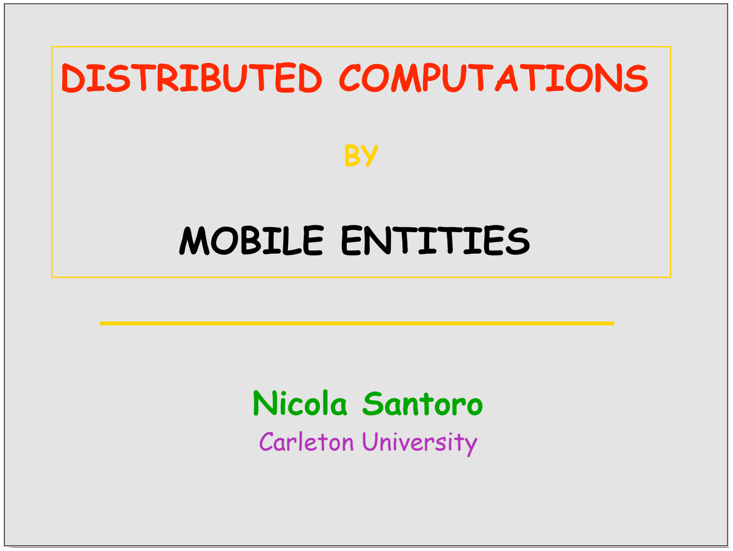


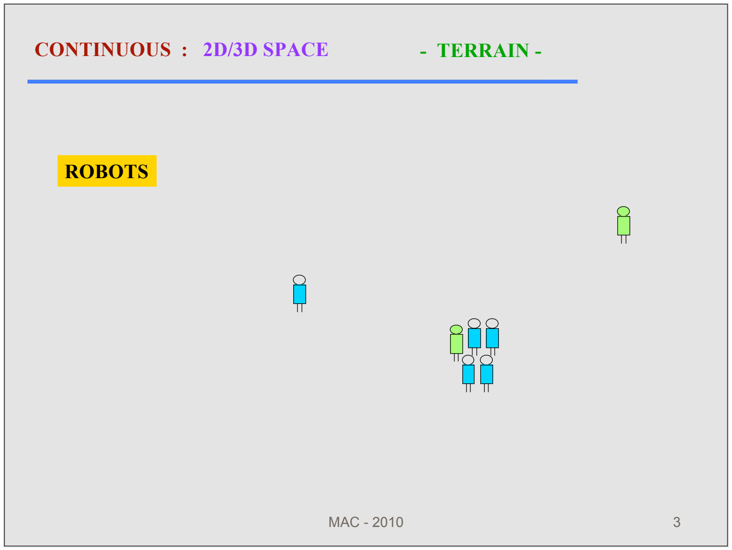
DISTRIBUTED COMPUTATIONS

BY

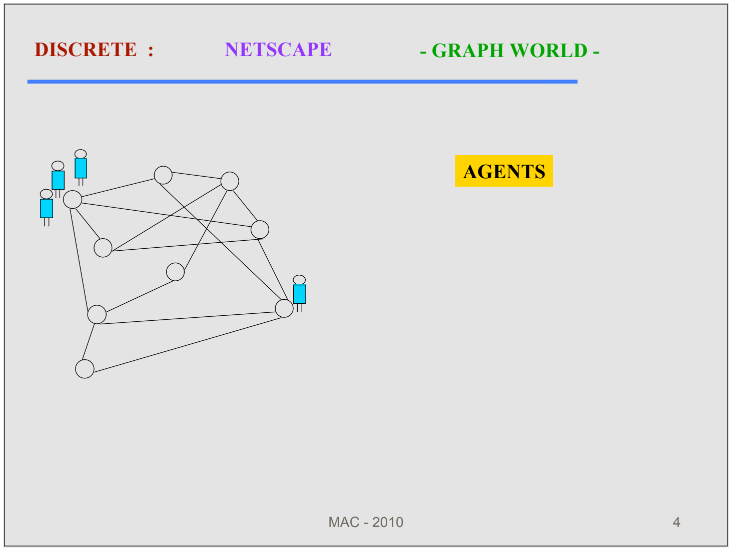
MOBILE ENTITIES

Nicola Santoro
Carleton University

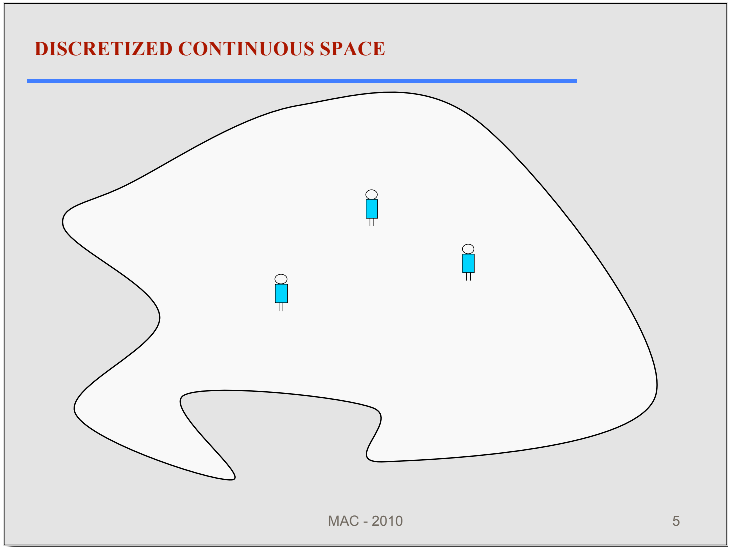
COMPUTATIONAL MOBILE ENTITIES



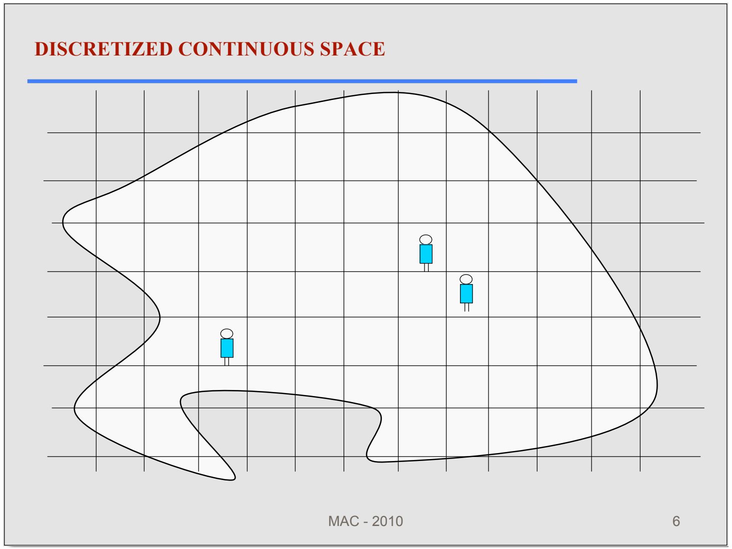
CONTINUOUS : 2D/3D SPACE - TERRAIN -



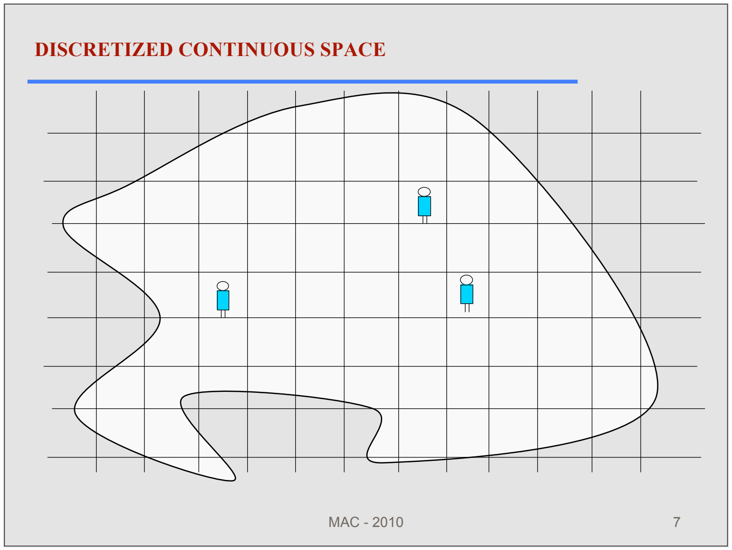
DISCRETE : NETSCAPE - GRAPH WORLD -



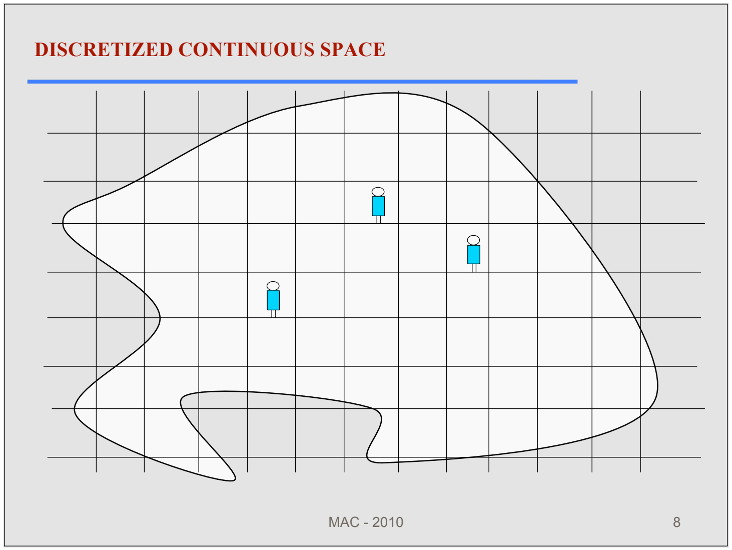
DISCRETIZED CONTINUOUS SPACE



DISCRETIZED CONTINUOUS SPACE

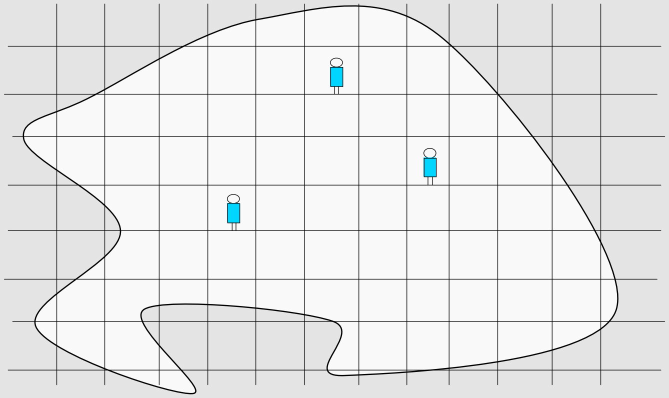


DISCRETIZED CONTINUOUS SPACE

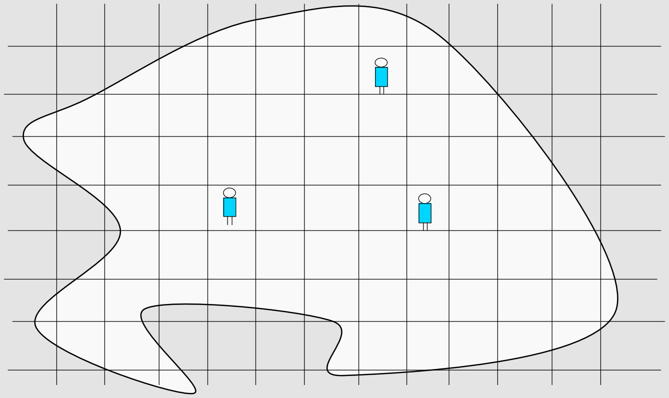


DISCRETIZED CONTINUOUS SPACE

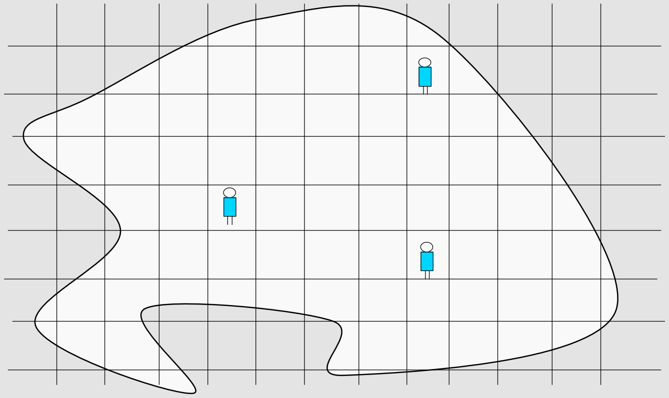
DISCRETIZED CONTINUOUS SPACE



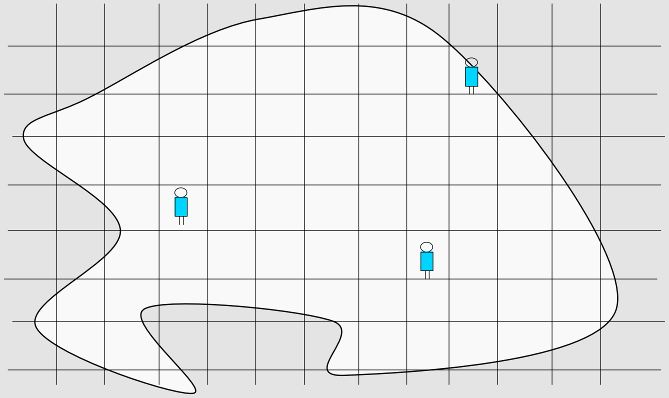
DISCRETIZED CONTINUOUS SPACE

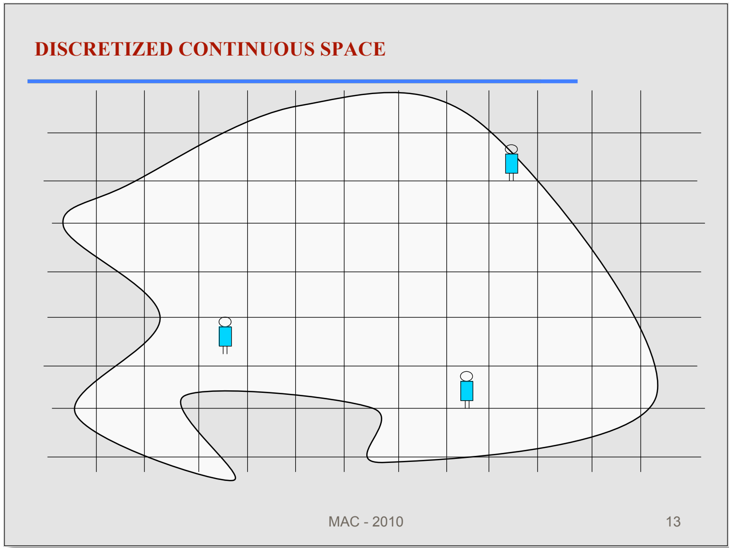


DISCRETIZED CONTINUOUS SPACE

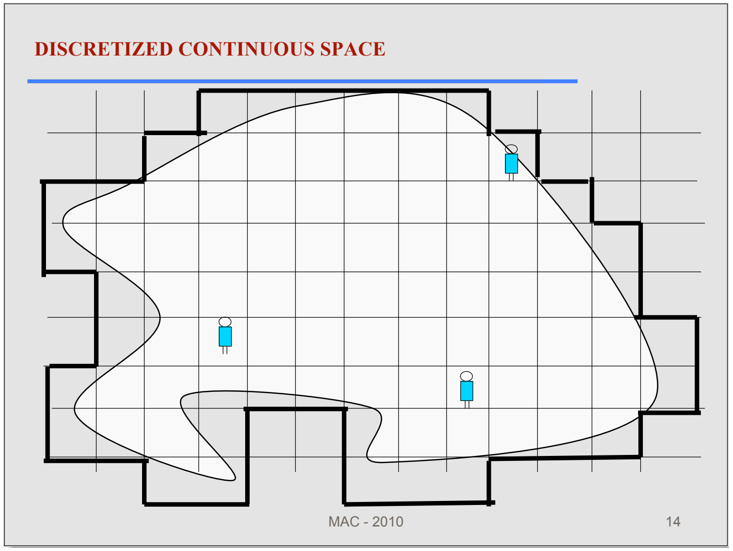


DISCRETIZED CONTINUOUS SPACE

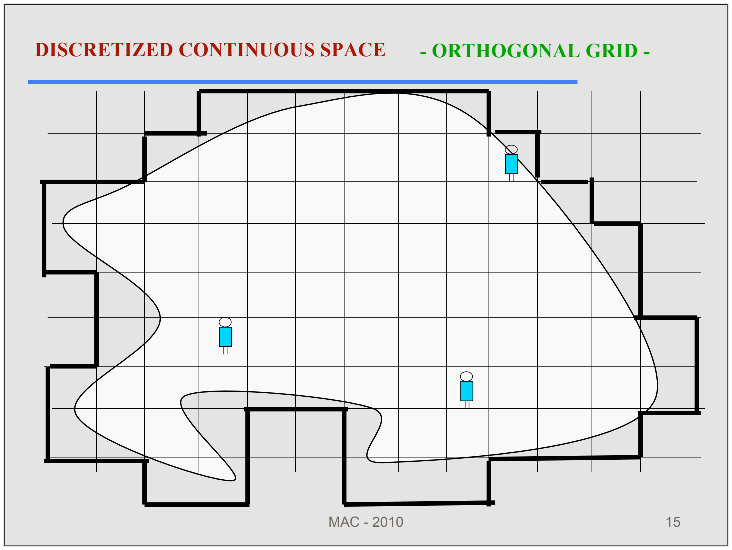




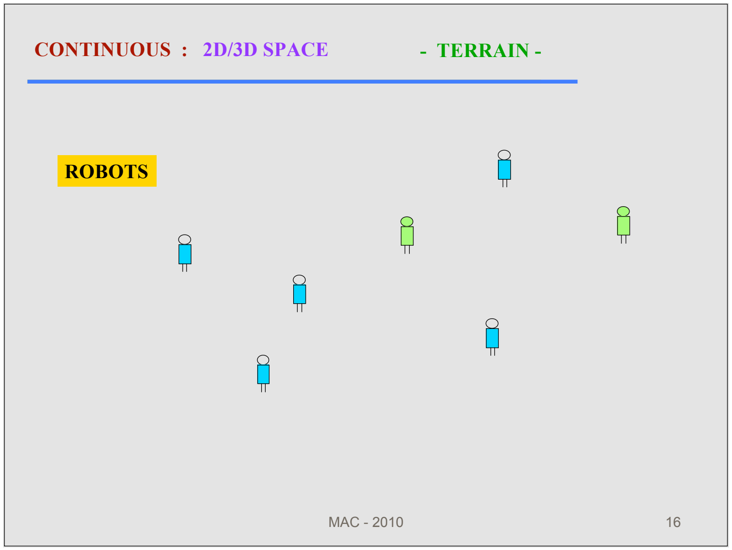
DISCRETIZED CONTINUOUS SPACE



DISCRETIZED CONTINUOUS SPACE

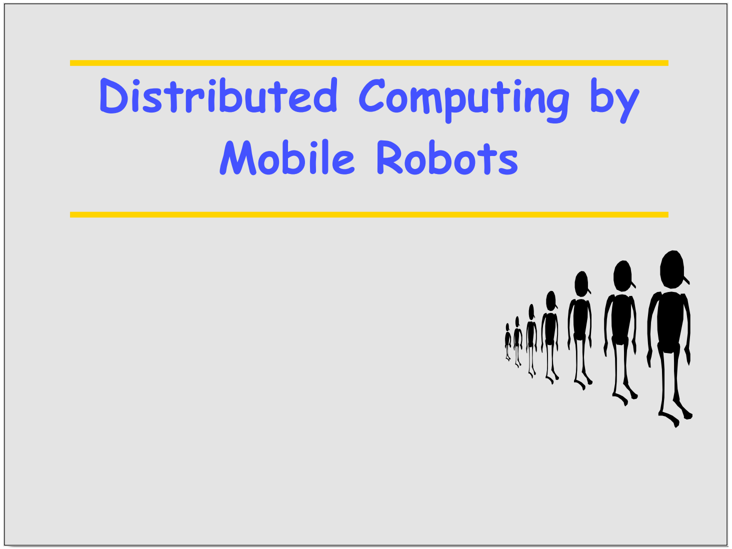


DISCRETIZED CONTINUOUS SPACE - ORTHOGONAL GRID -

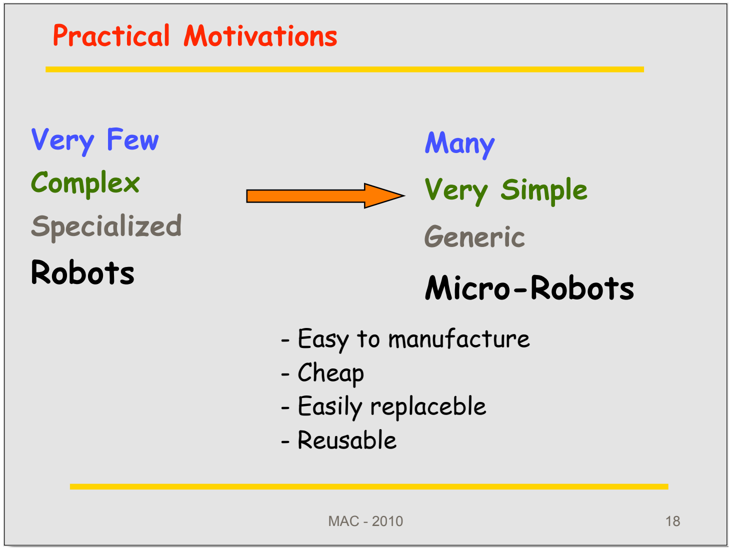


CONTINUOUS : 2D/3D SPACE - TERRAIN -

ROBOTS



Distributed Computing by Mobile Robots

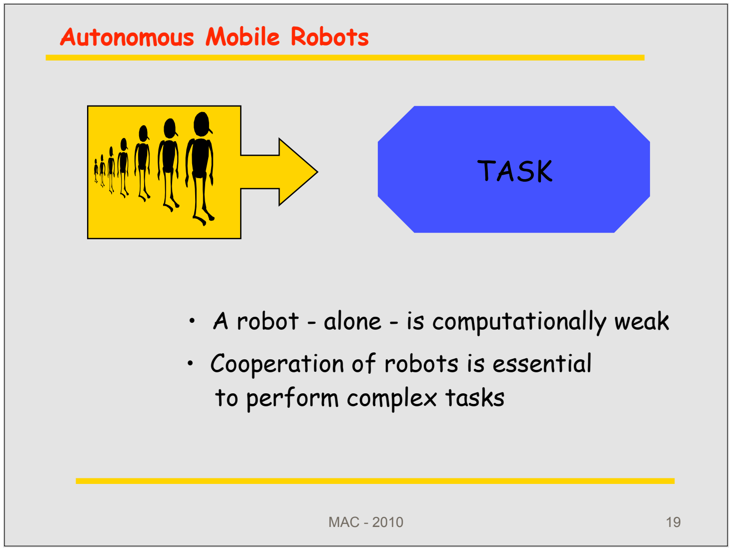


Practical Motivations

Very Few
Complex
Specialized
Robots

Many
Very Simple
Generic
Micro-Robots

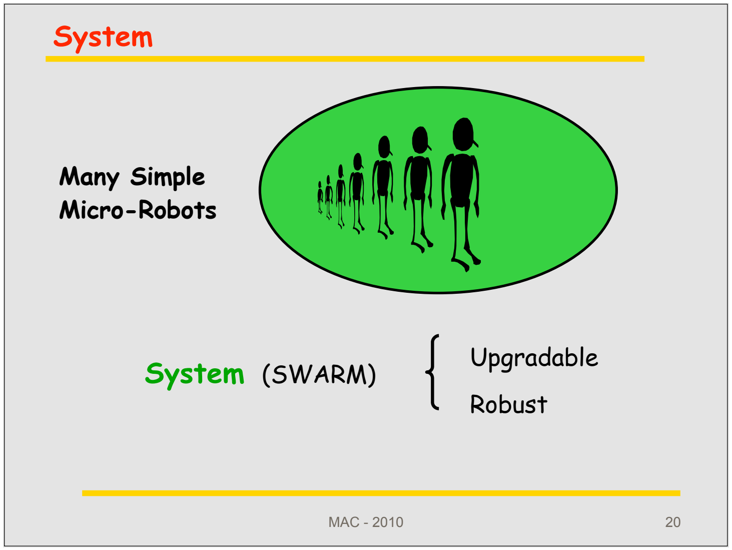
- Easy to manufacture
- Cheap
- Easily replaceble
- Reusable



Autonomous Mobile Robots

TASK

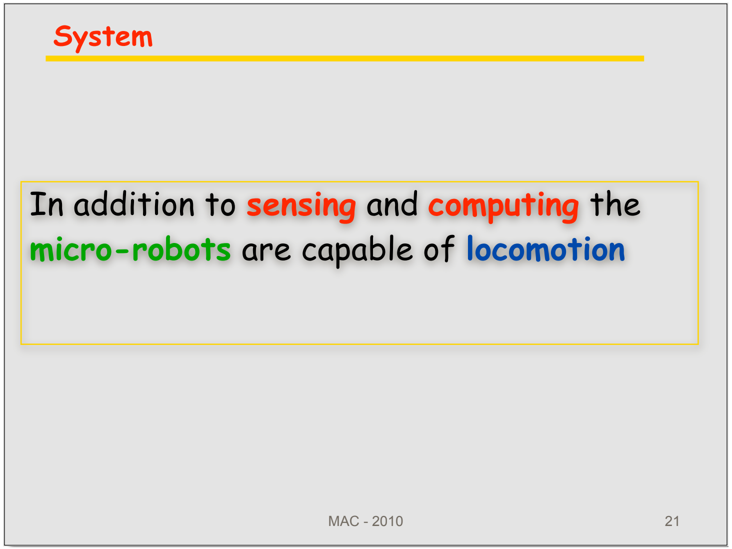
- A robot - alone - is computationally weak
- Cooperation of robots is essential to perform complex tasks



System

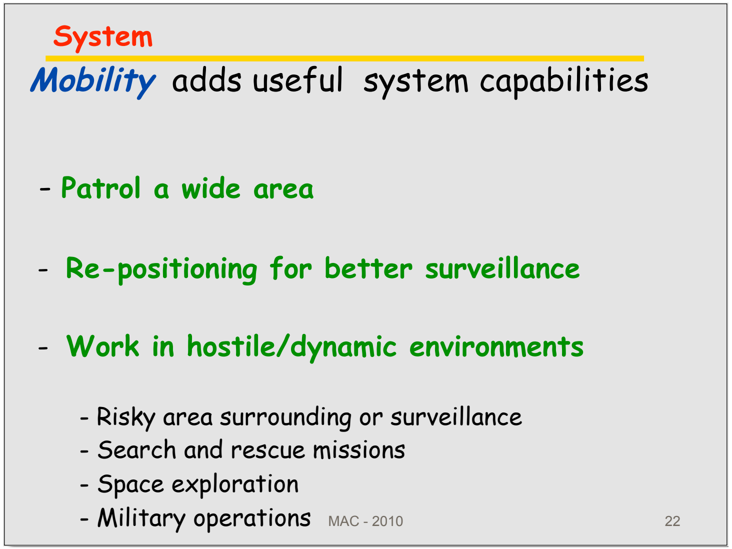
Many Simple Micro-Robots

System (SWARM) { Upgradable
Robust



System

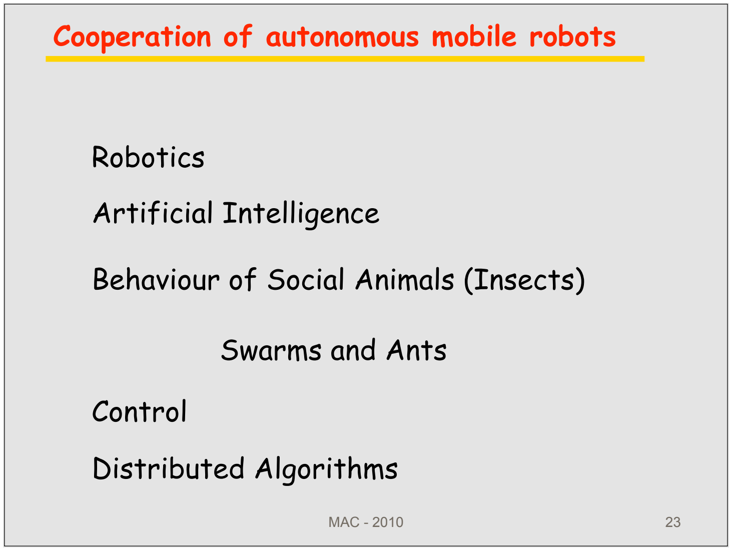
In addition to sensing and computing the micro-robots are capable of locomotion



System

Mobility adds useful system capabilities

- Patrol a wide area
- Re-positioning for better surveillance
- Work in hostile/dynamic environments
  - Risky area surrounding or surveillance
  - Search and rescue missions
  - Space exploration
  - Military operations



Cooperation of autonomous mobile robots

Robotics

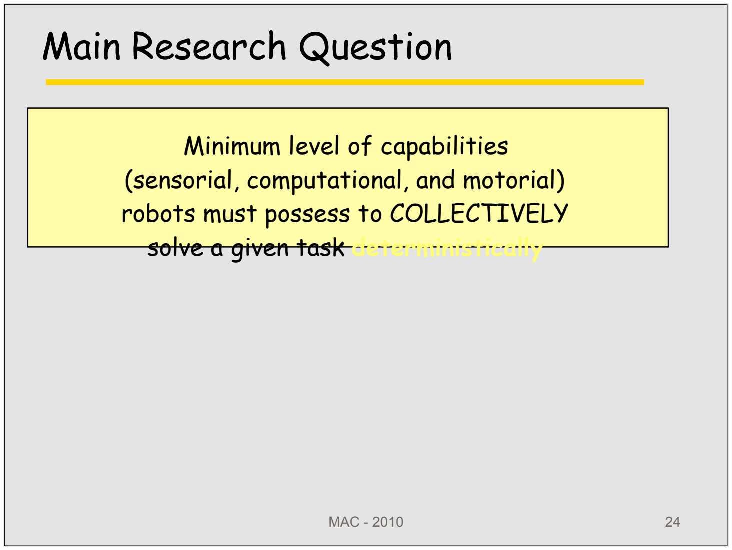
Artificial Intelligence

Behaviour of Social Animals (Insects)

Swarms and Ants

Control

Distributed Algorithms



Main Research Question

Minimum level of capabilities
(sensorial, computational, and motorial)
robots must possess to COLLECTIVELY
solve a given task deterministically

Main Research Question

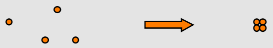
Minimum level of capabilities
(sensorial, computational, and motorial)
robots must possess to COLLECTIVELY
solve a given task **deterministically**

How <span>weak</span> can each single robot and still
globally accomplish the given task ?

