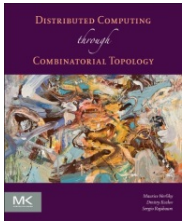
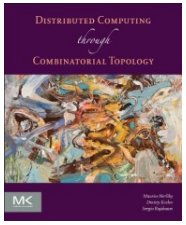
Distributed Computing through Combinatorial Topology

MITRO207, P4, 2019

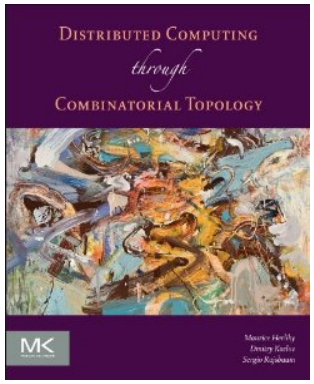


Administrivia

- Language: English? Français sur demande
- Lectures: Wednesday, 8:30-11:45
- Web page: http://perso.telecom-paristech.fr/~kuznetso/MITRO207-2019/
- Homeworks
  - Corrected, not graded
  - TD: 29.05.2019
- Office hours:
  - C213-2, appointments by email to petr.kuznetsov@telecom-paristech.fr
- Credit = written exam: June 25, 2019 (3 hours)
- Bonus for homeworks, participation, discussion of exercises, bugs found

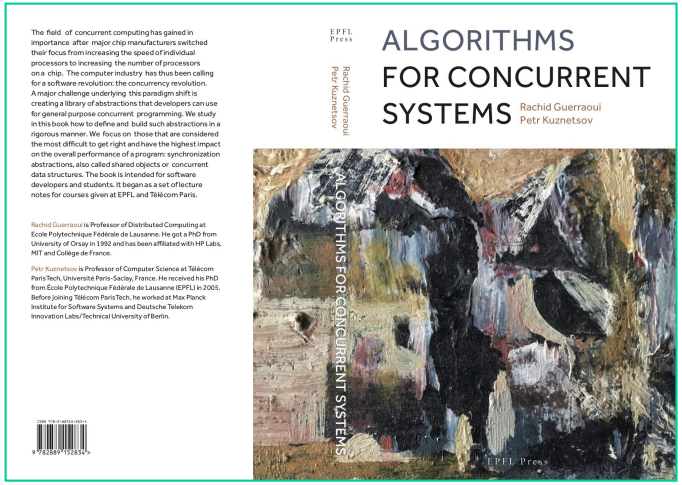


Literature



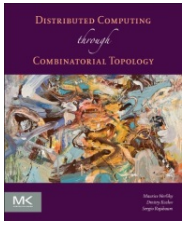
Distributed Computing
Through Combinatorial Topology
Maurice Herlihy, Dmitry Kozlov, Sergio Rajsbaum
Morgan Kaufman, 2013, available online (TPT library

- Algorithms for ConcurrentSystems.
  R. Guerraoui, P. Kuznetsov, 2018

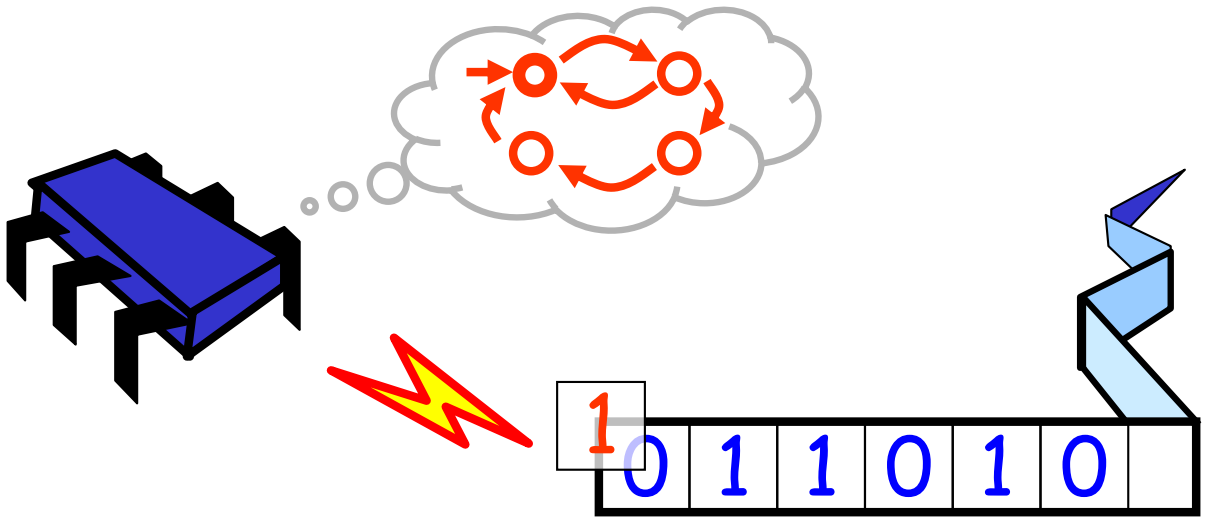


(Preliminary) road map

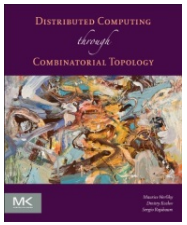
- The matter and the method of distributed computing
- Basics of combinatorial topology
- Colorless tasks
- Simulations and reductions
- Generic tasks and manifold computations
- Renaming and oriented manifolds



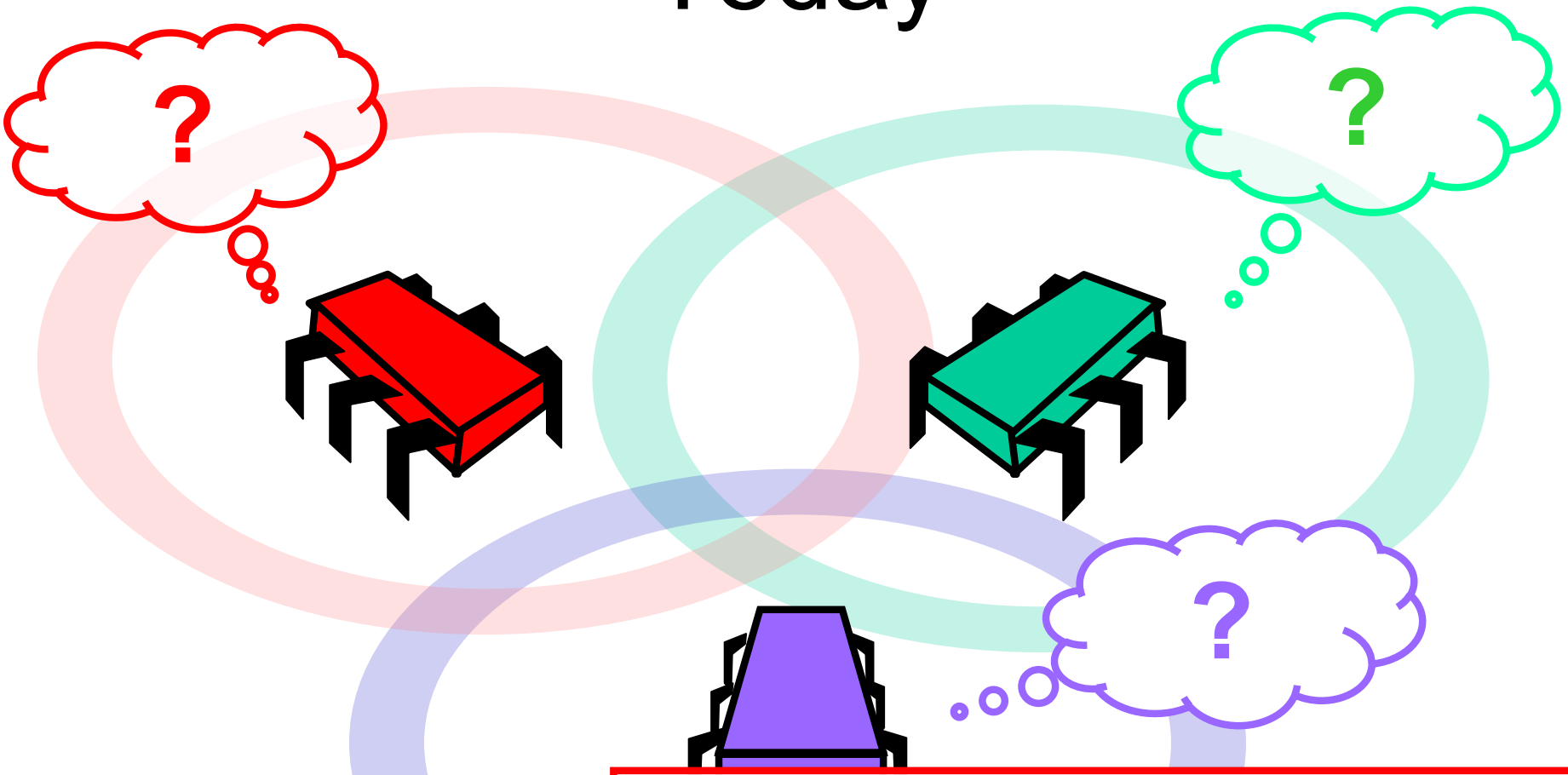
In the Beginning …



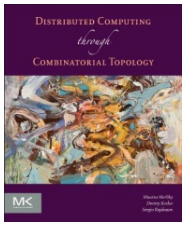
a computer was just a Turing machine …

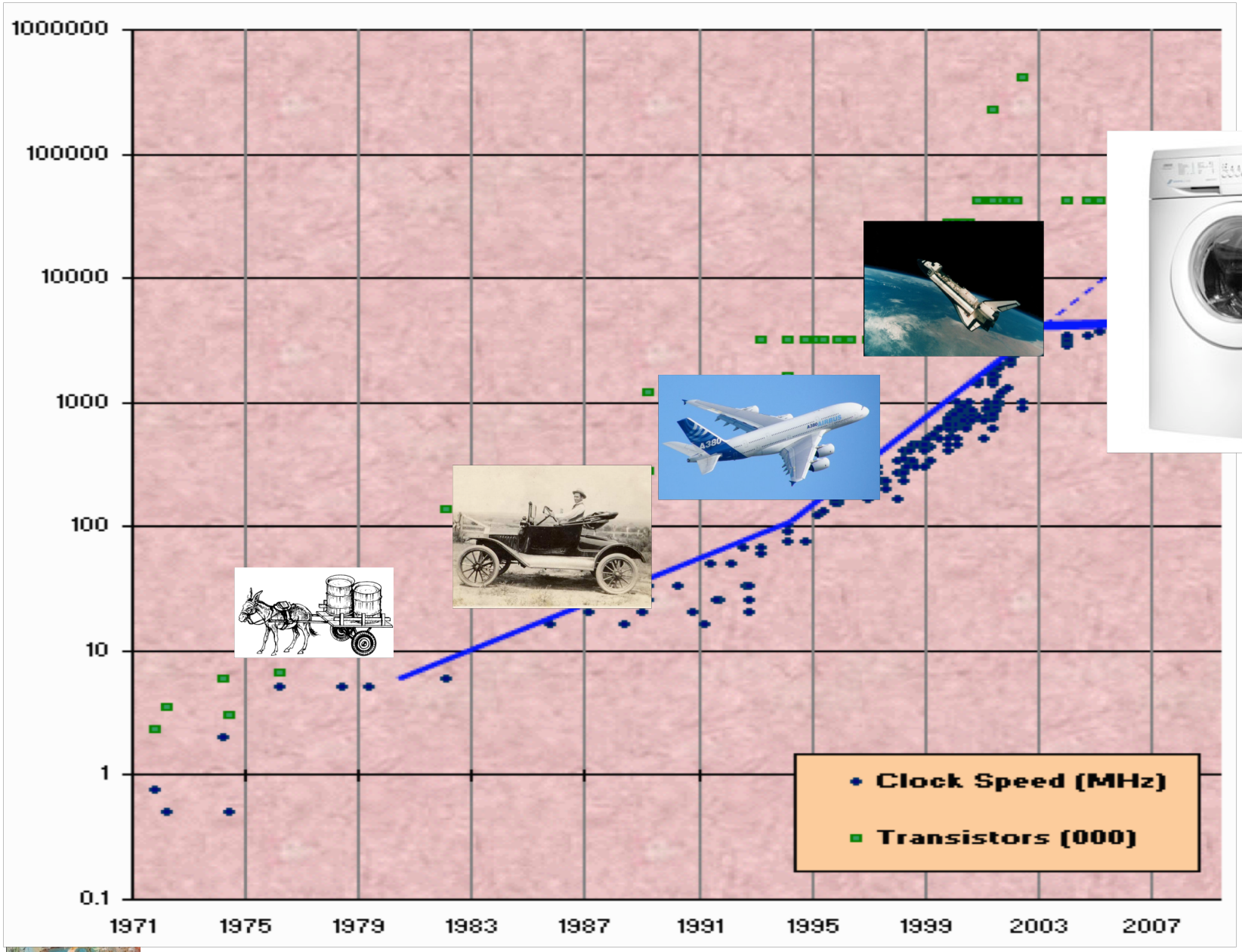


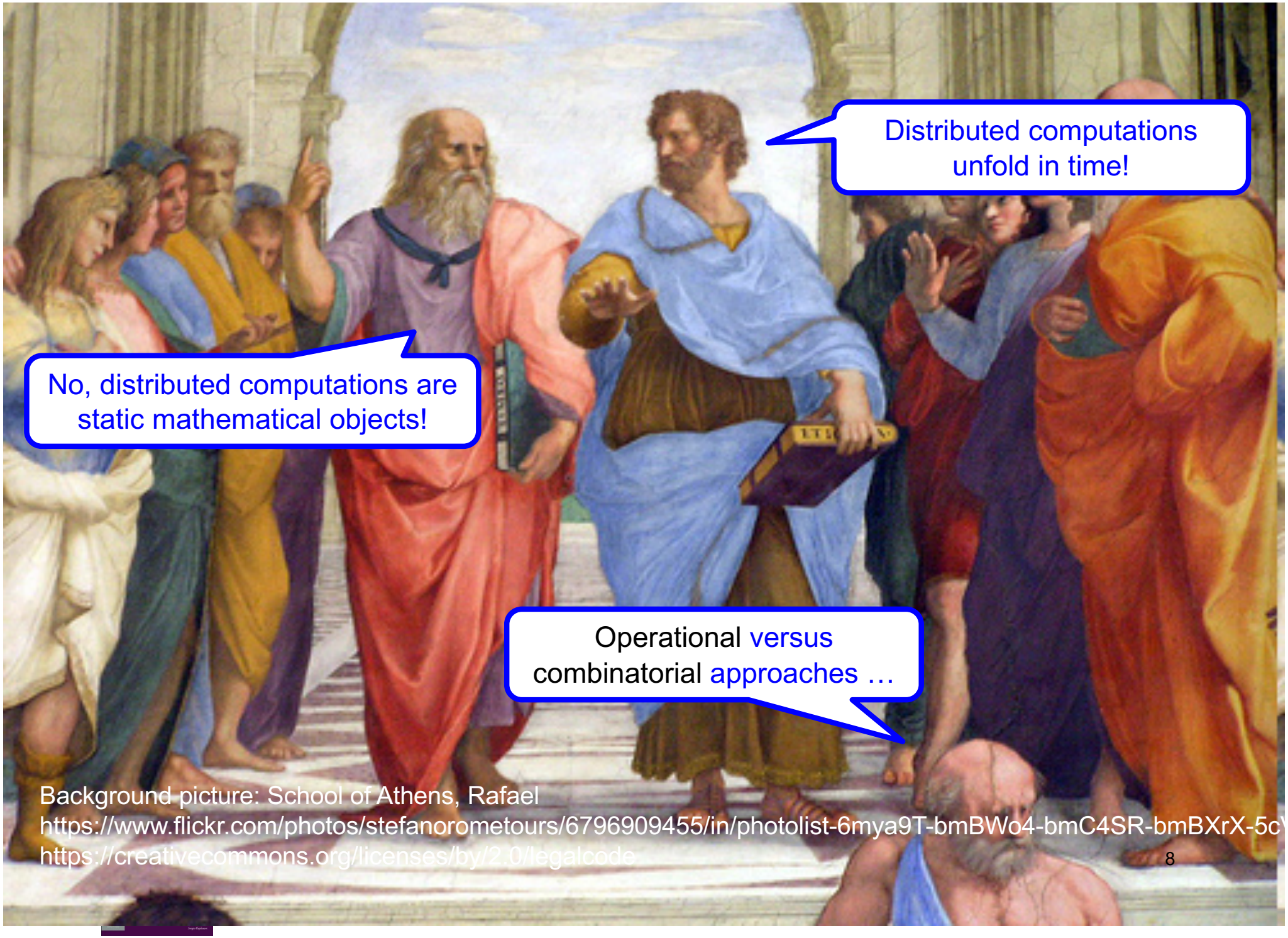
Today



Computing is co-ordination and communication




1000000
100000
10000
1000
100
10
1
0.1
1971
1975
1979
1983
1987
1991
1995
1999
2003
2007
Clock Speed (MHz)
Transistors (000)



Background picture: School of Athens, Rafael
https://www.flickr.com/photos/stefanorometours/6796909455/in/photolist-6mya9T-bmBWo4-bmC4SR-bmBXrX-5c

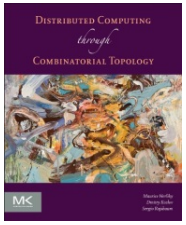
Road Map

Distributed Computing

Two Classic Distributed Problems

The Muddy Children

Coordinated Attack



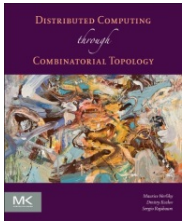
Road Map

Distributed Computing

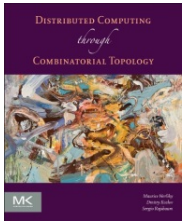
Two Classic Distributed Problems

The Muddy Children

Coordinated Attack

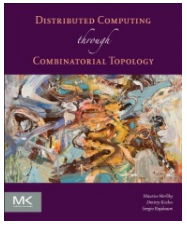


There are Many Models



There are Many Models

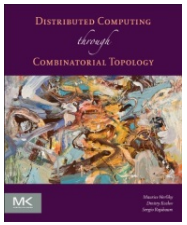
Communication?



There are Many Models

Communication?

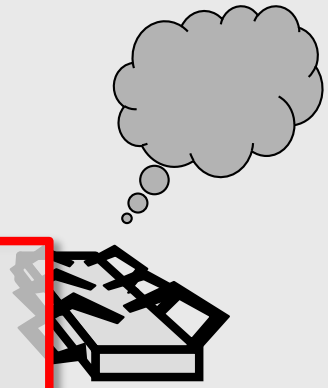
Failures?



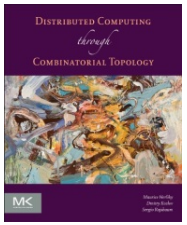
There are Many Models

Communication?

Timing?



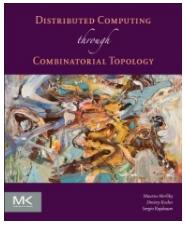
Failures?



Communication

http://commons.wikimedia.org/wiki/File:Pennyblack-pd.jpg

Message-Passing

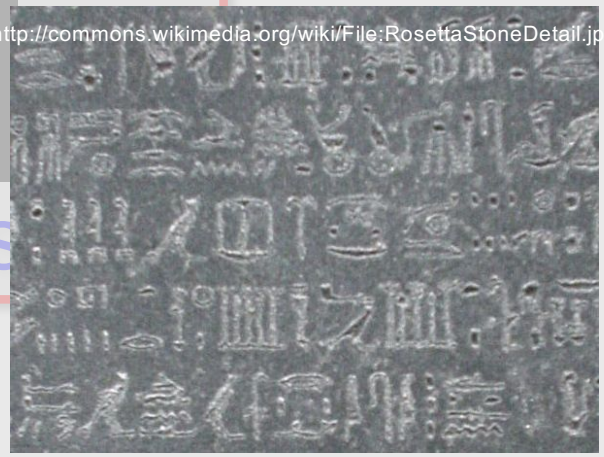


Communication

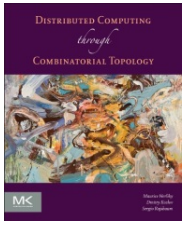
http://commons.wikimedia.org/wiki/File:Pennyblack-pd.jpg

Message-Pas

http://commons.wikimedia.org/wiki/File:RosettaStoneDetail.jpg



Read-Write Memory

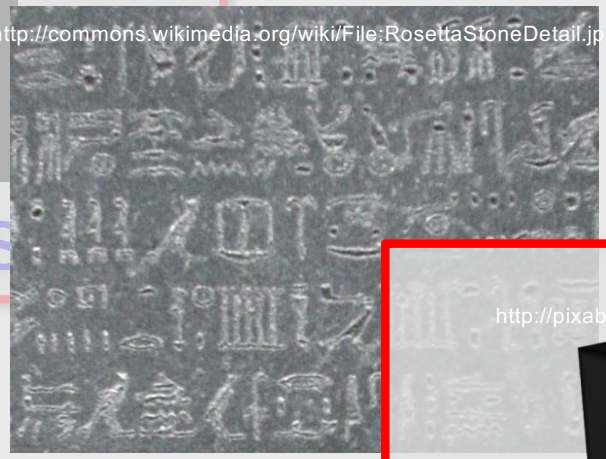


Communication

http://commons.wikimedia.org/wiki/File:Pennyblack-pd.jpg

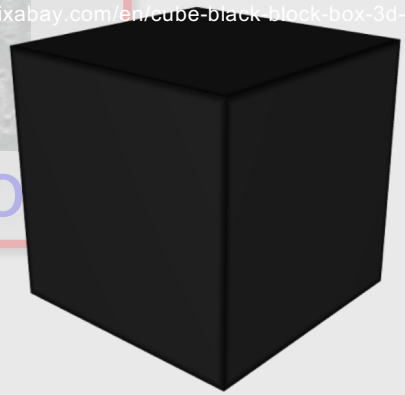
Message-Pas

http://commons.wikimedia.org/wiki/File:RosettaStoneDetail.jpg

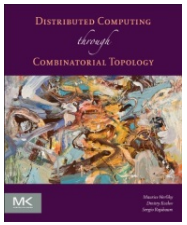


Read-Write Memo

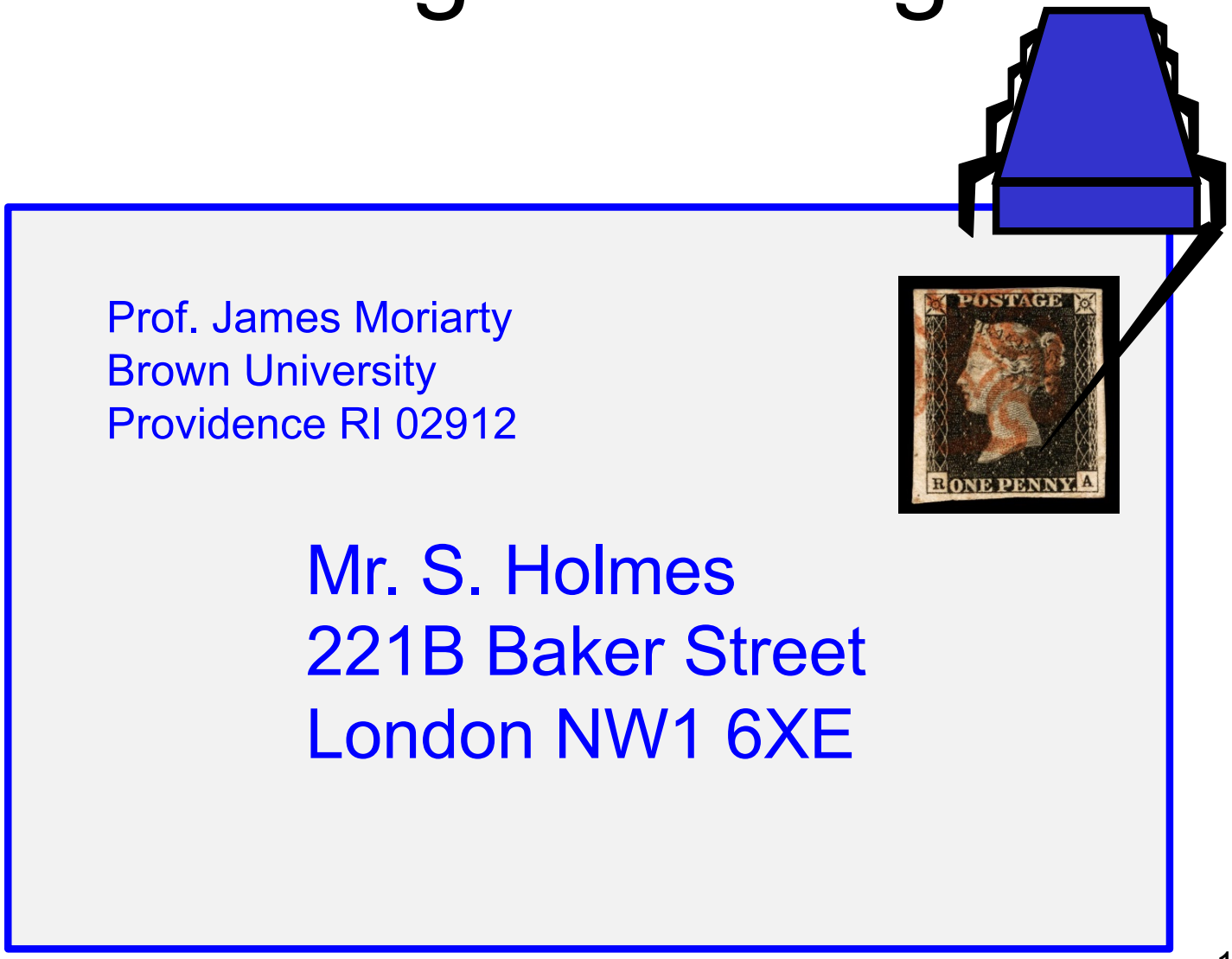
http://pixabay.com/en/cube-black-block-box-3d-250082/

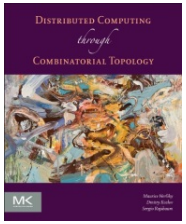


Black-Box Memory

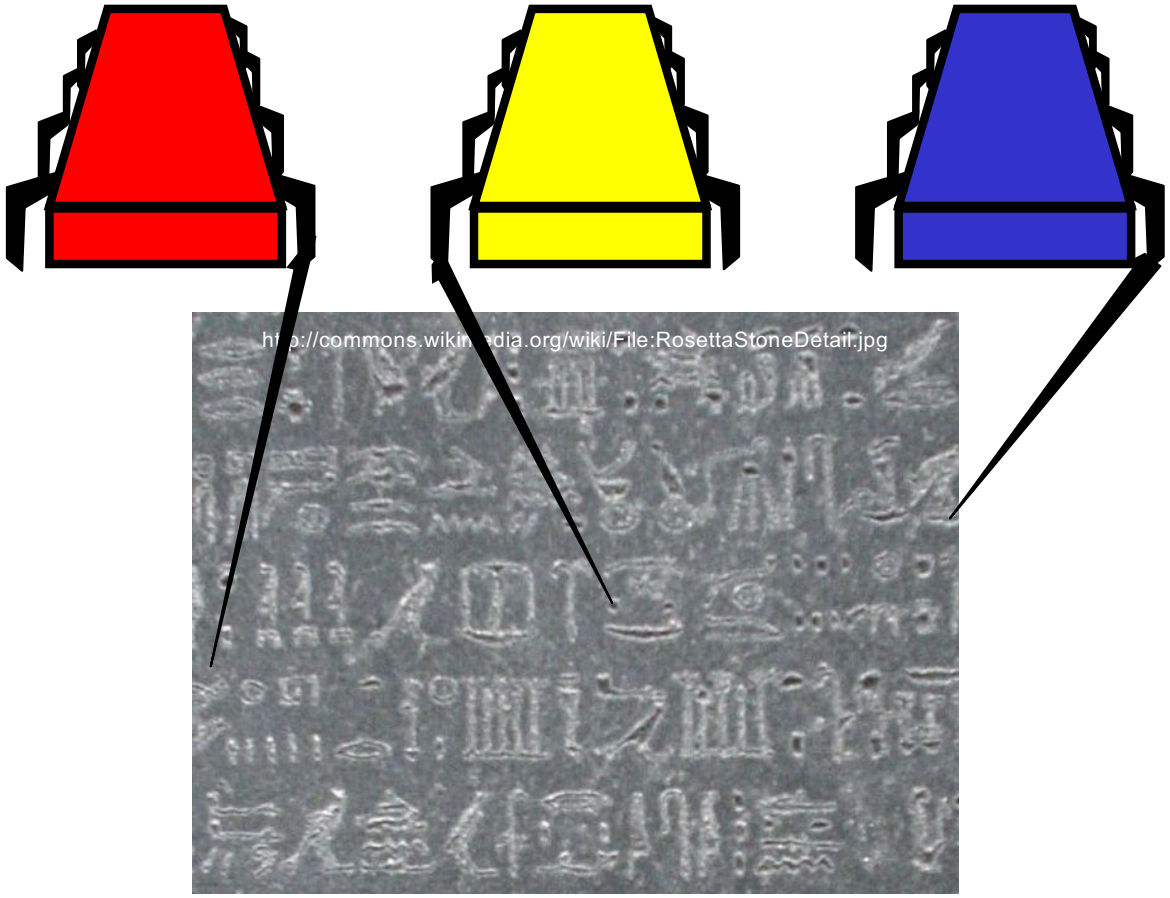


Message-Passing

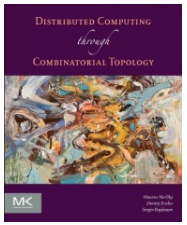




Read-Write Memory



http://www.alpa.ch/en/news/2011/dead-sea-scrolls

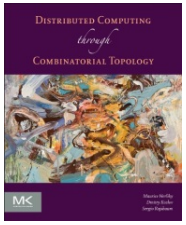


Read-Write Models

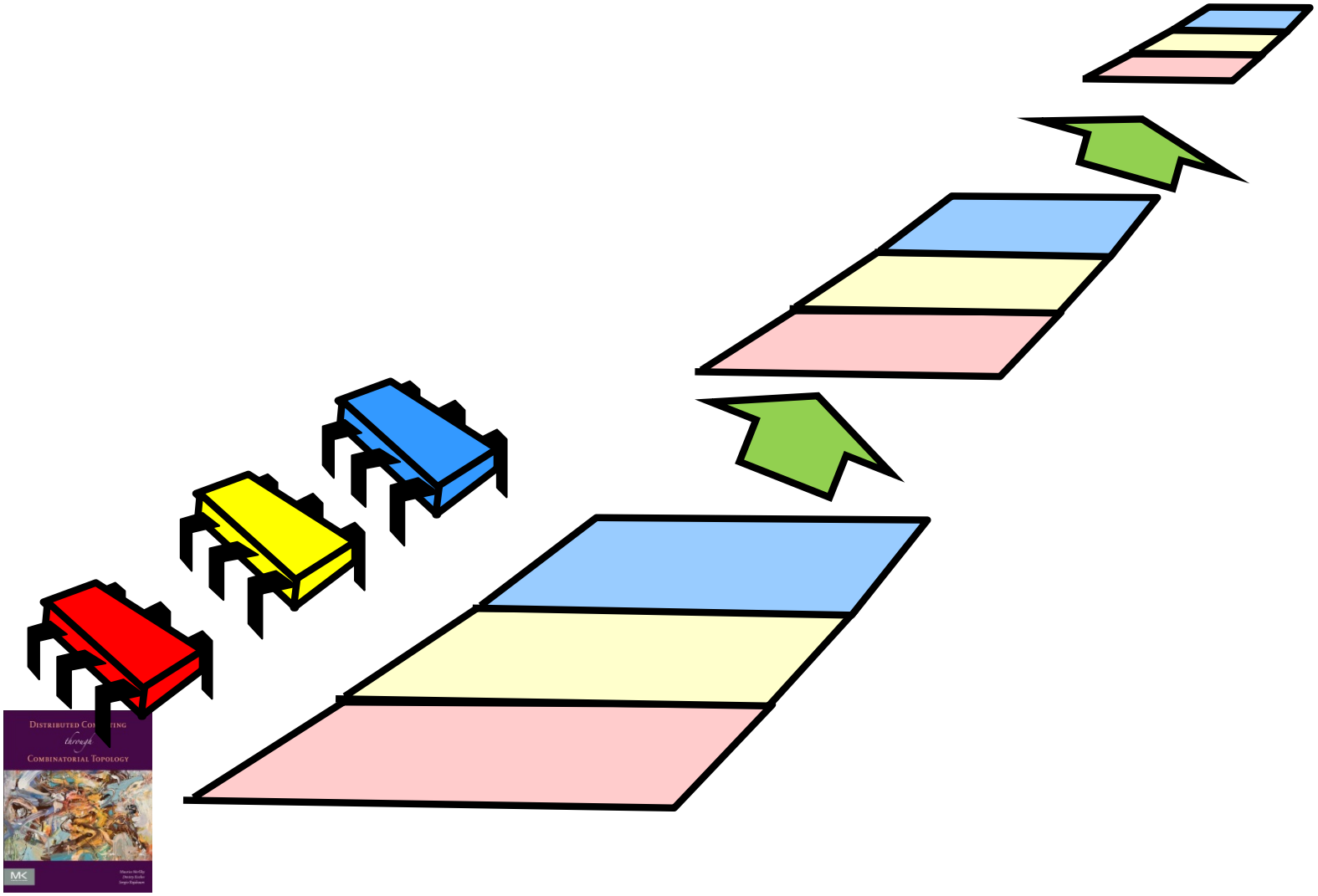
write & read individual locations

write & take memory snapshot

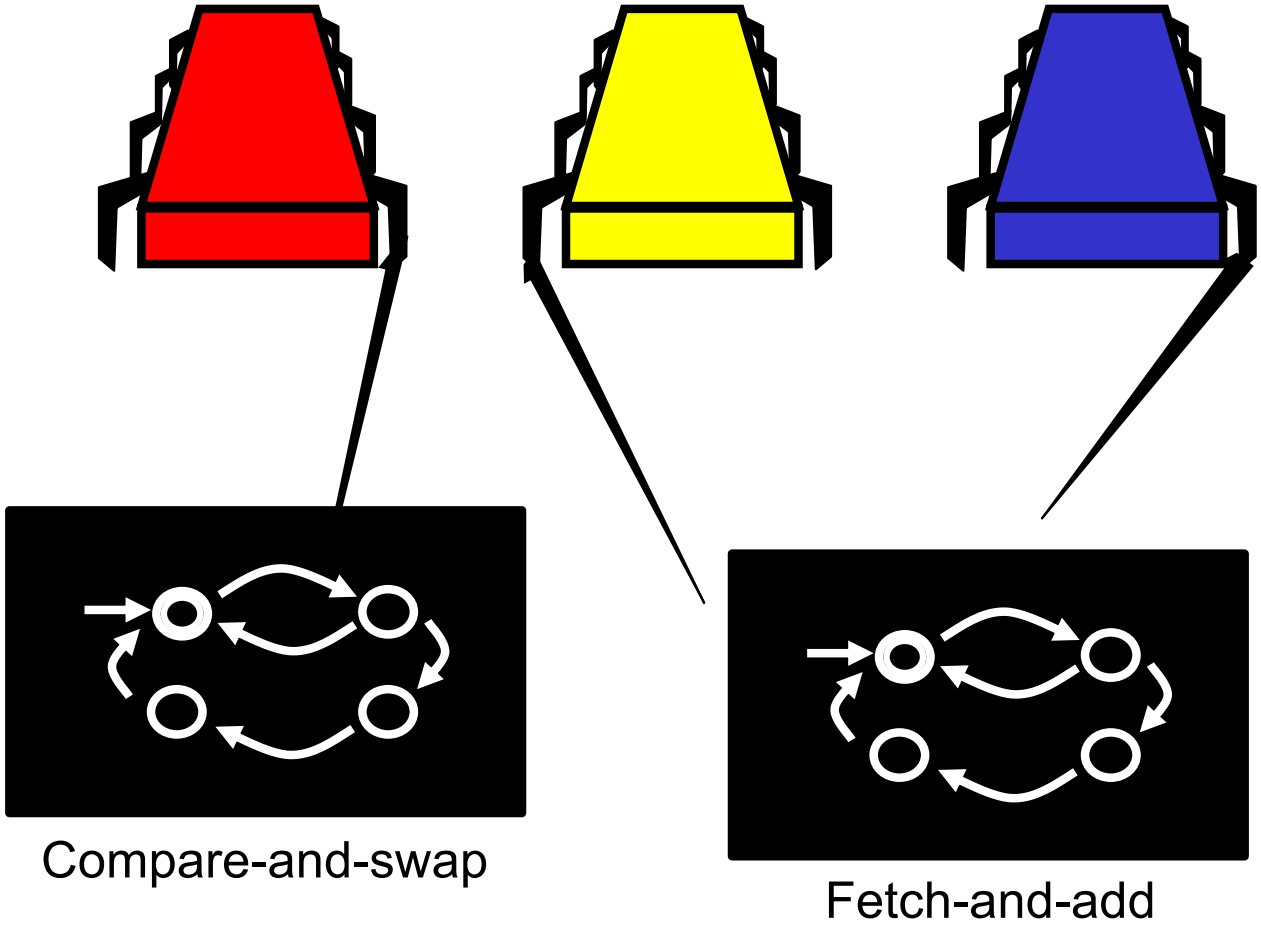
Group writes together then takes snapshots together

