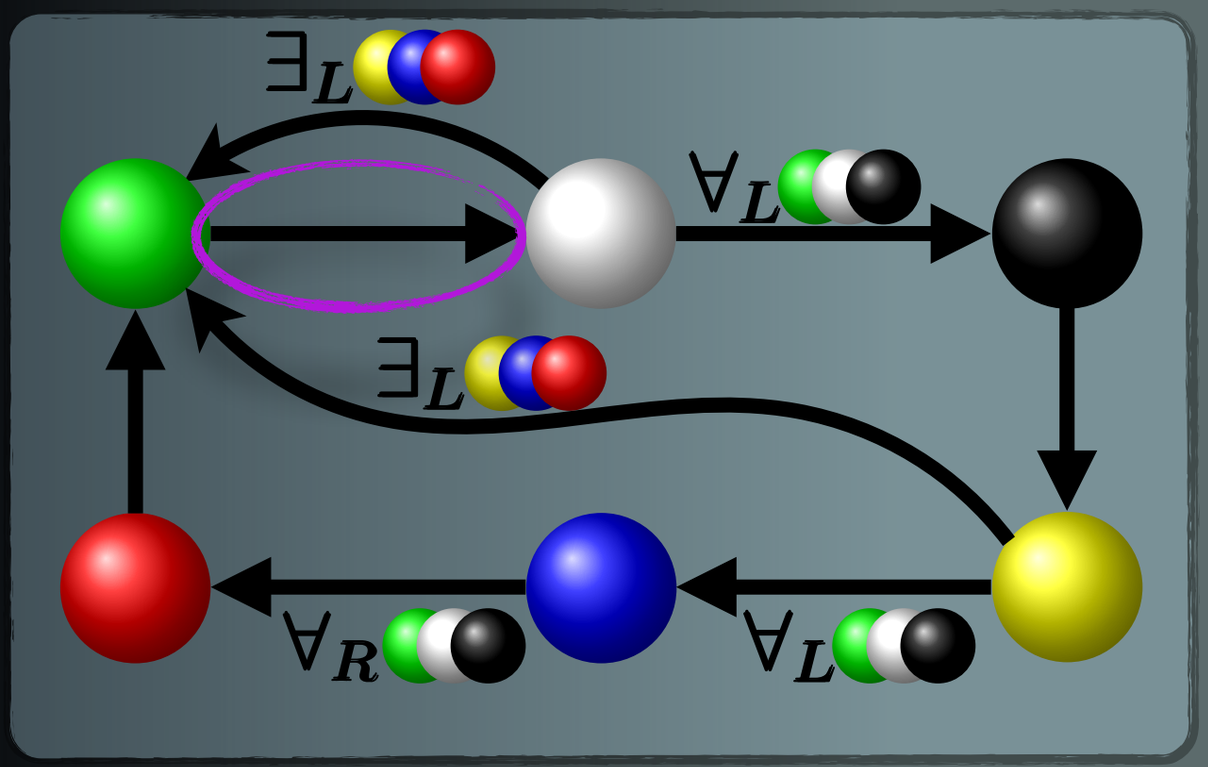


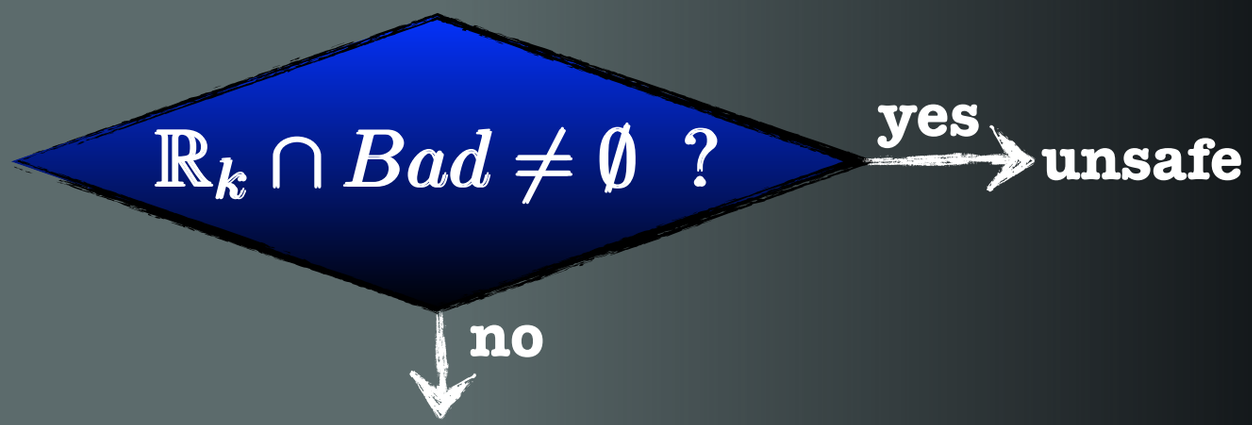

$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$


$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

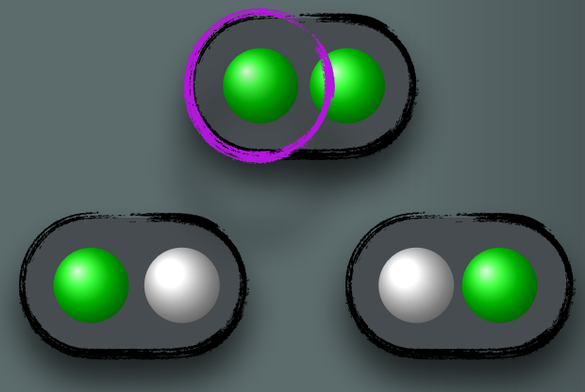
$\mathbb{R}_2$

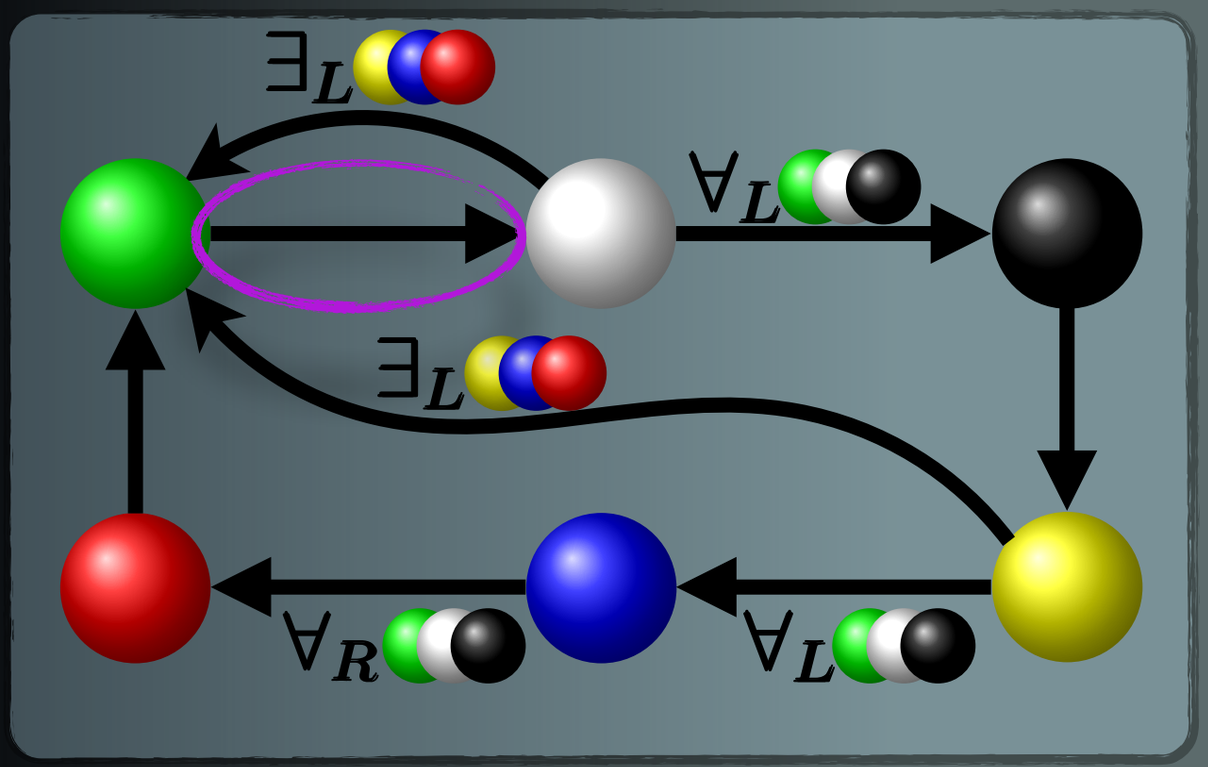


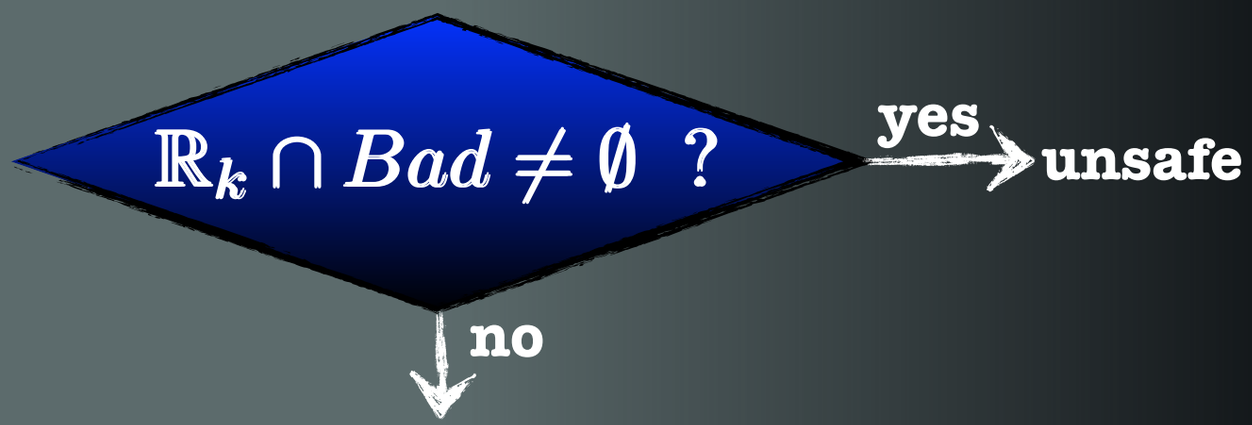

$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$


$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

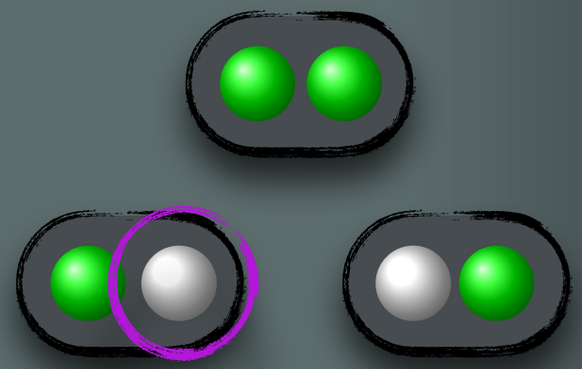
$\mathbb{R}_2$

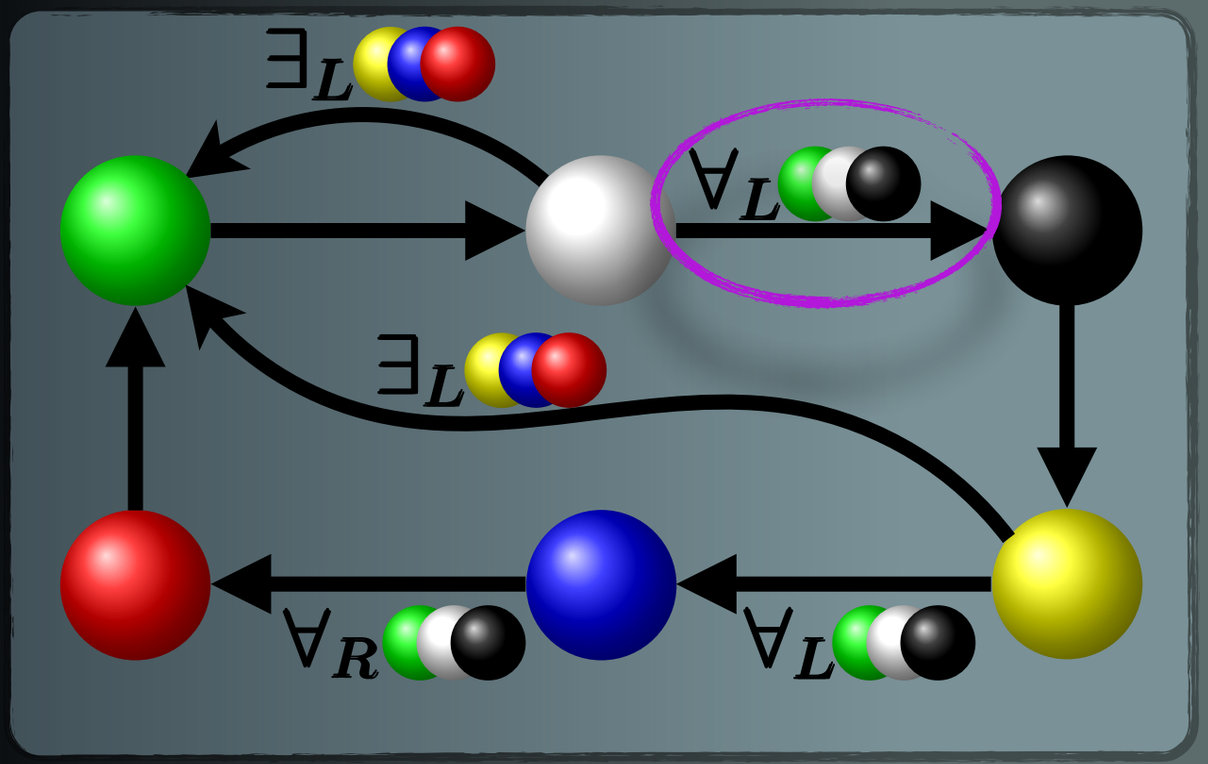


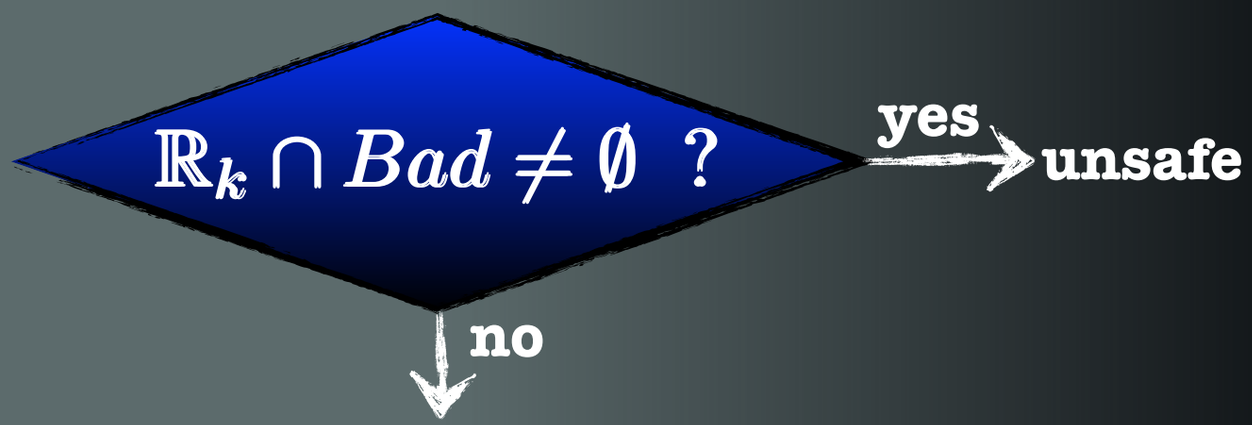

$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$


$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

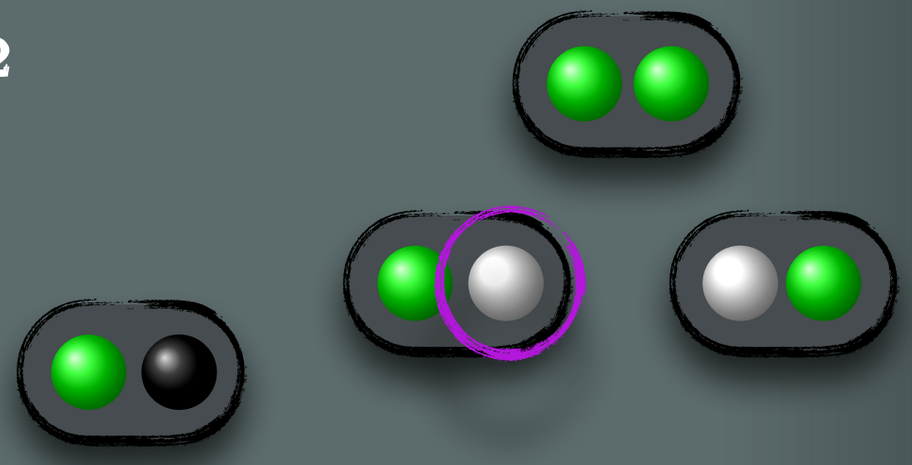
$\mathbb{R}_2$

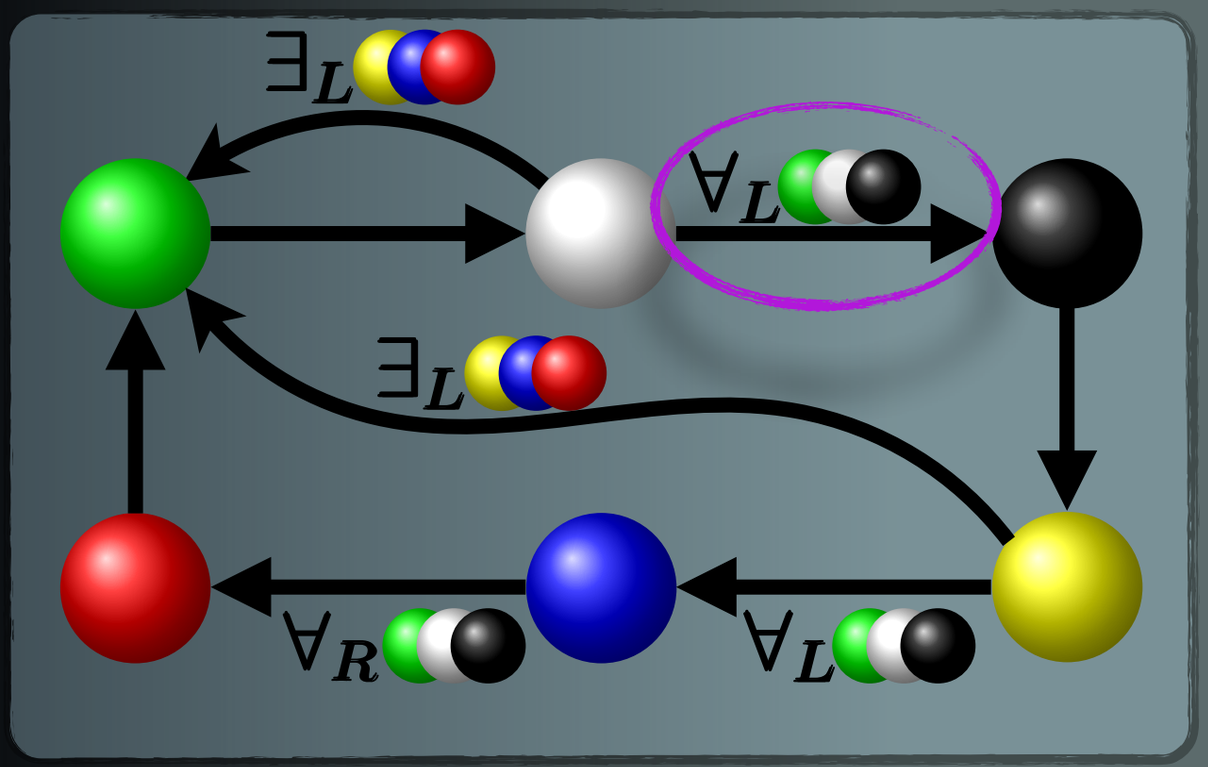


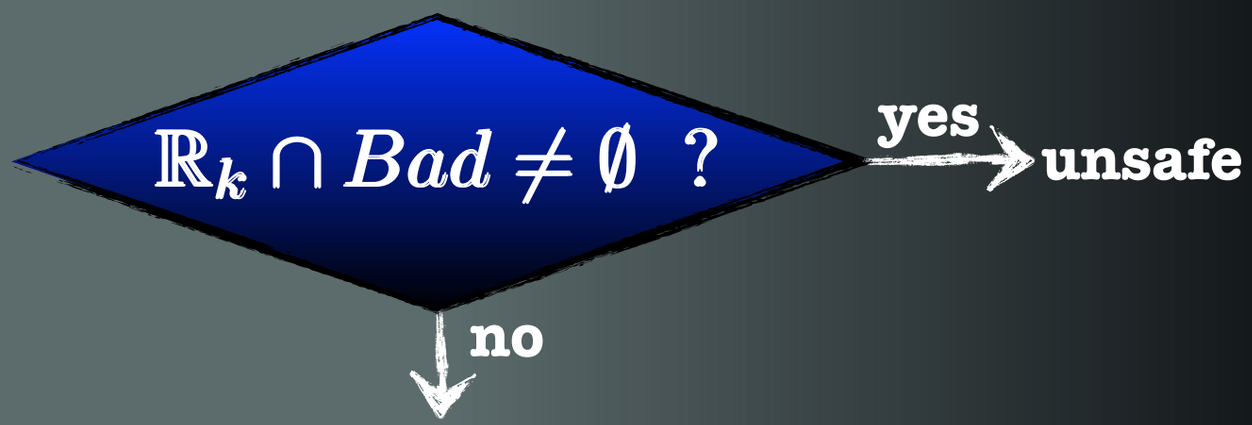

$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$


$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

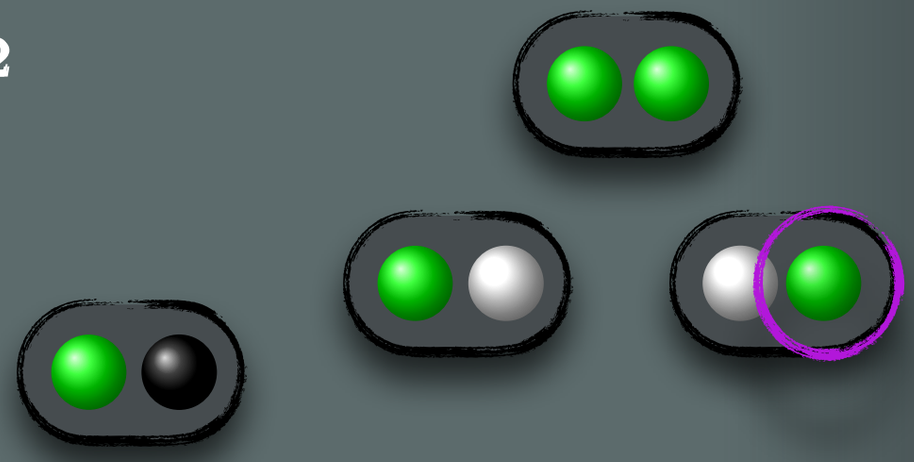
$\mathbb{R}_2$

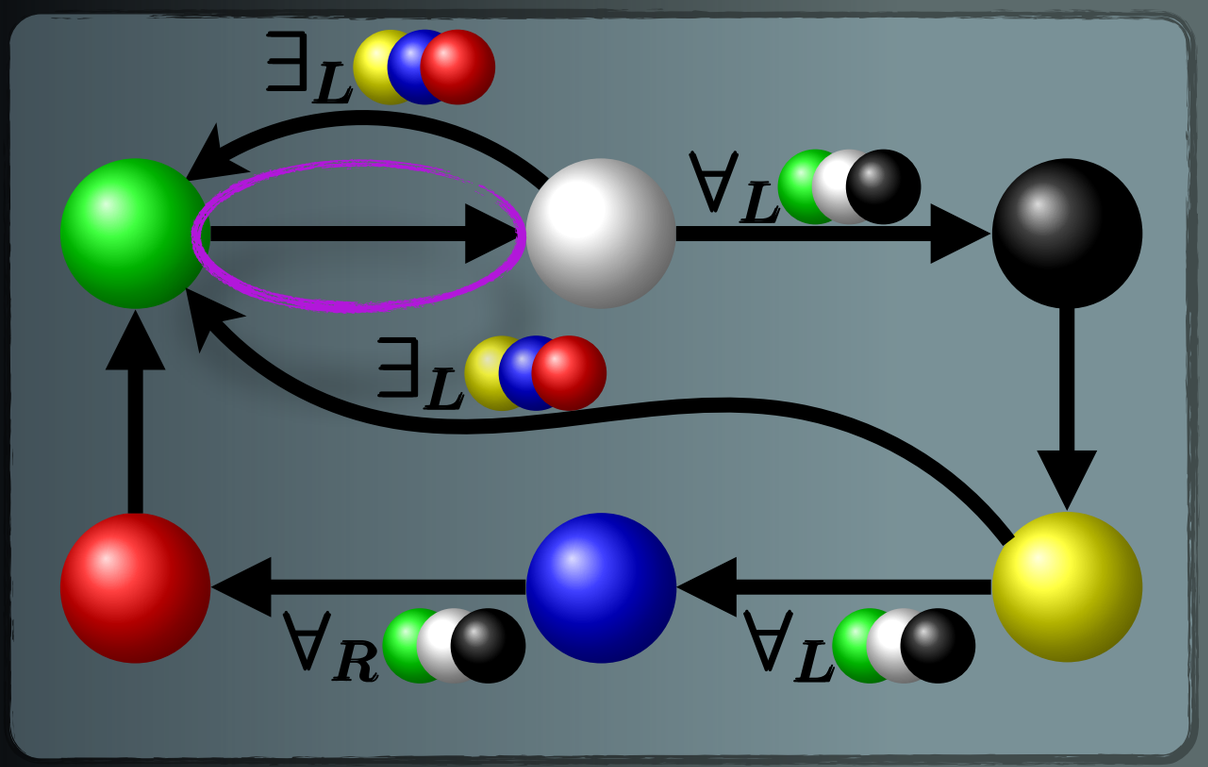


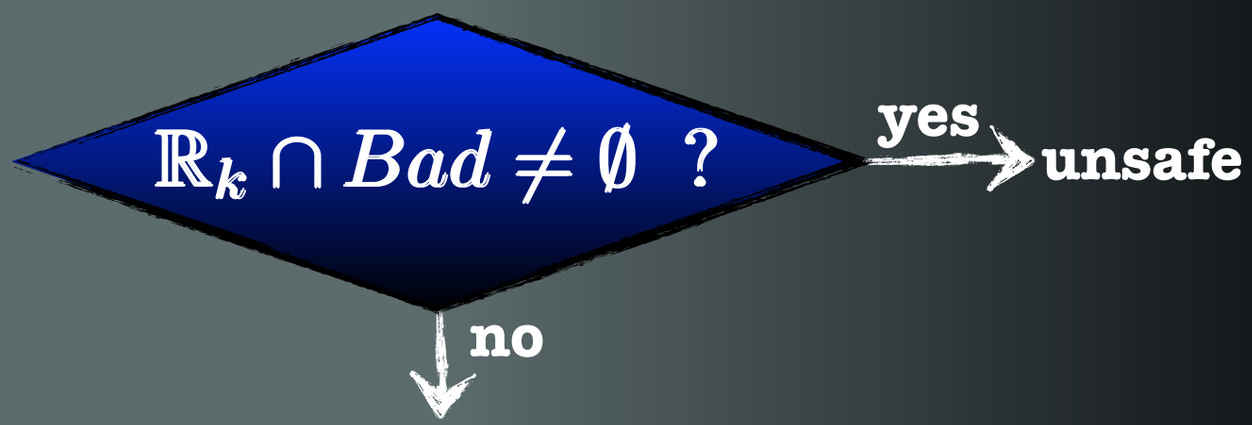

$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$


$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

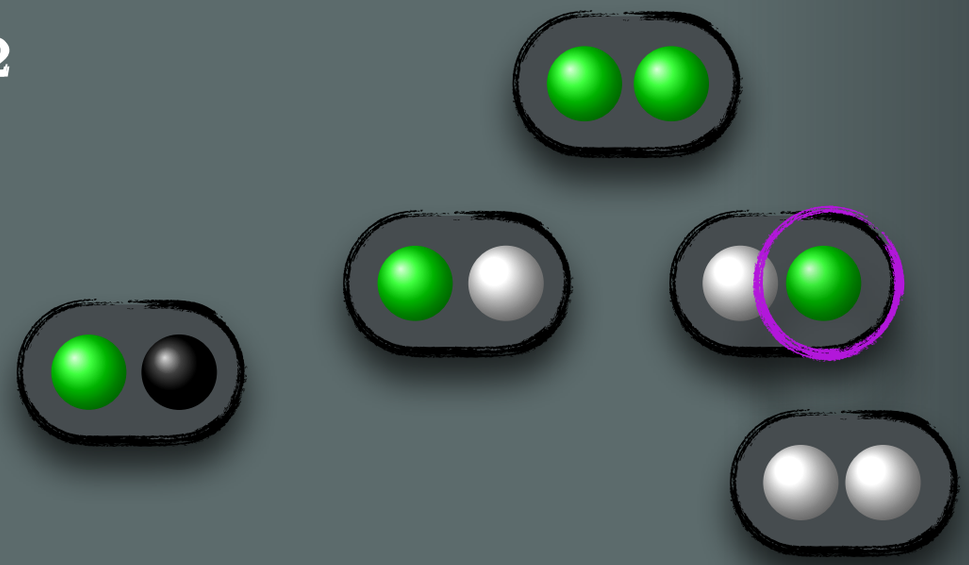
$\mathbb{R}_2$

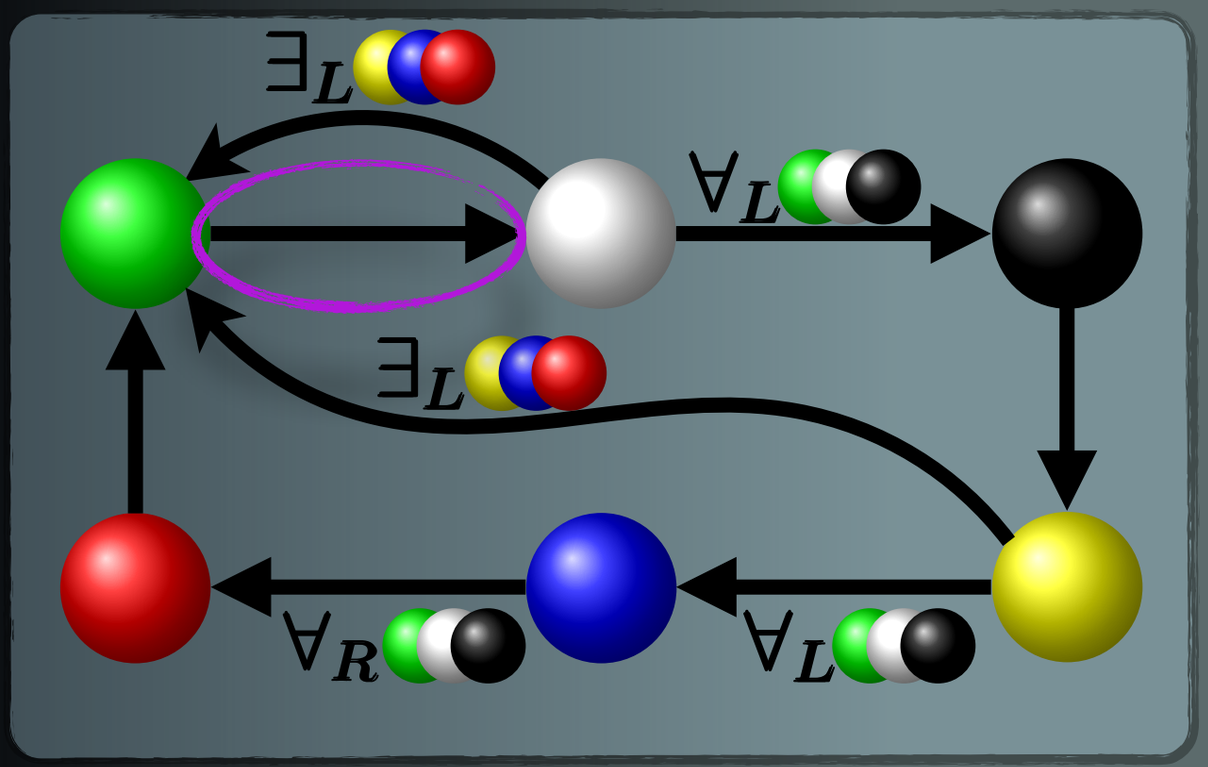


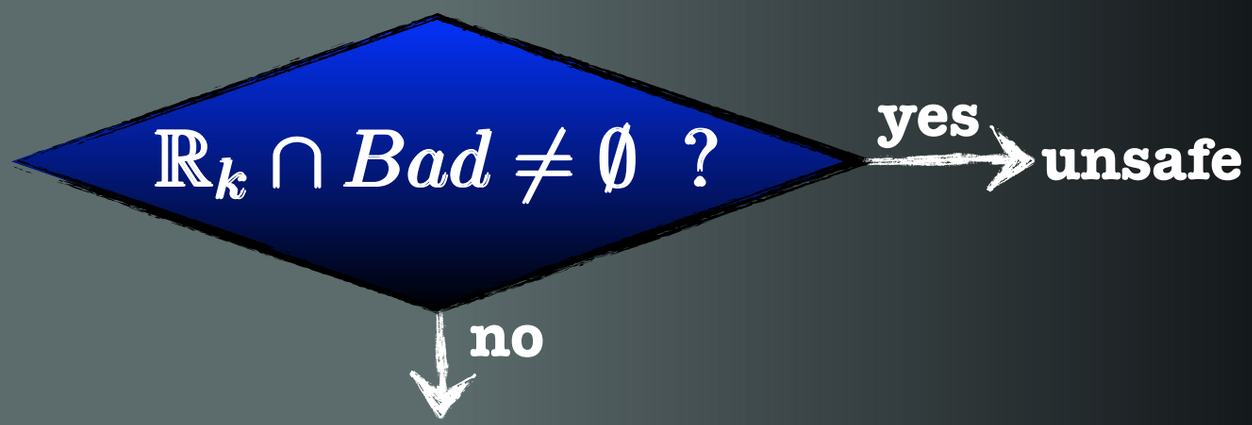

$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$


$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

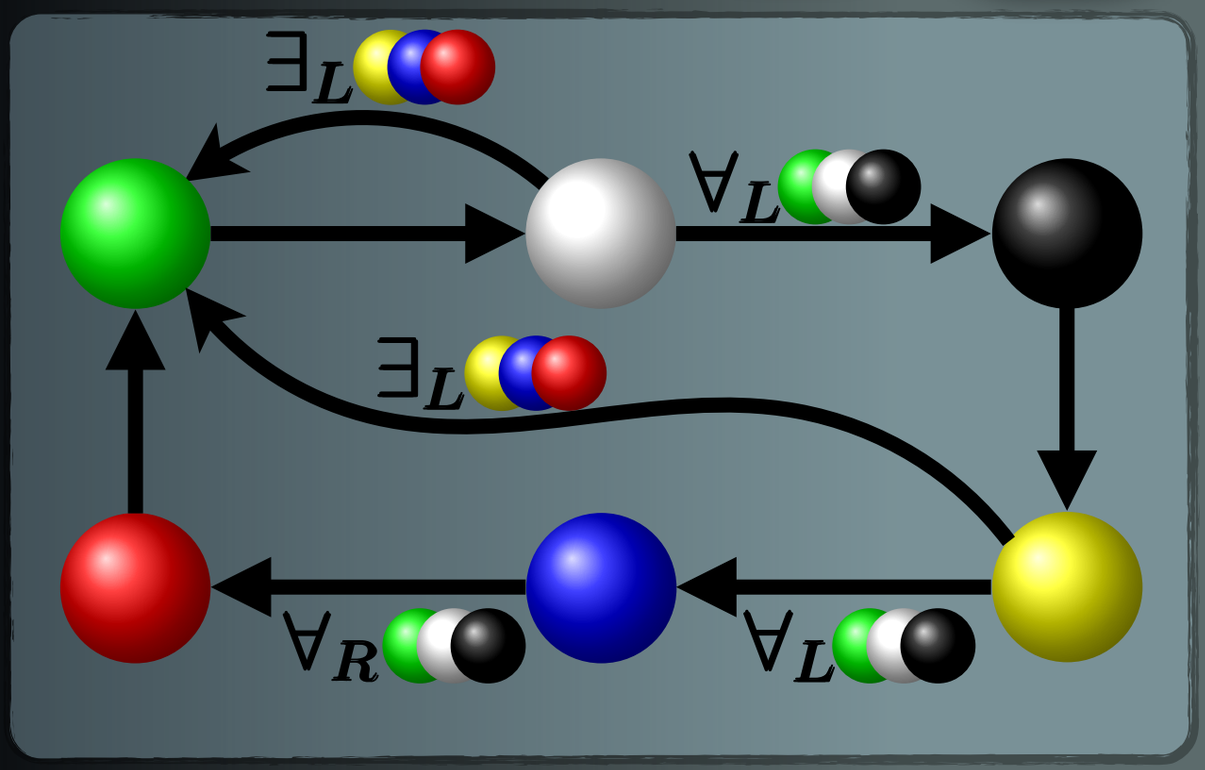
$\mathbb{R}_2$




$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$


$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no

$\mathbb{R}_2$

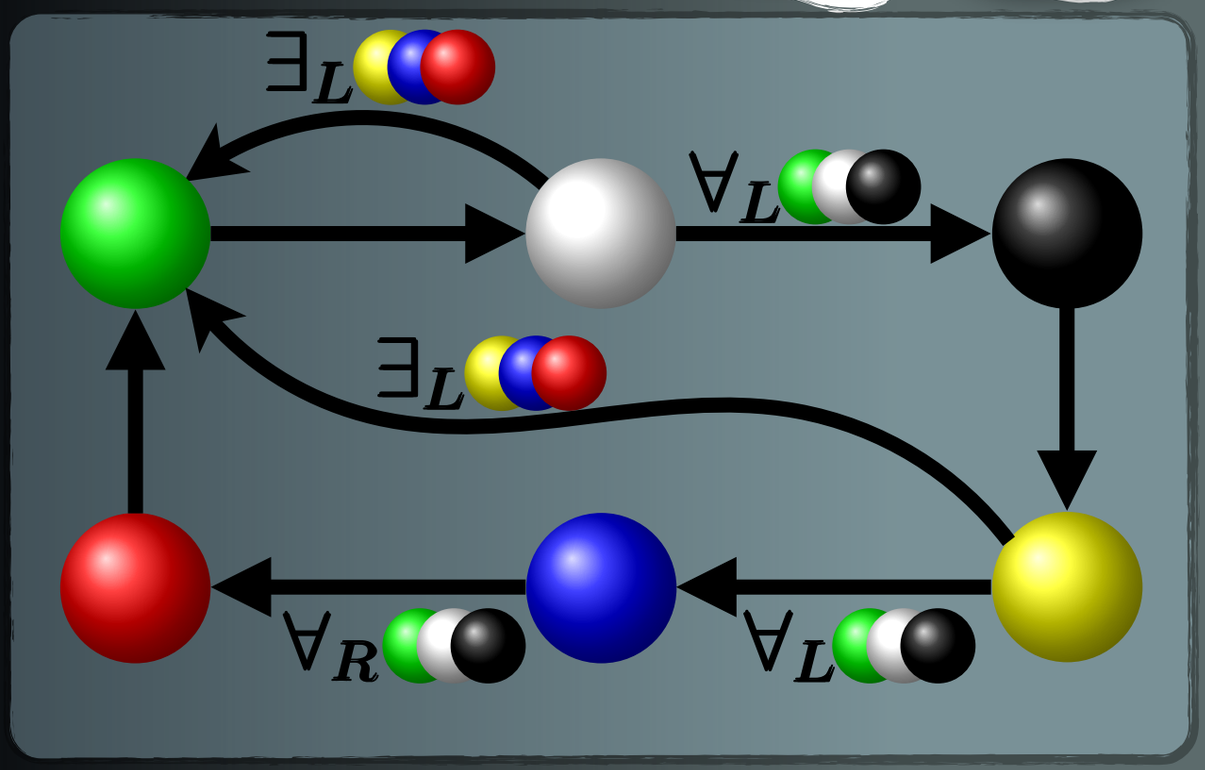


$\mathbb{R}_k \cap Bad \neq \emptyset$?

yes → unsafe

no

$\mathbb{R}_2$



$\mathbb{R}_k \cap Bad \neq \emptyset$?

yes → unsafe

no

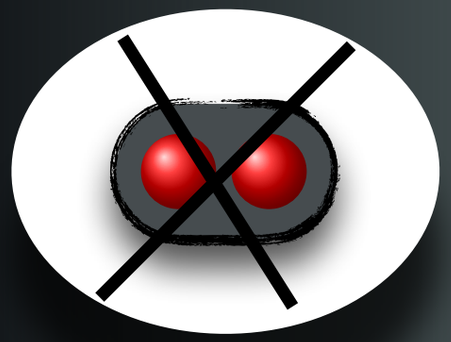
$\mathbb{R}_2$

$\exists_L$ $\forall_L$ $\exists_L$ $\forall_R$ $\forall_L$

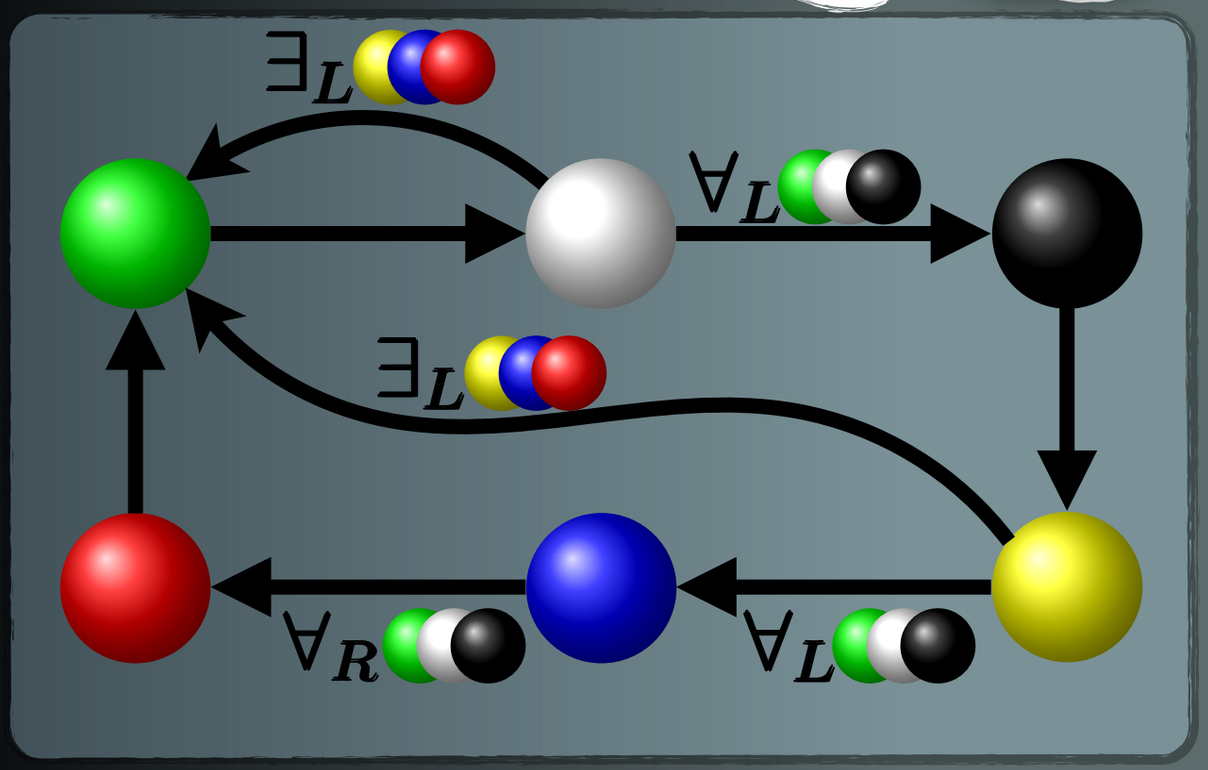
$\mathbb{R}_k \cap Bad \neq \emptyset$?

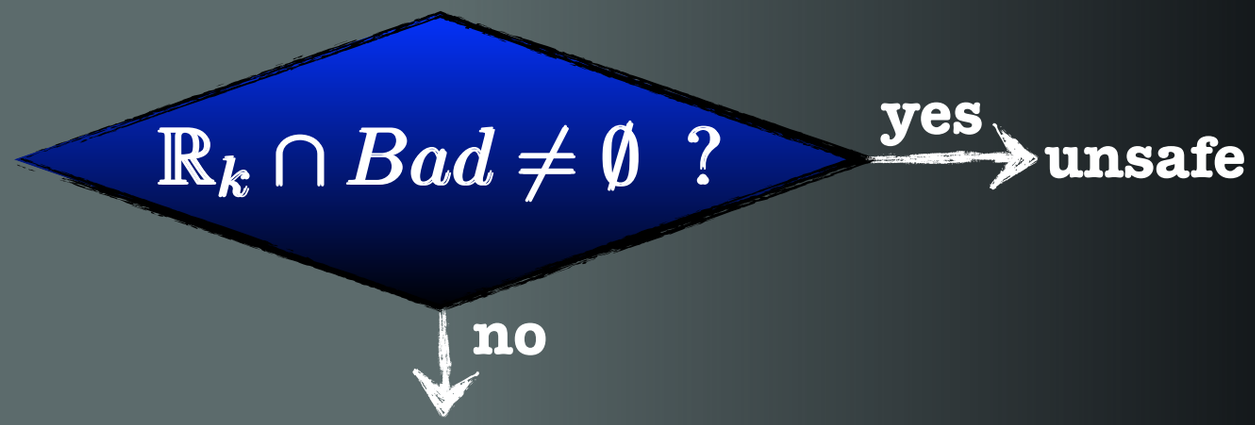
yes → unsafe

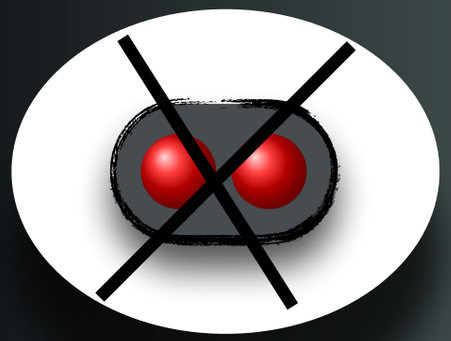
no




$\mathbb{R}_2$

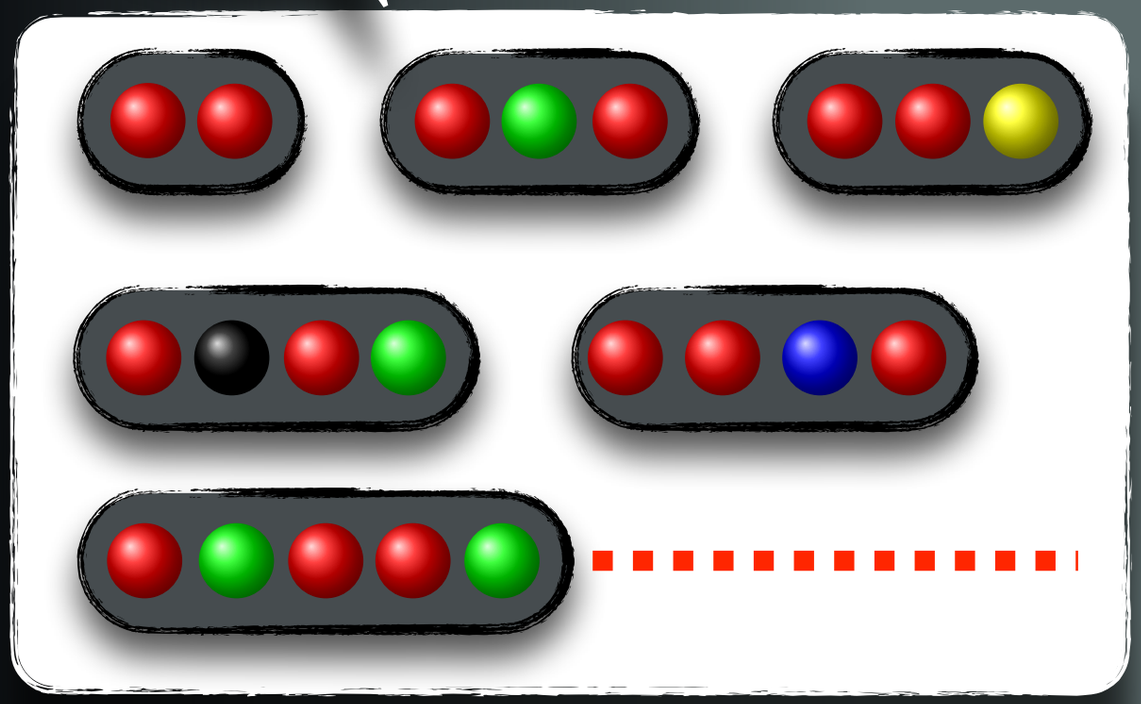

$\exists_L$
$\forall_L$
$\exists_L$
$\forall_R$
$\forall_L$


$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no



$\mathbb{R}_2$

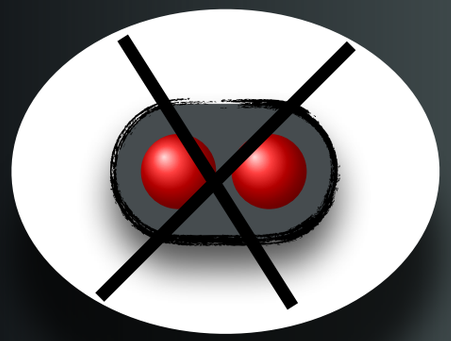
Bad Configurations



$\mathbb{R}_k \cap Bad \neq \emptyset$?

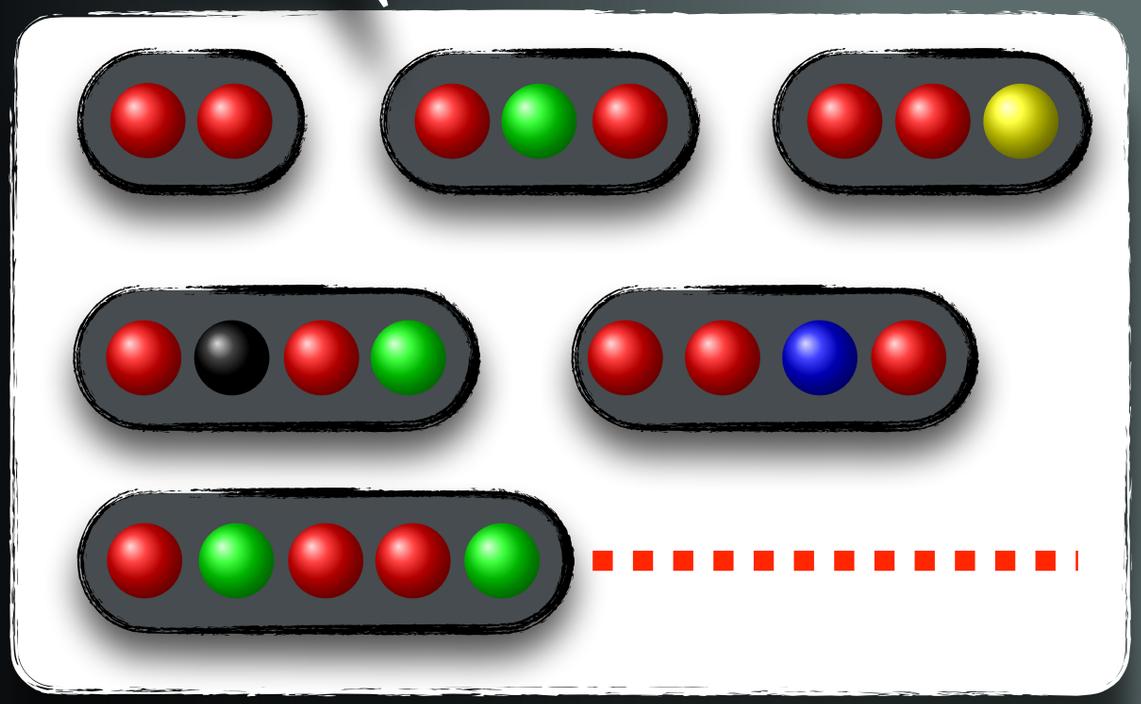
yes → unsafe

no



$\mathbb{R}_2$

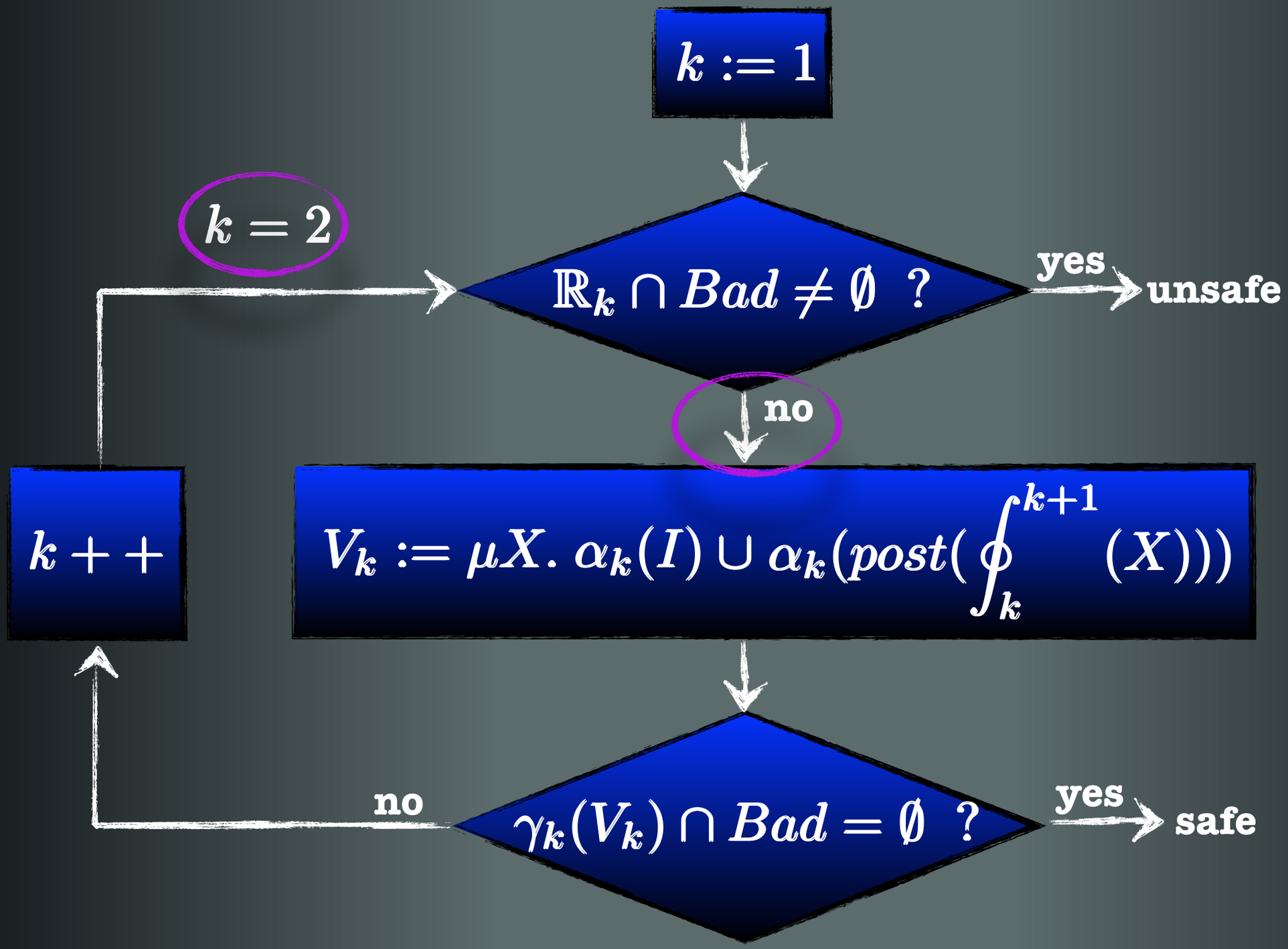
Bad Configurations

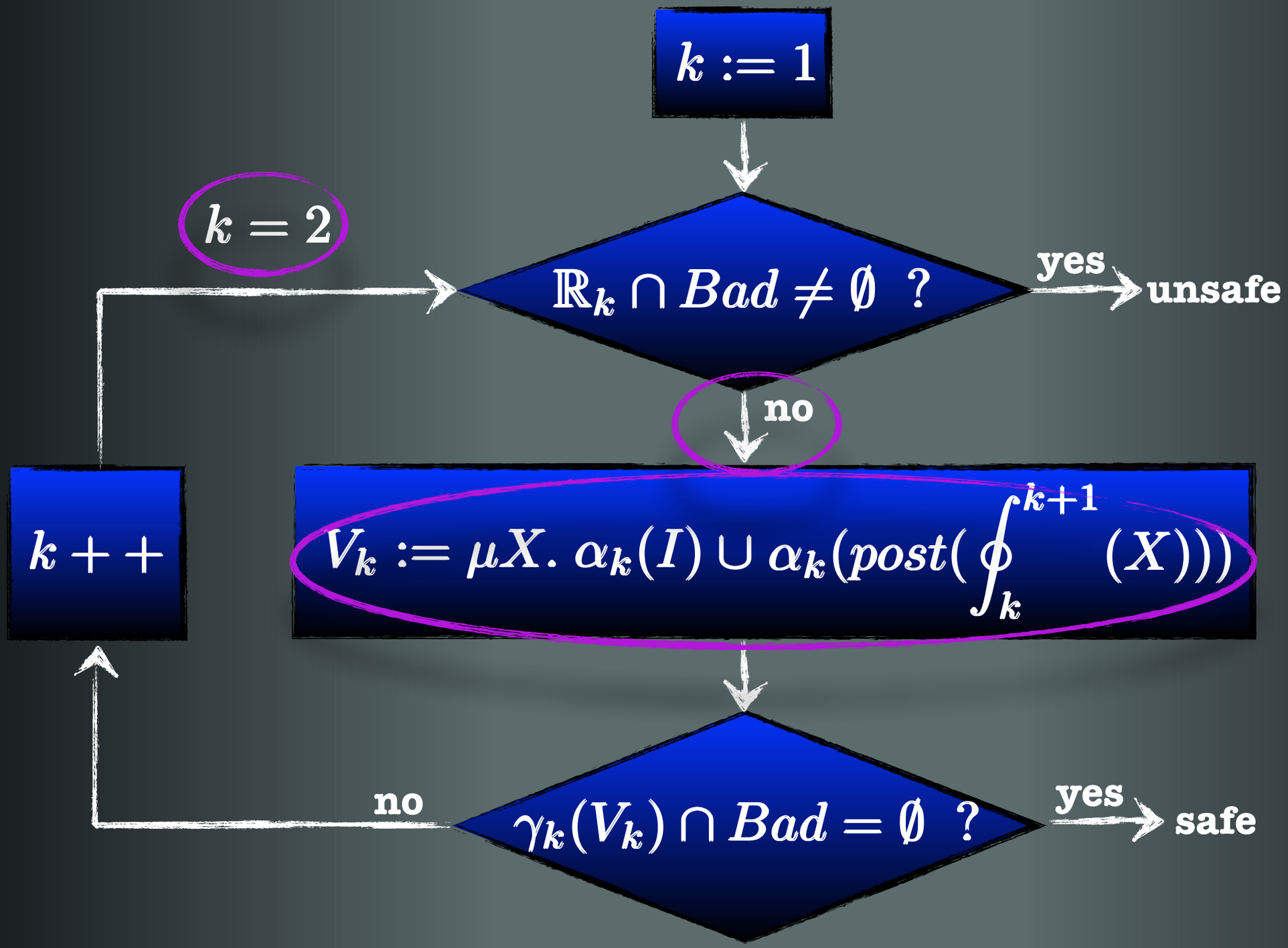


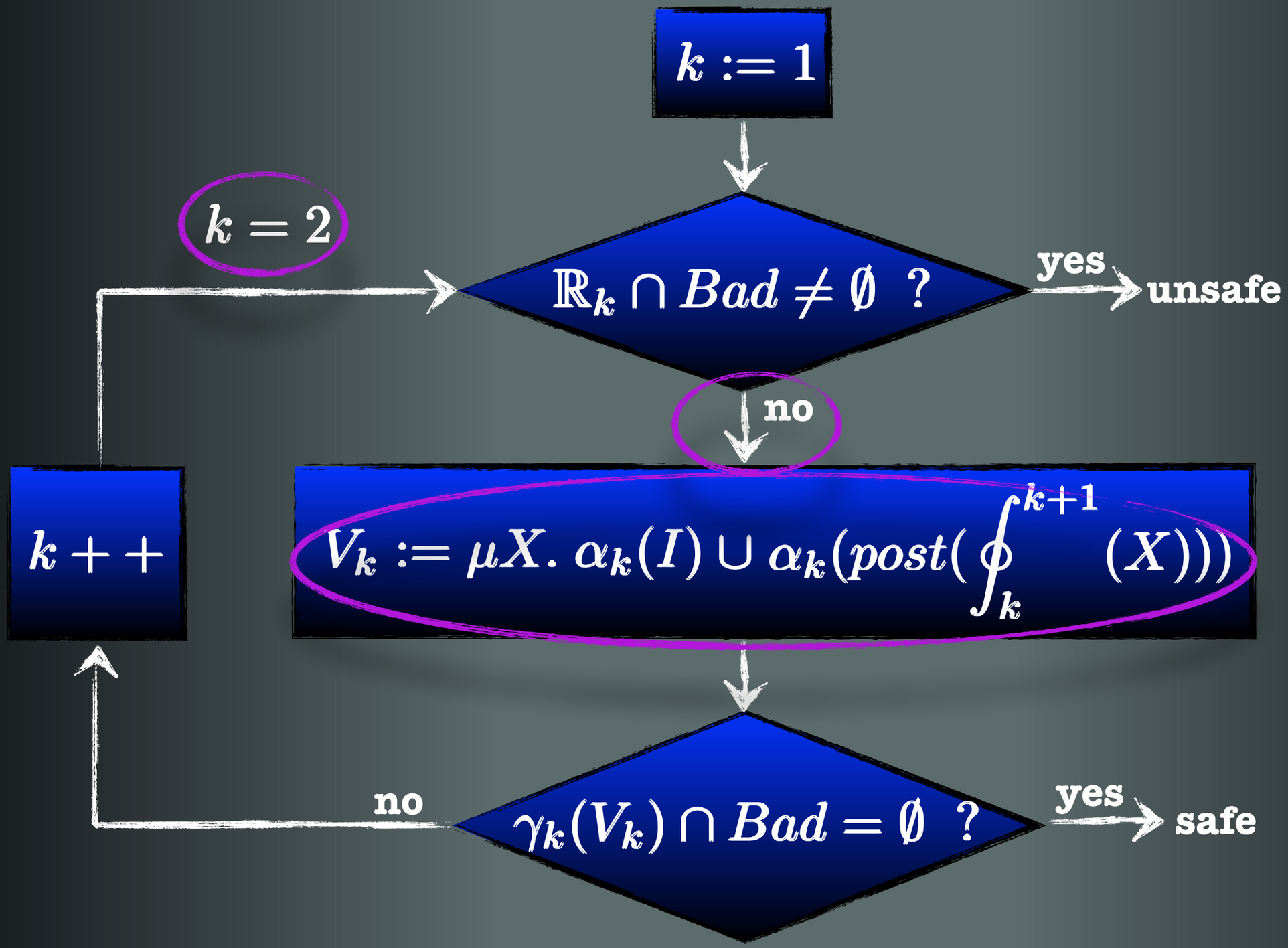
$\mathbb{R}_k \cap Bad \neq \emptyset$?

yes → unsafe

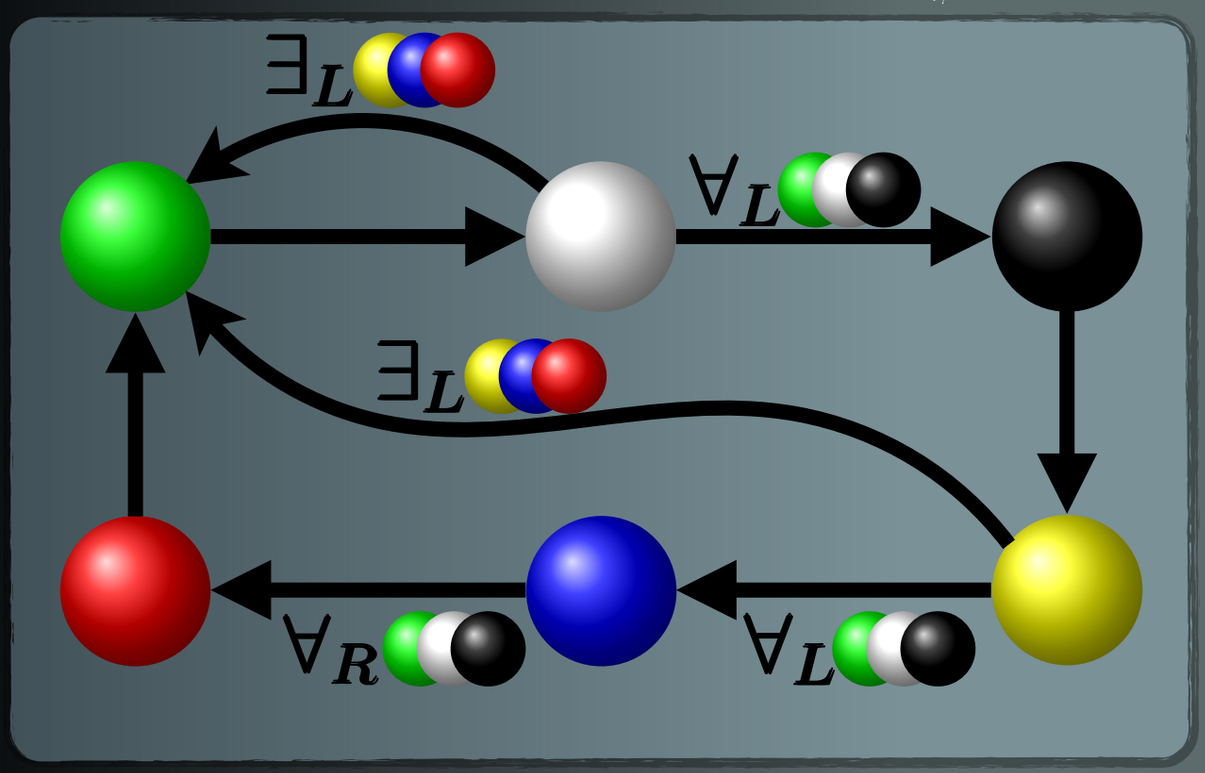
no


$k := 1$
$k = 2$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$k++$
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$
no
$\gamma_k(V_k) \cap Bad = \emptyset$?
yes
safe


$k := 1$
$k = 2$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$k++$
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$
no
$\gamma_k(V_k) \cap Bad = \emptyset$?
yes
safe

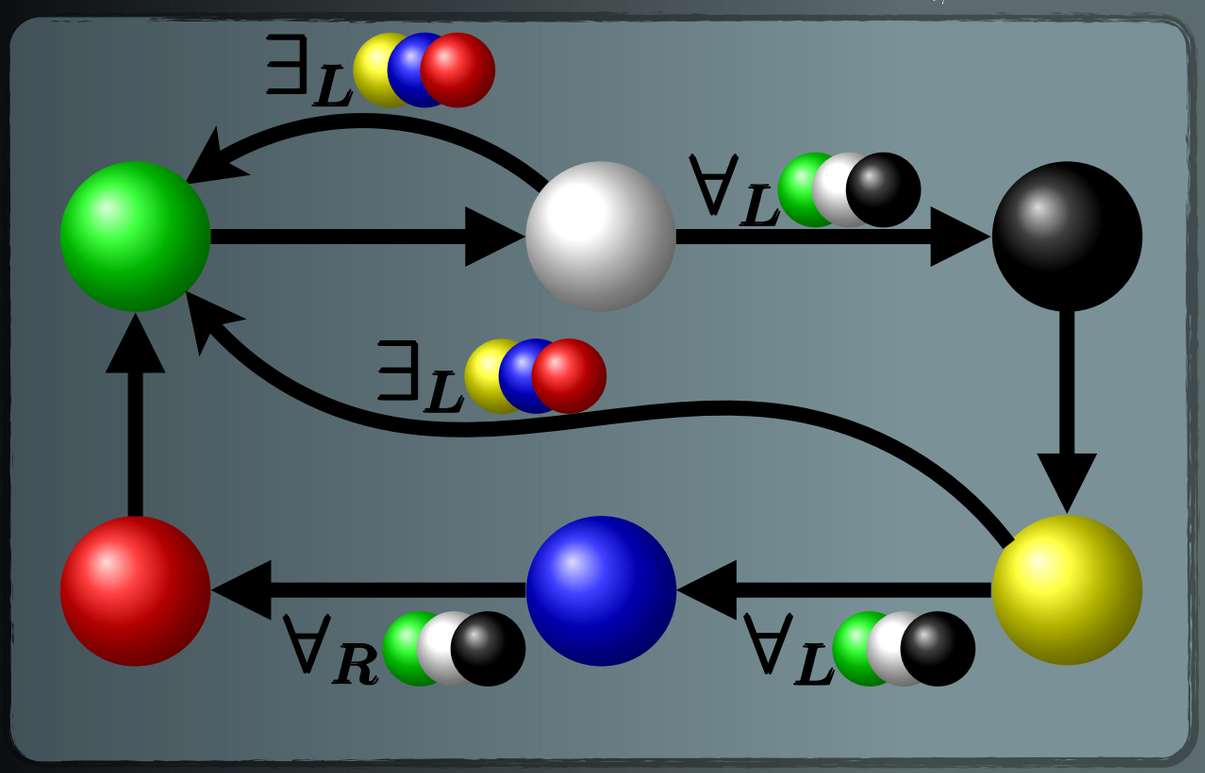


$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\oint_2^3(X)))$$



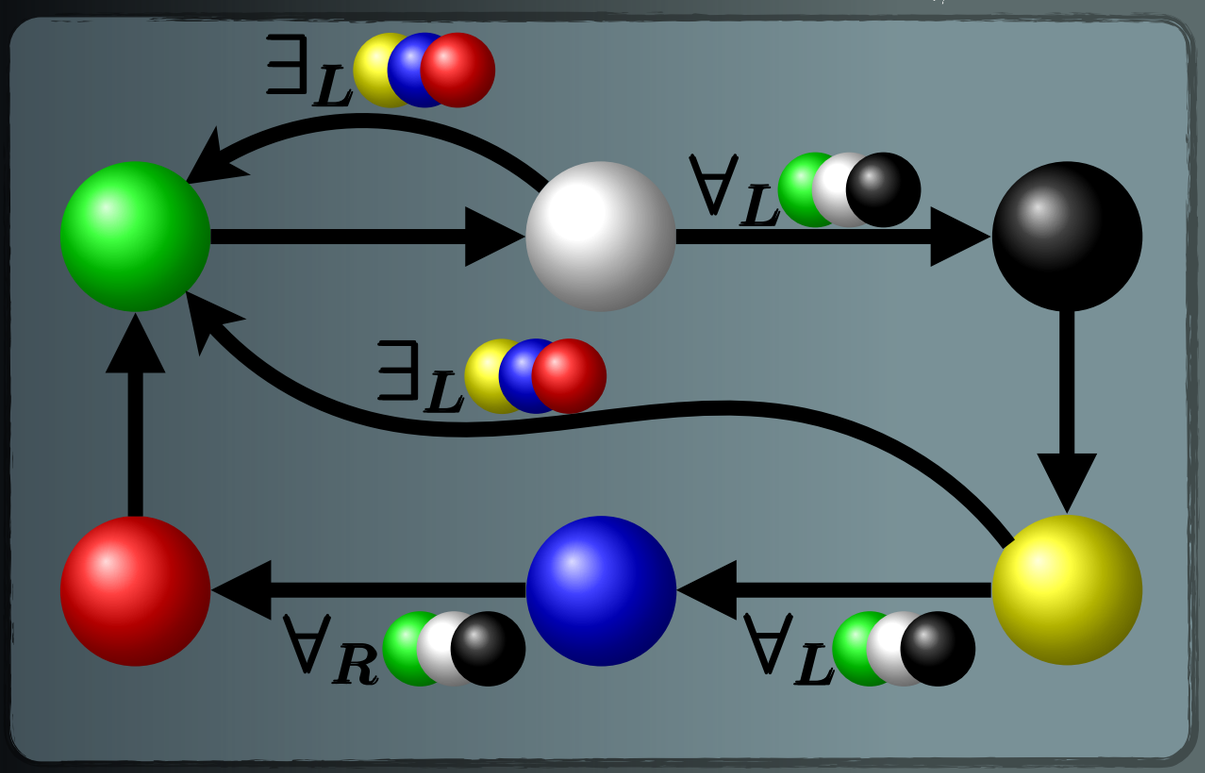
$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$





$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$





$$V_2 := \mu X.\, \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$

X

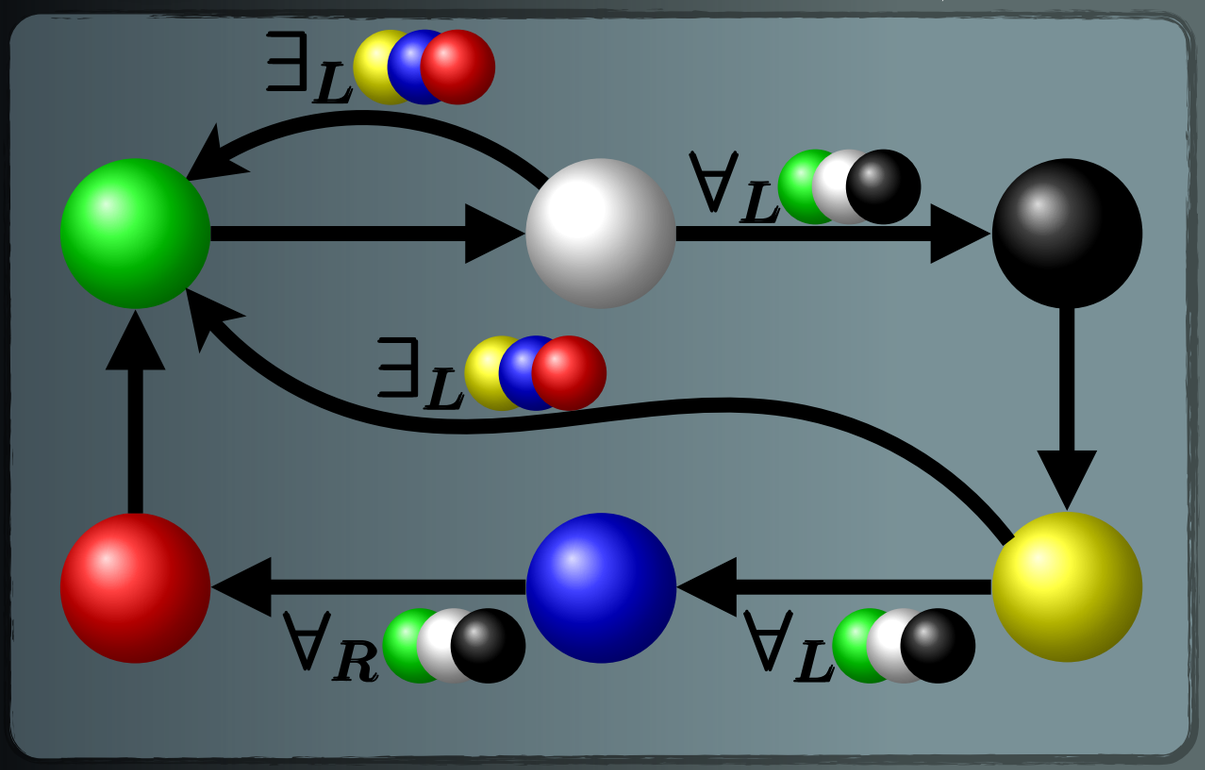
$Init_5$

$Init_4$

$Init_3$

$Init_2$

$Init_1$



$$V_2 := \mu X.\, \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$

X

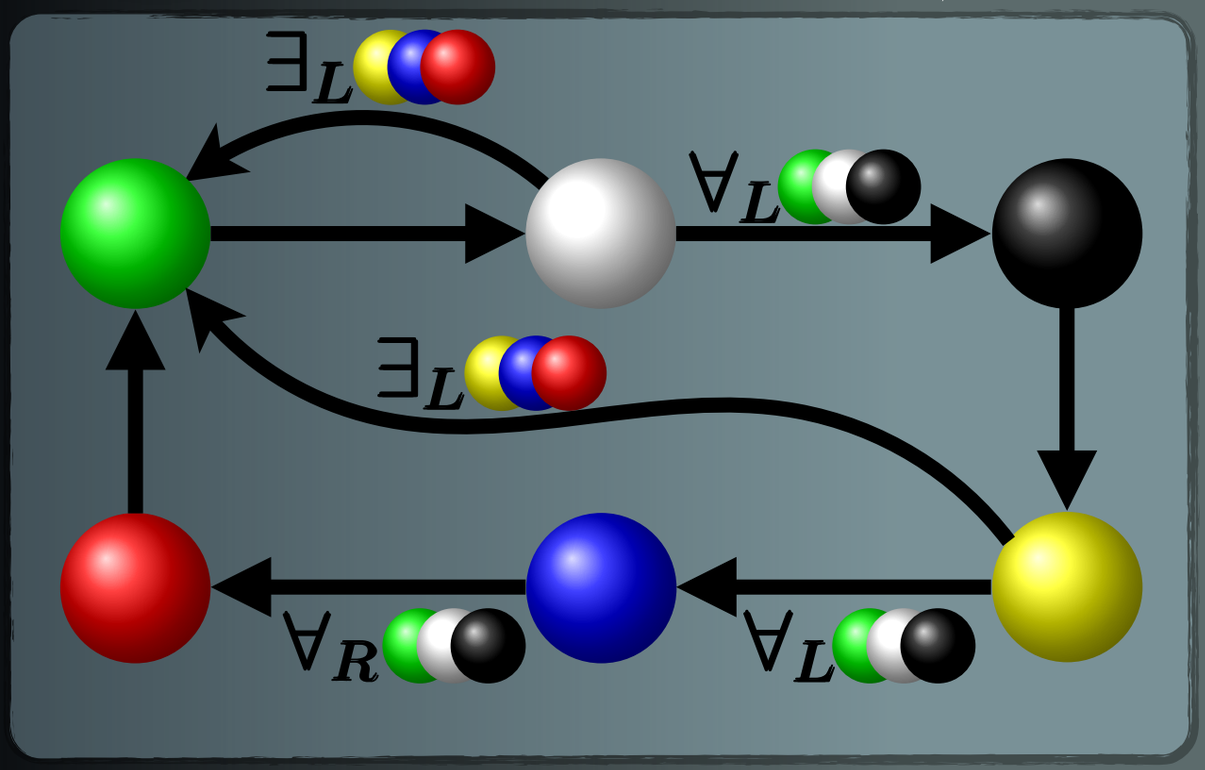
$Init_5$

$Init_4$

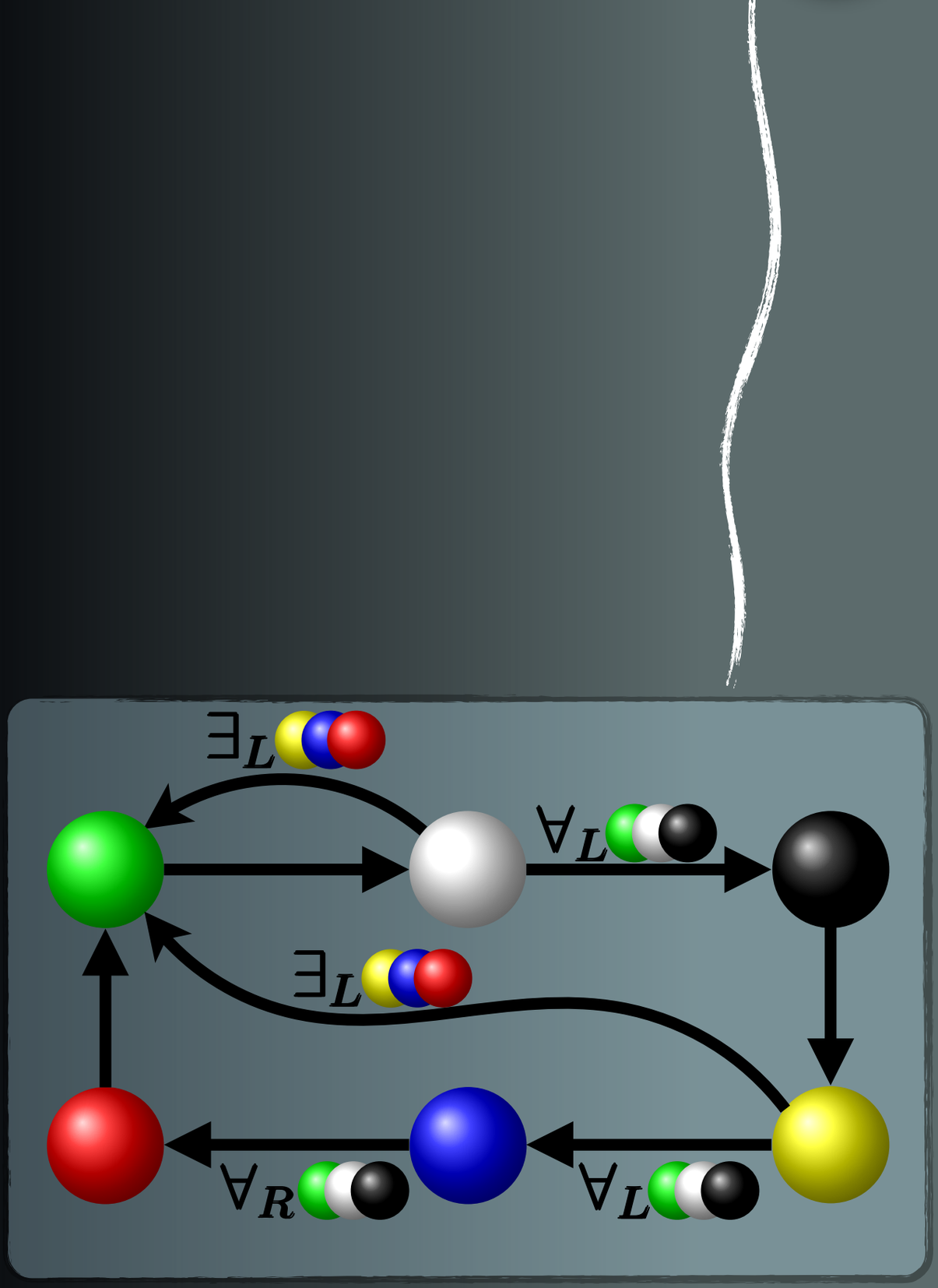
$Init_3$

$Init_2$

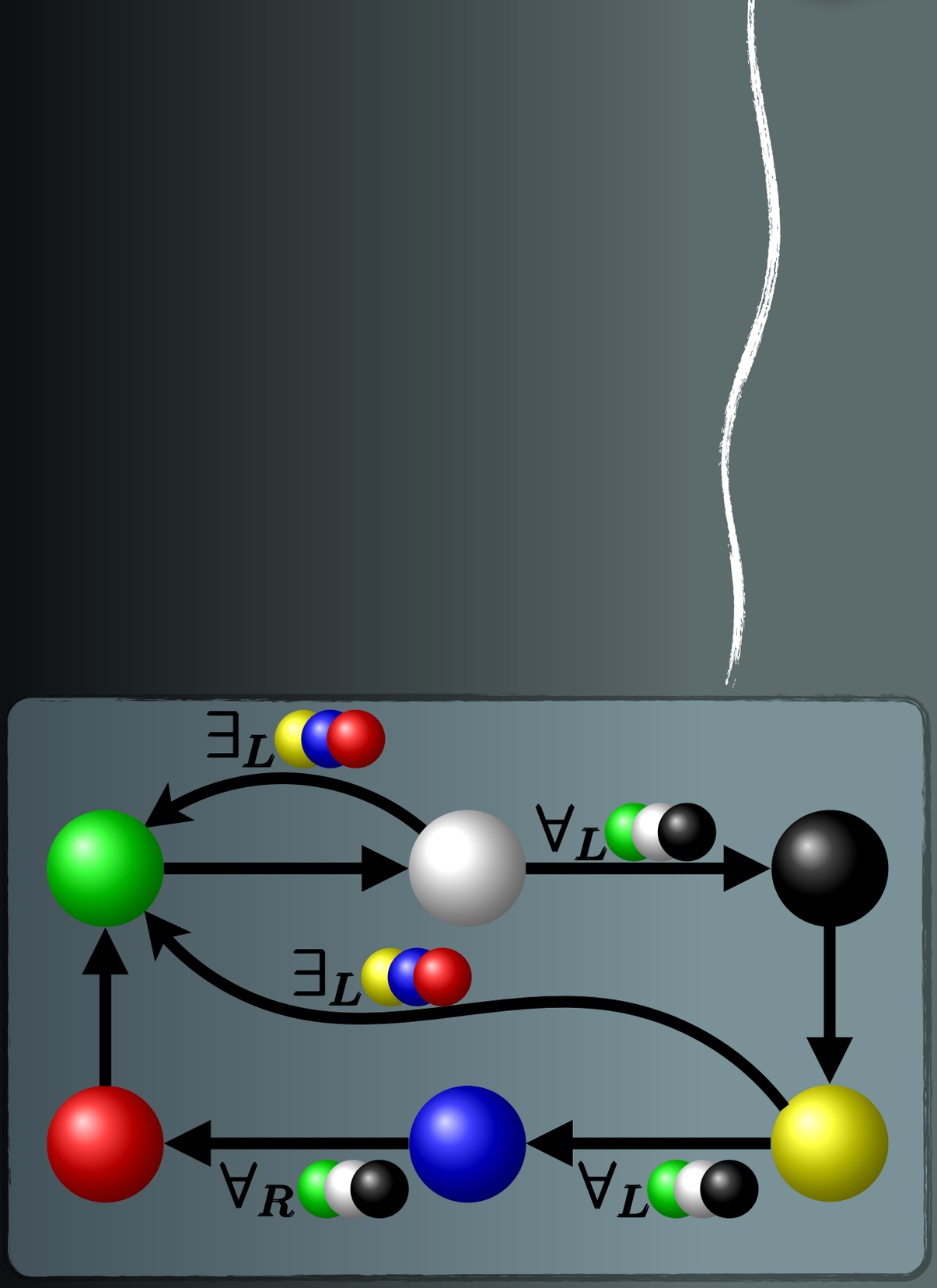
$Init_1$



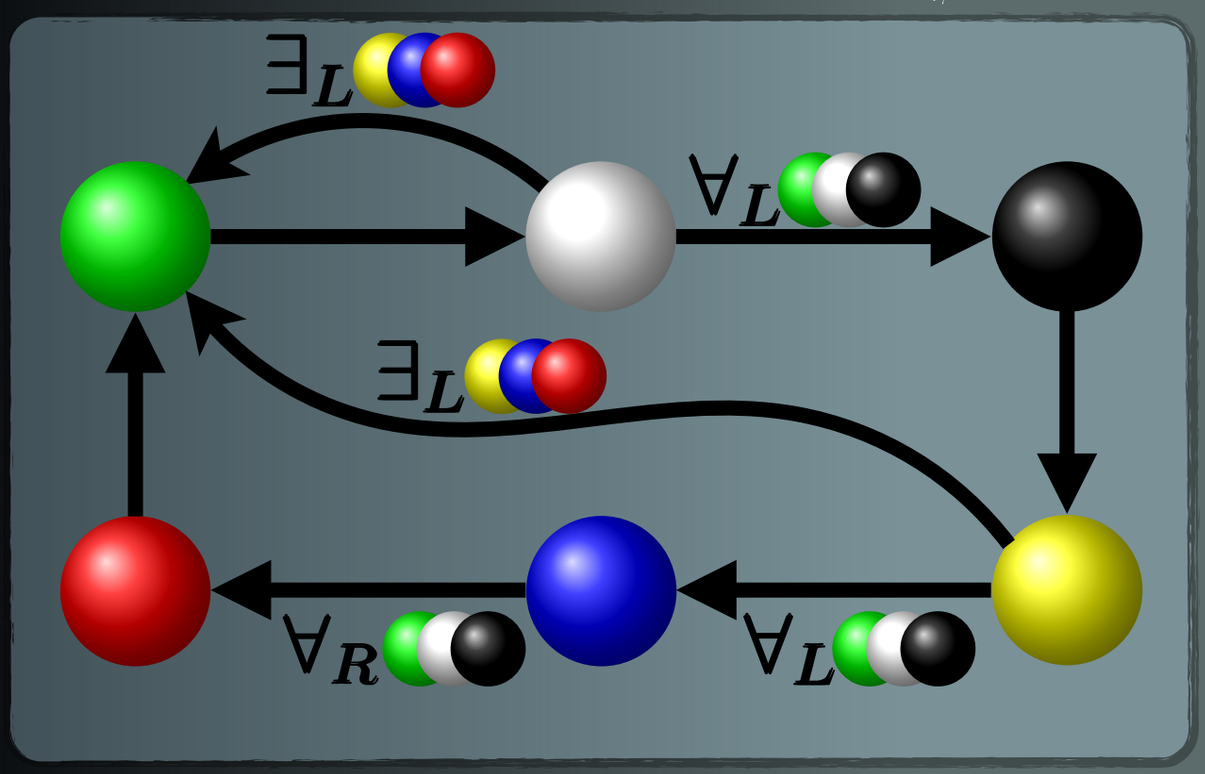
$$V_2 := \mu X. \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$



