Computability in distributed computing

Sergio Rajsbaum
Instituto de Matemáticas
UNAM

From the book coauthored with Maurice Herlihy and Dmitry Kozlov to be published by Elsevier

April 2013

Computability in distributed computing

Turing, topology, and three stories about love

Sergio Rajsbaum
Instituto de Matemáticas
UNAM

From the book coauthored with Maurice Herlihy and Dmitry Kozlov to be published by Elsevier

April 2013

Sequential Computing

- Turing Year 2012, on the occasion of the centenary of his birth, many celebrations all over the world
- Turing machine (TM) as a model of computation
- Church–Turing thesis. TM provides a precise definition of a "mechanical procedure".
- Today, the TM continues to be a model of choice for investigating theory of computation.

What about concurrency?

Concurrency is everywhere

Nearly every activity in our society (including our minds?) works as a distributed system

Concurrency is everywhere

At a smaller scale:

- as processor feature sizes shrink, they become harder to cool, manufacturers have given up trying to make processors faster.
- Instead, they have focused on making processors more parallel.

Concurrency is everywhere

At the other extreme:

- Internet, cloud computing and peer-to-peer systems may encompass thousands of machines that span every continent.

Very different from sequential computing

This revolution requires a fundamental change in how programs are written. Need new principles, algorithms, and tools

- The Art of Multiprocessor Programmin
Herlihy & Shavit book

Would not seem according to traditional views

Would not seem according to traditional views

- Equivalence between single-tape and multi-tape TM is interpreted as implying that sequential computing and distributed computing differ in questions of efficiency, but not computability.

Would not seem according to traditional views

- Equivalence between single-tape and multi-tape TM is interpreted as implying that sequential computing and distributed computing differ in questions of efficiency, but not computability.
- The TM wikipedia page mentions limitations: unbounded computation (OS) and concurrent processes starting others

Why concurrency is different ?

Why concurrency is different ?

Distributed systems are subject to *failures* and *timing uncertainties*, properties not captured by classical multi-tape models.

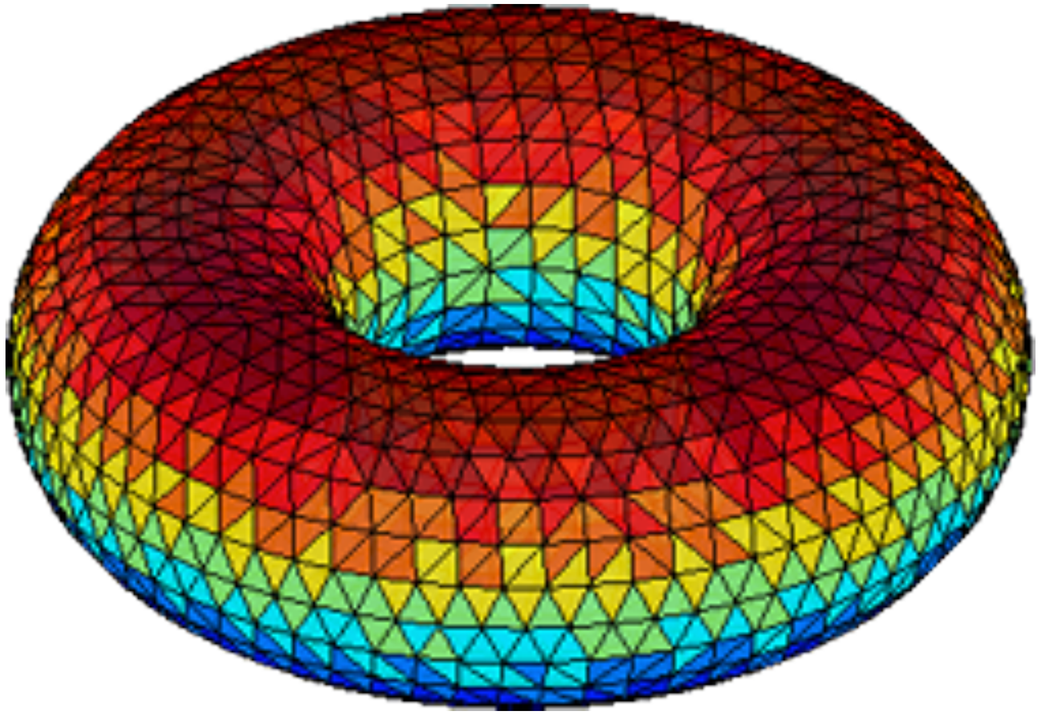
- Even if each process is more powerful than a Turing machine

- Even if each process is more powerful than a Turing machine
- and abstracting away the communication network (processes can directly talk to each other)

- Even if each process is more powerful than a Turing machine
- and abstracting away the communication network (processes can directly talk to each other)
- each one has partial information about reality

Topology

Placing together all these views yields a simplicial complex



"freeze" all possible interleavings and failure scenarios into a single, static, simplicial complex

How topology

The nature of the faults and asynchrony in a system imply topological invariants on the corresponding simplicial complexes, determining what can be computed, and the complexity of the solutions

Distributed computability has a topological nature

- Discovered in 1993: Herlihy, Shavit, Borowski, Gafni, Saks, Zaharoughlu
- Further developed by Attiya, Castaneda, Kouznetsov, Raynal, Travers, Corentin, etc.
- Work from semantics community Eric Goubault, M. Raussen, and others

Three stories about love

Three stories about love

Using topology

The stories

- Cheating wives

  (A.k.a. muddy children, from knowledge theory)

- Two insecure lovers

  (A.k.a. Coordinated attack, from databases and networking)

- ... (later)

Cheating wives

First story

Cheating wives

Cheating wives

- There were one million married couples.

Cheating wives

- There were one million married couples.
- 40 wives were unfaithful

Cheating wives

- There were one million married couples.
- 40 wives were unfaithful
- Each husband knew whether other men's wives were unfaithful but he did now know whether his wife was unfaithful.

Cheating wives

- There were one million married couples.
- 40 wives were unfaithful
- Each husband knew whether other men's wives were unfaithful but he did now know whether his wife was unfaithful.
- The King of the country announced "There is at least one unfaithful wife" and publicized the following decree

Cheating wives decree

He asks the following question over and over:

> can you tell for sure whether or not you are a cuckold?

Cheating wives decree

He asks the following question over and over:

> can you tell for sure whether or not you are a cuckold?

Assuming that all of the men are intelligent, honest, and answer simultaneously, what will happen?

Analysis of the puzzle

First operational, then combinatorial

Operational analysis (1)

First, suppose that exactly <u>one</u> is cuckold

Operational analysis (1)

First, suppose that exactly <u>one</u> is cuckold

- He sees nobody else, can conclude that he is the one

Operational analysis (1)

First, suppose that exactly <u>one</u> is cuckold

- He sees nobody else, can conclude that he is the one
- The others cannot tell whether or not they are cuckolds

Operational analysis (1)

First, suppose that exactly <u>one</u> is cuckold

- He sees nobody else, can conclude that he is the one
- The others cannot tell whether or not they are cuckolds
- At the first question, exactly one says "yes"

Operational analysis (1)

First, suppose that exactly <u>one</u> is cuckold

- He sees nobody else, can conclude that he is the one
- The others cannot tell whether or not they are cuckolds
- At the first question, exactly one says “yes”
- At the second, all others say “no”

Operational analysis (2)

Now, suppose that exactly two are cuckolds

Operational analysis (2)

Now, suppose that exactly <u>two</u> are cuckolds

- They know at least two are cuckolds, because nobody spoke in first round

Operational analysis (2)

Now, suppose that exactly two are cuckolds

- They know at least two are cuckolds, because nobody spoke in first round
- They see only one cuckold

Operational analysis (2)

Now, suppose that exactly <u>two</u> are cuckolds

- They know at least two are cuckolds, because nobody spoke in first round
- They see only one cuckold
- At the second question, exactly two says “yes”

Operational analysis (2)

Now, suppose that exactly <u>two</u> are cuckolds

- They know at least two are cuckolds, because nobody spoke in first round
- They see only one cuckold
- At the second question, exactly two says “yes”
- At the third, all others say “no”

Operational analysis (3)

Suppose that exactly *k* are cuckolds, by induction...

Operational analysis (3)

Suppose that exactly *k* are cuckolds, by induction...

- At the *k*-th question, exactly *k* say "yes"

Operational analysis (3)

Suppose that exactly *k* are cuckolds, by induction...

- At the *k*-th question, exactly *k* say "yes"
- At the *(k+1)*-th, all others say "no"

Combinatorial analysis

Local states

Combinatorial analysis

Local states

- A local state is a man's state of knowledge

Combinatorial analysis

Local states

- A local state is a man's state of knowledge
- It is represented by a vector: in position *i* has 0 if man *i* is known to be clean, and *1* if cuckold

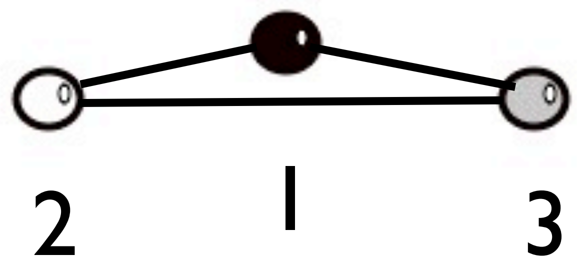
Combinatorial analysis

Local states

- A local state is a man's state of knowledge
- It is represented by a vector: in position *i* has 0 if man *i* is known to be clean, and *1* if cuckold
- Because man *i* does not know its own status, its input vector has $\perp$ in position i

Global inputs

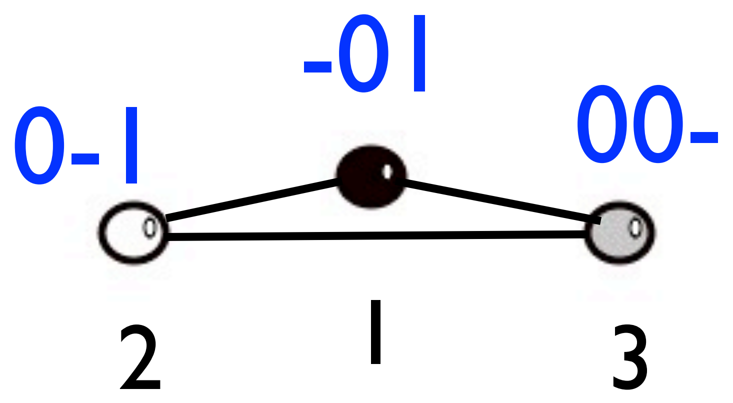
Each possible input configuration is represented as a simplex, linking compatible states for the men



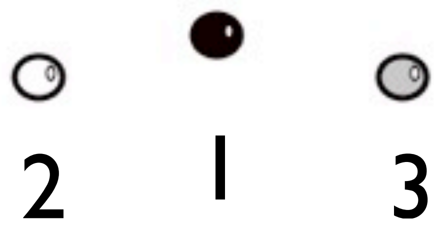
meaning that the men can be in these states together

