On the complexity of the guarding game

R. Šámal [⋆] and T. Valla [⋆⋆]

Charles University, Faculty of Mathematics and Physics,
Institute for Theoretical Computer Science (ITI)
Malostranské nám. 2/25, 118 00, Prague, Czech Republic
{samal,valla}@kam.mff.cuni.cz,

Abstract. The guarding game is a game in which several cops try to guard a region in a (directed or undirected) graph against a robber. The robber and the cops are placed on the vertices of the graph; they take turns in moving to adjacent vertices (or staying), cops inside the guarded region, the robber on the remaining vertices (the robber-region). The goal of the robber is to enter the guarded region at a vertex with no cop on it. The problem is to determine whether for a given graph and given number of cops the cops are able to prevent the robber from entering the guarded region. Fomin et al. [Fomin, Golovach, Hall, Mihalák, Vicari, Widmayer: How to Guard a Graph? Algorithmica, DOI: 10.1007/s00453-009-9382-4] proved that the problem is NP-complete when the robber-region is restricted to a tree. Further they prove that is it PSPACE-complete when the robber-region is restricted to a directed acyclic graph, and they ask about the problem complexity for arbitrary graphs. In this paper we prove that for arbitrary graphs (directed or undirected) the problem is E-complete. [1]

Key words: pursuit game, cops and robber game, graph guarding game, computational complexity, E-completeness

1 Introduction and motivation

The *guarding game* (G, V_C, c), introduced by Fomin et al. [2], is played on a graph $G = (V, E)$ (or directed graph $\overrightarrow{G} = (V, E)$) by two players, the *cop-player* and the *robber-player*, each having his pawns (c cops and one robber, respectively) on V. There is a protected region (also called cop-region) $V_C \subset V$. The remaining region $V \setminus V_C$ is called robber-region and denoted V_R. The robber aims to enter V_C by a move to a vertex of V_C with no cop on it. The cops try to prevent this. The game is played in alternating turns. In the first turn the robber-player places the robber on some vertex of V_R. In the second turn the cop-player places

[⋆] Partially supported by grant GA ČR P201/10/P337.

[⋆⋆] Supported by the GAUK Project 66010 of Charles University Prague and supported by ITI, Charles University Prague, under grant 1M0021620808.

[1] The conference version of this paper has appeared in Proceedings of IWOCA 2011, Springer-Verlag LNCS [1].

his c cops on vertices of V_C (more cops can share one vertex). In each subsequent turn the respective player can move each of his pawns to a neighbouring vertex of the pawn's position (or leave it where it is). However, the cops can move only inside V_C and the robber can move only on vertices with no cops. At any time of the game both players know the positions of all pawns. The robber-player wins if he is able to move the robber to some vertex of V_C in a finite number of steps. The cop-player wins if the cop-player can prevent the robber-player from placing the robber on a vertex in V_C indefinitely. Note that after exponentially many (in the size of the graph G) turns the positions has to repeat and obviously if the robber can win, he can win in less than $2|V|^{(c+1)}$ turns, Note that $2|V|^{c+1}$ is the upper bound on the number of all possible positions of the robber and all cops, so after that many turns the position has to repeat. Thus, if the robber can win, he can win in less than $2|V|^{c+1}$ turns. Consequently, we may define the robber to lose if he does not win in $2|V|^{c+1}$ turns.

For a given graph G and guarded region V_C, the task is to find the minimum number c such that cop-player wins. Note that this problem is polynomially equivalent with the problem of determining the outcome of the game for a fixed number c of cops.

The guarding game is a member of a big class called the pursuit-evasion games, see, e.g., Alspach [5] for introduction and survey. The discrete version of pursuit-evasion games on graphs is called the Cops-and-Robber game. This game was first defined for one cop by Winkler and Nowakowski [6] and by Quilliot [7]. Aigner and Fromme [8] initiated the study of the problem with several cops. The minimum number of cops required to capture the robber is called the cop number of the graph. In this setting, the Cops-and-Robber game can be viewed as a special case of search games played on graphs. Therefore, the guarding game is a natural variant of the original Cops-and-Robber game. The complexity of the decision problem related to the Cops-and-Robbers game was studied by Goldstein and Reingold [12]. They have shown that if the number of cops is not fixed and if either the graph is directed or initial positions are given, then the problem is E-complete. Another interesting variant is the "fast robber" game, which is studied in Fomin et al. [13]. See the annotated bibliography [11] for reference on further topics.

A different well-studied problem, the *Eternal Domination* problem (also known as *Eternal Security*) is strongly related to the guarding game. The objective in the Eternal Domination is to place the minimum number of guards on the vertices of a graph G such that the guards can protect the vertices of G from an infinite sequence of attacks. In response to an attack of an unguarded vertex v, at least one guard must move to v and the other guards can either stay put, or move to adjacent vertices. The Eternal Domination problem is a special case of the guarding game. This can be seen as follows. Let G be a graph on n vertices and we construct a graph H from G by adding a clique K_n on n vertices and connecting the clique and G by n edges which form a perfect matching. The cop-region is $V(G)$ and the robber-region is $V(K_n)$. Now G has an eternal dominating set of size k if and only if k cops can guard $V(G)$. Eternal Domination and its variant have been considered for example in [16–23].

In our paper we focus on the complexity issues of the following decision problem: Given the guarding game $\mathcal{G} = (G, V_C, c)$, who has the winning strategy?

Let us define the computational problem precisely. The *directed guarding decision problem* is, given a guarding game $(\overrightarrow{G}, V_C, c)$ where $\overrightarrow{G}$ is a directed graph, to decide whether it is a cop-win game or a robber-win game. Analogously, we define the *undirected guarding decision problem* with the difference that the underlying graph G is undirected. The *guarding problem* is, given a directed or undirected graph G and a cop-region $V_C \subseteq V(G)$, to compute the minimum number c such that the (G, V_C, c) is a cop-win.

The directed guarding decision problem was introduced and studied by Fomin et al. [2]. The computational complexity of the problem depends heavily on the chosen restrictions on the graph G. In particular, in [2] the authors show that if the robber's region is only a path, then the problem can be solved in polynomial time, and when the robber moves in a tree (or even in a star), then the problem is NP-complete. Furthermore, if the robber is moving in a directed acyclic graph, the problem becomes PSPACE-complete. Later Fomin, Golovach and Lokshtanov [14] studied the *reverse guarding game* with the same rules as in the guarding game, except that the cop-player plays first. They proved in [14] that the related decision problem is PSPACE-hard on undirected graphs. Nagamochi [9] has also shown that that the problem is NP-complete even if V_R induces a 3-star and that the problem is polynomially solvable if V_R induces a cycle. Also, Thirumala Reddy, Sai Krishna and Pandu Rangan have proved [10] that if the robber-region is an arbitrary undirected graph, then the decision problem is PSPACE-hard.

Fomin et al. [2] asked the following question.

Question 1. Is the guarding decision problem for general graphs PSPACE-complete?

Let us consider the class $\mathrm{E} = \mathrm{DTIME}(2^{O(n)})$ of languages recognisable by a deterministic Turing machine in time $2^{O(n)}$. We consider log-space reductions, this means that the reducing Turing machine is log-space bounded. In pursuit of Question 1 we prove the following result.

Theorem 1. *The directed guarding decision problem is E-complete under log-space reductions.*

We would like to point out the fact that we can prove Theorem 1 without prescribing the starting positions of players. Immediately, we get the following corollary.

Corollary 1. *The guarding problem is E-complete under log-space reductions.*

Let us explain here the relevance of Theorem 1 to Question 1. Very little is known how the class E is related to PSPACE. It is only known [4] that $\mathrm{E} \neq \mathrm{PSPACE}$. The following corollary shows that positive answer to Question 1 would give a relation between these two complexity classes. This gives unexpected and strong incentive to find positive answer to Question 1. (On the other hand, to the skeptics among us, it may also indicate that negative answer is more likely.)

