ISSN 1995-0802, Lobachevskii Journal of Mathematics, 2015, Vol. 36, No. 4, pp. 389–398. © Pleiades Publishing, Ltd., 2015.

Downward Density of Exact Degrees

J. Liu[1*], G. Wu[2], and M. Yamaleev[3***]**

(Submitted by S Barry Cooper)

[1] *CIGIT, Chinese Academy of Sciences, Chongqing 401122, China*
[2] *SPMS, Nanyang Technological University, 21 Nanyang Link, 637371 Singapore*
[3] *IMM, Kazan Federal University, Kremlyeskaya ul. 18, Kazan, 420008 Russia*
Received January 12, 2015

Abstract—In this paper we study exact d.c.e. degrees, the class of d.c.e. degrees which is strictly between the class of degrees of tops of bubbles and the class of isolated d.c.e. degrees. We show that exact degrees are downward dense similar to isolated d.c.e. degrees.

DOI: 10.1134/S1995080215040095

Keywords and phrases: *Erhsov's hierarchy, d.c.e. sets, Lachlans' sets, exact degrees.*

1. INTRODUCTION

The differences of computably enumerable (further, d.c.e.) sets gives a natural generalization of computably enumerable (further, c.e.) sets, which are one of the central objects in Computability Theory. The structures of Turing degrees of c.e. and d.c.e. sets are studied during last several decades and nowadays we know a lot of particular properties of these structures, but there are still several basic open problems related to their model-theoretical properties.

Arslanov, Kalimullin and Lempp [1] showed that there exists a bubble pair in the class of d.c.e. Turing degrees. Using this notion and its generalization they refuted the Downey's Conjecture showing that partial orders of d.c.e. and 3-c.e. Turing degrees are not elementarily equivalent. Recall that d.c.e. degrees $\mathbf{0} < \mathbf{b} < \mathbf{a}$ form a bubble pair (or also bubble) if all d.c.e. degrees below $\mathbf{a}$ are comparable with $\mathbf{b}$, also we say that $\mathbf{b}$ is the middle of bubble and $\mathbf{a}$ is the top of bubble. It's easy to see that $\mathbf{a}$ is always d.c.e. and it was proved in [1] that $\mathbf{b}$ must be c.e. This bubble pair phenomena is an essence of properly d.c.e. degrees and has many nontrivial properties. For example, it's easy to see that $\mathbf{a}$ is an exact d.c.e. degree (see [5]), in particular it's an isolated degree (see, e.g. [2]), also it's easy to see that $\mathbf{a}$ is not splittable avoiding the upper cone of $\mathbf{b}$ (see [3]).

The construction of a bubble is a highly nontrivial task and it's interesting whether it can be combined with other properties, like upward or downward density in the c.e. degrees. The positive answer for density question in the c.e. degrees allows to distinguish c.e. degrees in the class of d.c.e. degrees. Indeed, if between any two c.e. degrees $\mathbf{e} < \mathbf{c}$ there is a bubble then the middle of the bubble is also between them. Hence, for any c.e. degree $\mathbf{c}$ it has a nontrivial c.e. splitting $\mathbf{c_0} \cup \mathbf{c_1} = \mathbf{c}$ and there are middles of bubbles in the intervals $[\mathbf{c_0}, \mathbf{c}]$ and $[\mathbf{c_1}, \mathbf{c}]$. On the other hand, if $\mathbf{d}$ is a properly d.c.e. degree and if $\mathbf{d_0} \cup \mathbf{d_1} = \mathbf{d}$ is a nontrivial splitting in the d.c.e degrees then either $[\mathbf{d_0}, \mathbf{d}]$ or $[\mathbf{d_1}, \mathbf{d}]$ doesn't contain a c.e. degree. Therefore, it doesn't contain middle of bubble, contrary to the c.e. case above.

However, the interaction of the original bubble construction and permitting argument provides many difficulties. Hence, even downward density for bubbles is under question. Our approach is to consider different properties of bubbles separately. Namely, the exact degrees, introduced by Ishmukhametov [5], form the class of d.c.e. degrees which extend the class of tops of bubbles. So, the question on density,

*E-mail: liujiang@cigit.ac.cn
**E-mail: guohua@ntu.edu.sg
***E-mail: mars.yamaleev@kpfu.ru

particularly downward and upward density, can be naturally asked for this class. Also the density questions are well studied for the class of isolated d.c.e. degrees which extend the class of exact degrees. However, the nature of constructions of isolated degrees and exact degrees doesn't allow to transfer the results from isolated degrees into exacts degrees. Using some modifications we show in the main theorem that exact d.c.e. degrees are downward dense. We left open the following questions.

Question 1.1. *Are bubble pairs downward dense?*

Question 1.2. *Are exact degrees upward dense?*

For a d.c.e. set D and its enumeration $\{D_s\}_{s\in\omega}$ such that $|D_s - D_{s-1}| \leq 1$, we define partial computable (further, p.c.) function $s^D(x)$ such that $s^D(x)\downarrow = s$ if and only if x is enumerated into D at stage s. If later x is extracted from D then $s^D(x)$ is enumerated into its *Lachlan's set* $L(D) = \{s \mid \exists x\ s^D(x)\downarrow = s\ \ \&\ x \notin D\}$. Let $\mathbf{d} = \deg(D)$, then $L[\mathbf{d}] = \{\deg(L(B)) \mid B \in \mathbf{d}$ and B is d.c.e.$\}$, elements of $L[\mathbf{d}]$ we call *Lachlan's degrees* of $\mathbf{d}$. In [5], Ishmukhametov defined $R[\mathbf{d}]$ as the set of all degrees below $\mathbf{d}$, in which $\mathbf{d}$ is relatively computable enumerable. It's easy to show that $R[\mathbf{d}] = L[\mathbf{d}]$ (see, e.g. [4]). Following [5], a noncomputable d.c.e degree $\mathbf{d}$ is *exact* if $|L[\mathbf{d}]| = 1$. Now we proceed to the proof of the main result. Our notation and terminology generally follow Soare [6], for some special definitions we refer to [4, 5].

