INFORMATION AND CONTROL 10, 447–474 (1967)

Language Identification in the Limit

E MARK GOLD*

The RAND Corporation

Language learnability has been investigated. This refers to the following situation: A class of possible languages is specified, together with a method of presenting information to the learner about an unknown language, which is to be chosen from the class. The question is now asked, "Is the information sufficient to determine which of the possible languages is the unknown language?" Many definitions of learnability are possible, but only the following is considered here: Time is quantized and has a finite starting time. At each time the learner receives a unit of information and is to make a guess as to the identity of the unknown language on the basis of the information received so far. This process continues forever. The class of languages will be considered *learnable* with respect to the specified method of information presentation if there is an algorithm that the learner can use to make his guesses, the algorithm having the following property: Given any language of the class, there is some finite time after which the guesses will all be the same and they will be correct.

In this preliminary investigation, a *language* is taken to be a set of strings on some finite alphabet. The alphabet is the same for all languages of the class. Several variations of each of the following two basic methods of information presentation are investigated: A *text* for a language generates the strings of the language in any order such that every string of the language occurs at least once. An *informant* for a language tells whether a string is in the language, and chooses the strings in some order such that every string occurs at least once.

It was found that the class of context-sensitive languages is learnable from an informant, but that not even the class of regular languages is learnable from a text.

1. MOTIVATION: TO SPEAK A LANGUAGE

The study of language identification described here derives its motivation from artificial intelligence. The results and the methods used also

* Present address: Institute for Formal Studies, 1720 Pontius Ave., Los Angeles, California 90025. Present sponsor: Air Force Office of Scientific Research. Contract F44620-67-C-0018.

have implications in computational linguistics, in particular the construction of discovery procedures, and in psycholinguistics, in particular the study of child learning. These implications are discussed in Section 4.

I wish to construct a precise model for the intuitive notion "able to speak a language" in order to be able to investigate theoretically how it can be achieved artificially. Since we cannot explicitly write down the rules of English which we require one to know before we say he can "speak English," an artificial intelligence which is designed to speak English will have to learn its rules from implicit information. That is, its information will consist of examples of the use of English and/or of an informant who can state whether a given usage satisfies certain rules of English, but cannot state these rules explicitly.

For the purpose of artificial intelligence, a model of the rules of usage of natural languages must be general enough to include the rules which do occur in existing natural languages. This is a lower bound on the generality of an acceptable linguistic theory. On the other hand, the considerations of the last paragraph impose an upper bound on generality: For any language which can be defined within the model there must be a training program, consisting of implicit information, such that it it possible to determine which of the definable languages is being presented.

Therefore this research program consists of the study of two subjects: Linguistic structure and the learnability of these structures. This report describes the first step of this program. A very naive model of language is assumed, namely, a language is taken to be a distinguished set of strings. Such a language is too simple to do anything with (for instance, to give information or to pose problems), but it has enough structure to allow its learnability to be investigated as follows: Models of information presentation are defined, and for each I ask "For which classes of languages does a learning algorithm exist?"

In the second step of this program (Gold, 1966), which will not be discussed here, nontrivial models of the usages of language are constructed. The next step will be to return to learnability theory and determine whether reasonable training programs exist for linguistic structures of this type.

2. LANGUAGE IDENTIFICATION MODELS

Appendix II lists intuitive definitions of some of the terminology of recursive theory used herein.

Let A be a finite set (the alphabet of the languages to be considered)

and ΣA represent the set of all finite strings of elements from A. A is to be considered fixed throughout this paper. The results presented in the next chapter are independent of the cardinality of A so long as it is not void. A language L will signify any subset of ΣA. In an actual language this may represent, for instance, the set of meaningful strings of words.

A *language learnability model* will signify the following triple:

1. *A definition of learnability.*
2. *A method of information presentation.*
3. A *naming relation* which assigns names (perhaps more than one) to languages. The "learner" identifies a language by stating one of its names. The names could be called grammars.

Only one definition of learnability, which will be called *identifiability in the limit*, will be considered here. Six alternative methods of information presentation and two alternative naming relations will be considered, making a total of twelve models of language learnability. The definitions will now be given, and the results are stated in Section 3. The proofs are in Appendix I. The basic ideas behind the proofs are described in Sections 7 and 9.

Time will be taken to be quantized and start at a finite time:

$$t = 1, 2, \cdots$$

At each time t the learner is presented with a unit of information i_t concerning the unknown language L. In any language learnability model, the *method of information presentation* consists of assigning to each L a set of allowable training sequences, $i_1, i_2, \cdots$.

LEARNABILITY. At each time t the learner is to make a guess g_t of a name of L based on the information it has received through time t. Thus the learner is a function G which takes strings of units of information into names:

$$g_t = G(i_1, \cdots, i_t).$$

L will be said to be *identified in the limit* if, after some finite time, the guesses are all the same and are a name of L. A class of languages will be called *identifiable in the limit* with respect to a given language learnability model if there is an effective learner, i.e., an algorithm for making guesses, with the following property: Given any language of the class and given any allowable training sequence for this language, the language will be identified in the limit.

For each of the 12 models of language learnability the following ques-

tion has been investigated (the results are in the next Section): Which classes of languages are identifiable in the limit? Note that identifiability (learnability) is a property of classes of languages, not of individual languages.

In the case of identifiability in the limit the learner does not necessarily know when his guess is correct. He must go on processing information forever because there is always the possibility that information will appear which will force him to change his guess. If the learner were required to know when his answer is correct (this is equivalent to "finite identifiability" defined in Section 6), then none of the classes of languages investigated in the next chapter would be learnable in any of the learnability models. My justification for studying identifiability in the limit is this: A person does not know when he is speaking a language correctly; there is always the possibility that he will find that his grammar contains an error. But we can guarantee that a child will eventually learn a natural language, even if it will not know when it is correct.

INFORMATION PRESENTATION. Two basic methods of information presentation will be considered, "text" and "informant." Three variations of each will be defined.

A *text* for L is a sequence of strings $x_1, x_2, \cdots$ from L such that every string of L occurs at least once in the text. At time t the learner is presented x_t. Note that for any given language many texts are possible. The three variations of this method of information presentation to be considered are obtained by putting different restrictions on the class of allowed texts:

1. *Arbitrary Text:* x_t may be any function of t.
2. *Recursive Text:* x_t may be any recursive function of t.
3. *Primitive Recursive Text:* x_t may be any primitive recursive function of t.

An *informant* for L can tell the learner whether any string is an element of L, and does so at each time t for some string y_t. Three types of informant will be considered; these differ in how the y_t are chosen:

1. *Arbitrary Informant:* y_t may be any function of t so long as every string of ΣA occurs at least once.
2. *Methodical Informant:* An enumeration is assigned a priori to the strings of ΣA, and y_t is taken to be the tth string of the enumeration.
3. *Request Informant:* At time t the learner chooses y_t on the basis of information received so far.

