Ring Exploration with Myopic Luminous Robots

Fukuhito Ooshita
Nara Institute of Science and Technology,
Graduate School of Science and Technology,
Takayama 8916-5, Ikoma, Nara, Japan
f-oosita@is.naist.jp

Sébastien Tixeuil
Sorbonne Université, CNRS,
Laboratoire d’Informatique de Paris 6, LIP6,
FR-75005, Paris, France
Sebastien.Tixeuil@lip6.fr

May 11, 2018

Abstract

We investigate exploration algorithms for autonomous mobile robots evolving in uniform ring-shaped networks. Different from the usual Look-Compute-Move (LCM) model, we consider two characteristics: myopia and luminosity. Myopia means each robot has a limited visibility. We consider the weakest assumption for myopia: each robot can only observe its neighboring nodes. Luminosity means each robot maintains a non-volatile visible light. We consider the weakest assumption for luminosity: each robot can use only two colors for its light. The main interest of this paper is to clarify the impact of luminosity on exploration with myopic robots.

As a main contribution, we prove that 1) in the fully synchronous model, two and three robots are necessary and sufficient to achieve perpetual and terminating exploration, respectively, and 2) in the semi-synchronous and asynchronous models, three and four robots are necessary and sufficient to achieve perpetual and terminating exploration, respectively. These results clarify the power of lights for myopic robots since, without lights, five robots are necessary and sufficient to achieve terminating exploration in the fully synchronous model, and no terminating exploration algorithm exists in the semi-synchronous and asynchronous models.

We also show that, in the fully synchronous model (resp., the semi-synchronous and asynchronous models), the proposed perpetual exploration algorithm is universal, that is, the algorithm solves perpetual exploration from any solvable initial configuration with two (resp., three) robots and two colors. On the other hand, we show that, in the fully synchronous model (resp., the semi-synchronous and asynchronous models), no universal algorithm exists for terminating exploration, that is, no algorithm may solve terminating exploration from any solvable initial configuration with three (resp., four) robots and two colors.

Keywords: autonomous mobile robots, deterministic exploration, discrete environments, limited visibility, visible light
1 Introduction

1.1 Background and Motivation

Studies about cooperation of autonomous mobile robots have attracted a lot of attention recently in the field of Distributed Computing. The main goal of those works is to characterize the minimum capabilities of robots that permit to achieve a given task. Since the pioneering work of Suzuki and Yamashita [24], many results have been published in their Look-Compute-Move (LCM) model. In the LCM model, each robot repeats executing cycles of look, compute, and move phases. At the beginning of each cycle, the robot observes positions of other robots (look phase). According to its observation, the robot computes whether it moves somewhere or stays idle (compute phase). If the robot decides to move, it moves to the target position by the end of the cycle (move phase). To consider minimum capabilities, most studies assume that robots are identical (i.e., robots execute the same algorithm and cannot be distinguished), oblivious (i.e., robots have no memory of their past actions), and silent (i.e., robots cannot communicate with other robots explicitly). Indeed, communication among robots is done only in an implicit way by observing positions of other robots and moving to a new position. Previous works considered problem solvability of LCM robots in continuous environments (aka two- or three-dimensional Euclidean space) [17, 18, 24, 25], while others considered discrete environments (aka graph networks) [4, 6, 16, 22, 23].

In this paper, we focus on robots evolving in graph networks. The most fundamental tasks in graph networks are gathering and exploration. The goal of gathering is to make all robots meet at a non-predetermined single node. Gathering has been studied for rings [1, 6, 22, 23], grids and trees [5]. Two types of exploration tasks have been well studied: perpetual exploration requires robots to visit nodes so that every node is visited infinitely many times by a robot, and terminating exploration requires robots to terminate after every node is visited by a robot at least once. For example, perpetual exploration has been studied for rings [1] and grids [2], and terminating exploration has been studied for rings [14, 16], trees [15], grids [12], tori [13], and arbitrary networks [3].

All aforementioned works in graph networks make the assumption that each robot observes all other robots in the networks. That is, each robot has a sensor that can obtain a global snapshot. However, this powerful ability somewhat contradicts the principle of very weak mobile entities. For this reason, recent studies consider the more realistic case of myopic robots [8, 10, 19, 20]. A myopic robot has limited visibility, i.e., it can see nodes (and robots on them) only within a certain fixed distance $\phi$. Datta et al. studied terminating exploration of rings for $\phi = 1$ [8] and $\phi = 2, 3$ [9]. Guibault and Pelc studied gathering in bipartite graphs with $\phi = 1$ [19], and infinite lines with $\phi > 1$ [20]. Not surprisingly, in the weakest setting, i.e., $\phi = 1$, robots can only achieve few tasks. It is shown [8] that, when $\phi = 1$ holds, five robots are necessary and sufficient to achieve terminating exploration in the fully synchronous (FSYNC) model. On the other hand, no terminating exploration algorithm exists in the semi-synchronous (SSYNC) and asynchronous (ASYNC) models. Also, gathering [19] is possible when $\phi = 1$ only if robots initially form a star.

Since most results for myopic robots with $\phi = 1$ are negative, a natural question is which additional assumptions can improve task solvability. In this paper, we focus on a non-volatile visible light [7] as an additional assumption. A robot endowed with such a light is called a luminous robot. Each luminous robot is equipped with a light device that can emit a constant number of colors to other robots, a single color at a time. The light color is non-volatile, so it can be used as a constant-space memory. For non-myopic luminous robots, the power of lights is well understood [7, 11, 21].
Table 1: Ring exploration with myopic robots.

| Reference | Exploration | Synchrony          | $\phi$ | #colors | necessary | sufficient |
|-----------|-------------|--------------------|--------|---------|-----------|------------|
| [8]       | terminating | FSYNC              | 1      | 1       | 5         | 5          |
| [8]       | terminating | SSYNC & ASYNC      | 1      | 1       | impossible|            |
| [10]      | terminating | SSYNC & ASYNC      | 2      | 1       | 5         | 7          |
| This paper| perpetual   | FSYNC              | 1      | 2       | 2         | 2          |
| This paper| terminating | FSYNC              | 1      | 2       | 3         | 3          |
| This paper| perpetual   | SSYNC & ASYNC      | 1      | 2       | 3         | 3          |
| This paper| terminating | SSYNC & ASYNC      | 1      | 2       | 4         | 4          |

For example, if each robot has a five colors light, the difference between the asynchronous model and the semi-synchronous model disappears [7]. However, to the best of our knowledge, the impact of lights on myopic robots has not been studied yet.

### 1.2 Our Contributions

We focus on ring exploration and the impact of lights on myopic robots with $\phi = 1$. We consider the weakest assumption for lights: each robot can use only two colors for its light. Table 1 summarizes our contributions and related works. Note that robots with no light are equivalent to robots with a single color light.

As a main contribution, we prove that (i) in the fully synchronous model, two and three robots are necessary and sufficient to achieve perpetual and terminating exploration, respectively, and (ii) in the semi-synchronous and asynchronous models, three and four robots are necessary and sufficient to achieve perpetual and terminating exploration, respectively. These results clarify the power of lights for myopic robots since, without lights, five robots are necessary and sufficient to achieve terminating exploration in the fully synchronous model, and no terminating exploration algorithm exists in the semi-synchronous and asynchronous models. Interestingly, even if robots can observe nodes up to distance three (i.e., $\phi = 3$), five robots are required to achieve terminating exploration without light. This means that there exist some tasks that myopic luminous robots with small visibility can achieve, but that non-luminous robots with larger visibility cannot.

Similarly to previous works for myopic robots, all algorithms proposed in this paper assume some specific initial configurations because most configurations are not solvable. For example, when myopic robots are deployed so that no robot can observe other robots, they cannot achieve exploration. However, our perpetual exploration algorithms achieve the best possible property, that is, they are universal. This means that, in the fully synchronous model (resp., the semi-synchronous and asynchronous models), the proposed algorithm solves perpetual exploration from any solvable initial configuration with two (resp., three) robots and two colors. As for terminating exploration, we show that no universal algorithm exists. That is, in the fully synchronous model (resp., the semi-synchronous and asynchronous models), no algorithm may solve terminating exploration from any solvable initial configuration with three (resp., four) robots and two colors.

Due to space limitation, we omit some of proofs. The omitted proofs are given in the appendix.
2 Preliminaries

2.1 System model

The system consists of $n$ nodes and $k$ mobile robots. The nodes $v_0, v_1, \ldots, v_{n-1}$ form an undirected and unoriented ring-shaped graph, where a link exists between $v_i$ and $v_{i+1}$, for $i < n$, and between $v_{n-1}$ and $v_0$. For simplicity we consider mathematical operations on node indices as operations modulo $n$. Neither nodes nor links have identifiers or labels, and consequently robots cannot distinguish nodes and links. Robots do not know $n$, the size of the ring. Robots occupy some nodes of the ring. The distance between two nodes is the number of links in a shortest path between the nodes. The distance between two robots $a$ and $b$ is the distance between two nodes occupied by $a$ and $b$. Two robots $a$ and $b$ are neighbors if the distance between $a$ and $b$ is one. A set $S$ of robots is connected if the induced subgraph of nodes occupied by the robots in $S$ is connected; otherwise, $S$ is disconnected.

Robots we consider have the following characteristics and capabilities. Robots are identical, that is, robots execute the same deterministic algorithm and cannot be distinguished based on their appearance (in particular, their do not have unique identifiers). Robots are luminous, that is, each robot has a light (or state) that is visible to itself and other robots. A robot can choose the color of its lights from a discrete set $Col$. When the set $Col$ is finite, we denote by $\kappa$ the number of available colors (i.e., $\kappa = |Col|$). Robots have no other persistent memory and cannot remember the history of past actions. Robots cannot communicate with other robots explicitly, however they can communicate implicitly by observing positions and colors of other robots (for collecting information), and by changing their color and moving (for sending information). Each robot $r$ can observe positions and colors of robots within a fixed distance $\phi$ ($\phi > 0$) from its current position. Since robots are identical, they share the same $\phi$. If $\phi = \infty$, robots can observe all other robots in the ring. If $\phi = 1$, robots are myopic, that is, they can only observe robots that are located at neighboring nodes.

Each robot executes an algorithm by repeating three-phases cycles: Look, Compute, and Move (L-C-M). During the Look phase, the robot observes positions and colors of robots within distance $\phi$. During the Compute phase, the robot computes its next color and movement according to the observation in the Look phase. The robot may change its color at the end of the Compute phase. If the robot decides to move, it moves to a neighboring node during the Move phase. To model asynchrony of executions, we introduce the notion of scheduler that decides when each robot executes phases. When the scheduler makes robot $r$ execute some phase, we say the scheduler activates the phase of $r$ or simply activates $r$. We consider three types of synchronicity: the FSYNC (full-synchronous) model, the SSYNC (semi-synchronous) model, and the ASYNC (asynchronous) model. In the FSYNC model, the scheduler executes full cycles of all robots synchronously and concurrently. In the SSYNC model, the scheduler selects a non-empty subset of robots and executes full cycles of the selected robots synchronously and concurrently. In the ASYNC model, the scheduler executes cycles of robots asynchronously. Note that in the ASYNC model, a robot $r$ can move based on an outdated view observed previously by $r$. Throughout the paper we assume that the scheduler is fair, that is, each robot is activated infinitely often. We consider the scheduler as an adversary. That is, we assume that the scheduler is omniscient (it knows robot positions, colors, algorithms, etc.), and tries to activate robots in such a way that they fail executing the task.

In the sequel, $M_i(t)$ denotes the multiset of colors of robots located in node $v_i$ at instant $t$. If $v_i$ is not occupied by any robot at $t$, then $M_i(t) = \emptyset$ holds, and $v_i$ is free at instant $t$. 
Then, \( v_i \) is a *tower* at instant \( t \) if \( |M_i(t)| \geq 2 \). A *configuration* \( C(t) \) of the system at instant \( t \) is defined as \( C(t) = (M_0(t), M_1(t), \ldots, M_{n-1}(t)) \). If \( t \) is clear from the context, we simply write \( C = (M_0, M_1, \ldots, M_{n-1}) \). If there exists an index \( x \) such that \( M_{x+i} = M_{x-i} \) holds for any \( i \), or if \( M_{x+i} = M_{x-(i+1)} \) holds for any \( i \) (i.e., there exists at least one axis of symmetry in the configuration), configuration \( C \) is called *symmetric*.

When a robot observes its environment, it gets a *view* up to distance \( \phi \). Consider a robot \( r \) on node \( v_i \); then, \( r \) obtains two views: the forward view and the backward view. The forward and backward views of \( r \) are defined as \( V_f = (c_r, M_{i-\phi}, \ldots, M_{i-1}, M_i, M_{i+1}, \ldots, M_{i+\phi}) \), and \( V_b = (c_r, M_{i+\phi}, \ldots, M_{i+1}, M_i, M_{i-1}, \ldots, M_{i-\phi}) \), respectively, where \( c_r \) denotes \( r \)’s color. Since we assume unoriented rings (where robots may not share the same notion of left and right), each robot cannot distinguish its forward view from its backward view. If the forward view and the backward view of \( r \) are identical, then \( r \)’s view is *symmetric*. In this case, \( r \) cannot distinguish between the two directions when it moves, and the scheduler decides which direction \( r \) moves to. If \( r \) observes no other robot in its view, \( r \) is isolated.

