ADAPTIVE RECOGNIZERS

by

William Sakoda

Memorandum No. ERL-M500

February 1975

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

ADAPTIVE RECOGNIZERS

William Sakoda

Contents

ADAPTIVE RECOGNIZERS[†]

William Sakoda

Computer Science Division
Department of Electrical Engineering and Computer Science
and the Electronics Research Laboratory
University of California, Berkeley 94720

February 1975

1. Introduction

There are numerous interesting families of formal languages (e.g. regular, context-free) for which the problem of identification in the limit admits a trivial solution by enumeration. Since enumeration is computationally inefficient, one is led to ask whether the task of identifying such families may be accomplished more efficiently. The purpose of this report is to present some preliminary results in this area.

We begin by defining a class of inductive inference machines, the adaptive recognizers, which are suitable for analyzing the complexity of inferring such languages. After characterizing the classes of languages which can be inferred by adaptive recognizers, we will show that undecidability of the equivalence problem for a particular family of languages is sufficient to rule out the existence of an efficient adaptive recognizer for that family. We will then have the following as a corollary: Let r be any adaptive recognizer which can identify all context-free languages. Then the amount of data consumed by r before correctly identifying the language generated by an arbitrary context-free grammar G is not bounded above by any recursive function of G.

[†]This research sponsored by National Science Foundation grant DCR72-03725-A02.

2. Adaptive Recognizers

2.1 Preliminary Notation

We will use the symbol $\mathbb{N}$ to denote the set of natural numbers; that is, $\mathbb{N} = \{0,1,2,\ldots\}$. An object will be said to be an integer if and only if it is a member of $\mathbb{N}$. A language is a subset of $\mathbb{N}$.

We assume familiarity with some concepts from elementary recursive function theory. Our notation for recursive functions will follow that in [6].

2.2 Definition of the Model

Adaptive recognizers are machines which attempt to identify languages on the basis of finite samples from the languages. They differ from the usual rule inference models [1,4] in their mode of identification: an adaptive recognizer demonstrates its identification of a language by performing as a recognizer for that language.

Before proceeding, some notation for manipuating sequences will be useful.

2.2.1 Definition

(1) Let $x_1,x_2,\ldots,x_n$ be integers, with $n \geq 0$. Then $\langle x_1,x_2,\ldots,x_n\rangle$ denotes an integer which encodes the ordered sequence with elements $x_1,x_2,\ldots,x_n$, according to some fixed encoding.(1) We will usually use vectored variables to range over such sequence numbers. For

$$\vec{s} = \langle x_1,\ldots,x_n\rangle$$

and

$$\vec{t} = \langle y_1,\ldots,y_m \rangle ,$$

we let

$$\vec{s}\cdot\vec{t} = \langle x_1,\ldots,x_n,y_1,\ldots,y_m \rangle .$$

(2) Let $n \in \mathbb{N}$. Let $x_1,\ldots,x_n \in \mathbb{N}$. Let $b_1,\ldots,b_n \in \{0,1\}$. Then $\vec{s} = \langle\langle x_1,b_1\rangle,\langle x_2,b_2\rangle,\ldots,\langle x_n,b_n\rangle\rangle$ is a sample from language L if and only if, for each $1 \leq i \leq n$,

$$b_i = \begin{cases} 0 & \text{if } x_i \notin L \\ 1 & \text{if } b_i \in L . \end{cases}$$

The set $\{x_i \mid 1 \leq i \leq n\}$ is the base of the sample $\vec{s}$. The size of sample $\vec{s}$, denoted $sz(\vec{s})$, is defined to be $\max(\{0\} \cup \{x_i \mid 1 \leq i \leq n\})$. The length of sample $\vec{s}$ is the cardinality of the base of $\vec{s}$.

(3) $S = \{\vec{s} \mid (\exists \text{ language } L)$ such that $\vec{s}$ is a sample from $L\}$. S is the set of all samples. Note that S is recursive.

We can now indicate the difference between adaptive recognizers and the usual rule-inference machines. Let $\vec{s}$ be a sample from language L_o. A rule-inference machine would use $\vec{s}$ to attempt to produce a name (say, a partial recursive index) for L_o. An adaptive recognizer uses $\vec{s}$ to attempt to function as a recognizer for L_o. Thus, given $\vec{s}$ and an integer x, an adaptive recognizer generates a guess as to whether x is in the language L_o.