NAMING RELATION. Two naming relations will be considered, "tester"

and "generator." In both cases a name of a language, i.e., a grammar, will be a Turing machine: A *tester* for L is a Turing machine which is a decision procedure for L, that is, the Turing machine defines the function from strings to natural numbers which has the value 1 for strings in L and 0 for strings not in L. A *generator* for L is a Turing machine which generates L, that is, it defines a function from positive integers to strings such that the range of this function is exactly L. A tester exists iff L is recursive and a generator exists iff L is recursively enumerable.

Two *language learnability models* will be called *equivalent* if exactly the same classes of languages are identifiable in the limit with respect to either model. Two *naming relations* will be called *equivalent* if, for every method of information presentation, the two language learnability models obtained by using these naming relations are equivalent. Similarly, two *methods of information presentation* will be called *equivalent* if every naming relation yields two equivalent language learnability models.

Suppose two naming relations are effectively intertranslatable. That is, suppose there is an algorithm for each of the naming relations which, given a name of a language in this naming relation, would yield a name of the language in the other. Then these are equivalent naming relations.

It is well known that it is possible to effectively translate from testers to generators. Therefore, given any method of information presentation, any class of languages which is tester-identifiable in the limit must also be generator-identifiable in the limit. However, it is not possible to effectively translate from generators to testers, even if we restrict ourselves to recursive languages for which both are defined. Therefore, it is possible for a method of information presentation to exist such that a class of languages is generator identifiable in the limit but not tester identifiable in the limit. An example of this is given in the next Section. This subject is discussed further in Section 11.

The three variations of information presentation by informant are equivalent. They are defined separately only in order to make this point.

3. LANGUAGE IDENTIFICATION RESULTS

For every pair consisting of one of the 12 learnability models together with one of the language classes listed in Table I it has been determined whether the class of languages is identifiable in the limit. The language classes are listed in descending order, i.e., each class is properly contained in the class above it. The dividing lines between identifiable in the limit

TABLE I

DIVIDING LINES BETWEEN LEARNABILITY AND NONLEARNABILITY OF LANGUAGES

Learnability model	Class of languages
Anomalous text[a] ⟶	Recursively enumerable recursive
Informant ⟶	Primitive recursive Context-sensitive Context-free Regular Superfinite
Text ⟶	Finite cardinality languages

[a] Anomalous text refers to the use of the generator-naming relation and information presentation by means of primitive recursive text.

and nonidentifiable in the limit are shown in the table. The classes of languages below the dividing line shown for a given model of language learnability are identifiable in the limit with respect to this model; those above the dividing line are not. It is possible to represent the results by means of dividing lines in this way because of the following obvious facts: If a class of languages is identifiable in the limit with respect to a given language learnability model, then the same holds for any subclass; if a class is not identifiable in the limit, then the same holds for any superclass.

In the table, "informant" refers to any of the three variations of informant together with either the generator- or tester-naming relation. That is, the same results have been obtained, so far, for each of the six language learnability models which utilize an informant. Of the six language learnability models which utilize a text for information presentation, five of them have given the same results, shown as "text" in the table. The remaining model, shown as "anomalous text," is primitive recursive text with the generator-naming relation.

A *super-finite class of languages* denotes any class which contains all languages of finite cardinality and at least one of infinite cardinality.

The anomalous model using a text is of no practical interest, but three noteworthy conclusions can be drawn from it: (1) It shows that

restrictions on the order of presentation of elements of the text can greatly increase the learnability power of this method of information presentation. (2) Note that the difference between a text and an informant is that a text only presents the learner with positive instances, namely, elements of the language, whereas an informant presents both positive and negative instances. Therefore, one would expect the informant to be more powerful. However, "anomalous text" is more powerful than any of the "informant" models, which shows that one must carefully consider the details of the learnability model. (3) "Anomalous text" shows that the choice of naming relation can make a difference since, in this case, the generator-naming relation is far more powerful than tester.

4. IMPLICATIONS OF LANGUAGE LEARNABILITY RESULTS

To the Study of Child Learning of Language. Recently, psycholinguists have begun to study the acquisition of grammar by children (e.g., McNeill, 1966). Those working in the field generally agree that most children are rarely informed when they make grammatical errors, and those that are informed take little heed. In other words, it is believed that it is possible to learn the syntax of a natural language solely from positive instances, i.e., a "text." However, the results presented in the last Section show that only the most trivial class of languages considered is learnable (in the sense of identification in the limit) from text, neglecting "anomalous text." If one accepts identification in the limit as a model of learnability, then this conflict must lead to at least one of the following conclusions:

1. The class of possible natural languages is much smaller than one would expect from our present models of syntax. That is, even if English is context-sensitive, it is not true that any context-sensitive language can occur naturally. Equivalently, we may say that the child starts out with more information than that the language it will be presented is context-sensitive. In particular, the results on learnability from text imply the following: The class of possible natural languages, if it contains languages of infinite cardinality, cannot contain all languages of finite cardinality.

2. The child receives negative instances by being corrected in a way we do not recognize. If we can assume that the child receives both positive and negative instances, then it is being presented information by an "informant." The class of primitive recursive languages, which includes

the class of context-sensitive languages, is identifiable in the limit from an informant. The child may receive the equivalent of negative instances for the purpose of grammar acquisition when it does not get the desired response to an utterance. It is difficult to interpret the actual training program of a child in terms of the naive model of a language assumed here.

3. There is an a priori restriction on the class of texts which can occur, such as a restriction on the order of text presentation. The child may learn that a certain string is not acceptable by the fact that it never occurs in a certain context. This would constitute a negative instance.

To Artificial Intelligence. The training program of an artificial intelligence can certainly include an informant, whether or not children receive negative instances. Therefore, the results of Table I show that a learning algorithm can be constructed for the identification of primitive recursive predicates on strings, which probably include all the predicates children learn. However, for the purpose of efficiency it is still of significance to determine what additional information may be available to children, either in the form of an a priori restriction on the class of predicates which can occur in natural languages, or in the form of information which can be obtained from the order of presentation of naturally occurring texts.

To the Construction of Discovery Procedures. Attempts have been made to construct an algorithm for automatically generating a phrase structure grammar for a language solely by analyzing a text of the language. One approach (Lamb, 1961) uses the "distributional analysis" of Harris (1951, 1964) and Hockett (1958). Namely, one associates phrases which are found to occur in the same context, thereby defining phrase categories and simultaneously enlarging the set of contexts which can be considered equivalent; then one records how phrase categories are constructed by concatenation of phrase categories. Another approach which has been proposed (Solomonoff, 1964) uses "identification by enumeration," which is defined in Section 7.

