On a problem of Cooper and Epstein

Shamil Ishmukhametov
Kazan Federal University,
42008, Kazan, Russia
Shamil.Ishmukhametov@ksu.ru
J. Symbolic Logic Volume 68, Issue 1 (2003), 52-64

Abstract

In "Bounding minimal degrees by computably enumerable degrees" by A.Li and D.Yang, (this Journal, [1999]), the authors prove that there exist non-computable computably enumerable degrees \( c > a > 0 \) such that any minimal degree \( m \) being below \( c \) is also below \( a \). We analyze the proof of their result and show that the proof contains a mistake. Instead we give a proof for the opposite result.

In [1992] Seetapun and Slaman proved, extending the Posner and Robinson Complementation Theorem (Posner and Robinson [1981], Posner [1981]), that every non-zero degree below \( 0' \) is complemented by a minimal degree. In [1979] Epstein has conjectured that in a lower cone \( D(\leq c \text{ c.e.}) \) any non-zero c.e. degree \( a < c \) has a minimal complement and proved the conjecture for the case \( c \) is high (see [1981]). But the whole conjecture was rejected by Cooper and Epstein [1987]. Later Cooper [1989] and independently Slaman and Steel [1989] have shown that there is a lower cone \( D(\leq c \text{ c.e.}) \) in which some c.e. \( a \) is not cupped. Moreover, this \( a \) can be chosen (see Cooper [1994]) as the greatest degree in \( D(<c \text{ c.e.}) \) which is not cupped to \( c \).

In [1987] Cooper and Epstein proved that in \( D(<c \text{ c.e. low}) \) each c.e. \( a \) is capped to \( 0 \) by a minimal degree. They conjectured that their result is the best possible and one cannot drop assumption on \( a \) to be low or c.e.

In [1999] Li and Yang claimed to confirm the conjecture by showing that there exist two computably enumerable degrees \( a \) and \( c, 0 < a < c \), such that for
every minimal degree \( m, m < c \) implies \( m < a \). We analyze the proof and show that it contains a mistake. Instead using a new method for the construction of minimal degrees we show that in a lower cone \( D(<c\text{.c.e.}) \) any c.e. degree \( a \) can be capped to 0 by a minimal degree \( m < c \). As a corollary we obtain that 0 is branching in any cone \( D(<c\text{ c.e.}) \) in \( \omega\text{-c.e.} \) degrees. This result is the best possible since by the Lachlan Non-Bounding Theorem [1979] 0 is not branching in some cone \( D(<c\text{-c.e.}) \) in the c.e. degrees and, therefore, in the \( n\text{-c.e.} \) degrees for \( n \in \omega \). See also Downey and Stob [1997], where the authors prove that below every non-zero c.e. degree \( a \) there exists a c.e. degree that is not a half of a minimal pair (in the c.e.degrees) in \( D(<a\text{-c.e.}) \).

In our proof we use a priority version of the tree method for the construction of the required minimal degree. Such approach allows us to separate the work of different strategies and simplify the proof. Our notations are standard and follow R.I.Soare [1987]. We denote partial computable functionals by capital Greek letter and their uses by small Greek letters. We assume that all uses are non-decreasing on their arguments.

1 Analyzing Li and Yang’ theorem

**Theorem 1** (Li, Yang [1999]) There exist computably enumerable degrees \( a > b > 0 \) such that for every minimal degree \( m, m < a \) implies \( m < b \).

First we write down the list of requirements posed in the theorem:

- \( P_e : A \neq \Psi_e(B) \);
- \( R_e : A_e = \Phi_e(A, B) \text{ total} \rightarrow [A_e \leq_T B \text{ or } C_e = \Theta_e(A_e)] \);
- \( S_{e,0}^0 : A_e \text{ total } \& C_e = \xi_i \rightarrow A_e \leq_T B \);
- \( S_{e,i}^1 : A_e \text{ total } \& A_e = \Lambda_i(C_e) \rightarrow A_e \leq_T B \),

where \( \{\Psi_e, \Phi_e, \xi_i, \Lambda_e\}_{e \in \omega} \) is a computable enumeration of all possible tuples of p.c.functionals. Sets \( C_e \) and p.c.functionals \( \Theta_e \) and \( \Gamma_e \) reducing \( A_e \) to \( B \) for \( e \in \omega \) are built in the construction. The required degrees \( a \) and \( b \) are defined as the degrees of sets \( A \) and \( A \oplus B \) respectively.

Clearly, the requirements insure the theorem. We consider below a certain combination of requirements that cannot be satisfied by the strategies announced by the authors. Namely, we consider case \( R_e < S_{e,i} < P_k \) assuming \( A_e = \Phi_e(A, B) \) is total.
In the considered context $R$ has the highest priority and cannot be injured. The module for $R$ builds a p.r.f. $\Theta$, $C_e = \Theta_e(A_e)$. Let us analyze this module. Since both $A$ and $B$ must be non-computable and $e$ arbitrary, $A_e^s$ is an approximation to $\Delta_2^0$ set and takes different values during the construction. For example, if $A_e$ has a minimal degree, any approximation necessary oscillates around several positions taking them by turns. Assume that at stage $s_0 A_e^s \upharpoonright s = \tau$ and we define $\rho = \Theta(\tau)$, $C_e \supseteq \rho$. Later, at a stage $s_1 > s_0 A_e^s = \tau'$ where $\tau' \neq \tau$. Insuring $R$ we need to make a new definition, $\rho' = \Theta(\tau')$ forcing $C_e$ to extend $\rho$ when $A_e$ extends $\tau$ and $\rho'$ when $A_e$ extends $\tau'$. If set $C_e$ is non-computable then eventually range$(\Theta)$ contains pairs $(\tau, \rho)$ with incomparable $\rho$.

Notice that this means that set $C_e$ is completely defined by $\Theta$ and the authors’ instructions for $R$ module (to switch $C(x)$ when we see $C(x) \neq \Theta(A_e; x)$) are either redundant, or mistaken. The only case when we can choose a value for $C_e(x)$ is case when $A_e$ extends some $\tau$ at which $\Theta$ was not already defined but even in this case defining $\rho = \Theta(\tau)$ we must insure $\rho' \subset \rho$, if $\rho' = \Theta(\tau')$ was defined earlier and $\tau' \subset \tau$.

