Towards a Theory of Formal Distributed Systems

Why and how distributed systems can solve distributed problems?

Towards = immature or not ready for presenting
Formal = unrealistic or useless

Masafumi Yamashita (Kyushu Univ.)

Table of Contents

1. What is a Distributed System?
2. Distributed Views
3. Formal Distributed Systems
4. Simulating Other Distributed Computing Models
5. Self-Organization
6. Symmetry Breaking
7. Localization
8. Conclusions
9. Open Questions

What is a Distributed System?

- Lamport & Lynch in 1990:

  "Although one usually speaks of a distributed system, it is more accurate to speak of a distributed *view* of a system."

  E.g., a sequential computer is a distributed system for a hardware designer.

- Observations

  Every system has many distributed views (e.g., protocol stack).

  Completely different systems can have essentially the same distributed view.

Proposed Research Area:

Investigate abstract distributed views, Independently of actual systems.

Approach

- Typical Approach
  - Given a target distributed system and a target distributed problem.
  - Construct a model of the system.
  - Investigate the problem under the system model.
- Our Approach
  - Propose a formal model of general distributed view.
  - Construct a formal theory of the model

    like the theory of formal languages.

Distributed Views (Conflict Resolution) 1

- Presidential Election: Ms. Clinton vs. Mr. Trump
  - Candidates have unique names, which are not sufficient.
    If both get 50% votes, perhaps the candidates need to cast lots.
- Airplanes avoiding near misses
  - Planes have unique names.
    The air traffic controller controls the route of each plane.
- Vehicles crossing intersections
  - Vehicles have unique names, but uniqueness is not used.
    Traffic lights locally control their flows.
- Mutual exclusion among processes
  - The processes have unique names from totally ordered set.
    Each mutual exclusion algorithm uses this fact.

Distributed Views (Conflict Resolution) 2

- Seabirds in a small island
  - Competing for good nesting places like vehicles looking for parking space.
- Harem of sealions
  - The strongest male wins like fighters in dogfight.
- Molecules of water
  - Oxygens compete for the position in a molecule of water like team assembling by autonomous robots.

We investigate distributed views after abstraction.

Formal Distributed System (FDS) : a model of abstract distributed views (not an abstraction of the whole system).

Formal Distributed System 1

- FDS = Interacting (distributed) elements + Interaction model.
  - An extension of mobile robot model.
  - FDSs must model natural systems:

    We allow incomputable ``distributed algorithms.''

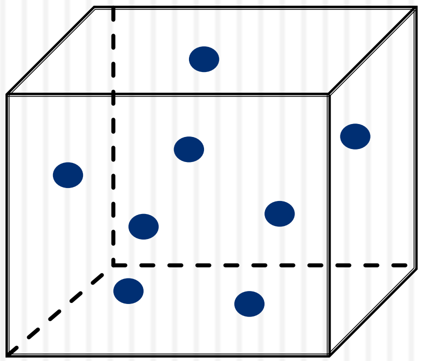
    Natural systems can ``compute'' incomputable function,

    since they behave according to physical/chemical laws.

    E.g., ``Go to geometric median'' can be a gathering algorithm.

MY DREAM: Construct a theory of FDSs.

Formal Distributed System 2



Interacting Elements:

Points in a *d*-dimensional Euclidean (sub-)space.

Interaction Model:

- Scheduler: Determine when an element interacts.
  FSYNC, SSYNC, ASYNC, central
  deterministic (adversary)/randomized
- Interaction rule: Determine how an element behaves.
  (Local state, Local snapshot, Transition function)

Fault Model

Why Elements of FDS are Points?

- Why points?
  - In graph theory, e.g., cities are modeled by points.
  - In mechanics, Sun and Earth are mass points in 3-D space.
  - In many distributed models, distributed elements are points.
- Why higher than 3-D space needed?
  - Configuration space of a multi-link arm.
  - To simulate a graph network by an FDS (or wireless network).

    Any graph can be represented by an intersection graph of *d*-D balls.
  - Curiosity

Taxonomy of Elements 1

Elements can be classified by the following concerns:

- Element types
  - An FDS can be heterogeneous.
- Identifiers
  - Unique identifiers from a totally ordered set
  - Unique identifiers from an unordered set
  - Identifiers which may not be distinct
  - Anonymous
- Memories (local variables)
  - Internal (local) memory

    Infinite size, constant size, oblivious
  - Visible (accessible) memory for communication

    Message, light, beep, smell, …

Taxonomy of Elements 2

- Local snapshot: all what element can sense.
  - Visibility range?

