Cut-Offs in Parameterized Verification

Parosh Aziz Abdulla Frédéric Haziza Lukáš Holík

Parameterized System

Analysis / Abstraction

Features

Completeness

Parameterized System

Parameterized System

Remarks

- Parameterized System
  - unbounded number of components
  - Parameterized Verification: verify correctness regardless of number of components
- Motivation: ubiquitous
  - unbounded number of processes
  - unbounded data structures
  - unbounded number of variables

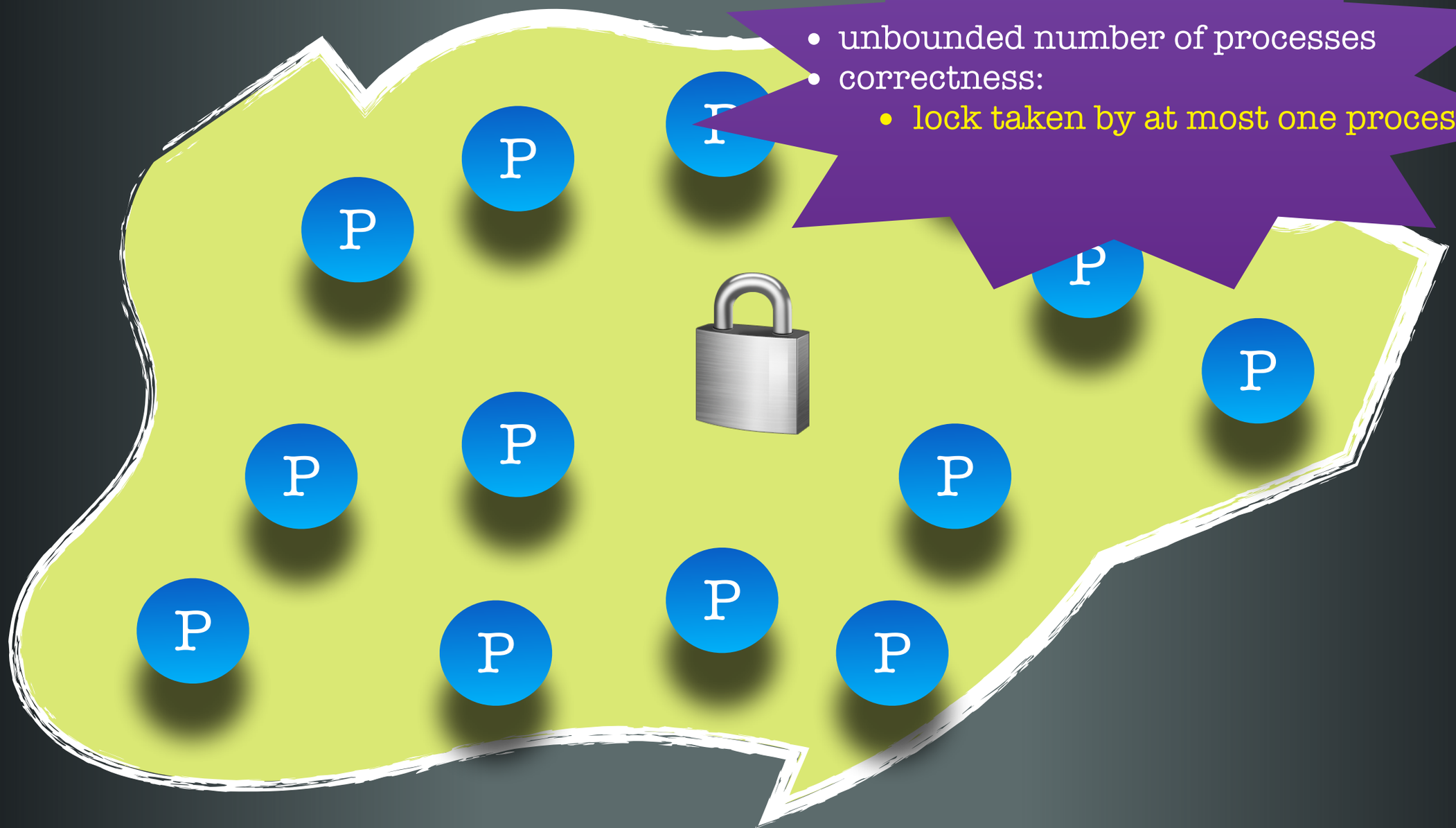
Parameterized System

Unbounded Number of Processes

Parameterized System

Unbounded Number of Processes

Mutual Exclusion Protocols



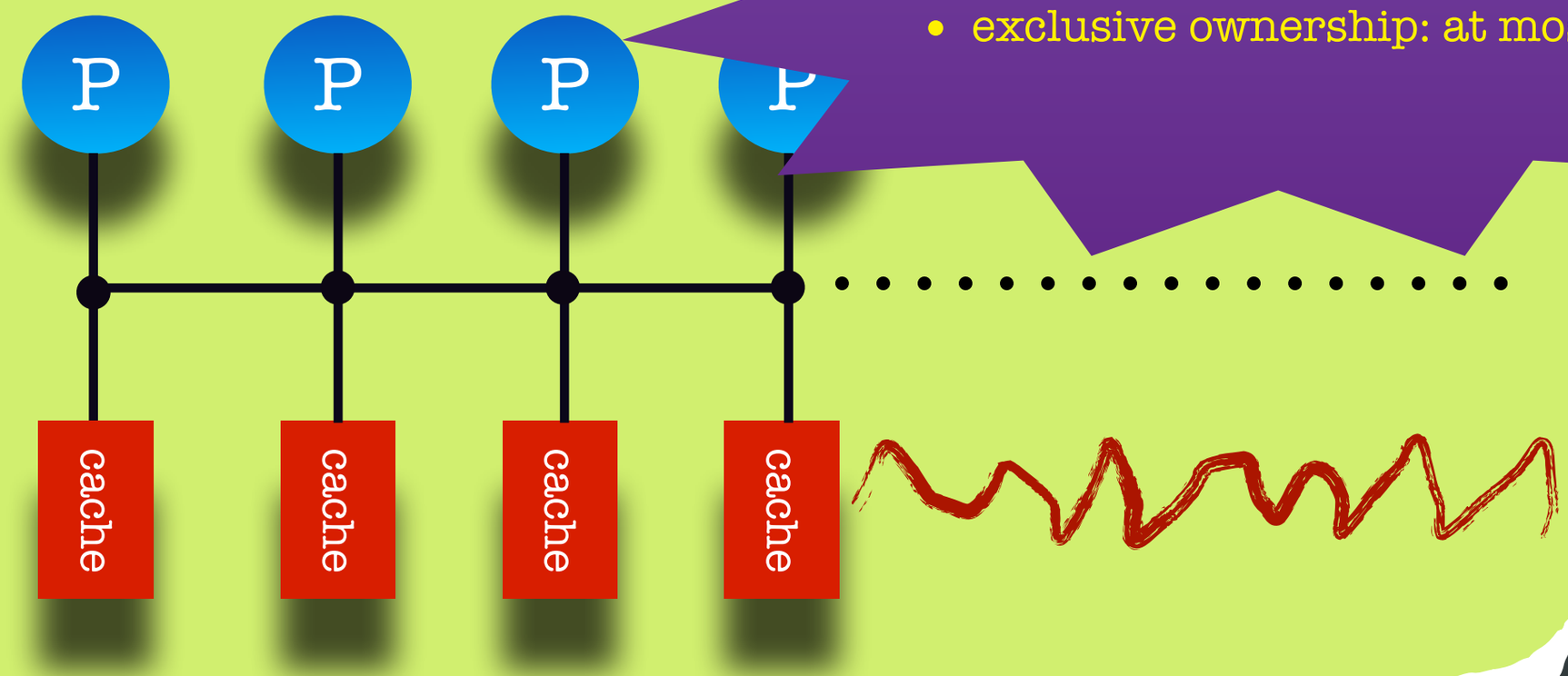
- unbounded number of processes
- correctness:
  - lock taken by at most one process

Parameterized System

Unbounded Number Processes

Cache Coherence Protocol

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



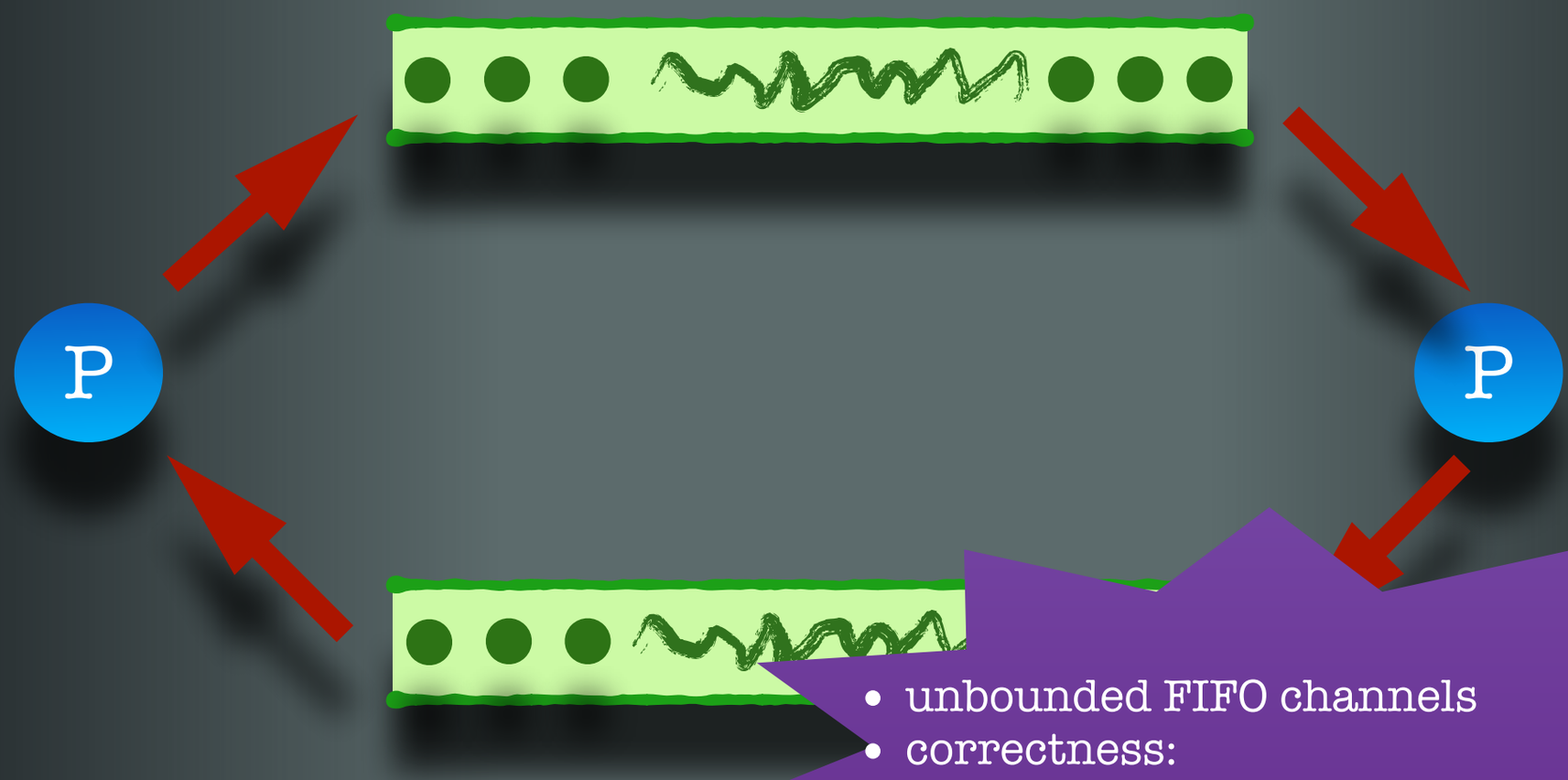
Parameterized System

Unbounded Data Structures

Parameterized System

Unbounded Data Structures

Unbounded Channels

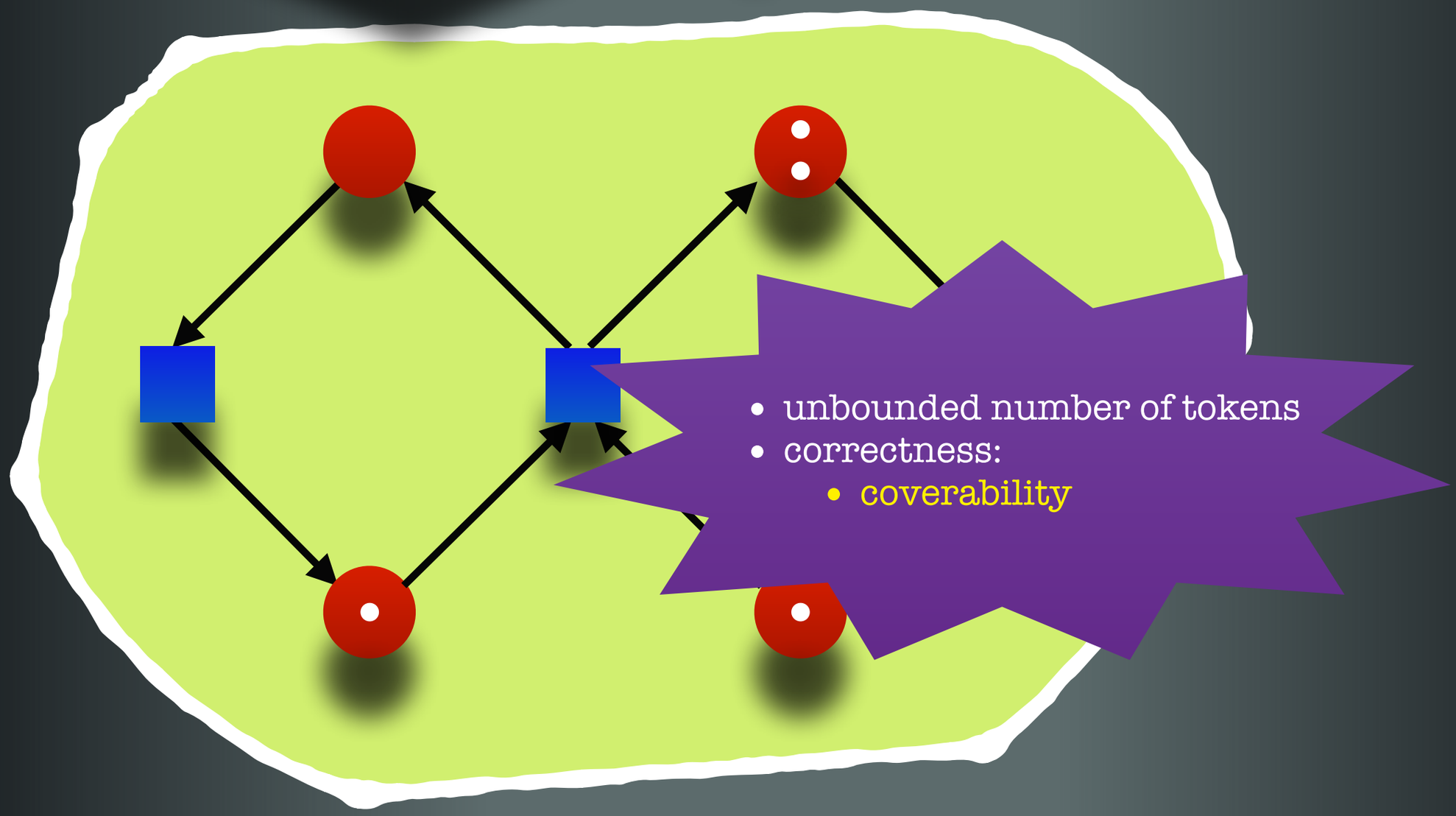


- unbounded FIFO channels
- correctness:
  - regardless of channels size

Parameterized System

Unbounded Data Structures

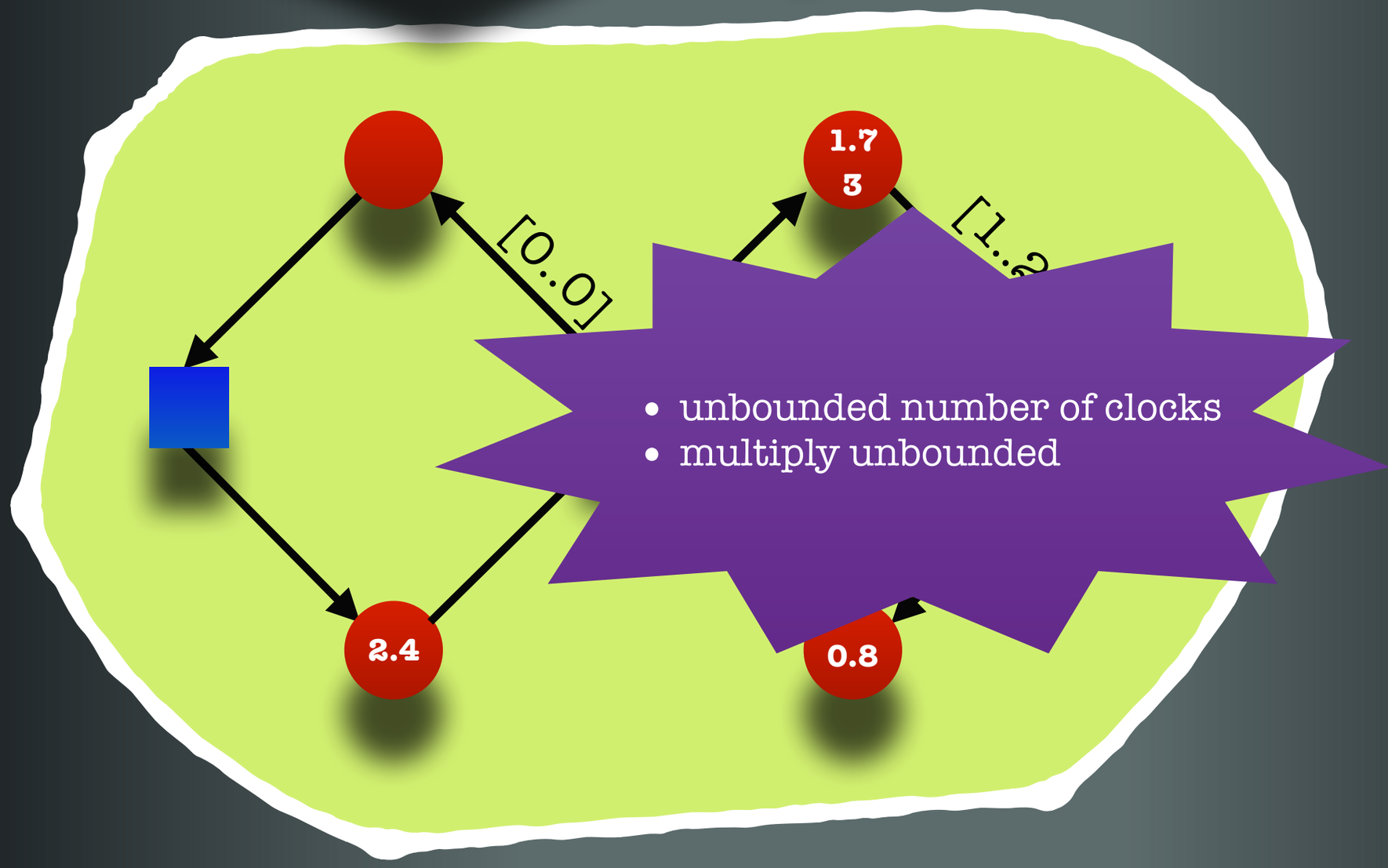
Petri Nets



Parameterized System

Unbounded Data Structures

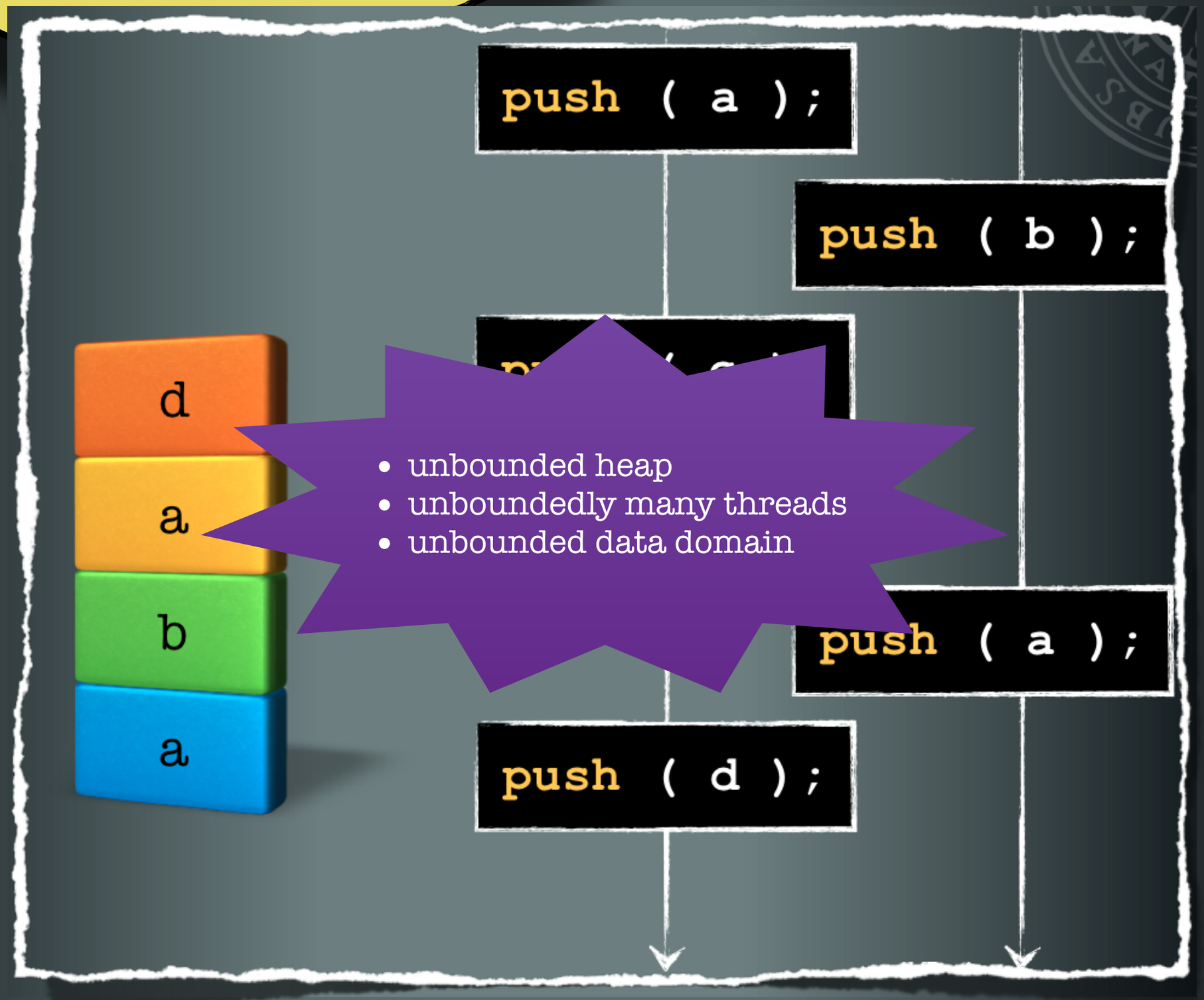
Petri Nets



- unbounded number of clocks
- multiply unbounded

Parameterized System

Unbounded Data Structures



- unbounded heap
- unboundedly many threads
- unbounded data domain

Parameter
Sy

Unbounded
Number of
Processes

Parameter Sy

Unbounded Number of Processes

Hierarchy

Processes:

- finite-state
- infinite-state
- dynamic

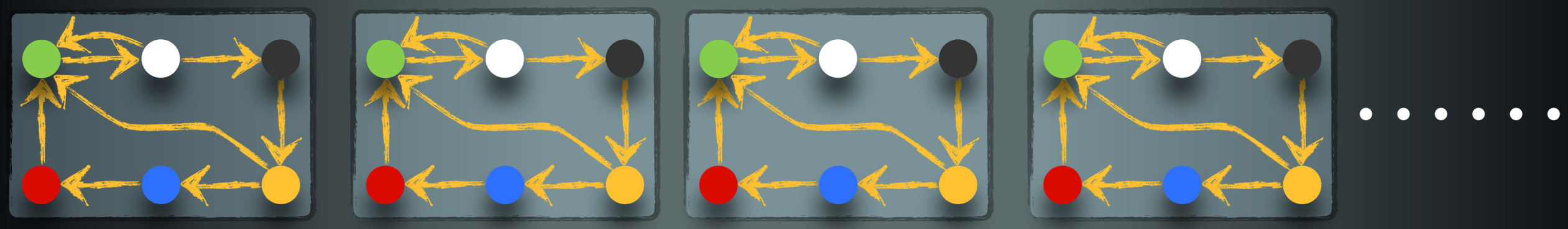
Topology:

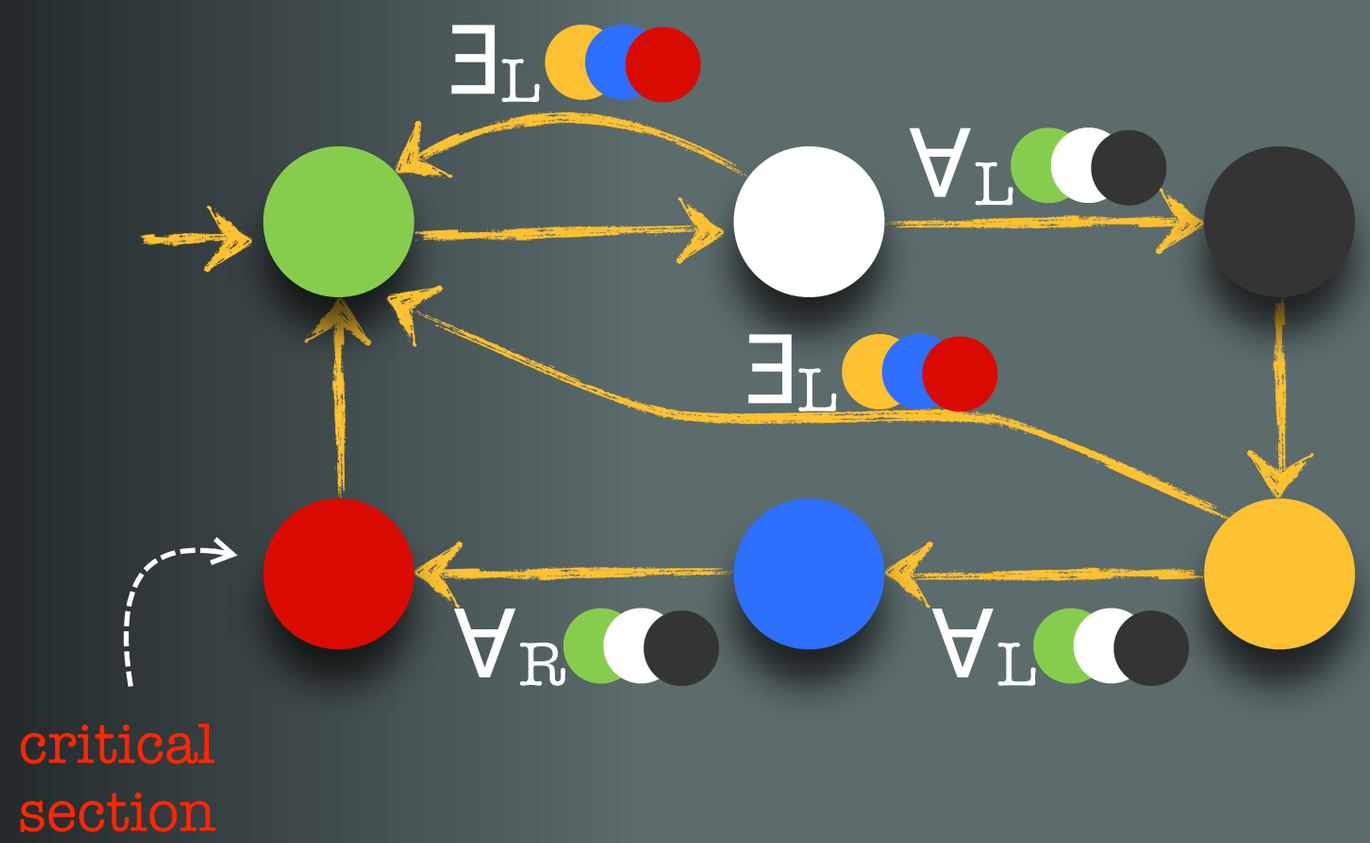
- array
- tree
- graph
- multiset

Communication:

- rendez-vous
- broadcast
- global

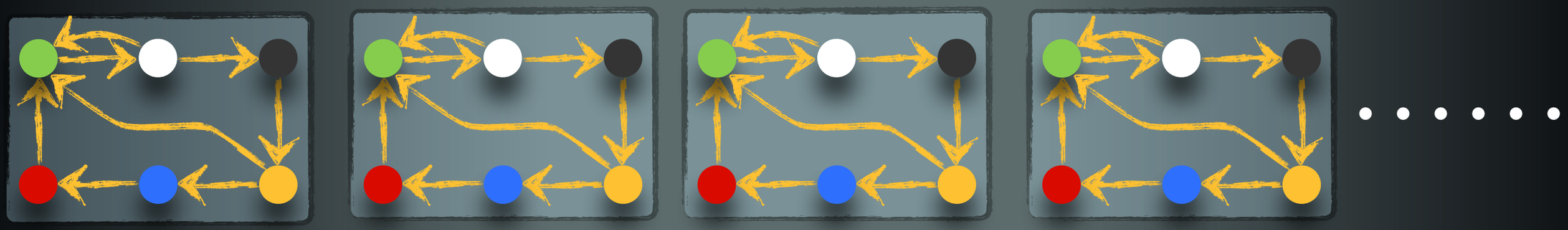
Parameterized System

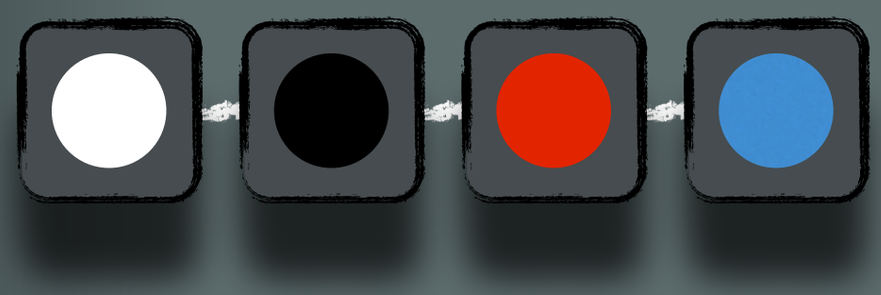




Burns' Mutual Exclusion Protocol

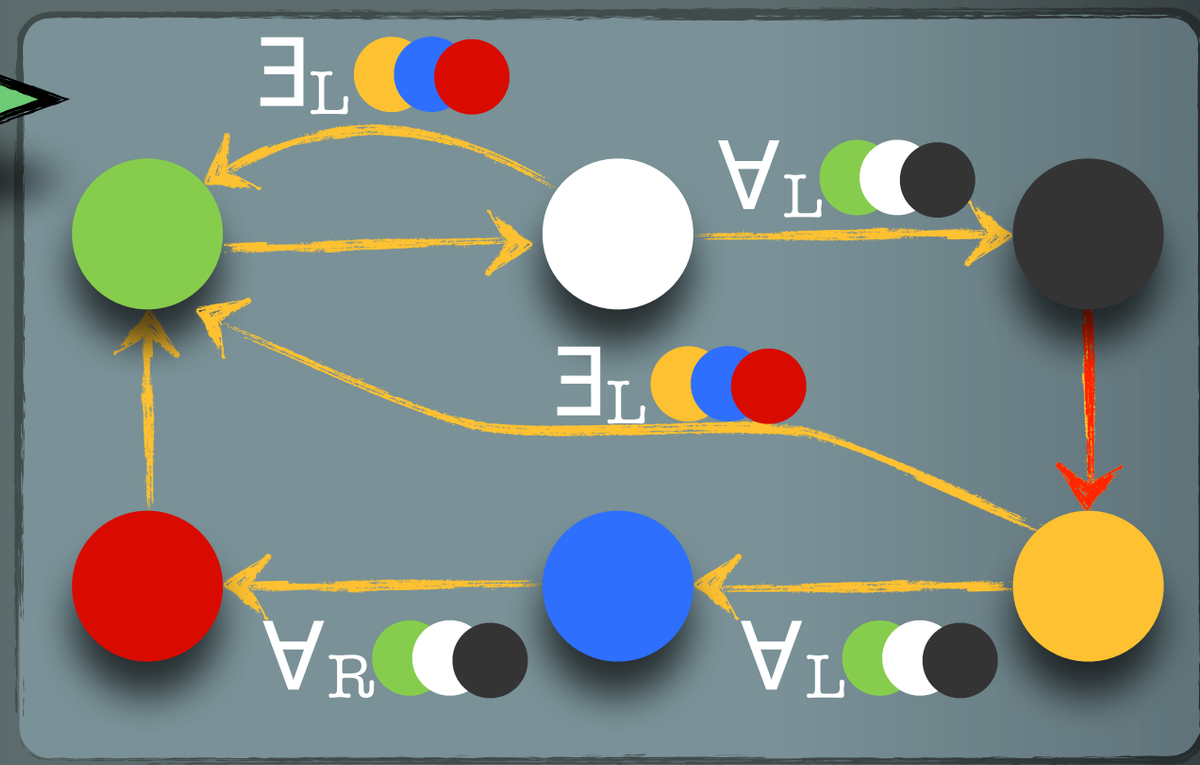
Parameterized System

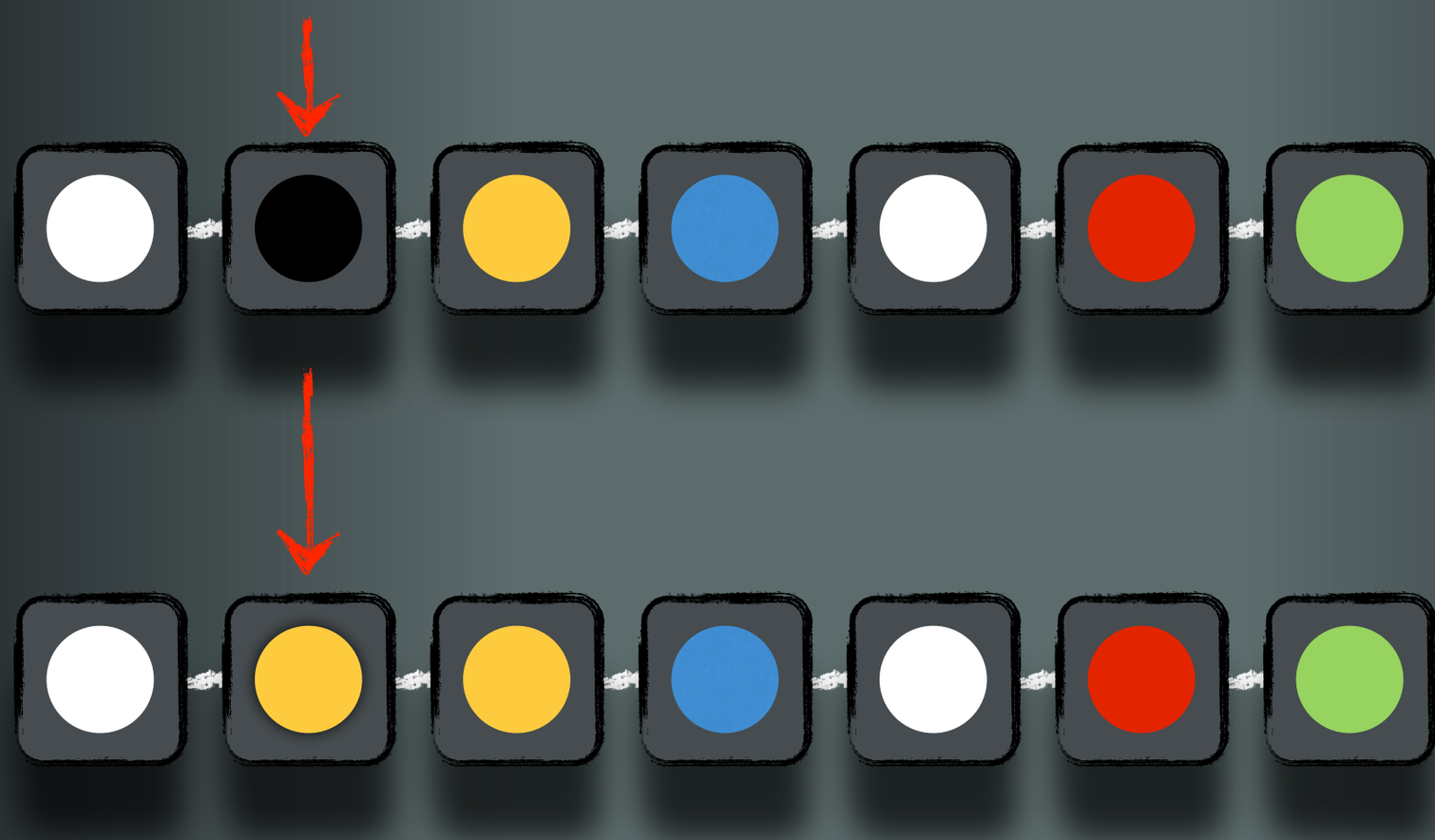




Configuration

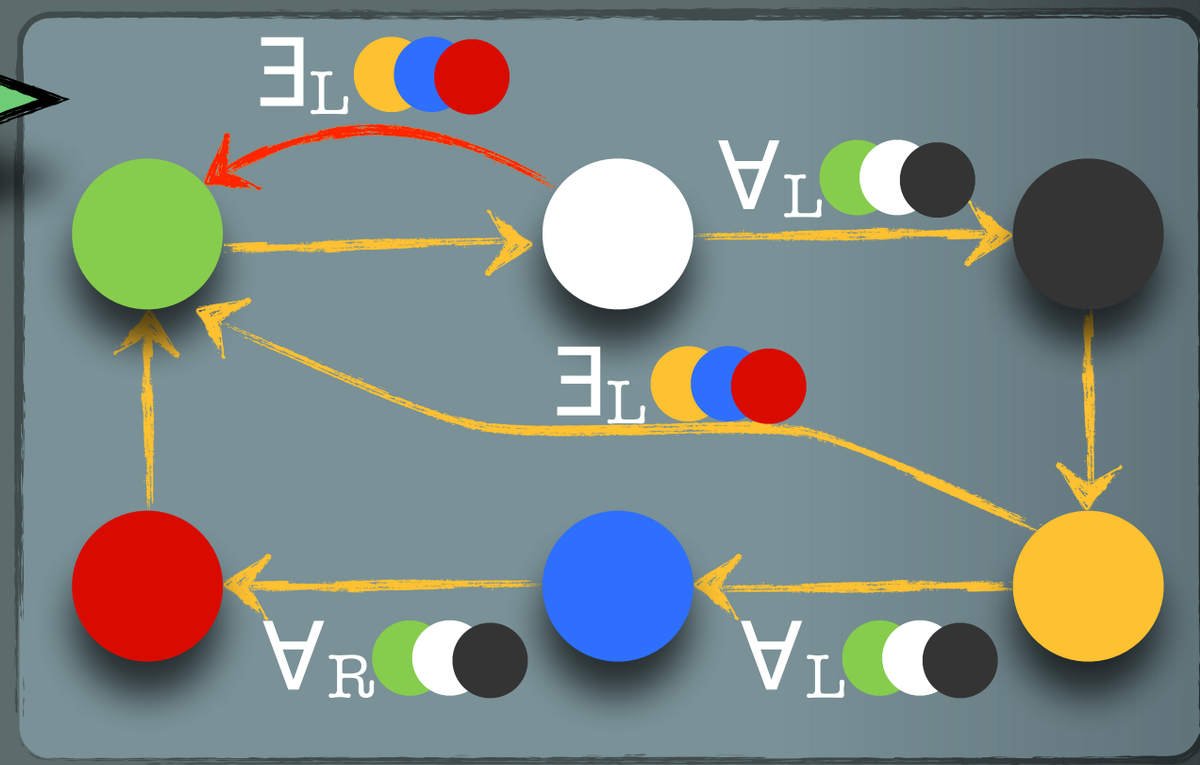
Parameterized System

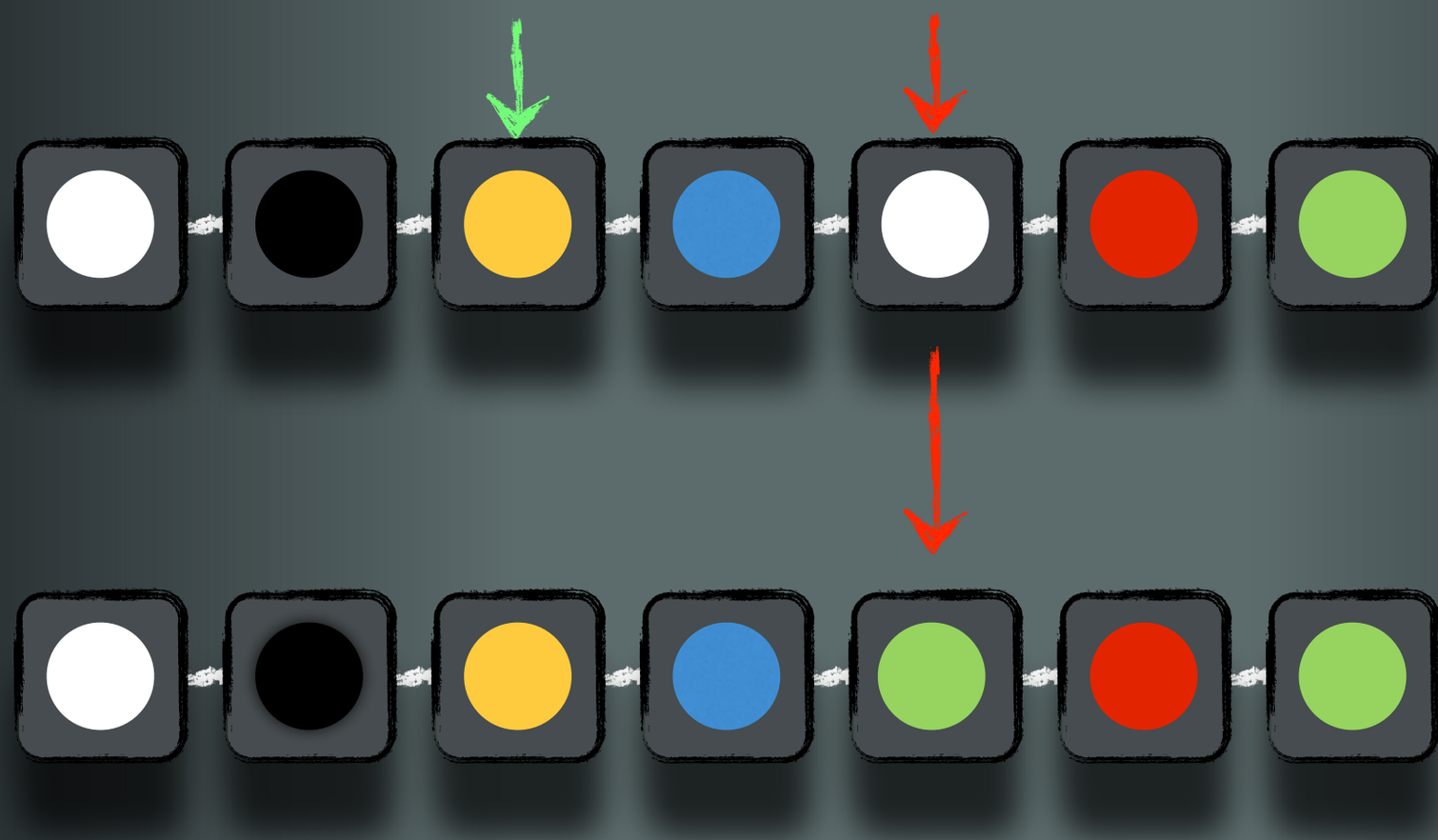




Local Transition

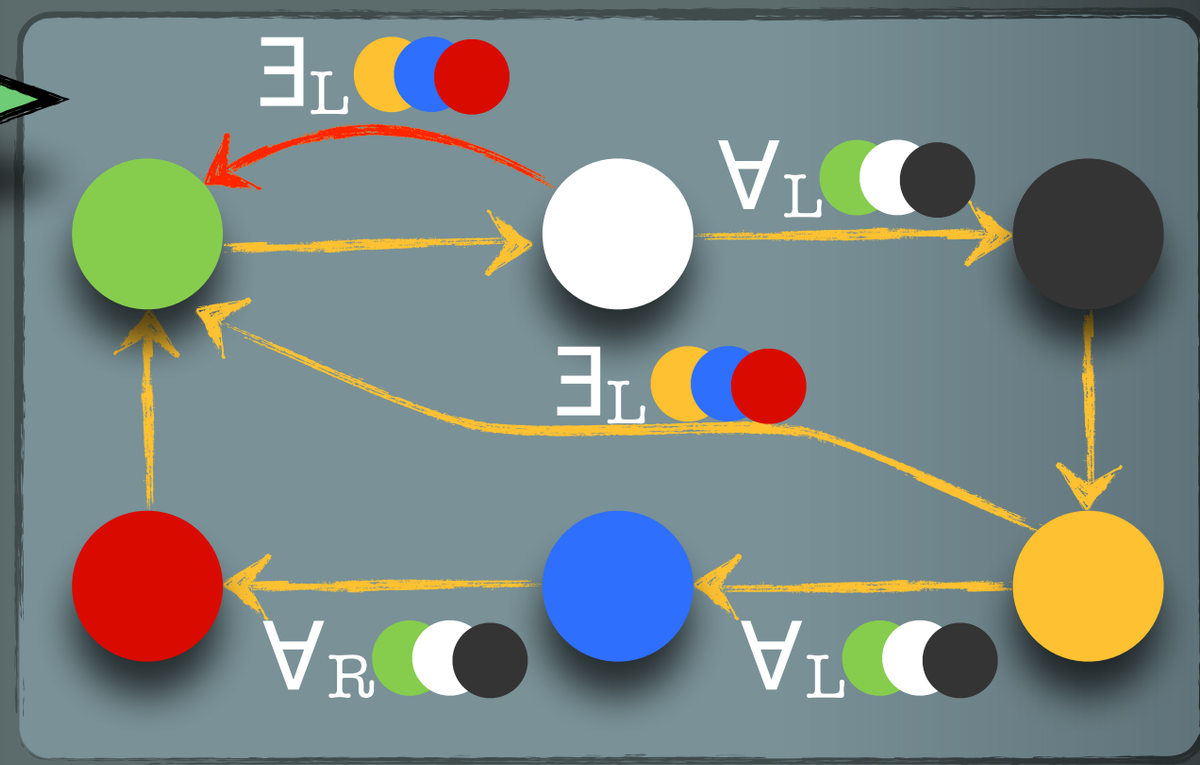
Parameterized System

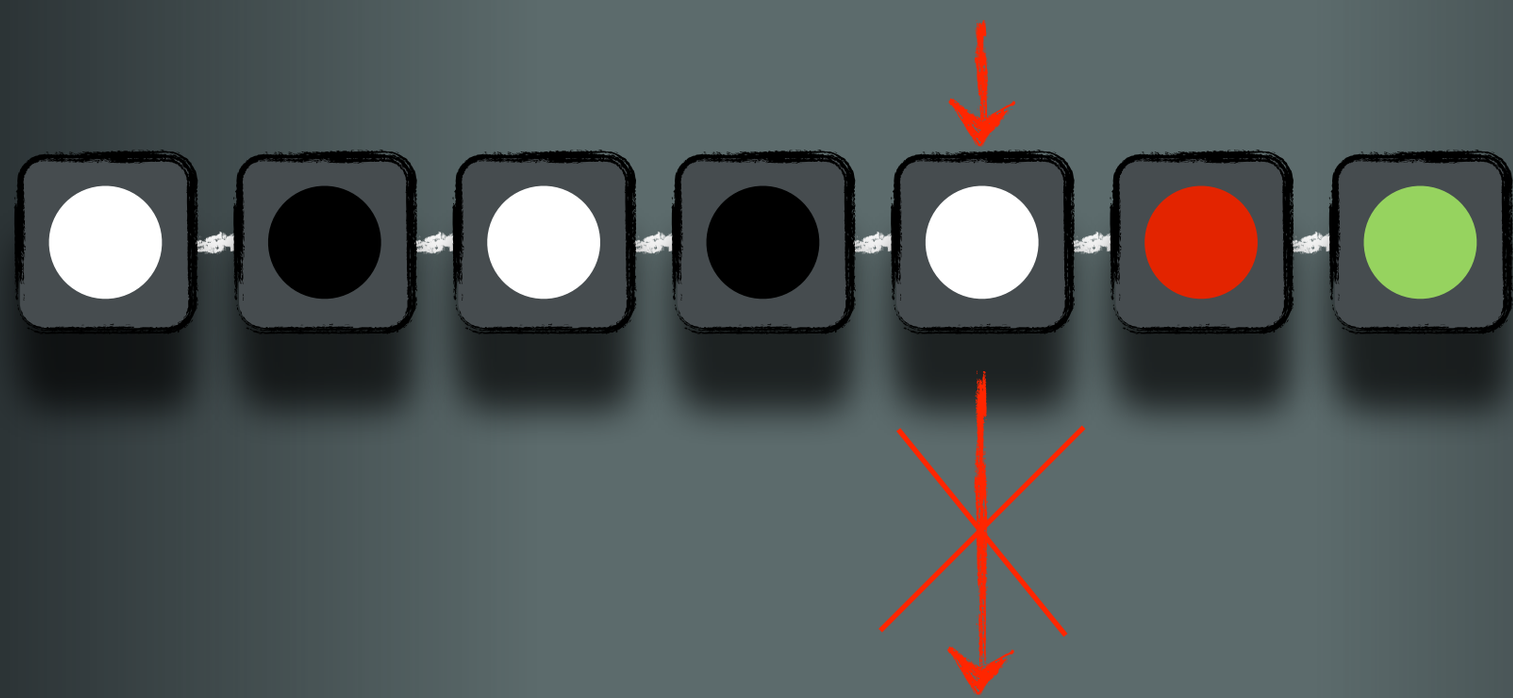




Existential Global Transition

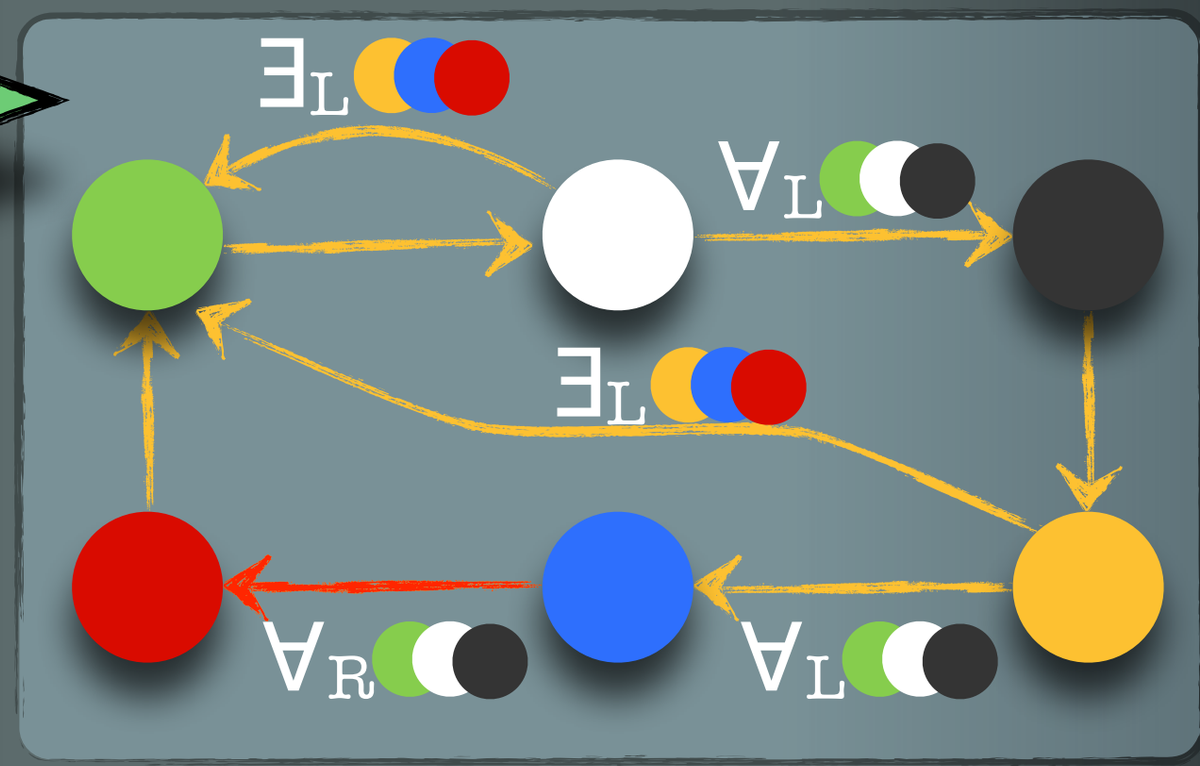
Parameterized System

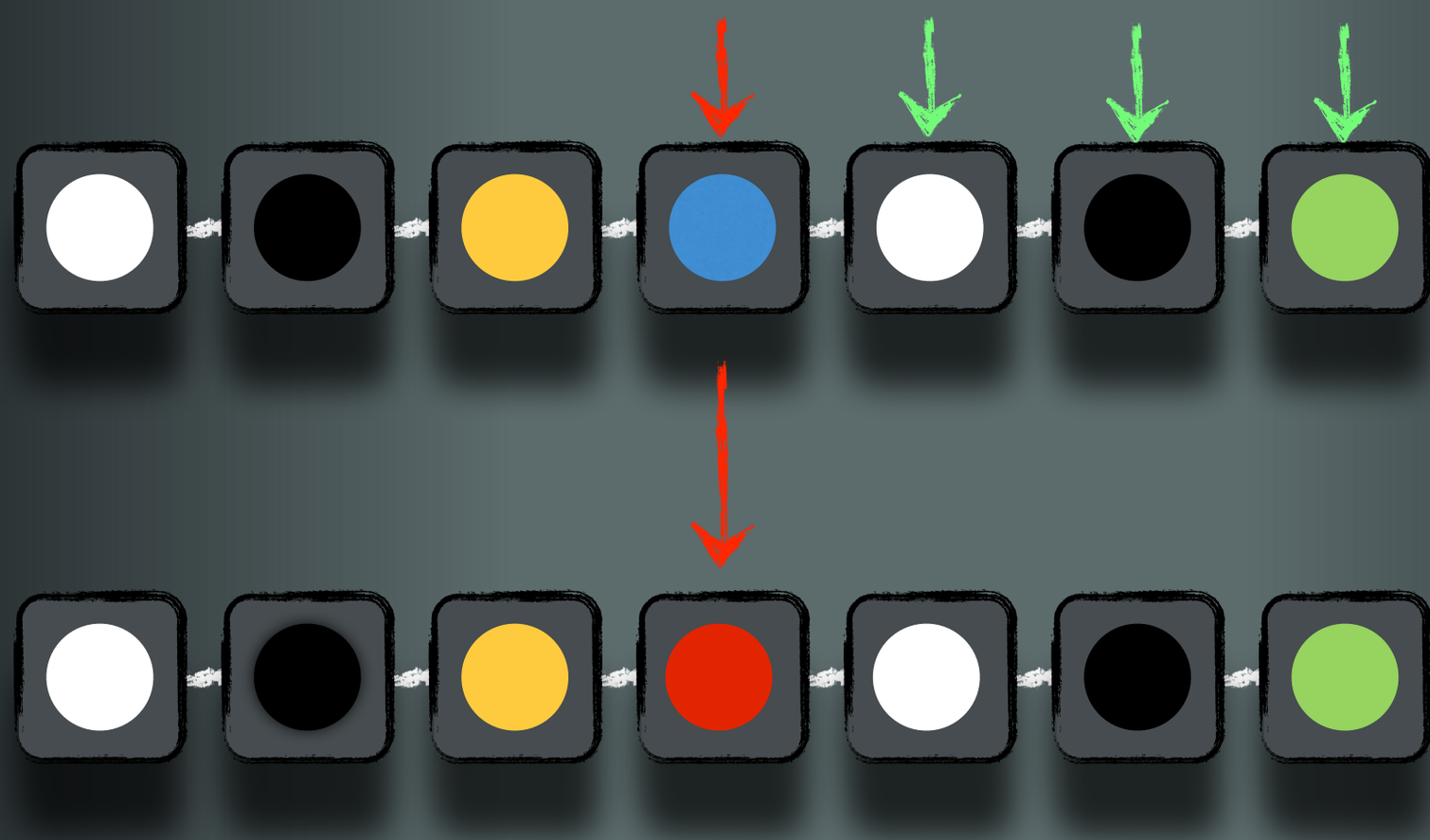




Existential Global Transition

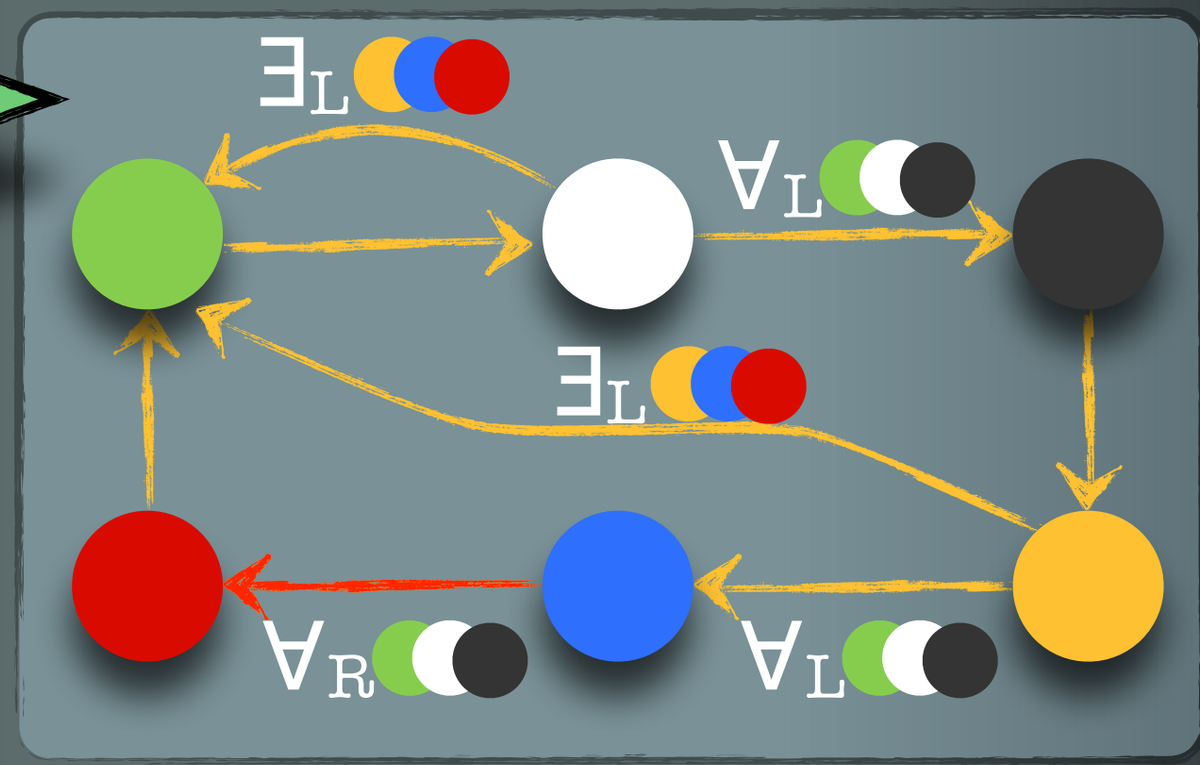
Parameterized System

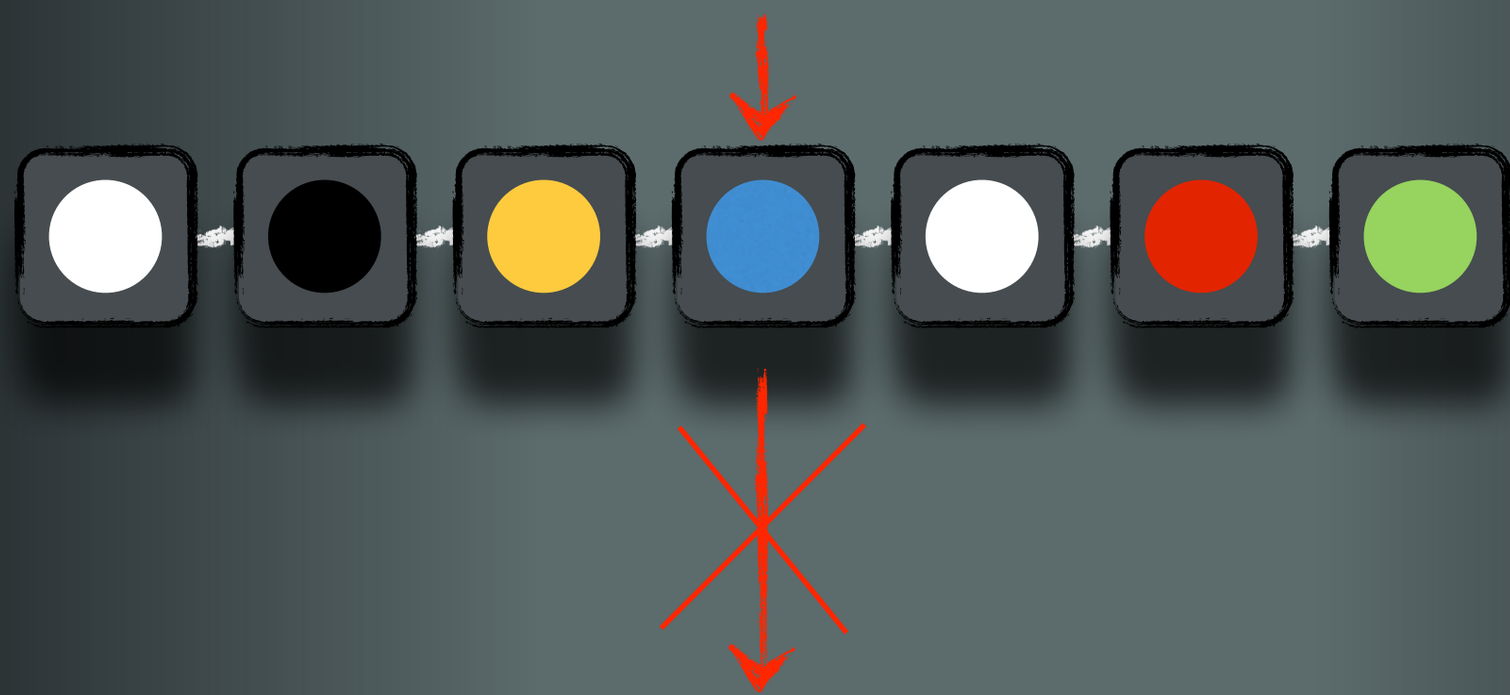




Universal Global Transition

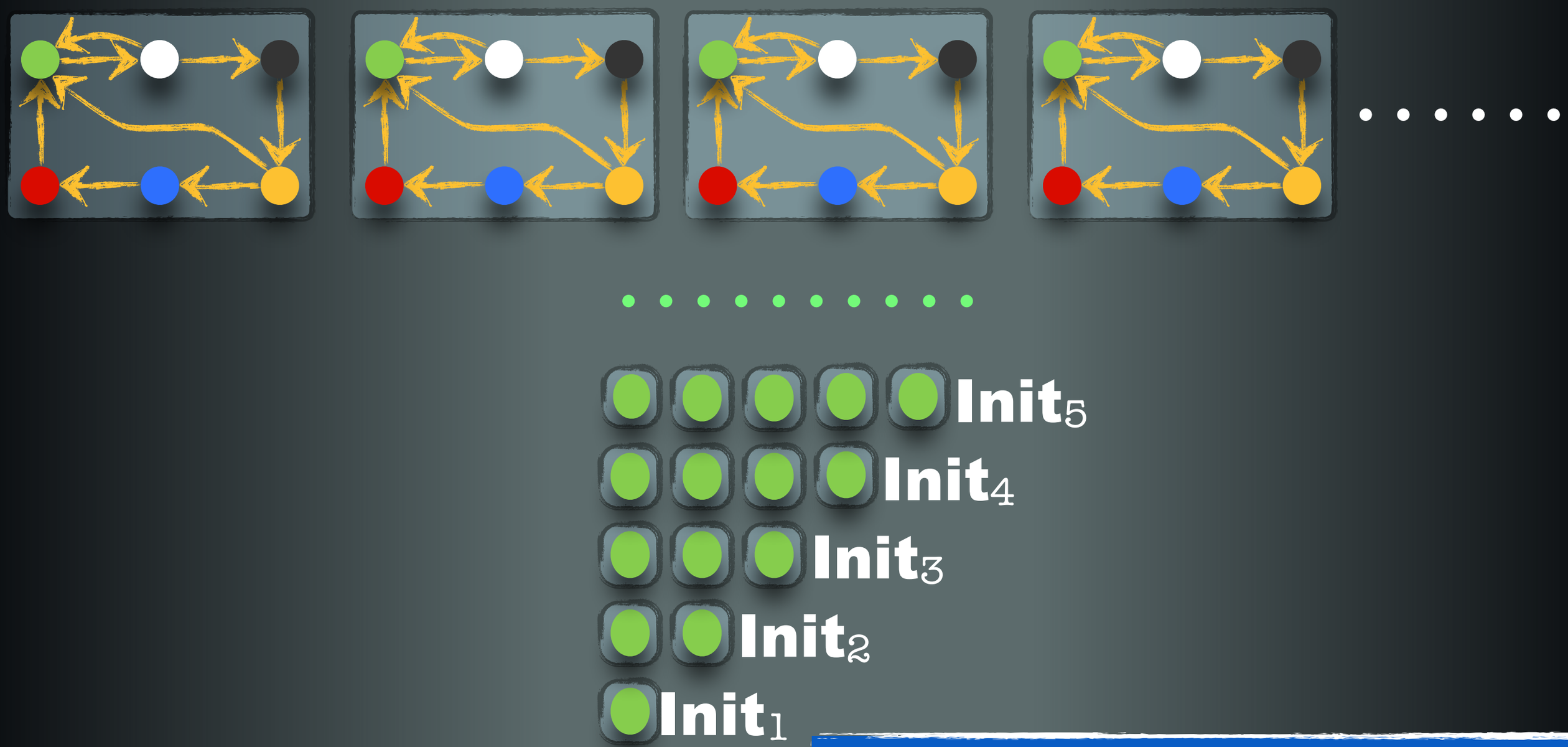
Parameterized System





Universal Global Transition

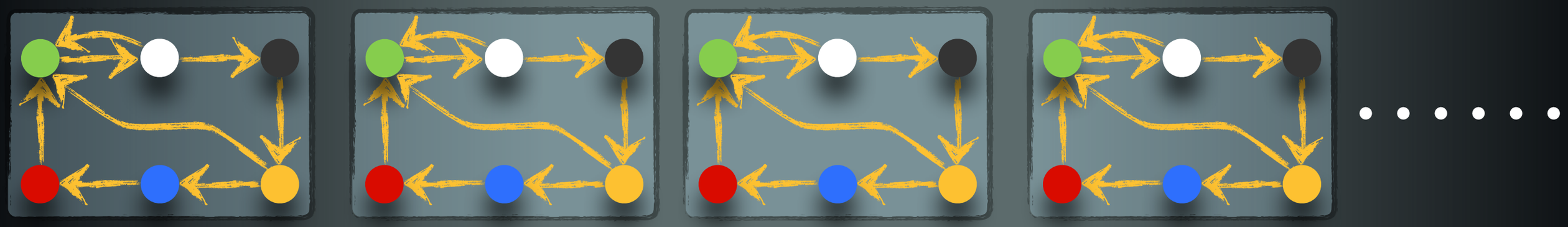
Parameterized System

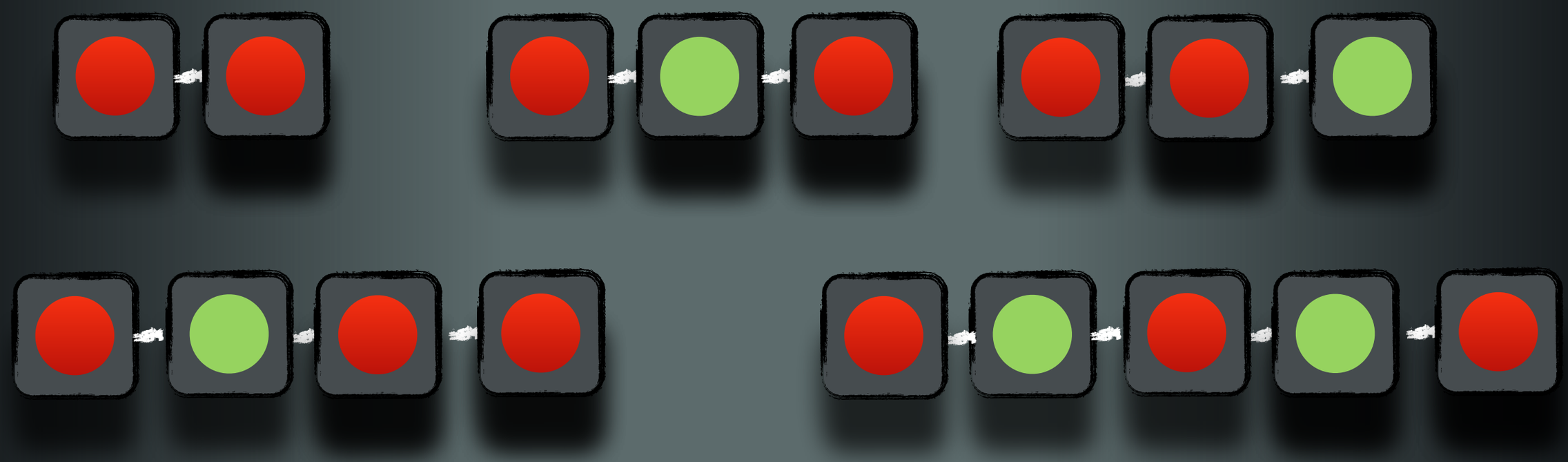


Initial Configurations

Set of initial configurations
- infinite, but
- regular

Parameterized System

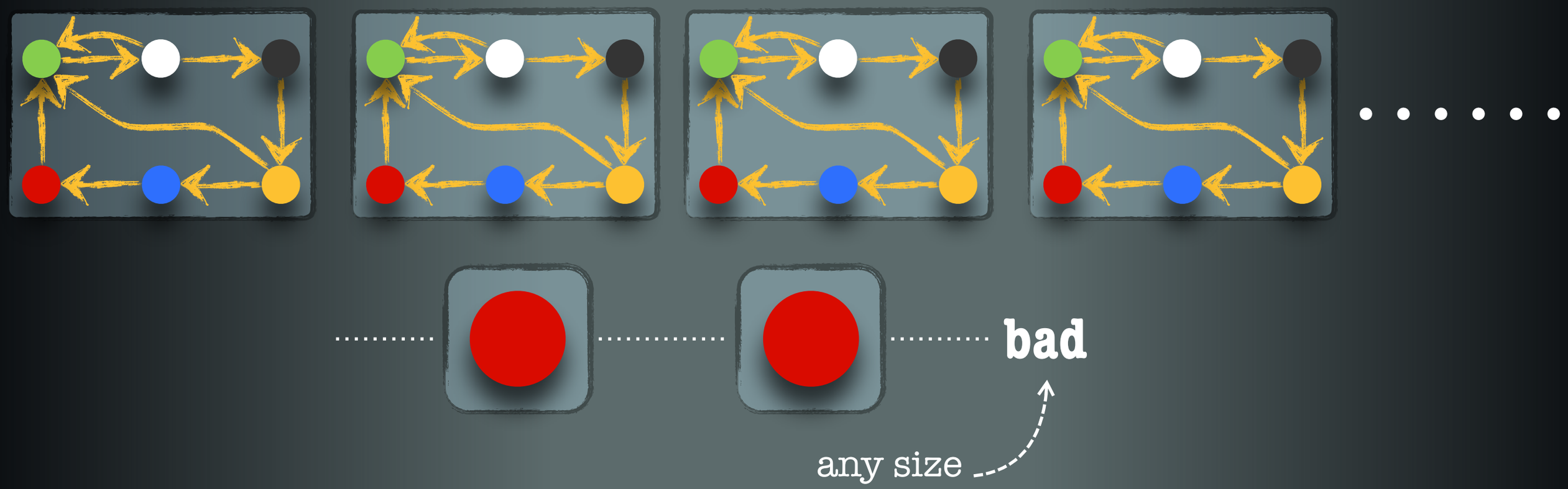




Bad Configurations

two or more processes in critical section

Parameterized System

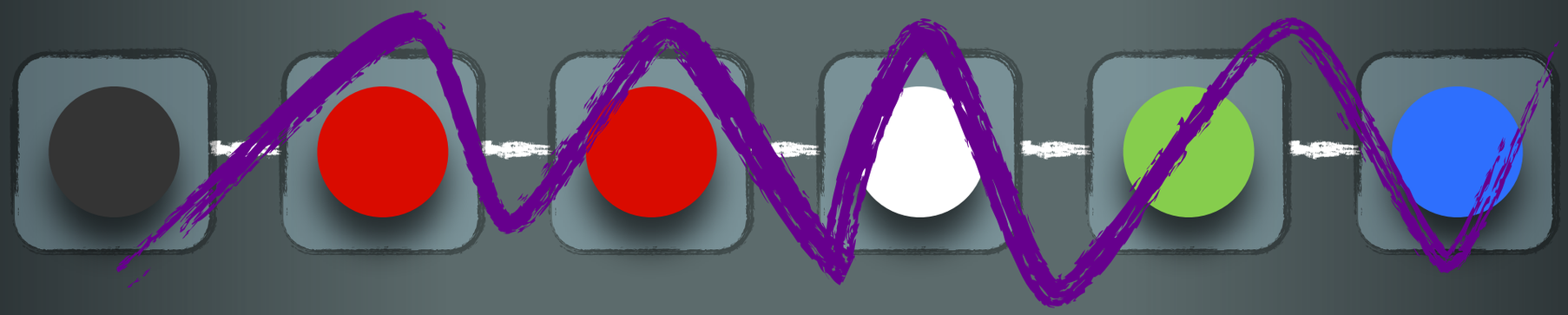


Set of bad configurations
- infinite, but
- upward closed

Bad Configurations

two or more processes in critical section

Parameterized System



Goal

Verify correctness regardless of # of processes

processes = parameter of the system

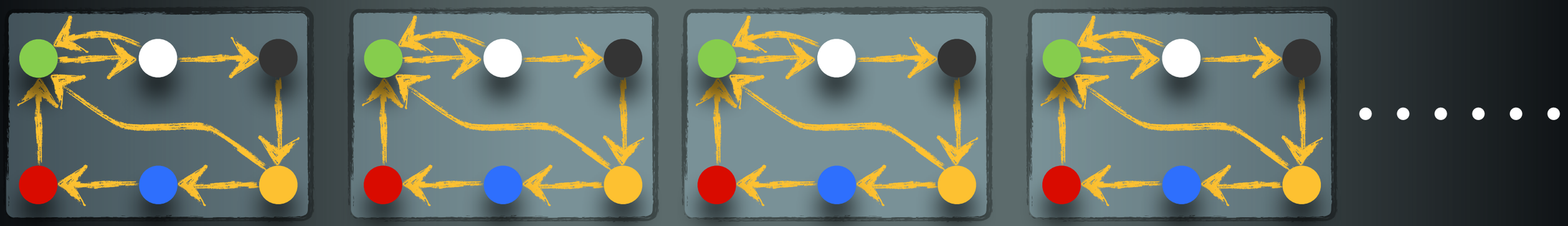
Parameterized Systems

- **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 System



Remarks

- Infinite-state system
  - unbounded number of processes
  - Parameterized Verification: verify correctness regardless of number of processes
- Problem undecidable in general
  - Challenge: find abstractions that work often