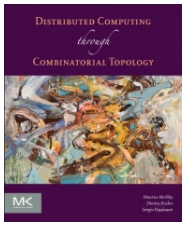


Layered Read-Write Memory

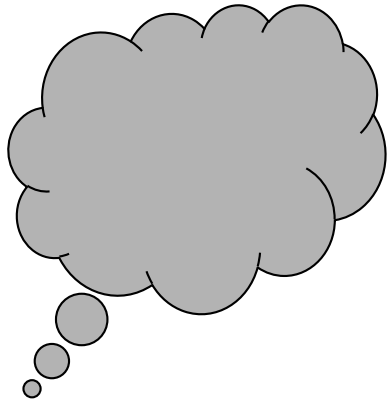


Black Box Memory

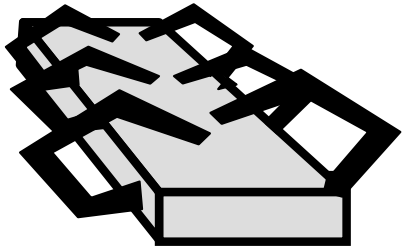




Failures

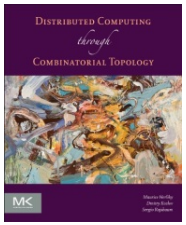


Crash failures: processes halt

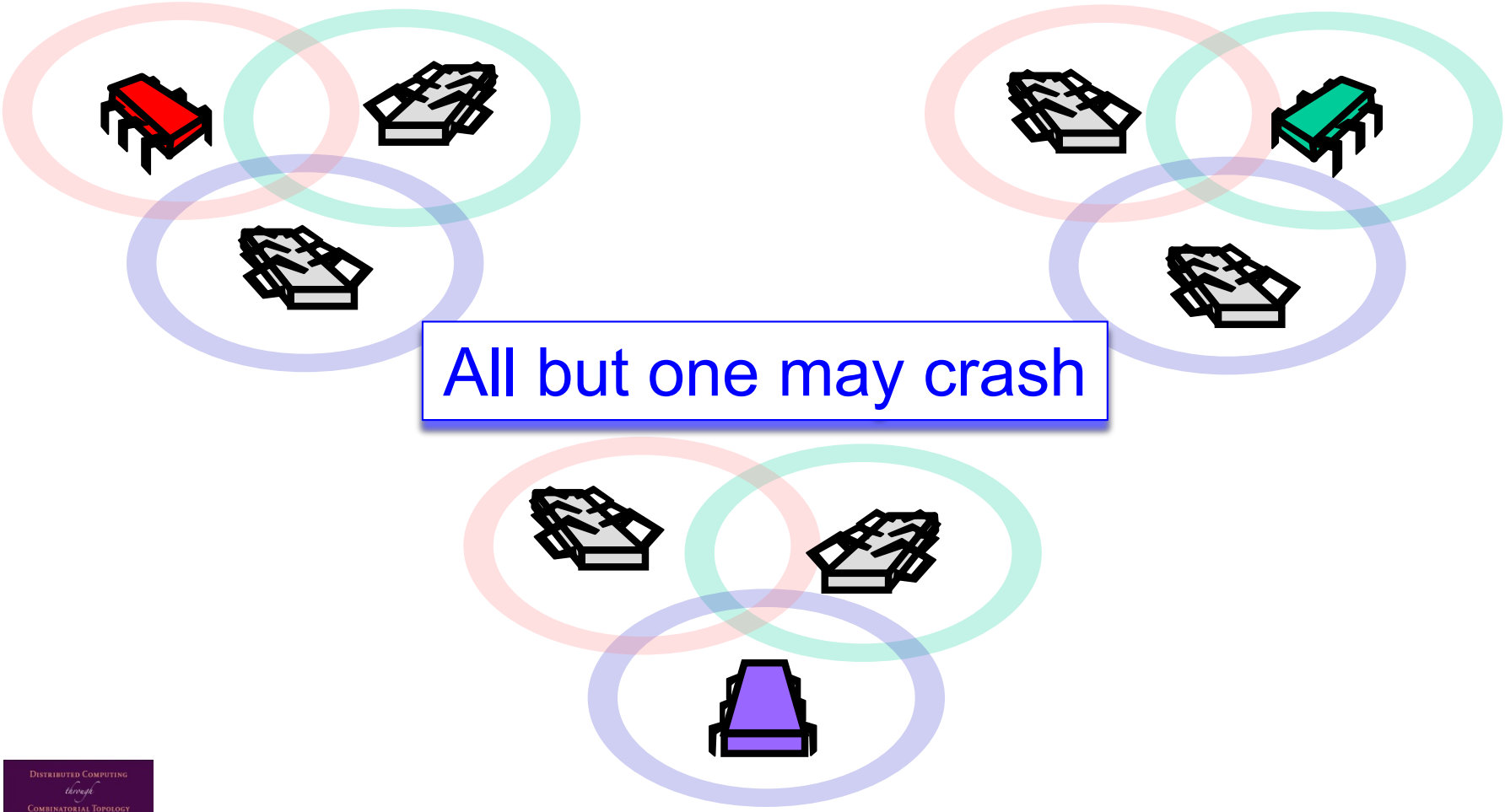


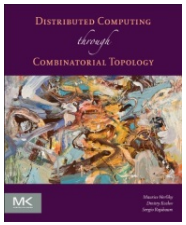
How many?

Which ones?

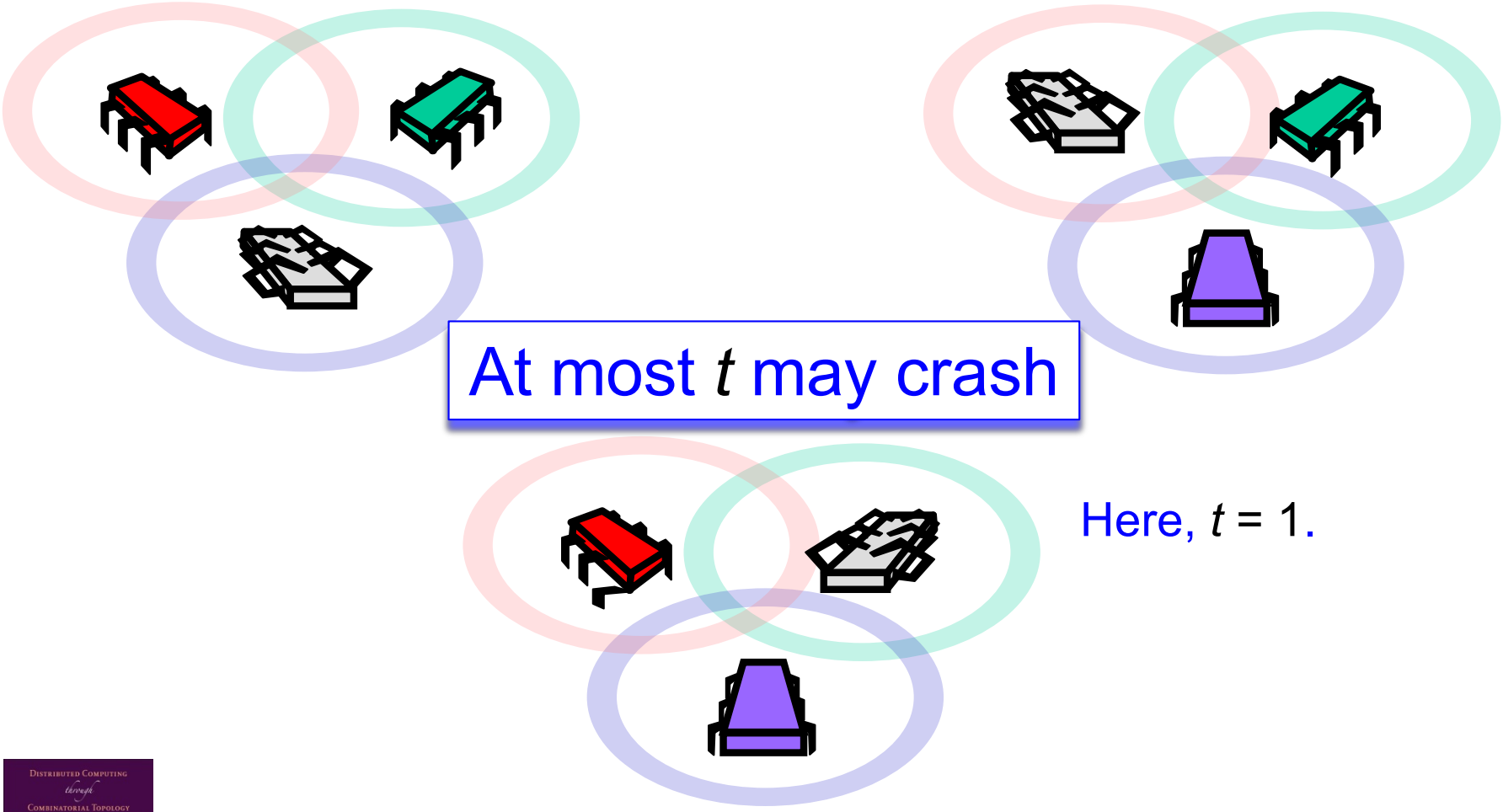


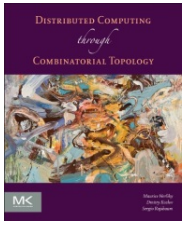
Wait-Free Failure Model



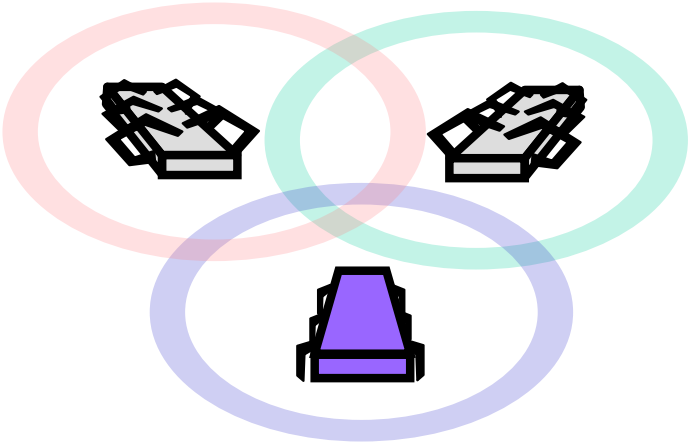


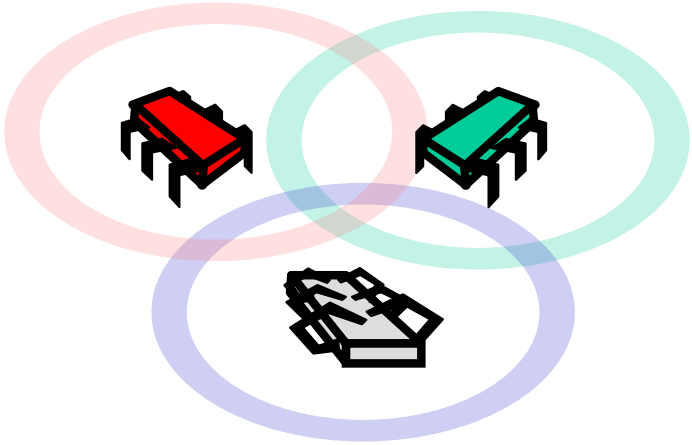
t-Resilient Failure Model



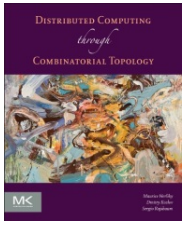


Correlated Failures

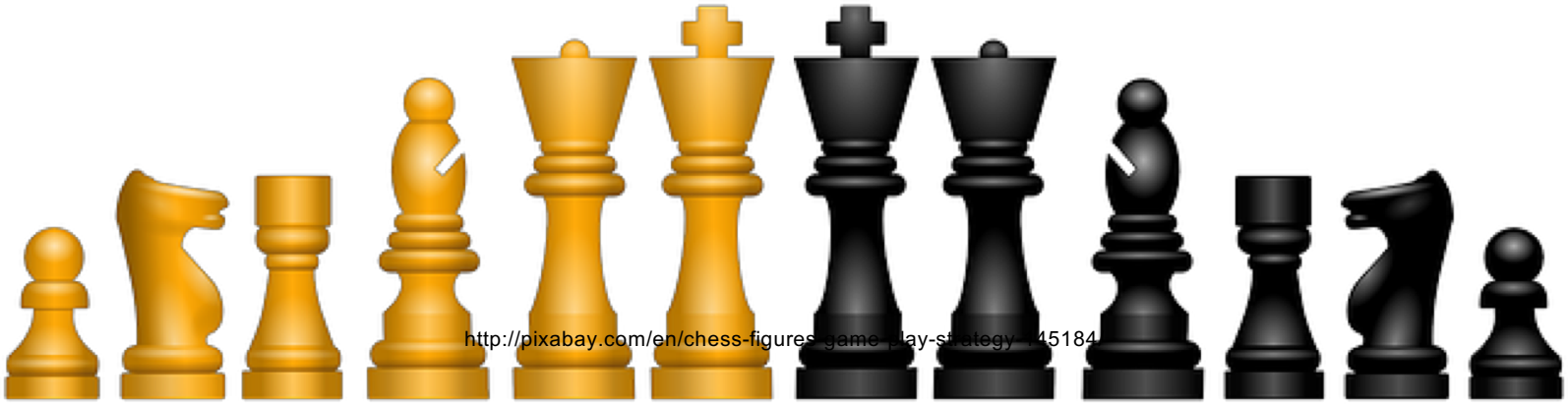


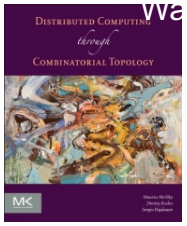


Processes on same server may crash



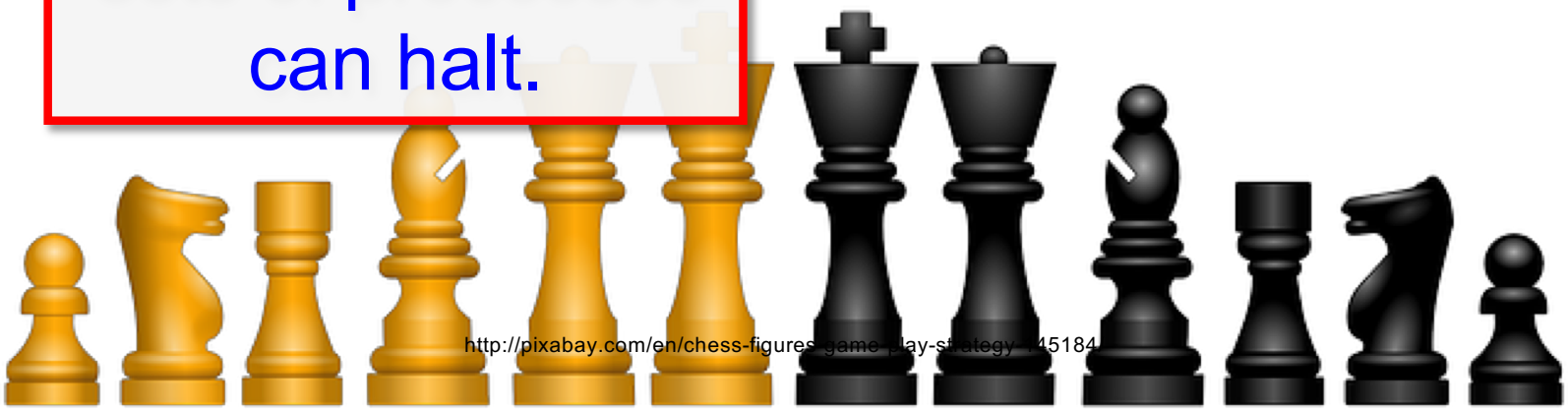
Adversaries

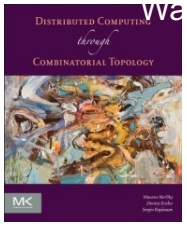




Adversaries

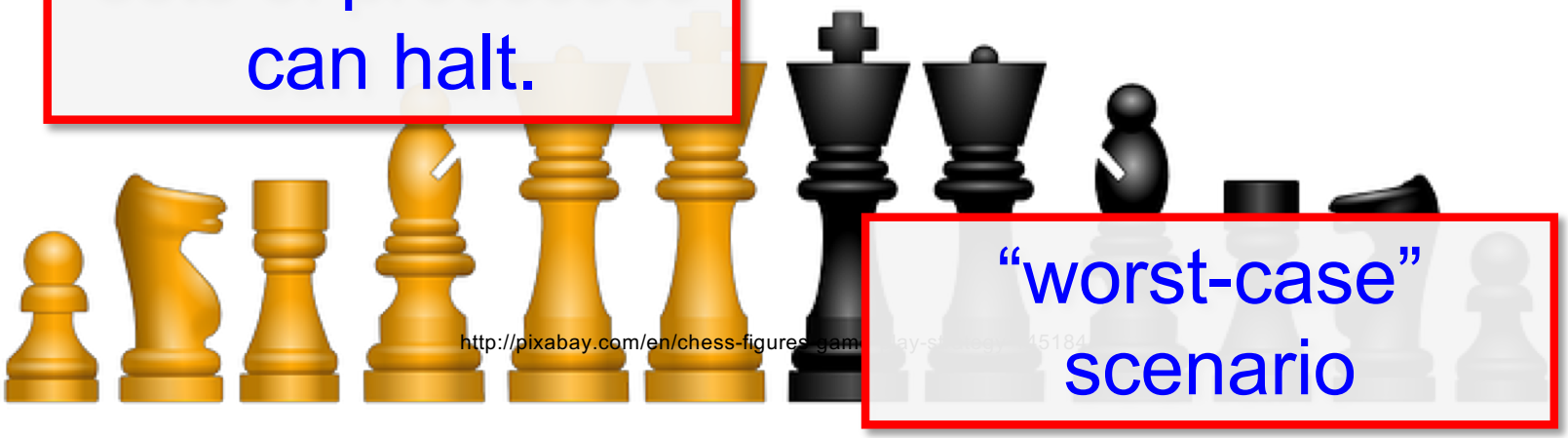
Determine which sets of processes can halt.



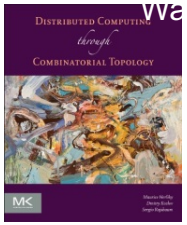


Adversaries

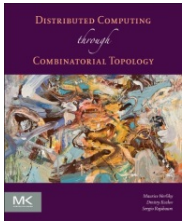
Determine which sets of processes can halt.



"worst-case" scenario

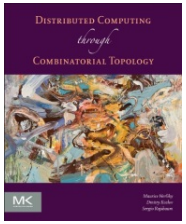


Timing Models



Timing Models

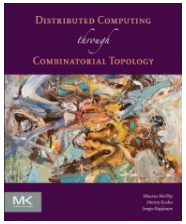
Processes share a clock



Timing Models

Processes share a clock

Synchronous

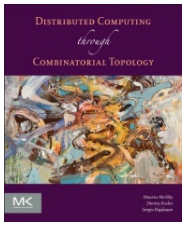


Timing Models

Processes share a clock

Synchronous

Processes do not share a clock



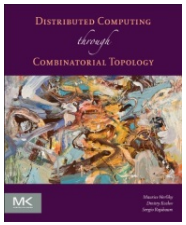
Timing Models

Processes share a clock

Synchronous

Processes do not share a clock

Asynchronous



Timing Models

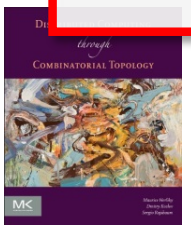
Processes share a clock

Synchronous

Processes do not share a clock

Asynchronous

Processes have approximately-synchronized clocks



Timing Models

Processes share a clock

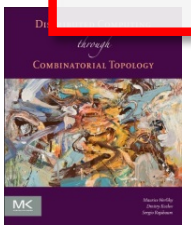
Synchronous

Processes do not share a clock

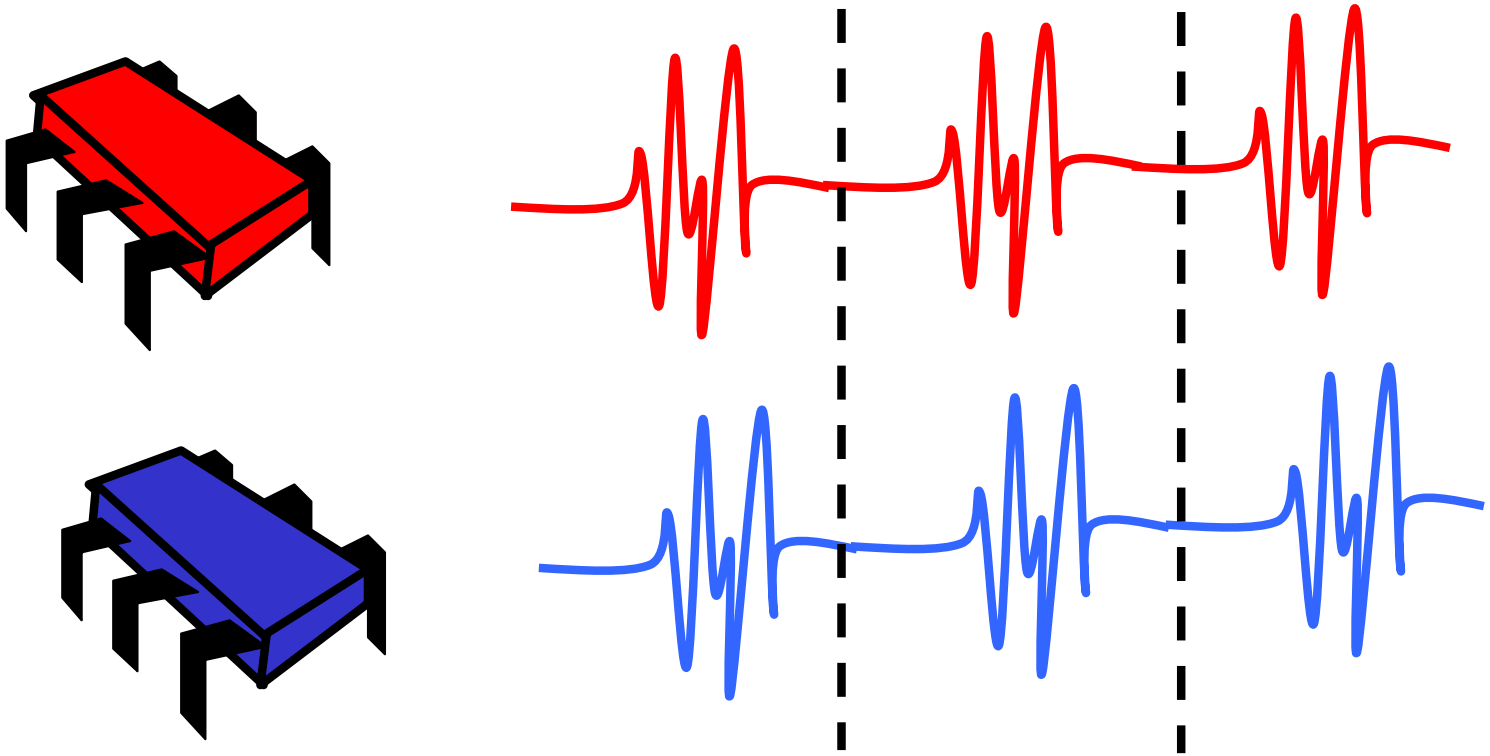
Asynchronous

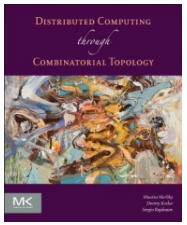
Processes have approximately-synchronized clocks

