

Collapse in a Transfinite Hierarchy of Turing Degrees

by

Ellen Hammatt

A thesis
submitted to the Victoria University of Wellington
in fulfilment of the
requirements for the degree of
Master of Science
in Mathematics.

Victoria University of Wellington

2021

Abstract

In [2], Downey and Greenberg use the ordinals below ϵ_0 to bound the number of mind-changes of computable approximations of Δ_2^0 functions. This gives rise to a new transfinite hierarchy in the c.e. degrees; the totally α -c.a. degrees. This hierarchy is significant because it unifies the combinatorics of many constructions as well as giving natural definability results in the c.e. Turing degrees. We study the structure of this hierarchy; in particular we investigate collapse in upper cones. We give a proof in which we build a c.e. set using a strategy tree to show there is no uniform way to find a maximal totally ω^2 -c.a. degree above a given totally ω -c.a. degree. Then we discuss extensions of this result.

Acknowledgments

First I would like to thank my supervisors, Noam and Rod, for their excellent guidance and advice, with a particular thanks to Noam for his continuous support in my times of stress.

Thank you to my fellow students: Malcolm for his help with TikZ and Linus for all his words of encouragement and advice.

I would also like to thank my parents and my brother for their support throughout my studies. Along with a special mention to my father for his help with proofreading.

Contents

1	Introduction	1
1.1	α -c.a.	2
1.2	Hierarchy Collapse	5
1.3	Maximal totally α -c.a. degrees	6
2	Maximality	9
2.1	Maximal totally ω -c.a. degrees	9
2.2	Maximal totally ω^ω -c.a. degrees in upper cones	12
2.3	Maximal totally ω^2 -c.a. degrees in upper cones	14
3	Theorem	21
3.1	Glossary	21
3.2	Technical Discussion	24
3.3	Tree of Strategies	31
3.3.1	Assigning Requirements	32
3.3.2	Types of Nodes	33
3.4	Construction	35
3.4.1	Verification	39
4	Extending Results	67
4.1	Two c.e. sets	67
4.2	Three c.e. sets	73

Chapter 1

Introduction

Computability theory seeks to understand the relative complexity of mathematical objects. We use reductions such as Turing reducibility to formalise the idea of relative information content. Another tool to further our understanding is to calibrate objects into various hierarchies; the arithmetical hierarchy is an example of this. It is significant that the arithmetical hierarchy aligns with the Turing jumps; this gives alignment between syntactical and algorithmic complexity. In [2], Downey and Greenberg use the notion that a degree can compute complicated objects if and only if some functions in the degree are difficult to approximate. Computability theory allows us to formalise this notion. In [2], Downey and Greenberg use the relative complexity of functions that bound the *mind-change* function of computable approximations of Δ_2^0 functions to give rise to a new transfinite hierarchy in the c.e. degrees. This hierarchy is significant because it unifies the combinatorics of many constructions as well as giving *natural* definability results in the c.e. Turing degrees. We give examples of this in the following section.

It is of interest to investigate the structure of this hierarchy. We look for levels of the hierarchy at which no new degrees exist; we say that at such levels the hierarchy *collapses*. Notions of *maximality* in this hierarchy are of particular interest because the existence of maximal elements of classes is rare in the c.e. degrees. It is also significant because the maximal totally

ω -c.a. degrees¹ form a naturally definable antichain in the c.e. degrees. In this thesis we investigate collapse in *upper cones* by considering the existence of maximal degrees above another degree.

In particular we investigate the question of whether there exists a degree \mathbf{d} that is totally ω -c.a. such that no degree above \mathbf{d} is maximal totally ω^2 -c.a. The result we find is that there is no uniform way to find a maximal totally ω^2 -c.a. degree above a given totally ω -c.a. degree. Priority arguments using strategy trees are used to answer such questions. We now provide the context to understand the meaning of this question we are considering.

1.1 α -c.a.

We start with the Shoenfield Limit Lemma [9] which states that a function $f : \omega \rightarrow \omega$ is Δ_2^0 (or equivalently, computable from $\mathbf{0}'$) if and only if it has a *computable approximation*. A computable approximation is a uniformly computable sequence $\langle f_s \rangle_{s < \omega}$ such that for all x and for all but finitely many stages s , $f_s(x) = f(x)$. Every computable approximation has an associated *mind-change function* which is the function that tells us how many times the computable approximation *changes its mind* on a given x . Formally $m^{(f_s)}(x) = \#\{s : f_{s+1}(x) \neq f_s(x)\}$. Now consider the complexity of functions which bound the mind-change function. The idea is that a function with approximations that has many mind changes requires greater computational power. Functions with a computable function bounding its mind-change function are called ω -computably approximable (ω -c.a.). The idea for the new hierarchy is to extend this to find a general definition of α -c.a for an ordinal α in such a way to give meaningful calibration of computational power.

For an ordinal α , an α -computable approximation of a function f is a computable approximation $\langle f_s \rangle_{s < \omega}$ of f , equipped with a uniformly computable sequence $\langle o_s \rangle_{s < \omega}$ of functions from ω to α such that for all x and s ,

¹see next section for formal definition.

$o_{s+1}(x) \leq o_s(x)$ and if $f_{s+1}(x) \neq f_s(x)$ then $o_{s+1}(x) < o_s(x)$. The idea is that we ‘count’ down along α as the computable approximation changes its mind. Then the ‘longer’ α is, the more the approximation can change its mind.

Notice that we need to be careful about which computable copy of the ordinal α we are using. Ershov proved that every Δ_2^0 function is ω -c.a. for some computable copy of ω [6, 7, 8]. Ershov’s idea was that the complexity of a computable approximation of f can be encoded into the presentation of ω . So if we just allow any presentation of α the hierarchy obtained from this definition is useless for calibration below $\mathbf{0}'$ as it collapses at ω , so it does not help us further understand the relative complexity. Ershov’s result arises from the existence of ‘bad’ copies of ω . For example, there is a presentation of ω where the successor function is not computable. Ershov [7], and Epstein, Haas and Kramer [5] used Kleene’s system of ordinal notations. But under this definition, every Δ_2^0 function is ω^2 -c.a. because it is possible that we are not able to compute which copy of ω a given element lies in. Therefore this does not suffice for study below $\mathbf{0}'$. This motivates the desire to find a well-behaved class of presentations of ordinals in which these things are computable. Downey and Greenberg found that Cantor’s normal form gives the required information for ordinals up to ϵ_0 .

Every ordinal α has a unique expression as the sum: $\omega^{\alpha_1}n_1 + \omega^{\alpha_2}n_2 + \dots + \omega^{\alpha_k}n_k$, where $n_i < \omega$ are non-zero and $\alpha_1 > \alpha_2 > \dots > \alpha_k$; this is called the Cantor normal form of α . $\epsilon_0 = \sup\{\omega, \omega^\omega, \omega^{\omega^\omega}, \dots\}$ is the least ordinal γ such that $\omega^\gamma = \gamma$. So for all ordinals $\alpha < \epsilon_0$, the ordinals appearing as the exponents in the Cantor normal form of α are all strictly smaller than α . A presentation of α is *canonical* if we can computably find the Cantor normal form of α . Now f is α -c.a. if there is an α -computable approximation for a canonical computable presentation of α . Note that any two canonical copies of α are computably isomorphic, and so the notion of α -c.a. does not depend on which canonical copy we use. Also note that we are able to find canonical computable presentations of all $\alpha \leq \epsilon_0$. Details can be found in [2].

Now notice that this measure of complexity does not align with Turing

reducibility. In terms of the definition of α -c.a. that we just defined, a c.e. set is very simple because the bound on the mind-change function is low as there is only at most one change on each input. But c.e. sets can compute complex functions; hence they can be complicated in terms of Turing degrees, such as \emptyset' . We are motivated to force alignment by making the following definition: a Turing degree \mathbf{d} is *totally α -c.a.* if every function $f \in \mathbf{d}$ is α -c.a. This is the definition that yields the new hierarchy.

Now an important fact that will be used in this thesis is the ability to effectively list the α -c.a. functions. To do this we use *tidy $(\alpha + 1)$ -computable approximations*. These are approximations $\langle f_s, o_s \rangle$ such that for all n , $f_0(n) = 0$ and for all n and s , if $o_s(n + 1) < \alpha$ then $o_s(n) < \alpha$. We may think of these as *partial α -computable approximations*. We can think of $o_s(n) = \alpha$ as $o_s(n) \uparrow$; at such a stage the ordinal in α has not yet been declared. Such an approximation is *eventually α -computable* if for all n there is a stage s such that $o_s(n) < \alpha$. Now there is a computable list $\langle \langle f_s^i, o_s^i \rangle_{s < \omega} \rangle_{i < \omega}$ of tidy $(\alpha + 1)$ -computable approximations such that letting $f^i = \lim_s f_s^i$, the sequence $\langle f^i \rangle_{i < \omega}$ lists the α -c.a. functions. This means that for every α -c.a. function there is an i such that $\langle \langle f_s^i, o_s^i \rangle_{s < \omega} \rangle$ is eventually α -computable and $f^i = f$. See [2] for the full details.

Note that equipped with this listing of tidy $(\alpha + 1)$ -computable approximations, it follows that there is a uniformly $\mathbf{0}'$ -computable listing of all α -c.a. functions. Using this sequence we are able to construct a $\mathbf{0}'$ -computable function which dominates every α -c.a. function; hence every totally α -c.a. degree is low_2 .

An *array computable degree* is a degree such that every function in the degree has a computable approximation such that its mind-change function is bounded by the identity function. Notice that this is a uniform version of the notion of totally ω -c.a. This class is significant because it captures the combinatorics of a wide range of constructions, for example: a degree is array non-computable if and only if it bounds a disjoint pair of c.e. sets which have no separator computing $\mathbf{0}'$ [4]. There are many more examples

of equivalence given in [2]. This motivates the hierarchy as the totally α -c.a. degrees allow us to generalise array non-computability.

The totally ω -c.a. degrees naturally align with another level of permitting hence also capture the dynamics of a number of constructions. A c.e. degree is not totally ω -c.a. if and only if it bounds a (weak) critical triple in the c.e. degrees [3]. This shows that the ω -c.a. degrees are naturally definable in the c.e. degrees. The totally ω -c.a. degrees also capture the dynamics of many constructions; for example, in [2] the class of not totally ω -c.a. degrees relates to presentations of left c.e. reals. See [2] for details.

Another class of particular significance is the totally $< \omega^\omega$ -c.a. degrees. A totally $< \omega^\omega$ -c.a. degree is such that every function in it is ω^n for some $n < \omega$. Now it is shown in [2] that a c.e. degree bounds a copy of the 1-3-1 lattice if and only if it is not totally $< \omega^\omega$ -c.a. This shows that totally $< \omega^\omega$ -c.a. degrees are naturally definable in the c.e. degrees. Now totally $< \omega^\omega$ -c.a. degrees also capture the dynamics of constructions; in particular they characterise non-uniform arguments with some ω levels of non-uniformity.

1.2 Hierarchy Collapse

Now that we have defined the hierarchy, we investigate the structure of this hierarchy. We ask, at what levels does the hierarchy collapse? Is there a level β such that every degree that is totally β -c.a. is also totally α -c.a. for some $\alpha < \beta$? At such levels β no new degrees exist hence the hierarchy has collapsed at this level. We call a degree *properly totally α -c.a.* if it is totally α -c.a. and not totally β -c.a. for any $\beta < \alpha$. Note that we are concerned with c.e. degrees, so all degrees mentioned in following theorems are c.e. degrees. The theorem below tells us that the levels which do not collapse are powers of ω .

Theorem 1.1. ([2, Thm.3.6]) Let $\alpha \leq \epsilon_0$. There is a properly totally α -c.a. c.e. degree if and only if α is a power of ω .

Therefore the first proper levels of the hierarchy are the totally ω -c.a. degrees, then the totally ω^2 -c.a. degrees, ... totally ω^ω -c.a. degrees, totally $\omega^{\omega+1}$ -c.a. degrees, From now on all ordinals are powers of ω .

Now we consider the structure of the hierarchy in upper cones. This is where we investigate the structure of the hierarchy above a specific degree \mathbf{d} . So we ask if there is a β such that every degree $\mathbf{c} >_T \mathbf{d}$ that is totally β -c.a. is also α -c.a. for some $\alpha < \beta$. The following is a result shown in [1]:

Theorem 1.2. Let $\alpha < \epsilon_0$ be a power of ω . Above every totally α -c.a. degree there is a totally $(\alpha \cdot \omega^3)$ -c.a. degree which is not totally α -c.a.

In particular, let $\alpha = \omega$. Then this theorem tells us that above every totally ω -c.a. degree there is a totally ω^4 -c.a. degree which is not ω -c.a. So we ask, can we make this stronger? Can this degree be made ω^3 -c.a. or even ω^2 -c.a.? Li Ling Ko (personal communication) recently showed that above every ω -c.a. degree there is a properly totally ω^2 -c.a. degree, with a non-uniform argument using results from [1]. It is not clear how this generalises, for example, is there a properly totally ω^3 -c.a. degree above every ω^2 -c.a. degree? The question of whether there is a uniform way to find a properly totally ω^2 -c.a. degree above a totally ω -c.a. degree is still open.

1.3 Maximal totally α -c.a. degrees

A degree \mathbf{d} is *maximal* totally α -c.a. if \mathbf{d} is totally α -c.a. and no $\mathbf{b} >_T \mathbf{d}$ is totally α -c.a. The following theorem is proved in [2]; this tells us that maximal totally α -c.a. degrees exist at every proper level of the hierarchy. In the next chapter we will discuss how we can construct such degrees.

Theorem 1.3. ([2, Thm.4.1]) If $\alpha \leq \epsilon_0$ is a power of ω , then there is a maximal totally α -c.a. c.e. degree.

We now ask, could there be a totally ω -c.a. degree \mathbf{d} which is a maximal totally ω^2 -c.a. degree? This would mean there is no degree above \mathbf{d} that

is totally ω^2 -c.a. but not totally ω -c.a. Which implies there is collapse in the upper cone. The following theorem, proved in [2], says there is no such degree:

Theorem 1.4. ([2, Thm.4.12]) Let $\beta < \epsilon$. Every totally ω^β -c.a. c.e. degree is bounded by a strictly greater totally $\omega^{\beta+1}$ -c.a. c.e. degree.

So we know that no totally α -c.a. degree can be maximal totally β -c.a. for $\beta > \alpha$. As a result we know that every maximal totally α -c.a. degree is properly totally α -c.a. Hence for $\alpha < \beta \leq \epsilon_0$, the ability to find a maximal totally β -c.a. degree above every totally α -c.a. degree, implies that level β does not collapse in upper cones. So clearly it is of interest to find maximal totally β -c.a. degrees above totally α -c.a. degrees. The following theorem was proved in [1]:

Theorem 1.5. Let $\alpha < \beta \leq \epsilon_0$ be powers of ω , and suppose that $\beta \geq \alpha^\omega$. Then every totally α -c.a. degree is bounded by a maximal totally β -c.a. degree.

We will discuss the idea of how to construct such degrees in the next chapter. This theorem shows that there is no collapse in upper cones for levels $\beta \geq \alpha^\omega$. It is still unknown whether there is collapse in upper cones for $\beta < \alpha^\omega$. In particular we ask, is every totally ω -c.a. degree bounded by a maximal totally ω^2 -c.a. degree?

In [1] it was proved that there is a totally ω -c.a. degree such that it is not bounded by any maximal totally ω -c.a. degree. Now in this thesis we investigate the question: does there exist a c.e. set A such that A is totally ω -c.a. and no c.e. set $B \geq_T A$ is maximal totally ω^2 -c.a. It is currently unknown whether this is true or not. In Chapter 3 we prove the following theorem:

Theorem 1.6. There is a computable function f such that for all k the following hold:

- $W_{f(k)}$ is totally ω -c.a.

- $W_k \oplus W_{f(k)}$ is not maximal totally ω^2 -c.a.

This implies the following corollary which means that a maximal totally ω^2 -c.a. degree above a given totally ω -c.a. degree cannot be found uniformly:

Corollary 1.6.1. There is no computable function g such that for all m , if W_m is totally ω -c.a. then $W_m \oplus W_{g(m)}$ is maximal totally ω^2 -c.a.

Proof. Suppose there is such a computable function g . Let f be the function given by Theorem 1.6. By the recursion theorem, there is an m such that $f(g(m)) = m$. Now by applying the function f on input $g(m)$: $W_{f(g(m))}$ is totally ω -c.a. and $W_{g(m)} \oplus W_{f(g(m))}$ is not maximal totally ω^2 -c.a. But $f(g(m)) = m$, so this means W_m is totally ω -c.a. and $W_{g(m)} \oplus W_m$ is not maximal totally ω^2 -c.a. Now W_m is totally ω -c.a., so by applying g on input m : $W_m \oplus W_{g(m)}$ is maximal totally ω^2 -c.a. This is a contradiction; hence there is no such computable function g . \square

Chapter 2

Maximality

2.1 Maximal totally ω -c.a. degrees

First we will discuss how to construct a maximal totally ω -c.a. degree. We build a c.e. set D such that the degree of D is maximal totally ω -c.a. To ensure $\deg_T(D)$ is totally ω -c.a. we meet the following set of requirements for all functionals Γ :

- N_Γ : If $\Gamma(D)$ is total then $\Gamma(D)$ is ω -c.a.

To ensure $\deg_T(D)$ is maximal we ensure that every c.e. set above D is not totally ω -c.a. So for each c.e. set W , we build a functional $\Psi_W(D, W)$ with the aim of showing that either $W \leq_T D$ or $\Psi_W(D, W)$ is not ω -c.a. Let $\langle f^k, o^k \rangle$ be the sequence of tidy $(\omega + 1)$ -computable approximations. As discussed in Chapter 1, for every ω -c.a. function there is some k such that $\langle f^k, o^k \rangle$ is eventually ω -computable and $f = f^k$. Then for all c.e. sets W and all $k < \omega$ we meet the following set of requirements:

- $P_{W,k}$: If $\langle f^k, o^k \rangle$ is eventually ω -computable then either $W \leq_T D$ or $\Psi_W(D, W) \neq f^k$.

This construction requires a strategy tree. A node working for an N_Γ requirement is denoted τ . To meet an N_Γ requirement we need to come up with an ω -computable approximation for $\Gamma(D)$. We can achieve this if we are

able to devise a strategy to compute a bound for the number of D changes that could possibly injure the computation $\Gamma(D, x)$ for each x . So at each expansionary stage we declare a bound for a new x and we need to ensure that this bound is correct. We build D , so any number that enters D was enumerated into D by some $P_{W,k}$ requirement; the idea is to restrict which requirements can injure a $\Gamma(D, x)$ computation.

Nodes working for a $P_{W,k}$ requirement are denoted σ . The strategy to meet a $P_{W,k}$ requirement is to appoint a follower p and whenever we see $\Psi_W(D, W, p)[s] = f_s^k(p)$ we enumerate $\psi_s(p)$ into D_{s+1} . But recall $\langle f^k, o^k \rangle$ is a tidy $(\omega + 1)$ -computable approximation so after this enumeration at stage s it is possible that there is a stage t where $f_t^k(p) \neq f_s^k(p)$. Then it is possible that there is some stage $r \geq t$ such that $\Psi_W(D, W, p)[t] = f_t^k(p)$; hence the diagonalisation attempt at stage s has failed and we need to enumerate $\psi_r(p)$ into D_{s+1} . Now notice that each time this happens there has been a change in $f^k(p)$; hence $o^k(p)$ must have decreased since this is a tidy $(\omega + 1)$ -computable approximation. Then we know that this will not happen infinitely often because $o^k(p)$ has to decrease after each failed diagonalisation attempt; hence this allows us to successfully diagonalise and achieve the requirement that $\Psi_W(D, W) \neq f^k$. Now we not only know that we can successfully diagonalise but once we see $o_s^k(p) < \omega$ we have a bound on how many numbers we will need to enumerate into D in order to achieve diagonalisation.

To meet N_Γ requirements we can only allow a follower to injure a $\Gamma(D, x)$ computation if it had seen $o_s^k(p) < \omega$ at the stage we declare the ordinal $o^\Gamma(x)$ for our ω -computable approximation, all other followers must *protect* this computation. When we first declare this ordinal we say that x has been *established*. The idea will be to define our ordinal approximation as the sum of $o^k(p)$ for all followers such that $o^k(p) < \omega$. Then as long as we can successfully protect the computation, $\Gamma(D, x)$, from all other followers this will successfully bound the number of injuries to $\Gamma(D, x)$.

Now to ensure $\Psi_W(D, W, p)$ is total we need to define this computation immediately after we appoint p . It is possible that $o^k(p) = \omega$ at this stage.

Therefore there could be some x established for an N_Γ requirement after the use of the computation $\Psi_W(D, W, p)$ is defined but before the stage we see $o^k(p) < \omega$. Since x was established before the ordinal was revealed we cannot allow p to injure $\Gamma(D, x)$. But since the computation $\Psi_W(D, W, p)$ was defined before x was established there is no guarantee that $\psi_t(p) > \gamma_t(x)$. Note that we cannot wait until we see $o^k(p) < \omega$ before declaring the ordinal for x because it is possible that this never happens. So we need to wait for an opportunity to lift $\psi(p)$ to be larger than the use of all computations it must protect. This means that before σ can start putting numbers into D to make progress on meeting its requirement we need to wait for a W change that permits us to lift the use. Now you may ask, what if this never happens? Well, there is no guarantee that W will permit a single (σ, p) , but using the fact that we only require $\Psi_W(D, W) \neq f^k$ if $W \not\leq_T D$, we know that if we appoint a sequence of followers p_0, p_1, \dots then one of them will eventually be permitted. Notice that the action of σ is finite, we only need to continue to appoint followers when no current follower has been permitted and only a single follower needs to enumerate numbers into D to achieve its goal of $\Psi_W(D, W, p)[s] \neq f_s^k(p)$. Then we are able to initialise all nodes weaker than σ each time we enumerate a number into D . Therefore only τ nodes such that $\tau \infty \preceq \sigma$ will sustain injury from (σ, p) . Notice that to ensure that the $\psi^\sigma(p) > \gamma^\tau(x)$ when the use is lifted large, we also require $\Gamma_\tau(D, x)[s] \downarrow$ for all (τ, x) protected by (σ, p) at the stage we lift the use large. But as long as $\Gamma_\tau(D)$ is actually total then this is guaranteed to eventually align with a W change. Now $\sigma \succeq \tau \infty$ but in some constructions it is possible that $\tau \infty$ is on the true path but the measured functional $\Gamma_\tau(D)$ is not total; this will be discussed later as this happens in the construction that will be detailed in Chapter 3. But in this construction we are able to make D totally ω -c.a.; hence D is low₂. Therefore in this construction if $\tau \infty$ lies on the true path, $\Gamma_\tau(D)$ is total.

Notice that this argument generalises for any $\alpha \leq \epsilon_0$.

2.2 Maximal totally ω^ω -c.a. degrees in upper cones

Now we will discuss the proof of Theorem 1.5 in the case for $\alpha = \omega$ and $\beta = \omega^\omega$. Let A be such that $\deg_T(A)$ is totally ω -c.a. Then we build a set D such that $\deg_T(A \oplus D)$ is maximal totally ω^ω -c.a. To ensure $\deg_T(A \oplus D)$ is totally ω^ω -c.a. we meet the following set of requirements for all functionals Γ :

- N_Γ : If $\Gamma(A, D)$ is total then $\Gamma(A, D)$ is ω^ω -c.a.

To ensure $\deg_T(A \oplus D)$ is maximal we ensure that every c.e. set above $A \oplus D$ is not ω^ω -c.a. So for all c.e. sets W we build a functional $\Psi_W(A, D, W)$ with the aim that either $W \leq_T A \oplus D$ or $\Psi_W(A, D, W)$ is not ω^ω -c.a. Let $\langle f^k \rangle$ be the sequence that lists all ω^ω -c.a. functions. So for all c.e. sets W and all $k < \omega$ we meet the following set of requirements:

- $P_{W,k}$: If $\langle f^k, o^k \rangle$ is eventually ω^ω -computable then either $W \leq_T A \oplus D$ or $\Psi_W(A, D, W) \neq f^k$.

Notice that the requirements are very similar to the requirements in the previous discussion. The basic module for meeting these requirements is the same as the previous discussion, in particular if A was empty then the exact strategy for building a maximal degree works. So now we ask, what happens when a number goes into A ?

We begin by following the strategy as in the previous section. Once $o_s^k(p) < \omega^\omega$ we wait for a W change to permit use to lift $\psi(p)$ large, above the use of $\Gamma_\tau(A, D, x)$ computations that (σ, p) must protect. But then after (σ, p) is permitted a number could enter A . Now while we get to redefine $\Psi_\sigma(A, D, W, p)$, lifting the use large again would not help us because we must define the new $\Psi_\sigma(A, D, W, p)$ computation before the protected $\Gamma_\tau(A, D, x)$ computations recover. So we must cancel follower p and appoint a new follower. If infinitely many followers are cancelled then $W \leq_T A \oplus D$; hence it is ok that diagonalisation will fail in this case.

