

RESTRICTED JUMP INTERPOLATION IN THE D.C.E. DEGREES

Carl G. Jockusch, Jr. and Angsheng Li*

July 29, 2009

Abstract

It is shown that for any 2-computably enumerable Turing degree l , any computably enumerable degree a , and any Turing degree s , if $l' = \mathbf{0}'$, $l < a$, $s \geq \mathbf{0}'$, and s is c.e. in a , then there is a 2-computably enumerable degree x with the following properties:

- (1) $l < x < a$, and
- (2) $x' = s$.

1 Introduction

The topic of inverting the Turing Jump in the Turing degrees started in the 1950's. Following the pioneering work of Kleene and Post (1954), Friedberg (1957) provided a completeness criterion for the Turing degrees: Given a Turing degree b , $b \geq \mathbf{0}'$ if and only if there is a degree a such that $a' = a \vee \mathbf{0}' = b$. This result has played a fundamental role in the theory of the global Turing structure. For Turing degrees s and d , say that s is CEA in d if s is c.e. in d and $s \geq d$. The first "local" result on jump inversion is due to Shoenfield (1959), who showed that for every degree s CEA in $\mathbf{0}'$, there is a degree $x \leq \mathbf{0}'$ such that $x' = s$.

Subsequently, Sacks (1963) improved this result by proving the Sacks Jump Theorem, which asserts that given degrees s and c with s CEA in $\mathbf{0}'$, and $c \neq \mathbf{0}$, then there exists a nonzero c.e. degree x such that $x' = s$ and $c \not\leq x$. It follows from this result that for every degree $a \leq \mathbf{0}'$ there is a c.e. degree x such that $x' = a'$.

The substructure of the c.e. Turing degrees is particularly interesting because it is completely different from the global structure of the Turing degrees. For instance, the c.e. degrees are dense by the Sacks density theorem (Sacks (1964)), while there are minimal Turing degrees by Spector (1956) and Sacks (1961).

*The second author is partially supported by National Distinguished Young Investigator Award No. 60325206 (P. R. CHINA). Both authors are partially supported by the NSFC Grand International Joint Project, New Directions in Theory and Applications of Models of Computation, No. 60310213 (P. R. CHINA). Corresponding author: angsheng@gcl.iscas.ac.cn, Angsheng LI

Furthermore, Robinson (1971) showed that Sacks' Density Theorem and his Jump Theorem can be combined: (Robinson's Interpolation Theorem) Given c.e. degrees $\mathbf{b} < \mathbf{c}$, and a degree \mathbf{s} , if $\mathbf{s} \geq \mathbf{b}'$, and \mathbf{s} is c.e. in \mathbf{c} , then there is a c.e. degree \mathbf{x} such that $\mathbf{b} < \mathbf{x} < \mathbf{c}$ and $\mathbf{x}' = \mathbf{s}$. (See Soare (1987), page 146 for a simplified proof.) Robinson's Interpolation Theorem has played a major role in the study of the c.e. degrees.

It is natural to ask whether there exist analogues of the Robinson Interpolation Theorem for degree structures other than the c.e. degrees. However, few examples of such structures are known which are dense, so it seems necessary to consider results which have a weaker form. For example, one may ask whether for every degree $\mathbf{a} \leq \mathbf{0}'$ and every degree $\mathbf{s} \geq \mathbf{0}'$ with \mathbf{s} c.e. in \mathbf{a} there is a degree $\mathbf{x} \leq \mathbf{a}$ such that $\mathbf{x}' = \mathbf{s}$. Here a negative answer follows from known results: S. B. Cooper, R. Epstein, and L. Sasso (see Sasso (1974)) showed that there is a minimal degree $\mathbf{a} < \mathbf{0}'$ with $\mathbf{a}' > \mathbf{0}'$, and then by the Sacks Density Theorem relative to \mathbf{a} there is a degree \mathbf{s} such that $\mathbf{0}' < \mathbf{s} < \mathbf{a}'$ and \mathbf{s} is c.e. in \mathbf{a} . Clearly, no degree $\mathbf{x} \leq \mathbf{a}$ satisfies $\mathbf{x}' = \mathbf{s}$. Thus, jump interpolation fails in the degrees $\leq \mathbf{0}'$ even for intervals $[\mathbf{b}, \mathbf{c}]$ with $\mathbf{b} = \mathbf{0}$.

It seems reasonable to give up on jump interpolation for the degrees $\leq \mathbf{0}'$ and instead concentrate on structures which are "closer" to the c.e. degrees. In this paper, we consider a jump interpolation theorem for the d.c.e. degrees.

A set A is called d.c.e. if it has the form $U - V$, where U and V are c.e. Of course, A is d.c.e. if and only if A is 2-c.e., where the n -c.e. sets are defined as follows:

We say that A is n -computably enumerable (n -c.e., for short), where n is a natural number, if there is a computable function f such that for all x ,

- (a) $f(x, 0) = 0$,
- (b) $\lim_s f(x, s) \downarrow = A(x)$, and
- (c) $|\{s \mid f(x, s) \neq f(x, s + 1)\}| \leq n$.

This hierarchy was extended by Ershov (1968) into the transfinite so as to include all Δ_2^0 sets, but in this paper we consider only 2-c.e. sets.

It was observed by Lachlan that every nonzero d.c.e. degree bounds a nonzero c.e. degree, and thus no d.c.e. degree is minimal in the d.c.e. degrees. This removes one obstacle to proving some version of the Robinson Interpolation Theorem for the d.c.e. degrees.

However, it was shown by Cooper, Harrington, Lachlan, Lempp and Soare (1991), that there is a d.c.e. degree $\mathbf{b} < \mathbf{0}'$ such that no d.c.e. degree \mathbf{c} satisfies $\mathbf{b} < \mathbf{c} < \mathbf{0}'$. Thus the full analogue of the Robinson Interpolation Theorem fails for the d.c.e. degrees (even for intervals with top element $\mathbf{0}'$), and we instead consider weaker versions. One possible version to consider for intervals with bottom element $\mathbf{0}$ is the following:

(*) Let \mathbf{a} be a d.c.e. degree, and suppose that \mathbf{s} is a degree such that $\mathbf{s} \geq \mathbf{0}'$ and \mathbf{s} is c.e. in \mathbf{a} .

Then there is a d.c.e. degree \mathbf{x} such that $\mathbf{x} \leq \mathbf{a}$ and $\mathbf{x}' = \mathbf{s}$.

It is an open question whether the statement (*) above holds. In fact we conjecture that (*) is false and will mention briefly here an obstacle to proving (*). This obstacle will suggest proving a different result, which will in fact be our main theorem. We assume that readers are familiar with the proof of previous jump theorems, such as the Sacks Jump Theorem. Suppose that A is a given d.c.e. set of degree \mathbf{a} , and we are constructing a d.c.e. set $X \leq_T A$ such that $X' \equiv_T S$, where S is a set of degree \mathbf{s} which is c.e. in A . Suppose

that our highest priority requirement is that S can compute whether $0 \in X'$. The standard strategy to meet such a requirement is to use restraint on X to preserve computations showing $0 \in X'[s]$, for any stage s .

Then, because this requirement has highest priority, we can argue that $0 \in X'$ if and only if there exists a stage s with $0 \in X'[s]$, so that $\mathbf{0}'$, and thus S , can compute whether $0 \in X'$. However, to ensure that $X \leq_T A$, we will also need to define a Turing functional Φ with $\Phi(A) = X$, and the axioms for Φ may require X to return to a previous neighborhood when A does so. Thus, changes in A can cause changes in X which injure our requirement of highest priority.

One would now seem to need an oracle for A' (which is not available), to compute whether $0 \in X'$. Note that this difficulty arises even if we require X to be merely Δ_2^0 rather than d.c.e., and in fact we conjecture that $(*)$ is false even if the condition that \mathbf{x} be d.c.e. is omitted.

The difficulty in the previous paragraph arises basically because we are constructing a set which is computable relative to a d.c.e. set A . If we assume that A is instead c.e., it cannot return to a previous neighborhood, and the problem no longer arises.

This in itself is not of interest because then we are dealing only with a special case of the Robinson Interpolation Theorem. However, it does suggest that we may be able to achieve jump interpolation in intervals (\mathbf{l}, \mathbf{a}) , where \mathbf{l} is a low d.c.e. degree, $\mathbf{l} < \mathbf{a}$, and \mathbf{a} is c.e.

In fact, that is what we do in our main theorem, which we now state.

THEOREM. Given a d.c.e. degree \mathbf{l} , a c.e. degree \mathbf{a} , and a Turing degree \mathbf{s} , if $\mathbf{l}' = \mathbf{0}'$, $\mathbf{l} < \mathbf{a}$, $\mathbf{s} \geq \mathbf{0}'$, and \mathbf{s} is c.e. in \mathbf{a} , then there exists a d.c.e. degree \mathbf{x} such that $\mathbf{l} < \mathbf{x} < \mathbf{a}$, and $\mathbf{x}' = \mathbf{s}$.

Before proceeding to the proof, we mention some additional open questions about alternate forms of the result.

Open Question. Is the following true? Given a d.c.e. degree \mathbf{l} , a c.e. degree \mathbf{a} , and a Turing degree \mathbf{s} , if $\mathbf{l} < \mathbf{a}$, $\mathbf{s} \geq \mathbf{0}'$, and \mathbf{s} is c.e. in \mathbf{a} , then there exists a d.c.e. degree \mathbf{x} such that $\mathbf{l} \leq \mathbf{x} \leq \mathbf{a}$, and $\mathbf{x}' = \mathbf{s}$. (Here we have dropped the hypothesis that \mathbf{l} is low from the above theorem and replaced the interval (\mathbf{l}, \mathbf{a}) by the interval $[\mathbf{l}, \mathbf{a}]$.)

A negative answer to the above question would follow if it were shown that the result of Cooper, Harrington, Lachlan, Lempp and Soare (1991) could be improved to show that there is a nonhigh d.c.e. degree \mathbf{a} such that no d.c.e. degree lies strictly between \mathbf{a} and $\mathbf{0}'$.

