THE BULLETIN OF SYMBOLIC LOGIC
Volume 1, Number 2, June 1995

A GENERAL FRAMEWORK FOR PRIORITY ARGUMENTS

STEFFEN LEMPP AND MANUEL LERMAN

The degrees of unsolvability were introduced in the ground-breaking papers of Post [20] and Kleene and Post [7] as an attempt to measure the *information content* of sets of natural numbers. Kleene and Post were interested in the relative complexity of decision problems arising naturally in mathematics; in particular, they wished to know when a solution to one decision problem contained the information necessary to solve a second decision problem. As decision problems can be coded by sets of natural numbers, this question is equivalent to: Given a computer with access to an oracle which will answer membership questions about a set A, can a program (allowing questions to the oracle) be written which will correctly compute the answers to all membership questions about a set B? If the answer is yes, then we say that B is *Turing reducible* to A and write $B \leq_T A$. We say that $B \equiv_T A$ if $B \leq_T A$ and $A \leq_T B$. $\equiv_T$ is an equivalence relation, and $\leq_T$ induces a partial ordering on the corresponding equivalence classes; the poset obtained in this way is called the *degrees of unsolvability*, and elements of this poset are called *degrees*.

Post was particularly interested in computability from sets which are partially generated by a computer, namely, those for which the elements of the set can be enumerated by a computer. These sets are called (*recursively*) *enumerable*, as are their degrees. He showed [20] that the enumerable degrees have a least and greatest element, and asked whether there were other enumerable degrees. This problem, which became known as *Post's Problem*, was solved a decade later by Friedberg [5] and Mučnik [19], and their solutions introduced a new technique, the *priority method*, which is the subject of this paper.

Received August 17, 1994; revised December 1, 1994.

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

The second author's research was partially supported by National Science Foundation Grants DMS-8521843, DMS-8900349, and DMS-9200539, and the Mathematical Sciences Research Institute.

The authors wish to express their gratitude to C. Ash, M. Groszek, L. Harrington, J. Knight, A. Kučera, R. Shore, and T. Slaman for helpful comments.

1079-8986/95/0102-0003/$2.30

The degrees of unsolvability form an algebraic structure which is induced by the information content of sets. One motivation for studying this structure is to see how closely this algebraic structure recaptures information content. Thus we would like to know if sets with different information content give rise to degrees which look different inside the structure, i.e., is it the case that the degrees have no non-trivial automorphisms? Many people have worked on this problem, and combined results of Shore [24], Slaman and Woodin [25], and Cooper [4], [3], show that there are few, if any, non-trivial automorphisms, and any such automorphism must move degrees to nearby degrees. Other research has pursued answers to similar questions about substructures of the degrees such as the enumerable degrees, and has investigated the relationship of these substructures to arithmetic. Attempts at answering these and other global questions about degree structures have required a careful analysis of the algebraic structure, and one of the key techniques for obtaining structure theorems, especially for the enumerable degrees, has been the priority method. As the complexity of the structural questions increased, more powerful variants of the priority method were developed to try to answer these questions. The framework which we will describe in this paper encompasses the techniques used to prove the individual structure theorems.

The priority method has been applied outside degree theory, e.g. in the study of the lattice of enumerable sets by Soare, Maass, Harrington, and many others; and in effective model theory by Nerode, Remmel, Ash, Knight, and others. There have been several important applications outside computability theory. Among these are Martin's [18] original proof of the Axiom of Borel Determinacy, Solovay's [27] characterization of the degrees of models of true arithmetic, and Slaman's result that ACA_0 is not conservative over RCA_0+ Ramsey's Theorem for Pairs. It is our hope that the framework which we present will provide a deeper understanding of the nature of the priority method; and that this understanding will aid in identifying classes of problems for which the priority method is useful for finding solutions. (We envision this to be similar to the way forcing provides a very useful framework for set theorists.)

The priority method can be viewed as an effective version of the Cantor diagonal argument. One wishes to carry out an effective construction whose result is the satisfaction of an infinite list $\{R_i : i \in N\}$ of requirements. A typical requirement will have the form $(\varphi \to \psi)$ & $(\neg\varphi \to \theta)$. We call φ the *directing sentence*. If φ is true, then we want to carry out *validated action* ψ, and if φ is false, then we want to carry out *activated action* θ. The need for effectiveness precludes the use of a Cantor diagonal argument, since we may not be able to effectively determine the truth of the directing sentence for a given requirement. The priority method provides schemes for carrying out action based on the truth of effectively generated guesses about the directing sentence. As action for one requirement may conflict with the ultimate truth of the action already carried out for another requirement, there is also a need to organize the construction to resolve such conflicts. The priority method produces ways of assigning priority to attempts at satisfying requirements, and having this priority organize the action carried out for the guesses at the truth of directing sentences.

