The Journal of Symbolic Logic
Volume 59, Number 4, December 1994

EMBEDDING LATTICES INTO THE wtt-DEGREES BELOW $0'$

ROD DOWNEY AND CHRISTINE HAUGHT

§1. **Introduction.** A reducibility $\leq_p$ is a procedure whereby a set A can be computed from a set B. The most general and most extensively studied reducibility is Turing reducibility ($\leq_T$). However, when one analyzes effectiveness considerations in classical mathematics, one often discovers that the relevant reducibilities are stronger (i.e., more restrictive) than $\leq_T$. To illustrate, in combinatorial group theory we find that the word problem is many-one reducible to the conjugacy problem, and that word problems occur in each r.e. truth table (tt-) degree (see, for example, Miller [17]).

In the present paper we are concerned with another strong reducibility: weak truth table (wtt-) reducibility. Here the reader should recall that $A \leq_{wtt} B$ means that there is a procedure Φ and a recursive function φ such that $\Phi(B) = A$ and for all x, the $u(\Phi(B; x)) < \varphi(x)$. That is, the amount of information used in the computation is bounded by φ. The critical difference between truth table and weak truth table reducibilities is that for tt we will at once be "given the whole table." Thus if Δ is a tt-procedure and δ is its use, then for all x and all strings σ of length $\delta(x)$ we can figure out $\Delta(\sigma; x)$. On the other hand if Δ is merely a wtt-procedure it may be that for some string σ, $\Delta(\sigma; x)\downarrow$, whilst for another string μ of the same length it may be that $\Delta(\mu; x)\uparrow$. We remark that wtt-reducibility arises very naturally both in effective algebra and in the structure of the r.e. T-degrees $\mathbf{R}$. The reader should see, for instance, Downey and Remmel [3], where it is shown that the complexity of r.e. bases of an r.e. vector space V is characterised precisely by the wtt-degrees below V, and also Ladner and Sasso [14] or Downey [1], where the wtt-degrees are used to investigate cupping and capping in $\mathbf{R}$.

Associated with each reducibility p is its natural degree structure $\mathscr{D}_p$. In this paper we shall investigate $\mathscr{D}_{wtt}(\leq 0')$—that is, the wtt-degrees below the halting set. The reader should note that these are not the wtt-degrees of Δ^0_2 sets but rather those that occur as wtt-degrees of ω-r.e. sets in Ershov's hierarchy (see, for example, Ershov [5] or Epstein-Haas-Kramer [4]). The ω-r.e. sets are those sets A for which there exist recursive functions f and g such that $A(x) = \lim_s f(x, s)$ and $|\{s : f(x, s+1) \neq f(x, s)| \leq g(x)$. We remark that this restriction makes

Received Octobert 29, 1991; revised June 15, 1993 and March 15, 1994.

This paper evolved over a number of years, during which the first author was partially supported by the NSF, the MSRI, the Victoria University IGC and the US/NZ binational cooperative foundation. The second author was partially supported by NSF Grant DMS 8705818, as well as by an NSF postdoctoral fellowship.

0022-4812/94/5904-0016/$03.30

constructions concerned with $\mathscr{D}_{wtt}(\leq 0')$ rather delicate and, in particular, seems to rather preclude the use of oracle construction techniques.

The main structural results currently known for $\mathscr{D}_{wtt}(\leq 0')$ are due to P. Fejer and R. Shore [7], where a direct construction of a minimal r.e. tt-degree is given and it is noted that the construction can be modified to construct a minimal wtt-degree below $0'_{wtt}$, and to Haught and Shore [11], where it is shown that $\mathscr{D}_{wtt}(\leq 0')$ has an undecidable first order theory. This latter result is proved using initial segment arguments.

In our paper, we shall prove that every finite lattice is embeddable into $\mathscr{D}_{wtt}(\leq 0')$ preserving 0 and 1. From this it follows that the existential theory in the language $(=, \wedge, \vee, 0, 1)$ is decidable. The analogous result was proved by Haught [8, 9] for the r.e. tt-degrees, and, in some as yet unpublished work of Lerman and Shore, for the T-degrees below $0'$. We remark that all of this work builds upon ideas of Odifreddi [18] and, in particular, Jockusch and Mohrherr [12]. (We make further comments upon this in the proof.)

It is very noticeable that results for the r.e. tt-degrees seem to so far always to have analogues in $\mathscr{D}_{wtt}(\leq 0')$. It is not beyond the realm of possibility that, in fact, *the r.e. tt-degrees may be an elementary substructure of* $\mathscr{D}_{wtt}(\leq 0')$. At present we know of no distinction that even *precludes isomorphism*.

We also remark that the sets we construct actually occur at the ω-level of the Ershov hierarchy. We do not know if they can be lowered to a finite level. Thus we leave the following question open: *is every finite lattice embeddable* (*preserving* 0 (*and* 1)) *into the d-r.e. wtt-* (*T-*) *degrees*? (See, for example, Downey [2] for information on the d-r.e. T-degrees.)

We remark that our construction is rather more difficult than the analogous one for the r.e. tt-degrees because of the "partialness", mentioned earlier, that distinguishes wtt from tt. This means that, whereas for tt we can "see" the effect of coding, for wtt we must be very careful due to delicate interactions between our coding and infimum requirements.

Notation is standard and follows Soare [21] or Odifreddi [18, 19]. The latter are good references for strong reducibilities. Lerman [15] and Haught [9] will provide the background on the lattice representation techniques we employ to preserve infimums. All computations, etc. are bounded by s at stage s.

§2. The proof.

THEOREM. *Every finite lattice can be embedded in the wtt-degrees below* $0'$ *by a map which preserves* $0, 1, \leq, \wedge, \vee$.

PROOF. Let $\mathscr{L} = \langle P, \leq, \wedge, \vee, 0_{\mathscr{L}}, 1_{\mathscr{L}}\rangle$ be a finite lattice with $P = \{p_0, \dots, p_n\}$, $p_0 = 0_{\mathscr{L}}$, and $p_n = 1_{\mathscr{L}}$. Let $\{p_{i_a} \not\leq p_{j_a} : a \in [1, h]\}$ be a complete listing of all such relations holding in $\mathscr{L}$, and $\{p_{i_a} \wedge p_{j_a} = p_{m_a} : a \in [h+1, l]\}$ be a complete listing of all nontrivial infs holding in $\mathscr{L}$. Let $\Phi = \{\alpha_r : r \in [0, d]\}$, where for each $r \in [0, d]$, $\alpha_r : [0, n] \to [0, d]$, be a finite lattice representation for $\mathscr{L}$. For completeness, we briefly remind the reader that such a representation has the following properties:

(a) $$p_i \leq p_j \leftrightarrow (\forall r_1, r_2)[\alpha_{r_1}(j) = \alpha_{r_2}(j) \to \alpha_{r_1}(i) = \alpha_{r_2}(i)],$$

(b) $p_i \vee p_j = p_k \leftrightarrow (\forall r_1, r_2)[(\alpha_{r_1}(i) = \alpha_{r_2}(i) \text{ and } \alpha_{r_1}(j) = \alpha_{r_2}(j)) \leftrightarrow \alpha_{r_1}(k) = \alpha_{r_2}(k)],$

(c) $p_i \wedge p_j = p_m \leftrightarrow (\forall r_1, r_2)[\alpha_{r_1}(m) = \alpha_{r_2}(m) \leftrightarrow (\exists q)(\exists s_0, \dots, s_q \in [0, d])[s_0 = r_1 \text{ and } s_q = r_2 \text{ and } (\forall e)[e \text{ even and } e < q \rightarrow \alpha_{s_e}(i) = \alpha_{s_{e+1}}(i)) \text{ and } (e \text{ odd and } e < q \rightarrow \alpha_{s_e}(j) = \alpha_{s_{e+1}}(j)]]].$

Every finite lattice has a finite representation (Pudlák and Tůma [20]). (For more details see Lerman [15] and Haught [9].) Fix a recursive partition of ω which satisfies the following:

$$\omega = \left(\bigcup_{e \in \omega} \left(\bigcup_{a \in [1,h]} D(\langle e, a \rangle) \right) \right) \cup \left(\bigcup_{x \in \omega} W(x) \right),$$

where each $D(\langle e, a \rangle)$ is infinite and

$$W(x) = \{\langle x, a \rangle : a \in [1, h]\} \cup \{\langle x, \langle a, k \rangle \rangle : a \in [h+1, l],\ k \in \{1, 2\}\}.$$

We construct $f : \omega^2 \rightarrow [0, d]$, and define $F(x) = \lim_s f(x, s)$,

$$A_{i,s} = \{\langle x, \alpha_{f(x,0)}(i) \rangle : x \in \omega\} \cup \{\langle x, \alpha_{f(x,s)}(i) \rangle : x \in \omega\},$$

and $A_i = \lim_s A_{i,s}$.

Our construction will satisfy the following, where K is a 1-complete r.e. set:

(1) $$A_n \leq_{wtt} K,$$

(2) $$K \leq_{wtt} A_n,$$

(3) $$P_{e_a} : A_{i_a} \neq \Phi_e[A_{j_a}], \quad \text{for } e \in \omega,\ a \in [1, h],$$

(4) $$Q_{e_a} : (\Phi_e[A_{i_a}] = \Phi_e[A_{j_a}] = Y \text{ total}) \Rightarrow Y \leq_{wtt} A_{m_a} \quad \text{for } e \in \omega,\ a \in [h+1, m].$$

Condition (1) is satisfied by guaranteeing that there is a recursive function g such that for all x,

$$|\{s : f(x, s) \neq f(x, s+1)\}| \leq g(x).$$

We assume without loss of generality that if $\alpha_r = \alpha_s$, then $r = s$. Then, $A_n \equiv_{wtt} F$ and $F \leq_{wtt} K$, so $A_n \leq_{wtt} K$. The function g is determined by our strategy for preserving infs: $g(x) = (d+5)(p(x)+1)$, where $p(x)$ is the priority of the requirement to which x is attached; i.e.,

$$p(x) = \begin{cases} \langle e, a\rangle & \text{if } x \in D(\langle e, a\rangle), \\ y & \text{if } x \in W(y). \end{cases}$$

Note that since $p(x) \geq 0$ for all x, we have $g(x) \geq 5$ for all x. The lattice representation provides that if $p_i \nleq p_j$, then there are rows $r_1, r_2 \leq d$ satisfying

$$\alpha_{r_1}(i) \neq \alpha_{r_2}(i) \quad \text{and} \quad \alpha_{r_1}(j) = \alpha_{r_2}(j).$$

We make use of these rows to satisfy (3) and (4). For $a \in [1, h]$, define $r(a)_1$ and $r(a)_2$ to be the least pair of rows such that

$$\alpha_{r(a)_1}(i_a) \neq \alpha_{r(a)_2}(i_a) \quad \text{and} \quad \alpha_{r(a)_1}(j_a) = \alpha_{r(a)_2}(j_a).$$

For $a \in [h+1, l]$, define $r(a,1)_1$ and $r(a,1)_2$ to be the least pair of rows such that

$$\alpha_{r(a,1)_1}(i_a) \neq \alpha_{r(a,1)_2}(i_a) \quad \text{and} \quad \alpha_{r(a,1)_1}(j_a) = \alpha_{r(a,1)_2}(j_a).$$

Finally, define $r(a,2)_1$ and $r(a,2)_2$ to be the least pair of rows such that

$$\alpha_{r(a,2)_1}(j_a) \neq \alpha_{r(a,2)_2}(j_a) \quad \text{and} \quad \alpha_{r(a,2)_1}(i_a) = \alpha_{r(a,2)_2}(i_a).$$

(2) is satisfied by coding K into A_n. We will establish

$$x \in K \iff \exists y \in W(x) \exists r \leq d (r \neq f(y, 0) \text{ and } \langle y, \alpha_r(n)\rangle \in A_n).$$

Let $\{k_t\}_{t \in \omega}$ be a one-to-one recursive enumeration of K. Then we define $K_s = \{k_t : 2t \leq s\}$. The coding will be established by maintaining at every stage the following coding invariant:

$$x \in K_s \iff \exists y \in W(x)[f(y, s) \neq f(y, 0)].$$

We use the numbers in $D(\langle e, a\rangle)$ to satisfy P_{e_a}. In isolation, our strategy is as follows:

If at stage s it appears as though $A_{i_a,s} = \Phi_e[A_{j_a,s}]$ and, for some $y \in D(\langle e, a\rangle)$, $\Phi_e[A_{j_a}](\langle y, \alpha_{r(a)_2}(i_a)\rangle)\downarrow = 0$, then we set $f(y, s+1) = r(a)_2$ (thereby causing $A_{i_a}(\langle y, \alpha_{r(a)_2}(i_a)\rangle) = 1$) and preserve $A_{j_a,s}$ up to the use, $\varphi_e(\langle y, \alpha_{r(a)_2}(i_a)\rangle)$.