### 2.2 Algorithm, execution, and problem

An algorithm is described as a set of rules. Each rule is represented in the following manner <Label>:<Guard>::<Action>. The guard <Guard> is a possible view obtained by a robot. If a forward or backward view of robot \( r \) matches a guard in an algorithm, we say \( r \) is enabled. We also say the corresponding rule <Label> is enabled. If a robot is enabled, the robot may change its color and move based on the corresponding action <Action> during the Compute and Move phases.

For an infinite sequence of configurations \( E = C_0, C_1, \ldots, C_t, \ldots \), we say \( E \) is an execution from initial configuration \( C_0 \) if, for every instant \( t \), \( C_{t+1} \) is obtained from \( C_t \) after some robots execute phases. We say \( C_i \) is the \( i \)-th configuration of execution \( E \).

A problem \( \mathcal{P} \) is defined as a set of executions: An execution \( E \) solves problem \( \mathcal{P} \) if \( E \in \mathcal{P} \) holds. An algorithm \( \mathcal{A} \) solves problem \( \mathcal{P} \) from initial configuration \( C_0 \) if any execution from initial configuration \( C_0 \) solves problem \( \mathcal{P} \). We simply say an algorithm \( \mathcal{A} \) solves problem \( \mathcal{P} \) if there exists an initial configuration \( C_0 \) such that \( \mathcal{A} \) solves \( \mathcal{P} \) from \( C_0 \). For configuration \( C \) and problem \( \mathcal{P} \), \( C \) is solvable for \( \mathcal{P} \) if there exists an algorithm (specific to \( C \)) that solves \( \mathcal{P} \) from initial configuration \( C \). Let \( C_s(\mathcal{P}) \) be a set of all configurations solvable for \( \mathcal{P} \). We say algorithm \( \mathcal{A} \) is universal with respect to problem \( \mathcal{P} \) if \( \mathcal{A} \) solves \( \mathcal{P} \) from any initial configuration in \( C_s(\mathcal{P}) \).

### 2.3 Exploration problems

In this paper, we consider perpetual exploration problem and terminating exploration problem in case of \( \phi = 1 \).

**Definition 1 (Perpetual exploration problem)** Perpetual exploration is defined as a set of executions \( E \) such that every node is infinitely many times visited by some robot in \( E \).

**Definition 2 (Terminating exploration problem)** Terminating exploration is defined as a set of executions \( E \) such that 1) every node is visited by at least one robot in \( E \) and 2) there exists a suffix of \( E \) such that no robots are enabled.
2.4 Descriptions

Let \( C = (M_0, \ldots, M_{n-1}) \) be a configuration. We say \( C' = (M'_0, \ldots, M'_{n'-1}) \) is a sub-configuration of \( C \) if there exists \( z \) such that \( M_{z+i} = M'_i \) holds for any \( i \) (\( 0 \leq i \leq n'-1 \)). In this case, we say \( n' \) is the length of sub-configuration \( C' \). We sometimes describe a sub-configuration \( C' = (M'_0, \ldots, M'_{n'-1}) \) by listing all colors in \( M'_i \) as the \( i \)-th column. That is, when \( M'_i = \{c'_1, \ldots, c'_{|M'_i|}\} \) holds for each \( i \) (\( 0 \leq i \leq n'-1 \)), we describe \( C' \) as follows:

\[
\begin{array}{ccc}
  c'_0 & c'_1 & \cdots & c'_{n'-1} \\
  c'_0 & c'_1 & \cdots & c'_{n'-1} \\
  \vdots & \vdots & \ddots & \vdots \\
  c'_0 & c'_{|M'_0|} & c'_{|M'_1|} & c'_{|M'_{n'-1}|}
\end{array}
\]

When \( M'_i = \emptyset \) holds, we write \( \emptyset^h \) instead of writing \( h \) columns with \( \emptyset \). For simplicity, when \( C' \) is a sub-configuration of \( C \) and all robots appear in \( C' \), we use \( C' \) instead of \( C \) to represent configuration \( C \). We also use this description to represent views of robots.

Throughout the paper, we consider the case of \( \phi = 1 \). We describe a rule in an algorithm in the following manner:

\[
\mathcal{R}_{\text{rule}} : \quad \begin{array}{ccc}
  c'_{-1,1} & c_{0,1} & c_{1,1} \\
  c'_{-1,2} & c_{0,2} & c_{1,2} \\
  \vdots & \vdots & \vdots \\
  c'_{-1,m-1} & (c_{0,m_0}) & c_{1,m_2}
\end{array} \quad :: \quad c_{\text{new}}, \text{Movement}
\]

Notation \( \mathcal{R}_{\text{rule}} \) is a label of the rule. The middle part represents a guard. This represents a view \( V = (c_{0,m_0}, M_{-1}, M_0, M_1) \), where \( M_i = \{c_{i,1}, \ldots, c_{i,m_i}\} \) holds for \( i \in \{-1,0,1\} \). Intuitively, each column represents colors of robots on a single node and a color within parentheses represents its current color. If a forward or backward view of robot \( r \) is equal to \( V \), \( r \) is enabled. In this case, \( r \) can execute an action represented by \( c_{\text{new}}, \text{Movement} \). Notation \( c_{\text{new}} \) represents a new color of the robot, and \( \text{Movement} \) represents the movement. Notation \( \text{Movement} \) can be \( \perp, \leftarrow, \rightarrow, \) or \( \leftarrow \lor \rightarrow \): 
1) \( \perp \) implies a robot does not move, 2) \( \leftarrow \) (resp., \( \rightarrow \)) implies a robot moves toward the node such that a set of robot colors is \( M_{-1} \) (resp., \( M_1 \)), and 3) \( \leftarrow \lor \rightarrow \) implies a robot moves toward one of two directions (the scheduler decides the direction). When the view \( V \) described in a guard is symmetric, \( \text{Movement} \) should be either \( \perp \) or \( \leftarrow \lor \rightarrow \). As an example, consider the following rule.

\[
\mathcal{R}_{\text{ex}} : \quad \begin{array}{ccc}
  \mathcal{G} \\
  \emptyset \ (W) \ \mathcal{G}
\end{array} \quad :: \quad \mathcal{G}, \rightarrow
\]

Robot \( r \) is enabled by \( \mathcal{R}_{\text{ex}} \) if 1) the color of \( r \) is \( W \), 2) the current node is occupied by two robots with colors \( \mathcal{G} \) and \( W \), 3) one neighboring node is occupied by no robot, and 4) another neighboring node is occupied by a robot with color \( \mathcal{G} \). If \( r \) is enabled by \( \mathcal{R}_{\text{ex}} \), \( r \) changes its color to \( \mathcal{G} \) and moves toward the node occupied by a robot with color \( \mathcal{G} \).
Algorithm 1 Fully-Synchronous Perpetual Exploration for $k = 2$

Initial configurations
GW and WG

Rules
0GW : $\emptyset$ (G) W :: G, ←
0WG : $\emptyset$ (W) G :: W, →

Figure 1: An example of an independent territory set. Notations W and G represent robots with colors W and G, respectively.

3 Full-synchronous Robots

3.1 A universal perpetual exploration algorithm for two robots with two colors

In this subsection, we provide a perpetual exploration algorithm for two robots with two colors. Note that, since one robot cannot achieve perpetual exploration clearly because the direction of its movement is decided by the scheduler, two robots are necessary to achieve perpetual exploration. A set of colors is $Col = \{G, W\}$. The algorithm is given in Algorithm 1. In the initial configuration, two robots with colors G and W stay at neighboring nodes. In this algorithm, the robot with color G moves against the other robot, and the robot with color W moves toward the other robot. This implies two robots move in the same direction. Since they move synchronously, the views of the two robots are not changed. Hence, the two robots continue to move and achieve perpetual exploration. Clearly we have the following theorem.

Theorem 3 In case of $\phi = 1$ and $k = 2$, Algorithm 1 solves perpetual exploration from initial configurations GW and WG for $n \geq 2$ in the FSYNC model.

In the following, we show that other initial configurations are unsolvable for $n \geq 6$. This implies Algorithm 1 is universal with respect to perpetual exploration for $n \geq 6$ in case of $\phi = 1$ and $k = 2$. To prove the impossibilities, we first prove Lemma 4. This lemma will be used for many impossibility proofs not only in the FSYNC model but also in the SSYNC and ASYNC models. For configuration $C$, we define $V_r(C)$ as a set of nodes occupied by at least one robot. We say a set of two neighboring nodes $T = \{v_i, v_{i+1}\}$ is a territory of robots on node $v$ if $v \in T$ holds. We say a territory set $T$ is independent if, for every pair of territories $T_1, T_2 \in T$, the distance between any node in $T_1$ and any node in $T_2$ is at least two (See Fig. 1).

Lemma 4 Consider a configuration $C$ such that no two nodes in $V_r(C)$ are neighbors and, for every node $v \in V_r(C)$, robots on $v$ have the same color. If there exists a territory set $T$ such that $T$ is independent and every node in $V_r(C)$ belongs to some territory in $T$, robots on $v \in V_r(C)$ cannot go out of their territory in $T$ after configuration $C$ in the FSYNC, SSYNC, and ASYNC models.
Proof: Assume that such a territory set $T$ exists. We prove the lemma by induction. At configuration $C$, every robot stays at its territory. Consider configuration $C'$ such that every robot stays at its territory. Since no two nodes in different territories share the same neighbor, each robot observes no robots on its neighbor nodes. This means the view of the robot is symmetric and consequently the scheduler can decide the direction of its movement. In addition, all robots on a single node have the same color, they make the same behaviors if the scheduler activates them at the same time. Hence, each robot moves to another node in its territory if it decides to move. That is, every robot stays at its territory again. Therefore, the lemma holds.

Now we go back to the FSYNC model, and prove that initial configurations other than $GW$ and $WG$ are unsolvable.

**Lemma 5** Assume $n \geq 6$. Let $C$ be a configuration such that two robots are disconnected. In this case, $C$ is unsolvable.

Proof: Let $d$ be the distance between two robots in $C$. Without loss of generality, we assume that two robots $r_1$ and $r_2$ occupy nodes $v_0$ and $v_d$ at $C$, respectively. We define territories $T_1$ and $T_2$ as follows: If $d = 2$ holds, we define $T_1 = \{v_0, v_{n-1}\}$ and $T_2 = \{v_d, v_{d+1}\}$. If $d > 2$, we define $T_1 = \{v_0, v_1\}$ and $T_2 = \{v_d, v_{d+1}\}$. Since $T = \{T_1, T_2\}$ is independent, two robots visit only nodes in $T_1 \cup T_2$ from Lemma 4. Therefore, they cannot achieve perpetual exploration.

**Lemma 6** Assume $n \geq 6$. Configurations $GG$ and $WW$ are unsolvable.

Proof: Since two robots have the same view, they move in a symmetric manner. If each robot moves toward the other robot, the robots just swap their positions. Hence, to achieve exploration, eventually each robot moves against the other robot. After the movement, the distance between them is three. Similarly to Lemma 5, robots cannot achieve perpetual exploration from the configuration.

From Theorem 3 and Lemmas 5 and 6, a set of solvable configurations is $C_s = \{GW, WG\}$. Therefore, we have the following theorem.

**Theorem 7** In case of $\phi = 1$, $k = 2$, and $Col = \{G, W\}$, Algorithm 7 is universal with respect to perpetual exploration for $n \geq 6$ in the FSYNC model.

### 3.2 Impossibility of terminating exploration with two robots

In this subsection, we prove that no algorithm solves terminating exploration for $k = 2$.

**Theorem 8** In case of $\phi = 1$ and $k = 2$, no algorithm solves terminating exploration in the FSYNC model. This holds even if robots can use an infinite number of colors.

Proof: Assume that such algorithm $A$ exists. Consider an execution $E = C_0, C_1, \ldots$ of $A$ in a $n_1$-node ring $R_1$ ($n_1 \geq 6$). Let $t$ be the minimum instant such that two robots terminate or become disconnected at $C_t$. Next, for some $n_2 > 2(t + 1)$, let us consider an execution $E' = C_0', C_1', \ldots$ of $A$ in a $n_2$-node ring $R_2$. Clearly, as long as two robots keep connected, they do not recognize the difference between $R_1$ and $R_2$. Hence, in $E'$, two robots move similarly to $E$ until $C_t'$. If two robots
Algorithm 2 Fully-Synchronous Terminating Exploration for $k = 3$

Initial configurations

WWW, GWW, WWG, and GWG

Rules

| Rule | Actions |
|------|---------|
| 0GW  | $\emptyset(G)W : G,\leftarrow$ |
| 0WG  | $\emptyset(W)G : W,\rightarrow$ |
| 0WW  | $\emptyset(W)W : G,\downarrow$ |
| GWW  | $G(W)W : W,\leftarrow$ |
| GWG  | $G(W)G : W,\leftarrow \lor \rightarrow$ |

terminate at $C'_t$, they have visited at most $2(t + 1)$ nodes and thus they do not achieve exploration. If two robots become disconnected at $C'_t$, we can define an independent territory set at $C'_t$. From Lemma 4, two robots cannot visit the remaining nodes and thus they cannot achieve exploration. This is a contradiction.

