Join Irreducibility in the Δ_2^0 Degrees

David Lippe

December 23, 2000

Abstract

Let R be a univalent, recursively related system of notations for ordinal α. Lippe [2000a] proves the existence of a Δ_2^0 degree x, such that for all R, α-r.e. degrees w and all n-REA degrees v, if $x \vee w \geq v$, then $w \geq v$. Such a degree is said to be *join independent for the* R, α*-r.e. degrees over the* n*-REA degrees.* In this paper, we explore what properties such an x can have. In particular, we show that x can jump to any degree REA in $0'$ and that x can be minimal. We also prove that even weak forms of these theorems fail in the r.e. degrees. In particular, we prove that there is an r.e. degree v, such that no non-zero r.e. x is join independent for the r.e. degrees over $\{v\}$. In other words, we build an r.e. degree v, such that for all non-zero r.e. x, there is r.e. $w \not\geq v$, such that $x \vee w \geq v$. This theorem fits into a long line of research into Δ_2^0 and r.e. degrees which posses special properties with respect to the join operation.

1 Introduction

Definition 1. *Let x be a Turing degree and let $\mathcal{W}$ and $\mathcal{V}$ be sets of Turing degrees such that for all $w \in \mathcal{W}$ and all $v \in \mathcal{V}$, if $x \vee w \geq v$, then $w \geq v$. Then we say that x is join independent for $\mathcal{W}$ over $\mathcal{V}$. We extend this definition to the case that X is a real and $\mathcal{W}$ and $\mathcal{V}$ are sets of reals in the obvious way.*

Lippe [2000a] proves the following.

Theorem 1. *If R is a univalent, recursively related system of notations for α, then there is a Δ_2^0 degree that is 1-generic over every incomplete R, α-r.e. degree.*

Corollary 1. *If R is a univalent, recursively related system of notations for α, then there is a Δ_2^0 degree that is join independent for the R, α-r.e. degrees over the n-REA degrees.*

This is an interesting combinatorial property for a Δ_2^0 degree, and one which obviously fails for every r.e. degree. But it seems *a priori* plausible that a weaker fact might hold in the r.e. degrees. Perhaps, for every r.e. v, there is a non-zero, r.e. x, which is join independent for the r.e. degrees over $\{v\}$. However, even this weak form of Corollary 1 fails in the r.e. degrees, as we will see in Section 3.

In Sections 4 and 5 we will look into other features that a Δ^0_2 degree can have while being join independent for the R, α-r.e. degrees over the n-REA degrees. In particular, we will show that x can jump to any degree REA in $0'$ or be minimal. These proofs are small variations of the proof used by Lippe [2000a] to produce a Δ^0_2 degree that is simultaneously 1-generic over every incomplete R, α-r.e. degree.

2 Preliminaries

We assume that the reader is familiar with the basic results and notation of mathematical logic in general and classical recursion theory in particular. We also assume familiarity with standard set theoretic conventions for sequences, functions, natural numbers, etc. A *real* is an element of Cantor space, ${}^{\omega}2$. Every real is the characteristic function of some subset of ω, and we casually identify these sets with their characteristic functions. If A is a subset of ω, we define $A^{[e]}$, the e^{th} column of A, to be $\{\langle e, n\rangle \,|\, \langle e, n\rangle \in A\}$ where $\langle ., .\rangle$ is a recursive bijection from ω^2 to ω. Given sets x and y, $x \subseteq y$ has its usual meaning, and $x \subset y$ means that x is a proper subset of y. We use upper case Roman letters for reals and lower case Roman letters for their Turing degrees. Thus, if A is a real, then a is its Turing degree. We denote the usual effective join of sets A and B by $A \oplus B$ while reserving $\vee$ for the join operation in the upper semi-lattice of Turing degrees. When no confusion can arise, we write Turing reducibility, $\leq_{\mathrm{T}}$, and Turing equivalence, $\equiv_T$, simply as $\leq$ and $\equiv$. We use lower case Greek letters such as σ, τ and ρ for elements of ${}^{<\omega}2$.

We use upper case Greek letters for Turing functionals and lower case Greek letters for the corresponding use functions. Thus, $\Gamma_e(A)$ refers to the function computed by the e^{th} oracle Turing machine when run with oracle A; and $\Gamma_e =_{\text{def}} \Gamma_e(\emptyset)$. We say that $\Gamma_e(A; n)$ converges (or $\Gamma_e(A; n)\downarrow$) to indicate that the relevant computation halts and gives output; otherwise, we say that $\Gamma_e(A; n)$ diverges (or $\Gamma_e(A; n)\uparrow$). As is common, we also discuss $\Gamma_e(\sigma; n)$ for $\sigma \in {}^{<\omega}2$ where $\Gamma_e(\sigma; n)\uparrow$ if oracle queries beyond $\text{dom}(\sigma)$ are made. $\gamma_e(A; n)$, the *use* of $\Gamma_e(A; n)$, is the smallest k such that $\Gamma_e(A{\restriction}k; n)\downarrow$ if $\Gamma_e(A; n)\downarrow$ and is undefined otherwise. We define $\gamma_e(\sigma; n)$ similarly. If A and B are reals, then $\Gamma_e(A, B; n) =_{\text{def}} \Gamma_e(A \oplus B; n)$. Without loss of generality, we can assume that whenever a query is made to either A or B, the same query is made to the other real. This guarantees that the use on A is identical to the use on B. We use the convenient, if technically inconsistent, convention that $\gamma_e(A, B; n)$ is this common use.

We define r.e. sets in the usual fashion: $W^A_e =_{\text{def}} \text{dom}(\Gamma_e(A))$ and $W_e =_{\text{def}} \text{dom}(\Gamma_e)$. In discussing an algorithm for enumerating a set W, we let $W[s]$ denote the set of numbers enumerated into W by the first $s - 1$ steps of the algorithm. If no algorithm for enumerating W_e is clear from context, then

$$W^A_e[s] =_{\text{def}} \{n < s \,|\, \Gamma_e(A; n)\downarrow \text{ in fewer than } s \text{ steps}\}\,.$$

In this paper, we will make use of two well known hierarchies of Δ^0_2 sets and degrees.

For completeness, we summarize the relevant basic facts and definitions.

Definition 2. *For $n \in \omega$, W is* n-r.e. in A *(or $W \in n$*-r.e.(A)*) if there is a total $f \leq A$ such that for all m, $f(0, m) = 0$, $W(m) = \lim_s f(s, m)$ and $|\{s \mid f(s, m) \neq f(s + 1, m)\}| \leq n$. If $A \equiv 0$, then we simply say that W is n-r.e. If W is n-r.e. in A, then W is n-r.e. in every member of a, so we say that W is n-r.e. in a. w is n-r.e. in A (or a) just in case it contains a set which is.*

Notice that only $\emptyset$ is 0-r.e.(a) and the 1-r.e.(a) sets coincide with the r.e.(a) sets. Obviously the n-r.e.(a) degrees are contained in the $(n + 1)$-r.e.(a) degrees and Cooper [1971] shows that the containment is proper. Finally, it is easy to see that if w is n-r.e.(a), then $w \in \Delta^0_2(a)$. Therefore, the n-r.e.(a) degrees provide a hierarchy of $\Delta^0_2(a)$ degrees. Unfortunately, this hierarchy does not exhaust the $\Delta^0_2(a)$ degrees because, for example, no minimal cover of a is n-r.e.(a) for any n as first shown by Cooper (unpublished). However, this hierarchy can be extended into the transfinite in order to capture all $\Delta^0_2(a)$ degrees. We present a relativization of the definition given by Epstein, Haas and Kramer [1981]. For the definition of a univalent, recursively related system of notations for an ordinal, the reader is referred to Rogers [1967].

Definition 3. *Let R be a univalent, recursively related system of notations for the infinite ordinal α. Then W is* R, α-r.e. in A *(or $W \in R, \alpha$*-r.e.(A)*) if there is $f \leq A$, such that $W(n) = f(s_n, n)$ where s_n is the R-least s such that $f(s, n)\downarrow$. If $A \equiv 0$, then we simply say that W is R, α-r.e. If W is R, α-r.e. in A, then W is R, α-r.e. in every member of a, so we say that W is R, α-r.e. in a. w is R, α-r.e. in A (or a) just in case it contains a set which is.*

We will often drop the qualifiers "univalent" and "recursively related" when discussing systems of notation since every system we use here is univalent and recursively related. Unfortunately, as suggested by the definition, there is no proper notion of α-r.e.(A), but rather a number of different notions depending upon choice of ordinal notations. However, each R, α does give a hierarchy with α levels. In particular, if $\beta < \alpha$ and R' is R restricted to notations for ordinals less than or equal to β, then the R', β-r.e. degrees are a proper subset of the R, α-r.e. degrees, as shown by Epstein et al. [1981]. It is also easy to check that any n-r.e.(A) set is R, α-r.e.(A) for any infinite α and any system of notations for α. Finally, there is the pleasing fact that the notion of ω-r.e.(A) does not depend on the system of notations. In fact, one can check that a set W is R, ω-r.e.(A) with respect to some (equivalently, all) systems R if and only if there are functions $f, g \leq A$ such that for every n, $W(n) = \lim_s f(s, n)$ and $|\{s \mid f(s, n) \neq f(s + 1, n)\}| \leq g(n)$. Finally, as proven by Epstein et al. [1981], each Δ^0_2 degree is R, ω^2-r.e. for some system of notations R.

Definition 4. *The functional $J_e : {}^\omega 2 \to {}^\omega 2$ is defined by $J_e(A) = A \oplus W^A_e$. We say that a functional J is* 1-REA *(or simply* REA*) if $J = J_e$ for some e. A functional J is n-REA if there are $e_1, e_2, \ldots e_n$, such that $J = J_{e_n} \circ J_{e_{n-1}} \circ \cdots \circ J_{e_1}$.*

Definition 5. *W is* n-REA in A *(or $W \in n$*-REA(A)*) if $W = J(A)$ for some n-REA operator J. If $A \equiv 0$, then we simply say that W is n-REA. If W is n-REA in A, then W is n-REA in every member of a, so we say that W is n-REA in a. w is n-REA in A (or a) just in case it contains a set which is.*

The two previous definitions originally appeared in Jockusch and Shore [1984]. While there is a natural way to extend the notion of n-REA into the transfinite, we have no need for that here and do not give the definition. The important result that we need about the relationship between n-REA and n-r.e. degrees is the following, proven by Jockusch and Shore [1984].

Theorem 2. *If w is n-r.e. in a, then $w \vee a$ is n-REA in a.*

3 Comparison With r.e. Degrees

3.1 Introduction

Theorem 3. *There is an r.e. degree v, such that for every non-zero r.e. degree x, there is an r.e. degree w, such that $x \vee w \geq v$ and $w \not\geq v$.*

Using different language, Theorem 3 says that there is an r.e. degree v, such that every non-zero r.e. degree can be non-trivially joined above v. There are many examples in the literature of Δ^0_2 degrees that have special properties with respect to join, although most of these concentrate on whether or not two degrees below a given degree can join up to it. For example, Posner and Robinson [1981] show that every Δ^0_2 degree can be joined to $0'$ by an incomplete Δ^0_2 degree. On the other hand, Miller [1981] provides a proof of Harrington's previously unpublished theorem that every high r.e. degree h has the so called *anti-cupping property*– there is an r.e. $w < h$, such that no r.e. $v < h$ joins w to h. Note that as a corollary of this, the degree v of Theorem 3 is necessarily incomplete. More along the lines of Theorem 3, Harrington [1978] constructs an r.e. degree a with the *plus-cupping property* which means that

$$\forall_{b \leq a} \forall_{d \geq a} \left(0 < b \rightarrow \exists_{c<d} \left(b \vee c = d\right)\right).$$

Lippe [2000b] proves that for if R is a system of notations for α, then there are R, α-r.e. degrees $b < a$ and an $R, \alpha + 1$ degree $x < a$, such that $b \vee x = a$ and for all R, α-r.e. degrees $w < a$, $b \vee w < a$. This fits into this class of theorems as a bit of a hybrid as it constructs a degree with a strong anti-cupping property and a mild join property. Slaman and Steel [1989] prove a related theorem. In particular, they show the existence of r.e. degrees $b < a$ such that for all Δ^0_2 $x < a$, $b \vee x < a$.

3.2 Conventions

We assume that the reader is familiar with tree based $0'''$ priority arguments. We use lower case Greek letters from the beginning of the alphabet to denote strategies. Strategies are named by their locations on the tree and thus are finite sequences. $\emptyset$ is the top node on the tree and $\alpha \subset \beta$ if and only if α is a tree ancestor of β. We fix an enumeration of all the strategies, and have strategy i choose all of its numbers from $\omega^{[i+2]}$. Sometimes a strategy chooses a "large, fresh number." Imprecisely, this means larger than any number used in the construction so far, including numbers chosen by other strategies; stage numbers; outputs, uses and running times of functions that have been evaluated; numbers enumerated into sets; restraints placed on sets; etc. In each case, one could reconstruct from the relevant proof exactly what is needed for the meaning of "large, fresh number."

