

# CEGAR for Communicating fifo Systems

Alexander Heußner (LaBRI, Bordeaux)

joint work with Tristan Le Gall (ULB Bruxelles) and Grégoire Sutre (LaBRI)

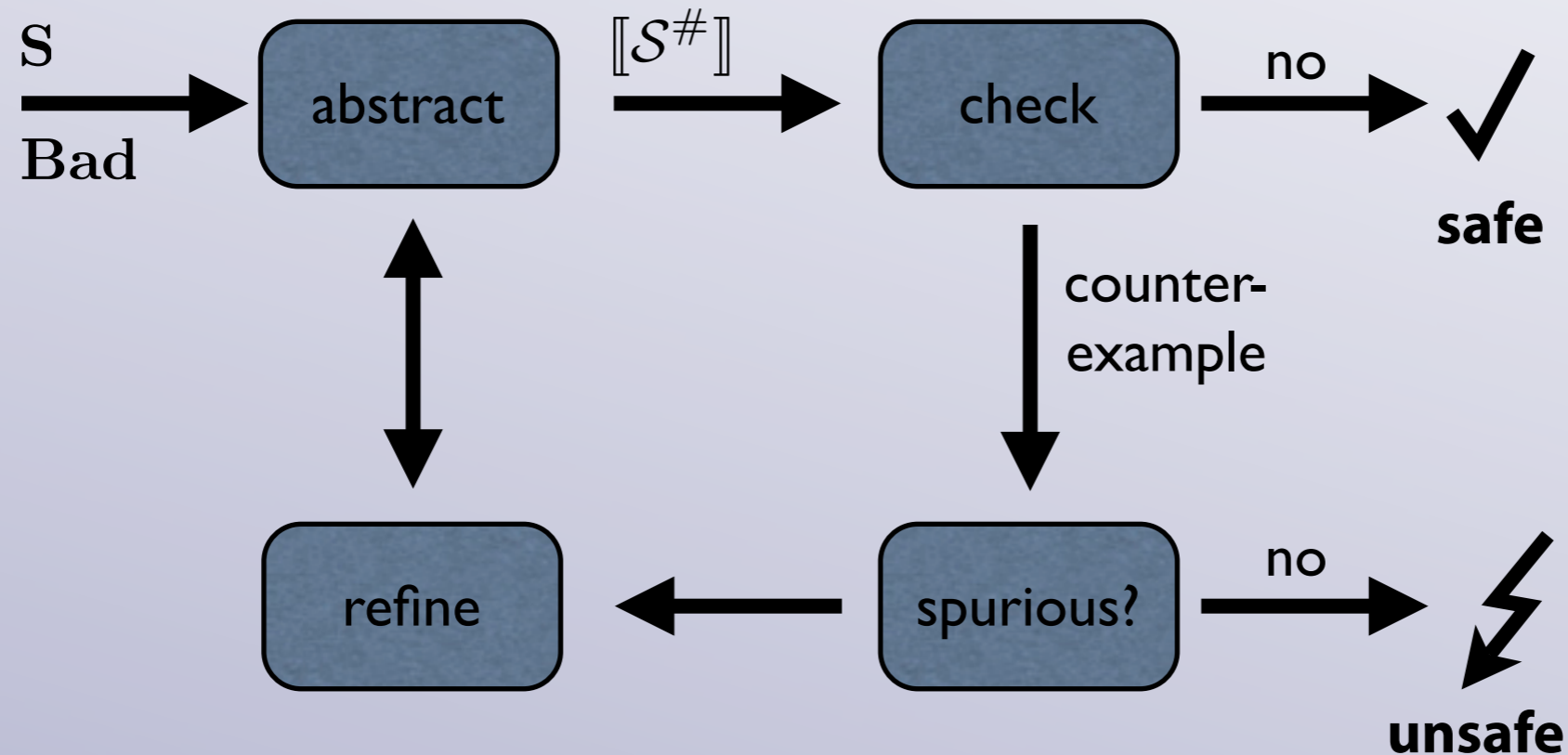
# Communication Protocols



- verify *safety* (system does not exhibit certain "bad" behavior)
- independent peers
- *asynchronous* communication
- reliable *unbounded* channels
- *state-based* protocols

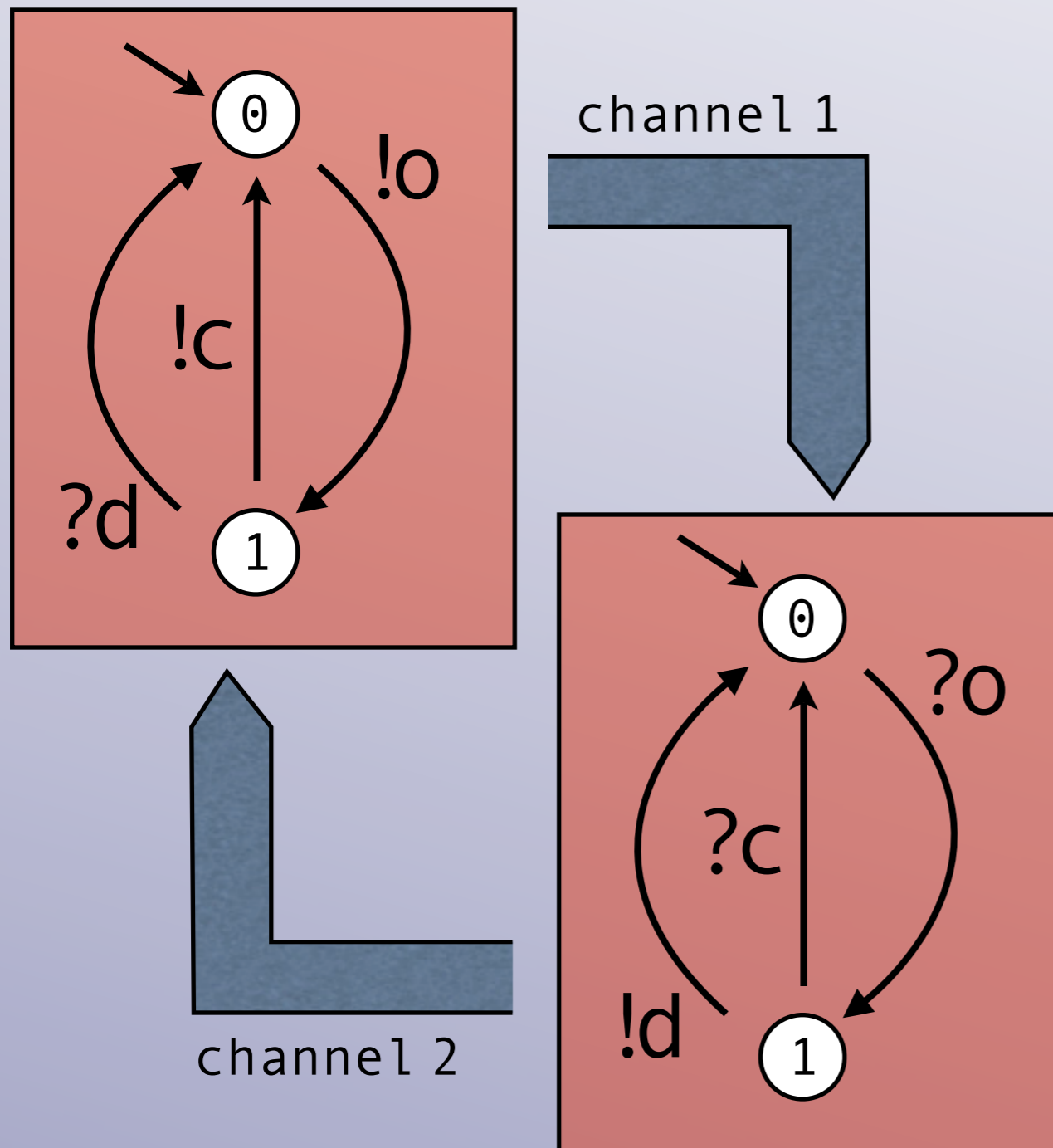
# CEGAR:

counterexample guided abstraction refinement



- successfully applied in software & hardware model checking tools (BLAST, SLAM,...)
- works for finite abstractions of infinite models (hybrid systems,...)

# Communicating fifo System:



A *communicating fifo system* is given by

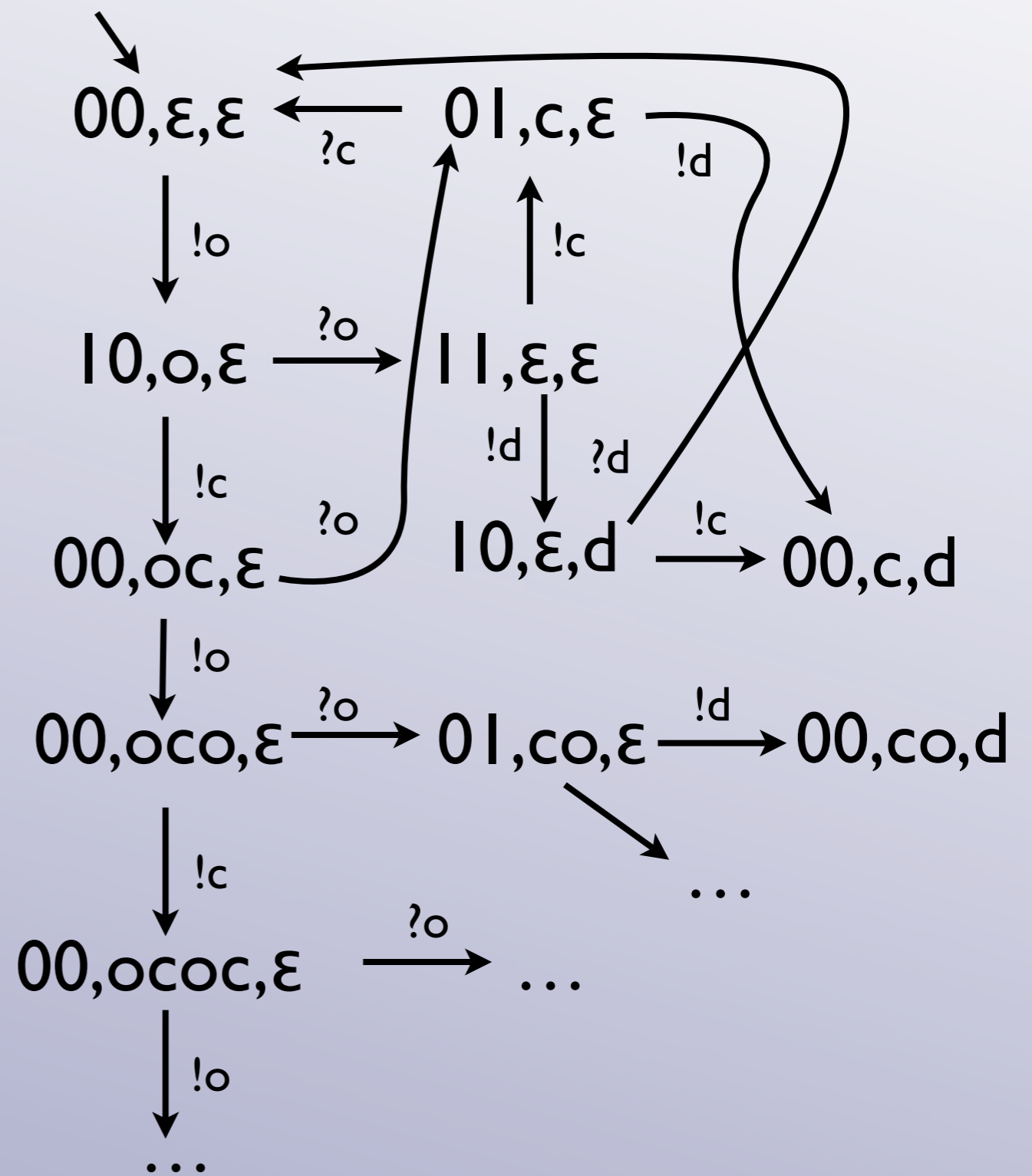
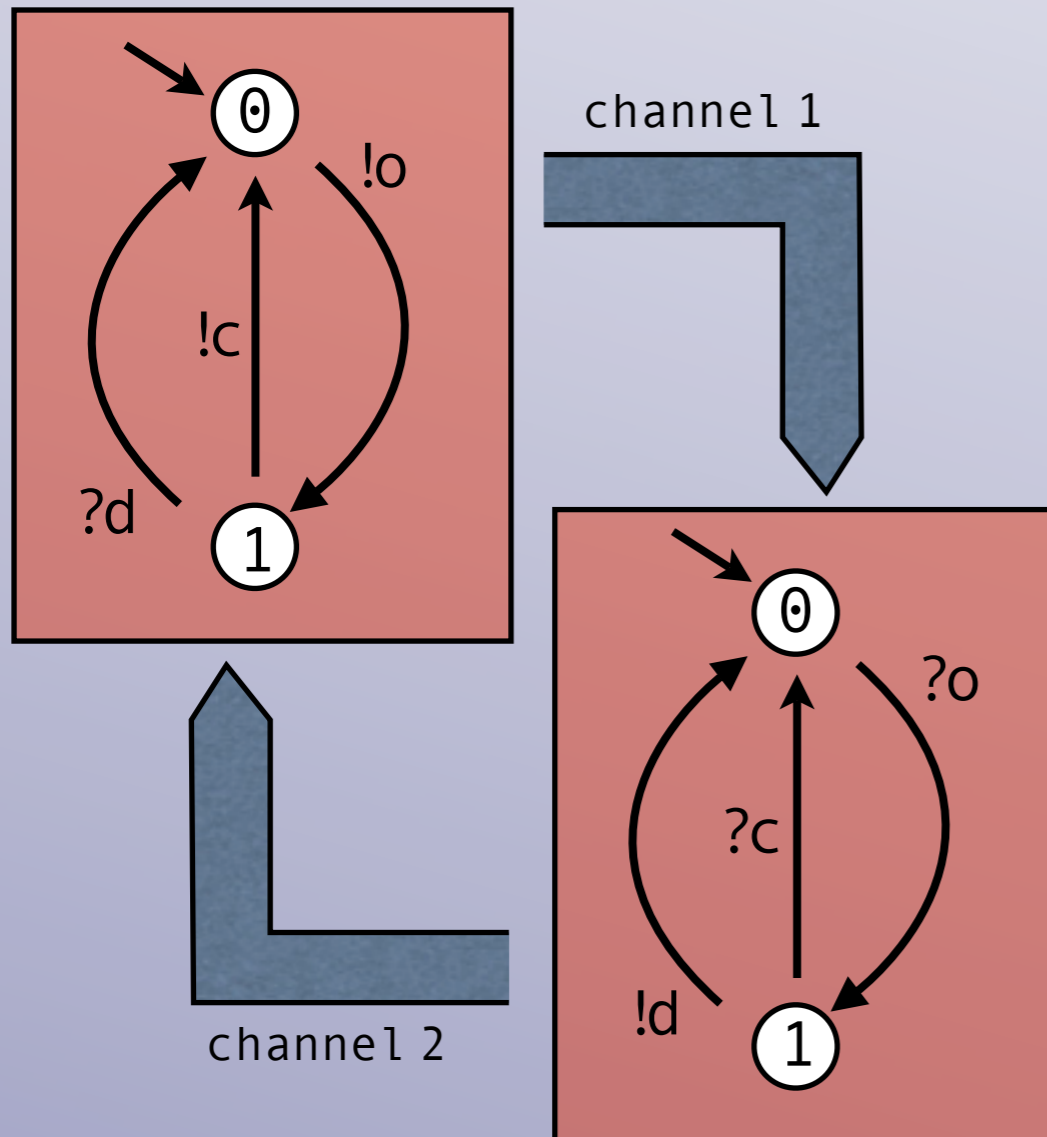
$$S = \langle Q, I, M, C, \Delta \rangle$$

with:

- a finite set of states  $Q$
- initial states  $I \subseteq Q$
- finite message alphabet  $M$
- finite set of channels  $C$
- transition relation  $\Delta$