Open Question. Is the following true? Given a d.c.e. degree \mathbf{l} , a d.c.e. degree \mathbf{a} , and a Turing degree \mathbf{s} , if $\mathbf{l} < \mathbf{a}$, $\mathbf{s} \geq \mathbf{0}'$, and \mathbf{s} is c.e. in \mathbf{a} , then there exists a d.c.e. degree \mathbf{x} such that $\mathbf{l} \leq \mathbf{x} \leq \mathbf{a}$, and $\mathbf{x}' = \mathbf{s}$. (This is the same as the previous open question, but the degree \mathbf{a} is now assumed to be merely d.c.e. instead of c.e. A positive answer seems to be even more unlikely. This question is also open if \mathbf{l} is again required to be low.)

The following fact will simplify the proof of our main result.

By Arslanov, Cooper and Li [2004], for any d.c.e. degree \mathbf{l} and any c.e. degree \mathbf{a} , if \mathbf{l} is low, and $\mathbf{l} < \mathbf{a}$, then there is a low d.c.e. degree \mathbf{x} such that $\mathbf{l} < \mathbf{x} < \mathbf{a}$. So in order to prove the theorem, we need only to satisfy $\mathbf{l} \leq \mathbf{x} < \mathbf{a}$, and $\mathbf{x}' = \mathbf{s}$.

We now turn to the proof of our main result. We organise the proof as follows. In section 2, we formulate the conditions of the theorem as requirements,

and describe the ideas and strategies to satisfy the requirements; in section 3, we describe a full construction to build the objects; and in section 4, we verify that the construction satisfies the requirements as desired.

Our notation and terminology are standard, and generally follow those of Soare [1987].

2 The Requirements and the Strategies

First we formulate the conditions of the theorem as requirements.

2.1 The Requirements

Let L be a d.c.e. set, A be a c.e. set, and S be a set c.e. in A via some c.e. set W . Suppose that $L' \leq_T K$, and $L <_T A$, where K is a fixed creative set. To prove the theorem, we construct a d.c.e. set X to satisfy the following requirements:

$$\begin{aligned} \mathcal{T}: & X \leq_T A \oplus L \\ \mathcal{N}_e: & A = \Psi_e(X, L) \rightarrow A \leq_T L \\ \mathcal{R}_e: & W^A(e) = \lim_y \Gamma(X, L; e, y) \\ \mathcal{S}_e: & (X \oplus L)'(e) = \Omega(C, W^A; e) \end{aligned}$$

where $e \in \omega$, $(X \oplus L)' = \{e \mid \Phi_e(X, L; e) \downarrow\}$, $L' = \{e \mid \Phi_e(L; e) \downarrow\}$, $\{(\Psi_e, \Phi_e) \mid e \in \omega\}$ is an effective enumeration of all pairs (Ψ, Φ) of Turing functionals Ψ, Φ ; C is an auxiliary c.e. set built by us; Ω is a Turing functional built by us such that $\Omega(C, W^A)$ is total; Γ is a Turing functional built by our construction such that for all e, y , $\Gamma(X, L; e, y)$ converges, and for all e , $\lim_y \Gamma(X, L; e, y)$ exists. Then the degree \mathbf{x} of $X \oplus L$ has the properties asserted in the theorem.

2.2 Overview of the Proof

In this subsection, we give an informal description of the ideas, the strategies and the proof of the theorem. To satisfy the \mathcal{T} -requirement, we need a 2-c.e. *permitting method*, which will be described as a c.e. set of axioms of the form (σ, τ, x, y) . The formal definition of the 2-c.e. permitting method will be given in subsection 2.4, but the basic idea is as follows.

A number x may enter X only when it is permitted to do so by entry of a sufficiently small number into A . A number x may be extracted from X only when this action is permitted by the exit of a sufficiently small number from L . To satisfy an \mathcal{N} -requirement, $\Psi(X, L) \neq A$, we will preserve computations $\Psi(X, L; x)$ for some x by imposing an X -restraint r^X so that if $\Psi(X, L) = A$, then $\Psi(X, L)$ is computable in L . We will build a Turing functional Λ say, to simulate the computations $\Psi(X, L; x)$ which have been preserved. However we can preserve a computation $\Psi(X, L; x)[v]$ for some v only if we have passed a test of the L -correctness by a lowness approximation procedure given in subsection 2.3. For a fixed x , we start a cycle w say, to test the L -correctness below some *configuration c say*, which is greater than the use $\psi(X, L; x)$ of the computation $\Psi(X, L; x)[v]$ for some stage v . For a fixed x , we ensure that if the configuration $c[s]$ associated with $\Psi(X, L; x)[s]$ is unbounded, then so is $\psi(X, L; x)[s]$. During the testing, if there is an $l \leq c$ which leaves L since stage v , then the test finishes with no progress, otherwise, and there is an $l \leq c$ which has entered L since

stage v , then we impose a *conditional restraint* (c, r) say, for $r = \psi(X, L; x)[v]$. To explicitly specify that a conditional restraint (c, r) is defined at stage v , say, we also use (σ, r) for $\sigma = L_v \upharpoonright (c + 1)$ to denote the conditional restraint (c, r) . (The notion of the conditional restraint was first used by Cooper and Li in [ta] to deal with a complex permitting markers system). The role of the configuration c is to guarantee that if there is no $l \leq c$ which has left L since stage v , then there is no $y \leq \psi(X, L; x)[v]$ which leaves X after stage v . Therefore if we reach a stage $s > v$ at which we have that for any $l \leq c$, $L_v(l) = L_s(l)$, then the conditional restraint (c, r) ensures that for any such y , the L -permitting marker of y is greater than c , so that we can extract all the elements $y \leq r$ which have been enumerated into X since stage v from X , and now impose a *restraint* to be r , which is called an *absolute restraint*. The point is that the absolute restraint r will be preserved forever, unless some $l \leq c$ with $l \in L_v$ leaves L . This means that we have to preserve the history of the simulation, because an L -enumeration may be proven false.

To satisfy an \mathcal{R}_e -requirement for some e , we define a Turing functional Γ and define for all y , $\Gamma(X, L; e, y)$. Γ will be built as a c.e. set of axioms of the form (B, σ, e, y, z) , where B is an interval $(u, 2u]$ of natural numbers for some u . For an axiom (B, σ, e, y, z) with $|\sigma| = |B|$, we say that B is the *use block* of the axiom, and that σ is the L -use of the axiom.

We now define $\Gamma(X, L; e, y) = z$ if and only if there is an axiom $(B, \sigma, e, y, z) \in \Gamma$ such that $B \cap X = \emptyset$, and $\sigma \subset L$.

Γ will have the following properties: Let (B, σ, e, y, z) be an axiom enumerated into Γ at stage v , and let s be a stage $> v$, then:

If $B \cap X_s \neq \emptyset$, then we say that the axiom (B, σ, e, y, z) is *invalid at stage* s , else *valid at stage* s .

Whenever we define $\Gamma(X, L; e, y)$, we define the use block of this computation to be fresh only if there is no previous computation $\Gamma(X, L; e, y)[s']$ with valid use block B for any s' , and while if there is a valid use block associated with some previous computation $\Gamma(X, L; e, y)[s']$ for some s' , we define the use block as the same as before. This will guarantee that L -changes will never make Γ non-total, so that for every y , $\Gamma(X, L; e, y)$ will be eventually defined.

Whenever we enumerate an axiom (B, σ, e, y, z) into Γ , we always have that $|\sigma| = |B|$. So whenever we define $\Gamma(X, L; e, y)$, we enumerate an axiom (B, σ, e, y, z) into Γ , which is defined as follows: at stage s ,

Case 1. If there is an axiom $(B', \sigma', e, y, z') \in \Gamma$ which is valid at the current stage, then for the most recently enumerated axiom (B', σ', e, y, z') , let $B = B'$, $\sigma = L \upharpoonright (|B| + 1)$, and enumerate (B, σ, e, y, z) into Γ , where z is the current value of $W^A(e)$.

We notice that $\sigma' \not\subseteq L_s$. If there is an l such that $\sigma'(l) > L_s(l)$, then the axiom (B', σ', e, y, z') cannot be the final axiom of $\Gamma(X, L; e, y)$. If there is an l such that $\sigma'(l) < L_s(l)$, then if $\sigma' \subset L$ holds eventually, then $\sigma \not\subseteq L$ holds permanently. So the definition in Case 1 above will never give different values to $\Gamma(X, L; e, y)$ for fixed e, y .

Case 2. Otherwise. Then let u be the least natural number which is greater than any number mentioned so far, let $B = \{u + 1, u + 2, \dots, 2u\}$, and $\sigma = L \upharpoonright (u + 1)$, and z be the current approximation of $W^A(e)$, and enumerate (B, σ, e, y, z) into Γ . In this case, we say that the use block B is defined *afresh*.

For the given set W relative to A , we use $w^A(e)$ to denote the use of computation for $e \in W^A$. We assume that the enumeration of W^A satisfies the usual

hat trick. That is if $e \in W^A[s]$, and A changes below $w^A(e)[s]$ at stage $s + 1$, then e leaves W^A automatically. This ensures that $e \in W^A$ if and only if for almost every s , $e \in W^A[s]$. We define the *possible outcomes* of an \mathcal{R}_e -strategy to be 0 and 1 with $0 <_L 1$, where 0 means that $e \notin W^A$, and 1 means that $e \in W^A$.

We analyse some typical problems of the \mathcal{R} -strategies.