So we are able to ensure that (σ, p) does not injure any $\Gamma_\tau(A, D, x)$ computations that it must protect. Now consider how we meet requirements N_Γ . Since protection works we can take the sum of the ordinals $o^k(p)$ such that $o^k(p) < \omega^\omega$ and this bounds the number of D changes that can injure a $\Gamma_\tau(A, D, y)$ computation. But what happens when a number enters A ? We need a way to count these A changes. So we use the fact that A is totally ω -c.a. To use this, we build a *shadow functional* $\hat{\Gamma}(A)$ and appoint a *tracker* z for input x . We define the use of the computation $\hat{\Gamma}(A, z)$ to have use $\gamma(x)$. Then if there is an A change below the use of the computation $\Gamma_\tau(A, D, x)$, this is also below the use of the computation $\hat{\Gamma}(A, z)$. Since A is totally ω -c.a., $\hat{\Gamma}(A)$ is ω -c.a. Then the ordinal given for input z in an ω -computable approximation of $\hat{\Gamma}(A)$ is a bound of how many A changes occur below the computation $\Gamma_\tau(A, D, x)$. To obtain this ordinal bound τ nodes will need to guess whether $\hat{\Gamma}(A) = g^i$ or not, where $\langle g^i, l^i \rangle$ is the list of tidy $\omega + 1$ -computable approximations. The tracker is able to ‘track’ the A changes below the use of $\Gamma_\tau(A, D, x)$ while $\hat{\gamma}(z) \geq \gamma(x)$. But notice that when there is a D change below $\gamma(x)$ then the tracker becomes useless since we may no longer have $\hat{\gamma}(z) \geq \gamma(x)$. Hence when this happens we will need to appoint a new tracker z' .

Now that a new tracker has been appointed, this will have its own ordinal bound for the number of A changes below $\Gamma(A, D, x)$ and it is quite possible that this is larger than the ordinal for the old tracker. So we need to be able to decrease our ordinal by a multiple of ω so we are able to incorporate the ordinal bound of the new tracker. Now we can see that the strategy is to use the fact that $\omega \cdot \omega^\omega = \omega^\omega$ so we are able to drop down by a multiple of ω for each D change and can incorporate the ordinal of the new tracker this way. This will work because we need to replace the tracker after each D change.

2.3 Maximal totally ω^2 -c.a. degrees in upper cones

Now we consider whether we can make the bound for β in Theorem 1.5 any better. In particular we ask whether there is a maximal totally ω^2 -c.a. degree above every totally ω -c.a. degree. We consider the discussion in the previous section and ask what happens to this argument if we try to make $\text{deg}_T(A \oplus D)$ totally ω^2 -c.a. instead of ω^ω -c.a.

The key to being able to make $\text{deg}_T(A \oplus W)$ totally ω^ω -c.a. was that $\omega \cdot \omega^\omega = \omega^\omega$. Now $\omega \cdot \omega^2 \neq \omega^2$ so it is clear that we are not able to use the same strategy as we are unable to change the tracker for every D change that occurs. For every (σ, p) we have an ordinal of the form $\omega \cdot m(p) + k(p)$. We define an ordinal for $\Gamma(A, D, x)$ of the form $\omega \cdot d(x) + b(x)$. Now we are able to fully incorporate the ordinal for (σ, p) as in the last discussion, allowing us to deal with D changes. We are able to appoint a tracker, z , for x but when there is a D change we are not able to incorporate the ordinal of a new tracker until we see $d(x)$ decrease by one and we can only do this once we see $m(p)$ decrease by one. Hence between the stage the old tracker became useless and the stage the new tracker becomes useful, many A changes could occur, and we had no way to know in advance how many such A changes could occur. Then it is quite possible that these A changes ‘use up’ some of the ordinal that we included to count D changes; clearly this will not work. We could ask whether we could enumerate numbers into D until we see $m(p)$ decrease by one. Since $m(p)$ has been incorporated into $d(x)$, once this happens we are able to incorporate the ordinal bound of the new tracker. But for this to work we must wait until this has happened before we certify the computation $\Gamma(A, D, x)$. But we need to see $\Psi_W(A, D, W, p) = f^k(p)$ each time we enumerate a number into D . We are not guaranteed that this will happen enough times to decrease $m(p)$ by one before we certify the computation $\Gamma(A, D, x)$. So it is evident that this strategy cannot work.

This leads us to consider that not every totally ω -c.a. degree has a maxi-

mal totally ω^2 -c.a. degree above it. So we instead attempt to prove that there is a c.e. set $\text{deg}_T(A)$ such that it is totally ω -c.a. and there is no maximal totally ω^2 -c.a. degree above. To do this we build A such that $\text{deg}_T(A)$ is totally ω -c.a. and for every c.e. set W , if $\text{deg}_T(A \oplus W)$ is totally ω^2 -c.a. we build a set Q_W not below $A \oplus W$ such that $\text{deg}_T(A \oplus W \oplus Q_W)$ is totally ω^2 -c.a., this ensures $\text{deg}_T(A \oplus W)$ is not maximal totally ω^2 -c.a.

Now our strategy is to exploit the problem discussed above. For every functional Ψ , we need to ensure $\Psi(A, W, Q_W)$ is ω^2 -c.a. if $\text{deg}_T(A \oplus W)$ is totally ω^2 -c.a. So for each $x < \omega$ we need to define an ordinal of the form $\omega \cdot m(x) + k(x)$ that will bound the number of injuries $\Psi(A, W, Q_W)$ may sustain. The strategy is to build a shadow functional $\Gamma(A, W)$ and if $\Gamma(A, W)$ is ω^2 -c.a. then we need to work towards making $\Psi(A, W, Q_W)$ ω^2 -c.a. Then we can use the fact that $\Gamma(A, W)$ is ω^2 -c.a. and appoint *trackers* to track the A and W changes below the use of a $\Psi(A, W, Q_W)$ computation. Notice that while Q_W is empty then making an ω^2 -computable approximation for $\Psi(A, W, Q_W)$ is easy because we can just copy the ordinal of the tracker. But if a number goes into Q_W then we need a new tracker. Now we enumerate numbers into Q_W to achieve diagonalisation $\Theta(A, W) \neq Q_W$ for all functionals Δ . Hence we will be able to bound how many enumerations into Q_W could injure a $\Psi(A, W, Q_W, x)$ computation. But now the ordinal of the tracker is ω^2 , of the form $\omega \cdot d(z) + b(z)$, and this could be larger than the ordinal of the previous tracker. To deal with this we decrease $m(x)$ by one to allow us to define a new large $k(x)$; then we are able to define $k(x) = d(z)$. Now after we do this suppose there is a W change injuring the computation $\Psi(A, W, Q_W, x)$, then we need to decrease our ordinal, but this means we need to see $d(z)$ decrease by one, it is not enough that $b(z)$ decreases by one. So the strategy is to ‘beat’ the ordinal $\omega \cdot d(z) + b(z)$ down until we see $d(z)$ decrease by one. We do this by starting an *attack* by enumerating $\gamma(z)$ into A . Each time we do this we force the ordinal of the tracker to decrease; we do this until we see $d(z)$ decrease by one, as desired. We are able to do this because we can wait until this process is finished before certifying the computation

$\Psi(A, W, Q_W, x)$. This is because if $\Gamma(A, W)$ is not ω^2 -c.a. then we no longer need to make $\Psi(A, W, Q_W)$ ω^2 -c.a.

The strategy to build Q_W such that Q_W is not below $A \oplus W$ is to ensure that $Q_W \neq \Theta(A, W)$ for every functional Θ . We appoint a follower q and if $\Theta(A, W, q)[s] = 0$ then we enumerate q into Q_W . Now suppose we enumerate q into Q_W at stage s . It is possible some number enters A or W below $\theta_s(q)$ at a stage $t > s$, then this injures the computation $\Theta(A, W, q)$. Now at a stage $r > t$ this computation will recover, but it is possible that $\Theta(A, W, q)[r] = 1$; hence our diagonalisation attempt has failed. Then we need to cancel the current follower and appoint a new follower q_1 . We repeat the above strategy and enumerate q_1 into Q_W at a stage where we see $\Theta(A, W, q_1) = 0$. But now it is possible that we need to cancel q_1 and appoint a new follower q_2 . To ensure we have successfully diagonalised we need to know that this does not happen infinitely often. So we build a shadow functional $\Delta_\Theta(A, W)$ and appoint an *anchor*, p . This anchor serves many followers and we ensure $\delta(p) \geq \theta(q)$ where q is the current follower. Then if we need to appoint a new follower then this is because the computation $\Theta(A, W, q)$ was injured; hence $\Delta_\Theta(A, W, p)$ was also injured. Now recall that we only need to successfully diagonalise if $\deg_T(A \oplus W)$ is totally ω^2 -c.a.; thus we can assume $\Delta_\Theta(A, W, p)$ is ω^2 -c.a. Then this anchor ensures only finitely many followers are appointed.

This is the basic strategy for an attempt to prove that there is a totally ω -c.a. degree such that no degree above it is maximal totally ω^2 -c.a. This leads to a rather complicated construction so first we consider a construction based on this but for a single c.e. set W . Such a construction proves Theorem 1.6.

For a given W we build a set A such that $\deg_T(A)$ is totally ω -c.a. and $\deg_T(A \oplus W)$ is not maximal totally ω^2 -c.a. To ensure $\deg_T(A)$ is totally ω -c.a. we meet the following set of requirements for all functionals Φ :

- N_Φ : If $\Phi(A)$ is total then $\Phi(A)$ is ω -c.a.

If $\deg_T(A \oplus W)$ is not ω^2 -c.a. then we are done; otherwise we show $\deg_T(A \oplus$

W) is not maximal totally ω^2 -c.a. by building Q such that $\deg_T(A \oplus W \oplus Q)$ is totally ω^2 -c.a. and $Q \not\leq_T A \oplus W$. To ensure $Q \not\leq_T A \oplus W$ we build a shadow functional $\Delta_\Theta(A, W)$ for every functional Θ with the aim that either $\Delta_\Theta(A, W)$ is not ω^2 -c.a. or $\Theta(A, W) \neq Q$. So we need to meet the following set of requirements:

- R_Θ : If $\Theta(A, W)$ is total then either $\Theta(A, W) \neq Q$ or $\Delta_\Theta(A, W)$ is not ω^2 -c.a.

Now let $\langle f^i \rangle$ be the sequence that lists all ω^2 -c.a. functions. Then we have the following set of subrequirements:

- $R_{\Theta,k}$: $\Delta_\Theta(A, W) \neq f^k$ where f^k is the k th ω^2 c.a. function, along with its approximation $\langle f_s^k, o_s^k \rangle$.

Nodes working for subrequirement $R_{\Theta,k}$ is called a child of the node working for requirement R_Θ . Children will be spread down the tree below its parent. If all subrequirements are met then $\Delta_\Theta(A, W)$ is not ω^2 -c.a.; hence we have met requirement R_Θ . Otherwise, a particular subrequirement $R_{\Theta,k}$ will show that $\Delta_\Theta(A, W) = f^k$ and also give us a tidy $(\omega^2 + 1)$ -computable approximation $\langle f_s^k, o_s^k \rangle$. This node will then work towards meeting the requirement that $\Theta(A, W) \neq Q$.

Now we also need to ensure that $\deg_T(A \oplus W \oplus Q)$ is totally ω^2 -c.a., we do this by building a shadow functional $\Gamma_\Psi(A, W)$ for every functional Ψ with the aim that either $\Gamma_\Psi(A, W)$ is not ω^2 -c.a. or $\Psi(A, W, Q)$ is ω^2 -c.a. So we meet the following set of requirements:

- P_Ψ : If $\Psi(A, W, Q)$ is total then either $\Gamma_\Psi(A, W)$ is not ω^2 -c.a. or $\Psi(A, W, Q)$ is ω^2 -c.a.

Similar to the R requirements, we have the following subrequirements:

- $P_{\Psi,k}$: $\Gamma_\Psi(A, W) \neq f^k$ where f^k is the k th ω^2 c.a. function, along with its approximation $\langle f_s^k, o_s^k \rangle$.

Nodes working for subrequirement $P_{\Psi,k}$ is called a child of the node working for requirement P_Ψ . Children are spread down the tree below its parent. If all subrequirements are met then $\Gamma_\Psi(A, W)$ is not ω^2 -c.a.; hence we have met

requirement P_Ψ . Otherwise, a particular subrequirement $P_{\Psi,k}$ will show that $\Gamma_\Psi(A, W) = f^k$ and also give us a tidy $(\omega^2 + 1)$ -computable approximation $\langle f^k, o^k \rangle$. This node will then work towards meeting the requirement that $\Psi(A, W, Q)$ is ω^2 -c.a.

The basic module is as discussed above, if there is a Q change that injures a $\Psi(A, W, Q, x)$ computation then we appoint a new tracker for x . Now if there is a W change we need to start an *attack* to ‘beat’ the ordinal of the tracker down a multiple of ω . Since we only need to ensure $\Psi(A, W, Q)$ is ω^2 -c.a. if $A \oplus W$ is totally ω^2 -c.a., we are able to wait for this attack to finish before issuing the next ordinal in our ω^2 -computable approximation for $\Psi(A, W, Q)$. Now we also need to show that A is totally ω -c.a. We do this in a similar way as how we create the ω^ω -computable approximation in the previous discussion. Since the start of an attack is triggered by a W change we lift the use large at this stage and continue to define it large throughout the length of the attack and this will ensure a $\Phi(A, y)$ computation can be protected. Now the ordinal for the tracker is of the form $\omega \cdot d(z) + b(z)$, so the number $b(z)$ gives a bound for how many numbers will enter A during an attack. Note that the bound for a subsequent attack is not revealed until the previous attack has finished. Now notice that because we are making $\Phi(A)$ ω -c.a. but $A \oplus W$ is totally ω^2 -c.a. we are only able to incorporate the $b(z)$ part of the ordinal for a tracker of x . Therefore we must update which $\Phi(A, y)$ computations need to be protected after each attack.

To meet the requirement that $Q \not\leq_T A \oplus W$ we appoint followers and diagonalise against $\Theta(A, W)$ for every functional Θ . To ensure we succeed we build a shadow functional $\Delta_\Theta(A, W)$ and appoint an anchor. Now we only need to succeed if $A \oplus W$ is ω^2 -c.a. so we can use the fact that $\Delta_\Theta(A, W)$ is ω^2 -c.a. to ensure finitely many followers are appointed. We must be careful of which $\Psi(A, W, Q)$ computations we allow a q enumeration to injure. So we will need to cancel followers not only after a failed diagonalisation but also when we are in danger of injuring a $\Psi(A, W, Q)$ computation that must be protected. The anchor allows us to do this without the risk of needing to

do this infinitely often; this will be discussed in more detail in the following chapter.

One feature of this construction is that there are nodes with potentially infinitary action. Although all attacks are finite, since they are bound by $b(z)$, action to ensure $\Psi(A, W, Q)$ is ω^2 -c.a. means we need to act for infinitely many x for each Ψ ; hence we may need to enumerate infinitely many numbers into A at a node working for this requirement. Therefore we are not able to initialise nodes extending a node working for such a requirement. Notice that an A change below a protected $\Phi(A, y)$ during an attack could cause us to start injuring a computation we should be protecting, but we are able to set up the construction so that this does not happen.

Another feature is that the tree of strategies is complicated. It is common to have every node of the same length working for the same requirement but this is not the case in this construction. We will have parents that measure totality and build a functional and then each parent has many children who measure length of agreement between their parents functional and a particular ω^2 -c.a. function. First notice that we do not need to place more children below the outcome that guess that the functional is indeed equal to f^k . Similarly since we have successfully met all P and R requirements if $\deg_T(A \oplus W)$ is not totally ω^2 -c.a. We don't need to place another parent node until we have seen confirmation that the current parents functional is in fact ω^2 -c.a. Since we are only considering a single c.e. set W , we are able to completely avoid having another parent node between a parent and its child. This is of great use because then every parent knows that the functionals above it on the tree are in fact total and they have enough information to keep their use above computations they must protect. This will be discussed further in the next chapter.

In section 2.1 we mentioned that in some constructions it is possible that the infinite outcome is on the true path but the measured functional is not total and the reason that this did not happen in the construction discussed in this construction is because we are able to succeed in making

D α -c.a. Now in this construction, the functionals $\Theta(A, W)$ and $\Psi(A, W, Q)$ are not necessarily ω^2 -c.a. Therefore even though the infinite outcome of the nodes that measure these functionals may be on the true path the functionals themselves may not be total; this is a $\mathbf{0}'''$ feature. An example of how this is problematic is because we need to cancel followers when a follower may injure a protected $\Psi(A, W, Q)$ but if $\Psi(A, W, Q)$ is not actually total then this may prevent this requirement from succeeding. Therefore we need to introduce nodes that will be spread down the tree below its parent that will measure $\liminf \text{dom } \Psi(A, W, Q)$. We will discuss this further in the next chapter.

Chapter 3

Theorem

Now we detail the construction to prove Theorem 1.6. We build a set A for a given W such that $\text{deg}_T(A)$ is totally ω -c.a. and $\text{deg}_T(A \oplus W)$ is not maximal totally ω^2 -c.a. To ensure A has the desired properties we meet the following set of requirements:

- N_Φ : If $\Phi(A)$ is total then $\Phi(A)$ is ω -c.a.
- P_Ψ : If $\Psi(A, W, Q)$ is total then either it is ω^2 -c.a. or $\Gamma_\Psi(A, W)$ is total and not ω^2 -c.a. Subrequirements:
 - $P_{\Psi,k}$: $\Gamma_\Psi(A, W) \neq f^k$ where f^k is the k th ω^2 c.a. function, along with its approximation $\langle f_s^k, o_s^k \rangle$
- R_Θ : $\Theta(A, W) \neq Q$ or $\Delta_\Theta(A, W)$ is not ω^2 -c.a. Subrequirements:
 - $R_{\Theta,k}$: $\Delta_\Theta(A, W) \neq f^k$ where f^k is the k th ω^2 c.a. function, along with its approximation $\langle f_s^k, o_s^k \rangle$

3.1 Glossary

Here we give a glossary of terms that will be used in this chapter as well as the following chapter. These terms will be introduced during the technical discussion.

- π is a node working for an N_Φ requirement.

- τ is a node working for a P_Ψ requirement. A *daughter* of τ works for requirement $P_{\Psi,k}$ for some $k < \omega$. A *son* of τ is a ζ_k node for some $k < \omega$.
- ρ is a node working for a $P_{\Psi,k}$ requirement. τ is the *parent* of ρ if τ works for requirement P_Ψ .
- η is a node working for an R_Θ requirement. A *daughter* of η works for requirement $R_{\Theta,k}$ for some $k < \omega$. A *son* of η is an ξ_k node for some $k < \omega$.
- μ is a node working for an $R_{\Theta,k}$ requirement. η is the *parent* of μ if η works for requirement R_Θ .
- $\text{tr}_s(\rho, x)$ is the tracker for (ρ, x) at stage s . We sometimes use the notation $\text{tr}_s(x)$ when it is clear which ρ this refers to.
- $\text{ac}_s(\mu)$ is the anchor for μ at stage s and $\text{fl}_s(\mu)$ is the follower for μ at stage s .
- $\text{pro}_s(\mu)$ is a set of (ρ, x) that μ *protects*.
- $\text{pro}_s(\hat{\rho}, \hat{x})$ is a set of (ρ, x) that $(\hat{\rho}, \hat{x})$ *protects*.
- $\langle f_s^\alpha, o_s^\alpha \rangle_{s < \omega}$ is the ordinal approximation given by an α node, where α is either a μ or ρ node. $o_s^\rho(z)$ has the form $o_s^\rho(z) = \omega \cdot d_s^\rho(z) + b_s^\rho(z)$.
- $\Gamma_\Psi(A, W) = \Gamma_\tau(A, W) = \Gamma_\rho(A, W)$ and $\Psi_\tau(A, W, Q) = \Psi_\rho(A, W, Q)$ if τ works for requirement P_Ψ and ρ is a daughter of τ . Similarly $\Delta_\Theta(A, W) = \Delta_\eta(A, W) = \Delta_\mu(A, W)$ and $\Theta_\eta(A, W) = \Theta_\mu(A, W)$ if η works for requirement R_Θ and μ is a daughter of η .
- $I_s(\rho, x)$ is a set of elements from ω such that x is the least element in the set and every other $x' \in I_s(\rho, x)$ has been declared taken over by x at some stage $t < s$.
- $\mathcal{C}_s(\rho)$ is the set of inputs x that have been established and not taken over for ρ . $\mathcal{C}_0(\rho) = \emptyset$ and then at each stage $\rho \infty$ is accessible, a new

input x is established. As inputs are taken over we remove them from $\mathcal{C}_s(\rho)$ so this set contains the least input from every interval for ρ .

- When (ρ, x) is *invented* means the first tracker for x has been appointed, we call this tracker the *original* tracker for x , denoted $\text{orig}(x)$.
- (ρ, x) is *established* once there has been a ρ -expansionary stage since the invention of x .
- We call (ρ, x) *corrupted* while the tracker of x is not the original tracker of x . This happens when a μ node enumerates a number into Q .
- (ρ, x) *fully corrupted* means there has been a ρ -expansionary stage after the corruption of x , this means the ordinal for the new tracker has been revealed.
- If (ρ, x) is *uncorrupted* then the new tracker, that was appointed after corruption, has been replaced by the original tracker. Then happens when there is a relatively small A or W change that allows us to use the original tracker after corruption.
- Suppose x is corrupted. Then the *corrupting* μ is the μ node that enumerated a number into Q at the stage x is declared corrupted. The *corrupting* q is the number that the corrupting μ enumerated into Q .
- We declare (ρ, x) to start an attack at the start of the stage. We say (ρ, x) is *fully in an attack* once there has been a τ -expansionary stage during the attack. Note that $\Gamma_\tau(A, W, \text{tr}(\rho, x)) \uparrow$ while (ρ, x) is in an attack but not yet fully in an attack.
- $\psi_s^\rho(I_s(\rho, x)) = \max\{\psi_s^\rho(x) : x \in I_s(\rho, x)\}$
- Define $\#t$ to be the largest number used by the construction at stage t .
- $\text{prec}(\alpha)$ is the set of τ such that there is $\zeta_k \uparrow \preceq \alpha$ where ζ_k is a son of τ .

- A node μ believes $\Psi_\tau(A, W, Q)$ is total if $\mu \succ \rho \infty$ where ρ is a daughter of τ .
- A node μ believes $\text{dom } \Psi_\tau(A, W, Q) = k$ if $\mu \succ \zeta_k \uparrow$ where ζ_k is a son of τ .
- When τ carries out a request or passes a request on to some $\hat{\tau}$ we say that it *acts on the request*.
- (ρ, x) *requires protection from μ at stage s* if $x \in \mathcal{C}_s(\rho)$, (ρ, x) is not currently in an attack and one of the following holds:
 - $\rho \infty \preceq \mu$
 - ρ is to the left of μ and μ believes $\Psi_\rho(A, W, Q)$ is total
 - ρ is to the left of μ and μ believes $\text{dom } \Psi_\rho(A, W, Q) = k$ and $I_s(\rho, x) \subseteq k$

3.2 Technical Discussion

First consider the requirement R_Θ . If $\Theta(A, W)$ is not total then we have met the requirement, otherwise we build a functional $\Delta_\Theta(A, W)$. If $\Theta(A, W)$ is total and $\Delta_\Theta(A, W)$ is ω^2 -c.a. then we need to enumerate a set Q such that $\Theta(A, W) \neq Q$. A node on the tree of strategies working for requirement R_Θ is denoted η . η guesses whether $\Theta(A, W)$ is total by looking at the length of $\text{dom } \Theta(A, W)$. η nodes have two outcomes, one that believes $\text{dom } \Theta(A, W)$ goes to infinity and the other believes $\text{dom } \Theta(A, W)$ is finite. Below the finite outcome we do not need to act for this requirement as if this is the correct outcome then we have met the requirement. At stages that we believe the infinite outcome we need to build an initial segment of the functional $\Delta_\Theta(A, W)$. Below the infinite outcome we need to guess if $\Delta_\Theta(A, W)$ is ω^2 -c.a. This gives subrequirements $R_{\Theta, k}$ which guess whether $\Delta_\Theta(A, W) = f^k$ or not, where f^k is the k th ω^2 c.a. function. The nodes working for subrequirements are called *children* of η and are denoted μ . μ nodes have two outcomes, one that believes $\Delta_\Theta(A, W) = f^k$ and the other believes

$\Delta_{\Theta}(A, W) \neq f^k$. Now notice that children of η are only on the tree below the infinite outcome of η . Also notice that below the outcome of child node that believes $\Delta_{\Theta}(A, W) = f^k$ we do not need to place any more children of η . Now at stages where we believe that $\Delta_{\Theta}(A, W)$ is ω^2 -c.a., which is indicated by the infinite outcome of a μ node, we need to ensure that $\Theta(A, W) \neq Q$. To diagonalise we appoint a follower q and then at stages s where we see $\Theta(A, W, q)[s] = 0$, we enumerate q into Q . Now suppose we enumerated q into Q at stage s because $\Theta(A, W, q)[s] = 0$. Notice that if there is an A or W change below the use $\theta_s(q)$ after stage s , the computation $\Theta(A, W, q)[s]$ is injured, so it is possible that $\Theta(A, W, q) = 1$; hence our diagonalisation has failed. If this happens we appoint a new follower and make another attempt to diagonalise. Now to guarantee that diagonalisation is successful we need to ensure that only finitely many followers are appointed. To do this we appoint an anchor p which will serve many followers and use the fact that $\Delta_{\Theta}(A, W)$ is ω^2 -c.a. So when we define the computation $\Delta_{\Theta}(A, W, p)$ we define it with use at least $\theta(q)$, where q is the current follower. Now when we enumerate q into Q because we have seen $\Theta(A, W, q)[s] = 0$, if here is an A or W change below the use of this computation then it is also below the use of the computation $\Delta_{\Theta}(A, W, p)$. Now since $\Delta_{\Theta}(A, W)$ is ω^2 -c.a., the ordinal of the ω^2 -computable approximation must decrease. This can only happen finitely many times; hence only finitely many followers are appointed and the last follower must be successful.

