Journal of Combinatorial Theory, Series A 118 (2011) 248–256

Contents lists available at ScienceDirect

Journal of Combinatorial Theory, Series A

www.elsevier.com/locate/jcta

The complexity of node blocking for dags

Dariusz Dereniowski [1]

Department of Algorithms and System Modeling, Gdańsk University of Technology, Poland

ARTICLE INFO

Article history:
Received 17 July 2009
Available online 23 March 2010

Keywords:
Annihilation game
Node blocking
PSPACE-completeness

ABSTRACT

We consider the following modification of annihilation games called node blocking. Given a directed graph, each vertex can be occupied by at most one token. There are two types of tokens, each player can move only tokens of his type. The players alternate their moves and the current player i selects one token of type i and moves the token along a directed edge to an unoccupied vertex. If a player cannot make a move then he loses. We consider the problem of determining the complexity of the game: given an arbitrary configuration of tokens in a planar directed acyclic graph (dag), does the current player have a winning strategy? We prove that the problem is PSPACE-complete.

1. Introduction

The study of annihilation games has been suggested by John Conway and the first papers were published by Fraenkel and Yesha [7,9]. They considered a 2-player game played on an underlying directed graph G (possibly with cycles). The current player selects a token and moves it along an arc outgoing from a vertex containing the token. If, as a result of this move, a vertex contains two tokens then they are removed from G (*annihilation*). The authors in [9] gave a polynomial-time algorithm for computing a winning strategy. In this paper we assume normal play, that is, the first player unable to make a move loses (for some results about misère annihilation games see [2]). Fraenkel considered in [4] a generalization of cellular-automata games to two-player games, which also generalizes the above annihilation game.

Fraenkel studied in [3] the connections between annihilation games and error-correcting codes. The authors in [6] gave an algorithm for computing error-correcting codes, which is polynomial in the size of the code and uses the theory of two-player cellular-automata games.

E-mail address: deren@eti.pg.gda.pl.

[1] Partially supported by the Foundation for Polish Science (FNP).

0097-3165/$ – see front matter
doi:10.1016/j.jcta.2010.03.011

Table 1

Game:	Directed acyclic graphs	General graphs
Annihilation[2]	PSPACE-complete [5]	?*
Hit	PSPACE-complete [5]	?*
Capture	PSPACE-complete [10]	EXPTIME-complete [10]
Node blocking	?	EXPTIME-complete [10]
Edge blocking	PSPACE-complete [5]	?*

[2] The version with two types of tokens.

In the following we are interested in generalizations of annihilation games, where there is more than one type of token and/or there is a different interaction between the tokens. The following generalization of annihilation has been proved to be PSPACE-complete for directed acyclic graphs [5]: given $r \geqslant 2$ types of tokens, each type of token can be moved along a subset of the edges (the subsets of edges do not have to be disjoint), and each player can move any token in his turn.

A modification called *hit*, where $r \geqslant 2$ types of tokens and edges are distinguished was considered in [5]. A move consists of selecting a token of type i and moving along an arc of type $i \in \{1, \ldots, r\}$. The target vertex v cannot be occupied by a token of type i, but if v contains token of another type then it is removed (so, when the move ends v is occupied by the token of type i). The complexity of determining the outcome of this game is PSPACE-complete for acyclic graphs and $r = 2$ [5]. A modification of hit called *capture* has the same rules except that each token can travel along any edge. Capture is PSPACE-complete for acyclic and EXPTIME-complete for general graphs [10].

In *node blocking* [8] each token is of one of the two types. Each vertex can contain at most one token. Player i can move the tokens of type i, $i = 1, 2$. All tokens can move along all arcs. Player i makes a move, by selecting one token of type i (occupying a vertex $v \in V$) and an unoccupied vertex $u \in V$ such that $(v, u) \in E$ and moving the token from v to u. The first player unable to make a move loses and his opponent wins the game. There is a tie if there is no last move. First, the game was proved to be NP-hard [8], then PSPACE-hard for general graphs [5]. The complexity for general graphs has been finally proved in [10] to be EXPTIME-complete.

In *edge blocking* [5] all tokens are identical, i.e. each player can move any token, while each arc is of type 1 or 2 and player i makes his move by moving a token along an arc of type i, $i = 1, 2$. Similarly as before, the first player who cannot make a move loses. A tie occurs if there is no last move. This game is PSPACE-complete for dags [5].

Table 1 summarizes the complexity of all the mentioned two-player annihilation games. We list only the strongest known results.

Note that for the entries labeled as '?*' can be replaced by 'PSPACE-hard' (which can be concluded from the corresponding results for acyclic graphs), but the question remains whether the games are in PSPACE. In this paper we are interested in the problem marked by '?', listed also in [1] as one of the open problems. In Section 3 we prove PSPACE-completeness of this game for dags. In Section 4 we modify the graph obtained in the reduction from Section 3 to prove that the problem remains PSPACE-complete for planar directed acyclic graphs.