Corollary 2. *If the conjecture of Fomin et al. is true, then E $\subseteq$ PSPACE.*

Proof. Suppose the guarding problem is PSPACE-complete. Let $L \in$ E. Then (by Theorem 1) an instance of L can be reduced by a log-space reduction to an instance of the guarding game, which we suppose to be in PSPACE. Consequently, $L \in$ PSPACE.

We also prove Theorem 2, a theorem similar to Theorem 1 for general undirected graphs. We define the *guarding game with prescribed starting positions* $\mathcal{G} = (G, V_C, c, S, r)$, where $S : \{1, \ldots, c\} \to V_C$ is the initial placement of cops and $r \in V_R$ is the initial placement of robber. The *undirected guarding decision problem with prescribed starting positions* is, given a guarding game with prescribed starting positions (G, V_C, c, S, r) where G is an undirected graph, to decide whether it is a cop-win game or a robber-win game. The *directed guarding decision problem with prescribed starting positions* is defined analogously.

Theorem 2. *The undirected guarding decision problem with prescribed starting positions is E-complete under log-space reductions.*

Here, we would like to point out the fact that with the exception of the result in [14], all known hardness results for cops and robbers, or pursuit evasion games are for the directed graph variants of the games [2, 12]. For example, the classical Cop and Robbers game was shown to be PSPACE-hard on directed graphs by Goldstein and Reingold in 1995 [12] while for *undirected* graphs, even an NP-hardness result was not known until recently by Fomin, Golovach and Kratochvíl [15].

Let us also consider the guarding game $\mathcal{G}^R = (\overrightarrow{G}, V_C, c)^R$, where the two initial turns are different: In the first turn, the cop-player places all cops on vertices of V_C, and in the second turn, the robber-player places the robber on some vertex of V_R. Then the game proceeds as usual, starting with the cop-player. In some sense, this game looks like being harder for the cop-player, because the robber during his initial placement has better chance to endanger the cop-region. Analogously with the definition of directed guarding decision problem we define the *reverse directed guarding decision problem.*

Fomin, Golovach and Lokshtanov [14] proved that the reverse undirected guarding problem is PSPACE-hard. We show the following theorem for the directed case.

Theorem 3. *The reverse directed guarding decision problem is E-complete under log-space reductions.*

For the original Cops-and-Robber game, Goldstein and Reingold [12] have proved that if the number c of cops is not fixed and if either the graph is directed or initial positions are given, then the related decision problem is E-complete.

In a sense, we show analogous result for the guarding game as Goldstein and Reingold [12] have shown for the original Cops-and-Robber game. Similarly to Goldstein and Reingold, we can prove the complexity of the undirected guarding decision problem only when having prescribed the initial positions of players. Dealing with this issue seems to be a nontrivial task in this family of games.

2 The directed case

In order to prove E-completeness of the directed guarding decision problem, we first note that the problem is in E.

Lemma 1. *The guarding decision problem (directed or undirected) is in E.*

Proof. We need to show that there is an algorithm deciding the outcome of a given guarding game $\mathcal{G} = (G, V_C, c)$ in $2^{O(n)}$ time, where n is the size of the input $\mathcal{G}$ in some encoding. Consider the directed graph H of all configurations of the game $\mathcal{G}$ – the vertices of H are all possible legal positions of all cops and the robber, together with the information whose turn it is. There is also a starting vertex s representing the empty board and the vertices $r_1, \ldots, r_{|V_R|}$ representing every possible initial placement of the robber with still no cops placed. More precisely,

$$V(H) = \{(C, r, t);\ C\,\{1, \ldots, c\} \to V_C,\ r \in V(G),\ t \in \{0, 1\}\} \cup \{s, r_1, \ldots, r_{|V_R|}\}.$$

Here $t = 0$ denotes the robber is on move, $t = 1$ denotes cops are on move, C is the position of cops and r is the position of the robber. There are edges from s to every vertex r_i and for every r_i there are edges to every possible initial subsequent placement of cops. The edge (k_1, k_2) belongs to $E(H)$ if and only if k_1 is cop turn and k_2 is robber turn (or vice versa) and the pawns of k_1 can be legally moved into pawn positions of k_2.

Using the following backwards-labelling algorithm we can decide the outcome of every position in polynomial time in the size of the graph H. Let us denote the robber-winning configurations by W_R.

1. Construct the graph H.
2. Initially set W_R to be all vertices that are a win for the robber-player, i.e. positions where the robber stands on some $v \in V_C$ and there is no cop on v.
3. Add to W_R all vertices v where robber is on turn and there is an edge $(v, w) \in E(H)$ and $w \in W_R$.
4. Add to W_R all vertices v where cop is on turn and for every edge $(v, w) \in E(H)$ the vertex $w \in W_R$.
5. Repeat $|V(H)|$-times the steps 3 and 4.
6. If $s \in W_R$ the game $\mathcal{G}$ is robber-win, otherwise the game $\mathcal{G}$ is cop-win.

Note that each step can be computed in time polynomial in the size of H. It remains to show that the size of H is $2^{O(n)}$. As mentioned in the introduction, the simplest upper bound on $|V(H)|$ is $2|V(G)|^{c+1}$, which is unfortunately super-exponential in n if c is close to n. To find a better upper bound, we use the fact that the cops are mutually indistinguishable. There are at most $|V(G)|$ positions of the robber. Counting the number of all positions of the cops is the classical problem of putting c indistinguishable balls into $|V_C|$ baskets. Then, taking into account also whose turn it is and the number of vertices r_i, we get that $|V(H)|$ is bounded by

$$|V(H)| \leq 4|V(G)|\binom{|V_C|+c-1}{c} \leq 4n\binom{n+c-1}{c} \leq 4n2^{n+c-1} = 2^{O(n)}.$$

Thus the total size of H is $2^{O(n)}$ as well. □

Let us first study the problem after the second move, where both players have already placed their pawns. We reduce the directed guarding decision problem with prescribed starting positions from the following formula-satisfying game $\mathcal{F}$.

A position in $\mathcal{F}$ is a 4-tuple $(\tau, F_R(C,R), F_C(C,R), \alpha)$ where $\tau \in \{1,2\}$, F_R and F_C are formulas in 12-DNF both defined on set of variables $C \cup R$, where C and R are disjoint and α is an initial $(C \cup R)$-assignment. The symbol τ serves only to differentiate the positions where the first or the second player is on move. Player I (II) moves by changing the values assigned to at most one variable in R (C); either player may pass since changing no variable amounts to a "pass". Player I (II) wins if the formula F_R (F_C) is true after some move of player I (II). More precisely, player I can move from $(1, F_R, F_C, \alpha)$ to $(2, F_R, F_C, \alpha')$ in one move if and only if α' differs from α in the assignment given to at most one variable in R and F_C is false under the assignment α; the moves of player II are defined symmetrically.

According to Stockmeyer and Chandra [3], the set of winning positions of player I in the game $\mathcal{F}$ is an E-complete language under log-space reduction.

Let us first informally sketch the reduction from $\mathcal{F}$ to $\mathcal{G}$, i.e., simulating $\mathcal{F}$ by an equivalent guarding game $\mathcal{G}$. The setting of variables is represented by positions of certain cops so that only one of these cops may move at a time (otherwise cop-player loses the game). The variables (or more precisely the corresponding cops) of C are under control of cop-player. However, in spite of being represented by cops, the variables of R are under control of the robber-player by a gadget in the graph $\overrightarrow{G}$, which allows him to force any setting of cops representing R.

When describing the features of various gadgets, we will often use the term *normal scenario*. By normal scenario S of certain gadget (or even the whole game) we mean a flow of the game that imitates the formula game $\mathcal{F}$. The graph G will be constructed in such a way that if the player (both cop-player and robber-player) does not exactly follow the normal scenario S, he loses the game in a few moves.