In order to maintain a recursive bound on the number of changes in $f(x, s)$, we keep track of it during the construction. The i-status of x at stage s is the device we use for this. The i-status of x at stage s is defined to be $|\{s' : s' < s$ and $f(x, s') \neq f(x, s'+1)$ and the change was caused by the action of the requirement of priority $i\}|$.

We use the following notation for potential changes to f and ensuing changes to A_i:

$$f_s(x) = f(x,s);$$

for $\sigma \in [0,d]^{<\omega}$, $f_s(\sigma)$ is a function from ω to $[0,d]$ given by

$$f_s(\sigma)(x) = \begin{cases} \sigma(x) & \text{if } x < |\sigma|, \\ f(x,s) & \text{if } x \geq |\sigma|; \end{cases}$$

$$A_{i,s}(\sigma) = \{\langle x, \alpha_{f(x,0)}(i)\rangle : x \in \omega\} \cup \{\langle x, \alpha_{f_s(\sigma)(x)}(i)\rangle : x \in \omega\}.$$

We satisfy Q_{e_a} by actively trying to cause $\Phi_e[A_{i_a}] \neq \Phi_e[A_{j_a}]$, and preserving some form of this difference after it appears. This idea goes back to Jockusch and Mohrherr [12] who used it to embed the diamond lattice preserving **0** and **1** in the much easier setting of the r.e. tt-degrees. In our context, there are two parts to this strategy: creating the difference, and then preserving the difference after it has appeared. Since we are building Δ_2^0 approximations, we now have the option of removing numbers from A_{i_a} and A_{j_a} as long as the coding invariant is maintained. This eliminates the need to look ahead and imitate future coding action (as was the case in the r.e. tt-degrees).

We can satisfy Q_{e_a} by looking back and rearranging previous coding action. The strategy for the inf is derived from an analysis of what could go wrong in recovering $\Phi_e[A_{i_a}] = \Phi_e[A_{j_a}]$ from A_{m_a}, and from the lattice representation of the inf. The second part of our strategy will guarantee that after Q_{e_a} has priority, if there is any stage s such that for some z, $\Phi_{e,s}[A_{i_a,s}](z)\downarrow \neq \Phi_{e,s}[A_{j_a,s}](z)\downarrow$, then $\Phi_e[A_{i_a}] \neq \Phi_e[A_{j_a}]$. The situation which needs to be handled with some care is when there are stages $s_1 < s_2$, both occurring after Q_{e_a} has priority, and some z such that

$$A_{m_a,s_1} \upharpoonright \varphi_e(z) = A_{m_a,s_2} \upharpoonright \varphi_e(z)$$

and

$$\Phi_{e,s_1}[A_{i_a,s_1}](z)\downarrow = \Phi_{e,s_1}[A_{j_a,s_1}](z)\downarrow$$

and

$$\Phi_{e,s_2}[A_{i_a,s_2}](z)\downarrow = \Phi_{e,s_2}[A_{j_a,s_2}](z)\downarrow$$

but

$$\Phi_{e,s_1}[A_{i_a,s_1}](z) \neq \Phi_{e,s_2}[A_{i_a,s_2}](z).$$

In this case the lattice representation will give us a way of filtering out the changes that have been made to A_{i_a} and A_{j_a} between stages s_1 and s_2 into something which can be manipulated into $\Phi_e[A_{i_a}](z)\downarrow \neq \Phi_e[A_{j_a}](z)\downarrow$. So, at stage s_2, Q_{e_a}'s action will be to rearrange the action which has occurred between stages s_1 and s_2 so as to result in such a set-up. Precise definitions follow; these are motivated by the interpolants of the lattice representation and the proof of Lemma 6.

We will use $R(\langle e,a\rangle, s)$ to denote the maximum of all restrains imposed by requirements of priority higher than $\langle e,a\rangle$ at the end of stage s.

DEFINITION. A "set-up for a disagreement at z for Q_{e_a} at stage $s+1$" is a sequence $\langle \sigma_1, \dots, \sigma_w\rangle$ such that the following conditions (5)–(10) hold:

(5) $w \leq d+1$.

(6) for $b \leq w$, $\sigma_b \in [0,d]^{<\omega}$ and $|\sigma_b| \geq \varphi_e(z)$.

(7) $(\forall b \leq w)(\sigma_b$ is a legal move for Q_{e_a} at stage $s)$ (i.e., (7.1)–(7.5) below hold).

(7.1) $f_s(\sigma_b)$ maintains the coding invariant at s,

(7.2) if $\sigma_b(x) \neq f(x,s)$ then $x > R(\langle e,a\rangle, s)$,

(7.3) if $\sigma_b(x) \neq f(x,s)$ then $p(x) > \langle e,a\rangle$,

(7.4) if $\sigma_b(x) \neq f(x,s)$ then $\forall i < \langle e,a\rangle$ (the i-status of x at stage s is 0),

(7.5) if $\sigma_b(x) \neq f(x,s)$ then x is not frozen for Q_{e_a} at stage s.

(8) If $w=1$ then some phase of the inf cycle will hold for Q_{e_a} at $s+1$ if we define $f_{s+1} = f_s(\sigma_1)$ (inf cycle will be defined below). If $w>1$, then one of (8.1)–(8.4) below holds for some z:

(8.1) $\Phi_{e,s}[A_{i_a,s}(\sigma_1)](z)\downarrow \neq \Phi_{e,s}[A_{i_a,s}(\sigma_w)](z)\downarrow$ and $\Phi_{e,s}[A_{j_a,s}(\sigma_1)](z)\uparrow$ and $\Phi_{e,s}[A_{j_a,s}(\sigma_w)](z)\uparrow$,

(8.2) $\Phi_{e,s}[A_{j_a,s}(\sigma_1)](z)\downarrow \neq \Phi_{e,s}[A_{j_a,s}(\sigma_w)](z)\downarrow$ and $\Phi_{e,s}[A_{i_a,s}(\sigma_1)](z)\uparrow$ and $\Phi_{e,s}[A_{i_a,s}(\sigma_w)](z)\uparrow$,

(8.3) $\Phi_{e,s}[A_{i_a,s}(\sigma_1)](z)\downarrow \neq \Phi_{e,s}[A_{j_a,s}(\sigma_w)](z)\downarrow$ and $\Phi_{e,s}[A_{j_a,s}(\sigma_1)](z)\uparrow$ and $\Phi_{e,s}[A_{i_a,s}(\sigma_w)](z)\uparrow$,

(8.4) $\Phi_{e,s}[A_{j_a,s}(\sigma_1)](z)\downarrow \neq \Phi_{e,s}[A_{i_a,s}(\sigma_w)](z)\downarrow$ and $\Phi_{e,s}[A_{i_a,s}(\sigma_1)](z)\uparrow$ and $\Phi_{e,s}[A_{j_a,s}(\sigma_w)](z)\uparrow$.

(9) For the z of condition (8) $\forall b(1<b<w)[\Phi_{e,s}[A_{i_a,s}(\sigma_b)](z)\uparrow$ and $\Phi_{e,s}[A_{j_a,s}(\sigma_b)](z)\uparrow]$.

(10) Either (10.1) or (10.2) below holds:

(10.1) $(\forall b<w)(\forall x<|\sigma_b|,|\sigma_{b+1}|)$
$[(b$ odd $\Rightarrow \alpha_{\sigma_b(x)}(i_a) = \alpha_{\sigma_{b+1}(x)}(i_a))$
and $(b$ even $\Rightarrow \alpha_{\sigma_b(x)}(j_a) = \alpha_{\sigma_{b+1}(x)}(j_a))]$;

(10.2) $(\forall b<w)(\forall x<|\sigma_b|,|\sigma_{b+1}|)$
$[(b$ odd $\Rightarrow \alpha_{\sigma_b(x)}(j_a) = \alpha_{\sigma_{b+1}(x)}(j_a))$
and $(b$ even $\Rightarrow \alpha_{\sigma_b(x)}(i_a) = \alpha_{\sigma_{b+1}(x)}(i_a))]$.

DEFINITION. Inf cycle for Q_{e_a} at z at stage s:

Phase A: $\Phi_{e,s}[A_{i_a,s}](z)\downarrow \neq \Phi_{e,s}[A_{j_a,s}](z)\downarrow$. See phase C_i below

Phase B_i: $\Phi_{e,s}[A_{i_a,s}](z)\uparrow$ and $\Phi_{e,s}[A_{j_a,s}](z)\downarrow$ and there is a finite set of numbers, $Add\text{-}j_s$, whose codes can be removed from $A_{i_a,s}$ by means of a legal move for Q_{e_a} at stage s which will create a convergent computation for $\Phi_{e,s}[A_{i_a,s}](z)$ different from $\Phi_{e,s}[A_{j_a,s}](z)$. Removing these codes from $A_{j_a,s}$ will violate the coding invariant, hence the numbers whose codes have been removed, i.e., those in $Add\text{-}j_s$, will have to be recoded into A_{j_a} if we ever make use of this move.

We state this more precisely by requiring that the set $Add\text{-}j_s$ satisfy conditions (11.1)–(11.5) below, where we replace X by $Add\text{-}j_s$:

(11.1) If we define τ of length $\varphi_e(z)$ by $\tau(\langle x,\langle a,1\rangle\rangle)=r(a,1)_1$ for all $x\in Add\text{-}j_s$ and $\tau(y)=f(y,s)$ for all other y, then $\Phi_{e,s}[A_{i_a,s}(\tau)](z)\downarrow\neq\Phi_{e,s}[A_{j_a,s}](z)\downarrow$. Note that this τ does not affect $A_{j_a,s}$, i.e., $A_{j_a,s}(\tau)=A_{j_a,s}$, and that $f_s(\tau)$ does not maintain the coding invariant. The effect of τ is to remove the codes of the numbers in $Add\text{-}j_s$ from A_{i_a}; τ does not recode these numbers, and so their codes will need to be added to A_{j_a} if we change f_s to $f_s(\tau)$.

(11.2) For all $x\in X$, for all $i\leq\langle e,a\rangle$ the i-status of $\langle x,\langle a,1\rangle\rangle$ is less than $g(\langle x,\langle a,1\rangle\rangle)-2$.

(11.3) For all $x\in X$, $x\in K_s$.

(11.4) For all $x\in X$, x is not frozen for Q_{e_a} at s.

(11.5) For all $x\in X$, $x\geq R(\langle e,a\rangle,s)$.

Phase B_j: $\Phi_{e,s}[A_{i_a,s}](z)\downarrow$ and $\Phi_{e,s}[A_{j_a,s}](z)\uparrow$ and there is a finite set of numbers, $Add\text{-}i_s$, whose codes can be removed from $A_{j_a,s}$ by means of a legal move for Q_{e_a} at stage s which will create a convergent computation for $\Phi_{e,s}[A_{j_a,s}](z)$ different from $\Phi_{e,s}[A_{i_a,s}](z)$. Removing these codes from $A_{j_a,s}$ will violate the coding invariant; hence the numbers whose codes have been removed, i.e., those in $Add\text{-}i_s$ will have to be recoded into A_{i_a} if we ever make use of this move.