The simplest priority method is called the *finite injury priority method*, and is the method discovered by Friedberg [5] and Mučnik [19]. In this case, the portion, σ, of the directing sentence determining the type of action to be taken is existential, so corresponds to an enumerable condition. While waiting to discover the truth of σ for a requirement R of high priority, action is taken for lower priority requirements under the assumption that σ is not true. Upon discovery of the truth of σ, action is taken for R which may *injure* the ultimate truth of the directing sentence or action already taken for requirements of lower (but not higher) priority than R, and new attempts at satisfying the lower priority requirements are begun which are compatible with the preservation of the truth of the directing sentence and the action taken for R.

As directing sentences become more complicated, their ultimate truth cannot be discovered in an enumerable fashion. What is required is a decomposition of the sentence into fragments of lower complexity, and a use of the truth of such fragments as a way both to generate action, and to guess at the truth of the full sentence. At the next level of quantifier complexity, the corresponding method has been called *infinite injury*, and was discovered by Sacks [21], Shoenfield [22], and Yates [28] and developed primarily by Sacks. The level after that was initially called *monstrous injury*, and was discovered by Lachlan [14]. Harrington introduced a nicer classification of the levels of complexity of priority arguments in terms of the degree of the oracle needed to determine how each requirement is satisfied; thus for each $n \in \mathbf{N}$, the $\mathbf{0}^{(n)}$-*priority method* is one which requires a $\mathbf{0}^{(n)}$-oracle to unravel the construction in this way. (So a finite injury argument is a $\mathbf{0}'$-argument, an infinite injury argument is a $\mathbf{0}''$-argument, etc.)

The need to use priority arguments of higher levels of complexity to answer structural questions about the enumerable degrees becomes evident as the quantifier complexity of the question increases. Such questions naturally occur when they involve iterates of the jump operator. The development of the framework presented in this paper was motivated by our proof of the decidability of the existential theory of the enumerable degrees in the language of least and greatest element and predicates $\leq_n$ for $n \in \mathbf{N}$, where $\mathbf{a} \leq_n \mathbf{b}$ iff $\mathbf{a}^{(n)} \leq \mathbf{b}^{(n)}$. The decision procedure required us to show that a sentence of this language is true of the enumerable degrees iff it is consistent with the poset axioms, and the strictly increasing and order-preserving properties of the jump operator. The proof of the latter result uses the $\mathbf{0}^{(n)}$-priority method for all n.

Attempts at finding frameworks for the priority method, or fragments thereof, were undertaken early in the development of the method. Each framework was viewed as a way to capture the common combinatorics of a given class of priority arguments, and so avoid redundancy in proofs using such arguments. Sacks [21] provided such a framework for finite injury, and Lachlan made some early attempts for infinite injury as well, taking both a game-theoretic approach [12], and an effective Baire category approach [13]. Yates [29] developed an approach for combining infinite injury arguments with effective perfect closed set forcing in the setting of the degrees below $\mathbf{0}'$, using Banach-Mazur games to model the constructions. Subsequent frameworks for priority arguments of restricted complexity have been developed by Shoenfield [23] using a tree of strategies, and Kontostathis [10], [11], and [9] using an effective Baire category approach. Harrington was the first to find a general way to approach priority arguments at all arithmetical levels, and accomplished this by combining the tree of strategies approach with the use of the Kleene Fixed-Point Theorem. Ash [1], [2], and Knight [8] have developed various frameworks which apply to restricted classes of problems in effective model theory using trees of enumerations, but cover all hyperarithmetic levels of complexity of the priority argument. Another approach, using trees of trees, has been introduced by Groszek and Slaman [6]. The latter two approaches have influenced the development of our framework, which is substantially developed in [15], and will be fully developed in [16]. This framework takes an inductive approach and uses a separate tree of strategies for each level of the induction. While it has been developed only for finite levels of priority through $\mathbf{0}^{(\omega)}$, it seems amenable to extension to hyperarithmetic levels.