Suppose that α is an \mathcal{R}_e -strategy for some e . Then α will locate its computation $\Gamma(X, L; e, y)$ and associated use block B at $\alpha \hat{\langle} i \rangle$, if i is the current value of $W^A(e)$ approximated by α . Suppose that a use block B with $\Gamma(X, L; e, y)[s_0] = 1$ for some s_0 is defined and then located at $\alpha \hat{\langle} 1 \rangle$. Then we define the A -permitting marker of elements in B to be the current use $w^A(e)$. Then we wait for an A -change below $w^A(e)$. If such an A -change occurs at a stage $s_1 > s_0$, then we enumerate the least unused element of B into X to invalidate the computation $\Gamma(X, L; e, y)$. Suppose that at a stage $s_2 > s_1$, α defines $\Gamma(X, L; e, y)$ for the same y but located at $\alpha \hat{\langle} 0 \rangle$, so that $\Gamma(X, L; e, y)[s_2] = 0$. However if there is a stage $s_3 > s_2$ say some strategy extracts element of $B \cap X$ from X , then we have both axioms $\Gamma(X, L; e, y)[s_0] = 1$ and $\Gamma(X, L; e, y)[s_2] = 0$. The tree construction will ensure that in this case, there is an \mathcal{N} - or an \mathcal{S} -strategy $\beta \subset \alpha$ such that

- (i) The enumeration of $b \in B$ into X at stage s_1 is injuring some conditional restraint (c, r) of β which has been defined before stage s_1 ,
- (ii) The use block B' of $\Gamma(X, L; e, y)[s_2]$ will be greater than c for any c in (i) above.

Therefore by stage s_3 , there is an $l \leq |B'|$ such that $l \in L_{s_2} \setminus L_{s_3}$, so that the axiom for $\Gamma(X, L; e, y)[s_2]$ proved permanently false.

We notice that α enumerates infinitely many elements into X only if α has true outcome 0. The reason why α does not injure a node $\beta \supseteq \alpha \hat{\langle} 0 \rangle$ is that:

- (a) α will not enumerate elements of use blocks located at $\alpha \hat{\langle} 0 \rangle$, and
- (b) At the stages at which β is visited, all use blocks located at $\alpha \hat{\langle} 1 \rangle$ have already been destroyed. That is to say, any use block B located at $\alpha \hat{\langle} 1 \rangle$ has an element in X before β is visited.

This idea is essential for us to design \mathcal{N} - and \mathcal{S} -strategies below \mathcal{R} -strategies.

All \mathcal{N} - and \mathcal{S} -strategies have a unique *outcome*, denoted by 1. Therefore the *true path* TP of the construction will be computable in $K \oplus W^A$, which is essential to the satisfaction of the \mathcal{S} -requirements. If $A \not\leq_T L$, then the true path TP exists, which entails the satisfaction of the requirements. The point is that, the function f assumed by the recursion theorem in subsection 2.3 is irrelevant to the assumption that $A \leq_T L$, so that it does exist.

An \mathcal{S}_e -strategy β say for some e , will define $\Omega(C, W^A; e)$ by enumerating some axiom of the form (c, β, e, i) into Ω , where c is a fresh number, and i is the current value of $(X \oplus L)'(e)$. Similarly to an \mathcal{N} -strategy, we will start a cycle of L -testing to approximate the L -correctness of a computation $\Phi_e(X, L; e) \downarrow$.

Based on this intuition, we design our strategies.

2.3 The Lowness Approximation

Since $L' \leq_T K$, we can choose a Turing functional Θ such that $\Theta(K)$ is total, and $L' = \Theta(K)$. To approximate the lowness condition of L , we build an auxiliary Turing functional Δ . Suppose by the recursion theorem with parameters that

f is a computable function given in advance of the construction such that for all x, y , the following property holds,

$$\Delta(L; x, y) = \Phi_{f(x)}(L; y).$$

The role of Δ is to open some cycles of the L -correctness testing of computations access to oracle L .

2.4 The \mathcal{T} -Strategy

To satisfy the \mathcal{T} -requirement $X \leq_T A \oplus L$, we build a Turing functional Ξ such that $\Xi(A, L) = X$.

Ξ will be built as a c.e. set of axioms of the form (σ, τ, x, y) , where $\sigma, \tau \in 2^{<\omega}$. Then $\Xi(A, L; x) \downarrow = y$ if and only if there is an axiom $(\sigma, \tau, x, y) \in \Xi$ such that both $\sigma \subset A$ and $\tau \subset L$ hold. To satisfy $X \leq_T A \oplus L$, we require that Ξ will have the following properties: at stage s ,

- (i) Whenever we enumerate (σ, τ, x, y) into Ξ , we have that $y = X[s](x)$, $\sigma \subseteq A[s]$, and $\tau \subseteq L[s]$.
- (ii) We enumerate x into X only if for all axioms $(\sigma, \tau, x, 0) \in \Xi$, either $\sigma \not\subseteq A$ or $\tau \not\subseteq L$.
- (iii) We extract x from X only if for any axiom $(\sigma, \tau, x, 1) \in \Xi$, either $\sigma \not\subseteq A$ or there is an l such that $\tau(l) > L(l)$.
- (iv) For every x , there are only finitely many axioms of the form $(\sigma, \tau, x, y) \in \Xi$.
- (v) For every x , there is an axiom $(\sigma, \tau, x, y) \in \Xi$ such that $\sigma \subset A$ and $\tau \subset L$.

We call (i)–(v) Ξ -rules. Clearly any construction with the Ξ -rules satisfies the \mathcal{T} -requirement $X \leq_T A \oplus L$.

2.5 An \mathcal{N} -Strategy

Suppose that we want to satisfy the following \mathcal{N} -requirement:

$$\mathcal{N}: \quad A = \Psi(X, L) \rightarrow A \leq_T L.$$

To satisfy \mathcal{N} , we will build a Turing functional Λ and show that if $A = \Psi(X, L)$, then $A = \Lambda(L)$. To realise this goal, we want to preserve the computation $\Psi(X, L; k)$ once $\Lambda(L; k)$ is defined.

The idea to preserve a computation $\Psi(X, L; k)$ is as follows:

- (1) Suppose that we are going to define $\Lambda(L; k)$ with use function $\lambda(k)$. Then we ensure that if there is an $x \leq \psi(k)$ which is in X , and which will leave X , then there is an $l \leq \lambda(k)$ which leaves L simultaneously.
- (2) We allow an $x \leq \psi(k)$ to be enumerated into X only if there is an $l \leq \lambda(k)$ which has entered L since the computation $\Psi(X, L; k)$ was observed. In so doing, at the stage at which $L \upharpoonright (\lambda(k) + 1)$ returns back to its value when we observed $\Psi(X, L; k)$, we can extract all $x \leq \psi(k)$ which have been enumerated into X since $\Psi(X, L; k)$ was observed.

The definition of $\lambda(k)$, which is actually the configuration of the computation at the stage at which we observed it, will be carefully chosen so that both (1) and (2) above hold. By (1), the computation $\Psi(X, L; k)$ will not be injured by X -extraction, unless the computation $\Lambda(L; k)$ becomes invalid due to L -extraction below $\lambda(k)$. Property (2) ensures that any injury of $\Psi(X, L; k)$ from X -enumeration will be recovered at the stage at which L becomes correct below $\lambda(k)$.

The \mathcal{N} -strategy will have to guarantee that for each k , $\Lambda(L; k)$ is (re)defined only finitely many times. This will be realised by delaying the building of Λ so that $\Lambda(L; k)$ is redefined only if we have passed the test of a lowness condition of L . Then the lowness of L ensures that $\Lambda(L; k)$ is redefined only finitely many times for each k .

A cycle k of the \mathcal{N} -strategy will define some *conditional restraints*. A conditional restraint is a pair (c, r) of natural numbers c , and r (or a pair (σ, r) for some 0, 1-string σ). Let v be a stage at which a conditional restraint (c, r) is created and $s > v$. We say that the conditional restraint (c, r) (or (σ, r)) is *valid at stage s* , if for any $l \leq c$, $L_v(l) \leq L_s(l)$, (or for any l , $\sigma(l) \leq L_s(l)$), and *invalid at stage s* , otherwise.

Let (c, r) be a conditional restraint created at stage v , and $s > v$. Then for any $x \leq r$, x is allowed to be enumerated into X at stage s only if there is an $l \leq c$ such that $l \in L_s \setminus L_v$, and in which case, the L -permitting marker of x is defined to be greater than c so that once $L_v \upharpoonright (c+1) = L \upharpoonright (c+1)$ at some stage $s' > s$, we are able to extract all the x 's from X .

[Notice that X is constructed below $A \oplus L$. An X -extraction needs permanent $A \oplus L$ changes below the permitting marker of the element of X .]

The \mathcal{N} -strategy will proceed as follows.

Cycle k : defining $\Lambda(L; k)$

1. Choose a *witness* $w(k)$ as fresh.
2. Wait for a stage v at which
 - (2a) $\Lambda(L; k') \downarrow = A(k')$ for all $k' < k$.
 - (2b) $\Lambda(L; k) \uparrow$.
 - (2c) $\Theta(K; f(w(k))) \downarrow = 0$.
 - (2d) $\Psi(X, L; k) \downarrow = A(k)$. Then:
 - define the *configuration of the computation* $\Psi(X, L; k)$ by

$$c = \max\{\psi(k), m^L(x) \mid x \leq \psi(k), x \in X\}$$

where $m^L(x)$ is the L -permitting marker of x , which will be defined in Definition 3.4 below,

- define $\Delta(L; w(k), f(w(k))) \downarrow = 0$ with $\delta(w(k), f(w(k))) = c$,
- define a *conditional restraint* to be $(c, \psi(k))$, or $(L \upharpoonright (c+1), \psi(k))$.

[The conditional restraint (c, r) means that if there is an $l \leq c$ which has entered L since stage v , then an element $x \leq r$ is allowed to be enumerated into X , under the condition that if at a stage $s > v$, $L_v \upharpoonright (c+1) = L_s \upharpoonright (c+1)$ occurs, then all those x can be extracted from X .]