2. Definitions

In the following a token of type 1 (respectively 2) will be called a *white token* (*black token*, resp.) and denoted by symbol T_W (T_B, resp.). The player moving the white (black) tokens will be denoted by W (B, respectively).

Let $G = (V(G), E(G))$ be a directed graph. A notation $u \rightarrow_p v$, where $u, v \in V(G)$, is used to denote a move made by player $p \in \{W, B\}$ in which the token has been removed from u and placed at vertex v. Given the positions of tokens, define $f(v)$ for $v \in V(G)$ to be one of three possible values $T_W, T_B, \emptyset$ indicating that a white or black token is at the vertex v or there is no token at v, respectively. In the latter case we say that v is *empty*. Note that a move $u \rightarrow_p v$ is correct only if $f(v) = \emptyset$, $(u, v) \in E(G)$ and $f(u) = T_W \wedge p = W$ or $f(u) = T_B \wedge p = B$.

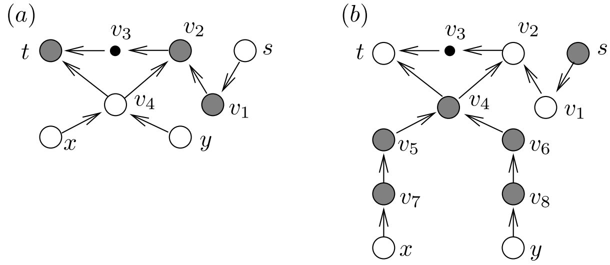

Fig. 1. The graphs G_i for (a) $i = 2j - 1$ (white component) and (b) $i = 2j$ (black component), $j = 1, \dots, n/2$.

Let us recall a PSPACE-complete Quantified Boolean Formula (*QBF*) problem [12]. The input for the problem is a formula Q in the form

$$Q_1 x_1 \dots Q_n x_n F(x_1, \dots, x_n),$$

where $Q_i \in \{\exists, \forall\}$ for $i = 1, \dots, n$. Decide whether Q is true. In our case we use a restricted case of this problem [11] where $Q_1 = \exists$, $Q_{i+1} \neq Q_i$ for $i = 1, \dots, n-1$, n is even, and F is a 3CNF formula, i.e. $F = F_1 \wedge F_2 \wedge \dots \wedge F_m$, where $F_i = (l_{i,1} \vee l_{i,2} \vee l_{i,3})$ and each literal $l_{i,j}$ is a variable or the negation of a variable, $i = 1, \dots, m$, $j = 1, 2, 3$.

3. PSPACE-completeness of node blocking

Define a *variable component* G_i corresponding to x_i as follows:

$$V(G_i) = \{s, t, x, y\} \cup \{v_1, \dots, v_4\},$$
$$E(G_i) = \{(s, v_1), (v_1, v_2), (v_2, v_3), (v_3, t), (v_4, t), (v_4, v_2), (x, v_4), (y, v_4)\}$$

for $i = 2j - 1$, and

$$V(G_i) = \{s, t, x, y\} \cup \{v_1, \dots, v_8\},$$
$$E(G_i) = \{(s, v_1), (v_1, v_2), (v_2, v_3), (v_3, t), (v_4, t), (v_4, v_2), (v_5, v_4), (v_6, v_4), (v_7, v_5), (v_8, v_6), (x, v_7), (y, v_8)\}$$

for $i = 2j$, where $j = 1, \dots, n/2$. Fig. 1 depicts these subgraphs. If i is odd then G_i is called a *white component* and in this case an initial placement of tokens in G_i is $f(s) = f(v_4) = f(x) = f(y) = T_W$, $f(v_3) = \emptyset$ and $f(v_1) = f(v_2) = f(t) = T_B$ (see also Fig. 1(a)). In a *black component* G_i, where i is even, we have $f(s) = f(v_4) = \dots = f(v_8) = T_B$, $f(v_3) = \emptyset$ and $f(v_1) = f(v_2) = f(x) = f(y) = f(t) = T_W$ (see also Fig. 1(b)). In both cases the above configuration of tokens will be called the *initial state* of G_i.

Removing a token from a graph without placing it on another vertex is an invalid operation. However, assume for now that, given an initial state of G_i, the first move is a deletion of a token occupying the vertex t (we will assume in Lemma 1 that the game starts in this way). Then, W (respectively B) becomes the current player in the white (black, resp.) component G_i. Furthermore, we assume that the game in G_i ends when $f(s)$ becomes $\emptyset$.