The framework which we have developed [15], [16] is built on the *tree of strategies* approach to priority arguments which was introduced by Lachlan [14], developed by Harrington, and popularized by Soare [26]. However, we use a sequence of trees of strategies rather than a single tree. We begin by assigning requirements of high quantifier complexity to a tree T^n of level n for some n. We then require a level-by-level decomposition of the requirement into fragments which are assigned to nodes of lower level trees. This procedure ends when we reach T^0, the level at which an effective construction takes place. It must be shown that the satisfaction of the fragments of the requirements which trace their heredity to the true paths through all trees will ensure the satisfaction of the high-level requirement assigned to a node of T^n.

$$\begin{tikzcd}
\langle P_0, \leq_0\rangle \arrow[r, "f_1"] \arrow[d, "\equiv"'] & \langle P_1, \leq_1\rangle \arrow[r, "f_2"] \arrow[d, "\equiv"'] & \cdots\cdots \arrow[r, "f_3"] & \langle P_m, \leq_m\rangle \arrow[d, "\equiv"'] \\
\langle G_0, \leq\rangle \arrow[r, "\prime"] \arrow[d, "\subseteq"'] & \langle G_1, \leq\rangle \arrow[r, "\prime"] \arrow[d, "\subseteq"'] & \cdots\cdots \arrow[r, "\prime"] & \langle G_m, \leq\rangle \arrow[d, "\subseteq"'] \\
\mathcal{R} \arrow[r, "\prime"] & \mathcal{R}(\mathbf{0}') \arrow[r, "\prime"] & \cdots\cdots \arrow[r, "\prime"] & \mathcal{R}(\mathbf{0}^{(m)})
\end{tikzcd}$$

FIGURE 1

The paper is organized as follows. In Section 1, we illustrate the basic features of the framework using the example of the above-mentioned decidability result [15]. In Section 2, we indicate how these ideas can be used to formulate a general framework and discuss the ways in which we expect the framework to be useful for proving additional results.

§1. An example. We illustrate the major ideas of our framework using the example of the above-mentioned decidability result by the authors [15]. We begin with some definitions. Let $\mathcal{R} = \langle \mathbf{R}, \leq, 0, 0' \rangle$ be the poset of enumerable degrees with least element $\mathbf{0}$ and greatest element $\mathbf{0}'$. For each $m \in \mathbf{N}$ and $\mathbf{a}, \mathbf{b} \in \mathbf{R}$, we define

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

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

$$\langle P_0, \leq_0, P_1, \leq_1, f_1, \dots, P_m, \leq_m, f_m \rangle$$

such that for each $k < m$, $\langle P_k, \leq_k, P_{k+1}, \leq_{k+1}, f_{k+1} \rangle$ is a jump poset. The following structure theorem is proved in [15]. Since this theorem implies that any existential statement not excluded by the trivial properties of Turing reducibility and Turing jump can be realized, it follows easily that the existential theory of the enumerable degrees in the language of least and greatest element and the predicates $\leq_n$ is decidable.

THEOREM. *Fix $n \in \mathbf{N}$, and let $\langle P_0, \leq_0, P_1, \leq_1, f_1, \dots, P_m, \leq_m, f_m \rangle$ be a finite m-jump poset such that P_0 has least element* 0 *and greatest element* 1. *Then there is a finite set $\mathbf{G}_0$ of enumerable degrees, and there are finite* *sets* $\mathbf{G}_k = \{\mathbf{d} : \exists \mathbf{a} \in \mathbf{G}_0(\mathbf{a}^{(k)} = \mathbf{d})\}$ *for each* $k \in [1, m]$ *such that the diagram in Figure* 1 *commutes. Furthermore, the embedding maps* $0 \in P_0$ *to* $\mathbf{0}$ *and* $1 \in P_0$ *to* $\mathbf{0}'$. (In fact, the proof of this theorem can easily be extended to all countable $< \omega$-jump posets.)

The proof of the theorem uses a $\mathbf{0}^{(n)}$-priority argument to show that a certain set of requirements can be satisfied. The embedding maps $c \in P_0$ to the degree of the enumerable set A_c (where we set $A_0 = \emptyset$ and $A_1 = \emptyset'$ for the least and greatest element 0 and 1 of P_0). Inductively define the function g_k by: g_0 is the identity, and for $k > 0$, $g_k = f_k \circ g_{k-1}$. There are three types of requirements to be satisfied.