Semi-synchronous

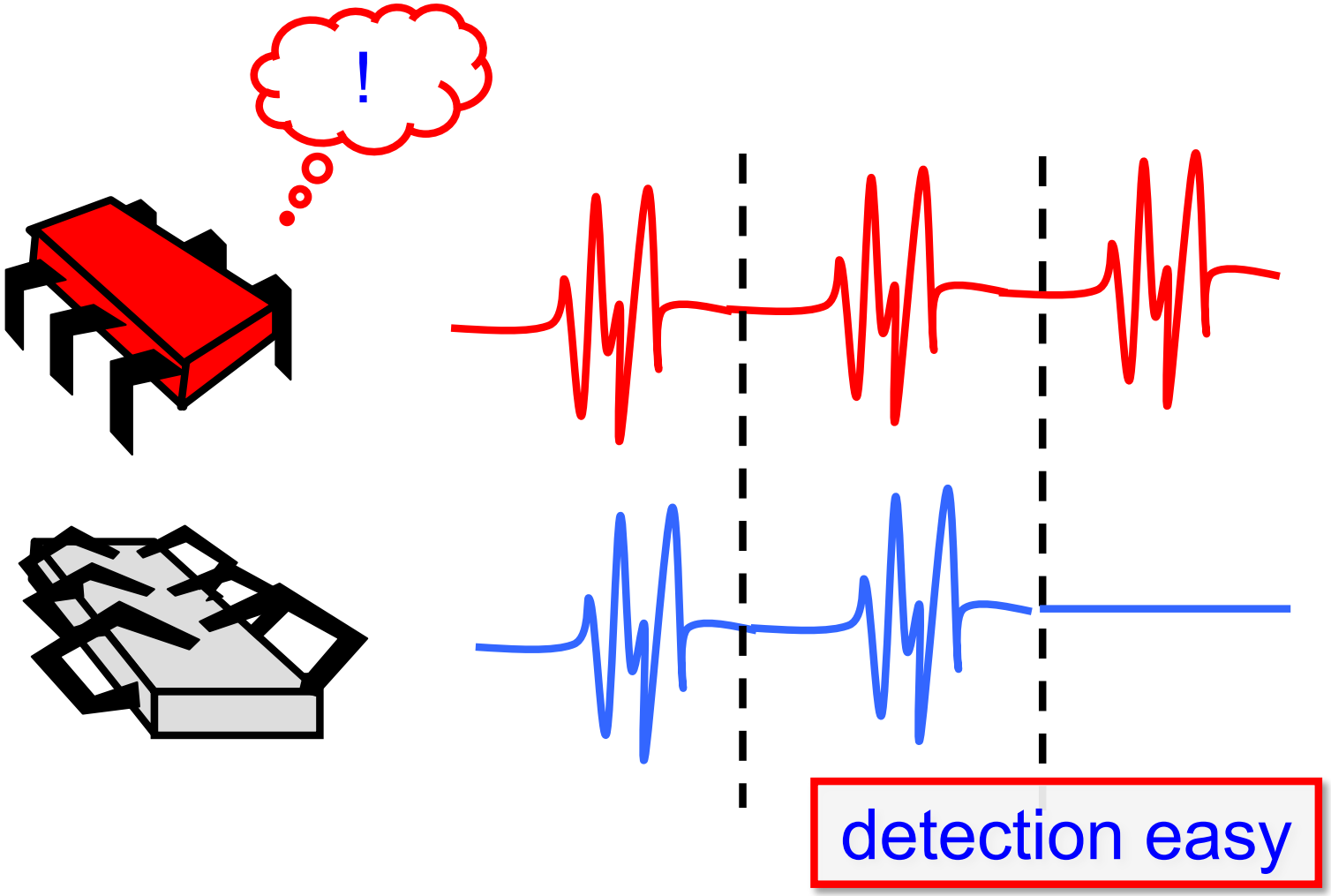


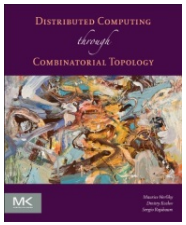
Synchronous



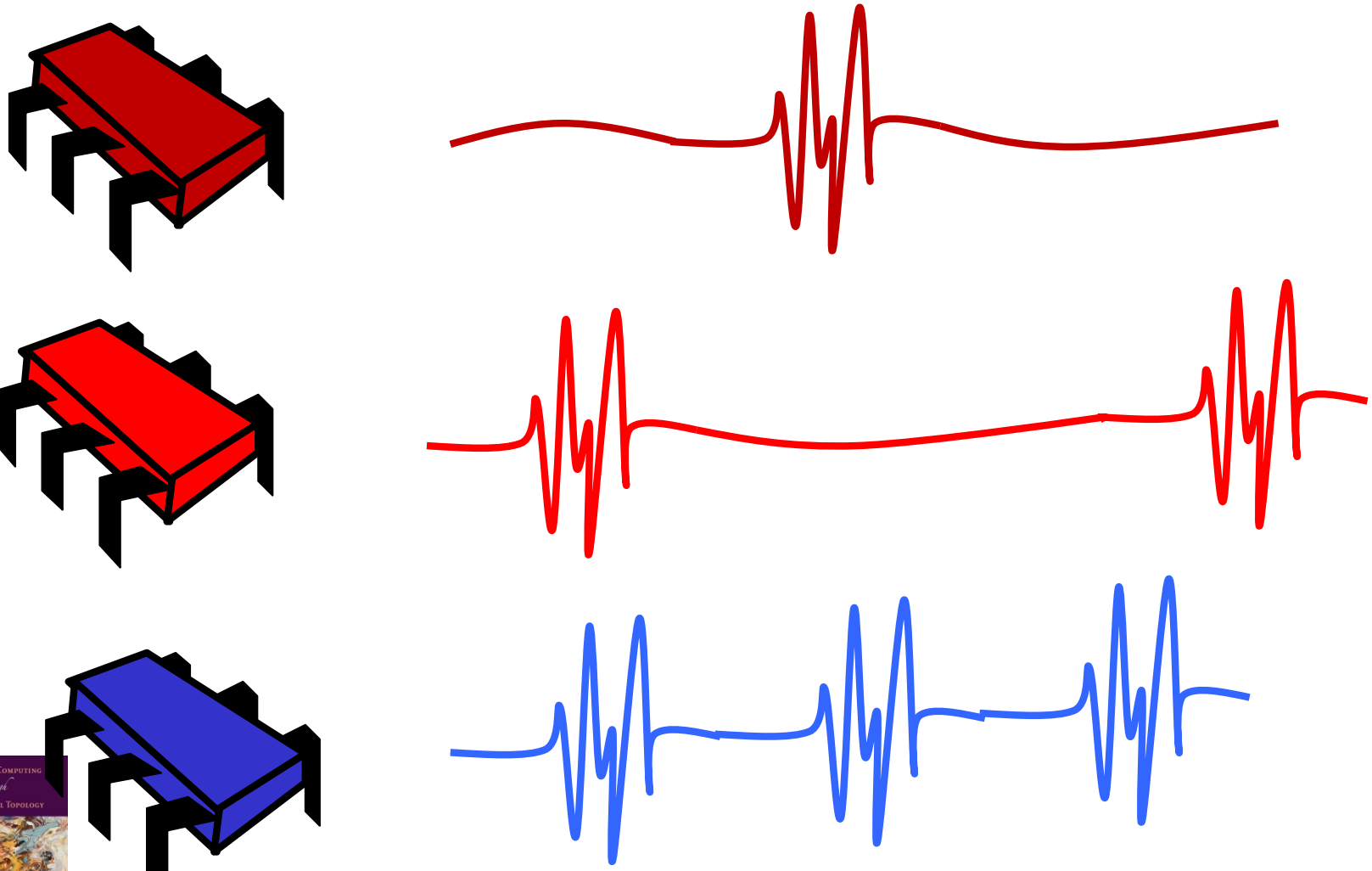


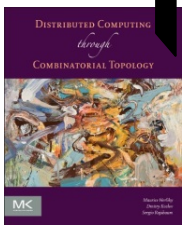
Synchronous Failures



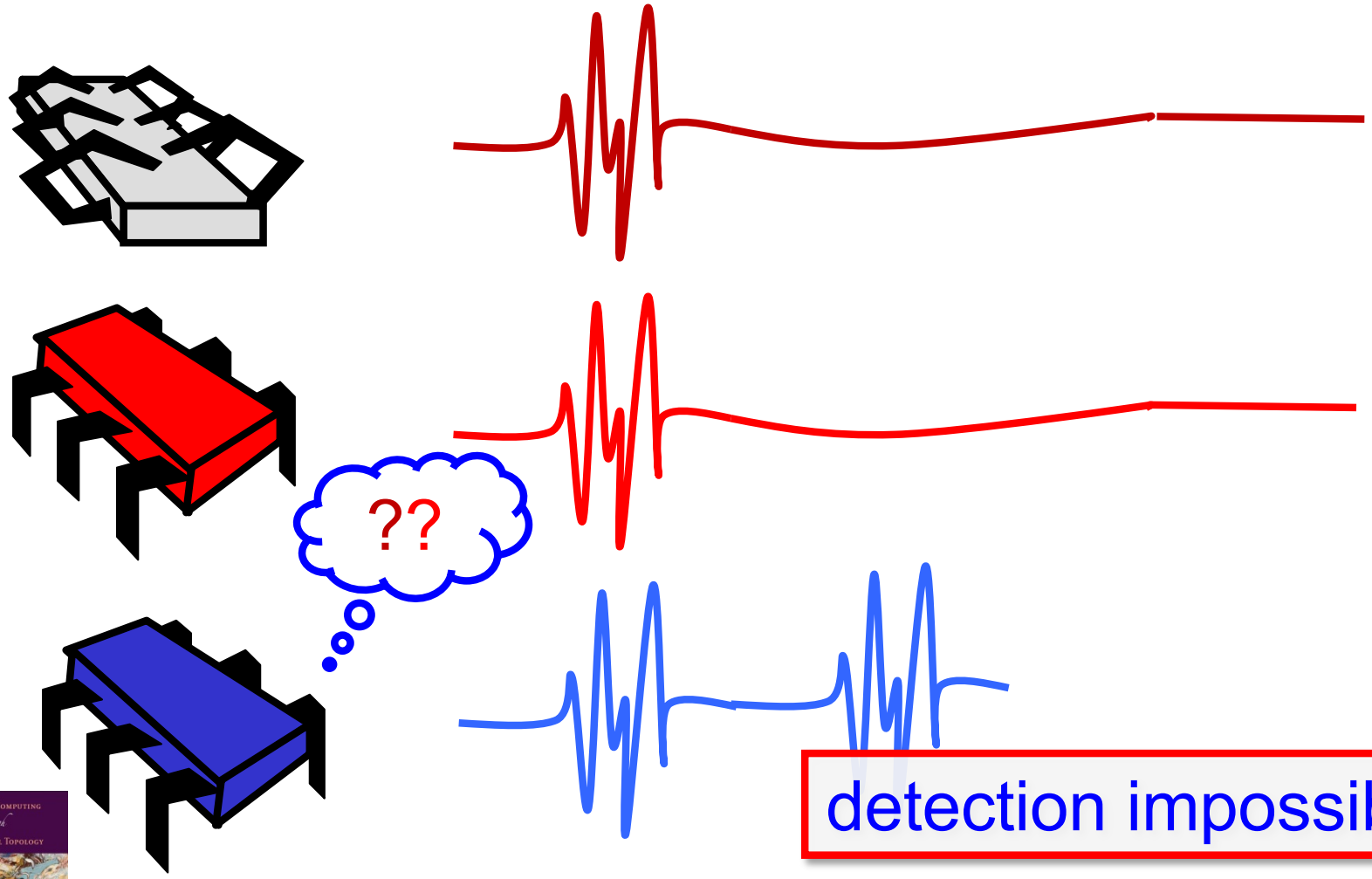


Asynchronous

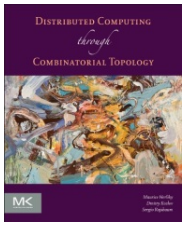




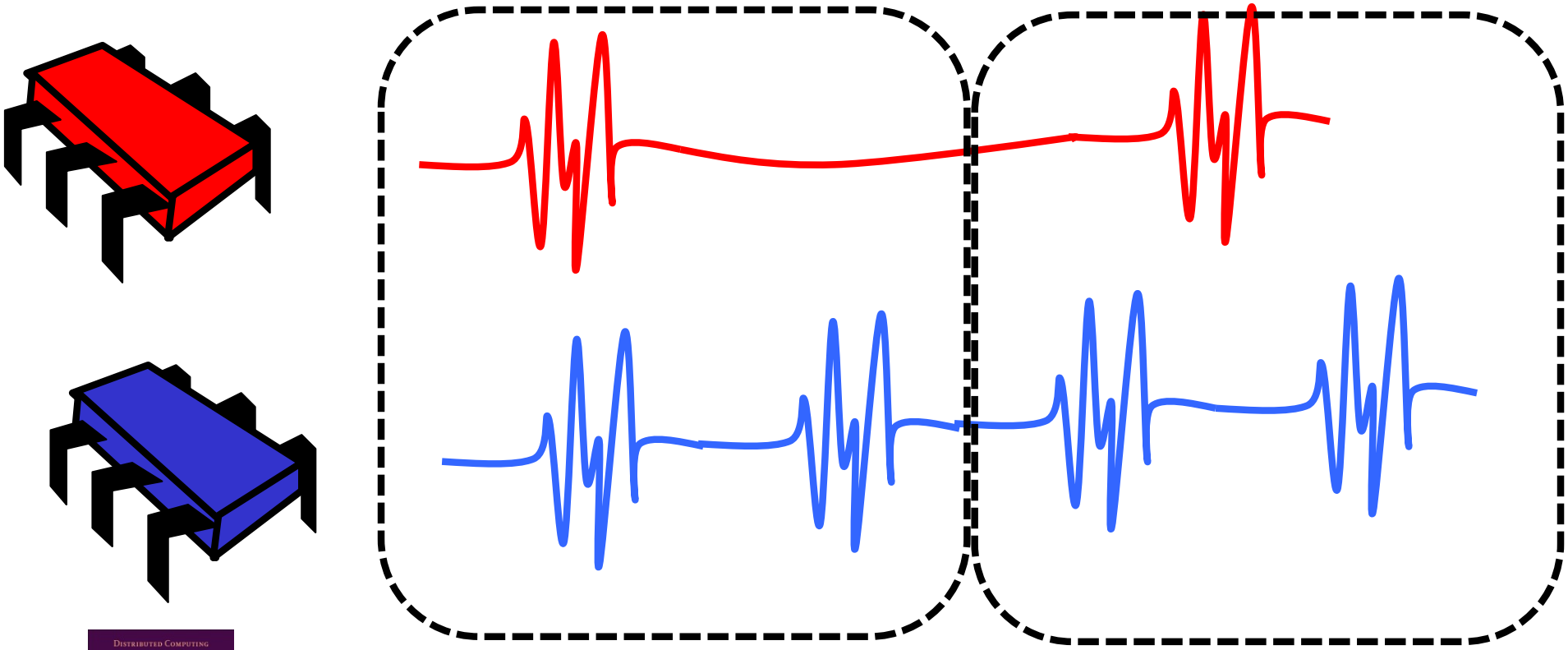
Asynchronous Failures

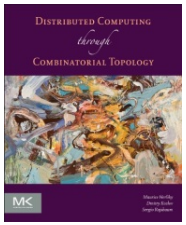


detection impossible

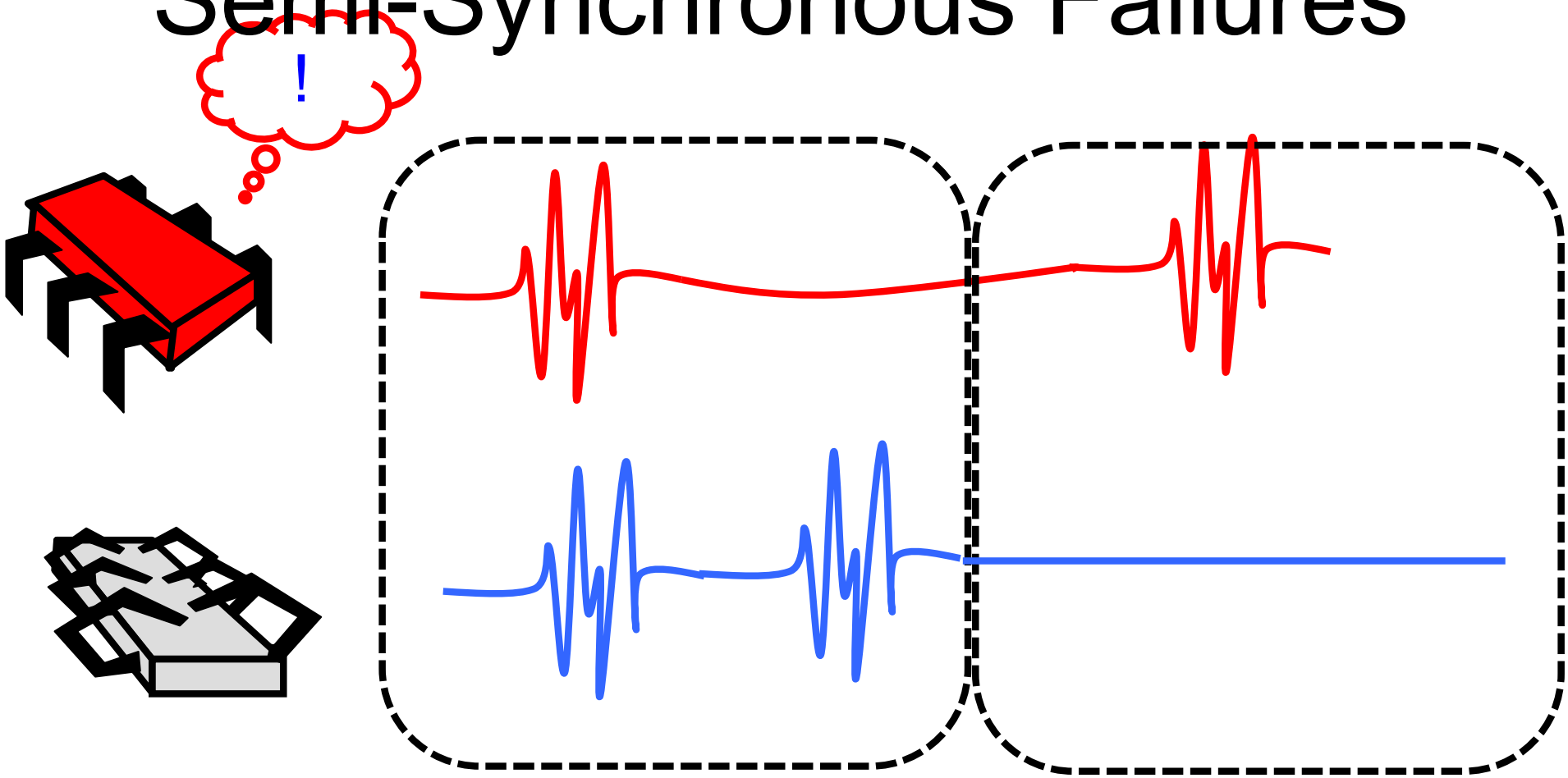


Semi-Synchronous

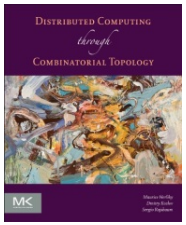




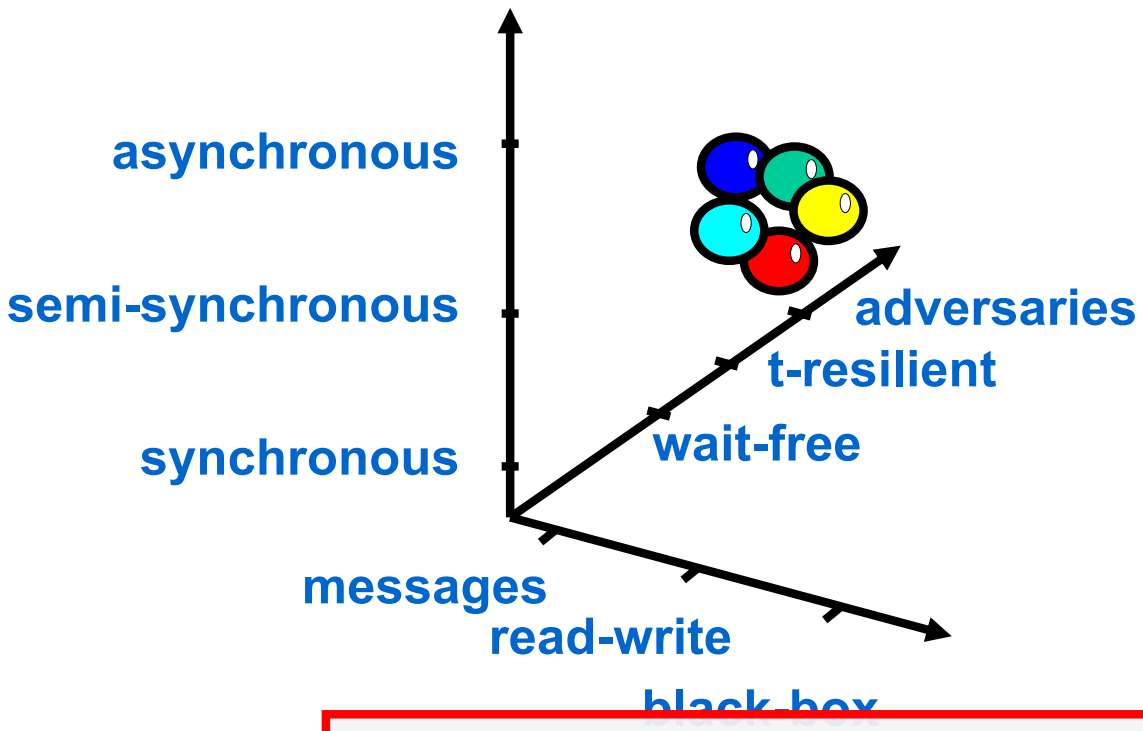
Semi-Synchronous Failures

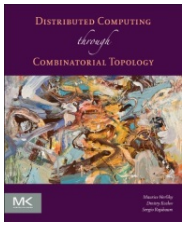


detection slow



Computation Model Space

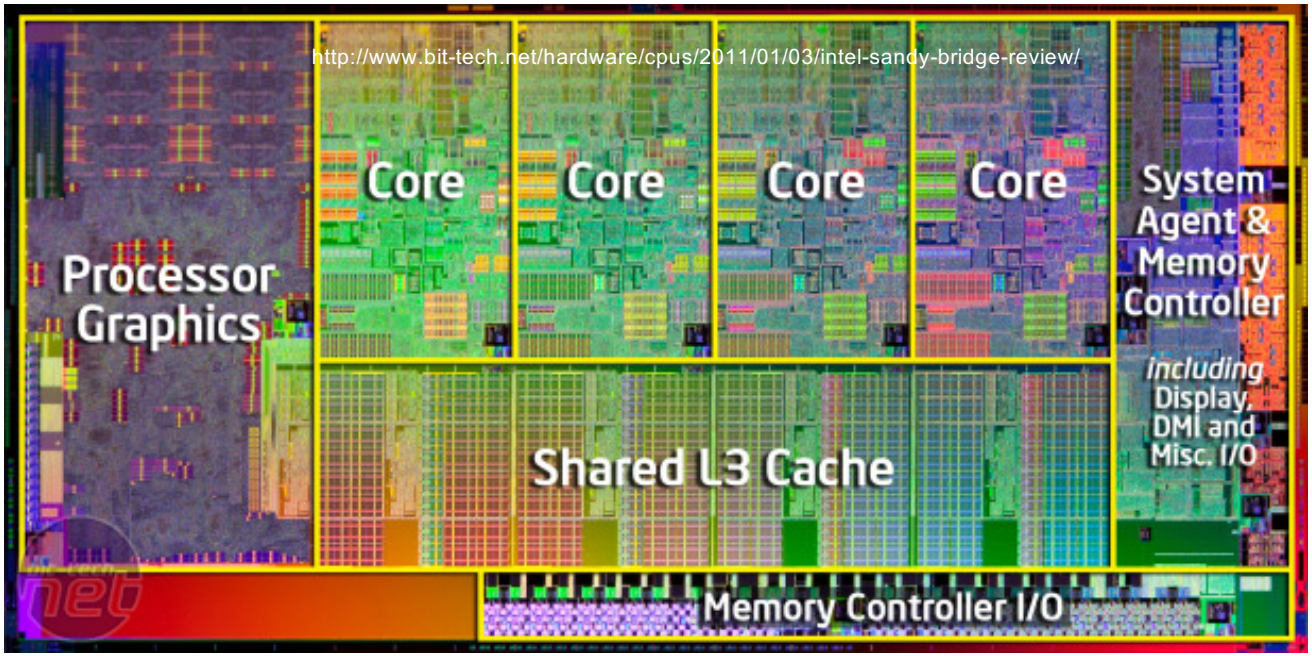


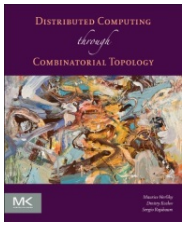


Which combinations make sense?

Multicores

Asynchronous
Wait-free
Shared Memory

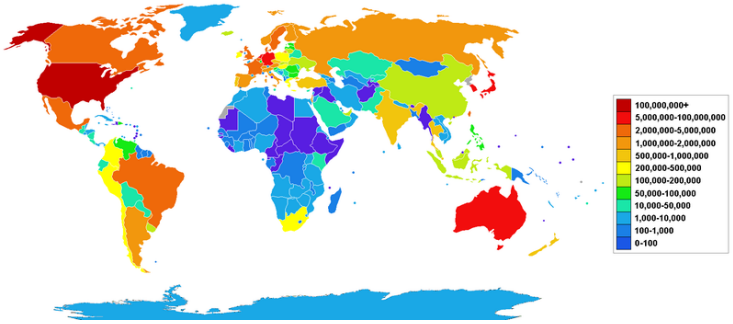




Distributed Computing

Asynchronous
Message-passing

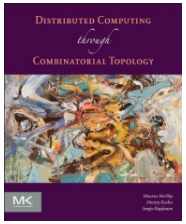
Internet



http://commons.wikimedia.org/wiki/File:Internet_hosts.PNG

Sensor
network

http://commons.wikimedia.org/wiki/File:XBee_Series_2_with_Whip_Antenna.jpg

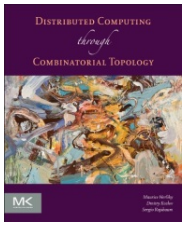


Parallel Computing

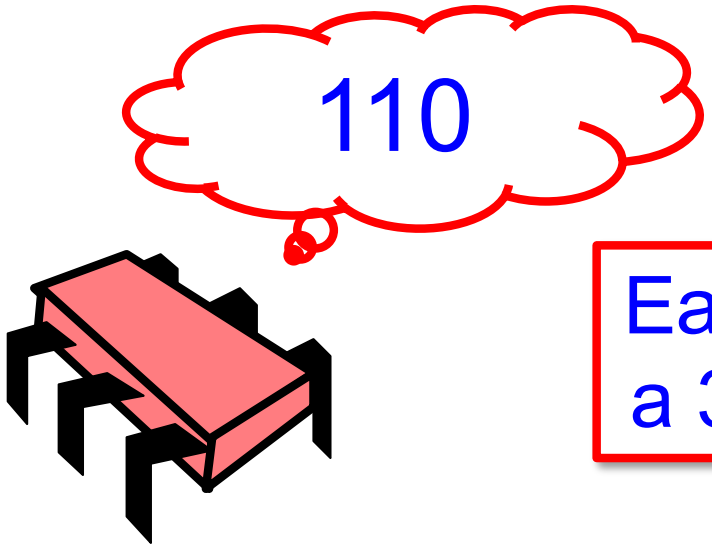
Synchronous
Message-passing
(or shared memory)

http://commons.wikimedia.org/wiki/File:GeForce2_mx400_gpu.JPG

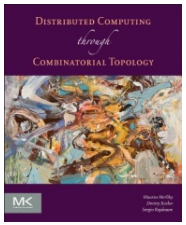
GPU



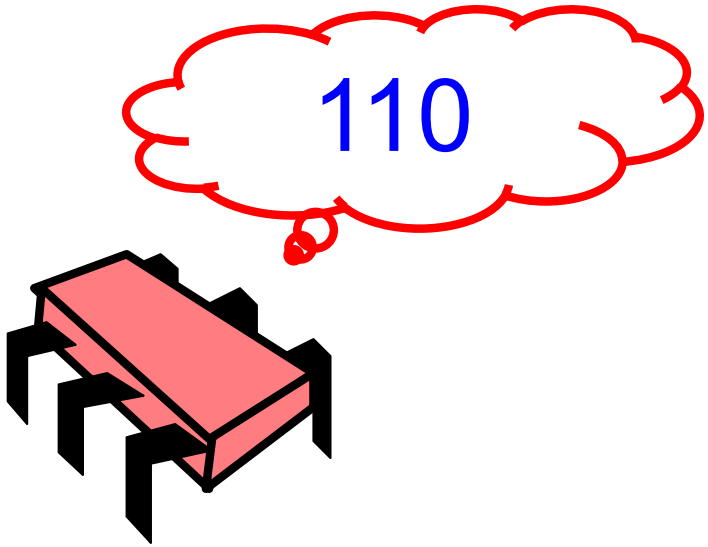
Local Views

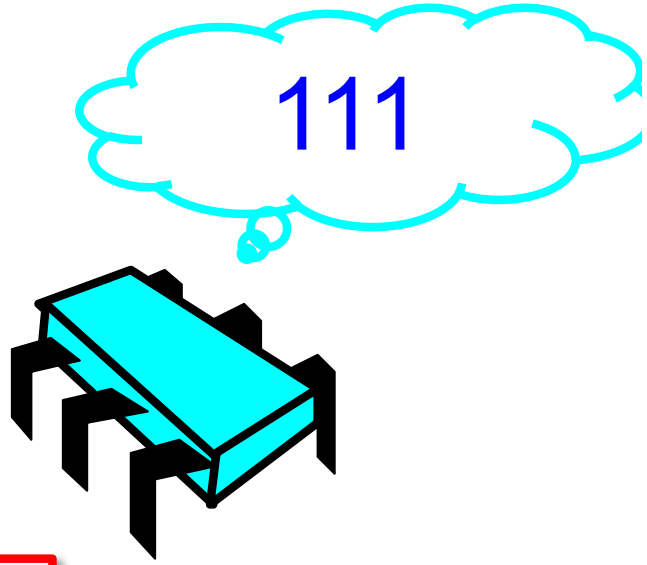


Each process has a 3-bit ***local view***

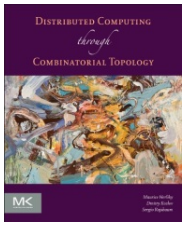


Multiple Local Views

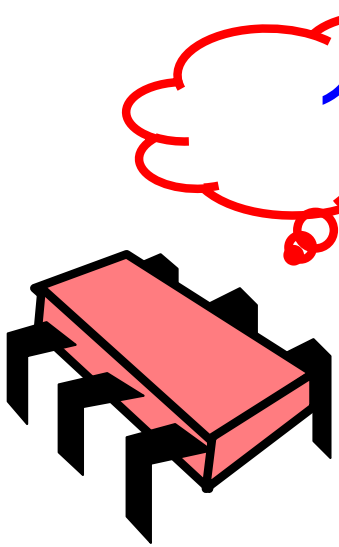


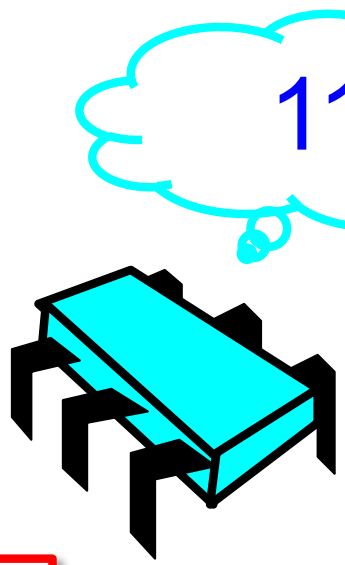


local views differ
by ***1 bit***



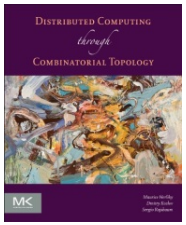
Multiple Local Views





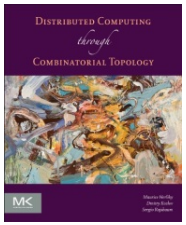
local views differ by ***1 bit***

but no process knows which one

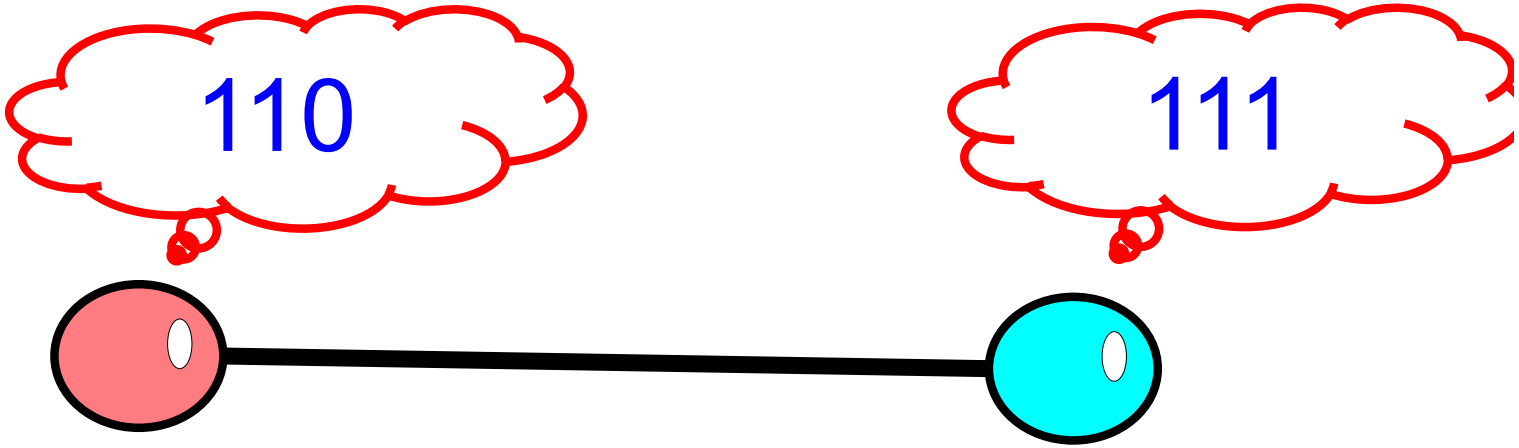


Multiple Local Views

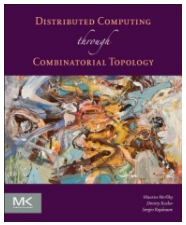
each view is represented by a ***labeled vertex***

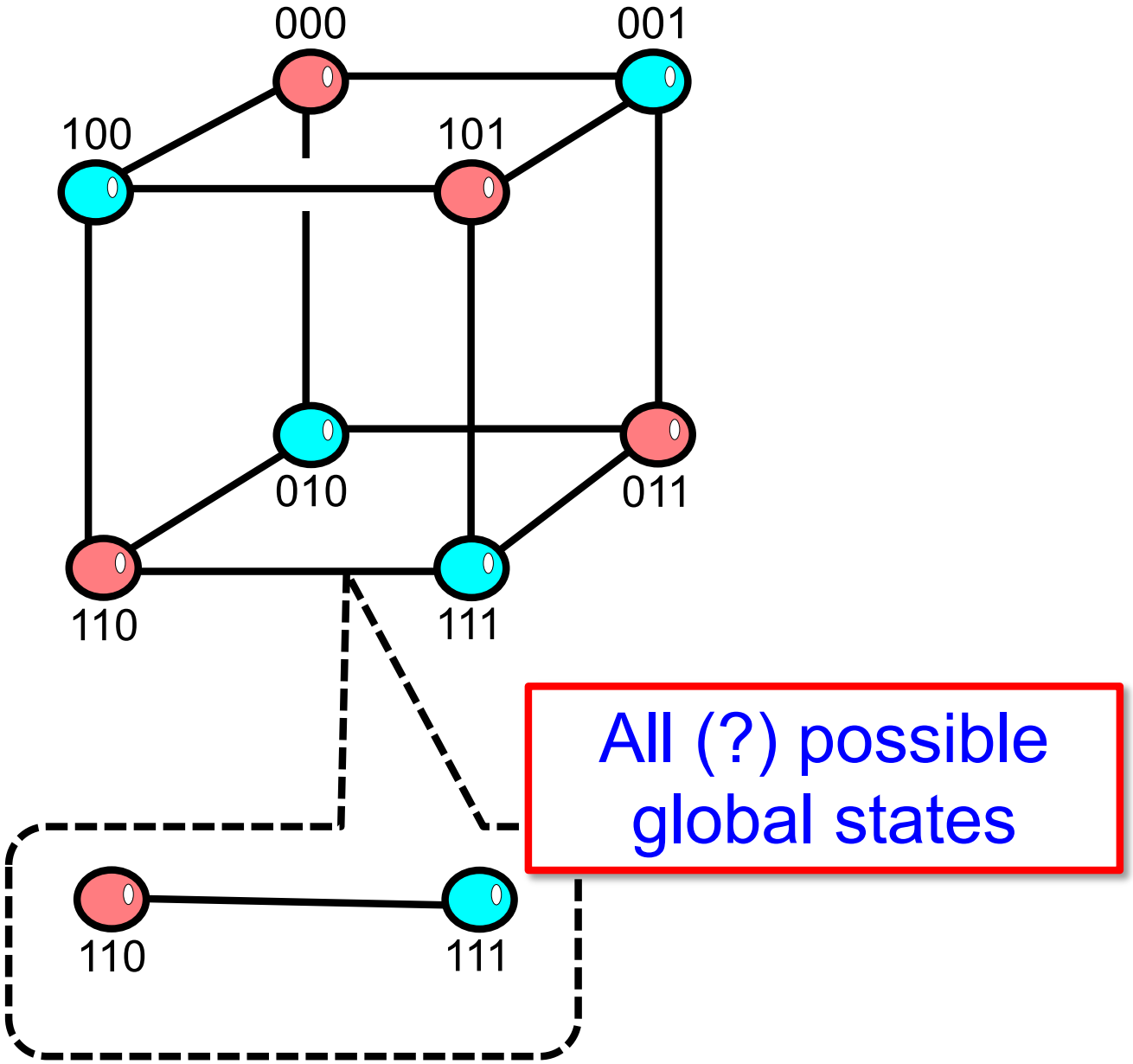


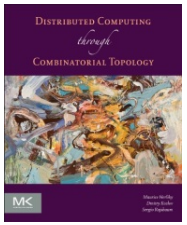
Global States



compatible views represented by an ***edge***

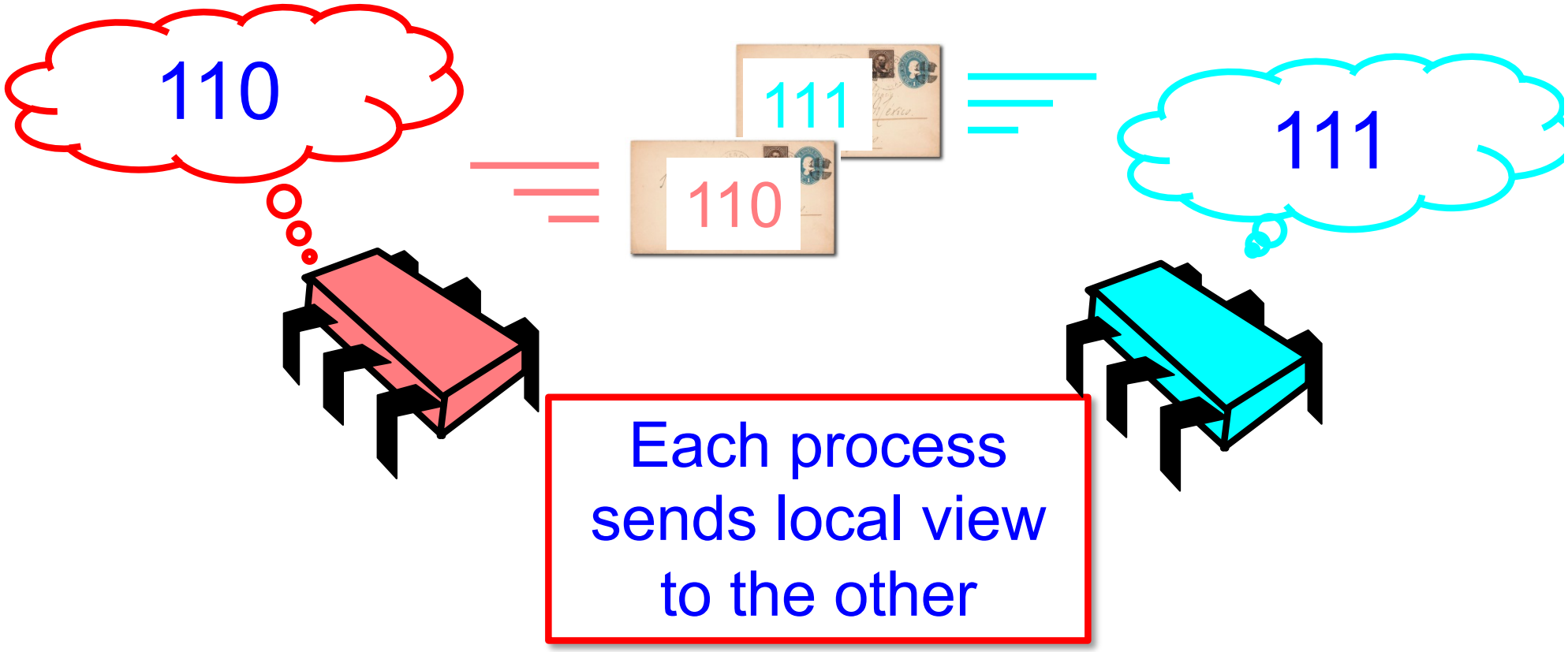


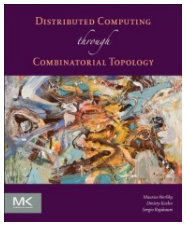




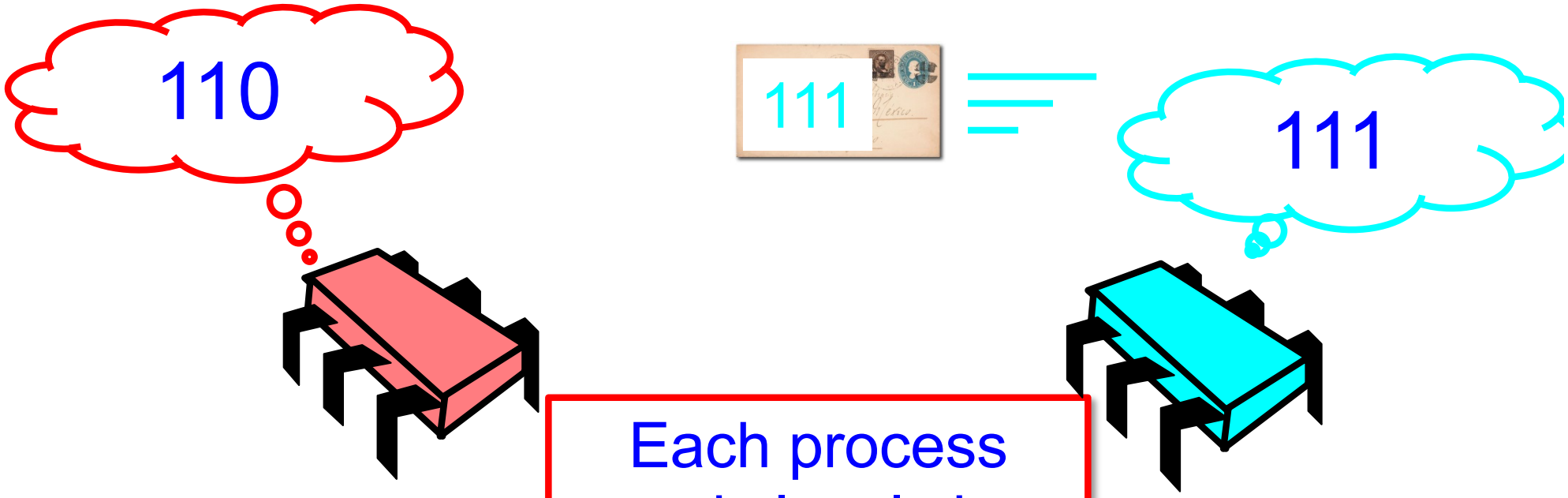
Distributed Computing though Combinatorial Topology