How much **local power** is necessary to perform
global computations ?

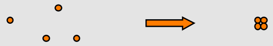
Basic Coordination Tasks

Gathering

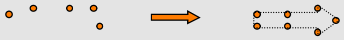


Basic Coordination Tasks

Gathering

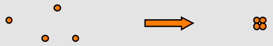


Specific Patterns

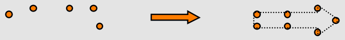


Basic Coordination Tasks

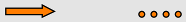
Gathering



Specific Patterns



Alignment



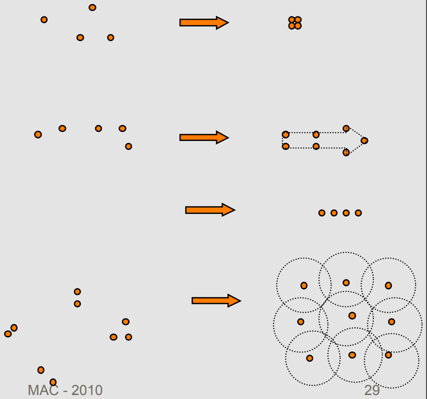
Basic Coordination Tasks

Gathering

Specific Patterns

Alignment

Scattering



Complex Tasks

Mine Sweeping

Hazardous Retrieval

Rescue Operation

COMPUTATIONAL MODEL

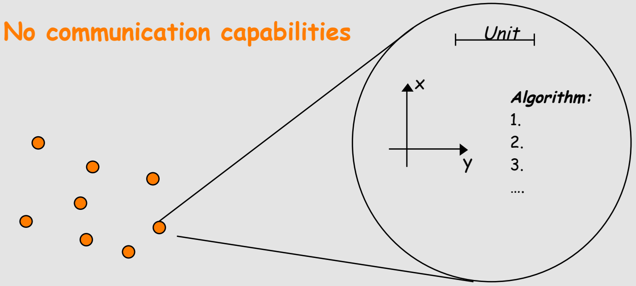
Weak Mobile Robots

Autonomous (no central control)

Homogeneous (run same software)

Anonymous (indistinguishable)

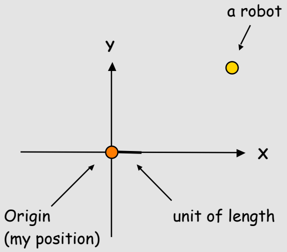
No communication capabilities



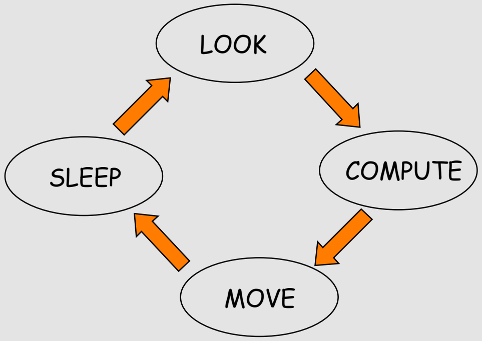
The robots are points.

Robot's Basic Capabilities

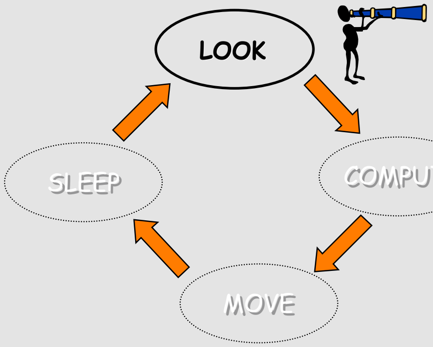
- **Processing and Storage**
  - limited
  - execute **same protocol**
- **Sensorial**
  - "see" other robots
  - local coordinate system
- **Motorial**
  - move towards a destination



Robot's behaviour : Life Cycle



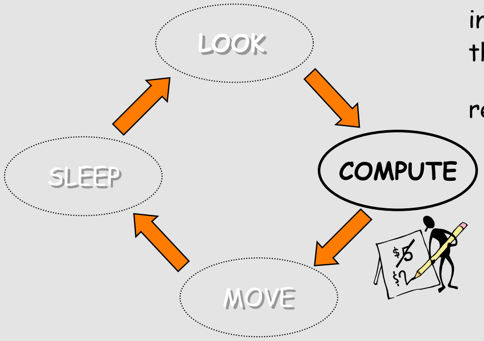
Robot's behaviour : Life Cycle



Uses its sensors to observe the world.

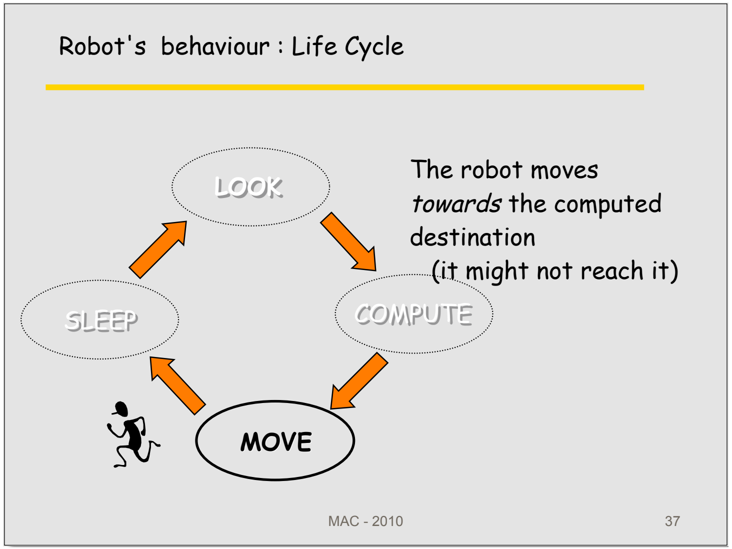
result = **SNAPSHOT** of the world

Robot's behaviour : Life Cycle



input = position of the other robots

result = destination point



Robot's behaviour : Life Cycle

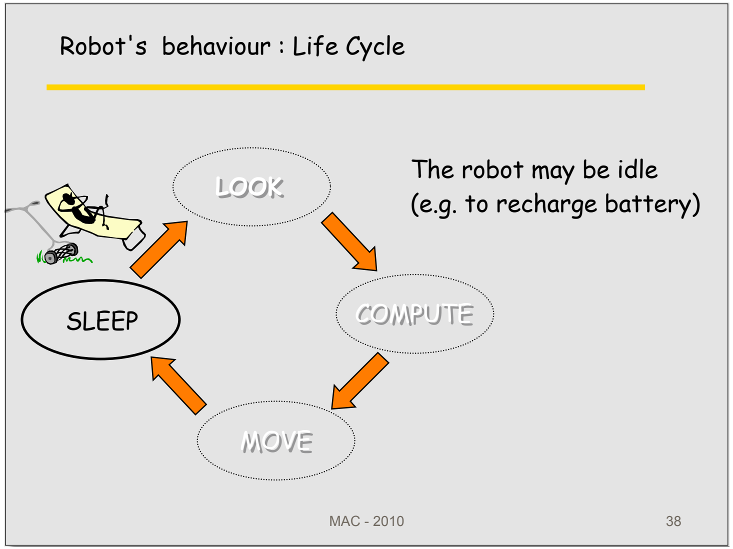
LOOK

COMPUTE

MOVE

SLEEP

The robot moves *towards* the computed destination
(it might not reach it)



Robot's behaviour : Life Cycle

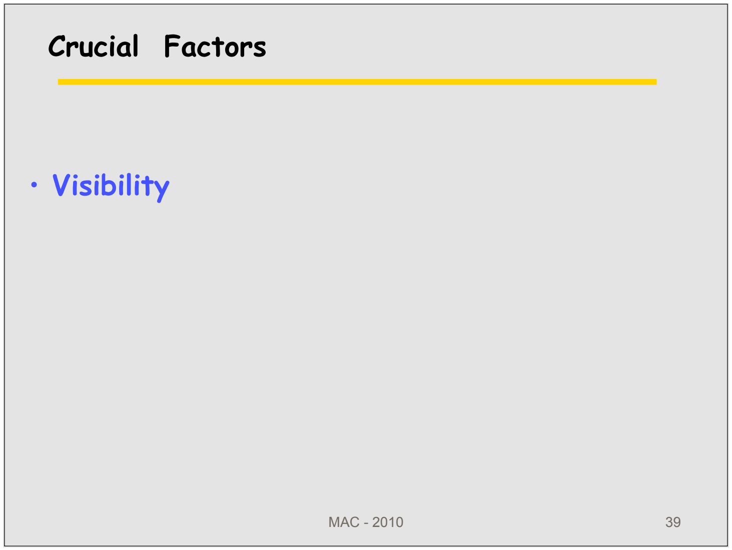
LOOK

COMPUTE

MOVE

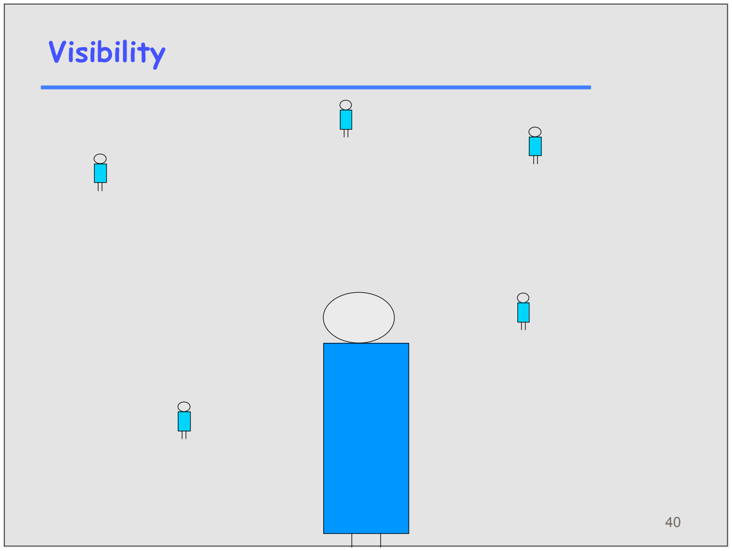
SLEEP

The robot may be idle
(e.g. to recharge battery)

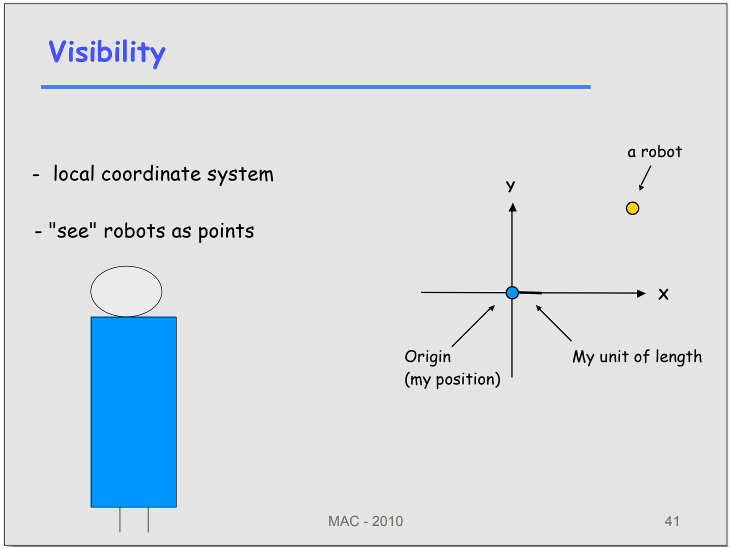


Crucial Factors

- Visibility



Visibility



Visibility

- local coordinate system

- "see" robots as points

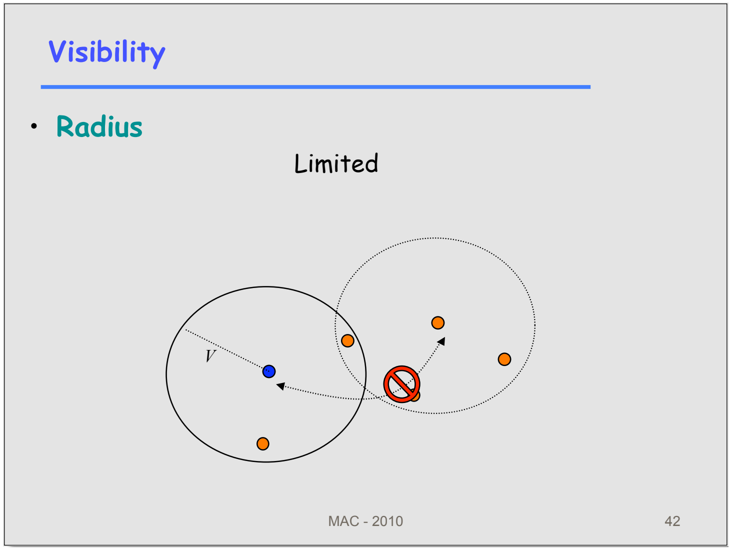
a robot

y

x

Origin
(my position)

My unit of length

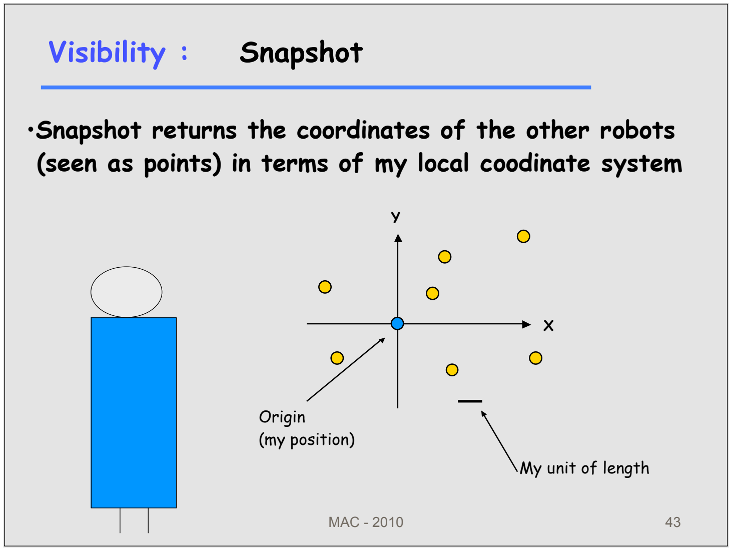


Visibility

- **Radius**

Limited

V



Visibility : Snapshot

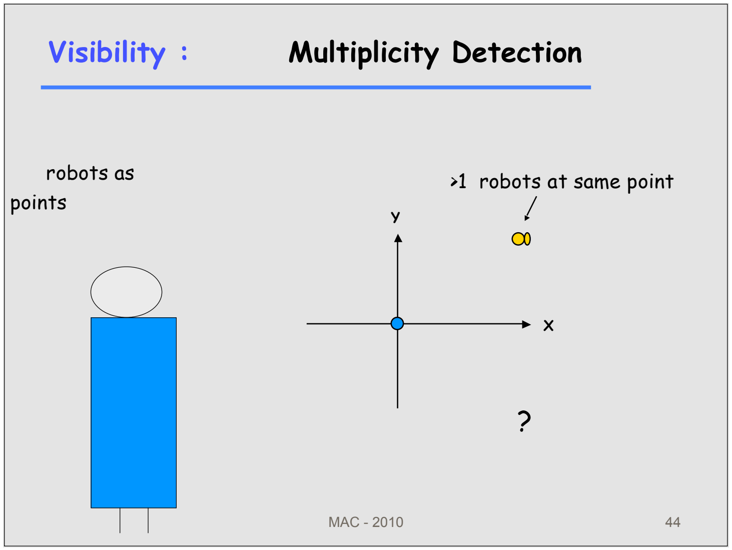
- **Snapshot returns the coordinates of the other robots (seen as points) in terms of my local coodinate system**

y

x

Origin
(my position)

My unit of length



Visibility : Multiplicity Detection

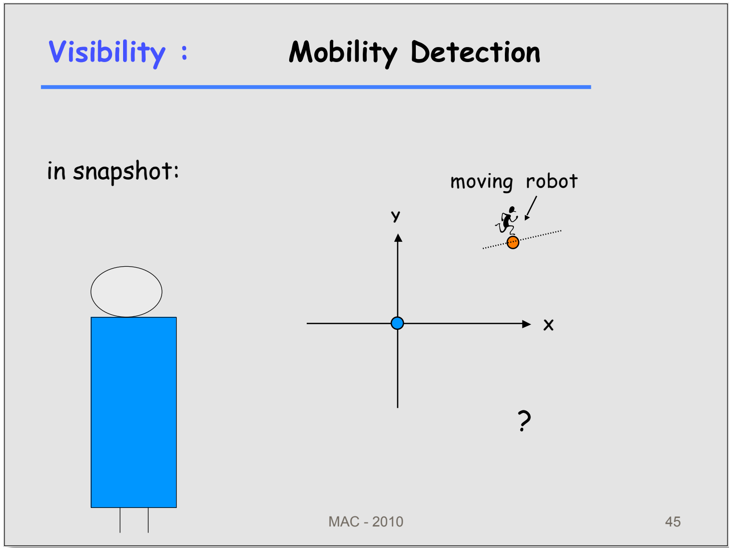
robots as
points

>1 robots at same point

y

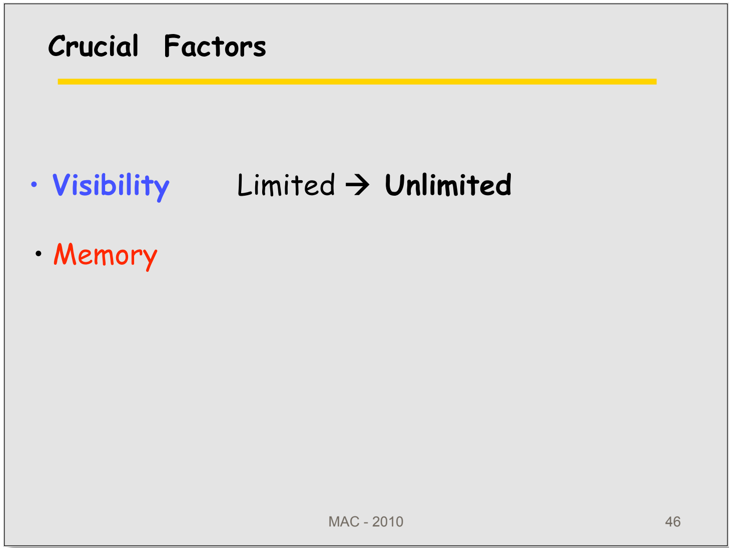
x

?



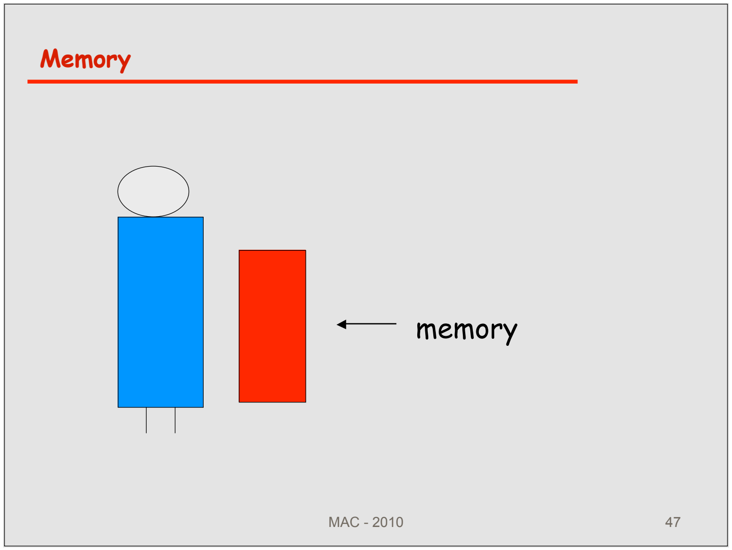
Visibility : Mobility Detection

in snapshot:

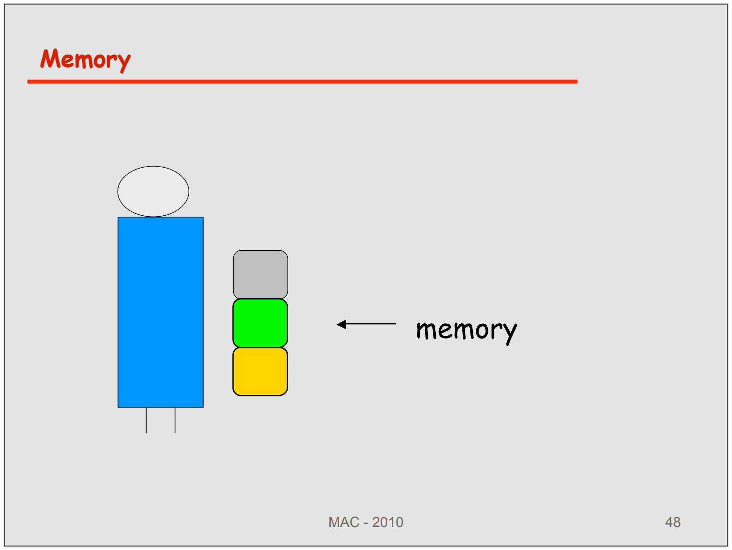


Crucial Factors

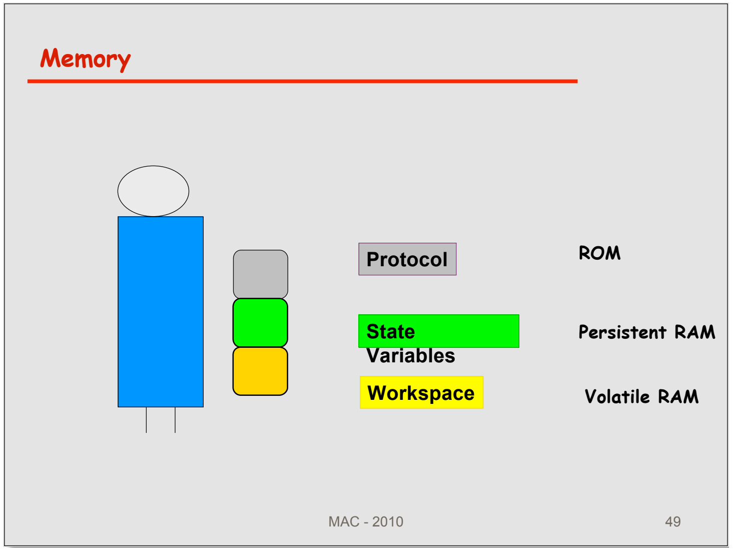
- Visibility Limited → **Unlimited**
- Memory



Memory



Memory

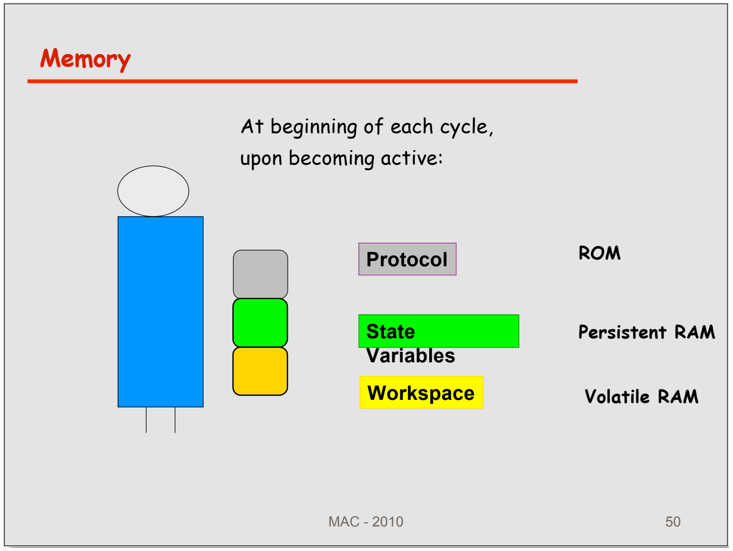


Memory

Protocol ROM

State Variables Persistent RAM

Workspace Volatile RAM



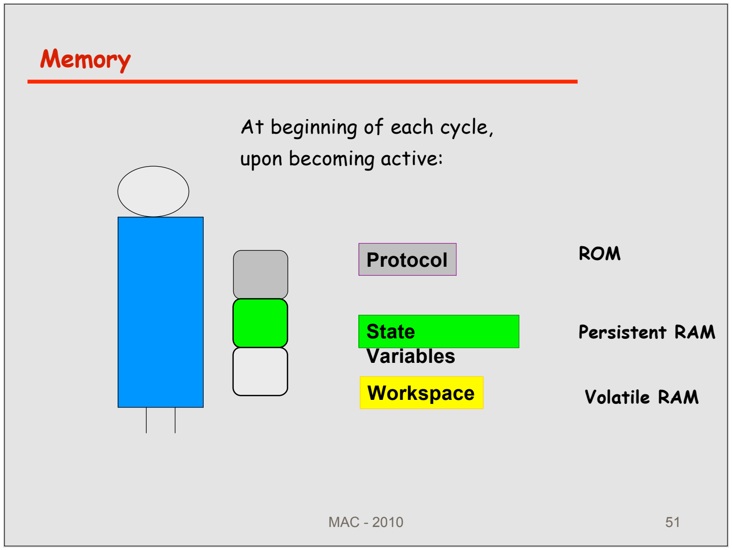
Memory

At beginning of each cycle,
upon becoming active:

Protocol ROM

State Variables Persistent RAM

Workspace Volatile RAM



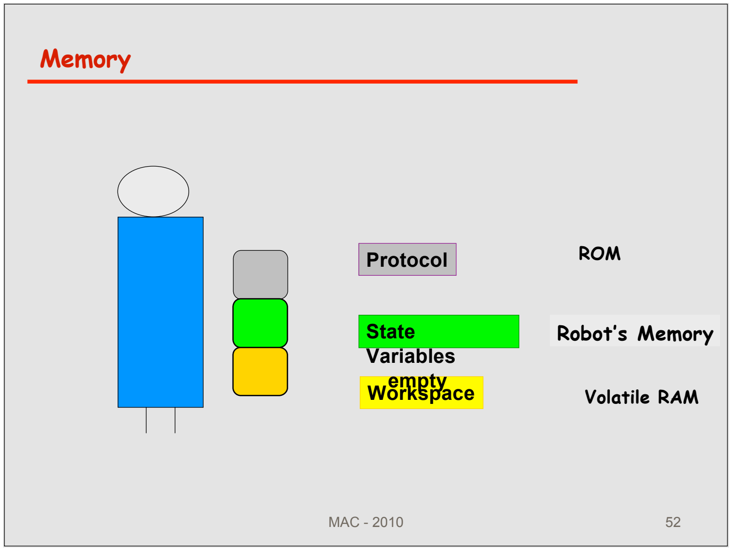
Memory

At beginning of each cycle,
upon becoming active:

Protocol ROM

State Variables Persistent RAM

Workspace Volatile RAM

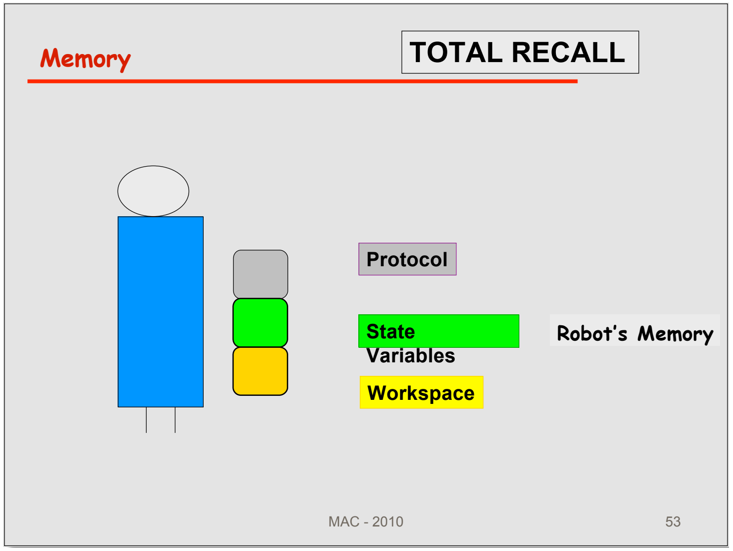


Memory

Protocol ROM

State Variables Robot's Memory

empty
Workspace Volatile RAM



Memory

TOTAL RECALL

Protocol

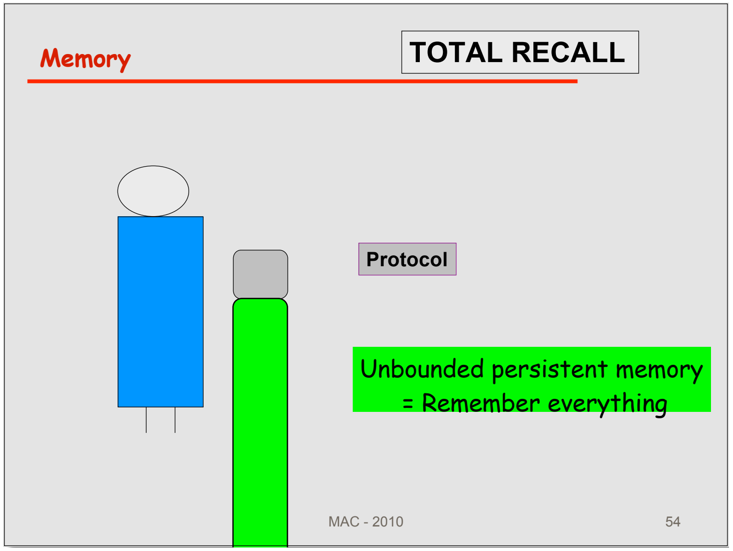
State Variables

Robot's Memory

Workspace

MAC - 2010

53



Memory

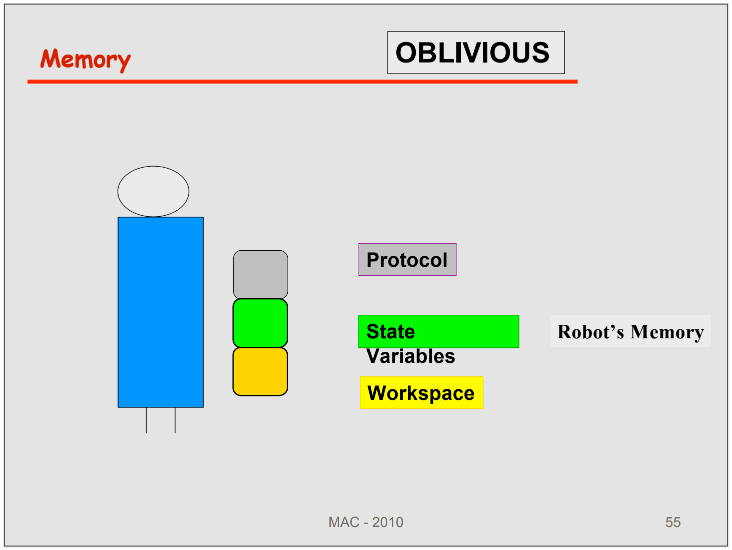
TOTAL RECALL

Protocol

Unbounded persistent memory
= Remember everything

MAC - 2010

54



Memory

OBLIVIOUS

Protocol

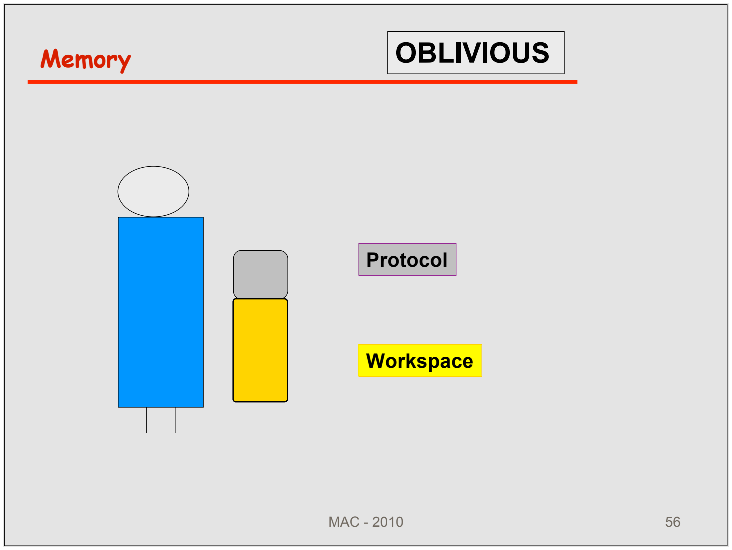
State Variables

Robot's Memory

Workspace

MAC - 2010

55



Memory

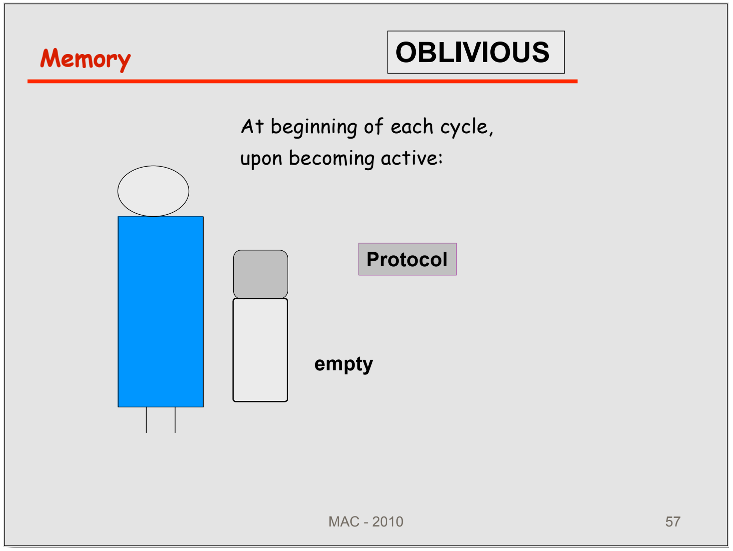
OBLIVIOUS

Protocol

Workspace

MAC - 2010

56



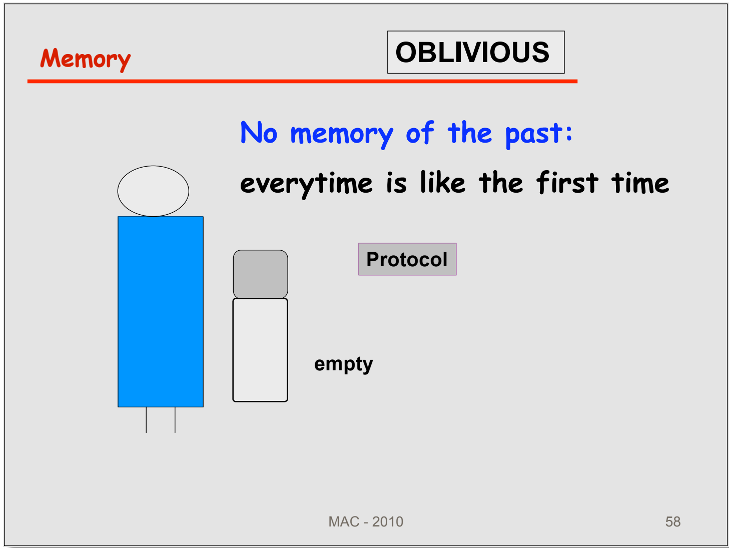
Memory **OBLIVIOUS**

At beginning of each cycle,
upon becoming active:

Protocol

empty

MAC - 2010 57



Memory **OBLIVIOUS**

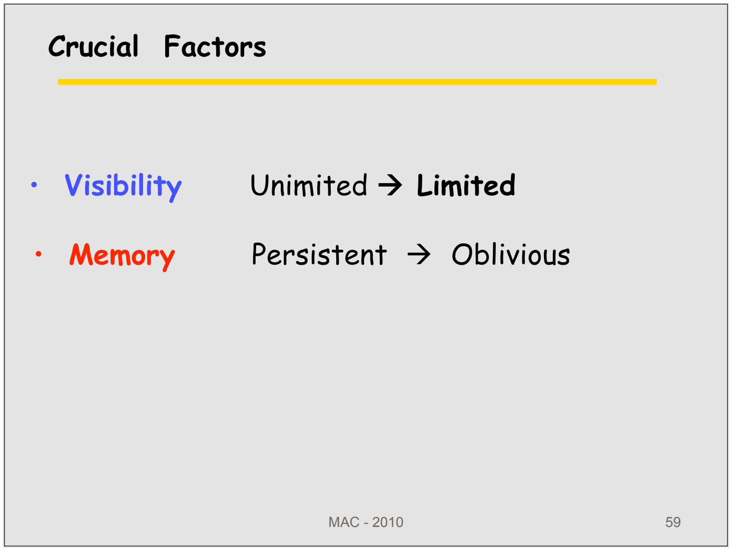
No memory of the past:

everytime is like the first time

Protocol

empty

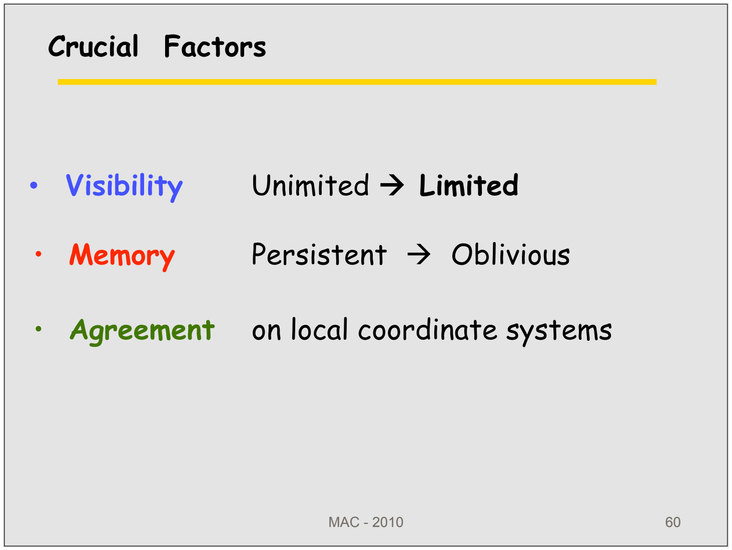
MAC - 2010 58



Crucial Factors

- Visibility Unimited → **Limited**
- Memory Persistent → Oblivious

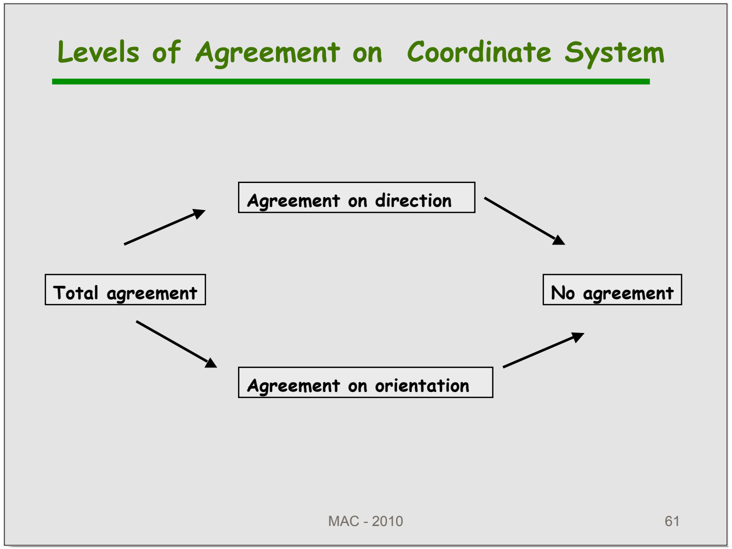
MAC - 2010 59



Crucial Factors

- Visibility Unimited → **Limited**
- Memory Persistent → Oblivious
- Agreement on local coordinate systems

MAC - 2010 60



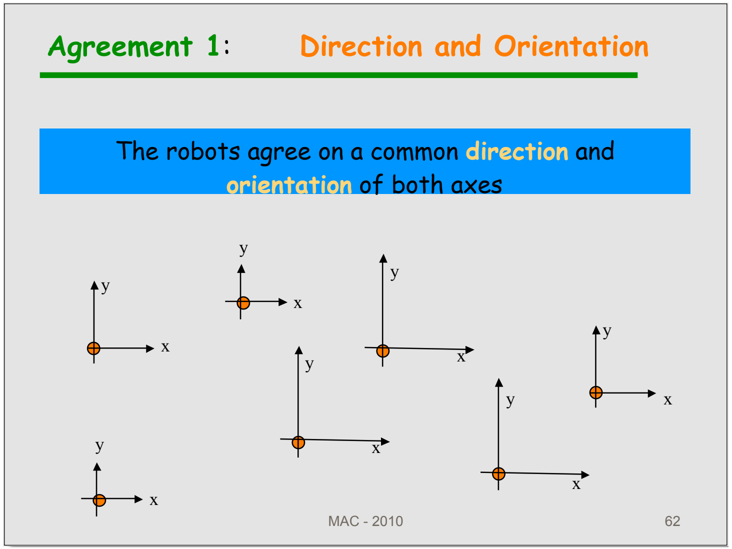
Levels of Agreement on Coordinate System

Agreement on direction

Total agreement

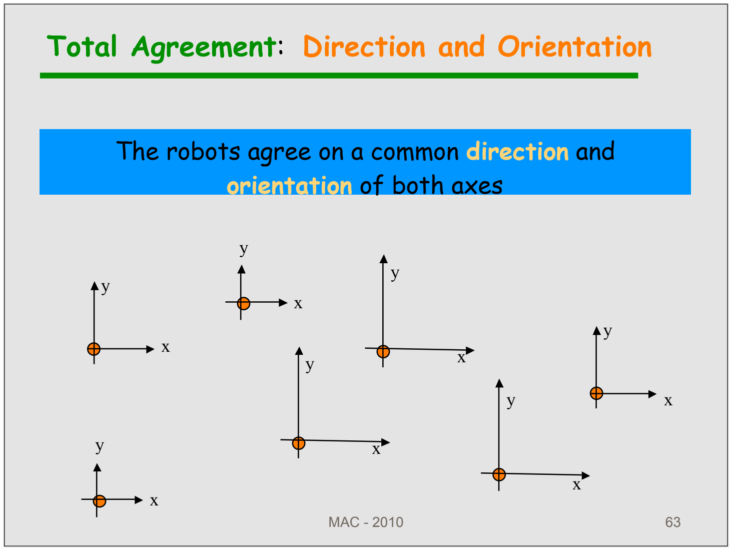
No agreement

Agreement on orientation



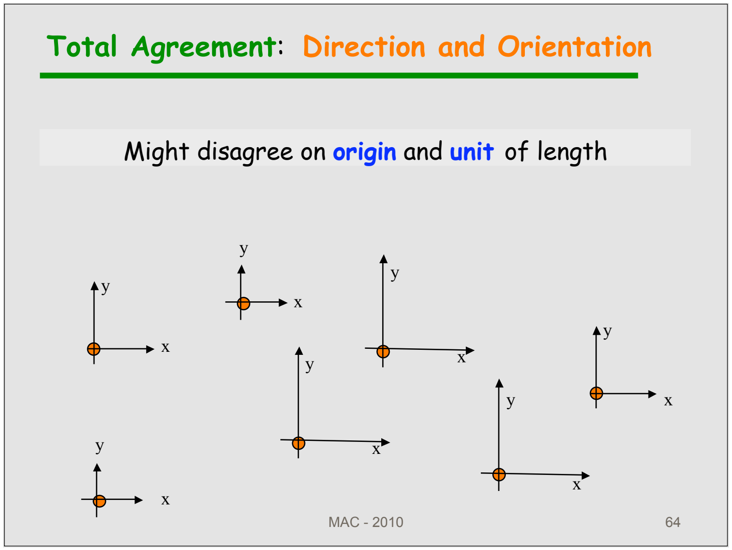
Agreement 1: Direction and Orientation

The robots agree on a common **direction** and **orientation** of both axes



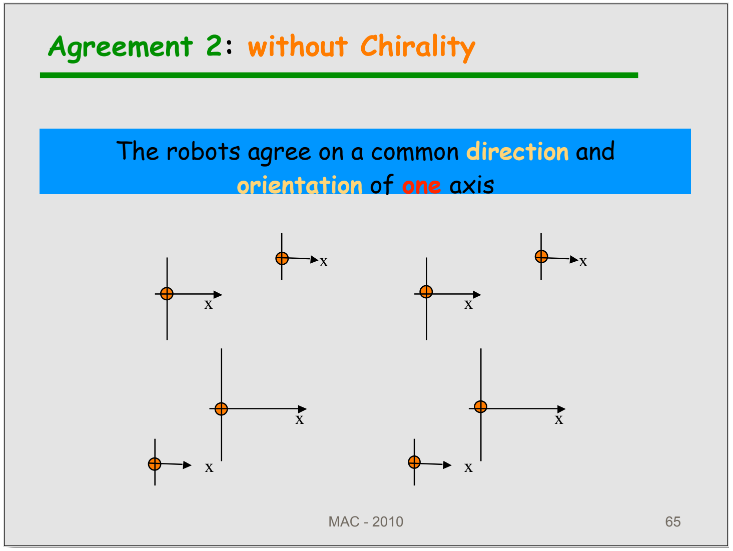
Total Agreement: Direction and Orientation

The robots agree on a common **direction** and **orientation** of both axes



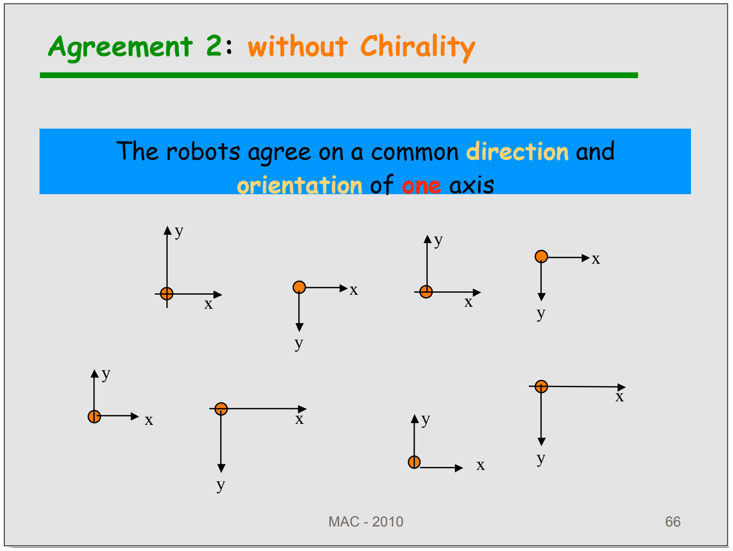
Total Agreement: Direction and Orientation

Might disagree on **origin** and **unit** of length



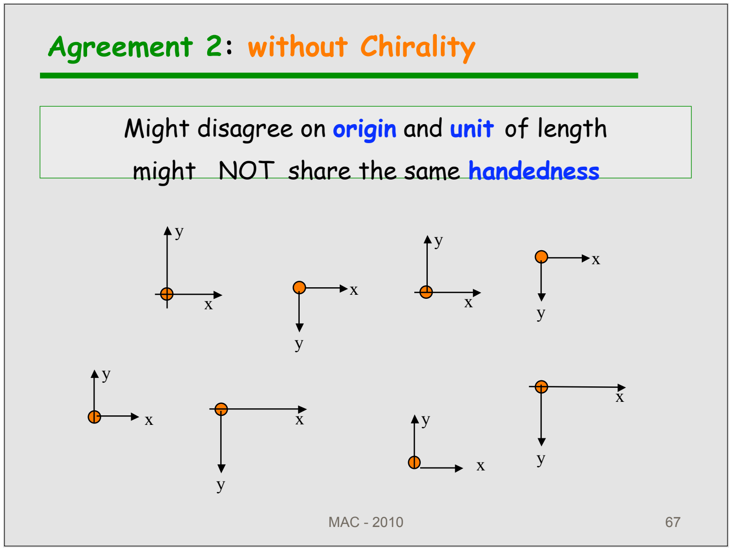
Agreement 2: without Chirality

The robots agree on a common direction and orientation of one axis



Agreement 2: without Chirality

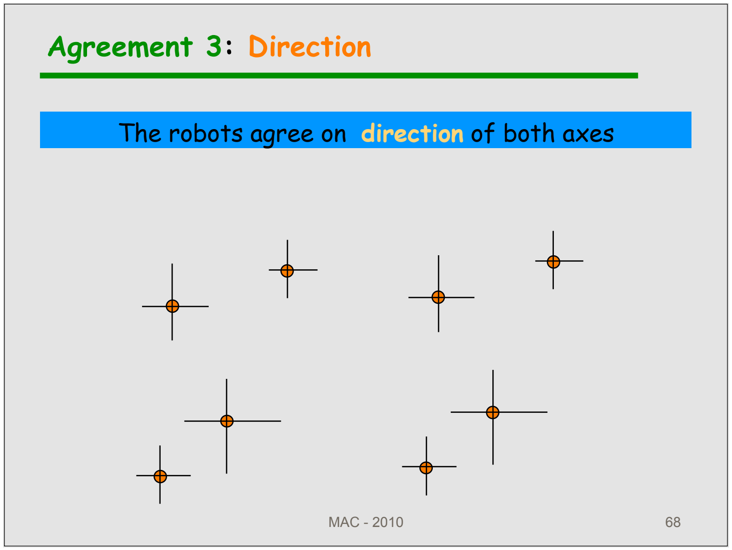
The robots agree on a common direction and orientation of one axis



Agreement 2: without Chirality

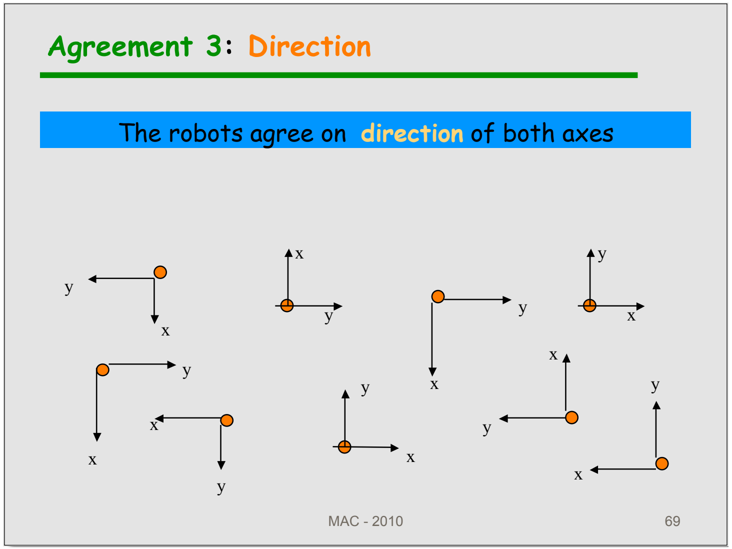
Might disagree on origin and unit of length

might NOT share the same handedness



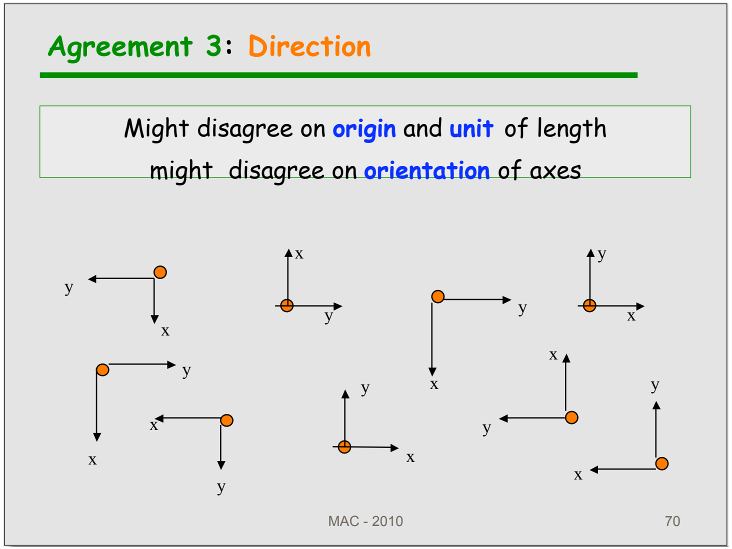
Agreement 3: Direction

The robots agree on direction of both axes



Agreement 3: Direction

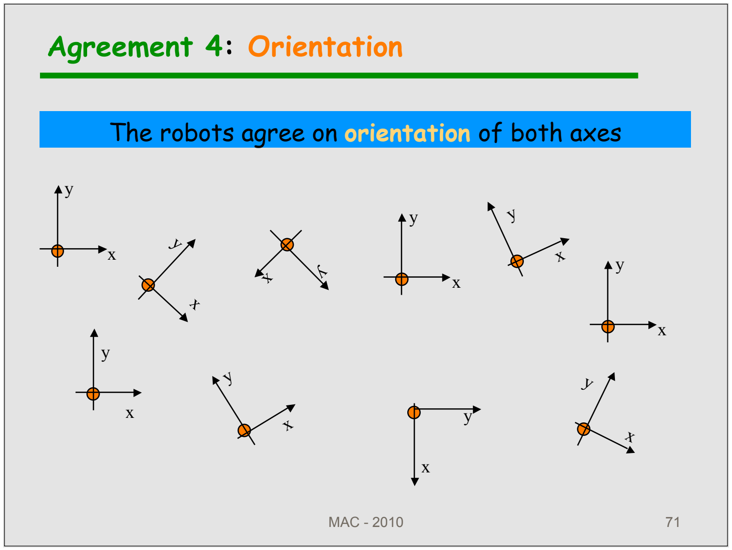
The robots agree on **direction** of both axes