Incomparability requirements ensure that if $g_k(c) \not\leq g_k(b)$ then $A_c^{(k)} \not\leq_T A_b^{(k)}$. Using an iterated version of Shoenfield's Limit Lemma, we will build a functional Δ computable in A_c whose k-fold iterated limit does not equal the k-fold iterated limit of any functional Φ_e computable in A_b. To make this precise, we must ensure

(1) $\Delta(A_c)$ is total & $\forall x(\lim_{u_1} \ldots \lim_{u_k} \Delta(A_c; u_1, \ldots, u_k, x)\downarrow)$, and

(2) $\exists x(\lim_{u_1} \ldots \lim_{u_k}(\Delta(A_c; u_1, \ldots, u_k, x) \neq \lim_{v_1} \ldots \lim_{v_k} \Phi_e(A_b; v_1, \ldots, v_k, x))$

for all e. (1) will follow from (2) and properties of the construction; so we will not have explicit strategies for it on our trees. To satisfy (2), we fix a distinct x whenever this requirement is assigned to a node $\alpha^k \in T^k$.

The basic module for satisfying (2) (for $k = 0$) is just the Friedberg-Mučnik argument, which we recast as follows in order to allow later generalizations: We first fix a diagonalization witness x. The directing sentence is

(3) $\exists s \forall t \geq s(\Phi_e(A_b; x)[s]\downarrow = 0 \;\&\; A_b \upharpoonright u(A_b; e, x)[s] = A_{b,t} \upharpoonright (u(A_b; e, x)[s]))$.

(Formally, this is a Σ_2-sentence; however, the inner universal quantifier over stages can be ignored by preserving the set A_b once a computation has been found, so we will treat (3) as a Σ_1-sentence.)

We now define $\Delta(A_c; x) = 0$ with some big use $\delta(x)$. As long as the directing sentence appears false (i.e., as long as no witness s has been found for (3)), we are done. Once (3) appears true, we reset $\Delta(A_c; x) = 1$ (after enumerating $\delta(x)$ into A_c) and try to preserve the truth of the directing sentence by restraining A_b.

The basic module for $k = 1$ is now essentially an ω-sequence of basic modules for $k = 0$: The directing sentence will be

(4) $$\exists^\infty v \exists s \forall t \geq s(\Phi_e(A_b; v, x)[t]\downarrow = 0),$$

which is equivalent to

$$\forall v \exists v' \geq v \exists s \forall t \geq s(\Phi_e(A_b; v', x)[t]\downarrow = 0). \tag{5}$$

(By the same remark as for (3), we can treat this as a Π_2-sentence.) We can now split up (5) into directing subsentences by bounding the outermost quantifier on v by some v_0, say. A substrategy working with this directing subsentence will perform subaction by first defining $\Delta(A_c; u, x) = 0$ for all $u \leq$ some u_0, and later, when the directing subsentence appears true, if ever, resetting $\Delta(A_c; u, x) = 1$ for these u. There are now two possibilities for satisfying this requirement. If (5) really holds then every substrategy will eventually reset $\Delta(A_c; u, x) = 1$ for its values u; otherwise, there will be some last substrategy (working with a sufficiently large v_0) that never finds a pair of witnesses (v', s) for its directing subsentence, and this substrategy will define $\Delta(A_c; u, x) = 0$ for almost all u.

The cases $k > 1$ are now treated similarly, using an inductive argument.

The second type of requirement which must be satisfied ensures comparability of kth jumps (for $k > 0$) if the smaller of the kth jumps is not to be the kth jump of $\mathbf{0}'$, i.e., $\mathbf{0}^{(k+1)}$. (We do not consider the case $k = 0$ here since it can be satisfied by direct coding. We treat the kth jump of $\mathbf{0}'$ separately, in the third type of requirement below, since we cannot restrain the set $\emptyset'$, and so the complexity of the directing sentence has to be computed differently.) The comparability requirements thus ensure that if $g_k(b) \leq g_k(c)$, then $A_b^{(k)} \leq_T A_c^{(k)}$. Again using an iterated version of Shoenfield's Limit Lemma, we will build a functional Δ computable in A_c whose k-fold iterated limit equals $A_b^{(k)}$. Thus we must ensure (1) and, for all e,

$$\lim_{u_1} \dots \lim_{u_k} \Delta(A_c; u_1, \dots, u_k, e) = \begin{cases} A_b^{(k)}(e) & \text{if } k \text{ is odd,} \\ \overline{A_b^{(k)}}(e) & \text{if } k \text{ is even.} \end{cases} \tag{6}$$