More precisely, we insist that $Add\text{-}i_s$ satisfy conditions (11.2)–(11.5) above on X, and condition (12.1) below:

(12.1) If we define τ of length $\varphi_e(z)$ by $\tau(\langle x,\langle a,2\rangle\rangle)=r(a,2)_1$ for all $x\in Add\text{-}i_s$ and $\tau(y)=f(y,s)$ for all other y, then $\Phi_{e,s}[A_{j_a,s}(\tau)](z)\downarrow\neq\Phi_{e,s}[A_{i_a,s}(\tau)](z)\downarrow$. Note that this τ does not affect $A_{i_a,s}$, i.e., $A_{i_a,s}(\tau)=A_{i_a,s}$. The effect of τ is to remove the codes of the numbers in X from A_{j_a}; τ does not recode these numbers, and so their codes will need to be added to A_{i_a} if we change f_s to $f_s(\tau)$.

Phase C_i: $\Phi_{e,s}[A_{i_a,s}](z)\uparrow$ and $\Phi_{e,s}[A_{j_a,s}](z)\downarrow$ and there is a finite set of numbers, *Remove-j_s*, whose codes can be removed from $A_{j_a,s}$ by means of a legal move for Q_{e_a} at s which will not violate the coding invariant and will cause $\Phi_{e,s+1}[A_{j_a,s+1}](z)\downarrow\neq\Phi_{e,s}[A_{j_a,s}](z)\downarrow$. This means that *Remove-j_s* must satisfy conditions (11.2)–(11.5) on X above and condition (13.1) below:

(13.1) If we define τ of length $\varphi_e(z)$ by

$$\tau(\langle x,\langle a,2\rangle\rangle)=r(a,2)_1 \text{ if } x\in \text{ Remove-}j_s,$$

and $\tau(y)=f(y,s)$ for all other y, then $\Phi_{e,s}[A_{j_a,s}(\tau)](z)\downarrow\neq\Phi_{e,s}[A_{j_a,s}](z)\downarrow$ and $f_s(\tau)$ does satisfy the coding invariant. Note that this τ does not affect $A_{i_a,s}$, i.e., $A_{i_a,s}(\tau)=A_{i_a,s}$. The effect of τ is to remove the codes of the numbers in *Remove-j_s*

from A_{j_a}. Further note that since $f_s(\tau)$ does satisfy the coding invariant, we can conclude that the numbers in *Remove*-j_s are already coded into A_{i_a}; these numbers do not need to be recoded.

Phase C_j: $\Phi_{e,s}[A_{i_a,s}](z)\downarrow$ and $\Phi_{e,s}[A_{j_a,s}](z)\uparrow$ and there is a finite set of numbers, *Remove*-i_s, whose codes can be removed from $A_{i_a,s}$ by means of a legal move for Q_{e_a} at s which will not violate the coding invariant, and will cause $\Phi_{e,s+1}[A_{i_a,s+1}](z)\downarrow\neq\Phi_{e,s}[A_{i_a,s}](z)\downarrow$. This means that *Remove*-i_s must satisfy conditions (11.2)–(11.5) on X above and condition (14.1) below:

(14.1) If we define τ of length $\varphi_e(z)$ by $\tau(\langle x,\langle a,1\rangle\rangle)=r(a,1)_1$ if $x\in$ *Remove*-i_s, and $\tau(y)=f(y,s)$ for all other y, then $\Phi_{e,s}[A_{i_a,s}(\tau)](z)\downarrow\neq\Phi_{e,s}[A_{i_a,s}](z)\downarrow$, and $f_s(\tau)$ maintains the coding invariant. Note that this τ does not affect $A_{j_a,s}$, i.e., $A_{j_a,s}(\tau)=A_{j_a,s}$. The effect of τ is to remove the codes of the numbers in *Remove*-i_s from A_{i_a}. Further note that since $f_s(\tau)$ does satisfy the coding invariant, we can conclude that the numbers in *Remove*-i_s are already coded into A_{j_a}; these numbers do not need to be recoded.

The second part of the strategy for Q_{e_a} is to preserve a difference once it has been created. There are two kinds of preservation here: first, if we have a sequence which gives a set-up for a difference at z, but no actual difference yet, and second, if there is an actual difference. We distinguish between these two kinds of preservation in order to be able to keep a recursive bound on the number of changes in $f(x,s)$. In either case we will have a restraint of priority $\langle e,a\rangle$ to protect the use. So, after higher priority actions have stopped, only two things can affect these set-ups and computations. The first is the coding of some k_t less than the use of the computation ($=\varphi_e(z)$). The second is computations which were once divergent becoming convergent with the passage of time. Below we give instructions for handling each of these situations.

Before giving the instructions for coding k_t which will maintain a restricted kind of difference at z, we need another bit of notation. The notation is used for possible ways to code k_t.

DEFINITION. $\sigma\oplus\langle k_t,\langle a,j\rangle\rangle=\sigma'$, where $\sigma'\in[0,d]^{<\omega}$,

$$|\sigma'|=\max\{|\sigma|,\langle k_t,\langle a,j\rangle\rangle+1\},$$

and

$$\sigma'(x)=\begin{cases} r(a,j)_2 & \text{if } x=\langle k_t,\langle a,j\rangle\rangle,\\ \sigma(x) & \text{if } x\neq\langle k_t,\langle a,j\rangle\rangle \text{ and } x<|\sigma|,\\ f(x,2t) & \text{otherwise}.\end{cases}$$

Instructions for coding k_t *at stage* $s+1=2t+1$, *with respect to* Q_{e_a}, *where* $\langle\sigma_1,\dots,\sigma_w\rangle$ *is the set-up for a difference at* z *attached to* Q_{e_a} *at stage* s:

For b such that $1<b<w$, define

$$\sigma_b'=\sigma_b\oplus\langle k_t,\langle a,1\rangle\rangle\oplus\langle k_t,\langle a,2\rangle\rangle.$$

If (8.1) holds, then define

$$\sigma_1'=\sigma_1\oplus\langle k_t,\langle a,2\rangle\rangle \quad\text{and}\quad \sigma_w'=\sigma_w\oplus\langle k_t,\langle a,2\rangle\rangle.$$

If (8.2) holds, then define

$$\sigma_1' = \sigma_1 \oplus \langle k_t, \langle a, 1\rangle\rangle \quad \text{and} \quad \sigma_w' = \sigma_w \oplus \langle k_t, \langle a, 1\rangle\rangle.$$

If (8.3) holds, then define

$$\sigma_1' = \sigma_1 \oplus \langle k_t, \langle a, 2\rangle\rangle \quad \text{and} \quad \sigma_w' = \sigma_w \oplus \langle k_t, \langle a, 1\rangle\rangle.$$

If (8.4) holds, then define

$$\sigma_1' = \sigma_1 \oplus \langle k_t, \langle a, 1\rangle\rangle \quad \text{and} \quad \sigma_w' = \sigma_w \oplus \langle k_t, \langle a, 2\rangle\rangle.$$

Let $\langle \tau_1, \dots, \tau_p\rangle$ be the shortest subsequence of $\langle \sigma_1', \dots, \sigma_w'\rangle$ which is a set-up for a disagreement at z for Q_{e_a}. (Lemma 2 will show that such a subsequence must exist.) Define $f_{s+1} = f_s(\tau_1)$ and attach $\langle \tau_1, \dots, \tau_p\rangle$ to Q_{e_a} at stage $s+1$.

Instructions for coding k_t at stage $s+1 = 2t+1$ with respect to Q_{e_a}, where some phase of the inf *cycle holds at the end of stage s.* (There is no set-up for a disagreement attached to Q_{e_a} at stage s.)

Define $\tau_1 = f_s \restriction l \oplus \langle k_t, \langle a, 1\rangle\rangle$, $\tau_2 = f_s \restriction l \oplus \langle k_t, \langle a, 2\rangle\rangle$ and $\tau_3 = f_s \restriction l \oplus \langle k_t, \langle a, 1\rangle\rangle \oplus \langle k_t, \langle a, 2\rangle\rangle$, where $l = \max\{\langle k_t, \langle a, 1\rangle\rangle, \langle k_t, \langle a, 2\rangle\rangle\}$.

If Phase A holds:

If for $\tau = \tau_1, \tau_2$, or τ_3, $\Phi_{e,s+1}[A_{i_a,s}(\tau)](z)\downarrow \neq \Phi_{e,s+1}[A_{j_a,s}(\tau)](z)\downarrow$, then define $f_{s+1} = f_s(\tau)$. We remain in Phase A.

Otherwise either $\Phi_{e,s+1}[A_{i_a,s}(\tau_3)](z)\uparrow$ or $\Phi_{e,s+1}[A_{j_a,s}(\tau_3)](z)\uparrow$.

If $\Phi_{e,s+1}[A_{i_a,s}(\tau_3)](z)\uparrow$ then define $f_{s+1} = f_s(\tau_1)$. We enter Phase B_i of the inf cycle, with $Add\text{-}j_{s+1} = \{k_t\}$.

Since we are coding k_t at stage $s+1 = 2t+1$, this is our first attempt to code k_t, so $f(\langle k_t, \langle a, 1\rangle\rangle, s') = f(\langle k_t, \langle a, 1\rangle\rangle, 0)$, for all s', $0 \leq s' \leq s$, so for any $i < \langle e, a\rangle$, the i-status of $\langle k_t, \langle a, 1\rangle\rangle$ is 0, and hence $< g(\langle k_t, \langle a, 1\rangle\rangle) - 2$.

Similarly, if $\Phi_{e,s+1}[A_{j_a,s}(\tau_3)](z)\uparrow$ but $\Phi_{e,s+1}[A_{i_a,s}(\tau_3)](z)\downarrow$, then define $f_{s+1} = f_2(\tau_2)$. We enter Phase B_j of the inf cycle, with $Add\text{-}i_{s+1} = \{k_t\}$.

If Phase B_i holds:

If for $\tau = \tau_1, \tau_2$, or τ_3, $\Phi_{e,s+1}[A_{i_a,s}(\tau)](z)\downarrow \neq \Phi_{e,s+1}[A_{j_a,s}(\tau)](z)\downarrow$, then define $f_{s+1} = f_s(\tau)$. We return to Phase A.

Otherwise, if $\Phi_{e,s+1}[A_{i_a,s}(\tau_1)](z)\uparrow$, then define $f_{s+1} = f_s(\tau_1)$ and $Add\text{-}j_{s+1} = Add\text{-}j_s \cup \{k_t\}$. We remain in Phase B_i.

As was the case with Phase A, we are coding k_t for the first time now, so for any $i < \langle e, a\rangle$, the i-status of $\langle k_t, \langle a, 1\rangle\rangle$ is less than $g(\langle k_t, \langle a, 1\rangle\rangle) - 2$.

Otherwise, $\Phi_{e,s+1}[A_{i_a,s}(\tau_1)](z)\downarrow = \Phi_{e,s+1}[A_{j_a,s}(\tau_1)](z)\downarrow$, since we could not return to Phase A.

Define η_2 and η_3 of length equal to the maximum of $\varphi_e(z)$ and $\langle k_t, \langle a, 2\rangle\rangle$ by

$$\begin{aligned} \eta_2'(\langle x, \langle a,1\rangle\rangle) &= r(a,1)_1 \quad \text{for } x \in Add\text{-}j_s;\\ \eta_2'(\langle x, \langle a,2\rangle\rangle) &= r(a,2)_2 \quad \text{for } x \in Add\text{-}j_s;\\ \eta_2'(y) &= f(y,s) \quad \text{for all other } y \leq \varphi_e(z);\\ \eta_2 &= \eta_2' \oplus \langle k_t, \langle a,2\rangle\rangle, \end{aligned}$$

and

$$\begin{aligned} \eta_3'(\langle x, \langle a,2\rangle\rangle) &= r(a,2)_2 \quad \text{for } x \in Add\text{-}j_s;\\ \eta_3'(y) &= f(y,s) \quad \text{for all other } y \leq \varphi_e(z);\\ \eta_3 &= \eta_3' \oplus \langle k_t, \langle a,1\rangle\rangle \oplus \langle k_t, \langle a,2\rangle\rangle. \end{aligned}$$

(Note: η_2 codes the numbers in $Add\text{-}j_s$ on the j_a-side and removes the coding from the i_a-side, thereby changing A_{i_a} back to what it was at the beginning of this cycle. η_3 duplicates the coding which has already been done on the i_a-side onto the j_a-side, keeping $A_{i_a,s}$ unchanged. k_t is coded into A_{j_a}.)