Parameterized System ✓

Analysis / Abstraction

Features

Completeness

Parameterized System

Analysis
Abstraction

Parameter System

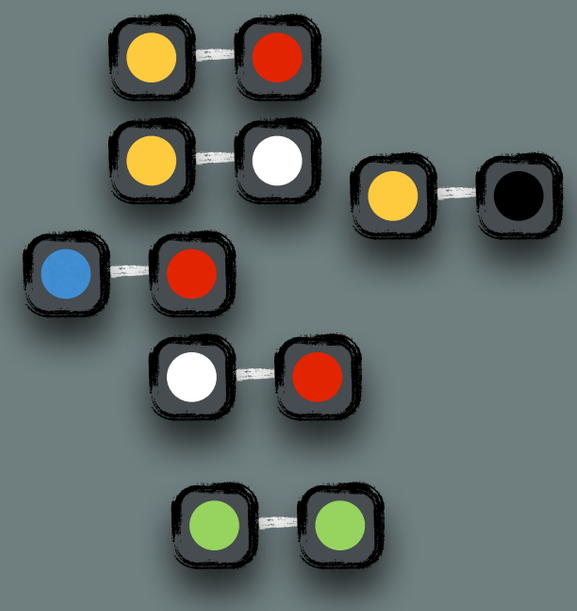
Analysis
Abstraction

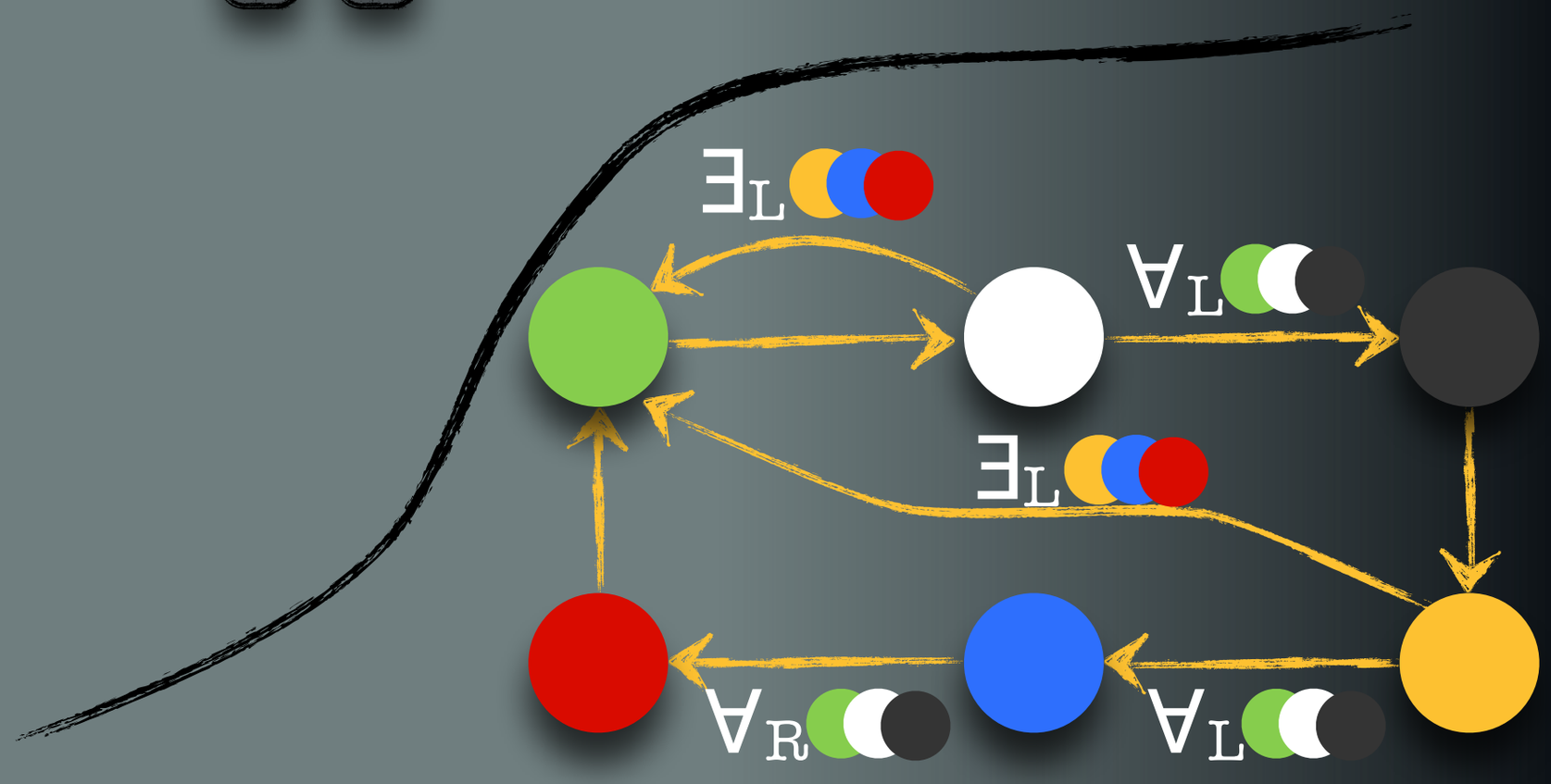
Small model property

⇒ inspect small instances of the system

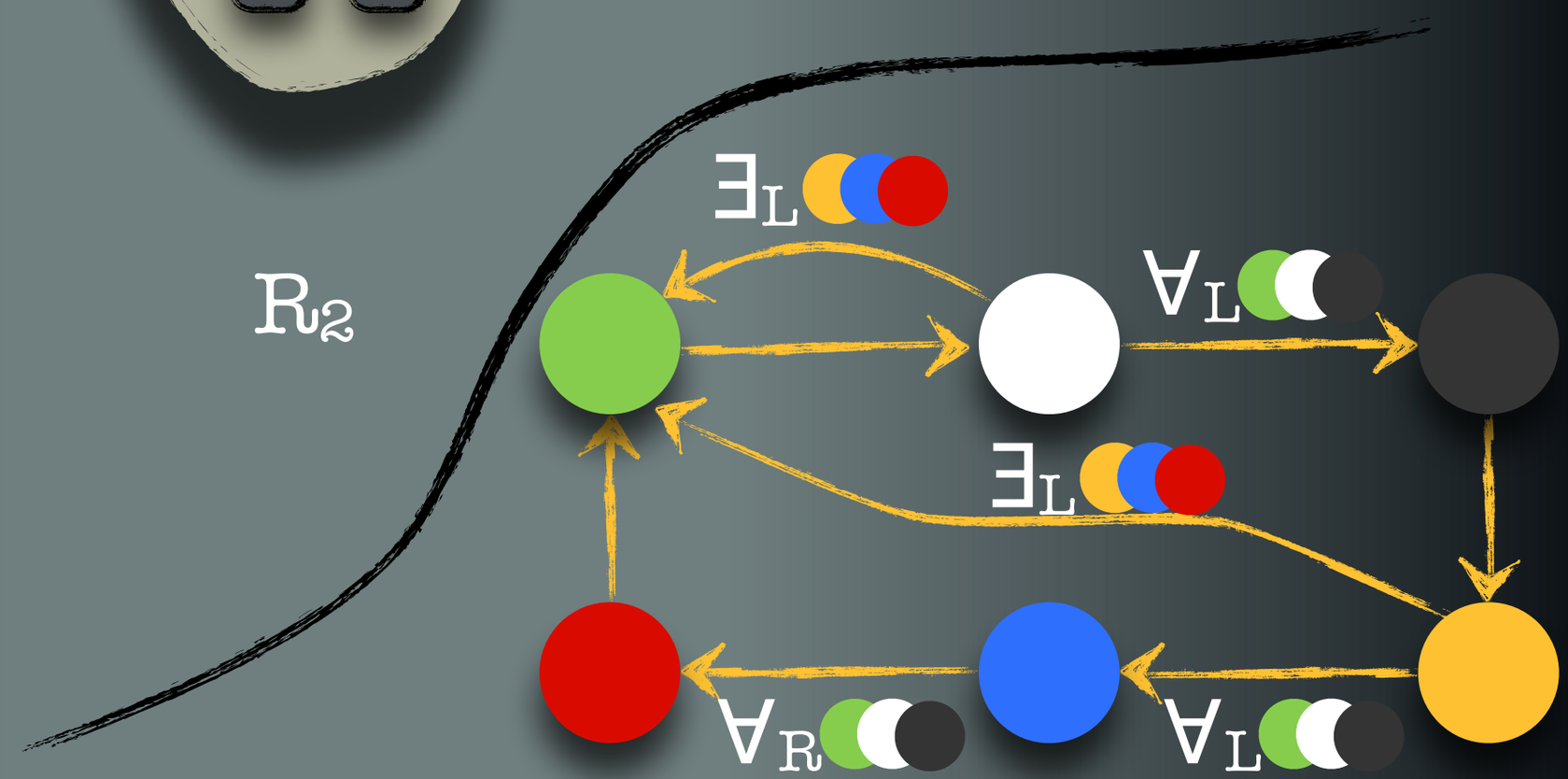
⇒ efficient method

Parameterized Systems





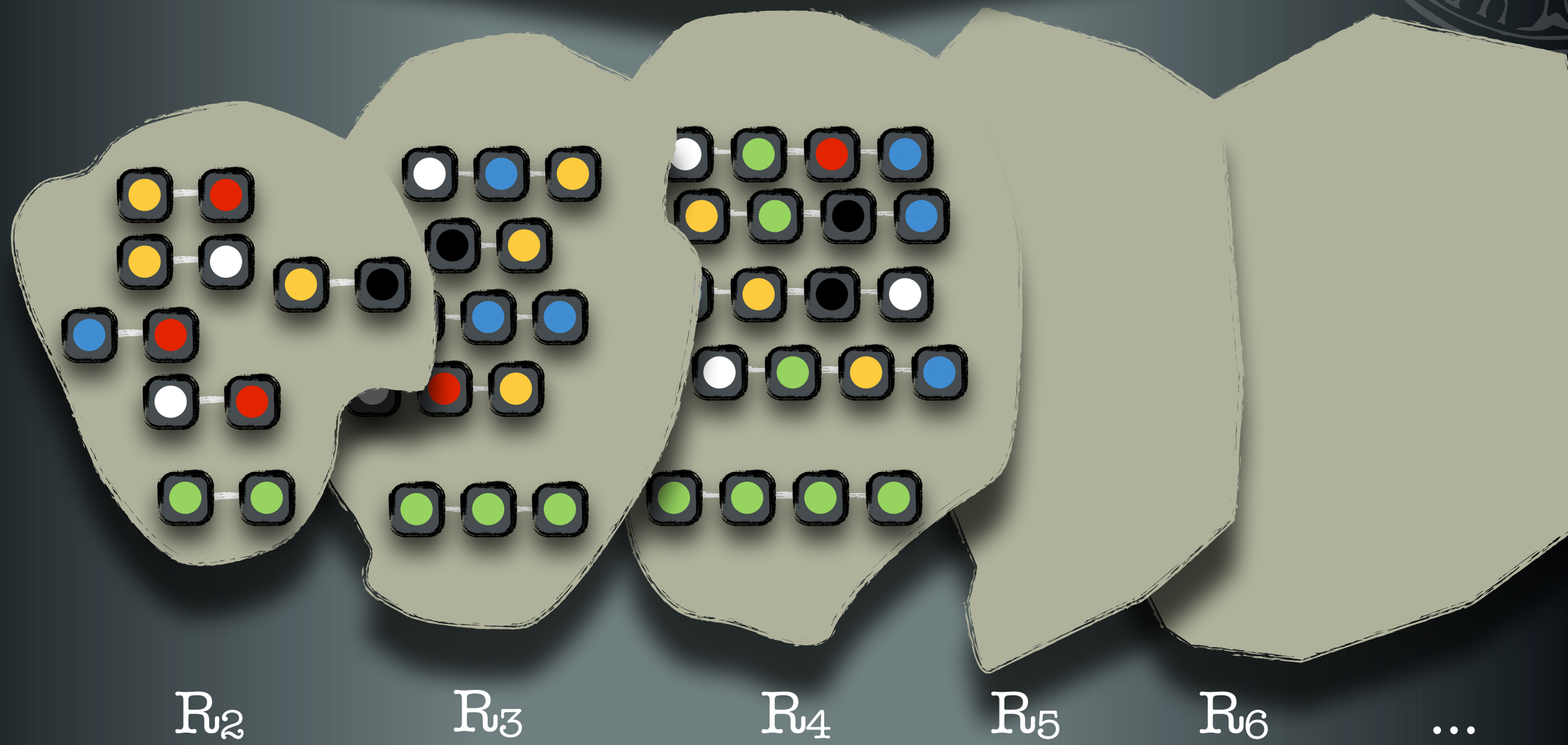
Parameterized Systems



Parameterized Systems

R_2 R_3 R_4

Parameterized Systems

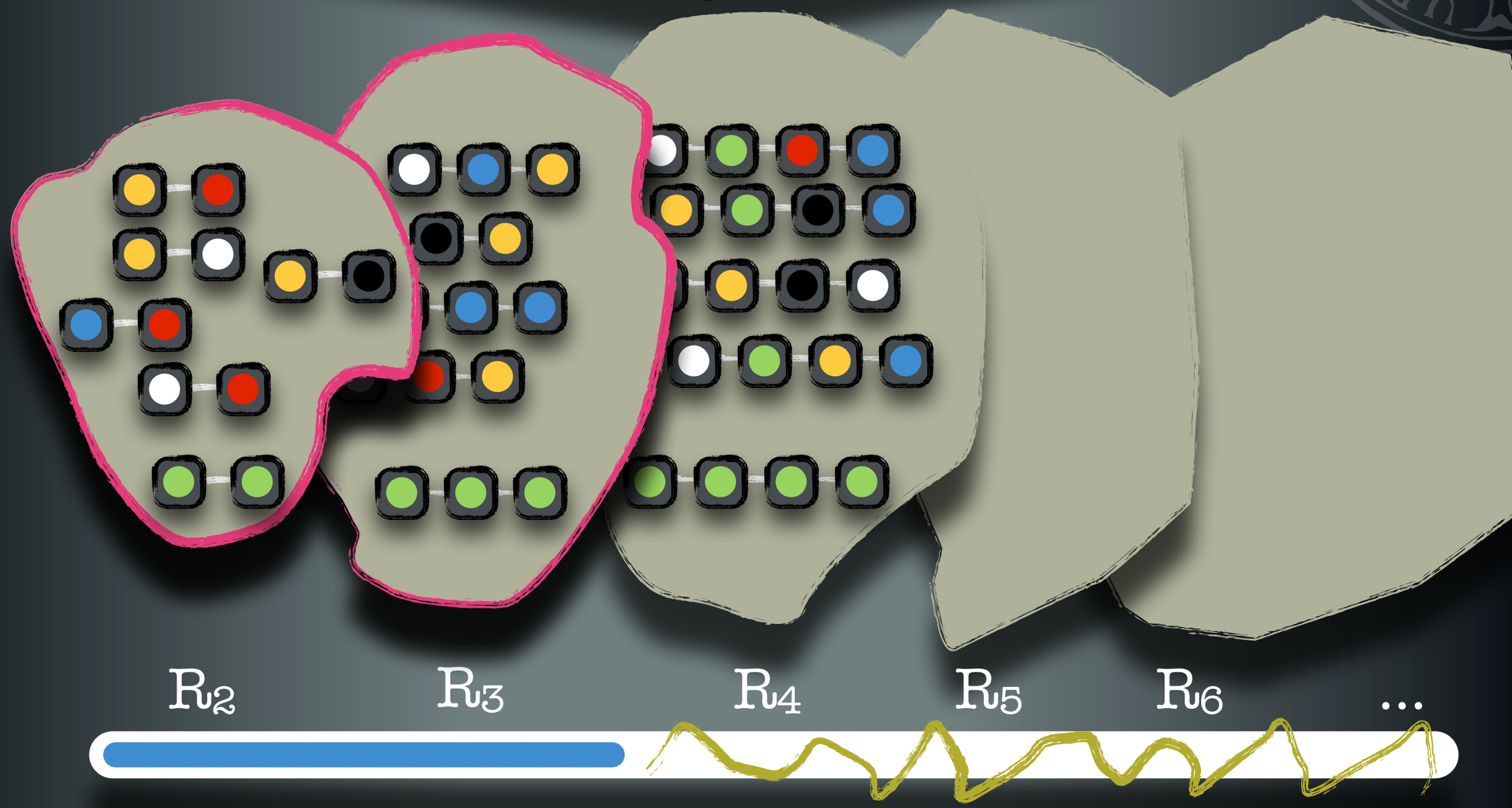


Goal

Safety

infinite family

Parameterized Systems



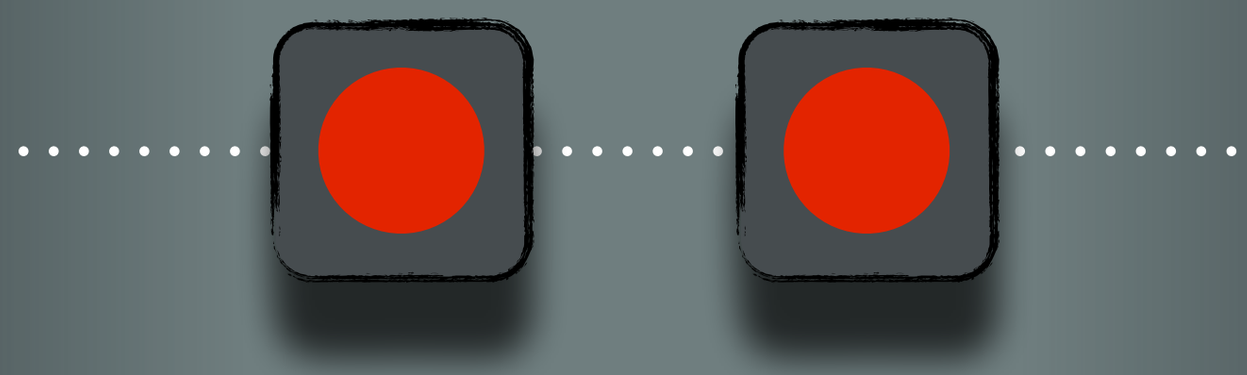
Small Model

- Efficient method

Parameterized Systems

Intuition

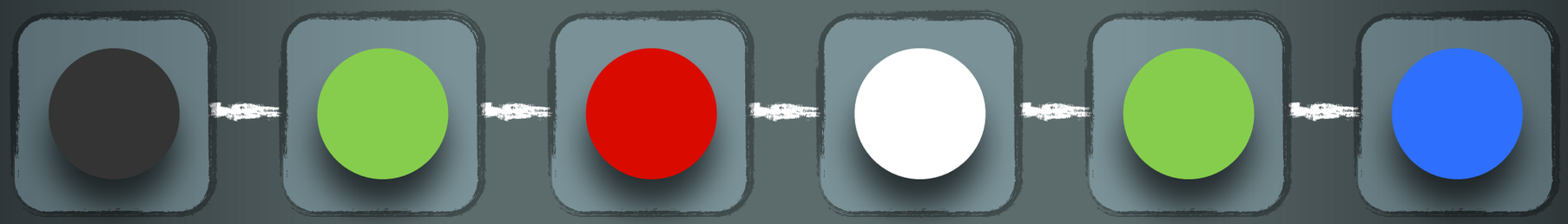
- Bad configurations:
  - can be characterized by fixed number of witness processes
- Bad patterns:
  - appear in small system instances



Small Model

Abstraction modulo k

- k: natural number



$a_k: \mathcal{C} \longrightarrow \mathcal{V}$

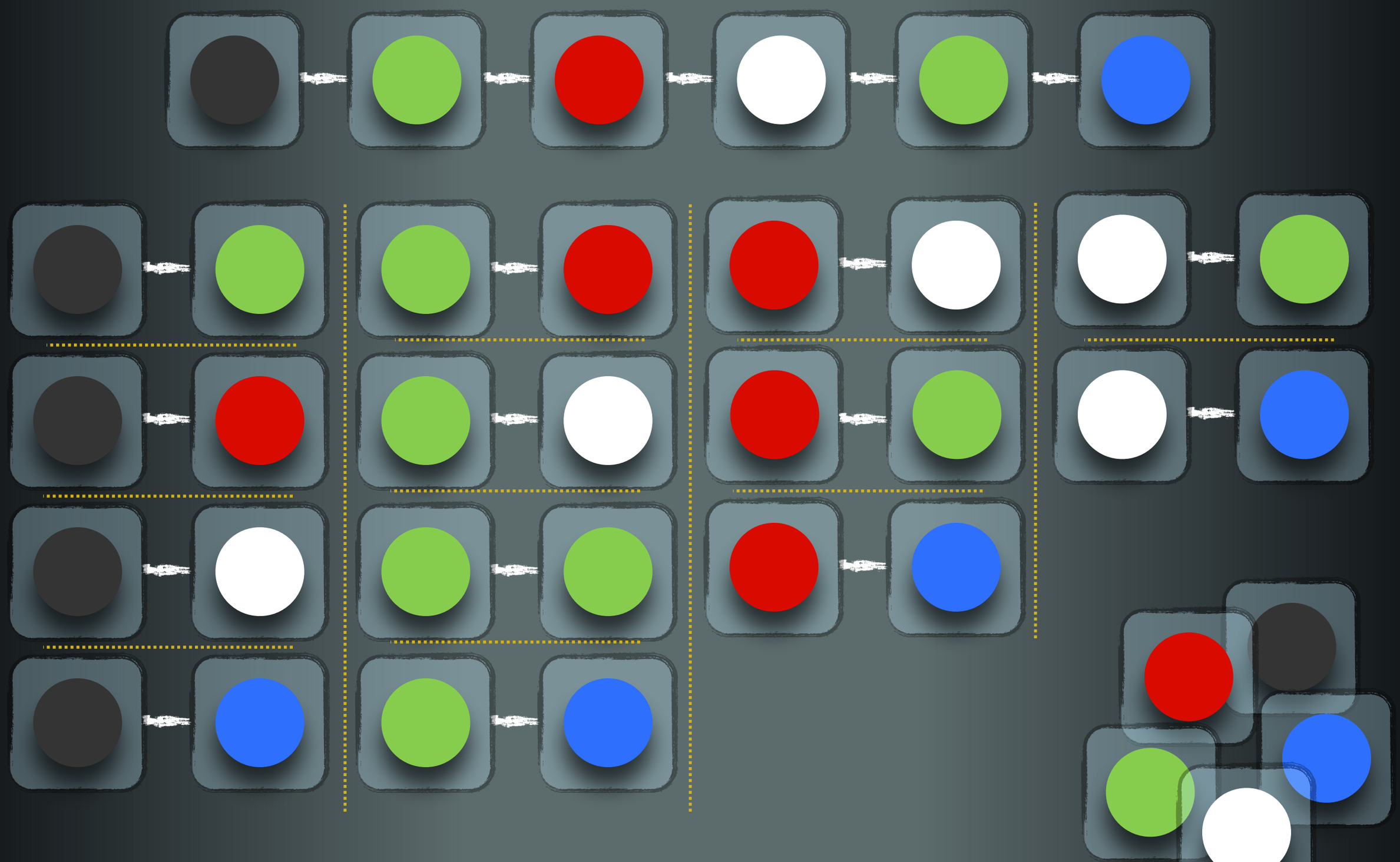
k=2

Parameter System

Analysis
Abstraction

$a_k: \mathcal{C} \longrightarrow \mathcal{V}$

k=2



Parameter System

Analysis
Abstraction

$a_k : \mathcal{C} \longrightarrow \mathcal{V}$

k=2

Parameter System

Analysis
Abstraction

For each k : two procedures (in parallel):

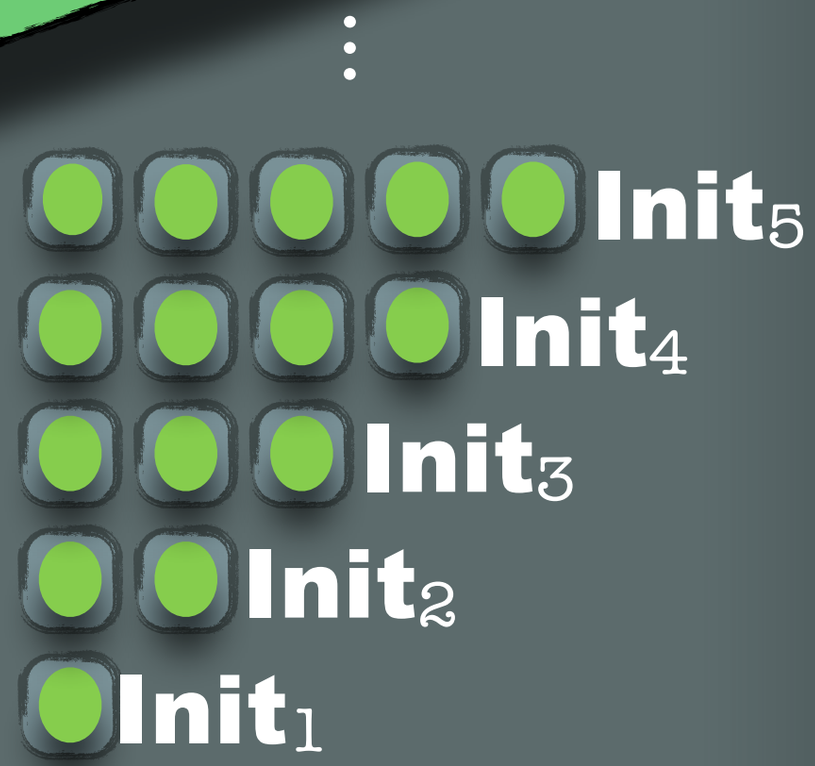
- Bug detection
  - under-approximation
  - concrete domain
- Verification
  - over-approximation
  - abstract domain