Lemma 1. *W (respectively B) has a winning strategy in a white (respectively black) component. At the end of the game in a white (black) component exactly one of the vertices x, y (x, y, v_5, v_6, respectively) is empty.*

Proof. First assume that G_i is a white component. Let $f(t) = \emptyset$ and W is the current player. The first two moves are $v_4 \to_W t$, $v_2 \to_B v_3$. Then, there are two possibilities:

$$x \to_W v_4 \text{ or } y \to_W v_4. \tag{1}$$

In both cases the game continues as follows: $v_1 \to_B v_2$, $s \to_W v_1$. The thesis follows.

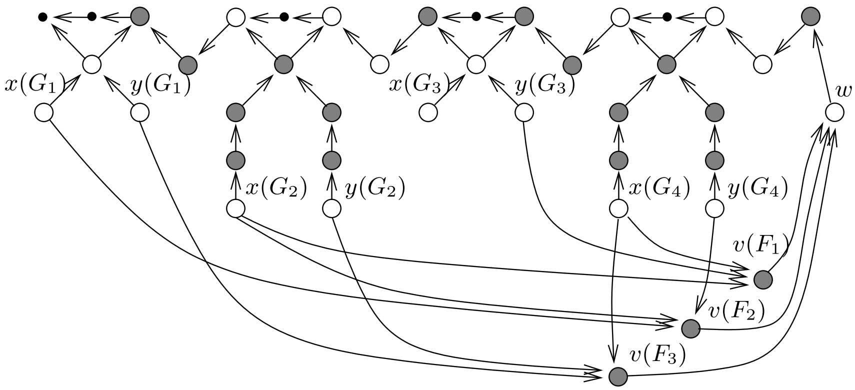

Fig. 2. A complete instance of the graph G_Q corresponding to the formula in (3).

Let G_i be a black component with $f(t) = \emptyset$ and B is the current player. Similarly as before we have $v_4 \rightarrow_B t$, $v_2 \rightarrow_W v_3$. The third move is $v_5 \rightarrow_B v_4$ or $v_6 \rightarrow_B v_4$. Since they are symmetrical, assume in the following that the first case occurred. We have $v_1 \rightarrow_W v_2$. Then B has a choice:

$$v_7 \rightarrow_B v_5 \text{ or } s \rightarrow_B v_1. \tag{2}$$

In the former case the moves $x \rightarrow_W v_7$ and $s \rightarrow_B v_2$ follow, which ends the game and the vertex x is empty among the vertices listed in the lemma. In the latter case in (2) the game ends immediately with $f(v_5) = \emptyset$. □

Now we define a graph G_Q, corresponding to the quantified Boolean formula Q. We will use the symbol $v(G_i)$ in order to distinguish a vertex $v \in V(G_i)$ from the vertices of the other variable components. G_Q contains disjoint white components G_{2i-1} for $i = 1, \ldots, n/2$ and disjoint black components G_{2i}, $i = 1, \ldots, n/2$, connected in such a way that $s(G_i) = t(G_{i+1})$ for $i = 1, \ldots, n-1$. The graph G_Q contains additionally the vertices $w, v(F_1), \ldots, v(F_m)$, an arc $(w, s(G_n))$, the arcs $(v(F_j), w)$ for $j = 1, \ldots, m$, and $(x(G_i), v(F_j)) \in E(G_Q)$ iff F_j contains x_i, while $(y(G_i), v(F_j)) \in E(G_Q)$ iff F_j contains $\overline{x_i}$, a negation of the variable x_i. Initially, all the subgraphs G_i are in the initial state, except that $f(t(G_1)) = \emptyset$. Let $f(w) = T_W$, $f(v(F_j)) = T_B$ for $j = 1, \ldots, m$. Before we prove the main theorem, let us demonstrate the above reduction by giving an example. Let

$$Q = \exists_{x_1} \forall_{x_2} \exists_{x_3} \forall_{x_4} (x_2 \vee \overline{x_3} \vee x_4) \wedge (x_1 \vee x_2 \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_4). \tag{3}$$

Fig. 2 shows the corresponding graph G_Q.

For brevity we introduce a notation: we say that the game *arrives at* a component G_i (and *leaves* G_{i-1}, $i > 1$) if $f(t(G_i)) = \emptyset$ (note that for $i > 1$ this is equivalent to $f(s(G_{i-1})) = \emptyset$ in the graph G_Q). The game *is in* G_i if it arrived at G_i but did not leave G_i.