These attempts suggest the question, "Is there enough information in a text, even one of unlimited length, to allow the identification of a context-free language?" The results presented in Section 3 show that it is impossible to construct a learning algorithm for the entire class of context-free languages if the only information is an arbitrary text. If one wishes to assume restrictions on the order of presentation of the text, then a successful learning algorithm must be sensitive to the order

of the text. Thus, statistical approaches such as distributional analysis are not suitable for this purpose. However, it would be useful to determine if there are interesting subclasses of the class of context-free languages which can be identified in the limit by either of these approaches.

5. IDENTIFICATION OF FUNCTIONS AND BLACK BOXES

This Section is a summary of the results of a previous paper (Gold, 1965) which is devoted to the learnability of two types of objects other than languages:

TIME FUNCTION. At each time t, a *time function* produces an output, a positive integer, which depends only on t. Formally, a time function is a function of one variable which takes positive integers (time) into positive integers (outputs).

BLACK BOX. A black box has provision for an input at each time, as well as an output. Each output is determined by the inputs that have previously been applied to the black box. More precisely, let an *alphabet* here signify either a finite set with at least two elements, or else the set of positive integers. Then a black box consists of the following triple: An input alphabet I; an output alphabet O; a black box function b which takes input strings into the output alphabet, thereby determining the output at time t: $o_t = b(i_1, \cdots, i_t)$.

Thus, a time function is a special case of a black box. In the case of a time function, o_t depends only on t and not on a previous input string. A time function can be described as a black box with a degenerate input alphabet consisting of one element.

Throughout the study of black box learnability, I and O are to be considered as fixed alphabets, i.e., I and O are chosen a priori, and all black boxes are to use these two alphabets.

In the case of time function learnability the following situation is studied. The learner observes the successive outputs of a time function and is to guess what function it is observing; that is, the learner consists of an identity guessing algorithm G which yields a guess g_t at each time t as to the identity of the time function, g_t being determined by the outputs which the time function has produced so far: $g_t = G(o_1, \cdots, o_t)$.

In the case of black box identification, the learner consists of an experimenting algorithm E as well as an identity guessing algorithm G. E determines the input which the learner will apply to the unknown black box at any time as a function of the previous outputs of the black box: $i_t = E(o_1, \cdots, o_{t-1})$. The identity guessing algorithm makes a

TABLE II

DIVIDING LINES BETWEEN LEARNABILITY AND NONLEARNABILITY FOR TIME FUNCTIONS AND BLACK BOXES

Type of object	Class of objects
Time functions	Recursive → Primitive recursive
Black boxes	→ Finite automata

guess, at each time t, as to the identity of the black box: $g_t = G(o_1, \cdots, o_t)$.

It is too much to require the learner to identify a black box in the sense of finding its identity at the beginning of the experiment, $t = 1$. This is because, for instance, the black box may be such that the first input which the learner applies to it may trap the black box in a subset of its possible states, so that the learner will never be able to determine what the behavior of the black box would have been if its first input had been different. Therefore, only *weak learnability* will be considered; namely, the learner will be asked to predict, at each time, the future behavior of the black box. That is, the learner is to guess the present black box function, rather than that at $t = 1$.

Only one model of time function learnability and one of black box learnability will be considered. The method of information presentation for each model was described above. As in the models of language learnability, in both of these models "learnability" will signify "identification in the limit." The naming relation will be the following: The names of a time function, or of a black box (actually, its black box function), will be taken to be those Turing machines which compute it.

Three classes each of time functions and of black boxes have been considered. Table II shows which of these are identifiable in the limit. As in Table I, the classes are listed in descending order in Table II. Finite automata time functions denote ultimately periodic functions.

6. ABSTRACT MODEL OF IDENTIFICATION

An *identification situation* consists of the following three items:

1. A *class* Ω *of objects.* One of the objects will be chosen, the learner will be presented information about it, and the learner is to figure out which one it is.

2. A method of information presentation. At each time t the learner receives a unit of information i_t which is chosen from a set I. The *method of information presentation* consists of specifying, for each $\omega \in \Omega$, which sequences of units of information, $i_1, i_2, \cdots$, are allowable. Let the set of allowable sequences be designated $I^\infty(\omega)$.

3. A naming relation. The learner is to identify the unknown object by finding one of its names. A *naming relation* consists of a set N of names and a function f which assigns an object to each name, $f: N \to \Omega$.

The identification problem is to determine whether there is a rule the learner can use to accomplish the following: For any object $\omega \in \Omega$ and for any information sequence from $I^\infty(\omega)$, on the basis of that information sequence the rule will yield a name n of ω, that is, $f(n) = \omega$. Three variations of the identification problem are the following, of which only the first is considered in this paper.

Identification in the limit has made some appearances previously in the pattern recognition literature (e.g., Aizerman *et al.*, 1964). In this case the learner is to guess a name of the unknown object at each time. It is required that there be a finite time after which the guesses are all the same and are correct.

Finite identification is the type of identification problem usually considered. It is best known in automata theory (e.g., Gill, 1961). In finite identification, the learner is to stop the presentation of information at some finite time when it thinks it has received enough, and state the identity of the unknown object. This is not possible unless there is some finite time at which the information distinguishes the unknown object. That is, no other object satisfies the information.

Fixed-time identification. In this case the information sequence stops after some finite time which is specified a priori and which is independent of the object being described. The learner is to then state the identity of the unknown object.

Saying that a class of objects is identifiable in the limit implies not only that a suitable guessing function G exists, but that it is effective; that is, there exists an algorithm which computes it. The class of objects will be called *ineffectively identifiable in the limit* if a suitable G exists, regardless of whether it is effective. Note that whether a class of objects is ineffectively identifiable in the limit does not depend on the naming relation so long as every object has at least one name. This is because any two naming relations are intertranslatable if we do not require translation to be effective.

An identification situation will be said to satisfy the *distinguishability*

condition if the $I^{\infty}(\omega)$ are disjoint; that is, if there is no information sequence which describes two different objects.

An identification situation will be said to satisfy the *collapsing uncertainty condition* if the following holds: For any information string $i_1, \cdots, i_t$, let Ω_t denote the set of those objects which agree with the information received so far, i.e., those ω such that $I^{\infty}(\omega)$ contains an information sequence which begins $i_1, \cdots, i_t$. For any information sequence, the Ω_t will be a descending sequence. The collapsing uncertainty condition requires that, for any object ω and any information sequence of $I^{\infty}(\omega)$, the limit set of the Ω_t contains only ω. That is, for any ω' different from ω there is a time after which the information will eliminate ω', namely $\omega' \notin \Omega_t$.

