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

Syste

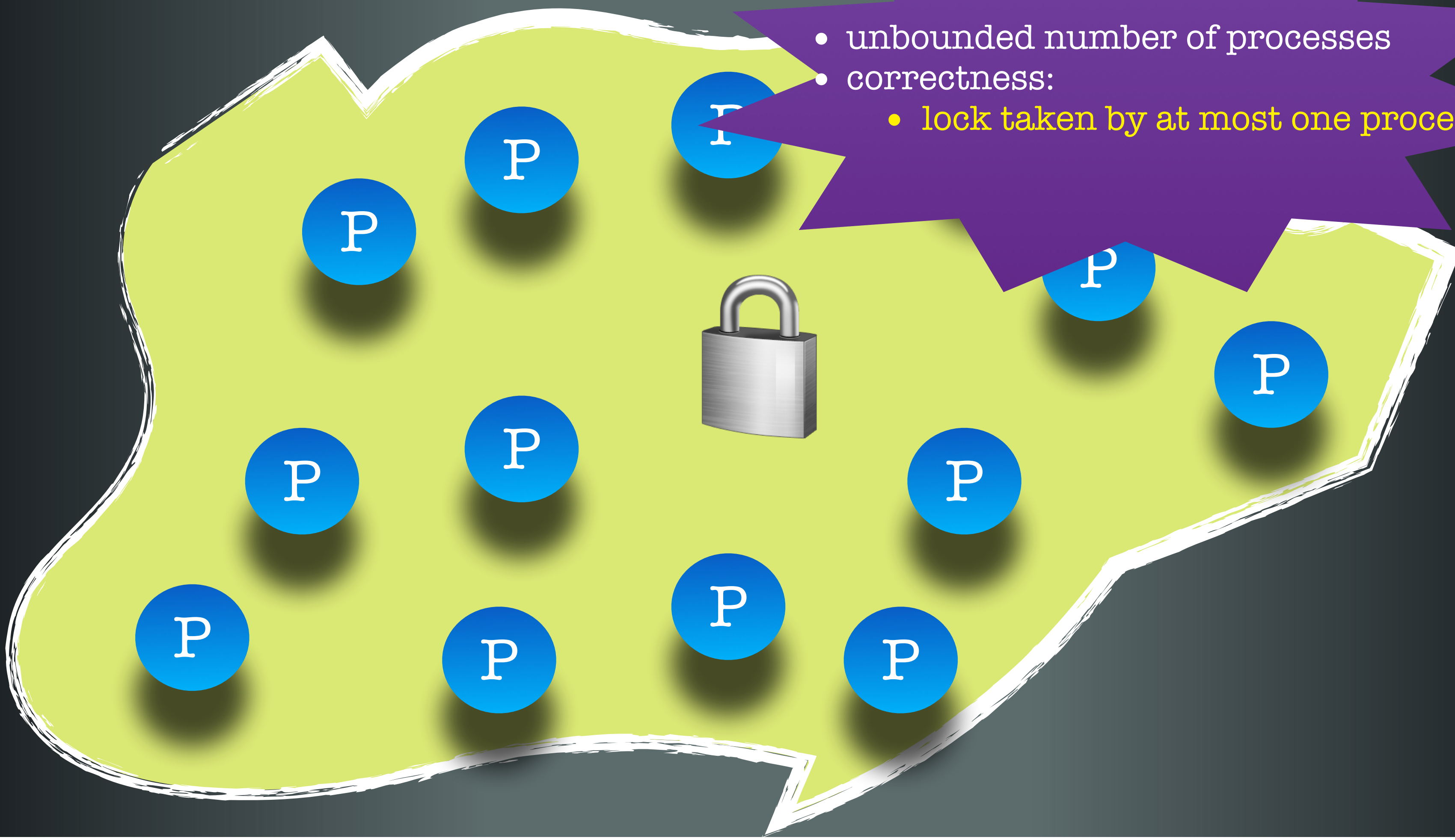
Unbounded Number of Processes

Parameterized Syste

Unbounded Numbe Processes

Mutual Exclusion Protocols

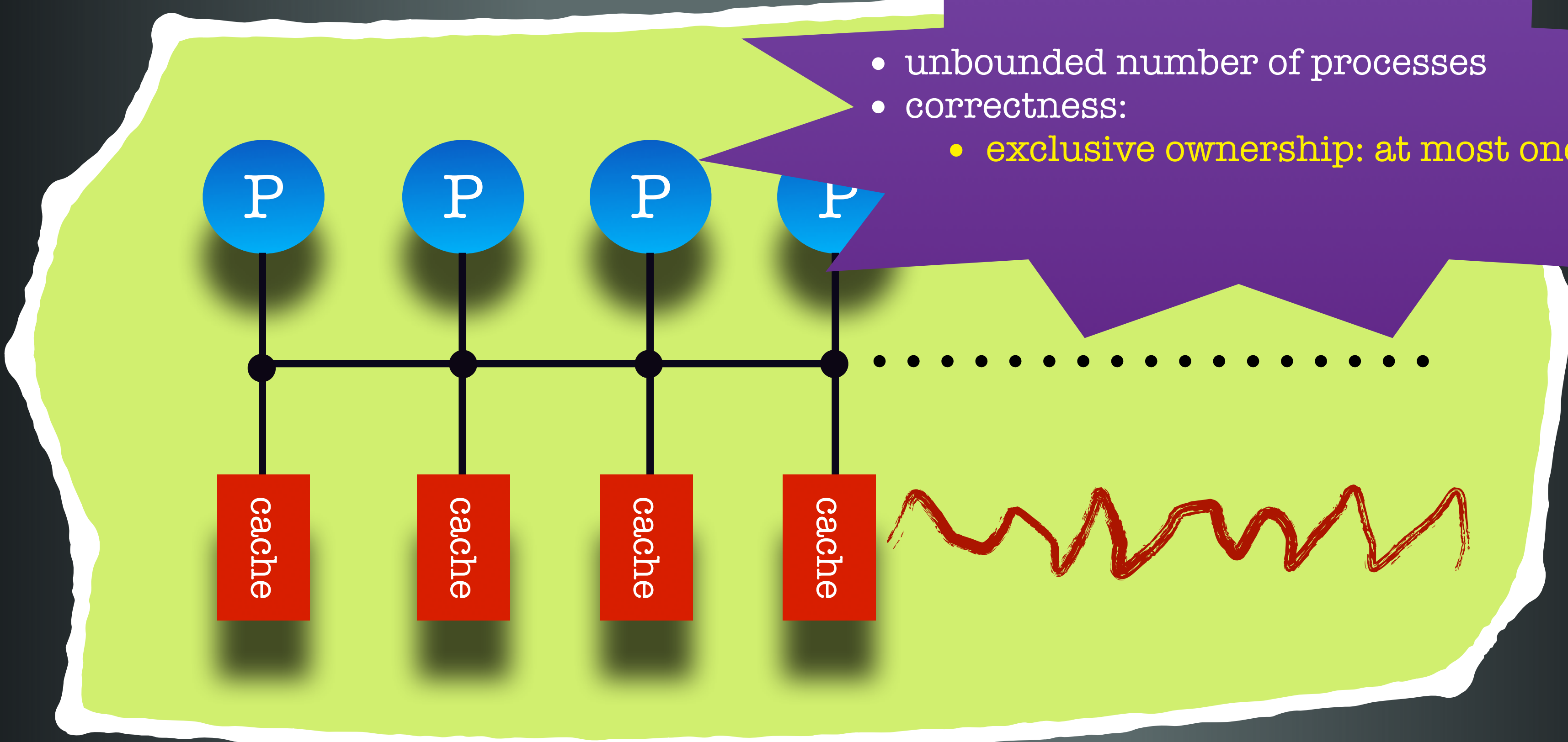
- unbounded number of processes
- correctness:
  - lock taken by at most one process



Parameterized Systems

Unbounded Number of Processes

Cache Coherence Protocol



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

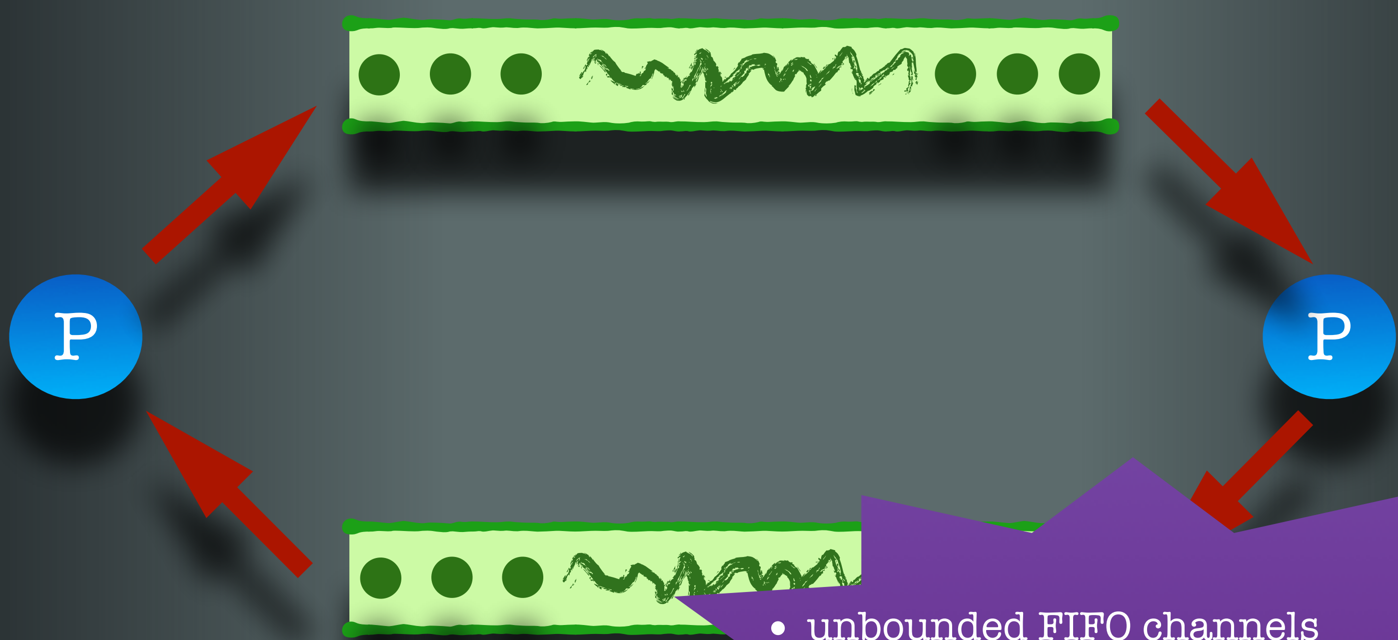
Parameterized
Syste

Unbounded
Data
Structures

Parameterized Syste

Unbounded Data Structures

Unbounded Channels

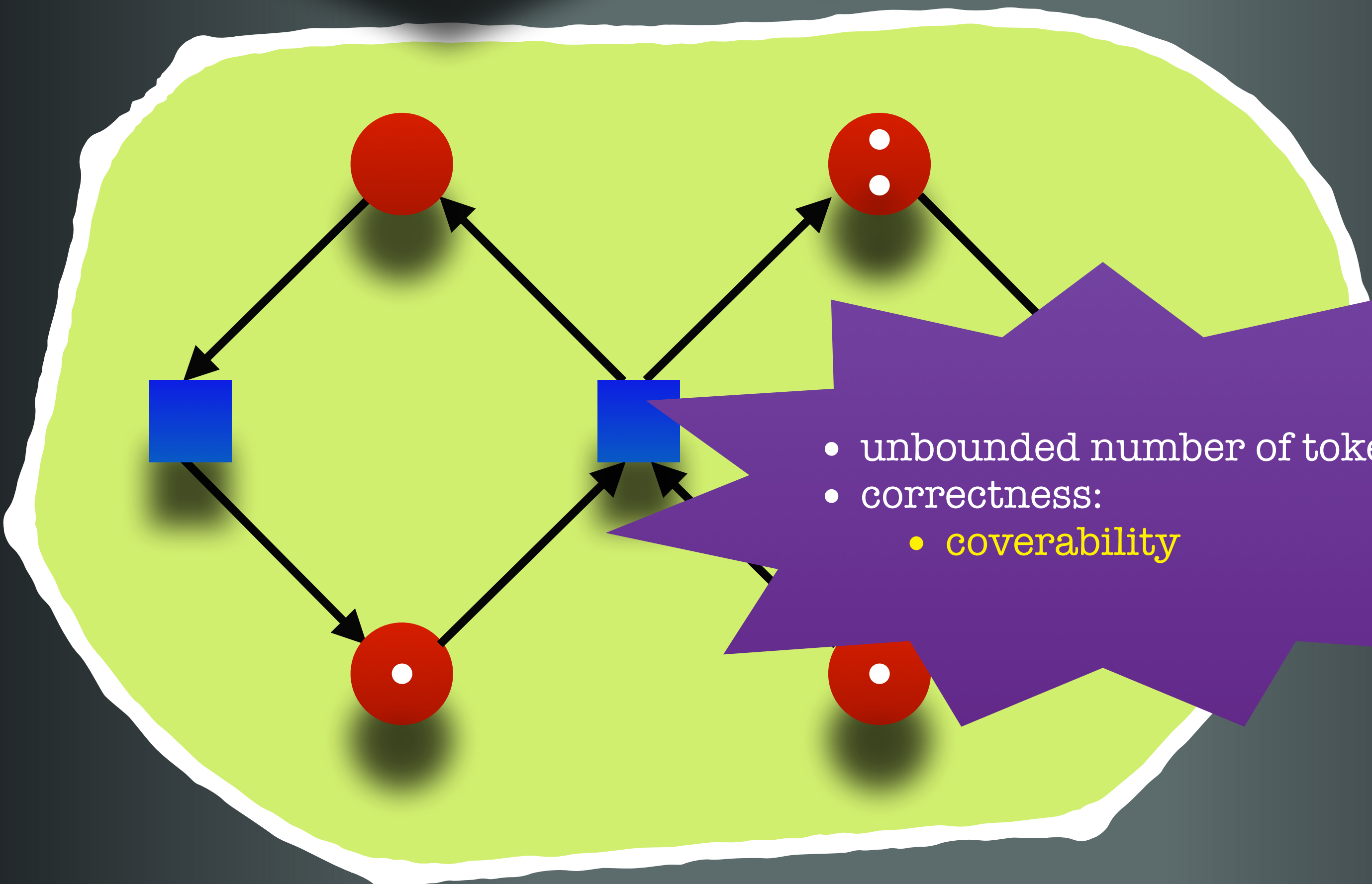


- unbounded FIFO channels
- correctness:
  - regardless of channels size

Parameterized Systems

Unbounded Data Structures

Petri Nets

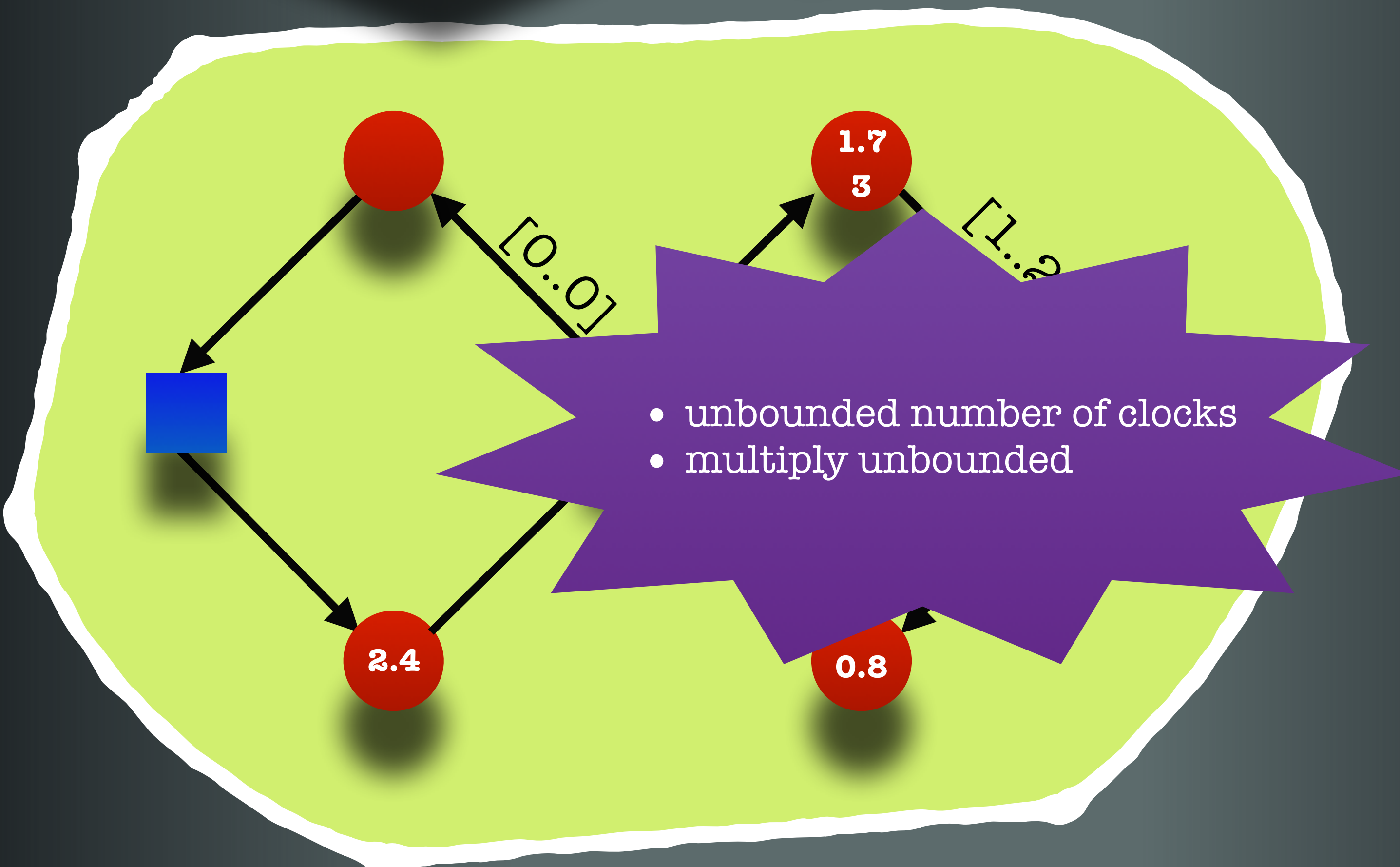


- unbounded number of tokens
- correctness:
    - coverability

Parameterized Syste

Unbounded Data Structures

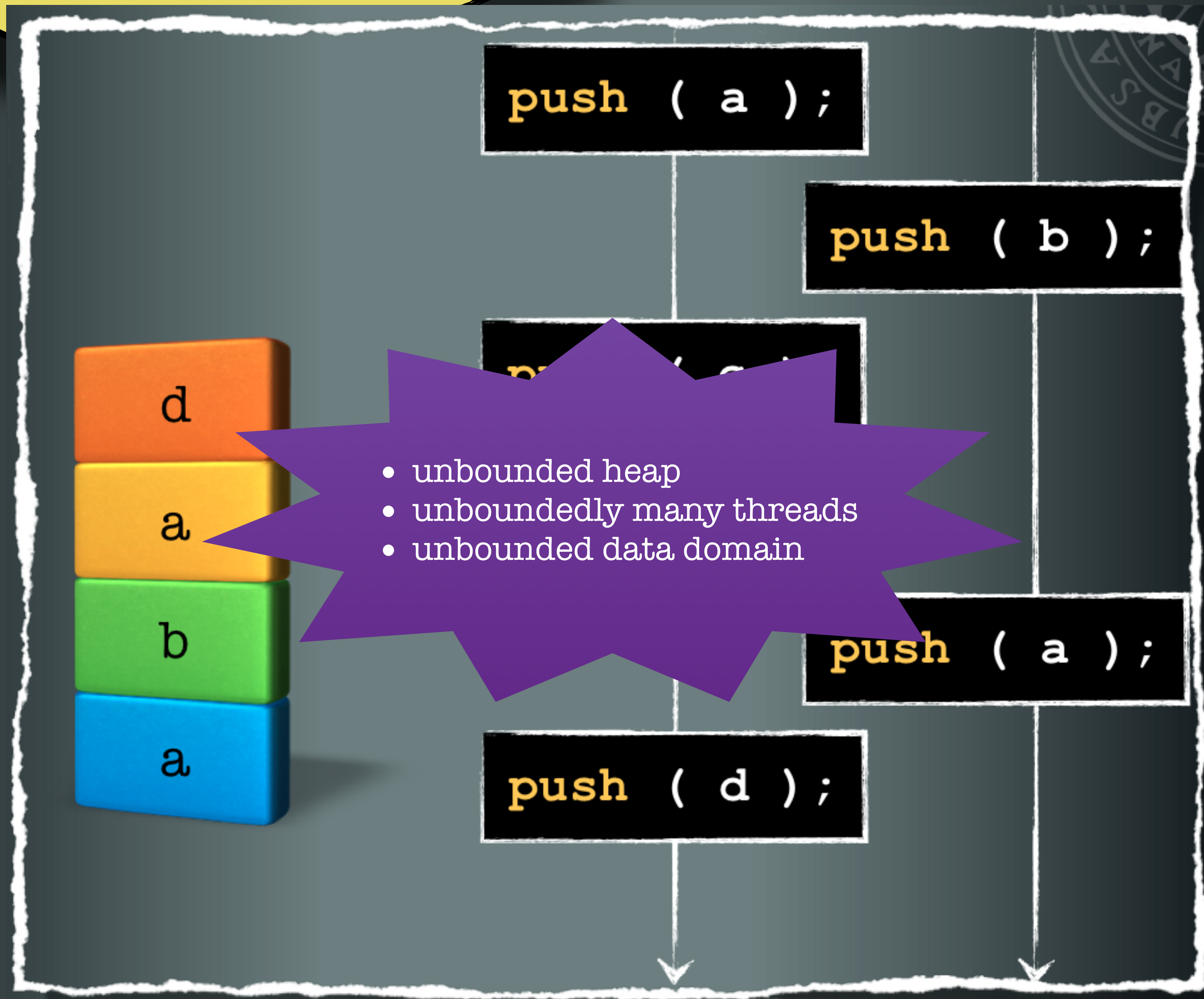
Petri Nets



- unbounded number of clocks
- multiply unbounded

Parameterized Syste

Unbounded Data Structures



Paramete
Sy

Unbounded
Number of
Processes

Parameter Sy

Unbounded Number of Processes

Hierarchy

Processes:

- finite-state
- infinite-state
- dynamic

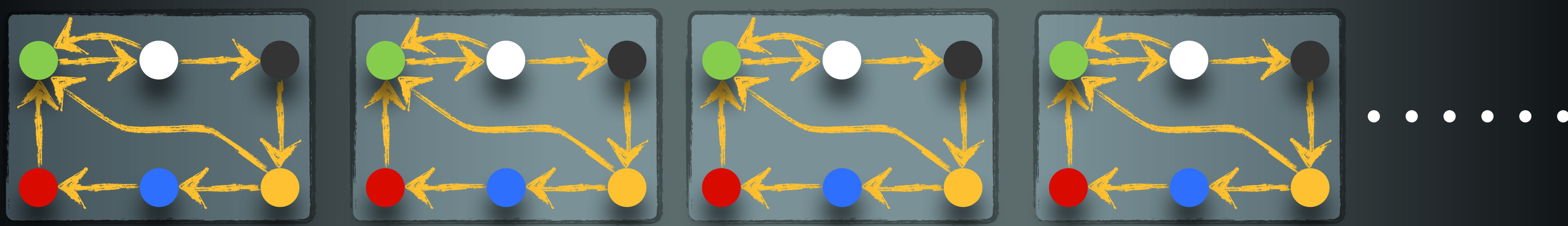
Topology:

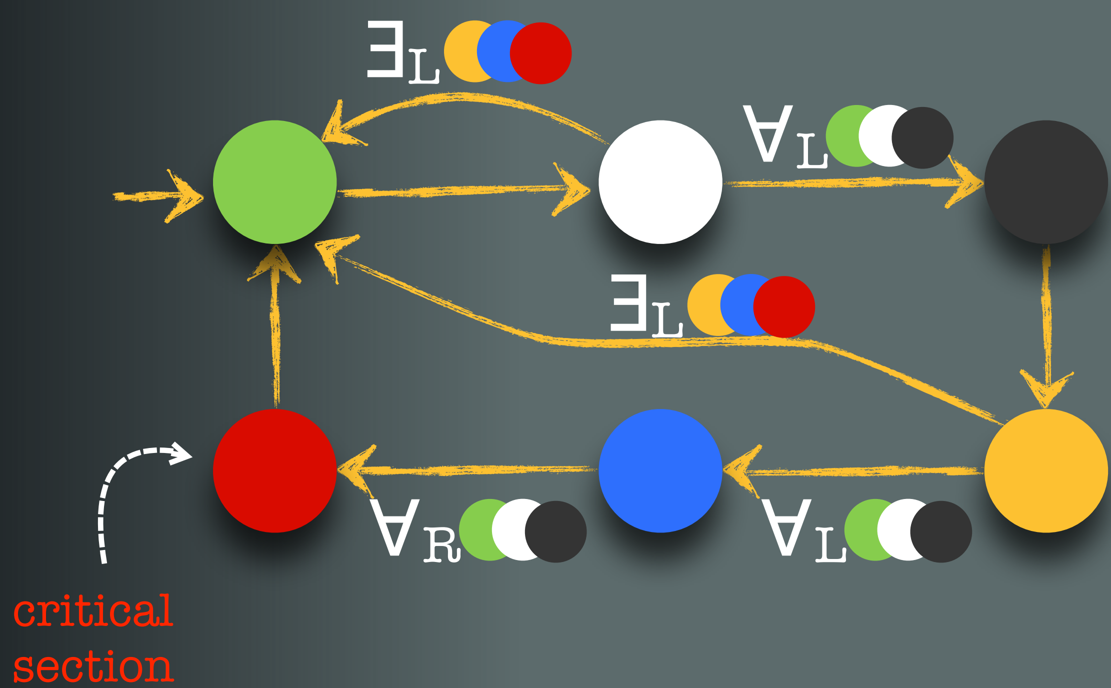
- array
- tree
- graph
- multiset

Communication:

- rendez-vous
- broadcast
- global

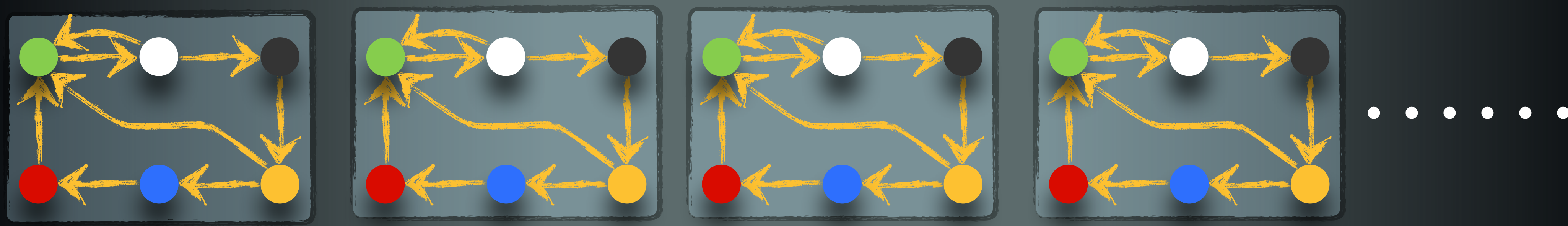
Parameterized System





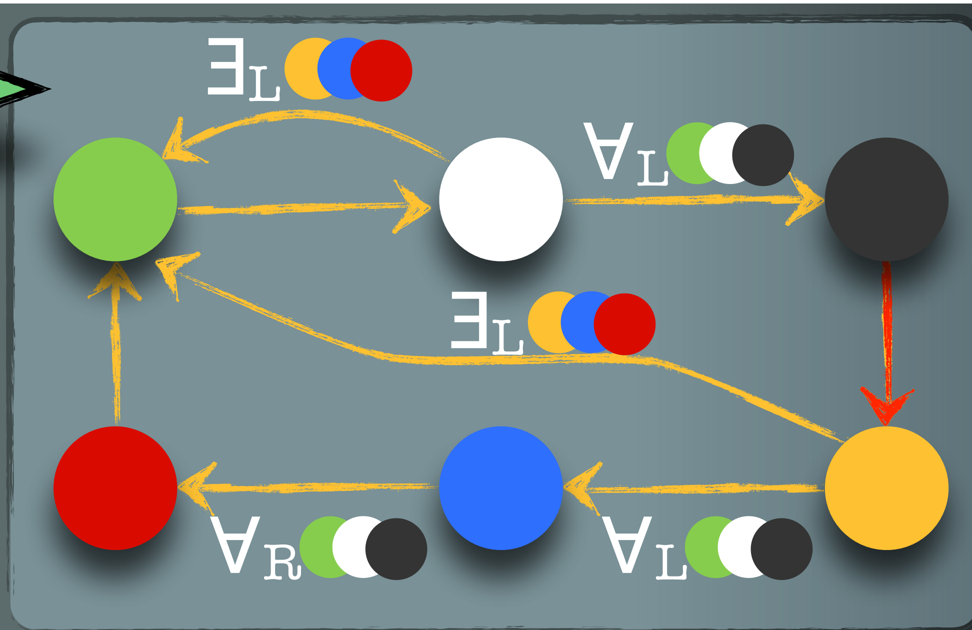
Burns' Mutual Exclusion Protocol

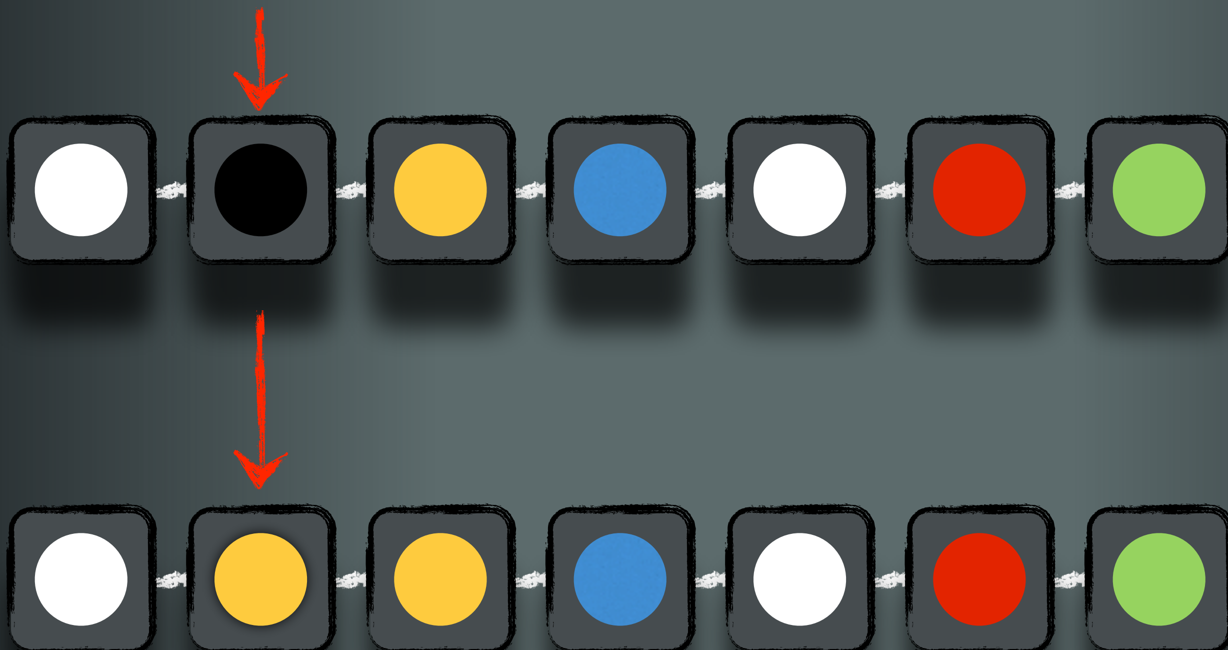
Parameterized System



Configuration

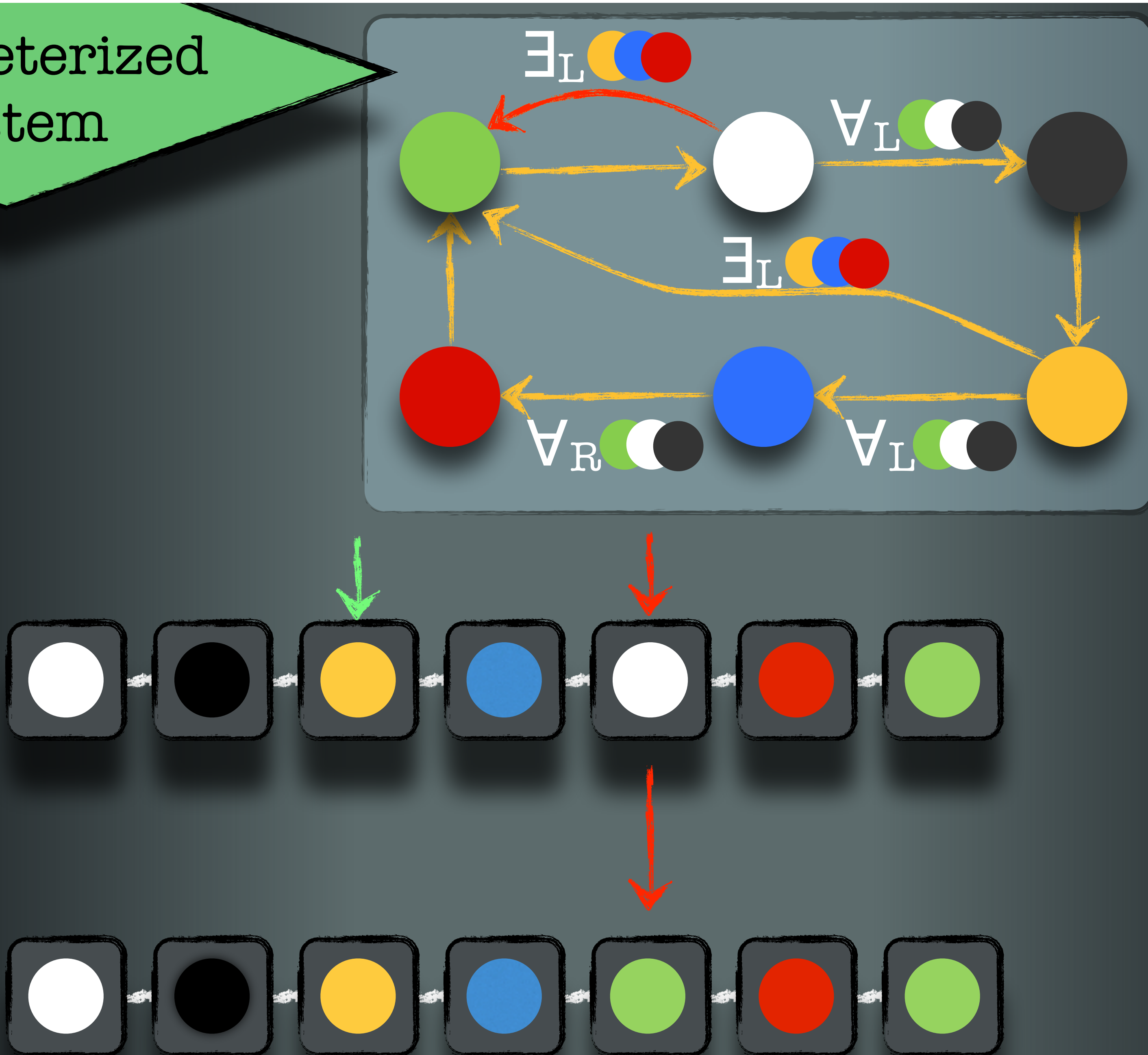
Parameterized System





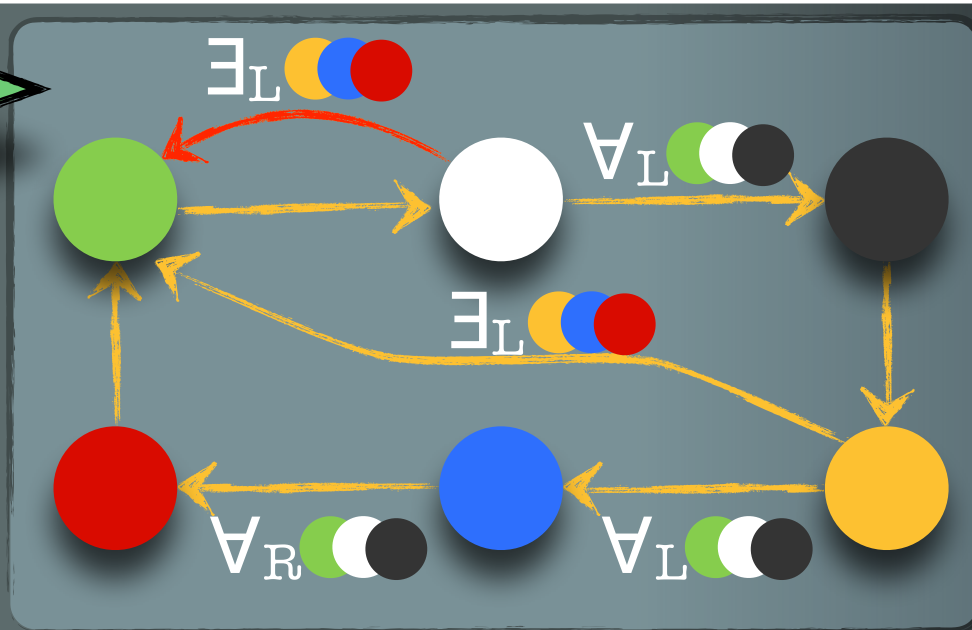
Local Transition

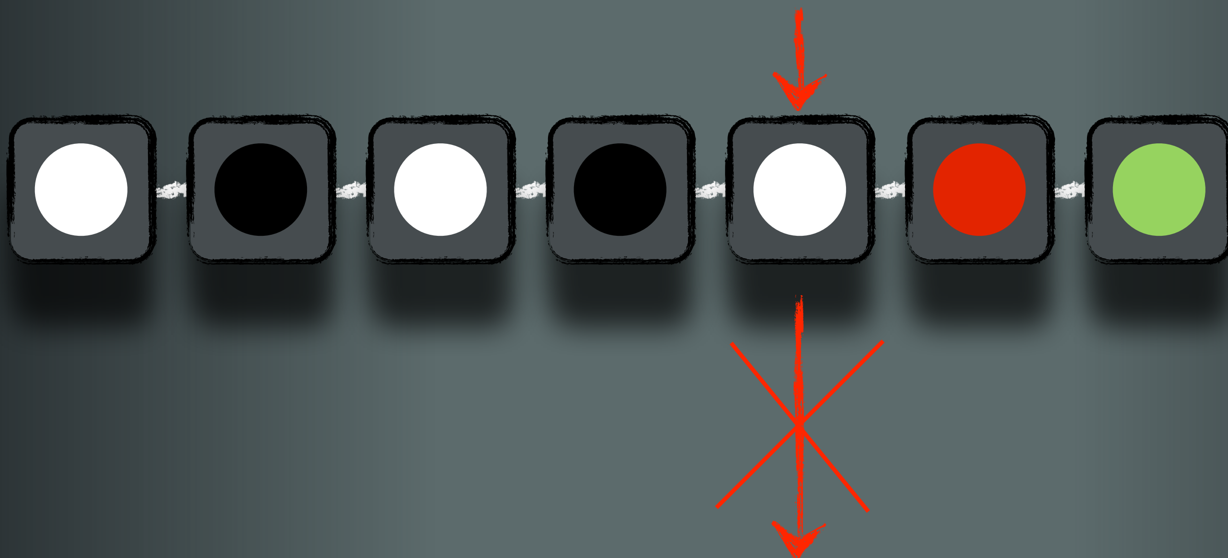
Parameterized System



Existential Global Transition

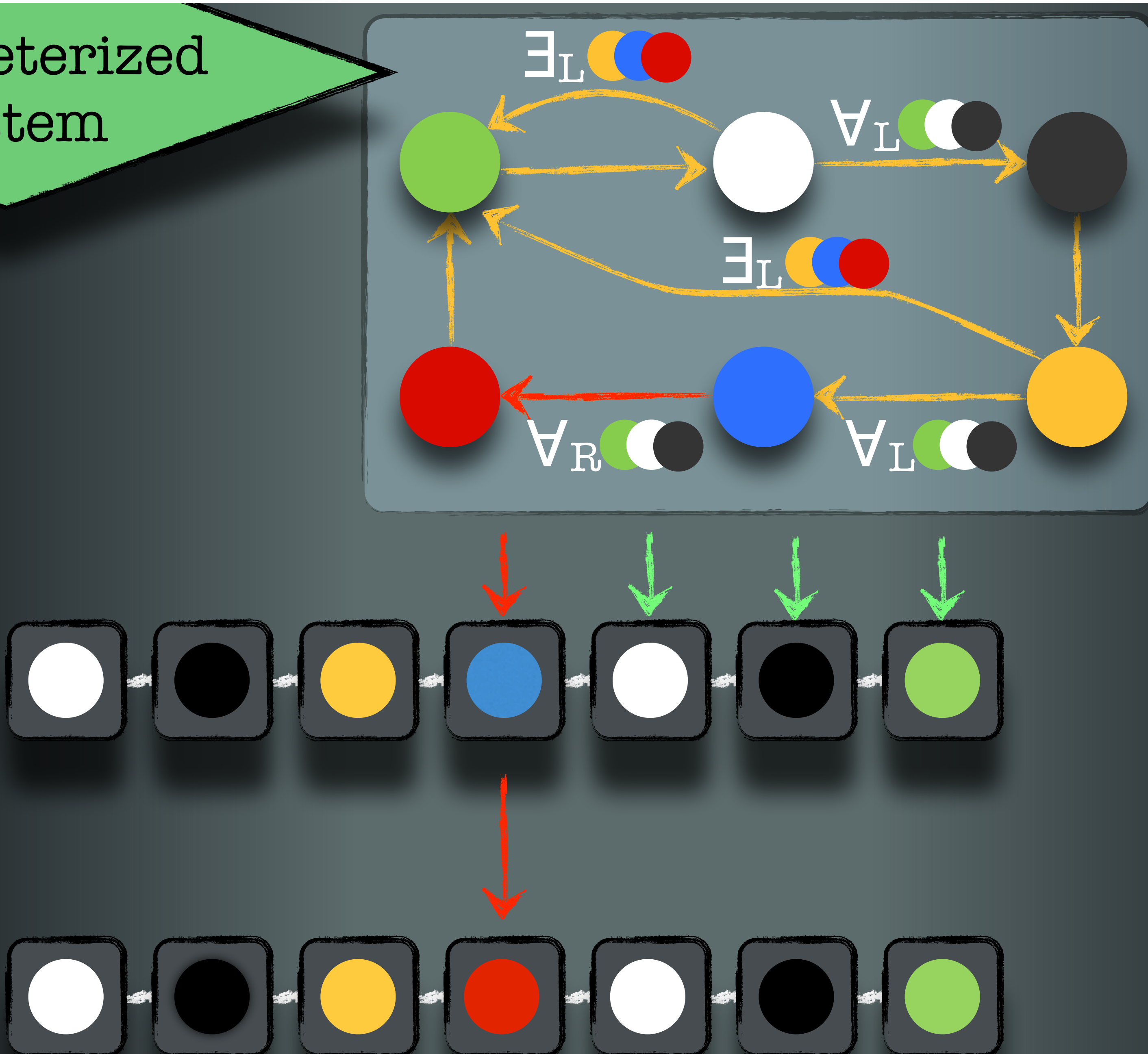
Parameterized System



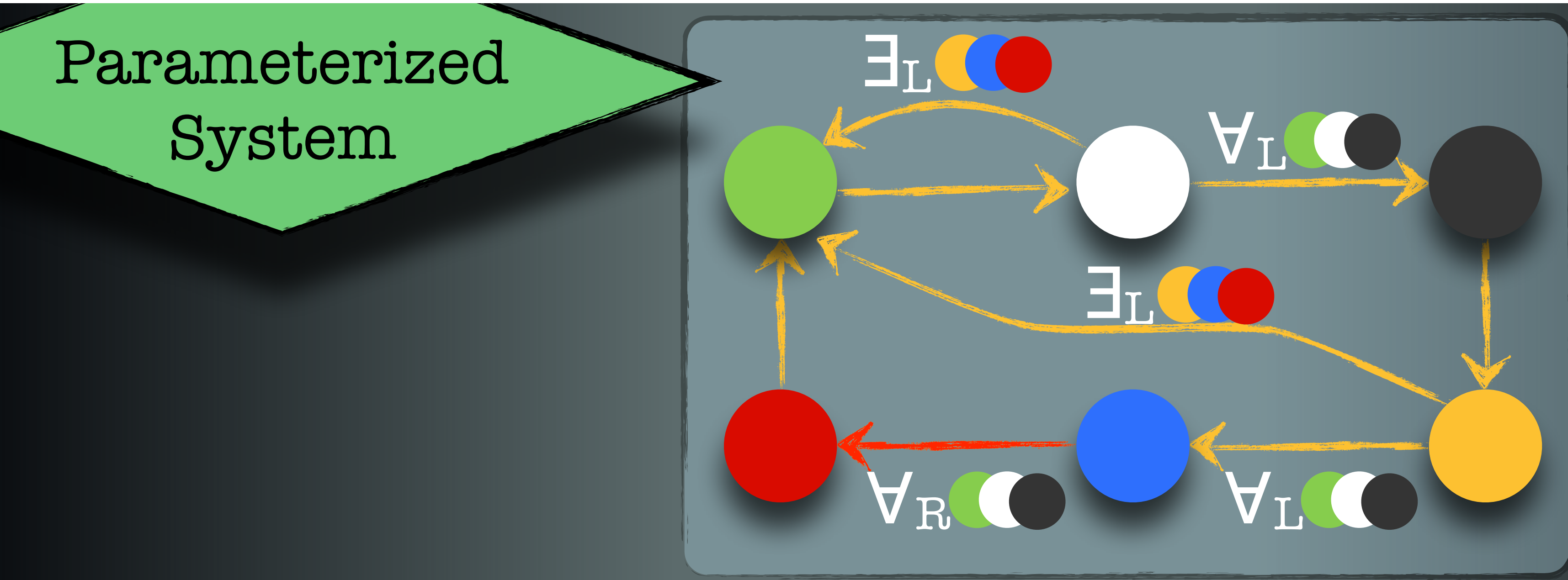


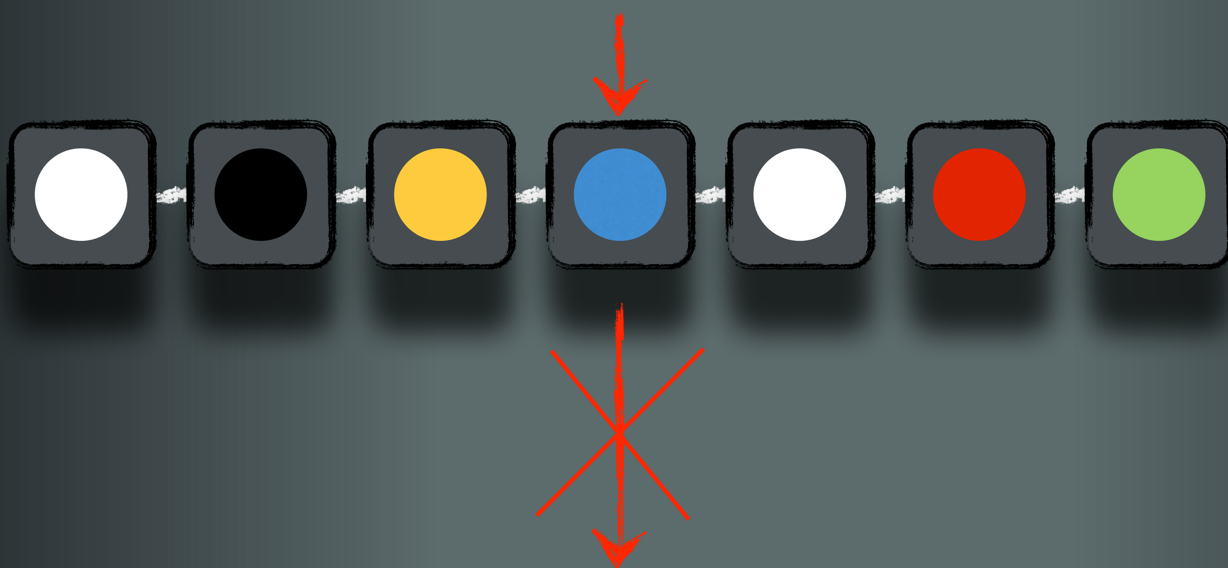
Existential Global Transition

Parameterized System



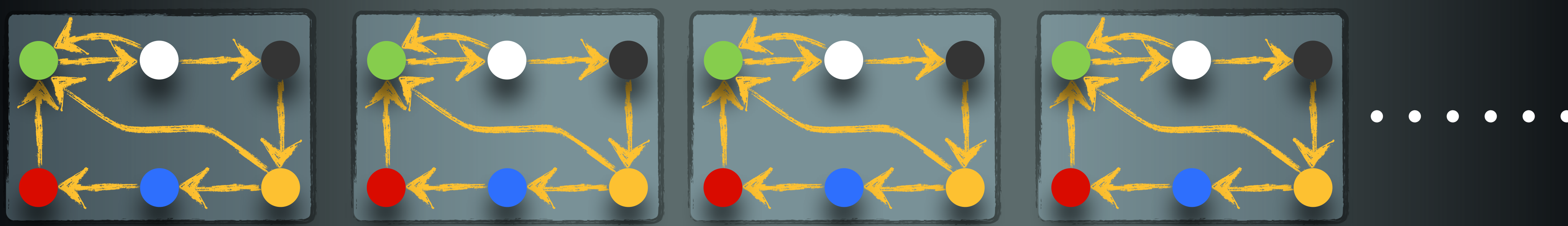
Universal Global Transition





Universal Global Transition

Parameterized System

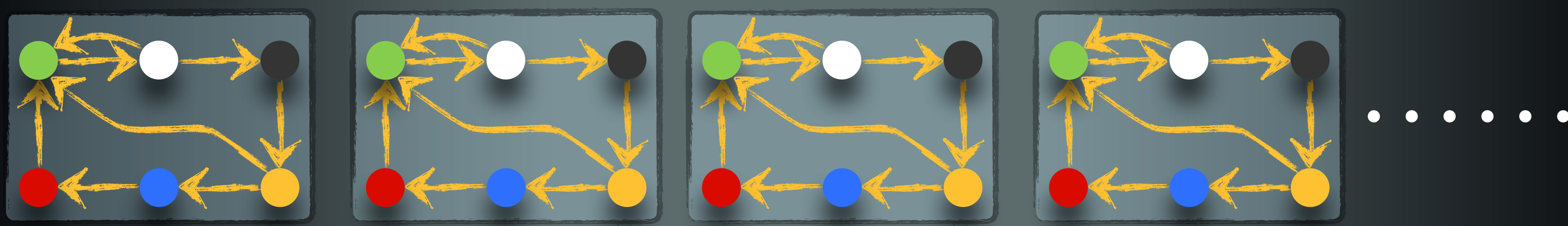


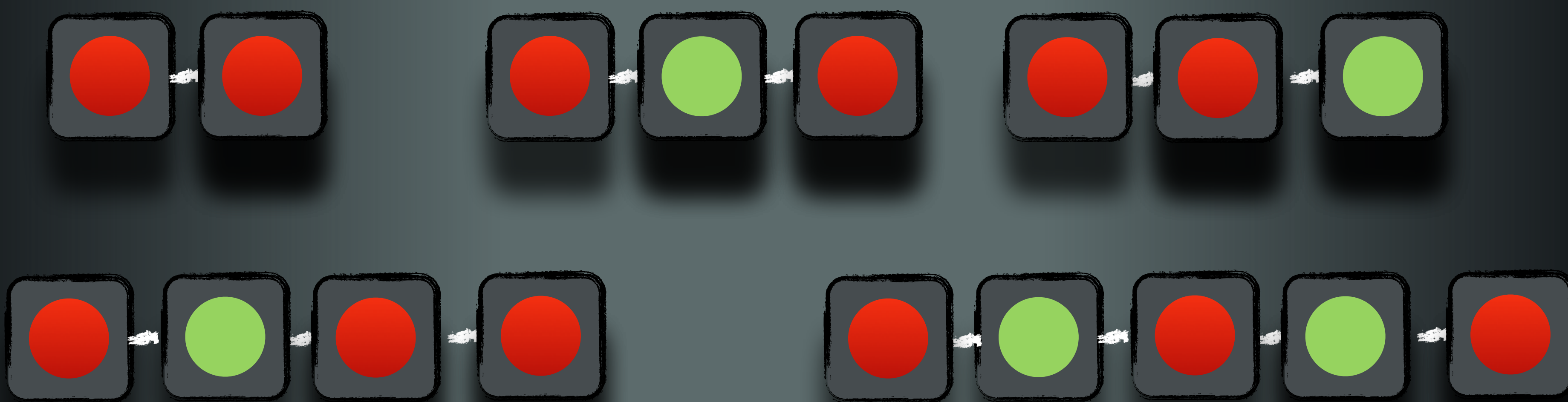
Initial Configurations

Set of initial configurations

- infinite, but
- regular

Parameterized System

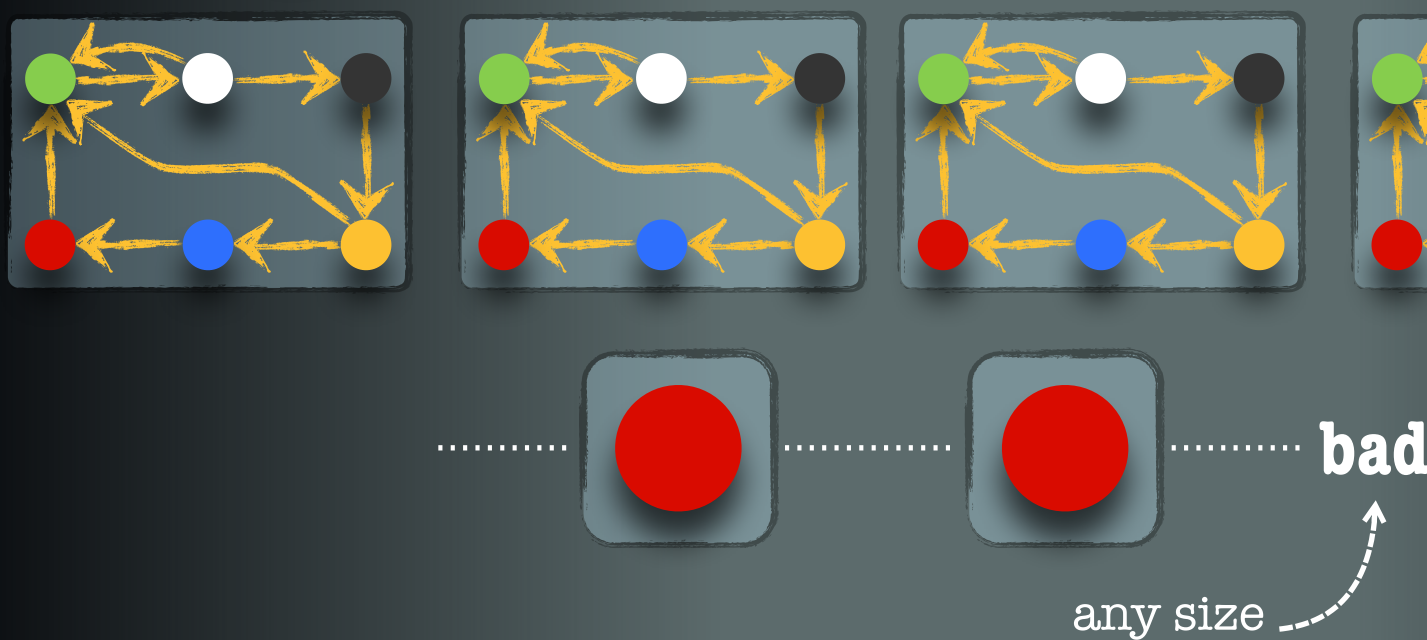




Bad Configurations

two or more processes in critical section

Parameterized System

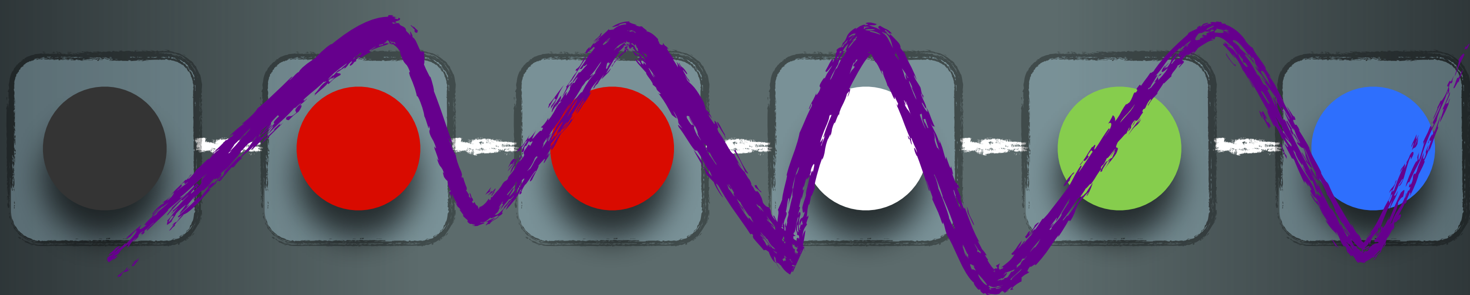


Set of bad configurations
- infinite, but
- upward closed

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

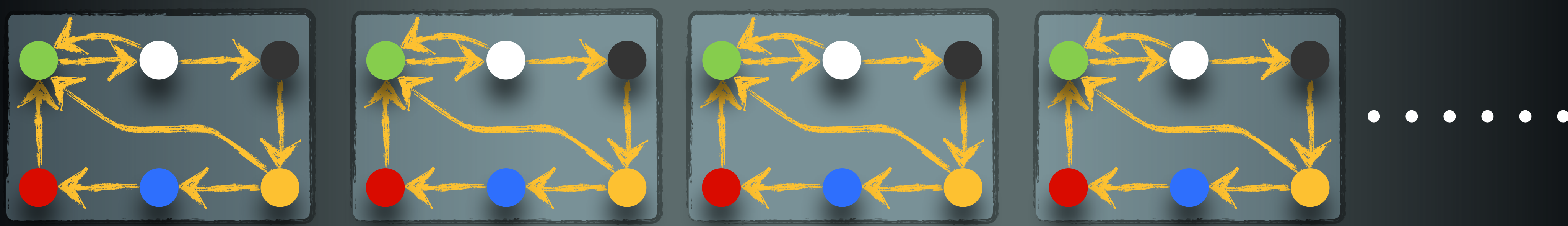
- MOSI
- German

➲ Petri Nets

➲ Trees

➲ Rings

Parameterized System



Remarks

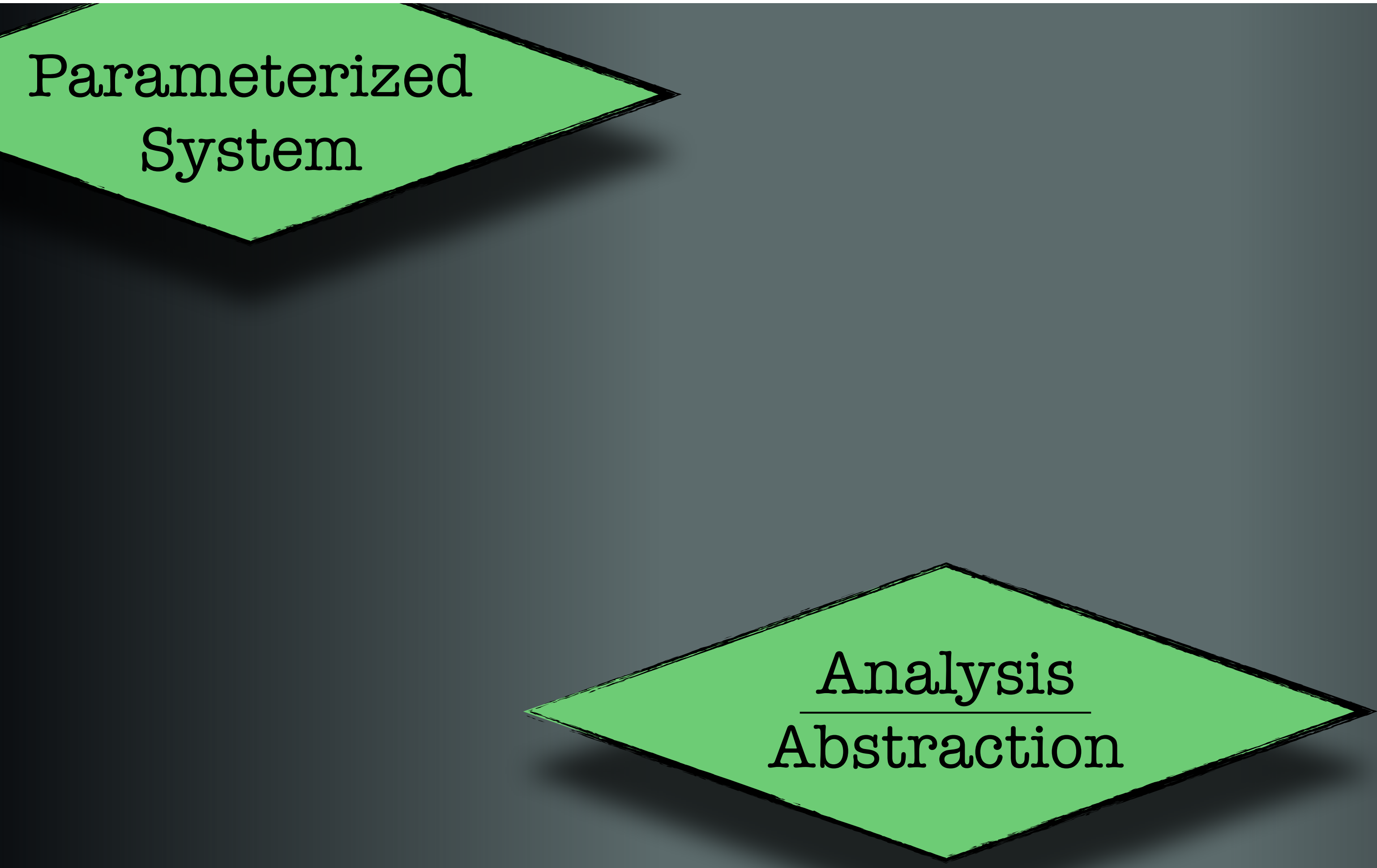
- Infinite-state system
  - unbounded number of processes
  - Parameterized Verification: verify correctness regardless of number of processes
- Problem undecidable in general
  - Challenge: find abstractions that work often

