Mathematical Problems of Computer Science 43, 5–25, 2015.

On pre-Hamiltonian Cycles in Hamiltonian Digraphs

Samvel Kh. Darbinyan and Iskandar A. Karapetyan

Institute for Informatics and Automation Problems of NAS RA
e-mail: samdarbin@ipia.sci.am, isko@ipia.sci.am

Abstract

Let D be a strongly connected directed graph of order $n \geq 4$. In [14] (J. of Graph Theory, Vol.16, No. 5, 51-59, 1992) Y. Manoussakis proved the following theorem: Suppose that D satisfies the following condition for every triple x, y, z of vertices such that x and y are nonadjacent: If there is no arc from x to z, then $d(x) + d(y) + d^+(x) + d^-(z) \geq 3n-2$. If there is no arc from z to x, then $d(x) + d(y) + d^-(x) + d^+(z) \geq 3n-2$. Then D is Hamiltonian. In this paper we show that: If D satisfies the condition of Manoussakis' theorem, then D contains a pre-Hamiltonian cycle (i.e., a cycle of length $n-1$) or n is even and D is isomorphic to the complete bipartite digraph with partite sets of cardinalities $n/2$ and $n/2$.

Keywords: Digraphs, Cycles, Hamiltonian cycles, Pre-Hamiltonian cycles, Longest non-Hamiltonian cycles.

1. Introduction

A directed graph (digraph) D is Hamiltonian if it contains a Hamiltonian cycle, i.e., a cycle of length n, and is pancyclic if it contains cycles of all lengths m, $3 \leq m \leq n$, where n is the number of vertices in D. We recall the following well-known degree conditions (Theorems 1.1-1.8) which guarantee that a digraph is Hamiltonian. In each of the conditions (Theorems 1.1-1.8) below D is a strongly connected digraph of order n:

Theorem 1.1: (Ghouila-Houri [12]). *If* $d(x) \geq n$ *for all vertices* $x \in V(D)$, *then* D *is Hamiltonian.*

Theorem 1.2: (Woodall [18]). *If* $d^+(x) + d^-(y) \geq n$ *for all pairs of vertices* x *and* y *such that there is no arc from* x *to* y, *then* D *is Hamiltonian.*

Theorem 1.3: (Meyniel [15]). *If* $n \geq 2$ *and* $d(x) + d(y) \geq 2n-1$ *for all pairs of non-adjacent vertices in* D, *then* D *is Hamiltonian.*

It is easy to see that Meyniel's theorem is a common generalization of Ghouila-Houri's and Woodall's theorems. For a short proof of Theorem 1.3, see [5].

C. Thomassen [17] (for $n = 2k+1$) and S. Darbinyan [7] (for $n = 2k$) proved the following:

Theorem 1.4: (C. Thomassen [17], S. Darbinyan [7]). *If D is a digraph of order $n \geq 5$ with minimum degree at least $n-1$ and with minimum semi-degree at least $n/2-1$, then D is Hamiltonian (unless some extremal cases which are characterized).*

For the next theorem we need the following:

Definition 1: ([14]). *Let k be an arbitrary nonnegative integer. A digraph D satisfies the condition A_k if and only if for every triple x,y,z of vertices such that x and y are nonadjacent: If there is no arc from x to z, then $d(x)+d(y)+d^+(x)+d^-(z) \geq 3n-2+k$. If there is no arc from z to x, then $d(x)+d(y)+d^-(x)+d^+(z) \geq 3n-2+k$.*

Theorem 1.5: (Y. Manoussakis [14]). *If a digraph D of order $n \geq 4$ satisfies the condition A_0, then D is Hamiltonian.*

Each of these theorems imposes a degree condition on all pairs of nonadjacent vertices (or on all vertices). The following three theorems impose a degree condition only for some pairs of nonadjacent vertices.

Theorem 1.6: (Bang-Jensen, Gutin, H. Li [2]). *Suppose that $\min\{d(x),d(y)\} \geq n-1$ and $d(x)+d(y) \geq 2n-1$ for any pair of nonadjacent vertices x,y with a common in-neighbour, then D is Hamiltonian.*

Theorem 1.7: (Bang-Jensen, Gutin, H. Li [2]). *Suppose that $\min\{d^+(x)+d^-(y), d^-(x)+d^+(y)\} \geq n$ for any pair of nonadjacent vertices x,y with a common out-neighbour or a common in-neighbour, then D is Hamiltonian.*

Theorem 1.8: (Bang-Jensen, Guo, Yeo [3]). *Suppose that $d(x)+d(y) \geq 2n-1$ and $\min\{d^+(x)+d^-(y), d^-(x)+d^+(y)\} \geq n-1$ for any pair of nonadjacent vertices x,y with a common out-neighbour or a common in-neighbour, then D is Hamiltonian.*

Note that Theorem 1.8 generalizes Theorem 1.7.

In [11, 16, 6, 8] it was shown that if a digraph D satisfies the condition of one of Theorems 1.1, 1.2, 1.3 and 1.4, respectively, then D also is pancyclic (unless some extremal cases which are characterized). It is natural to set the following problem:

Characterize those digraphs which satisfy the conditions of Theorem 1.6 (1.7, 1.8) but are not pancyclic.

In many papers (in the mentioned papers as well), the existence of a pre-Hamiltonian cycle (i.e., a cycle of length $n-1$) is essential to the show that a given digraph (graph) is pancyclic or not. This indicates that the existence of a pre-Hamiltonian cycle in a digraph (graph) in a sense makes the pancyclic problem significantly easier. For the digraphs which satisfy the conditions of Theorem 1.6 or 1.7 or 1.8 in [9] and [10] the following results are proved:

(i) if the minimum semi-degree of a digraph D at least two and D satisfies the conditions of Theorem 1.6 or a digraph D is not a directed cycle and satisfies the conditions of Theorem 1.7, then either D contains a pre-Hamiltonian cycle (i.e., a cycle of length $n-1$) or n is even and D is isomorphic to the complete bipartite digraph $K^*_{n/2,n/2}$ or to the complete bipartite digraph $K^*_{n/2,n/2}$ minus one arc

(ii) if a digraph D is not a directed cycle and satisfies the conditions of Theorem 1.8, then

D *contains a pre-Hamiltonian cycle or a cycle of length* $n-2$.

In [14] the following conjecture was proposed:

Conjecture 1.9: *Any strongly connected digraph satisfying the condition* A_3 *is pancyclic.*

In this paper using some claims of the proof of Theorem 1.5 (see [14]) we prove the following theorem:

Theorem 1.10: *Any strongly connected digraph* D *on* $n \geq 4$ *vertices satisfying the condition* A_0 *contains a pre-Hamiltonian cycle or* n *is even and* D *is isomorphic to the complete bipartite digraph* $K^*_{n/2,n/2}$.

The following examples show the sharpness of the bound $3n-2$ in the theorem. The digraph consisting of the disjoint union of two complete digraphs with one common vertex or the digraph obtained from a complete bipartite digraph after deleting one arc show that the bound $3n-2$ in the above theorem is best possible.

2. Terminology and Notations

We shall assume that the reader is familiar with the standard terminology on the directed graphs (digraph) and refer the reader to [1] for terminology not discussed here. In this paper we consider finite digraphs without loops and multiple arcs. For a digraph D, we denote by $V(D)$ the vertex set of D and by $A(D)$ the set of arcs in D. The order of D is the number of its vertices. Often we will write D instead of $A(D)$ and $V(D)$. The arc of a digraph D directed from x to y is denoted by xy. For disjoint subsets A and B of $V(D)$ we define $A(A \to B)$ as the set $\{xy \in A(D)/x \in A, y \in B\}$ and $A(A,B) = A(A \to B) \cup A(B \to A)$. If $x \in V(D)$ and $A = \{x\}$ we write x instead of $\{x\}$. If A and B are two disjoint subsets of $V(D)$ such that every vertex of A dominates every vertex of B, then we say that A dominates B, denoted by $A \to B$. The out-neighborhood of a vertex x is the set $N^+(x) = \{y \in V(D)/xy \in A(D)\}$ and $N^-(x) = \{y \in V(D)/yx \in A(D)\}$ is the in-neighborhood of x. Similarly, if $A \subseteq V(D)$, then $N^+(x,A) = \{y \in A/xy \in A(D)\}$ and $N^-(x,A) = \{y \in A/yx \in A(D)\}$. The out-degree of x is $d^+(x) = |N^+(x)|$ and $d^-(x) = |N^-(x)|$ is the in-degree of x. Similarly, $d^+(x,A) = |N^+(x,A)|$ and $d^-(x,A) = |N^-(x,A)|$. The degree of the vertex x in D is defined as $d(x) = d^+(x) + d^-(x)$ (similarly, $d(x,A) = d^+(x,A) + d^-(x,A)$). The subdigraph of D induced by a subset A of $V(D)$ is denoted by $\langle A \rangle$. The path (respectively, the cycle) consisting of the distinct vertices $x_1, x_2, \ldots, x_m$ ($m \geq 2$) and the arcs x_ix_{i+1}, $i \in [1, m-1]$ (respectively, x_ix_{i+1}, $i \in [1, m-1]$, and x_mx_1), is denoted by $x_1x_2 \cdots x_m$ (respectively, $x_1x_2 \cdots x_mx_1$). We say that $x_1x_2 \cdots x_m$ is a path from x_1 to x_m or is an (x_1, x_m)-path. For a cycle $C_k := x_1x_2 \cdots x_kx_1$ of length k, the subscripts considered modulo k, i.e., $x_i = x_s$ for every s and i such that $i \equiv s \pmod{k}$. A cycle that contains all the vertices of D (respectively, all the vertices of D except one) is a Hamiltonian cycle (respectively, is a pre-Hamiltonian cycle). The concept of the pre-Hamiltonian cycle was given in [13]. If P is a path containing a subpath from x to y we let $P[x,y]$ denote that subpath. Similarly, if C is a cycle containing vertices x and y, $C[x,y]$ denotes the subpath of C from x to y. A digraph D is strongly connected (or, just, strong) if there exists a path from x to y and a path from y to x for every pair of distinct vertices x,y. For an undirected graph G, we denote by G^* the symmetric digraph obtained from G by replacing every edge xy with the pair xy, yx of arcs. $K_{p,q}$ denotes the complete bipartite graph with partite sets of cardinalities p and q. Two distinct vertices x and y are adjacent if $xy \in A(D)$ or $yx \in A(D)$ (or both). For integers a and b, $a \leq b$, let $[a, b]$ denote the set of all integers which are not less than a and are not greater than b. Let C be a non-Hamiltonian cycle in digraph D. An (x, y)-path P is a C-bypass if $|V(P)| \geq 3$, $x \neq y$ and $V(P) \cap V(C) = \{x, y\}$.

3. Preliminaries

The following well-known simple Lemmas 3.1-3.4 are the basis of our results and other theorems on directed cycles and paths in digraphs. They will be used extensively in the proofs of our results.

Lemma 3.1: **[11].** *Let D be a digraph of order $n \geq 3$ containing a cycle C_m, $m \in [2, n-1]$. Let x be a vertex not contained in this cycle. If $d(x, C_m) \geq m+1$, then D contains a cycle C_k for all $k \in [2, m+1]$.*

The following lemma is a slight modification of the lemma by Bondy and Tomassen [5].

Lemma 3.2: *Let D be a digraph of order $n \geq 3$ containing a path $P := x_1x_2 \ldots x_m$, $m \in [2, n-1]$ and let x be a vertex not contained in this path. If one of the following conditions holds:*

(i) $d(x, P) \geq m+2$;

(ii) $d(x, P) \geq m+1$ and $xx_1 \notin D$ or $x_mx \notin D$;

(iii) $d(x, P) \geq m$, $xx_1 \notin D$ and $x_mx \notin D$, then there is an $i \in [1, m-1]$ such that $x_ix, xx_{i+1} \in D$, i.e., D contains a path $x_1x_2 \ldots x_ixx_{i+1} \ldots x_m$ of length m (we say that x can be inserted into P or the path $x_1x_2 \ldots x_ixx_{i+1} \ldots x_m$ is an extended path from P with x).

If in Lemmas 3.1 and 3.2 instead of the vertex x consider a path Q, then we get the following Lemmas 3.3 and 3.4, respectively.

Lemma 3.3: *Let $C_k := x_1x_2 \ldots x_kx_1$, $k \geq 2$, be a non-Hamiltonian cycle in a digraph D. Moreover, assume that there exists a path $Q := y_1y_2 \ldots y_r$, $r \geq 1$, in $D - C_k$. If $d^-(y_1, C_k) + d^+(y_r, C_k) \geq k+1$, then for all $m \in [r+1, k+r]$ the digraph D contains a cycle C_m of length m with vertex set $V(C_m) \subseteq V(C_k) \cup V(Q)$.*

Lemma 3.4: *Let $P := x_1x_2 \ldots x_k$, $k \geq 2$, be a non-Hamiltonian path in a digraph D. Moreover, assume that there exists a path $Q := y_1y_2 \ldots y_r$, $r \geq 1$, in $D - P$. If $d^-(y_1, P) + d^+(y_r, P) \geq k + d^-(y_1, \{x_k\}) + d^+(y_r, \{x_1\})$, then D contains a path from x_1 to x_k with vertex set $V(P) \cup V(Q)$.*

For the proof of our result we also need the following:

Lemma 3.5: **([14]).** *Let D be a digraph on $n \geq 3$ vertices satisfying the condition A_0. Assume that there are two distinct pairs of nonadjacent vertices x, y and x, z in D. Then either $d(x) + d(y) \geq 2n-1$ or $d(x) + d(z) \geq 2n-1$.*

4. The Proof of Theorem 1.10

In the proof of Theorem 1.10 we often will use the following definition:

Definition 2: *Let* $P_0 := x_1x_2\ldots x_m$, $m \geq 2$, *be an arbitrary* (x_1,x_m)*-path in a digraph* D *and let* $y_1,y_2,\ldots,y_k \in V(D) - V(P_0)$. *For* $i \in [1,k]$ *we denote by* P_i *an* (x_1,x_m)*-path in* D *with vertex set* $V(P_{i-1}) \cup \{y_j\}$ *(if it exists) such that* P_i *is an extended path obtained from* P_{i-1} *with some vertex* y_j, *where* $y_j \notin V(P_{i-1})$. *If* $e+1$ *is the maximum possible number of these paths* $P_0,P_1,\ldots,P_e$, $e \in [0,k]$, *then we say that* P_e *is an extended path obtained from* P_0 *with vertices* $y_1,y_2,\ldots,y_k$ *as much as possible. Notice that* P_i *(*$i \in [0,e]$*) is an* (x_1,x_m)*-path of length* $m+i-1$.

Proof of Theorem 1.10: Let $C := x_1x_2\ldots x_kx_1$ be a longest non-Hamiltonian cycle in D of length k, and let C be chosen so that $\langle V(D) - V(C)\rangle$ has the minimum number of connected components. Suppose that $k \leq n-2$ and $n \geq 5$ (the case $n = 4$ is trivial). It is easy to show that $k \geq 3$. We will prove that D is isomorphic to the complete bipartite digraph $K^*_{n/2,n/2}$. Put $R := V(D) - V(C)$. Let $R_1,R_2,\ldots,R_q$ be the connected components of $\langle R\rangle$ (i.e., if $q \geq 2$, then for any pair i,j, $i \neq j$, there is no arc between R_i and R_j). In [14] it was proved that for any R_i, $i \in [1,q]$, the subdigraph $\langle V(C) \cup V(R_i)\rangle$ contains a C-bypass. (The existence of a C-bypass also follows from Bypass Lemma (see [4]), since $\langle V(C) \cup V(R_i)\rangle$ is strong and the condition A_0 implies that the underlying graph of the subdigraph $\langle V(C) \cup V(R_i)\rangle$ is 2-connected). Let $P := x_my_1y_2\ldots y_{t_i}x_{m+\gamma_i}$ be a C-bypass in $\langle V(C) \cup V(R_i)\rangle$ ($i \in [1,q]$ is arbitrary) and γ_i is considered to be minimum in the sense that there is no C-bypass $x_au_1u_2\ldots u_{l_i}x_{a+r_i}$ in $\langle V(C) \cup V(R_i)\rangle$ such that $r_i < \gamma_i$ and $\{x_a,x_{a+r_i}\}$ is a subset of $\{x_m,x_{m+1},\ldots,x_{m+\gamma_i}\}$.

We will distinguish two cases, according as there is a γ_i, $i \in [1,q]$, such that $\gamma_i = 1$ or not.

Assume first that $\gamma_i \geq 2$ for all $i \in [1,q]$. For this case one can show that (the proofs are the same as the proofs of Case 1, Lemma 2.3 and Claim 1 in [14]) if $\gamma_i \geq 2$, then $t_i = |R_i| = 1$, in $\langle V(C)\rangle$ there is an $(x_{m+\gamma_i},x_m)$-path (say, P') of length $k-2$ with vertex set $V(P') = V(C) - \{z_i\}$, where $z_i \in \{x_{m+1},x_{m+2},\ldots,x_{m+\gamma_i-1}\}$ and $d(y_1)+d(z_i) \leq 2n-2$ (note that y_1 and z_i are nonadjacent). From $|R| \geq 2$ and $|R_i| = 1$ (for all i) it follows that $q \geq 2$. If $u \in R_2$, then $d(u) = d(u,C) \leq k$ (by Lemma 3.1) and $d(z_1,R) = 0$ (by minimality of q), in particular, the vertices z_1 and u are nonadjacent. Therefore, $d(z_1) = d(z_1,C) \leq k$ and $d(z_1)+d(u) \leq 2n-2$. This in connection with $d(y_1)+d(z_1) \leq 2n-2$ contradicts Lemma 3.5.

Assume second that $\gamma_i = 1$ for all $i \in [1,q]$. It is clear that $q = 1$. Put $t := t_1$ and $\gamma := \gamma_1 = 1$.

Observe that if $v_1v_2\ldots v_j$ (maybe, $j = 1$) is a path in $\langle R\rangle$ and $x_iv_1 \in D$, then $v_jx_{i+j} \notin D$ since C is the longest non-Hamiltonian cycle in D and $k \leq n-2$. We shall use this often, without mentioning this explicitly.

The following claim follows immediately from $\gamma = 1$ and the maximality of C.

Claim 1: $R = \{y_1,y_2,\ldots,y_t\}$ *(i.e.,* $t = n-k \geq 2$*),* $y_1y_2\ldots y_t$ *is a Hamiltonian path in* $\langle R\rangle$ *and if* $1 \leq i < j-1 \leq t-1$, *then* $y_iy_j \notin D$.

From Claim 1 it follows that

$$d^+(y_1,R) = d^-(y_t,R) = 1 \quad \text{and if} \quad i \in [1,t-1]; \quad \text{then } d^+(y_i,R) \leq i; \qquad (1)$$

$$d(y_1,R),\ d(y_t,R) \le n-k \quad \text{and if} \quad i \in [2,t-1]; \quad \text{then} \quad d(y_i,R) \le n-k+1. \tag{2}$$

Claim 2: *(i). If* $x_iy_1 \in D$, *then* $d^-(x_{i+1}, \{y_1, y_2, \dots, y_{t-1}\}) = 0$;

(ii). If $y_tx_{i+1} \in D$, *then* $d^+(x_i, \{y_2, y_3, \dots, y_t\}) = d^+(x_{i-1}, \{y_1, y_2, \dots, y_{t-1}\}) = 0$;

(iii). $d(y_1,C) \le k$, $d(y_t,C) \le k$ *and* $d(y_j,C) \le k-1$ *for all* $j \in [2,t-1]$ *(by Lemma 3.2(iii) and Claim 2(ii) since* $\gamma = 1$*).* □

Claim 3: *Assume that* $\langle R\rangle$ *is strong. If* $d^+(x_i,R) \ge 1$, $d^-(x_j,R) \ge 1$ *and* $|C[x_i,x_j]| \ge 3$ *for some two distinct vertices* x_i,x_j *(*$i,j \in [1,k]$*), then the following holds:*

(i) $d^-(x_{j-1},R) \ne 0$ *or* $A(R,C[x_{i+1},x_{j-2}]) \ne \emptyset$;

(ii) $d^+(x_{i+1},R) \ne 0$*) or* $A(R,C[x_{i+2},x_{j-1}]) \ne \emptyset$.

(Here if $|C[x_i,x_j]| = 3$, *then* $C[x_{i+1},x_{j-2}] = \emptyset$ *and* $C[x_{i+2},x_{j-1}] = \emptyset$*).*

Proof of Claim 3: Suppose that Claim 3(i) is false. Without loss of generality, assume that $x_ky_f, y_gx_l \in D$ ($l \in [2,k-1]$)

$$d^-(x_{l-1},R) = 0 \quad \text{and} \quad A(R,C[x_1,x_{l-2}]) = \emptyset. \tag{3}$$

The subdigraph $\langle R\rangle$ contains a (y_f,y_g)-path (say $P(y_f,y_g)$) since R is strong. We extend the path $P_0 := C[x_l,x_k]$ with the vertices $x_1, x_2, \dots, x_{l-1}$ as much as possible. Then some vertices $z_1, z_2, \dots, x_d \in \{x_1, x_2, \dots, x_{l-1}\}$, $d \in [1,l-1]$, are not on the extended path P_e (for otherwise, it is not difficult to see that by Definition 2 there is an (x_l,x_k)-path P_i, $i \in [0,e]$, which together with the path $P(y_f,y_g)$ and the arcs x_ky_f, y_gx_l forms a non-Hamiltonian cycle longer than C). Therefore, by Lemma 3.2(i), for all $s \in [1,d]$ the following holds

$$d(z_s,C) \le k+d-1. \tag{4}$$

From (3) it follows that $y_1x_{l-1} \notin D$ and $y_tx_{l-1} \notin D$. Hence, by Lemma 3.2(ii), we have

$$d(y_1,C) \le k-l+2 \quad \text{and} \quad d(y_t,C) \le k-l+2$$

since neither y_1 nor y_t cannot be inserted into $C[x_{l-1},x_k]$. This together with (2) implies that

$$d(y_1) \le n-l+2 \quad \text{and} \quad d(y_t) \le n-l+2. \tag{5}$$

If there exists a z_s such that $d(z_s,R) = 0$, then by $d \le l-1$, (4) and (5) we obtain that

$$d(z_s)+d(y_1) \le 2n-2 \quad \text{and} \quad d(z_s)+d(y_t) \le 2n-2,$$

which contradicts Lemma 3.5 since z_s,y_1 and z_s,y_t are two distinct pairs of nonadjacent vertices. Assume, therefore, that there is no z_s such that $d(z_s,R) = 0$. Then from (3) it follows that $d = 1$, $z_1 = x_{l-1}$ and $d^+(x_{l-1},R) \ge 1$. Therefore, D contains an (x_l,x_k)-path, say Q, with vertex set $V(C) - \{x_{l-1}\}$. Since $\langle R\rangle$ is strong, it follows that in $\langle R\rangle$ there is a (y_f,y_g)-path, say T. This path T together with the path Q and the arcs x_ky_f, y_gx_l forms a cycle C' which does not contain x_{l-1}. From the maximality of C it follows that $|T| = 1$ (i.e., $y_f = y_g$) and

$$d^+(x_k,R-\{y_f\}) = d^-(x_l,R-\{y_f\}) = 0. \tag{6}$$

So, the cycle C' has the length k and $V(C') = V(C) \cup \{y_f\} - \{x_{l-1}\}$. It is not difficult to see that the vertices x_{l-1}, y_f are nonadjacent (for otherwise $x_{l-1}y_f \in D$ and $x_{l-1}y_fx_l \dots x_kx_1 \dots x_{l-1}$ is a cycle of length $k+1$, a contradiction). From this and

$d^-(x_{l-1};R)=0$ (by (3)) we have $d(x_{l-1};R)\leq n-k-1$. This together with $d=1$ and (4) implies that $d(x_{l-1})\leq n-1$.

Assume first that $y_f\neq y_1$.

Let $x_{l-1}y_1\in D$. Then $y_f=y_t$ (by Claim 2(i)) and for the triple of vertices y_t, x_{l-1}, y_1 condition A_0 holds, since $y_1x_{l-1}\notin D$ and y_t, x_{l-1} are nonadjacent. Since $y_tx_l\in D$, from (3) and Claim 2(ii) it follows that $d(x_{l-1};R-\{y_1\})=0$, i.e., $d(x_{l-1};R)=1$. This together with (4) and $d=1$ gives $d(x_{l-1})\leq k+1$. Since D contains no cycle of length $k+1$, it follows that for the arc $x_{l-1}y_1$ and the cycle C', by Lemma 3.3, the following holds $d^-(x_{l-1};C')+d^+(y_1;C')\leq k$. This together with $d^+(y_1;R)=1$ and $d^-(x_{l-1};R)=0$ implies that $d^-(x_{l-1})+d^+(y_1)\leq n-2$ (here we consider the cases $k=n-2$ and $k\leq n-3$ separately). Therefore, using condition A_0, (5), $d(x_{l-1})\leq n-1$ and $l\geq 2$ we obtain

$$3n-2\leq d(y_t)+d(x_{l-1})+d^-(x_{l-1})+d^+(y_1)\leq 3n-3,$$

a contradiction.

Let now $x_{l-1}y_1\notin D$. Then by (3) the vertices x_{l-1}, y_1 are nonadjacent. From this $t\geq 3$ since y_f, x_{l-1} are nonadjacent and $d^+(x_{l-1};R)\geq 1$. Thus, we have $x_ky_1\notin D$, $y_1x_l\notin D$ (by (6)) and $d(y_1;C[x_1;x_{l-1}])=0$. Therefore, since y_1 cannot be inserted into $C[x_l;x_k]$, using Lemma 3.2(iii) and (2) we obtain $d(y_1)\leq n-l$. Notice that (by (2) and (4))

$$d(x_{l-1})=d(x_{l-1};C)+d(x_{l-1};R-\{y_1,y_f\})\leq k+d(x_{l-1};R-\{y_1,y_f\})\leq n-2,$$

and (by Lemma 3.2(i) and $d(y_f;C[x_1;x_{l-1}])=0$),

$$d(y_f)=d(y_f;C)+d(y_f;R)\leq k-l+2+d(y_f;R).$$

From the last three inequalities we obtain that

$$d(y_1)+d(x_{l-1})\leq 2n-l-2,$$

and

$$d(y_f)+d(x_{l-1})\leq 2k-l+2+d(x_{l-1};R-\{y_1,y_f\})+d(y_f;R).$$

Notice that

$$d(x_{l-1};R-\{y_1,y_f\})+d(y_f;R)\leq n-k-2+n-k=2n-2k-2$$

since if $x_{l-1}y_j\in D$, then $y_jy_f\notin D$, where $y_j\neq y_1, y_f$. The last two inequalities give $d(y_f)+d(x_{l-1})\leq 2n-l\leq 2n-2$. This together with $d(y_1)+d(x_{l-1})\leq 2n-l-2$ contradicts Lemma 3.5 since x_{l-1}, y_1 and x_{l-1}, y_f are two distinct pairs of nonadjacent vertices.

Assume next that $y_f=y_1$. If x_{l-1}, y_t are nonadjacent, then $d(x_{l-1};\{y_1, y_t\})=0$ and $d(x_{l-1};R)\leq n-k-2$. Hence, by (4) and $d=1$ we have $d(x_{l-1})\leq n-2$. This together with (5) implies that

$$d(y_1)+d(x_{l-1})\leq 2n-2 \quad\text{and}\quad d(y_t)+d(x_{l-1})\leq 2n-2,$$

which contradicts Lemma 3.5, since y_1, x_{l-1} and y_t, x_{l-1} are two distinct pairs of nonadjacent vertices. So, we can assume that $x_{l-1}y_t\in D$. Since C' is a longest non-Hamiltonian cycle, $d^-(x_{l-1};R)=0$, (3) and $d^+(y_t;R-\{y_1\})\leq n-k-2$, from Lemma 3.3 it follows that

$d^-(x_{l-1}) + d^+(y_t) \le n-2$. Now using (5), $d(x_{l-1}) \le n-1$ and the condition A_0, for the triple of the vertices x_{l-1}, y_1, y_t we obtain

$$3n-2 \le d(y_1) + d(x_{l-1}) + d^+(y_t) + d^-(x_{l-1}) \le 3n-l-1 \le 3n-3,$$

which is a contradiction. Claim 3 is proved.