Global inputs

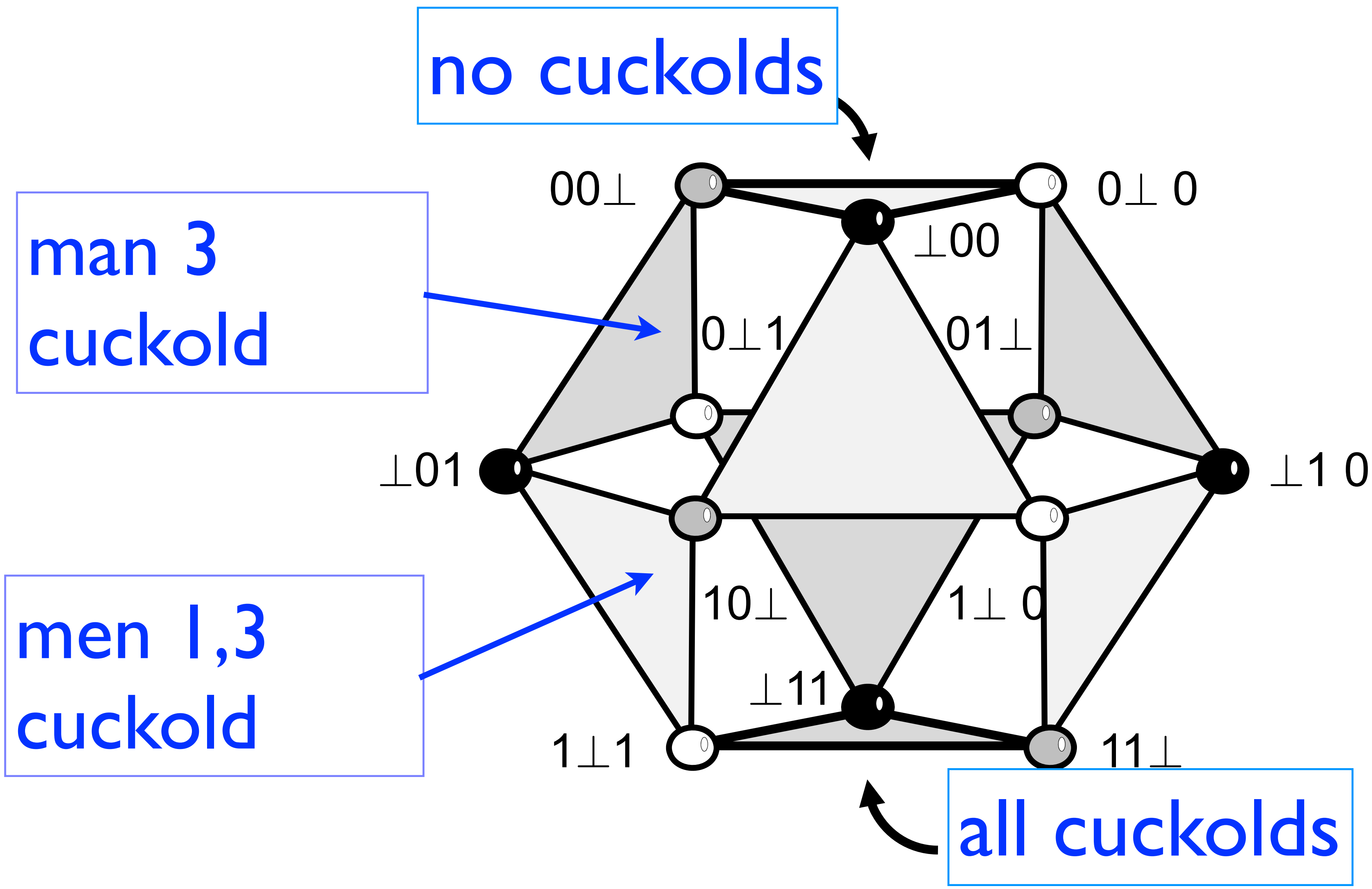
Each possible input configuration is represented as a simplex, linking compatible states for the men