Parameterized System ✓

Analysis
Abstraction

Features

Completeness

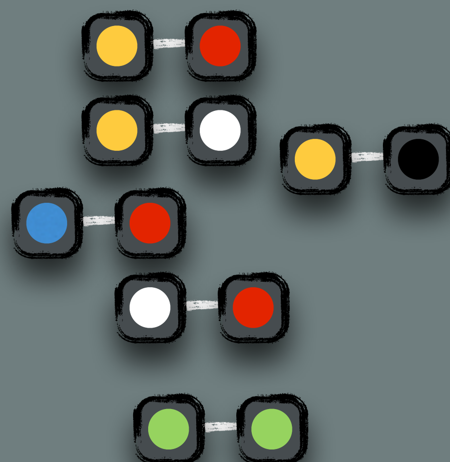

Parameterized
System
Analysis
Abstraction

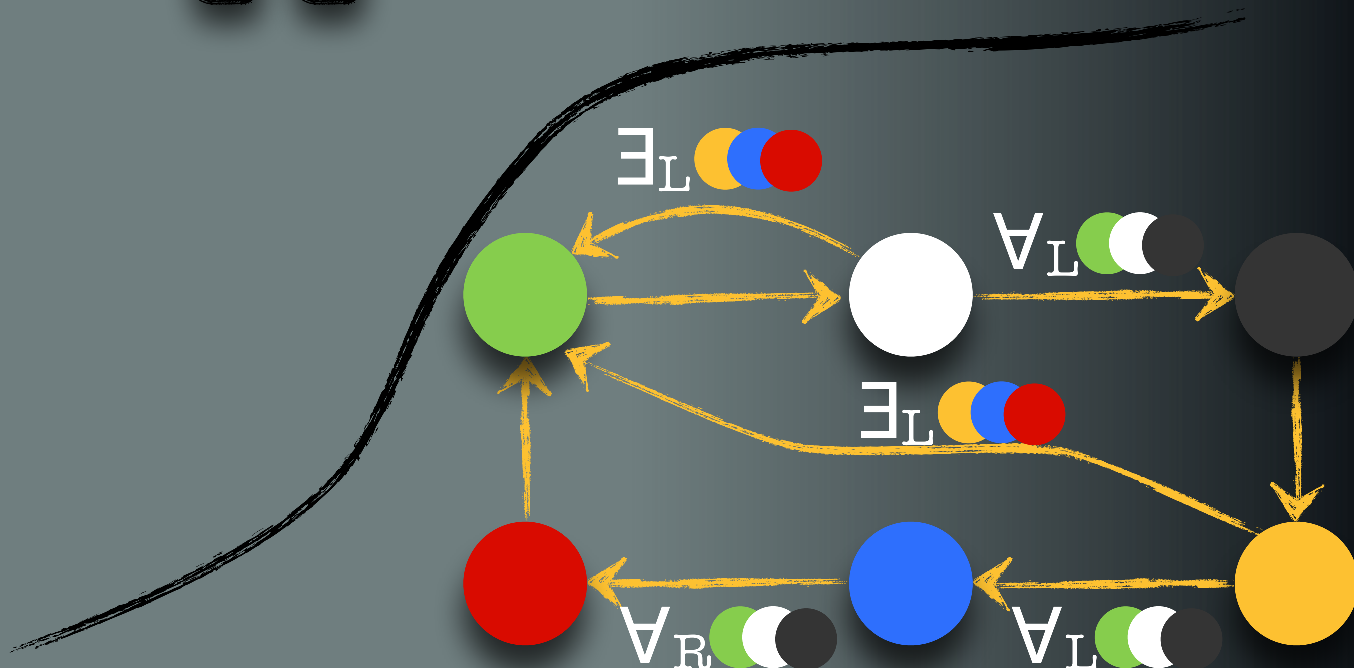
Small model property

⇒ inspect small instances of the system

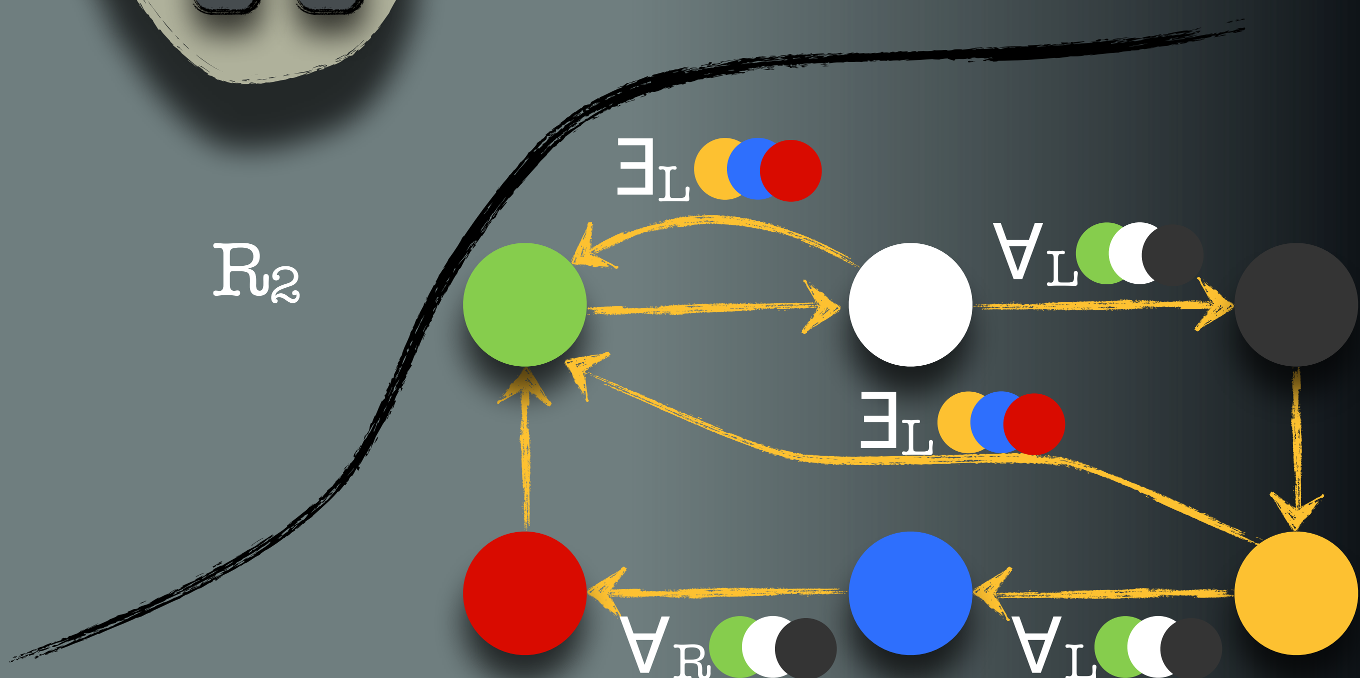
⇒ efficient method

Parameterized Systems





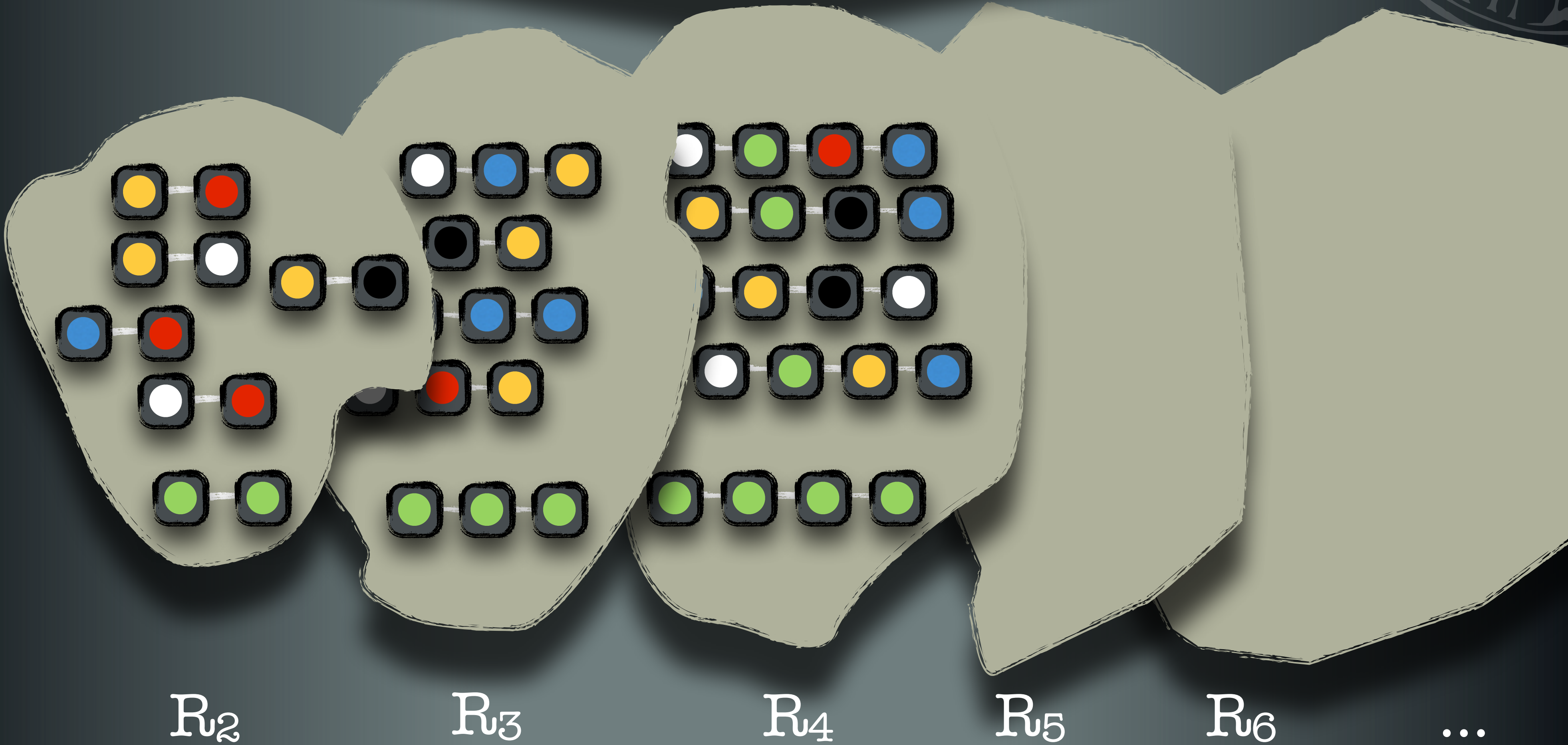
Parameterized Systems



Parameterized Systems

R_2 R_3 R_4

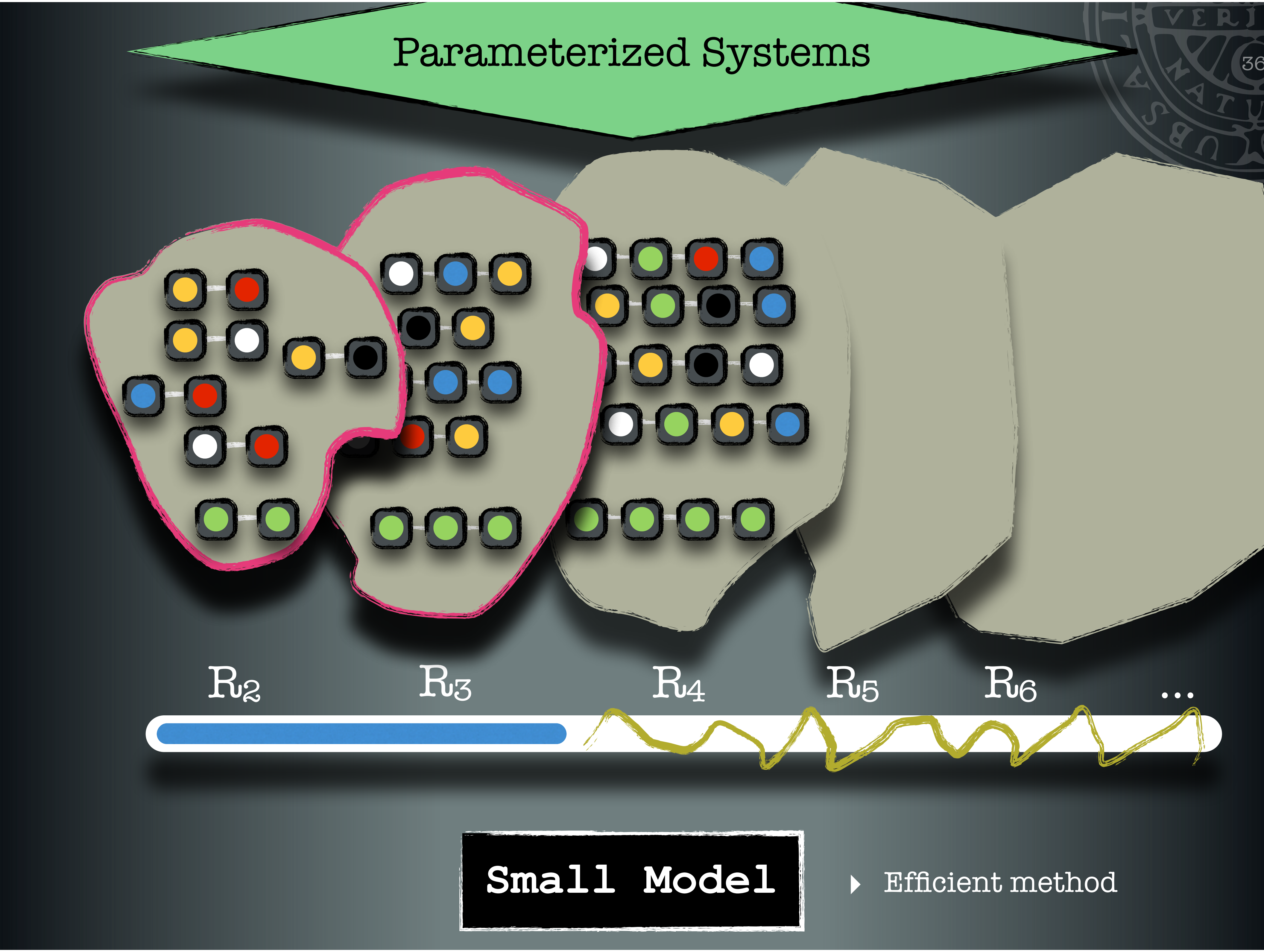
Parameterized Systems



Goal

Safety

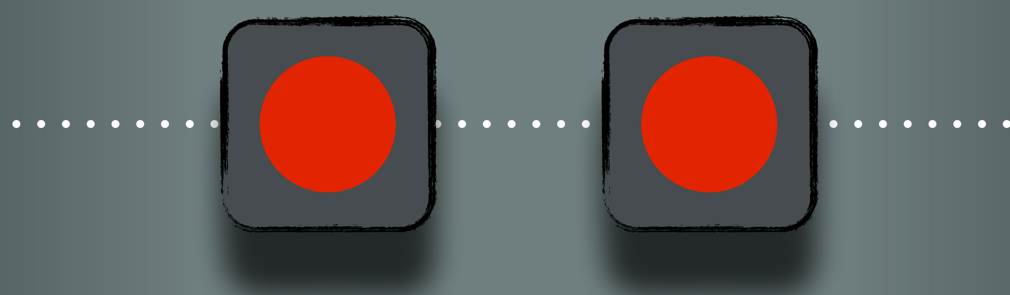
infinite family


Parameterized Systems
R2
R3
R4
R5
R6
...
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



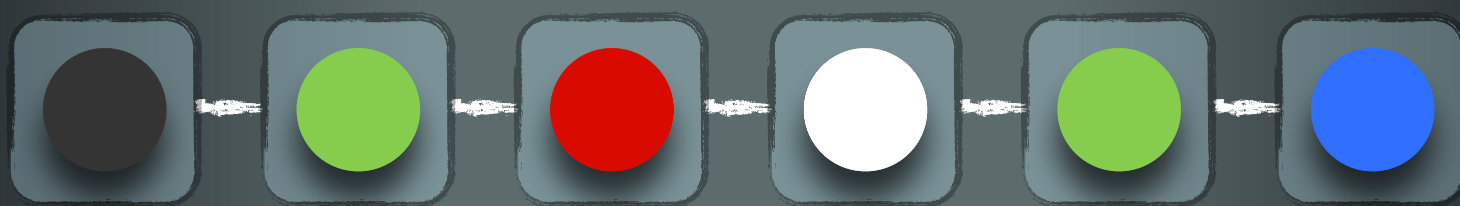
Small Model

Paramete System

Analysis Abstraction

Abstraction modulo k

- k: natural number



$\alpha_k : \mathbb{C} \longrightarrow \mathbb{V}$

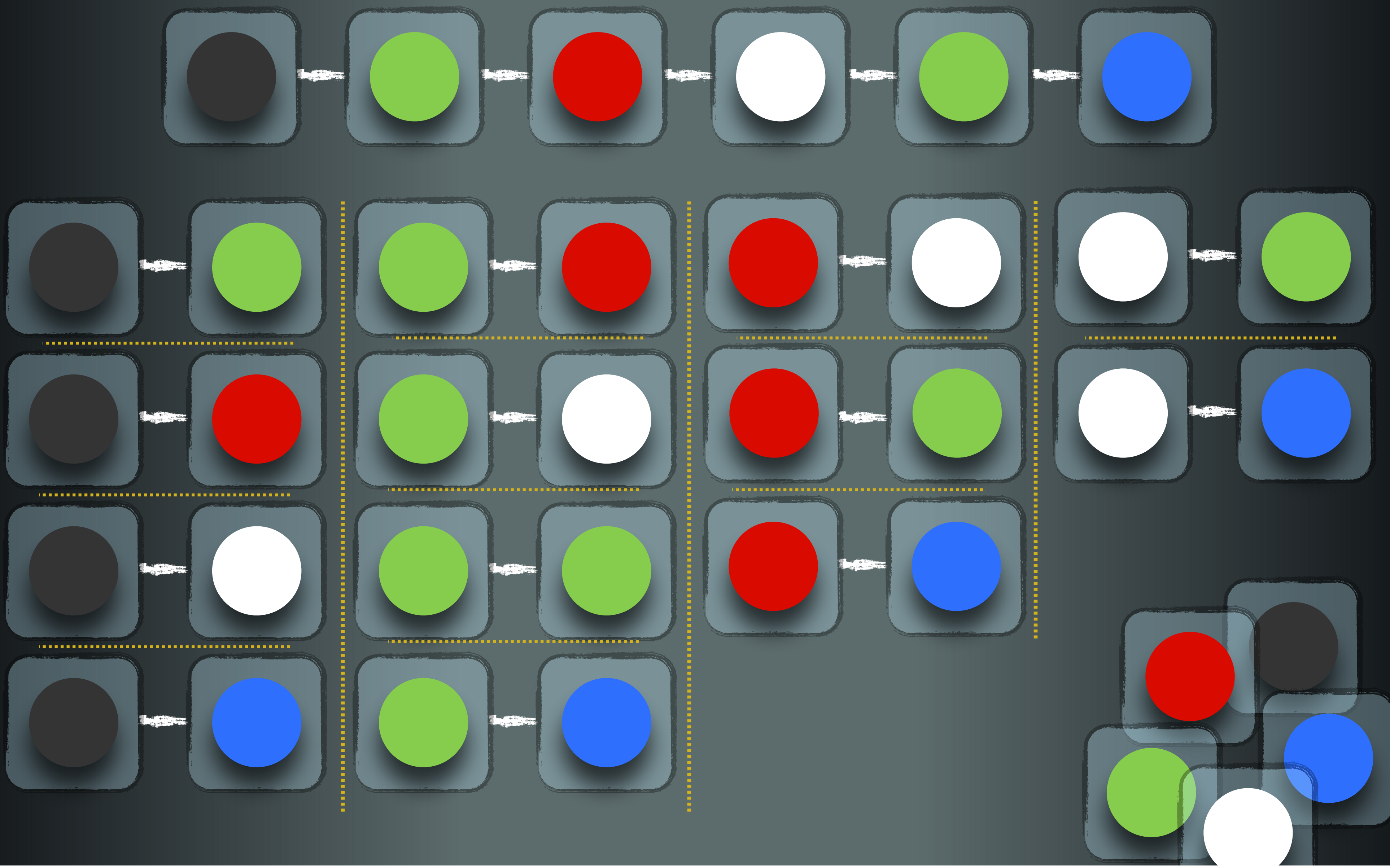
k=2

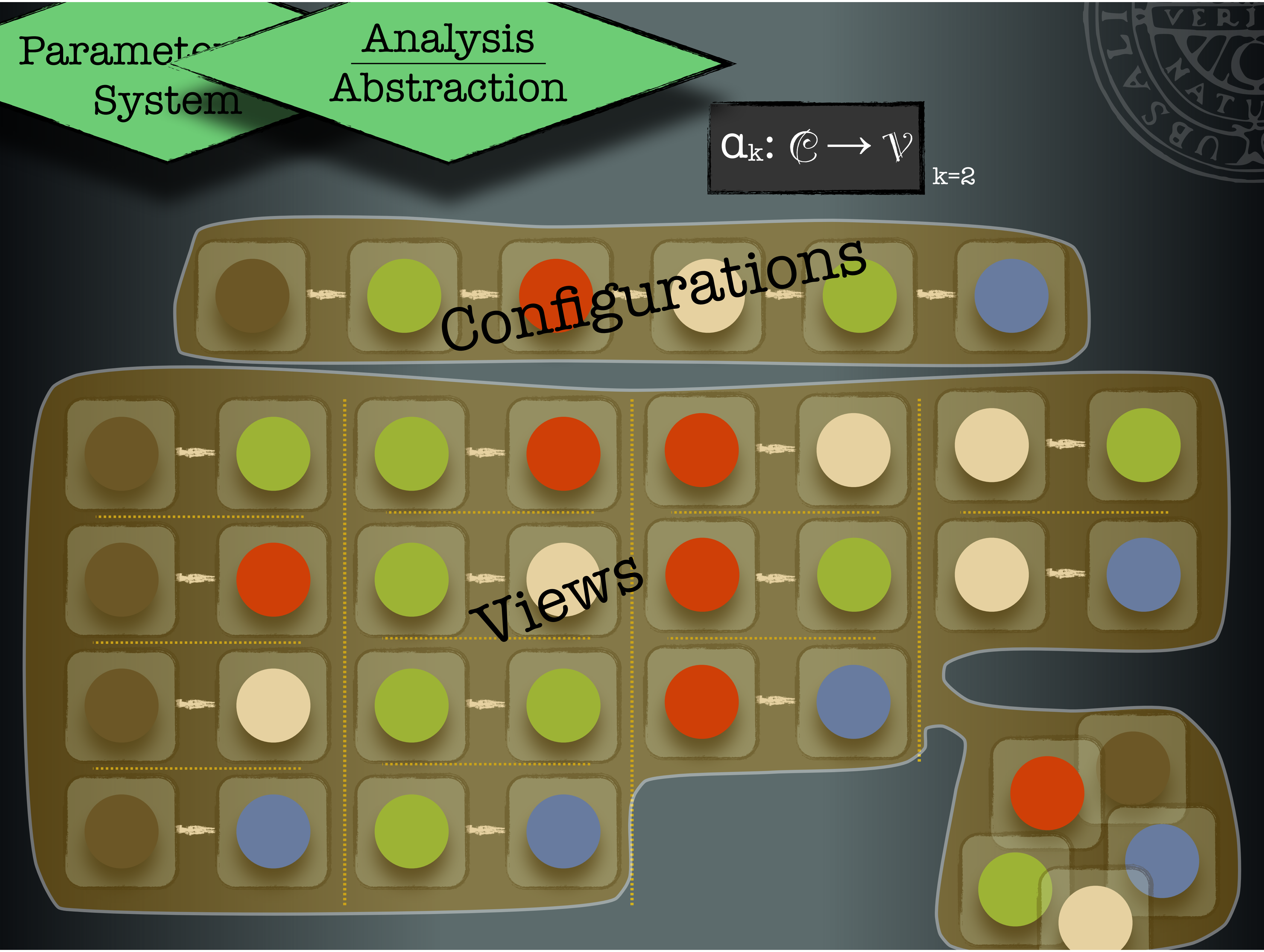
Parameter System

Analysis Abstraction

$\alpha_k : \mathbb{C} \longrightarrow \mathbb{V}$

k=2




Parameter
System
Analysis
Abstraction
$\alpha_k : \mathbb{C} \longrightarrow \mathbb{V}$
k=2
Configurations
Views

For each k : two procedures (in parallel):

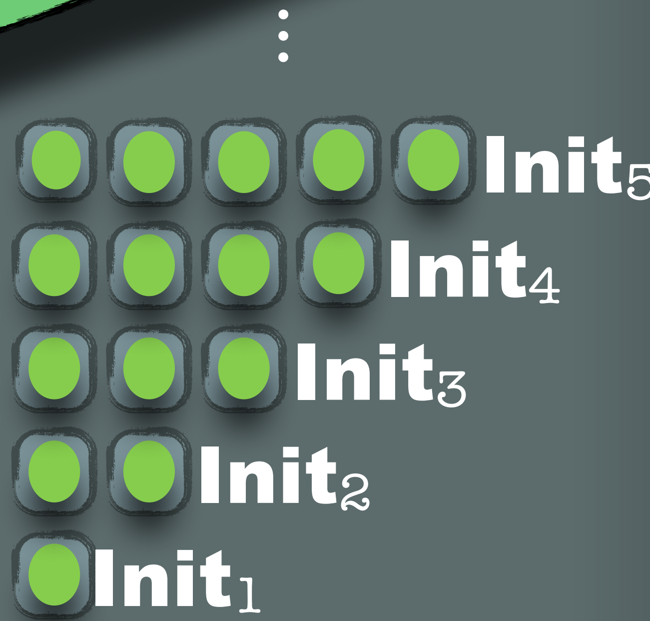
- Bug detection
  - under-approximation
  - concrete domain
- Verification
  - over-approximation
  - abstract domain