Discussion of priority arguments often calls for conventions for dealing with sets, variables and other things that change value over time. We deal with this as follows. $\langle\text{expr}\rangle\,[s]$ indicates that the expression $\langle\text{expr}\rangle$ should be evaluated immediately before stage s of the construction. $\langle\text{expr}\rangle\,[s,\alpha]$ indicates that the expression $\langle\text{expr}\rangle$ should be evaluated immediately before strategy α runs on stage s. $\langle\text{expr}\rangle$, appearing in the pseudo-code for a strategy, indicates that the expression should be evaluated at the moment the expression is encountered. Note that this might be different from $\langle\text{expr}\rangle\,[s,\alpha]$ even when $[s,\alpha]$ is the current sub-stage. This is because other lines that execute during the sub-stage can change the values of variables that appear inside $\langle\text{expr}\rangle$. We make the parsing convention that [.] has lower priority than '$=$' and higher priority than logical connectives. So, for example, $\langle\text{expr1}\rangle = \langle\text{expr2}\rangle\,[s,\alpha]$ indicates that, when evaluated at sub-stage $[s,\alpha]$, $\langle\text{expr1}\rangle$ and $\langle\text{expr2}\rangle$ are equal. Note that it is consistent with this convention to write $\langle\text{expr1}\rangle\,[s,\alpha] = \langle\text{expr2}\rangle\,[t,\beta]$ to express what this is obviously intended to express. Since all strategies of a certain type share the same pseudo-code, there will be many different strategies sharing variable names. If v is a variable, then we refer to the value of the v belonging to strategy α as v_α. This convention can be combined with the previous one in the obvious manner, leading to expressions such as $v_\beta[s,\alpha]$.

When strategy α runs on stage s, we designate its outcome by $o(\alpha)[s]$. The global behavior of our priority argument is mostly standard. Stage s begins with execution of $\emptyset$. After strategy α has executed, the stage ends if $\text{lh}(\alpha) \geq s$. Otherwise, strategy $\alpha^\frown o(\alpha)[s]$ is executed. Unfortunately, there are some exceptions to the simple rule above. The simplest exception is that any strategy that runs during a stage can end that stage prematurely by executing the command "Stage s ends." A more complicated exception involves *links*. Any time a strategy α runs, it can *link up to* any strategy β, such that $\beta \subset \alpha$. If β runs and some α is linked up to β, then β can choose to *follow the link down to* α. As soon as β does this, strategy α begins running and the link disappears. After α finishes, the construction continues exactly as if α had been reached in the usual fashion.

The *estimated true path* on stage s is the sequence of strategies that run on stage s, in order, up to and including the first strategy that follows a link down to another strategy during the stage. Neither the first strategy to be reached by following a link nor anything that runs after it, is on the estimated true path. As usual, the true outcome of a strategy is the leftmost outcome visited infinitely often, and the true path is defined inductively by appending true outcomes starting at $\emptyset$.

Strategies are specified by giving informal pseudo-code with line numbers. Most of the pseudo-code should be self explanatory. Following notation that is common in some areas of computer science, we say "$v := \langle\text{expr}\rangle$" to assign the value of $\langle\text{expr}\rangle$ to variable v and reserve the '$=$' symbol for comparison of values. The first time that a strategy runs, it begins at line 1. Lines are executed in order until either a "Stage s ends", "Stop and give outcome o", or "Follow link down to β" command is reached. If any of these commands executes, no more lines of that strategy execute during the stage. The next time the strategy is reached, it begins executing at the line immediately following the last line that executed. There are two exceptions to this last rule. The command "Goto n, stop and give outcome o" is identical to the command "Stop and give outcome o" except that the strategy begins execution at line n the next time it runs. The other exception is when a strategy is *canceled*. The first time a strategy runs after being canceled, the construction acts as if the strategy never before executed. In particular, it begins at line 1 with all of its variables completely reset. If there are any links up to the strategy or from the strategy up to another strategy, they disappear. If the strategy is responsible for enumerating some set or functional, then the set or functional it enumerates is also reset when the strategy is canceled. The final set or functional is determined by enumerations after the strategy's last cancellation. Strategy α is canceled whenever a strategy to its left executes, whenever another strategy explicitly cancels it or whenever a higher priority strategy enumerates into W. The only strategies that can explicitly cancel a strategy are those of higher priority.

3.3 The Requirements

We construct an r.e. representative, V, of v with a priority argument. For each r.e. set X, there are two types of requirements:

A_X: X is recursive or there is an r.e. set W and a functional Ψ, such that $\Psi(X, W) = V$.

$D_{X,f}$: X is recursive or $\Theta_f(W) \neq V$ where W is as above.

Note that we index requirements with r.e. sets rather than indices for r.e. sets. We do this only to improve readability of the proof. All computation is being done with indices. In order to meet these requirements, we will employ three types of strategies. Two of the types correspond directly to the two requirements above. A_X strategies enumerate functionals, and $D_{X,f}$ strategies are Friedberg style diagonalization strategies. The third type are called R_X strategies. R_X strategies are positioned on the tree so that they act when it seems that higher priority A_X and $D_{X,f}$ strategies are failing. They act to ensure that X is in fact recursive. For each A_X strategy α, we denote the associated X, W and Ψ as X_α, W_α and Ψ_α, respectively. When α and f are clear from context, we will often refer to X, W, $\Psi(n)$ and $\Theta(n)$ instead of X_α, W_α, $\Psi_\alpha(X_\alpha, W_\alpha; n)$ and $\Theta_f(W_\alpha; n)$.

A_X strategy α enumerates Ψ_α axioms and enumerates numbers into W_α as required to cancel incorrect Ψ_α computations. In other words, if, when α runs, $\Psi_\alpha(X_\alpha, W_\alpha; n)\downarrow \neq V(n)$, then α enumerates a number smaller than $\psi_\alpha(X_\alpha, W_\alpha; n)$ into W_α.[1]. Now consider some $D_{X_\alpha,f}$ strategy, β, of lower priority than α. β is basically a Friedberg style diagonalization strategy. At some point in the construction, β may want to put diagonalization witness a into V while preserving $W_\alpha \restriction \theta_f(W_\alpha; a)$. Unfortunately, putting a into V threatens to make $\Psi_\alpha(X_\alpha, W_\alpha; a) \neq V(a)$. If $\psi_\alpha(X_\alpha, W_\alpha; a) > \theta_f(W_\alpha; a)$, this is no problem because α will fix Ψ_α without disturbing the relevant part of W_α. But

[1] Note that we have left out any indication of temporal evaluation of changing values as we have suppressed any explicit mention of the stage numbers at which the expressions in the previous sentence should be evaluated. We will continue to neglect such precision in the informal discussion that follows. Of course we will be more precise in the proofs and formal definitions.

if $\psi_\alpha(X_\alpha, W_\alpha; a) \leq \theta_f(W_\alpha; a)$, then putting a into V is fruitless unless X_α changes below $\psi_\alpha(X_\alpha, W_\alpha; a)$ so that α need not disturb the relevant part of W_α to fix Ψ_α.

However, we do have some leverage. Note that β's dilemma only occurs if X_α does not change below $\psi_\alpha(X_\alpha, W_\alpha; a)$. So while β might be prevented from diagonalizing, progress can be made toward showing that X_α is recursive by preserving $W_\alpha \restriction \psi_\alpha(X_\alpha, W_\alpha; a)$. If X_α ever changes below $\psi_\alpha(X_\alpha, W_\alpha; a)$, then diagonalization can proceed, and if X_α never changes below $\psi_\alpha(X_\alpha, W_\alpha; a)$, then $X_\alpha \restriction \psi_\alpha(X_\alpha, W_\alpha; a)$ is known. The most obvious way to exploit this is to have β choose larger and larger potential diagonalization witnesses, $(a_n)_{n\in\omega}$, and repeat the aforementioned behavior for each one. Either β will succeed in diagonalizing with one of the a_n, or X_α will be shown to be recursive as X_α never changes below $\psi_\alpha(X_\alpha, W_\alpha; a_n)$ after the stage during which β first wants to put a_n into V. Unfortunately, while there is no problem with the discussion so far, such behavior makes it impossible for strategies of lower priority than β to succeed. For if γ is such a strategy, then γ sees infinite restraint on W_α during the course of the construction. This may make it impossible for γ to ever enumerate a number into V. For if γ wants to enumerate b into V but $\Psi_\alpha(X_\alpha, W_\alpha; b)\downarrow = 0$, then γ has no good solution. It can either interfere with higher priority strategy α by simply putting b into V and causing damage to $\Psi_\alpha(X_\alpha, W_\alpha)$, or it can ignore β's restraint and put a small number into W_α so that $\Psi_\alpha(X_\alpha, W_\alpha)$ will not be damaged by putting b into V. Obviously, neither solution is tenable as it involves a lower priority strategy injuring a higher priority strategy. In case the reader is thinking that, because X_α will turn out to be recursive in our example, γ can safely place b into V without worrying about $\Psi_\alpha(X_\alpha, W_\alpha)$, we point out that it is impossible at any point in the construction to be certain that X_α is recursive. The proverbial interested reader can confirm for himself that adding outcomes to β, one of which believes β will prove X to be recursive and the other which believes that diagonalization will eventually succeed, does not overcome the problem. Even the strategies below the former outcome can not safely ignore $\Psi_\alpha(X_\alpha, W_\alpha)$.

Despite this discouraging first attempt, it is possible to exploit the observation made at the beginning of the last paragraph. In order to do this, we have β follow its leftmost outcome (also its second outcome) if the current sub-stage is $\Theta_f(W_\alpha)$-expansionary and yet no diagonalization is possible. Assume that α is β's only ancestor of type A_{X_α}. Then we make $\beta^\frown 2$ of type A_{X_α}, and follow $\beta^\frown 2$ by a full compliment of $D_{X_\alpha, f'}$ strategies, each of which attempts to ensure that $\Theta_{f'}(W_{\beta^\frown 2}) \neq V$. Analogously to β, each of these follows its leftmost outcome during any sub-stage which is expansionary and during which diagonalization is impossible. However, if $D_{X_\alpha, f'}$ strategy γ is one of these strategies, then $\gamma^\frown 2$ is not yet another A_{X_α} strategy, but is an R_{X_α} strategy. This strategy works under the assumption that both β and γ will experience infinitely many expansionary sub-stages and yet fail to diagonalize. If $\gamma^\frown 2$ is on the true path, then $\gamma^\frown 2$ will ensure that X_α is recursive.

$\gamma^\frown 2$ accomplishes this by choosing numbers b and b', and exploiting the fact that whenever it runs, $\psi_\alpha(X_\alpha, W_\alpha; b')\downarrow \leq \theta_f(W_\alpha; b')$ and $\psi_{\beta^\frown 2}(X_\alpha, W_{\beta^\frown 2}; b) \leq \theta_{f'}(W_{\beta^\frown 2}; b)$. In particular, $\gamma^\frown 2$ switches between the *primary holding state* and the *secondary holding state*, in which it guarantees that $X_\alpha \restriction \psi_\alpha(X_\alpha, W_\alpha; b')$ and $X_\alpha \restriction \psi_{\beta^\frown 2}(X_\alpha, W_{\beta^\frown 2}; b)$, respectively, fail to change. This is accomplished by placing restraint on $W_\alpha \restriction \theta_f(W_\alpha; b')$ and $W_{\beta^\frown 2} \restriction \theta_{f'}(W_{\beta^\frown 2}; b)$, respectively, and arranging with

α and $\beta^\frown 2$, respectively, that if $X_\alpha \restriction \psi_\alpha(X_\alpha, W_\alpha; b')$ and $X_\alpha \restriction \psi_{\beta^\frown 2}(X_\alpha, W_{\beta^\frown 2}; b)$, respectively, change, then $\psi_\alpha(X_\alpha, W_\alpha; b')$ and $\psi_{\beta^\frown 2}(X_\alpha, W_{\beta^\frown 2}; b)$, respectively, will be redefined as larger than $\theta_f(W_\alpha; b')$ and $\theta_{f'}(W_{\beta^\frown 2}; b)$, respectively. This evidently prevents such X_α changes from occurring, for otherwise β and γ, respectively, would be able to diagonalize with witnesses b' and b, respectively, and $\gamma^\frown 2$ would not in fact be on the true path. Furthermore, every time γ switches from the primary holding state to the secondary holding state, or vice-versa, it enumerates into W_α or $W_{\beta^\frown 2}$, respectively, to cancel $\Psi_\alpha(X_\alpha, W_\alpha; b')$ or $\Psi_{\beta^\frown 2}(X_\alpha, W_{\beta^\frown 2}; b)$, respectively, so that $\psi_\alpha(X_\alpha, W_\alpha; b')$ or $\psi_{\beta^\frown 2}(X_\alpha, W_{\beta^\frown 2}; b)$, respectively, can be redefined as large, fresh numbers. In this way γ guarantees that the initial segment of X_α that it holds is in fact cofinal during the construction. Note that by alternating between the primary and secondary holding states, $\gamma^\frown 2$ provides windows to lower priority strategies, during which they can enumerate into $W_{\beta^\frown 2}$ and W_α, respectively. Modulo some timing issues, this makes it possible for such lower priority strategies to enumerate into V without damaging either $\Psi_\alpha(X_\alpha, W_\alpha)$ or $\Psi_{\beta^\frown 2}(X_\alpha, W_{\beta^\frown 2})$.

3.4 The Tree of Strategies

This section is the only place in the proof where we must be careful about the fact that strategy types are actually indexed by indices for r.e. sets rather than by r.e. sets. If α is a strategy and $X = X_\alpha$, then *α's X-index* is simply the index that α has for X_α.