There are four cyclically repeating phases of the game, determined by the current position of the robber. The normal scenario is that robber cyclically goes through the following phases marked by four special vertices and in different phases he can enter certain gadgets.

1. "Robber Move" (RM): In this step the robber can enter the Manipulator gadget, allowing him to force setting of at most one variable in R.
2. "Robber Test" (RT): In this step the robber may pass through the *Robber Gate* into the protected region V_C, provided that the formula F_R is satisfied under the current setting of variables.
3. "Cop Move" (CM): In this step (and only in this step) one (and at most one) variable cell V_x for $x \in C$ is allowed to change its value. This is realized by a gadget called *Commander*.
4. "Cop Test" (CT): In this step, if the formula F_C is satisfied under the current setting of variables, the cops are able to block the entrance to the protected region forever (by temporarily leaving the *Cop Gate* gadget unguarded and sending a cop to block the entrance to V_C provided by the Robber Gate gadgets).

See Fig. 1 for the overview of the construction.

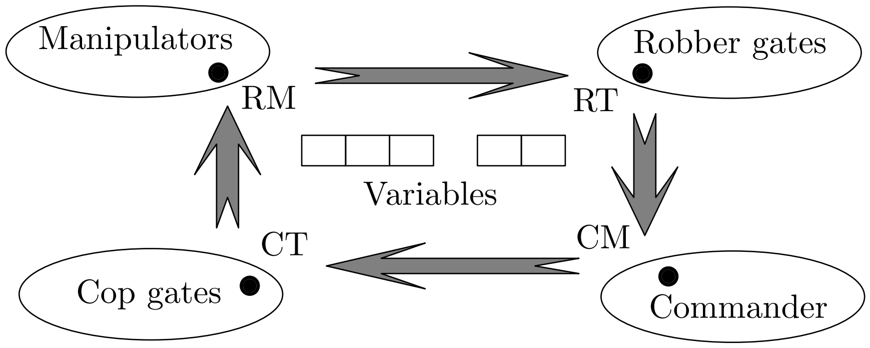

Fig. 1. The sketch of the construction

2.1 The variable cells

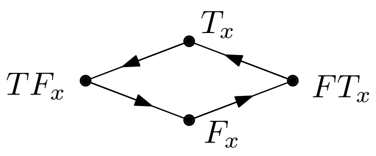

Fig. 2. Variable cell V_x

For every variable $x \in C \cup R$ we introduce a *variable cell* V_x, which is a directed cycle (T_x, TF_x, F_x, FT_x) (see Fig. 2). There is one cop (*variable cop*) located in every V_x and the position of the cop on vertices T_x, F_x represents the boolean values true and false, respectively. The prescribed starting position of the variable cop is T_x if $\alpha(x)$ is true, and F_x otherwise. All the vertices of V_x belong to V_C.

The cells are organised into blocks C and R. The block C is under control of cop-player via the Commander gadget, the block R is represented by cops as well, however, there are the *Manipulator* gadgets allowing the robber-player to force any setting of variables in R, by changing at most one variable in his turn.

Every variable cell V_y, $y \in R$ has assigned the Manipulator gadget M_y. Manipulator M_y consists of directed paths (RM, T'_y, T''_y, T_y) and (RM, F'_y, F''_y, F_y) and edges (T'_y, RT) and (F'_y, RT) (see Fig. 3).

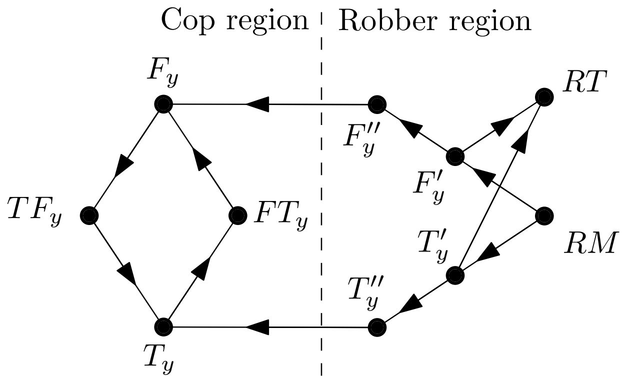

Fig. 3. The Manipulator gadget M_y

The vertices $\{T'_y, F'_y, T''_y, F''_y, RM, RT\} \subset V_R$, the rest belongs to V_C.

Lemma 2. *Let us consider variable cell V_y, $y \in R$, and the corresponding Manipulator M_y. Let the robber be at the vertex RM, let the cop be either on T_y or F_y and suppose no other cop can access any vertex of M_y in less than three moves. Then the normal scenario is following: By entering the vertex T'_y (F'_y), the robber forces the cop to move towards the vertex T_y (F_y). Robber then has to enter the vertex RT.*

Proof. If the cop refuses to move, the robber advances to T''_y or F''_y and easily reaches V_C before the cop can block him. On the other hand, if the robber moves to T''_y or F''_y even though the cop moved towards the opposite vertex, then cop finishes his movement to the opposite vertex and robber cannot move anymore. □

Note that this is not enough to ensure that the variable cop really reaches the opposite vertex and that only one variable cop from variable cells can move. We deal with this issue later.

When changing variables of C, we have to make sure that at most one variable is changed at a time. We ensure that by the gadget Commander (see Fig. 4), connected to every V_x, $x \in C$. It consists of the vertices $\{f_x, g_x, h_x;\ x \in C\} \cup \{HQ\}$ and the edges

$$\{(HQ, h_x), (h_x, HQ), (h_x, f_x), (T_x, f_x), (F_x, f_x), (g_x, f_x), (CM, g_x);\ x \in C\}.$$

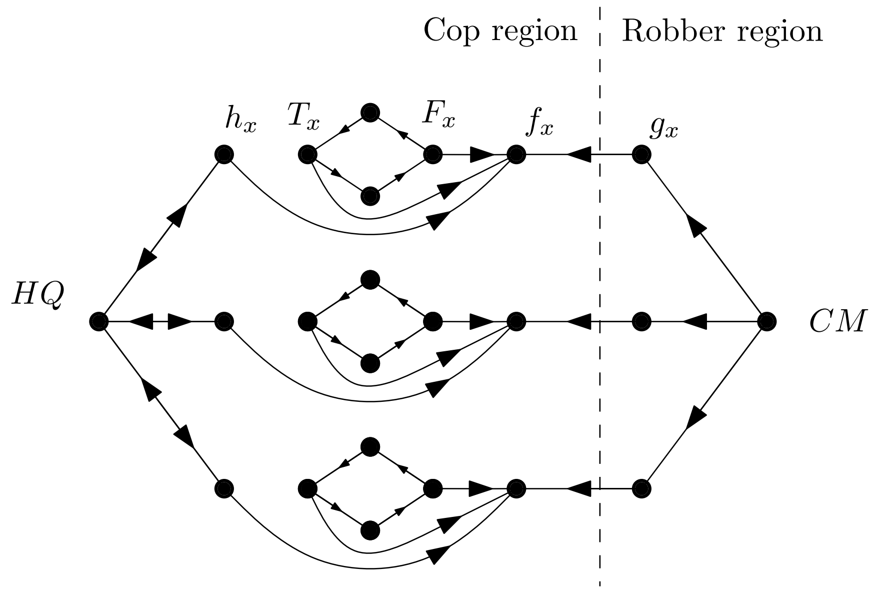

Fig. 4. The Commander gadget

The vertices $\{g_x;\ x \in C\}$ and CM belong to V_R, the rest belongs to V_C. There is one cop, the "commander", whose prescribed starting position is the vertex HQ. From every vertex $w \in V \setminus (V_C \cup \{CM\} \cup \{g_x;\ x \in C\})$ we add the edge (w, HQ) to $\overrightarrow{G}$, thus the only time the commander can leave HQ is when the robber stands at CM. The normal scenario is as follows: If the robber moves to CM, the commander decides one variable x to be changed and moves to h_x, simultaneously the cop in the variable cell V_x starts its movement towards the opposite vertex. The commander temporarily guards the vertex f_x, which is otherwise guarded by the cop in the cell V_x. Then the robber moves (away from CM) and the commander has to return to HQ in the next move.