Note that η_2 and η_3 both maintain the coding invariant, and that

$$\Phi_{e,s+1}[A_{i_a,s}(\eta_2)](z)\downarrow \neq \Phi_{e,s+1}[A_{i_a,s}(\eta_3)](z)\downarrow.$$

This follows from the definition of Phase B_i of the inf cycle, since we are assuming in this case that $\Phi_{e,s+1}[A_{i_a,s}(\tau_1)](z)\downarrow = \Phi_{e,s+1}[A_{j_a,s}(\tau_1)](z)\downarrow$, and the definition of Phase B_i guarantees that $\Phi_{e,s+1}[A_{i_a,s}(\tau_1)](z)\downarrow \neq \Phi_{e,s+1}[A_{j_a,s}](z)\downarrow$. Combining these facts with $A_{j_a,s} = A_{j_a,s}(\tau_1)$ and $A_{i_a,s}(\eta_3) = A_{i_a,s}(\tau_1)$ gives the above inequality.

Also

$$\Phi_{e,s+1}[A_{j_a,s}(\eta_2)](z) = \Phi_{e,s+1}[A_{j_a,s}(\eta_3)](z),$$

(both might diverge), since $A_{j_a,s}(\eta_2) = A_{j_a,s}(\eta_3)$.

If $\Phi_{e,s+1}[A_{j_a,s}(\eta_3)](z)\downarrow$, then choose $\eta = \eta_2$ or η_3 so that

$$\Phi_{e,s+1}[A_{i_a,s}(\eta)](z)\downarrow \neq \Phi_{e,s+1}[A_{j_a,s}(\eta)](z)\downarrow.$$

Define $f_{s+1} = f_s(\eta)$. We return to Phase A.

Otherwise, $\Phi_{e,s+1}[A_{j_a,s}(\eta_3)](z)\uparrow$.

Define $f_{s+1} = f_s(\eta_3)$. We enter Phase C_j with $Remove\text{-}i_{s+1} = Add\text{-}j_s \cup \{k_t\}$.

If Phase B_j holds:

If for $\tau = \tau_1, \tau_2$, or τ_3, $\Phi_{e,s+1}[A_{i_a,s}(\tau)](z)\downarrow \neq \Phi_{e,s+1}[A_{j_a,s}(\tau)](z)\downarrow$, then define $f_{s+1} = f_s(\tau)$. We return to Phase A.

Otherwise, if $\Phi_{e,s+1}[A_{j_a,s}(\tau_2)](z)\uparrow$, then define $f_{s+1} = f_s(\tau_2)$ and $Add\text{-}i_{s+1} = Add\text{-}i_s \cup \{k_t\}$. We remain in Phase B_j.

Otherwise, $\Phi_{e,s+1}[A_{j_a,s}(\tau_2)](z)\downarrow = \Phi_{e,s+1}[A_{i_a,s}(\tau_2)](z)\downarrow$ (since we could not return to Phase A).

Define η_1 and η_3 of length $\varphi_e(z)$ by

$$\eta_1'(\langle x, \langle a, 2\rangle\rangle) = r(a,2)_1 \quad \text{if } x \in \mathit{Add}\text{-}i_s\,;$$
$$\eta_1'(\langle x, \langle a, 1\rangle\rangle) = r(a,1)_2 \quad \text{if } x \in \mathit{Add}\text{-}i_s\,;$$
$$\eta_1'(y) = f(y,s) \quad \text{for all other } y;$$
$$\eta_1 = \eta_1' \oplus \langle k_t, \langle a, 1\rangle\rangle,$$

and

$$\eta_3'(\langle x, \langle a, 1\rangle\rangle) = r(a,1)_2 \quad \text{if } x \in \mathit{Add}\text{-}i_s\,;$$
$$\eta_3'(y) = f(y,s) \quad \text{for all other } y;$$
$$\eta_3 = \eta_3' \oplus \langle k_t, \langle a, 1\rangle\rangle \oplus \langle k_t, \langle a, 2\rangle\rangle\,.$$

(η_1 copies the coding which has already been done on the j_a-side onto the i_a-side and removes it from the j_a-side, thereby changing A_{j_a} back to what it was at the beginning of this cycle. η_3 duplicates the coding onto the i_a-side, keeping $A_{j_a,s}$ unchanged. k_t is coded into A_{i_a}.)

Note that η_1 and η_3 both maintain the coding invariant, and

$$\Phi_{e,s+1}[A_{j_a,s}(\eta_1)](z)\downarrow \neq \Phi_{e,s+1}[A_{j_a,s}(\eta_3)](z)\downarrow,$$

by definition of Phase B_j of the inf cycle, and

$$\Phi_{e,s+1}[A_{i_a,s}(\eta_1)](z) = \Phi_{e,s+1}[A_{i_a,s}(\eta_3)](z)$$

(both might diverge), since $A_{i_a,s}(\eta_1) = A_{i_a,s}(\eta_3)$.

If $\Phi_{e,s+1}[A_{i_a,s}(\eta_3)](z)\downarrow$, then choose $\eta = \eta_1$ or η_3 so that

$$\Phi_{e,s+1}[A_{i_a,s}(\eta)](z)\downarrow \neq \Phi_{e,s+1}[A_{j_a,s}(\eta)](z)\downarrow.$$

Define $f_{s+1} = f_s(\eta)$. We return to Phase A.

Otherwise, $\Phi_{e,s+1}[A_{i_a,s}(\eta_3)](z)\uparrow$.

In this case define $f_{s+1} = f_s(\eta_3)$. We enter Phase C_i with $\mathit{Remove}\text{-}j_{s+1} = \mathit{Add}\text{-}i_s \cup \{k_t\}$.

If Phase C_i holds:

If for $\tau = \tau_1, \tau_2$, or τ_3, $\Phi_{e,s+1}[A_{i_a,s}(\tau)](z)\downarrow \neq \Phi_{e,s+1}[A_{j_a,s}(\tau)](z)\downarrow$, then define $f_{s+1} = f_s(\tau)$. We return to Phase A.

Otherwise, if $\Phi_{e,s+1}[A_{i_a,s}(\tau_1)](z)\uparrow$, then define $f_{s+1} = f_s(\tau_1)$. We remain in Phase C_i with $\mathit{Remove}\text{-}j_{s+1} = \mathit{Remove}\text{-}j_s$.

Otherwise, $\Phi_{e,s+1}[A_{i_a,s}(\tau_1)](z)\downarrow = \Phi_{e,s+1}[A_{j_a,s}(\tau_1)](z)\downarrow$ (since we were unable to re-enter Phase A).

Define τ of length $\varphi_e(z)$ by

$$\tau(\langle x, \langle a, 2\rangle\rangle) = r(a,2)_1 \quad \text{for all } x \in \mathit{Remove}\text{-}j_s\,;$$
$$\tau(\langle k_t, \langle a, 1\rangle\rangle) = r(a,1)_2\,;$$
$$\tau(y) = f(y,s) \quad \text{for all other } y.$$

Define $f_{s+1} = f_s(\tau)$. We return to Phase A.

If Phase C_j holds:

If for $\tau = \tau_1, \tau_2$, or τ_3, $\Phi_{e,s+1}[A_{i_a,s}(\tau)](z)\downarrow \neq \Phi_{e,s+1}[A_{j_a,s}(\tau)](z)\downarrow$, then define $f_{s+1} = f_s(\tau)$. We return to Phase A.

Otherwise, if $\Phi_{e,s+1}[A_{j_a,s}(\tau_2)](z)\uparrow$, then define $f_{s+1} = f_s(\tau_2)$. We remain in Phase C_j with $Remove\text{-}i_{s+1} = Remove\text{-}i_s$.

Otherwise, $\Phi_{e,s+1}[A_{j_a,s}(\tau_2)](z)\downarrow = \Phi_{e,s+1}[A_{i_a,s}(\tau_2)](z)\downarrow$ (since we were unable to reenter Phase A).

Define τ of length $\varphi_e(z)$ by

$$\tau(\langle x, \langle a, 1\rangle\rangle) = r(a,1)_1 \quad \text{for all } x \in Remove\text{-}i_s\,;$$

$$\tau(\langle k_t, \langle a, 2\rangle\rangle) = r(a,2)_2;$$

$$\tau(y) = f(y,s) \quad \text{for all other } y.$$

Set $f_{s+1} = f_s(\tau)$. We return to Phase A.

The inf cycle or a set-up for Q_{e_a} may also be affected by a formerly divergent computation becoming convergent. This adds more information to the picture and so has the potential of moving us through the inf cycle more quickly to an actual preservable difference between $\Phi_{e,s+1}[A_{i_a,s}]$ and $\Phi_{e,s+1}[A_{j_a,s}]$.

Instructions for maintaining the inf *cycle if a formerly divergent computation becomes convergent.* Assume that at stage s we were in one of the following phases and at stage $s+1$ a divergent compuatation in the definition of that phase became convergent.

Phase A. There are no divergent computations in the definition of Phase A, so this is not applicable.

Phase B_i. It must be the case that $\Phi_{e,s}[A_{i_a,s}](z)\uparrow$ and $\Phi_{e,s+1}[A_{i_a,s+1}](z)\downarrow$. (Since $\Phi_{e,s}[A_{j_a,s}](z)\downarrow$ we know that $\Phi_{e,s+1}[A_{j_a,s+1}](z)\downarrow = \Phi_{e,s}[A_{j_a,s}](z)\downarrow$.) If $\Phi_{e,s+1}[A_{i_a,s+1}](z)\downarrow \neq \Phi_{e,s+1}[A_{j_a,s+1}](z)\downarrow$, then we have a difference and we move to Phase A with no adjustment to f_{s+1}. Otherwise it must be the case that $\Phi_{e,s+1}[A_{i_a,s+1}](z)\downarrow = \Phi_{e,s+1}[A_{j_a,s+1}](z)\downarrow$. Recall the move τ which was part of the definition of Phase B_i of the inf cycle: $\tau(\langle x, \langle a,1\rangle\rangle) = r(a,1)_1$ for all $x \in Add\text{-}j_s$ and $\tau(y) = f(y,s)$ for all other $y < |\tau|$. This τ will give us $\Phi_{e,s+1}[A_{i_a,s+1}(\tau)](z)\downarrow \neq \Phi_{e,s}[A_{j_a,s+1}](z)\downarrow$. As in the instructions for coding k_t from Phase B_i, we must modify this move in order to maintain the coding invariant. Define τ' of length $|\tau|$ by $\tau'(\langle x, \langle a,2\rangle\rangle) = r(a,2)_2$ for $x \in Add\text{-}j_s$ and $\tau'(y) = \tau(y)$ for all other y. Then $\Phi_{e,s+1}[A_{i_a,s+1}(\tau)](z)\downarrow = \Phi_{e,s+1}[A_{i_a,s+1}(\tau')](z)\downarrow$. If $\Phi_{e,s+1}[A_{i_a,s+1}(\tau')](z)\downarrow \neq \Phi_{e,s+1}[A_{j_a,s+1}(\tau')](z)\downarrow$, then use τ' to modify the definition of f_{s+1} and move to Phase A. Otherwise we use τ' to modify the definition of f_{s+1} and enter Phase C_j with $Remove\text{-}i_{s+1} = Add\text{-}j_s$.

Phase B_j. This case is the dual of Phase B_i with the roles of i and j reversed.

