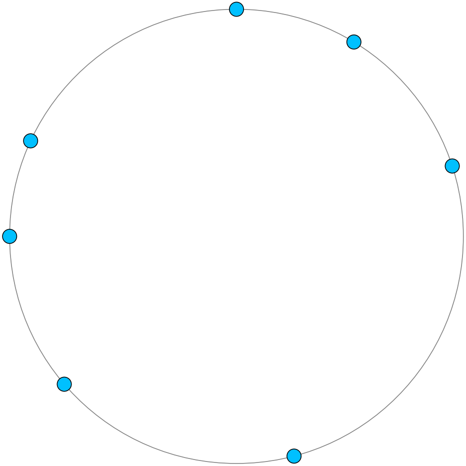
Gathering on a Circle with Limited Visibility by Anonymous Oblivious Robots

(DISC 2020)

Giuseppe A. Di Luna, Ryuhei Uehara,
Giovanni Viglietta, and Yukiko Yamauchi

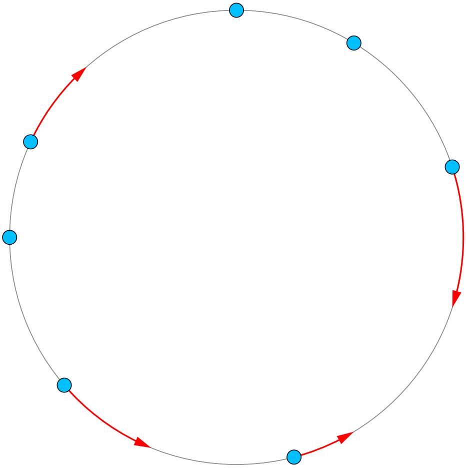
October 15, 2020

Gathering on a circle with limited visibility



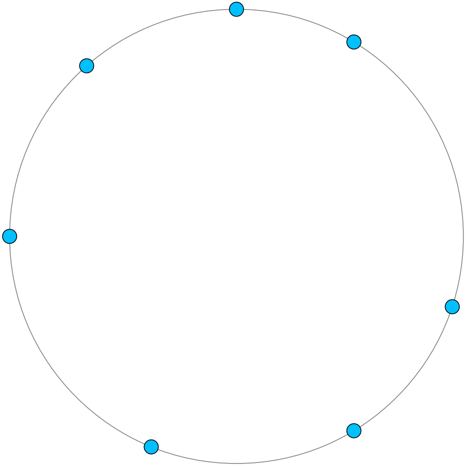
Setting: a team of *robots* on a circle, initially at distinct locations.

Gathering on a circle with limited visibility



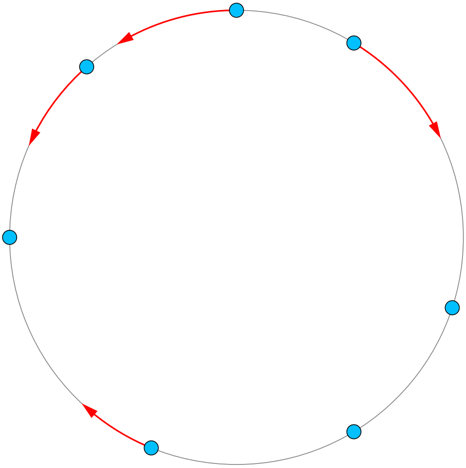
Robots can only move along the circle.

Gathering on a circle with limited visibility



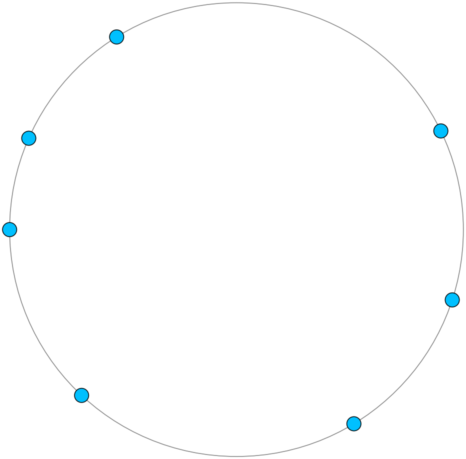
At every time unit, an adversarial (semi-synchronous) *scheduler* decides which robots are active and which are inactive.

Gathering on a circle with limited visibility



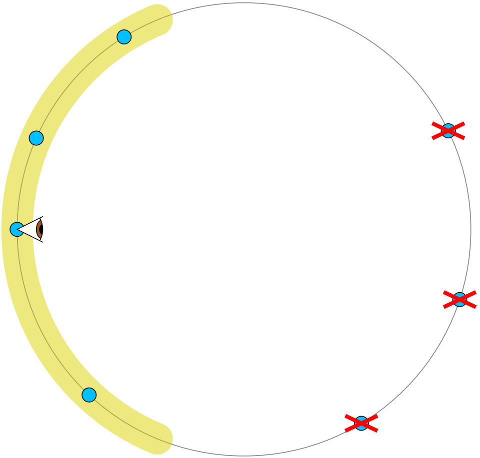
At every time unit, an adversarial (semi-synchronous) *scheduler* decides which robots are active and which are inactive.

Gathering on a circle with limited visibility



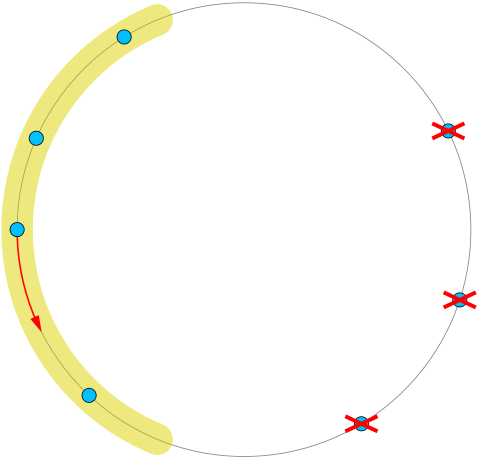
At every time unit, an adversarial (semi-synchronous) *scheduler* decides which robots are active and which are inactive.

Gathering on a circle with limited visibility



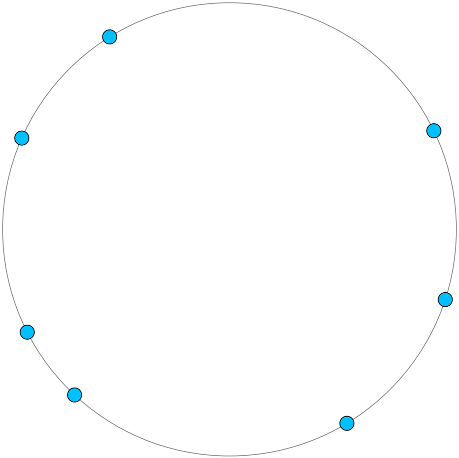
A robot can see other robots only within a fixed range.

Gathering on a circle with limited visibility



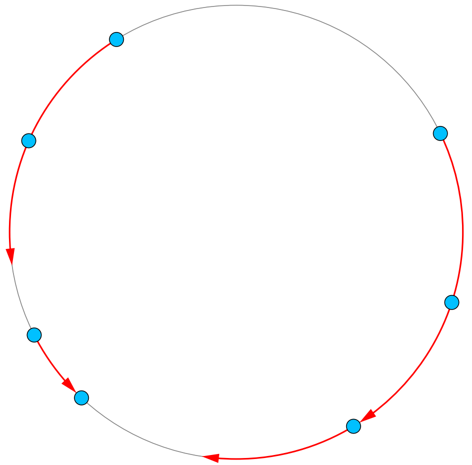
Its destination point is determined based on the *visible robots* only.

Gathering on a circle with limited visibility



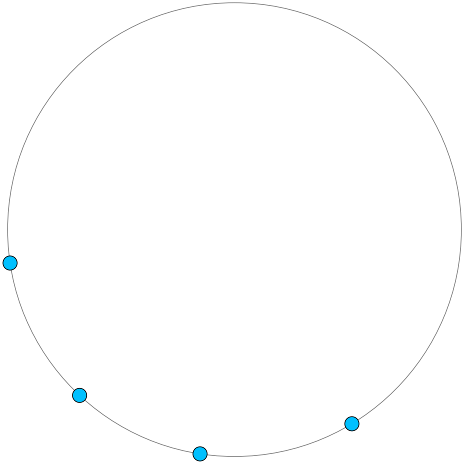
Its destination point is determined based on the *visible robots* only.

Gathering on a circle with limited visibility



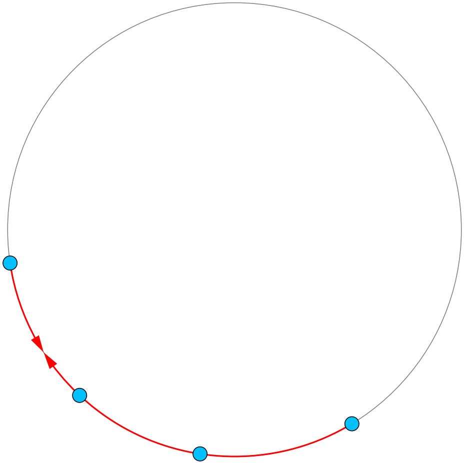
Goal of the team: eventually gather in a point and stop moving.

Gathering on a circle with limited visibility



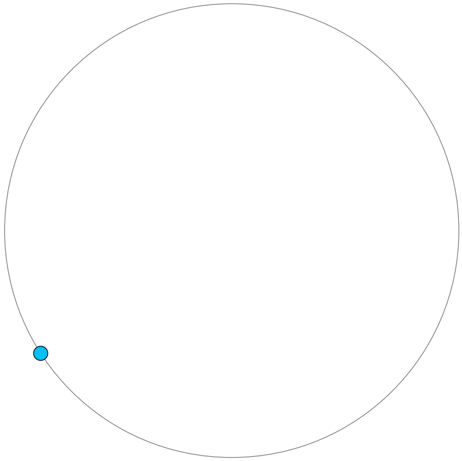
Goal of the team: eventually gather in a point and stop moving.

Gathering on a circle with limited visibility



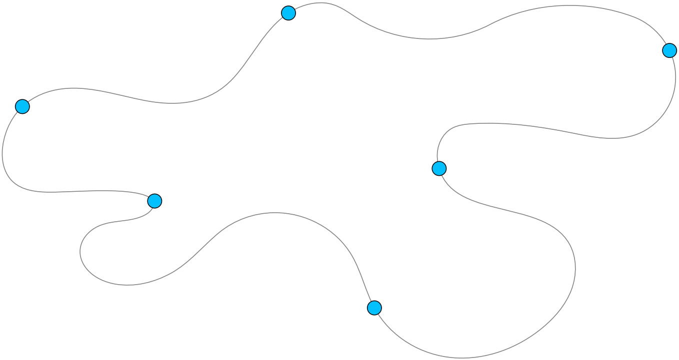
Goal of the team: eventually gather in a point and stop moving.

Gathering on a circle with limited visibility



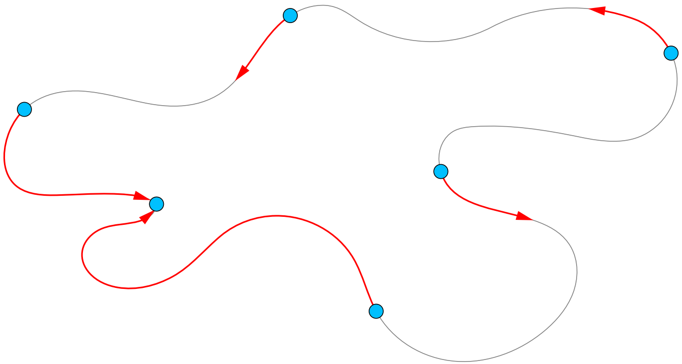
A gathering algorithm should be successful no matter how the adversarial scheduler decides to activates the robots.

Gathering on a circle with limited visibility



A gathering algorithm for the circle extends to any *closed curve*.