Lemma 3. *Let us consider the Commander gadget and the variable cells V_x for $x \in C$ with exactly one cop each, standing either on T_x or F_x. Let the robber be at the vertex CM and the cop at HQ, with the cop-player on move. Suppose no other cop can access the vertices in the Commander gadget. Then the normal scenario is that in at most one variable cell V_x, $x \in X$ the cop can start moving from T_x to F_x or vice versa.*

Proof. Only the vertex f_x is temporarily (for one move) guarded by the commander. If two variable cops starts moving, some f_y is unsecured and robber exploits it by moving to g_y in his next move. □

Note that the Manipulator allows the robber to "pass" changing of his variable by setting the current position of cop in some variable. Also note, that the robber may stay on the vertex CM, thus allowing the cop-player to change more than one of his variables. However, in any winning strategy of the robber-player this is not necessary and if the robber-player does not have a winning strategy, this trick does not help him as the cops may pass.

2.2 The gates to V_C

For every clause ϕ of F_R, there is one Robber gate gadget R_ϕ. If ϕ is satisfied by the current setting of variables, R_ϕ allows the robber to enter V_C.

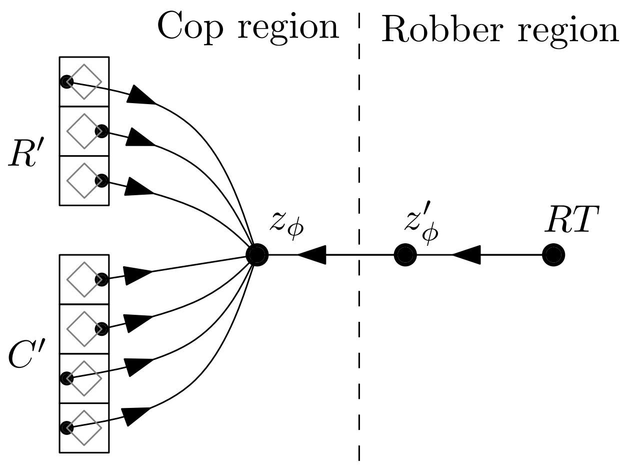

Fig. 5. The Robber Gate R_ϕ

The Robber gate R_ϕ consists of a directed path (RT, z'_ϕ, z_ϕ) and the following edges. Let $\phi = (\ell_1 \& \ldots \& \ell_{12})$ where each ℓ_i is a literal. If $\ell_i = x$ then we put the edge (F_x, z_ϕ) to $\overrightarrow{G}$. If $\ell_i = \neg x$ then we put the edge (T_x, z_ϕ) to $\overrightarrow{G}$. See Fig. 5 for illustration. The vertices $\{z'_\phi;\ \phi \in F_R\}$ and RT belong to V_R, the rest belongs to V_C.

Lemma 4. *Let ϕ be a clause of F_R, consider a Robber Gate R_ϕ. Let the robber stand at the vertex RT and let there be exactly one cop in each V_x, $x \in \phi$, standing either on T_x or F_x. Suppose no other cop can access R_ϕ in less than three moves. Then in the normal scenario robber can reach z_ϕ if and only if ϕ is satisfied under the current setting of variables (given by the positions of cops on variable cells).*

Proof. If ϕ is satisfied, no cop at the variable cells can reach z_ϕ in two (or less) steps. Therefore, the robber may enter z_ϕ. On the other hand, if ϕ is not satisfied, at least one cop is one step from z_ϕ and the robber would be blocked forever if he moves to z'_ϕ. □

For every clause ψ of F_C, there is one *Cop Gate* gadget C_ψ (see Fig. 6). If ψ is satisfied, C_ψ allows cops to forever block the entrance to V_C, the vertices z_ϕ from each Robber Gate R_ϕ. The Cop Gate C_ψ consists of directed paths $(CT, b'_{\psi,x}, b_{\psi,x})$ for each variable x of the clause ψ, the directed cycle $(a_\psi, a'_\psi, a''_\psi, a'''_\psi)$ and edges $\{(a_\psi, b_{\psi,x}), (a''_\psi, b_{\psi,x});\ x \in \psi\}$ and $\{(a''_\psi, z_\phi);\ \phi \in F_R\}$.

Let $\psi = (\ell_1 \& \ldots \& \ell_{12})$ where each ℓ_i is a literal. If $\ell_i = x$ then we put the edge $(T_x, b_{\psi,x})$ to $\overrightarrow{G}$. If $\ell_i = \neg x$ then we put the edge $(F_x, b_{\psi,x})$ to $\overrightarrow{G}$. From the

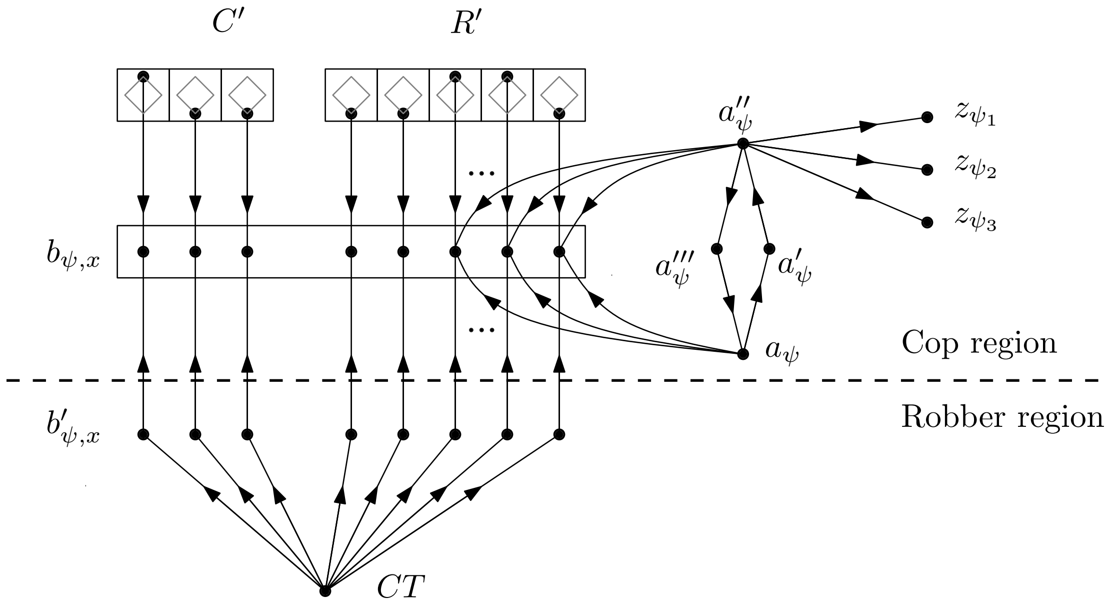

Fig. 6. The Cop Gate C_ψ

vertices a_ψ and a''_ψ there is an edge to every $b_{\psi,x}$ and from a''_ψ there is an edge to every z_ϕ (from each Robber Gate R_ϕ). There is a cop, we call him Arnold, and his prescribed starting position is a_ψ. Each C_ψ has its own Arnold, it would be therefore more correct to name him ψ-Arnold, however, we would use the shorter name if no confusion can occur. The vertices $\{b'_{\psi,x};\ \psi \in F_C,\ x \in \psi\}$ and CT belong to V_R, the rest belongs to V_C.