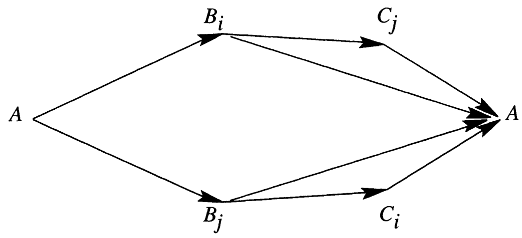
Phase C_i. It must be the case that $\Phi_{e,s}[A_{i_a,s}](z)\uparrow$ and $\Phi_{e,s+1}[A_{i_a,s+1}](z)\downarrow$. (Since $\Phi_{e,s}[A_{j_a,s}](z)\downarrow$ we know that $\Phi_{e,s+1}[A_{j_a,s+1}](z)\downarrow = \Phi_{e,s}[A_{j_a,s}](z)\downarrow$.) If $\Phi_{e,s+1}[A_{i_a,s+1}](z)\downarrow \neq \Phi_{e,s+1}[A_{j_a,s+1}](z)\downarrow$, then we have a difference and we move to Phase A with no adjustment to f_{s+1}. Otherwise it must be the case that $\Phi_{e,s+1}[A_{i_a,s+1}](z)\downarrow = \Phi_{e,s+1}[A_{j_a,s+1}](z)\downarrow$. Recall the move τ which was part of the definition of Phase C_i of the inf cycle: $\tau(\langle x, \langle a, 2\rangle\rangle) = r(a,2)_1$ for all $x \in$ *Remove-*j_s and $\tau(y) = f(y,s)$ for all other $y < |\tau|$. This τ will give us $\Phi_{e,s+1}[A_{j_a,s+1}(\tau)](z)\downarrow \neq \Phi_{e,s+1}[A_{j_a,s+1}](z)\downarrow$. As in the instructions for coding k_t from Phase C_i we recognize that this τ already satisfies the coding invariant, and since $A_{i_a,s+1}(\tau) = A_{i_a,s+1}$, we have $\Phi_{e,s+1}[A_{i_a,s+1}(\tau)](z)\downarrow = \Phi_{e,s+1}[A_{i_a,s+1}](z)\downarrow = \Phi_{e,s+1}[A_{j_a,s+1}](z)\downarrow \neq \Phi_{e,s+1}[A_{j_a,s+1}(\tau)](z)\downarrow$. So, in order to maintain the inf cycle, we modify the definition of f_{s+1} to $f_{s+1}(\tau)$ and move to Phase A.

Phase C_j: This case is the dual of Phase C_i with the roles of i and j reversed.

Instructions for maintaining a set-up for a difference if formerly divergent computations become convergent.

Suppose that $\langle \sigma_1, \dots, \sigma_w\rangle$ was a set-up for a disagreement for Q_{e_a} at z at stage s. Let $b_1 \geq 1$ be the greatest such that $\exists b\ (b_1 \leq b \leq w)$ ($\langle \sigma_{b_1}, \dots, \sigma_b\rangle$ is a set-up for a disagreement at z for Q_{e_a} at stage $s+1$). Let $b_2 \geq b_1$ be the least corresponding b. Then attach $\langle \sigma_{b_1}, \dots, \sigma_{b_2}\rangle$ to Q_{e_a} as the new set-up for a disagreement at z. Lemma 1 will establish that such a sequence exists.

Note that the sets of instructions above for maintaining the inf cycle through a coding action or the appearance of a newly convergent computation make only the following transitions among phases possible:



Also, if $x \in$ *Add-*$i_s \cup$ *Add-*$j_s \cup$ *Remove-*$i_s \cup$ *Remove-*j_s, then $x \in K_s - K_{s'}$, where $s' < s$ is the largest such that we were in Phase A at stage s'.

Further note that if $f(\langle x, \langle a, l\rangle\rangle, s) \neq f(\langle x, \langle a, l\rangle\rangle, s+1)$, then one of the following three conditions must hold:

(1) $x \in$ *Add-*l_s, and we have moved from a B-phase to a C-phase, or from a B-phase to Phase A.

(2) $x \in$ *Remove-*l_s, and we have moved from a C-phase to Phase A.

(3) $x = k_t$ and $s + 1 = 2t + 1$.

Thus, for any x, a, and l,

$$
\begin{aligned}
&|\{s : f(\langle x, \langle a, l\rangle\rangle, s+1) \neq f(\langle x, \langle a, l\rangle\rangle, s) \text{ and } Q_{e_a} \text{ was active at stage}\\
&\quad s+1 \text{ and we were in the inf cycle for } Q_{e_a} \text{ for } z \text{ at stage } s+1\}|\\
&\quad \leq 1 \quad \text{(for the transition out of the } A\text{-phase)}\\
&\quad + 1 \quad \text{(for the transition out of a } B\text{-phase)}\\
&\quad + 1 \quad \text{(for the transition out of a } C\text{-phase)}\\
&\quad + 1 \quad \text{(for the (potential) stage when } x = k_t)\\
&\quad \leq 4.
\end{aligned}
$$

We use $p(\langle e, a\rangle, s)$ to denote the restraint imposed by the requirement P_{e_a} at stage s and $q(\langle e, a\rangle, s)$ to denote the restraint imposed by the requirement Q_{e_a} at stage s. If a requirement imposes no restraint, then the respective function has the value -1.

Now $R(\langle e, a\rangle, s)$, the maximum of all restraints imposed by requirements of priority higher than $\langle e, a\rangle$, can be more precisely defined by

$$
\begin{aligned}
R(\langle e, a\rangle, s) = \max(&\{p(\langle e', a'\rangle, s) : \langle e', a'\rangle < \langle e, a\rangle\}\\
&\cup \{q(\langle e', a'\rangle, s) : \langle e', a'\rangle < \langle e, a\rangle\}).
\end{aligned}
$$

Fix a recursive priority ordering of the requirements so that if $\langle e', a'\rangle < \langle e, a\rangle$, then any requirement with index $\langle e', a'\rangle$ has higher priority than any requirement with index $\langle e, a\rangle$.

We say that P_{e_a} requires attention at $s+1 = 2t$ through y if $p(\langle e, a\rangle, s) = -1$, $y \in D(\langle e, a\rangle)$, the $\langle e, a\rangle$-status of y is $\leq g(y) - 1$, y is not yet frozen for P_{e_a} at $s+1$, $y > R(\langle e, a\rangle, s)$, $f(y, s) = r(a)_1$, and $\Phi_e[A_{j_a, s}](\langle y, \alpha_{r(a)_2(i_a)}\rangle)\downarrow = 0$.

P_{e_a} requires attention at $s+1 = 2t+1$ if $k_t < p(\langle e, a\rangle, s)$ and $k_t > \langle e, a\rangle$.

Q_{e_a} can require attention in one of 3 ways:

Q_{e_a} requires attention at $s+1 = 2t$ to set up a disagreement at z if there is a sequence $\langle \sigma_1, \dots, \sigma_w\rangle$ which is a set-up for a disagreement for Q_{e_a} at z at stage $s+1, q(\langle e, a\rangle, s) = -1$, and for each $b \leq w$ and each $x \leq |\sigma_b|$, if $\sigma_b(x) \neq f(x, s)$ then the $\langle e, a\rangle$-status of x at stage s is 0. (If $w = 1$ then we enter the inf cycle directly.)

Q_{e_a} requires attention at $s+1 = 2t$ to maintain a set-up for a disagreement if $q(\langle e, a\rangle, s) \neq -1$, but the attached sequence $\langle \sigma_1, \dots, \sigma_w\rangle$ is no longer a set-up for a disagreement. This includes the case where we need to maintain a phase of the inf cycle, since when $w = 1$ a set-up for a disagreement is just a phase of the inf cycle.

Q_{e_a} requires attention at $s+1 = 2t+1$ if $k_t < q(\langle e, a\rangle, s)$ and $k_t > \langle e, a\rangle$.

We say that a requirement Q_{e_a} is *injured at stage* $s+1$ if Q_{e_a} is not active at stage $s+1$ and there is a number $x < q(\langle e, a\rangle, s)$ such that $f(x, s) \neq f(x, s+1)$.

We say that a requirement P_{e_a} is *injured at stage* $s+1$ if P_{e_a} is not active at stage $s+1$ and there is a number $x < p(\langle e,a\rangle,s)$ such that $f(x,s) \neq f(x,s+1)$.

A requirement's action becomes *cancelled* by resetting its restraint function to be -1. Our priority system for handling requirements in the construction below will guarantee that whenever a requirement is injured its action becomes cancelled.

Construction.

Stage 0. For $x \in D(\langle e,a\rangle)$, set $f(x,0) = r(a)_1$. For $x = \langle y,a\rangle$, where $a \in [1,h]$, set $f(x,0) = r(a)_1$. For $x = \langle y,\langle a,k\rangle\rangle$, where $a \in [h+1,l]$ and $k \in \{0,1\}$, set $f(x,0) = r(a,k)_1$. Set $p(\langle e,a\rangle,0) = q(\langle e,a\rangle,0) = -1$ for all $e \in \omega$ and $a \in [1,m]$.

Stage $s+1=2t$. Let R be the highest priority requirement requiring attention at stage $s+1$. If there is no such R, then take no action.

(a) If $R = P_{e_a}$ requires attention through y, then define $f(y,s+1) = r(a)_2$ and $p(\langle e,a\rangle,s+1) = \varphi_e(\langle y,\alpha_{r(a)_2}(i_a)\rangle)$.

(b) If $R = Q_{e_a}$ requires attention through z and $\langle\sigma_1,\dots,\sigma_w\rangle$ to establish a set-up for a disagreement, then define $f_{s+1} = f_s(\sigma_1)$ and $q(\langle e,a\rangle,s+1) = \varphi_e(z)$.

(c) If $R = Q_{e_a}$ requires attention to maintain a set-up for a disagreement at z, or a phase of the inf cycle for Q_{e_a} then follow the appropriate set of instructions given before the construction.

In either case, for all $i > \langle e,a\rangle$, define $p(i,s+1) = q(i,s+1) = -1$. For all $i < \langle e,a\rangle$, define $p(i,s+1) = p(i,s)$ and $q(i,s+1) = q(i,s)$. For all y such that $f(y,s) \neq f(y,s+1)$, set $\langle e,a\rangle$-status of y at $s+1 = (\langle e,a\rangle$-status of y at $s)+1$, and, for all requirements R' of lower priority than R, announce that y is frozen for R'.

Stage $s+1=2t+1$ $(t \geq 0)$. Let R be the highest priority requirement requiring attention at stage $s+1$, if one exists. If such a requirement exists, then

(a) If $R = P_{e_a}$, define $f(\langle k_t,a\rangle,s+1) = r(a)_2$, $f(y,s+1) = f(y,s)$ for all other y.

(b) If $R = Q_{e_a}$, then follow the appropriate coding instructions for Q_{e_a} at $s+1$.

In either case, for all $i > \langle e,a\rangle$, define $p(i,s+1) = q(i,s+1) = -1$. For all $i < \langle e,a\rangle$, define $p(i,s+1) = p(i,s)$ and $q(i,s+1) = q(i,s)$. For all y such that $f(y,s) \neq f(y,s+1)$, set $\langle e,a\rangle$-status of y at $s+1 = (\langle e,a\rangle$-status of y at $s)+1$, and, for all requirements R' of lower priority than R, announce that y is frozen for R'.

If no requirement requires attention at stage $s+1$, define $f(\langle k_t,1\rangle,s+1) = r(1)_2$ and $f(y,s+1) = f(y,s)$ for all other y. For all $i \geq k_t$, define $p(i,s+1) = -1$ and $q(i,s+1) = -1$. For all $i < k_t$, define $p(i,s+1) = p(i,s)$ and $q(i,s+1) = q(i,s)$. For all i and y, the i-status of y at $s+1 =$ the i-status of y at stage s.

End of construction.

LEMMA 1. *Suppose that* $s+1 = 2t+1$ *and* $\langle\sigma_1,\dots,\sigma_w\rangle$ *was the set-up for a disagreement at* z *attached to* Q_{e_a} *at stage* s *and that this set-up has not been* *injured by stage s. Then there is a subsequence of $\langle \sigma_1, \dots, \sigma_w \rangle$ which is a set-up for a difference at z at stage $s+2 = 2t+2$.*

PROOF. If, for some b, $1 \le b \le w$ and $\Phi_{e,s}[A_{i_a,s}(\sigma_b)](z)\downarrow \ne \Phi_{e,s}[A_{j_a,s}(\sigma_b)](z)\downarrow$, then the subsequence $\langle \sigma_b \rangle$ satisfies the lemma. Otherwise we argue by cases, depending on which of (8.1)–(8.4) held for the original $\langle \sigma_1, \dots, \sigma_w \rangle$. Note that if this sequence has ceased being a set-up for a disagreement, but Q_{e_a} has not been injured, then each convergent computation in the original set-up remains unchanged. The only changes will be that divergent computations may become convergent.