- abstraction of initial configurations
- abstract post operator

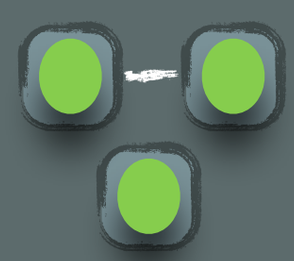
Parameter System

Analysis
Abstraction

$$\mathbb{I} = \bigcup_{n \geq 0} \mathbf{Init}_n$$



$$\mathbf{V} = \alpha_2(\mathbb{I})$$

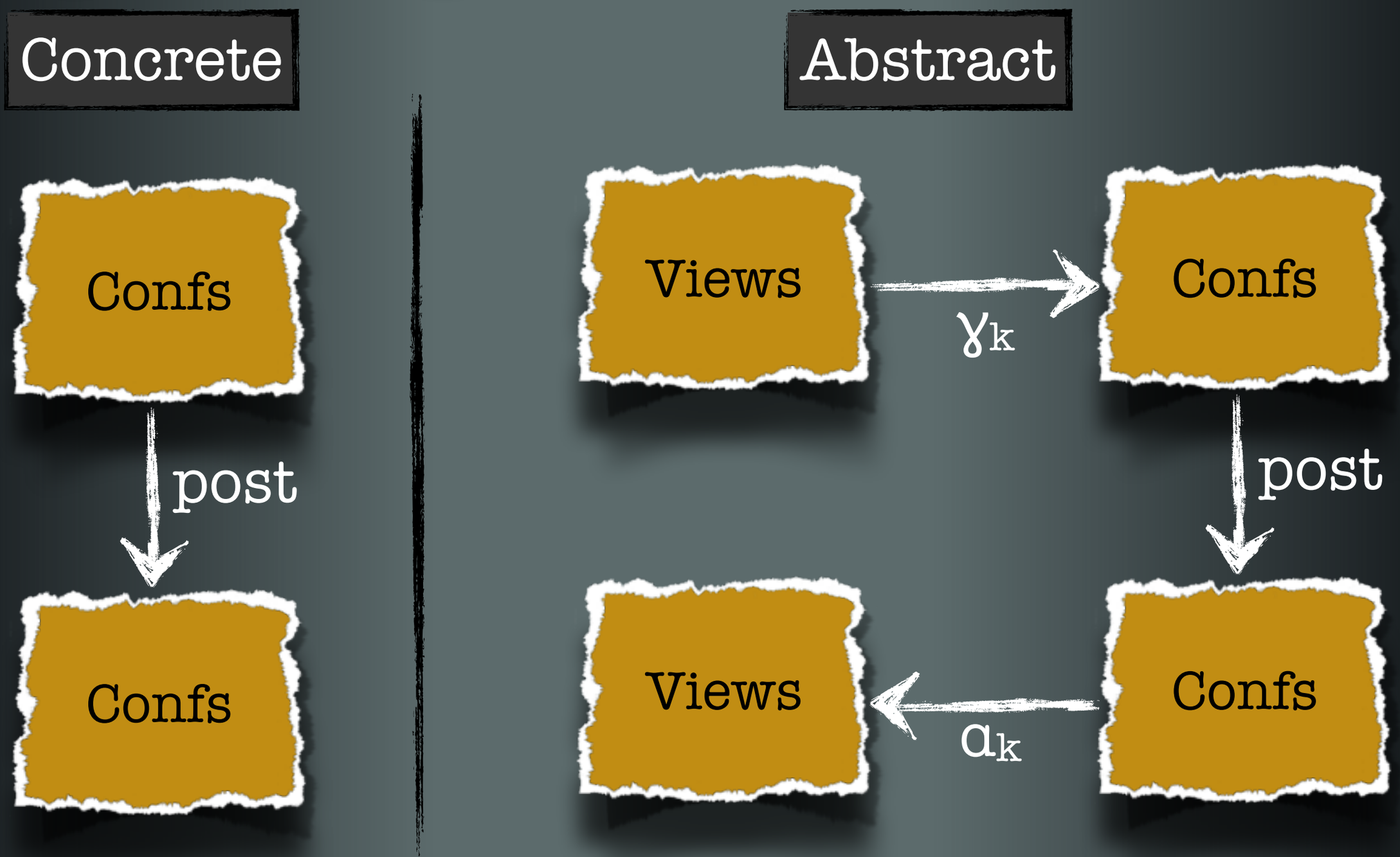


Parameter System

Analysis
Abstraction

Parameter System

Analysis
Abstraction



Parameter System

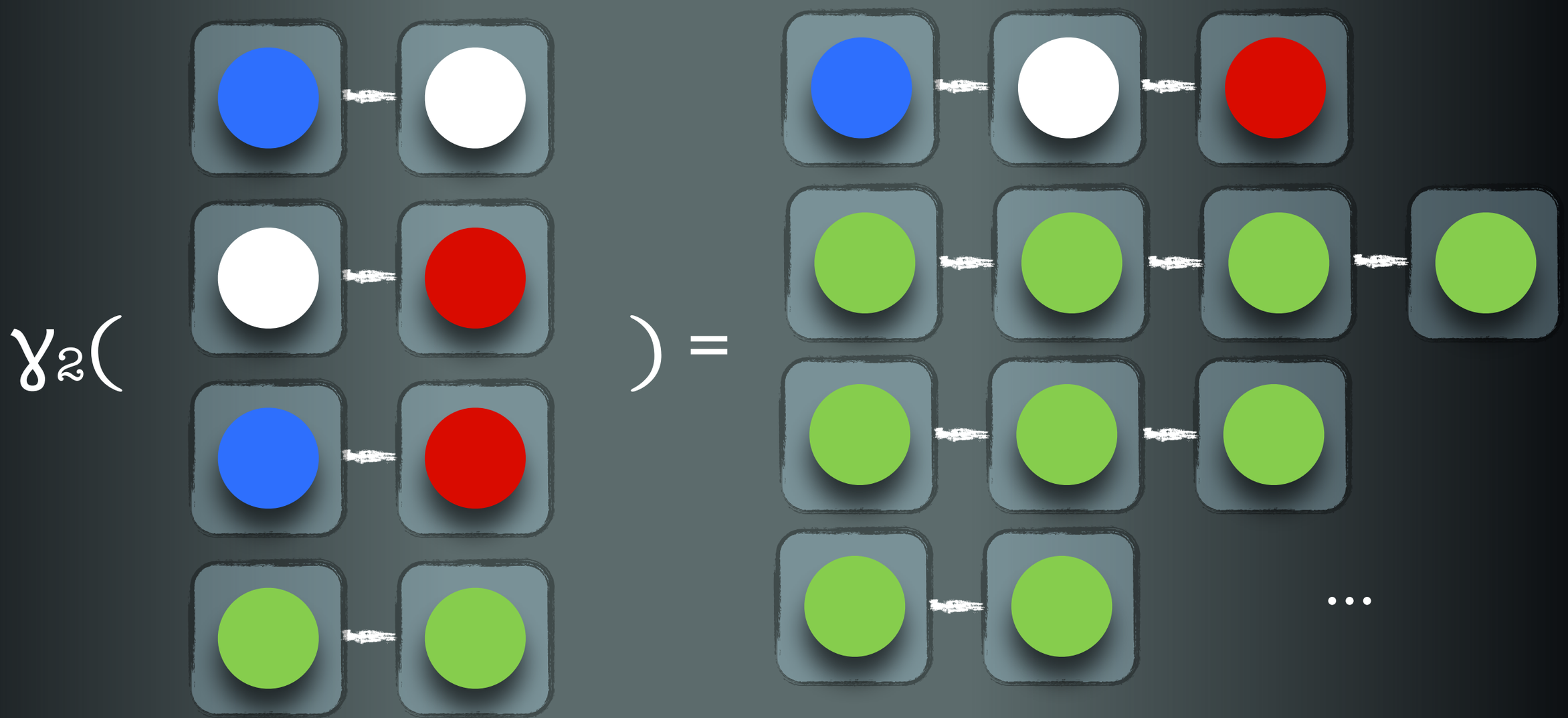
Analysis
Abstraction

γ_k: set of $\mathcal{V} \rightarrow$ set of $\mathcal{C}$

$$\gamma_k(\mathbf{X}) = \{ \, c \in \mathcal{C} \mid a_k(c) \subseteq \mathbf{X} \, \}$$

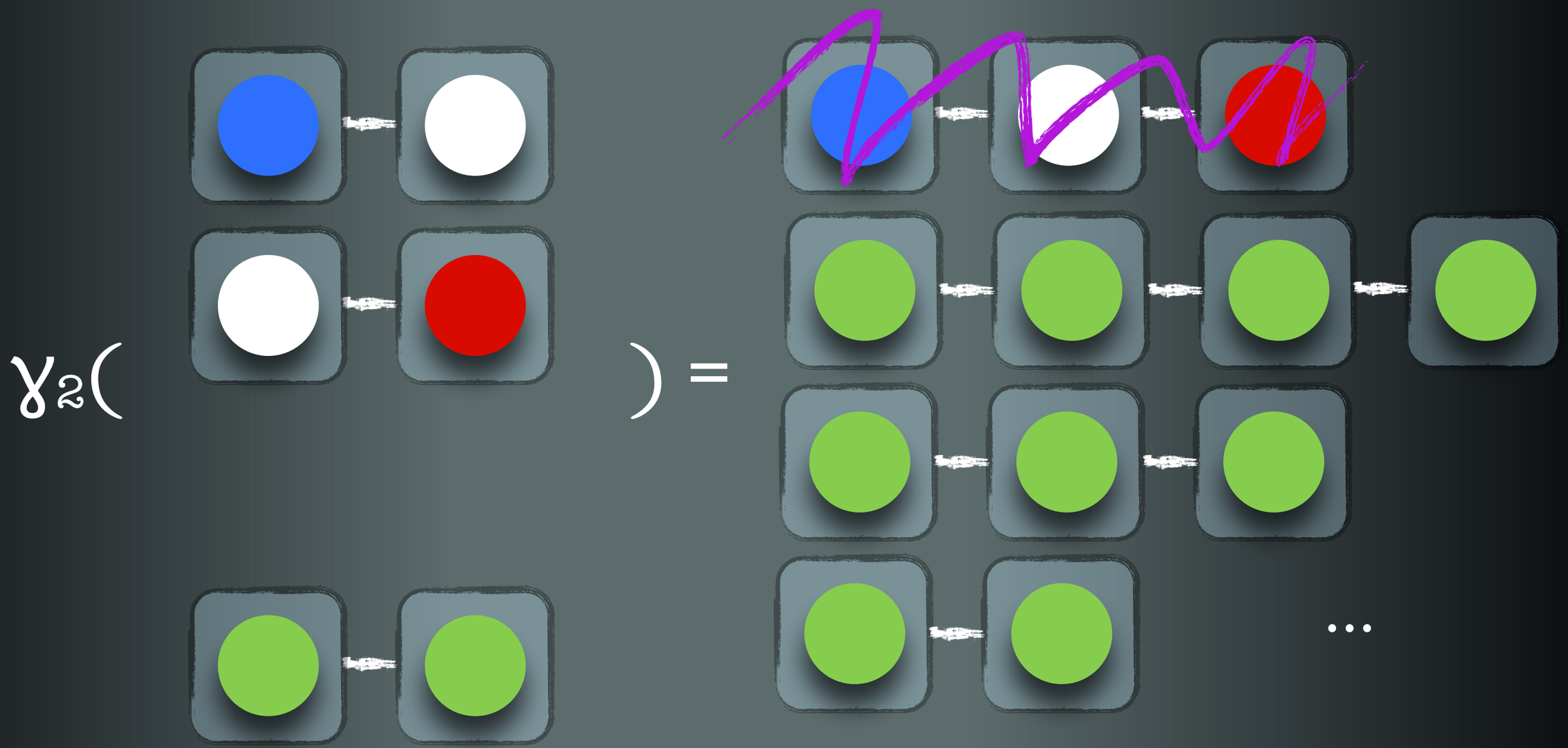
Parameter System

Analysis
Abstraction



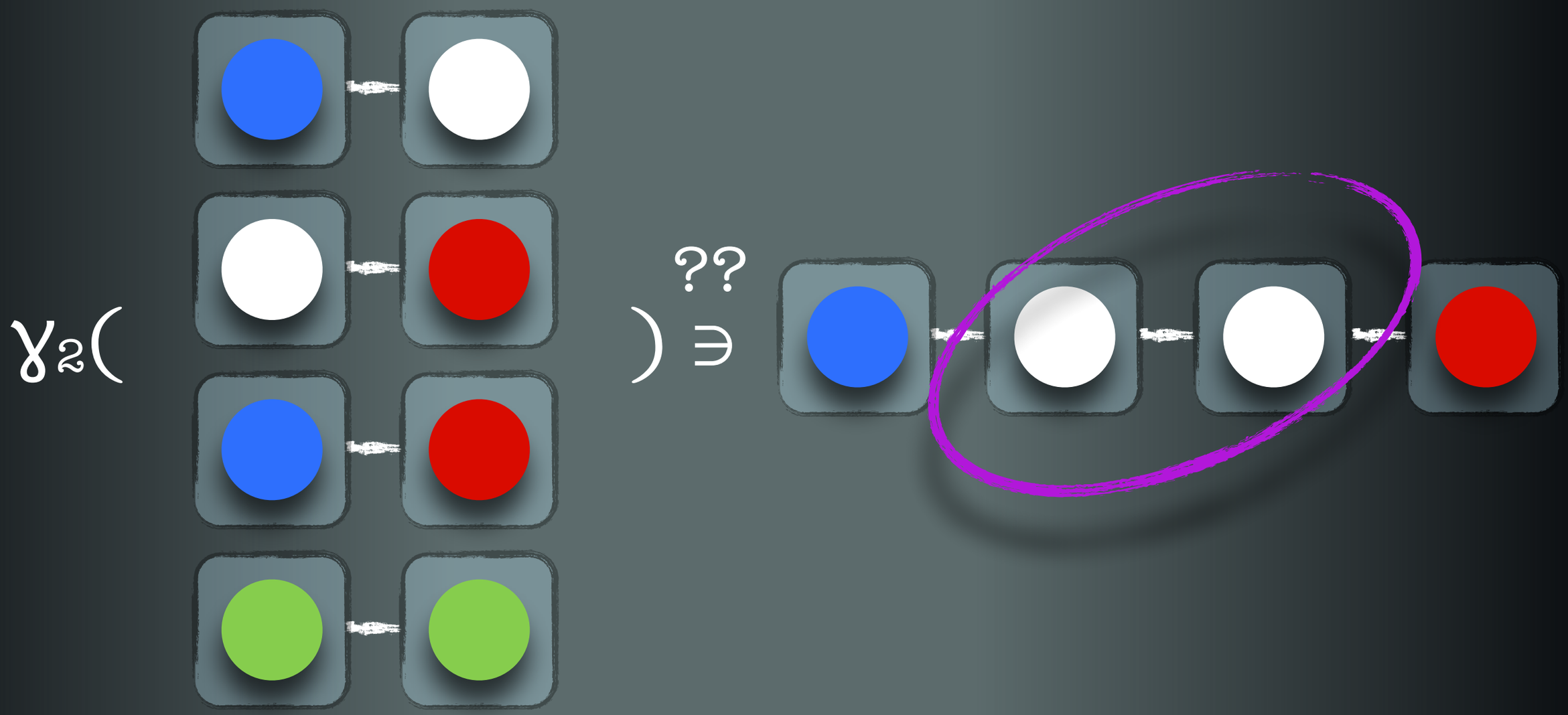
Parameter System

Analysis
Abstraction



Parameter System

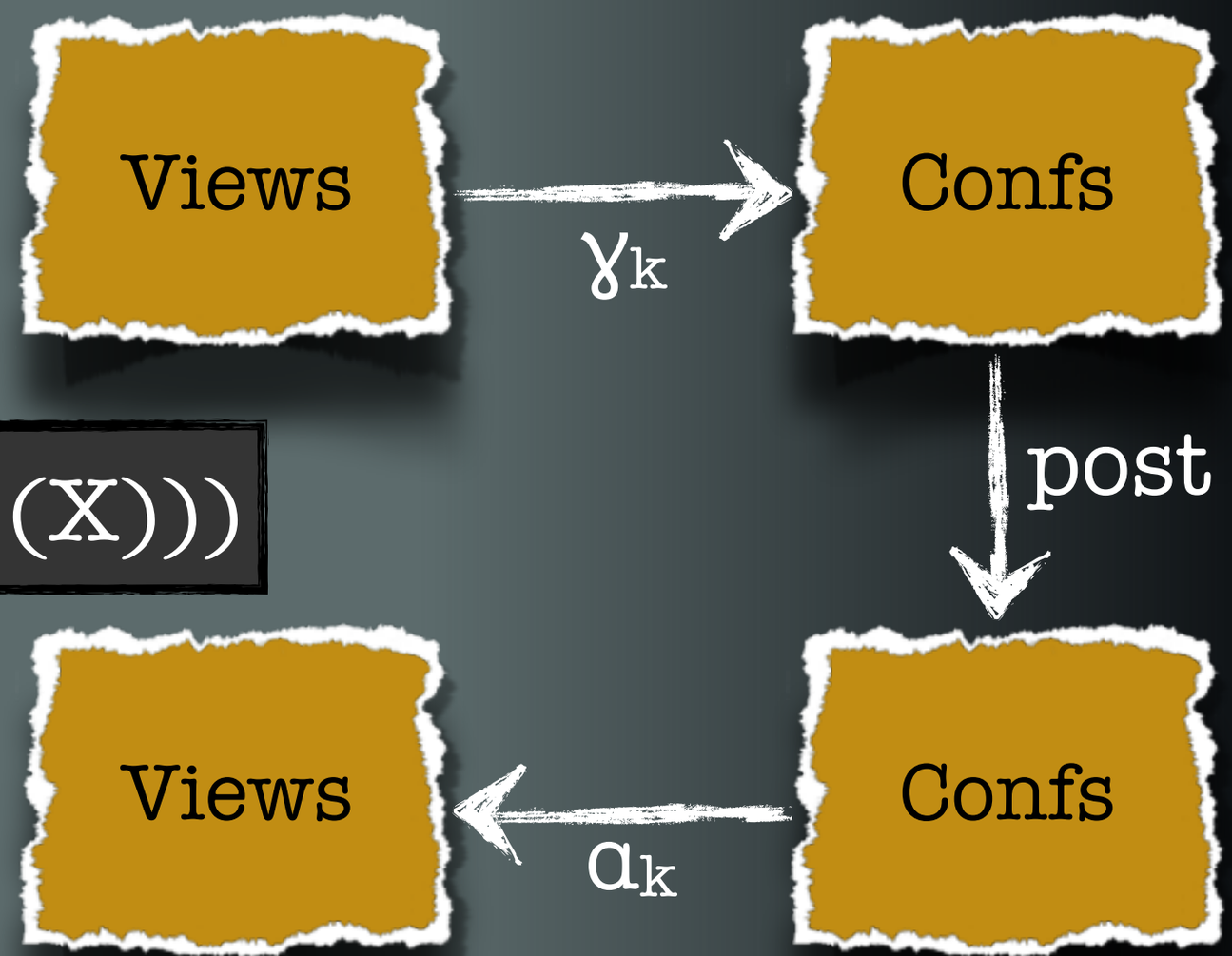
Analysis
Abstraction



Parameter System

Analysis
Abstraction

$$\mathrm{Apost}_k(X) := \alpha_k(\,\mathrm{post}\,(\,\gamma_k(X)))$$



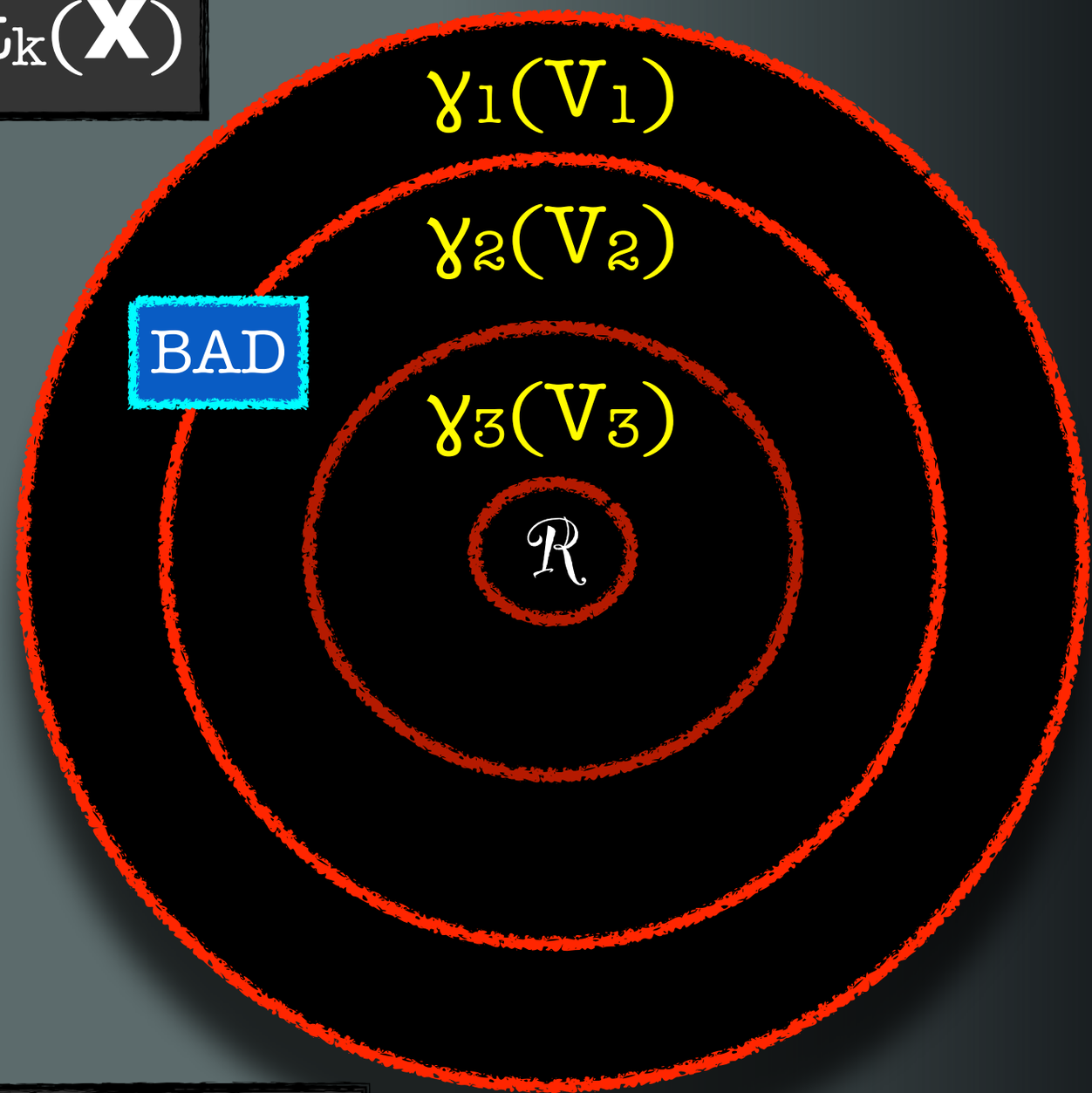
Parameter System

Analysis / Abstraction

$$V_k := \mu \mathbf{X} \,.\, \alpha_k(\mathbb{I}) \cup \mathrm{Apost}_k(\mathbf{X})$$

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

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



$$\mathrm{Apost}_k(\mathbf{X}) = \alpha_k(\,\mathrm{post}\,(\,\gamma_k(\mathbf{X})))$$