In particular, from Claim 3 immediately follows the following

Claim 4: *Assume that* $\langle R\rangle$ *is strong and* $d^+(x_i,R) \ge 1$, $d^-(x_j,R) \ge 1$ *for some two distinct vertices* x_i *and* x_j. *Then the following holds:*

(i) if $|C[x_i,x_j]| \ge 3$, *then* $A(R, C[x_{i+1},x_{j-1}]) \ne \emptyset$;

(ii) if $|C[x_i,x_j]| = 3$, *then* $d^+(x_{i+1},R) \ge 1$ *and* $d^-(x_{j-1},R) \ge 1$.

Now we divide the proof of the theorem into two parts: $k \le n-3$ and $k = n-2$.

Part 1. $k \le n-3$, *i.e.,* $t \ge 3$.

For this part first we will prove the following Claims 5-10 below.

Claim 5: *Let* $t \ge 3$ *and* $y_ty_1 \in D$. *Then the following holds*

(i) if x_iy_1D, *then* $d^-(x_{i+2},R) = 0$; *(ii) if* $y_tx_i \in D$, *then* $d^+(x_{i-2},R) = 0$, *where* $i \in [1,k]$.

Proof of Claim 5: (i). Suppose, on the contrary, that for some $i \in [1,k]$ $x_iy_1 \in D$ and $d^-(x_{i+2},R) \ne 0$. Without loss of generality, we assume that $x_i = x_1$ and $d^-(x_3,R) \ne 0$. Then $d^-(x_3, R-\{y_1\}) = 0$ and $y_1x_3 \in D$. It is easy to see that y_1, x_2 are nonadjacent and

$$d^-(x_2,\{y_1,y_2,\ldots,y_{t-1}\}) = d^+(x_2,\{y_1,y_3,y_4,\ldots,y_t\}) = 0, \quad \text{i.e.,} \quad d(x_2,R) \le 2. \tag{7}$$

Since neither y_1 nor x_2 can be inserted into $C[x_3,x_1]$, using (2), (7) and Lemma 3.2, we obtain that

$$d(y_1) = d(y_1,C) + d(y_1,R) \le k+n-k = n \quad \text{and} \quad d(x_2) = d(x_2,C) + d(x_2,R) \le k+2.$$

On the other hand, by Lemma 3.3 and (1) we have that $d^-(y_t) + d^+(y_1) \le k+2$ since the arc y_ty_1 cannot be inserted into C. Therefore, by condition A_0, the following holds

$$3n-2 \le d(y_1) + d(x_2) + d^-(y_t) + d^+(y_1) \le n+2k+4,$$

since y_1, x_2 are nonadjacent and $y_1y_t \notin D$. From this and $k \le n-3$ it follows that $k = n-3$, $x_2y_2, y_2y_1 \in D$ and hence, the cycle $x_2y_2y_1x_3x_4\ldots x_kx_1x_2$ has length $k+2$. This contradicts the supposition that C is a maximal non-Hamiltonian cycle.

To show that (ii) is true, it is sufficient to apply the same arguments to the converse digraph of D. Claim 5 is proved.

Claim 6: *If* $t \ge 3$ *and the vertices* y_1, y_t *are nonadjacent, then* $t = 3$ *and* y_3y_2, $y_2y_1 \in D$.

Proof of Claim 6: Without loss of generality, we can assume that x_1y_1, $y_tx_2 \in D$ (since $\gamma = 1$).

Assume first that $t \ge 4$ and $y_ty_i \in D$ for some $i \in [2,t-2]$. Since the arc y_ty_i cannot be inserted into C, using Lemma 3.3, we obtain

$$d^-(y_t,C) + d^+(y_i,C) \le k. \tag{8}$$

From Claim 1 and the condition that y_1, y_t are nonadjacent it follows that

$$d(y_1,R) \le n-k-1 \quad \text{and} \quad d(y_t,R) \le n-k-1.$$

This together with Claim 2(iii) implies that $d(y_1)$ and $d(y_t) \le n-1$. Since y_1, y_t are nonadjacent and $y_iy_t \notin D$, using (1), (8) and applying the condition A_0 to the triple of the vertices y_1, y_t, y_i, we obtain

$$3n-2 \le d(y_1)+d(y_t)+d^-(y_t,C)+d^+(y_i,C)+d^-(y_t,R)+d^+(y_i,R) \le 3n-3,$$

which is a contradiction.

Assume second that $t \ge 4$ and $y_ty_i \notin D$ for all $i \in [2, t-2]$. We also can assume that $y_iy_1 \notin D$ for all $i \in [3, t-1]$. Therefore, $d(y_1,R) \le 2$ and $d(y_t,R) \le 2$. This together with Claim 2(iii) implies that $d(y_1) \le k+2$, $d(y_t) \le k+2$ and hence

$$d(y_1)+d(y_t) \le 2k+4. \qquad (9)$$

From $t \ge 4$ and the above assumptions it follows that y_1, y_t and y_1, y_{t-1} are two distinct pairs of nonadjacent vertices. From (9) and $k \le n-4$ it follows that $d(y_1)+d(y_t) \le 2n-4$. On the other hand, since $d(y_1) \le k+2$, $d(y_{t-1},C) \le k-1$ (by Claim 2(iii)) and $d(y_{t-1},R) \le n-k$ (by Claim 1), we have

$$d(y_1)+d(y_{t-1}) \le 2n-3.$$

This together with $d(y_1)+d(y_t) \le 2n-4$ contradicts Lemma 3.5. We, thus, proved that the case $t \ge 4$ is impossible.

Assume finally that $t = 3$. Now we will show that $y_3y_2 \in D$. Assume that this is not the case, i.e., $y_3y_2 \notin D$. Then we can apply the condition A_0 to the triple of the vertices y_1, y_3, y_2, since the vertices y_1, y_3 are nonadjacent and $y_3y_2 \notin D$. Notice that the arc y_2y_3 cannot be inserted into C and hence $d^-(y_2,C)+d^+(y_3,C) \le k$ (by Lemma 3.3). Therefore, by A_0 and Claim 2(iii), we obtain

$$3n-2 \le d(y_1)+d(y_3)+d^-(y_2)+d^+(y_3) \le 3k+4 \le 3n-5,$$

which is a contradiction. Therefore $y_3y_2 \in D$.

Similarly we obtain a contradiction if we assume that $y_2y_1 \notin D$. Therefore, $y_2y_1 \in D$. Claim 6 is proved.

Claim 7: *If* $t \ge 3$, *then* $y_ty_1 \in D$.

Proof of Claim 7: Suppose, on the contrary, that $t \ge 3$ and $y_ty_1 \notin D$, i.e., y_1, y_t are nonadjacent. Then by Claim 6, $t = 3$ and $y_3y_2, y_2y_1 \in D$. Without loss of generality, assume that x_1y_1 and $y_3x_2 \in D$ (since $\ge = 1$). Notice that $d(y_1), d(y_3) \le n-1$ (by Lemma 3.1) and hence, $d(y_1)+d(y_3) \le 2n-2$. We will distinguish two cases, according as there is an arc from R to $\{x_3, x_4, \ldots, x_k\}$ or not.

Case 7.1. $A(R \to \{x_3, x_4, \ldots, x_k\}) \ne \emptyset$.

Then there exists a vertex x_l with $l \in [3,k]$ such that $d^-(x_l,R) \ge 1$ and for $l \ge 4$, $A(R \to \{x_3, x_4, \ldots, x_{l-1}\}) = \emptyset$.

If $l = 3$, then from $d^-(x_3, \{y_2, y_3\}) = 0$ it follows that $y_1x_3 \in D$. From this it is easy to see that $d(x_2, \{y_1, y_2\}) = 0$. Since neither y_1 nor y_3 and x_2 can be inserted into $C[x_3, x_1]$ using Lemma 3.2 we obtain that $d(y_1)$, $d(y_3)$ and $d(x_2) \le n-1$. Hence, $d(y_1)+d(y_3) \le 2n-2$ and $d(y_1)+d(x_2) \le 2n-2$, which contradicts Lemma 3.5 since y_1, y_3 and y_1, x_2 are two distinct pairs of nonadjacent vertices.

Assume, therefore, that $l \ge 4$. If $d^+(x_{l-1},R) = 0$, then $d(x_{l-1},R) = 0$ by minimality of l. Therefore, Claim 4 implies that there is no $x_i \in C[x_2, x_{l-2}]$ such that $d^+(x_i,R) \ge 1$. Therefore, by the minimality of l we have

$$A(R, C[x_3, x_{l-1}]) = \emptyset \quad \text{and} \quad d^+(x_2,R) = 0,$$

which contradicts Claim 3(ii) since $x_1y_1 \in D$ and $d^-(x_l,R) \ge 1$. Assume, therefore, that $d^+(x_{l-1},R) \ge 1$. Without loss of generality, we may assume that $y_gx_l \in D$ and $x_{l-1}y_f \in D$. It is easy to see that $y_f \ne y_g$, $y_f,y_g \in \{y_1,y_3\}$ (i.e., $x_{l-1}y_2 \notin D$ and $y_2x_l \notin D$) and the vertices x_{l-1},x_g are nonadjacent.

Assume first that $l = 4$. If $y_g = y_3$ (i.e., $y_3x_4 \in D$), then $x_1y_1y_2y_3x_4 \ldots x_{n-3}x_1$ is a cycle of length $n-1$, a contradiction. Assume, therefore, that $y_g = y_1$ and $y_f = y_3$, i.e., y_1x_4 and $x_3y_3 \in D$. Then the vertices x_2,y_2 are clearly nonadjacent and $x_2y_3 \notin D$. Since $y_1x_4 \in D$ and $d^-(x_3,R) = 0$, Claim 4(ii) implies that $x_2y_1 \notin D$. Therefore, $d(x_2,\{y_1,y_2\}) = 0$. Notice that x_2 cannot be inserted into the path $C[x_4,x_1]$ (for otherwise in D there is a cycle of length $n-3$ which does not contain the vertices y_2,y_3,x_3 but this contradicts Claim 6 since y_2,x_3 are nonadjacent and $y_3x_3 \notin D$). Now by Lemma 3.2 and the above observation we obtain that

$$d(x_2) = d(x_2,C[x_4,x_1]) + d(x_2,R) + d(x_2,\{x_3\}) \le n-1.$$

Therefore, $d(y_1) + d(x_2) \le 2n-2$, which together with $d(y_1) + d(y_3) \le 2n-2$ contradicts Lemma 3.5, since y_1,x_2 and y_1,y_3 are two distinct pairs of nonadjacent vertices.

Assume next that $l \ge 5$. From the minimality of l, $d^-(x_{l-1},R) = 0$ and Claim 4(ii) it follows that $d(x_{l-2},R) = 0$. Therefore, there is no $x_i \in C[x_2,x_{l-2}]$ such that $d^+(x_i,R) \ge 1$, in particular, $x_2y_3 \notin D$. Therefore

$$A(C[x_3,x_{l-2}],R) = \emptyset \quad \text{and} \quad d(x_2,R) = 1;$$

(only $y_3x_2 \in D$). Since $y_g \ne y_2$ and x_{l-1},y_g are nonadjacent, we have $d(x_{l-1},R) = 1$ (only $x_{l-1}y_{4-g} \in D$). By the above observation we have

$$d(y_1,C[x_2,x_{l-2}]) = d(y_3,C[x_3,x_{l-2}]) = 0. \tag{10}$$

Since y_1 cannot be inserted into C, $x_2y_3 \notin D$ and $d^-(x_{l-1},R) = 0$, using (10) and Lemma 3.2 we obtain that $d(y_1,C) \le k-l+3$. This together with $d(y_1,R) = 2$ implies that $d(y_1) \le k-l+5$.

Now we extend the path $P_0 := C[x_l,x_1]$ with the vertices $x_2,x_3,\ldots,x_{l-1}$ as much as possible. Then some vertices $z_1,z_2,\ldots,z_d \in \{x_2,x_3,\ldots,x_{l-1}\}$, $d \in [1,l-2]$, are not on the extended path P_e. Therefore, $d(z_i,C) \le k+d-1$ and hence, $d(z_i) \le k+d$ for all $i \in [1,d]$. Thus we have $d(y_1) + d(z_i) \le 2n-3$ and $d(y_3) + d(z_i) \le 2n-3$ since there is a vertex z_i which is not adjacent to y_1 or y_3. This together with $d(y_1) + d(y_3) \le 2n-2$ contradicts Lemma 3.5 since y_1,z_i (or y_3,z_i) and y_1,y_3 are two distinct pairs of nonadjacent vertices. In each case we have a contradiction. The discussion of Case 7.1 is completed.

Case 7.2. $A(R \to \{x_3,x_4,\ldots,x_k\}) = \emptyset$.

Without loss of generality, we may assume that $A(\{x_3,x_4,\ldots,x_k\} \to R) = \emptyset$ (for otherwise, we consider the converse digraph of D for which the considered Case 7.1 holds). Therefore $A(R,\{x_3,x_4,\ldots,x_k\}) = \emptyset$. In particular, x_k is not adjacent to the vertices y_1 and y_3. Notice that

$$d(y_1) = d(y_1,R) + d(y_1,C) \le 2 + d(y_1,\{x_1,x_2\}) \le 5;$$

$d(y_3) \le 5$ and $d(x_k) = d(x_k,C) \le 2n-8$. Therefore $d(x_k) + d(y_1) \le 2n-3$ and $d(x_k) + d(y_3) \le 2n-3$, which contradicts Lemma 3.5. Claim 7 is proved. □

Claim 8: *If* $t\geq 3$ *and for some* $i\in[1,k]$ x_iy_1, *then* $A(R\to C[x_{i+2},x_{i-1}])=\emptyset$.

Proof of Claim 8: Suppose that the claim is not true. Without loss of generality, we may assume that $x_1y_1\in D$ and $A(R\to\{x_3,x_4,\ldots,x_k\})\neq\emptyset$. Then there is a vertex x_l with $l\in[3,k]$ such that $d^-(x_l,R)\geq 1$ and if $l\geq 4$, then $A(R\to\{x_3,x_4,\ldots,x_{l-1}\})=\emptyset$. We have that $y_ty_1\in D$ (by Claim 7). In particular, $y_ty_1\in D$ implies that $\langle R\rangle$ is strong. On the other hand, by Claim 5(i), $d^-(x_3,R)=0$ and hence, $l\geq 4$. From $x_1y_1\in D$ it follows that there exists a vertex x_r with $r\in[1,l-1]$ such that $d^+(x_r,R)\geq 1$. Choose r with these properties as maximal as possible. Let x_ry_f and $y_gx_l\in D$. Notice that in $\langle R\rangle$ there is a (y_f,y_g)-path since $\langle R\rangle$ is strong. Using Claims 4(i) and 3(ii) we obtain that $r=l-1$. Then $y_f\neq y_g$ and in $\langle R\rangle$ any (y_f,y_g)-path is a Hamiltonian path. Since $\langle R\rangle$ is strong, from $d^-(x_{l-1},R)=0$, $d^-(x_l,R)\geq 1$ and from Claim 3(i) it follows that $A(\{x_2,x_3,\ldots,x_{l-2}\}\to R)=\emptyset$, in particular, $d^+(x_2,R)=0$. By the above observations we have

$$A(\{x_3,x_4,\ldots,x_{l-2}\},R)=\emptyset;\quad d(y_1,\{x_2,x_3,\ldots,x_{l-2}\})=d(x_2,\{y_1,y_2,\ldots,y_{t-1}\})=0. \quad (11)$$

Note that x_2, y_1 and x_2, y_2 are two distinct pairs of nonadjacent vertices. We extend the path $P_0:=C[x_l,x_1]$ with the vertices $x_2,x_3,\ldots,x_{l-1}$ as much as possible. Then some vertices $z_1,z_2,\ldots,z_d\in\{x_2,x_3,\ldots,x_{l-1}\}$, where $d\in[1,l-2]$, are not on the extended path P_e (for otherwise, since in $\langle R\rangle$ there is a (y_f,y_g)-path, using the path P_{e-1} or P_e we obtain a non-Hamiltonian cycle longer than C). By Lemma 3.2, for all $i\in[1,d]$ we have

$$d(z_i,C)\leq k+d-1 \quad\text{and}\quad d(z_i)=d(z_i,C)+d(z_i,R)\leq k+d-1+d(z_i,R). \qquad (12)$$

Assume that there is a vertex $z_i\neq x_{l-1}$. Then, by (11), $d(z_i,R)\leq 1$ (since $d(x_2,R)\leq 1$). Notice that y_1, z_i and y_2, z_i are two distinct pairs of nonadjacent vertices (by (11)). Since neither y_1 nor y_2 can be inserted into $C[x_{l-1},x_1]$ and $y_1x_{l-1}\notin D$, $y_2x_{l-1}\notin D$, by Lemma 3.2(ii) and (11) for $j=1$ and 2 we obtain

$$d(y_j,C)=d(y_j,C[x_{l-1},x_1])\leq k-l+3. \qquad (13)$$

In particular, by (2),

$$d(y_1)=d(y_1,C)+d(y_1,R)\leq k-l+3+n-k=n-l+3.$$

This together with (12) and $d(z_i,R)\leq 1$ implies that

$$d(y_1)+d(z_i)\leq 2n-2;$$

since $k\leq n-3$ and $d\leq l-2$. Therefore, by Lemma 3.5, $d(y_2)+d(z_i)\geq 2n-1$. Hence, by (2) and (12) we have

$$2n-1\leq d(y_2)+d(z_i)\leq n+d+d(z_i,R)+d(y_2,C).$$

From this, $d\leq l-2$ and (13) it follows that $d(y_2,C)=k-l+3$, $d(z_i,R)=1$ and $k=n-3$. Then $z_i=x_2$ and $y_tx_2\in D$ (by (11) and $d^+(x_2,R)=0$). Therefore, $x_1y_2\notin D$. From this, $y_2x_{l-1}\notin D$ and $d(y_2,C)=k-l+3$, by Lemma 3.2(iii), we conclude that y_2 can be inserted into C, which is contrary to our supposition that C is a longest non-Hamiltonian cycle.

Now assume that there is no $z_i\neq x_{l-1}$. Then $d=1$, $z_1=x_{l-1}$ and there is an (x_l,x_1)-path with vertex set $V(C)-\{x_{l-1}\}$. Therefore, $d^-(x_l,\{y_2,y_3,\ldots,y_t\})=0$ (since $x_1y_1\in D$)

and $y_1x_l \in D$. From this we have, $d(x_{l-1}; R - \{y_2\}) = 0$ since $y_ty_1 \in D$ and l is minimal, in particular, the vertices $y_t; x_{l-1}$ are nonadjacent. This together with (12) implies that $d(x_{l-1}) \le k + 1$ (only $x_{l-1}y_2 \in D$ is possible). Notice that neither y_t nor the arc y_ty_1 can be inserted into C, and therefore, by Lemmas 3.2, 3.3 and by (1), (2) we obtain that $d(y_t) \le n$ and $d^-(y_t) + d^+(y_1) \le k + 2$. Since $y_1y_t \notin D$ and $y_t; x_{l-1}$ are nonadjacent we have that the triple of the vertices $y_t; x_{l-1}$, y_1 satisfies condition A_0. Therefore

$$3n - 2 \le d(x_{l-1}) + d(y_t) + d^-(y_t) + d^+(y_1) \le 3n - 3$$

since $k \le n - 3$, which is a contradiction. Claim 8 is proved. □

Claim 9: *If* $t \ge 3$, x_1y_1 *and* $y_tx_2 \in D$, *then* $d^-(x_1; R) = 0$.

Proof of Claim 9: Assume that $d^-(x_1; R) \ge 1$. By Claim 7, $y_ty_1 \in D$. Now using Claims 5(ii) and 8, we obtain that $d^+(x_k; R) = 0$ and

$$A(R \to \{x_3; x_4; \ldots; x_k\}) = \emptyset. \qquad (14)$$

In particular, $d(x_k; R) = 0$. This together with $d^-(x_1; R) \ge 1$, (14) and Claim 3 implies that $A(\{x_2; x_3; \ldots; x_{k-1}\} \to R) = \emptyset$. Now again using (14) we get that $A(\{x_3; x_4; \ldots; x_k\}; R) = \emptyset$. This together with $d^+(x_2; R) = d^-(x_2; \{y_1; y_2; \ldots; y_{t-1}\}) = 0$ implies that $d(x_2; R) = 1$, $d(y_2; C) \le 1$ (only $y_2x_1 \in D$ is possible) and $d(x_3; R) = 0$. Therefore, by (2),

$$d(y_2) + d(x_3) = d(y_2; C) + d(y_2; R) + d(x_3; R) + d(x_3; C) \le n + k \le 2n - 3$$

and $d(y_2) + d(x_2) \le 2n - 2$, which contradicts Lemma 3.5 since $y_2; x_3$ and $y_2; x_2$ are two distinct pairs of nonadjacent vertices. This completes the proof of Claim 9. □

Claim 10: *If* $t \ge 3$, x_1y_1 *and* $y_tx_2 \in D$, *then* $A(\{x_3; x_4; \ldots; x_k\} \to R) = \emptyset$.

Proof of Claim 10: By Claim 7, $y_ty_1 \in D$. Suppose that $A(\{x_3; x_4; \ldots; x_k\} \to R) \ne \emptyset$. Recall that Claim 5(ii) implies that $d^+(x_k; R) = 0$. Let x_r, $r \in [3; k - 1]$, be chosen so that $x_ry_i \in D$ for some $i \in [1; t]$ and r is maximum possible. Then $A(\{x_{r+1}; x_{r+2}; \ldots; x_k\}; R) = \emptyset$ and $d^-(x_1; R) = 0$ by Claim 8 and Claim 9, respectively. This together with $y_tx_2 \in D$ contradicts Claim 3(i). Claim 10 is proved.

Now we are ready to complete the proof of Theorem 1.10 for Part 1 (when $k \le n - 3$, i.e., $t \ge 3$). By Claim 7, $y_ty_1 \in D$. Without loss of generality, we may assume that x_1y_1 and $y_tx_2 \in D$ since , $= 1$. Then from Claims 8, 9 and 10 it follows that

$$A(R \to \{x_3; x_4; \ldots; x_k; x_1\}) = A(\{x_3; x_4; \ldots; x_k\} \to R) = \emptyset.$$

From this and

$$d^-(x_2; \{y_1; y_2; \ldots; y_{t-1}\}) = d^+(x_1; \{y_2; y_3; \ldots; y_t\}) = 0$$

we obtain that $x_1; y_2$ and $x_1; y_t$ are two distinct pairs of nonadjacent vertices and $d(y_2; C) \le 1$, $d(y_t; C) \le 2$, $d(x_1; R) = 1$. Therefore, $d(y_2) \le n - k + 2$, $d(y_t) \le n - k + 2$ (by (2)) and $d(x_1) \le 2k - 1$. The last three inequalities imply that $d(y_2) + d(x_1) \le 2n - 2$ and $d(y_t) + d(x_1) \le 2n - 2$, which contradicts Lemma 3.5 and completes the discussion of Part 1.

Part 2. $k = n - 2$, *i.e.,* $t = 2$.

For this part first we will prove Claims 11-16 below.

Claim 11: *If* $x_iy_f \in D$ *and* $y_2y_1 \notin D$*, where* $i \in [1, n-2]$ *and* $f \in [1, 2]$*, then there is no* $l \in [3, n-2]$ *such that* $y_fx_{i+l-1} \in D$ *and* $d(y_f, \{x_{i+1}, x_{i+2}, \ldots, x_{i+l-2}\}) = 0$.

Proof of Claim 11: The proof is by contradiction. Suppose that $x_iy_f, y_fx_{i+l-1} \in D$ and $d(y_f, \{x_{i+1}, x_{i+2}, \ldots, x_{i+l-2}\}) = 0$ for some $l \in [3, n-2]$. Without loss of generality, we may assume that $x_i = x_1$. Then $x_1y_f, y_fx_l \in D$ and $d(y_f, \{x_2, x_3, \ldots, x_{l-1}\}) = 0$. Since D contains no cycle of length $n-1$, using Lemmas 3.2 and 3.3, we obtain that

$$d^-(y_1) + d^+(y_2) \le n-2 \quad \text{and} \quad d(y_f) \le n-l+2. \qquad (15)$$

We extend the path $P_0 := C[x_l, x_1]$ with the vertices $x_2, x_3, \ldots, x_{l-1}$ as much as possible. Then some vertices $z_1, z_2, \ldots, z_d \in \{x_2, x_3, \ldots, x_{l-1}\}$, $d \in [1, l-2]$, are not on the extended path P_e. Therefore, by Lemma 3.2, $d(z_1) = d(z_1, C) + d(z_1, \{y_{3-f}\}) \le n + d - 1$. Now, since the vertices y_f, z_1 are nonadjacent and $y_2y_1 \notin D$, by condition A_0 and (15) we have

$$3n-2 \le d(y_f) + d(z_1) + d^-(y_1) + d^+(y_2) \le 3n-3,$$

a contradiction. Claim 11 is proved. □

Claim 12: $y_2y_1 \in D$ *(i.e., if* $k = n-2$*, then* $\langle V(D) - V(C)\rangle$ *is strong).*

Proof of Claim 12: Suppose, on the contrary, that $y_2y_1 \notin D$. Without loss of generality, we may assume that $x_1y_1 \in D$ and the vertices y_1, x_2 are nonadjacent. Then $y_2x_3 \notin D$ and since D contains no cycle of length $n-1$, using Lemma 3.3 for the arc y_1y_2 we obtain that

$$d^-(y_1) + d^+(y_2) \le n-2. \qquad (16)$$

Case 12.1. $d^+(y_1, C[x_3, x_{n-2}]) \ge 1$.

Let x_l, $l \in [3, n-2]$, be chosen so that $y_1x_l \in D$ and l is minimum, i.e., $d^+(y_1, C[x_2, x_{l-1}]) = 0$. It is easy to see that the vertices y_1 and x_{l-1} are nonadjacent. By Claim 11, we can assume that $l \ge 5$ (if $l \le 4$, then $d(y_1, C[x_2, x_{l-1}]) = 0$, a contradiction to Claim 11) and $d^-(y_1, C[x_3, x_{l-2}]) \ge 1$. It follows that there exists a vertex x_r with $r \in [3, l-2]$ such that $x_ry_1 \in D$ and $d(y_1, C[x_{r+1}, x_{l-1}]) = 0$. Consequently, for the vertices y_1, x_r and x_l Claim 11 is not true, a contradiction.

Case 12.2. $d^+(y_1, C[x_3, x_{n-2}]) = 0$.

Then $d^+(y_1, C[x_2, x_{n-2}]) = 0$ and either $y_1x_1 \in D$ or $y_1x_1 \notin D$.

Subcase 12.2.1. $y_1x_1 \in D$.

Then $x_{n-2}y_1 \notin D$ and hence, the vertices y_1, x_{n-2} are nonadjacent. Therefore, the triple of the vertices y_1, x_{n-2}, y_2 satisfies the condition A_0. Claim 11 implies that $d^-(y_1, C[x_2, x_{n-2}]) = 0$. This together with $d^+(y_1, C[x_2, x_{n-2}]) = 0$ and $y_2y_1 \notin D$ gives $d(y_1) = 3$. Clearly, $d(x_2) \le 2n-4$ and hence, for the vertices y_1, y_2, x_2 by condition A_0 and (16) we have,

$$3n-2 \le d(y_1) + d(x_2) + d^-(y_1) + d^+(y_2) \le 3n-3,$$

which is a contradiction.

Subcase 12.2.2. $y_1x_1 \notin D$.

Then $d^+(y_1, C) = 0$, $d^+(y_1) = 1$ and $d^+(y_2, C) \ge 1$ since D is strong. Without loss of generality, we may assume that $d^-(y_2, C) = 0$ (for otherwise for the vertex y_2 in the converse digraph of D we would have the above considered Case 12.1 or Subcase 12.2.1). Since the triple of the vertices y_1, y_2, x_3 satisfies the condition A_0, $d(y_1) \le n-2$, $d(x_2) \le 2n-5$ and (16), it is not difficult to show that $n \ge 7$.

Suppose first that $y_2x_2 \in D$. Then $x_{n-2}y_1 \notin D$ and hence, the vertices x_{n-2}, y_1 are nonadjacent.