2.2.2 Definition

An adaptive recognizer is a function $r: S \times \mathbb{N} \to \{0,1\}$ such that:

(a) r is partial recursive, with $dom(r) = S \times \mathbb{N}$;

and (b) (permutation independence): If $\vec{s}$, $\vec{t}$ are samples and $\vec{s}$ is

a permutation of $\vec{t}$, then $r(\vec{s},x) = r(\vec{t},x)$, for all $x \in \mathbb{N}$.

The restriction of permutation independence is made to force r to base its guesses entirely on the membership information contained in sample $\vec{s}$, and not on the order in which this information appears in $\vec{s}$.

2.2.3 Definition

$\vec{s} \in S$ is an r-primer for language L if, for any sample $\vec{e}$ from the language L and any $x \in \mathbb{N}$,

$$r(\vec{s}\bullet\vec{e},x) = \begin{cases} 0 & \text{if } x \notin L \\ 1 & \text{if } x \in L \,. \end{cases}$$

r is said to recognize (equivalently, identify) L if there exists an r-primer for L. R_r is by definition the set of languages recognized by r.

Since primers are samples, they inherit the notion of size defined for samples in 2.2.1(2). The size of a smallest r-primer for a particular language L is a useful measure of the amount of data required by r to identify L.

2.3 Recursively Indexed Families of Languages

Our interest in adaptive recognizers arises from their close connection with recursively indexed families of languages. Intuitively, we want the recursively indexed families to encompass exactly the classes of languages which can be identified in the limit by enumeration.

2.3.1 Definition

A family F of languages is said to be recursively indexed if there is a recursive $\psi: \mathbb{N} \to \mathbb{N}$ which enumerates at least one index for the characteristic function of each language in F, and only such indices.

Whenever ψ is a recursive indexing, we let $L_i^{\psi} = \{x \in \mathbb{N} \mid \phi_{\psi(i)} = 1\}$. Thus, L_i^{ψ} is that language for which $\phi_{\psi(i)}$ is the characteristic function.

2.3.2 Examples of Recursively Indexed Families

(i) The class of languages having primitive recursive characteristic functions is recursively indexed.

(ii) By establishing a suitable identification between $\mathbb{N}$ and $(0,1)^*$, the notion of recursive indexing may be extended to families of subsets of $(0,1)^*$.[2]

Assuming that such an identification has been established, we can then say that the families of regular, context-free, and context-sensitive subsets of $(0,1)^*$ are each recursively indexed. In each of these families, a recursive indexing can be obtained by using the fact that there is a recursive enumeration $G_0, G_1, G_2, \ldots$ of Gödel numbers for the class of grammars in question, such that the predicate $x \in L(G_i)$ is decidable uniformly in x and i.

(iii) More generally, if L is a recursive subset of $(0,1)^*$, the family of subsets of $(0,1)^*$ in the principal AFL generated by L can be recursively indexed.

2.4 Summary of Results

Our first result is a characterization of the families of languages which can be identified by an adaptive recognizer.

Theorem 1. Let F be a family of languages. Then the following are equivalent.

1) There is an adaptive recognizer which can identify at least every language in F.

2) F is contained in a recursively indexed family.

3) Every language in F is h-easy[3] for some fixed recursive $h: \mathbb{N} \to \mathbb{N}$.

The second result yields a condition on a recursively indexed family F which is sufficient to rule out the existence of an efficient adaptive recognizer identifying every language in F. Some notation is necessary before proceeding.

2.4.1 Definition

Let ψ be a recursive indexing. The equivalence problem for ψ is that function $e: \mathbb{N} \times \mathbb{N} \to \{0,1\}$ which is defined by

$$e(i,j) = \begin{cases} 0 , & L_i^{\psi} \neq L_j^{\psi} \\ 1 , & L_i^{\psi} = L_j^{\psi} . \end{cases}$$

The equivalence problem for ψ is decidable just in case e is recursive.

