The Decidability of the Existential Theory of the Poset of Recursively Enumerable Degrees with Jump Relations

STEFFEN LEMPP[1] AND MANUEL LERMAN[2,3]

Department of Mathematics, University of Wisconsin,
Madison, WI 53706, USA
lempp@math.wisc.edu

Department of Mathematics, University of Connecticut
Storrs, CT 06269, USA
mlerman@uconnvm.uconn.edu

Abstract. We show that the existential theory of the recursively enumerable degrees in the language L containing predicates for order and n-jump comparability for all n, and constant symbols for least and greatest elements, is decidable. The decidability follows from our main theorem, where we show that any finite L-structure which is consistent with the order relation, the order-preserving property of the jump operator, and the property of the jump operator that the jump of an element is strictly greater than the element, can be embedded into the r.e. degrees.

1 Research partially supported by National Science Foundation Grants DMS-8701891, DMS-8901529, DMS-9100114, U.S.-W. Germany Binational Grant INT-8722296, and post-doctoral fellowships of the Deutsche Forschungsgemeinschaft and the Mathematical Sciences Research Institute.

2 Research partially supported by National Science Foundation Grants DMS-8521843, DMS-8900349, and DMS-9200539, and by the Mathematical Sciences Research Institute.

3 The authors wish to express their gratitude to C. Ash, M. Groszek, L. Harrington, J. Knight, A. Kucera, R. Shore, and T. Slaman for helpful comments during the seven years in which this research was being carried out.

TABLE OF CONTENTS

0. Introduction. Decidability and undecidability of (fragments of) elementary theories of recursion-theoretic structures have been central topics of research in recursion theory for more than two decades. Results of this nature have been obtained by Lachlan [La1], [La2], Simpson [Si], Herrmann [He], Harrington (unpublished), Lerman and Soare [LrSo], Schmerl (cf. [Lr1]), Epstein [Ep], Shore [Sh1], [Sh2], Lerman [Lr1], Lerman and Shore [LrSh], Sacks [Sa1], Harrington and Slaman [HaSl] (cf. [SlWo]), Jockusch and Soare (unpublished, cf. [Lr1]), Jockusch and Slaman [JoSl] and Hinman and Slaman [HiSl]. Sharp results have been obtained for the poset of degrees $\mathcal{D} = \langle \mathbf{D}, \le \rangle$. In this case, Lachlan [La1] showed that Th($\mathcal{D}$) is undecidable, and Simpson [Si] showed that this theory is recursively isomorphic to second order arithmetic; Shore [Sh1] and Lerman [Lr1] showed that $\forall\exists\cap$Th($\mathcal{D}$) (the $\forall\exists$-fragment of the elementary theory of $\mathcal{D}$) is decidable, and this decidability result has been extended by Jockusch and Slaman [JoSl] to $\forall\exists\cap$Th($\langle \mathbf{D}, \le, \cup \rangle$). Schmerl (cf. [Lr1]) showed that $\exists\forall\exists\cap$Th($\mathcal{D}$) is undecidable. Sharp results have also been obtained for the elementary theory of the poset $\mathcal{D}[\mathbf{0},\mathbf{0}'] = \langle \mathbf{D}[\mathbf{0},\mathbf{0}'], \le \rangle$ of the degrees below $\mathbf{0}'$. Epstein [Ep] and Lerman [Lr1] showed that Th($\mathcal{D}[\mathbf{0},\mathbf{0}']$) is undecidable and Shore [Sh2] showed that this theory has degree $\mathbf{0}^{(\omega)}$; Lerman and Shore [LrSh] showed that $\forall\exists\cap$Th($\mathcal{D}[\mathbf{0},\mathbf{0}']$) is decidable, while Schmerl (cf. [Lr1]) showed that $\exists\forall\exists\cap$Th($\mathcal{D}[\mathbf{0},\mathbf{0}']$) is undecidable. Gaps in our knowledge remain for other structures.

There are some natural operations on degree structures which motivate the study of decidability in languages other than the language of posets. Most degree structures are uppersemilattices and so support a *join operator* $\cup$. The join operator is definable from $\le$ by an $\forall$-predicate. Hence $\exists\cap$Th($\langle \mathbf{D}[\mathbf{0},\mathbf{0}'], \le, \cup \rangle$) is decidable. $\mathcal{D}$ also supports the jump operator, ', an order-preserving function of one variable on degrees such that $\mathbf{a}' > \mathbf{a}$ for all $\mathbf{a}$. Cooper [Co1, Co2] has shown that the jump operator is definable over $\mathcal{D}$, but Lerman and Shore [LrSh] have shown that the definition cannot be an $\forall\exists$-definition. Thus the study of the elementary theory of $\mathcal{D}' = \langle \mathbf{D}, \le, ' \rangle$ is more complex than the study of Th($\mathcal{D}$). The results of Lachlan [La1] and Simpson [Si] cover Th($\mathcal{D}'$) as well; thus Th($\mathcal{D}'$) is undecidable, and is recursively isomorphic to second-order arithmetic. On the other hand, Jockusch and Soare (cf. [Lr1]) have shown that Th(⟨**D,'**⟩) is decidable.

In this paper, we develop methods which enable us to build configurations of recursively enumerable degrees while simultaneously controlling the configurations of the mth-jumps of these degrees. Our results will yield a decision procedure for the existential theory of the recursively enumerable degrees in an expanded language, and can be used to give a decision procedure for a fragment of ∃∩Th(⟨**D**,≤,**'**,**0**⟩). We expect the methods introduced in this paper to be useful in providing a decision procedure for ∃∩Th(⟨**D**,≤,**'**,**0**⟩). (Hinman and Slaman [HiSl] have recently proved that ∃∩Th(D') is decidable using a forcing argument.)

For every degree **a**, we let R(**a**) = ⟨**R(a)**,≤⟩ be the poset of degrees r.e. in **a**, and we set R = R(**0**). For each m ∈ **N** (**N** is the set of natural numbers) and **a,b** ∈ **R**, we define

$$\mathbf{a} \leq_m \mathbf{b} \iff \mathbf{a}^{(m)} \leq \mathbf{b}^{(m)}.$$

A *jump poset* is a 5-tuple ⟨P,≤,P',≤',f⟩, such that ⟨P,≤⟩ and ⟨P',≤'⟩ are posets of cardinality ≥ 2 with least and greatest elements, and f is an order-preserving map from P onto P'. An *m-jump poset* is a structure

$$\mathsf{P} = \langle P_0,\leq_0,P_1,\leq_1,f_1,\cdots,P_m,\leq_m,f_m\rangle$$

such that for each k < m, $\langle P_k,\leq_k,P_{k+1},\leq_{k+1},f_{k+1}\rangle$ is a jump poset. We define a <ω-jump poset analogously.

We now state our main theorem.

Theorem 7.8: Fix n ∈ **N**, and let $\mathsf{P} = \langle P_0,\leq_0,P_1,\leq_1,f_1,\ldots,P_m,\leq_m,f_m\rangle$ be a finite m-jump poset such that P_0 has least element 0 and greatest element 1. Then there is a finite set $\mathbf{G}_0$ of r.e. degrees, and there are finite sets $\mathbf{G_k} = \{\mathbf{d}: \exists \mathbf{a}\in\mathbf{G}_0(\mathbf{a}^{(k)} = \mathbf{d})\}$ for each k ∈ [1,m] such that the following diagram of Figure 1 commutes. Furthermore, the embedding maps 0 ∈ P_0 to **0** and 1 ∈ P_0 to **0'**. (In fact, the proof of Theorem 9.9 can easily be extended to countable <ω-jump posets.) ∎

We specify a finite set of axioms for m-jump posets. These axioms assert that $\langle P_i,\leq_i\rangle$ is a poset for each i ≤ m, 0 is the least element of P_0, 1 is the greatest element of P_0, and f_i is a surjective order-preserving map from $\langle P_{i\pm1},\leq_{i\pm1}\rangle$ onto $\langle P_i,\leq_i\rangle$. Given any existential sentence in the language $\mathsf{L} = \langle 0,1,\leq_0,\leq_1,\ldots,\leq_m,...\rangle$, the sentence asserts that one of a finite number of diagrams is consistent with the axioms of m-jump posets, i.e., can be embedded into an m-jump poset. We can recursively determine whether or not one of these

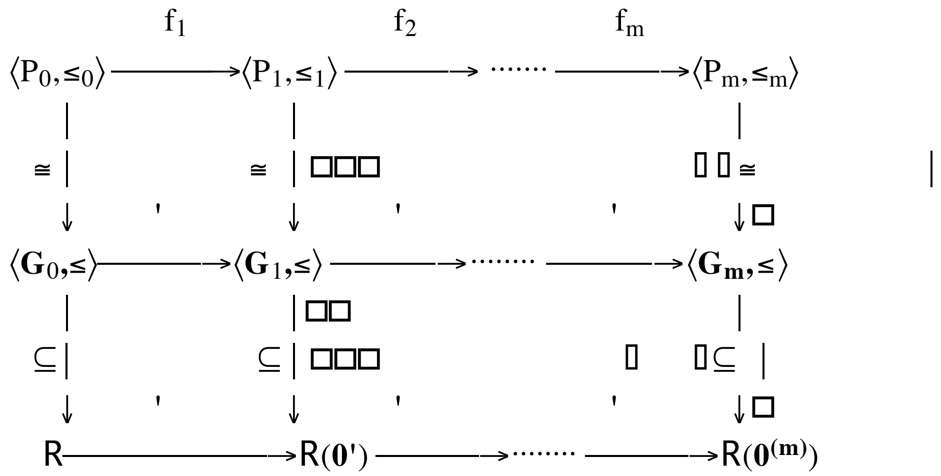

Figure 1

diagrams is consistent. If not, then the sentence is false; if so, then by Theorem 7.8, the sentence is true. Furthermore, this process is uniform in m, and mentions only finitely many relations $\leq_k$. Hence:

Corollary 7.9: The existential theory of $\mathsf{R}^{(<\omega)} = \langle \mathbf{R},\mathbf{0},\mathbf{0}',\leq,\leq_1 \ldots,\leq_m,\ldots\rangle$ is decidable. n

This corollary extends the result in [LmLr2] where it is shown that the existential theory of $\langle \mathbf{R},\mathbf{0},\mathbf{0}',\leq,\leq_1\rangle$ is decidable.

Fix m. The simplest sentences of this existential theory in the language L require us to construct a degree **a** such that $\mathbf{0}^{(k)} < \mathbf{a}^{(k)} < \mathbf{0}^{(k+1)}$ for all $k \leq m$. The Sacks Jump Inversion Theorem [Sa2] allows us to construct such degrees. One begins with a degree $\mathbf{d}_m$ such that $\mathbf{0}^{(m)} < \mathbf{d}_m < \mathbf{0}^{(m+1)}$ and $\mathbf{d}_m$ is r.e. in $\mathbf{0}^{(m)}$. The jump inversion theorem is now applied to obtain a degree $\mathbf{d}_{m-1}$ which is r.e. in $\mathbf{0}^{(m-1)}$ such that $(\mathbf{d}_{m-1})' = \mathbf{d}_m$ and $\mathbf{0}^{(m-1)} < \mathbf{d}_{m-1} < \mathbf{0}^{(m)}$. This procedure can be iterated, producing $\mathbf{d}_0 = \mathbf{a}$. Attempts were made to decide the full ∃-Theory in this way, but the Shore Non-Inversion Theorem [Sh3] showed that such attempts were doomed to failure. Our approach is to construct the r.e. degrees directly. To do this, we introduce a $\mathbf{0}^{(n)}$-priority argument for each $n \in \mathbf{N}$. Priority arguments of this sort were developed by Ash [A1,A2] and Knight [Kn] for recursive model theory, but their approach does not seem to be applicable here. Groszek and Slaman [GS] have been developing a different general framework for $\mathbf{0}^{(n)}$-priority arguments, and our proof has been influenced by the techniques of Ash, Groszek and Slaman. In particular, although our trees are different from those used by Ash, the properties used in the tree decomposition are, in many cases, based on ideas introduced in Ash [A1]. A framework is also being developed by Kontosthatis [Ko1,Ko2,Ko3]. Other theorems proved using our framework can be found in [LmLr1], [LmLr2], and [LLW], and an overview of the framework is presented in [LmLr3]. Our treatment of individual requirements is modeled after the solution to the "deep degree" problem by Lempp and Slaman [LmSl].

We use the following notation. If $A \subseteq \mathbf{N}$, then we let $\overline{A}$ denote the complement of A. For $A,B \subseteq \mathbf{N}$, we let $A\backslash B$ denote the difference of A and B. Given a set P, we let $|P|$ denote the cardinality of P. A k-dimensional space is a set $S = \{\bar{a}\}\times\mathbf{N}^k\times\{\bar{b}\}$ for some choice of finite sequences $\bar{a}$ and $\bar{b}$ of elements of $\mathbf{N}$; in this case, we write $\dim(S) = k$. If $A = \{\bar{a}\}\times\mathbf{N}^k\times\{\bar{b}\}$ and $i \in \mathbf{N}$, then we let $A^{[i]}$ denote $\{\langle\bar{a},i,\bar{x},\bar{b}\rangle: \bar{x} \in \mathbf{N}^{k-1}\}\cap A$ and call $A^{[i]}$ a *section* of A.

We depart a little from the standard classification of sentences, although our classification is equivalent to the standard classification. Thus a Σ_0- or Π_0-formula is one in which all quantifiers are bounded. A Σ_m-formula is one of the form $Q_1\bar{x}_1\ldots Q_k\bar{x}_k\exists\bar{y}R(\bar{x}_1,\ldots,\bar{x}_k,\bar{y})$, where each $Q_i\bar{x}_i$ is a finite block of bounded universal quantifiers or a finite block of bounded existential quantifiers and R is a $\Pi_{m\pm1}$-formula. Similarly, a Π_m-formula is one of the form $Q_1\bar{x}_1\ldots Q_k\bar{x}_k\forall\bar{y}R(\bar{x}_1,\ldots,\bar{x}_k,\bar{y})$, where each $Q_i\bar{x}_i$ is a finite block of bounded universal quantifiers or a finite block of bounded existential quantifiers and R is a $\Sigma_{m\pm1}$-formula.

Let γ be a Σ_m- or Π_m-sentence. Then γ can either be written as $\overline{Q}\bar{x}\exists\bar{y}\delta(\bar{x},\bar{y})$ where δ is Π_{m-1}, or as $\overline{Q}\bar{x}\forall\bar{y}\delta(\bar{x},\bar{y})$ where δ is $\Sigma_{m\pm1}$. The formula $\gamma^{[z]}$ is obtained from γ by replacing the first block of unbounded quantifiers $\exists\bar{y}$ or $\forall\bar{y}$ with a similar block where all variables are restricted to numbers $\leq z$.

A *string* is a finite sequence of letters from an alphabet. If S is an alphabet, we let $S^{<\omega}$ be the set of all strings from S. We write $\sigma \subset \tau$ if τ properly extends σ, and $\sigma|\tau$ if σ and τ are incomparable. We say that σ *lies along* τ if $\sigma \subseteq \tau$. For $\sigma,\tau \in S^{<\omega}$, we let $lh(\sigma)$ denote the cardinality of the domain of σ. If $\sigma \neq \langle\,\rangle$ (the empty string), then σ^- is the unique $\tau \subset \sigma$ such that $lh(\tau) = lh(\sigma)-1$. We define the string $\sigma\hat{}\tau$ by

$$\sigma\hat{}\tau(x) = \begin{cases} \sigma(x) & \text{if } x < lh(\sigma) \\ \tau(x-lh(\sigma)) & \text{if } lh(\sigma) \leq x < lh(\sigma)+lh(\tau). \end{cases}$$

If $x \leq lh(\sigma)$, then $\sigma\restriction x$, *the restriction of* σ *to x,* is the string τ of length x such that $\tau(y) = \sigma(y)$ for all $y \leq x$. Restriction is defined similarly for infinite sequences from an alphabet. We also use interval notation for strings. Thus $[\sigma,\tau] = \{\rho: \sigma \subseteq \rho \subseteq \tau\}$. $\sigma\wedge\tau$ denotes the longest ρ such that $\rho \subseteq \sigma,\tau$, and if σ and τ are comparable, then $\sigma\vee\tau$ is the longer of σ and τ.