Let for some $l \in [3, n-3]$ $x_ly_1 \in D$ and $d^-(y_1, C[x_{l+1}, x_{n-2}]) = 0$. Then $d(y_1, C[x_{l+1}, x_{n-2}]) = 0$ and $d(y_1) \le l$ since $d^+(y_1, C) = 0$ and x_2, y_1 are nonadjacent. Extend the path $P_0 := C[x_2, x_l]$ with the vertices $x_{l+1}, x_{l+2}, \ldots, x_{n-2}, x_1$ as much as possible. Then some vertices $z_1, z_2, \ldots, z_d \in \{x_{l+1}, x_{l+2}, \ldots, x_{n-2}, x_1\}$, $d \in [2, n-l-1]$, are not on the extended path P_e. For a vertex $z_i \ne x_1$ by Lemma 3.2 we obtain that $d(z_i) = d(z_i, C) + d(z_i, \{y_2\}) \le n + d - 1$. Therefore, since $y_2y_1 \notin D$ and the vertices z_i, y_1 are nonadjacent, by condition A_0 and (16), we get that

$$3n-2 \le d(y_1) + d(z_i) + d^-(y_1) + d^+(y_2) \le 3n-4,$$

which is a contradiction.

Let now $x_ly_1 \notin D$ for all $l \in [3, n-2]$, i.e., $d^-(y_1, C[x_3, x_{n-2}]) = 0$. Then from $d^+(y_1, C[x_2, x_{n-2}]) = 0$, $y_1x_1 \notin D$ and $x_{n-2}y_2 \notin D$ (since $d^-(y_2, C) = 0$) it follows that $d(y_1) = 2$ and $d(x_{n-2}) \le 2n-5$. From this, since the vertices y_1, x_{n-2} are nonadjacent and $y_2y_1 \notin D$, by condition A_0 and (16) we have that

$$3n-2 \le d(y_1) + d(x_{n-2}) + d^-(y_1) + d^+(y_2) \le 3n-5,$$

which is a contradiction.

Suppose next that $y_2x_2 \notin D$. Then $d(y_2, \{x_2, x_3\}) = 0$, since $d^-(y_2, C) = 0$.

Let for some $l \in [4, n-2]$ $y_2x_l \in D$ and $d^+(y_2, C[x_2, x_{l-1}]) = 0$. Then $d(y_2, C[x_2, x_{l-1}]) = 0$ and the vertices y_1, x_{l-2} are nonadjacent since $d^+(y_1, C[x_2, x_{n-2}]) = 0$. It is easy to see that there exists a vertex $x_r \in \{x_1, x_2, \ldots, x_{l-3}\}$ such that $x_ry_1 \in D$ and $d(y_1, C[x_{r+1}, x_{l-2}]) = 0$. Thus, we have that $A(R, C[x_{r+1}, x_{l-2}]) = \emptyset$. Notice that $d(y_2) \le n-l+1$ since $d^-(y_2, C) = 0$ and $d(y_2, C[x_2, x_{l-1}]) = 0$. We extend the path $P_0 := C[x_l, x_r]$ with the vertices $x_{r+1}, x_{r+2}, \ldots, x_{l-1}$ as much as possible. Then some vertices $z_1, z_2, \ldots, z_d \in \{x_{r+1}, x_{r+2}, \ldots, x_{l-1}\}$, $d \in [2, l-r-1]$, are not on the extended path P_e. Therefore, by Lemma 3.2 for $z_i \ne x_{l-1}$ we have, $d(z_i) \le n+d-3$. Now by condition A_0 and (16) we obtain

$$3n-2 \le d(y_2) + d(z_i) + d^-(y_1) + d^+(y_2) < 3n-3,$$

a contradiction.

Let now $d^+(y_2, \{x_2, x_3, \ldots, x_{n-2}\}) = 0$. Then $d(y_2) = 2$, $d(x_2) \le 2n-6$ and the vertices x_2, y_2 are nonadjacent. By condition A_0 we have

$$3n-2 \le d(y_2) + d(x_2) + d^-(y_1) + d^+(y_2) < 3n-3,$$

a contradiction. Claim 12 is proved. □

Claim 13: *For any* $i \in [1, n-2]$ *and* $f \in [1, 2]$ *the following holds*
i) $d^-(y_f, \{x_{i-1}, x_i\}) \le 1$ *and ii)* $d^+(y_f, \{x_{i-1}, x_i\}) \le 1$.

Proof of Claim 13: The proof is by contradiction. By Claim 12, $y_2y_1 \in D$. Without loss of generality, we may assume that $x_{n-3}y_1$, $x_{n-2}y_1 \in D$ and y_1, x_1 are nonadjacent. It is easy to see that $d^+(y_2, \{x_1, x_2\}) = 0$, $y_1x_{n-2} \notin D$ and $y_1x_2 \notin D$ (for otherwise, if $y_1x_2 \in D$, then $x_{n-2}y_1x_2x_3\dots x_{n-3}x_{n-2}$ is a cycle of length $n-2$ for which $\langle\{y_2,x_1\}\rangle$ is not strong, a contradiction to Claim 12). Therefore, $A(R\to\{x_1,x_2\})=\emptyset$. Again using Claim 12, it is not difficult to check that $n\ge 6$.

Assume first that $A(R\to\{x_3,x_4,\dots,x_{n-3}\})\neq\emptyset$. Now let x_l, $l\in[3,n-3]$, be the first vertex after x_2 that $d^-(x_l,R)\ge 1$. Then $A(R\to\{x_1,x_2,\dots,x_{l-1}\})=\emptyset$ since $A(R\to\{x_1,x_2\})=\emptyset$ (in particular, $d^-(x_{l-1},R)=\emptyset$). From the minimality of l and $x_{n-2}y_1\in D$ it follows that there is a vertex $x_r\in\{x_{n-2},x_1,x_2,\dots,x_{l-2}\}$ such that $d^+(x_r,R)\ge 1$ and $A(\{x_{r+1},x_{r+2},\dots,x_{l-2}\},R)=\emptyset$ (if $x_r=x_{n-2}$, then $x_{r+1}=x_1$). This contradicts Claim 3(i) since $d^-(x_{l-1},R)=0$ and $\langle R\rangle$ is strong.

Assume next that $A(R\to\{x_3,x_4,\dots,x_{n-3}\})=\emptyset$. This together with $A(R\to\{x_1,x_2\})=\emptyset$ gives that $A(R\to\{x_1,x_2,\dots,x_{n-3}\})=\emptyset$. From this, since D is strong and $y_1x_{n-2}\notin D$, it follows that $y_2x_{n-2}\in D$. Then $x_{n-3}y_2\notin D$ and $x_{n-4}y_1\notin D$. Now using Claim 12 we obtain that $d(y_2,\{x_{n-4},x_{n-3}\})=0$. Since $d^-(x_{n-3},R)=0$ and $y_2x_{n-2}\in D$, from Claim 3(i) it follows that $d^+(x_{n-4},R)=0$. Therefore, $d(x_{n-4},R)=0$. If $A(\{x_1,x_2,\dots,x_{n-5}\}\to R)\neq\emptyset$, then there is a vertex x_r with $r\in[1,n-5]$ such that $d^+(x_r,R)\ge 1$ and $A(R,\{x_{r+1},x_{r+2},\dots,x_{n-4}\})=\emptyset$ ($n\ge 6$) which contradicts Claim 3(i), since $y_2x_{n-2}\in D$ and $d^-(x_{n-3},R)=0$. Assume therefore that $A(\{x_1,x_2,\dots,x_{n-4}\}\to R)=\emptyset$. Thus, we have that $A(\{x_1,x_2,\dots,x_{n-4}\},R)=\emptyset$ and $d^-(x_{n-3},R)=0$. Then $d(y_1)=4$, $d(y_2)\le 4$ and $d(x_1)\le 2n-6$. From this it follows that $d(y_1)+d(x_1)\le 2n-2$ and $d(y_2)+d(x_1)\le 2n-2$ which contradicts Lemma 3.5. This contradiction proves that $d^-(y_f,\{x_{i-1},x_i\})\le 1$ for all $i\in[1,n-2]$ and $f\in[1,2]$. Similarly, one can show that $d^+(y_f,\{x_{i-1},x_i\})\le 1$. Claim 13 is proved. □

Claim 14: *If* $x_iy_f\in D$ *(respectively,* $y_fx_i\in D$*), then* $d(y_f,\{x_{i+2}\})\neq 0$ *(respectively,* $d(y_f,\{x_{i-2}\})\neq 0$*), where* $i\in[1,n-2]$ *and* $f\in[1,2]$*.*

Proof of Claim 14: Suppose that the claim is not true. By Claim 12, $y_2y_1\in D$. Without loss of generality, we may assume that $x_{n-2}y_1\in D$ and $d(y_1,\{x_2\})=0$, i.e., the vertices y_1 and x_2 are nonadjacent. Claim 13 implies that the vertices y_1,x_1 also are nonadjacent. Thus, $d(y_1,\{x_1,x_2\})=0$. Note that $y_2x_2\notin D$ and hence, $d^-(x_2,R)=0$. Now it is not difficult to cheek that if $n=5$, then $d(y_1)+d(x_1)\le 8$ and $d(y_1)+d(x_2)\le 8$, a contradiction to Lemma 3.5.

Assume, therefore, that $n\ge 6$ and consider the following cases.

Case 14.1. $A(R\to\{x_3,x_4,\dots,x_{n-3}\})\neq\emptyset$.

Then there is a vertex x_l with $l\in[3,n-3]$ such that $d^-(x_l,R)\ge 1$ and $A(R\to\{x_2,x_3,\dots,x_{l-1}\})=\emptyset$ since $d(y_1,\{x_1,x_2\})=d^-(x_2,R)=0$. We now consider the case $l=3$ and the case $l\ge 4$ separately.

Assume that $l=3$. Then $y_2x_3\in D$ or $y_1x_3\in D$.

Let $y_2x_3\in D$. Then the vertices y_2,x_2 are nonadjacent. Since the vertices y_1,x_2 are nonadjacent Claim 12 implies that $x_1y_2\notin D$ (for otherwise $x_{n-2}x_1y_2x_3\dots x_{n-4}x_{n-2}$ is a cycle of length $n-2$ which does not contain the vertices y_1,x_2 and $\langle\{y_1,x_2\}\rangle$ is not strong, a contradiction to Claim 12). This contradicts Claim 3(ii) because of $d(x_2,R)=0$ and $d^+(x_1,R)=0$.

Let now $y_1x_3\in D$ and $y_2x_3\notin D$. Then it is easy to see that $x_1y_2\notin D$ and $y_2x_2\notin D$. From this and Claim 12 implies that neither x_1 nor x_2 can be inserted into $C[x_3,x_{n-2}]$. Notice that if $x_2y_2\in D$, then $x_{n-2}x_2\notin D$, and if $y_2x_1\in D$, then $x_1x_3\notin D$. Now using Lemma 3.2, we obtain that $d(y_1)$, $d(x_1)$ and $d(x_2)\le n-1$ since $d(y_1,\{x_1,x_2\})=0$. Therefore

$$d(y_1)+d(x_1)\le 2n-2 \quad\text{and}\quad d(y_1)+d(x_2)\le 2n-2,$$

which contradicts Lemma 3.5 since y_1, x_1 and y_1, x_2 are two distinct pairs of nonadjacent vertices. This contradiction completes the discussion of Case 14.1 when $l = 3$.

Assume that $l \geq 4$. Let $y_g x_l \in D$, where $g \in [1, 2]$. Then, by the minimality of l, the vertices y_g, x_{l-1} are nonadjacent, $y_{3-g}x_{l-1} \notin D$ and $x_{l-2}y_{3-g} \notin D$. Hence, by Claim 12 we get that $x_{l-2}y_g \notin D$. From the minimality of l and $d^-(x_2, R) = 0$ (for $l = 4$) it follows that x_{l-2} is not adjacent to y_1 and y_2, i.e., $d(x_{l-2}, R) = 0$. This together with $d^-(x_2, R) = d^-(x_{l-1}, R) = 0$, the minimality of l and Claim 3(i) implies that

$$A(R, \{x_2, x_3, \ldots, x_{l-2}\}) = \emptyset \quad \text{and} \quad d^+(x_1, R) = 0$$

(if $d^+(x_1, R) \geq 1$, then $l \geq 5$ and there is an x_r with $r \in [1, l-3]$ such that $d^+(x_{l-1}, R) = 0$ and $A(R, C[x_{r+1}, x_{l-3}]) = \emptyset$ but this contradicts Claim 3(i)). If $d^-(x_1, R) = 0$ or $d^+(x_{l-1}, R) = 0$, then $d(x_1, R) = 0$ or $d(x_{l-1}, R) = 0$, respectively. This together with $A(R, C[x_2, x_{l-2}]) = \emptyset$ contradicts Claim 3 since $d^+(x_{n-2}, R) \geq 1$ and $d^-(x_l, R) \geq 1$. Assume, therefore, that $d^-(x_1, R) \geq 1$ and $d^+(x_{l-1}, R) \geq 1$. It follows that $y_2x_1 \in D$ since $y_1x_1 \notin D$.

Assume first that $y_g = y_2$. Then $x_{l-1}y_1 \in D$. Since $y_1x_{l-1} \notin D$, $x_1y_2 \notin D$ and

$$d(y_1, C[x_1, x_{l-2}]) = d(y_2, C[x_2, x_{l-1}]) = 0$$

using Lemma 3.2(ii) we obtain that

$$d(y_1) = d(y_1, \{y_2\}) + d(y_1, C[x_{l-1}, x_{n-2}]) \leq n - l + 2 \quad \text{and}$$

$$d(y_2) = d(y_2, \{y_1\}) + d(y_2, C[x_l, x_1]) \leq n - l + 2. \tag{17}$$

Now we extend the path $P_0 := C[x_l, x_{n-2}]$ with the vertices $x_1, x_2, \ldots, x_{l-1}$ as much as possible. Then some vertices $z_1, z_2, \ldots, z_d \in \{x_1, x_2, \ldots, x_{l-1}\}$, $d \in [2, l-1]$, are not on the extended path P_e since otherwise P_{e-1} or P_e together with the arcs $x_{n-2}y_1, y_1y_2$ and y_2x_l forms a cycle of length $n-1$. Therefore, by Lemma 3.2, we have that $d(z_i, C) \leq n + d - 3$. If there is a $z_i \notin \{x_1, x_{l-1}\}$, then $d(z_i) \leq n + d - 3$ and by (17),

$$d(z_i) + d(y_1) \leq 2n - 2 \quad \text{and} \quad d(z_i) + d(y_2) \leq 2n - 2,$$

which contradicts Lemma 3.5 since z_i is not adjacent to y_1 and y_2. Therefore, assume that $\{z_1, z_2\} = \{x_1, x_{l-1}\}$ ($d = 2$). Then P_e ($e = l - 3 \geq 1$) is an (x_l, x_{n-2})-path with vertex set $V(C) - \{x_1, x_{l-1}\}$. Thus, we have that $y_2P_ey_1y_2$ is a cycle of length $n-2$. Therefore, by Claim 12, $x_1x_{l-1} \in D$, and hence, $x_1x_{l-1}P_{e-1}y_1y_2x_1$ is a cycle of length $n-1$, which contradicts the initial supposition that D contains no cycle of length $n-1$.

Assume second that $y_g = y_1$. Then by the above observation we conclude that $x_{l-1}y_2 \in D$ and $d(y_1, C[x_1, x_{l-1}]) = 0$. Using Lemma 3.2, we obtain that for this case (17) also holds, since $x_1y_2 \notin D$ and $y_2x_{l-1} \notin D$. Again we extend the path $C[x_l, x_{n-2}]$ with vertices $x_1, x_2, \ldots, x_{l-1}$ as much as possible. Then some vertices $z_1, z_2, \ldots, z_d \in \{x_1, x_2, \ldots, x_{l-1}\}$, $d \in [1, l-1]$, are not on the extended path P_e. Similar to the first case when $y_g = y_2$, we will obtain that $z_i \notin \{x_2, x_3, \ldots, x_{l-2}\}$ (i.e., $z_i = x_1$ or $z_i = x_{l-1}$) and $d(z_i) \leq n + d - 2$. Notice that $C' := y_1P_ey_1$ is a cycle of length $n - d - 1$ with vertex set $V(C) \cup \{y_1\} - \{z_1, z_d\}$. From Claim 12 it follows that $d = 2$, i.e., $\{z_1, z_d\} = \{x_1, x_{l-1}\}$ (since $x_1y_2 \notin D$ and $yx_{l-1} \notin D$). Now from $l \geq 4$, $d = 2$, (17) and $d(z_i) \leq n + d - 2$ we obtain that

$$d(y_1) + d(x_1) \leq 2n - 2 \quad \text{and} \quad d(y_1) + d(x_{l-1}) \leq 2n - 2,$$

which contradicts Lemma 3.5, since y_1, x_1 and y_1, x_{l-1} are two distinct pairs of nonadjacent vertices.

Case 14.2. $A(R \to \{x_3, x_4, \ldots, x_{n-3}\}) = \emptyset$.

Then $A(R \to \{x_{n-2}, x_1\}) \neq \emptyset$, since $d^-(x_2, R) = 0$ and D is strong, and y_1, x_{n-3} are nonadjacent (by Claim 13). For this case we distinguish three subcases.

Subcase 14.2.1. $y_2x_{n-2} \in D$.

Then, using Claim 13, it is easy to see that x_{n-3}, y_2 are nonadjacent. Therefore, $d(x_{n-3}, R) = 0$. This together with $y_2x_{n-2} \in D$ and Claim 3 implies that $A(\{x_1, x_2, \ldots, x_{n-3}\} \to R) = \emptyset$. Therefore, $d(R, \{x_2, x_3, \ldots, x_{n-3}\}) = \emptyset$ and $d(y_1)$, $d(y_2) \le 4$ (since $y_2x_1 \notin D$ by Claim 13) and $d(x_{n-3}) \le 2n-6$. From this it follows that $d(y_1) + d(x_{n-3}) \le 2n-2$ and $d(y_2) + d(x_{n-3}) \le 2n-2$, which contradicts Lemma 3.5.

Subcase 14.2.2. $y_2x_{n-2} \notin D$ *and* $y_2x_1 \in D$.

Then using Claim 13 it is easy to see that y_2 and x_{n-2} are nonadjacent.

Let $x_{n-3}y_2 \in D$. Then $y_1x_{n-2} \in D$ (by Claim 12). Using Claims 12 and 13 we obtain that x_{n-4} is not adjacent to y_1 and y_2. Since $d^-(x_{n-3}, R) = 0$ and $y_1x_{n-2} \in D$, from Claim 3(i) it follows that $A(\{x_1, x_2, \ldots, x_{n-4}\} \to R) = \emptyset$ and $A(R, C[x_2, x_{n-4}]) = \emptyset$. Therefore and $d(y_1) = d(y_2) = 4$ and $d(x_2) \le 2n-6$. From these it follows that

$$d(y_1) + d(x_2) \le 2n-2 \quad \text{and} \quad d(y_2) + d(x_2) \le 2n-2,$$

which contradicts Lemma 3.5 since x_2, y_1 and x_2, y_2 are two distinct pairs of nonadjacent vertices.

Let now $x_{n-3}y_2 \notin D$. Then y_2, x_{n-3} are nonadjacent and hence, $d(x_{n-3}, R) = 0$. Now, since $y_1x_{n-2} \in D$ or $d^-(x_{n-2}, R) = 0$ and $y_2x_1 \in D$, from Claim 3 it follows that $A(\{x_2, x_3, \ldots, x_{n-3}\} \to R) = \emptyset$. Therefore

$$d(y_1, C[x_1, x_{n-3}]) = d(y_2, C[x_2, x_{n-2}]) = 0,$$

$d(y_1) \le 4$, $d(y_2) \le 4$ and $d(x_2) \le 2n-6$. This contradicts Lemma 3.5 since x_2, y_1 and x_2, y_2 are two distinct pairs of nonadjacent vertices.

Subcase 14.2.3. $y_2x_{n-2} \notin D$ *and* $y_2x_1 \notin D$.

Then $y_1x_{n-2} \in D$ (since D is strong), the vertex y_1 is not adjacent to the vertices x_{n-3}, x_{n-4} and $x_{n-4}y_2 \notin D$, i.e., the vertices y_2, x_{n-4} also are nonadjacent. Using Claim 3, we can assume that $A(C[x_1, x_{n-4}] \to R) = \emptyset$. Therefore, $d(y_1) = 4$, $d(y_2) \le 3$ and $d(x_1) \le 2n-6$. This contradicts Lemma 3.5 since x_1 is not adjacent to y_1 and y_2. This completes the proof of Claim 14. □

Claim 15: *If* $x_iy_f \in D$ *and the vertices* y_f, x_{i+1} *are nonadjacent, then the vertices* x_{i+1}, y_{3-f} *are adjacent, where* $i \in [1, n-2]$ *and* $f \in [1, 2]$.

Proof of Claim 15: Without loss of generality, we may assume that $x_i = x_{n-2}$ (i.e., $x_{i+1} = x_1$) and $y_f = y_1$. Suppose, on the contrary, that x_1, y_2 are nonadjacent. From Claims 12 and 14 it follows that $y_1x_2 \notin D$ and $x_2y_1 \in D$. Therefore, $A(R \to \{x_1, x_2\}) = \emptyset$. If $n = 5$, then $x_2y_1, x_3y_1 \in D$ which contradicts Claim 13. Assume, therefore, that $n \ge 6$. As D is strong, there is a vertex x_l with $l \in [3, n-2]$ such that $d^-(x_l, R) \ge 1$ (say $y_gx_l \in D$) and $A(R \to C[x_1, x_{l-1}]) = \emptyset$. Then the vertices x_{l-1}, y_g are nonadjacent and $d(x_{l-2}, R) = 0$ (by $x_{l-2}y_{3-g} \notin D$ and by Claim 12). Now, since $x_{n-2}y_1$ and $x_2y_1 \in D$, there exists a vertex $x_r \in C[x_{n-2}, x_{l-3}]$ (if $l = 3$, then $x_{n-2} = x_{l-3}$) such that $d^+(x_r, R) \ge 1$ and

$A(R, C[x_{r+1}, x_{l-2}]) = \emptyset$. This contradicts Claim 3 since $d^-(x_{l-1}, R) = 0$ and $d^-(x_l, R) \ge 1$. Claim 15 is proved.

Claim 16: *If* $x_iy_j \in D$, *where* $i \in [1, n-2]$ *and* $j \in [1, 2]$, *then* $y_jx_{i+2} \in D$.

Proof of Claim 16: Without loss of generality, we may assume that $x_i = x_{n-2}$ and $y_j = y_1$. Suppose that the claim is not true, that is $x_{n-2}y_1 \in D$ and $y_1x_2 \notin D$. Then, by Claims 13 and 14, the vertices y_1, x_1 are nonadjacent, $x_2y_1 \in D$ (hence, $n \ge 6$) and y_1, x_3 are also nonadjacent. From this, by Claim 15 we obtain that the vertex y_2 is adjacent to the vertices x_1 and x_3. Therefore either $y_2x_3 \in D$ or $x_3y_2 \in D$.

Case 16.1. $y_2x_3 \in D$.

Then x_2, y_2 are nonadjacent (by Claim 13), $x_2x_1 \in D$ and $x_1y_2 \notin D$ by Claim 12 (for otherwise D would have a cycle C' of length $n-2$ for which $\langle V(D) - V(C')\rangle$ is not strong). Notice that $y_2x_1 \in D$ (by Claim 15). Since neither y_1 nor y_2 can be inserted into C, $y_1x_2 \notin D$ and y_1, x_1 are nonadjacent (respectively, $x_1y_2 \notin D$ and y_2, x_2 are nonadjacent) using Lemma 3.2(ii), we obtain that

$$d(y_1) \le n-1 \quad \text{and} \quad d(y_2) \le n-1. \tag{18}$$

It is not difficult to see that $x_{n-2}x_2 \notin D$ and $x_1x_3 \notin D$ (for otherwise D contains a cycle of length $n-1$). Therefore, since neither x_1 nor x_2 cannot be inserted into $C[x_3, x_{n-2}]$ (otherwise we obtain a cycle of length $n-1$), again using Lemma 3.2(ii), we obtain

$$d(x_1) \le n-1 \quad \text{and} \quad d(x_2) \le n-1. \tag{19}$$

It is easy to check that $n \ge 7$.

Remark: *Observe that from (18), (19) and Lemma 3.5 it follows that if* $x_i \ne x_1$ *and* y_1, x_i *are nonadjacent or* $x_i \ne x_2$ *and* x_i, y_2 *are nonadjacent, then* $d(x_i) \ge n$.

Assume first that $d^+(y_1, C[x_4, x_{n-2}]) \ge 1$. Let x_l, $l \in [4, n-2]$, be the first vertex after x_3 that $y_1x_l \in D$. Then the vertices y_1 and x_{l-1} are nonadjacent. Therefore, y_1 and x_{l-2} are adjacent (by Claim 14) and hence, $x_{l-2}y_1 \in D$ because of $x_2y_1 \in D$ and minimality of l ($l-1 \ne 4$ by Claim 14, since $x_2y_1 \in D$). Since x_{l-1} cannot be inserted into $C[x_l, x_{l-2}]$, using Lemma 3.2 and the above Remark, we obtain that $d(x_{l-1}) = n$ and hence, $d(y_1) = n-1$ (by Lemma 3.5). This together with $d(y_1, \{x_1, x_2, x_3, y_2\}) = 3$ implies that $d(y_1, C[x_4, x_{n-2}]) = n-4$. Again using Lemma 3.2, we obtain that $y_1x_4 \in D$ (since $|C[x_4, x_{n-2}]| = n-5$). Thus, $y_1C[x_4, x_2]y_1$ is a cycle of length $n-2$. Therefore, $x_3y_2 \in D$ (by Claim 12), $y_1x_5 \notin D$ and the vertices y_2, x_4 are nonadjacent (by Claim 13). From $y_1x_5 \notin D$ (by Lemma 3.2) we obtain that $d(y_1, C[x_5, x_{n-2}]) \le n-6$. Therefore $x_4y_1 \in D$ and $d(y_1, C[x_5, x_{n-2}]) = n-6$. Now it is easy to see that y_1, x_5 are nonadjacent (by Claim 13) and y_2, x_5 are adjacent (by Claim 14). Therefore, $d(y_1, C[x_6, x_{n-2}]) = n-6$ and $y_1x_6 \in D$ (by Lemma 3.2), $y_2x_5, x_5y_2 \in D$ (by Claim 12), $y_1x_7 \notin D$ (by Claim 13). One readily sees that, by continuing the above procedure, we eventually obtain that n is even and

$$N^-(y_1) = \{y_2, x_2, x_4, x_6, \ldots, x_{n-2}\}, \quad N^+(y_1) = \{y_2, x_4, x_6, \ldots, x_{n-2}\},$$

$$N^-(y_2) = \{y_1, x_3, x_5, \ldots, x_{n-3}\}, \quad N^+(y_2) = \{y_1, x_1, x_3, x_5, \ldots, x_{n-3}\}.$$

From Claim 12 it follows that $x_ix_{i-1} \in D$ for all $i \in [4, n-2]$ and $x_2x_1 \in D$. It is easy to see that $x_1x_3 \notin D$ and $x_3x_5 \notin D$. Therefore, since x_3 cannot be inserted into $C[x_5, x_1]$, by Lemma 3.2, we have $d(x_3, C[x_5, x_1]) \le n-6$. This together with $d(x_3) \ge n$ (by Remark) implies that $d(x_3,\{x_2,x_4,y_2\}) = 6$. In particular, $x_3x_2 \in D$. Now we consider the vertex x_{n-2}. Note that $d(x_{n-2}) \geq n$ (by Remark), $x_{n-2}x_2 \notin D$ and $x_{n-4}x_{n-2} \notin D$. From this it is not difficult to see that $d(x_{n-2},C[x_2,x_{n-4}]) \leq n-6$ and $x_1x_{n-2} \in D$. It follows that $x_{n-2}x_{n-3}\ldots x_4x_3y_2x_1x_{n-2}$ is a cycle of length $n-2$, which does not contain the vertices y_1 and x_2. This contradicts Claim 12, since $y_1x_2 \notin D$ (by our supposition), i.e., $\langle\{y_1,x_2\}\rangle$ is not strong.