2. THEOREM AND STRATEGIES

Theorem 1. *Below any noncomputable c.e. degree* $\mathbf{a}$ *there exists an exact d.c.e. degree* $\mathbf{d}$.

2.1. Requirements

Given a c.e. set $A \in \mathbf{a}$ with a computable enumeration $\{A_s\}_{s\in\omega}$, we construct a noncomputable d.c.e. set D such that $D \leq_T A$ and $L(B) \equiv_T L(D)$ for any d.c.e. set $B \equiv_T D$. So it suffices to build a d.c.e. set D and functionals Δ_e and Γ_e satisfying the following requirements:

$\mathcal{G}$: $D \leq_T A$,

$\mathcal{R}_e$: $D = \Phi_e^{B_e}\ \&\ B_e = \Psi_e^D \Rightarrow \exists\, \Delta_e(L(D) = \Delta_e^{L(B_e)})$,

$\mathcal{S}_e$: $W_e = \Theta_e^D \Rightarrow \exists\, \Gamma_e(W_e = \Gamma_e^{L(D)})$,

$\mathcal{P}_e$: $L(D) \neq \Omega_e$,

where $\{\langle W_e, B_e, \Phi_e, \Psi_e, \Omega_e\rangle\}_{e\in\omega}$ is an effective enumeration of all corresponding 5-tuples of c.e. and d.c.e. sets and p.c. functionals. Note that the $\mathcal{S}$-requirements indicate that $L(D)$ has the greatest c.e. degree below $\mathbf{d}$, so it isolates D. On the other hand, the $\mathcal{R}$-requirements indicate that $L(D)$ has the least c.e. degree among all Lachlan degrees of d.c.e. sets from the degree $\deg(D)$. At the same time, $\mathcal{P}$-requirements ensure that $\deg(L(D))$ is not computable, hence $\deg(D)$ is an exact d.c.e. degree.

2.2. Strategies

Basic $\mathcal{P}$-strategy. The basic idea of a $\mathcal{P}$-strategy, say π, is the following. First of all, π respects the global requirement $D \leq_T A$ and uses direct permitting argument. That is, whenever π puts or extracts a number from D there is the corresponding A-change. So, π picks step by step an increasing sequence of numbers $x_0 < x_1 < \dots < x_n < \dots$ and waits for an A-permission in order to put one of them into D. If it never gets an A-permission, then A is computable, which contradicts with the assumption of the theorem. Suppose π enumerates some x into D at stage s. To make $L(D) \neq \Omega_e$, π chooses such s and waits for $\Omega_e(s)\downarrow = 0$. If $\Omega_e(s)\downarrow = 0$, then π may extract x from D (and force s entering $L(D)$) making $L(D)(s) = 1 \neq 0 = \Omega_e(s)$. However, π's extraction of x also need an A-permission. So, π starts waiting for permission for extraction of x and simultaneously organize another infinite sequence $x'_0 < x'_1 < \dots < x'_n < \dots$ in order to get a new witness $s' > s$ as an entry stage of some $x' > x$ from this sequence. By this argument we can get an infinite sequence of witnesses and the corresponding numbers prepared for extraction and waiting for A-permission. Again, one of these numbers will get an A-permission since A is not computable. Note that an enumeration process doesn't depend on π's Ω-functional, but an extraction process does.

Basic $\mathcal{S}$-strategy. An $\mathcal{S}$-strategy, say σ, tests the length agreement of its $W_e = \Theta_e^D$, and if the length agreement is larger than the ones before then σ defines $\Gamma_e^{L(D)}$ as current W_e up to the length agreement. The problem is that for some argument m, $\Gamma_e^{L(D)}(m)$ is first defined as $W_e(m) = 0$ at stage s, later a small number is enumerated into or extracted from D making $\Phi_e^D(m)[t] = 1$ at a bigger stage $t > s$, which allows $W_e(m)$ to change.

To solve this problem, we use the following strategy. If the $W_e(m)$-change is derived from an enumeration into D, then σ will extract this number from D to make a disagreement $\Theta_e^D(m) = 0 \neq 1 = W_e(m)$. Otherwise, $W_e(m)$-change is caused by an extraction from D and the corresponding number, say s, is enumerated into $L(D)$. In this case the $L(D)$-change allows to correct $\Gamma_e^{L(D)}(m)$.

Particularly, such $W_e(m)$-change comes from a $\mathcal{P}$-strategy enumerating a number into D. Formally, we illustrate the process by considering an $\mathcal{S}$-strategy σ with a lower priority $\mathcal{P}$-strategy $\pi \supset \sigma^\frown\infty$.

First, we define the length agreement functions:

- $l(\sigma, s) = \max\{y < s : \forall x < y\ (W_e(x)[s] = \Theta_e^D(x) \downarrow [s])\}$,
- $m(\sigma, s) = \max\{0, l(\sigma, t) : t < s \text{ and } t \text{ is } \sigma\text{-stage}\}$.

We say that s is a σ-expansionary stage if $s = 0$ or $l(\sigma, s) > m(\sigma, s)$. Proceed to an example of simultaneous work of σ and π.

(σ1) The $\mathcal{P}$-strategy π chooses a fresh number x at stage s_0.

(σ2) The $\mathcal{P}$-strategy π waits for a π-stage $s_1 > s_0$, which is also a σ-expansionary stage, such that $l(\sigma, s_1) > x$, then puts x into D. From now on, restrain $D \upharpoonright s_1$ except for removing x.

(σ3) Wait for a σ-expansionary stage $s_2 > s_1$ such that $l(\sigma, s_2) > x$. First, σ checks whether $W_{e,s_1} \upharpoonright l(\sigma, s_1)$ changed.

(σ3.1) If there is a $W_{e,s_1} \upharpoonright l(\sigma, s_1)$-change at a number $m < l(\sigma, s_1)$, then it must be $W_{e,s_2}(m) = 1$. Now σ extracts x to restore the computation $\Theta_e^D(m)$ to $\Theta_e^D(m)[s_1] = 0$ and we restrain $D \upharpoonright s_2$, this allows the desired disagreement $\Theta_e^D(m) = 0 \neq 1 = W_e(m)$.