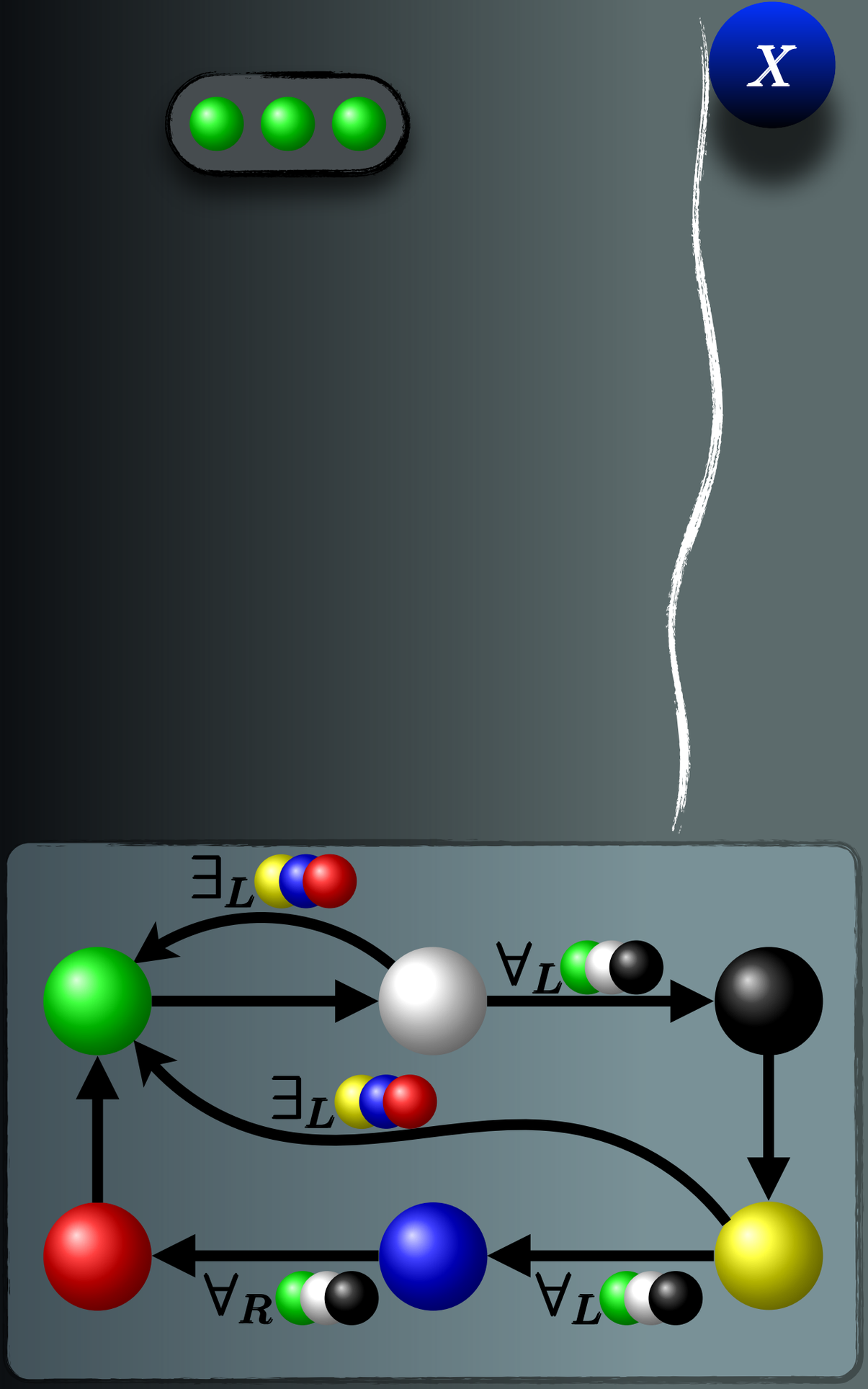
$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$



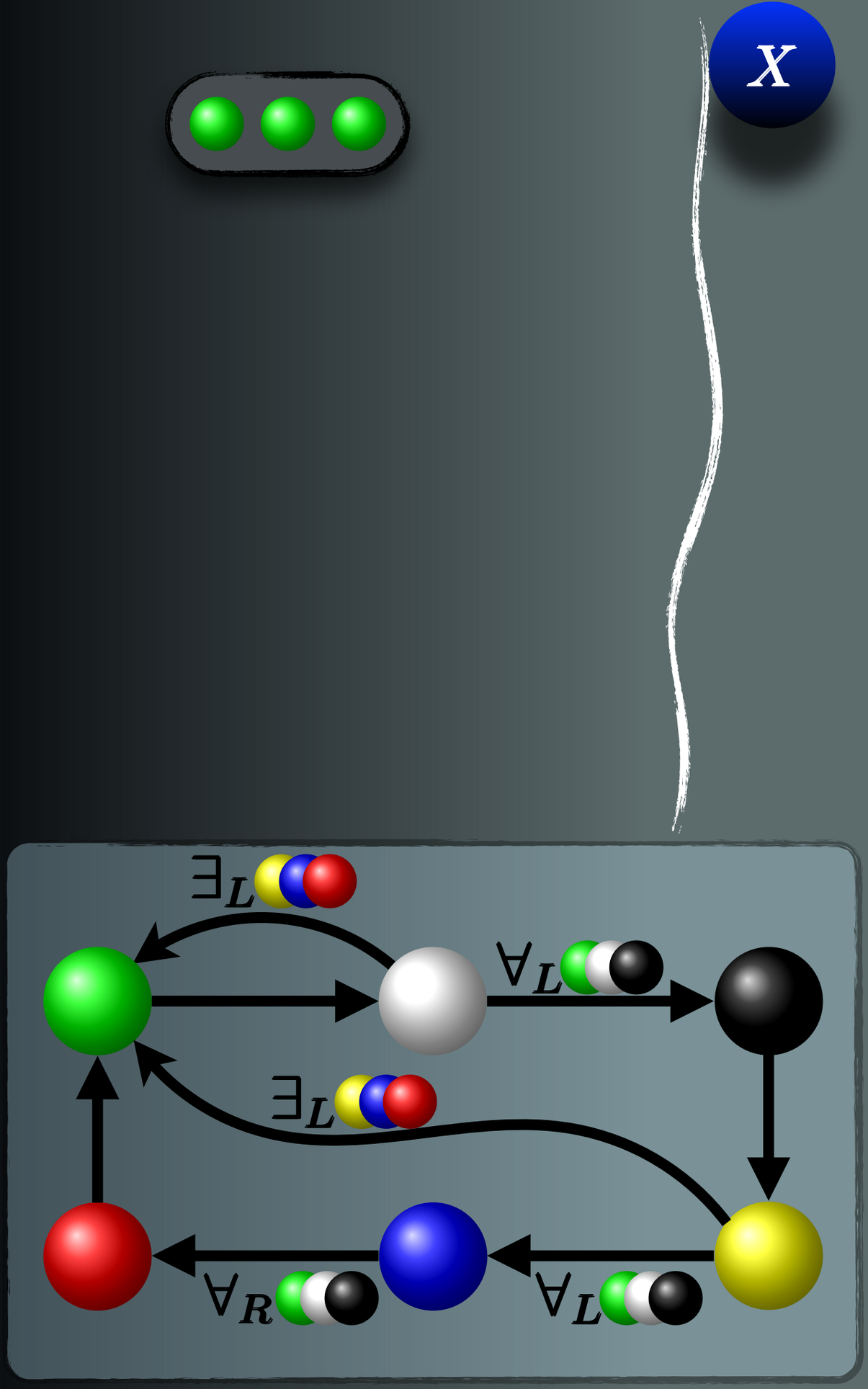
$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$



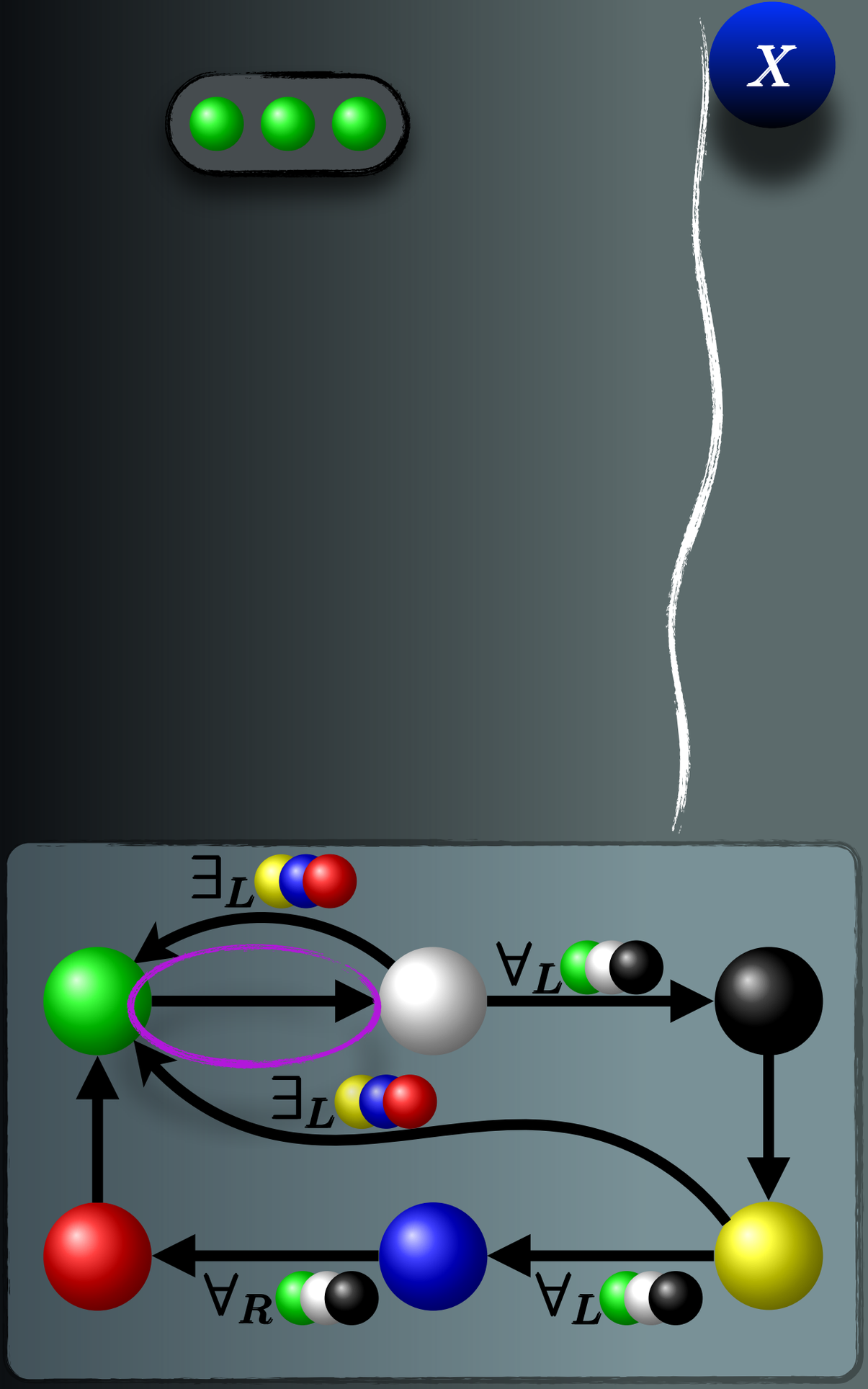
$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$



$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$



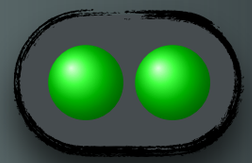
$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$



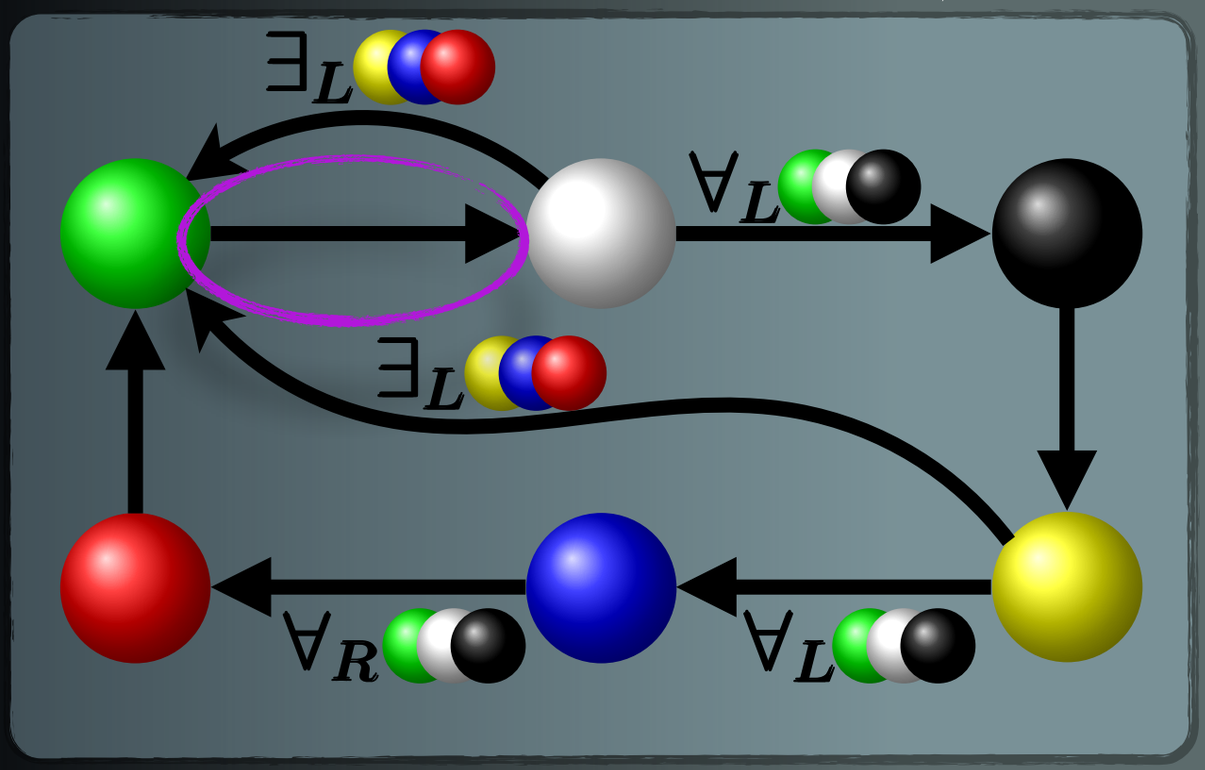
$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$







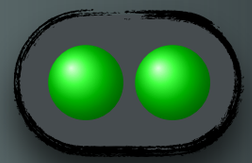




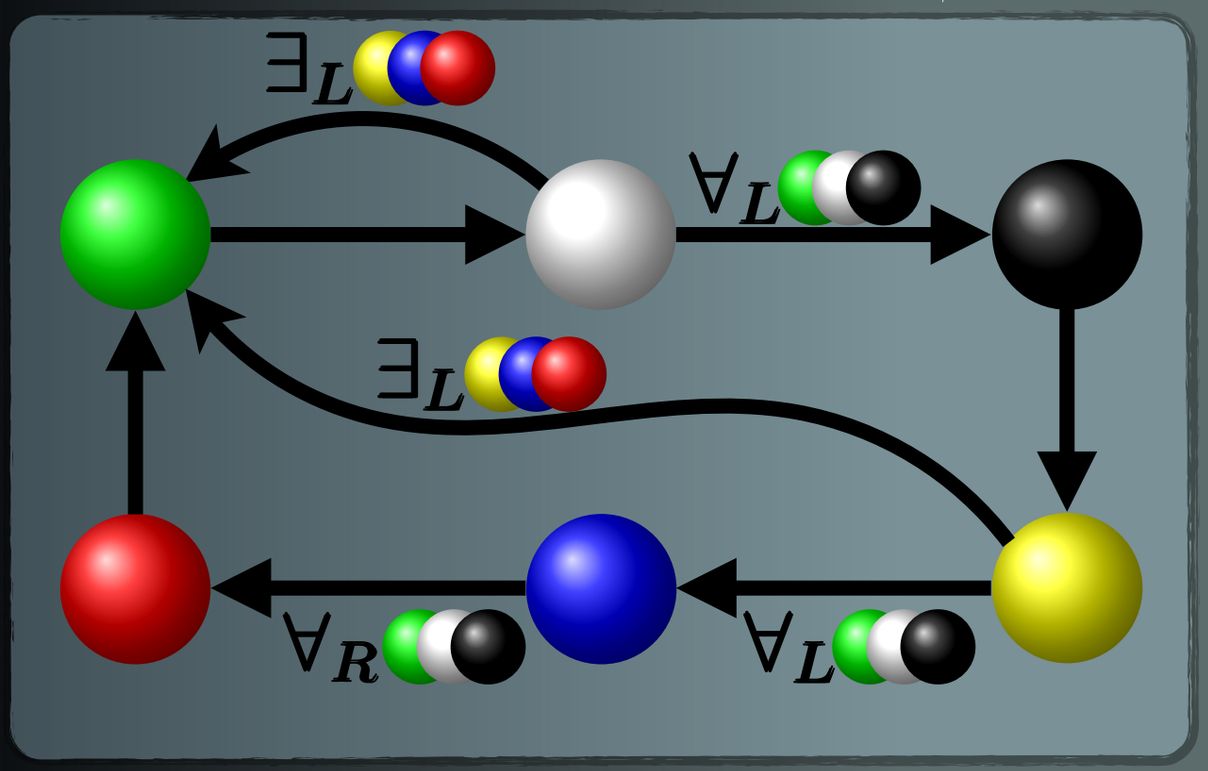
$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$







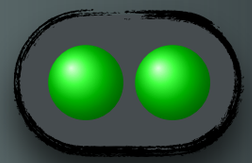




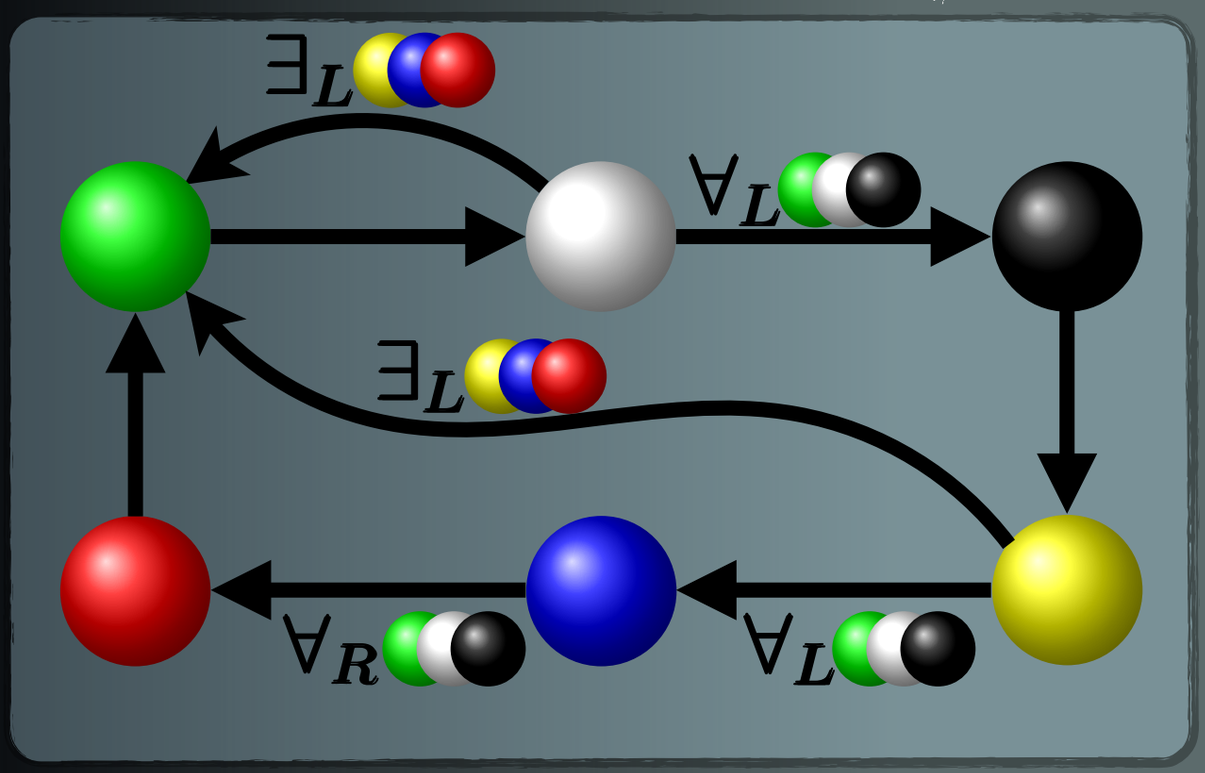
$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$



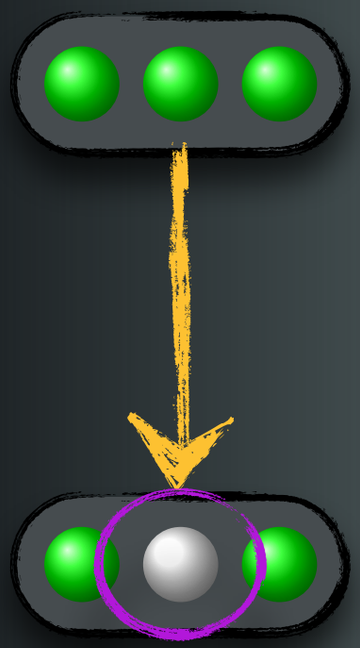




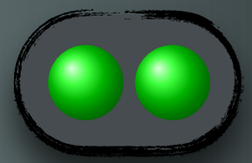




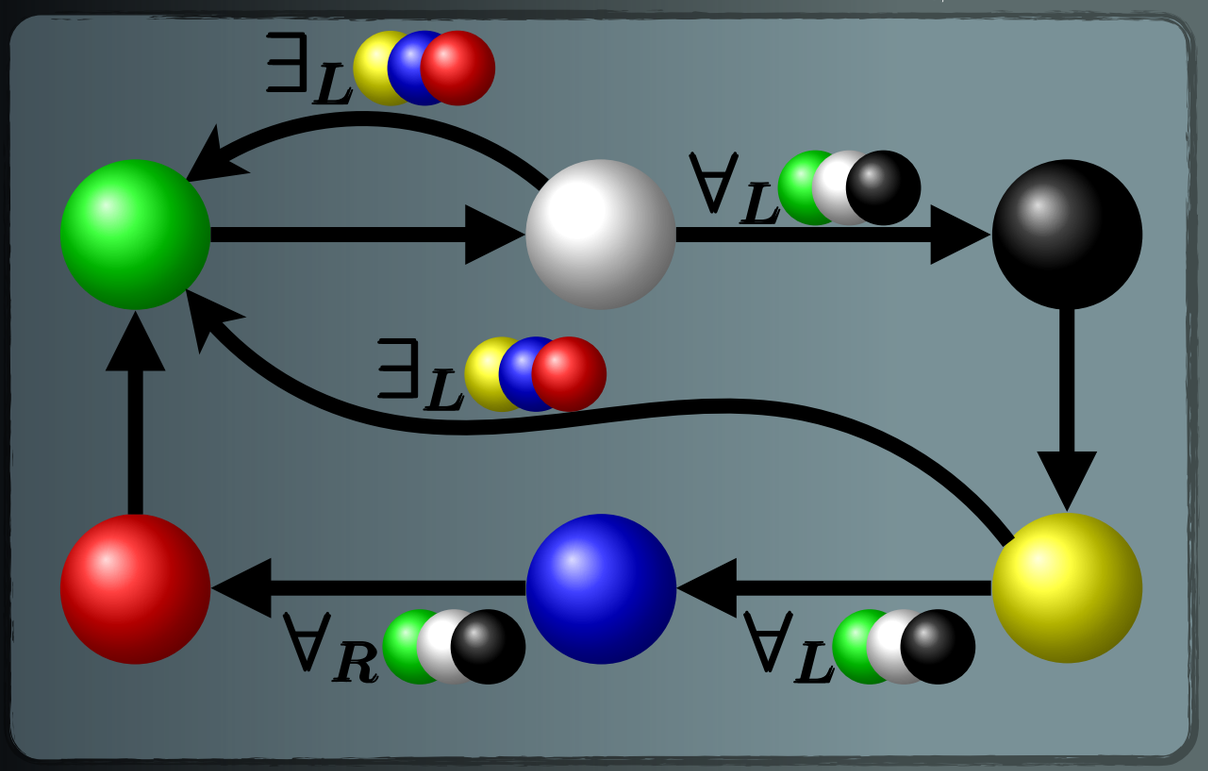
$$V_2 := \mu X.\ \alpha_2(I) \sqcup \alpha_2(post(\oint_2^3 (X)))$$



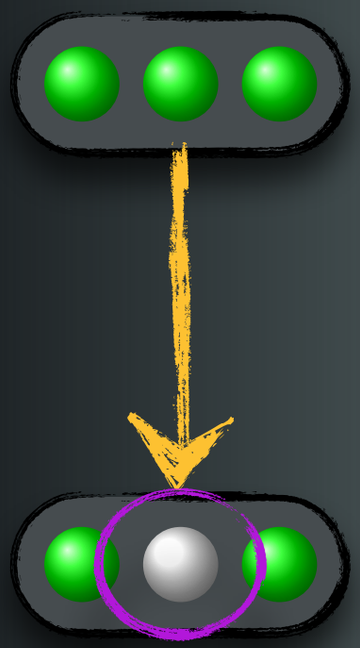




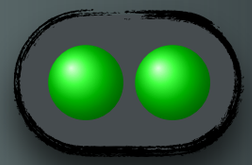


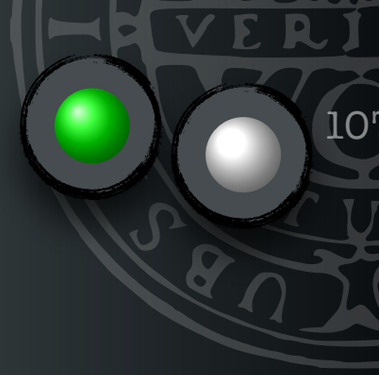


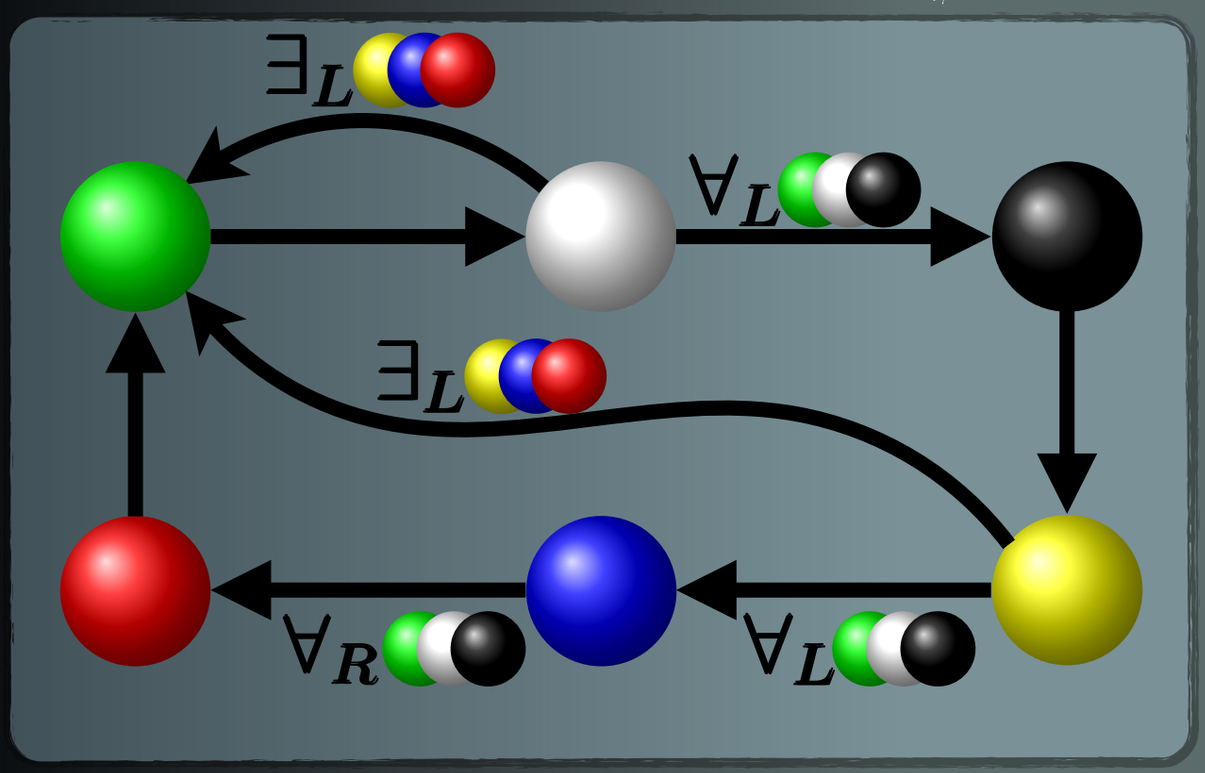
$$V_2 := \mu X.\ \alpha_2(I) \sqcup \alpha_2(post(\oint_2^3 (X)))$$



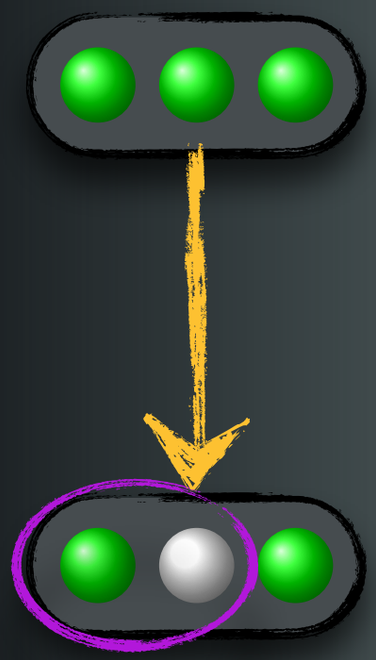




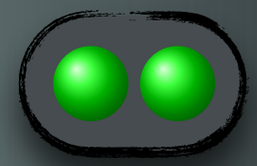


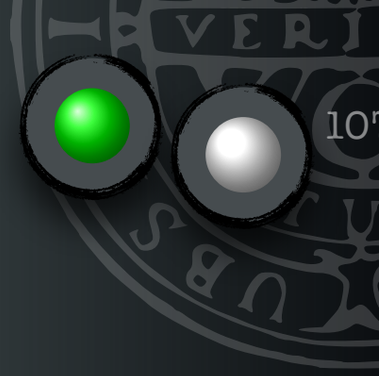


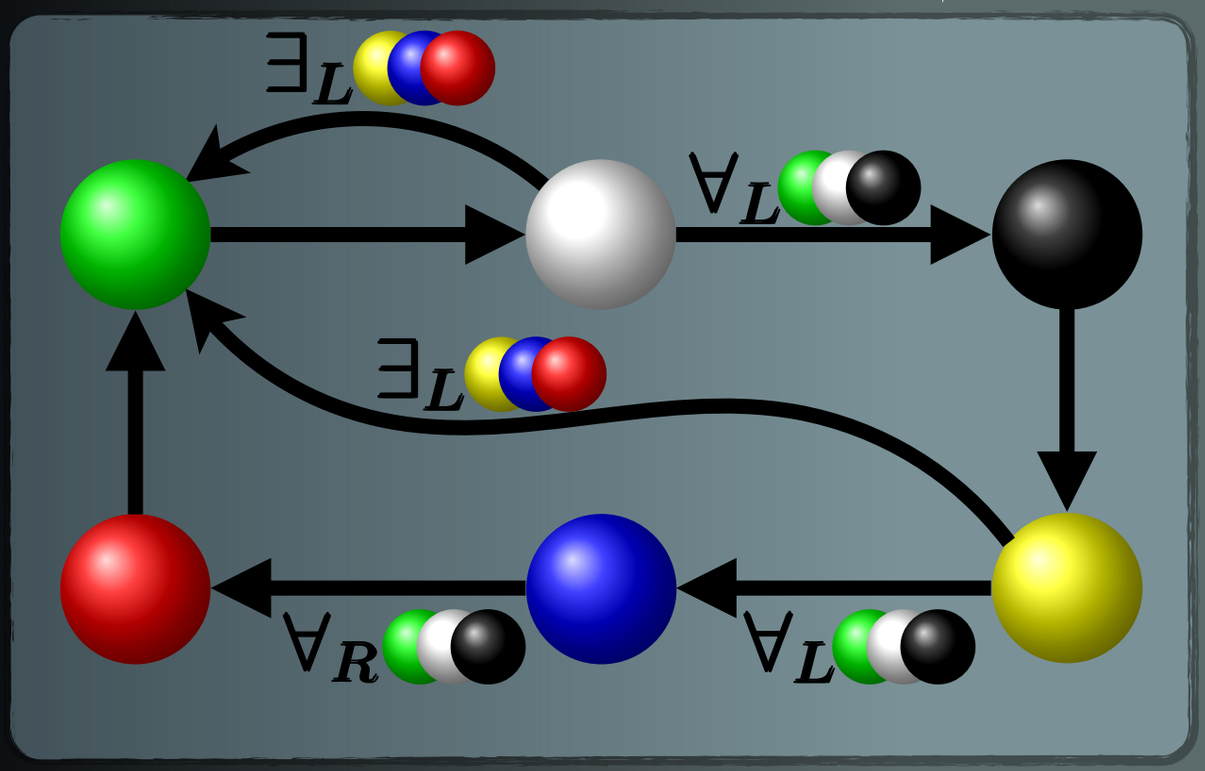
$$V_2 := \mu X.\ \alpha_2(I) \sqcup \alpha_2(post(\oint_2^3 (X)))$$



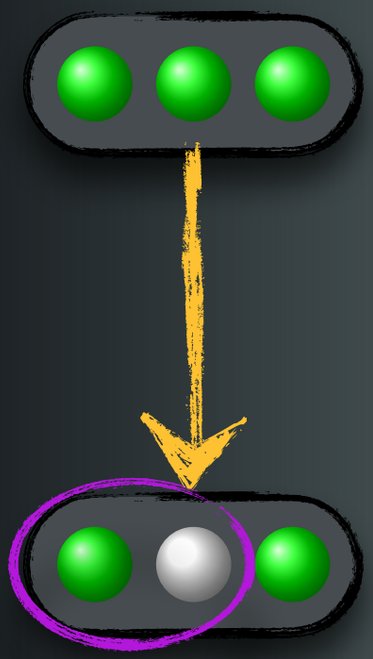




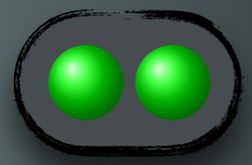


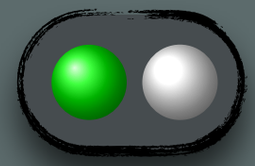


$$V_2 := \mu X.\ \alpha_2(I) \sqcup \alpha_2(post(\oint_2^3 (X)))$$

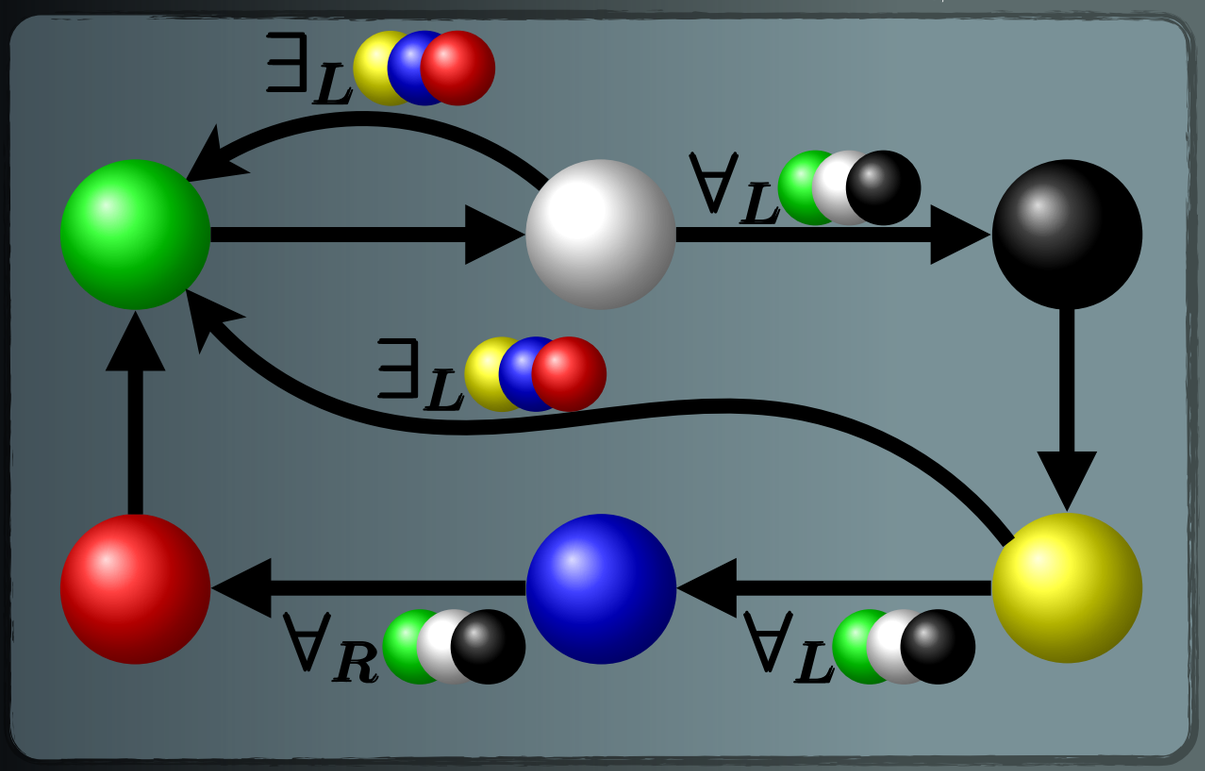








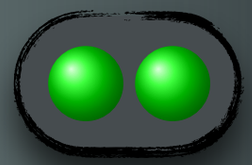


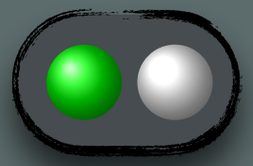


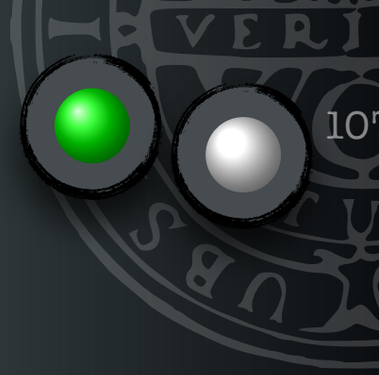
$$V_2 := \mu X.\ \alpha_2(I) \sqcup \alpha_2(post(\oint_2^3 (X)))$$

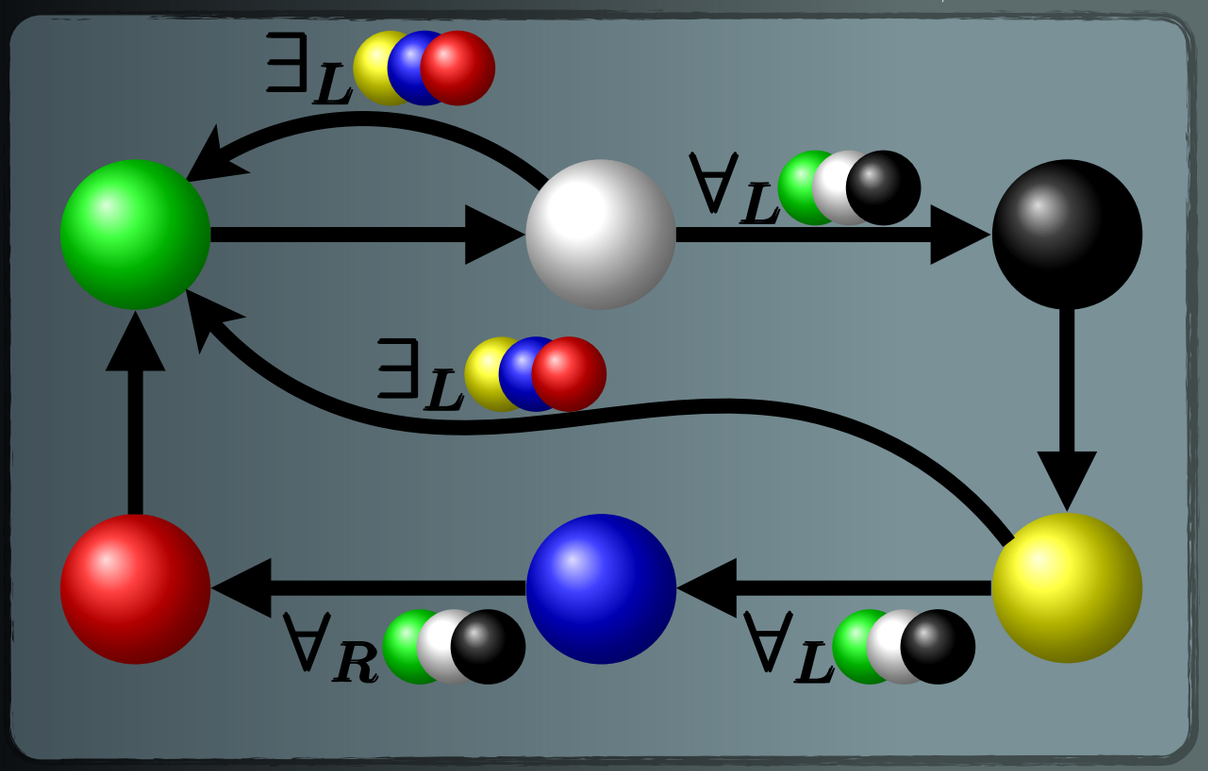








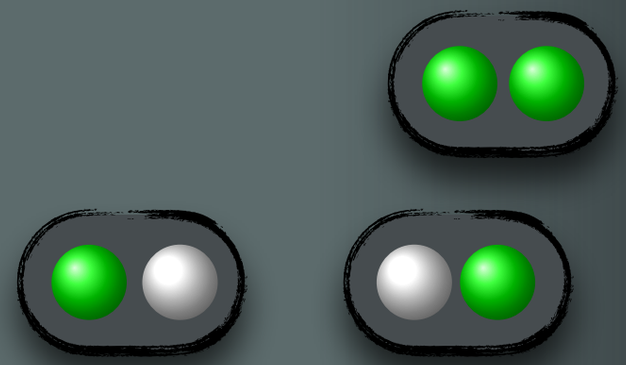


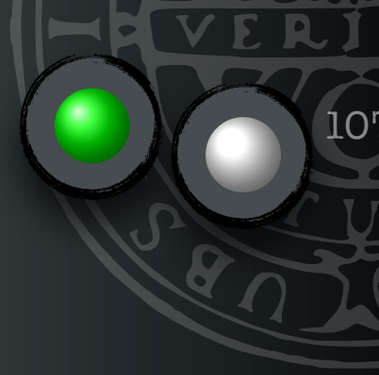


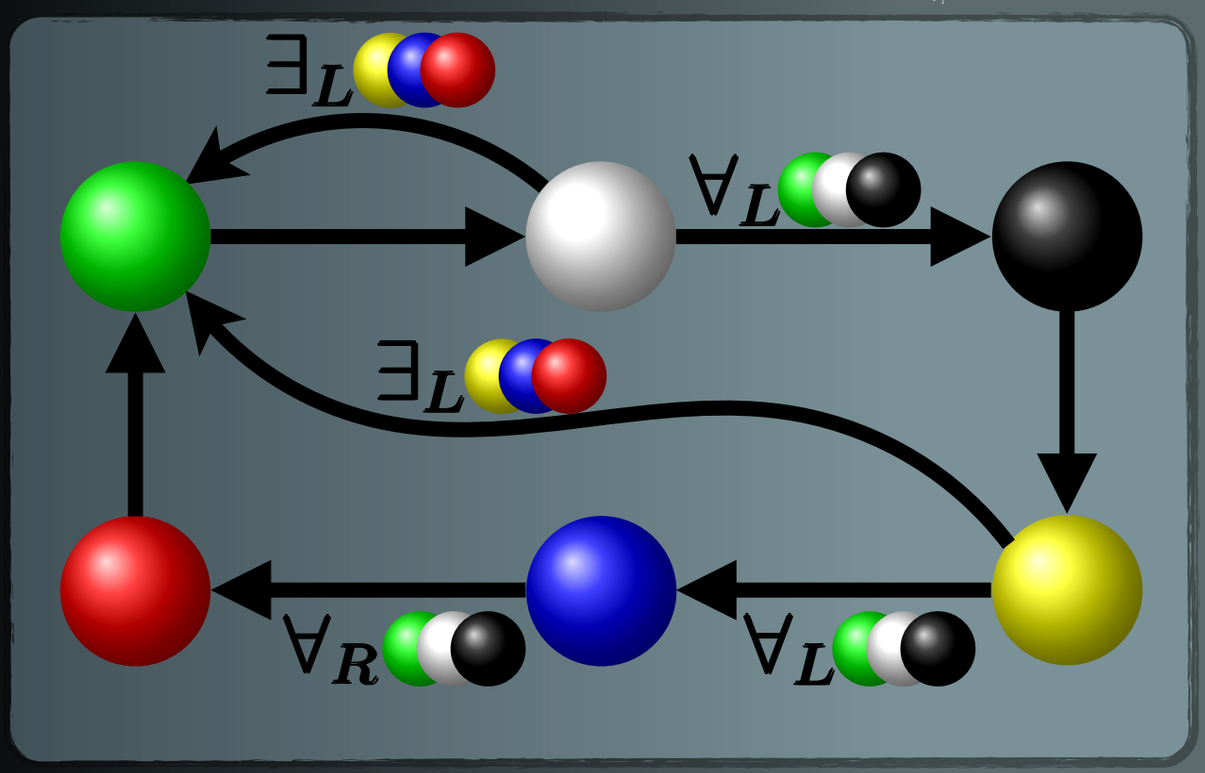
$$V_2 := \mu X.\ \alpha_2(I) \sqcup \alpha_2(post(\oint_2^3 (X)))$$



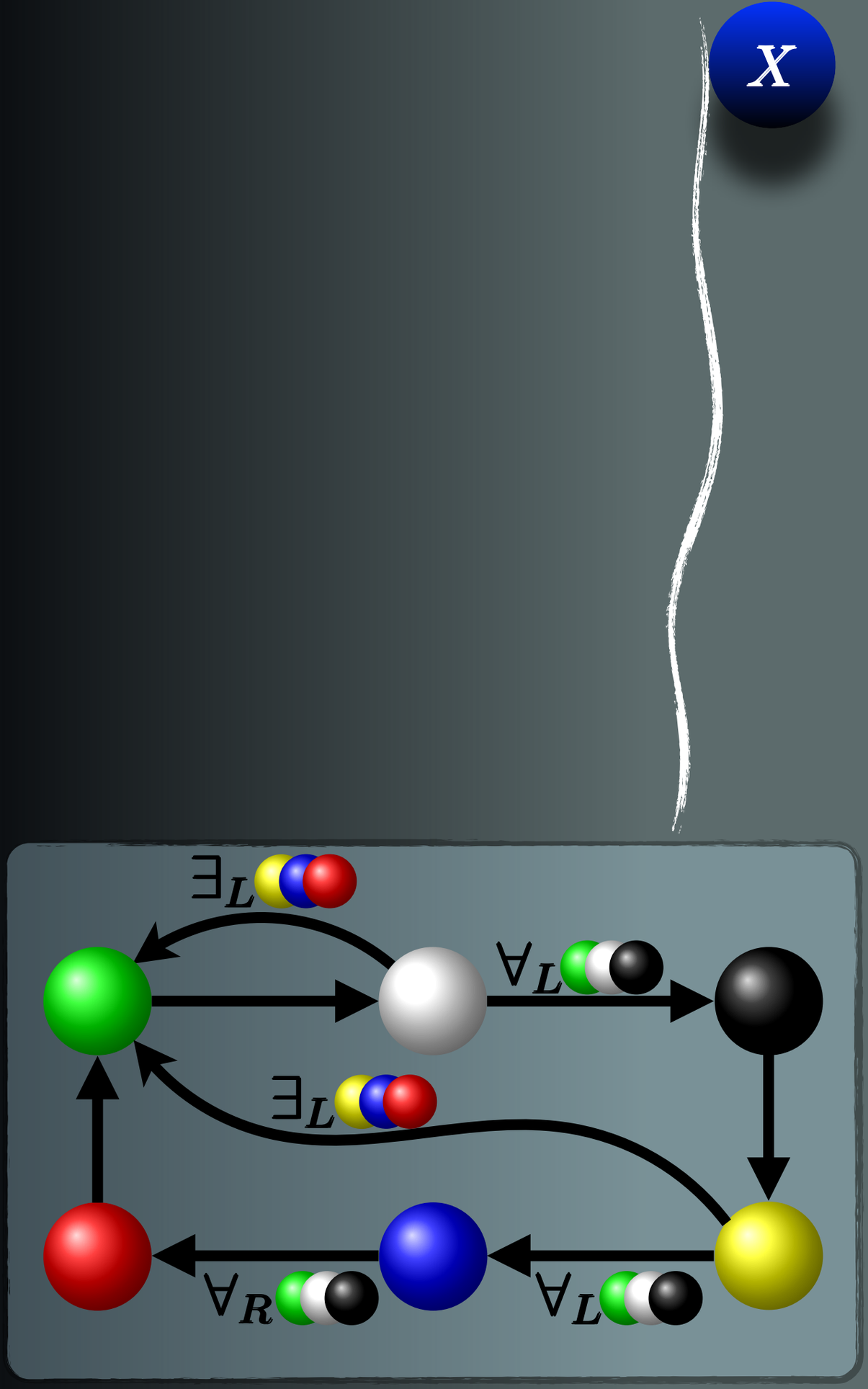




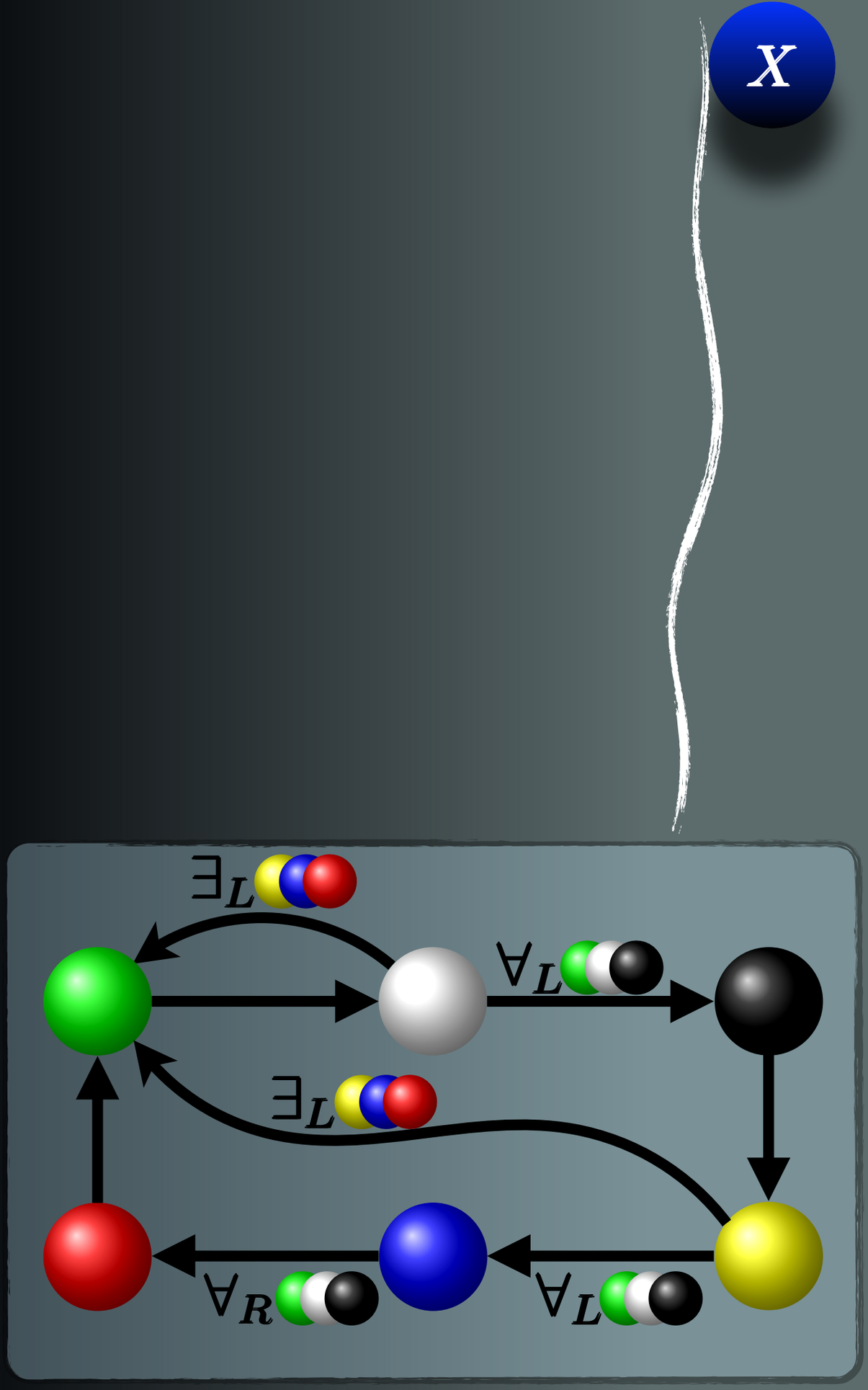




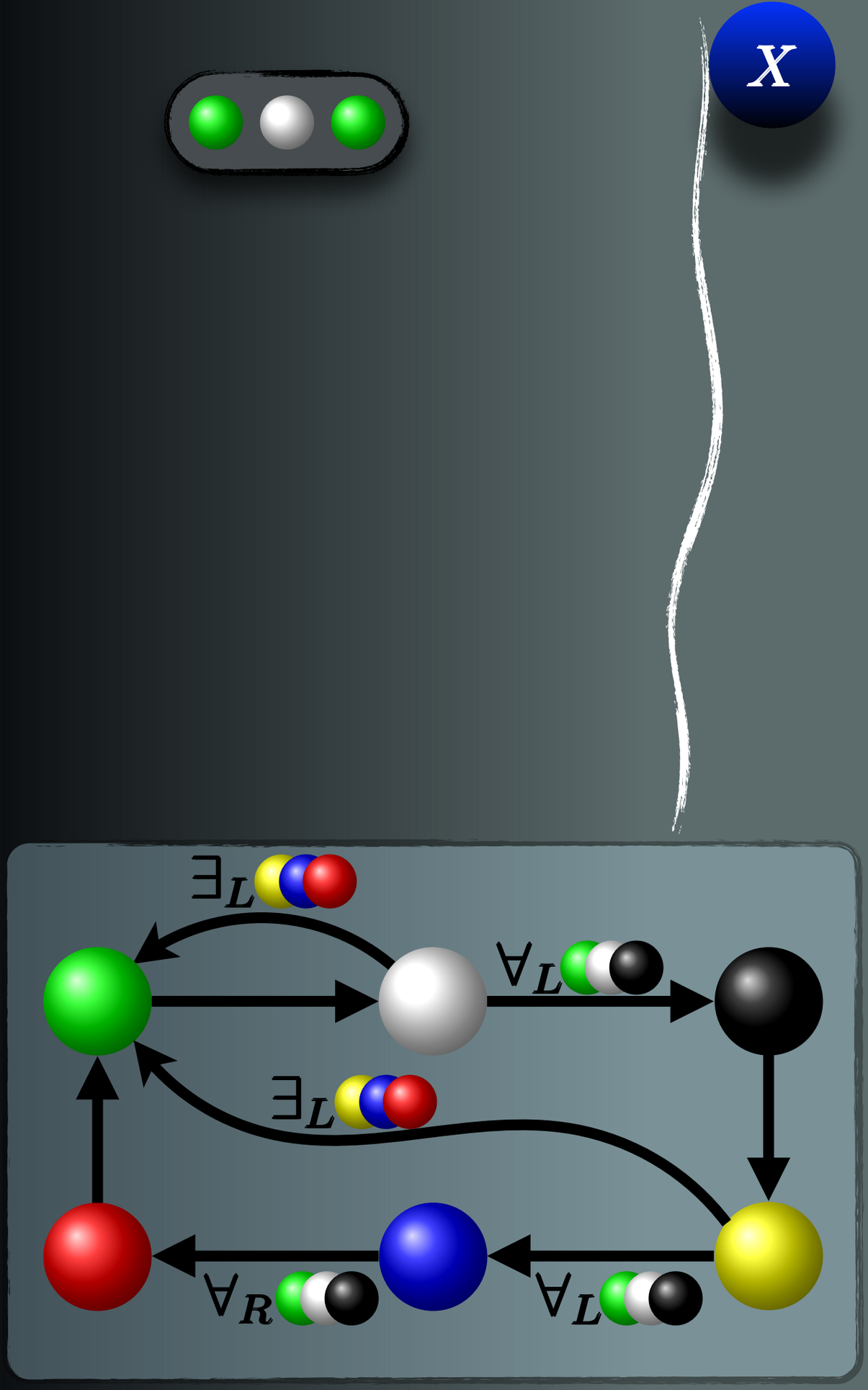
$$V_2 := \mu X.\ \alpha_2(I) \sqcup \alpha_2(post(\oint_2^3 (X)))$$



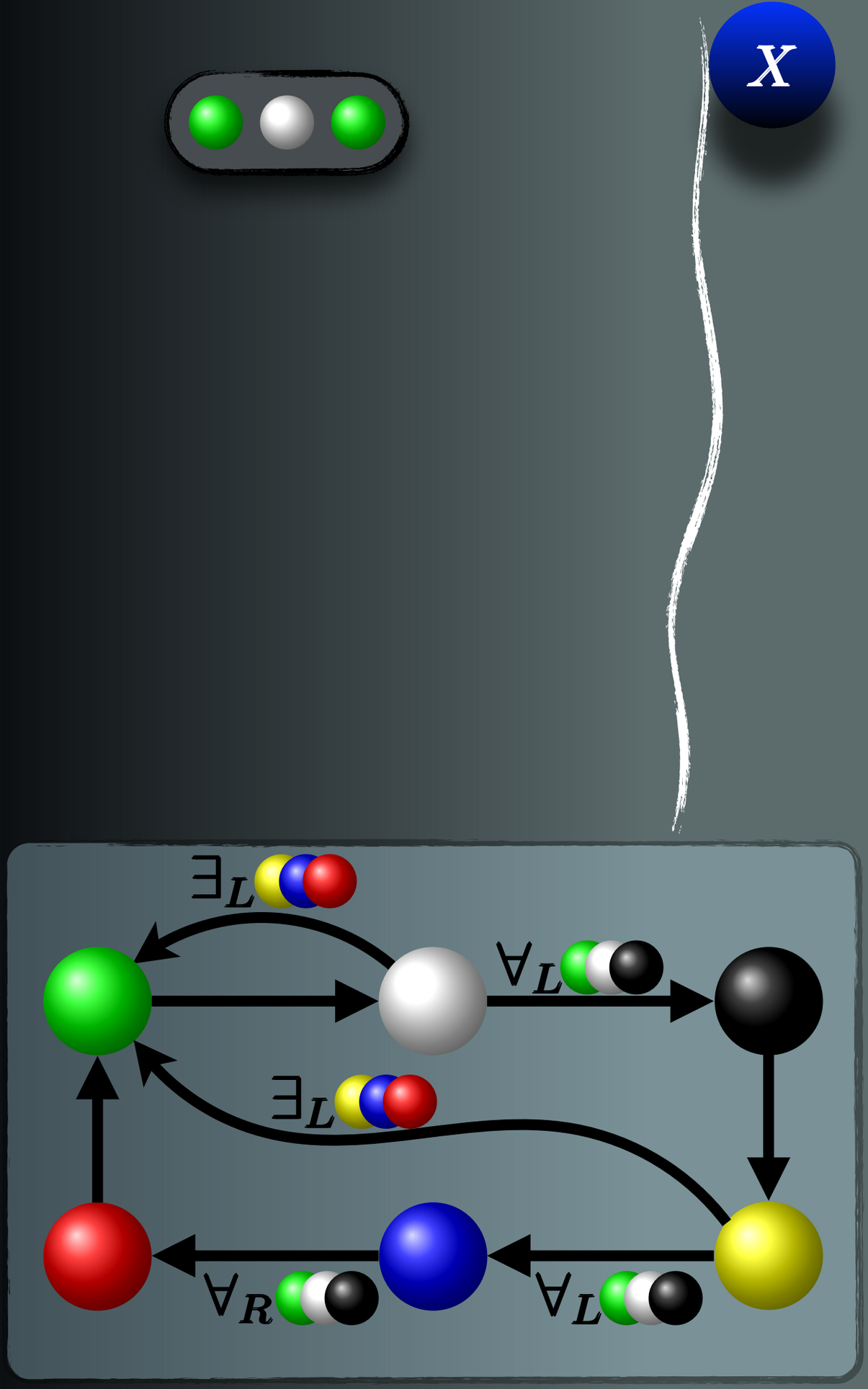
$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$



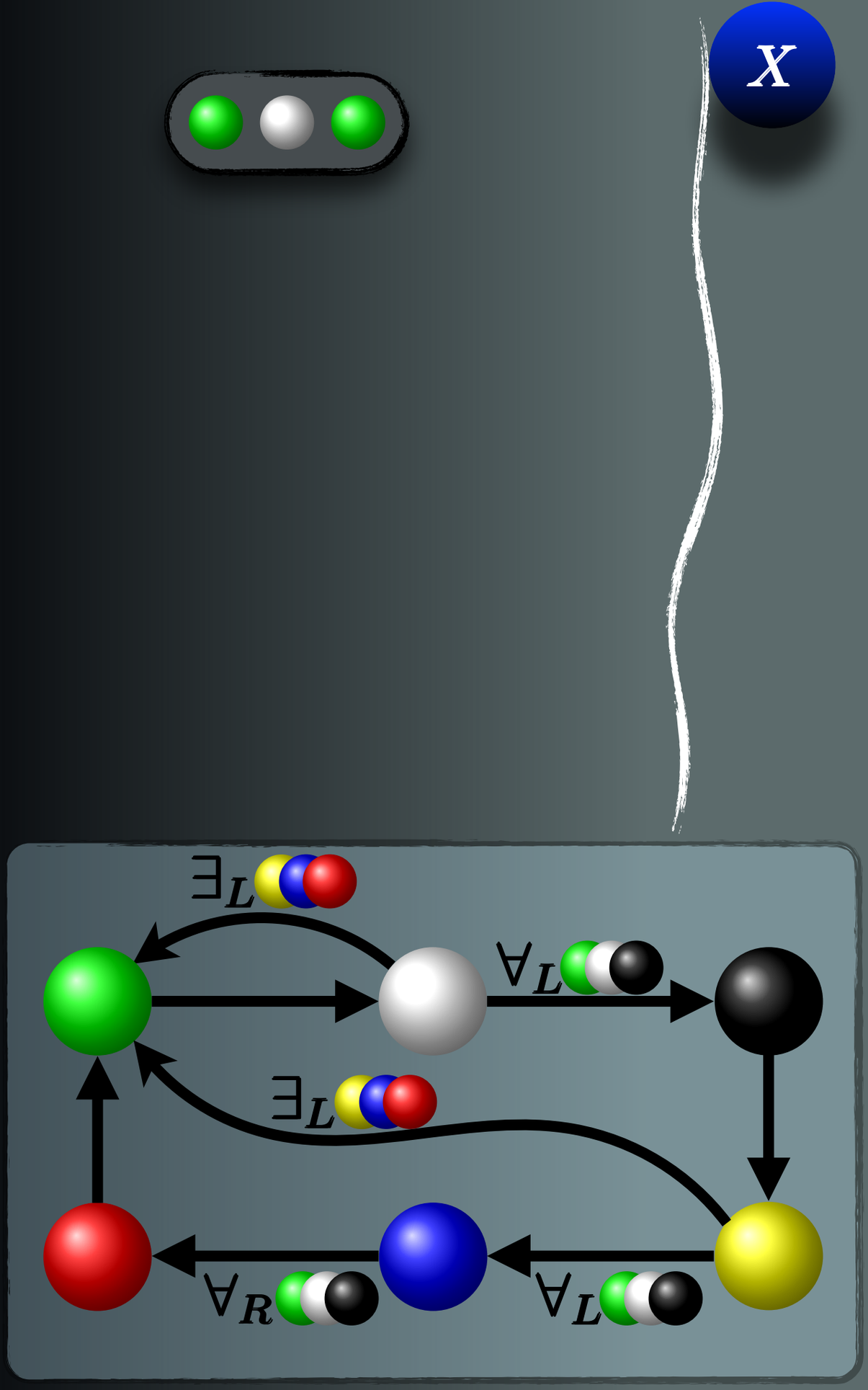
$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$



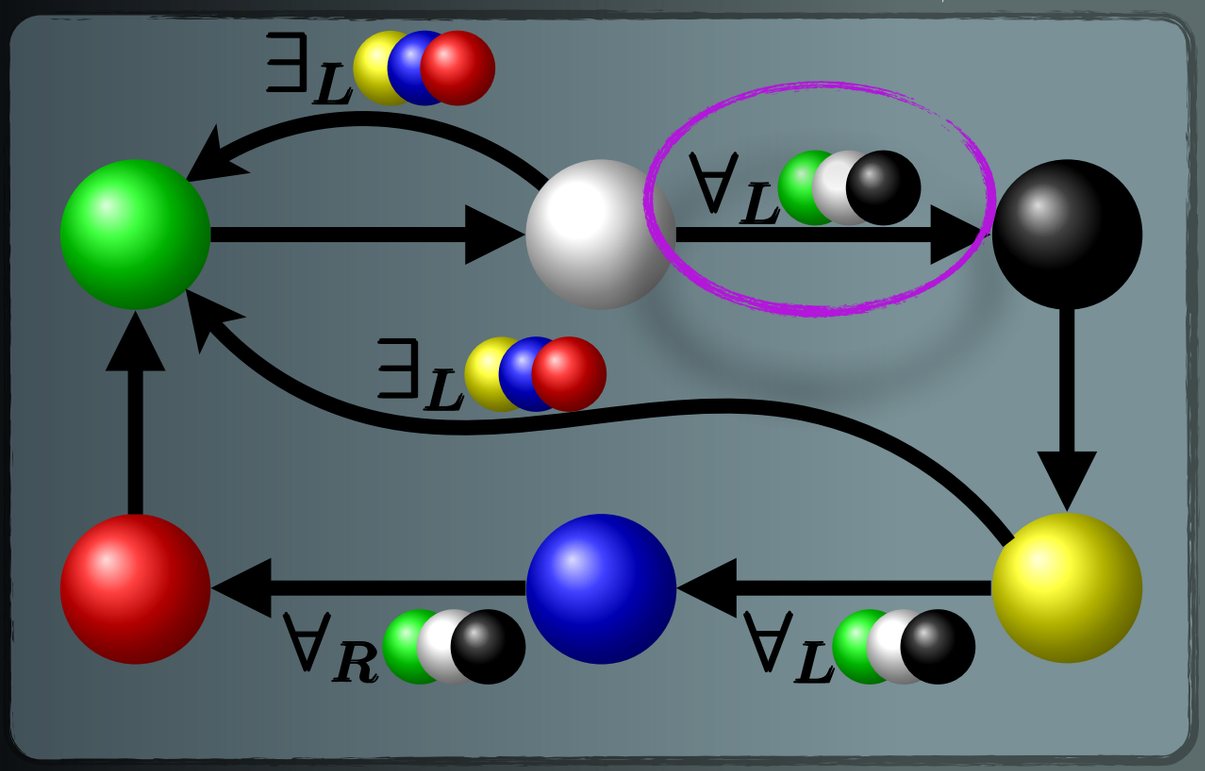
$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$



$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$



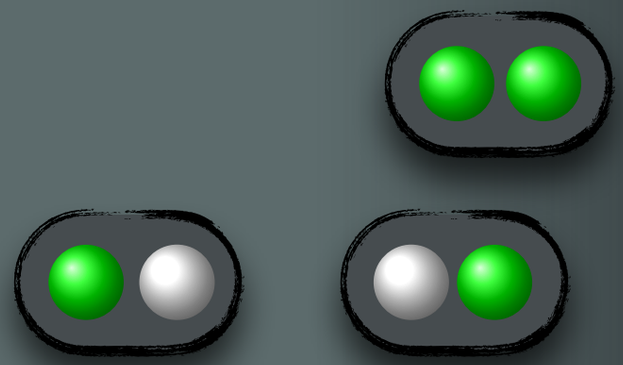
$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$

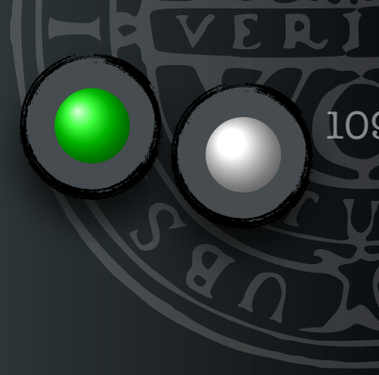


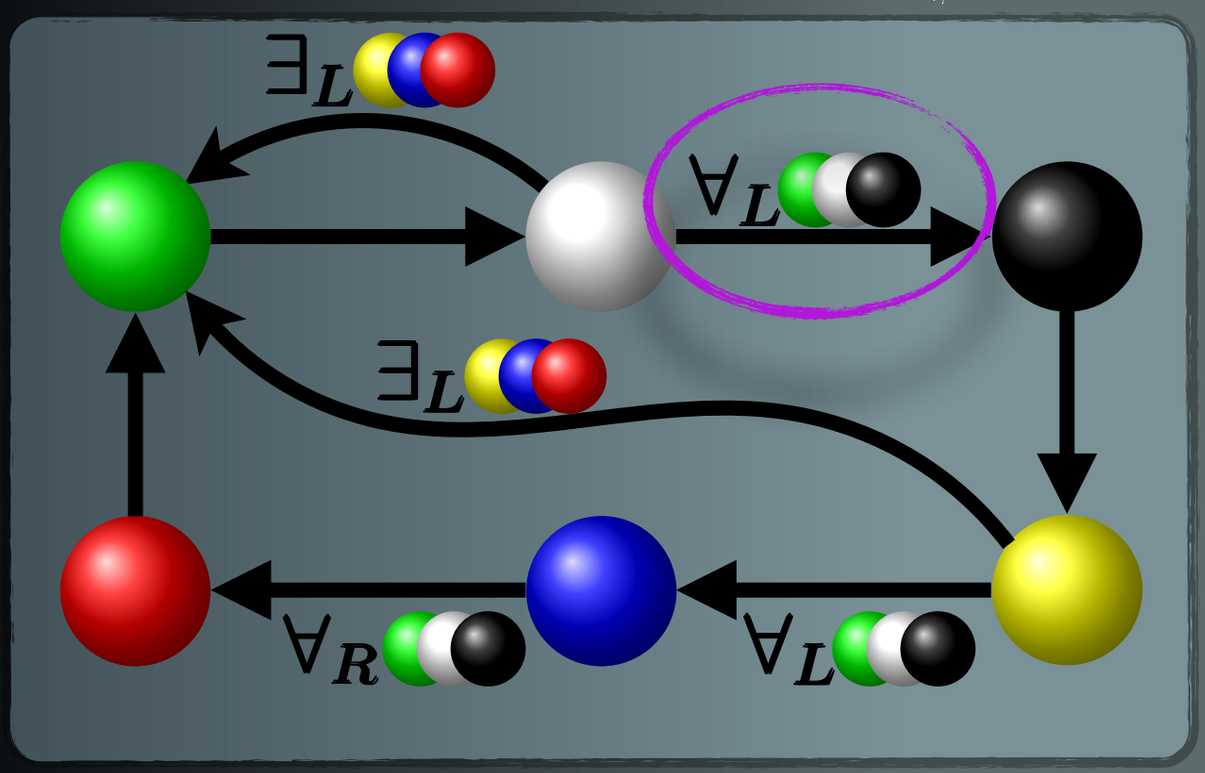
$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\int_2^3 (X)))$$







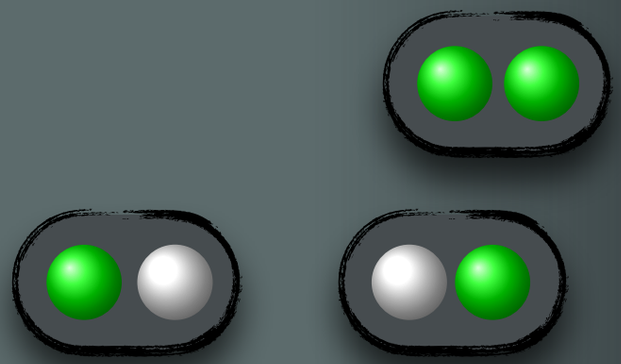




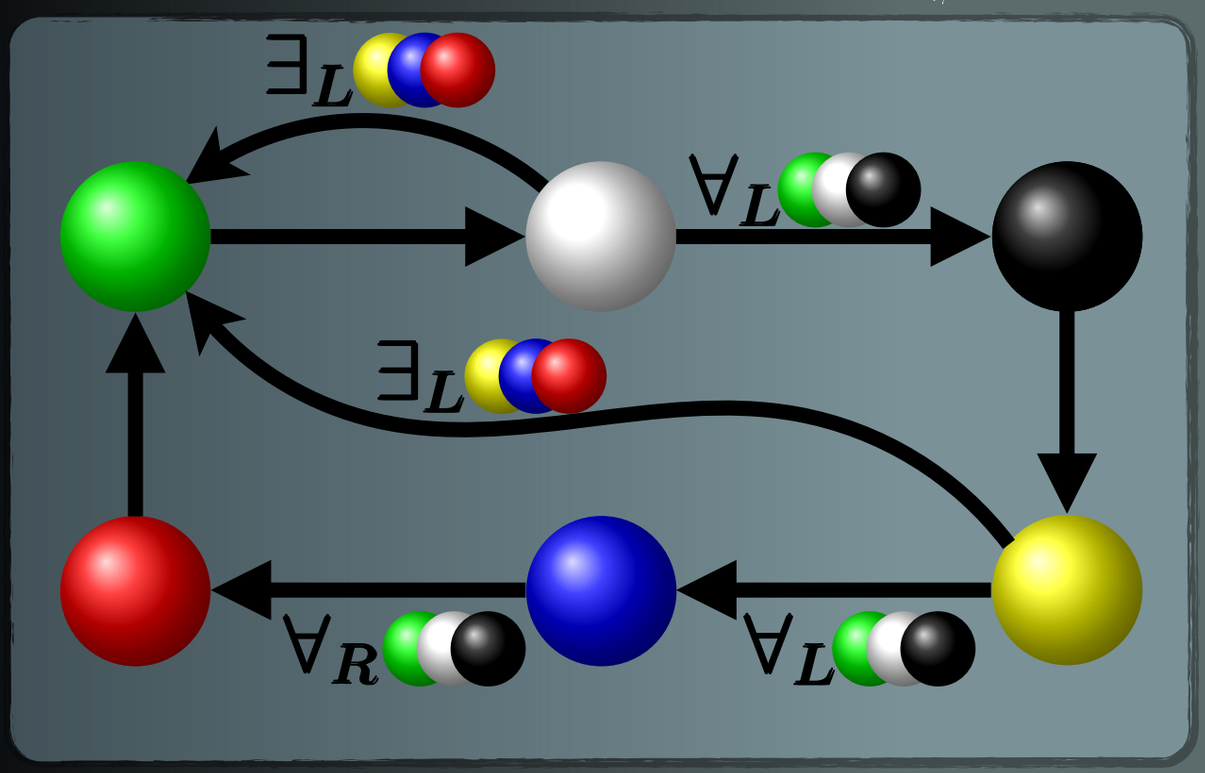
$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\int_2^3(X)))$$







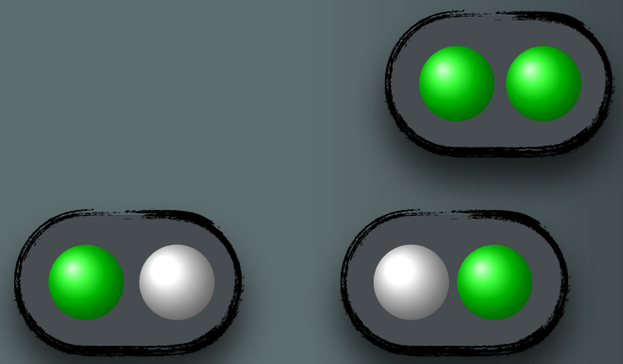




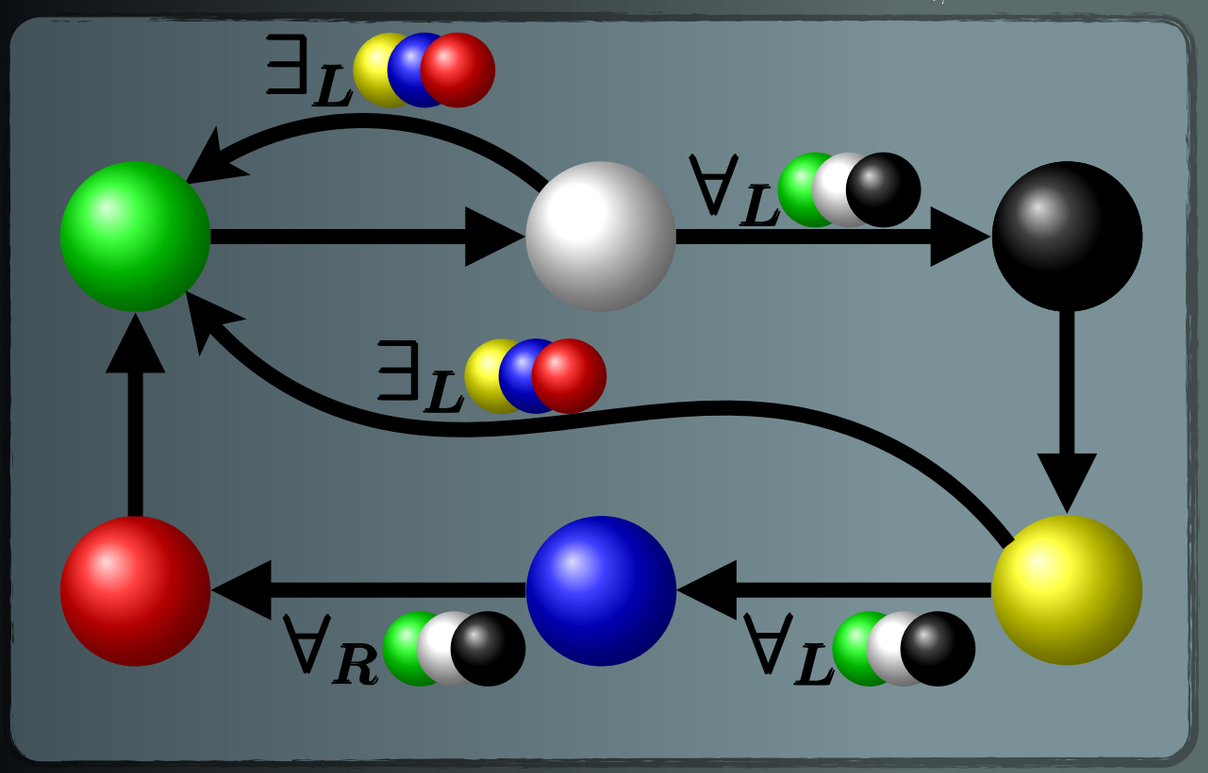
$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$



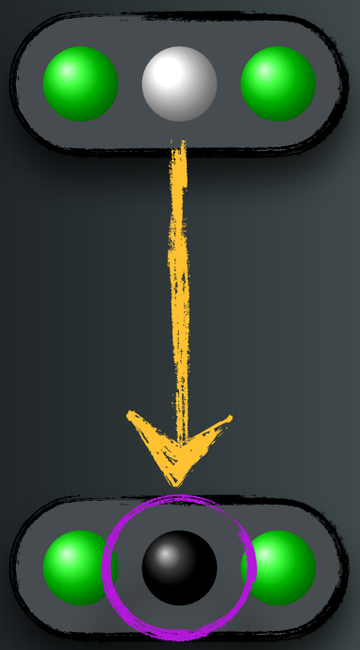




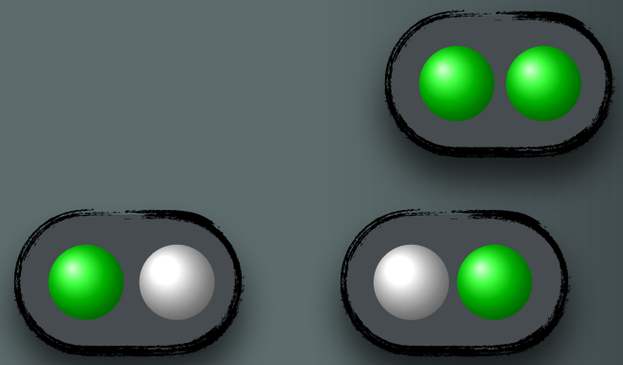




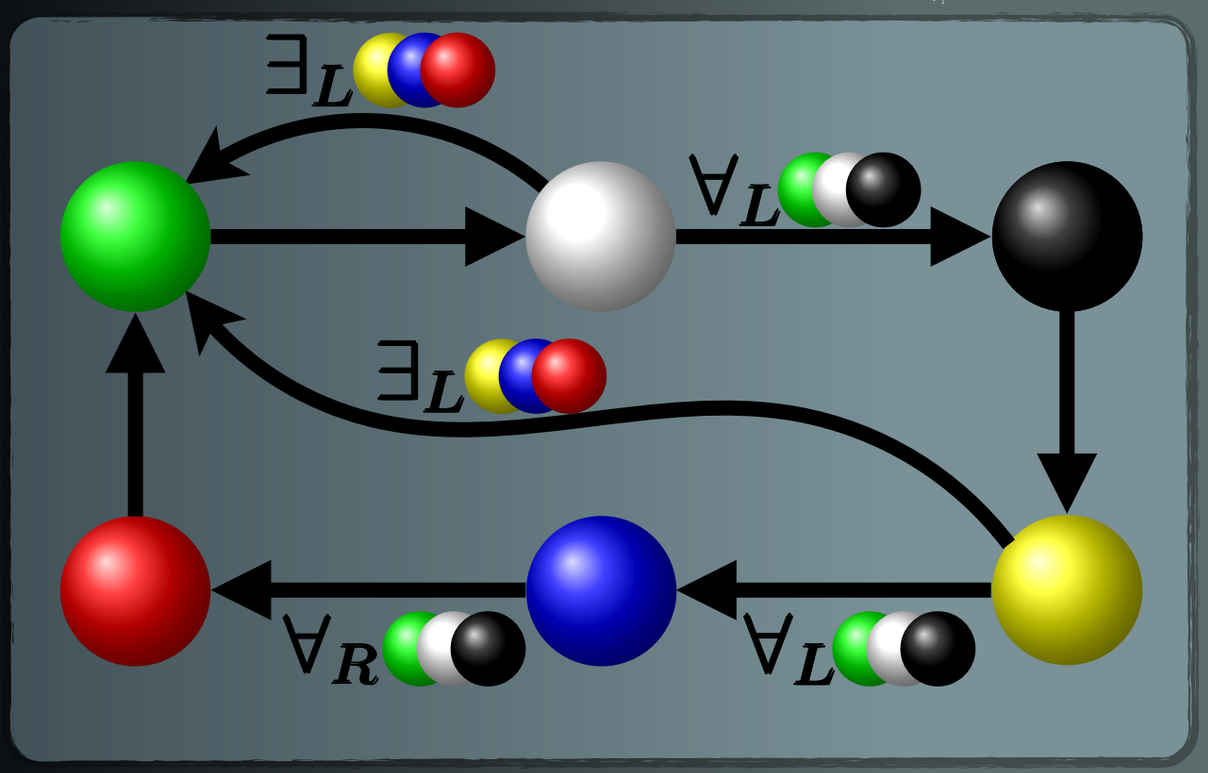
$$V_2 := \mu X.\ \alpha_2(I) \sqcup \alpha_2(post(\oint_2^3 (X)))$$



