# A Universal Finite Memory Source

Marcelo J. Weinberger, *Member, IEEE*, Jorma J. Rissanen, *Senior Member, IEEE*, and Meir Feder, *Senior Member, IEEE*

***Abstract*— In this paper an irreducible parameterization for a finite memory source is constructed in the form of a tree machine. A universal information source for the set of finite memory sources is constructed by a predictive modification of an earlier studied algorithm—Context. It is shown that this universal source incorporates any minimal data-generating tree machine in an asymptotically optimal manner in the following sense: the negative logarithm of the probability it assigns to any long typical sequence, generated by any tree machine, approaches that assigned by the tree machine at the best possible rate.**

***Index Terms*—Universal coding, Context algorithm, finite memory sources, sequential decision, stochastic complexity, prediction.**

## I. Introduction

FINITE memory sources are distinguished by the property that the conditional probabilities of a symbol, given all the past observations, actually depend only on a fixed smallest number $k$ of contiguous past observations. Hence, technically, such sources are Markovian, the number $k$ defining the order of the process. However, when trying to fit Markov models to the data by estimating the conditional probabilities at the $d^k$ states, where $d$ is the alphabet size, a number of difficulties arise. First, we have to deal with the familiar explosive increase in the number of states and the fitted parameters if we increase the order of the Markov model to find the best fit. Secondly, even when the order $k$ is minimal, the probability parameters defining the process are not necessarily irreducible, and hence they cannot be efficiently estimated. This is because there are in general equivalent states having identical conditional probabilities. Again, if we attempt to remove the redundant parameters by "lumping" together the equivalent states, we may arrive at the disturbing situation that the result is not a finite state machine implementation of the process, let alone one of a Markov type.

There is, however, a different but equally simple implementation of the process, which is described by an irreducible set of probability parameters. We call it a "tree source," and define it formally in Section II after a discussion of finite memory sources. In an incomplete form, such a tree was utilized in the data compression algorithm Context, introduced in [1] and modified and generalized in [2] and [5]. This algorithm has two main stages, a tree that grows with a string for gathering the conditional symbol statistics, and a rule for selecting the optimal context on which to condition each observed symbol. Inspired by the Context selection rule in [5] we describe in this paper a sequential modeling scheme based on a predictive rule, which overcomes the state explosion problem and is both simple to implement and to analyze. Most importantly, as it is fully sequential it actually defines in a practical manner a *universal* finite memory source, which incorporates asymptotically optimal estimation of the best minimal data generating tree source that fits the data.

Let us further elaborate on the concept of "universal source." The idea of a universal data compression algorithm goes back to Kolmogorov. Here we go a step further and introduce the notion of a *universal information source* which is close, in an asymptotic sense, to any source in the class with respect to which it is universal. It is meant to replace any source in the class wherever models are needed; we discuss some of the applications in the last section. Another more general context-based universal random process for time series and chaotic processes was discussed in [4]. The sense of universality in a model is akin to a universal Turing machine which can imitate any special-purpose machine; i.e., a program. Since the particular tree machines are here given or presented to us only through the data they generate, the universal information source imitates each such machine exactly only in the limit. However, it assigns to every long typical string, generated by any Markov source, a probability which is almost as large as the probability of the string assigned by the data generating source. This means that for long typical strings, the model provided by the universal source behaves like the "true" system for all tasks we wish to use the model for, such as coding, prediction, and decision in general. In particular, the universal source with arithmetic coding provides a universal data compression algorithm such that the mean per symbol code length not only approaches the entropy of any data generating source but does it at an optimal rate, [2].

More specifically, let $P(x^n)$ and $P_T(x^n)$ denote the probabilities assigned to the string $x^n$ by the universal source and by any data generating finite-memory source with $K$ sets of equivalent states, respectively. Using techniques developed in [6] we prove a theorem stating a strong asymptotic optimality of the universal source both in the almost sure sense and in

Manuscript received September 21, 1992; revised October 11, 1994. The material in this paper was presented in part at the 1993 IEEE International Symposium on Information Theory, San Antonio, TX, January 17–22, 1993. This research was done while one of the authors (M. J. Weinberger) was with the Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel.

M. J. Weinberger is with Hewlett-Packard Laboratories, Building 3U, POB 10490, Palo Alto, CA 94303 USA.

J. J. Rissanen is with IBM Almaden Research Center, San Jose, CA 95120-6099 USA.

M. Feder is with the Department of Electrical Engineering–Systems, Tel-Aviv University, 69978 Tel-Aviv, Israel.

IEEE Log Number 9410422.