(σ3.2) Otherwise, no $W_{e,s_1} \upharpoonright l(\sigma, s_1)$-change appears between stages s_1 and s_2. Then σ takes no action on x, and π waits for $\Omega_\pi(s_1) \downarrow = 0$.

(σ3.2A) If $\Omega_\pi(s_1) \downarrow = 0$ never happens, then π never extracts x from D. So, σ is safe from π.

(σ3.2B) If $\Omega_\pi(s_1) \downarrow = 0$ happens and π does extract x from D at some stage $s_3 \geq s_2$, then the corresponding s_1 is enumerated into $L(D)$ immediately. Now, the extraction of x may allow a W_e-change, say at m. However, this m must be greater than s_2 and s_1. So $\Gamma_e^{L(D)}(m)$ is defined after stage s_2 (so it has a bigger use $> s_1$). Hence, the enumeration of s_1 into $L(D)$ allows to automatically correct $\Gamma_e^{L(D)}(m)$.

Remark. Note that a number (which is enumerated into D by π) extraction from D could be performed by some other $\mathcal{S}$- or $\mathcal{R}$-strategy between π and σ. This can happen only when case (σ3.1) is false at a bigger stage $s > s_2$. Then it is similar to case (σ3.2B) where any incorrect $\Gamma_e^{L(D)}(m)$ is due to an extraction of some x which was enumerated before the stage s_2.

Basic $\mathcal{R}$-strategy. The basic idea of an $\mathcal{R}$-strategy is similar to the one of an $\mathcal{S}$-strategy. Roughly, an $\mathcal{R}$-strategy ρ would define $\Delta_e^{L(B_e)}$ computing $L(D)$ or produce a disagreement between B_e and Ψ_e^D by extracting numbers from D. However, the situation here is more complicated since $L(B_e)$ is given and not controlled by us directly. Specifically, after $L(D)(s)$-change, ρ can either force a corresponding $L(B_e) \upharpoonright \delta_e(s)$-change or make a successful diagonalization between B_e and Ψ_e^D. To demonstrate the idea, we consider a process as follows.

First of all, we define the length functions at stage s by

- $l(\rho, s) = \max\{y < s : \forall x < y(\Phi_e^{B_e}(x)[s] \downarrow = D_s(x) \,\&\, \forall u < \varphi_e(x,s)(\Psi_e^D(u)[s] \downarrow = B_{e,s}(u)))\}$,
- $u(\rho, s) = \max\{\varphi_e(x,s) | x < l(\rho, s)\}$,
- $m(\rho, s) = \max\{0, l(\rho, t) : t < s \text{ and } t \text{ is an } \rho\text{-stage}\}$.

We say that s is an ρ-expansionary stage if $s = 0$ or $l(\rho, s) > m(\rho, s)$. By s^- we denote the previous expansionary stage before s (we will use it in the construction). Clearly, at each ρ-expansionary stage, $z < u(\rho, s)$ implies $\Psi_e^D(z)[s] \downarrow = B_{e,s}(z)$. Note that only $\mathcal{P}$-strategy can enumerate numbers into D. So we present an $\mathcal{R}$-strategy ρ with a lower priority $\mathcal{P}$-strategy $\pi \supset \rho^\frown\infty$.

(ρ1) The $\mathcal{P}$-strategy π chooses a fresh number x at stage s_0.

(ρ2) The $\mathcal{P}$-strategy π waits for a π-stage $s_1 > s_0$, which is also a ρ-expansionary stage, such that $l(\rho, s_1) > x$, then puts x into D. From now on, restrain $D \upharpoonright s_1$ except for removing x.

(ρ3) Wait for the ρ-expansionary stage $s_2 > s_1$. Then $\Phi_e^{B_e}(x)$ changed its value between stages s_1 and s_2 since $0 = \Phi_e^{B_e}(x)[s_1] \neq \Phi_e^{B_e}(x)[s_2] = D(x)[s_2] = 1$. Thus, there must be some $y < \varphi_e(x, s_1)$ which entered or exited B_e between these stages.

(ρ3.1) If $y \in B_{e,s_1}$ but $y \notin B_{e,s_2}$, then ρ extracts x from D to restore the computation $\Psi_e^D(y) = 1 = \Psi_e^D(y)[s_1]$, from now on, restrain $D \upharpoonright s_2$. Hence we have the disagreement $\Psi_e^D(y) = 1 \neq 0 = B_e(y)$ since B is a d.c.e. set.

(ρ3.2) Otherwise, $y \notin B_{e,s_1}$ but $y \in B_{e,s_2}$, then ρ takes no action on x. However, π may extract x from D afterward.

(ρ3.2A) If π never extracts x from D, then it does not put any number into $L(D)$. So ρ is safe from π.

(ρ3.2B) If π extracts x from D at some stage $s_3 \geq s_2$, then s_1 is enumerated into $L(D)$. Hence, ρ must be able to correct $\Delta_e^{L(B_e)}(s_1)$ at the next ρ-expansionary stage. In this case, y should leave B_e (otherwise, we get a diagonalization), hence some number $s_1 < s \leq s_2$ should enter $L(B_e)$ and we use it for the correction. Note that ρ delays the first definition of $\Delta_e^{L(B_e)}(s_1)$ and define it only at the next ρ-expansionary stage s_2.

Remark. Note that a number (that enumerated into D by π) can be extracted by some other $\mathcal{S}$ or $\mathcal{R}$-strategy between π and ρ. This can happen only if case (ρ3.1) is false at a bigger stage $s > s_2$. Then we correct the value at point s_1 similar to (ρ3.2B).