Definition 6. *If X is an r.e. set and β is of type $D_{X,f}$, then $A(\beta)$ is defined to be β's closest ancestor of type A_X which uses the same X-index as β. The tree will be defined so that β always has an ancestor of type A_X using the same X-index.*

Definition 7. *If A_X strategy α has no ancestor of type A_X which uses the same X-index, then α is a* primary *strategy. Otherwise, α is a* secondary *strategy. If β is a $D_{X,f}$ strategy, then β is primary if $A(\beta)$ is, and is secondary otherwise.*

Definition 8. *If X is an r.e. set and γ is of type R_X, then we define $S(\gamma) = \gamma^- =$ γ's unique secondary $D_{X,f}$ ancestor which shares γ's X-index. Similarly, we define $P(\gamma) = \gamma$'s unique primary $D_{X,f}$ ancestor which shares γ's X-index. The tree will be defined so that every such γ has exactly one such primary and one such secondary $D_{X,f}$ ancestor.*

We arrange the tree so that every $D_{X,f}$ strategy, β, has at least one, and no more than two, ancestors of type A_X which share β's X-index. β attempts to ensure that $\Theta_f(X, W_{A(\beta)}) \neq V$. The tree is also arranged so that if α is of type A_X, then for every path $p \supset \alpha$ and every f, there is a strategy β, of type $D_{X,f}$, with $A(\beta) = \alpha$ and $p \supset \beta$. If $D_{X,f}$ strategy β is primary, then $\beta^\frown 2$ is of type A_X, and otherwise $\beta^\frown 2$ is of type R_X; in either case, $\beta^\frown 2$ shares β's X-index. If $\delta \subset \beta$ where β is of type $D_{X,f}$, then we do not need strategies working on the same requirement as δ below $\beta^\frown 2$ unless β is primary and δ is an A_X or $D_{X,f'}$ strategy which shares β's X-index. In this way, our construction differs from some other $0'''$ constructions with which the reader may be familiar.

We now proceed with the formal definition of the tree. Recall that there are three types of strategies: A_X strategies, $D_{X,f}$ strategies and R_X strategies. A_X and R_X

strategies each have one outcome while $D_{X,f}$ strategies have two outcomes. If β is a $D_{X,f}$ strategy, then $\beta^\frown 2$ is left of $\beta^\frown 1$. If α is an A_X or R_X strategy, then $\alpha^\frown 1$ is α's unique immediate successor. Let $(X_n)_{n\in\omega}$ be the standard enumeration of all r.e. sets and let $\langle x_i, y_i\rangle$ be such that the function defined by $\lambda(i)\langle x_i, y_i\rangle$ is a recursive bijection of ω with ω^2. When we say below that a strategy is of type A_{X_n}, $D_{X_n,f}$ or R_{X_n} we mean that it uses n as an X_n-index.

1. The top node in the tree, $\emptyset$, is of type A_{X_0}.
2. Assume that node α is of type A_{X_n}. For each $i \in \omega$, let β_i be α's closest predecessor of type A_{X_i}, with the convention that α is α's closest predecessor of type A_{X_n}. Let i be least such that α has a predecessor of type $A_{X_{x_i}}$, no predecessor of type $R_{X_{x_i}}$ and no predecessor γ, such that γ is of type $D_{X_{x_i},y_i}$ and $A(\gamma) = \beta_{x_i}$. Then $\alpha^\frown 1$ is of type $D_{X_{x_i},y_i}$.
3. Assume that node α is of type $D_{X_n,f}$.
   (a) $\alpha^\frown 1$ is of type A_{X_i} were i is least such that α has no predecessor of type A_{X_i}.
   (b) If α is primary, then $\alpha^\frown 2$ is of type A_{X_n}. Otherwise, $\alpha^\frown 2$ is of type R_{X_n}.
4. Assume that node α is of type R_X. Then $\alpha^\frown 1$ is of type A_{X_i} where i is least such that α has no predecessor of type A_{X_i}.

3.5 The Individual Strategies

In this section we give pseudo-code for the individual strategies. In this pseudo-code, we make the convention that α refers to the strategy to which the pseudo-code belongs, and s refers to the stage during which the pseudo-code is executing.

Definition 9. *At any time in the construction, any $D_{X,f}$ strategy can begin or finish* waiting for an X change for b *where X is any r.e. set and b is any integer.*

This is the mechanism we use to implement the "arrangement" between R_X strategies and A_X strategies alluded to in the comments in Section 3.3. In particular, if $D_{X,f}$ strategy α is waiting for an X change for b and $A(\alpha)$ notes that $\Psi_{A(\alpha)}(b)$ has been canceled by an X change below $\psi_{A(\alpha)}(b)$, then $A(\alpha)$ will redefine $\psi_{A(\alpha)}(b)$ to be a large, fresh number. The intention is that this will allow α to use b as a diagonalization witness because when α next runs, we will have $\psi_{A(\alpha)}(b) > \theta_f(b)$. The details of all of this are of course fleshed out in the pseudo-code of the various strategies and in the proofs that everything works as intended.

Definition 10. *At any time in the construction, any secondary A_X strategy β, can execute lines from the pseudo-code of any R_X strategy γ, such that $A(S(\gamma)) = \beta$. In this case we say that those lines of γ are* executed due to β*. If one of those lines enumerates a number, then we say that γ* enumerated due to β*. If one of those lines issues restraint, then we say that γ* issued restraint due to β*. In these cases the enumeration and restraint, respectively, happen with γ's priority.*

It will be our convention that inside of pseudo-code, α refers to the actual instance of the strategy running that code. When γ's pseudo-code is executed due to β, any appearance of α evaluates to γ instead of β. We could of course simply copy the relevant pseudo-code into β (with appropriate changes) and avoid this unusual mechanism altogether. But doing things this way emphasizes the logic of the construction– β is merely allowing γ to act out of turn in order for γ to accomplish its goals. Any restraint is issued with γ's priority, and we will prove that any enumerations do not violate the restraint that γ is expected to respect.

Definition 11. *R_X strategies can enter or exit the* primary holding state *and the* secondary holding state *on any sub-stage. A strategy need not be in any holding state on a given sub-stage, and no strategy can be in both the primary holding state and the secondary holding state at the same time. Immediately after being canceled, a strategy is in no holding state.*

Definition 12. *We indicate the restraint that strategy α has on set W during sub-stage $[s, \alpha]$ by $r(\alpha, W)[s, \alpha]$.*

3.5.1 A_X

Here is the pseudo-code for an A_X strategy, α, running on stage s. Such strategies have only one outcome.

1. Let n be least such that $\Psi(n)\uparrow$ or $\Psi(n)\downarrow \neq V(n)$.
2. If $\Psi(n)\downarrow \neq V(n)$ then
   (a) $\psi(n) - 1 \searrow W$.
   (b) If there are any R_X β such that $A(S(\beta)) = \alpha$, β is in the secondary holding state and $r(\beta, W) \geq (\psi(n)[s, \alpha])$, then let β be the leftmost. Note that, if there are any such β, then there is a leftmost because no two R_X strategies lie on one path according to the definition of the tree given in Section 3.4.
   (c) If a β was chosen above, then execute lines 2 and 4b-4g of β's pseudo-code in order and cancel all strategies right of β.
   (d) Goto 4.
3. If there is $[s', \alpha'] < [s, \alpha]$ such that $\Psi(n)[s', \alpha']\downarrow$ then
   (a) Let $[s', \alpha'] < [s, \alpha]$ be largest such that $\Psi(n)[s', \alpha']\downarrow$.
   (b) If $W\restriction\psi(n)[s', \alpha'] = W\restriction\psi(n)[s, \alpha]$,
   $$\forall_\beta(A(\beta) = \alpha \rightarrow \beta \text{ is not waiting for an } X \text{ change for n})$$
   and $\forall_{m<n}(\psi(n)[s', \alpha'] > \psi(m)[s, \alpha])$, then enumerate a correct $\Psi(n)$ axiom with use $\psi(n)[s', \alpha']$, stop and goto 1. Note that the last conjunct in the if-conditional is only to ensure that ψ is a strictly increasing function.
   (c) If $A(\beta) = \alpha$ and β is waiting for an X change for n, then β is no longer waiting for an X change for n.

(d) Goto 4.

End if

4. Enumerate a correct $\Psi(n)$ axiom with large, fresh use.

5. Stop and goto line 1.

The strategy presented above is the usual one for enumerating a functional with some extra bells and whistles to implement the novel features of our global construction. Basically, the least place where Ψ is not computing V is found, and a new axiom to make Ψ correct at that argument is enumerated. If an old axiom needs to be canceled, this is handled by enumerating into W at line 2a. If α is secondary, then such an enumeration may violate the restraint of a lower priority R_X strategy which is in the secondary holding state. In this case, that strategy is switched into the primary holding state or canceled by line 2c. The reader may see no reason why we don't worry about the case that α is primary and the enumeration at line 2a violates the restraint of a lower priority R_X strategy which is in the primary holding state. The answer is that this situation simply doesn't arise because of the way elements are enumerated into V. See the pseudo-code of the subroutine into$V(a)$ below for details.

In general, if $\Psi(n)$ is canceled by a W change, this may be because a lower priority R_X strategy wants $\psi(n)$ to be defined as a large, fresh number as described in the comments in Section 3.3. On the other hand, if we let any X change which cancels $\Psi(n)$ cause $\psi(n)$ to be redefined as a large, fresh number then X, over which we have no control, may in fact cause $\Psi(n)$ to diverge at end of the construction. In general, therefore, we wish to keep $\psi(n)$ the same if it is canceled by an X change, but redefine $\psi(n)$ as a large, fresh number if it is canceled by a W change. The only exceptions to this are when some strategy is waiting for an X change for n, or when using an old value of $\psi(n)$ would stop ψ from being a monotonic function. All of these considerations are addressed in lines 3b and 3c.

3.5.2 $D_{X,f}$

Before giving the pseudo-code for $D_{X,f}$ strategies, we define an important subroutine, into$V(a)$, which is used to enumerate numbers into V. This routine takes care of the delicate timing issues that are involved in getting a number into V without injuring higher priority R_X strategies. In this following pseudo-code, α refers to the calling strategy.

Proc into$V(a)$

1. Let $S := \{\beta \subset \alpha \mid \exists_{X'} (\beta \text{ is of type } R_{X'})\}$.

2. For each $\beta \in S$, taken in decreasing order with respect to $\subset$, do

   (a) Link up to β.

   (b) Stop and give no outcome.

End for loop. Note that this loop will only cause α to be linked up to one strategy at a time. After linking to a given β, α stops and, as shown in Lemma 3, does not run again until β follows this link back down to α. At that point another iteration of the loop is performed, linking to the next β.

3. Let $T := \{\beta | \beta$ has higher priority than $\alpha \wedge \exists_{X'}(\beta$ is of type $A_{X'}) \wedge \forall_{\gamma \in S}(A(S(\gamma)) \neq \beta)\}$.

4. For each $\beta \in T$, if $\psi_\beta(a)\downarrow$ then $\psi_\beta(a) - 1 \searrow W_\beta$. Lemma 13 will show that this does not violate any higher priority restraint.

5. $a \searrow V$.

Examining the pseudo-code of R_X strategies (yet to be presented), the reader will note that they are always in the secondary holding state before following a link down to some $D_{X,f}$ strategy. into$V(a)$ exploits this behavior to get all of its caller's R_X ancestor strategies into the secondary holding state before enumerating a into V (Lemma 11 will confirm that it succeeds at this). With all of these strategies in the secondary holding state, into$V(a)$ can enumerate a into V and cancel all the $\Psi_\beta(a)$ computations for $\beta \in T$ without violating any higher priority restraint (as confirmed in Lemma 13). If $\beta \notin T$ but β is a higher priority A_X strategy, then there is some $\gamma \in S$ such that $\beta = A(S(\gamma))$. The $\Psi_\beta(a)$ computations associated with such β can be corrected by β the next time it runs. In this case, β can have γ switch into the primary holding state if this correction violates γ's restraint. As described above, this is the purpose of line 2c of the A_X pseudo-code.

$D_{X,f}$ strategies have only one outcome. The pseudo-code for a $D_{X,f}$ strategy, α, on stage s:

1. Let a_0 be a large, fresh number.

2. If such exists, choose $a > a_0$ such that

   $$\Theta(a)\downarrow = 1 \vee (\Theta(a)\downarrow = 0 \wedge (\theta(a) < \psi(a)\downarrow \vee \psi(a)\uparrow))$$

   and goto 4.

3. If $[s, \alpha]$ is a θ expansionary sub-stage then give outcome 2, stop and goto 2. Else, give outcome 1, stop and goto 2.

4. If $\Theta(a) \downarrow = 1$ then drop all previous restraint, issue restraint $\theta(a)$ on $W_{A(\alpha)}$, cancel all lower priority strategies and goto 8. We note that $a \notin V$ because of the convention, to be explained in Section 3.6, that strategies choose numbers from disjoint sets.

5. Drop all previous restraint, issue restraint $\theta(a)$ on W and cancel all lower priority strategies.

6. Call into$V(a)$.

7. Stop, give no outcome and goto 8.

8. Stop, give outcome 1 and goto 8.

3.5.3 R_X

The pseudo-code for an R_X strategy, α, on stage s:

1. Let b and b' be large, fresh numbers with b in $\omega^{[i+2]}$ and b' in $\omega^{[i'+2]}$ where $S(\alpha)$ is strategy number i and $P(\alpha)$ is strategy number i'. All that is going on here is that b and b', respectively, are being chosen from the set of numbers that, according to our conventions, $S(\alpha)$ and $P(\alpha)$, respectively, choose numbers from.