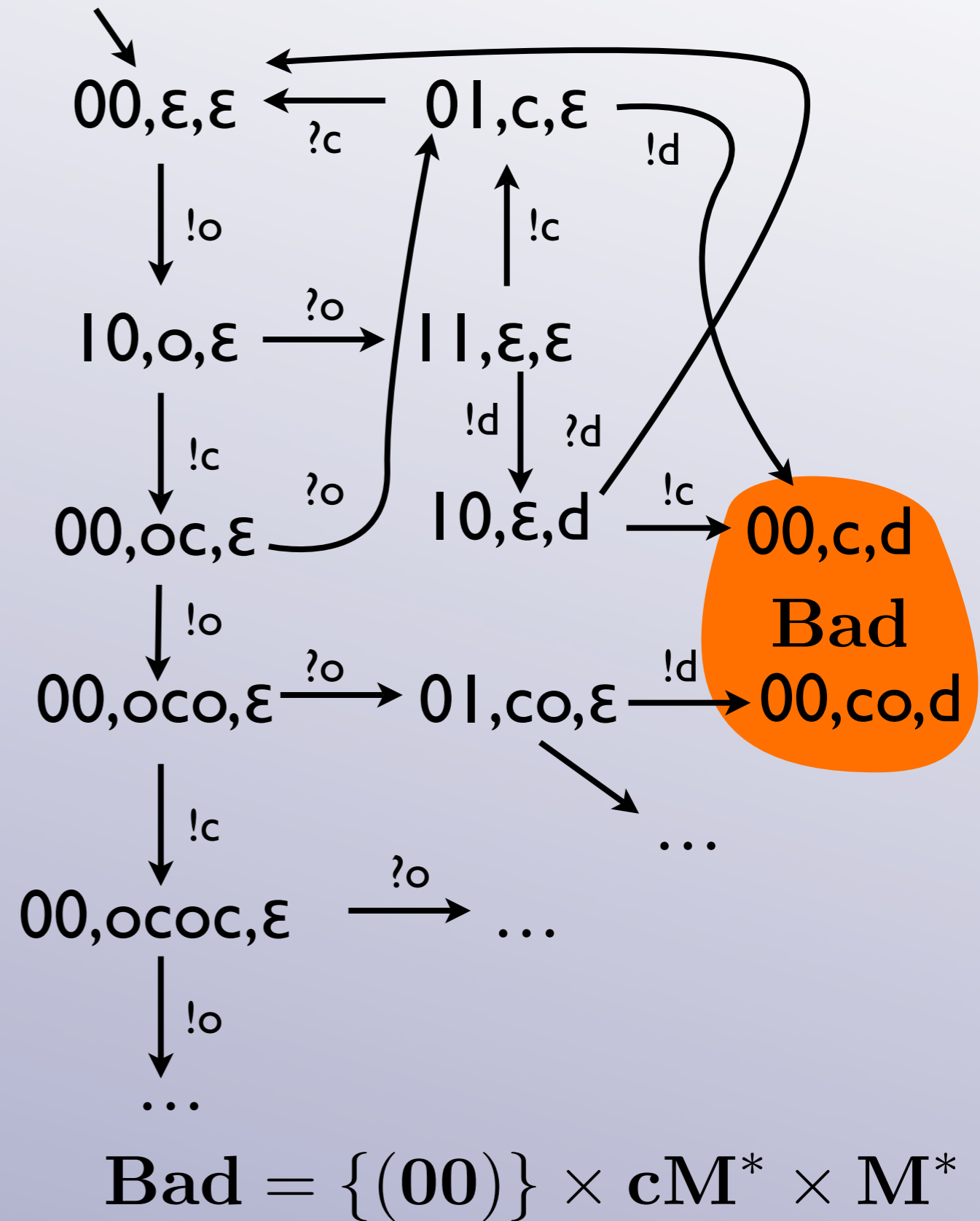
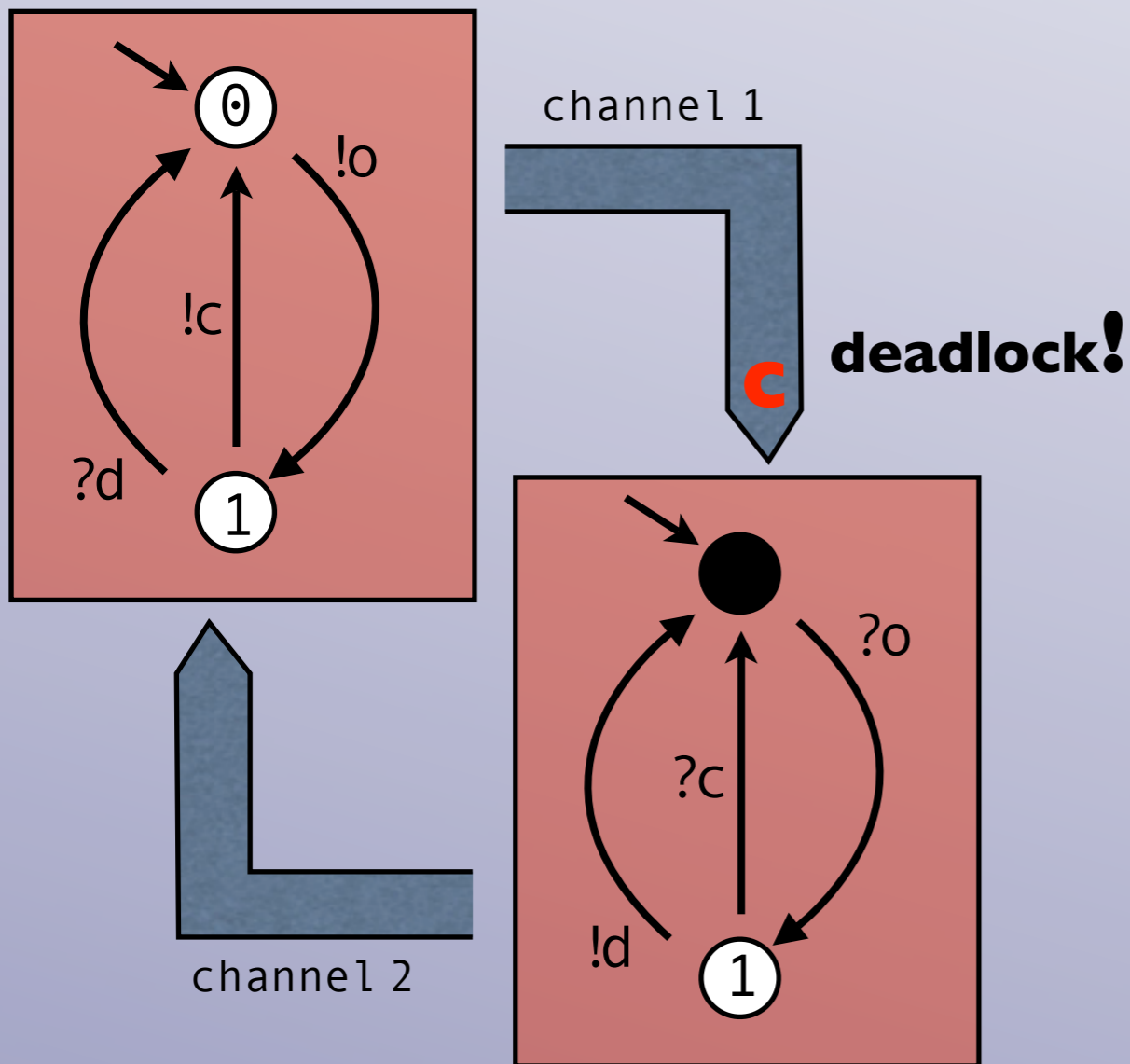
# operational semantics:

The behaviour of  $S$  is given as *infinite labeled transition system*:

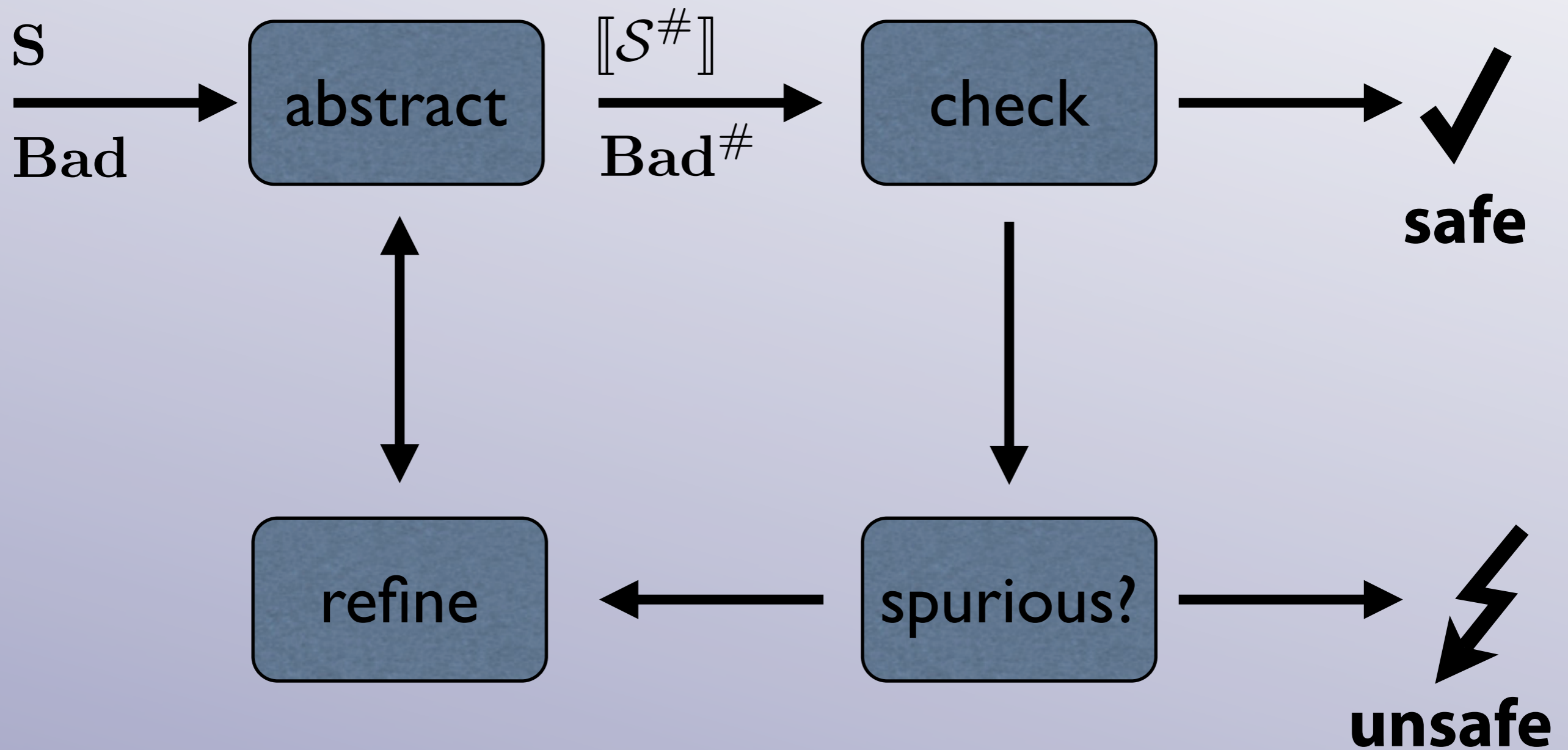


# safety verification:

**Safe** iff there exists no run that leads to **Bad**



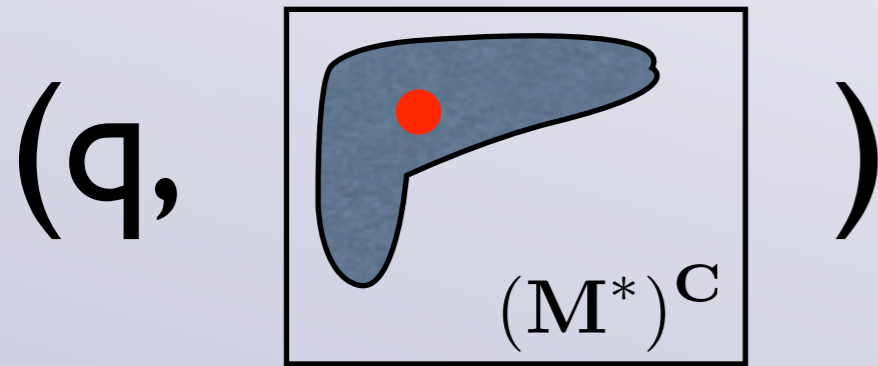
# CEGAR-loop:



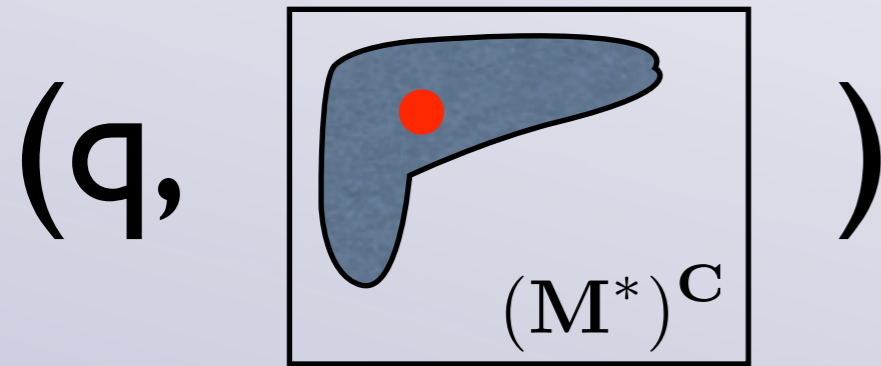
# Abstraction: configurations

$(q, w_1, \dots, w_n)$

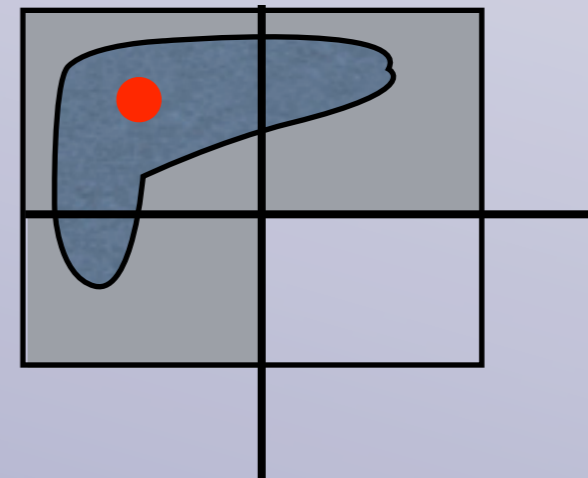
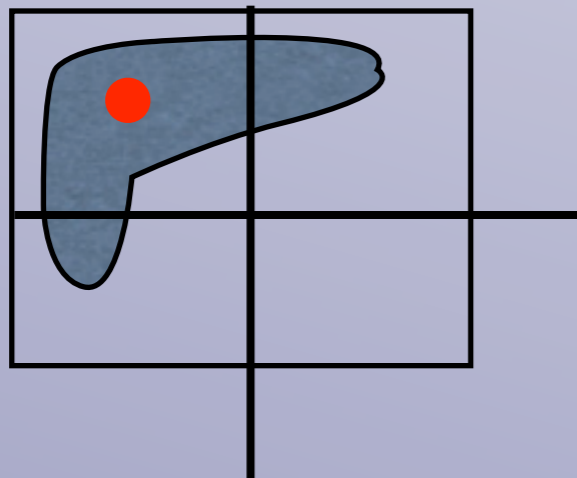
# Abstraction: configurations



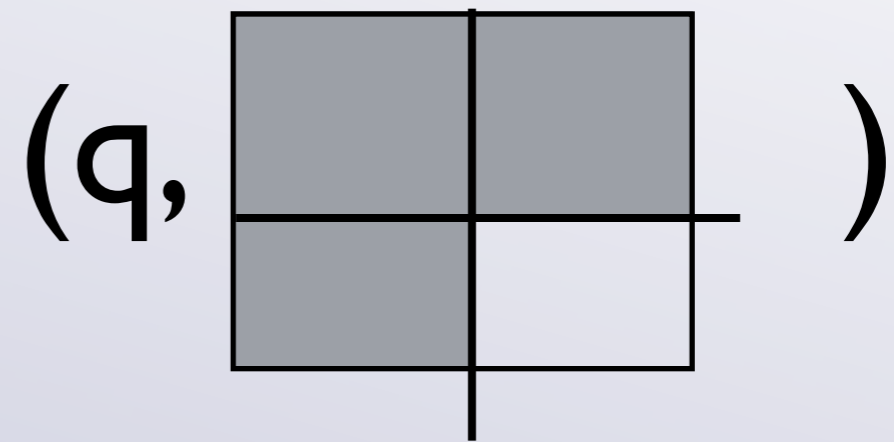
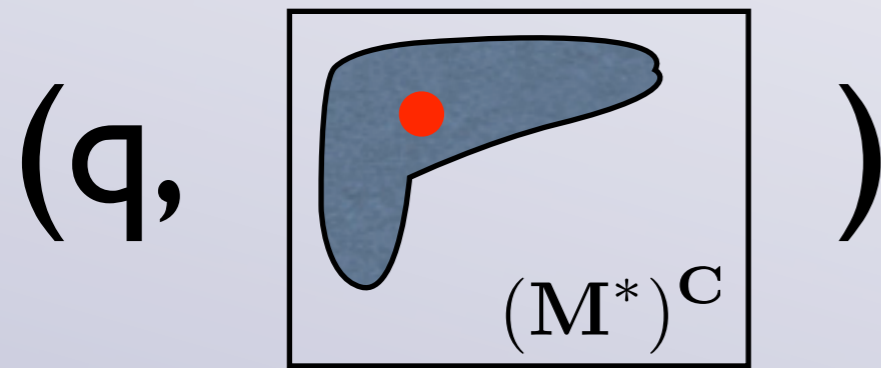
# Abstraction: configurations



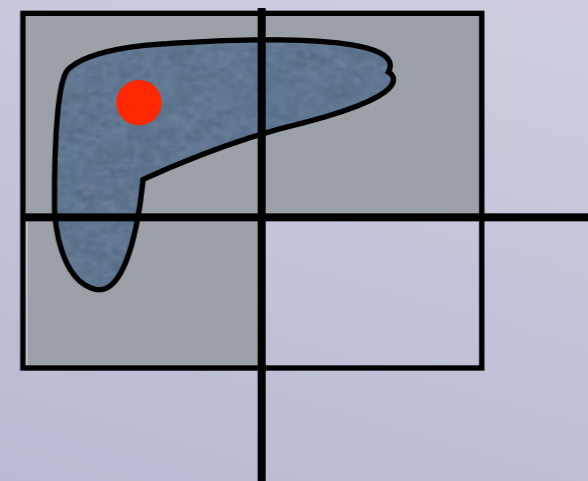
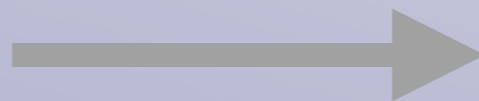
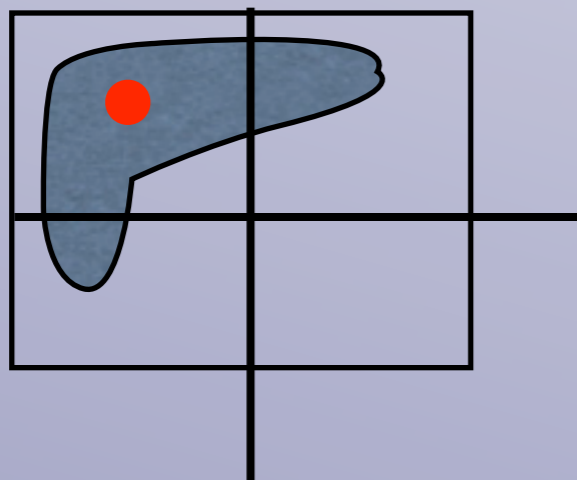

(finite)  
Partition of  
 $(M^*)^C$

A thick, grey arrow pointing downwards from the top diagram to the bottom-left diagram.