Theorem 2'. Let $\psi: \mathbb{N} \to \mathbb{N}$ recursively index family F, and suppose the equivalence problem for ψ is undecidable. Let r be any

adaptive recognizer which identifies at least the language in F. Let $\|L_i^\psi\|$ be the size of a smallest r-primer for L_i^ψ. Then $\|L_i^\psi\|$ is not bounded above any any recursive function $m(i)$.

Note that if the restriction that adaptive recognizers be permutation independent is removed, the theorem fails rather dramatically: for any recursive indexing ψ, there is a non-permutation-independent r such that $\|L_i^\psi\| = 1$ for every i! An algorithm for such an r is: "On input $(<<x_1,b_1>,\ldots>,x)$, output 1 if $x \in L_{x_1}^\psi$, 0 otherwise." Then $<<i,\phi_{\psi(i)}(i)>>$ is an r-primer for L_i^ψ.

Corollary - Application to the Context-Free Languages

We remarked in 2.3.2 that the notion of recursive indexing could be extended to languages which are subsets of $(0,1)^*$ by establishing an identification between $(0,1)^*$ and $\mathbb{N}$; note (2) indicates a suitable bijection $\wedge: (0,1)^* \rightarrowtail\!\!\!\!\!\rightarrow \mathbb{N}$. This identification also allows results about adaptive recognizers to be extended to such languages. We will outline this technique.

The following conventions are useful. For $x \in (0,1)^*$, we let $\hat{x}$ denote the image of x under the map $\wedge$. For $A \subseteq (0,1)^*$, $\hat{A} = \{\hat{x} \mid x \in A\}$. Finally, $\hat{CFL} = \{\hat{L} \mid L \text{ is a context free subset of } (0,1)^*\}$.

We can now restate the preceding definitions using this notation. It should be emphasized that we are literally repeating what was said before, using a slightly different notation.

Let $n \geq 0$; let $x_1,\ldots,x_n \in (0,1)^*$; and let $b_1,\ldots,b_n \in \{0,1\}$. Then $\vec{s} = <<\hat{x}_1,b_1>,<\hat{x}_2,b_2>,\ldots,<\hat{x}_n,b_n>>$ is a sample from language $L \subseteq (0,1)^*$ if for each $1 \leq i \leq n$,

$$b_i = \begin{cases} 0 , & x_i \notin L \\ 1 , & x_i \in L . \end{cases}$$

S, the set of all samples, is by definition $\{\vec{s} \mid \vec{s}$ is a sample from some $L \subseteq (0,1)^*\}$.

Sample $\vec{s}$ is an r-primer for $L \subseteq (0,1)^*$ if for every sample $\vec{e}$ from L and every $x \in (0,1)^*$,

$$r(\vec{s} \cdot \vec{e}, \hat{x}) = \begin{cases} 0 , & x \notin L \\ 1 , & x \in L . \end{cases}$$

One modification is in order. If $\vec{s} = \langle\langle \hat{x}_1, b_1 \rangle, \ldots, \langle \hat{x}_n, b_n \rangle\rangle$ is a sample from $L \subseteq (0,1)^*$, it is more natural to define the size of $\vec{s}$ to be the length of the longest string appearing in $\vec{s}$. Thus, $\hat{sz}(\vec{s})$ is by definition $\max(\{0\} \cup \{\text{length}(x_i) \mid 1 \leq i \leq n\})$.

We are now prepared to apply Theorem 2' to the context-free languages. Let ψ be that recursive indexing of $\hat{CFL}$ which is induced by the standard Gödel numbering of the context-free grammars over the terminal alphabet $\{0,1\}$; i.e. $L_i^{\psi} = \hat{L}(G_i)$, where G_i is the i^{th} standard context-free grammar. Since it is undecidable, given CFG's G_i and G_j, whether $L(G_i) = L(G_j)$, it follows that the equivalence problem for ψ is undecidable. Theorem 2' then supports the following result: Let r be any adaptive recognizer which can identify at least $\hat{CFL}$. Let $\hat{\Vert} G_i \hat{\Vert}$ denote the smallest r-primer (in the sense of $\hat{sz}$) for $L(G_i)$. Then there is no algorithm which, given an arbitrary G_i, will compute an upper bound on $\hat{\Vert} G_i \hat{\Vert}$.