2. Let f and f' be such that $S(\alpha)$ is of type $D_{X,f}$ and $P(\alpha)$ is of type $D_{X,f'}$. Let $W = W_{A(S(\alpha))}$ and let $W' = W_{A(P(\alpha))}$. Let Ψ be the functional enumerated by $A(S(\alpha))$ and let Ψ' be the functional enumerated by $A(P(\alpha))$.

3. If $\theta_f(W; b)[s, S(\alpha)]\uparrow \vee \theta_{f'}(W'; b')[s, P(\alpha)]\uparrow$ then stop, give no outcome and goto 3.

4. If α is in the secondary holding state then

   (a) If some strategy is linked up to α, then transfer control to it. Lemma 4 will show that this operation is fact in well-defined– there is at most one strategy linked up to α.

   (b) Drop any restraint on W.

   (c) If $\psi(X, W; b)\downarrow$ then $\psi(X, W; b) - 1 \searrow W$. Lemma 13 will show that this does not violate any higher priority restraint.

   (d) Issue restraint $\theta_{f'}(W'; b')$ on W'. Lemma 2 will show that indeed $\theta_{f'}(W'; b')\downarrow$.

   (e) $P(\alpha)$ is waiting for an X change for b'.

   (f) $S(\alpha)$ is not waiting for an X change for b.

   (g) Enter the primary holding state.

   (h) Stop, give outcome and goto 4.

5. Else

   (a) Drop any restraint on W'.

   (b) If $\psi'(X, W'; b')\downarrow$, then $\psi'(X, W'; b') - 1 \searrow W'$. Lemma 13 will show that this does not violate any higher priority restraint.

   (c) Issue restraint $\theta_f(W; b)$ on W. Lemma 2 will show that indeed $\theta_f(W; b)\downarrow$.

   (d) $S(\alpha)$ is waiting for an X change for b.

   (e) $P(\alpha)$ is not waiting for an X change for b'.

   (f) Enter the secondary holding state.

   (g) If some strategy is linked up to α, then transfer control to it. Lemma 4 will show that this operation is in fact well-defined– there is at most one strategy linked up to α.

   (h) Stop, give outcome and goto 4.

   End else

3.6 The Global Construction

Most of our conventions for the global construction have already been covered. As mentioned, a strategy is canceled whenever the estimated true path goes to its left, whenever a link is followed down to a strategy to its left, whenever an explicit command to cancel it is given by a higher priority strategy or whenever a strategy of higher priority enumerates a number into V. Whenever a strategy is canceled, any links to or from that strategy are removed; it leaves any holding states it may be in; and it ceases waiting for any X changes. Note that because of links, the global construction can visit a node left of α on stage s, canceling α, without the estimated true path going left of α on stage s.

Recall that strategy i chooses numbers from $\omega^{[i+2]}$. It is important to note that this includes the choice of a made in line 2 of the $D_{X,f}$ pseudo-code. The one exception to this rule is when an R_X strategy chooses b and b' in line 1.

3.7 The Verification

3.7.1 Preliminary Lemmas

We begin with a series of lemmas proving that the dynamic behavior of the construction is well behaved. Many of these lemmas elucidate the sort of dynamics introduced by links; others confirm usual facts, such as the existence of an infinite true path containing strategies which are canceled finitely often, or that lower priority strategies respect restraint from higher priority strategies; the rest show that our implementation works as expected in various miscellaneous particulars.

Lemma 1. *Assume that a link is followed down to α on stage s. Then α is the last strategy to run on stage s. Therefore, at most one link is followed on every stage.*

Proof. Links are only established at line 2a of into$V(a)$. So, after the link is followed down to α, the loop containing 2a resumes. Either another iteration of the loop is performed, and no outcome is given by line 2b, or the loop finishes, into$V(a)$ finishes and control reaches line 7 of $D_{X,f}$ strategy α. Again, no outcome is given. □

Lemma 2. *If line 4d of R_X strategy α executes on stage s, then $\theta_{f'}(W'; b')\downarrow [s, \alpha]$. If line 5c of R_X strategy α executes on stage s then $\theta_f(W; b)\downarrow [s, \alpha]$.*

Proof. If line 4d executes on stage s, there is a greatest stage $s_0 \leq s$ during which α got past line 3. $P(\alpha)$ runs and gives outcome 2 during both s and s_0 since $P(\alpha)^\frown 2 \subset \alpha$ by the construction of the tree. $[s, P(\alpha)]$ is a $\theta_{f'}(W')$-expansionary sub-stage since $D_{X,f}$ strategies only give outcome 2 on such stages. Furthermore, $\theta_{f'}(W'; b')[s_0, P(\alpha)]\downarrow$ since α got past line 3 during stage s_0. Therefore, $\theta_{f'}(W'; b')[s, P(\alpha)]\downarrow$. No strategy between $P(\alpha)^\frown 2$ and α enumerates into W' on stage s at line 4 of into$V(a)$, for otherwise that strategy gives no outcome on stage s, contradicting the fact that α executes during stage s. Therefore, if such a strategy, β, enumerates into W' during stage s, then β is an R_X strategy with $A(P(\beta)) = A(P(\alpha))$ and $\beta \subset \alpha$. However, there is no such β by definition of the tree of strategies. The proof for when line 5c executes is similar but easier because there are no strategies between $S(\alpha) = \alpha^-$ and α. □

Lemma 3. *If β runs on stage s and α is linked up to β at the beginning of $[s, \beta]$, then β does not give outcome on stage s. Instead, β transfers control to some strategy (we have not yet shown that at most one strategy can be linked up to β at a time).*

Proof. If α links to β then β is an R_X strategy. If β is in the secondary holding state at the beginning of $[s, \beta]$, then line 4a transfers control before β can give outcome. Otherwise, line 5g transfers control before β can give outcome. □

Lemma 4. *Let α,β,α', and β' be such that $\alpha \neq \beta$ and $\alpha' \subset \beta$. If there is a sub-stage during which both α is linked up to α' and β is linked up to β', then $\alpha' \subset \beta'$.*

Proof. This is proven by induction on sub-stages $[s, \gamma]$. Assume it holds for all α, β, α', β' at the beginning of all sub-stages up to and including $[s, \gamma]$, and fix α and β. We show it holds for all choices of α', β' immediately after sub-stage $[s, \gamma]$. It is easy to see that only one link can be established per stage. So, we are done unless either α is linked up to α' or β is linked up to β' at the beginning of stage s.

Assume that β is linked up to β' at the beginning of stage s and that α links to α' during stage s, with $\alpha' \subset \beta$ and $\beta' \subseteq \alpha'$. Obviously, $\beta' \subset \alpha$. For the global construction to reach α on stage s, either β' must give outcome or a link must be followed from some $\alpha'' \subseteq \beta'$. The first is impossible by Lemma 3. The second is impossible because, by α's pseudo-code, we would have $\alpha' \subset \alpha'' \subseteq \beta'$, which contradicts the assumption that $\beta' \subseteq \alpha'$.

Assume that α is linked up to some α' at the beginning of stage s and that $\alpha' \subset \beta$. We claim that control can never reach β on stage s. Control certainly can not reach β by following the sequence of approximated outcomes, because α' will not give outcome on stage s by Lemma 3. If control reaches β by following a link from β'' then $\alpha' \subset \beta''$ by induction. But this is impossible because, again by Lemma 3, α' does not give outcome on stage s. □

Lemma 5. *Assume that α links to β on stage s. Assume that the responsible invocation of intoV(a) finishes on stage u. Then no strategy below β, other than α, runs between sub-stages $[s, \alpha]$ and $[u, \alpha]$.*

Proof. This follows more or less directly from Lemmas 3 and 4. For every sub-stage $[t, \gamma]$ between $[s, \alpha]$ and $[u, \alpha]$, α is linked up to some $\delta \subseteq \beta$ during $[t, \gamma]$. Therefore, by Lemma 3, the global construction does not get below β by following approximated outcomes. By Lemma 4, no strategy below β can be linked up to a strategy above δ during $[t, \gamma]$ if α is linked up to δ during $[t, \gamma]$. Therefore, the global construction can not go below β, except to α, by following a link during such sub-stages $[t, \gamma]$. □

As usual, we will confirm our theorem by showing that the action along the true path guarantees that all requirements are met. Since the true path is just the limit infimum of the estimated paths, it is trivial to show that a true path exists. Therefore, we will mention the true path before we have guaranteed its usual properties– that it is infinite and that strategies along it are canceled only finitely often.

Lemma 6. *Assume that $D_{X,f}$ strategy α executes line 2a of intoV(a) on stage s, linking to β. Assume further that β is on the true path. Either α is canceled after sub-stage $[s, \alpha]$ or the relevant instantiation of intoV(a) completes.*

Proof. Assume that α is not canceled after stage s. Until the loop containing line 2a finishes, α will continue linking to strategies $\gamma \subseteq \beta$. The only thing to prove is that if α links to such a γ in line 2a after or during sub-stage $[s, \alpha]$, then γ eventually follows that link down to α so that the loop may proceed to the next step. Note that the link from α up to γ can not go away unless this happens, because the only two ways for a link to disappear are for it to be followed or for a strategy at one end of the link to be canceled. Since $\gamma \subset \alpha$ and α is not canceled after stage s by assumption, the latter alternative is impossible. So, by Lemma 3, γ follows the link down to α as soon as it runs; and it is guaranteed to run again since it is on the true path. □

Lemma 7. *Let α, of type $D_{X,f}$, be on the true path. If α calls into$V(a)$ on stage s and is never canceled after stage $s-1$, then into$V(a)$ eventually completes. It follows that α enumerates some number into V after stage $s-1$ and has true outcome 1.*

Proof. This is a trivial consequence of the previous lemma. □

Lemma 8. *If α is on the true path, then α is canceled finitely often.*

In many priority arguments, this fact follows almost entirely from the definition of the true path as the limit infimum of approximated outcomes. In our case, however, the situation is complicated by links and by line 2c in the pseudo-code of A_X strategies. The problem with links is that it seems possible for the construction to go left of α infinitely often without the estimated true path going left of α infinitely often. While this can not in fact happen, that fact requires proof. The problem with line 2c is that it seems possible for one strategy on the true path to cancel another strategy on the true path infinitely often. Again, we must prove that does not happen.

Proof. Note that, although there are other ways for a strategy to be canceled, we only need to concern ourselves with the two mentioned above. This is because other sources of cancellation from higher priority strategies (in particular, enumerations into V and cancellation at lines 4 and 5 of $D_{X,f}$ strategies) are inherently finite. After it is last canceled, any given strategy will enumerate at most one number into V and execute each of those lines at most once.

Assume that β establishes a link up to another strategy during stage s, and is not on the estimated true path between stages s and t. Then β is on the estimated true path on stage s, and, by the definition of R in line 1 of into$V(a)$, β does not run more than $\mathrm{lh}(\beta)$ many times between stages s and t. Thus, for the global construction to go left of α infinitely often, the estimated true path must go left of α infinitely often. It now follows, from the definition of the true path as the limit infimum of the estimated true paths, that the global construction goes left of α finitely often.

Now we address 2c of the A_X pseudo-code. Since only finitely many A_X strategies of higher priority than α execute, it suffices to show that each one cancels α at line 2c finitely often. Fix some such strategy, β. If β executes line 2c, canceling α on stage s, then there must be some γ of type R_X which is left of α and such that $A(S(\gamma)) = \beta$ and γ is in the secondary holding state immediately before sub-stage $[s, \beta]$. Note that γ is in the primary holding state immediately after sub-stage $[s, \beta]$. Since α is on the true path, every γ that is left of α enters the secondary holding state finitely often, and only finitely many strategies left of α ever run. This suffices to show that β cancels α finitely often, and the lemma follows. □

Lemma 9. *If α is on the true path, then α gives an outcome on infinitely many stages.*

Proof. By the previous lemma, assume that all stages and events discussed in this proof take place after α is last canceled. The case where α is of type A_X is trivial because such strategies give an outcome every time they run. For other strategy types, it suffices to show that if α runs on a given stage without giving outcome, then there is a later stage during which α gives outcome. If $D_{X,f}$ strategy α runs without giving outcome, then the last line executed must be either line 2b of into$V(a)$ or line 7 of α's pseudo-code. If the latter, then Lemma 7 guarantees that α gives outcome 1 on some later stage. If the former, then α obviously gives outcome the next time it runs.