7. METHODS OF IDENTIFICATION IN THE LIMIT

Identification by enumeration refers to the following guessing rule: Enumerate the class of objects in any way, perhaps with repetitions. That is, choose a function from the positive integers to the class of objects such that the range of the function is the entire class. At time t guess the unknown object to be the first object of the enumeration which agrees with the information received so far, i.e., which is in Ω_t. This guessing rule will be effective if the following two conditions hold: (1) Given any information string $i_1, \cdots, i_t$ and any positive integer n, there is an effective method for determining whether the nth object of the enumeration is in Ω_t. (2) There is an effective method for finding a name of the nth object of the enumeration.

To be precise, "identification by enumeration" refers to a class of guessing rules, since there are many possible enumerations.

If we assume that I is countable, then any class of objects which is ineffectively identifiable in the limit must be countable. This is because the domain of the guessing function G, namely, finite strings of elements of I, is countable.

Henceforth, it will be assumed that I and Ω are countable, and that every object has at least one name.

THEOREM 7.1. *For ineffective identifiability in the limit, the distinguishability condition is necessary and the collapsing uncertainty condition is sufficient. Indeed, the collapsing uncertainty condition implies that identification by enumeration gives ineffective identification in the limit for any enumeration. If $I^{\infty}(\omega)$ is countable for every ω, then the distinguishability condition is sufficient for ineffective identifiability in the limit.*

PROOF. Ineffective identifiability in the limit $\Rightarrow$ distinguishability

If the distinguishability condition does not hold, then there is an allowable information sequence which describes two different objects, so that it is impossible to know which is the unknown object.

Collapsing uncertainty $\Rightarrow$ identification by enumeration gives ineffective identification in the limit for any enumeration: The object to be identified must occur somewhere in the enumeration. Let its first occurrence be at position n, There are at most $n - 1$ different objects before this position in the enumeration. The collapsing uncertainty condition implies that there is some finite time after which none of these prior objects will agree with the information presented to the learner. After that time the unknown object will be the first object of the enumeration which satisfies the information received, and will therefore be correctly guessed by the learner.

$I^{\infty}(\omega)$ is countable for every ω, together with distinguishability $\Rightarrow$ ineffective identifiability in the limit. If we wish to identify the information sequence, then the collapsing uncertainty condition always holds. Saying that $I^{\infty}(\omega)$ is countable for every ω is equivalent to saying that the set of allowable information sequences is countable. In this case, identification by enumeration may be used to identify in the limit the information sequence being presented to the learner. The distinguishability condition implies that one can translate (not necessarily effectively) from information sequences to objects, and therefore to names of objects. Q.E.D.

Returning to the specific case of language identification, note that information presentation by informant satisfies the collapsing uncertainty condition no matter what class of languages is considered. That is why the class of primitive recursive languages is identifiable in the limit from an informant; namely, an effective enumeration of the characteristic functions of this class of languages exists thereby giving an effective identification-by-enumeration guessing rule (see Theorem I.4, Appendix I).

Information presentation by text satisfies the distinguishability condition for any class of languages, but it does not satisfy collapsing uncertainty for any class of languages which contains two languages such that one is a subset of the other.

The following guessing algorithm shows that the class of languages of finite cardinality is identifiable in the limit from an arbitrary text: Guess the unknown language to consist solely of the strings generated so far by the text (see Theorem I.6).

To see that the entire class of recursively enumerable languages is

identifiable in the limit from primitive recursive text using the generator-naming relation (see Theorem I.7), note that the class of primitive recursive texts is effectively enumerable. Therefore, the text can be effectively identified in the limit using identification by enumeration. Since the text is a generator for the language, this is all that is needed. This is an example of the method of identification described at the end of the proof of Theorem 7.1.

8. INEFFECTIVE IDENTIFIABILITY IN THE LIMIT RESULTS

LANGUAGE IDENTIFIABILITY. If information presentation is by informant, then the collapsing uncertainty condition is satisfied, so that any countable class of languages is ineffectively identifiable in the limit using identification by enumeration. Of course, all the languages must have names.

If information presentation is by recursive or primitive recursive text, then, again, any countable class of languages is ineffectively identifiable in the limit. This is because there are only a countable number of possible texts, so that the text can be identified in the limit by means of identification by enumeration.

If information presentation is by arbitrary text, then the results for ineffective identifiability in the limit are the same as for effective identifiability in the limit; namely, the class of languages of finite cardinality is ineffectively identifiable in the limit, but every proper superclass is not. This can be proved by the same methods used to prove Theorems I.6 and I.8.

TIME FUNCTION IDENTIFIABILITY. The method of information presentation in the model of time function learnability satisfies the collapsing uncertainty condition. Therefore, any countable class of time functions is ineffectively identifiable in the limit.

BLACK BOX IDENTIFIABILITY. Any countable class of black boxes is ineffectively (weak) identifiable in the limit. This can be proved by the same method used to prove Theorems 9 and 10 in Gold (1965).

9. THE WEAKNESS OF TEXT

It is of great interest to find why information presentation by text is so weak and under what circumstances it becomes stronger. Therefore, it is worthwhile to understand the method used in Theorems I.8 and I.9 to prove that any class of languages containing all finite languages and at least one infinite language is not identifiable in the limit from a text in five out of six of the models using text.

The basic idea is proof by contradiction. Consider any proposed guessing algorithm. It must identify any finite language correctly after a finite amount of text. This makes it possible to construct a text for the infinite language which will fool the learner into making a wrong guess an infinite number of times as follows. The text ranges over successively larger, finite subsets of the infinite language. At each stage it repeats the elements of the current subset long enough to fool the learner.

Thus, the method of proof of the negative results concerning text depends on the possibility of there being a huge amount of repetition in the text. Perhaps this can be prevented by some reasonable probabilistic assumption concerning the generation of the text. In this case one would only require identification in the limit with probability one, rather than for every allowed text.

I have been asked, "If information presentation is by means of text, why not guess the unknown language to be the simplest one which accepts the text available?" This is identification by enumeration. It is instructive to see why it will not work for most interesting classes of languages: The universal language (if it is in the class) will have some finite complexity. If the unknown language is more complex, then the guessing procedure being considered will always guess wrong, since the universal language is consistent with any finite text. This follows from the fact that, if L is the unknown language and if $L' \supset L$, then L' is consistent with any finite segment of any text for L. The problem with text is that, if you guess too large a language, the text will never tell you that you are wrong.

10. LEARNING TIME

Consider an identification situation which satisfies the collapsing uncertainty condition. Choose an enumeration of the class of objects and let G_0 be the identification-by-enumeration guessing rule which uses this enumeration. At first sight, identification by enumeration appears to be a naive approach to learning. However, it will be shown that G_0 is the most efficient possible guessing rule with respect to learning time. This holds even if ineffective guessing rules are allowed and if the enumeration has duplications. This result is somewhat surprising in view of the fact that there are many different identification-by-enumeration guessing rules, obtained by using different enumerations. This means that none of them is uniformly better than any other, in the sense defined below, for the purpose of minimizing learning time.