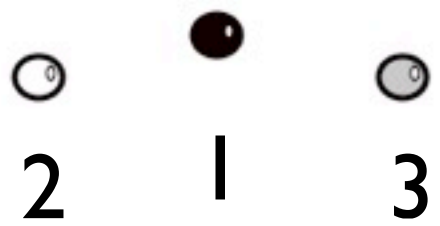


meaning that the men can be in these states together

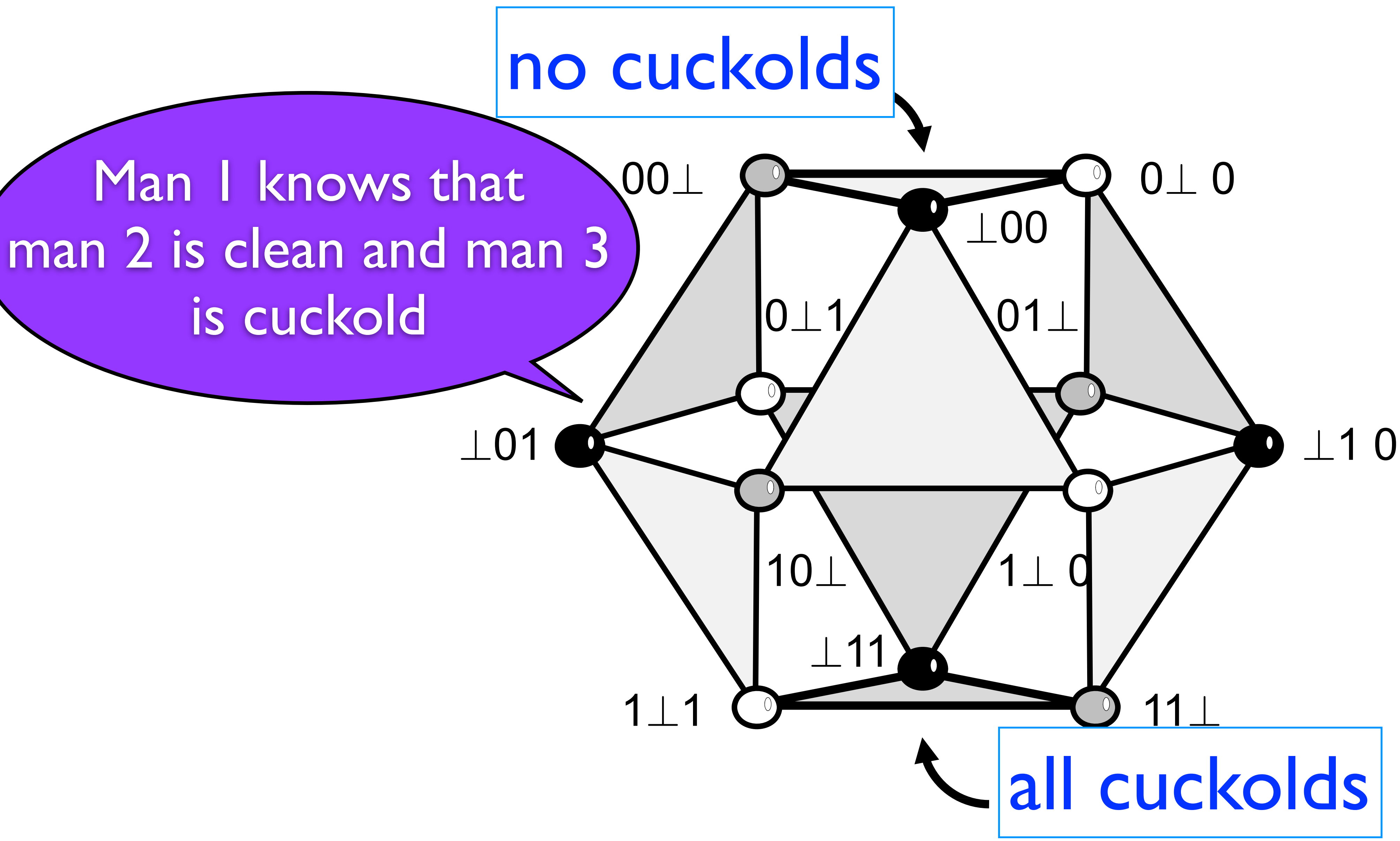


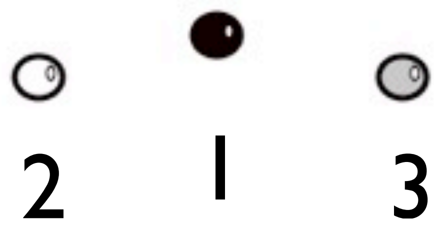
Initial Complex



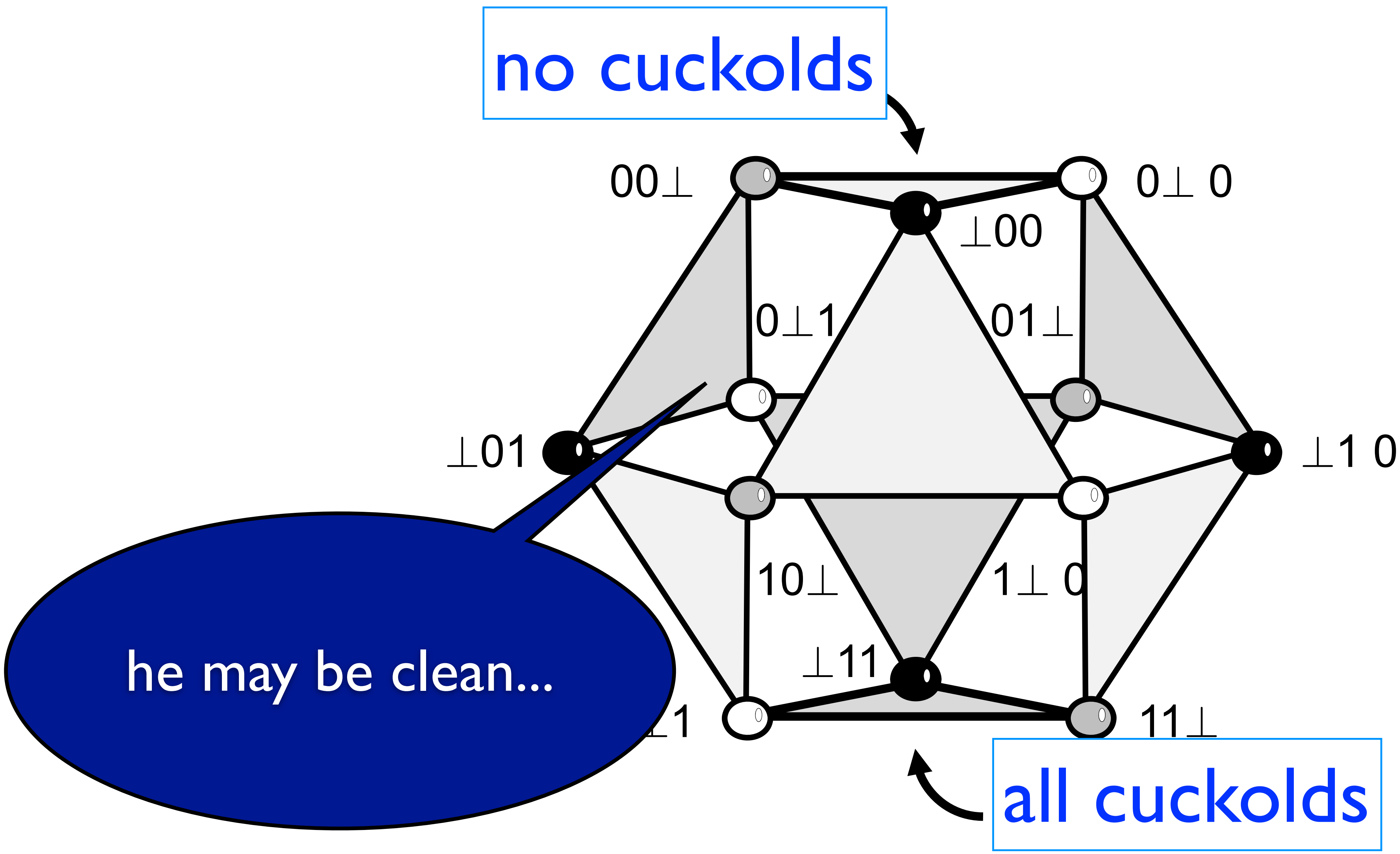


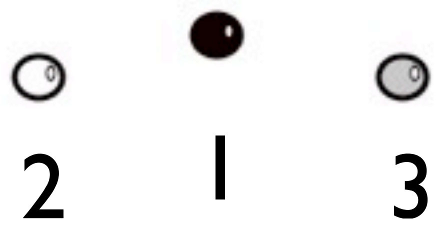
Initial Complex



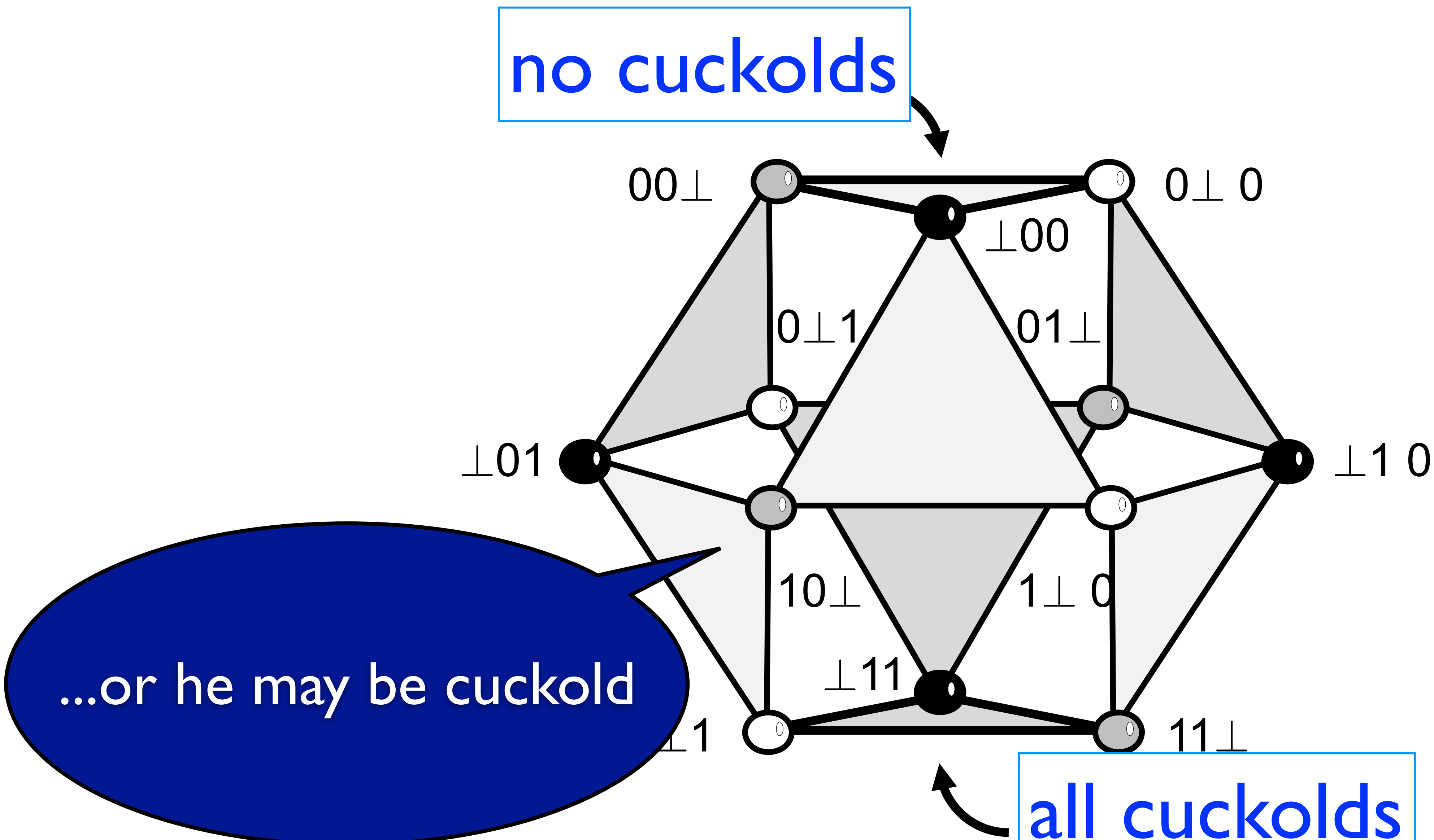


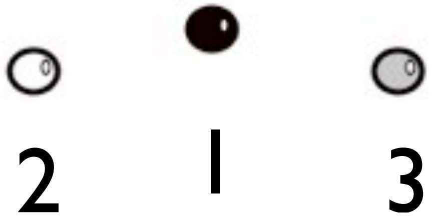