If R_X strategy α runs without giving outcome, then the last line executed must be either line 4a or 5g. So assume that α executes line 4a or 5g, transferring control to strategy β on stage s. It suffices to show that there is some stage $t > s$, such that α runs on stage t and no strategy is linked up to α on stage t. In fact, we can let t be the first stage, after s, during which α runs. By Lemma 6, β's invocation of into$V(a)$ returns between stages s and t, and by lemma 5, no strategy below α runs between sub-stages $[s, \beta]$ and $[t, \alpha]$. So if γ is linked up to α at sub-stage $[t, \alpha]$, then γ was linked up to α at sub-stage $[s, \beta]$ and $\gamma \neq \beta$. This is impossible by Lemma 4. □

Lemma 10. *The true path is infinite and every strategy on the true path is canceled finitely often.*

Proof. This follows directly from the two previous lemmas. □

Lemma 11. *Assume that α is on the true path, α links up to β on stage s and α is never canceled after stage $s-1$. When α's invocation of into$V(a)$ reaches line 3 (which it will by Lemma 7), β is in the secondary holding state.*

Proof. By Lemma 6, there is a stage $t > s$ such that β follows the link down to α. As a quick glance at its pseudo-code confirms, β is in the secondary holding state at the end of sub-stage $[t, \beta]$ and, by Lemma 5, does not run again until after α's invocation of into$V(a)$ returns, say at stage v. Therefore, if β leaves the secondary holding state between sub-stages $[t, \beta]$ and $[v, \alpha]$, it must be because line 4g of β executes due to line 2c of $A(S(\beta))$ at some u with $[t, \beta] < [u, A(S(\beta))] < [v, \alpha]$. Assume this is so. Then some γ enumerates some a into V between sub-stages $[t, \alpha]$ and $[u, A(S(\beta))]$, such that $\psi_{A(S(\beta))}(a) < r(\beta, W_{A(S(\beta))})[u, A(S(\beta))]$. Note that the enumeration did indeed happen after sub-stage $[t, \alpha]$ because $A(S(\beta))$ must have run on stage t, from which it follows that $\forall_n \left(\Psi_{A(S(\beta))}(n)\downarrow \rightarrow \Psi_{A(S(\beta))}(n) = V(n)[t, \beta]\right)$. But β enumerates nothing into V, and transfers control to α. Therefore, $\forall_n \left(\Psi_{A(S(\beta))}(n)\downarrow \rightarrow \Psi_{A(S(\beta))}(n) = V(n)[t, \alpha]\right)$. γ has lower priority than β since the enumeration of a into V does not cancel β. Unless $\beta \subset \gamma$, α is left of γ and γ was canceled at sub-stage $[t, \alpha]$. Therefore, a is greater than $r\left(\beta, W_{A(S(\beta))}\right)[t, \alpha] = r\left(\beta, W_{A(S(\beta))}\right)[u, A(S(\beta))]$, the restraint that β has on $W_{A(S(\beta))}$ when line 2c of $A(S(\beta))$ executes. But this is a contradiction. If $\beta \subset \gamma$ then γ does not run between $[t, \beta]$ and $[v, \alpha]$ by Lemma 5. Again, this is a contradiction. □

Lemma 12. *Assume that β is left of α, or that is not of type R_X and has higher priority than α. If β issues restraint during $[s,\gamma]$, then α is canceled during $[s,\gamma]$.*

Proof. If β is left of α, then the conclusion follows from the definition of the global construction and line 2c of the A_X pseudo-code. If β is not of type R_X and has higher priority than α, then the conclusion follows directly from β's pseudo-code. $\square$

Lemma 13. *No numbers enumerated by any strategy are in violation of higher priority restraint.*

In interpreting the statement of this lemma, it is important to recall that if R_X strategy α enumerates a number at line 4c due to A_X strategy β, then this enumeration is done by α. In other words, we must show that this enumeration does not violate the restraint of any strategy with higher priority than α. This is stronger than showing the trivial fact that it does not violate the restraint of any strategy with higher priority than β. On the other hand, if α issues restraint at line 4d due to β, then this restraint only need be respected by strategies with lower priority than α.

Proof. The case of A_X strategies is trivial because such a strategy, α, only enumerates into W_α and no higher priority strategy ever places restraint on W_α. So it suffices to consider numbers enumerated by lines 4c and 5b of R_X strategies and line 4 of into$V(a)$.

Let α be a $D_{X,f}$ strategy that enumerates $\psi_\beta(a)-1$ into W_β at line 4 of into$V(a)$ on stage s, where β is an $A_{X'}$ strategy. Assume further that γ has higher priority than α and γ has non-zero restraint $r=r(\gamma,W_\beta)[s,\alpha]$ on W_β at sub-stage $[s,\alpha]$. Then γ is left of α or γ is a $D_{X',f'}$ strategy. For the only alternative is that γ is an $R_{X'}$ strategy and $\gamma\subset\alpha$. By Lemma 11, γ is in the secondary holding state on sub-stage $[s,\alpha]$ from which it follows that $\beta=A(S(\gamma))$ since γ only restrains $W_{A(S(\gamma))}$ when in the secondary holding state. Therefore, $\beta\notin T$ on line into$V(a)$ of α's pseudo-code, a contradiction. The conclusion now follows because, by lemma 12, α was canceled when r was issued. $(a_0)_\alpha$ was subsequently chosen as a large, fresh number and hence $\psi_\beta(a)\geq a>(a_0)_\alpha>r[s,\alpha]$.

Assume that R_X strategy α enumerates $\psi_{A(S(\alpha))}(b)$ into $W_{A(S(\alpha))}$ at line 4c or enumerates $\psi_{A(P(\alpha))}(b')$ into $W_{A(P(\alpha))}$ at line 5b on sub-stage $[s,\beta]$ where $\beta=\alpha$ or $\beta=A(S(\alpha))$. Assume further that γ is a higher priority strategy that has non-zero restraint r on $W_{A(S(\alpha))}$ or $W_{A(P(\alpha))}$, respectively, at sub-stage $[s,\beta]$. Then γ is left of α or γ is a $D_{X,f'}$ strategy since no path contains two R_X strategies by construction of the tree. As above, this means that α was canceled when r was issued. It follows that $\psi_{A(S(\alpha))}(b)\geq b>r[s,\beta]$ or $\psi_{A(P(\alpha))}(b')\geq b>r[s,\beta]$, respectively, and the enumeration does not injure γ. $\square$

Lemma 14. *Let β be a secondary $D_{X,f}$ strategy and let β' be β's unique tree ancestor of type $D_{X,f'}$ such that $\beta'^\frown 2\subset\beta$ (i.e., $\beta'=A(\beta)^-$). Let $\alpha'=A(\beta')$. If any strategy of higher priority than $\beta^\frown 2$ enumerates into $W_{\alpha'}$ during stage s, then $\beta^\frown 2$ is canceled during stage s. Therefore, if $\beta^\frown 2$ is never canceled after stage s_0-1 and γ enumerates into $W_{\alpha'}$ after stage s_0-1, then $\gamma=\beta^\frown 2$ or γ has lower priority than $\beta^\frown 2$.*

Proof. Let $\alpha = A(\beta)$. Assume γ enumerates into $W_{\alpha'}$ on stage s and has higher priority than $\beta^\frown 2$. First we note that $\gamma \neq \alpha'$ because primary A_X strategies never enumerate numbers into W's. To see this, note that A_X strategies only enumerate numbers into W's to fix incorrect Ψ computations. But if γ has lower priority than α' and enumerates into V, then α' will be in T on line 3 of the responsible invocation of into$V(a)$. into$V(a)$ will ensure that $\Psi_{\alpha'}(a)$ is canceled before enumerating into V.

If the enumeration happens inside of into$V(a)$, then γ also enumerates into V on s, and β is canceled. If the enumeration happens outside of into$V(a)$, then γ is of type R_X. Since no path contains two R_X strategies, γ is left of $\beta^\frown 2$, and $\beta^\frown 2$ is canceled when the global construction reaches γ. □

Lemma 15. *Let β be a secondary $D_{X,f}$ strategy and let $\alpha = A(\beta)$. If any strategy of higher priority than $\beta^\frown 2$, other than α, enumerates into W_α during stage s, then $\beta^\frown 2$ is canceled during stage s. Therefore, if $\beta^\frown 2$ is never canceled after stage $s_0 - 1$ and γ enumerates into W_α after stage $s_0 - 1$, then $\gamma = \beta^\frown 2$, $\gamma = \alpha$ or γ has lower priority than $\beta^\frown 2$.*

Proof. Assume $\gamma \neq \alpha$ enumerates into W_α on stage s and has higher priority than $\beta^\frown 2$. If the enumeration happens inside of into$V(a)$, then γ also enumerates into V on s, and β is canceled. If the enumeration happens outside of into$V(a)$, then γ is of type R_X, and as above is left of $\beta^\frown 2$. Again, β is canceled. □

Lemma 16. *If the global construction reaches α finitely often, then α enumerates finitely many numbers during the construction.*

Proof. The only non-trivial thing to show is that if α is of type R_X, then only finitely many numbers are enumerated at line 4c due to $A(S(\alpha))$. This is true because during any sub-stage in which line 4c is executed due to $A(S(\alpha))$, line 4g is also executed due to $A(S(\alpha))$. 4c can not be executed again until α returns to the secondary holding state, and α can not return to the secondary holding state without actually being reached by the global construction. □

3.7.2 The Main Lemmas

Fix X. We show that requirements A_X and $(D_{X,f})_{f\in\omega}$ are met. There are two cases to consider. The first is addressed in Lemma 17 and the other in Lemma 18.

Lemma 17. *Let p be the true path. Let α be the maximal (as a string), or equivalently lowest priority, strategy of type A_X on p. Assume that no $D_{X,f}$ strategy below α on p has outcome 2. Then $\Psi_\alpha(X; W_\alpha) = V$ and $W_\alpha \not\geq V$.*

Proof. Let $\beta \supset \alpha$ be a $D_{X,f}$ strategy on p. We show that $\Theta_f(W) \neq V$. Using Lemma 10 and assuming familiarity with Friedberg style diagonalization strategies, it suffices to show that after some stage, any restraint issued by lines 4 and/or 5 of β's pseudo-code is respected. Lemma 13 guarantees that lower priority strategies respect all such restraint. Any strategies left of β are left of the true path and hence enumerate finitely many numbers during the construction by lemma 16. Each of β's ancestors of type $D_{X,f'}$ makes at most one call to into$V(a)$ after it is last canceled, and hence enumerates finitely many numbers during the construction. Finally, by assumption, there are no R_X strategies on p. Since all strategies that enumerate into W_α, other than α, are of type R_X or $D_{X,f'}$, we are done if we can show that α eventually stops injuring β through execution of line 2a.

If α enumerates into W_α at line 2a on stage s, it is because the conditional in line 2 is true. So, some γ enumerated some a into V on some $s' \leq s$. Choose s_0 large enough that if $s \geq s_0$ then γ is of lower priority than β. Assume that line 4 or 5 of β places restraint r on W_α at stage $s_1 \geq s_0$. Note that $\forall_n(\Psi_\alpha(n)\downarrow \rightarrow \Psi_\alpha(n) = V(n)[s_1, \beta])$ since α runs on stage s_1. Assume that $s > s_1$, that restraint r has not been dropped between sub-stages $[s_1, \beta]$ and $[s, \alpha]$, and that α enumerates $\psi_\alpha(a)[s, \alpha] - 1$ into W_α on sub-stage $[s, \alpha]$. Then some γ of lower priority than β enumerated a into V on some stage $s' \geq s_1$. γ was canceled during sub-stage $[s_1, \beta]$ and consequently $a > r$. But $\psi_\alpha(a)[s, \alpha] \geq a$, so there is no injury to β.

To see that $\Psi(X, W) = V$, let $s_0 - 1$ be the last stage on which α is canceled. On the next stage, α begins enumerating the functional Ψ. Since α, being on the true path, runs infinitely often, line 2 of α's pseudo-code ensures that if $\Psi(X, W)$ is total then $\Psi(X, W) = V$. Ψ is total if for each n there is a stage $s^*_{n+1} \geq s_0$, such that $\psi(n)\downarrow [t] = \psi(n)\downarrow [s_{n+1}]$ all $t \geq s_{n+1}$. In order for $\psi(n)$ to grow, W must change below the current value of $\psi(n)$, or there must an $m \leq n$ such that X changes below the current value of $\psi(m)$ and some strategy is waiting for an X change for m when the new $\psi(m)$ axiom is enumerated. However, the latter can only happen n times [note:this requires a bit of proof]. Therefore, it suffices to confirm that there is a stage $s_{n+1} \geq s_0$ such that for all $t \geq s_{n+1}$, $W{\upharpoonright}\psi(n)[s_{n+1}] = W{\upharpoonright}\psi(n)[t]$.

There are four ways that numbers can enter W: line 2a of α; line 4 inside into$V(a)$; and lines 4c and 5b of R_X strategies. We show that each of these possibilities is responsible for finitely many instances of a number below $\psi(n)[s, \beta]$ entering W during sub-stage $[s, \beta]$.

It is easy to see that, after α is last canceled, there are at most n many stages s, such that α enumerates $\psi(m)[s, \alpha] - 1$ into W at sub-stage $[s, \alpha]$ for some $m \leq n$ at line 2a. If R_X strategy β enumerates a number smaller than $\psi(n)[s, \gamma]$ into W during sub-stage $[s, \gamma]$ at line 4c due to γ ($\gamma = \beta$ or $\gamma = \alpha$), then $b_\beta[s] \leq n$ and $A(S(\beta)) = \alpha$. Since only finitely many strategies β ever choose $b_\beta \leq n$, we wish to show that, for each such β, there are only finitely many $[s, \gamma]$ during which $b_\beta[s, \gamma] \leq n$ and $\psi(b_\beta)[s, \gamma] - 1$ is enumerated into W at line 4c due to γ. We are done if β is right of p, for then there is a t such that $\forall_{s \geq t}(b_\beta[s] > n)$, and we are done if β is left of p by Lemma 16. One of these cases must apply, for there is no R_X strategy on p by assumption.

If β enumerates a number smaller than $\psi(n)[s, \beta]$ into W during sub-stage $[s, \beta]$ at line 5b, then β enumerates $\psi_{A(P(\beta))}(b'_\beta)[s, \beta] - 1$ into $W_{A(P(\beta))}$ and hence $A(P(\beta)) = \alpha$ and $b'_\beta[s, \beta] \leq n$. Since only finitely many strategies β ever choose $b'_\beta \leq n$, we wish to show that, for each such β with $A(P(\beta)) = \alpha$, there are only finitely many $[s, \beta]$ during which $b'_\beta[s, \gamma] \leq n$ and line 5b is reached. By assumption, such a β is not on p. If β is right of the true path then we are done because eventually $b'_\beta > n$. If β is left of p, then we are done by lemma 16.