3.3 A terminating exploration algorithm for three robots with two colors

In this subsection, we give a terminating exploration algorithm for three robots with two colors in case of $n \geq 3$. A set of colors is $Col = \{G, W\}$. The algorithm is given in Algorithm 2.

Executions of Algorithm 2 for $n \geq 5$ are given in Fig. 2. We consider three robots $r_1$, $r_2$, and $r_3$. In the figure, $W_i$ (resp., $G_i$) represents robot $r_i$ with color $W$ (resp., $G$). Arrows represent that indicated robots are enabled. At configuration WWW, $r_1$ and $r_3$ are enabled by rule 0WW (Fig. 2(a)) and change their colors to $G$. At configuration GWG (Fig. 2(b)), robots $r_1$ and $r_2$ (i.e., a pair of robots GW) and $r_3$ (i.e., another robot G) move to the opposite directions by rules 0GW and GWG. Note that, since the view of $r_2$ is symmetric at configuration GWG, the scheduler decides the direction of $r_2$. This implies that the next configuration is GW∅∅G (Fig. 2(c)) or G∅∅WG. However, since the two configurations are symmetric to each other, robots move in the same manner after the configuration. At configuration GWW (Fig. 2(d)), robots $r_1$ and $r_2$ move to the opposite direction of $r_3$ by rules 0GW and GWW, and then the configuration becomes one in Fig. 2(e). Note that, since configuration WWG is symmetric to GWW, robots move in the same manner from configuration WWG. After configurations in Fig. 2(c)(e), robots $r_1$ and $r_2$ continue to move to the same direction by rules 0GW and 0WG. After $r_1$ and $r_2$ explore the ring, they reach $r_3$ (Fig. 2(f)). After robot $r_2$ moves, they terminate at a configuration in Fig. 2(g).

We can easily verify that Algorithm 2 works for $n = 3$ or $n = 4$. Hence we have the following theorem.

**Theorem 9** In case of $\phi = 1$ and $k = 3$, Algorithm 2 solves terminating exploration from initial configurations WWW, GWW, WWG, and GWG for $n \geq 3$ in the FSYNC model.

Note that we can construct another algorithm by swapping the roles of colors $G$ and $W$ in Algorithm 2. Clearly this algorithm solves terminating exploration from configurations such that colors $G$ and $W$ are swapped from solvable configurations for Algorithm 2. This implies configurations GGG, WGG, GGW, and GGW are also solvable. Hence, we have the following lemma.

**Lemma 10** If $k = 3$ holds and a set of colors is $\{G, W\}$, configurations WWW, WWG, WG, WGG, GWW, GWG, GG, and GGG are solvable for terminating exploration in the FSYNC model.
We also prove that there exists no universal algorithm with respect to terminating exploration for three robots with two colors. This validates the assumption that Algorithm 2 starts from some designated initial configuration.

**Theorem 11** In case of \( \phi = 1 \), \( k = 3 \), and \( \kappa = 2 \), no universal algorithm exists with respect to terminating exploration in the FSYNC model.

**Proof:** (Sketch) To prove the lemma by contradiction, we assume universal algorithm \( \mathcal{A} \) exists. We first prove that, by \( \mathcal{A} \), the distance between some pair of two robots becomes at least five in some large ring. Intuitively this holds because, if the three robots always stay within distance four, they cannot distinguish small rings and large rings, and terminate without exploration in large rings.

When the distance becomes at least five, some two robots should be connected. Otherwise, we can define an independent territory set and exploration is impossible from Lemma 4. Intuitively the two connected robots should explore the ring because another robot cannot go out of its territory. This implies the two robots should form sub-configuration \( GW \). In addition, since one of the two robots should move toward another robot to explore the ring, \( \mathcal{A} \) should include rule \( 0WGX : \emptyset(W)G :: X, \rightarrow \) or \( 0GWX : \emptyset(G)W :: X, \rightarrow \) for some \( X \in \text{Col} \).

Assume \( \mathcal{A} \) includes rule \( 0WGX \) (we can prove another case similarly). Consider a solvable configuration \( WGW \). From this configuration, two robots with color \( W \) move toward \( G \) at the same time by rule \( 0WGX \). This implies two robots with the same color make a tower in the next configuration. After the configuration, since the two robots move in the same manner, three robots behave as if there were only two robots. Similarly to Theorem 8, we can prove that they cannot achieve terminating exploration. This is a contradiction because \( \mathcal{A} \) cannot achieve terminating exploration from a solvable initial configuration. \( \square \)

## 4 Semi-synchronous and Asynchronous Robots

### 4.1 Impossibility of perpetual exploration with two robots

In this subsection, we prove that two robots are not sufficient to achieve perpetual exploration in the SSYNC model. Clearly this impossibility result holds in the ASYNC model.
Algorithm 3 Asynchronous Perpetual Exploration for \( k = 3 \)

**Initial configurations**

\[ \text{WWG, WGG, GWW, GGW, W'W G, W'W' G, and W'G'}. \]

**Rules**

\[
\begin{align*}
0GW & : \emptyset (G) W :: \rightarrow \\
0TW & : \emptyset (W) W :: \rightarrow \\
0TG & : \emptyset (G) G :: \leftarrow \\
0WG & : \emptyset (W) G :: \rightarrow
\end{align*}
\]

**Theorem 12** In case of \( \phi = 1 \) and \( k = 2 \), no algorithm solves perpetual exploration in the SSYNC model. This holds even if robots can use an infinite number of colors.

**Proof:** Assume that such algorithm \( A \) exists for \( n \geq 6 \). First consider the case that initially two robots \( r_1 \) and \( r_2 \) are connected. Clearly some robot \( r_i (i \in \{1, 2\}) \) eventually moves against another robot. At that time, we assume that the scheduler activates only \( r_i \). Since another robot does not move, the two robots become disconnected.

From the above fact, two robots are initially disconnected or eventually become disconnected. Let us consider the configuration such that two robots are disconnected. In this case, we can define an independent territory set, and thus robots cannot achieve exploration from Lemma 4. This is a contradiction. \( \square \)

### 4.2 A universal perpetual exploration algorithm for three robots with two colors

In this subsection, we give a universal perpetual exploration algorithm for three robots with two colors in the SSYNC and ASYNC models. We first give a perpetual exploration algorithm by three robots with two colors in the ASYNC model, and after that we prove the algorithm is universal in the SSYNC and ASYNC models. A set of colors is \( Col = \{G, W\} \). The algorithm is given in Algorithm 3.

Executions of Algorithm 3 for \( n \geq 4 \) are given in Fig. 3. Let us consider configuration \( \text{WWG} \), and assume that \( r_1, r_2, \) and \( r_3 \) compose the configuration in this order (Fig. 3(a)). Here only \( r_3 \) is enabled with rule 0GW, and \( r_3 \) moves toward \( r_2 \). In a configuration in Fig. 3(b), only \( r_2 \) is enabled with rule 0TW. If \( r_2 \) is activated, \( r_2 \) changes its color to \( G \) and moves toward \( r_1 \) (Fig. 3(c)). Note that, in the ASYNC model, after \( r_2 \) changes its color, some robots may observe the intermediate configuration before \( r_2 \) moves toward \( r_1 \). However, since no rule matches the intermediate configuration, robots do not move based on the configuration. After \( r_2 \) moves from Fig. 3(c) by rule 0TG, the sub-configuration becomes \( \text{WWG} \) (Fig. 3(e)) but the robots change their positions from Fig. 3(a) to Fig. 3(e). Similarly, robots repeat the behavior from Fig. 3(a) to Fig. 3(e), and they achieve perpetual exploration. From configuration \( \text{WGG} \) in Fig. 3(d), \( r_1 \) moves by rule 0WG and becomes a configuration in Fig. 3(c). After that, they move similarly to the case from a
configuration in Fig. 3(a). These executions include configurations:

\[ \text{WWG, WGG, } \underbrace{\text{G}}_{\text{W W}^*}, \text{ and } \underbrace{\text{W}}_{\text{G G}^*} \]

and consequently from these configurations robots can achieve perpetual exploration. Since remaining configurations:

\[ \text{GWW, GGW, } \underbrace{\text{G}}_{\text{W W}^*}, \text{ and } \underbrace{\text{W}}_{\text{G G}^*} \]

are symmetric to the above configurations, robots can also achieve perpetual exploration from the configurations. Therefore, we have the following theorem.

**Theorem 13** In case of \( \phi = 1 \) and \( k = 3 \), Algorithm 3 solves perpetual exploration from initial configurations:

\[ \text{WWG, WGG, GWW, GGW, } \underbrace{\text{G}}_{\text{W W}^*}, \underbrace{\text{G}}_{\text{W W}^*}, \text{ and } \underbrace{\text{W}}_{\text{G G}^*} \]

for \( n \geq 3 \) in the ASYNC model.

We can also show that other initial configurations are unsolvable for \( n \geq 9 \) in the SSYNC model. This implies Algorithm 3 is universal with respect to perpetual exploration for \( n \geq 9 \) in the SSYNC and ASYNC models.

**Theorem 14** In case of \( \phi = 1 \), \( k = 3 \), and \( \text{Col} = \{G, W\} \), Algorithm 3 is universal with respect to perpetual exploration for \( n \geq 9 \) in the SSYNC and ASYNC models.

### 4.3 Impossibility of terminating exploration with three robots

In this subsection, we prove that three robots are not sufficient to achieve terminating exploration in the SSYNC model. Clearly this impossibility result holds in the ASYNC model.

**Theorem 15** In case of \( \phi = 1 \) and \( k = 3 \), no algorithm solves terminating exploration in the SSYNC model. This holds even if robots can use an infinite number of colors.

**Proof:** For contradiction, assume that such algorithm \( \mathcal{A} \) exists. Consider an execution \( E = C_0, C_1, \ldots \) of \( \mathcal{A} \) in a \( n_1 \)-node ring \( R_1 \) (\( n_1 \geq 9 \)). Let \( t \) be the minimum instant such that three robots terminate or become disconnected at \( C_t \). Next, for \( n_2 = 4(t + 1) \), let us consider an execution...
Algorithm 4 Asynchronous Terminating Exploration for $k = 4$

Initial configurations

WWGG, WWWW, WWGW, GGWW, GWWW, WGWW.

Rules

\begin{align*}
0GW & : \emptyset (G) W :: G, \rightarrow \\
GW & : G (G) W :: G, \rightarrow \\
0TW & : G \emptyset (W) W :: G, \rightarrow \\
GTW & : G G (W) W :: G, \rightarrow \\
0TG & : \emptyset (W) G :: W, \leftarrow \\
0WG & : \emptyset (W) G :: W, \rightarrow 
\end{align*}

$E' = C'_0, C'_1, \ldots$ of $A$ in a $n_2$-node ring $R_2$ such that the scheduler activates robots similarly to $E$. Clearly, as long as three robots keep connected, they do not recognize the difference between $R_1$ and $R_2$. Hence, in $E'$, three robots move similarly to $E$ until $C'_t$. Note that, since robots have visited at most $3(t + 1)$ until $C'_t$, they must explore the remaining $t + 1$ nodes. If three robots terminate at $C'_t$, they do not achieve exploration. If three robots become disconnected at $C'_t$, robots cannot achieve exploration from the proof of Theorem 14. □

4.4 A terminating exploration algorithm for four robots with two colors

In this subsection, we give a terminating exploration algorithm for four robots with two colors in case of $n \geq 4$. A set of colors is $Col = \{G, W\}$. The algorithm is given in Algorithm 4. Note that rules 0GW, 0TW, 0TG are identical to Algorithm 3. Hence, once three robots construct sub-configurations

$$C_{pe} = \left\{ WWG, G W W, W G W G \right\},$$

they explore the ring similarly to Algorithm 3.

Executions of Algorithm 4 for $n \geq 5$ are given in Fig. 4. At configuration WWGG (Fig.4(a)) only $r_3$ can move by rule GGW, and the configuration becomes one in Fig.4(b) after it moves. Since $r_1$, $r_2$, and $r_3$ form a sub-configuration in $C_{pe}$, they explore the ring. Since $r_4$ does not move, three robots eventually join $r_4$ from the opposite direction (Fig.4(c)).

Figure 5 shows executions after a configuration in Fig.4(c). In this figure, we reassign indices to robots: $r_1$, $r_2$, $r_3$, and $r_4$ form sub-configuration GWWG in this order. Since $r_1$ and $r_4$ are enabled, robots can make several behaviors depending on activation by the scheduler. Notation LC-i (resp., M-i) means the scheduler activates Look and Compute phases (resp., Move phase) of $r_i$. Although the scheduler can activate Look and Compute phases separately, we combine the two phases because a Look phase does not change a configuration. At configuration GWWG, the scheduler can activate $r_1$ and $r_4$ to change the configuration. If the scheduler activates exactly one
robot, we only show the case of \( r_1 \) because the case of \( r_4 \) is symmetric to \( r_1 \). An arrow below a robot means that the robot decides to move to the direction. In every execution from configuration GWWW, robots eventually terminate. That is, Algorithm 4 solves terminating exploration from initial configuration WWGG.