Gathering on a circle with limited visibility



The circle is the hardest curve for gathering, because all its points are equivalent, with no “landmarks” that may help orientation.

Outline

- Model definition

- If each robot sees less than *half a circle*:

  Gathering is unsolvable

- If each robot sees the whole circle except its *antipodal point*:

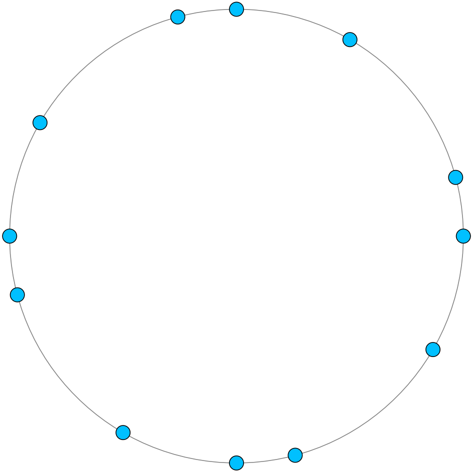
  There is a gathering algorithm

Model definition

Robots are:

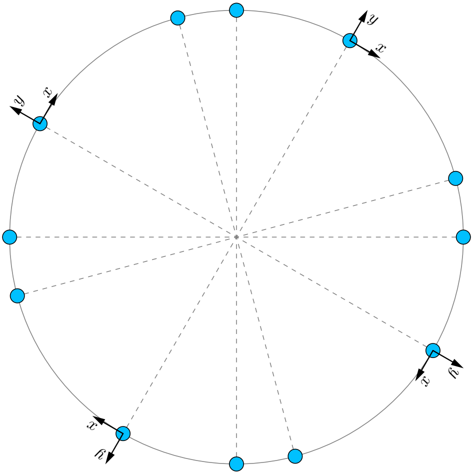
- **Dimensionless** (robots are modeled as geometric points)
- **Anonymous** (no unique identifiers)
- **Homogeneous** (the same algorithm is executed by all robots)
- **Deterministic** (robots cannot use randomization)
- **Disoriented** (robots do not share a common reference frame)
- **Autonomous** (no centralized control)
- **Semi-Synchronous** (robots may occasionally skip turns)
- **Oblivious** (no memory of past events and observations)
- **Silent** (no explicit way of communicating)
- **Short-sighted** (visibility of other robots limited to a range)
- **Unknowing** (no knowledge of the total number of robots)

Rotationally symmetric configurations



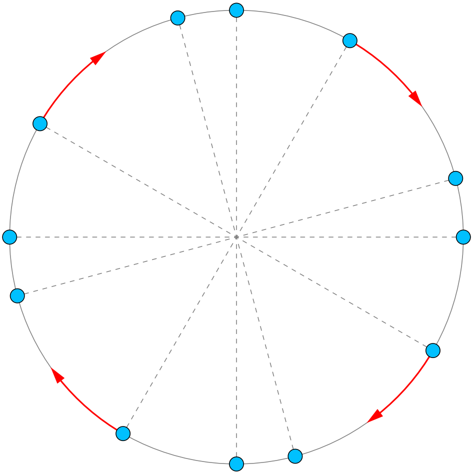
Consider a configuration with a *rotational symmetry*.

Rotationally symmetric configurations



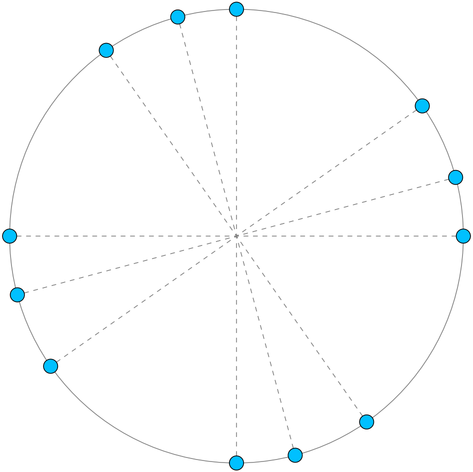
Symmetric robots have *identical views*.

Rotationally symmetric configurations



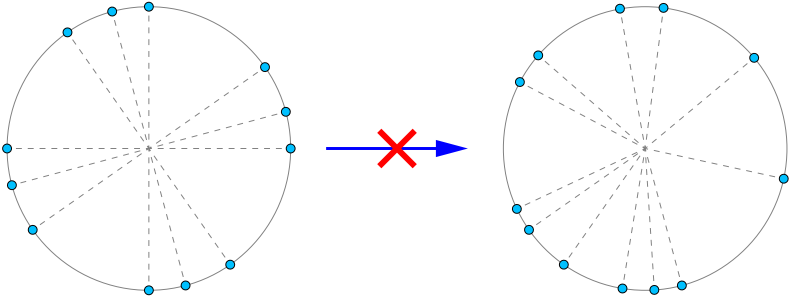
If the scheduler decides to activate all of them at the same time,
they move in *symmetric ways*.

Rotationally symmetric configurations



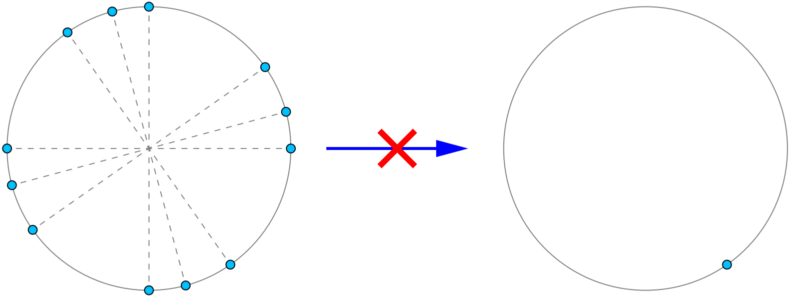
So, the configuration remains *rotationally symmetric*.

Rotationally symmetric configurations



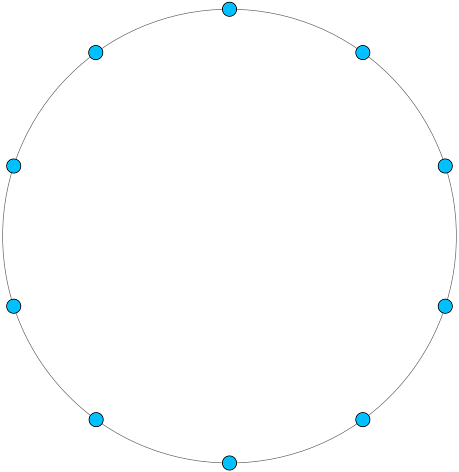
We conclude that, from a *rotationally symmetric* configuration, the robots cannot form an *asymmetric* one.

Rotationally symmetric configurations



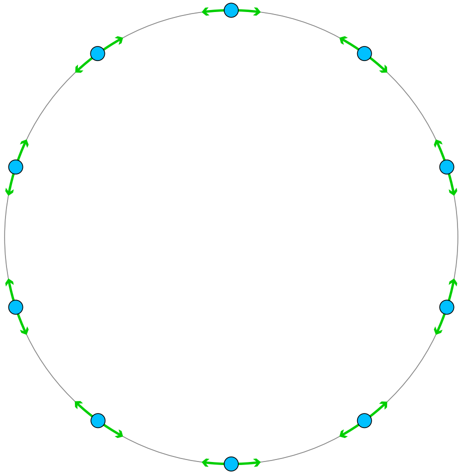
A necessary condition for the gathering problem to be solvable is that the initial configuration be rotationally asymmetric.

Visibility range too short: gathering impossible



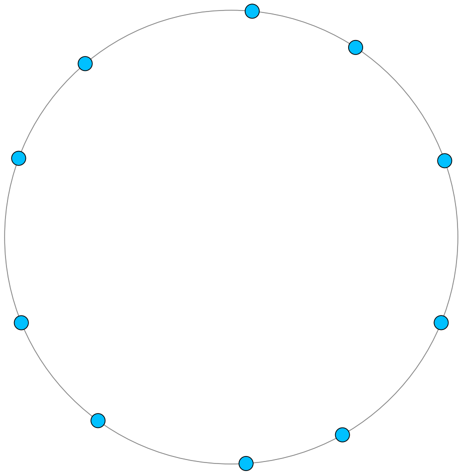
Let the robots be evenly spaced around the circle.

Visibility range too short: gathering impossible



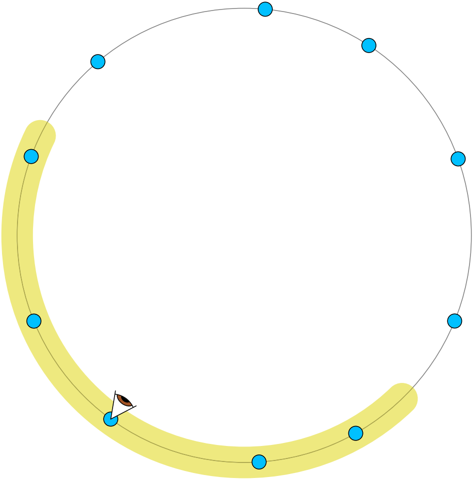
Let us *randomly perturb* them, and let us study their behavior.

Visibility range too short: gathering impossible



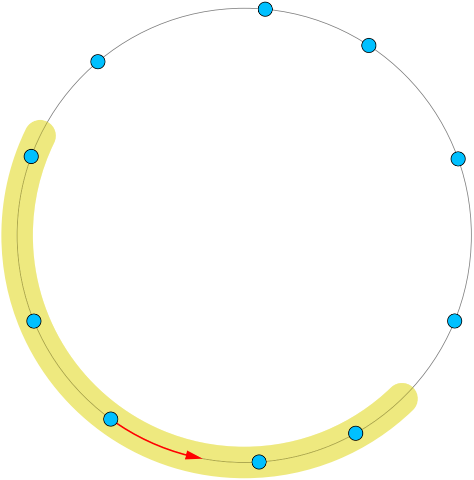
Let us *randomly perturb* them, and let us study their behavior.

Visibility range too short: gathering impossible



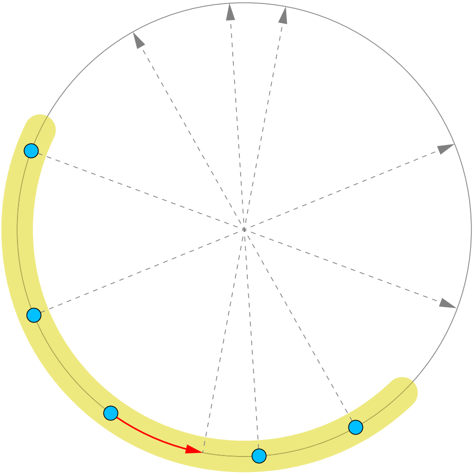
Let us focus on an active robot,
and assume that its visibility range is less than a semicircle.

Visibility range too short: gathering impossible



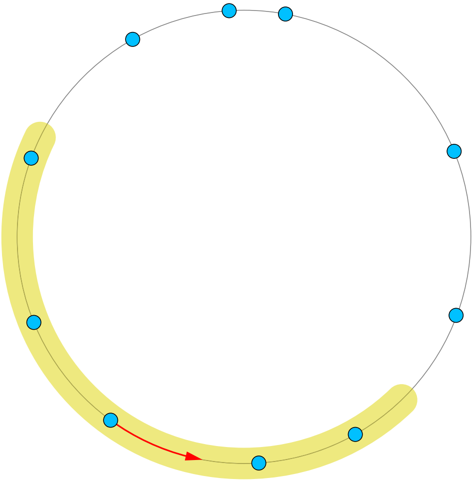
The robot will compute a destination point within its visibility range. Assume this point is currently not occupied by a robot.