If β enumerates a number smaller than $\psi(n)[s, \beta]$ into W during sub-stage $[s, \beta]$ at line 4 of into$V(a)$, then $(a_0)_\beta[s] \leq n$. Since only finitely many strategies β ever choose $(a_0)_\beta \leq n$, we wish to show that, for each such β, there are only finitely many $[s, \beta]$ during which $(a_0)_\beta[s, \beta] \leq n$ and something is enumerated into W at line 4 of an instantiation of into$V(a)$ called by β. We are done if β is right of p, for then there is a t such that $\forall_{s \geq t}((a_0)_\beta[s] > n)$. We are also done if β is left of p, for then β runs finitely often. If β is on p, then eventually β stops being canceled. Once this happens, it follows from Lemma 7 that into$V(a)$ will be invoked at most once by β. ☐

Lemma 18. *Let p be the true path. Let α be the maximal (as a string), or equivalently lowest priority, strategy of type A_X on p. Assume that there is $\beta \supset \alpha$ of type $D_{X,f}$ on p with outcome 2. Then X is recursive.*

The proof of this lemma is simplified by some auxiliary lemmas.

Lemma 19. *Under the hypotheses of Lemma 18, there are α' and β' such that $\alpha' \subset \beta' \subset \beta'^\frown 2 = \alpha \subset \beta$, α' is of type A_X, β' is of type $D_{X,f'}$ and $A(\beta') = \alpha'$.*

Proof. The conclusion of the lemma follows from the construction of the tree of strategies since β is a secondary strategy. If β were not a secondary strategy, then, by construction of the tree, $\beta^\frown 2$ would be of type A_X, contradicting α's maximality. ☐

For the remainder of this section, α' and β' refer to the strategies in the conclusion of Lemma 19; β' is of type $D_{X,f'}$; $W = W_\alpha$ and $W' = W_{\alpha'}$; s_0 is a fixed stage so that $\beta^\frown 2$ (and hence α, α', β and β') are never canceled after stage $s_0 - 1$, $\beta^\frown 2$ runs on s_0, $\Theta_{f'}(b')[s_0, \beta']\downarrow$ and $\Theta_f(b)[s_0, \beta]\downarrow$; and b and b' refer to the unique values that $b_{\beta^\frown 2}$ and $b'_{\beta^\frown 2}$ assume at all stages after $s_0 - 1$.

Lemma 20. *Let $s \geq s_0$ and assume that $\beta^\frown 2$ enters the primary holding state on sub-stage $[s, \gamma]$ and next leaves the primary holding state during stage $u \geq s$. Then*

$$X\restriction(\psi_{\alpha'}(b')\downarrow)[s, \gamma] = X\restriction(\psi_{\alpha'}(b')\downarrow)[u, \beta^\frown 2].$$

Furthermore, a number smaller than $\psi_{\alpha'}(b')[u, \beta^\frown 2]$ enters W' during sub-stage $[u, \beta^\frown 2]$.

Proof. The fact that a number smaller than $\psi_{\alpha'}(b')[u, \beta^\frown 2]$ enters W' during sub-stage $[u, \beta^\frown 2]$ will follow easily if we can prove the rest of the lemma, because then line 5b of $\beta^\frown 2$ will enumerate $\psi_{\alpha'}(b') - 1[u, \beta^\frown 2]$ into W' during sub-sub-stage $[u, \beta^\frown 2]$. As usual, $\gamma = \alpha$ or $\gamma = \beta^\frown 2$. In either case, β' runs on stage s and gives outcome 2. Thus, $\psi_{\alpha'}(b')\downarrow \leq \theta_{f'}(b')\downarrow [s, \beta']$ and

$$W'\restriction(\theta_{f'}(b')\downarrow)[s, \beta'] = W'\restriction(\theta_{f'}(b')\downarrow)[s, \gamma]$$

by Lemma 14. Since X only changes between stages, we see that $\psi_{\alpha'}(b')\downarrow [s, \gamma]$, as claimed in the lemma. $\beta^\frown 2$ places restraint $r = \theta_{f'}(b')[s, \gamma] = \theta_{f'}(b')[s, \beta']$ on W' on sub-stage $[s, \gamma]$. By Lemmas 13 and 14 and the fact that $\beta^\frown 2$ does not drop r until $[u, \beta^\frown 2]$, we see that

$$W'\restriction(\theta_{f'}(b')\downarrow)[s, \beta'] = W'\restriction(\theta_{f'}(b')\downarrow)[u, \beta'].$$

If the conclusion of the lemma is false, then it must be that

$$X\restriction\psi_{\alpha'}(b')[s, \beta'] \neq X\restriction(\psi_{\alpha'}(b')[s, \beta'])[u, \beta'].$$

Note that the last sentence is meaningful– control reaches $\beta^\frown 2$ on u, so β' runs on u. Let t be least such that $s < t \le u$, β' runs on t and

$$X\restriction\psi_{\alpha'}(b')[s, \beta'] \neq X\restriction(\psi_{\alpha'}(b')[s, \beta'])[t, \beta'].$$

Note that β' is waiting for an X change for b' immediately after sub-stage $[s, \gamma]$. Therefore, lines 3b and 3c of α guarantee that $\psi_{\alpha'}(b')\uparrow\ [t, \beta']$ or $\psi_{\alpha'}(b') > \theta_{f'}(b')[s, \beta'] = \theta_{f'}(b')[t, \beta']$. β' will call into$V(b')$ during $[t, \beta']$, so that β' has true outcome 1 by Lemma 7, a contradiction. □

Lemma 21. *Let $s \ge s_0$ and assume that $\beta^\frown 2$ enters the secondary holding state on sub-stage $[s, \beta^\frown 2]$ and next leaves the secondary holding state during sub-stage $[v, \gamma]$. Then*

$$X\restriction(\psi_\alpha(b)\downarrow)[s, \beta^\frown 2] = X\restriction(\psi_\alpha(b)\downarrow)[v, \gamma].$$

Furthermore, a number smaller than $\psi_\alpha(b)[v, \gamma]$ enters W during sub-stage $[v, \gamma]$.

Proof. For $\beta^\frown 2$ to enter the secondary holding state on stage s, it must be that β runs and gives outcome 2 during s so that $\psi_\alpha(b)\downarrow \le \theta_f(b)\downarrow\ [s, \beta]$, and $\beta^\frown 2$ places restraint $r = \theta_f(b)[s, \beta^\frown 2] = \theta_f(b)[s, \beta]$ on W during sub-stage $[s, \beta^\frown 2]$. Since neither X nor W changes from immediately before $[s, \beta]$ to immediately before $[s, \beta^\frown 2]$, we see that $\psi_\alpha(b)\downarrow\ [s, \beta^\frown 2]$, as claimed in the lemma. By Lemmas 13 and 15,

$$W\restriction(\theta_f(b)\downarrow)[v, \gamma] = W\restriction\theta_f(b)[s, \beta^\frown 2].$$

To see this, note that it follows directly from Lemmas 13 and 15 that the only strategy that can enumerate into W between $[s, \beta^\frown 2]$ and $[v, \gamma]$ is α. If α enumerates into $W\restriction(\theta_f(b)[s, \beta^\frown 2])$ during that time, it is at line 2a and either $\beta^\frown 2$ is chosen in line 2b or $\beta^\frown 2$ is canceled at line 2c. The latter is impossible by choice of s_0, so we see that this enumeration by α in fact takes place during stage v. Proceeding with the proof of the lemma, assume for the sake of contradiction that the lemma fails. Then

$$X\restriction(\psi_\alpha(b)[s, \beta^\frown 2])[v, \gamma] \neq X\restriction\psi_\alpha(b)[s, \beta^\frown 2]. \tag{1}$$

Note that, because α runs on stage s, $\forall_n(\Psi_\alpha(n)\downarrow \rightarrow \Psi_\alpha(n) = V(n)[s, \beta])$. Therefore, if $\gamma = \alpha$, then, letting n be the number chosen in line 1 of α's pseudo-code on sub-stage $[v, \alpha]$, $\psi_\alpha(n)[v, \alpha]\downarrow \le r$, and some strategy δ enumerated n into V on some sub-stage $[u, \delta]$ between $[s, \beta]$ and $[v, \alpha]$. δ must have lower priority than β since this enumeration did not cancel β, and therefore $n \ge (a_0)_\delta[u, \delta] > b$. Since n is chosen to as the least number for which Ψ needs to be corrected, $\Psi_\alpha(b)\downarrow\ [v, \alpha]$. β is waiting for an X change immediately after $[s, \beta^\frown 2]$, so by 1, $\psi_\alpha(b)[v, \alpha] >$. But this is a contradiction because $\psi_\alpha[v, \alpha]$ is a monotonic function.

If $\gamma = \beta^\frown 2$ then let t be least such that $s < t \le u$, β runs on t and

$$X\restriction\psi_\alpha(b)[s, \beta] = X\restriction\psi_\alpha(b)[s, \beta^\frown 2] \neq X\restriction(\psi_\alpha(b)[s, \beta])[u, \beta].$$

β is waiting for an X change for b immediately after $[s, \beta^\frown 2]$, so $\psi_\alpha(b)\uparrow\ [t, \beta]$ or $\psi_\alpha(b) > \theta_f(b)[s, \beta] = \psi_\alpha(b)[t, \beta]$. β will call into$V(b)$ during $[t, \beta]$, so that β has true outcome 1 by Lemma 7, a contradiction. □

With these lemmas complete, we proceed with the proof of Lemma 18.

Proof. To compute whether $n \in X$, find $s > s_0$ such that $\Theta_f(W_\alpha; b)\downarrow = 0[s]$, $\psi_\alpha(X, W_\alpha; b)\downarrow > n[s]$, and $\beta^\frown 2$ is in the secondary holding state at the end of stage s. Then $n \in X \iff n \in X[s]$. In order to confirm that this algorithm works, we must confirm both that for every n there is such an s and that $X(n)[s] = X(n)$ for every such s. If β has outcome 2 on stage $s > s_0$, then $\Theta_f(W_\alpha; a)\downarrow = 0[s]$. Also, since $\beta^\frown 2$ is on the true path, $\beta^\frown 2$ is in the secondary holding state at the end of infinitely many stages. Therefore, to check the first condition above, if suffices to show that if $s > s' > s_0$, $\beta^\frown 2$ runs on both stages s and s', and $\beta^\frown 2$ is in the secondary holding state at the end of both s and s', then $\psi_\alpha(X, W_\alpha; b)[s] > \psi_\alpha(X, W_\alpha; b)[s']$. To see this, note that there must be a least sub-stage $[t, \gamma]$ such that $s > t > s'$ such that $\beta^\frown 2$ is in the primary holding state at the end of sub-stage $[t, \gamma]$ (as usual, $\gamma = \alpha$ or $\gamma = \beta$). By Lemma 21, some number smaller than $\psi_\alpha(b)[t, \gamma]$ enters W_α during $[t, \gamma]$. Since $\psi_\alpha(b)[s]\downarrow$, $\psi_\alpha(b)[s] > \psi_\alpha(b)[s']$.

To finish the lemma, we need to show that when s is as in the first paragraph, $X(n)[s] = X(n)$. This is easy given Lemmas 20 and 21. All that needs to be noted is that for every $s > s_0$ and every γ that executes on s, $\beta^\frown 2$ is either in the primary or secondary holding state after $[s, \gamma]$, and that

1. If $[t, \gamma] > [t', \gamma'] > [s_0, \emptyset]$, $\beta^\frown 2$ enters the primary holding state during sub-stage $[t', \gamma']$ and $\beta^\frown 2$ next enters the secondary holding state during $[t, \gamma]$, then $\psi_\alpha(b)[t, \gamma]\downarrow > \psi_{\alpha'}(b')[t', \gamma']\downarrow$; and

2. If $[t, \gamma] > [t', \gamma'] > [s_0, \emptyset]$, $\beta^\frown 2$ enters the secondary holding state during sub-stage $[t', \gamma']$ and $\beta^\frown 2$ next enters the primary holding state during sub-stage $[t, \gamma]$, then $\psi_{\alpha'}(b')[t, \gamma]\downarrow > \psi_\alpha(b)[t', \gamma']\downarrow$.

The fact that the functions in questions converge follows directly from Lemmas 20 and 21. The rest of 1, for example, is true because, by Lemmas 21, some number smaller than $\psi_\alpha(b)[t', \gamma']$ enters W during $[t', \gamma']$ and $\psi_\alpha(b)$ is subsequently redefined with large use at line 4 of α. □

4 Jump Inversion

If $x \leq 0'$, then x' is obviously REA in $0'$. A converse, Shoenfield's [1959] jump inversion theorem, is that every degree REA in $0'$ is the jump of some $x \leq 0'$. Theorem 4 extends Shoenfield's theorem by showing that in fact x can be chosen to be join independent for the R, α-r.e. degrees over the n-REA degrees. The two following Lemmas will be useful to us.