Parameter System

Analysis
Abstraction

$$\alpha_k (\text{post} (\gamma_k (\mathbf{V})))$$

$$\alpha_k (\text{post} (\gamma_{k|k+1} (\mathbf{V})))$$

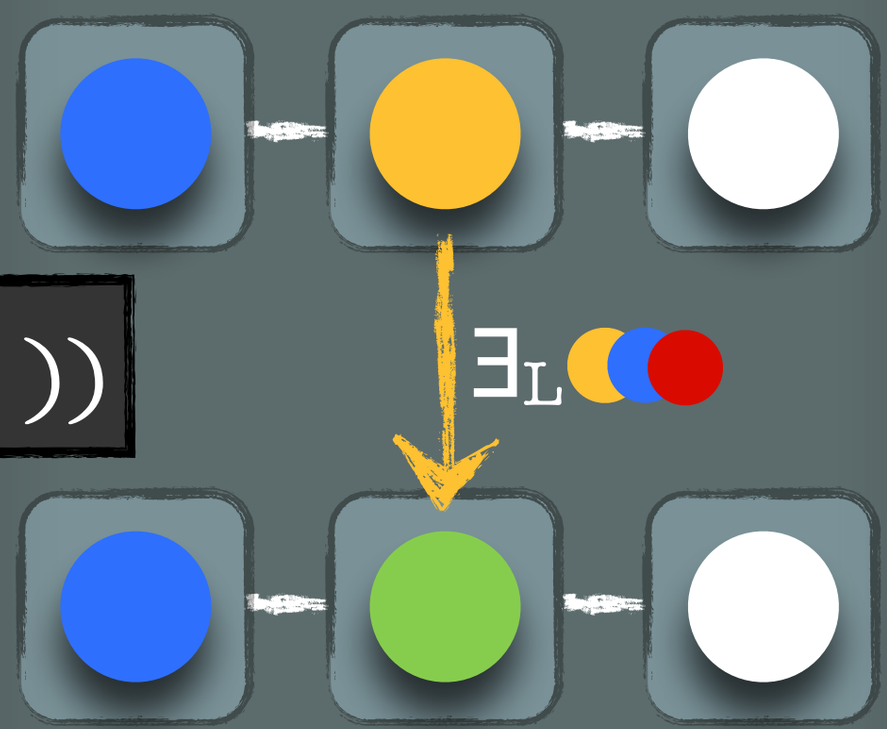
only of size k+1

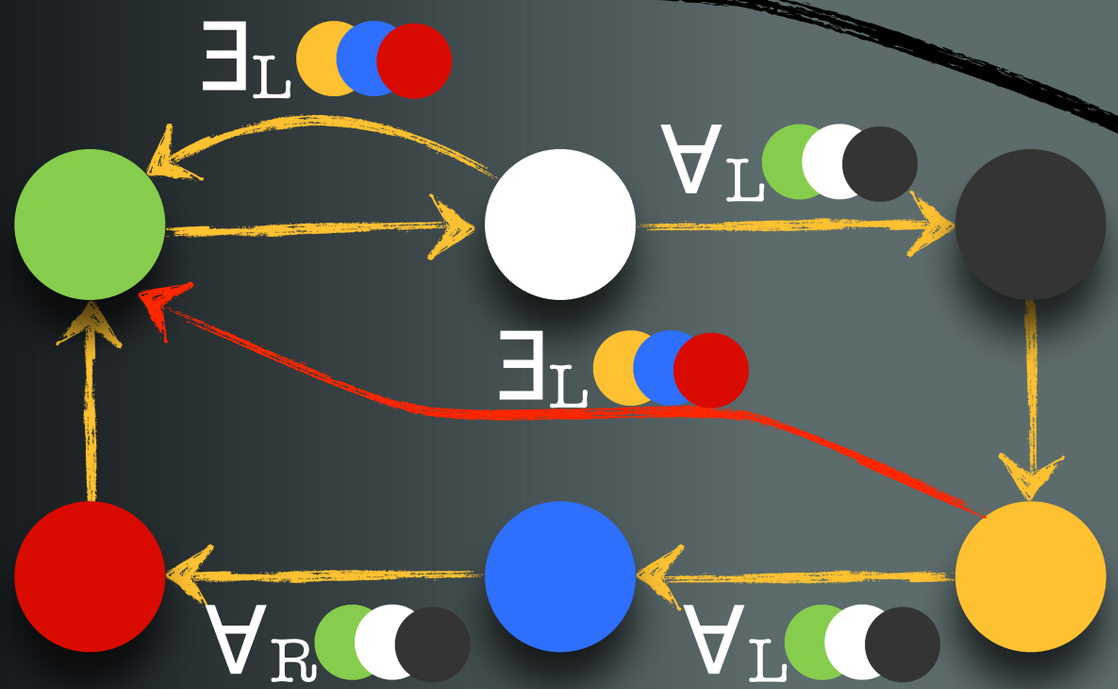
$$\gamma_{k|m}(\mathbf{X}) = \{ c \in \mathcal{C} \mid \alpha_k(c) \subseteq \mathbf{X} , |c| \leq m \}$$

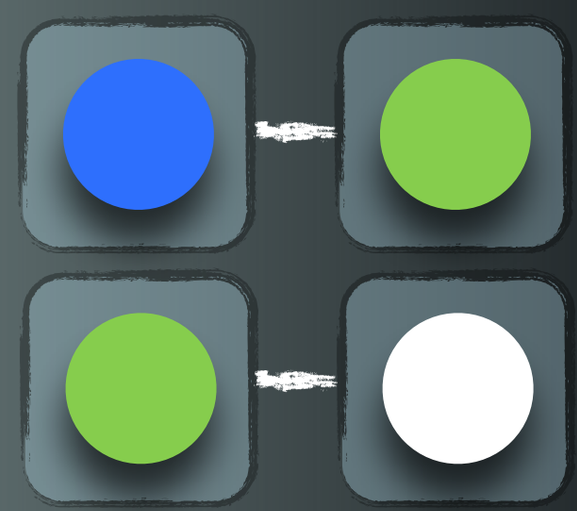
Parameter System

Analysis / Abstraction

$\alpha_2(\ \mathrm{post}\ (\ \gamma_{2|3}(\mathbf{V})\ ))$







Parameter System

Analysis
Abstraction

$$\mathrm{Apost}_k(\mathbf{X}) := \alpha_k(\,\mathrm{post}\,(\,\gamma_{k|k+1}\,(\mathbf{X})))$$

$$\cancel{V_k := \mu\mathbf{X}\,.\,\alpha_k\,(\mathbb{I}) \cup \alpha_k\,(\,\mathrm{post}\,(\,\gamma_k\,(\mathbf{X})))}$$

$$V_k := \mu\mathbf{X}\,.\,\alpha_k\,(\mathbb{I}) \cup \alpha_k\,(\,\mathrm{post}\,(\,\gamma_{k|k+1}\,(\mathbf{X})))$$

Parameter System

Analysis / Abstraction

for k = 1 to ∞ do

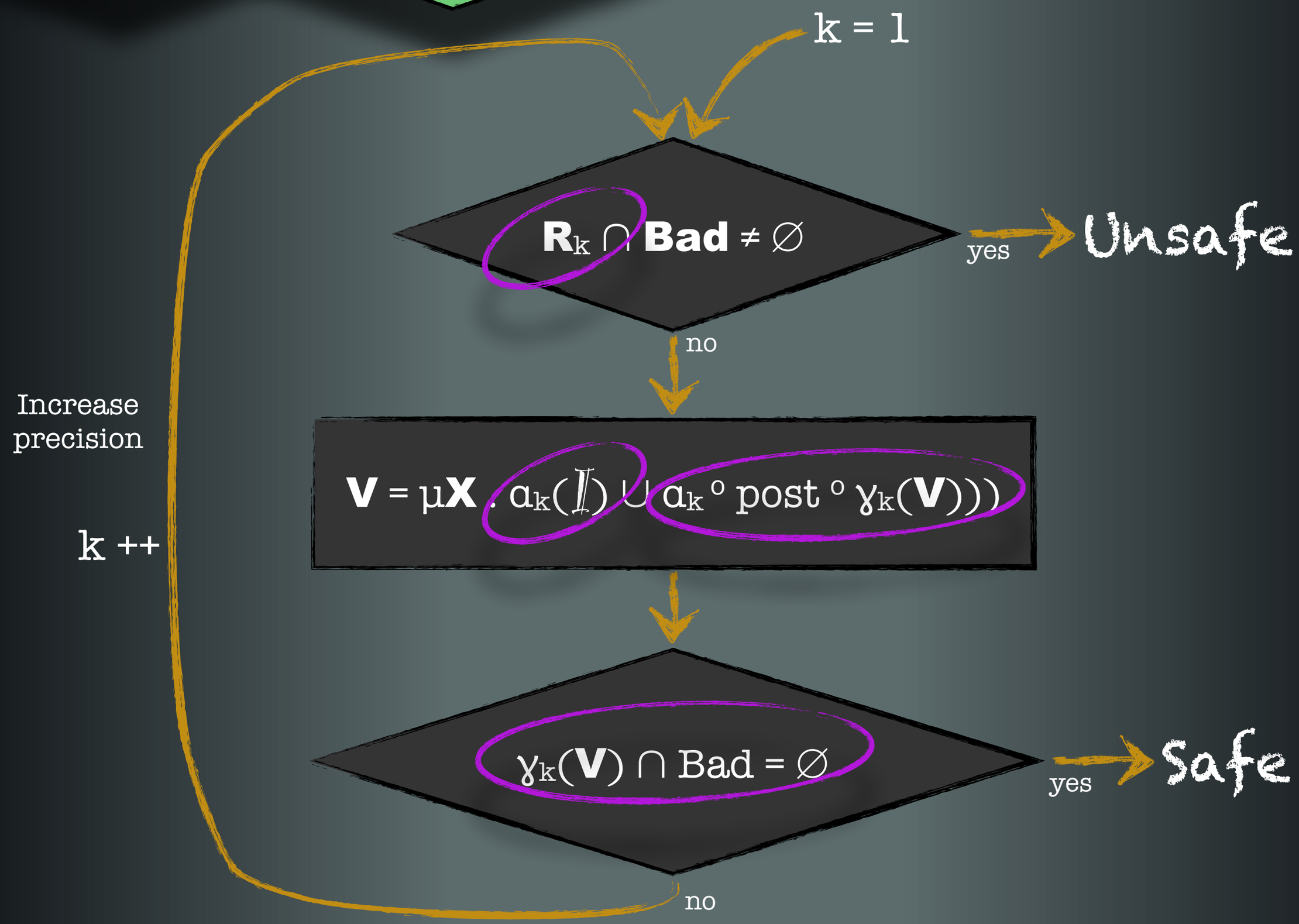
if $\mathbf{R}_k \cap \mathbf{Bad} \neq \varnothing$ then return Unsafe

$\mathbf{V} = \mu\mathbf{X} \,.\, \alpha_k(\mathbb{I}) \cup \alpha_k \circ \mathrm{post} \circ \gamma_k(\mathbf{X})$

if $\gamma_k(\mathbf{V}) \cap \mathbf{Bad} = \varnothing$ then return Safe

Parameteri
System

Analysis
Abstraction

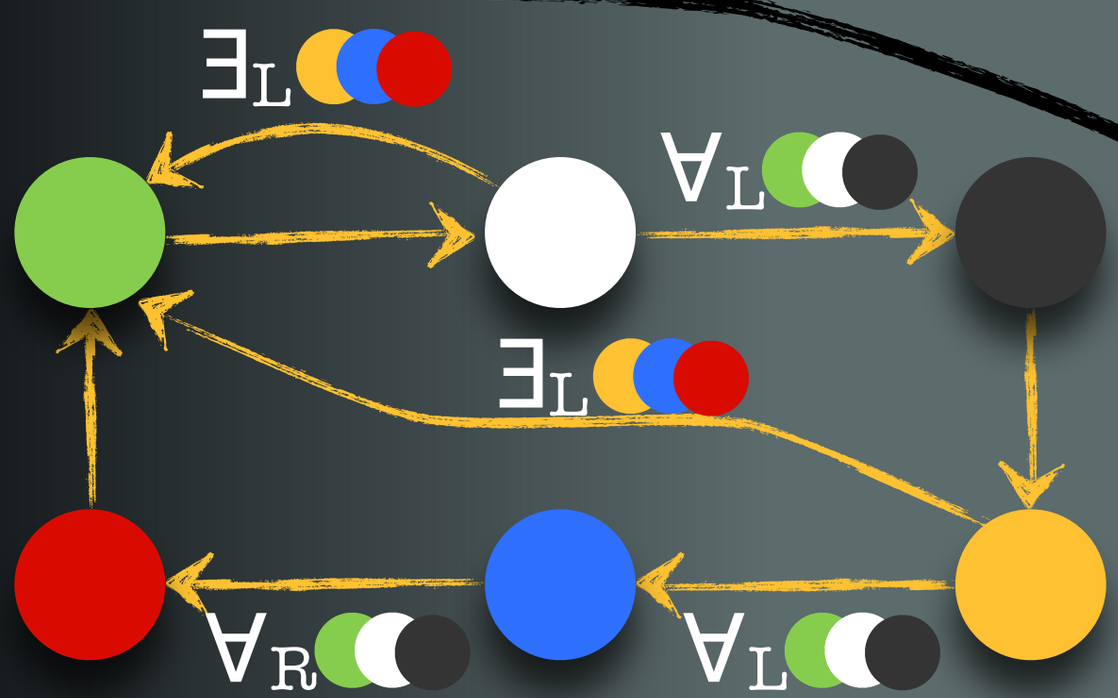


Parameter System

Analysis
Abstraction

k=1 $\mathbf{R}_1$:

$\mathbf{R}_1 \cap \mathbf{Bad} = \varnothing$

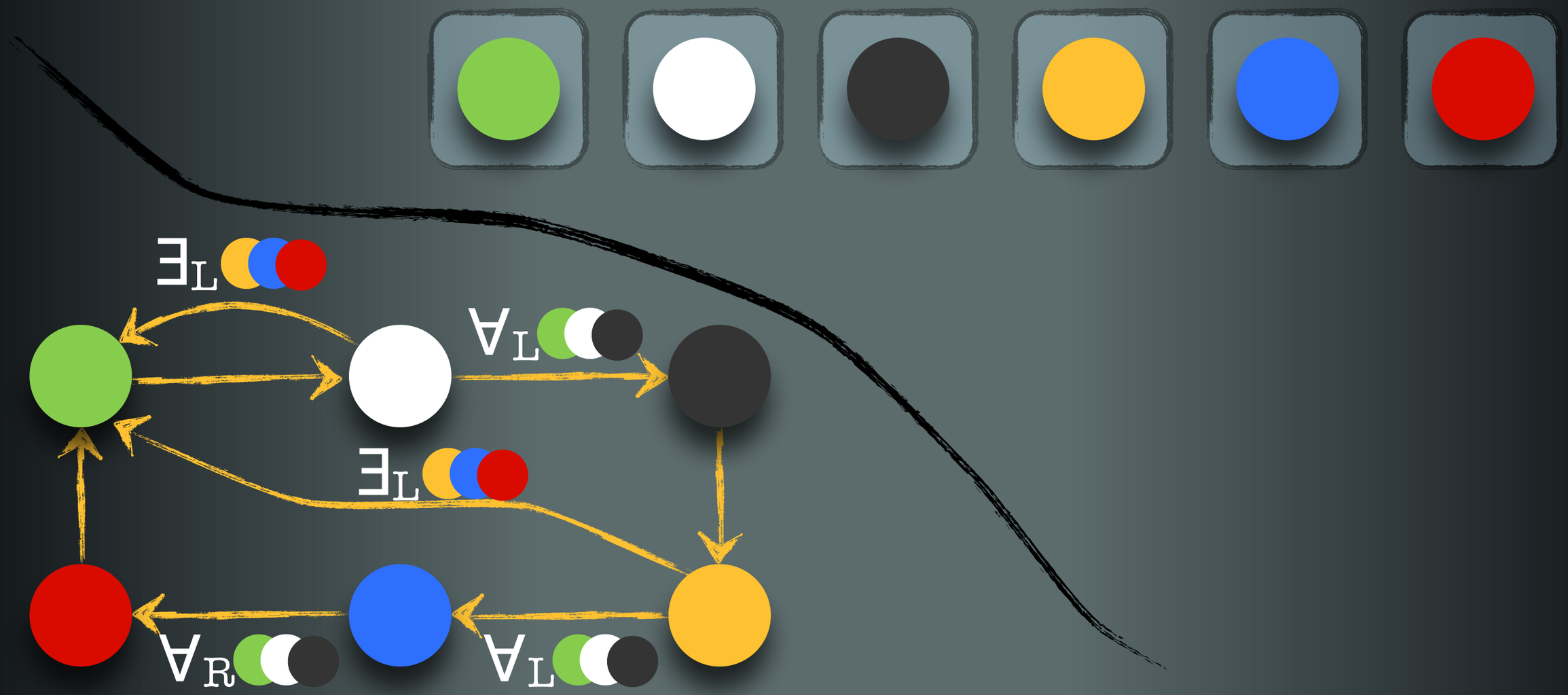


Parameter System

Analysis / Abstraction

k=1

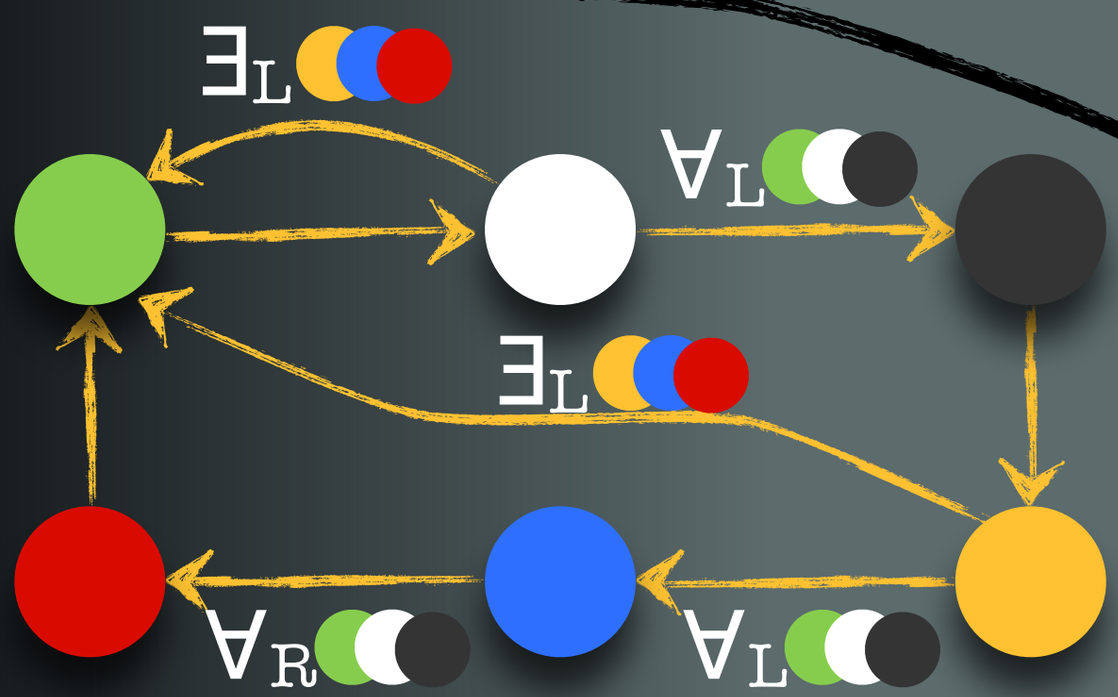
$$\mathbf{V} = \mu\mathbf{X} \,.\, \alpha_1(\mathbb{I}) \cup \alpha_1 \circ \mathrm{post} \circ \gamma_1(\mathbf{X})$$

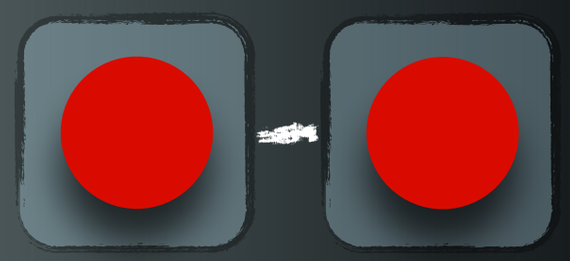


$k = 1$

if $\gamma_1(\mathbf{V}) \cap \mathbf{Bad} = \varnothing$ then return Safe

$\gamma_1(\mathbf{V})$ contains everything!

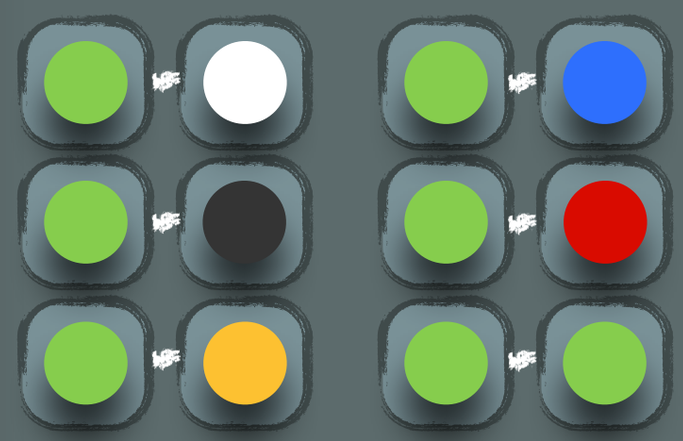




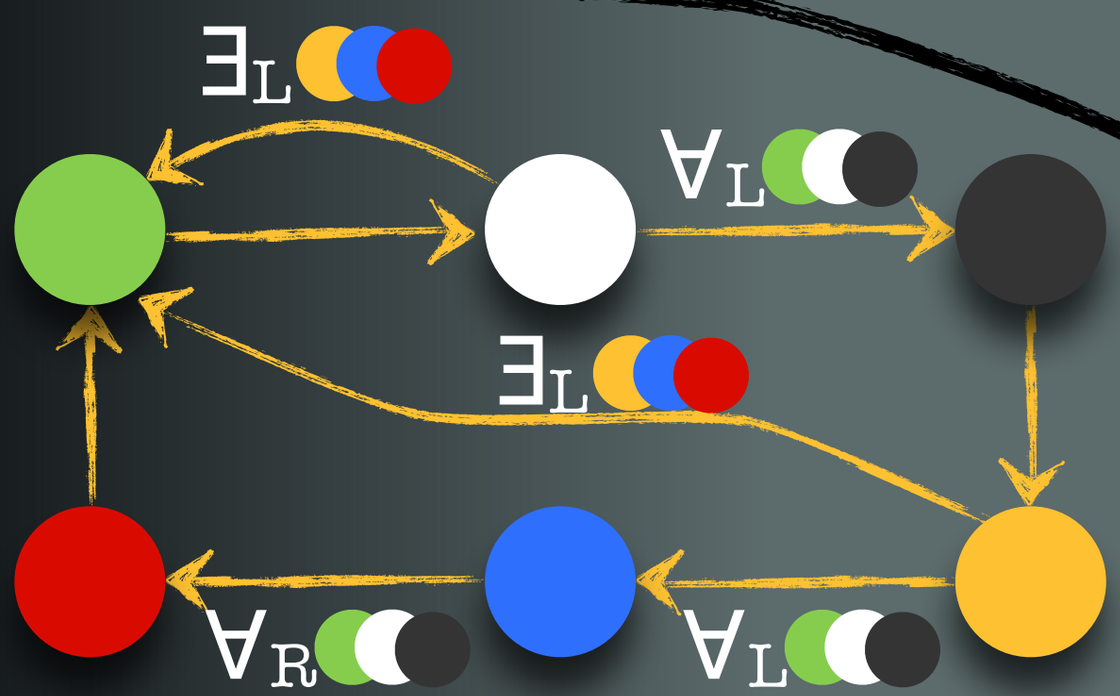
Parameter System

Analysis
Abstraction

k=2 $\mathbf{R}_2$:



...