A *tree* is a set of strings which is closed under restriction. The *paths* through a tree T are the infinite sequences Λ such that $\Lambda\restriction x \in T$ for all $x \in \mathbf{N}$. We let [T] denote the set of paths through T.

The *high/low hierarchy* for $\mathcal{R}$ is defined as follows. For $n \geq 0$, we say that **a** is *low*$_n$ ($\mathbf{a} \in \mathbf{L}_n$) if $\mathbf{a}^{(n)} = \mathbf{0}^{(n)}$, and **a** is *high*$_n$ ($\mathbf{a} \in \mathbf{H}_n$) if $\mathbf{a}^{(n)} = \mathbf{0}^{(n+1)}$. If $\mathbf{0}^{(n)} < \mathbf{a}^{(n)} < \mathbf{0}^{(n+1)}$ for all n, then we say that **a** is *intermediate*.

$\langle \Phi^k_e : e \in \mathbf{N}\rangle$ will be the standard enumeration of all partial recursive functionals of k variables. (We will frequently suppress the superscript, writing Φ_e for Φ^k_e.) Thus $\Phi^k_e(A;x_1,\cdots,x_k) = y$ if the eth partial recursive functional of k variables, computing from oracle A and input $x_1,\cdots,x_k$, outputs the value y. For each $e,k \in \mathbf{N}$, we will have a recursive approximation $\langle \Phi^k_{e,s} : s \in \mathbf{N}\rangle$ to Φ^k_e. We say that $\Phi^k_{e,s}(A;x_1,\ldots,x_k)\downarrow$ if we obtain an output from this computation in fewer than s steps; otherwise, $\Phi^k_{e,s}(A;x_1,\ldots,x_k)\uparrow$. If $\Phi^k_{e,s}(A;x_1,\ldots,x_k)\downarrow$, then we let the *use* of this computation be the greatest element u for which a question "$u \in A$?" is asked of the A oracle during the computation. We will work under the convention that:

(0.1) If u is the use of a computation at stage s, then $u < s$.

We will be constructing partial recursive functionals within a recursive construction by *declaring axioms* $\Delta(\sigma;\bar{x}) = y$ to reflect the fact that the partial recursive functional Δ with input $\bar{x}$ produces output y when computing from any oracle σ, (so from any oracle $A \supset \sigma$). If $\bar{x} = \langle x_1,\ldots,x_m\rangle$ and $\bar{z} = \langle z_{m+1},\ldots,z_k\rangle$, then $\lim_{\bar{x}} \Phi^k_e(A;\bar{x},\bar{z})$ denotes $\lim_{x_1}\ldots\lim_{x_m}\Phi^k_e(A;x_1,\ldots,x_m,z_{m+1},\ldots,z_k)$. Other notation follows [So].

1. The Basic Modules. We will introduce the basic modules for requirements of dimensions 1 and 2 in this section. While the proof of the theorem will need requirements of higher dimensions, the descriptions of the basic modules for these higher dimension requirements is similar to the descriptions for requirements of dimensions 1 and 2, requiring only more iterations of the limit process.

Fix a finite m-jump poset $\mathsf{P} = \langle P_0,\leq_0,P_1,\leq_1,f_1,\cdots,P_m,\leq_m,f_m\rangle$, and assume that P_0 has least element 0 and greatest element 1. Let g_0 be the identity function, and for each $k \in [1,m]$, let $g_k = f_k \circ f_{k-1} \circ \ldots \circ f_1$. Assume, also, that there are $d \neq \tilde{d} \in P_0$ with the following properties:

(1.1) For all $k \leq m$, $g_k(0) <_k g_k(d) <_k g_k(1)$ and for all $c \in P_0 \setminus \{d\}$ such that $g_k(0) <_k g_k(c) <_k g_k(1)$, $g_k(d)$ and $g_k(c)$ are incomparable.

(1.2) For all $k \le m$, $g_k(0) <_k g_k(\tilde{d}) <_k g_k(1)$ and for all $c \in P_0 \backslash \{\tilde{d}\}$ such that $g_k(0) <_k g_k(c) <_k g_k(1)$, $g_k(\tilde{d})$ and $g_k(c)$ are incomparable.

These conditions will reduce the number of different types of requirements needed for our construction. In particular, we will not have to treat, as special cases, requirements to make any of the sets which we are constructing non-low_k or non-high_k for any $k \le m$.

We will construct an r.e. set A_b for each $b \in P_0$. We specify that $A_0 = \emptyset$ and A_1 is the complete r.e. set K. We will also be constructing partial recursive functionals Δ with subscripts and superscripts designating the requirement for which Δ is acting.

Definition 1.1: Uniformly in $A \subseteq \mathbf{N}$ and $e,r \in \mathbf{N}$, we fix a sentence $\beta_r(A;e)$ which is Σ_{r+1} and whose validity agrees with that of $e \in A^{(r)}$ if r is odd, and which is Π_{r+1} and whose validity agrees with that of $e \notin A^{(r)}$ if r is even. n

For all $b,c \in P_0$ and $k \le m$, we will have to show that

$$g_k(b) \le g_k(c) \Leftrightarrow A_b^{(k)} \le_T A_c^{(k)}.$$

Each such equivalence will be satisfied if we satisfy the following conditions for sufficiently many b and c: There is a partial recursive functional Δ (depending on the condition) such that for all $e \in \mathbf{N}$:

(1.3) $R^{0,k+1}_{e,b,c}$: $g_k(c) \not\le g_k(b) \Rightarrow \Delta(A_c)$ is total & $\forall x(\lim_{u_1}\ldots\lim_{u_k}\Delta(A_c;u_1,\ldots,u_k,x)\downarrow)$ & $\exists x(\lim_{u_1}\ldots\lim_{u_k}\Delta(A_c;u_1,\ldots,u_k,x)\downarrow \ne \lim_{v_1}\ldots\lim_{v_k}\Phi^{k+1}_e(A_b;v_1,\ldots,v_k,x))$.

(1.4) $R^{1,k}_{e,b,c}$: $g_k(b) \le g_k(c)$ & $b \ne 1 \Rightarrow \Box \forall e \exists q \le 1((\lim_{u_1}\ldots\lim_{u_k}\Delta(A_c;u_1,\ldots,u_k,e)\downarrow = q)$ & $(q = 1$ iff $\beta_k(A_b;e)$ is true)).

(1.5) $R^{2,k+1}_{e,b,c}$: $g_k(1) \le g_k(c) \Rightarrow \Box \forall e \exists q \le 1((\lim_{u_1}\ldots\lim_{u_k}\Delta(A_c;u_1,\ldots,u_k,e)\downarrow = q)$ & $(q = 1$ iff $\beta_{k+1}(\emptyset;e)$ is not true)).

(Conditions (1.1) and (1.2) and properties of the jump operator will allow us to be selective about those b and c for which we satisfy requirements. The second superscript in a requirement $R^{i,k}_{e,b,c}$ is the *dimension* of the requirement, and corresponds to the particular tree of strategies on which we begin to split the requirement up.) We refer to requirements listed in (1.3) as *incomparability* requirements, to those listed in (1.4) as *comparability* requirements, and to those listed in (1.5) as *highness* requirements.

The Basic Module: *Dimension 0 comparability requirements.* $R^{1,0}_{e,b,c}$ is satisfied by coding A_b into A_c. The construction will have the following property:

(1.6) If $e \in A_b^{s+1} \setminus A_b^s$, and $b \leq c$, then $e \in A_c^{s+1} \setminus A_c^s$.

Thus to decide whether $e \in A_b$, we ask if $e \in A_c$. If the answer is no, then $e \notin A_b$. If the answer is yes, then we find the least s such that $e \in A_c^s$, and note that $e \in A_b$ iff $e \in A_b^s$. We have thus proved:

Lemma 1.1: Let $b,c \in P_0$ be given such that $b \leq c$. Assume that the construction satisfies (1.6). Then $A_b \leq_T A_c$. ∎

The Basic Module: *Dimension 1 incomparability requirements.* We satisfy $\{R^{0,1}_{e,b,c}: e \in \mathbf{N}\}$ through a modification of the Friedberg-Muchnik strategy. We construct a partial recursive functional Δ such that $\Delta(A_c)$ is total. We will appoint a witness x and try to guarantee that $\Phi^1_e(A_b;x) \neq \Delta(A_c;x)\downarrow$. We begin by *activating* this requirement. To do so, we declare an axiom $\Delta(A_c^s;x) = 0$ with use x. If, at some later stage t, we find that $\Phi^1_{e,t}(A_b^t;x)\downarrow = 0$, with use q, then we will *restrain* $A_b \restriction (q+1)$ from changing after stage t, and place x into A_c. This will allow us to redefine $\Delta(A_c;x) = 1$ with use x. There are two possible types of outcomes. If, during the construction, we never see a computation $\Phi^1_{e,t}(A_b^t;x) = 0$, then $\Delta(A_c;x) = 0$ and either $\Phi^1_e(A_b;x)\uparrow$ or $\Phi^1_e(A_b;x)\downarrow \neq 0$. If we eventually place x into A_c, then $\Delta(A_c;x) = 1 \neq 0 = \Phi^1_e(A_b;x)$.

The Basic Module: *Dimension 1 comparability requirements.* If $f_1(b) \leq f_1(c)$ and $f_1(1) \not\leq f_1(c)$, then we will want

$$\lim_u \Delta(A_c;u,e) = \begin{cases} & \text{if } \beta_1(A_b;e) \text{ is true (i.e., } e \in A_b') \\ 0 & \text{if } \beta_1(A_b;e) \text{ is not true (i.e., } e \notin A_b'). \end{cases}$$

We wait for a stage s at which $\beta_1(A_b^s;e)$ is true with use < s, declaring axioms $\Delta(A_c^s;u,e) = 0$ for progressively larger u until such an s is found. If we ever find such a stage s, we restrain $A_b^s \restriction s$ (so $A_b \restriction s = A_b^s \restriction s$), and declare axioms $\Delta(A_c^t;v,e) = 1$ for all $t \geq s$ and all sufficiently large v, having use s.

The Basic Module: *Dimension 2 incomparability requirements.* $R^{0,2}_{e,b,c}$ will be treated as a connected infinite set of dimension 1 incomparability requirements, producing the desired result in the limit. Fix Δ. We will appoint a witness x, and try to guarantee that if $\{v: \Phi^2_e(A_b;v,x) = 0\}$ is infinite, then $\{u: \Delta(A_c;u,x) = 1\}$ is cofinite, and if $\{v: \Phi^2_e(A_b;v,x) = 0\}$ is finite, then $\{u: \Delta(A_c;u,x) = 0\}$ is cofinite. We thus begin to declare axioms $\Delta(A_c;u,x) = 0$ for progressively larger u, with large use p. (At each stage, only finitely many axioms of this kind are declared, and several may share the same use.) If, at some later stage t, we find some v_0 such that $\Phi^2_{e,t}(A^t_b;v_0,x) = 0$, then we restrain $A^t_b \upharpoonright t$ (so $A_b \upharpoonright t = A^t_b \upharpoonright t$), and place p into A_c at stage t, allowing us to declare new axioms $\Delta(A^t_c;u,x) = 1$ with use p for each u for which we have previously declared an axiom with output 0. We now repeat this procedure for larger values u for which no axiom has yet been declared, trying to find $v_1 > v_0$ for Φ^2_e. Thus either $\{v: \Phi^2_e(A_b;v,x)\downarrow = 0\}$ is finite and $\{u: \Delta(A_c;u,x)\downarrow = 0\}$ is cofinite, or $\{v: \Phi^2_e(A_b;v,x)\downarrow = 0\}$ is infinite and $\{u: \Delta(A_c;u,x)\downarrow = 1\} = \mathbf{N}$. In either case, $R^{0,2}_{e,b,c}$ is satisfied.

The Basic Module: *Dimension 2 highness requirements.* If $f_1(1) \leq_1 f_1(c)$ and K is the complete recursively enumerable set, then we will want to satisfy

$$\lim_u \Delta(A_c;u,e) = \begin{cases} 0 & \text{if } e \in K' \text{ (i.e., } \beta_2(\emptyset,e) \text{ is not true or, equivalently, } \beta_1(K,e) \text{ is true)} \\ 1 & \text{if } e \notin K' \text{ (i.e., } \beta_2(\emptyset,e) \text{ is true or, equivalently, } \beta_1(K,e) \text{ is not true).} \end{cases}$$

The strategy for satisfying this requirement is similar to that for the dimension 2 incomparability requirements. While $\Phi^1_{e,s}(K^s;e)\uparrow$, we declare axioms $\Delta(A^s_c;u,e) = 1$ for progressively larger u, with use 0. If we discover that $\Phi^1_{e,s}(K^s;e)\downarrow$ with use v, then we declare axioms $\Delta(A^s_c;u,e) = 0$ with large use p for progressively larger u for which axioms have not yet been defined. If $K^t \upharpoonright v \neq K^s \upharpoonright v$ at some later stage t, then we place p into A_c at stage t, and reset the axiom $\Delta(A^t_c;u,e) = 1$ with use p for values of u for which the axiom was previously set to 0, and begin again to declare axioms having output 1 and use $r \geq t$ for yet larger values of u. Thus either that $\Phi^1_e(K;e)\downarrow$ and $\{u: \Delta(A_c;u,x)\downarrow = 0\}$ is cofinite, or $\Phi^1_e(K;e)\uparrow$ and $\{u: \Delta(A_c;u,x)\downarrow = 1\} = \mathbf{N}$. In either case, $R^{2,2}_{e,1,c}$ is satisfied.

Comparability requirements of dimension 2, and requirements of dimension 3 or greater are handled in a way similar to that in which their counterparts of lower dimension are handled, except that more iterations of the limit operation are required. As no new strategies are involved, we will not discuss basic modules for these requirements.

Conflicts between requirements are resolved by placing requirements on iterated trees of strategies, and using the trees to determine when requirements should act. We fix the maximum dimension n of the requirements to be satisfied, and assign each requirement to all nodes of a given level of the tree T^n. Associated with a requirement is a sentence of the form $(\varphi^n \rightarrow \psi^n) \& (\neg \varphi^n \rightarrow \chi^n)$, where φ^n is a sentence which determines when to initiate action during the construction, and ψ^n and χ^n are properties which must result from this action. For $k < n$, nodes of a tree T^k will be derivatives of nodes of a tree T^{k+1} . Each derivative is to generate action based on the truth of a sentence φ^k obtained from the sentence φ^{k+1} assigned to the node from which it is derived, by appropriately bounding the outer block of quantifiers. And the action must result in satisfying some property derived from ψ^n and χ^n. Rather than bounding quantifiers for the sentence obtained from ψ^n and χ^n, we assign spaces to the nodes on which we ensure that a given functional has an iterated limit (sometimes requiring a specified value for this limit). When we reach T^0, we will be able to recursively specify when action should be taken and what that action should be. We then piece together the various sentences and actions on T^0 taken to show that the sentence assigned to T^n is satisfied.

The processes of assigning derivatives and of determining which derivatives should act are delicate, and differ with the type of requirement. This assignment must be done in such a way that the action taken by derivatives can be pieced together to show that the original sentence on T^n is true. The priority argument is hidden in this decomposition; thus if the decomposition is done correctly, there are no conflicts between requirements on T^0 as we have determined when nodes can act in a way which avoids conflicts (which, however, are seen on T^1). The key is to determine the derivative responsible for the definition of a given axiom. For incomparability requirements, the nodes specifying axioms for a given functional and argument in the limit are all derived from a single node of T^n, and control must be expanded to ensure that limits exist when the node is not on the true path of the construction. In the case of comparability and highness requirements, nodes on each path through T^n may specify axioms for a given functional and argument. In these cases, we must define control of axioms carefully, dividing control among many nodes. We also take advantage of the fact that if k is the dimension of the requirement, axioms will frequently be corrected (since the oracle set will change below the use of the axiom) whenever the true path approximation changes its mind about the ultimate node of T^k which is responsible for defining the axioms.