Initial Complex





Initial Complex

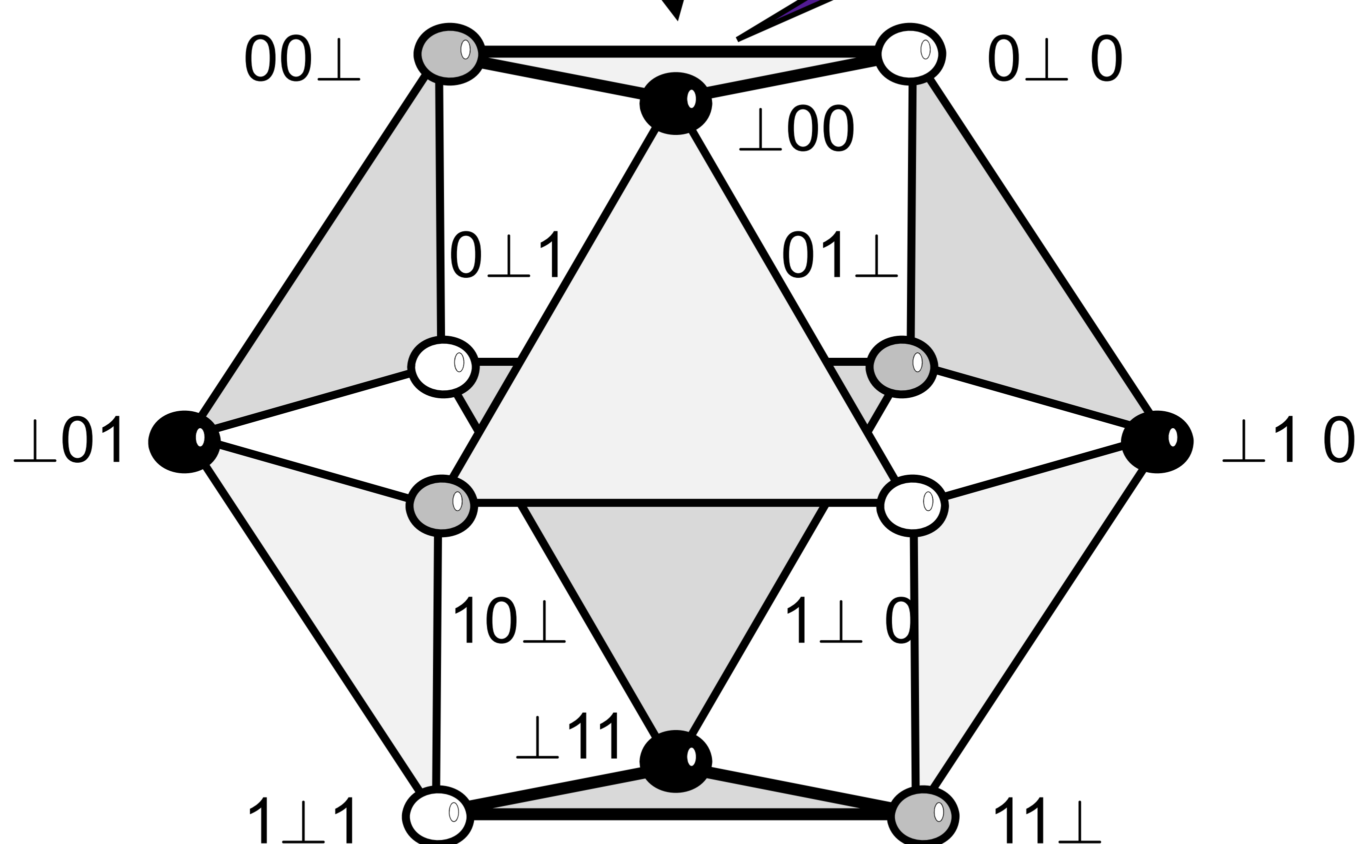




Initial Com

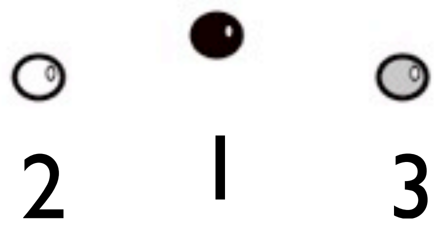
disappears when announced "at least one cuckold"

no cuckolds

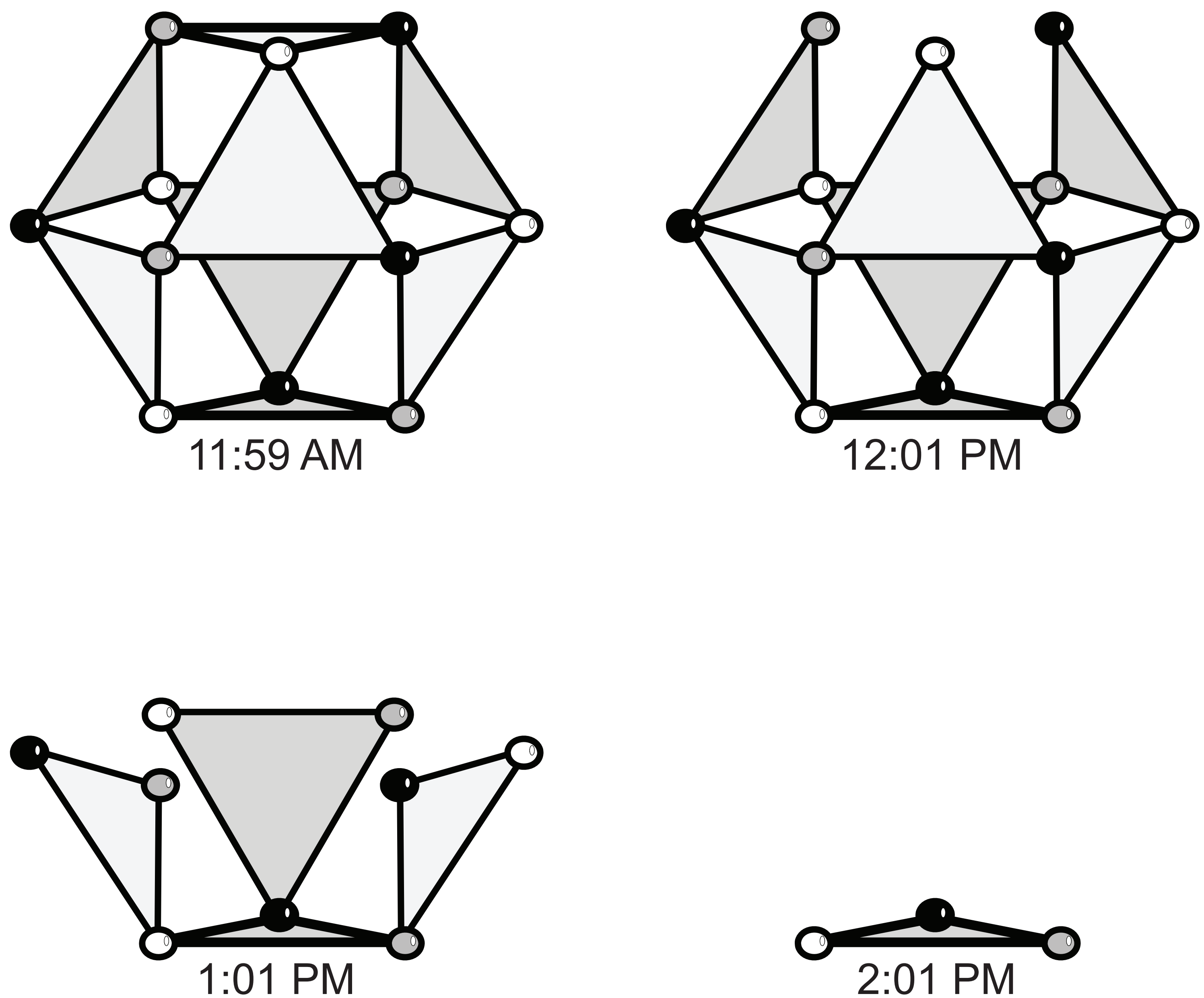


all cuckolds

that is, men know that each 2-simplex is a possible initial state, except for the one where all are clean