Interactions Among Strategies. In the previous $\mathcal{S}$- and $\mathcal{R}$-strategies, we omitted the $\mathcal{G}$-requirement when D changed. However, similar to $\mathcal{P}$-strategy, we need an A-permission whenever $\mathcal{S}$- and $\mathcal{R}$-strategies make any change on D. Basic idea is to code the latent disagreement actions at steps (σ3.1) and (ρ3.1) as cycles. So each cycle constructs the corresponding functional $\Gamma_{\sigma,k}^{L(D)}$ or $\Delta_{\rho,k}^{L(B_e)}$ associated with numbers x_k for $k = 0, 1, 2, \ldots$. At the same time, we compute $A(k)$ using computable functions $f_\sigma(k)$ and $f_\rho(k)$, respectively. As A is noncomputable, the strategies σ and ρ eventually make successful disagreements.

As we make use of direct permitting argument, it allows consecutive D-changes without any expansionary stages. By the proceeding discussion, if that is only consecutive extraction from D then the $\mathcal{S}$- and $\mathcal{R}$-strategies take effect. It is also easy to see that pure consecutive enumeration into D has no essential injury to the $\mathcal{S}$- and $\mathcal{R}$-strategies. In this case, they pick out the least enumeration that causes W_e-change or B_e-change to process their basic strategies.

If the consecutive D-changes contains both enumeration and extraction, then it can threaten to the basic $\mathcal{S}$- and $\mathcal{R}$-strategies that restrain D at non-expansionary stages. Without loss of generality, we consider the following case with two $\mathcal{P}$-strategies π_1 and π_2 below an $\mathcal{R}$-strategy ρ such that $\rho^\frown\infty \subset \pi_1 \subset \pi_2$.

(1) Suppose s_0 is a ρ-expansionary stage, x was chosen by π_1 and z by π_2 at some stages before $< s_0$.

(2) The $\mathcal{P}$-strategy π_2 gets A-permitting and extracts z at stage $s_1 > s_0$. Accordingly it enumerates the corresponding number $s^D(z) < s$ into $L(D)$.

(3) At a bigger stage $s_2 > s_1$, the $\mathcal{P}$-strategy π_1 gets A-permitting and puts x into D. Furthermore, there is no any ρ-expansionary stage between s_1 and s_2. Hence ρ still needs its restraint for z, but enumerating x into D we injury it (on the other hand, we cannot delay the enumeration of x due to the direct permitting method).

(4) At the next ρ-expansionary stage $s_3 > s_2$ the strategy ρ should correct $\Delta_{\rho,k}^{L(B_e)}(s^D(z))$, which was defined at some stage $s < s_1$. Now $\Phi_e^{B_e}(z)[s_3] = 0 \neq 1 = \Phi_e^{B_e}(z)[s_0]$, and so $B_e \upharpoonright \varphi(z, s_0)$-changes. The basic $\mathcal{R}$-strategy wants this change at some y that entered B_e before stage s. So y should leave B_e like in the case (ρ3.2B). However, as π_1 put x into D at $s_2 < s_3$, the corresponding $B_e \upharpoonright \varphi(z, s_0)$-change could be some $m < \varphi(z, s_0)$ entering B_e at some stage $t \in (s_2, s_3)$. As a consequence, ρ cannot correct $\Delta_{\rho,k}^{L(B_e)}(s^D(z))[s]$.

To solve above problem, it is adequate to avoid the enumeration like π_1 performed at stage s_2. The basic idea is to divide the enumeration and extraction of $\mathcal{P}$-strategy into two classes. Accordingly, we arrange 2-dimensional cycle outcomes, say (i, j), with lexicographic order priority for each $\mathcal{P}$-strategy.

Hence the basic $\mathcal{P}$-strategy is modified as follows. It has numbers for an enumeration x_{ij}^+ (we call them *attackers*) and numbers for an extraction x_{ij}^- (we call them *diagonalizational witnesses*). A cycle (i, j) can have only one of them, moreover a diagonalizational witnesses can be produced only from attackers. Generally, suppose π currently works at cycle (i, j) at stage s_0.

(π1) First, π checks if it has a witness for enumeration into D. If no, then it chooses a fresh number, say x_{ij}^+ and waits for $A(j)$-change, so initializes all lower priority strategies and defines $f_\pi^i(j) = A(j)[s_0]$. Then π starts a new cycle $(i, j+1)$.

(π2) If $A(j)$-change appears at stage $s_1 > s_0$, then π puts x_{ij}^+ into D. Also π initializes all lower priority strategies, starts waiting for $\Omega_\pi(s^D(x_{ij})) \downarrow= 0$.

(π3) If $\Omega_\pi(s^D(x_{ij})) \downarrow [s_2] = 0$ for some $s_2 > s_1$, then let $x_{ij}^- = x_{ij}^+$ be a witness for extracting from D. So, π starts waiting for $A(i)$-change, initializes all lower priority strategies and defines $f_\pi(i) = A(i)[s_0]$. Now π starts a new cycle $(i+1, 0)$.

By this modification, if π_1 enumerates a number then it initialize all lower priority strategies. This means that π_1 can only extract numbers from D whenever π_2 is alive simultaneously. The same as for π_2 can be said about $\mathcal{S}$- and $\mathcal{R}$-strategies. So the case (3) never happens now.

3. OUTCOMES AND PRIORITY TREE

Outcomes. $\mathcal{R}$- and $\mathcal{S}$-strategies consist of cycles k and have outcomes (k, ∞) and (k, fin), where $k \in \omega$ correspond to these cycles. For the sake of unifying we will use $(k, 0)$ and $(k, 1)$ instead of (k, ∞) and (k, fin), respectively. An additional outcome d denotes a diagonalization. The ordering is the following: $d < (0, 0) < (0, 1) < (1, 0) < (1, 1) < \ldots$. An outcome $(k, 0)$ means that we extend $\Delta_{\rho,k}^{L(B_\rho)}$ or $\Gamma_{\sigma,k}^{L(D)}$, where ρ is the corresponding $\mathcal{R}$-strategy and σ is the corresponding $\mathcal{S}$-strategy. Since A is not computable we will show that each such strategy has the greatest opened cycle k.