The notion of control will determine the axioms for which a node is responsible. Determining when a node should act will involve additional concepts, such as freeness, admissibility, implication chains, and links. We will define the assignment of requirements to trees in Section 2, and prove some lemmas about paths in Section 3. Links will be analyzed in Section 4. Implication chains and backtracking will be discussed in Section 5. Control will be discussed in Section 6. The construction and proof based on the machinery introduced in previous sections will be discussed in Section 7.

2. The Requirements and systems of trees. The framework for our priority argument uses systems of trees, and much of it can be presented independently of the set of requirements to be satisfied. Systems of trees are introduced in this section, and the mechanism for assigning requirements to the trees is described. (The reader is referred to [LmLr1], where systems of trees are used to prove some standard theorems of recursion theory. The framework there is a little different, as some of the subtleties needed here do not occur at the lower levels, but the many similarities in the approaches might be helpful.) Fix $n \in \mathbf{N}$ henceforth.

Definition 2.1 (Definition of trees): We set $T^{\pm1} = \{0,\infty\}$ and $T^0 = \{0,\infty\}^{<\omega}$. If $0 < k \leq n$ and $T^{k\pm1}$ has been defined, let

$$T^k = \{\sigma \in (T^{k\pm1})^{<\omega}: \forall i<lh(\sigma)\forall j<lh(\sigma)(i < j \to \sigma(i) \subset \sigma(j))\}.$$

$\mathbf{T}^k = \langle T^k, \subseteq\rangle$ is the *kth tree of strategies,* ordered by inclusion. We refer to the elements of T^k as nodes of T^k, and view each node of T^k as following its immediate predecessor by a designated node of $T^{k\pm1}$. If $\sigma \in T^k$, $\xi \in T^{k\pm1}$, and $\sigma = \sigma^- {}^\wedge\langle\xi\rangle$, then we say that σ^- has *outcome* ξ *along* σ, and define $out(\sigma) = \xi$. If $j \leq k$, then we define $out^j(\sigma)$ by reverse induction; $out^k(\sigma) = \sigma$, and $out^{j\pm1}(\sigma) = out(out^j(\sigma))$. Outcomes are of two types, activated or validated. If $k = 0$, $\tau \supseteq \sigma$, and $lh(\sigma) > 0$, then we say that σ^- *is activated* (*validated,* resp.) *along* τ if $out(\sigma) = 0$ ($out(\sigma) = \infty$, resp.). If $k > 0$, then σ^- *is activated* (*validated,* resp.) *along* τ if $out(\sigma)^-$ is activated (validated, resp.) along $out(\sigma)$. (Activation and validation represent different ways of satisfying a requirement depending on whether the sentence generating action is true or false. The steps taken when a requirement associated with the node σ^- is first activated may be later extended when that requirement is validated.) If $\sigma \subseteq \tau \in T^k$ and $lh(\sigma) > 0$, then we say that σ^- *has finite (infinite, resp.) outcome along* τ if either $k = 0$ and $out(\sigma) = 0$ ($out(\sigma) = \infty$, resp.), or $k > 0$ and $out(\sigma)^-$ has infinite (finite, resp.) outcome along $out(\sigma)$. (Note that σ^- is activated (validated, resp.) along σ if either k is even and σ^- has finite (infinite, resp.) outcome along σ, or k is odd and σ^- has infinite (finite, resp.) outcome along σ.) n

In order to provide the reader with some intuition about these trees, we relate them to the *tree of strategies* approach introduced by Harrington, and indicate the relationship between the way certain concrete requirements are treated by these approaches. First consider a typical Friedberg-Muchnik requirement $\Phi(A) \neq B$. The standard tree of strategies approach assigns such a requirement to a node σ of $T^1 = \{0,1\}^{<\omega}$, and proceeds by stages. When σ first appears on the true path, a *follower* x is assigned to the requirement. As long as σ is on the path through T^1 computed at stage s and $\Phi_s(A^s) \neq 0$, the path computation at s follows $\sigma^\wedge\langle 0\rangle$, and we set $B^s(x) = 0$. (We now say that σ is *activated*.) If, at some later stage t, we find that $\Phi_t(A^t;x) = 0$, the path computation at t follows $\sigma^\wedge\langle 1\rangle$, we set $B^t(x) = 1$, and never again consider this requirement. (We now say that σ is *validated*.)

Our approach replaces stages by the tree $T^0 = \{0,\infty\}^{<\omega}$. (We use ∞ in place of 1 because we want to talk about *finite* and *infinite* outcomes.) In place of stage t, we form a δ-block on T^0 which consists of *derivatives* of nodes of T^1 which lie on true path computation through T^1 generated by $\delta \in T^0$, and such that the requirement for these nodes of T^1 has not yet been validated. Each derivative along the path gets a chance to try to satisfy its requirement when it is reached, and the block ends either when we newly validate a node, or begin to deal with one new requirement. The outcome $\xi = \nu^\wedge\langle\beta\rangle$ of σ along $\rho = \sigma^\wedge\langle\xi\rangle \in T^1$ is used to code whether or not $\Phi(A;x) = 0$. ξ will tell us whether or not the requirement assigned to σ has been activated or validated, and in addition, that the decision to activate or validate was made based on the outcome of ν along ξ. Thus if σ is activated along ρ, then σ will will have infinitely many derivatives along the true path Λ^0 through T^0, all of which will be activated. The outcome $\xi = \nu^\wedge\langle\beta\rangle$ of σ will indicate that ν is the derivative of σ at which we made the decision to determine the outcome of σ along ρ, namely, the first derivative of σ along Λ^0 (we will call ν both the *initial* and *principal* derivative of σ along Λ^0), and the outcome $\beta = 0$ of ν along Λ^0 indicates that ν is activated along Λ^0. If σ is validated along ρ, then the outcome ξ of σ will determine the node ν of T^0 at which we made the decision to determine the outcome of σ along ρ (we will call ν the *principal derivative* of σ along δ), and the outcome $\beta = \infty$ of ν along Λ^0 indicates that ν is validated along ξ. If ν is not the first (i.e., initial) derivative μ of σ along ρ, then we create a link from μ to ν. These links partially correspond, in standard priority arguments, to initializing all extensions of ρ. At higher levels, they also serve the purpose of not allowing nodes restrained by the link to act and cause a change in the approximation to the true path. This allows us to show that when the outcome of a node is switched by the approximation, it must be switched because of action taken for the requirement for which the node is responsible. (We note that this approach differs from that in [LmLr1], where the outcome of ν was not coded along the outcome of σ, and ∞ was used in place of a node of T^0 to represent activation on T^1, i.e., denoting that σ has infinitely many derivatives along Λ^0. This approach works for $\mathbf{0}'''$-constructions, i.e., constructions which do not require a tree beyond T^3. Once T^4 is reached, initial derivatives of σ, i.e. derivatives of a node σ on T^4 which do not properly contain another derivative of σ, are no longer unique, and our approximations need to code these initial derivatives, rather than use a catch-all symbol ∞ to denote an infinite outcome.)

Next, consider a typical *thickness* requirement on T^2. We are given an infinite recursive set R, and activation corresponds to putting only finitely many elements of R into a set A, while validation corresponds to putting all elements of R into A. Suppose that this thickness requirement is assigned to a node σ of the true path Λ^2 through T^2. Then σ will have derivatives along the true path Λ^1 through T^1, each of which will have the role of placing finitely many elements of R into A if a certain Σ_1-sentence is true. First suppose that one of these sentences is false, say the one corresponding to the derivative ξ of σ. Then ξ will be the last (and *principal*) derivative of σ along Λ^1, and will have infinite outcome along Λ^1, designating that no derivative of ξ is validated along the true path Λ^0 on T^0, i.e., that no derivative of ξ finds a witness for its existential sentence at the stage specified by the framework. No elements ≥ the least element of R for which ξ has responsibility are placed into A in this case.

Now suppose that all of the sentences are true. Then σ will have infinite outcome along Λ^2, indicating that σ has infinitely many derivatives along Λ^1, each of which is validated. Each such node will place the elements of R for which it is responsible into A. As each element of R will be assigned to such a node, all elements of R will be placed into A. (The infinite outcome of σ along Λ^2 is the *initial* derivative of σ along Λ^1 (which also is the *principal* derivative of σ along Λ^1), followed by its first validated derivative ν along Λ^0 and the outcome ∞ for ν indicating that ν is validated.)

We will need a one-to-one *weight* function on elements of $T = \cup\{T^k: 0 \le k \le n\}$ which will ω-order T. (We take the disjoint union here, differentiating between the empty nodes of the various trees. A similar function was called *par* in [LmLr1].) The weight function will have various properties, which will be used to show that constructions in which action is determined by weight are able to protect certain computations.

Definition 2.2: It is routine to check that a one-to-one recursive *weight function* wt: $T \to \mathbf{N}$ can be defined to satisfy the following properties for all $\sigma,\tau \in T^k$:

(2.1) If $\sigma \subset \tau$ then $wt(\sigma) < wt(\tau)$.

(2.2) If $k > 0$, then $wt(out(\sigma)) < wt(\sigma)$.

(2.3) If $k > 0$ and $out(\sigma) \subset out(\tau)$, then $wt(\sigma) < wt(\tau)$. n

Definition 2.3: The action taken at each stage of the construction will be associated with a node of T^0. This node will be derived from a node of T^k where k is the *dimension* of the requirement, i.e., we begin to split up the requirement into subrequirements on $T^{k\pm1}$. Nodes of T^k will be of one of two types. Each node on T^k working on a given incomparability requirement can pick a different witness for its functional, and we call such requirements *locally distributed*. For any comparability or highness requirement and argument x, each path through T^k must contain a node working to define a value for a functional on argument x; such requirements are called *densely distributed*. ∎

We will take action to ensure the satisfaction of requirements. This action will consist in placing numbers into certain sets, and in trying to keep numbers out of other sets. Certain sets will be associated with a requirement R. OS(R), the *oracle set* of R, will contain a particular oracle from which the requirement wants to define axioms. We will want to prevent numbers from entering the oracles in RS(R), the *restraint set* of R. And TS(R) will be the *target set* of R, a set of oracles into which numbers should be placed in order to satisfy the requirement while preserving the ability to satisfy other requirements. If a requirement is assigned to a node σ, then the above definitions and notation are inherited by σ from R, and inherited by all derivatives of σ from σ.

Fix an m-jump poset, $\mathsf{P} = \langle P_0,\leq_0,P_1,\leq_1,f_1,\ldots,P_m,\leq_m,f_m\rangle$. There will be three types of requirements, which we now define.

Definition 2.4: Incomparability requirements have *dimension* $k \geq 1$ and *type* 0. They are locally distributed requirements, each associated with an element of

$$Z_{0,k} = \{\langle b,c\rangle \in P^2\colon g_{k-1}(c) \not\leq g_{k-1}(b) \mathbin{\&} (g_k(c) \leq g_k(b) \text{ or } k = m+1) \\ \mathbin{\&} g_{k-1}(b) \neq g_{k-1}(0) \mathbin{\&} g_{k-1}(c) \neq g_{k-1}(1)\}.$$

We establish a requirement $R = R^{0,k}_{e,b,c}$ for each $\langle b,c\rangle \in Z_{0,k}$ and $e \in \mathbf{N}$ as described in Section 1, whose goal is to make the condition

$$\exists x(\lim_{u_1}\ldots\lim_{u_{k-1}}\Delta(A_c;u_1,\ldots,u_{k-1},x) \neq \lim_{v_1}\ldots\lim_{v_{k-1}}\Phi^k_e(A_b;v_1,\ldots,v_{k-1},x))$$

true, if the latter limit exists. (The construction will automatically ensure that the first of the above limits exists for all x.) We set $RS(R) = \{A_a\colon g_{k-1}(a) \leq g_{k-1}(b)\}$, $OS(R) = \{A_c\}$, and $TS(R) = \{A_a\colon g_{k-1}(a) \not\leq g_{k-1}(b) \mathbin{\&} a \neq 1\}$. ∎

Definition 2.5: Comparability requirements have *type* 1 and *dimension* $k \geq 1$. They are densely distributed requirements, each associated with an element of

$$Z_{1,k} =\{\langle b,c\rangle \in P^2\colon g_k(b) \leq g_k(c) \mathbin{\&} g_{k-1}(b) \not\leq g_{k-1}(c) \mathbin{\&} g_k(1) \not\leq g_k(c)\}.$$

We establish a requirement $R = R^{1,k}_{e,b,c}$ for each $\langle b,c\rangle \in Z_{1,k}$ and $e \in \mathbf{N}$ as described in Section 1, whose goal is to ensure that

□

$$\lim_{u_1}\ldots\lim_{u_k}\Delta(A_c;u_1,\ldots,u_k,e) = \begin{cases} A_b^{(k)}(e) & \text{if k is odd} \\ \overline{A_b^{(k)}}(e) & \text{if k is even.} \end{cases}$$

We set $RS(R) = \{A_a: g_k(a) \leq g_k(b)\}$, $TS(R) = \{A_a: g_k(a) \nleq g_k(b)\ \&\ a \neq 1\}$, and $OS(R) = \{A_c\}$. n

Definition 2.6: Highness requirements have *type 2* and dimension $k \geq 2$. They are densely distributed requirements, each associated with an element of

$$Z_{2,k} = \{\langle 1,c\rangle \in P^2: g_{k-1}(1) \leq g_{k-1}(c)\ \&\ g_{k-2}(1) \nleq g_{k-2}(c)\}.$$

We establish a requirement $R = R^{2,k}_{e,1,c}$ for each $\langle 1,c\rangle \in Z_{2,k}$ and $e \in \mathbf{N}$ as described in Section 1, whose goal is to ensure that

$$\lim_{u_1}\ldots\lim_{u_{k-1}}\Delta(A_c;u_1,\ldots,u_{k-1},e) = \begin{cases} \emptyset^{(k)}(e) & \text{if k is odd} \\ \overline{\emptyset^{(k)}}(e) & \text{if k is even.} \end{cases}$$

We set $RS(R) = \emptyset$, $TS(R) = \{A_a: g_{k-1}(1) \leq g_{k-1}(a)\ \&\ a \neq 1\}$, and $OS(R) = \{A_c\}$. n

Lemma 2.1: Let $\mathsf{P} = \langle P,\leq,P_1,\leq_1,f_1,\ldots,P_m,\leq_m,f_m\rangle$ be an m-jump poset with least element 0 and greatest element 1. Suppose that we have a map h from P to the r.e. sets given by $b \rightarrow A_b$ which maps 0 to $\emptyset$, 1 to K, and satisfies the following conditions for all $b,c \in P$:

(i) $\langle b,c\rangle \in Z_{0,k} \Rightarrow A_c^{(k\pm1)} \nleq_T A_b^{(k\pm1)}$.

(ii) $b \leq c \Rightarrow A_b \leq_T A_c$.

(iii) $\langle b,c\rangle \in Z_{1,k} \Rightarrow A_b^{(k)} \leq_T A_c^{(k)}$.

(iv) $\langle 1,c\rangle \in Z_{2,k} \Rightarrow È^{(k)} \leq_T A_c^{(k\pm1)}$.

Then the m-jump poset generated by the image of h in the r.e. degrees is isomorphic to P.

Proof: Fix $b,c \in P$. We proceed by cases.

Case 1: $g_k(b) \leq_k g_k(c)$. As the jump operator is order-preserving, we can assume that k is the least r such that $g_r(b) \leq_r g_r(c)$.

Subcase 1.1: $k = 0$. Then $A_b^{(k)} \leq_T A_c^{(k)}$ by (ii).

Subcase 1.2: k□> 0.

Subcase 1.2.1: $g_k(1) \not\leq_k g_k(c)$. Then $A_b^{(k)} \leq_T A_c^{(k)}$ by (iii).

Subcase 1.2.2: $g_k(1) \leq_k g_k(c)$. As the jump operator is order-preserving, $A_b^{(k)} \leq_T A_1^{(k)}$. By (iv), $A_1^{(k)} \leq_T A_c^{(k)}$. But the degrees form a poset, so $A_b^{(k)} \leq_T A_c^{(k)}$.

Case 2: $g_k(c) \not\leq_k g_k(b)$. As the jump operator is order-preserving, it suffices to show that $A_c^{(k)} \not\leq_T A_b^{(k)}$ under the assumption that k is the largest $r \leq m$ such that $g_r(c) \not\leq_r g_r(b)$.