$\mathbf{R}_2 \cap \mathbf{Bad} = \varnothing$

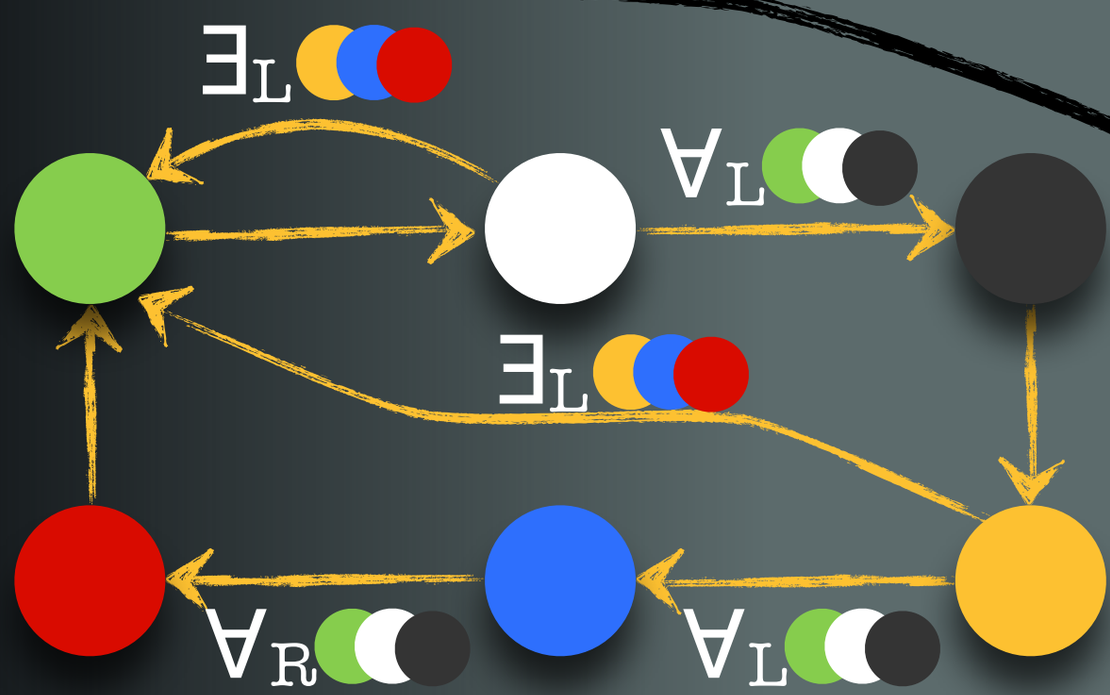
Parameter System

Analysis
Abstraction

k=2

$$\mathbf{V} = \mu\mathbf{X} \,.\, \alpha_2(\mathbb{I}) \cup \alpha_2 \circ \mathrm{post} \circ \gamma_{2|3}(\mathbf{X})$$

extensions of size 3 only



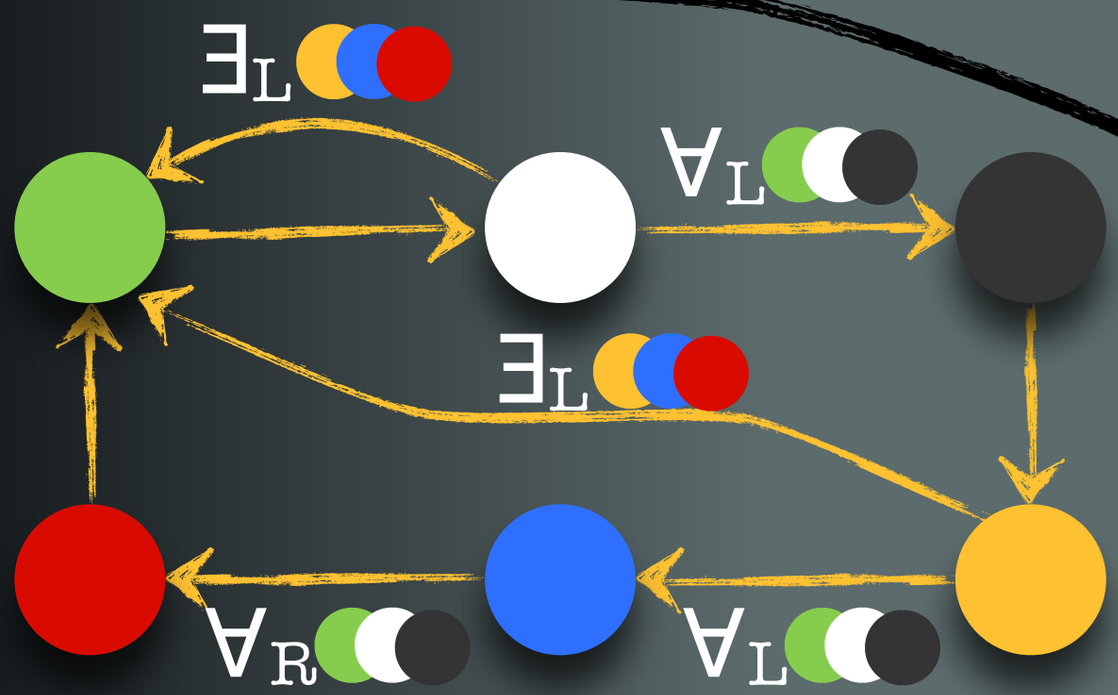
All pairs but

Parameter System

Analysis
Abstraction

k = 2

if $\gamma_2(\mathbf{V}) \cap \mathbf{Bad} = \varnothing$ then return Safe



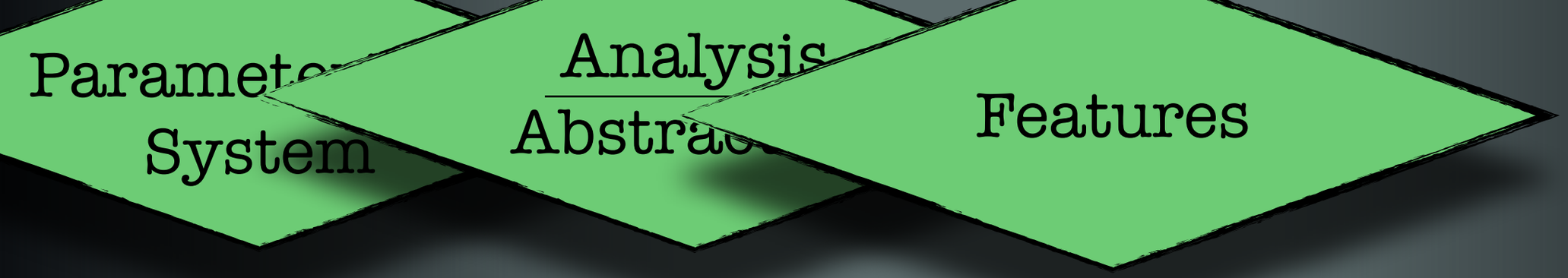
Collected all views

k=2 is cutoff point

Parameter
System

Analysis
Abstraction

Features


Parameter
System
Analysis
Abstract
Features


Universal condition
$\forall_L$


Existential condition
$\exists_R$

Parameter System

Analysis
Abstract

Features

Universal condition

Existential condition

Global variables

Broadcast

Creation/Deletion

Rendez-Vous

Parameter System

Analysis
Abstract

Features

Universal condition

Existential condition

Global variables

Broadcast

Creation/Deletion

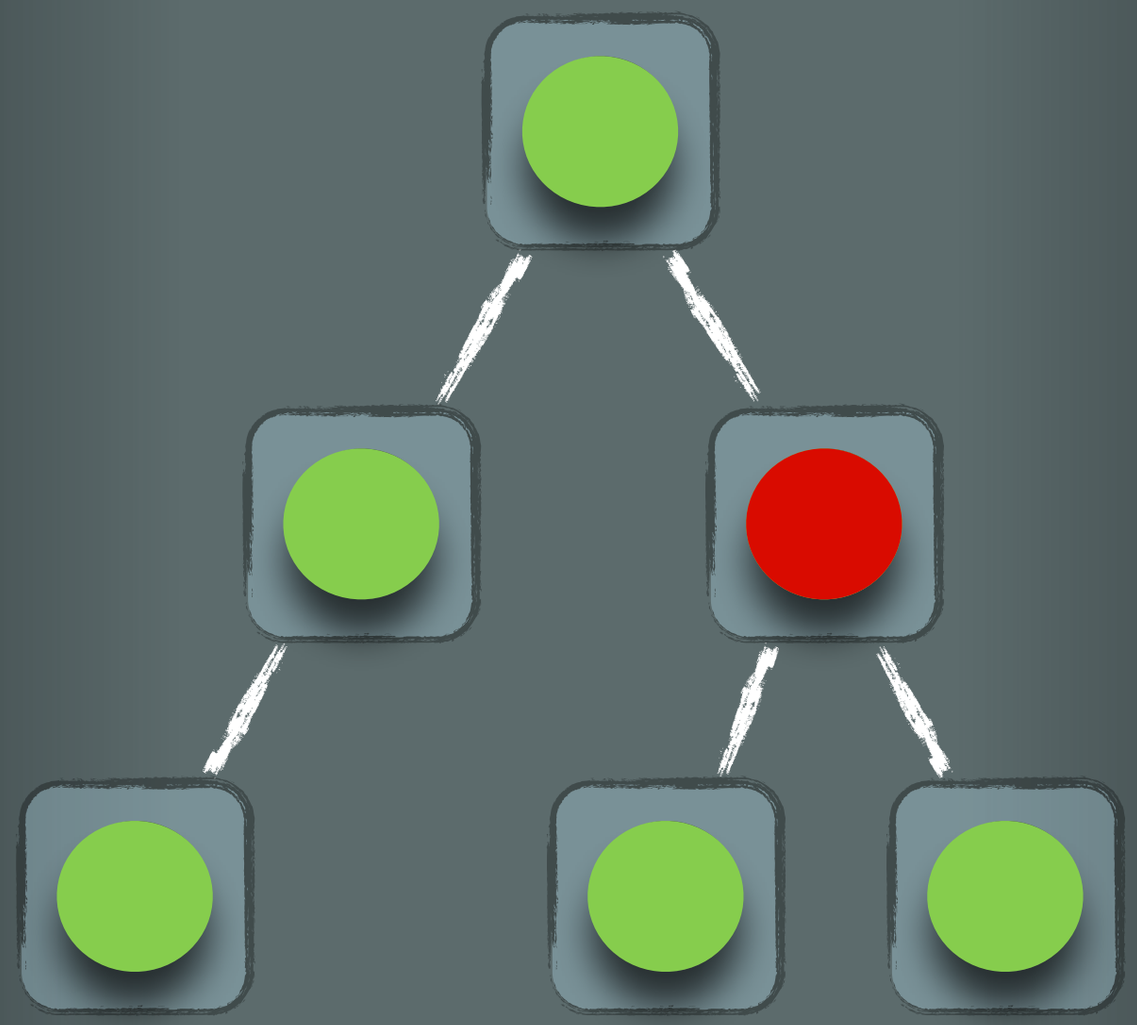
Rendez-Vous

Topologies

- Linear
- Ring
- Tree
- Multiset

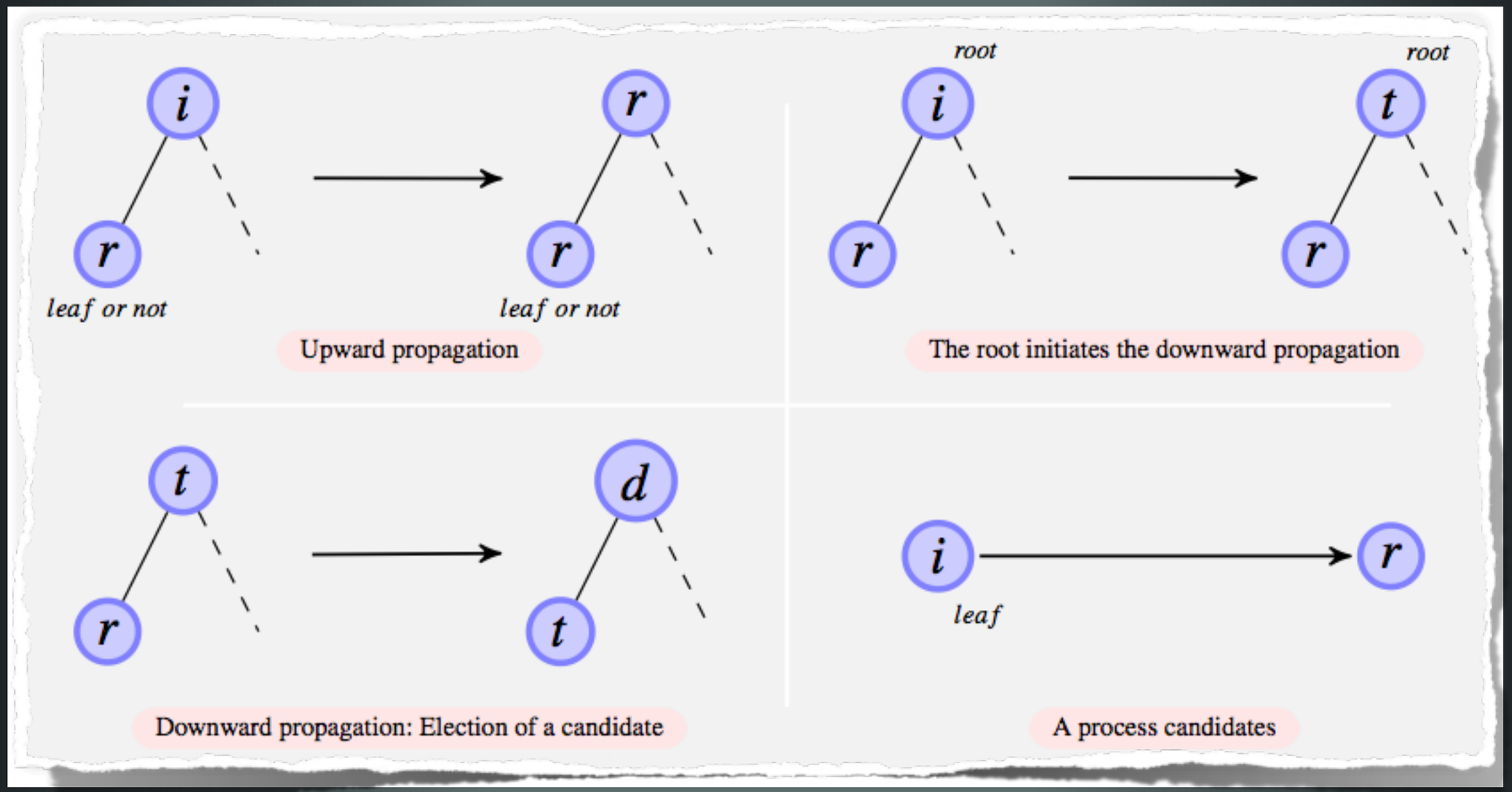
Parameterized System

Leader Election



Parameterized System

Leader Election



Parameter System

Analysis
Abstract

Features

Universal condition

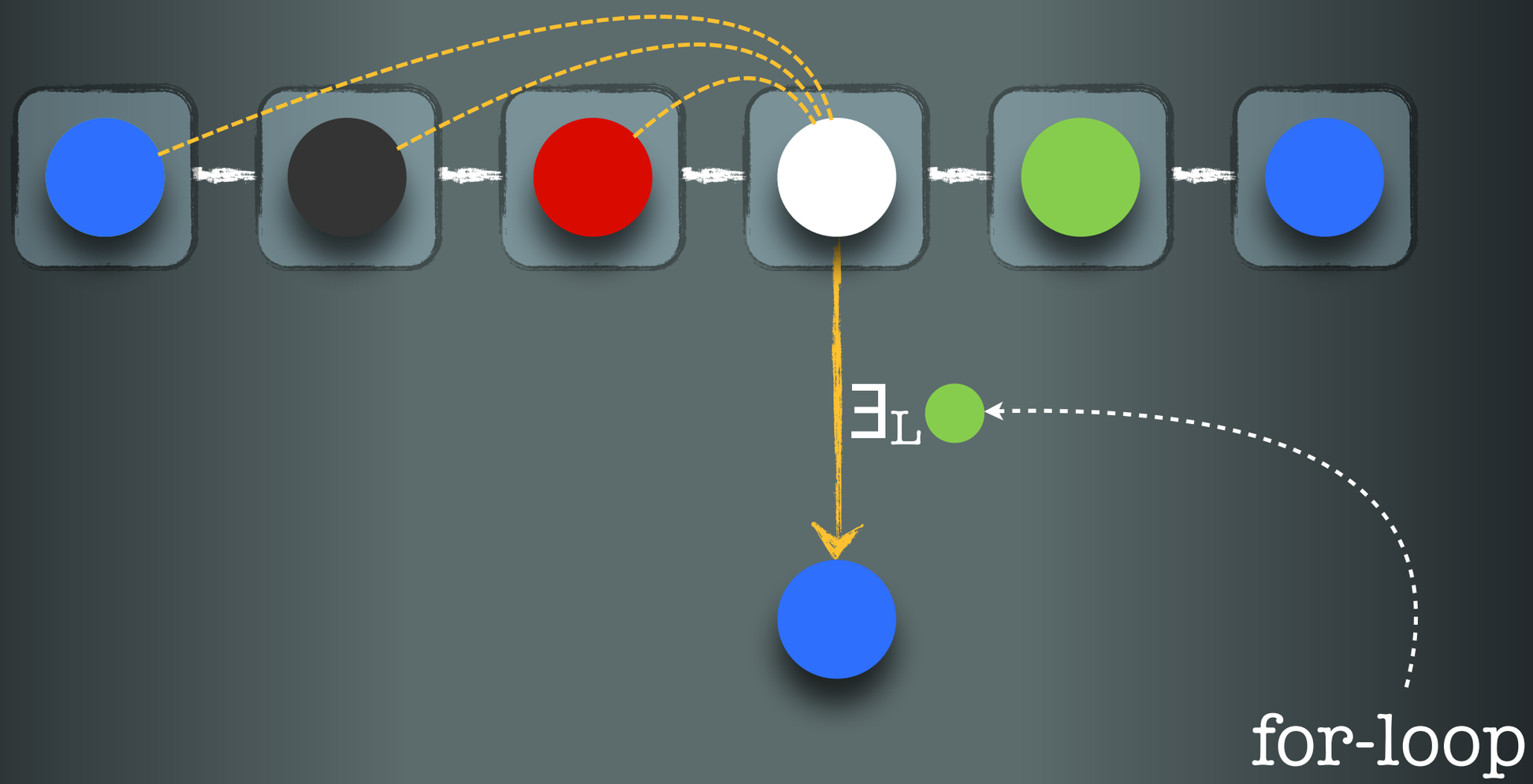
Existential condition

Non-atomic
global conditions

Parameter System

Analysis
Abstract

Features

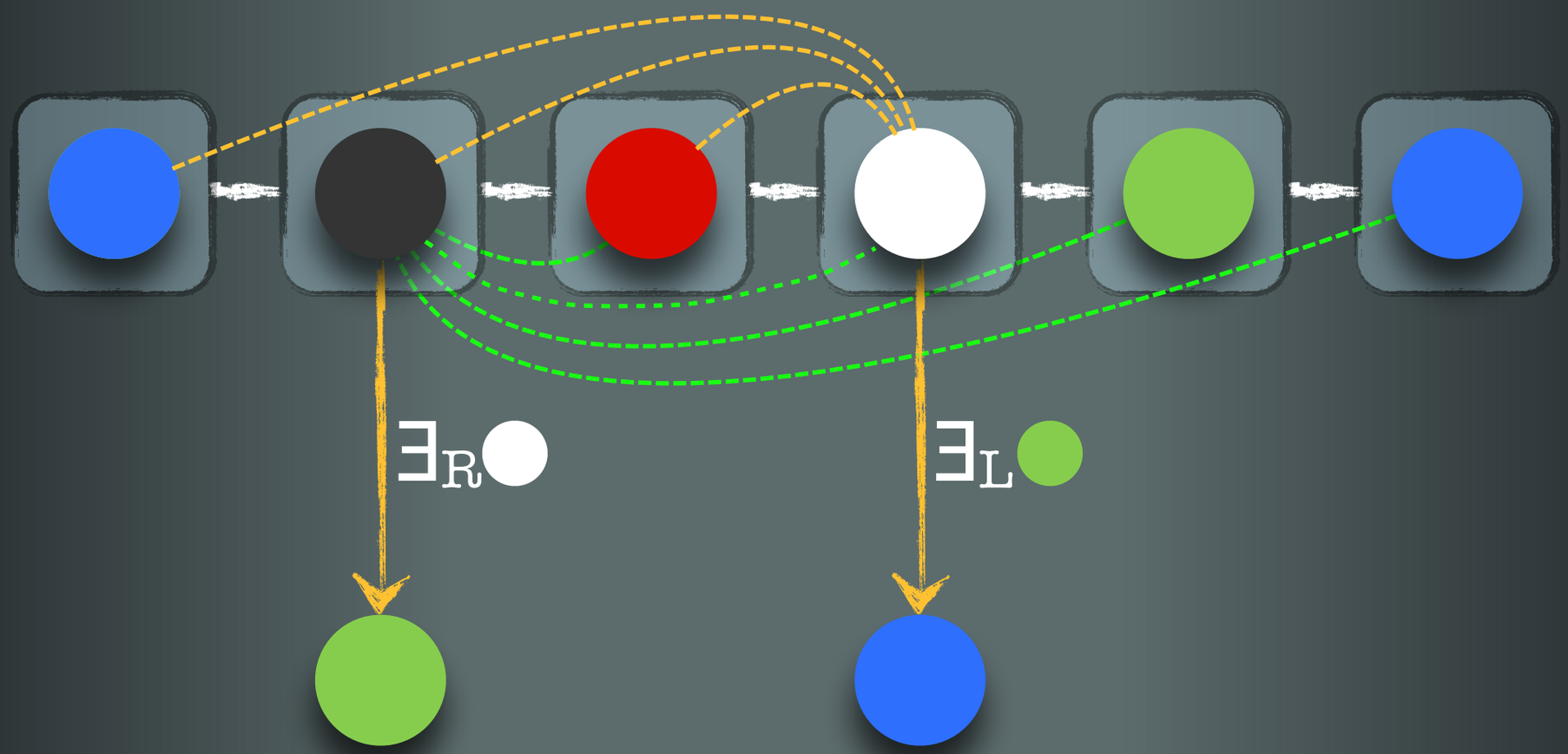


Parameter System

Analysis

Abstract

Features



Parameter System

Analysis
Abstract

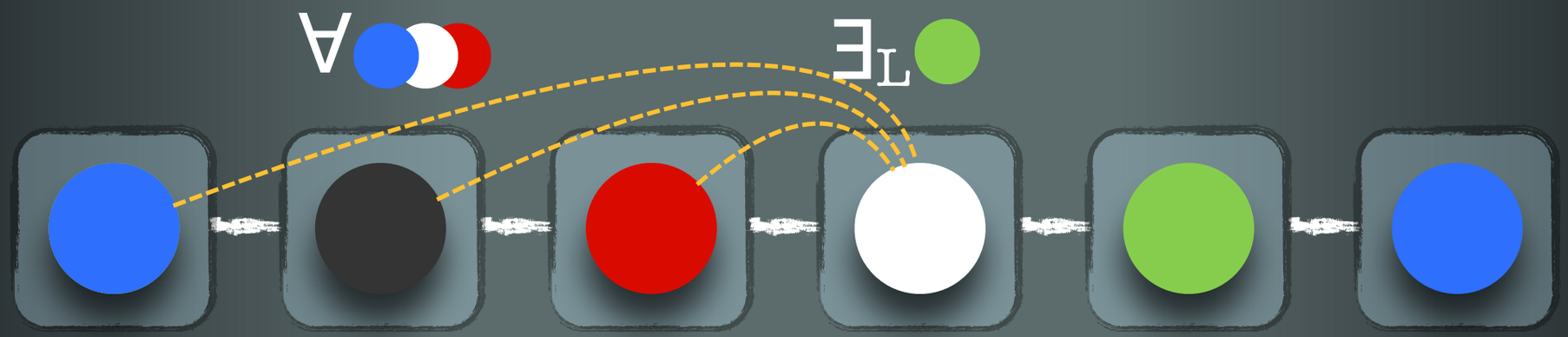
Features

Parameter System

Analysis

Abstract

Features

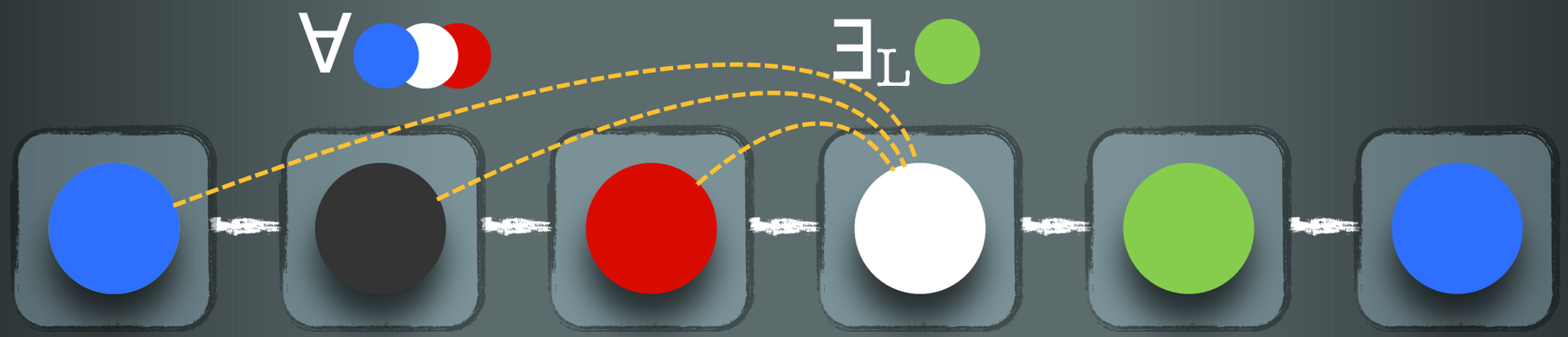


.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
?	?	✓	o	_	_
.	.	.	.	.	.
.	.	.	.	.	.

Parameter System

Analysis
Abstract

Features

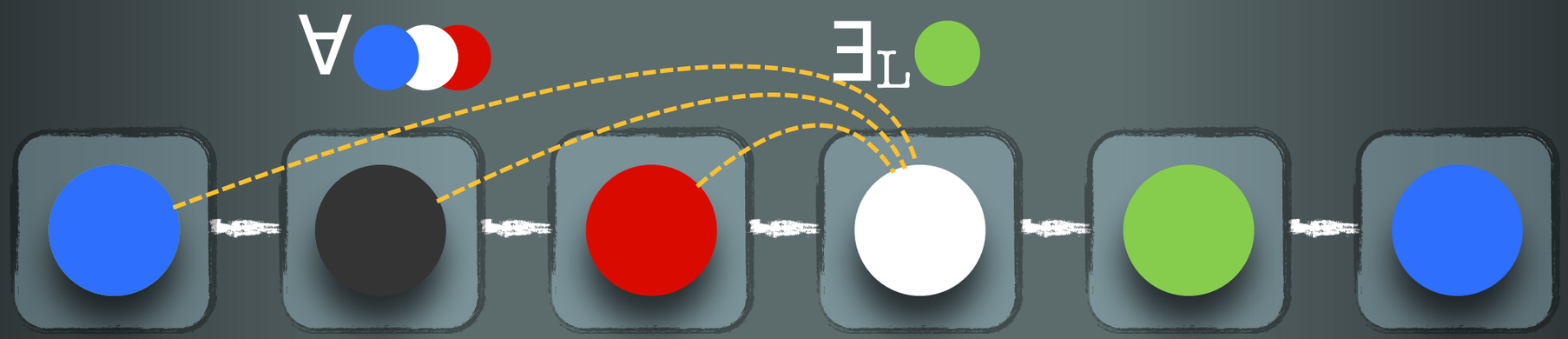


.	.	.	.	.	.
?	o	?	?	?	?
.	.	.	.	.	.
?	?	✓	o	_	_
.	.	.	.	.	.
.	.	.	.	.	.