3. Proofs

3.1 Characterization Theorem for Adaptive Recognizers

The following lemma records an important property of recursively indexed families. Indeed, the definition of recursive indexing was chosen specifically to ensure that this property held.

3.1.1 Lemma

Let ψ be a recursive indexing. Then there exists, uniformly effectively in ψ, a recursive characteristic function for the relation $\lambda x,i[x \in L_i^{\psi}]$.

Proof. By definition of L_i^{ψ}, $\phi_{\psi(i)}$ is the characteristic function for L_i^{ψ}. Therefore $\lambda x,i[\phi_{\psi(i)}(x)]$ is the required characteristic function.

This lemma will be used implicitly in subsequent constructions.

Theorem 1. Let F be a family of languages. Then the following are equivalent.

(1) $F \subseteq R_r$ for some adaptive recognizer r.

(2) $F \subseteq G$ for some recursively indexed family G.

(3) Every language in F is h-easy for some fixed recursive $h: \mathbb{N} \to \mathbb{N}$.

The proof that (2) → (1) involves a construction which will be of use later. We record it below.

3.1.2 Lemma

Let family F of languages be recursively indexed by ψ. Then there exists, uniformly effectively in ψ, an adaptive recognizer r with the following properties:

(1) r recognizes exactly F.

(2) For each $j \in \mathbb{N}$, the following non-effective procedure leads to a primer, $p(j)$, for L_j^ψ.

(i) Pick i least such that $L_i^\psi = L_j^\psi$.

(ii) For each $k < i$, pick an x_k such that $x_k \in L_k^\psi \leftrightarrow x_k \notin L_i^\psi$.

(iii) Let $p(j)$ be a sample from L_j^ψ with base $\{x_k \mid 1 \leq k < i\}$.

Proof. We will construct the required r.

On input $(\vec{s},m)$, r will try to find an $\vec{s}$-consistent hypothesis $L \in F$, and output 0 if $m \notin L$, 1 otherwise. Since F is recursively indexed, this search may be carried out in an orderly fashion by testing $L_0^\psi, L_1^\psi, \ldots$ for $\vec{s}$-consistency. The danger of never finding an $\vec{s}$-consistent L_i^ψ may be handled by bounding the number of L_i^ψ tested by the length ℓ of the sample $\vec{s}$.

Thus, if $L_0^\psi, \ldots, L_{\ell+1}^\psi$ are all $\vec{s}$-inconsistent, we set

$$r(\vec{s},m) = \begin{cases} 0, & m \notin L_0^\psi \\ 1, & m \in L_0^\psi \end{cases}.$$

If, on the other hand, one of $L_0^\psi, \ldots, L_{\ell+1}^\psi$ is $\vec{s}$-consistent, we pick the least n such that L_n^ψ is $\vec{s}$-consistent, setting

$$r(\vec{s},m) = \begin{cases} 0, & m \notin L_n^\psi \\ 1, & m \in L_n^\psi \end{cases}.$$

It's clear that r is defined on all of $S \times \mathbb{N}$ and satisfies permutation independence. Since for any $\vec{s}$, $(\lambda x[r(\vec{s},x)]) = \chi_L$ for some $L \in F$, we have $R_r \subseteq F$. Finally, to verify property (2) of the lemma, let $j \in \mathbb{N}$, and let $p(j)$ be a sample from L_j^{ψ} with base $\{x_k \mid 1 \leq k < i\}$ supplied by the construction indicated in (2). We need to show that $(\lambda x[r(p(j)\cdot\vec{e},x)]) = \chi_{L_j^{\psi}}$ for any sample $\vec{e}$ from L_j^{ψ}. To do this, it will certainly suffice to show that, when given $p(j)\cdot\vec{e}$ as input, r selects L_j^{ψ} as its hypothesis. But the information contained in $p(j)$ is sufficient to cause all L_k^{ψ}, $k < j$ to be rejected; the length of $p(j)$ is sufficient to allow the search to reach at least to L_j^{ψ}; and L_j^{ψ} cannot be rejected, since it is certainly consistent with $p(j)\cdot\vec{e}$. Thus $p(j)$ is a primer for L_j^{ψ}, and (2) has been verified. (2) now implies that $R_r \supseteq F$, which fact, combined with the reverse containment proved earlier, yields $R_r = F$.