Resuming we note that in the construction we have a very weak influence on $C_e$ and when $A_e$ returns to a previous branch at which $\Theta$ was already defined, $C_e$ must eventually take the appropriate value without any possibility for us to change $C_e$.

Now we turn to the authors’ module for the combination of requirements above. The module for $S_1^1$ defines a p.r.f. $\Gamma$, $A_e = \Gamma(B)$. When we need to put a witness $a$ to $A$ satisfying some $P$ below this can imply the change $A_e(x)$ for some $x$ and in order to restore $A_e(x) = \Gamma(B; x)$ we need to put into $B$ $\gamma(x)$ where $\gamma(x)$ denotes the number equal to the use of $\Gamma(B; x)$ minus 1. The last action destroys the strategy for $P$ if $\gamma(x) < \psi(a)$, and $P$ is not satisfied.

In order to prevent this situation the authors suggest the following corrections to the module for $P$ (we omit a cumbersome system of the notations of the authors retaining only sufficient details):

1. Choose a witness $a$ for $P$ and wait for a stage $s_0$ when $\Psi(B; a) \downarrow = 0$.
2. Choose a witness $c > \psi(a)$ and wait for $A_e \upharpoonright (c + 1)$ defined.
3. Enumerate $a$ into $A$.
4. Wait for the next $R$-expansionary stage. Then consider the corresponding case:

   A. There is a (least) $x_0 \leq c$ such that $\Gamma(B; x_0) \neq A_e(x_0)$. Redefine
\[ \Theta(A_e; y) \uparrow = \text{old } C_e(y) \text{ for all } y \leq \lambda(x_0) \]

B. Otherwise. Restrain \( A \upharpoonright s, B \upharpoonright s \) where \( s \) is the current stage.

By the authors’ intention the action of item A should keep the old value \( \Lambda_i(C; x_0) \) while \( A_e(x_0) \) changes. This creates the disagreement between \( \Lambda_i(C) \) and \( A_e \) on \( x_0 \) and the restraint on sets A and B keeps it at later stages thus satisfying \( S_{1,i} \) by a finite way.

We assert, however, that this is incorrect and the action of item A cannot insure the disagreement \( \Lambda_i(C; x_0) \neq A_e(x_0) \). Indeed, in order to preserve \( \Lambda_i(C; x_0) \) we need to preserve \( C(y) \) for all \( y \leq \lambda(x_0) \) and if even a single \( C_e(y) \) changes, this can imply the change of \( \Lambda_i(C; x_0) \) thus failing the authors’ strategy. But the module described can insure only one uncommitted point, namely, \( C_e(c) \) which we can remain unchanged. Thus, the module described in parts 1.5-1.6 cannot insure the satisfaction of requirement \( P \).

We note also that a complicated system of the checking in 1.6 that all parts of the strategy are implemented according to the authors’ intention is useless because all damaging actions consist of several different definitions of functional \( \Theta \) and are made between stage \( s_0 \) when witness \( a \) was chosen and \( s_1 \) when \( \Psi(B; a) \downarrow = 0 \). While \( \Phi(B; a)[s] \) is undefined we cannot stop the definition of \( \Theta \) since \( \Phi(B; a) \) can diverge. Between stages \( s_0 \) and \( s_1 \) positive requirements \( P \) act causing a change of \( A_e \) and add to range of \( \Theta \) new pairs \( \langle \tau, \rho \rangle \) with incomparable \( \rho \). Therefore, there is no obvious way of implementing the authors’ strategy. Nothing new appears in descriptions of other combinations of the modules.

In the description of the whole construction the corresponding case is considered on page 1337, program \( \gamma \), case 2a. Here the authors only claim to change those \( \Theta(A_e; y) \) for \( y \leq \lambda(x_0) \) which are undefined. There is no proof that all required values \( \Theta(A_e; y) \) can be redefined in the such way.

Finally, in S-satisfaction lemma 5.16 (the last lemma of the paper) we find that the required argument \( x \) insuring the disagreement \( \Lambda_i(C; x) \neq A_e(x) \) exists due to the \( S \)-module (or, \( \beta \)-program) without any further explanations.

Resuming we note that the proof of the theorem cannot be considered as sufficient. In the next section we give a proof of the opposite theorem.
2 Capping in lower cones

Theorem 2 (The Capping Theorem). For any c.e. \( a \) and \( c \) such that \( 0 < a < c \) there is an \( \omega \)-c.e. minimal degree \( m < c \) incomparable with \( a \).

Proof. Let \( A = \lim A^s \) and \( C = \lim C^s \) be c.e. sets from degrees \( a \) and \( c \) respectively with their approximations. We construct a \( \Delta^0_2 \)-approximation \( \{M^s : s \in \omega\} \) to a set \( M \) satisfying the following list of requirements:

- \( N_e : M_e = \Phi^M_e \) total \( \rightarrow (M = \Gamma_e(M_e) \lor M_e = \Delta_e) \),
- \( P_e : M \neq \Theta_e(A), e \in \omega \).

where \( \{\Phi_e, \Theta_e\}_{e \in \omega} \) is a standard enumeration of all possible pairs of partial computable functionals.

Requirements \( N \) ensure that any set \( M_e \) computable in \( M \) (via p.c.f. \( \Phi_e \)) is either computable, or \( T \)-equivalent to \( M \). This implies that the degree of \( M \) is minimal. Requirements \( P \) ensure that the degree of \( M \) is incomparable with \( a \). Additionally we pose a global requirement \( M \leq_T C \).

We define two length functions \( l_N(e, s) \) and \( l_P(e, s) \):

\[
\begin{align*}
\text{\( l_N(e, s) = \min\{x : \Phi_{e, s}(M^s; x) \uparrow\}\)} \\
\text{\( l_P(e, s) = \max\{x : \Theta_{e, s}(A^s; x) \uparrow x = M^s \uparrow x\}\).}
\end{align*}
\]

Below we shall describe modules for various combinations of these requirements. We begin with the module for a negative requirement \( N_e \):

The module for requirement \( N_e \):

1. Wait for a \( N_e \)-expansionary stage \( s \) (i.e. a stage when \( l_N(e, s) \) exceeds \( \max\{l_N(e, t), t < s\} \)).
2. Define \( \Delta(y) = M_e(y) \) for all \( y < l_N(e, s) \), at which \( \Delta(y) \) was undefined yet. Go to 1.
3. Let \( s' < s \) be the largest stage at which the current \( \Delta \) was compatible with \( M_e \).