Communication



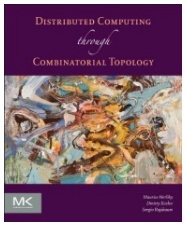


Communication

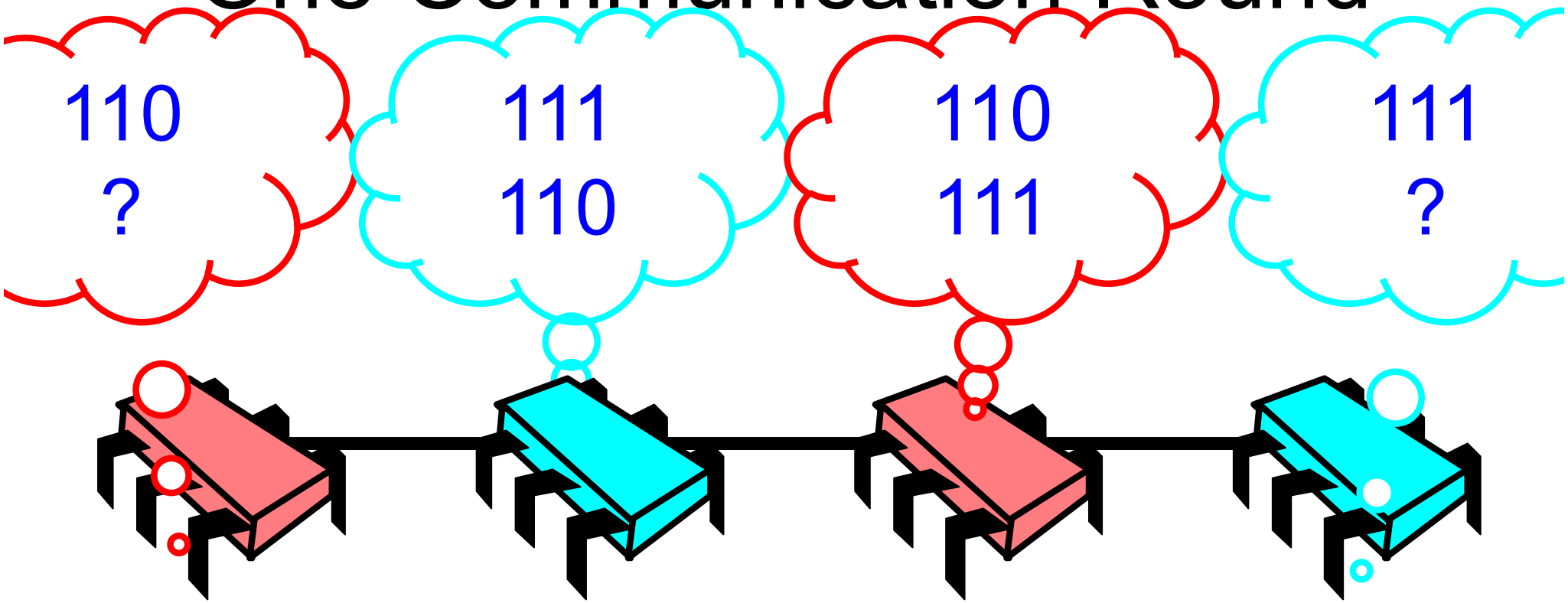


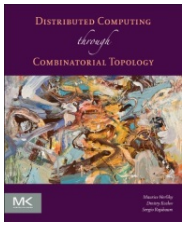
Each process sends local view to the other

but at most one message may be lost!

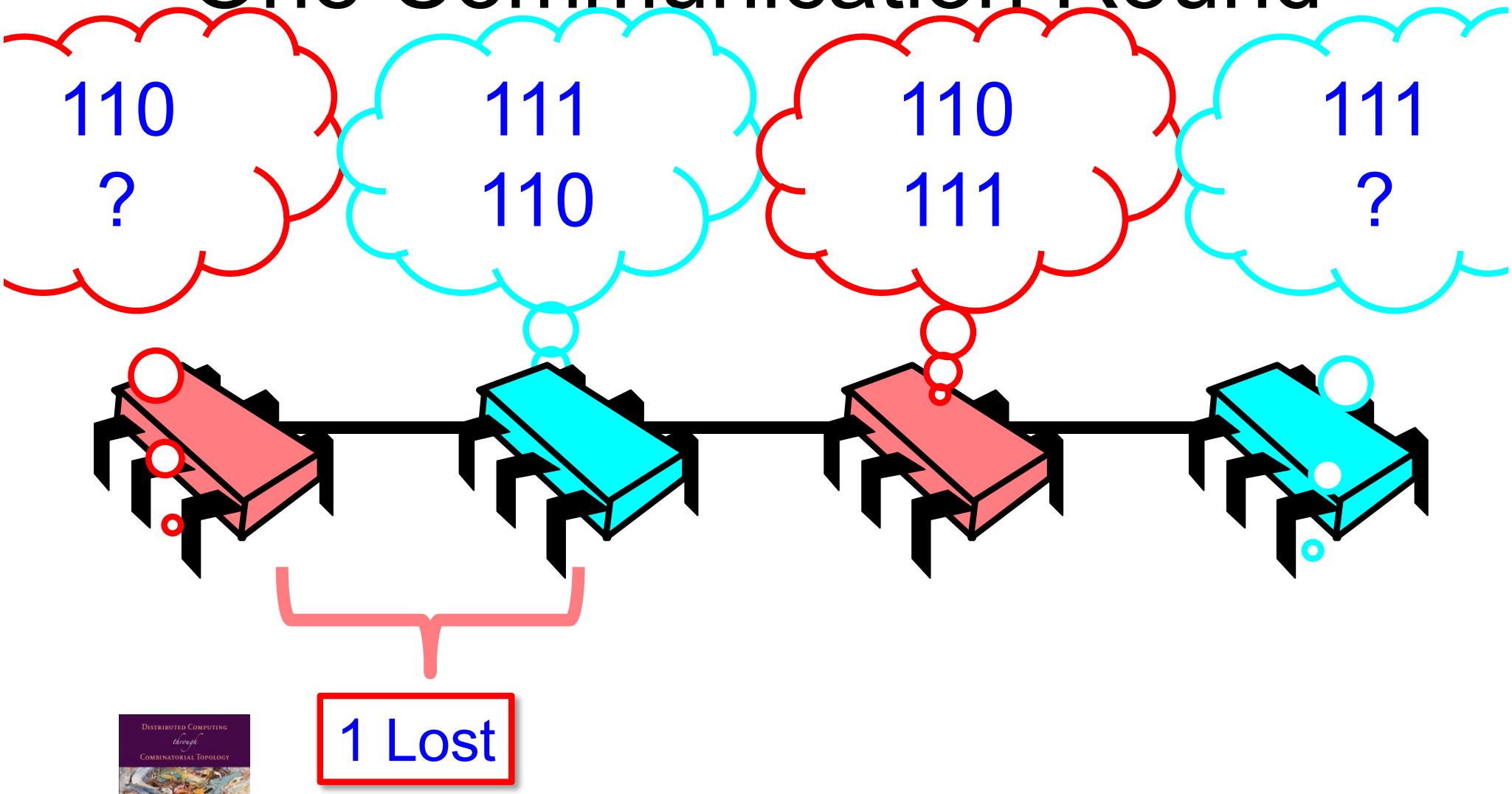


One Communication Round

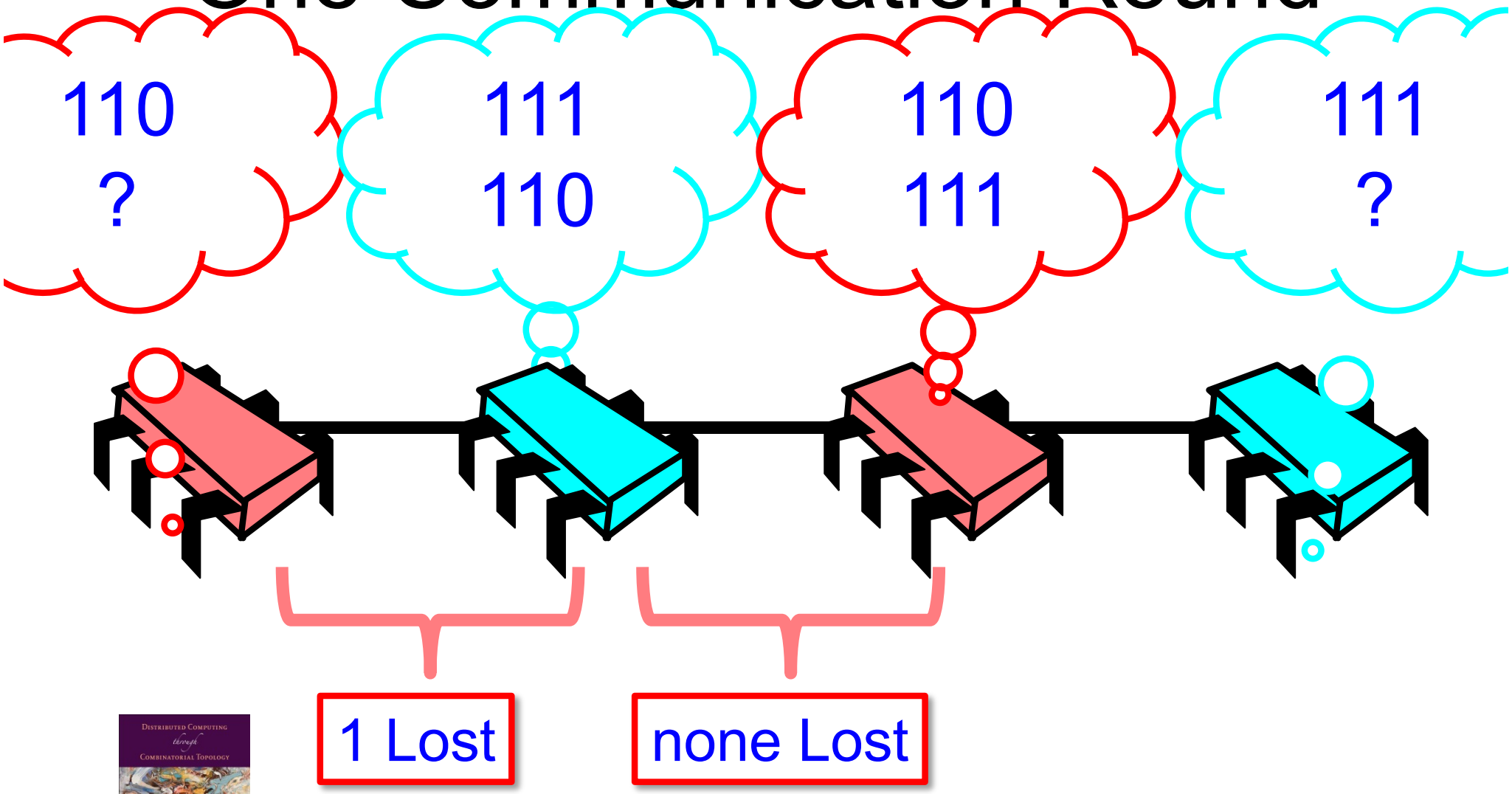


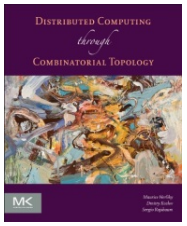


One Communication Round

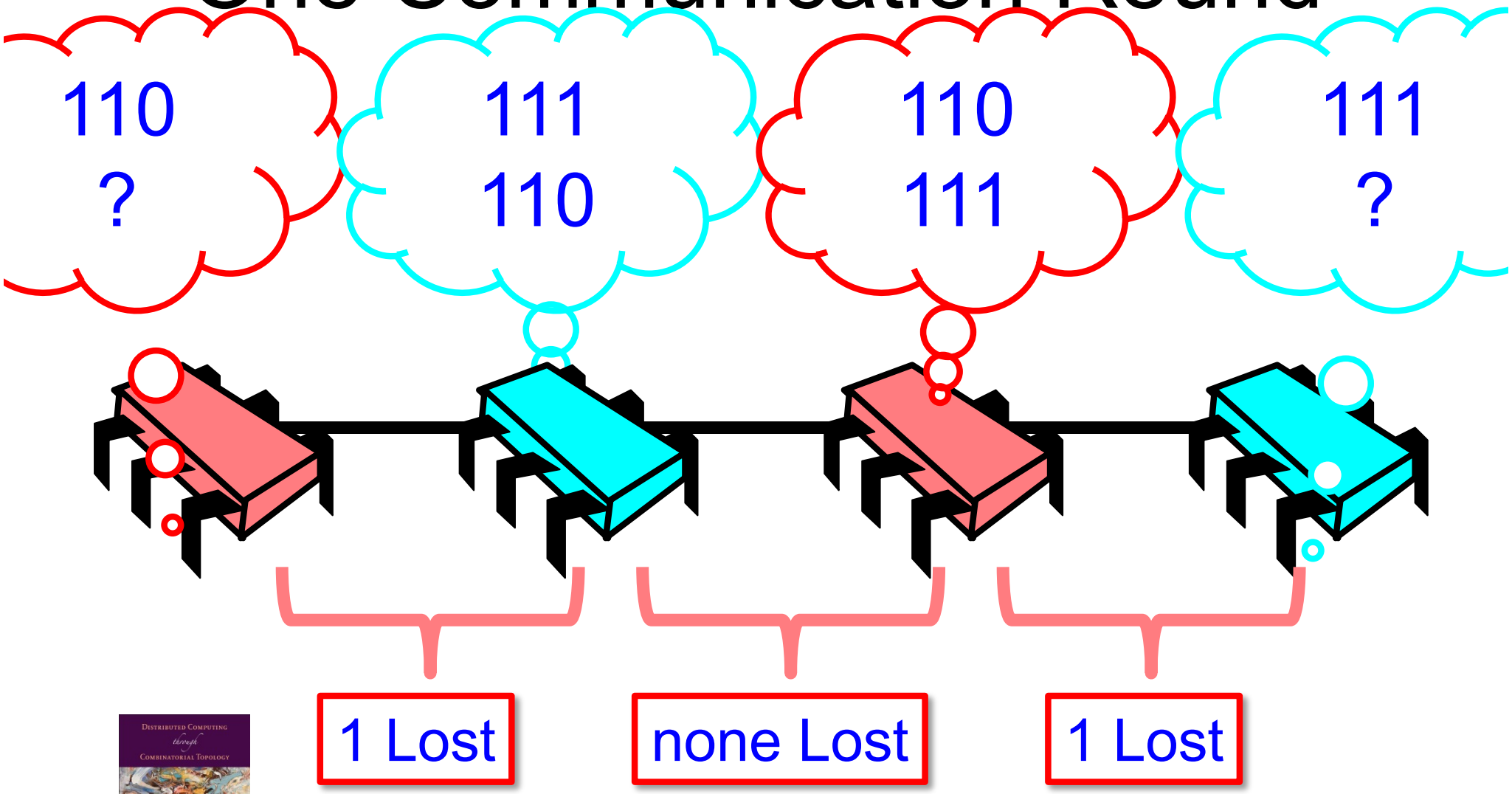


One Communication Round

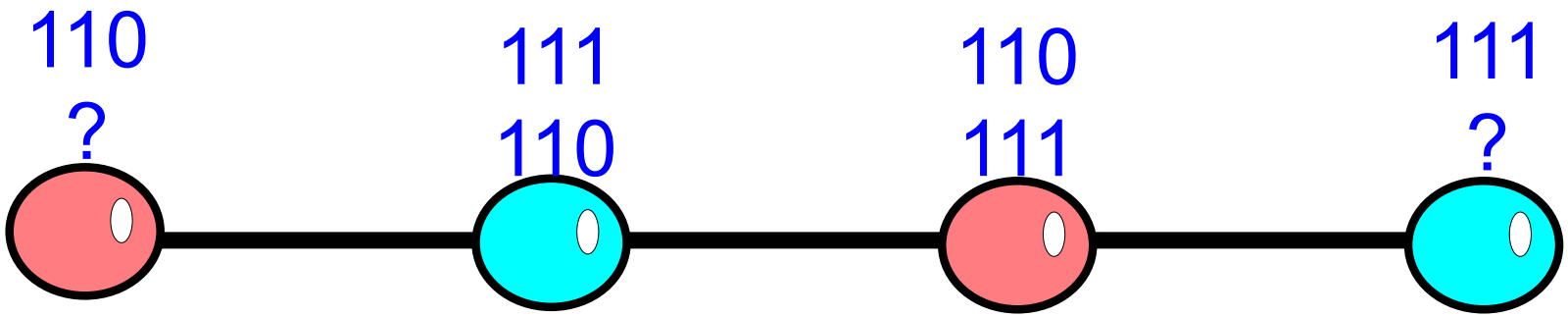


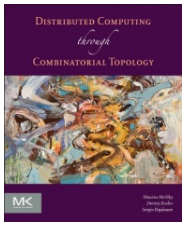


One Communication Round

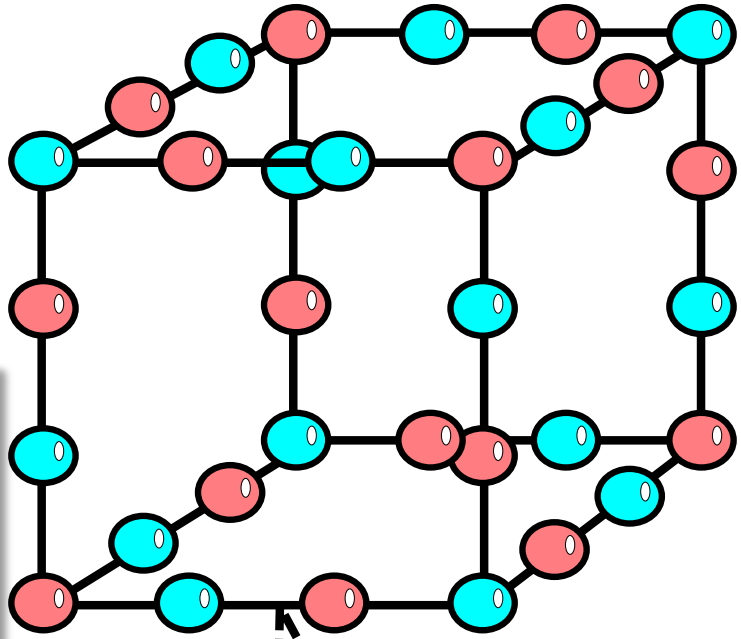


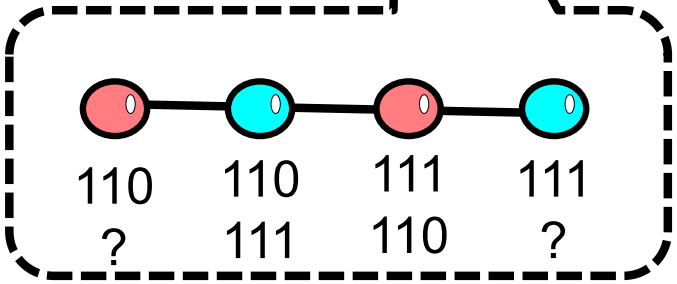
One Communication Round

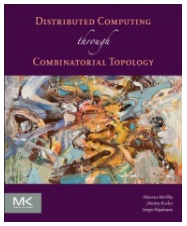




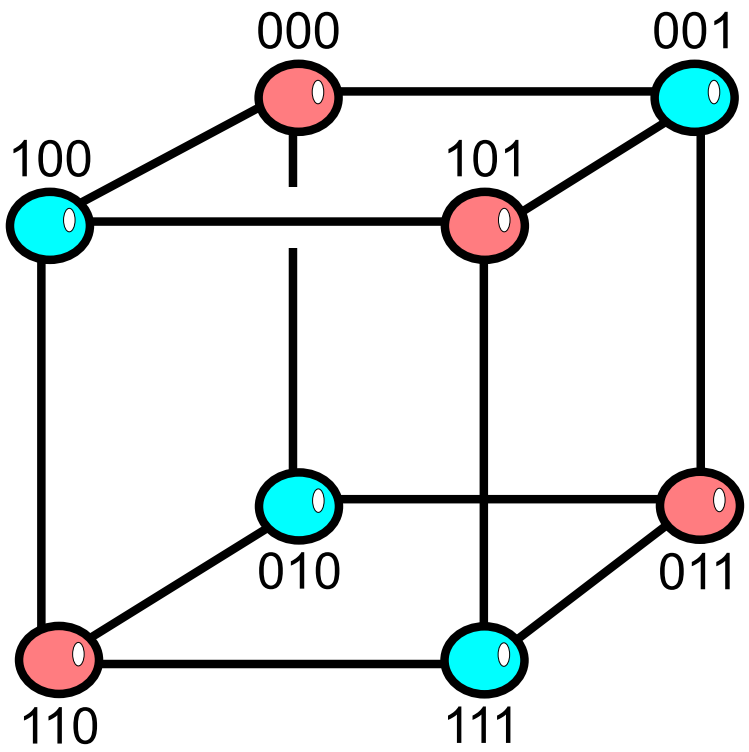
All possible global states after one round unreliable communication

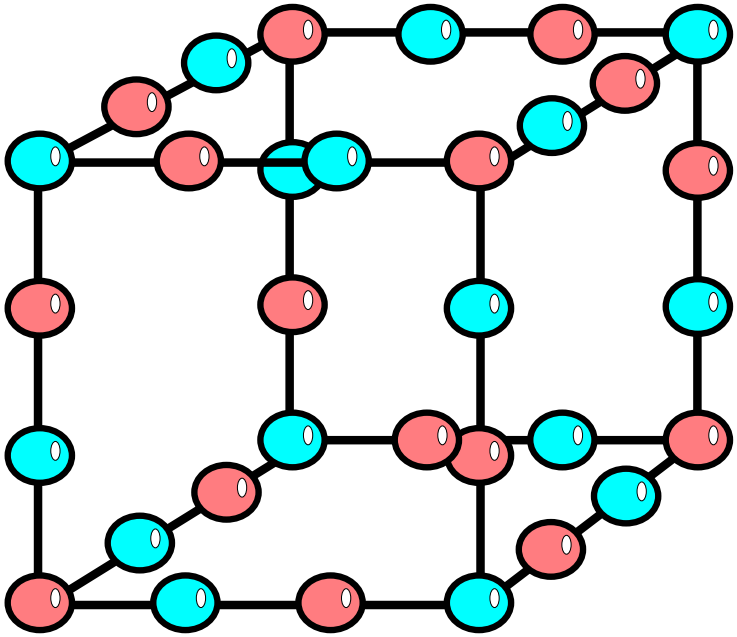


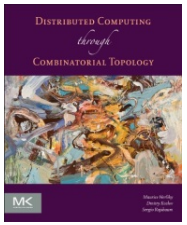


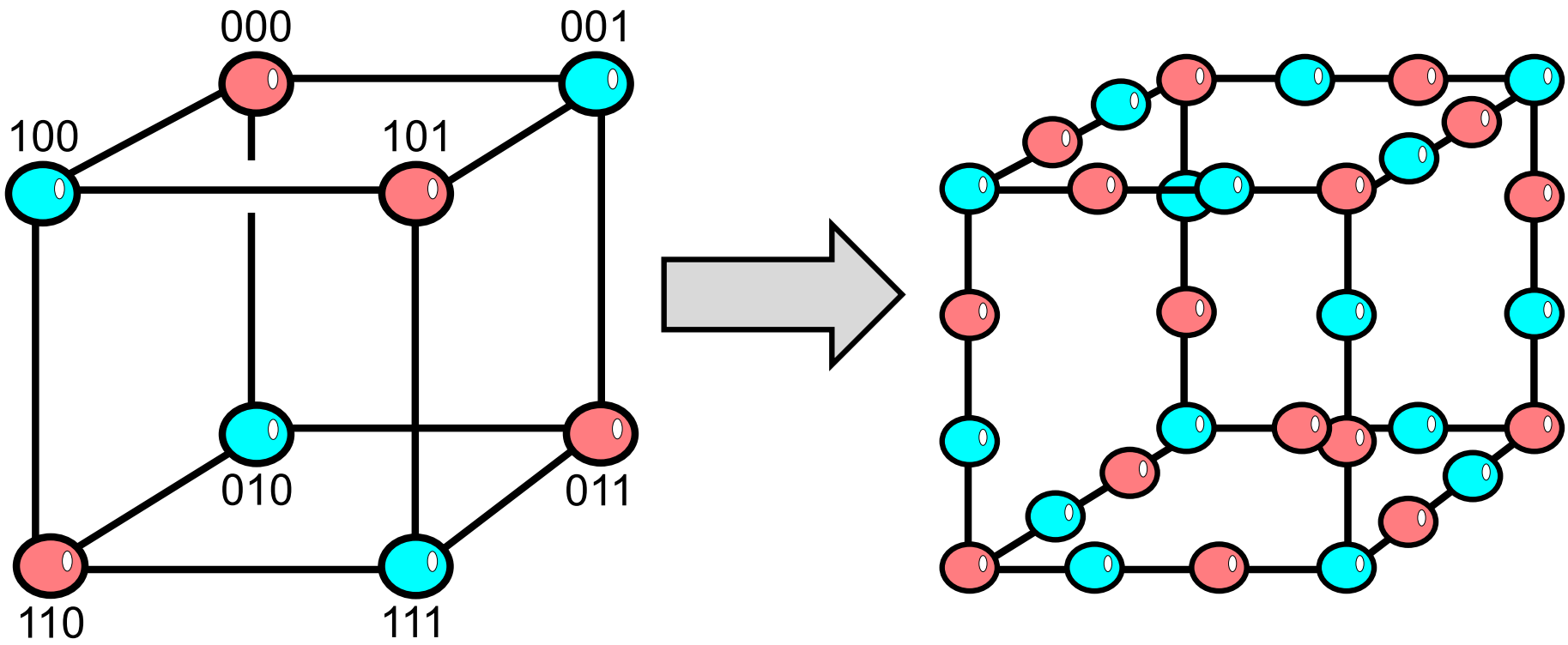


Distributed Computing though Combinatorial Topology


000
001
100
101
010
011
110
111

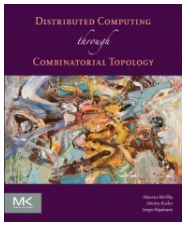



Distributed Computing
through
Combinatorial Topology

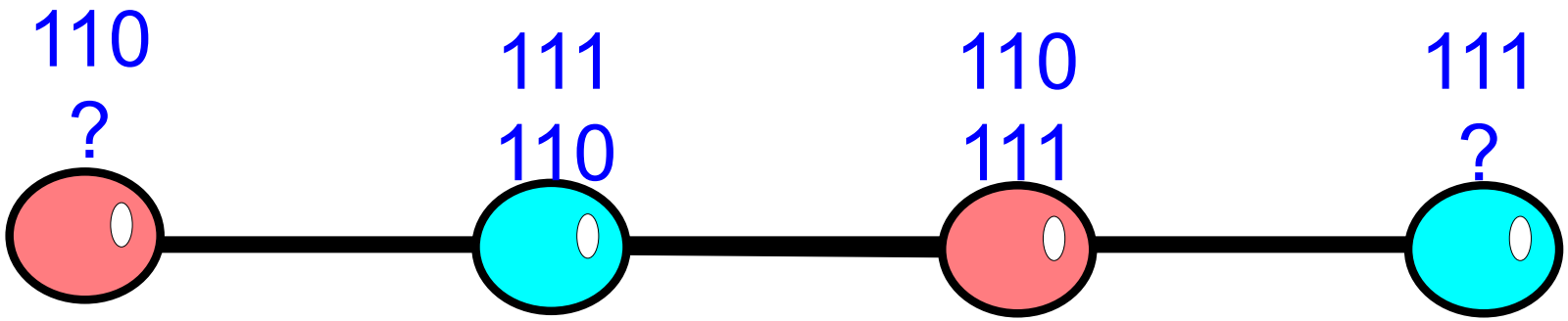


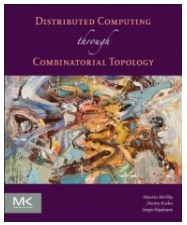
Informally ….