Subcase 2.1: $g_k(b) \neq g_k(0)$ and $g_k(c) \neq g_k(1)$. Then $A_c^{(k)} \not\leq_T A_b^{(k)}$ by (i).

Subcase 2.2: $g_k(b) = g_k(0)$ and $g_k(c) = g_k(1)$. As the jump operator has the property that $\mathbf{0}^{(k)} < \mathbf{0}^{(k+1)}$ and h(1) = K has degree $\mathbf{0}'$, $A_c^{(k)} \not\leq_T A_b^{(k)}$.

Subcase 2.3: $g_k(b) = g_k(0)$, $g_k(c) \neq g_k(1)$, and $c \neq d$. Then by (1.1), $g_k(c) \not\leq_k g_k(d)$, so as $g_k(0) <_k g_k(d) <_k g_k(1)$ by (1.1), we can apply (i) to conclude that $A_c^{(k)} \not\leq_T A_d^{(k)}$. As the jump operator is order-preserving, $A_0^{(k)} \leq_T A_d^{(k)}$, so as the degrees form a poset, $A_c^{(k)} \not\leq_T$□$A_0^{(k)}$. By Case 1, $A_b^{(k)} \equiv_T A_0^{(k)}$, so $A_c^{(k)} \not\leq_T A_b^{(k)}$.

Subcase 2.4: $g_k(b) = g_k(0)$, $g_k(c) \neq g_k(1)$, and c = d. We proceed as in Case 2.3, replacing d with $\tilde{d}$ and (1.1) with (1.2).

Subcase 2.5: $g_k(c) = g_k(1)$, $g_k(b) \neq g_k(0)$ and $b \neq d$. By (1.1), $g_k(d) \not\leq_k g_k(b)$, so as $g_k(0) <_k g_k(d) <_k g_k(1)$ by (1.1), we can apply (i) to conclude that $A_d^{(k)} \not\leq_T A_b^{(k)}$. As the jump operator is order-preserving, $A_d^{(k)} \leq_T A_1^{(k)}$, so as the degrees form a poset, $A_1^{(k)} \not\leq_T A_b^{(k)}$. By Case 1, $A_c^{(k)} \equiv_T A_1^{(k)}$. Hence $A_c^{(k)} \not\leq_T A_b^{(k)}$.

Subcase 2.6: $g_k(c) = g_k(1)$, $g_k(b) \neq g_k(0)$ and b = d. We proceed as in Case 2.5, replacing d with $\tilde{d}$ and (1.1) with (1.2). n

The next lemma relates target sets, oracle sets and restraint sets for various requirements. It is used to show that once we satisfy requirements, we can preserve this satisfaction if the requirement lies on the true path for the construction. It is also used to show that action taken for requirements which do not lie along the true path for the construction is corrected, when necessary, in the process of returning to the true path. This lemma provides a crucial connection between the general framework and the particular set of requirements which we must satisfy.

Lemma 2.2 (Interaction Lemma): Fix requirements $R = R^{j,k}_{e,b,c}$ and $\tilde{R} = R^{\tilde{j},\tilde{k}}_{\tilde{e},\tilde{b},\tilde{c}}$ such that $\tilde{k} \geq k$. Then the following conditions hold:

(i) $TS(R) \cap RS(R) = \emptyset$.

(ii) If $tp(R) \in \{0,2\}$, then $OS(R) \subseteq TS(R)$.

(iii) Suppose that $tp(R) = 1$, that $A_b \in TS(\tilde{R})$, and that if $tp(\tilde{R}) \in \{0,2\}$ then $k < \tilde{k}$. Then $A_c \in TS(\tilde{R})$.

(iv) If $tp(R) = 2$, then $A_c \in TS(\tilde{R})$.

Proof: We note that that if $tp(R) = 0$, then $g_{k-1}(c) \not\leq_{k-1} g_{k-1}(b)$. (i) and (ii) are now routine to verify.

(iii): Suppose that $tp(R) = 1$ and $A_b \in TS(\tilde{R})$. Then $g_k(b) \leq_k g_k(c)$, so as $k \leq \tilde{k}$, $g_{\tilde{k}}(b) \leq_{\tilde{k}} g_{\tilde{k}}(c)$. (iii) now follows if $tp(\tilde{R}) = 1$.

Suppose that $tp(\tilde{R}) = 2$. Since $A_b \in TS(\tilde{R})$, $g_{\tilde{k}-1}(1) \leq_{\tilde{k}-1} g_{\tilde{k}-1}(b)$. As $g_k(b) \leq_k g_k(c)$ and $k < \tilde{k}$, $g_{\tilde{k}-1}(1) \leq_{\tilde{k}-1} g_{\tilde{k}-1}(c)$. Hence $A_c \in TS(\tilde{R})$.

Finally, suppose that $tp(\tilde{R}) = 0$. Since $A_b \in TS(\tilde{R})$, $g_{\tilde{k}-1}(b) \not\leq_{\tilde{k}-1} g_{\tilde{k}-1}(\tilde{b})$. As $g_k(b) \leq_k g_k(c)$ and $k < \tilde{k}$, $g_{\tilde{k}-1}(b) \leq_{\tilde{k}-1} g_{\tilde{k}-1}(c)$. It now follows that $g_{\tilde{k}-1}(c) \not\leq_{\tilde{k}-1} g_{\tilde{k}-1}(\tilde{b})$, else by transitivity, $g_{\tilde{k}-1}(b) \not\leq_{\tilde{k}-1} g_{\tilde{k}-1}(\tilde{b})$, yielding a contradiction. Hence $A_c \in TS(\tilde{R})$.

(iv): Suppose that $tp(R) = 2$. Then $g_{k-1}(1) \leq_{k-1} g_{k-1}(c)$. As $k \leq \tilde{k}$, $g_{\tilde{k}-1}(1) \leq_{\tilde{k}-1} g_{\tilde{k}-1}(c)$. It is now easily checked that $A_c \in TS(\tilde{R})$. ∎

Definition 2.7: Fix a recursive ordering $\{R_i : i \in \mathbf{N}\}$ of all requirements. We say that R_i *has higher priority than* R_j if $i < j$. If $R_i = R^{j,k}_{e,b,c}$, is assigned to node $\sigma \in T^k$, then we say that $tp(\sigma) = j$ and $dim(\sigma) = k$. ∎

Requirements of dimension k will be assigned to nodes of trees T^r for $r \geq k$, and subrequirements of these requirements will be assigned to nodes of T^j for $j < k$. Whenever R_i is assigned to two nodes σ and τ and $tp(\sigma) \in \{1,2\}$, then we say that $\sigma \equiv \tau$. (We will extend the definition of $\equiv$ to additional pairs of nodes later, and then take the reflexive, symmetric, transitive closure of the relation defined to make $\equiv$ an equivalence relation. Equivalent nodes work on the same functional, and sometimes on the same arguments for that functional. To satisfy requirements for all e, we will define a given functional as a disjoint union of many partial recursive functionals. The union over nodes in many equivalence classes will define the functional (for a specified oracle) on a recursive domain. We will take steps to ensure that functionals are total on specified oracles, whenever this is required.)

The assignment of requirements to nodes of trees will proceed by induction on $k = n-j$ for $j \leq n$. (n will be the largest dimension of a requirement in our list.) The inductive step of the definition will proceed in four steps. In Step 1, we will define the path generating function λ on nodes of trees which have already had requirements assigned to all of their predecessors. If $\sigma \in T^k$, then $\lambda(\sigma)$ will be a node on T^{k+1}. Given a path $\Lambda \in [T^k]$, $\{\lambda(\sigma):\square\sigma \subset \Lambda\}$ gives an approximation to a path $\lambda(\Lambda) \in [T^{k+1}]$. When $\lambda(\sigma)|\lambda(\sigma^-)$, a link will be formed on T^k. These links, defined in Step 2, will prevent action by nodes of T^k which do not seem to come from the true path approximation for trees of higher dimension. We will have to decide which nodes of T^{k+1} are eligible to assign subrequirements to a given node of T^k. Conditions ensuring consistency between the different trees enter into this decision, and these conditions are delineated in Step 3. The requirement assignment process is described in Step 4.

Definition 2.8: We proceed by induction on $k = n-j$, assigning requirements to nodes of T^k and dividing T^k into blocks of requirements. If $k = n$, then the requirement R_i is assigned to every node σ of T^n such that $lh(\sigma) = i$. Each node of T^n is a *block*. Thus for $\sigma \in T^n$, we say that σ *lies in the σ-block,* that we *begin the σ-block at σ,* and that a *path through the σ-block is completed at σ.*

Suppose that $k < n$. There are four steps.

Step 1: **Definition of the path generating function λ.** Given a node $\eta \in T^k$ such that requirements have been assigned to all predecessors of η, the function λ will define a node $\lambda(\eta) \in T^{k+1}$. The process is meant to capture the following situation. For each $\xi \subset \eta$, ξ will be derived from a node $\sigma \in T^{k+1}$. A sentence M_σ will be associated with σ, and a fragment M_ξ of that sentence will be associated with ξ. Suppose that the first unbounded quantifier of M_σ is a universal quantifier. If σ has dimension $\geq k+1$, we bound the leading block of universal quantifiers by numbers which are strictly increasing with $lh(\xi)$. As long as each ξ succeeds in satisfying its sentence M_ξ, the approximation given by λ predicts that $\sigma\hat{}\langle\nu\hat{}\langle\beta\rangle\rangle \subseteq \lambda(\eta)$, where ν will be the initial derivative of σ along η (defined formally below) and β is the outcome of ν along η. If we find a first ξ for which M_ξ is false, then $\sigma\hat{}\langle\xi\hat{}\langle\beta\rangle\rangle \subseteq \lambda(\eta)$, where β is the outcome of ξ along η. If the first unbounded quantifier of M_σ is an existential quantifier, then we proceed as above after replacing M_ξ with $\neg M_\xi$. (If $\dim(\sigma) \leq k$, then outcomes on T^k give rise to unique outcomes on T^{k+1}.)

If $\eta = \langle\,\rangle$ then $\lambda(\eta) = \langle\,\rangle$. Suppose that $\eta \neq \langle\,\rangle$. By (2.4), it will follow by induction that $up(\eta^-) \subseteq \lambda(\eta^-)$, where $up(\eta^-)$ is the node of T^{k+1} from which η^- is derived.

(up(η^-) has been defined inductively in Step 4 for η^-.)

(2.4) If either up(η^-) = $\lambda(\eta^-)$ or η^- has infinite outcome along η, then we set $\lambda(\eta)$ = up(η^-)^$\langle\eta\rangle$. We set $\lambda(\eta) = \lambda(\eta^-)$ otherwise.

(It follows from Definition 2.1 that $\lambda(\eta) \in T^{k+1}$.) It follows from (2.4) that:

(2.5) If $\sigma \subseteq \lambda(\eta)$ then out(σ) $\subseteq \eta$ and λ(out(σ)) = σ; and

(2.6) If $\lambda(\eta^-) \supseteq \sigma$ and $\lambda(\eta) \not\supseteq \sigma$, then for all $\delta \supseteq \eta$, $\lambda(\delta) \not\supseteq \sigma$.

We define $\lambda^r(\eta)$ for $r \in [k,n]$ by $\lambda^k(\eta) = \eta$ and $\lambda^r(\eta) = \lambda(\lambda^{r\pm1}(\eta))$ for $r > k$. Given $\xi \subseteq \eta$, we say that ξ is the *principal derivative of* up(ξ) (defined in Step 4) *along* η if either ξ has infinite outcome along η, or ξ is the shortest derivative of up(ξ) along η and for all $\gamma \subset \eta$, if up(γ) = up(ξ), then γ has finite outcome along η. (We do not require that up(ξ) $\subseteq \lambda(\eta)$.) And if $r \geq k$ and $\zeta \in T^r$, we call ξ the *principal derivative of* ζ *along* η if either $r = k$ and $\xi = \zeta \subset \eta$, or $r > k$, ξ is the principal derivative of up(ξ) along η and up(ξ) is the principal derivative of ζ along $\lambda(\eta)$.

Step 2: Links. We will place restrictions on the stages of the construction at which nodes are eligible to be *switched* by the approximation to the true path. One restriction requires a node to be *free* when it is switched by the true path approximation, i.e., that it not be contained in any *link*. Links are formed when a switch occurs, and can be broken when the outcome of a switched node is switched back. (Links correspond to initialization, after injury, in the standard approach to infinite injury priority arguments. Suppose that a node $\sigma \in T^2$ has initial derivative ν (defined inductively in Step 4) along a path Λ^1 through T^1, and principal derivative $\pi \supset \nu$ along $\eta \subset \Lambda^1$. Then we form a primary η-link $[\nu,\pi]$ from ν to π, thereby restraining any node $\xi \in [\nu,\pi)$ from acting and destroying computations declared by π. (Note that if $[\nu,\pi]$ is an η-link, then π is not restrained by $[\nu,\pi]$. However, as we can have $[\nu,\pi) = [\nu,\delta)$ as intervals with $\pi \neq \delta$, we use closed interval notation $[\nu,\pi]$ for η-links to make sure that there is a one-to-one correspondence between intervals which determine links, and the links themselves.) Any such $\xi \in [\nu,\pi)$ will either be a derivative of a node which is no longer on the approximation to the true path, or a derivative of a node $\rho \subseteq \sigma$. In the former case, clause (2.7) of the definition of η-consistency which is presented in Step 3 will also prevent derivatives of ξ from acting. The links are aimed at preventing derivatives of $\rho \subseteq \sigma$ from acting. Derivatives of such a node ρ which extend π will be able to act, and we will show that there is no harm in preventing derivatives of ρ restrained by the link from acting. We will allow derivatives of π to act, and so do not restrain π in this link.)

A node $\eta \in T^k$ such that $lh(\eta) > 0$ is said to be *switching* if there is an $r > k$ such that $\lambda^r(\eta^-)|\lambda^r(\eta)$. For the least such r, we say that η is *r-switching*. If $j \in [r,n]$ and η is r-switching, we say that *η switches* $up^j(\eta^-)$.

Fix $\eta \in T^k$. Each η-link will be derived from a *primary* $\lambda^j(\eta)$-link for some $j \geq k$, and will have either *finite* or *infinite outcome*. We define the *η-links* of T^k by induction on n-k. If k = n, then there are no η-links. Suppose that k < n.

We first determine the *primary η-links*. Suppose that $\xi \subseteq \eta$ and ξ^- is the principal derivative of $\gamma = up(\xi^-)$ along η, but is not the initial derivative of γ along η. Let μ be the initial derivative of γ along η. Then $[\mu,\xi^-]$ is a *primary η-link* and has *infinite outcome*.

η-links can also be created by pulling down λ(η)-links. Suppose that [ρ,τ] is a λ(η)-link on T^{k+1}. Assume that the initial derivative μ of ρ along η and the principal derivative π of τ along η both exist. Then [μ,π] is an *η-link derived from* [ρ,τ]. [μ,π] *has finite outcome* if [ρ,τ] has infinite outcome, and has infinite outcome otherwise.

If [ρ,τ] is derived from some link [ζ,κ], then every link derived from [ρ,τ] is *derived from* [ζ,κ]. We say that *ξ is η-restrained* if there is an η-link [μ,π] such that $\mu \subseteq \xi \subset \pi$. In this case, we say that *ξ is η-restrained by* [μ,π]. *ξ is η-free* if ξ is not η-restrained. *ξ is free* if ξ is ξ-free.

Step 3: η-consistency. We decide, in this step, whether a node $\sigma \in T^{k+1}$ is allowed to assign subrequirements at η. This will depend on four conditions. The first condition, (2.7), requires η to predict that σ is on the true path of T^r for all $r \in [k+1,n]$. The second condition, (2.8), requires that if $\sigma \in T^{k+1}$, once a witness $\xi \subset \eta$ for an existential sentence associated with σ is found, no derivatives of σ can extend ξ. In this case, η has all the information needed to correctly predict the outcome of σ. However, we do not search for such witnesses on T^k if $k \geq \dim(\sigma)$, as we have not yet begun to decompose the sentence associated with σ in this case. Rather, we will require an outcome of σ to code the outcome of a unique derivative of σ along a path of T^k, and so impose condition (2.9) requiring that there be a unique such derivative. Condition (2.10) requires that σ be λ(η)-free. Lemma 4.4 will show that this condition implies condition (2.7), but for now, it is convenient to require both conditions.