We consider other initial configurations WWWG and WWGW in Fig. 6. From initial configuration WWWG (Fig. 6(a)), robots eventually form configuration WWGG (Fig. 6(e)) and thus they can solve terminating exploration. From initial configuration WWGW (Fig. 6(f)), robots form a configuration in Fig. 6(g) and the configuration is the same as in Fig. 6(b). Hence, they can solve terminating exploration.

Since configurations GGWW, GWWW, and WGWW are symmetric to WWGG, WWWG, and WWGW, respectively, we have the following theorem.

**Theorem 16** In case of \( \phi = 1 \) and \( k = 4 \), Algorithm 4 solves terminating exploration from initial configurations WWGG, WWWG, WWGW, GGWW, GWWW, and WGWW for \( n \geq 5 \) in the ASYNC model.

Note that we can construct another algorithm by swapping the roles of colors G and W in Algorithm 4. Clearly this algorithm solves terminating exploration from configurations such that colors G and W are swapped from solvable configurations for Algorithm 4. This implies configurations GGGW, GGGW, GGWW, and GWGG are also solvable. Hence, we have the following lemma.

**Lemma 17** If \( k = 4 \) holds and a set of colors is \( \{ G, W \} \), configurations WWGG, WWWG, WWGW, GGWW, GWWW, WGGW, GGWG, GGGW, and GWGG are solvable for terminating exploration in the ASYNC model.
We also prove that there exists no universal algorithm with respect to terminating exploration for four robots with two colors. This validates the assumption that Algorithm 4 starts from some designated initial configuration.

**Theorem 18** In case of $\phi = 1$, $k = 4$, and $\kappa = 2$, no universal algorithm exists with respect to terminating exploration in the SSYNC and ASYNC models.

**Proof:** (Sketch) Similarly to Theorem 11, we can show that the distance between some pair of robots become large and some three robots must explore the ring. At that time, we also show that the three robots always form a sub-configuration in

$$C_{exp} = \{ WGG, GGW, GWW, WWG, G\text{ W W'}, W' W' G' G' G' G \}.$$  

Note that this is a set of solvable configurations for perpetual exploration. Since a universal algorithm makes the three robots explore the ring by changing their sub-configuration from one in $C_{exp}$ to another in $C_{exp}$, we can show that the algorithm should include some set of rules. Lastly, we prove the set of rules makes some solvable initial configuration unsolvable.  

5 Conclusions

In this paper, we investigated the possibility of exploration algorithms for myopic luminous robots evolving in uniform ring-shaped networks. Considering weakest possible assumptions for myopia and luminosity, we proved that: (i) in the fully synchronous model, two and three robots are necessary and sufficient to achieve perpetual and terminating exploration, respectively, and (ii) in the semi-synchronous and asynchronous models, three and four robots are necessary and sufficient to achieve perpetual and terminating exploration, respectively. These tight results characterize the power of lights for myopic robots since, without lights, five robots are necessary and sufficient to achieve terminating exploration in the fully synchronous model, and no terminating exploration algorithm exists in the semi-synchronous and asynchronous models. We also showed that our perpetual exploration algorithms are universal, and that no universal algorithm exists for terminating exploration.
This paper leaves open many issues with respect to problem solvability for myopic luminous robots. In case of non-myopic luminous robots, the difference between the semi-synchronous model and the asynchronous model disappears. Does this difference still hold for myopic luminous robots? If visibility $\phi$ is large, robots may be able to use distance to neighboring robots to store information instead of lights. Now, is there some relation between tasks achieved by myopic luminous robots with a large number of colors, and tasks achieved by non-luminous robots with large visibility? Is there a tradeoff between the visibility distance and the number of colors? It is also interesting to consider other tasks and topologies with myopic luminous robots.

Acknowledgments

This work was partially supported by a mobility scholarship of the author at Sorbonne University in the frame of the Erasmus Mundus Action 2 Project TEAM Technologies for Information and Communication Technologies, funded by the European Commission. This publication reflects the view only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein. This work was also partially supported by JSPS KAKENHI Grant Number 18K11167.

References

[1] Lélia Blin, Alessia Milani, Maria Potop-Butucaru, and Sébastien Tixeuil. Exclusive perpetual ring exploration without chirality. In 24th International Symposium on Distributed Computing, pages 312–327, 2010.

[2] François Bonnet, Alessia Milani, Maria Potop-Butucaru, and Sébastien Tixeuil. Asynchronous exclusive perpetual grid exploration without sense of direction. In 15th International Conference on Principles of Distributed Systems, pages 251–265, 2011.

[3] Jérémie Chalopin, Paola Flocchini, Bernard Mans, and Nicola Santoro. Network exploration by silent and oblivious robots. In 36th International Workshop on Graph Theoretic Concepts in Computer Science, pages 208–219, 2010.

[4] Gianlorenzo D’Angelo, Alfredo Navarra, and Nicolas Nisse. A unified approach for gathering and exclusive searching on rings under weak assumptions. Distributed Computing, 30(1):17–48, 2017.

[5] Gianlorenzo D’Angelo, Gabriele Di Stefano, Ralf Klasing, and Alfredo Navarra. Gathering of robots on anonymous grids and trees without multiplicity detection. Theoretical Computer Science, 610:158–168, 2016.

[6] Gianlorenzo D’Angelo, Gabriele Di Stefano, and Alfredo Navarra. Gathering on rings under the look-compute-move model. Distributed Computing, 27(4):255–285, 2014.

[7] Shantanu Das, Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Masafumi Yamashita. Autonomous mobile robots with lights. Theoretical Computer Science, 609:171–184, 2016.
[8] Ajoy Kumar Datta, Anissa Lamani, Lawrence L. Larmore, and Franck Petit. Ring exploration by oblivious agents with local vision. In IEEE 33rd International Conference on Distributed Computing Systems, pages 347–356, 2013.

[9] Ajoy Kumar Datta, Anissa Lamani, Lawrence L. Larmore, and Franck Petit. Ring exploration by oblivious robots with vision limited to 2 or 3. In 15th International Symposium on Stabilization, Safety, and Security of Distributed Systems, pages 363–366, 2013.

[10] Ajoy Kumar Datta, Anissa Lamani, Lawrence L. Larmore, and Franck Petit. Enabling ring exploration with myopic oblivious robots. In 2015 IEEE International Parallel and Distributed Processing Symposium Workshop, pages 490–499, 2015.

[11] Mattia D’Emidio, Daniele Frigioni, and Alfredo Navarra. Characterizing the computational power of anonymous mobile robots. In 36th IEEE International Conference on Distributed Computing Systems, pages 293–302, 2016.

[12] Stéphane Devismes, Anissa Lamani, Franck Petit, Pascal Raymond, and Sébastien Tixeuil. Optimal grid exploration by asynchronous oblivious robots. In 14th International Symposium on Stabilization, Safety, and Security of Distributed Systems, pages 64–76, 2012.

[13] Stéphane Devismes, Anissa Lamani, Franck Petit, and Sébastien Tixeuil. Optimal torus exploration by oblivious robots. In Third International Conference on Networked Systems, pages 183–199, 2015.

[14] Stéphane Devismes, Franck Petit, and Sébastien Tixeuil. Optimal probabilistic ring exploration by semi-synchronous oblivious robots. Theoretical Computer Science, 498:10–27, 2013.

[15] Paola Flocchini, David Ilcinkas, Andrzej Pelc, and Nicola Santoro. Remembering without memory: Tree exploration by asynchronous oblivious robots. Theoretical Computer Science, 411(14-15):1583–1598, 2010.

[16] Paola Flocchini, David Ilcinkas, Andrzej Pelc, and Nicola Santoro. Computing without communicating: Ring exploration by asynchronous oblivious robots. Algorithmica, 65(3):562–583, 2013.

[17] Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Peter Widmayer. Gathering of asynchronous robots with limited visibility. Theoretical Computer Science, 337(1-3):147–168, 2005.

[18] Nao Fujinaga, Yukiko Yamauchi, Hirotaka Ono, Shuji Kijima, and Masafumi Yamashita. Pattern formation by oblivious asynchronous mobile robots. SIAM Journal on Computing, 44(3):740–785, 2015.

[19] Samuel Guilbault and Andrzej Pelc. Gathering asynchronous oblivious agents with local vision in regular bipartite graphs. Theoretical Computer Science, 509:86–96, 2013.

[20] Samuel Guilbault and Andrzej Pelc. Gathering asynchronous oblivious agents with restricted vision in an infinite line. In 15th International Symposium on Stabilization, Safety, and Security of Distributed Systems, pages 296–310, 2013.

[21] Adam Heriban, Xavier Défago, and Sébastien Tixeuil. Optimally gathering two robots. In 19th International Conference on Distributed Computing and Networking, pages 3:1–3:10, 2018.
[22] Ralf Klasing, Adrian Kosowski, and Alfredo Navarra. Taking advantage of symmetries: Gathering of many asynchronous oblivious robots on a ring. *Theoretical Computer Science*, 411(34-36):3235–3246, 2010.

[23] Ralf Klasing, Euripides Markou, and Andrzej Pelc. Gathering asynchronous oblivious mobile robots in a ring. *Theoretical Computer Science*, 390(1):27–39, 2008.

[24] Ichiro Suzuki and Masafumi Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM Journal on Computing*, 28(4):1347–1363, 1999.

[25] Yukiko Yamauchi, Taichi Uehara, Shuji Kijima, and Masafumi Yamashita. Plane formation by synchronous mobile robots in the three-dimensional euclidean space. *Journal of the ACM*, 64(3):16:1–16:43, 2017.
To prove Theorem 11 by contradiction, we assume universal algorithm $A$ exists. We assume $Col = \{G, W\}$. We consider a ring with $n > 4 \cdot 10^3$ nodes and an execution $E = C_0, C_1, \ldots$ of $A$ in the ring. Note that, since three robots can visit at most three new nodes in each configuration, $E$ includes at least $n/3 > (4/3) \cdot 10^3$ configurations.

The outline of the proof is as follows. We first prove that three robots eventually become disconnected (Lemma 19), and at that time two robots compose a sub-configuration GW (Lemma 21). Then, we prove that some rule is necessary to make the two robots explore the ring (Lemma 22), however the rule makes a solvable configuration unsolvable (Lemma 23).

**Lemma 19** The distance between some pair of two robots becomes at least five before the $10^3$-th configuration of execution $E$.

*Proof:* For contradiction, assume that the distance between any pair of robots is at most four until the $10^3$-th configuration of $E$. For each configuration $C_i$ ($0 \leq i \leq 10^3$), we define $C'_i$ as a sub-configuration of $C_i$ such that $C'_i$ contains five nodes (i.e., the length of $C'_i$ is five), every robot occupies one of the five nodes, and some robot occupies the first node of $C'_i$. Since each robot has one of two colors and occupies one of five nodes, the number of such sub-configurations is at most $10^3$. Hence, for some $t$ and $u$ ($t < u \leq 10^3$), sub-configurations $C'_t$ and $C'_u$ are identical. In the following, we consider two cases.

The first case is that the five nodes contained in $C'_t$ and $C'_u$ are identical. This implies configurations $C_t$ and $C_u$ are identical. Since $A$ is deterministic, robots repeat the behavior from $C_t$ to $C_u$ after $C_u$. Since robots can visit at most $3 \cdot 10^3$ nodes before configuration $C_t$, they cannot visit remaining nodes after $C_u$. Thus robots cannot complete exploration.

The second case is that the five nodes contained in $C'_t$ and $C'_u$ are not identical. This means robots change their position from $C_t$ to $C_u$. However, since no robots exist out of the five nodes, the behaviors of robots depend only on the sub-configuration. Hence, robots repeat the behaviors from $C_t$ to $C_u$ after $C_u$. This implies robots continue to explore the ring, and thus they cannot terminate.

For both cases, robots cannot achieve terminating exploration. This is a contradiction. \(\square\)

From Lemma 19, the distance between some pair of two robots becomes at least five before the $10^3$-th configuration. Let $t_1$ be the smallest instant such that the distance between some pair of two robots is at least five at $C_{t_1}$. Clearly $t_1 \leq 10^3$ holds. Since three robots can visit at most $3(t_1 + 1)$ nodes before $C_{t_1}$, at least $10^3 - 3$ unexplored nodes exist at $C_{t_1}$. The next two lemmas show that, to explore the unexplored nodes, robots have to construct a specific sub-configuration.

**Lemma 20** If a tower exists at configuration $C_{t_1}$, robots break the tower before they visit three unexplored nodes.

*Proof:* Assume that a tower exists at configuration $C_{t_1}$. Since the number of robots is three and the distance between some pair of robots is at least five at $C_{t_1}$, the tower includes two robots and the views of the robots are symmetric. The view of a robot not in the tower is also symmetric. Hence, we can define their territories similarly to Lemma 4. This implies robots do not go out of their territories before they break the tower. Therefore, the lemma holds. \(\square\)
Let $t_2$ be the smallest instant such that the distance between some pair of two robots is at least five and no tower exists at $C_{t_2}$. Since at least $10^3 - 3$ unexplored nodes exist at $C_{t_1}$, at least $10^3 - 7$ unexplored nodes exist at $C_{t_2}$ from Lemma 20.