Theorem 1. *Node blocking is* PSPACE*-complete for directed acyclic graphs.*

Proof. First we prove by induction on $i = 1, \ldots, n$ that

(i) if the game arrives at the component G_i, then for each $j < i$ exactly one of the vertices $x(G_j), y(G_j)$ (if G_j is a white component) or exactly one of the vertices $x(G_j), y(G_j), v_5(G_j), v_6(G_j)$ (if G_j is a black component) is empty in G_j,
(ii) if the game arrives at the component G_i, then all components G_j, for $j = i, \ldots, n$ are in the initial state, except that $f(t(G_i)) = \emptyset$,
(iii) when the game is not in G_i then no moves along the arcs in G_i are performed, $i = 1, \ldots, n$.

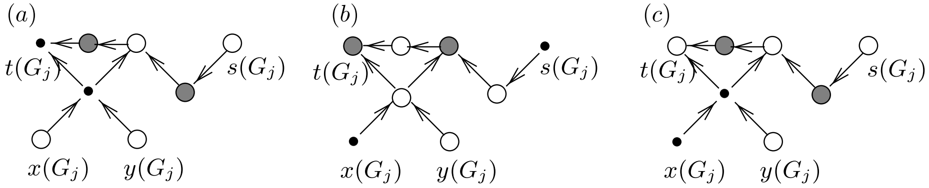

Fig. 3. (a) the game arrives at G_j, (b) the game leaves G_j, (c) W wins the game.

The cases when $i = 1$ and $i > 1$ are analogous. If the game is in G_i then (by the induction hypothesis) all possible moves are the ones along the arcs in G_i,

$$v_2(G_j) \to_p v_3(G_j) \quad \text{for } j > i \tag{4}$$

and

$$v_7(G_j) \to_B v_5(G_j) \text{ or } v_8(G_j) \to_B v_6(G_j) \tag{5}$$

for a black component G_j, $j < i$.

First we exclude (5). The white player may respond to (5) by

$$x(G_j) \to_W v_7(G_j) \text{ or } y(G_j) \to_W v_8(G_j), \tag{6}$$

respectively, and the game continues in G_i. The result is equivalent to the situation where (5) was done when the game was in G_j, because in both cases the vertices $v_4(G_j)$ and $v(F_k)$, $k = 1, \ldots, m$, are not empty. So, if B has a winning strategy in which the corresponding moves (5) and (6) occur then we may w.l.o.g. assume that they are done while the game is in G_j. We will conclude at the end of the proof that B does not have to consider other strategies.

Suppose now that (4) happens while the game is in G_i, where G_j is a white component (the other case is analogous). We have that $p = B$. Let W respond by

$$v_4(G_j) \to_W v_2(G_j). \tag{7}$$

For other moves of B along the arcs of G_i, W replies as in the proof of Lemma 1. Note that B cannot move another token occupying a vertex of G_j until the game arrives at G_j. The game finally arrives at a component G_j which is not in the initial state. This situation is given in Fig. 3(a). Since W is the current player, the first move in G_j is $x(G_j) \to_W v_4(G_j)$ or $y(G_j) \to_W v_4(G_j)$. In both cases the remaining sequence of moves is identical: $v_3(G_j) \to_B t(G_j)$, $v_2(G_j) \to_W v_3(G_j)$, $v_1(G_j) \to_B v_2(G_j)$, $s(G_j) \to_W v_1(G_j)$. The result is shown in Fig. 3(b). This proves that if B performs a move along an arc which is not in G_i when the game is in G_i then W decides among one of the moves $x(G_j) \to_W v_4(G_j)$ or $y(G_j) \to_W v_4(G_j)$ when the game is in G_j. This, however is only true under the assumption that after (4) and (7) W plays according to the schema given in the proof of Lemma 1. If the white player managed to place a token at the vertex $v_4(G_j)$ before the game arrived at G_j, then, when the game arrives at G_j, the move $v_4(G_j) \to_W t(G_j)$ gives a situation depicted in Fig. 3(c)—the black player cannot make a move in G_j. So, if the game is in G_i and a move (4) occurred, then either the game creates the same configuration of tokens in variable components (restricted to the vertices $x(G_k), y(G_k), k = 1, \ldots, n$), or B loses the game. Thus, w.l.o.g. we may assume that if the game is in G_i then the components G_j, $j > i$ are in the initial state, i.e. (ii) is true.

Since we have excluded the moves (4) and (5) when the game is in G_i, we have that (iii) holds. Lemma 1 and (iii) imply that (i) is satisfied.

Now we prove the theorem. The QBF problem in 3CNF form is equivalent to a two-player game where the players take turns choosing variable assignment. We assume here that the players are called the ∃-player and the ∀-player. The ∃-player (respectively ∀-player) sets the values of variables bounded by the existential (universal, resp.) quantifier. If the values of all the variables are determined then the ∃-player wins if and only if F is satisfied. The game proceeds in such a way that the value of x_i is set in the ith turn, $i = 1, \ldots, n$. We will show that our graph game on G_Q simulates the above game for Q, by proving on induction on $i \geqslant 1$ that a player in the QBF game assigns a Boolean value to the variable x_i if and only if the game is in G_i. Moreover, the ∃-player has a winning strategy for the QBF game (which means that Q is true) if and only if W has a winning strategy for node blocking in G_Q.