For $\eta \in T^k$, we say that $\sigma \in T^{k+1}$ *is η-consistent* if the following conditions hold:

(2.7) $up^r(\sigma) \subseteq \lambda^r(\eta)$ for all $r \in [k+1,n]$.

(2.8) If $\sigma \subset \lambda(\eta)$ and $\dim(\sigma) > k$, then for all $\nu \subset \eta$, if $up(\nu) = \sigma$ then ν has finite outcome along η.

(2.9) For all $\nu \subset \eta$, if $\dim(\sigma) \le k$ then $up(\nu) \neq \sigma$.

(2.10) σ is $\lambda(\eta)$-free.

(We note that our definition of η-consistency differs from that in [LmLr1] in that we impose (2.10) as an additional restriction. This restriction is needed to show that whenever a path approximation is switched precisely at $\sigma \in T^r$, then for all $k \ge r$, the path approximation on T^k is switched precisely at the node from which σ is derived. Many of the lemmas we prove rely on this fact. If $k \le 3$, however, this property of switching is automatic, so (2.10) does not need to be imposed.)

Step 4: Assignment of Derivatives. Let $\eta \in T^k$ be given such that requirements have been assigned to all predecessors of η, but not to η. We want to assign a requirement to η. The requirement chosen will be one which has been assigned to some η-consistent node of T^{k+1}.

Requirements are assigned in *blocks*. (Blocks on T^0 are the counterpart of *stages* in the usual approach to priority constructions. Thus if a block is begun at $\delta \in T^0$ and a path through the block is completed at $\xi \in T^0$, then $[\delta,\xi]$ corresponds to a set of substages of a given stage.) We *begin* a block at $\delta \in T^k$ if either $\delta = \langle\,\rangle$ or a path through a block was completed at δ^-. If we begin a block at δ, then this block is called the δ-*block*. A *path through the* δ*-block is completed at* $\xi \supseteq \delta$ if $up(\xi)$ completes a path through some block of T^{k+1} and ξ is an initial derivative of $up(\xi)$. We say that γ *lies in* the δ-block if $\delta \subseteq \gamma$ and no path through the δ-block has been completed at any $\beta \subset \gamma$.

Fix δ such that η is in the δ-block. If either $\eta = \langle\,\rangle$, $\eta = \delta$, or η is switching, set $\rho = \langle\,\rangle$. Otherwise, fix $\rho \subseteq \lambda(\eta)$ such that $\rho^- = up(\eta^-)$. (By induction using (2.7), $up(\eta^-) \subseteq \lambda(\eta^-)$ and η provides an outcome for a derivative of $up(\eta^-)$; hence by (2.7), $up(\eta^-) \subset \lambda(\eta)$ so such a ρ must exist.) Fix the shortest σ such that $\rho \subseteq \sigma \subseteq \lambda(\eta)$ and σ is η-consistent. (We note that for any $j \ge k$, any $\lambda^j(\eta)$-link $[\mu^j,\pi^j]$ satisfies $\pi^j \subset \lambda^j(\eta)$, so $\lambda(\eta)$ is $\lambda(\eta)$-free. Furthermore, (2.7) for k will follow from (2.7) for k+1. It thus follows that $\lambda(\eta)$ is η-consistent, so σ must exist.) Let R_i be the requirement assigned to σ. We assign R_i to η, designate η as a *derivative* of σ, and say that $up(\eta) = \sigma$. We assign a type, dimension, oracle set, target set, and restraint set to η in the same way as these were assigned to σ.

The derivative operation can be iterated; thus for every ζ such that σ is a derivative of ζ, we call η a *derivative* of ζ. η is also a derivative of η. If $r > k$, $\zeta \in T^r$, and η is a derivative of ζ then we write $up^r(\eta) = \zeta$. If there is no $\xi \subset \eta$ such that $up(\xi) = \sigma$, then for all $\nu \supseteq \eta$, we call η the *initial derivative of* σ *along* ν, and if σ is the initial derivative of ζ

along σ, then η is also the *initial derivative of* ζ *along any* $\nu \supseteq \eta$. We specify that $\eta \equiv \sigma$. ζ is an *antiderivative* of ξ if ξ is a derivative of ζ.

If Λ^k is a path through T^k, then we let $\lambda(\Lambda^k) = \lim_s\{\lambda(\Lambda^k \restriction s)\}$, and define $\Lambda^{k+1} = \lambda(\Lambda^k)$. (We will show in Lemma 3.2 that $lh(\Lambda^{k+1})$ exists and is infinite.) For $\Lambda^k \in [T^k]$, the Λ^k-*links* are the η-links for those η such that $\lambda^j(\eta) \subset \Lambda^j$ for all $j \in [k,n]$. We now define ξ to be Λ^k-*restrained* or Λ^k-*free* as in Step 2, with Λ^k in place of η.

The description of the assignment of requirements to nodes is now complete. We take the reflexive, symmetric, and transitive closure of $\equiv$ as defined in Step 4 and before Step 1 to generate an equivalence relation. ∎

We note an important relationship between the functions wt and λ. Suppose that $k < n$, $\sigma \subset \tau \in T^k$, and $\lambda(\sigma) \neq \lambda(\tau)$. By (2.5), $out(\lambda(\sigma)) \subseteq \sigma$ and $out(\lambda(\tau)) \subseteq \tau$, so by (2.4) and (2.5) and as $\sigma \subset \tau$, $out(\lambda(\sigma)) \subseteq \sigma \subset out(\lambda(\tau)) \subseteq \tau$. It now follows from (2.3) that:

(2.11) For all $k < n$ and $\sigma \subset \tau \in T^k$, if $\lambda(\sigma) \neq \lambda(\tau)$, then $wt(\lambda(\sigma)) < wt(\lambda(\tau))$.

We now indicate how to specify the sentence which generates the action for a requirement assigned to a given node. Our requirements will be of the form $(\varphi \rightarrow \psi)$ & $(\neg\varphi \rightarrow \chi)$. We will show that for requirements, all of whose antiderivatives lie on the true paths determined by the construction, ψ is true if φ is true, and χ is true if φ is false. To achieve this goal, we will have to correct action taken when it seemed that φ was false if we later discover that φ is true. The interplay between this correction feature and the determination of the node which controls the definition of a given axiom is the essence of priority arguments. Furthermore, as requirements will be introduced on T^k for $k > 0$ and the construction takes place on T^0, we must work with fragments of φ on T^0 rather than φ itself. When introduced, φ is assigned to a node σ of T^k, and fragments of φ, obtained by bounding some of the quantifiers of φ, are assigned to derivatives of σ. We now define the sentences and describe the decomposition process. In order to avoid notational confusion later, we use M in place of φ.

Definition 2.9 (Sentences, Base Step): For each $\sigma \in T^k$ such that $\dim(\sigma) = k$, there is a requirement $R = R^{j,k}_{e,b,c}$ which is assigned to $up^n(\sigma)$. We will assign a sentence M_σ to σ such that M_σ is Π_{k+1} if k is even, and is Σ_{k+1} if k is odd. Thus we require M_σ to have, as its final quantifier, a universal quantifier.

Suppose that $R^{j,k}_{e,b,c}$, is assigned to node σ of T^k. For $k \geq 1$, let $\gamma^k(e,x,b)$ be the formula with free variable x

$$\exists x_0 \forall y_0 \geq x_0 \forall x_1 \exists y_1 \geq x_1 \cdots \forall x_{k-2} \exists y_{k-2} \geq x_{k-2} \exists s \forall t \geq s(\Phi_e^t(A_b^t;\overline{y},x) = 0)$$

if k is odd (there is no block $\exists x_0 \ldots \exists y_{k-2} \geq x_{k-2}$ when k = 1), and

$$\forall x_0 \exists y_0 \geq x_0 \exists x_1 \forall y_1 \geq x_1 \cdots \forall x_{k-2} \exists y_{k-2} \geq x_{k-2} \exists s \forall t \geq s(\Phi_e^t(A_b^t;\overline{y},x) = 0)$$

if k is even. If j = 0, we let M_σ be the sentence $\gamma^k(e,wt(\sigma),b)$. If A is recursively enumerable, then by Definition 1.1, we can fix a sentence $\beta_r(A;e)$, such that, if r is odd, then $\beta_r(A;e)$ is a Σ_{r+1} sentence whose truth agrees with the truth of "$e \in A^{(r)}$", and if r is even, then $\beta_r(A;e)$ is a Π_{r+1} sentence whose truth agrees with the truth of "$e \notin A^{(r)}$". Furthermore, we can write $\beta_r(A;e)$ as $\overline{Qx}\exists s \forall t \geq s \tilde{\beta}_r(A^t;e)$, where $\overline{Qx}$ is a quantifier block and $\tilde{\beta}_r(A^t,e)$ is quantifier free. Suppose that the requirement $R^{j,k}_{e,b,c}$ is assigned to σ for $j \in \{1,2\}$. We let M_σ be the sentence $\beta_k(A_b,e)$ if j = 1, and $\beta_k(\emptyset,e)$ if j = 2. Our construction will have the property that if $up^r(\sigma) = \zeta^r$ lies on the true path of T^r for all $r \in [k.n]$, then M_{ζ^r} is true iff $\Delta_{\zeta^r}(A_c;\overline{y})$ takes the value which ensures the satisfaction of the requirement assigned to ζ^r. ■

Definition 2.10 (Sentences, Inductive Step): Suppose that $\dim(\sigma) > k$. Define $M_\sigma = (M_{up(\sigma)})^{[wt(\sigma)]}$. ■

3. Paths and Switching. In this section, we prove some technical lemmas about properties of the path generation process. The first lemma shows that the paths through the trees are infinite, and that initial and principal derivatives exist. This lemma is used many times to analyze the process of decomposing requirements.

Lemma 3.1 (Limit Path Lemma): Fix $k \in [0,n)$ and a path $\Lambda^k \in [T^k]$, and let $\Lambda^{k+1} = \lambda(\Lambda^k) = \lim\{\lambda(\eta): \eta \subset \Lambda^k\}$. Then:

(i) If $\sigma \subset \lambda(\Lambda^k)$, then σ has an initial derivative ν along Λ^k and $\lambda(\nu)^- = \sigma$.

(ii) If $\sigma \subset \lambda(\Lambda^k)$, then there is a $\pi \subseteq \Lambda^k$ such that π^- is the principal derivative of σ along Λ^k, $\lambda(\pi)^- = \sigma$, and for all $\eta \subseteq \Lambda^k$, $\lambda(\pi) \subseteq \lambda(\eta)$ iff $\pi \subseteq \eta$.

(iii) For any δ-block such that $\delta \subset \Lambda^k$, there is a $\xi \subset \Lambda^k$ such that ξ completes a path through the δ-block.

(iv) $lh(\Lambda^{k+1}) = \infty$.

Proof: We proceed by induction on $j = n-k$.

(i): By (2.4) and as $\sigma \subset \lambda(\Lambda^k)$, σ must have a derivative along Λ^k. Hence if ν is the shortest derivative of σ along Λ^k, then ν is the initial derivative of σ along Λ^k. By (2.7), $\lambda(\nu) \supseteq \sigma$. By (2.4) and (2.7), no $\tau \supset \sigma$ can have a derivative $\mu \subset \nu$. Hence by (2.4), $\lambda(\nu) = \sigma$.

(ii): If $\dim(\sigma) \leq k$, then by (2.9), the initial derivative ν of σ along Λ^k is the principal derivative of σ along Λ^k. (ii) follows in this case from (i), (2.5) and (2.6).

Suppose that $\dim(\sigma) > k$. By (i), let ν be the initial derivative of σ along Λ^k. If there is no $\pi \subset \Lambda^k$ such that $up(\pi^-) = \sigma$ and π^- has infinite outcome along π, then it follows as in the case for $\dim(\sigma) \leq k$ that ν is the principal derivative of σ along Λ^k. Otherwise, fix the shortest such π. We note that π^- is the principal derivative of σ along Λ^k. By (2.4), induction, (2.7) and (2.6), $\lambda(\pi)\square= \sigma\hat{}\langle\pi\rangle \subseteq \Lambda^{k+1}$, and if $\eta \subset \Lambda^k$ then $\lambda(\eta)\square\supseteq \sigma\hat{}\langle\pi\rangle$ iff $\eta \supseteq \pi$.

(iii),(iv): It follows easily from (2.7) that $\Lambda^{k+1} = \lambda(\Lambda^k) = \lim\{\lambda(\eta): \eta \subset \Lambda^k\}$ exists. First suppose that $lh(\Lambda^{k+1}) = \infty$. By (iii) inductively, there are infinitely many blocks along Λ^{k+1}, so infinitely many $\tau \subset \Lambda^{k+1}$ such that τ completes a path through a block. By (i), each such τ has an initial derivative along Λ^k. Hence by Definition 2.8, Step 4, there are infinitely many $\xi \subset \Lambda^k$ which complete paths through blocks, and (iii) holds in this case.

Now suppose that $lh(\Lambda^{k+1}) < \infty$ in order to obtain a contradiction. Then by (2.7), there is an $\eta \subset \Lambda^k$ such that for all ξ satisfying $\eta \subseteq \xi \subset \Lambda^k$, $\lambda(\xi) = \Lambda^{k+1}$. If $\eta \subseteq \xi \subset \Lambda^k$ and ξ completes a path through a block, then ξ must be an initial derivative of some node $\subseteq \Lambda^{k+1}$. As this is possible only finitely often and $lh(\Lambda^k) = \infty$, we can assume without loss of generality that there is no ξ such that $\eta \subseteq \xi \subset \Lambda^k$ and ξ completes a path through any block. By (2.6) and the choice of ρ in Step 4 of Definition 2.8, if $\eta \subseteq \xi \subset \delta \subset \Lambda^k$ then ξ is nonswitching, so $up(\xi) \subset up(\delta) \subseteq \Lambda^{k+1}$. But this is impossible if $lh(\Lambda^{k+1}) < \infty$ and $lh(\Lambda^k) = \infty$. n

From now on, whenever we write $\Lambda^k \in [T^k]$, we assume that there is a $\Lambda^0 \in [T^0]$ such that $\Lambda^k = \lambda^k(\Lambda^0)$. Furthermore, if we write $\eta \in T^k$, we assume that $\eta \subset \Lambda^k$ for some $\Lambda^k \in [T^k]$. If this is not the case, then η and Λ^k are irrelevant to our construction.

The next lemma describes some useful properties of the out function.

Lemma 3.2 (Out Lemma): Fix $k \leq n$ and $\rho^k \in T^k$. Then:

(i) If $k > 0$ then $\lambda(\text{out}(\rho^k)) = \rho^k$.

(ii) If $k < n$ and $\text{lh}(\rho^k) > 0$, then there is a unique $\rho^{k+1} \in T^{k+1}$ such that $\text{out}(\rho^{k+1}) = \rho^k$.

Proof: (i): $\rho^k = (\rho^k)^- \wedge \langle \text{out}(\rho^k) \rangle$, and $(\text{out}(\rho^k))^-$ is the principal derivative of $(\rho^k)^-$ along $\text{out}(\rho^k)$. Hence (i) follows from Lemma 3.1(ii) (Limit Path).

(ii): Let $\nu^k = (\rho^k)^-$, $\nu^{k+1} = \text{up}(\nu^k)$, and $\rho^{k+1} = \nu^{k+1} \wedge \langle \rho^k \rangle$. Then $\text{out}(\rho^{k+1}) = \rho^k$. To see uniqueness, we note that if $\text{out}(\tau^{k+1}) = \rho^k$, then $\tau^{k+1} = (\tau^{k+1})^- \wedge \langle \rho^k \rangle$, and $\text{up}((\rho^k)^-) = (\tau^{k+1})^-$. Hence $(\tau^{k+1})^- = \nu^{k+1}$ and $\tau^{k+1} = \rho^{k+1}$. ∎

Our next lemma analyzes the behavior of the function λ. Suppose that η extends η^- on T^k in Step 4 of Definition 2.8. We discuss the relationship of the path computed by $\lambda^j(\eta^-)$ to the path computed by $\lambda^j(\eta)$ for all j such that $k \leq j \leq n$. Three types of phenomena can occur, and one will occur for each j. These phenomena induce a partition of [k,n] into three intervals.