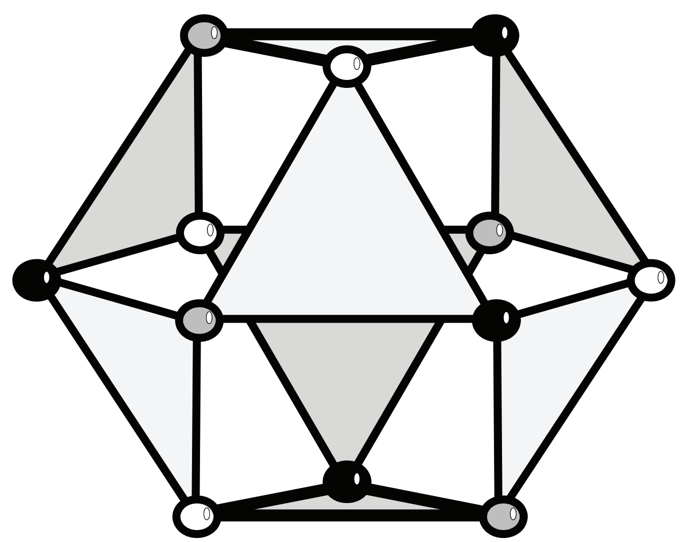


Evolution

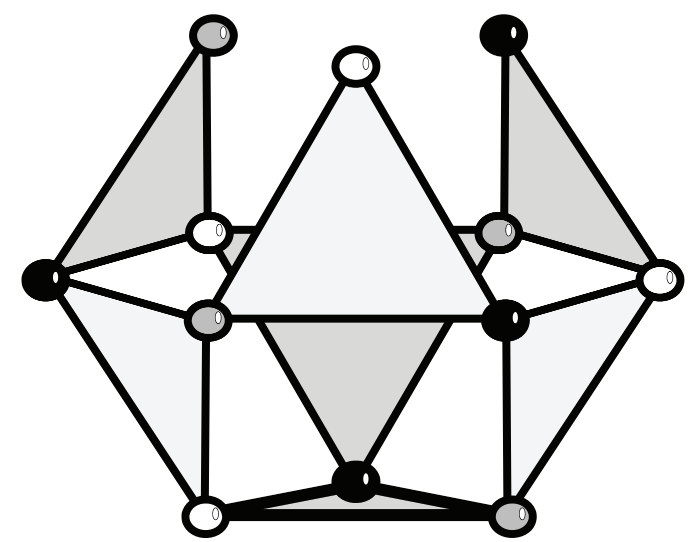


2 1 3

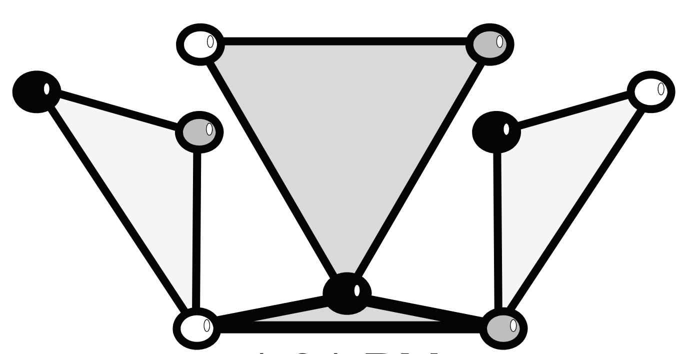
Evolution



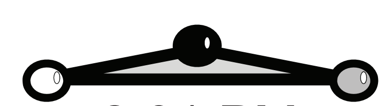
11:59 AM



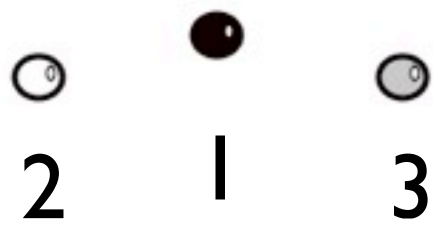
12:01 PM



1:01 PM

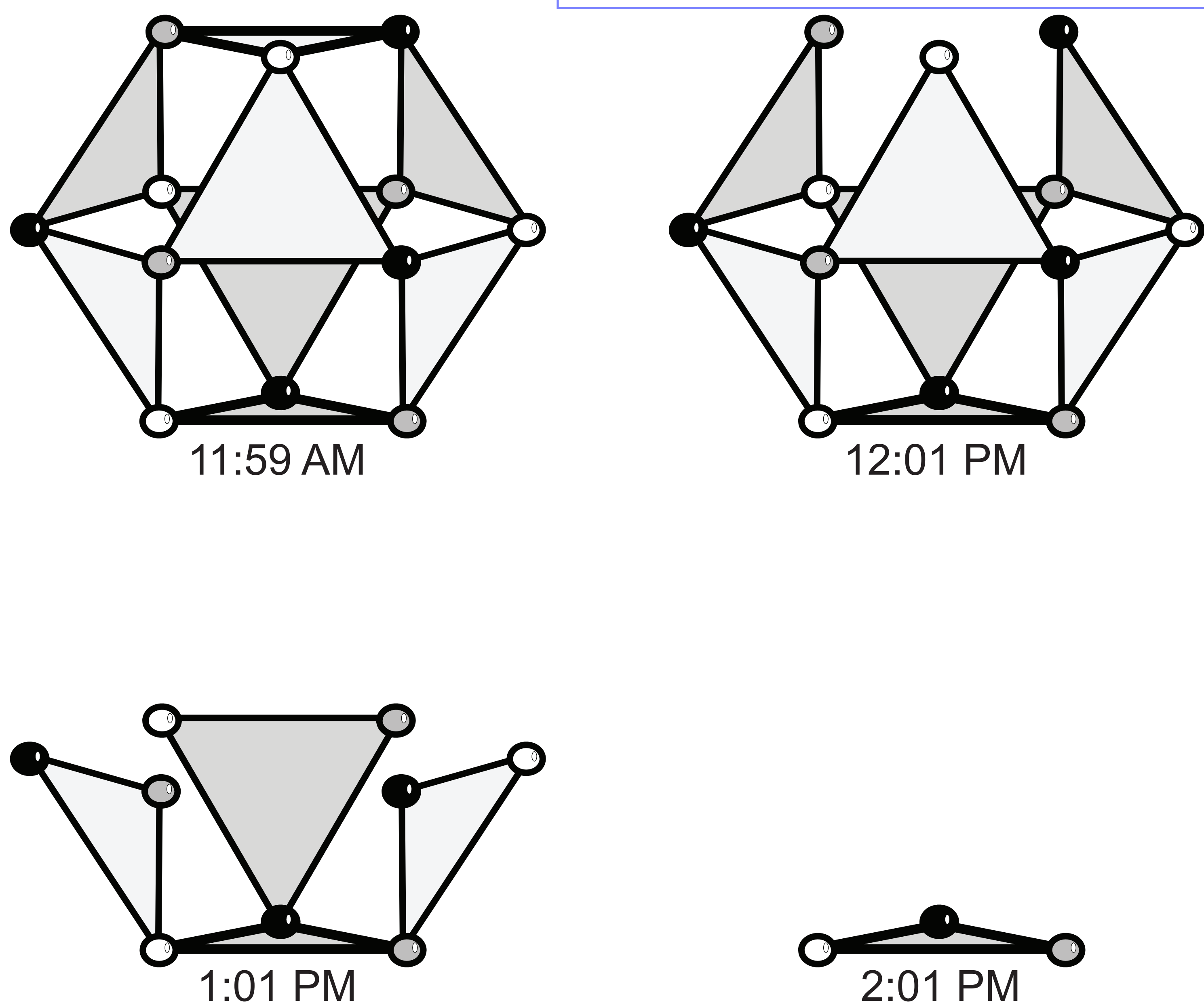


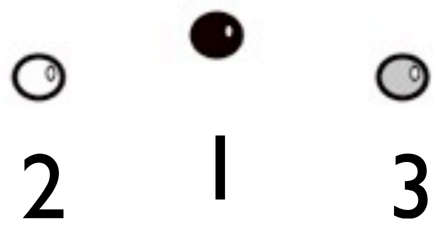
2:01 PM



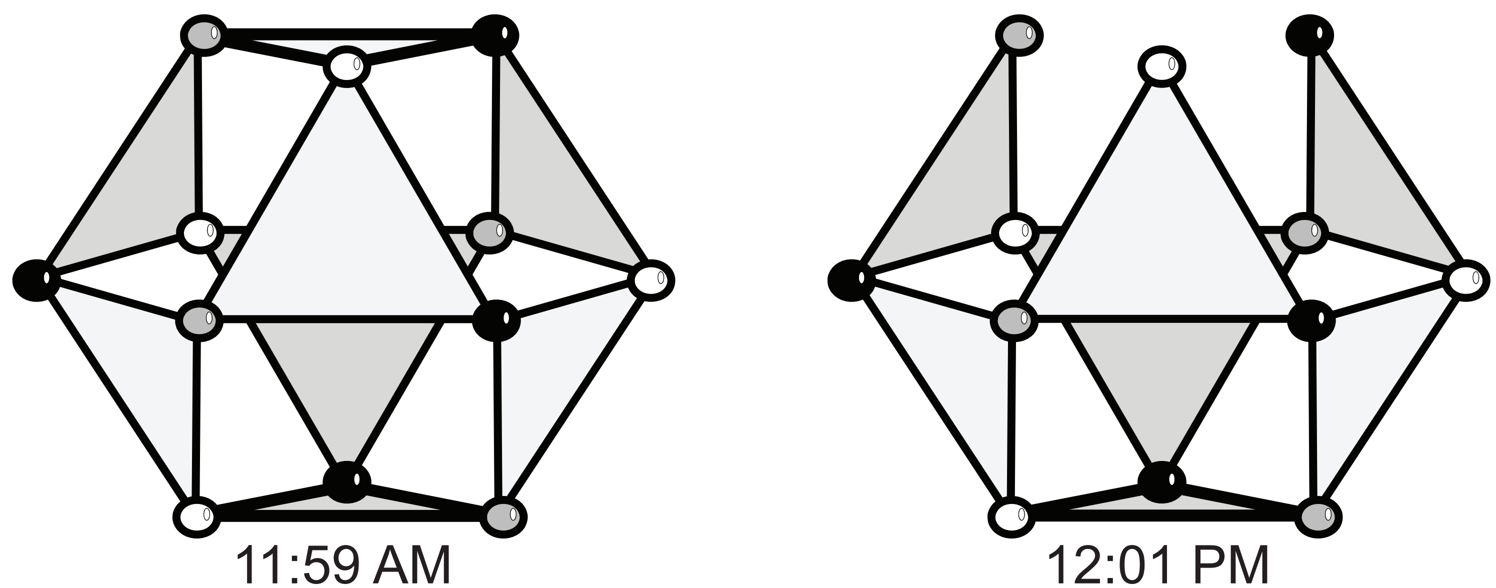
Evolution

3 vertexes exposed, where someone knows its status

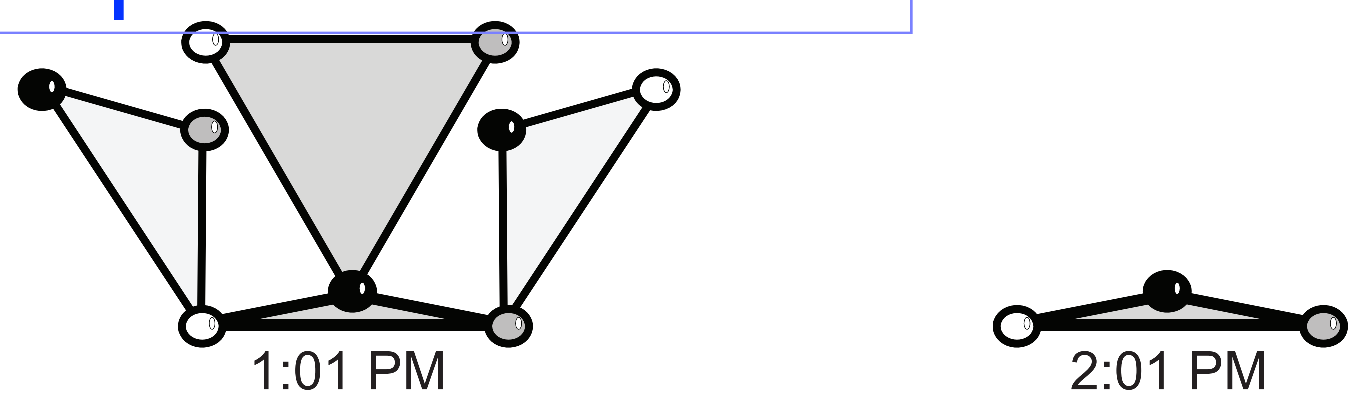


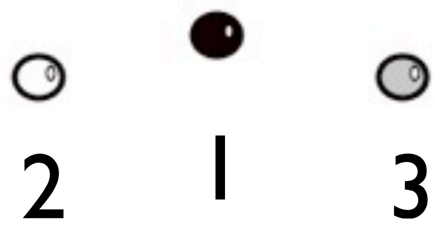


Evolution

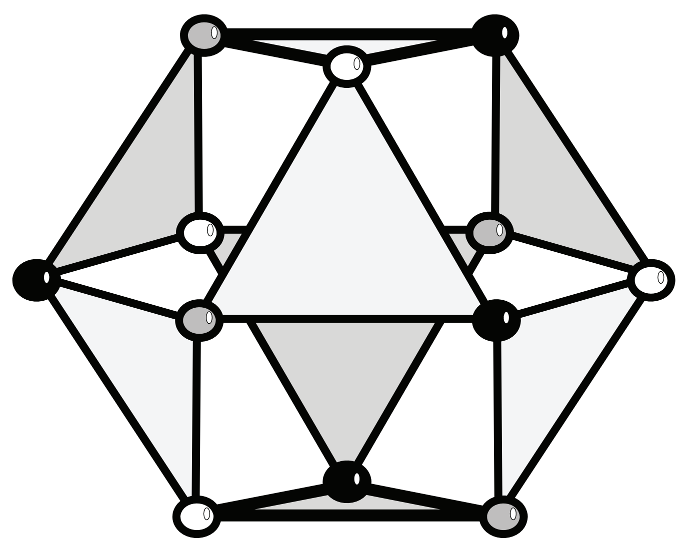


Nobody spoke previous round,
6 vertexes exposed

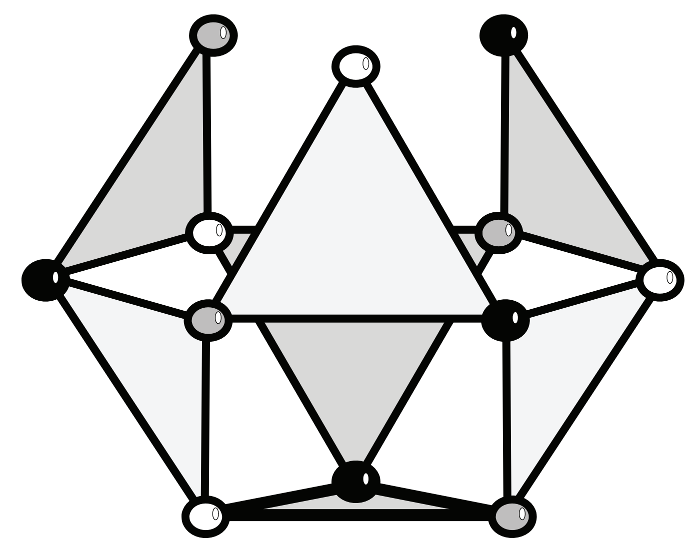




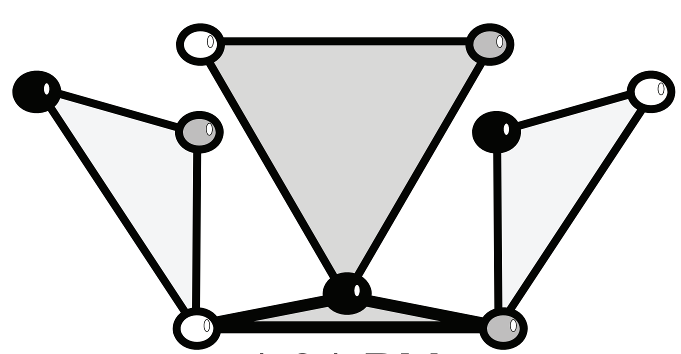
Evolution



11:59 AM

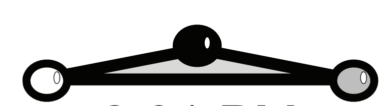


12:01 PM

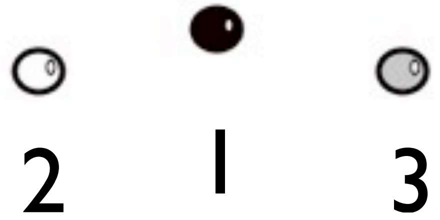


1:01 PM

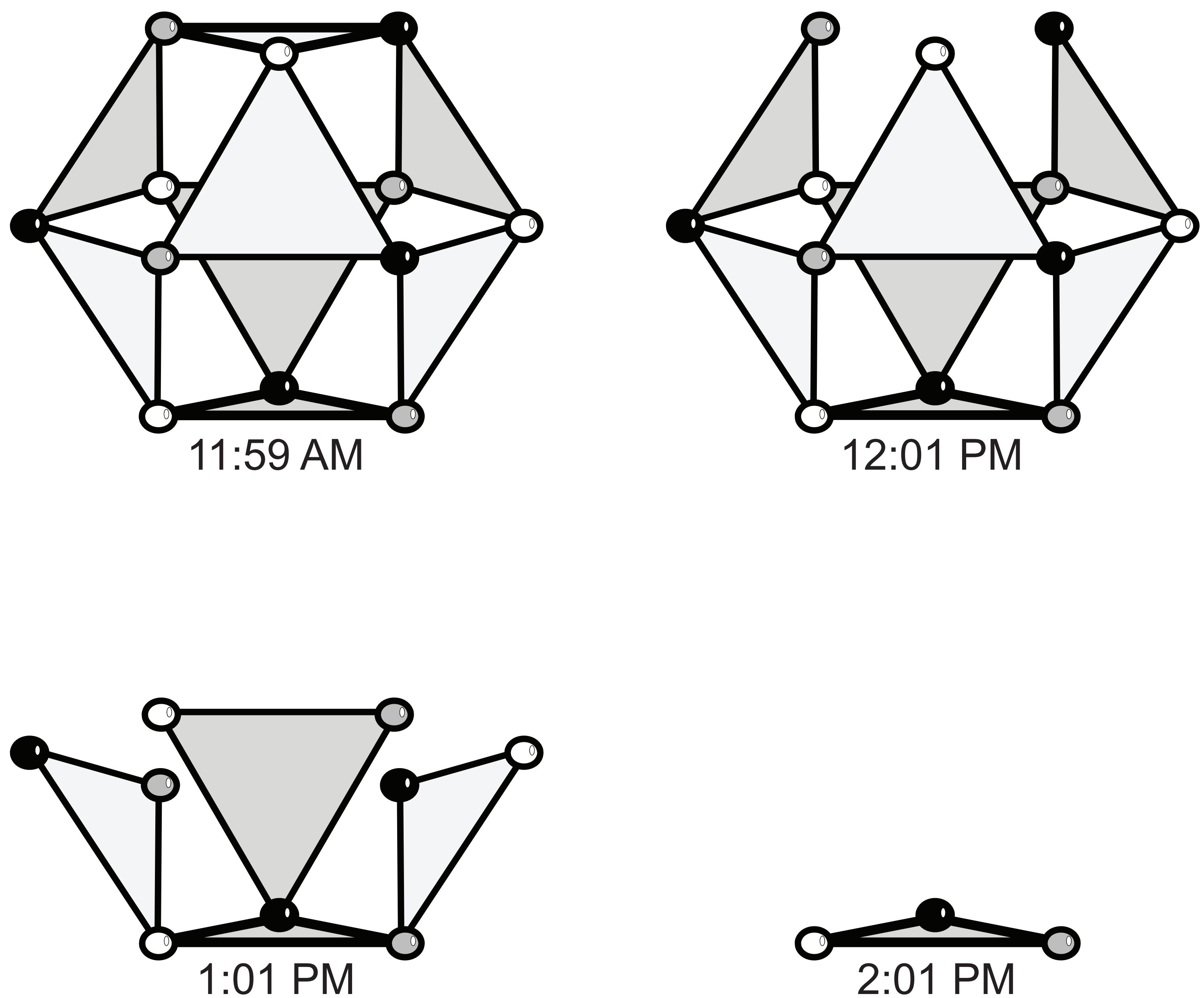
All 3 announce “cuckolds”