The basic module for $k = 1$ has directing sentence $e \in A_b'$, i.e.,

$$\exists s \forall t \geq s(\Phi_e(A_b; e)[s]\downarrow), \tag{7}$$

which we may (again by preservation) treat as a Σ_1-sentence. We now define $\Delta(A_c; u, e) = 0$ for larger and larger u until, if ever, (7) appears true, at which time we start setting $\Delta(A_c; u, e) = 1$ for all subsequent u. The crucial point is that if (7) later becomes false (due to injury) then, almost always, the number entering A_b will also enter A_c and thus allow us to reset $\Delta(A_c; u, e)$ from 1 back to 0. (We call this feature *automatic correction*.)

The cases $k > 1$ are again handled by induction as for the incomparability requirements.

The third type of requirement which must be satisfied preserves comparability of kth jumps if the smaller of the kth jumps is the kth jump of $\mathbf{0}'$, i.e., $\mathbf{0}^{(k+1)}$. These highness requirements ensure that if $g_k(1) \leq g_k(c)$, then $\emptyset^{(k+1)} \leq_T A_c^{(k)}$. As for the second type of requirement, we thus must satisfy (1) and, for all e,

$$(8) \qquad \lim_{u_1} \dots \lim_{u_k} \Delta(A_c; u_1, \dots, u_k, e) = \begin{cases} \emptyset^{(k+1)}(e) & \text{if } k \text{ is even,} \\ \overline{\emptyset^{(k+1)}}(e) & \text{if } k \text{ is odd.} \end{cases}$$

The basic module for satisfying this requirement is the same for incomparability requirements, except that we now do not have the ability to preserve computations relative to $A_b = \emptyset'$; so we lose the ability to preserve computations, and (7), for example, has to be treated as a Σ_2-sentence.

We define the *dimension* of a requirement as the quantifier-complexity of its directing sentence (modulo preservability as mentioned above). So incomparability and highness requirements have dimension $k+1$ while comparability requirements have dimension k.

Our construction takes place on a finite number of *trees of strategies*, which we define by setting

$$(9) \qquad T^0 = \{0, \infty\}^{<\omega}, \text{ and}$$
$$(10) \quad T^{k+1} = \{\sigma \in (T^k)^{<\omega} \mid \forall i, j < \mathrm{lh}(\sigma)(i < j \to \sigma(i) \subset \sigma(j))\}.$$

The intuition is the following: On T_0, a node can have outcome 0 (denoting that the directing subsentence was found by the node to be false) and ∞ (denoting that it was found to be true). On T^{k+1}, the outcome $\xi^k \in T^k$ of a node $\sigma^{k+1} \in T^{k+1}$ denotes that $\sigma^k = (\xi^k)^-$ is a substrategy of the strategy σ^{k+1} and that the outcome of σ^k also gives the "final" outcome of σ^{k+1}, namely, that either σ^k has found the witness to the directing sentence of σ^{k+1}, or that σ^k is a node of minimal length working for σ^{k+1} and that neither σ^k nor any of its extensions working for σ^{k+1} find a witness for σ^{k+1}. (We denote by $\mathrm{up}(\sigma^k) = \sigma^{k+1}$ the fact that σ^k is a substrategy of σ^{k+1}.)

We next define a function $\lambda : T^k \to T^{k+1}$, denoting that a node $\eta^k \in T^k$ guesses that the true path (coding the correct outcomes) through T^{k+1} extends the node $\lambda(\eta^k)$. The node $\lambda(\eta^k)$ is defined by induction as follows: Given that $\sigma^{k+1} \subseteq \lambda(\eta^k)$ (where $|\sigma^{k+1}| = n$, say), we specify

$$(11) \quad (\lambda(\eta^k))(n) = \begin{cases} \xi^k & \text{if } (\xi^k)^- \text{ finds a witness for } \sigma^{k+1} \\ & \qquad \text{and } \xi^k \subseteq \eta^k \text{ codes this outcome,} \\ \mu^k & \text{if } \mu^k \text{ is the least } \sigma^{k+1}\text{-substrategy} \subset \eta^k, \\ & \qquad \text{but none finds a witness,} \\ \uparrow & \text{if there is no substrategy} \subset \eta^k \\ & \qquad \text{working for } \sigma^{k+1}. \end{cases}$$