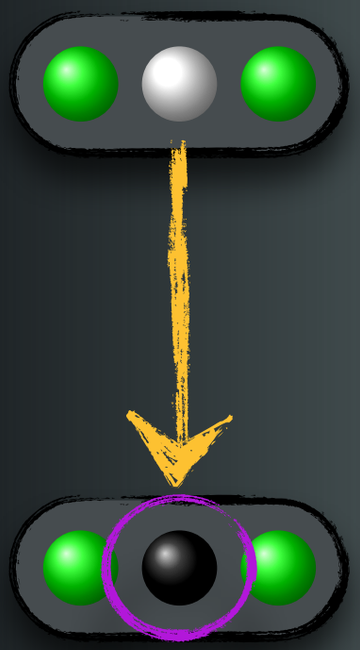




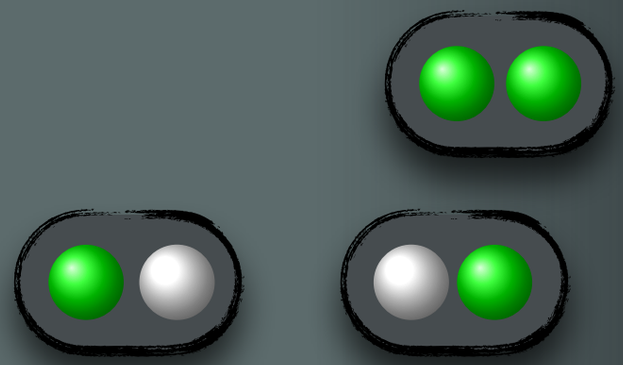




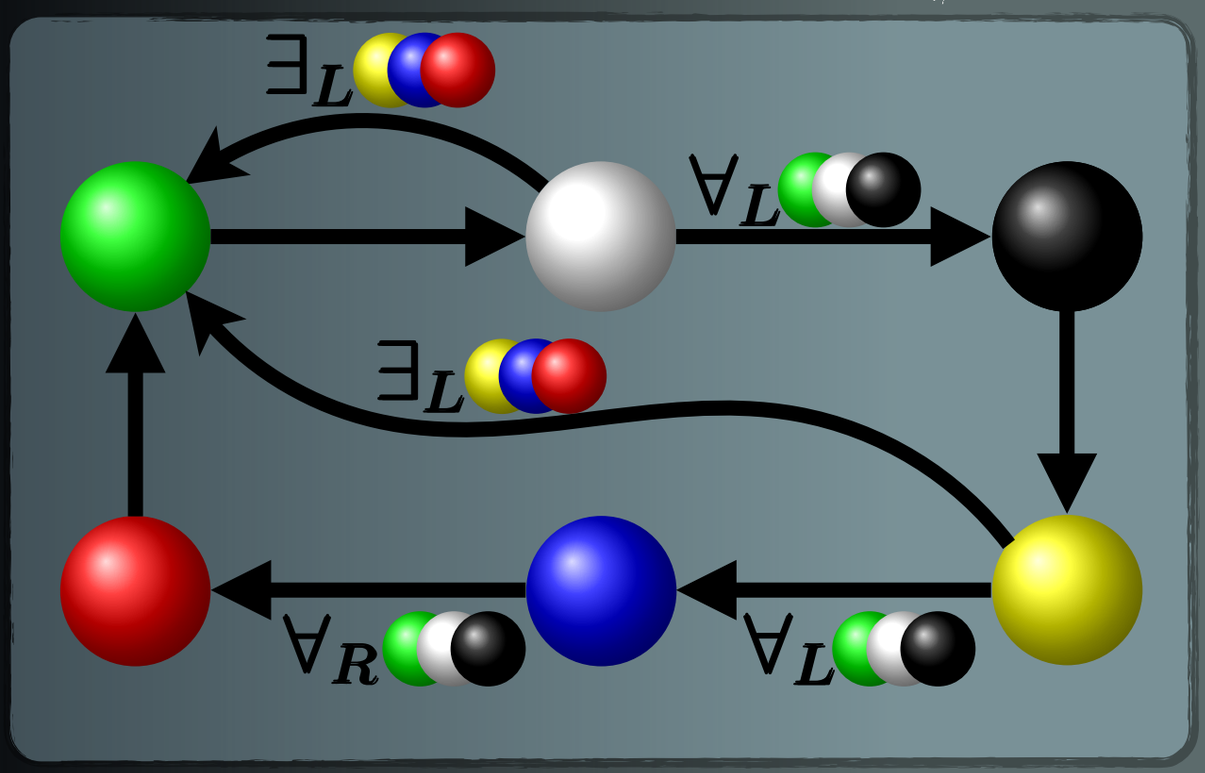
$$V_2 := \mu X.\ \alpha_2(I) \sqcup \alpha_2(post(\oint_2^3 (X)))$$



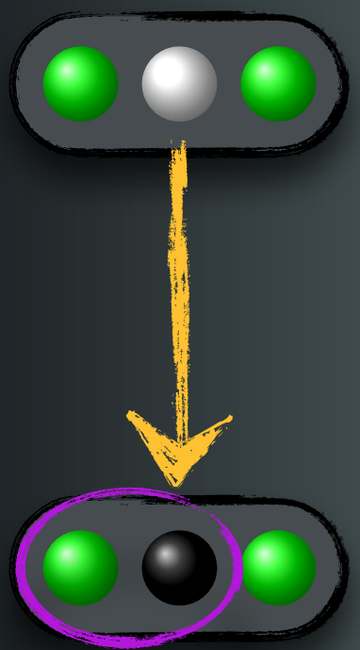




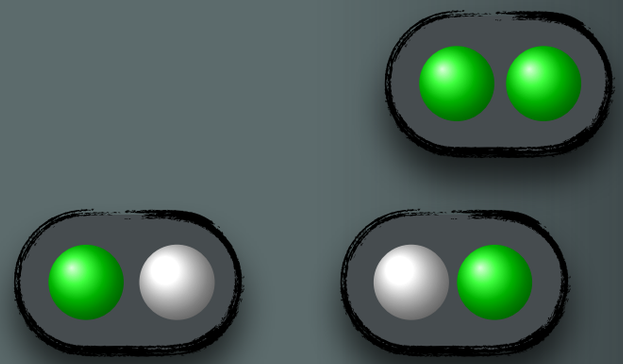




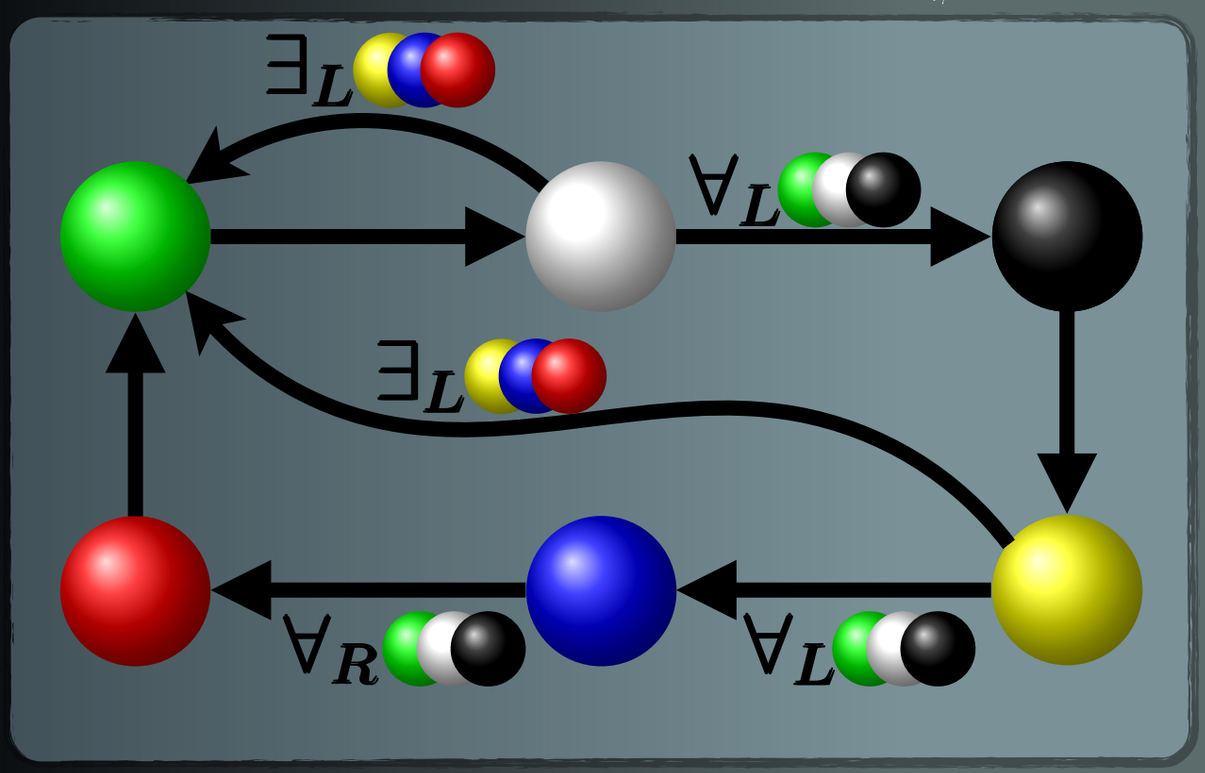
$$V_2 := \mu X.\ \alpha_2(I) \sqcup \alpha_2(post(\oint_2^3 (X)))$$



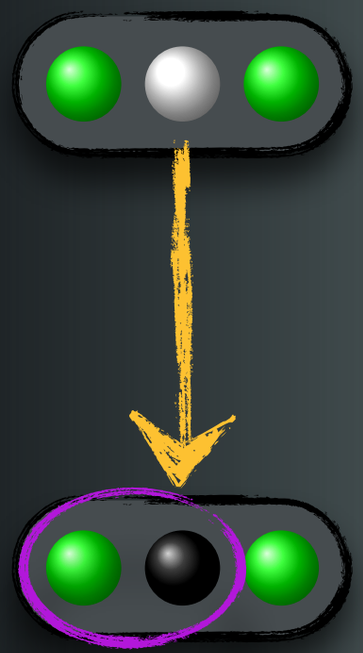




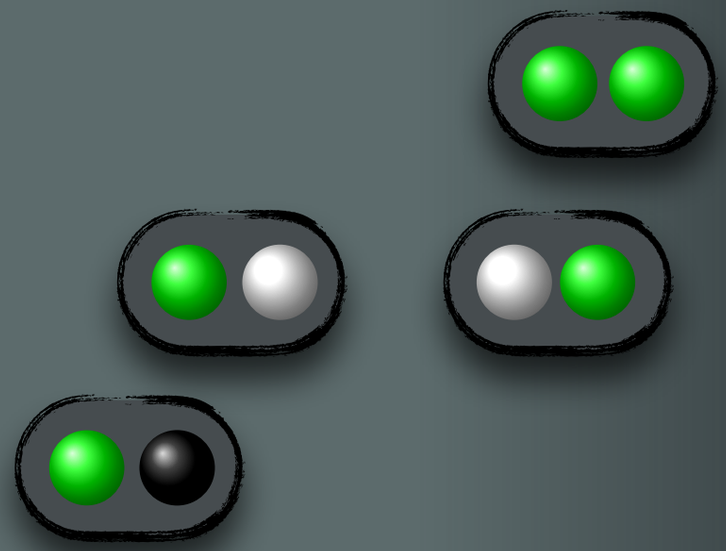




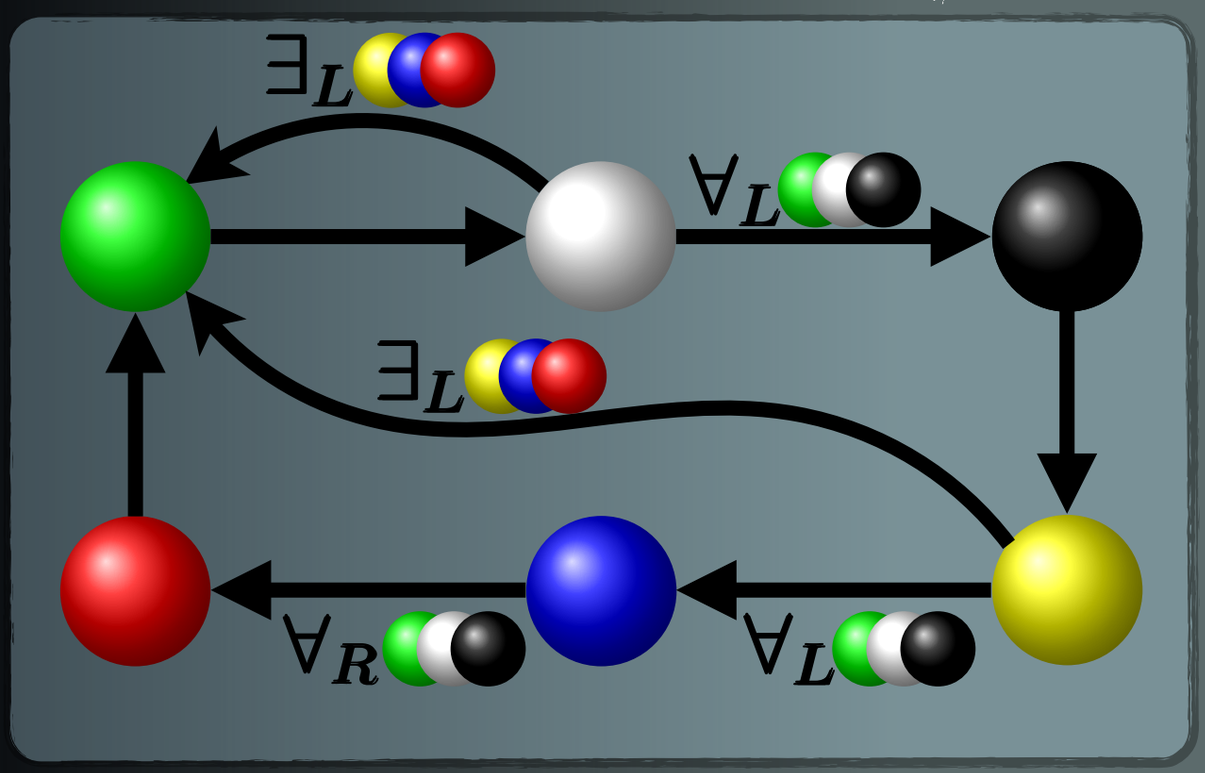
$$V_2 := \mu X.\ \alpha_2(I) \sqcup \alpha_2(post(\oint_2^3 (X)))$$







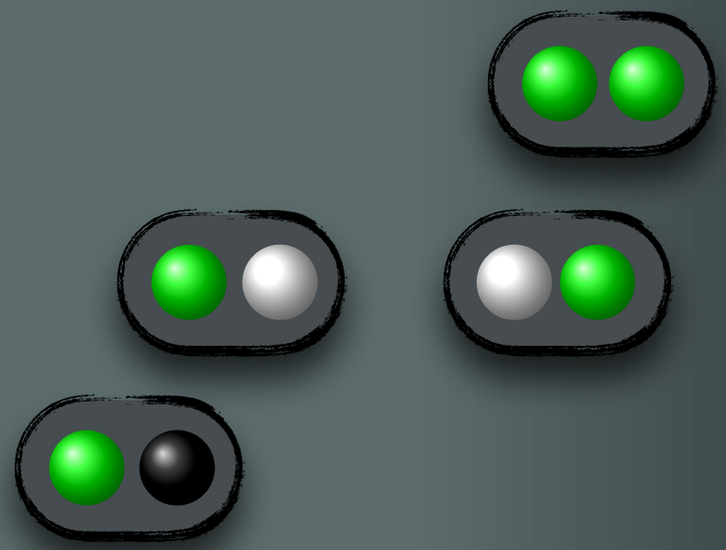




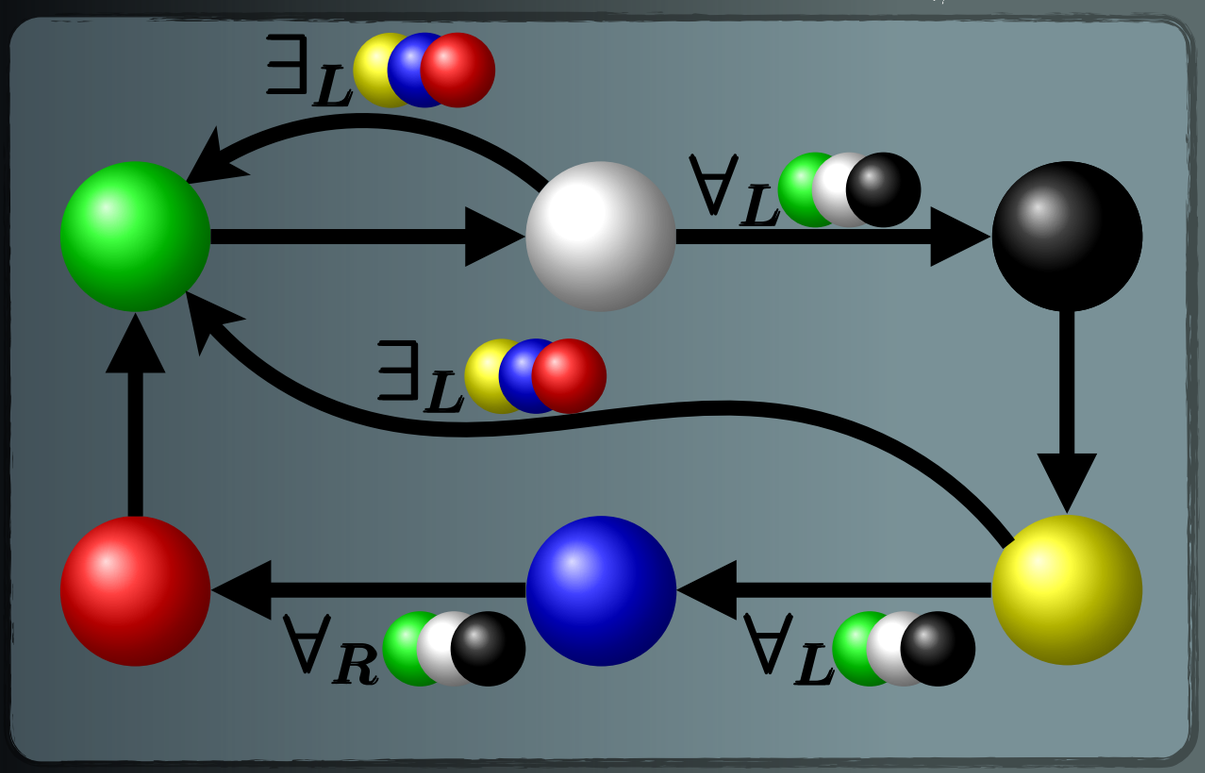
$$V_2 := \mu X.\ \alpha_2(I) \sqcup \alpha_2(post(\oint_2^3 (X)))$$







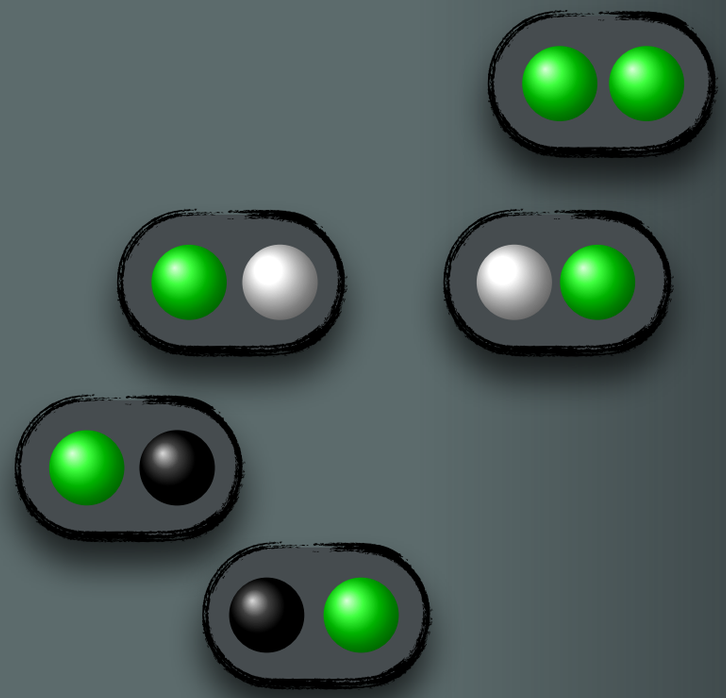




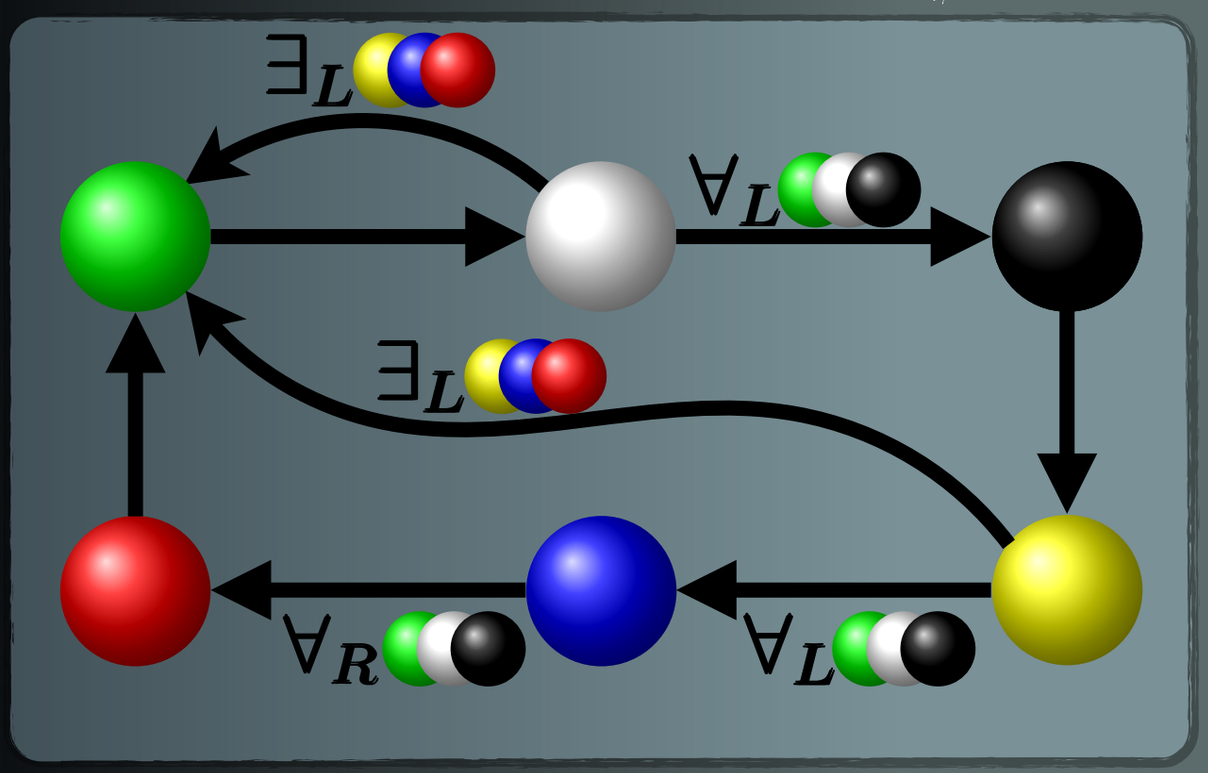
$$V_2 := \mu X.\ \alpha_2(I) \sqcup \alpha_2(post(\oint_2^3 (X)))$$



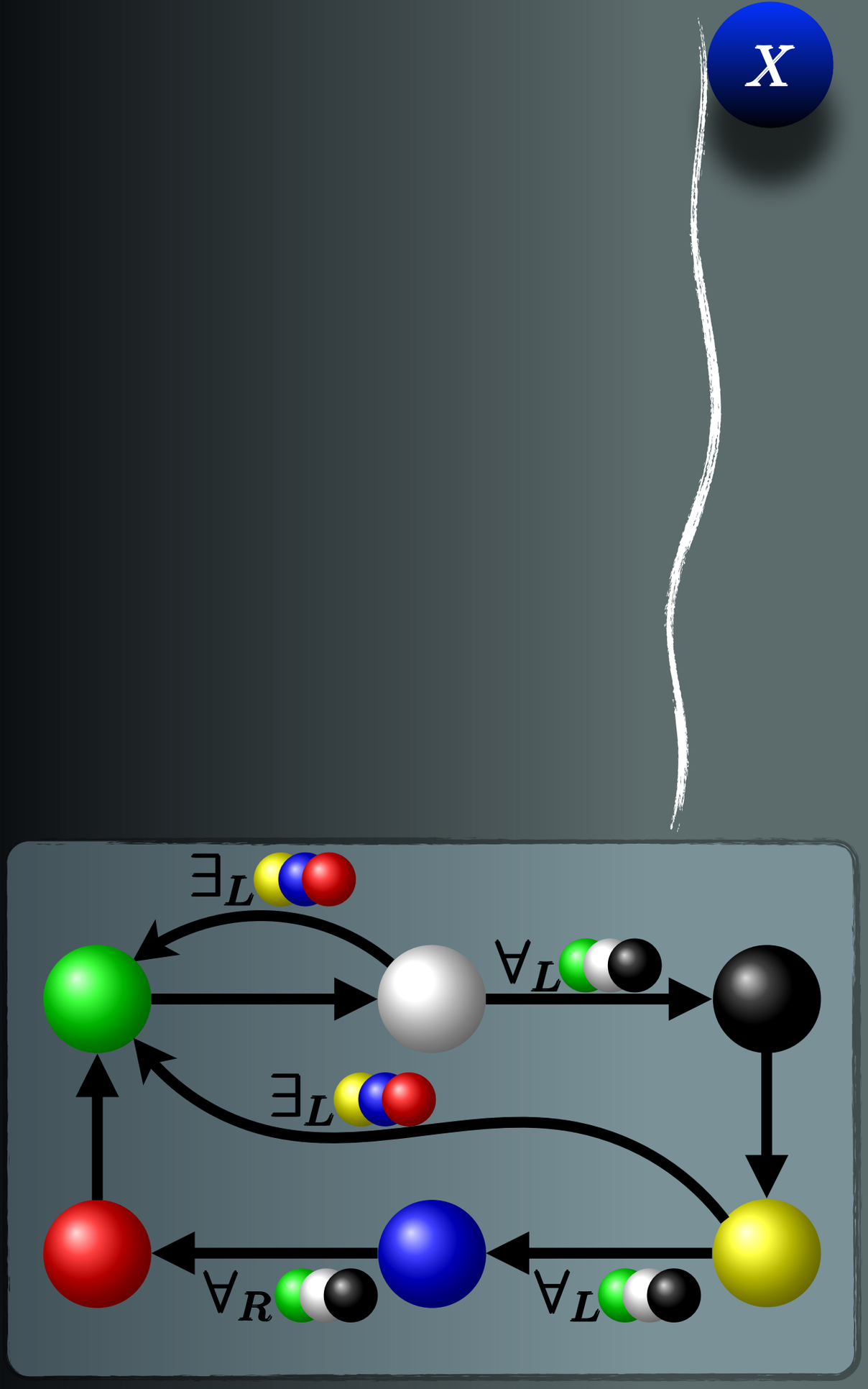






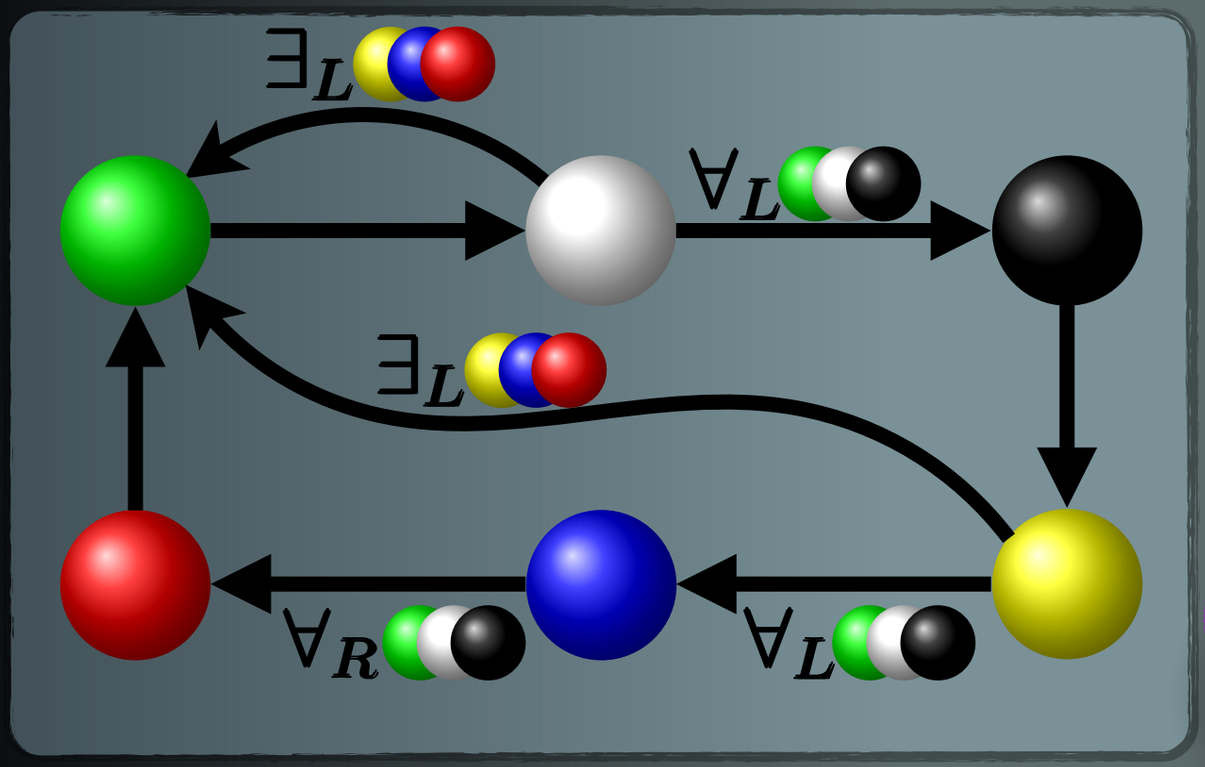


$$V_2 := \mu X.\ \alpha_2(I) \sqcup \alpha_2(post(\oint_2^3 (X)))$$

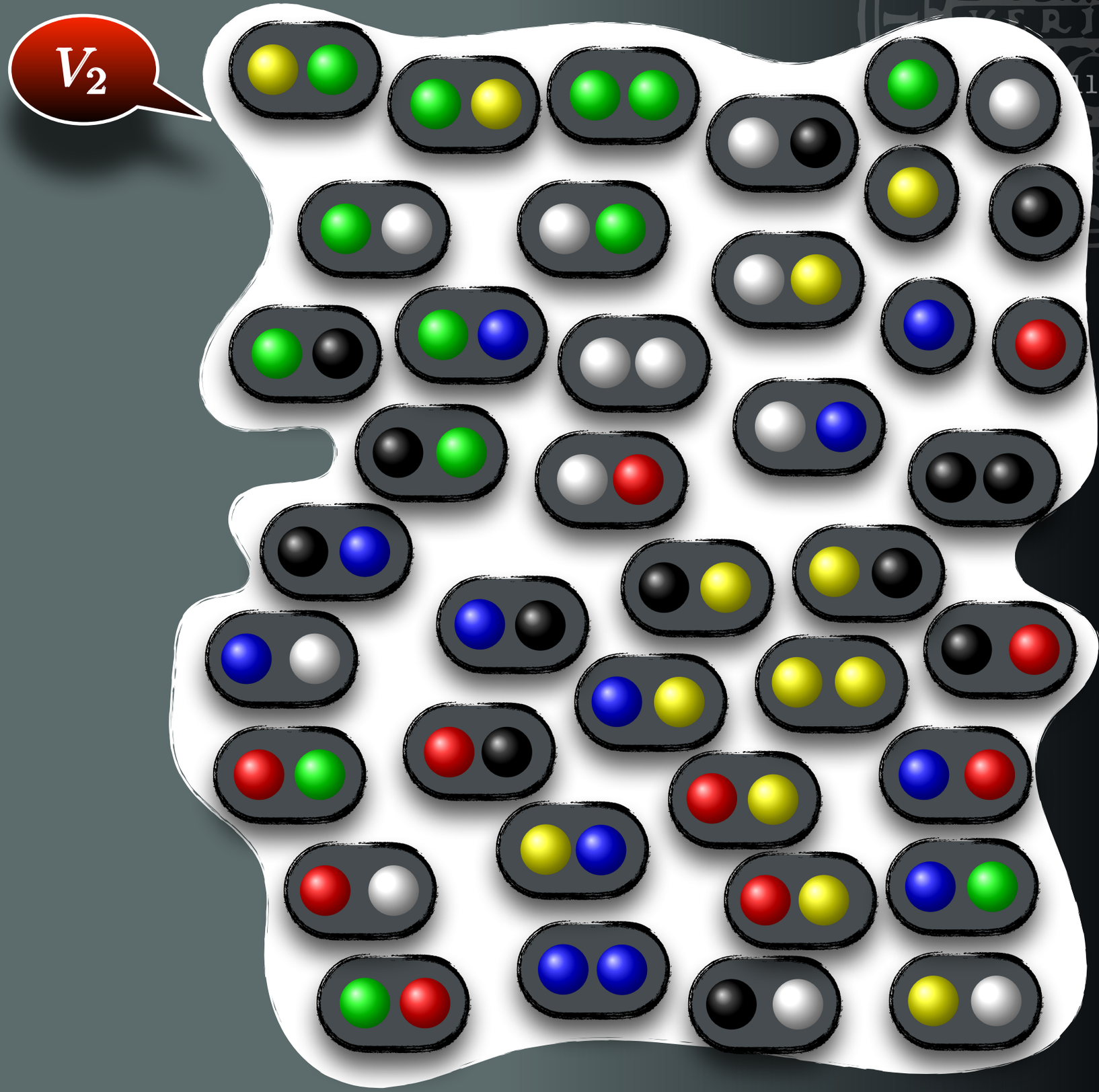


$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$

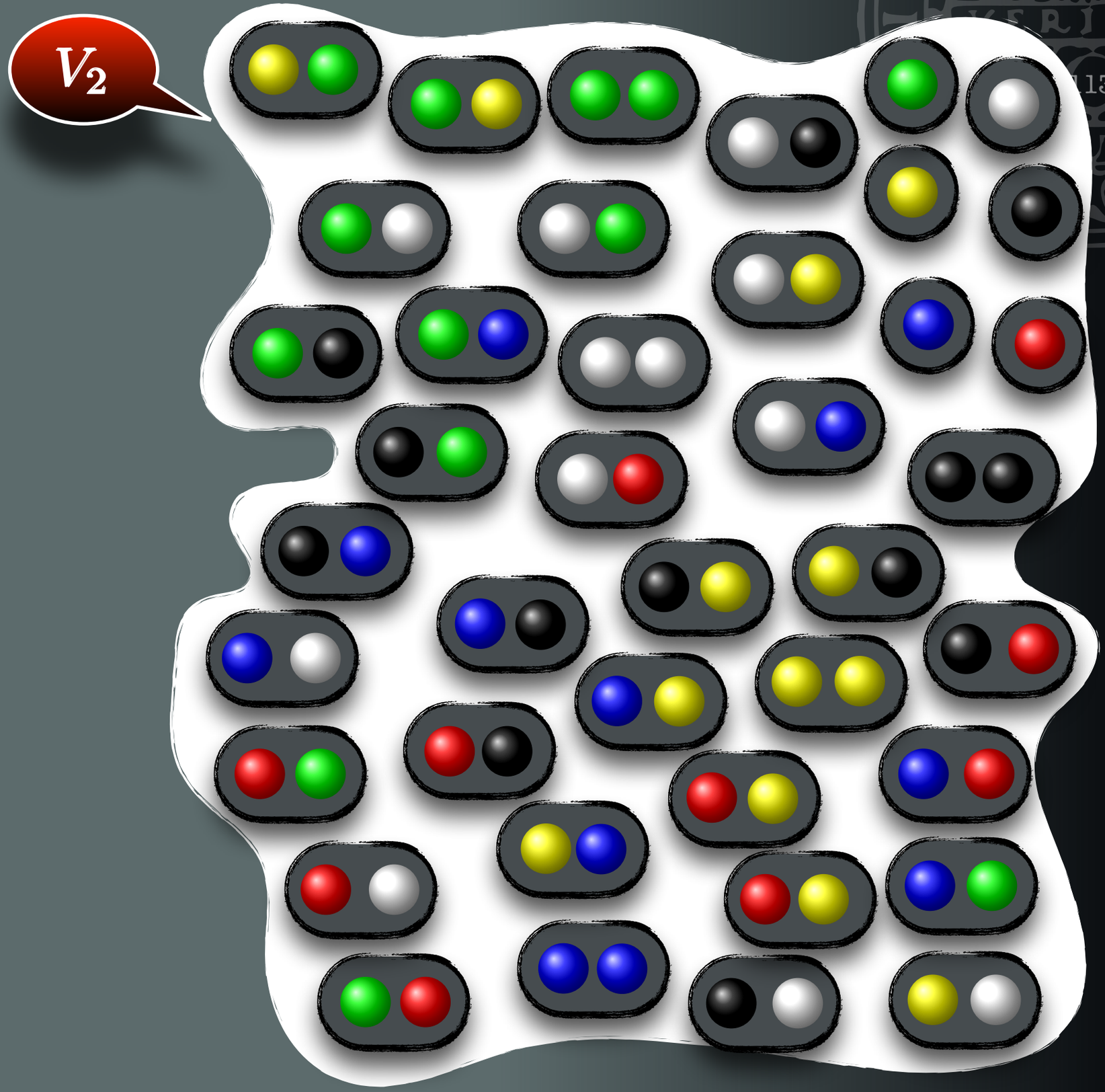
X

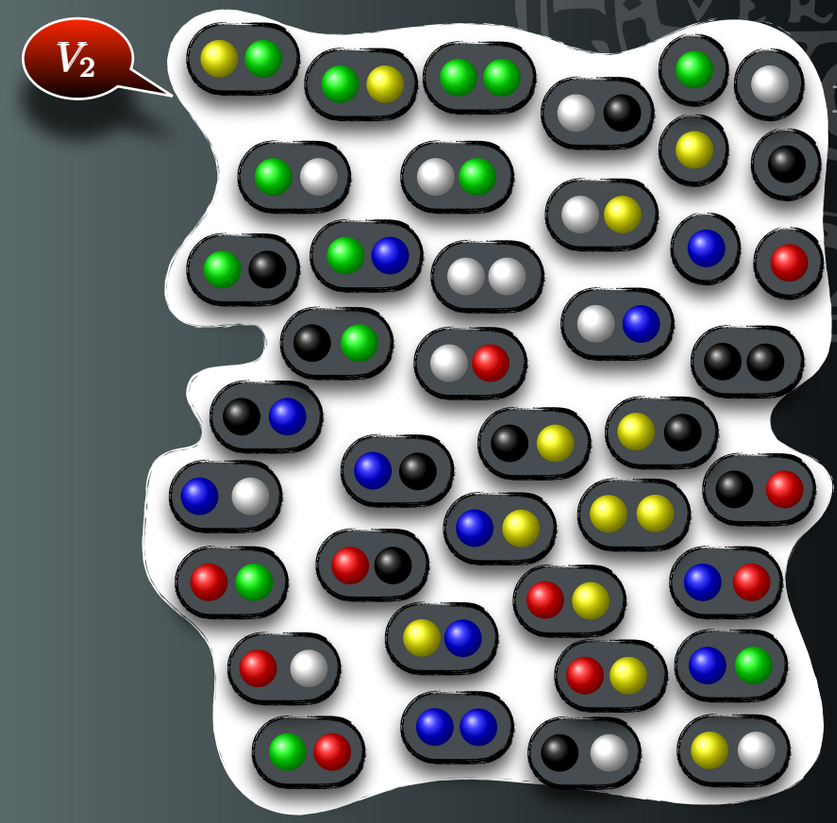


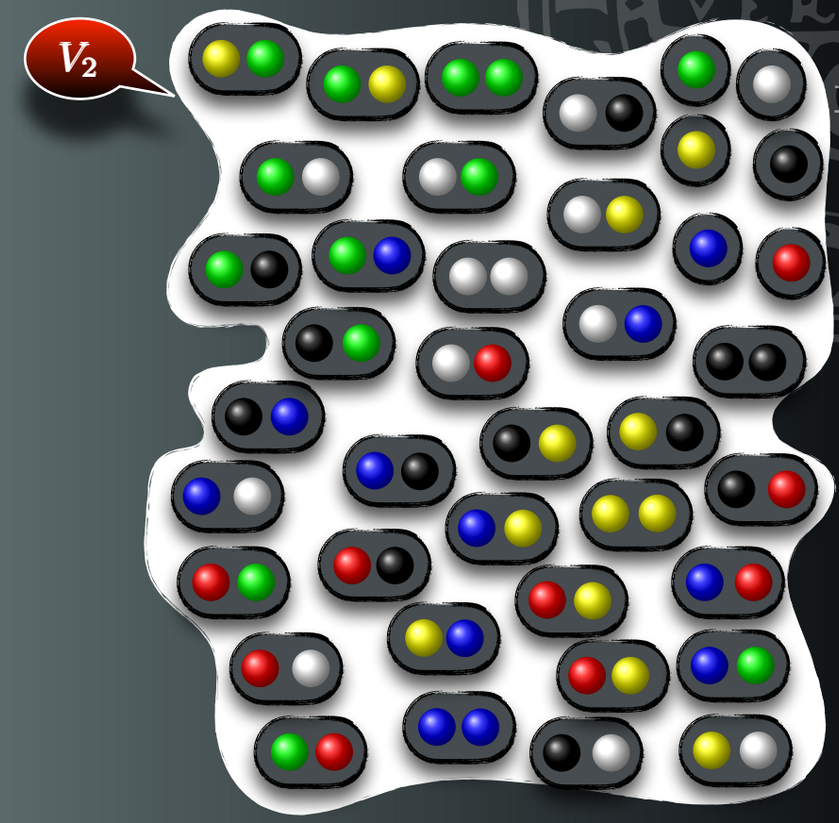
$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$



$$V_2 := \mu X.\ \alpha_2(I) \cup \alpha_2(post(\oint_2^3 (X)))$$


V_2


V_2



$k := 1$

$k = 2$

$\mathbb{R}_k \cap Bad \neq \emptyset$?

yes

unsafe

no

$k++$

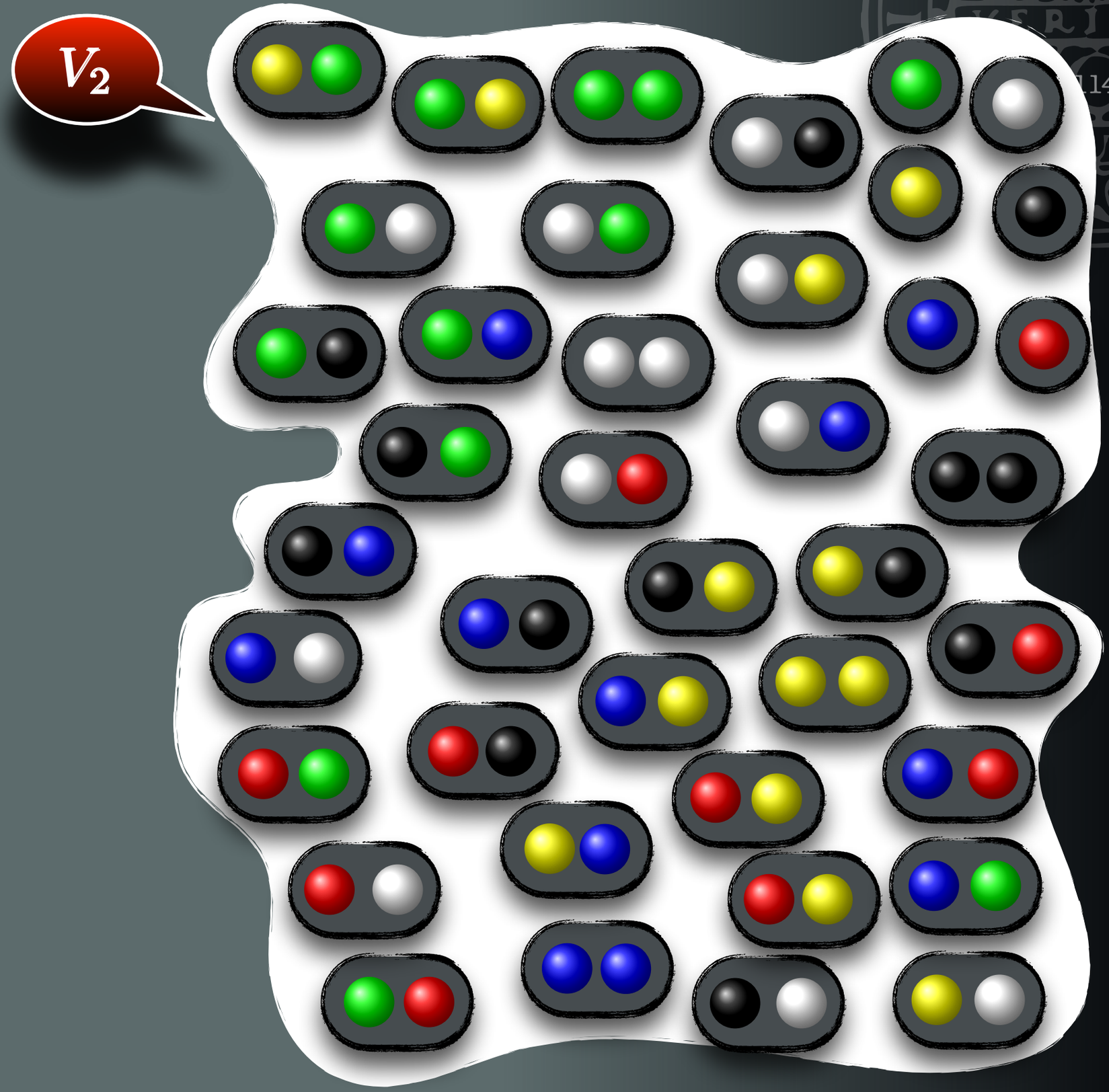
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$

no

$\gamma_k(V_k) \cap Bad = \emptyset$?

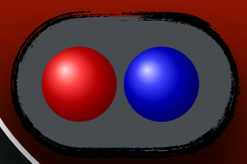
yes

safe

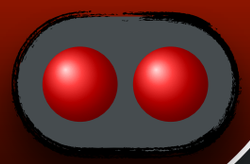

V_2

V_2

All pairs but



and



V_2

All pairs but and

$\gamma_2(V_2)$

$$\left(\bigcirc + \bigcirc + \bigcirc + \bigcirc + \bigcirc\right)^* \left(\bigcirc + \epsilon\right) \left(\bigcirc + \bigcirc + \bigcirc + \bigcirc\right)^*$$

V_2

All pairs but

and

no

$\gamma_2(V_2)$

$$\left(\quad + \quad + \quad + \quad + \quad\right)^* \left(\quad + \epsilon\right) \left(\quad + \quad + \quad + \quad + \quad\right)^*$$

Bad Configurations

V_2

All pairs but and

no

$\gamma_2(V_2)$

$$\left(\bullet + \bullet + \bullet + \bullet + \bullet\right)^* \left(\bullet + \epsilon\right) \left(\bullet + \bullet + \bullet + \bullet\right)^*$$

Bad Configurations

All pairs but and

V_2

$\gamma_2(V_2) \cap Bad = \emptyset$

no

$\gamma_2(V_2)$

$$\left(\quad + \quad + \quad + \quad + \quad\right)^* \left(\quad + \epsilon \right) \left(\quad + \quad + \quad + \quad \right)^*$$

$$\gamma_2(V_2) \cap Bad = \emptyset$$

$\gamma_2(V_2) \cap Bad = \emptyset$

$k := 1$

$k = 2$

$\mathbb{R}_k \cap Bad \neq \emptyset$?

yes

unsafe

no

$k++$

$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$

no

$\gamma_k(V_k) \cap Bad = \emptyset$?

yes

safe

$\gamma_2(V_2) \cap Bad = \emptyset$

$k := 1$

$k = 2$

$\mathbb{R}_k \cap Bad \neq \emptyset$?

yes

unsafe

no

$k++$

$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$

no

$\gamma_k(V_k) \cap Bad = \emptyset$?

yes

safe

$\gamma_2(V_2) \cap Bad = \emptyset$

cut-off point =2

$k := 1$

$k = 2$

$\mathbb{R}_k \cap Bad \neq \emptyset$?

yes

unsafe

no

$k++$

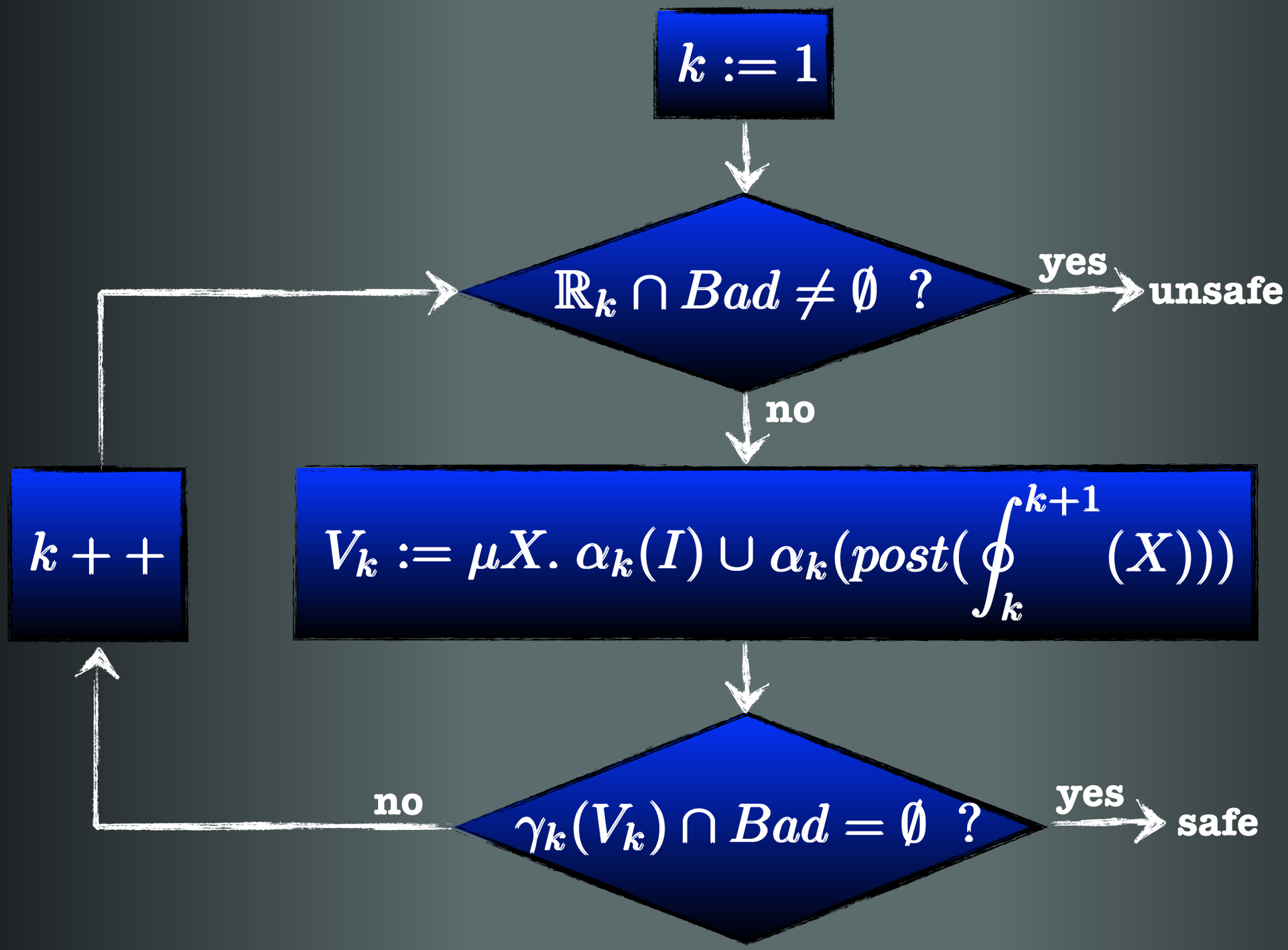
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$

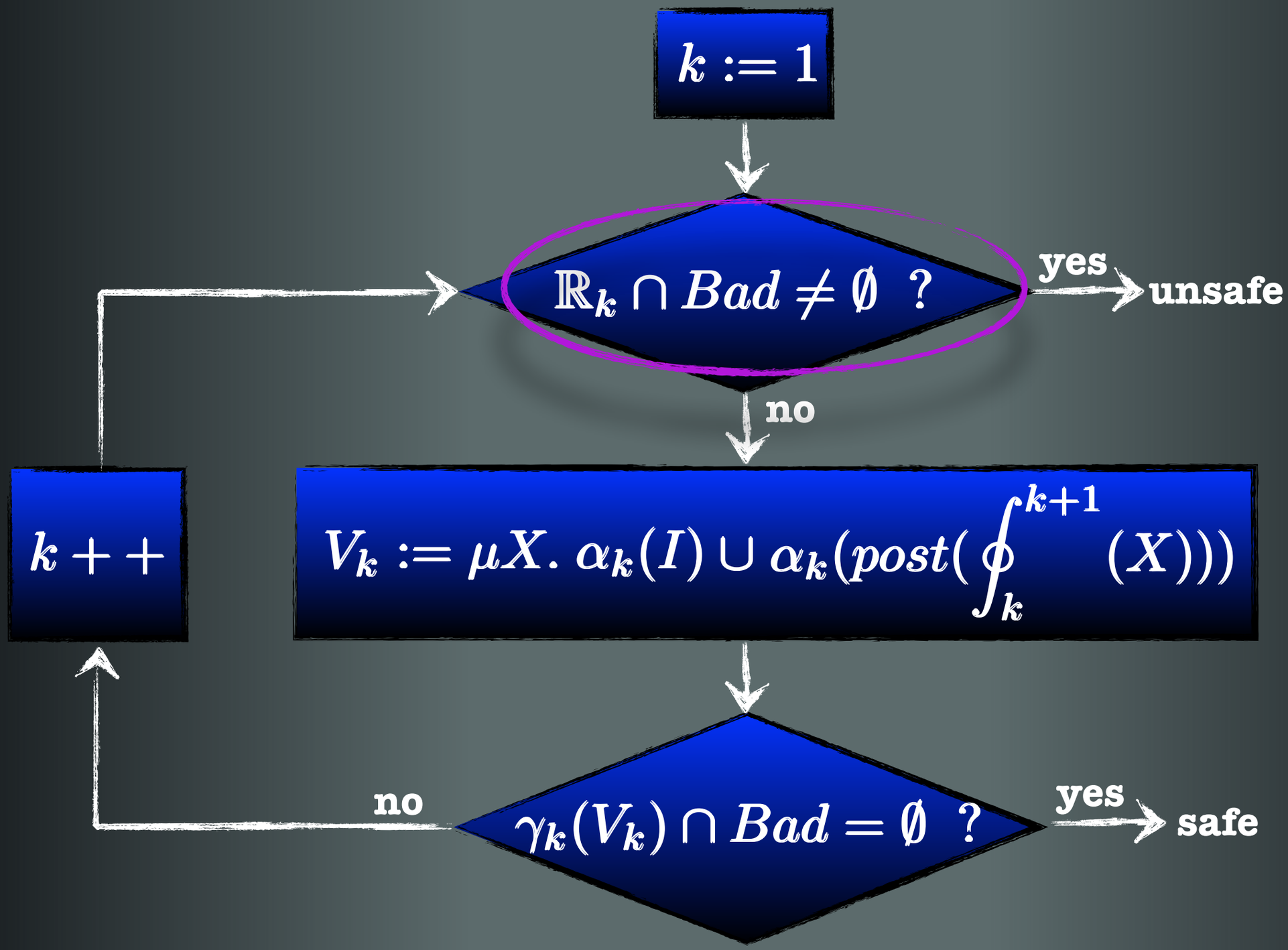
no

$\gamma_k(V_k) \cap Bad = \emptyset$?

yes

safe


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$k++$
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$
$\gamma_k(V_k) \cap Bad = \emptyset$?
no
yes
safe


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1} (X)))$
$k++$
$\gamma_k(V_k) \cap Bad = \emptyset$?
no
yes
safe

$\mathbb{R}_k \cap Bad \neq \emptyset$?

$k := 1$

$\mathbb{R}_k \cap Bad \neq \emptyset$?

yes

unsafe

no

$k++$

$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$

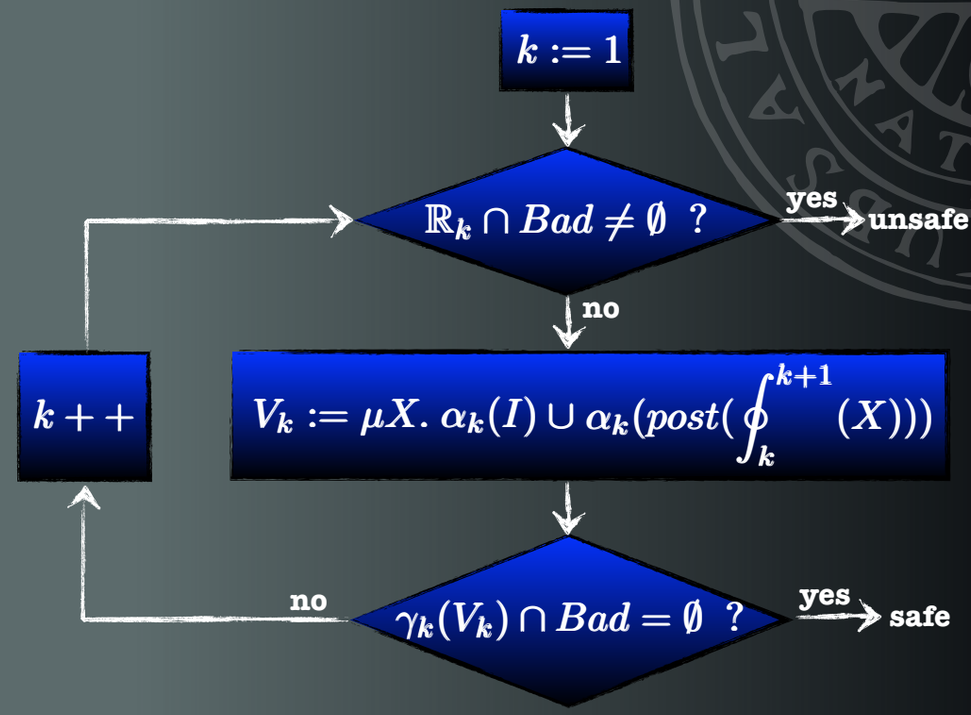
no

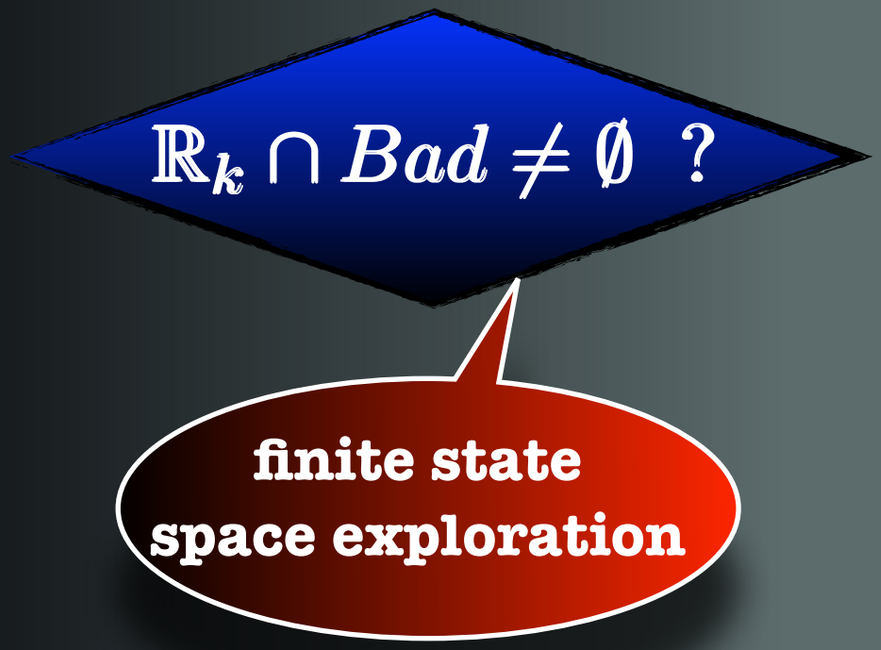
$\gamma_k(V_k) \cap Bad = \emptyset$?

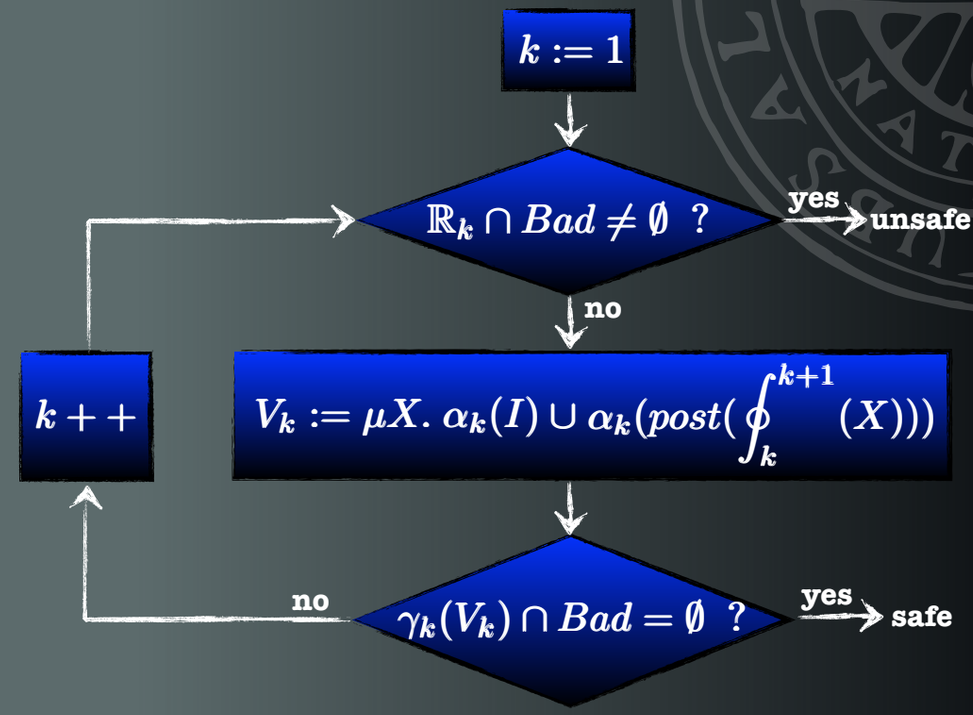
yes

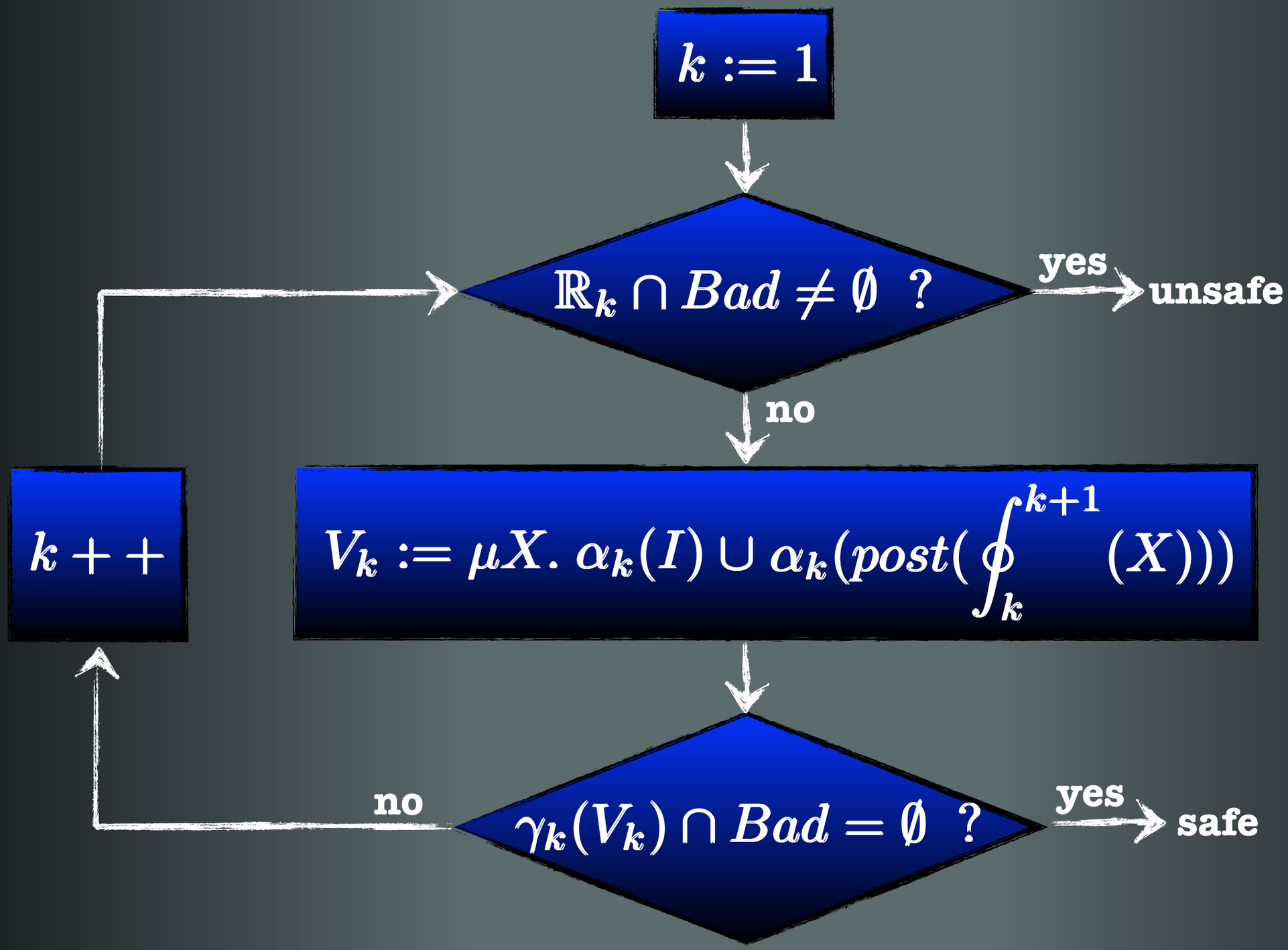
safe


$\mathbb{R}_k \cap Bad \neq \emptyset$?


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$k++$
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$
$\gamma_k(V_k) \cap Bad = \emptyset$?
no
yes
safe


$\mathbb{R}_k \cap Bad \neq \emptyset$?
finite state
space exploration


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$k++$
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$
no
$\gamma_k(V_k) \cap Bad = \emptyset$?
yes
safe


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$k++$
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$
$\gamma_k(V_k) \cap Bad = \emptyset$?
no
yes
safe

$k := 1$

$\mathbb{R}_k \cap Bad \neq \emptyset$?

yes → unsafe

no

$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$

$k++$

$\gamma_k(V_k) \cap Bad = \emptyset$?

no

yes → safe

$\alpha_k(I)$

$k := 1$

$\mathbb{R}_k \cap Bad \neq \emptyset$?

yes

unsafe

no

$k++$

$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$

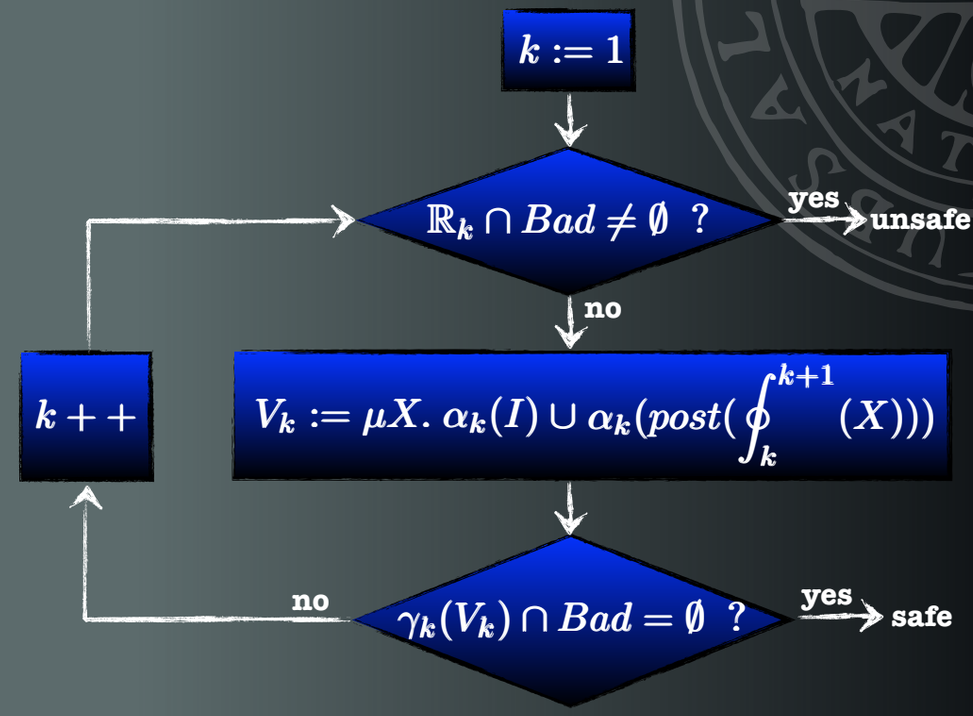
$\gamma_k(V_k) \cap Bad = \emptyset$?

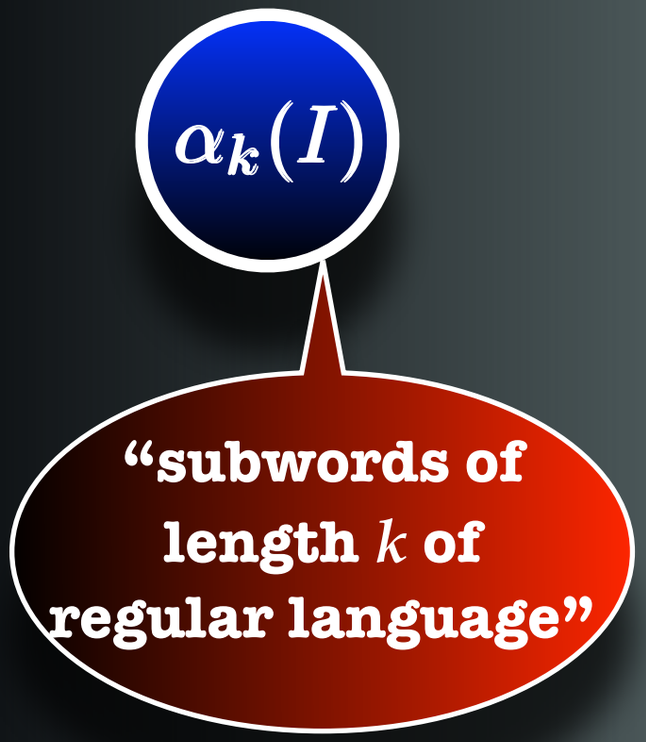
no

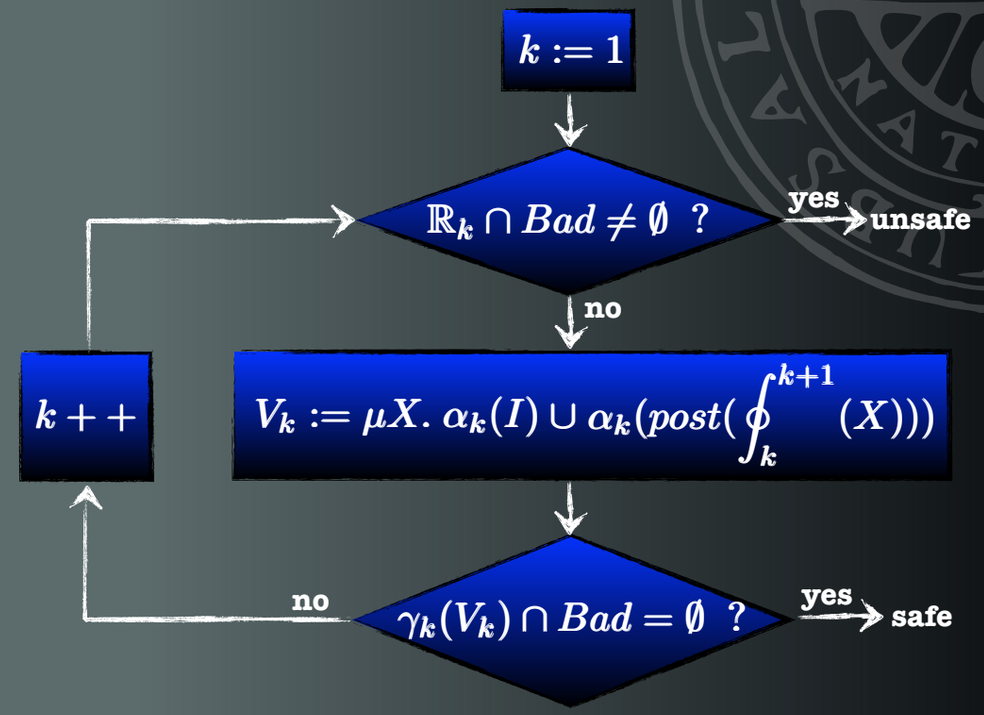
yes

safe

$\alpha_k(I)$


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$k++$
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$
$\gamma_k(V_k) \cap Bad = \emptyset$?
no
yes
safe

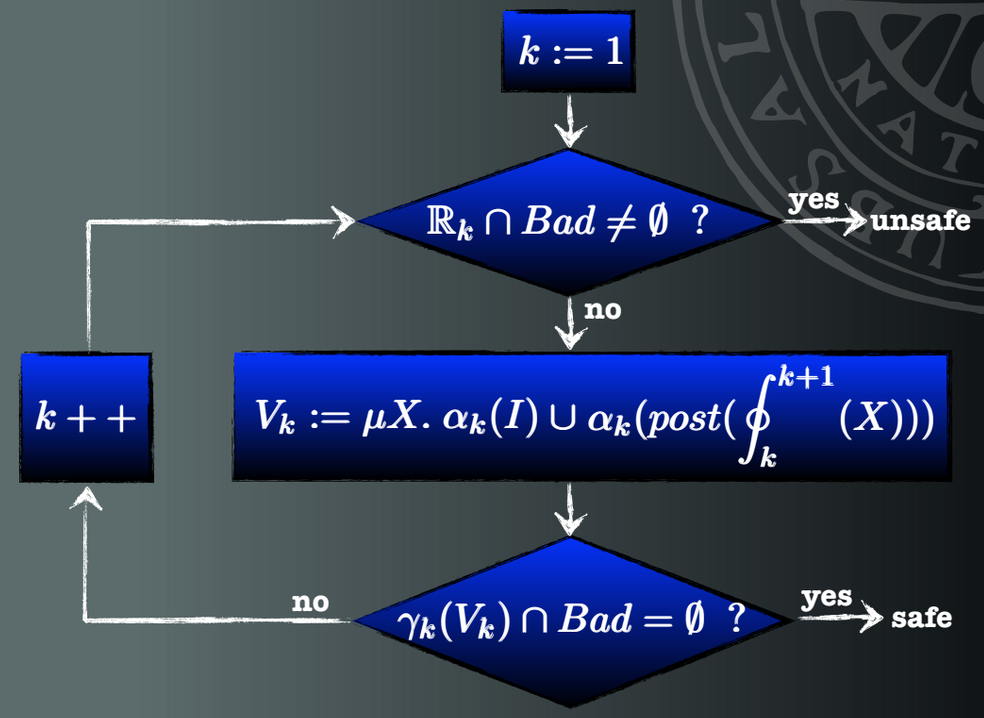

$\alpha_k(I)$
“subwords of length k of regular language”


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$k++$
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$
$\gamma_k(V_k) \cap Bad = \emptyset$?
no
yes
safe

$\alpha_k(I)$

“subwords of length k of regular language”

“simple automata construction”

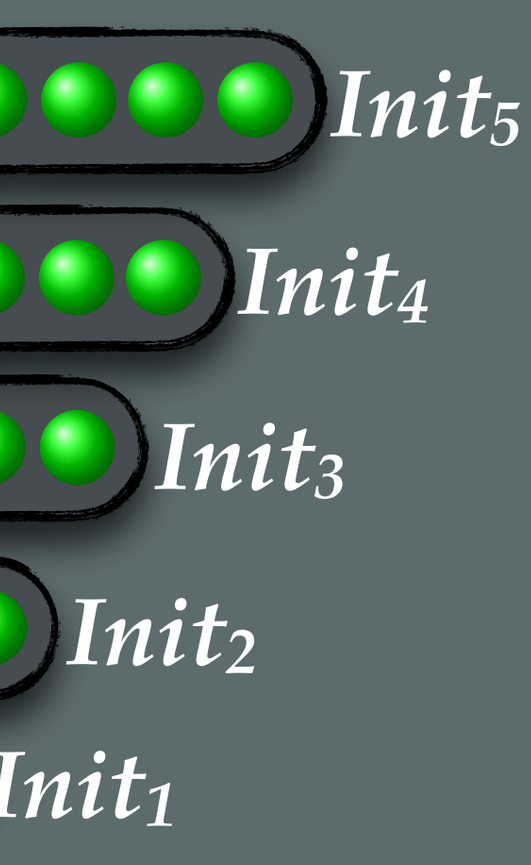


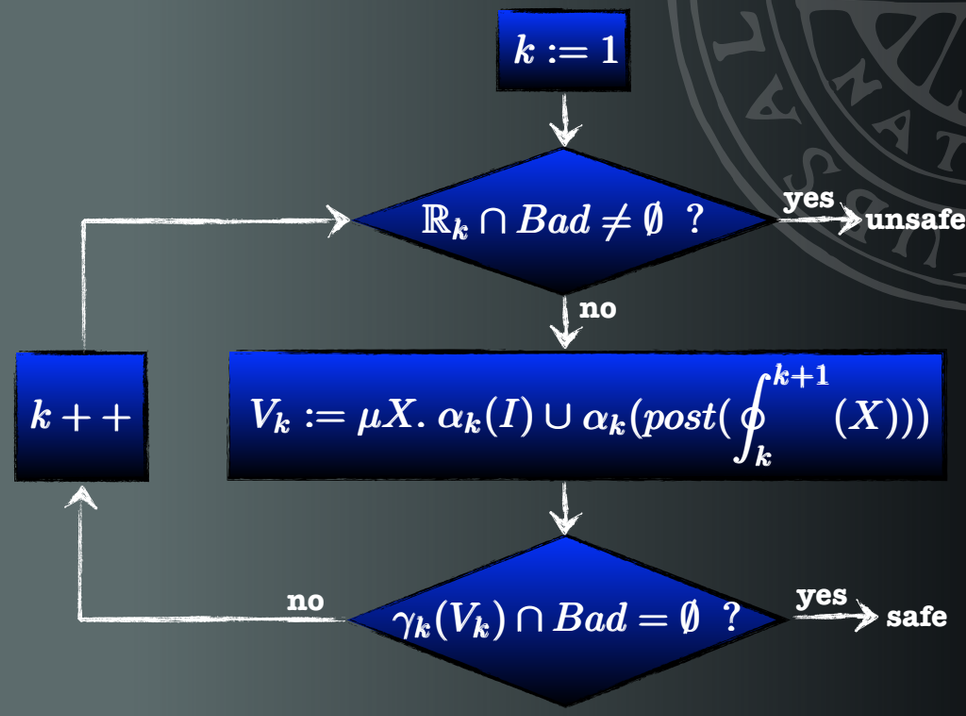
$\alpha_k(I)$

"subwords of length k of regular language"

"simple automata construction"