Lemma 5. *Let us consider a Cop Gate C_ψ. Let there be one cop at the vertex a_ψ (we call him Arnold) and let there be exactly one cop in each V_x, $x \in \psi$, standing on either T_x or F_x. Let the robber be at the vertex CT and no other cop can access C_ψ in less than three moves. Then in the normal scenario, Arnold is able to move to a''_ψ (and therefore block all the entrances z_ϕ forever) without permitting robber to enter V_C if and only if ψ is satisfied under the current setting of variables (given by the position of cops in the variable cells).*

Proof. If ψ is satisfied, the vertices $b_{\psi,x}$, $x \in \psi$ are all guarded by the variable cops, therefore Arnold can start moving from a_ψ towards a''_ψ. If the robber meanwhile moves to some $b'_{\psi,x}$, the variable cop from V_x will intercept him by moving to $b_{\psi,x}$ and the robber loses the game. On the other hand, if ψ is not satisfied, there is some $b_{\psi,x}$ unguarded by the cop from V_x. Therefore, Arnold cannot leave a_ψ, because otherwise robber would reach $b_{\psi,x}$ before Arnold or the cop from V_x could block him. □

2.3 The big picture

We further need to assure that the cops cannot move arbitrarily. This means, that the following must be the normal scenario:

1. During the “Robber Move” phase, the only cop who can move is the cop in variable cell V_x chosen by the robber when he enters Manipulator M_x. All

other variable cops must stand on either T_x of F_x vertices for some variable x. The cop in V_x must reach the vertex T_x from F_x (or vice versa) in two consecutive moves.
2. During the "Robber Test" phase, no cop can move.
3. During the "Cop Move" phase, only the commander and the cop in exactly one variable cell V_x can move. The cop in V_x must reach the vertex T_x from F_x (or vice versa) in two consecutive moves.
4. During the "Cop Test" phase, no other cop than Arnold may move. Arnold may move from vertex a_ψ to a''_ψ and he must do that in two consecutive steps (and of course Arnold may do that only if the clause ψ is satisfied).

We say that we *force* the vertex w by the vertex set S, when for every $v \in S$ we add the oriented path $P_{v,w} = (v, p_{vw}, p'_{vw}, w)$ of length 3 to the graph $\overrightarrow{G}$. The vertices p_{vw}, p'_{vw} belong to V_R. We say that we *block* the vertex w by the vertex set S, when for every $v \in S$ we add the Blocker gadget B_{wv}. The Blocker B_{wv} consists of vertices $p_1^v, p_2^v \in V_R$ and $q_1^v, q_2^v \in V_C$ and the edges $(v, p_i^v), (p_i^v, q_i^v), (w, q_i^v)$ for $i = 1, 2$.

A cop on a vertex w blocked by v cannot leave w even for one move when the robber is on v. Note also that if the cop on w enters q_i^v when it is not necessary to block p_i^v, then he is permanently disabled until the end of the game and the next time the robber visits v he may enter the cop-region through the other p_j^v.

Forcing serves as a tool to prevent moving of more than one variable cops (and Arnolds) however, because of the structure of variable cells, we cannot do it by simply blocking the vertices T_x, F_x and we have to develop the notation of forcing.

Case 1: For every variable $x \in C \cup R$ do the following construction. Let $S_x = \{RM, RT\} \cup \{V(M_y);\ y \in R,\ x \neq y\}$ where $V(M_y)$ are the vertices of Manipulator for variable y. We force the vertices T_x and F_x by the set S_x. Let $S_1 = \{RM\} \cup \{V(M_y);\ y \in R\}$. For each Cop Gate C_ψ, we force the vertex a_ψ by the set S_1. Finally, we block the vertex HQ by the set S_1. Observe that whenever a cop from any other V_y than given by the Manipulator M_x is not on T_y or F_y, the robber can reach V_C faster than the variable cop can block him. On the other hand, if all variable cops are in the right places, the robber may never reach V_C unless being forever blocked. The same holds for Arnold on vertices a_ψ and a''_ψ. The commander cannot move because of the properties of the Blocker gadget. If the variable cop does not use his second turn to finish his movement, the robber will exploit this by reaching V_C by a path from the vertex RT.

Case 2: Let $S_2 = \{RT\} \cup \{z'_\phi;\ \phi \in F_R\}$ and let $F = \{T_x, F_x;\ x \in C \cup R\} \cup \{a_\psi;\ \psi \in F_C\}$. We force every $v \in F$ by the set S and we block the vertex HQ by S_2. Observe that in the normal scenario no cop may move.

Case 3: Let $S_3 = \{CM\}$ and let $F = \{T_x, F_x;\ x \in R\} \cup \{a_\psi;\ \psi \in F_C\}$. We force every $v \in F$ by S_3. Now, in normal scenario, no variable cop from V_x, $x \in R$ may move and by Lemma 3, only commander and exactly one variable cop from V_y, $y \in C$ may move.

Case 4: Let $S_4 = \{CT\}$ and let $F = \{T_x, F_x;\ x \in C \cup R\}$. We force every $v \in F$ by S_4 and we block the vertex HQ by S_4. Observe that in normal scenario no variable cop and the commander may move. The rest follows from Lemma 5 and the fact, that a''_ψ is forced by the vertex RM.

Finally, we connect the vertices in a directed cycle (RM, RT, CM, CT) and let the prescribed starting position r of the robber be the vertex RM. All the construction elements so far presented prove the following corollary.

Corollary 3. *For every game $\mathcal{F} = (\tau, F_C(C,R), F_R(C,R), \alpha)$ there exists a guarding game $\mathcal{G} = (\overrightarrow{G}, V_C, c, S, r)$, $\overrightarrow{G}$ directed, with a prescribed starting positions such that player I wins $\mathcal{F}$ if and only if the robber-player wins the game $\mathcal{G}$.*

Next we note, that we can modify our current construction so that it fully conforms to the definition of the guarding game on a directed graph.

Lemma 6. *Let $\mathcal{G} = (\overrightarrow{G}, V_C, c, S, r)$ be a guarding game with a prescribed starting positions. Let the position r has no in-going edge and let no two cops start at the same vertex. Then there exists a guarding game $\mathcal{G}' = (\overrightarrow{G}', V'_C, c')$, $\overrightarrow{G} \subseteq \overrightarrow{G}'$, $V_C \subseteq V'_C$ such that*

- *the robber-player wins $\mathcal{G}'$ if and only if the robber-player wins the game $\mathcal{G}$*
- *if the cop-player does not place the cops to completely cover S in his first move, he will lose*
- *if the robber-player does not place the robber on r in his first move, the cops win.*

Proof. Consider an edge $(u,v) \in E(\overrightarrow{G})$ such that $u \in V_R$ and $v \in V_C$ (a border edge). Observe, that the out-degree of each such vertex u in our construction is exactly 1. Let $m = |\{v \in V_C;\ (u,v) \in E(\overrightarrow{G}),\ u \in V_R\}|$ be the number of vertices from V_C directly threatened (i.e. in distance 1) from the robber region.

Let us define the graph $\overrightarrow{G}' = (V', E')$ such that $V' = V(\overrightarrow{G}) \cup \{r\} \cup T$ where $T = \{t_1, \ldots, t_m\}$ is the set of new vertices and $E' = E(\overrightarrow{G}) \cup \{(r,v);\ v \in T \cup S\}$. Consider the game $\mathcal{G}' = (\overrightarrow{G}', V'_C, c')$ where $V'_C = V_C \cup T$ and $c' = c + m$. See Fig. 7 for illustration.