Proof of Theorem 1. The equivalence of (2) and (3) is easily verified. We will show that (1) and (2) are equivalent.

(1) → (2). Let r be an arbitrary adaptive recognizer. We will construct a ψ which recursively indexes a superset of R_r.

Roughly speaking, what we are trying to do is establish an effective correspondence between the integers and the languages in R_r. This can be accomplished by exploiting a natural correspondence between S and R_r. The latter correspondence is the following. To each $\vec{s} \in S$, associate $X_{\vec{s}} = \{m \in \mathbb{N} \mid r(\vec{s},m) = 1\}$. This association is sufficiently effective since, given $\vec{s}$, we can find an algorithm for $\chi_{X_{\vec{s}}}$: on input y, the algorithm simply evaluates $r(\vec{s},y)$. To see that this scheme does indeed manage to assign some $\vec{s} \in S$ to every $L \in R_r$, note that

if $L \in R_r$, there is an r-primer, $\vec{p}$, for L. Then clearly $L = X_{\vec{p}}$.

We may end up indexing a proper superset of F, as unless $\vec{s}$ is a primer, there is no reason to expect that $X_{\vec{s}} \in R_r$.

To finish up, let $s: \mathbb{N} \mapsto\!\!\!\!\rightarrow S$ be a recursive bijection between $\mathbb{N}$ and S. Then an appropriate algorithm for ψ is: "On input x, output the index of an algorithm which computes $\lambda y[r(s(x),y)]$". Then ψ is a recursive indexing, since r is 0-1 valued and convergent on $S \times \mathbb{N}$. ψ indexes at least R_r, since if $L \in R_r$ and $\vec{p}$ is an r-primer for L, then $\psi(s^{-1}(\vec{p}))$ is an index for the characteristic function of L.

(2) → (1). If G is recursively indexed, Lemma 3.1.2 supplies an r which recognizes exactly G.

3.2 The Role of the Equivalence Problem

We now turn to a proof of the second result. Roughly stated, this result is that undecidability of the equivalence problem for a recursive indexing ψ implies that for any adaptive recognizer, r, which identifies at least all the L_i^{ψ}, it is difficult to generate r-primers for ψ. Our method for measuring said difficulty will be to test for the existence of an effective procedure which, given an arbitrary integer i, will produce an r-primer for L_i^{ψ}.

3.2.1 Definition

Let ψ be a recursive indexing of family F, and let r be an adaptive recognizer identifying at least F. Then $p: \mathbb{N} \to \mathbb{N}$ is a generator of r-primers for ψ if, for all $i \in \mathbb{N}$, p(i) is an r-primer for L_i^{ψ}.

We will prove a slightly stronger version of Theorem 2' of Section 2.4, namely:

Theorem 2. Let ψ be a recursive indexing of $\mathcal{F}$.

(a) Let r be an adaptive recognizer identifying at least $\mathcal{F}$, and let $p: \mathbb{N} \to \mathbb{N}$ be a generator of r-primers for ψ. Then the equivalence problem for ψ is recursive in p.[4]

(b) There is an adaptive recognizer, r, such that

(i) r recognizes exactly $\mathcal{F}$; and

(ii) there is a generator of r-primers for ψ which is recursive in the equivalence problem for ψ.

The following lemma is the basis for our proof of part (a) of the theorem.

3.2.2 Lemma

Primers for distinct languages are inconsistent. That is, let L, L' be languages recognized by adaptive recognizer r. Let $\vec{q}$ be an r-primer for L. Let $\vec{q}'$ be an r-primer for L'. If $L \neq L'$, then either

(*) $\vec{q}$ is not a sample from L'; or

(*') $\vec{q}'$ is not a sample from L.

Proof. Suppose the lemma is false. Then for some $L \neq L'$, both (*) and (*') fail. We will use this to get a contradiction.

Pick an $x \in \mathbb{N}$ witnessing the fact that $L \neq L'$ (say, $x \in L$ $x \notin L'$; the other case will follow by symmetry). Then

(1) $1 = r(\vec{q}\cdot\vec{q}',x)$ ($\vec{q}'$ is a sample from L since (*') is false; therefore guess $r(\vec{q}\cdot\vec{q}',x)$ must be correct, since $\vec{q}$ is a primer for L.)