Once the third clause of (11) applies, the definition of the node $\lambda(\eta^k)$ is complete. The λ-function naturally extends to a function $\lambda : T^k \to T^{k+1}$

on infinite paths. (We will denote by Λ^k the *true path* of the construction on T^k, which will compute the outcome of each node along it correctly according to the truth of the directing sentences. The true path Λ^0 of T^0 will be computable, while the paths $\Lambda^{k+1} = \lambda(\Lambda^k)$ will be computable in $\emptyset^{(k+1)}$.)

We will always maintain the condition

$$\text{up}(\eta^k) \subseteq \lambda(\eta^k), \tag{12}$$

i.e., any substrategy on T^k works for a node on T^{k+1} which it believes to lie along the true path. This is part of our condition of *consistency* on the way we determine $\text{up}(\eta^k)$. We omit the formal definition of this notion; besides (12), consistency requires essentially that we have not yet found a witness for $\text{up}(\eta^k)$ along η^k; and that $\text{up}(\eta^k)$ not be *restrained by a* $\lambda(\eta^k)$*-link* $[\mu^{k+1}, \tau^{k+1}]$, i.e., we do not want

$$\mu^{k+1} \subseteq \text{up}(\eta^k) \subset \tau^{k+1} \subset \lambda(\eta^k),$$

where τ^{k+1} has a different guess about the outcome of a node on a higher tree than μ^{k+1}. (Links are a very general concept, unifying the notions of initialization, links in the sense of Soare [26, Ch. XIV], and several others into one. It is possible to show that the absence of a link around $\text{up}(\eta^k)$ actually implies (12).)

The definition of our functionals Δ is now determined by a notion of *control*, deciding which strategy is allowed to define a functional at which arguments. The definition is rather straightforward for the incomparability requirements, since each strategy at the top level will work with a distinct diagonalization witness. For the comparability and highness requirements, however, we have, even at the top level, many incomparable strategies, all competing to define the same functional according to their (possibly conflicting) guesses about the truth of their directing sentences (as well as the effect of other strategies). The notion of control alone is not able to handle these conflicts on T^{k-1}, i.e., one level below the dimension of the requirement. Instead, we introduce the notion of *implication chain* to resolve these conflicts.

We illustrate the implication chain machinery in the following example. Suppose that we have $\sigma^{k-1} \subset \hat{\sigma}^{k-1} \subset \Lambda^{k-1} \in [T^{k-1}]$ such that σ^{k-1} and $\hat{\sigma}^{k-1}$ are trying to define the same functional on possibly the same argument. A problem will occur if there is no way to determine which of σ^{k-1} and $\hat{\sigma}^{k-1}$ is derived from nodes along the true paths for the construction (although $\hat{\sigma}^{k-1}$ seems to have this property based on the current approximation to the true paths), and σ^{k-1} and $\hat{\sigma}^{k-1}$ wish to make definitions with different values. We then delay this definition and investigate the process of returning σ^{k-1} and its antiderivatives to a later approximation to the true paths. There will be a minimum set of nodes of the top tree whose outcomes will have to be (forcibly) switched by switching the outcomes of some of their derivatives, regardless of the truth of their directing sentences. We determine whether such switches would injure the oracle set for any axioms we would currently declare, if we were to declare axioms. If the answer is yes (the *non-amenable* case), then it is safe to declare axioms; if σ^{k-1} returns to the true path, these axioms will not reflect computation from the final oracle, so σ^{k-1} will have the ability to declare new axioms. Otherwise (the *amenable* case), we start building an *implication chain* to resolve the conflicts. We first go through the process of returning σ^{k-1} to the true path without any functional definitions, and determining an outcome for the new derivative of either σ^{k-1} or $\mathrm{up}(\sigma^{k-1})$ obtained in this manner. If the outcome differs from that for σ^{k-1}, then we will have resolved our conflict on the new path, and can proceed to define axioms. Otherwise, we will have replicated our starting situation on T^{k-2}, and can now repeat the procedure described above. We continue until we either resolve the conflict by changing the outcome of one of the nodes, or reach T^0. The process is arranged to preserve implications between directing sentences in such a way that the outcomes are contradictory (i.e., one node claims to have found a witness below a certain bound while the other claims there is none below a higher bound). Once we reach T^0, this contradiction cannot occur (as the outcomes are now computable and we will have an implication between directing sentences), so the conflict can be resolved by changing the outcome of one of the nodes. Thus the construction must also validate $\hat{\sigma}^0$, resolving the conflict.