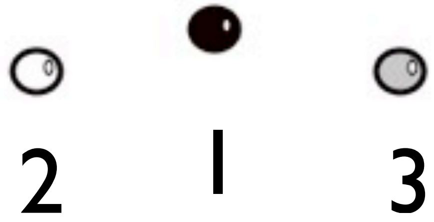


2:01 PM

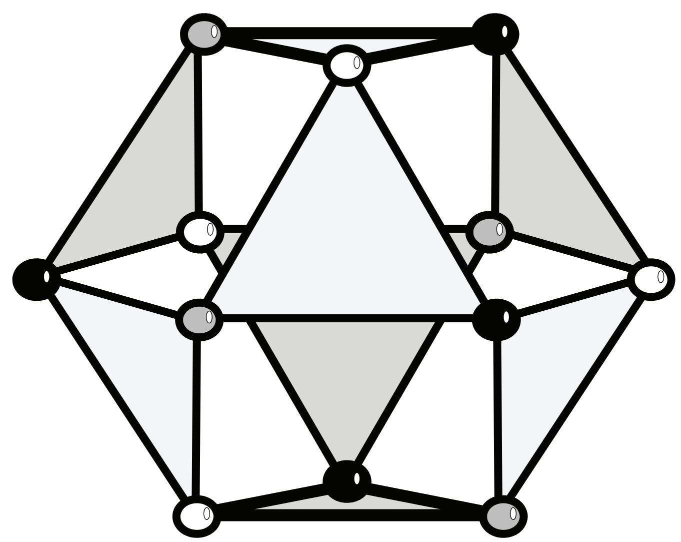


Evolution

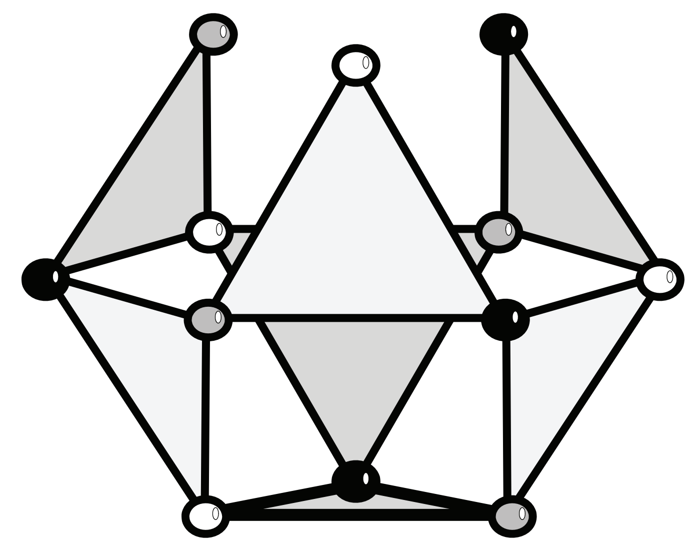




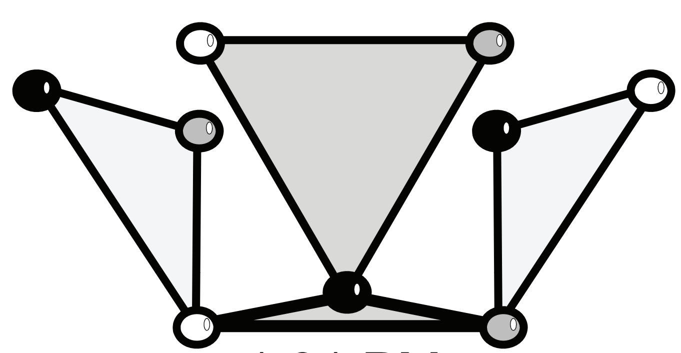
Decisions



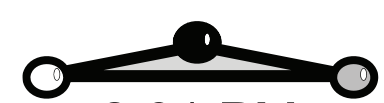
11:59 AM



12:01 PM



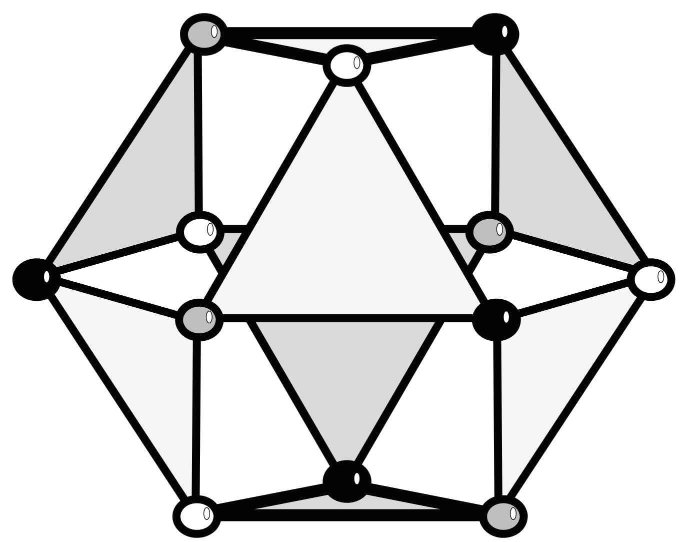
1:01 PM



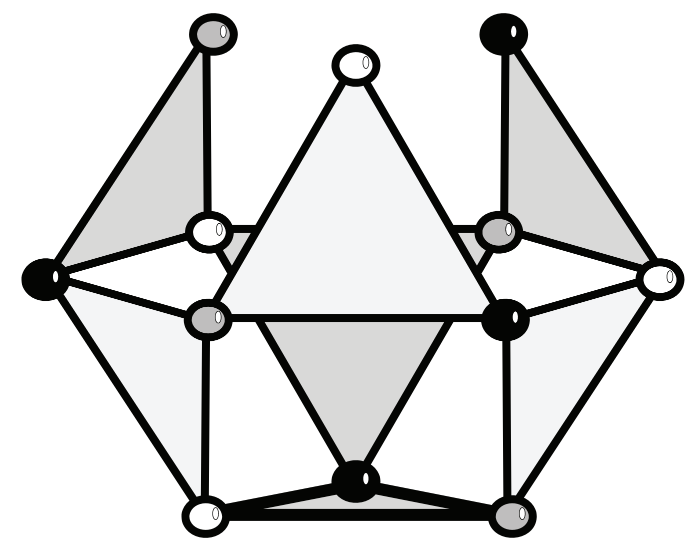
2:01 PM

Decisions

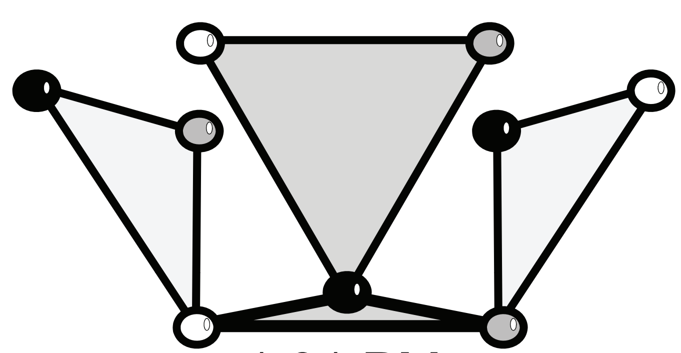
No decisions



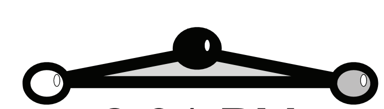
11:59 AM



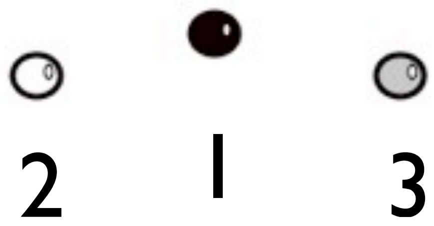
12:01 PM



1:01 PM

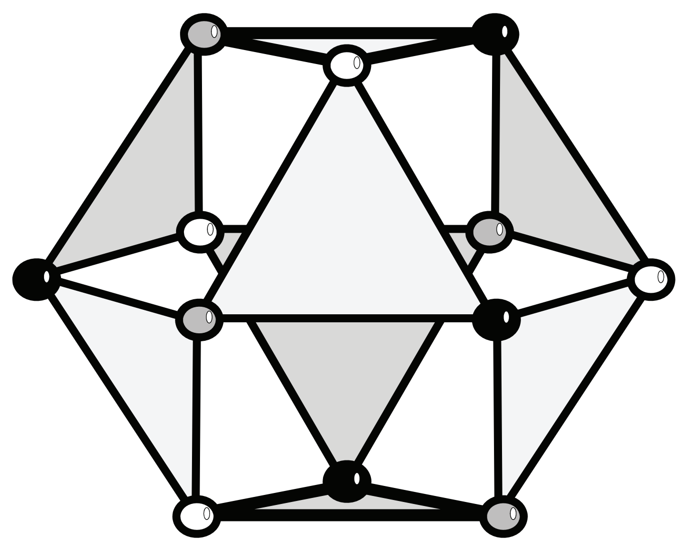


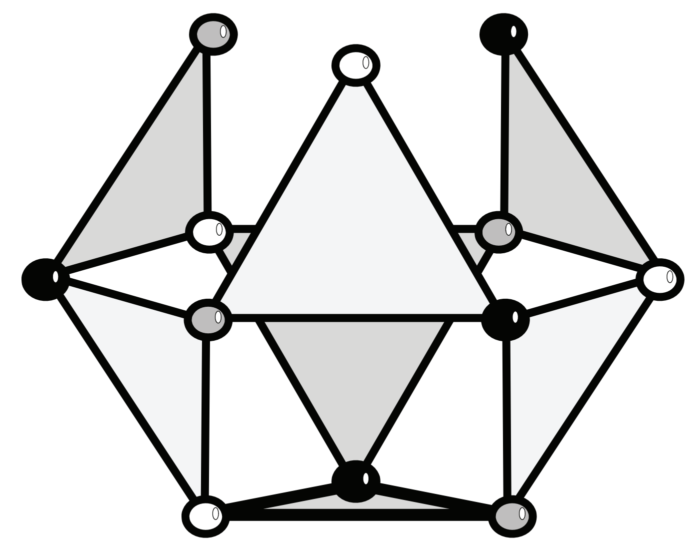
2:01 PM

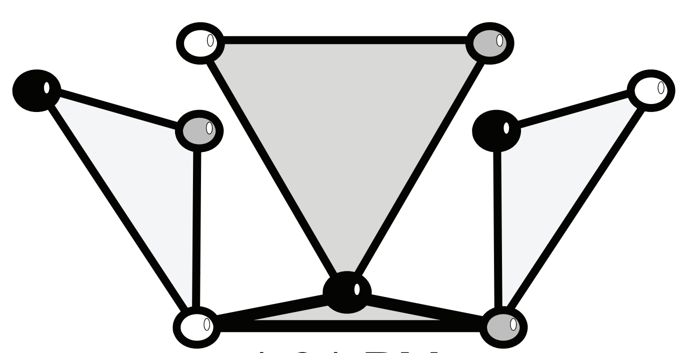


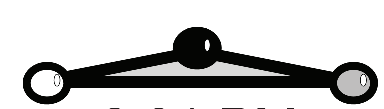
Decisions

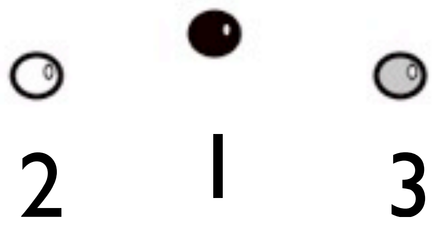
3 vertexes labeled, “cuckold”


11:59 AM

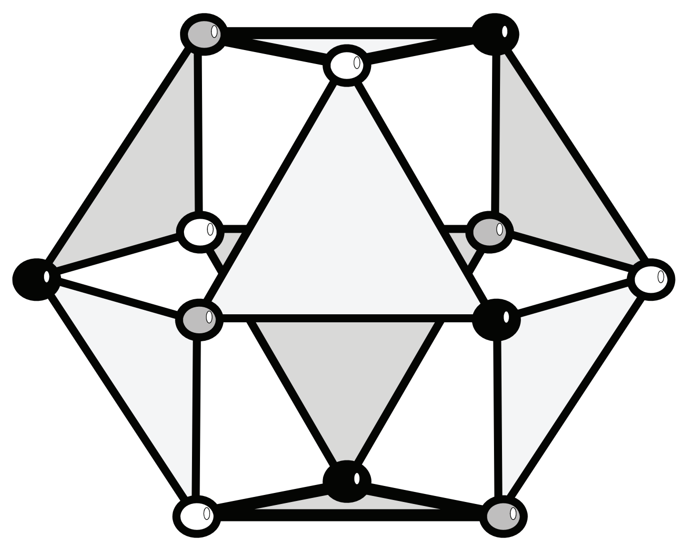

12:01 PM


1:01 PM

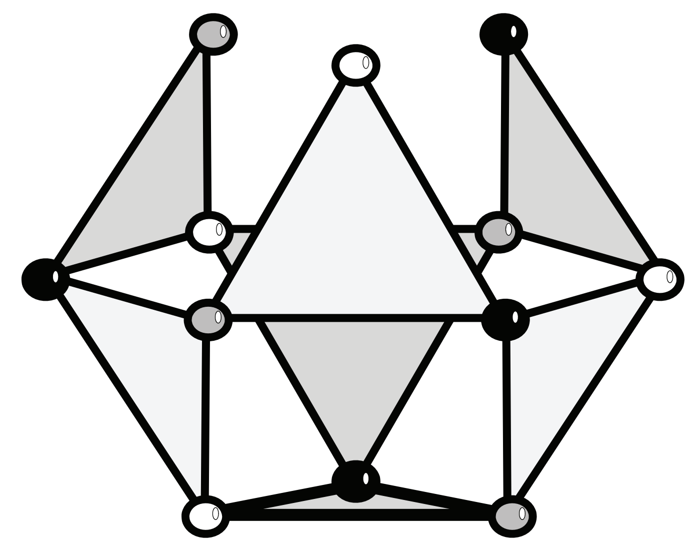

2:01 PM



Decisions

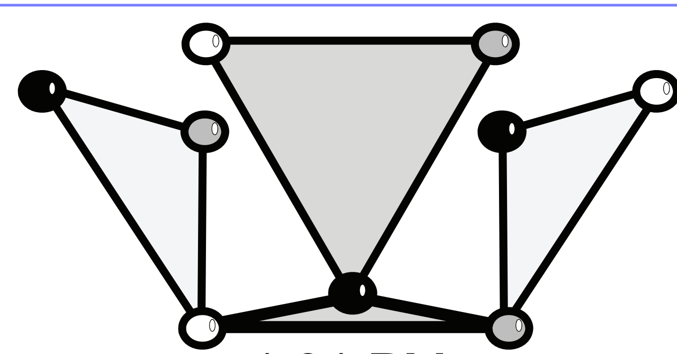


11:59 AM

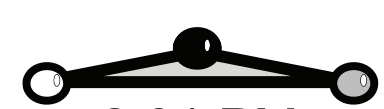


12:01 PM

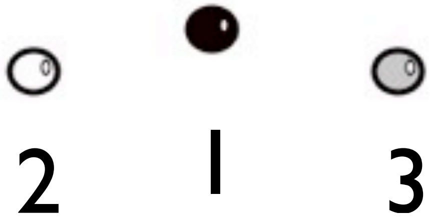
Nobody spoke previous round,
6 vertexes labeled “cuckold”