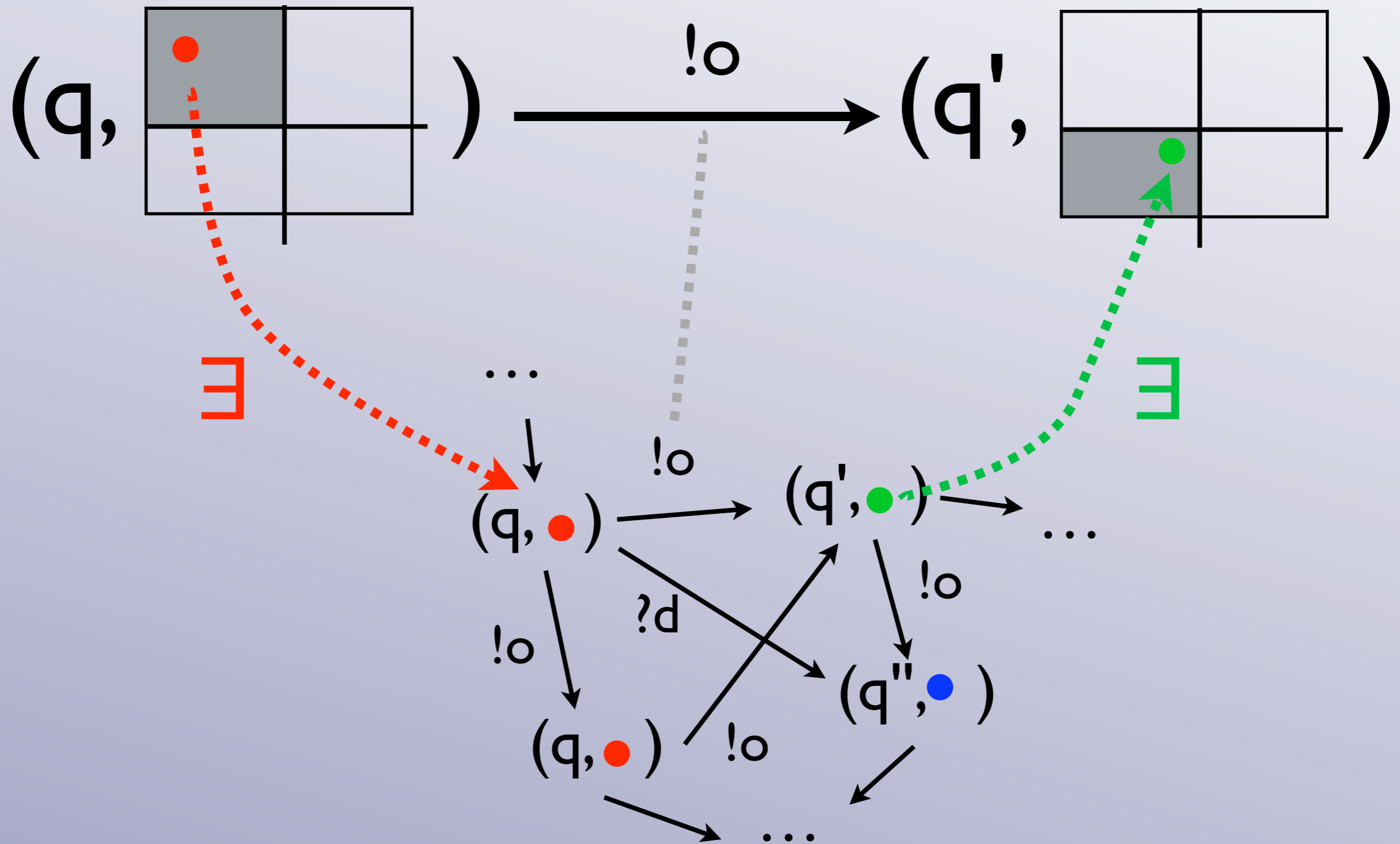
# Abstraction: configurations



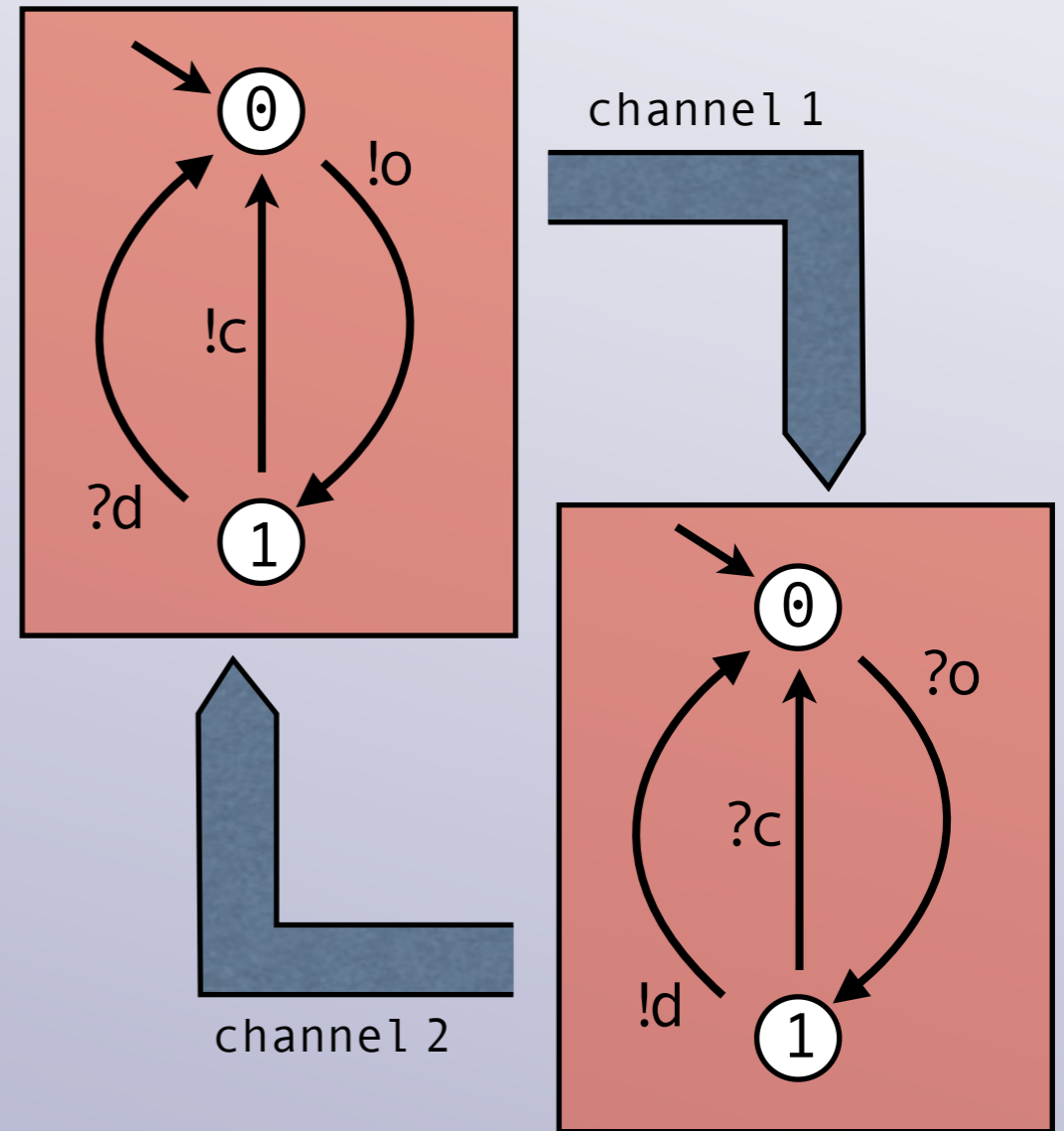
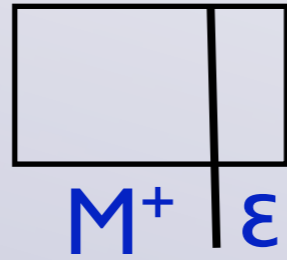
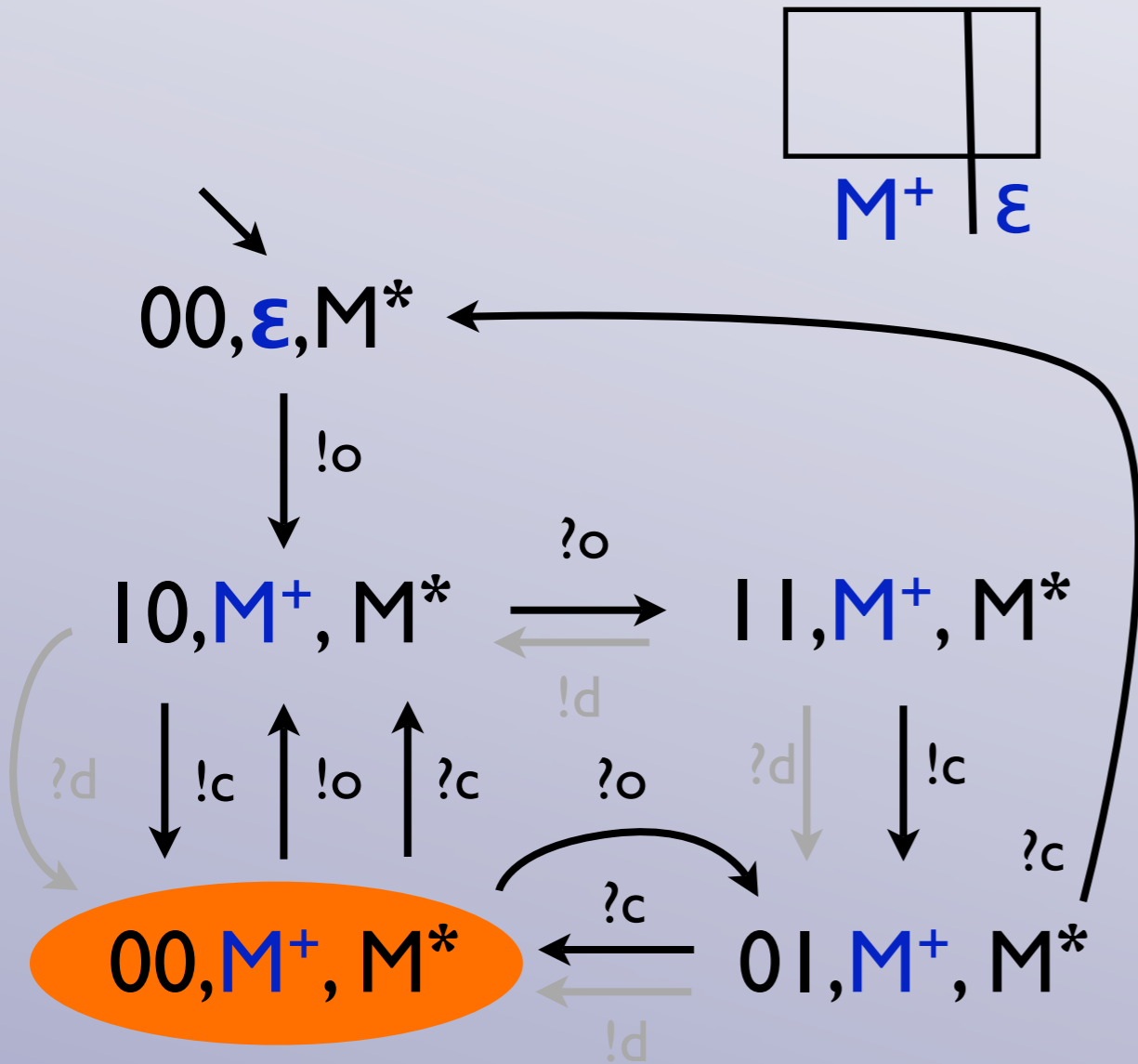
(finite)  
Partition of  
 $(M^*)^C$



# Abstraction: transitions

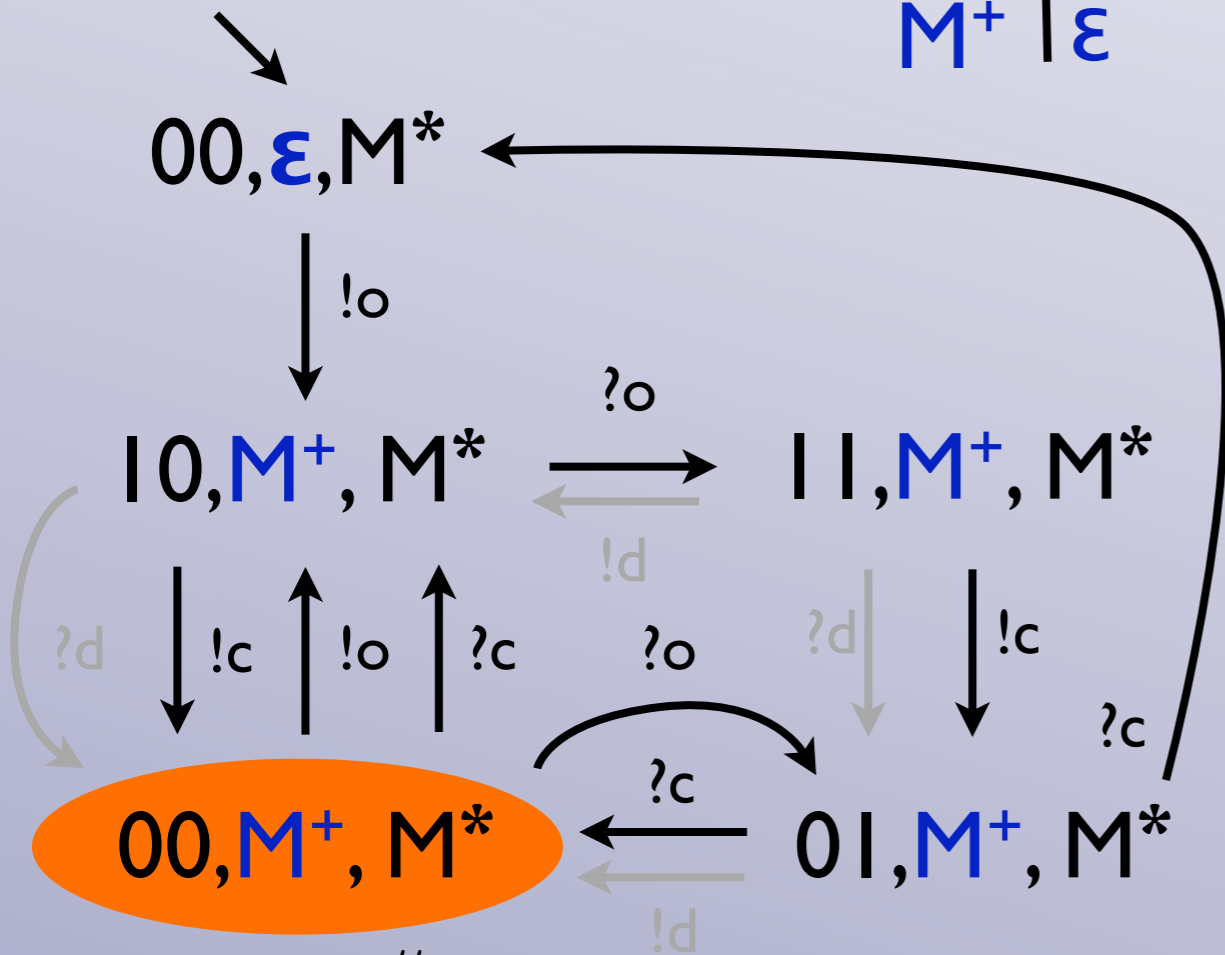
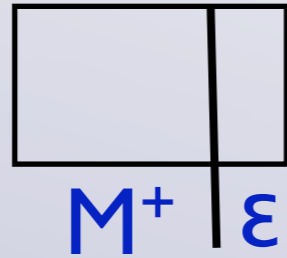