In using the implication chain machinery, there is a need to preserve implications between directing sentences, level-by-level, while preventing uncorrectable axioms from being declared. The machinery which ensures that this can be achieved is complex and delicate.

As can be seen from the above very rough description, there will be many nodes on our trees, even along the true path, which are not allowed to act, or determine their outcomes, according to the truth of their directing sentences. In the above, this can happen in two ways: The node may be restrained by a link, or it may be involved in bringing another node back on the truth path as part of the implication chain machinery. We need to verify that enough *critical nodes* remain (i.e., nodes which are allowed to act according to the truth of their directing sentence) so that we can argue that the substrategies of a strategy work together to ensure their strategy's success.

§2. The general framework and the framework theorem. In Section §1, we illustrated some basic features of the framework, and the properties which need to be verified to ensure that all requirements in a given construction are satisfied. The example given in the present paper indicates that such a verification can still be very complicated—in fact, it is much lengthier than the development of the necessary lemmas about the whole framework. However, the notions used for the particular example, namely, *link*, *consistency*, *control*, *implication chains*, and *critical nodes*, seem to be applicable to many constructions, although some modifications to their exact definitions may be necessary. Thus we envision a modular approach to priority arguments. In addition to the properties which are universally required, one can axiomatize notions such as control and implication chains, and the development of their properties will follow from this axiomatization. For other proofs in which these notions are useful, we will merely have to verify that these axioms hold, and then will have available the properties spelled out by the lemmas which follow from the axioms. In this way, we can avoid having to prove the same facts in different situations. Instead, we can appeal to a *Framework Theorem*, which will state that if certain properties of the above notions are satisfied then so are all the requirements. We will show how to use the Framework Theorem to prove a number of structure theorems in [16].

Another feature of the *iterated trees of strategies* approach is that the description of the basic module used to satisfy a requirement is a finite tree rather than a flow chart. This makes the analysis easier, and certainly simpler to describe. The loops which enter into the flow chart description are absorbed into the properties we require of our decompositions of directing sentences and action as we pass from tree to tree.

The usefulness of the iterated trees of strategies increases with the level of complexity of the priority argument. At the lower levels, the combinatorial facts proved within this framework are relatively simple, and their proofs are shorter than the characterization of the decomposition from tree to tree. As the combinatorial interactions become more complicated, the decomposition becomes a smaller part of the proof. The framework is especially useful when there are requirements of the same nature at several levels which can be handled uniformly, or the level of the argument is too high to be easily visualized in one step. (An advantage of the framework in the latter case is that its presentation is as close as possible to the standard presentation, if an inductive component is to be introduced.) In [17], we used the framework to prove a new theorem, the existence of a minimal pair of enumerable degrees whose jumps form a minimal pair over $\mathbf{0}'$. This theorem can probably be generalized to carry the simultaneity of the minimal pairs through all finite levels, and perhaps to decide the existential theory of $\mathcal{R}$ when a relation symbol for meet is added to the language of Section 1. While the two-level proof could have been carried out without the framework, the framework would be a natural way to obtain a proof for infinitely many levels. Many results of this nature should be accessible through the use of iterated trees of strategies.

REFERENCES

[1] C. J. ASH, *Stability of recursive structures in arithmetical degrees*, ***Annals of Pure and Applied Logic***, vol. 32 (1986), pp. 113–135.

[2] ———, *Labelling systems and r.e. structures*, ***Annals of Pure and Applied Logic***, vol. 47 (1990), pp. 99–119.

[3] S. B. COOPER, *Rigidity and definability in the noncomputable universe*, ***Proceedings of the international congress for logic, methodology and the philosophy of science*** (D. Prawitz and D. Westerståhl, editors), North-Holland, Amsterdam, New York, to appear.

[4] ———, *The jump is definable in the structure of the degrees of unsolvability*, ***Bulletin of the American Mathematical Society (New Series)***, vol. 23 (1990), pp. 151–158.

[5] R. M. FRIEDBERG, *Two recursively enumerable sets of incomparable degrees of unsolvability*, ***Proceedings of the National Academy of Sciences of the USA***, vol. 43 (1957), pp. 236–238.