Agreement 3: Direction

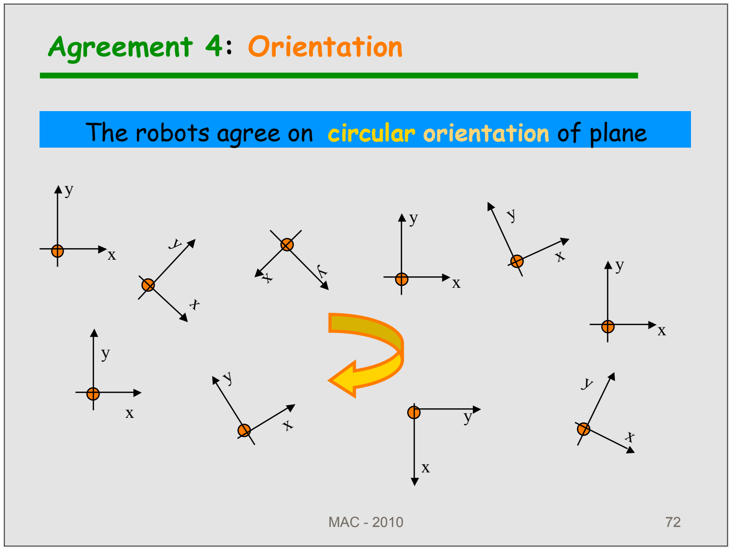
Might disagree on **origin** and **unit** of length

might disagree on **orientation** of axes



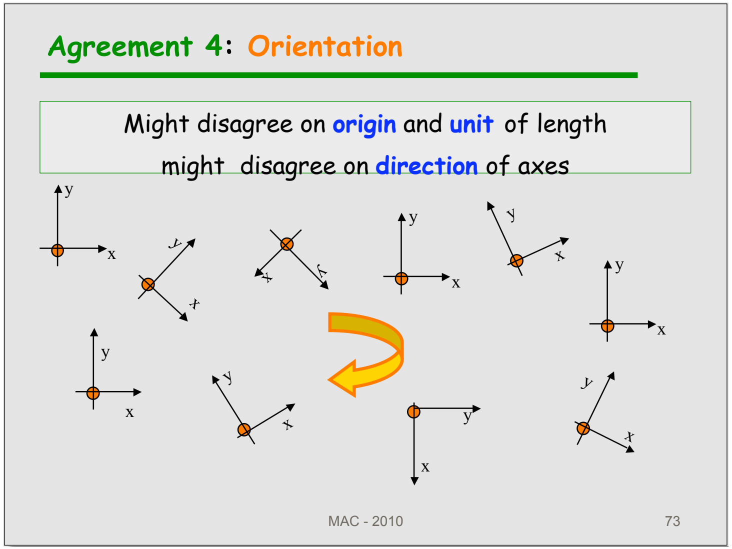
Agreement 4: Orientation

The robots agree on **orientation** of both axes



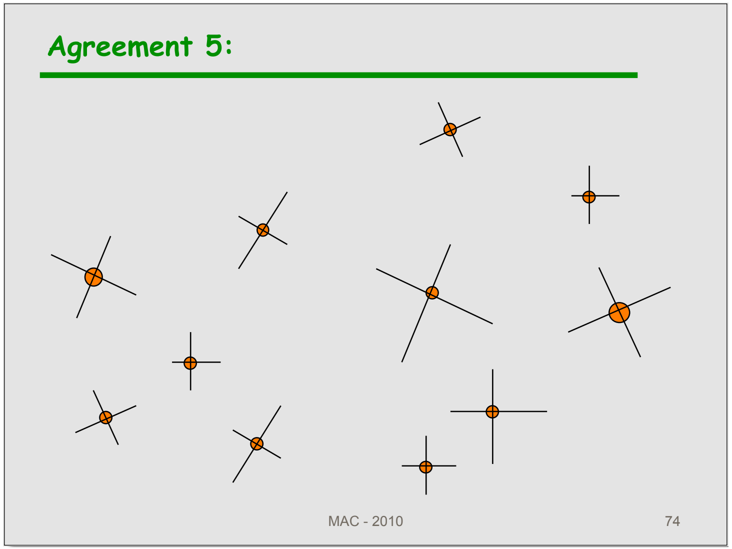
Agreement 4: Orientation

The robots agree on **circular orientation** of plane

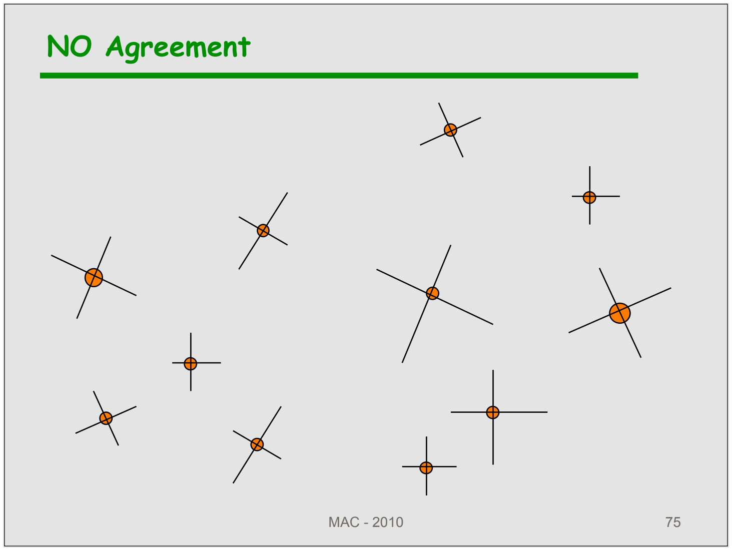


Agreement 4: Orientation

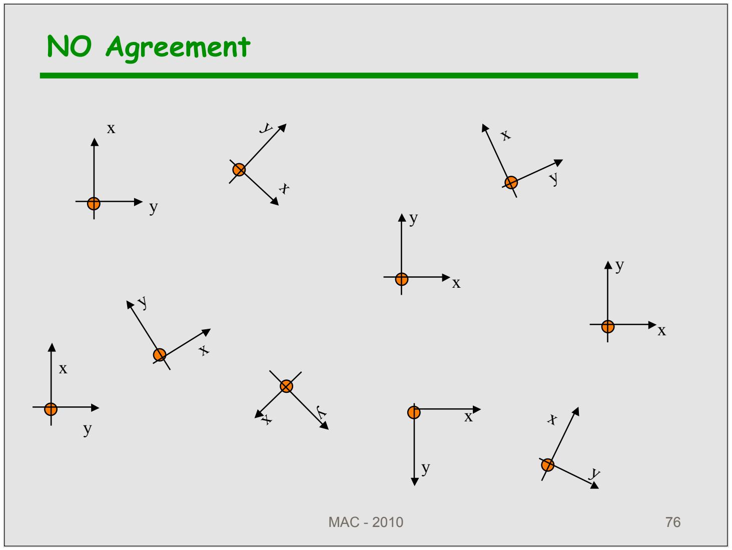
Might disagree on **origin** and **unit** of length
might disagree on **direction** of axes



Agreement 5:

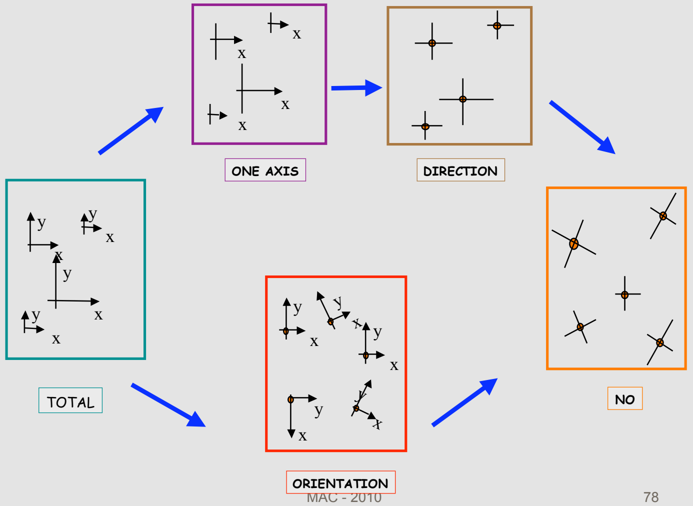


NO Agreement



NO Agreement

Levels of Agreement on Coordinate System

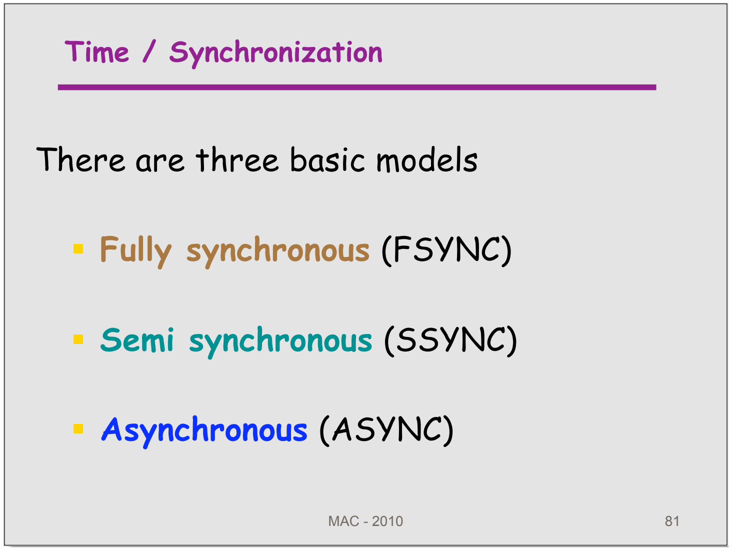


Crucial Factors

- **Visibility** Unimited → **Limited**
- **Memory** Persistent → Oblivious
- **Agreement** on local coordinate systems

Crucial Factors

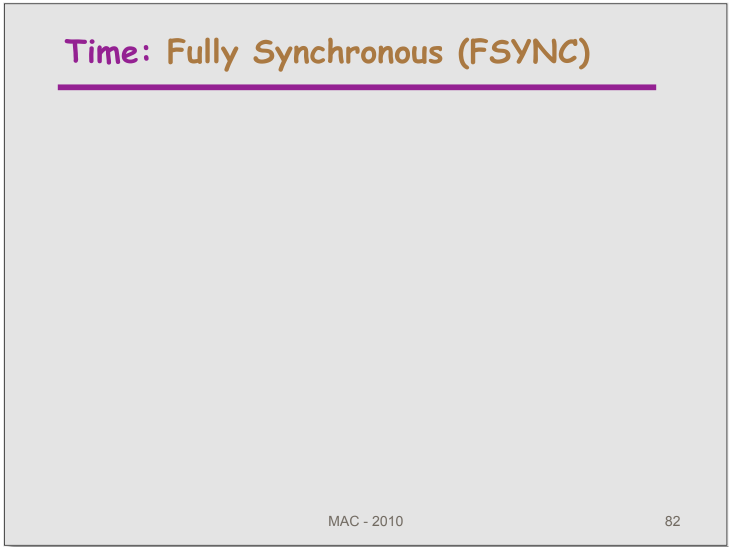
- **Visibility** Unimited → **Limited**
- **Memory** Persistent → Oblivious
- **Agreement** on local coordinate systems
- **Time/Synchronization**



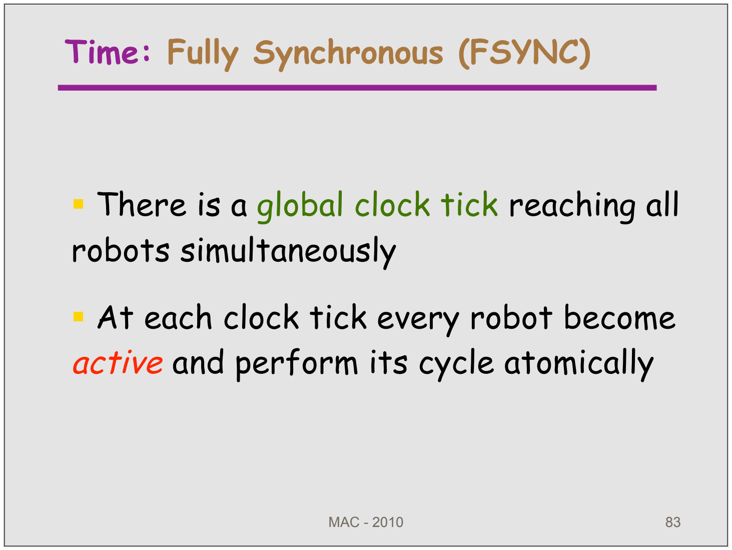
Time / Synchronization

There are three basic models

- Fully synchronous (FSYNC)
- Semi synchronous (SSYNC)
- Asynchronous (ASYNC)

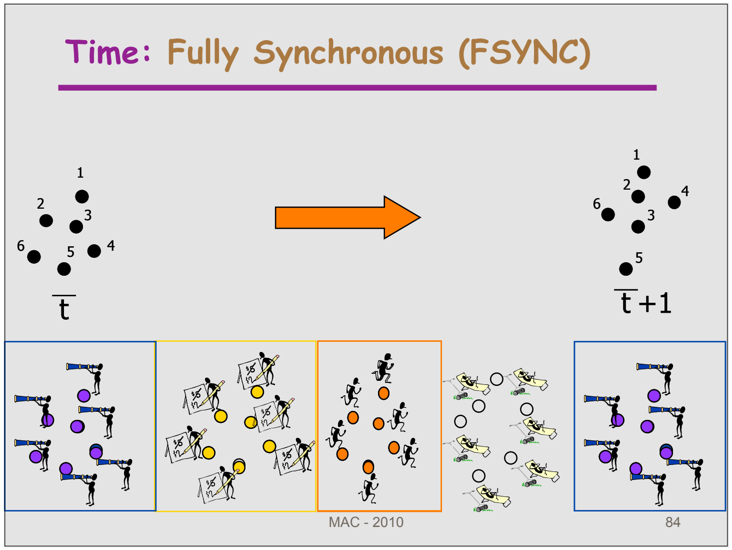


Time: Fully Synchronous (FSYNC)



Time: Fully Synchronous (FSYNC)

- There is a global clock tick reaching all robots simultaneously
- At each clock tick every robot become *active* and perform its cycle atomically



Time: Fully Synchronous (FSYNC)

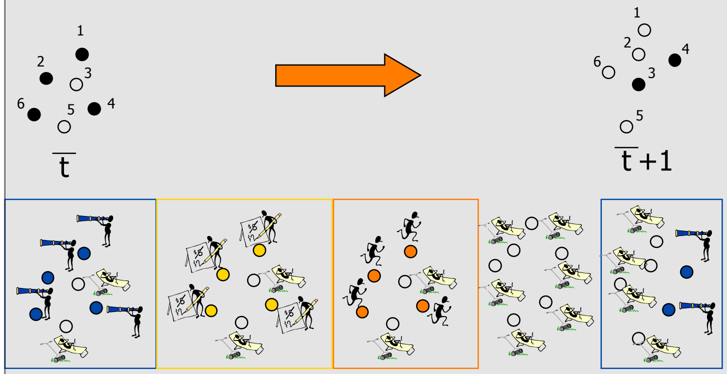
Time: Semi Synchronous (SSYNC)

Time: Semi Synchronous (SSYNC)

- There is a global clock tick reaching all robots simultaneously
- At each clock tick every robot is either *active* or *inactive*, and only active robots perform their cycle atomically.

[Suzuki Yamashita 96]

Time: Semi Synchronous (SSYNC)



Time: Semi Synchronous (SSYNC)

Fair Scheduler:

every robots becomes active infinitely often

Time: Asynchronous (ASYNC)

Time: Asynchronous (ASYNC)

-There is **NO** global clock and robots do not have a common notion of time

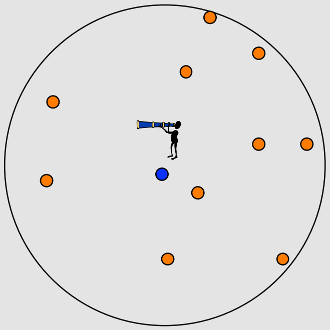
- Each robot becomes active at **unpredictable** time instants

- Each computation and movement takes a finite but **unpredictable** amount of time

[Flocchini Prencipe Santoro Widmayer 99]

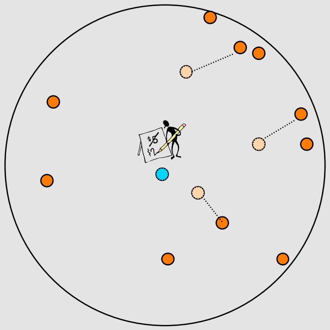
Time: Asynchronous (ASYNC)

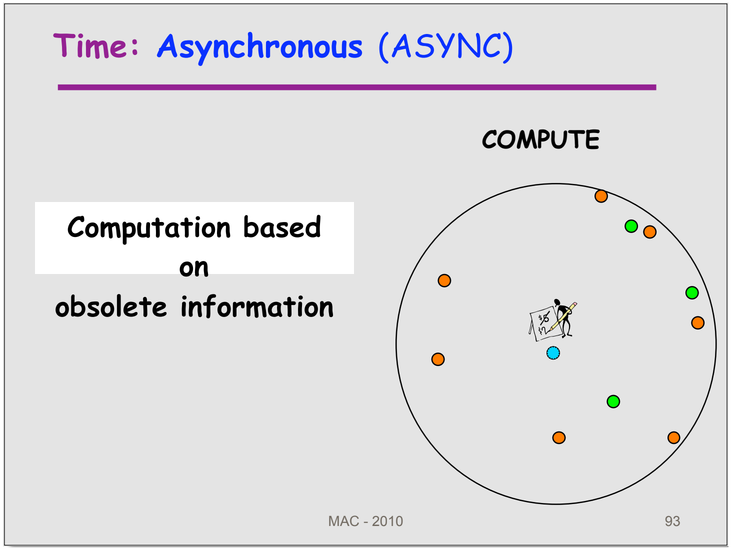
LOOK



Time: Asynchronous (ASYNC)

COMPUTE

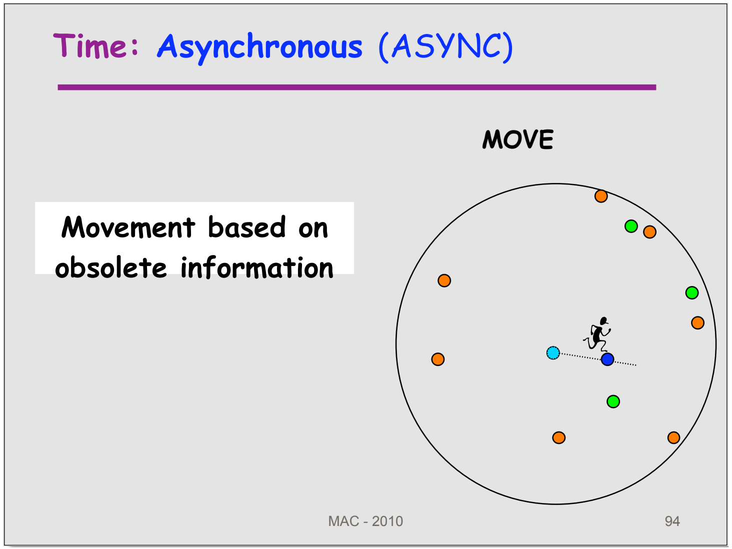




Time: Asynchronous (ASYNC)

COMPUTE

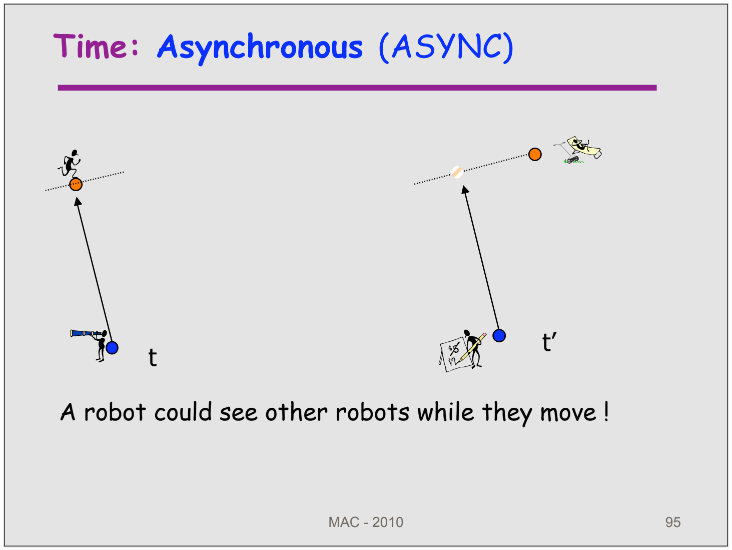
Computation based on obsolete information



Time: Asynchronous (ASYNC)

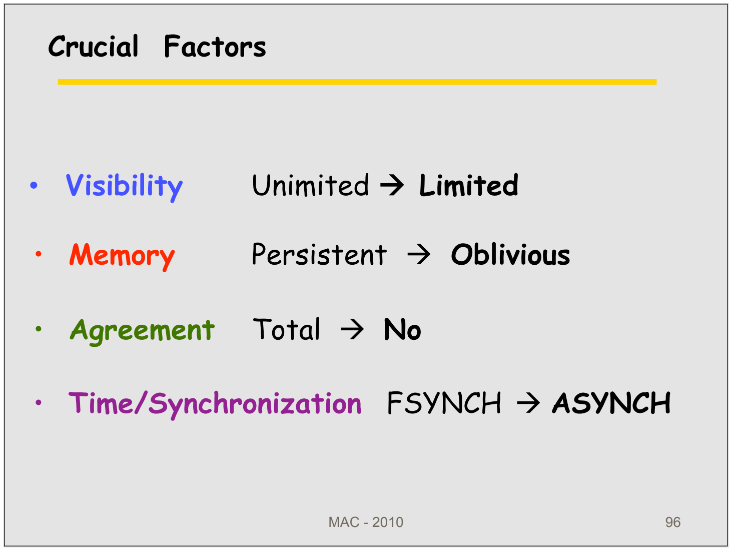
MOVE

Movement based on obsolete information



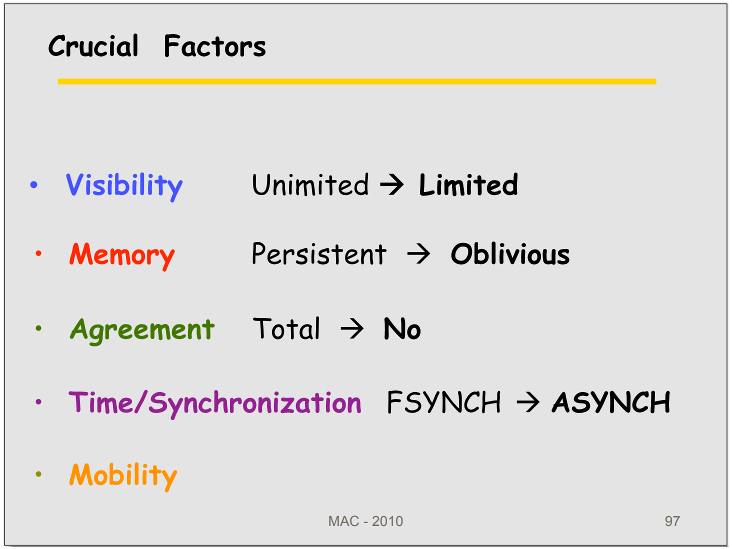
Time: Asynchronous (ASYNC)

A robot could see other robots while they move !



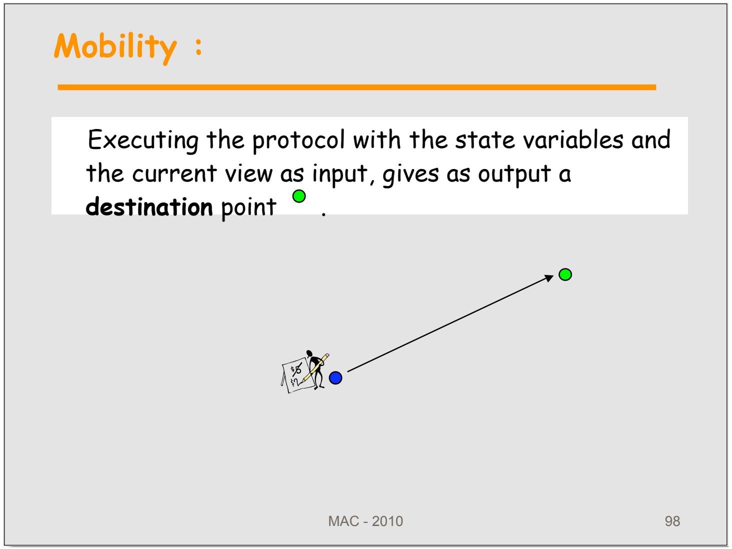
Crucial Factors

- Visibility Unimited → **Limited**
- Memory Persistent → **Oblivious**
- Agreement Total → **No**
- Time/Synchronization FSYNCH → ***ASYNCH***



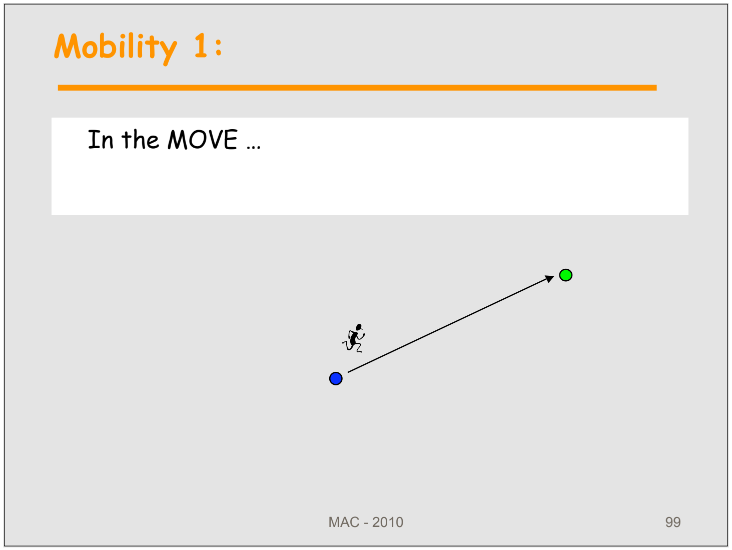
Crucial Factors

- **Visibility** Unimited → **Limited**
- **Memory** Persistent → **Oblivious**
- **Agreement** Total → **No**
- **Time/Synchronization** FSYNCH → ***ASYNCH***
- **Mobility**



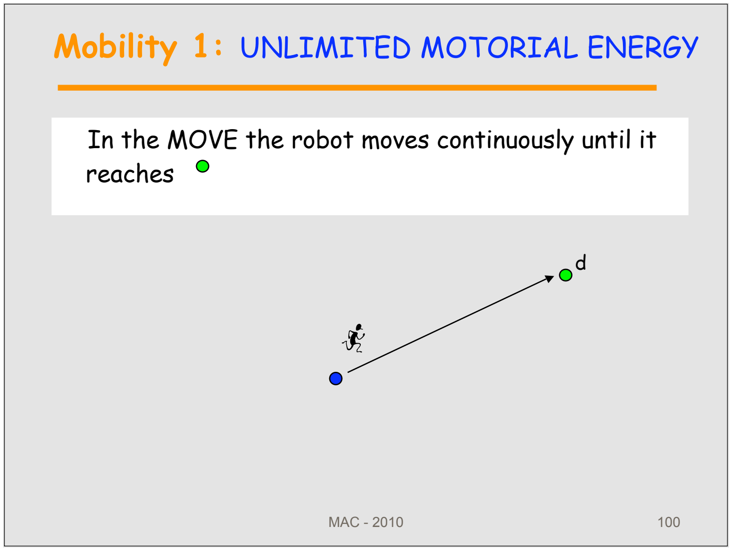
Mobility :

Executing the protocol with the state variables and the current view as input, gives as output a **destination** point ● .