Assume first that the ∃-player has a winning strategy in the QBF game and that the ith turn begins in the QBF game (the cases when $i=1$ and $i>1$ are similar). At this point, by the induction hypothesis, the values of the variables $x_1,\dots,x_{i-1}$ have been selected by the players and the node blocking game arrives at G_i. By (ii), G_i is in the initial state, except that $f(t(G_i))=\emptyset$. If i is odd, then the ∃-player makes his decision concerning x_i, that is, he sets it to be true or false. The white player 'mirrors' the move made by the ∃-player so that if the ∃-player decides x_i to be true (respectively false), then W plays in G_i in such a way that if the game leaves G_i then $f(x(G_i))=T_W$ ($f(y(G_i))=T_W$, respectively). If i is even, then the ∀-player assigns a Boolean value to x_i arbitrarily, as well as B makes the corresponding decision in the variable component G_i. In both cases, when the value of x_i has been set, then it cannot be changed later. Similarly, once the blocking game left G_i, by (iii), no moves along the arcs in G_i will be performed later during the remaining part of the game.

When the node blocking game leaves G_n, W is the current player. Simultaneously, the last turn in the QBF game ended and F is satisfied under the variable assignment produced during the game. We have $w\to_W s(G_n)$ and $v(F_j)\to_B w$, for some $j\in\{1,\dots,m\}$. Since Q is true, or equivalently, the formula F is satisfied under the variable assignment obtained during the QBF game regardless of the choices of the ∀-player, there is a true literal $l_{j,k}$ in F_j, $k\in\{1,2,3\}$. If $l_{j,k}=x_i$, $i\in\{1,\dots,n\}$, then $f(x(G_i))=T_W$ and $(x(G_i),v(F_j))\in E(G_Q)$, so W can make the move $x(G_i)\to_W v(F_j)$. If $l_{j,k}=\overline{x_i}$, then $f(y(G_i))=T_W$, $(y(G_i),v(F_j))\in E(G_Q)$ and the move $y(G_i)\to_W v(F_j)$ is possible. Note that if $x(G_i)$ or $y(G_i)$ belongs to a black component, then (because Q is true) W always has a possibility to make the above move in such a way that it holds $f(v_5(G_i))=T_B$ or $f(v_6(G_i))=T_B$ (or equivalently, no move $v_5(G_i)\to_B v_4(G_i)$ or $v_6(G_i)\to_B v_4(G_i)$ occurred during the game in G_i). If B can make a move then it must be $v_7(G_k)\to_B v_5(G_k)$ or $v_8(G_k)\to_B v_6(G_k)$ for some $k\in\{1,\dots,n\}$, but then W responds $x(G_k)\to_B v_7(G_k)$ or $y(G_k)\to_B v_8(G_k)$, respectively. The above holds for each index k. No other moves are possible, so W wins the game.

Let now W have a winning strategy. By the induction hypothesis we have that when the blocking game leaves a component G_{i-1}, then the values of $x_1,\dots,x_{i-1}$ are selected in the QBF game. By (iii), no moves along the edges of G_j, $j<i$, will be done during the remaining part of the game, which is consistent with the fact that changing the values of the variables x_j, $j<i$, is not allowed in the QBF game. If i is odd, then the ∃-player, mirrors the way W plays in the white component as follows: he sets x_i to be true if we have the move $y(G_i)\to_W v_4(G_i)$ during the game in G_i, while he decides x_i to be false otherwise, i.e. if there is a move $x(G_i)\to_W v_4(G_i)$ during the game in G_i. If i is even, then the ∀-player assigns a Boolean value to x_i arbitrarily. The game leaves G_n and we have the moves $w\to_W s(G_n)$, $v(F_j)\to_B w$ for some $j\in\{1,\dots,m\}$. The black player chooses j arbitrarily and, since W has a winning strategy, there is possible a move

$$x(G_i)\to_W v(F_j)\text{ or }y(G_i)\to_W v(F_j)\quad\text{for some }i\in\{1,\dots,n\}.\tag{8}$$

If G_i is a black component and $f(v_5(G_i))=\emptyset$ or $f(v_6(G_i))=\emptyset$ then a move $v_7(G_i)\to_B v_5(G_i)$ or $v_8(G_i)\to_B v_6(G_i)$, resp., is possible and B has a win. However, B could make this move while the game was in G_i and force W to make, respectively, $x(G_i)\to_W v_7(G_i)$ or $y(G_i)\to_W v_8(G_i)$. This will make the move in (8) impossible and give the black player a different winning strategy. This justifies our earlier assumption that if the moves (5) and (6) are possible then they can be done when the game is in G_j. From the construction of the strategy for W we have that there is a literal $x_i=\text{true}$ in F_j or a literal $\overline{x_i}=\text{true}$ in F_j, respectively, as a result of the QBF game, regardless of the choices made by the ∀-player during the game.