I

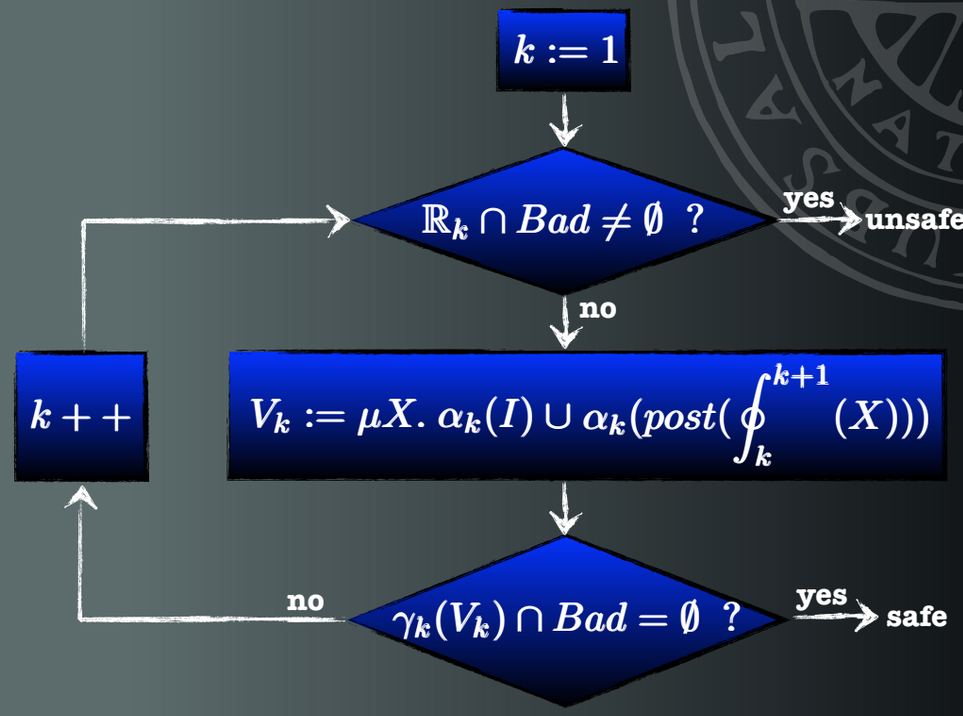


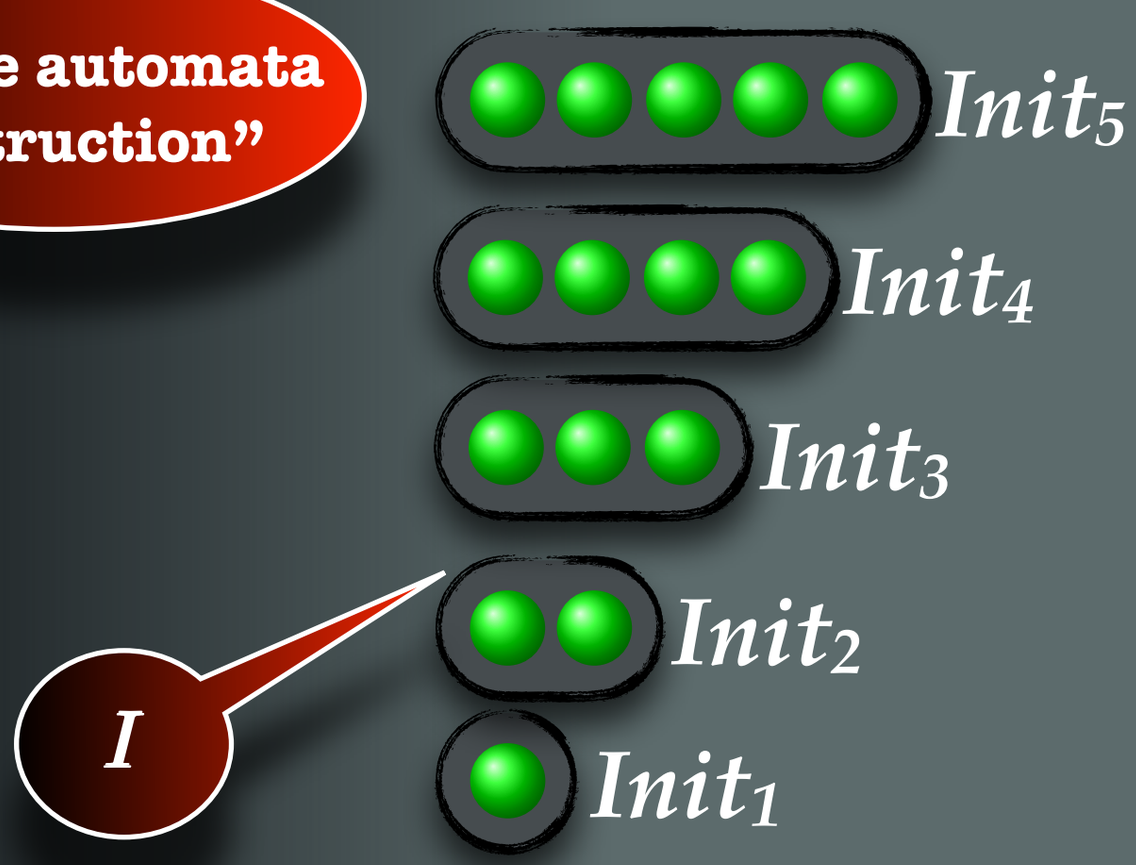


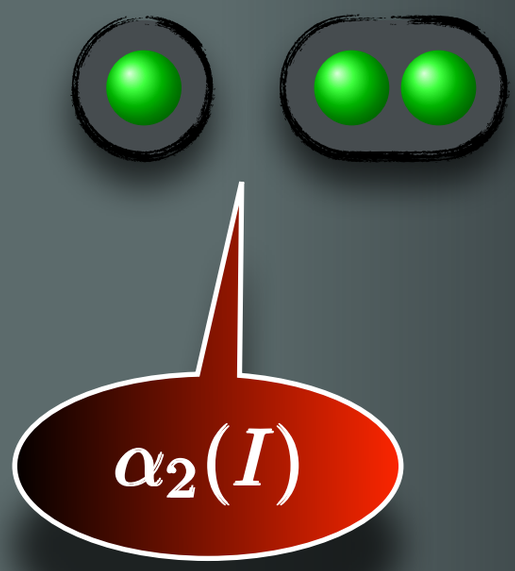
$\alpha_k(I)$

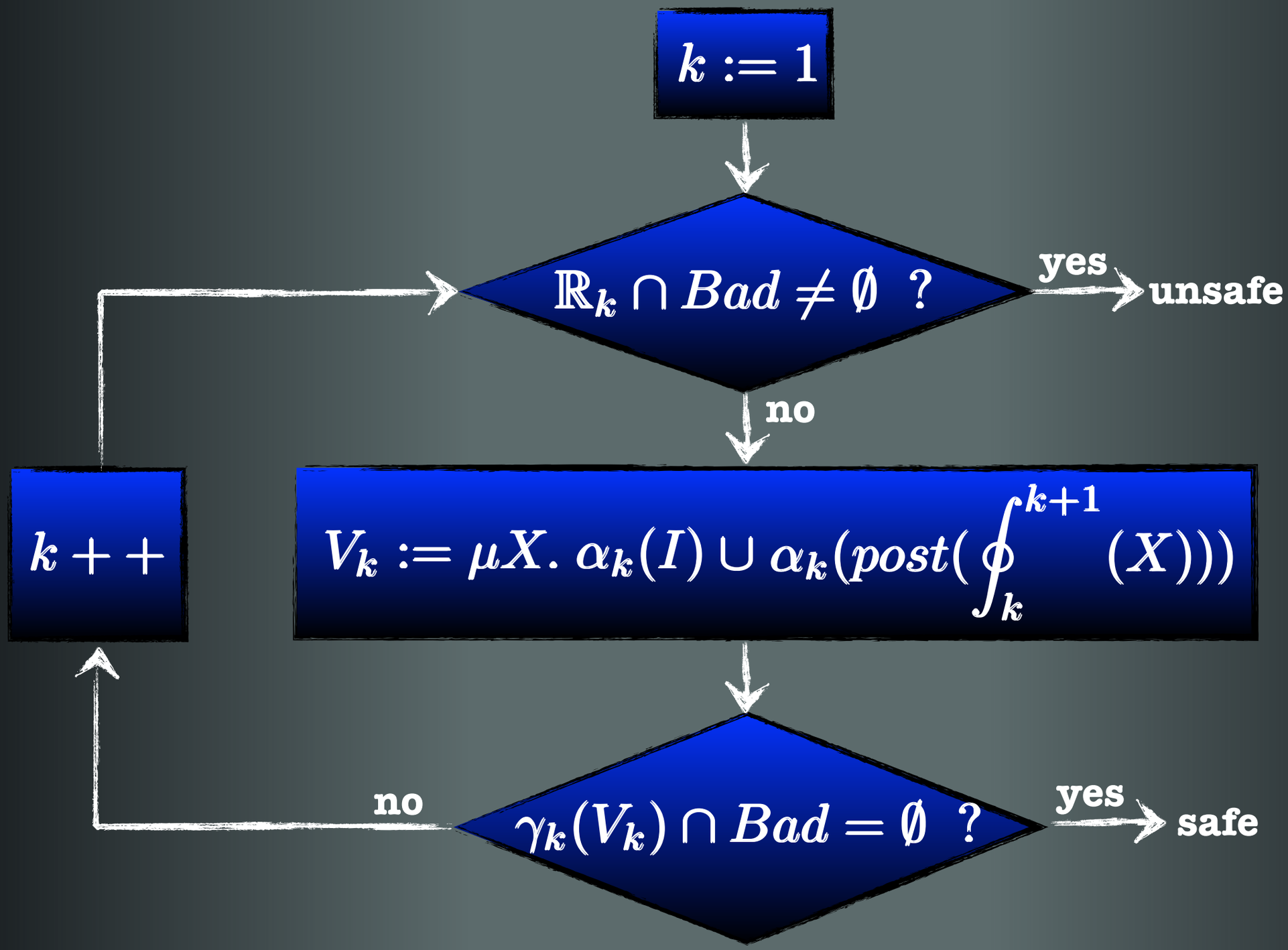
"subwords of length k of regular language"

"simple automata construction"








$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$k++$
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$
$\gamma_k(V_k) \cap Bad = \emptyset$?
no
yes
safe

$k := 1$

$\mathbb{R}_k \cap Bad \neq \emptyset$?

yes → unsafe

no

$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$

$k++$

$\gamma_k(V_k) \cap Bad = \emptyset$?

no

yes → safe

$k := 1$

$\mathbb{R}_k \cap Bad \neq \emptyset$?

yes

"finite set"

no

$k++$

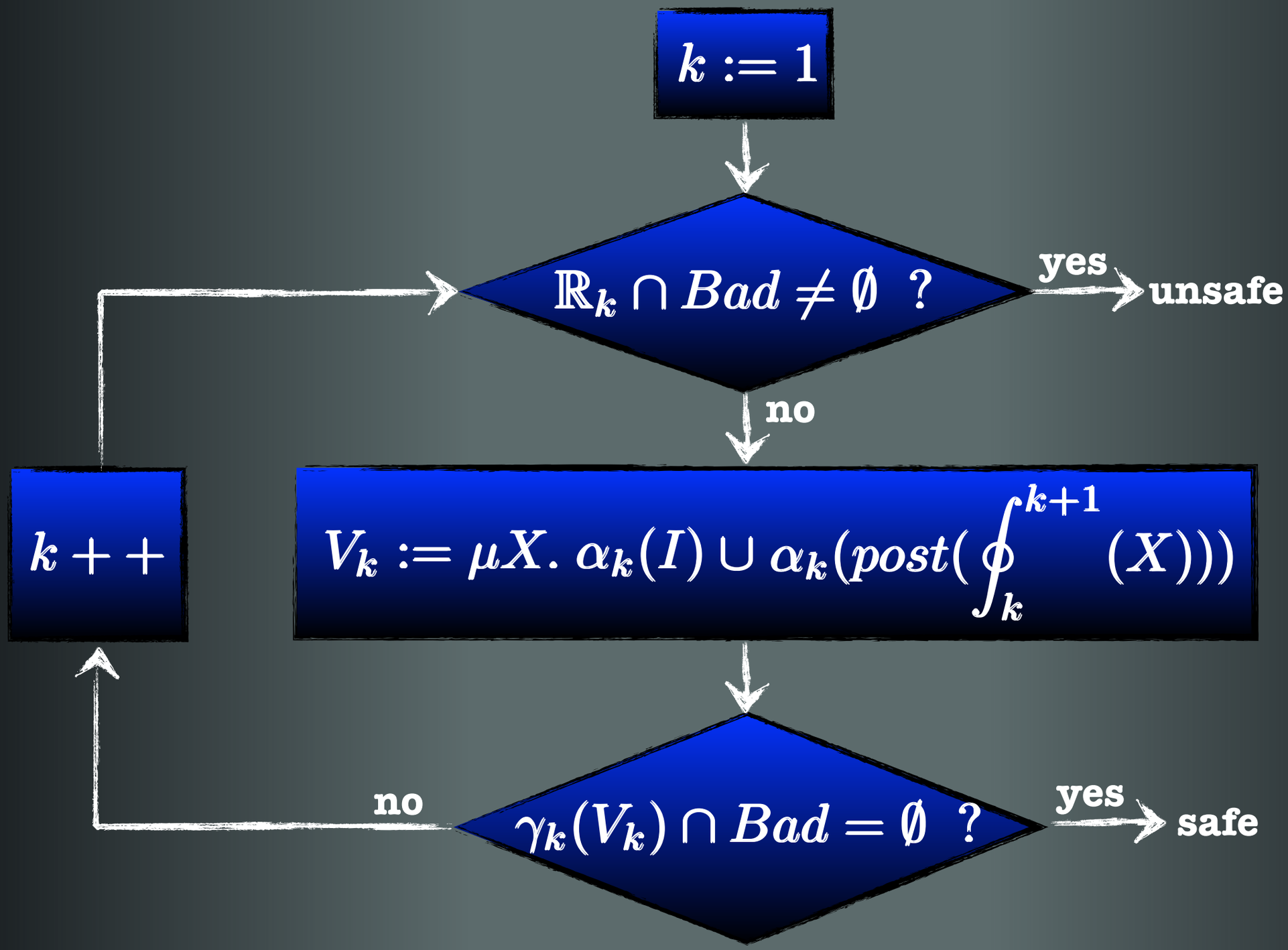
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$

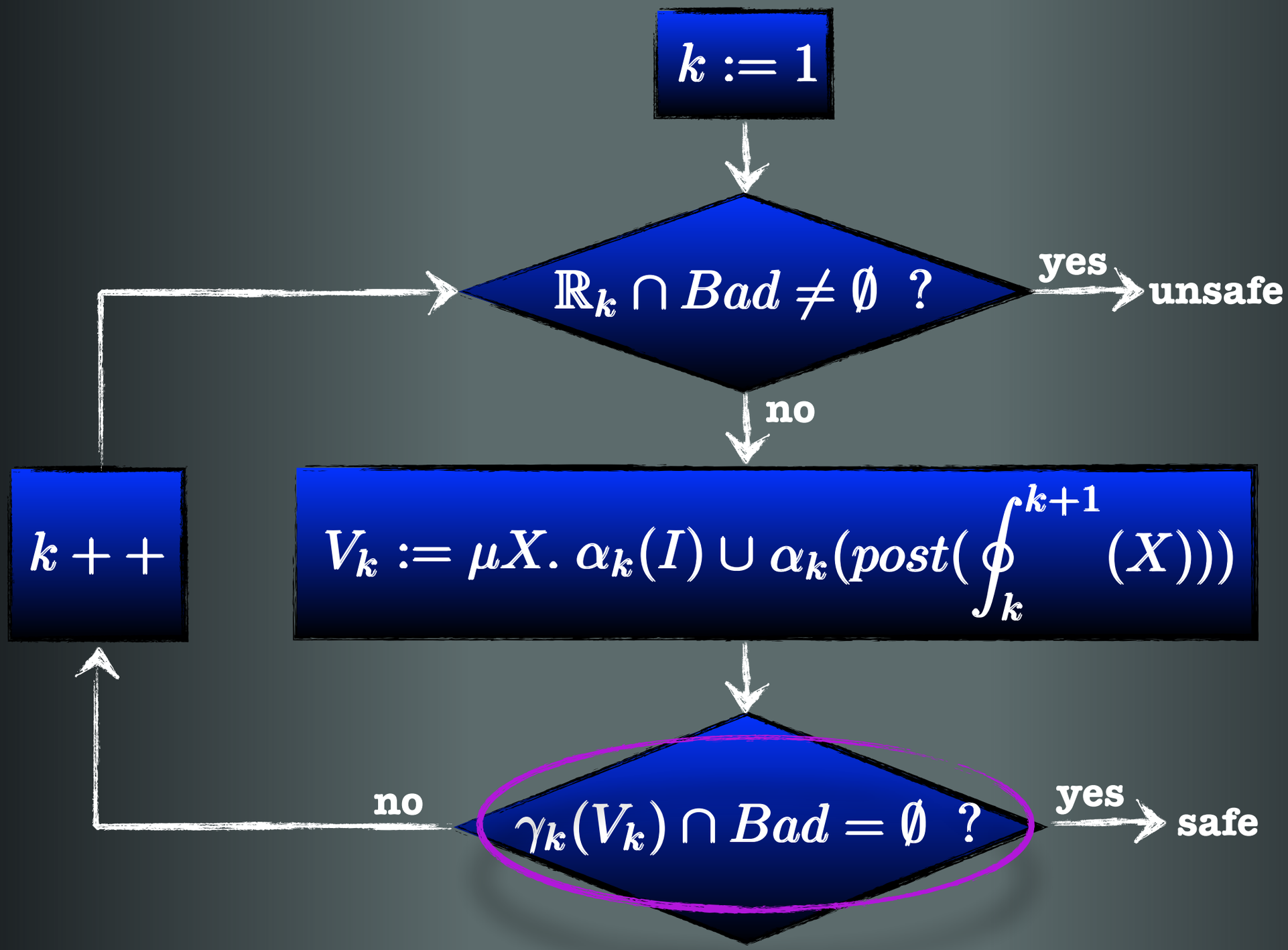
$\gamma_k(V_k) \cap Bad = \emptyset$?

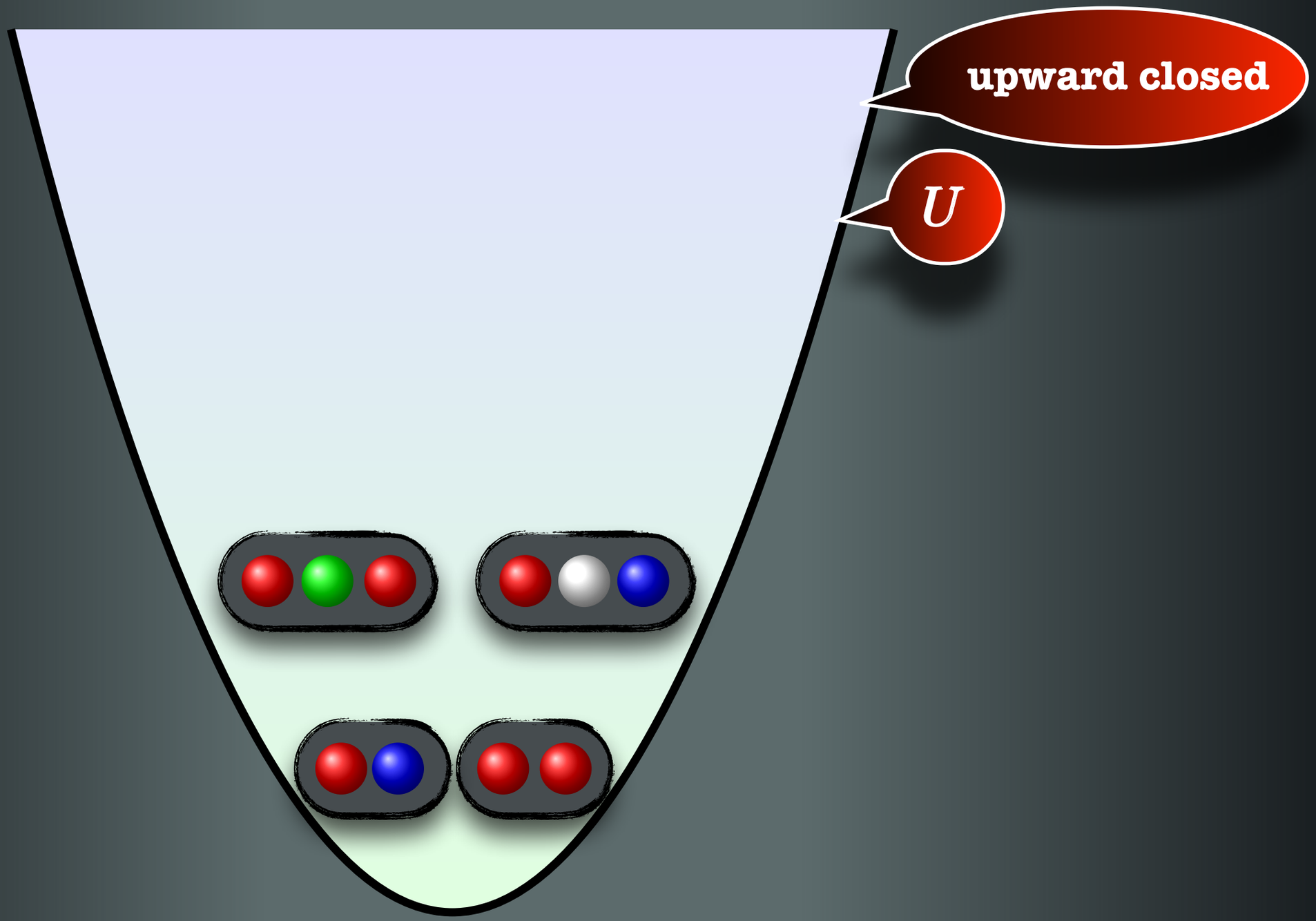
no

yes

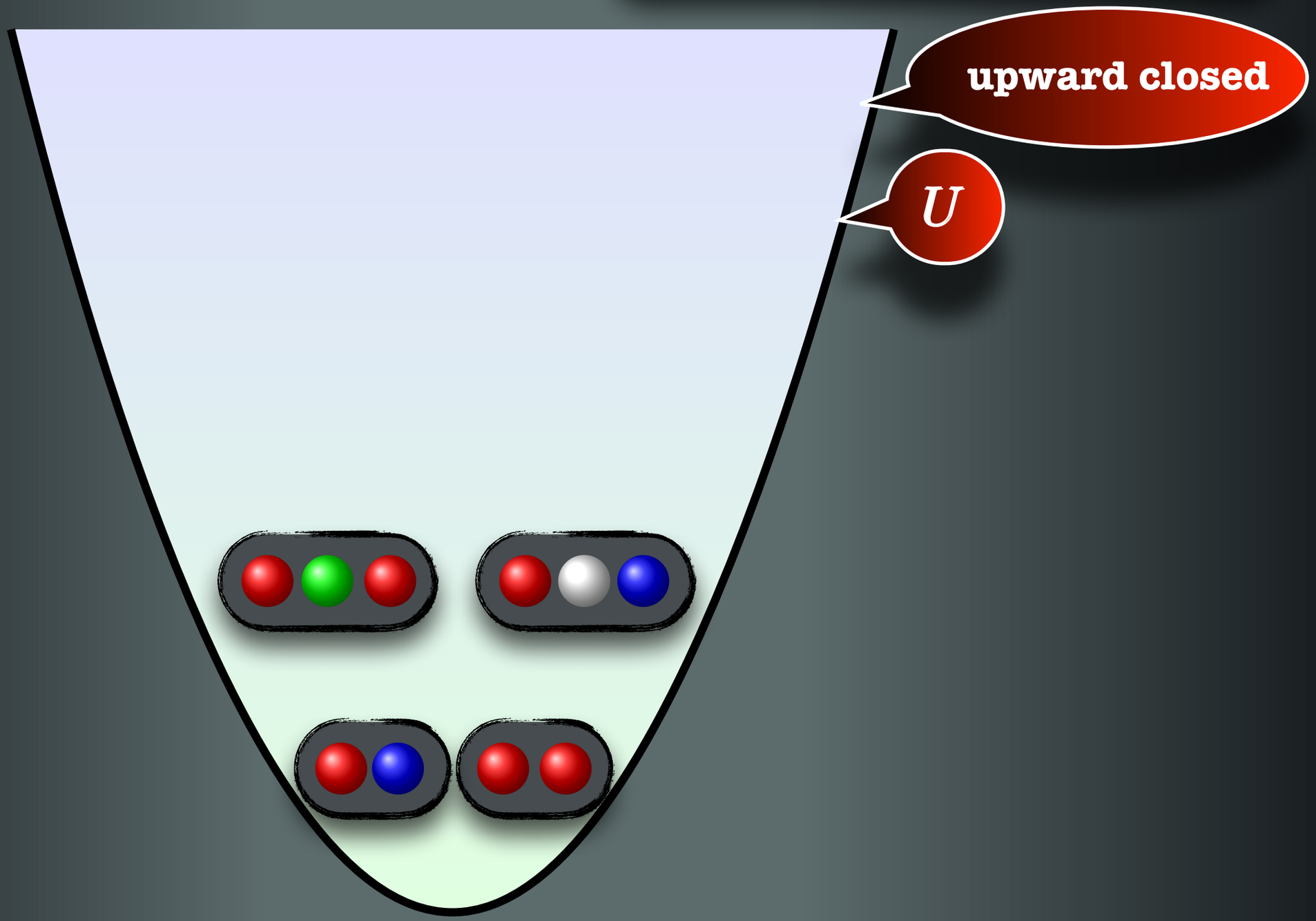
safe


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$k++$
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$
$\gamma_k(V_k) \cap Bad = \emptyset$?
no
yes
safe

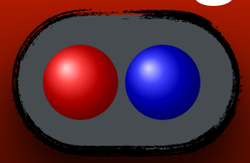

$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$k++$
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$
no
$\gamma_k(V_k) \cap Bad = \emptyset$?
yes
safe


upward closed
U

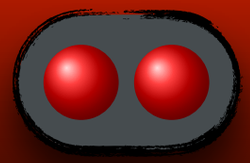
- if $c_1 \in U$ and $c_1 \sqsubseteq c_2$
- then $c_2 \in U$



all configurations with



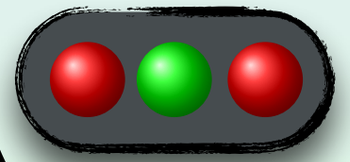
or

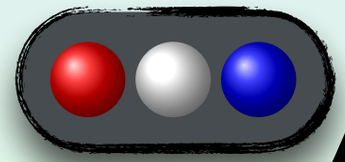


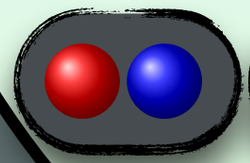
- if $c_1 \in U$ and $c_1 \sqsubseteq c_2$
- then $c_2 \in U$

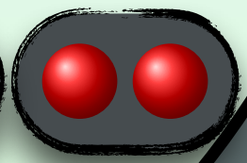
upward closed

U

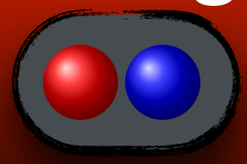




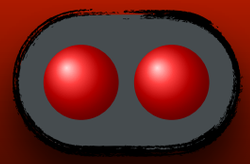




all configurations with



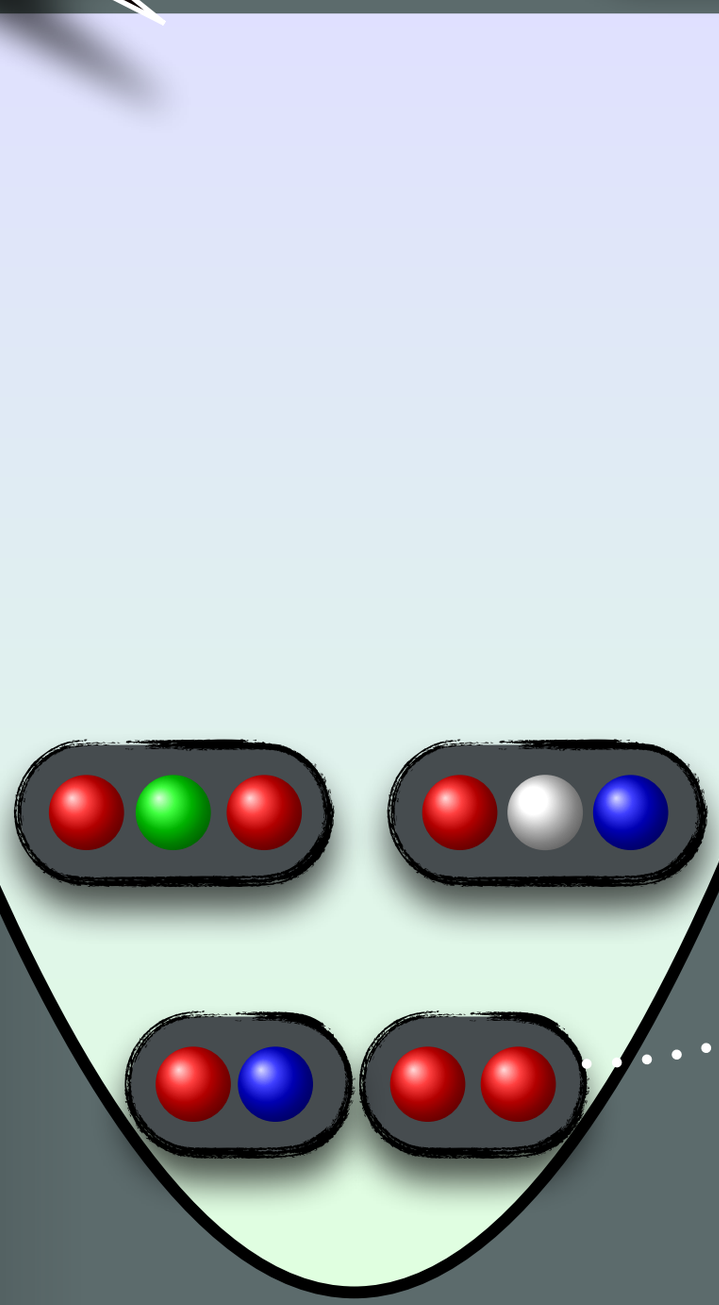
or

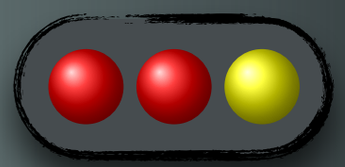


- if $c_1 \in U$ and $c_1 \sqsubseteq c_2$
- then $c_2 \in U$

upward closed

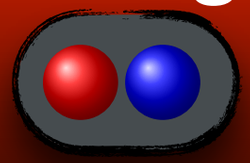
U



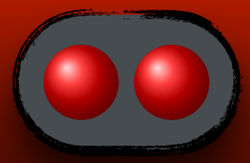


$\sqsubseteq$

all configurations with



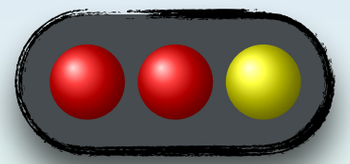
or

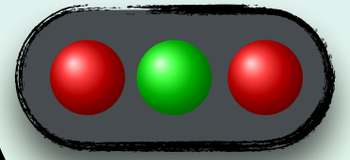


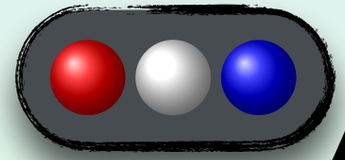
- if $c_1 \in U$ and $c_1 \sqsubseteq c_2$
- then $c_2 \in U$

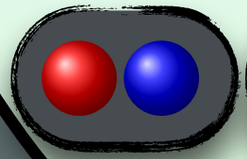
upward closed

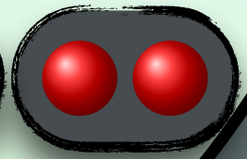
U



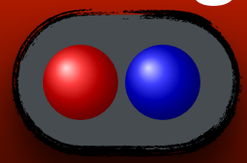




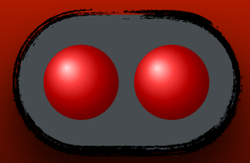




all configurations with



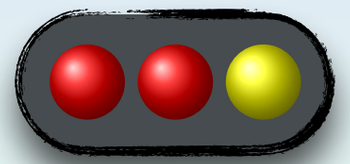
or

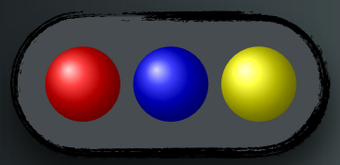


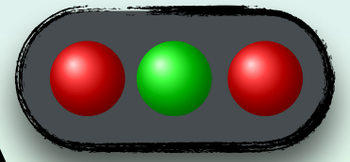
- if $c_1 \in U$ and $c_1 \sqsubseteq c_2$
- then $c_2 \in U$

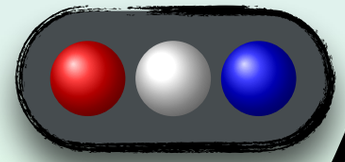
upward closed

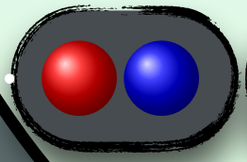
U

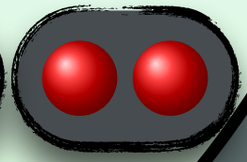












all configurations with

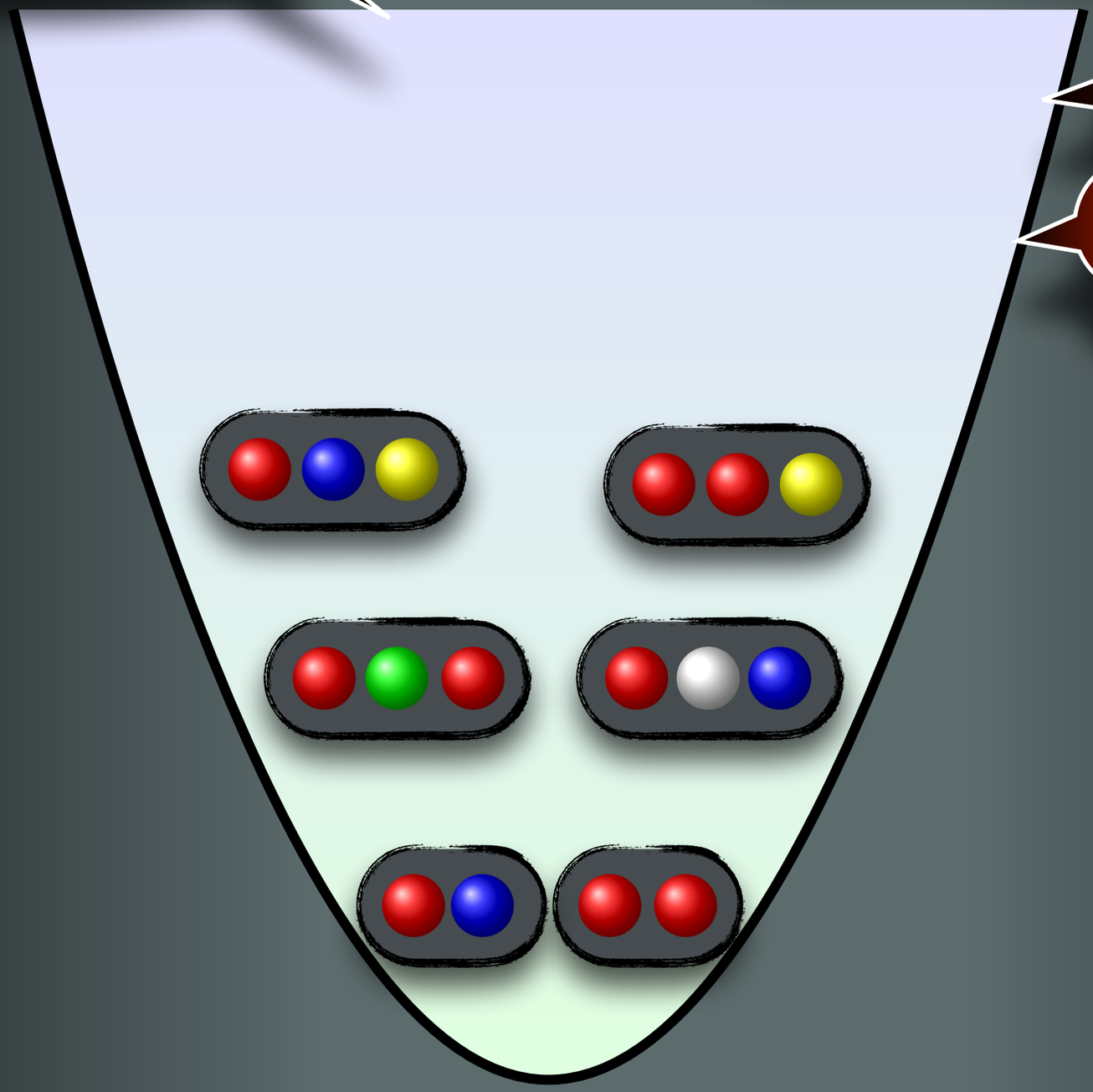


or

- if $c_1 \in U$ and $c_1 \sqsubseteq c_2$
- then $c_2 \in U$

upward closed

U



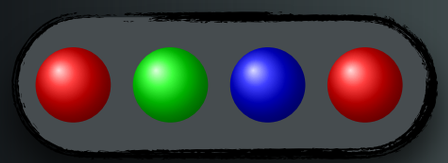
all configurations with

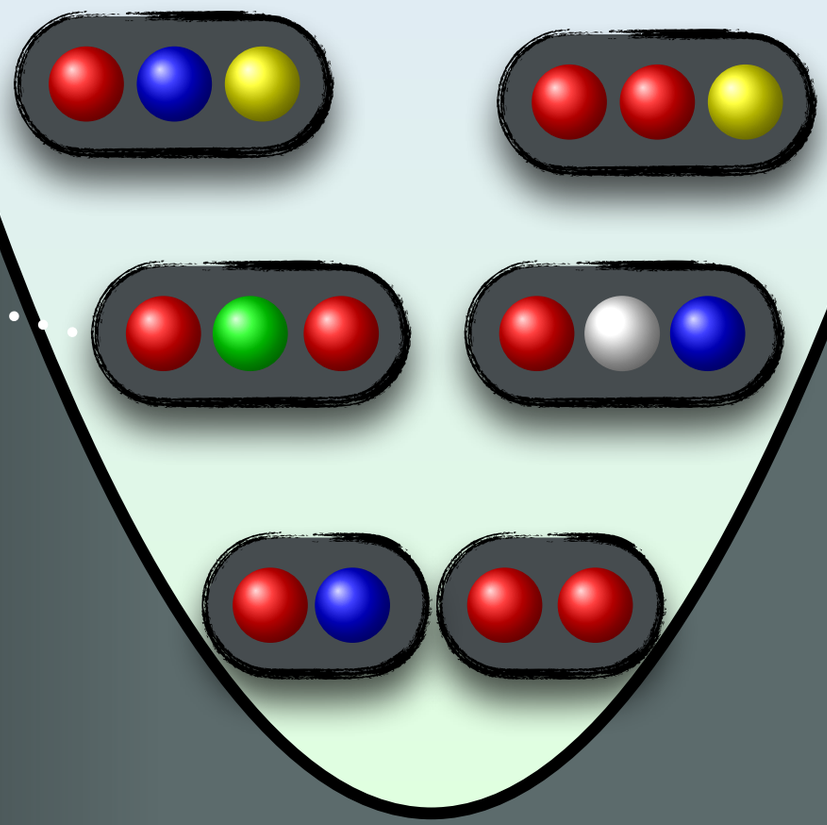


- if $c_1 \in U$ and $c_1 \sqsubseteq c_2$
- then $c_2 \in U$

upward closed

U





all configurations with

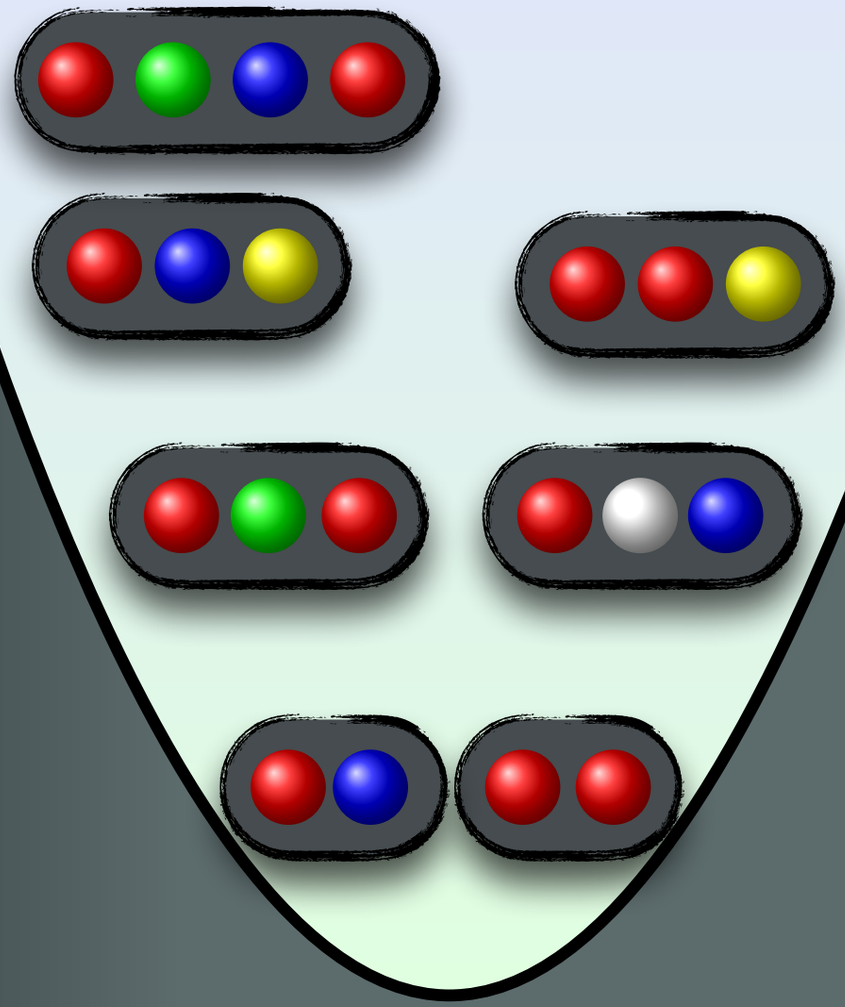


or

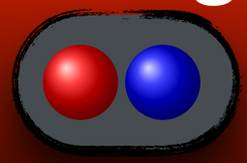
- if $c_1 \in U$ and $c_1 \sqsubseteq c_2$
- then $c_2 \in U$

upward closed

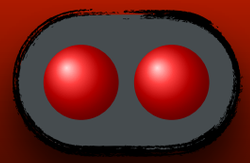
U



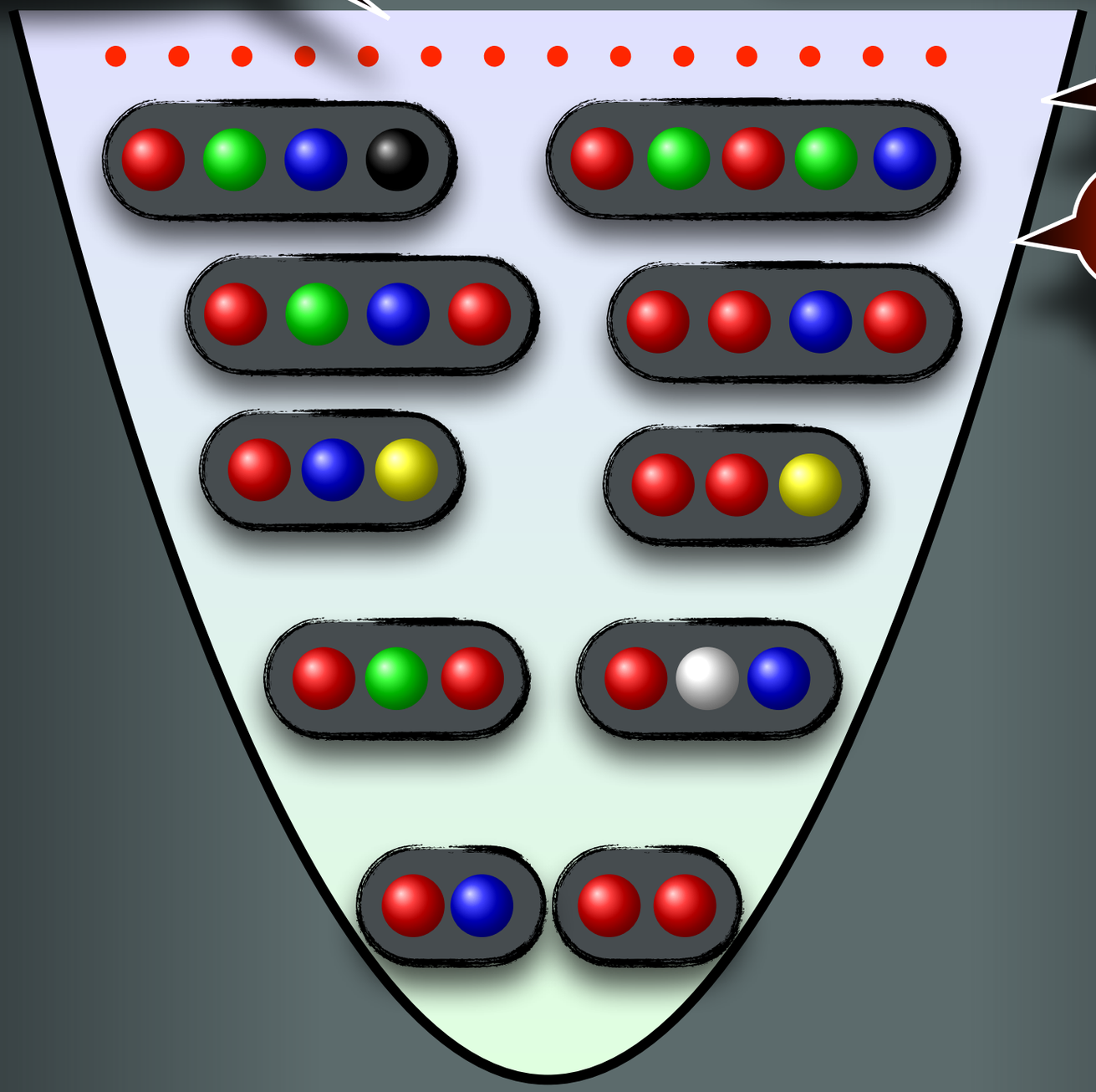
all configurations with



or



- if $c_1 \in U$ and $c_1 \sqsubseteq c_2$
- then $c_2 \in U$



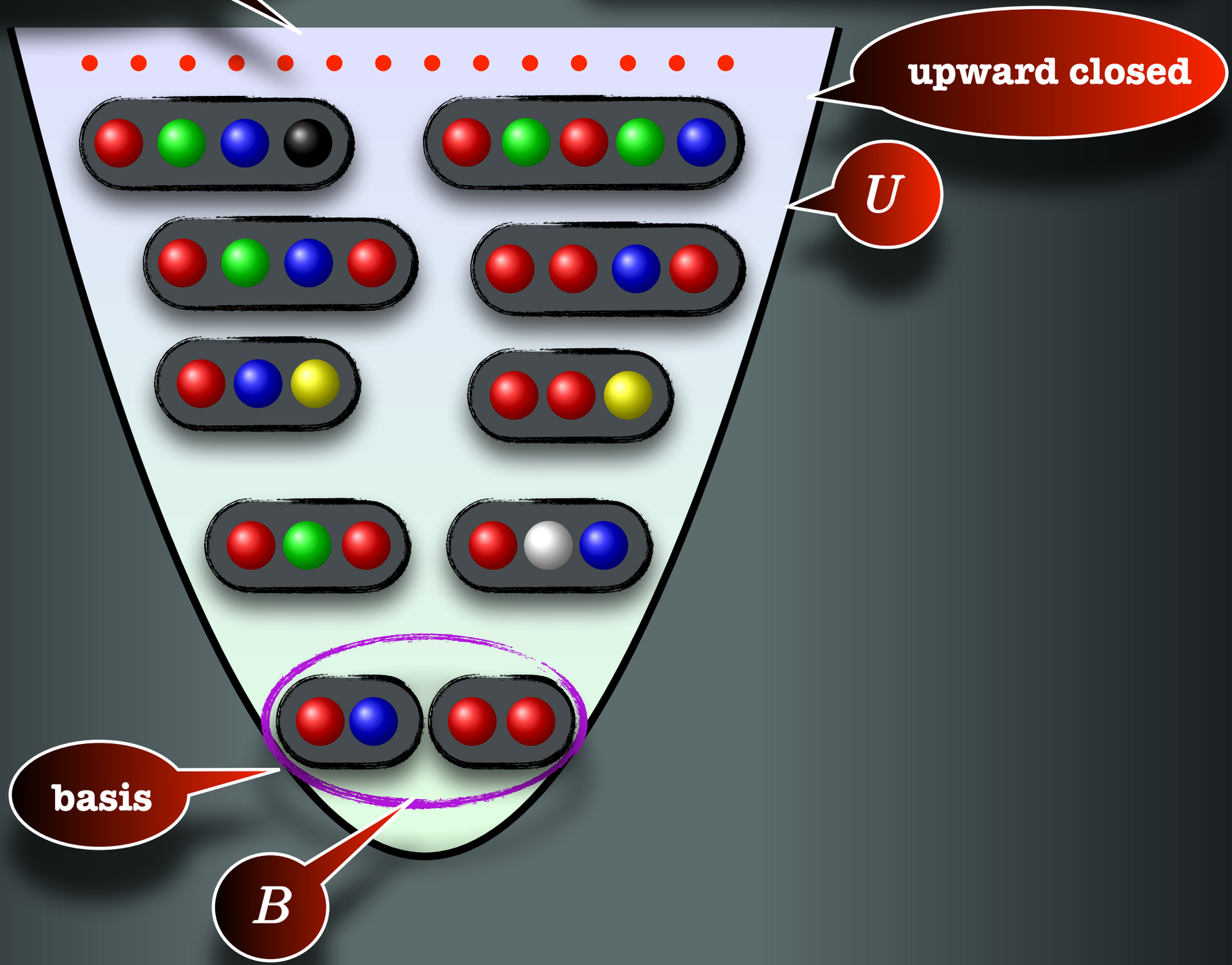
upward closed

U

all configurations with



- if $c_1 \in U$ and $c_1 \sqsubseteq c_2$
- then $c_2 \in U$

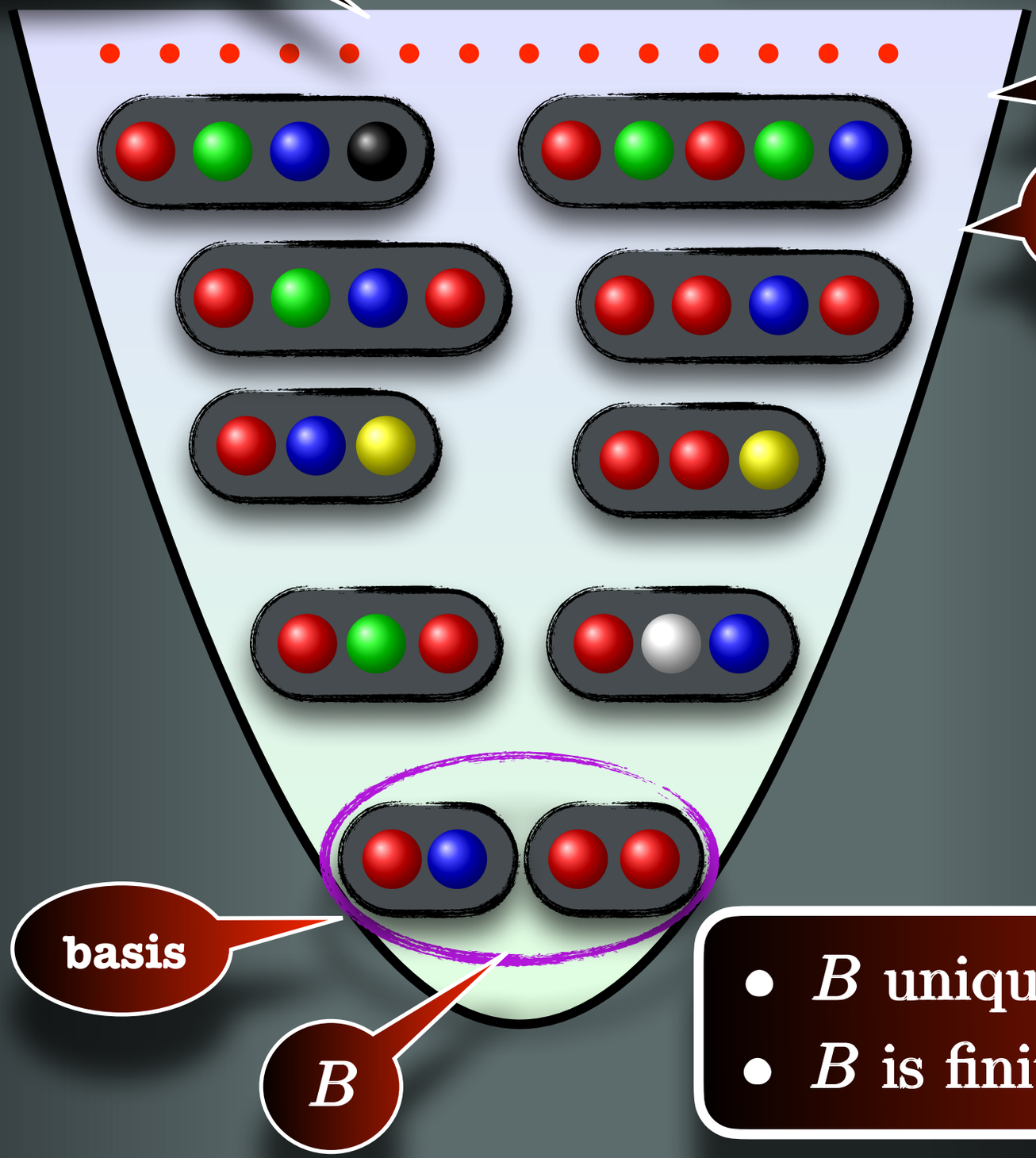


all configurations with



or

- if $c_1 \in U$ and $c_1 \sqsubseteq c_2$
- then $c_2 \in U$



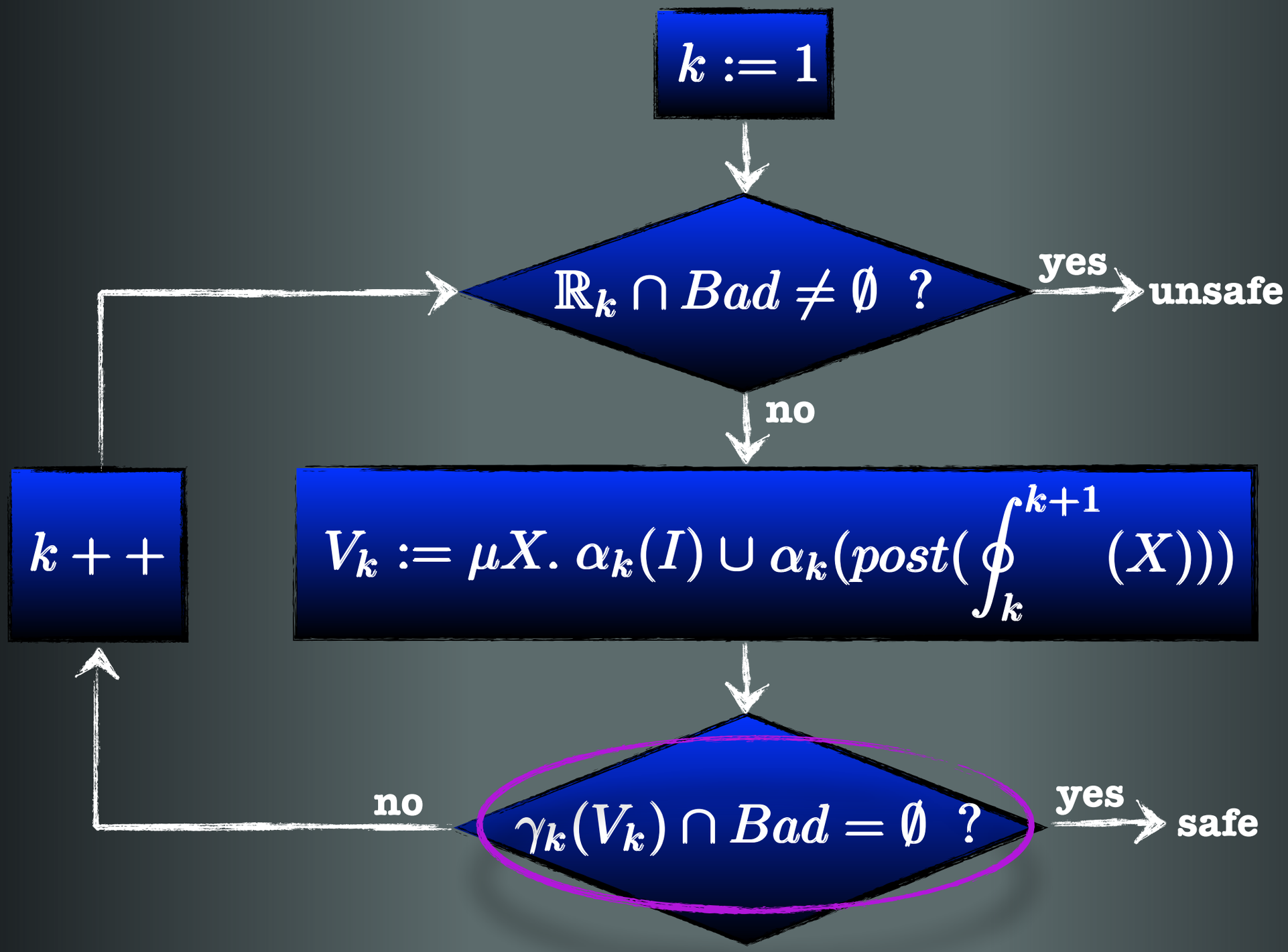
upward closed

U

basis

B

- B uniquely characterizes U
- B is finite


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$k++$
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$
no
$\gamma_k(V_k) \cap Bad = \emptyset$?
yes
safe

$\gamma_k(V_k) \cap Bad = \emptyset \ ?$

$k := 1$

$\mathbb{R}_k \cap Bad \neq \emptyset \ ?$

yes

unsafe

no

$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$

$k++$

$\gamma_k(V_k) \cap Bad = \emptyset \ ?$

no

yes

safe

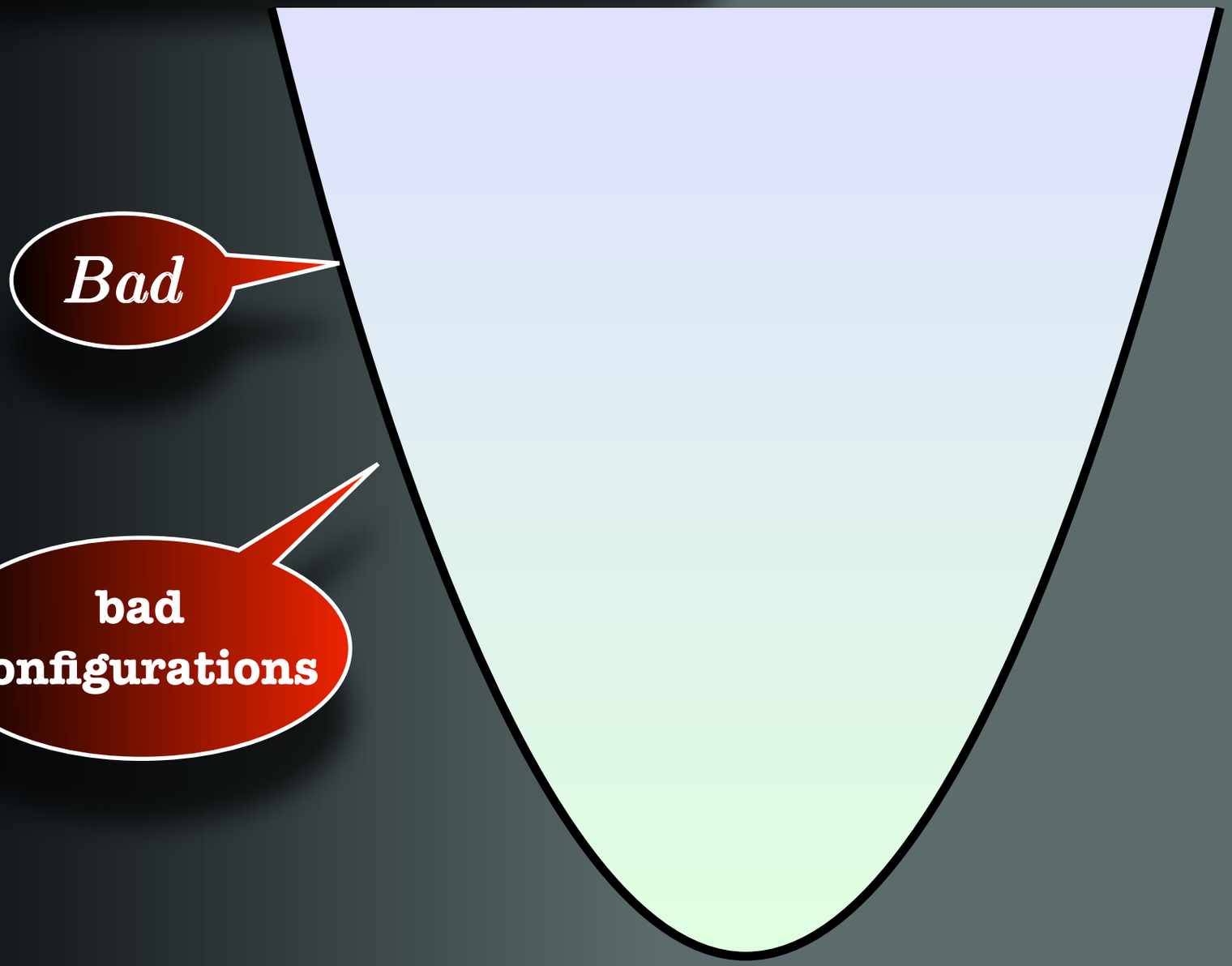
Bad configuration:

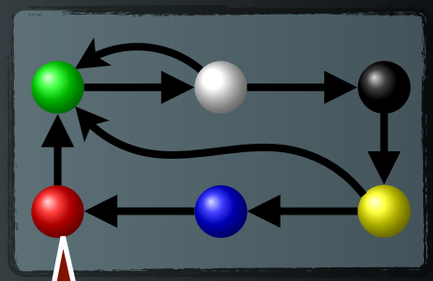
- **2 or more processes in critical section**

Set of bad configurations:

- **infinite, but**
- **“upward closed”**

$$\gamma_k(V_k) \cap Bad = \emptyset \ ?$$





critical section

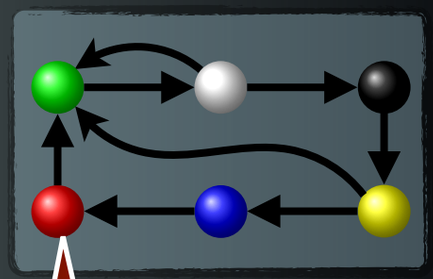
Bad configuration:

- **2 or more processes in critical section**

Set of bad configurations:

- **infinite, but**
- **“upward closed”**

$$\gamma_k(V_k) \cap Bad = \emptyset \ ?$$



critical section

Bad

bad configurations

basis

B

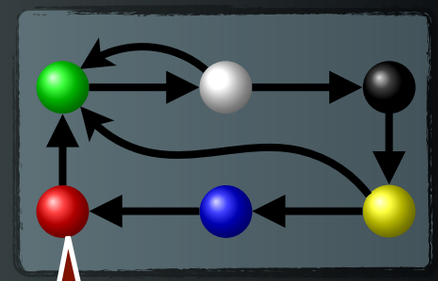
Bad configuration:

- **2 or more processes in critical section**

Set of bad configurations:

- **infinite, but**
- **“upward closed”**

$$\gamma_k(V_k) \cap Bad = \emptyset \ ?$$



critical section

Bad

bad configurations

basis

B

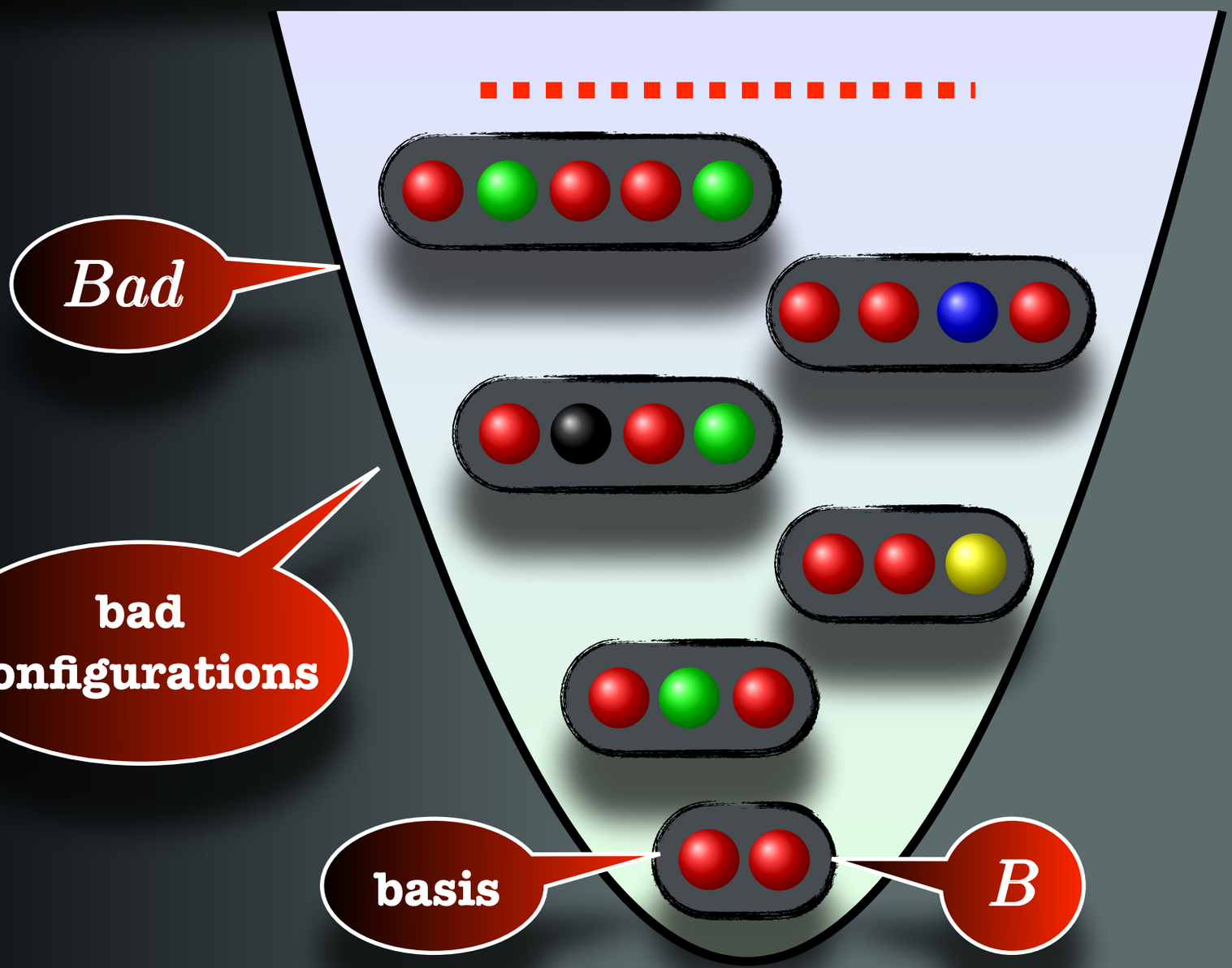
Bad configuration:

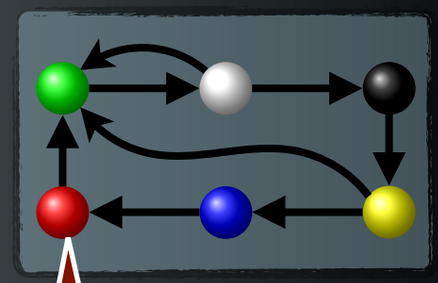
- **2 or more processes in critical section**

Set of bad configurations:

- **infinite, but**
- **“upward closed”**

$$\gamma_k(V_k) \cap Bad = \emptyset \ ?$$





critical section

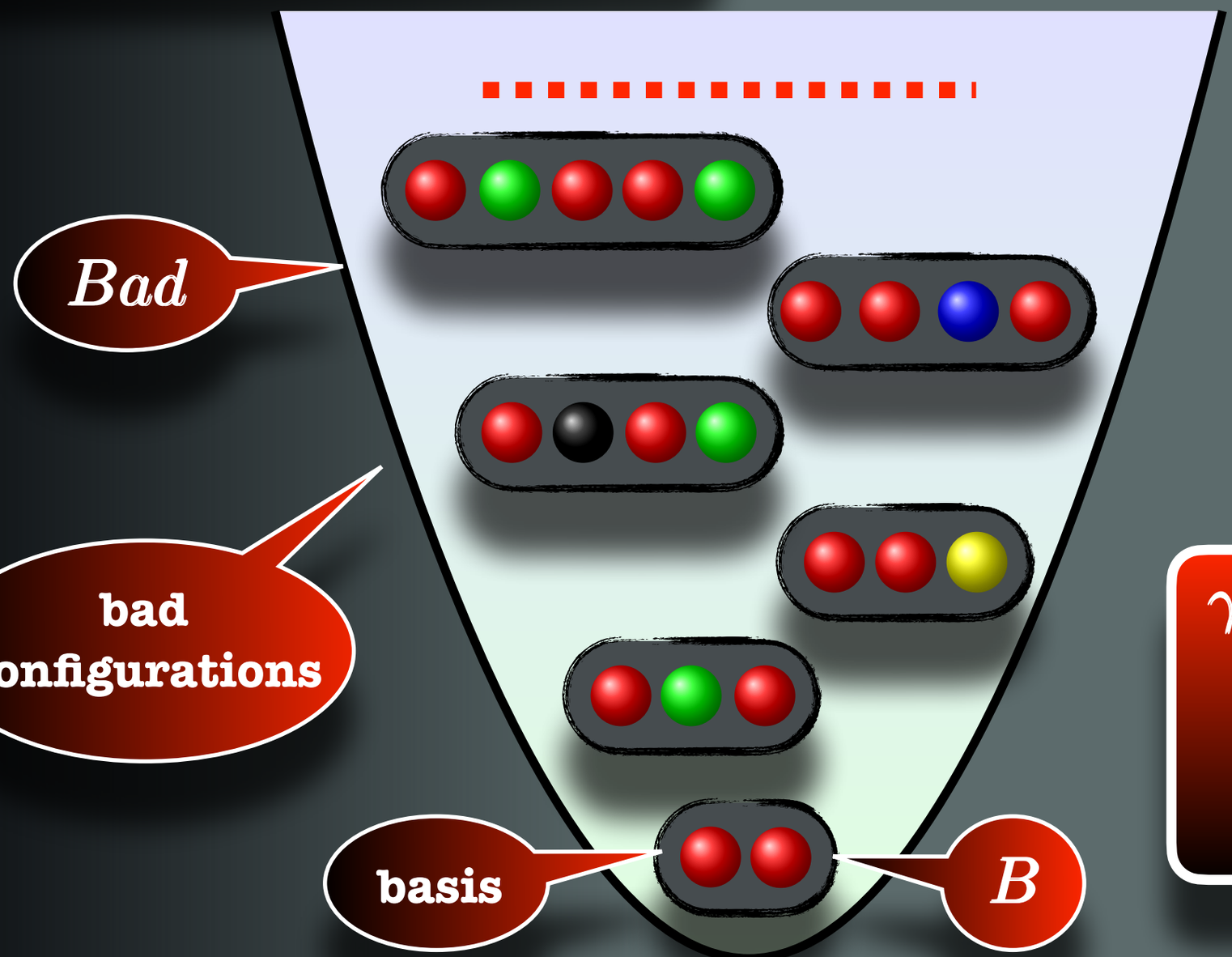
Bad configuration:

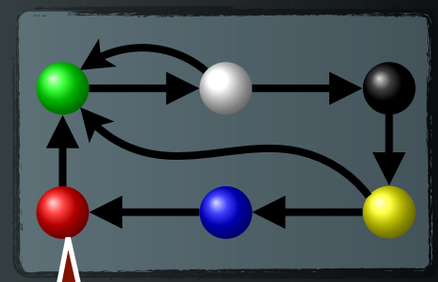
- **2 or more processes in critical section**

Set of bad configurations:

- **infinite, but**
- **“upward closed”**

$$\gamma_k(V_k) \cap Bad = \emptyset \ ?$$



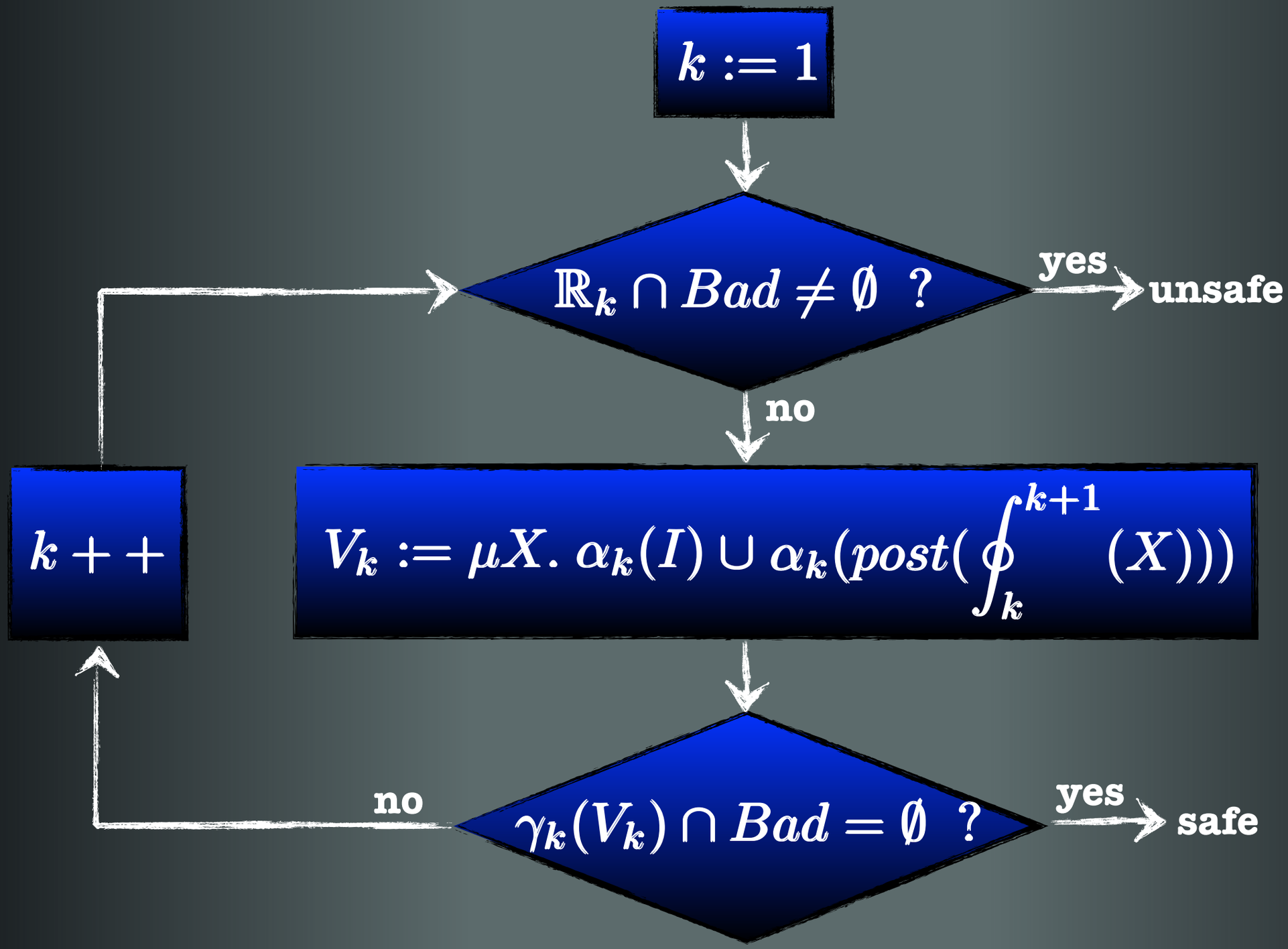


critical section

$$\gamma_k(V_k) \cap Bad \neq \emptyset$$

iff

$$\alpha_k(B) \subseteq V_k$$


$k := 1$
$\mathbb{R}_k \cap Bad \neq \emptyset$?
yes
unsafe
no
$k++$
$V_k := \mu X.\ \alpha_k(I) \cup \alpha_k(post(\oint_k^{k+1}(X)))$
$\gamma_k(V_k) \cap Bad = \emptyset$?
no
yes
safe