Parameter System

Analysis
Abstract

Features



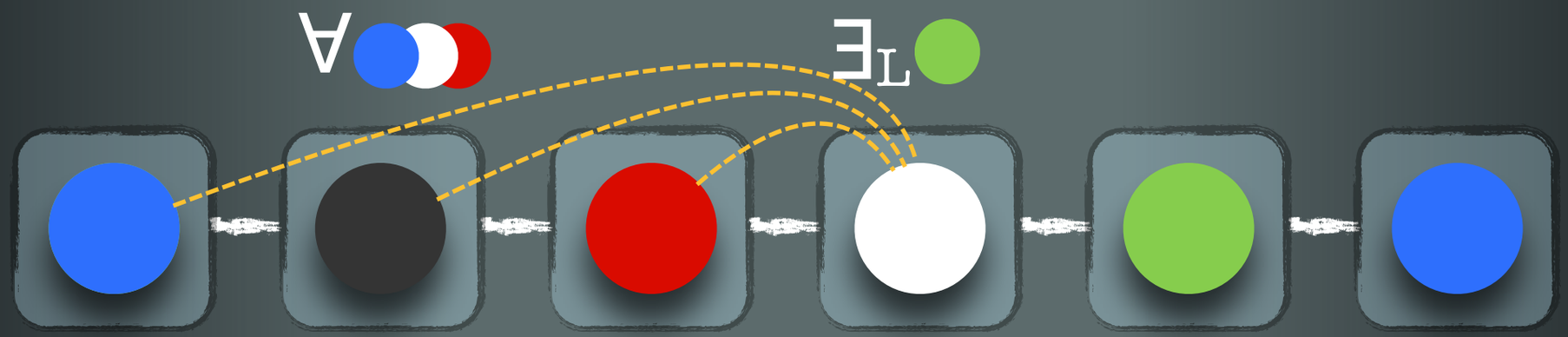
.	.	.	.	.	.
✓	o	✓	✓	✗	?
.	.	.	.	.	.
?	?	✓	o	_	_
.	.	.	.	.	.
.	.	.	.	.	.

Parameter System

Analysis

Abstract

Features



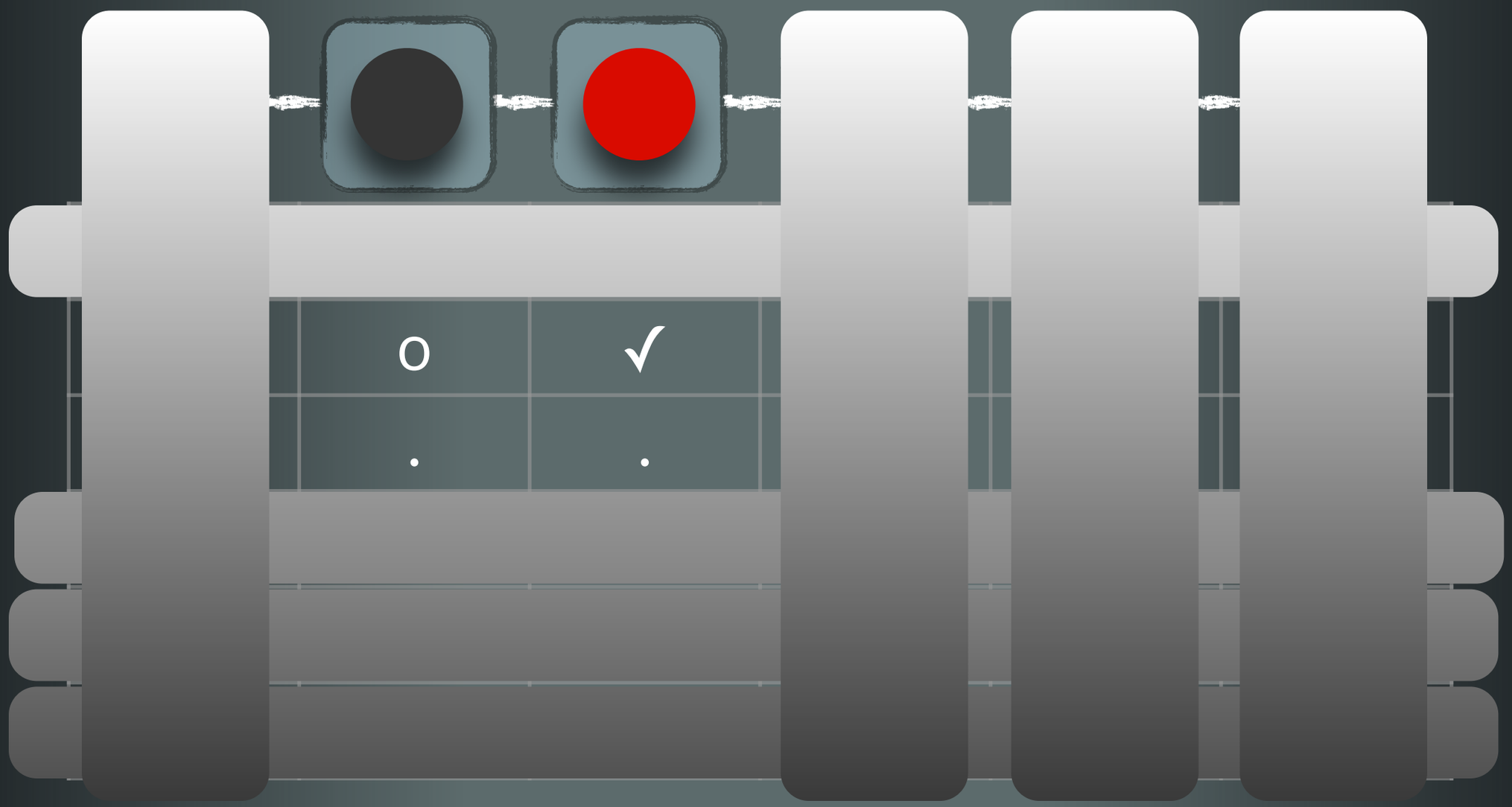
.	.	.	.	.	.
✓	o	✓	✓	✗	?
.	.	.	.	.	.
?	?	✓	o	_	_
.	.	.	.	.	.
.	.	.	.	.	.

Parameter System

Analysis

Abstract

Features



Parameter System

Analysis

Abstract

Features

o	✓
.	.

o	✓
?	o

.	.
✓	o

o	✓	✓
.	.	.
?	✓	o

Parameterized System ✓

Analysis ✓
Abstraction

Features ✓

Completeness

Paramet
System

Analysis
Abstrac

Features

Completeness

Parameteri
System

Analysis
Abstrac

Featur

Completeness

Well
Quasi-Ordering
(WQO)

$(A, \preccurlyeq)$

$a_0\ a_1\ a_2\ \ldots\ldots\ a_i\ \ldots\ldots\ \preccurlyeq\ \ldots\ldots\ a_j\ \ldots$

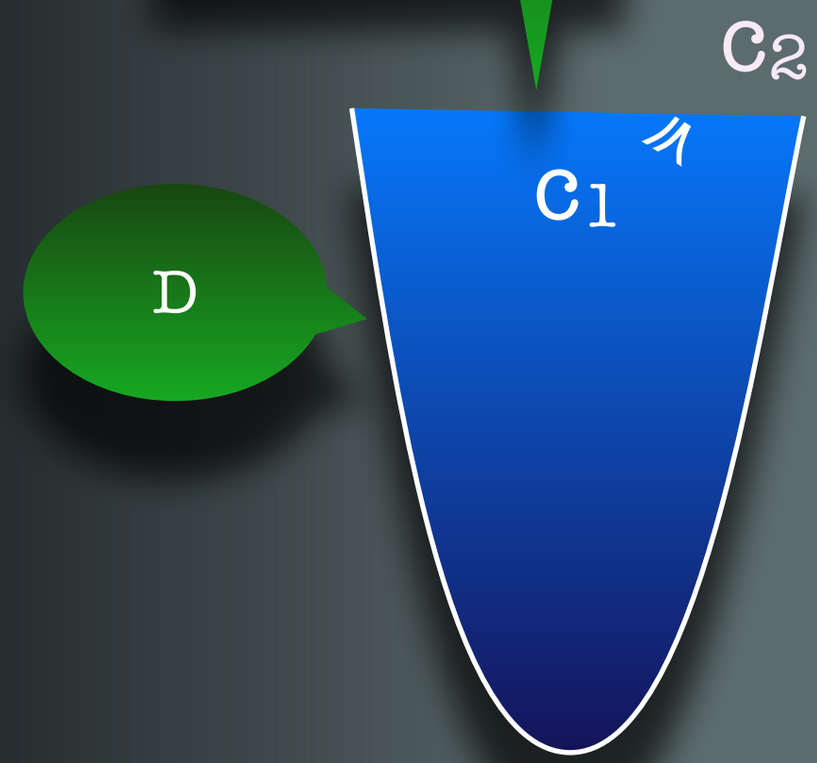
Parameteri
System

Analysis
Abstrac

Featur

Completeness

Downward-
closed set



if $c_1 \in D$ and $c_2 \preccurlyeq c_1$ then
$c_2 \in D$

Parameteri System

Analysis Abstra

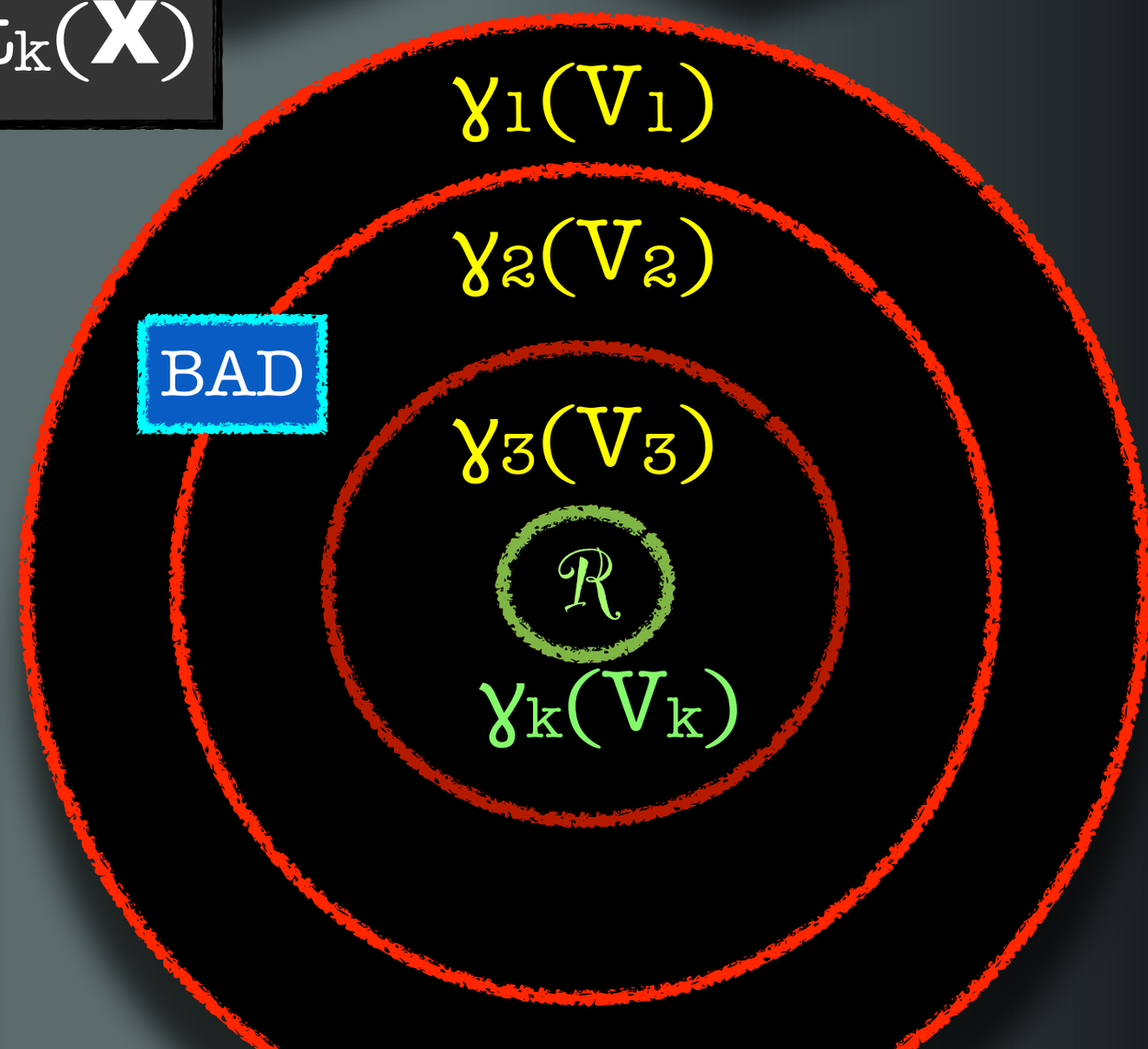
Featu

Completeness

$$V_k := \mu \mathbf{X} \,.\, \alpha_k(\mathbb{I}) \cup \mathrm{Apost}_k(\mathbf{X})$$

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

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



if

- $\mathcal{D}$ downward-closed
- $\mathcal{D}$ inductive

then

- $\mathcal{D} \supseteq \gamma_k(V_k)$ for some k

$\mathcal{R} = \gamma_k(V_k)$ for some k

Parameteri
System

Analysis
Abstrac

Featur

Completeness

A system $(\mathcal{C}, \longrightarrow)$ is WQO w.r.t a WQO $\preccurlyeq$

if $\longrightarrow$ is monotonic

$$
\begin{array}{ccc}
c_2 & \longrightarrow & c_2' \\
\preccurlyeq & & \preccurlyeq \\
c_1 & \longrightarrow & c_1'
\end{array}
$$

Parameteri
System

Analysis
Abstrac

Featur

Completeness

A system $(\mathscr{C}, \longrightarrow)$ is WQO w.r.t a WQO $\preccurlyeq$

if $\longrightarrow$ is monotonic

Class of

- Lossy channel systems
- Petri Nets
- Parameterized Systems (no $\forall$)
- ...

Parameteri System

Analysis Abstrac

Featur

Completeness

Theorem if System is WQO and Safe

Procedure guaranteed to terminate

Procedure is complete

Applications: Petri Nets

Parameterized System ✓

Analysis / Abstraction ✓

Features ✓

Completeness ✓

Parameter System

Analysis
Abstract

Feat

Completeness

	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

Future Challenges

Shape Analysis

Future Work

Contexts

Shape analysis

CEGAR

Future Work

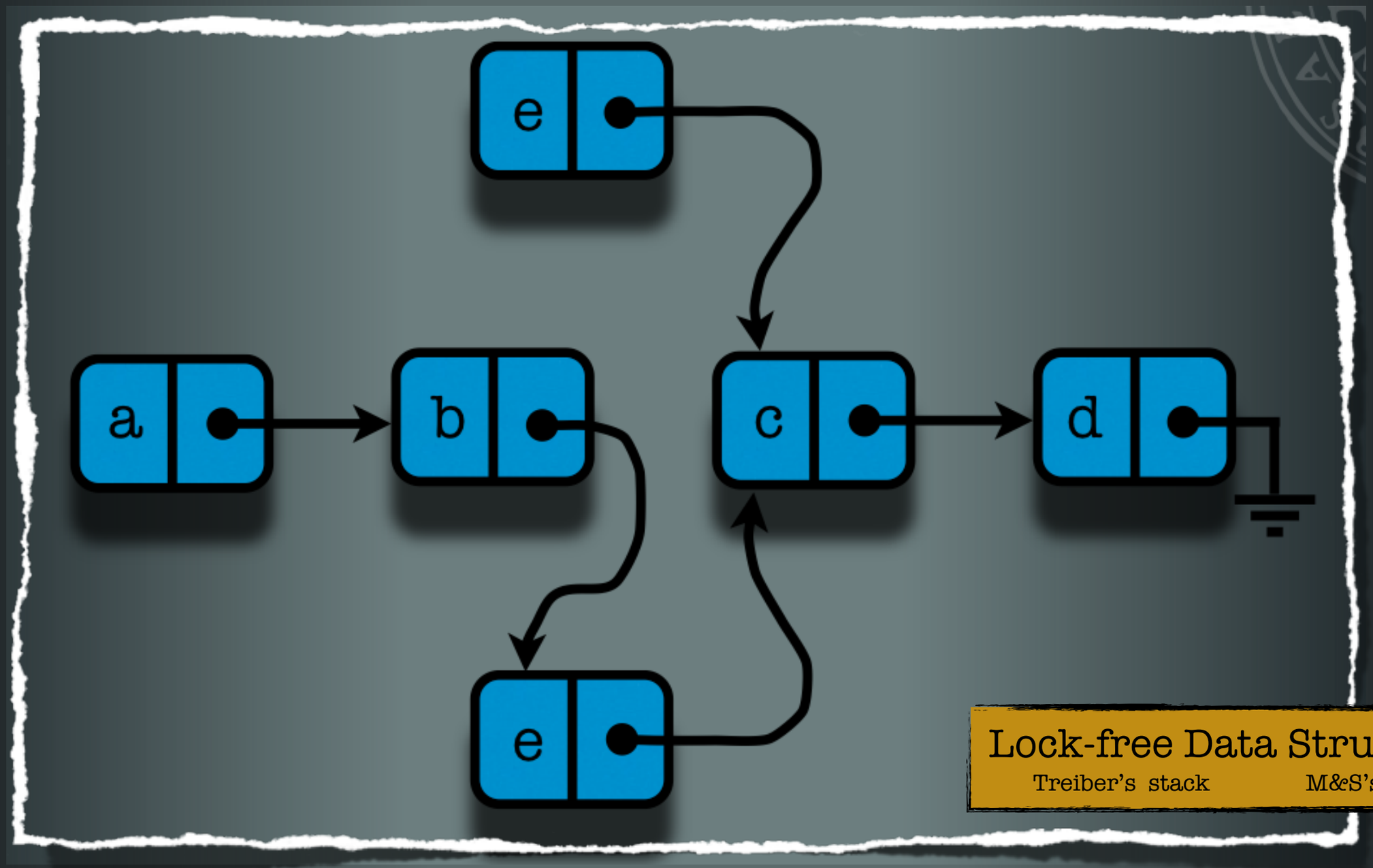
Parameteri

Systemi

ontexts

GAR

Shape analysis



Lock-free Data Structures

Treiber's stack

M&S's queue

Future Work

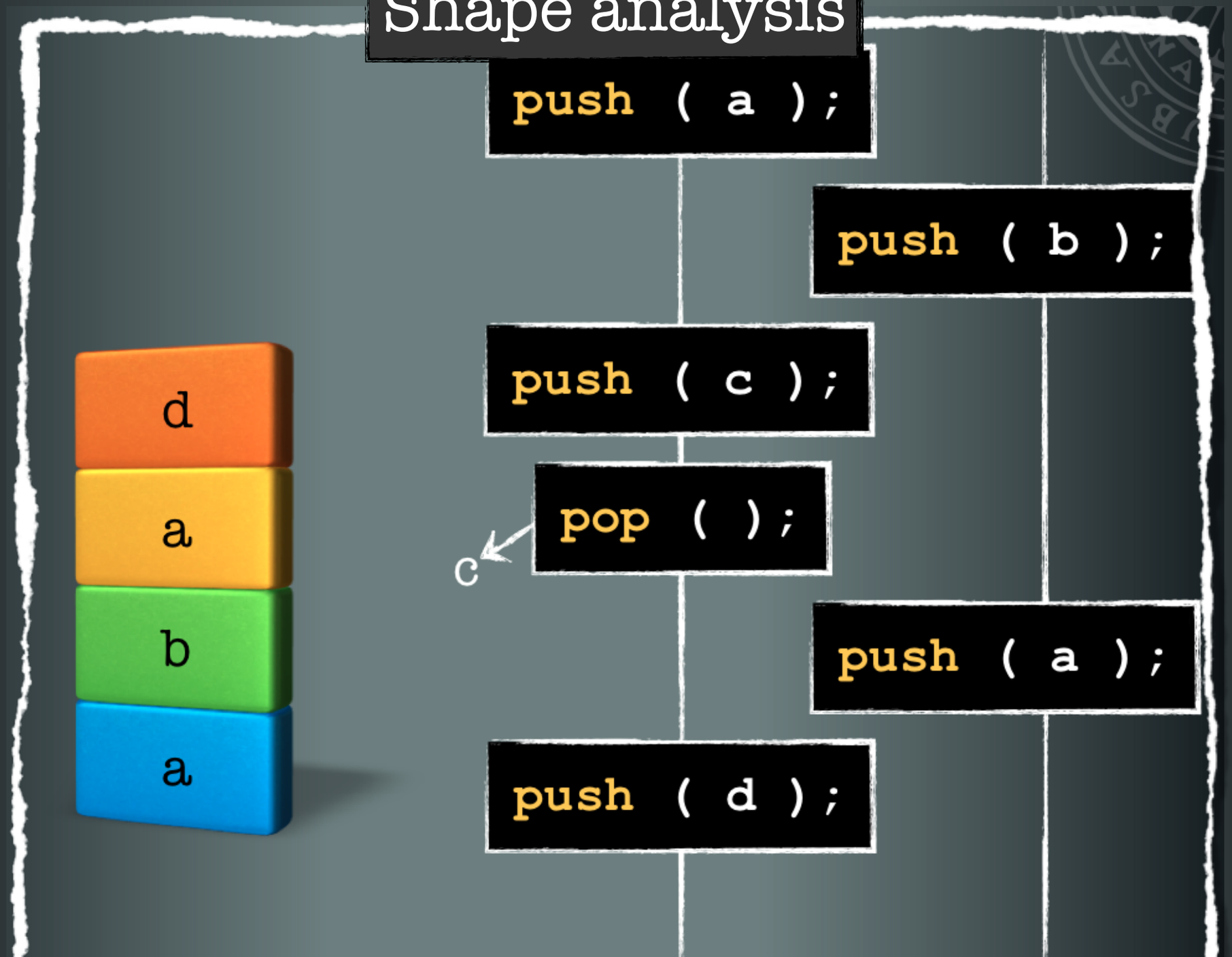
Parameteri

Systemi

ontexts

GAR

Shape analysis



Future Work

Parameteri

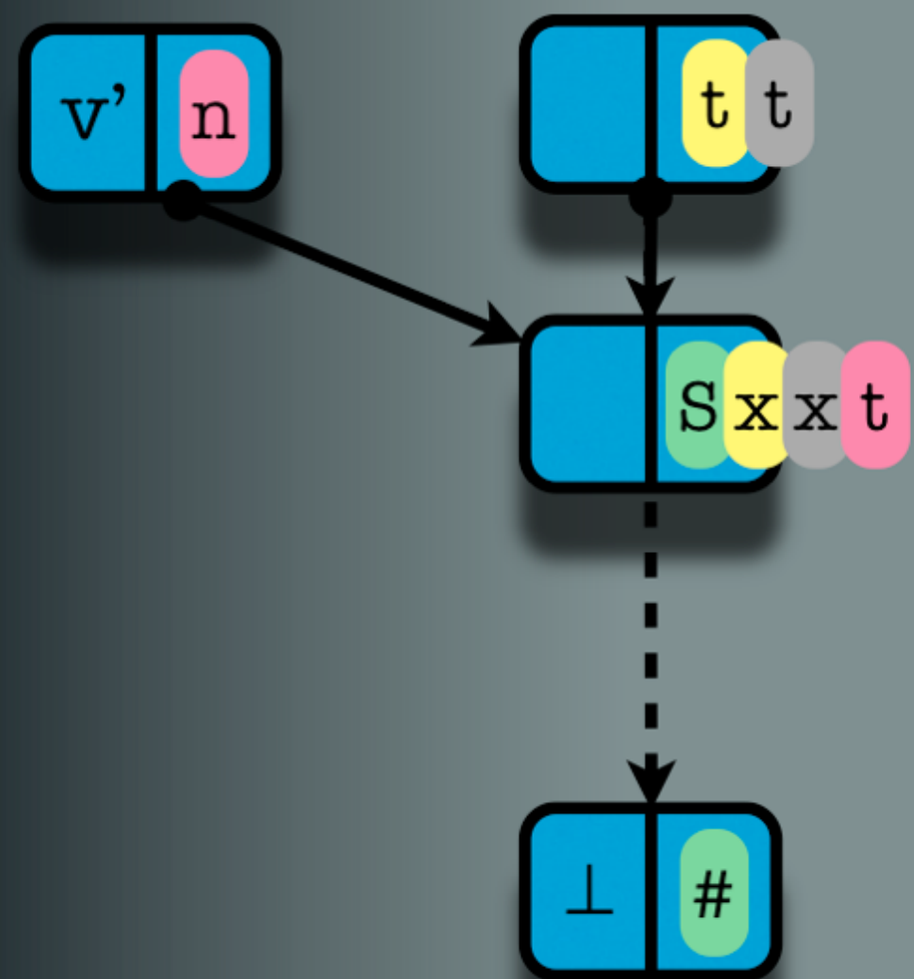
Systemi

ontexts

GAR

Shape analysis

```
t := S
n.next := t
CAS( S, t, n)
```




```
t := S
x := t.next
CAS( S, t, x)
```

```
t := S
x := t.next
CAS( S, t, x)
```

Future Work

Contexts

Shape analysis

CEGAR

Parameteri
System

Analysis
Abstractio

Feat

Completeness

Future Challenges

Shape Analysis

Infinite-State Processes

Unbounded Data
Structures

Weak Memory
Models


VERITAS
GRATIA
NATURAE
ACADEMIAE
REGIAE
SCIENTIARUM
UPSALENSIS