Observe that $|V(G_Q)|=7n/2+11n/2+m+2$, so this is a polynomial reduction. This proves PSPACE-hardness of node blocking. One can argument that G_Q is acyclic which implies that the game is in PSPACE. □

4. Planar instances

In the following we describe a modification that can be applied to G_Q to obtain a new graph, which is planar and simulates the QBF problem. Define

$$C = \{(u, v(F_i)) \in E(G_Q)\colon\ i = 1, \ldots, m\},$$

i.e. C is the set of arcs of G_Q between the vertices $x(G_i)$, $y(G_i)$, $i = 1, \ldots, n$ and $v(F_j)$, $j = 1, \ldots, m$. The subgraph of G_Q containing the arcs in $E(G_Q) \setminus C$ is clearly planar. We skip here a formal description of an embedding of G_Q in the plane—we assume that if two arcs are intersecting then they both belong to C, and it is a straightforward fact to prove (see Fig. 2 for an example). Moreover, the set C has the following property, assuming that G_Q is in the initial state:

$$(u, v) \in C \quad \Rightarrow \quad (f(u) = T_W \wedge f(v) = T_B). \tag{9}$$

Now we define a gadget, denoted by H, used to modify G_Q in order to eliminate arc intersections. We have

$$V(H) = \{a, b, c, d\} \cup \{u_1, \ldots, u_8\},$$
$$E(H) = \{(a, u_1), (u_1, b), (b, c), (b, u_2), (u_3, d), (u_2, u_3), (u_2, u_4), (u_3, u_5), (u_4, u_5), (u_6, u_4), (u_5, u_7), (u_8, u_6)\}.$$

The *initial state* of H is: $f(a) = f(b) = f(u_1) = f(u_2) = f(u_3) = T_B$, $f(u_7) = \emptyset$ and the remaining vertices of H are occupied by white tokens. The digraph H and its initial configuration are given in Fig. 4(a).

We apply the following modification to G_Q as long as there are intersecting arcs $e_1 = (v_1, w_1), e_2 = (v_2, w_2)$ in C. We remove e_1 and e_2 from G_Q and we place a copy of the graph H at the intersection point. Then, e_1 is replaced by $(v_1, a), (d, w_1)$ while e_2 is replaced by $(v_2, b), (c, w_2)$. The new set C is

$$(C \setminus \{e_1, e_2\}) \cup \{(v_1, a), (d, w_1), (v_2, b), (c, w_2)\}. \tag{10}$$

This process is illustrated in Figs. 4(b) and 4(c). We have the following.

Lemma 2. *If C satisfies* (9)*, then the new set C given in* (10)*, obtained by the above modification, also satisfies* (9)*.* □

We will use the symbol G'_Q to denote the planar graph obtained from G_Q by a series of the above modifications (G_Q will refer to the original (non-planar) graph). When one of the white tokens occupying c or d has been moved along the arc outgoing from c or d, respectively, then we say that the game *arrives at* H. Similarly, the game *leaves* H if one of the vertices a, b has been occupied by a token which initially does not belong to H. If the game arrived at H, but did not leave H, then we say that the game *is in* H.

Observe that if the game did not arrive at a subgraph H then the only move that can be performed along an arc of H is $u_5 \to_W u_7$. Now we prove that the white player does not contribute by making this move when the game is not in H.

Lemma 3. *Let a configuration of tokens in G'_Q be given, such that $H \subseteq G'_Q$ is in the initial configuration, or in the configuration obtained from the initial one by setting $f(b) = T_W$ and $f(c) = T_B$. If W has a winning strategy, then W has a winning strategy that does not perform a move $u_5 \to_W u_7$ in H while the game is not in H.*

Proof. Suppose that the thesis does not hold, i.e. W has no winning strategy that does not make a move $u_5 \to_W u_7$ in a subgraph $H \subseteq G'_Q$ while the game is not in H. We prove that W does not win by performing this move.