Assume next that $d^+(y_1,C[x_4,x_{n-2}]) = 0$. Then from Claims 13 and 14 it follows that

$$N^-(y_1) = \{y_2,x_2,x_4,\ldots,x_{n-2}\} \quad \text{and} \quad N^+(y_1) = \{y_2\}. \tag{20}$$

By Claim 15 we have that the vertex y_2 is adjacent to each vertex $x_i \in \{x_1,x_3,\ldots,x_{n-3}\}$. It is easy to see that $x_{n-3}y_2 \notin D$ and hence, $y_2x_{n-3} \in D$ (for otherwise if $x_{n-3}y_2 \in D$, then $y_2C[x_1,x_{n-3}]y_2$ is a cycle of length $n-2$, but $\langle\{x_{n-2},y_1\}\rangle$ is not strong, a contradiction to Claim 12). By an argument similar to that in the proof of (20) we deduce that

$$N^+(y_2) = \{y_1,x_1,x_3,\ldots,x_{n-3}\} \quad \text{and} \quad N^-(y_2) = \{y_1\}.$$

Thus we have that $y_1y_2C[x_5,x_2]y_1$ is a cycle of length $n-2$ and x_3 cannot be inserted into $C[x_5,x_2]$. Therefore, by Lemma 3.2(ii), $d(x_3,C[x_5,x_2]) \leq n-4$ since $x_3x_5 \notin D$. This together with $d(x_3,\{x_4,y_1,y_2\}) \leq 3$ implies that $d(x_3) \leq n-1$ which contradicts the above Remark that $d(x_3) \geq n$.

Case 16.2. $y_2x_3 \notin D$.

Then, as noted above, $x_3y_2 \in D$. Therefore $d(y_2,\{x_2,x_4\}) = 0$ (by Claim 13 and $y_2x_2 \notin D$), $y_1x_4 \notin D$ (by Claim 12), $x_4y_1 \in D$ (by Claim 15), the vertices x_5,y_1 are nonadjacent and the vertices y_2,x_5 are adjacent (by Claim 15). Since $x_3y_2 \in D$, $y_1x_4 \notin D$ and y_2,x_5 are adjacent, from Claim 12 it follows that $y_2x_5 \notin D$ and $x_5y_2 \in D$. For the same reason, we deduce that

$$N^-(y_1) = \{y_2,x_2,x_4,\ldots,x_{n-2}\} \quad N^-(y_2) = \{y_1,x_1,x_3,\ldots,x_{n-3}\} \quad \text{and} \quad A(R \to V(C)) = \emptyset,$$

which contradicts that D is strong. This contradiction completes the proof of Claim 16. □

We will now complete the proof of Theorem by showing that D is isomorphic to $K^*_{n/2,n/2}$. Without loss of generality, we assume that $x_{n-2}y_1 \in D$. Then using Claims 12, 13, 14 and 16 we conclude that y_1,x_1 are nonadjacent (Claim 13), $y_1x_2 \in D$ (Claim 16), $x_1y_2,y_2x_1 \in D$ (Claim 12), x_2,y_2 also are nonadjacent (Claim 13), $y_2x_3 \in D$ (Claim 16) and $x_2y_1 \in D$ (Claim 12). By continuing this procedure, we eventually obtain that n is even and

$$N^+(y_1) = N^-(y_1) = \{y_2,x_2,x_4,\ldots,x_{n-2}\} \quad \text{and} \quad N^+(y_2) = N^-(y_2) = \{y_1,x_1,x_3,\ldots,x_{n-3}\}.$$

If $x_ix_j \in D$ for some $x_i,x_j \in \{x_1,x_3,\ldots,x_{n-3}\}$, then clearly $|C[x_i,x_j]| \geq 5$ and $x_ix_jx_{j+1}\ldots x_{i-1}y_1x_{i+1}\ldots x_{j-2}y_2x_i$ is a cycle of length $n-1$, contrary to our assumption. Therefore, $\{y_1,x_1,x_3,\ldots,x_{n-3}\}$ is an independent set of vertices. For the same reason $\{y_2,x_2,x_4,\ldots,x_{n-2}\}$ also is an independent set of vertices. Now from the condition A_0 it follows that D is isomorphic to $K^*_{n/2,n/2}$. This completes the proof of Theorem 1.10. □

5. Concluding Remarks

A Hamiltonian bypass in a digraph is a subdigraph obtained from a Hamiltonian cycle of D by reversing one arc.

Using Theorem 1.10, the first author has proved that if a strong digraph D of order $n \geq 4$ satisfies the condition A_0, then D contains a Hamiltonian bypass or D is isomorphic to one tournament of order 5.

References

[1] J. Bang-Jensen and G. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer, 2000.

[2] J. Bang-Jensen, G. Gutin and H. Li, “Sufficient conditions for a digraph to be Hamiltonian", *Journal of Graph Theory*, vol. 22, no. 2, pp. 181-187, 1996.

[3] J. Bang-Jensen, Y. Guo and A. Yeo, “A new sufficient condition for a digraph to be Hamiltonian", *Discrete Applied Mathematics*, vol. 95, pp. 61-72, 1999.

[4] J. A. Bondy, “Basic graph theory: paths and circuits", *In Handbook of combinatorics*, vol. 1, no. 2, Elsevier, Amsterdam, 1995.

[5] J. A. Bondy and C. Thomassen, “A short proof of Meyniel's theorem", *Discrete Mathematics*, vol. 19, no. 1, pp. 195-197, 1977.

[6] S. Kh. Darbinyan, “Pancyclic and panconnected digraphs", *Ph. D. Thesis, Institute Mathematici Akademy Nauk BSSR, Minsk, 1981* (see also, Pancyclicity of digraphs with the Meyniel condition, *Studia Scientiarum Mathematicarum Hungarica*, (in Russian), vol.20, no. 1-4, 95-117, 1985.)

[7] S. Kh. Darbinyan, “A sufficient condition for the Hamiltonian property of digraphs with large semidegrees", *Akademy Nauk Armyan. SSR Doklady*, vol. 82, no. 1, pp. 6-8, 1986. (see also arXiv: 1111.1843v1 [math.CO] 8 Nov 2011).

[8] S. Kh. Darbinyan, “On the pancyclicity of digraphs with large semidegrees", *Akademy Nauk Armyan. SSR Doklady*, vol. 83, no.4, pp. 99-101, 1986. (see also arXiv: 1111.1841v1 [math.CO] 8 Nov 2011).

[9] S.Kh. Darbinyan and I.A. Karapetyan, “On longest non-Hamiltonian cycles in digraphs with the conditions of Bang-Jensen, Gutin and Li", *Preprint available at htte: arXiv 1207.5643v2 [math.CO]*, 20 Sep 2012.

[10] S. Kh. Darbinyan and I.A. Karapetyan, “A note on long non-Hamiltonian cycles in one class of digraphs", *Preprint available at htte: arXiv 1209.4456v1 [math.CO]*, 20 Sep 2012.

[11] R. Häggkvist and C. Thomassen, “On pancyclic digraphs", *Journal Combinatorial Theory Ser. B*, vol. 20, no. 1, pp. 20-40, 1976.

[12] A. Ghouila-Houri, “Une condition suffisante d'existence d'un circuit hamiltonien", *Comptes Rendus de I' Academie des Sciences Paris Ser. A-B*, vol. ??, no. 25, pp. 495-497, 1960.

[13] G. Gutin, “Characterizations of vertex pancyclic and pancyclic ordinary complete multipartite digraphs", *Discrete Mathematics*, vol.141, pp. 153-162, 1995.

[14] Y. Manoussakis, “Directed Hamiltonian graphs", *Journal of Graph Theory*, vol. 16, no. 1, pp. 51-59, 1992.

[15] **M. Meyniel, \Une condition suffisante d'existence d'un circuit hamiltonien dans un graphe oriente",** *Journal Combinatorial Theory Ser. B*, **vol. 14, pp. 137-147, 1973.**

[16] **C. Thomassen, \An Ore-type condition implying a digraph to be pancyclic",** *Discrete Mathematics*, **vol.19, no 1, pp.85-92, 1977.**

[17] **C. Thomassen, \Long cycles in digraphs",** *Proceeding London Mathematical Society*, **vol. 3, no. 42, pp. 231-251, 1981.**

[18] **D. R. Woodall, "Sufficient conditions for circuits in graphs",** *Proceeding London Mathematical Society*, **no. 24, pp. 739-755, 1972.**

Submitted 16.10.2014, accepted 27.01.2015.

Կողմնորոշված համիլտոնյան գրաֆների նախահամիլտոնյան ցիկլերի մասին

Ս.Դարբինյան և Ի. Կարապետյան

Ամփոփում

Կողմնորոշված գրաֆի կողմնորոշված ցիկլը, որն անցնում է նրա բոլոր գագաթներով, բացի մեկից, կոչվում է նախահամիլտոնյան ցիկլ: Ներկա աշխատանքում ապացուցված է, որ եթե կողմնորոշված գրաֆը բավարարում է Մանոուսակիսի համիլտնյանության բավարար պայմանին (J. of Graph Theory 16(1) (1992) 51-59), ապա այն պարունակում է նախահամիլտոնյան ցիկլ, բացի այն դեպքից, երբ այդ գրաֆը իզոմորֆ է երկմաս հավասարակշռված կողմնորոշված լրիվ գրաֆին:

О предгамильтоновых контурах в гамильтоновых ориентированных графах

С. Дарбинян и И. Карапетян

Аннотация

Ориентированный контур, который содержит все вершины ориентированного графа (орграфа), называется предгамильтоновым контуром. В работе доказано, что любой орграф, который удовлетворяет достаточному условию гамильтоновости орграфов Маноусакиса (J. of Graph Theory 16(1) (1992) 51-59), содержит предгамильтоновый контур или является двудольным балансированным полным орграфом.

Mathematical Problems of Computer Science 43, 26{31, 2015.

On k-| nded Spanning and Dominating Trees

Zhora G. | ikoghosyan⁄

Institute for Informatics and Automation Problems of NAS RA
e-mail: zhora@ipia.sci.am

Abstract

A tree with at most k leaves is called a k-ended tree. Let t_k be the order of a largest k-ended tree in a graph. A tree T of a graph G is said to be dominating if V(G ¡ T) is an independent set of vertices. The minimum degree sum of any pair (triple) of nonadjacent vertices in G will be denoted by ¾$_2$ (¾$_3$). The earliest result concerning spanning trees with few leaves (by the author, 1976) states: (⁄) if G is a connected graph of order n with ¾$_2$ ¸ n ¡ k + 1 for some positive integer k, then G has a spanning k-ended tree. In this paper we show: (i) the connectivity condition in (⁄) can be removed; (ii) the condition ¾$_2$ ¸ n ¡ k + 1 in (⁄) can be relaxed by replacing n with t_{k+1}; (iii) if G is a connected graph with ¾$_3$ ¸ t_{k+1} ¡ 2k + 4 for some integer k ¸ 2, then G has a dominating k-ended tree. All results are sharp.

Keywords: Hamilton cycle, Hamilton path, Dominating cycle, Dominating path, Longest path, k-ended tree.

1. Introduction

Throughout this article we consider only finite undirected graphs without loops or multiple edges. The set of vertices of a graph G is denoted by V(G) and the set of edges by E(G). | good reference for any undefined terms is in [1].

For a graph G, we use n, ± and ® to denote the order (the number of vertices), the minimum degree and the independence number of G, respectively. For a subset S µ V(G) we denote by G[S] the subgraph of G induced by S. If ® ¸ k for some integer k, let ¾$_k$ be the minimum degree sum of an independent set of k vertices; otherwise we let ¾$_k$ = +1.

If Q is a path or a cycle in a graph G, then the order of Q, denoted by jQj, is jV(Q)j. Each vertex and edge in G can be interpreted as simple cycles of orders 1 and 2, respectively. The graph G is | amiltonian if G contains a | amilton cycle, i.e. a cycle containing every vertex of G. | cycle C of G is said to be dominating if V(G ¡ C) is an independent set of vertices.

We write a cycle Q with a given orientation by $\overset{¡!}{Q}$. For x; y 2 V(Q), we denote by x$\overset{¡!}{Q}$y the subpath of Q in the chosen direction from x to y. For x 2 V(Q), we denote the successor and the predecessor of x on $\overset{¡!}{Q}$ by x^+ and $x^¡$, respectively.

| vertex of degree one is called an end-vertex, and an end-vertex of a tree is usually called a leaf. The set of end-vertices of G is denoted by End(G). For a positive integer k, a

⁄G.G. Nicoghossian (up to 1997)

tree T is said to be a k-ended tree if $|End(T)| \le k$. A Hamilton path is a spanning 2-ended tree. A Hamilton cycle can be interpreted as a spanning 1-ended tree. In particular, K_2 is Hamiltonian and a 1-ended tree. We denote by t_k the order of a largest k-ended tree in G. By the definition, t_1 is the order of a longest cycle, and t_2 is the order of a longest path in G.

Our starting point is the earliest sufficient condition for a graph to be Hamiltonian due to Dirac [2].

Theorem A ([2]): Every graph with $\delta \ge \frac{n}{2}$ is Hamiltonian.

In 1960, Ore [3] improved Theorem A by replacing the minimum degree δ with the arithmetic mean $\frac{1}{2}\sigma_2$ of two smallest degrees among pairwise nonadjacent vertices.

Theorem B ([3]): Every graph with $\sigma_2 \ge n$ is Hamiltonian.

The analog of Theorem B for Hamilton paths follows easily.

Theorem C ([3]): Every graph with $\sigma_2 \ge n - 1$ has a Hamilton path.

In 1971, Las Vergnas [4] gave a degree condition that guarantees that any forest in G of limited size and with a limited number of leaves can be extended to a spanning tree of G with a limited number of leaves in an appropriate sense. As a corollary, this result implies a degree sum condition for the existence of a tree with at most k leaves including Theorem B and Theorem C as special cases for $k = 1$ and $k = 2$, respectively.

Theorem D ([4], [5], [6]): If G is a connected graph with $\sigma_2 \ge n - k + 1$ for some positive integer k, then G has a spanning k-ended tree.

However, Theorem D was first openly formulated and proved in 1976 by the author [6] and was reproved in 1998 by Broersma and Tuinstra [5]. Moreover, the full characterization of connected graphs without spanning k-ended trees was given in [7] when $\sigma_2 \ge n - k$ including the well-known characterization of connected graphs without Hamilton cycles when $\sigma_2 \ge n - 1$. This particular result was reproved in 1980 by Nara Chie [8].

The next two results on this subject are not included in the recent survey paper [9]. We call a graph G hypo-k-ended if G has no spanning k-ended tree, but for any $v \in V(G)$, $G - v$ has a spanning k-ended tree.

Theorem E ([10]): For each $k \ge 3$, the minimum number of vertices (edges, faces, respectively) of a simple 3-polytope without a spanning k-ended tree is $8 + 3k$ ($12 + 6k$, $6 + 3k$, respectively).

Theorem F ([11]): For each $n \ge 17k$ and $k \ge 2$, except possible for $n = 17k + 1$, $17k + 2$, $17k + 4$ and $17k + 7$, there exist hypo-k-ended graphs of order n.

In this paper we prove that the connectivity condition in Theorem D can be removed, and the conclusion can be strengthened.

Theorem 1: If G is a graph with $\sigma_2 \ge n - k + 1$ for some positive integer k, then G has a spanning k-ended forest.

Next, we show that Theorem D can be improved by relaxing the condition $\sigma_2 \ge n - k + 1$ to $\sigma_2 \ge t_{k+1} - k + 1$.

Theorem 2: Let **G** be a connected graph with ¾$_2$, t$_{k+1}$ ¡ **k + 1** for some positive integer **k**. Then **G** has a spanning **k**-ended tree.

The graph (± + k)K$_1$ + K$_±$ shows that the bound t$_{k+1}$ ¡ k + 1 in Theorem 2 cannot be relaxed to t$_k$ ¡ k + 1. Finally, we give a dominating analog of Theorem | .

Theorem 3: If **G** is a connected graph with ¾$_3$, t$_{k+1}$ ¡ **2k + 4** for some integer **k** , **2**, then **G** has a dominating **k**-ended tree.

The graph (± + k ¡ 1)K$_2$ + K$_{± ¡ 1}$ shows that the bound t$_{k+1}$ ¡ 2k + 4 in Theorem 3 cannot be relaxed to t$_k$ ¡ 2k + 4.

The following corollary follows immediately.

Corollary 1: If **G** is a connected graph with ¾$_3$, **n** ¡ **2k + 4** for some integer **k** , **2**, then **G** has a dominating **k**-ended tree.

The graph (± + k ¡ 1)K$_2$ + K$_{± ¡ 1}$ shows that the bound ¾$_3$, t$_{k+1}$ ¡ 2k + 4 in Theorem 3 cannot be relaxed to ¾$_3$, t$_{k+1}$ ¡ 2k + 3.

2. Proofs

Proof of Theorem 1: Let G be a graph with ¾$_2$, n ¡ k + 1 and let H$_1$; :::; H$_m$ be the connected components of G. Let $\overset{¡!}{P}$ = x$\overset{¡!}{P}$ y be a longest path in H$_1$. If jPj , n ¡ k + 2 then jG ¡ Pj = n ¡ jPj • k ¡ 2, implying that G has a spanning k-ended forest. | ow let jPj • n ¡ k + 1. Since P is extreme, we have N(x) [N(y) µ V(P). Recalling also that ¾$_2$, n ¡ k + 1, we have (by standard arguments) N(x) \ N$^+$(y) 6= ;, implying that G[V(P)] is | amiltonian. Further, if jV(P)j < jV(H$_1$)j then we can form a path longer than P, contradicting the maximality of P. | ence, jV(P)j = jV(H$_1$)j, that is H$_1$ is | amiltonian as well. By a similar argument, H$_i$ is | amiltonian for each i 2 f1; :::; mg and therefore, has a spanning tree with exactly one leaf. Thus, G has a spanning forest with exactly m leaves.

It remains to show that m • k. If m = 1 then G has a spanning 1-ended tree and therefore, has a spanning k-ended tree. Let m , 2 and let x$_i$ 2 V(H$_i$) (i = 1; :::; m). Clearly, fx$_1$; x$_2$; :::; x$_m$g is an independent set of vertices. Since d(x$_i$) • jV(H$_i$)j ¡ 1, we have

$$¾_2 • ¾_m • \overset{m}{\underset{i=1}{X}} d(x_i) • \overset{m}{\underset{i=1}{X}} jV(H_i)j ¡ m = n ¡ m:$$

On the other hand, by the hypothesis, ¾$_2$, n ¡ k + 1, implying that m • k ¡ 1. ■

Proof of Theorem 2: Let G be a connected graph with ¾$_2$, t$_{k+1}$ ¡ k + 1 for some positive integer k.

Case 1: **G** is Hamiltonian.

By the de¯nition, G has a spanning 1-ended tree T$_1$. Since k , 1, T$_1$ is a spanning k-ended tree.

Case 2: **G** is not Hamiltonian.

Let T$_2$ be a longest path in G.

Case 2.1: ¾$_2$, t$_2$.

By standard arguments, $G[V(T_2)]$ is Hamiltonian. If $t_2 < n$ then recalling that G is connected, we can form a path longer than T_2, contradicting the maximality of T_2. Otherwise G is Hamiltonian and we can argue as in Case 1.

Case 2.2: $\sigma_2 \le t_2 - 1$.

If $k = 1$ then by the hypothesis, $\sigma_2 \ge t_2$, implying that G is Hamiltonian and we can argue as in Case 1. Let $k \ge 2$. Extend T_2 to a k-ended tree T_k and assume that T_k is as large as possible. If T_k is a spanning tree then we are done. Let T_k be not spanning. Then $|End(T_k)| = k$ since otherwise we can form a new k-ended tree larger than T_k, contradicting the maximality of T_k. Now extend T_k to a largest $(k+1)$-ended tree T_{k+1}. Recalling that T_k is a largest k-ended tree, we get $|End(T_{k+1})| = k + 1$ and therefore,

$$t_{k+1} \ge |T_{k+1}| = |T_2| + |T_{k+1} - T_2|.$$

Observing that $|T_2| = t_2$ and $|T_{k+1} - T_2| \ge |End(T_{k+1})| - 2 = k - 1$, we get

$$t_{k+1} \ge t_2 + k - 1 \ge \sigma_2 + k,$$

contradicting the hypothesis. ■

Proof of Theorem 3: Let G be a connected graph with $\sigma_3 \ge t_{k+1} - 2k + 4$ for some integer $k \ge 2$, and let $\overrightarrow{T}_2 = x\overrightarrow{T}_2y$ be a longest path in G. If T_2 is a dominating path then we are done. Otherwise, since G is connected, we can choose a path $\overrightarrow{Q} = w\overrightarrow{Q}z$ such that $V(T_2 \cap Q) = \{w\}$ and $|Q| \ge 3$. Assume that $|Q|$ is as large as possible. Put $T_3 = T_2 \cup Q$. Since T_2 and Q are extreme, we have $N(x) \cup N(y) \subseteq V(T_2)$ and $N(z) \subseteq V(T_3)$. Let w^+ be the successor of w on T_2. If $xy \in E$ then $T_3 + xy - w^+w$ is a path longer than T_2, a contradiction. Let $xy \notin E$. By the same reason, we have $xz, yz \notin E$. Thus, $\{x, y, z\}$ is an independent set of vertices.

Claim 1: $N^-(x) \cap N^+(y) \cap N(z) = \emptyset$.
Proof: Assume the contrary.

Case 1: $v \in N^-(x) \cap N^+(y)$.

If $v = w$ then $xv^+ \in E$ and $T_3 + xv^+ - vv^+$ is a path longer than T_2, a contradiction. Suppose without loss of generality that $v \in V(w^+\overrightarrow{T}_2y)$. If $v = w^+$ then $T_3 + xv^+ - wv - vv^+$ is a path longer than T_2, a contradiction. Finally, if $v \in V(w^{+2}\overrightarrow{T}_2y)$ then

$$T_3 + xv^+ + yv^- - vv^- - vv^+ - ww^+$$

is a path longer than T_2, a contradiction.

Case 2: $v \in N^-(x) \cap N(z)$.

If $v \in V(x\overrightarrow{T}_2w^{-2})$ then

$$T_2 + xv^+ + zv - vv^+ - ww^-$$

is a path longer than T_2, a contradiction. Next, if $v = w^-$ then $T_2 + zw^- - ww^-$ is a path longer than T_2, a contradiction. Further, if $v = w$ then $T_2 + xv^+ - ww^+$ is a path longer than T_2, a contradiction. Finally, if $v \in V(w^+\overrightarrow{T}_2y)$ then

$$T_2 + xv^+ + zv - ww^+ - vv^+$$

is a path longer than T_2, a contradiction.

Case 3: v 2 N^+(y) \ N(z).
By a symmetric argument, we can argue as in Case 2. Claim 1 is proved. ■

By Claim 1,

$$t_3 \;,\; jT_3j \;,\; jN^{¡}(x)j + jN^{+}(y)j + jN(z)j + jfzgj$$

$$= d(x) + d(y) + d(z) + 1 \;,\; ¾_3 + 1: \qquad (1)$$

If k = 2 then by the hypothesis, ¾$_3$, t_{k+1} ¡ 2k + 4 = t_3, contradicting (1). Let k , 3. If T_3 is a dominating 3-ended tree then clearly we are done. Otherwise G ¡ T_3 contains an edge and we can extend T_3 to a largest 4-ended tree T_4 with jT_4j , jT_3j + 2. If k = 3, then by the hypothesis, ¾$_3$, t_{k+1} ¡ 2k+4 = t_4 ¡ 2. On the other hand, by (1), t_4 , jT_4j , jT_3j+2 , ¾$_3$+3, a contradiction. ence, k , 4. If T_4 is dominating, then we are done. Otherwise we can extend T_4 to a largest 5-ended tree T_5 with jT_5j , jT_4j + 2 , jT_3j + 4. This procedure may be repeated until a dominating (r + 1)-ended tree T_{r+1} is found. If r + 1 • k then we are done. Let r , k. Then

$$t_{k+1} \;,\; jT_{k+1}j \;,\; jT_3j + 2(k ¡ 2)$$

$$,\; ¾_3 + 2k ¡ 3 \;,\; t_{k+1} + 1;$$

a contradiction. The proof is complete. ■

References

[1] J. . Bondy and . S. R. Murty, *Graph Theory with Applications*, Macmillan, London and Elsevier, ew ork, 1976.

[2] G. . irac, \Some theorems on abstract graphs", *roceedings of the London Mathematical Society*, vol. 2, no. 3, pp. 69-81, 1952.

[3] O. Ore, \ note on hamiltonian circuits", *American Mathematical Monthly*, vol. 67, p. 55, 1960.

[4] M. Las ergnas, \Sur une propri¶et¶e des arbres maximaux dans un graphe", *Comptes Rendus de l'Acad¶emie des sciences aris S¶er. A 272*, pp. 1297-1300, 1971.

[5] . Broersma and . Tuinstra, \Independence trees and amilton cycles", *J. Graph Theory*, vol. 29, no. 4, pp. 227-237, 1998.

[6] Zh.G. ikoghosyan, \Two theorems on spanning trees", *Uchenie Zapiski GU (Scientiflc Transactions of the Yerevan StatUniversity)*, Ser. Matematika, (in Russian), no. 3, pp. 3-6, 1976.

[7] Zh. G. ikoghosyan, \Theorems on amilton cycles and spanning trees",*Molodoi nauchnii rabotnik*, Ser. atural Sciences, erevan State niversity, (in Russian), vol. 24, no. 2, pp. 15-20, 1976.

[8] . Chie, \On su±cient conditions for a graph to be hamiltonian", *Natural Science*, Ochanomizu niversity, vol. 31, no. 2, pp. 75-80, 1980.

[9] K. Ozeki and T. amashita, \Spanning trees - survey", *Graphs Combinatorics*, vol. 27, no. 1, pp. 1-26, 2011.

[10] Zh. G. ikoghosyan, \Spanning trees on 3-polytopes", *Kibernetika*, (in Russian), no. 4, pp. 35-42, 1982.

[11] **Zh. G. ǀ ikoghosyan, \ n-spanning and hypo-n-spanning graphs",** Tanulmanyok**, Budapest, (in Russian), no. 135, pp. 153-167, 1982.**

Submitted 06.11.2014, accepted 28.01.2015.

Գրաֆում k-ավարտ կմախքային և դոմինանտ ծառերի մասին

Ժ. Նիկողոսյան

Ամփոփում

Ծառի մեկ աստիճան ունեցող գագաթը կոչվում է տերև: Գրաֆում **k**-ից ոչ ավել տերև ունեցող ծառը կոչվում է **k**-ավարտ ծառ: Գրաֆում ամենամեծ **k**-ավարտ ծառի գագաթների քանակը նշանակվում է t_k-ով: **G** գրաֆի **T** ծառը կոչվում է դոմինանտ, եթե V(G-T)-ն գագաթների անկախ բազմություն է: Դիցուք, ¾2-ը (¾3 -ը) գրաֆում ոչ հարևան զույգ (եռյակ) գագաթների աստիճանների հնարավոր ամենափոքր գումարն է: Քիչ տերևներով կմախքային ծառերին առնչվող ամենավաղ արդյունքը (որը ստացվել է հեղինակի կողմից 1976-ին) պնդում է՝ **(/)** եթե **n** գագաթանի **G** կապակցված գրաֆը բավարարում է ¾2 **, n ¡ k + 1** պայմանին ինչ- որ մի **k** դրական ամբողջ թվի համար, ապա **G**-ն ունի **k**-ավարտ կմախքային ծառ: Ներկա աշխատանքում ապացուցվում է, որ **(/)**-ում կապակցվածության պայմանը կարելի է բաց թողնել: Երկրորդ արդյունքը **(/)**-ի ուժեղացումն է՝ **n**-ը փոխարինելով t_{k+1}-ով (ընդհանրապես t_{k+1} • **n**): Երրորդ արդյունքը երկրորդի տարբերակն է՝ դոմինանտ **k**-ավարտ ծառերի համար: Բերված բոլոր արդյունքները ենթակա չեն բարելավման:

О k-висячих остовных и доминантных деревьях

Ж. Никогосян

Аннотация

Дерево с не более чем **k**-висячими вершинами называется **k**-висячим деревом. Число вершин максимального **k**-висячего дерева обозначается через t_k. Через ¾2 (¾3)обозначается минимальная сумма степеней двух (трех) попарно несмежных вершин. Дерево **T** в графе **G** называется доминантным, если V(G-T) является независимым множеством вершин. В 1976 году доказано (автором): **(/)** если n вершинный связный граф G удовлетворяет условию ¾2 **, n ¡ k + 1** для некоторого целого числа **k**, то **G** содержит **k**-висячое остовное дерево. В настоящей работе доказывается, что условие связности в **(/)** можно опускать. Второй результат является усилением **(/)** с помощью замены n через t_{k+1} (напомним, что t_{k+1} • **n**). Приводится также версия второго результата для доминантных **k**-висячих деревьев. Все результаты неулучшаемы.

Mathematical Problems of Computer Science 43, 32--41, 2015.