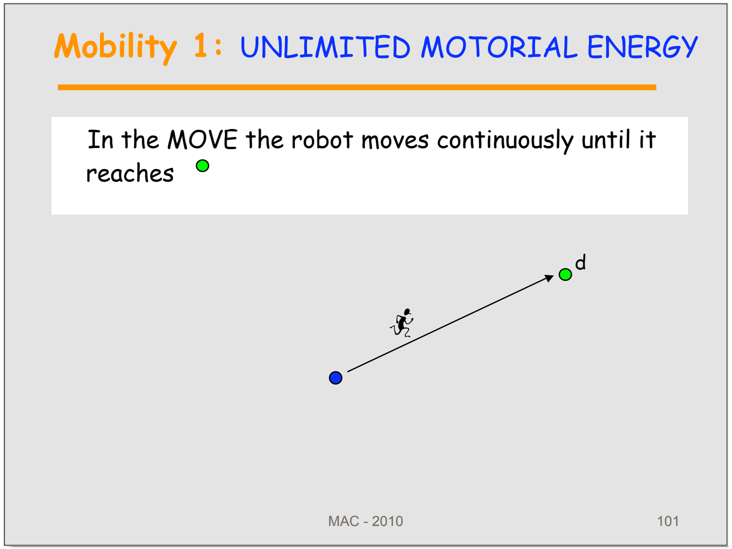
Mobility 1:

In the MOVE …



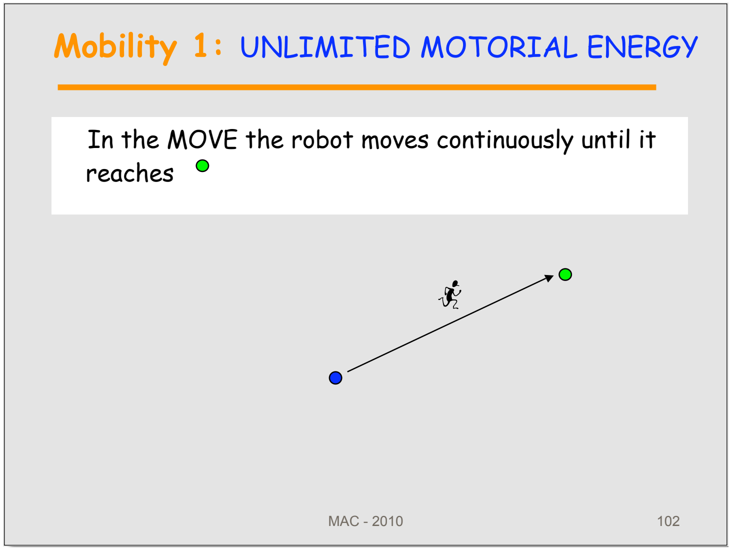
Mobility 1: UNLIMITED MOTORIAL ENERGY

In the MOVE the robot moves continuously until it reaches ●



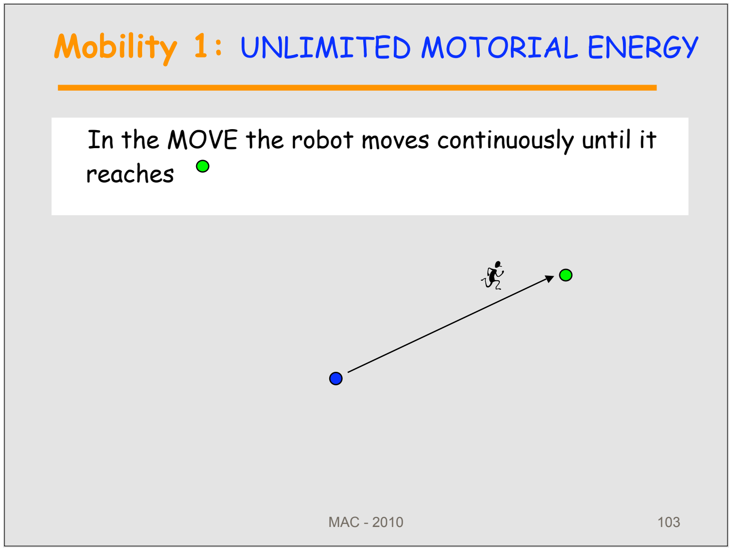
Mobility 1: UNLIMITED MOTORIAL ENERGY

In the MOVE the robot moves continuously until it reaches ●



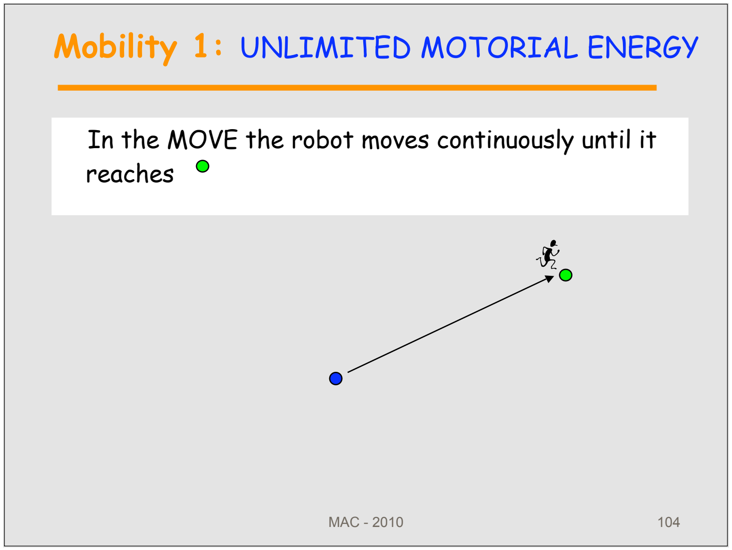
Mobility 1: UNLIMITED MOTORIAL ENERGY

In the MOVE the robot moves continuously until it reaches ●



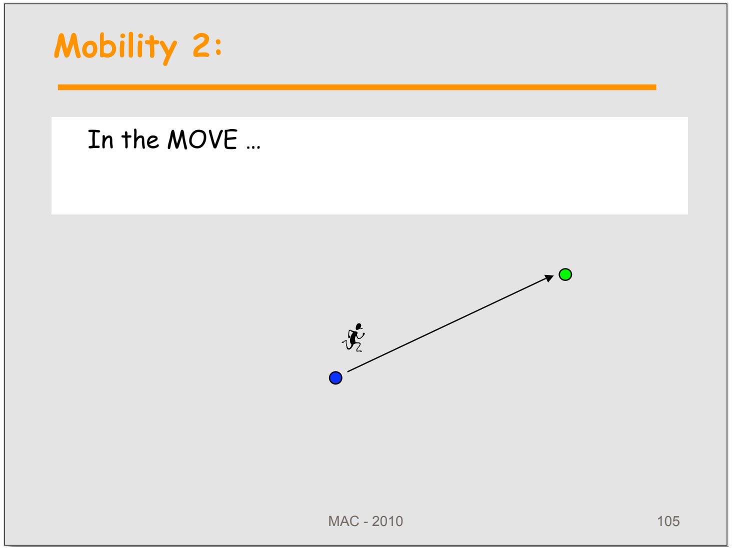
Mobility 1: UNLIMITED MOTORIAL ENERGY

In the MOVE the robot moves continuously until it reaches ●



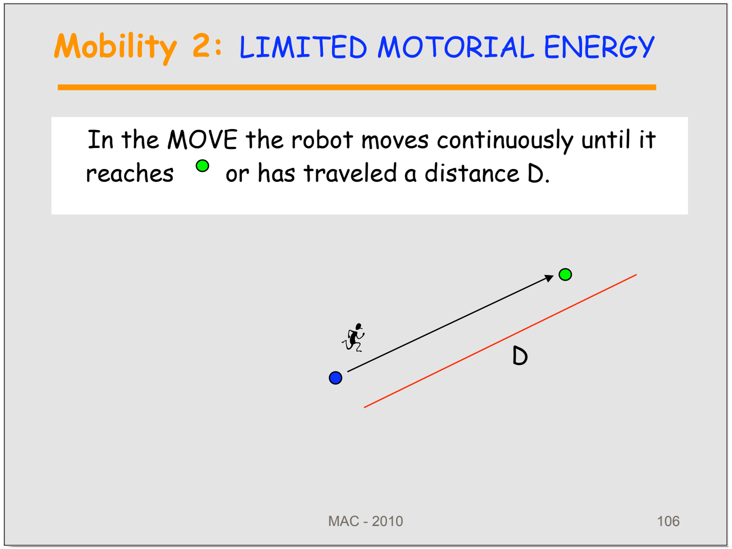
Mobility 1: UNLIMITED MOTORIAL ENERGY

In the MOVE the robot moves continuously until it reaches ●



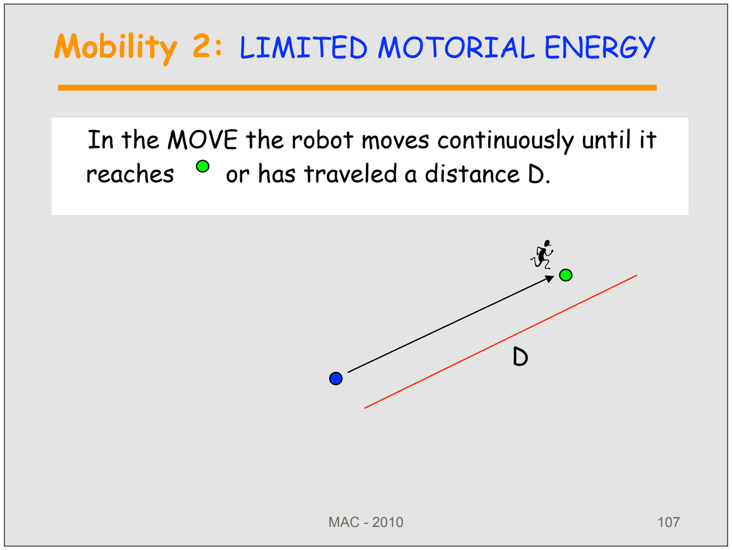
Mobility 2:

In the MOVE …



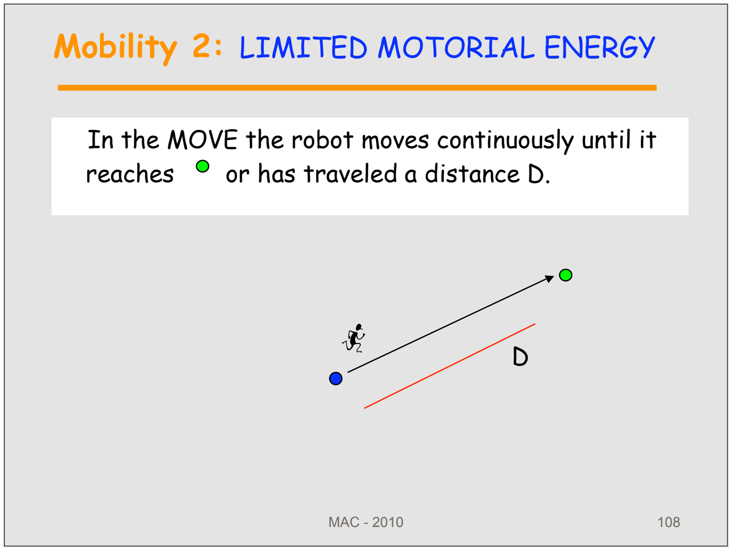
Mobility 2: LIMITED MOTORIAL ENERGY

In the MOVE the robot moves continuously until it reaches ● or has traveled a distance D.



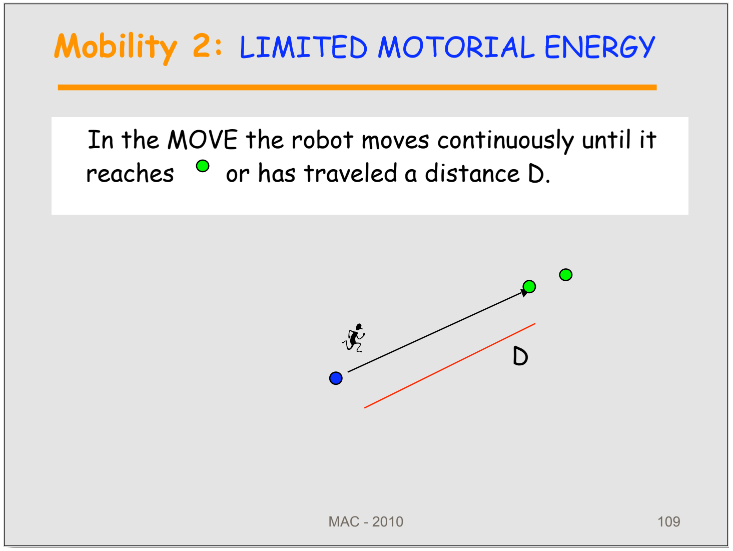
Mobility 2: LIMITED MOTORIAL ENERGY

In the MOVE the robot moves continuously until it reaches ● or has traveled a distance D.



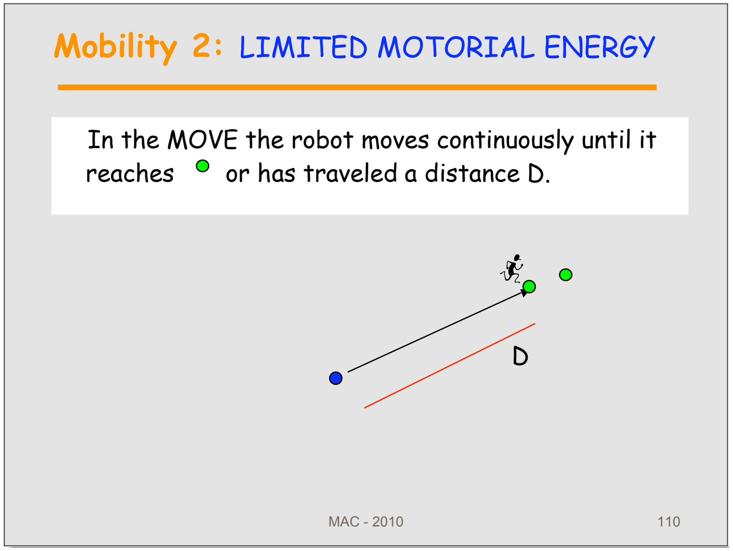
Mobility 2: LIMITED MOTORIAL ENERGY

In the MOVE the robot moves continuously until it reaches ● or has traveled a distance D.



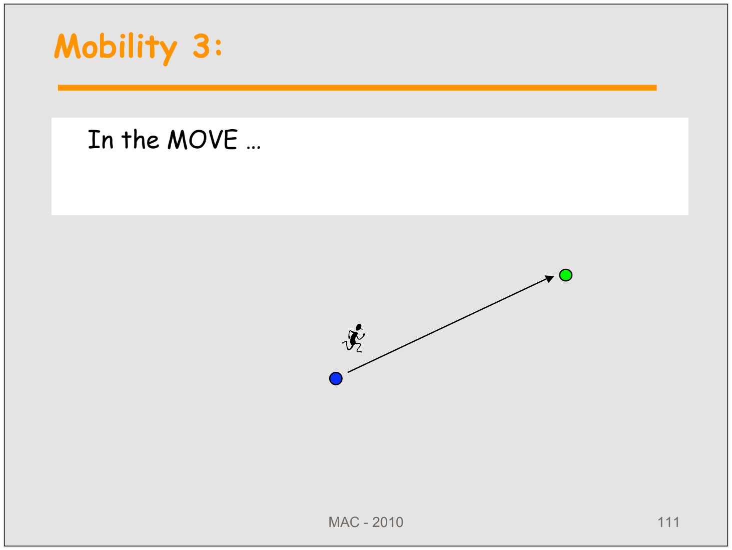
Mobility 2: LIMITED MOTORIAL ENERGY

In the MOVE the robot moves continuously until it reaches ● or has traveled a distance D.



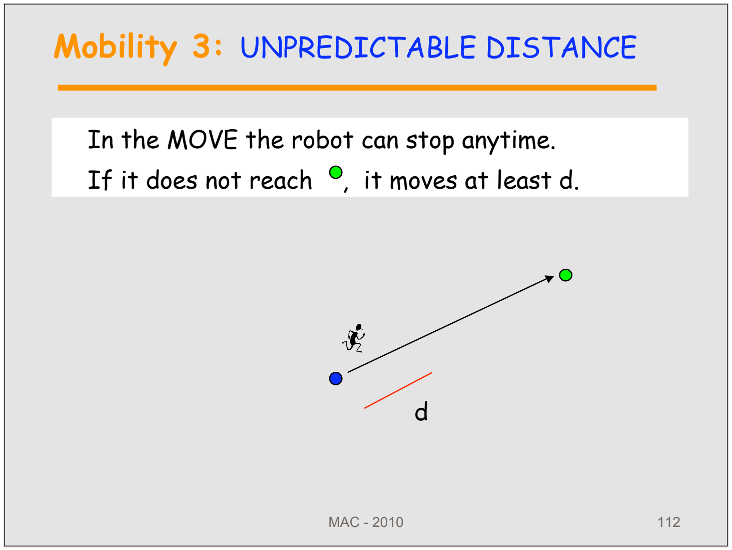
Mobility 2: LIMITED MOTORIAL ENERGY

In the MOVE the robot moves continuously until it reaches ● or has traveled a distance D.



Mobility 3:

In the MOVE …



Mobility 3: UNPREDICTABLE DISTANCE

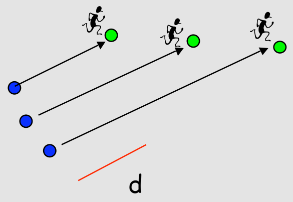
In the MOVE the robot can stop anytime.

If it does not reach ●, it moves at least d.

Mobility 3: UNPREDICTABLE DISTANCE

In the MOVE the robot can stop anytime.

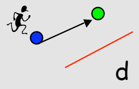
If it does not reach ●, it moves at least d.



Mobility 3: UNPREDICTABLE DISTANCE

In the MOVE the robot can stop anytime.

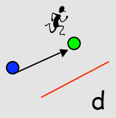
If it does not reach ●, it moves at least d.



Mobility 3: UNPREDICTABLE DISTANCE

In the MOVE the robot can stop anytime.

If it does not reach ●, it moves at least d.



HOMOGENEOUS

every robot executes the same **protocol**

all robots are **identical** and **indistingushable**

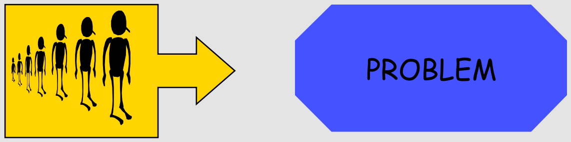
TERMINATION

NON-OBLIVIOUS

within finite time, every robot enters a **terminal** state

OBLIVIOUS

within finite time, every robot performs only **null** actions


PROBLEM

IMPOSSIBILITY

No **deterministic** protocol exists which **always** correctly solves the problem terminating in finite time

Arbitrary Pattern Formation

Pattern Formation

Unlimited Visibility

Pattern Formation

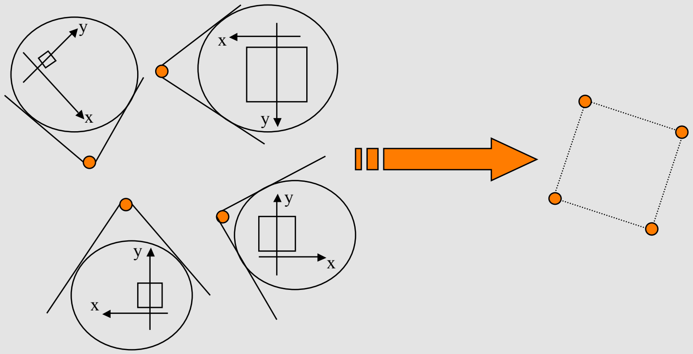
Pattern *P* is a set of (distinct) coordinates in the plane.

Number of robots is equal to the number of coordinates.

At the beginning,
the robots are in distinct arbitrary positions.

At the end,
the positions of the robots must "coincide" with ***P*** (*translation*, *scaling* or *rotation).*

Pattern Formation



Computation/Protocol

FORMATION
the pattern is **formed within finite time**
(terminating protocol)

CONVERGENCE
the pattern is never formed but the robots' position converges to the pattern in the limit
(non-terminating protocol)
(terminating approximate solution)

Arbitrary Pattern Formation

Arbitrary Pattern Formation

The n robots must form
any input pattern *P* of n points
starting from *any* initial position

Arbitrary Pattern Formation

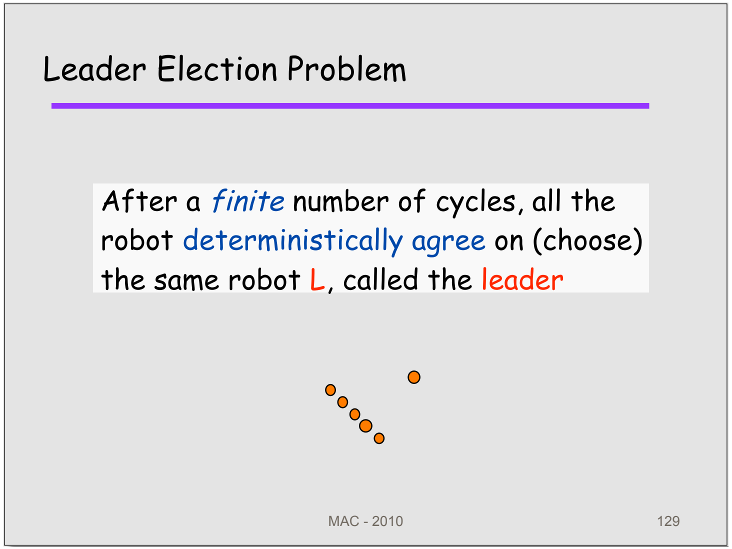
…. is not always possible

Theorem

For $n>2$, if ***Arbitrary Pattern Formation*** *is solvable with level of agreement A*
then
the ***Leader Election*** *problem is solvable with level of agreement A*

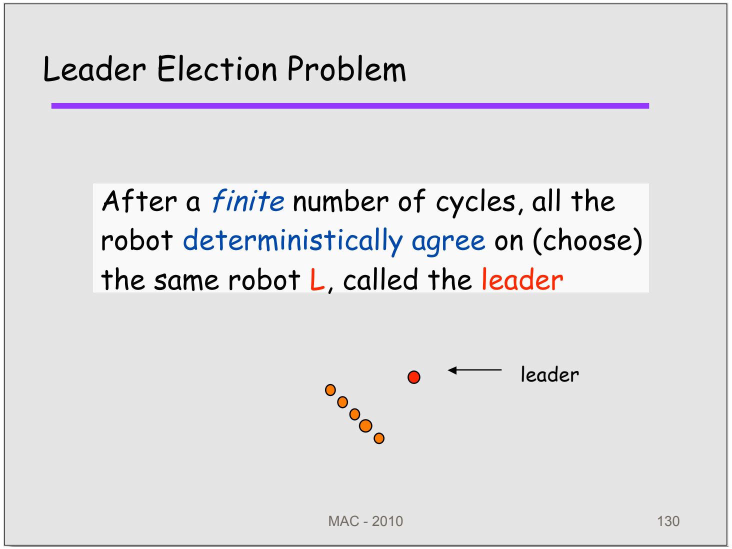
Leader Election Problem

After a *finite* number of cycles, all the robot deterministically agree on (choose) the same robot L, called the leader



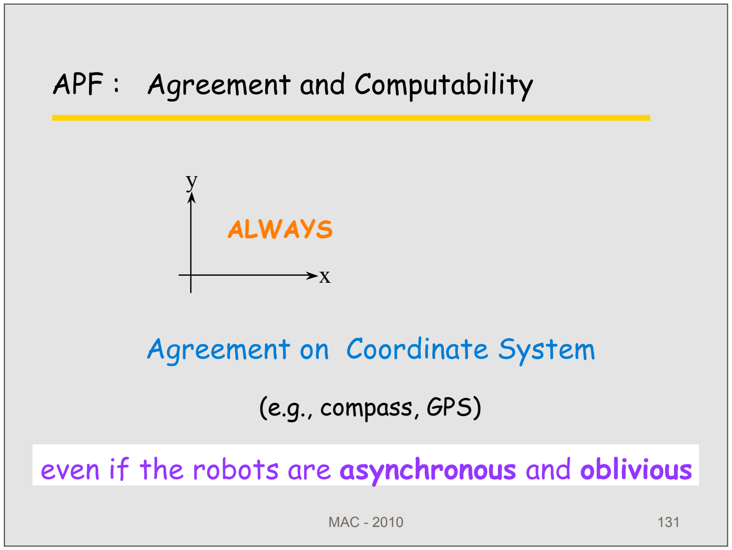
Leader Election Problem

After a *finite* number of cycles, all the robot deterministically agree on (choose) the same robot L, called the leader



Leader Election Problem

After a *finite* number of cycles, all the robot deterministically agree on (choose) the same robot L, called the leader

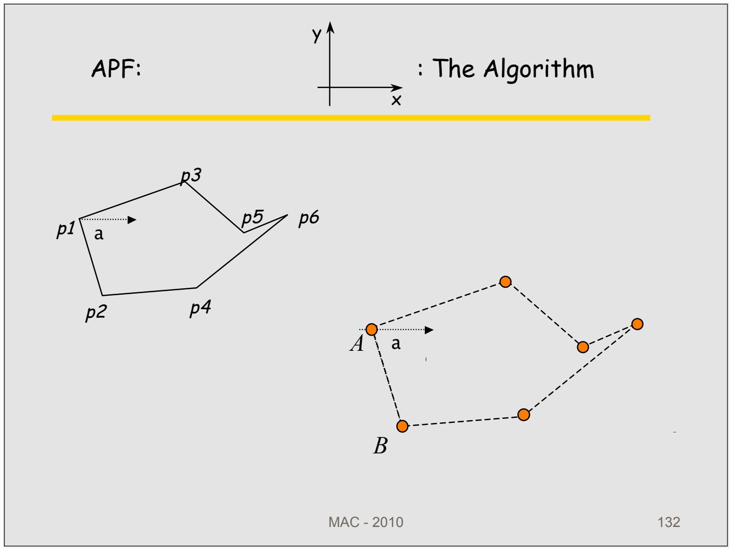


APF : Agreement and Computability

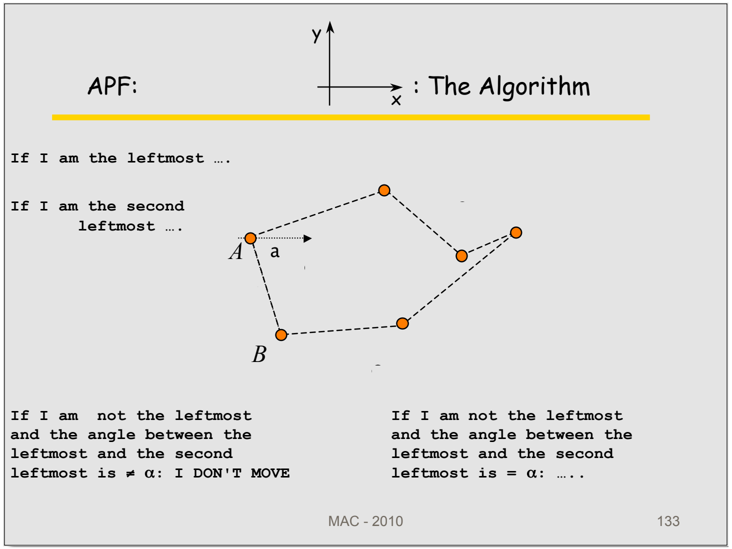
Agreement on Coordinate System

(e.g., compass, GPS)

even if the robots are **asynchronous** and **oblivious**



APF: : The Algorithm



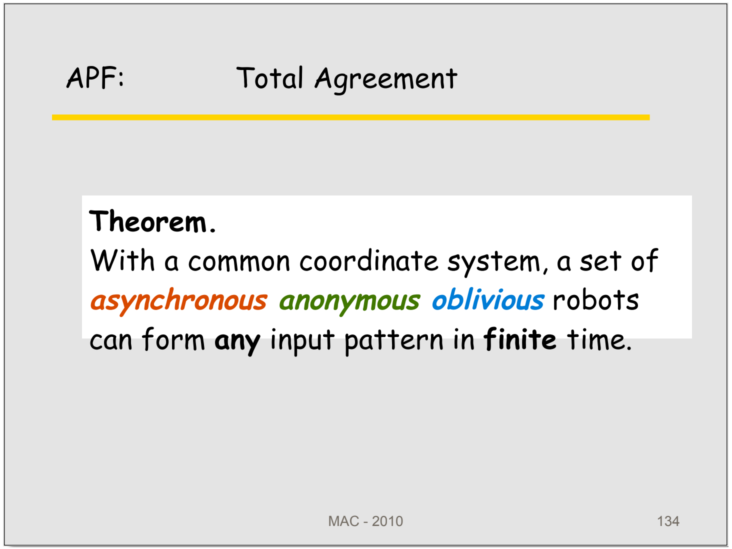
APF: : The Algorithm

If I am the leftmost ….