Suppose that (8.1) or (8.4) is true of $\langle \sigma_1, \dots, \sigma_w \rangle$. Then $\Phi_{e,s}[A_{i_a,s}(\sigma_w)](z)\downarrow$ and differs from one of $\Phi_{e,s}[A_{i_a,s}(\sigma_1)](z)\downarrow$ or $\Phi_{e,s}[A_{j_a,s}(\sigma_1)](z)\downarrow$. Let $b_1 \le w$ be the greatest such that either

$$\Phi_{e,s}[A_{i_a,s}(\sigma_{b_1})](z)\downarrow \ne \Phi_{e,s}[A_{i_a,s}(\sigma_w)](z)\downarrow$$

or

$$\Phi_{e,s}[A_{j_a,s}(\sigma_{b_1})](z)\downarrow \ne \Phi_{e,s}[A_{i_a,s}(\sigma_w)](z)\downarrow .$$

Without loss of generality, say $\Phi_{e,s}[A_{i_a,s}(\sigma_{b_1})](z)\downarrow \ne \Phi_{e,s}[A_{i_a,s}(\sigma_w)](z)\downarrow$. Then it must be that $\Phi_{e,s}[A_{j_a,s}(\sigma_{b_1})](z)\uparrow$. This is true since it cannot converge and be different from $\Phi_{e,s}[A_{i_a,s}(\sigma_b)](z)$, and if it converged and were the same as $\Phi_{e,s}[A_{i_a,s}(\sigma_b)](z)$, this would violate either the maximality of b_1 among $\{1, \dots, w\}$, or $\langle \sigma_1, \dots, \sigma_w \rangle$ being a sequence of interpolants (condition (10)).

Let $b_2 > b_1$ be the least such that

$$\Phi_{e,s}[A_{i_a,s}(\sigma_{b_1})](z)\downarrow \ne \Phi_{e,s}[A_{i_a,s}(\sigma_{b_2})](z)\downarrow$$

or

$$\Phi_{e,s}[A_{i_a,s}(\sigma_{b_1})](z)\downarrow \ne \Phi_{e,s}[A_{j_a,s}(\sigma_{b_2})](z)\downarrow .$$

Such a b_2 must exist because $b_2 = w$ will work. Since b_2 is the least $> b_1$, it must be that either $\Phi_{e,s}[A_{i_a,s}(\sigma_{b_2})](z)\uparrow$ or $\Phi_{e,s}[A_{j_a,s}(\sigma_{b_2})](z)\uparrow$, and, for all b such that $b_1 < b < b_2$, $\Phi_{e,s}[A_{i_a,s}(\sigma_b)](z)\uparrow$ and $\Phi_{e,s}[A_{j_a,s}(\sigma_b)](z)\uparrow$.

Note that condition (7) still holds for $\langle \sigma_{b_1} \dots, \sigma_{b_2} \rangle$ because Q_{e_a}'s set-up was not injured, only changed by the appearance of new convergent computations. So the sequence $\langle \sigma_{b_1}, \dots, \sigma_{b_2} \rangle$ satisfies the lemma.

The verification for cases (8.2) and (8.3) is quite similar. It is included in detail in the proof of Lemma 2. □

LEMMA 2. *Suppose that $s+1 = 2t+1$ and $\langle \sigma_1, \dots, \sigma_w \rangle$ is a set-up for a difference at z at stage s attached to Q_{e_a} which has not been injured by stage s and $\langle \sigma'_1, \dots, \sigma'_w \rangle$ is defined as follows:*
For b such that $1 < b < w$, define $\sigma'_b = \sigma_b \oplus \langle k_t, \langle a, 1 \rangle \rangle \oplus \langle k_t, \langle a, 2 \rangle \rangle$.
If (8.1) holds then define $\sigma'_1 = \sigma_1 \oplus \langle k_t, \langle a, 2 \rangle \rangle$ and $\sigma'_w = \sigma_w \oplus \langle k_t, \langle a, 2 \rangle \rangle$.
If (8.2) holds then define $\sigma'_1 = \sigma_1 \oplus \langle k_t, \langle a, 1 \rangle \rangle$ and $\sigma'_w = \sigma_w \oplus \langle k_t, \langle a, 1 \rangle \rangle$.
If (8.3) holds then define $\sigma'_1 = \sigma_1 \oplus \langle k_t, \langle a, 2 \rangle \rangle$ and $\sigma'_w = \sigma_w \oplus \langle k_t, \langle a, 1 \rangle \rangle$.
If (8.4) holds then define $\sigma'_1 = \sigma_1 \oplus \langle k_t, \langle a, 1 \rangle \rangle$ and $\sigma'_w = \sigma_w \oplus \langle k_t, \langle a, 2 \rangle \rangle$.

Then there is a subsequence of $\langle\sigma_1',\dots,\sigma_w'\rangle$, $\langle\tau_1,\dots,\tau_p\rangle$ *which is a set-up for a disagreement at* z *for* Q_{e_a}.

Proof. If, for some b, $1\le b\le w$ and $\Phi_{e,s}[A_{i_a,s}(\sigma_b')](z)\downarrow\neq\Phi_{e,s}[A_{j_a,s}(\sigma_b')](z)\downarrow$, then $p=1$ and $\langle\tau_1\rangle=\langle\sigma_b'\rangle$. Otherwise, we argue by cases, depending on which of (8.1)–(8.4) holds.

Suppose that (8.1) or (8.4) is true of $\langle\sigma_1,\dots,\sigma_w\rangle$. Then $\Phi_{e,s}[A_{i_a,s}(\sigma_w')](z)\downarrow$ and differs from one of $\Phi_{e,s}[A_{i_a,s}(\sigma_1')](z)\downarrow$ or $\Phi_{e,s}[A_{j_a,s}(\sigma_1')](z)\downarrow$. Let $b_1\le w$ be the greatest such that either

$$\Phi_{e,s}[A_{i_a,s}(\sigma_{b_1}')](z)\downarrow\neq\Phi_{e,s}[A_{i_a,s}(\sigma_w')](z)\downarrow$$

or

$$\Phi_{e,s}[A_{j_a,s}(\sigma_{b_1}')](z)\downarrow\neq\Phi_{e,s}[A_{i_a,s}(\sigma_w')](z)\downarrow\,.$$

Without loss of generality, say $\Phi_{e,s}[A_{i_a,s}(\sigma_{b_1}')](z)\downarrow\neq\Phi_{e,s}[A_{i_a,s}(\sigma_w')](z)\downarrow$. Then it must be that $\Phi_{e,s}[A_{j_a,s}(\sigma_{b_1}')](z)\uparrow$, since $\langle\sigma_1',\dots,\sigma_w'\rangle$ is a sequence of interpolants, and there is no actual difference, and b_1 was maximal.

Let $b_2>b_1$ be the least such that

$$\Phi_{e,s}[A_{i_a,s}(\sigma_{b_1}')](z)\downarrow\neq\Phi_{e,s}[A_{i_a,s}(\sigma_{b_2}')](z)\downarrow$$

or

$$\Phi_{e,s}[A_{i_a,s}(\sigma_{b_1}')](z)\downarrow\neq\Phi_{e,s}[A_{j_a,s}(\sigma_{b_2}')](z)\downarrow\,.$$

Such a b_2 must exist because $b_2=w$ will work. Since b_2 is the least $>b_1$, it must be that either $\Phi_{e,s}[A_{i_a,s}(\sigma_{b_2}')](z)\uparrow$ or $\Phi_{e,s}[A_{j_a,s}(\sigma_{b_2}')](z)\uparrow$, and $\Phi_{e,s}[A_{i_a,s}(\sigma_b')](z)\uparrow$ and $\Phi_{e,s}[A_{j_a,s}(\sigma_b')](z)\uparrow$ for all b such that $b_1<b<b_2$.

Next we verify that condition (7) holds for the sequence $\langle\sigma_{b_1}',\dots,\sigma_{b_2}'\rangle$. If $x\neq\langle k_t,\langle a,i\rangle\rangle$, then the argument is just as in the proof of Lemma 1. Q_{e_a}'s initial set-up has not been injured, so the i-status of x has remained the same and no new freezing of numbers used by Q_{e_a} has occurred.

Now suppose $x=\langle k_t,\langle a,i\rangle\rangle$. Condition (7.1) is clear. (7.2) holds because Q_{e_a} is the requirement which received attention at this stage. If $x\le R(\langle e,a\rangle,s)$, then the coding of k_t injures some higher priority requirement, and so that requirement should have been the one to receive attention. (7.3) holds by the definition of requires attention. (7.4) and (7. 5) hold because k_t is being coded for the first time; hence its i-status for $i\le\langle e,a\rangle$ cannot have been changed yet and it cannot be frozen. Thus $p=b_2-b_1+1$, and $\langle\tau_1,\dots,\tau_p\rangle=\langle\sigma_{b_1}',\dots,\sigma_{b_2}'\rangle$ satisfies the lemma.

Next, suppose that (8.2) or (8.3) is true of $\langle\sigma_1,\dots,\sigma_w\rangle$. Then $\Phi_{e,s}[A_{j_a,s}(\sigma_w')](z)\downarrow$ and differs from one of $\Phi_{e,s}[A_{i_a,s}(\sigma_1')](z)\downarrow$ or $\Phi_{e,s}[A_{j_a,s}(\sigma_1')](z)\downarrow$. Let b_1 be the greatest $\le w$ such that either

$$\Phi_{e,s}[A_{i_a,s}(\sigma_{b_1}')](z)\downarrow\neq\Phi_{e,s}[A_{j_a,s}(\sigma_w')](z)\downarrow$$

or

$$\Phi_{e,s}[A_{j_a,s}(\sigma_{b_1}')](z)\downarrow\neq\Phi_{e,s}[A_{j_a,s}(\sigma_w')](z)\downarrow\,.$$

Without loss of generality, say $\Phi_{e,s}[A_{i_a,s}(\sigma_{b_1}')](z)\downarrow\neq\Phi_{e,s}[A_{j_a,s}(\sigma_w')](z)\downarrow$. Then it must be that $\Phi_{e,s}[A_{j_a,s}(\sigma'_{b_1})](z)\uparrow$, since $\langle\sigma'_1,\dots,\sigma'_w\rangle$ is a sequence of interpolants, and there is no actual difference, and b_1 was maximum.

Let $b_2 > b_1$ be the least such that

$$\Phi_{e,s}[A_{i_a,s}(\sigma'_{b_1})](z)\downarrow\neq\Phi_{e,s}[A_{i_a,s}(\sigma'_{b_2})](z)\downarrow$$

or

$$\Phi_{e,s}[A_{i_a,s}(\sigma'_{b_1})](z)\downarrow\neq\Phi_{e,s}[A_{j_a,s}(\sigma'_{b_2})](z)\downarrow .$$

Such a b_2 must exist because $b_2 = w$ will work. Since b_2 is the least $> b_1$, it must be that either $\Phi_{e,s}[A_{i_a,s}(\sigma'_{b_2})](z)\uparrow$ or $\Phi_{e,s}[A_{j_a,s}(\sigma'_{b_2})](z)\uparrow$, and $\Phi_{e,s}[A_{i_a,s}(\sigma'_b)](z)\uparrow$ and $\Phi_{e,s}[A_{j_a,s}(\sigma'_b)](z)\uparrow$ for all b such that $b_1 < b < b_2$. Condition (7) holds for $\langle\sigma'_{b_1},\dots,\sigma'_{b_2}\rangle$ for the same reasons that it did in the cases (8.1) or (8.4). Thus $p = b_2 - b_1 + 1$, and $\langle\tau_1,\dots,\tau_p\rangle = \langle\sigma'_{b_1},\dots,\sigma'_{b_2}\rangle$ satisfies the lemma.

LEMMA 3. (a) $\lim_s f(x,s)$ *exists.*

(b) $A_n \leq_{wtt} K$.

(c) A_0 *is recursive.*

(d) $p_i \leq p_j \Longrightarrow A_i \leq_{wtt} A_j$.

(e) $p_i \vee p_j = p_k \Longrightarrow A_i \oplus A_j \equiv_{wtt} A_k$.

(f) $K \leq_{wtt} A_n$.