Parameterized Systems

Analysis / Abstraction

Features

Completeness

Parameterized Systems ✓

Analysis / Abstraction ✓

Features

Completeness

Features

non-atomic global conditions

shared variables

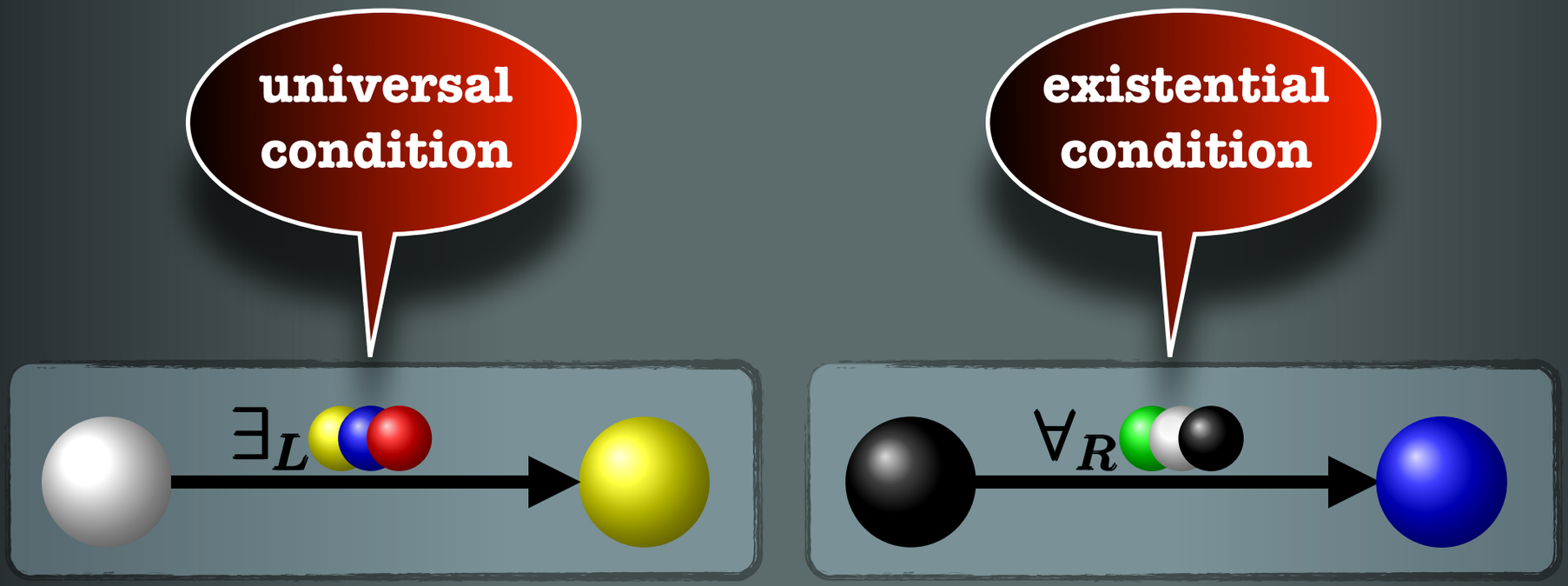
broadcast

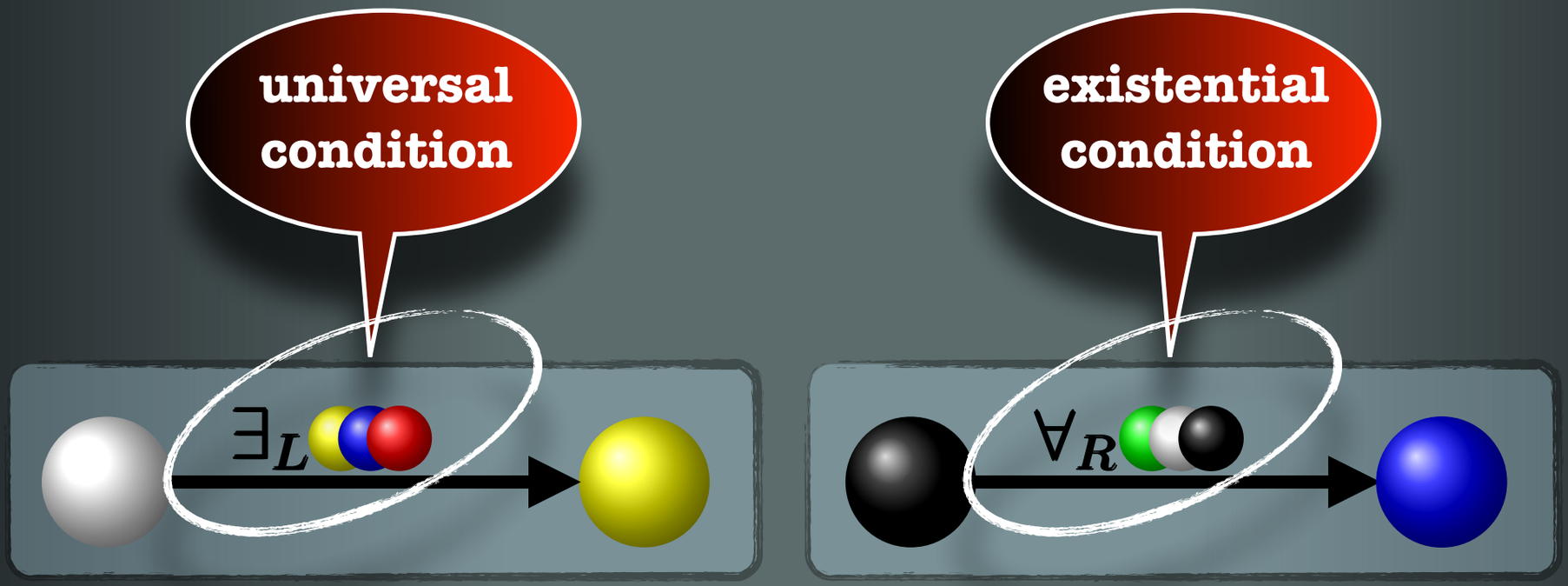
rendez-vous

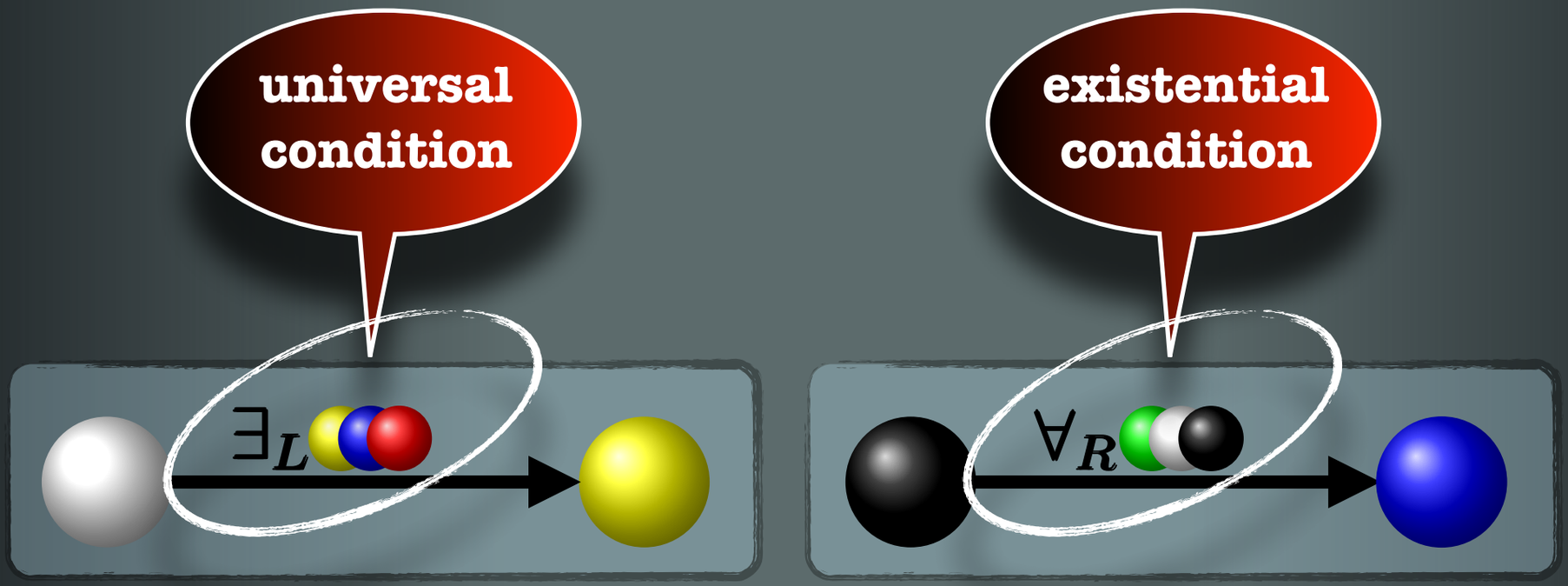
process creation/deletion

Topologies

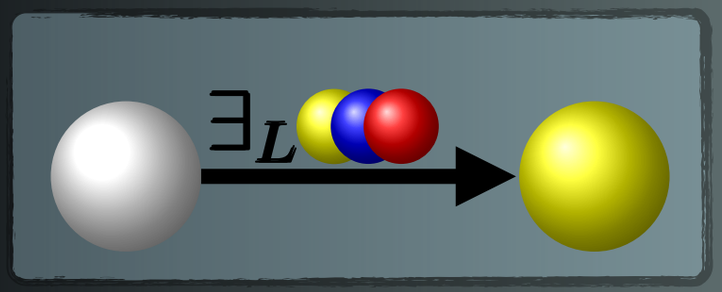
- linear
- ring
- tree
- multiset

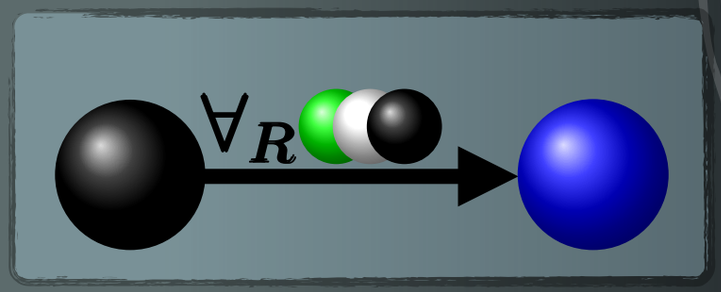

universal condition
existential condition
$\exists_L$
$\forall_R$

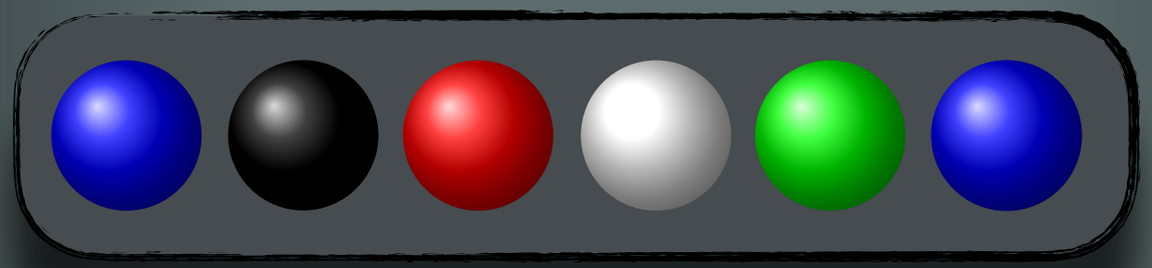

universal condition
existential condition
$\exists_L$
$\forall_R$

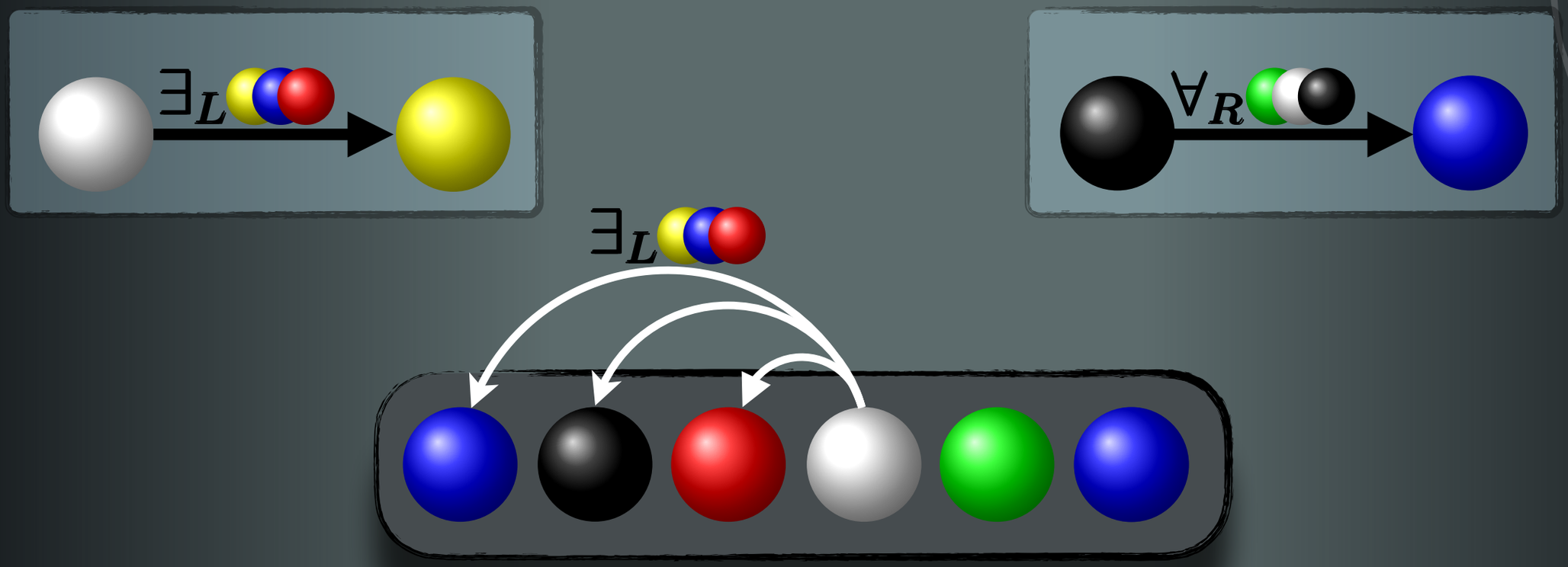


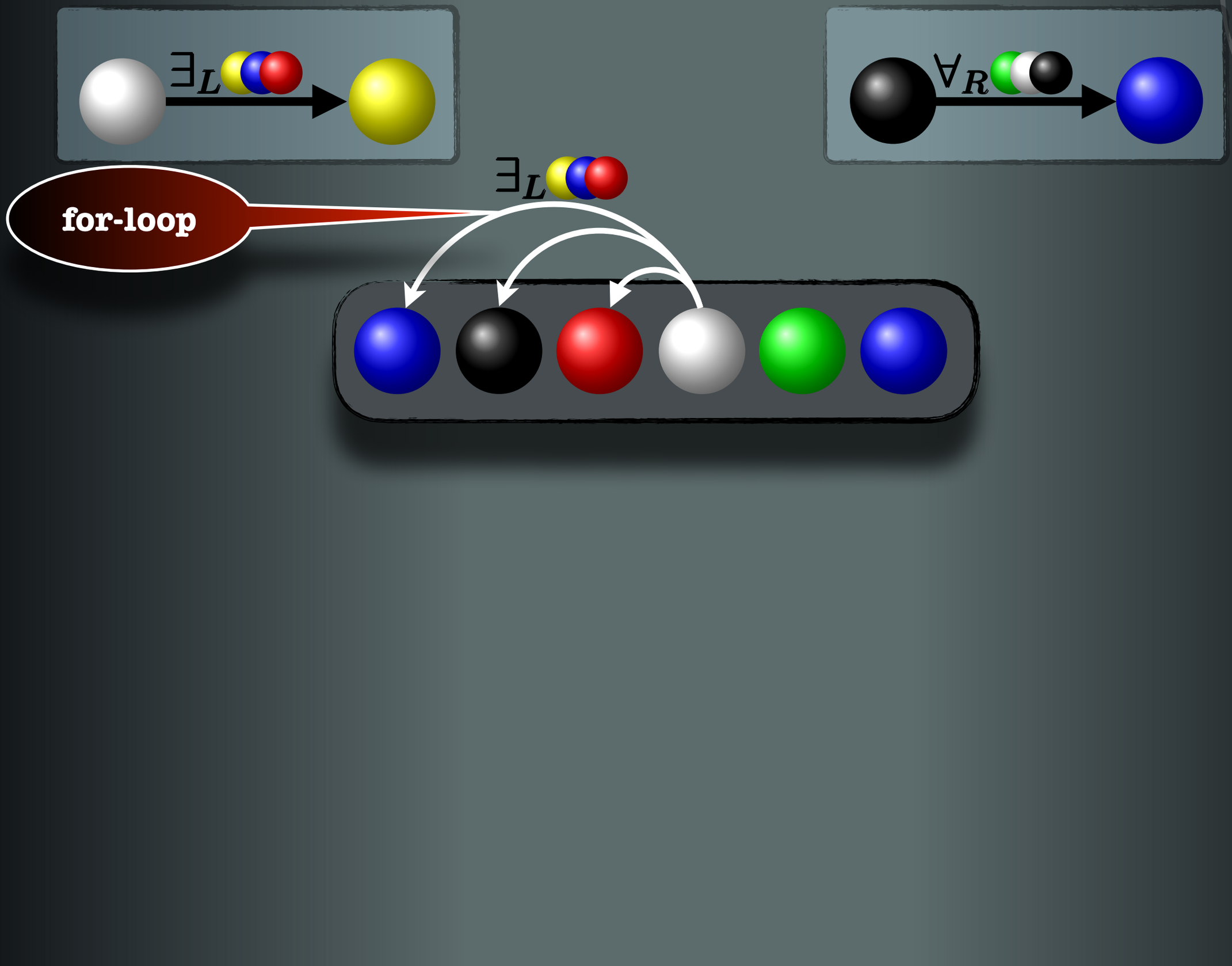
non-atomic global conditions

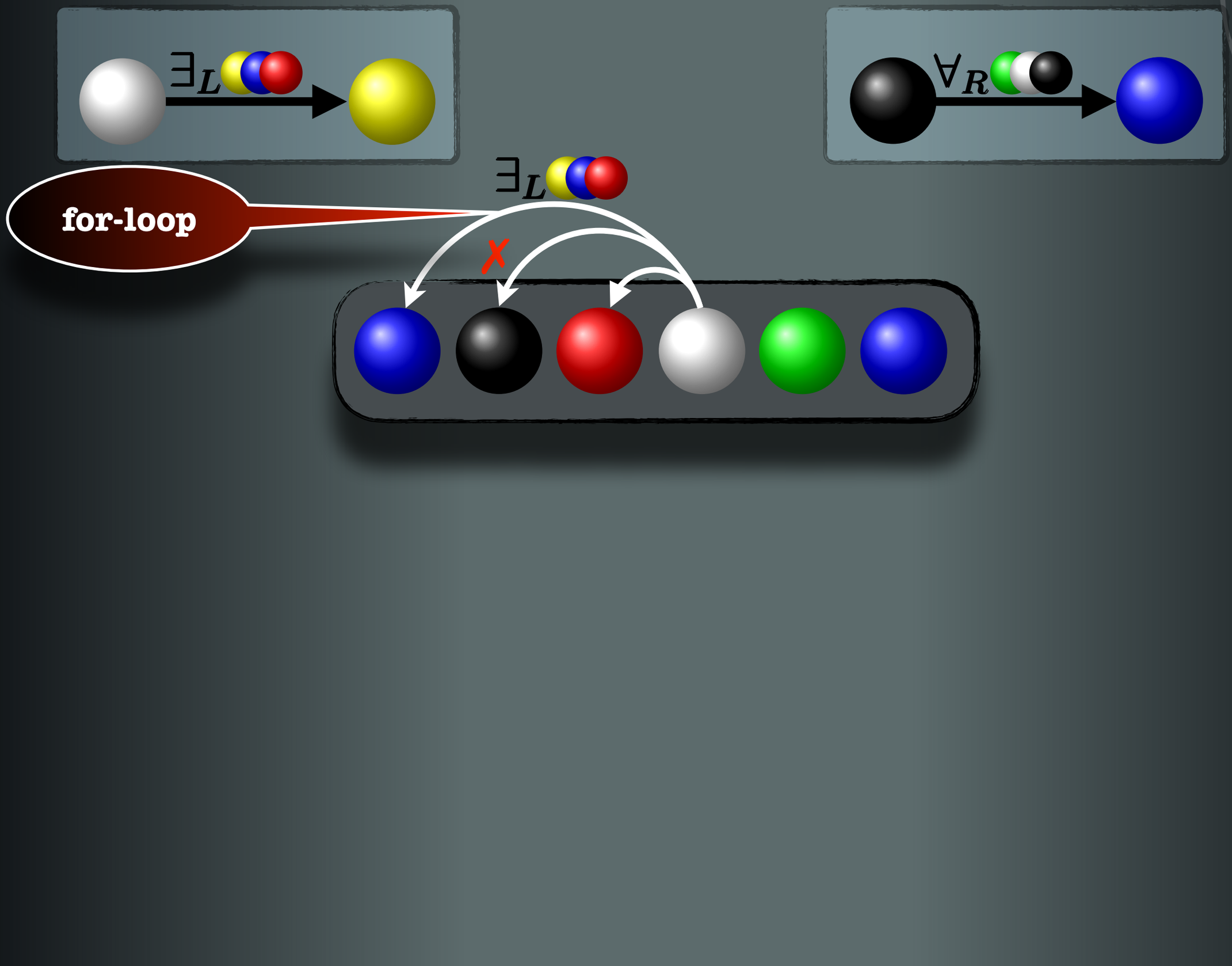

$\exists_L$

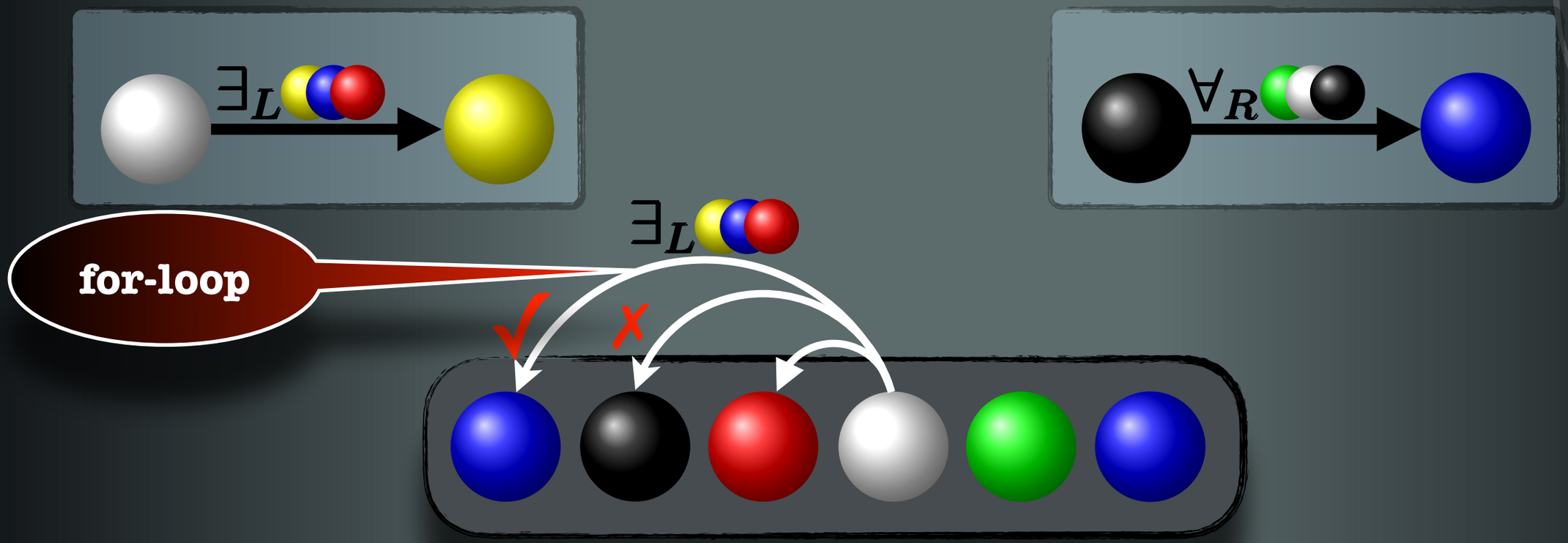

$\forall_R$

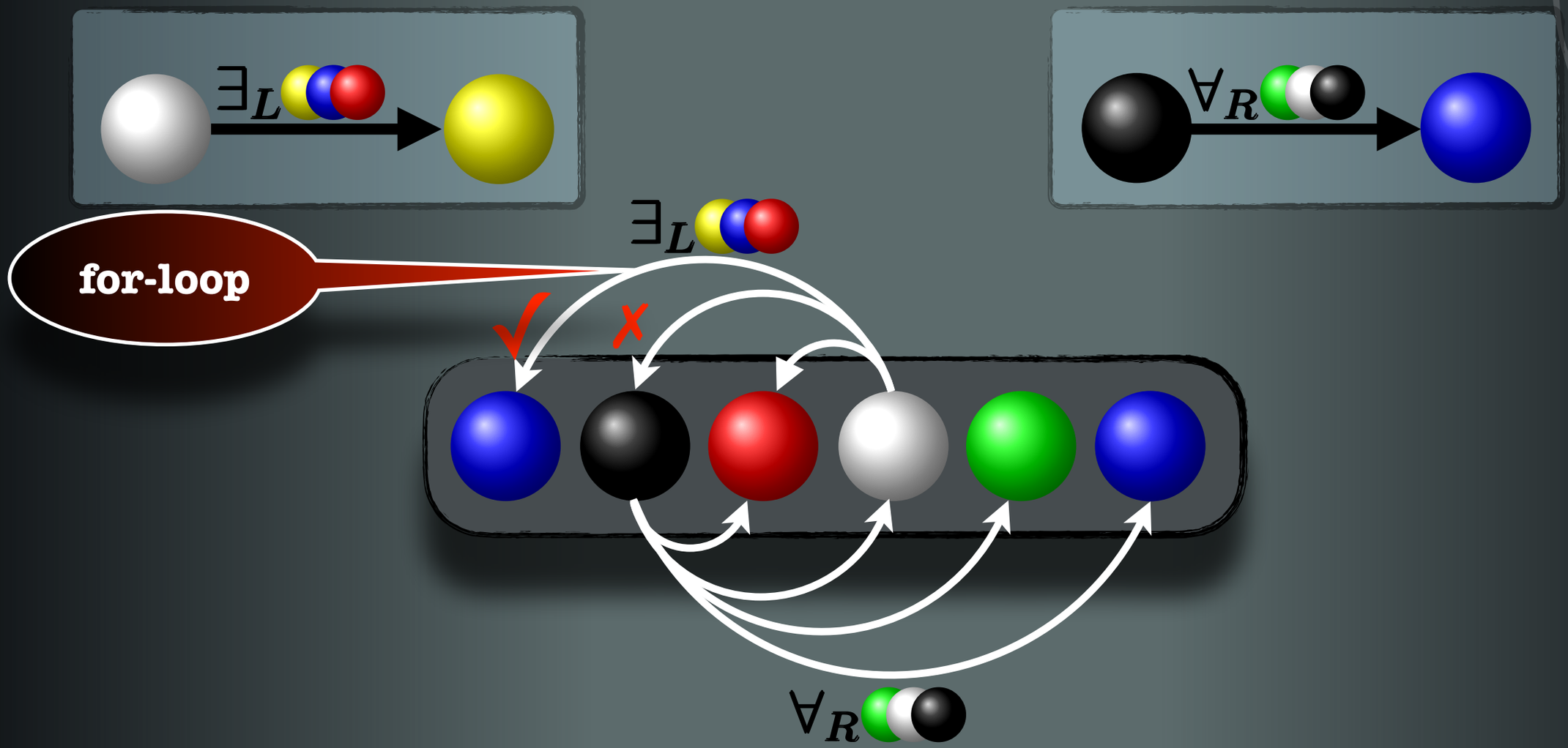


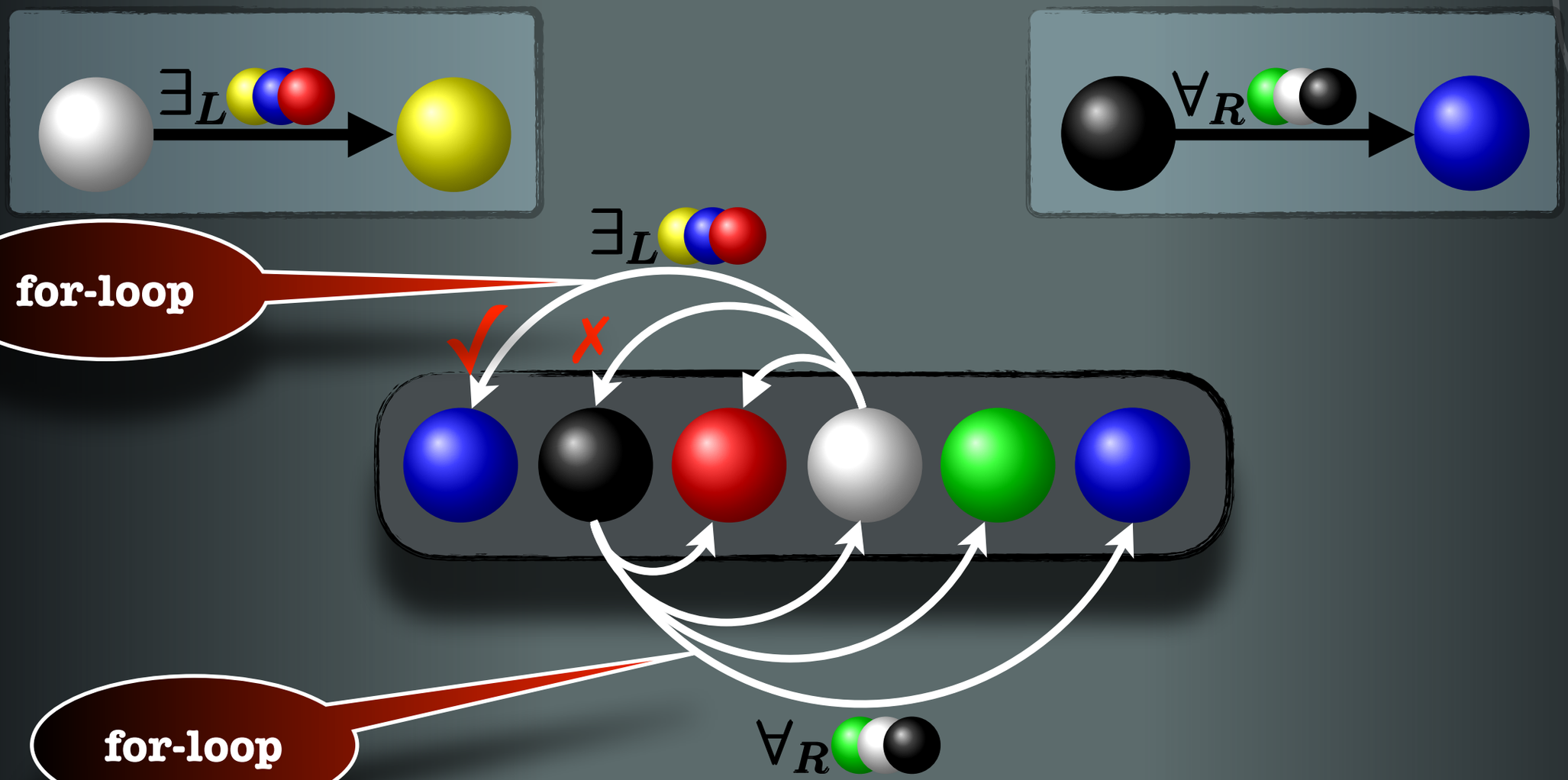

$\exists_L$
$\forall_R$
$\exists_L$

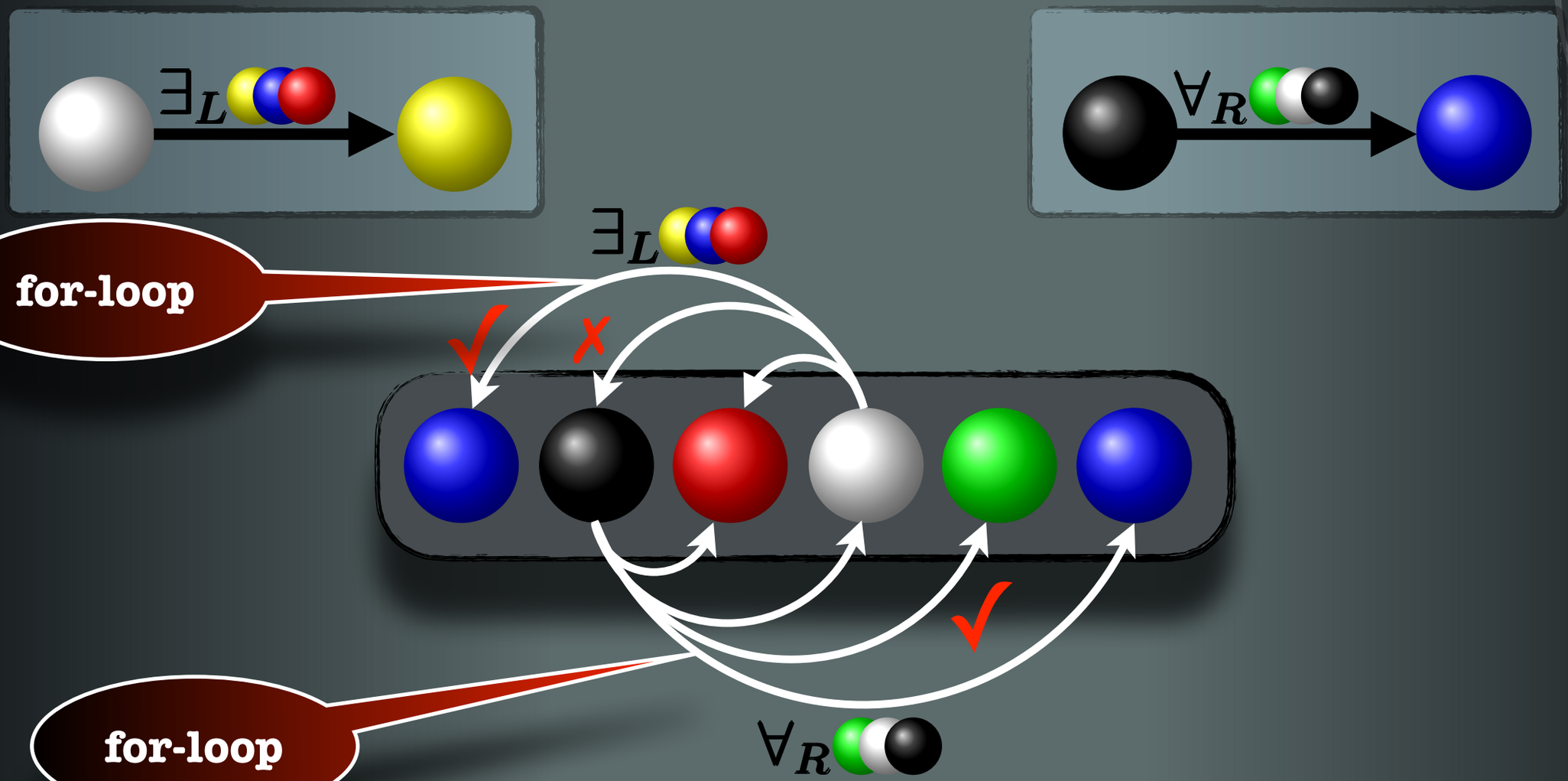

$\exists_L$
$\forall_R$
$\exists_L$
for-loop

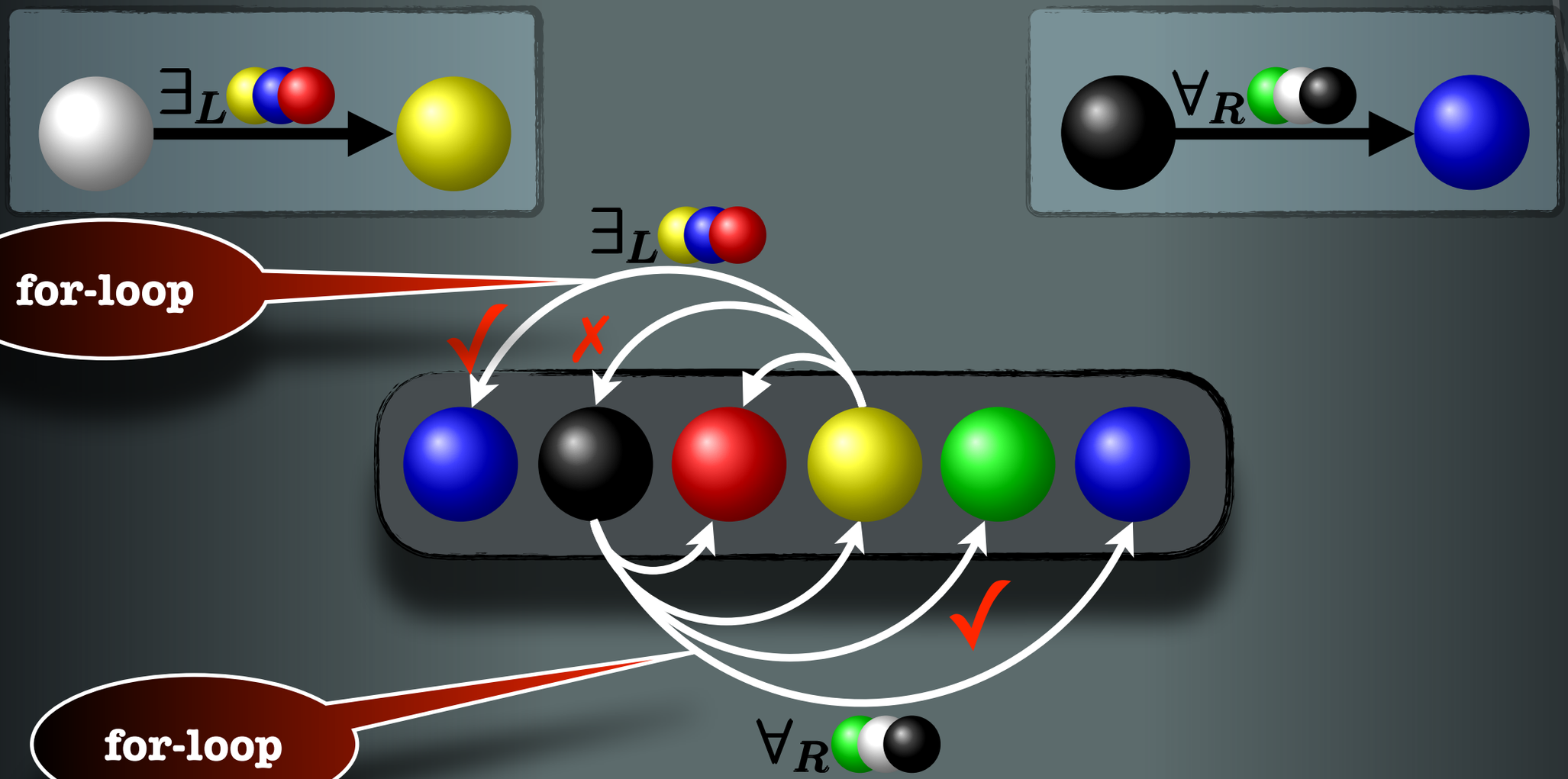

$\exists_L$
$\forall_R$
$\exists_L$
for-loop
✗

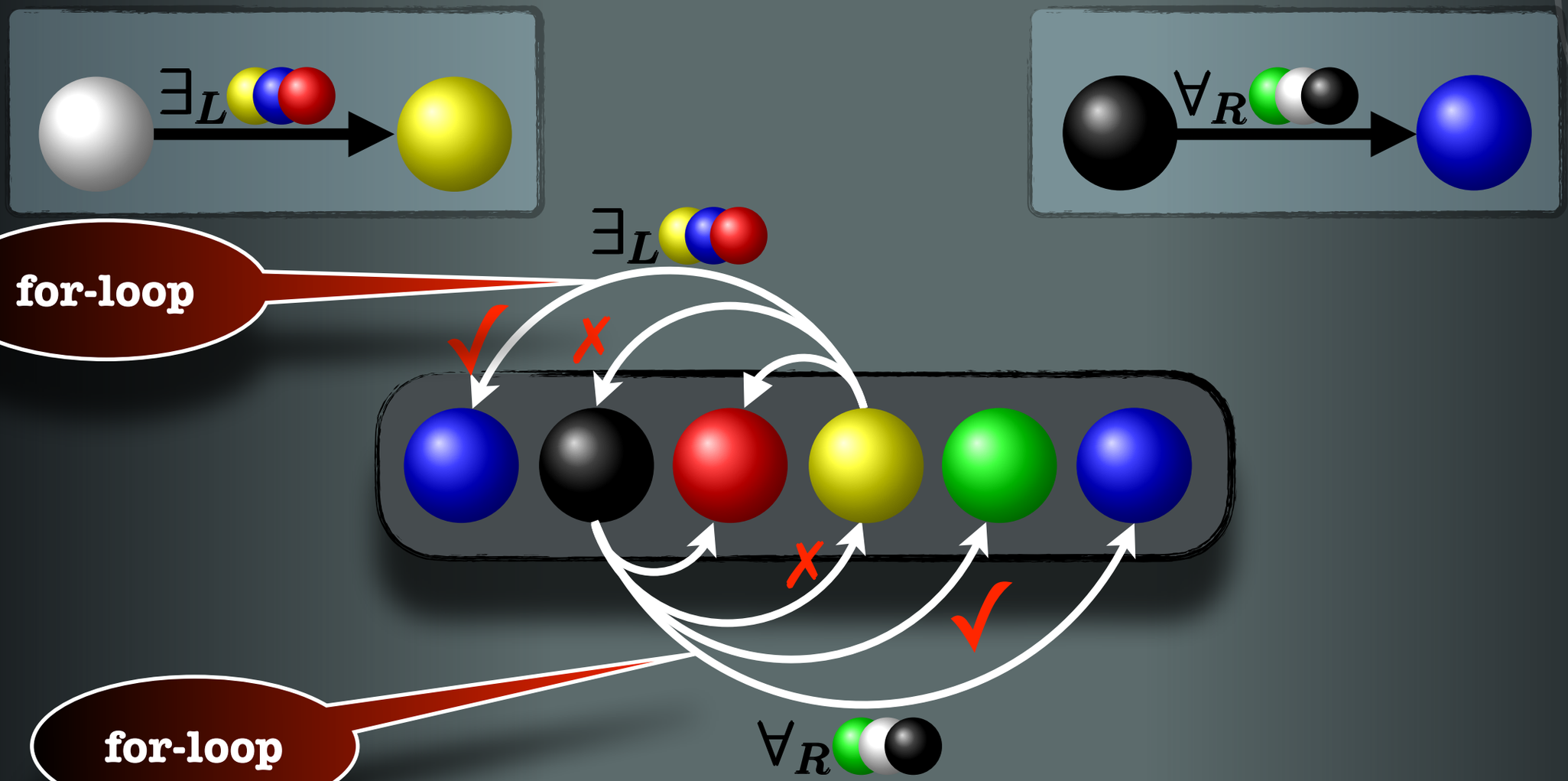

$\exists_L$
$\forall_R$
$\exists_L$
for-loop
✓
✗

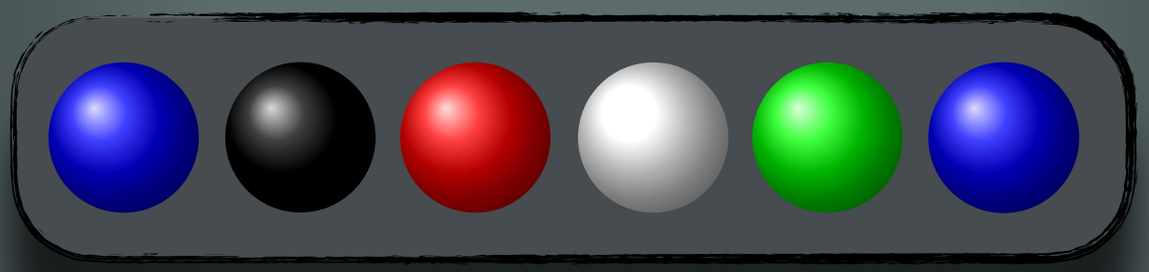

$\exists_L$
$\forall_R$
for-loop
$\exists_L$
✓
✗
$\forall_R$


$\exists_L$
$\forall_R$
for-loop
$\exists_L$
✓
✗
for-loop
$\forall_R$

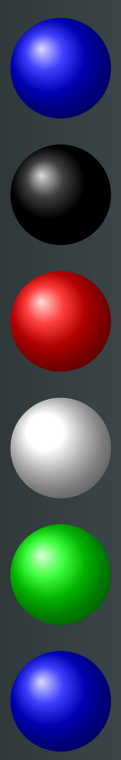

$\exists_L$
$\forall_R$
for-loop
$\exists_L$
✓
✗
✓
for-loop
$\forall_R$

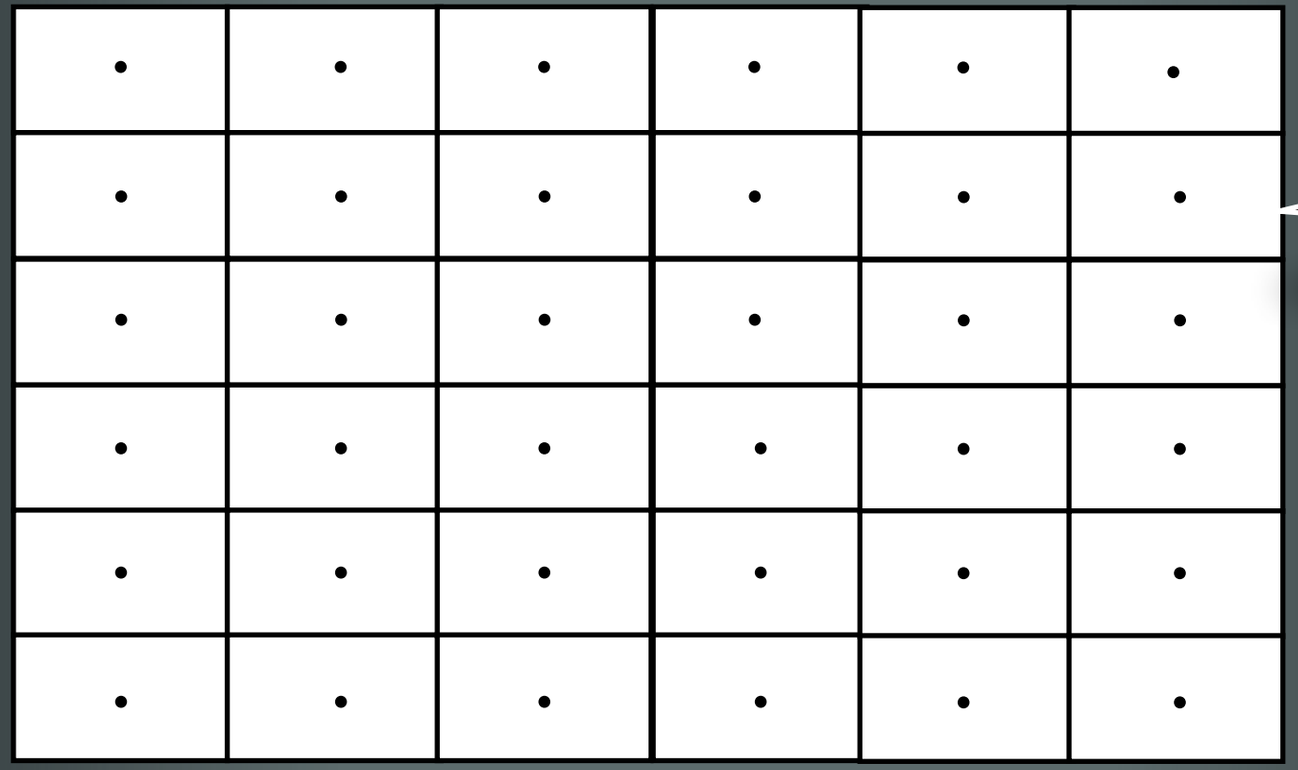

$\exists_L$
$\forall_R$
for-loop
$\exists_L$
✓
✗
✓
for-loop
$\forall_R$


$\exists_L$
$\forall_R$
for-loop
$\exists_L$
✓
✗
✗
✓
for-loop
$\forall_R$





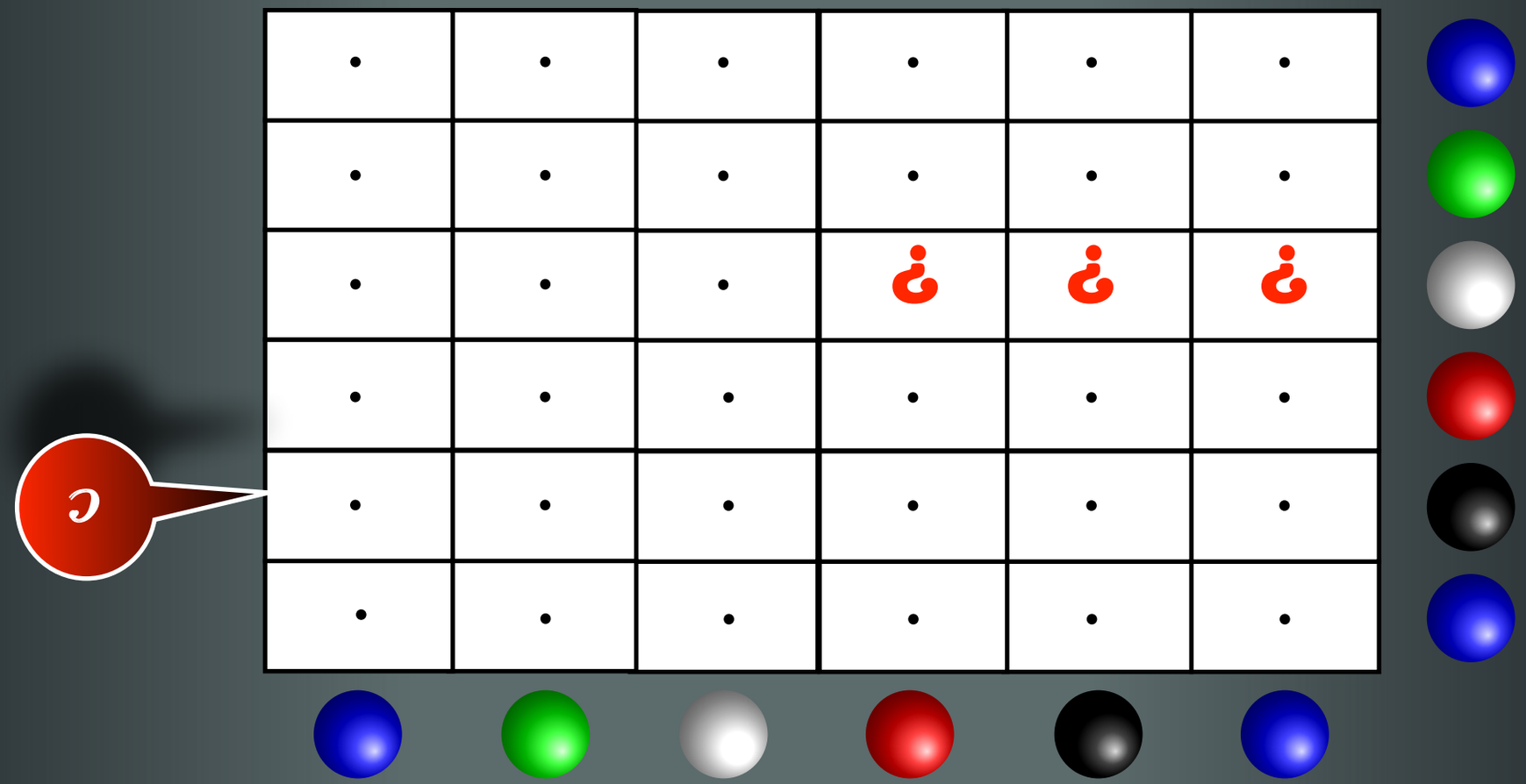


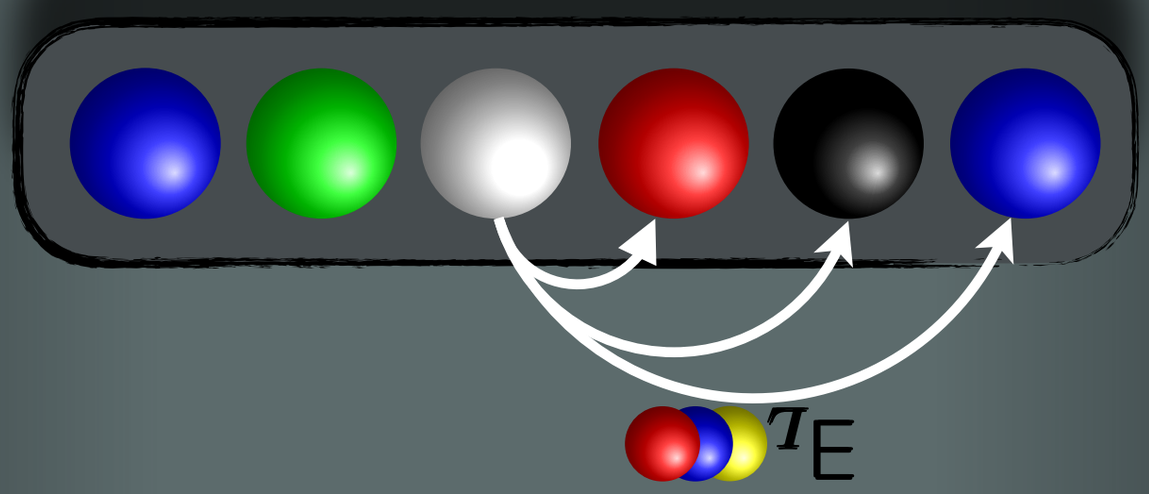


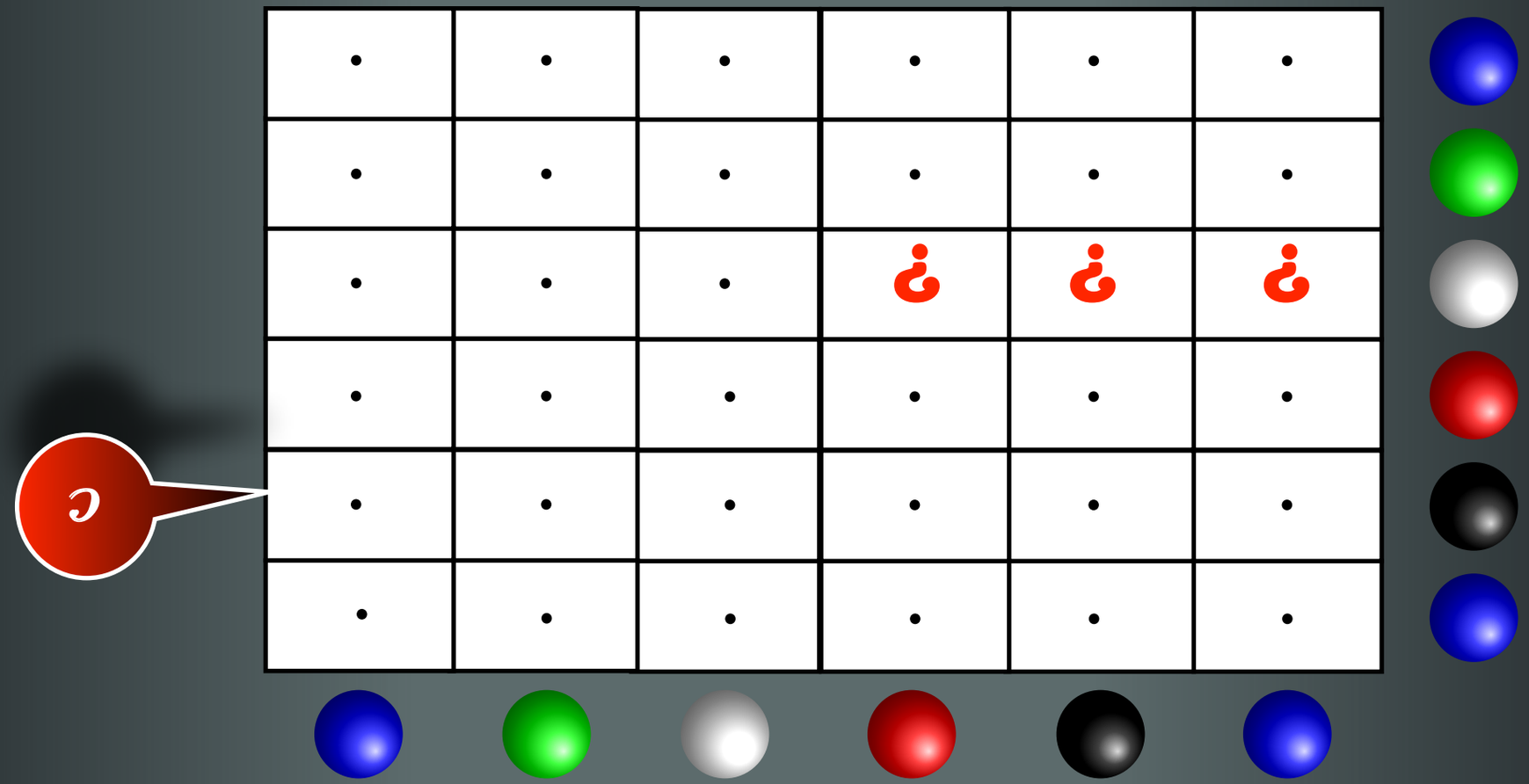

c

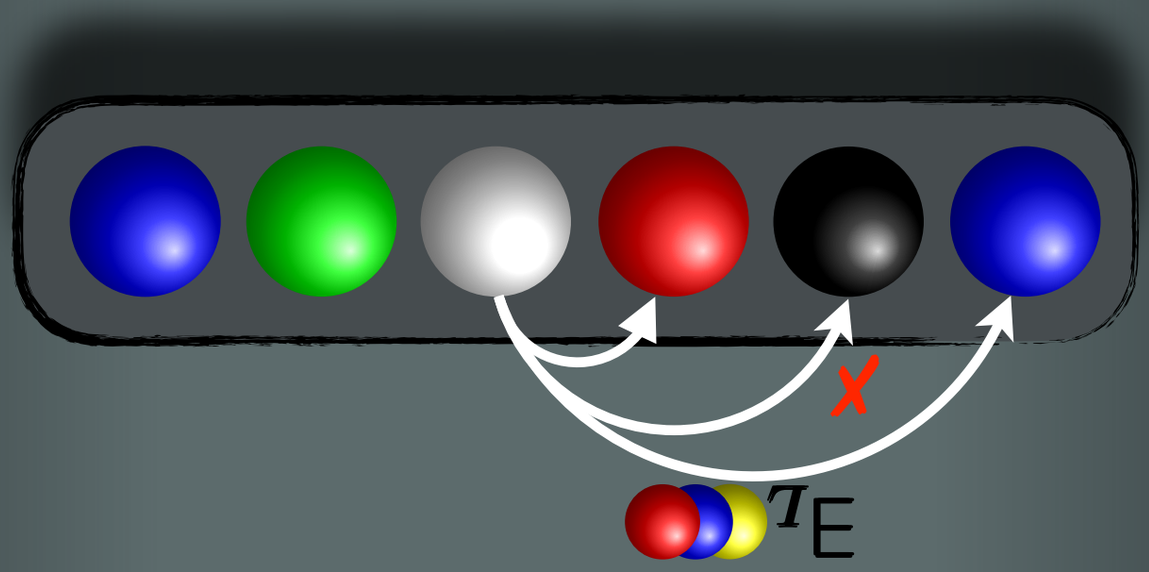
c

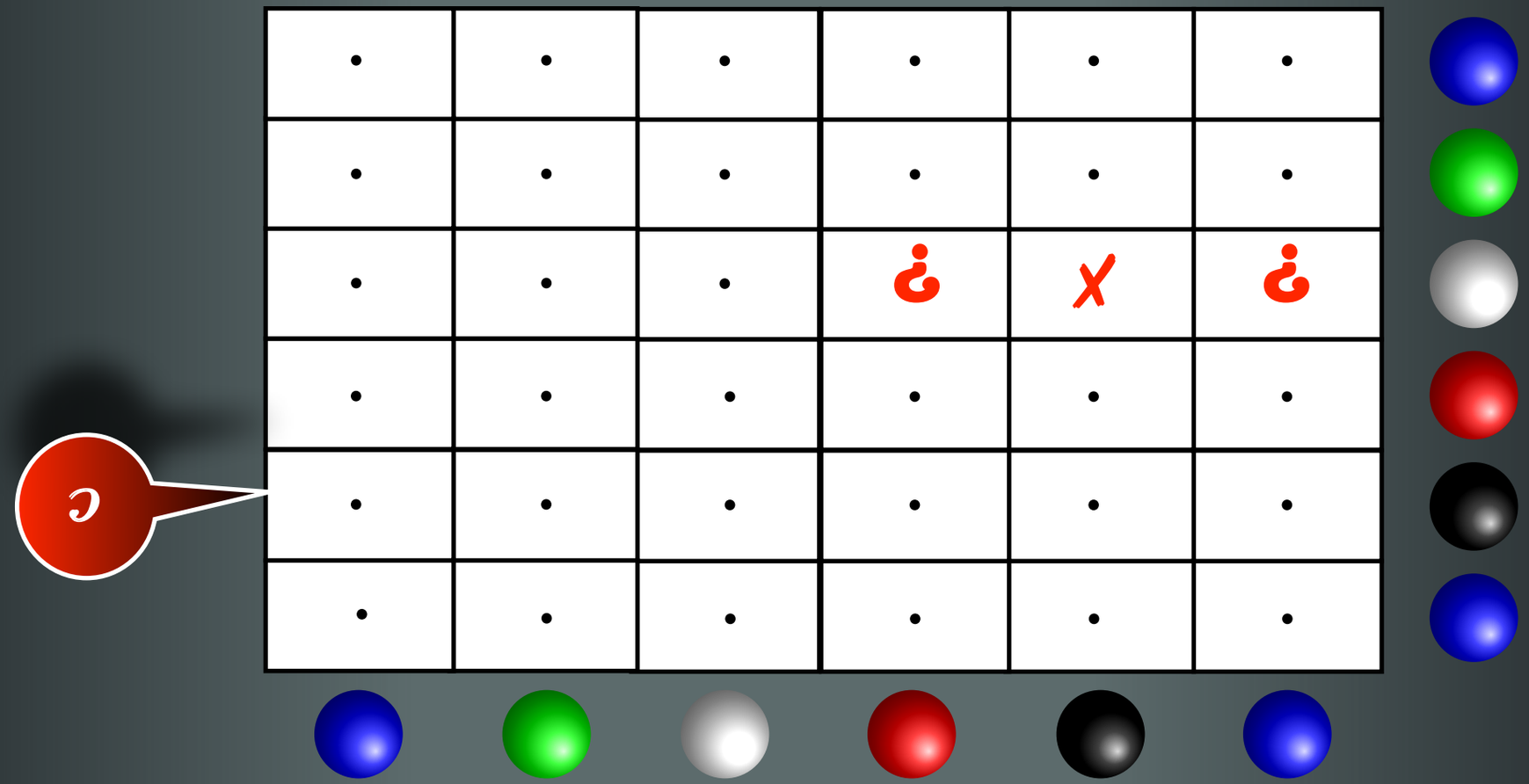
τ_E

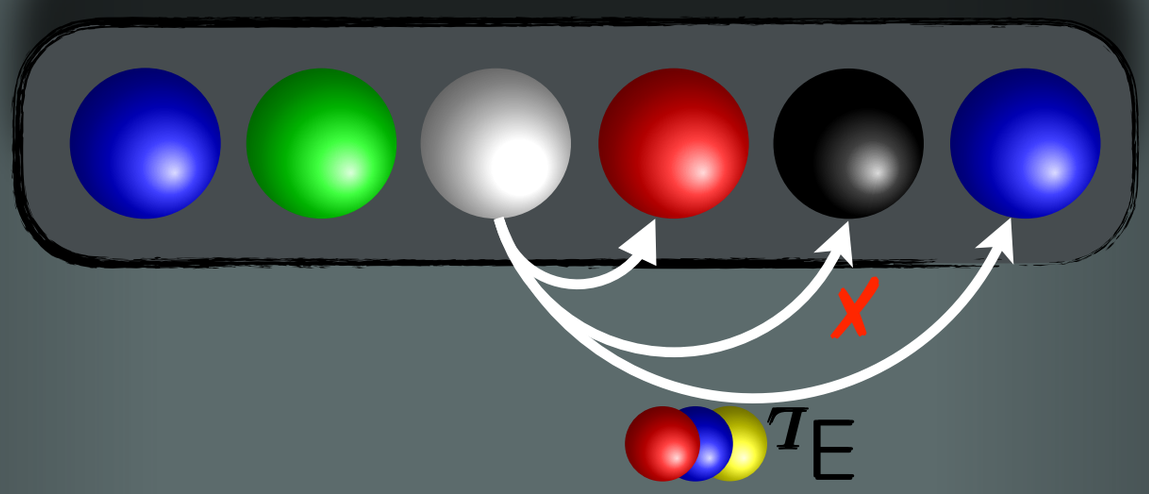


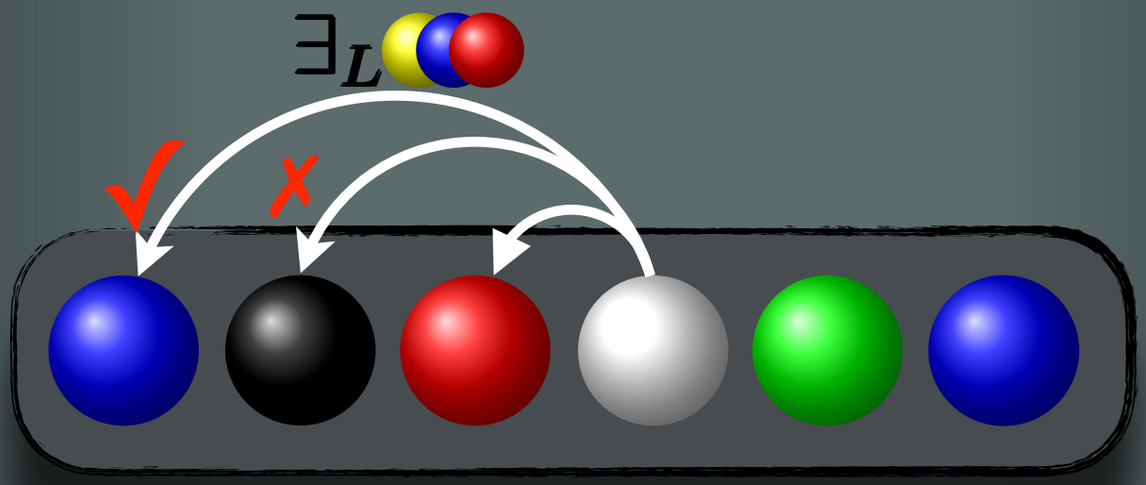

τ_E


c

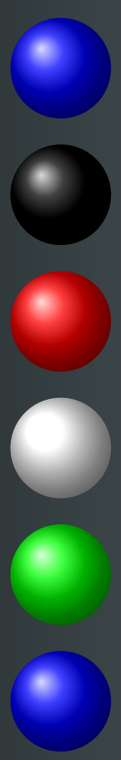

X
τ_E


c


X
τ_E


$\exists_L$
✓
✗



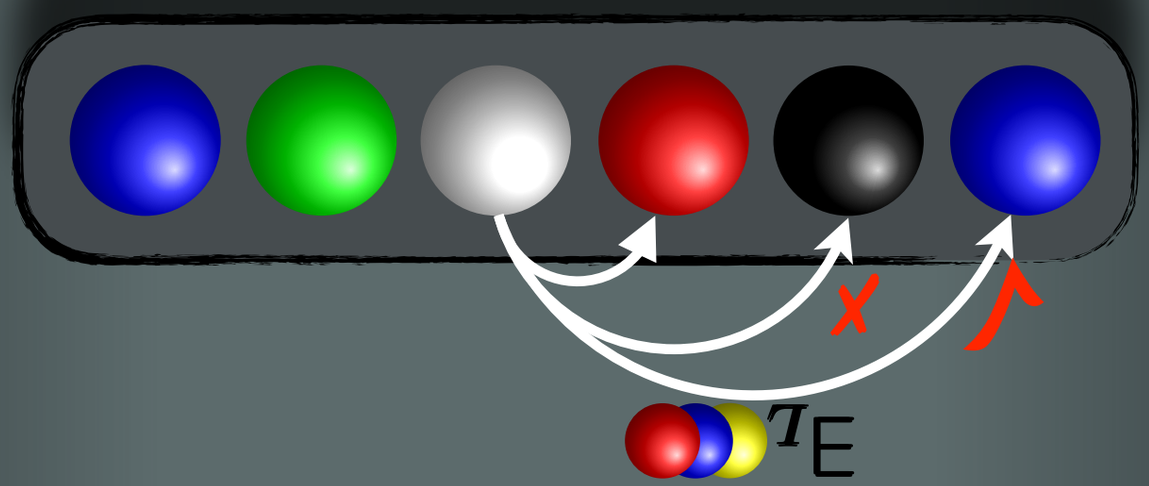


·	·	·	·	·	·
·	·	·	·	·	·
·	·	·	·	·	·
?	✗	?	·	·	·
·	·	·	·	·	·
·	·	·	·	·	·

c

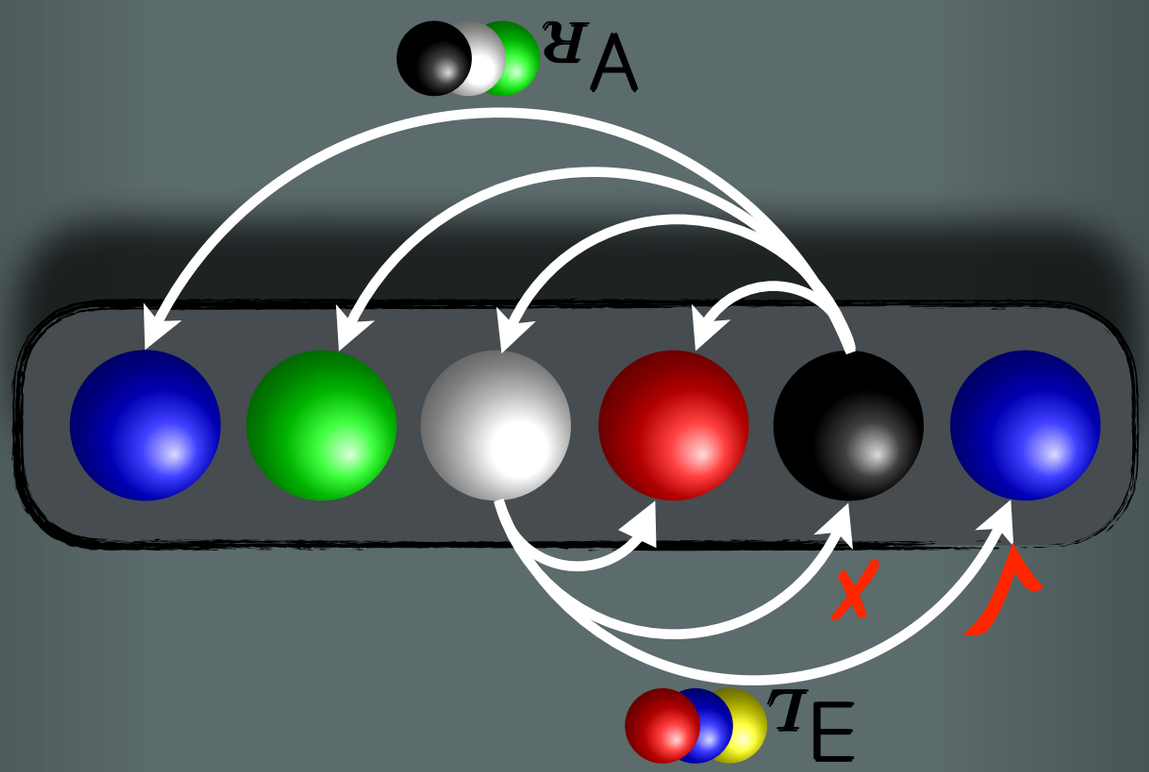
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	?	X	Y
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•

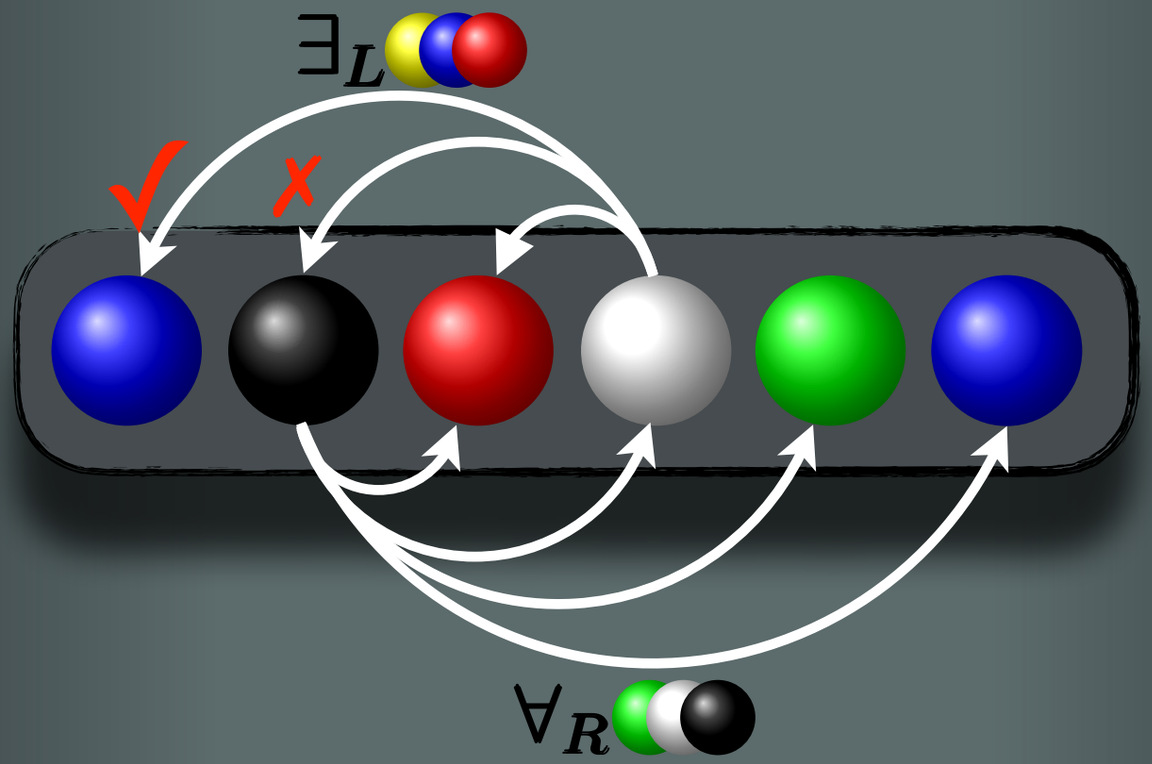
C



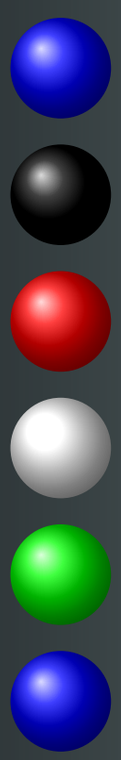
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	?	X	٨
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•