**Lemma 21** Configuration $C_{t_2}$ includes sub-configuration $GW^iX$ or $WG^iX$ for some $i \geq 3$ and $X \in \text{Col}$ (or sub-configuration symmetric to one of them).

**Proof:** For contradiction, assume that $C_{t_2}$ does not include such sub-configurations. Since the distance between some pair of two robots is at least five and no tower exists, $C_{t_2}$ includes either 1) sub-configuration $Y^iZ^jX$ for some $i+j \geq 3$ and $X,Y,Z \in \text{Col}$, 2) sub-configuration $YY^iX$ for some $i \geq 3$ and $X,Y \in \text{Col}$, or 3) sub-configuration symmetric to Case 1 or 2.

In Case 1, we can define an independent territory set. This implies robots cannot visit remaining unexplored nodes from Lemma 4. Thus, they cannot achieve exploration.

Let us consider Case 2. We assume robots $r_1$ and $r_2$ have color $Y$ and $r_3$ has color $X$ at $C_{t_2}$. Since the scheduler can make $r_3$ move forward and backward, $r_1$ and $r_2$ should move to achieve exploration. Since the view of $r_1$ is identical (symmetric) to the view of $r_2$, $r_1$ and $r_2$ make symmetric behaviors. If $r_1$ moves toward $r_2$, $r_1$ and $r_2$ swap their positions and visit no unexplored nodes. If $r_1$ moves against $r_2$, its sub-configuration becomes $Y^i0Y^iX'$ for some $j \in \{i-2, i-1, i\}$ and $X', Y' \in \text{Col}$. Here $Y'$ is a new color of $r_1$ and $r_2$, and $j$ and $X'$ depend on the behavior of $r_3$. For any case, this sub-configuration reduces to Case 1. This implies robots cannot achieve exploration.

We consider Case 3 similarly to Cases 1 and 2. For every case, robots cannot achieve exploration. This is a contradiction.

We consider Case 2. We assume robots $r_1$ and $r_2$ have color $Y$ and $r_3$ has color $X$ at $C_{t_2}$. Since the scheduler can make $r_3$ move forward and backward, $r_1$ and $r_2$ should move to achieve exploration. Since the view of $r_1$ is identical (symmetric) to the view of $r_2$, $r_1$ and $r_2$ make symmetric behaviors. If $r_1$ moves toward $r_2$, $r_1$ and $r_2$ swap their positions and visit no unexplored nodes. If $r_1$ moves against $r_2$, its sub-configuration becomes $Y^i0Y^iX'$ for some $j \in \{i-2, i-1, i\}$ and $X', Y' \in \text{Col}$. Here $Y'$ is a new color of $r_1$ and $r_2$, and $j$ and $X'$ depend on the behavior of $r_3$. For any case, this sub-configuration reduces to Case 1. This implies robots cannot achieve exploration.

We consider Case 3 similarly to Cases 1 and 2. For every case, robots cannot achieve exploration. This is a contradiction.

From Lemma 21, at configuration $C_{t_2}$, two robots with colors $G$ and $W$ are away from the other robot. Since the remaining isolated robot cannot explore by itself, two robots with colors $G$ and $W$ explore the remaining part of the ring. The following lemma proves that $\mathcal{A}$ includes some rules to realize this behavior.

**Lemma 22** We consider the following rules:

- $0WGX : 0(W)G : X, \rightarrow$ (for some $X \in \text{Col}$)
- $0GWX : 0(G)W : X, \rightarrow$ (for some $X \in \text{Col}$)

Algorithm $\mathcal{A}$ includes a rule $0WGX$ or $0GWX$

**Proof:** For contradiction, assume that $\mathcal{A}$ includes neither $0WGX$ nor $0GWX$. From Lemma 21 at configuration $C_{t_2}$, two robots with colors $G$ and $W$ are away from the other robot and compose sub-configuration $GW$ or $WG$. Since sub-configurations $GW$ and $WG$ are symmetric, we consider only sub-configuration $GW$. To complete exploration, the two robots must explore the remaining part of the ring.

Since $\mathcal{A}$ includes neither $0WGX$ nor $0GWX$, a robot in a sub-configuration $GW$ does not move toward another robot in $GW$. We consider three cases about the movement of robots in $GW$:

1. One or both of robots in $GW$ move against the other robots in $GW$. In this case, these two robots become isolated. At this configuration, we can define an independent territory set, and thus robots cannot achieve exploration from Lemma 4.
2. Both robots in $GW$ change their colors and do not move. In this case, robots just swap their colors and thus the views of them are not changed. This implies robots repeatedly change their colors and never move. Therefore, robots cannot achieve exploration.

3. One robot in $GW$ changes its color. That is, two robots in $GW$ at $C_{t_2}$ compose sub-configuration $WW$ or $GG$ at $C_{t_2+1}$. Recall that, from Lemma 21, sub-configurations of $C_{t_2}$ include $GW^0X$ or $WG^0iX$ for some $i \geq 3$ and $X \in Col$. This implies sub-configurations of $C_{t_2+1}$ include $YY\psi X'$ for some $j \in \{i, i + 1\}$ and some $X', Y \in Col$. Here $X'$ and $j$ depend on the behavior of a robot with color $X$ at $C_{t_2}$. As proved in Case 2 of Lemma 21, from sub-configuration $YY\psi X'$, robots cannot achieve exploration.

For every case, robots cannot achieve exploration. This is a contradiction. 

From Lemma 22, $A$ includes a rule $0WGX$ or $0GWX$. The following lemma shows, if $A$ includes rule $0WGX$ or $0GWX$, $A$ cannot solve terminating exploration from some solvable configuration.

**Lemma 23** 1) If $A$ includes rule $0WGX$, $A$ cannot solve terminating exploration from configuration $WG$. 2) If $A$ includes rule $0GWX$, $A$ cannot solve terminating exploration from configuration $GWG$.

**Proof:** Without loss of generality, we consider only statement 1 (Statement 2 is obtained by swapping the roles of two colors). Assume that, initially, robots $r_1$, $r_2$, and $r_3$ construct a configuration $WGW$ in this order. Since $A$ includes rule $0WGX$, the next configuration includes a sub-configuration

$$\begin{array}{c}
X \\
Y X \text{ or } X Y \\
\end{array}$$

for some $X, Y \in Col$. That is, $r_1$ and $r_3$ assign $X$ to its color, move based on rule $0WGX$, and makes a tower, and $r_2$ assigns $Y$ to its color and moves (or stays) based on some rule. Note that $r_1$ and $r_3$ stay at the same node with the same color. This implies, since the views of $r_1$ and $r_3$ are identical, they move together after this configuration. Similarly to Lemma 19, the distance between two agents becomes at least five. In this configuration, since $r_1$ and $r_3$ move together, one node is occupied by $r_1$ and $r_3$ with the same color, and another node is occupied by $r_2$. Since we can define an independent territory set, robots cannot achieve exploration from Lemma 4. That is, $A$ cannot solve terminating exploration from configuration $WG$. 

From Lemmas 22 and 23, $A$ cannot solve terminating exploration from configuration $WGW$ or $GWG$. However, from Lemma 10, configurations $WGW$ and $GWG$ are solvable. This contradicts to the assumption that $A$ is universal. Therefore, Theorem 11 holds.

**B  Proof of Theorem 14**

To prove Theorem 14, we show that initial configurations other than ones in Theorem 13 are unsolvable for $n \geq 9$ in the SSYNC model. First we prove the following simple lemma.

**Lemma 24** Consider a configuration $C$ such that, for some node $v_i$, $v_i$ is occupied by at least one robot, all robots on $v_i$ have the same color, and $v_{i-1}$ and $v_{i+1}$ are occupied by no robot. After configuration $C$, the following statements hold.
(1) If no robots appear in \( v_{i-2} \) and \( v_{i+1} \), robots on \( v_i \) cannot visit nodes other than \( v_{i-1} \) and \( v_i \).

(2) If no robots appear in \( v_{i+2} \) and \( v_{i-1} \), robots on \( v_i \) cannot visit nodes other than \( v_{i+1} \) and \( v_i \).

Proof: We prove only statement (1) because we can similarly prove statement (2). Since robots on \( v_i \) have the same color, they have the same view. Consequently, if the scheduler repeatedly activates them at the same time, they continue to make the same behaviors. In addition, since the views of robots on \( v_i \) are symmetric, the scheduler decides the directions of their movements. Consequently, robots on \( v_i \) move to \( v_{i-1} \) if they decide to move. From statement (1), no robots exist on \( v_{i-2} \) and \( v_i \), and hence the views of the robots are the same as in the previous configuration. By repeating such behaviors, the robots can visit only \( v_{i-1} \) and \( v_i \). Therefore, the lemma holds. \( \square \)

In the following, we show that initial configurations other than ones in Theorem 13 are unsolvable.

**Lemma 25** Assume \( n \geq 6 \). Let \( C \) be a configuration such that two or three robots with the same color stay at a single node. In this case, \( C \) is unsolvable in the SSYNC model.

Proof: We consider three cases 1) three robots with the same color stay at a single node, 2) two robots with the same color stay at a single node and the distance to another robot is at least two, 3) two robots with the same color stay at a single node and the distance to another robot is one, and 4) three robots stay at a single node but one robot has a different color from other two robots. In Cases 1 and 2, robots cannot achieve exploration from Lemma 3.

To consider Cases 3 and 4, assume that \( r_1 \) and \( r_2 \) have the same color and stay at node \( v \), and \( r_3 \) stays at \( v \) or a neighbor of \( v \). We consider the scheduler that repeats 1) activation of \( r_1 \) and \( r_2 \) and 2) activation of \( r_3 \). Since \( r_1 \) and \( r_2 \) have the same view, they make the same behavior. If \( r_1 \) and \( r_2 \) join the node with \( r_3 \) and they have the same color as \( r_3 \), this reduces to Case 1. If such a situation never happens, to achieve exploration, eventually \( r_1 \) and \( r_2 \) move against \( r_3 \), or \( r_3 \) moves against \( r_1 \) and \( r_2 \). Both cases reduce to Case 2. Therefore, the lemma holds. \( \square \)

**Lemma 26** Assume \( n \geq 9 \). Let \( C \) be a configuration such that the longest distance between two robots is at least four. In this case, \( C \) is unsolvable in the SSYNC model.

Proof: We consider three robots \( r_1, r_2, \) and \( r_3 \). If the distance between any pair of robots is at least three at \( C \), robots cannot achieve exploration from Lemma 4. To consider the remaining cases, without loss of generality, we assume that the distance between \( r_1 \) and \( r_2 \) is at most two at \( C \). We assume that \( r_1 \) occupies \( v_0 \) and \( r_2 \) occupies \( v_0, v_1, \) or \( v_2 \) at \( C \). Let \( v_d \) be the node occupied by \( r_3 \) at \( C \). Since the distance between some pair of two robots is at least four, we assume \( d \geq 4 \) and \( d \leq n - 4 \) hold without loss of generality.

In the following, we consider the scheduler that activates one robot at each instant. We prove that, under this scheduler, robots cannot achieve exploration from every possible configuration. More concretely, we prove that \( r_1 \) and \( r_2 \) always stay at \( v_{-1} \) (i.e., \( v_{n-1} \), \( v_0, v_1, \) or \( v_2 \)). This implies, since no robots appear in \( v_{d-1} \) and \( v_{d+2} \), \( r_3 \) cannot visit nodes other than \( v_d \) and \( v_{d+1} \) from Lemma 24. Therefore, robots cannot achieve exploration.

First consider the case that \( r_2 \) occupies \( v_2 \) at \( C \). In this case, the view of every robot is symmetric. Consequently, if a robot moves, its direction is decided by the scheduler. This implies
$r_1$ cannot move because, if $r_1$ moves, it moves to $v_{-1}$ (i.e., $v_{n-1}$), and from Lemma 4 robots cannot achieve exploration. If $r_2$ moves, it moves to $v_1$ and this reduces the next case.

Next, consider the case that $r_2$ occupies $v_1$ at $C$. We consider two sub-cases.

- Case that $r_1$ can move. If $r_1$ moves to $v_{-1}$, robots cannot achieve exploration from Lemma 4. Assume that $r_1$ moves to $v_1$, and let $C'$ be the resultant configuration. Then the views of $r_1$ and $r_2$ become symmetric at $C'$. This implies, if they move, the directions of their movements are decided by the schedule. Hence, if $r_1$ or $r_2$ moves, it moves to $v_0$. This makes the configuration go back to $C$ (possibly, $r_1$ and $r_2$ are swapped). If neither $r_1$ nor $r_2$ moves at $C'$, clearly robots cannot achieve exploration.