    Full visibility, limited visibility
  - What are sensible?

    Location, velocity, visible memory, energy, …
  - How to describe?

    Local coordinate system, chirality, multiplicity detection ability, …
- Transition function
  - Input: Local state and local snapshot
  - Output: New local state and the route to the next destination
  - Not necessarily computable
- Travel
  - Rigid, non-rigid

How Elements Interact?

Interaction Cycle (on each element)

1. Scheduler activates the element.
2. The element takes the following actions:
   - Take the local snapshot, which is an atomic action.
   - Take local action specified by the transition function, which may take a long time.
3. Repeat until forever.

Notes:

- Transition function may be given by an oracle.
- Local action: update local variables and traverse a designated route.

Simulating Other Models 1

[Modeling ability of FDS]

- Wireless computer network:
  - Computers in *d*-D space: stationary elements in *d*-D space
  - Broadcast radius: visibility range
  - Message: data in a visible memory (e.g., lights)
- Point-to-point computer network (graph model):
  - Can be simulated by wireless computer network.
    Any graph can be represented by an intersection graph of *d*-D balls.
- Shared memory distributed system:
  - Can be simulated by oblivious mobile robot with full visibility.

Simulating Other Models 2

- Mobile robot [Suzuki&Yamashita '96]:
  - A class of FDSs
    Anonymous, no visible memory, no agreement on location and time
- Mobile agents on graph [Klasing et.al '08]:
  - Can be simulated by mobile robot model
    The destination is selected from a set of points specified in advance.
    The travel to a destination is an atomic action.
- Beeping network [Cornejo&Kuhn '10], Stone-age network [Emek et.al '13], Cellular automaton in *d*-D lattice space:
  - Anonymous systems with weak communication mechanisms
- Mass points under Newtonian mechanics:
  - Global time, global coordinate system, rigid move, velocity and acceleration are visible variables, mass is type.
  - Transition function represents the laws of physics.

Simulating Other Models 3

- Population protocol model [Angluin et al. '06]:
  Simulating bidirectional synchronous communication between anonymous elements by FDSs.
  - Assume central scheduler + full visibility + visible constant memory.

[Element e]
If no lights on, change light blue.

[Element e']
If it finds blue, change light green.
(Communicate with e.)

[Element e]
If it finds green, change light red.
(Communicate with e'.)

[Element e']
If it finds red, turn off light.

[Element e]
If no lights on, turn off light.

Simulating Other Models 4

- Rough Conclusion:

FDSs can model sufficiently wide variety of distributed views.

OPEN QUESTIONS:

- Extend FDSs so that they can describe environment.
- Are FDSs universal?

Distributed Problems

1. Self-Organization
2. Symmetry breaking
3. Localization

- Global snapshot
- Synchronization
- Searching mobile intruders
- Fault tolerance

Self-Organization 1

- Comparing natural and artificial distributed systems:

Natural Systems
1. Anonymous
2. Memory-less
3. Asynchronous
4. Fluctuation

Artificial Systems
1. Unique IDs from ordered set
2. Memory available
3. Synchronous
4. Deterministic

- Artificial systems enjoy the existence of infrastructure.

Question: Why implementing self-organization in artificial distributed systems difficult?

Self-Organization 2

- Model: Mobile robot in 2-D space (a class of FDSs)
  - Identifiers: unique IDs/anonymous
  - Memory: non-oblivious/oblivious
  - Scheduler: FSYNC/ASYNC (SSYNC)
  - Transition function: deterministic/probabilistic
  - Visibility range: full visibility
  - Local coordinate systems: with chirality
  - Interaction cycle: Look-Compute-Move cycle
- Definitions:

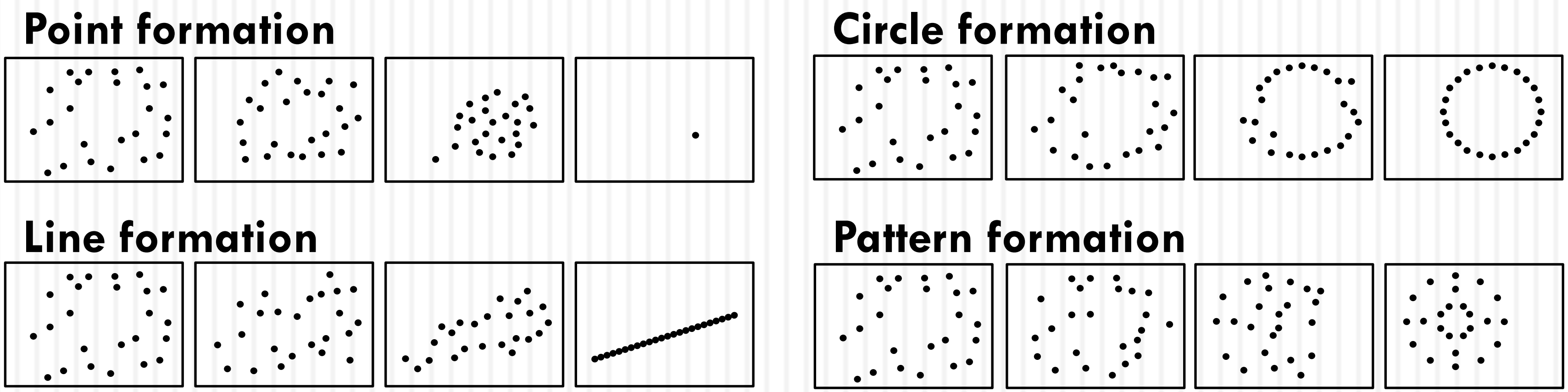
  Self-organization = self-stabilizing pattern formation

  Self-stabilization = tolerate finite number of transient failures

  Transient failure = change the position to a random location

Self-Organization 3

- Pattern formation problem in 2D space



(Yamauchi)

A pattern *F* may not be formable from every initial configuration *I*.
Sym(*F*) is divisible by Sym(*I*) is necessary and (roughly) sufficient.
[Fujinaga et al. '15]

Sym(*P*) = (roughly) the order of rotation group of *P*

Self-Organization 4

[Theorem] Deterministic, non-oblivious, FSYNC, anonymous robots and deterministic, oblivious, ASYNC, anonymous robots have the same pattern formation ability, except rendezvous. [Fujinaga et al. '15]

- Memory and synchrony do not help in pattern formation.

[Theorem] Deterministic, oblivious, SSYNC, anonymous pattern formation algorithm is a self-stabilizing algorithm. [Suzuki et al. '99]

- Obliviousness and anonymity help in self-stabilization.

  Memory and unique IDs are harmful in self-stabilization.

[Theorem] Probabilistic, oblivious, ASYNC, anonymous robots can form any pattern from any initial configuration with probability 1. [Yamauchi et al. '14]

- Probabilistic algorithm simulates fluctuation in nature and remove the restriction caused by anonymity.

[Corollary] There is a probabilistic self-organizing algorithm for oblivious, SSYNC, anonymous robots that forms any pattern with probability 1.

Self-Organization 5

- Rough Conclusion:

Natural systems have more properties suitable to make them self-organizing than artificial systems.

As long as the mobile robots in 2D space are concerned.

Pattern formation in 3D space will appear in PODC'16 [Yamauchi et al.'16]

Note: Asynchrony governed by adversary does not help. However, it helps if it is governed by a random scheduler:

[Corollary] There is a deterministic self-organizing algorithm for oblivious, SSYNC, anonymous robots that forms any pattern with probability 1.

Symmetry Breaking 1

Question:

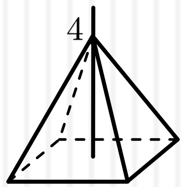
What is symmetry and what is symmetry breaking?

- Anonymous network (Angluin's model):
  - Symmetry is based on covering concept.
- Anonymous network (YK model):
  - Vertex election is possible iff Sym(G) = 1.
  - Symmetry is based on automorphism group of graph.
- Mobile robots in 2D space:
  - F is formable from I iff Sym(*F*) is divisible by Sym(*I*).
  - Symmetry is based on rotation group.