PROOF. (a) We prove parts (a) and (b) together by showing that there is a recursive function g such that for all x, $|\{s : f(x,s)\neq f(x,s+1)\}|\leq g(x)$. This guarantees that $\lim_s f(x,s)$ exists, and, by the proof of the limit lemma, since $F(x) = \lim_s f(x,s)$, $F \leq_{wtt} K$. Since $A_n \leq_{wtt} F$, we get $A_n \leq_{wtt} K$. We show that $g(x) = (d+5)(p(x)+1)$ serves as a bound on the number of changes in $f(x,s)$, where $p(x)$ gives the priority of x, i.e.,

$$p(x) = \begin{cases} \langle e,a\rangle & \text{if } x\in D(\langle e,a\rangle)\\ y & \text{if } x\in W(y)\,. \end{cases}$$

$p(x)$ is the number of requirements whose action could cause $f(x,s)\neq f(x,s+1)$. Once a requirement of priority i causes $f(x,s)\neq f(x,s+1)$, x is frozen for all requirements of lower priority. Thus it suffices to check that each requirement of priority higher than $p(x)+1$ can cause at most $d+5$ changes in $F(x)$.

A requirement of the form P_{e_a} causes at most one change, for its diagonalization action. A requirement of the form Q_{e_a} may cause changes in several ways. If Q_{e_a} causes changes to $f(x,s)$, then it is because of a specific set-up for a disagreement attached to Q_{e_a}. (If the set-up is destroyed, a new set-up will make use of new numbers.) Call this set-up $\langle\sigma_1,\dots,\sigma_w\rangle$. By definition, $w\leq d+1$. Q_{e_a} may act to maintain its set-up (either in response to coding or in response to new information about computations converging). The action taken here is to look at a modified set-up which is a subsequence of $\langle\sigma_1,\dots,\sigma_w\rangle$, potentially modified by coding as well. The effect of the action will be to change f_s to $f_s(\sigma_b)$ for some b such that $1\leq b\leq w$. The modified sequence will be the same as the original, except possibly coding a new value k_t. Since $w\leq d+1$, at most $d+1$ such changes can be made. Coding action may also cause us to enter the inf cycle. By construction, each x can be involved in at most one inf cycle, so changes caused by passage through the inf cycle are at most 4 in number. Hence Q_{e_a} can cause at most $d+5$ changes.

(c) Since $p_0 \leq p_i$ for all $i \in [0, n]$, we may assume that $\alpha_r(0) = 0$ for all $r \in [0, d]$. Thus $A_0 = \{\langle x, \alpha_{f(x,0)}(0)\rangle : x \in \omega\} \cup \{\langle x, \alpha_{F(x)}(0)\rangle : x \in \omega\} = \{\langle x, 0\rangle : x \in \omega\}$, and is recursive.

(d) Suppose $p_i \leq p_j$. We give a wtt (in fact tt) reduction for computing A_i from A_j. To determine $A_i(z)$, first determine if z is of the form $z = \langle x, u\rangle$ for some $u \in [0, d]$. If not, then $z \notin A_i$. Otherwise, let $y_1, \dots, y_b$, $b \leq d$, be all of the values of y such that $\langle x, \alpha_y(j)\rangle \in A_j$. If for some $y \in \{y_1, \dots, y_b\}$ we have $u = \alpha_y(i)$, then $z \in A_i$. Otherwise, $z \notin A_i$.

(e) Suppose $p_i \vee p_j = p_k$. By (b), $A_i \leq_{wtt} A_k$ and $A_j \leq_{wtt} A_k$, so $A_i \oplus A_j \leq_{wtt} A_k$, so it suffices to check that $A_k \leq_{wtt} A_i \oplus A_j$. As in the proof of (b), if z is not of the form $z = \langle x, u\rangle$, where $u \leq d$, then $z \notin A_k$. So assume $z = \langle x, u\rangle$. By definition of A_k, $\langle x, u\rangle \in A_k \leftrightarrow u = \alpha_{f(x,0)}(k)$ or ($u \neq \alpha_{f(x,0)}(k)$ and $u = \alpha_{f(x)}(k)$, where $F(x) = \lim_s f(x, s)$). If $\alpha_{f(x,0)}(k) \neq \alpha_{f(x)}(k)$, then by the lattice representation of $\mathscr{L}$ either $\alpha_{f(x,0)}(i) \neq \alpha_{f(x)}(i)$ or $\alpha_{f(x,0)}(j) \neq \alpha_{f(x)}(j)$, so we get $\langle x, u\rangle \in A_k \leftrightarrow u = \alpha_{f(x,0)}(k)$ or $\exists r \leq d[u = \alpha_r(k)$ and $\langle x, \alpha_r(i)\rangle \in A_i$ and $\langle x, \alpha_r(j)\rangle \in A_j$ and $\forall z \leq d$ $((\langle x, z\rangle \in A_i \rightarrow z = \alpha_r(i)$ or $z = \alpha_{f(x,0)}(i))$ and $(\langle x, z\rangle \in A_j \rightarrow z = \alpha_r(j)$ or $z = \alpha_{f(x,0)}(j)))]$. This gives the desired wtt-reduction from $A_i \oplus A_j$.

(f) To see that $K \leq_{wtt} A_n$, we need only verify that our coding was successful. Whenever a number k_t was enumerated in K our construction acted to code it into A_n. The rearrangements made by the Q_{e_a} strategies all maintained the coding invariant; thus, for any x, $x \in K \iff \exists y \in W(x)$ $(\exists r \leq d)$ $[r \neq f(y, 0)$ and $\langle y, \alpha_r(n)\rangle \in A_n]$. Since $W(x)$ is a finite set, this is indeed a wtt reduction.

LEMMA 4. *Each requirement $R = P_{e_a}$ or $R = Q_{e_a}$ is active at most finitely often, and for each i, $\lim_s R(i, s)$ exists.*

PROOF. We use induction on the priority ordering. Fix a requirement of priority i. We assume that each requirement of priority higher than i is active at most finitely often, and for all $i' < i$, $\lim_s R(i' + 1, s)$ exists. Let s_0 be a stage such that no requirement of priority higher than i is active at any stage $s \geq s_0$ and $\forall s \geq s_0$, $R(i, s) = R(i, s_0)$.

Suppose first that the requirement of priority i is P_{e_a}. P_{e_a} acts at at most one even stage $s > s_0$. If such a stage exists, call it s_1. Otherwise let $s_1 = s_0$. Then for all $s \geq s_1$, $R(\langle e, a\rangle + 1, s) = R(\langle e, a\rangle + 1, s_1)$. Let $s_2 \geq s_1$ be such that $K \restriction R(\langle e, a\rangle + 1, s_1) = K_{s_2} \restriction R(\langle e, a\rangle + 1, s_1)$ and $K \restriction \langle e, a\rangle + 1 = K_{s_2} \restriction \langle e, a\rangle + 1$. Then P_{e_a} will not be active at any stage $s \geq s_2$, since coding after stage s_2 will not affect A below P_{e_a}'s restraint.

Next suppose that the requirement of priority i is Q_{e_a}. Q_{e_a} will be active at at most finitely many even stages $s > s_0$. This is the case since Q_{e_a} acts at most once (after injury to Q_{e_a} has stopped) to establish a set-up, and at later times only to maintain the established set-up. The actions to maintain a set-up occur only in response to divergent computations in the set-up becoming convergent. This can happen at most w times, where w is the length of the sequence comprising the set-up. Q_{e_a} could also become active to maintain a phase of the inf cycle in response to divergent computations becoming convergent with time. This can happen at most twice—we could move from phase B to phase C to phase A, or from phase B directly to phase A. Let $s_1 \geq s_0$ be the stage at which Q_{e_a} becomes active to establish a set-up, if such a stage exists; otherwise let $s_1 = s_0$. Then for all $s \geq s_1$, $R(\langle e,a\rangle + 1, s) = R(\langle e,a\rangle + 1, s_1)$. Let $s_2 \geq s_1$ be such that $K \upharpoonright R(\langle e,a\rangle + 1, s_1) = K_{s_2} \upharpoonright R(\langle e,a\rangle + 1, s_1)$ and $K \upharpoonright \langle e,a\rangle + 1 = K_{s_2} \upharpoonright \langle e,a\rangle + 1$. Then Q_{e_a} will not be active at any odd stages $s \geq s_2$. Q_{e_a} will be active at at most finitely many even stages $s \geq s_2$ (as discussed above)—until finitely many computations become (potentially) convergent.

Lemma 5. *Each requirement P_{e_a} is satisfied.*

Proof. Let s_0 be a stage after which neither P_{e_a} nor any requirement of priority higher than P_{e_a} acts and $\forall s \geq s_0$ $R(\langle e,a\rangle + 1, s) = R(\langle e,a\rangle + 1, s_0)$. Note that for $x \in D(\langle e,a\rangle)$, and $s > s_0$ if $f(x,s) \neq f(x,s_0)$, then P_{e_a} or some higher priority requirement must have been active at some stage s' with $s_0 \leq s' \leq s$. Since each such requirement is active at most finitely often, and $D(\langle e,a\rangle)$ is infinite, there is an $x_0 \in D(\langle e,a\rangle)$ such that $x_0 > R(\langle e,a\rangle + 1, s_0)$ and for all s, $f(x_0, s) = f(x_0, 0)$. Assume for a contradiction that $\Phi_e[A_{j_a}] = A_{i_a}$. Then it must be that $\Phi_e[A_{j_a}](\langle x_0, \alpha_{r(a)_2}(i_a)\rangle) = 0$, since $\langle x_0, \alpha_{r(a)_2}(i_a)\rangle \notin A_{i_a}$. Let $s_1 \geq s_0$ be such that $\Phi_{e,s_1}[A_{j_a,s_1}](\langle x_0, \alpha_{r(a)_2}(i_a)\rangle)\downarrow = 0$. Then either P_{e_a} will become active at stage $s_1 + 1$, contrary to our hypothesis, or $p(\langle e,a\rangle, s_1) \geq 0$. In this case, let $s_2 \leq s_1$ be the stage at which P_{e_a} was active and $p(\langle e,a\rangle, s_1)$ was set. Then P_{e_a} acted at stage s_2 and caused $\Phi_{e,s_2}[A_{j_a,s_2}](\langle x, \alpha_{r(a)_2}(i_a)\rangle) \neq A_{i_a,s_2}(\langle x, \alpha_{r(a)_2}(i_a)\rangle)$. This computation was not injured between stages s_2 and s_1 (since $p(\langle e,a\rangle, s_1) = p(\langle e,a\rangle, s_2) > -1$), and so $\Phi_e[A_{j_a}] \neq A_{i_a}$.

Lemma 6. *Each requirement Q_{e_a} is satisfied.*

Proof. Suppose that $\Phi_e[A_{i_a}] = \Phi_e[A_{j_a}] = Y$ is total. We compute $Y \leq_{wtt} A_{m_a}$. Let s_0 be a stage such that no requirement of priority higher than Q_{e_a} is active at any stage $s \geq s_0$, and $R(\langle e,a\rangle + 1, s) = R(\langle e,a\rangle + 1, s_0)$ for all $s > s_0$, and $K_{s_0} \upharpoonright R(\langle e,a\rangle + 1, s) = K \upharpoonright R(\langle e,a\rangle + 1, s)$. To compute $Y(z)$, let $s_1 > s_0$ be such that $A_{m_a,s_1} \upharpoonright \varphi_e(z) = A_{m_a} \upharpoonright \varphi_e(z)$ and $\Phi_{e,s_1}[A_{i_a,s_1}](z)\downarrow = \Phi_{e,s_1}[A_{j_a,s_1}](z)\downarrow$. Then $Y(z) = \Phi_e[A_{i_a,s_1}](z)$. We verify this below.

Claim 6.1. $\Phi_{e,s_1}[A_{i_a,s_1}](z)\downarrow = \Phi_e[A_{i_a}](z)$.

Proof. Let $s_2 \geq s_1$ be such that $A_{i_a,s_2} \upharpoonright \varphi_e(z) = A_{i_a,s} \upharpoonright \varphi_e(z)$ and $A_{j_a,s_2} \upharpoonright \varphi_e(z) = A_{j_a,s} \upharpoonright \varphi_e(z)$ and $A_{m_a,s_2} \upharpoonright \varphi_e(z) = A_{m_a,s} \upharpoonright \varphi_e(z)$ for all $s \geq s_2$. Then $\Phi_{e,s_2}[A_{i_a,s_2}](z) = \Phi_e[A_{i_a}](z)$, so it suffices to show that $\Phi_{e,s_1}[A_{i_a,s_1}](z) = \Phi_{e,s_2}[A_{i_a,s_2}](z)$.