- Case that $r_2$ can move. If $r_2$ moves to $v_2$, this reduces to the case that $r_2$ occupied $v_2$. Assume that $r_2$ moves to $v_0$, and let $C'$ be the resultant configuration. Then the views of $r_1$ and $r_2$ become symmetric at $C'$. This implies, if they move, the directions of their movements are decided by the schedule. Hence, if $r_1$ or $r_2$ moves, it moves to $v_1$. This makes the configuration go back to $C$ (possibly, $r_1$ and $r_2$ are swapped). If neither $r_1$ nor $r_2$ moves at $C'$, clearly robots cannot achieve exploration.

Next, consider the case that $r_2$ occupies $v_0$ at $C$. In this case, the views of $r_1$ and $r_2$ are symmetric, and hence, if they move, the directions of their movements are decided by the scheduler. If $r_1$ or $r_2$ moves, it moves to $v_1$. This reduces to the case that $r_1$ and $r_2$ occupy nodes $v_0$ and $v_1$ at $C$. If neither $r_1$ nor $r_2$ moves, clearly robots cannot achieve exploration.

From the above discussion, $r_1$ and $r_2$ always stay at $v_{-1}$, $v_0$, $v_1$, or $v_2$. Therefore robots cannot achieve exploration, and consequently the lemma holds.  

\[\text{Lemma 27} \quad \text{Assume } n \geq 9. \text{ Let } C \text{ be a configuration such that the longest distance between two robots is three. In this case, } C \text{ is unsolvable in the SSYNC model.} \]

\[\text{Proof:} \quad \text{Consider three robots } r_1, r_2, \text{ and } r_3. \text{ Without loss of generality, we assume that, at } C, \text{ robots } r_1 \text{ and } r_3 \text{ occupy } v_0 \text{ and } v_3, \text{ respectively, and } r_2 \text{ occupies } v_0 \text{ or } v_1. \]

First, we claim that $r_3$ never moves as long as nodes $v_2$ and $v_4$ are free. Otherwise, by activating $r_3$ several times after $C$, the scheduler can make $r_3$ move. Since the view of $r_3$ is symmetric, the direction of the movement is also decided by the scheduler. Hence $r_3$ moves to $v_4$ and the resultant configuration is unsolvable from Lemma 26. Therefore, the above claim holds.

Consider the case that $r_2$ occupies $v_0$ at $C$. Since $r_3$ cannot move, $r_1$ or $r_2$ should move. Since the views of $r_1$ and $r_2$ are symmetric, the directions of their movements are decided by the scheduler. Hence, $r_1$ or $r_2$ moves to $v_{-1}$, and the resultant configuration is unsolvable from Lemma 26.

Consider the case that $r_2$ occupies $v_1$ at $C$. Since $r_3$ cannot move, $r_1$ or $r_2$ should move. We consider the following three sub-cases.

- If $r_1$ moves to $v_{-1}$, the resultant configuration is unsolvable from Lemma 26.

- If $r_1$ moves to $v_1$ (resp., if $r_2$ moves to $v_0$), $r_1$ and $r_2$ occupy $v_1$ (resp., $v_0$) at the resultant configuration $C'$. Since $r_3$ cannot move at $C'$, $r_1$ or $r_2$ should move after $C'$. Since views of $r_1$ and $r_2$ are symmetric, the directions of their movements are decided by the scheduler. Hence, $r_1$ or $r_2$ moves to $v_0$ (resp., $v_1$), and this makes the configuration go back to $C$ (possibly, $r_1$ and $r_2$ are swapped).
• If \( r_2 \) moves to \( v_2 \), \( r_1 \), \( r_2 \), and \( r_3 \) occupies \( v_0 \), \( v_2 \), \( v_3 \), respectively. Since we do not consider colors here, the resultant configuration is symmetric to \( C \) considered in the current case (i.e., \( r_1, r_2 \), and \( r_3 \) occupies \( v_0, v_1, v_3 \)).

For all sub-cases, the scheduler can make the configuration unsolvable or move robots infinitely among nodes \( v_0, v_1, v_2, \) and \( v_3 \). Hence, robots cannot achieve exploration in this case.

\[ \square \]

Lemma 28 Assume \( n \geq 9 \). Let \( C \) be a configuration such that the longest distance between two robots is two and two robots occupy a single node. In this case, \( C \) is unsolvable in the SSYNC model.

Proof: We consider three robots \( r_1, r_2, \) and \( r_3 \). Without loss of generality, we assume that \( r_1 \) and \( r_2 \) occupy \( v_1 \) and \( r_3 \) occupies \( v_3 \). Since views of all robots are symmetric, the directions of their movements are decided by the scheduler. Hence, if \( r_1 \) or \( r_2 \) moves, it moves to \( v_0 \), and if \( r_3 \) moves, it moves to \( v_4 \). For both cases, the longest distance between two robots becomes three. Therefore, robots cannot achieve exploration from Lemma 27.

\[ \square \]

Lemma 29 Assume \( n \geq 9 \). Let \( C \) be a configuration \( XYX \) for some \( X \in \text{Col} \) and \( Y \in \text{Col} \). In this case, \( C \) is unsolvable in the SSYNC model.

Proof: Without loss of generality, we assume that robots \( r_1, r_2, \) and \( r_3 \) have colors \( X, \) \( Y \), and \( X \) and occupy \( v_1, v_2, \) and \( v_3 \), respectively. We consider the scheduler that repeats the following activation: Activate \( r_1 \) and \( r_3 \) at the same time, and then activate \( r_2 \).

First consider the case that \( r_1 \) and \( r_3 \) move. Since \( r_1 \) and \( r_3 \) have the same views, they make symmetric behaviors. If \( r_1 \) and \( r_3 \) move to \( v_2 \), \( r_1 \) and \( r_3 \) occupy \( v_2 \) and they have the same color. Hence, the resultant configuration is unsolvable from Lemma 25. If \( r_1 \) and \( r_3 \) move to \( v_0 \) and \( v_4 \), respectively, the distance between \( r_1 \) and \( r_3 \) becomes four. Hence, the configuration is unsolvable from Lemma 26.

Next consider the case that \( r_2 \) moves. In this case, \( r_2 \) joins \( r_1 \) or \( r_3 \) and makes a tower. The resultant configuration is unsolvable from Lemma 28.

\[ \square \]

From Lemmas 26 and 27, the longest distance between two robots is at most two. Three robots must be connected from Lemma 28, and they must not form a symmetric configuration from Lemma 29. In addition, robots with the same color must not occupy a single node from Lemma 25. This implies all configurations other than initial configurations described in Theorem 13 are unsolvable. Clearly, if configuration \( C \) is unsolvable in the FSYNC model, \( C \) is also unsolvable in the ASYNC model. Therefore, we have Theorem 14.

\section{C Proof of Theorem 18}

To prove Theorem 18 we show the following lemma.

Lemma 30 Assume that \( \phi = 1, k = 4, \) and \( \text{Col} = \{ G, W \} \). We define a set of configurations \( C_{sol} \) as follows:

\[ C_{sol} = \{ WWGG, WWWW, WWGW, GGWW, GWWW, WGWW, GGGW, GGWG, WGCG, GWGG \} \]
In the SSYNC model, no algorithm exists that solves terminating exploration from any initial configuration in \( C_{sol} \).

From Lemma \[17\] \( C_{sol} \) is a subset of solvable initial configurations in the SSYNC and ASYNC models. Hence, universal algorithms for the SSYNC model must solve terminating exploration from any initial configuration in \( C_{sol} \) in the SSYNC model. In addition, universal algorithms for the ASYNC model must also solve terminating exploration from any initial configuration in \( C_{sol} \) in the SSYNC model. However, Lemma \[30\] implies no such algorithms exist. This implies no universal algorithm exists in the SSYNC and ASYNC model. Therefore, we have Theorem \[18\].

In the rest of this section, we prove Lemma \[30\] by contradiction. Assume that there exists an algorithm \( A \) that solves terminating exploration from any initial configuration in \( C_{sol} \) in the SSYNC model. Let \( d_1 = 10^3 + 1 \), \( d_2 = 3d_1 \), and \( d_3 = (2d_2)^4 \). We consider a ring with \( n > 5d_3 \) nodes and an execution \( E = C_0, C_1, \ldots \) of \( A \) in the ring. Here we assume that, in each instant, the scheduler activates at least one robot that changes its position or its color unless \( A \) terminates. That is, \( C_i \neq C_{i+1} \) holds for every \( i \leq t \), where \( t \) is the instant such that \( A \) terminates in \( C_t \). Note that, since four robots can visit at most four new nodes in each configuration, \( E \) includes at least \( n/4 > (5/4)d_3 \) configurations.

The outline of the proof is as follows. In Lemmas \[31\] and \[32\] we give unsolvable configurations. Then, we prove that the distance between some pair of two robots becomes at least \( d_2 \) (Lemma \[33\]). We also prove that, in Lemmas \[35\] and \[36\] that some three robots must explore the ring by always forming a sub-configuration in

\[
C_{exp} = \left\{ \begin{array}{ll}
WGG, GGW, GWW, WWG, & W \quad G \\
W' W', W' W', & G \quad G' \quad G \quad G
\end{array} \right\}.
\]

Note that these sub-configurations are identical to ones used in the perpetual exploration algorithm in Section 4. The above fact limits a possible set of rules used in algorithm \( A \). Lastly, we prove that, for every possible set of rules, the set of rules makes some configuration in \( C_{sol} \) unsolvable (Lemmas \[38\] and \[39\]). This contradicts to the assumption that \( A \) solves terminating exploration from any initial configuration in \( C_{sol} \).

First, we give examples of unsolvable configurations.

**Lemma 31** Let \( C \) be a configuration that contains a sub-configuration in

\[
C_{sym} = \left\{ \begin{array}{ll}
WGGW, GWWG, WWWW, GGGG, & W \quad W \quad G \quad G \\
W' W', W' W', & G \quad G' \quad G \quad G
\end{array} \right\}.
\]

In this case, \( C \) is unsolvable.

**Proof:** Note that configurations in \( C_{sym} \) are symmetric and the axis of symmetry goes through a link between robots. We consider the scheduler that always activates two symmetric robots. Clearly, all robots keep their symmetry. Hence, to explore the ring, one or two robots must continue to move in one direction (and other one or two robots move in another direction). However, as shown in Lemma \[24\] and Theorem \[12\], one or two robots cannot continue to move in one direction. Therefore, robots cannot achieve exploration from configurations in \( C_{sym} \).

**Lemma 32** Let \( C \) be a configuration such that two robots with the same color occupy a single node. In this case, \( C \) is unsolvable.

25
Proof: Let \( r_1 \) and \( r_2 \) be robots that have the same color and occupy a single node. We consider the scheduler that always activates \( r_1 \) and \( r_2 \) at the same time. Clearly, since \( r_1 \) and \( r_2 \) have the same view, they make the same behaviors. This means \( r_1 \) and \( r_2 \) move as if they were a single robot. Therefore, similarly to Theorem 15, robots cannot achieve exploration from configuration \( C \). \( \square \)

Next, we consider an execution \( E \) of algorithm \( A \), and prove that the distance between some pair of two robots becomes large in \( E \).

Lemma 33 The distance between some pair of two robots becomes at least \( d_2 \) before the \( d_3 = (2d_2)^4 \)-th configuration of execution \( E \).

Proof: For contradiction, assume that the distance between any pair of robots is at most \( d_2 - 1 \) until \( d_3 \)-th configuration of \( E \). For each configuration \( C_i \) (\( 0 \leq i \leq d_3 \)), we define \( C'_i \) as a sub-configuration of \( C_i \) such that \( C'_i \) contains \( d_2 \) nodes (i.e., the length of \( C'_i \) is \( d_2 \)), every robot occupies one of the \( d_2 \) nodes, and some robot occupies the first node of \( C'_i \). Since each robot has one of two colors and occupies one of \( d_2 \) nodes, the number of such sub-configurations is at most \( d_3 \). Hence, for some \( t \) and \( u \) (\( t < u \leq d_3 \)), sub-configurations \( C'_t \) and \( C'_u \) are identical. In the following, we consider two cases.

The first case is that the \( d_2 \) nodes contained in \( C'_t \) and \( C'_u \) are identical. This implies configurations \( C_t \) and \( C_u \) are identical. Since \( A \) is deterministic, robots repeat the behavior from \( C_t \) to \( C_u \) after \( C_u \). Since robots can visit at most \( 4d_3 \) nodes before configuration \( C_t \), they cannot visit remaining nodes after \( C_u \). Thus robots cannot complete exploration.

The second case is that the \( d_2 \) nodes contained in \( C'_t \) and \( C'_u \) are not identical. This means robots change their positions from \( C_t \) to \( C_u \). However, since no robots exist out of the \( d_2 \) nodes, the behaviors of robots depend only on the sub-configuration. Hence, robots repeat the behaviors from \( C_t \) to \( C_u \) after \( C_u \). This implies that robots continue to explore the ring, and thus they cannot terminate.