Visibility range too short: gathering impossible



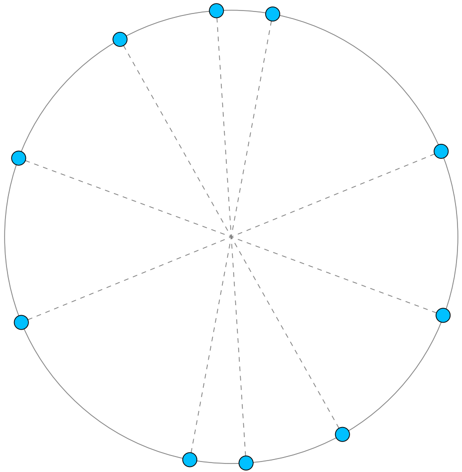
Re-locate the robots on the opposite semicircle as shown.

Visibility range too short: gathering impossible



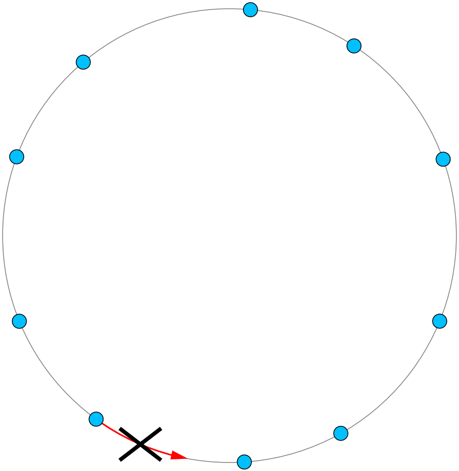
The selected robot will still compute the <u>same destination point</u>, because its visible region has not changed.

Visibility range too short: gathering impossible



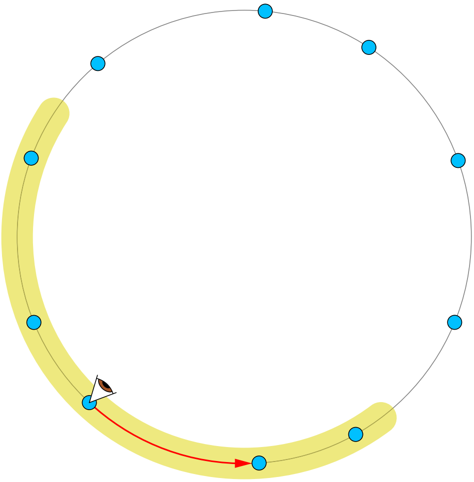
As a result, we have an asymmetric configuration that can evolve into a symmetric one: <u>gathering is impossible</u>.

Visibility range too short: gathering impossible



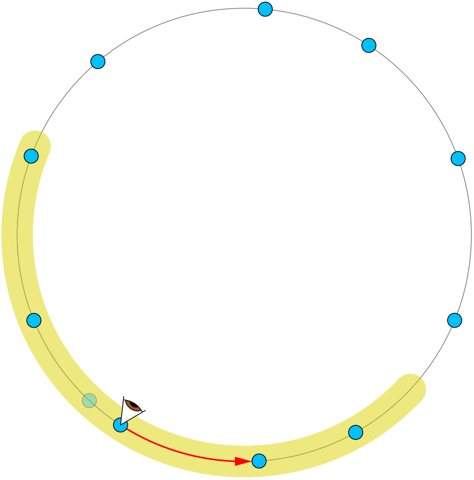
Therefore, a gathering algorithm should not instruct a robot to move to an unoccupied location.

Visibility range too short: gathering impossible



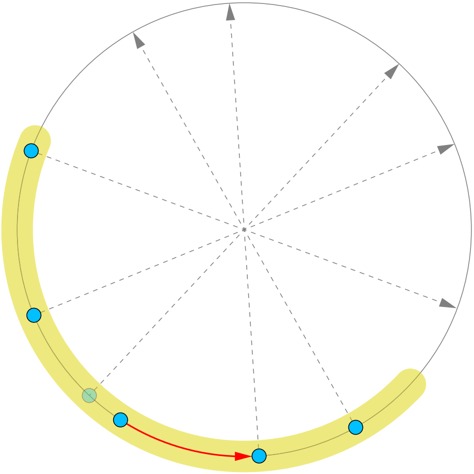
So, let us assume that the robot's destination point is
another robot's current location.

Visibility range too short: gathering impossible



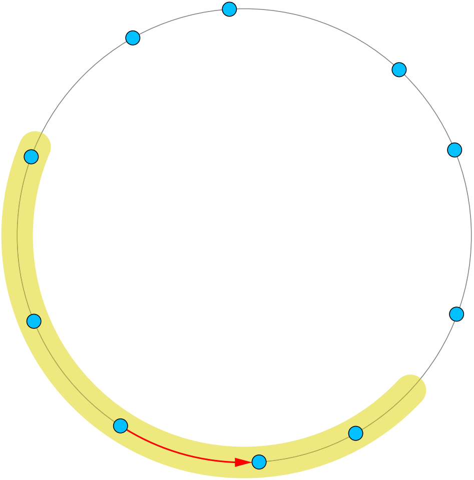
Suppose that also *another perturbation* of the robot causes it to move to the same robot's location.

Visibility range too short: gathering impossible



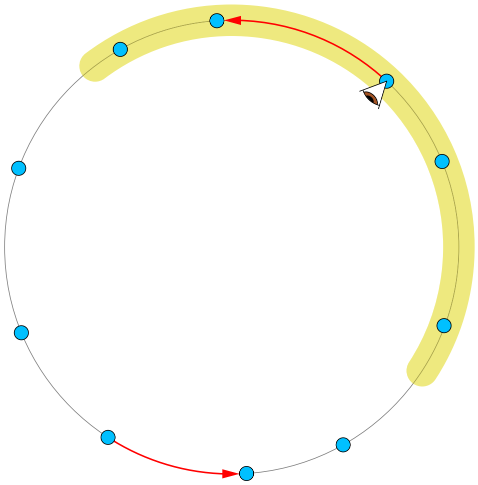
Re-locate the robots on the opposite semicircle as shown.

Visibility range too short: gathering impossible



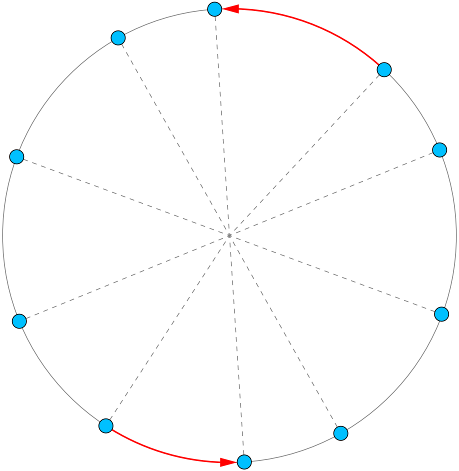
Again, the selected robot will still compute the same destination point, because its visible region has not changed.

Visibility range too short: gathering impossible



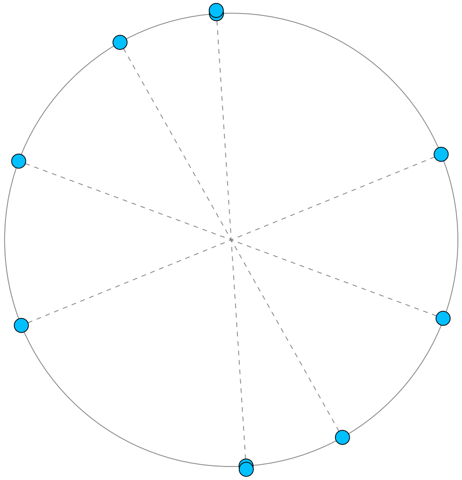
Its copy on the opposite semicircle will move to the corresponding destination point.

Visibility range too short: gathering impossible



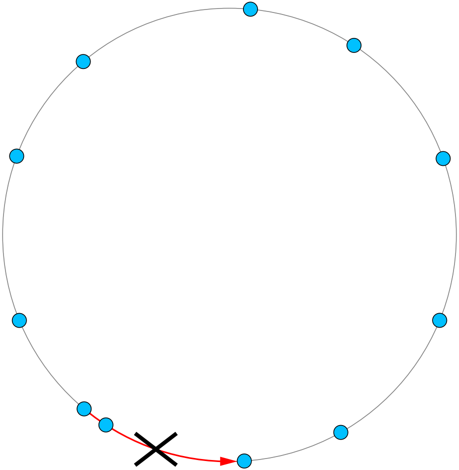
Assume that the scheduler activates *both copies* of the robot.

Visibility range too short: gathering impossible



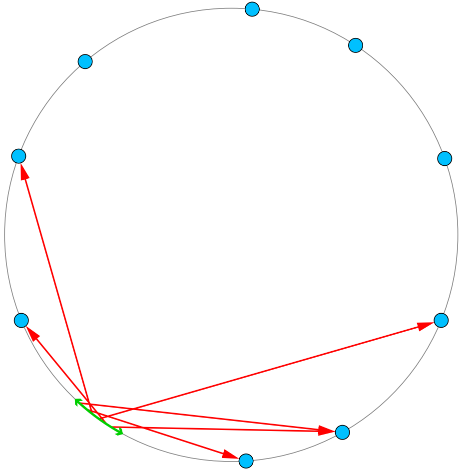
Once again, we have an asymmetric configuration that can evolve into a symmetric one: gathering is impossible.

Visibility range too short: gathering impossible



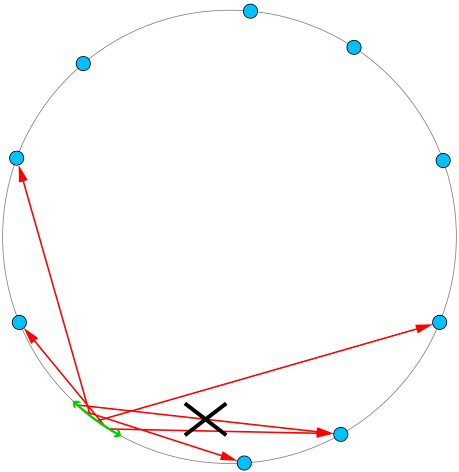
So, there should not be two perturbations of the same robot that cause it to move to the same robot's location.

Visibility range too short: gathering impossible



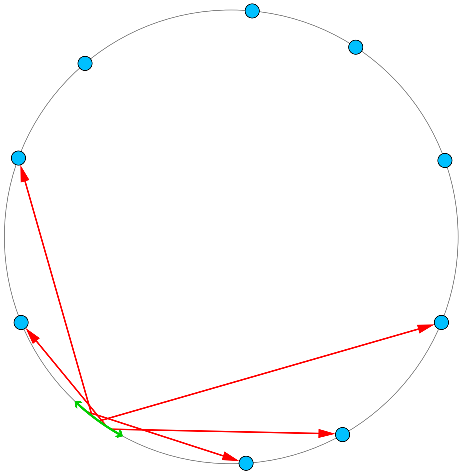
As a consequence, only *finitely many perturbations* of a robot should cause it to move at all.

Visibility range too short: gathering impossible



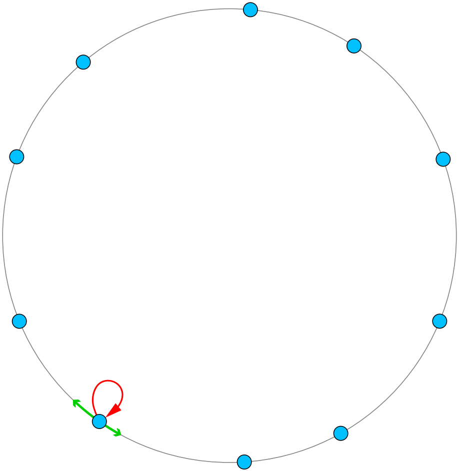
As a consequence, only *finitely many perturbations* of a robot should cause it to move at all.