$\mathcal{P}$-strategies consist of 2-dimensional cycles (i, j) and have outcomes (i, j), where $i, j \in \omega$. An additional outcome d denotes a diagonalization. The ordering is the following: $d < (0, 0) < (0, 1) < (0, 2) <$

... $< (1,0) < (1,1) < (1,2) < ... < (2,0) < (2,1) < (2,2)$.... The priority between cycles corresponds to this ordering. As in above case we argue that each $\mathcal{P}$-strategy has the greatest opened cycle (i, j).

Each strategy ξ has a parameter $w^-(\xi, k, j)$, denoting whether outer cycle k of (k, j) is waiting for permitting (for $\mathcal{R}$- or $\mathcal{S}$-strategies we assume that $j = 0$ or $j = 1$, respectively). By default it equals to 0, and when outer cycle k of (k, j) of the strategy ξ is waiting for A-permitting (namely, for k entering A) we define it as 1. When outer cycle k gets the permitting we reset $w^-(\xi, k, j)$ as 0 again. For a $\mathcal{P}$-strategy ξ we also have parameter $w^+(\xi, i, k)$. The idea is the same and we define $w^+(\xi, i, k)$ as 1, when inner cycle k of (i, k) is waiting for A-permitting (namely, for k entering A).

The priority tree. Let $\Lambda = \{d, (i, j)\}_{i,j\in\omega}$ be the set of outcomes with the ordering described above. The tree of strategy T is a subtree of $\Lambda^{<\omega}$. At levels $3m$ we put $\mathcal{R}$-strategies, at levels $3m + 1$ we put $\mathcal{S}$-strategies, and at levels $3m + 2$ we put $\mathcal{P}$-strategies. Whenever we put a strategy on a tree we allow only outcomes corresponding to it.

We use standard definitions and notations for priority constructions involving tree of strategies. Also by $\Phi_\xi^{X_\xi}$ we denote $\Phi_e^{X_e}$, where e corresponds to the node ξ. For the sake of convenience we use Greek letters ρ for $\mathcal{R}$-strategies, σ for $\mathcal{S}$-strategies, and π for $\mathcal{P}$-strategies. We will omit indices when they are clear from the context. Initializing a strategy we cancel all its witnesses, functionals, close all its cycles, etc. When we initialize a strategy which was declared as satisfied we cancel this declaration.

4. CONSTRUCTION

Stage $s = 0$. $D = \varnothing$, all the strategies on T are initialized.

Stage $s + 1 > 0$. In a usual manner, by induction we define a computable approximation TP_{s+1} of the true path TP using substages $t < s$, where $TP_{s+1,t}$ is an approximation of TP_{s+1} at substage t. Whenever TP_{s+1} is defined we initialize all nodes $\eta \nleq TP_{s+1}$ and proceed to the next stage.

Each stage consists of two phases: in Phase 1 we check whether some node got a permitting, if the answer is affirmative then we go to this node immediately and work with it; in Phase 2 we work in a usual way. If we succeed in Phase 1 then we skip Phase 2.

Phase 1. Let k be the least element enumerated into A at stage $s + 1$ (w.l.o.g, we assume that only one element is enumerated into A at each stage). Check whether there exists a node ξ such that $w^-(\xi, k, j)[s] = 1$ or $w^+(\xi, i, k)[s] = 1$ for some j and i, correspondingly. If there is no such node then proceed to Phase 2. Otherwise, let ξ be the least such node.

- If $w^-(\xi, k, j)[s] = 1$ then ξ performs a diagonalization through cycle k by extracting x^-_{kj} from D, define $TP_{s+1} = \xi$, define $w^-(\xi, p, q)[s + 1] = 0$ for all $p, q \in \omega$, and declare that ξ is satisfied.
- If $w^-(\xi, k, j)[s] = 0$ but $w^+(\xi, i, k)[s] = 1$ then enumerate the attacker x^+_{ik} into D finishing attack of the cycle (i, k), close cycles (i, q) for all $q > k$, open cycle $(i + 1, 0)$, define $x^-_{ik} = x^+_{ik}$ as a diagonalization witness of outer cycle i, define $TP_{s+1} = \xi$, and define $w^+(\xi, i, q)[s + 1] = 0$ for all $q \in \omega$.

Phase 2. At substage $0 < t + 1 < s + 1$ we define $T_{s+1,t+1}$ or T_{s+1} according to the following cases (for simplicity we define $TP_{s+1,0} = \lambda$):

Case $t = 3m$, $T_{s+1,t} = \rho$, an $\mathcal{R}$-strategy.

If ρ is satisfied then define $T_{s+1,t+1} = \rho^\frown d$ and proceed to the next substage $t + 2$. Otherwise, let k be the greatest opened cycle for ρ (if there is no such cycle then open cycle $k = 0$ and proceed as below). Cycle k works as follows.

$\rho1$. If the stage $s + 1$ is not ρ-expansionary, then define $T_{s+1,t+1} = \rho^\frown(k, fin)$ and proceed to the next substage $t + 2$.

$\rho2$. Otherwise ($\rho1$ fails), let s^- be the previous ρ-expansionary stage.

ρ2.1. If there is some x (the least and unique) and y such that (1) $x < l(\rho, s^-)$, (2) $y < \varphi_\rho(x)[s^-]$, (3) $y \in B_\rho[s^-] - B_\rho[s]$, (4) x was enumerated after s^- and no numbers less than x was extracted from D after s^-, then define $w^-(\rho, k, 0)[s+1] = 1$ (so, x starts waiting for an $A(k)$-change), define the corresponding number $x^-_{k0} = x$ as a diagonalization witness associated with k. Also open the next cycle $k+1$, define $TP_{s+1} = \rho$ and proceed to the next stage $s+2$.