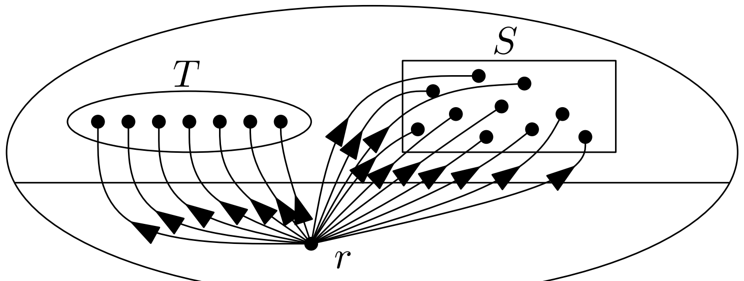

Fig. 7. Forcing starting positions

Suppose that the robber-player places the robber in the first move to some vertex $v \in V_R \setminus \{r\}$. Then there are m vertices in V_C directly threatened by edges going from V_R and because we have at least m cops available, the cops in the second move can occupy all these vertices and prevent the robber from entering V_C forever. So the robber must start at the vertex r. Then observe, that c cops must occupy the positions S and m cops must occupy the vertices T. If any cop does not start either on T or S, the robber wins in the next move. The cops on T remain there harmless to the end of the game. The cops cannot move until the robber decides to leave the vertex r. □

Let us have a guarding game $\mathcal{G} = (\overrightarrow{G}, V_C, c, S, r)$ with prescribed starting positions. Note that in our construction no two cops had the same starting position. We add new vertex r and edge (r, RM) to $\overrightarrow{G}$ and by the previous lemma there is an equivalent guarding game $\mathcal{G}'$, $\mathcal{G} \subseteq \mathcal{G}'$, without prescribed starting positions.

Theorem 1 is now proved.

3 The undirected case

In this section we prove Theorem 2. The idea follows: We take the same construction of directed graph $\overrightarrow{G}$ we used to prove Corollary 3. For each edge we build a gadget such that whenever the resulting undirected edge is used by cop (robber) in bad direction, the cop-player (robber-player) will lose the game.

To obtain the undirected graph G, let us take the graph $\overrightarrow{G}$, and subdivide each edge $e \in E(\overrightarrow{G})$, $e = (u, v)$ by three vertices (see Fig. 8). We number all vertices by $0, 1, 2, 3$, where 0 belongs to starting point of each edge $e \in E(\overrightarrow{G})$ and the newly added vertices e_1, e_2, e_3 are numbered by $1, 2, 3$ according to the orientation of e. If $u \in V_R(\overrightarrow{G})$ then $e_1, e_2, e_3 \in V_R(G)$, otherwise $e_1, e_2, e_3 \in V_C(G)$. Now forget the orientation.

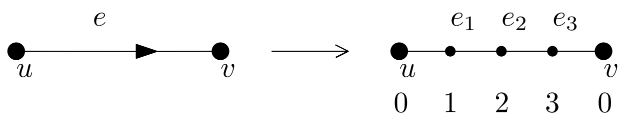

Fig. 8. Subdividing directed edges

We introduce the gadget *Clock* (see Fig. 9). The vertices of Clock are Ω, $c_0, \ldots, c_3$, $c'_0, \ldots, c'_3$ and $V' = \{v';\ v \in V_R\}$. The vertices Ω and c_i, c'_i, $i = 0, 1, 2, 3$, belong to V_C, the set V' belongs to V_R. There are edges $\{\{v, v'\};\ v \in V_R\}$, the neighbours of c_i for $i = 0, 1, 2, 3$ are $c_{(i+1) \bmod 4}, c_{(i-1) \bmod 4}, c'_i$ and $c'_{(i-1) \bmod 4}$. If the number of $v \in V_R$ is j, then the vertex v' is connected to c'_j. From the vertex Ω there are edges to every vertex z_ϕ from each Robber gate gadget and to vertices $c'_0, \ldots, c'_3$. Every subdivided edge is connected to Clock as in Fig. 9.

There is one cop (we call him Chuck) in the Clock, whose prescribed starting position is the vertex c_0. His purpose is following: If the robber is on vertex with number i and moves to vertex with number $j = (i + 1) \bmod 4$, the only thing Chuck can do is to go to vertex c_j (otherwise he loses the game). However, whenever robber does a stupid move (to a vertex with number $j = (i-1) \bmod 4$), Chuck may enter Ω, thus winning the game. Therefore, this gadget forces the robber to pass through undirected edges only in the direction from the old graph $\overrightarrow{G}$.

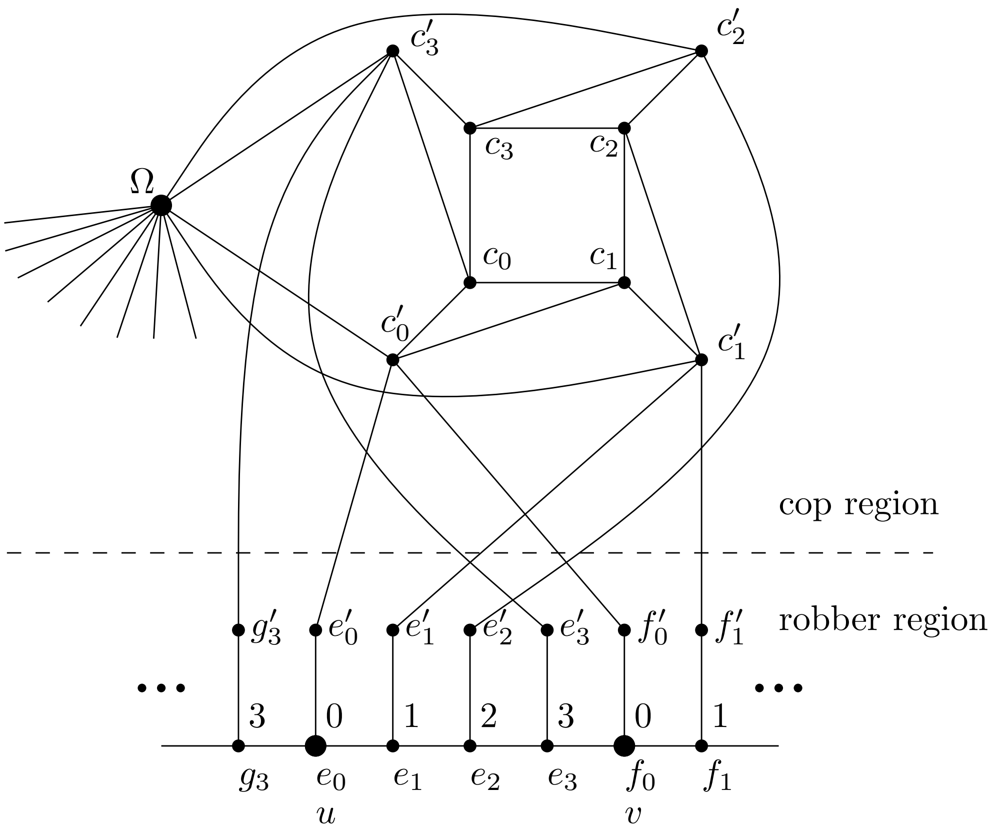

Fig. 9. The Clock gadget

Lemma 7. *Let there be exactly one cop in the Clock gadget (we call him Chuck). Let the robber be at a vertex with number i and let Chuck be at the vertex c_i. Then the normal scenario is that robber must move to a vertex with number* $j = (i + 1) \bmod 4$ *and Chuck must move to vertex c_j.*

Proof. Suppose first that the robber moved to a vertex v_j with number j and Chuck did not move to c_j (he may be at c_i', $c'_{(i-1) \bmod 4}$, $c_{(i-1) \bmod 4}$ or stay at c_i). Then robber may move to v_j' and Chuck cannot prevent him from entering cop-region in the next move.

Suppose now that the robber moved to vertex v_k with number $k = (i - 1) \bmod 4$. Then Chuck goes to c_k', preventing robber from moving to e_k', c_k'. In the next move, robber may or may not enter v_k'. Is he does so, Chuck moves c_k' and guards it until the robber leaves. In both cases, Chuck moves afterwards to Ω, thus being able to block all entrances to the cop-region. Note that Chuck needs only to be able to block vertices z_ϕ and c_i', because other entrances to the cop-region in other gadgets are protected by cops staying at these gadgets, as proved in appropriate lemmas.