On Strongly Positive Multidimensional Arithmetical Sets[1]

Seda N. Manukian

Institute for Informatics and Automation Problems of NAS RA
e-mail: zaslav@ipia.sci.am

Abstract

The notion of positive arithmetical formula in the signature $(0, =, S)$, where $S(x) = x+1$, is defined and investigated in [1] and [2]. A multidimensional arithmetical set is said to be positive if it is determined by a positive formula. Some subclass of the class of positive sets, namely, the class of strongly positive sets, is considered. It is proved that for any $n \geq 3$ there exists a $2n$-dimensional strongly positive set such that its transitive closure is non-recursive. On the other side, it is noted that the transitive closure of any 2-dimensional strongly positive set is primitive recursive.

Keywords: Arithmetical formula, Transitive closure, Recursive set, Signature.

1. Introduction

The classes of recursive sets having in general non-recursive transitive closures have been investigated in the theory of algorithms since the first steps of this theory ([3]-[8]). The works [9]-[13] are dedicated mainly to algebraic problems, however, some examples of recursive sets having non-recursive transitive closures are actually given also in these works. In [14] it is noted that there exists a two-dimensional arithmetical set belonging to the class Σ_4 and having a non-recursive transitive closure (the classes Σ_n for $n \geq 0$ are defined in [14] as some classes of arithmetical sets determined by formulas in M. Presburger's system ([4], [15], [16])). Below the class of strongly positive arithmetical sets is considered (the definition will be given in Section 2) such that the sets belonging to this class have a more simple structure than the sets noted above, and have the following properties: (1) for any $n \geq 3$ there exists a $2n$-dimensional strongly positive set such that its transitive closure is non-recursive; (2) any 2-dimensional strongly positive set has a primitive recursive transitive closure (see below, Theorem 1 and Theorem 2).

[1] This work was supported by State Committee of Science, MES RA, in frame of the research project №SCL 13-1B321.

2. Main Definitions and Results

By N we denote the set of all non-negative integers, $N = \{0,1,2,...\}$. By N^n we denote the set of n-tuples $(x_1, x_2, ..., x_n)$, where $n \geq 1$, $x_i \in N$ for $1 \leq i \leq n$.

An n-dimensional arithmetical set, where $n \geq 1$, is defined as any subset of N^n.

An n-dimensional arithmetical predicate P is defined as a predicate which is true on some set $A \subseteq N^n$ and false out of it; in this case we say that A is the set of truth for P, and P is the representing predicate for A.

The notions of primitive recursive function, general recursive function, partially recursive function, primitive recursive set, recursive set are defined in a usual way ([3]-[8]). The corresponding terms will be shortly denoted below by PmRF, GRF, PtRF, PmRS, RS.

We will consider arithmetical formulas in the signature $(0, =, S)$, where $S(x) = x + 1$, for $x \in N$ (see [1]-[8]). Any term included in a formula of the mentioned kind has the form $S(S(...S(x)...))$ or $S(S(...S(0)...))$, where x is a variable. Such terms we will denote correspondingly by $S^k(x)$ and $S^k(0)$, where k is the quantity of symbols S contained in the considered term. We replace $S^0(x)$ and $S^0(0)$ with x and 0. Any elementary subformula of a formula of this kind has the form $t_1 = t_2$, where t_1 and t_2 are terms. Any arithmetical formula of this kind is obtained by the logical operations $\&, \vee, \supset, \neg, \forall, \exists$ from elementary formulas. We say that a formula is semi-elementary if it has the form $t_1 = t_2$ or $\neg(t_1 = t_2)$, where t_1 and t_2 are terms.

The deductive system of formal arithmetic in the signature $(0, =, S)$ is defined as in [4], [6]; we will denote this system by Ded$_S$ (cf. [1], [2]). As it is proved in [4], this system is complete. We say that formulas F and G in the signature $(0, =, S)$ are Ded$_S$-equivalent (or simply equivalent) if the formula $(F \supset G) \& (G \supset F)$ is deducible in Ded$_S$. Below we consider formulas of the mentioned kind up to their Ded$_S$-equivalence.

An arithmetical formula of the mentioned kind is said to be positive if it contains no other symbols of logical operations except $\exists, \&, \vee, \neg$, and all the symbols $\neg$ of negation relate to elementary subformulas containing no more than one variable (see [1], [2]). An arithmetical formula of this kind is said to be strongly positive if it can be obtained by the logical operations $\&$ and $\vee$ from semi-elementary formulas of the following forms: $x = a$, where x is a variable, a is a constant, $a \in N$; $x = y$, where x and y are variables; $x = S(y)$, where x and y are variables; $\neg(x = 0)$, where x is a variable. An arithmetical predicate is said to be positive (correspondingly, strongly positive), if it can be expressed by a positive (correspondingly, strongly positive) formula. An arithmetical set is said to be positive (correspondingly, strongly positive) if its representing predicate is positive (correspondingly, strongly positive).

The notion of one-dimensional creative set is given in a usual way ([3], [5], [7], [8]). We will slightly generalize this notion. We use a PmRF $c_n(x_1, x_2, ..., x_n)$, where $n \geq 2$, establishing a one-to-one correspondence between N^n and N (for example, $c_n(x_1, x_2, ..., x_n) = c_2(c_2(...c_2(c_2(x_1, x_2), x_3)..., x_{n-1}), x_n)$, where $c_2(x, y) = 2^x \cdot (2y + 1) - 1$). We say that a set $B \subseteq N^n$ is an n-dimensional image of a set $A \subseteq N$ when $c_n(x_1, x_2, ..., x_n) \in A$ if and only if $(x_1, x_2, ..., x_n) \in B$. The set $B \in N^n$ is said to be creative in the generalized sense if it is an n-dimensional image of some one-dimensional creative set. Clearly, the properties of creative sets in the generalized sense are similar to the properties of one-dimensional creative sets (for example, all sets creative in the generalized sense are non-recursive).

Transitive closure A^* of an arithmetical set A having an even dimension $2k$ is defined in a usual way by the following generating rules (cf. [1], [2], [13]): (1) if $(x_1, x_2, ..., x_{2k}) \in A$, then $(x_1, x_2, ..., x_{2k}) \in A^*$, (2) if $(x_1, x_2, ..., x_k, y_1, y_2, ... y_k) \in A^*$, and $(y_1, y_2, ..., y_k, z_1, z_2, ... z_k) \in A^*$, then $(x_1, x_2, ..., x_k, z_1, z_2, ... z_k) \in A^*$.

Theorem 1: *For any $n \geq 3$ there exists a $2n$-dimensional strongly positive set such that its transitive closure is creative in the generalized sense.*

Theorem 2: *Transitive closure of any 2-dimensional strongly positive set is primitive recursive.*

The proof of Theorem 1 will be given below. The proof of Theorem 2 will be published later.

3. Auxiliary Notions and Statements

We will use some class of operator algorithms ([8], [17]) having a special structure. The algorithms belonging to this class we will call Ω-algorithms. Any Ω-algorithm consists of finite number of elementary Ω-algorithms, which will be called below "Ω-operators". The set of all Ω-operators included in the considered Ω-algorithm we call "scheme" of this Ω-algorithm. We suppose that some non-negative integer is attached to any Ω-operator in the scheme of a given Ω-algorithm in such a way, that different integers are attached to different Ω-operators. The integer attached to some Ω-operator we call "an identifier" of this Ω-operator. In this case we say that this Ω-operator has the mentioned identifier. Any Ω-operator implements one step of the process of computation realized by the considered Ω-algorithm. The objects transformed in the process of computation are non-negative integers. The state of the mentioned computation process is defined as a pair (α, w), where α is the identifier attached to the Ω-operator which is working on the considered step of the process, and w is the number obtained by the previous steps of the process. Ω-operators are algorithms having one of the following forms (where α is the identifier attached to the considered Ω-operator, β and γ are identifiers attached to Ω-operators which should work after the working of this Ω-operator):

(1) (α, end). This Ω-operator is called below "a final operator"; it finishes the process of computation.

(2) $(\alpha, \times 2, \beta)$. This Ω-operator transforms the state (α, w) to the state $(\beta, 2w)$.

(3) $(\alpha, \times 3, \beta)$. This Ω-operator transforms the state (α, w) to the state $(\beta, 3w)$.

(4) $(\alpha, :6, \beta, \gamma)$. This Ω-operator transforms the state (α, w) to the state $(\beta, \frac{w}{6})$ if the number w is divisible by 6; in the opposite case it transforms the state (α, w) to the state (γ, w).

Note that such forms of operators are considered actually in [17] (see also [8], p. 292, p. 312).

We suppose that any scheme of Ω-algorithm contains only a single final Ω-operator which has the identifier $\alpha = 0$. Among the operators contained in the scheme of the considered Ω-algorithm we distinguish the initial Ω-operator having the identifier $\alpha = 1$; the working of this operator begins the process of computation. The whole process of working of the given Ω-algorithm is described by the sequence of states (α_1, w_1), (α_2, w_2),..., (α_k, w_k),...,(where

$\alpha_1 = 1$) obtained during the working of this Ω-algorithm. The process is described by a finite sequence $(1, w_1)$, (α_2, w_2),..., $(0, w_m)$ if it is finished by the working of the final Ω-operator.

In this case we say that the considered Ω-algorithm transforms the state $(1, w_1)$ to the state $(0, w_m)$, and is applicable to the state $(1, w_1)$. If the final Ω-operator does not work during the process of computation, then the mentioned sequence $(1, w_1)$, (α_2, w_2),... is infinite. In this case we say that the considered Ω-algorithm is not applicable to the state $(1, w_1)$.

The following theorem is proved in [17] (see also [8], pp. 312-315) in some other terms.

Theorem 3 ([17]): *For any PtRF $f(x)$ there exists an Ω-algorithm which transforms the state $(1, 2^{2^x})$ to the state $(0, 2^{2^{f(x)}})$ when the value $f(x)$ is defined, and is not applicable to the state $(1, 2^{2^x})$ in the opposite case.*

If some Ω-algorithm has the property described in Theorem 3, then we say that this Ω-algorithm realizes the PtRF $f(x)$. For example, the following Ω-algorithm:

$$(0, end),\ (1, \times 3, 2),\ (2, : 6, 1, 3),\ (3, \times 2, 0)$$

realizes the GRF $f(x) = 0$.

We will use also another classes of algorithms, namely, Γ_n-algorithms for $n \geq 1$.

These algorithms are actually special cases of graph-schemes with memory ([18]), though they will be described below in some other terms than the descriptions in [18].

Any Γ_n-algorithm consists of finite number of Γ_n-operators. The set of all Γ_n-operators included in the considered Γ_n-algorithm we call "scheme" of this Γ_n-algorithm. The index n in the notation Γ_n denotes that the objects transformed by the considered Γ_n-algorithm are n-tuples $(x_1, x_2, ..., x_n)$, where $x_i \in N$ for $1 \leq i \leq n$. The notion of identifier attached to the considered Γ_n-operator is defined similarly to the notion of "identifier attached to the considered Ω-operator" which is given above; we suppose that different Γ_n-operators have different identifiers attached to them. If some identifier is attached to a Γ_n-operator, we will say that this Γ_n-operator has the mentioned identifier.

The state of the computation process realized by a Γ_n-algorithm is defined as an $(n+1)$-tuple $(\alpha, x_2, x_3, ..., x_{n+1})$, where α is the identifier attached to the Γ_n-operator which is working on the considered step of the process, and $(x_2, x_3, ..., x_{n+1})$ is the n-tuple of numbers obtained by the previous steps of the process. Γ_n-operators are algorithms having one of the following forms (where the notations α, β, γ have the same sense as α, β, γ in the description of Ω-operators given above):

(1) (α, end). This Γ_n-operator we call "a final operator"; it finishes the process of computation.

(2) $(\alpha, x_i + 1, \beta)$, where $2 \leq i \leq n+1$. This Γ_n-operator transforms the state $(\alpha, x_2, x_3, ..., x_{i-1}, x_i, x_{i+1}, ..., x_{n+1})$ to the state $(\beta, x_2, x_3, ..., x_{i-1}, x_i + 1, x_{i+1}, ..., x_{n+1})$.

(3) $(\alpha, x_i \dot{-} 1, \beta)$, where $2 \le i \le n+1$; we denote by the symbol $\dot{-}$ the PmRF such that $x \dot{-} y = x - y$ when $x \ge y$, and $x \dot{-} y = 0$ when $x < y$ (cf. [3]-[8]). This Γ_n-operator transforms the state $(\alpha, x_2, x_3, ..., x_{i-1}, x_i, x_{i+1}, ..., x_{n+1})$ to the state $(\beta, x_2, x_3, ..., x_{i-1}, x_i \dot{-} 1, x_{i+1}, ..., x_{n+1})$.

(4) $(\alpha, x_i = 0, \beta, \gamma)$,where $2 \le i \le n+1$. This Γ_n-operator transforms the state $(\alpha, x_2, x_3, ..., x_{n+1})$ to the state $(\beta, x_2, x_3, ..., x_{n+1})$ when $x_i = 0$, and to the state $(\gamma, x_2, x_3, ..., x_{n+1})$ when $x_i \ne 0$.

We suppose that any scheme of Γ_n-algorithm contains only a single final Γ_n-operator which has the identifier $\alpha = 0$. Among the Γ_n-operators contained in the scheme of the considered Γ_n-algorithm we distinguish the initial Γ_n-operator having the identifier $\alpha = 1$; the working of this operator begins the process of computation. This process is described by a sequence of states (α_1, Q_1), (α_2, Q_2),..., (α_k, Q_k),... where $\alpha_1 = 1$, and any Q_i is an n-tuple $(x_2^{(i)}, x_3^{(i)}, ..., x_{n+1}^{(i)})$. Such a sequence is finite if the final Γ_n-operator works during the mentioned process, and is infinite in the opposite case. If the sequence of states is finite, then we say that the considered Γ_n-algorithm is applicable to the state $(1, Q_1)$; in this case we say also that Γ_n-algorithm transforms the state $(1, Q_1)$ to the state $(0, Q_m)$, where $(0, Q_m)$ is the last state in the considered sequence. If the sequence of states $(1, Q_1)$, $(2, Q_2)$,... is infinite, then we say that the considered Γ_n-algorithm is not applicable to the state $(1, Q_1)$.

We say that a Γ_n-algorithm (where $n \ge 2$) realizes a PtRF $f(x)$, if for any $x \in N$ it transforms the state $(1, 2^x, 0, 0, ..., 0)$ to the state $(0, 2^{f(x)}, 0, 0, ..., 0)$ when the value $f(x)$ is defined, and is not applicable to the state $(1, 2^x, 0, 0, ..., 0)$ when the value $f(x)$ is not defined. For example, the following Γ_n-algorithm realizes the PtRF $f(x)$ which is nowhere defined:
$(0, end)$, $(1, x_2 \dot{-} 1, 1)$.

Lemma 3.1: *If the initial state in the process of computation realized by some Ω-algorithm has the form $(1, 2^u, 3^v)$, where $u \in N$, $v \in N$, then any state (α_m, w_m) included in this process satisfies the condition $w_m = 2^t \cdot 3^s$, where $t, s \in N$.*

The proof is easily obtained from the definitions.

Lemma 3.2: *For any Ω-algorithm φ realizing some PtRF $f(x)$ there exists a Γ_2-algorithm ψ realizing the same PtRF $f(x)$.*

Proof: We will consider the process of computation realized by the Ω-algorithm φ. Any initial state in such a process has the form $(1, 2^{2^x})$ that is $(1, 2^{2^x} \cdot 3^0)$. As it is proved in Lemma 3.1 any state included in such a process has the form $(\alpha_m, 2^t \cdot 3^s)$ where $t, s \in N$. For any Ω-operator included in the scheme of Ω-algorithm φ we will construct some subscheme of the supposed Γ_2-algorithm ψ which has the following property: if the considered Ω-operator transforms the state $(\alpha, 2^u \cdot 3^v)$ to the state $(\beta, 2^t \cdot 3^s)$ then the corresponding subscheme of the supposed Γ_2-

algorithm ψ transforms the state (α,u,v) of Γ_2-algorithm ψ to the state (β,t,s). We will consider the following cases.

Case 1. The considered Ω-operator has the form $(\alpha,\times 2,\beta)$. In this case the required subscheme of the supposed Γ_2-algorithm ψ consists of the single Γ_2-operator (α,x_2+1,β).

Case 2. The considered Ω-operator has the form $(\alpha,\times 3,\beta)$. In this case the required subscheme of the supposed Γ_2-algorithm ψ consists of the single Γ_2-operator (α,x_3+1,β).

Case 3. The considered Ω-operator has the form $(\alpha,:6,\beta,\gamma)$. In this case the required subscheme of the supposed Γ_2-algorithm ψ consists of the following Γ_2-operators: $(\alpha,x_2=0,\gamma,\delta_1)$, $(\delta_1,x_3=0,\gamma,\delta_2)$, $(\delta_2,x_2\dot{-}1,\delta_3)$, $(\delta_3,x_3\dot{-}1,\beta)$. Here δ_1, δ_2, δ_3 are identifiers attached to additional Γ_2-operators which are included in the scheme of the supposed Γ_2-algorithm for modeling the working of the considered Ω-operator. Of course, these identifiers should be different in different subschemes of this kind.

Case 4. The considered Ω-operator has the form $(0,end)$. This Ω-operator does not transform the states of Ω-algorithm. So, the corresponding Γ_2-operator has the same form $(0,end)$.

The scheme of the supposed Γ_2-algorithm is obtained as the union of subschemes of the mentioned forms constructed for all Ω-operators included in the scheme of the given Ω-algorithm. It is easily seen that such Γ_2-algorithm satisfies the conditions of Lemma 3.2. This completes the proof.

Corollary 1: *For any PtRF $f(x)$ and any $n\geq 2$ there exists a Γ_n-algorithm realizing the PtRF $f(x)$.*

The proof is based on Theorem 3 and is similar to that of Lemma 3.2.

Note: *The statements established in Lemma 3.2 and in its Corollary 1 are similar to Theorem 7.1 in [18], where it is proved that any PtRF may be realized by some graph-scheme with memory constructed on the base of the functions $x+1$, $x\dot{-}1$ and of the predicate $x=0$. However, graph-schemes with memory corresponding to Γ_n-algorithms are essentially simpler than the graph-schemes considered in Theorem 7.1 in [18]. Besides, the definition of realizability of PtRf by Γ_n-algorithm differs from the corresponding definition in [18].*

Now let us define for any Γ_n-algorithm, where $n\geq 1$, the predicate describing one step of computation process realized by this Γ_n-algorithm. Such a predicate we will call “a step describing predicate”, or, shortly, “SD-predicate” for a given Γ_n-algorithm. Namely, if η is the SD-predicate for a given Γ_n-algorithm, then $\eta(x_1,x_2,...,x_{2n+2})$ is true if and only if the given Γ_n-algorithm transforms the state $(x_1,x_2,...,x_{n+1})$ to the state $(x_{n+2},x_{n+3},...,x_{2n+2})$ by one step of the corresponding computation process. Let us note the following property of the predicate η: if $(x_1,x_2,...,x_{n+1})$ is a state of the computational process realized by the considered Γ_n-algorithm, such that $x_1 \neq 0$, then there exists a single $(n+1)$-tuple $(x_{n+2}, x_{n+3}, ..., x_{2n+2})$ such that $\eta(x_1, x_2, ..., x_{2n+2})$ is true.

The set of truth for the mentioned predicate η we will call "SD-set" for the considered Γ_n-algorithm. Clearly, such a set π has the following property: $(x_1, x_2, ..., x_{2n+2}) \in \pi$ if and only if $(x_1, x_2, ..., x_{n+1})$ is a state of computation process realized by the considered Γ_n-algorithm, and this Γ_n-algorithm transforms the state $(x_1, x_2, ..., x_{n+1})$ to the state $(x_{n+2}, x_{n+3}, ..., x_{2n+2})$ by one step of the computation process.

Now let us define the forms of SD-predicates and SD-sets for Γ_n-algorithms. We suppose that some Γ_n-algorithm ψ, where $n \geq 1$ is fixed. We will define the forms of SD-predicates for any Γ_n-operator included in the scheme of ψ.

Case 1. The considered Γ_n-operator has the form $(\alpha, x_i + 1, \beta)$. Such Γ_n-operator transforms the state $(\alpha, x_2, x_3, ..., x_{i-1}, x_i, x_{i+1}, ..., x_{n+1})$ to the state $(\beta, x_{n+3}, x_{n+4}, ..., x_{n+i}, x_{n+i+1}, x_{n+i+2}, ..., x_{2n+2})$, where $x_{n+3} = x_2$, $x_{n+4} = x_3, \ldots, x_{n+i} = x_{i-1}$, $x_{n+i+1} = x_i + 1$, $x_{n+i+2} = x_{i+1}, \ldots,$ $x_{2n+2} = x_{n+1}$.

The SD-predicate for such a Γ_n-operator is expressed by the following formula: $(x_1 = \alpha) \,\&\, (x_{n+2} = \beta) \,\&\, (x_{n+3} = x_2) \,\&\, (x_{n+4} = x_3) \,\&\, ... \,\&\, (x_{n+i} = x_{i-1}) \,\&\, (x_{n+i+1} = S(x_i)) \,\&$
$\&(x_{n+i+2} = x_{i+1}) \,\&\, ... \,\&\, (x_{2n+2} = x_{n+1})$.

Case 2. The considered Γ_n-operator has the form $(\alpha, x_i \dot{-} 1, \beta)$. Such Γ_n-operator transforms the state $(\alpha, x_2, x_3, ..., x_{i-1}, x_i, x_{i+1}, ..., x_{n+1})$ to the state $(\beta, x_{n+3}, x_{n+4}, ..., x_{n+i}, x_{n+i+1}, x_{n+i+2}, ..., x_{2n+2})$,where $x_{n+3} = x_2$, $x_{n+4} = x_3, \ldots, x_{n+i} = x_{i-1}$, $x_{n+i+1} = x_i \dot{-} 1$, $x_{n+i+2} = x_{i+1}, \ldots,$ $x_{2n+2} = x_{n+1}$.

The SD-predicate for such a Γ_n-operator is expressed by the following formula: $(x_1 = \alpha) \,\&\, (x_{n+2} = \beta) \,\&\, (x_{n+3} = x_2) \,\&\, (x_{n+4} = x_3) \,\&\, ... \,\&\, (x_{n+i} = x_{i-1}) \,\&\, (x_{n+i+2} = x_{i+1}) \,\&\, ...$
$\&\, (x_{2n+2} = x_{n+1}) \,\&\, (((x_{n+i+1} = 0) \,\&\, (x_i = 0)) \vee (\neg(x_i = 0) \,\&\, (x_i = S(x_{n+i+1}))))$.

Case 3. The considered Γ_n-operator has the form $(\alpha, x_i = 0, \beta, \gamma)$. Such Γ_n-operator transforms the state $(\alpha, x_2, x_3, ..., x_{n+1})$ to the states $(\beta, x_{n+3}, x_{n+4}, ..., x_{2n+2})$ or $(\gamma, x_{n+3}, x_{n+4}, ..., x_{2n+2})$ (where $x_{n+3} = x_2$, $x_{n+4} = x_3, \ldots, x_{2n+2} = x_{n+1}$) in the cases, when, correspondingly, $x_i = 0$ or $x_i \neq 0$. The SD-predicate for such a Γ_n-operator is expressed by the following formula: $(x_1 = \alpha) \,\&\, (x_{n+3} = x_2) \,\&\, (x_{n+4} = x_3) \,\&\, ... \,\&\, (x_{2n+2} = x_{n+1}) \,\&\, (((x_{n+2} = \beta) \,\&\, (x_i = 0)) \vee$
$((x_{n+2} = \gamma) \,\&\, \neg(x_i = 0)))$.

Case 4. The considered Γ_n-operator has the form $(0, end)$. Such Γ_n-operator does not transform the states of Γ_n-algorithm, so, an SD-predicate is not considered for such Γ_n-operator.

The SD-predicate for Γ_n-algorithm ψ is expressed by the formula obtained as the disjunction of formulas expressing SD-predicates constructed above for all Γ_n-operators contained in the scheme of ψ and different from the operator $(0, end)$. The SD-set for Γ_n-algorithm ψ is obtained as the set of truth for the corresponding SD-predicate. Clearly, such SD-set is a $(2n+2)$-dimensional arithmetical set.

Lemma 3.3: *SD-predicate and SD-set constructed for any Γ_n-algorithm, where $n \geq 1$, are strongly positive.*

The proof is obtained evidently from the definitions.

Lemma 3.4: (cf. [13], p.72). *If A is a $2k$-dimensional set, $A \subseteq N^{2k}$, then $2k$-tuple $(x_1, x_2, ..., x_k, y_1, y_2, ..., y_k)$ belongs to the transitive closure A^* of the set A if and only if there exists a sequence $(Q_1, Q_2, ..., Q_m)$ of k-tuples, such that $m \geq 2$, $Q_1 = (x_1, x_2, ..., x_k)$, $Q_m = (y_1, y_2, ..., y_k)$ and any $2k$-tuple (Q_i, Q_{i+1}) for $1 \leq i \leq m-1$ belongs to A.*

The proof is easily obtained using the definition of the transitive closure A^*.

4. Proof of Theorem 1

Let M be any one-dimensional creative set ([3], [5], [7], [8]). We consider the PtRF $f(x)$ such that $f(x) = 0$ when $x \in M$, and the value $f(x)$ is indefined when $x \notin M$. For any fixed $n \geq 2$ we construct (using Corollary of Lemma 3.2) a Γ_n-algorithm ψ realizing the PtRF $f(x)$; clearly, ψ transforms the state $(1,2^x,0,0,...,0)$ to the state $(0,1,0,0,...,0)$ when $x \in M$ and is not applicable to the state $(1,2^x,0,0,...,0)$ when $x \notin M$. Now, let us consider the SD-predicate η and SD-set π for ψ. Clearly, η is true for $(2n+2)$-tuple $(x_1, x_2, ..., x_{n+1}, y_1, y_2, ..., y_{n+1})$ (and the statement $(x_1, x_2, ..., x_{n+1}, y_1, y_2, ..., y_{n+1}) \in \pi$ holds) if and only if ψ transforms the state $(x_1, x_2, ..., x_{n+1})$ to the state $(y_1, y_2, ..., y_{n+1})$ by one step of the process of computation. Let us consider the transitive closure π^* of the SD-set π.

Using Lemma 3.4 we conclude that $(x_1, x_2, ..., x_{n+1}, y_1, y_2, ..., y_{n+1}) \in \pi^*$ if and only if there exists a sequence $(Q_1, Q_2, ..., Q_m)$ of $(n+1)$-tuples such that $Q_1 = (x_1, x_2, ..., x_{n+1})$, $Q_m = (y_1, y_2, ..., y_{n+1})$, and $(Q_i, Q_{i+1}) \in \pi$ for any i such that $1 \leq i < m$. But in this case the sequence $(Q_1, Q_2, ..., Q_m)$ is a sequence of states of the Γ_n-algorithm ψ which describes some part of a process of computation implemented by the Γ_n-algorithm ψ.

Hence, the $(2n+2)$-tuple $(1,2^x,0,0,...,0,0,1,0,0,...,0)$ belongs to π^* if $x \in M$. It is easily seen that the mentioned $(2n+2)$-tuple does not belong to π^* if $x \notin M$. Let us consider the set $\pi^{**} \in N$ such that its $(2n+2)$-dimensional image is π^*. Then $c_{2n+2}(1,2^x,0,0,...,0,0,1,00,...,0) \in \pi^{**}$ if and only if $x \in M$. So the set M is m-reducible to the set π^{**}. Using the corresponding theorem concerning m-reducibility (see, for example, [8], p. 161), we conclude that the set π^{**} is creative, the set π^* is creative in the generalized sense, and the set π is strongly positive (see Lemma 3.3). This completes the proof.

Note: *It is seen from Theorem 1 that the transitive closures of some strongly positive sets having the dimensions 6, 8, 10, … are creative in the generalized sense. On the other side (Theorem 2) the transitive closure of any 2-dimensional strongly positive set is primitive recursive. Similar problem concerning 4-dimensional strongly positive sets remains open.*

References

[1] S. N. Manukian, “On the representation of recursively enumerable sets in weak arithmetics”, *Transactions of the IIAP of NAS RA, Mathematical Problems of Computer Science*, vol. 27, pp. 90--110, 2006.

[2] S. N. Manukian, “On an algebraic classification of multidimensional recursively enumerable sets expressible in formal arithmetical systems”, *Transactions of the IIAP of NAS RA, Mathematical Problems of Computer Science*, vol. 41, pp. 103-113, 2014.