For both cases, robots cannot achieve terminating exploration. This is a contradiction. \( \square \)

From Lemma 33 the distance between some pair of two robots becomes at least \( d_2 \) before the \( d_3 \)-th configuration. Let \( t_1 \) be the smallest instant such that the distance between some pair of two robots is at least \( d_2 \) at \( C_{t_1} \). Clearly \( t_1 \leq d_3 \) holds. Since four robots can visit at most \( 4(t_1 + 1) \) nodes before configuration \( C_{t_1} \), at least \( d_3 - 4 \) unexplored nodes exist at \( C_{t_1} \). Next, we consider instant \( t_2 = t_1 + d_1 \). Since robots can visit at most \( 4d_1 \) nodes during configurations \( C_{t_1} \) to \( C_{t_2} \), at least \( d_3 - 4 - 4d_1 \) unexplored nodes exist at \( C_{t_2} \).

In Lemmas 34 to 36 we show that there exist three robots that explore the network by forming a sub-configuration in \( C_{exp} \) from \( C_{t_1} \) to \( C_{t_2} \).

Lemma 34 Let \( C \) be a configuration in \( \{C_{t_1}, C_{t_1+1}, \ldots, C_{t_2}\} \). At configuration \( C \), there exist three robots such that the distance among them is at most four.

Proof: Recall that the distance between some pair of two robots is at least \( d_2 \) at configuration \( C_{t_1} \). Since the distance between two robots decreases by at most two after one cycle, the distance between some pair of two robots is at least \( d_2 - 2d_1 \geq d_1 \). We consider four robots \( r_1, r_2, r_3, \) and \( r_4 \). Let \( d \geq d_1 \) be the longest distance between two robots. Without loss of generality, we assume the distance between \( r_1 \) and \( r_4 \) is \( d \), \( r_1 \) occupies \( v_0 \), and \( r_4 \) occupies \( v_d \) at \( C \). We can also assume that
r_2 and r_3 occupy v_i and v_j (0 ≤ i ≤ j ≤ d), respectively, and i ≤ d − j holds. For contradiction, we assume j > 4.

Consider the case that i = 2 or i = 3 holds. Since j > 4 and j ≤ d − 2 holds, we can define an independent territory set \{\{v_{i−1}, v_0\}, \{v_2, v_3\}, \{v_j, v_{j+1}\}, \{v_d, v_{d+1}\}\}. From Lemma 4, robots cannot explore the ring.

Consider the case that i = 0 or i = 1 holds, that is, r_1 and r_2 are connected. First, we show that r_1 and r_2 continue to occupy nodes in \{v_h|−2 ≤ h ≤ 3\} as long as no robot appears on v_{−3} or v_4. In the case of i = 0, if r_1 or r_2 moves, it moves v_1 because the direction of its movement is decided by the scheduler. This reduces to the case of i = 1. Let us consider the case of i = 1. If r_1 moves to v_1 or r_2 moves to v_0, they make a tower. In this case, after r_1 or r_2 moves again, the configuration goes back to the previous one (possibly, r_1 and r_2 are swapped). Hence, we assume that eventually r_1 moves to v_{−1} or r_2 moves to v_2. If r_1 moves to v_{−1}, we can define territories \{v_{−2}, v_{−1}\} for r_1 and \{v_1, v_2\} for r_2. Similarly, if r_2 moves to v_2, we can define territories \{v_{−1}, v_0\} for r_1 and \{v_2, v_3\} for r_2. Hence, as long as no robot appears on v_{−3} or v_4, r_1 and r_2 continue to occupy nodes in \{v_h|−2 ≤ h ≤ 3\}. Let us consider r_3 and r_4 in this case. If r_3 and r_4 are connected at C, we can show the same proposition as r_1 and r_2. That is, as long as no robot appears on v_{d−4} or v_{d+4}, r_3 and r_4 continue to occupy nodes in \{v_h|d − 3 ≤ h' ≤ d + 2\}. Therefore, in this case, the four robots cannot achieve exploration. If r_3 and r_4 are not connected at C, we can define territories for r_3 and r_4 such that r_3 and r_4 do not go out of their territories. Hence, robots cannot achieve exploration.

Now we consider the remaining case i > 3. Clearly r_1 continues to occupy v_0 or v_{−1} as long as no robot appears on v_{−2} or v_1, and r_4 continues to occupy v_d or v_{d+1} as long as no robot appears on v_{d−1} or v_{d+2}. First consider the case that r_2 and r_3 are connected. As described above, r_1 and r_4 cannot go out of their current and neighboring nodes as long as r_2 or r_3 moves toward them. This implies r_2 or r_3 must move toward r_1 or r_4 to achieve exploration. Here we assume the scheduler activates either r_2 or r_3 at the same time. This implies r_2 or r_3 moves against another robot, and the distance between r_2 and r_3 becomes two. At this configuration, we can define an independent territory set and thus robots cannot achieve exploration from Lemma 4. Next consider the case that r_2 and r_3 are disconnected. In this case, we can similarly define an independent territory set and thus, from Lemma 4, robots cannot achieve exploration.

For all cases, robots cannot explore the ring. Hence, j ≤ 4 holds. That is, the distance among robots r_1, r_2, and r_3 is at most four.

From Lemma 34, for each configuration C in \{C_{t_1}, C_{t_1+1}, \ldots, C_{t_2}\}, there exist three robots such that the distance among them is at most four. Note that, since the distance from one of the three robots to another robot is at least d_1, a set of three robots within distance four does not change from C_{t_1} to C_{t_2}. In the following, we define r_1, r_2, and r_3 as three robots that stay within distance four from C_{t_1} to C_{t_2}. Let r_4 be another robot. In the following lemma, we show that r_1, r_2, and r_3 execute a perpetual exploration algorithm from C_{t_1} to C_{t_2}.

**Lemma 35** Consider configuration C* such that r_1, r_2, and r_3 stay on the same node with the same color as configuration C_{t_1} and r_4 does not exist. Then, from configuration C*, robots r_1, r_2, and r_3 achieve a perpetual exploration.

**Proof:** First we consider configurations from C_{t_1} to C_{t_2}. From Lemma 34, three robots r_1, r_2, and r_3 always stay within distance four. Hence, for each configuration C_i for t_1 ≤ i ≤ t_2, we can
define $C'_i$ as a sub-configuration of $C_i$ such that $C'_i$ contains five nodes (i.e., the length of $C'_i$ is five), each of $r_1, r_2$, and $r_3$ occupies one of the five nodes, and some robot occupies the first node of $C'_i$. Since each robot has one of two colors and occupies one of five nodes, the number of such sub-configurations is at most $10^5 < d_1$. Hence, for some $g$ and $h$ ($t_1 \leq g < h \leq t_2$), sub-configurations $C'_g$ and $C'_h$ are identical. We consider two cases.

The first case is that the five nodes contained in $C'_g$ and $C'_h$ are identical. This implies that $r_1, r_2$, and $r_3$ do not change their positions and colors in $C_g$ and $C_h$. Hence, unless $r_1, r_2$, or $r_3$ observes $r_4$, they repeat the behavior from $C_g$ to $C_h$ after $C_h$ and consequently continue to visit the same nodes. On the other hand, $r_4$ is far from the three robots and cannot go out of its territory (i.e., its current and neighboring nodes). This implies that $r_1, r_2$, and $r_3$ do not observe $r_4$, and thus they cannot visit the remaining unexplored nodes.

Let us consider another case, that is, the five nodes contained in $C'_g$ and $C'_h$ are different. This implies $r_1, r_2$, and $r_3$ change their positions from $C_g$ to $C_h$, that is, they move forward or backward in the ring. Clearly, the three robots do not observe $r_4$ from $C_{t_1}$ to $C_h$. This implies that, if $r_4$ does not exist, they repeat the behavior from $C_g$ to $C_h$ and continue to explore the ring. That is, they can achieve a perpetual exploration. Therefore, the lemma holds.

**Lemma 36** From $C_{t_1}$ to $C_{t_2}$, robots $r_1, r_2$, and $r_3$ form a sub-configuration in $C_{exp}$.

**Proof:** For contradiction, assume that $r_1, r_2$, and $r_3$ form a sub-configuration $C_s$ not in $C_{exp}$. From Lemma 35, if $r_4$ does not exist, the three robots achieve perpetual exploration from configuration $C_s$. However, $C_s$ is unsolvable for perpetual exploration from Lemmas 25, 26, 27, 28, and 29. This is a contradiction.

From Lemmas 35 and 36, during configurations $C_{t_1}$ to $C_{t_2}$, robots $r_1, r_2$, and $r_3$ continue to form a sub-configuration in $C_{exp}$ and explore the ring. That is, they must change their colors and positions from a sub-configuration to another sub-configuration in $C_{exp}$. We show transitions among sub-configurations in Fig. 7 and list all rules to realize such transitions as follows. Note that, in Fig. 7, we do not distinguish symmetric sub-configurations such as GWW and WWG.
The next lemma shows $A$ includes neither $R_{15}$ nor $R_{16}$.

**Lemma 37** Algorithm $A$ includes neither $R_{15}$ nor $R_{16}$.

**Proof:** Assume that $A$ includes $R_{15}$ for contradiction. Consider a configuration $WGWW \in C_{sol}$. This configuration transits to $WGGW$ by $R_{15}$. From Lemma 31, configuration $WGGW$ is unsolvable. This is a contradiction. Similarly we can show that $A$ does not include $R_{16}$. \hfill $\Box$

29
Since three robots that form a sub-configuration in $C_{exp}$ have to explore the ring, they repeat state transitions by the above rules. This means $\mathcal{A}$ includes a set of rules that form a (simple) cycle in Fig. 7. We list all such sets of rules as follows. Note that, from Lemma 37, each set includes neither $\mathcal{R}_{15}$ nor $\mathcal{R}_{16}$.

\{
\mathcal{R}_{11}, \{\mathcal{R}_{12}\}
\}
\{
\mathcal{R}_{1}, \{\mathcal{R}_{2}\}, \{\mathcal{R}_{3}, \mathcal{R}_{4}\}, \{\mathcal{R}_{3}, \mathcal{R}_{14}\}, \{\mathcal{R}_{13}, \mathcal{R}_{4}\}, \{\mathcal{R}_{13}, \mathcal{R}_{14}\}
\}
\{
\mathcal{R}_{5}, \mathcal{R}_{6}, \{\mathcal{R}_{7}, \mathcal{R}_{8}\}, \{\mathcal{R}_{9}, \mathcal{R}_{10}\}
\}
\{
\mathcal{R}_{2}, \mathcal{R}_{3}, \mathcal{R}_{7}, \{\mathcal{R}_{2}, \mathcal{R}_{13}, \mathcal{R}_{7}\}, \{\mathcal{R}_{8}, \mathcal{R}_{4}, \mathcal{R}_{1}\}, \{\mathcal{R}_{8}, \mathcal{R}_{14}, \mathcal{R}_{1}\},
\{\mathcal{R}_{10}, \mathcal{R}_{4}, \mathcal{R}_{5}\}, \{\mathcal{R}_{10}, \mathcal{R}_{14}, \mathcal{R}_{5}\}, \{\mathcal{R}_{6}, \mathcal{R}_{3}, \mathcal{R}_{9}\}, \{\mathcal{R}_{6}, \mathcal{R}_{13}, \mathcal{R}_{9}\},
\{\mathcal{R}_{2}, \mathcal{R}_{5}, \mathcal{R}_{10}, \mathcal{R}_{7}\}, \{\mathcal{R}_{8}, \mathcal{R}_{9}, \mathcal{R}_{6}, \mathcal{R}_{1}\}
\}

However, $\mathcal{A}$ cannot include some of the above sets. For example, let us consider a set \{\mathcal{R}_{2}, \mathcal{R}_{5}, \mathcal{R}_{10}, \mathcal{R}_{7}\}. Starting from sub-configuration GWW, the three robots change their sub-configuration as follows.

\[GWW \xrightarrow{\mathcal{R}_{2}} \ G \ G \ \emptyset \xrightarrow{\mathcal{R}_{5}} \ G \ G \emptyset \xrightarrow{\mathcal{R}_{10}} \emptyset \ W \ W \xrightarrow{\mathcal{R}_{7}} GWW\]

Note that these sub-configurations include the same nodes. This implies that, when the robots form GWW again, they do not change their positions. Hence, if $\mathcal{A}$ includes this set, the three robots cannot explore the ring. On the other hand, let us consider a set \{\mathcal{R}_{2}, \mathcal{R}_{3}, \mathcal{R}_{7}\}. Starting from sub-configuration GWW, the three robots change their sub-configuration as follows.

\[\emptyset GWW \xrightarrow{\mathcal{R}_{2}} \emptyset G \ G \emptyset \xrightarrow{\mathcal{R}_{3}} \emptyset G \ G \emptyset \xrightarrow{\mathcal{R}_{7}} GWW\]

In this case, when the robots form GWW again, they change their positions. Hence, by this set of rules, the three robots can explore the ring. By considering all sets of rules, the following four sets of rules allow robots to explore the ring:

\{\mathcal{R}_{2}, \mathcal{R}_{3}, \mathcal{R}_{7}\}, \{\mathcal{R}_{8}, \mathcal{R}_{4}, \mathcal{R}_{1}\}, \{\mathcal{R}_{10}, \mathcal{R}_{4}, \mathcal{R}_{5}\}, \{\mathcal{R}_{6}, \mathcal{R}_{3}, \mathcal{R}_{9}\}.