Visibility range too short: gathering impossible



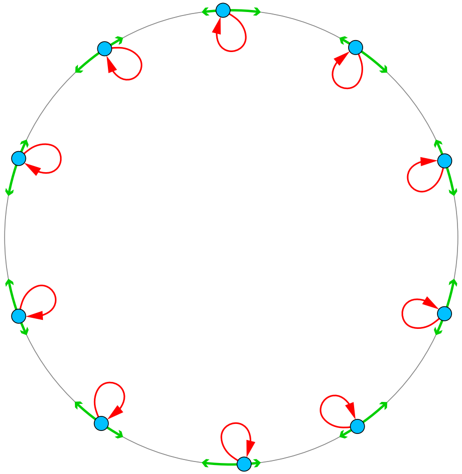
As a consequence, only *finitely many perturbations* of a robot should cause it to move at all.

Visibility range too short: gathering impossible



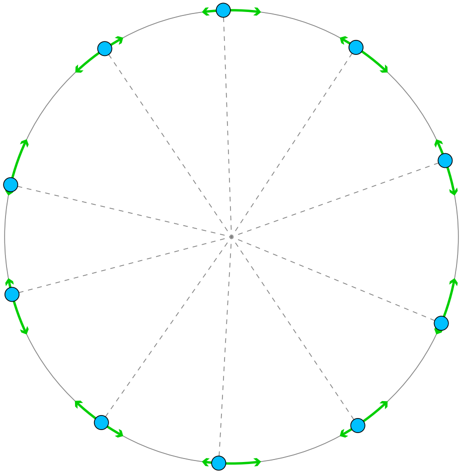
In particular, a random perturbation of a robot should cause it to stay still with probability 1.

Visibility range too short: gathering impossible



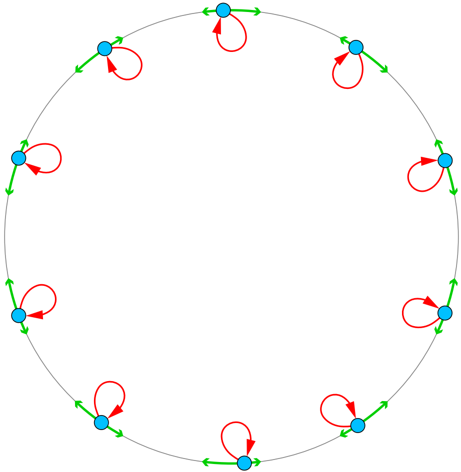
This holds for each robot independently, so it holds for all robots: if randomly perturbed, they will all <u>stay still</u> with probability 1.

Visibility range too short: gathering impossible



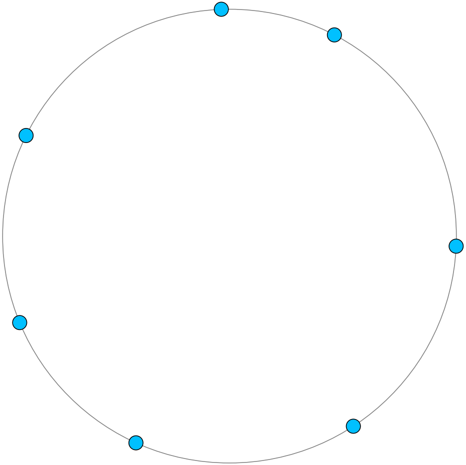
Also, a random perturbation is *asymmetric* with probability 1.

Visibility range too short: gathering impossible



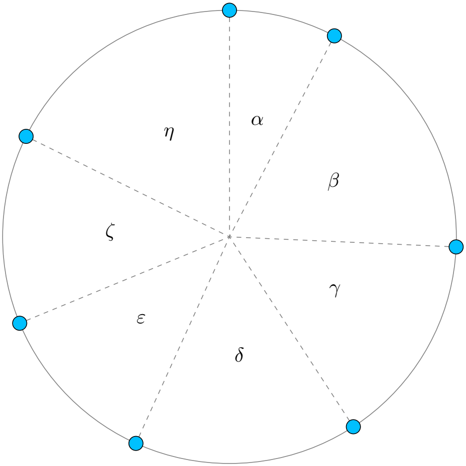
We conclude that there is one asymmetric configuration where no robot moves. In particular, gathering is impossible.

Gathering algorithm for full visibility



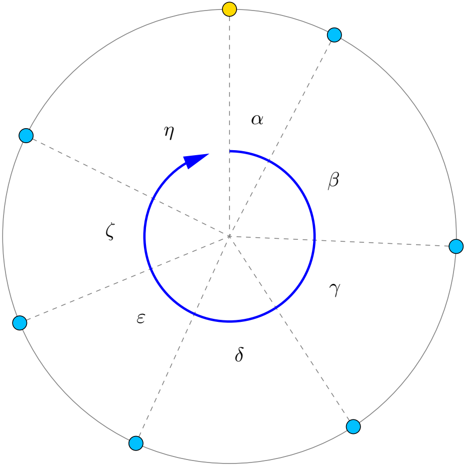
Assume now that all robots have full visibility of the whole circle.

Gathering algorithm for full visibility



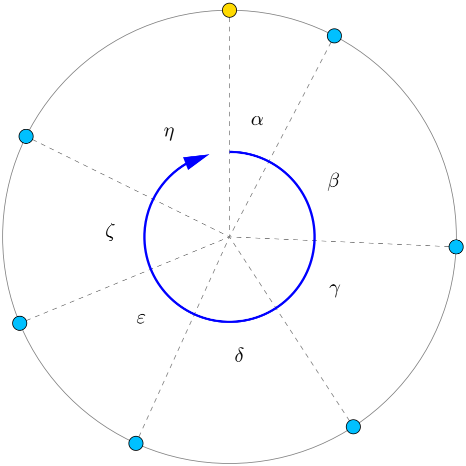
Let the configuration be rotationally asymmetric, and consider the *cyclic sequence of angles* induced by the robots' locations.

Gathering algorithm for full visibility



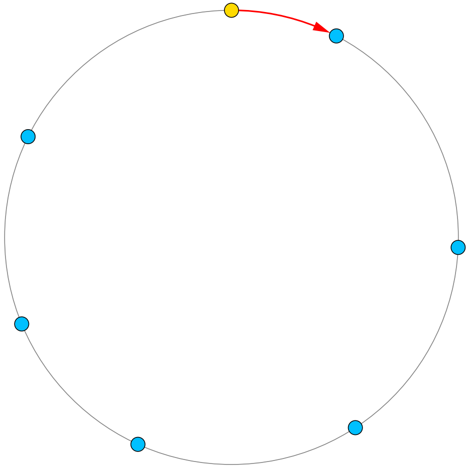
Each robot has an associated angle sequence; the robot with the *lexicographically smallest* angle sequence is the leader.

Gathering algorithm for full visibility



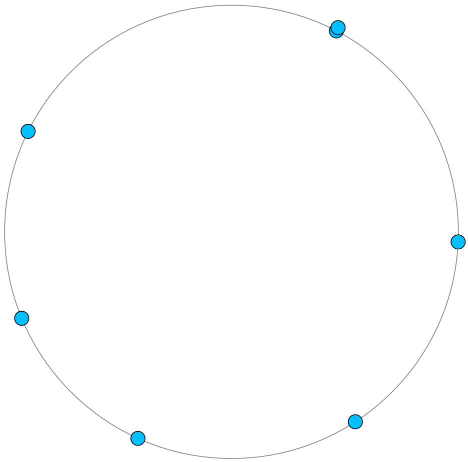
Note that the leader is *unique* because the configuration is asymmetric, and all robots *agree* on the same leader.

Gathering algorithm for full visibility



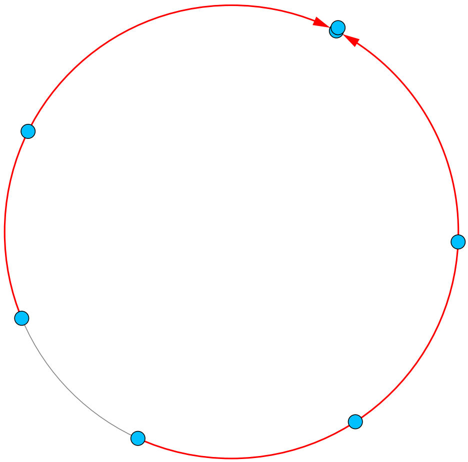
Gathering algorithm: the leader moves clockwise to the next robot's location.

Gathering algorithm for full visibility



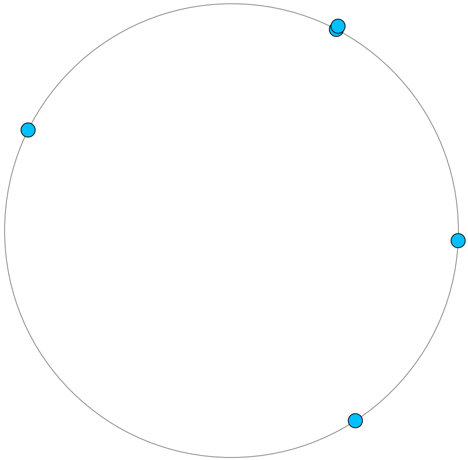
A unique *multiplicity point* is thus formed, i.e., a point where two or more robots are co-located.

Gathering algorithm for full visibility



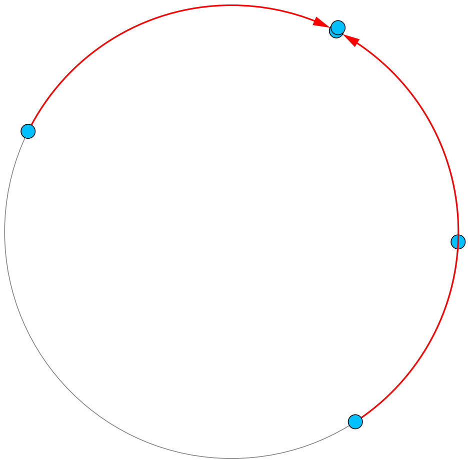
Next, all robots move to the multiplicity point.

Gathering algorithm for full visibility



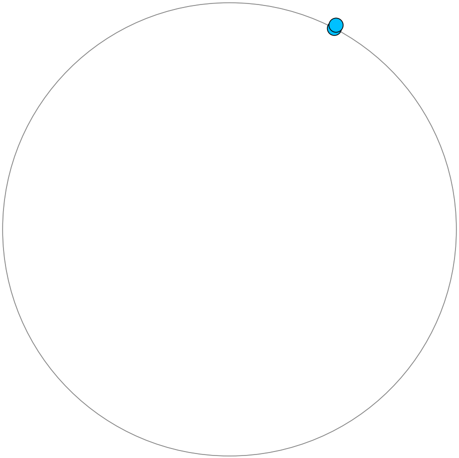
Next, all robots move to the multiplicity point.

Gathering algorithm for full visibility



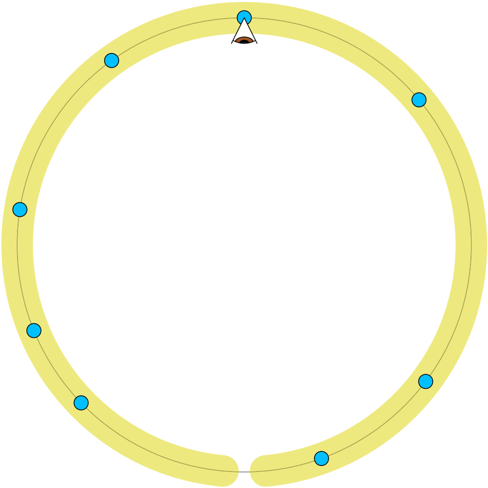
Next, all robots move to the multiplicity point.

Gathering algorithm for full visibility



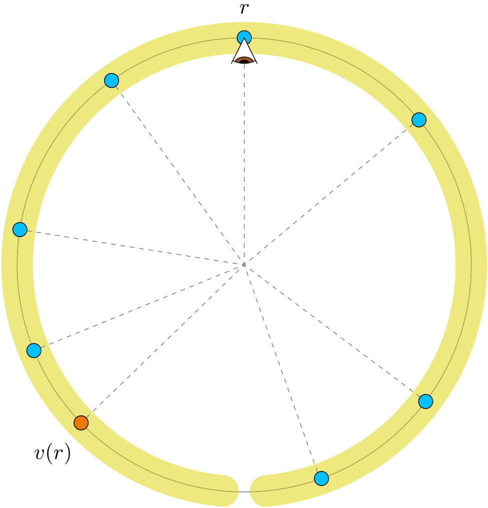
Can we adapt this strategy to robots with *limited visibility*?