If I am the second
leftmost ….

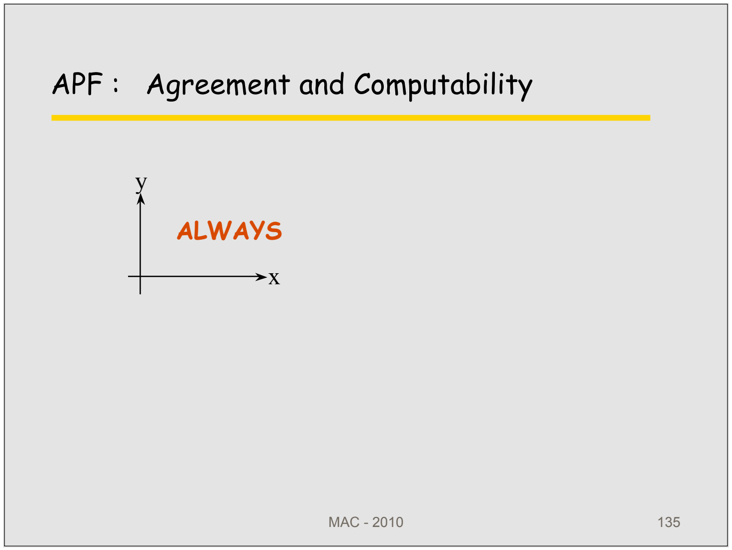
If I am not the leftmost
and the angle between the
leftmost and the second
leftmost is ≠ α: I DON'T MOVE

If I am not the leftmost
and the angle between the
leftmost and the second
leftmost is = α: …..



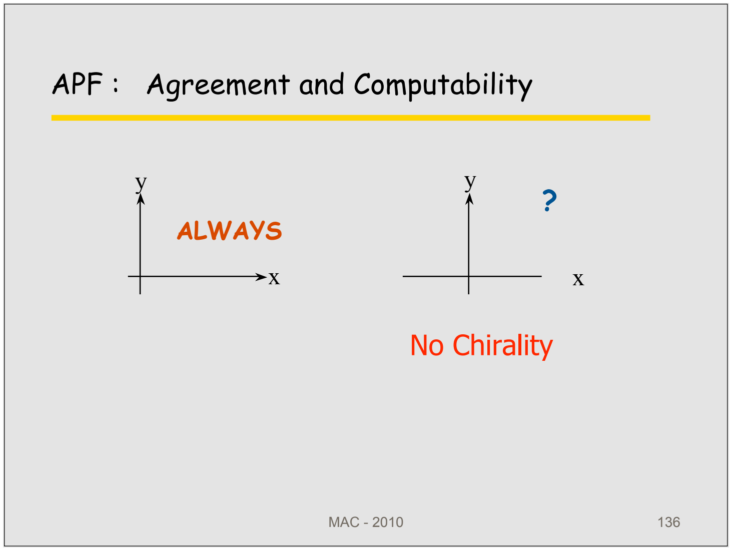
APF: Total Agreement

Theorem.
With a common coordinate system, a set of *asynchronous* *anonymous* *oblivious* robots can form **any** input pattern in **finite** time.



APF : Agreement and Computability

ALWAYS

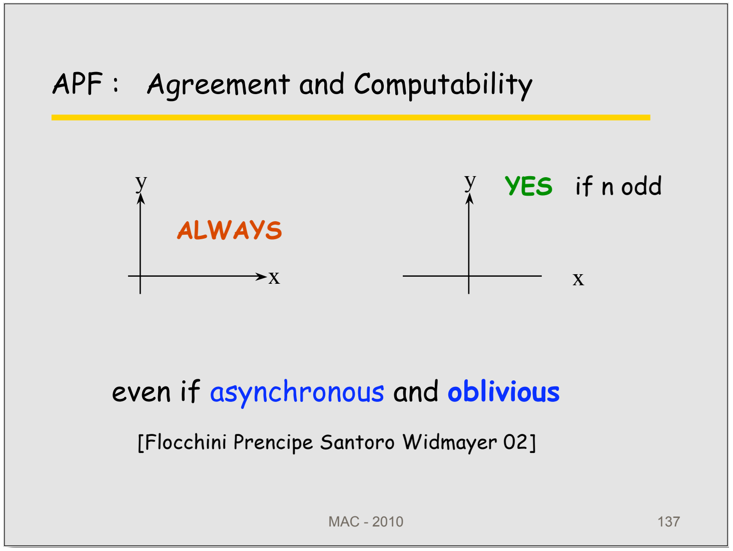


APF : Agreement and Computability

ALWAYS

?

No Chirality



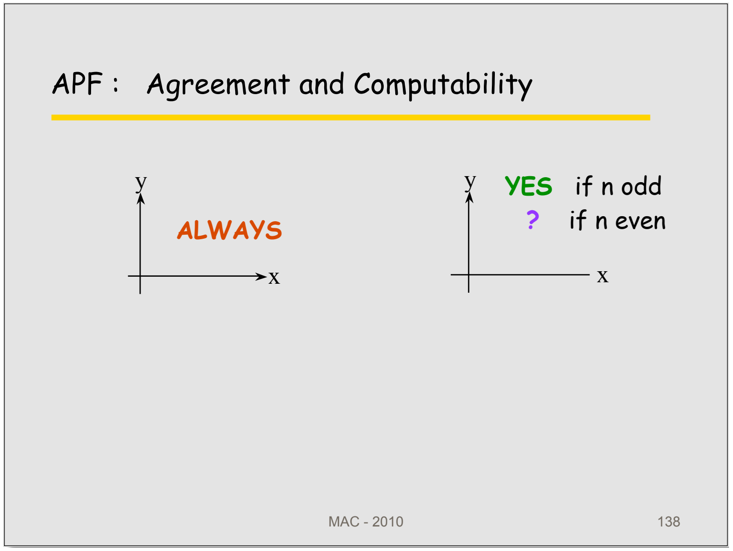
APF : Agreement and Computability

ALWAYS

YES if n odd

even if asynchronous and **oblivious**

[Flocchini Prencipe Santoro Widmayer 02]

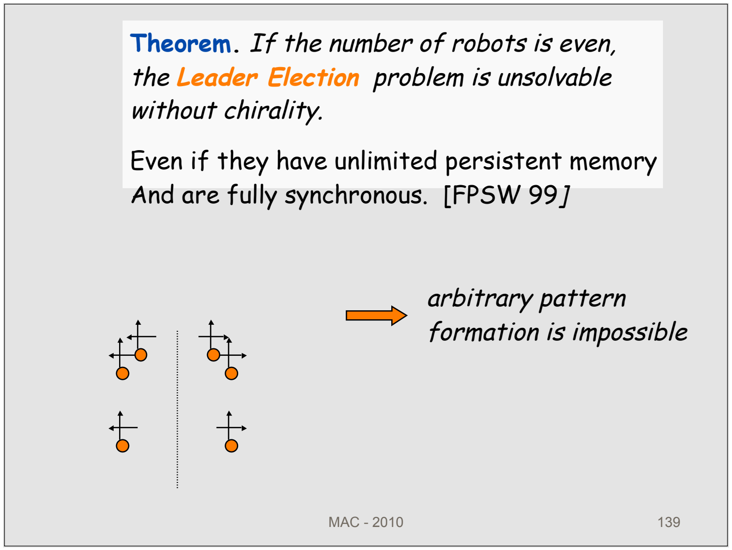


APF : Agreement and Computability

ALWAYS

YES if n odd

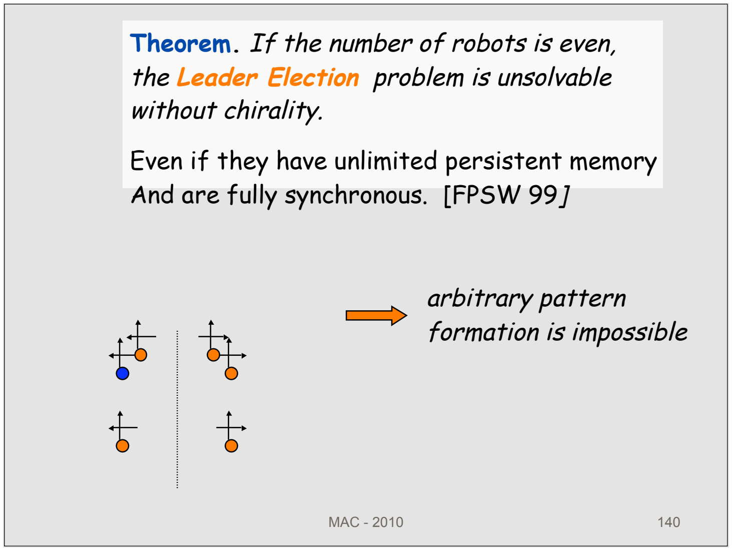
? if n even



Theorem. *If the number of robots is even, the Leader Election problem is unsolvable without chirality.*

Even if they have unlimited persistent memory And are fully synchronous. [FPSW 99]

→ *arbitrary pattern formation is impossible*



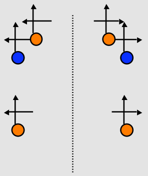
Theorem. *If the number of robots is even, the Leader Election problem is unsolvable without chirality.*

Even if they have unlimited persistent memory And are fully synchronous. [FPSW 99]

→ *arbitrary pattern formation is impossible*

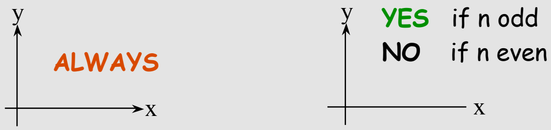
Theorem. *If the number of robots is even, the* ***Leader Election*** *problem is unsolvable without chirality.*

Even if they have unlimited persistent memory
And are fully synchronous. [FPSW 99]



→ *arbitrary pattern formation is impossible*

APF : Agreement and Computability



ALWAYS

YES if n odd
NO if n even

Pattern Formation : n even and no chirality

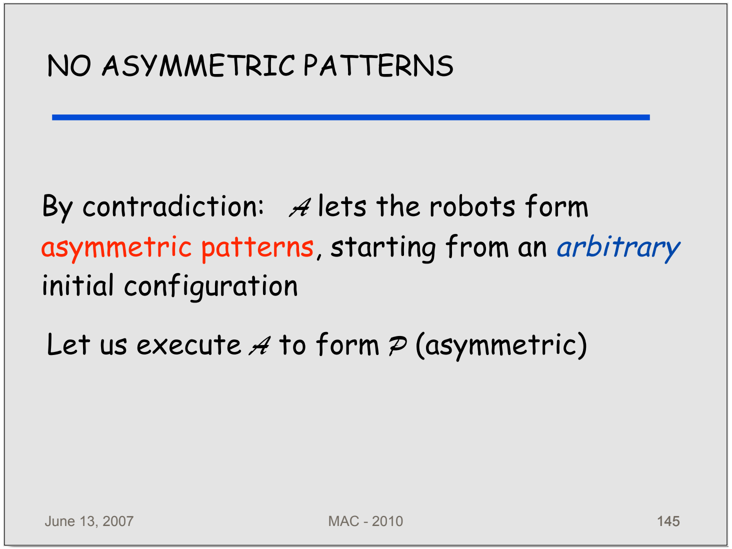
Q: Which patterns can be formed when n is even and no chirality?

Pattern Formation : n even and no chirality

Q: Which patterns can be formed when n is even and no chirality?

A: NO asymmetric patterns

even if
- unbounded persistent memory
- fully synchronous system

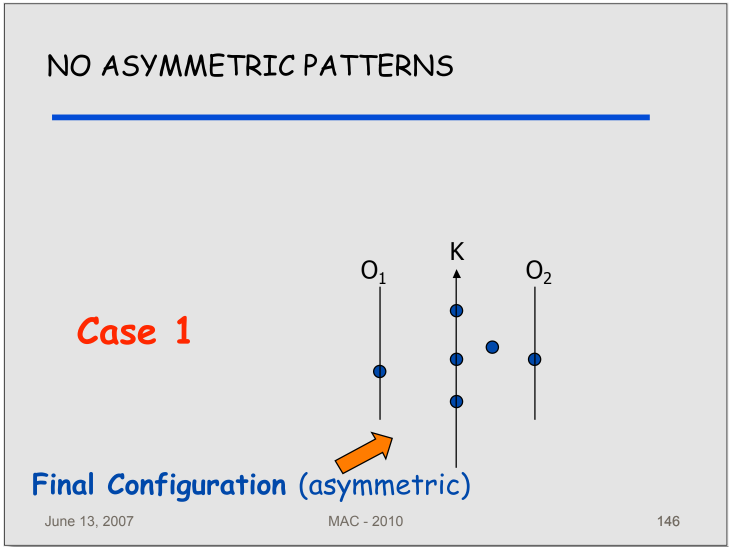


NO ASYMMETRIC PATTERNS

By contradiction: $\mathcal{A}$ lets the robots form asymmetric patterns, starting from an *arbitrary* initial configuration

Let us execute $\mathcal{A}$ to form $\mathcal{P}$ (asymmetric)

June 13, 2007 MAC - 2010 145

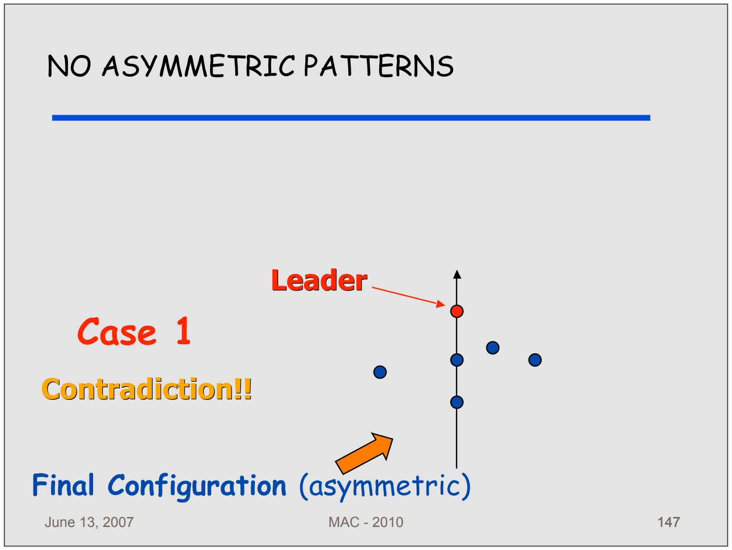


NO ASYMMETRIC PATTERNS

Case 1

Final Configuration (asymmetric)

June 13, 2007 MAC - 2010 146



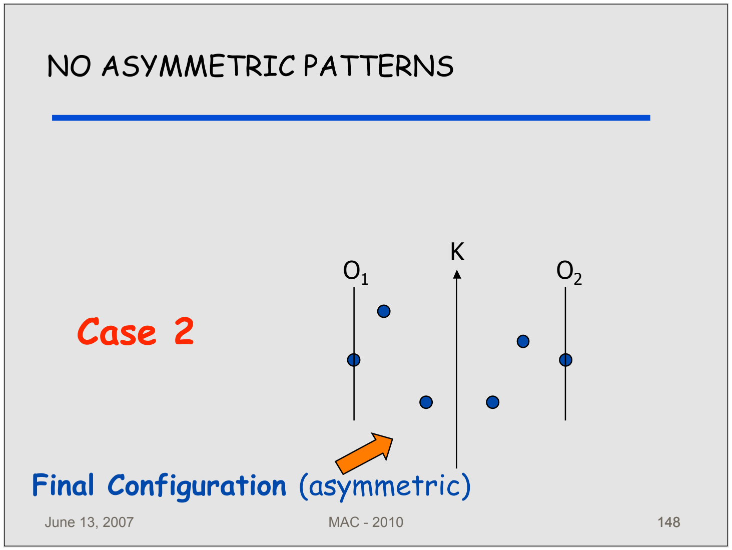
NO ASYMMETRIC PATTERNS

Case 1

Contradiction!!

Final Configuration (asymmetric)

June 13, 2007 MAC - 2010 147

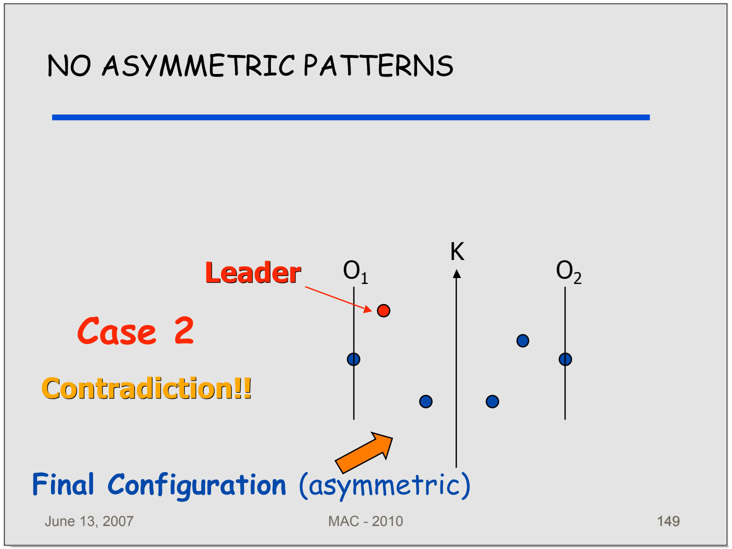


NO ASYMMETRIC PATTERNS

Case 2

Final Configuration (asymmetric)

June 13, 2007 MAC - 2010 148

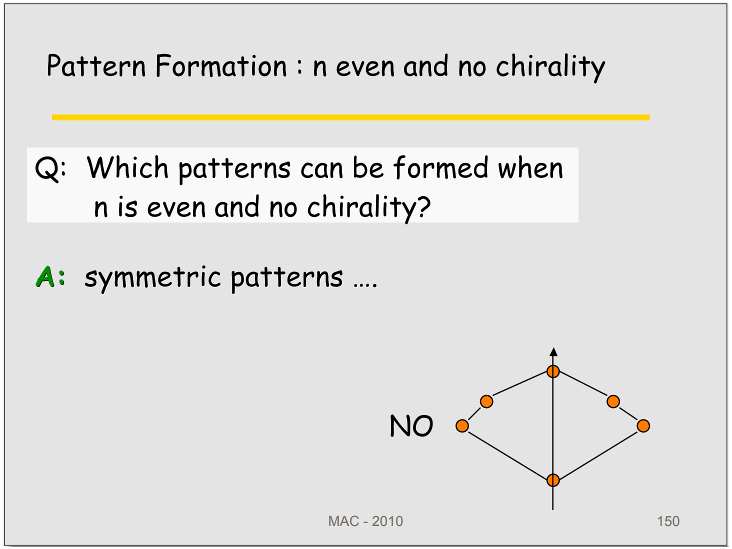


NO ASYMMETRIC PATTERNS

Case 2

Contradiction!!

Final Configuration (asymmetric)

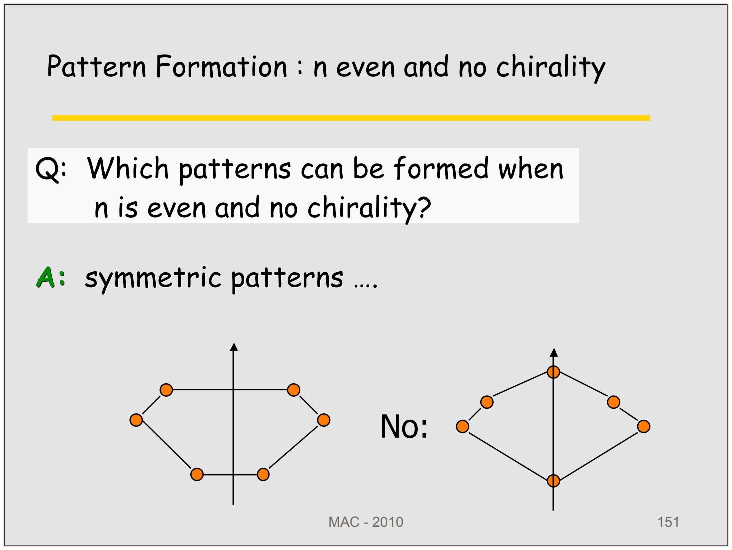


Pattern Formation : n even and no chirality

Q: Which patterns can be formed when n is even and no chirality?

A: symmetric patterns ….

NO

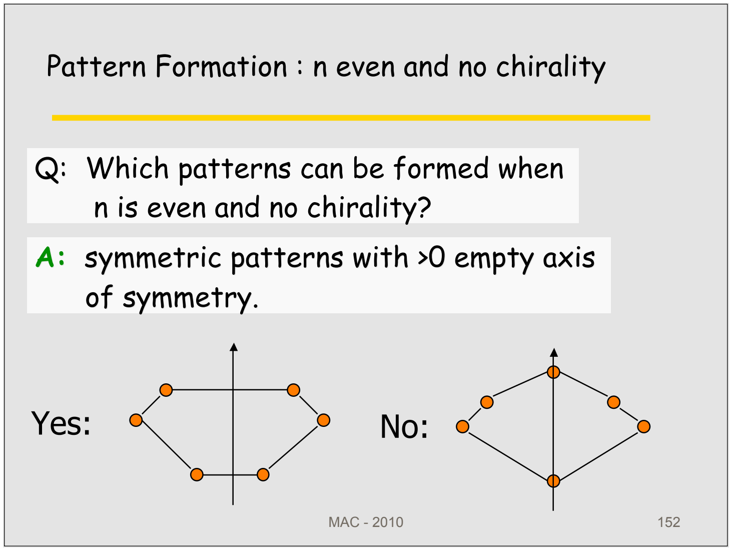


Pattern Formation : n even and no chirality

Q: Which patterns can be formed when n is even and no chirality?

A: symmetric patterns ….

No:



Pattern Formation : n even and no chirality

Q: Which patterns can be formed when n is even and no chirality?

A: symmetric patterns with >0 empty axis of symmetry.

Yes:

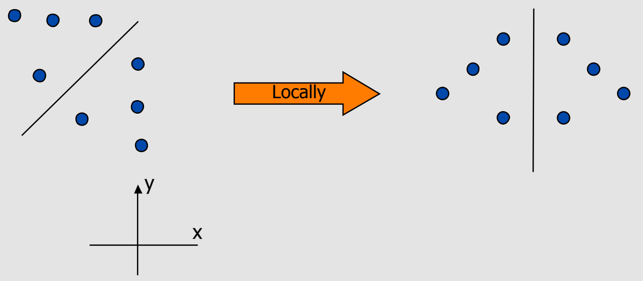
No:

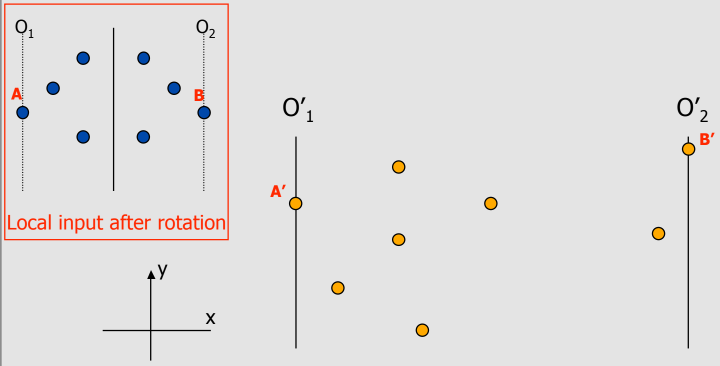
Pattern Formation : n even and no chirality

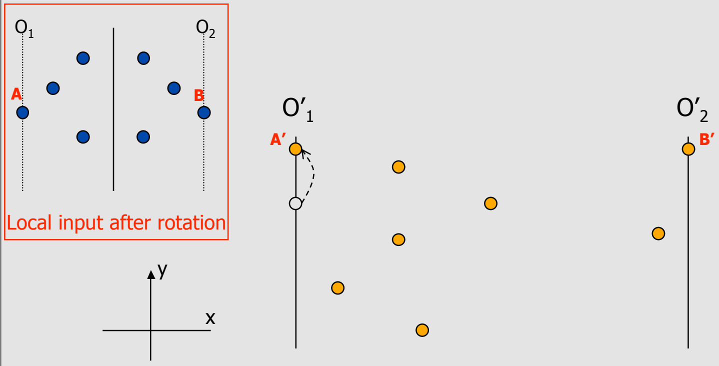
A: symmetric patterns with >0 empty axis of symmetry.