Lemma 22. *Assume that whenever B is REA in A and $X \oplus A \geq B$, it is the case that $A \geq B$. Then if B is n-REA in A and $X \oplus A \geq B$, then $A \geq B$.*

Proof. This is a simple induction.If $J_1, J_2, \ldots, J_n$ are REA operators, $B = J_1 \circ J_2 \circ \cdots \circ J_n(A)$ and $X \oplus A \geq B$, then $X \oplus A \geq J_1(A)$ from which it follows by induction that $A \geq J_1(A)$. Thus, B has $(n-1)$-REA(A) degree, so $A \geq B$ by induction. □

Lemma 23. *Assume that whenever B is REA in A and $X \oplus A \geq B$, it is the case that $A \geq B$. Then if B is n-r.e. in A and $X \oplus A \geq B$, then $A \geq B$.*

Proof. $A \oplus B$ is n-REA(A) by Theorem 2. Assume that $X \oplus A \geq B$. Then $X \oplus A \geq A \oplus B$ which, by Lemma 22, implies that $A \geq A \oplus B$. It follows that $A \geq B$, as desired. ☐

Theorem 4. *Let y be REA in $0'$ and let R be a univalent, recursively related system of notations for α. There is a Δ^0_2 degree x, such that $x' = y$ and x is join independent for the R, α-r.e. degrees over the n-REA degrees.*

Proof. Let Y be REA in $0'$ and fix some effective (in $0'$) enumeration $Y[s]$, of Y. It is easy to see that we can fix some enumeration, $(S_f)_{f \in \omega}$, of the R, α-r.e. sets so that there is a recursive function which takes f to a $0'$ index for S_f.[2] Let $p : \omega \to \omega^2$ be a recursive bijection, define

$$U_n = W_{p(n)_1}^{S_{p(n)_2}}$$

and let g be recursive such that

$$\forall_n \left(W^{0'}_{g(n)}[s] = U_n[s] \right).$$

This construction is very similar to the proof of Theorem 1, which can be found in Lippe [2000a]. It may be useful to read the descriptive comments given during the proof of that paper's Theorem 2. For this construction, we treat X as a element of ${}^{\omega}2$, rather than as a subset of ω. We build X in stages where each stage is recursive in $0'$. For each stage s, $X[s] \in {}^{<\omega}2$ and $X = \bigcup_s X[s]$. In order to make $X' \geq Y$, we make column q of X co-finite if and only if $q \notin Y$ and column q finite if and only if $q \in Y$. In order give X the required join independence, we try to make X 1-generic over each incomplete R, α-r.e. set, in so far as this is possible given the constraints on X's columns. We accomplish this by having strategy i attempt to ensure that there is some $\sigma \in U_i \cap {}^{<\omega}2$, such that $\sigma \subset X$. Since this construction is a finite injury argument, it is easier to avoid the machinery of independently acting strategies, cancellation, and so on. Instead, we will give an explicit algorithm for the entire construction.

At each stage s we check, for each $q \leq s$, whether $q \in Y[s]$. When strategy i looks in U_i for an extension of $X[s]$, it only considers extensions of $X[s]$ which add ones (zeroes) in column $q < i$ if $q \notin Y[s]$ ($q \in Y[s]$). Any such extension is said to be *acceptable to* $Y[s] \upharpoonright i$. Here is the exact algorithm for stage s of the construction:

1. Let $M = |\{i \leq s | \text{strategy i has not yet acted}\}|$. Denote these strategies, in *decreasing* order, by $(i_{s,t})_{t \in M}$.

2. Let $u_{s,-1}$ be bigger than both s and the portion of oracle $0'$ needed to calculate $Y[s] \upharpoonright s$.

3. Let $n_{s,-1} := 0$.

4. For t from 0 to $M - 1$ do

[2] The skeptical reader is referred to Lippe's [2000a] Corollary 1 for a proof of this fact.

(a) Let

$$n_{s,t} := \max\left\{\mu s'\left(0'\restriction u_{s,t-1}\left[s'\right] = 0'\restriction u_{s,t-1}\right), n_{s,t-1}\right\}.$$

(b) Let $u_{s,t}$ be bigger than the portion of oracle $0'$ used by program $g(i_{s,t})$ to calculate $U_{i_{s,t}}[n_{s,t}]$.

End for loop.

5. If there is a t such that $\exists_{\sigma\supset X[s]}\left(\sigma \in U_{i_{s,t}}[n_{s,t}]\right)$ which is acceptable to $Y[s]\restriction i_{s,t}$, then let t_0 be the greatest such t and let $X[s+1]$ be the least σ witness to the statement for t_0. In this case we say that strategy $i_{s,t}$ has acted and the stage ends. Otherwise, let $X[s+1]$ be the lexicographically least extension of $X[s]$ which is acceptable to $Y[s]\restriction s$ and the stage ends. Note that we allowed the highest priority strategy to act by choosing t_0 large since $(i_{s,t})_{t\in M}$ is a decreasing sequence.

Lemma 24. $X' \geq Y$.

Proof. It suffices to check that column e of X is co-finite if and only if $e \notin Y$ and is finite if and only if $e \in Y$. If $e \notin Y$, then no strategy $i > e$ will ever put a zero in $X^{[e]}$, and each strategy $i \leq e$ acts at most once. Thus, $X^{[e]}$ is co-finite. If, on the other hand, $e \in Y$, then there is an s_0, such that $e \in Y[s_0]$. No strategy $i > \max(e, s_0)$ will put a one in $X^{[e]}$, and each strategy $i \leq \max(e, s_0)$ acts at most once. Thus, $X^{[e]}$ is finite. □

Obviously we can not prove, as in Theorem 1, that X is 1-generic over every incomplete R, α-r.e. degree as every 1-generic is low. What we can prove is that X hits the relevant dense sets for ensuring the necessary join independence.

Lemma 25. *Assume that* $S_{p(i)_2} \ngeq 0'$ *and that*

$$\forall_s\left(\exists_{\sigma\supseteq X[s]}\left(\sigma \in U_i \wedge \sigma \text{ is acceptable to } Y\restriction i\right)\right).$$

Then $\exists_{\sigma\subset X}(\sigma \in U_i)$.

Proof. The proof of this lemma is similar to the proof of Theorem 1, which appears in Lippe [2000a]. Therefore, we will be somewhat terse in our presentation. If i is a least counterexample, assume that all strategies of higher priority have finished acting by stage s_0 and that $Y[s_0]\restriction q = Y\restriction q$. If $s \geq s_0$ and $i = i_{s,t}$, then let

$$r_s = \mu r'\left(\exists_{\sigma\supset X[s]}\left(\sigma \in U_i[r'] \wedge \sigma \text{ is acceptable to } Y[s]\restriction i\right)\right).$$

Then, just as in the proof of Theorem 1, $0'\restriction u_{s,t}$ is enough of $0'$ to reconstruct the s^{th} stage of the construction, and $0'\restriction u_{s,t}[r_s] = 0'\restriction u_{s,t}$. $S_{p(i)_2}$ can compute r_s from $X[s]$ and can compute $X[s+1]$ from $X[s]$ and r_s. Thus, $S_{p(i)_2}$ can compute every r_s. Finally, $0'\restriction s = 0'\restriction s[r_s]$ so that $S_{p(i)_2} \geq 0'$. □

Lemma 26. $Y \geq X'$.

Proof. Fix e_0 such that $S_{e_0} = \emptyset$. In order to decide if $\{e\}^X(e)\downarrow$, Y first computes n such that $p(n)_1 = e_0$ and $U_n = \{\sigma \,|\{e\}^\sigma(e)\downarrow\}$. This can be done because U_n is obviously r.e. Next, Y finds a stage s_0, such that $Y\restriction n[s_0] = Y\restriction n$. By Lemma 25, there is $s_1 > s_0$ such that either

$$X[s_1] \in U_n \text{ or} \tag{2}$$

$$\forall_{\tau \supseteq X[s_1]} (\tau \text{ is acceptable to } Y\restriction n \to \tau \notin U_n)\,. \tag{3}$$

In the first case, $\{e\}^X(e)\downarrow$ and in the second case, $\{e\}^X(e)\uparrow$ since $X[t]$ is always acceptable to $Y\restriction n$ for $t > s_1$. It suffices for Y to be able to find such an s_1 since $\{e\}^X(e)\downarrow$ if and only if $\{e\}^{X[s_1]}(e)\downarrow$. In order to find such an s_1, Y checks every $s > s_0$ in turn, deciding for each whether either (2) or (3) applies. Y can do this because $Y \geq 0'$ by assumption. ☐

To finish the theorem, we need to show that X is join independent for the S,α-r.e. degrees over the n-REA degrees. In fact, we will actually have the somewhat stronger fact that for all R,α-r.e. S_f and all $V \in (n\text{-REA}(S_f) \bigcup n\text{-r.e. }(S_f))$, if $S_f \vee X \geq V$, then $S_f \geq V$. By Lemmas 22 and 23, all of this follows from proving the theorem for $V \in 1\text{-REA}(S_f)$, which we now proceed to do.

Assume that $\Theta(X, S_f) = V$ where V is REA in S_f. We need to show that $S_f \geq V$. If $S_f \equiv 0'$ then we are done. Otherwise, let i be such that $p(i)_2 = f$ and

$$U_i = \left\{\sigma \,\middle|\, \exists_n(\Theta(\sigma, S_f; n)\downarrow = 0 \wedge V(n) = 1)\right\}.$$

By the previous lemma, there is an s_0 such that

$$\forall_{\sigma \supseteq X[s_0]} (\sigma \text{ acceptable to } Y\restriction i \to \sigma \notin U_i)\,.$$

Without loss of generality, choose s_0 large enough that $Y[s_0]\restriction i = Y\restriction i$. Note that for all $n \notin V$, there is $\sigma \supset X[s_0]$ such that σ is acceptable to $Y\restriction i$ and $\Theta(\sigma, S_f; n) = 0$ as witnessed by any sufficiently large initial segment of X. It follows that

$$\forall_n \left(n \notin V \iff \exists_{\sigma \supseteq X[s_0]} (\sigma \text{ is acceptable to } Y\restriction i) \wedge \Theta(\sigma, S_f; n) = 0\right).$$

Thus V is co-r.e. in S_f, which suffices to show that V is recursive in S_f. ☐

5 Minimality

Theorem 5. *Let R be a univalent, recursively related system of notations for α. There is a minimal, Δ^0_2 degree that is join independent for the R,α-r.e. degrees over the n-REA degrees.*

Our construction relies heavily on Soare's [1987] presentation of Sacks's [1961] construction of a minimal degree below $0'$.

Definition 13. *A tree is a partial injective function $T : {}^{<\omega}2 \to {}^{<\omega}2$ such that*

1. *If $T(\sigma)\downarrow$ and $T(\tau)\downarrow$, then $T(\sigma) \supseteq T(\tau)$ iff $\sigma \supseteq \tau$.*

2. *If* $\sigma \supseteq \tau$ *and* $T(\sigma)\downarrow$, *then* $T(\tau)\downarrow$.

3. $T(\sigma^\frown 0)\downarrow$ *iff* $T(\sigma^\frown 1)\downarrow$.

Definition 14. *If T is a tree, then* $\mathrm{Cl}(T)$, *the* closure of T, *is* $\left\{\sigma \mid \exists_{\tau \supseteq \sigma} (\tau \in ran\,(T))\right\}$.

Definition 15. *If T is a tree and $\sigma \in Cl(T)$, then σ* is on T.

Definition 16. *If T is a tree and $X \in {}^{\omega}2$ then X is a* branch *of T if for all $\sigma \supset X$, σ is on T. $[T]$ denotes the set of branches of T.*

Definition 17. *ρ, τ are an e-*splitting *of σ if $\rho, \tau \supseteq \sigma$, $T(\rho)\downarrow$, $T(\tau)\downarrow$ and there is an n such that $\Psi_e\,(T(\tau); n)\downarrow \neq \Psi_e\,(T(\rho); n)\downarrow$. σ e-splits on T if there are ρ, τ as above.*

Definition 18. *T is an e-splitting tree if $\forall_\sigma (T(\sigma^\frown 0)\downarrow \rightarrow \sigma^\frown 0, \sigma^\frown 1$ e-split σ on $T)$.*

Definition 19. *If T is a partial recursive tree and σ is on T, then Split(e, T, σ), the* e-splitting subtree of T above σ, *is inductively defined as follows:*

- *Let τ be the first element of dom(T), such that $T(\tau) \supseteq \sigma$, discovered in some canonical search for such elements. Split$(e, T, \sigma)(\emptyset) = T(\tau)$.*

- *If Split$(e, T, \sigma)(\rho)\downarrow = T(\tau)$ and there is no e-splitting of τ on T, then we let Split$(e, T, \sigma)(\tau^\frown 0)\uparrow$ and Split$(e, T, \sigma)(\tau^\frown 1)\uparrow$. Otherwise, let τ_0, τ_1 be the first e-splittings of τ discovered in some canonical search for such splittings. Without loss of generality, assume that τ_0 is lexicographically less than τ_1. We let Split$(e, T, \sigma)(\rho^\frown 0) = T(\tau_0)$ and Split$(e, T, \sigma)(\rho^\frown 1) = T(\tau_1)$.*

Definition 20. *If T is a partial recursive tree and σ is on T, then Full(T, σ), the* full subtree of T above σ, *is defined as follows:*

- *Let τ be the first element of dom(T), such that $T(\tau) \supseteq \sigma$, discovered in some canonical search for such elements.*

- *Let Full$(T, \sigma)(\rho) = T(\tau^\frown \rho)$.*

Note that the two previous definitions are a bit sloppy as they depend on making a choice for the aforementioned canonical searches. But we can easily fix recursive functions that compute Σ_1 indices for such searches from a Σ_1 index for T. And therefore we can recursively compute indices for Split(e, T, σ) and Full(T, σ) from an index for T. In this case, Split and Full are actually functions of indices and not of partial recursive functions. However, we will not split hairs over our splitting trees. The pseudo-code that follows is written as if certain variables take on trees as values, but all the computation is done with indices.