Let G be any guessing rule, ω be any element of the class Ω of objects, and $\bar{\imath}$ any information sequence allowed for ω. Define the learning time $\tau(G, \omega, \bar{\imath})$ to be the first time such that at that time and all following times all the guesses of G as to the identity of ω will be the same and correct. Define the learning time to be ∞ if no such time exists.

Let G and G' be any two guessing rules. G will be said to be *uniformly faster* than G' if the following two conditions hold: (1) Given any ω and any allowable $\bar{\imath}$ for ω, then G will identify ω at least as soon as G' will identify ω, that is,

$$\tau(G, \omega, \bar{\imath}) \leqq \tau(G', \omega, \bar{\imath}).$$

(2) There is some ω_0 and an allowable $\bar{\imath}_0$ for ω_0 such that G will identify ω_0 sooner that G':

$$\tau(G, \omega_0, \bar{\imath}_0) < \tau(G', \omega_0, \bar{\imath}_0)$$

THEOREM 10.1. *If G_0 is an identification-by-enumeration guessing rule, then there is no guessing rule uniformly faster than G_0.*

PROOF. This is what has to be proved: Let G be any guessing rule. If there is an ω and an allowable $\bar{\imath}$ for ω such that G is faster than G_0, i.e., $\tau(G, \omega, \bar{\imath}) < \tau(G_0, \omega, \bar{\imath})$, then there is an ω' and an allowable $\bar{\imath}'$ for ω' such that G_0 is faster than G, i.e., $\tau(G_0, \omega', \bar{\imath}') < \tau(G, \omega', \bar{\imath}')$.

G_0 is constructed in such a way that, once it guesses correctly, its guesses never change. Therefore, if G_0 is presented with the object ω to identify, by being given information sequence $\bar{\imath}$, then $\tau(G_0, \omega, \bar{\imath})$ is the first time that G_0 guesses the identity of the unknown object to be ω. At the earlier time, $\tau(G, \omega, \bar{\imath})$, G_0 must guess the name of some other object, say ω'. At any time that G_0 guesses the name of an object, that object must agree with the information received so far. That is, at the time that G_0 guesses ω' there must be an allowable $\bar{\imath}'$ for ω' such that $\bar{\imath}'$ is the same as $\bar{\imath}$ up to this time. Thus, if ω' were the unknown object and $\bar{\imath}'$ the information sequence, then at that time, namely $\tau(G, \omega, \bar{\imath})$, G and G_0 would make the same guesses as they would if presented ω and $\bar{\imath}$: G_0 would guess ω' and G would guess ω. That is, if presented with ω' and $\bar{\imath}'$, G_0 would be correct before G. Q.E.D.

Note that the proof of Theorem 9.1 remains valid even if G_0 does not identify every object of the class in the limit and G does. It is only necessary that, for every finite initial subsequence of every allowable information sequence, there exist an object in the enumeration which is consistent with it.

11. TRANSLATION FROM GENERATORS TO TESTERS

The purpose of this Section is to use the results on language identification in the limit presented in Section 3 to solve a problem in recursive theory. In addition to the generator- and tester-naming relations, a third method of assigning names to recursive sets, called domain generators, is defined below. Suzuki (1959) has shown that there is no effective method for going from recursive sets, described by domain generators, to their complements, described in the same way. This result will here be strengthened in two ways: It will be shown that there is no 2-recursive (defined below) translation of this type, and that, rather than the entire class of recursive sets, one can restrict one's consideration to any class of sets which contains all finite sets and at least one infinite set without changing this result. It has been pointed out to me by Norman Shapiro that one can easily construct a 3-recursive translation from recursive sets to their complements, using the domain generator-naming relation, thus completely establishing the difficulty, in the Kleene hierarchy, of this type of translation.

For the purpose of this Section it is desirable to think of languages as sets of positive integers, rather than sets of strings. This may be accomplished by means of any recursive one-to-one correspondence between the strings of ΣA and the positive integers.

Let $Z_n(x)$ be the number-theoretic function of one variable defined by the Turing machine whose Gödel number is n. The two naming relations for languages are defined formally as follows.

Generator. A *generator* for L is a positive integer n such that L is the range of $Z_n(x)$.

Tester. A *tester* for L is a positive integer n such that $Z_n(x) = \chi_L(x)$, where χ_L is the characteristic function of L.

It will be assumed in this Section that in all naming relations the *names are numbers.*

Translation. Given two naming relations, N_1 and N_2, a *translation* from N_1 to N_2 is a partial, number-theoretic function $f(n)$ such that, if n is a name of an object in N_1, then f(n) is defined and is a name of that object in N_2.

Limiting recursive function. A partial, number-theoretic function $f(n)$ will be called *limiting recursive* if there is a total recursive "guessing function" $g(n, t)$ such that

$$f(n) = \lim_t g(n, t), \tag{10.1}$$

where the limit of a sequence of positive integers is taken to be undefined if the sequence is not constant after some finite point; otherwise the limit is defined to be the value at which the sequence ultimately becomes constant.

THEOREM 10.1. *If a class of objects is identifiable in limit using some method of information presentation and using naming relation N_1, and if there is a limiting recursive translation from N_1 to naming relation N_2, then the class of objects is identifiable in the limit using the same method of information presentation and N_2.*

PROOF. Let $f(n)$ be a limiting recursive translation from N_1 to N_2, and $g(n, t)$ be a total recursive function such that Eq. (10.1) holds, and let G_1 be a suitable guessing rule using N_1. Then a suitable guessing rule using N_2 is the following:

For a given information sequence, suppose that at time t the guess made by G_1 is g_t. Then, when using N_2, let the guess be $g(g_t, t)$. Call this guessing rule G_2.

To see that G_2 is suitable note that, using N_1, there is a time t_1 after which all the g_t will equal a fixed value g_0, which is correct in N_1. By Eq. (10.1), there is a t_2 such that, for all $t \geqq t_2$,

$$f(g_0) = g(g_0, t). \tag{10.2}$$

Therefore, for all times greater than t_1 and t_2, the guess of G_2 will be $f(g_0)$, which is correct in N_2. Q.E.D.

Consider language learnability with information presentation by means of primitive recursive text. The results shown in Table I differ for the two naming relations, generator and tester. This leads to the following conclusion:

COROLLARY 10.1. *If C is a class of languages which contains all finite languages and at least one infinite recursive language, then there is no limiting recursive translation from testers for C to generators.*

In order to compare this result with that of Suzuki, it is necessary to define two more naming relations for recursive languages:

Domain generator. A *domain generator* for L is a positive integer n such that L is the domain of $Z_n(x)$.

Anti-domain generator. n is an *anti-domain generator* for L if it is a domain generator for the complement of L.