We notice that at many positions taken by \( M^r, r \leq s \), before this stage we found two, namely, \( M^s \uparrow s' \) and \( M^s \uparrow s \), at which the functional \( \Phi_e \) differs on argument \( x \). In other words we found an \( e \)-splitting on \( \Phi_e \).
Let $\tau_0 = M^{s'} \upharpoonright s'$ and $\tau_1 = M^s \upharpoonright s$. At this moment we pose a $N$-restraint on $M$ allowing $M$ to change only such that $M$ always extends either $\tau_0$ or $\tau_1$. Let $\rho_i = \Phi_e(\tau_i) \upharpoonright x + 1$, $i \in \{0, 1\}$. Define $\Gamma$ at nodes $\rho_0$ and $\rho_1$ equal respectively to $\tau_0$ and $\tau_1$. This definition is correct since $\rho_0$ and $\rho_1$ differ on $x$.

4. Cancel current $\Delta$, go to 1, and begin to define a new $\Delta$.

This completes the description of the module. During the procedure we can infinitely often cancel functionals $\Delta$. Each time when the current $\Delta$ is cancelled we extend the functional $\Gamma$. This procedure is similar to the procedure of the definition of a computable splitting tree in the finite approximations. The range of functional $\Gamma$ forms a partial computable tree and consists of pairs $\langle \tau_0, \tau_1 \rangle$ added during the procedure to $\Gamma$. Arguments of each next pair $\langle \tau'_0, \tau'_1 \rangle$ always extend an argument $\tau_i$ of some previous pair $\langle \tau_0, \tau_1 \rangle$.

We call these pairs candidats of $N_e$ since they are used for the satisfaction of positive requirements $P$.

Constructing $\Gamma$ we build a tree of possible beginnings for $M$. Each splitting point of this tree has two possible extensions at one of which the previous $\Delta$ remains correct and at another a new $\Delta$ begins. If the tree has infinitely many splittings then we have a total $\Gamma(M_e) = M$, otherwise, $\Phi_e(M)$ is either partial or coincides with a computable $\Delta$.

The module for $N_e$ has two possible outcomes. The finitary outcome 1 means that either there exists a total $\Delta = M_e$ or $\Phi_e(M)$ is not total. Notice that in this case there is a stage $s_0$ after which new values of $\Gamma$ are not defined. The infinitary outcome 0 means that both $M_e$ and $\Gamma(M_e)$ are total. In this case there are infinitely many candidates of $N$ (or $e$-splittings in the range of $\Gamma$).

Now we turn to the module for a positive requirement.

**The module for requirement $P_e$:**

It consists of a list of procedures. Procedure $\langle 0 \rangle$ starts first. Procedure $\langle n \rangle$ defines the value $\Sigma^A(n)$ of a partial computable functional $\Sigma$ such that if the procedure fails to satisfy $P$ then $C(n) = \Sigma^A(n)$. Since $C \nleq_T A$, then eventually a natural $n$ will appear such that procedure $\langle n \rangle$ satisfies $P$. Fix some $n$ and consider instructions for procedure $\langle n \rangle$: 
1. If \( n \in C^s \) define \( \Sigma^A(n) = 1 \) with use \( \sigma(n) = \sigma(n - 1) \) (by definition, \( \sigma(-1) = 0 \)) and start procedure \( \langle n + 1 \rangle \). Otherwise, go to the next item.

2. Choose a free witness \( x_n > n \).

3. Wait for a stage \( s' \) when either \( l_P(e, s') > x_n \), or \( n \in C^{s'} \).
   If the latter occurs first, go to 1, otherwise, define \( \Sigma^A(n) = 0 \) with use \( \sigma(n) = \theta(x_n) \). Start procedure \( \langle n + 1 \rangle \), then go to the next step.

4. Wait for \( C(n) \) or \( A \upharpoonright \sigma(n) \) to change.
   If the latter occurs first, initialize all procedures \( > n \) then return to 3.
   Otherwise, change \( M(x_n) \) and go to 5.

5. Wait for \( A \upharpoonright \sigma(n) \) to change. Then initialize all procedures \( > n \) and return to 3.

This completes the instructions. Note that by the module, the change of \( M(x_n) \) is preceded by the change of \( C(n) \) so the module ensures a permitting of \( M \) by \( C \) and thus the global requirement \( M \leq_T C \). Below we verify the module and show that \( M \neq \Theta_e(A) \).

Assume the converse. Then \( \lim_s l_P(e, s) = \infty \) and infinitely many procedures are started. Each procedure is initialized finitely often (by the previous ones). Indeed, procedure \( \langle 0 \rangle \) is never initialized. Assume this for all \( k \leq n \) and let \( s_0 \) be the least stage after which procedure \( \langle n \rangle \) is not initialized. By the module, the witness \( x_n \) of procedure \( \langle n \rangle \) chosen after stage \( s_0 \) is never changed and \( \Sigma(n) \downarrow = C(n) \) iff \( \Theta(x_n) \downarrow = M(x_n) \). There exists a stage \( s_1 \geq s_0 \) when the value of \( \Theta(A; x_n) \) establishes \( (A^s \upharpoonright \theta_s(x_n) = A \upharpoonright \theta_s(x_n)) \). After this stage \( \Sigma(A; n) \) never changes and procedure \( \langle n \rangle \) cannot initialize procedures with greater numbers.

So our assumption that \( \Sigma(A) \) is total implies \( \Sigma(A) = C \) contradicting to \( C \not\leq_T A \). Therefore, the requirement \( P \) is satisfied and there is a least \( n \) such that \( C(n) \neq \Sigma(n) \). This completes the module.

The module for \( P \) below \( N \).

We have two strategies for \( P \) below \( N \). The module for \( P \) assuming the finitary outcome for \( N \) works as alone. Below we give instructions for the second module assuming the infinitary outcome for \( N \). Let \( \alpha \) and \( \beta \) denote the modules for \( N \) and \( P \) respectively.