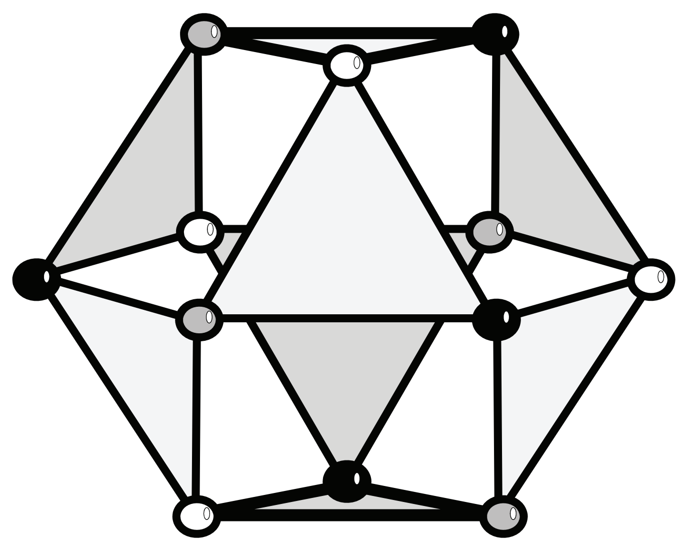
1:01 PM



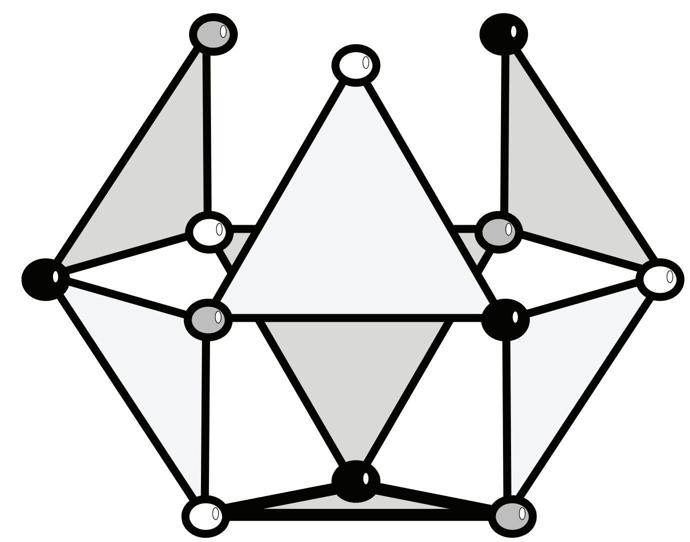
2:01 PM



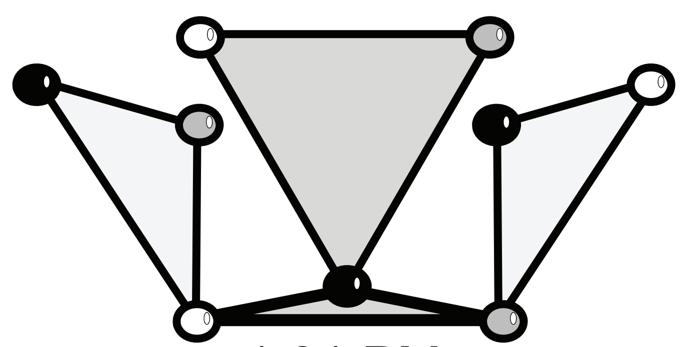
Decisions



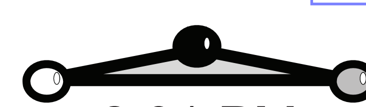
11:59 AM



12:01 PM

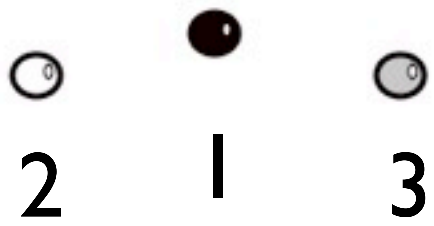


1:01 PM

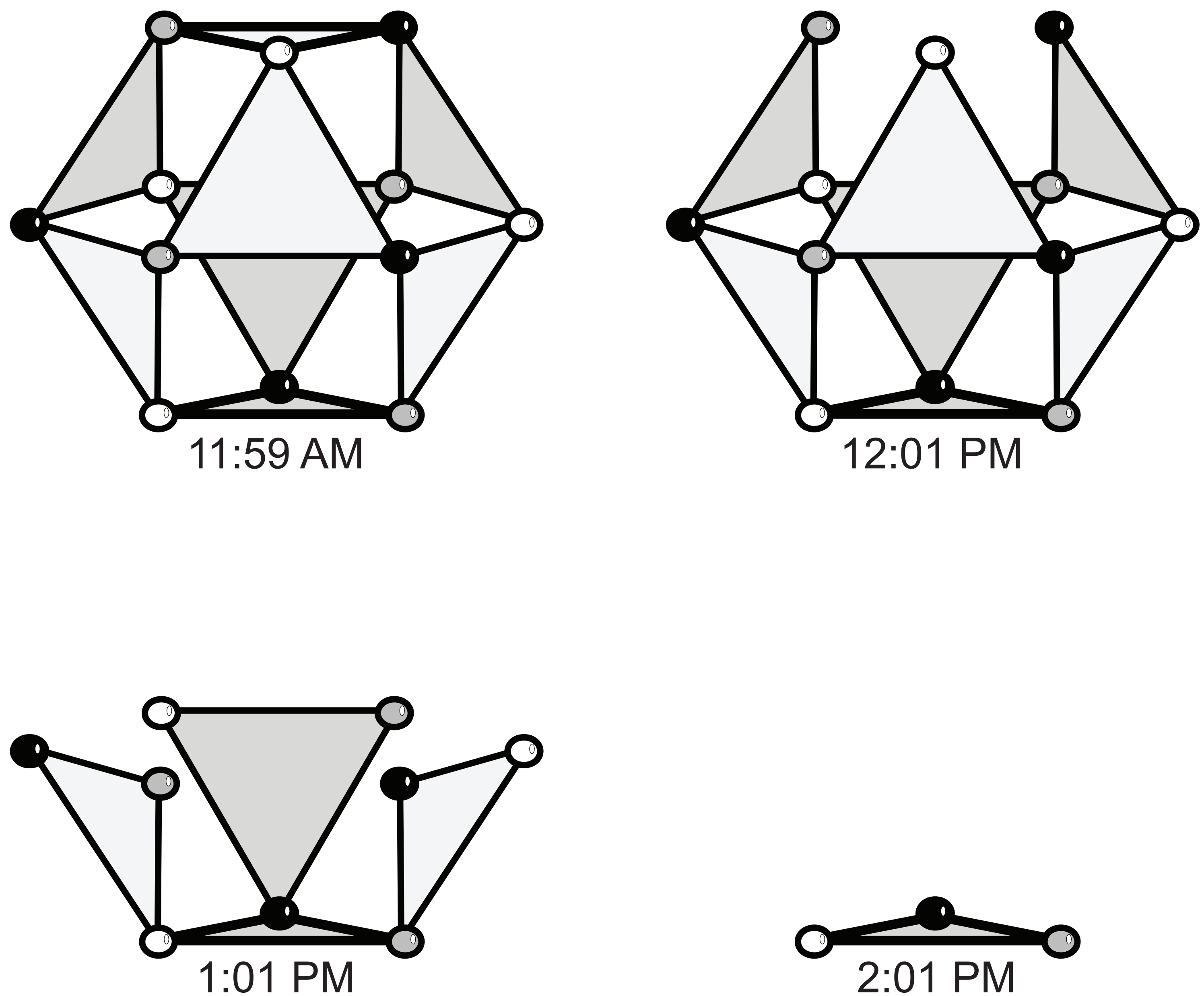


2:01 PM

3 vertexes
labeled “cuckold”

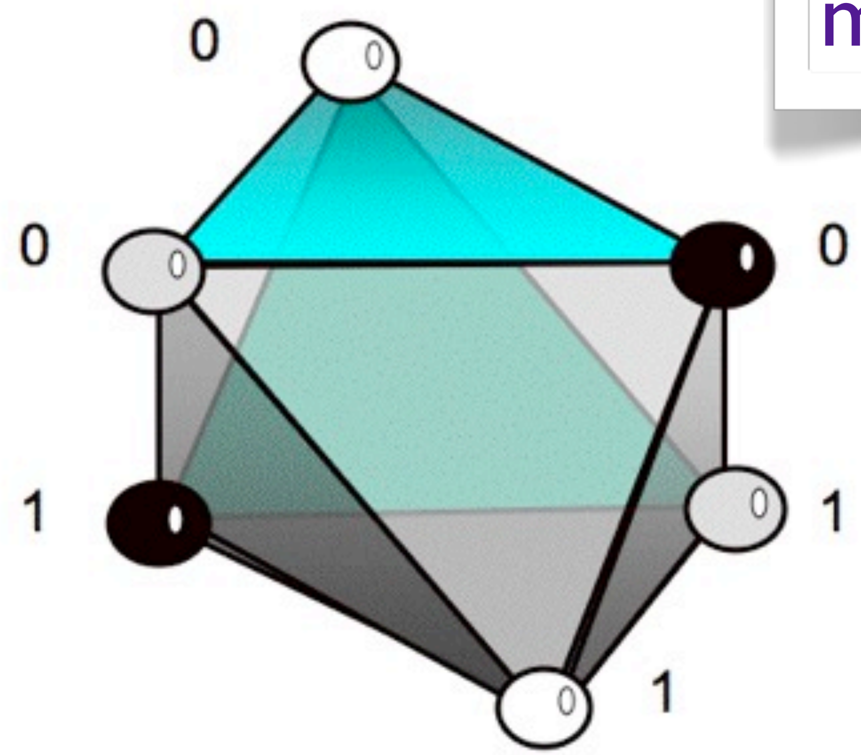


Decisions



Output complex

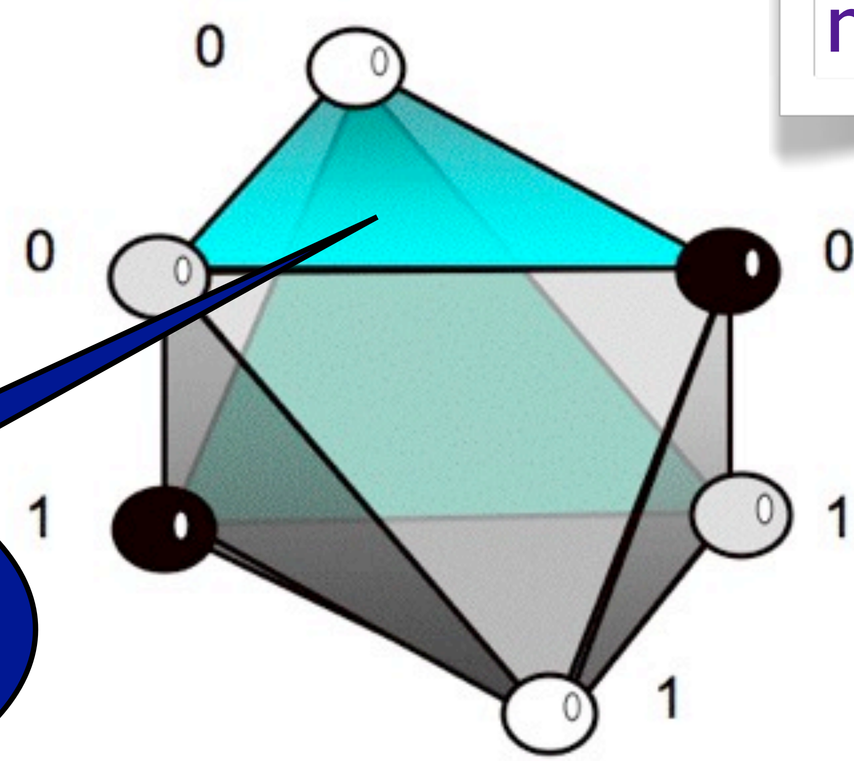
Decisions induce a map to this complex



Each man should say "yes" or "no"
All combinations are possible...

Output complex

Decisions induce a map to this complex



... except all “no” after King’s announcement

Each man should say “yes” or “no”
All combinations are possible...

Solving the cheating wives task

Each man decides an output value, on one of its local states

Decisions define a simplicial map from input complex to output complex that respects the task's specification

In this task communication is very limited. More generally, for any task...

Solving any task

In the basic, wait-free model

A task is solvable if and only if there exists a *subdivision* of the input complex and a simplicial map to the output complex that respects the task's specification

Herlihy, Shavit 1993

Wait-free: asynchronous model where any number of processes can crash

Two insecure lovers

Second story

Coordination

We often need to ensure that two things happen together or not at all.

For example, a banking system needs to ensure that if an automatic teller dispenses cash, then the corresponding account balance is debited, and vice-versa.

Two insecure lovers

Two insecure lovers

- Alice and Bob want to schedule a meeting.

Two insecure lovers

- Alice and Bob want to schedule a meeting.
- If both attend, they win, but if only one attends, defeat and humiliation is felt.

Two insecure lovers

- Alice and Bob want to schedule a meeting.
- If both attend, they win, but if only one attends, defeat and humiliation is felt.
- As a result, neither will show up without a guarantee that the other will show up at the same time.

Two insecure lovers

- Alice and Bob want to schedule a meeting.
- If both attend, they win, but if only one attends, defeat and humiliation is felt.
- As a result, neither will show up without a guarantee that the other will show up at the same time.
- Communication is be SMS only.

Communication problems

- Normally, it takes a message one hour to arrive.
- However, it is possible that it is gets lost.

The puzzle

Fortunately, on this particular night,
all the messages arrive safely.

How long will it take Alice and Bob
to coordinate their meeting?

Analysis of the puzzle

First operational, then combinatorial

Operational analysis (1)

Suppose Alice initiates the communication

Operational analysis
(1)

Operational analysis (1)

- Suppose Bob receives a message at 1:00 from Alice saying "meet at midnight". Should Bob show up?

Operational analysis (1)

- Suppose Bob receives a message at 1:00 from Alice saying "meet at midnight". Should Bob show up?
- Although her message was in fact delivered, Alice does not know. She therefore considers it possible that Bob did not receive the message.

Operational analysis (1)

- Suppose Bob receives a message at 1:00 from Alice saying "meet at midnight". Should Bob show up?
- Although her message was in fact delivered, Alice does not know. She therefore considers it possible that Bob did not receive the message.
- Hence Alice cannot decide to show up, given her current state of knowledge.

Operational analysis (1)

- Suppose Bob receives a message at 1:00 from Alice saying "meet at midnight". Should Bob show up?
- Although her message was in fact delivered, Alice does not know. She therefore considers it possible that Bob did not receive the message.
- Hence Alice cannot decide to show up, given her current state of knowledge.
- Knowing this, Bob will not show up based solely on Alice's message.

Operational analysis
(2)

Operational analysis (2)

- Naturally, Bob reacts by sending an acknowledgment back to Alice, which arrives at 2:00

Operational analysis (2)

- Naturally, Bob reacts by sending an acknowledgment back to Alice, which arrives at 2:00
- Will Alice plan to show up?

Operational analysis (2)

- Naturally, Bob reacts by sending an acknowledgment back to Alice, which arrives at 2:00
- Will Alice plan to show up?
- Unfortunately, Alice's predicament is similar to Bob's predicament at 1:00, she cannot yet decide to show up

No number of successfully delivered acknowledgments will be enough to ensure that show up safely!

The key insight is that the difficulty is not caused by what actually happens (all messages actually arrive) but by the uncertainty regarding what might have happened.

Combinatorial analysis

Combinatorial analysis

- Initially Alice has two possible decisions: meet at dawn, or meet at noon the next day.