Unreliable communication does not change “topology” of global states

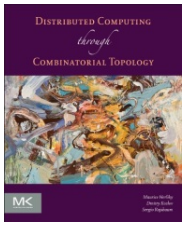


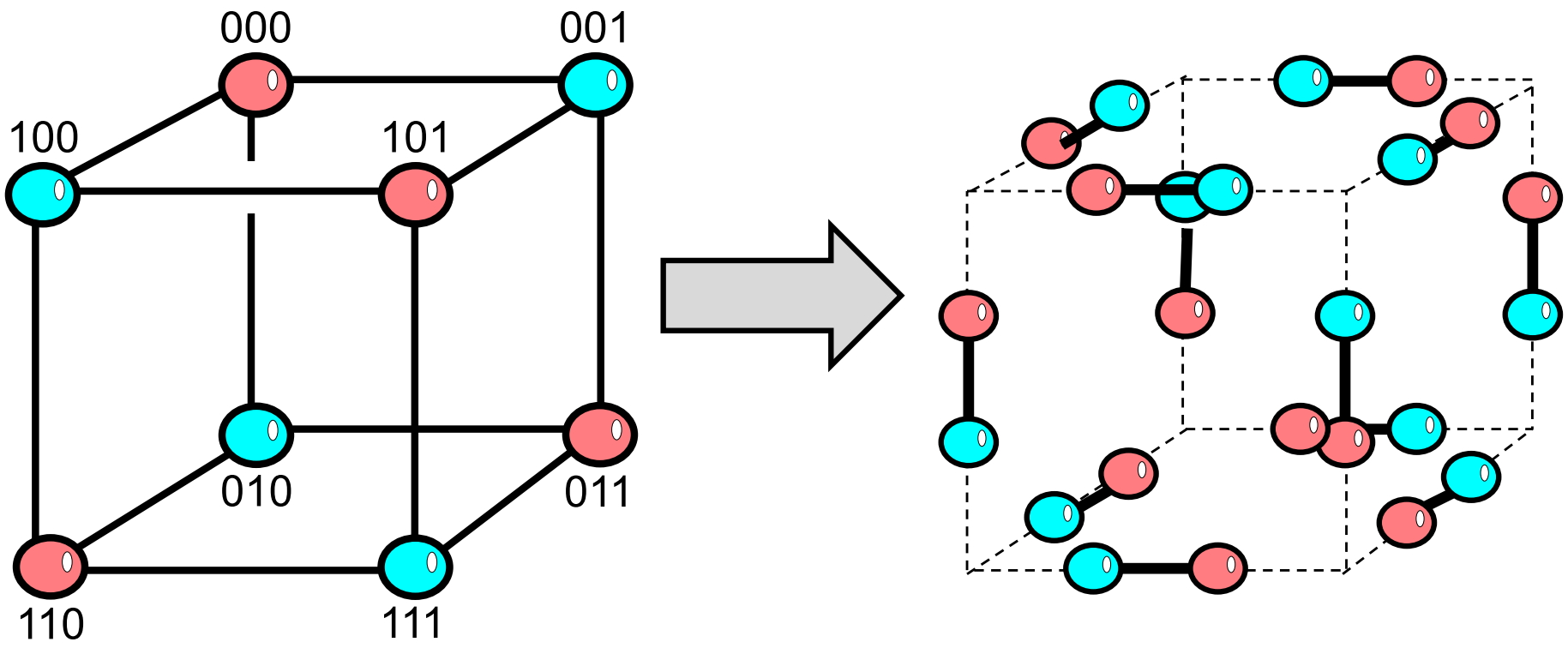
Reliable Communication?

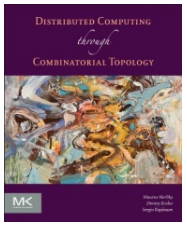




Reliable Communication?

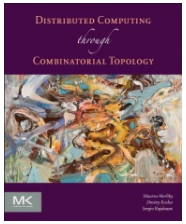




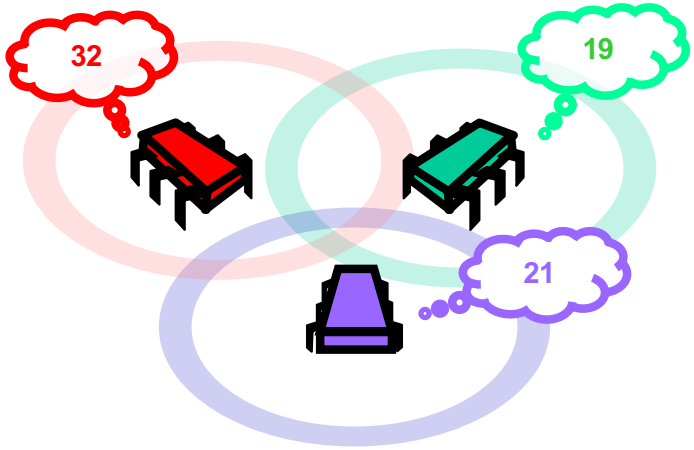


Distributed Computing though
Combinatorial Topology

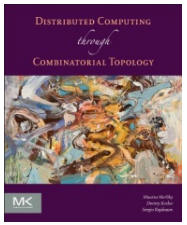
Tasks



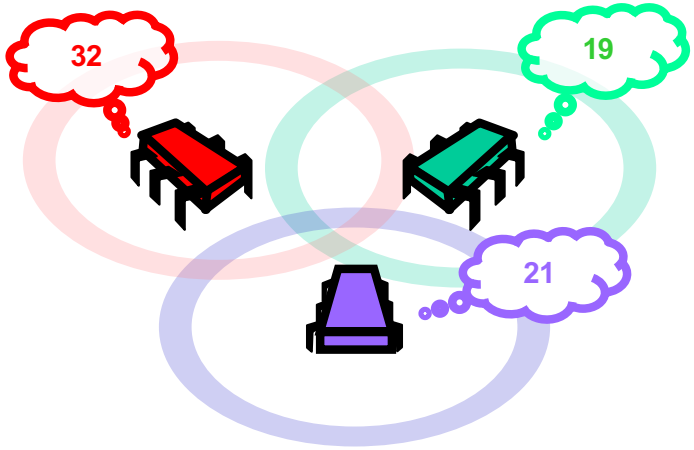
Tasks



Possible set of *input* values

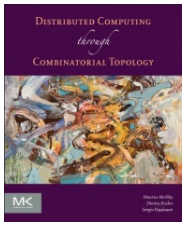


Tasks

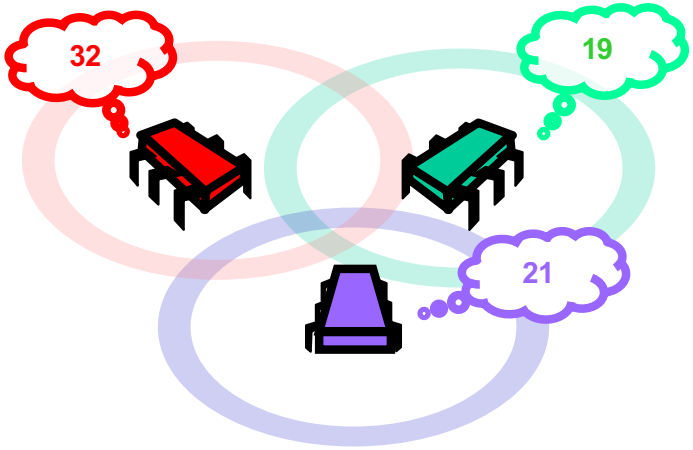


Possible set of *input* values

Finite computation



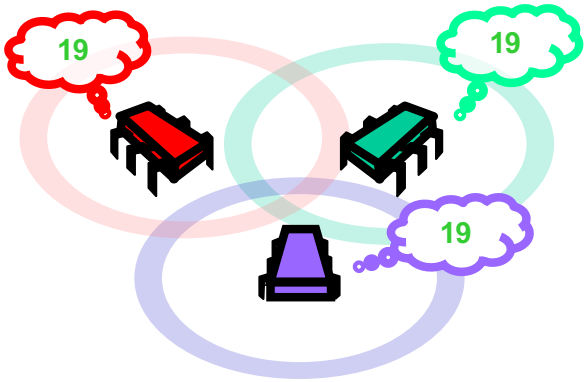
Tasks

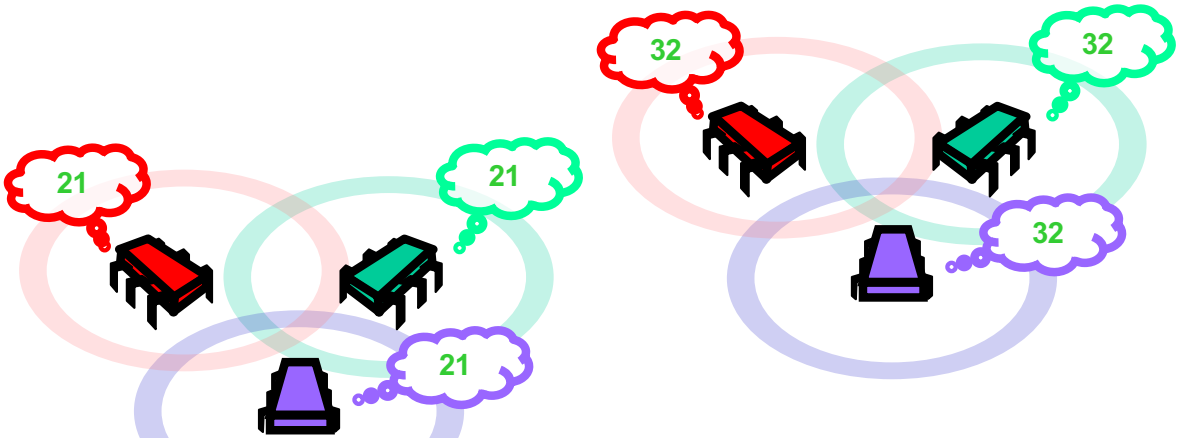


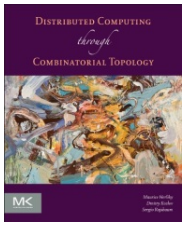
Possible set of *input* values

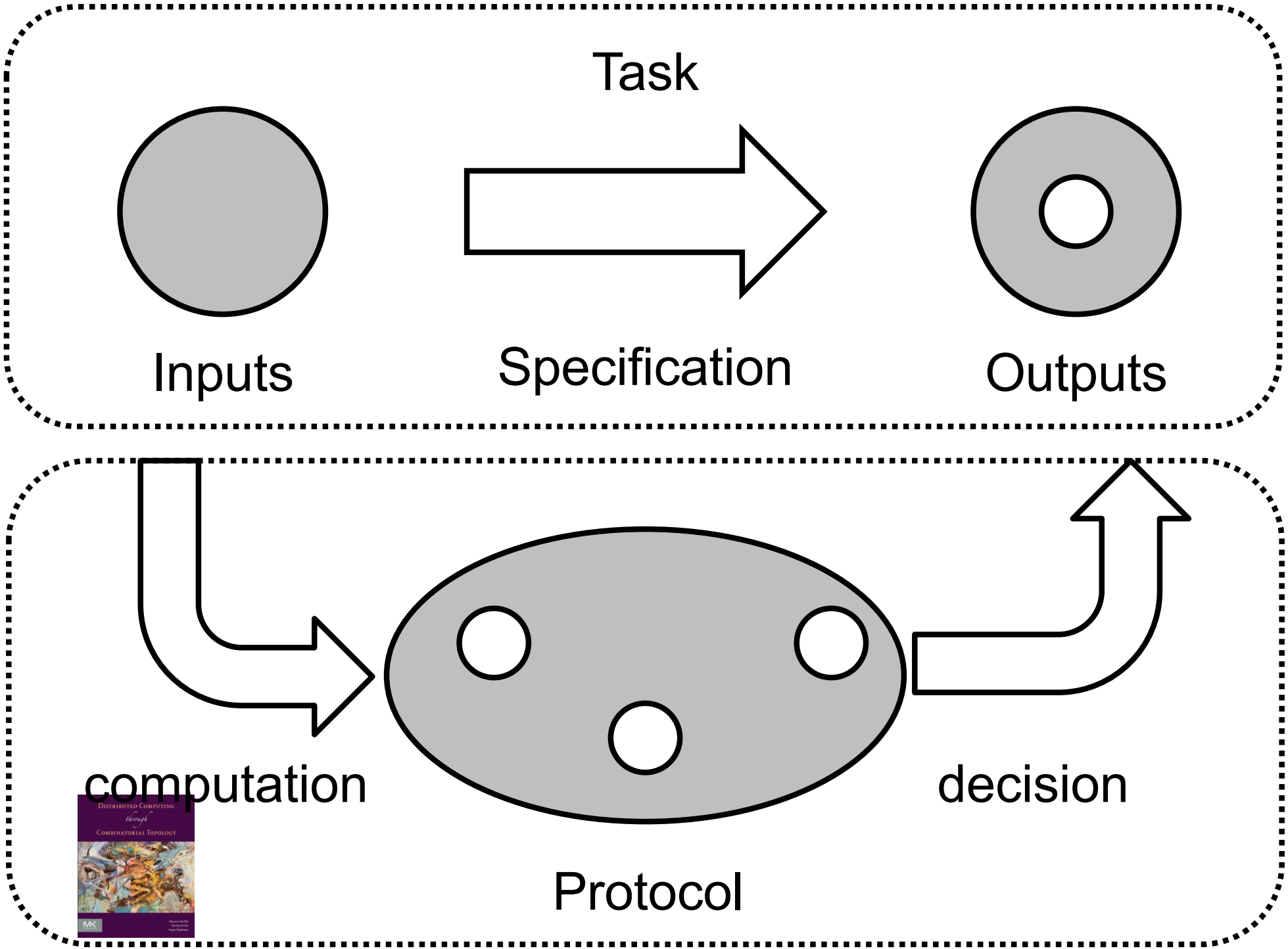
Finite computation

Possible set of *output* values








Task
Inputs
Specification
Outputs
computation
decision
Protocol

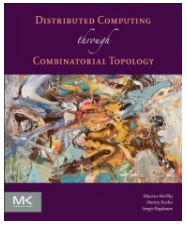
Road Map

Distributed Computing

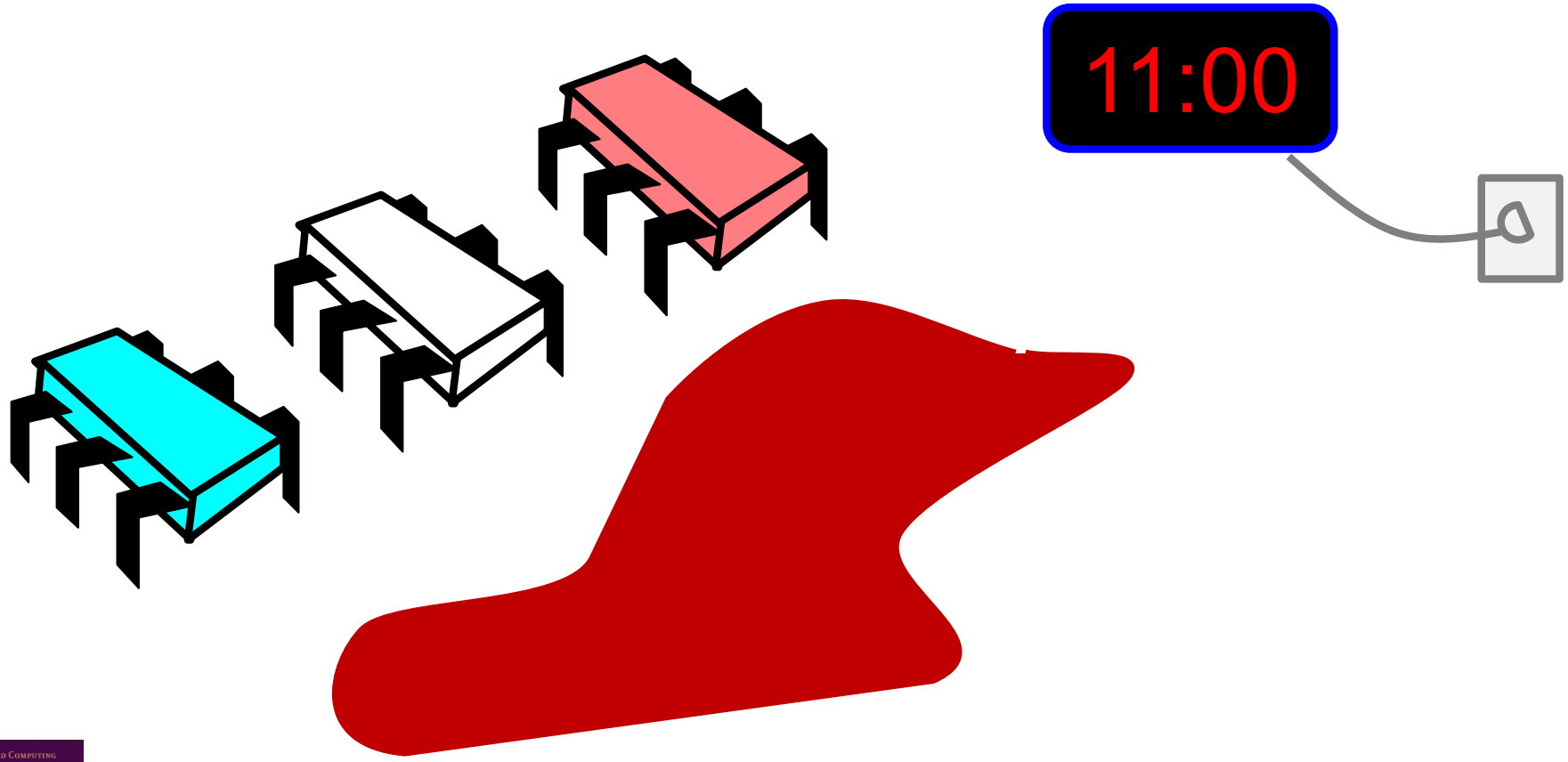
Two Classic Distributed Problems

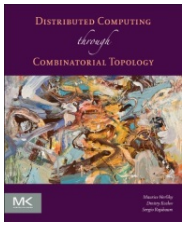
The Muddy Children

Coordinated Attack

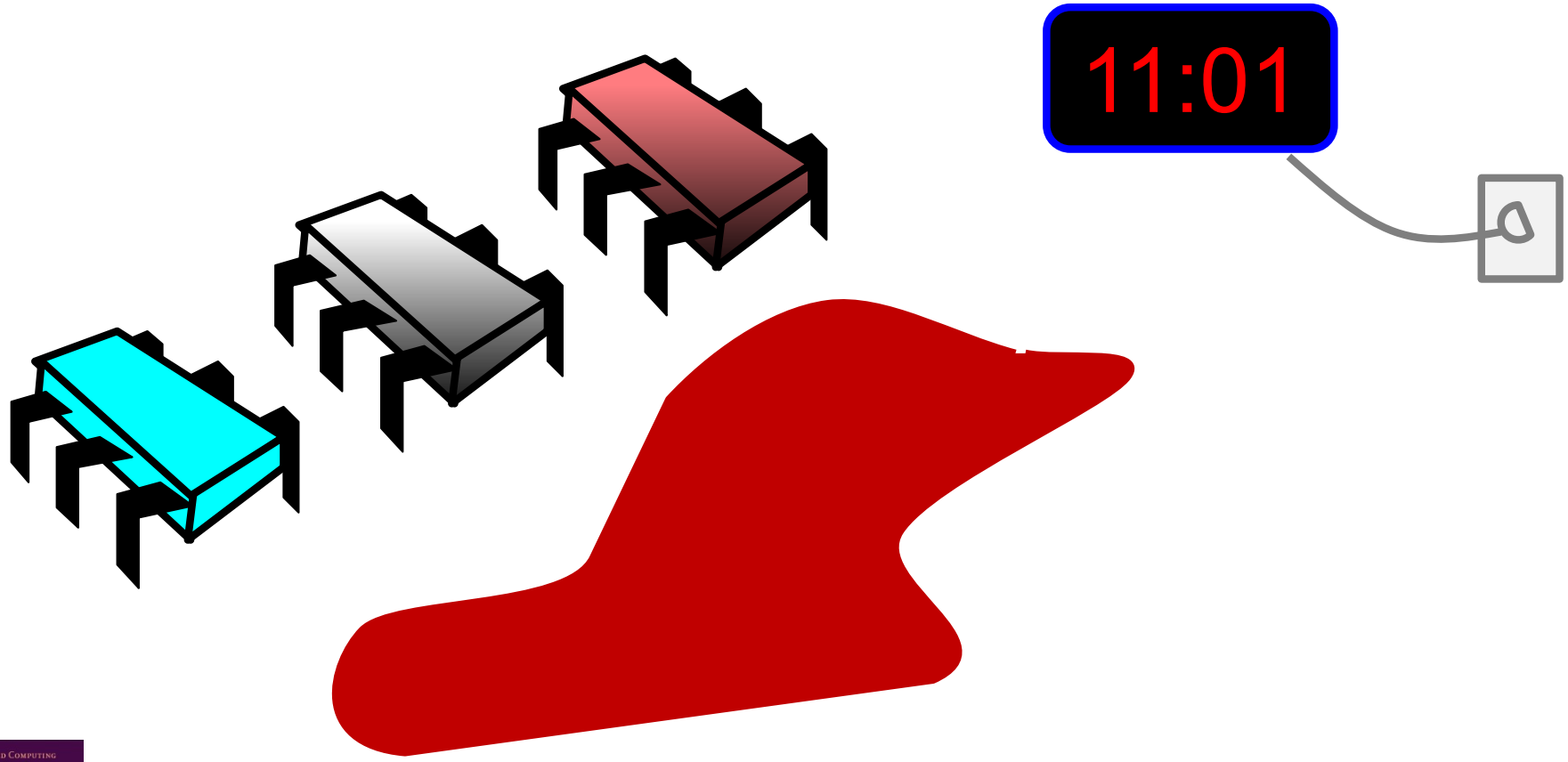


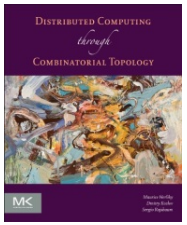
Muddy Children



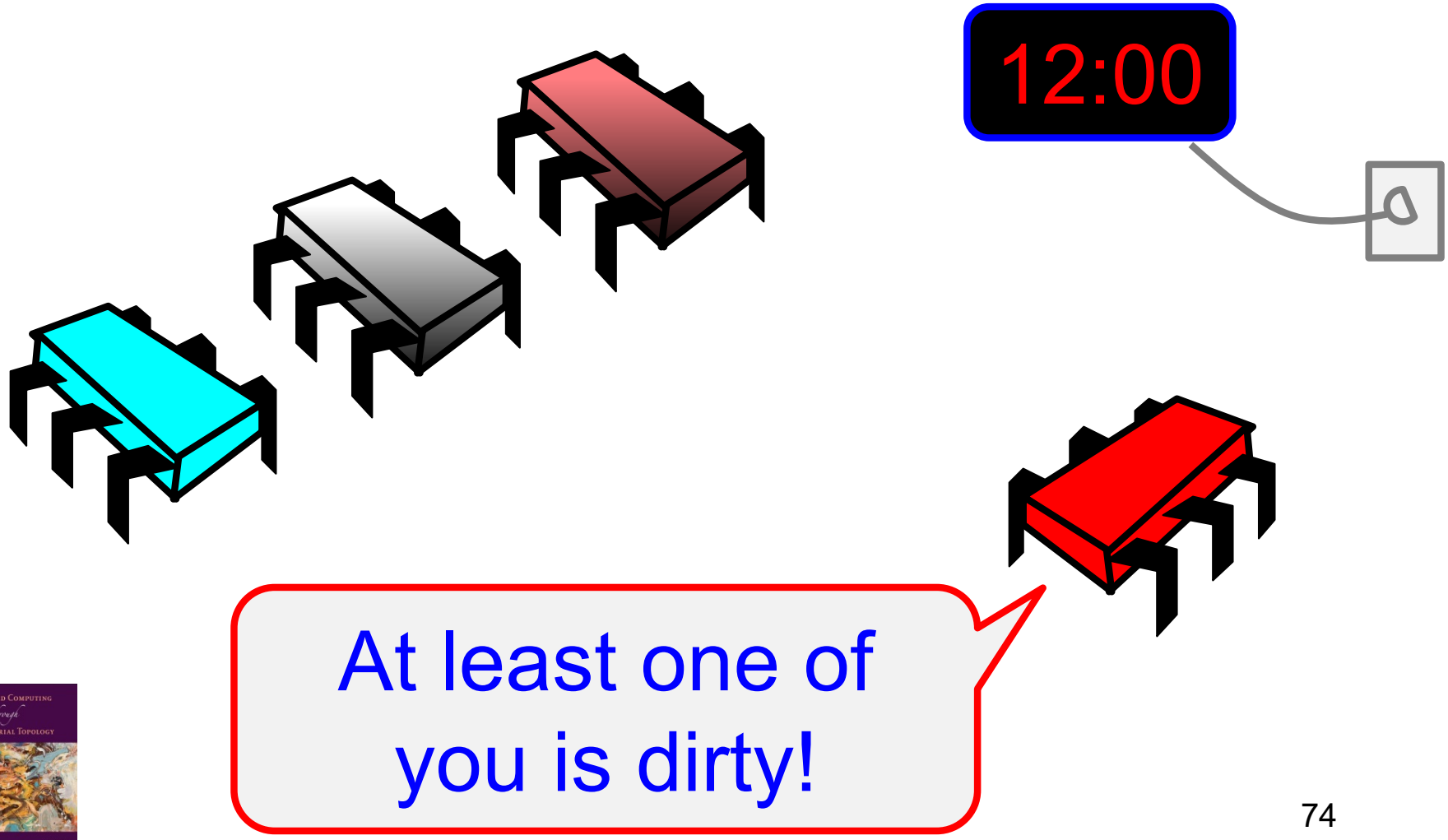


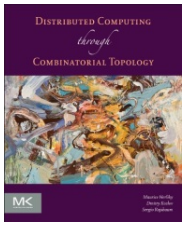
Muddy Children



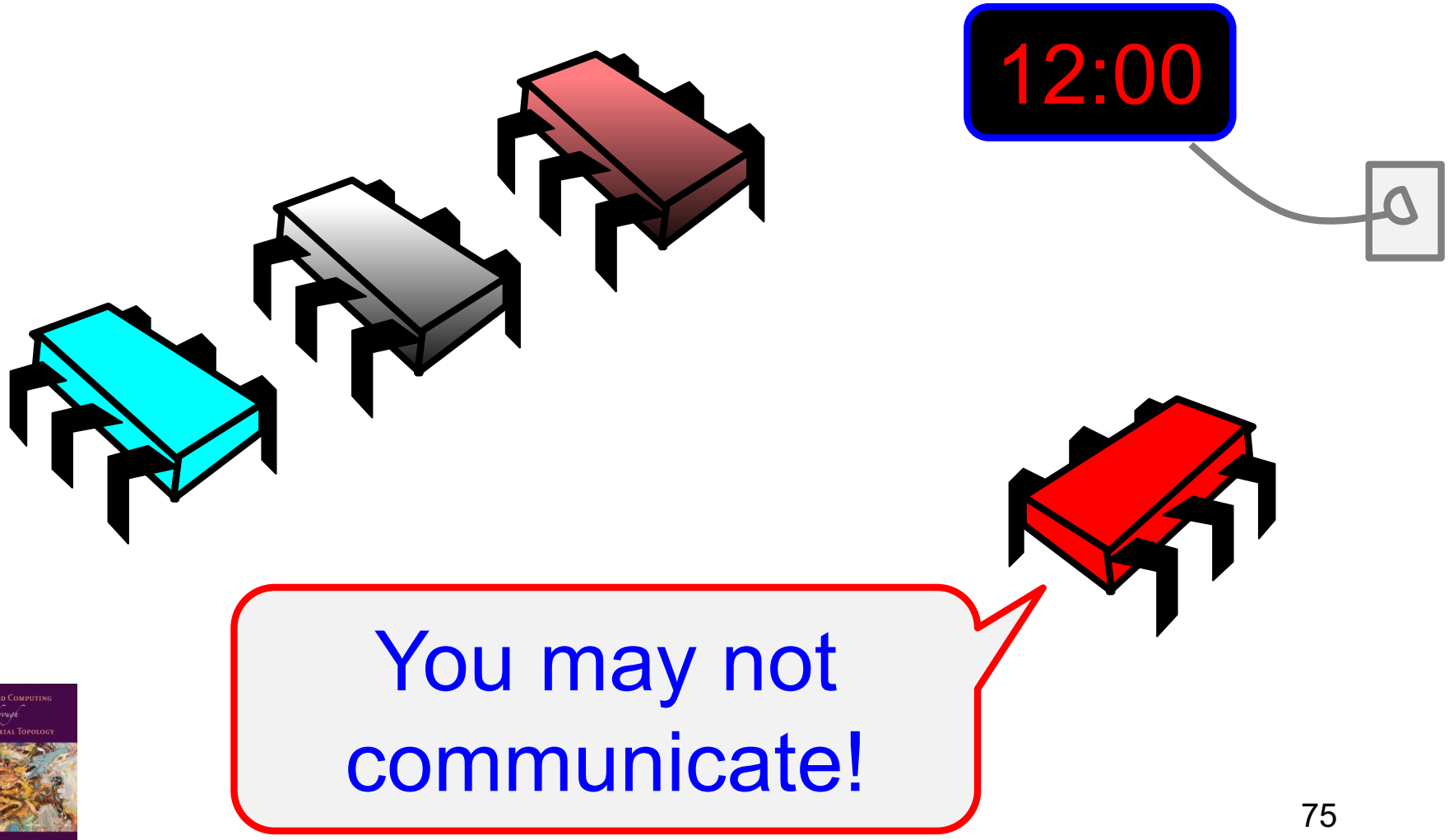


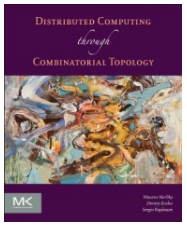
Muddy Children



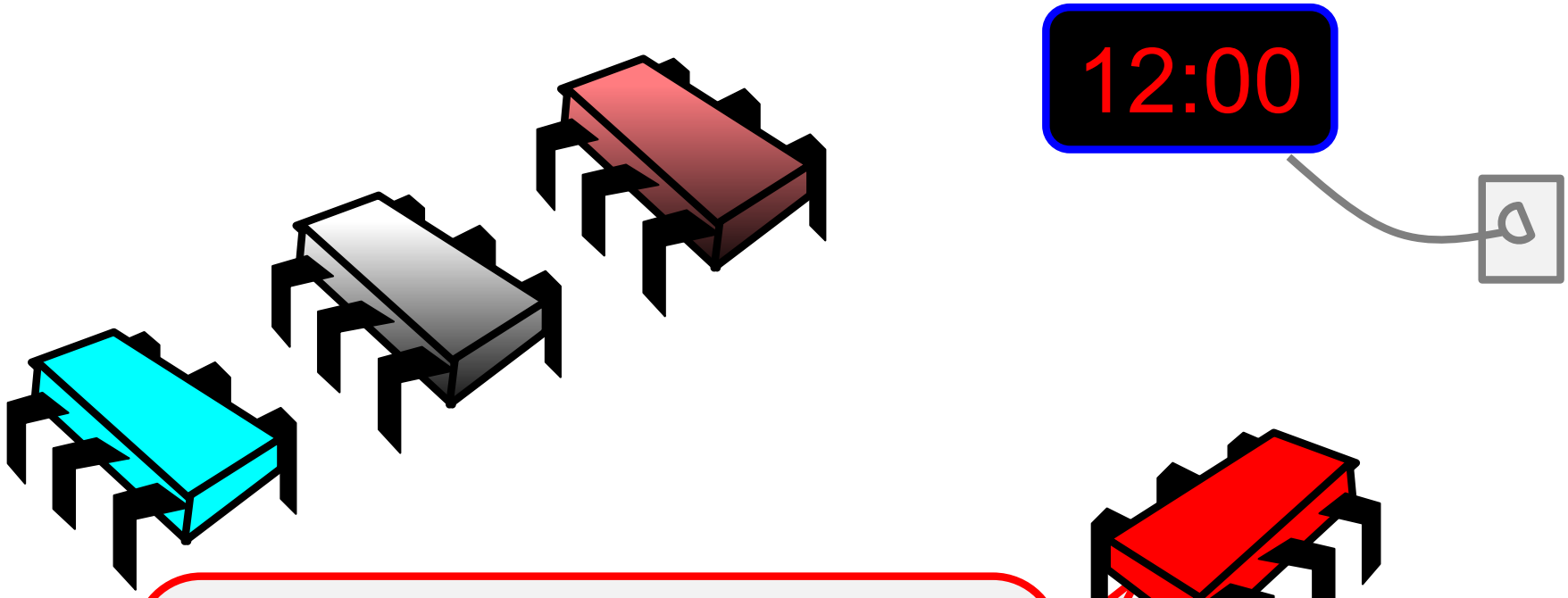


Muddy Children

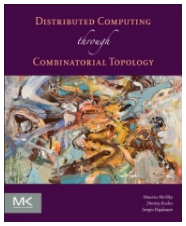




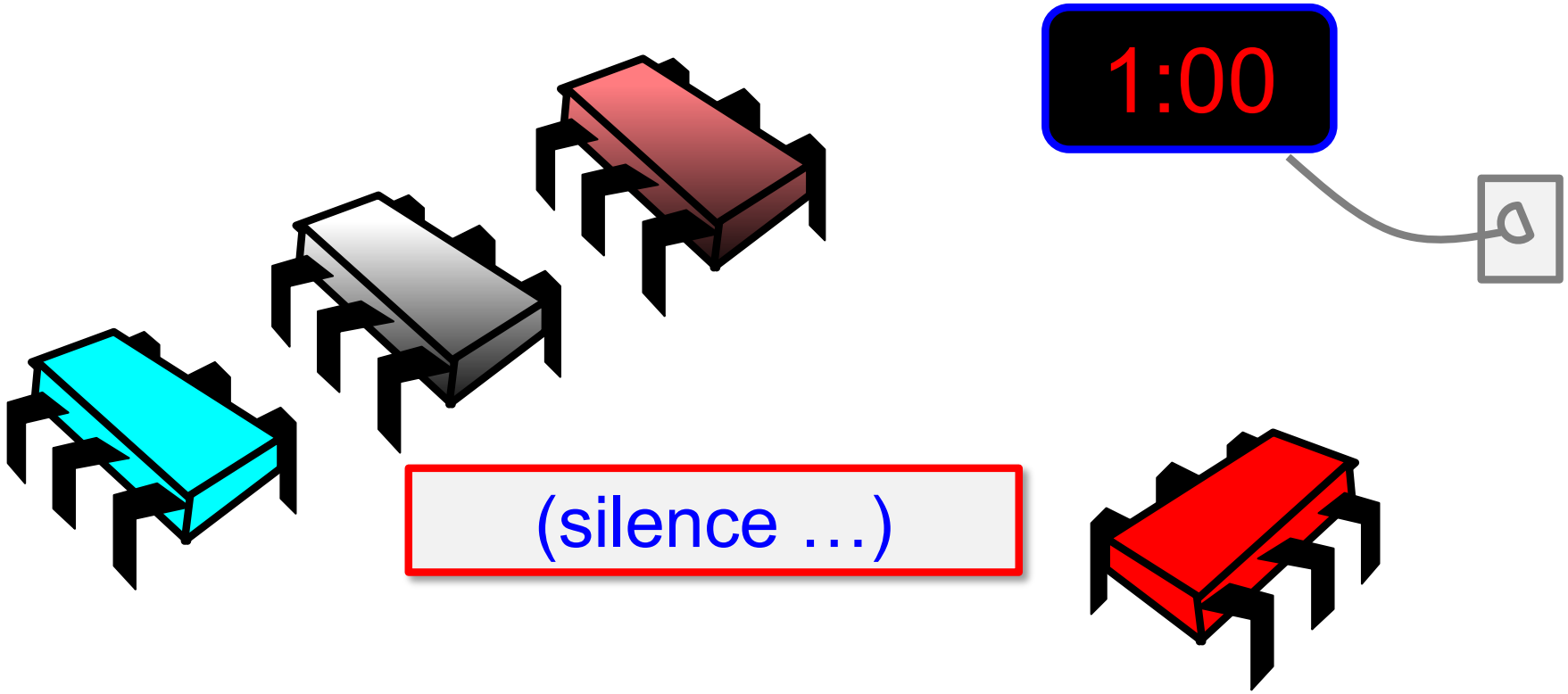
Muddy Children

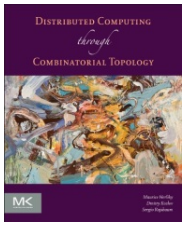


When you realize
you are dirty,
confess on the hour!