Now consider the requirement P_{Ψ} . Similar to the R requirements, if $\Psi(A, W, Q)$ is not total then we have met the requirement, otherwise we build a functional $\Gamma_{\Psi}(A, W)$. Then if $\Psi(A, W, Q)$ is total and $\Gamma_{\Psi}(A, W)$ is ω^2 -c.a. then we need to show that $\Psi(A, W, Q)$ is also ω^2 -c.a. A node on the tree of strategies working for requirement P_{Ψ} is denoted τ . A τ node guesses whether $\Psi(A, W, Q)$ is total or not by looking at the length of $\text{dom } \Psi(A, W, Q)$. These τ nodes work in the same way as the η nodes. At stages that we believe the infinite outcome we need to build an initial segment of the functional $\Gamma_{\Psi}(A, W)$. Then below the infinite outcome we

need to guess if $\Gamma_\Psi(A, W)$ is ω^2 -c.a. So similar to the R requirements, we get subrequirements guessing whether $\Gamma_\Psi(A, W) = f^k$ or not. Nodes working for these subrequirements are called *children* of τ and are denoted ρ . ρ nodes have two outcomes similar to μ nodes. If $\Gamma_\Psi(A, W)$ is ω^2 -c.a. then there is a ρ node which guesses $\Gamma_\Psi(A, W) = f^k$. Then this ρ node gives us an ω^2 -computable approximation $\langle f_s^\rho, o_s^\rho \rangle$. So for each $z < \omega$ there is an ordinal $o_s^\rho(z)$ with the form $\omega \cdot d_s^\rho(z) + b_s^\rho(z)$. Now to show $\Psi(A, W, Q)$ is ω^2 -c.a. we need to define an ω^2 -computable approximation for $\Psi(A, W, Q)$. For each $x < \omega$ we need to define a non-increasing sequence of ordinals $o_s^\Psi(x)$ of the form $\omega \cdot m_s^\Psi(x) + k_s^\Psi(x)$ such that if $\Psi(A, W, Q, x)[s+1] \neq \Psi(A, W, Q, x)[s]$ then $o_{s+1}^\Psi(x) < o_s^\Psi(x)$. To do this we appoint a tracker for each $x < \omega$, $\text{tr}(x)$, once we have appointed a tracker for x we say x has been *invented*. When we first issue an ordinal to x we say x has been *established*. Fix x and let z be the tracker of x . First assume Q is empty. When we define the computation $\Gamma_\Psi(A, W, z)$ at stage s we define it with use at least $\psi_s(x)$. Now if $\Psi(A, W, Q, x)[s+1] \neq \Psi(A, W, Q, x)[s]$, then there has been some change below the use $\psi_s(x)$. Since $\gamma_s(z) \geq \psi_s(x)$ and Q is empty, it follows that $\Gamma_\Psi(A, W, z)[s+1] \neq \Gamma_\Psi(A, W, z)[s]$; hence $o_{s+1}^\rho(z) < o_s^\rho(z)$. Then we see that we can just follow the ordinal of the tracker. So if we define $m_s^\Psi(x) = d_s^\rho(z)$ and $k_s^\Psi(x) = b_s^\rho(z)$ then this gives us an ω^2 -computable approximation for $\Psi(A, W, Q)$.

So now we need to consider how R requirements enumerating numbers into Q affect the P requirements. Fix x and let z be the tracker of x . Suppose at stage s an R requirement enumerates its follower q into Q and $q < \psi_s(x)$. Now $\Psi(A, W, Q, x)[s+1] \neq \Psi(A, W, Q, x)[s]$ but it could be that $\Gamma_\Psi(A, W, z)[s+1] = \Gamma_\Psi(A, W, z)[s]$ and $\psi_{s+1}(x) > \gamma_{s+1}(z)$. Then the current tracker is useless, so we need to appoint a new tracker z' and we declare that x is *corrupted*. Now this new tracker has an ordinal $o_s^\rho(z') = \omega \cdot d_s^\rho(z') + b_s^\rho(z')$. But it is quite possible the $d_s^\rho(z') > d_s^\rho(z)$, so we cannot just follow the ordinal of the new tracker like our previous strategy. Instead, when x gets corrupted we decrease $m^\Psi(x)$ by one and this allows us to define a new large value for

$k^\Psi(x)$, so we define $k^\Psi(x) = d^\rho(z')$. Now consider what happens if there is a W change below $\psi(x)$. Now $o^\rho(z')$ must have decreased since this change was also below $\gamma(z')$ but it could be that $d_s^\rho(z')$ has not changed. But now $k^\Psi(x) = d^\rho(z')$ and we need to see a decrease in $k^\Psi(x)$. So we begin an *attack* where we lift the use $\gamma(z')$ to be large and enumerate this into A . Each time we do this we will see the ordinal $o^\rho(z')$ decrease, so eventually we will see $d^\rho(z')$ decrease and this gives us the decrease in $k^\Psi(x)$ that we needed.

Now suppose x has been corrupted and let $\mathbf{orig}(x)$ be the original tracker for x . If there is a stage t such that $\Gamma_\Psi(A, W, \mathbf{orig}(x))[t] \uparrow$ then we are able to once again define the use of this computation to be at least the use $\psi(x)$; hence we can use $\mathbf{orig}(x)$ as the tracker for x once again. If this happens then we call x *uncorrupted*. Once x is uncorrupted then we decrease $m^\Psi(x)$ by one and define $k^\Psi(x)$ to be $b^\rho(\mathbf{orig}(x))$ and go back to following the ordinal of the original tracker as discussed while Q was empty. We can continue to do this until there is another stage where a number enters Q below $\psi(x)$.

To meet the N requirements we first need to guess whether $\Phi(A)$ is total or not, we do this in the same way as the P and R requirements. If $\Phi(A)$ is not total then we are done with the requirement, otherwise we need to show that $\Phi(A)$ is ω -c.a. First consider the interaction of only one P requirement with an N requirement. Now when y is established, at stage s , we can see the ordinal, $o_s(\mathbf{tr}(x)) = \omega \cdot d_s(\mathbf{tr}(x)) + b_s(\mathbf{tr}(x))$, for established inputs x . Now if x has already been corrupted then if there is a W change an attack could start and this attack puts at most $b_s(\mathbf{tr}(x))$ numbers into A . Since this number has been seen by y when it was established we can allow injury from this attack. Notice that if we did let an attack beat the ordinal all the way down y has no idea how many injuries this could cause because at stage s all it can incorporate into its ordinal is $b_s(\mathbf{tr}(x))$; hence we can see that is important that we stop attacking once $d(z)$ has decreased by one. Once this attack finishes we cannot let any future (ρ, x) attack injure $\Phi(A, y)$ because we had not seen the bound for any future (ρ, x) attacks when y was established.

Consider the interaction between two P requirements. We only start an attack when there is a W change; hence we need to be able to control how one P requirement could cause an A enumeration that could injure the computation of another P requirement. Now recall that since we are done if $A \oplus W$ is not totally ω^2 -c.a. which allows us to be able avoid having another parent node on the tree between a parent and its child. Hence if $\hat{\rho} \succeq \rho \infty$ then $\hat{\tau} \succeq \rho \infty$. This is very useful because this way $\hat{\tau}$ can see that $\Gamma(A, W)$ is totally ω^2 -c.a. So during an attack it can define a set of inputs that it must *protect*, denote this set by $\text{pro}(\hat{\rho}, \hat{x})$. Now if there is an A or W change below $\psi(x)$ then we get to lift $\gamma^{\hat{\rho}}(\hat{z})$ to be large at the next $\hat{\tau}$ -expansionary stage. Since $\hat{\tau} \succeq \tau \infty$ then we have seen $\Psi(A, W, Q, x)$ recover; hence we can ensure $\gamma^{\hat{\rho}}(\hat{z}) > \psi(x)$. We can lift $\gamma^{\hat{\rho}}(\hat{z})$ to be large when this happens because $\hat{\tau}$ can see that $\Gamma(A, W)$ is totally ω^2 -c.a.; hence we know this happens finitely often and so this cannot send the use to infinity.

A τ node guesses whether $\Psi(A, W, Q)$ is total by measuring the length of the domain. But notice that it could be that $\text{dom } \Psi(A, W, Q)$ goes to infinity but there could be an input with unbounded use. If this happens then $\Psi(A, W, Q)$ is not total. In this situation it is clear that $\Gamma(A, W)$ will not be ω^2 -c.a. Above we discussed that along the path containing the finite outcome of all the children of a P or R requirement shows that $A \oplus W$ is not totally ω^2 -c.a. but this is not entirely true since it could be that $\Psi(A, W, Q)$ is actually not total due to the reason above. Therefore we see we also need nodes that check whether there is an input with unbounded use; this is a $\mathbf{0}'''$ feature of the construction. We cannot measure this using a single node (unless we allow nodes with $\omega+2$ outcomes [10]), so we spread nodes, denoted ζ_k , down the tree each measuring whether the computation $\Psi(A, W, Q, k)$ converges or not for a fixed k . We call these nodes *sons* and the children discussed previously are referred to as *daughters*. We place son nodes down with k increasing as we go down the tree; hence we are also able to determine the least point where $\Psi(A, W, Q)$ diverges. Notice that R requirements will also need to have sons nodes on the tree. Similarly they will measure whether

the computation $\Theta(A, W, k)$ converges or not for a fixed k .

Now back to considering the interaction between the R and P requirements. We need to decrease $m(x)$ by one each time a number enters Q below $\psi(x)$. Then we need to know how many times this can happen when we established x . Let η be a node working for an R requirement and let τ be a node working for a P requirement. Let μ and ρ be children of η and τ respectively. When x is established it can look to see which $\mu \succ \rho \infty$ currently have followers appointed and then this could tell us how many times we need to decrease $m(x)$ by one. For this to work we need to make sure that these are the only enumerations into Q that can injure $\Psi(A, W, Q, x)$. First consider the case where $\eta \succeq \rho \infty$. Suppose μ is not allowed to injure $\Psi(A, W, Q, x)$, then μ needs to *protect* (ρ, x) . Now $\eta \succeq \rho \infty$ so η knows $\Gamma(A, W)$ is ω^2 -c.a. Then we are able to define $\delta(p) \geq \gamma(z)$ because we know that this cannot drive the use to infinity. Now whenever $\Delta(A, W, p) \uparrow$ we are able to appoint a new large follower. So now whenever there is a change below $\gamma(z)$, we are able to appoint a new large follower; hence we can ensure $q > \psi(x)$. Now consider μ to the right of $\rho \infty$. Now due to the way we arrange the tree, η must extend a child of τ . If η extends a daughter of τ , then η believes $\Gamma(A, W)$ is total so we are able to protect (ρ, x) . But if η extends a son of τ , then η believes that $\text{dom } \Psi(A, W, Q) = k$. In this case we are able to protect (ρ, x) if $x < k$. If $x \geq k$ then we are not able to protect because we cannot define $\delta(p) \geq \gamma(z)$ as this would drive $\delta(p)$ to infinity.

So consider the case where η believes that $\text{dom } \Psi(A, W, Q) = k$ and $x \geq k$. If η is correct then the computation $\Psi(A, W, Q, x)$ changes infinitely often. We want to set things up so that the enumeration by μ does not corrupt $\Psi(A, W, Q, x)$. Now notice that μ is only accessible at a stage where there has been a change in this computation and μ is initialised at every ρ -expansionary stage (this is because μ is to the right of ρ). Then an enumeration by μ is *invisible* to ρ because this will only happen along with some other change since the last ρ -expansionary stage. So when it comes to the ordinal counting the Q change is invisible. But notice that $\Gamma(A, W, z) \downarrow$

at the stage μ enumerates q into Q . Then although the Q change did not cause a problem with the counting, the Q change still renders the current tracker useless; hence this enumeration still corrupts x . Now since the computation $\Psi(A, W, Q, x)$ changes infinitely often there are also infinitely many stages such that $\Gamma(A, W, z)[s] \uparrow$. So the strategy is to wait for a stage where $\Gamma(A, W, z)[s] \uparrow$ and then at this stage we can enumerate q into Q . This ensures that we do not need to replace the tracker because we are able to define $\Gamma(A, W, z)$ after the Q change. To do this μ sends a *request token* to τ at the stage μ sees $\Theta(A, W, q)[s] = 0$. Then at the next τ -expansionary stage such that there has been a change in the computation $\Psi(A, W, Q, x)$ since the last τ -expansionary stage then τ enumerates q into Q for μ and we leave $\Gamma(A, W, z)[s] \uparrow$. Notice that if μ is correct then we must eventually see a change in the computation $\Psi(A, W, Q, x)$, so we will eventually successfully enumerate q into Q . So when a new follower for μ is appointed after an enumeration into Q or after initialisation, we define a set $\mathbf{pro}(\mu)$ containing all (ρ, x) such that x has already been established and either μ extends some daughter of τ or μ extends a son of τ and $x < k$. Notice that when we have multiple P requirements τ may need to pass the request on to another $\hat{\tau}$ node if τ believes $\text{dom } \Psi_{\hat{\tau}}(A, W, Q) = \hat{k}$.

When we first appoint the anchor for μ, p , we define the set of protected (ρ, x) , $\mathbf{pro}_s(\mu)$. We redefine the set of protected (ρ, x) at any stage we appoint a new follower due to a failed diagonalisation attempt (this is a stage where we see an A or W change below $\delta(p)$ and the current follower is in Q). At stages where we redefine $\mathbf{pro}_s(\mu)$ we include all established (ρ, x) such that $\rho \infty \preceq \mu$. Now we also halt and initialise weaker nodes when we do this. So notice that every time a protected set for any μ is defined, there is a different least element for every (ρ, x) such that $\rho \infty \preceq \mu$. Now when μ enumerates a number into Q at stage s , for each ρ such that $\rho \infty \preceq \mu$, we can declare the least x such that $(\rho, x) \notin \mathbf{pro}_s(\mu)$ to be corrupted and x can take over all established $x' > x$. Then because this least x is unique to this enumeration, every x is only corrupted once. We choose to declare the least unprotected x

to be corrupted even when $q > \psi(x)$, to ensure x is only declared corrupted once. Also notice that if x is the least corrupted by an enumeration by μ at stage s then x was invented at the stage $\text{pro}_s(\mu)$ was defined.

Consider two P subrequirements such that (ρ, x) is currently in an attack, $\hat{\rho} \succeq \rho \infty$, \hat{x} is corrupted and $(\hat{\rho}, \hat{x}) \notin \text{pro}_s(\rho, x)$. Suppose (ρ, x) enumerates a number into A at stage s . $(\hat{\rho}, \hat{x}) \notin \text{pro}_s(\rho, x)$ so \hat{x} was established after $\text{pro}_s(\rho, x)$ was defined. Note that nodes to the right of ρ are initialised at every stage $\gamma(z)$ is lifted large. Now \hat{x} is corrupted at stage s so the μ that corrupts \hat{x} must be accessible during the (ρ, x) attack; hence μ is to the right of ρ . This means that μ is initialised at every stage $\gamma(z)$ is lifted large. Since $\text{pro}_t(\mu)$ was defined at the stage \hat{x} was invented, it follows that $\gamma(z)$ was last lifted large before \hat{x} was established. Notice that this means the number that (ρ, x) enumerates into A at stage s is relatively small to $(\hat{\rho}, \hat{x})$. So if we define the use of $\gamma^{\hat{\rho}}(\text{tr}(\hat{x}))$ to be large at the first $\hat{\tau}$ -expansionary stage after \hat{x} was invented, then the enumeration at stage s uncorrupts \hat{x} . Now notice that by doing this if there is ever an A change below $\psi(x)$ while corrupted, this A change must be small enough to uncorrupt x .

z acts as a tracker for x when $\gamma_s(z) \geq \psi_s(x)$. So notice that z could act as a tracker for many x . We just need to ensure that $\Gamma_\Psi(A, W)$ is total, so we need to ensure z is the tracker for finitely many x . So now we consider z as a tracker for an *interval* rather than for a single input. Corruption and attacks are actions that can only happen finitely often for each x so we choose these stages to allow use to increase the size of the interval z acts as a tracker for. Each interval has a least element, we call all other inputs in the interval *taken over* by the least element.

3.3 Tree of Strategies

Consider a parent node (τ or η), below the finite outcome the parent does not have any children and below the infinite outcome we will start assigning nodes to be children of this parent node, alternating between daughters and

sons. We call a parent node *closed* below the finite outcome of the parent as well as below the infinite outcome of a daughter and the divergent outcome of a son. Once the parent is *closed*, we stop assigning nodes to be children of this parent and we move to the next parent, alternating between τ and η . We need to ensure there is a node working for every N_Φ on every path, so to do this we assign every node of even length to work for an N requirement.

3.3.1 Assigning Requirements

List all functionals in order type ω . Then we assign requirements by induction. Let λ be the root of the tree. Assign λ to working for requirement P_Ψ where Ψ is the first functional on the list. Now let β be the longest node such that $\beta \prec \alpha$ and β is not a node working for N_Φ . Then we assign a requirement to α as follows:

Let α be a node of length l , then if l is odd then let γ be the longest node such that $\gamma \prec \alpha$ and γ is a node working for $N_{\hat{\Phi}}$. Now assign α to work for requirement N_Φ , where Φ is the next functional on the list after $\hat{\Phi}$. Otherwise find which of the following cases apply:

Case 1. β is a node working for P_Ψ .

- If $\beta \infty \preceq \alpha$, then let α be a node working for requirement $P_{\Psi,0}$.
- If $\beta \text{ fin} \preceq \alpha$, then let α be a node working for requirement R_Θ where $\Theta = \Psi$.

Case 2. β is a node working for $P_{\Psi,k}$.

- If $\beta \infty \preceq \alpha$, then let α be a node working for requirement R_Θ where $\Theta = \Psi$.
- If $\beta \text{ fin} \preceq \alpha$, then let α be a ζ_k node, son of P_Ψ .

Case 3. β is a ζ_k node, son of P_Ψ .

- If $\beta \uparrow \preceq \alpha$, then let α be a node working for requirement R_Θ where $\Theta = \Psi$.
- If $\beta \downarrow \preceq \alpha$, then let α be a node working for $P_{\Psi,k+1}$.

Case 4. β is a node working for R_Θ .

- If $\beta \infty \preceq \alpha$, then let α be a node working for requirement $R_{\Psi,0}$.
- If $\beta \text{ fin} \preceq \alpha$, then let α be a node working for requirement P_Ψ where Ψ is the next functional on the list after Θ .

Case 5. β is a node working for $R_{\Theta,k}$.

- If $\beta \infty \preceq \alpha$, then let α be a node working for requirement P_Ψ where Ψ is the next functional on the list after Θ .
- If $\beta \text{ fin} \preceq \alpha$, then let α be a ξ_k node, son of R_Θ .

Case 6. β is an ξ_k node, son of R_Θ .

- If $\beta \uparrow \preceq \alpha$, then let α be a node working for requirement P_Ψ where Ψ is the next functional on the list after Θ .
- If $\beta \downarrow \preceq \alpha$, then let α be a node working for $R_{\Theta,k+1}$.

3.3.2 Types of Nodes

There are seven types of nodes. These have been discussed in the technical discussion; here we give a summary of the action each node takes.

Nodes working for requirement N_Φ are denoted π and have outcomes $\infty < \text{fin}$. π nodes measure the length of the domain of $\Phi(A)$. At $\pi \infty$ stages we *establish* a new input y . When y is established we define the first ordinal for the ω -computable approximation for $\Phi_\pi(A, y)$.

Nodes working for requirement R_Θ are denoted η . η nodes have outcomes $\infty < \text{fin}$. η nodes measure the length of the domain of $\Theta(A, W)$ and at stages where η believes $\Theta(A, W)$ is total it will define an initial segment of $\Delta_\eta(A, W)$. η also checks whether any of its children need their current follower replaced by a new one. Note that we need to do this at η because we need to define $\delta(p) \geq \theta(q)$, where p is the anchor and q is the follower.

Nodes working for requirement $R_{\Theta,k}$, denoted μ , are called *daughter* nodes, placed on the tree below its parent η . μ nodes have outcomes $\infty < \text{fin}$. μ nodes have an anchor $p = \text{ac}_s(\mu)$ and a follower $q = \text{fl}_s(\mu)$ and enumerate

numbers into Q to diagonalise. If μ extends the divergent outcome of a son of τ , μ will send a request for the enumeration of a follower into Q instead of doing the enumeration itself. When we appoint the first follower for μ we define the set of (ρ, x) that μ must *protect*, $\text{pro}(\mu)$. Then at stages we appoint a new follower after a previous enumeration we redefine this set. When we enumerate a number into Q we declare (ρ, x) that are not protected by μ to be *corrupted*.

ξ_k nodes are called *sons* of η . A ξ_k node measures whether $\Theta_\eta(A, W, k)$ converges or diverges. ξ_k nodes have outcomes $\uparrow < \downarrow$.

Nodes working for requirement P_Ψ are denoted τ and have outcomes $\infty < \text{fin}$. A τ node measures the length of the domain of $\Psi(A, W, Q)$. At stages where τ believes $\Psi(A, W, Q)$ is total it will define an initial segment of $\Gamma_\tau(A, W)$. τ also has the job of dealing with requests from μ nodes below the divergent outcome of a son of τ .

Nodes working for requirement $P_{\Psi, k}$, denoted ρ , are called *daughter* nodes, placed on the tree below the infinite outcome of its parent τ . ρ nodes have outcomes $\infty < \text{fin}$. At ρ -expansionary stages a new input, x , is *invented*, this means we give x a tracker $\text{tr}_s(x)$, this is the first tracker for x which we also denote $\text{orig}(x)$. At the next ρ -expansionary stage x is *established*, at this stage we define the first ordinal for the ω^2 -computable approximation for $\Psi(A, W, Q, x)$, $o_t^\Psi(x)$. So $z = \text{tr}_s(x)$ means z is the tracker for x at stage s , but also the tracker for all inputs that have been taken over by x . When an input is taken over by x , it is added to the interval $I(x)$. So at stage s , z is the tracker for all inputs in $I_s(x)$. For action on the interval $I(x)$ we refer to action on the least input in that interval. $\mathcal{C}_s(\rho)$ denotes the collection of these inputs that have been established but not taken over. While x is corrupted (ρ, x) will start an attack if there is a W change below $\psi(I(x))$. While (ρ, x) is in an attack it will define the set of $(\hat{\rho}, \hat{x})$ that (ρ, x) must *protect*, $\text{pro}(\rho, x)$; this will be updated after each enumeration into A . Now notice we only update the ordinal for x , $o_s^\Psi(x)$, at ρ -expansionary stages. This means that when x is declared corrupted, we do not decrease the ordinal

until the next ρ -expansionary stage. So if there is a W change below $\psi_s(I(x))$ between the stage x is declared corrupted and the next ρ -expansionary stage we do not need to start an attack. At the first ρ -expansionary stage after x is declared corrupted we declare x to be *fully corrupted*. Note that we do not see the ordinal for the new tracker until x has been declared to be fully corrupted. Similarly, we do not see the new ordinal after finishing an attack until the next ρ -expansionary stage after the attack was declared finished. We declare a (ρ, x) attack to be finished at a τ node, so we will not see the new ordinal until the next ρ -expansionary stage; hence we do not want to start another attack between the stage the attack is declared finished and the next ρ -expansionary stage.