If the robber does not move in his turn, Chuck enters c'_i. If robber moves to some v'_i with number i, Chuck keeps guarding c'_i until robber leaves v'_i and then Chuck moves to Ω. Using the same argument as above, the cop is able to win the game.

It remains to show that robber may not enter some vertex v'_i with number i. If he does so, Chuck will move to c'_i, preventing the robber from entering V_C, and guarding there until robber leaves v'_i. Then Chuck moves to Ω and again wins the game using the same argument as above. □

We have ensured the correct movement of robber, imitating the functionality of the graph $\overrightarrow{G}$ on an undirected graph. We still have to do similar thing for cops, as they have to respect the edge orientation as well. In our argument, every pawn has to move in his move. Because staying at one vertex may be a desired part of the cop-player's strategy, for every $v \in V_C(G)$ we glue a subdivided loop l^v of length 4 to v, such that the pawn will move, but in fact stay at one vertex of the original graph $\overrightarrow{G}$. The vertices of l^v are again numbered as above.

There will be four CopDir gadgets D_0, D_1, D_2, D_3. The task of the gadget D_i is to ensure that whenever the robber moves from vertex with number $i-1$ to a vertex with number i, all cops must move to a vertex with number i. If they don't, the robber will be given a chance to enter the cop-region.

The gadget D_i (for $i = 0, 1, 2, 3$) consists of vertices $k_i^1, \dots, k_i^c \in V_C$ and $\ell_i^1, \dots, \ell_i^c \in V_R$, where c is the number of cops in the original $\overrightarrow{G}$, and edges $\{k_i^1, \ell_i^1\}, \dots, \{k_i^c, \ell_i^c\}$. Let $u_i \in V_R$ and $v_i \in V_C$ be vertices of G with number i. For u_i, v_i we add new vertices $u'_i \in V_R$ and $m_{v_i}^1, \dots, m_{v_i}^c$ and edges $\{u_i, u'_i\}$ and $\{\{u'_i, \ell_i^j\}, \{k_i^j, m_{v_i}^j\}, \{m_{v_i}^j, v_i\};\ j = 1, \dots, c\}$. See Fig. 10 for illustration. This means that all vertices v_i and u_i with number i are connected to the gadget D_i via this construction.

Lemma 8. *Let us have the CopDir gadgets D_i for every $u_i \in V_R$ and $v_i \in V_C$ with number i, $i = 0, 1, 2, 3$, and let all pawns be on a vertex with number i, robber being on move. Suppose the robber moves to a vertex with number $j = (i+1) \bmod 4$. Then the normal scenario is that the cop on the vertex with number i moves to a vertex with number j.*

Proof. Suppose that some cop does not move and does not arrive at a vertex with number j and he chooses some other vertex instead. Then the robber moves to u'_j. All cops must now enter the vertices $m_{v_j}^1, \dots, m_{v_j}^c$ to protect the vertices $k_j^1, \dots, k_j^c$ in the next move. However, since there are c cops in total (not counting Chuck on the Clock) and one did not go to a vertex with number j, there is one cop-less $m_{v_j}^p$. Now the robber chooses to go to ℓ_j^p and in the next move he wins, because the cops do not have time to cover k_j^p. Note also, that the robber cannot also pointlessly approach u'_j when all cops are on vertices with number j. All cops would approach $m_{v_j}^1, \dots, m_{v_j}^c$ or even $k_j^1, \dots, k_j^c$ if necessary and then return back to their original positions when the robber decided to go back. □

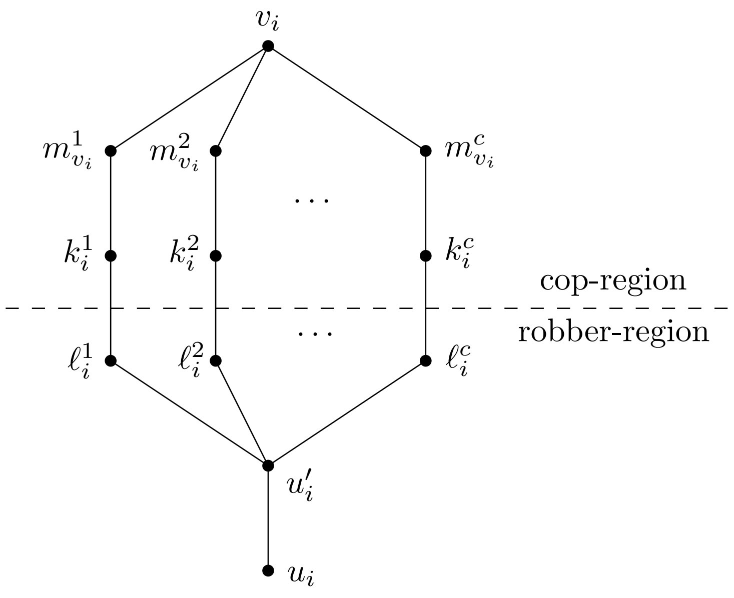

Fig. 10. The CopDir gadget D_i

Note that we have now proved that in the normal scenario, at the beginning of the robber move all pawns stand on vertices with the same number. Observe that using Lemma 7 and Lemma 8, the following corollary holds.

Corollary 4. *For every game $\mathcal{F} = (\tau, F_C(C, R), F_R(C, R), \alpha)$ there exist a guarding game $\mathcal{G} = (G, V_C, c, S, r)$, G undirected, with a prescribed starting positions such that player I wins $\mathcal{F}$ if and only if the robber-player wins the game $\mathcal{G}$.*

The proof of Theorem 2 is now complete.

4 Guarding game, where the cops start

Here we prove Theorem 3.

Proof. To prove the theorem, we first use Corollary 3 to obtain the instance of the guarding game $\mathcal{G} = (G, V_C, c, S, r)$ with prescribed starting positions. Then we construct a device forcing the initial positions of all pawns under the reverse rules that yields exactly the starting positions of the game $\mathcal{G}$.

Let $k = |\{v \in V_C;\ (u, v) \in E(G),\ u \in V_R\}|$ be the number of directly threatened vertices (in distance 1 from V_R). We introduce new vertices Ω, $a_1, \ldots, a_k$, $b_1, \ldots, b_k \in V_C$ and $t \in V_R$. edges $\{(\Omega, a_i), (\Omega, b_i), (r, a_i), (t, b_i);\ i = 1, \ldots, k\}$, and edges $\{(r, s);\ s \in S\}$. From Ω there will be edges (Ω, v) and (v, Ω) to every directly threatened vertex $v \in V_C$, i.e. each v such that there is some $w \in V_R$ with $(w, v) \in E(G)$ (with the exception of r and t). There will be also k new cops, so there will be $c + k$ cops in total.

In the first turn, the cops are placed. We will show that the only meaningful placement of cops (not leading to their defeat) is this:

– There must be a cop on each $s \in S$, or for each cop-less $u \in S$ there must be a cop c_u able to reach u in one step. This gives c cops in total.