There will be a largest $p \geq k$ such that for all $j \in [k,p]$, $\lambda^j(\eta)$ is an immediate successor of $\lambda^j(\eta^-) = \text{up}^j(\eta^-)$. η is not j-switching for any $j \in [k,p]$.

If $p \neq n$, then there will be two possibilities. The first is that $\text{up}^p(\eta^-) = (\lambda^p(\eta))^-$ has infinite outcome along $\lambda^p(\eta)$. Then η will be (p+1)-switching, and will switch $\text{up}^j(\eta^-)$ for all $j \in [p+1,n]$. η will switch the outcome of $\text{up}^{p+1}(\eta^-)$ from infinite along $\lambda^{p+1}(\eta^-)$ to finite along $\lambda^{p+1}(\eta)$. It will follow from (2.10) that for all $j \geq p+1$, η will switch the outcome of $\text{up}^j(\eta^-)$ from infinite along $\lambda^j(\eta^-)$ to finite along $\lambda^j(\eta)$ if j-(p+1) is even, and from finite along $\lambda^j(\eta^-)$ to infinite along $\lambda^j(\eta)$ if j-(p+1) is odd. There will be a largest $s \in (p,n]$ such that for all $j \in [p,s)$, $\text{up}^j(\eta^-)$ will be the principal derivative of $\text{up}^{j+1}(\eta^-)$ along $\lambda^j(\eta)$, and $\lambda^j(\eta)$ will be an immediate successor of $\text{up}^j(\eta^-)$. [p+1,s] is the interval where the second type of phenomenon occurs.

If $s < n$, then the third type of phenomenon begins at s+1 (we set s = p if $\text{up}^p(\eta^-) = (\lambda^p(\eta))^-$ has finite outcome along $\lambda^p(\eta)$, which is the second possibility alluded to in the preceding paragraph, and if this is the case, then η is not switching). Here $\text{up}^s(\eta^-) = (\lambda^s(\eta))^-$ will have finite outcome along $\lambda^s(\eta)$ and will not be the principal derivative of $\text{up}^{s+1}(\eta^-)$ along $\lambda^s(\eta)$. Thus by (2.4), $\lambda^{s+1}(\eta) = \lambda(\text{up}^s(\eta^-))$, so $\lambda^t(\eta) = \lambda^t(\text{up}^s(\eta^-))$ for all $t > s$.

The three types of phenomena mentioned above can be observed if we consider the usual way for satisfying a thickness requirement on T^2. η decides, for η^-, whether to place an additional element x into a set S which is to be either a finite or a thick subset of a recursive set R. We assume that some elements have already been placed in S. If η is the first stage at which we consider x, then we set p = s = 1 if we decide to place x into S and we set p = 1 and s = 2 otherwise. And if η is not the first stage at which we consider x, and we decide to place x into S at η, then we set p = 0 and s = 1.

Lemma 3.3 (λ-Behavior Lemma): Fix $\eta \in T^k$ and assume that a requirement has been assigned to η. Then there are p and s such that $k \le p \le s \le n$ and the following conditions hold:

(i) For all $i \in [k,p]$, $\lambda^i(\eta^-) = up^i(\eta^-) = (\lambda^i(\eta))^-$, if $i < p$ then $\lambda^i(\eta^-)$ is the initial derivative of $\lambda^{i+1}(\eta^-)$ along $\lambda^i(\eta)$, and if $i > k$ then $out(\lambda^i(\eta)) = \lambda^{i-1}(\eta)$.

(ii) For all $i \in (p,s]$, $up^i(\eta^-) = \lambda^i(\eta)^-$, $\lambda^i(\eta^-)|\lambda^i(\eta) = \lambda^i(\eta)^- {}^\wedge\langle\lambda^{i-1}(\eta)\rangle$.

(iii) For all $i \in (s,n]$, $\lambda^i(\eta) = \lambda^i((\lambda^s(\eta))^-)$.

Proof: We verify (i)-(iii) by induction on $lh(\eta)$, analyzing what can happen when requirements are assigned in Step 4 of Definition 2.8 for η^-.

If i = k, then $\eta^- = \lambda^k(\eta^-) = (\lambda^k(\eta))^- = up^k(\eta^-)$. Fix the largest $p \le n$ such that for all $i \in [k,p]$, $\lambda^i(\eta^-) = (\lambda^i(\eta))^- = up^i(\eta^-)$. (i) now follows from (2.4).

If η is nonswitching, then we set s = p and note that (ii) holds vacuously, and that (iii) holds vacuously if s = n. So suppose that s < n. As it is not the case that $\lambda^{s+1}(\eta^-) = (\lambda^{s+1}(\eta))^- = up^{s+1}(\eta^-)$, it follows from (2.4) that $(\lambda^s(\eta))^-$ cannot be an initial derivative of $up^{s+1}(\eta^-)$. As η is nonswitching, it follows from (2.4) that $(\lambda^s(\eta))^- = \lambda^s(\eta^-)$ has finite outcome along $\lambda^s(\eta)$. Thus by (2.4), $\lambda^{s+1}(\eta) = \lambda^{s+1}(\eta^-)$, so for all i > s,

$$\lambda^i(\eta) = \lambda^i(\lambda^{s+1}(\eta)) = \lambda^i(\lambda^{s+1}(\eta^-)) = \lambda^i(\lambda^s(\eta^-)) = \lambda^i((\lambda^s(\eta))^-),$$

and (iii) must also hold.

Suppose that η is switching. By (i), let $\zeta^p = up^p(\eta^-) = (\lambda^p(\eta))^-$, and let $\zeta^{p+1} = up^{p+1}(\eta^-)$. We first show that η is (p+1)-switching, and that ζ^p has infinite outcome along $\lambda^p(\eta)$. For suppose that ζ^p has finite outcome along $\lambda^p(\eta)$ in order to obtain a contradiction. ζ^p cannot be an initial derivative of ζ^{p+1}, else p would have been chosen $\ge$ p+1. But if ζ^p is not an initial derivative of ζ^{p+1}, then as ζ^p has finite outcome along $\lambda^p(\eta)$, $\lambda^{p+1}(\eta) = \lambda(\zeta^p) = \lambda(\lambda^p(\eta^-))$ and so η is nonswitching, contrary to assumption. We conclude that ζ^p has infinite outcome along $\lambda^p(\eta)$, and so by (2.4) and (i), that $\lambda^{p+1}(\eta) = \zeta^{p+1}{}^\wedge\langle\lambda^p(\eta)\rangle$. Thus (ii) holds for i = p+1. Fix the least $s \in (p,n)$, if any, such that the conditions of (ii) fail for i = s+1; otherwise, let s = n. (ii) now follows. (iii) holds vacuously for s = n, so assume that s < n.

As s+1 > p+1, we note that $up^s(\eta^-) = (\lambda^s(\eta))^-$ cannot have infinite outcome along $\lambda^s(\eta)$, else $up^s(\eta^-)$ would be the principal derivative of $up^{s+1}(\eta^-)$ along $\lambda^s(\eta)$, so by (2.4), the condition specified in (ii) would hold for i = s+1. For the same reason, $up^s(\eta^-) = (\lambda^s(\eta))^-$ cannot be the initial derivative of $up^{s+1}(\eta^-)$ along $\lambda^{s+1}(\eta)$. Thus by (2.4), $\lambda^{s+1}(\eta) = \lambda(\lambda^s(\eta)^-)$, so $\lambda^i(\eta) = \lambda^i((\lambda^s(\eta))^-)$ for all $i \in (s,n]$. ■

Definition 3.1: Fix $k \le r \le n$, $\xi \subseteq \eta \in T^k$ and $\Lambda \in [T^k]$. We say that ξ *is (η,r)-true* if $\lambda^j(\xi) \subseteq \lambda^j(\eta)$ for all $j \in [k,r]$, and that ξ *is (Λ,r)-true* if $\lambda^j(\xi) \subset \lambda^j(\Lambda)$ for all $j \in [k,r]$. ξ *is η-true* if ξ is (η,n)-true, and ξ *is Λ-true* if ξ is (Λ,n)-true. ξ *is true* if ξ is ξ-true. ■

The next lemma, which is an easy corollary of Lemma 3.1 (Limit Path), proves the existence of many true nodes.

Lemma 3.4 (True Node Lemma): Fix $k \le r \le n$ and $\eta \subseteq \Lambda^k \in [T^k]$. Then:

(i) Every $\zeta \subseteq \lambda^r(\eta)$ is $(\lambda^r(\eta),r)$-true.

(ii) If σ is (λ(η),r)-true, then the initial derivative of σ along η is (η,r)-true.

(iii) If σ is (λ(η),r)-true, $\xi \subseteq \eta$, and ξ^- is the principal derivative of σ^- along η, then ξ is (η,r)-true.

Proof: (i) follows by definition. (ii) and (iii) follow from clauses (i) and (ii), respectively, of Lemma 3.1 (Limit Path). ■

We now turn our attention to an analysis of the possible ways of extending paths. We first show that we can always take nonswitching extensions.

Lemma 3.5 (Nonswitching Extension Lemma): Fix $\nu \in T^k$. Then either $\nu\hat{}\langle\beta\rangle$ is nonswitching for every $\beta \in T^{k-1}$ such that $\nu\hat{}\langle\beta\rangle \in T^k$ and β^- has infinite outcome along β, or $\nu\hat{}\langle\beta\rangle$ is nonswitching for every $\beta \in T^{k-1}$ such that $\nu\hat{}\langle\beta\rangle \in T^k$ and β^- has finite outcome along β. Moreover, if $\nu\hat{}\langle\beta\rangle$ is a nonswitching extension of ν, then for all $j \ge k$, the $\lambda^j(\nu)$-links and the $\lambda^j(\nu\hat{}\langle\beta\rangle)$-links coincide.

Proof: We proceed by induction on n-k. We note that no node on T^n is switching, and that there are no links on T^n.

Suppose that k < n and ν is not the initial derivative of up(ν) along ν. Fix $\beta \in T^{k-1}$ such that β^- has finite outcome along β. By (2.4), $\lambda(\nu\hat{}\langle\beta\rangle) = \lambda(\nu)$. Thus

$$\lambda^j(\nu\hat{}\langle\beta\rangle) = \lambda^j(\lambda(\nu\hat{}\langle\beta\rangle)) = \lambda^j(\lambda(\nu)) = \lambda^j(\nu)$$

for all $j \in (k,n]$, so $\nu\hat{}\langle\beta\rangle$ is not switching.

Suppose that $k < n$ and ν is the initial derivative of up(ν) along ν. By Lemma 3.1(i) (Limit Path), up(ν) = $\lambda(\nu)$. By induction, either up(ν)^$\langle\xi\rangle$ is nonswitching for all $\xi \in T^k$ such that up(ν)^$\langle\xi\rangle \in T^{k+1}$ and ξ^- has infinite outcome along ξ, or up(ν)^$\langle\xi\rangle$ is nonswitching for all $\xi \in T^k$ such that up(ν)^$\langle\xi\rangle \in T^{k+1}$ and ξ^- has finite outcome along ξ. If dim(ν) $\leq$ k, then $\lambda(\nu\hat{}\langle\beta\rangle)$ = up(ν)^$\langle\nu\hat{}\langle\beta\rangle\rangle$ by (2.9), (2.4) and Lemma 3.1(ii) (Limit Path), so by Lemma 3.1(i) (Limit Path),

$$(3.1)\quad \lambda^j(\nu\hat{}\langle\beta\rangle) = \lambda^j(\lambda(\nu\hat{}\langle\beta\rangle)) = \lambda^j(\mathrm{up}(\nu)\hat{}\langle\nu\hat{}\langle\beta\rangle\rangle) \supseteq \lambda^j(\mathrm{up}(\nu)) = \lambda^j(\lambda(\nu)) = \lambda^j(\nu)$$

for all $j \in (k,n]$, and $\nu\hat{}\langle\beta\rangle$ is nonswitching. Otherwise, dim(ν) > k. If up(ν)^$\langle\xi\rangle$ is nonswitching for every $\xi \in T^k$ such that up(ν)^$\langle\xi\rangle \in T^{k+1}$ and ξ^- has infinite (finite, resp.) outcome along ξ, fix $\beta \in T^{k-1}$ such that $\nu\hat{}\langle\beta\rangle \in T^k$ and β^- has finite (infinite, resp.) outcome along β. By (2.4) and Lemma 3.1(i) (Limit Path), $\lambda(\nu\hat{}\langle\beta\rangle)$ = up(ν)^$\langle\nu\hat{}\langle\beta\rangle\rangle$. Hence (3.1) holds for all $j \in (k,n]$, and $\nu\hat{}\langle\beta\rangle$ is nonswitching.

Fix $j \geq k$. As $\lambda^j(\nu\hat{}\langle\beta\rangle) \supseteq \lambda^j(\nu)$, any primary $\lambda^j(\nu)$-link is a primary $\lambda^j(\nu\hat{}\langle\beta\rangle)$-link. Fix p and s as in the proof of Lemma 3.3 (λ-Behavior). As $\nu\hat{}\langle\beta\rangle$ is a nonswitching extension of ν, it follows from Lemma 3.3(ii) (λ-Behavior) that p = s. If $j \leq p$, then by Lemma 3.3(i) (λ-Behavior), $(\lambda^j(\nu\hat{}\langle\beta\rangle))^- = \lambda^j(\nu)$ and $\lambda^j(\nu)$ is the initial derivative of up($\lambda^j(\nu)$) along $\lambda^j(\nu\hat{}\langle\beta\rangle)$; and if j > p, then by Lemma 3.3(iii) (λ-Behavior), $\lambda^j(\nu\hat{}\langle\beta\rangle)$ = $\lambda^j(\nu)$. In either case, any primary $\lambda^j(\nu\hat{}\langle\beta\rangle)$-link is a primary $\lambda^j(\nu)$-link. The lemma now follows from Step 2 of Definition 2.8. ∎

The next lemma, together with the Nonswitching Extension Lemma, allows us to analyze the computation of λ.

Lemma 3.6 (Switching Lemma): Fix $k \leq n$ and $\xi \in T^k$. Then either:

(i) $\xi\hat{}\langle\beta\rangle$ is switching for all $\beta \in T^{k-1}$ such that β^- has finite outcome along β; or

(ii) $\xi\hat{}\langle\beta\rangle$ is switching for all $\beta \in T^{k-1}$ such that β^- has infinite outcome along β; or

(iii) ξ is the initial derivative of up$^j(\xi)$ along $\lambda^j(\xi)$ for all $j \in [k,n]$.

Proof: We proceed by induction on n-k. (iii) holds if k = n. Suppose that k < n. Let μ be the initial derivative of up(ξ) along ξ. If $\mu \subset \xi$, then by (2.7) and (2.8), up(ξ) must have infinite outcome along $\lambda(\xi)$, so if β is an infinite outcome for ξ, then

$\lambda(\xi)|\lambda(\xi^\frown\langle\beta\rangle)$, so (ii) holds. Suppose that $\mu = \xi$. By Lemma 3.1(i) (Limit Path), $up(\xi) = \lambda(\xi)$. If $up(\xi)^\frown\langle\nu\rangle$ is switching for all ν□$\in T^k$ such that ν^- has finite outcome along ν, then (ii) holds. If $up(\xi)^\frown\langle\nu\rangle$ is switching for all ν□$\in T^k$ such that ν^- has infinite outcome along ν, then (i) holds. Otherwise, by induction, $up(\xi)$ is the initial derivative of $up^j(up(\xi)) = up^j(\xi)$ along $\lambda^j(\lambda(\xi)) = \lambda^j(\xi)$ for all $j \in [k+1,n]$, so (iii) holds. n

Our next lemma shows that if $\Lambda^k \in [T^k]$, then there is a nice approximation to $\lambda^{k+2}(\Lambda^k)$ from Λ^k. This lemma will enable us to show that, under certain circumstances, nodes not along $\lambda^{k+2}(\Lambda^k)$ will not declare many axioms. The lemma is a standard infinite injury lemma, stating that for any node τ which is not along the true path, i.e., not along $\lambda^{k+2}(\Lambda^k)$, there will only be finitely many nodes α along Λ^k which think that τ is along the true path, i.e., such that $\lambda^{k+2}(\alpha) \supseteq \tau$. The machinery which we develop does not require us to look, locally, beyond the interaction of nodes from three consecutive levels. This is also the case with Harrington's approach to $\mathbf{0^{(n)}}$-priority arguments, which uses the Recursion Theorem.