Gathering algorithm for almost full visibility



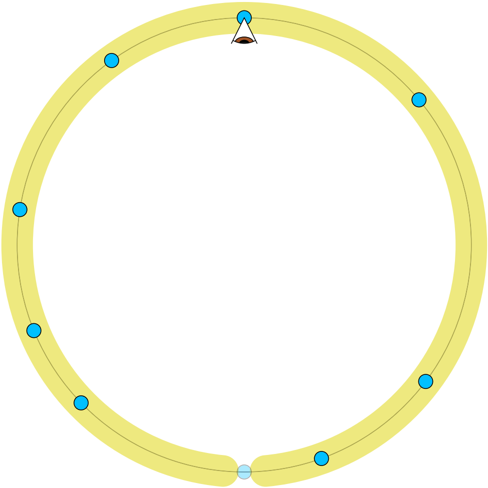
Almost full visibility: each robot sees the whole circle except its *antipodal point*.

Gathering algorithm for almost full visibility



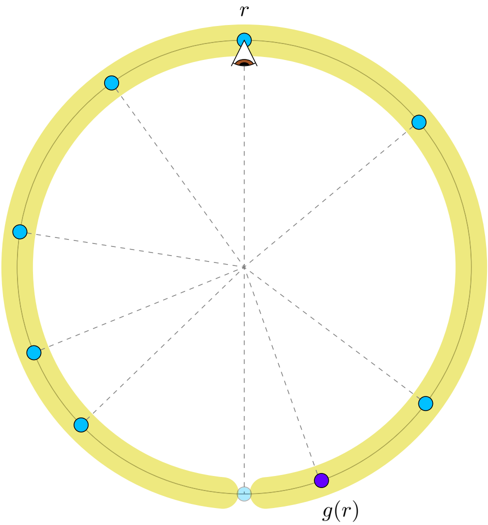
From the point of view of a robot r, two scenarios are possible:
the antipodal point is *not occupied*, and $v(r)$ is the visible leader...

Gathering algorithm for almost full visibility



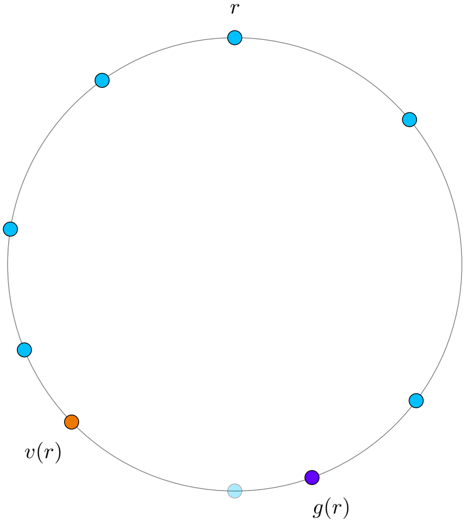
...Or the antipodal point is *occupied by a robot*,
and in this case the leader $g(r)$ is called the ghost leader.

Gathering algorithm for almost full visibility



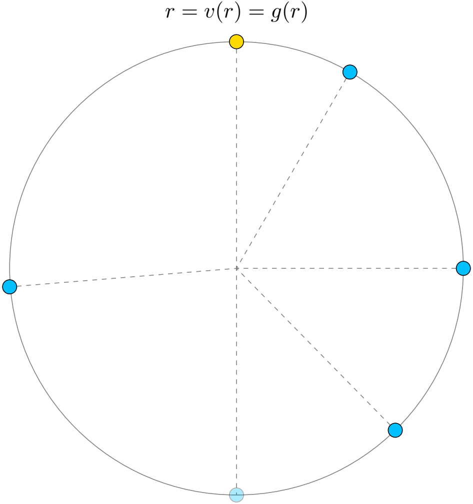
...Or the antipodal point is *occupied by a robot*,
and in this case the leader $g(r)$ is called the ghost leader.

Gathering algorithm for almost full visibility



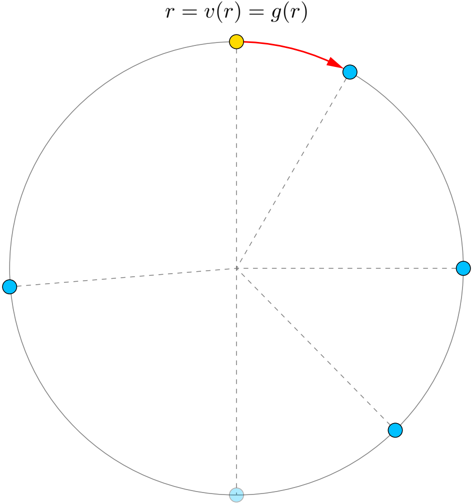
Note that either $g(r)$ or $v(r)$ is the "true leader", depending on whether the point opposite to r is occupied or not.

Gathering algorithm for almost full visibility



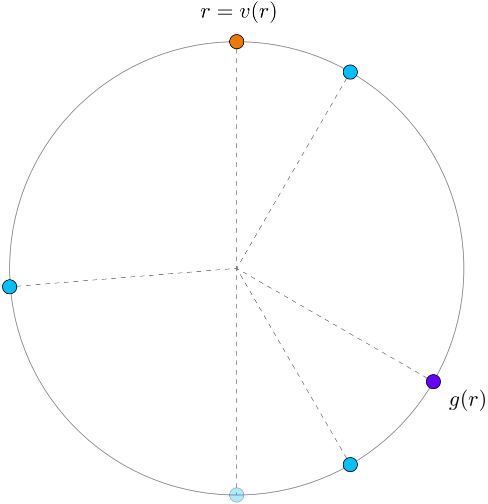
If $r = v(r) = g(r)$, then r is a cognizant leader:
r is certainly the true leader, and it is *aware* of it.

Gathering algorithm for almost full visibility



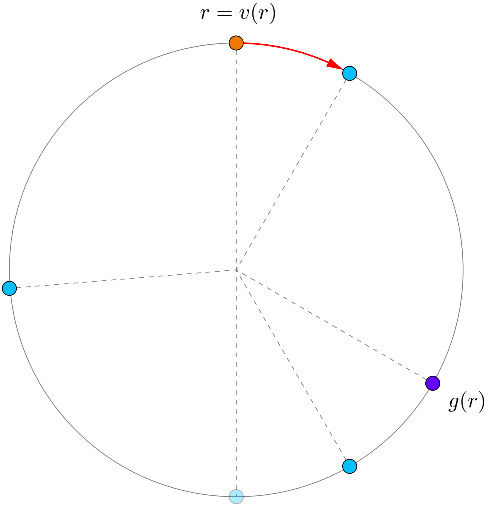
In this case, r acts like in the full-visibility setting:
it moves to the next robot clockwise, forming a *multiplicity point*.

Gathering algorithm for almost full visibility



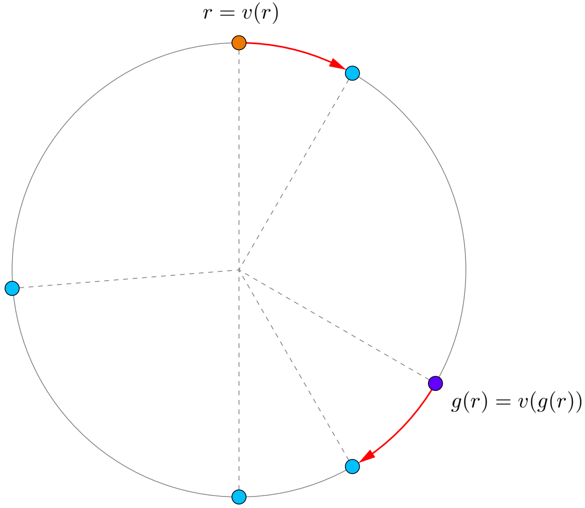
If $r = v(r) \neq g(r)$, then r is an undecided leader:
r sees itself as the leader, but it knows it *may be wrong*.

Gathering algorithm for almost full visibility



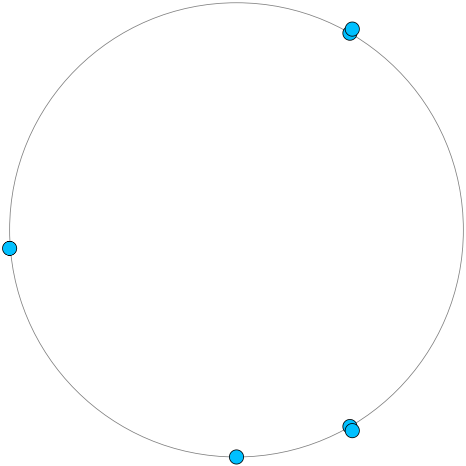
What if an *undecided leader* moves to the next robot, as well?

Gathering algorithm for almost full visibility



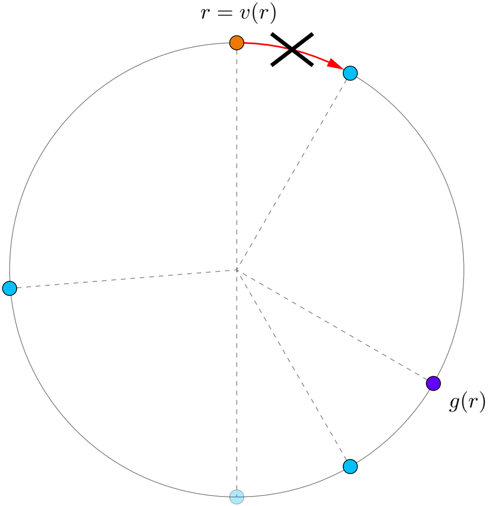
There may be *more than one* undecided leader in a configuration.
If both are activated, two distinct multiplicity points are created.

Gathering algorithm for almost full visibility



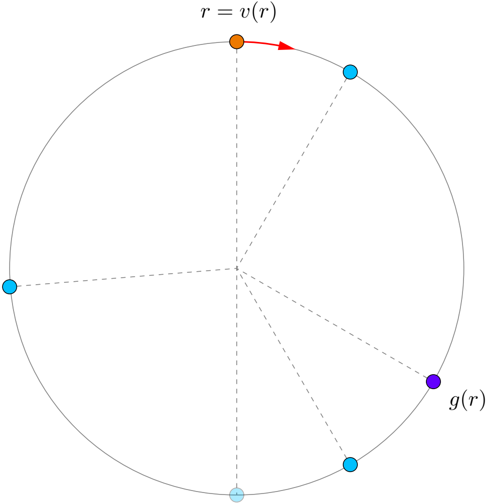
There may be *more than one* undecided leader in a configuration.
If both are activated, two distinct multiplicity points are created.

Gathering algorithm for almost full visibility



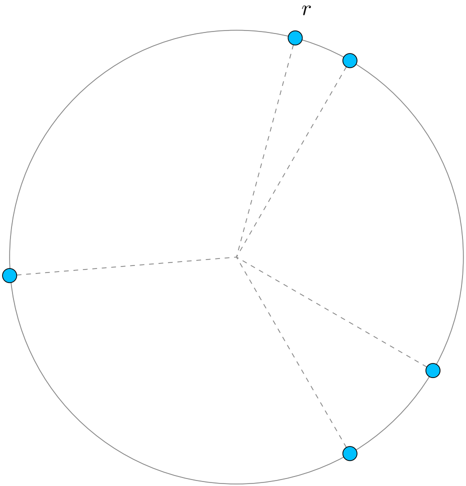
As we would like to have at most one multiplicity point,
we should not let an undecided leader move to the next robot.