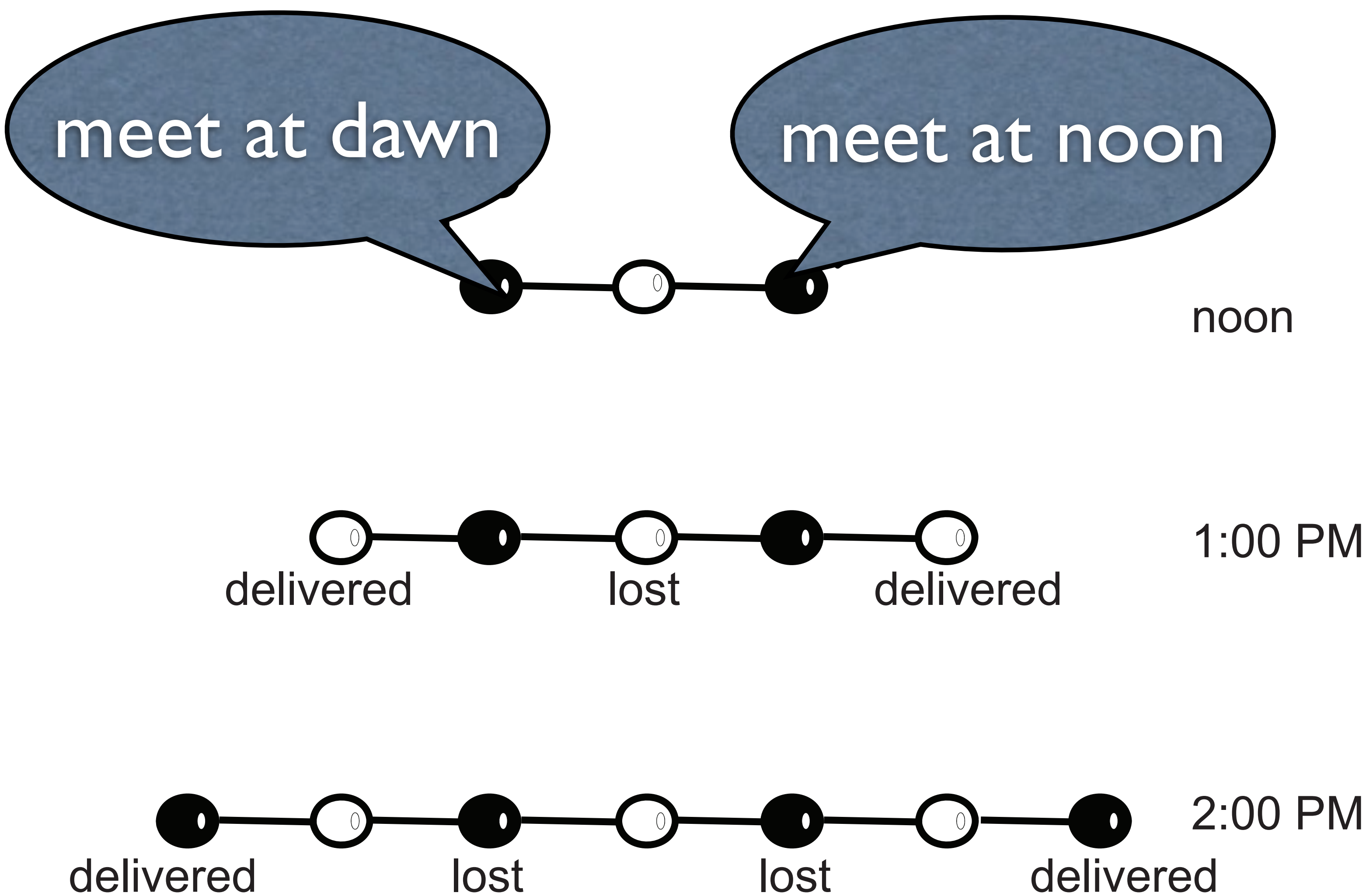
Combinatorial analysis

- Initially Alice has two possible decisions: meet at dawn, or meet at noon the next day.
- Bob has only one initial state, the white vertex in the middle, waiting to hear Alice's preference.

Combinatorial analysis

- Initially Alice has two possible decisions: meet at dawn, or meet at noon the next day.
- Bob has only one initial state, the white vertex in the middle, waiting to hear Alice's preference.
- This vertex belongs to two edges (simplexes)

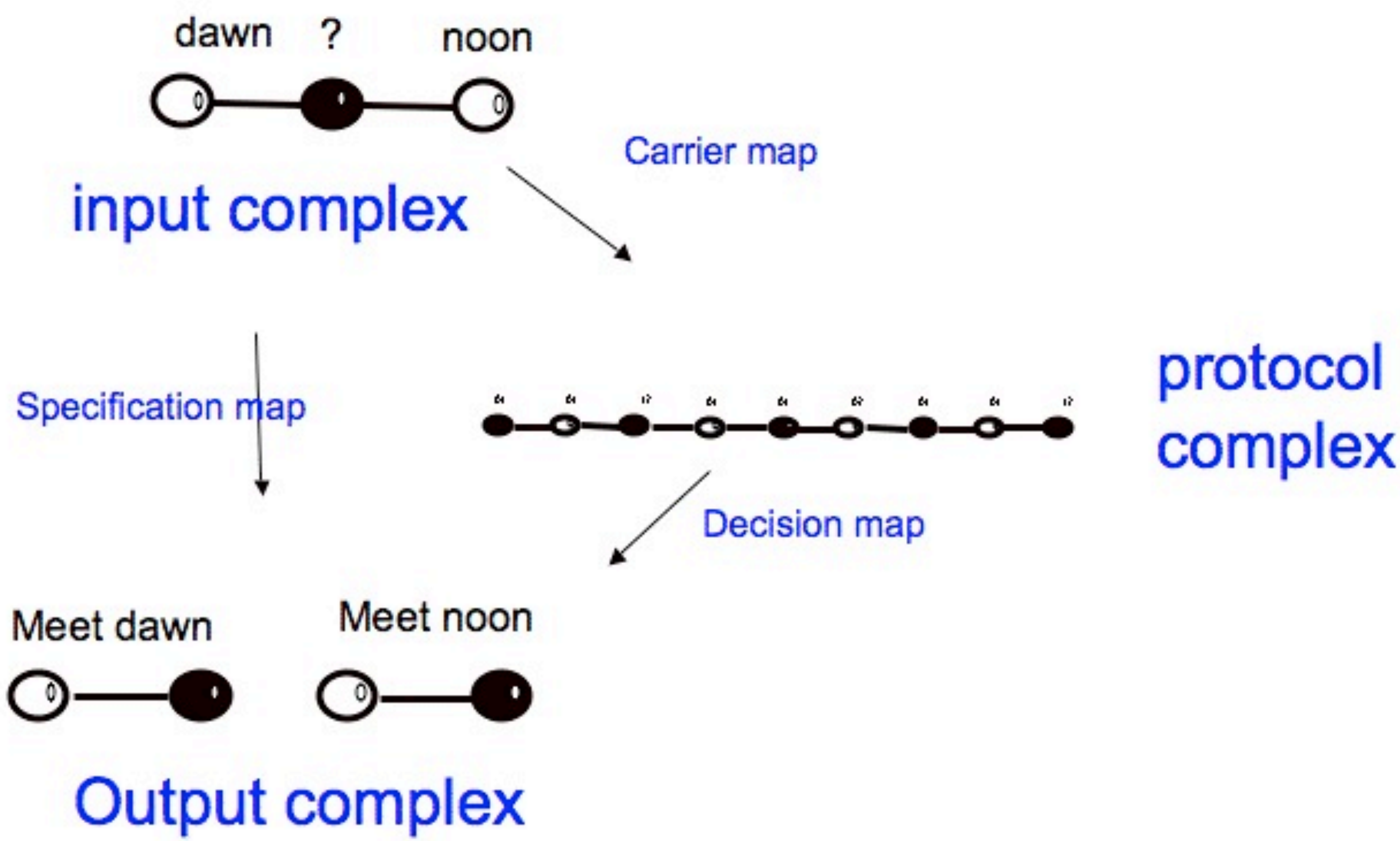
Evolution


meet at dawn
meet at noon
noon
1:00 PM
delivered
lost
delivered
2:00 PM
delivered
lost
lost
delivered

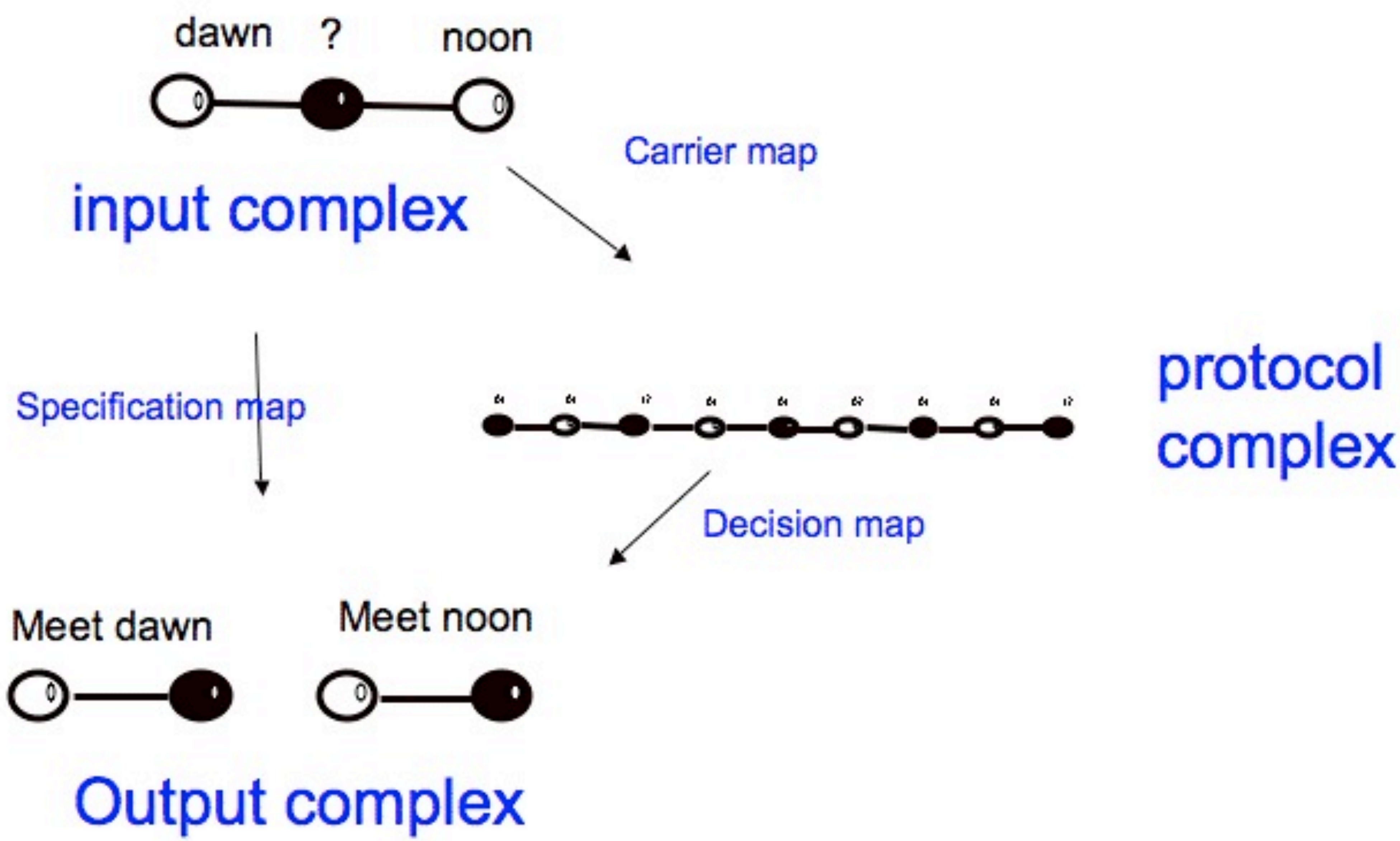
Topology implies impossibility

No number of successfully delivered acknowledgments will be enough to ensure that show up safely, because the complex is subdivided, and remains connected!

No number of successfully delivered acknowledgments will be enough to ensure that show up safely!



Because not possible to map a connected input complex into a disconnected output complex



Conclusions
(1)

- A Turing machine that computes a *function* gets one input and produces one output. A distributed algorithm that solves a *task*, each process gets one part of the input (doesn't know what inputs others got), and after communication, computes one part of the output

- A distributed computing model is parametrized by the failures and asynchrony that can occur. These parameters determine the tasks that can be solved, and at what cost.

Conclusions
(2)

- The *nature* of sequential computability and complexity has to do with Turing machines, while distributed computability and complexity in the presence of failures and timing uncertainties is of a topological nature
- The two notions are *orthogonal*. In distributed computing each sequential process is not restricted to be a Turing machine

End