[3] S. C. Kleene, *Introduction to Metamathematics*, D, Van Nostrand Comp., Inc., New York-Toronto, 1952.

[4] H. B. Enderton, *A Mathematical Introduction to Logic*, 2nd edition, San Diego, Harcourt, Academic Press, 2001.

[5] E. Mendelson, *Introduction to Mathematical Logic*, D, Van Nostrand Comp., Inc., Princeton-Toronto-New York-London, 1964.

[6] D. Hilbert and P. Bernays, *Grundlagen der Mathematik*, Band1, Zweite Auflage, Berlin-Heidelberg-New York, Springer Verlag, 1968.

[7] H. Rogers, *Theory of Recursive Functions and Effective Computability*, McGraw Hill Book Comp., New York-St. Louis-San Francisco-Toronto-London-Sydney, 1967.

[8] A. I. Malcev, *Algorithms and Recursive Functions*, 2nd edition, (in Russian), 1986.

[9] A. A. Markov, “Impossibility of some algorithms in the theory of associative systems”, *Reports of the Acad. Sci. USSR*, (in Russian), vol. 55, no. 7, pp. 587-590, 1947.

[10] E. L. Post, “Recursive unsolvability of a problem of Thue”, *Journ. of Symb. Logic*, vol. 12, pp. 1-11, 1947.

[11] P. S. Novikov, “On the algorithmic unsolvability of identity problem in the group theory”,*Transactions of Steklov Institute of the Acad. Sci. USSR*, (in Russian), vol. 44, 1955.

[12] G. S. Tseytin, “Associative calculus with the unsolvable problem of equivalence”, *Transactions of Steklov Institute of the Acad. Sci. USSR*, (in Russian), vol. 52, pp. 172-189, 1958.

[13] G. S. Tseytin, “One method of representation for the theory of algorithms and enumerable sets”, *Transactions of Steklov Institute of the Acad. Sci. USSR*, (in Russian), vol. 72, pp. 69-98, 1964.

[14] S. N. Manukian,. “Classification of many-dimensional arithmetical sets represented in M. Presburger’s system”, *Reports of the National Acad. Sci. of Armenia*, (in Russian), vol. 111, no. 2, pp. 114--120, 2011.

[15] M. Presburger, “Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt”, *Comptes Rendu du I Congres des Mathematiciens des Pays Slaves*, Warszawa, pp. 92-101, 1930.

[16] R. Stransifer, *Presburger’s Article on Integer Arithmetics: Remarks and Translation*, Department of Computer Science, CornellUniversity, Ithaca, New York, 1984.

[17] M. L. Minsky, “Recursive unsolvability of Post’s problem of “Tag” and topics in theory of Turing machines”, *Ann. Math.*, vol. 74, pp. 437-455, 1961.

[18] I.D. Zaslavsky, “Graph-schemes with memory”, *Transactions of Steklov Institute of Acad. Sci. USSR*, (in Russian), vol. 72, pp. 99-192, 1964.

Submitted 25.11.2014, accepted 27.01.2015.

Խիստ պոզիտիվ բազմաչափ թվաբանական բազմությունների մասին

Ս. Մանուկյան

Ամփոփում

[1]-ում և [2]-ում սահմանվում և հետազոտվում է պոզիտիվ թվաբանական բանաձևի գաղափարը $(0,=,S)$ սիգնատուրայում, (որտեղ $S(x)=x+1$)։ Բազմաչափթվաբանական բազմությունը կոչվում է պոզիտիվ, եթե այն որոշվում է որևէ պոզիտիվ բանաձևի միջոցով։ Դիտարկվում է պոզիտիվ բազմությունների դասի որևէ ենթադաս, այսինքն՝ խիստ պոզիտիվ բազմությունների դասը։ Ապացուցվում է, որ ցանկացած n-ի համար, որտեղ $n\geq 3$, գոյություն ունի $2n$-չափանի խիստ պոզիտիվ բազմություն, որի տրանզիտիվ փակումը ռեկուրսիվ չէ։ Մյուս կողմից նշվում է, որ ցանկացած 2-չափանի խիստ պոզիտիվ բազմություն ունի պարզագույն ռեկուրսիվ տրանզիտիվ փակում։

О строго позитивных многомерных арифметических множествах

С. Манукян

Аннотация

Понятие позитивной арифметической формулы в сигнатуре $(0,=,S)$, где $S(x)=x+1$, определено и исследовано в [1] и [2]. Многомерное арифметическое множество называем позитивным, если оно задаётся позитивной формулой. Рассматривается подкласс класса позитивных множеств, а именно, класс строго позитивных множеств. Доказывается, что для всякого $n\geq 3$ существует строго позитивное множество размерности $2n$, такое, что его транзитивное замыкание нерекурсивно. С другой стороны, указывается, что транзитивное замыкание всякого строго позитивного множества размерности 2 примитивно рекурсивно.

Mathematical Problems of Computer Science 43, 42--46, 2015.

On Generalized Primitive Recursive String Functions[1]

Mikayel H. Khachatryan

Institute for Informatics and Automation Problems of NAS RA
e-mail: mikayel.khachatur@gmail.com

Abstract

The notion of generalized primitive recursive string function is introduced and relations between such functions and primitive recursive string functions in the usual sense ([1], [2]) are investigated. It is proved that any generalized primitive recursive string function is everywhere defined if and only if it is a primitive recursive string function in the usual sense.

Keywords: String function, Primitive recursive string function, Superposition, Alphabetic primitive recursion.

1. Introduction

The notion of primitive recursive string function ([1], [2]) is generalized in the following sense: string functions are considered which are defined similar to the definition of primitive recursive string functions, however, such functions are in general not everywhere defined. Namely, the definition of generalized primitive recursive string function is obtained from the definition of primitive recursive string function in the usual sense by adding the everywhere undefined one-dimensional string function to the set of basic functions. It is proved that any generalized primitive recursive string function is everywhere defined if and only if it is a primitive recursive string function in the usual sense.

Similar problems concerning arithmetical functions are considered in [3].

2. Generalized Primitive Recursive String Functions

The notion of many-dimensional primitive recursive string function is given in [1], [2]. For the convenience of the reader let us recall the corresponding definitions.

Let A be an alphabet, i.e. a list of different symbols, $A = \{a_1, a_2, \dots, a_p\}$ $(p > 1)$. By A^* we denote the set of all words in A (including the empty word Λ). The symbols $a_1, a_2, \dots, a_p$

[1] This work was supported by State Committee of Science, MESRA, in frame of the research project № SCS 13-1B321.

we call *letters* in *A*. The *length* of a word $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ is the number *k* (the length of the empty word Λ is 0).

We say that the function *F* is an *n-dimensional string function* $(n \geq 1)$ in the alphabet *A* if for any *n*-tuple $(P_1, P_2, \cdots, P_n)$ where all P_i are words in *A*, the value $F(P_1, P_2, \cdots, P_n)$ is either undefined, or is a word in *A*. By $!F(P_1, P_2, \cdots, P_n)$ we denote the statement: the value $F(P_1, P_2, \cdots, P_n)$ is defined.

Below we consider only the string functions in the fixed alphabet *A*.

Basic string functions are defined as functions of the following kinds.

1. One-dimensional function $D(P)$such that $D(P) = \Lambda$ for any word *P* in *A*.
2. One-dimensional function $S_i(P)$, where $1 \leq i \leq p$ such that $S_i(P) = Pa_i$, for any word *P* in *A*.
3. *n*-dimensional functions $I_m^n(P_1, P_2, \cdots, P_n)$ where $n \geq 1,\ 1 \leq m \leq n,$ such that $I_m^n(P_1, P_2, \cdots, P_n) = P_m$ for any *n*-tuple $(P_1, P_2, \cdots, P_n)$ of words in *A*.

The operator $\boldsymbol{S}$ of *superposition* is defined as follows. If *G* is an *n*-dimensional string function, $G_1, G_2, \cdots, G_n$ are *m*-dimensional string functions, then the *m*-dimensional string function $f = \boldsymbol{S}(G,\ G_1,\ G_2, \cdots,\ G_n)$ is defined by the following equality:

$$f(P_1, P_2, \cdots, P_m) = G\big(G_1(P_1, P_2, \cdots, P_m),\ G_2(P_1, P_2, \cdots, P_m), \cdots,\ G_n(P_1, P_2, \cdots, P_m)\big),$$

where $P_1, P_2, \cdots, P_m$ are any words in *A*.

The operator $\boldsymbol{R}$ of *alphabetic primitive recursion* is defined as follows. If *G* is an *n*-dimensional string function, $H_1,\ H_2, \cdots,\ H_p$ are (*n*+2)-dimensional string functions, then the (*n*+1)-dimensional string function $f = \boldsymbol{R}(G, H_1,\ H_2, \cdots,\ H_p)$ is defined by the following equalities:

$$f(P_1, P_2, \cdots, P_n, \Lambda) = G(P_1, P_2, \cdots, P_n),$$

$$f(P_1, P_2, \cdots, P_n, Pa_i) = H_i\big(P_1, P_2, \cdots, P_n, P, f(P_1, P_2, \cdots, P_n, P)\big),$$

where $1 \leq i \leq p$ and $P_1, P_2, \cdots, P_n, P$ are any words in *A*.

We say that a string function is a *primitive recursive string function* (***PRSF***), if it can be obtained from basic functions by the operators of superposition and alphabetic primitive recursion.

The notion of *generalized primitive recursive string function* (***GPRSF***) is defined similar to the notion of *PRSF* with the only difference: one-dimensional everywhere undefined *U(P)* function is added to the set of basic functions.

Below the statements "*F* is a primitive recursive string function in the usual sense", "*F* is a generalized primitive recursive string function", will be denoted correspondingly by $F \in PRSF$ and $F \in GPRSF$. As it is known ([1]. [2]) every function $F \in PRSF$ is everywhere defined.

Clearly, any primitive recursive string function in the usual sense is a generalized primitive recursive string function, and, on the other side, the set of generalized primitive recursive string functions is wider than one of primitive recursive string functions in the usual sense. However, the following theorem (which will be proved below) takes place.

Theorem 1: *Any everywhere defined string function is a generalized primitive recursive string function iff it is a primitive recursive string function in the usual sense.*

The proof of Theorem is based on Lemma 1 which will be considered below. We will use primitive recursive string functions which are defined by the following equalities (where $P_1,\ P_2,\cdots,P_k,P_m,P,Q$ are any words in A).

1. The function $P\dot{-}a_1$ is defined as follows:
   $\Lambda\dot{-}a_1 = \Lambda,$
   $Pa_1\dot{-}a_1 = P,$
   $Pa_i\dot{-}a_1 = \Lambda,$ (where $2 \leq i \leq p$).
2. The function $Sg(P)$ is defined as follows:
   $Sg(\Lambda) = \Lambda,$
   $Sg(Pa_i) = a_1$ (where $1 \leq i \leq p$).

3. The function $\overline{Sg}(P)$ is defined as follows:
   $\overline{Sg}(\Lambda) = a_1,$
   $\overline{Sg}(Pa_i) = \Lambda,$ (where $1 \leq i \leq p$).

4. The functions $\Pi_{\mathrm{k}}(P_1,P_2,\cdots,P_k)$ for $k \geq 2$ are defined as follows:
   $\Pi_{2}(P_1,\Lambda) = \Lambda,$
   $\Pi_{2}(P_1,Qa_i) = P_1$ (where1 $\leq i \leq p$),

   $\Pi_{3}(P_1,P_2,\Lambda) = \Lambda,$
   $\Pi_{3}(P_1,P_2,Qa_i) = \Pi_{2}(P_1,P_2)$ (where $1 \leq i \leq p$),
   $\vdots$
   $\Pi_{\mathrm{m+1}}(P_1,P_2,\cdots,P_m,\Lambda) = \Lambda$ (where $m \geq 1$),
   $\Pi_{\mathrm{m+1}}(P_1,P_2,\cdots,P_m,Qa_i) = \Pi_{\mathrm{m}}(P_1,P_2,\cdots,P_m)$ (where1 $\leq i \leq p$),

   $\vdots$

It is easily seen that $\Pi_{\mathrm{k}}(P_1,P_2,\cdots,P_k) = \Lambda$ when one of the words $P_2,\cdots,P_k$ is equal to the empty word Λ: otherwise $\Pi_{\mathrm{k}}(P_1,P_2,\cdots,P_k) = P_1$.

A generalized primitive recursive string function$BR(P,Q)$(“Branching function”) is defined by the following conditions: (1) $BR(P,\Lambda) = P$:(2) $BR(P,Q)$ is undefined when $Q \neq \Lambda$. . Such a function is obtained by the operator of alphabetic primitive recursion using the everywhere undefined basic function $U(P)$:
$BR(P,\Lambda) = P,$
$BR(P,Qa_i) = U(I_3^1(P,Q,BR(P,Q)))$(where $1 \leq i \leq p$).

Now let us introduce the notion of *standard image* (or *S-image*) of string function F in A. Namely for any n-dimensional string function F in A its S-image is defined as a function F^* such that for any words $P_1,P_2,\cdots,P_n$ in A:

$$F^*(P_1,P_2,\cdots,P_n) = \begin{cases} S_1(F(P_1,P_2,\cdots,P_n)), & \text{when} \quad !\,F(P_1,P_2,\cdots,P_n), \\ \Lambda, & \text{otherwise.} \end{cases}$$

(let us recall that$S_1(Q) = Qa_1$for any word Qin A). Obviously, for any string function F in A the function F^* is an everywhere defined string function.

Lemma 1: *Any string function F in A is a generalized primitive recursive string function if and only if its S-image F* is a primitive recursive string function in the usual sense.*

Proof: Let F be an n-dimensional generalized primitive recursive string function. We will prove that its S-image F^* is a primitive recursive string function in the usual sense. The proof will be implemented using the induction on the process of constructing F from the basic functions by the operators of substitution and alphabetic primitive recursion.

If F is a basic function then it has one of the forms $D(P)$, $I_m^n(P_1, P_2, \cdots, P_n)$, (where $n \geq 1, 1 \leq m \leq n$),$S_i(P)$(where $1 \leq i \leq p$), $U(P)$ which is everywhere undefined. It is easily seen that in these cases F^* has correspondingly the following forms $D^*(P) = a_1$, $(I_m^n)^*(P_1, P_2, \cdots, P_n) = P_m a_1$, $S_i^*(P) = P a_i a_1$, $U^*(P) = \Lambda$. So F^* is a primitive recursive string function in the usual sense.

Now if a function $F \in GPRSF$ is obtained by the operator $\boldsymbol{S}$ of superposition from functions $G, H_1, H_2, \cdots, H_k$, and the functions $G^*, H_1^*, H_2^*, \cdots, H_k^*$ are primitive recursive string functions in the usual sense, then the S-image F^* of the function F satisfies the following equation:

$$F^*(P_1, P_2, \cdots, P_n) =$$
$$\Pi_{k+1}(G^*(H_1^*(P_1, P_2, \cdots, P_n) \dot{-} a_1,\ H_2^*(P_1, P_2, \cdots, P_n) \dot{-} a_1, \cdots, H_k^*(P_1, P_2, \cdots, P_n) \dot{-} a_1),$$
$$H_1^*(P_1, P_2, \cdots, P_n) \dot{-} a_1,\ H_2^*(P_1, P_2, \cdots, P_n) \dot{-} a_1, \cdots, H_k^*(P_1, P_2, \cdots, P_n) \dot{-} a_1)),$$

where Π_{k+1} is the function described above. It is easily seen that $F^* \in PRSF.$

Finally, if a function $F \in GPRSF$ is obtained by the operator $\boldsymbol{R}$ of alphabetic primitive recursion from functions $G, H_1, H_2, \cdots, H_p$, and the functions $G^*, H_1^*, H_2^*, \cdots, H_p^*$ are primitive recursive string functions in the usual sense, then the S-image F^* of the function F satisfies the following equalities:

$$F^*(P_1, P_2, \cdots, P_n, \Lambda) = G^*(P_1, P_2, \cdots, P_n),$$
$$F^*(P_1, P_2, \cdots, P_n, Pa_1) = H_1^{**}(P_1, P_2, \cdots, P_n, P, F^*(P_1, P_2, \cdots, P_n, P)),$$
$$\vdots$$
$$F^*(P_1, P_2, \cdots, P_n, Pa_p) = H_p^{**}(P_1, P_2, \cdots, P_n, P, F^*(P_1, P_2, \cdots, P_n, P)),$$

where for any i such that $1 \leq i \leq p$,

$$H_i^{**}(P_1, P_2, \cdots, P_n, P, Q) = \Pi_2(H_i^*(P_1, P_2, \cdots, P_n, P, Q \dot{-} a_1), Q),$$

and G^*, H_i^*, $1 \leq i \leq p$, are S-images correspondingly, of G, H_i; the function Π_2 is defined above. It is easily seen that $F^* \in PRSF.$

So, it is proved that S-image of any generalized primitive recursive string function is a primitive recursive string function in the usual sense.

Now let us suppose that the S-image F^* of some n-dimensional string function F is a primitive recursive string function in the usual sense. Clearly, $F^* \in GPRSF.$ Then the function F satisfies the following equality:

$$F(P_1, P_2, \cdots, P_n) = BR(F^*(P_1, P_2, \cdots, P_n) \dot{-} a_1, \overline{Sg}(F^*(P_1, P_2, \cdots, P_n))),$$

were BR is the function defined above. This completes the proof of Lemma.

The proof of Theorem 1 is obtained now as follows. If F is an n-dimensional everywhere defined string function such that $F \in GPRSF.$ then $F^* \in PRSF.$ and $F(P_1, P_2, \cdots, P_n) = F^*(P_1, P_2, \cdots, P_n) \dot{-} a_1$, for any words $P_1, P_2, \cdots, P_n$ in A. Hence, $F \in PRSF.$ On the other side, if $F \in PRSF.$ then clearly F is an everywhere defined string function such that $F \in GPRSF.$ This completes the proof of Theorem.

References

[1] A. I. Malcev, *Algorithms and Recursive Functions*, $2^{\underline{nd}}$ edition, Moscow, "Nauka", (in Russian), 1986.

[2] M. H.Khachatryan, "On the representation of arithmetical and string functions in formal languages", *Transactions of the IIAP of NAS of RA, Mathematical problems of computer science*, vol. 27, pp. 37-53, 2006.

[3] I.D. Zaslavsky, "On some generalizations of the primitive recursive arithmetic", *Theoretical Computer Science*, vol. 322, pp. 221-230, 2004.

Submitted 10.12.2014, accepted 16. 02. 2015.

Ընդհանրացված պարզագույն կարգընթաց բառային ֆունկցիաների մասին

Մ. Խաչատրյան

Ամփոփում

Սահմանվում է ընդհանրացված պարզագույն կարգընթաց բառային ֆունկցիայի հասկացությունը, ինչպես նաև հետազոտվում են այդպիսի ֆունկցիաների փոխառնչությունները սովորական ձևով սահմանված ([1], [2]) պարզագույն կարգընթաց բառային ֆունկցիաների հետ: Ապացուցվում է, որ ցանկացած ընդհանրացված պարզագույն կարգընթաց բառային ֆունկցիա ամենուրեք որոշված է այն և միայն այն դեպքում, երբ այն սովորական իմաստով պարզագույն կարգընթաց բառային ֆունկցիա է:

Об обобщенных примитивно рекурсивных словарных функциях

М. Хачатрян

Аннотация

Определяется понятие обобщенной примитивно рекурсивной словарной функции и исследуются взаимоотношения таких функций с примитивно рекурсивными словарными функциями ([1], [2]) в обычном смысле этого понятия. Доказывается, что обобщенная примитивно рекурсивная словарная функция всюду определена тогда и только тогда, когда она является примитивно рекурсивной словарной функцией в обычном смысле этого понятия.

A Modified Fuzzy Vault Scheme for Increased Accuracy

Hovik G. Khasikyan

Institute for Informatics and Automation Problems of NAS RA
e-mail: hkhasikyan@aua.am

Abstract

In this paper a new "Fuzzy Vault" scheme is proposed, which improves the False Rejection Rate of the original scheme. When constructing the vault of the original scheme there was a need to trim the biometric data, which is an information loss and affects the performance of the system. This was resolved in the suggested version of this construction. The schemes were implemented for fingerprint data, and the comparisons are brought in the last section of this paper.

Keywords: Biometrics, Fuzzy Vault, Fingerprints.

1. Introduction

Conventional passwords are usually simple and, as a rule, easy to guess or to break. People remember only short passwords. What is more, they tend to choose passwords, which are easily cracked by dictionary attacks [1, 2, 3]. Thus, there was a suggestion to use some biometric properties of a user to provide an access to the personal data. The biometric characteristics of a person, such as DNA, palm vein, fingerprints, face and iris features can be used to generate passwords or lock secrets.

Such schemes were introduced by A. Juels and M. Wattenberg [4], which was not order invariant, and this was the weakest point of the algorithm described in [4] as the data extracted from the biometric template is not in the same order each time. Thereafter A. Juels and M. Sudan presented a new scheme called "A Fuzzy Vault Scheme"[5], which already had a property of order invariance. The notion of *fuzzy vault* was first given by Juels and Sudan. For analysis of the concepts False Acceptance Rate (FAR) and False Rejection Rate (FRR) are used.

FAR is the probability that a random vector is accepted as valid biometric data at the authentication phase.

FRR is the probability that the observed genuine biometric data has too many errors and is rejected at the authentication phase.

The scheme in [5] can be modified for decreasing the False Rejection Rate (FRR) while keeping the FAR of the system the same.

The rest of this paper is organized as follows. Section 2 gives a review of the fuzzy vault construction. Section 3 outlines the modified construction of Fuzzy Vault. In section 4 the experimental results of the two schemes are introduced. Section 5 gives the summary of the paper.

2. Review of Fuzzy Vault

The fuzzy commitment scheme is presented by Juels and Wattenberg [4], which as it was already mentioned is not order invariant. Order invariance is a very important property, because not always we can obtain biometric data of a user in the same order. Then Juels and Sudan presented their new Fuzzy Vault construction [5]. The brief description of the scheme is given below.

Let F be a finite field of size n. The biometric template of the user can be written as follows: $w = (x_1, x_2, \dots, x_s),\ where\ \forall i = 1, \dots, s: x_i \in F$ and let $r \in \{s + 1, \dots n\}$.

A. Enrollment Phase

1. Take the secret polynomial *p(x)* of degree $\mathrm{k} = \mathrm{s} - \mathrm{t} - 1,\ t \in \{1, \dots, s\}$ and evaluate it on the points of the biometric data. Let $y_i = p(x_i),\ i = 1,2 \dots, s$.

2. Add $r - s$ distinct random points from the set $F - w$. Let them be $x_{S+1}, \dots, x_r$. These points are called chaff points.

3. Choose y_i ∈F ,$i = s + 1 \dots r$ such that $y_i \neq p(x_i)$.
4. Store $ss(w) = \{(x_1, y_1), \dots, (x_r, y_r)\}$ as a reference. The $ss(w)$ is called a vault.

B. Authentication Phase

Let the new biometric be $w' = (x'_1, x'_2, \dots, x'_s)$. If it has at least $s - t$ common points with the original biometric using Lagrange interpolation or Reed Solomon codes the secret polynomial can be reconstructed.

3. The Proposed Scheme

Again F is a finite field of size n. The biometric template for enrollment is $w = (x_1, x_2, \dots, x_s)$, however, in this scheme the condition $x_i \in F$ is not mandatory. The secret polynomial is the *p(v)*. The degree of *p(v)* is $\mathrm{k} = \mathrm{s} - \mathrm{t} - 1, \mathrm{t} < s$, and the coefficients belong to F. The enrollment and authentication phases are the following.

A. Enrollment Phase

1. Take the secret polynomial *p(v)*, generate s random values $q = (v_1, v_2, \dots, v_s)$, where $v_i \in F$ and evaluate the *p(v)*on$q$. Let $y_i = p(v_i),\ i = 1,2 \dots, s$.

2. Add $r - s$ distinct random points from the set $F - q$. Let them be $v_{S+1}, \dots, v_r$. Add $r - s$ distinct random points, which are not in the set of w, but are within the possible set of the points of the considered biometric data. Let them be $x_{S+1}, \dots, x_r$.

3. Choose y_i ∈F ,$i = s + 1 \dots r$ such that $y_i \neq p(v_i)$.
4. Store $\{(x_1, y_1, v_1), \dots, (x_r, y_r, v_r)\}$ as a reference in database. Let's denote this vault by $ms(w)$.

B. Authentication Phase

Now suppose the new biometric measurement is $w' = (x'_1, x'_2, \dots , x'_s)$ and we want to recover the secret polynomial *p(v)*. Thus, in case the w'coincides with at least $s - t$ points with original biometrics, the corresponding triplets(x_i, y_i, v_i) can be chosen from the vault $ms(w)$ and using the pairs (y_i, v_i) the secret can be recovered.

The advantage of this scheme is that in this case there is no need to concatenate the biometric template, as it should be done with the most types of biometrics for the fuzzy vault [6, 7, 8]. In addition, the user is free to choose the Galois Field he wants to use in this system. As a result of these modifications there is no information loss in the enrollment stage, which leads to the increased accuracy of verification.

4. Experimental Results for Fingerprints

The experiments were conducted on GF(2^{16}). In the case of the original scheme the minutiae point descriptors are formed by concatenating some parts of x, y coordinates and some part of the minutiae angle θ. In order to form a 16 bit value, 5 bits were taken for x coordinate, 5 bits for y and 6 bits for θ angle.

In the case of the new scheme, the coordinates are kept in the original form and the 16 bit values are random. In all experiments FVC 2000 DB2 pre-aligned database of fingerprints was used. The results of the experiments are illustrated in the Table1.

Table 1

	Juels-Sudan scheme	*The Proposed scheme*
FAR (%)	0.2%	0.2%
FRR (%)	20.4%	13.8%
secret size	128 bit	128 bit
reference size	960 Byte	1700 Byte
r (the size of vault)	240	240

5. Conclusion

In this paper an improvement was suggested for increasing the performance of a well-known scheme for biometric key binding. It was shown that during the vault construction the concatenation of biometric data affects the accuracy of the system. To overcome this issue, in the new scheme the reference data were kept as triplets; first is the biometric data in its original form, the second is a random value and the third is the evaluation of the value. The scheme was implemented for fingerprints and the experiments have shown that it provides better FRR, while maintaining the FAR of the system the same.

Acknowledgement

This work was supported by the State Committee Science MES RA, in the frame of the research project SCS 13-1B352.

References

[1] E. Spafford, “Observations on reusable password choices”,*Proceedings of the 3rd USENIX Security Symposium*, September, pp. 299--312, 1992.

[2] T. Wu, “A real-world analysis of Kerberos password security”, Proceedings of the 1999 Network and Distributed System Security Symposium,February 1999.

[3] R. Morris and K. Thompson, “Password security: A case history,” *Communications of the ACM*, vol. 22, no. 11, pp. 594–597, 1979.

[4] A. Juels and M. Wattenberg, “A fuzzy commitment scheme”,*Proceedings of the 6th ACM Conference on Computer and Communications Security*, pp. 28–36, 1999.

[5] A. Juels and M. Sudan, “A fuzzy vault scheme”, *Proceedings of IEEE International Symposium of Information Theory*, Lausanne, Switzerland, p. 408, 2002.

[6] U. Uludag, S. Pankanti and A. K. Jain, “Fuzzy vault for fingerprints”, *Proceedings of Audio- and Video-Based Biometric Person Authentication*, Rye Town, NY, pp. 310–319, July 2005.

[7] Y. J. Lee, K. Bae, S. J. Lee, K. R. Park and J. Kim, “Biometric key binding: fuzzy vault based on iris images”,*Proceedings of 2nd International Conference on Biometrics, Seoul*, South Korea, pp. 800–808, August 2007.

[8] A. Kumar and A. Kumar, “Development of a new cryptographic construct using palmprint-based fuzzy vault”, *EURASIP Journal on Advances in Signal Process*,vol. 2009, Article ID 967046, 11 pages, 2009.

Submitted 10.11.2014, accepted 12. 02. 2015.