Gathering algorithm for almost full visibility



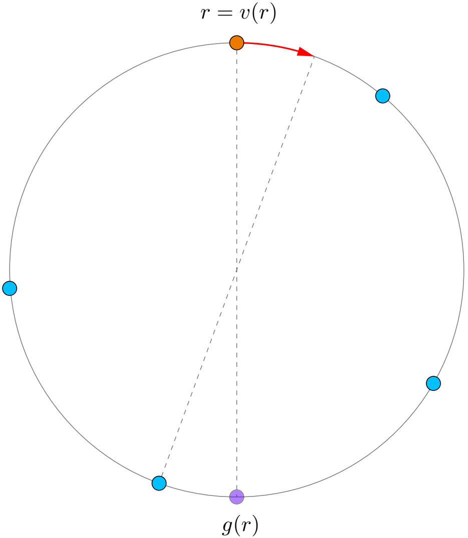
Instead, an undecided leader will attempt to "strengthen its leadership" by moving halfway toward the next robot clockwise.

Gathering algorithm for almost full visibility



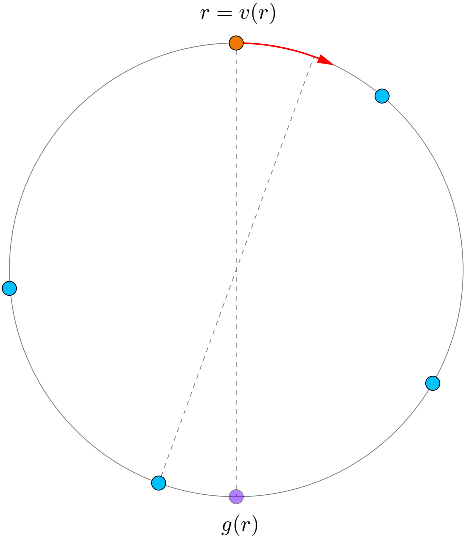
After that, it will have a smaller angle sequence, and it will be "more likely" to be the true leader.

Gathering algorithm for almost full visibility



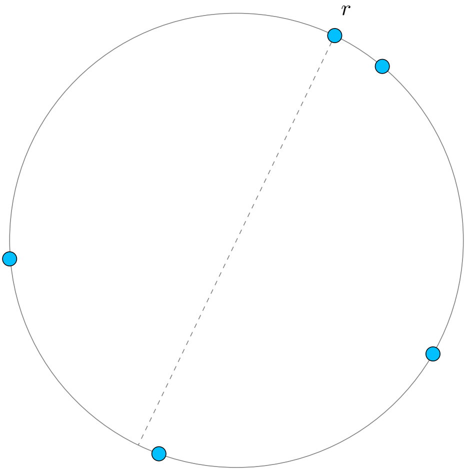
We also want to prevent robots from having *antipodal robots*, in order to promote *mutual visibility*.

Gathering algorithm for almost full visibility



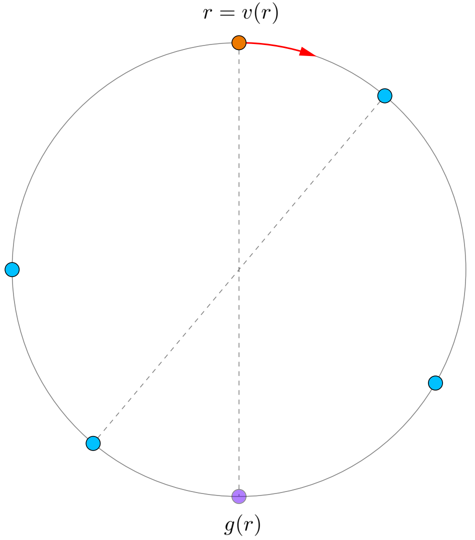
So, if the halfway point is antipodal to some robot, an undecided leader will move *slightly further*.

Gathering algorithm for almost full visibility

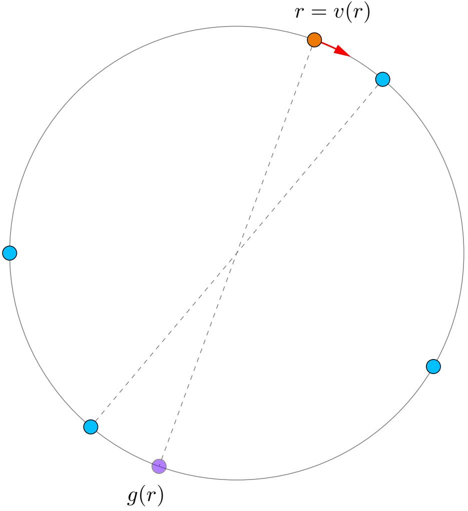


So, if the halfway point is antipodal to some robot, an undecided leader will move *slightly further*.

Gathering algorithm for almost full visibility

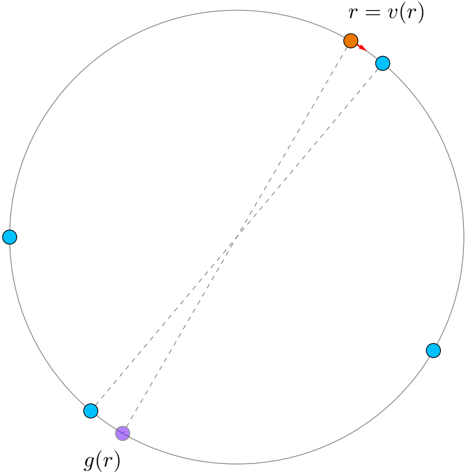


There is one more special case to consider: what if the robot next to an undecided leader has an antipodal robot?



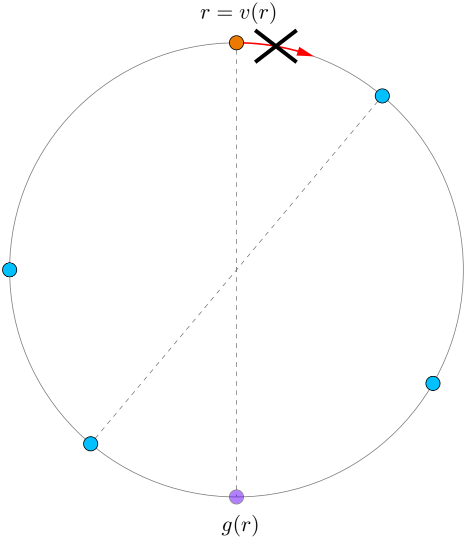
In this case, the undecided leader will *keep approaching* the next robot indefinitely.

Gathering algorithm for almost full visibility



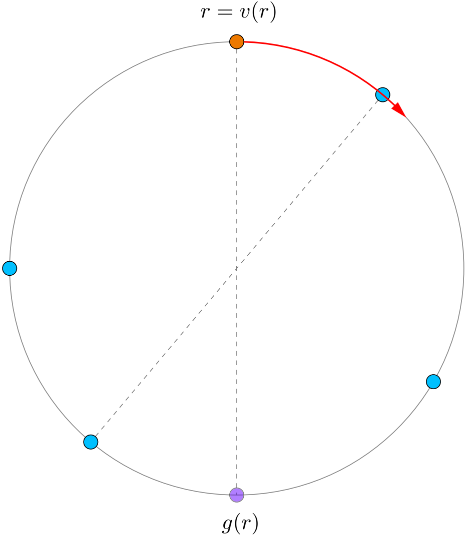
In this case, the undecided leader will *keep approaching* the next robot indefinitely.

Gathering algorithm for almost full visibility



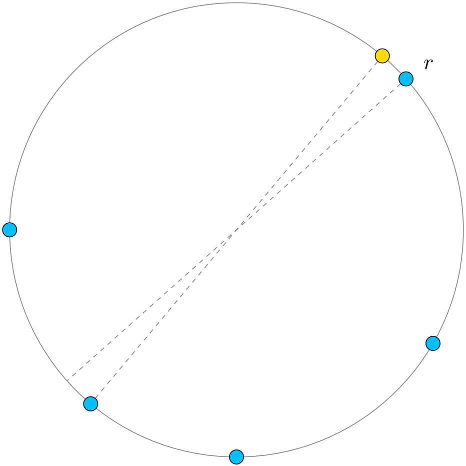
So, in this special case, approaching the next robot is a *bad idea*.

Gathering algorithm for almost full visibility



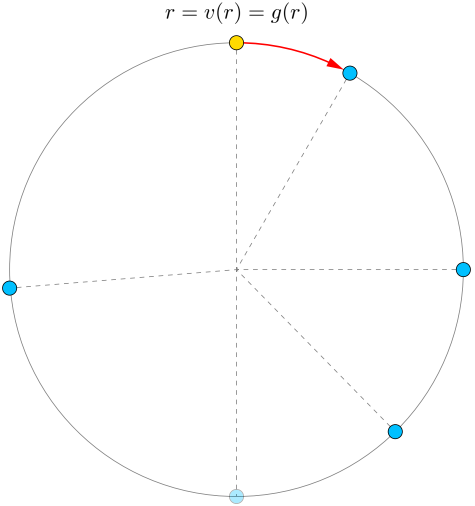
The correct move is to go *slightly past* the next robot.

Gathering algorithm for almost full visibility



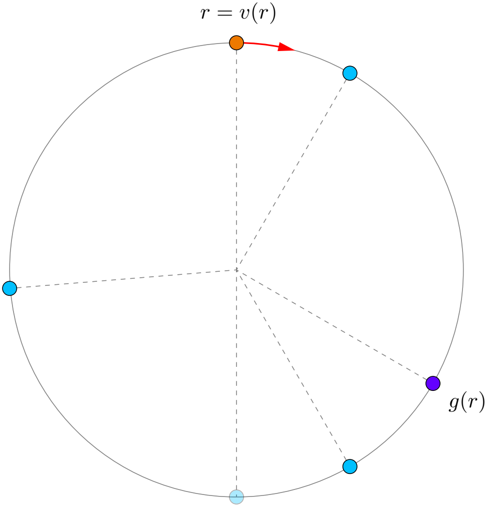
This will “unlock” the configuration, and is likely to create a *cognizant leader*.

Gathering algorithm: summary



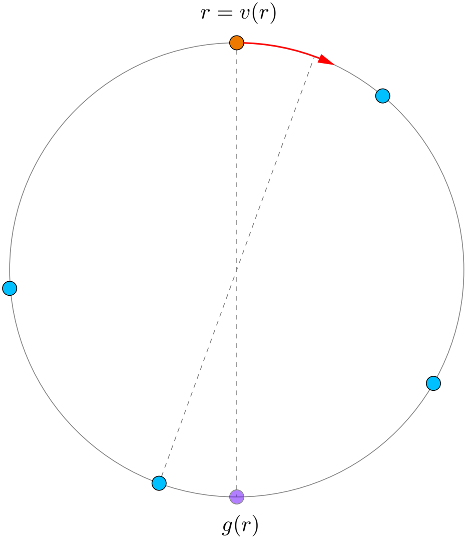
Rule 1: a *cognizant leader* moves to the next robot.

Gathering algorithm: summary



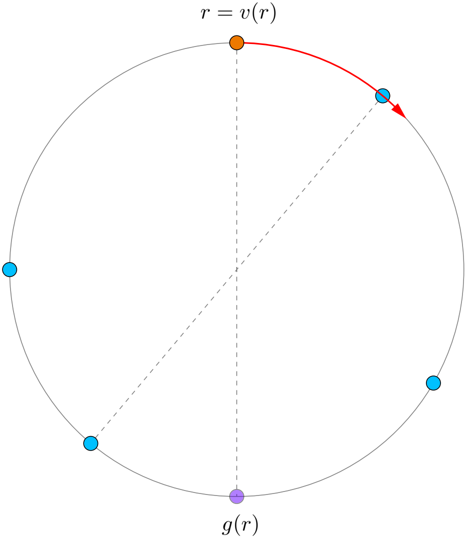
Rule 2: an *undecided leader* moves halfway to the next robot.