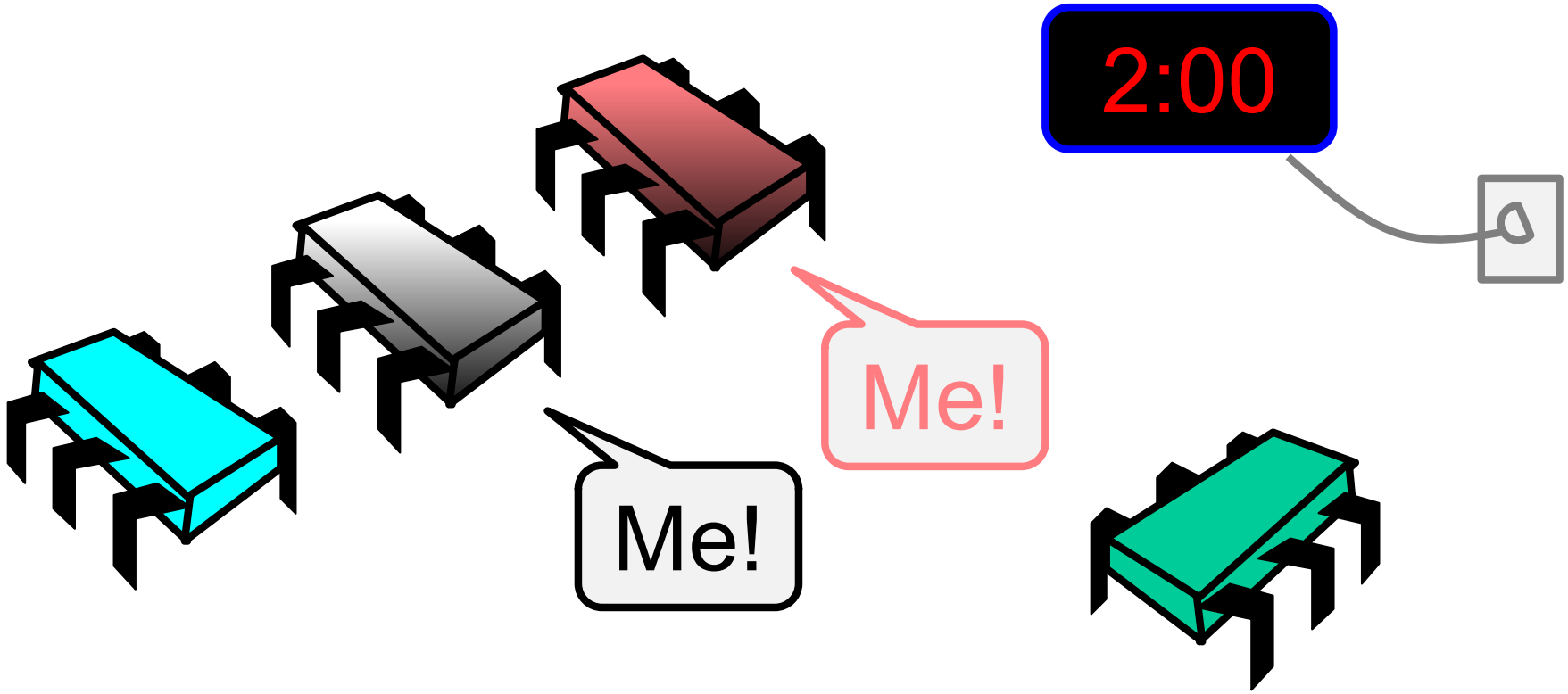


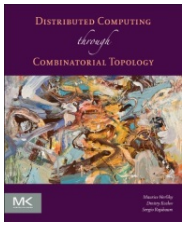
Muddy Children



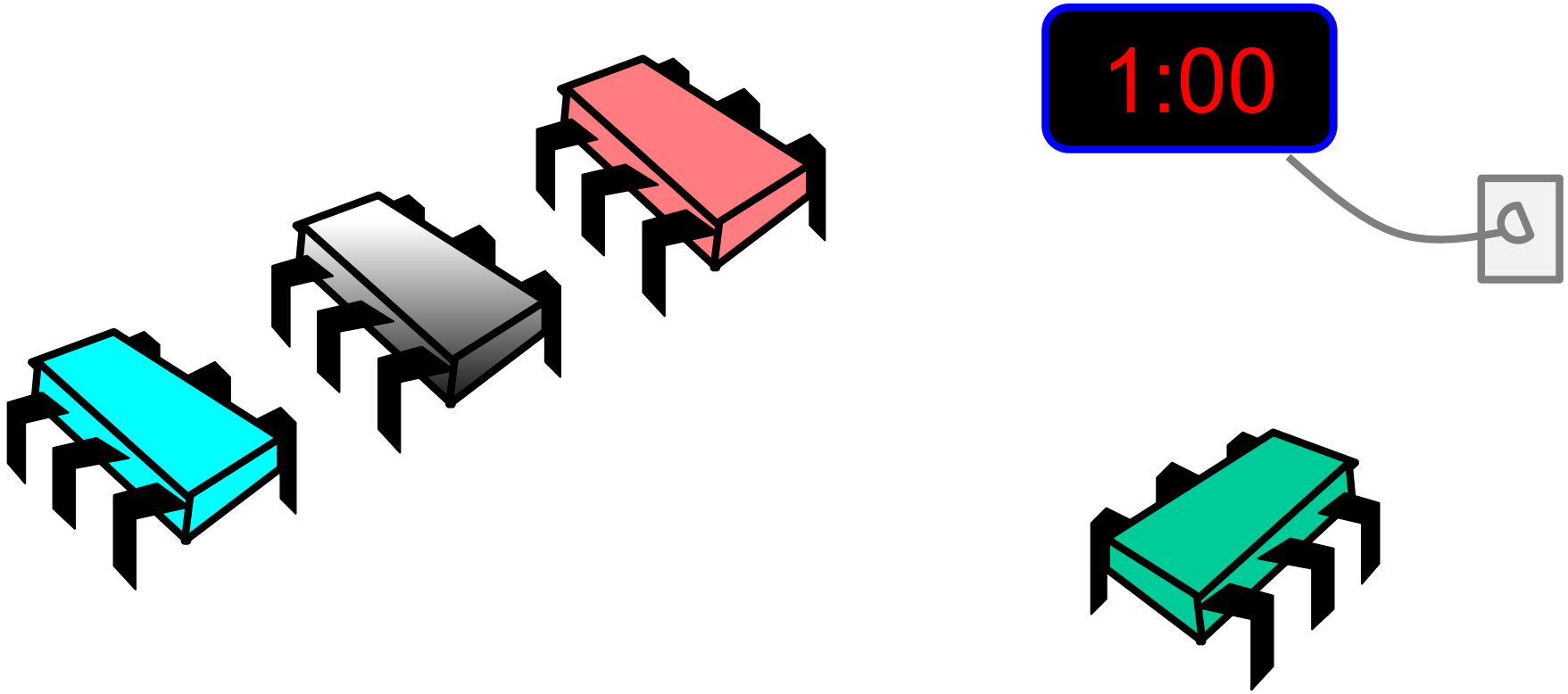


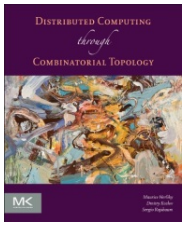
Muddy Children



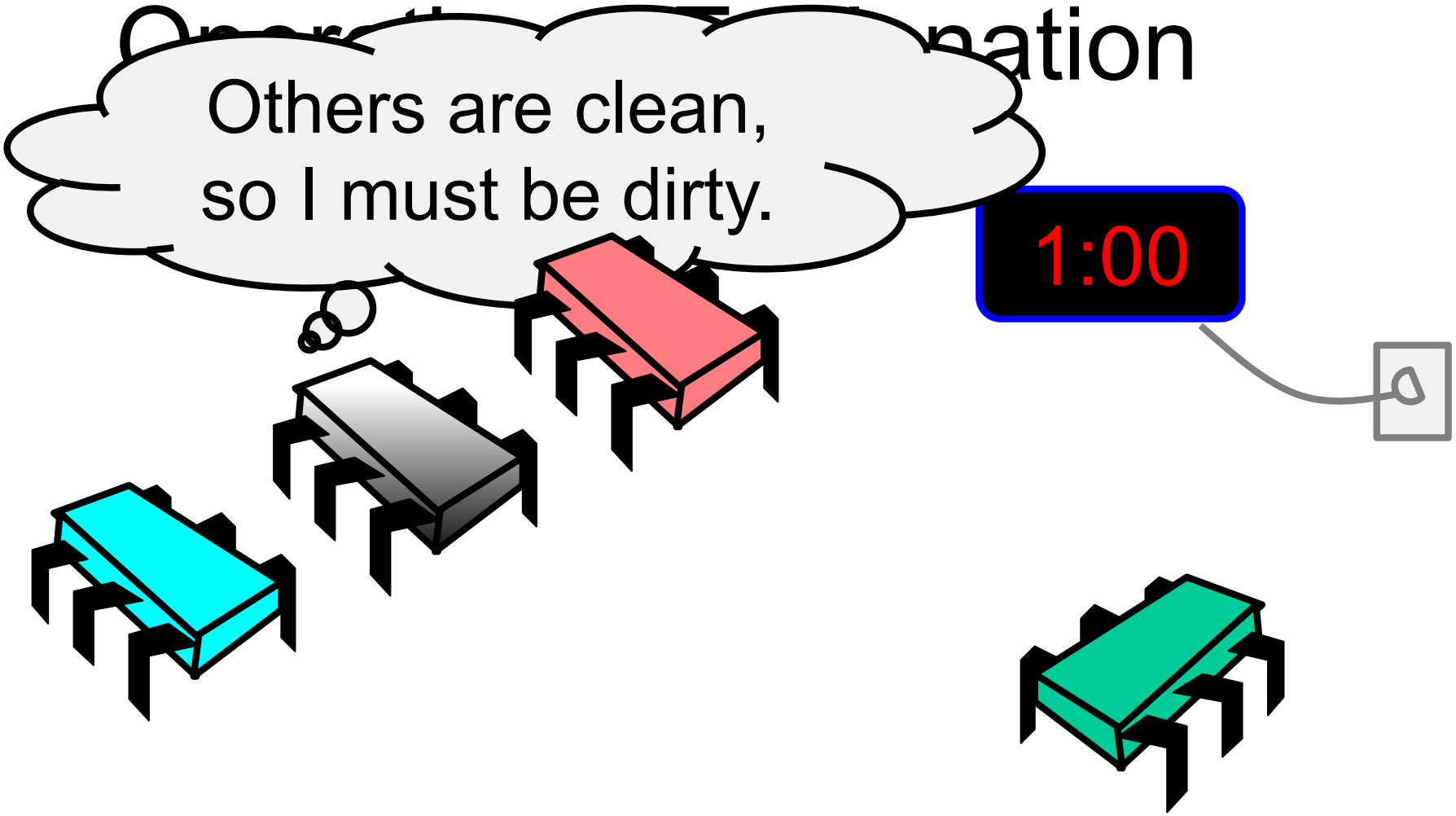


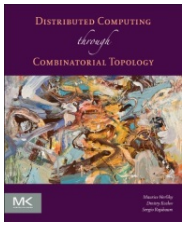
Operational Explanation



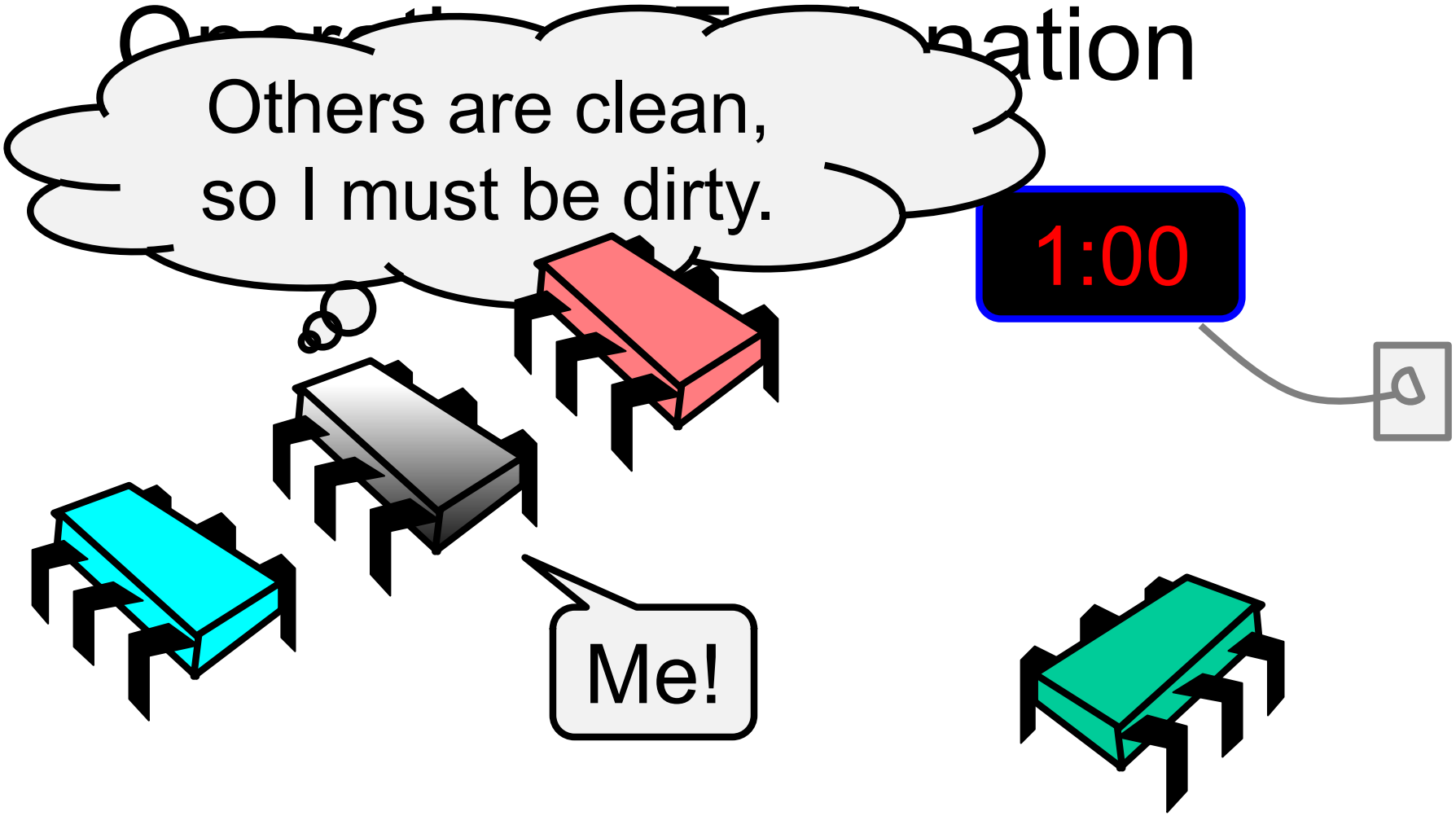


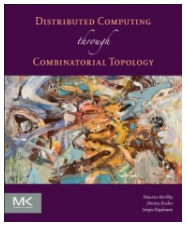
Operational Termination



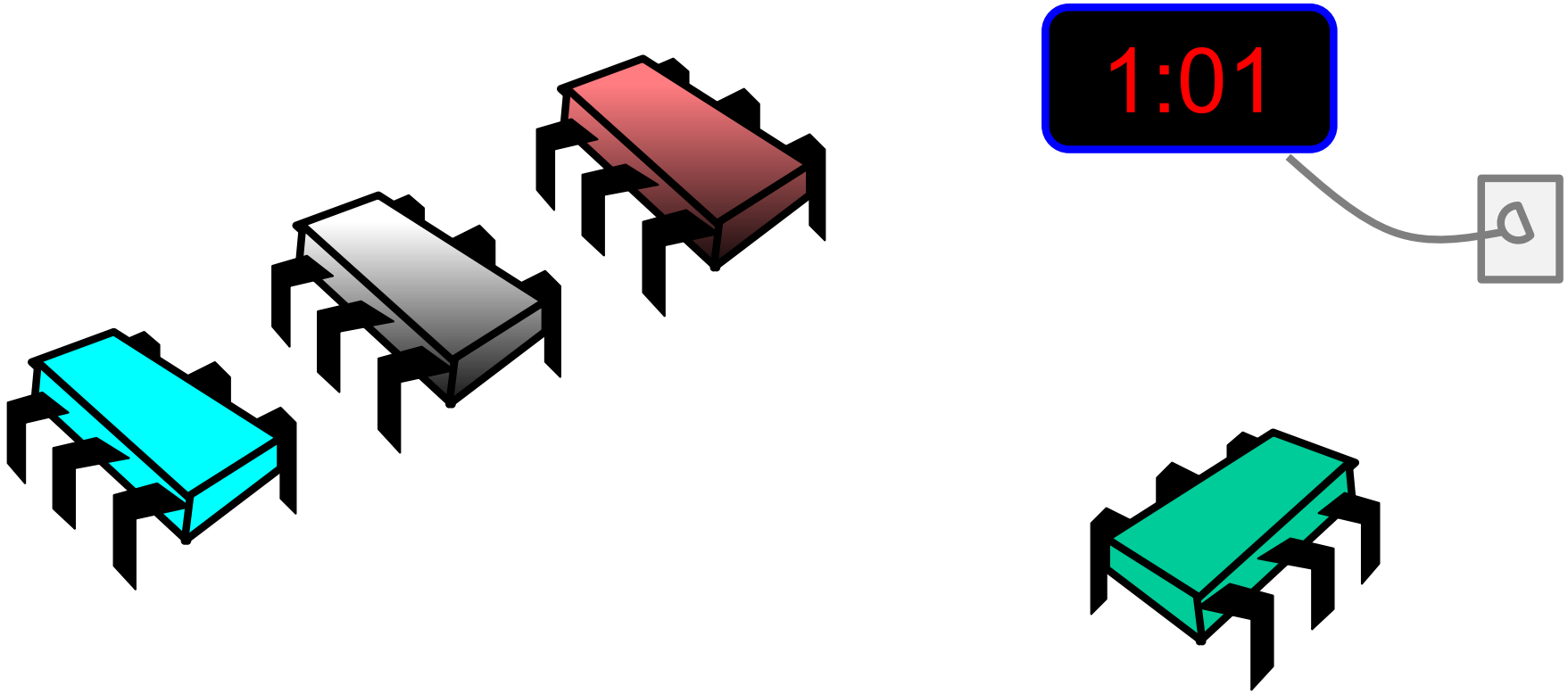


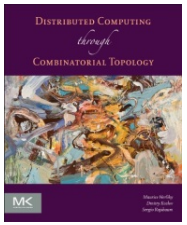
Operational Termination



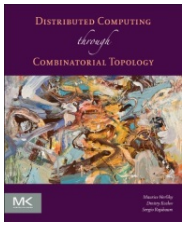


Operational Explanation

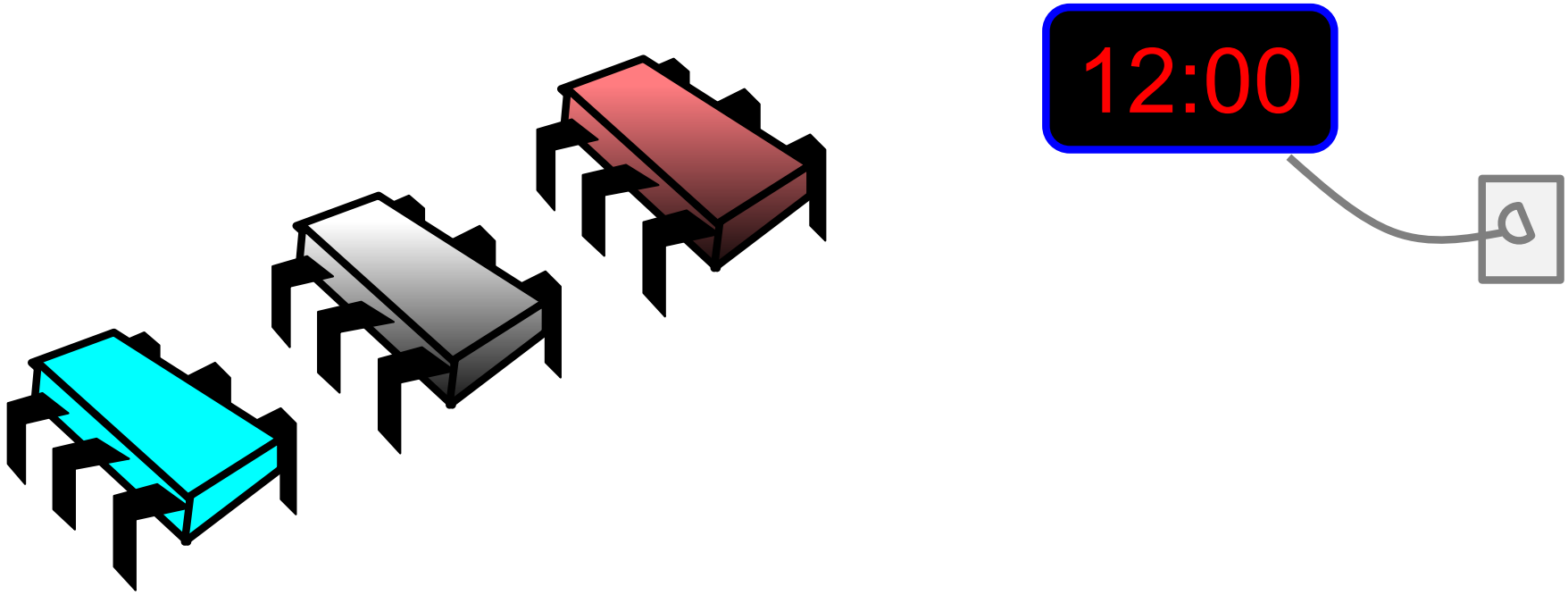


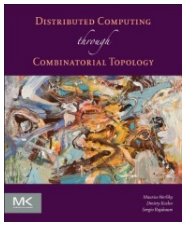


Operational Explanation

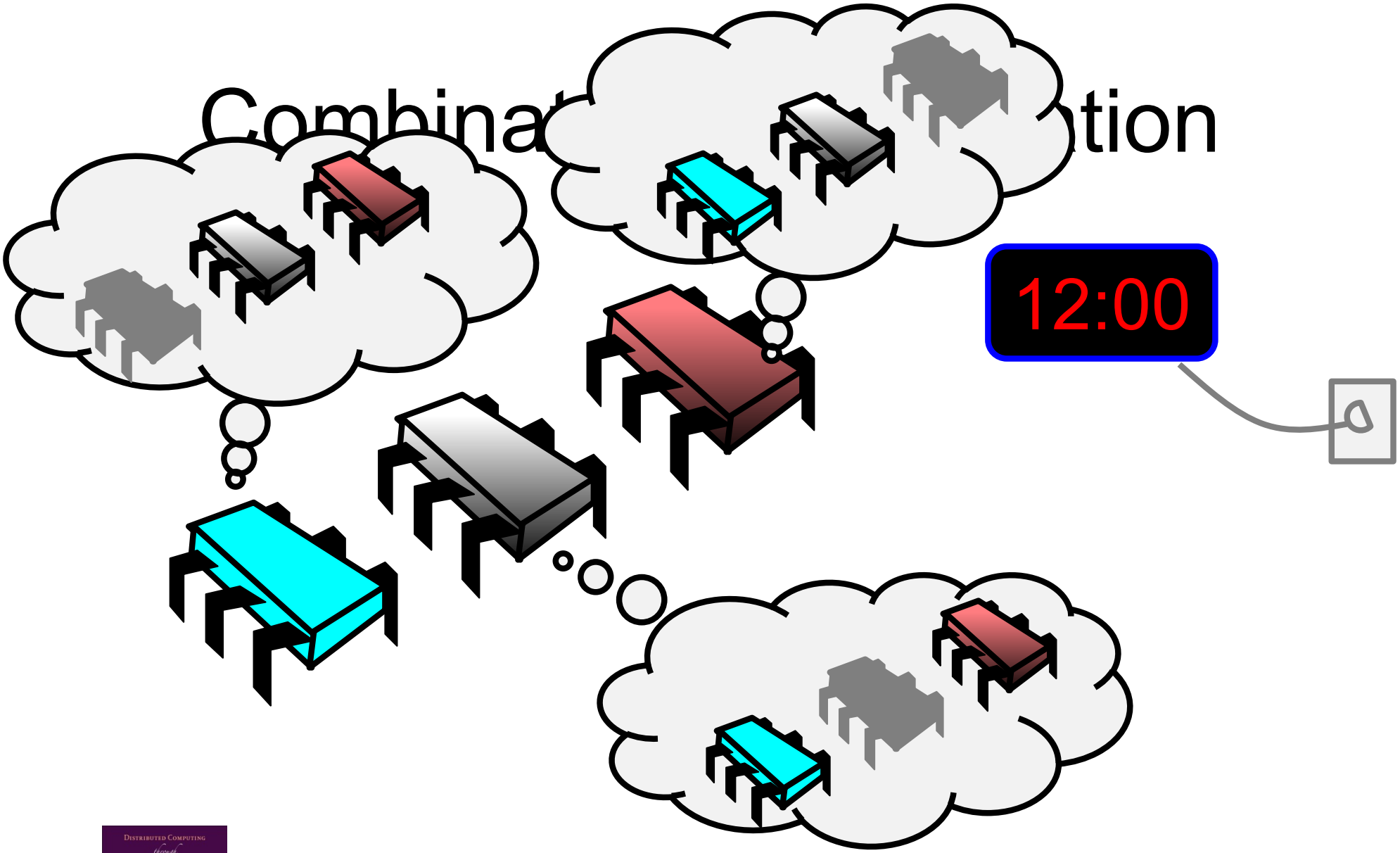


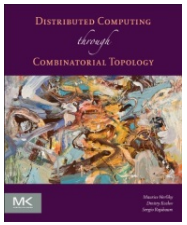
Combinatorial Explanation



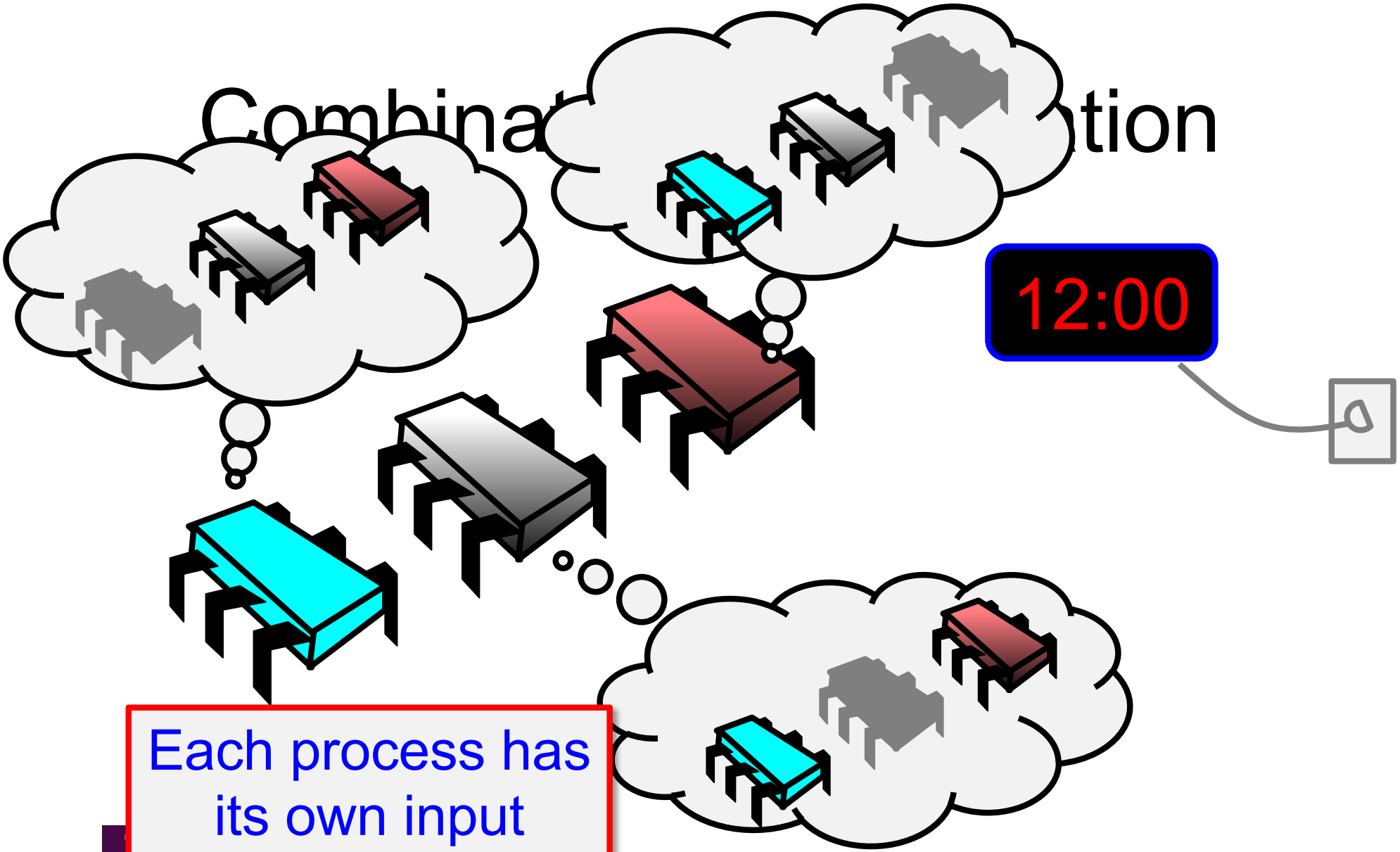


Combinatorial Simulation

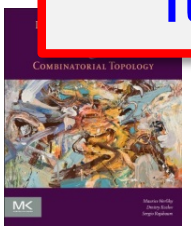




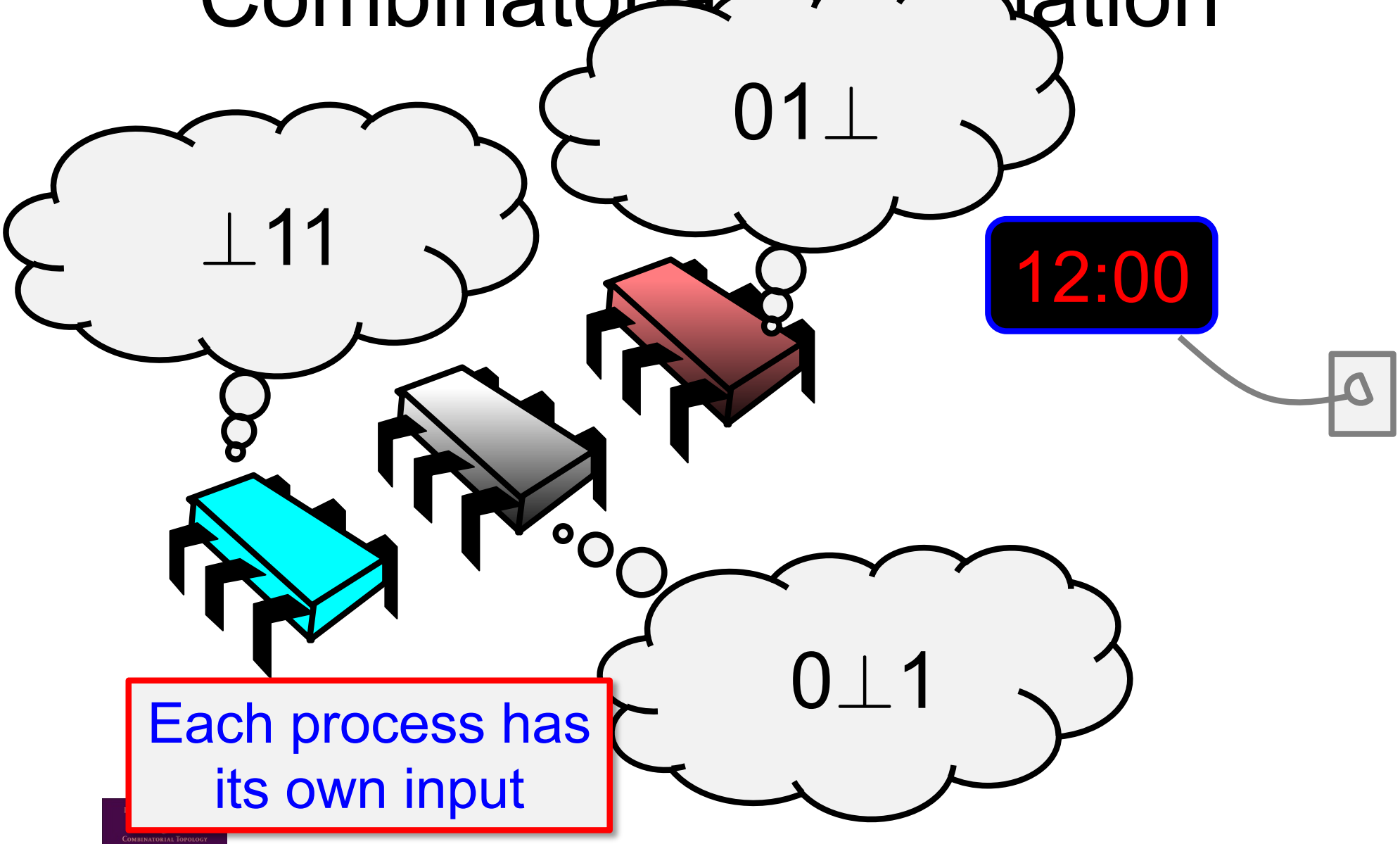
Combinatorial Topology



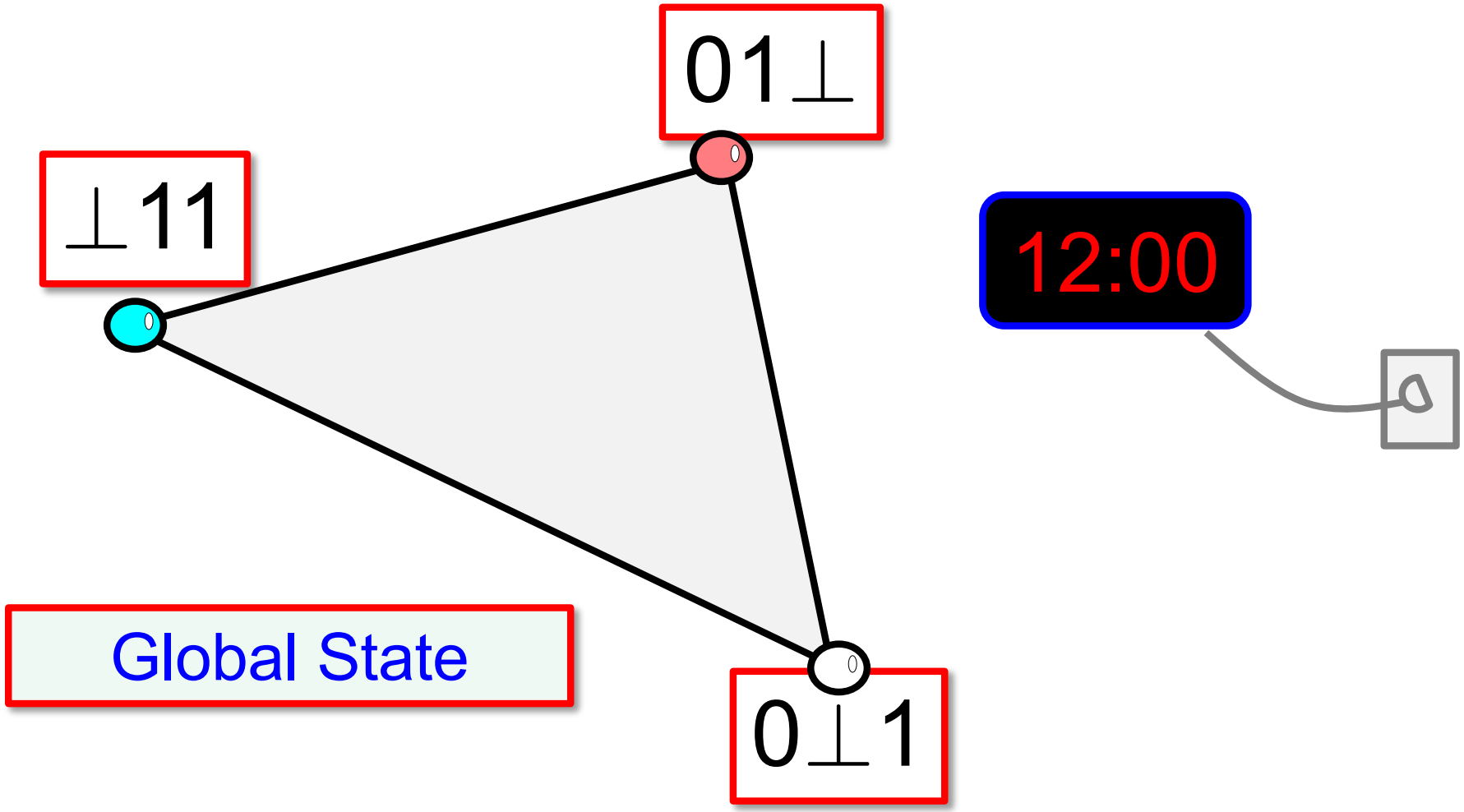
Each process has its own input

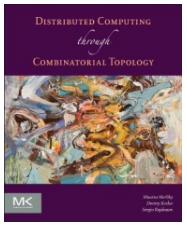


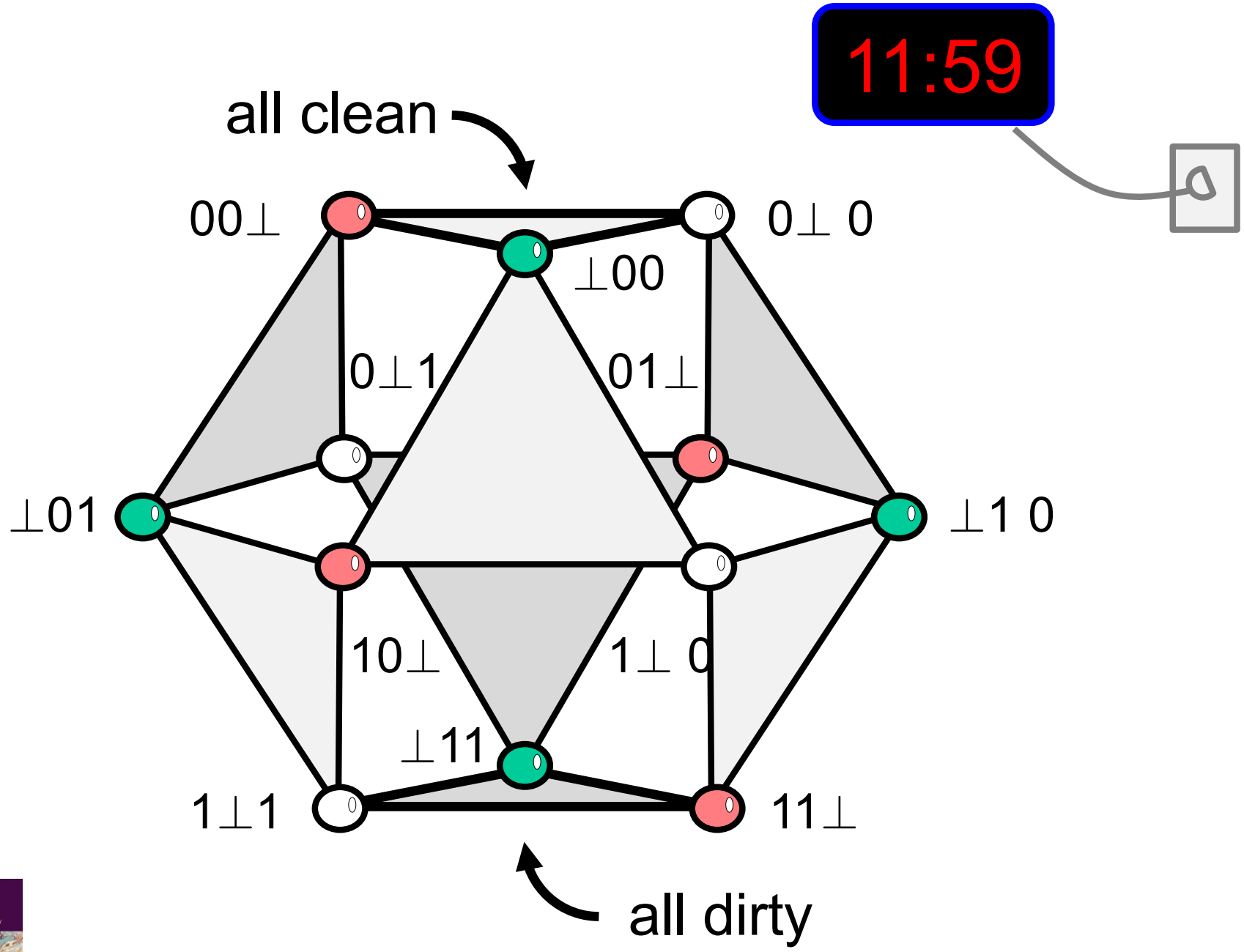
Combinatorial Explanation

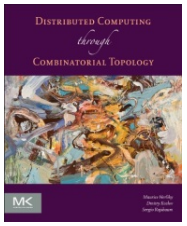


Combinatorial Explanation

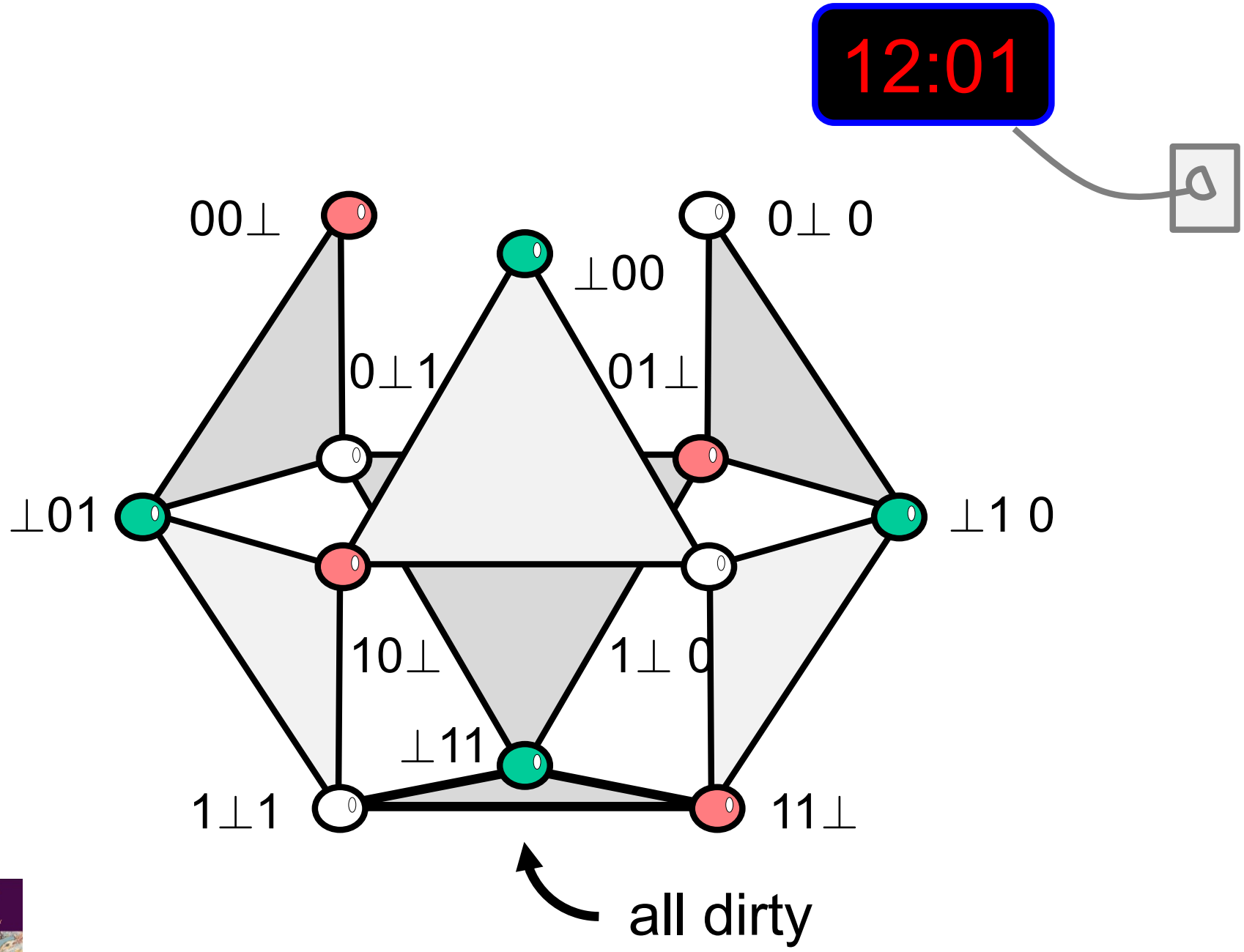


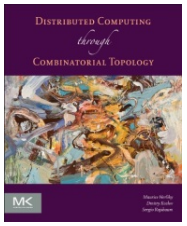




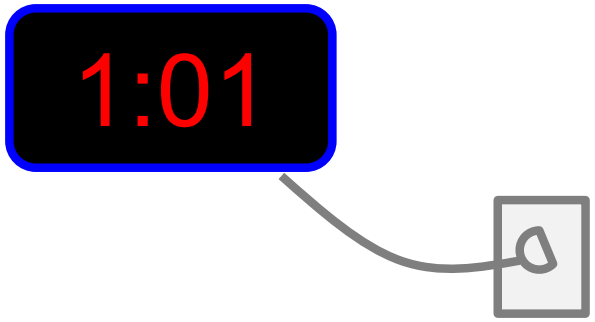


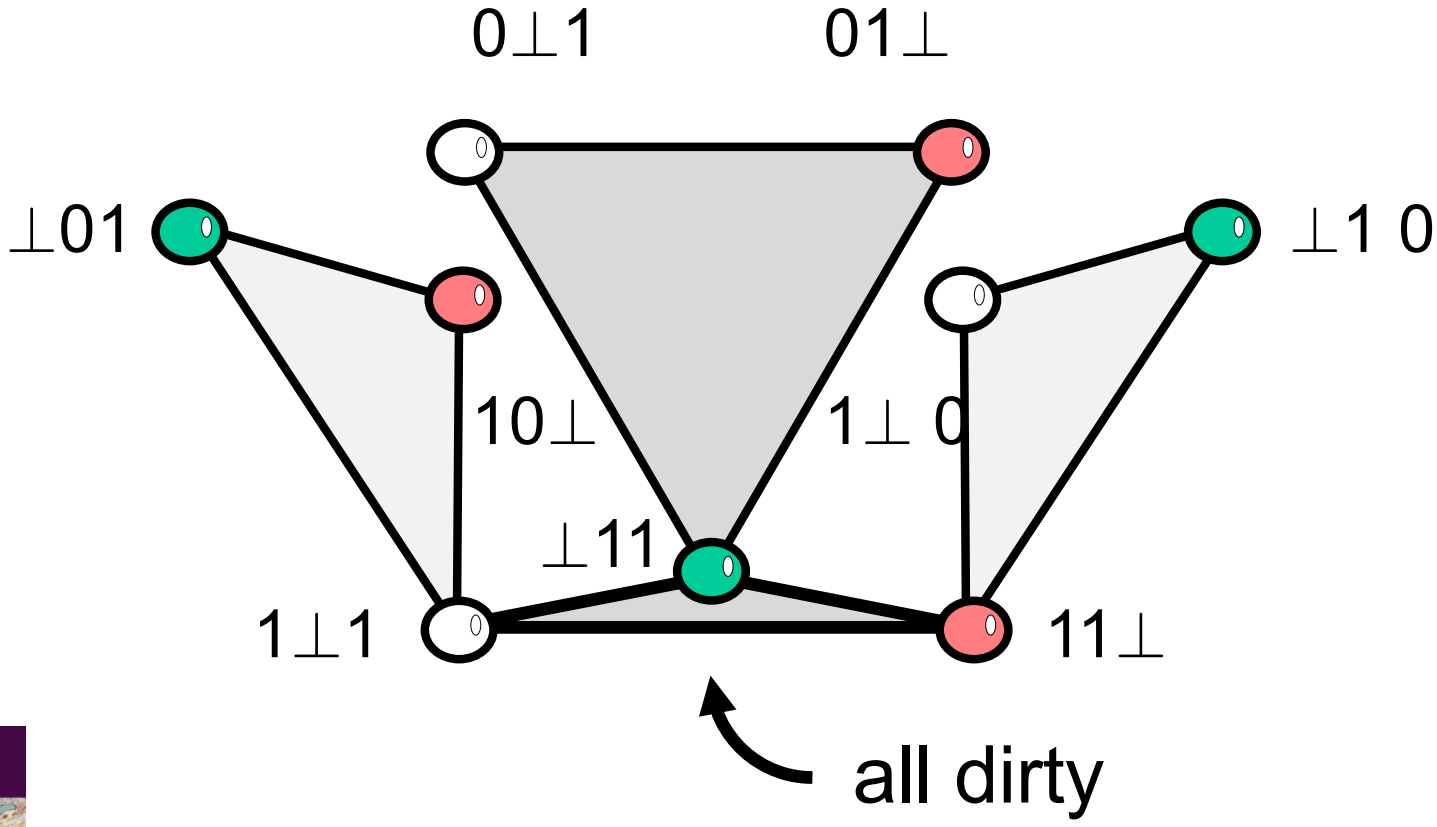
Distributed Computing though
Combinatorial Topology

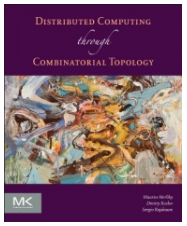




Distributed Computing though
Combinatorial Topology