# abstract safety:

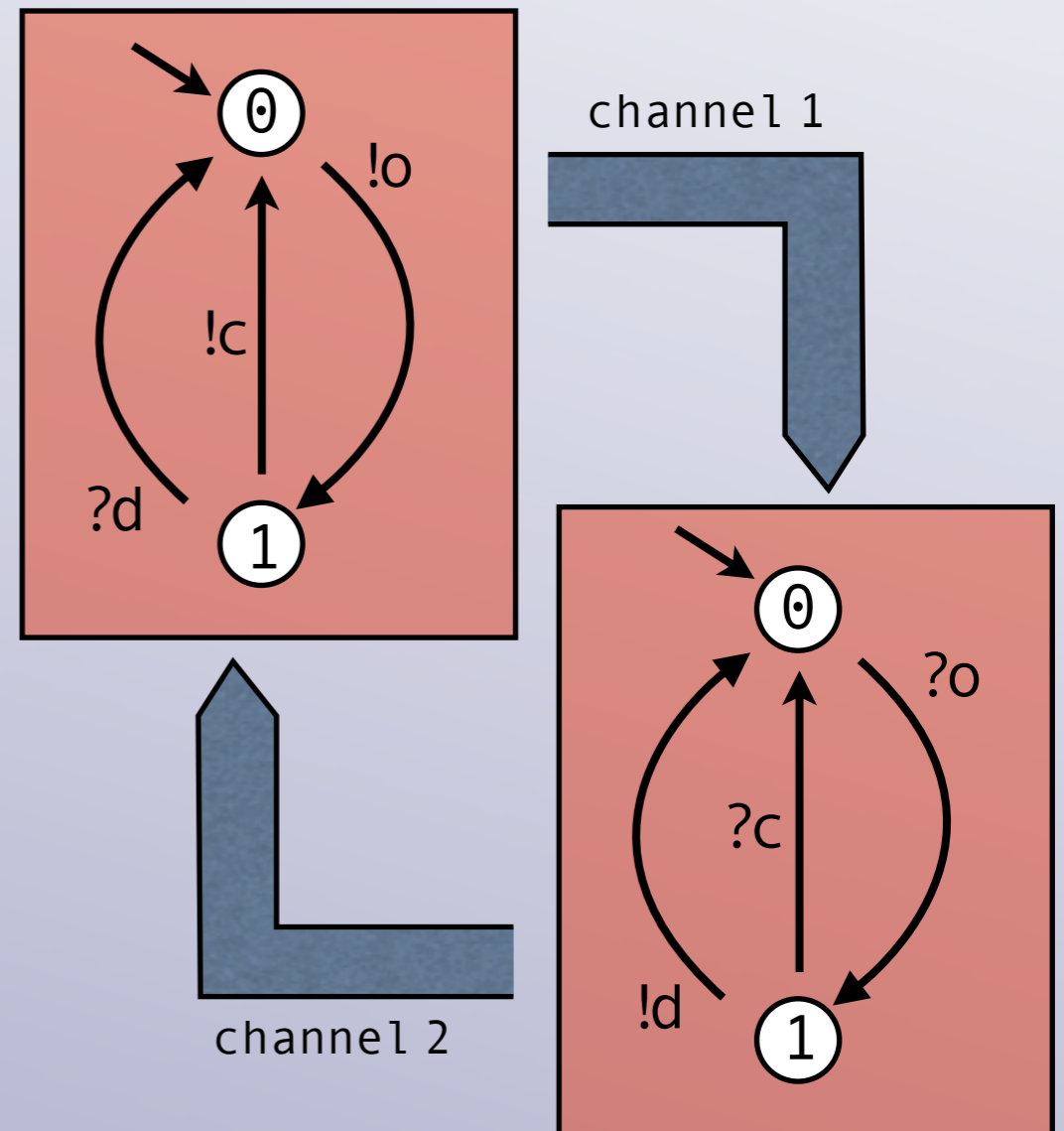


# abstract safety:

abstract labelled transition system regarding:



**Bad<sup>#</sup>**  
 configurations that intersect with **Bad**

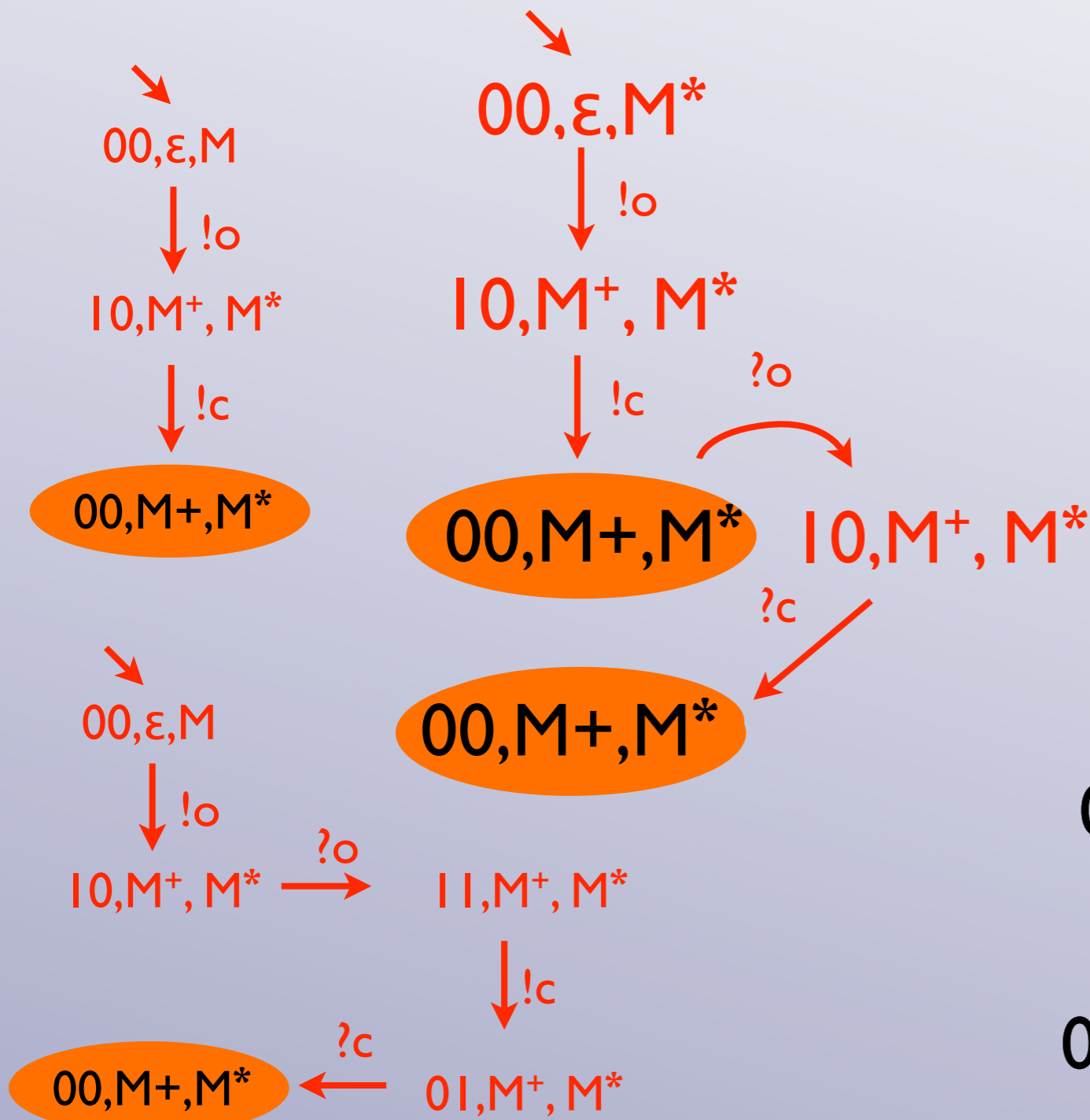


## Proposition:

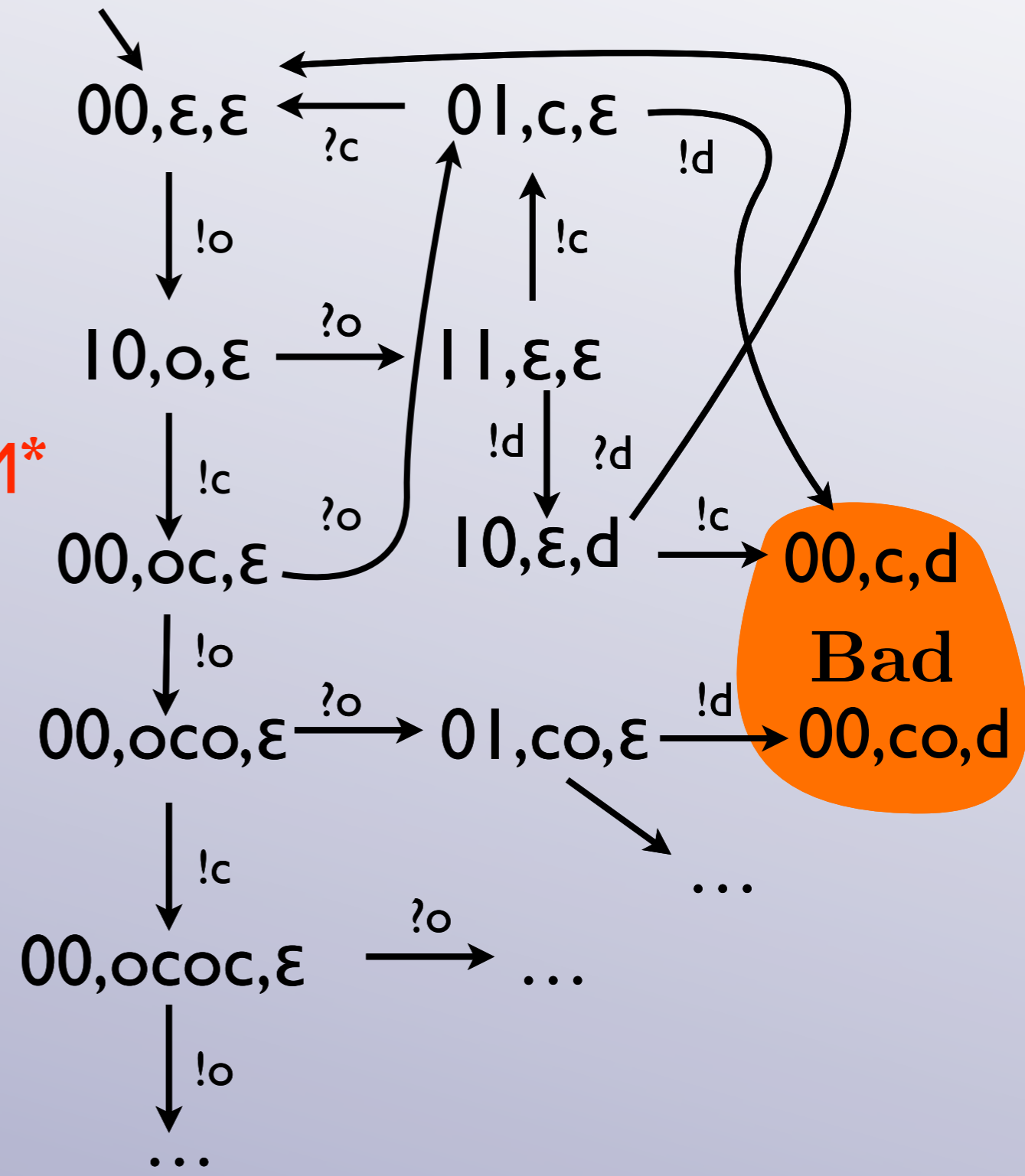
Given a partition abstraction of the fifo system  $\mathcal{S}$ , the labeled transition system is **Bad-safe**, if its partition abstraction is safe with respect to **Bad**<sup>#</sup>

proof-idea: conservative over-approximation...

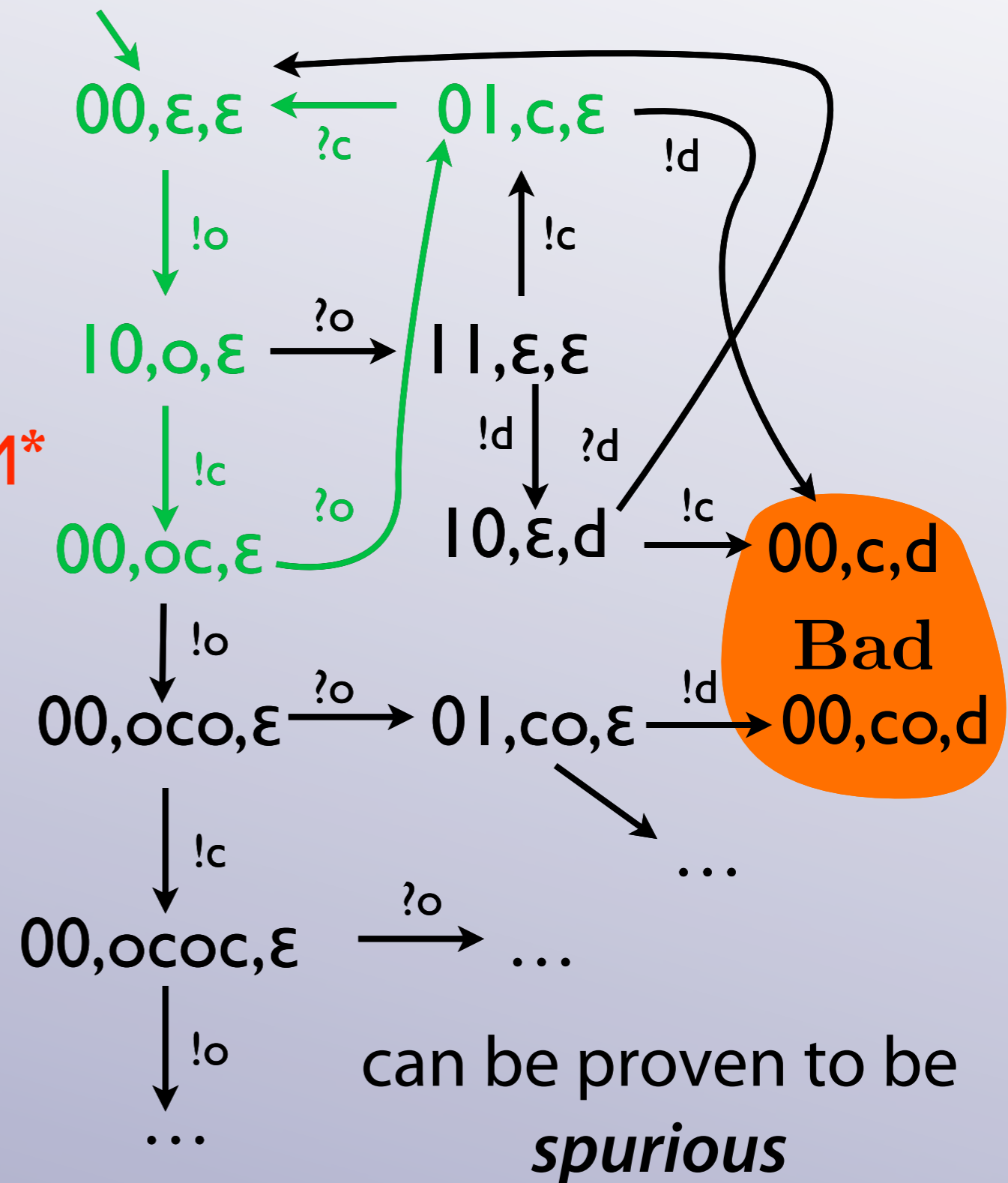
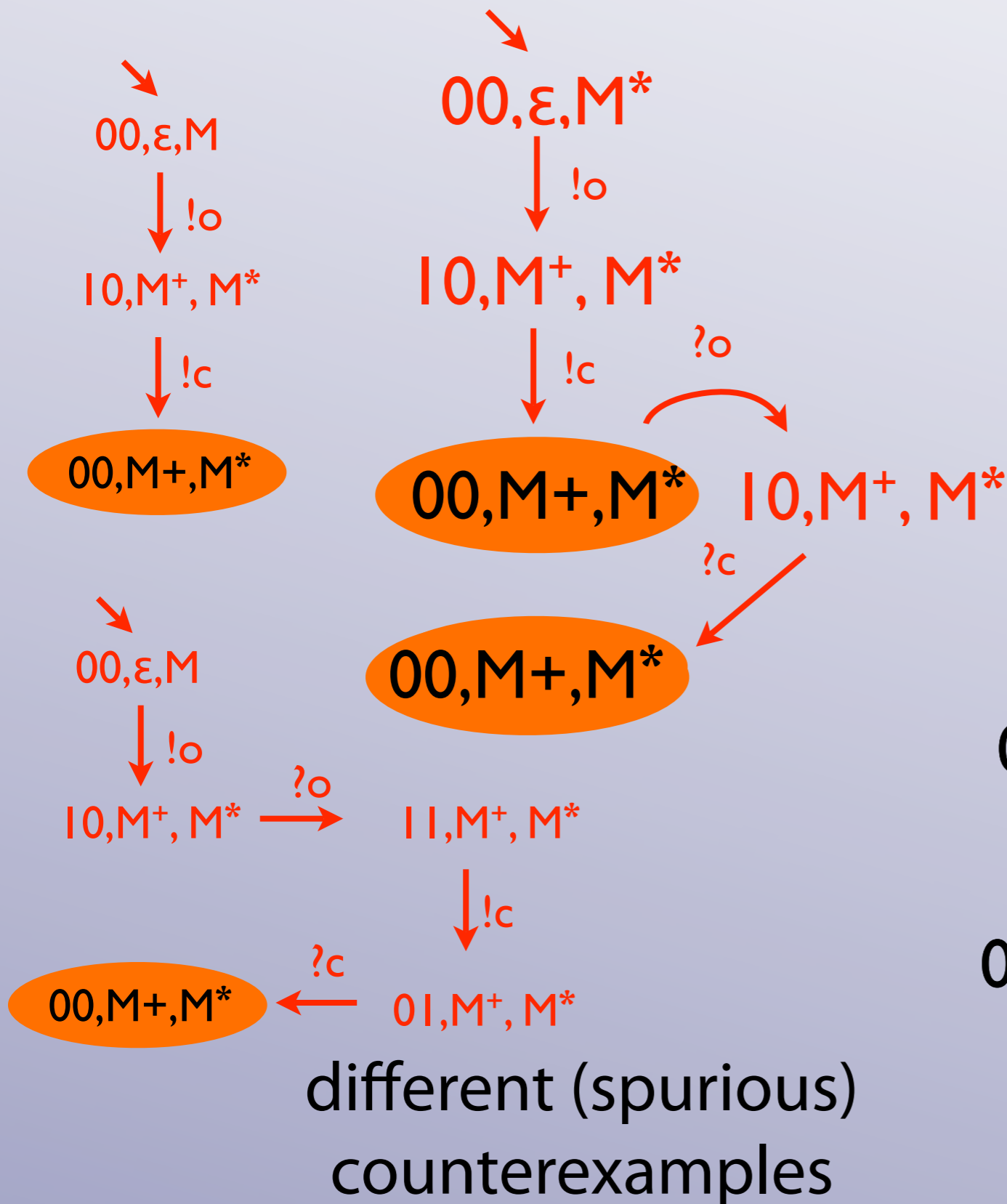
# spurious counterexamples:



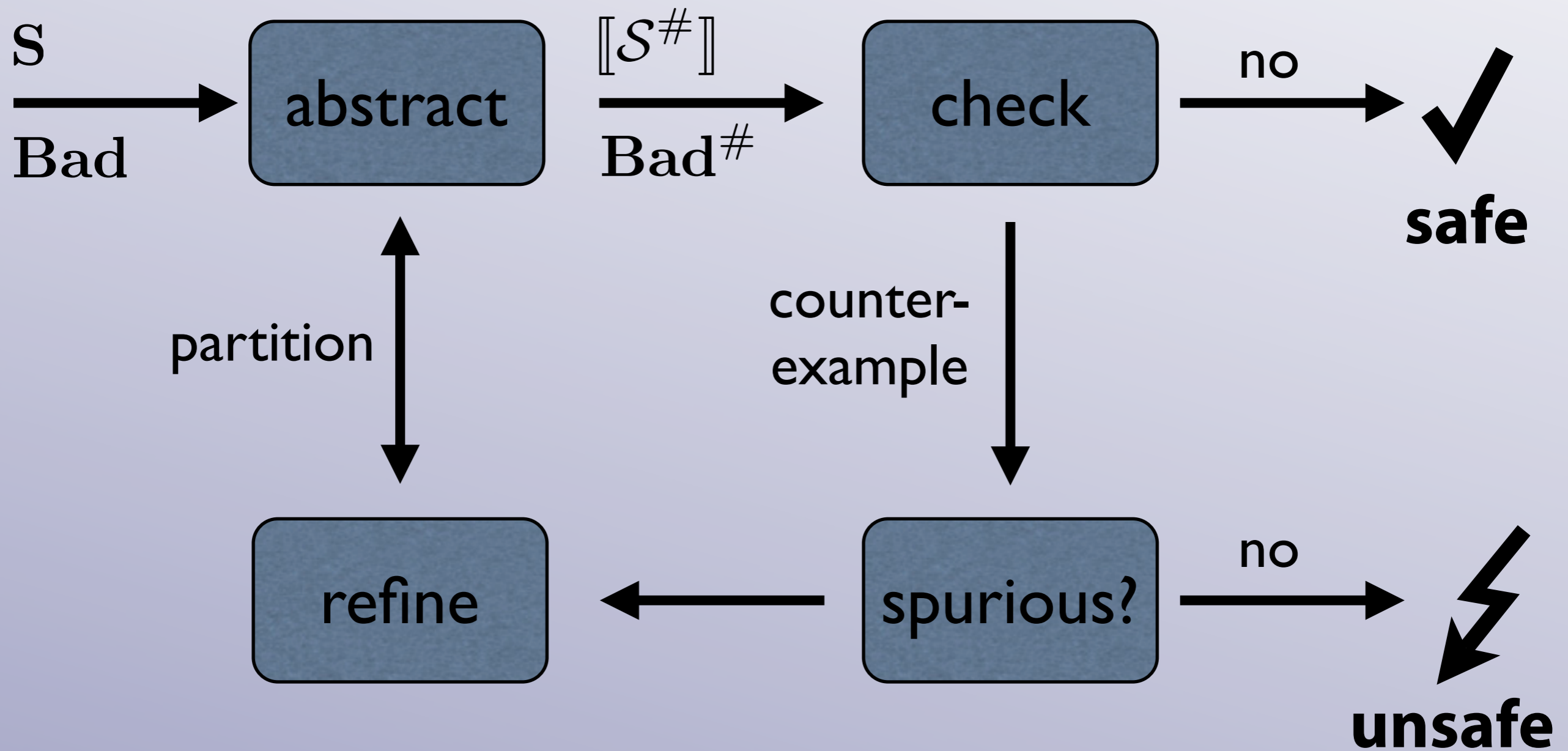
different (spurious) counterexamples



# spurious counterexamples:



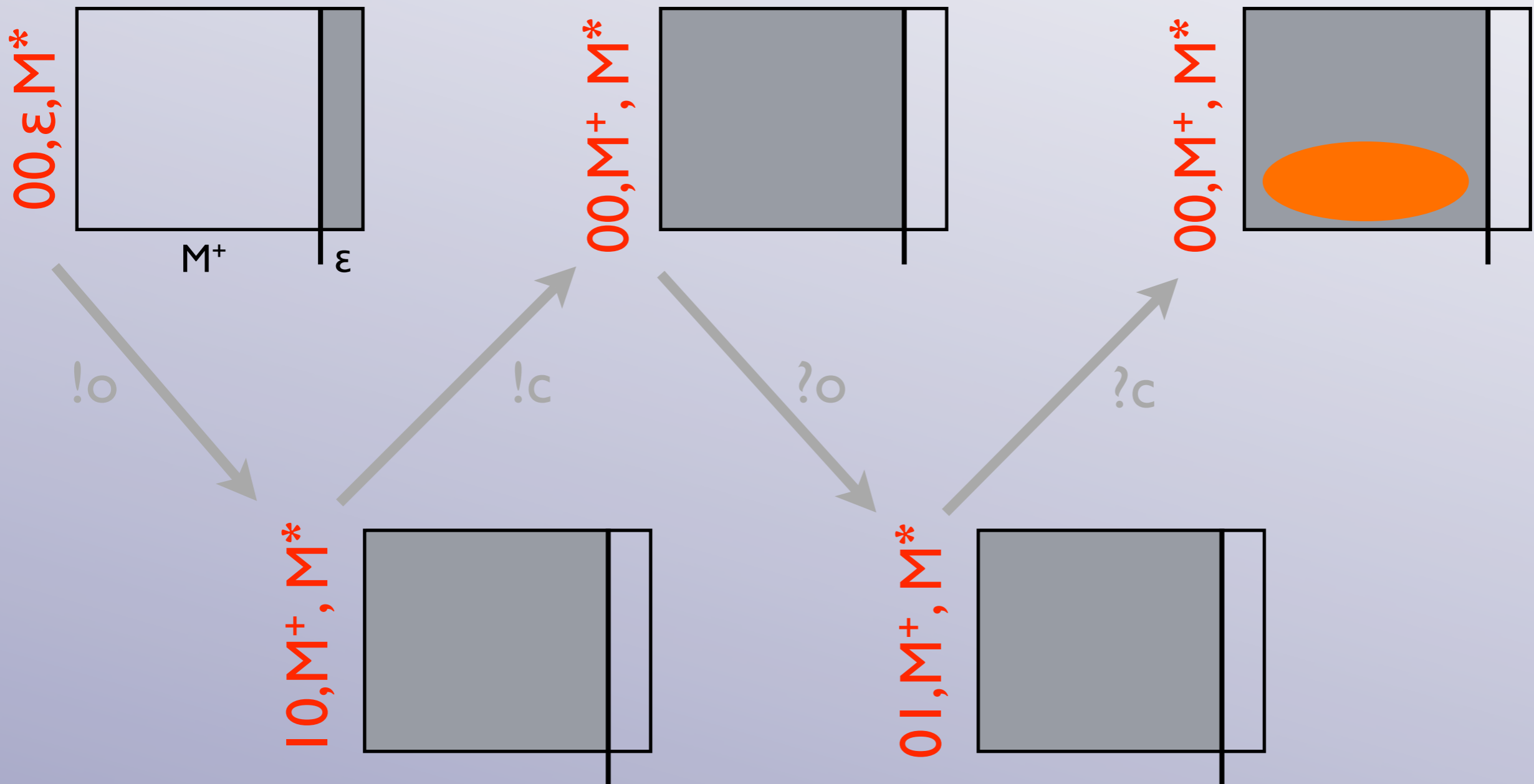
# CEGAR-loop:



# Path Invariants:

given a spurious counterexample...

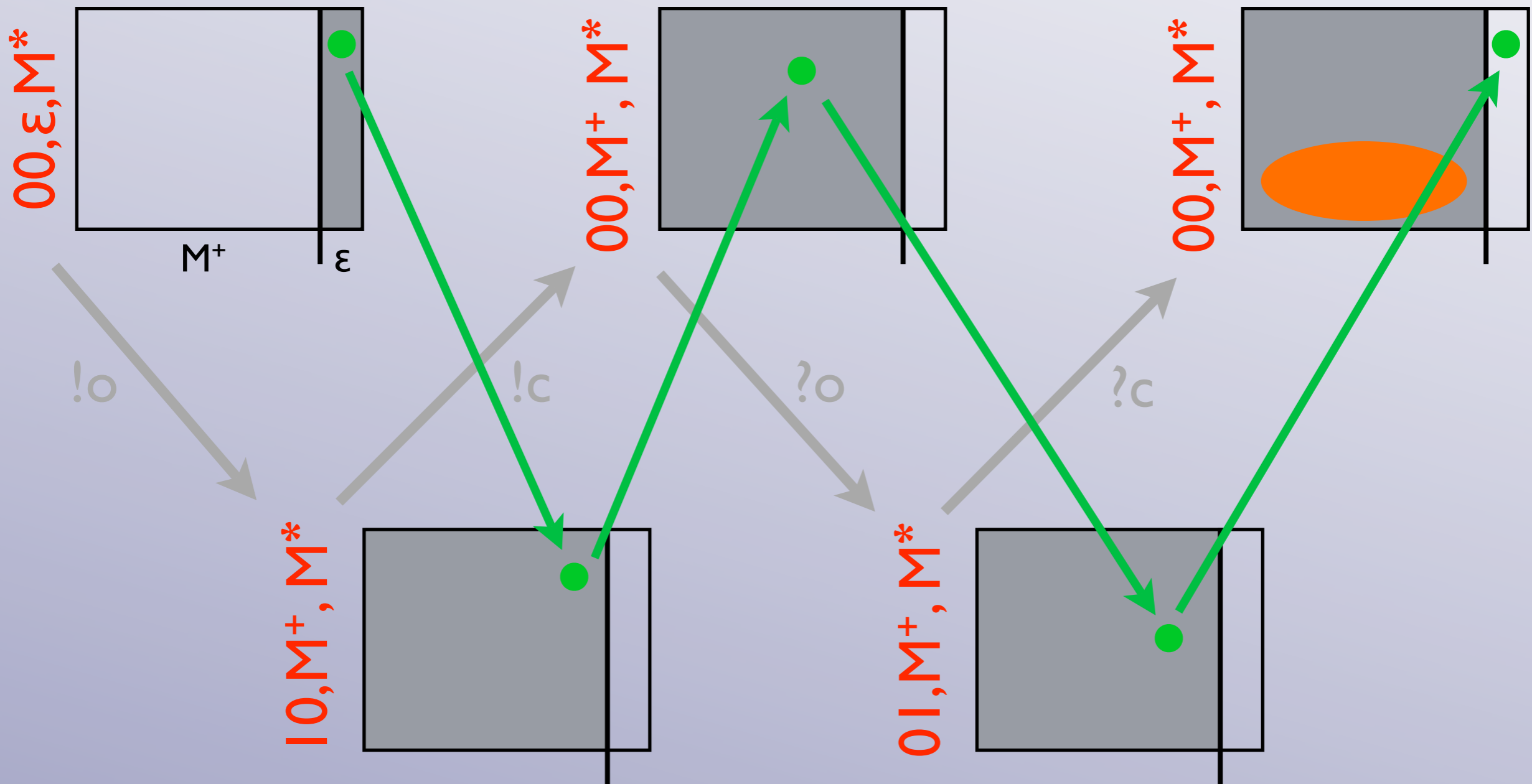
**Bad<sup>#</sup> unsafe**



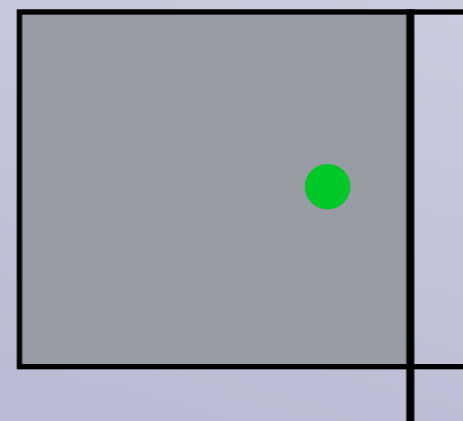
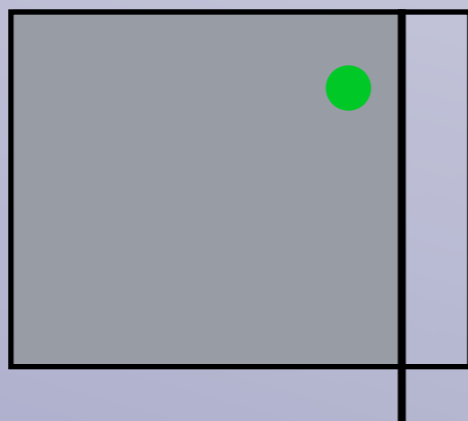
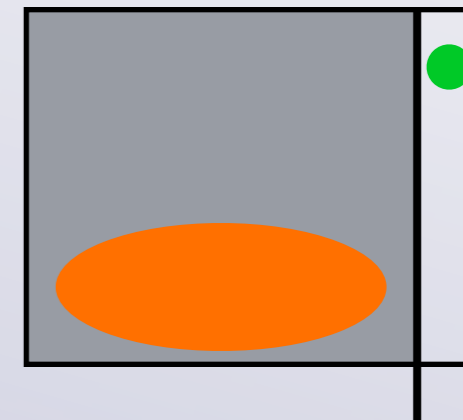
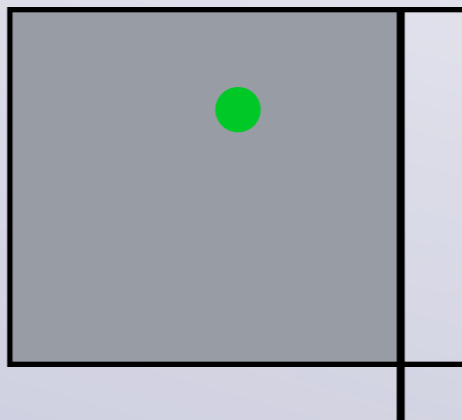
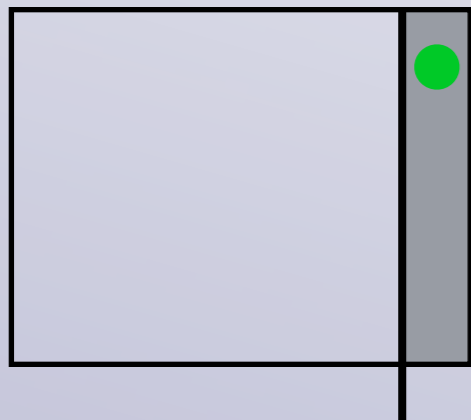
# Path Invariants:

given a spurious counterexample...