Gathering algorithm: summary



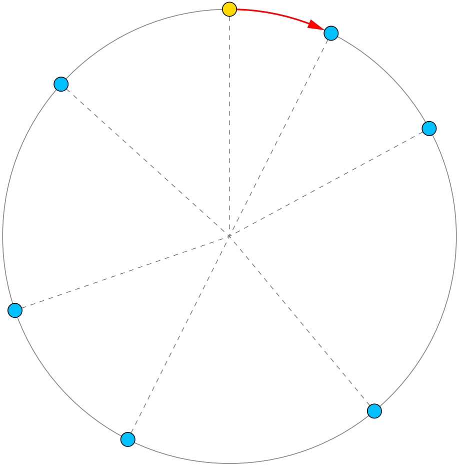
Rule 3: if the midpoint has an antipodal robot,
an *undecided leader* moves slightly past it.

Gathering algorithm: summary



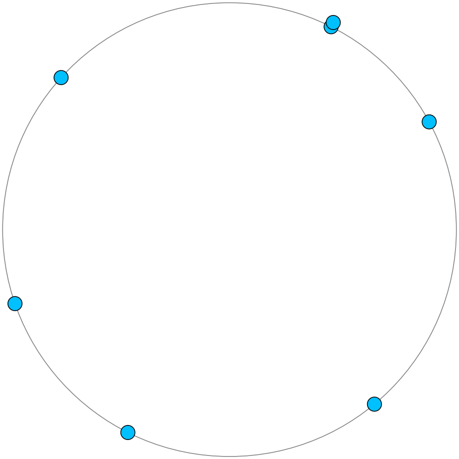
Rule 4: if the next robot has an antipodal robot, an *undecided leader* moves slightly past it.

Correctness: Rule 1



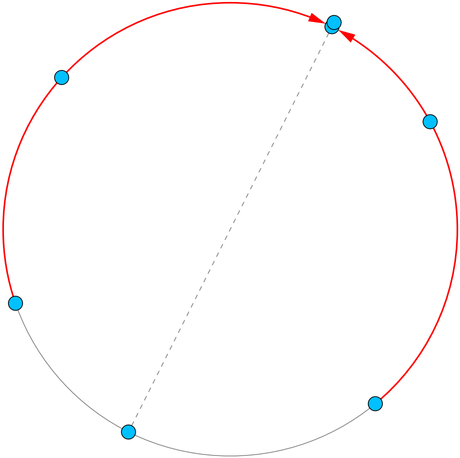
Suppose that a cognizant leader executes Rule 1.

Correctness: Rule 1



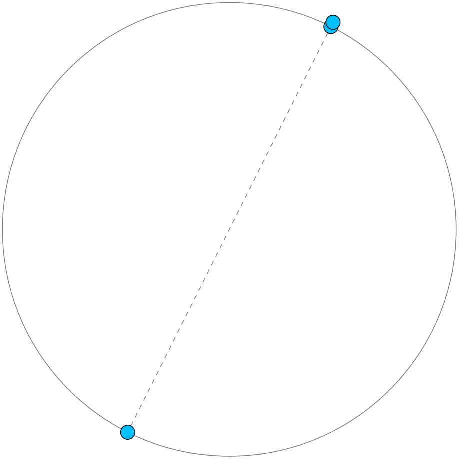
There can be at most one cognizant leader,
so a unique multiplicity point is formed.

Correctness: Rule 1



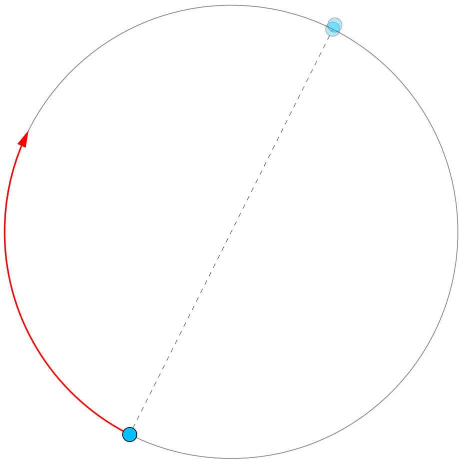
Now, all robots that see the multiplicity point move to it.

Correctness: Rule 1



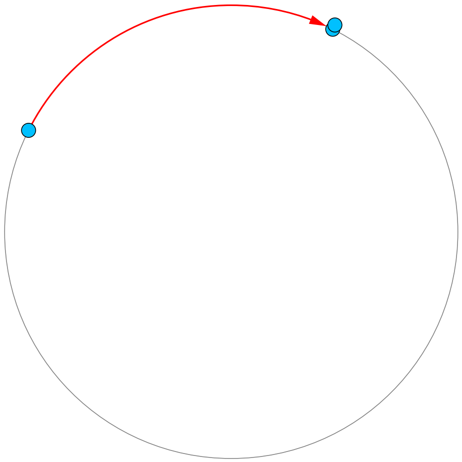
At most one robot will not join the multiplicity point:
its *antipodal robot*.

Correctness: Rule 1



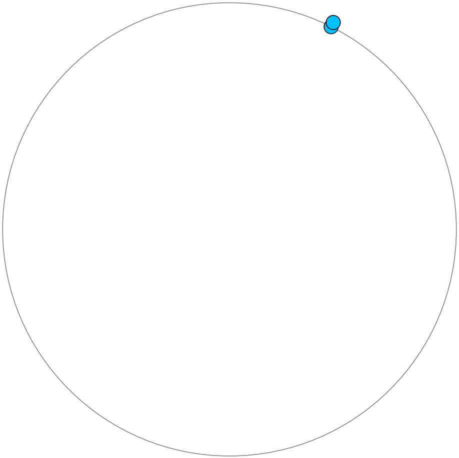
This robot will eventually see no other robot.
When this happens, it makes a move in *any direction*.

Correctness: Rule 1



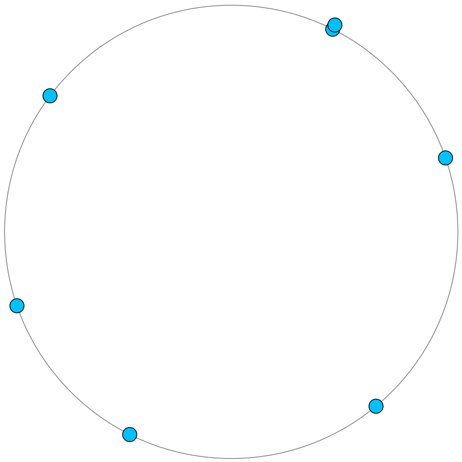
From there, the robot will be able to see the multiplicity point, and it will finally join it.

Correctness: Rule 1



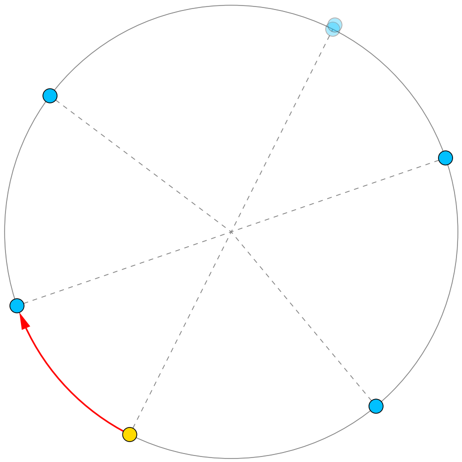
From there, the robot will be able to see the multiplicity point, and it will finally join it.

Correctness: Rule 1



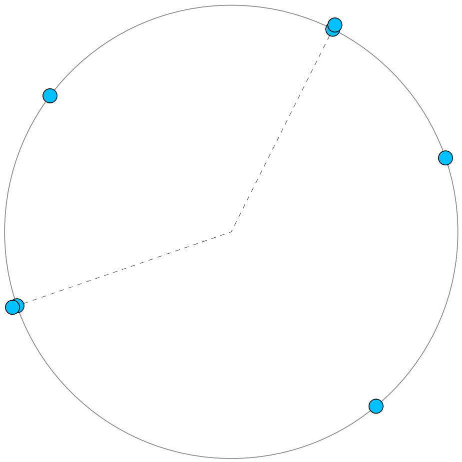
There is one special case to consider: the robot antipodal to the multiplicity point may become a *cognizant leader*.

Correctness: Rule 1



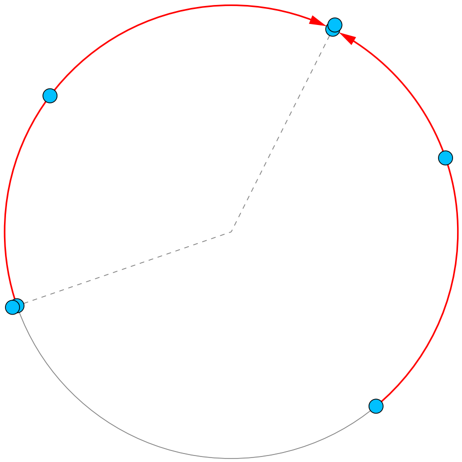
There is one special case to consider: the robot antipodal to the multiplicity point may become a *cognizant leader*.

Correctness: Rule 1



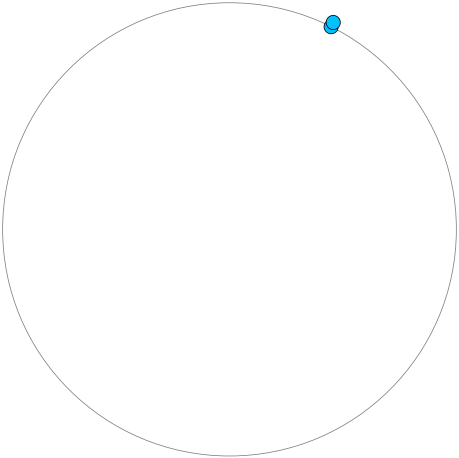
This may originate a second multiplicity point.
However, the two multiplicity points are not antipodes.

Correctness: Rule 1



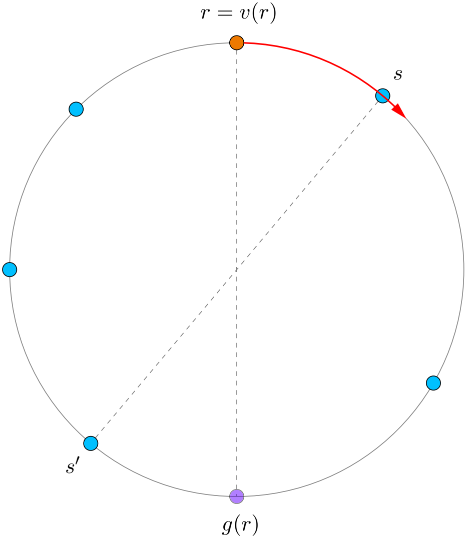
So, the two multiplicity points can be *distinguished*,
and all robots can deterministically join the same one.

Correctness: Rule 1



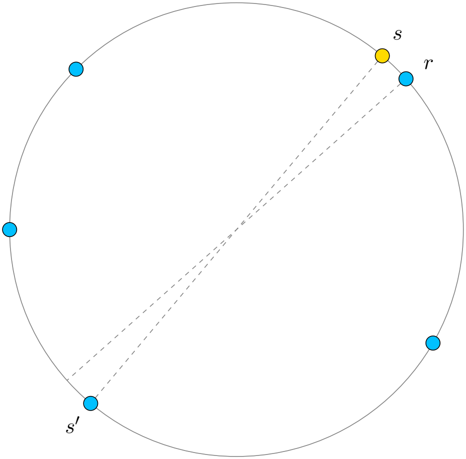
Thus, if a robot ever executes Rule 1,
all robots eventually gather.

Correctness: Rule 4



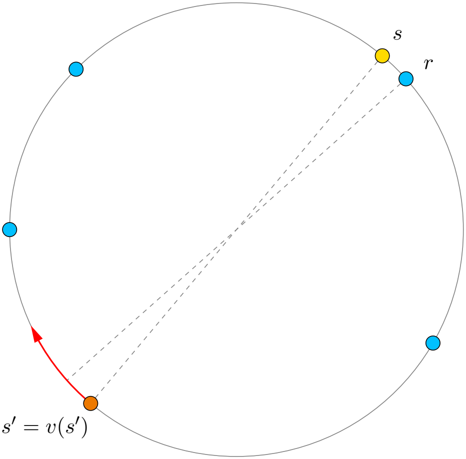
Suppose that an undecided leader r executes Rule 4, moving slightly past the next robot s, which has an antipodal robot s'.