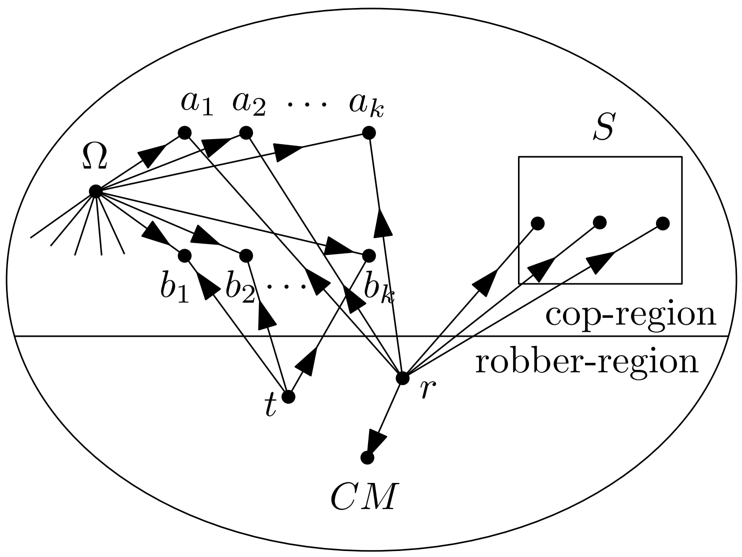

Fig. 11. Forcing starting positions in reverse game

– There must be at least k cops on the vertices $\Omega, a_1, \ldots, a_k$ in total.

– There must be at least k cops on the vertices $\Omega, b_1, \ldots, b_k$ in total.

First suppose that there is less than k cops on the vertices $\{\Omega, b_1, \ldots, b_k\}$ in total. Then in the second turn the robber may be placed on t and the cop-player does not have enough cops to block all vertices, thus in the next move the robber can enter V_C. If there is less than k cops on the vertices $\{\Omega, a_1, \ldots, a_k\}$ in total, the robber may be placed on r and again he wins in the next move.

If there is less than c cops on S or able to cover all vertices of S in one move, then again the robber wins by starting on the vertex r and entering V_C through some unblocked $s \in S$ in the next move. Since we have $c + k$ cops together, at least k cops must start at the vertex Ω.

In the second turn, the robber must be placed at the vertex r. If he starts at t, the vertices $b_1, \ldots, b_k$ are immediately blocked by the cops from Ω and the robber loses. If he starts on some other vertex $v \in V_R$, the cops from Ω will immediately block all entrances to V_C, thus winning the game.

Therefore the robber must be placed on r. Now in the next step k cops from Ω must move to cover all vertices $a_1, \ldots, a_k$ and c cops must move to completely cover the set S. Since we have $c + k$ cops together, all players are now on the prescribed positions and the k extra cops will remain on the vertices $a_1, \ldots, a_k$ until the end of the game. From this we also see that on Ω there had to be exactly k cops placed in the initial move of the game.

No cop may move until the robber leaves r and the game continues as in the case with prescribed positions (recall that r has no in-going edge and the robber never returns to r). □

5 Further questions and Acknowledgements

As we have already mentioned, the relation of the classes PSPACE and E is unclear as we only know that PSPACE $\neq$ E and the current state of the art is missing some deeper understanding of the relation. Therefore, the conjecture of Fomin et al. whether the guarding problem is PSPACE-complete still remain open. However, we believe that the conjecture is not true.

For a guarding game $\mathcal{G} = (G, V_C, c)$, what happens if we restrict the size of strongly connected components of G? If the sizes are restricted by 1, we get DAG, for which the decision problem is PSPACE-complete. For unrestricted sizes we have shown that $\mathcal{G}$ is E-complete. Is there some threshold for $\mathcal{G}$ to become E-complete from being PSPACE-complete? This may give us some insight into the original conjecture. We are also working on forcing the starting position in the guarding game on undirected graphs in a way similar to Theorem 1.

We would like to thank Ruda Stolař for drawing the pictures, extremely useful discussions and some ideas. We thank Peter Golovach for giving a nice talk about the problem, which inspired us to work on it. We would also like to thank Jarik Nešetřil for suggesting some of the previous open questions and to Honza Kratochvíl for fruitful discussion of the paper structure.

References

1. Šámal, R., Stolař, R., Valla, T.: Complexity of the Cop and Robber Guarding Game. Proceedings of IWOCA 2011, Springer-Verlag Lecture Notes in Computer Science vol. 7056, 361–373 (2011)
2. Fomin, F., Golovach, P., Hall, A., Mihalák, M., Vicari, E., Widmayer, P.: How to Guard a Graph? Algorithmica, DOI: 10.1007/s00453-009-9382-4
3. Stockmeyer, L., Chandra, A.: Provably Difficult Combinatorial Games. SIAM J. Comput. Volume 8, Issue 2, 151–174 (1979)
4. Book, R. V.: Comparing complexity classes. J. of Computer and System Sciences 9(2), 213–229 (1974)
5. Alspach, B.: Searching and sweeping graphs: a brief survey. Matematiche (Catania) 59 (1-2), 5–37 (2006)
6. Nowakowski, R., Winkler, P.: Vertex-to-vertex pursuit in a graph. Discrete Math. 43(2-3), 235–239 (1983)
7. Quilliot, A.: Some results about pursuit games on metric spaces obtained through graph theory techniques. European J. Combin. 7(1), 55–66 (1986)
8. Aigner, M., Fromme, M.: A game of cops and robbers. Discrete Appl. Math. 8(1), 1–11 (1984)
9. Nagamochi, H.: Cop-robber guarding game with cycle robber-region. Theoretical Computer Science 412, 383–390 (2011).
10. Thirumala Reddy, T. V., Sai Krishna, D., Pandu Rangan, C.: The guarding problem – complexity and approximation. In IWOCA, volume 5874 of Lecture Notes in Computer Science, 460–470, Springer (2009).
11. Fomin, F.V., Thilikos, D.M.: An annotated bibliography on guaranteed graph searching. Theor. Comp. Sci. 399, 236–245 (2008)

12. Goldstein, A.S., Reingold, E.M.: The complexity of pursuit on a graph. Theor. Comp. Sci. 143, 93–112 (1995)
13. Fomin, F.V., Golovach, P.A., Kratochvíl, J., Nisse, N., Suchan, K.: Pursuing a fast robber on a graph. Theor. Comp. Sci. 411, 1167–1181 (2010)
14. Fomin, F.V., Golovach, P.A., Lokshtanov, D.: Guard Games on Graphs: Keep the Intruder Out! Proceedings of the 7th Workshop on Approximation and Online Algorithms (WAOA 2009), Springer-Verlag Lecture Notes in Computer Science, to appear
15. Fomin, F.V., Golovach, P.A., Kratochvíl, J.: On tractability Cops and Robbers Game. Proceedings of the 5th IFIP International Conference on Theoretical Computer Science (TCS 2008), Springer-Verlag IFIP International Federation for Information Processing 237, 171–185 (2008)
16. Anderson, M., Barrientos, C., Brigham, R., Carrington, J., Vitray, R., Yellen, J.: Maximum demand graphs for eternal security. J. Combin. Math. Combin. Comput. 61, 111–128 (2007)
17. Burger, A.P., Cockayne, E.J., Grundlingh, W.R., Mynhardt, C.M., van Vuuren, J.H., Winterbach, W.: Infinite order domination in graphs. J. Combin. Math. Combin. Comput. 50, 179–194 (2004)
18. Goddard, W., Hedetniemi, S.M., Hedetniemi, S.T.: Eternal security in graphs. J. Combin. Math. Combin. Comput. 52, 169–180 (2005)
19. Goldwasser, J., Klostermeyer, W.F.: Tight bounds for eternal dominating sets in graphs. Discrete Math. 308, 2589–2593 (2008)
20. Klostermeyer, W.F.: Complexity of Eternal Security. J. Comb. Math. Comb. Comput. 61, 135–141 (2007)
21. Klostermeyer, W.F., MacGillivray, G.: Eternal security in graphs of fixed independence number. J. Combin. Math. Combin. Comput. 63, 97–101 (2007)
22. Klostermeyer, W.F., MacGillivray, G.: Eternally Secure Sets, Independence Sets, and Cliques. AKCE International Journal of Graphs and Combinatorics, 2, 119–122 (2005)
23. Klostermeyer, W.F.,MacGillivray, G.: Eternal dominating sets in graphs. J. Combin. Math. Combin. Comput. 68, 97–111 (2009)