c



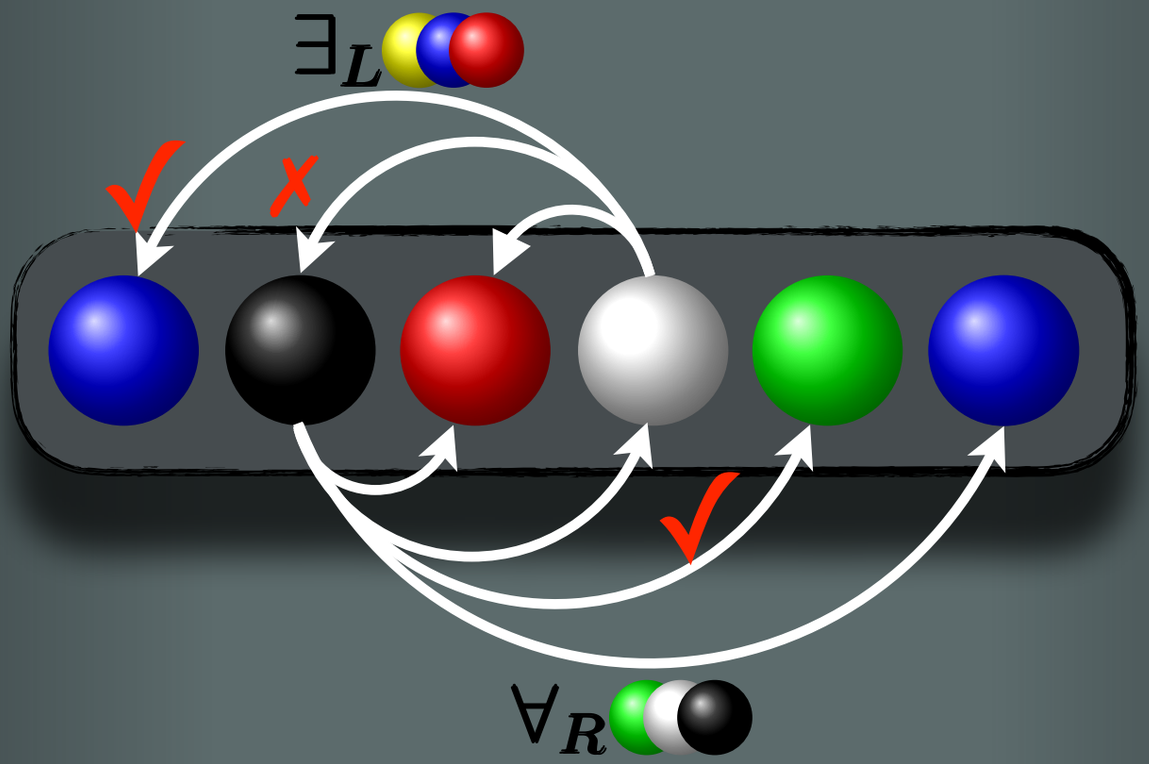




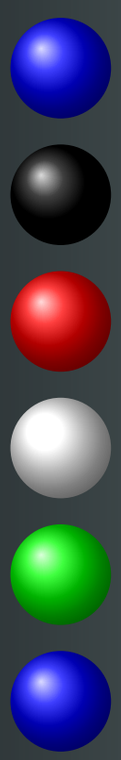


·	·	·	·	·	·
·	·	?	?	?	?
·	·	·	·	·	·
✓	✗	?	·	·	·
·	·	·	·	·	·
·	·	·	·	·	·

c

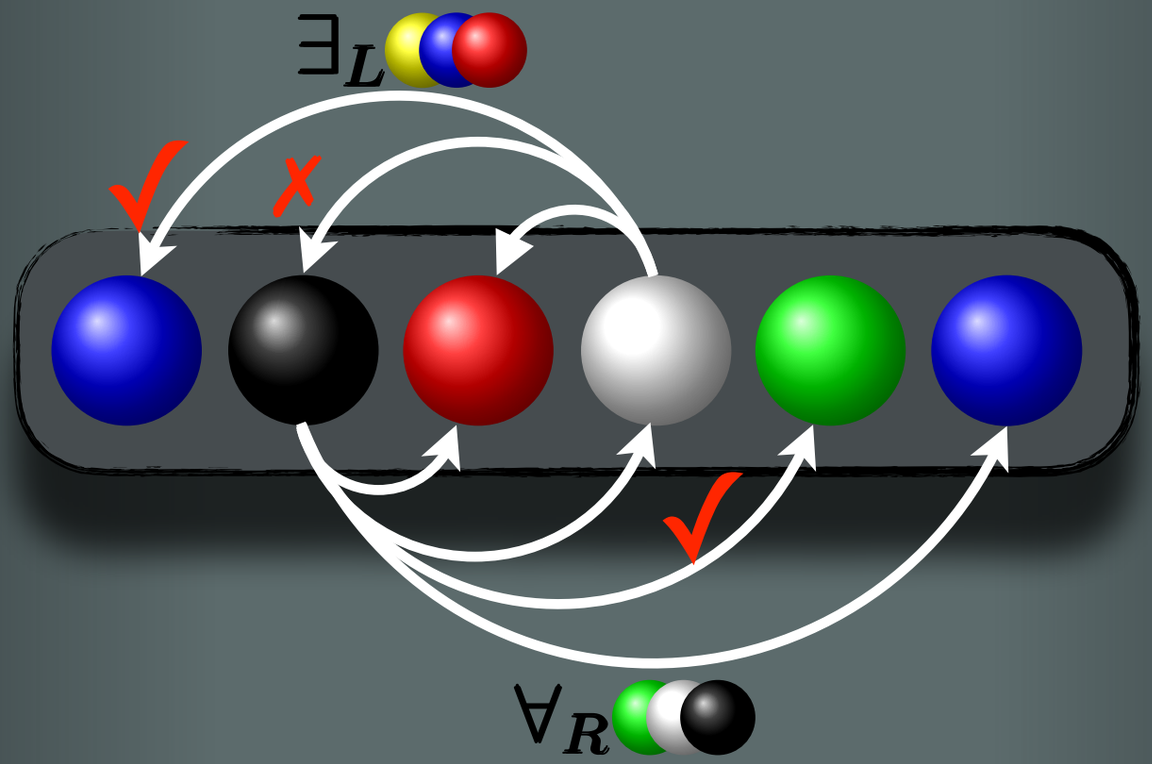




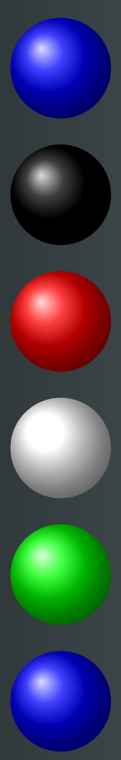


·	·	·	·	·	·
·	·	?	?	?	?
·	·	·	·	·	·
✓	✗	?	·	·	·
·	·	·	·	·	·
·	·	·	·	·	·

c

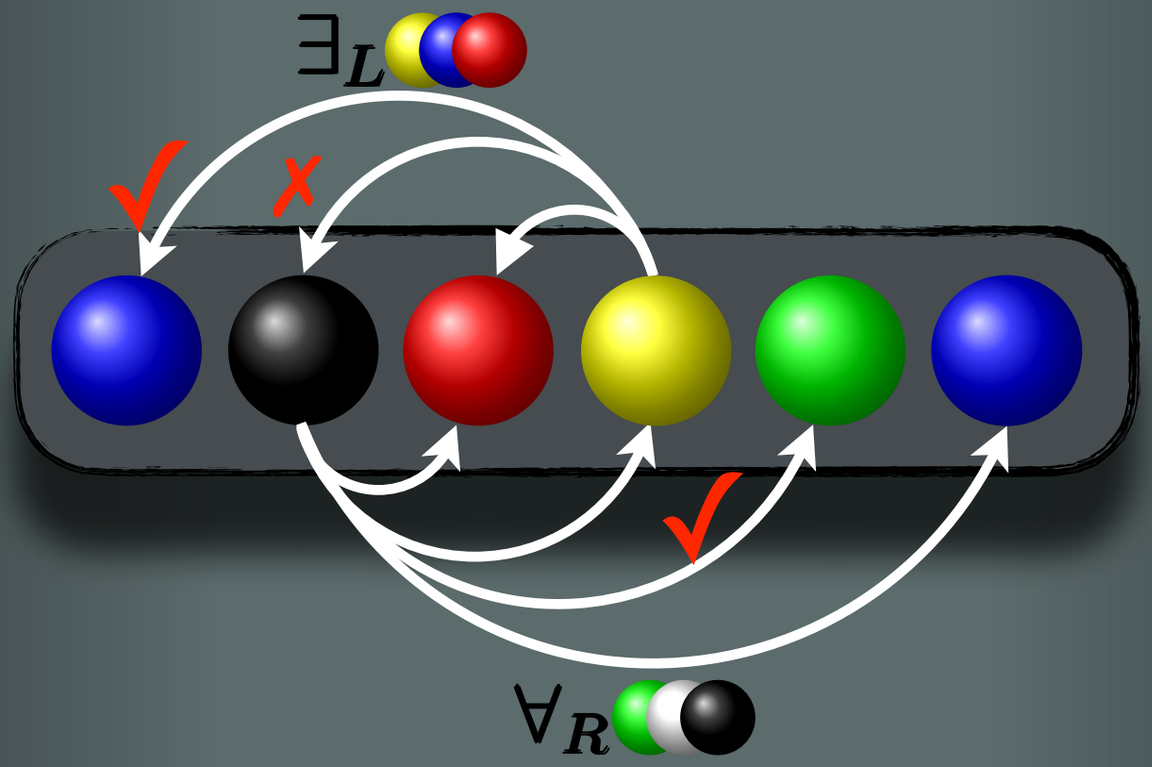




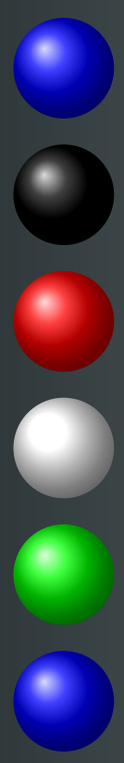


·	·	·	·	·	·
·	·	?	?	✓	?
·	·	·	·	·	·
✓	✗	?	·	·	·
·	·	·	·	·	·
·	·	·	·	·	·

c

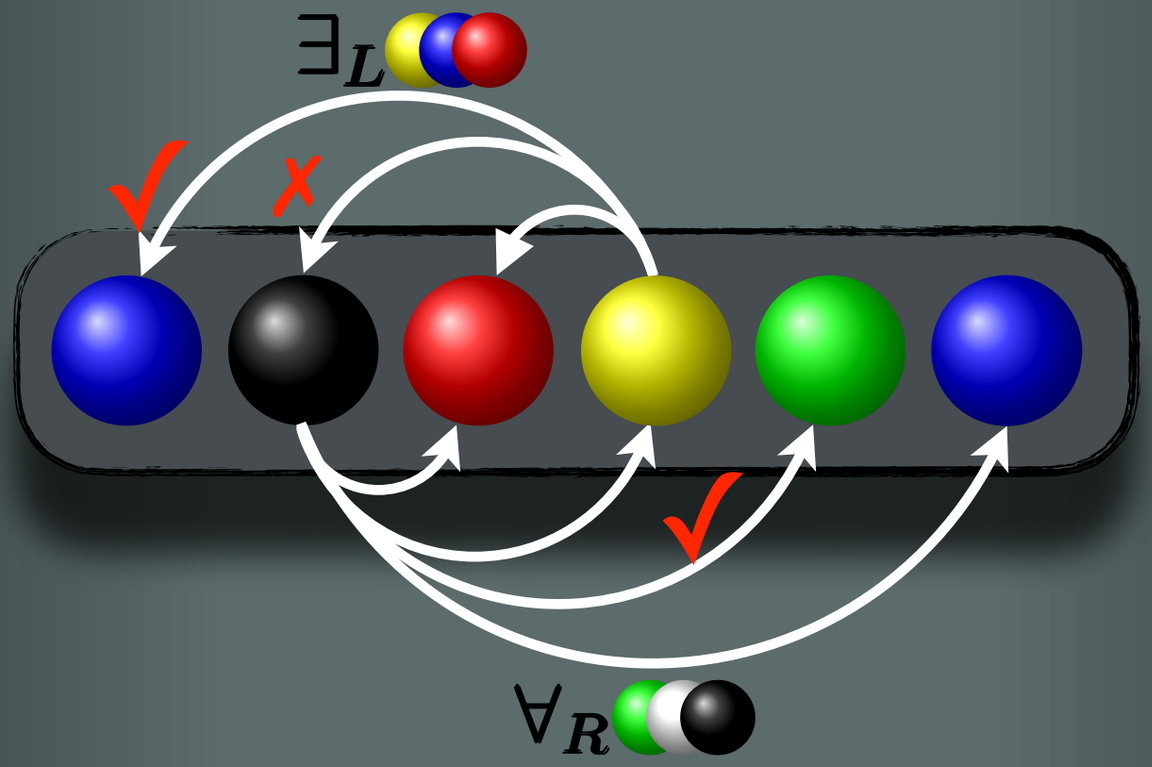




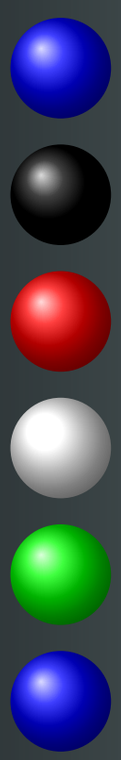


·	·	·	·	·	·
·	·	?	?	✓	?
·	·	·	·	·	·
✓	✗	?	·	·	·
·	·	·	·	·	·
·	·	·	·	·	·

c

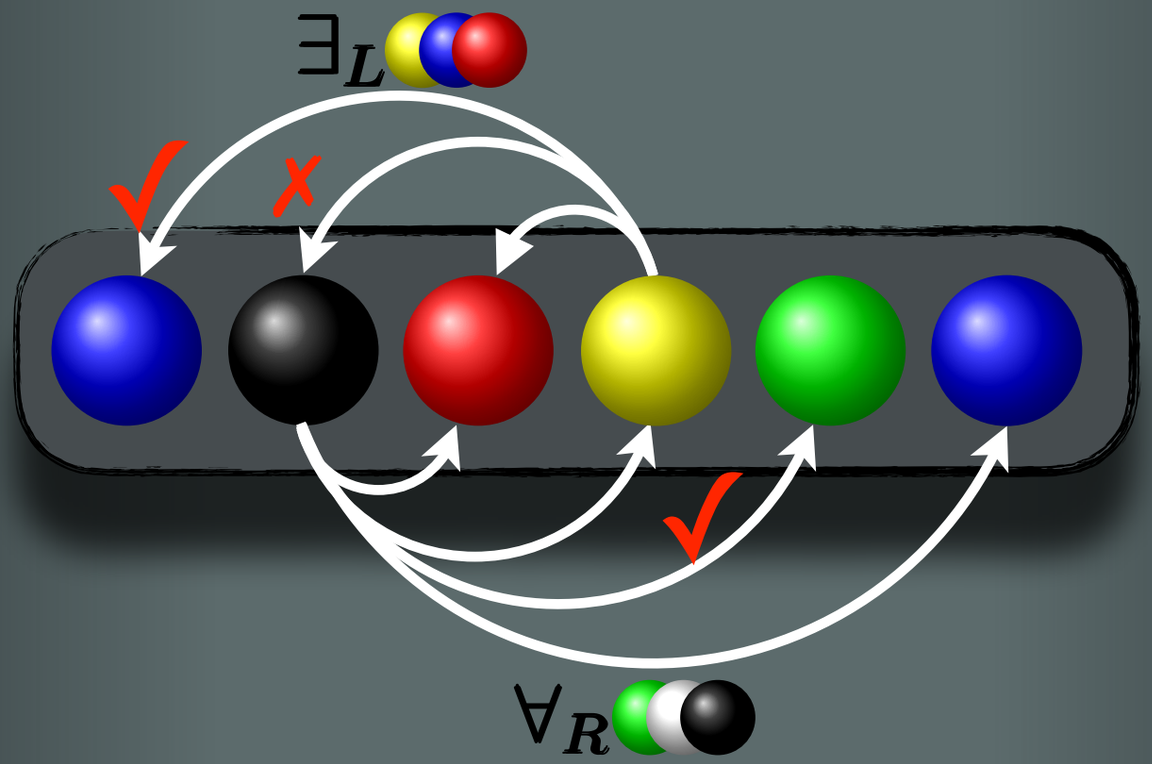




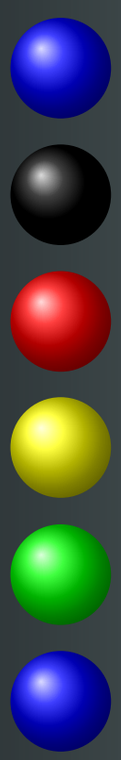


·	·	·	·	·	·
·	·	?	?	✓	?
·	·	·	·	·	·
✓	✗	?	·	·	·
·	·	·	·	·	·
·	·	·	·	·	·

c

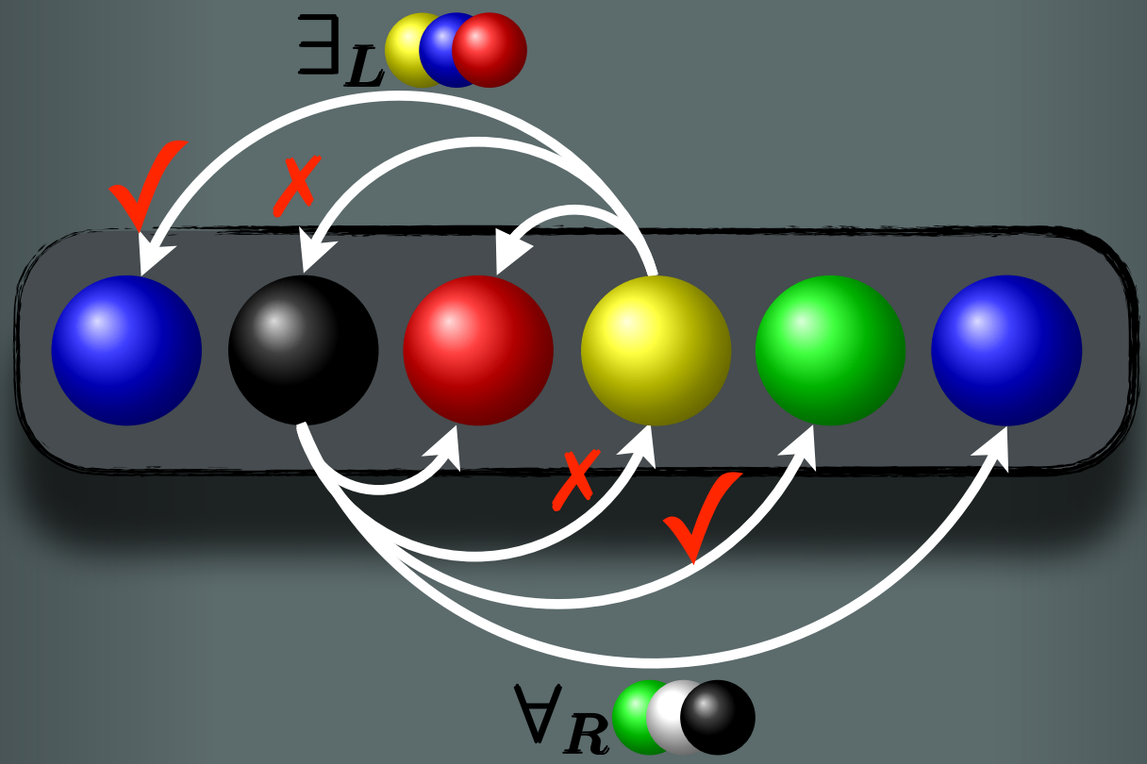




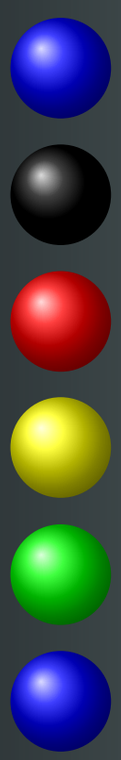


·	·	·	·	·	·
·	·	?	?	✓	?
·	·	·	·	·	·
✓	✗	?	·	·	·
·	·	·	·	·	·
·	·	·	·	·	·

c

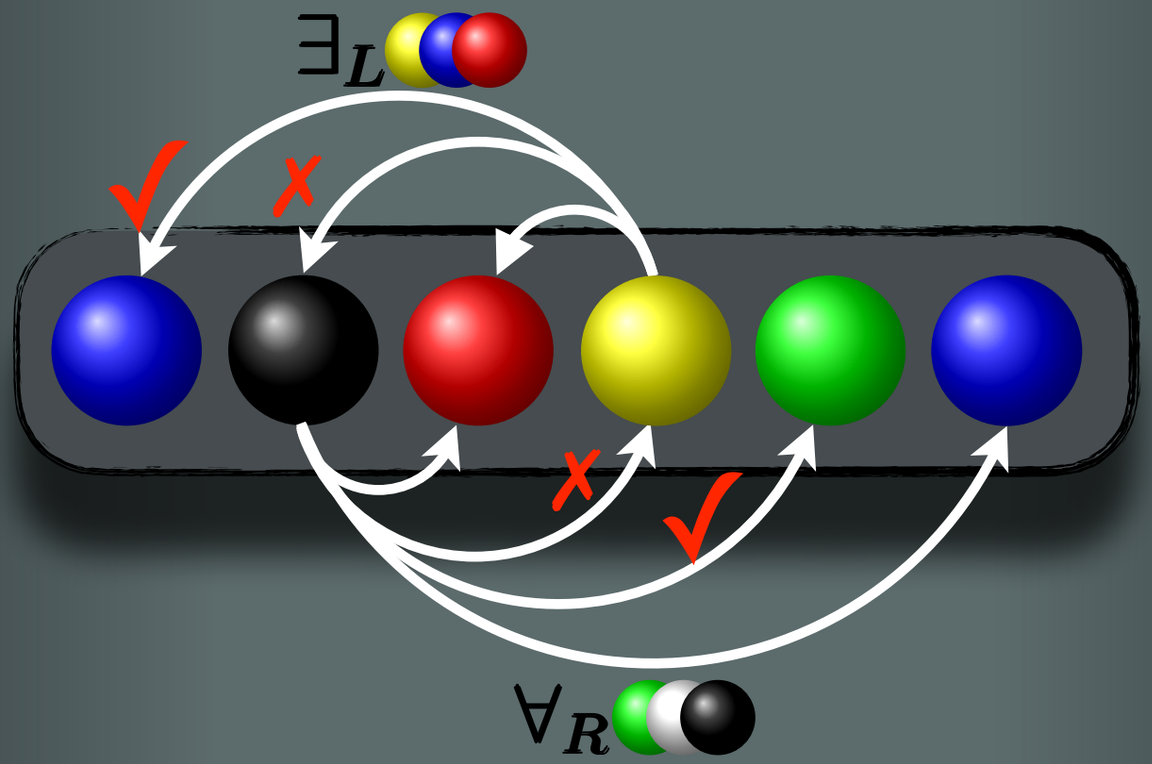




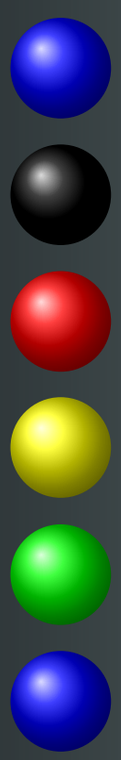


·	·	·	·	·	·
·	·	?	?	✓	?
·	·	·	·	·	·
✓	✗	?	·	·	·
·	·	·	·	·	·
·	·	·	·	·	·

c

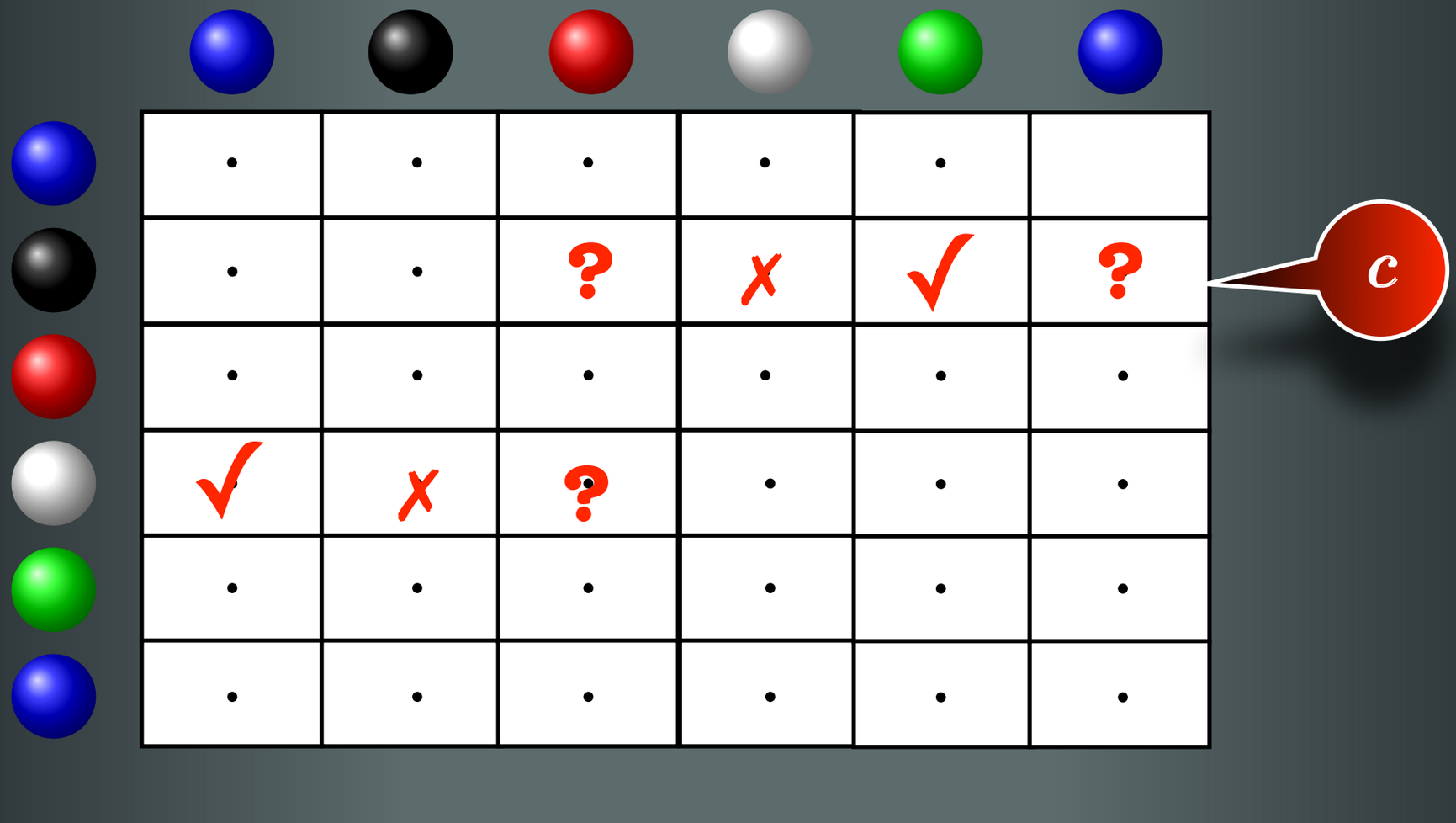


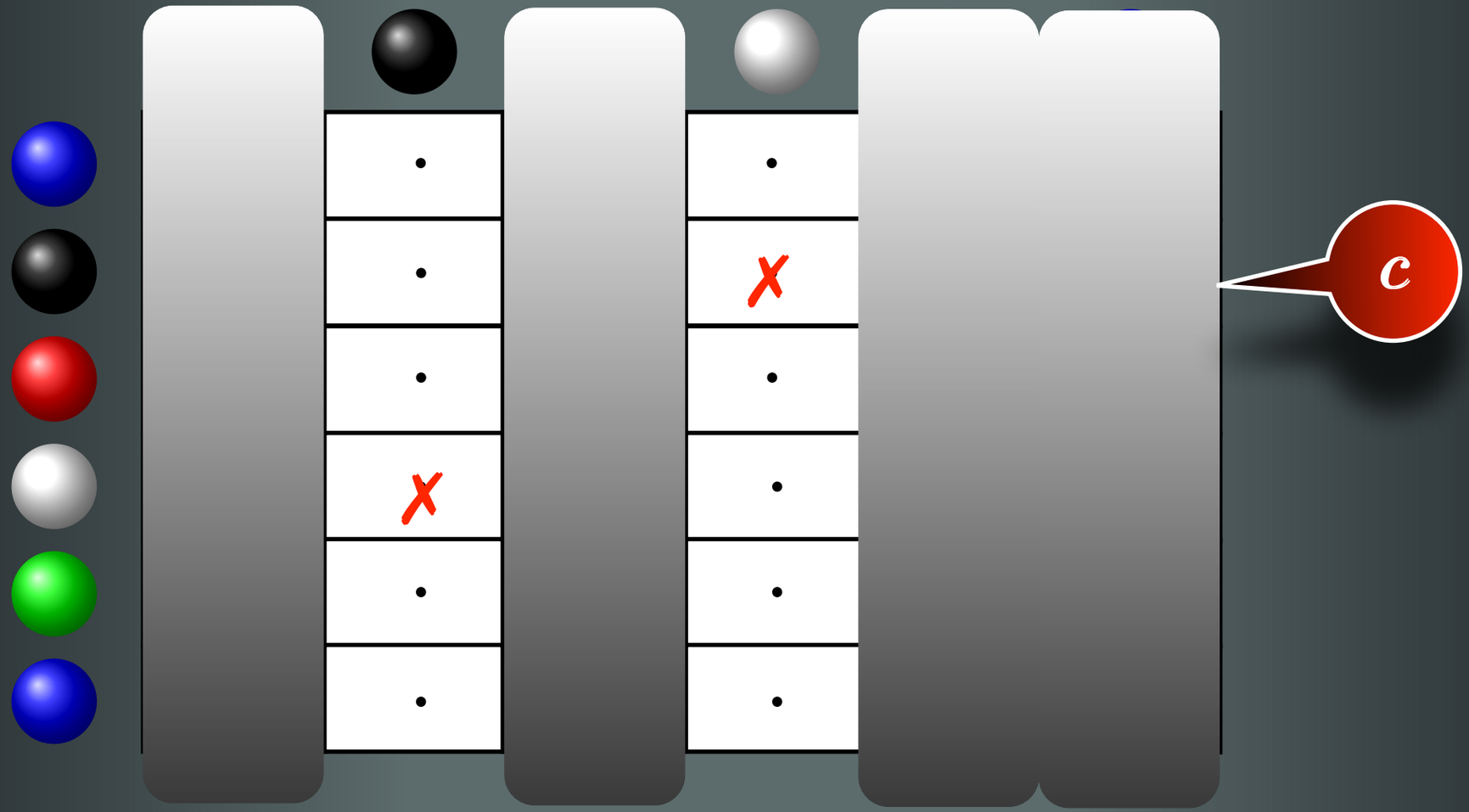


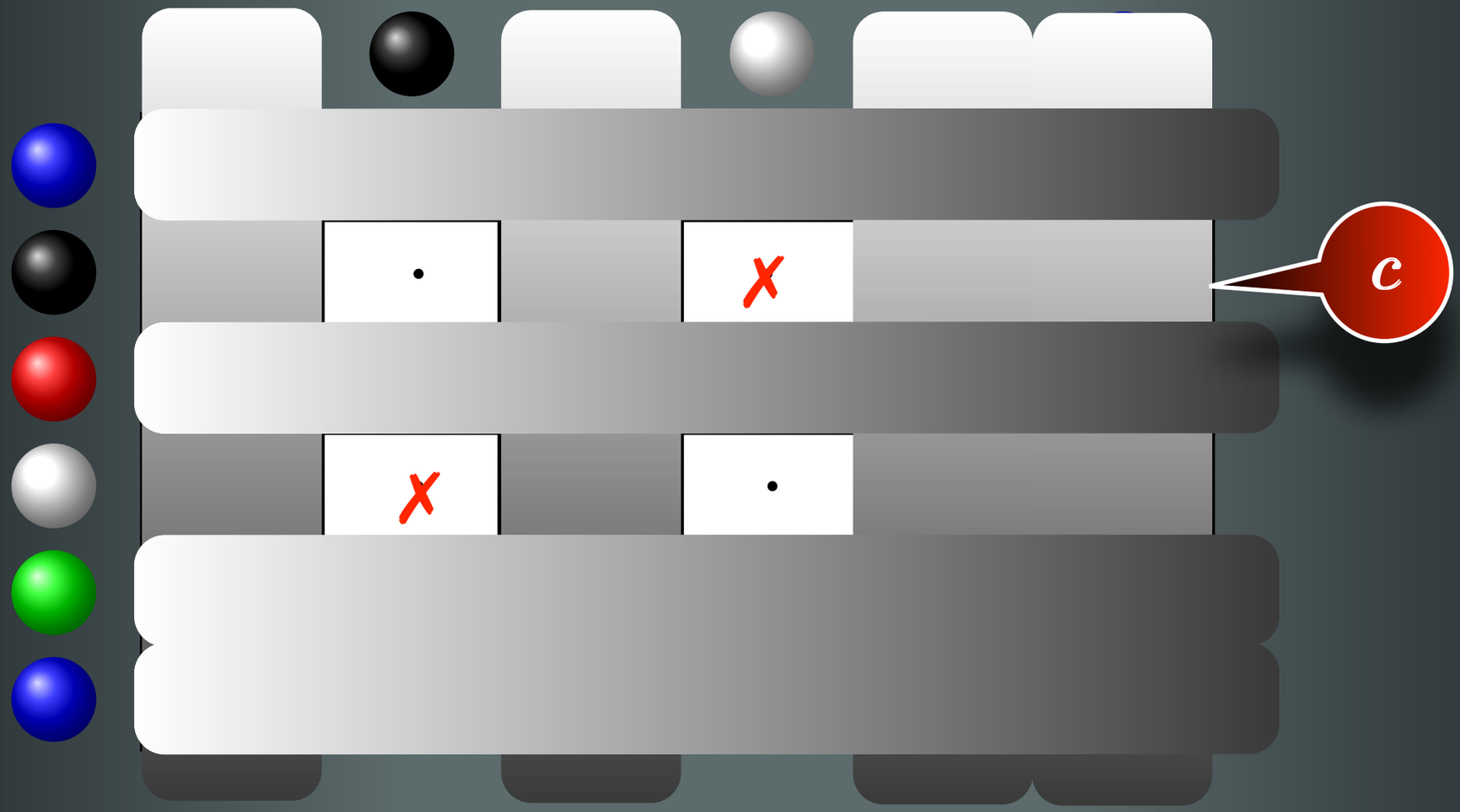


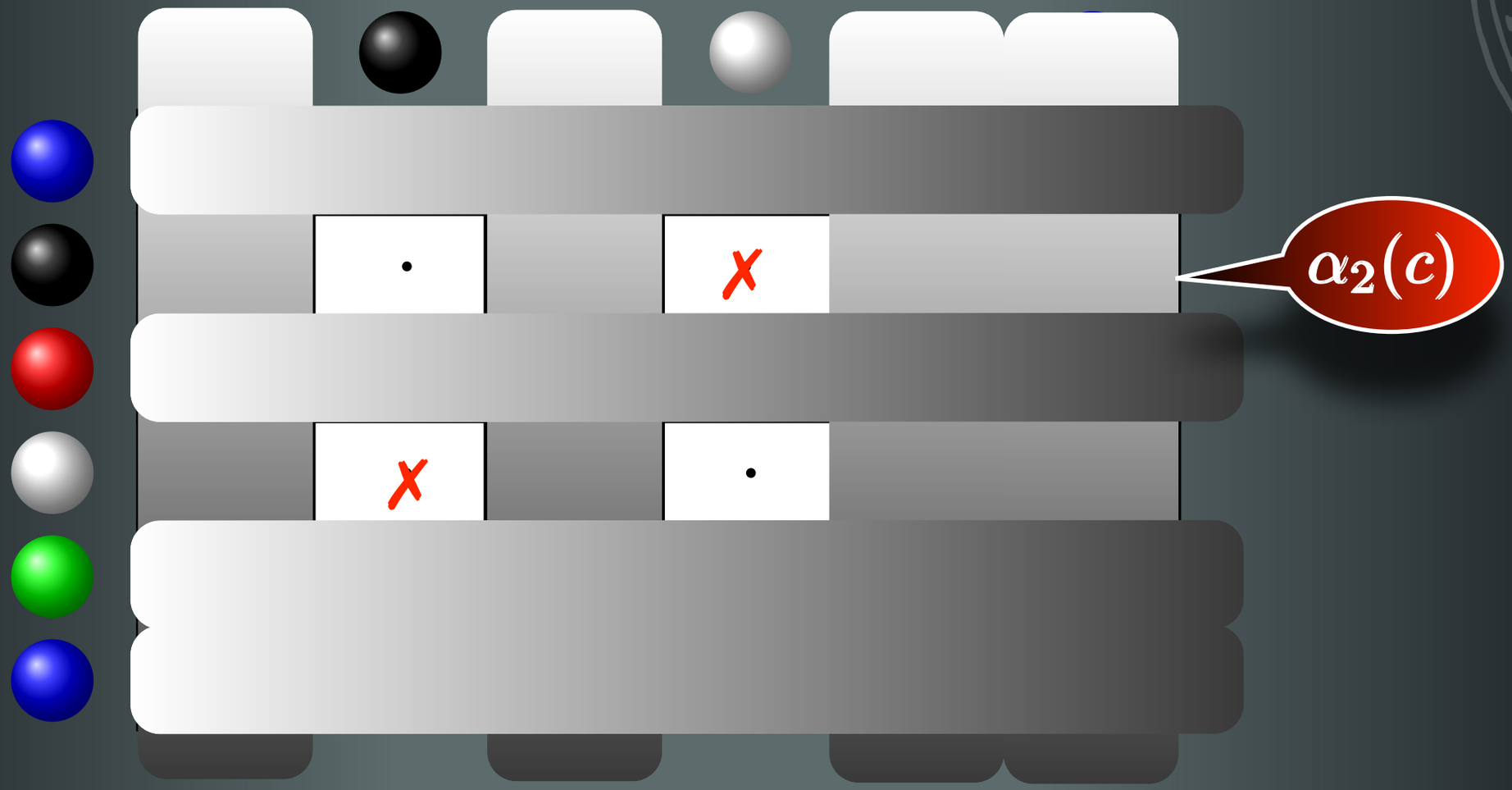
·	·	·	·	·	·
·	·	?	✗	✓	?
·	·	·	·	·	·
✓	✗	?	·	·	·
·	·	·	·	·	·
·	·	·	·	·	·

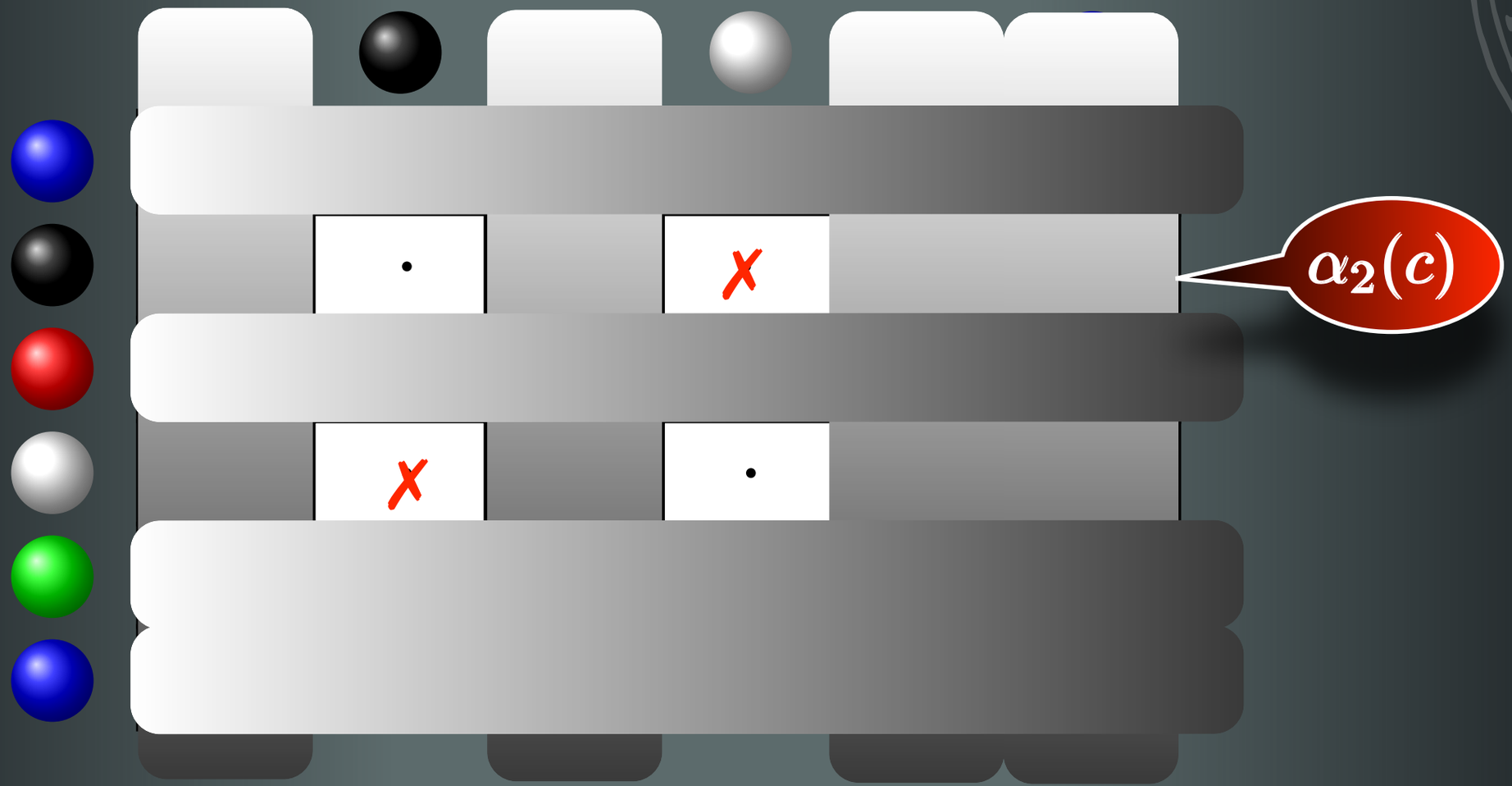
c

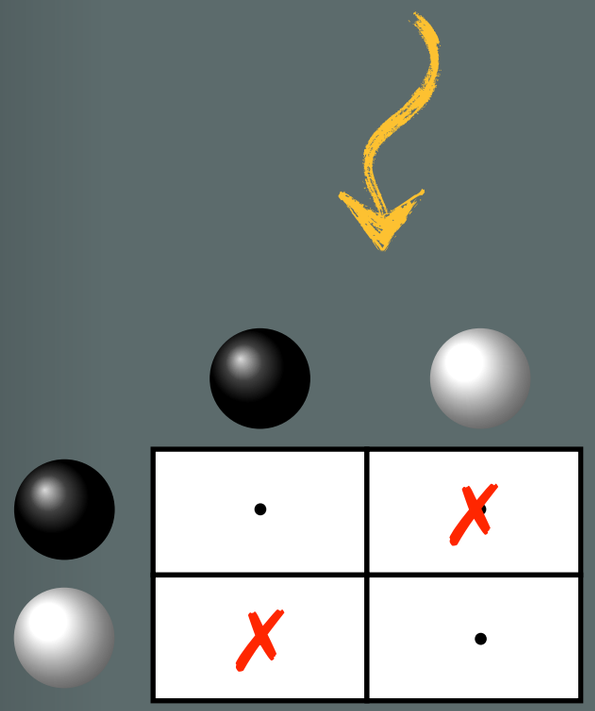

?
✗
✓
?
c

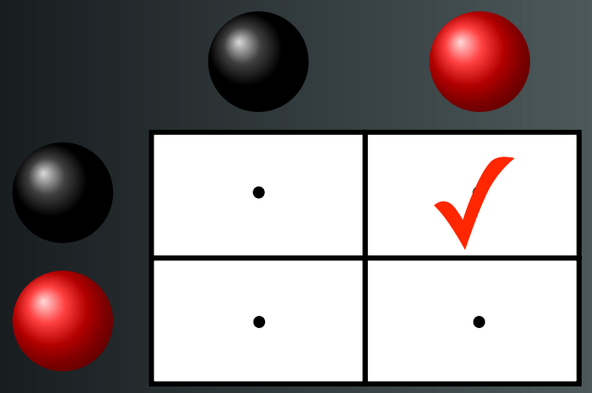

c

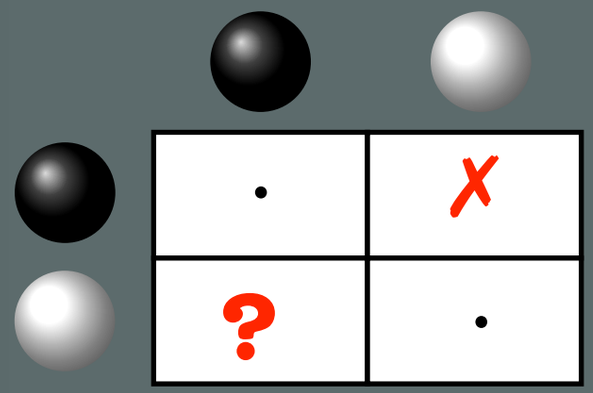

c

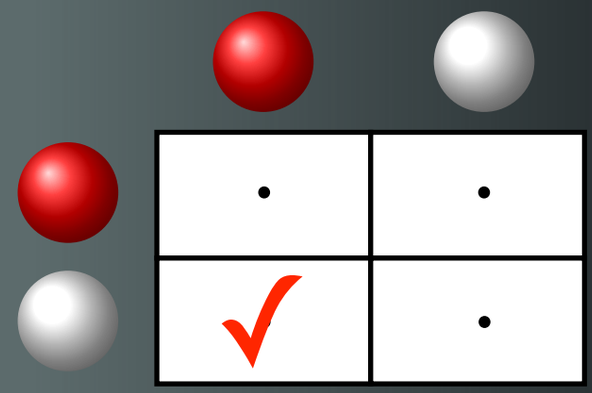

$\alpha_2(c)$

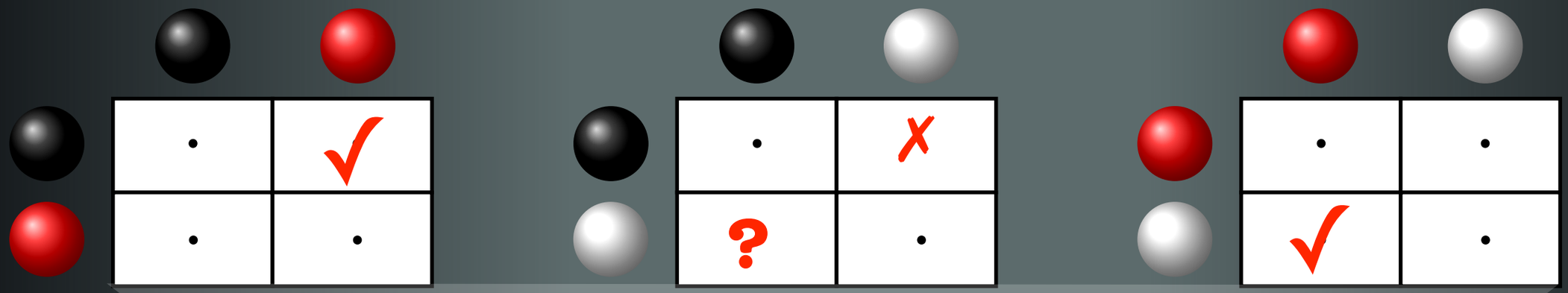

$\alpha_2(c)$



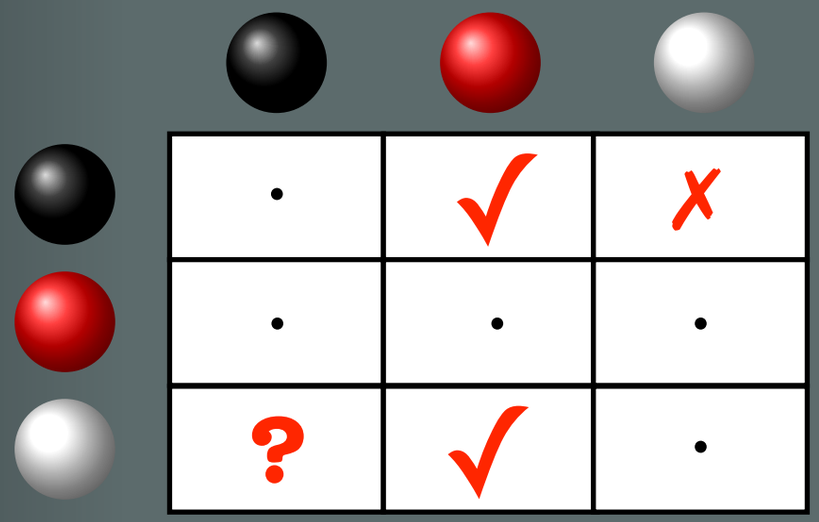








$\gamma_2(X)$



non-atomic global conditions

shared variables

broadcast

rendez-vous

process creation/deletion

Topologies

- linear
- ring
- tree
- multiset

non-atomic global conditions ✓

shared variables

broadcast

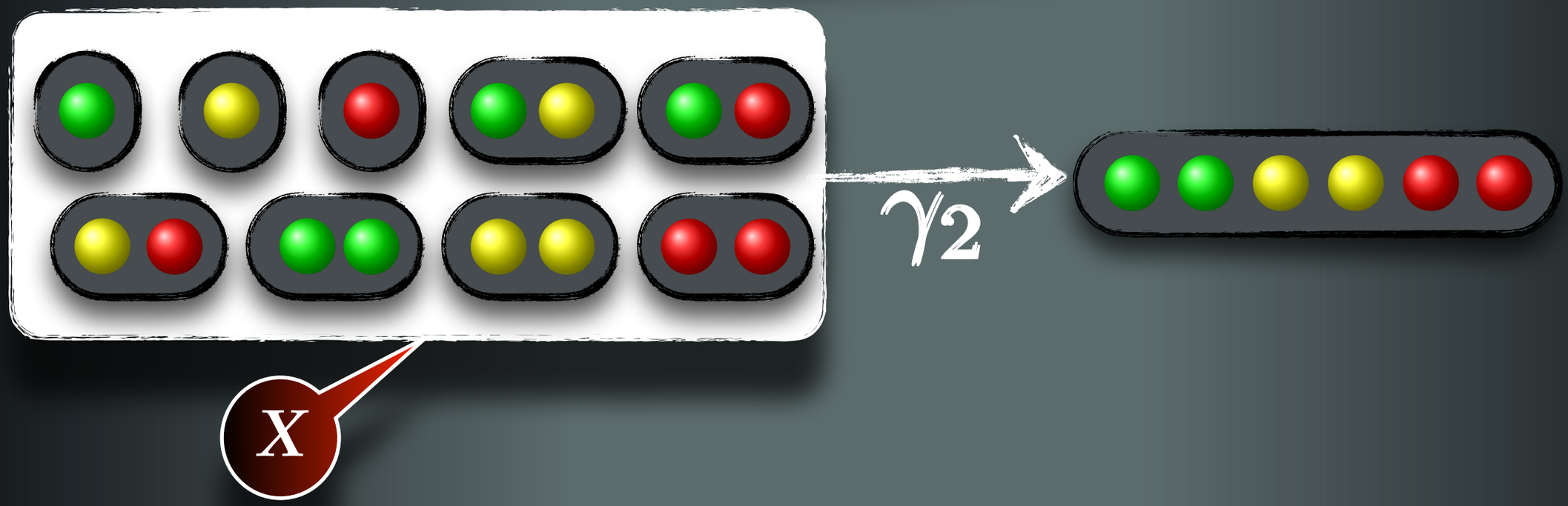
rendez-vous

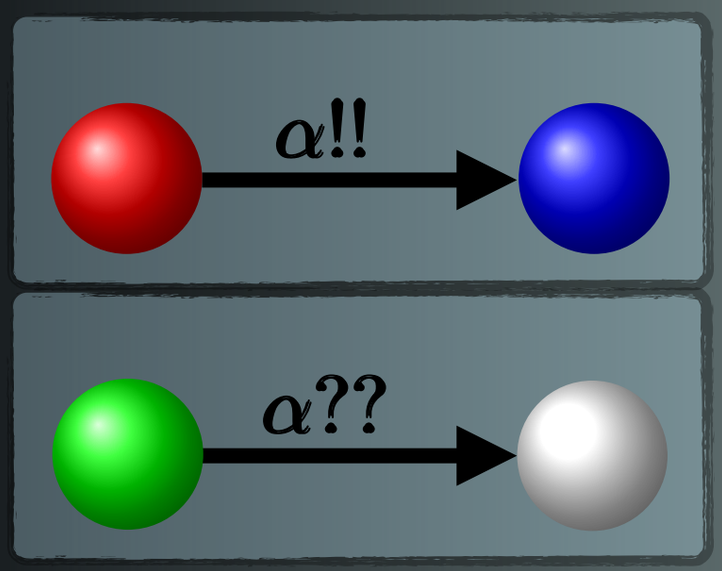
process creation/deletion

Topologies

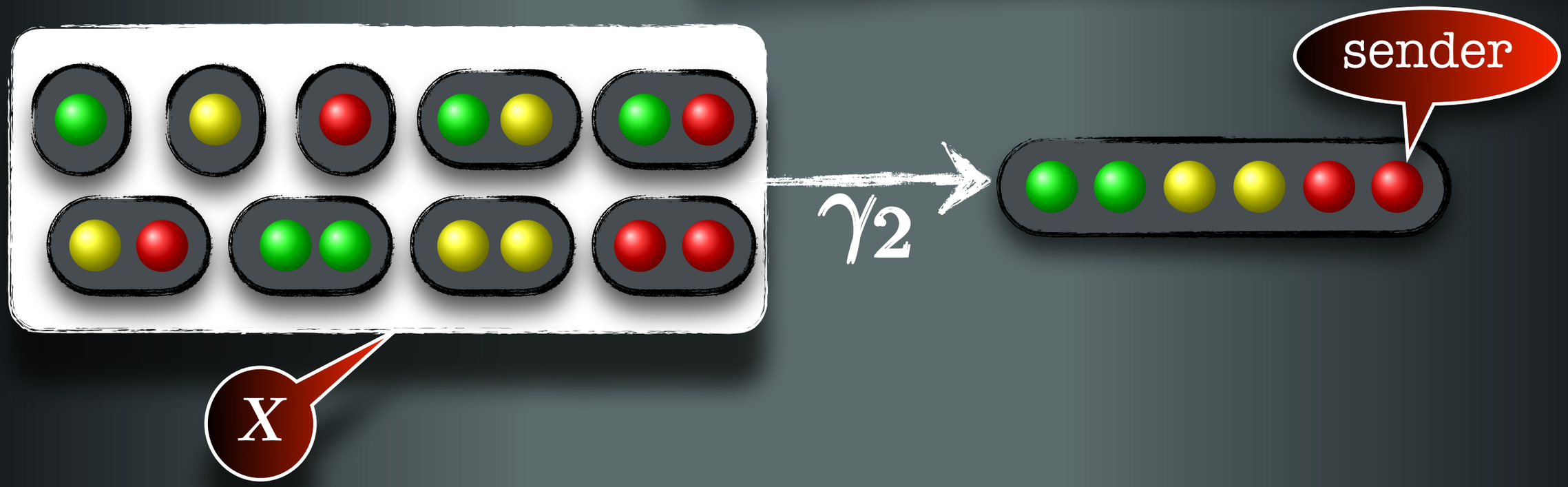
- linear
- ring
- tree
- multiset

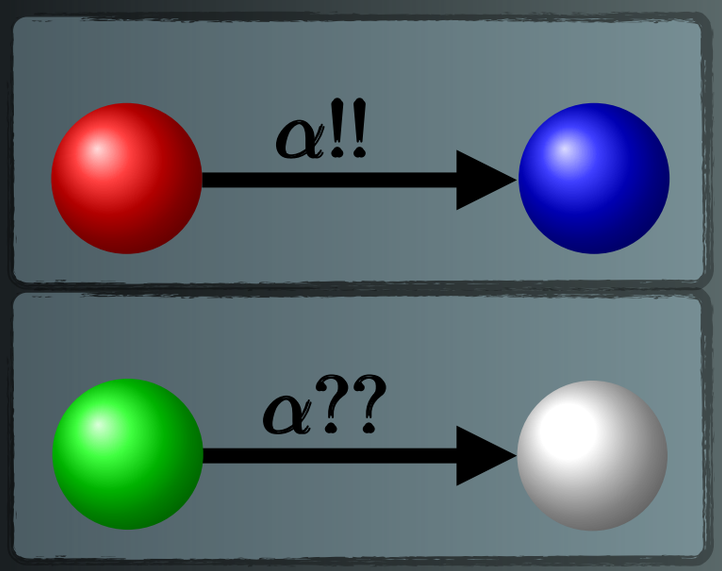
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



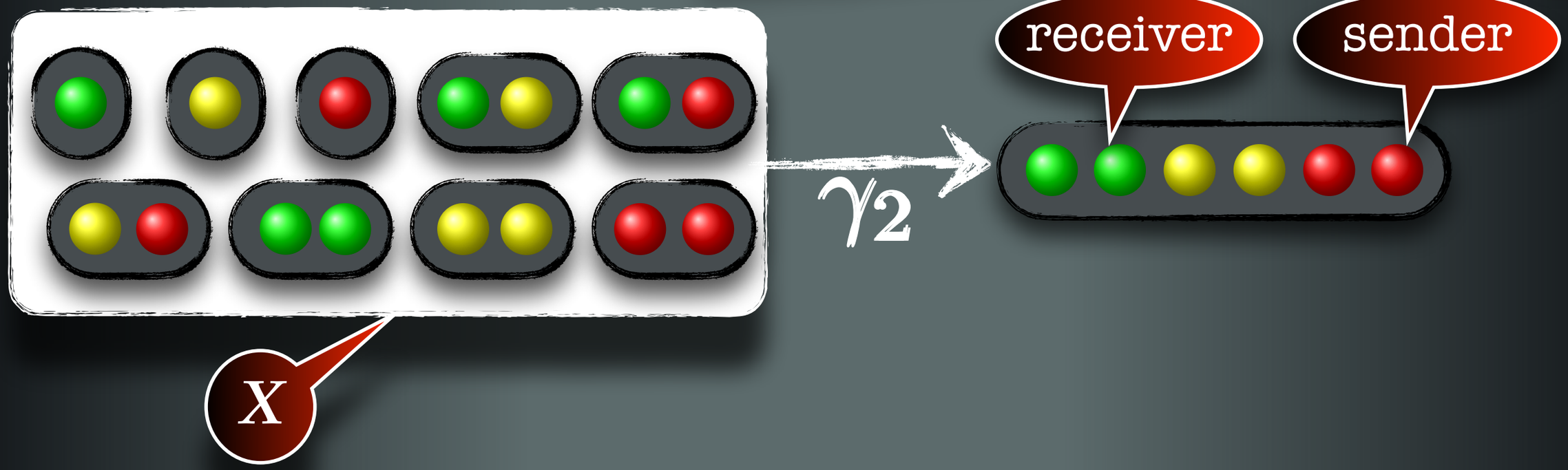


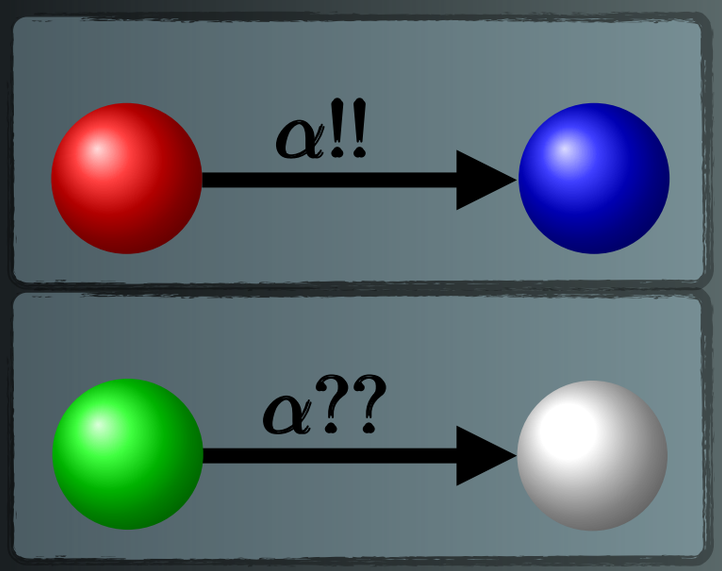
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



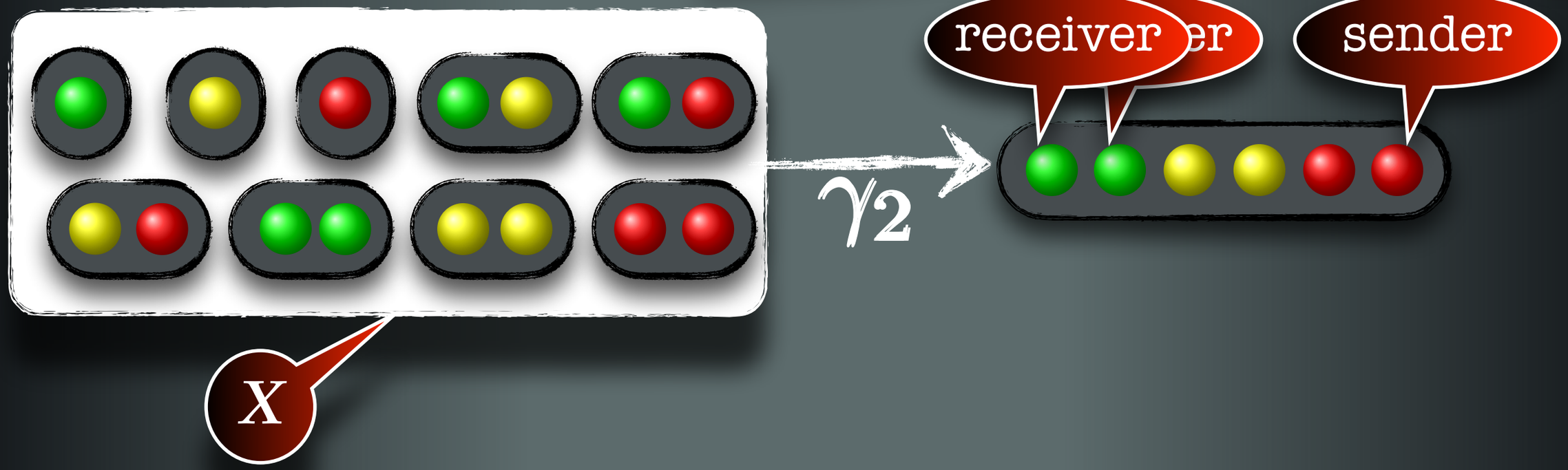


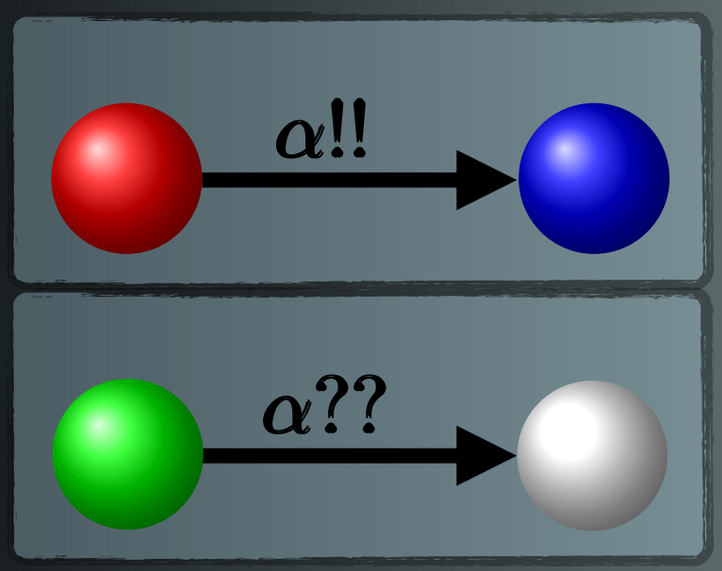
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



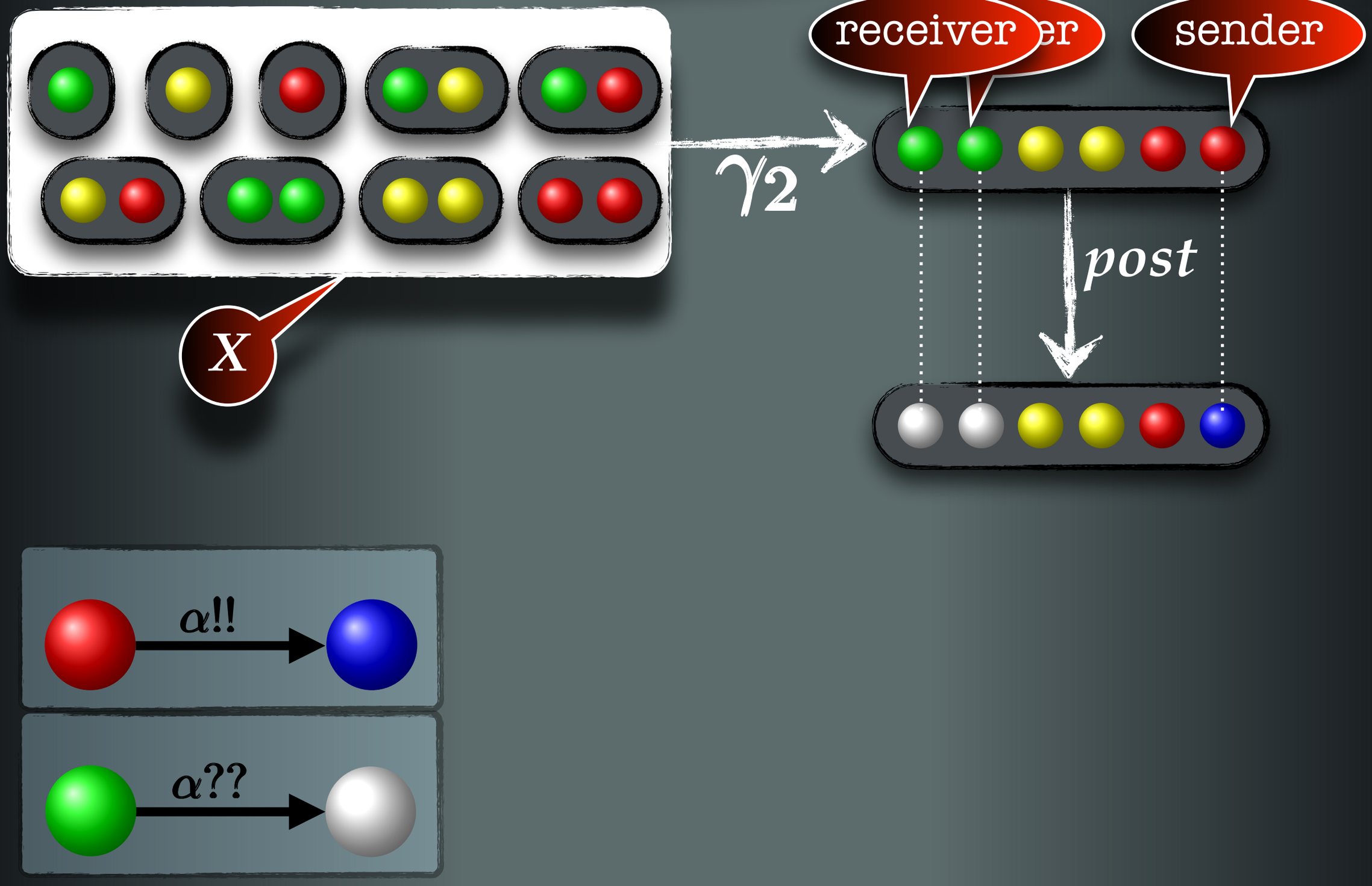


$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$

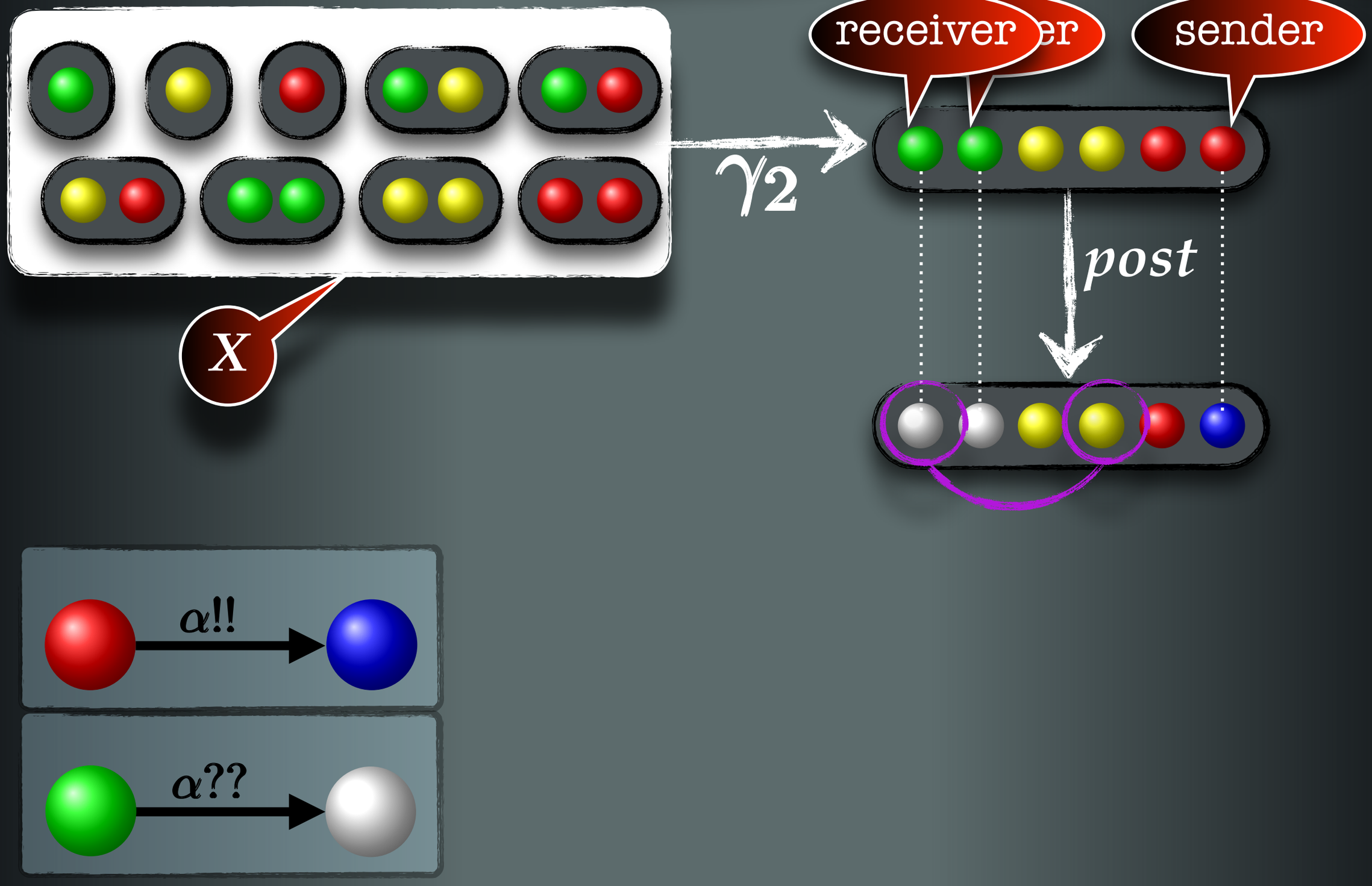




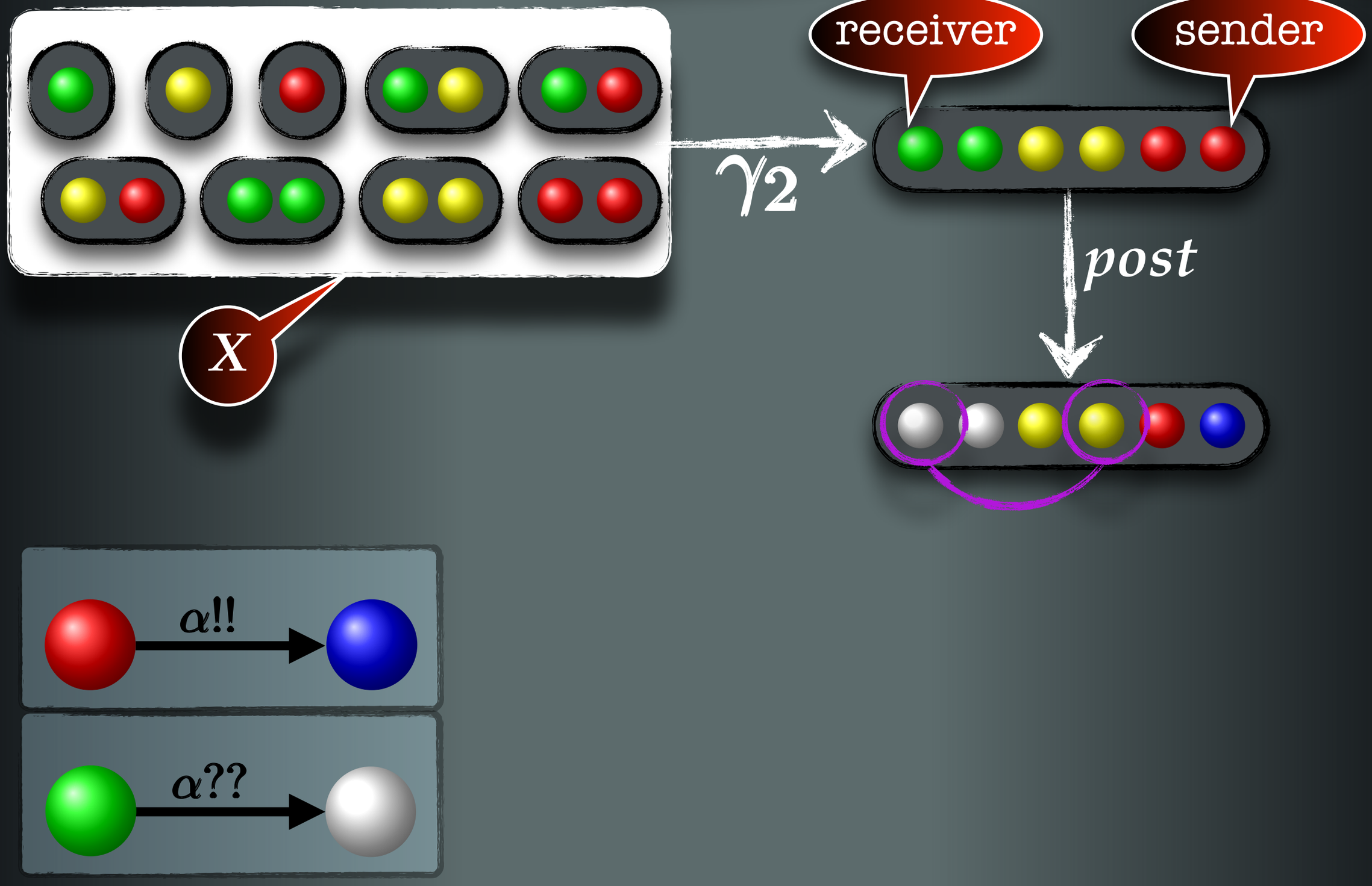
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



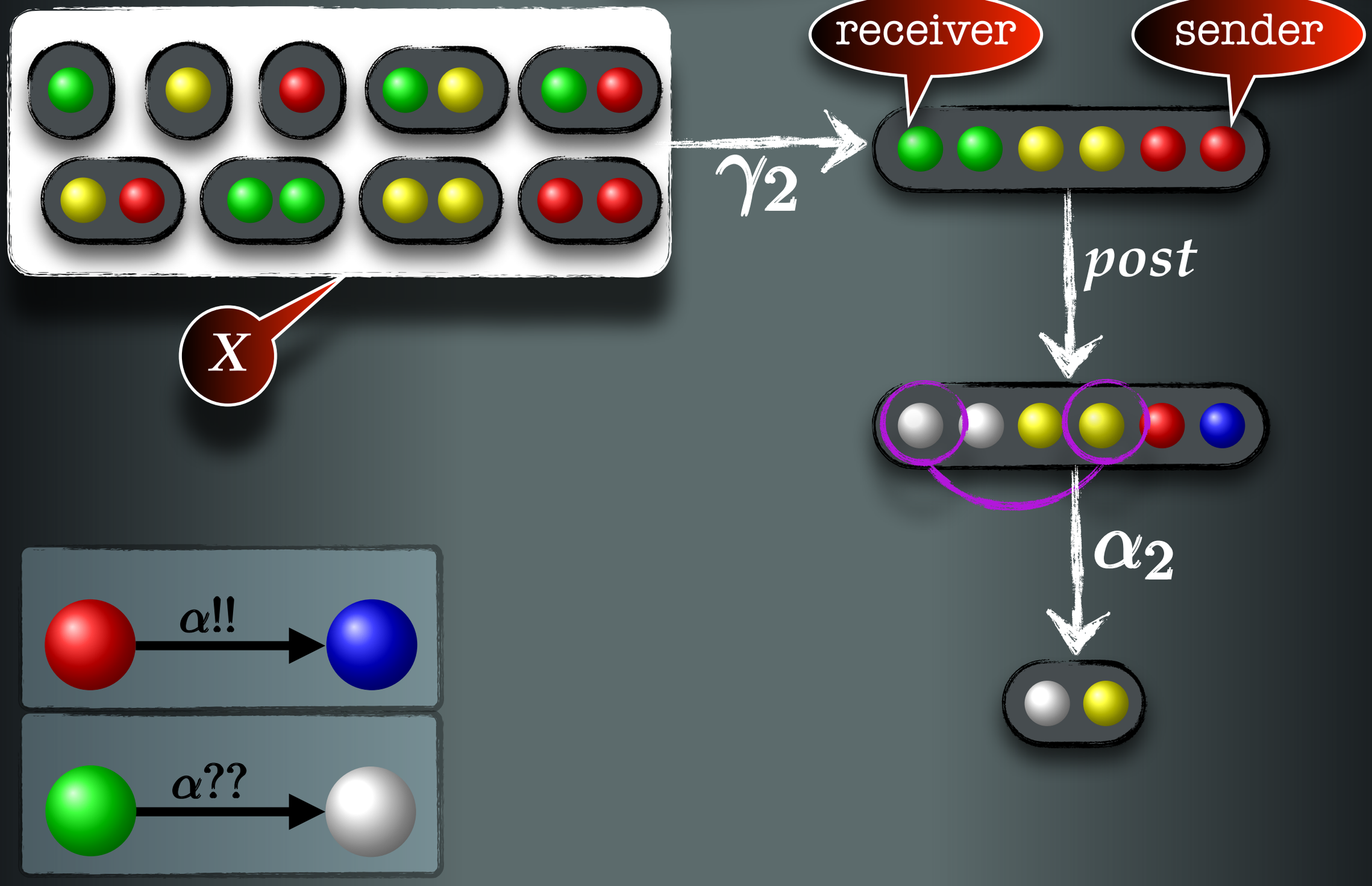
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



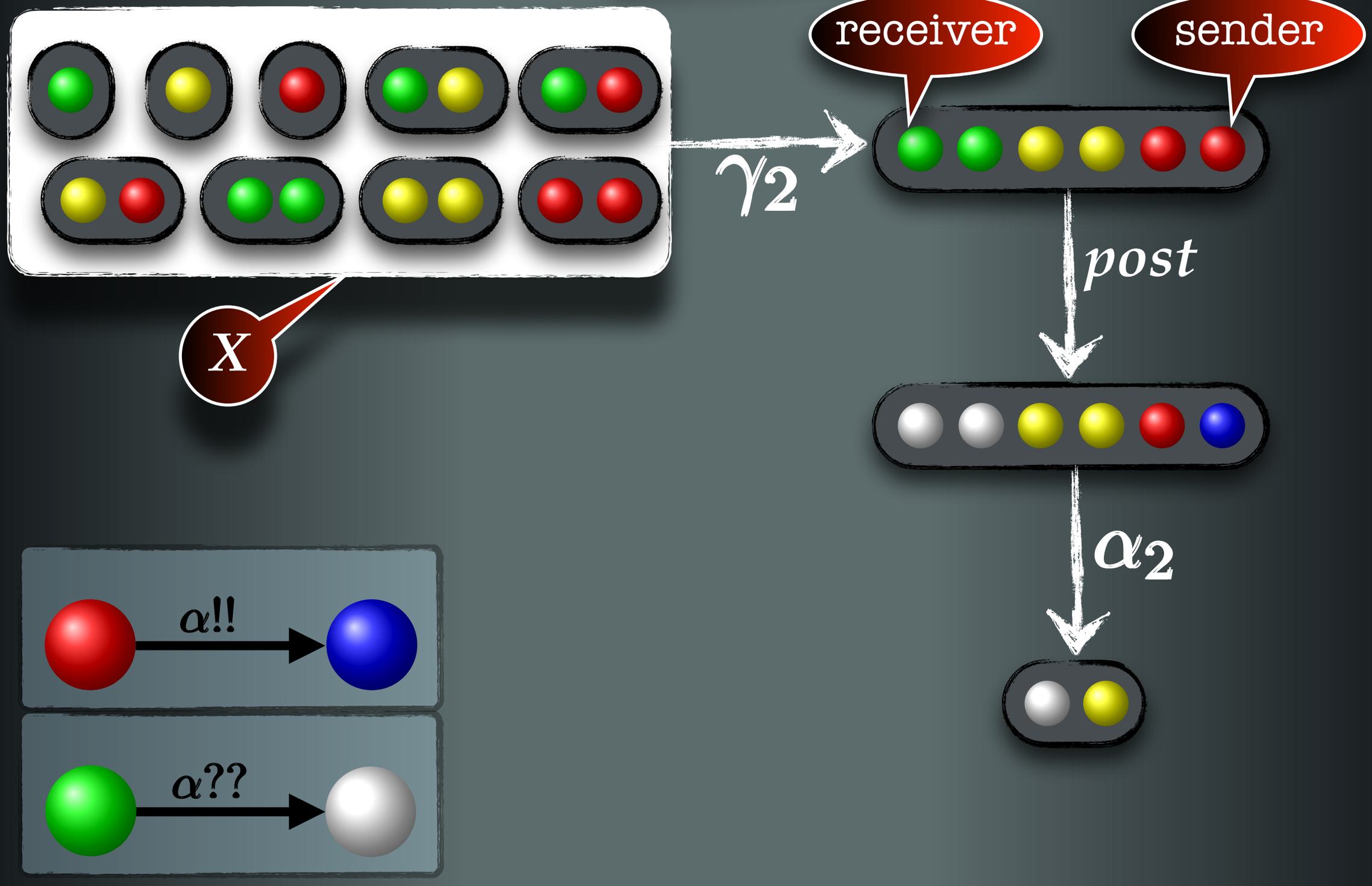
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



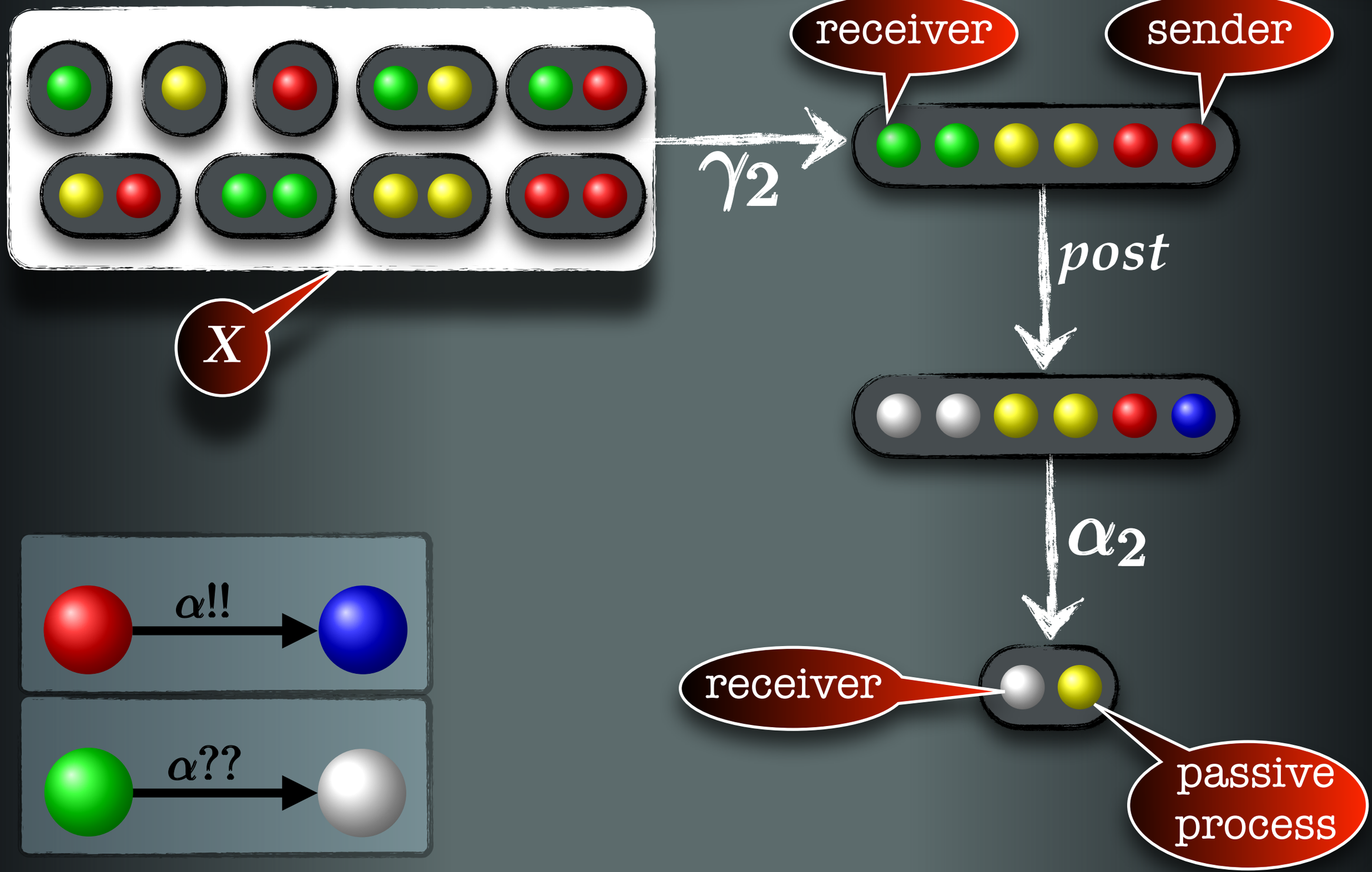
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



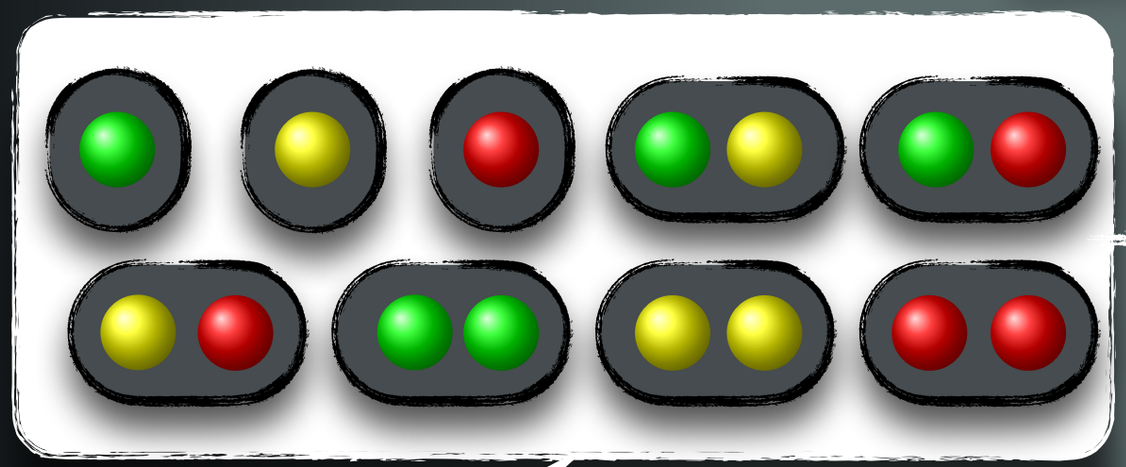
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



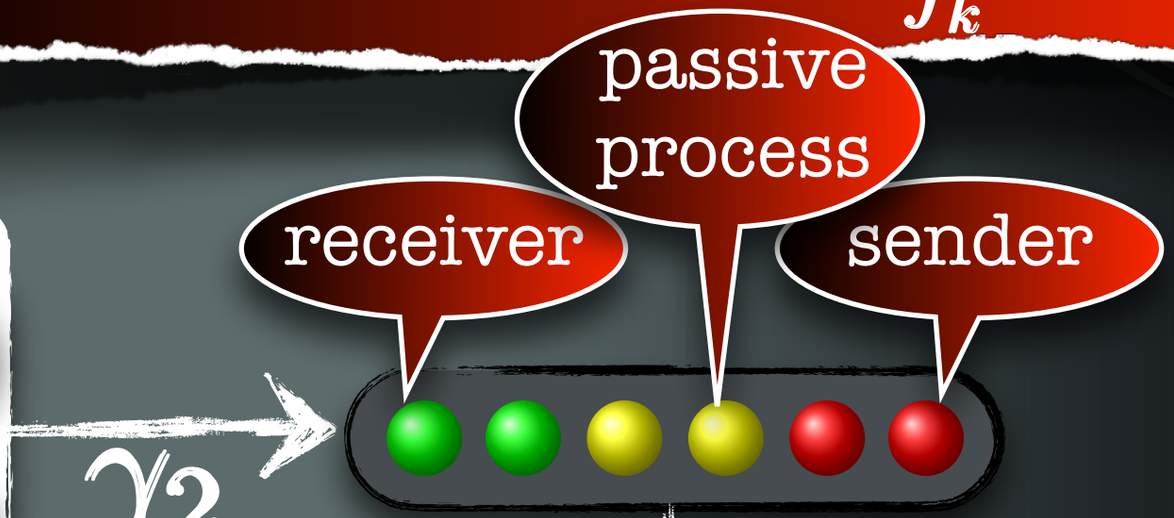
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