Correctness: Rule 4



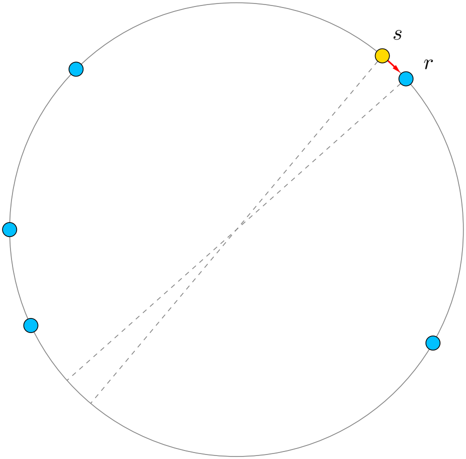
Now the distance between s and r is *minimum*, and all robots except s' can see both s and r.

Correctness: Rule 4



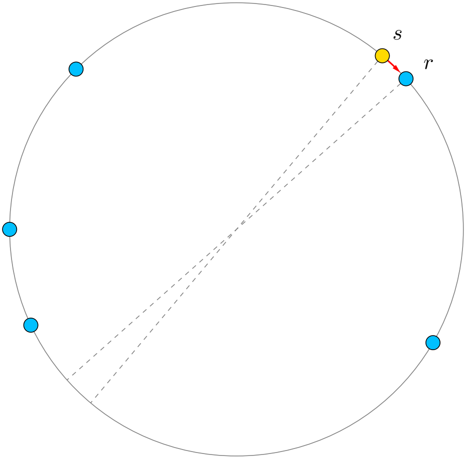
So, no robot other than s' can be an *undecided leader*. In this case, s' will move to a location where it can see both s and r.

Correctness: Rule 4



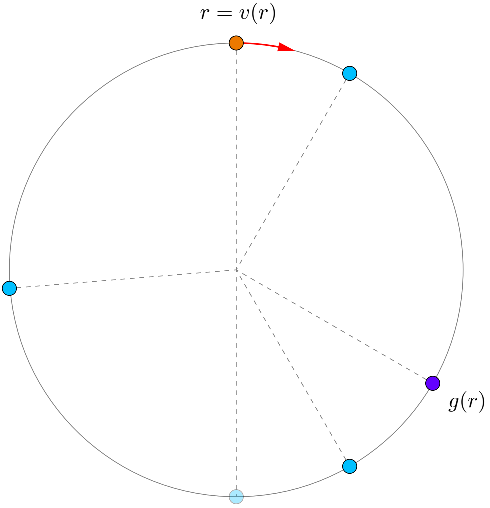
After this, all robots will wait until s,
which is a *cognizant leader*, executes Rule 1.

Correctness: Rule 4



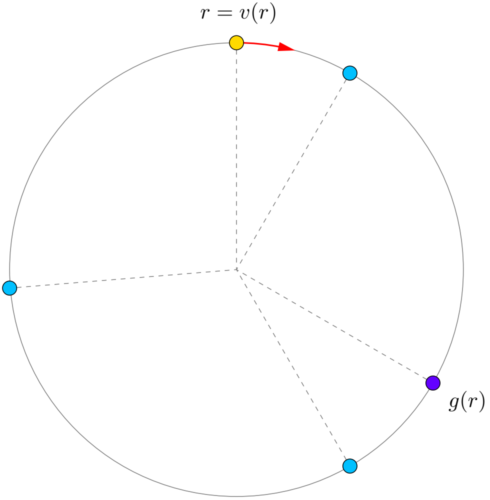
Thus, if a robot ever executes Rule 4,
all robots eventually gather.

Correctness: Rules 2 and 3



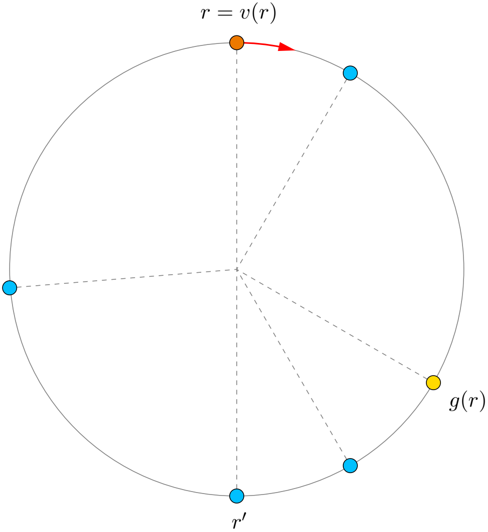
Assume now that all moving robots only ever execute
Rule 2 or Rule 3.

Correctness: Rules 2 and 3



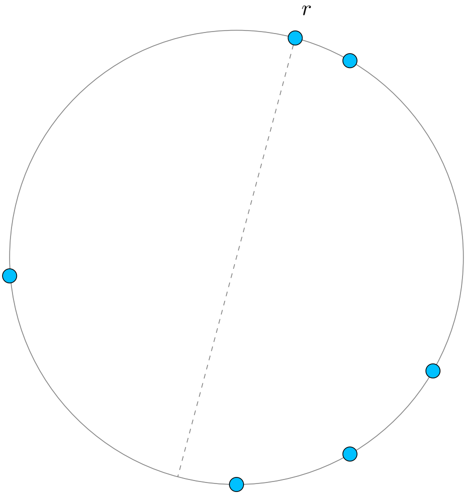
Claim: a robot that executes Rule 2 or 3 is either the *true leader* or it has an *antipodal robot*.

Correctness: Rules 2 and 3



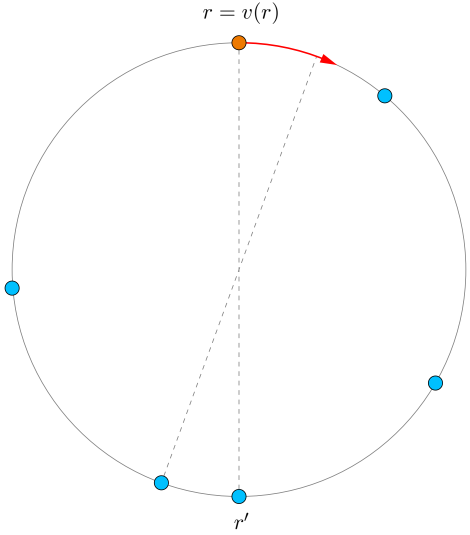
Indeed, the true leader is either $r = v(r)$ or $g(r)$.
In the latter case, r must have an antipodal robot.

Correctness: Rules 2 and 3



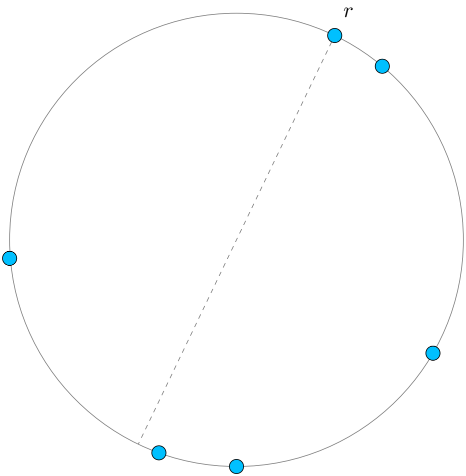
Moreover, Rules 2 and 3 ensure that, after r has moved, it will *never* have an antipodal robot again.

Correctness: Rules 2 and 3



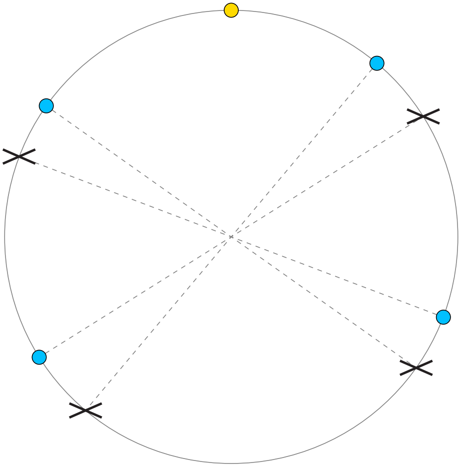
Moreover, Rules 2 and 3 ensure that, after r has moved, it will *never* have an antipodal robot again.

Correctness: Rules 2 and 3



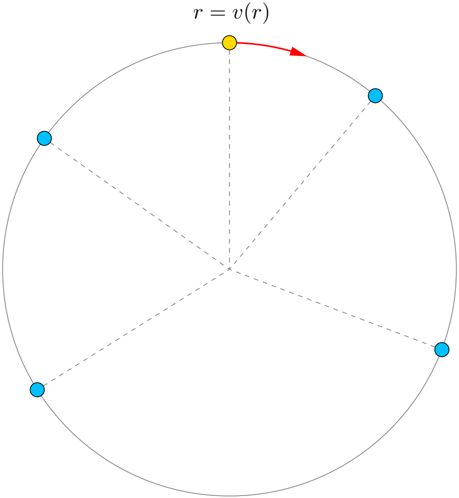
Moreover, Rules 2 and 3 ensure that, after r has moved, it will *never* have an antipodal robot again.

Correctness: Rules 2 and 3



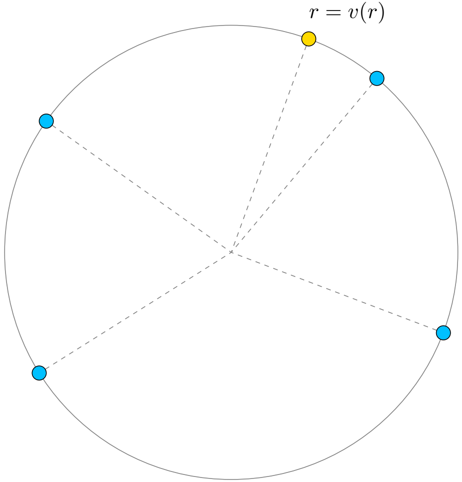
But any robot other than the true leader moves only if it has an *antipodal robot*, and so it moves *at most once*.

Correctness: Rules 2 and 3



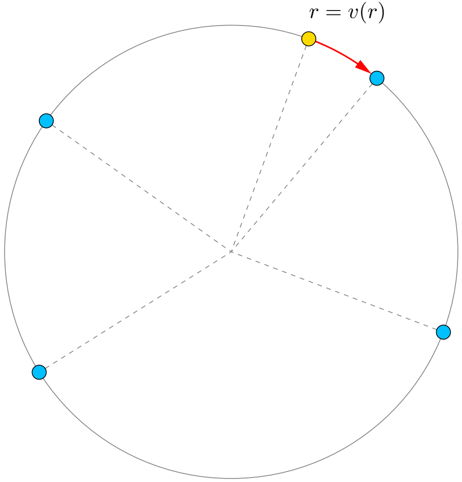
Thus, eventually, only the true leader will move.

Correctness: Rules 2 and 3



After the true leader r has executed Rule 2 or 3,
it is still the true leader.

Correctness: Rules 2 and 3



Eventually, r becomes a *cognizant leader*, and executes Rule 1.
We conclude that the robots gather in every case.

Conclusion

Results:

- If each robot can see *less than a semicircle*, the gathering problem is <u>unsolvable</u>
  - This is true even if the *total number of robots* is known
  - The result extends to all *pattern-formation* problems where the pattern is *not centrally symmetric*
- If each robot sees the whole circle except its *antipodal point*, there is a <u>gathering algorithm</u>

Open problems:

- Find the *smallest visibility range* such that the gathering problem is *solvable*
- What if robots are *asynchronous*?
- What if robots can *fail* to reach their destination point?
- What if robots disagree on the *clockwise direction*?