ζ_k nodes are called *sons* of τ . A ζ_k node measures whether $\Psi_\tau(A, W, Q, k)$ converges or diverges. ζ_k nodes have outcomes $\{\uparrow, \downarrow\}$ with $\uparrow < \downarrow$.

Note that we may assume that all computations we define have non-decreasing use.

3.4 Construction

At the beginning of each stage s we check if any (ρ, x) needs declaring uncorrupted or needs to start an attack. Let ρ be a node working for requirement $P_{\Psi, k}$ and let x be such that $x \in \mathcal{C}_s(\rho)$ and x is corrupted. If $\Gamma_\rho(A, W, \text{orig}(x))[s] \uparrow$, then define $\text{tr}_{s+1}(\rho, x) = \text{orig}(\rho, x)$ and declare x to be uncorrupted.

If (ρ, x) has not been uncorrupted then let $u = \psi_{s-1}^\rho(I_{s-1}(\rho, x))$. (ρ, x) wants to attack if all of the following hold:

- $W_{s-1} \upharpoonright u \neq W_s \upharpoonright u$
- (ρ, x) is not currently in an attack
- (ρ, x) is fully corrupted
- (ρ, x) was not in an attack at the last ρ -expansionary stage

If (ρ, x) wants to attack and there is no $(\hat{\rho}, \hat{x})$ to its left that wants to attack, then declare (ρ, x) to begin an attack.

If an attack has been started or an x has been declared uncorrupted then halt the stage and initialise all nodes to the right of $\rho \infty$. For each such (ρ, x) declare all established $x' > x$ taken over by x . Formally, define $\mathcal{C}_{s+1}(\rho) = \{y \leq x : y \in \mathcal{C}_s(\rho)\}$ and for established $z \geq x$, $I_{s+1}(z) = \{y \in I_s(w) : w \in \mathcal{C}_s(\rho) \text{ and } w \leq x\}$, and for $z < x$ $I_{s+1}(z) = I_s(z)$.

If the stage has not been halted then let the collection of accessible nodes δ_s be an initial segment of the tree of strategies. δ_s is defined by recursion; the root of the tree is in δ_s , then the action of each node defines the next accessible node.

If a π node is accessible at stage s . Let $t < s$ be the last stage $\pi \infty$ was accessible, $t = 0$ if there was no such stage. If $\text{dom } \Phi_\pi(A_s) < \#t$ then $\pi \text{ fin}$ is accessible; otherwise $\pi \infty$ is accessible. If $\pi \infty$ is accessible, then declare the least y that has not already been established to be established.

If a τ node is accessible at stage s . Let $t < s$ be the last τ -expansionary stage, $t = 0$ if there was no such stage. If $\text{dom } \Psi_\tau(A, W, Q)[s] < \#t$ then $\tau \text{ fin}$ is accessible; otherwise s is a τ -expansionary stage.

If s is a τ -expansionary stage and there is a request token on τ from a μ node, then $\mu \succ \zeta_k \uparrow$ for some son, ζ_k , of τ . If $\Delta_\mu(A, W, p)[s] \uparrow$ then cancel the request. If the request has not been cancelled and the computation $\Psi_\tau(A, W, Q, k)[t]$ does not hold at stage s , then:

- If $\text{prec}(\tau) = \emptyset$ then carry out the request by enumerating $q = \text{fl}_s(\mu)$ into Q_{s+1} . Declare (ρ, x) corrupted if $\rho \infty \prec \mu$, $x \in \mathcal{C}_s(\rho)$ and $(\rho, x) \notin \text{pro}_s(\mu)$. For each ρ let x be the least declared corrupted, then declare all established $x' > x$ taken over by x . If (ρ, x) has been declared corrupted and has not been taken over, then define a new large tracker $\text{tr}_{s+1}(\rho, x)$ for x , halt and initialise all nodes weaker than $\mu \infty$.
- Otherwise move the request token to the longest $\hat{\tau} \in \text{prec}(\tau)$, halt and initialise all nodes weaker than $\mu \infty$.

If the stage has not been halted, then let ρ be a daughter of τ and $z =$

$\text{tr}_s(\rho, x)$ for $x \in \mathcal{C}_s(\rho)$. If $\Gamma_\tau(A, W, z)[s] \uparrow$ then do the first of the following that applies:

- If (ρ, x) is not in an attack then:
 - If x was invented at stage t , define $\Gamma_{\tau, s+1}(A_s, W_s, z) = s$ with large use.
 - Otherwise define $\Gamma_{\tau, s+1}(A_s, W_s, z) = s$ with use $\psi_s^\rho(I_s(\rho, x))$.
- Let r be the stage (ρ, x) began its attack. If $d_r^\rho(z) > d_s^\rho(z)$ then define $\text{pro}_{s+1}(\rho, x)$ to be the empty set, declare the (ρ, x) attack to be finished and define $\Gamma_{\tau, s+1}(A_s, W_s, z) = s$ with use $\psi_s^\rho(I_s(\rho, x))$.
- If (ρ, x) was not in an attack at stage t (the last τ -expansionary stage), then declare (ρ, x) to be fully in an attack and define $\Gamma_{\tau, s+1}(A_s, W_s, z) = s$ with large use and define $\text{pro}_{s+1}(\rho, x)$ to be the set of $(\hat{\rho}, \hat{x})$ such that $\hat{x} \in \mathcal{C}_s(\hat{\rho})$ and $\hat{\rho} \infty \prec \tau$, halt and initialise all nodes to the right of $\rho \infty$.
- If (ρ, x) enumerated a number into A at stage t , then define $\text{pro}_{s+1}(\rho, x)$ to be the set of $(\hat{\rho}, \hat{x})$ such that $\hat{x} \in \mathcal{C}_s(\hat{\rho})$ and $\hat{\rho} \infty \prec \tau$, and define $\Gamma_{\tau, s+1}(A_s, W_s, z) = s$ with large use, halt and initialise all nodes to the right of $\rho \infty$.
- if there is some $(\hat{\rho}, \hat{x}) \in \text{pro}_s(\rho, x)$ with tracker $\text{tr}_t(\hat{\rho}, \hat{x}) = \hat{z}$ such that the computation $\Gamma_{\hat{\rho}}(A, W, \hat{z})[t]$ no longer holds at stage s , or \hat{x} has been uncorrupted between stages t and s , then define $\Gamma_{\tau, s+1}(A_s, W_s, z) = s$ with large use, halt and initialise all nodes to the right of $\rho \infty$.
- Otherwise define $\Gamma_{\tau, s+1}(A_s, W_s, z) = s$ with use $\gamma_{t+1}(z)$.

For $z < t$ if $\Gamma_\tau(A, W, z)[s] \uparrow$ and z is not a tracker for any daughter of τ , then define $\Gamma_{\tau, s+1}(A_s, W_s, z) = 0$ with use 0. If the stage has not been halted then let $\tau \infty$ be accessible.

If a ρ node working for requirement $P_{\Psi, k}$ is accessible at stage s .

Let $t < s$ be the last ρ -expansionary stage, $t = 0$ if there is no such stage. If $\forall z < \#t$, $\Gamma_\tau(A, W, z)[s] = f_s^\rho(z)$ and $o_s^\rho(z) < \omega^2$, then s is a ρ -expansionary stage; otherwise $\rho \text{ fin}$ is accessible.

Suppose s is a ρ -expansionary stage. Let $x \in \mathcal{C}_s(\rho)$ and let $z = \mathbf{tr}_s(x)$. Then do the following:

- If (ρ, x) was either corrupted at stage t then declare x to be fully corrupted.
- If (ρ, x) is in an attack then enumerate $\gamma_s(z)$ into A_{s+1} , halt the stage and initialise all nodes to the right of $\rho \infty$.

If the stage has not been halted then invent a new input by letting $x + 1$ be least such that $\mathbf{tr}_s(x + 1) \uparrow$ then define a new large tracker for $x + 1$. Define $\mathbf{orig}(\rho, x + 1) = \mathbf{tr}_{s+1}(x + 1)$. If $x + 1 > 0$ then declare x established and define $\mathcal{C}_{s+1}(\rho) = \mathcal{C}_s(\rho) \cup \{x\}$ and $I_s(x) = \{x\}$. Let $\rho \infty$ be accessible.

If a ζ_k node, son of τ , is accessible at stage s . Let $t < s$ be the last $\zeta_k \uparrow$ stage, $t = 0$ if there is no such stage. If the computation $\Psi_\tau(A, W, Q, k)[t]$ does not hold at stage s then let $\zeta_k \uparrow$ be accessible; otherwise let $\zeta_k \downarrow$ be accessible.

If an η node is accessible at stage s . Let $t < s$ be the last η -expansionary stage, $t = 0$ if there is no such stage. If $\text{dom } \Theta(A, W)[s] < \#t$ then $\eta \mathbf{fin}$ is accessible; otherwise s is an η -expansionary stage.

Suppose s is an η -expansionary stage. If $\Delta_\eta(A, W, p)[s] \uparrow$ and $p < s$ is not the anchor of any daughter of η , then define $\Delta_{\eta, s+1}(A_s, W_s, p) = 0$ with use 0. Otherwise, if $\Delta_\eta(A, W, p)[s] \uparrow$ and $p = \mathbf{ac}_s(\mu)$ for μ a daughter of η , then let $q = \mathbf{fl}_s(\mu)$ and do the first of the following that applies:

- If $q \in Q_s$ then cancel the follower q , appoint a new large follower, and leave $\Delta_\eta(A, W, p) \uparrow$. Define $\mathbf{pro}_{s+1}(\mu)$ to be the set of (ρ, x) such that (ρ, x) needs protection from μ at stage s . Halt and initialise all nodes weaker than $\mu \infty$.
- If $q < \psi_s^\rho(I_s(\rho, x))$ for some $(\rho, x) \in \mathbf{pro}_s(\mu)$ then cancel the follower q and appoint a new large follower and leave $\Delta_\eta(A, W, p) \uparrow$. Halt and initialise all nodes weaker than $\mu \infty$.
- Otherwise define $\Delta_{\eta, s+1}(A_s, W_s, p) = s$ with use $\theta_s(q)$. Note that we do not halt the stage.

If the stage has not been halted then let $\eta \infty$ be accessible.

If a μ node working for requirement $R_{\Theta,k}$ is accessible at stage s .

Let $t < s$ be the last μ -expansionary stage, $t = 0$ if there is no such stage. If $\forall p < \#t, \Delta_\eta(A, W, p)[s] = f_s^\mu(p)$ and $o_s^\mu(p) < \omega^2$, then s is a μ -expansionary stage, otherwise $\mu \text{ fin}$ is accessible. If s is a μ -expansionary stage then:

- If $\text{ac}_s(\mu) \uparrow$ then define a new large anchor and a new large follower. Define $\text{pro}_{s+1}(\mu)$ to be the set of (ρ, x) such that (ρ, x) needs protection from μ at stage s . Halt and initialise all nodes weaker than $\mu \infty$.
- Let $q = \text{fl}_s(\mu)$. If $q \notin Q_s, \Theta(A, W, q)[s] = 0$ and $\text{prec}(\mu) = \emptyset$, then enumerate q into Q_{s+1} . Declare (ρ, x) corrupted if $\rho \infty \prec \mu, x \in \mathcal{C}_s(\rho)$ and $(\rho, x) \notin \text{pro}_s(\mu)$. For each ρ let x be the least declared corrupted, then declare all established $x' > x$ taken over by x . If (ρ, x) has been declared corrupted and has not been taken over, then define a new large tracker, halt and initialise all nodes weaker than $\mu \infty$.
- If $q \notin Q_s, \Theta(A, W, q)[s] = 0$ and $\text{prec}(\mu) \neq \emptyset$, then send a request token for μ to the longest $\tau \in \text{prec}(\mu)$, halt and initialise all nodes weaker than $\mu \infty$.

If the stage has not been halted then let $\mu \infty$ be accessible.

If a ξ_k node, son of η , is accessible at stage s . Let $t < s$ be the last $\xi \uparrow$ stage, $t = 0$ if there is no such stage. If the computation $\Theta_\eta(A, W, k)[t]$ does not hold at stage s , then let $\xi_k \uparrow$ be accessible; otherwise let $\xi_k \downarrow$ be accessible.

3.4.1 Verification

First note that if $(\rho, x) \in \text{pro}_s(\alpha)$ then $\tau \infty \preceq \alpha$, so $\psi_s^\rho(I_s(\rho, x)) \downarrow$; hence the uses defined in the construction exist.

Lemma 3.1. Let μ be a node working for requirement $R_{\Theta,k}$ with parent η . For all stages s , if $q = \text{fl}_s(\mu)$ is defined then $p = \text{ac}_s(\mu)$ is also defined, and if $\Delta_\eta(A, W, p)[s] \downarrow$ then $\Theta_\eta(A, W, q)[s] \downarrow$ and $\theta_s(q) \leq \delta_s(p)$.

Proof. The anchor only becomes undefined at initialisation. Note that at the first μ -expansionary stage after initialisation an anchor and a follower is appointed. Then at any stage at which a new follower is appointed either a new anchor is also appointed or the anchor is already defined.

Suppose $\Delta_\eta(A, W, p)[s] \downarrow$ and let $t < s$ be the stage at which this computation was defined, and u be the use of this computation. Then stage t is an η -expansionary stage, so $\Theta_\eta(A, W, q)[t] \downarrow$. The computation $\Delta_\eta(A, W, p)$ has not changed between stages t and s so $A_s \upharpoonright u = A_t \upharpoonright u$ and $W_s \upharpoonright u = W_t \upharpoonright u$. At stage t , u was defined so that $u \geq \theta_t(q)$; hence $A_s \upharpoonright \theta_t(q) = A_t \upharpoonright \theta_t(q)$ and $W_s \upharpoonright \theta_t(q) = W_t \upharpoonright \theta_t(q)$. Then it follows that the use of $\Theta_\eta(A, W, q)[t]$ has not changed, so $\Theta_\eta(A, W, q)[s] \downarrow$ and $\theta_s(q) \leq \delta_s(p)$. \square

Lemma 3.2. Let τ be a node working for requirement P_Ψ . Let s be the stage a request is carried out and $t \leq s$ be the stage τ acts on the request, then there are no τ -expansionary stages between stage t and stage s .

Proof. If $\text{prec}(\tau) = \emptyset$ then τ carries out the request; hence $s = t$ and we are done. So we suppose $\text{prec}(\tau) \neq \emptyset$, then at stage t , τ passes the request to a node $\hat{\tau}$ which has a son, $\hat{\zeta}_{\hat{k}}$, with $\hat{\zeta}_{\hat{k}} \upharpoonright \preceq \tau$. Then τ is only accessible at $\hat{\zeta}_{\hat{k}} \upharpoonright$ stages, so there has been a change in the computation $\Psi_{\hat{\rho}}(A, W, Q, \hat{k})$ between two τ stages. Then while $\hat{\tau}$ has the request any such changes would cause the request to be acted on, and when this happens the stage is halted. Let t_0 be the stage $\hat{\tau}$ acts on the request, then there are no τ -expansionary stages between stage t and t_0 . If $t_0 = s$ we are done; otherwise it is passed to some τ' . Let t_1 be the stage τ' acts on the request; then we repeat the same argument to find there are no $\hat{\tau}$ stages between stage t_0 and t_1 . If $t_1 = s$ then since every τ stage is a $\hat{\tau}$ stage we are done. Continue this argument, since $\text{prec}(\tau)$ is finite and the request is carried out at stage s , $t_n = s$ for some n ; hence there are no τ -expansionary stages between stage t and stage s . \square

Lemma 3.3. For all stages s , (1), (2), and (3) hold.

(1) Let ρ be a node working for requirement $P_{\Psi, k}$ with parent τ . Let $z = \text{tr}_s(\rho, x)$ for $x \in \mathcal{C}_s(\rho)$. Suppose (ρ, x) is not in an attack at stage s and

$\Gamma_\tau(A, W, z)[s] \downarrow$. Let t be the stage the computation $\Gamma_\tau(A, W, z)[s]$ was defined. Then $Q_t \upharpoonright \psi_t(I_t(x)) = Q_s \upharpoonright \psi_t(I_t(x))$ and for all $\hat{x} \in I_s(\rho, x)$, $\Psi(A, W, Q, \hat{x})[s] \downarrow$ and $\psi_s(I_s(\rho, x)) \leq \gamma_s(z)$.

- (2) Let μ be a node working for requirement $R_{\Theta, k}$ and let $q = \mathbf{fl}_s(\mu)$. Suppose ρ is a node such that $\rho \infty$ to the left of μ such that $x \in \mathcal{C}_s(\rho)$ and $(\rho, x) \notin \mathbf{pro}_s(\mu)$. If q is enumerated into Q_{s+1} at stage s and $q < \psi_s^\rho(I_s(\rho, x))$, then either (ρ, x) is in an attack at stage s or $\Gamma_\tau(A, W, \mathbf{tr}(x))[s] \uparrow$.
- (3) Let μ be a node working for requirement $R_{\Theta, k}$, and let $p = \mathbf{ac}_s(\mu)$ and $q = \mathbf{fl}_s(\mu)$. If $q \notin Q_s$ and there is a $(\rho, x) \in \mathbf{pro}_s(\mu)$ such that $q < \psi_s^\rho(I_s(\rho, x))$, then $\Delta_\eta(A, W, p)[s] \uparrow$ and q is not enumerated into Q_{s+1} at stage s .

Proof. We prove the lemma by simultaneous induction on the stage s . Clearly at $s = 0$, (1), (2) and (3) hold. So let $s > 0$ and suppose (1), (2) and (3) hold for all stages $t < s$.

Consider (1) at stage s . Suppose $Q_t \upharpoonright \psi_t(I_t(x)) \neq Q_s \upharpoonright \psi_t(I_t(x))$. Then at stage $r \in [t, s)$ some μ enumerated its follower q into Q_{r+1} and $q < \psi_t(x)$. Note that $\Gamma_\rho(A, W, \mathbf{tr}(x))[r] \uparrow$ at a stage r where a new tracker is appointed. Therefore $z = \mathbf{tr}_s(x) = \mathbf{tr}_t(x)$ and so x has not been corrupted between stages t and s . Then since $x \in \mathcal{C}_s(\rho)$ it is also the case that x has not been taken over between stages t and s . Recall that by the construction, if $\mu \succeq \rho \infty$ and $(\rho, x) \notin \mathbf{pro}_r(\mu)$ then x is either corrupted or taken over; hence either $(\rho, x) \in \mathbf{pro}_r(\mu)$ or μ is to the right of $\rho \infty$. Now note that an attack is only declared finished at a stage where $\Gamma_\rho(A, W, z) \uparrow$. Then it follows that since (ρ, x) is not in an attack at stage s , it is also the case that (ρ, x) is not in an attack at any stage r such that $r \in [t, s]$. (3) holds at stage r ; hence $(\rho, x) \notin \mathbf{pro}_r(\mu)$. (2) holds at stage r ; hence either $\Gamma_\rho(A, W, \mathbf{tr}(x))[r] \uparrow$ or (ρ, x) is in an attack at stage r . But (ρ, x) is not in an attack at any stage r such that $r \in [t, s]$ and $\Gamma_\rho(A, W, \mathbf{tr}(x))[r] \downarrow$, a contradiction. Hence

$Q_t \upharpoonright \psi_t(I_t(x)) = Q_s \upharpoonright \psi_t(I_t(x))$. Let u be the use of the computation $\Gamma_\tau(A, W, z)[s]$. At stage t , the use u was defined to be $\psi_t(I_t(x))$ so $(A, W, Q)_t \upharpoonright \psi_t(I_t(x)) = (A, W, Q)_s \upharpoonright \psi_t(I_t(x))$. Then it follows that for all $\hat{x} \in I_s(x)$, $\Psi(A, W, Q, \hat{x})[s] \downarrow$ and $\psi_s(I_s(x)) \leq \gamma_s(z)$. Thus **(1)** holds at stage s .

Consider **(2)** at stage s . First, if (ρ, x) is in an attack at stage s then we are done. So suppose (ρ, x) is not in an attack at stage s . Let τ be the parent of ρ and let r_0 be the stage $\text{pro}_s(\mu)$ was defined. Notice that μ is initialised at every ρ -expansionary stage; hence x was established before stage r_0 . $(\rho, x) \notin \text{pro}_s(\mu)$; therefore either μ believes $\text{dom } \Psi_\rho(A, W, Q) = k'$ and $I_{r_0}(x) \not\subseteq k'$, or μ lies to the right of $\tau \infty$.

Case 1. μ lies to the right of $\tau \infty$. Then μ is initialised at every τ -expansionary stage; hence we do not see any $\Psi_\tau(A, W, Q)$ computations recover while q is the follower of μ . Let r_1 be the stage q is appointed. At stage r_1 , q is appointed to be large; therefore for all x' such that $\Psi_\tau(A, W, Q, x')[r_1] \downarrow$, $q > \psi_{r_1}(x')$. If there are any changes below the use of any of these computations while q is the follower for μ then they do not recover until after stage s . So it follows that for all x' such that $\Psi_\tau(A, W, Q, x')[s] \downarrow$, $q > \psi_s(x')$. So it is not possible to have $q < \psi_s^p(I_s(\rho, x))$ at stage s .

Case 2. μ believes $\text{dom } \Psi_\rho(A, W, Q) = k'$ and $I_{r'}(x) \not\subseteq k'$. Now $\tau \in \text{prec}(\mu)$ so $\text{prec}(\mu) \neq \emptyset$. Therefore enumeration of q into Q_{s+1} at stage s is carried out by a request token at some $\hat{\tau}$ (note that it could be that $\hat{\tau} = \tau$). Let $t \leq s$ be the stage τ acts on the request and $r < t$ be the stage τ receives the request. Then stage t is the first τ -expansionary stage after a change in the computation $\Psi_\rho(A, W, Q, k')[r]$. Notice that if there is a stage $t > r'$ such that $I_{r'}(x) \neq I_t(x)$ then μ is initialised; hence $I_s(x) \not\subseteq k'$. Now let x' be the least element in the interval $I_t(k')$. Suppose (ρ, x') is in an attack at stage t . Note that μ is not initialised between stages r_0 and s , so this attack started before stage r_0 . Then at the last ρ -expansionary stage a number is enumerated into A and all established inputs greater than x' are taken over by x' . Then $x = x'$, but an attack can only be declared finished at τ -expansionary stages, and by Lemma 3.2 there are no

τ -expansionary stages between stage t and s ; hence if (ρ, x') is in an attack at stage t then (ρ, x) is in an attack at stage s , a contradiction. Then (ρ, x') is not in an attack at stage t , and **(1)** holds at stage t , so it follows that $\psi_t(k') \leq \gamma_{t+1}(\mathbf{tr}_{t+1}(x')) \leq \gamma_{t+1}(\mathbf{tr}_{t+1}(x))$. Then the change in the computation $\Psi_\rho(A, W, Q, k')[r]$ between stages r and t causes $\Gamma_\rho(A, W, \mathbf{tr}(x))[t] \uparrow$. t is the first τ -expansionary stage after stage r ; at this stage the request is acted on and we do not define $\Gamma_\rho(A, W, \mathbf{tr}(x))[t]$. $\Gamma_\rho(A, W, \mathbf{tr}(x))[s]$ is only defined at τ -expansionary stages, and by Lemma 3.2 there are no such stages between stage t and stage s ; hence $\Gamma_\rho(A, W, \mathbf{tr}(x))[s] \uparrow$.

Consider **(3)** at stage s . Suppose at stage s there is a node μ with follower q and anchor p , and there is a $(\rho, x) \in \mathbf{pro}_s(\mu)$ such that $q < \psi_s^\rho(I_s(\rho, x))$. If $\Delta_\mu(A, W, p)[s] \uparrow$ and $\mathbf{prec}(\mu) \neq \emptyset$ then the enumeration is carried out by a τ node acting on a request, but since $\Delta_\mu(A, W, p)[s] \uparrow$ the request will be cancelled; hence q is not enumerated into Q_{s+1} . If $\Delta_\mu(A, W, p)[s] \uparrow$ and $\mathbf{prec}(\mu) = \emptyset$ then q is enumerated into Q_{s+1} at an η -expansionary stage, but $\Delta_\mu(A, W, p)[s] \uparrow$ and there is a $(\rho, x) \in \mathbf{pro}_s(\mu)$ such that $q < \psi_s^\rho(I_s(\rho, x))$, so q will be cancelled before it can be enumerated into Q_{s+1} . Therefore it suffices to show that $\Delta_\mu(A, W, p)[s] \uparrow$.