1. If \( n \in C^s \), define \( \Sigma(A; n) = 1 \) with use \( \sigma(n) = \sigma(n - 1) \) and start procedure \( \langle n + 1 \rangle \). Otherwise, go to the next item.
2. Wait for a candidate \( \langle \tau_0, \tau_1 \rangle \) of \( N \) with the splitting point \( z_n = \mu z[\tau_0(z) \neq \tau_1(z)] \) exceeding all parameters of procedures \( \langle k \rangle, k < n \), to appear. Appoint it to be the current witness of procedure \( \langle n \rangle \).

3. Wait for a stage \( s' \) when either \( l_P(e, s') > z_n \) or \( n \in C_{s'} \).

If the latter occurs first, return to 1, otherwise, define \( \Sigma^A(n) = 0 \) with use \( \sigma(n) = \theta(z_n) \). Start procedure \( \langle n + 1 \rangle \) and go to the next step.

4. Wait for \( C(n) \) or \( A \upharpoonright \sigma(n) \) to change.

If the latter occurs first initialize all procedures \( \rangle n \) and return to 3. Otherwise, replace initial segment of \( M \) of length \( |\tau_0| \) by \( \tau_0 \) and go to 5.

5. Wait for \( A \upharpoonright \sigma(n) \) to change. Then initialize all procedures \( \rangle n \) and return to 3.

This completes the instructions. Notice that the main difference with the previous module is how we change \( M \). Now instead of the change of \( M \) on a single argument \( x_n \) we simultaneously change \( M(y) \) for all \( y < |\tau_0| \) such that \( \tau_0(y) \neq \tau_1(y) \). By the choice of pair \( \langle \tau_0, \tau_1 \rangle \) the splitting point \( z_n = \mu y[\tau_0(y) \neq \tau_1(y)] \) exceeds \( n \) and again the change of \( M \) is permitted by the change of \( C(n) \) and the module insures \( M \leq_T C \). The appearance of the required witness \( \langle \tau_0, \tau_1 \rangle \) in item 2 is insured by the assumption \( \alpha \) has the infinitary outcome.

Note that in the both cases \( M \) extends a maximal (by inclusion) node of range(\( \Gamma \)) defined by \( \alpha \) so \( \beta \) preserves the correct implementation of \( \alpha \).

Comparing two versions of the module for \( P \) we conclude that the first version is a partial case of the second. Indeed, to make the modules compatible we need only to supply the first module with the tree consisting at a stage \( s \) of all possible pairs \( \langle \tau_0, \tau_1 \rangle \) such that \( |\tau_0| = |\tau_1| \leq s \), \( \tau_1 \) is an initial segment of \( M^s \), and strings \( \tau_0 \) and \( \tau_1 \) differ exactly on the last argument \( z = |\tau_0| - 1 \). Then the replacement of the current initial segment of \( M \) of length \( z + 1 \) by \( \tau_0 \) is equivalent to the change of \( M(z) \). So we have the common module for each kind of a positive requirement.

The module has \( \omega \) possible outcomes which we put into the tree of outcomes. Outcome \( n \) denotes the case \( n \) is the least number such that \( C(n) \neq \Sigma(A; n) \). If \( \Sigma(A; n) \downarrow \) then finitely many procedures are started and there is a stage \( s_0 \) such that lower priority requirements are not destroyed by the strategy after stage \( s_0 \).

We consider case \( \Sigma(A; n) \uparrow \). Let \( l_P(\beta, s) = \mu k\{\Sigma(A; k) \neq C(k)[s]\} \) be a length function for positive strategies.
As in the other infinite injury constructions, we need to ensure that
\( \lim \inf l_P(\beta, s) < \infty \). With this purpose we correct approximations \( \{\Theta_s\} \) to functionals \( \Theta \) as in Soare [1987]. Namely, we accept that for any \( x \) and \( s \) if \( \Theta(A^s; x) \downarrow \) then \( \theta(x) \leq a_s \) where \( a_s \) is element enumerated in \( A \) at stage \( s \). Besides, we assume that the use function \( \theta \) is non-decreasing on \( x \). The latter implies that if \( \Theta(x) \) becomes undefined at a stage \( s \), so are all values \( \Theta(y) \) for \( y > x \).

**Definition.** Stage \( s + 1 \) is called \( A \)-true if \( A^s \upharpoonright a_s = \{ a \upharpoonright \} \).

Due to the correction made to functionals \( \Theta \) all computations of kind \( \Theta(A^s; x) \) with \( A^s \upharpoonright \theta(x) \neq A \upharpoonright \theta(x) \) are destroyed at \( A \)-true stages and only valid computations remain. In the case of the module for \( P \) this means that at \( A \)-true stages all procedures \( \langle k \rangle, k > n \), are initialized and strategies \( \alpha \supseteq \beta \ast \langle n \rangle \) can act.

**The module for \( N_e \) below \( N \):**

In this description we outline only sufficient details. Let \( \alpha \) denote the strategy for \( N \). There are two modules for \( N_e \) below \( N \). The first one assumes the finitary outcome of \( \alpha \) and works as in isolation.

Let \( \beta \) be the module for \( N_e \) assuming the infinite outcome for \( \alpha \). \( \alpha \) builds a functional \( \Gamma_\alpha \) and all modules below \( \alpha \ast \langle 0 \rangle \) work under restriction on \( M \) to extend at each stage \( s \) of the construction some maximal by inclusion node of \( \text{range}(\Gamma_\alpha) \). \( \gamma \) constructs its own functionals \( \Delta_\beta \) and \( \Gamma_\beta \) and should insure that its candidates \( \langle \tau_0, \tau_1 \rangle \) form \( e \)-splittings as well as \( |N| \)-splittings. The instructions do not differ from the isolated module described above with the only exception concerning the way of the definition of new candidates.

Since \( \beta \) is located below \( \alpha \) it is reached (due to the standard procedure of the traversing of the priority tree) only at \( \beta \)-stages which form a subset of \( \alpha \ast \langle 0 \rangle \)-stages while the last is by definition the set of \( \alpha \)-stages when \( \Gamma_\alpha \) is extended. This automatically implies that arguments of every pair \( \langle \tau_0', \tau_1' \rangle \) added to \( \text{range}(\Gamma_\beta) \) are initial substrings of maximal by inclusion elements of \( \text{range}(\Gamma_\alpha) \).