Lemma 3.7 (Infinite Injury Lemma): Fix $k \le n-2$ and $\Lambda^k \in [T^k]$. Let $\Lambda^{k+1} = \lambda(\Lambda^k)$ and $\Lambda^{k+2} = \lambda(\Lambda^{k+1})$. Fix $\tau \in T^{k+2}$ such that $\tau \not\subset \Lambda^{k+2}$. Then $\{\alpha \subset \Lambda^k: \lambda^{k+2}(\alpha) \supseteq \tau\}$ is finite.

Proof: Let $\rho = \Lambda^{k+2} \wedge \tau$. As $\tau \not\subset \Lambda^{k+2}$, $\rho \subset \tau$. Fix $\xi \in T^{k+1}$ such that $\rho^\frown\langle\xi\rangle \subseteq \tau$. By (2.4), if $\alpha \subset \Lambda^k$ and $\lambda^{k+2}(\alpha) \supseteq \tau$, then $\lambda(\alpha) \supseteq \xi$. If $\xi \not\subset \Lambda^{k+1}$, then by Lemma 3.1 (Limit Path), $\{\alpha \subset \Lambda^k: \lambda(\alpha) \supseteq \xi\}$ is finite, and the lemma follows.

Suppose that $\xi \subset \Lambda^{k+1}$. Fix $\eta \neq \xi$ such that $\rho^\frown\langle\eta\rangle \subset \Lambda^{k+2}$. By (2.4), $\eta \subset \Lambda^{k+1}$, so ξ and η are comparable. $\dim(\rho) > k+1$, else $\xi = \eta$ by (2.9), a contradiction. ξ^- cannot have infinite outcome along ξ, else by (2.8), $\eta \subseteq \xi$, so by (2.4) and as $\xi \subset \Lambda^{k+1}$, $\rho^\frown\langle\xi\rangle \subseteq \Lambda^{k+2}$. Hence by (2.4), $\xi \subset \eta$, ξ^- has finite outcome along Λ^{k+1} and η^- has infinite outcome along Λ^{k+1}. By Lemma 3.1 (Limit Path), we can fix $\beta \subset \Lambda^k$ such that for all γ, if $\beta \subseteq \gamma \subset \Lambda^k$ then $\lambda(\gamma) \supseteq \eta$. But then for all such γ, if $\lambda^{k+2}(\gamma) \supseteq \rho$, then $\lambda^{k+2}(\gamma) \supseteq \rho^\frown\langle\eta\rangle$. n

4. Links. In this section, we analyze the effect of link formation on the path generation process. We show that links must be nested, so that the process of removing links by switching paths must be done in an orderly way, no matter how the path is extended. We relate the restraint of $\xi \in T^k$ by an η-link to the satisfaction of whether or not the antiderivatives of ξ lie along the paths computed by η. If η switches ξ, then by (2.10), ξ will be η^--free; we show that ξ is also η-free. And we show that if $up^j(\xi)$ is Λ^j-free for all $j \geq k$ and $k \leq \dim(\xi)$, then ξ has sufficiently many Λ^{k-1}-free derivatives.

We will need to show that we can return a node to the true path by taking switching extensions which change the outcomes at the ends of links. We will need to determine which links must be switched in this way. This determination depends on the fact that η-links on T^k are either nested or disjoint.

Lemma 4.1 (Nesting Lemma): Fix $k \leq n$ and $\eta \in T^k$. Suppose that, for $i \leq 1$, $[\mu_i,\pi_i]$ is an η-link and that $\pi_0 \subset \pi_1$. Then $\mu_1 \subseteq \mu_0$ or $\pi_0 \subset \mu_1$.

Proof: We proceed by induction on n-k. We note that there are no links on T^n, so the lemma holds trivially for $k = n$. Assume that the lemma is true for $k+1$ in place of k.

For each $i \leq 1$, fix the $\lambda(\eta)$-link $[\rho_i,\tau_i]$ from which $[\mu_i,\pi_i]$ is derived if such a link exists; otherwise, $[\mu_i,\pi_i]$ is a primary η-link, and we set $\rho_i = \tau_i = up(\mu_i) = up(\pi_i)$. It follows from the definition of links that for $i \leq 1$, μ_i is the initial derivative of ρ_i along η and π_i is the principal derivative of τ_i along η. As $\pi_0 \subset \pi_1$, it follows from Lemma 3.1(ii) (Limit Path) that $\tau_0 \subset \tau_1$. If $\tau_0 \subset \rho_1$, then by Lemma 3.1(i),(ii) (Limit Path), $\mu_0 \subset \pi_0 \subset \mu_1 \subset \pi_1$ and the lemma holds. Otherwise, $\rho_1 \subseteq \tau_0 \subset \tau_1$. If $\rho_0 = \tau_0$, then $\rho_1 \subseteq \rho_0$. And if $\rho_0 \neq \tau_0$, then by induction, $\rho_1 \subseteq \rho_0$. So in either case, it follows from Lemma 3.1(i) (Limit Path), that $\mu_1 \subseteq \mu_0$. n

Our next definition traces links back to higher trees. Again, a node is *free* if it is not restrained by a link. We will show that all free nodes have all their antiderivatives along the computed paths. The converse of this statement is not true.

Definition 4.1: Fix $k \leq r$□$\leq n$ and $\xi \subset \eta \in T^k$. Let $[\mu,\pi]$ be an η-link. We say that $[\mu,\pi]$ is an *(η,r)-link* if $[\mu,\pi]$ is derived from a primary $\lambda^r(\eta)$-link. $[\mu,\pi]$ is an *$(\eta,\leq r)$-link* if $[\mu,\pi]$ is an (η,j)-link for some $j \in [k,r]$. We say that *ξ is (η,r)-restrained* if there is an (η,r)-link $[\mu,\pi]$ which η-restrains ξ. In this case, we say that *ξ is (η,r)-restrained by $[\mu,\pi]$*. We say that ξ is *(η,r)-free* if ξ is not (η,r)-restrained. We say that *ξ is r-free* if ξ is (ξ,r)-free. If $\Lambda^k \in [T^k]$, then we say that *ξ is (Λ^k,r)-free* if ξ is (η,r)-free for all (Λ^k,r)-true $\eta \supseteq \xi$, and that *ξ is Λ^k-free* if ξ is (Λ^k,n)-free. n

Recall that if $[\mu,\pi]$ is an η-link, then π is *not* restrained by $[\mu,\pi]$. However, as we can have $[\mu,\pi) = [\mu,\delta)$ with $\pi \neq \delta$ for intervals, we used closed interval notation $[\mu,\pi]$ for

η-links to make sure that there is a one-one correspondence between intervals which determine links, and the links themselves.

The next lemma identifies the outcome of a link with the actual outcome of the node ending the link.

Lemma 4.2 (Faithful Outcome Lemma): Fix $\mu \subset \pi \subset \eta \in T^k$ such that $[\mu,\pi]$ is an η-link. Then $[\mu,\pi]$ has finite outcome iff π has finite outcome along η.

Proof: We proceed by induction on n-k. As there are no links on T^n, the lemma holds for $k = n$. Assume that $k < n$.

If $[\mu,\pi]$ is a primary η-link, then π has infinite outcome along η, and by Step 2 of Definition 2.8, $[\mu,\pi]$ has infinite outcome. Otherwise, $[\mu,\pi]$ is derived from some $\lambda(\eta)$-link $[\rho,\tau]$. By Definition 2.1, induction, and Step 2 of Definition 2.8,

π has finite outcome along η iff $up(\pi) = \tau$ has infinite outcome along $\lambda(\eta)$
iff $[\rho,\tau]$ has infinite outcome iff $[\mu,\pi]$ has finite outcome. ∎

The next lemma relates the presence of antiderivatives of a node ξ on a path computed by η to the restraint of ξ by an η-link. It will follow from this lemma that (2.10) implies (2.7).

Lemma 4.3 (Link Analysis Lemma): Fix $k \le r \le n$ and $\xi \subseteq \eta \in T^k$. Then:

(i) (a) If $up(\xi) \not\subseteq \lambda(\eta)$, then there is a primary η-link $[\mu,\pi]$ such that $up(\mu) \neq up(\xi)$ and $[\mu,\pi]$ η-restrains all $\delta \subseteq \eta$ such that $up(\delta) = up(\xi)$.

(b) If $\xi \subset \delta \subset \eta$, $up(\xi) \subseteq up(\delta)$ and $up(\xi) \not\subseteq \lambda(\eta)$, then there is a primary η-link $[\mu,\pi]$ which η-restrains both ξ and δ.

(c) If $[\tilde{\mu},\tilde{\pi}]$ is an η-link which η-restrains the principal derivative ξ of $up(\xi)$, then either there is a primary η-link $[\mu,\pi]$ which η-restrains ξ and $up(\xi) \not\subseteq \lambda(\eta)$, or $[\tilde{\mu},\tilde{\pi}]$ is derived from a $\lambda(\eta)$-link which $\lambda(\eta)$-restrains $up(\xi)$.

(d) If there is a primary η-link $[\mu,\pi]$ which η-restrains the initial derivative of $up(\xi)$, then either $up(\xi) \not\subseteq \lambda(\eta)$ or μ is the initial derivative of $up(\xi)$ along η.

(ii) If $up(\xi) \subseteq \lambda(\eta)$ and $up(\xi)$ is $(\lambda(\eta),r)$-restrained by $[\rho,\tau]$, then ξ is (η,r)-restrained by an η-link $[\mu,\pi]$ derived from $[\rho,\tau]$.

(iii) Suppose that $j \le r$ and $up^j(\xi)$ is $(\lambda^j(\eta),r)$-restrained by $[\rho,\tau]$. Then ξ is $(\eta,\le r)$-restrained. Furthermore, if $up^i(\xi) \subseteq \lambda^i(\eta)$ for all $i \in [k,j]$, then ξ is (η,r)-restrained by an η-link derived from $[\rho,\tau]$.

Proof: (ia): Let $\kappa = up(\xi)\wedge\lambda(\eta)$ and let δ be any derivative of $up(\xi)$ along η. As $up(\xi) \not\subseteq \lambda(\eta)$, $\kappa \subset up(\xi)$. Hence by Lemma 3.1(i),(ii) (Limit Path), κ has an initial derivative $\mu \subset \delta$ and a principal derivative $\pi \subset \eta$ such that $\mu \subset \pi$. By (2.4) and as $\kappa \subset up(\xi)$, $\pi \not\subseteq \delta$. Hence δ is η-restrained by the primary η-link $[\mu,\pi]$. As $up(\mu) = \kappa \subseteq \lambda(\eta)$, $up(\mu) \neq up(\xi)$.

(ib): As $up(\xi) \not\subseteq \lambda(\eta)$, it follows from the proof of (ia) that there is a primary η-link $[\mu,\pi]$ which restrains ξ, and that $up(\mu) = up(\pi) \subset up(\xi)$. Fix $\rho \subseteq \eta$ such that $\rho^- = \pi$. By (2.4), $up(\mu)$ must have finite outcome along $\lambda(\rho)$, but infinite outcome along $up(\xi) \subseteq up(\delta)$. If $\pi \subset \delta$, then by (2.4), δ cannot lie along δ, contradicting (2.7). $\pi \neq \delta$ as $up(\pi) \neq up(\delta)$. But $\pi,\delta \subset \eta$, so $\pi \supset \delta$.

(ic): Suppose that $[\mu,\pi]$ is a primary η-link which η-restrains the principal derivative of $up(\xi)$ along η. We assume that $up(\xi) \subseteq \lambda(\eta)$ and derive a contradiction. We compare the relative locations of $up(\xi)$ and $up(\mu) = up(\pi)$ on T^{k+1}.

First suppose that $up(\mu) \subset up(\xi)$. Fix $\beta \in T^k$ such that $up(\mu)\hat{}\langle\beta\rangle \subseteq up(\xi) \subseteq \lambda(\eta)$. By (2.5), $\beta \subseteq \xi$. By (2.4), β^- is the principal derivative of $up(\mu)$ along η, so $\beta^- = \pi$. But then $\pi \subset \xi$ so $[\mu,\pi]$ cannot η-restrain ξ.

Suppose that $up(\mu) = up(\xi)$. As ξ and π are, respectively, the principal derivatives of $up(\xi)$ and $up(\pi)$ along η and $up(\mu) = up(\pi)$, $\xi = \pi$. So $[\mu,\pi]$ cannot η-restrain ξ.

Suppose that $up(\xi) \subset up(\mu)$. Fix $\beta \in T^k$ such that $up(\xi)\hat{}\langle\beta\rangle \subseteq up(\mu)$. By (2.5), $\beta \subseteq \mu$. By (2.4), β^- is the principal derivative of $up(\xi)$ along $\pi \subseteq \eta$; as $\xi \subset \pi$ and ξ is the principal derivative of $up(\xi)$ along η, it follows from (2.4) that $\beta^- = \xi$. But then $\xi \subset \mu$ so $[\mu,\pi]$ cannot η-restrain ξ.

Suppose that $up(\xi)|up(\mu)$. Let $\tau = up(\xi)\wedge up(\mu)$, and fix α,β such that $\tau\hat{}\langle\alpha\rangle \subseteq up(\xi)$ and $\tau\hat{}\langle\beta\rangle \subseteq up(\mu)$. As ξ is η-restrained by the primary η-link $[\mu,\pi]$ and $up(\xi) \neq up(\mu) = up(\pi)$, $\mu \subset \xi \subset \pi$. By (2.7), $\tau\hat{}\langle\alpha\rangle \subseteq \lambda(\xi)$ and $\tau\hat{}\langle\beta\rangle \subseteq \lambda(\mu),\lambda(\pi)$, contradicting (2.6). Thus $[\mu,\pi]$ cannot η-restrain ξ.

Now suppose that $[\tilde{\mu},\tilde{\pi}]$ is an η-link which η-restrains the principal derivative ξ of $up(\xi)$, but that there is no primary η-link which η-restrains ξ. By (ia), $up(\xi) \subseteq \lambda(\eta)$, so $up(\xi)$, $up(\tilde{\mu})$, and $up(\tilde{\pi})$ are all comparable, and $[up(\tilde{\mu}),up(\tilde{\pi})]$ is a $\lambda(\eta)$-link. As $[\tilde{\mu},\tilde{\pi}]$ η-restrains ξ, $\tilde{\mu} \subseteq \xi \subset \tilde{\pi}$. As ξ is the principal derivative of $up(\xi)$ along η, it follows from Lemma 3.1(ii) (Limit Path) that $up(\tilde{\mu}) \subseteq up(\xi)$. And as $\tilde{\pi}$ is the principal derivative of $up(\tilde{\pi})$ along η, it follows from Lemma 3.1(ii) (Limit Path) that $up(\xi) \subset up(\tilde{\pi})$. Hence $[up(\tilde{\mu}),up(\tilde{\pi})]$ $\lambda(\eta)$-restrains $up(\xi)$.

(id): Suppose that $[\mu,\pi]$ is a primary η-link which η-restrains the initial derivative ν of $up(\xi)$, and that $up(\xi) \subseteq \lambda(\eta)$. We assume that $\mu \neq \nu$ and derive a contradiction. By

Lemma 4.1 (Nesting), there can be no primary ν-link which ν-restrains μ, so by (ia), up(μ) $\subseteq \lambda(\nu)$. By (2.7), up(ν) $\subseteq \lambda(\nu)$, so up(μ) and up(ν) are comparable. By Lemma 3.1(i) (Limit Path), up(μ) $\subseteq$ up(ν). As $[\mu,\pi]$ is a primary η-link and $\nu \subset \pi$, all derivatives of up(μ) which are $\subset \nu$ must have finite outcome along ν, but π has infinite outcome along η. As $\mu \neq \nu$, it follows that up(μ) has infinite outcome along $\lambda(\nu)$, but finite outcome along $\lambda(\eta)$. But up(μ) $\subset$ up(ν) = up(ξ) $\subseteq \lambda(\eta)$, so by (2.7), up(ν) $\subseteq \lambda(\nu),\lambda(\eta)$. Thus up($\mu$) must have the same outcome along both $\lambda(\nu)$ and $\lambda(\eta)$, yielding a contradiction.