ρ2.2. Otherwise (ρ1 fails, ρ2.1 fails), (re-)define $\Delta^{L(B_\rho)}_{\rho,k}(z) = L(D)(z)$ with

$$\delta_{\rho,k}(z)[s+1] = \begin{cases} s+1, & \text{if} \quad z = s^D(v)\downarrow \text{ for some } v \quad \text{and} \quad \delta_{\rho,k}(z)[s^-]\uparrow, \\ \delta_{\rho,k}(z)[s^-], & \text{if} \quad z = s^D(v)\downarrow \text{ for some } v \quad \text{and} \quad \delta_{\rho,k}(z)[s^-]\downarrow, \\ 0, & \text{otherwise} \end{cases}$$

where $z < s+1$, also define $TP_{s+1,t+1} = \rho^\frown(k, \infty)$, and proceed to the next substage $t+2$.

Case $n = 3m+1$, $TP_{s+1,t} = \sigma$, an $\mathcal{S}$-strategy.

If σ is satisfied then define $T_{s+1,t+1} = \sigma^\frown d$ and proceed to the next substage $t+2$. Otherwise, let k be the greatest opened cycle for σ (if there is no such cycle then open cycle $k = 0$ and proceed as below). Cycle k works as follows.

σ1. If the stage $s+1$ is not σ-expansionary then define $T_{s+1,t+1} = \sigma^\frown(k, fin)$ and proceed to the next substage $t+2$.

σ2. Otherwise (σ1 fails), let s^- be the previous σ-expansionary stage.

σ2.1. If there is some x (the least and unique) and y such that (1) $y < l(\sigma, s^-)$, (2) $x < \theta_\sigma(y)[s^-]$, (3) $y \in W_\sigma[s+1] - W_\sigma[s^-]$, (4) x was enumerated after s^- and no numbers less than x was extracted from D after s^-, then define $w^-(\sigma, k, 0)[s+1] = 1$ (so, x starts waiting for an $A(k)$-change), define the corresponding number $x^-_{k0} = x$ as a diagonalization witness associated with cycle k. Also open the next cycle $k+1$, define $TP_{s+1} = \sigma$ and proceed to the next stage $s+2$.

σ2.2. Otherwise (σ1 and σ2.1 fail), (re-)define $\Gamma^{L(D)}_{\sigma,k}(z) = W(z)$ with $\gamma_{\sigma,k}(z) = s+1$ for all $z < l(\sigma, s+1)$, define $TP_{s+1,t+1} = \sigma^\frown(k, \infty)$ and proceed to the next substage $t+2$.

Case $t = 3m+2$, $T_{s+1,t} = \pi$, a $\mathcal{P}$-strategy.

If π is satisfied then define $T_{s+1,t+1} = \rho^\frown d$ and proceed to the next substage $t+2$. Otherwise, assume that (k, j) is the greatest opened cycle (if there is no an opened cycle then open cycle $(k, j) = (0, 0)$ and proceed as below). Recall that x^+_{kj} is the attacker for enumeration into D and x^-_{kj} is the diagonalization witness for extraction from D. Cycle (k, j) works as follows.

π1. If there is no attacker x^+_{kj} then choose x^+_{kj} as a "big" number, define $TP_{s+1} = \pi$, and proceed to the next stage $s+2$.

π2. Otherwise (π1 fails), if there is an $\mathcal{R}$-strategy $\rho^\frown(i, \infty) \subset \pi$ for some i such that $x^+_{kj} > l(\rho, s+1)$ then define $TP_{s+1} = \pi$ and proceed to the next stage $s+2$.

π3. Otherwise ($\pi1 - \pi2$ fail), if $x^+_{kj} \notin D[s+1]$ and $s^D(x^+_{kj})\uparrow$ then define $w^+(\pi, k, j)[s+1] = 1$ (so, inner cycle j starts waiting for $A(j)$-change), open cycle $(k, j+1)$, define $TP_{s+1} = \pi$, and proceed to the next stage $s+2$.

π4. Otherwise ($\pi1 - \pi3$ fail), if $s^D(x^+_{kj})\downarrow$ and $\Omega(s^D(x^+_{kj}))[s+1] \neq 0$ then define $TP_{s+1,t+1} = \pi^\frown(k, j)$, and proceed to the next substage $t+2$.

$\pi5$. Otherwise ($\pi1 - \pi4$ fail), if $s^D(x_{kj}^+)\downarrow$ and $\Omega(s^D(x_{kj}^+))\downarrow[s+1]=0$ then define $x_{kj}^- = x_{kj}^+$ and $w^-(\pi,k,j)[s+1]=1$ (so, outer cycle k defines its witness and starts waiting for $A(k)$-change), open cycle $(k+1,0)$, define $TP_{s+1}=\pi$, and proceed to the next stage $s+2$.

5. VERIFICATION

The true path is defined as $TP = \liminf_s TP_s$. Below we prove by induction that TP is infinite and exists, and that each requirement is satisfied by some strategy on TP.

Lemma 5.1. *For any $n\in\omega$ the following holds:*

(i) *$TP\restriction n$ is initialized finitely often,*

(ii) *$TP\restriction n$ initializes other nodes finitely often,*

(iii) *there exists $\liminf_s TP_s \restriction (n+1)$.*

Proof. We use an inductive argument to prove this lemma. If $n=0$ then, clearly, $TP\restriction 0 = \curlywedge$ exists, is visited infinitely often and no other strategies initialize it. So, (i) holds. For $n=0$, it is a case of an $\mathcal{R}$-strategy which can initialize other nodes only in Phase 1 or in case $\rho2.1.$ of Phase 2. After an action in Phase 1 the strategy is satisfied and doesn't initialize other strategies. Now consider case $\rho2.1.$ of Phase 2 and assume that it happens infinitely often. This means that infinitely many cycles are opened and that each cycle k has $w^-(\curlywedge,k,0)=1$ eventually. Let t_k be the least stage such that $w^-(\curlywedge,k,0)[t_k]=1$ and cycle $k+1$ is opened at this stage, then $A(k)=A_{t_k}(k)$. Hence A is computable, a contradiction. This proves (ii). Moreover, we proved that $\curlywedge$ has a greatest opened cycle, hence there exists $\liminf_s TP_s(0) = \liminf_s TP_s\restriction 1 = TP\restriction 1$ and (iii) is proved.