3. (Special Attention) Wait for a stage $s > v$ at which:

(3a) $\Theta(K; f(w(k))) \downarrow = 1$, and

(3b) $\Delta(L; w(k), f(w(k))) \downarrow = 0$. Then:

- for any x , if $x \leq \psi(k)[v]$, and $x \in X - X_v$, then extract x from X ,
- define $\Lambda(L; k) \downarrow = A(k)$ with $\lambda(k) = \delta(w(k), f(w(k)))$,
- define the X -restraint r^X to be the maximum of all r 's such that there are c with which conditional restraints (c, r) which were created by the \mathcal{N} -strategy and which are still valid at the current stage,

[Remarks. (i) The X -restraint r^X is increased only by this special action of the \mathcal{N} -strategy.

(ii) A conditional restraint (c, r) counted at this stage in r^X will be excluded from the restraint r^X at the stage at which it becomes invalid. This means that we may use set \mathcal{C} to denote the collection of all conditional restraints observed at this stage, then $r^X = \max\{r \mid (\exists c)[(c, r) \in \mathcal{C}]\}$. We will always define r^X by using \mathcal{C} . However if a conditional restraint $(c, r) \in \mathcal{C}$ becomes invalid at stage $t > s$, then it leaves \mathcal{C} automatically.]

- we say that the \mathcal{N} -strategy *receives special attention at stage s* .

Suppose that $\Delta(L; w(k), f(w(k)))[v]$ is created at stage v . Let $s > v$ be a stage.

1. We say that $\Delta(L; w(k), f(w(k)))[v]$ is *valid at stage s* , if for any $z \leq \delta(w(k), f(w(k)))[v]$, $L_v(z) \leq L_s(z)$, and *invalid at stage s* , otherwise; and that $\Delta(L; w(k), f(w(k)))[v]$ is *correct at stage s* , if for every $z \leq \delta(w(k), f(w(k)))[v]$, $L_v(z) = L_s(z)$, and *incorrect at stage s* , otherwise.
2. We say that $\Psi(X, L; k)[v]$ is *recoverable at stage s* , if for any $x \leq \psi(k)[v]$, $X_v(x) \leq X_s(x)$, and *non-recoverable at stage s* , otherwise.

The principle to satisfy \mathcal{N} is that, for any k, v , and $s > v$, if $\Delta(L; w(k), f(w(k)))[v]$ is created at stage v , and $\Delta(L; w(k), f(w(k)))[v]$ is valid at stage s , then $\Psi(X, L; k)[v]$ is recoverable at stage s .

For a fixed k , we define $\Lambda(L; k)$ only in step 3, in which case, $\Theta(K; f(w(k)))$ has changed from 0 to 1. Since $\Theta(K)$ is total, $\Lambda(L; k)$ is redefined only finitely many times. Suppose that $\Lambda(L)$ is built infinitely many times. Then for every k , $\Lambda(L; k) \downarrow = A(k)$. Otherwise, there is a k such that $\Psi(X, L; k) \downarrow \neq A(k)$ holds permanently. This also ensures that if $A \not\leq_T L$, then the \mathcal{N} -strategy extracts numbers from X only finitely many times.

We use 1 to denote the unique outcome of the \mathcal{N} -strategy.

2.6 An \mathcal{R} -Strategy

Given e , an \mathcal{R}_e -strategy will define $\Gamma(X, L; e, y)$ for all $y \in \omega$. Recall that we have assumed that for every e , $e \notin W^A$ if and only if there are infinitely many stages s such that $e \notin W^A[s]$.

Whenever we define $\Gamma(X, L; e, y)$ at a stage s , we define a use block B , and enumerate an axiom (B, σ, e, y, z) into Γ , and we locate the use block B at some node ξ of the priority tree T .

Let B be the use block and (B, σ, e, y, z) to be the axiom associated with $\Gamma(X, L; e, y)$ which is created at stage v , and let $s > v$ be a stage.

We say that $\Gamma(X, L; e, y)[v]$ is *invalid* at stage s , if there is an element $b \in B$ such that $b \in X$, and *valid at stage s* , otherwise.

Suppose that α is an \mathcal{R}_e -strategy. Then α will proceed as follows.

1. If there is a stage v with $s^- < v \leq s$ such that $e \notin W^A[v]$, where s^- is the last stage at which $\alpha^{\langle 0 \rangle}$ was visited, then let y be the least z such that $\Gamma(X, L; e, z) \uparrow$, and define $\Gamma(X, L; e, y)$ by one of the following cases:

Case 1a. If there is a previous $\Gamma(X, L; e, y)$ which is valid at stage s , then let (B, σ, e, y, z) be the axiom corresponding to the most recently defined valid computation $\Gamma(X, L; e, y)[v']$ say for some v' , let $\sigma' = L_s \upharpoonright (|\sigma|)$, enumerate $(B, \sigma', e, y, 0)$ into Γ and locate the computation and its use block at the same node as before.

Case 1b. If there is no previous computation $\Gamma(X, L; e, y)$ which is valid at this stage, then create a use block as follows: we define u to be the least natural number greater than any number mentioned so far, define $B = \{u+1, u+2, \dots, 2u\}$, and $\sigma = L_s \upharpoonright (u+1)$, and enumerate the axiom $(B, \sigma, e, y, 0)$ into Γ , and locate the use block at $\alpha^{\langle 0 \rangle}$.

2. Otherwise, then let y be the least z such that $\Gamma(X; e, z) \uparrow$, and define $\Gamma(X, L; e, y)$ by one of the following cases:

Case 2a. If there is an axiom $(B, \sigma, e, y, z') \in \Gamma$ which is still valid at this stage, then define $\Gamma(X, L; e, y) \downarrow = 1$ with $\gamma(e, y)$ -use block as the same as that in case 1a above.

Case 2b. If there is no previous $\Gamma(X, L; e, y)$ axiom which is still valid at stage s , then define $\Gamma(X, L; e, y) = 1$ with $\gamma(e, y)$ -use block as in Case 1b above, and locate the computation and its $\gamma(e, y)$ -use block B at $\alpha^{\langle 1 \rangle}$, and define the permitting marker of B by

$$p^B = w^A(e) = (\mu u)[\langle e, A \upharpoonright (u+1) \rangle \in W].$$

We also denote p^B by $p^{\alpha^{\langle 1 \rangle}}$

The \mathcal{R}_e -strategy α will have two possible outcomes $0, 1$ with $0 <_{\mathbb{L}} 1$ to denote $e \notin W^A$, and $e \in W^A$ respectively. Whenever we define $\Gamma(X, L; e, y)$, we define a $\gamma(e, y)$ -use block B say, and define the *location* $\text{loc}(B)$ to be the node at which B is located. Notice that every use block has a location.

For every use block B , we define an *A-permitting marker* of B as follows:

$$\widehat{p}(B) = \max\{p^{\xi'} \mid \xi' \leq \text{loc}(B)\}.$$

For every x , if $x \in B$ for some use block B , then define the *A-permitting marker* of x by

$$m^A(x) = \widehat{p}(B).$$

[Notice that if $m^A(x)$ is not defined as above, then it is -1 .]

The enumeration of X will be determined by the *A-permitting marker* of the use blocks, and the X -restraints.

Enumerating X . If at stage s :

- (i) There is an element $a \leq \widehat{p}(B)$ which enters A at stage s ,
- (ii) $B \cap X = \emptyset$,
- (iii) $\min B > \max\{r^X(\xi) \mid \xi < \text{loc}(B)\}$, then:
 - enumerate the least unused element $b \in B$ into X ,
 - define the *L -permitting marker of B* , denoted by $\widehat{q}(B)$, to be the maximal length of strings σ with which some axiom of Δ with use σ was created by some strategy $\beta < \text{loc}(B)$ which is still valid at stage s , and which has conditional restraint (c, r) with r greater than $\min B$.

If $B \cap X \neq \emptyset$, then for every $x \in B$, define the *L -permitting marker $m^L(x)$ of x* by

$$m^L(x) = \widehat{q}(B)$$

where if $m^L(x)$ is not defined as above, we regard it as -1 .

The definition of the *L -permitting marker $m^L(x)$* allows us to extract all elements of $B \cap X$ from X if there is a strategy with priority higher than $\text{loc}(B)$ which receives special attention via the existing axiom of Δ with $\min B \leq r$. The reason is that whenever we define $\Xi(A, L; x)$ for any $x \in B$, we define $\xi(x) \geq \widehat{q}(B)$, so if a strategy $< \text{loc}(B)$ receives special attention then we get an $l \leq \widehat{q}(B)$ which leaves L , giving a permanent *L -permission* for all elements $b \in B$ to be extracted from X .

We then need to examine further the special attention.

Special Attention. We say that a node γ *requires special attention at stage s* via $\Delta(L; w(k), f(w(k)))[v]$, if:

- (i) $\Delta(L; w(k), f(w(k)))$ was created at stage v by γ ,
- (ii) $\Theta(K; f(w(k)))[s] \downarrow = 1$, and
- (iii) $\sigma \subset L_s$, where σ is the initial segment of L used in the computation mentioned in (i).

In this case, let (c, r) be the conditional restraint corresponding to $\Delta(L; w(k), f(w(k)))[v]$. Let γ receive special attention as follows: For any computation $\Gamma(X, L; e, y)[s']$ with use block B , if it is located at some node $\xi \supset \gamma$, then

- If $\min B \leq r$ and $B \cap (X - X_v) \neq \emptyset$, then,
 - extract all elements $b \in B \cap X$ from X , and
 - redefine the location of both the computation and its use block B to be γ .

The role of this step is that the rectification of $\Gamma(X, L; e, y)[s']$ will be decided by the *A -permitting markers* and *X -restraints* defined by nodes $\leq \gamma$. The reason is that all strategies below γ have been injured at this stage, we don't need to respect the *X -restraints* defined by strategies strictly below γ , and the *A -permission* which occurred before has been lost.

- Let \mathcal{C} be the set of all *valid conditional restraints of γ* , and define the *X -restraint of γ* to be the maximum of all r 's with some conditional restraint $(c, r) \in \mathcal{C}$.