- abstraction of initial configurations
- abstract post operator

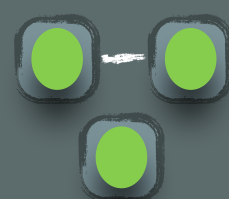
Parameter System

Analysis Abstraction

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



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



Concrete

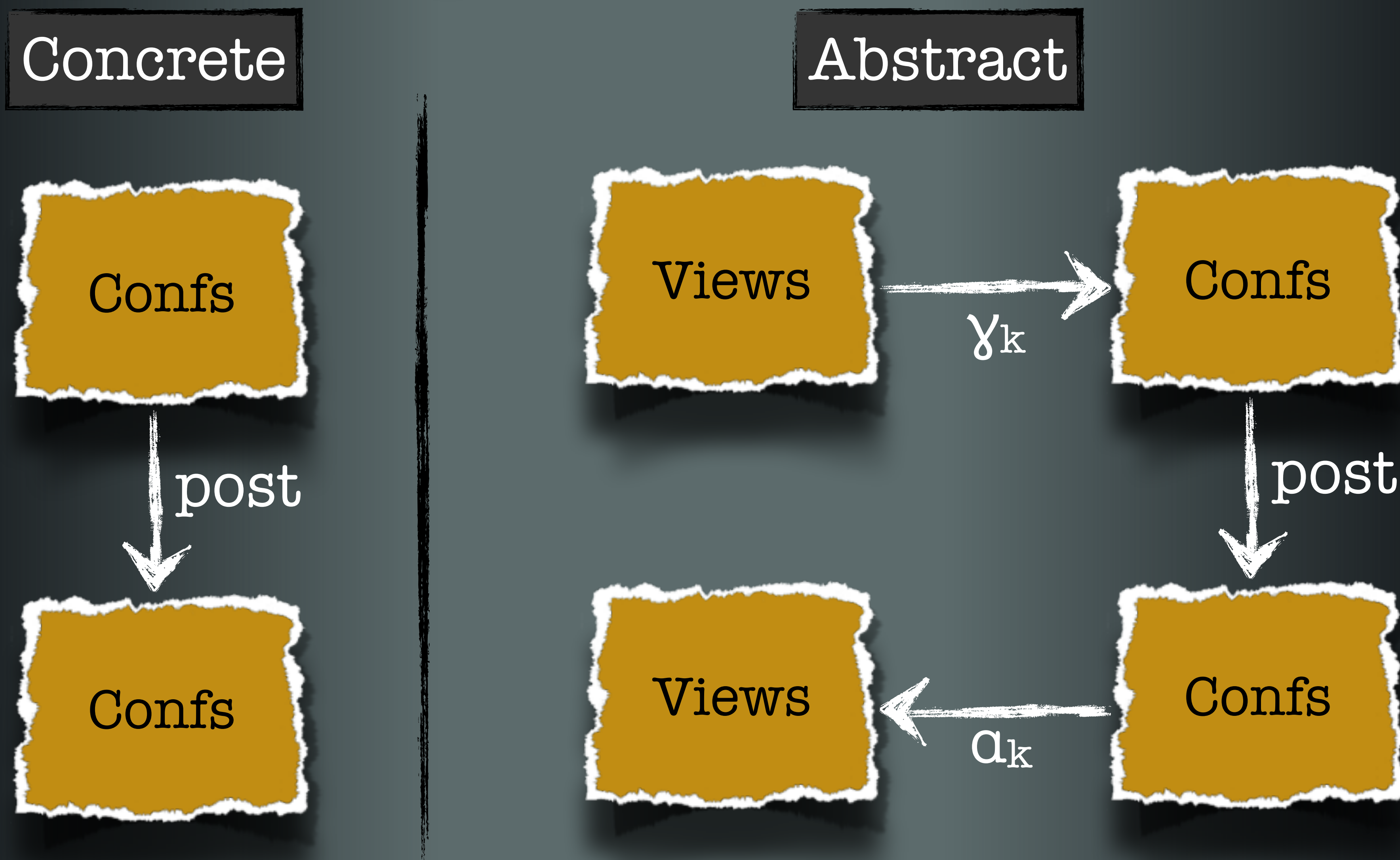
Confs

$\Big\downarrow$ post

Confs

Parameterized System

Analysis Abstraction

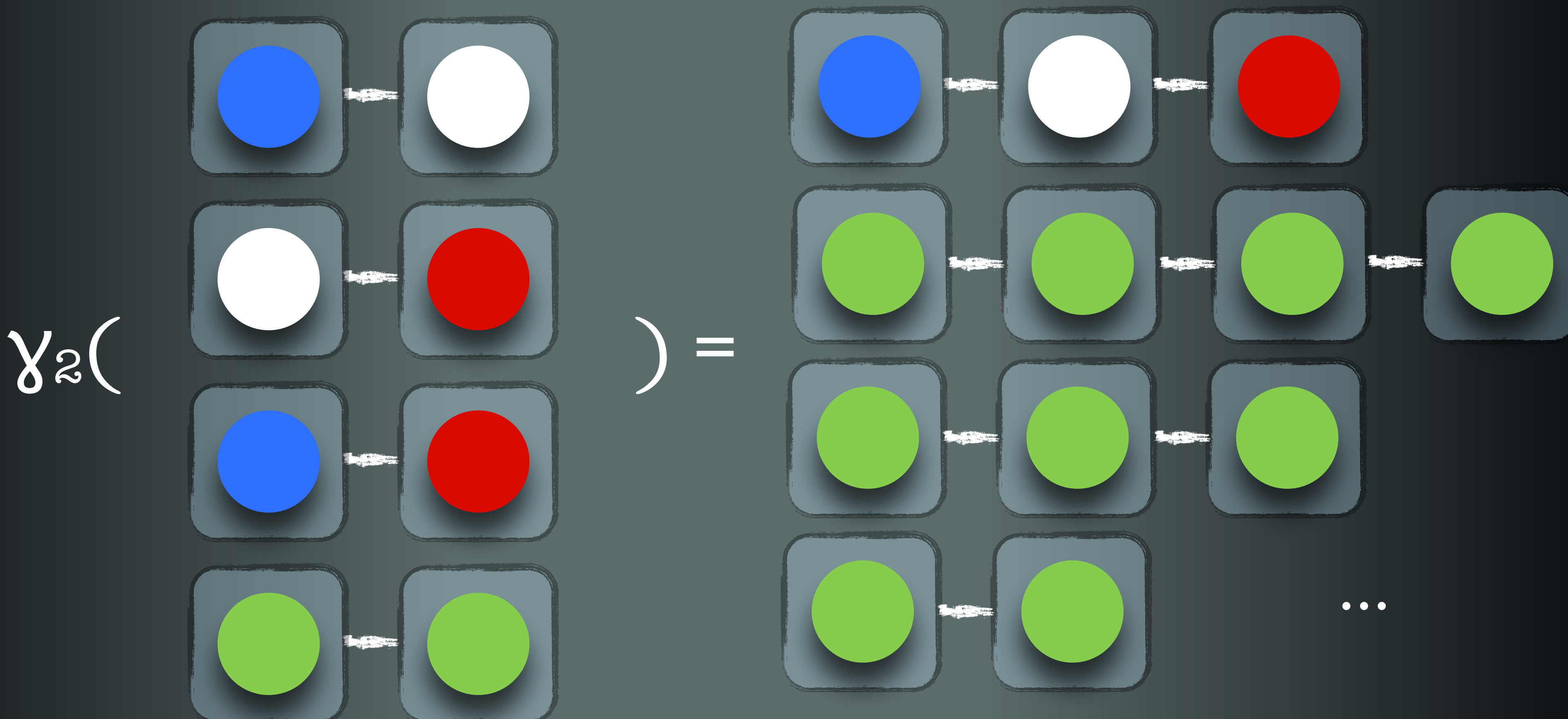


Parameter System

Analysis Abstraction

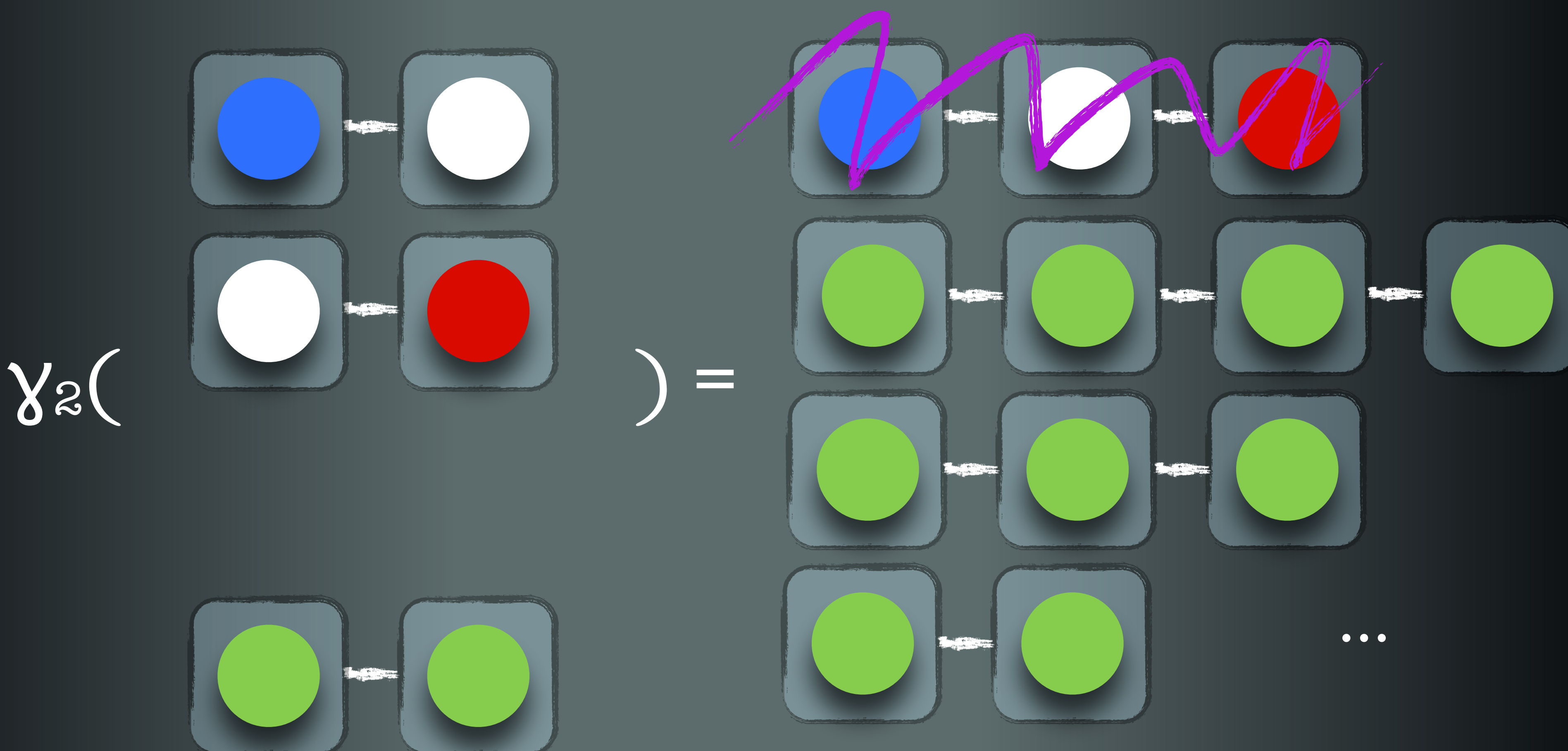
γ_k: set of $\mathcal{V} \longrightarrow$ set of $\mathbb{C}$

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


$\gamma_2($
$) =$
...

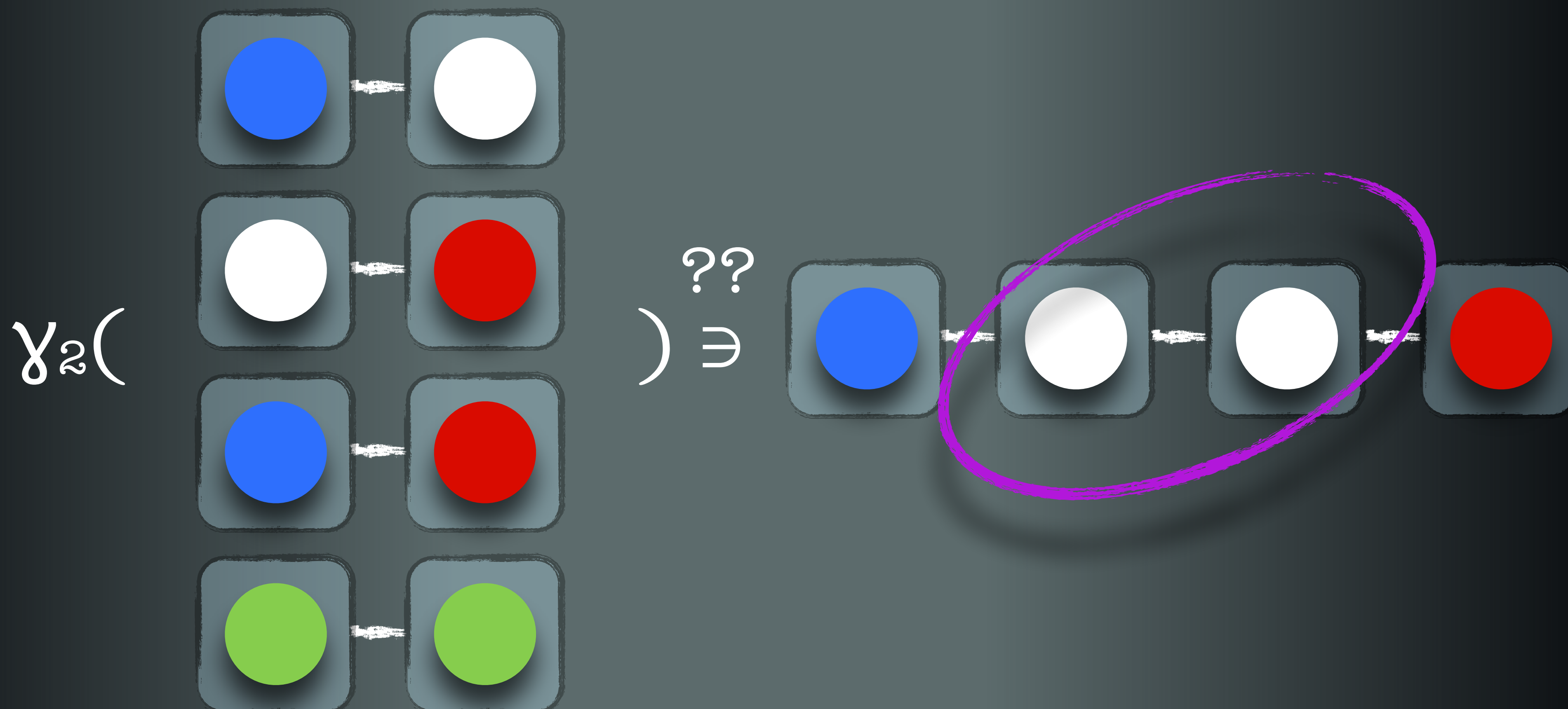
Parameter System

Analysis Abstraction



Parameter System

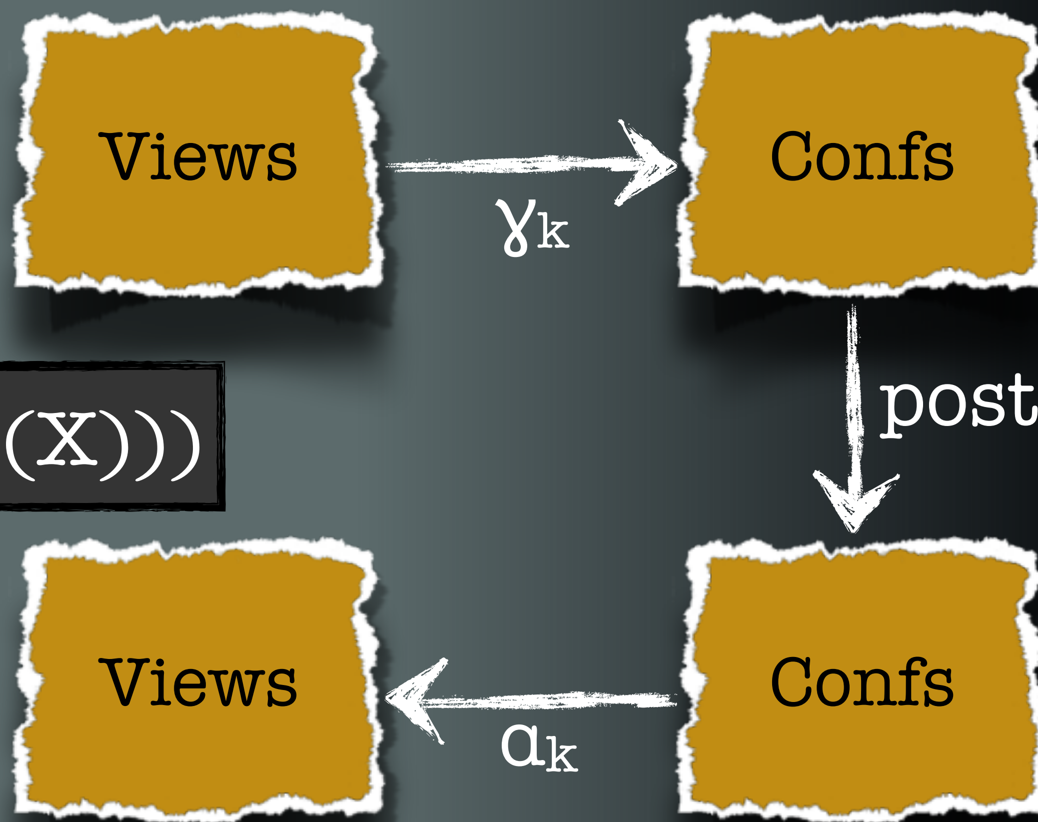
Analysis Abstraction



Parameter System

Analysis Abstraction

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



Parameter System

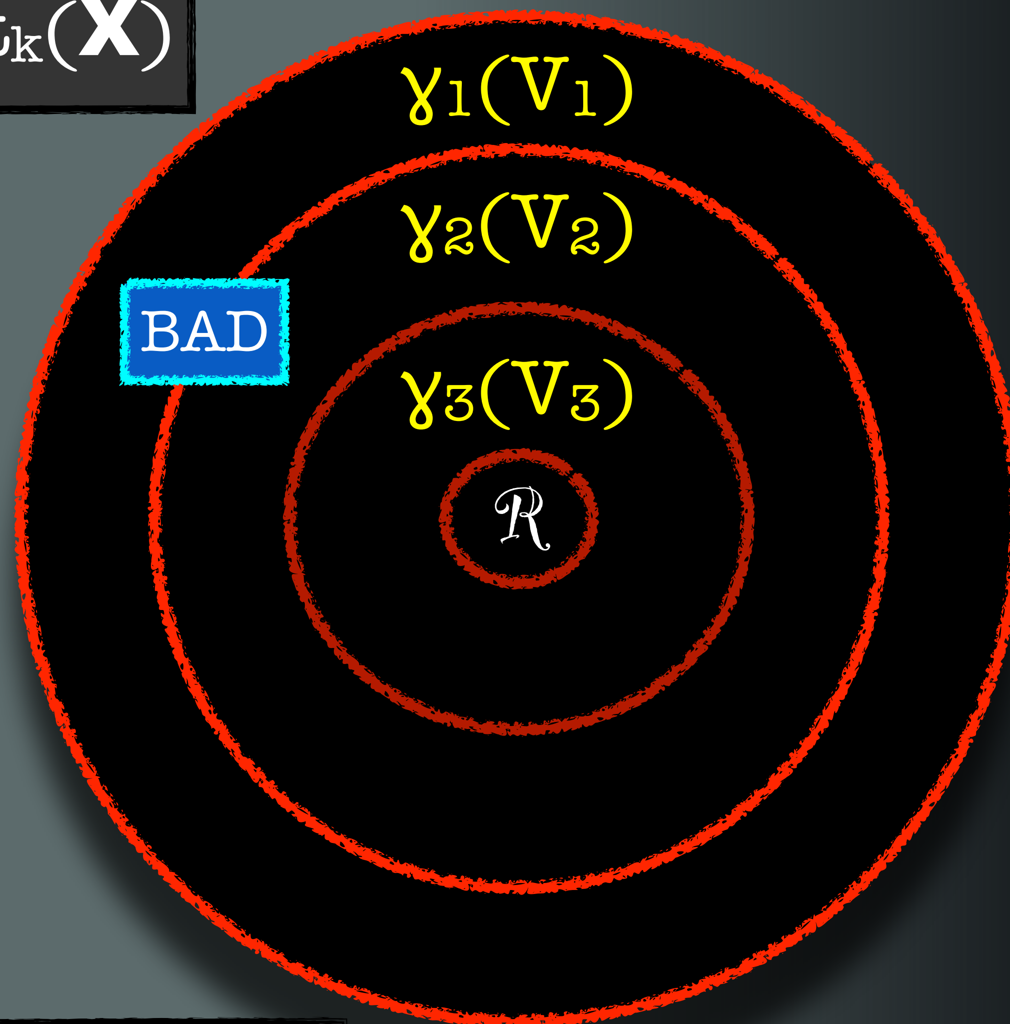
Analysis Abstraction

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

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

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

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



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

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

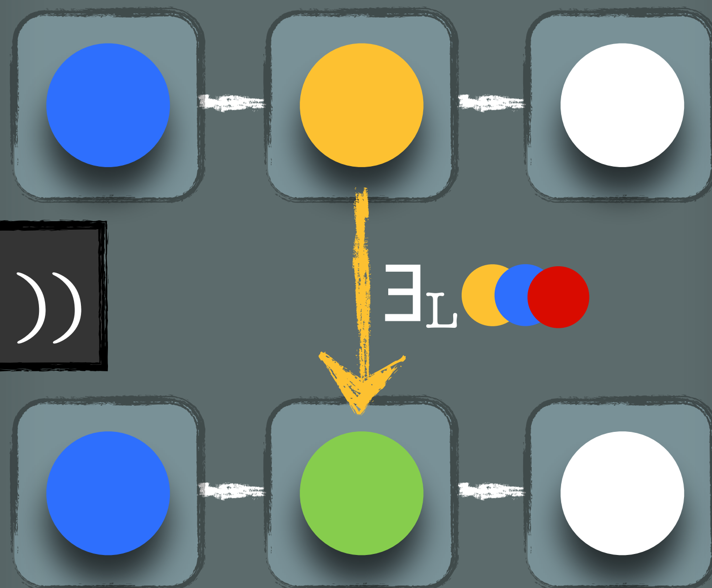
only of size k+1

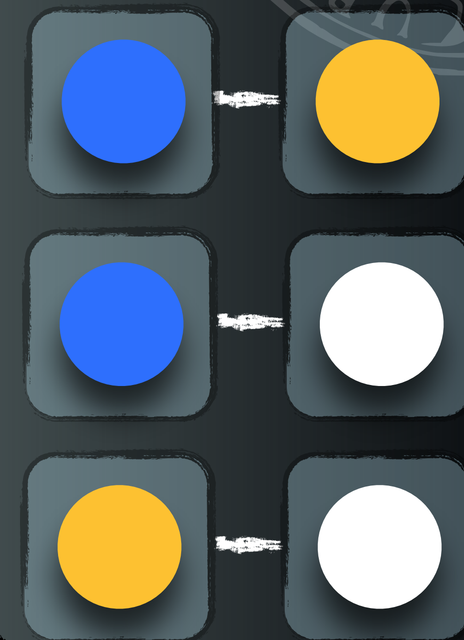
$$\gamma_{k|m}(\mathbf{X}) = \{\, c \in \mathbb{C} \mid \alpha_k(c) \subseteq \mathbf{X},\ |c| \leq m \,\}$$