For inductive step, we assume that (i)-(iii) hold for $i\le n$. Clearly, $TP\restriction(n+1)$ exists and is visited infinitely often since we have (iii) by induction. By (ii) we fix a stage s_0 such $TP\restriction n$ doesn't initialize other nodes and is not initialized by other nodes after s_0. Since $TP\restriction(n+1)$ is on the True path the nodes to the left of it will be visited only finitely many times. So, fix a stage $s_1>s_0$ such that this situation will not happen. From then on, $TP\restriction(n+1)$ cannot be initialized in Phase 1 or Phase 2 by other nodes. So, (i) holds for $n+1$.

Further, we work at stages $>s_1$ and proceed to the proof of (ii) and (iii). Consider the cases when $TP\restriction(n+1)$ correspond to $\mathcal{R}$-, $\mathcal{S}$- and $\mathcal{P}$-strategies. If $n+1=3m$, then we repeat the induction basis.

Assume that $n+1=3m+1$. An $\mathcal{S}$-strategy σ can initialize other nodes only in Phase 1 or in case $\sigma2.1.$ of Phase 2. After an action in Phase 1 the $\mathcal{S}$-strategy is satisfied and then doesn't initialize other strategies. Now we need consider only case $\sigma2.1.$ of Phase 2. Assume that it happens infinitely often. This means that infinitely many cycles are opened and that each cycle k has $w^-(\sigma,k,0)=1$ eventually. Hence A is computable (we fix a stage $s_2>s_1$ such that Phase 1 is never happened, and to compute $A(k)$ we wait until cycle $k+1$ is opened after the stage s_2, then $A(k)=A_{s_k}(k)$), a contradiction.

Assume that $n+1=3m+2$. An $\mathcal{P}$-strategy π can initialize other nodes only in Phase 1 or in case $\pi1$, $\pi2$, $\pi3$, and $\pi5$ of Phase 2. After an action in Phase 1 the $\mathcal{P}$-strategy is satisfied and then doesn't initialize other strategies. Now we need consider only the cases with Phase 2. For contrary, assume that one of the mentioned cases ($\pi1$, $\pi2$, $\pi3$, or $\pi5$) happens infinitely often. Clearly, if $\pi1$ happens infinitely often then the same holds for $\pi2$ and $\pi3$. If $\pi3$ happens infinitely often then $\pi5$ happens either finitely often or infinitely often.

• In the first (finite) case, there exist i such that for all $k\in\omega$ cycle (i,k) is opened. So, we can effectively wait until cycle $(i,k+1)$ becomes opened at stage $t_k>s_1$. Hence $A(k)=A_{t_k}(k)$ and A is computable, a contradiction.

• In the second (infinite) case, there are infinitely many opened cycles $(k,0)$. So we can effectively wait until cycle $(k+1,0)$ becomes opened at stage $t_k>s_1$. Hence $A(k)=A_{t_k}(k)$ and A is computable, a contradiction.

This finishes the proof of (ii) for $n+1$. Also we proved that $TP\restriction n$ has the greatest opened cycle, hence there exists $\liminf_s TP_s\restriction(n+1)$. So, (iii) and the whole lemma are proved. □

Lemma 5.2. *For any $n\in\omega$ the requirements $\mathcal{R}_n$, $\mathcal{S}_n$, $\mathcal{P}_n$ are satisfied.*

Proof. We prove that $\mathcal{R}_n$ is satisfied by $\rho \in TP$, where $|\rho| = 3n$. By Lemma 5.1 fix the least stage s_0 such that ρ is not initialized after it. If $\rho^\frown d \in TP$ or ρ has finitely many expansionary stages then $\mathcal{R}_n$ is satisfied vacuously. Otherwise, let k be the greatest opened cycle of ρ after the stage s_0, so we prove that $L(D) = \Delta_{\rho,k}^{L(B_n)}$.

Note that $\delta_{\rho,k}$ is well-defined (see case $\rho 2.2$ of the construction). For the functional, consider the non-trivial case when x was an attacker of some $\pi \supset \rho^\frown(k,\infty)$ and $s^D(x)$ is defined. Let s_1 be a ρ-expansionary stage when we defined $L(D)(s^D(x))[s_1] = \Delta_{\rho,k}^{L(B_n)}(s^D(x))[s_1] = 0$. There are two cases.

Case (1) There exists a ρ-expansionary stage $s_2 > s_1$ such that $s^D(x) \in L(D)[s_2]$. So, x got A-permitting and later only numbers $< x$ could get an A-permitting between s_1 and s_2. Consider the following cases which can happen between these stages:

(1.1) There wasn't a Phase 1. When we defined $\Delta_{\rho,k}^{L(B_n)}(s^D(x))[s_1] = 0$ we got $\delta_{\rho,k}^{L(B_n)}(s^D(x)) = s_1$. Also between stages $s^D(x) < s_2$ no other numbers $< x$ got A-permitting (otherwise π (or some $\mathcal{S}$- or $\mathcal{R}$- strategy below π which got this x) never extracts x from D). So, at stage s_1 we have some $y \in B_n[s_1] - B_n[s_1^-]$ such that $y < \varphi_n(x)[s_1^-]$, where s_1^- is the previous ρ-expansionary stage. Moreover, $D[s_1^-] \upharpoonright r = D[s_2] \upharpoonright r$, where $r = \psi_n(y)[s_1^-]$. Therefore $B_n(y)[s_2] = B_n(y)[s_1^-]$, but this means that $L(B_n) \upharpoonright s_1$ changed because of $s^{B_n}(y) < s_1$. Hence $L(D)(s^D(x))[s_2] = \Delta_{\rho,k}^{L(B_n)}(s^D(x))[s_2] = 1$ is the final value.

(1.2) Between $\rho^\frown(k,\infty)$ and π it was Phase 1 with an enumeration, then it should enumerate some $x_{ik} < x$ and this x_{ik} was waiting for $A(k)$-change, however during this waiting it can only circulate between cases $\pi 1$, $\pi 2$ and $\pi 3$. Hence we always initialize all nodes below it, and in particular π. So π cannot come even to its enumeration stage and this case is not valid.