Ձևափոխված թերորոշ բանալային պահոցով սխեմա ճշգրտության բարձրացման համար

Հ. Խասիկյան

Ամփոփում

Այս աշխատանքում առաջարկվում է «Fuzzy Vault» սխեմայի մոդիֆիկացված տարբերակ, որը բարելավում է օրիգինալ սխեմայի սխալ մերժման գործակիցը: Օրիգինալ սխեմայի կառուցման ժամանակ կարիք կար կարճացնել կենսաչափական տվյալները, ինչը ինֆորմացիայի կորուստ էր և ազդում էր համակարգի արդյունավետության վրա: Այս խնդիրը լուծվել է սխեմայի նոր առաջարկվող տարբերակում: Սխեմաները իրականացվել են մատնահետքային տվյալների հայտնի պահոցների վրա, իսկ համեմատությունները բերված են վերջին բաժնում:

Модифицированный вариант схемы "нечеткого хранилища" для увеличения точности

О. Хасикян

Аннотация

В этой работе предлагается модифицированный вариант схемы "нечеткого хранилища", в котором улучшается коэффициент ложного отказа в доступе. При построении оригинальной схемы была необходимость в сокращении биометрических данных, что само по себе потеря информации и влияет на эффективность системы. Эта проблема разрешена в новом предложенном варианте схемы. Схемы были реализованы для известных баз отпечатков пальцев, а сравнения приведены в последнем разделе.

Mathematical Problems of Computer Science 43, 52--56, 2015.

Reconstruction of Distorted Images

Souren B. Alaverdyan

Institute for Informatics and Automation Problems of NAS RA
e-mail: souren@ipia.sci.am

Abstract

The non-focused images quality arising algorithm based on Winer filtration is presented in the paper. Filtration is realized in spectral domain of image.

Keywords: Filter, Image, Spectrum, Wiener.

1. Introduction

During the image registration there often appear distortions of different types depending on registering devices characteristics (permission), technical situation of location and also peculiarities of the area being registered. On images obtained by optical devices there can be violations of focal distance; there can appear diffusions when moving objects are being registered, etc.

We consider the image as a function of two variables $f(x,y)$ which is the projection of two or three dimensional fields of view, where (x,y) is a coordinate of any point of plane and $f(x,y)$ is the light intensity in the point (x,y).

We'll consider a problem of optically [1] registered distorted images reconstructon in spectral area, because optical systems focus the falling light and that can be expressed by Fourie transform, so the image reconstruction problem reduces the solving of integral equations of second order.

2. Image reconstruction

Let $g(x,y)$ be the given image and $f(x,y)$ be the reconstructing image. Then the following equation [2] takes place:

$$g(x,y)=\iint f(u,v)h(x,y,u,v)dudv, \qquad (1)$$

where the function $h(x,y,u,v)$ is called an image registering system's impulse response (output value corresponding to unit impulse).

To solve this equation we'll give some assumptions.

Definition 1: *The system is called space-invariant, if its impulse function response depends on the difference between the input*(x, y) *and output*(x, y) *planes coordinates:*

$$h(x, y, u, v) = h(x - u, y - v).$$

For such system the equation (1) will be represented as

$$g(x, y) = \iint f(u, v)h(x - u, y - v)dudv, \tag{2}$$

which is usually called a convolution. Equation (2) can also be represented as

$$g(x, y) = f(x, y) * h(x, y). \tag{3}$$

Since $f(x, y)$ is a function of image describing the range of vision, and $g(x, y)$ is a function of registered image, we can see that $h(x, y)$ is a noise describing function.

In general case linear filtration algorithms are realized by transforms of type (2) having the following discrete representation

$$g_{i,j} = \sum_{k=i-r/2}^{i+r/2} \sum_{l=j-r/2}^{j+r/2} f_{k,l} h_{k-i+{}^{r}/_{2}, l-j+{}^{r}/_{2}}, \; i \in \left[{}^{r}/_{2}, M + {}^{r}/_{2}\right], j \in \left[{}^{r}/_{2}, N + {}^{r}/_{2}\right]. \tag{4}$$

M is the number of image rows, N is the number of image columns, the sum includes the points of rectangular with centre(i, j) and $2r + 1$ sides. Before calculation of transform (4) all sides of image should be already widened by rectangular layers of width $r/2$.

In spectral domain the linear filtration algorithm is also based on convolution theorem, so instead of calculating by formula (4) it can be realized by the following formula:

$$G(u, v) = F(u, v)H(u, v), \tag{5}$$

where G, F, H are Fourier transforms of functions g, f, h. Note, that complex multiplication is realized by all u, v frequencies.

Now we'll represent the mathematical model of the system:

$f(x, y)$ -input image function (undistorted),

$h(x, y)$ - noise causing function,

$n(x, y)$ - total noise,

$g(x, y)$ - distorted image (fuzzified, unfocused).

So we have the following representation of the process:

$$g(x, y) = f(x, y) * h(x, y) + n(x, y). \tag{6}$$

It is required to find the impulse characteristic function which will be for the system the best reconstruction function by mean square deviation

$$\sigma = \sqrt{\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left(\hat{f}_{i,j}\text{-}f_{i,j}\right)^2} \to min.$$

The problem solution for linear stationary processes was given by Wiener, the detailed proof is given in [3]. The best approximating filter's spectral representation of function $f(x,y)$ is represented as [3]

$$\hat{F}(u,v) = \frac{1}{H(u,v)} \cdot \frac{|H(u,v)|^2}{|H(u,v)|^2 + {S_n(u.v)}/{S_f(u.v)}} G(u,v), \tag{7}$$

where $S_n(u.v)$ is the spectral density of additive noise and $S_f(u.v)$- $f(x,y)$ is the spectral density of the function. Generally these values are unknown. The ratio $S_n(u.v)/S_f(u.v)$ is the inverse value of signal-noise value. Its value in time domain is considered acceptable if it is in the interval of 30-40 decibels.

The noises induced by focal distance violations on the images registered by the optical devices mainly depend on the light dispersion problem described by the following two functions:

$$h(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}},$$

$$h(x,y) = \begin{cases} \frac{1}{\pi r^2}, & if\ x^2 + y^2 < r^2 \\ 0, & if\ x^2 + y^2 \geq r^2. \end{cases}$$

If the image doesn't include an additive noise then $n(x,y) = 0$ and the formula (7) is represented as

$$\hat{F}(u,v) = G(u,v)/H(u,v)\,, \tag{8}$$

and is called an inverse filter.

3. Inverse Filters

Indeterminacy appears when because of some device errors during image registering under some frequencies the value of denominator $H(u,v)$ of equation (8) is equal to 0. In such cases the value of spectrum corresponding to this value of image is set equal to zero. As a result, on the filtered image there appear obvious horizontal or vertical (sometimes curved) phenomena.

To reduce such occurrences we offer to realize the low- frequency interpolation in spectral domain:

$$\widehat{F}(u,v) = \sum_{i=u-w}^{u+w}\sum_{j=v-w}^{u+w} s_{i,j,} \tag{9}$$

where

$$s_{i,j} = \begin{cases} F(i,j)\dfrac{\sin(2\pi f i)}{i}\dfrac{\sin(2\pi f j)}{j}, & if\ i > 0, j > 0, \\ 2\pi f F(i,j), & if\ i = 0\ or\ j = 0, \\ 0, & if\ i < 0\ or\ j < 0. \end{cases}$$

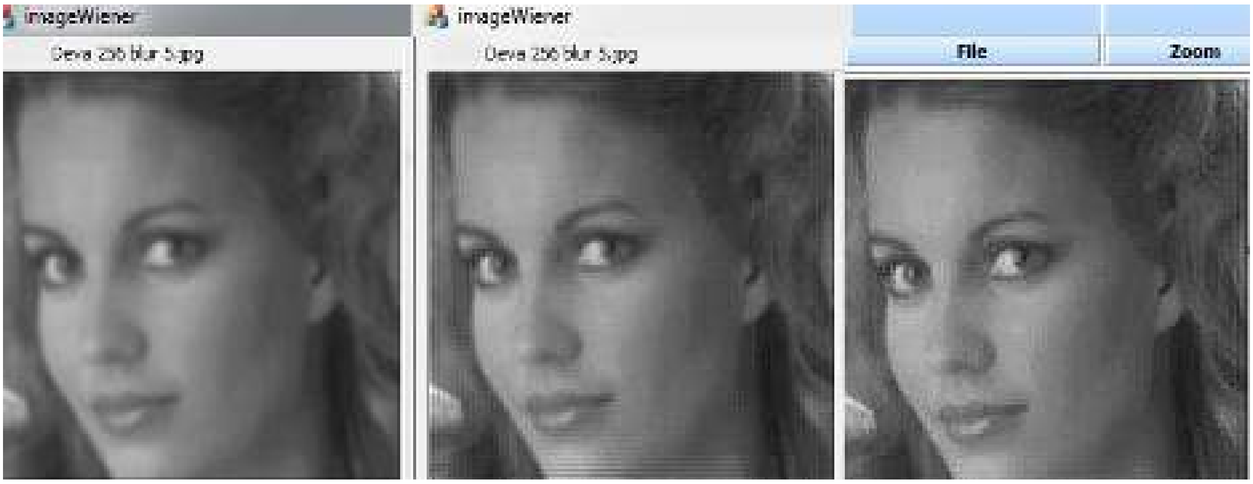
In case of $i = w, j = w$, $\hat{F}(u,v) = 2\pi f,\ f \in (0;\ 0{,}5)$.

There are many internet investigations and program realizations of this problem.

I think, the system SmartDeblur-1.27-win is one of the best program realizations, but its mathematical apparatus is not presented in the work.

The program realization of the method(5)-(9) presented in this paper has been fulfilled.

The result of the system work and comparison with system SmartDeblur-1.27-win [4] are given below.



a b c

Fig. 1.

a) input image including gauss noise with domain of dispersion $\sigma = 3$ and radius $r = 5, f = 0.45$;

b) the result of program realization of developed system;

c) the result of SmartDeblur-1.27-win system work.

References

[1] X. Chen, J. Yong,“Image deblur in gradient domain”,*Optical Enginering*, vol. 49, no. 11,117003, 2010.

[2] У. Пратт, *Цифровая обработка изображений*, Мир, Москва 1982.

[3] Н. Ахмед, К.Р. Рао,*Ортогональные преобразования при обработке цифровых сигналов*, Москва, “Связь”, стр. 164-181, 1980.

[4] [Online]. Available: http://smartdeblur.net/

Submitted 10.12.2014, accepted 12. 02. 2015.

Չֆոկուսացված պատկերների որակի բարձրացում

Ս. Ալավերդյան

Ամփոփում

Աշխատանքում ներկայացվում է չֆոկուսացված պատկերների որակի բարձրացման նոր ալգորիթմ, որը հիմնված է Վիների ֆիլտրացիայի վրա: Ֆիլտրացիան իրականացվում է պատկերի սպեկտրալ տիրույթում:

Улучшение качества расфокусированных изображений

С. Алавердян

Аннотация

В работе представляется новый алгоритм улучшения качества расфокусированных изображений, который основан на фильтре Винера. Фильтрация выполняется в спектральной области изображения.

Mathematical Problems of Computer Science 43, 57--61, 2015.

Encoding and Decoding Procedures for Double ± 1 Error Correcting Linear Code over Ring Z_5

Hamlet K. Khachatryan

Institute for Informatics and Automation Problems of NAS RA
e-mail: hamletxachatryan.08@gmail.com

Abstract

From practical point of view the codes over Z_{2m} or Z_{2m+1} are interesting, because they can be used in 2^{2m} **–QAM (Quadrature amplitude modulation) schemes.** In this paper a construction of encoding and decoding procedures for double ± 1 error correcting optimal(12,8) linear code over ring Z_5 is presented.

Keywords: Error correcting codes, Codes over the ring Z_5, Encoding and Decoding procedures.

1. Introduction

Codes over finite rings, particularly over integer residue rings and their applications in coding theory have been studied for a long time. Errors happening in the channel are basically asymmetrical; they also have a limited magnitude and this effect is particularly applicable to flash memories.There are many constructed codes capable to correct up to two errors of value ± 1 .The earliest paper discussing the codes over the ring Z_A of integers modulo A are due to Blake [1], [2].

The optimality criteria for the linear code over fixed ring Z_m was considered in 2 ways in [3]. First of all, recall that the code of the length n is optimal-1 if it has a minimum possible number of parity check symbols. Secondly, optimality-2 criteria for the code is that for a given number of parity check symbols it has a maximum possible length. Here, we propose to construct encoding and decoding algorithms for the optimal codes. The code presented in this paper satisfies the optimality criteria-1([3]). At this point we do not know any codes that satisfy the optimality criteria-2. There have been encoding and decoding procedures for the (4, 2) code over ring Z_9 in [4]. Implementation of codes over large alphabets is much more difficult compared with small alphabets. In this paper a construction of encoding and decoding procedures of (12, 8) linear code over ring Z_5 correcting double ± 1 errors is presented.

2. Presentation of the Code

Let a linear (12, 4) code over ring Z5 be given by the following parity check matrix H:

$$H=\begin{bmatrix}1&1&1&1&1&0&1&2&3&4&1&1\\0&1&2&3&4&2&2&2&2&2&1&1\\3&2&4&4&2&3&2&4&4&2&1&1\\1&1&1&1&1&3&2&4&4&2&0&4\end{bmatrix}.$$

A linear code over Z5 given by the parity check matrix H can correct up to two errors of the type ± 1, because H has a property according to which all the syndromes resulting from adding and subtracting operations between any two columns of the matrix H are different ($\pm h_i \pm h_j$ and $h_i \neq h_j$)(proof of this you can see in [3]).

In this case the number of combinations for each code word that can be corrected is $(1 + 12 * 2 + (12\, choose\, 2) * 4) = 289$.

For encoding every vector in Z_5 we should have the generator matrix G. For this we should construct a combinatorial equivalent matrix H' from matrix H.

$$H'=\begin{bmatrix}1&0&0&0&3&1&0&3&4&3&4&0\\0&1&0&0&2&2&4&4&0&2&4&3\\0&0&1&0&0&0&2&4&2&1&3&1\\0&0&0&1&1&3&2&0&4&4&3&3\end{bmatrix}.$$

In this matrix all 289 possible syndromes will be different, too.

From [5] we know, that

$$GH'^T = 0. \tag{1}$$

If $\mathrm{H}' = [-\mathrm{P}^{\mathrm{T}}|\mathrm{I}_{\mathrm{n-k}}]$, then $\mathrm{G} = [\mathrm{I}_{\mathrm{k}}|\mathrm{P}]$ (the reverse statement is also true), where I_{k} is a $k * k$ identity matrix and P is a $k * (n-k)$ matrix[5].

Thus, we can construct the generator matrix G:

$$G=\begin{bmatrix}2&3&0&4&1&0&0&0&0&0&0&0\\4&3&0&2&0&1&0&0&0&0&0&0\\0&1&3&3&0&0&1&0&0&0&0&0\\2&1&1&0&0&0&0&1&0&0&0&0\\1&0&3&1&0&0&0&0&1&0&0&0\\2&3&4&1&0&0&0&0&0&1&0&0\\1&1&2&2&0&0&0&0&0&0&1&0\\0&2&4&2&0&0&0&0&0&0&0&1\end{bmatrix}.$$

3. Encoding and Decoding Procedures

3.1 Encoding Procedure

In our scheme the message was presented by 8-tuples in Z5. Let G be a generator matrix for (12,8) linear code. $v = (a_1, a_2, a_3, \dots, a_8)$ is an arbitrary 8-tuple, and consider the vector u that is the linear combination of columns G with a_i is the i^{th} coefficient.

$$u = vG = (c_1, c_2, c_3, c_4, a_1, a_2, a_3, \dots, a_8),$$

where the first 4 components of the code vector are the check symbols, the next 8 components are information symbols and

$$c_j = \left(\sum_{i=1}^{k} a_i p_{ij}\right) mod 5. \qquad (2)$$

Example.
Let (3 4 0 0 2 1 1 4) be the message vector in Z_5. From (2) we can obtain check symbols. For example, the first check symbol is c1:

$$c_1 = (3*2) + (4*4) + (0*0) + (0*2) + (2*1) + (1*2) + (1*1) + (4*0) =$$
$$= 6 + 16 + 0 + 0 + 2 + 2 + 1 + 0 = 27 mod 5 = 2\,.$$

Similarly, we can find other 3 check symbols:

$$c_2 = 3, \; c_3 = 3, \; c_4 = 3.$$

After performing other multiple operations with matrix G we obtain this encoded vector: (2 3 3 3 3 4 0 0 2 1 1 4).

3.2 Decoding Procedure

In this section we describe the decoding procedure:

1. Receiver multiplies the vector with every column of matrix H' and gets the syndrome $\mathrm{S} = \mathrm{vH}'$. If $\mathrm{S} = (0,0,0,0)$ then there were not any errors in the received vector.
2. If the calculated syndrome S is a nonzero vector, then there are some errors. This (12,8) code can correct only up to two errors with magnitude ±1. We know that all possible syndromes of matrix H' are different $(\pm \mathrm{h_i}, \pm \mathrm{h_j}$ and $\mathrm{h_i} \neq \mathrm{h_j})$ (the number of them is 288 and syndrome (0,0,0,0)). After calculating the syndrome the receiver knows from which two columns of the matrix H' the syndrome was resulted, consequently, he can find the two corresponding components of the vector, where the error was occurred and the direction of the error (if $+\mathrm{h_i}$, then upward direction or if $-\mathrm{h_i}$ downward direction). On the other hand, if in the table of syndromes we do not have the resulted syndrome, then we cannot correct this kind of errors.
3. After finding the error components the receiver adds or subtracts 1 from them (he adds if downward, else subtracts) and obtains the sent code vector $(\boldsymbol{c_1, c_2, c_3, c_4, a_1, a_2, a_3, \dots, a_8})$. So $(\boldsymbol{a_1, a_2, a_3, \dots, a_8})$ ***is our message vector.***

Example.

$(2\ 3\ 3\ 3\ 3\ 4\ 0\ 0\ 2\ 1\ 1\ 4)$ is an encoded vector from the previous example. Let there occur 2 errors in the channel, and the receiver get the vector $(2\ 3\ 3\ 3\ 3\ 4\ 0\ 4\ 2\ 2\ 1\ 4)$. After performing multiple operations with rows of matrix $H'\ the$ receiver obtains the syndrome $(0\ 3\ 2\ 4)$. Next from the table of syndromes he finds the corresponding columns, now they are -8 and 10. Hence, the syndrome $(0\ 3\ 2\ 4)$ was resulted from adding a negated column 8 of matrix H to column

$$\begin{matrix}-3\\-4\\-4\\0\end{matrix}\quad\begin{matrix}+3\\+2\\+1\\+4\end{matrix}=\begin{matrix}0\\-2\\-3\\4\end{matrix}(mod 5)\ =\ (0\ 3\ 2\ 4),$$

(because in $Z_5, 0\ =\ 5,\ -1\ =\ 4,\ -2\ =\ 3\ ,\ -3\ =\ 2,\ -4\ =\ 1)$.
Hence, the error positions of encoded vector are 8 and 10 (in 8^{th} downward direction and in 10^{th} upward).
So, he adds 1 to 8^{th} component and subtracts 1 from 10^{th} of vector $(2\ 3\ 3\ 3\ 3\ 4\ 0\ 4\ 2\ 2\ 1\ 4)$ and obtains the sent encoded vector $(2\ 3\ 3\ 3\ 3\ 4\ 0\ 0\ 2\ 1\ 1\ 4)$. Consequently, the message vector is $(3\ 4\ 0\ 0\ 2\ 1\ 1\ 4)$.

4. Conclusion

In this paper a construction of encoding and decoding procedures of optimal-1 (12,8) linear code over ring Z_5 correcting double ± 1 errors is presented. We propose that this approach can be extended for constructing similar procedures for the optimal codes over other rings Z_n and the research in this direction will follow.

References

[1] I. Blake, “Codes over certain rings”, *Information and Control,* vol. 20, pp. 396-404, 1972.

[2] I. Blake, “Codes over integer residue rings”, *Information and Control*, vol. 29, 1975, 295-300.

[3] G. Khachatrian and H. Morita, “Construction of optimal 1 double error correcting linear codes over ring Z_5”, *3th International Workshop on Advances in Communications*, Boppard, Germany, pp. 10-12, May 2014.

[4] H. Kostadinov, N. Manev and H. Morita, “Double ±1-error correctable codes and their applications to modulation schemes”, *Proc. Elev. Intern. Workshop ACCT*, Pamporovo,June 16-22, pp 155-160,2008.

[5] W. W. Peterson and E.J. Weldon, *Error-Correcting Codes*, M.I.T. Press, Cambridge, Mass.,1972.

Submitted 15.10.2014, accepted 12. 02. 2015.

Կոդավորման և ապակոդավորման ալգորիթմը Z_5 օղակում ±1 մեծությամբ կրկնակի սխալ ուղղող կոդերի համար

Հ. Խաչատրյան

Ամփոփում

Պրակտիկ տեսանկյունից մեծ հետաքրքրություն են առաջացնում $\boldsymbol{Z_{2m}}$ կամ $\boldsymbol{Z_{2m+1}}$ օղակների վրա դիտարկված կոդերը, քանի որ նրանք ունեն լայն կիրառություն $\mathbf{2^{2m}}$ **–QAM** մոդուլյացիոն սխեմաներում: Այս հոդվածի շրջանակներում ներկայացված է կոդավորման և ապակոդավորման ալգորիթմը $\mathbf{Z_5}$ օղակում ± 1 մեծության մինչև 2 սխալ ուղղող օպտիմալ (12, 8) գծային կոդի համար:

Алгоритм кодирования и декодирования в кольце Z_5 для кодов исправляющих двойные ошибки размера ±1

Г. Хачатрян

Аннотация

С практической точки зрения большой интерес вызывают коды на кольцах Z_{2m} и Z_{2m+1}, так как они имеют широкое применение в 2^{2m} **–КАМ** модуляционных схемах. В данной статье представлен алгоритм кодирования и декодирования в кольце Z_5 для оптимального (12,8) линейного кода, исправляющего до двух ошибок размера ± 1.

Mathematical Problems of Computer Science 43, 62--75, 2015.

Structuring of Goals and Plans for Personalized Planning and Integrated Testing of Plans

Sedrak H. Grigoryan

Institute for Informatics and Automation Problems of NAS RA
e-mail: addressforsd@gmail.com

Abstract

We study competition problems defined in the class where Space of Solutions is a Reproducible Game Tree (RGT). Personalized Planning and Integrated Testing algorithms were developed for searching optimal strategies in RGT problems. Hereinafter we develop structures for plans and goals in PPIT, construct strategy searching algorithms by plans and demonstrate their adequacy for chess endgame examples.

Keywords: Strategy, Plan, Competition, Chess, Goal.

1. Introduction

1.1. In [2] the variety of problems was identified as a class where *Space* of possible *Solutions* can be ***specified*** by *Reproducible* combinatorial *Game Trees* (RGT) and ***unified*** algorithms and software were developed, ***RGT Solver***, for elaborating optimal strategies for any input specified problem of the class.

The RGT is a spacious class of problems with only a few following requirements to belong to:

- there are (a) interacting actors (players, competitors, etc.) performing (b) identified types of actions in the (c) specified moments of time and (d) specified types of situations
- there are identified benefits for each of the actors
- the situations the actors act in and transformed after the actions can be specified by certain rules, regularities.

Many security and competition problems belong to RGT class. Specifically, these are network Intrusion Protection (IP), Management in oligopoly competitions and Chess-like combinatorial problems, many other security problems such as Computer Terrorism Countermeasures, Disaster Forecast and Prevention, Information Security.

1.2. Unified RGT specification of problems makes possible to design a unified Solver for the problems of the class.

Solver of the RGT problems is a package [7] aimed to acquire strategic expert knowledge to become comparable with a human in solving hard combinatorial competing and combating problems. In fact, the following three tasks of expert knowledge acquisition can be identified in the process:

- construction of the package of programs *sufficient* to acquire the ***meanings*** of the units of vocabulary (UV) of problems
- construction of procedures for *regular* acquisition of the meanings of UV by the package
- provision of means *measuring the effectiveness* of solutions of RGT problems.

The limitations in designing effective package were formulated as follows:

- be able to store typical categories of communalized knowledge as well as the personalized one and depend on them in strategy formation
- be able to test approximate knowledge-based hypothesis on strategies in questioned situation by reliable means, for example, using game tree search techniques.

The second task of *acquisition of complex expert knowledge* was planned to solve in the following two stages:

- *proving the sufficiency*, i.e. proving that Solver, in principle, can acquire the meanings of expert knowledge of an intensive RGT problem, e.g., for the kernel RGT chess game
- *ensuring regularity,* i.e. to develop procedures for regular acquisition of RGT problems and meanings of UV of those problems.

1.3. Regular improvement of Solver by expert knowledge is studied for chess, where the problems of knowledge representation and consistent inclusion into the programs stay central since the pioneering work by Shannon in 1950.

Players indicate and communicate chess knowledge by units of vocabulary and are able to form corresponding contents. Whether it is possible to form equal contents by computers remains questionionable.

The approaches to regular inclusion of chess knowledge into strategy formation process are described in [5]. Then try to bring common handbook knowledge to cut the search in the game tree. The frontiers of those approaches can be revealed by understanding the role and proportion of the personalized chess expertise compared with the common, communicable one.

1.4. Studies of knowledge-based strategies in the Institute for Informatics and Automation Problems of the National Academy of Sciences of the Republic of Armenia have been started in 1961 and noticeable results were published in the Laboratories of "Mathematical Logic" and "Cognitive Algorithms and Models" led by I. Zaslavski [1] and E. Pogossian [2, 3, 6].

Designed and developed PPIT (Personalized planning and integrated testing) [2, 6] algorithms indicate the optimal strategy by effective usage of expert knowledge. The algorithms had been tested for a variety of problems, for chess, Reti and Nodarishvili etudes [6], for intrusion protection problems [3]. In the PPIT algorithms predefined set of knowledge was used which was strongly specific to the solving problem and did not provide a generic and regular way to define knowledge and reveal strategies from them. This approach reduced the abilities of algorithm execution, since it required writing a new program to solve each certain situation and each of them was useful only for the given situations, so the program developed for Reti etude could be used only to solve this etude.

In the RGT Solver strategy searching algorithms were not yet suggested to provide general solution while plans used in PPIT algorithms are only generic descriptions of strategies.

In the following we describe planning-based strategy searching algorithms within the frame of Solver package.

In the first section we consider structuring of plans and goals. We need these structures for strategy generation algorithms. In the second one the algorithm that searches for a strategy to accomplish the plan and in the last section an example demonstrating adequacy of structures and strategy searching algorithm are described.

2. Contributing to Personalized Planning and Integrated Testing (PPIT) Algorithms

2.1. State of the Art

2.1.1. The Basics of PPIT

For the strategy construction we use PPIT algorithm, which creates strategies using plans. Plans are certain general descriptions of strategies. For some positions in chess plans might be occupying the center or the corners of the board. Each plan represents a hierarchy of goals. Those are the goals which a player tries to achieve in current situation while playing by the plan. The essences of the plans are to select the goals which get the maximal profit.

The PPIT program was designed as a composition of the following basic units:
Reducing Hopeless Plans (RHP)
Choosing Plans with Max Utility (CPMU)
Generating Moves by a Plan (GMP)

Given a questioned position P1 and a store of plans, RHP recommends to CPMU a list L1 of plans promising by some not necessary proved reasons to be analyzed in P1. The core of the unit is knowledge in classification of chess positions allowing identifying the niche in the store of knowledge the most relevant for analysis the position. If the store of knowledge is rich and P1 is identified properly it can provide a ready-to-use portion of knowledge to direct further game playing process by GMP unit. Otherwise, RHP, realizing a reduced version of CPMU, identifies L1 and passes the control to CPMU.

CPMU recommends to GMP to continue to play by current plan if L1 coincides with list L0 of plans formed in the previous position P0 and changes in P1 are not essential enough to influence the utility of current plan.

If changes in P1 are essential, CPMU analyzes L1 completely to find a plan with max utility and to address it to GMP as a new current plan. Otherwise, CPMU forms a new complementary list L1/ L1*L0 from the plans of L1 have not been analyzed, yet, in L0, finds a plan with the best utility in that list and comparing it with the utility of the current plan recommends one of them with a higher utility.

To calculate utilities of the attribute, goal and plan type units of chess knowledge, we represent them as operators over the corresponding arguments as follows:

- for basic attributes the arguments are characteristics of the states of squares in the questioned positions, including data on captures of pieces, threats, occupations, etc.;
- for composed attributes, including concepts and goals, the arguments are subsets of values of basic attributes relevant to the analyzed positions;
- for plans the arguments are utilities of the goals associated with the realization of those plans.