2.7 An \mathcal{S} -Strategy

Given e , an \mathcal{S}_e -strategy will define $\Omega(C, W^A; e)$ such that $\Omega(C, W^A; e) \downarrow = (X \oplus L)'(e)$. Suppose that β is an \mathcal{S}_e -strategy. Then β defines $\Omega(C, W^A; e)$ by enumerating some axiom of the form (c, β, e, i) into Ω , where c is a fresh number, denoted by $\omega(e)$, and $i = 0, 1$. We can rectify $\Omega(C, W^A; e)$ by enumerating $\omega(e)$ into C . The building of Ω ensures that if β is on the true path, then $\Omega(C, W^A; e)$ will be defined by β . The construction of the priority tree ensures that the true path TP is computable relative to $K \oplus W^A$.

Initially we define $\Omega(C, W^A; e) = 0$, while the \mathcal{S}_e -strategy will try to preserve a computation $\Phi_e(X, L; e)[v] \downarrow$ for some v which has passed the L -test through $\Theta(K)$ as the same as a cycle of an \mathcal{N} -strategy, then we rectify $\Omega(C, W^A; e)$ by enumerating $\omega(e)$ into C , redefine $\Omega(C, W^A; e) = 1$, and preserve $\Phi_e(X, L; e)[v]$. We now look at the instructions of the \mathcal{S}_e -strategy β .

1. Define a *witness* w as fresh.
2. Define $\Omega(C, W^A; e) \downarrow = 0$ with $\omega(e)$ fresh.
3. Wait for a stage v at which
 - (3a) $\Theta(K; f(w)) \downarrow = 0$, and
 - (3b) $\Phi_e(X, L; e) \downarrow$. Then:
 - define $\Delta(L; w, f(w)) \downarrow = 0$ with $\delta(w, f(w)) = \max\{\phi_e(e), m^L(x) \mid x \leq \phi_e(e), x \in X\}$,
 - we say that $c = \delta(w, f(w))$ is the *configuration of the computation* $\Phi_e(X, L; e)$, and
 - define a *conditional restraint* to be $(c, \phi_e(e)[v])$.
4. (Special Attention) Wait for a stage $s > v$ at which
 - (4a) $\Theta(K; f(w)) \downarrow = 1$, and
 - (4b) $L_s \uparrow (\delta(w, f(w))[v] + 1) = L_v \uparrow (\delta(w, f(w))[v] + 1)$. Then:
 - for any $x \leq \phi_e(e)[v]$, if $x \in X_s - X_v$, then extract x from X ,
 - redefine $\Omega(C, W^A; e) \downarrow = 1$,
 - for any $\Gamma(X, L; e', y')$ defined at stage s' with use block B for some s' , if the computation $\Gamma(X, L; e', y,) [s']$ is located at some node below β , then, re-locate $\Gamma(X, L; e', y') [s']$ and its use block at β ,
 - let $\mathcal{C}(\beta)$ be the set of all valid conditional restraints of β ,
 - define the X -restraint $r^X(\beta)$ to be the maximum of all r 's with which there are c such that $(c, r) \in \mathcal{C}(\beta)$, and
 - from now on we ensure that $\Omega(C, W^A; e) \downarrow = 1$ unless $L \uparrow (\delta(w, f(w))[v] + 1) \neq L_v \uparrow (\delta(w, f(w))[v] + 1)$.

The Possible Outcomes. Since step 4 occurs only if $\Theta(K; f(w))$ has changed from 0 to 1, step 4 occurs only finitely many times. Therefore $\Omega(C, W^A; e)$ converges, and by the building of Ω , $\Omega(C, W^A; e) = (X \oplus L)'(e)$.

Updating the A -, L -Permitting Markers.

At the stage at which a use block B say is (re)-located at a node ξ say, we define the A -*permitting marker* of the use block B by

$$\widehat{p}(B) = \max\{p^{\xi'} \mid \xi' \leq \xi\}.$$

If A changes below $\widehat{p}(B)$, we are trying to enumerate an element $b \in B$ into X , and set $\widehat{p}(B)$ to be undefined, and we define the L -*permitting marker* of B to be the conditional restraints imposed by strategies higher than that of $\text{loc}(B)$. That is, we define the L -*permitting marker* of B to be the maximum of all c such that for some $r \geq |B|$, (c, r) is a valid but incorrect conditional restraint imposed by a strategy of higher priority than $\text{loc}(B)$.

The L -*permitting marker* of a use block B is created at the stage at which some element $b \in B$ is enumerated into X , which may decrease due to some conditional restraint (c, r) which was counted in the definition of the L -*permitting marker* becomes invalid, and which will never increase due to new Δ -axioms created by the strategies of higher priority than $\text{loc}(B)$.

This completes the description of the strategies.

3 The Construction

Before describing the construction, we define the priority tree of strategies.

3.1 DEFINITION. (i) We define the *priority ranking of the requirements* by

$$\mathcal{N}_0 < \mathcal{R}_0 < \mathcal{S}_0 < \mathcal{N}_1 < \mathcal{R}_1 < \mathcal{S}_1 < \dots.$$

(ii) Define the *possible outcomes of an \mathcal{R} -strategy* by 0, 1 with $0 <_{\text{L}} 1$.

(iii) Define the unique *possible outcome* of an \mathcal{N} - or an \mathcal{S} -strategy by 1.

Then the priority tree is defined as follows.

3.2 DEFINITION. We arrange the requirements on nodes of a tree level by level according to the priority ranking of the requirements. Each level will have only one requirement. The immediate successors of a node are the possible outcomes of the corresponding strategy.

The construction may be delayed if $\Theta(K; f(w)) \neq L'(f(w))$ for some witness w . So we define the following:

3.3 DEFINITION. We say that s is a *progressive stage*, if for any witness w of a strategy, $\Theta(K; f(w)) = L'(f(w))$ holds during stage s .

By the assumption that $\Theta(K) = L'$, there are infinitely many progressive stages.

During the course of the construction, we define some parameters associated with a strategy. We describe them as follows:

- (i) Given an \mathcal{N} -strategy γ , we will have the following parameters,
 - (a) $w(\gamma, k)$: the witness of cycle k of γ ,
 - (b) Λ_γ : the Turing functional built by γ ,
 - (c) $r^X(\gamma)$: the X -restraint of γ , and
 - (d) $u(\gamma)$: the ψ -use of γ .

- (ii) Given an \mathcal{S} -strategy β , we will have the following parameters,

- (a) $w(\beta)$: the witness of β ,
 - (b) $r^X(\beta)$: the X -restraint of β , and
 - (c) $u(\beta)$: the ϕ -use of β .
- (iii) Given an \mathcal{R}_e -strategy α , we use $p(\alpha \uparrow (1))$ to denote the least u such that $\langle e, A \uparrow (u+1) \rangle \in W$, if $e \in W^A$ holds.
- (iv) During the course of a computation, we may define certain use blocks.
- (a) A use block B will be assigned a *location*, denoted by $\text{loc}(B)$, which is the node on which the use block B is placed, and
 - (b) We use \mathcal{P}_ξ to denote the class of all use blocks B 's located at ξ .

3.4 DEFINITION. Given a use block B :

- (i) Define the *A-permitting marker of B* by

$$\widehat{p}(B) = \max\{p(\xi) \mid \xi \leq \text{loc}(B), \text{ and } p(\xi) \downarrow\}.$$

Then for every $x \in B$, we define $m^A(x) = \widehat{p}(B)$.

- (ii) If $B \cap X \neq \emptyset$, we define the *L-permitting marker of B* , denoted by $\widehat{q}(B)$, to be the maximal length of σ with which there is some axiom for Δ created by some strategy $\beta < \text{loc}(B)$, such that the Δ -axiom is both valid and incorrect, and $|B| \leq r$, where r is the conditional restraint of the Δ -axiom.

In this case, for every $x \in B$, we define the *L-permitting marker of x* by

$$m^L(x) = \widehat{q}(B).$$

The *L-permitting marker of a use block B* will be updated as follows. Given a stage s at which $B \cap X \neq \emptyset$, let v be the greatest stage $< s$ at which an element of B is enumerated into X . We use $\mathcal{L}[s]$ to denote the class of all conditional restraints (c, r) that were imposed by some strategy of higher priority than $\text{loc}(B)[v]$ such that (c, r) was incorrect at stage v , is valid at stage s , and $r \geq |B|$. Then the *L-permitting marker of B at stage s* is the maximum of all c such that for some r , there are conditional restraints $(c, r) \in \mathcal{L}[s]$.

This means that in the above situation, the *L-permitting marker of the use block B* is independent of the Δ -axioms created after stage v , the last stage at which some element of B is enumerated into X .

Notice that the location $\text{loc}(B)$ of a use block B may be updated during the course of the construction.

3.5 DEFINITION. Let β be an \mathcal{N} -, or an \mathcal{S} -strategy. We say that β *requires special attention at stage s* , if there are v, w such that

- (i) β enumerated an axiom for $\Delta(L; w, f(w))$ at stage v ,
- (ii) $\Theta(K; f(w)) = 1$ holds during stage s ,
- (iii) $L_v \uparrow (\delta(w, f(w))[v] + 1) \subseteq L_s$, and
- (iv) β has not received special attention via the axiom for $\Delta(L; w, f(w))[v]$.

In this case, let v_0 be the least v , and w_0 be the corresponding w with v_0 . We say that β *requires special attention at stage s via $\Delta(L, w_0, f(w_0))[v_0]$* .

During the construction, if an \mathcal{N} -strategy γ is initialised, then Λ_γ is set to be totally undefined, $w(\gamma, k)$ are all cancelled, $r^X(\gamma)$ is cancelled; and if an \mathcal{S} -strategy β is initialised, then $w(\beta)$ and $r^X(\beta)$ are both cancelled.

Given an \mathcal{N} -, or an \mathcal{S} -strategy β , and a use block B , if $\text{loc}(B) < \beta$, and an element of B is either enumerated into X or extracted from X at a stage s , then β is initialised at stage s simultaneously and automatically.

Now we are ready to describe the construction.