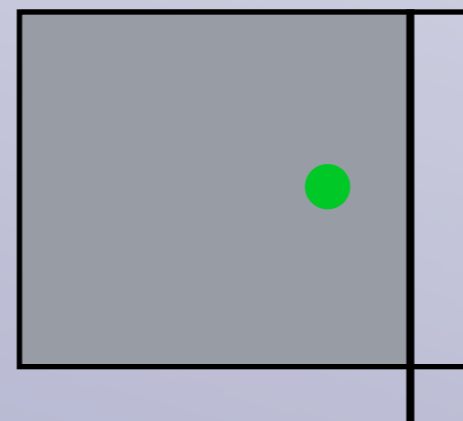
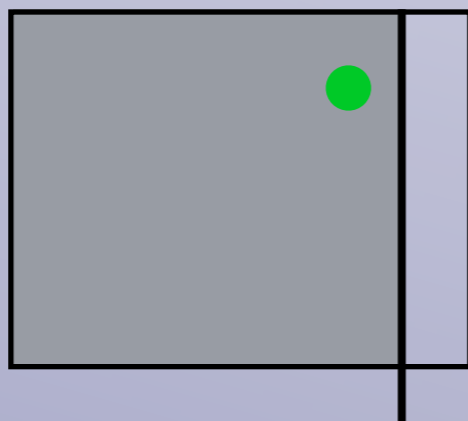
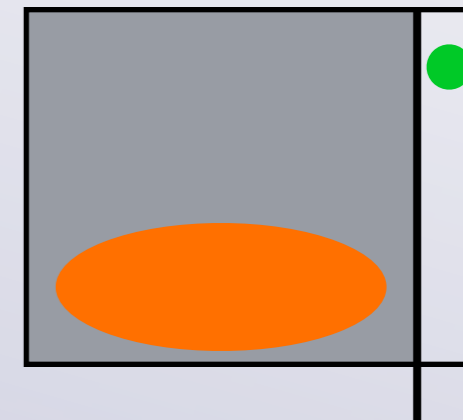
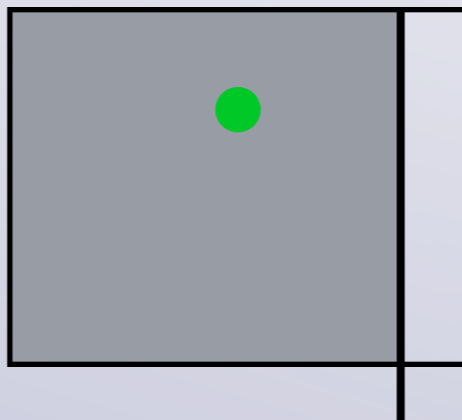
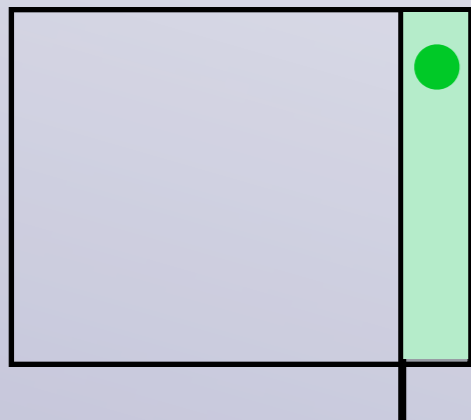
**Bad<sup>#</sup> unsafe**



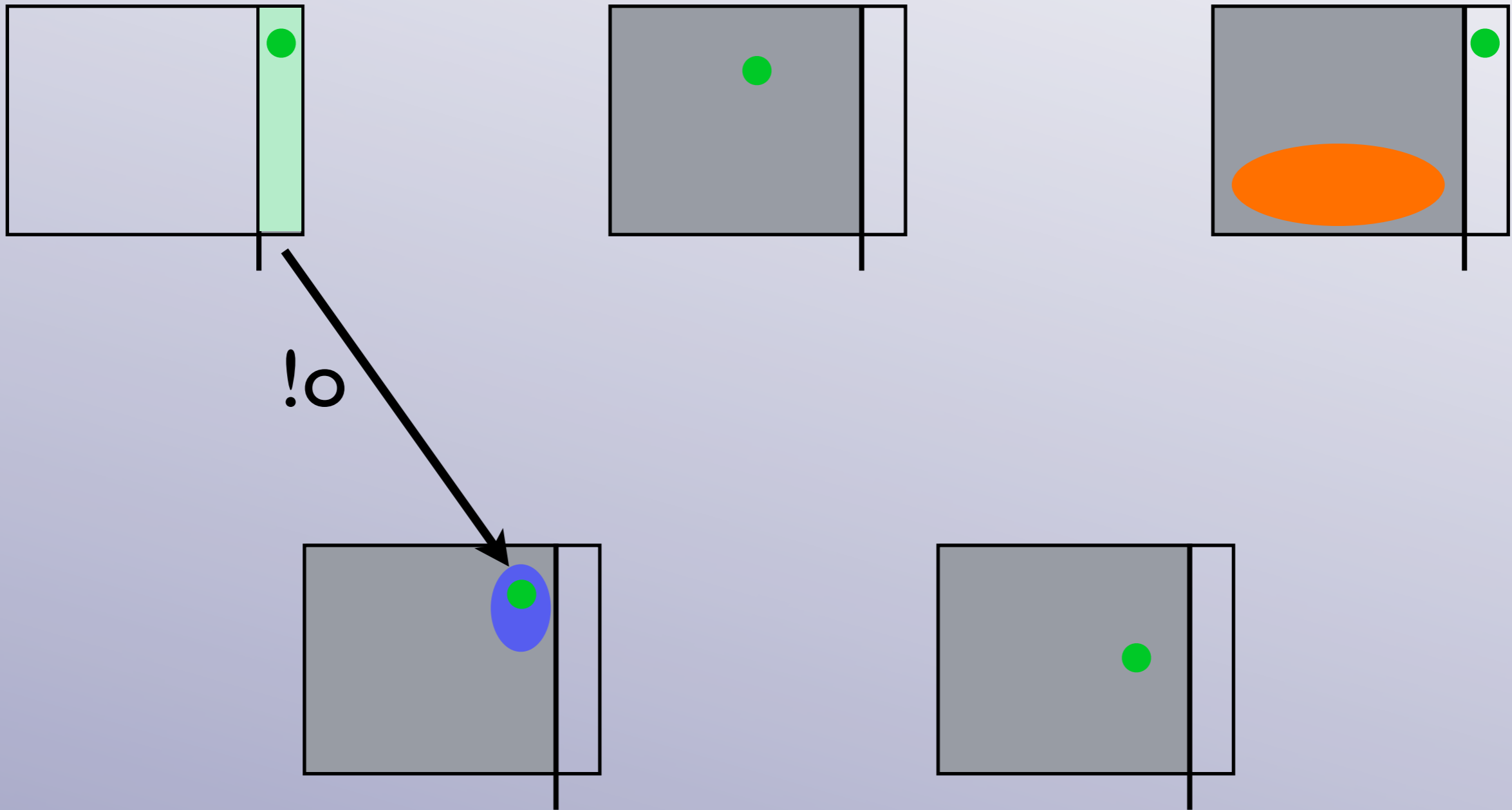
one calculates a path invariant as follows:



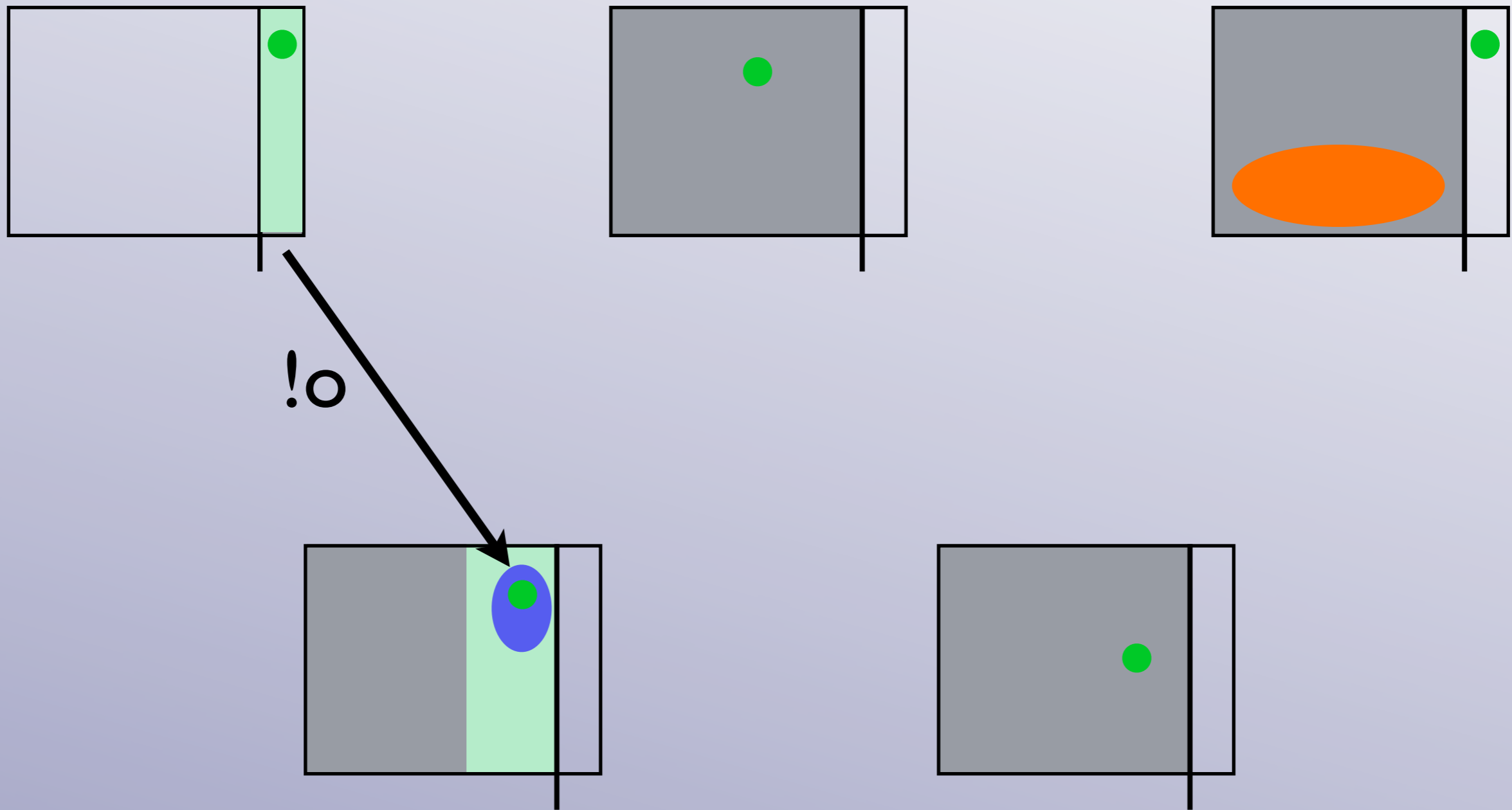
one calculates a path invariant as follows:



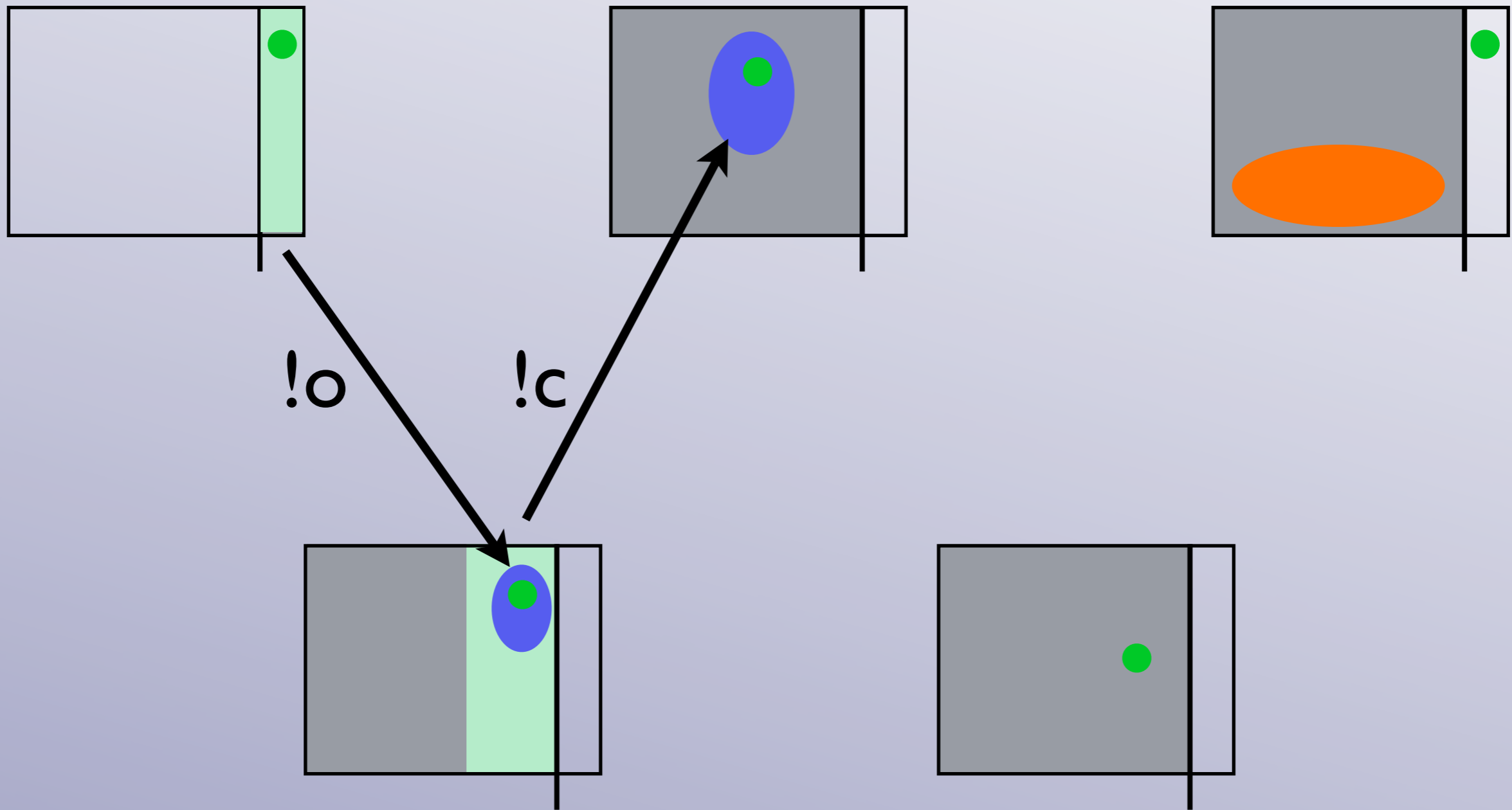
one calculates a path invariant as follows:



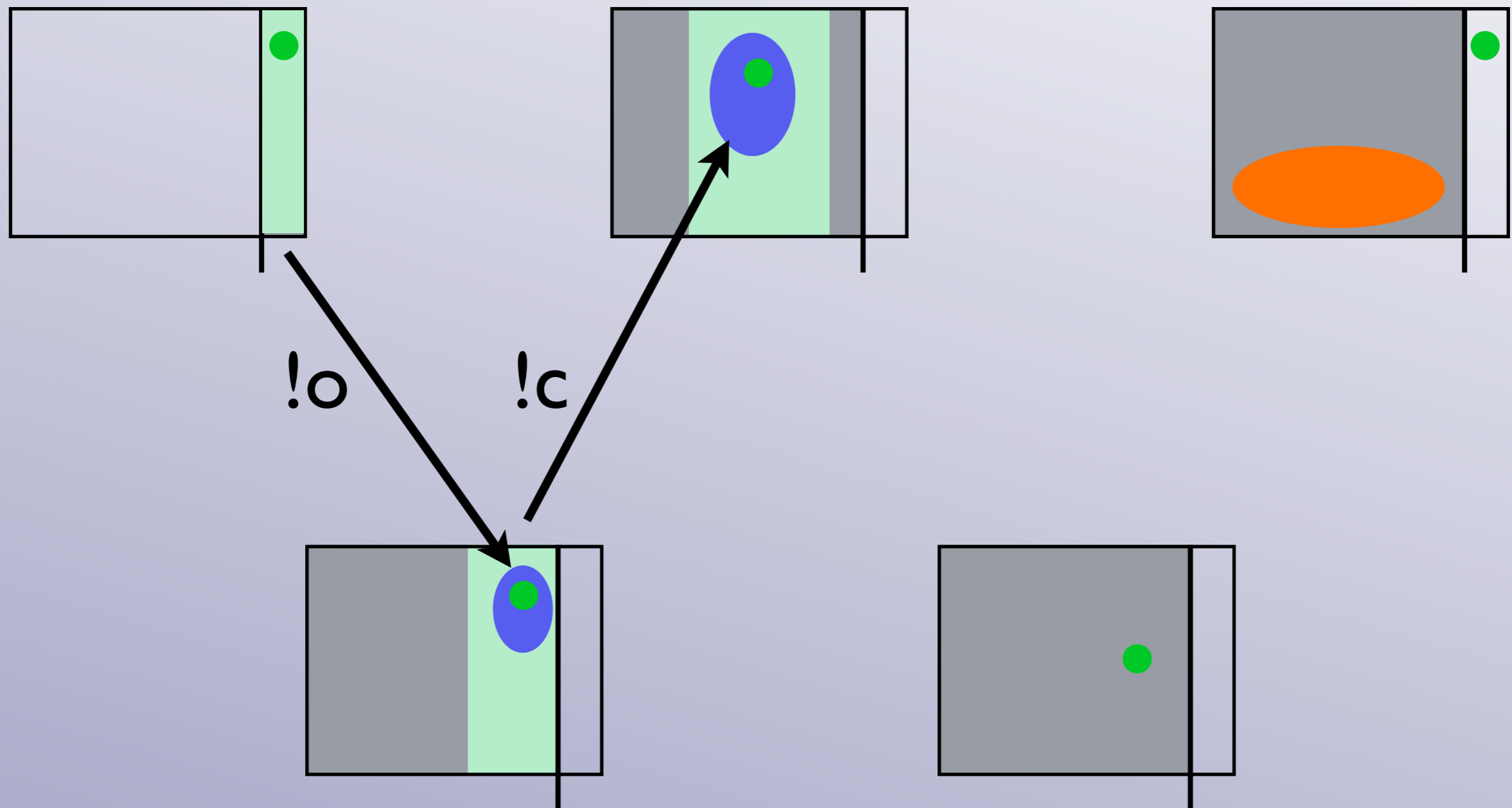
one calculates a path invariant as follows:



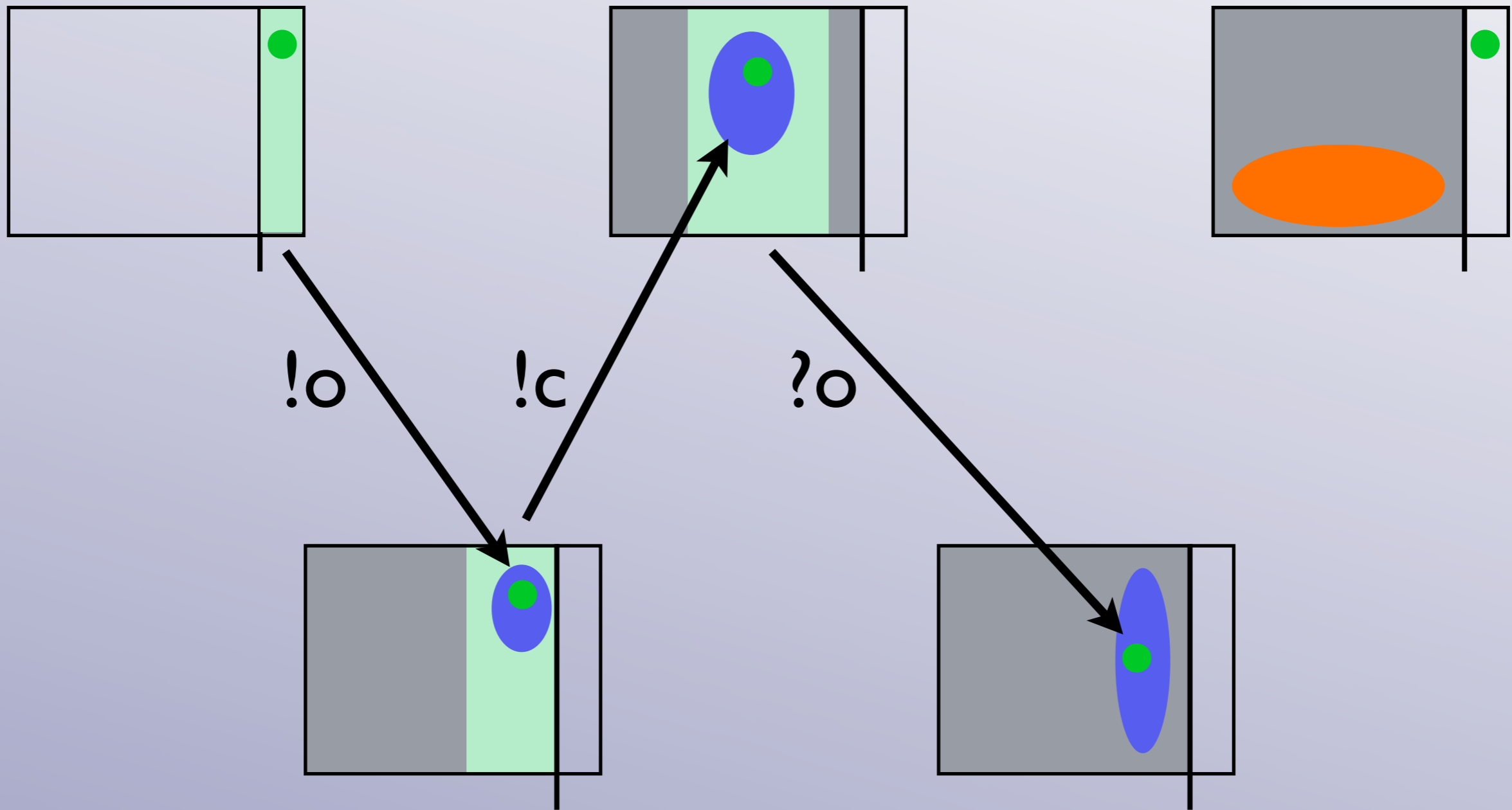
one calculates a path invariant as follows:



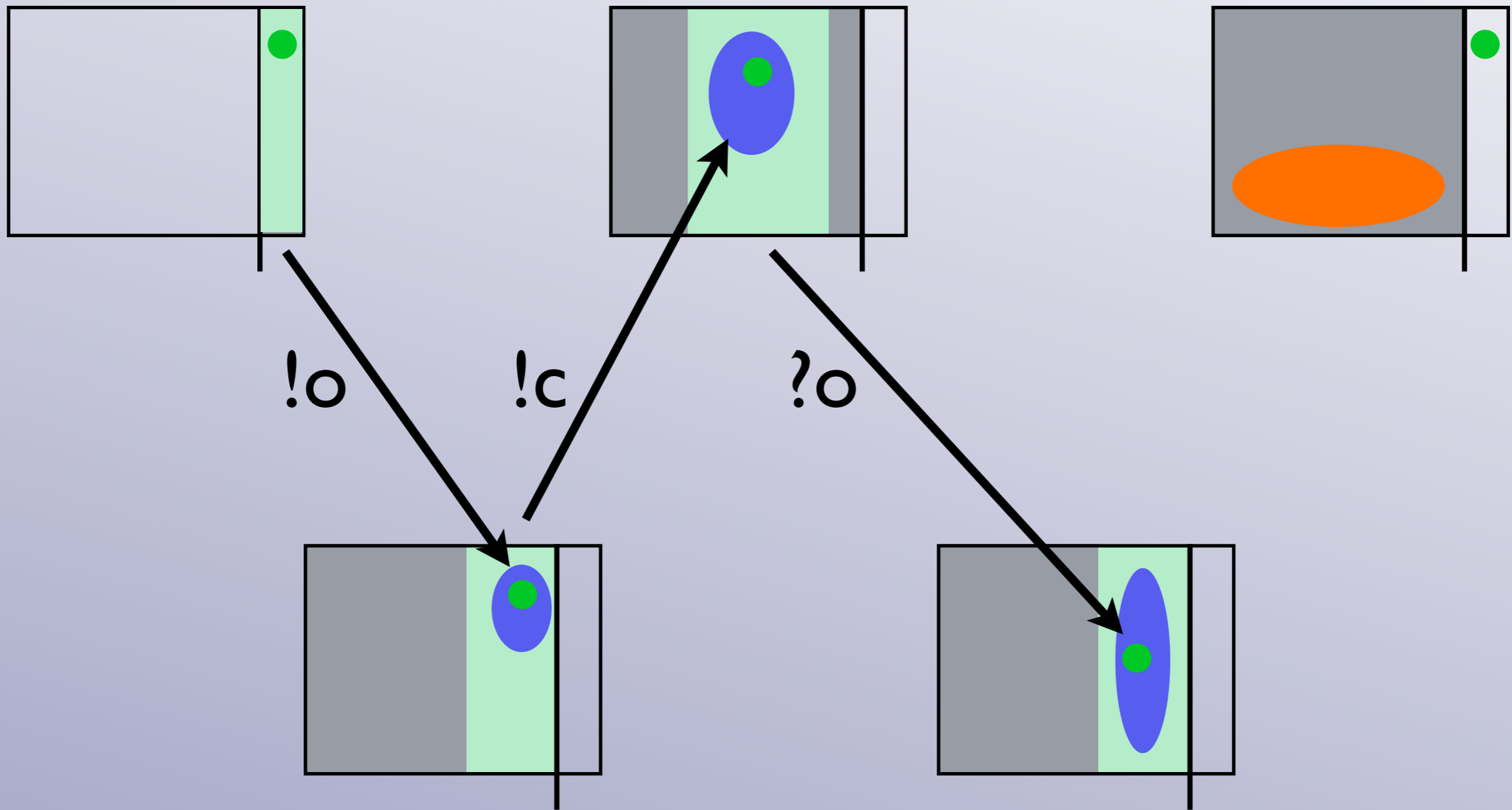
one calculates a path invariant as follows:



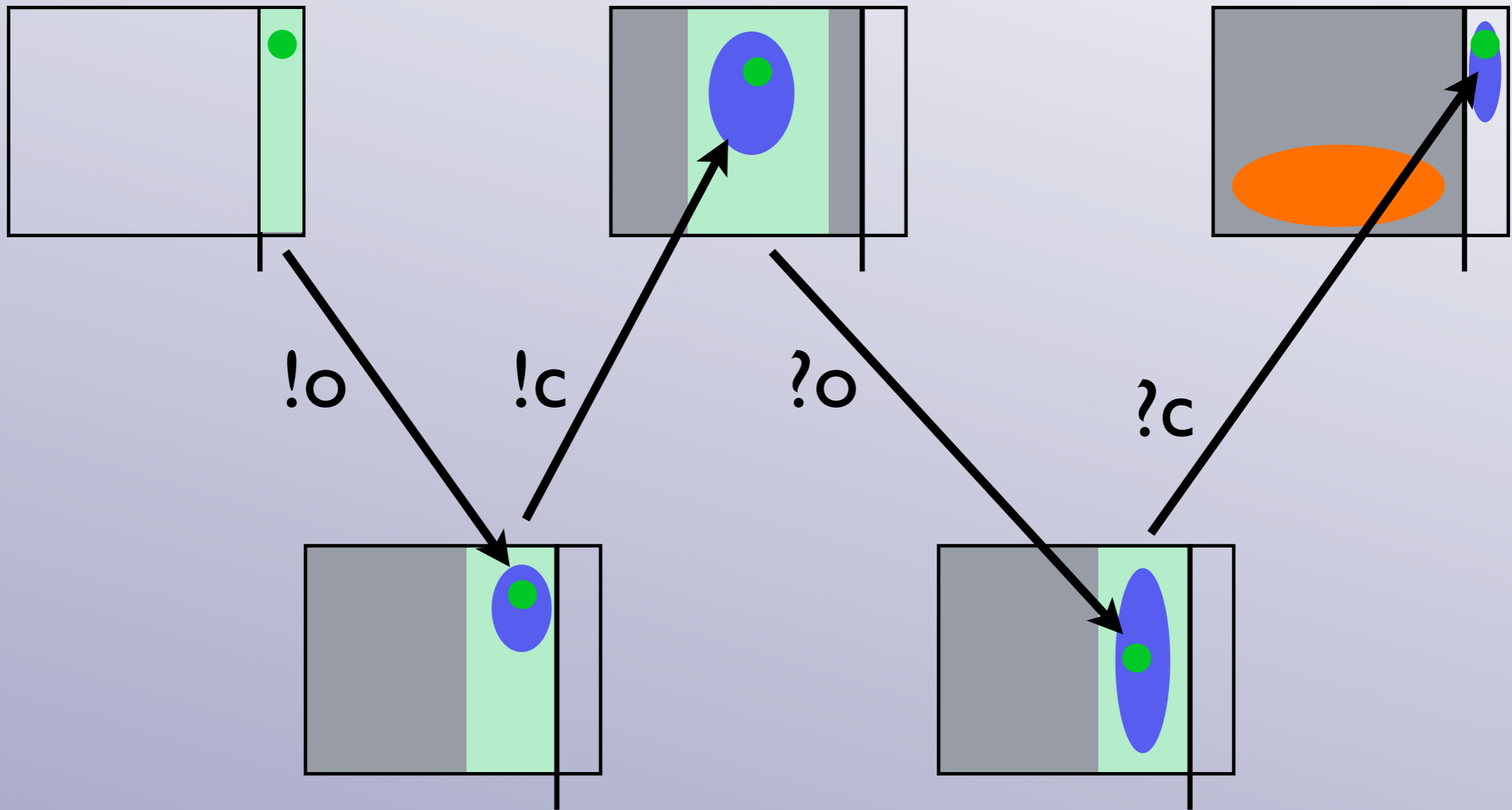
one calculates a path invariant as follows:



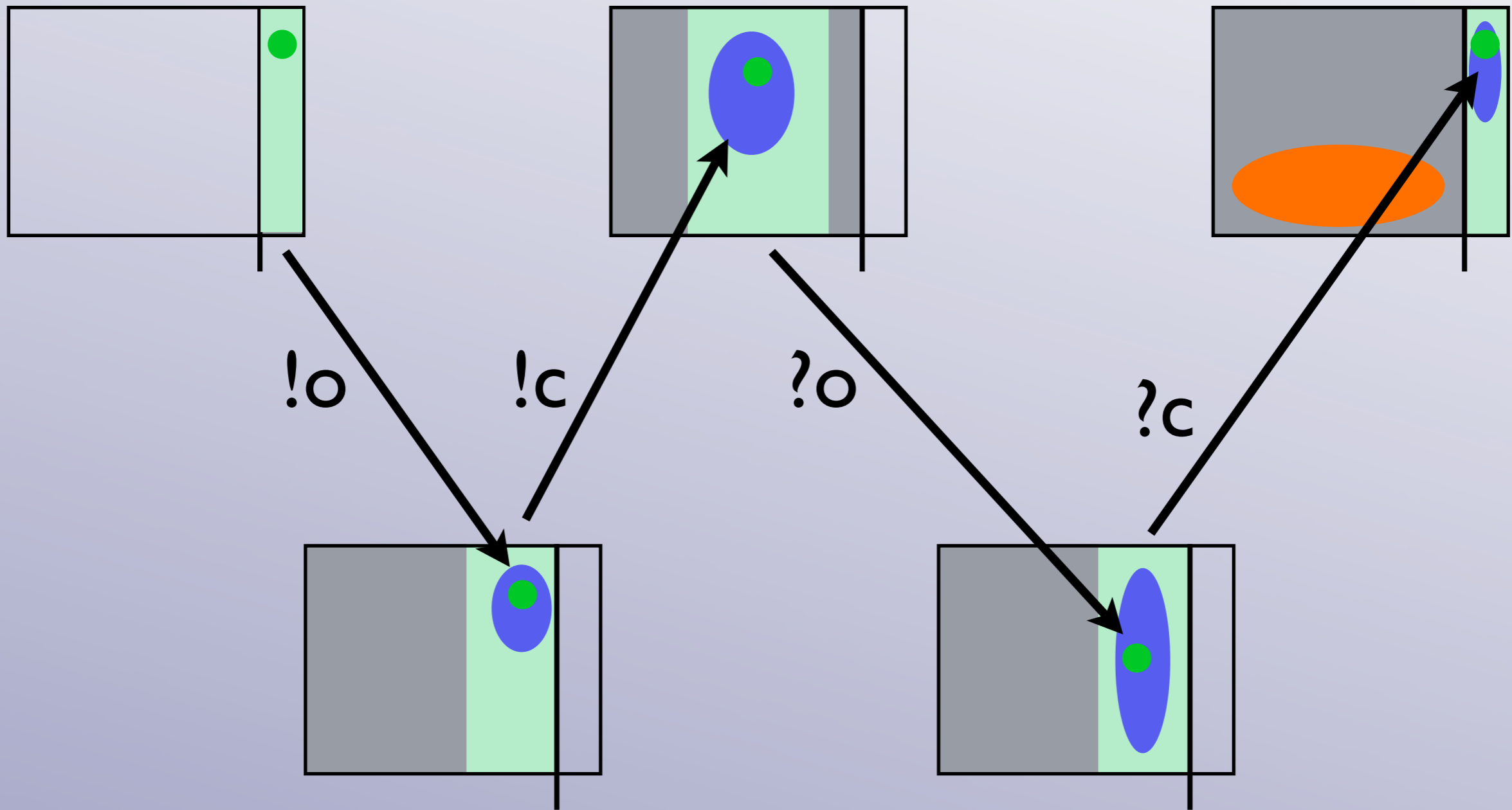
one calculates a path invariant as follows:



one calculates a path invariant as follows:

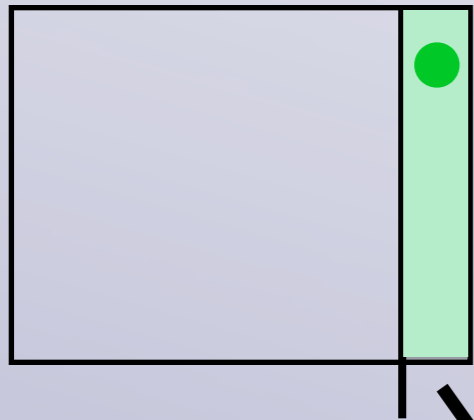


one calculates a path invariant as follows:

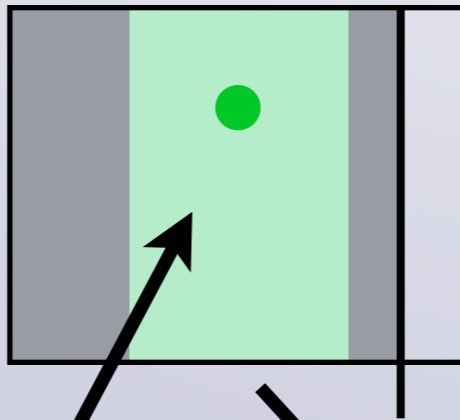


# Path Invariants:

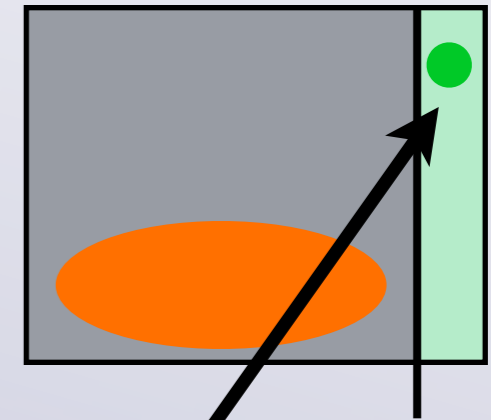
(i) first includes initial configs



(ii) are correct transitions



(iii) last is disjoint with **Bad**

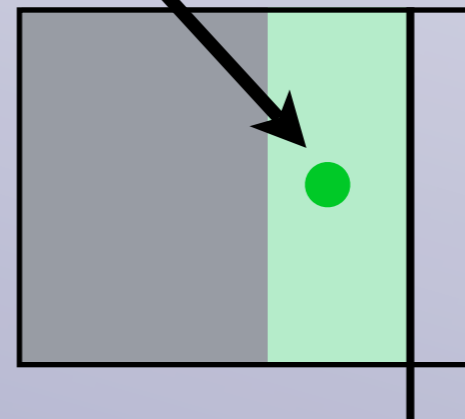
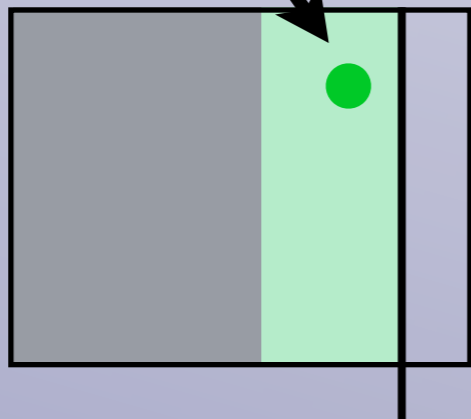