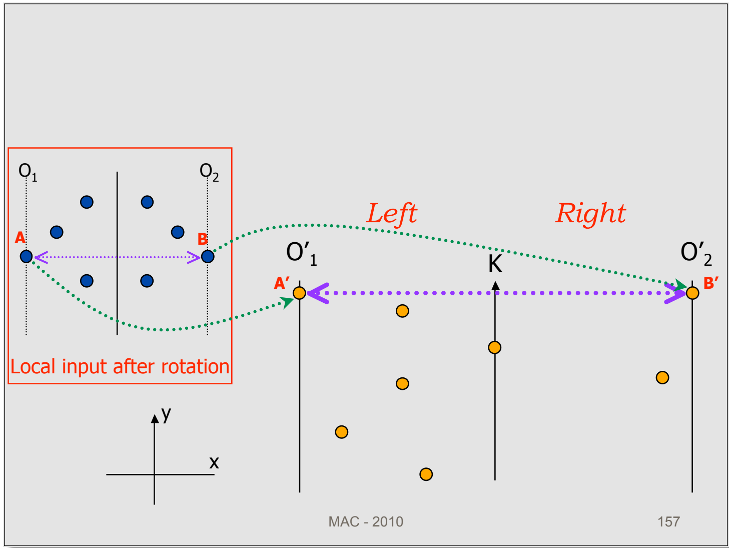
Local operations:

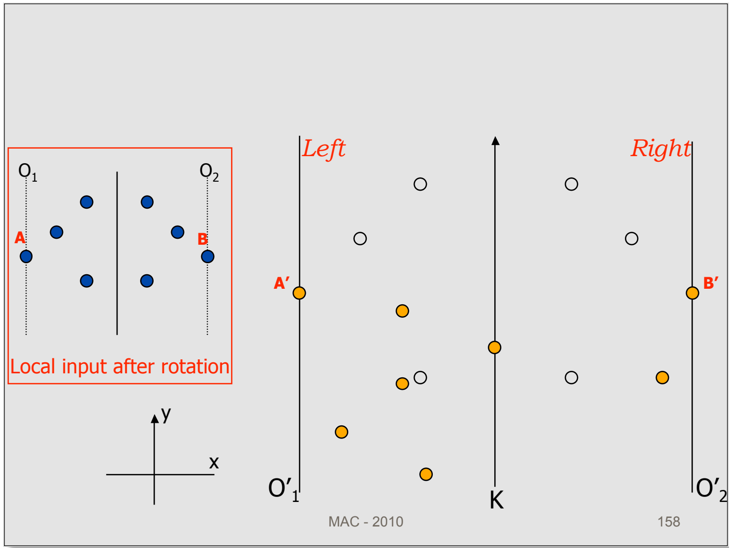
- Compute an empty axis of symmetry S (the same for all robots) of input pattern P
- Rotate P so that S is parallel to the agreed orientation of Y

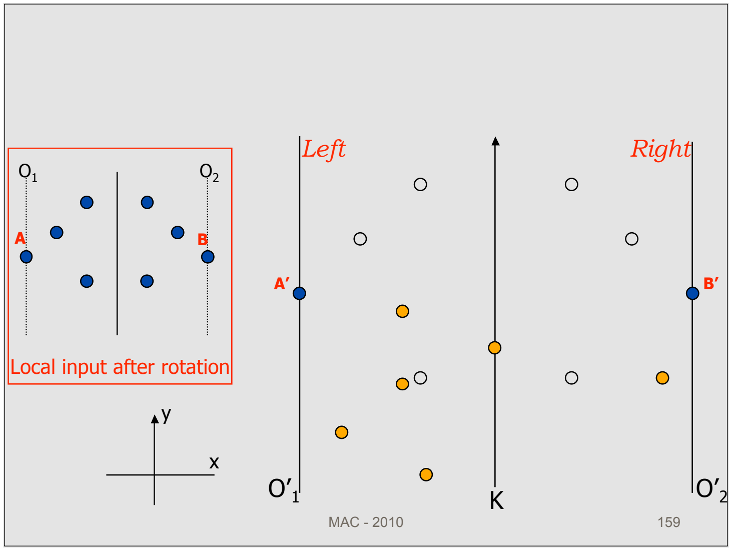










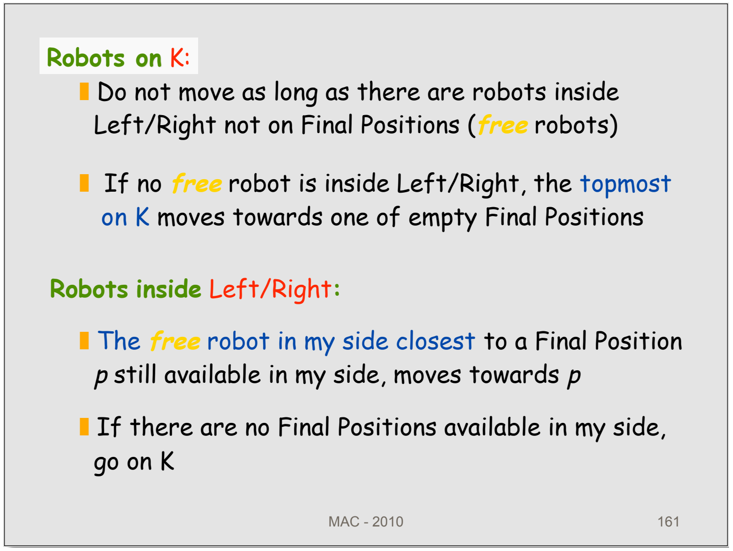


Robots on K:

- Do not move as long as there are robots inside Left/Right not on Final Positions (*free* robots)
- If no *free* robot is inside Left/Right, the topmost on K moves towards one of the Final Positions still available

Robots inside Left/Right:

- The *free* robot in my side closest to a Final Position *p* still available in my side, moves towards *p*
- If there are no Final Positions available in my side, go on K

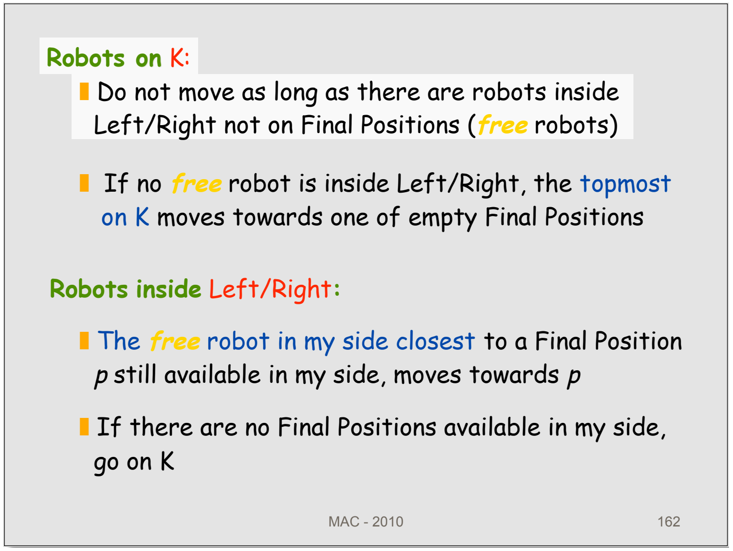


Robots on K:

- Do not move as long as there are robots inside Left/Right not on Final Positions (*free* robots)
- If no *free* robot is inside Left/Right, the topmost on K moves towards one of empty Final Positions

Robots inside Left/Right:

- The *free* robot in my side closest to a Final Position *p* still available in my side, moves towards *p*
- If there are no Final Positions available in my side, go on K

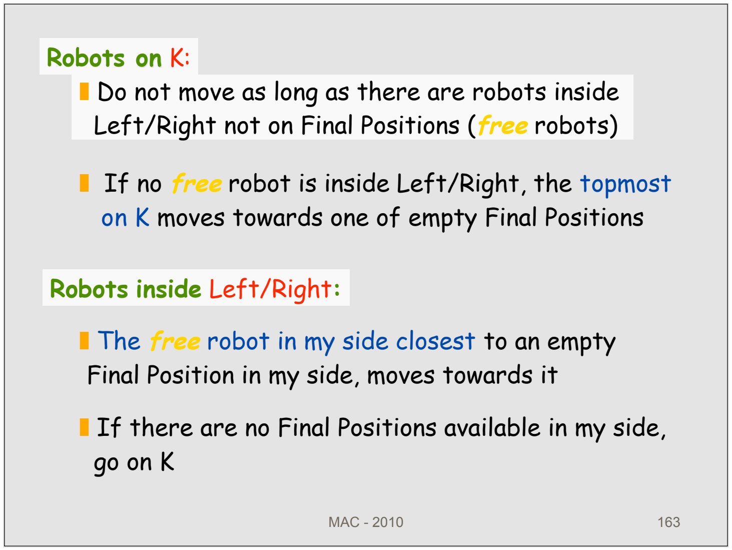


Robots on K:

- Do not move as long as there are robots inside Left/Right not on Final Positions (*free* robots)
- If no *free* robot is inside Left/Right, the topmost on K moves towards one of empty Final Positions

Robots inside Left/Right:

- The *free* robot in my side closest to a Final Position *p* still available in my side, moves towards *p*
- If there are no Final Positions available in my side, go on K

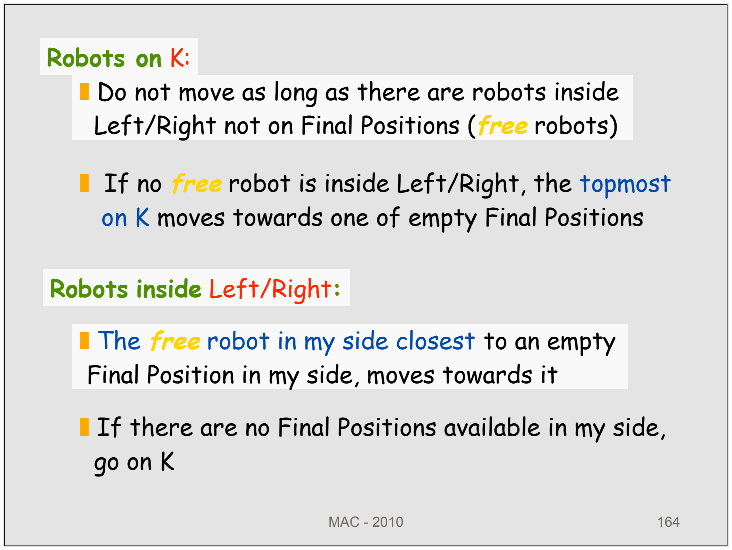


Robots on K:

- Do not move as long as there are robots inside Left/Right not on Final Positions (*free* robots)
- If no *free* robot is inside Left/Right, the topmost on K moves towards one of empty Final Positions

Robots inside Left/Right:

- The *free* robot in my side closest to an empty Final Position in my side, moves towards it
- If there are no Final Positions available in my side, go on K

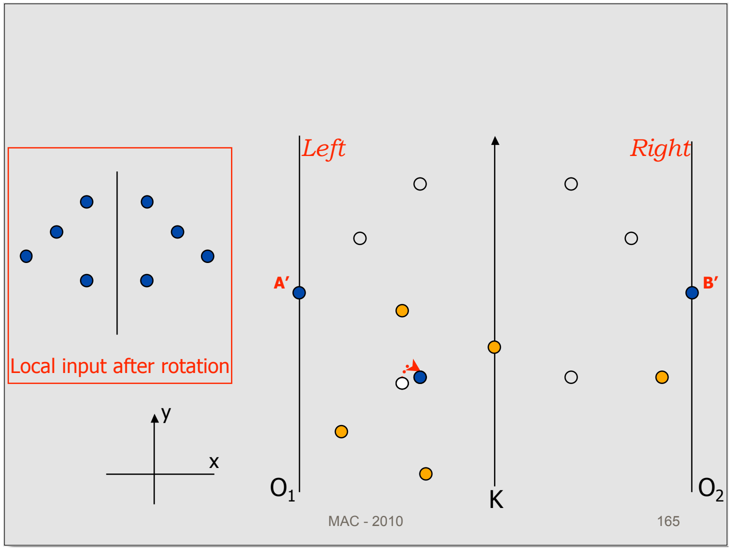


Robots on K:

- Do not move as long as there are robots inside Left/Right not on Final Positions (*free* robots)
- If no *free* robot is inside Left/Right, the topmost on K moves towards one of empty Final Positions

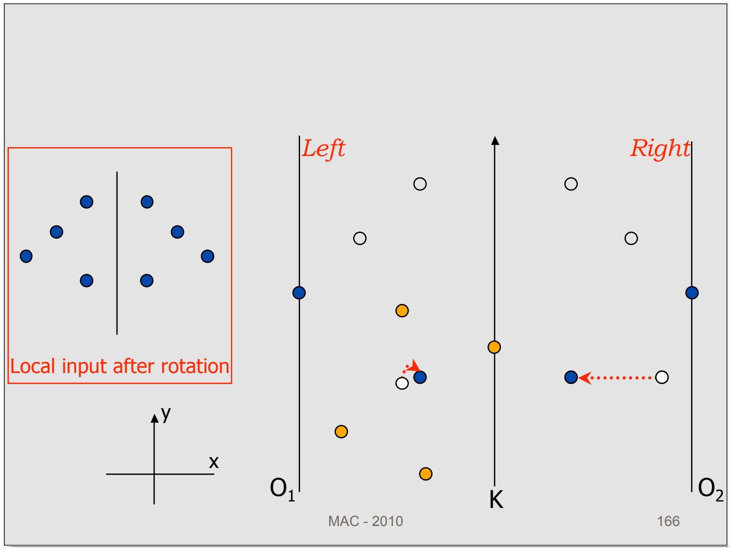
Robots inside Left/Right:

- The *free* robot in my side closest to an empty Final Position in my side, moves towards it
- If there are no Final Positions available in my side, go on K


Local input after rotation

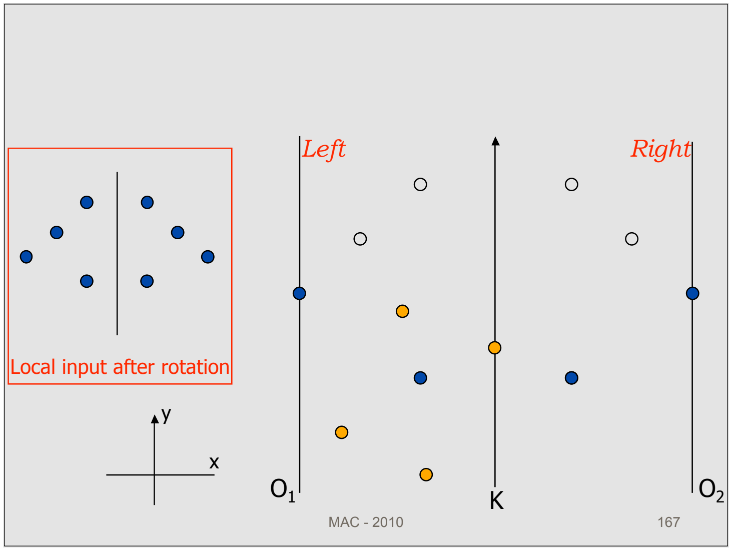
MAC - 2010

165


Local input after rotation

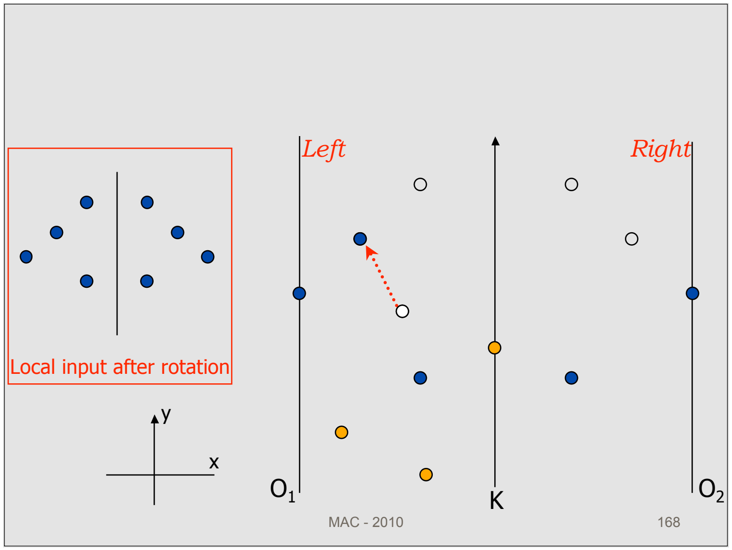
MAC - 2010

166


Local input after rotation

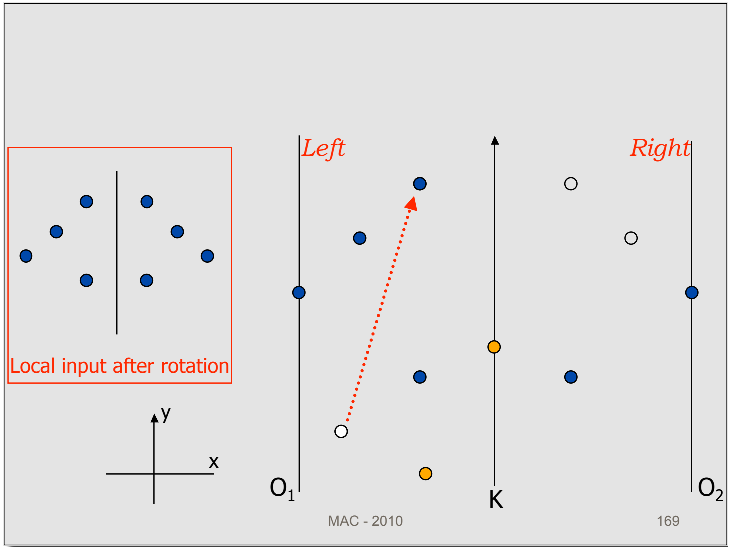
MAC - 2010

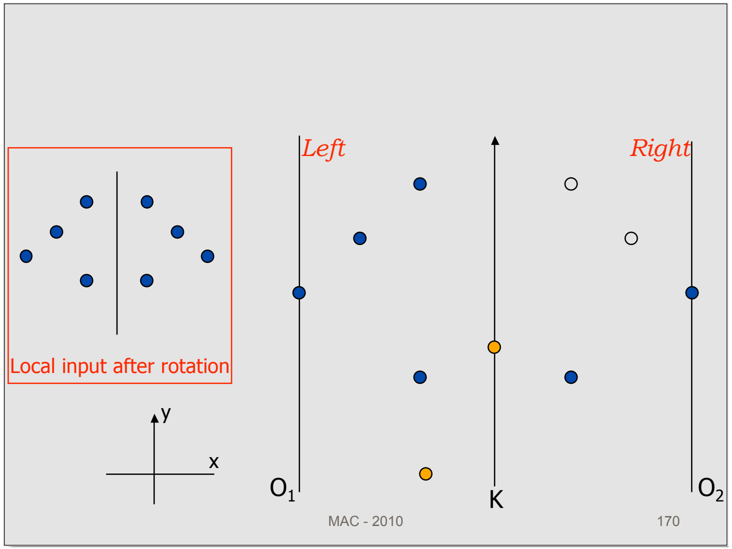
167


Local input after rotation

MAC - 2010

168



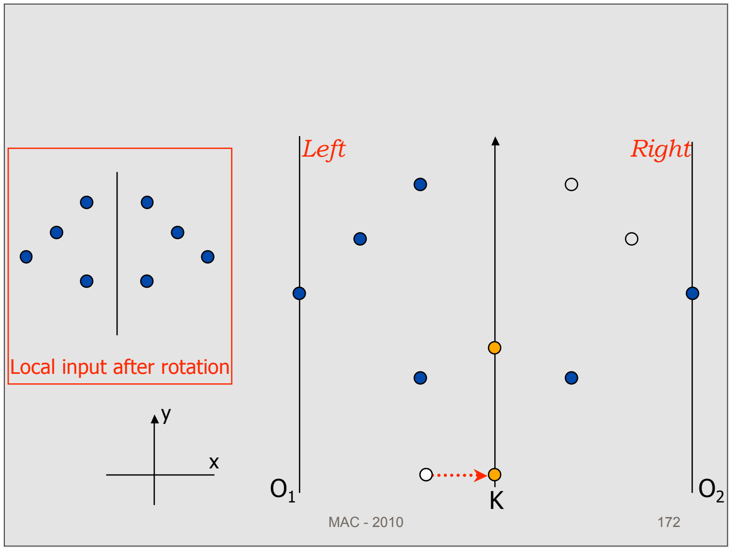


Robots on K:

- Do not move as long as there are robots inside Left/Right not on Final Positions (*free* robots)
- If no *free* robot is inside Left/Right, the topmost on K moves towards one of empty Final Positions

Robots inside Left/Right:

- The *free* robot in my side closest to an empty Final Position in my side, moves towards it
- If there are no Final Positions available in my side, go on K

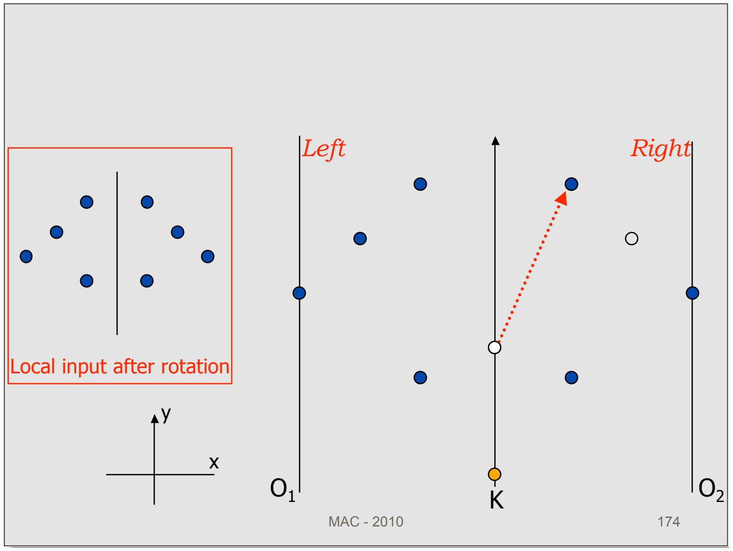


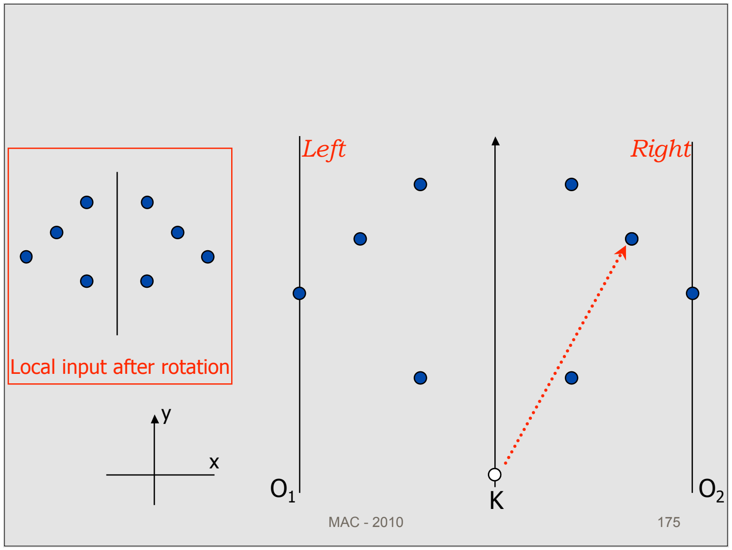
Robots on K:

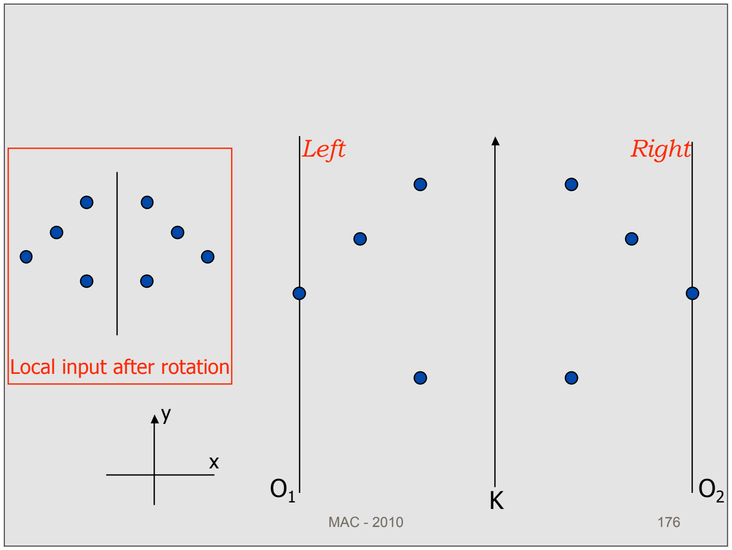
- Do not move as long as there are robots inside Left/Right not on Final Positions (*free* robots)
- If no *free* robot is inside Left/Right, the topmost on K moves towards one of empty Final Positions

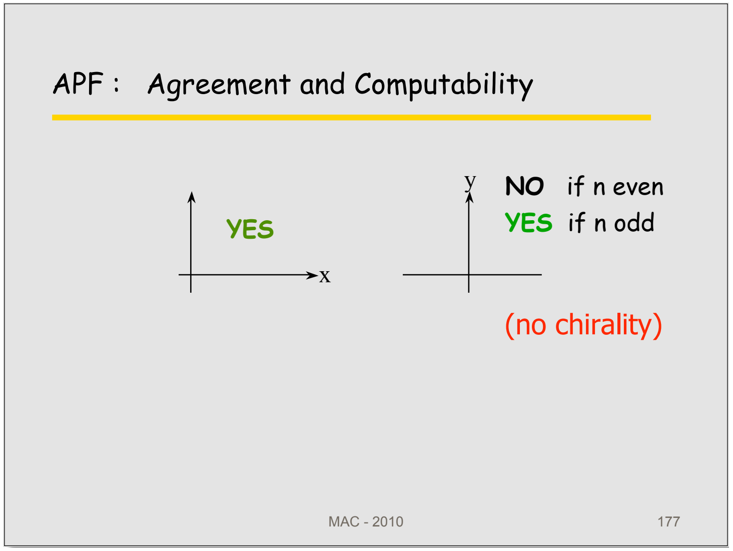
Robots inside Left/Right:

- The *free* robot in my side closest to an empty Final Position in my side, moves towards it
- If there are no Final Positions available in my side, go on K





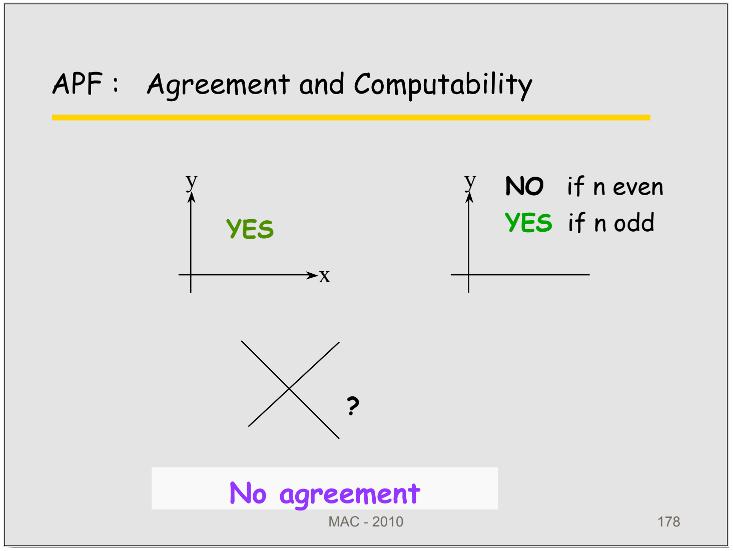




APF : Agreement and Computability

NO if n even
YES if n odd

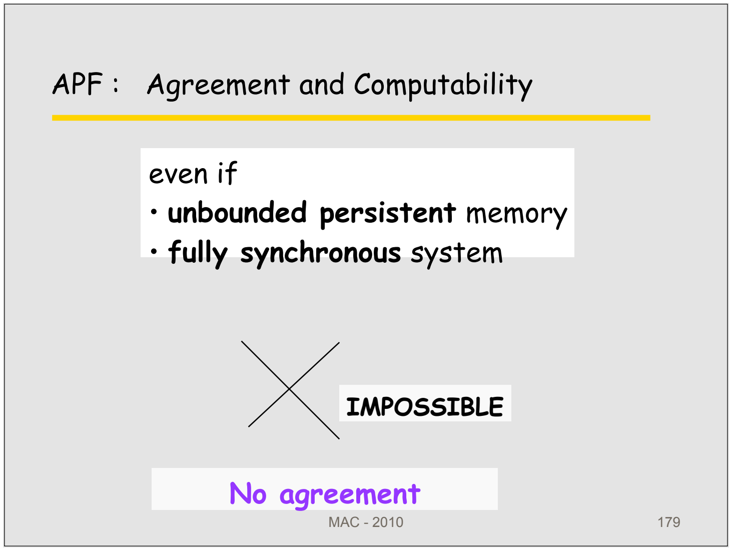
(no chirality)