So suppose $\Delta_\mu(A, W, p)[s] \downarrow$ and let t be the stage this computation was defined. Let r_0 be the stage $\mathbf{pro}_s(\mu)$ was defined, and let r_1 be the stage q was appointed (note that $r_0 \leq r_1$). x was established before stage r_1 and $\tau \infty \preceq \eta$, so it follows that $q > \psi_{r_1}^\rho(I_{r_1}(x))$. Notice that at stage t , $\Delta_\mu(A, W, p)[t] \uparrow$. If $q < \psi_t^\rho(I_t(x))$ we would not define $\Delta_\mu(A, W, p)[t+1]$, instead we would cancel q , but $q = \mathbf{fl}_s(\mu)$; hence $q > \psi_t^\rho(I_t(x))$. Then at stage t we define $\Delta_\mu(A, W, p)[s]$ with use $\theta(q)$, and since $\theta(q) \geq q$ it follows that $\delta_s(p) > \psi_t^\rho(I_t(x))$. Now at stage s , $q < \psi_s^\rho(I_s(x))$ so $Q_t \upharpoonright \psi_t^\rho(I_t(x)) \neq Q_s \upharpoonright \psi_t^\rho(I_t(x))$. Let $\hat{\mu}$ be the node that enumerates its follower \hat{q} into Q_{r+1} at stage $r \in [t, s)$. Notice $q \notin Q_s$ and $q = \mathbf{fl}_s(\mu) = \mathbf{fl}_t(\mu)$, so $\hat{\mu} \neq \mu$.

Now μ has not been initialised between stages t and s ; hence $\hat{\mu}$ is weaker than μ . Let r_2 be the stage that $\mathbf{pro}_r(\hat{\mu})$ was defined. Notice $r_2 > r_1$ because $\hat{\mu}$ is initialised when q is appointed. Now this means x was already

established at stage r_2 . If $(\rho, x) \in \text{pro}_r(\hat{\mu})$, $r < s$ so **(3)** holds at stage r ; hence \hat{q} is not enumerated into Q_{r+1} . But \hat{q} is enumerated into Q_{r+1} ; therefore $(\rho, x) \notin \text{pro}_r(\hat{\mu})$. But x was established before stage r_2 , so $\hat{\mu}$ lies to the right of $\rho \infty$, $\hat{\mu}$ believes $\text{dom } \Psi_\rho(A, W, Q) = k'$ and $I_{r_2}(\rho, x) \not\subseteq k'$. Now **(2)** holds at stage r , so either (ρ, x) is in an attack at stage r or $\Gamma_\rho(A, W, z)[r] \uparrow$.

Suppose (ρ, x) is in an attack at stage r . Since $(\rho, x) \in \text{pro}_s(\mu)$, (ρ, x) was not in an attack at stage r_0 ; hence this (ρ, x) attack starts at stage $r_3 \in (r_0, r)$. Notice that this attack is prompted by a W change below $\psi(I(x))$, as discussed above $\delta_s(p) > \psi_t^\rho(I_t(x))$ and the computation $\Delta_\mu(A, W, p)[s]$ was defined at stage t ; hence this attack was started before stage t . Now if μ was to the right of $\rho \infty$ then it would be initialised at stage r_3 but $r_3 \in (r_0, r)$ so $\mu \succ \rho \infty$ and hence $\eta \succeq \rho \infty$. Then there are no η -expansionary stage until the attack is finished. (ρ, x) is still in an attack at stage r ; hence $t > r$, but this is a contradiction because $r \in [t, s)$. Therefore $Q_t \upharpoonright \psi_t^\rho(I_t(x)) = Q_s \upharpoonright \psi_t^\rho(I_t(x))$ and so $q > \psi_s^\rho(I_s(x))$.

Therefore if $q < \psi_s^\rho(I_s(x))$ then $\Delta_\mu(A, W, p)[s] \uparrow$ and q is not enumerated into Q_{s+1} at stage s . \square

Lemma 3.4. Let ρ be a node working for requirement $P_{\Psi, k}$ and let $x \in \mathcal{C}_s(\rho)$. Let μ be a node working for requirement $R_{\Theta, k}$, and let $q = \text{fl}_s(\mu)$. Suppose q is enumerated into Q_{s+1} at stage s . If (ρ, x) is not in an attack, $q < \psi_s(I_s(\rho, x))$ and $\Gamma_\tau(A, W, \text{tr}(x))[s] \downarrow$, then $\mu \succ \rho \infty$ and $(\rho, x) \notin \text{pro}_s(\mu)$.

Proof. If $(\rho, x) \in \text{pro}_s(\mu)$ then by Lemma 3.3 **(3)**, q is not enumerated into Q_{s+1} , a contradiction.

If $(\rho, x) \notin \text{pro}_s(\mu)$ and μ is to the right of $\rho \infty$, then μ is initialised at every ρ -expansionary stage; hence x was established before q was appointed. But $\Gamma_\tau(A, W, \text{tr}(x))[s] \downarrow$ and (ρ, x) is not in an attack; this is a contradiction to Lemma 3.3 **(2)**. \square

Lemma 3.5. Let ρ be a node working for requirement $P_{\Psi, k}$ and let $x \in \mathcal{C}_s(\rho)$. Suppose x is declared corrupted at stage s . If at stage $t > s$, (ρ, x) is

not in an attack, and some μ enumerates a follower q into Q_{t+1} such that $q < \psi_t(I_t(\rho, x))$ and $\Gamma_\tau(A, W, \text{tr}(x))[t] \downarrow$, then x is taken over at stage t .

Proof. (ρ, x) was declared corrupted at stage s . Then at stage s some $\hat{\mu}$ enumerated its follower \hat{q} into Q_{s+1} , such that $\hat{\mu} \succ \rho \infty$ and x was the least input for ρ such that $(\rho, x) \notin \text{pro}_s(\hat{\mu})$. At stage $t > s$, μ enumerates q into Q_{t+1} such that $q < \psi_t(I_t(\rho, x))$ and $\Gamma_\tau(A, W, \text{tr}(x))[t] \downarrow$. It follows from Lemma 3.4 that $\mu \succ \rho \infty$. The stage is halted when a protected set is defined; hence $\text{pro}_s(\hat{\mu})$ and $\text{pro}_t(\mu)$ were defined at different stages; call these stages \hat{r} and r respectively. x was the least input that was not yet established at stage \hat{r} . If $r > \hat{r}$ then since $\mu \succ \rho \infty$, a new input has been established since stage \hat{r} ; hence x has already been established at stage when we define $\text{pro}_t(\mu)$ at stage r ; hence $(\rho, x) \in \text{pro}_t(\mu)$, but this contradicts Lemma 3.3 (3). Now suppose $r < \hat{r}$. Let x' be the least input for ρ that was not yet established at stage r . $\hat{\mu} \succ \rho \infty$, so \hat{r} is a ρ -expansionary stage; hence at stage \hat{r} a new input is established. So $x' \neq x$. Then x is not the least input that is not in $\text{pro}_s(\mu)$; hence at stage t , x is taken over. \square

Lemma 3.6. Let ρ and $\hat{\rho}$ be nodes working for requirements $P_{\Psi, k}$ and $P_{\hat{\Psi}, \hat{k}}$ respectively. Let $x \in \mathcal{C}_s(\rho)$ and $\hat{x} \in \mathcal{C}_s(\hat{\rho})$, with trackers z and \hat{z} respectively. Let $(\hat{\rho}, \hat{x})$ be in an attack at stage s , and let t be the stage the $(\hat{\rho}, \hat{x})$ attack is finished. If $(\rho, x) \in \text{pro}_s(\hat{\rho}, \hat{x})$ then x is not corrupted at stage $r \in [s, t)$.

Proof. $(\rho, x) \in \text{pro}_s(\hat{\rho}, \hat{x})$ so $\rho \infty \prec \hat{\rho}$. x is corrupted by the enumeration of a follower of μ , q , into Q_{r+1} at stage r such that $\mu \succ \rho \infty$. Then consider the following cases:

Case 1. $\mu \succ \hat{\rho} \infty$. Then μ is not accessible during the $(\hat{\rho}, \hat{x})$ attack; hence x is not corrupted at stage $r \in [s, t)$.

Case 2. $\mu \infty \prec \hat{\rho}$ or μ is to the left of $\hat{\rho}$. Then when x is corrupted at stage r , $\hat{\rho}$ is initialised causing the attack to stop; hence $t = r$.

Case 3. μ is to the right of $\hat{\rho}$. $(\rho, x) \in \text{pro}_s(\hat{\rho}, \hat{x})$, so x was already established at the stage $\text{pro}_s(\hat{\rho}, \hat{x})$ was defined and at this stage μ is initialised. Now if x was corrupted after stage s then the corrupting q was appointed

after stage s , but $x \in \mathcal{C}_s(\rho)$ so $(\rho, x) \in \text{pro}_r(\mu)$ and so x is not corrupted after stage s .

Case 4. $\mu \succ \hat{\tau} \infty$ and μ is to the right of ρ . Then q was appointed at a $\hat{\tau}$ -expansionary stage. Consider the stage $\text{pro}_s(\hat{\rho}, \hat{x})$ was defined. Then at the last $\hat{\tau}$ -expansionary stage before we define $\text{pro}_s(\hat{\rho}, \hat{x})$ either $(\hat{\rho}, \hat{x})$ was not in an attack or $(\hat{\rho}, \hat{x})$ enumerated a number into A . Then μ was initialised at this stage. $\mu \succ \hat{\tau} \infty$ so $\text{pro}_r(\mu)$ was defined after (or at) the stage $\text{pro}_s(\hat{\rho}, \hat{x})$ was defined, but x was already established at this stage; hence $(\rho, x) \in \text{pro}_r(\mu)$ and so x is not corrupted after stage s .

Therefore x is not corrupted at stage $r \in [s, t)$. □

Lemma 3.7. Let ρ be a node working for requirement $P_{\Psi, k}$ with parent τ . Let $z = \text{tr}_s(\rho, x)$ for $x \in \mathcal{C}_s(\rho)$. For all stages s such that (ρ, x) is in an attack, if $\Gamma_\tau(A, W, z)[s] \downarrow$ then for all $(\hat{\rho}, \hat{x}) \in \text{pro}_s(\rho, x)$, $\Gamma_{\hat{\rho}}(A, W, \text{tr}(\hat{\rho}, \hat{x}))[s] \downarrow$ and $\gamma_s(z) \geq \gamma_s^{\hat{\rho}}(\text{tr}_s(\hat{\rho}, \hat{x}))$.

Proof. Let s be the least counterexample. At stage s , $\Gamma_\tau(A, W, z)[s] \downarrow$ and there is some $(\hat{\rho}, \hat{x}) \in \text{pro}_s(\rho, x)$ with tracker \hat{z} , such that $\Gamma_{\hat{\rho}}(A, W, \hat{z})[s] \uparrow$ or $\gamma_s(z) < \gamma_s^{\hat{\rho}}(\hat{z})$. Let t be the stage at which the computation $\Gamma_\tau(A, W, z)[s]$ was defined, and let u be the use of this computation. $\text{pro}_s(\rho, x)$ is only redefined when $\Gamma_\tau(A, W, z)$ is defined, so $\text{pro}_s(\rho, x) = \text{pro}_t(\rho, x)$. $(\hat{\rho}, \hat{x}) \in \text{pro}_s(\rho, x)$, so $\hat{\tau} \infty \prec \tau$; hence t is a $\hat{\tau}$ -expansionary stage. Then at stage t , $\Gamma_{\hat{\rho}}(A, W, \text{tr}_t(\hat{x}))[t] \downarrow$.

If u was defined to be large at stage t , then $\gamma_t(z) \geq \gamma_t^{\hat{\rho}}(\text{tr}_t(\hat{x}))$. If u was not defined to be large, then it was defined to be $\gamma_r(z)$ where $r < t$ is the last τ -expansionary stage. s is the least counterexample, so at stage r $\gamma_r(z) \geq \gamma_r^{\hat{\rho}}(\text{tr}_r(\hat{x}))$. Note that by Lemma 3.6, \hat{x} has not been corrupted between stages r and t . Since u was not defined to be large, it is also the case that \hat{x} has not been uncorrupted between stages r and t ; hence $\text{tr}_r(\hat{x}) = \text{tr}_t(\hat{x})$. If $\gamma_r^{\hat{\rho}}(\text{tr}_t(\hat{x})) \neq \gamma_t^{\hat{\rho}}(\text{tr}_t(\hat{x}))$ then the computation $\Gamma_{\hat{\rho}}(A, W, \text{tr}(\hat{\rho}, \hat{x}))[r]$ no longer holds at stage t ; hence we would define the use to be large, a contradiction. So at stage r , $\gamma_r^{\hat{\rho}}(\text{tr}_t(\hat{x})) = \gamma_t^{\hat{\rho}}(\text{tr}_t(\hat{x}))$, then it follows that $\gamma_t(z) \geq \gamma_t^{\hat{\rho}}(\text{tr}_t(\hat{\rho}, \hat{x}))$.

Now the computation $\Gamma_\tau(A, W, z)[s]$ was defined at stage t , so $A_t \upharpoonright u = A_s \upharpoonright u$ and $W_t \upharpoonright u = W_s \upharpoonright u$. $u = \gamma_t(z) \geq \gamma_t^{\hat{\rho}}(\text{tr}_t(\hat{x}))$, so it follows that the computation $\Gamma_{\hat{\rho}}(A, W, \text{tr}_t(\hat{\rho}, \hat{x}))[t]$ has not changed between stage t and stage s . By Lemma 3.6, \hat{x} has not been corrupted between stages t and s . $u \geq \gamma_t^{\hat{\rho}}(\text{tr}_t(\hat{x})) > \gamma_t^{\hat{\rho}}(\text{orig}(\hat{x}))$ therefore \hat{x} has not been uncorrupted between stages t and s ; hence $\text{tr}_t(\hat{\rho}, \hat{x}) = \text{tr}_s(\hat{\rho}, \hat{x}) = \hat{z}$. Then $\Gamma_{\hat{\rho}}(A, W, \hat{z})[s] \downarrow$ and $\gamma_s(z) \geq \gamma_s^{\hat{\rho}}(\hat{z})$. \square

Lemma 3.8. Let μ be a node working for requirement $R_{\Theta, k}$. Let s and $t > s$ be successive μ -expansionary stages with $p = \text{ac}_s(\mu) = \text{ac}_t(\mu)$. Let $u = \delta_s(p)$. If $A_t \upharpoonright u \neq A_s \upharpoonright u$ or $W_t \upharpoonright u \neq W_s \upharpoonright u$, then $o_t^\mu(p) < o_s^\mu(p)$.

Proof. At stage $r \in (s, t)$, $\Delta_\eta(A, W, p)[r] \uparrow$, then at the next η -expansionary stage $r' \in (r, t]$, $\Delta_\eta(A, W, p)[r'] = r'$.

$$\Delta_\eta(A, W, p)[s] \leq s < r' = \Delta_\eta(A, W, p)[r'] \leq \Delta_\eta(A, W, p)[t]$$

so $\Delta_\eta(A, W, p)[s] \neq \Delta_\eta(A, W, p)[t]$. s and t are μ -expansionary stages, so $\Delta_\eta(A, W, p)[s] = f_s^\mu(p)$ and $\Delta_\eta(A, W, p)[t] = f_t^\mu(p)$, then $f_s^\mu(p) \neq f_t^\mu(p)$; hence $o_t^\mu(p) < o_s^\mu(p)$ since $\langle f_s^\mu, o_s^\mu \rangle$ is an ω^2 -computable approximation. \square

Lemma 3.9. Let μ be a node working for requirement $R_{\Theta, k}$. μ sends finitely many requests and enumerates finitely many followers into Q while a particular p is the anchor.

Proof. Suppose there are finitely many μ -expansionary stages. A request is sent at a μ expansionary stage; hence only finitely many requests are sent. If $\text{prec}(\mu) = \emptyset$ then a follower is enumerated into Q at a μ -expansionary stage; hence only finitely many followers are enumerated into Q . If $\text{prec}(\mu) \neq \emptyset$ then a follower is enumerated into Q when a request is carried out. There are only finitely many requests sent and only one follower can be enumerated into Q per request; thus finitely many followers are enumerated into Q .

Now suppose there are infinitely many μ -expansionary stages. Now if p is eventually cancelled, clearly only finitely many request are sent and finitely

many followers are enumerated into Q while p is the anchor. So suppose p is never cancelled. Then μ is not initialised while p is the anchor. Once a follower has been enumerated into Q , a new follower is appointed only if there is a stage t where $\Delta_\mu(A, W, p)[t] \uparrow$. Note that when a new follower is appointed, the stage is halted. A follower is enumerated at a μ -expansionary; this stage is also an η -expansionary stage, so $\Delta_\mu(A, W, p)[s] \downarrow$ at the stage a follower is enumerated into Q . But infinitely many followers are enumerated into Q , so after each stage s where a follower is enumerated into Q , there is a stage t where $\Delta_\mu(A, W, p)[t] \uparrow$. Let r be the first μ -expansionary stage after stage t and let $u = \delta_s(p)$. Then $A_t \upharpoonright u \neq A_r \upharpoonright u$ or $W_t \upharpoonright u \neq W_r \upharpoonright u$, and by Lemma 3.8, $o_t^\mu(p) < o_s^\mu(p)$. There are infinitely many of these stages; then it follows that the ordinal $o^\mu(p)$ decreases infinitely often, but $\langle f_s^\mu, o_s^\mu \rangle$ is an ω^2 -computable approximation, so this is a contradiction. Therefore finitely many numbers are enumerated into Q . Suppose μ sends infinitely many requests. Then only finitely many of these requests are carried out by the previous argument. μ is not initialised while p is the anchor, so these requests are cancelled due to $\Delta_\mu(A, W, p) \uparrow$. Then we can follow the same argument as above to show that this cannot happen as $\langle f_s^\mu, o_s^\mu \rangle$ is an ω^2 -computable approximation. \square

Lemma 3.10. Let μ be a node working for requirement $R_{\Theta, k}$. Then while a particular p is the anchor for μ , there are finitely many stages such that $\text{pro}_s(\mu) \neq \text{pro}_{s+1}(\mu)$.

Proof. If p is eventually cancelled, then clearly $\text{pro}(\mu)$ can only be redefined finitely many times while p is the anchor. So let p be the anchor of μ that is never cancelled. While p is the anchor for μ , $\text{pro}(\mu)$ gets redefined at a stage s where $\Delta_\eta(A, W, p)[s] \uparrow$ and $\text{fl}_s(\mu) \in Q_s$. Notice that this means that a new follower is appointed at every stage that $\text{pro}(\mu)$ gets redefined while p is the anchor. Then by Lemma 3.9, finitely many followers are enumerated into Q while p is the anchor; hence $\Delta_\eta(A, W, p)[s] \uparrow$ and $\text{fl}_s(\mu) \in Q_s$, finitely often. Therefore we redefine $\text{pro}(\mu)$ at finitely many stages. \square

Lemma 3.11. Let ρ be a node working for requirement $P_{\Psi,k}$ with parent τ , such that $\rho \infty$ is initialised finitely often. Then for all x there exists a stage t such that for all $s > t$, $I_s(\rho, x) = I_t(\rho, x)$.

Proof. First note that the interval $I(\rho, x)$ can only be redefined to be larger than it previously was. Suppose there are finitely many ρ -expansionary stages. Notice that finitely many inputs are ever established. Then there are only finitely many x' that could be added to the interval $I(\rho, x)$; hence it cannot change infinitely often.

Now suppose there are infinitely many ρ -expansionary stages. Then $\Gamma_\tau(A, W)$ is ω^2 -c.a. Suppose there is some x such that the interval $I(\rho, x)$ changes infinitely often. ρ is initialised finitely often, so let r be the last stage that ρ is initialised. Then after stage r , x will change trackers at most three times (one when x is established, one when it is corrupted, and then back to the original tracker if x is ever uncorrupted). So let z be the last tracker for x . Now the interval $I(\rho, x)$ increases infinitely often, so the use $\psi_s(I_s(\rho, x))$ will go to infinity. By Lemma 3.3 (1), for all stages s such that $z = \mathbf{tr}_s(x)$, $\gamma_s(z) \geq \psi_s(I_s(\rho, x))$. z is the last tracker of x (and is never cancelled), so the use $\gamma_s(z)$ must go to infinity. But this contradicts that $\Gamma_\tau(A, W)$ is ω^2 -c.a. Therefore there cannot be an x such that its interval $I(\rho, x)$ changes infinitely often. \square

Lemma 3.12. Let μ be a node working for requirement $R_{\Theta,k}$ with parent η . At all stages s at which $p = \mathbf{ac}_s(\mu)$, there are finitely many followers appointed.

Proof. If p is eventually cancelled then certainly only finitely many followers are appointed while p is the anchor. So let p be an anchor that is never cancelled. While a particular p is the anchor, a new follower is appointed at an η -expansionary stage s , where $\Delta_\eta(A, W, p)[s] \uparrow$ and either $\mathbf{fl}_s(\mu) \in Q_s$ or $q < \psi_s^\rho(I_s(\rho, x))$ for some $(\rho, x) \in \mathbf{pro}_s(\mu)$. Notice that if there are finitely many η -expansionary stages, then clearly only finitely many followers are appointed. So suppose there are infinitely many η -expansionary stages. At

each η -expansionary stage such that $\Delta_\eta(A, W, p)[s] \uparrow$ and $\mathbf{fl}_s(\mu) \in Q_s$, we appoint a new follower. Then it follows that if this happens infinitely often, μ could enumerate infinitely many followers into Q while p is the anchor, a contradiction to Lemma 3.9. By Lemma 3.10, since p is never cancelled, $\mathbf{pro}_s(\mu)$ eventually stabilises. Let $\mathbf{pro}(\mu)$ be the last protected set defined for μ while p is the anchor. Note $(\rho, x) \in \mathbf{pro}(\mu)$, $\rho \infty$ is either to the left of μ or $\rho \infty \prec \mu$. μ is initialised finitely often since p is never cancelled, so it follows that ρ is also initialised finitely often. Then by Lemma 3.11, for each $(\rho, x) \in \mathbf{pro}(\mu)$ the interval $I_s(\rho, x)$ eventually stabilises. Let $I(x)$ be the last interval defined for (ρ, x) while p is the anchor. Now if $(\rho, x) \in \mathbf{pro}(\mu)$ then either μ believes $\Psi_\tau(A, W, Q)$ is total or μ believes $\text{dom } \Psi_\rho(A, W, Q) = k'$ for some k' , and $I_r(\rho, x) \subseteq k'$ where r is the stage $\mathbf{pro}(\mu)$ was defined. We consider each case separately.

Case 1. $(\rho, x) \in \mathbf{pro}(\mu)$ and μ believes $\Psi_\tau(A, W, Q)$ is total, where τ is the parent of ρ . Then there is a daughter of τ , $\hat{\rho}$, such that $\hat{\rho} \infty \preceq \mu$. Note that it could be that $\rho = \hat{\rho}$. There are infinitely many η -expansionary stages, so $\hat{\rho} \infty$ is accessible infinitely often; hence $\Psi_\tau(A, W, Q)$ is total. Therefore the use $\psi^\rho(I(x))$ eventually stabilises; hence for $(\rho, x) \in \mathbf{pro}(\mu)$ such that μ believes $\Psi_\tau(A, W, Q)$ is total, we see $q < \psi_s(I_s(\rho, x))$ finitely often.

Case 2. $(\rho, x) \in \mathbf{pro}(\mu)$, μ believes $\text{dom } \Psi_\rho(A, W, Q) = k'$ for some k' , and $I_r(\rho, x) \subseteq k'$ where r is the stage $\mathbf{pro}(\mu)$ was defined. Notice that if there is a stage $t > r$ such that $I_r(x) \neq I_t(x)$, then μ would be initialised. But the assumption is that p is not cancelled after stage r ; hence $I_r(x) = I(x)$. Now suppose we see $\mathbf{fl}_s(\mu) < \psi_s^\rho(I_s(\rho, x))$ at infinitely many η -expansionary stages. Let x' be the least input in the interval $I(x)$ such that the use $\psi_s(x')$ increases infinitely often. $I_r(x) \subseteq k'$ so $x' < k$. Then by the assignment of requirements, there is a brother of ρ , $\zeta_{x'}$ such that $\zeta_{x'} \downarrow \preceq \mu$. If the use $\psi(I(x))$ increases infinitely often then there must be infinitely many $\zeta_{x'}$ stages where the computation $\Psi_\rho(A, W, Q, x')$ has changed; hence there are infinitely many $\zeta_{x'} \uparrow$ stages. μ is initialised at every $\zeta_{x'} \uparrow$ stage, but μ is only initialised finitely often, a contradiction.

Therefore only finitely many followers are appointed while p is the anchor of μ . \square

Lemma 3.13. Suppose η is accessible infinitely often and initialised finitely often, then either $\eta \infty$ or $\eta \mathbf{fin}$ is accessible infinitely often and initialised finitely often.