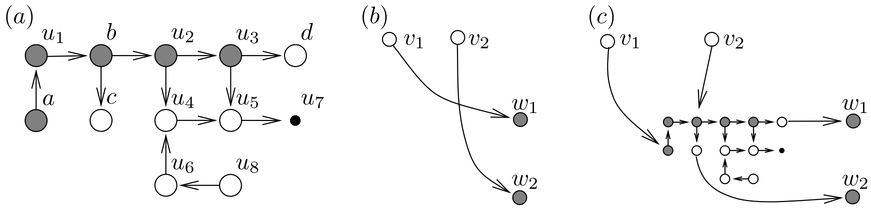

Fig. 4. (a) The subgraph H with its initial configuration; (b) two intersecting arcs; (c) using H to eliminate arc intersections.

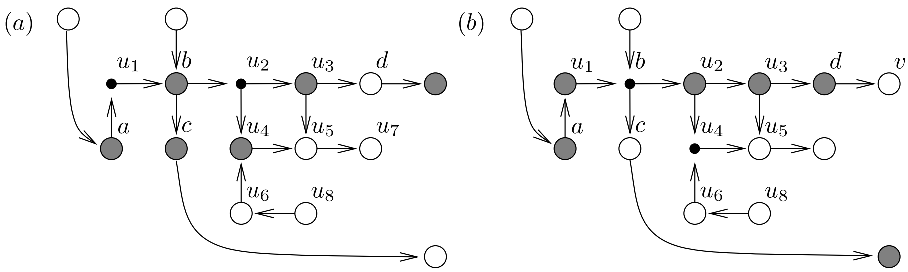

Fig. 5. (a) if the game started with $c \to_W v$, then W should not play $u_5 \to_W u_7$; (b) a configuration after 5 moves following $d \to_W v$.

Assume first that H in the initial configuration. The B's response to $u_5 \to_W u_7$ is $u_3 \to_B u_5$ which leads to such a situation that W cannot move a token along an arc of H until the game is in H. So, the game continues and if it never arrives at this particular component H then (by assumption) B is the winner. On the other hand, if the game arrives at H then, by Corollary 2, this happens as a result of one of the two moves: $c \to_W v$ or $d \to_W v$ for some $v \in V(G'_Q) \setminus V(H)$. The response is $u_2 \to_B u_3$ in both cases and W cannot make a move.

If H is not in the initial configuration, i.e. $f(b) = W$, then we have a situation when the game already was in H. (By the construction of G'_Q, the game cannot arrive at the same component H more than once.) So, after the moves $u_5 \to_W u_7$ and $u_3 \to_B u_5$ the white player cannot respond. □

Similarly as in the case of G_i's we will analyze the flow of the game for H when it arrives at H.

Lemma 4. *If the game arrives at H as a result of move $c \to_W v$ ($d \to_W v$) for some $v \in V(G'_Q)$, then the game leaves H with a move $u \to_W b$ ($u \to_W a$, respectively) for some $u \in V(G'_Q)$. Moreover, if $p \in \{B, W\}$ has a winning strategy when the game arrives at H then p has a winning strategy when the game leaves H.*

Proof. In the case of $c \to_W v$ the black player performs $b \to_B c$. If W plays $u_5 \to_W u_7$ (by Lemma 3 this move did not occur before), then B wins as follows: $u_1 \to_B b$, $u_4 \to_W u_5$, $u_2 \to_B u_4$ and W cannot continue (this final configuration of tokens is shown in Fig. 5(a)). So, as a response to $b \to_B c$, W plays $u \to_W b$ for some $u \in V(G'_Q)$ and the game leaves H. By Lemma 3, no other moves will be done in H during the game.

Assume now that the game arrives at H by a move $d \to_W v$, $v \in V(G'_Q)$. Then, the following must occur: $u_3 \to_B d$, $u_5 \to_W u_7$, $u_2 \to_B u_3$, $u_4 \to_W u_5$, $b \to_B u_2$. Fig. 5(b) depicts the resulting configuration of tokens. So, W has a choice: he can either play $u \to_W b$ for some vertex $u \in V(G'_Q)$ or $u_6 \to_W u_4$. In the former case B plays $u_2 \to_B u_4$ and W responds by $b \to_W u_2$. Then, $u_1 \to_B b$. The result is that W cannot proceed, because it follows from the construction of G'_Q that for all the vertices u' such that $(u', u) \in E(G'_Q)$ we have $f(u') \neq T_W$. In the latter case, i.e. $u_6 \to_W u_4$, we have the sequence of moves $u_1 \to_B b$, $u_8 \to_W u_6$, $a \to_B u_1$, $u \to_W a$ for some $u \in V(G'_Q)$. So, the game leaves H and the thesis follows. □

Theorem 2. *Node blocking is* PSPACE-*complete for planar directed acyclic graphs.*

Proof. By Lemma 3 and Theorem 1 we have that the game simulates assigning Boolean values to the variables and when they all are set then B makes a move $v(F_j) \to_B w$. In the original graph G_Q the white player has a winning strategy if and only if he could perform one more move by choosing a vertex $v \in \{x(G_i), y(G_i)\colon i = 1, \ldots, n\}$, occupied by a white token, and sliding the token along an arc $(v, v(F_j))$. Different subgraphs H in G'_Q will be distinguished by their indices, i.e. we have subgraphs H_k, $k = 1, \ldots, h$, in G'_Q. To refer to a vertex $u \in V(H_k)$ we will write $u(H_k)$.