Representing predicate. For any partial number-theoretic function $f(n)$, its *representing predicate* $P(n, m)$ will be defined to be true for just those pairs (n, m) such that $f(n)$ is defined and $f(n) = m$.

k-Recursive function. A partial, number-theoretic function will be called *k-recursive* if its representing predicate is k-r.e. (recursively enumerable) in the Kleene hierarchy.

Note that the 1-recursive functions are just the partial recursive functions.

It is shown elsewhere (Gold, 1965) that the limiting recursive functions are the same as the 2-recursive functions.

The strengthening of Suzuki's result described at the beginning of this chapter can now be stated formally:

THEOREM 10.2. *If C is a class of languages which contains all finite languages and at least one infinite recursive language, then there is no 2-recursive translation from domain generators for C to anti-domain generators.*

PROOF. Theorem 10.2 follows from Corollary 10.1 together with the following facts which can be proved by standard methods:

There is a partial recursive translation from testers to domain generators.

Given both a domain generator and an anti-domain generator for a recursive set, there is an effective procedure for finding a generator for it.

Composition of 2-recursive and recursive functions yields 2-recursive functions. Q.E.D.

12. INDUCTIVE INFERENCE

Concerning inductive inference, philosophers often occupy themselves with the following type of question: Suppose we are given a body of information and a set of possible conclusions, from which we are to choose one. Some of the conclusions are eliminated by the information. The question is, of the conclusions which are consistent with the information, which is "correct"?

If some sort of probability distribution is imposed on the set of conclusions, then the problem is meaningful. But if no basis for choosing between the consistent conclusions is postulated a priori, then inductive inference can do no more than state the set of consistent conclusions.

The difficulty with the inductive inference problem, when it is stated this way, is that it asks, "What is the correct guess at a specific time with a fixed amount of information?" There is no basis for choosing between possible guesses at a specific time. However, it *is* interesting to study a *guessing strategy*. Now one can investigate the limiting behavior of the guesses as successively larger bodies of information are considered. This report is an example of such a study. Namely, in in-

teresting identification problems, a learner cannot help but make errors due to incomplete knowledge. But, using an "identification in the limit" guessing rule, a learner can guarantee that he will be wrong only a finite number of times.

RECEIVED: February 20, 1967

REFERENCES

AIZERMAN, M. A., BRAVERMAN, E. M., AND ROZONOER, L. I. (1964). Theoretical foundations of the potential function method in pattern recognition. *Automation and Remote Control* **25**, 821–837 and 1175–1190.

GILL, A. (1961). State-identification experiments in finite automata. *Information and Control* **4**, 132–154.

GOLD, E MARK (1965). Limiting recursion. *J. Symb. Logic* **30**, 28–48.

GOLD, E MARK (1966). "Usages of Natural Language." Informal report available from the author at Institute for Formal Studies, 1720 Pontius Ave., Los Angeles, California 90025.

HARRIS, ZELLIG (1951). "Methods in Structural Linguistics." Univ. of Chicago Press.

HARRIS, ZELLIG (1964). Distributional structure. *In* "The Structure of Language" (J. A. Fodor and J. J. Katz, eds.). Prentice Hall, New York.

HOCKETT, C. F. (1958). "A Course in Modern Linguistics." Macmillan, New York.

LAMB, SYDNEY, M. (1961). On the mechanization of syntactic analysis. *In* "1961 Conference on Machine Translation of Languages and Applied Language Analysis" (National Physical Laboratory Symposium No. 13), Vol. II, pp. 674–685. Her Majesty's Stationery Office, London.

MCNEILL, DAVID (1966). Developmental psycholinguistics. Part of the chapter "Psycholinguistics" by George Miller and David McNeill to appear in the "Handbook of Social Psychology."

SOLOMONOFF (1964). A formal theory of inductive inference. *Information and Control* **7**, 1–22 and 224–254.

SUZUKI, Y. (1959). Enumeration of recursive sets. *J. Symb. Logic* **24**, 311.

APPENDIX I

PROOFS OF LANGUAGE IDENTIFICATION RESULTS

It can be shown by standard methods that there is a recursive translation from testers to generators. Thus, Theorem 10.1 gives

THEOREM I.1. *Given any method of information presentation, if a class of languages is identifiable in the limit using the tester-naming relation, then it is identifiable in the limit using the generator-naming relation.*

COROLLARY I.2. *Given any method of information presentation, if a class of languages is not identifiable in the limit using the generator-naming*

relation, then it is not identifiable in the limit using the tester-naming relation.

THEOREM I.3. *Arbitrary informant, methodical informant, and request informant are equivalent methods of information presentation.*

PROOF. Identifiability from arbitrary informant $\Rightarrow$ identifiability from methodical informant $\Rightarrow$ identifiability from request informant: The first implication follows from the fact that an identity guessing algorithm which will work for any informant will obviously work for the special case of a methodical informant. The second follows from the fact that the learner can ask a request informant for methodical information.

Identifiability from request informant $\Rightarrow$ identifiability from arbitrary informant: Suppose we have an identity guessing algorithm suitable for a request informant and we are faced with an arbitrary informant. Whatever information our learner wishes to request at some time, an arbitrary informant is required to provide it eventually. We can modify our learner so that it will wait until it receives the information it currently desires before it makes its next guess. Q.E.D.

The previous five theorems and corollaries compare the methods of information presentation and the naming relations for language identification. These results, together with the following six theorems, yield the language learnability results presented in Table I.

As in Section 10, it will be desirable to think of languages as sets of positive integers, rather than of strings. However, here it will be necessary to achieve this by means of a primitive recursive one-to-one correspondence, so that primitive recursive sets of strings will be taken into primitive recursive sets of positive integers and vice-versa.

THEOREM I.4. *Using information presentation by methodical informant and the tester-naming relation, the class of primitive recursive languages* IS *identifiable in the limit.*

PROOF. There is an effective enumeration of the primitive recursive functions of one variable, that is, a total recursive function $p_n(x)$ of two variables such that the class of p_n is the class of primitive recursive functions. Define $w(x)$ to be the function which takes 1 into 1 and all other values of x into 0. Then $wp_n(x)$ is an effective enumeration of the characteristic functions of the primitive recursive languages. Let L_n be the language whose characteristic function is wp_n. It will now be shown that identification by enumeration using this enumeration of the primitive recursive languages is effective. First it must be shown that, given any

information sequence up to time t and any n, it can be effectively determined whether L_n satisfies this information sequence. The information sequence will tell us, for each x from 1 through t, whether x is an element of the unknown language L. The fact that $wp_n(x)$ is a recursive 2-place function implies that we can effectively determine the desired information, namely, whether or not each x from 1 through t is an element of L_n. Now it only remains to show that a tester can effectively be found for L_n. This follows from the well known result of recursive theory that, for any recursive 2-place function $wp_n(x)$, there is a total recursive function $\phi(n)$ such that

$$Z_{\phi(n)}(x) = wp_n(x); \tag{I.1}$$

that is, $\phi(n)$ is a tester for L_n.