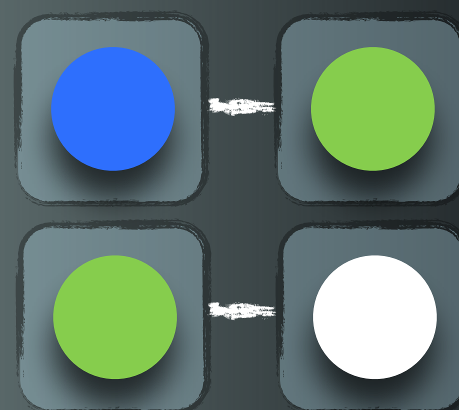
Paramete[...] System

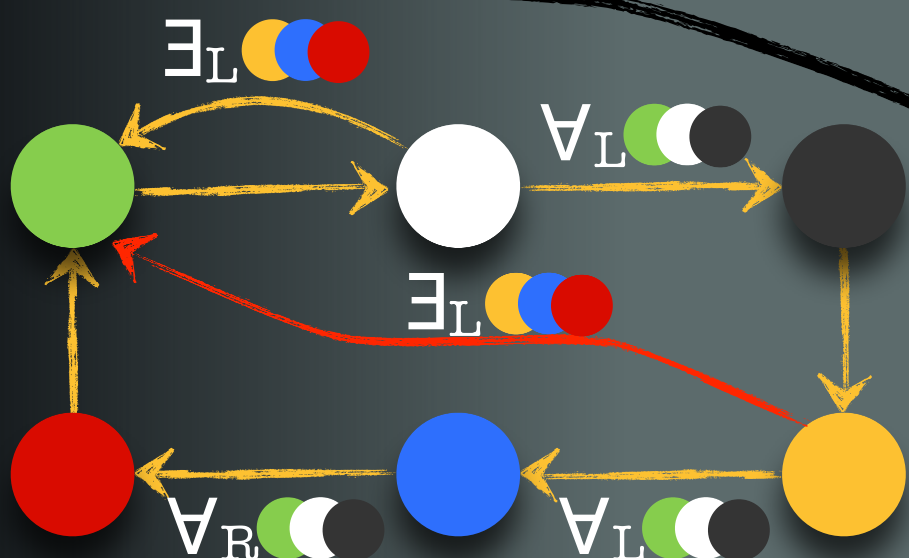
Analysis
Abstraction

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









Parameter System

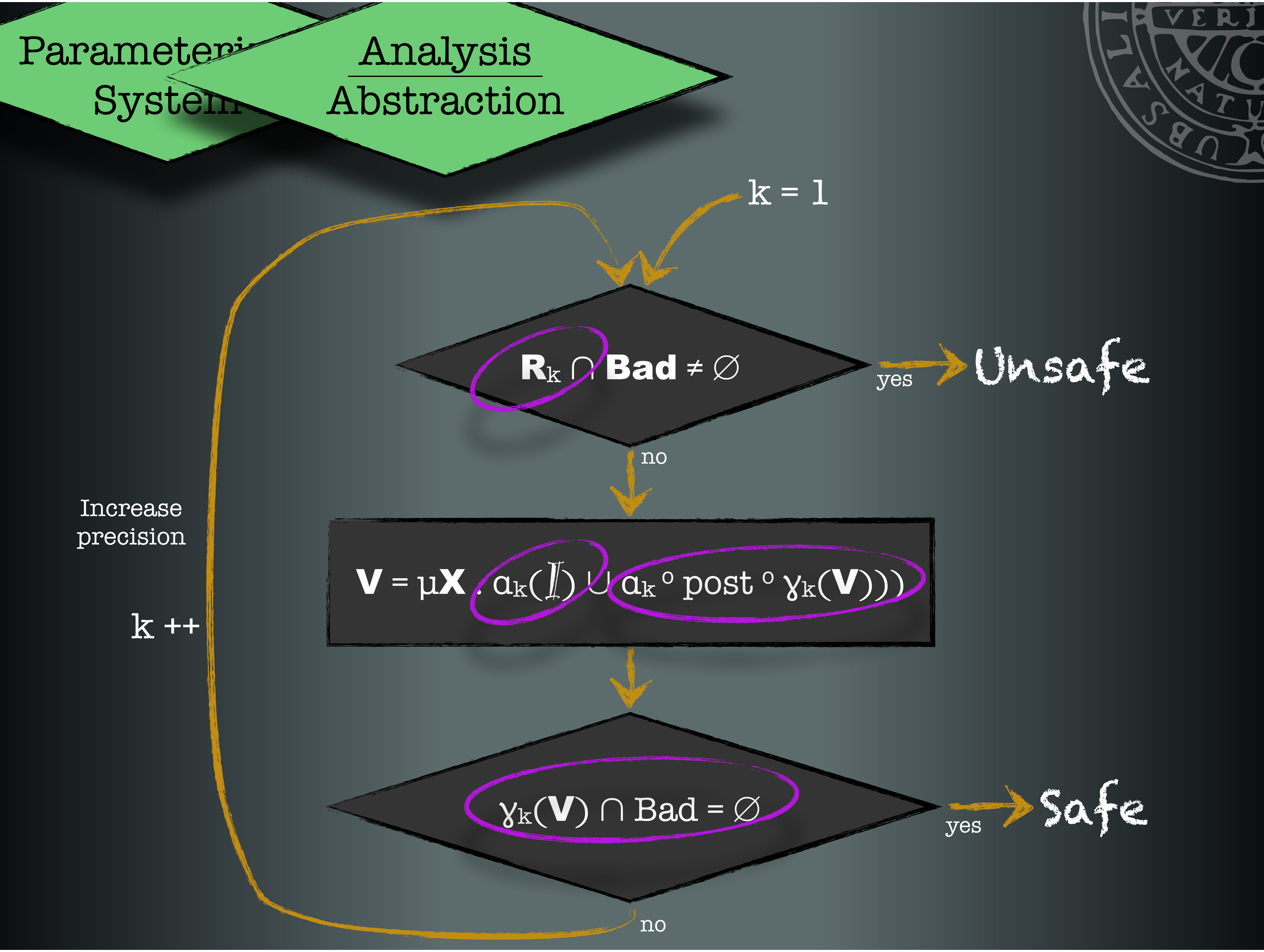
Analysis Abstraction

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

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

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

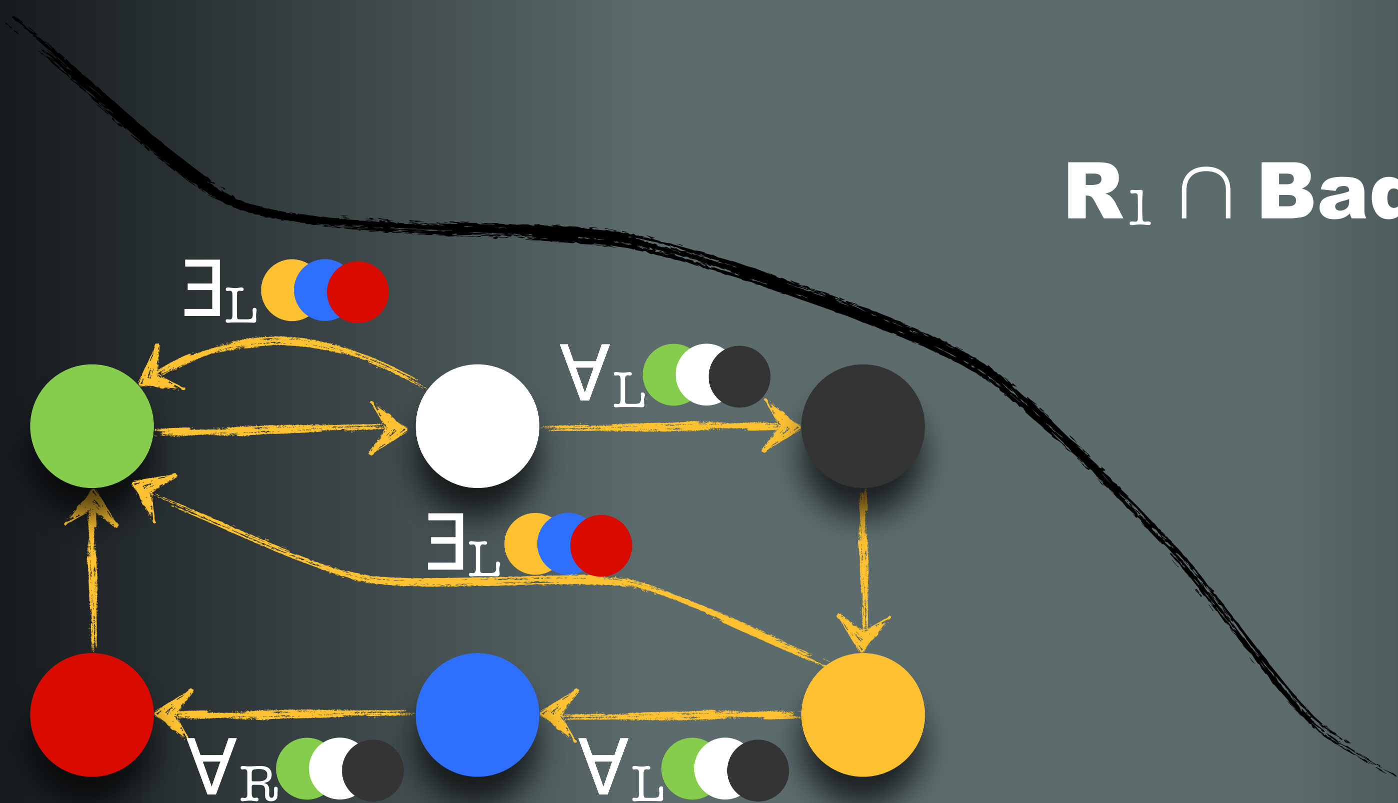
$$
\begin{array}{l}
\text{for } k = 1 \text{ to } \infty \text{ do} \\
\quad \text{if } \mathbf{R}_k \cap \mathbf{Bad} \neq \varnothing \text{ then return Unsafe} \\
\quad \mathbf{V} = \mu \mathbf{X} \,.\, \alpha_k(\mathcal{I}) \cup \alpha_k \circ \mathit{post} \circ \gamma_k(\mathbf{X}) \\
\quad \text{if } \gamma_k(\mathbf{V}) \cap \mathbf{Bad} = \varnothing \text{ then return Safe}
\end{array}
$$



Parameteri
Syste
Analysis
Abstraction
k = 1
$\mathbf{R}_k \cap \mathbf{Bad} \neq \varnothing$
yes
Unsafe
no
$\mathbf{V} = \mu\mathbf{X} . \alpha_k(\mathbb{I}) \cup \alpha_k \circ \mathrm{post} \circ \gamma_k(\mathbf{V})))$
$\gamma_k(\mathbf{V}) \cap \mathrm{Bad} = \varnothing$
yes
Safe
no
Increase precision
k ++

k=1 **R**$_1$:

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

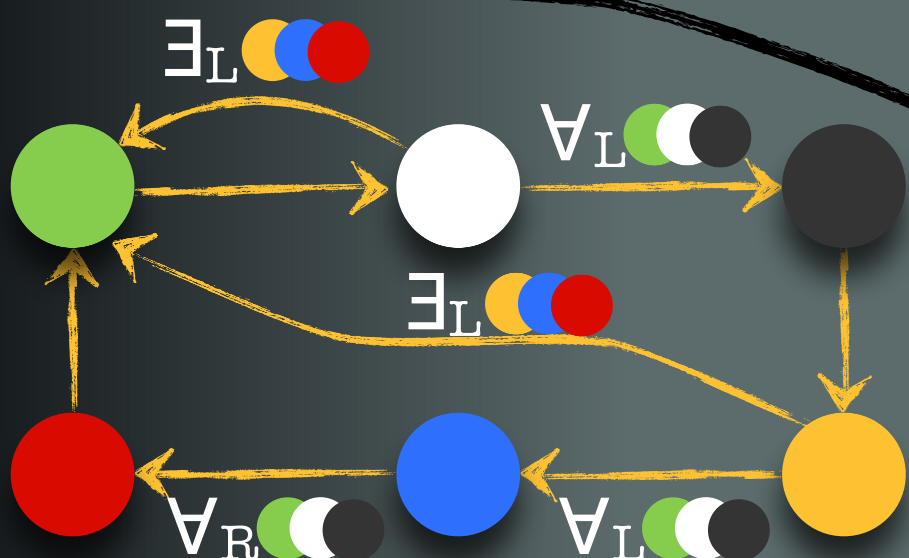


Parameter System

Analysis Abstraction

k=1

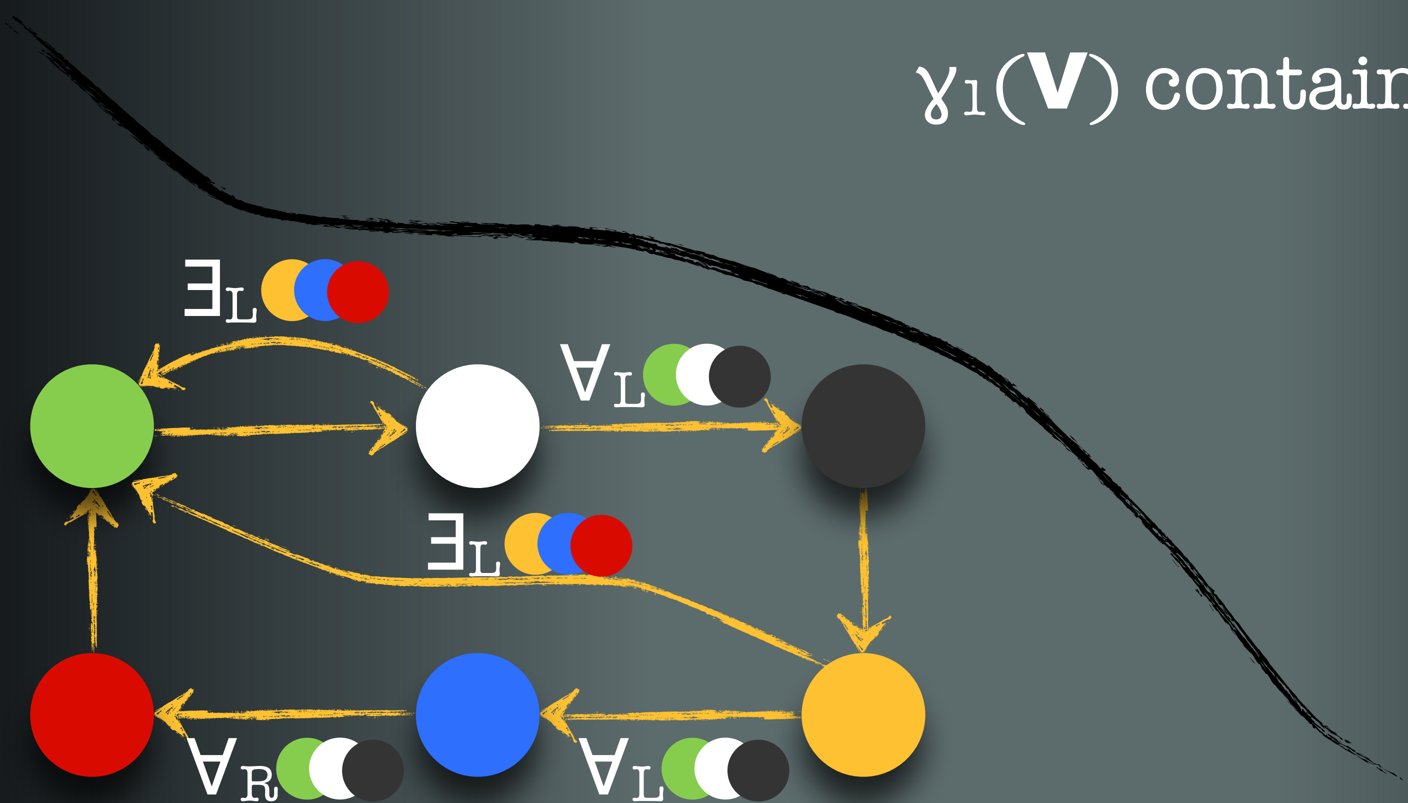
$$\mathbf{V} = \mu\mathbf{X} \,.\, \alpha_1(I) \cup \alpha_1 \circ \mathrm{post} \circ \gamma_1(\mathbf{X})$$



k = 1

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

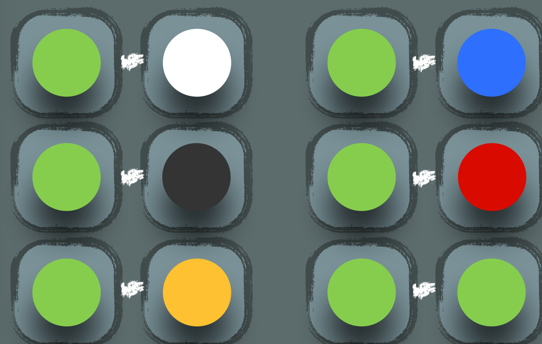
$\gamma_1(\mathbf{V})$ contains everything!



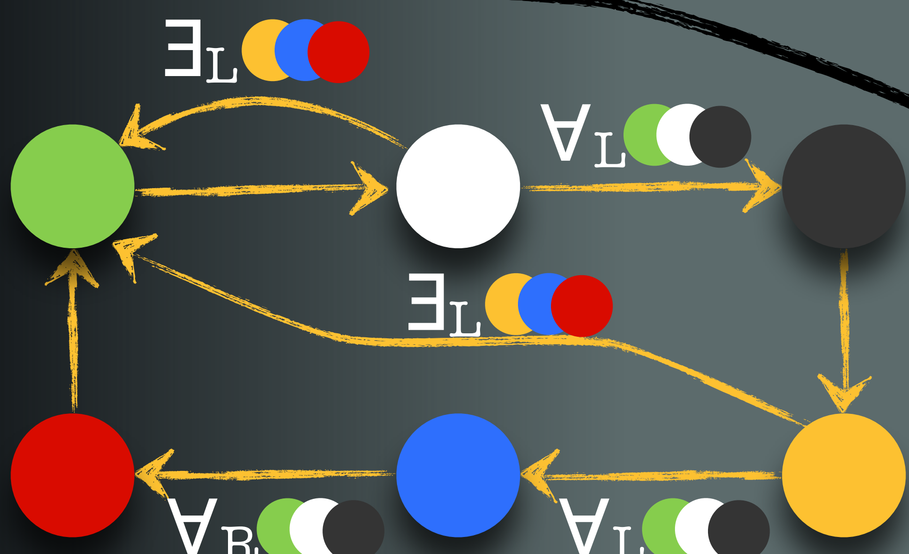
Parameterised System

Analysis
Abstraction

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



...