[6] M. J. GROSZEK and T. A. SLAMAN, *Foundations of the Priority Method, I: Finite and infinite injury*, (manuscript).

[7] S. C. KLEENE and E. L. POST, *The upper semi-lattice of degrees of recursive unsolvability*, ***Annals of Mathematics***, vol. 59 (1954), pp. 379–407.

[8] J. KNIGHT, *A metatheorem for constructions by finitely many workers*, ***Journal of Symbolic Logic***, vol. 55 (1990), pp. 787–804.

[9] K. KONTOSTATHIS, *Topological framework for infinite injury*, (to appear).

[10] ———, *Topological framework for non-priority*, ***Zeitschrift für Mathematische Logik und Grundlagen der Mathematik***, vol. 37 (1991), pp. 495–500.

[11] ———, *Topological framework for finite injury*, ***Zeitschrift für Mathematische Logik und Grundlagen der Mathematik***, vol. 38 (1992), pp. 189–195.

[12] A. H. LACHLAN, *On some games which are relevant to the theory of recursively enumerable sets*, ***Annals of Mathematics***, vol. 91 (1970), pp. 291–310.

[13] ———, *The priority method for the construction of recursively enumerable sets*, ***Cambridge summer school in mathematical logic*** (A. R. D. Mathias and H. Rogers, Jr., editors), Lecture Notes in Mathematics, no. 337, Springer-Verlag, New York, 1973.

[14] ———, *A recursively enumerable degree which will not split over all lesser ones*, ***Annals of Mathematical Logic***, vol. 9 (1975), pp. 307–365.

[15] S. LEMPP and M. LERMAN, *The decidability of the existential theory of the poset of recursively enumerable degrees with jump relations*, to appear in ***Advances in Mathematics***.

[16] ———, *Iterated trees of strategies and priority arguments*, monograph: (to appear).

[17] S. LEMPP, M. LERMAN, and F. WEBER, *Minimal pair constructions and iterated trees of strategies*, ***Logical methods, in honor of Anil Nerode's 60th birthday***, Birkhäuser, Boston, Basel, Berlin, 1993, pp. 512–554.

[18] D. A. MARTIN, *Borel determinacy*, ***Annals of Mathematics***, vol. 102 (1975), pp. 363–371.

[19] A. A. MUČNIK, *On the unsolvability of the problem of reducibility in the theory of algorithms*, ***Doklady Akademii Nauk SSSR, N.S.***, vol. 108 (1956), pp. 194–197, (in Russian).

[20] E. L. POST, *Recursively enumerable sets of positive integers and their decision problems*, ***Bulletin of the American Mathematical Society***, vol. 50 (1944), pp. 284–316.

[21] G. E. SACKS, ***Degrees of unsolvability***, Annals of Mathematical Studies, no. 55, Princeton University Press, Princeton, N.J., 1963.

[22] J. R. SHOENFIELD, *Undecidable and creative theories*, ***Fundamenta Mathematica***, vol. 49 (1961), pp. 171–179.

[23] ———, *Non-bounding constructions*, ***Annals of Pure and Applied Logic***, vol. 50 (1990), pp. 191–205.

[24] R. A. SHORE, *On homogeneity and definability in the first-order theory of the Turing

degrees, ***Journal of Symbolic Logic***, vol. 47 (1982), pp. 8–16.

[25] T. A. SLAMAN and H. WOODIN, *Definability in degree structures*, (to appear).

[26] R. I. SOARE, ***Recursively enumerable sets and degrees***, Perspectives in Mathematical Logic, Springer-Verlag, Berlin, 1987.

[27] R. M. SOLOVAY, *Degrees of models of true arithmetic*, (preliminary version, unpublished).

[28] C. E. M. YATES, *On the degrees of index sets*, ***Transactions of the American Mathematical Society***, vol. 121 (1966), pp. 309–328.

[29] ———, *Banach-Mazur games, comeagre sets, and degrees of unsolvability*, ***Mathematical Proceedings of the Cambridge Philosophical Society***, vol. 79 (1976), pp. 195–220.

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF WISCONSIN
MADISON, WI 53706-1388, USA

E-mail: lempp@math.wisc.edu

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF CONNECTICUT
STORRS, CT 06269-3009, USA

E-mail: mlerman@math.uconn.edu