In terms of the tree constructions this means that the partial tree \( T_\beta \) formed by range of \( \Gamma_\beta \) is a subtree of the corresponding tree \( T_\alpha \).

Assume now that (at a stage \( s \)) \( \beta \) needs to add to the range of \( \Gamma_\beta \) a new pair \( \langle \tau_0, \tau_1 \rangle \). In order to guarantee that at each stage \( s \) \( M \) extends some maximal node of both functionals \( \Gamma_\alpha \) and \( \Gamma_\beta \) we need to ensure that the first
argument $\tau_0$ is a maximal node in range($\Gamma_\alpha$). With this purpose we make the following:

1. We define pair $\langle \tau_0, \tau_1 \rangle$ by the instructions for $\beta$,
2. Find in the range of ($\Gamma_\alpha$) a pair $\langle \tau'_0, \tau'_1 \rangle$ with $\tau'_0$ extending $\tau_0$,
3. Add to range($\Gamma_\beta$) the candidate $\langle \tau'_0, \tau_1 \rangle$.

The module for $N$ below $P$.

Now we consider a collaboration of a positive strategy $\beta$ with negative strategies located below $\beta$. Let length function $l_P(\beta, s)$ be as earlier, i.e. $l_P(\beta, s) = \mu k [\Sigma_\beta(A^s; k) \neq C^s(k)]$. We allow at stage $s + 1$ to act those strategies $\gamma$ that extends $\beta \ast \langle k \rangle$ with $k = l_P(\beta, s)$. Let $n = \lim \inf l_P(\beta, s)$ and consider $\alpha = \beta \ast \langle k \rangle$ for some $k$. There are three possible cases:

1. $k < n$. Then $\alpha$ is visited finitely often. Defining tree $T_\alpha$ it poses additional restrictions on change of $M$ and procedures of $\beta$ with numbers $m > k$ can choose witnesses with both arguments $\tau_0, \tau_1$ extending a maximal by inclusion node of $T_\alpha$. So actions of such procedures cannot destroy $\alpha$. If some procedure $m \leq k$ of $\beta$ acts it initializes $\alpha$. So the correct collaboration of $\beta$ and $\alpha$ is preserved in this case.

2. $k = n$. Now $\alpha$ is visited infinitely often. There is a stage $s_0$ after which no procedure $< n$ acts and procedure $\langle n \rangle$ has a permanent witness not cancelled later. After this stage each time when $\alpha$ is visited it poses new restrictions on $M$ but when it is visited, values $\Sigma_\beta(A;m)$ for $m \geq n$ are destroyed due to a change of $A$ and the initialization of procedures $> n$ by $\alpha$ does no harm.

2. $k > n$. This case is clear since again when $\alpha$ is visited values $\Sigma_\beta(A;m)$ for $m \geq k$ are destroyed and $\alpha$ causes only a change of witnesses for procedures $m > k$.

So the collaboration of $\beta$ with below negative strategies is successful.

The global requirement $M \leq C$.

We should notice that the described modules are not sufficient to ensure the satisfaction of this requirement. Indeed, let $\beta$ be located on the true path and serve a requirement $P$. Procedure $\langle n \rangle$ of $\beta$ should replace an initial segment of $M$ by another after the change of $C(n)$. But if $n$ enters $C$, say
at stage $t$, $\beta$ receives attention at the next $\beta$-stage $s$ probably much larger than $t$ and there is no a real permitting of $M$ by $C$.

In order to insure the requirement we introduce a new modification to the construction. Namely, let some $n$ enter $C$ at a stage $s$. We say, strategy $\beta$ for a positive requirement $P$ requires attention at this stage if for all $x < n$ 

$$\Sigma_\beta(A^s; x) \downarrow= C^s(x), \Sigma_\beta(A^s; n) \downarrow= 0.$$

In other words, by the instructions for the isolated module, $\beta$ is ready to change $M$ at the moment. If there are several such $\beta$ we choose among them the strategy with the highest priority and allow it to act.

These actions are implemented outside of the true path but we will prove in the verification part that they do not crush the construction.

Now we begin the description of the whole construction. Tree of strategies $T_{str}$ consists of strings $\alpha$ such that for even arguments $2e < |\alpha| \alpha(2e)$ takes two possible values 0 or 1 denoting the outcomes for a negative strategy. Odd arguments $\alpha(2e + 1)$ take natural values and denote outcomes of the module for $P$. Each strategy $\alpha$ with even length $2e$ serves requirement $N_e$ and builds functionals $\Delta_\alpha$ and $\Gamma_\alpha$. Strategy $\beta$ of odd length $2e + 1$ serve $P_e$ and builds a functional $\Sigma_\beta^A$. In the construction we define also function $r_s(\alpha)$ which denotes (at the current stage $s$) the length of the restraint posed on $\alpha$ by strategies of higher priority. Let $<_L$ denote the lexicographical ordering of elements of $T_{str}$. Define initial values $r(\alpha) = |\alpha|$ for $\alpha \in T_{str}$ and consider stage $s + 1$ of the construction.

Instructions of stage $s + 1$ are divided into two parts. At the first substage we skip down beginning from the root of the tree along a node $f_s \in T_{str}$ with $|f_s| \leq s$. This $f_s$ is defined by induction at stage $s + 1$. At the second substage we make actions on a change of $M$.

**Definition.** Let $\alpha \in T_{str}$. Stage $t + 1$ is $\alpha$-stage if $\alpha \subseteq f_t$. Stage 0 is by definition $\alpha$-stage for every $\alpha \in T_{str}$.

**Substage I.**

Let consider $\alpha \in f_s$. Initialize first all nodes $\gamma >_L \alpha$. This means that all parameters defined by these $\gamma$ are cancelled and the implementation of the $\gamma$ begins anew. If $\alpha$ is empty define tree $T_\alpha^s$ equal to the set of all pairs $\langle \tau_0, \tau_1 \rangle$ such that $|\tau_0| = |\tau_1| \leq s$, $\tau_1$ is an initial segment of $M^s$, and strings $\tau_0$ and $\tau_1$ differ exactly on the last argument.
Instructions for \( \alpha \) of even length 2\( e \).