(ii): Suppose that $[\rho,\tau]$ is a $\lambda(\eta)$-link which restrains up(ξ) and is derived from the primary $\lambda^r(\eta)$-link $[\kappa,\zeta]$. By Lemma 3.1 (Limit Path), let μ (π, resp.) be the initial (principal, resp.) derivative of ρ (τ, resp.) along η. As up(ξ) $\in [\rho,\tau)$, it follows from (2.10) and Lemma 3.1(i),(ii) (Limit Path) that $\xi \in [\mu,\pi)$, so ξ is η-restrained by $[\mu,\pi]$ which is derived from $[\kappa,\zeta]$.

(iii): Immediate by induction from (i) and (ii). ∎

We now show that free nodes lie along the computed path, so (2.10) implies (2.7).

Lemma 4.4 (Free Implies True Path Lemma): Fix $k \le r \le n$ and $\xi \subset \eta \subseteq \Lambda^k \in [T^k]$ such that ξ is (η,r)-free. Then for all $j \in [k,r]$, $up^j(\xi) \subseteq \lambda^j(\eta)$.

Proof: Immediate from Lemma 4.3(ia),(iii) (Link Analysis). √

When a requirement is assigned to $\delta \in T^0$, then by (2.10) and the process of pulling links down from tree to tree, $up^i(\delta)$ is $\lambda^i(\delta)$-free for all $i \le n$. If $\eta^- = \delta$ and $i \le n$, i.e., η determines an outcome for δ, then η may or may not switch $up^i(\delta)$. We show that in either case, no $\lambda^i(\eta)$-link restrains $up^i(\delta)$ for any $i \le n$. In fact, we show that this happens not only for $\delta \in T^0$, but also for all $k \le n$, $\delta \in T^k$ and all $i \in$ □$[k,n]$.

Lemma 4.5 (Free Extension Lemma): Fix $k \le n$ and $\eta \in T^k$. Then for all $i \in [k,n]$, $up^i(\eta^-)$ is $\lambda^i(\eta)$-free. Furthermore, if $r \in [k+1,n]$ and η is r-switching, then for all $i \in [r,n]$, $\lambda^i(\eta^-) | \lambda^i(\eta)$, $up^i(\eta^-) = \lambda^i(\eta^-) \wedge \lambda^i(\eta)$, and $up^i(\eta^-)$ has finite outcome along $\lambda^i(\eta)$ iff $up^i(\eta^-)$ has infinite outcome along $\lambda^i(\eta^-)$.

Proof: Fix p and s as in Lemma 3.3 (λ-Behavior). First suppose that η is nonswitching. Then by Lemma 3.3 (λ-Behavior), p = s, $up^i(\eta^-) = (\lambda^i(\eta))^- = \lambda^i(\eta^-)$ if $i \in [k,s]$, and $\lambda^i(\eta) = \lambda^i(up^s(\eta^-)) = \lambda^i(\lambda^s(\eta^-)) = \lambda^i(\eta^-)$ if $i \in [s+1,n]$. It follows from Lemma 3.5 (Nonswitching Extension) that for all $i \in [k,n]$, the primary $\lambda^i(\eta)$-links and the primary $\lambda^i(\eta^-)$-links coincide. By (2.10), for all i ∈ [k,n], $up^i(\eta^-)$ is $\lambda^i(\eta^-)$-free. Hence for all i ∈ [k,n], $up^i(\eta^-)$ must be $\lambda^i(\eta)$-free.

Now suppose that η is switching. We first show that for all i ∈ [k,n], $up^i(\eta^-) \subseteq \lambda^i(\eta)$. If i ∈ [k,p], then by Lemma 3.3 (λ-Behavior), $up^i(\eta^-) = (\lambda^i(\eta))^- = \lambda^i(\eta^-)$; so as, by (2.7), $up^i(\eta^-) \subseteq \lambda^i(\eta^-)$, it must be the case that $up^i(\eta^-) \subseteq \lambda^i(\eta)$. If i ∈ [p+1,s], then by Lemma 3.3 (λ-Behavior), $\lambda^i(\eta^-)|\lambda^i(\eta)$ and $up^i(\eta^-) = \lambda^i(\eta^-)\wedge\lambda^i(\eta)$; so $up^i(\eta^-) \subseteq \lambda^i(\eta)$. If i ∈ [s+1,n], then by Lemma 3.3 (λ-Behavior), $\lambda^i(\eta) = \lambda^i(up^s(\eta^-))$. By (2.7), $up^i(up^s(\eta^-)) \subseteq \lambda^i(up^s(\eta^-))$. Hence $up^i(\eta^-) = up^i(up^s(\eta^-)) \subseteq \lambda^i(\eta)$.

We next show that for all i ∈ [k,n], $up^i(\eta^-)$ is both $\lambda^i(\eta^-)$-free and $\lambda^i(\eta)$-free. By (2.10), $up^i(\eta^-)$ is $\lambda^i(\eta^-)$-free. Suppose that $up^i(\eta^-)$ is not $\lambda^i(\eta)$-free for some i, which we fix in order to obtain a contradiction. Then there is a $\lambda^i(\eta)$-link which restrains $up^i(\eta^-)$. We note that for any m ∈ [k,n] and any $\lambda^m(\eta)$-link $[\mu^m,\pi^m]$, $\pi^m \subset \lambda^m(\eta)$ and π^m is not $\lambda^m(\eta)$-restrained by $[\mu^m,\pi^m]$. By Lemma 3.3(i),(ii) (λ-Behavior), if $i \le s$ then $up^i(\eta^-) = (\lambda^i(\eta))^-$. Hence it must be the case that $i > s$. But then by Lemma 3.3 (λ-Behavior), $\lambda^i(\eta) = \lambda^i(up^s(\eta^-))$, and by (2.10), $up^i(up^s(\eta^-))$ is $\lambda^i(up^s(\eta^-))$-free. Hence $up^i(\eta^-) = up^i(up^s(\eta^-))$ is $\lambda^i(\eta)$-free, yielding the desired contradiction.

Finally, we must show that for all i ∈ [p+1,n], $\lambda^i(\eta^-)|\lambda^i(\eta)$, $up^i(\eta^-) = \lambda^i(\eta^-)\wedge\lambda^i(\eta)$, and $up^i(\eta^-)$ has finite outcome along $\lambda^i(\eta)$ iff $up^i(\eta^-)$ has infinite outcome along $\lambda^i(\eta^-)$. This follows from Lemma 3.3(ii) (λ-Behavior) for i ∈ [p+1,s]. We now proceed by induction on i ∈ [s+1,n]. By (2.7) and as $up^i(\eta^-) \subseteq \lambda^i(\eta)$, $up^i(\eta^-) \subseteq \lambda^i(\eta^-)\wedge\lambda^i(\eta)$. Suppose that $up^i(\eta^-) \subset \lambda^i(\eta^-)\wedge\lambda^i(\eta)$ in order to obtain a contradiction. Fix $\rho^i \subseteq \lambda^i(\eta^-)\wedge\lambda^i(\eta)$ such that $(\rho^i)^- = up^i(\eta^-)$. Then by (2.4), $(out(\rho^i))^-$ is the principal derivative of $up^i(\eta^-)$ along $out(\rho^i) \subseteq \lambda^{i-1}(\eta^-)\wedge\lambda^{i-1}(\eta)$. Now $(out(\rho^i))^- \subset up^{i-1}(\eta^-)$, else either $\lambda^{i-1}(\eta^-)$ and $\lambda^{i-1}(\eta)$ are comparable, or $up^{i-1}(\eta^-) \neq \lambda^{i-1}(\eta^-)\wedge\lambda^{i-1}(\eta)$. Hence by (2.8), $(out(\rho^i))^-$ is the initial derivative of $up^i(\eta^-)$ along both $\lambda^{i-1}(\eta^-)$ and $\lambda^{i-1}(\eta)$, and $(\rho^i)^-$ has infinite outcome along both $\lambda^i(\eta^-)$ and $\lambda^i(\eta)$. But then all derivatives of $up^i(\eta^-)$ must have finite outcome along both $\lambda^{i-1}(\eta^-)$ and $\lambda^{i-1}(\eta)$. As $up^{i-1}(\eta^-)$ is such a derivative, we have contradicted our induction hypothesis.

Suppose that $up^i(\eta^-)$ has finite outcome along $\lambda^i(\eta)$ ($\lambda^i(\eta^-)$, resp.). Then there is a derivative ν^{i-1} of $up^i(\eta^-)$ which has infinite outcome along $\lambda^{i-1}(\eta)$ ($\lambda^{i-1}(\eta^-)$, resp.). Since $up^{i-1}(\eta^-)$ is $\lambda^{i-1}(\eta)$-free ($\lambda^{i-1}(\eta^-)$-free, resp.), $\nu^{i-1} = up^{i-1}(\eta^-)$. By induction, $up^{i-1}(\eta^-)$ has finite outcome along $\lambda^{i-1}(\eta^-)$ ($\lambda^{i-1}(\eta)$, resp.). As $up^{i-1}(\eta^-)$ is $\lambda^{i-1}(\eta^-)$-free ($\lambda^{i-1}(\eta)$-free, resp.), all derivatives of $up^i(\eta^-)$ along $\lambda^{i-1}(\eta^-)$ ($\lambda^{i-1}(\eta)$, resp.) must have finite outcome along $\lambda^{i-1}(\eta^-)$ ($\lambda^{i-1}(\eta)$, resp.). Hence by (2.4), $up^i(\eta^-)$ has infinite outcome along $\lambda^i(\eta^-)$ ($\lambda^i(\eta)$, resp.). Now $up^i(\eta^-)$ cannot have infinite outcome along both $\lambda^i(\eta^-)$ and $\lambda^i(\eta)$, else by (2.4), $up^{i-1}(\eta^-)$ would have finite outcome along both $\lambda^{i-1}(\eta^-)$ and $\lambda^{i-1}(\eta)$ contrary to our induction assumption. Hence $p^i(\eta^-)$ has finite outcome along $\lambda^i(\eta)$ iff $up^i(\eta^-)$ has infinite outcome along $\lambda^i(\eta^-)$. ∎

The nodes which are Λ^k-free are the nodes which have the responsibility to determine the value of most of the axioms. However, we will be unable to recursively identify these nodes, and so, will be unable to prevent other nodes from defining a large number of axioms. We will have to show that the nodes to which we want to assign responsibility for defining most of the axioms can automatically transfer this responsibility to their derivatives. In order for this transfer to occur, we will need to show that principal derivatives of free nodes are free, and that if a free node has infinite outcome along the true path Λ^{k+1} through T^{k+1}, then it has infinitely many free derivatives along the true path Λ^k through T^k. We show this in our next lemma. We will later show that these nodes also have the opportunity to correct many axioms defined by other nodes. We first note an important fact, whose proof we leave to the reader.

(4.1) Fix $k \le r \le n$, $\beta \subseteq \Lambda^k \in [T^k]$, and $\mu \subset \nu \subset \eta$ such that η is (β,r)-true. Then $[\mu,\nu]$ is a (β,r)-link iff $[\mu,\nu]$ is an (η,r)-link.

Lemma 4.6 (Free Derivative Lemma): Fix $k < n$ and $\Lambda^k \in [T^k]$. For all $r \in [k,n]$, let $\Lambda^r = \lambda^r(\Lambda^k)$. Suppose that $\sigma \subset \Lambda^{k+1}$ is Λ^{k+1}-free. Then:

(i) If $\delta \subset \Lambda^k$ is the principal derivative of σ along Λ^k, then δ is Λ^k-free.

(ii) If σ has infinite outcome along Λ^{k+1} and $\dim(\sigma) > k$, then there are infinitely many Λ^k-free derivatives of σ.

Proof: (i): By Lemma 3.1(ii) (Limit Path), fix the principal derivative δ of σ along Λ^k. Suppose that δ is Λ^k-restrained by a Λ^k-link $[\mu,\pi]$ in order to obtain a contradiction. By repeated applications of Lemma 3.1(i) (Limit Path) we may fix $\eta \subset \Lambda^k$ such that $\pi \subset \eta$ and η is Λ^k-true. By (4.1), $[\mu,\pi]$ is an η-link. Without loss of generality, we may assume that $[\mu,\pi]$ is a primary η-link. (Else by Lemma 4.3(ic) (Link Analysis), σ would be $\lambda(\eta)$-restrained by a $\lambda(\eta)$-link. But then $\lambda(\eta)$ is Λ^{k+1}-true, so by (4.1), σ would not be Λ^{k+1}-free, a contradiction.) Let $up(\mu) = up(\pi) = \tau$. By (2.7) and assumption, $\sigma \subseteq \lambda(\delta),\lambda(\eta)$, so by (2.6), $\sigma \subseteq \lambda(\pi)$. By (2.7), $\tau \subseteq \lambda(\pi)$; hence σ and τ are comparable. By

Lemma 3.1(ii) (Limit Path) and as $\delta \subset \pi$, it must be the case that $\sigma \subset \tau$. δ cannot be the initial derivative of σ along δ, else by Lemma 3.1(i) (Limit Path) and as $\mu \subseteq \delta$, $\tau \subseteq \sigma$, yielding a contradiction. Let ν be the initial derivative of σ along η. By Lemma 3.1(i) (Limit Path), $\nu \subset \mu \subset \delta$, and we have already shown that $\delta \subset \pi$. But $[\nu,\delta]$ and $[\mu,\pi]$ are η-links, contradicting Lemma 4.1 (Nesting).

(ii): We note by (i) that if $\zeta \subset \Lambda^n$, then for all j such that $k \le j \le n$, the principal derivative ζ^j of ζ along Λ^j is Λ^j-free. As $lh(\Lambda^k) = \infty$, it follows inductively from Lemma 3.1(iv) (Limit Path) that $lh(\Lambda^j) = \infty$ for all j such that $k \le j \le n$. Hence there are infinitely many $\zeta \subset \Lambda^n$ such that ζ^j extends $up^j(\sigma)$ for all j such that $k \le j \le n$. Fix such a node ξ. It suffices to show that σ has a free derivative along Λ^k which extends ζ^k.

By (2.4) and Lemma 3.1(ii) (Limit Path), if we fix $\gamma^k \subset \Lambda^k$ such that $(\gamma^k)^- = \zeta^k$, then γ^k is Λ^k-true. Hence by (4.1), for all j such that $k \le j \le n$ and all $\delta^j \subseteq \lambda^j(\gamma^k)$, δ^j is Λ^j-free iff δ^j is $\lambda^j(\gamma^k)$-free. In particular, σ is $\lambda(\gamma^k)$-free. It thus follows easily from Lemma 4.4 (Free Implies True Path) and hypothesis that σ is γ^k-consistent. Furthermore, as ζ^j is the principal derivative of ζ^{j+1} along $\lambda^j(\gamma^k)$ for all $j \in [k,n-1]$ and $(\gamma^k)^- = \zeta^k$, it follows from Lemma 3.3 (λ-Behavior) that either γ^k is switching, or $(\gamma^k)^-$ is the initial derivative of ζ along Λ^k. In either case, we set $\rho = \langle\,\rangle$ in Definition 2.8, Step 4 when we are ready to assign a requirement to γ^k. As $\sigma \subset \lambda(\gamma^k) \subset \Lambda^{k+1}$, it follows from (2.4) and (2.6) that we will take nonswitching extensions in Definition 2.8, Step 4, beginning at γ^k, and reach a node β^k at which σ is the shortest node eligible to determine a derivative along Λ^k. By Lemma 3.5 (Nonswitching Extension), no new links are formed when nonswitching extensions are taken. Since $\sigma \subset \lambda(\gamma^k)$, it will be the case that $\lambda(\beta^k) = \lambda(\gamma^k)$, and so, that σ is β^k-free and β^k-consistent. By Definition 2.8, Step 4, we define $up(\beta^k) = \sigma$, and β^k will be Λ^k-true and Λ^k-free. Hence by (4.1), β^k will be a Λ^k-free derivative of σ. ∎