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

Paramete[r]
System

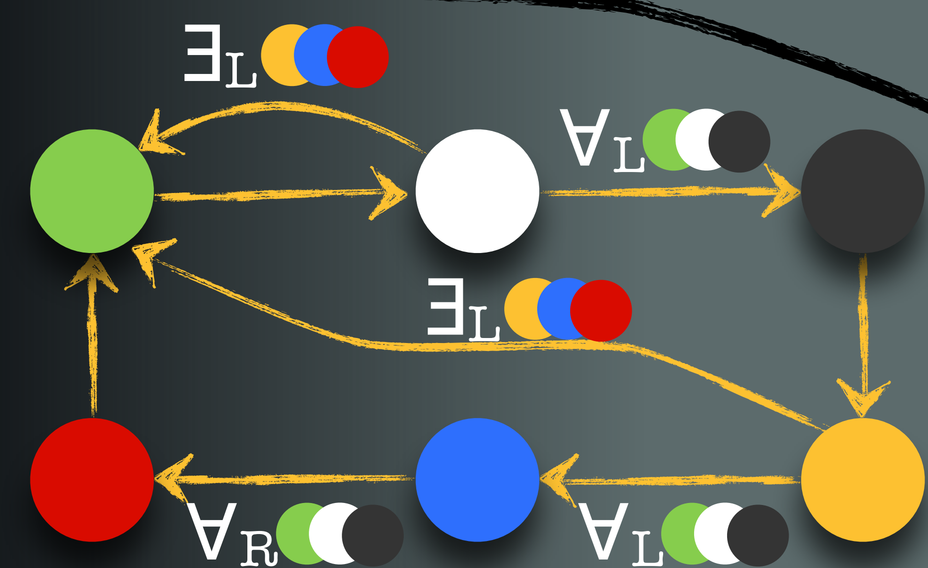
Analysis
Abstraction

k=2

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

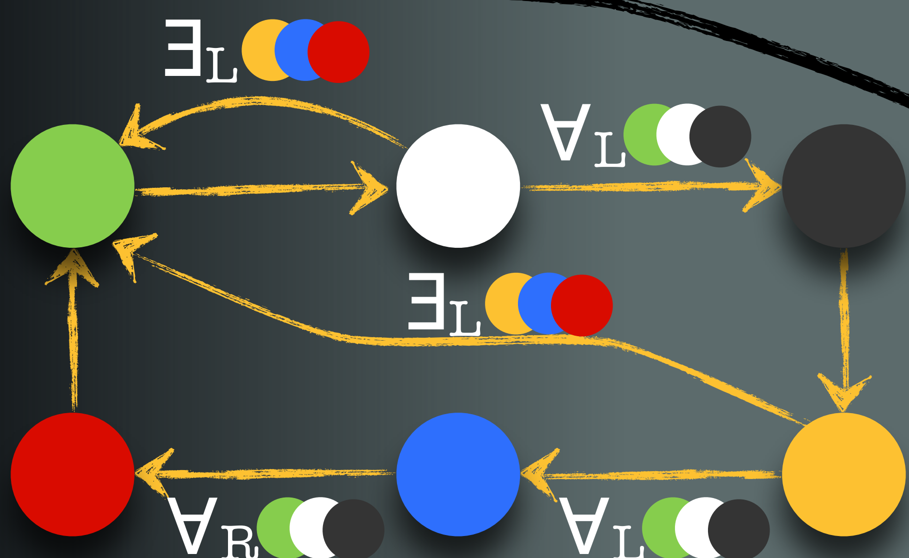
extensions of size 3 only

All pairs but



k = 2

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



Collected all views

k=2 is cutoff point

Parameter
System

Analysis
Abstraction

Features

Universal condition

Existential condition

Universal condition

Existential condition

Global variables

Broadcast

Creation/Deletion

Rendez-Vous

Paramete
System

Analysis
Abstrac

Features

Universal condition

Existential condition

Global variables

Broadcast

Creation/Deletion

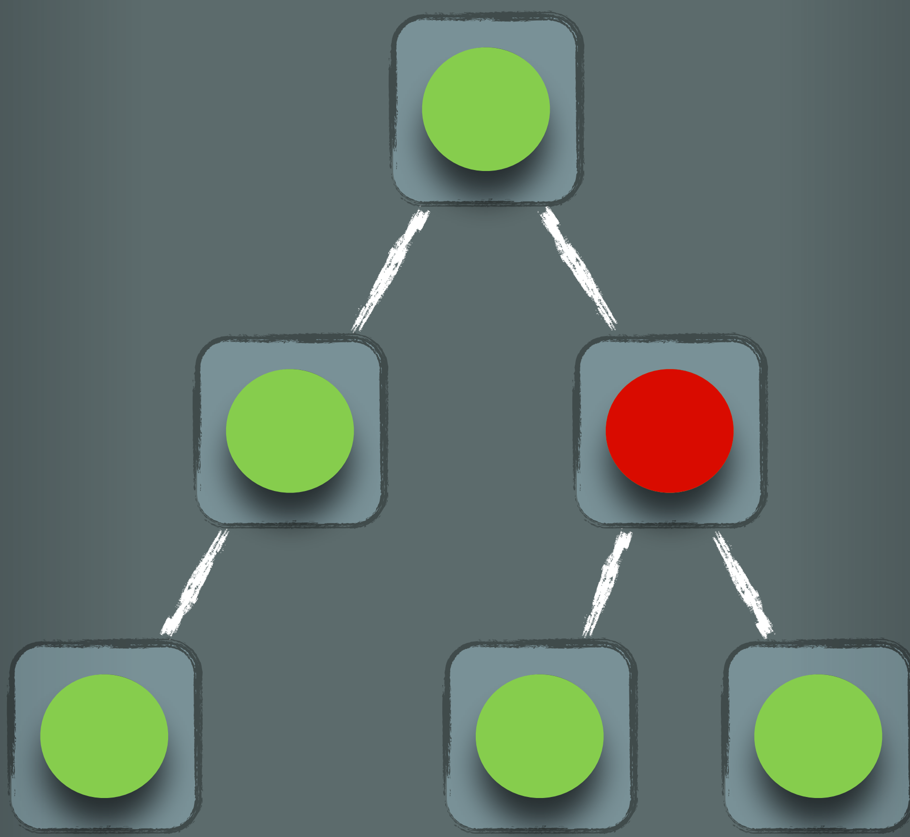
Rendez-Vous

Topologies

- Linear
- Ring
- Tree
- Multiset

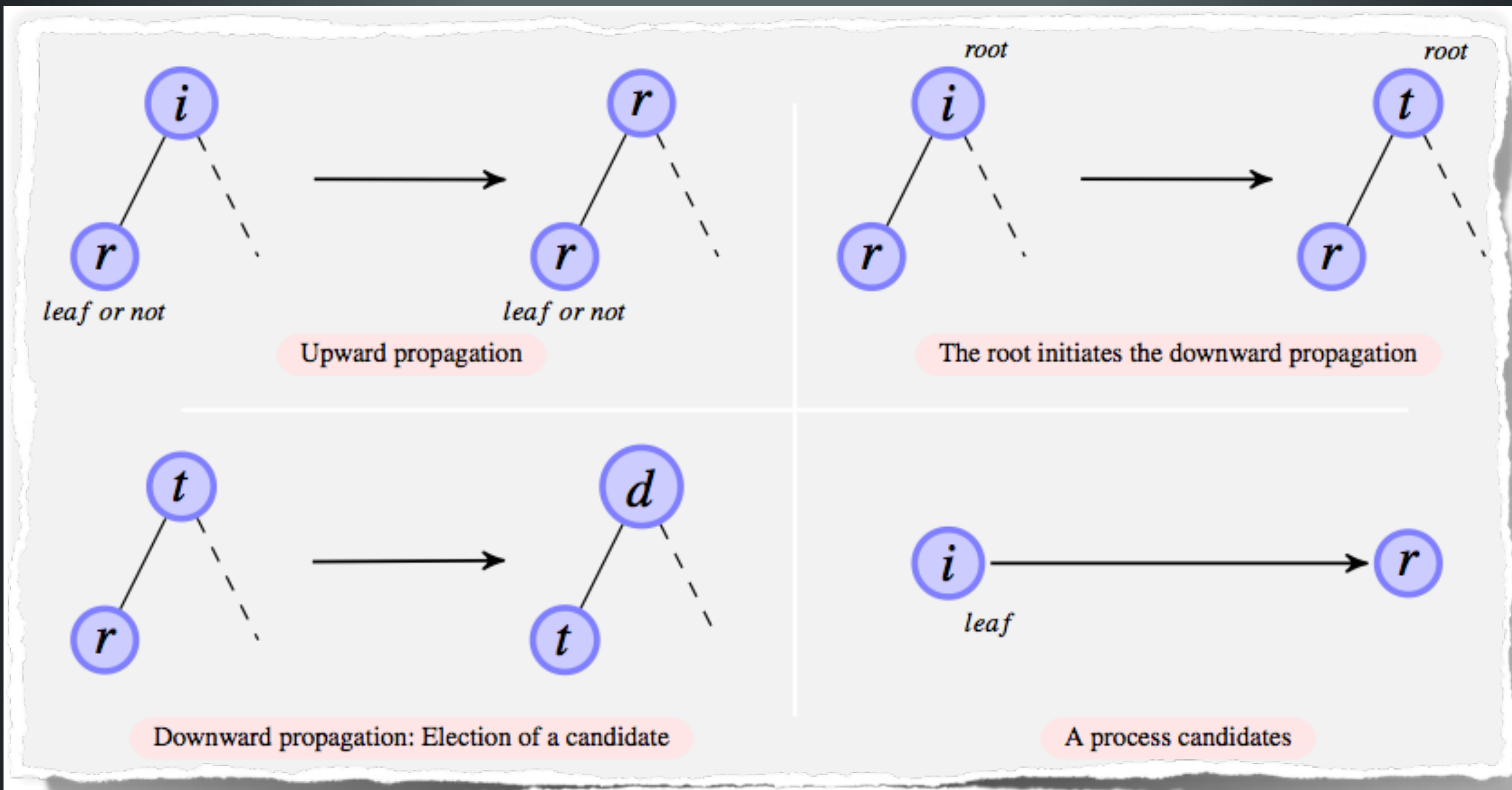
Parameterized System

Leader Election



Parameterized System

Leader Election



Universal condition

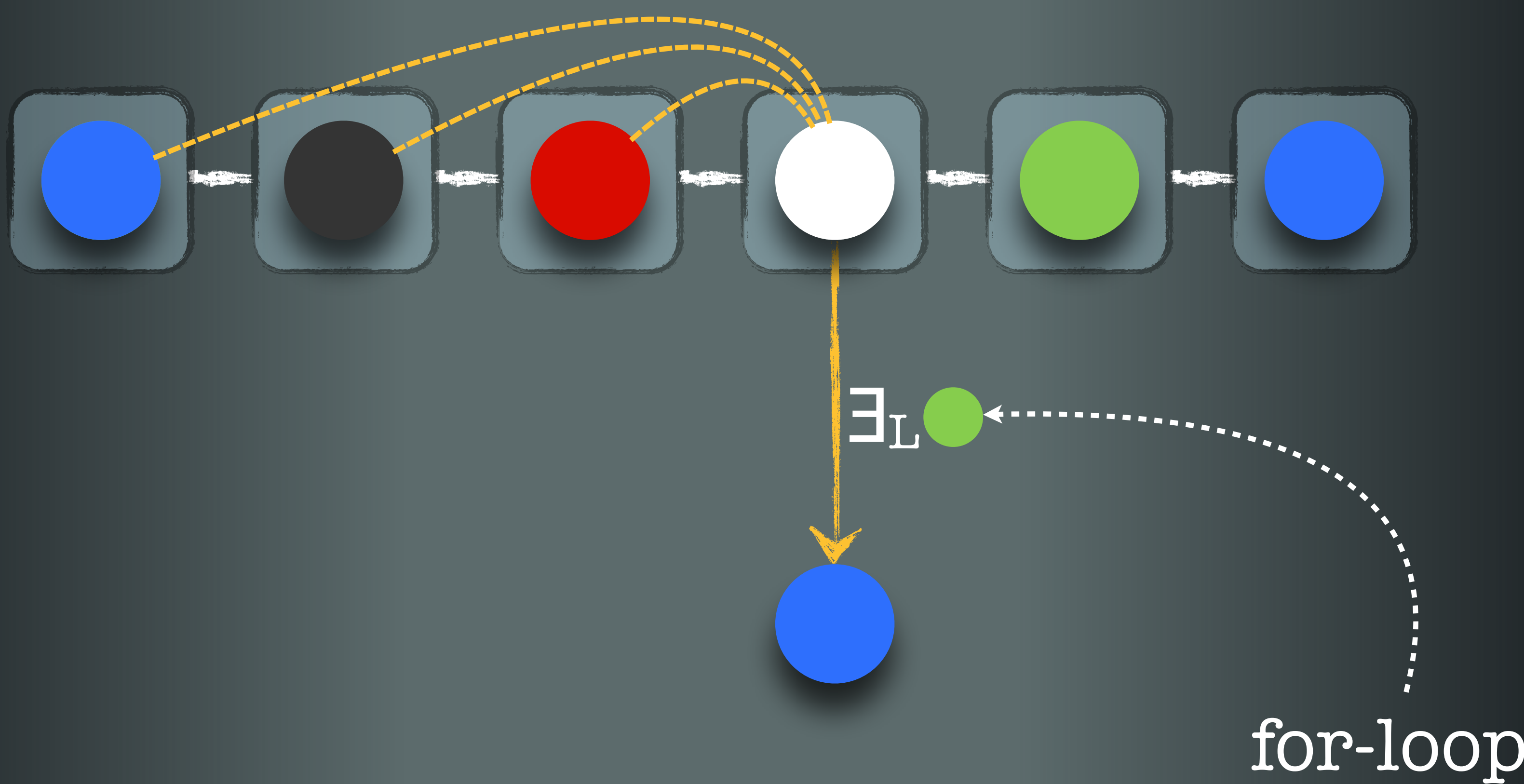
Existential condition

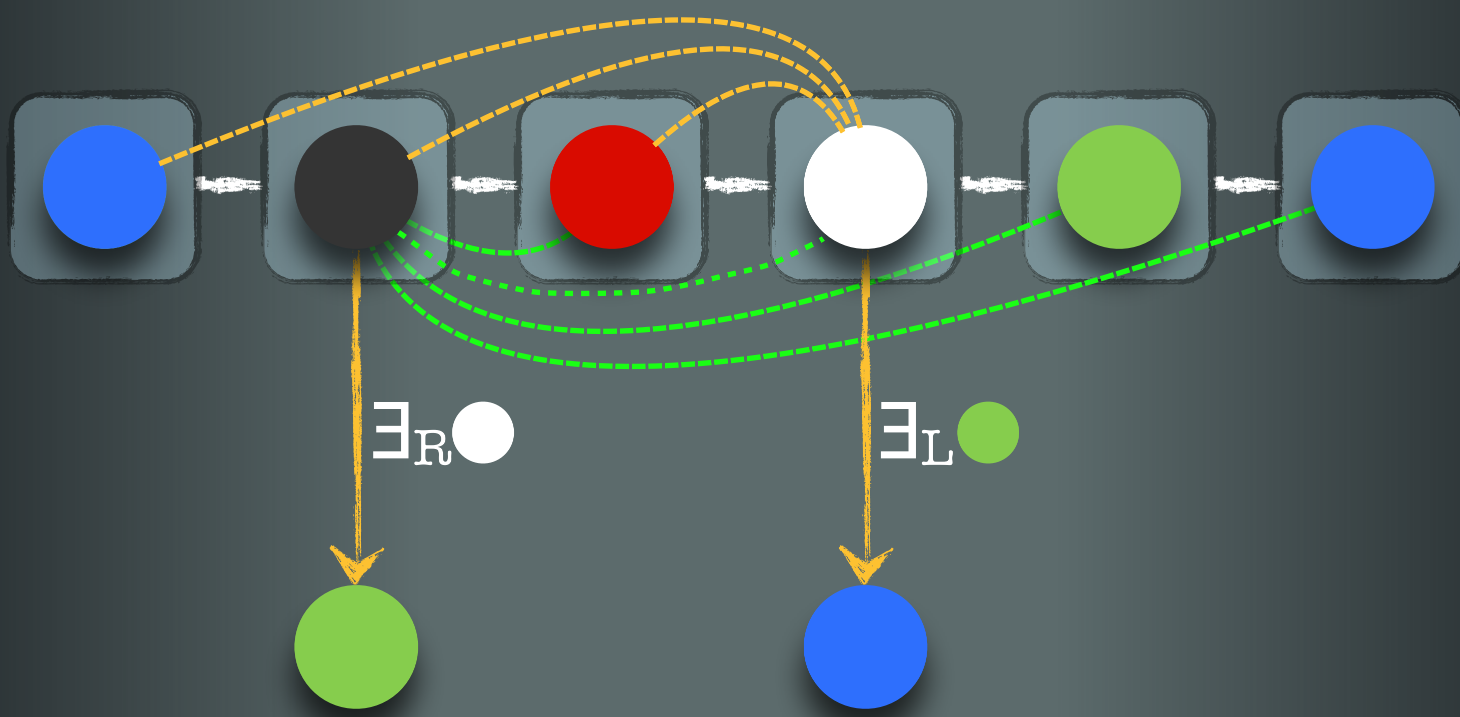
Non-atomic
global conditions

Paramete
System

Analysis
Abstra

Features

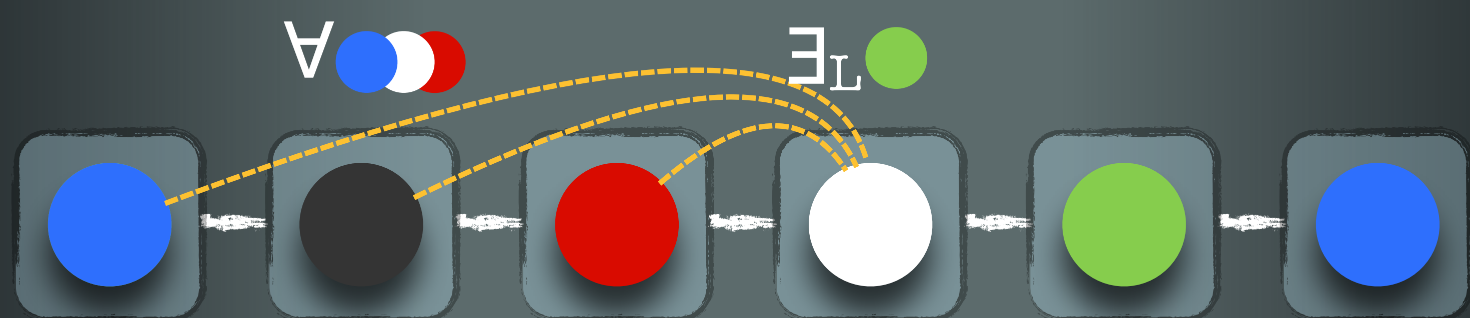



∃R
∃L

Paramete
System

Analysis
Abstra

Features

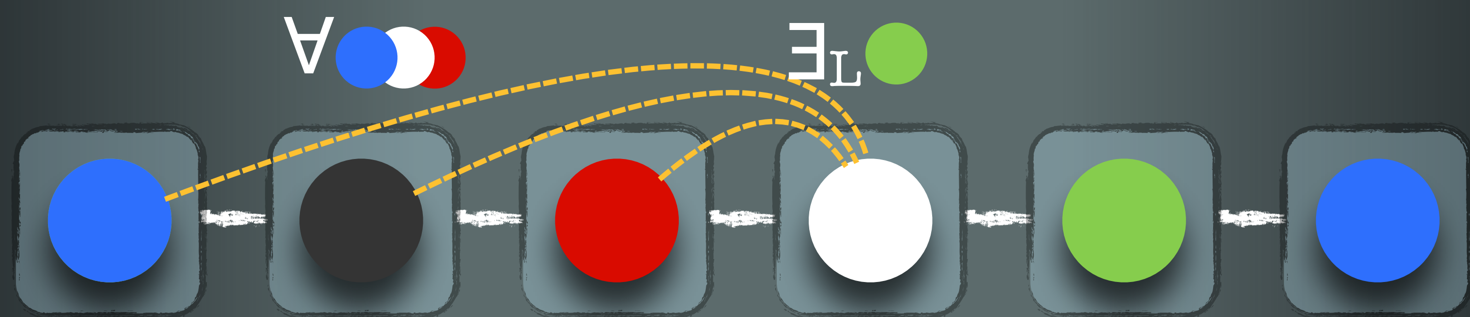


.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
?	?	✓	o	_	_
.	.	.	.	.	.
.	.	.	.	.	.

Parameter System

Analysis Abstraction

Features

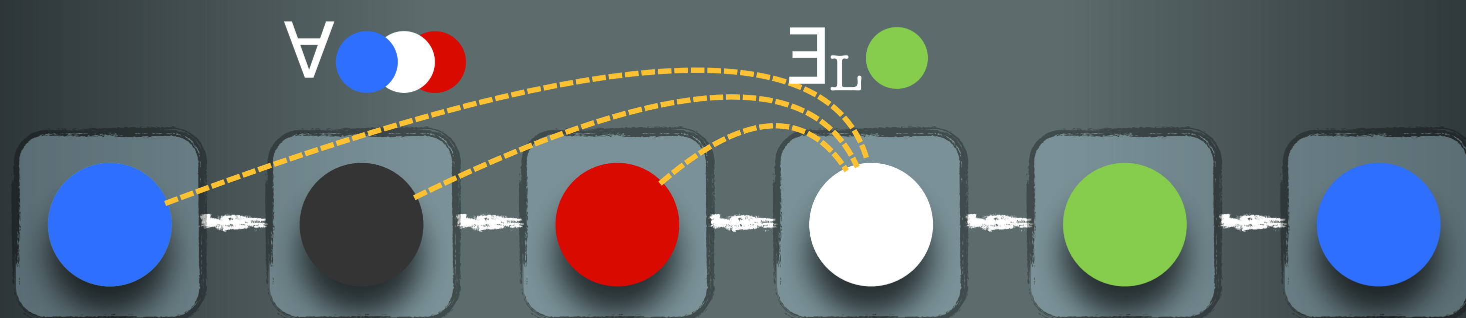


.	.	.	.	.	.
?	o	?	?	?	?
.	.	.	.	.	.
?	?	✓	o	_	_
.	.	.	.	.	.
.	.	.	.	.	.

Paramete System

Analysis
Abstrac

Features

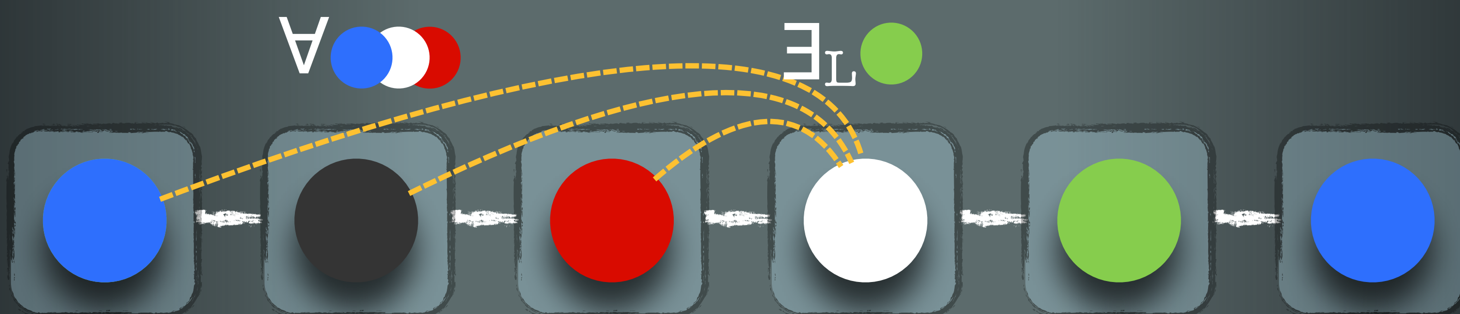


.	.	.	.	.	.
✓	o	✓	✓	✗	?
.	.	.	.	.	.
?	?	✓	o	_	_
.	.	.	.	.	.
.	.	.	.	.	.

Parameter System

Analysis Abstraction

Features

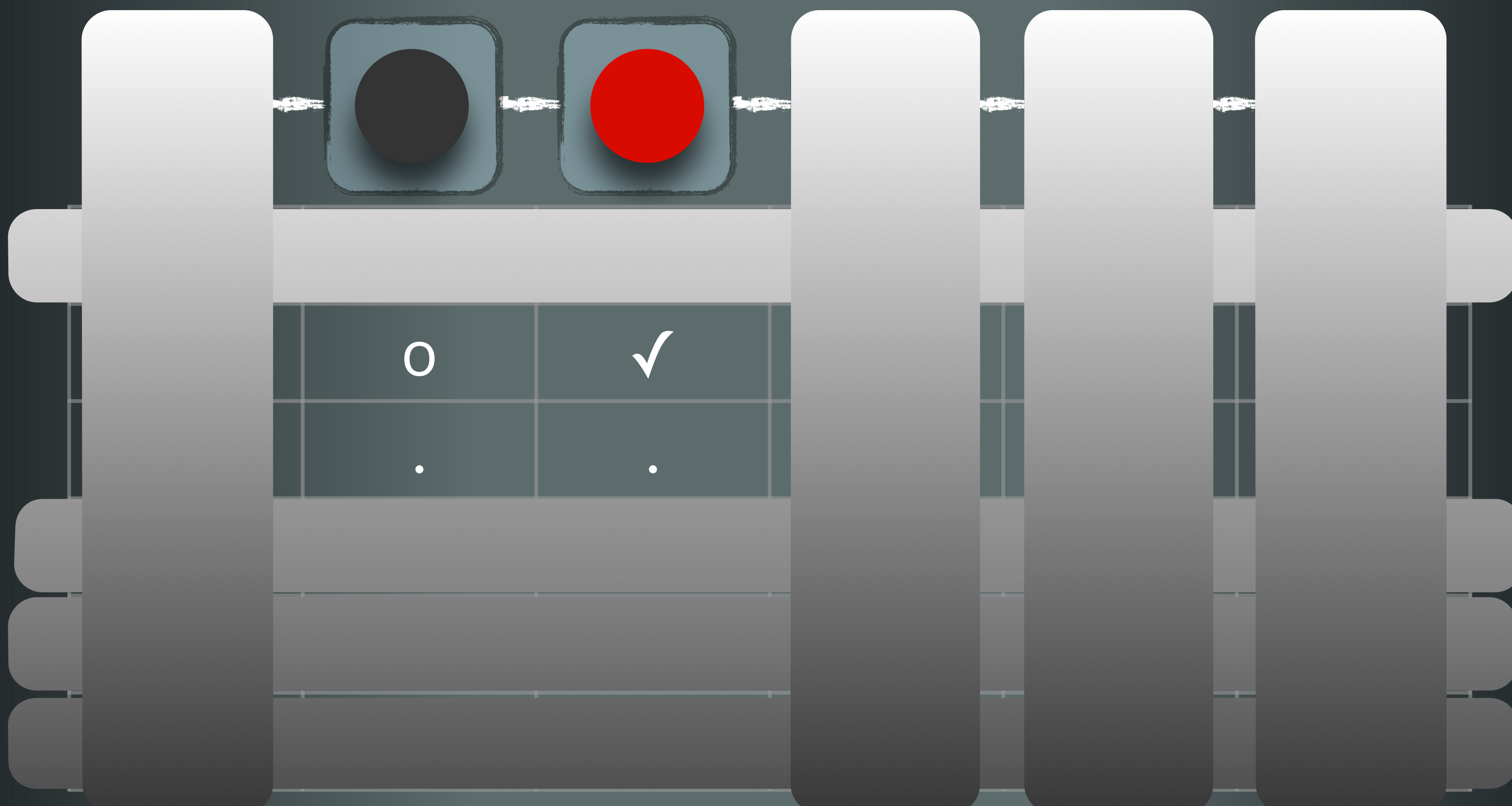


.	.	.	.	.	.
✓	o	✓	✓	✗	?
.	.	.	.	.	.
?	?	✓	o	_	_
.	.	.	.	.	.
.	.	.	.	.	.