Carry out instructions of the first possible case for \( \alpha \).

**Case A1.** \( \Delta_\alpha(x) \downarrow \neq \Phi_e(M^e; x) \) for a (least) \( x \).
1. Define \( \tau_1 = M^e \upharpoonright s \) and \( \tau^e = M^r \upharpoonright r \) where \( r + 1 \) is the largest \( \alpha \)-stage less than \( s + 1 \) such that \( \Delta_\alpha \) is compatible with \( \Phi_e(M^r) \).
2. Let \( I(\alpha) = \{i_0 = e > i_1 > \ldots > i_k\} \) be the ordered set consisting of \( e \) and all even \( i < e \) such that \( \alpha(i) = 0 \).
   For each \( i \in I(\alpha) \) we define by induction a string \( \tau_i \) (\( \tau_e \) is already defined) such that \( \tau_i \subseteq \tau_{i+1} \). Assume \( \tau_j \) for some \( j \leq e \) is defined and let \( i < j \) be the largest element of \( I(\alpha) \) less than \( j \) (if any) and \( \beta = \alpha \upharpoonright i \). Define \( \tau_i \) to be a maximal by inclusion element of \( \text{range}(\Gamma_\beta) \) extending \( \tau_j \).
3. Cancel current \( \Delta_\alpha \) and begin the definition of a new \( \Delta_\alpha \).
4. Skip down to \( \alpha * \langle 0 \rangle \).

**Case A2.** \( s + 1 \) is an \( \alpha \)-expansionary stage, i.e. \( l_N(e, s) > l_N(e, t) \) for every \( \alpha \)-stage \( t + 1 \leq s \).
1. Define undefined yet values \( \Delta_\alpha(y) \) for \( y < l_N(e, s) \) equal to \( \Phi_e(M^e; x) \).
2. Skip down to \( \alpha * \langle 1 \rangle \).

**Case A3.** No of cases A1, A2 holds. Skip down to \( \alpha * \langle 1 \rangle \).

Define set of candidates of \( \alpha * \langle 1 \rangle \) equal to that of \( \alpha \) (accordingly, tree \( T_{\alpha*\langle 1 \rangle} \) is defined equal to \( T_\alpha \)).

Instructions for string \( \beta \) of odd length 2\( e + 1 \).

Let \( n = l_p(\beta, s) = \mu k[\Sigma^A_\beta(k) \neq C(k)][s] \). Initialize first procedures \( k > n \) of \( \beta \), then carry out instructions of the first possible case for procedure \( \langle n \rangle \):

**Case B1.** Witness of procedure \( \langle n \rangle \) is not chosen yet.

Subcase B1.1. There exists a candidate \( \langle \tau_0, \tau_1 \rangle \) of \( \alpha \) with splitting point
\[ z_n = \mu x[\tau_0(x) \neq \tau_1(x)] \] exceeding maximum of \( r(\beta) \) and \( s' \) where \( s' = \max\{t : l_P(\beta, t) = n-1\} \)

(this \( s' \) denotes the length of restraint on \( A \) posed by the procedures with numbers \( k < n \)).

Appoint \( \langle \tau_0, \tau_1 \rangle \) to be the current witness of \( \beta \). Initialize all \( \gamma \supseteq \beta \ast \langle n \rangle \) and redefine for them \( r(\gamma) = s \). Stop traversing \( T_{str} \) and go to substage II below.

Subcase B1.2. Subcase B1.1 does not hold.

Go to substage II.

**Case B2.** Witness \( \langle \tau_0, \tau_1 \rangle \) of procedure \( \langle n \rangle \) is defined, \( M \rvert (|\tau_0| + 1) = \Theta(A) \rvert (|\tau_0| + 1) \) with use \( \theta(|\tau_0|) < a(s) \), where \( a(s) = \mu a[a \in A^* - A^t] \), \( t + 1 \leq s \) is the last \( \beta \)-stage, and \( \Sigma_{\beta}(A; n) \uparrow \).

Subcase B2.1. \( n \in C^* \). Define \( \Sigma_{\beta}(A; n) = 1 \) with use \( \sigma(n) = \sigma(n - 1) \) (we assume \( \sigma(-1) = 0 \) by definition).

Subcase B2.2. \( n \notin C^* \). Define \( \Sigma_{\beta}(A; n) = 0 \) with use \( \sigma(n) = \theta_e(|\tau_0|) \).

In both subcases skip down to \( \beta \ast \langle n \rangle \).

**Case B3.** In all other cases skip down directly to \( \beta \ast \langle n \rangle \).

This completes instructions of the first substage.

**Substage II.**

**Definition.** Let \( n(s) = \min\{n \in C^{s+1} - C^* \} \lor ((\exists t < s) n \in C^{t+1} - C^t \land a_{s+1} < a_{r+1} \text{ for all } r, t \leq r < s\} \).

String \( \beta \in T_{str} \) of odd length \( 2e + 1 \) is called \( P \)-correct at stage \( s + 1 \) if

1. For any \( \gamma \subset \beta \) of odd length \( l_P(\gamma, s) = \beta(|\gamma|) \),
2. \( l_P(\beta, s) = n(s) \),
3. Procedure \( \langle n(s) \rangle \) of \( \beta \) has an active witness \( \langle \tau_0, \tau_1 \rangle \),
4. \( \Theta_e(A) \rvert (|\tau_0| + 1) = M \rvert (|\tau_0| + 1) \).

Informally, this condition means that first, the guess of \( \beta \) on positive \( \gamma \subset \beta \) seems valid at the considered stage, and second, procedure \( \langle n(s) \rangle \) of
β is ready to replace initial segment of M due to item 4 of the main module for a positive P.

If such β exist carry out the following instructions:

1. Among all of P-correct β choose the string of the highest priority. Let \( \langle \tau_0, \tau_1 \rangle \) be the current witness of procedure \( \langle n(s) \rangle \) of the chosen β.
2. Replace initial segment of M of length \(|\tau_0|\) by \( \tau_0 \).
3. Initialize all \( \gamma \supset \beta \ast \langle n(s) \rangle \) and \( \gamma > L \beta \ast \langle n(s) \rangle \). Redefine \( r(\gamma) = s \) for all initialized \( \gamma \).

This completes stage \( s + 1 \).

Verification.