X



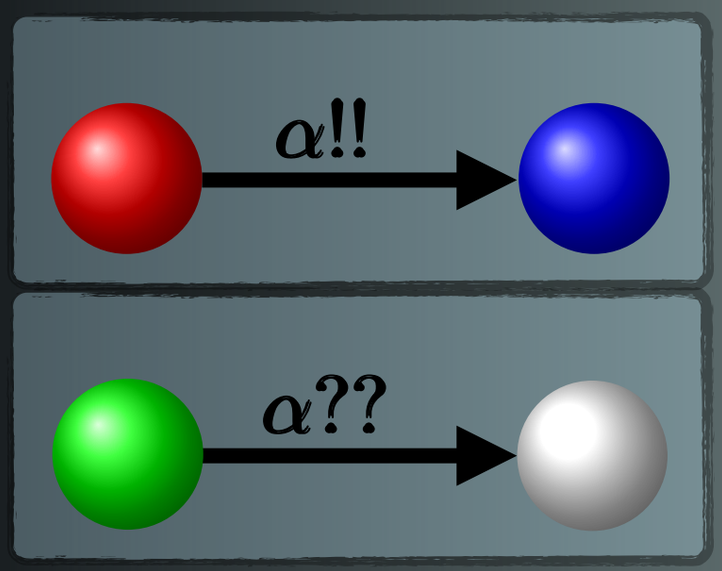
γ_2

post

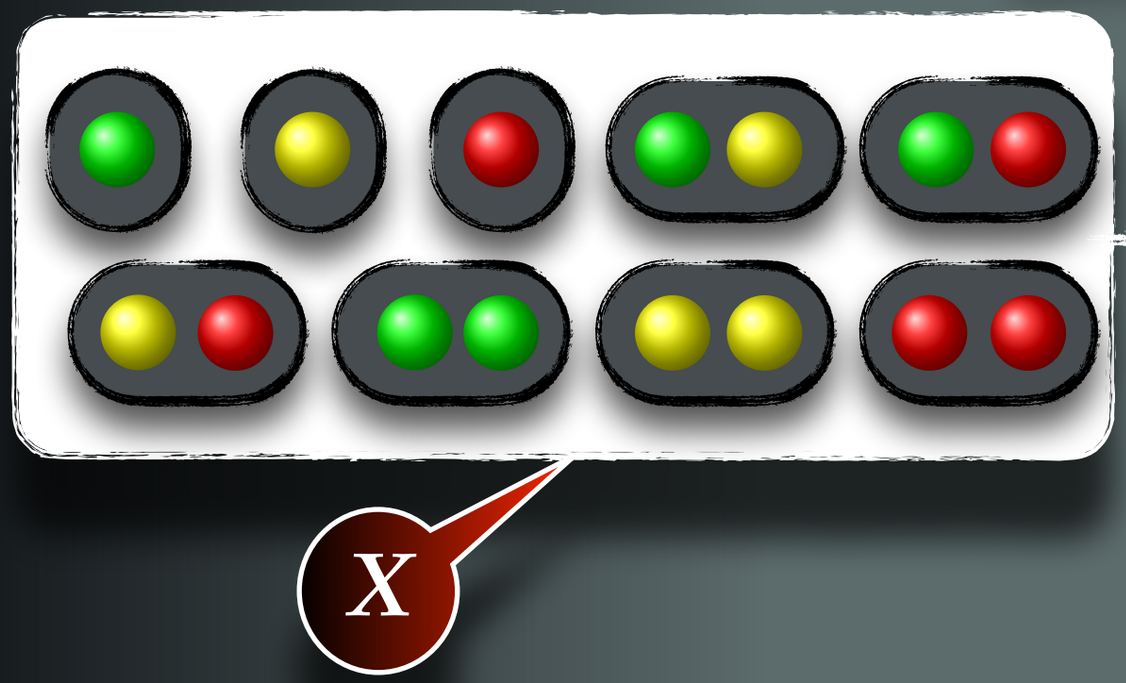


α_2

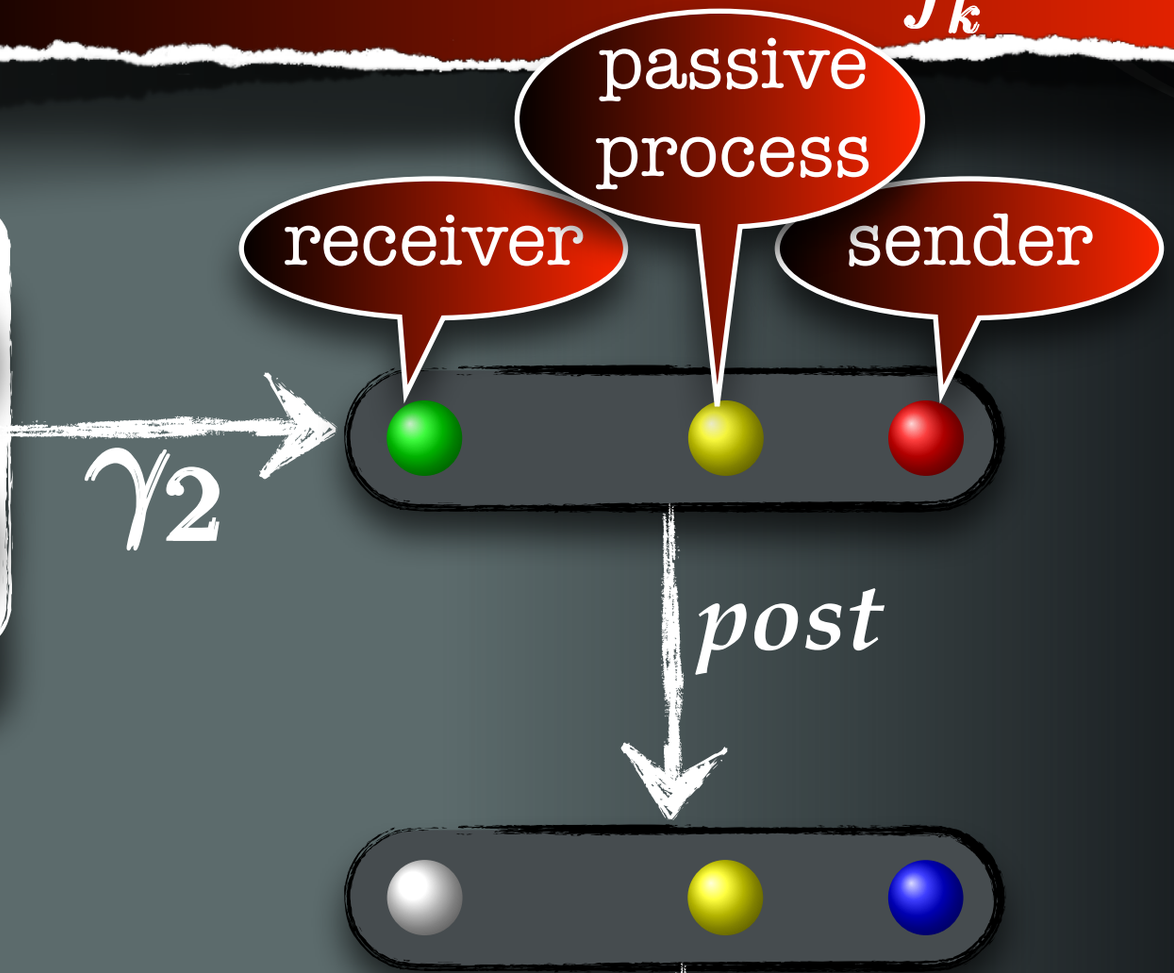




$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$

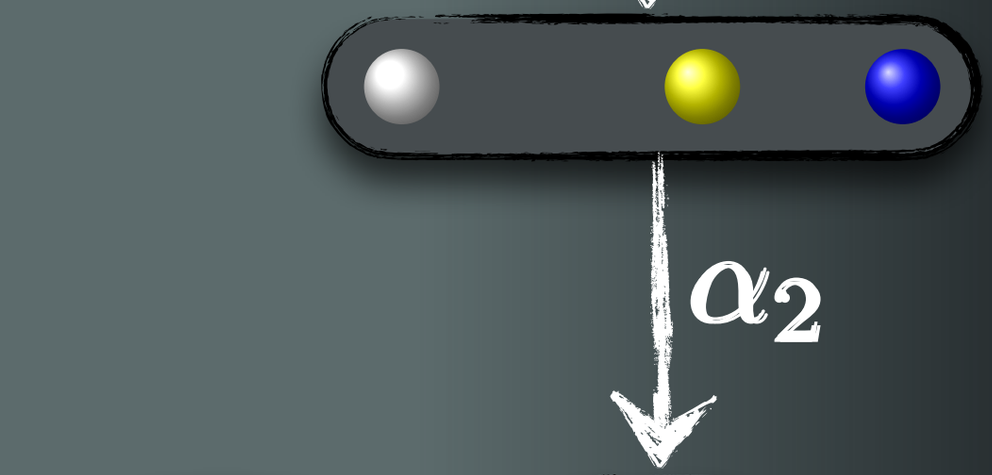


X

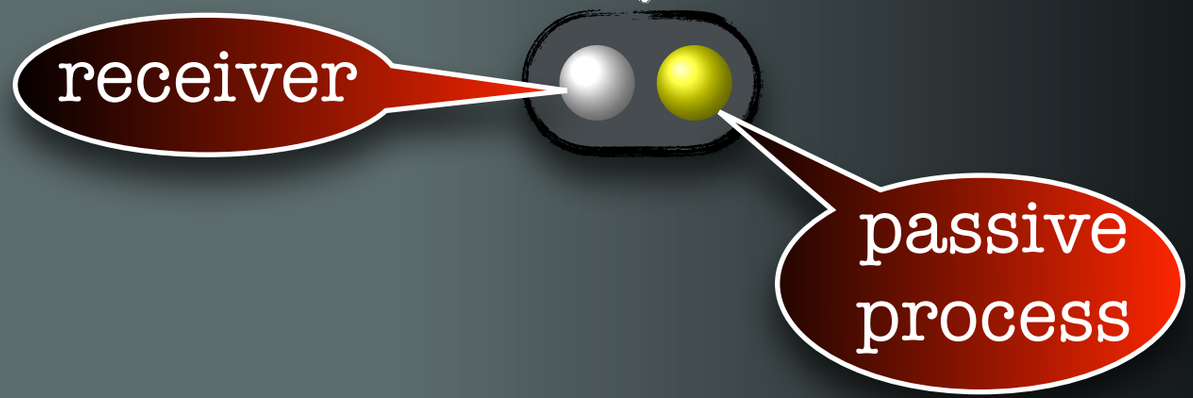


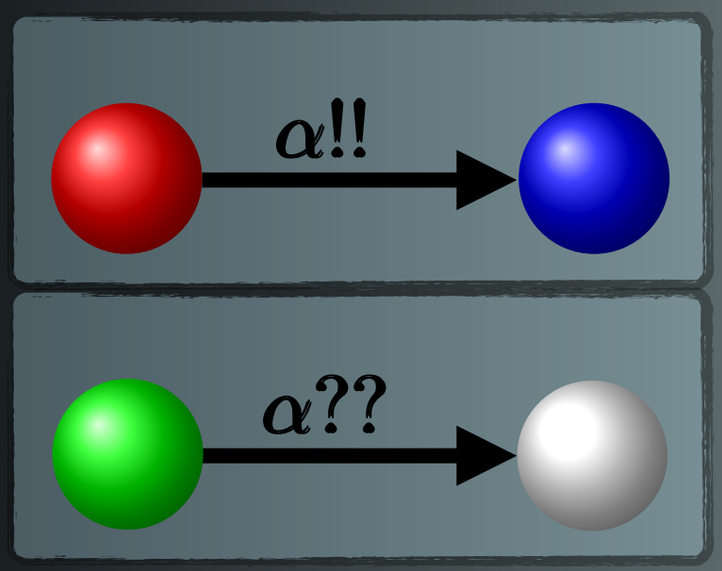
γ_2

post

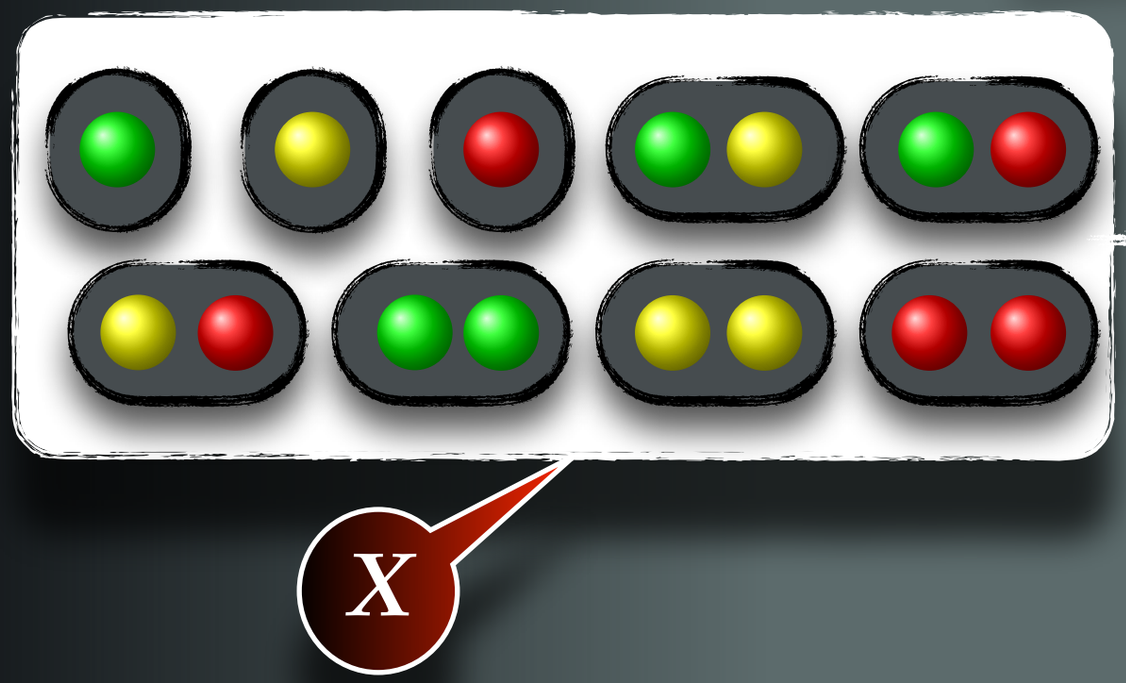


α_2

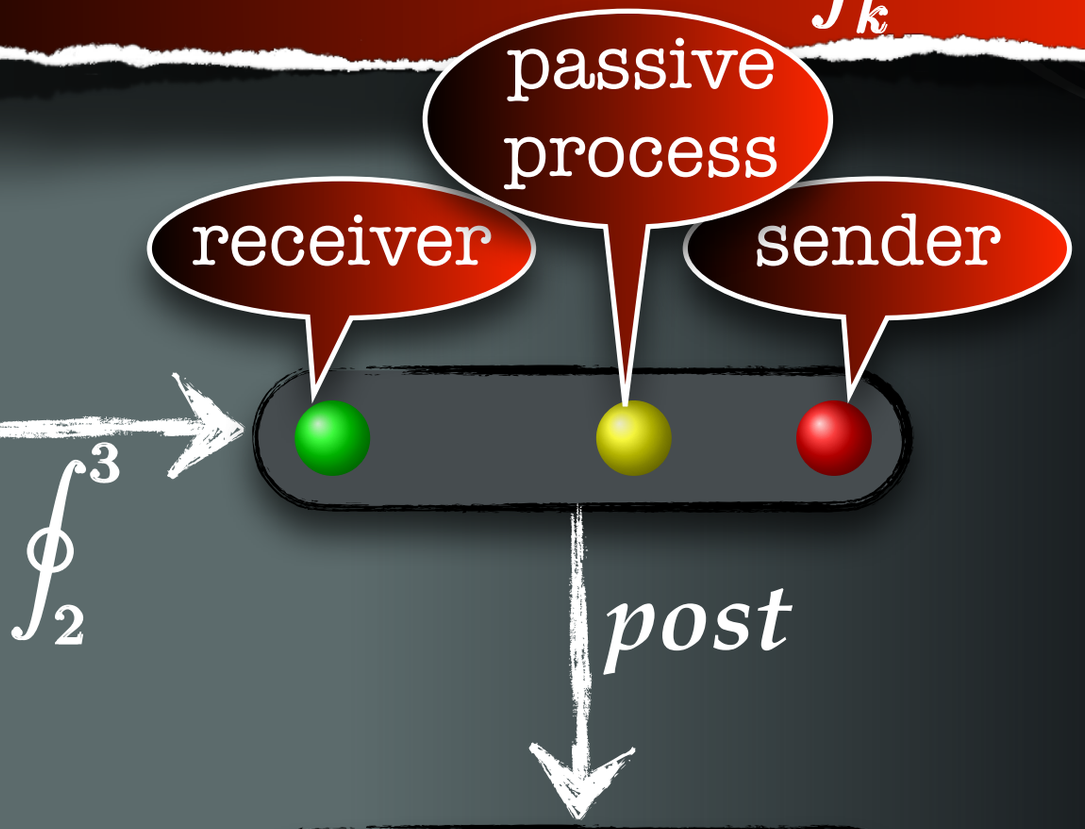




$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$

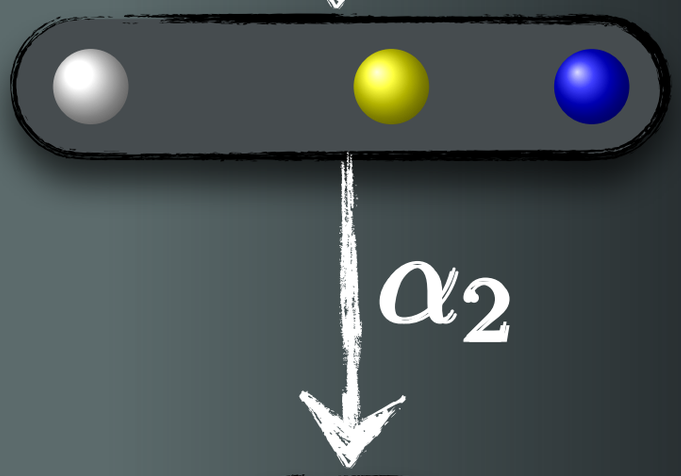


X

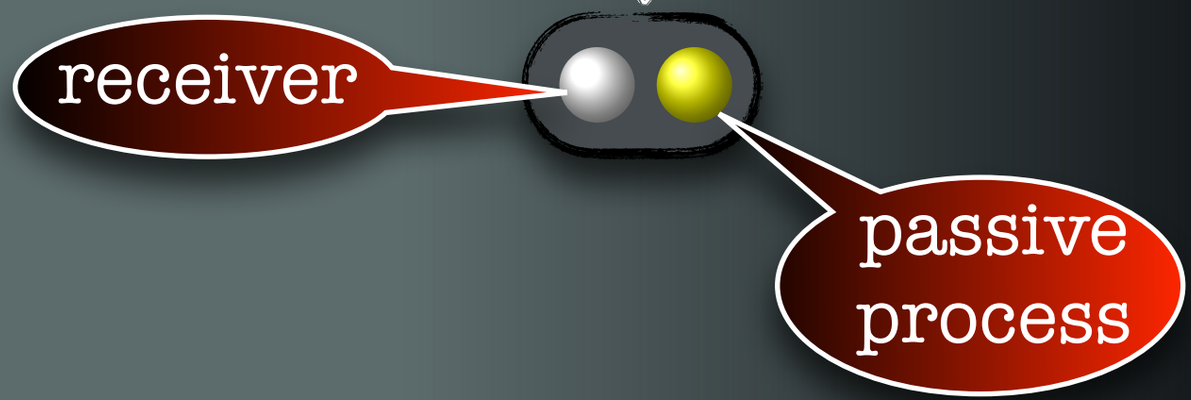


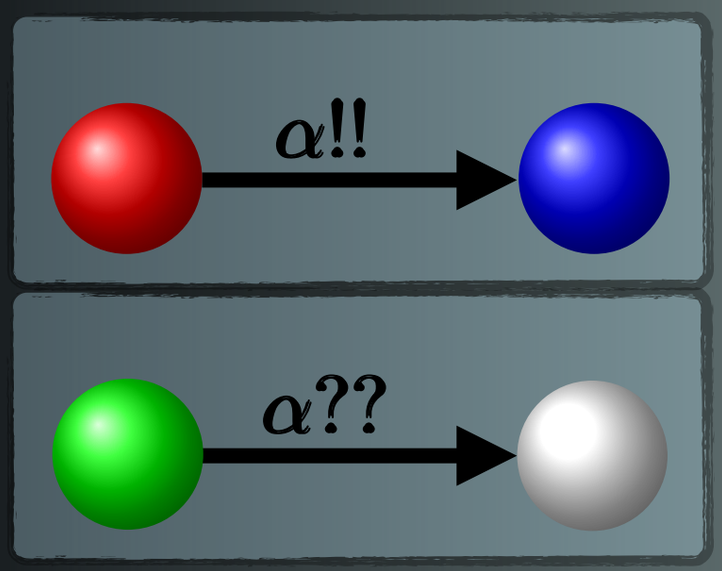
$\oint_2^3$

$post$

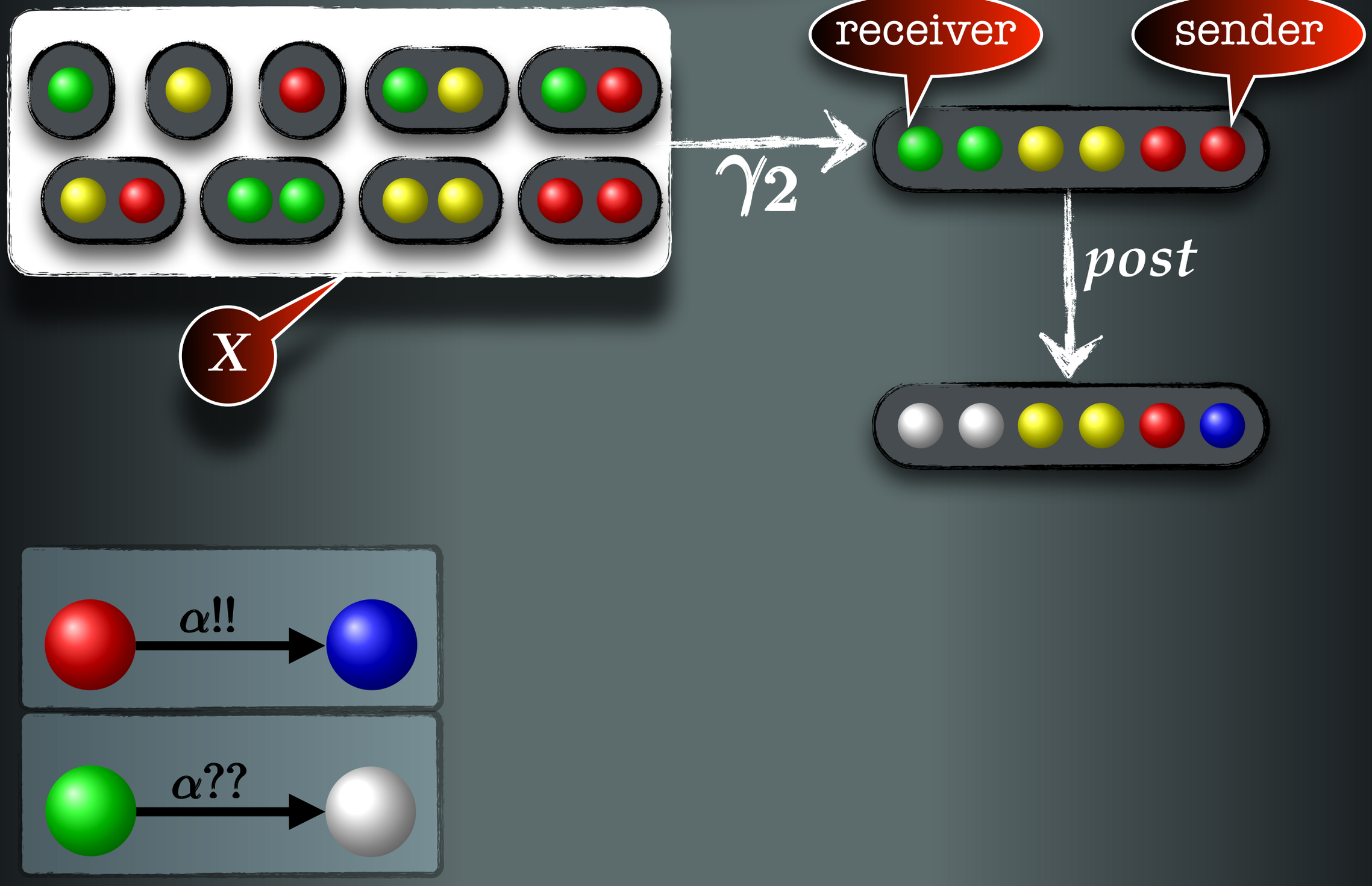


α_2

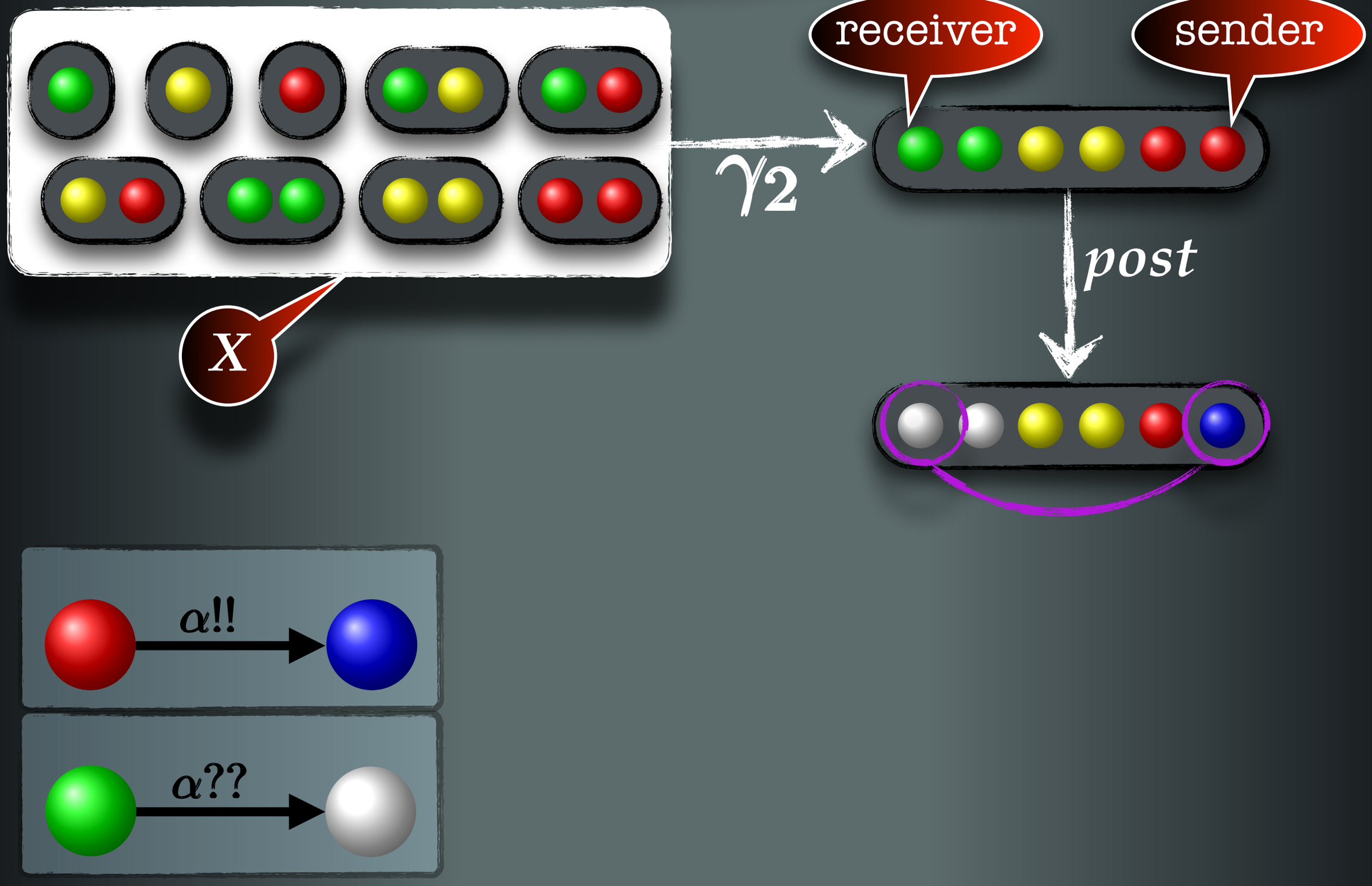




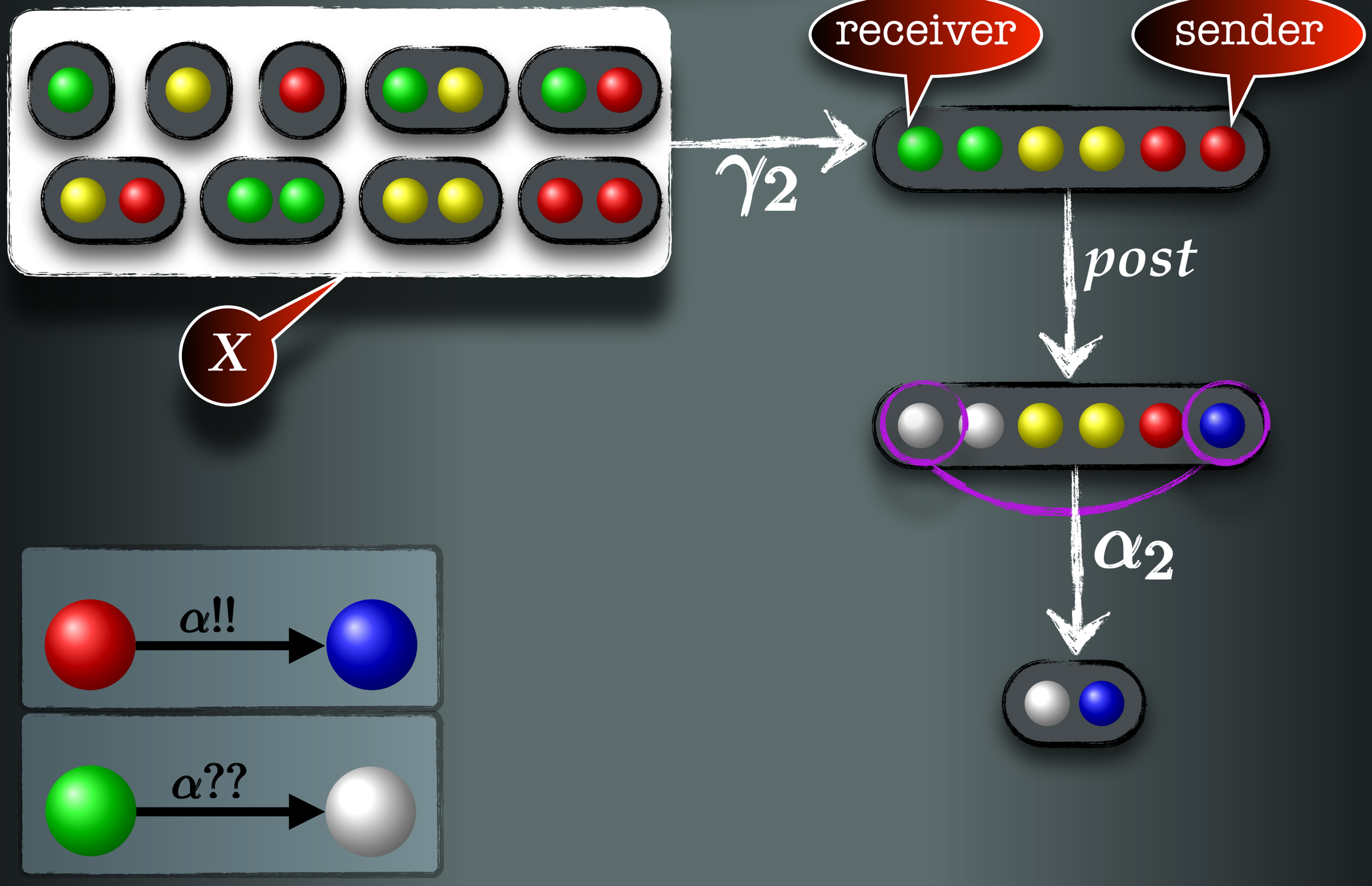
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



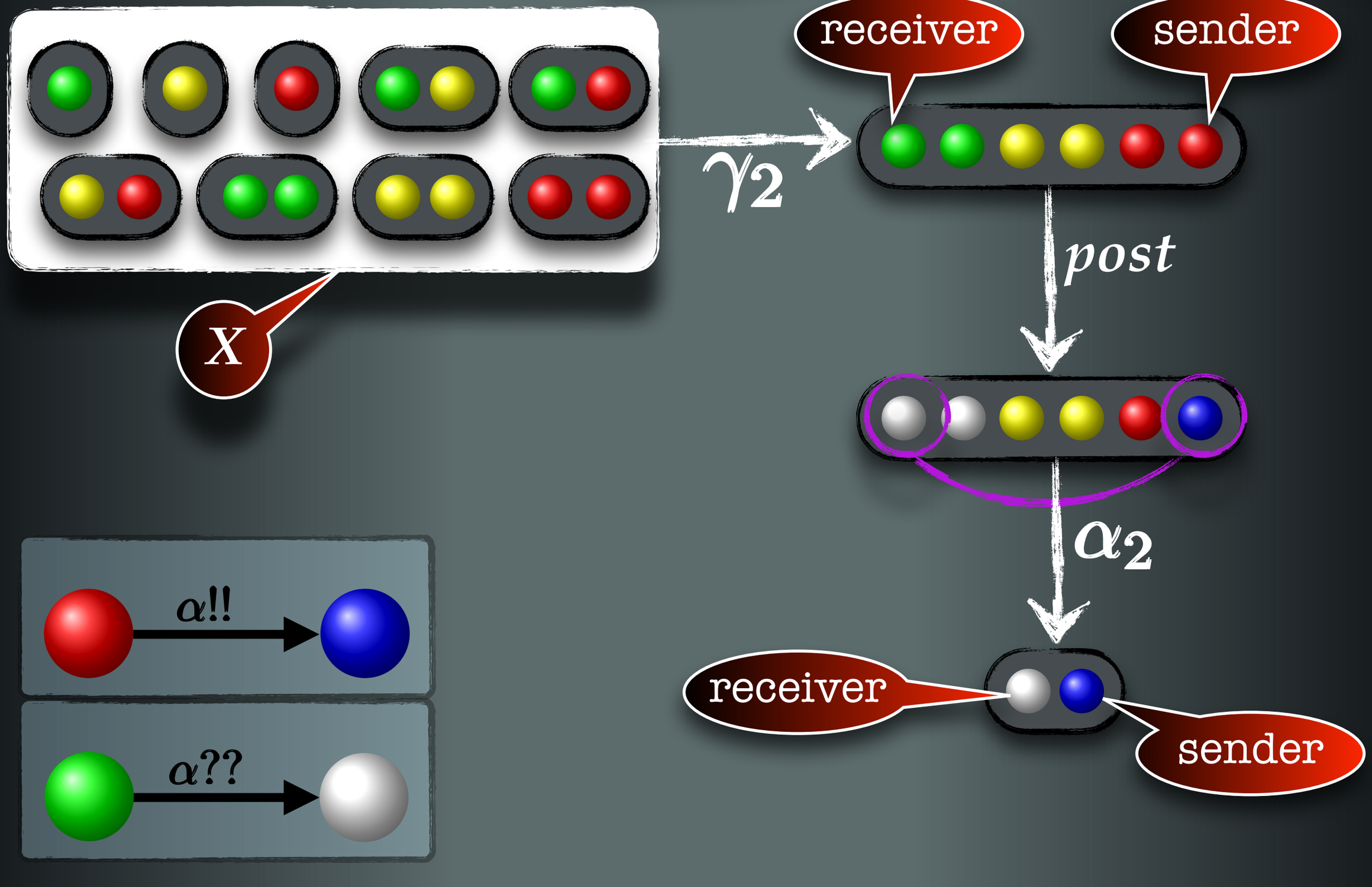
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



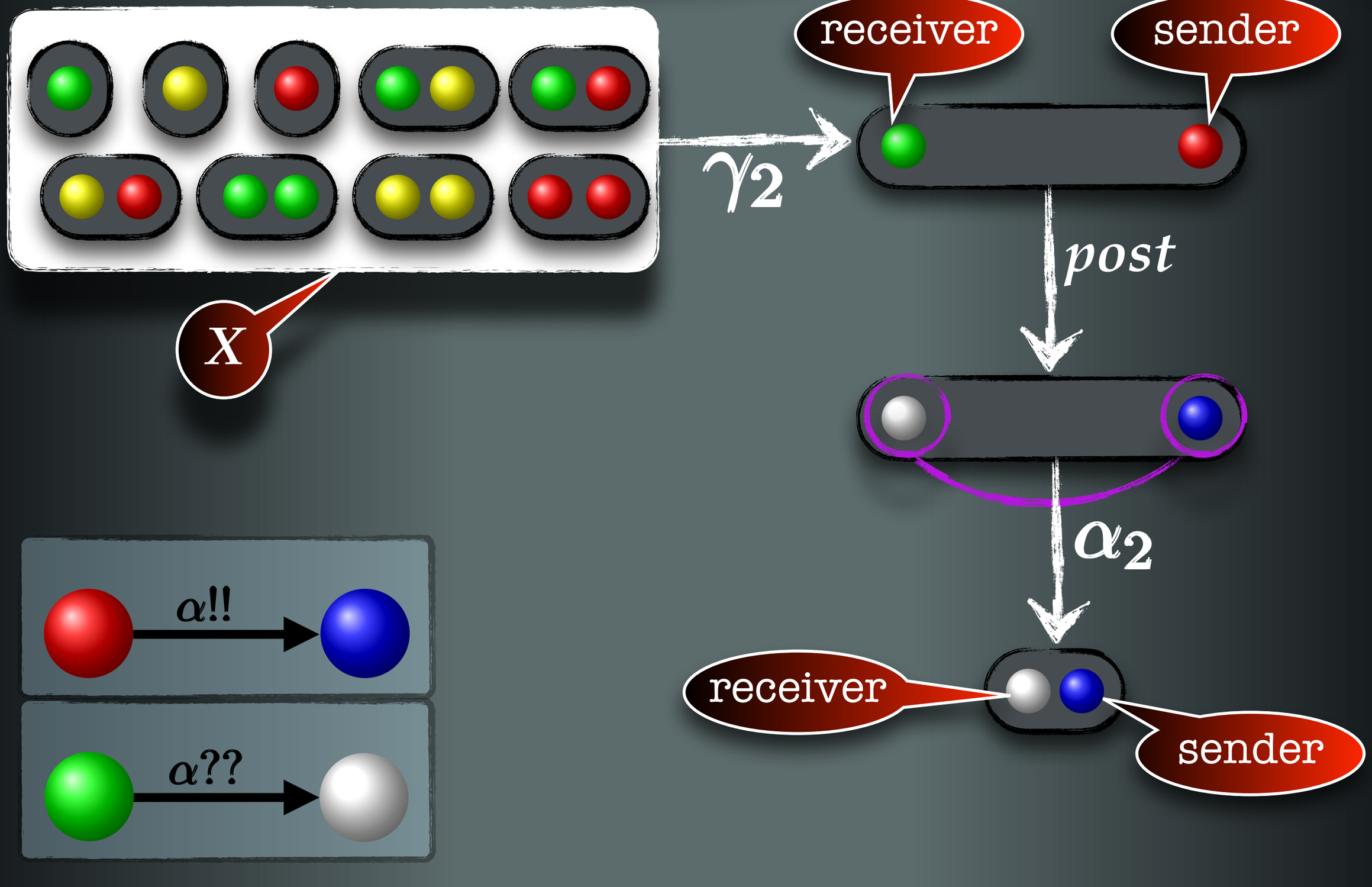
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



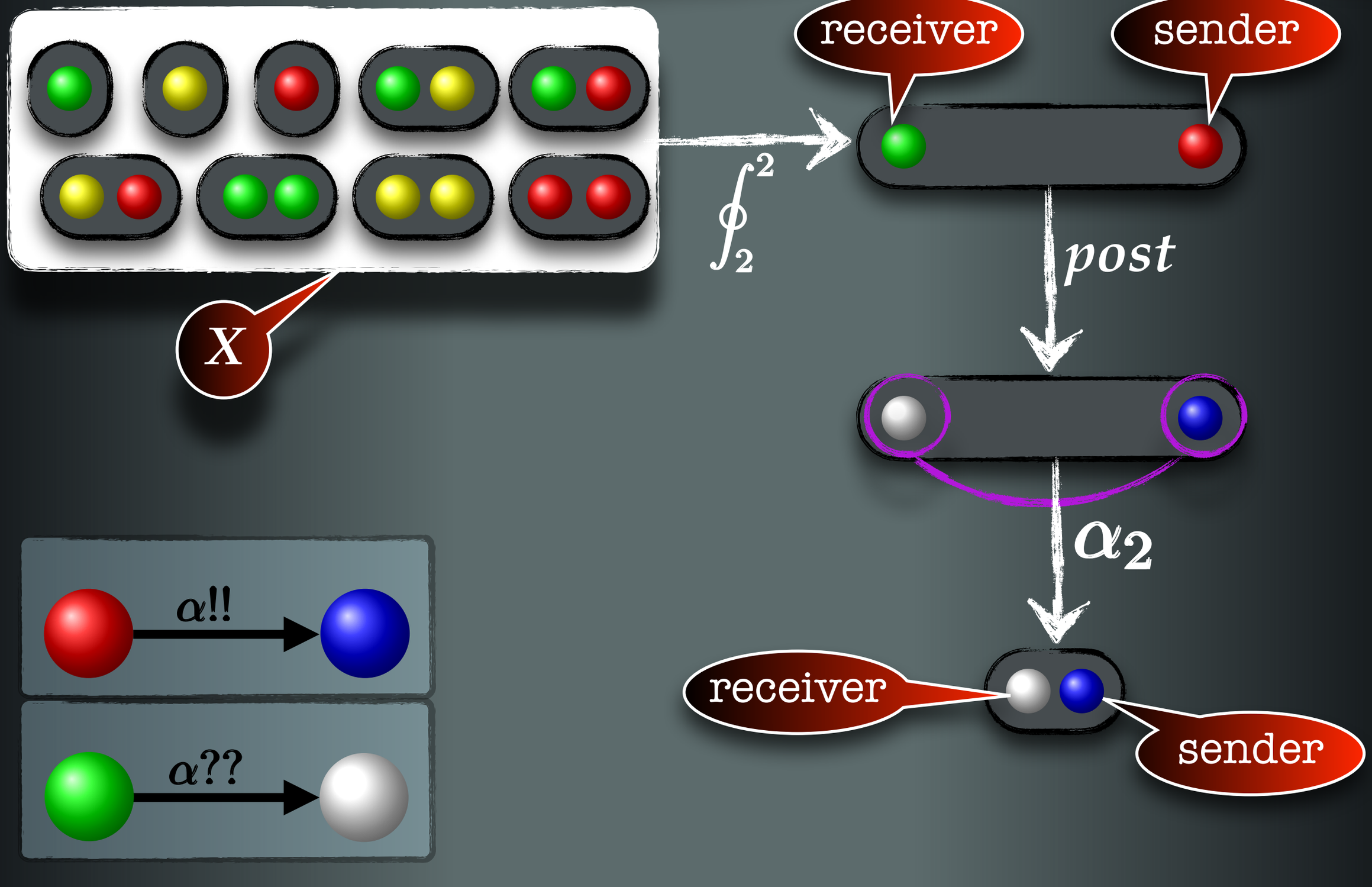
$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



$$\alpha_k(post(\gamma_k(X))) = \alpha_k(post(\oint_k^{k+1}(X)))$$



non-atomic global conditions

shared variables

broadcast

rendez-vous

process creation/deletion

Topologies

- linear
- ring
- tree
- multiset

non-atomic global conditions ✓

shared variables ✓

broadcast ✓

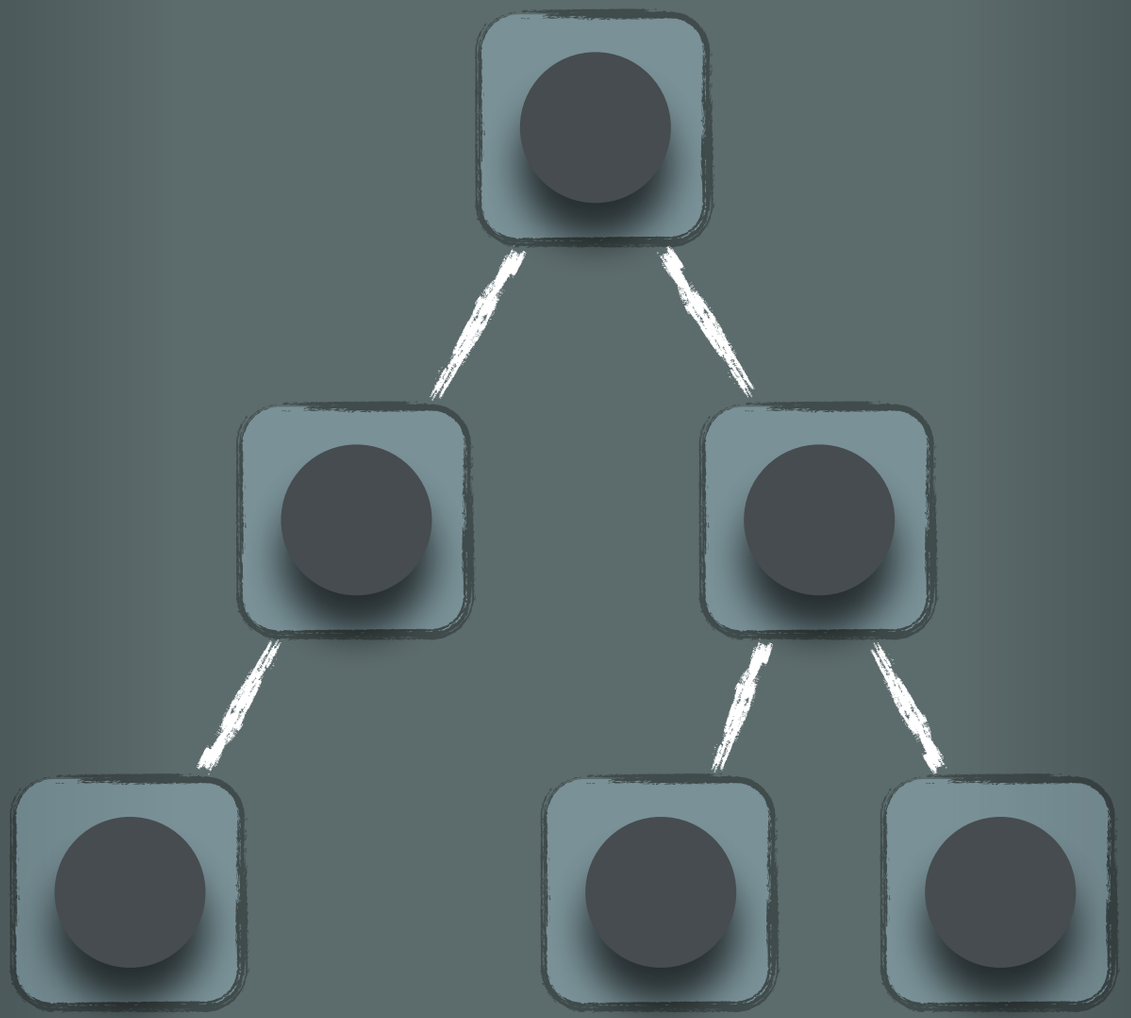
rendez-vous ✓

process creation/deletion ✓

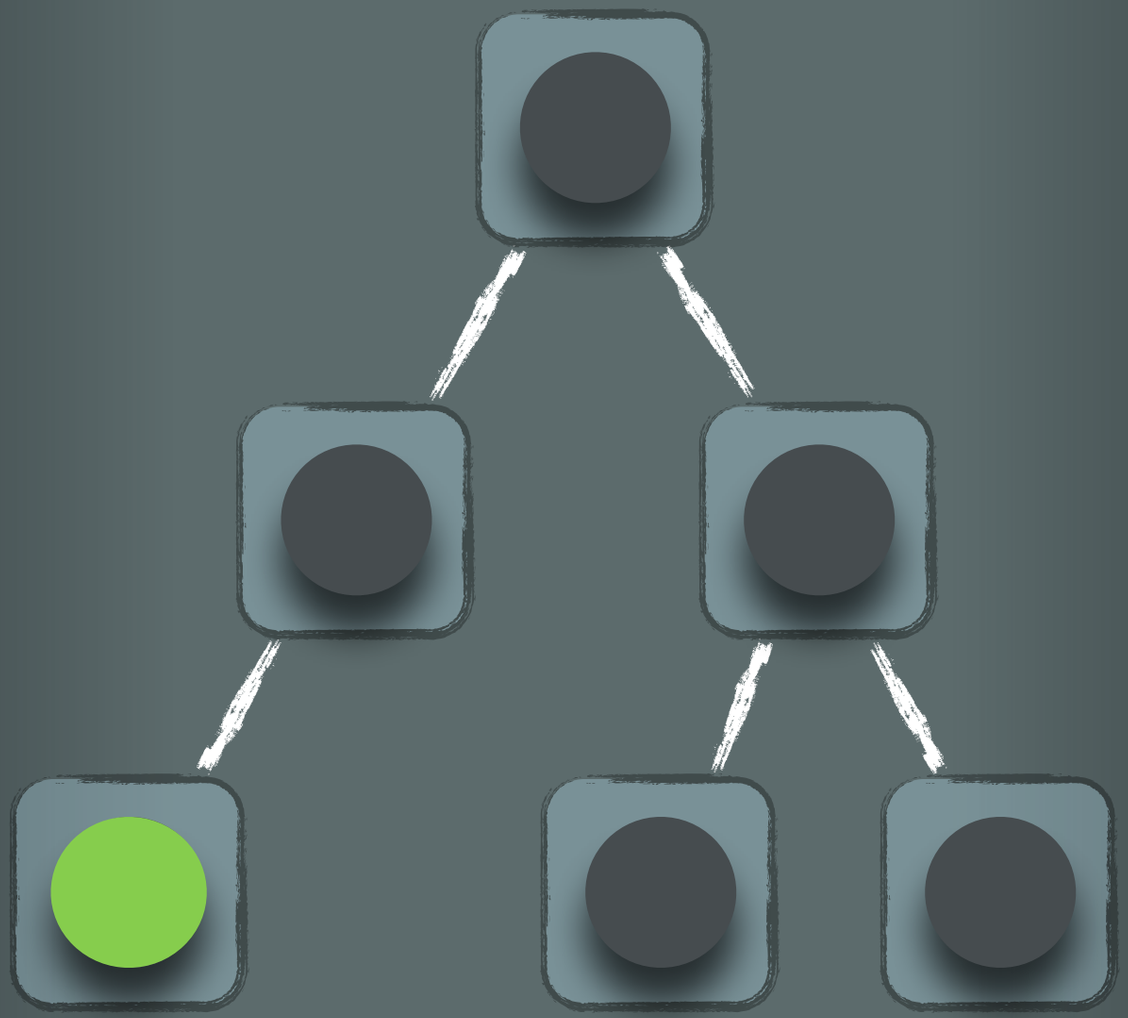
Topologies

- linear
- ring
- tree
- multiset

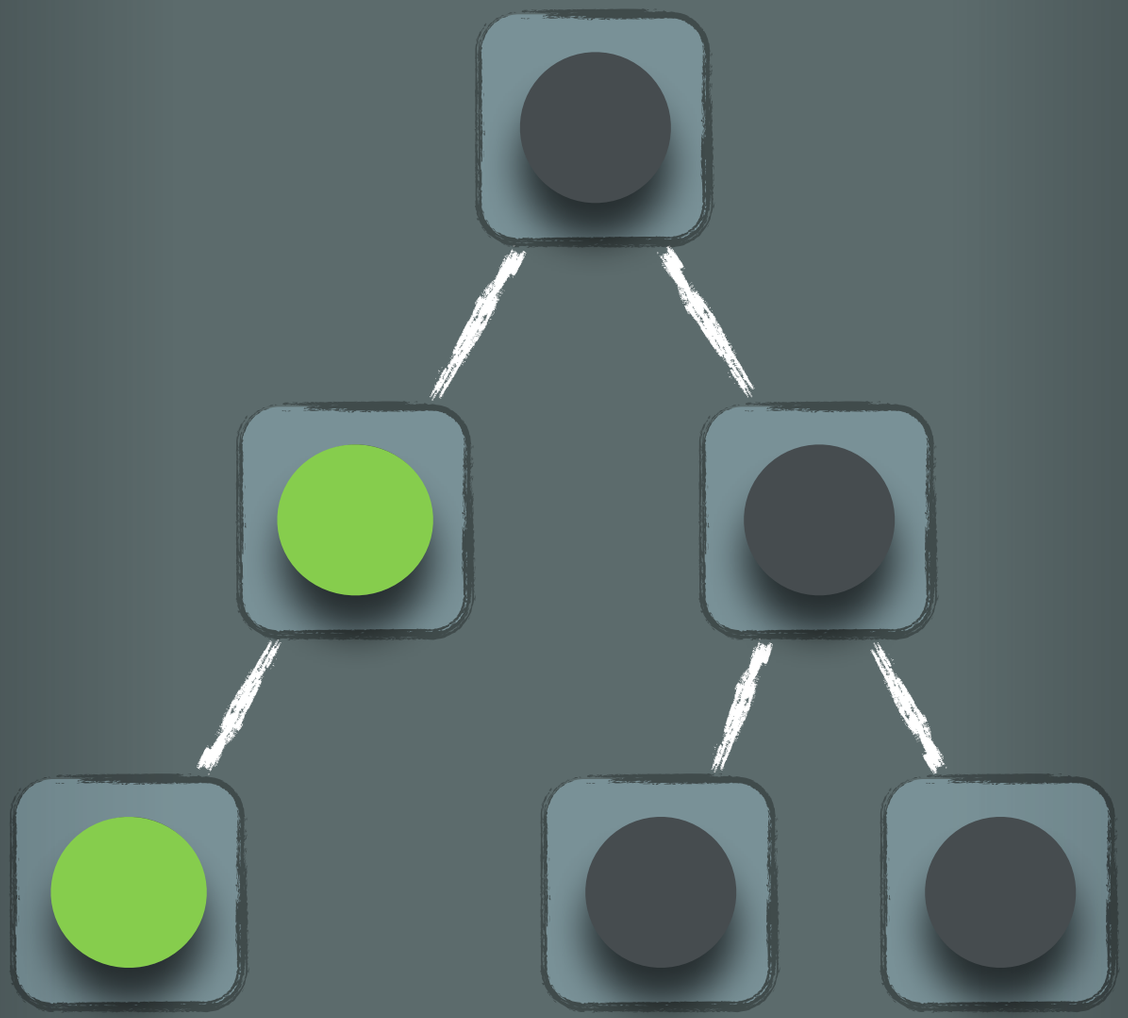
Leader Election



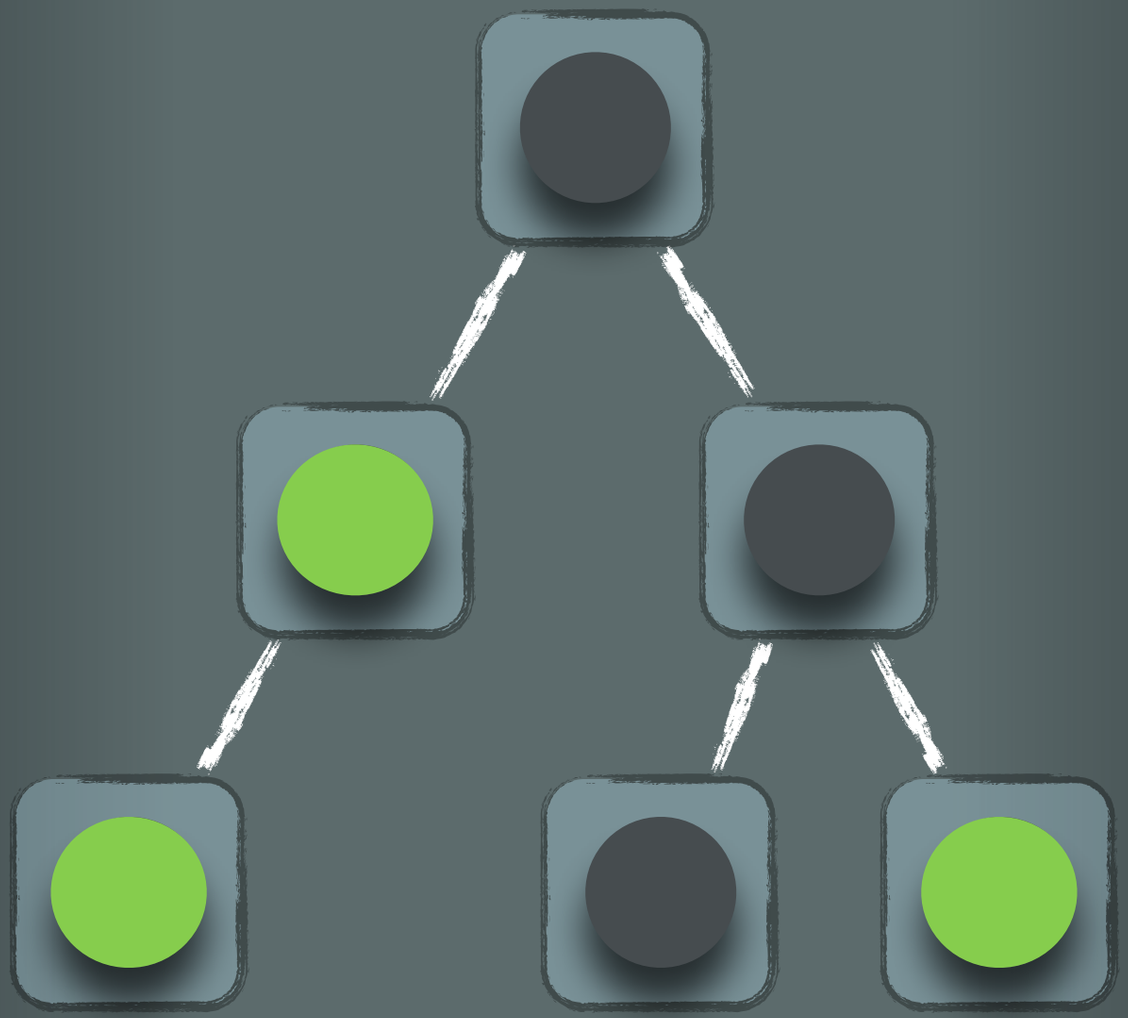
Leader Election



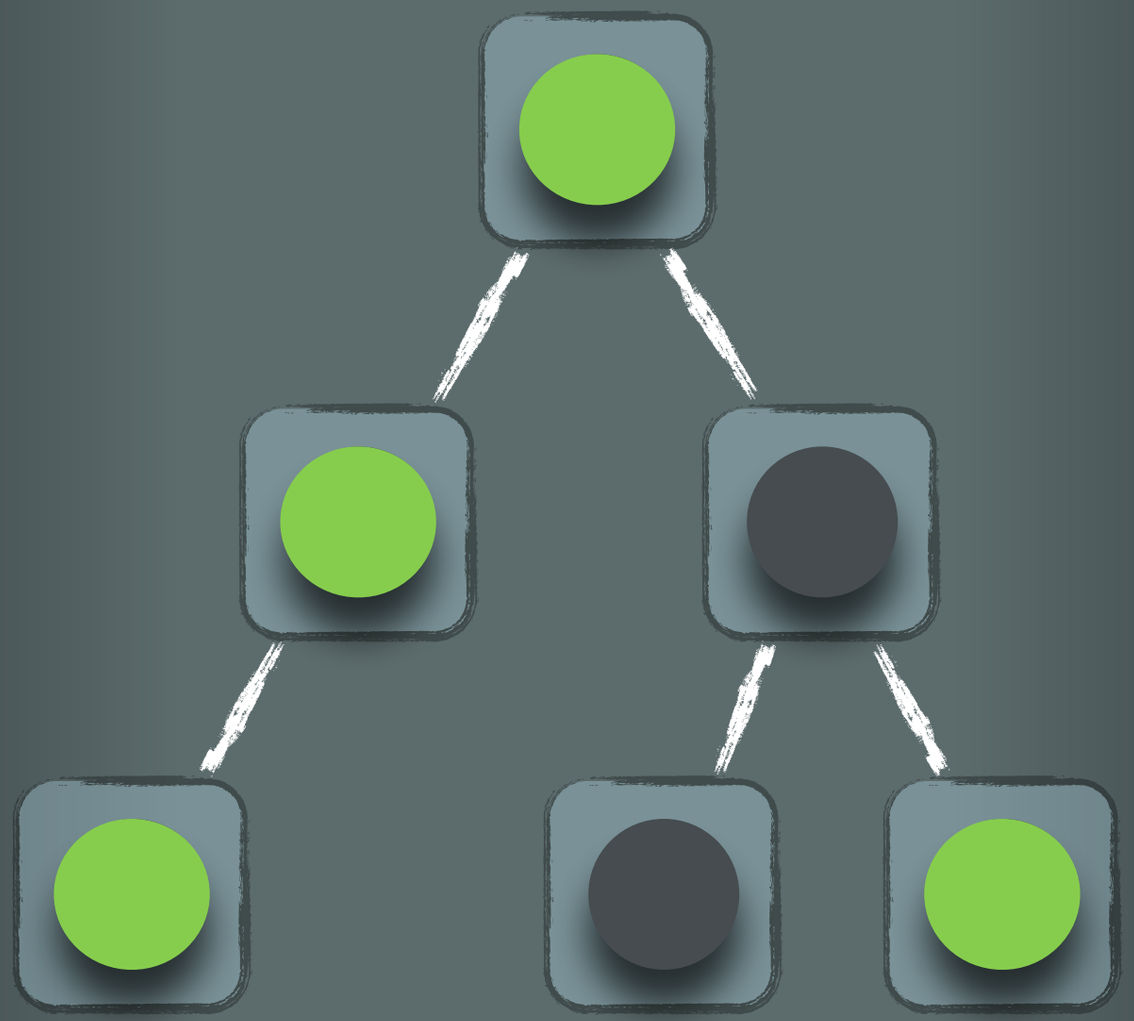
Leader Election



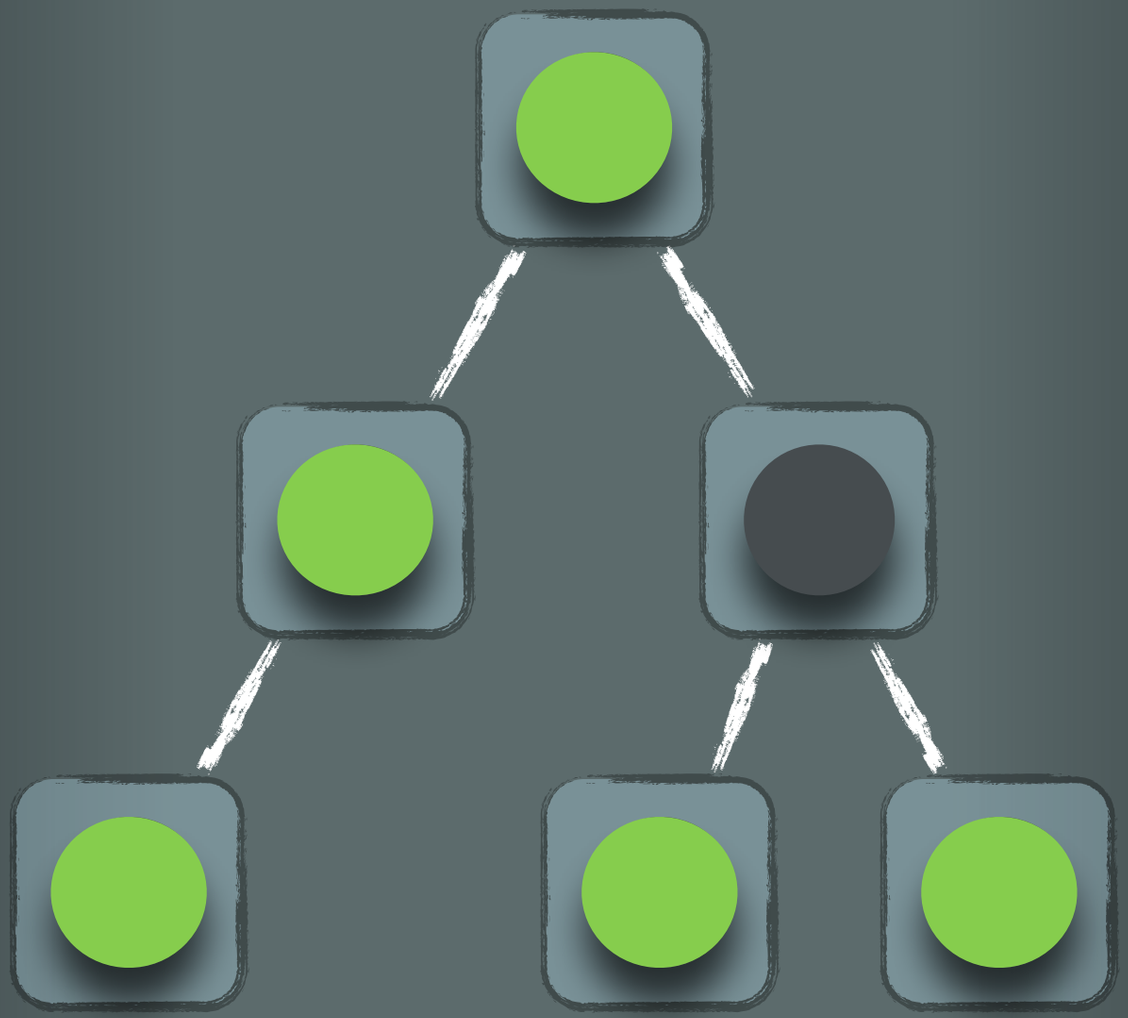
Leader Election



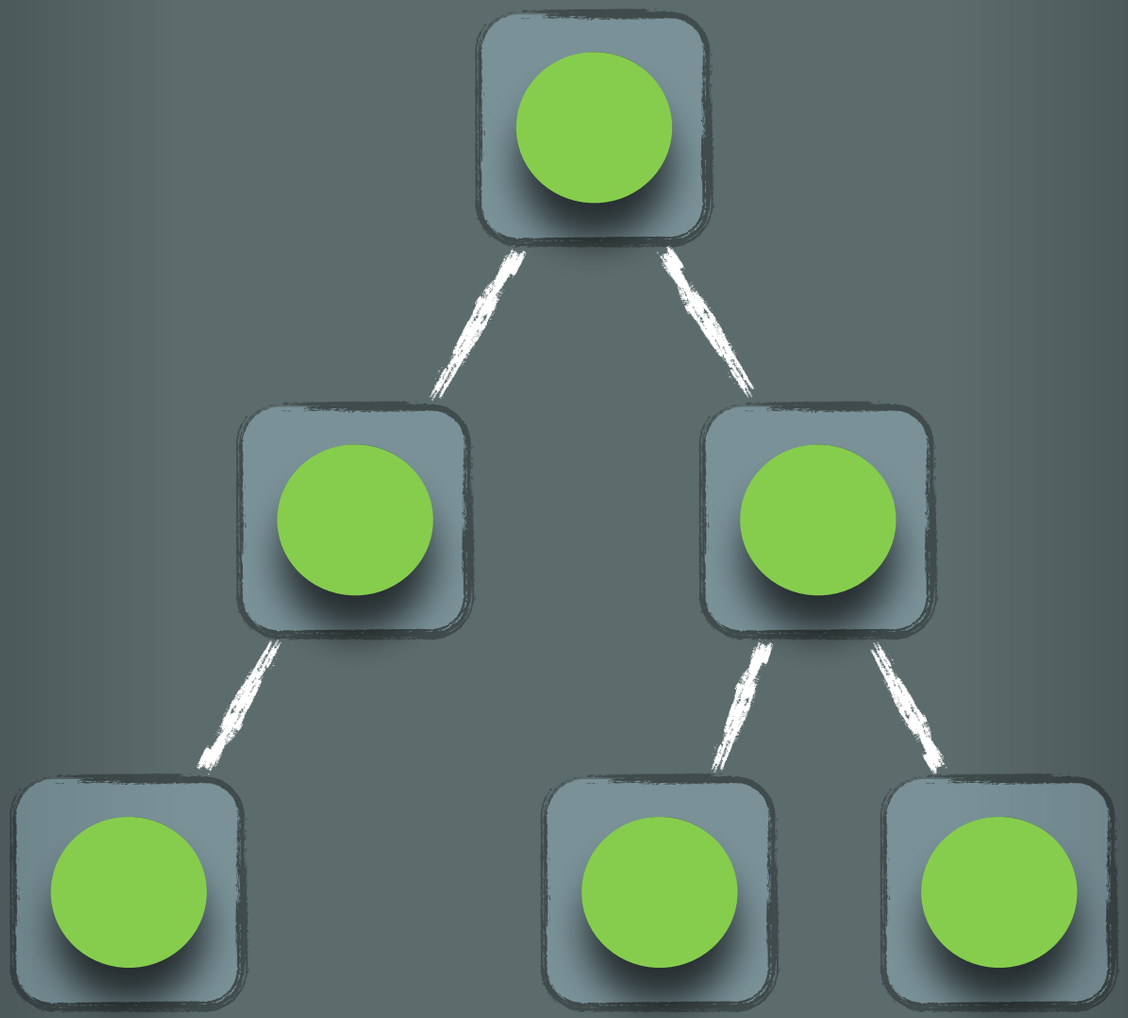
Leader Election



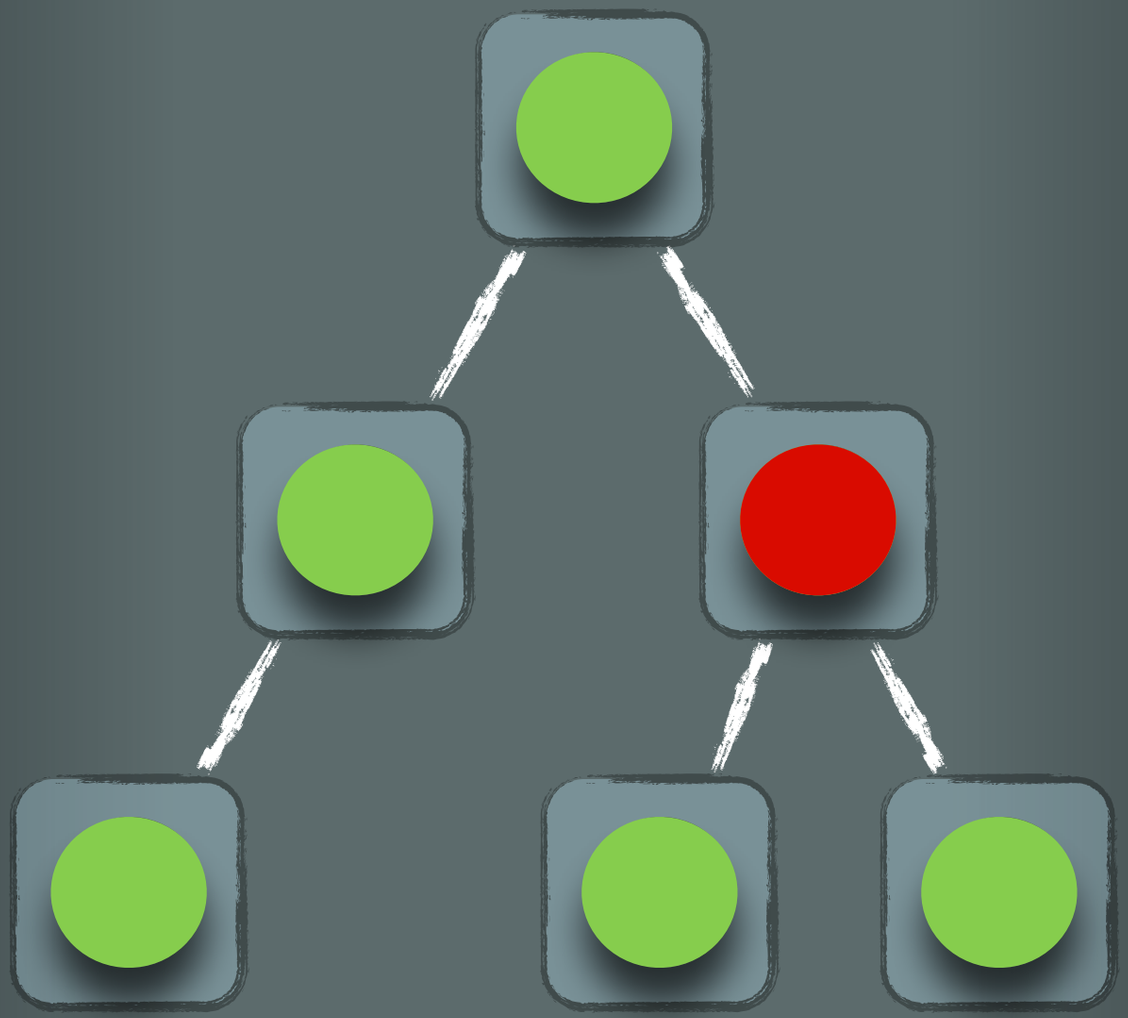
Leader Election



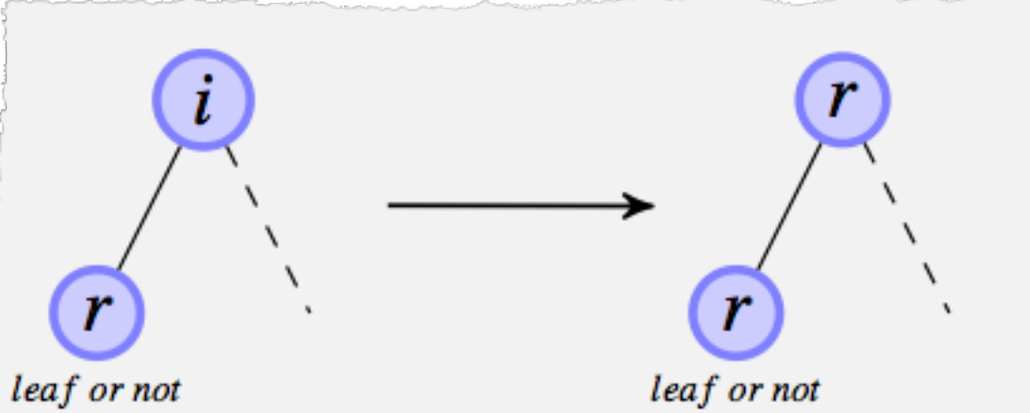
Leader Election



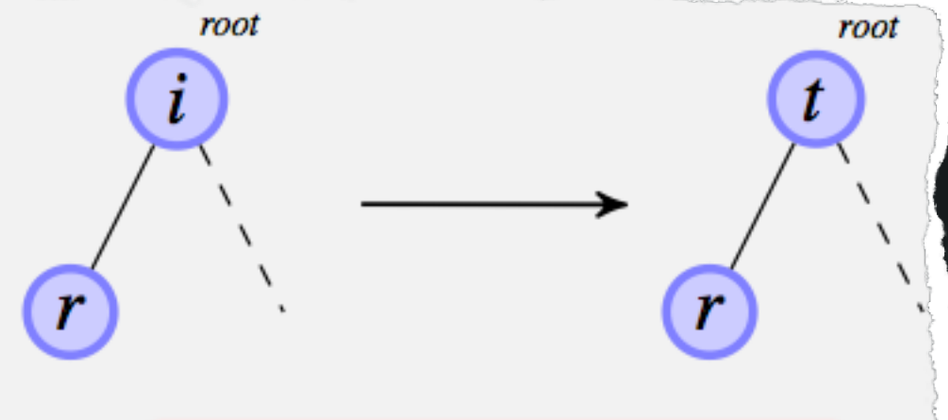
Leader Election



Leader Election



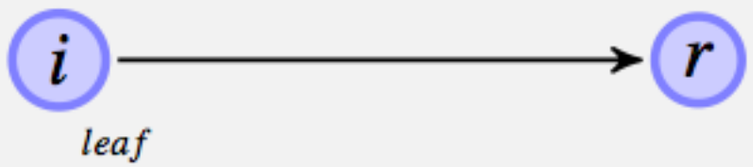
Upward propagation



The root initiates the downward propagation



Downward propagation: Election of a candidate



A process candidates

Parameterized System ✓

Analysis ✓
Abstraction

Features

Completeness

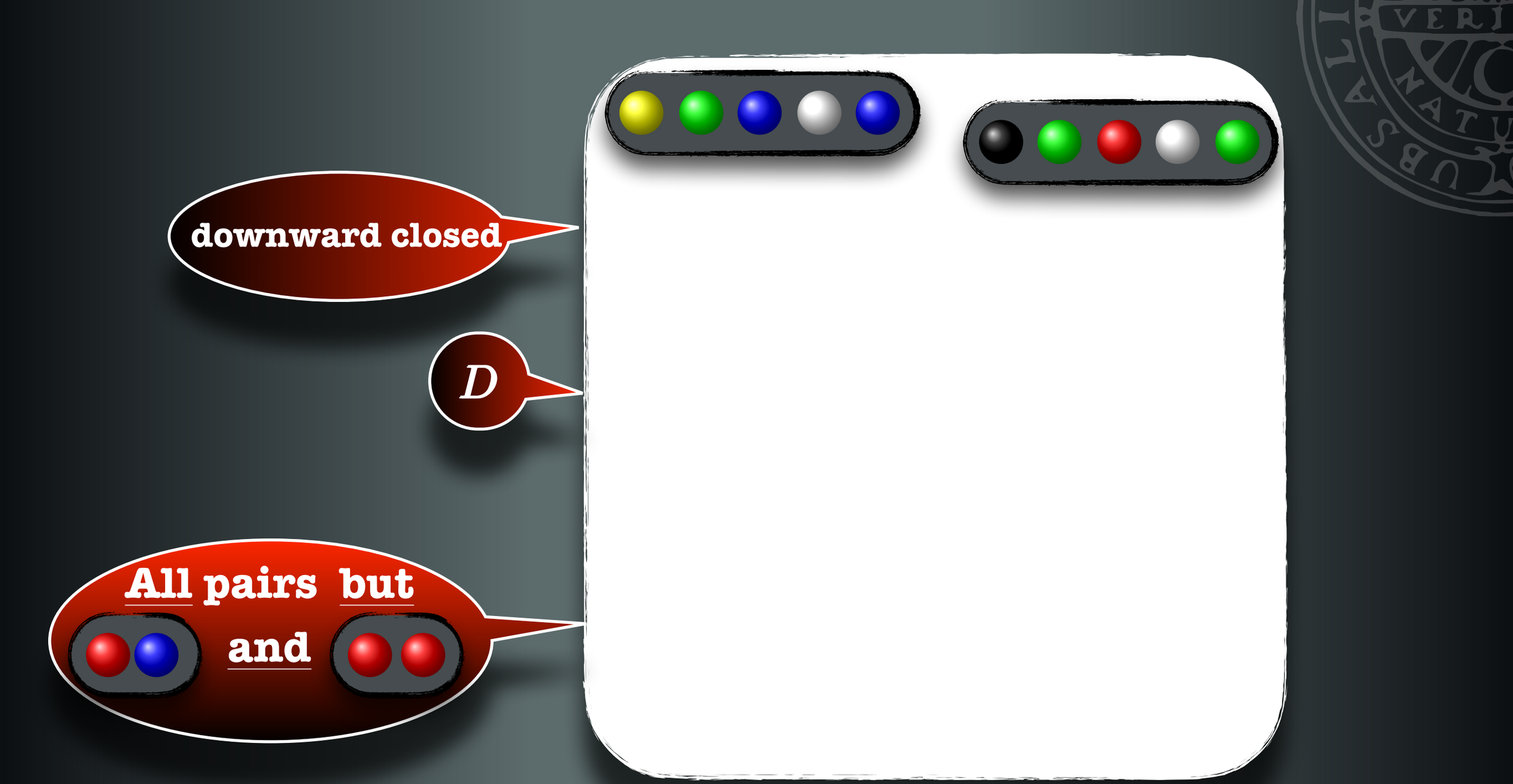
Parameterized System ✓

Analysis / Abstraction ✓

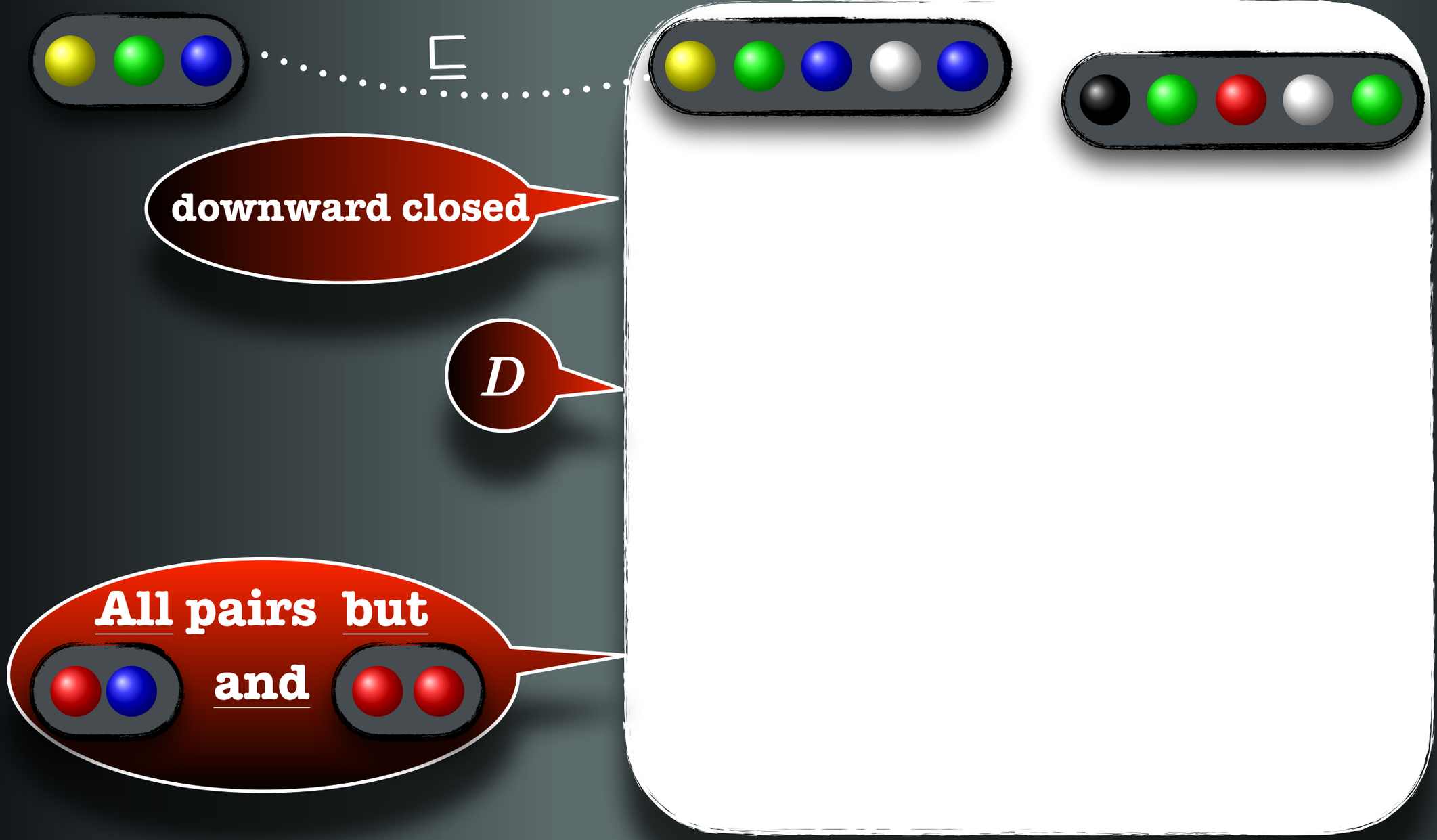
Features ✓

Completeness

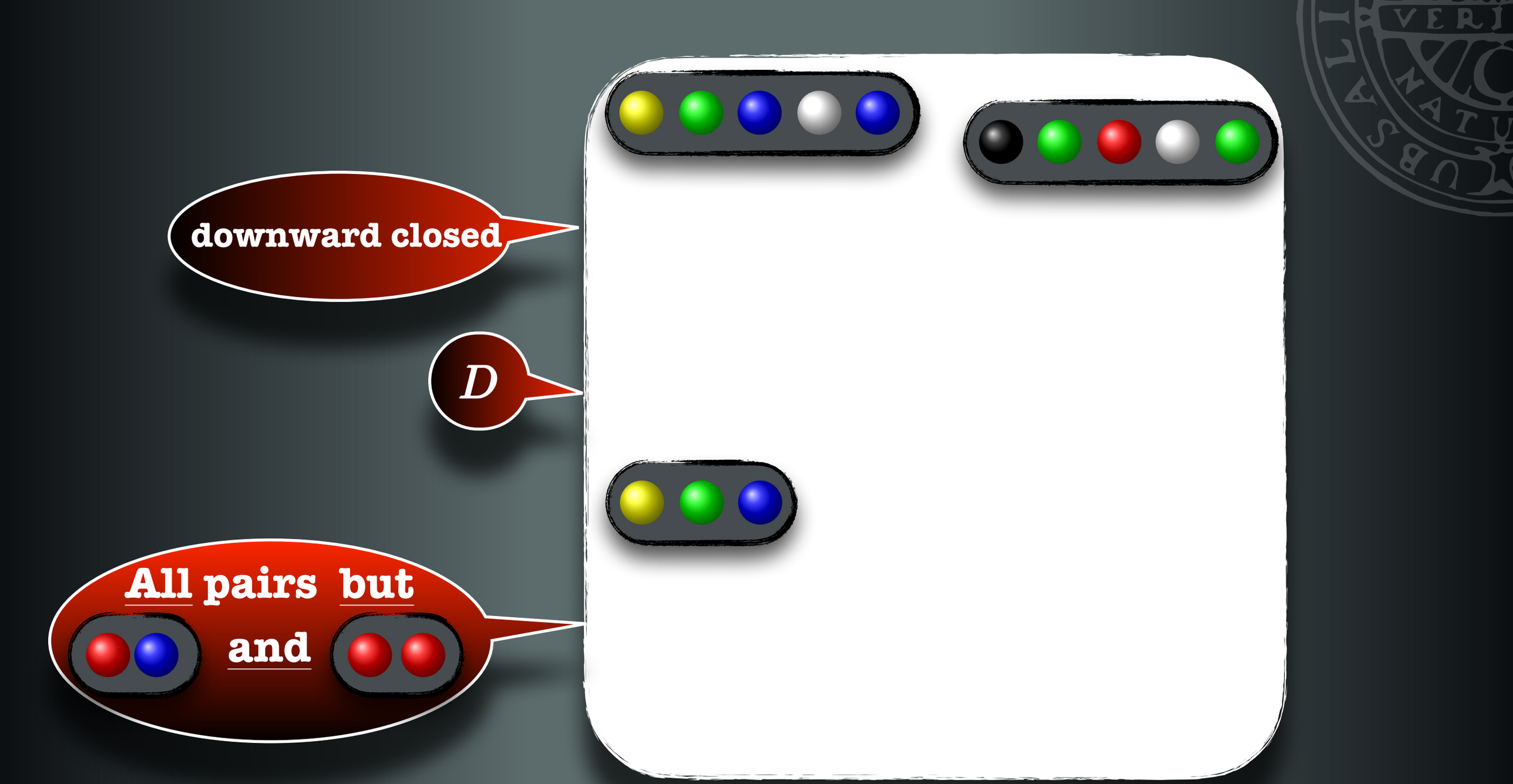
Completeness



- if $c_1 \in D$ and $c_2 \sqsubseteq c_1$
- then $c_2 \in D$



- if $c_1 \in D$ and $c_2 \sqsubseteq c_1$
- then $c_2 \in D$



- if $c_1 \in D$ and $c_2 \sqsubseteq c_1$
- then $c_2 \in D$

downward closed

D

All pairs but and

$\mathcal{U}$

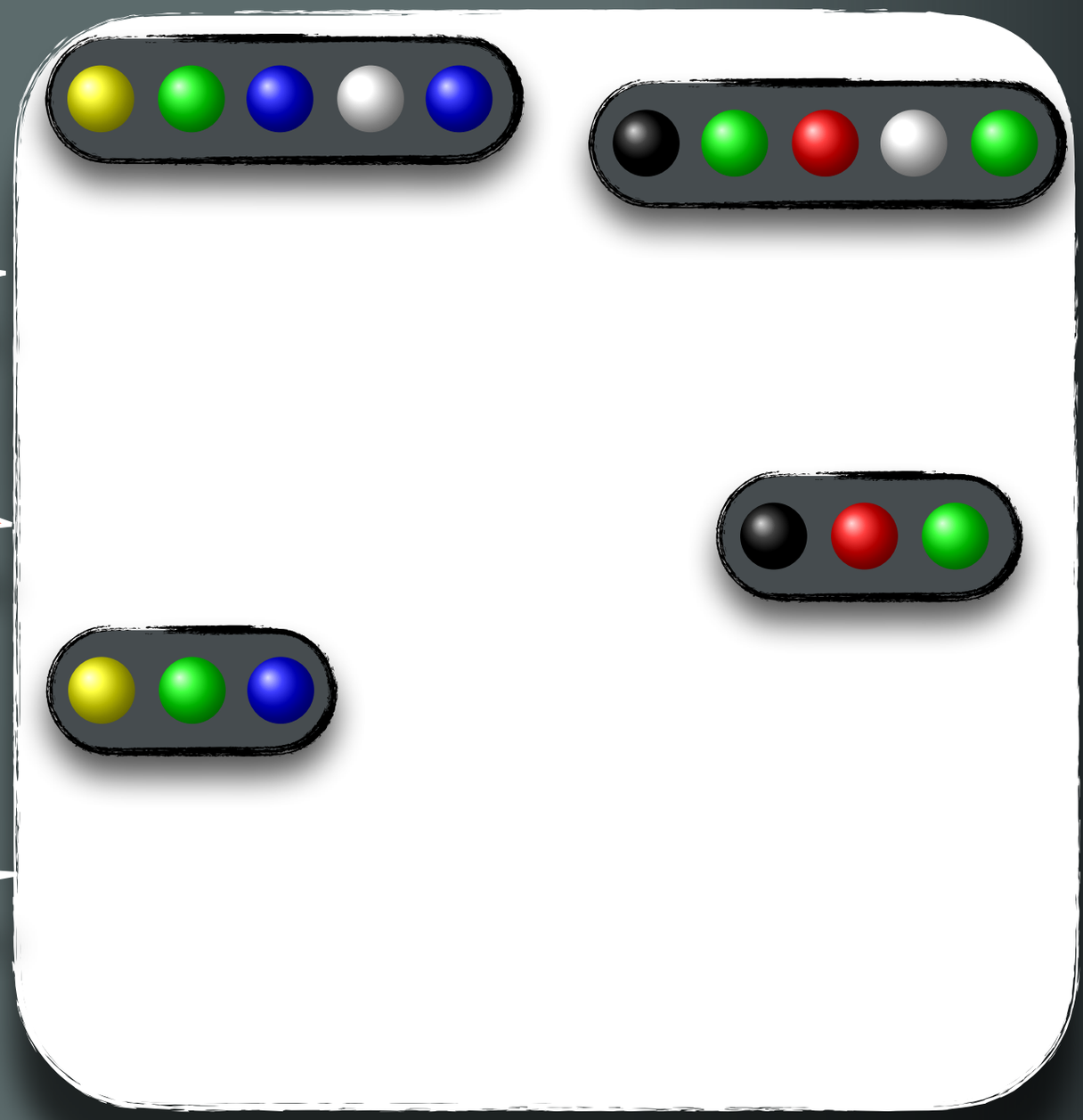
- if $c_1 \in D$ and $c_2 \sqsubseteq c_1$
- then $c_2 \in D$

downward closed

D

All pairs but

and



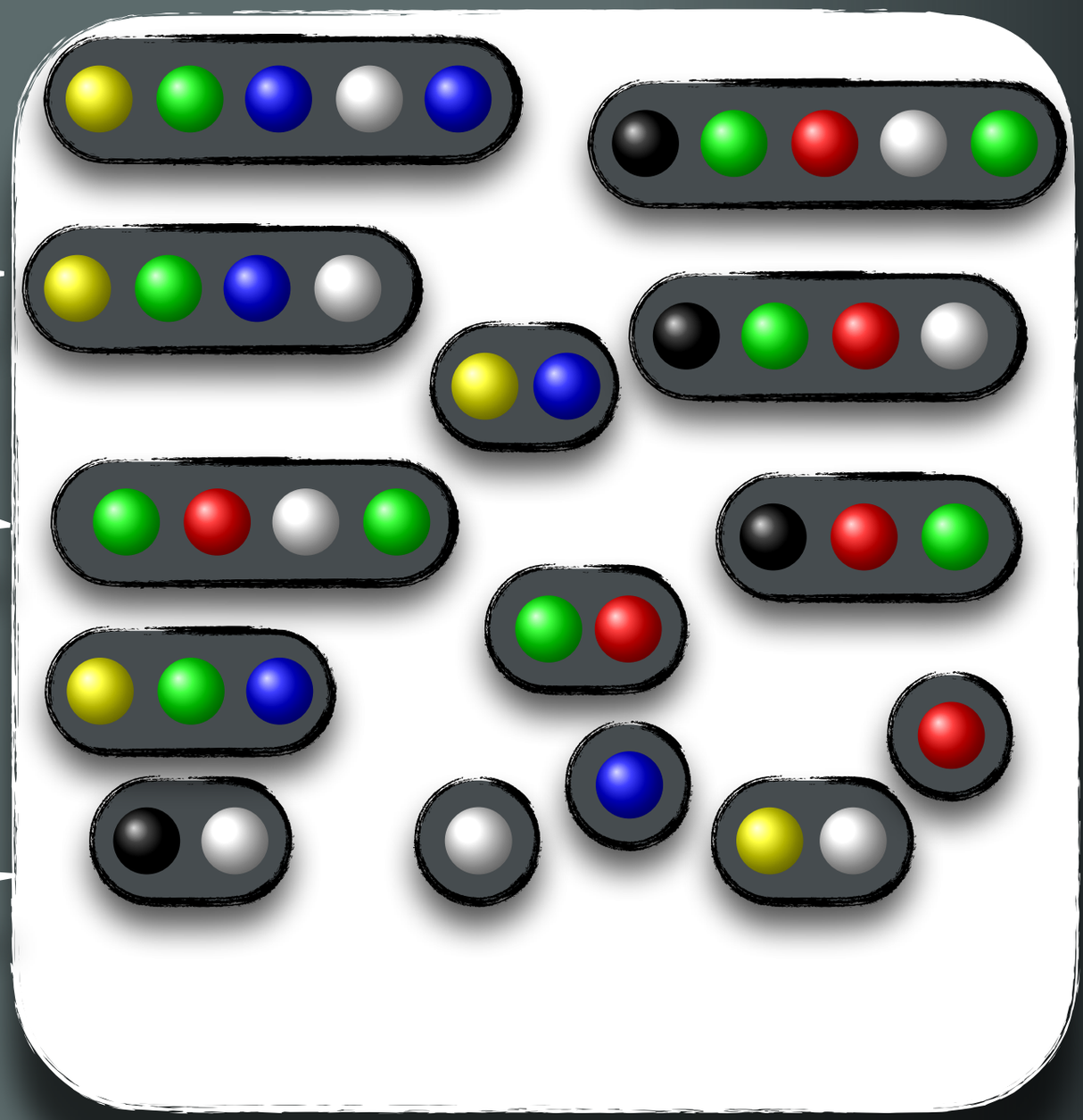
- if $c_1 \in D$ and $c_2 \sqsubseteq c_1$
- then $c_2 \in D$