(1.3) Between $\rho^\frown(k,\infty)$ and π it was Phase 1 with an extraction, then it should extract some $x^-_{\xi,kj} < x$, moreover $s^D(x^-_{\xi,kj}) < s^D(x)$. Let ξ be the least node with such Phase 1. Hence $L(B_n)[s_1] \upharpoonright s^D(x^-_{\xi,kj})$ is changed (see 1.1) and we can redefine $\Delta_{\rho,k}^{L(B_n)}(s^D(x^-_{\xi,kj}))[s_2]$ and so $L(D)(s^D(x))[s_2] = \Delta_{\rho,k}^{L(B_n)}(s^D(x))[s_2] = 1$.

Case (2) Such stage $s_2 > s_1$ doesn't exist.

In both cases, $L(D)(s^D(x)) = L(D)(s^D(x))[s] = \Delta_{\rho,k}^{L(B_n)}(s^D(x))[s] = \Delta_{\rho,k}^{L(B_n)}(s^D(x))$, where $s = s_2$ or s_1, respectively.

Now prove that $\mathcal{S}_n$ is satisfied by $\sigma \in TP$, where $|\sigma| = 3n + 1$. By Lemma 5.1 fix the least stage s_0 such that σ is not initialized after s_0. If $\sigma^\frown d \in TP$ or σ has finitely many expansionary stages then $\mathcal{S}_n$ is satisfied vacuously. Otherwise, let k be the greatest opened cycle of σ after the stage s_0, so we prove that $W_n = \Gamma_{\sigma,k}^{L(D)}$. Assume that $\Gamma_{\sigma,k}^{L(D)}(y)[s_1] = W_n(y)[s_1] = 0$ was defined at σ-expansionary stage $s_1 > s_0$. There are two possibilities now.

Case (1) There exists a σ-expansionary stage $s_2 > s_1$ such that $y \in W_n[s_2] - W_n[s_1]$. Consider the following cases which can happen between these stages:

(1.1) There exists $z < \theta_n(y)[s_1]$ such that $z \in D[s_1] - D[s_2]$. Since $\gamma_{\sigma,k}(y)[s_1] = s_1$ by case $\sigma 2.2$ and $s^D(z) < s_1$, we can redefine $\Gamma_{\sigma,k}(y)[s_2] = W_n(y)[s_2] = 1$ with $\gamma_{\sigma,k}(y)[s_2] = s_2$.

(1.2) Case (1.1) doesn't hold and there exists $z < \theta_n(y)[s_1]$ such that $z \in D[s_2] - D[s_1]$. However, according to the construction case $\sigma 2.1$ we must open cycle $k + 1$, a contradiction.

Case (2) Such stage $s_2 > s_1$ doesn't exist. Here we need show that $\gamma_{\sigma,k}(y)$ is well defined. Indeed, assume the contrary. Then there exist infinitely many σ-expansionary stages $s_1 < ... < s_m$ such that $z_m \notin D[s_m]$ and $s^D(z_m) \downarrow < s_m < \gamma_{\sigma,k}(y)[s_m]$. However, this means that between σ and the node, which extracted some z_m, there are infinitely many nodes which got permitting, a contradiction.

In both cases, $W_n(y) = W_n(y)[s] = \Gamma^{L(D)}_{\sigma,k}(y)[s] = \Gamma^{L(D)}_{\sigma,k}(y)$, where $s = s_2$ or s_1, respectively.

Eventually, prove that $\mathcal{P}_n$ is satisfied by $\pi \in TP$, where $|\pi| = 3n + 2$. By 5.1 fix the least stage s_0 such that π is not initialized after it and fix the greatest opened cycle (i, j) after s_0 (henceforth, no cycle $> (i, j)$ is opened after s_0). If $\pi^\frown d \in TP$ then it is satisfied vacuously. Otherwise, we assume that only cycle (i, j) is working after some stage $s_1 > s_0$. This means that it gets stuck at $\pi 4$, i.e. there is no a stage $s > s_1$ such that $\Omega_n(s^D(x^+_{ij}))[s] = 0$, hence it is a diagonalization by $\Omega_n(s^D(x^+_{ij})) \neq 0$ (if A changes at some $k \leq i$ after s_0 then we extract x^-_{km}, where m is such that (k, m) is the greatest cycle in the series k; hence $\Omega_n(s^D(x^-_{km})) \downarrow= 0 \neq 1 = L(D)(s^D(x^-_{km}))$ and $\pi^\frown d \in TP$). This completes the proof of the lemma and the theorem. □

ACKNOWLEDGMENTS

Liu is partially supported by NSFC grant (no. 61202131), the CAS Western Light Program and CAS Youth Innovation Promotion Association (no. 2015315). Wu is partially supported by MOE2011-T2-1-071 (ARC 17/11, M45110030) from Ministry of Education of Singapore, and by AcRF grants RG29/14, M4011274. Yamaleev is supported by The President grant of Russian Federation (project NSh-941.2014.1), by Russian Foundation for Basic Research (projects 14-01-31200, 15-01-08252), by the subsidy allocated to Kazan Federal University for the project part of the state assignment in the sphere of scientific activities, and by MOE2011-T2-1-071 (ARC 17/11, M45110030) from Ministry of Education of Singapore. The work is performed according to the Russian Government Program of Competitive Growth of Kazan Federal University.

REFERENCES

1. M. Arslanov, I. Kalimullin, and S. Lempp, J. Symbolic Logic **75** (2), 401–441 (2010).
2. G. Wu, J. Symbolic Logic **67**, 1055–1064 (2002).
3. B. Cooper and A. Li, Science in China (Series A) **45**, 1135–1146 (2002).
4. C. Fang, G. Wu, and M. Yamaleev, Arch. Math. Logic **52**, 733–741 (2013).
5. Sh. Ishmukhametov, Arch. Math. Logic **38**, 373–386 (1999).
6. R. Soare, in *Recursively Enumerable Sets and Degrees* (Springer-Verlag, Heidelberg, 1987), p. 437.