Let \( f \) be the true path defined as the leftmost path of \( T_{str} \) visited infinitely often during the construction by the instructions of substage I.

First we prove that the procedure of the definition of the set \( \{ \tau^i, i \in I(\alpha) \} \), in the description of case B, substage I, is correct.

Indeed, by the construction, every \( \alpha \)-stage is also \( \beta \ast \langle 0 \rangle \)-stage for every \( \beta \subset \alpha \) with even \(|\beta|\) and \( \gamma(|\beta|) = 0 \) so at each \( \alpha \)-stage functional \( \Gamma_{\beta} \) is extended and string \( M^t \upharpoonright t \) is added to \( \text{range}(\Gamma_{\beta}) \) for every such \( \beta \).

The first element \( \tau^e \) of \( I(\alpha) \) is defined equal to \( M^t \upharpoonright t \) for some \( \alpha \)-stage \( t + 1 < s + 1 \). Let \( i \) be the largest element of \( I(\alpha) \) less \( \tau^e , \beta = \alpha \upharpoonright i \). By the remark above, string \( \tau^e \) is added at stage \( t + 1 \) to the range of \( \Gamma_{\beta} \) so \( \tau^e \) is contained in the range of \( \Gamma_{\beta} \) and \( \tau^e \) is a substring of a maximal string \( \tau^1 \in \text{range}(\Gamma_{\beta}) \).

We note that \( \tau^i \) is searched at stage \( s + 1 > t + 1 \) when \( \Gamma_{\beta} \) was probably extended and \( \tau^i \) can be a proper extension of \( \tau^e \).

The induction on elements of \( I(\alpha) \) completes the proof of the assertion.

It implies that both arguments \( \tau_0, \tau_1 \) of each candidate of \( \alpha \) extend at the moment of definition maximal by inclusion elements of each \( T_{\beta} \) with \( \beta \subset \alpha \), \(|\beta|\) is even, \( \alpha(|\beta|) = 0 \). This insures that positive strategies below \( \alpha \) changing \( M \) do not damage functionals \( \Gamma_{\beta} \) for even \( \beta \subset \alpha \).

Lemma 1 For every \( \alpha \in f \) there is a stage after which no \( \beta <_L \alpha \) is visited.

Proof. Indeed, \( \beta <_L \alpha \) can be visited at a stage \( s \) either by the instructions of the first substage, or by the instructions of the second substage. By definition of the true path there is a stage after which no \( \beta <_L \alpha \) is visited by the instructions of the first substage.
As for instructions of the second substage, \( \beta \) can be chosen at a stage 
\( s \) only in the case if this \( \beta \) was visited earlier, at a stage \( t < s \). Besides, 
each \( \beta \) being not initialized can be visited no more than once during the 
construction by the instructions of the second substage. This establishes the 
lemma.

**Lemma 2 (Outcome Lemma).** Every condition \( N_e, P_e, e \in \omega \), is satisfied 
by strategies \( \alpha \) and \( \beta \) of length \( 2e \) and \( 2e + 1 \) located in \( \mathbf{f} \).

**Proof.**

1. Let \( \alpha \subset \mathbf{f} \) have even length \( 2e \). We accept the following inductive 
assumptions for \( \alpha \):

- **C1.** There exists a stage \( s_0 \) after which no \( \beta \subset \alpha \) initializes \( \alpha \),
- **C2.** At each \( \alpha \)-stage a new candidate \( \tau_0, \tau_1 \) is given to \( \alpha \).

Let \( s_1 \geq s_0 \) denote the least stage after which no \( \beta \) to the left of \( \alpha \) is 
visited in the construction by instructions of stage I. Since each visit of a 
string \( \beta \in T_{str} \) by instructions of stage II answers to its previous visit by 
instructions of substage I and this correspondence is one to many so there is 
a stage \( s_\alpha \geq s_1 \) after which \( \alpha \) is not initialized by the strategies located to 
the left or above.

We note that after stage \( s_\alpha \) \( \alpha \) works at \( \alpha \)-stages as in isolation. Indeed, by 
the choice of \( s_\alpha \) no \( \beta \) to the left or above \( \alpha \) can act initializing \( \alpha \). Strategies 
located below \( \alpha \ast \langle 1 \rangle \) are initialized each time when \( \Gamma_{\alpha} \) is extended and their 
restraint is increased. So they cannot damage the correct definition of \( \Gamma_{\alpha} \) 
and \( \Delta_{\alpha} \). As for strategies extending \( \alpha \ast \langle 0 \rangle \) they work under restriction on 
\( M \) to extend a maximal node of \( \Gamma_{\alpha} \). In the assertion preceding the lemma 
we checked that the first branch of each new candidate defined by these 
strategies is a maximal by inclusion node in \( \Gamma_{\alpha} \); therefore, \( M \) being changed 
by \( \gamma \supseteq \alpha \ast \langle 0 \rangle \) continues to extend a maximal node of \( \Gamma_{\alpha} \). The latter insures 
the success of \( \alpha \). We can conclude that \( \alpha \) satisfies \( N_e \).

If \( \alpha \) has outcome \( 0 \) then \( \alpha \) never initializes strategies \( \gamma \supseteq \alpha \ast \langle 0 \rangle \) and each 
time when we skip to \( \alpha \ast \langle 0 \rangle \), a new candidate is formed so both assumptions 
C1 and C2 hold for \( \alpha \ast \langle 0 \rangle \).

If \( \alpha \) has outcome \( 1 \) then there is a stage \( s' \geq s_\alpha \) after which \( \alpha \ast \langle 0 \rangle \) is 
not visited and strategies below \( \alpha \ast \langle 1 \rangle \) are not initialized by \( \gamma \supseteq \alpha \). We skip 
down to \( \alpha \ast \langle 1 \rangle \) at all \( \alpha \)-stages \( s \geq s' \) and each candidate of \( \alpha \) becomes a 
candidate of \( \alpha \ast \langle 1 \rangle \). Therefore, both assumptions C1 and C2 hold for \( \alpha \ast \langle 1 \rangle \).

2. Consider now a positive \( \beta \subset \mathbf{f}, |\beta| = 2e + 1 \). We assume that:
C1. There exists a stage $s_0$ after which no $\gamma \subset \beta$ initializes $\beta$, 

C2. At each $\beta$-stage a new candidate $\tau_0, \tau_1$ is given to $\beta$.