3.6 DEFINITION. The construction is defined stage by stage.

Stage $s = 0$. Set $X = \Gamma = \Omega = \Delta = \Xi = \emptyset$, and initialise all strategies.

Stage $s > 0$. There are four cases.

Case 1. (Delayed Action) If s is not a progressive stage, then go to stage $s + 1$.

Case 2. (Special Attention) If there is an \mathcal{N} - or an \mathcal{S} -strategy β which requires special attention at stage s , then let β_0 be the $<$ -least such β , and let v_0, w_0 be such that β_0 requires special attention at stage s via $\Delta(L; w_0, f(w_0))[v_0]$. Let β_0 receive special attention as follows.

- for any x , if $x \leq u(\beta_0)[v_0]$ and $x \in X - X_{v_0}$, then extract x from X ,
- for any computation $\Gamma(X, L; e', y')[s']$ with use block B and defined at stage s' for some s' , if $\text{loc}(B) \supset \beta_0$, then update $\text{loc}(B) = \beta_0$,
- initialise any strategy $\xi \not\leq \beta_0$,
- let \mathcal{C}_{β_0} be the set of all valid conditional restraints of β_0 ,
- let $r^X(\beta_0)$ be the maximum of all r 's with which there are c such that $(c, r) \in \mathcal{C}_{\beta_0}$, and
- go on to Case 3.

Case 3. (Enumerating X) If there is a use block B such that

- (i) There is an element $a \leq \widehat{p}(B)$ which has entered A since the construction reached Case 3 for the last time, if any,
- (ii) $B \cap X = \emptyset$, and
- (iii) $\min B > \max\{r^X(\xi) \mid \xi < \text{loc}(B)\}$, then:
 - let ξ_0 be the $<$ -least node on which there is a use block B satisfying all of (i), (ii), and (iii) above, for any \mathcal{N} - or \mathcal{S} -strategy β , if $\xi_0 < \beta$, then β is initialised,
 - for every use block B satisfying (i)–(iii), enumerate the least unused element $b \in B$ into X , and
 - go to Action Phase of Stage s .

Case 4. (The Main Body of the Construction) We specify certain nodes to be eligible to act at a substage t of stage s . At substage t of stage s , we allow some strategy to be eligible to act at substage t , and then either close the current stage or specify some strategy to be eligible to act at the next substage, i.e. substage $t + 1$, of stage s . First we allow the root node to be eligible to act at substage $t = 0$ of stage s .

Substage t . Let ξ be eligible to act at substage t . If $t = s$, then initialise all ξ' with $\xi' \not\leq \xi$, and go to stage $s + 1$. If $t < s$, then there are three subcases.

Subcase 4A. $\xi = \gamma$ is an \mathcal{N} -strategy. Then execute the following:

Program γ

1. Let k be the least x such that $\Lambda(L; x) \uparrow$.
2. If $w(k)$ is undefined, then define $w(k)$ as fresh, initialise all ξ' with $\xi' \not\leq \gamma$, and go to stage $s + 1$.

3. If:

(3a) $\Theta(K; f(w(k))) \downarrow = 0$, and

(3b) $\Psi_\gamma(X, L; k) \downarrow = A(k)$, then

– define $\Delta(L; w(k), f(w(k))) \downarrow = 0$ with use $\delta(w(k), f(w(k))) = \max\{\psi_\gamma(k), \hat{q}(B) \mid |B| \leq \psi_\gamma(k), \gamma \leq \text{loc } B, B \cap X \neq \emptyset\}$,

– create a *conditional restraint* to be $(\sigma, \psi_\gamma(k))$, where $\sigma = L_s \upharpoonright (\delta(w(k), f(w(k))) + 1)$, and

– let $\gamma \hat{\langle} 1 \rangle$ be eligible to act next.

4. Otherwise, then let $\gamma \hat{\langle} 1 \rangle$ be eligible to act next.

Subcase 4B. $\xi = \alpha$ is an \mathcal{R}_e -strategy for some e .

Suppose that the enumeration of W relative to A satisfies the usual hat trick so that for every e , $e \notin W^A$ if and only if there are infinitely many stages s at which $e \notin W^A[s]$.

We say that s is α -*expansionary*, if there is a stage v such that $s^- < v \leq s$ and $e \notin W^A[v]$, where s^- is the last stage at which $\alpha \hat{\langle} 0 \rangle$ was visited, if any. We implement the following:

Program α

1. If s is α -expansionary, then

– let y be the least z such that $\Gamma(X, L; e, z) \uparrow$,

– define $\Gamma(X, L; e, y) = 0$ and define $\gamma(e, y)$ -use block as follows:

If there is a use block B which was specified to a previous definition of $\Gamma(X, L; e, y)$ such that it is still valid at stage s , then let $\sigma = L_s \upharpoonright (|B| + 1)$, enumerate $(B, \sigma, e, y, 0)$ into Γ , and define the use block to be the same B , and locate it at old $\text{loc}(B)$, otherwise, then let u be the least number greater than any number mentioned so far, let $B = \{u + 1, \dots, 2u\}$, and let $\sigma = L_s \upharpoonright (u + 1)$, enumerate $(B, \sigma, e, y, 0)$ into Γ , define the use block to be B , and locate both the computation $\Gamma(X, L; e, y)$ and its use block B at $\alpha \hat{\langle} 0 \rangle$, and

– let $\alpha \hat{\langle} 0 \rangle$ be eligible to act at the next substage.

2. Otherwise, then:

– let y be the least z such that $\Gamma(X, L; e, z) \uparrow$,

– define $\Gamma(X, L; e, y) = 1$, and define $\gamma(e, y)$ -use block B as follows:

If there is a valid computation $\Gamma(X, L; e, y)[s']$ say, then define the use block B and its location as the same as above, otherwise, then let u be the least number greater than any number mentioned so far, let $B = \{u + 1, u + 2, \dots, 2u\}$, let $\sigma = L_s \upharpoonright (u + 1)$, enumerate $(B, \sigma, e, y, 1)$ into Γ , define the use block to be B and locate both the computation $\Gamma(X, L; e, y)$ and its use block at $\alpha \hat{\langle} 1 \rangle$,

– define $p(\alpha \hat{\langle} 1 \rangle) = (\mu u)[e \in W^{A \upharpoonright (u+1)}[s]]$, and

– let $\alpha \hat{\langle} 1 \rangle$ be eligible to act next.

Subcase 4C. $\xi = \beta$ is an \mathcal{S}_e -strategy for some e . Then implement the following program.

Program β

1. If $w(\beta)$ is not defined, then define it as fresh, initialise all ξ' with $\xi' \not\leq \beta$, and go to stage $s + 1$.

Suppose that $w(\beta) \downarrow = w$.

2. If $\Omega(C, W^A; e) = 1$ and $\Phi_e(X, L; e) \uparrow$, then enumerate $\omega(e)$ into C .
3. If $\Omega(C, W^A; e) = 0$, and there is a stage $v \leq s$ such that β has received special attention via $\Delta(L; w, f(w))[v]$, and $L_v \upharpoonright (\delta(w, f(w))[v] + 1) \subset L_s$, then
 - enumerate $\omega(e)$ into C , and
 - redefine $\Omega(C, W^A; e) = 1$ by enumerating an axiom $(\omega(e), \beta, e, 1)$ into Ω , where $\omega(e)$ is fresh.
4. If $\Omega(C, W^A; e) \uparrow$, then
 - let c be fresh, and
 - enumerate $(c, \beta, e, 0)$ into Ω .

5. If:

(5a) $\Delta(L; w(\beta), f(w(\beta))) \uparrow$,

(5b) $\Theta(K; f(w(\beta))) = 0$, and

(5c) $\Phi_e(X, L; e) \downarrow$. Then:

– define $\Delta(L; w(\beta), f(w(\beta))) = 0$ with $\delta(w(\beta), f(w(\beta))) = \max\{\phi_e(e), \hat{q}(B) \mid \min B \leq \phi_e(e), \beta \leq \text{loc } B, B \cap X \neq \emptyset\}$, where $\hat{q}(B)$ is the L -permitting marker of B defined in Definition 3.4, and

– create a *conditional restraint* to be $(\sigma, \phi_e(e))$, where $\sigma = L_s \upharpoonright (\delta(w, f(w)) + 1)$.

In any of steps 2, 3, 4, and 5, let $\beta \hat{\langle} 1$ be eligible to act next.

Action Phase of Stage s . Let $b = \max\{x \in B \mid B \text{ is a use block}\}$. For every $x \leq b$, if $\Xi(A, L; x) \uparrow$, then define $\Xi(A, L; x) \downarrow = X(x)$, and define $\xi(x)$ as follows.

Case 1. $x \notin B$ for any use block B . Then

– define $\xi(x) = -1$.

Let $x \in B$ for some use block B .

Case 2. $B \cap X = \emptyset$. Then

– define $\xi(x) = \hat{p}(B)$.

Case 3. $B \cap X \neq \emptyset$. Then

– define $\xi(x) = \hat{q}(B)$.

This completes the description of the construction.

4 The Verification

In this section, we verify that the construction satisfies all of the requirements. The verification will consist of the following propositions.

4.1 PROPOSITION. There are infinitely many progressive stages.

Proof. This is straightforward from the assumption of $\Theta(K) = L'$. \square

We now investigate the basic properties of the use blocks.

4.2 PROPOSITION. (Use Blocks Properties) Given a stage s :

- (i) If a use block B is created at stage s , then $|B| < \min B$, and if $\text{loc}(B)[s] = \xi$ for some ξ , then $|\xi|, s < |B|$.
- (ii) If B is created at a stage $v \leq s$, then

$$\text{loc}(B)[v] \supseteq \text{loc}(B)[v+1] \supseteq \cdots \supseteq \text{loc}(B)[s].$$

We say that B meets X at stage s , if there is an element $b \in B$ which is enumerated into X at stage s .

- (iii) A use block B meets X at stage s if and only if (a), (b), (c) and (d) below all hold.