Definition 21. *X meets M_e, the e^{th}* minimality requirement *if there is a partial recursive tree T, such that $X \in [T]$ and either*

1. *T is an e-splitting tree or*

2. *There is σ with no e-splittings on T such that $T(\sigma)\downarrow \subset X$.*

It is well known that if X meets requirement M_e, then either $\Psi_e(X) \equiv X$ or $\Psi_e(X) \equiv 0$. Furthermore, Posner showed that if X meets all M_e requirements, then $X \not\equiv 0$. So the M_e requirements together ensure that X is minimal.

Familiarity with the proof that there is a minimal degree below $0'$ is useful for understanding the following proof. At each stage, s, of the construction, there are $s+1$ many trees, $(T_{s,q})_{q \leq s}$. We arrange the construction so that, for each q, $T_q = \lim_s T_{s,q}$ exists and witnesses the fact that X meets the q^{th} minimality requirement. We essentially accomplish this by finite injury on the T_q. We start by guessing that T_q will be a q-splitting tree, but change our minds if and when we see that $X[s]$ has no more q-splits on $T_{s-1,q}$. Recall that in the proof of Theorem 4, strategy i is constrained to look for extensions of $X[s]$ in U_i that are acceptable to $Y[s]{\upharpoonright}i$. In this construction, strategy i is constrained to look for extensions of $X[s]$ that are in $\bigcap_{q \leq i} \mathrm{Cl}\left(T_{s,q}\right)$. Since, in fact, we arrange that $\mathrm{Cl}\left(T_{s,q}\right) \supseteq \mathrm{Cl}\left(T_{s,q+1}\right)$, it suffices to look for extensions that are on $T_{s,i}$.

To make the bookkeeping cleaner in the following construction, assume that $T_{0,0} = \mathrm{id}$ and that the construction starts at stage 1 instead of stage 0. The pseudo-code for stage s of the construction is:

1. Let k_s be greatest so that $0 \leq k_s < s$ and $X[s] \in \mathrm{Cl}\left(T_{s-1,k_s}\right)$ (k_s exists since $T_{s-1,0} = \mathrm{id}$).
2. $T_{s,q} := T_{s-1,q}$ for $0 \leq q \leq k_s$.
3. If $k_s = s-1$ then $T_{s,s} := \mathrm{Split}(s-1, T_{s,s-1}, X[s])$.
4. If $k_s < s-1$ then $T_{s,k_s+1} := \mathrm{Full}(T_{s,k_s}, X[s])$ and $T_{s,q} := \mathrm{Split}(q-1, T_{s,q-1}, X[s])$ for $k_s + 1 < q \leq s$.
5. Let $M = |\{i \leq s | \text{strategy i has not yet acted}\}|$. Denote these strategies, in *decreasing* order, by $(i_{s,t})_{t \in M}$.
6. Let $u_{s,-1}$ be bigger than any of: s; the use on oracle $0'$ for determining k_s; the use on oracle $0'$ for finding q in line 10; and the use on oracle $0'$ and the portion of oracle $0'$ needed to calculate $Y[s]{\upharpoonright}s$.
7. Let $n_{s,-1} := 0$.
8. For t from 0 to $M-1$ do

   (a) Let
   $$n_{s,t} := \max\left\{\mu s' \left(0'{\upharpoonright}u_{s,t-1}\left[s'\right] = 0'{\upharpoonright}u_{s,t-1}\right), n_{s,t-1}\right\}.$$

   (b) Let $u_{s,t}$ be bigger than both: the use on oracle $0'$ by program $g(i_{s,t})$ calculating $U_{i_{s,t}}[n_{s,t}]$; and the use on oracle $0'$ needed to check, for each $\sigma \in U_{i_{s,t}}[n_{s,t}]$, whether or not $\sigma \in \mathrm{Cl}\left(T_{s,i_{s,t}}\right)$.

   End 'for loop'.
9. If there is a $t < M$ such that $\exists_{\sigma \supset X[s]}\left(\sigma \in U_{i_{s,t}}[n_{s,t}]\right) \bigcap \mathrm{Cl}\left(T_{s,i_{s,t}}\right)$ and strategy $i_{s,t}$ has not acted, then let t_0 be the greatest such t and

(a) Let $X[s+1]$ be the least σ witness to the statement for t_0.

(b) Strategy i_{s,t_0} has acted.

(c) For every $q \in [i_{s,t_0}+1, s]$ (taken in order), let

$$T_{s,q} := \text{Split}(q-1, T_{s,q-1}, X[s+1]).$$

Note that we may be changing each of these $T_{s,q}$ from how they were defined in lines 1-4. This line essentially cancels lower priority minimality strategies because i_{s,t_0} has acted.

10. Else let $q \le s$ be greatest such that $\text{Cl}\left(T_{s,q}\right)$ contains a proper extension of $X[s]$ and let $X[s+1]$ be the first such extension found in the standard enumeration of $\text{Cl}\left(T_{s,q}\right)$. Note that q exists since $T_{s,0} = \text{id}$ and there is indeed a unique standard enumeration since our calculations with trees are really calculations with r.e. indices for trees.

Lemma 27. *For every q, there is a stage s_q, such that $k_s \ge q$ whenever $s > s_q$.*

Proof. This is proven inductively. For $q = 0$, we see that $k_s \ge 0$ for all s because $T_{s,0} = \text{id}$ contains extensions of all strings. Assume it is true for every $r \le q$. Let t be bigger than $q+1$, the last s such that $k_s < q$ and the last stage during which T_{q+1} is redefined at line 9c. Either there is a stage $s > t$, such that $T_{s,q+1}$ contains no extensions of $X[s]$, or there is not. If there is not, then $k_s \ge q+1$ for all $s > t$. If there is, call it s_{q+1}. We claim that $k_s \ge q+1$ for all $s > s_{q+1}$. To see this, note that $T_{s_{q+1},q+1} = \text{Full}\left(T_{s_{q+1},q}, X[s_{q+1}]\right)$. Thus, $T_{s_{q+1},q+1}$ contains extensions of every $X[s]$ since $T_{s_{q+1},q}$ does. Furthermore, by choice of t, $T_{s,q+1}$ never gets redefined at line 9c for $s > s_{q+1} > t$. Therefore, $T_{s,q+1} = T_{s_{q+1},q+1}$ for $s \ge q+1$ and $k_s \ge q+1$ for all $s > s_{q+1}$. □

Lemma 28. *For every q, there is a stage s_q, such that $T_{s,q} = T_{s_q,q}$ whenever $s > s_q$.*

Proof. This follows trivially from the previous lemma and the fact for each q, there are only finitely many stages s, such that $T_{s,q}$ gets redefined at line 9c during stage s. □

For the remainder of this section, let $T_q = \lim_s T_{s,q}$ and let s_q be the *least* stage satisfying $k_s > q$ and $T_{s,q} = T_{s_q,q}$ whenever $s > s_q$.

Lemma 29. *X satisfies requirement M_q.*

Proof. Not surprisingly, this is witnessed by T_q. It follows directly from the definition of k_s in line 1 that for $s > s_q$, $X[s] \in \text{Cl}\left(T_{s,q} = T_q\right)$. Therefore, $X \in \left[T_q\right]$. If $s_q = s_{q-1}$, $s_q = q$ or s_q is the last stage during which T_q is redefined at line 9c, then

$$T_q = T_{s_q,q} = \text{Split}\left(q-1, T_{s_q,q-1} = T_{q-1}, X[s_q]\right),$$

so that T_q is a $q-1$ splitting tree. Otherwise, $s_q > s_{q-1}$, $s_q > q$ and s_q is greater than the last stage during which T_q is redefined at line 9c. It follows that

$$T_{s_q-1,q} = \text{Split}\left(q-1, T_{s_q-1,q-1} = T_{q-1}, X[s_{(q-1)}]\right),$$

and that $T_{s_q-1,q}$ contains no extensions of $X[s_q]$. But $X[s_q]$ is on T_{q-1}, so that T_{q-1} contains no $q-1$ splittings of $X[s_q]$. Since

$$T_q = T_{s_q,q} = \text{Full}\left(T_{q-1}, X[s_q]\right),$$

T_q contains no q splits above $X[s_q]$, and M_q is indeed satisfied by T_q. $\square$

Obviously we can not prove, as in Theorem 1, that X is 1-generic over every incomplete R, α-r.e. degree as no 1-generic is minimal. What we can prove is that X hits the relevant dense sets for ensuring the necessary join independence.

Lemma 30. *Assume that* $S_{p(i)_2} \not\geq 0'$ *and that*

$$\forall_s \left(\exists_{\sigma \supseteq X[s]} (\sigma \in Cl(T_i) \cap U_i)\right).$$

Then $\exists_{\sigma \subset X} (\sigma \in U_i)$.

Proof. The proof of this lemma is similar to the proof of Theorem 1, which appears in Lippe [2000a]. If i is a least counterexample, assume that all strategies of higher priority have finished acting by stage s^* and that $T_i = T_{s,i}$ for all $s > s^*$. If $s \geq s^*$ and $i = i_{s,t}$, then let

$$r_s = \mu r' \left(\exists_{\sigma \supset X[s]} \left(\sigma \in \text{Cl}(T_i)\left[r'\right] \cap U_i\left[r'\right]\right)\right).$$

Then $0' \restriction u_{s,t}$ is enough of $0'$ to reconstruct the s^{th} stage of the construction, and $0' \restriction u_{s,t}[r_s] = 0' \restriction u_{s,t}$. As before, S_f can compute r_s from $X[s]$ and can compute $X[s+1]$ from $X[s]$ and r_s. Thus, S_f can compute every r_s. Finally, $0' \restriction s = 0' \restriction s[r_s]$ so that $S_f \geq 0'$. $\square$

To finish the theorem, we need to show that X is join independent for the R, α-r.e. degrees over the n-REA degrees. In fact, we will actually have the somewhat stronger fact that for all R, α-r.e. S_f and all $V \in (n\text{-REA}(S_f) \bigcup n\text{-r.e. }(S_f))$, if $S_f \vee X \geq V$, then $S_f \geq V$. By Lemmas 22 and 23, all of this follows from proving the theorem for $V \in 1\text{-REA}(S_f)$, which we now proceed to do.

Assume that $\Theta(X, S_f) = V$ where V is REA in S_f. We need to show that $S_f \geq V$. If $S_f \equiv 0'$ then we are done. Otherwise, let i be such that $p(i)_2 = f$ and

$$U_i = \left\{\sigma \,\middle|\, \exists_n \left(\Theta\left(\sigma, S_f; n\right)\downarrow = 0 \wedge V(n) = 1\right)\right\}.$$

By the previous lemma, there is an $s^* > s_i$ such that

$$\forall_{\sigma \supseteq X[s^*]} (\sigma \in \text{Cl}(T_i) \rightarrow \sigma \notin U_i).$$

Note that for all $n \notin V$ there is $\sigma \supset X[s^*]$ such that $\sigma \in \text{Cl}(T_i)$ and $\Theta(\sigma, S_f; n) = 0$ as witnessed by any sufficiently large initial segment of X. It follows that

$$\forall_n \left(n \notin V \iff \exists_{\sigma \supseteq X[s^*]} \left(\sigma \in \text{Cl}(T_i) \wedge \Theta(\sigma, S_f; n) = 0\right)\right).$$

Thus V is co-r.e. in S_f, which suffices to show that V is recursive in S_f.

References

Cooper, B. S. [1971]. PhD thesis, University of Leicester.

Epstein, R., Haas, R. and Kramer, R. [1981]. Hierarchies of sets and degrees below $0'$, *Logic Year 1979-1980: University of Connecticut*, number 859 in *Lecture Notes in Mathematics*, Springer-Verlag, pp. 32–48.

Harrington, L. [1978]. Plus cupping in the recursively enumerable degrees, Handwritten notes.

Jockusch, C. G. and Shore, R. A. [1984]. Pseudo-jump operators II: transfinite iterations, hierarchies and minimal covers, *Journal of Symbolic Logic* **49**: 1205–1236.

Lippe, D. A. [2000a]. Relative genericity in the Δ^0_2 degrees, manuscript.

Lippe, D. A. [2000b]. Substructures of Δ^0_2 degrees, in preparation.

Miller, D. [1981]. High recursively enumerable degrees and the anti-cupping property, *Logic year 1979-1980: University of Connecticut*, number 859 in *Lecture Notes in Mathematics*, Springer-Verlag.

Posner, D. and Robinson, R. W. [1981]. Degrees joining to $0'$, *Journal of Symbolic Logic* **46**: 714–722.

Rogers, H. [1967]. *Theory of Recursive Functions and Effective Computability*, McGraw-Hill.

Sacks, G. E. [1961]. A minimal degree less than $0'$, *Bulletin of American Mathematical Society* **67**: 416–419.

Shoenfield, J. R. [1959]. On degrees of unsolvability, *Annals of Mathematics* **69**: 644–653.

Slaman, T. A. and Steel, J. R. [1989]. Complementation in the Turing degrees, *Bulletin of Symbolic Logic* **54**: 160–176.

Soare, R. I. [1987]. *Recursively Enumerable Sets and Degrees*, Springer-Verlag.