!o

!c

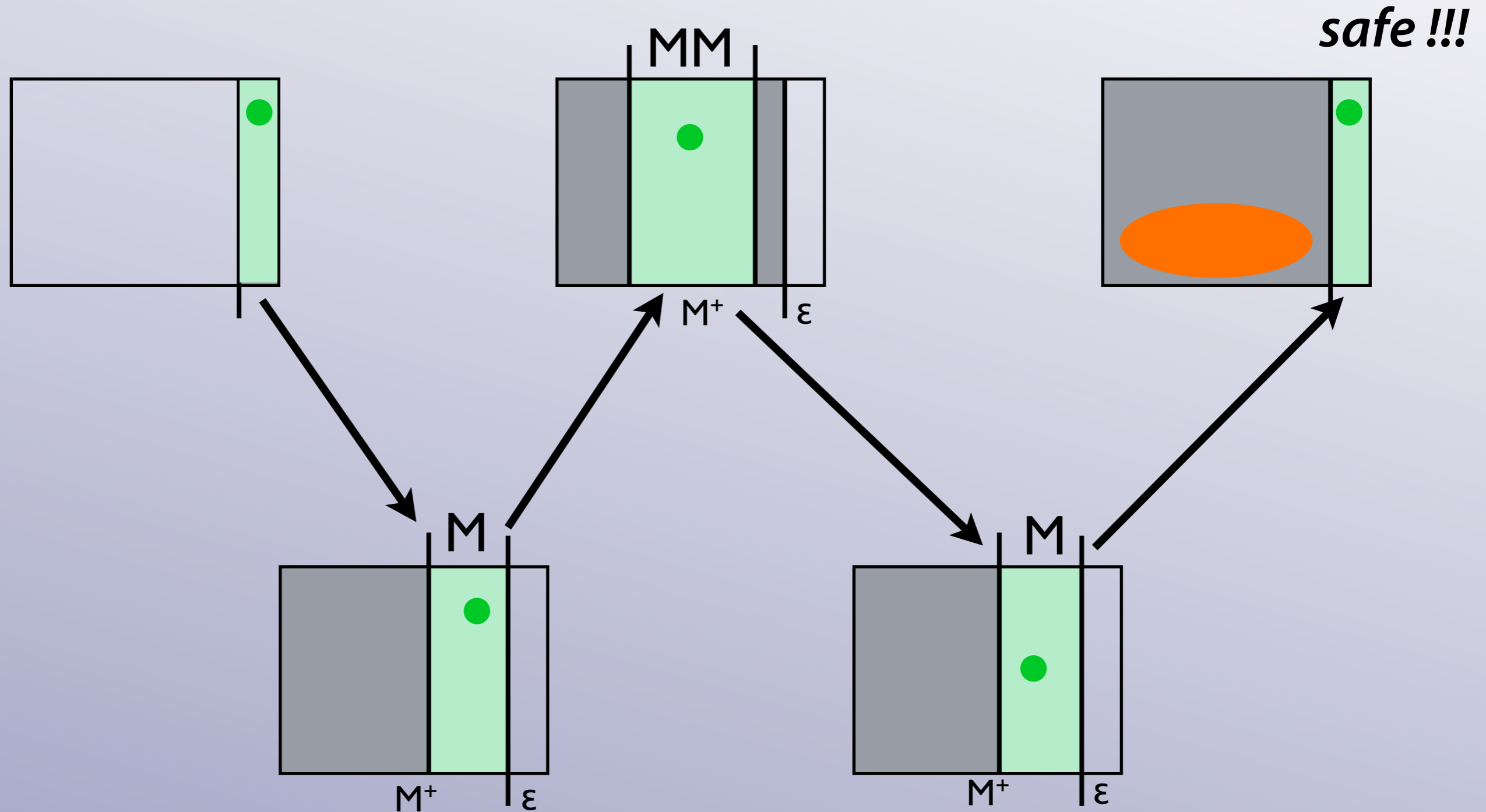
?o

?c



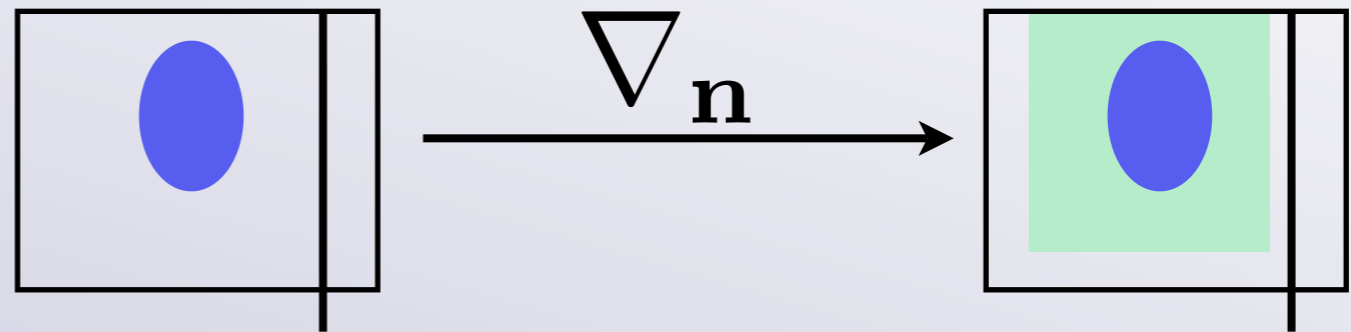
# Refine the Partition:

to rule out the spurious counterexample !

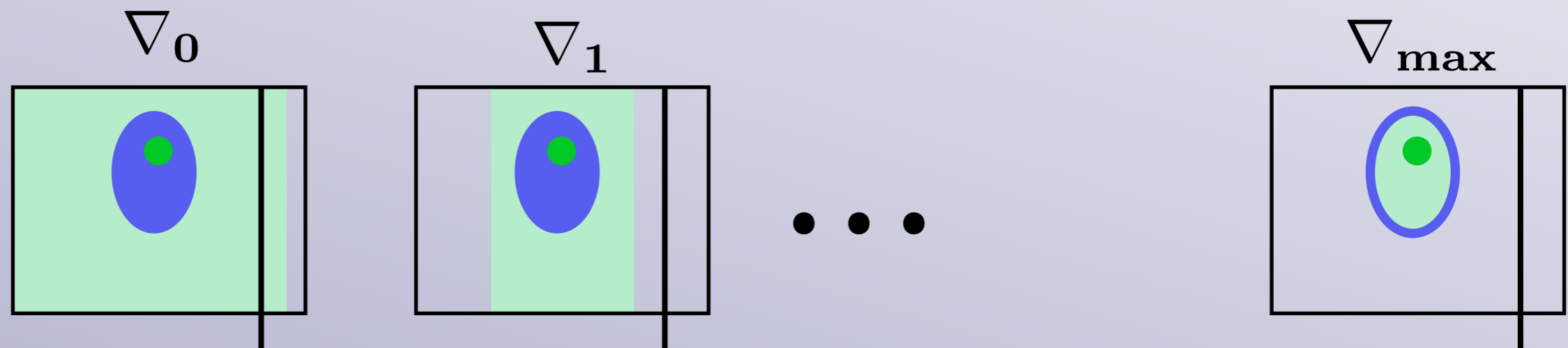


# Extrapolation:

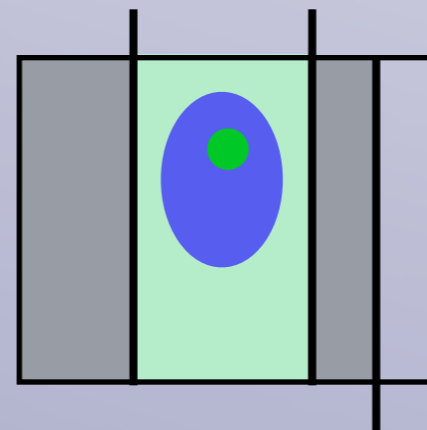
- overapproximate



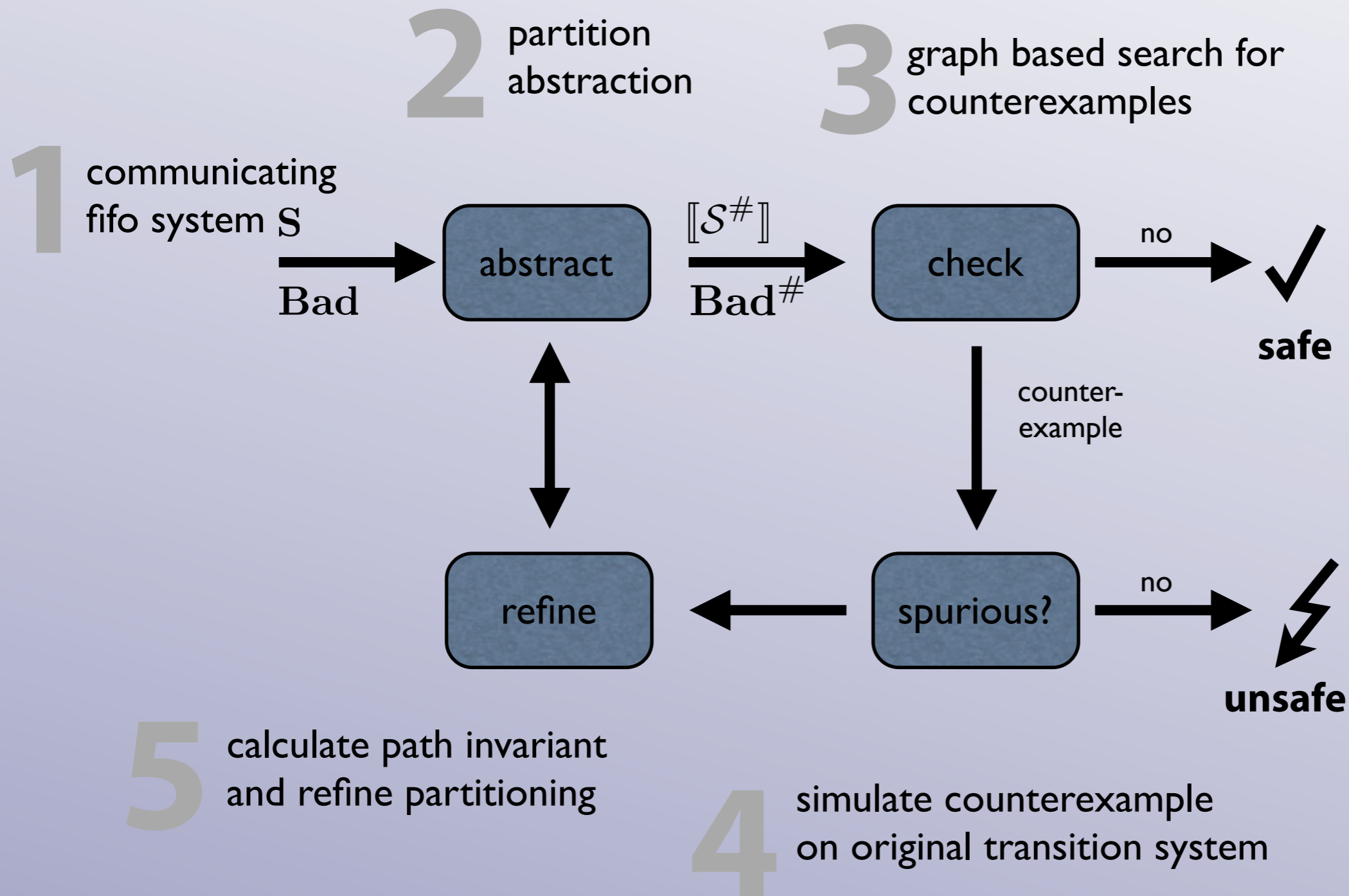
- fine-tune precision via parametrized operator



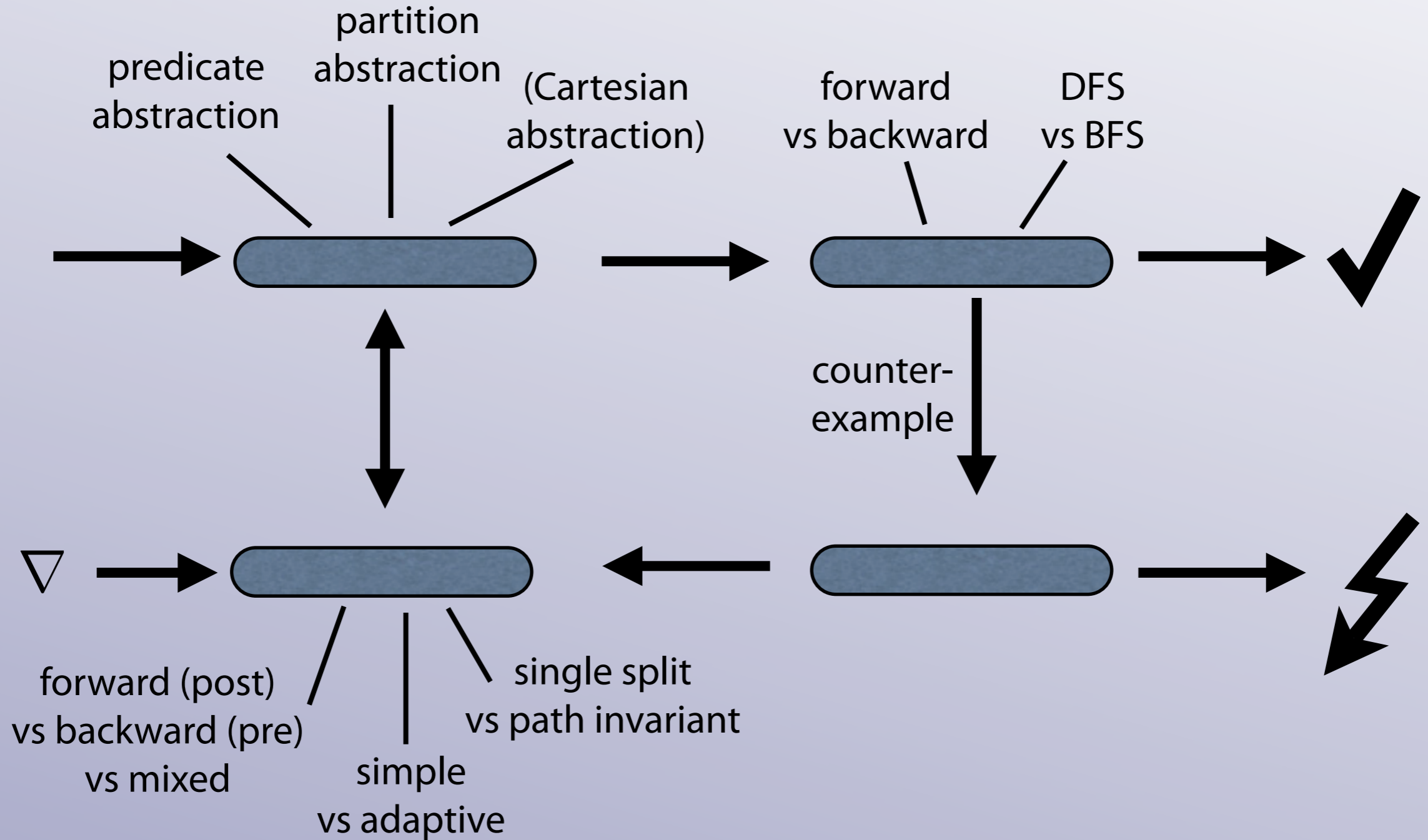
- allow to refine partition



# CEGAR-loop:



# Prototypical Implementation:



# Empirical Results:

Examples	Single-split			Path-invariant		
	Time(s)	Space(MiB)	Method	Time(s)	Space(MiB)	Method
(1) ABP	2.98	2	BFS	4.64	2.23	BFS
(2) Conn./disc.	0.02	0.36	Mix	0.02	0.36	BFS
(3) Non-regular	0.04	0.59	Mix	0.17	0.59	Mix
(4) Simplified TCP	11.36	6.22	DFS	3.44	2.47	DFS
(5) nested loops	7.13	2.47	DFS	1.84	1.06	DFS
(6) Ring	>1000	36.7	Mix	7.83	4.32	BFS

# Future Work:

- extend approach to the communication of data based on the model of communicating lattice automata  
(model sliding window protocols etc.)
- extensive empirical testing and in-depth comparison to other tools (TReX, Lash,...)
- characterize for which classes of protocols the CEGAR algorithm terminates
- ... (feel free to append 😊 )