downward closed

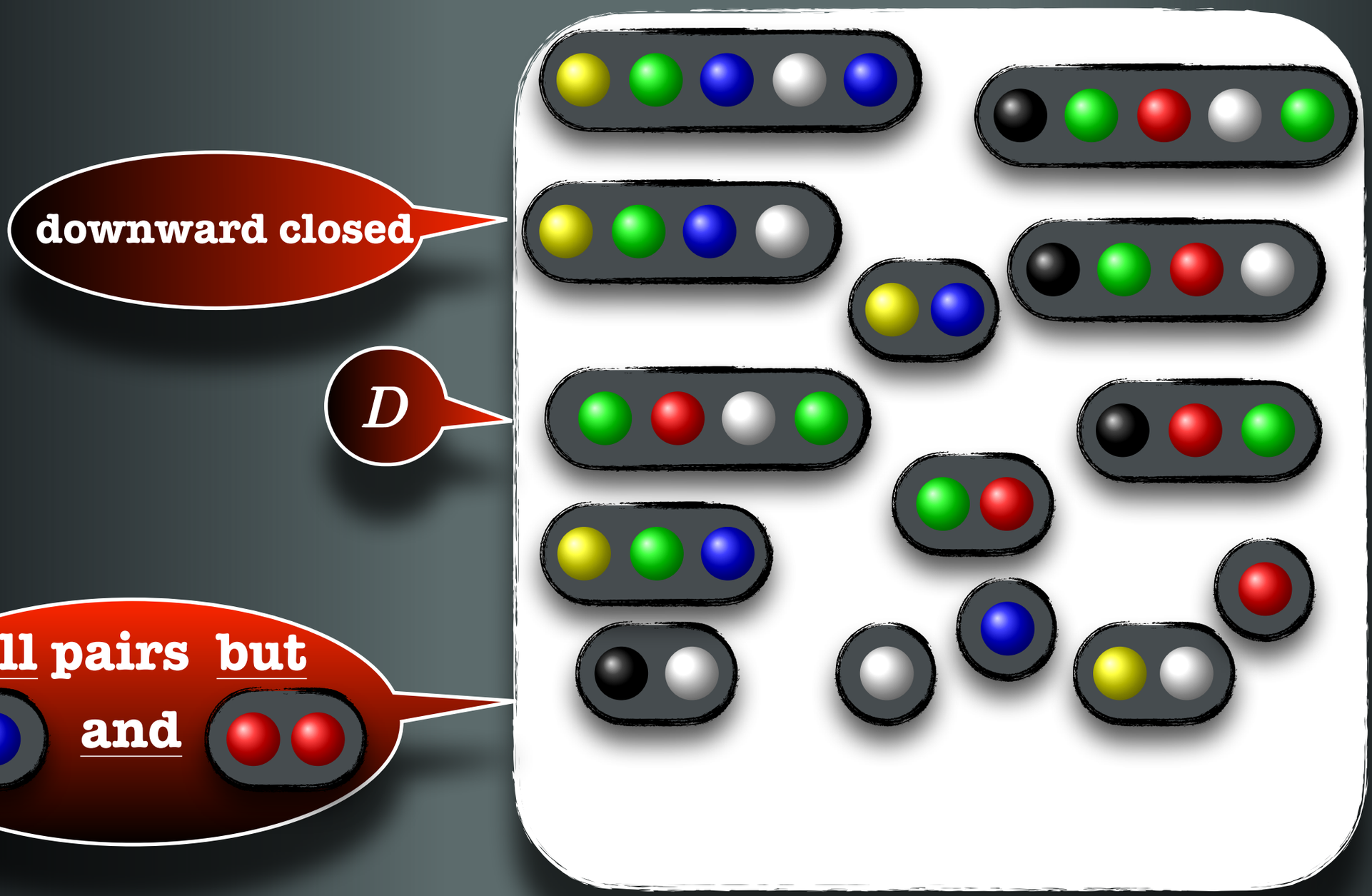
D

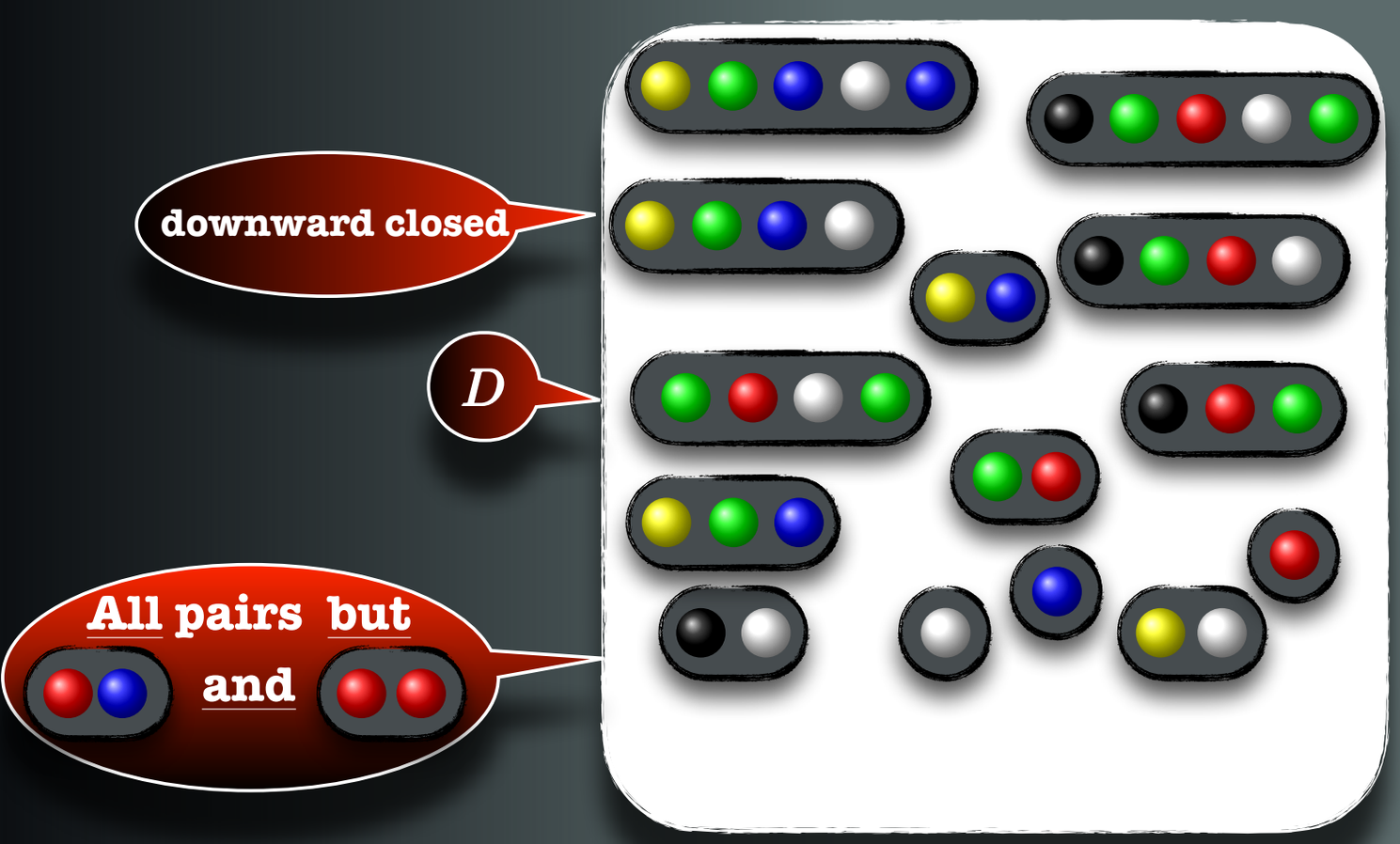
All pairs but

and



- if $c_1 \in D$ and $c_2 \sqsubseteq c_1$
- then $c_2 \in D$


downward closed
D
All pairs but
and


downward closed
D
All pairs but
and

downward closed

D

All pairs but

and

upward closed

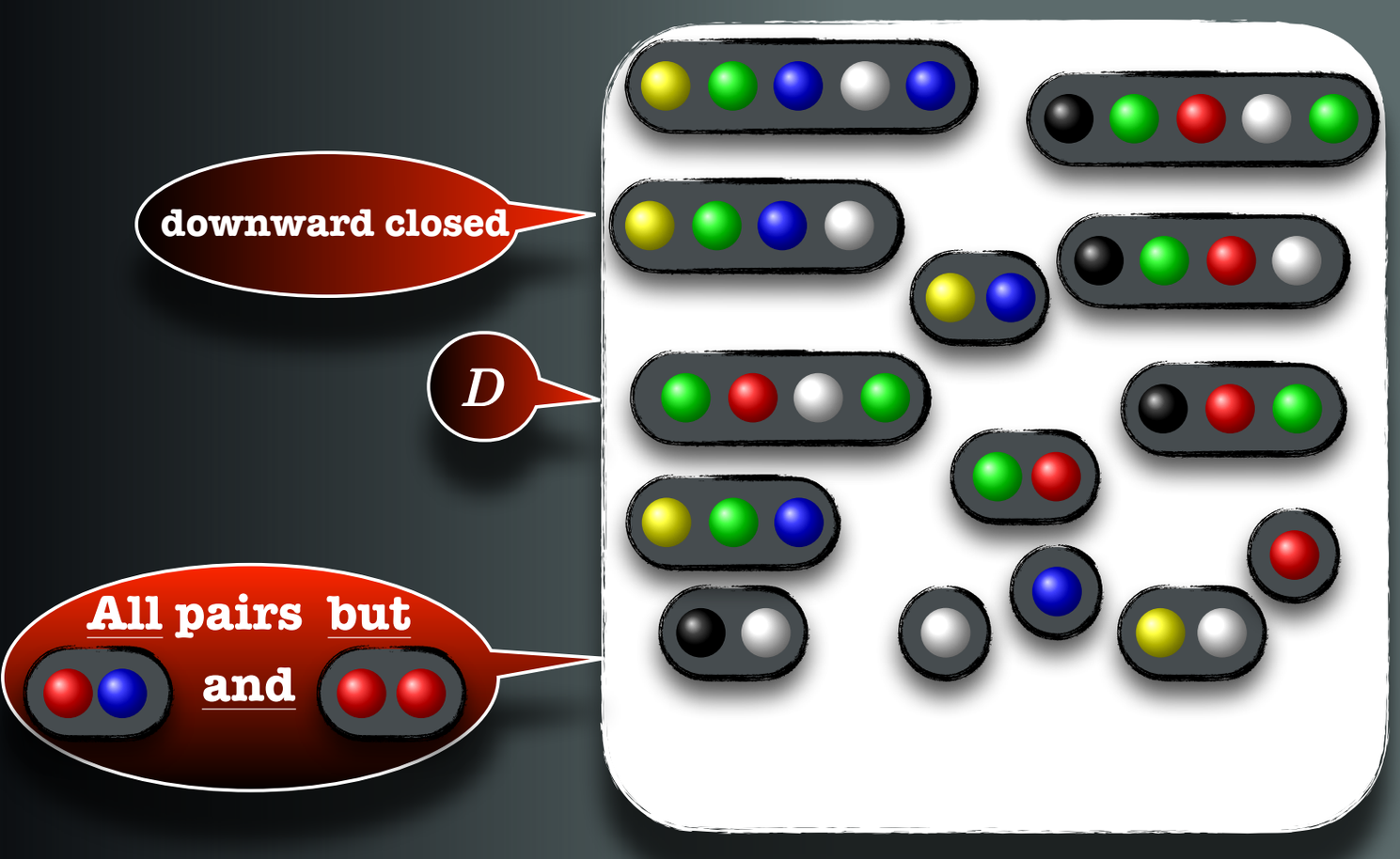
U

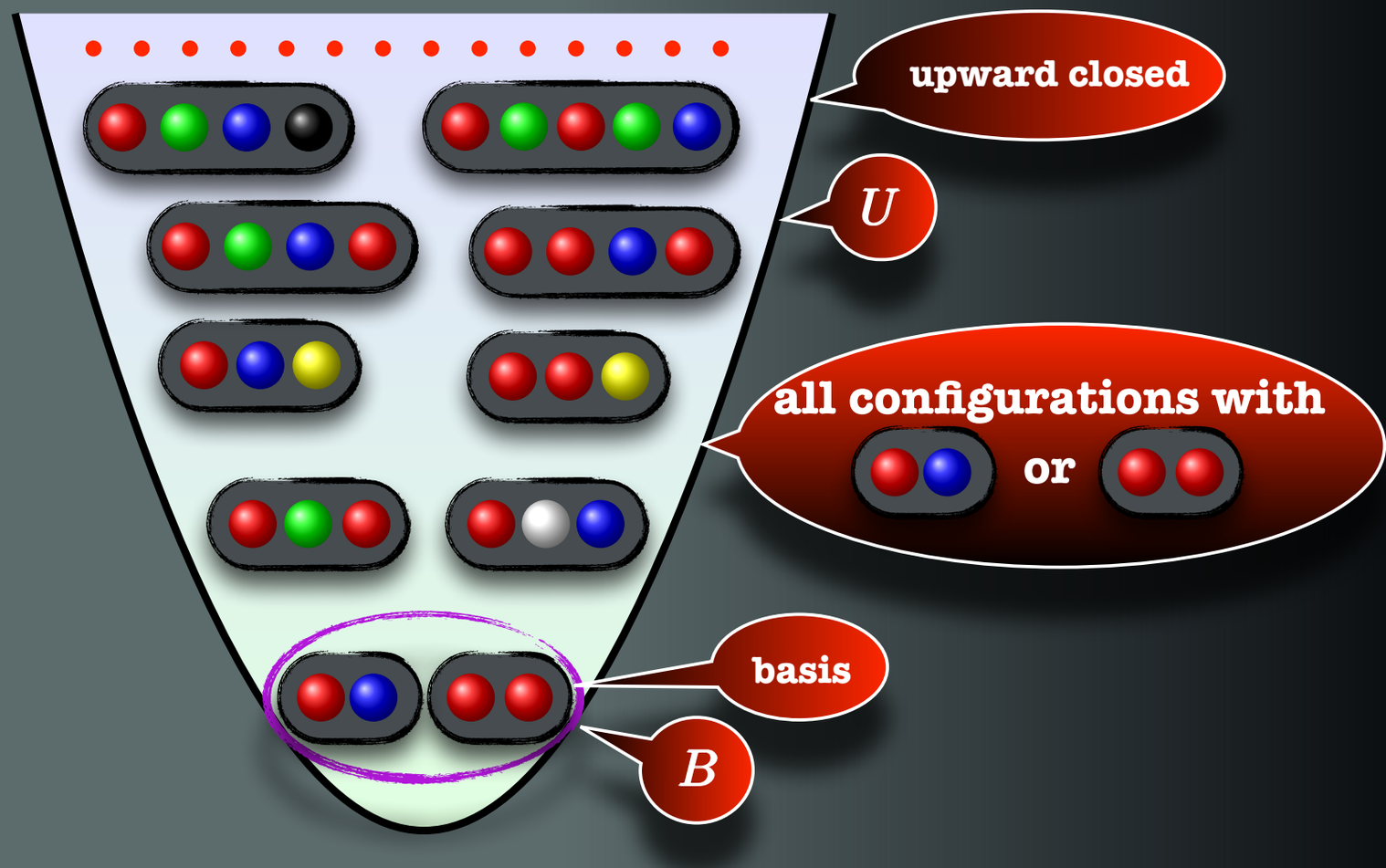
all configurations with

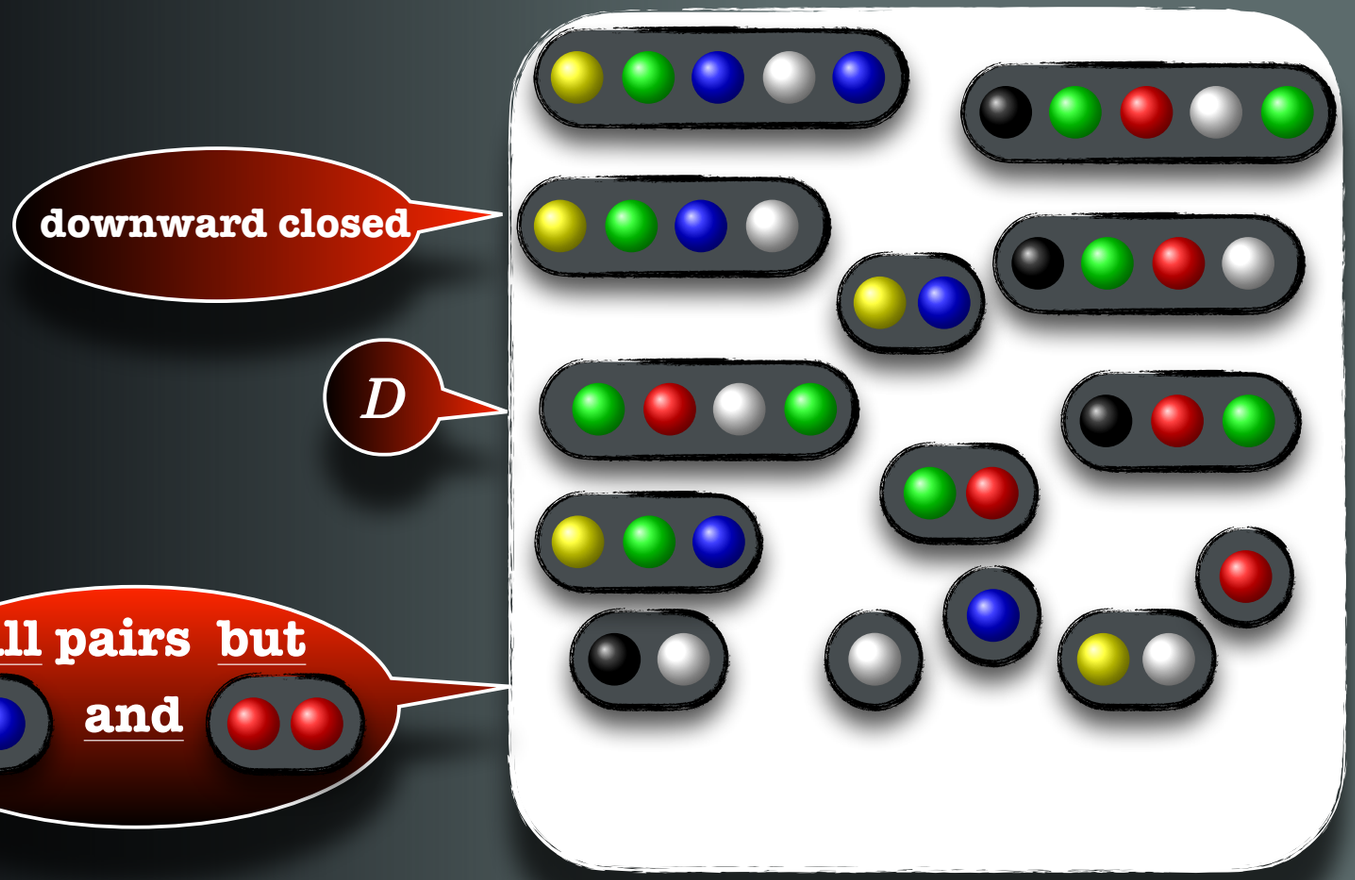
or

basis

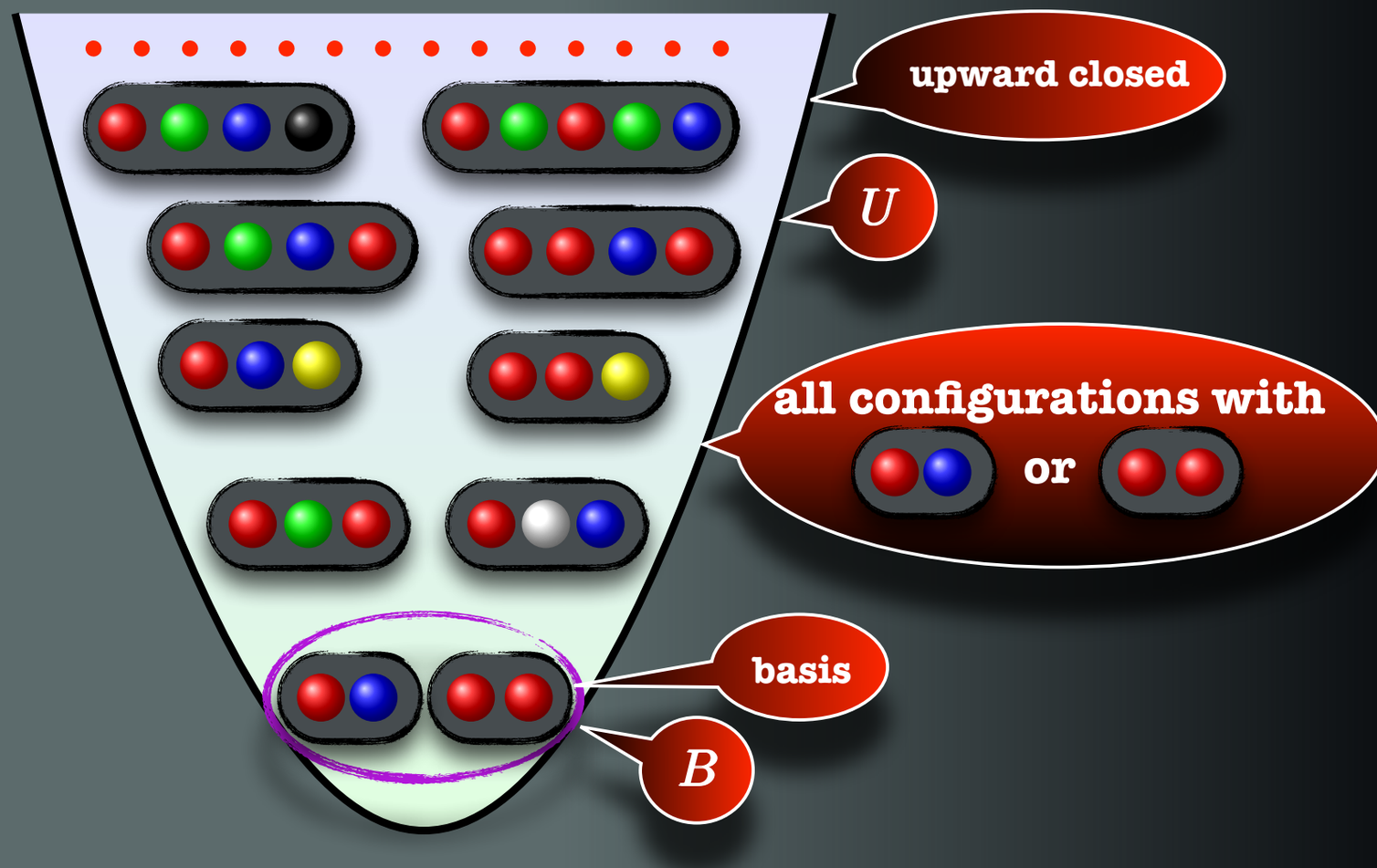
B

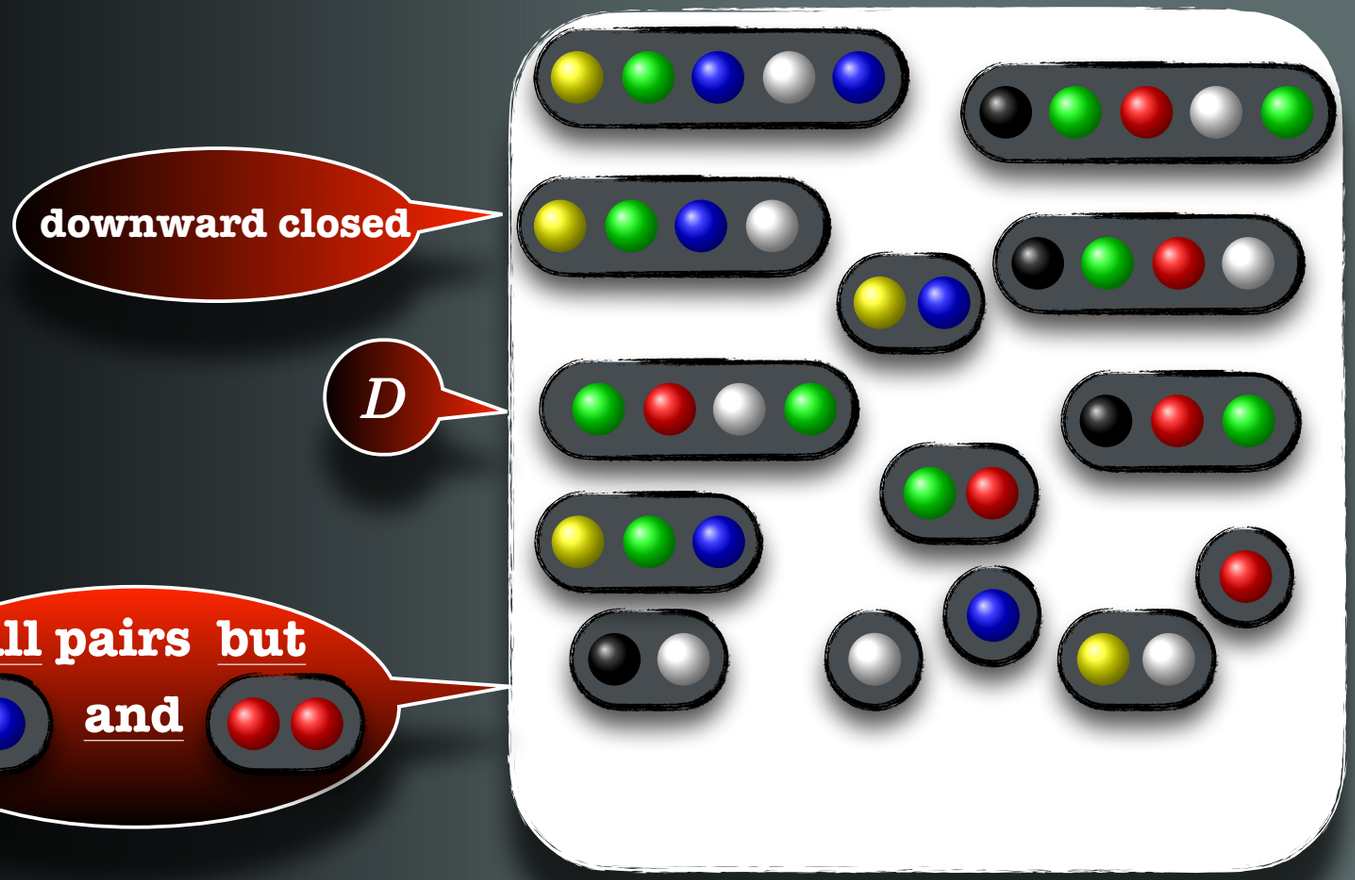

downward closed
D
All pairs but
and


upward closed
U
all configurations with
or
basis
B

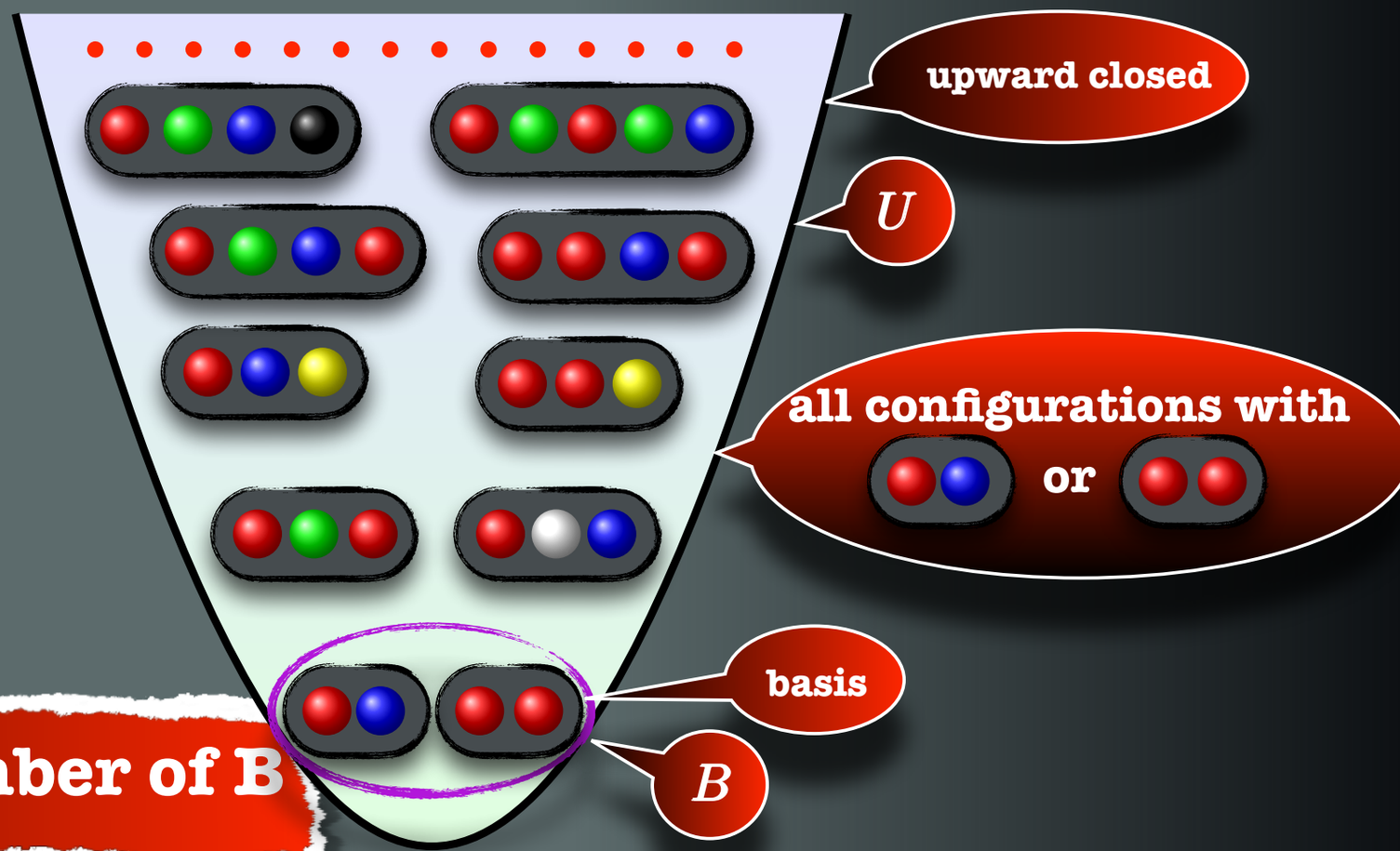

downward closed
D
All pairs but
and

$$\neg D = B$$


upward closed
U
all configurations with
or
basis
B



$$\neg D = B$$



$\#D =$ size of largest member of B

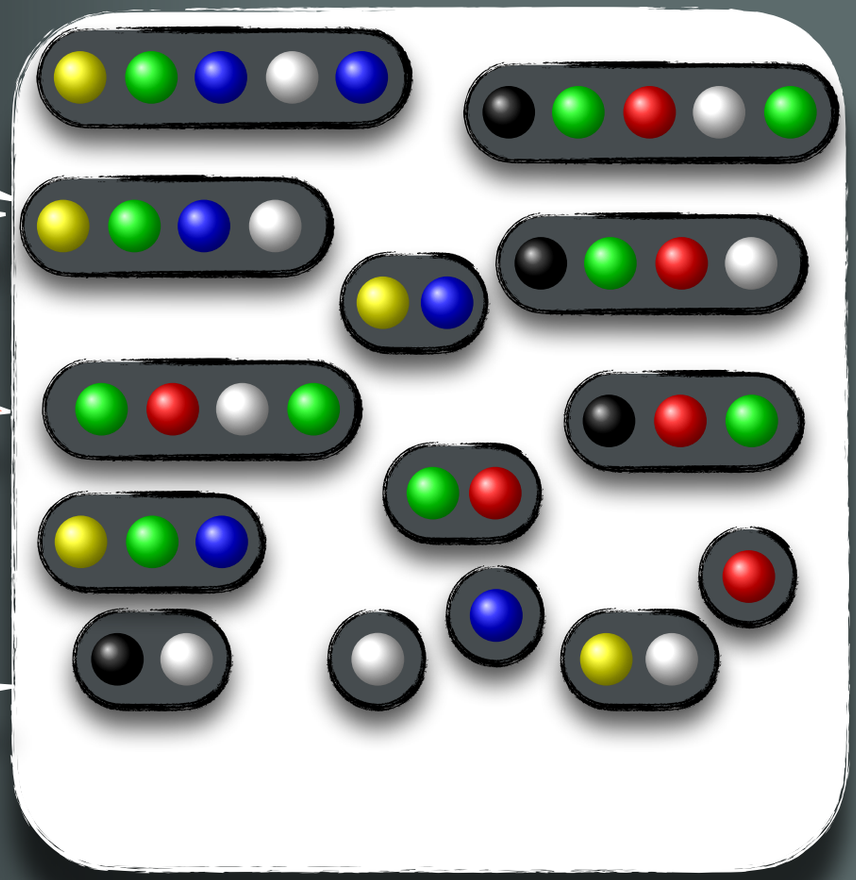
$\#D = 2$

downward closed

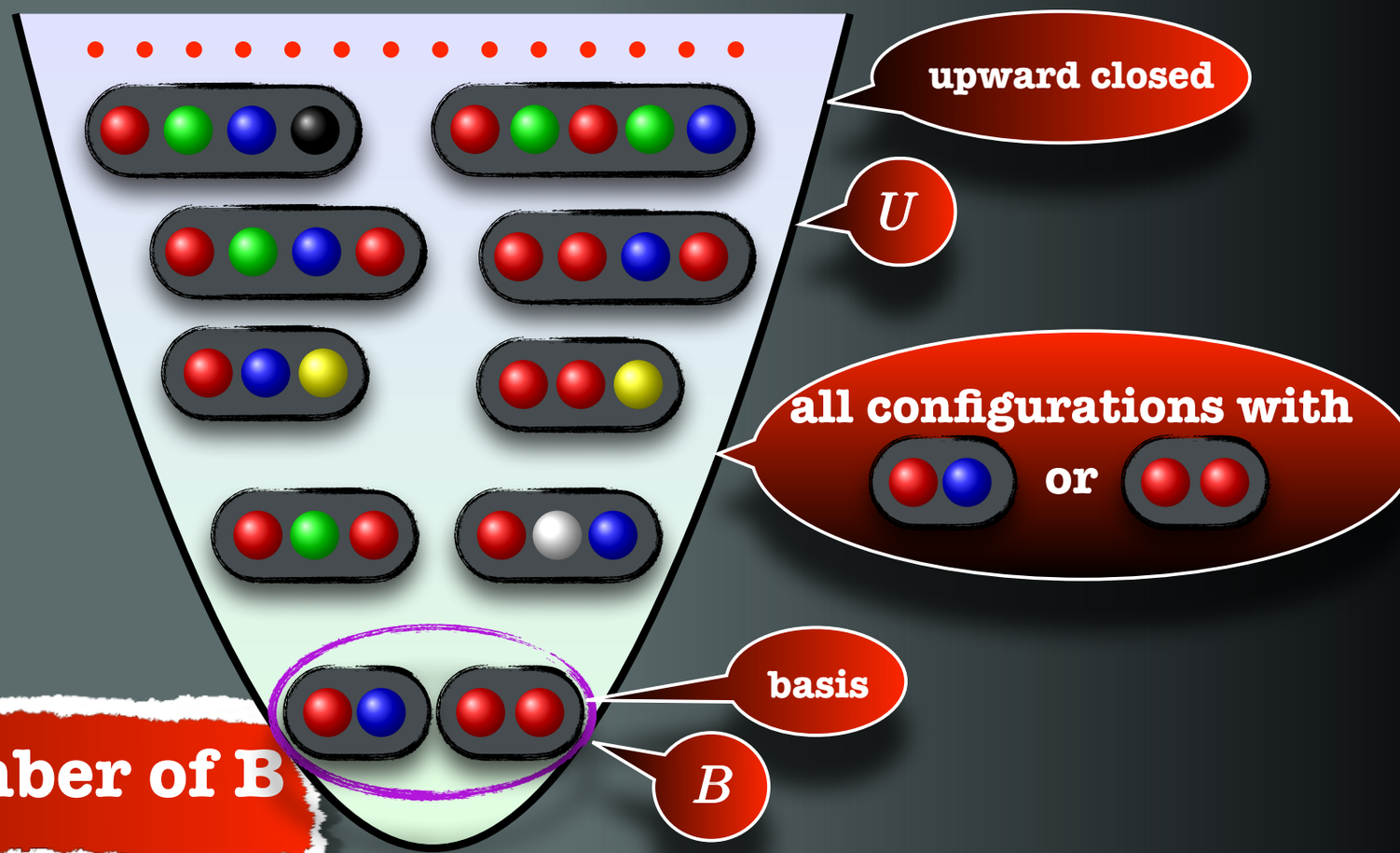
D

All pairs but

and



$\neg D = B$



$\#D =$ size of largest member of B

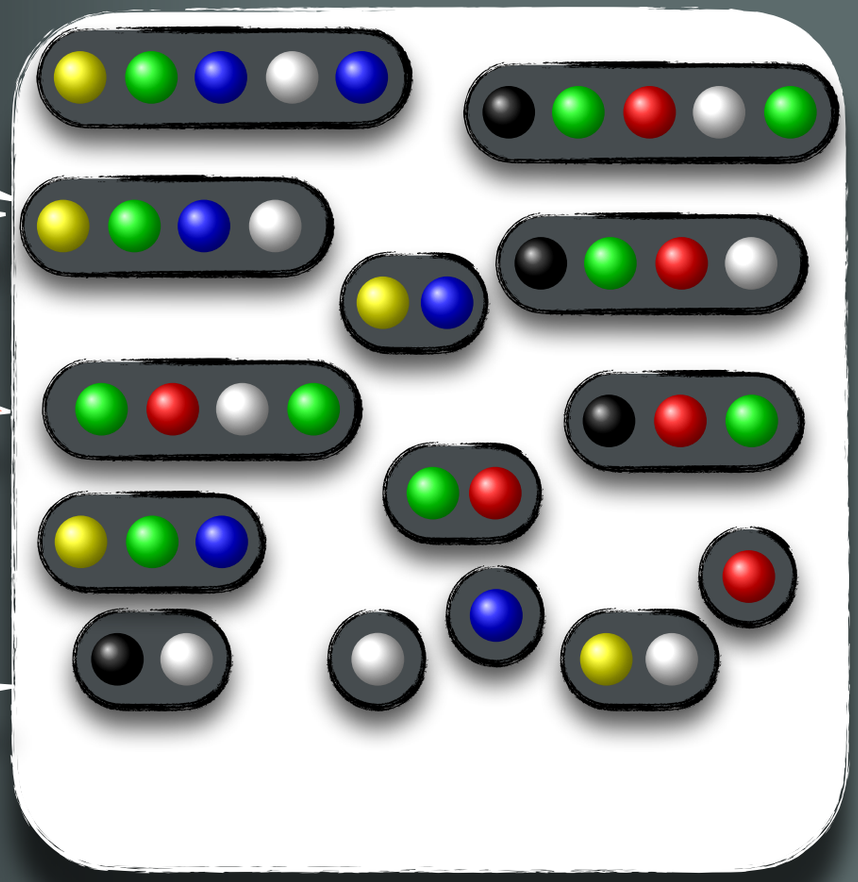
$\#D = 2$

downward closed

D

All pairs but

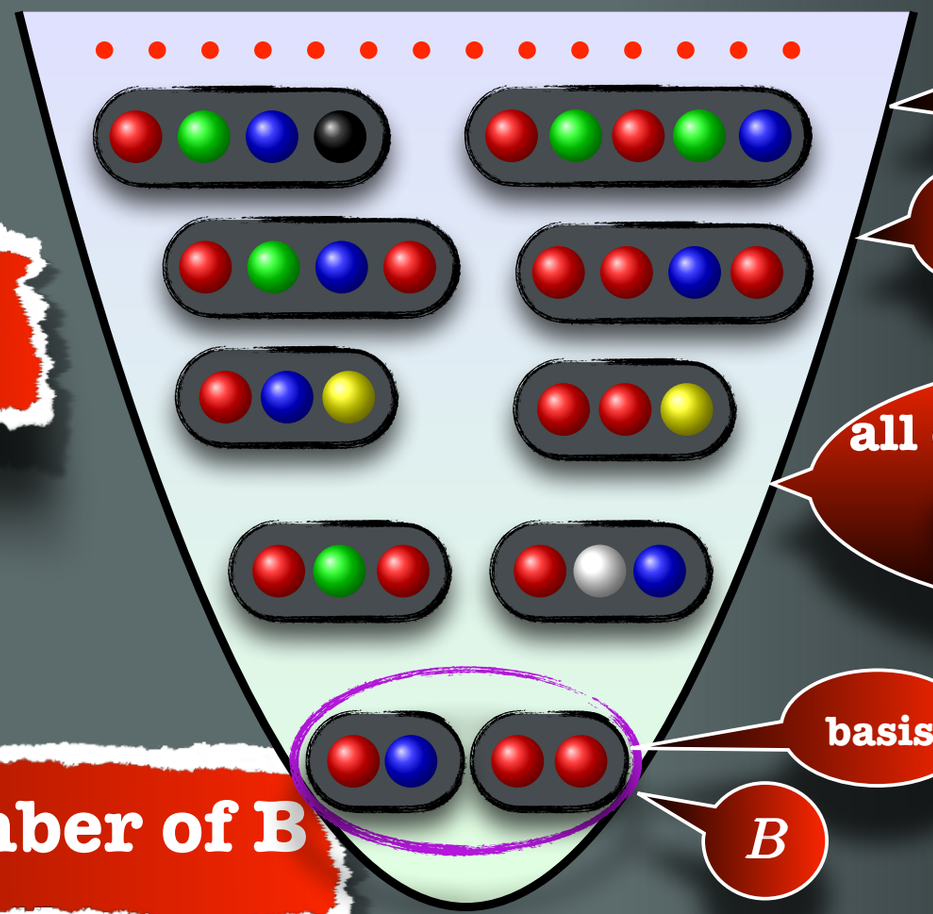
and



$\neg D = B$

$k = \#D$

$\gamma_k(\alpha_k(D) = D$



upward closed

U

all configurations with

or

basis

B

$\#D =$ size of largest member of B

$\#D = 2$

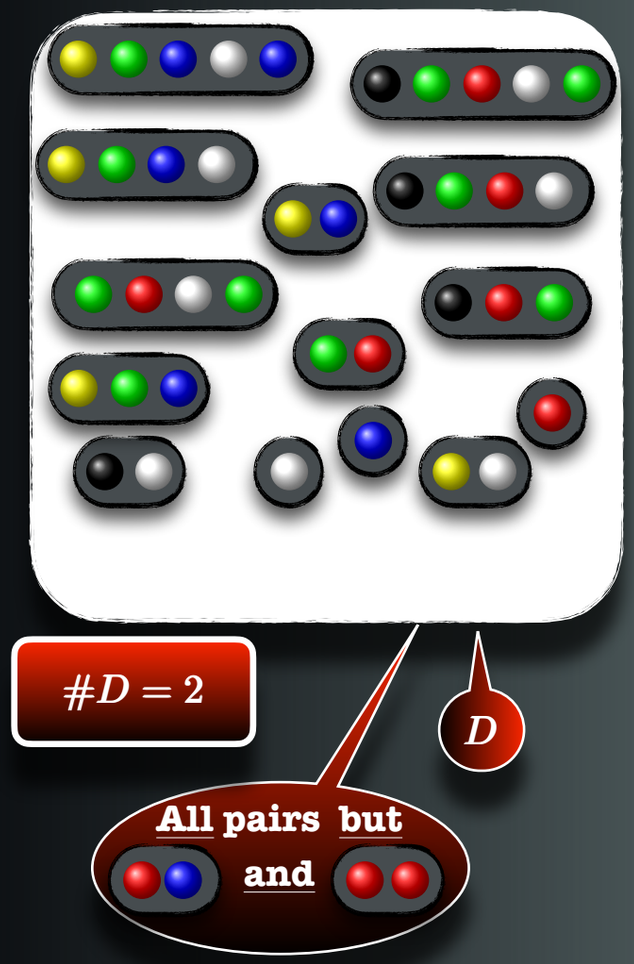
D

All pairs but

and

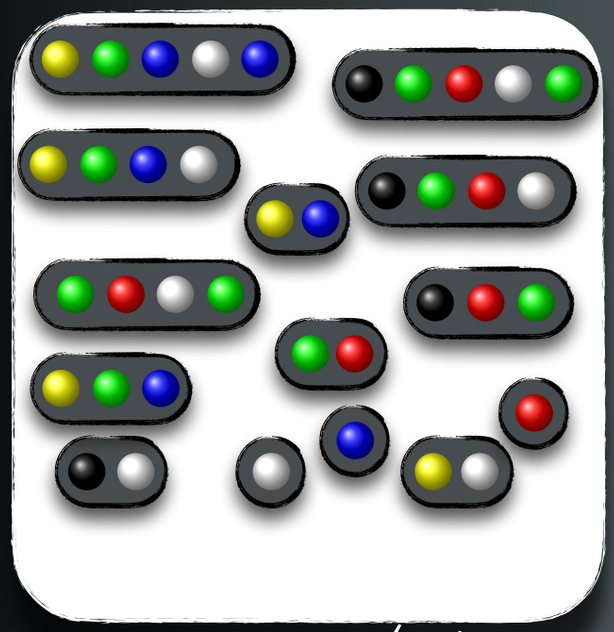
$k = \#D$

$\gamma_k(\alpha_k(D) = D$



$k = \#D$

$\gamma_k(\alpha_k(D) = D$



$\#D = 2$

D

All pairs but

and

α_2

$k = \#D$

$\gamma_k(\alpha_k(D) = D$

α_2

$\#D = 2$

D

All pairs but

and

All pairs but

and

$k = \#D$

$\gamma_k(\alpha_k(D) = D$

α_2

γ_2

$\#D = 2$

D

All pairs but

and

All pairs but

and

$k = \#D$

$\gamma_k(\alpha_k(D) = D$

1

$$\mathbb{R} \subseteq \gamma_k(V_k)$$

2

$$\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$$

$k = \#D$

$$\gamma_k(\alpha_k(D) = D$$

if

- D **is downward closed**
- D **is inductive**

then

- $\exists k.\ D = \gamma_k(V_k)$

1

$$\mathbb{R} \subseteq \gamma_k(V_k)$$

2

$$\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$$

$k = \#D$

$$\gamma_k(\alpha_k(D) = D$$

if

- D **is downward closed**
- D **is inductive**

then

- $\exists k.\ D = \gamma_k(V_k)$

1

$$\mathbb{R} \subseteq \gamma_k(V_k)$$

2

$$\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$$

$k = \#D$

$\gamma_k(\alpha_k(D) = D$

B

$\mathbb{R}$

if

- D **is downward closed**
- D **is inductive**

then

- $\exists k.\ D = \gamma_k(V_k)$

1

$$\mathbb{R} \subseteq \gamma_k(V_k)$$

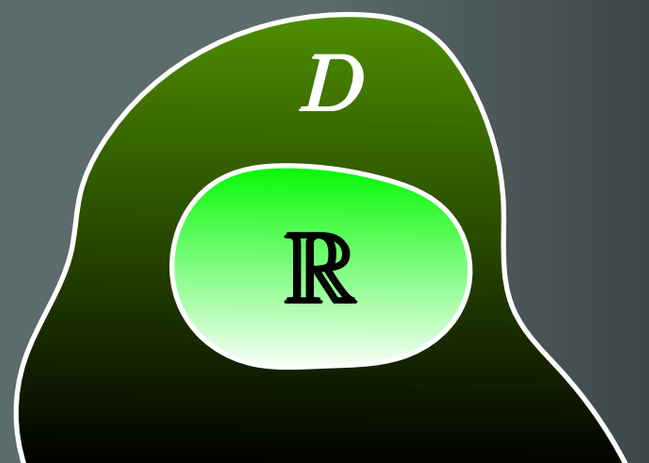
2

$$\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$$

$k = \#D$

$\gamma_k(\alpha_k(D) = D$

B



if

- D **is downward closed**
- D **is inductive**

then

- $\exists k.\ D = \gamma_k(V_k)$

1

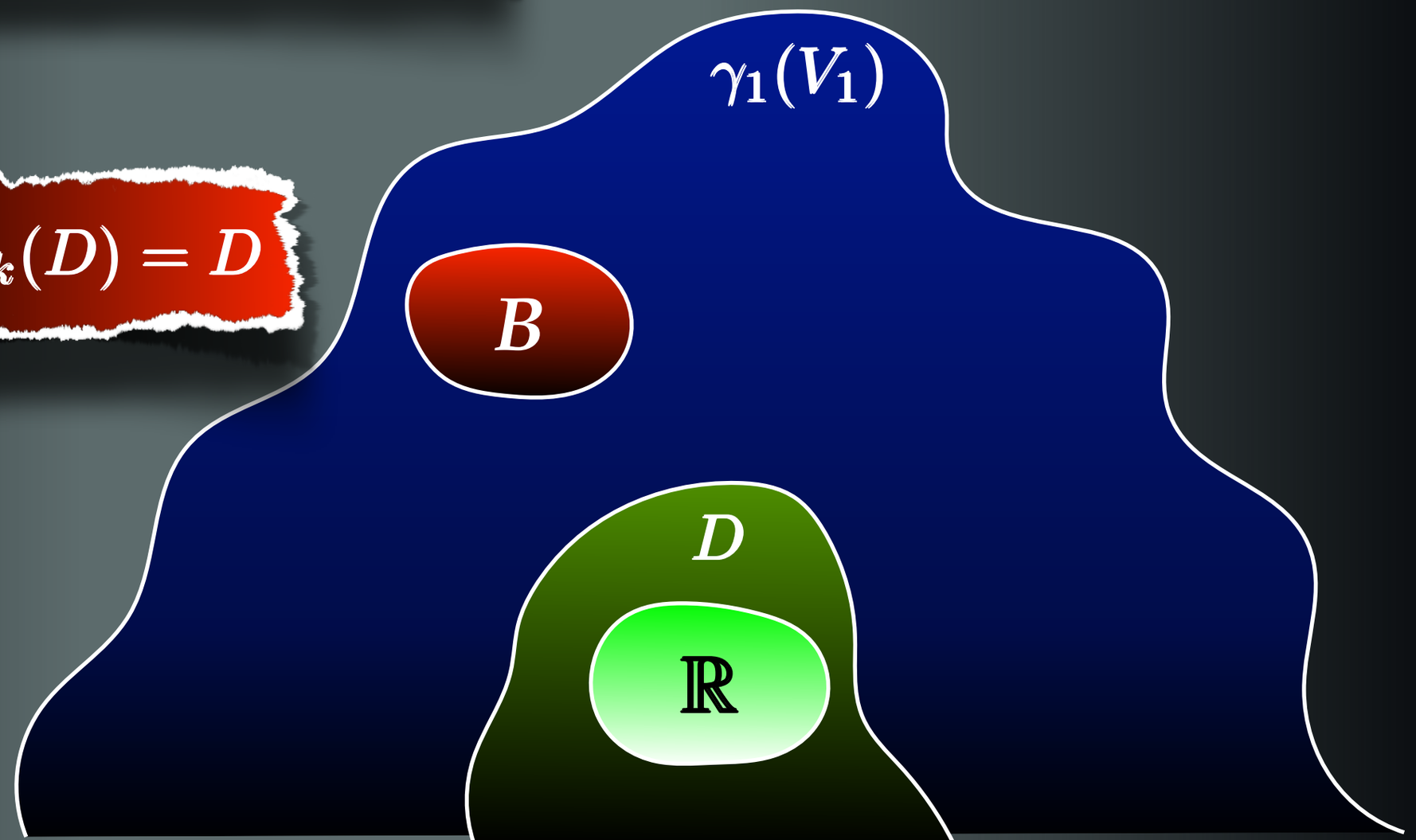
$$\mathbb{R} \subseteq \gamma_k(V_k)$$

2

$$\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$$

$k = \#D$

$\gamma_k(\alpha_k(D) = D$



if

- D **is downward closed**
- D **is inductive**

then

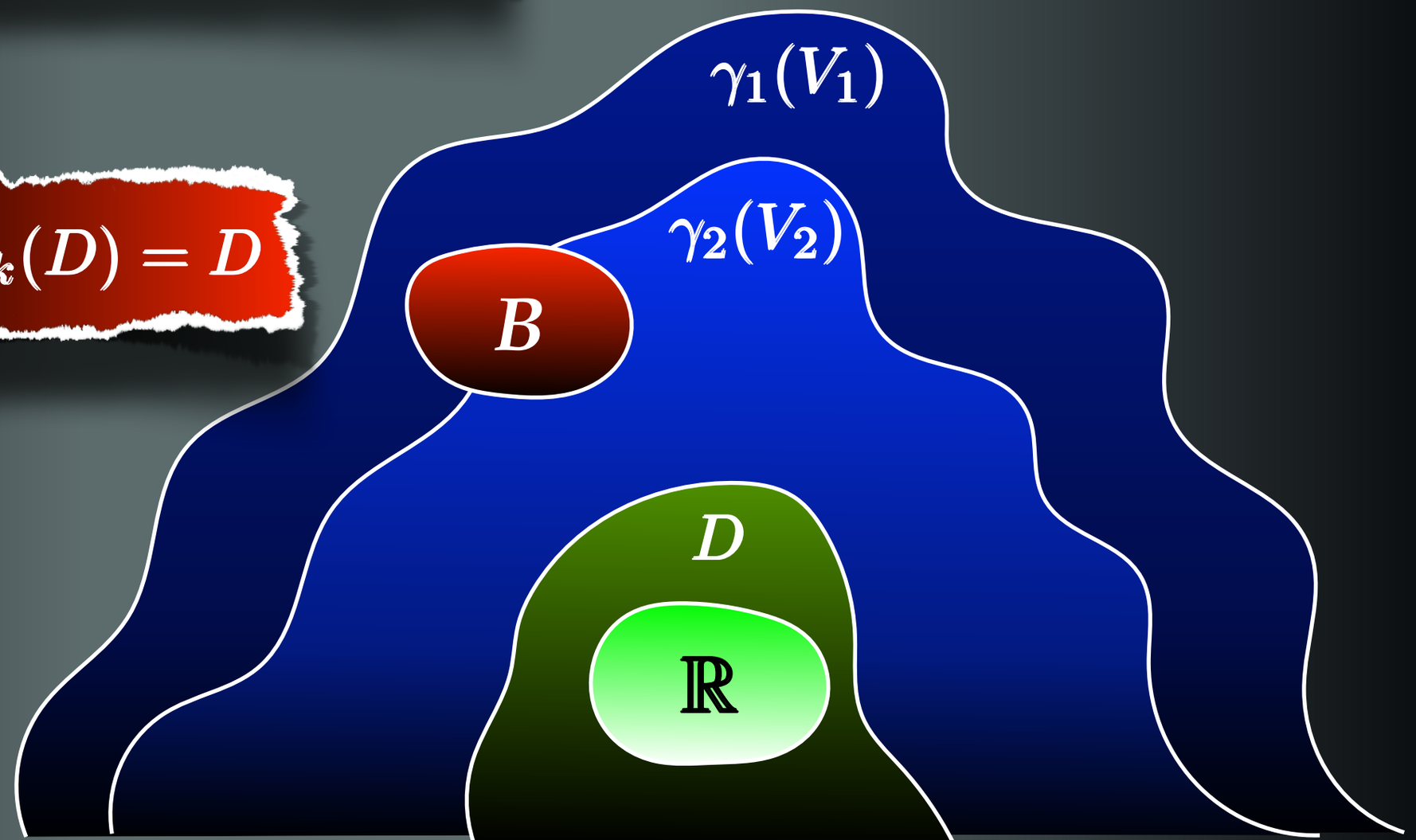
- $\exists k.\ D = \gamma_k(V_k)$

1 $\mathbb{R} \subseteq \gamma_k(V_k)$

2 $\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$

$k = \#D$

$\gamma_k(\alpha_k(D) = D$



if

- D **is downward closed**
- D **is inductive**

then

- $\exists k.\ D = \gamma_k(V_k)$

1

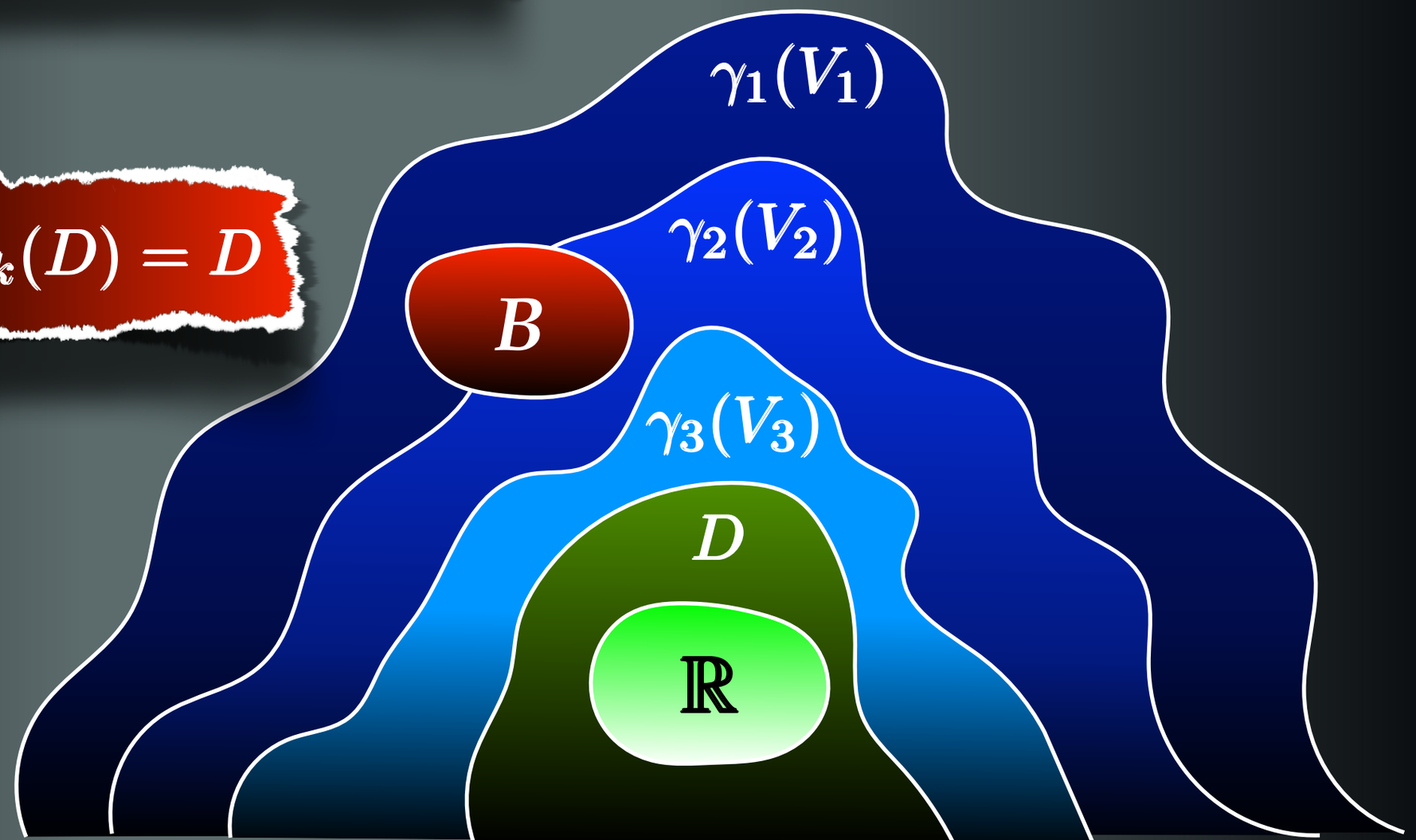
$$\mathbb{R} \subseteq \gamma_k(V_k)$$

2

$$\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$$

$k = \#D$

$\gamma_k(\alpha_k(D) = D$



if

- D **is downward closed**
- D **is inductive**

then

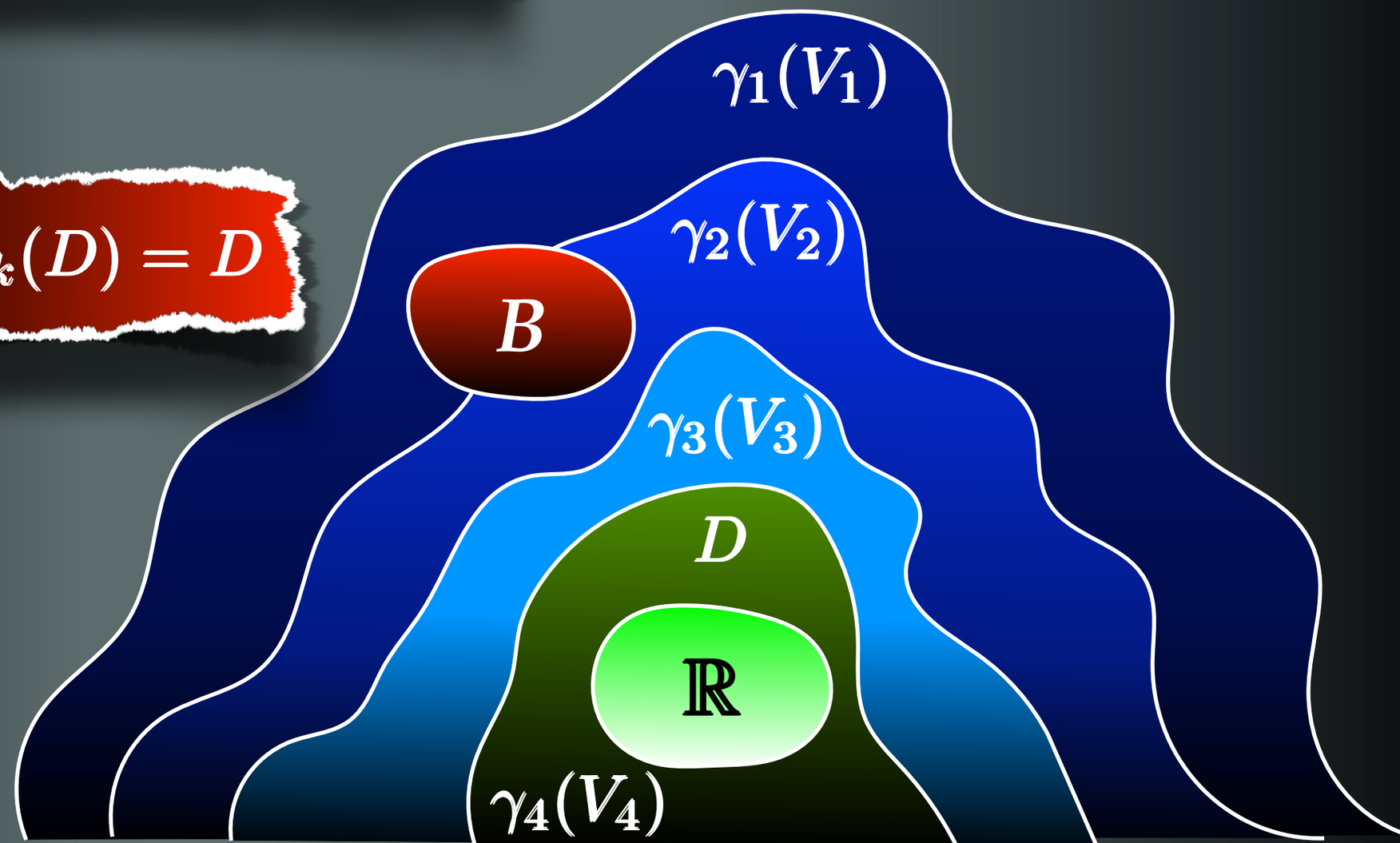
- $\exists k.\ D = \gamma_k(V_k)$

1 $\mathbb{R} \subseteq \gamma_k(V_k)$

2 $\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$

$k = \#D$

$\gamma_k(\alpha_k(D) = D$



if

- D **is downward closed**
- D **is inductive**

then

- $\exists k.\ D = \gamma_k(V_k)$

1

$$\mathbb{R} \subseteq \gamma_k(V_k)$$

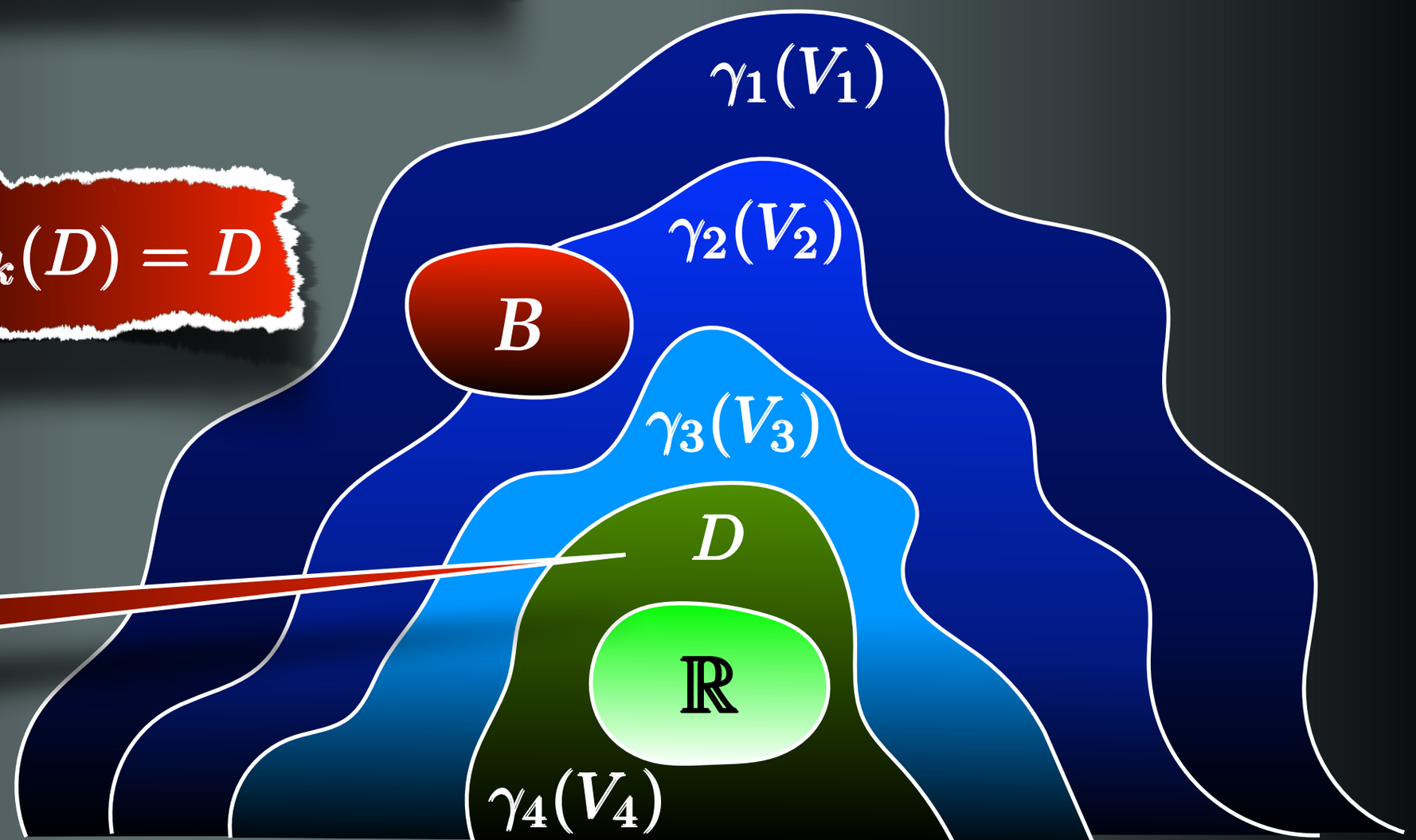
2

$$\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$$

$k = \#D$

$\gamma_k(\alpha_k(D) = D$

cut-off point
$k{=}4$



if

- D **is downward closed**
- D **is inductive**
- $D \cap B = \emptyset$

then termination guaranteed

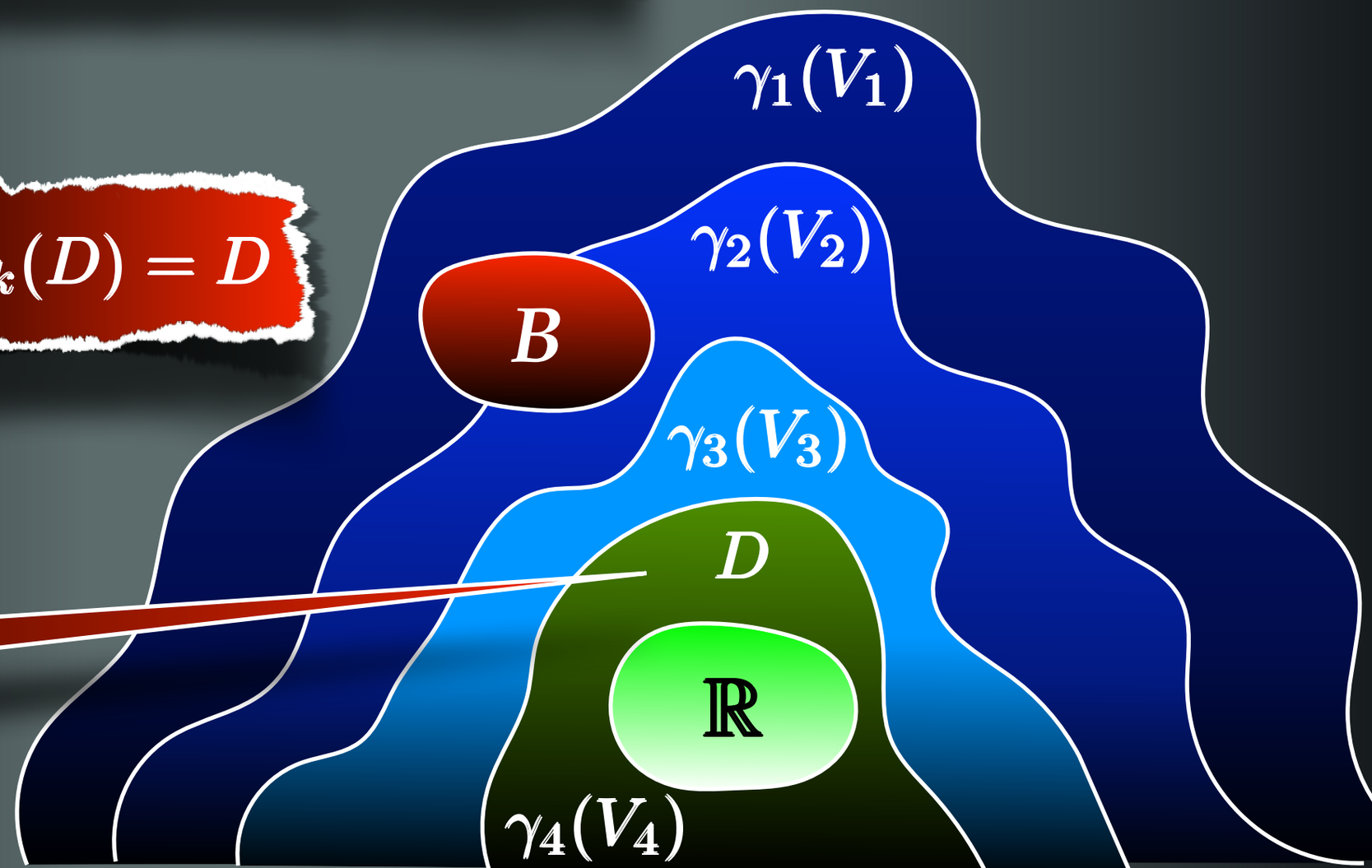
1 $\mathbb{R} \subseteq \gamma_k(V_k)$

2 $\gamma_{k+1}(V_{k+1}) \subseteq \gamma_k(V_k)$

$k = \#D$

$\gamma_k(\alpha_k(D) = D$

cut-off point
k=4



c_1 ⟶ c_2

$c_1 \sqcap c_3$

Monotonicity

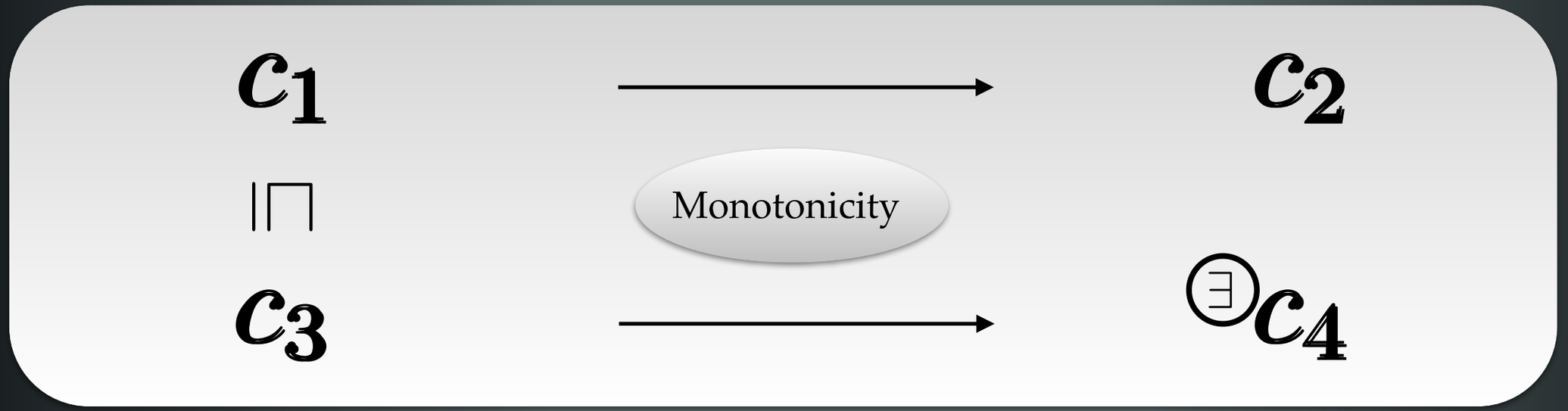
c_1 → c_2

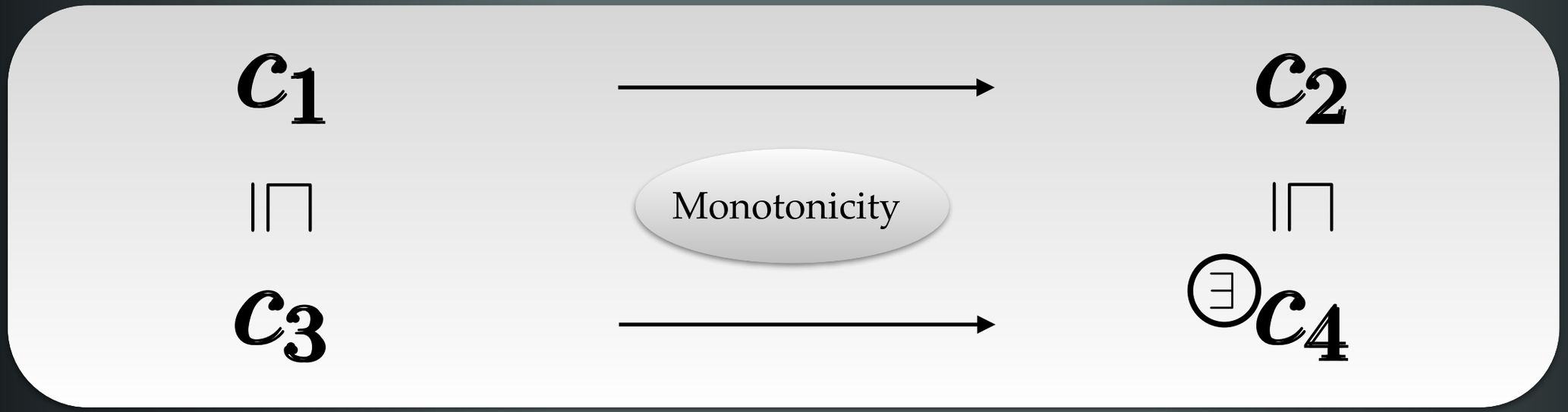
$\sqcap$

c_3

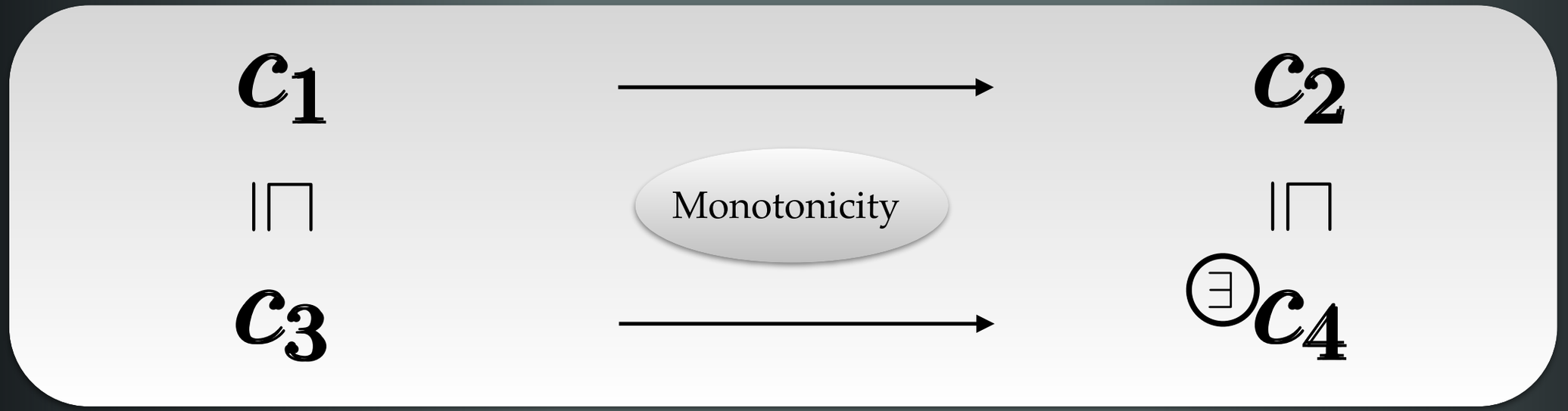
Monotonicity

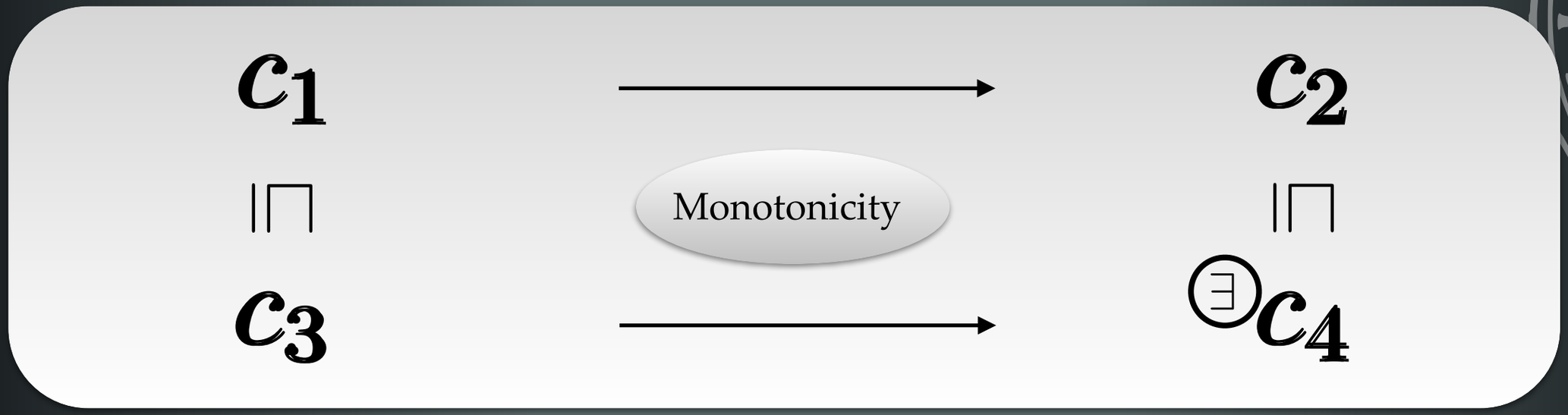
$\exists c_4$


c_1
c_2
$\sqsubseteq$
Monotonicity
c_3
$\exists c_4$


c_1
c_2
$\sqcap$
Monotonicity
$\sqcap$
c_3
$\exists c_4$

larger configuration
“simulate”
smaller configurations

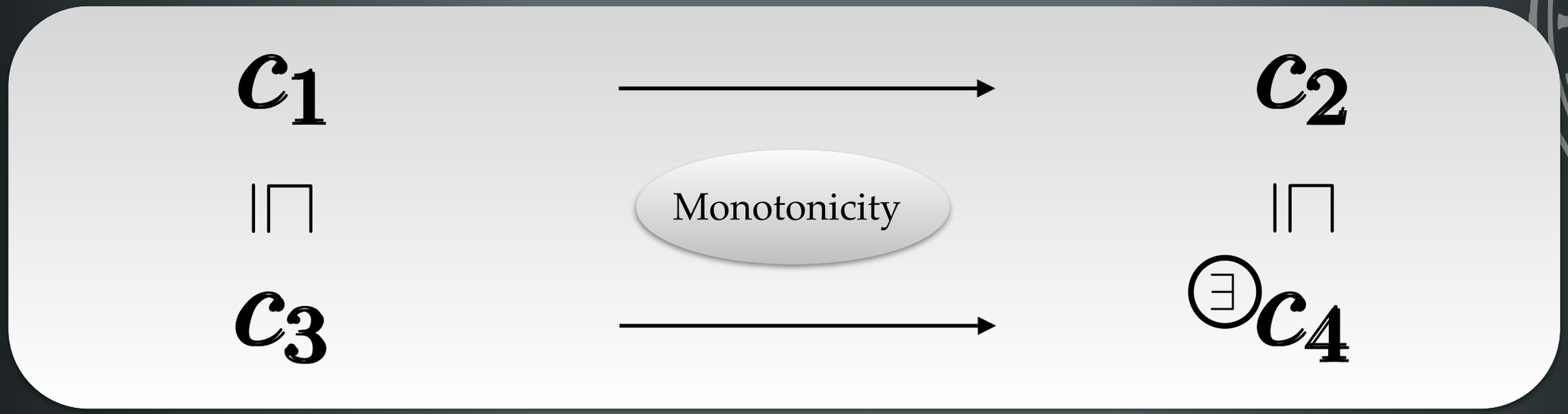




if

- D **is downward closed**
- D **is inductive**
- $D \cap B = \emptyset$

then termination guaranteed

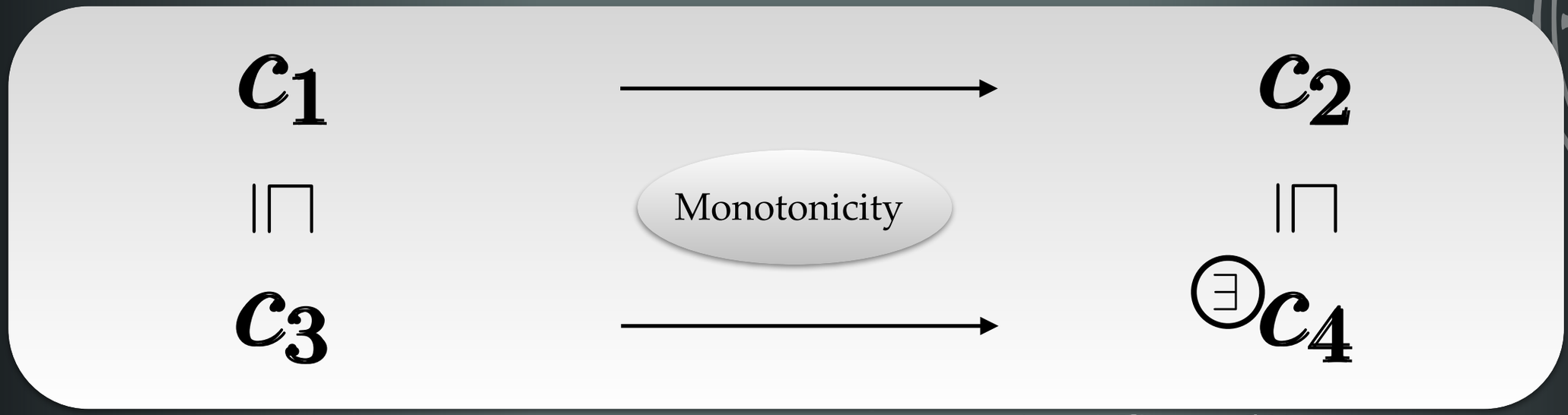


$$\mathbb{R}\downarrow \cap B = \emptyset \quad \textbf{iff} \quad \mathbb{R} \cap B = \emptyset$$

if

- D **is downward closed**
- D **is inductive**
- $D \cap B = \emptyset$

then termination guaranteed



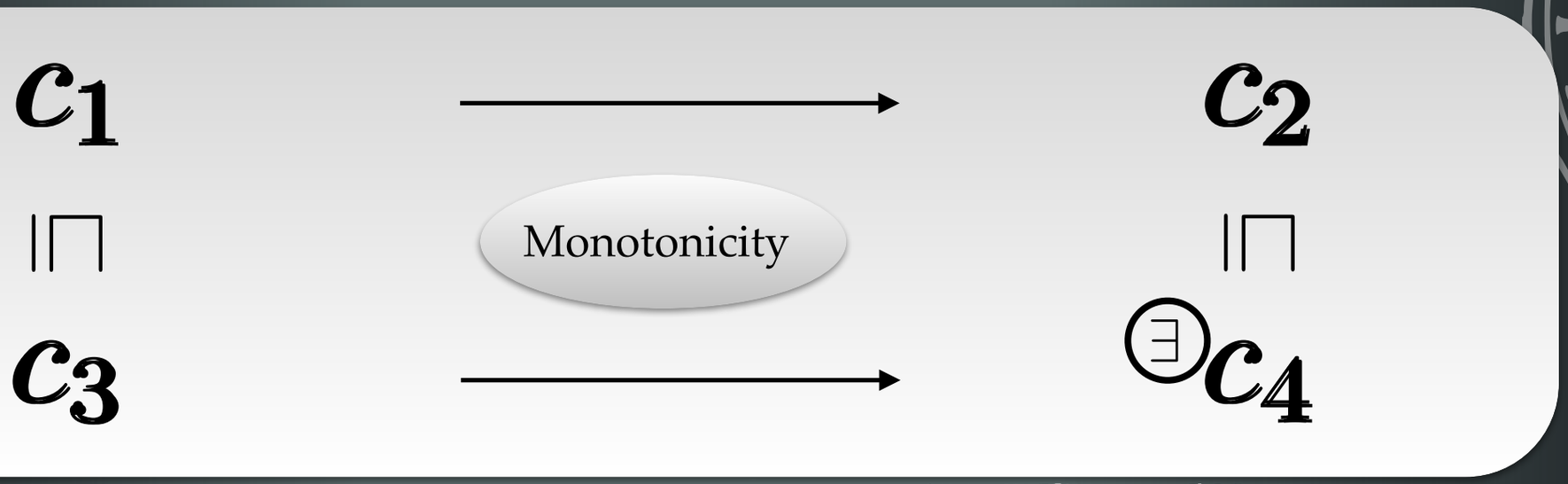
$$\mathbb{R}\downarrow \cap B = \emptyset \quad \textbf{iff} \quad \mathbb{R} \cap B = \emptyset$$

$\mathbb{R}\downarrow$ is inductive

if

- **D is downward closed**
- **D is inductive**
- $D \cap B = \emptyset$

then termination guaranteed



$$\mathbb{R}\downarrow \cap B = \emptyset \quad \textbf{iff} \quad \mathbb{R} \cap B = \emptyset$$

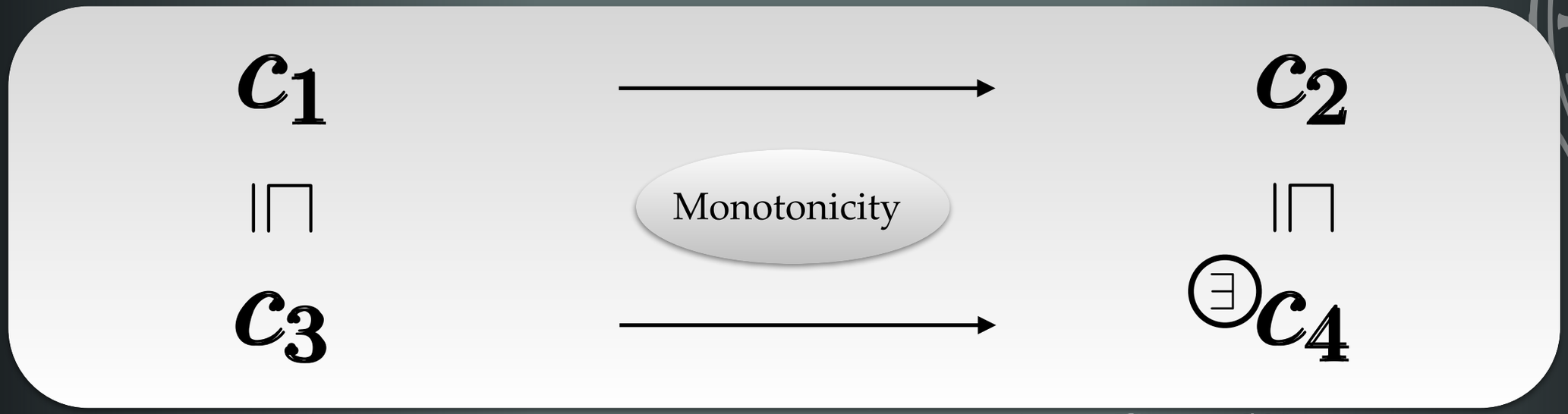
$\mathbb{R}\downarrow$ is inductive

if

- **D is downward closed**
- **D is inductive**
- $D \cap B = \emptyset$

then termination guaranteed

- **if system safe**
- **then termination guaranteed**



$$\mathbb{R}\downarrow \cap B = \emptyset \quad \textbf{iff} \quad \mathbb{R} \cap B = \emptyset$$

$\mathbb{R}$

Method complete:
Petri nets, lossy channels systems, reset nets, transfer nets, broadcast nets, etc

if

- D **is downwa**
- D **is inductiv**
- $D \cap B = \emptyset$

then termination guaranteed

- **if system safe**
- **then termination guaranteed**

Parameterized System ✓

Analysis ✓
Abstraction

Features ✓

Completeness

Parameterized System ✓

Analysis / Abstraction ✓

Features ✓

Completeness ✓

https://github.com/Nehlin/uu_tool_1.git

	Protocol	Time	k	$\|V\|$	$\gamma_{k+\ell}(V)$
Array	Demo (toy example)	0.01s	2	17	53
	Burns	0.01s	2	34	186
	Dijkstra	0.07s	2	93	695
	Szymanski	0.02s	2	48	264
Multiset	MOSI Coherency	0.01s	1	10	23
	German's Coherency	15.3s	6	1890	15567
Petri Net	German (simplified)	0.03s	2	43	96
	BH250	2.85s	2	503	503
	MOESI Coherency	0.01s	1	13	20
	Critical Section	0.01s	5	27	46
	Kanban	?	≥ 20	?	?
Tree	Percolate	0.05s	2	34	933
	Tree Arbiter	0.7s	2	88	7680
	Leader Election	0.1s	2	74	362
Ring	Token Passing	0.01s	2	2	2

https://github.com/Nehlin/uu_tool_1.git

	Protocol	Time	k	$\lvert V\rvert$	$\gamma_{k+\ell}(V)$
Array	Demo (toy example)	0.01s	2	17	53
	Burns	0.01s	2	34	186
	Dijkstra	0.07s	2	93	695
	Szymanski	0.02s	2	48	264
Multiset	MOSI Coherency	0.01s	1	10	23
	German's Coherency	15.3s	6	1890	15567
Petri Net	German (simplified)	0.03s	2	43	96
	BH250	2.85s	2	503	503
	MOESI Coherency	0.01s	1	13	20
	Critical Section	0.01s	5	27	46
	Kanban	?	≥ 20	?	?
Tree	Percolate	0.05s	2	34	933
	Tree Arbiter	0.7s	2	88	7680
	Leader Election	0.1s	2	74	362
Ring	Token Passing	0.01s	2	2	2

https://github.com/Nehlin/uu_tool_1.git

	Protocol	Time	k	$\|V\|$	$\gamma_{k+\ell}(V)$
Array	Demo (toy example)	0.01s	2	17	53
	Burns	0.01s	2	34	186
	Dijkstra	0.07s	2	93	695
	Szymanski	0.02s	2	48	264
Multiset	MOSI Coherency	0.01s	1	10	23
	German's Coherency	15.3s	6	1890	15567
Petri Net	German (simplified)	0.03s	2	43	96
	BH250	2.85s	2	503	503
	MOESI Coherency	0.01s	1	13	20
	Critical Section	0.01s	5	27	46
	Kanban	?	≥ 20	?	?
Tree	Percolate	0.05s	2	34	933
	Tree Arbiter	0.7s	2	88	7680
	Leader Election	0.1s	2	74	362
Ring	Token Passing	0.01s	2	2	2

Parameteri

System

Analysis

Abstractio

Feat

Completeness

Current Challenges

Parameteri
System

Analysis
Abstractio

Feat

Completeness

Current Challenges

Shape Analysis

Parameteri
Systeri

Analysis
Abstractio

Feat

Completeness

Current Challenges

Shape Analysis

Concurrent Libraries

Parameteri
System

Analysis
Abstractio

Feat

Completeness

Current Challenges

Shape Analysis

Concurrent Libraries

Unbounded Data
Structures

Parameteri
System

Analysis
Abstractio

Feat

Completeness

Current Challenges

Shape Analysis

Concurrent Libraries

Unbounded Data Structures

Weak Memory Models

Parameteri
System

Analysis
Abstractio

Feat

Completeness

Current Challenges

Shape Analysis

Concurrent Libraries

Unbounded Data Structures

Weak Memory Models

CEGAR


VERITAS
NATURAE