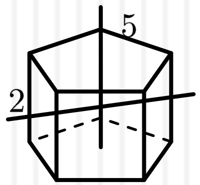
The impossibilities arise from symmetry among elements,
and cannot be overcome by using incomputable functions.

Symmetry Breaking 2

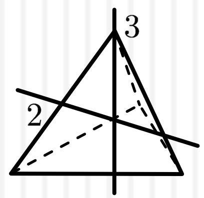
- Symmetry among robots (with chirality) in 3D space:



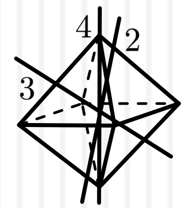
Cyclic group C_k



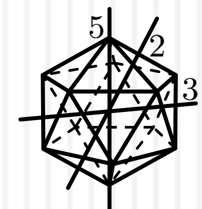
Dihedral group D_k



Tetrahedral group T



Octahedral group O

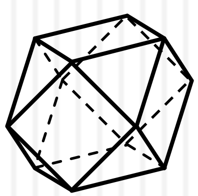


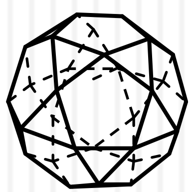
Icosahedral group I

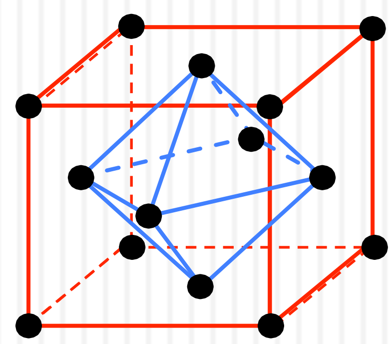
(Yamauchi)

- A configuration can be decomposed Into several vertex-transitive polyhedra.

[5 regular, 13 semi-regular and other polyhedra]

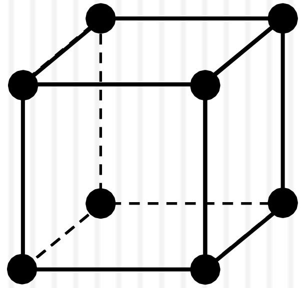






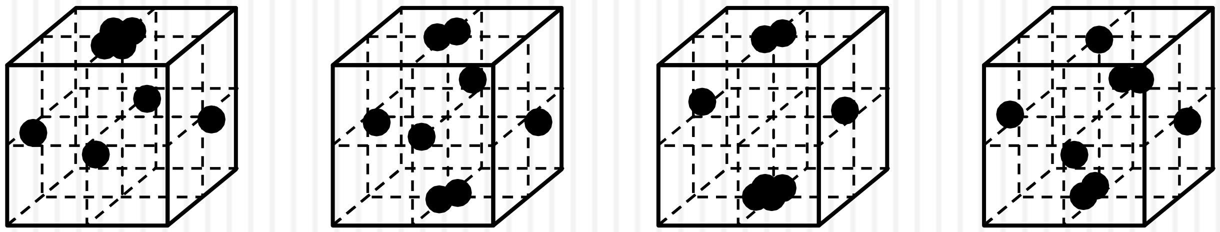
Symmetry Breaking 3

- How to break symmetry in cube?
  - 8 vertices (robots)
  - 6 faces



Go-to-center algorithm

Robot selects an adjacent face and approaches the center,
but stops ε before the center.



(Yamauchi)

Symmetry Breaking 4

- Rotation group = arrangement of rotation axes.
  - Symmetry breaking = reduction of rotation axes.

Octahedral group O

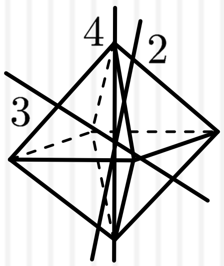
Rotations on regular octahedron

Order 24

6 2-fold axes

4 3-fold axes

3 4-fold axes



Robots on 4-fold axes can remove them by leaving them.

The other axes are not removable.