- (a) $B \cap X_{s-1} = \emptyset$,
- (b) Case 3 of the construction occurs at stage s ,
- (c) $\min B > \max\{r^X(\xi)[s] \mid \xi \leq \text{loc}(B)[s]\}$, and
- (d) there is an element $a \leq \widehat{p}(B)$ which has entered A since the construction last reached Case 3 of the construction.

- (iv) If B is a use block during stage s , then $B \cap X_s$ contains at most one element.

- (v) If an element $b \in B$ is extracted from X at stage s , then $\text{loc}(B)[s-1] = \xi$ for some ξ , and there is an \mathcal{N} - or an \mathcal{S} -strategy $\beta \subset \xi$ which receives special attention at stage s , in which case, B is relocated at β .

- (vi) If an \mathcal{N} - or an \mathcal{S} -strategy β receives special attention at stage s , and β extracts an element $b \in B \cap X$ from X for some use block B , then $\beta \subset \text{loc}(B)$ held at the beginning of stage s .

Proof. By observing the construction, the proposition follows. \square

Let $\beta \subset \beta'$ be \mathcal{N} - or \mathcal{S} -strategies, and let $v' < v < s' < s$. Suppose that β' defined some axiom for Δ at stage v' , and extracted some numbers from $X - X_{v'}$ at stage s' . In this case, if β created some axioms for Δ at stage v , then β fails to restore its computation back to stage v at stage s , since the computation might have become nonrecoverable at stage s' . The crucial point is to make sure that this dangerous case will never occur in the construction.

To see this, we first look at the idea behind the construction. Let B be a use block. The L -permitting marker of B , $\widehat{q}(B)$, records the possible extractions of L which will occur at the stage at which we extract elements of $B \cap X$ from X . In the above example, suppose furthermore that β creates a conditional restraint to be (σ, r) at stage v , and β' created a conditional restraint (σ', r') at stage v' . By the construction, Case 4 of the construction occurs at stage v , so the σ' is both valid and incorrect at stage v . By the definition of conditional restraint (σ, r) , if there is an $x \leq r$ such that $x \leq r'$ which has been enumerated into X during stages $(v', v]$, then $|\sigma| \geq |\sigma'|$. This ensures that if β' extracts x

from X at stage s' , then there is an l such that $\sigma(l) = 1$ and $l \notin L_{s'}$, so that the Δ axiom created by β at stage v is no longer valid at stage s' .

4.3 PROPOSITION. Let s be a stage. Suppose that a conditional restraint (σ, r) was created at stage v , and that $v \leq s$. If for every $l < |\sigma|$, $\sigma(l) \leq L_s(l)$, in which case we say that the conditional restraint is valid at stage s , then

$$X_v \upharpoonright (r+1) \subseteq X_s.$$

Proof. We prove this by induction on stages s . Suppose by induction that the Proposition holds for all $s' < s$. We verify the proposition at stage s by dividing into the following cases.

Case 1. s is not progressive.

In this case, there is no action which is taken at stage s . The Proposition at stage s follows from the inductive hypothesis.

Case 2. Case 2 of the construction occurs at stage s .

Let β' be the strategy which receives special attention at stage s . Suppose that β' receives special attention at stage s via $\Delta(L; w', f(w'))$, that $\Delta(L; w', f(w'))$ was created at a stage $v' \leq s$, and that (σ', r') is the conditional restraint created by β' at stage v' .

Let (σ, r) be the conditional restraint created by β at stage v . If $X_v \upharpoonright (r+1) \not\subseteq X_s$, then there is an $x \leq r$ in B for some use block B which has been extracted from X at stage s . By the construction at stage s , we have that $v' < v < s$, that $x \leq r'$, and that x was enumerated into X at a stage s' with $v' < s' < v$. We observe the construction at stage v . Clearly, $\Delta(L; w', f(w'))[v']$ is both valid and incorrect at stage s .

We consider the following subcases.

Subcase 2a. $\beta \subseteq \text{loc}(B)[v]$, $B \cap X_v \neq \emptyset$, and $|B| \leq r$.

In this case, the length of σ is greater than or equal to the length of σ' , so that $\Delta(L; w, f(w))[v]$ is no longer valid at the end of stage s .

Subcase 2b. $\beta <_{\mathbb{L}} \text{loc}(B)[v]$.

Let γ be the longest strategy such that $\gamma \subset \beta$, $\gamma \subset \text{loc}(B)[v]$. Then at the stage at which some element $x \in B$ is extracted from X at stage s , then some node $\beta_0 \subseteq \gamma$ receives special attention at stage s . So β is initialised at stage s .

Subcase 2c. Otherwise. Then:

If either $B \cap X_v = \emptyset$, or $|B| > r$, then the extraction of elements of B from X at stage s does not injure the inclusion of the Proposition. Otherwise, then either $\text{loc}(B)[v] <_{\mathbb{L}} \beta$, or $\text{loc}(B)[v] \subset \beta$. In either case, since we extract elements of B from X at stage s , β is initialised during stage s , so that $\Delta(L; w, f(w))$ is no longer valid at the end of stage s .

The proposition holds at stage s in Case 2.

Case 3. Otherwise.

Then the Proposition at stage s follows from the inductive hypothesis.

Proposition 4.3 follows. \square

Before proving the satisfaction of the \mathcal{T} -requirement, we examine the locations of a use block.

4.4 DEFINITION. Let B be a use block created in the construction, and s be a stage.

(i) We say that s is in a B -gap, if $B \cap X_s = \emptyset$, and in a co- B -gap, if $B \cap X_s \neq \emptyset$.

Let v be the stage at which the use block B is created. Then

(ii) Suppose that $v = v_1 < v_2 < v_3 < \dots$ are all stages at which a B -gap is opened, and that $s_1 < s_2 < \dots$ are all stages at which the B -gaps opened at stages $v_1 < v_2 < \dots$ are closed respectively.

We first look at the permitting markers of a use block B in either B -gaps or co- B -gaps.

4.5 PROPOSITION. Let B be a use block. Assume the notations in Definition 4.4 for the use block B . Then:

(i) For each $n \geq 1$, and each $s \in [v_n, s_n)$, the A -permitting marker $\hat{p}(B)[s] \downarrow = \hat{p}(B)[v_n]$.

(ii) For each $n \geq 1$, and for all $s \in [s_n, v_{n+1})$, $\hat{q}(B)[s]$ is defined and non-increasing in stages.

(iii) For each $n \geq 1$, all numbers in B get a permanent A -permission on $\hat{p}(B)[v_n]$ at stage s_n .

(iv) For each n , and for all numbers in B , we get a permanent L -permission on $\hat{q}(B)$ at stage v_{n+1} .

(v) For every $n > 1$, $\text{loc}(B)$ is updated to some proper initial segment of the last location of B at stage v_{n+1} .

Proof. Suppose by induction that the Proposition holds for all $m < n$. We examine each part of the Proposition for n .

(i), (iii), (iv), and (v) are straightforward from the construction. For (ii), by the definition of Δ , at stage v_{n+1} , for a conditional restraint (σ, r) which has been created during stages (s_n, v_{n+1}) , if there is an $x \leq r$ which is extracted from X at stage v_{n+1} , then either (σ, r) is no longer valid at the end of stage v_{n+1} , or the strategy that created the conditional restraint (σ, r) is initialised. (ii) follows.

Proposition 4.5 follows. \square

4.6 PROPOSITION. $X \leq_T A \oplus L$.

Proof. Given x , if x is not in any use block by the end of stage x , then $x \notin X$. Let $x \in B$ for some use block B . By Proposition 4.5, $\Xi(A, L; x)$ is defined eventually. By the construction and by Proposition 4.5, whenever $\Xi(A, L; x)$ becomes undefined, we get a permanent $A \oplus L$ -permission below $\xi(x)$. By the construction, whenever we define $\Xi(A, L; x)$, we define $\Xi(A, L; x) = X(x)$. This establishes that $\Xi(A, L)$ is total and $\Xi(A, L) = X$. Proposition 4.6 follows. \square

We now define the *true path TP of the construction*.

4.7 DEFINITION. Given a stage s at which the main body of the construction is implemented:

(i) Let δ_s be the longest strategy which is visited during stage s .

(ii) Define the *true path TP of the construction* by

$$TP = \liminf_s \delta_s.$$

The true path TP has the usual property of tree constructions.

4.8 PROPOSITION. Let $A \not\leq_T L$. Then for any $\alpha \in TP$, there is a possible outcome a of α with the following properties,

- (i) $\alpha \hat{\langle} a \rangle \in TP$.
- (ii) $\alpha \hat{\langle} a \rangle$ is initialised only finitely many times.
- (iii) $\alpha \hat{\langle} a \rangle$ is visited infinitely many times.
- (iv) If α is either an \mathcal{N} - or an \mathcal{S} -strategy, then α receives special attention only finitely many times.

Proof. We prove this by induction on the length of α . Suppose by induction that the proposition holds for all $\alpha' \subset \alpha$, and that $\alpha = \alpha' \hat{\langle} a' \rangle$ for some possible outcome a' of α' . Let s_0 be minimal after which α will not be initialised. We consider the following cases.

Case 1. $\alpha = \gamma$ is an \mathcal{N} -strategy.

By the action of special attention, if there is a k such that $\Lambda_\gamma(L; k)$ is created at a stage greater than s_0 , and $\Lambda_\gamma(L; k) \downarrow \neq A(k)$ holds permanently, then for the least such k , we have that $\Psi_\gamma(X, L; k) \downarrow \neq A(k)$ holds permanently. In this case, γ acts only finitely many times, and $\alpha \hat{\langle} 1 \rangle$ is on the true path TP , and all of the (ii), (iii), and (iv) hold for α .

This argument also follows from that if $\Lambda_\gamma(L)$ is built infinitely many times, then $\Lambda_\gamma(L)$ is total, and $\Lambda_\gamma(L) =^* A$, contradicting the assumption that $A \not\leq_T L$.