Proof. Suppose there are finitely many η -expansionary stages, then since η can only halt the stage at η -expansionary stages, $\eta \mathbf{fin}$ is accessible infinitely often. Nodes extending $\eta \infty$ are accessible finitely often and η is initialised finitely often, so $\eta \mathbf{fin}$ is initialised finitely often.

Now suppose there are infinitely many η -expansionary stages, but finitely many $\eta \infty$ stages, so let s be the stage such that for all $t > s$, t is not an $\eta \infty$ stage. Anchors are only appointed at μ -expansionary stages, so after stage s no daughter of η is accessible. Then after stage s there are no new anchors appointed for any daughter of η . η must halt the stage infinitely often (in particular infinitely often after stage s). When η halts the stage, a new follower for some daughter of μ is appointed. Then since no new anchors are appointed and there are finitely many daughters of η with anchors, there is some μ such that a new follower is appointed infinitely often. But this is a contradiction to Lemma 3.12. Hence η halts the stage finitely often, so there are infinitely many $\eta \infty$ stages. Since η is initialised finitely often and halts the stage finitely often, $\eta \infty$ is initialised finitely often. \square

Lemma 3.14. Suppose μ is accessible infinitely often and initialised finitely often, then either $\mu \infty$ or $\mu \mathbf{fin}$ is accessible infinitely often and initialised finitely often.

Proof. Suppose there are finitely many μ -expansionary stages. Then since μ can only halt the stage at a μ -expansionary stage, $\mu \mathbf{fin}$ is accessible infinitely often. Since nodes extending $\mu \infty$ are accessible finitely often and μ is initialised finitely often, $\mu \mathbf{fin}$ is initialised finitely often.

Now suppose there are infinitely many μ -expansionary stages. μ is initialised finitely often, so there is an anchor that is never cancelled; let p be

this anchor. μ halts the stage if μ enumerates its follower into Q or sends a request token. By Lemma 3.9, this happens finitely often; hence μ halts the stage finitely often. So $\mu \infty$ is accessible infinitely often and since μ is initialised finitely often, $\mu \infty$ is initialised finitely often. \square

Lemma 3.15. Let ρ be a node working for requirement $P_{\Psi,k}$. Let s and $t > s$ be successive ρ -expansionary stages, and let $x \in \mathcal{C}_s(\rho)$. Suppose $z = \text{tr}_s(\rho, x) = \text{tr}_t(\rho, x)$ and let $u = \gamma_s(z)$. If $A_t \upharpoonright u \neq A_s \upharpoonright u$ or $W_t \upharpoonright u \neq W_s \upharpoonright u$, then $o_t^\rho(z) < o_s^\rho(z)$.

Proof. At stage $r \in (s, t)$, $\Gamma(A, W, z)[r] \uparrow$, then at the next τ -expansionary stage $r' \in (r, t]$ $\Gamma(A, W, z)[r'] = r'$. $\Gamma(A, W, z)[t] \geq \Gamma(A, W, z)[r']$ and $r' > \Gamma(A, W, z)[s]$, so $\Gamma(A, W, z)[t] \neq \Gamma(A, W, z)[s]$. s is a ρ -expansionary stage so $\Gamma(A, W, z)[s] = f_s^\rho(z)$, and t is also a ρ -expansionary stage so $\Gamma(A, W, z)[t] = f_t^\rho(z)$, then $f_s^\rho(z) \neq f_t^\rho(z)$, hence $o_t^\rho(z) < o_s^\rho(z)$. \square

Lemma 3.16. Let ρ be a node working for requirement $P_{\Psi,k}$, then for all x only finitely many attacks are started for (ρ, x) while a particular z is the tracker for x .

Proof. Fix x ; if z is the tracker of x for finitely many stages, then clearly only finitely many attacks are started for (ρ, x) while z is the tracker. So let z be the tracker for x for infinitely many stages. For an attack started at stage r , it is declared finished at stage s when we see $d_s^\rho(z) < d_r^\rho(z)$, so $o_s^\rho(z) < o_r^\rho(z)$. Note that (ρ, x) does not start another attack until there has been a ρ -expansionary stage after the previous attack was declared finished. Now if (ρ, x) starts infinitely many attacks while z is the tracker, then every one of these attacks started must be declared finished. Then we must see $o_s^\rho(z)$ decrease infinitely often, but this ordinal is from the ω^2 -computable approximation of f^ρ so this cannot happen. Hence for all x only finitely many attacks are started for (ρ, x) while a particular z is the tracker for x . \square

Lemma 3.17. Let ρ be a node working for requirement $P_{\Psi,k}$, and $x \in \mathcal{C}_s(\rho)$, then ρ enumerates finitely many numbers into A during each (ρ, x) attack.

Proof. Suppose not. Then there is a (ρ, x) attack such that ρ enumerates infinitely many numbers into A . Enumerations happen at ρ -expansionary stages, so by Lemma 3.15 we see the ordinal $o^\rho(z)$ decrease after each enumeration. Then if (ρ, x) enumerates infinitely many numbers into A we must see this ordinal decrease infinitely often, but this ordinal is from the ω^2 -computable approximation of f^k so this cannot happen. Hence ρ enumerates finitely many numbers into A during each (ρ, x) attack. \square

Lemma 3.18. Suppose τ is accessible infinitely often and initialised finitely often, then either $\tau \infty$ or $\tau \text{ fin}$ is accessible infinitely often and initialised finitely often.

Proof. Suppose there are finitely many τ -expansionary stages. τ halts the stage only at τ -expansionary stages; hence τ halts the stage finitely often, so $\tau \text{ fin}$ is accessible infinitely often. τ is initialised finitely often and $\tau \infty$ is accessible finitely often, so $\tau \text{ fin}$ is initialised finitely often.

Now suppose there are infinitely many τ -expansionary stages but finitely many $\tau \infty$ stages. τ halts the stage if it acts on a request or defines $\gamma^\rho(z)$ large during a (ρ, x) attack, where $z = \text{tr}(x)$ and ρ is a daughter of τ .

Let t be the last stage $\tau \infty$ was accessible. Since there are finitely many $\tau \infty$ stages, finitely many sons ζ_k are visited, and there are finitely many $\zeta_k \uparrow$ stages for each son ever visited. A request is sent to τ only at $\zeta_k \uparrow$ stages, but there are finitely many of these so τ deals with finitely many requests; hence τ acts on a request finitely often.

Then since τ halts the stage infinitely often, $\gamma^\rho(z)$ large infinitely often during a (ρ, x) attack. Now since there are only finitely many $\tau \infty$ stages, only finitely many daughters are ever visited; hence only finitely many x are ever established. By Lemma 3.16, each of these only start finitely many attacks. Then there is a particular (ρ, x) attack where $\gamma^\rho(z)$ is defined large infinitely often. Now there are no ρ -expansionary stages after stage t , so after stage t , $\gamma^\rho(z)$ is defined large because there is some $(\hat{\rho}, \hat{x}) \in \text{pro}(\rho, x)$ such that the computation $\Gamma_{\hat{\rho}}(A, W, \hat{z})$ has changed since the last τ -expansionary stage. Now $\text{pro}(\rho, x)$ is only redefined after an enumeration into A ; hence $\text{pro}(\rho, x)$

does not change after stage t . Then there is a particular $(\hat{\rho}, \hat{x}) \in \text{pro}(\rho, x)$ such that the computation $\Gamma_{\hat{\rho}}(A, W, \text{tr}(\hat{\rho}, x))$ has changed since the last τ -expansionary stage infinitely often. Now the tracker for \hat{x} can only change finitely often so there is some tracker \hat{z} such that there are infinitely many A or W changes below $\gamma^{\hat{\rho}}(\hat{z})$. By Lemma 3.15, every time there is an A or W change below $\gamma(\hat{z})$ the ordinal $o^{\hat{\rho}}(\hat{z})$ must decrease. Now $\hat{\rho} \infty \preceq \tau$, so $\Gamma_{\hat{\rho}}(A, W)$ is ω^2 -c.a. But $o^{\hat{\rho}}(\hat{z})$ decreases infinitely often, so this is a contradiction.

Therefore τ halts the stage finitely often, so there are infinitely many $\tau \infty$ stages. \square

Lemma 3.19. Suppose ρ is accessible infinitely often and initialised finitely often, then either $\rho \infty$ or $\rho \text{ fin}$ is accessible infinitely often and initialised finitely often.

Proof. Suppose there are finitely many ρ -expansionary stages. Then ρ only halts the stage at ρ -expansionary stages, so ρ halts the stage finitely often; hence $\rho \text{ fin}$ is accessible infinitely often. Since ρ is initialised finitely often and $\rho \infty$ is accessible finitely often, then $\rho \text{ fin}$ is initialised finitely often.

Now suppose there are infinitely many ρ -expansionary stages but there are only finitely many $\rho \infty$ stages. New x are established only at $\rho \infty$ stages, so only finitely many (ρ, x) are established. Let t be the last $\rho \infty$ stage. Only μ such that $\mu \succ \rho \infty$ can corrupt x ; hence no x is corrupted after stage s . So a new tracker is only appointed if an x is uncorrupted; hence the tracker for each x changes at most once after stage t . By Lemma 3.16, each established x starts finitely many attacks. It follows from Lemma 3.17 that each (ρ, x) attack puts finitely many numbers into A . There are only finitely many established x and finitely many attacks for each x ; hence ρ enumerates finitely many numbers into A . Therefore ρ halts the stage finitely often; thus there are infinitely many $\rho \infty$ stages. \square

Lemma 3.20. The true path is infinite.

Proof. Suppose there are only finitely many stages at which we define the collection of accessible nodes. Then since nodes are accessible finitely often, only finitely many (ρ, x) are ever established. After the last stage nodes are accessible no attack can be declared finished, so after this stage an attack can start for each (ρ, x) at most once. There are finitely many (ρ, x) so only finitely many attacks are ever started, but then the stage halts without defining the accessible nodes finitely often, a contradiction. So there are infinitely many stages at which we define the collection of accessible nodes. Then by Lemma 3.13, 3.14, 3.18 and 3.19, the true path is infinite. \square

Lemma 3.21. Let ρ and $\hat{\rho}$ be nodes working for requirements $P_{\Psi, k}$ and $P_{\hat{\Psi}, \hat{k}}$ respectively. Let $x \in \mathcal{C}_s(\rho)$ and $\hat{x} \in \mathcal{C}_s(\hat{\rho})$ with trackers z and \hat{z} respectively. Suppose (ρ, x) is in an attack at stage s . If $(\hat{\rho}, \hat{x})$ is in an attack at stage s and $\hat{\rho}$ is to the right of $\rho \infty$, then $\gamma_s^{\hat{\rho}}(\hat{z}) > \gamma_s^{\rho}(z)$.

Proof. Notice that every stage such that $\gamma^{\rho}(z)$ is defined large, $\hat{\rho}$ is initialised. Let t be the stage $\gamma^{\rho}(z)$ was last defined large. Then $\gamma_{t+1}^{\rho}(z) = \gamma_s^{\rho}(z)$. Now let r be the stage $(\hat{\rho}, \hat{x})$ was declared fully in an attack. $\hat{\rho}$ was initialised at stage t , so $r > t$; hence $\gamma_r^{\hat{\rho}}(\hat{z}) > \gamma_r^{\rho}(z) = \gamma_{t+1}^{\rho}(z)$. Now $\gamma_s^{\hat{\rho}}(\hat{z}) \geq \gamma_r^{\hat{\rho}}(\hat{z})$ and $\gamma_{t+1}^{\rho}(z) = \gamma_s^{\rho}(z)$; hence $\gamma_s^{\hat{\rho}}(\hat{z}) > \gamma_s^{\rho}(z)$. \square

Lemma 3.22. Suppose y is established for π at stage s . Suppose a (ρ, x) attack causes a number to enter A below $\varphi_t(y)$ at stage $t > s$. If π is not initialised then $\rho \succeq \pi \infty$.

Proof. Clearly if ρ is to the left of π then π gets initialised at stage t . Suppose $\rho \infty \preceq \pi$ or ρ is to the right of π . Let r be the stage (ρ, x) is declared fully in an attack. Notice that π is not accessible between stages r and t . If y was not established at stage r then it will not be established until after stage t ; hence $r > s$. $\varphi_t(y) \downarrow$ so it follows that $\varphi_r(y) \downarrow$ and $\varphi_t(y) = \varphi_r(y)$. At stage r , $\gamma_{r+1}(z)$ is defined with large use, so $\gamma_r(z) > \varphi_r(y)$. Since the use is non-decreasing it follows that $\gamma_t(z) > \varphi_t(y)$, a contradiction. Therefore if a (ρ, x) attack causes a number to enter A below $\varphi_t(y)$ at stage $t > s$ then $\rho \succeq \pi \infty$. \square

Lemma 3.23. Let π be a node working for requirement N_Φ and let ρ be a node working for requirement $P_{\Psi,k}$ with parent τ . Let z be the tracker for x . Suppose $\rho \succeq \pi \infty$. Let r be a stage $\gamma_{r+1}^\rho(z)$ is defined to be large. If $\Phi(A, y)[r] \downarrow$ then for all stages $s > r$ during this attack $A_s \upharpoonright \varphi_r(y) = A_r \upharpoonright \varphi_r(y)$.

Proof. Let $s + 1$ be the least counterexample. Then at stage s some $(\hat{\rho}, \hat{x})$ enumerated $\gamma_s^{\hat{\rho}}(\hat{z})$ into A and $\gamma_s^{\hat{\rho}}(\hat{z}) < \varphi_r(y)$. Notice that $(\hat{\rho}, \hat{x}) \neq (\rho, x)$ because $s + 1$ is the least counterexample so $\varphi_r(y) = \varphi_s(y)$ and $\gamma_{r+1}^\rho(z) > \varphi_r(y)$. Now at stage s , (ρ, x) is still in its attack; hence either $\hat{\rho}$ is to the right of $\rho \infty$ or $\hat{\rho} \infty \preceq \rho$. If $\hat{\rho} \infty \preceq \rho$ then by the assignment of requirements $\hat{\rho} \infty \preceq \tau$. Thus the $(\hat{\rho}, \hat{x})$ attack started after stage r . If $\hat{\rho}$ is to the right of $\rho \infty$ then $\hat{\rho}$ was initialised at stage r ; thus the $(\hat{\rho}, \hat{x})$ attack started after stage r .

Let r' be the stage $(\hat{\rho}, \hat{x})$ was declared fully in an attack. At stage r' we define $\gamma_{r'+1}^{\hat{\rho}}(\hat{z})$ to be large. Since $r' > r$, $\gamma_{r'+1}^{\hat{\rho}}(\hat{z}) > \varphi_r(y)$. Since $\gamma^{\hat{\rho}}(\hat{z})$ is non-decreasing, so it follows that $\gamma_s^{\hat{\rho}} > \varphi_r(y)$, a contradiction. \square

Lemma 3.24. Let π be a node working for requirement N_Φ and let ρ be a node working for requirement $P_{\Psi,k}$ with parent τ . Let z be the tracker for x and let s be the stage y was established. Suppose $\tau \succeq \pi \infty$. Suppose (ρ, x) started an attack or enumerated a number into A at stage $r > s$. Then for all stages $t > r$ during this attack, if $\Gamma_\rho(A, W, z)[t] \downarrow$ then $\gamma_t(z) > \varphi_t(y)$.

Proof. Let r' be the next τ -expansionary stage after stage r . Then at stage r' we define $\gamma_{r'+1}^\rho(z)$ is defined to be large. Now $\tau \succeq \pi \infty$ and y is established; hence $\Phi(A, y)[r'] \downarrow$. Then by Lemma 3.23, for all stages $t > r'$ during this attack $A_t \upharpoonright \varphi_{r'}(y) = A_{r'} \upharpoonright \varphi_{r'}(y)$. Then $\varphi_{r'}(y) = \varphi_t(y)$. Now $\gamma_{r'+1}^\rho(z) > \varphi_{r'}(y)$; hence $\gamma_{r'+1}^\rho(z) > \varphi_t(y)$. $\gamma^\rho(z)$ is non-decreasing so it follows that $\gamma_t^\rho(z) > \varphi_t(y)$. Now between stages r and r' , $\Gamma_\rho(A, W, z)[t] \uparrow$. Therefore we are done. \square

Lemma 3.25. Suppose (ρ, x) is in an attack at stage s . Let $z = \text{tr}_s(x)$ and

$o_s^\rho(z) = \omega \cdot d_s^\rho(z) + b_s^\rho(z)$. Then during this attack, (ρ, x) enumerates at most $b_s^\rho(z)$ numbers into A after stage s .

Proof. Enumeration into A happens at ρ -expansionary stages so by Lemma 3.15, each time we enumerate a number into A during this attack we will see the ordinal $o_t^\rho(z)$ decrease. Let t be the first ρ -expansionary stage after (ρ, x) has enumerated $b_s^\rho(z)$ numbers into A since stage s . Then we have seen the ordinal $o_s^\rho(z)$ decrease $b_s^\rho(z)$ many times. So $o_t^\rho(z) \leq o_s^\rho(z) - b_s^\rho(z) = \omega \cdot d_s^\rho(z) + b_s^\rho(z) - b_s^\rho(z) = \omega \cdot (d_s^\rho(z) - n) + b_t^\rho(z)$ for some $n > 0$. Hence $d_t^\rho(z) < d_s^\rho(z)$, so the attack is declared finished and (ρ, x) will not enumerate any more numbers into A for this attack. \square

Lemma 3.26. Let ρ and $\hat{\rho}$ be nodes working for requirements $P_{\Psi, k}$ and $P_{\hat{\Psi}, \hat{k}}$ respectively. Let $x \in \mathcal{C}_s(\rho)$ and $\hat{x} \in \mathcal{C}_s(\hat{\rho})$ with trackers z and \hat{z} respectively. Suppose x is corrupted, $(\hat{\rho}, \hat{x})$ is in an attack at stage s , and $\hat{\rho}$ is to the right of $\rho \infty$. Let r be the last ρ -expansionary stage. If $\gamma_s^{\hat{\rho}}(\hat{z}) < \gamma_s^\rho(z)$ then either x was corrupted at stage r or (ρ, x) was in an attack at stage r .

Proof. Let s be the least counterexample. Note that the last ρ -expansionary stage r is before the stage that the $(\hat{\rho}, \hat{x})$ attack started, and x is neither corrupted nor uncorrupted during the $(\hat{\rho}, \hat{x})$ attack as these actions would initialise $\hat{\rho}$. $\gamma_s^{\hat{\rho}}(\hat{z}) < \gamma_s^\rho(z)$, so by Lemma 3.21, (ρ, x) is not in an attack at stage s .

As in the argument for Lemma 3.21, if $\hat{\rho}$ was to the right of $\tau \infty$ then $\gamma_s^{\hat{\rho}}(\hat{z}) > \gamma_s^\rho(z)$; hence $\hat{\rho} \succ \tau \infty$. Let t be the stage the computation $\Gamma_{\hat{\rho}}(A, W, \hat{z})[s]$ was defined. Note that this means $(A, W)_t \upharpoonright \gamma_s^{\hat{\rho}}(\hat{z}) = (A, W)_s \upharpoonright \gamma_s^{\hat{\rho}}(\hat{z})$. Now if $\gamma_s^{\hat{\rho}}(\hat{z})$ was defined to be large then $\gamma_t^{\hat{\rho}}(\hat{z}) > \gamma_t^\rho(z)$ because $\hat{\tau} \succ \tau \infty$, but then it would follow that $\gamma_s^{\hat{\rho}}(\hat{z}) > \gamma_s^\rho(z)$; hence we did not define $\gamma_s^{\hat{\rho}}(\hat{z})$ to be large at stage t . Then it was defined to be $\gamma_{r_0}^{\hat{\rho}}(\hat{z})$ where r_0 is the last $\hat{\tau}$ -expansionary stage before stage t .

Suppose $(A, W)_{r_0} \upharpoonright \gamma_{r_0}^\rho(z) \neq (A, W)_s \upharpoonright \gamma_{r_0}^\rho(z)$. Let $r_1 \in [r_0, s)$ be the least stage such that $(A, W)_{r_0} \upharpoonright \gamma_{r_0}^\rho(z) \neq (A, W)_{r_1+1} \upharpoonright \gamma_{r_0}^\rho(z)$.

Suppose $W_{r_0} \upharpoonright \gamma_{r_0}^\rho(z) \neq W_{r_1+1} \upharpoonright \gamma_{r_0}^\rho(z)$. Now (ρ, x) does not start an attack at this stage because it would initialise $\hat{\rho}$. If there is a W change while (ρ, x) is corrupted but we do not start an attack, then either x was corrupted at the last ρ -expansionary stage or (ρ, x) was in an attack at the last ρ -expansionary stage. Stage r was the last ρ -expansionary stage so the Lemma holds.

Now suppose $W_{r_0} \upharpoonright \gamma_{r_0}^\rho(z) = W_{r_1+1} \upharpoonright \gamma_{r_0}^\rho(z)$. Then it must be that $A_{r_0} \upharpoonright \gamma_{r_0}^\rho(z) \neq A_{r_1+1} \upharpoonright \gamma_{r_0}^\rho(z)$. Then this was caused by a (ρ', x') attack enumerating a number into A at stage r_1 . Now this ρ' cannot be to the left of $\hat{\rho}$ as then $\hat{\rho}$ would be initialised at stage r_1 . But then ρ' is to the right of $\rho \infty$ and $r_1 < s$; hence the Lemma holds at stage r_1 and so either x was corrupted at stage r or (ρ, x) was in an attack at stage r , so we are done.

Now suppose $(A, W)_{r_0} \upharpoonright \gamma_{r_0}^\rho(z) = (A, W)_s \upharpoonright \gamma_{r_0}^\rho(z)$. If $\gamma_s^{\hat{\rho}}(\hat{z}) < \gamma_s^\rho(z)$ then $\gamma_{r_0}^{\hat{\rho}}(\hat{z}) < \gamma_{r_0}^\rho(z)$. s is the least counterexample so the Lemma holds at stage r_0 ; hence either x was corrupted at stage r or (ρ, x) was in an attack at stage r , so we are done. \square

Lemma 3.27. Let ρ and $\hat{\rho}$ be nodes working for requirements $P_{\Psi, k}$ and $P_{\hat{\Psi}, \hat{k}}$ respectively. Let $x \in \mathcal{C}_s(\rho)$ and $\hat{x} \in \mathcal{C}_s(\hat{\rho})$ with trackers z and \hat{z} respectively. Suppose x is corrupted and $\rho \infty \preceq \hat{\rho}$. If $(\rho, x) \notin \text{pro}_s(\hat{\rho}, \hat{x})$ and $(\hat{\rho}, \hat{x})$ enumerates $\gamma_s^{\hat{\rho}}(\hat{z})$ into A at stage s then x is uncorrupted.

Proof. $\rho \infty \preceq \hat{\rho}$ and $(\rho, x) \notin \text{pro}_s(\hat{\rho}, \hat{x})$ so x was established after the stage $\text{pro}_s(\hat{\rho}, \hat{x})$ was defined. x is corrupted so μ is to the right of $\hat{\rho} \infty$. Then μ is initialised at every stage $\gamma^{\hat{\rho}}(\hat{z})$ is defined large. Let t be the last stage $\gamma_{t+1}^{\hat{\rho}}(\hat{z})$ is defined large. Now μ is initialised at stage t ; hence q is appointed after stage t . $\mu \succ \rho \infty$ so x is invented after stage t . Let r be the first τ -expansionary stage after x is invented, $\gamma_r^\rho(\text{orig}(x))$ is defined large; hence $\gamma_r^\rho(\text{orig}(x)) > \gamma_{t+1}^{\hat{\rho}}(\hat{z})$. Now $\gamma_r^\rho(\text{orig}(x)) \leq \gamma_s^\rho(\text{orig}(x))$ and $\gamma_{t+1}^{\hat{\rho}}(\hat{z}) = \gamma_s^{\hat{\rho}}(\hat{z})$ so it follows that $\gamma_s^{\hat{\rho}}(\hat{z}) < \gamma_s^\rho(\text{orig}(x))$. Therefore x is uncorrupted. \square

Lemma 3.28. Let π be a node working for requirement N_Φ . Let r be the stage y is established. Let ρ be a node working for requirement $P_{\Psi, k}$. Suppose

$\tau \infty \preceq \pi \infty \preceq \rho$. Suppose either x was declared fully corrupted at stage $t > r$ or (ρ, x) was declared to have fully finished an attack at stage $t > r$. If (ρ, x) wants to attack at stage $s > t$ then $\Phi_\pi(A, y)[s] \downarrow$.