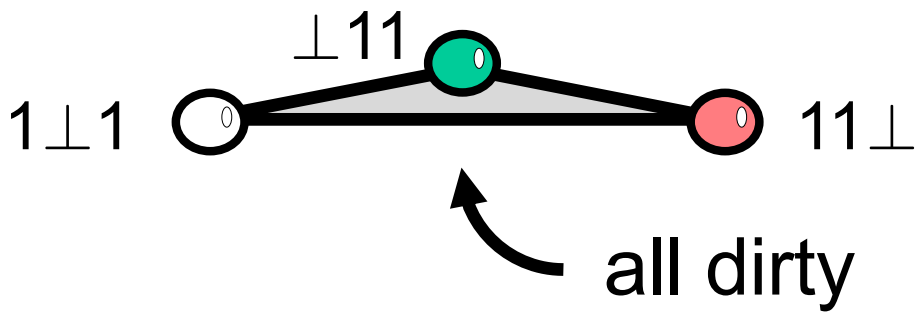


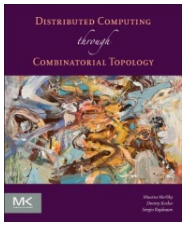




Distributed Computing though
Combinatorial Topology

2:01





Distributed Computing though
Combinatorial Topology

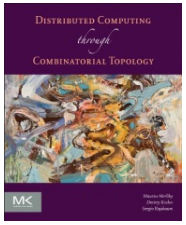
Road Map

Distributed Computing

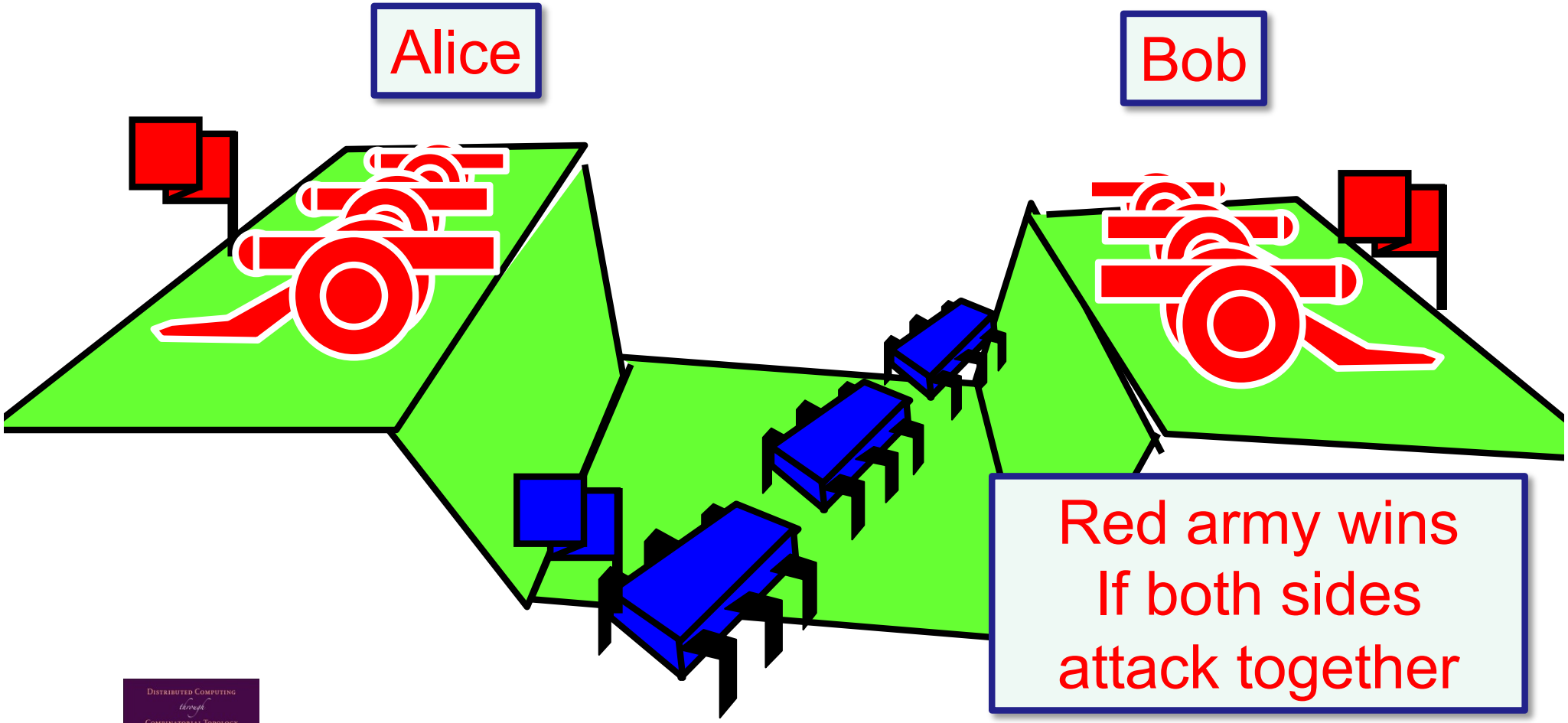
Two Classic Distributed Problems

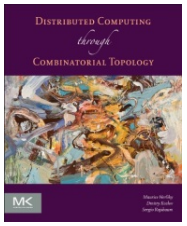
The Muddy Children

Coordinated Attack

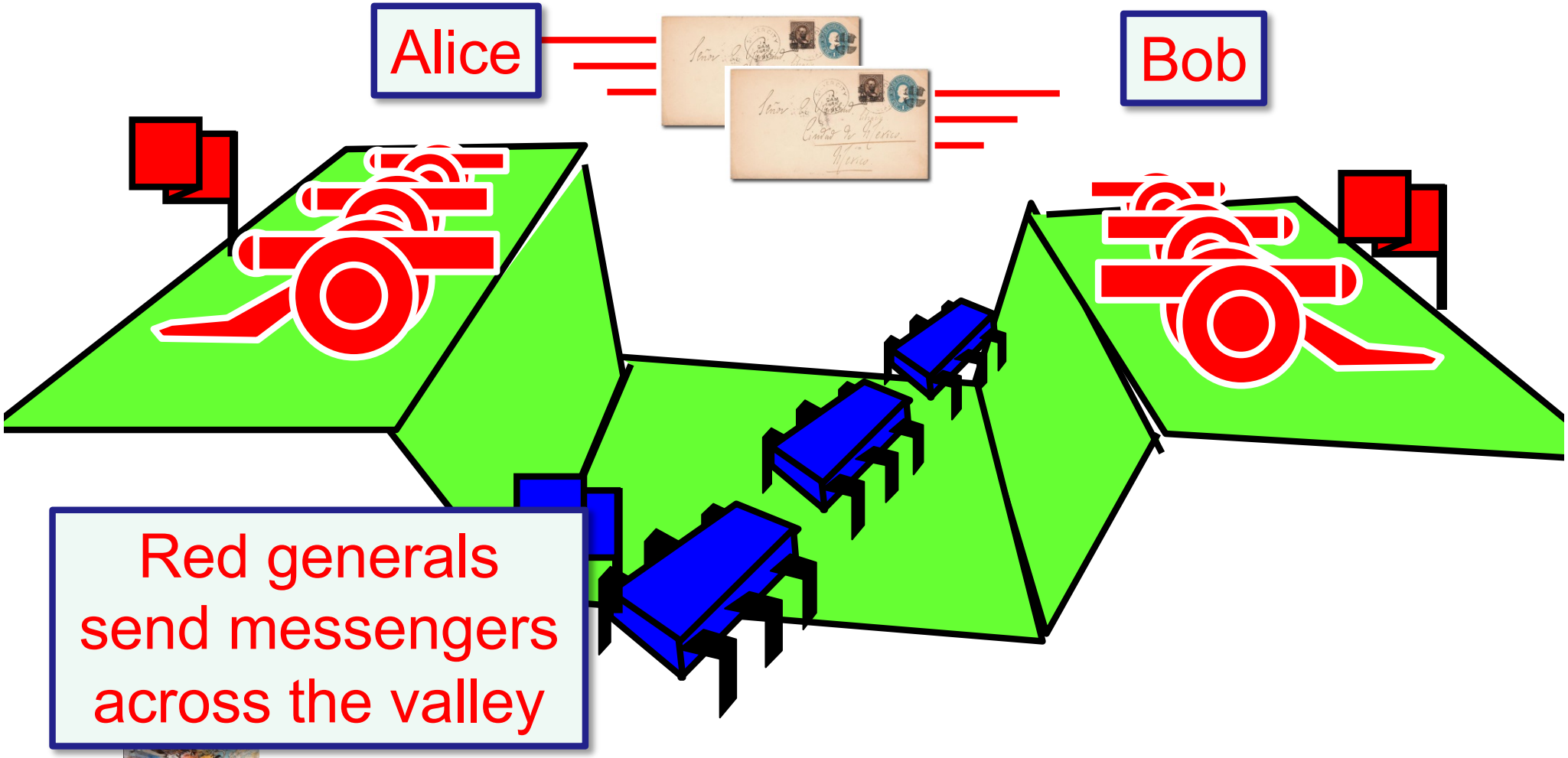


Coordinated Attack

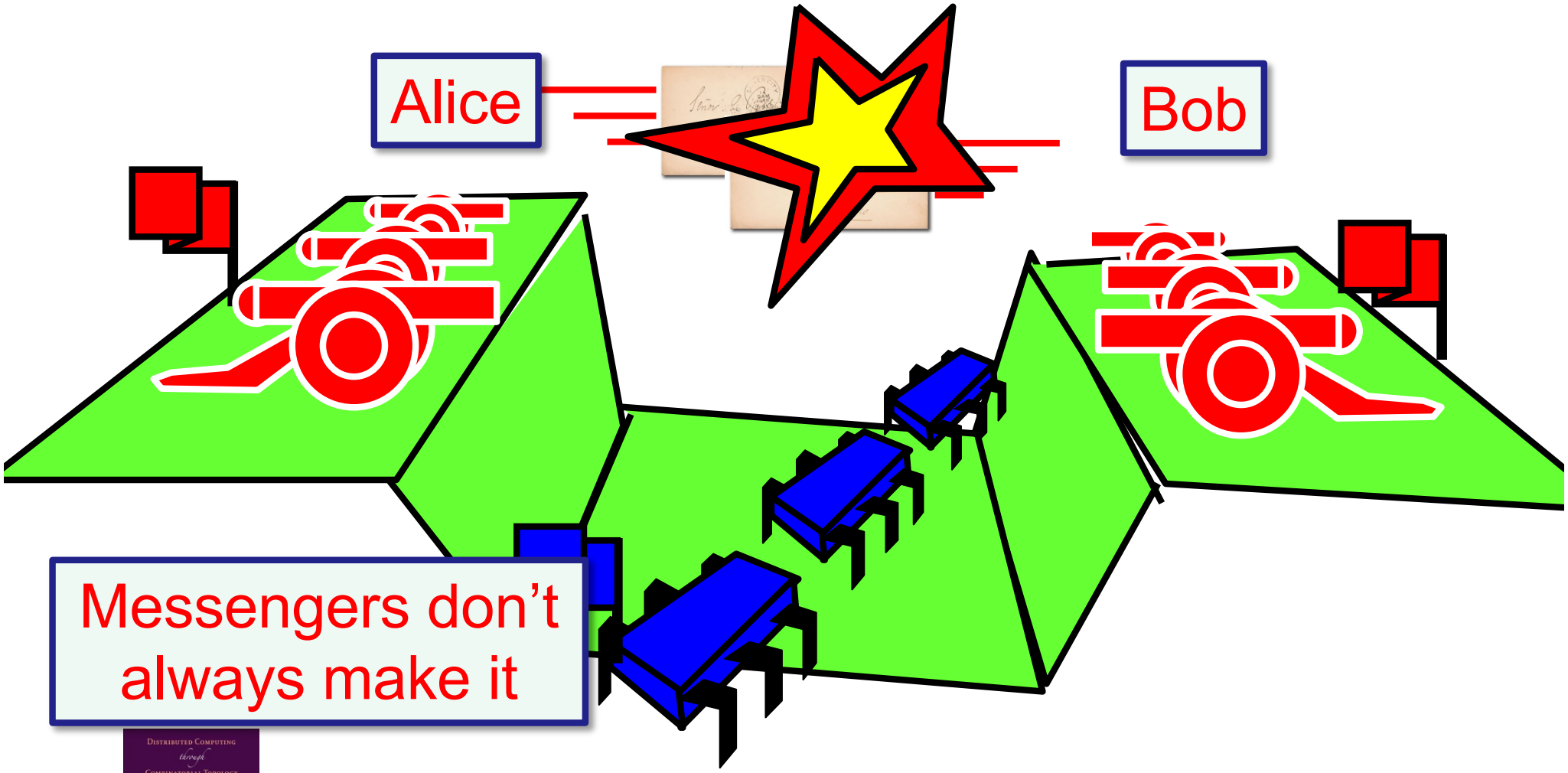


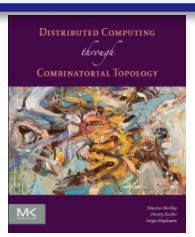


The Two Generals



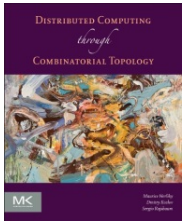
The Two Generals





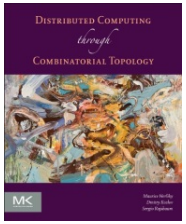
Your Mission

Design a protocol to ensure that Alice and Bob attack simultaneously



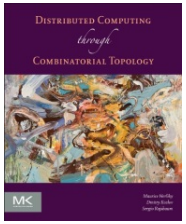
Theorem

There is no protocol that ensures that the Red armies attack simultaneously



Operational Proof

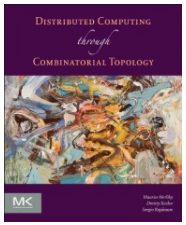
Suppose Bob receives a message at 1:00 saying "attack at Dawn".



Operational Proof

Suppose Bob receives a message at 1:00 saying "attack at Dawn".

Are we done?



Operational Proof

Suppose Bob receives a message at 1:00 saying "attack at Dawn".

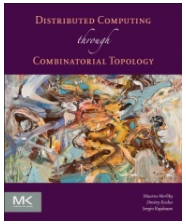
Are we done?

No, because Alice doesn't know if Bob got that message …

Operational Proof

So Bob sends an acknowledgment to Alice

…

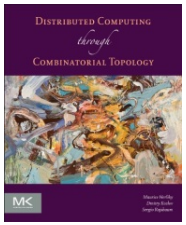


Operational Proof

So Bob sends an acknowledgment to Alice

...

Are we done?



Operational Proof

So Bob sends an acknowledgment to Alice

…

Are we done?