Let k be the least argument on which $\Lambda_\gamma(L)$ diverges, and let $w = w(k)$ be the witness of cycle k of γ . If $\Delta(L; w, f(w))$ is redefined only finitely many times, then either $\Psi_\gamma(X, L; k) \neq A(k)$ occurs infinitely many times, or $\Theta(K; f(w)) \neq 0$. If $f(w) \in L'$, then $\Delta(L; w, f(w)) \downarrow$, so that γ receives special attention via $\Delta(L; w, f(w))$, which is a permanent computation, we have defined $\Lambda_\gamma(L; k)$, contradicting the choice of k . If $f(w) \notin L'$, and $\Theta(K; f(w)) \neq 0$, then $\Theta(K) \neq L'$, contradicting the assumption of $\Theta(K) = L'$. Therefore if $\Delta(L; w, f(w))$ is redefined only finitely many times, $\gamma \hat{\langle} 1 \rangle$ is initialised only finitely many times, $\gamma \hat{\langle} 1 \rangle \in TP$, and it is visited infinitely many times, and clearly, γ receives special attention only finitely many times. Suppose that $\Delta(L; w, f(w))$ is redefined infinitely many times. By the construction, there are infinitely many stages at which γ is visited, cycle k of γ defines an axiom for $\Delta(L; w, f(w))$, and there are only finitely many stages at which cycle $k' > k$ of γ acts during the course of the construction. Therefore, $\gamma \hat{\langle} 1 \rangle \in TP$, $\gamma \hat{\langle} 1 \rangle$ is initialised only finitely many times, $\gamma \hat{\langle} 1 \rangle$ is visited infinitely many times, and γ receives special attention only finitely many times.

In the case that $\Delta_\gamma(L; w, f(w))$ is redefined infinitely many times, we are able to prove that $\psi_\gamma(k)[s]$ is unbounded over the course of the construction as follows. Towards a contradiction, we assume that $\psi_\gamma(k)[s]$ is bounded in the construction. Then there are only finitely many use blocks B such that $|B| \leq \psi_\gamma(k)$ occurred during the course of the construction. By Proposition 4.5, for a fixed use block B , $\hat{q}(B)[s]$ is bounded during the course of the construction. This ensures that the length of axioms for $\Delta(L; w, f(w))$ are bounded during the course of the construction, contradicting the assumption that $\Delta(L; w, f(w))$ is redefined infinitely many times.

Case 2. α is an \mathcal{R}_e -strategy for some e .

If there are infinitely many α -expansionary stages, then $\alpha \hat{\langle} 0 \rangle \in TP$. By the construction, $\alpha \hat{\langle} 0 \rangle$ is initialised only finitely many times, and by program α in the main body of the construction, $\alpha \hat{\langle} 0 \rangle$ is visited infinitely many times.

Otherwise, then clearly, $\alpha^\wedge\langle 1 \rangle$ is initialised only finitely many times, $\alpha^\wedge\langle 1 \rangle \in TP$, and it is visited infinitely many times. The Proposition follows in this case.

Case 3. $\alpha = \beta$ is an \mathcal{S}_e -strategy for some e .

By the proof in Case 1, β receives special attention only finitely many times. Therefore, $\beta^\wedge\langle 1 \rangle$ is initialised only finitely many times. By the construction, $\beta^\wedge\langle 1 \rangle \in TP$, and it is visited infinitely many times.

Proposition 4.8 follows. \square

We now verify that the construction satisfies the requirements.

4.9 PROPOSITION. Every \mathcal{N}_e is satisfied.

Proof. This follows from the proof in Case 1 of Proposition 4.8. \square

4.10 PROPOSITION. Every \mathcal{R}_e is satisfied.

Proof. Let α be the \mathcal{R}_e -strategy on the true path. Let r_0 be the limit of the maximum of the X -restraints imposed by strategies $\beta < \alpha$. The existence of r_0 is guaranteed by Proposition 4.8 (iv).

We claim that for any s , if α is visited at stage s , then for any $\gamma(e, y)$ -use block B , if B is located at some node ξ to the right of α , then either $|B| \leq r_0$, or $B \cap X_s \neq \emptyset$. By the choice of B and by the assumption of $\text{loc } B = \xi$ and $\alpha <_L \xi$, there is an \mathcal{R} -strategy α' such that $\alpha'^\wedge\langle 0 \rangle \subseteq \alpha$, and $\alpha'^\wedge\langle 1 \rangle \subseteq \xi$. At the stage at which B was specified, $e' = e(\alpha') \in W^A$. Therefore at the stage at which A changed below the use function of $e' \in W^A$, all \mathcal{N} -, and \mathcal{S} -strategies β to the right of $\alpha'^\wedge\langle 0 \rangle$ are cancelled, and an A -permission for enumeration of elements of B into X occurred. So $B \cap X \neq \emptyset$. By the construction, the elements of $B \cap X$ can be extracted from X at a later stage only if some strategy $\subset \alpha'$ receives special attention.

Therefore there are only finitely many $\gamma(e, y)$ -use blocks B such that both $B \cap X = \emptyset$, and $\text{loc } (B) > \alpha$ hold permanently.

Suppose that there are infinitely many α -expansionary stages. Then by the same argument as the proof of the above claim, there are only finitely many $\gamma(e, y)$ -use blocks B such that $B \cap X = \emptyset$, and $\text{loc } (B) = \alpha^\wedge\langle 1 \rangle$ hold permanently. By program α of the construction, for almost every y , the permanent $\gamma(e, y)$ -use block is located at $\alpha^\wedge\langle 0 \rangle$. This ensures that $\lim_y \Gamma(X, L; e, y) = 0$, and that $e \notin W^A$.

If there are only finitely many α -expansionary stages, then by the same argument as above, we have that for almost every y , the permanent $\gamma(e, y)$ -use block is defined and located at $\alpha^\wedge\langle 1 \rangle$. It follows that $\lim_y \Gamma(X, L; e, y) = 1 = W^A(e)$.

So in any case, we have that $\Gamma(X, L; e, y)$ is defined for every y , and that $\lim_y \Gamma(X, L; e, y) = W^A(e)$. \mathcal{R}_e is satisfied.

Proposition 4.10 follows. \square

4.11 PROPOSITION. Every \mathcal{S}_e is satisfied.

Proof. Let β be the \mathcal{S}_e -strategy on the true path TP . By the proof of Proposition 4.9, β receives special attention only finitely many times. Therefore there are only finitely many axioms of the form (c, β, e, i) which are enumerated into Ω during the course of the construction. By the construction, $(X \oplus L)'(e) = i$ if and only if there is an axiom $(c, \beta, e, i) \in \Omega$ such that $c \notin C$.

To prove Proposition 4.11, we need the following:

4.12 PROPOSITION. Let $A \not\leq_T L$. Then $TP \leq_T K \oplus W^A$.

Proof. We compute by induction on the length of $\alpha \in TP$. For an \mathcal{N} - or an \mathcal{S} -strategy $\beta \in TP$, β receives special attention only finitely many times, and we always have that $\beta \hat{\ } \langle 1 \rangle \in TP$. For an \mathcal{R}_e -strategy α , by the construction, for $i = W^A(e)$, $\alpha \hat{\ } \langle i \rangle \in TP$. This establishes that $TP \leq_T K \oplus W^A$.

Proposition 4.12 follows. \square

So does Proposition 4.11. \square

This completes the proof of the theorem. \square

References

1. M. M. Arslanov, S. B. Cooper and A. Li, There is no low maximal d.c.e. degree, *Math. Logic Quart.* 46 (2000), 409–416; 50, No. 6 628–636 (2004).
2. S. B. Cooper, L. Harrington, A. H. Lachlan, S. Lempp, and R. I. Soare, The d.r.e. degrees are not dense, *Ann. Pure Appl. Logic* 55 (1991) 125–151.
3. S. Barry Cooper and A. Li, On Lachlan’s major sub-degree problem, to appear.
4. Y. L. Ershov, A hierarchy of sets, Part I, *Algebra i Logika* 7 (1968) 47–73 (Russian) (*Algebra and Logic* 7 (1968) 24–43 (English translation)).
5. Y. L. Ershov, A hierarchy of sets, Part II, *Algebra i Logika* 7 (1968) 15–47 (Russian) (*Algebra and Logic* 7 (1968) 212–232 (English translation)).
6. R. M. Friedberg, A criterion for completeness of degrees of unsolvability, *J. Symbolic Logic* 22 (1957), 159 – 160.
7. S. C. Kleene and E. L. Post, The upper semi-lattice of degrees of recursive unsolvability, *Ann. of Math.* (2) 59 (1954), 379–407.
8. R. W. Robinson, Interpolation and embedding in the recursively enumerable degrees, *Ann. of Math.*, (2) 93 (1971), 285 – 314.
9. R. W. Robinson, Jump restricted interpolation in the recursively enumerable degrees, *Ann. of Math.*, (2) 93 (1971), 586–596.
10. G. E. Sacks, A minimal degree less than $\mathbf{0}'$, *Bull. Amer. Math. Soc.* 67 (1961), 416–419.
11. G. E. Sacks, On the degrees less than $\mathbf{0}'$, *Ann. of Math.* 77 (1963) 211–231.
12. G. E. Sacks, The recursively enumerable degrees are dense, *Ann. of Math.* (2) 80 (1964), 300–312.
13. L. P. Sasso, A minimal degree not realizing least possible jump, *J. Symbolic Logic* 39 (1974), 571–574.
14. R. I. Soare, *Recursively enumerable sets and degrees* (Springer, 1987).
15. C. Spector, On degrees of recursive unsolvability, *Ann. of Math.* (2) 64 (1956), 581–592.

Department of Mathematics, University of Illinois, 1409 W. Green St., Urbana, Illinois 61801, USA. Email: jockusch@math.uiuc.edu

Laboratory for Models and Systems of Computation, Institute of Software, Chinese Academy of Sciences, and State Key Laboratory of Computer Science. P. O. Box 8718, Beijing, 100080, P. R. CHINA. Email: angsheng@gcl.iscas.ac.cn