(2) $= r(\vec{q}'\cdot\vec{q},x)$ (permutation independence)

(3) $= 0$ (Argue as in line 1, exchanging primed and unprimed variables everywhere.)

Contradiction, as required.

Corollary to Lemma (criterion for language equivalence). Let the notation be as in the lemma. Then

$$L = L' \leftrightarrow (\vec{q} \text{ is a sample from } L' \text{ and } \vec{q}' \text{ is a sample from } L) .$$

Proof of Corollary. $\rightarrow$ is obvious.

$\leftarrow$ is the contrapositive of the implication of the lemma.

Armed with this corollary, its now easy to prove the theorem.

Proof of Theorem 2.

(a) To decide recursively in p, given integers i, j, whether $L_i^{\psi} = L_j^{\psi}$, proceed as follows:

(1) Using the oracle for p, compute

$$\vec{q} = p(i)$$
$$\vec{q}' = p(j) .$$

(2) By the corollary to the lemma,

$$L_i^{\psi} = L_j^{\psi} \leftrightarrow (\vec{q} \text{ is a sample from } L_i^{\psi} \text{ and } \vec{q}' \text{ is a sample from } L_j^{\psi}) .$$

Thus, if we could effectively test whether the right-hand side of the

equivalence held, we would be done. It is clear that if we had a method for evaluating the characteristic functions of L_i^{ψ} and L_j^{ψ}, this test could be performed. Such a method is indeed available, since

$$\lambda x[r(\vec{q},x)] = \chi_{L_i^{\psi}}$$

and

$$\lambda x[r(\vec{q}',x)] = \chi_{L_j^{\psi}} .$$

(b) Let r be the adaptive recognizer supplied by Lemma 3.1.2. Property (1) of the lemma implies that requirement (i) of the theorem is satisfied. Given an oracle for the equivalence problem for ψ, the non-effective procedure (2) of the lemma for generating r-primers for ψ becomes effective, thus establishing claim (ii) of the theorem.

Corollary to Theorem 2 (Theorem 2' of Section 2.3). Let $\psi: \mathbb{N} \to \mathbb{N}$ recursively index family F, and suppose the equivalence problem for ψ is undecidable. Let r be any adaptive recognizer which identifies at least the languages in F. Let $\|L_i^{\psi}\|$ be the size of a smallest r-primer for L_i^{ψ}. Then $\|L_i^{\psi}\|$ is not bounded above by any recursive function m(i).

Proof of Corollary. Suppose, to the contrary, that there is a recursive m(i) which bounds $\|L_i^{\psi}\|$ from above. We will use this to get a contradiction.

The point is that using m, we can construct a recursive generator, p, of r-primers for ψ: On input i, p simply outputs a sample from L_i^{ψ} with base $\{x \mid x \leq m(i)\}$. We must verify that p(i) is a primer for

L_i^{ψ}. Since $m(i)$ is an upper bound on the integers appearing in the base of some r-primer for L_i^{ψ}, there is an r-primer $\vec{q}_i$ for L_i^{ψ} and a sample $\vec{f}_i$ from L_i^{ψ} such that $p(i)$ is a permutation of $\vec{q}_i \circ \vec{f}_i$. Then

$$r(p(i)\circ\vec{e},x) = r(\vec{q}_i\circ(\vec{f}_i\circ\vec{e}),x) \quad \text{(permutation independence)}$$

$$= \begin{cases} 0\ , & x \notin L_i^{\psi} \\ 1\ , & x \in L_i^{\psi} \end{cases} \quad \text{(since } \vec{q}_i \text{ is an r-primer for } L_i^{\psi}\text{)}\ .$$

This being true for arbitrary $i \in \mathbb{N}$, p is a recursive generator of r-primers for ψ.

Now by part (a) of Theorem 2, the equivalence problem for ψ is recursive in p. But as p is recursive, this means that the equivalence problem for ψ is recursive outright, contradicting the undecidability of the equivalence problem for ψ.

4. Connections With Other Models of Inductive Inference

We will conclude by indicating a connection with a rule-inference model.