Suppose not, for a contradiction. Define $\sigma = f_{s_1} \upharpoonright \varphi_e(z)$ and $\tau = f_{s_2} \upharpoonright \varphi_e(z)$. Note that, since $A_{m_a,s_1} \upharpoonright \varphi_e(z) = A_{m_a,s_2} \upharpoonright \varphi_e(z)$, $\alpha_{\sigma(x)}(m_a) = \alpha_{\tau(x)}(m_a)$ for all $x < |\sigma|$. So there is a $w \leq d+1$ such that for each $x < |\sigma|$ such that there exist $r_1^x, \dots, r_w^x$ such that for $b \leq w$, and b odd,

$$\alpha_{r_b^x}(i_a) = \alpha_{r_{b+1}^x}(i_a) \quad \text{and} \quad \alpha_{r_{b+1}^x}(j_a) = \alpha_{r_{b+2}^x}(j_a),$$

$$\text{and} \quad r_1^x = \sigma(x) \quad \text{and} \quad r_b^x = \tau(x).$$

Define a sequence $\langle \sigma_1, \dots, \sigma_w\rangle$ by $\sigma_b(x) = r_b^x$ if $\sigma(x) \neq \tau(x)$, $\sigma_b(x) = \tau(x)$ otherwise. Note that since $\sigma_1 = \sigma$ and $\sigma_w = \tau$, any subsequence of $\langle \sigma_1, \dots, \sigma_w\rangle$ satisfies (5), (6), and (10). Next we check that for every $b \leq w$, σ_b satisfies (7.2)–(7.5) at stage $s_2 + 1$. Note that if $\sigma_b(x) \neq f(x, s_2)$, then it must be the case that $f(x, s_1) \neq f(x, s_2)$. So there must be some s, $s_1 \leq s < s_2$, such that some requirement R was active at stage $s+1$ and caused $f(x,s) \neq f(x,s+1)$. Since $s \geq s_1 \geq s_0$, it must be that R has priority lower than $\langle e,a\rangle$ (say i), and $R(i,s) \geq R(\langle e,a\rangle+1,s) = R(\langle e,a\rangle+1,s_2) \geq R(\langle e,a\rangle,s)$. If R was active at $s+1$, then it must be that $x > R(i,s)$, so (7.2) holds. Also, $p(x) \geq i > \langle e,a\rangle$, so (7. 3) holds. Also, for all $i' < i$, the i'-status of x at $s+1$ must be 0. Since the i'-status cannot change unless the requirement of priority i' is active and no requirement of priority above $\langle e,a\rangle$ is active after $s+1$, we get that the i'-status is still 0 at stage s_2, if $i' < \langle e,a\rangle$. Likewise, the $\langle e,a\rangle$-status is also 0 at s_2. Now consider condition (7.5). We know that x was not frozen for Q_{e_a} at stage s_1. If x is frozen for Q_{e_a} at stage s_2, then it must be the case that some requirement of priority higher than Q_{e_a} was active at some stage s with $s_1+1 \leq s_2$. But we have assumed that no requirement of priority higher than Q_{e_a} is active at any stage $s \geq s_0$, and $s_0 \leq s_1$. Hence (7.5) is satisfied.

Now there are two cases to consider. First, for some $b \leq d$,

$$\Phi_{e,s_2}[A_{i_a,s_2}(\sigma_b)](z)\downarrow \neq \Phi_{e,s_2}[A_{j_a,s_2}(\sigma_b)](z)\downarrow .$$

We would like to claim that defining $f_{s_2+1} = f_{s_2}(\sigma_b)$ would allow us to enter the inf cycle for Q_{e_a}, thus causing Q_{e_a} to require attention at stage s_2+1, a contradiction. There are two obstacles to this. The first is that $q(\langle e,a\rangle, s_2)$ might be greater than -1. This means that Q_{e_a} was active at some stage prior to s_2, say $s_3 < s_2$, and was not injured between stages s_3 and s_2. But in this case Q_{e_a} has a set-up for a disagreement attached which is never injured. If $\Phi_e[A_{i_a}]$ and $\Phi_e[A_{j_a}]$ are total, then Q_{e_a} will continue to require attention until an actual disagreement is established, and so $\Phi_e[A_{i_a}] \neq \Phi_e[A_{j_a}]$.

The other obstacle is that σ_b might not maintain the coding invariant at s_2. The instructions for the inf cycle allow us to correct this. Let $C = \{x : f(y',s_2) \neq f(y',0)$ for some $y' \in W(x)$, but $f(\sigma_b)(y) = f(y,0)$ for all $y \in W(x)\}$. For each $x \in C$ in turn, follow the instructions of the inf cycle to code x to define a string τ_t and sets *Add*-i_t, *Remove*-j_t, *Add*-j_t, or *Remove*-i_t, for $t = 1,\dots,|C|$, so that $f_{s_2}(\tau_1)$ maintains the coding invariant for the first t elements of C, and takes us to some phase of the inf cycle for Q_{e_a} at z. Then $\langle \tau_{|C|}\rangle$ is a set-up for a difference at z for Q_{e_a} at stage s_2+1, so Q_{e_a} will become active at stage $s+1$, a contradiction. Note that defining $f_{s_2+1} = f(\tau_t)$ causes the $\langle e,a\rangle$-status of any x to increase by at most one.

Otherwise, for all b, $1 \leq b \leq w$, it is not the case that $\Phi_{e,s_2}[A_{i_a,s_2}(\sigma_b)](z)\downarrow \neq \Phi_{e,s_2}[A_{j_a,s_2}(\sigma_b)](z)\downarrow$. Then there must be some subsequence of $\langle \sigma_1,\dots,\sigma_w\rangle$, call it $\langle \sigma_1',\dots,\sigma_v'\rangle$, such that (8) holds for σ_1' and σ_v'. This is true because the original sequence was a sequence of interpolants, and

$$\Phi_{e,s_2}[A_{i_a,s_2}(\sigma_1)](z)\downarrow \neq \Phi_{e,s_2}[A_{i_a,s_2}(\sigma_w)](z)\downarrow .$$

The proof of Lemma 1 shows that there is some subsequence of $\langle \sigma_1',\dots,\sigma_v'\rangle$ which satisfies (8) and (9). We have already established that this subsequence must satisfy (5), (6), (7.2)–(7.4), and (10). We next need to modify this sequence so that it will maintain the coding invariant. Define $C = \{x : f(y',s_2) \neq f(y',0)$ for some $y' \in W(x)$, but for some $b \leq w$, $f_{s_2}(\sigma_b)(y) = f(y,0)$ for all $y \in W(x)\}$. For each $x \in C$

in turn, follow the instructions of Lemma 2 for coding x with respect to Q_{e_a} and the current subsequence of $\langle \sigma_1', \dots, \sigma_v' \rangle$. If at some point we enter the inf cycle, then we follow the appropriate instructions. The result will be a set-up for a difference at z for Q_{e_a} at $s_2 + 1$, which will cause Q_{e_a} to become active at stage $s_2 + 1$, a contradiction.

Therefore, it must be that $\Phi_{e,s_1}[A_{i_a,s_1}(\sigma)](z)\downarrow = \Phi_{e,s_2}[A_{i_a,s_2}(\tau)](z)\downarrow = \Phi_e[A_{i_a}](z) = Y(z)$.

REFERENCES

[1] R. G. Downey, Δ_2^0 *degrees and transfer theorems*, ***Illinois Journal of Mathematics***, vol. 31 (1987), pp. 419–427.

[2] ———, *D.r.e. degrees and the nondiamond theorem*, ***Bulletin of the London Mathematical Society***, vol. 21 (1989), pp. 43–50.

[3] R. G. Downey and J. B. Remmel, *Classification of degree classes associated with r.e. subspaces*, ***Annals Pure and Applied Logic***, vol. 42 (1989), pp. 105–125.

[4] R. L. Epstein, R. Haas, and R. Kramer, *Hierarchies of sets and degrees below* 0′, ***Logic year 1979–80***, (M. Lerman, J. Schmerl, and R. I. Soare, eds.), Lecture Notes in Mathematics, vol. 859, Springer–Verlag, Berlin, 1981, pp. 32–48.

[5] Y. L. Ershov, *The hierarchy of* Δ_2^0 *sets*, ***Proceedings of the Fourth International Congress of Logic, Methodology and Philosophy of Science***, North–Holland, Amsterdam, 1973, pp. 69–76.

[6] P. Fejer and R. Shore, *Embeddings and extensions in the r.e. tt- and wtt-degrees*, ***Recursion theory week, Proceedings Oberwolfach, 1984***, Lecture Notes in Mathematics, vol. 1141, Springer–Verlag, New York, 1985, pp. 121–140.

[7] P. Fejer and R. Shore, *A direct construction of a minimal recursively enumerable truth table degree*, ***Recursion theory week, Proceedings Oberwolfach, 1989***, Lecture Notes in Mathematics, vol. 1432, Springer–Verlag, New York, 1990, pp. 187–204.

[8] C. A. Haught, ***Turing and truth table degrees of generic and recursively enumerable sets***, Ph.D. Thesis, Cornell University, Ithaca, New York, 1985.

[9] ———, *Lattice embeddings in the r.e. tt-degrees*, ***Transactions of the American Mathematical Society***, vol. 301 (1987), pp. 515–535.

[10] C. A. Haught and R. A. Shore, *Undecidability and initial segments of the (r.e.) tt-degrees*, this Journal, vol. 55 (1990), pp. 987–1006.

[11] ———, *Undecidability and initial segments of the wtt-degrees below* 0′, ***Recursion theory week, Proceedings Oberwolfach, 1989***, Lecture Notes in Mathematics, vol. 1432, Springer–Verlag, New York, 1990, pp. 223–244.

[12] C. G. Jockusch and J. Mohrherr, *Embedding the diamond lattice in the recursively enumerable truth table degrees*, ***Proceedings of the American Mathematical Society***, vol. 94 (1985), pp. 123–128.

[13] G. N. Kobzev, *On tt-degrees of recursively enumerable Turing degrees*, ***Matematischeskiĭ Sbornik (Novaya Seriya)***, vol. 106 (1978), pp. 507–514; English translation, ***Mathematics of the USSR—Sbornik***, vol. 35 (1979), pp. 173–180.

[14] R. Ladner and P. Sasso, *The weak truth table degrees of recursively enumerable sets*, ***Annals of Mathematical Logic***, vol. 8 (1975), pp. 429–448.

[15] M. Lerman, ***Degrees of unsolvability***, Springer–Verlag, New York, 1983.

[16] S. S. Marchenkov, *The existence of recursively enumerable truth table degrees*, ***Algebra and Logic***, vol. 14 (1975), pp. 257–261.

[17] C. F. Miller, *Decision problems for groups—survey and reflections*, ***Algorithms and classification in combinatorial group theory*** (G. Baumslag and C. F. Miller, III, editors), Mathematical Sciences Research Institute Publications, vol. 23, Springer–Verlag, Berlin, 1992, pp. 1–59.

[18] P. Odifreddi, *Strong reducibilities*, ***Bulletin (New Series) of the American Mathematical Society***, vol. 4 (1981), pp. 37–86.

[19] ———, ***Classical recursion theory***, North–Holland, New York, 1989.

[20] P. Pudlák and J. Tůma, *Every finite lattice can be embedded into a finite partition lattice*, ***Algebra Universalis***, vol. 10 (1980), pp. 74–95.

[21] R. I. Soare, ***Recursively enumerable sets and degrees***, Springer–Verlag, New York, 1987.

MATHEMATICS DEPARTMENT
VICTORIA UNIVERSITY OF WELLINGTON
WELLINGTON, NEW ZEALAND

E-mail: rod.downey@vuw.ac.nz

DEPARTMENT OF MATHEMATICAL SCIENCES
LOYOLA UNIVERSITY CHICAGO
CHICAGO, ILLINOIS 60626

E-mail: cah@math.luc.edu