THEOREM I.5. *Using information presentation by methodical informant and the generator-naming relation, the class of recursive languages* IS NOT *identifiable in the limit.*

PROOF. Let G be an effective identity guessing rule for methodical informant which correctly identifies in the limit every finite language and the complement of every finite language. A recursive language L will be constructed for which the guesses of G will change an infinite number of times.

The information sequence for L will be a semi-infinite sequence of 0's and 1's: $i_1, i_2, \cdots$, where $i_t = 1$ if $t \in L$, $i_t = 0$ if $t \notin L$. An effective rule will be given for constructing this sequence. The construction will proceed in steps. If at the beginning of a step of the construction the construction has so far produced $i_1, \cdots, i_n$, then at the end of the step the information sequence will have been extended to be of the form

$$i_1, \cdots, i_n, 0^x, 1^y, \tag{I.2}$$

where a^b denotes a b-long string of a's. An effective procedure will be given for choosing x and y which will guarantee that the guess made by G at the end of string I.2 will be different from the guess made earlier, at the end of the information string

$$i_1, \cdots, i_n, 0^x. \tag{I.3}$$

It is only necessary to show that a pair (x, y) with this property exists, because then such a pair can be effectively found as follows: Meth-

odically search all pairs of positive integers until such a pair is found. That one can effectively determine, for any pair, whether it has the desired property follows from the fact that G is effective.

Let L_1 be the language whose information sequence is

$$i_1 , \cdots , i_n , 0^\infty, \tag{I.4}$$

where a^∞ denotes the semi-infinite string $a, a, \cdots$. Since L_1 is a finite language, if G is presented with this information sequence it must, after some finite time, continually guess a generator g_1 for L_1. Let $n + x$ be a time at which G guesses g_1.

Let L_2 be the language whose information sequence is

$$i_1 , \cdots , i_n , 0^x, 1^\infty. \tag{I.5}$$

Since L_2 is the complement of a finite language, if G is presented with this information sequence, there must be a time $n + x + y$ at which G guesses a generator g_2 for L_2. Since g_1 and g_2 must differ, this shows the existence of a pair (x, y) with the desired property. Q.E.D.

THEOREM I.6. *Using information presentation by arbitrary text and the tester-naming relation, the class of languages of finite cardinality* IS *identifiable in the limit.*

PROOF. The information sequence $i_1 , i_2 , \cdots$ will be a sequence of positive integers, the range of which is the unknown language L. A suitable identity guessing algorithm is the following: At time t, guess L to consist solely of the numbers which have occurred so far in the information sequence. Since L is finite, there will be a finite time after which all elements of L will have occurred in the information sequence, so that the guesses will be correct. It is a straightforward but tedious exercise to show that there is an effective method for finding a tester for the language which consists of $i_1 , \cdots , i_t$. Q.E.D.

THEOREM I.7. *Using information presentation by primitive recursive text and the generator-naming relation, the entire class of r.e. languages* IS *identifiable in the limit.*

PROOF. As in Theorem I.4, let $p_n(x)$ be an effective enumeration of the primitive recursive functions. The information sequence $i_1 , i_2 , \cdots$ will be the same as the sequence $p_n(1), p_n(2), \cdots$ for some n. Such an n can be effectively identified in the limit by using identification by enumeration. That is, the text describing the unknown language L can be identified in the limit. Since $p_n(x)$ is a recursive function of two variables,

there is a total recursive function $\phi(n)$ such that

$$Z_{\phi(n)}(x) = p_n(x); \tag{I.7}$$

that is, $\phi(n)$ is a generator for L. Q.E.D.

THEOREM I.8. *Using information presentation by recursive text and the generator-naming relation, any class of languages which contains all finite languages and at lease one infinite language L* IS NOT *identifiable in the limit.*

PROOF. We may assume that L is r.e. since, otherwise, it would not have a generator and the theorem would follow immediately. It can be shown by straightforward methods that there is a recursive sequence of positive integers $a_1, a_2, \cdots$ which ranges over L without repetitions. Suppose G is an effective identity guessing rule which identifies generators for all finite languages in the limit from recursive text. A recursive text for L will now be constructed which will cause G to change its guess an infinite number of times. This text will be of the form

$$i_1, i_2, \cdots = a_1^{x_1}, a_2^{x_2}, \cdots . \tag{I.8}$$

As in the proof of Theorem I.5, this text will be constructed in steps. Let

$$t_n = x_1 + \cdots + x_n . \tag{I.9}$$

At the beginning of the nth step, the desired information sequence will have been constructed through time t_{2n-2}. During the nth step, x_{2n-1} and x_{2n} will be effectively chosen in such a manner that the guess made by G at time t_{2n-1} will differ from that at time t_{2n}. As in the proof of Theorem I.5, it is sufficient to show that such a pair (x_{2n-1}, x_{2n}) exists.

Let $\bar{\imath}_n$ signify the desired information sequence through time t_n. The information sequence

$$i_1, i_2, \cdots = \bar{\imath}_{2n-2}, a_{2n-1}^{\infty} \tag{I.10}$$

is a recursive text for the finite language

$$L_1 = \{a_1, \cdots, a_{2n-1}\}. \tag{I.11}$$

Therefore, there is an x_{2n-1} such that at time t_{2n-1} the guess made by G will be a generator for L_1. Similarly, the information sequence

$$i_1, i_2, \cdots = \bar{\imath}_{2n-1}, a_{2n}^{\infty} \tag{I.12}$$

is a recursive text for the finite language

$$L_2 = \{a_1, \cdots, a_{2n}\}, \tag{I.13}$$

which is different from L_1. For a large enough x_{2n} the guess made by G at time t_{2n} will be a generator for L_2, which cannot be the same as a generator for L_1. Q.E.D.

THEOREM I.9. *Using information presentation by primitive recursive text and the tester-naming relation, any class of languages which contains all finite languages and at least one infinite language L* IS NOT *identifiable in the limit.*

PROOF BY CONTRADICTION. An information sequence $i_1, i_2, \cdots$ is here a sequence of positive integers. A guessing rule consists of a computable function G which determines the guess g_t at time t as a function of the information received by the learner through t:

$$g_t = G(i_1, \cdots, i_t). \tag{I.14}$$

In the terminology of Gold (1965), G determines a limiting recursive functional whose domain is information sequences. It is shown in Theorem 5 of that reference that any limiting recursive functional can be defined by means of a primitive recursive guessing function. It will be assumed that G is primitive recursive. It will also be assumed that L is recursive since, otherwise, L cannot be tester identified and the conclusion of the theorem is immediate. Let $f(x)$ be a primitive recursive function with a range equal to L.