Utilities of arguments of basic attributes are calculated by the trajectory-zones based technique (TZT) [4, 6] originally suggested to estimate utilities of captures only of the opponent pieces. For example, to choose capture with max utility TZT chains the moves to each piece of the opponent (trajectories) without accounting possible handicaps for real capturing then using all available knowledge “plays the zones” of the game tree induced by the trajectories followed by estimation of their values to choose the best.

The utility of units of knowledge the operators assemble from the utilities of the corresponding arguments in some predetermined ordering. Thus, each operator can provide by a request the arguments which are analyzed at the moment.

For example, realizing the current plan the shell can determine the goal in the agenda which in turn determines basic attributes to be considered followed by indication of the arguments of those attributes.

Utility estimation operators rely on the principle of integration of all diversity of units of knowledge the shell possesses at the moment. In fact, the operators represent a kind of expert knowledge with a variety of mechanisms and leverages to make them better. Along with dynamically changed parametric values of pieces they can include rules, positions with known values and strategies to realize them, other combinatorial structures. To estimate expected utilities the operators take into account the cost of resources necessary to get them.

2.1.2. What has been done

In the initial C++ realization the units of knowledge are realized as OO classes with specialized interfaces for each type of knowledge and one common for the shell itself.

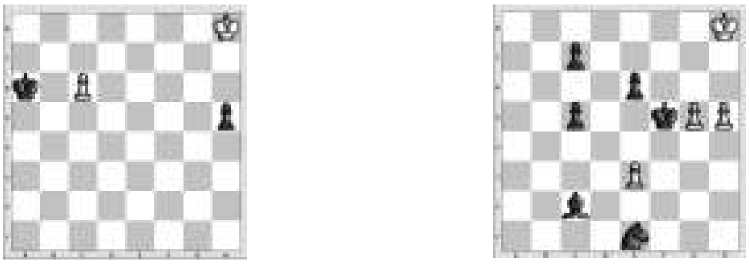


Fig 1. Reti and Nodareishvili etudes.

The Solver is experimented in solving Reti and Nodareishvili etudes (Fig 1.) required by Botvinnik[4, 5] intensive expert knowledge-based analysis not available to conventional chess programs.

Experiments with these etudes proved that the shells, in principle, can acquire the contents of units of vocabulary used by chess players and allow tuning them properly to solve expert knowledge intensive chess problems.

The initial implementation of the PPIT algorithm used knowledge units that were hardcoded as C++ language classes. The approach didn’t allow adding expert knowledge in a regular way – there wasn’t any regular method for formalization and representation of the expert knowledge.

To achieve a regularity of expert knowledge acquisition for RGT problems a graphical language similar to the UML, using which experts have possibility to formalize and insert meanings of the communicable knowledge into the Solver.

The constituents of the Interface have been designed for specifying both game attributes and rules. It was designed to acquire an expert knowledge in a form of patterns (abstracts). Abstracts are used to define classes as well as operations, thereby providing a considerable uniformity of the structure of the language [7, 10].

2.1.3. What We are Going to Do

We are developing algorithms and structures of strategy construction in the Solver package by putting the stress on GMP module of PPIT algorithms first. So for the current state of development we suppose that we already have plans defined in the Solver and we just need to execute the defined plan. Plans are being defined by experts.

In PPIT Plan is defined as a set of Goals. We will describe their definitions below.

As mentioned above the third module of PPIT GMP chooses the best move from a plan. We meet the following issues

1. Goals' and Plans' structures need to be generic and need to allow definition of the goals independent of the problems they relate to.
2. An algorithm needs to be developed to search for strategies using defined plans and goals. The algorithm needs to be generic and allow constructing strategies for any of defined plans.

2.2. Structuring Goals and Plans

As we used to do before in our research, now we're going to apply all the defined and developed to the chess as a classical example of RGT game.

The goal in general needs to have the following structure.

A. It needs a preCondition situation, for which this goal is applicable, because there are situations where a goal is not achievable, e.g., if the situation contains only two kings and a pawn, a goal like "make check with the queen" can't be applied. This basically defines the pattern of situations where goal is meaningful. Note that for some goals the preCondition can be any situation, so this is not obligatory to define some pattern in preCondition.

B. It needs to have a postCondition situation. This is the situation which appears when the goal is achieved, e.g., if the goal is "make check with the queen", after it is achieved the opponent king is under check of queen in the given situation, this describes the postCondition situation. This defines the pattern of achieved by the goal situations. Similar to preCondition, postCondition also can be any situation.

C. For some goals the depth of game tree needs to be more than one move, e.g., if the goal is "make perpetual check", we need to construct a tree and make several moves to see if this goal can be achieved.

D. Goals need to have some evaluation. There are goals like "put mate" or "avoid stalemate" where there are only two evaluation states, which indicate whether the goal is achieved or not, but there are some goals which do not show "an achieved" or "not" result, they show how good the goal is achieved, e.g., a goal "keep king closer to the opponent king" goal does need some criterion to define that the lesser distance between kings is, the better is the goal evaluation. For that purpose we define evaluator, which is a set of prioritized criteria that are being defined to evaluate the goal. For the above example only one criterion exists and it is the distance between two kings.

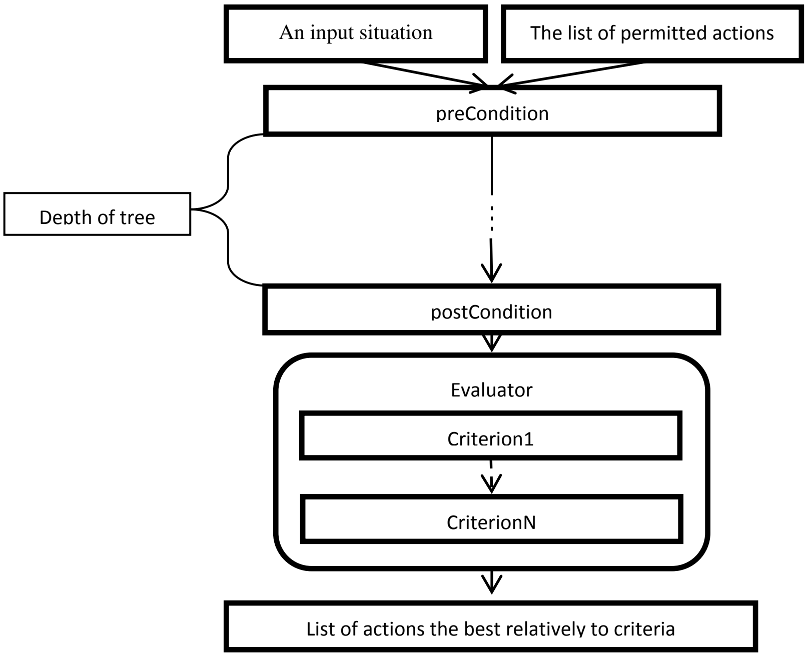


Fig 2. The structure of goals, their inputs and outputs.

From the described above we reveal that the goal consists of preCondition and postCondition, which are situations (in the Solver we define these situations as composite abstracts), depth of three, which is a number that defines how deep the tree can be constructed for checking if the postCondition is achieved (by default it is 1) and the evaluator which evaluates how good the goal is achieved.

Also one important point we need to define the concept of absolute goal (which is just a flag on the goal), like mate in chess and indicates that the game is over.

The plan structure is basically defined in previous works of our team and nothing more is required. It consists of prioritized goals.

2.3. Searching Strategies by Plans

Now when we have the structures of goals and plans, we can define how the algorithm should work to find the best move from the given plan.

As described above the goal and the plan are completely generic in their structures regardless of the problem they solve in RGT class and can be defined by a user, not only injected initially for a certain problem.

The algorithm we have developed to execute the plans and to choose the best action by the defined plan is the following.

As said above plan is a set of prioritized goals, we need to run over the goals and find the move which best satisfies the highest priority goal.

The algorithm initially requires input situation (IS). For IS Solver does matching and finds the list of active abstracts [8], where there are also actions active in that situation (the actions that are possible to perform in IS), let's call the list of active in IS actions <A>. Let's assume we have Plan Pl which has G1 to Gn goals in it (G1 has the highest priority and Gn has the lowest priority). For the given P1 plan, the algorithm will take goals from the highest to the lowest priority and do the following procedure.

1. Passing list of actions <A_{i-1}> (for G1 <A> is passed instead of <A_{i-1}>). If the list is empty then nothing can be done for this goal, just returning, else if the list has only one action, then the list is returned and the procedure is stopped, as nothing to do if only one action can be done, no need of further processing, we just do the action.
2. preCondition of the goal is checked against IS. If IS satisfies the pattern defined in the preCondition all actions in the action set <A_{i-1}> are applied to the IS situation and postCondition of current G_i is checked to be achieved. The goal is being evaluated by the criterion defined in the evaluator if there are any and the actions which satisfy the goal best are being returned in the list <A_i> (this list will be used in the next goal processing). An important point here is that if the goal is absolute and the list of actions achieving this goal is not empty, then the procedure is stopped after this step and the list of actions is returned.
3. If the returned list <A_i> is empty, <A_{i-1}> list is being used instead, otherwise if the returned list has only one item in it, the list is just returned and the procedure is stopped.
4. New Actions list is passed to the next goal and the procedure is being done for it from the beginning (1 to 4).
5. When the procedure is done for all goals or stopped somewhere while performing 1-4 steps, it returns the list of actions, which indicate the best actions to achieve the plan in the current situation. Any of those actions brings to the best move selection and thus brings to the best strategy for the given P1 plan.

Any action from the returned list of actions is being selected (we just select the first one) and applied to the IS. New situation is achieved after opponent's action, so we already have a changed situation, a new input situation. The plan execution starts again for the new situation and a new best move is selected for the plan. The algorithm is stopped when the highest priority goal is achieved or is not achievable at all (e.g., we have already put mate or no mate can be achieved), which means that either the strategy for the given plan already worked or cannot be achieved anymore.

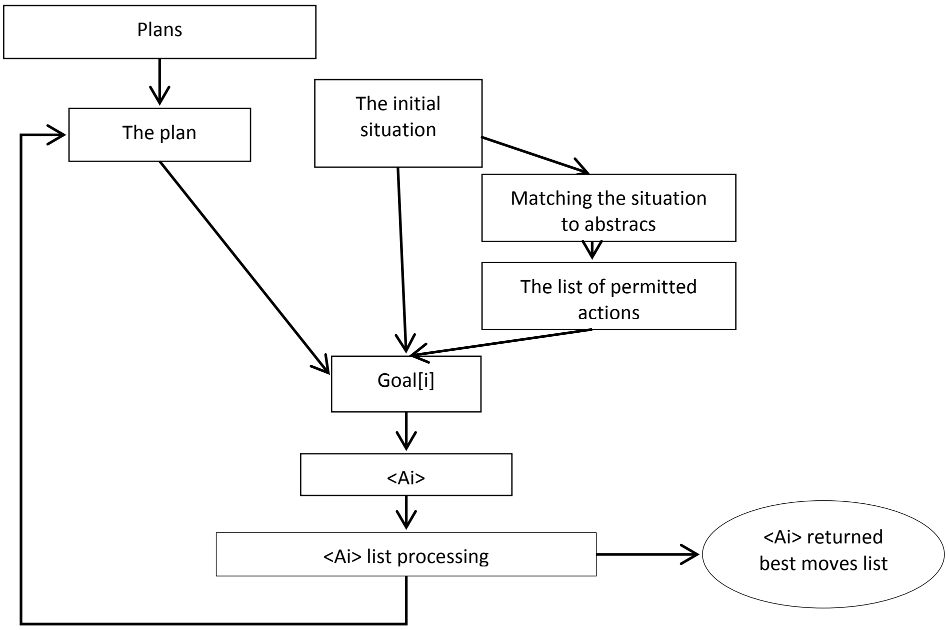

Plans
The plan
The initial situation
Matching the situation to abstracs
The list of permitted actions
Goal[i]
<Ai>
<Ai> list processing
<Ai> returned best moves list

Fig 3. The schema of searching the best moves.

3. Testing Adequacy of PPIT for Chess Endgames

3.1. Planning Chess Endgames

3.1.1. Planning "rook against king" endgame

Previously the strategy description language was defined in [9], where exact algorithms were used to define each plan and its realization. For the demonstration of the language adequacy "rook against king" chess endgame was described. For the demonstration of our algorithms we will also consider chess endgames, like "rook against king" or "two rooks against king".

To simplify the definitions we just assume our color is predefined and is white. We will try to define only the mate on one direction to make it simpler, the same is done in [9]. Let's take vertical direction only for our future definitions. Similarly we will be able to define putting mate on horizontal direction. Which one to choose vertical or horizontal is a job for another module in PPIT algorithms expected to be developed in the scope of Solver during the future steps of our research. Currently it will just construct strategy with the given certain plan.

A plan for the "rook against king" endgame will have the below goals

1. Put mate
2. Avoid stalemate (note that this is quite important because some situations can appear with stalemate and we need to avoid it)
3. Escape rook from attack
4. Push king to the edge (without putting rook under attack)
5. Make a waiting move when preOpposition appears
6. Bring white king closer to the black king

The definition of each goal is described in details.

1. Putting mate - preCondition is any situation, and postCondition is a situation where there's mate, the depth is 1, this is absolute goal. There is no evaluator defined for this goal.
2. Avoid stalemate - preCondition is again any situation and the postCondition is a situation where no stalemate appears. The depth is 1 and no evaluator again.
3. Escape rook from attack - the preCondition is "rook under kings attack" abstract, so the goal is applicable only for situations where the rook is under the opponent king's attack. The postCondition is a situation where rook is not under attack and the vertical coordinate of the rook is not changed. It has a depth value 1 and the evaluator will have one criterion defined which calculates the distance of the rook and opponent king by vertical direction.
4. Push king to the edge- preCondition can be any situation and postCondition is "rook is not under attack" situation and depth is 2. The evaluator has two criteria. First is: moves of opponent king are closer to the edge are better (this basically means the horizontal distance of opponent king from the edge is calculated and for each action the value of criterion is calculated as the highest value of king's distance from the edge). The second criterion for this goal evaluator is the number of actions opponent king can do, and the better action is the action which allows fewer number of actions by opponent king.
5. Make a waiting move when preOpposition appears - preCondition is preOpposition situation. PreOppositionByVertical abstract in the Solver can be defined as below.
   This is a virtual abstract which has two attributes – black and white kings. It must have 4 specifications

A. Whiteking.cordX = BlackKing.cordX + 2
whiteking.cordY = blackking.cordY + 1
B. Whiteking.cordX = BlackKing.cordX + 2
whiteking.cordY = blackking.cordY – 1
C. Whiteking.cordX = BlackKing.cordX - 2
whiteking.cordY = blackking.cordY + 1
D. Whiteking.cordX = BlackKing.cordX - 2
whiteking.cordY = blackking.cordY – 1

which is complete enough to define the precondition of preOpposition.
The postCondition is a situation where the king position is not changed and the rook vertical coordinate is not changed. Depth of goal is 1. The evaluator again has one criterion, which shows the distance of the rook from the opponent king.

6. Bring white king closer to the black king, but avoid opposition – preCondition and is any situation and postCondition is a situation where no opposition appears, depth is 1. The evaluator has one criterion, which defines the distance of the king from the opponent king to be minimal. We can calculate this by the following formula "$(\text{king.cordX-opponentKing.cordX})^2$ + $(\text{king.cordY-opponentKing.cordY})^2$".

3.1.2. Planning "two rooks against king" Endgame

A winning plan for chess endgame "two rooks against king" will be

1. Put mate
2. Avoid stalemate
3. Escape rook from attack
4. Push king to the edge, where postCondition will be two rooks on the board and the criterion of evaluator will be only opponent king's distance from edge is minimal.
5. Escape rook which vertical coordinate is different from opponent king's coordinate by 1 (rook.y = king.y + 1 or rook.y = king.y - 1).

3.2. Searching for Winning Strategy of "rook against king"

Chapter 3.1 describes how chess endgames can be brought into Solver and this chapter describes the execution of the plans by the designed algorithm for "rook against king" example. For other plans its work is similar. Let's see how the algorithm works for a situation.

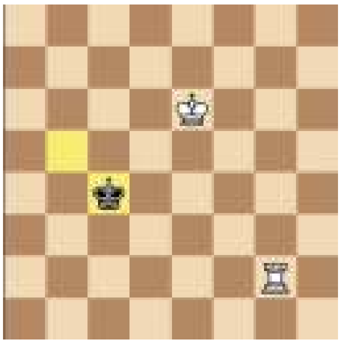


Fig 4. K., R. vs. B.K., An initial position.

1. Algorithm tries to find moves which bring to mate, and returns the empty list.
2. Since the returned list of the 1st goal is empty it takes the initial list of moves and returns the whole list of possible moves since all of them brings to situations where there is no stalemate, so the whole list of moves is passed to the 3rd step
3. "Escape rook from attack" goal is not applicable for this situation, so it just does nothing
4. "Push king to the edge" for all the moves that does not put rook under attack it calculates the first criterion value. Let's see what values it assigns to three of moves.
    a. 1. Rc2… this puts check to the black king, for all king moves it calculates the distance from the vertical edge. King moves can be Kd4, Kd3, Kb4… for Kd4 and Kd3 it assigns will assign the highest value of 4 (the distance from edge is 4). Kb4 will have value 2, so the value assigned to move Rc2 is 4.
    b. 1. Rd2… king can do moves Kc3, Kc5, Kb4… for Kb4 again value as mentioned above is 2, for Kc3 and Kc5 is 3, so the value for Rd2 move is 3.
    c. 1. Rg3… in this case also black king can move to d4 position, so the value will be 4.

    Similarly all moves other than Rd2 will have 4 value, the minimum value is 3, and only Rd2 has that, so after processing the 4th goal the algorithm will return move Rd2

Since only Rd2 move is returned the algorithm is not processed anymore and this move is applied.

Let's assume black does Kc3 move (attacking rook).

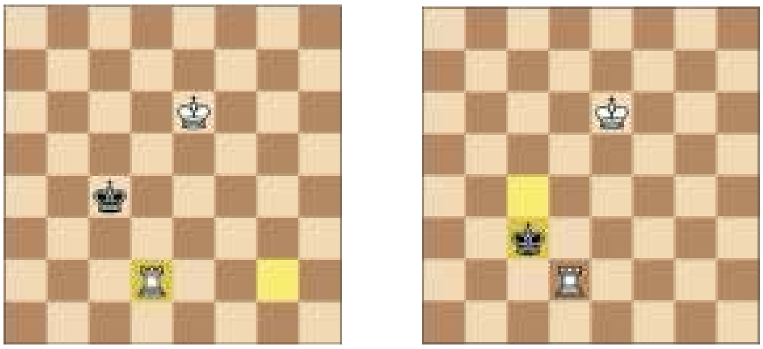


Fig 5. The left: the position after Rd2. The right: the position after Kc3.

After Kc3 move algorithm works again

1. For mate goal again empty list is returned
2. For stalemate all moves list is returned
3. "Escape rook from attack" goal's preCondition is matched to the situation and rook moves are considered to achieve the goal where rook is not under black king's attack since postCondition is "rook not under attack". The criterion to evaluate the move is vertical distance of rook and black king, so Rd8 move is chosen since it has the highest vertical distance from black king. Since the list has only one move in it, the procedure is stopped here and Rd8 move is returned

Rd8 is applied to the situation. Let's assume black makes Kc4 move.

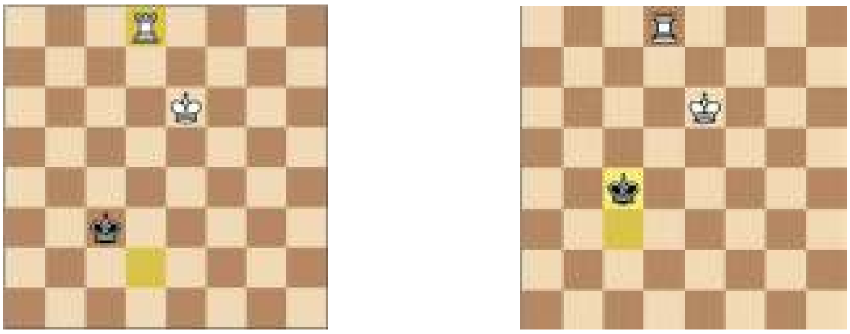


Fig 6. The left: the position after Rd8. The right: the position after Kc4.

Algorithm works again and now with the following result.

1. For goal mate again empty list is returned
2. For stalemate all moves list is returned
3. No rook under attack so this is just omitted
4. "Push king to the edge" for all the moves where rook is not under attack it checks the evaluator, which have two criteria, the 1st is kings distance from the edge is minimum. So for moves Rd1, Rd2, Rd6, Rd7,Ke7, Ke5, Kf7, Kf6, Kf5 the distance of king from the edge will be calculated as it was done for the 1st move, and the value will be 3, which is selected as the minimum value. Then the second criterion (which is the number of moves opponent king can make) is checked for the moves which are best for criterion 1. Number of moves of black king is always 5 for all the mentioned moves. So the whole list is returned from this goal processing procedure.
5. The situation is not a preOpposition, so preCondition is not matched, this goal is just omitted.
6. "Bringing king closer" preCondition is any situations, and postCondition is a situation where no opposition appears. The list of moves is [Rd1, Rd2, Rd6, Rd7, Ke7, Ke5, Kf7, Kf6, Kf5], which does not bring to opposition, so all of them satisfy postCondition. The evaluator criterion is that distance between two kings needs to be minimum. For the moves by rook distance value will be 8 ($(5 -3)^2 + (6-4)^2$). For king moving by f vertical the value will be rising, e.g., after Kf6 criterion returns 13 ($(6 -3)^2 + (6-4)^2$). The best move will be Ke5, which will have evaluation value 5 ($(5 -3)^2 + (5-4)^2$). Ke5 will be returned.

Since only Ke5 is returned this is applied to the situation. To make the example shorter let's consider Kd5 move for black.

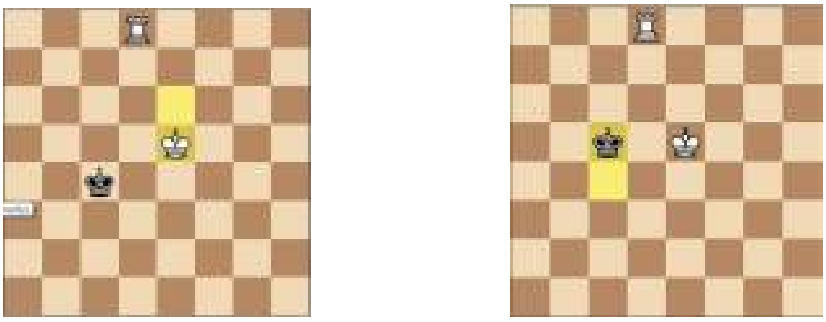


Fig 7. The left: the position after Ke5. The right: the position after Kc5.

After Kc5 move similar to the 1st move for "push king to the edge goal" Rc8 move will be selected. Again we will assign black king moves which finish the game sooner, we will consider the move Kb4. So after the following moves

1. Rd2 Kc3 2. Rd8 Kc4 3. Ke5 Kc5 4. Rc8 Kb4 5. Kd5 Kb5 6. Rb8 Ka4 7. Kc5 Ka3 8. Kc4 Ka2 9. Kc3 Ka1 10. Kc2 Ka2.

After the 10th move (Ka2 by black) the algorithm will work and find that mate is achievable and Ra8 move will be returned. This move will be applied and the plan is achieved.

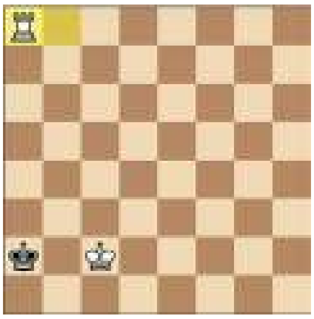


Fig 8. The position of putting mate.

4. Conclusion

1. Structures of plans and goals are defined for the Solver of RGT class allowing user to describe generic plans and goals for any problem of this class in a regular manner. Goals are defined as a composition of preCondition, postCondition situations, depth of game tree to achieve the goal and evaluator to evaluate the utility achieved in a situation while accomplishing the goal. Plans are sets of prioritized goals.
2. An algorithm of searching strategy by a plan was constructed and developed based on PPIT algorithms previously developed by our team for certain problems and with injected knowledge usage. The algorithm works only with defined plans and goals, regardless of the problem it solves. Previously the constructed PPIT consists of three modules RHP, CPMU, GMP.
    a. In the following we developed algorithms for GMP module
    b. Future development of other modules within the scope of Solver to complete PPIT algorithm are in progress now, which is related to constructing algorithms to choosing the best plan from the given list of plans. This corresponds to CPMU module.

   For the current state we assume that expert knowledge for plans is being defined by a user but in the development process we aim to achieve creating algorithms for Solver to generate plans by itself relying on the knowledge set it already has for the game.
3. Demonstration of the structures and the algorithms were carried out for chess endgames, their adequacy is shown. More experiments are in progress now for different chess situations, particularly Reti etude planning is in progress now.

Acknowledgement

Author expresses his deep gratitude to Professor Edward Pogossian for supervising the work and to Boris Karapetyan for support in defining the winning plan of chess endgame “Rook against king”.

References

[1] Ch. Brutyan, I. Zaslavski and L. Mkrtchyan, “On methods of automated synthesis of positional strategies in games”, *Problemi Kibernetiki,* Moscow, Russia, vol. 19, pp. 141-175, 1967.

[2] E. Pogossian, V. Vahradyan and A. Grigoryan, “On competing agents consistent with expert knowledge”, *Lecture Notes in Computer Science, AIS-ADM-07: The International Workshop on Autonomous Intelligent Systems - Agents and Data Mining*, pp. 229-241, St. Petersburg, Russia, June 6-7, 2007.

[3] E. Pogossian, A. Javadyan and E. Ivanyan, “Effective discovery of intrusion protection strategies”, *The International Workshop on Agents and Data Mining, Lecture Notes in Computer Science,* St. Petersburg, Russia, vol. 3505, pp. 263-274, 2005.

[4] M. M. Botvinnik, *About Solving Approximate Problems*, (in Russian), S. Radio, Moscow, 1979.

[5] M.M. Botvinnik, “Computers in chess: solving inexact search problems”, *Springer Series in Symbolic Computation, with Appendixes, Springer-Verlag*, New York, 1984.

[6] E. Pogossian, V. Vahradyan and A. Grigoryan, “Experiments in consistency of chess expertise with decision making for etudes of Retie and Nodareishvili”, *Transactions of IIAP NAS RA, Mathematical Problems of Computer Science*, (in Russian), vol. 28, pp. 94—113, 2007.

[7] K. Khachatryan and S. Grigoryan, “Java programs for presentation and acquisition of meanings in SSRGT games”, *Proceedings of SEUA Annual conference*, pp. 127-135, Yerevan, Armenia, 2013.

[8] K. Khachatryan and S. Grigoryan, “Java programs for matching situations to the meanings of SSRGT games”, *Proceedings of SEUA Annual conference*, pp. 135-141 Yerevan, Armenia, 2013.

[9] B. Karapetyan, “High level strategy description language in games”, *Mathematical Problems of Computer Science*, (in Russian), vol. 16, pp. 167-183, 1986.

[10] E. Pogossian, “On modeling cognition”, *International Conference in Computer Sciences and Information Technologies*, Yerevan, Armenia, Sept. 26-30, pp. 194-198, 2011.

Submitted 10.12.2014, accepted 16.02.2015.

Նպատակների և պլանների կառուցում անձնավորված պլանավորման և պլանների ինտեգրացված թեստավորման համար

Ս. Գրիգորյան

Ամփոփում

Մենք ուսումնասիրում ենք մրցակցային խնդիրները՝ սահմանված որպես դաս, որտեղ լուծումների բազմությունը վերարտադրելի ծառ է (RGT): Մշակված են Անձնավորված պլանավորման և ինտեգրացված թեստավորման ալգորիթմներ RGT խնդիրներում լավագույն ռազմավարության փնտրման համար: Աշխատանքում զարգացվում են նպատակների և պլանների կառուցվածքներ, կառուցվում է ռազմավարության փնտրման ալգորիթմ ըստ պլանի և ցուցադրվում է նրանց հիմնավորությունը:

Структурирование целей и планов для персонализированного планирования и интегрированного тестирования

С. Григорян

Аннотация

Разработаны алгоритмы и программы представления планов и целей при решении задач класса RGT. Представлено описание поиска стратегий на основе планов для пакета Solver. Обоснованность алгоритмов показана на примере шахматных эндшпилей.