That is, $\mathcal{A}$ includes at least one of the above four sets of rules. However, in the following lemmas, each set of rules makes some configuration in $C_{sol}$ unsolvable.

**Lemma 38** If $\mathcal{A}$ includes \{\mathcal{R}_{2}, \mathcal{R}_{3}, \mathcal{R}_{7}\} or \{\mathcal{R}_{10}, \mathcal{R}_{4}, \mathcal{R}_{5}\}, $\mathcal{A}$ cannot solve terminating exploration from some configuration in $C_{sol}$.

**Proof:** First, we consider the case that $\mathcal{A}$ includes \{\mathcal{R}_{2}, \mathcal{R}_{3}, \mathcal{R}_{7}\}. Let us consider a configuration WGGW $\in C_{sol}$. From rules $\mathcal{R}_{2}$ and $\mathcal{R}_{3}$, the configuration changes as follows.

\[WGGW \xrightarrow{\mathcal{R}_{2}} W G G \xrightarrow{\mathcal{R}_{3}} W G W\]

Let $C_{a}$ be the last configuration. Since robots cannot move from $C_{a}$ by rules $\mathcal{R}_{2}, \mathcal{R}_{3}, \mathcal{R}_{7}$, $\mathcal{A}$ includes some other rules to move robots from $C_{a}$. We consider all possibilities in the following.

- $\mathcal{A}$ includes $\mathcal{R}_{1}^{i}$: $W$ $\rightarrow$ $G$, $\leftarrow$ $\lor$ $\rightarrow$. In this case, robots change their states as follows.

\[W G W \xrightarrow{\mathcal{R}_{1}^{i}} G \ G \xrightarrow{\mathcal{R}_{2}} G G \xrightarrow{\mathcal{R}_{2}} W W\]

From Lemma 31, the last configuration is unsolvable.
• \( A \) includes \( R'_2 : W (G) W :: W, \leftarrow \lor \rightarrow \). In this case, robots change their states as follows.

\[
\begin{array}{c}
W \\
W G W \xrightarrow{R'_2} W W W
\end{array}
\]

From Lemma 32, the last configuration is unsolvable.

• \( A \) includes \( R'_3 : W (G) W :: W, \bot \). In this case, robots change their states as follows.

\[
\begin{array}{c}
W \\
W G W \xrightarrow{R'_3} W W W
\end{array}
\]

From Lemma 32, the last configuration is unsolvable.

• \( A \) includes \( R'_4 : W (W) W :: G, \leftarrow \lor \rightarrow \). In this case, robots change their states as follows.

\[
\begin{array}{c}
W \\
W G W \xrightarrow{R'_4} W G G \xrightarrow{R_3} W G W
\end{array}
\]

That is, the robots repeatedly change their states while keeping their positions. This implies that they cannot achieve exploration.

• \( A \) includes \( R'_5 : W (W) W :: W, \leftarrow \lor \rightarrow \). In this case, robots change their states as follows.

\[
\begin{array}{c}
W \\
W G W \xrightarrow{R'_5} W G W
\end{array}
\]

From Lemma 32, the last configuration is unsolvable.

• \( A \) includes \( R'_6 : W (W) W :: G, \bot \). In this case, robots change their states as follows.

\[
\begin{array}{c}
W \\
W G W \xrightarrow{R'_6} W G W
\end{array}
\]

From Lemma 32, the last configuration is unsolvable.

• \( A \) includes \( R'_7 : W (W) G :: W, \rightarrow \). In this case, robots change their states as follows.

\[
\begin{array}{c}
W \\
W G W \xrightarrow{R'_7} W G W
\end{array}
\]

From Lemma 32, the last configuration is unsolvable.
• $A$ includes $R'_8 : \emptyset (W) G :: G, \rightarrow$. In this case, robots change their states as follows.

$$
\begin{align*}
W & \xrightarrow{R'_8} W \\
W G W & \xrightarrow{R'_8} G W
\end{align*}
$$

From Lemma 32, the last configuration is unsolvable.

• $A$ includes $R'_9 : \emptyset (W) G :: W, \leftarrow$. Recall that, from $C_{t_1}$ to $C_{t_2}$, three robots change their states by rules $R_2, R_3,$ and $R_7$. Let us consider a configuration such that three robots form $G W W$. In the next configuration, three robots can form $G W \emptyset W$ by rule $R'_9$. This contradicts to Lemma 36.

• $A$ includes $R'_{10} : \emptyset (W) G :: G, \leftarrow$. Let us consider a configuration such that three robots form $G W W$ to explore the ring. In the next configuration, three robots can form $G W \emptyset G$ by rule $R'_{10}$. This contradicts to Lemma 36.

• $A$ includes $R'_{11} : \emptyset (W) G :: G, \perp$. Let us consider a configuration such that three robots form $G W W$ to explore the ring. After that, the three robots change their states as follows.

$$
\begin{align*}
G W W & \xrightarrow{R'_{11}} G W G \xrightarrow{R_3} W G W \xrightarrow{R'_{11}} W G G \xrightarrow{R_3} G W W
\end{align*}
$$

That is, the robots repeatedly change their states while keeping their positions. This implies that they cannot explore the ring, and thus this contradicts to Lemma 35.

From the above discussion, for any rule included in $A$, some configuration in $C_{sol}$ becomes unsolvable or exploration of three robots becomes impossible. Therefore, if $A$ includes $\{R_2, R_3, R_7\}$, $A$ cannot solve terminating exploration from some configuration in $C_{sol}$.

We can prove the case of $\{R_{10}, R_4, R_5\}$ similarly because, in rules $R_{10}, R_4,$ and $R_5$, the roles of colors are just swapped from rules $R_2, R_3,$ and $R_7$.

\textbf{Lemma 39} \ If $A$ includes $\{R_8, R_4, R_1\}$ or $\{R_6, R_3, R_9\}$, $A$ cannot solve terminating exploration from some configuration in $C_{sol}$

\textit{Proof:} First, we consider the case that $A$ includes $\{R_8, R_4, R_1\}$. Let us consider configuration $GGWW \in C_{sol}$. Assume that robots $r_1, r_2, r_3,$ and $r_4$ form the sub-configuration $GGWW$ in this order. Since robots cannot move from this configuration by rules $R_8, R_4,$ and $R_1$, $A$ includes some other rules to move robots from the configuration. We consider all possibilities in the following.

First consider the case that $r_1$ can move. That is, $A$ includes a rule such that the guard is $\emptyset (G) G$.
• A includes $R''_1 : \emptyset(G)G :: G, \rightarrow$. In this case, robots change configuration $GGWW \in C_{sol}$ to configuration $GWGW \emptyset W$, which is unsolvable from Lemma 32.

• A includes $R''_2 : \emptyset(G)W :: W, \rightarrow$. In this case, from rules $R''_2$ and $R_8$, robots change configuration $GWGG \in C_{sol}$ to configuration $G \emptyset W W$, which is unsolvable from Lemma 31.

• A includes $R''_3 : \emptyset(G)G :: G, \leftarrow$. In this case, from rules $R''_3$ and $R_8$, robots change their states from configuration $GWGG \in C_{sol}$ as follows.

$$\text{GWGG} \xrightarrow{R''_3} \text{GWG} \emptyset \xrightarrow{R_8} G \emptyset \emptyset \emptyset W$$

The last configuration is unsolvable from Lemma 32.

• A includes $R''_4 : \emptyset(G)G :: W, \leftarrow$. In this case, from rules $R''_4$ and $R_8$, robots change their states from configuration $GWGG \in C_{sol}$ as follows.

$$\text{GWGG} \xrightarrow{R''_4} \text{GWG} \emptyset \xrightarrow{R_8} G \emptyset \emptyset \emptyset W$$

The last configuration is unsolvable from Lemma 32.

• A includes $R''_5 : \emptyset(G)G :: G, \perp$. In this case, robots change configuration $GGGW \in C_{sol}$ to configuration $WGGW$, which is unsolvable from Lemma 31.

Next consider the case that $r_2$ can move. That is, $A$ includes a rule such that the guard is $G(G)W$.

• A includes $R''_6 : G(G)W :: G, \rightarrow$. In this case, from rules $R''_6$ and $R_8$, robots change configuration $GGWG \in C_{sol}$ to configuration $GWGW \emptyset G$, which is unsolvable from Lemma 32.

• A includes $R''_7 : G(G)W :: W, \rightarrow$. In this case, robots change configuration $GGWG \in C_{sol}$ to configuration $GWGW \emptyset G$, which is unsolvable from Lemma 32.

• A includes $R''_8 : G(G)W :: G, \leftarrow$. In this case, robots change configuration $GGWG \in C_{sol}$ to configuration $GWGW \emptyset G$, which is unsolvable from Lemma 32.

• A includes $R''_9 : G(G)W :: W, \leftarrow$. In this case, from rules $R''_9$ and $R_8$, robots change configuration $GGWG \in C_{sol}$ to configuration $GWGW \emptyset G$. Since two robots with the same color have the same view, they make the same behaviors if they are activated at the same time. In addition, the view is symmetric, the direction of the movement is decided by the scheduler. Hence,
when some robot moves from the configuration, two robots move to the middle node of the two towers and create a tower. Since they have the same color, the configuration is unsolvable from Lemma 32.

- \( \mathcal{A} \) includes \( \mathcal{R}''_{10}: G(W) : W, \bot \). In this case, robots change configuration \( GWG \in C_{sol} \) to configuration \( GWWG \), which is unsolvable from Lemma 31.

Next consider the case that \( r_3 \) can move. That is, \( \mathcal{A} \) includes a rule such that the guard is \( G(W) \). In the first four cases, we consider configurations from \( C_{t_1} \) to \( C_{t_2} \). During these configurations, three robots change their states by rules \( \mathcal{R}_8, \mathcal{R}_4, \) and \( \mathcal{R}_1 \). In particular, we consider a sub-configuration \( GWW \) in \( C_{exp} \).

- \( \mathcal{A} \) includes \( \mathcal{R}''_{16}: G(W) : G, \rightarrow \). From a sub-configuration \( GWW \), the robots change their states to \( G \emptyset W \). This sub-configuration is not in \( C_{exp} \), which contradicts to Lemma 36.

- \( \mathcal{A} \) includes \( \mathcal{R}''_{17}: G(W) : W, \rightarrow \). From a sub-configuration \( GWW \), the robots change their states to \( W G \emptyset W \). This sub-configuration is not in \( C_{exp} \), which contradicts to Lemma 36.

Lastly consider the case that \( r_4 \) can move. That is, \( \mathcal{A} \) includes a rule such that the guard is \( \emptyset(W) \). Similarly to the previous case, we consider a sub-configuration \( GWW \) in \( C_{exp} \).

- \( \mathcal{A} \) includes \( \mathcal{R}''_{16}: \emptyset(W) : G, \rightarrow \). From rules \( \mathcal{R}''_{16} \) and \( \mathcal{R}_1 \), robots change their states from a sub-configuration \( GWW \) as follows.

\[
GWW \xrightarrow{\mathcal{R}''_{16}} W G \emptyset \xrightarrow{\mathcal{R}_1} GWW
\]

That is, the robots repeatedly change their states while keeping their positions. This implies that they cannot explore the ring, and thus this contradicts to Lemma 35.

- \( \mathcal{A} \) includes \( \mathcal{R}''_{17}: \emptyset(W) : W, \rightarrow \). From a sub-configuration \( GWW \), the robots change their states to \( W G \emptyset W \). This sub-configuration is not in \( C_{exp} \), which contradicts to Lemma 36.

- \( \mathcal{A} \) includes \( \mathcal{R}''_{18}: \emptyset(W) : G, \leftarrow \). From a sub-configuration \( GWW \), the robots change their states to \( GWG \emptyset \). This sub-configuration is not in \( C_{exp} \), which contradicts to Lemma 36.
• \( A \) includes \( R''_{19} : \emptyset(W)W :: W, \leftarrow \). From a sub-configuration \( GWW \), the robots change their states to \( GW\emptyset W \). This sub-configuration is not in \( C_{\text{exp}} \), which contradicts to Lemma 36.

• \( A \) includes \( R''_{20} : \emptyset(W)W :: G, \perp \). From a sub-configuration \( GWW \), the robots change their states to \( GWG \). This sub-configuration is not in \( C_{\text{exp}} \), which contradicts to Lemma 36.

From the above discussion, for any rule included in \( A \), some configuration in \( C_{\text{sol}} \) becomes unsolvable or exploration of three robots becomes impossible. Therefore, if \( A \) includes \( \{ R_8, R_4, R_1 \} \), \( A \) cannot solve terminating exploration from some configuration in \( C_{\text{sol}} \).

We can prove the case of \( \{ R_6, R_3, R_9 \} \) similarly because, in rules \( R_6, R_3, \) and \( R_9 \), the roles of colors are just swapped from rules \( R_8, R_4, \) and \( R_1 \).  

From Lemmas 38 and 39, \( A \) cannot solve terminating exploration from some configuration in \( C_{\text{sol}} \). Therefore, we have Lemma 30. As described above, this implies Theorem 18.