A primitive recursive text i_t will be constructed which contradicts the assumption that G is a suitable guessing rule. A function X_t will also be defined. Let P_t and Q_t signify the following predicates:

$$P_t \equiv [f(X_t) = i_1] \vee \cdots \vee [f(X_t) = i_t] \tag{I.15}$$

$$Q_t \equiv (\exists y \leqq t)\{T[g_t, f(X_t), y] \ \& \ [U(y) = 0]\}, \tag{I.16}$$

where $T(a, x, y)$ is the primitive recursive predicate which says that the Turing machine with Gödel number a, if presented with x as an input, will stop after performing the computation with Gödel number y; and $U(y)$ is a primitive recursive function such that, if y is the Gödel number of a Turing machine computation, then $U(y)$ is the number it produces at its end.

i_t and X_t are simultaneously defined by course-of-values recursion as follows:

$$X_1 = 1 \tag{I.17}$$

$$i_1 = f(1) \tag{I.18}$$

$$X_{t+1} = X_t + 1 \quad \text{if} \quad P_t \tag{I.19}$$

$$= X_t \quad \text{if} \quad \neg P_t \tag{I.20}$$

$$i_{t+1} = f(X_t) \quad \text{if} \quad \neg P_t \quad \text{and} \quad Q_t \tag{I.21}$$

$$= i_t \quad \text{otherwise.} \tag{I.22}$$

The idea behind the construction is this: i_t is designed to generate L, but very slowly. After a finite number of elements of L have been generated, $x_1, \cdots, x_{r-1}$, if i_t repeats x_r long enough the guessing procedure will have to guess a decision procedure for the set $\{x_1, \cdots, x_r\}$, which must reject x_{r+1}. As soon as g_t is known to reject x_{r+1}, i_t starts producing it. Q_t implies $Z_{g_t}(x_{r+1}) = 0$, where $x_{r+1} = f(X_t)$. The details follow.

Case I. $\neg P_t$ holds for only a finite number of t. It will be shown that this implies that $Rng(f)$ is finite. Let P_t hold for all $t \geqq \alpha$. Then, by induction,

[by (I.19)] $\quad t \geqq \alpha \Rightarrow X_t = X_\alpha + (t - \alpha)$

[by (I.22)] $\quad i_t = i_\alpha$

[by (I.15)] $\quad f(X_t) \in \{i_1, \cdots, i_t\}.$

Thus,

$$n \geqq X_\alpha \Rightarrow f(n) \in \{i_1, \cdots, i_\alpha\}.$$

Case II. $\neg P_t$ holds infinitely often, but Q_t holds for only a finite number of t. It will be shown that $Rng(i_t)$ is finite, but if $g_t \rightarrow a$, then there is an $x \notin Rng(i_t)$ such that $Z_a(x) \neq 0$. Choose α large enough so that $\neg Q_t$ holds for all $t \geqq \alpha$. Induction on Eq. (I.22) gives

$$i_t = i_\alpha \quad \text{for all} \quad t \geqq \alpha. \tag{I.23}$$

Thus,

$$Rng(i_t) = \{i_1, \cdots, i_\alpha\}.$$

Let β be large enough that

$$\beta \geqq \alpha \tag{I.24}$$

$$g_t = a \quad \text{for all} \quad t \geqq \beta$$

$$\neg P_\beta \quad \text{holds.} \tag{I.25}$$

Induction on Eqs. (I.20) and (I.15) gives, using (I.25) and (I.23),

$$\begin{aligned} X_t &= X_\beta \quad \text{for all} \quad t \geqq \beta \\ f(X_\beta) &\in \{i_1, \cdots, i_\alpha\}. \end{aligned} \tag{I.26}$$

Let $x = f(X_\beta)$. Now (I.16). (I.24), and (I.26) give

$$\neg(\exists y \leqq t)\{T(a, x, y) \quad \& \quad U(y) = 0\} \quad \text{for all} \quad t \geqq \beta.$$

Thus,

$$Z_a(x) \neq 0.$$

Case III. Q_t holds infinitely often. It will be shown that $Rng(i_t) = Rng(f)$; but if $g_t \to a$, then there is an x such that $x \in Rng(f)$ and $Z_a(x) = 0$. Let α satisfy

$$g_t = a \quad \text{for all} \quad t \geqq \alpha.$$

Let $\beta \geqq \alpha$ such that Q_β holds. Set

$$x = f(X_\beta).$$

Then Q_β gives $Z_a(x) = 0$.

$Rng(i_t) = Rng(f)$ will be shown by contradiction. We know, by (I.18) and (I.21), that

$$f(1) \in Rng(i_t) \subset Rng(f).$$

Let x be the lowest number such that

$$f(X) \notin Rng(i_t)$$

(I.19) and (I.20) show that X_t is monotone increasing and either takes on all values, or is ultimately constant.

Case IIIA. $X_t = X_\alpha$ for all $t \geqq \alpha$. Then (I.19) and (I.20) show that $\neg P_t$ holds for $t \geqq \alpha$. By the assumption in Case III, there is a $\beta > \alpha$ such that $\neg P_\beta$ and Q_β. (I.21) shows that $i_{\beta+1} = f(X_\beta)$. Then $P_{\beta+1}$ holds, since $X_{\beta+1} = X_\beta$ by the assumption of Case IIIA.

Case IIIB. α is the last t such that $X_t = X$. Then, by (I.19), P_α must hold, i.e.,

$$f(X_\alpha) \in Rng(i_t). \quad \text{Q.E.D.}$$

APPENDIX II

Definitions of Some of the Technical Terminology

Turing machines are a special class of algorithms which are precisely defined, so that they can be investigated mathematically, but are be-

lieved to be perfectly general in the following sense. Given any computational rule which we would intuitively accept as an effectively defined algorithm, the function defined by this algorithm is also defined by some Turing machine. The *recursive functions* are those functions which can be defined by Turing machines. The inputs to a Turing machine may be considered to be either strings or positive integers, and the same is true of its outputs.

The *primitive recursive algorithms* are a special class of algorithms which are not general in the sense of Turing machines, but are general enough to include all algorithms ordinarily constructed. *Primitive recursive functions* are functions which can be defined by primitive recursive algorithms.

A *decision procedure* for a language L is an algorithm defined on strings such that the result of using the algorithm is 1 or 0, depending on whether the string it starts with is an element of L or not. A *generator* for L is an algorithm which takes positive integers into strings such that the range of the function it determines is exactly L.

L is called *recursively enumerable* (r.e.) if there is a generator for it, *recursive* if there is a decision procedure for it, *primitive recursive* if there is a primitive recursive decision procedure for it, and *regular* if there is a decision procedure for it which can be computed by a finite state automaton.