No, because Bob doesn't know if Alice got that message …

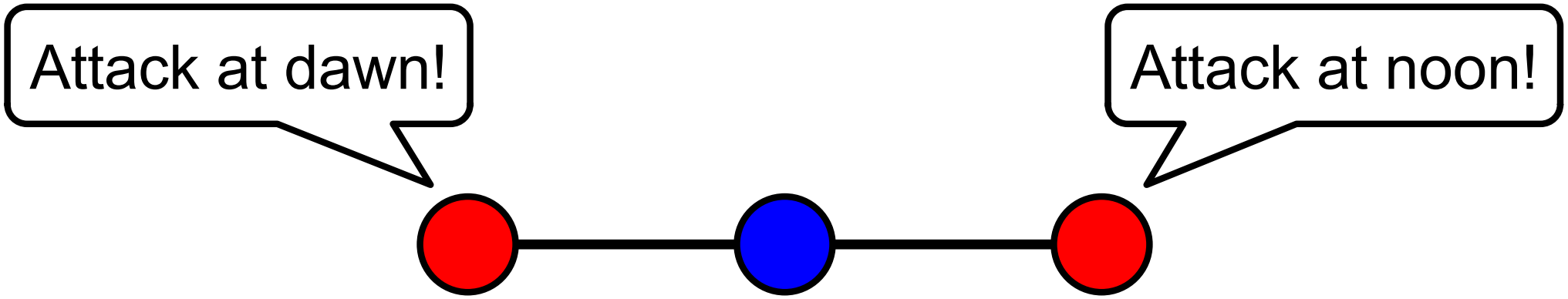
Operational Proof

S Bob sends an ack t to Alice

Are we

No, because Bob de know if Alice got that message …

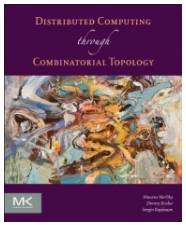
This goes on forever … so no protocol is possible



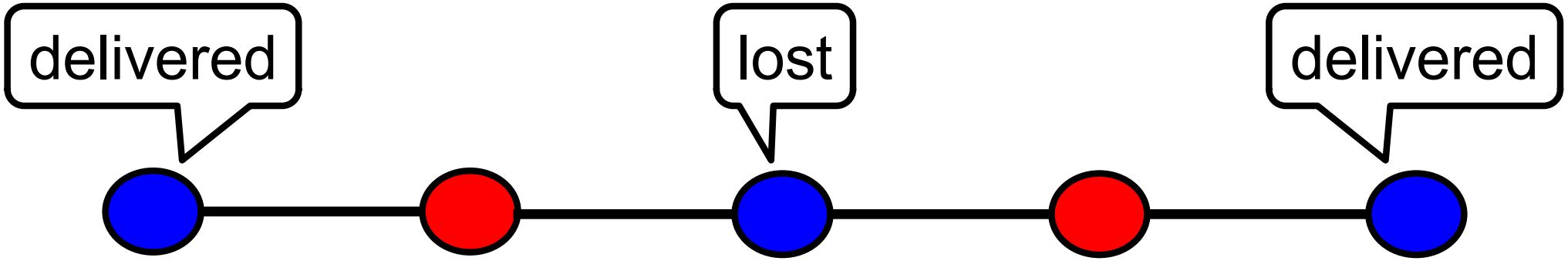
Noon

Bob is

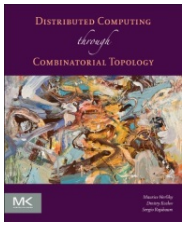
Alice is



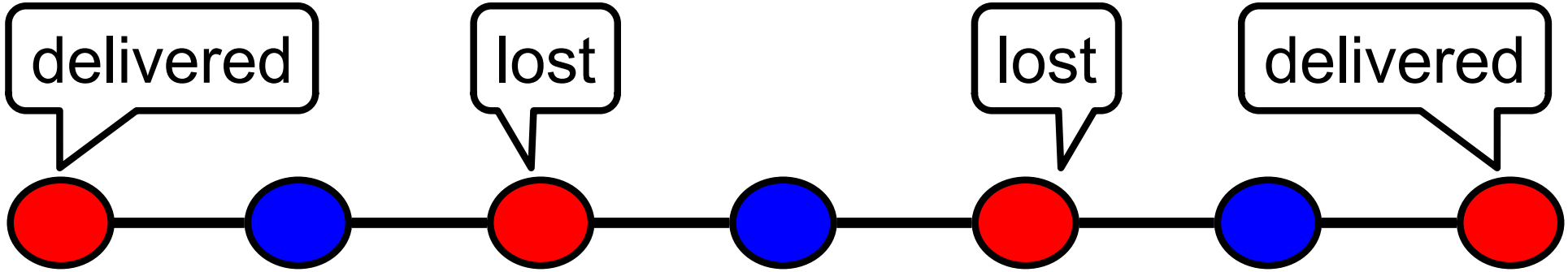
Distributed Computing though
Combinatorial Topology



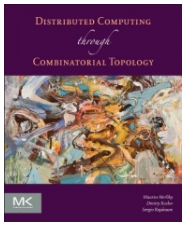
1:00 PM



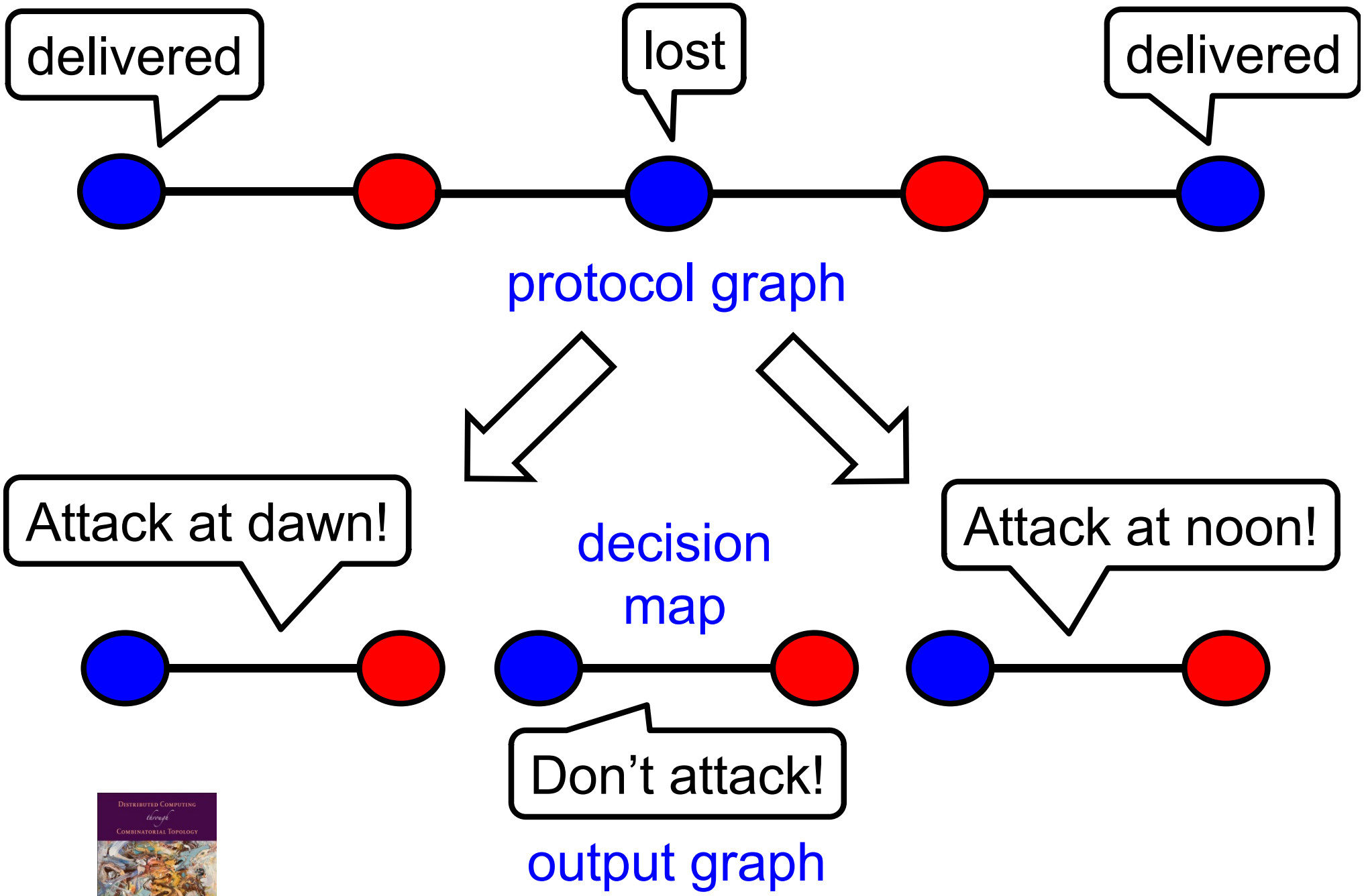
Distributed Computing though
Combinatorial Topology

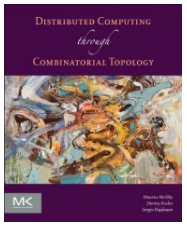


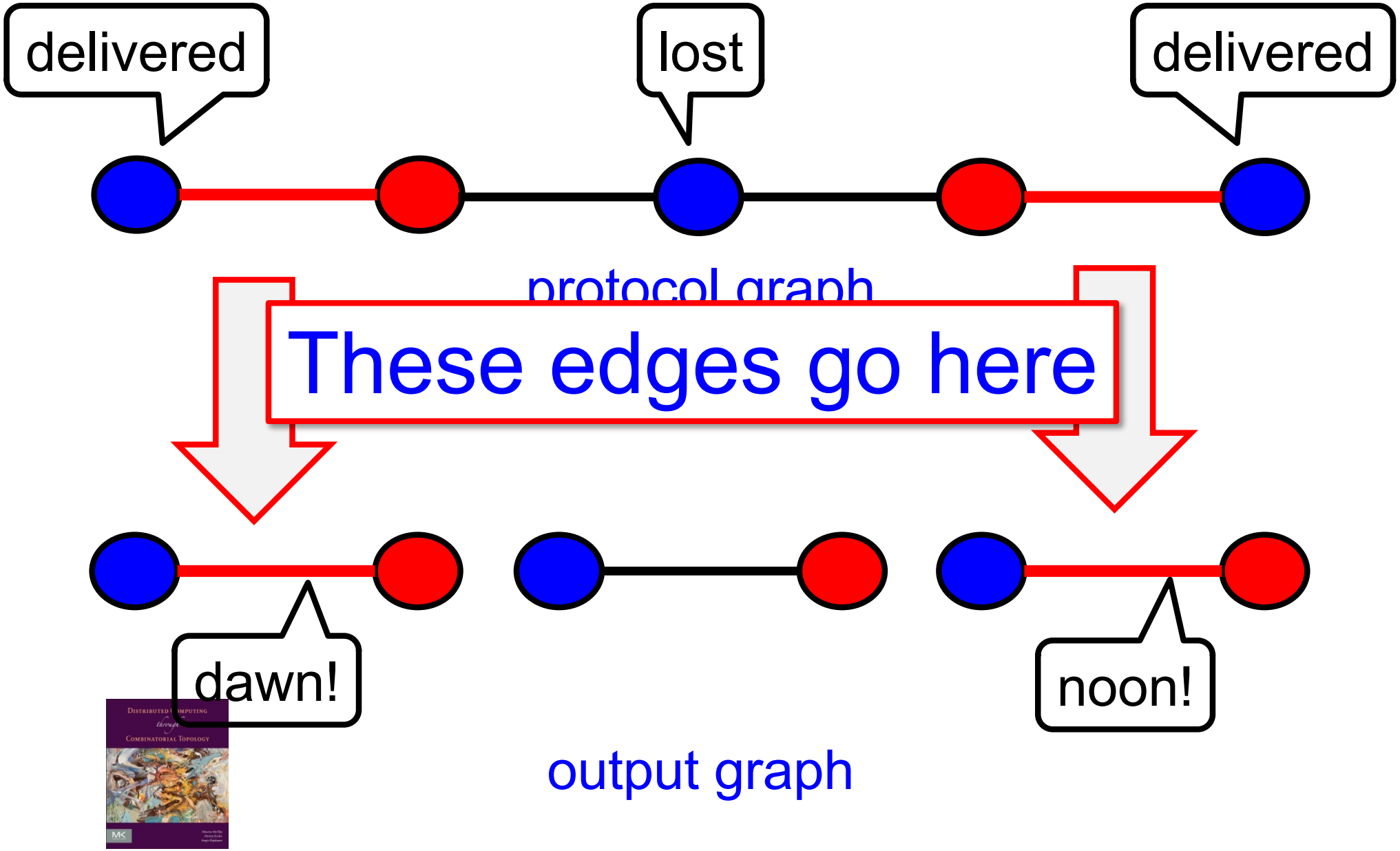
2:00 PM

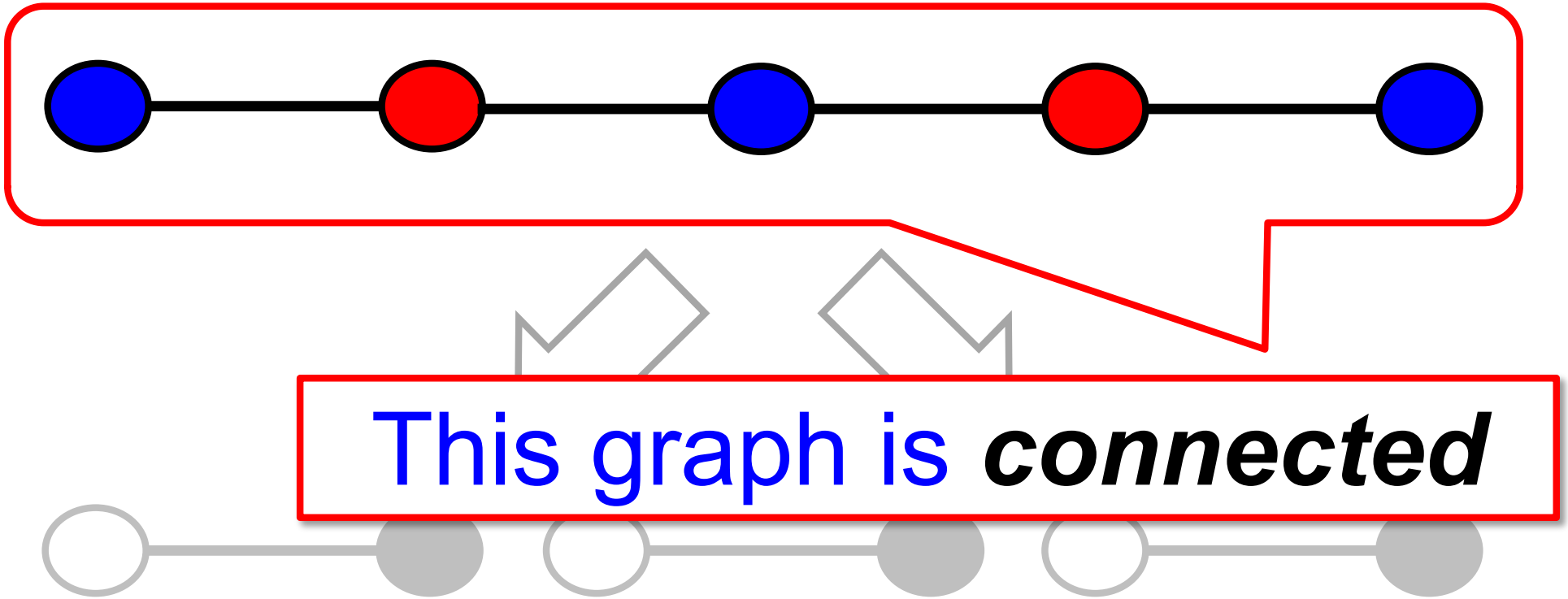


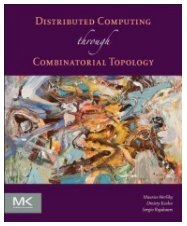
Distributed Computing though
Combinatorial Topology


delivered
lost
delivered
protocol graph
Attack at dawn!
decision
map
Attack at noon!
Don’t attack!
output graph

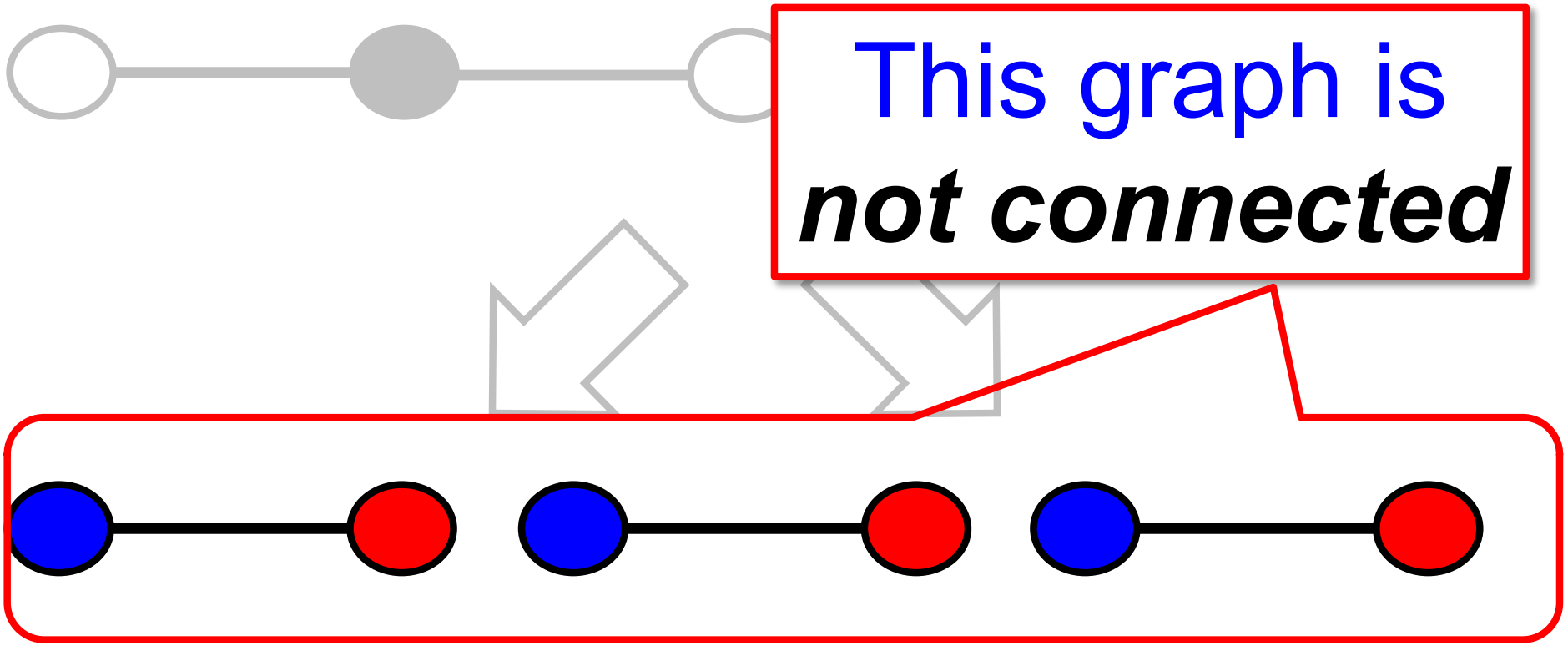

Distributed Computing
through
Combinatorial Topology

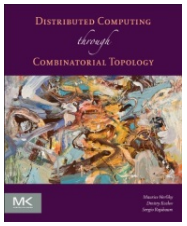

delivered
lost
delivered
protocol graph
These edges go here
dawn!
noon!
output graph



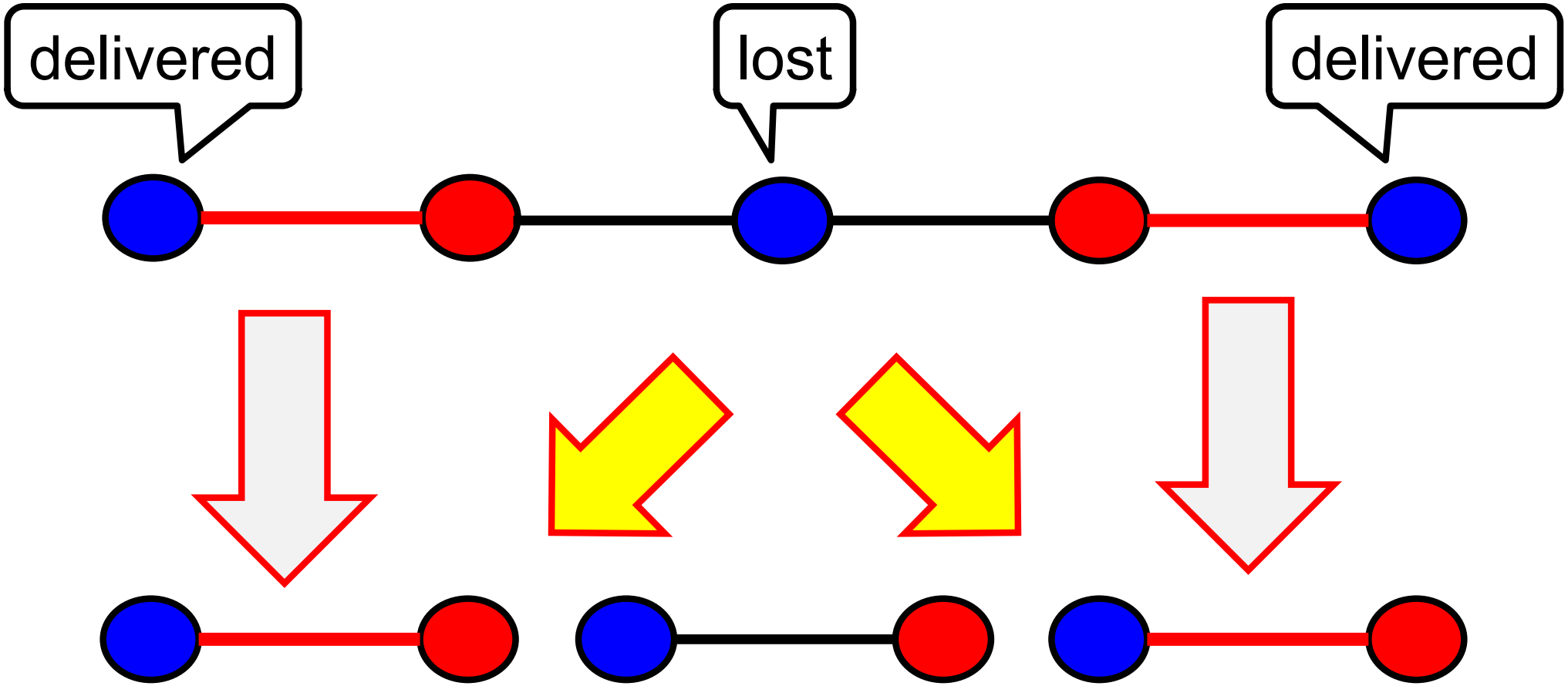


Distributed Computing though
Combinatorial Topology

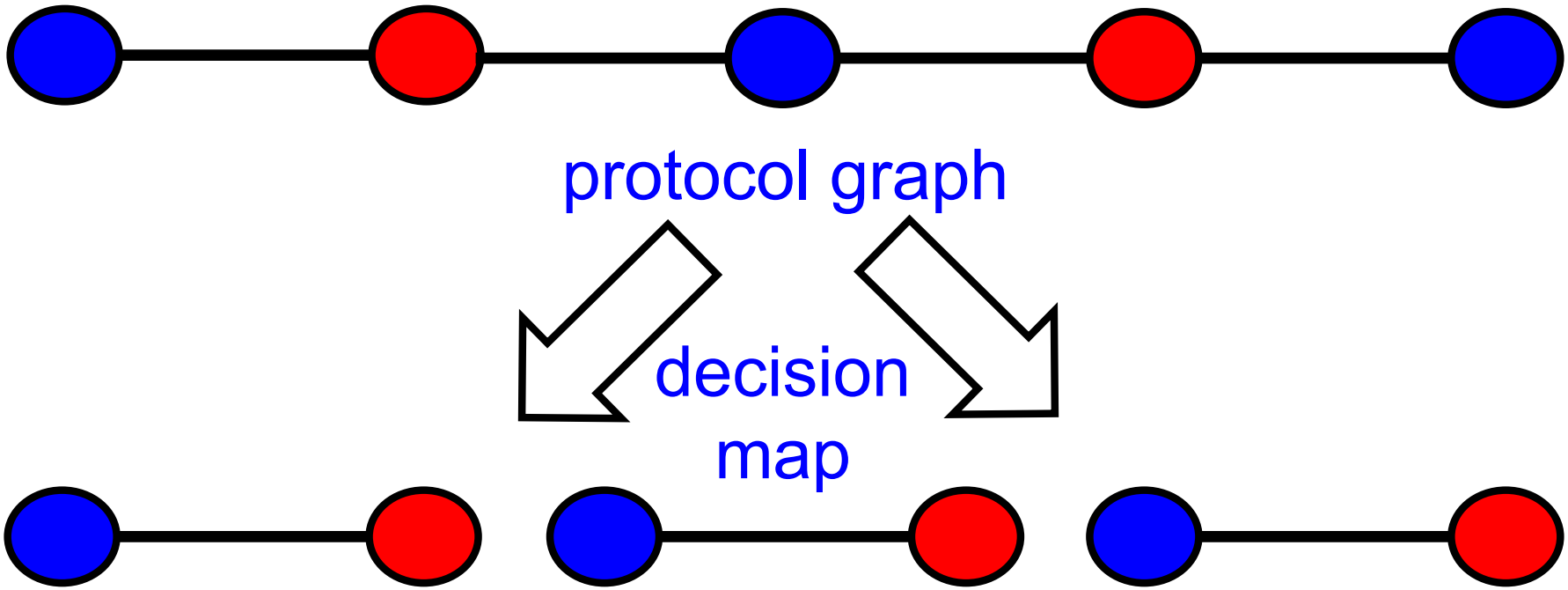


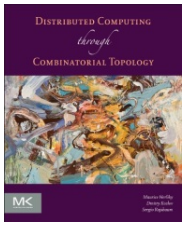


Distributed Computing though
Combinatorial Topology



Map not allowed to “tear” protocol complex



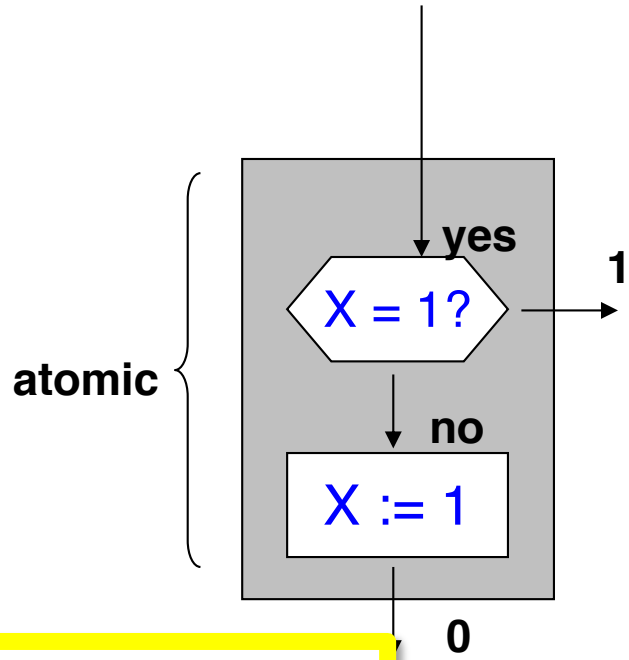


output graph

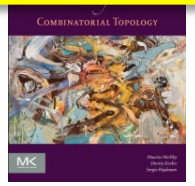
Test and Set

- TAS(X) **tests** if X = 1, **sets** X to 1 if not, and returns the old value of X
  - Instruction available on almost all processors

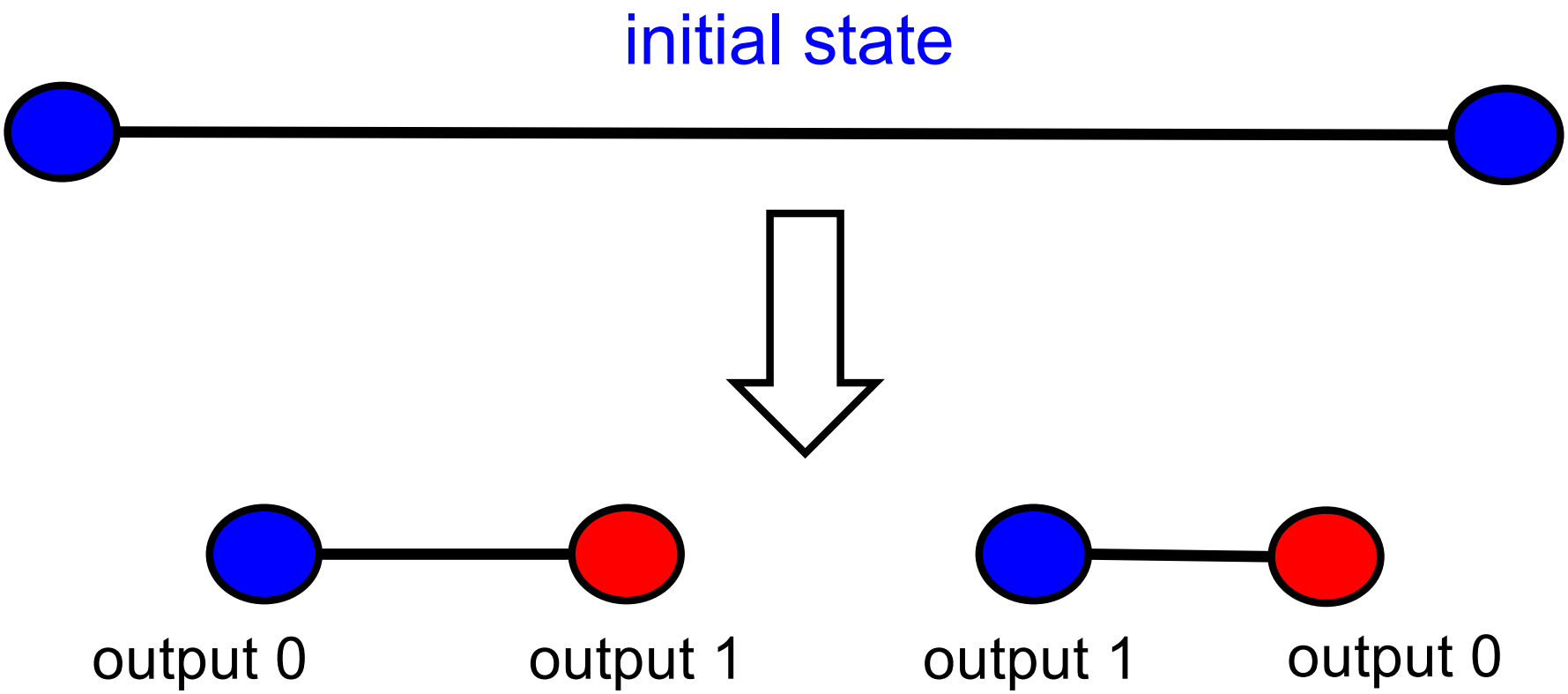
```
TAS(X):

atomic { if X == 1 return 1;
         X = 1;
         return 0;
```

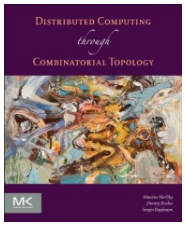


2 processes, P and Q, perform TAS(X)
What is the protocol complex?



2-Process Test and Set





What if there are 3 processes: P, Q, and R?

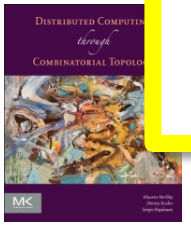
HW: (Simplified) Peterson's lock: 2 processes

```
bool flag[0]   = false;
bool flag[1]   = false;
int turn;
```

```
P0:

flag[0] = true;
turn = 1;
if(flag[1] and turn==1){
return false // failure
}
return true
       // critical section
```

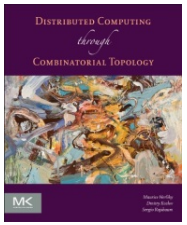
```
P1:

flag[1] = true;
turn = 0;
if(flag[0] and turn==0)
{
return false // failure
}
return true
                // critical
             section
```

What is the protocol complex?
Can we prove that the two processes cannot be both in the critical section



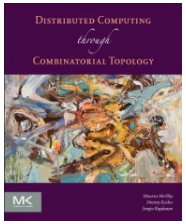
Operational Reasoning

http://commons.wikimedia.org/wiki/File:Professor_Lucifer_Butts.gif



Combinatorial Reasoning

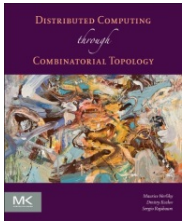
ttp://commons.wikimedia.org/wiki/File:Blake_ancient_of_days.jp



Combinatorial Reasoning

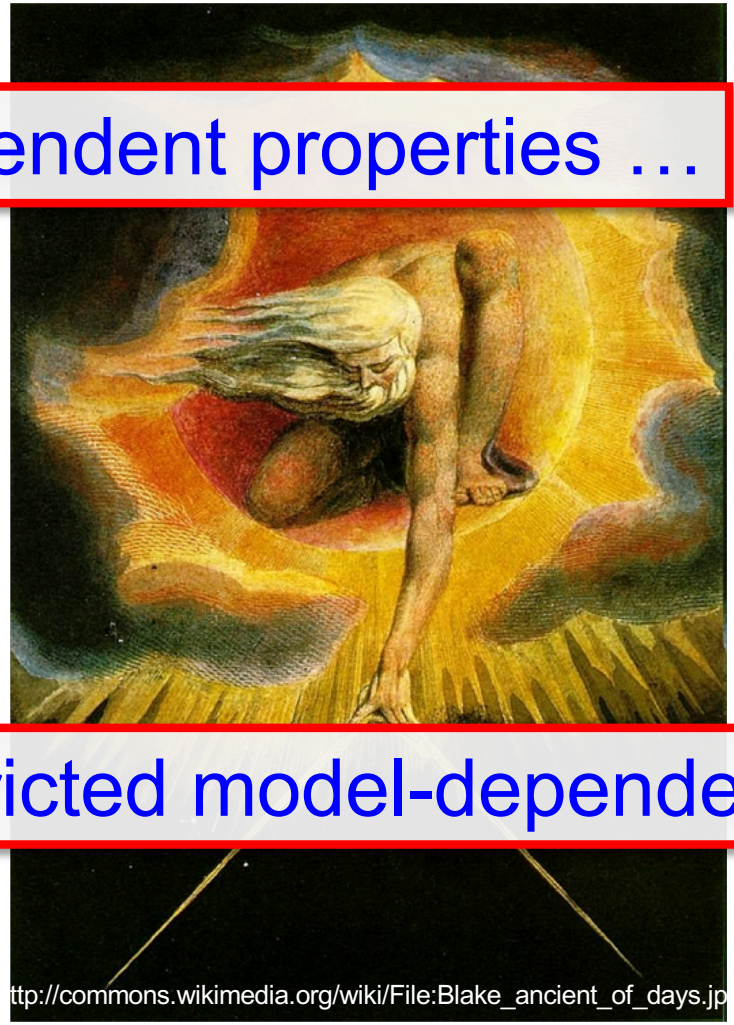
Model-independent properties …

ttp://commons.wikimedia.org/wiki/File:Blake_ancient_of_days.jp



Combinatorial Reasoning

Model-independent properties …



… restricted model-dependent reasoning