4.1 Definition

(1) A rule inference machine is a function $M: S \to \mathbb{N}$ such that

(i) M is partial recursive with $dom(M) = S$; and

(ii) for all $\vec{s} \in S$, $\phi_{M(\vec{s})}$ is 0-1 valued and defined everywhere.

Thus, $\phi_{M(i)}$ is always the characteristic function of some subset of $\mathbb{N}$.

(2) Rule inference machine M is permutation-independent if for any $\vec{s}, \vec{t} \in S$ such that $\vec{s}$ is a permutation of $\vec{t}$, $\phi_{M(\vec{s})} = \phi_{M(\vec{t})}$.

(3) Let $L \subseteq \mathbb{N}$ and let $\vec{s}$ be a sample from L. $\vec{s}$ is an M-primer for matching L [5] if for every sample $\vec{e}$ from L, $\phi_{M(\vec{s}\cdot\vec{e})} = \chi_L$. M is said to match L if and only if there is an M-primer for matching L. The notion of a generator of M-primers is defined in the obvious way.

The following proposition establishes the connection between adaptive recognizers and permutation-independent rule-inference machines.

4.2 Proposition

(a) Let M be a permutation independent rule-inference machine. Then there is an adaptive recognizer r such that for any $L \subseteq \mathbb{N}$ and any sample $\vec{s}$ from L, $\vec{s}$ is an M-primer for matching L if and only if $\vec{s}$ is an r-primer for L.

(b) Conversely, let r be any adaptive recognizer. Then there is a permutation independent rule inference machine M such that for any $L \subseteq \mathbb{N}$ and any sample $\vec{s}$ from L, $\vec{s}$ is an r-primer for L if and only if $\vec{s}$ is an M-primer for matching L.

Proof. (a) $r(\vec{s},x) = \phi_{M(\vec{s})}(x)$ is suitable.

(b) On input $\vec{s}$, M outputs the index of the following program: "On input x, output the value $r(\vec{s},x)$."

Thus, Theorems 1 and 2 may be applied to permutation-independent rule-inference machines by replacing:

"adaptive recognizer"	by	"permutation-independent rule-inference machine"
"r"	by	"M"
"r-primer"	by	"M-primer" .

Footnotes

(1) For example, the following is suitable. Let p_i denote the i^{th} prime. Let $\langle\ \rangle = 1$ and, for $n \geq 1$, let $\langle x_1,\ldots,x_n\rangle = \prod_{i=1}^{n} p_i^{(x_i+1)}$.

(2) The following map $\wedge: (0,1)^* \mapsto\!\!\!\!\rightarrow \mathbb{N}$ works. Let $<$ be a total ordering of $(0,1)^*$ in order of increasing length of strings, ties between strings of equal length being broken by a lexicographic ordering. For $x \in (0,1)^*$, let $\wedge(x)$ be the number of strings preceding x in the $<$ ordering.

(3) Total recursive function ϕ_i is h-easy if $\Phi_i(x) \leq h(x)$ for almost all $x \in \mathbb{N}$, where $\Phi_i(x)$ denotes the running time of ϕ_i on input x. See [2,5] for details.

(4) For $f, g: \mathbb{N} \to \mathbb{N}$, f is <u>recursive in</u> g if, given an "oracle" for computing the function g, f may be computed effectively. See [6] for details.

(5) The notion of matching is due to Feldman [3].

References

[1] L. Blum and M. Blum, "Inductive Inference: A Recursion Theoretic Approach," Memorandum ERL-M386, Electronics Research Laboratory, University of California, Berkeley, March 1973.

[2] M. Blum, "A Machine-independent Theory of the Complexity of Recursive Functions," J. ACM 14 (2), April 1967, pp. 322-336.

[3] J.A. Feldman, "Some Decidability Results on Grammatical Inference and Complexity," Artificial Intelligence Memorandum 93.1, Computer Science Department, Stanford University, May 1970.

[4] E. Mark Gold, "Language Identification in the Limit," Information and Control 10, pp. 447-474.

[5] J. Hartmanis and J.E. Hopcroft, "An Overview of the Theory of Computational Complexity," J. ACM 18 (3), July 1971, pp. 444-475.

[6] H. Rogers, Jr., Theory of Recursive Functions and Effective Computability, McGraw-Hill, New York, 1967.