Symmetry Breaking 5

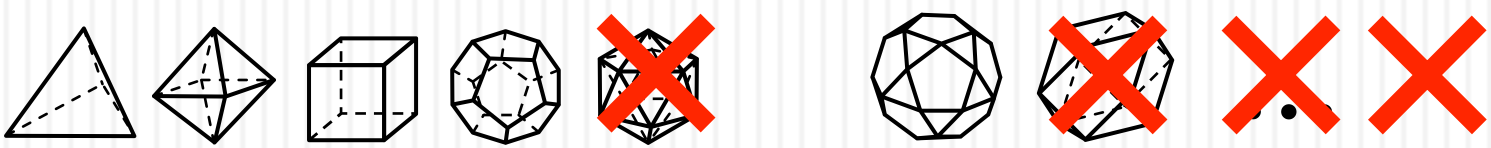
- Rough Conclusion:

Consider deterministic, oblivious, FSYNC, anonymous mobile robots in 3D space. They can remove a rotation axis of the group that acts on the initial configuration if and only if it includes vertices (robots).

- Plane Formation Problem (example):

5 Regular polyhedra

13 Semi-regular polyhedra

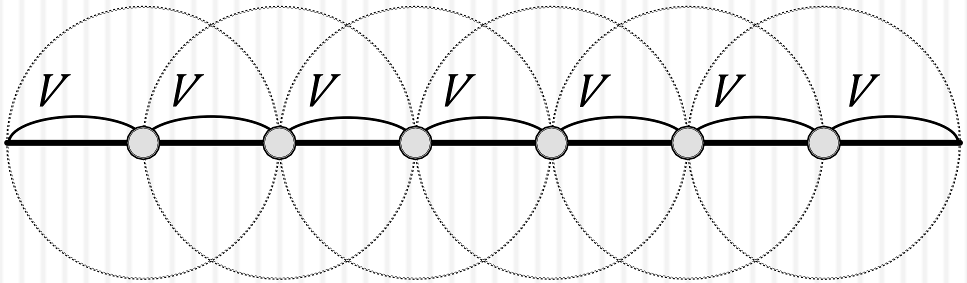


[Yamauchi et al. '15]

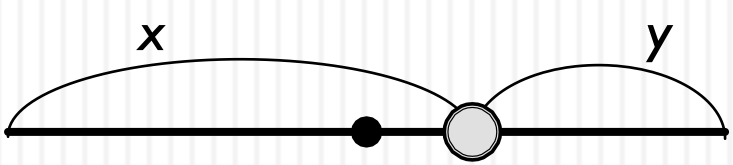
Similar result holds for anonymous networks.

Localization 1

- Locate k oblivious FSYNC robots with visibility radius V equally divided points in line segment of length $D = (k+1)V$.



  - When $k = 1$,



define transition functions L(eft) and R(ight) that work for any $D = 2V$.

In this case, R(y) is its new position from the right end.

Localization 2

Question:

What is the impact of the complexity of transition functions?

- The following results will appear in Mac '16 [Monde et al. '16]
  - D is real and both L and R are real-valued functions:
    Solvable but we need 1 bit of memory in our solution.
  - D is rational and L and R are real-valued functions:
    Solvable but at least L or R is not computable in our solution.
  - D is rational and both L and R are rational-valued functions:
    Solvable but we need 1 bit of memory in our solution.
  - D is integral, then solvable by integral-valued functions L and R.

Conclusions

- Propose a new research area of FDSs.
- Propose a candidate for a general model of FDSs.
  - Elements are points and communication is by interaction cycle.
- Discuss three research topics.
  - Self-organization
  - Symmetry breaking
  - Localization

Open Questions 1

- Model of FDSs:
  - How to extend our model to include environment?
  - How to simulate synchronous communication when scheduler is not central?
- Self-Organization:
  - What is the impact of limited visibility?
  - Can randomness bury the gap between full and limited visibilities?
- Symmetry Breaking:
  - ASYNC symmetry breaking algorithm for 3D robots.
  - How to characterize removable rotation axes in 4D or higher space?
  - Can memory help in symmetry breaking?

Open Questions 2

- Localization:
  - What is the impact of memory in localization?
  - Can we define a hierarchy of localization problem classes in terms of the difficulty of transition functions?
- Genereral
  - Relation with information theory. Information theory analyzes the amount of information. Can we state besides quantity?
  - Relation with computation theory. Can distributed computing allowing incomputable distributed algorithms add some new perspectives? My conjecture is Yes, and this is a purpose of this talk.

Thank You Very Much!

- Colleagues:

Tiko Kameda, Ichiro Suzuki, Paola Flocchinni, Nicola Santoro, Shuji Kijima, Yukiko Yamauchi, …