Proof. Let s be the least counterexample. Let $s_0 < s$ be the last $\pi \infty$ stage. Then since $\Phi_\pi(A, y)[s] \uparrow$, some $(\hat{\rho}, \hat{x})$ enumerated a number into A at stage $t_1 \in [s_0, s)$ such that $\gamma_{t_1}^{\hat{\rho}}(\hat{z}) < \varphi_{t_1}(y)$. Notice that by Lemma 3.22, $\hat{\rho} \succeq \pi \infty$; hence $t_1 = s_0$. $\tau \infty \preceq \pi \infty \preceq \rho$ so by the assignment of requirements, $\hat{\tau} \succeq \rho \infty$ or $\hat{\rho}$ is to the right of $\rho \infty$.

In the latter case, stage t is a ρ -expansionary stage so $\hat{\rho}$ is initialised at stage t . Let t_0 be the stage that $(\hat{\rho}, \hat{x})$ attack starts. Now \hat{x} was established after stage t ; hence \hat{x} was declared fully corrupted at stage $t' \in (r, t_0)$. By s is the least counterexample and $s > t_0$ so it follows that $\Phi_\pi(A, y)[t_0] \downarrow$. Then by Lemma 3.23, $\gamma_{t_1}^{\hat{\rho}}(\hat{z}) > \varphi_{t_1}(y)$, a contradiction.

Then $\hat{\tau} \succeq \rho \infty$. By Lemma 3.24, y was established after $(\hat{\rho}, \hat{x})$ was declared fully in an attack. Then the $(\hat{\rho}, \hat{x})$ was declared fully in an attack before stage t . Now suppose we are in the case where x was declared fully corrupted at stage x at stage t . Note that the stage $(\hat{\rho}, \hat{x})$ is declared fully in an attack is a ρ -expansionary stage; hence x was corrupted during the $(\hat{\rho}, \hat{x})$ attack. Then it follows that μ is to the right of $\hat{\rho}$. Therefore x was established after the last stage $\gamma^{\hat{\rho}}(\hat{z})$ was defined large; hence $(\rho, x) \notin \text{pro}_{t_1}(\hat{\rho}, \hat{x})$. Then by Lemma 3.27, x is uncorrupted at stage t_1 . Therefore (ρ, x) does not want to attack at stage s , a contradiction.

Now consider the case that (ρ, x) was declared to have fully finished an attack at stage t . Now notice that $\hat{\tau}$ is not accessible during the (ρ, x) attack; hence $\gamma^{\hat{\rho}}(\hat{z})$ was last defined large before the (ρ, x) attack starts. Let $t_2 + 1$ be the stage the (ρ, x) attack starts. Now consider the following cases:

Case 1. $(\rho, x) \in \text{pro}_{t_2}(\hat{\rho}, \hat{x})$. Then by Lemma 3.7, $\gamma_{t_2}^{\hat{\rho}}(\hat{z}) > \gamma_{t_2}^{\rho}(z)$. But at stage $t_2 + 1$ there is a W below $\gamma_t^{\rho}(z)$ prompting the start of an attack. But this change is also below $\gamma_{t_2}^{\hat{\rho}}(\hat{z})$ and causes the use to be lifted large at the next $\hat{\tau}$ -expansionary stage. But then the use $\gamma^{\hat{\rho}}(\hat{z})$ is lifted large after the (ρ, x) attack and hence also after y is established. Therefore $\gamma_{t_1}^{\hat{\rho}}(\hat{z}) > \varphi_{t_1}(y)$,

a contradiction.

Case 2. $(\rho, x) \notin \text{pro}_{t_1}(\hat{\rho}, \hat{x})$. Then by Lemma 3.27, x is uncorrupted at stage t_1 . Therefore (ρ, x) does not want to attack at stage s , a contradiction. \square

Lemma 3.29. Every N_Φ requirement is met.

Proof. Fix Φ . The true path is infinite and every infinite path on the tree of strategies has a node working for requirement N_Φ ; let π be the node on the true path working for N_Φ . If $\Phi(A)$ is not total then we are done, so suppose $\Phi(A)$ is total. Then we need to show $\Phi(A)$ that is ω -c.a. by defining an ω -computable approximation. At each $\pi \infty$ stage we establish a new input y by giving it an ordinal as follows. Let s first $\pi \infty$ stage after y is established, then define

$$o_s^\Phi(y) = n + \sum_{(\rho, x) \in \mathcal{B}(y)} b_s^\rho(x)$$

where: $\mathcal{B}(y)$ is the set of (ρ, x) such that $x \in \mathcal{C}_s(\rho)$ is fully corrupted and $\tau \infty \preceq \pi \infty \preceq \rho$; n is the number of ρ such that (ρ, x) is in an attack at stage s , $\Gamma_\rho(A, W, \text{tr}(x))[s] \downarrow$ and the parent of ρ , τ , is such that $\tau \succ \pi \infty$.

By Lemma 3.24, the only (ρ, x) with $\tau \succ \pi \infty$ that can injure $\Phi(A, y)$ are in an attack at stage s and $\Gamma_\rho(A, W, \text{tr}(x))[s] \downarrow$. Also by Lemma 3.24, such ρ can only injure $\Phi(A, y)$ at most once. At stage s there are n such (ρ, x) in an attack and each of these can only injure $\Phi(A, y)$ at most once; hence $\Phi(A, y)$ can be injured by such enumerations at most n many times.

By Lemma 3.22, the only (ρ, x) that can injure $\Phi(A, y)$ extend $\pi \infty$. We have just dealt with the case where $\tau \succ \pi \infty$, so all that is left is the case where $\tau \infty \preceq \pi$. Now suppose at stage $t > s$, x is declared fully corrupted or (ρ, x) is declared to have fully finished an attack. By Lemma 3.28, if (ρ, x) wants to attack at a stage $t' > t$, then $\Phi(A, y)[t'] \downarrow$; then by Lemma 3.23, it follows that for all stages r during this attack $\gamma_r^\rho(\text{tr}_r(x)) > \varphi_r(y)$.

Then the only (ρ, x) with $\tau \infty \preceq \pi \infty \preceq \rho \infty$ that can injure $\Phi(A, y)$ at stage t are such that x was declared fully corrupted at a stage $r < s$; hence

they are in the set $\mathcal{B}(y)$. After this attack is declared finished, by Lemma 3.28, for all stages r during any attack started after this stage $\gamma_r^\rho(\text{tr}_r(x)) > \varphi_r(y)$. Therefore, by Lemma 3.25, each $(\rho, x) \in \mathcal{B}(y)$ can injure $\Phi(A, y)$ at most $b_s^\rho(z)$ times.

So the ordinal defined above is the maximum possible number of enumerations into A below $\phi(y)$. Therefore this is the maximum number of changes to the computation $\Phi(A, y)$. Hence we have an ω -computable approximation for $\Phi(A)$ as desired. \square

Lemma 3.30. Let η be a node working for requirement R_Θ . If $\Theta(A, W)$ is total and η is on the true path then $\Delta_\eta(A, W)$ is total.

Proof. $\Theta(A, W)$ is total so there are infinitely many η -expansionary stages. At an η -expansionary we leave $\Delta_\Theta(A, W, p)$ undefined only if p is an anchor for some μ and we appoint a new follower, but by Lemma 3.12 for each p this happens finitely often. So $\limsup_s \Delta_\Theta(A, W)[s]$ goes to infinity. Now we need to check that the use is bounded for all p . Suppose p is eventually not an anchor for any daughter of η . Then $\Delta_\Theta(A, W, p)$ is eventually defined with use 0; hence the use of such p is bounded. So now suppose p is an anchor for μ , a daughter of η , and $p = \text{ac}_s(\mu)$ for infinitely many stages. Now by Lemma 3.12, there are only finitely many followers appointed while p is the anchor. Let q be the last follower appointed and let t be the stage q is appointed. Then if $\Delta_\Theta(A, W, p)[s]$ is defined at stage $s > t$, its use is defined to be $\theta_s(q)$. $\theta_s(q)$ has bounded use because $\Theta(A, W)$ is total. Therefore $\Delta_\Theta(A, W, p)$ has bounded use for all p ; hence $\Delta_\Theta(A, W)$ is total. \square

Lemma 3.31. Let τ be a node working for requirement P_Ψ . If $\Psi(A, W, Q)$ is total and τ is on the true path then $\Gamma_\Psi(A, W)$ is total.

Proof. Since $\Psi(A, W, Q)$ is total, there are infinitely many τ -expansionary stages. At each τ -expansionary stage we defined a longer initial segment of $\Gamma_\Psi(A, W)$, so $\limsup_s \text{dom } \Gamma_\Psi(A, W,) [s]$ goes to infinity. If z is eventually not a tracker of any x for any daughter of τ then $\Gamma_\Psi(A, W, z)$ is defined with

use 0, so the use of such z is bounded. Now consider z which is the tracker for some x for some daughter of τ , ρ , for infinitely many stages. Suppose there is a stage t such that for all $s > t$, (ρ, x) is not in an attack at stage s . A computation $\Gamma_\Psi(A, W, z)$ defined at stage $s > t$ is defined with use $\psi_s^\rho(I_s(\rho, x))$. Since z is never cancelled, ρ is initialised finitely often, so by Lemma 3.11 the interval $I(x)$ must stabilise. $\Psi(A, W, Q)$ is total so the use $\psi_s(x')$ for every $x' \in I(x)$ is bounded; hence the use of $\Gamma_\Psi(A, W, z)$ is bounded for such z . So now consider that there is no such stage t , then by Lemma 3.16 it must be that there is some attack that is never finished. During an attack a number is enumerated into A at every ρ -expansionary stage, then by Lemma 3.17 there are finitely many ρ -expansionary stages. So let t be the last ρ -expansionary stage. Now $\mathbf{pro}_s(\rho, x)$ is only redefined after an A enumeration by (ρ, x) , but there are no ρ -expansionary stages after stage t ; hence for all $s, r > t$, $\mathbf{pro}_s(\rho, x) = \mathbf{pro}_r(\rho, x)$. Now for $(\hat{\rho}, \hat{x}) \in \mathbf{pro}_s(\rho, x)$, $\hat{\rho} \infty \prec \tau$, so $\hat{\rho} \infty$ is also initialised finitely often. Then the tracker for $(\hat{\rho}, \hat{x})$ is only change finitely often; let \hat{z} be the last tracker. $\hat{\rho} \infty$ is also accessible infinitely often; hence for all $(\hat{\rho}, \hat{x}) \in \mathbf{pro}_s(\rho, x)$, $\Gamma_{\hat{\rho}}(A, W)$ is ω^2 -c.a., so the computation $\Gamma_{\hat{\rho}}(A, W, \mathbf{tr}_s(\hat{\rho}, \hat{x}))$ changes finitely often. For each $(\hat{\rho}, \hat{x}) \in \mathbf{pro}_s(\rho, x)$, \hat{z} is the last tracker so \hat{x} is never uncorrupted while \hat{z} is the tracker. Then the use is defined to be large due to the uncorruption of some protected \hat{x} finitely often. Then it follows that there is a stage r such that at every stage $s > r$ where we define the computation $\Gamma_\Psi(A, W, z)[s]$, we define it with use $\gamma_r(z)$. Hence the use of $\Gamma_\Psi(A, W, z)$ is bounded for all z , and therefore $\Gamma_\Psi(A, W)$ is total. \square

Remark. If there is no node on the true path that works for requirement P_Ψ , then there is some node α working for requirement P_Ψ or R_Θ on the true path such that every son and daughter of α has been assigned a node on the true path. Recall that we say a parent is been closed below the infinite outcome of a daughter or the divergent outcome of a son. Also recall that we stop placing children on the tree once the parent has been closed. Then if α has infinitely many sons and daughters on the true path then the finite

outcome of every daughter of α and the convergent outcome of every son of α is on the true path. Without loss of generality, let α be a node working for requirement P_Ψ . Then it follows that $\Psi(A, W, Q)$ is total. Then by Lemma 3.31, $\Gamma_\Psi(A, W)$ is total. Since the finite outcome of every daughter of α is on the true path, $\Gamma_\Psi(A, W)$ is not ω^2 -c.a.; then $A \oplus W$ is not totally ω^2 -c.a. This means that for all functionals Ψ and Θ , requirements R_Θ and P_Ψ are met immediately, so we are done.

So suppose $A \oplus W$ is totally ω^2 -c.a. Then every parent node is eventually closed; hence there is a node on the true path working for every P_Ψ and R_Θ requirement. The following lemmas will use this assumption.

Lemma 3.32. Every R_Θ requirement is met.

Proof. Fix Θ . $A \oplus W$ is totally ω^2 -c.a. and the true path is infinite, so there is a node η on the true path working for R_Θ . If $\Theta(A, W)$ is not total then we are done. So suppose $\Theta(A, W)$ is total. Then $\eta \infty$ is on the true path, and by Lemma 3.30 $\Delta_\Theta(A, W)$ is total. Since $A \oplus W$ is totally ω^2 -c.a. and $\Delta_\Theta(A, W)$ is total, $\Delta_\Theta(A, W)$ is ω^2 -c.a. Then since the true path is infinite there must be a daughter of η , μ , such that $\mu \infty$ is on the true path. Then μ is initialised finitely often; hence it has finitely many anchors, so let p be the last anchor. By Lemma 3.12 only finitely many followers are appointed, so let q be the last follower appointed.

If $\Theta(A, W, q) = 0$, then there is a μ -expansionary stage s such that $\Theta(A, W, q)[s] = 0$ after q is appointed. q is the last follower, so there is a stage where μ will either enumerate q into Q_{s+1} or send a request. If this request was cancelled at stage r then it is cancelled because $\Delta_\eta(A, W, p)[r] \uparrow$. Let t be the next η -expansionary stage after the request was cancelled. If there was some $(\rho, x) \in \text{pro}_t(\mu)$ such that $q < \psi_t^\rho(I_t(x))$ then a new follower is appointed, but q is the last follower so this does not happen. Then another request to enumerate q into Q will be sent. $\Delta_\Theta(A, W)$ is total so we will only cancel the request finitely many times. So there is a request sent that is never cancelled. Then μ is not accessible until the request is carried

out. There are infinitely many μ -expansionary stages; hence the request is eventually carried out, so $q \in Q$.

$\Theta(A, W, q)[s] = 0$ prompting the enumeration of q into Q . Then since $\Theta(A, W, q) = 1$, there is some stage $t > s$ such that either $A_s \upharpoonright \theta_s(q) \neq A_t \upharpoonright \theta_s(q)$ or $W_s \upharpoonright \theta_s(q) \neq W_t \upharpoonright \theta_s(q)$. By Lemma 3.1, $\delta_s(p) \geq \theta_s(q)$, so it is also the case that either $A_s \upharpoonright \delta_s(p) \neq A_t \upharpoonright \delta_s(p)$ or $W_s \upharpoonright \delta_s(p) \neq W_t \upharpoonright \delta_s(p)$. So there is an η -expansionary stage after stage $r > s$ where $\Delta_\Theta(A, W, p)[r] \uparrow$ and $q \in Q_r$; this prompts the appointment of a new follower, but this contradicts that q is the last follower appointed. Therefore if $\Theta(A, W, q) = 1$ then $q \notin Q$.

So we have shown that diagonalisation is successful, $\Theta(A, W) \neq Q$; hence requirement R_Θ is met. \square

Lemma 3.33. Every P_Ψ requirement is met.

Proof. Fix Ψ . $A \oplus W$ is totally ω^2 -c.a. and the true path is infinite, so there is a node τ on the true path working for R_Ψ . If $\Psi(A, W, Q)$ is not total then we are done, so suppose $\Psi(A, W, Q)$ is total. Then by Lemma 3.31, $\Gamma_\Psi(A, W)$ is total. $A \oplus W$ is totally ω^2 -c.a. so there is a daughter of τ , ρ , such that $\rho \infty$ is on the true path. Since $\rho \infty$ is on the true path, every x is eventually established, so now we show that there is an ω^2 -computable approximation for $\Psi(A, W, Q)$.

Now we define $\alpha_t(x) = \omega \cdot m_t(x) + k_t(x)$. ρ is initialised finitely often so consider the stages after the last initialisation. Now let s be the first ρ -expansionary stage after x is established, and let $z_0 = \text{tr}_s(\rho, x) = \text{orig}(\rho, x)$. Then define $m_s(x) = d_s^\rho(z_0) + 2$ and $k_s(x) = b_s^\rho(z_0)$. For stages r where x is not corrupted define $m_{r+1}(x) = m_r(x)$ and $k_{r+1}(x) = b_r^\rho(z_0)$.

Suppose x is corrupted. Let t be the first ρ -expansionary stage after corruption and let z_1 be the new tracker. Then define $m_{t+1}(x) = m_t(x) - 1$ and $k_{t+1}(x) = d_t^\rho(z_1)$.

If x is uncorrupted. Let t be the first ρ -expansionary stage after uncorruption, and define $m_t(x) = m_{t-1}(x) - 1$ and $k_t(x) = b_t^\rho(z_0)$.

For stages r while x is corrupted define $m_{r+1}(x) = m_r(x)$ and $k_{r+1}(x) = d_r^\rho(z_1)$.

Now define

$$o_s^\Psi(x) = \sum_{y \leq x} \alpha_s(y)$$

Suppose x has not been taken over. While x uses its original tracker it just follows the ordinal of its tracker. By Lemma 3.4, if a number enters Q below the use $\psi(I(x))$ then x is either taken over or declared corrupted and by Lemma 3.5 x is taken over if another number enters Q below $\psi(I(x))$. Therefore while x uses its original tracker there are no numbers entering Q ; hence this ordinal works while x is not corrupted. Now if x gets corrupted or uncorrupted then we need to decrease $m(x)$ by one. By Lemma 3.5 we will only need to do this a maximum of two times hence $m_s(x) = d_s^\rho(z_0) + 2$ gives us enough room to count for these changes.

Now suppose x is corrupted. Note that we declare a new ordinal at the next ρ -expansionary stage, so if there are multiple changes between ρ -expansionary stages we only need to account for one of these changes. Also note that we do not need to worry about any changes during a (ρ, x) attack because wait until the first ρ -expansionary stage after the attack is finished before declaring a new ordinal.

If there is an A change below $\psi(I(x))$ at some stage t while (ρ, x) is not in an attack then by Lemma 3.3 (4) and Lemma 3.7 this must come from the attack of some $(\hat{\rho}, \hat{x})$ such that $(\rho, x) \notin \text{pro}_t(\hat{\rho}, \hat{x})$. Now it is the case that either $\hat{\rho} \succ \rho \infty$ or $\hat{\rho}$ is to the right of $\rho \infty$. In the latter case, it follows from Lemma 3.26 that either x was corrupted at stage r or (ρ, x) was in an attack at stage r , where r is the last ρ -expansionary stage before the computation $\Gamma_\rho(A, W, \text{tr}(x))[s]$ was defined. Then we do not need to declare a new ordinal until the next ρ -expansionary stage; hence we can charge the decrease to the case that holds. In the case that $\hat{\rho} \succ \rho \infty$ then since $(\rho, x) \notin \text{pro}_t(\hat{\rho}, \hat{x})$, by Lemma 3.27, when $(\hat{\rho}, \hat{x})$ enumerates a number into A at stage t , x is uncorrupted.

By Lemma 3.5, if there is a Q change below $\psi(I(x))$ while x is corrupted then x is taken over by some $x' < x$.

If there is a W change below $\psi(I(x))$, then we start an attack for (ρ, x)

at some stage t . Since there are infinitely many ρ -expansionary stages this attack eventually finishes. Then there is a stage r such that $d_r^\rho(z_1) < d_t^\rho(z_1)$, and since $k_t(x) = d_t^\rho(z_1)$, we have $k_r(x) < k_t(x)$; hence we see a decrease in $\alpha(x)$.

Suppose x has been taken over by some x' . Now if there is a change below $\psi(x)$ then there is a change below $\psi(I(x'))$. Therefore by the above argument, we see a decrease in $\alpha(x')$. x can only be taken over by $x' < x$ and $\alpha(x')$ has been included in $o^\psi(x)$ for $x' < x$; hence if x is taken over and there is a change below $\psi(x)$ then there is a decrease in $o^\psi(x)$ as required.

Therefore this definition of o^ψ gives us an ω^2 -computable approximation for $\Psi(A, W, Q)$. □

Chapter 4

Extending Results

In the previous chapter we showed that for a given c.e. set W we can uniformly find a totally ω -c.a. c.e. set A such that $A \oplus W$ is not maximal totally ω^2 -c.a. Now in this chapter we will discuss the modifications to the construction from the previous chapter that are required to show that for given c.e. sets W_0 and W_1 we can uniformly find a totally ω -c.a. c.e. set A such that both $A \oplus W_0$ and $A \oplus W_1$ are not maximal totally ω^2 -c.a. From this it follows that there is no computable g such that for all m , if W_m is totally ω -c.a. then either $W_m \oplus W_{g(m,0)}$ or $W_m \oplus W_{g(m,1)}$ is maximal totally ω^2 -c.a. In the second section of this chapter we will discuss the problems that arise when we try to extend to three c.e. sets.

Note that for this section we will refer to the fixed c.e. sets as W_i and refer to the c.e. set that ρ and $\hat{\rho}$ work for as W and \hat{W} respectively. So W and \hat{W} range over the fixed c.e. sets W_i . It will be specified whether $W = \hat{W}$ or $W \neq \hat{W}$ during the discussion.

4.1 Two c.e. sets

First recall the structure of the tree for a single W ; we were able to ensure that there were no parent nodes between any given parent and a child of this parent. Recall that we were able to do this because if $A \oplus W$ is not

totally ω^2 -c.a. then we are done with all P and R requirements; hence we are able to wait for a node that guesses that $\Gamma(A, W)$ is ω^2 (or similarly for $\Theta(A, W)$) before working on the next requirement. But now notice that while it remains true that if $A \oplus W$ is not totally ω^2 -c.a. then we are done with all $P_{\Psi, W}$ and $R_{\Theta, W}$ requirements, we still need to meet $P_{\Psi, \hat{W}}$ and $R_{\Theta, \hat{W}}$ requirements, where $W \neq \hat{W}$. Hence we must continue placing P_{W_1} and R_{W_1} requirements even while we have not yet closed a parent working for a P_{W_0} or R_{W_0} requirement. Hence between a parent and child for some P_{W_0} or R_{W_0} requirement it is possible that there is a parent working for a P_{W_1} or R_{W_1} requirement. But notice that we are still able to maintain this property for parents and children working with the same W .

Let τ and $\hat{\tau}$ be nodes working for requirements $P_{\Psi, W}$ and $P_{\Psi, \hat{W}}$ respectively. Let ρ and $\hat{\rho}$ be children of τ and $\hat{\tau}$ respectively. Now in the case where $W = \hat{W}$ it remains true that if $\tau \infty \preceq \hat{\rho} \infty$ then $\rho \infty \preceq \hat{\tau} \infty$. But now if $W \neq \hat{W}$ then the configuration $\tau \infty \preceq \hat{\tau} \infty \preceq \hat{\rho} \infty \preceq \rho$ is possible. Note that the configuration $\tau \infty \preceq \hat{\tau} \infty \preceq \rho \infty \preceq \hat{\rho}$ is not possible because we are able to restart $\hat{\tau}$ below $\rho \infty$. So consider we have the case where $\tau \infty \preceq \hat{\tau} \infty \preceq \hat{\rho} \infty \preceq \rho$. Now at some stage (ρ, x) could start an attack and so at the next τ -expansionary stage the use will be lifted large. Before the stage $\gamma^\rho(z)$ is lifted large it is possible that there is a \hat{W} change prompting a $(\hat{\rho}, \hat{x})$ attack. Notice that because $\tau \infty \preceq \hat{\tau}$ when we lift $\gamma^\rho(z)$ large $\Psi_{\hat{\tau}}(A, \hat{W}, Q_{\hat{W}}, \hat{x}) \uparrow$, hence the (ρ, x) attack could injure this computation. Note that $\hat{\rho} \infty \preceq \rho$ so we are not able to wait until the (ρ, x) attack is finished before certifying the new $\Psi_{\hat{\tau}}(A, \hat{W}, Q_{\hat{W}}, \hat{x})$ computation. Now consider the following cases:

Case 1: x was fully corrupted after \hat{x} was fully corrupted. Now the length of the (ρ, x) attack, $b^\rho(z)$, was revealed after \hat{x} was fully corrupted. So $(\hat{\rho}, \hat{x})$ cannot deal with injury from any (ρ, x) attacks. Now notice that we can see $k^{\hat{\rho}}(\hat{x})$ at the stage x is fully corrupted and this number represents the maximum number of attacks $(\hat{\rho}, \hat{x})$ could carry out. So we are able to incorporate this number into our new value for $k^\rho(x)$. Now if there is a \hat{W}

change prompting the start of a $(\hat{\rho}, \hat{x})$ attack during the (ρ, x) attack then we are at risk of the (ρ, x) attack injuring $(\hat{\rho}, \hat{x})$ which we cannot allow, so we *abort* the (ρ, x) attack. Now we are able to do this because we incorporated $k^{\hat{\rho}}(\hat{x})$ into $k^{\rho}(x)$ so although we did not see $d^{\rho}(z)$ decrease we are able to charge the decrease to the $(\hat{\rho}, \hat{x})$ attack. We can do this because $\hat{\rho} \infty \preceq \rho$ so the $(\hat{\rho}, \hat{x})$ attack will be finished at the next ρ -expansionary stage. Then at the next ρ -expansionary stage $k^{\hat{\rho}}(\hat{x})$ will have decreased by one, so this gives us the required decrease in $k^{\rho}(x)$.