By the transformation of G_Q into G'_Q, each arc $(v, v(F_j)) \in E(G_Q)$ corresponds to a sequence of arcs

$$(v, s_{j_1}), (t_{j_1}, s_{j_2}), \ldots, (t_{j_{l-1}}, s_{j_l}), \bigl(t_{j_l}, v(F_j)\bigr) \tag{11}$$

of G'_Q, where $j_k \in \{1, \ldots, h\}$, the values of j_k are pairwise different, and

$$\bigl(s_{j_k} = a(H_{j_k}) \wedge t_{j_k} = d(H_{j_k})\bigr) \vee \bigl(s_{j_k} = b(H_{j_k}) \wedge t_{j_k} = c(H_{j_k})\bigr) \tag{12}$$

for each $k = 1, \ldots, l$. By Lemma 4, if the game arrives at H_{j_k}, $k \in \{1, \ldots, l\}$, as a result of moving the token occupying $d(H_{j_k})$ (respectively $c(H_{j_k})$), then the game has to leave this subgraph by a move $u \to_W a(H_{j_k})$ ($u \to_W b(H_{j_k})$, respectively) for some $u \in V(G'_Q)$. By Lemma 3, no moves along the arcs of H_{j_k} will be performed once the game leaves H_{j_k}. By (11) and (12) we have that $u \in \{d(H_{j_{k-1}}), c(H_{j_{k-1}})\}$. We obtain that if the game leaves H_{j_k}, $k > 1$, then it arrives at $H_{j_{k-1}}$. If $k = 1$ then the game leaves H_{j_1} by a move $v \to_W s(H_{j_1})$, $s(H_{j_1}) \in \{a(H_{j_1}), b(H_{j_1})\}$. So, the white player makes the last move $v \to_W v(F_i)$ in the game on G_Q and wins the game if and only if W makes the last move $v \to_W s(H_{j_1})$ in G'_Q. □

References

[1] E.D. Demaine, Playing games with algorithms: Algorithmic combinatorial game theory, in: MFCS '01: Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science, in: Lecture Notes in Comput. Sci., vol. 2136, Springer-Verlag, London, UK, 2001, pp. 18–32.
[2] T.S. Ferguson, Misère annihilation games, J. Combin. Theory Ser. A 37 (3) (1984) 205–230.
[3] A.S. Fraenkel, Error-correcting codes derived from combinatorial games, in: R.J. Nowakowski (Ed.), Games of No Chance, Proc. MSRI Workshop on Combinatorial Games, Berkeley, CA, in: Math. Sci. Res. Inst. Publ., vol. 29, Cambridge Univ. Press, 1994, pp. 417–431.
[4] A.S. Fraenkel, Two-player games on cellular automata, in: R.J. Nowakowski (Ed.), More Games of No Chance, Proc. MSRI Workshop Combinatorial Games, Cambridge Univ. Press, 2002.
[5] A.S. Fraenkel, E. Goldschmidt, PSPACE-hardness of some combinatorial games, J. Combin. Theory Ser. A 46 (1) (1987) 21–38.
[6] A.S. Fraenkel, O. Rahat, Complexity of error-correcting codes derived from combinatorial games, in: Lecture Notes in Comput. Sci., vol. 2883, 2003, pp. 201–212.
[7] A.S. Fraenkel, Y. Yesha, Theory of annihilation games, Bull. Amer. Math. Soc. 82 (5) (1976) 775–777.
[8] A.S. Fraenkel, Y. Yesha, Complexity of problems in games, graphs and algebraic equations, Discrete Appl. Math. 1 (1–2) (1979) 15–30.
[9] A.S. Fraenkel, Y. Yesha, Theory of annihilation games, J. Combin. Theory Ser. B 33 (1982) 60–86.
[10] A.S. Goldstein, E.M. Reingold, The complexity of pursuit on a graph, Theoret. Comput. Sci. 143 (1) (1995) 93–112.
[11] Thomas J. Schaefer, On the complexity of some two-person perfect-information games, J. Comput. System Sci. 16 (2) (1978) 185–225.
[12] L.J. Stockmeyer, A.R. Meyer, Word problems requiring exponential time, in: STOC'73: Proceedings of the Fifth Annual ACM Symposium on Theory of Computing, ACM, New York, NY, 1973, pp. 1–9.