APF : Agreement and Computability

NO if n even
YES if n odd

No agreement

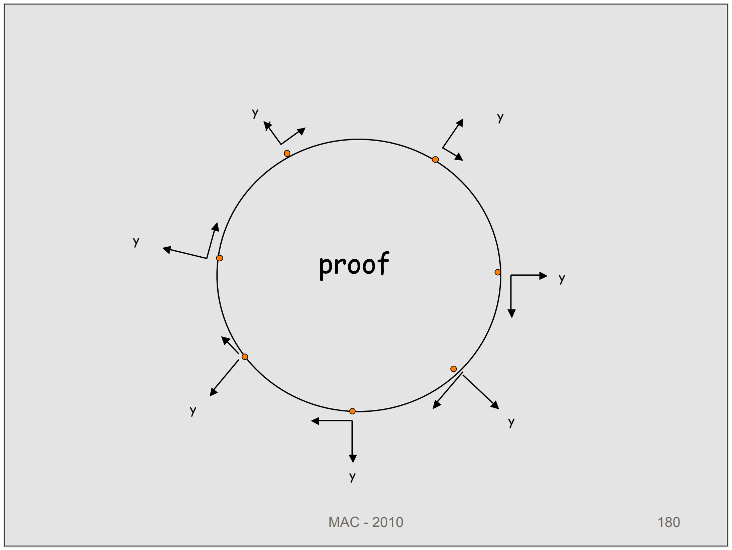


APF : Agreement and Computability

even if
- **unbounded persistent** memory
- **fully synchronous** system

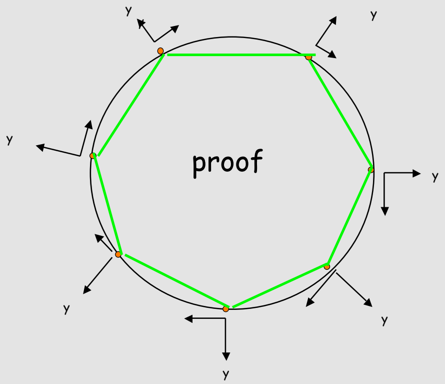
IMPOSSIBLE

No agreement

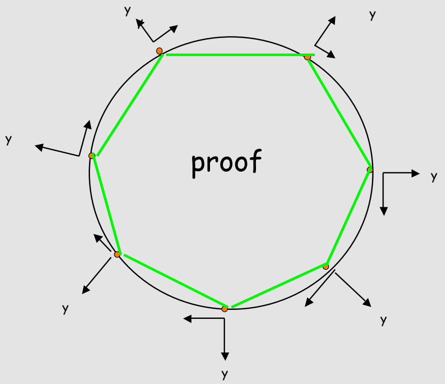


proof

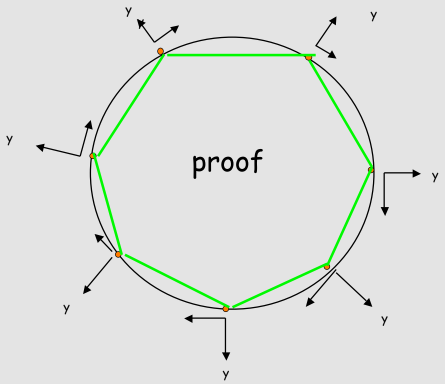
n-GON



any totally synchronous execution will transform the n-GON into another n-GON



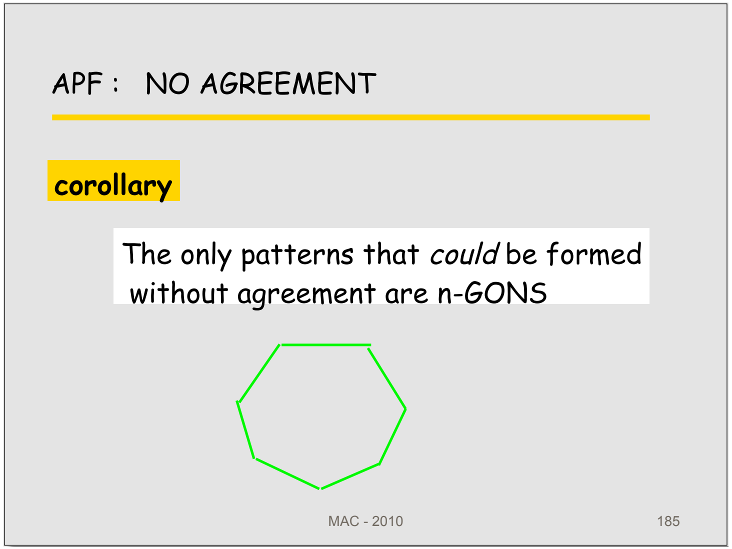
… thus the only patterns that *can* be formed are n-GONS



APF : NO AGREEMENT

even if
- **unbounded persistent** memory
- **fully synchronous** system

╳ **IMPOSSIBLE**



APF : NO AGREEMENT

corollary

The only patterns that *could* be formed without agreement are n-GONS

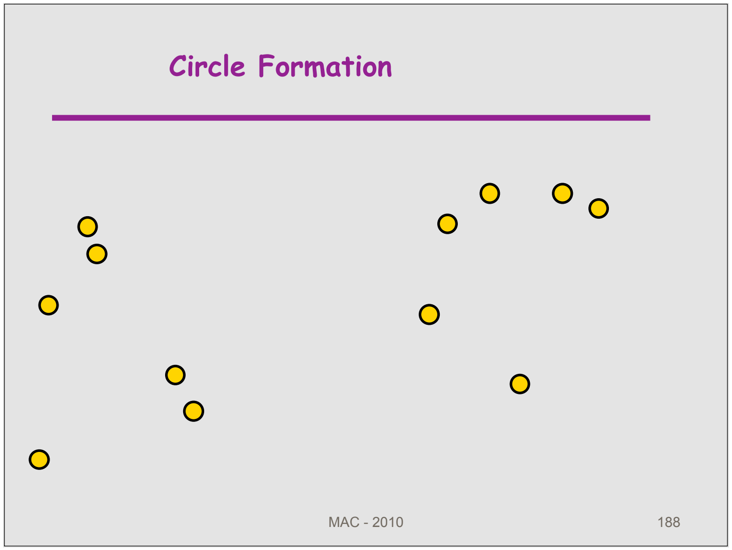
Pattern Formation : no agreement

Q: can n-GONs be formed when there is no agreement ?

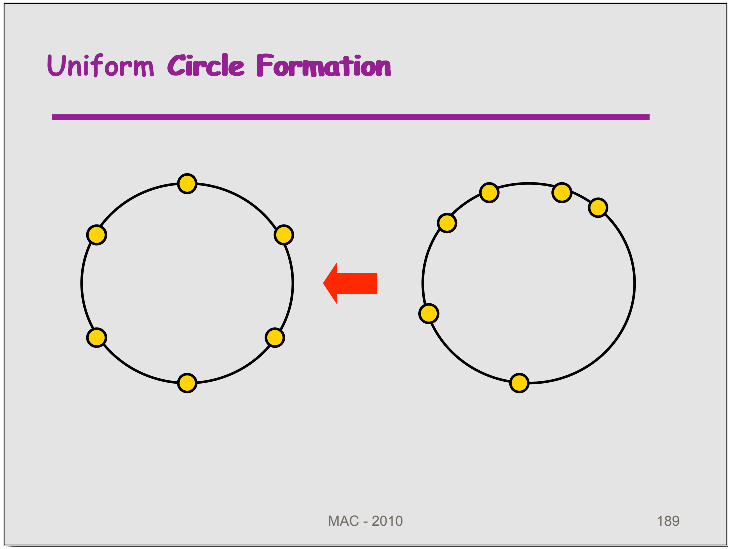
Pattern Formation : no agreement

Q: can n-GONs be formed when there is no agreement ?

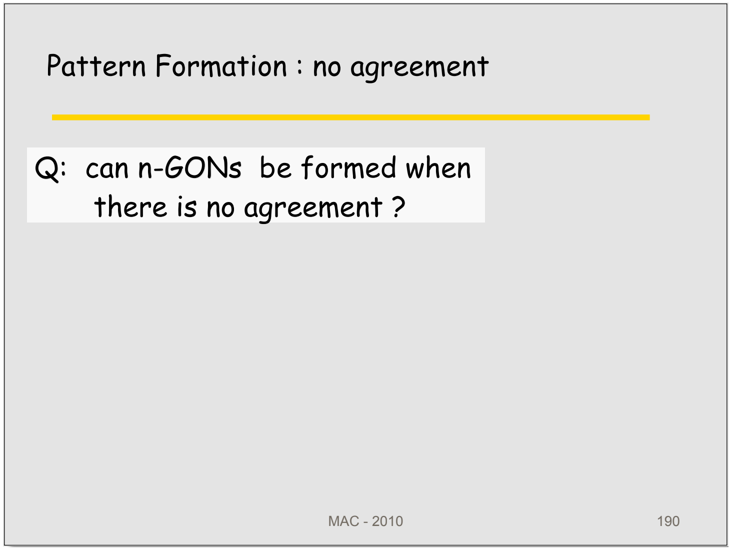
Forming n-gon = **uniform circle formation**



Circle Formation

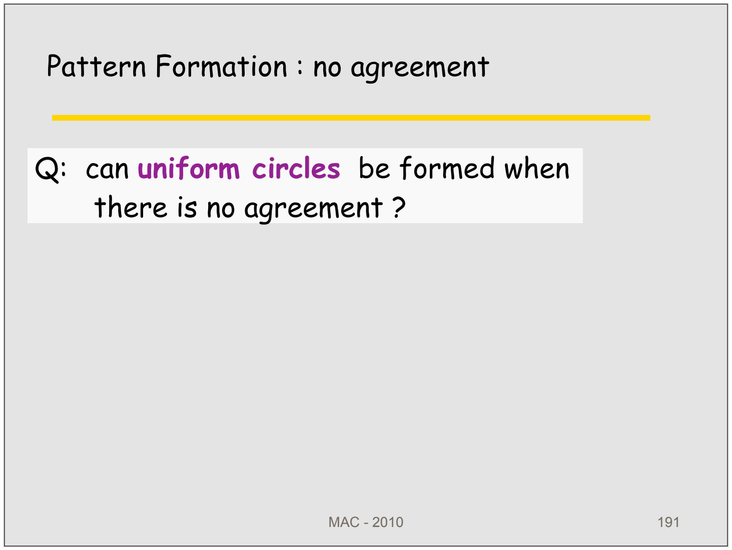


Uniform Circle Formation



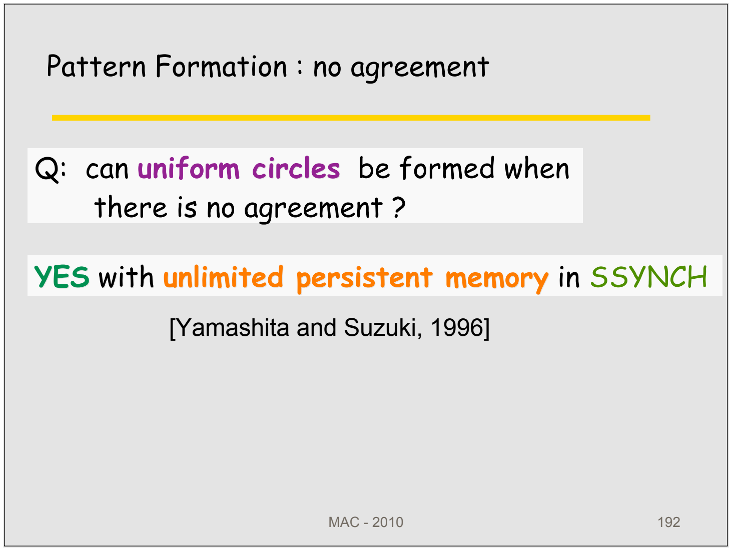
Pattern Formation : no agreement

Q: can n-GONs be formed when there is no agreement ?



Pattern Formation : no agreement

Q: can **uniform circles** be formed when there is no agreement ?

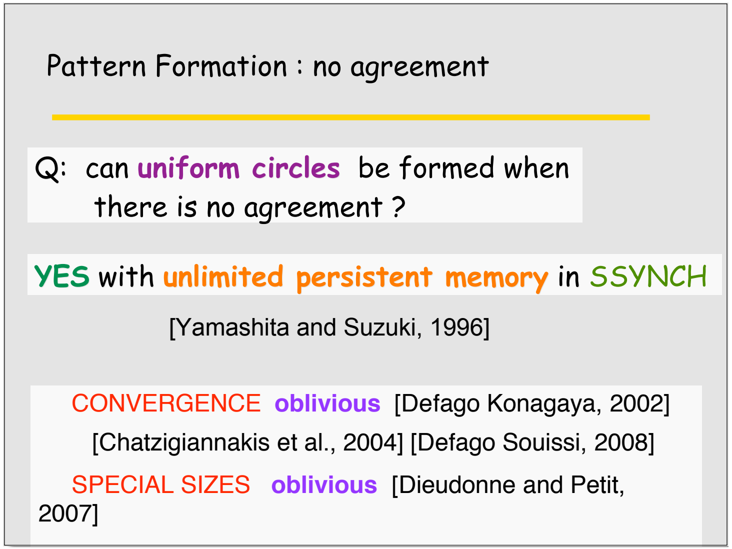


Pattern Formation : no agreement

Q: can **uniform circles** be formed when there is no agreement ?

YES with **unlimited persistent memory** in SSYNCH

[Yamashita and Suzuki, 1996]



Pattern Formation : no agreement

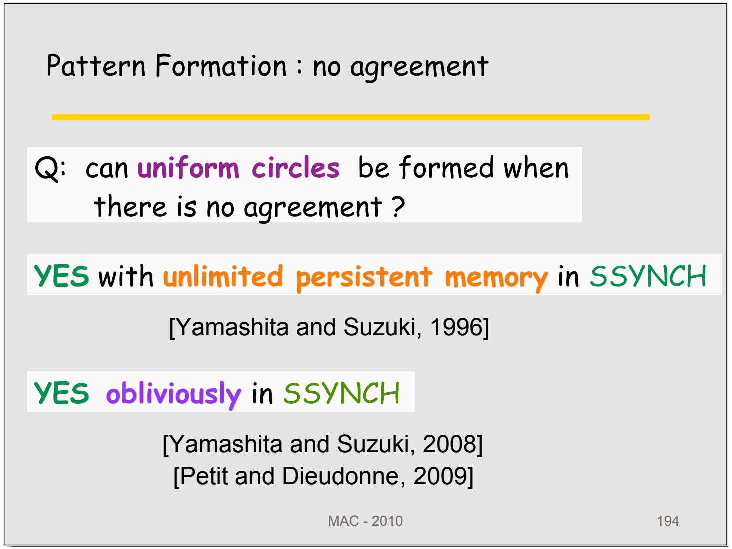
Q: can **uniform circles** be formed when there is no agreement ?

YES with **unlimited persistent memory** in SSYNCH

[Yamashita and Suzuki, 1996]

CONVERGENCE **oblivious** [Defago Konagaya, 2002] [Chatzigiannakis et al., 2004] [Defago Souissi, 2008]

SPECIAL SIZES **oblivious** [Dieudonne and Petit, 2007]



Pattern Formation : no agreement

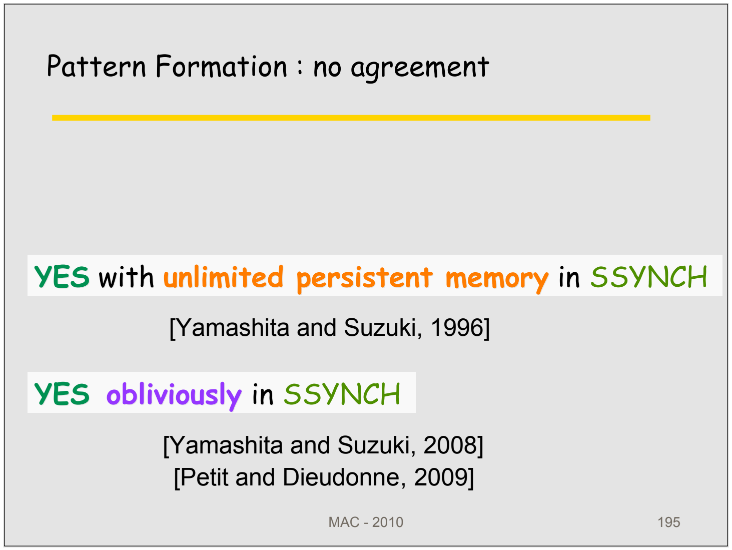
Q: can **uniform circles** be formed when there is no agreement ?

YES with **unlimited persistent memory** in SSYNCH

[Yamashita and Suzuki, 1996]

YES **obliviously** in SSYNCH

[Yamashita and Suzuki, 2008]
[Petit and Dieudonne, 2009]



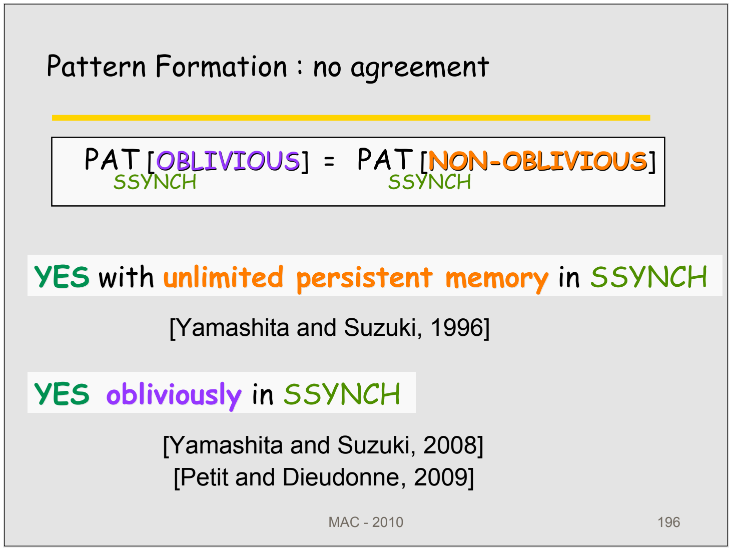
Pattern Formation : no agreement

YES with **unlimited persistent memory** in SSYNCH

[Yamashita and Suzuki, 1996]

YES **obliviously** in SSYNCH

[Yamashita and Suzuki, 2008]
[Petit and Dieudonne, 2009]



Pattern Formation : no agreement

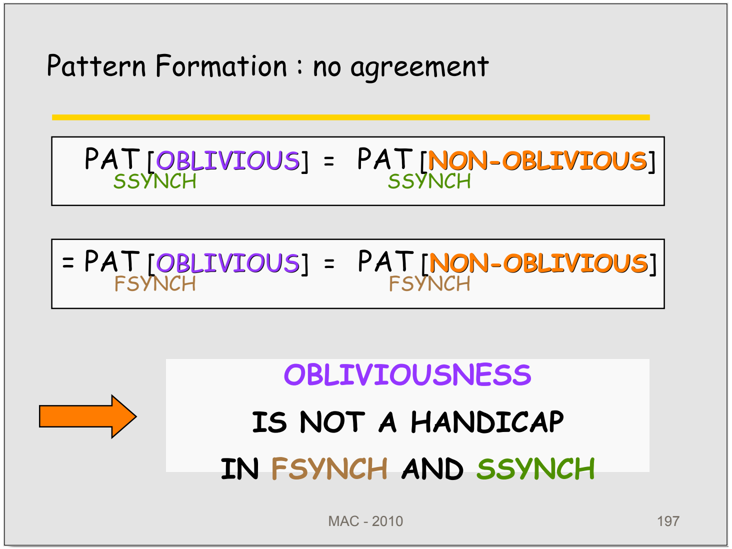
$$PAT_{SSYNCH}[OBLIVIOUS] = PAT_{SSYNCH}[NON\text{-}OBLIVIOUS]$$

YES with **unlimited persistent memory** in SSYNCH

[Yamashita and Suzuki, 1996]

YES **obliviously** in SSYNCH

[Yamashita and Suzuki, 2008]
[Petit and Dieudonne, 2009]



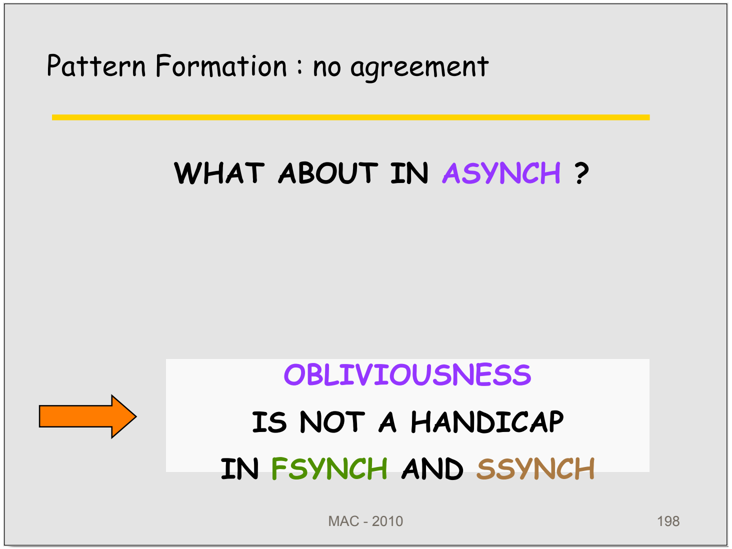
Pattern Formation : no agreement

$PAT[\text{OBLIVIOUS}]_{\text{SSYNCH}} = PAT[\text{NON-OBLIVIOUS}]_{\text{SSYNCH}}$

$= PAT[\text{OBLIVIOUS}]_{\text{FSYNCH}} = PAT[\text{NON-OBLIVIOUS}]_{\text{FSYNCH}}$

OBLIVIOUSNESS
IS NOT A HANDICAP
IN FSYNCH AND SSYNCH

MAC - 2010 197

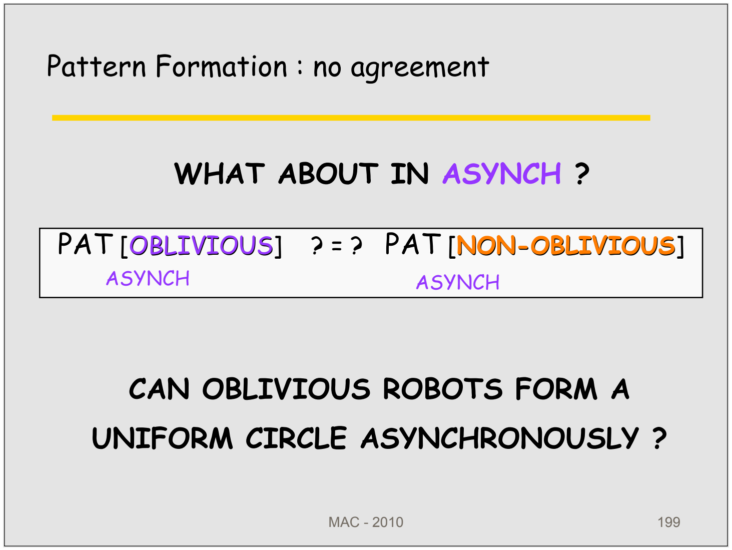


Pattern Formation : no agreement

WHAT ABOUT IN ASYNCH ?

OBLIVIOUSNESS
IS NOT A HANDICAP
IN FSYNCH AND SSYNCH

MAC - 2010 198



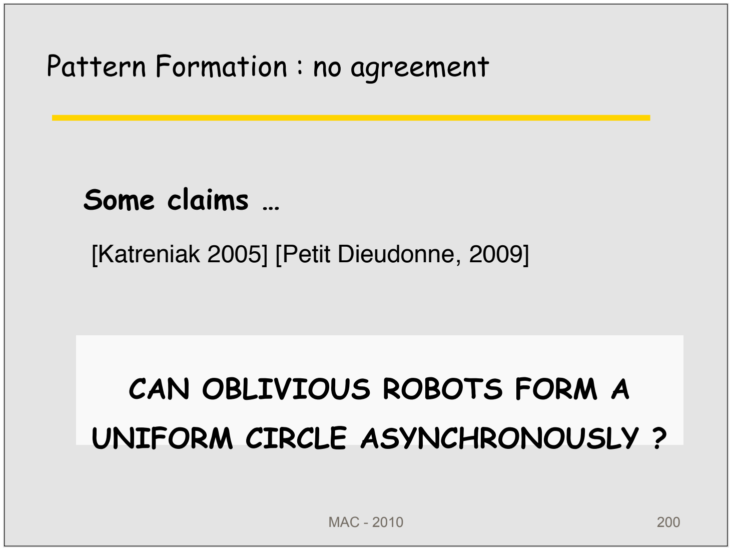
Pattern Formation : no agreement

WHAT ABOUT IN ASYNCH ?

$PAT[\text{OBLIVIOUS}]_{\text{ASYNCH}} \; ?=? \; PAT[\text{NON-OBLIVIOUS}]_{\text{ASYNCH}}$

CAN OBLIVIOUS ROBOTS FORM A
UNIFORM CIRCLE ASYNCHRONOUSLY ?

MAC - 2010 199



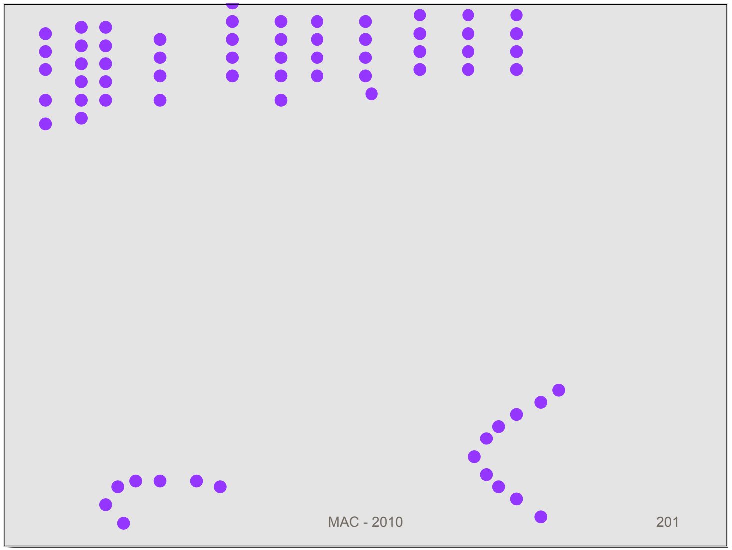
Pattern Formation : no agreement

Some claims ...

[Katreniak 2005] [Petit Dieudonne, 2009]

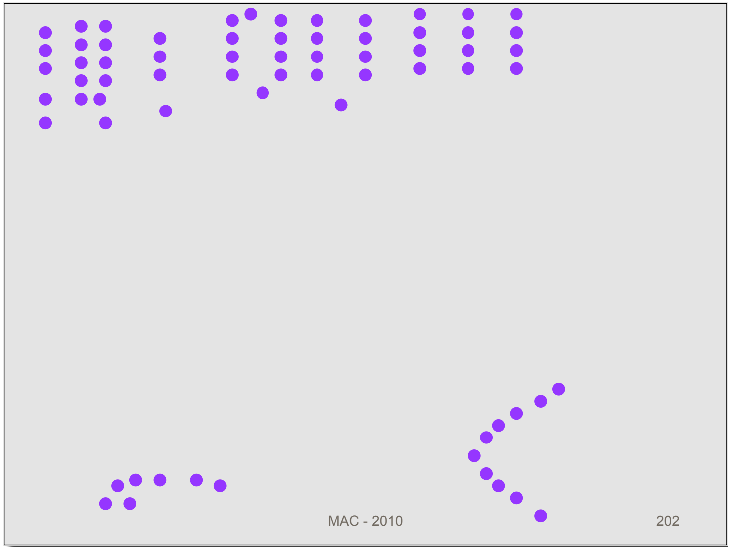
CAN OBLIVIOUS ROBOTS FORM A
UNIFORM CIRCLE ASYNCHRONOUSLY ?

MAC - 2010 200



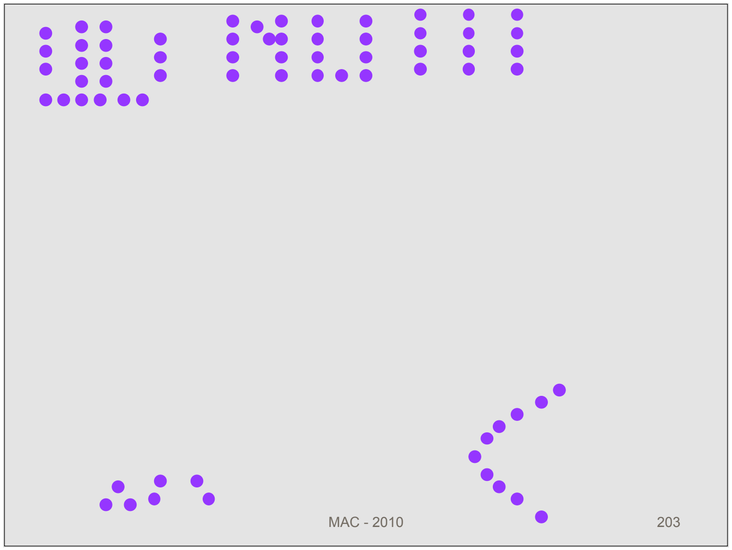
MAC - 2010

201



MAC - 2010

202



MAC - 2010

203