Paramete… System

Analysis
Abstrac…

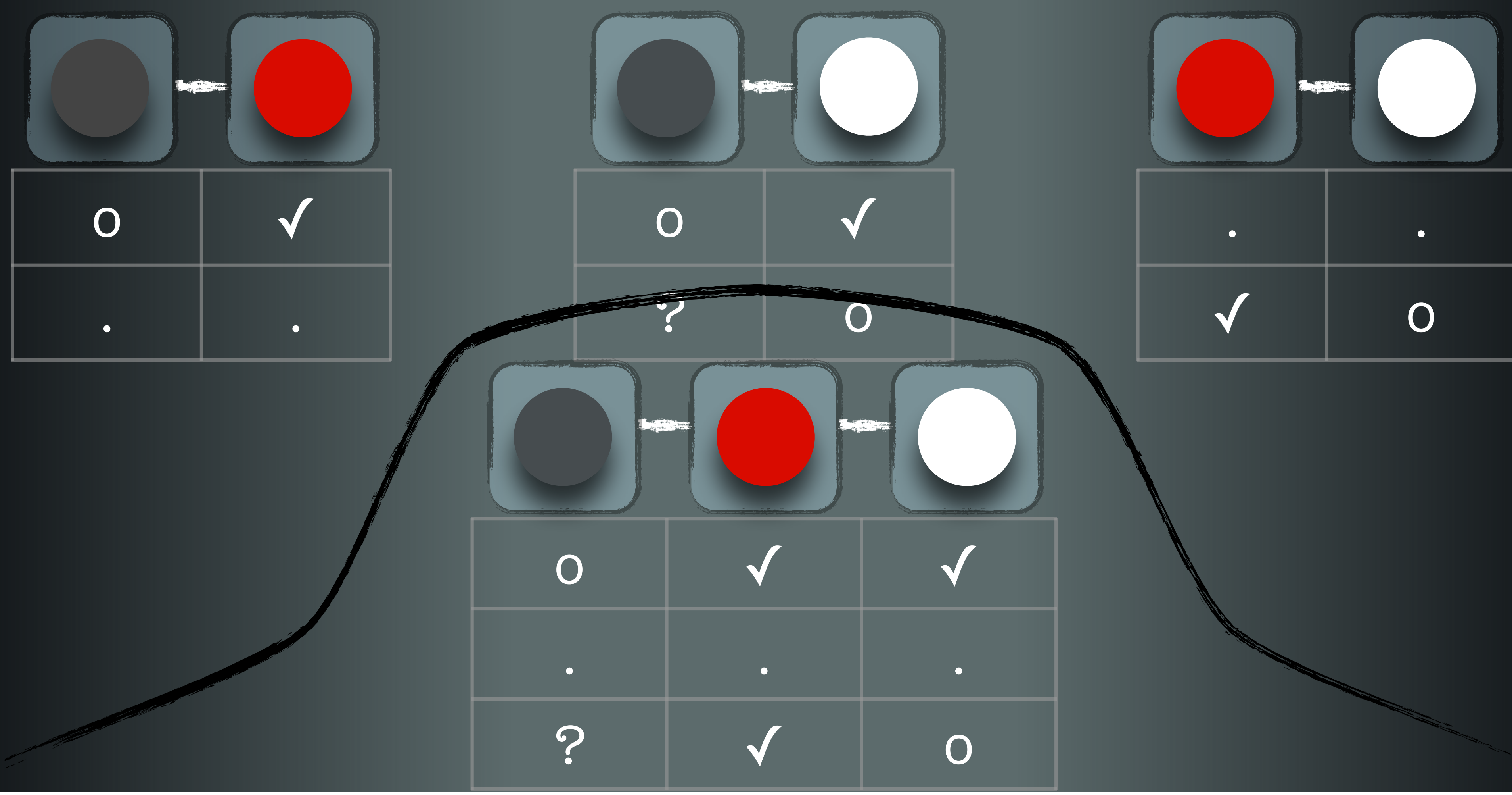
Features

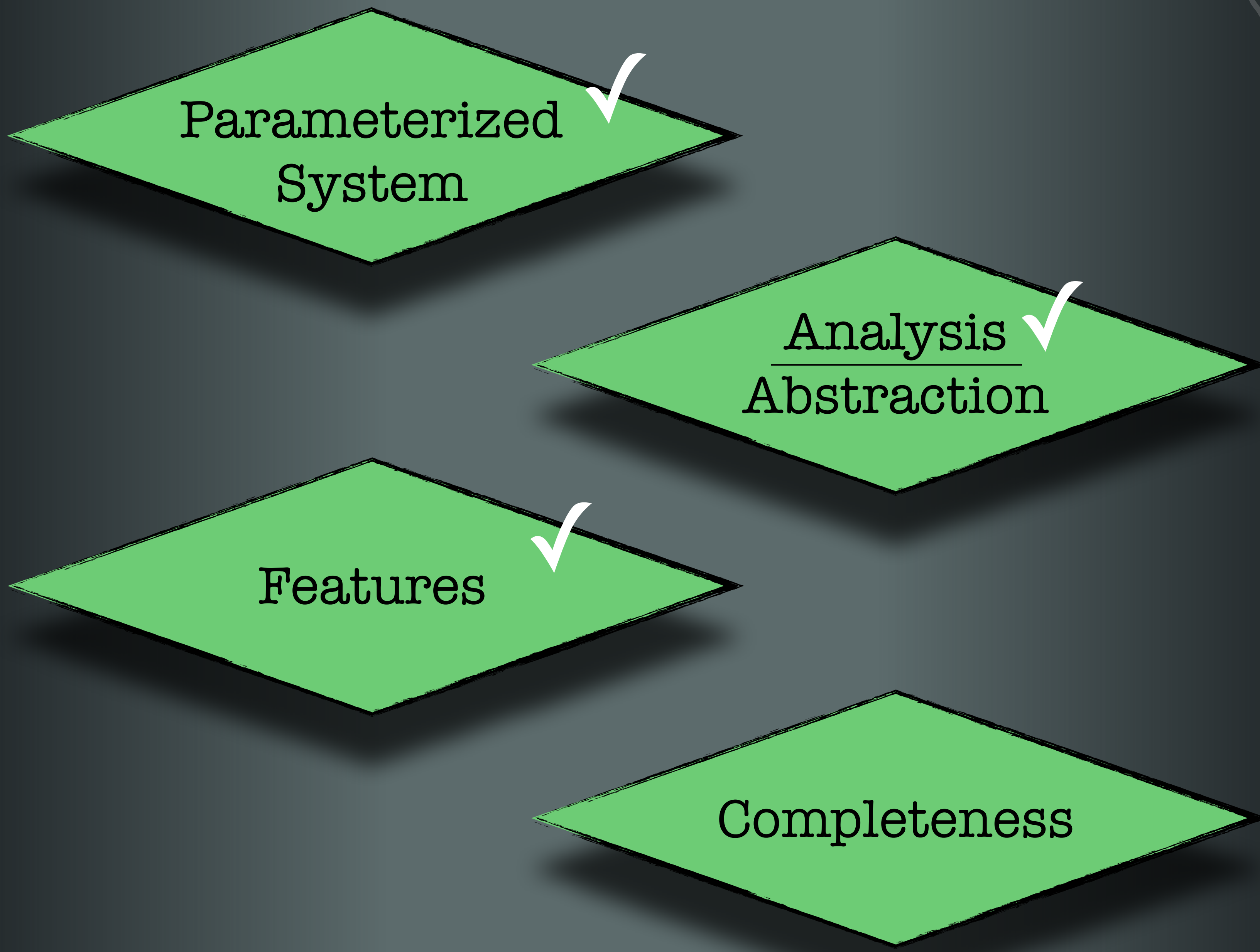


Paramete System

Analysis Abstrac

Features




Parameterized System
Analysis
Abstraction
Features
Completeness

Paramet
System

Analysis

Abstra

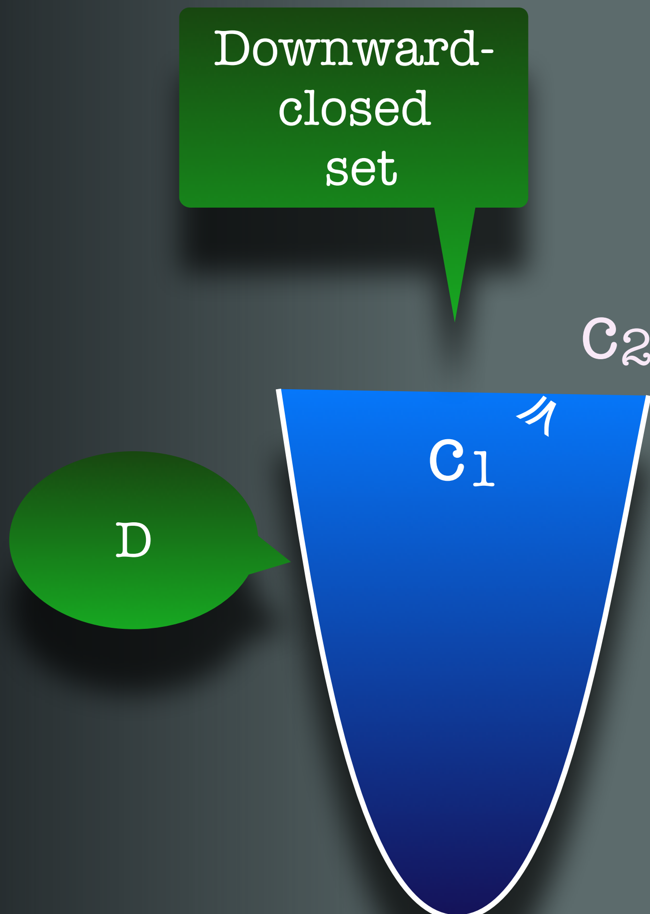
Features

Completeness

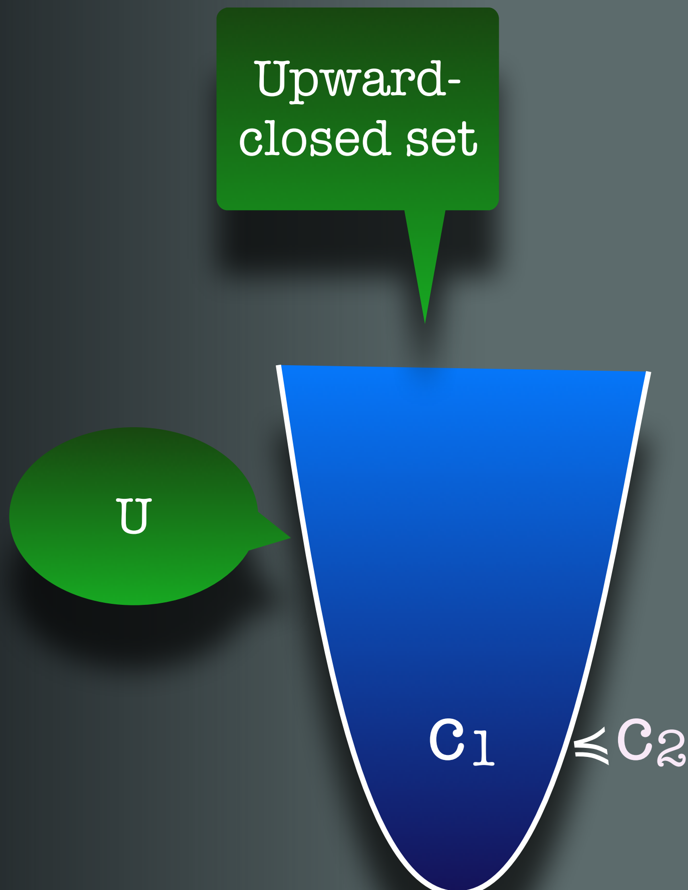
Well Quasi-Ordering (WQO)

$(A, \preccurlyeq)$

$a_0\ a_1\ a_2 \cdots\cdots \boxed{a_i} \cdots\cdots \boxed{\preccurlyeq} \cdots\cdots \boxed{a_j} \cdots$



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

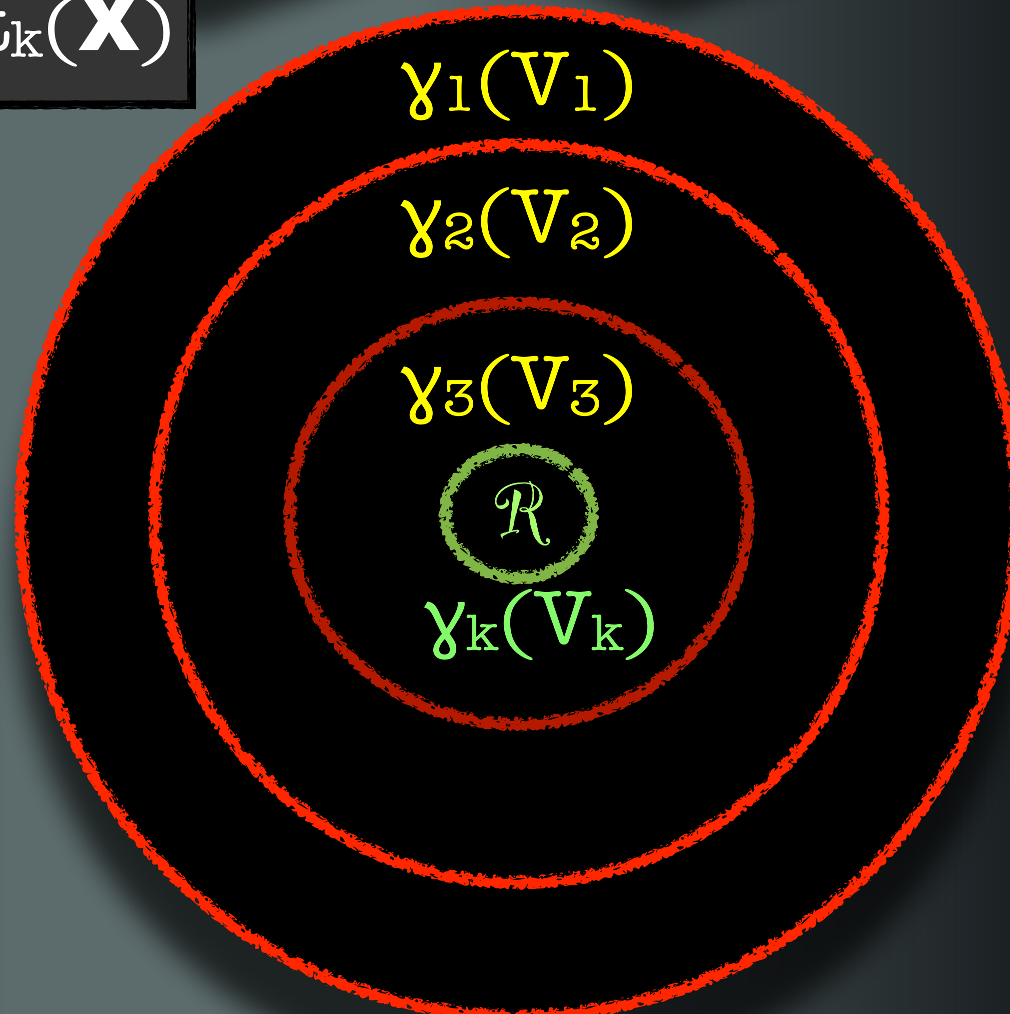


if $c_1 \in U$ and $c_1 \preccurlyeq c_2$ then $c_2 \in D$

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

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

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



if

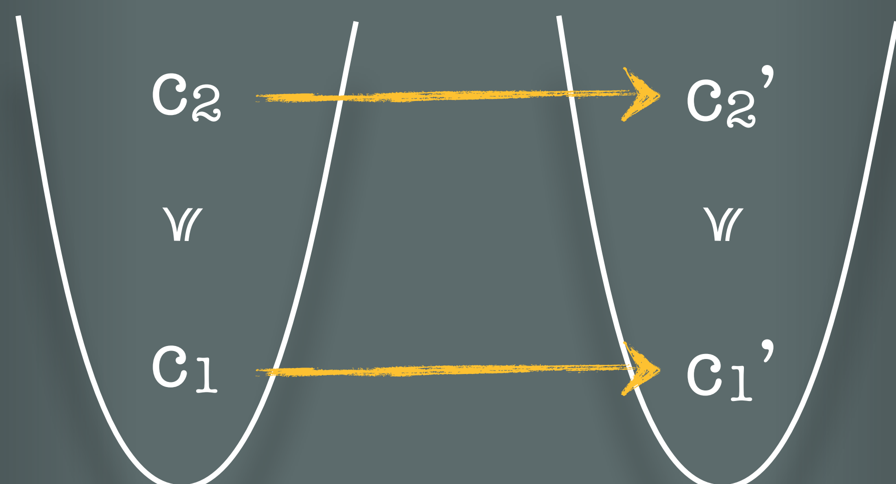
- $\mathcal{R}$ downward-closed
- $\mathcal{R}$ inductive

then

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

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

if $\longrightarrow$ is monotonic



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

if $\longrightarrow$ is monotonic

Class of

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

Theorem

if System is WQO and Safe

Procedure guaranteed to terminate

Procedure is complete

Applications: Petri Nets

Parameterized System

Analysis

Abstraction

Features

Completeness

	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

Parameteri
System

Analysis
Abstractio

Feat

Completeness

Future Challenges

Shape Analysis

Future Work

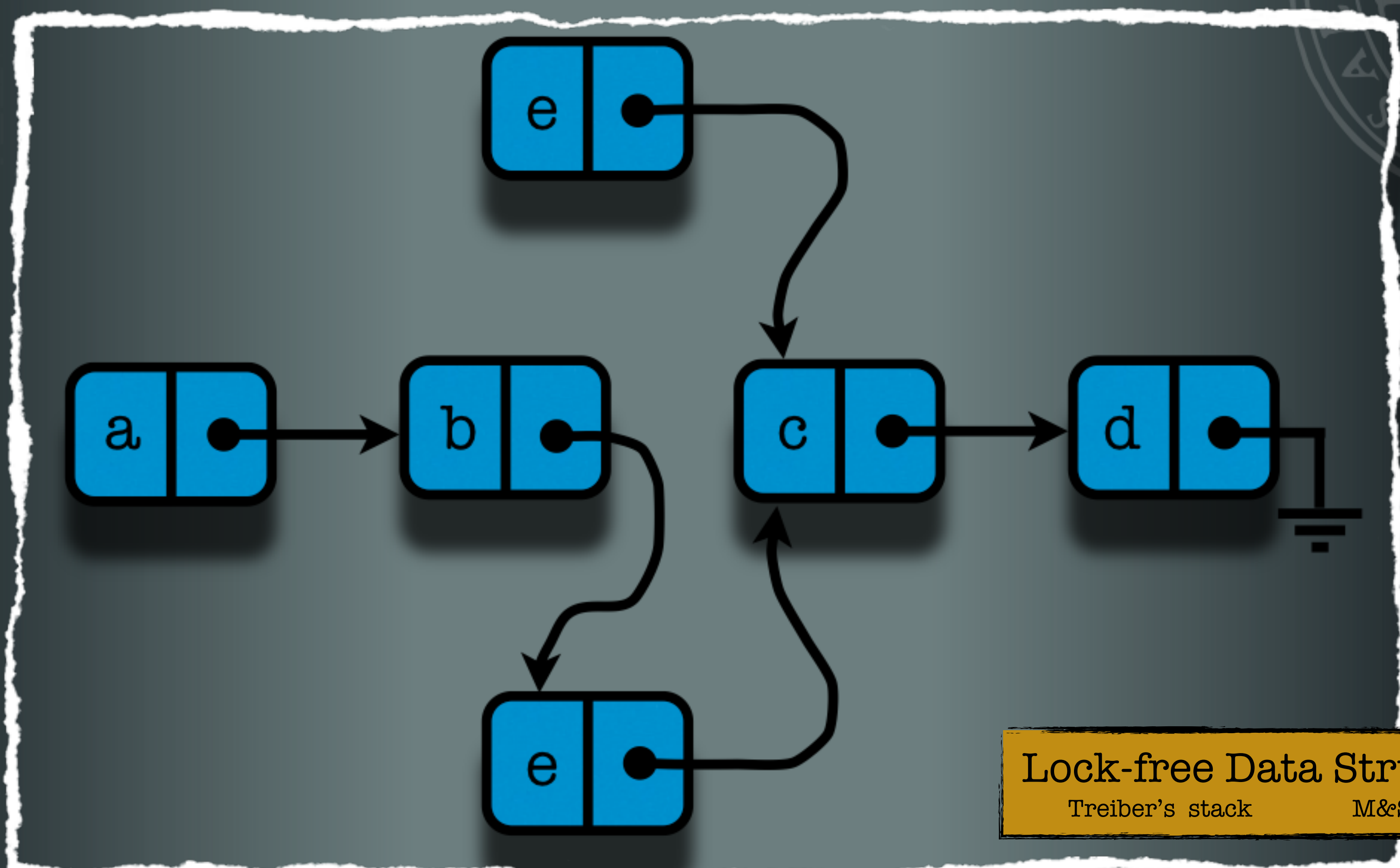
Contexts

CEGAR

Shape analysis

Future Work

Shape analysis



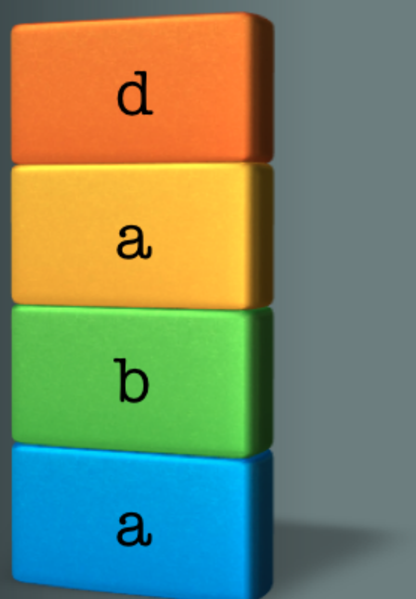
Lock-free Data Structures

Treiber's stack M&S's queue

Future Work

Shape analysis

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



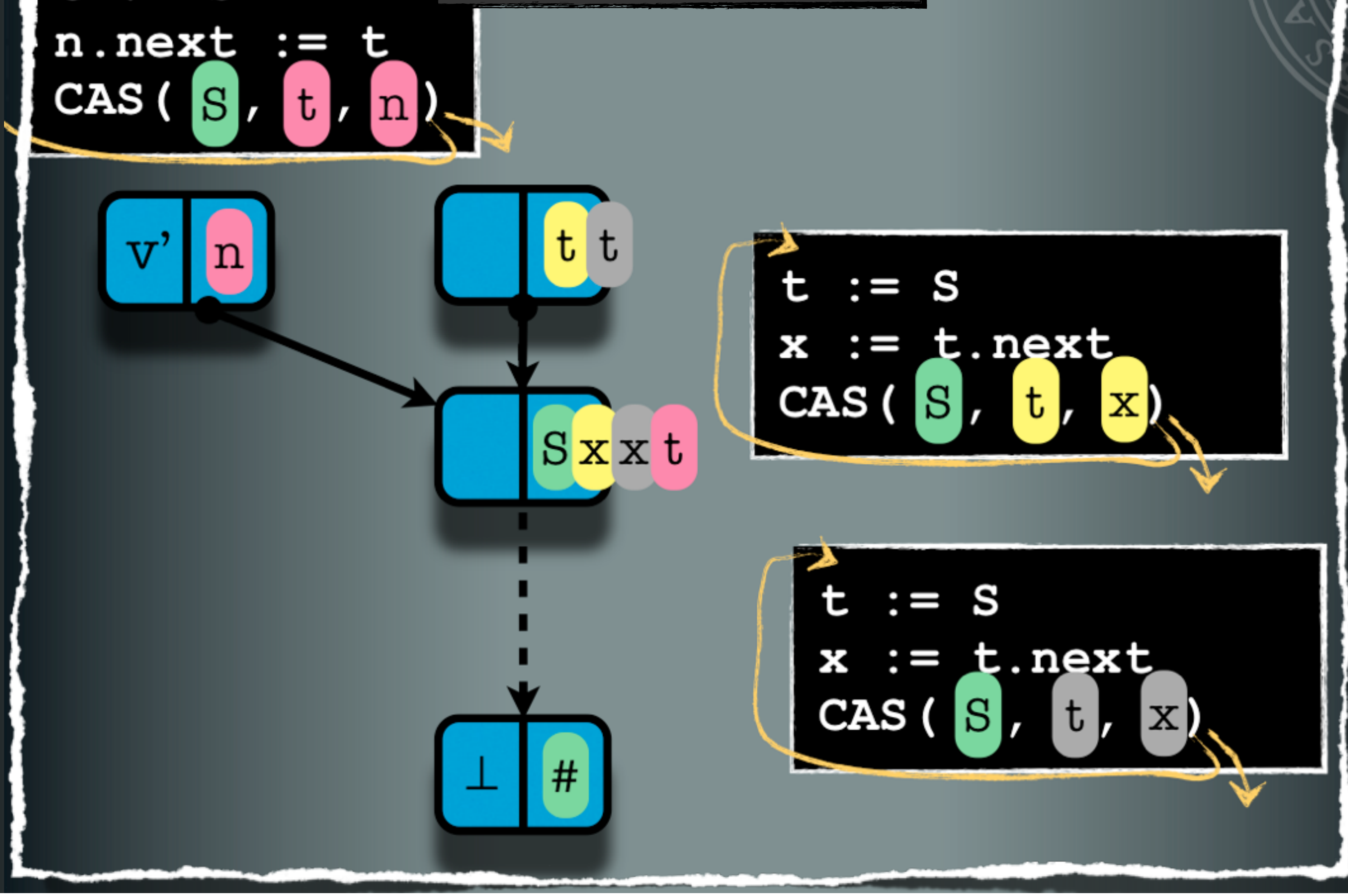
Parameteri…
System…

Future Work

ontexts

GAR

Shape analysis

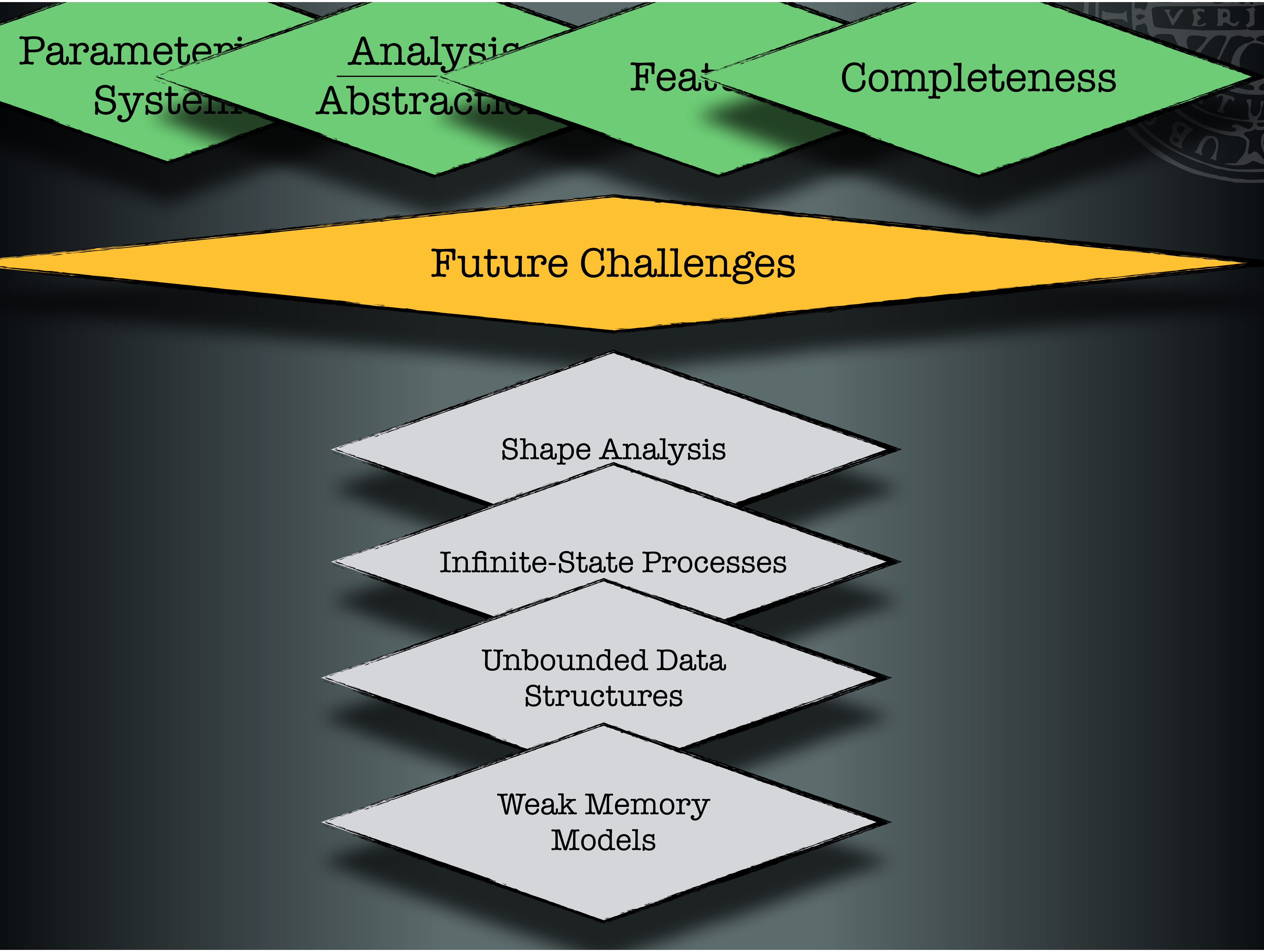


Future Work

Contexts

CEGAR

Shape analysis


Completeness
Future Challenges
Shape Analysis
Infinite-State Processes
Unbounded Data Structures
Weak Memory Models

S. ACADEMIAE UBSALIENSIS

GRATIAE

VERITAS

NATURAE