As in the previous part, there exists a stage $s_\beta \geq s_0$ after which $\beta$ is not initialized by the strategies to the left and above. We analyze the work of $\beta$ after stage $s_\beta$ and shall prove the following:

1. Each procedure $\langle n \rangle$ of $\beta$ being started and not initialized by procedures with smaller numbers eventually receives its witness, say $x_n$.

2. Each procedure $\langle n \rangle$ of $\beta$ being not initialized by procedures with smaller numbers either satisfy $P_e$ by setting the disagreement

$$M \upharpoonright (|x_n|+1) \neq \Theta(A) \upharpoonright (|x_n|+1) \quad (*)$$

or determine the agreement

$$C(n) = \Sigma_\beta(A; n) \quad (**$$

allowing to act procedure $\langle n + 1 \rangle$.

3. If procedure $\langle n \rangle$ is initialized by procedures with smaller numbers then

($*$) holds.

The assertions 1-3 imply that $P_e$ is satisfied since, otherwise, $C = \Sigma(A)$ contrary to $C \not\leq_T A$. Let $n = \mu z[C(n) \neq \Sigma(A; n)]$.

4. There are infinitely many $\beta*\langle n \rangle$-stages.

5. Assumptions C1-C2 hold for $\beta*\langle n \rangle$.

Proof of assertions 1-5.

1. Assume procedure $\langle n \rangle$ is started and is not initialized by procedures with smaller numbers. Each time when $\beta$ is visited it receives a new candidate $\langle \tau_0, \tau_1 \rangle$ due to assumption C2.

If a current candidate does not satisfy to condition of case B1.1 we go directly to substage II delaying the construction till a moment when such a candidate appears. Since the restraint on the witness does not grow we eventually will receive a required candidate.

Notice that from that moment procedure $\langle n \rangle$ is never initialized and the chosen witness never changes.

2. If $l_P(\beta, s)$ grows then eventually case B2 will occur and defining $\Sigma_\beta(A; n) = C(n)$ we determine condition (**).
Assume that at a later stage (**) becomes invalid and let \( n \) be the least number for which this occurs. If \( n \) was in \( C \) at the moment of definition of \( \Sigma(A; n) \) then \( \sigma(n - 1) = \sigma(n) \) and a damage of (**) for procedure \( \langle n \rangle \) means the same for procedure \( \langle n - 1 \rangle \). So we can exclude this case from consideration. Consider two possible cases:

Case 1. \( A \upharpoonright \sigma(n) \) has changed.

In this case disagreement (*) becomes valid and assertion 2 continues to hold. If later \( M \upharpoonright (|x_n| + 1) = \Theta_c(A) \upharpoonright (|x_n| + 1) \) with new \( \theta(|x_n|) \) then we come again to case B2 of the construction and are able to redefine \( \Sigma(\beta; A; n) \) equal to \( C(n) \).

Case 2. \( C(n) \) has changed.

Let \( t \geq s \) be the least \( A \)-true stage. At a stage \( r, s + 1 \leq r \leq t + 1 \) string \( \beta \) is P-correct and due to the inductive assumptions has the greatest priority among all \( P \)-correct at this stage strings. Parameter \( n(s) \) computed at this stage is equal to \( n \) due to the assumption of minimality of \( n \) and procedure \( \langle n \rangle \) acts by the instructions of substage II changing \( M \) and restoring (*).

Thus, assertion 2 holds.

3. Assume now that procedure \( \langle k \rangle \) is initialized when \( l_P(\beta, s) \) drops its value to \( n < k \). This can occur as mentioned in the previous item either due to the change of \( C(n) \), or to a change of \( A \upharpoonright \sigma(n) \). In the first case we change \( M \upharpoonright (|x_n + 1|) \) forcing (*) and while the latter holds we do not need to correct \( \Sigma_\beta \). If later \( M \upharpoonright (|x_n| + 1) = \Theta_c(A) \upharpoonright (|x_n| + 1) \) then \( A \upharpoonright \sigma(n) \) changes and \( \Sigma(A; k) \uparrow \) so we can redefine it for a new witness of procedure \( \langle \rangle \).

If \( C(n) \) preserves and only \( A \upharpoonright \sigma(n) \) changes then again \( \Sigma(A; k) \) becomes undefined and again can be redefined.

4. Let \( n = \lim \inf_A l_P(\beta, s) \). If \( \Sigma_\beta(A; n) \downarrow \) then there is a stage after which \( l_P(\beta, s) \) is equal permanently to \( n \) and we directly skip down to \( \beta \ast \langle n \rangle \) by the instruction of case B3.

If \( \Sigma_\beta(A; n) \uparrow \) then it becomes undefined at \( A \)-true stages and at each \( \beta \) next to \( A \)-true stage \( l_P(\beta, s) \) drops its value to \( n \) then we skip down to \( \beta \ast \langle n \rangle \) by the instruction of case B2.

5. There is a stage after which procedures with numbers \( k < n \) do not act and not pose new restraints. So the assumption C1 holds for \( \beta \ast \langle n \rangle \).
We skip down to $\beta \ast \langle n \rangle$ only in cases B2 and B3 and in each of these cases no candidate is taken by $\beta$ so each candidate of $\beta$ becomes a candidate of $\beta \ast \langle n \rangle$. This proves assumption C2 for $\beta \ast \langle n \rangle$.

This establishes the lemma.

**Lemma 3** $M \leq_T C$.

Proof. By the construction, each change of $M(x)$ is preceded by the change of $C(n)$ for a $n < x$ and occurs no later than before the next $A$-true stage following the stage when $C(n)$ changed.

**Lemma 4** $M$ is $\omega$-c.e.

Proof. A value $M(x)$ can be changed no more than once by strategies $\beta$ with $|\beta| < x$ and there is a finite number of such strategies.

This establishes the theorem.

**Corollary.** In any cone $D(<c$ c.e.) degree $0$ is branching in $\omega$-c.e. degrees.

Proof. Take degrees $b_1$ and $b_2$ equal to a c.e. degree $a$, $0 < a < c$, and the minimal degree $m$ built in the theorem for the considered $a$ and $c$.

**References**