Case 2: x was fully corrupted before \hat{x} was fully corrupted and x was established after \hat{x} was established. When we establish x we can see \hat{x} because it has already been established, then we are able take \hat{x} into account when we define $m^{\rho}(x)$. Then when \hat{x} is corrupted we can decrease $m^{\rho}(x)$ by one. Now at the next ρ -expansionary stage \hat{x} is fully corrupted; hence we are essentially back in the first case and are able to incorporate $k^{\hat{\rho}}(\hat{x})$ into $k^{\rho}(x)$. So in this case we can allow $(\hat{\rho}, \hat{x})$ attacks to abort (ρ, x) attacks.

Case 3: x was fully corrupted before \hat{x} was fully corrupted and x was established before \hat{x} was established. In this case notice that due to the relative strength of the corrupting numbers it must be that \hat{x} was actually established after x was fully corrupted. Recall that this is because μ initialises all weaker nodes when a new follower is appointed. Then this means when \hat{x} is established it has access to an upper bound for how many attacks (ρ, x) can do. So \hat{x} can incorporate this bound into $m^{\hat{\rho}}(\hat{x})$. Then every time (ρ, x) attack we are able to decrease $m^{\hat{\rho}}(\hat{x})$ by one and incorporate the length of attack into $k^{\hat{\rho}}(\hat{x})$; this allows $(\hat{\rho}, \hat{x})$ to actually be able to take the injury from any (ρ, x) attack. However notice that when a (ρ, x) attack starts it may not finish; it may be aborted by some (ρ', x') . Then after it is aborted it may start another attack but it still has the bound on the length of the attack as the first attack because it did not finish, so it is possible $d^{\rho}(z)$ has not decreased by one. So in this case we do not need to decrease $m^{\hat{\rho}}(\hat{x})$ again; hence what we actually need to do is decrease $m^{\hat{\rho}}(\hat{x})$ by one every time a new bound for the length of attack, $b^{\rho}(z)$, is revealed. Notice that every

time this happens we will have seen $d^\rho(z)$. Therefore all that is required is to incorporate $d^\rho(z)$ into $m^{\hat{\rho}}(\hat{x})$ and decrease $m^{\hat{\rho}}(\hat{x})$ each time we see $d^\rho(z)$ decrease. This allows us to always be able to take the injury from the next (ρ, x) attack.

As discussed, a (ρ, x) attack will abort a $(\hat{\rho}, \hat{x})$ attack if $\hat{\rho} \succeq \rho \infty$, $W \neq \hat{W}$ and either x was established before \hat{x} was established or x was fully corrupted before \hat{x} was fully corrupted (note that this is equivalent to the condition that x was established before \hat{x} was fully corrupted; this is due to the relative strength of the corrupting numbers). So notice that we do not abort attacks that work with the same W ; this is because we know we do not need to as shown in the construction in Chapter 3. This does cause careful treatment when it comes to defining the ordinal approximation for P_Ψ requirements, but does not affect the construction.

Now an important thing to note is that although it is possible to get the configuration $\eta \infty \preceq \tau \infty \preceq \rho \infty \preceq \mu$ this is only possible if η and τ work for different W . This means that μ enumerates numbers into Q_W while τ works with oracle $Q_{\hat{W}}$ ($W \neq \hat{W}$); hence enumerations by such a μ do not affect τ and ρ . Therefore it is still the case that a μ node can only cause injury to ρ such that $\rho \infty \preceq \eta$. Note that we still require requests to take care of injury from the right. Since R and P requirements only interact if they work for the same W , these interactions work the exact same way as in the previous construction and require no modification.

But now we will notice that two W affects the interaction between N and P requirements. Let ρ and $\hat{\rho}$ be nodes working for requirements $P_{\Psi, W_0, k}$ and $P_{\Psi, W_1, \hat{k}}$ respectively. Now suppose we have the configuration:

$$\tau \infty \preceq \pi \infty \preceq \rho \infty \preceq \hat{\tau} \infty \preceq \hat{\rho}$$

Suppose $(\hat{\rho}, \hat{x})$ is in an attack and $\Gamma_{\hat{\rho}}(A, W, \hat{z}) \downarrow$ at the stage y was established; hence it is possible that $\gamma^{\hat{\rho}}(\hat{z}) < \varphi(y)$. Then at the next $\hat{\rho}$ -expansionary stage, the enumeration into A injures $\Phi_\pi(A, y)$. Then before the next $\pi \infty$ stage it is possible that there is a W change prompting the start of a (ρ, x) attack. If

(ρ, x) starts this attack then since $\tau \infty \preceq \pi$, the use $\gamma^\rho(z)$ will be lifted large before we see $\Phi(A, y)$ recover. So now (ρ, x) could injure the computation $\Phi_\pi(A, y)$. Notice that this is not necessarily a problem; as long as y could see the length of this attack when it was established, it is ok for this attack to injure it. However if (ρ, x) was fully corrupted or finished an attack after y was established, then we cannot allow (ρ, x) to injure the computation $\Phi_\pi(A, y)$. Notice this is a significant difference from the construction we had in the previous chapter, because we were able to argue that if (ρ, x) was fully corrupted or finished an attack after y was established, then at any later stage where (ρ, x) wants to start an attack, $\Phi_\pi(A, y) \downarrow$ (shown in Lemma 3.28)).

To protect (π, y) computations we are not allowed to injure we define a *guarded* set, $\mathbf{grd}(\rho, x)$. We define this when x is fully corrupted and then redefine it every time (ρ, x) finishes an attack. Then if there is a W change prompting (ρ, x) to start an attack but there is some $(\pi, y) \in \mathbf{grd}(\rho, x)$ such that $\Phi_\pi(A, y) \uparrow$, we do not let (ρ, x) start an attack. But now how do we count this? We still need to decrease the ordinal $o^\Psi(x)$ because there has been a W change; since we have not started an attack we need to count this in a different way. So clearly we need some way to bound how many times we may need to *prevent* a (ρ, x) attack due to $\Phi_\pi(A, y) \uparrow$ for some *guarded* (π, y) . To do this we investigate what could have caused this to happen.

Let $(\pi, y) \in \mathbf{grd}_s(\rho, x)$ and suppose $\Phi_\pi(A, y)[s] \uparrow$. Then this was caused by an enumeration into A at stage t ; let $(\hat{\rho}, \hat{x})$ be responsible for this enumeration. Now if $\hat{\tau} \succeq \rho \infty$ then $\hat{\tau} \succeq \pi \infty$, hence the number $(\hat{\rho}, \hat{x})$ enumerated into A at stage t was defined before y was established and $(\hat{\rho}, \hat{x})$ has been in an attack since then. Suppose $\mathbf{grd}_s(\rho, x)$ was defined due to the corruption of x . Then μ must be to the right of $\hat{\rho}$ and hence was last initialised when $(\hat{\rho}, \hat{x})$ lifted the use $\gamma^{\hat{\rho}}(\hat{z})$ large. Therefore the number enumerated into A at stage t is small enough to uncorrupt x . So in this case we are fine. Now suppose $\mathbf{grd}_s(\rho, x)$ was defined due to finishing an attack. Then since $(\hat{\rho}, \hat{x})$ was in an attack during a (ρ, x) attacks, (ρ, x) attacks do not abort $(\hat{\rho}, \hat{x})$ attack;

hence \hat{x} was fully corrupted before x was established. Now notice that this means that x can see the bound for how many attacks $(\hat{\rho}, \hat{x})$ can do, then as discussed above this allows us to incorporate $k^{\hat{\rho}}(\hat{x})$ into $m^{\rho}(x)$; hence every time $(\hat{\rho}, \hat{x})$ starts an attack we can incorporate $b^{\hat{\rho}}(\hat{x})$ into $k^{\rho}(x)$. Then we are able to count the prevented attack against this single enumeration by $(\hat{\rho}, \hat{x})$.

Now we need to consider the case where $\hat{\tau} \infty \preceq \rho$. Now recall that to injure $\Phi_{\pi}(A, y)$ it must be that $\hat{\rho} \succeq \pi \infty$ (shown in Lemma 3.22). In this case notice that the argument does become a bit more complicated because it could be that the $(\hat{\rho}, \hat{x})$ attack actually started after y was established. So suppose $(\hat{\rho}, \hat{x})$ is not in an attack at the stage y was established. Then an enumeration by some (ρ', x') caused $(\hat{\rho}, \hat{x})$ to begin injuring $\Phi(A, y)$. Now the key is that since there are only two W , then it must be that $\tau' \succeq \rho \infty$ and so $\tau' \succeq \pi \infty$; hence (ρ', x') was in an attack when y was established. Moreover ρ' must work with the same W as ρ . This means (ρ', x') cannot injure $\Phi_{\pi}(A, y)$ when $(\pi, y) \in \mathbf{grd}(\rho, x)$ because otherwise we would contradict Lemma 3.28. Therefore, if (ρ', x') causes $(\hat{\rho}, \hat{x})$ to start injuring $\Phi_{\pi}(A, y)$, then this happens before $\mathbf{grd}_s(\rho, x)$ is defined. Then we are able to use the argument above to show that \hat{x} was corrupted before x was established and hence we are able to count for the attack we must prevent.

We believe that a construction with the following modifications to the construction detail in Chapter 3 will work:

Tree of strategies. The idea for the the tree of strategies is to take two versions of the tree from the previous chapter, one working for W_0 and one for W_1 , and then interweave them together. However then to avoid the configuration $\tau \infty \preceq \hat{\tau} \infty \preceq \rho \infty \preceq \hat{\rho}$ we will also need to *restart nodes*. So below the infinite/divergent outcome of child nodes we can restart all parent nodes between it and its parent.

Guarding Set. Define $\mathbf{grd}(\rho, x)$ to be the collection of (π, y) such that $\pi \infty \preceq \rho$ and y has been established. This is first defined when x is fully corrupted and then redefined each time a new $b^{\rho}(z)$ is revealed (each time we see $d^{\rho}(z)$ decrease). Then when there is a W change prompting an attack

we first check whether there is some $(\pi, y) \in \text{grd}(\rho, x)$ such that $\Phi_\pi(A, y) \uparrow$; if so then we do not start an attack and say this attack was *prevented* and otherwise we start an attack.

Aborting attacks. When (ρ, x) starts an attack, we abort any $(\hat{\rho}, \hat{x})$ such that $W \neq \hat{W}$, $\hat{\rho} \succeq \rho \infty$ and either x was established before \hat{x} was established or x was fully corrupted before \hat{x} was fully corrupted.

Now the guarding set ensures N requirements can be met with the same ordinal definition as in the previous construction (Lemma 3.29). Aborting attacks ensures each (ρ, x) can take any A injuries that cannot be avoided. Then as discussed we are able to count for attacks that are prevented and aborted; hence we are able to meet P requirements. Note that the ordinal definition for P requirements will be much more complicated than in the previous construction.

4.2 Three c.e. sets

Now we ask: what happens when we consider three c.e. sets, W_0 , W_1 and W_2 ? Consider the discussion in the previous section about how we are able to prevent (ρ, x) attacks from starting if the computation of a guarded (π, y) is undefined. The key difference is that now it is possible to have the configuration:

$$\tau' \infty \preceq \hat{\tau} \infty \preceq \tau \infty \preceq \pi \infty \preceq \rho \infty \preceq \hat{\rho} \infty \preceq \rho'$$

The naive strategy is to argue that because (ρ, x) would have aborted the attack of all $(\hat{\rho}, \hat{x})$ that extend $\rho \infty$ and were that were fully corrupted after x was established, then if (ρ, x) wants to start another attack later at a stage where $\Phi_\pi(A, y) \uparrow$, then some $(\hat{\rho}, \hat{x})$ that was fully corrupted before x was established must have enumerated a number into A since the stage it finished its last attack. Then we are able to count for this prevented attack as we did above. This is in fact true, but the problem is that we must ensure that we have not already counted towards this enumeration. So what we actually need to show is that this enumeration happened since the last

stage a (ρ, x) attack was prevented. This would work if we are able to abort attacks whenever there is a W change prompting a (ρ, x) attack regardless of whether we actually begin this attack or not; however this does not always work. Now to see this we consider the configuration above and the order of events shown in the timeline in figure 4.1 on page 77. x is established at the start of an arc labelled x and fully corrupted at the end of this arc. An arc labelled (ρ, x) attack means this attack started at the start of this arc and finished at the end.

Clearly we do not want to start the last (ρ, x) attack, because $(\pi, y) \in \text{grd}(\rho, x)$ and $\Phi_\pi(A, y) \uparrow$. So if this attack starts it could injure this computation, but the length of this attack was revealed after y was established since it finished an attack after y was established. So we definitely need to abort this attack, but the question is: how do we count this? Notice that we are able to abort an attack for every enumeration into A due to (ρ', x') , but we already used this enumeration to abort an attack. Now since $\hat{\rho} \succeq \rho \infty$ and \hat{x} was fully corrupted after x was fully corrupted, (ρ, x) attacks abort $(\hat{\rho}, \hat{x})$ attacks. Then if we let (ρ, x) abort attacks at the stage we prevented its attack, then we would not have a problem because $(\hat{\rho}, \hat{x})$ would no longer be in an attack injuring (π, y) . But now how would $(\hat{\rho}, \hat{x})$ count for this aborted attack? (ρ, x) attacks can abort $(\hat{\rho}, \hat{x})$ attacks because when \hat{x} is fully corrupted x has already been fully corrupted; hence it is able to incorporate $k^\rho(x)$ into its $k^{\hat{\rho}}(\hat{x})$. Now when the first (ρ, x) attack is prevented it is due to the (ρ', x') enumeration. This was counted because x' was fully corrupted before x was established so we were able to incorporate $k^{\rho'}(x')$ into $m^\rho(x)$; hence we decreased $m^\rho(x)$ when the (ρ', x') attack started so that the length of the attack $b^{\rho'}(z')$ can be incorporated into $k^\rho(x)$. So the enumeration into A causes $b^{\rho'}(z')$ to decrease which gives us a decrease in $k^\rho(x)$. Now notice that this $k^\rho(x)$ is different from the $k^\rho(x)$ that $(\hat{\rho}, \hat{x})$ was able to incorporate into its $k^{\hat{\rho}}(\hat{x})$; hence we cannot abort the $(\hat{\rho}, \hat{x})$ attack unless (ρ, x) actually begins an attack. Then the timeline on page 77 is possible; $(\hat{\rho}, \hat{x})$ could do another enumeration that again injures $\Phi_\pi(A, y)$, followed by a W change

prompting a (ρ, x) attack and we need to prevent this attack if $\Phi_\pi(A, y)$ has not yet recovered. Now this could happen for the whole length of the $(\hat{\rho}, \hat{x})$ attack; hence a (ρ, x) attack may need to be aborted up to $b^{\hat{\rho}}(\hat{z})$ many times. So the question is: can we take this into account? When was $b^{\hat{\rho}}(\hat{z})$ revealed?

Now $(\hat{\rho}, \hat{x})$ carries out an attack before y is established, so $b^{\hat{\rho}}(\hat{z})$ is revealed at the first $\hat{\rho}$ -expansionary stage after this attack has finished. Now recall that since x' was corrupted before x was established, we decrease $m^\rho(x)$ by one each time $d^{\rho'}(z')$ decreases. When this happens we get to choose a new $k^\rho(x)$; we do this so that we can incorporate $b^{\rho'}(z')$ into $k^\rho(x)$. So notice that when this happens we could also incorporate anything else we want; in particular we can incorporate $b^{\hat{\rho}}(\hat{z})$ into $k^\rho(x)$ as well. But we can only do this when we have seen $d^{\hat{\rho}}(\hat{z})$ decrease and it is possible that given the timeline above this has not happened after $(\hat{\rho}, \hat{x})$ finishes its attack; hence the $b^{\hat{\rho}}(\hat{z})$ that we are able to incorporate is the length for a different attack and is small compared to the length of the attack that we need to count against. Now naively we may argue that since the (ρ', x') attack was aborted by that $(\hat{\rho}, \hat{x})$ attack we had a way to count this, so $k^{\hat{\rho}}(\hat{x})$ decreased at the first ρ' -expansionary stage after the $(\hat{\rho}, \hat{x})$ attack finished. Then at this stage the new $b^{\hat{\rho}}(\hat{z})$ has been revealed and so if we incorporate $k^{\hat{\rho}}(\hat{x})$ into $m^\rho(x)$ instead of just $d^{\rho'}(z')$, we could at this stage decrease $m^\rho(x)$ by one and incorporate $b^{\hat{\rho}}(\hat{z})$ as desired. Suppose we do this and now we explain why this fails.

First recall that $(\hat{\rho}, \hat{x})$ attacks are able to abort (ρ', x') attacks because x' was able to take \hat{x} into account when it is established; hence at the stage \hat{x} is fully corrupted we can decrease $m^{\rho'}(x')$ by one and incorporate $k^{\hat{\rho}}(\hat{x})$ into $k^{\rho'}(x')$. Now notice that we incorporated $k^{\hat{\rho}}(\hat{x})$ into $m^\rho(x)$ at the stage x was established, but this was before the stage \hat{x} was fully corrupted; hence at this stage $k^{\hat{\rho}}(\hat{x})$ has not yet been incorporated into $k^{\rho'}(x')$. Then while we are able to abort the (ρ', x') attack because we can charge the decrease to $k^{\hat{\rho}}(\hat{x})$, we are not able to decrease $m^\rho(x)$ by one. So it is important that we only decrease $m^\rho(x)$ only when $d^{\rho'}(z')$ decreases.

Therefore notice that this sequence of events is in fact a problem; we must prevent a (ρ, x) attack if there has been a $(\hat{\rho}, \hat{x})$ enumeration injuring (π, y) , but there is no way for (ρ, x) to incorporate a bound for this into its ordinal. Then if we do not prevent such attacks, we are not guaranteed to meet all N_Φ requirements, and if we prevent such attacks then we are not guaranteed to meet all P_Ψ requirements.

Clearly the modifications discussed in the two c.e. sets case will not be enough for three c.e. sets. It is not yet clear to us what modifications are required to make this work. If this is true then it seems a construction to prove it will need new ideas. But this situation is concrete enough that it could be exploited in order to attempt to prove the negation. For the negation we prove that given a totally ω -c.a. degree, there is a maximal totally ω^2 -c.a. degree above it. Notice that due to Theorem 1.6 and the discussion about the two c.e. set case above, such a construction will require us to build three sets, one of which will be a maximal totally ω^2 -c.a. degree above the given totally ω -c.a. degree.

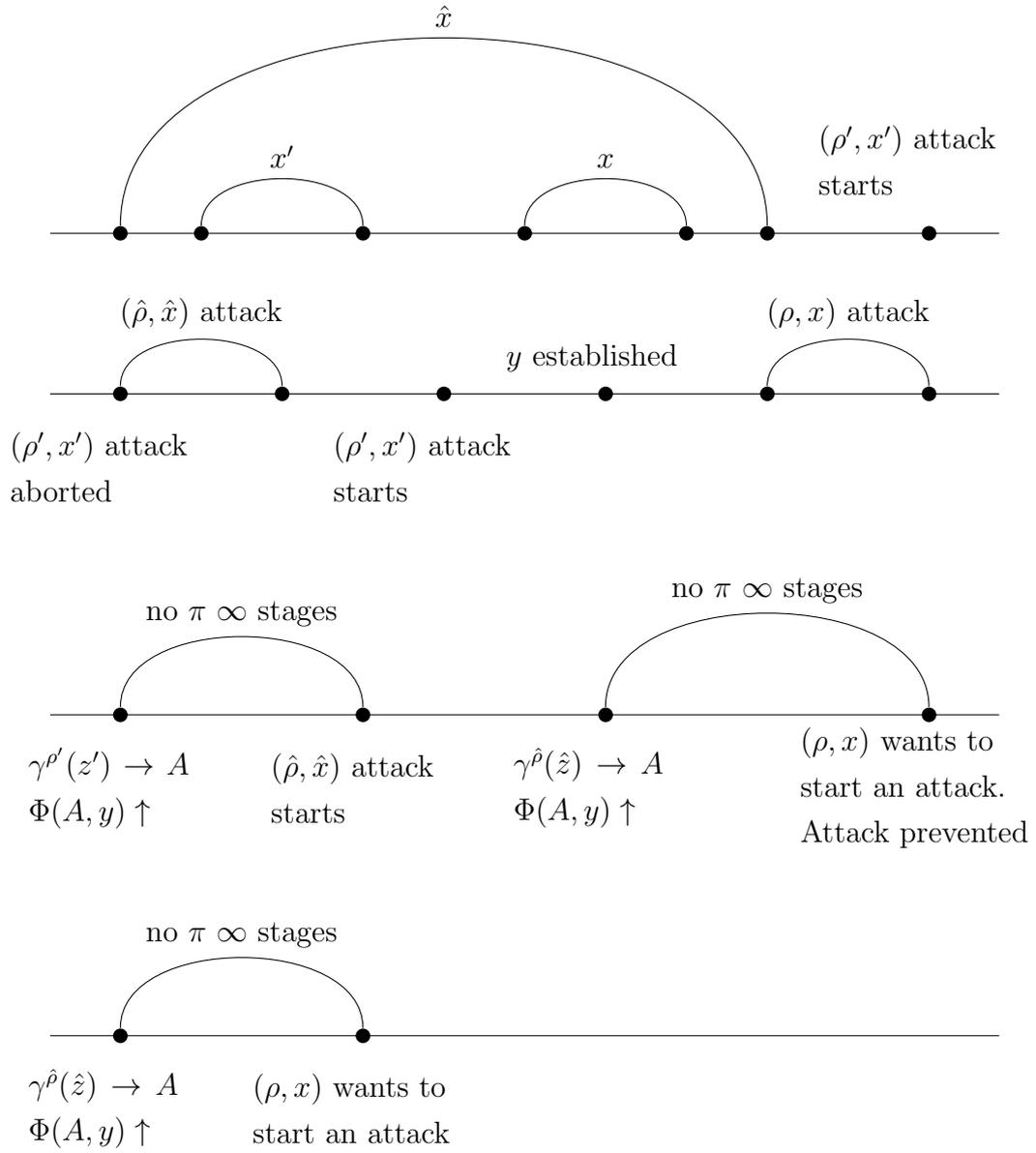


Figure 4.1: Timeline showing a possible interaction between three P requirements and an N requirement.

Bibliography

- [1] ARTHUR, K., DOWNEY, R., AND GREENBERG, N. Maximality and collapse in the hierarchy of -c.a. degrees. *Computability* (Feb. 2021), 134.
- [2] DOWNEY, R., AND GREENBERG, N. *A hierarchy of Turing degrees : a transfinite hierarchy of lowness notions in the computably enumerable degrees, unifying classes, and natural definability*. Princeton University Press, Princeton, New Jersey, 2020.
- [3] DOWNEY, R., GREENBERG, N., AND WEBER, R. Totally ω -computably enumerable degrees and bounding critical triples. *Journal of Mathematical Logic* 07, 02 (2007), 145171.
- [4] DOWNEY, R., JOCKUSCH, C., AND STOB, M. Array nonrecursive sets and multiple permitting arguments. *Lecture Notes in Mathematics Recursion Theory Week* (1990), 141173.
- [5] EPSTEIN, R. L., HAAS, R., AND KRAMER, R. L. Hierarchies of sets and degrees below $\mathbf{0}'$. In *Logic Year 1979–80* (Berlin, Heidelberg, 1981), M. Lerman, J. H. Schmerl, and R. I. Soare, Eds., Springer Berlin Heidelberg, pp. 32–48.
- [6] ERSHOV, Y. L. A hierarchy of sets. i. *Algebra and Logic* 7, 1 (Jan. 1968), 25–43.

- [7] ERSHOV, Y. L. On a hierarchy of sets, II. *Algebra and Logic* 7, 4 (July 1968), 212–232.
- [8] ERSHOV, Y. L. On a hierarchy of sets. III. *Algebra and Logic* 9, 1 (Jan. 1970), 20–31.
- [9] SHOENFIELD, J. R. On degrees of unsolvability. *Annals of Mathematics* 69, 3 (1959), 644–653.
- [10] SHORE, R. A. A non-inversion theorem for the jump operator. *Annals of Pure and Applied Logic* 40, 3 (Dec. 1988), 277–303.