0018–9448/95$04.00 © 1995 IEEE

the mean, namely,

$$\frac{1}{n}\log\frac{P_T(X^n)}{P(X^n)} \le \frac{K(d-1)}{2n}\log n + O(1/n). \quad (1)$$

The set of the tree sources satisfies also the conditions in [2], which imply that the strict reverse inequality in (1) holds for *any* source $P(X^n)$, however arrived at, and all $T$, with its parameters in a compact $K(d-1)$-dimensional set with nonempty interior, except in a subset of measure zero. The right-hand side, then, represents the asymptotically optimal *model complexity* in terms of the code length needed to encode the model which leads to the asymptotically shortest code length for the data. This means that regardless of which estimation method one employs it is impossible, at least asymptotically, to "learn" the data-generating mechanism, the tree machine, better than is done by using the universal algorithm Context. In [6] a variant of algorithm Context was studied in which the context selection was done using hypothesis testing, modified by data-length-dependent thresholds (the right-hand side of (1)) given in the theory of stochastic complexity. The resulting algorithm was shown to be asymptotically optimal for the class of FSMX sources [2] defined in Section II, which is a subclass of the tree sources. However, unlike our results here, the implementation of the context selection rule becomes impractical in case an upper bound on the length of the contexts is not available, a case that needs to be covered for a complete universality.

The universal source presented in this paper is based on the explicit sequential estimation of a finite memory model, which is used at each time instant to define the probability distribution assigned to the next symbol. This plug-in approach not only assigns to the data a probability that is close to that assigned by any finite memory source, but it also tracks the best model that fits the data. It generates a random process whether or not the data have been generated by a finite memory source, and (1) holds for long typical sequences generated by such sources. As discussed in [7], it turns out that (1) does not hold for *every* sequence. An alternative approach, in which a (Bayesian) "mixture" of all models in a class is used to define a universal random process, is investigated in [7]. Although the "mixture" approach lacks the explicit model estimation, it is shown in [7] that the probability it assigns satisfies (1) pointwise, i.e., for every sequence. A practical implementation of this approach for tree sources, however, requires an upper bound on the length of the contexts as in [6].

The remainder of this paper is organized as follows. In Section II, we discuss finite-memory sources and formally define the tree sources. In Section III, the algorithm Context is presented as a universal source for the finite memory class. The main results, which establish the universality of the source, are stated in Section IV and proved in Appendixes I and II. Finally, applications of the universal source beyond compression are discussed in Section V.

## II. Finite -Memory Sources

An information source over a finite alphabet $A$ of $d$ symbols is defined by a family of probability measures $P_n(x^n)$, $n = 0, 1, \cdots$, each $P_n$ defined on the set $A^n$ of all strings $x^n = x_1 \cdots x_n$ of length $n$, such that the marginality condition

$$\sum_{x\in A} P_{n+1}(x^n x) = P_n(x^n) \quad (2)$$

holds for all $n$; here $x^n x = x_1 \cdots x_n x$, and $P_0(x^0) = 1$, where $x^0$ is the empty string, also written as $\lambda$. The additional requirement of stationarity may be expressed by the symmetric condition

$$\sum_{x\in A} P_{n+1}(x x^n) = P_n(x^n) \quad (3)$$

for all $n$. As a rule, we write $P_n(x^n) = P(x^n)$ by omitting the subindex. Note that (2) and (3) define a consistent and stationary measure $P$.

In applications the probability function $P(x^n)$ must be such that it can be easily constructed. One of the familiar ways to do it is in terms of the conditional probability function

$$P(x|x^n) = P(x^n x)/P(x^n)$$

which in view of (2) is well-defined. Then, it is required to satisfy the condition

$$P(x|x^n) = P(x|s(x^n)) \quad (4)$$

for all $x \in A$, where $s(x^n)$ is a function with finite range $S$, a fixed (with respect to $n$) state space. One way of specifying the function $s(x^n)$ is by a finite state machine (FSM) as follows: First, $s(\lambda) = s_0$ selects an initial state. Then, recursively

$$s(x^{n+1}) = f(s(x^n), x_{n+1}) \quad (5)$$

where $f : S \times A \mapsto S$ is the state transition map of the machine. The source is then defined by the $K \times (d-1)$ conditional probabilities and the function $f(s, x)$ together with the initial state, where $K$ is the number of states. Such sources are called finite-state-machine-generated sources or FSM sources for short.

However, fitting FSM sources to data is still a complex matter, especially when one wants to search through machines with different numbers of states, because of the large number of possible state transition functions to be explored. There is an important subclass of sources, which we call *Finite Memory* or *Tree* sources, which can be estimated much more conveniently using a modification of the algorithm Context, [1], [2], and [5], described below. Although strictly speaking these random processes are still Markovian, they cannot be efficiently estimated by fitting a Markov machine to the data. The distinguishing property of a Markov process is that the function $s(x^n)$ not only has a finite fixed range $S$ and satisfies (5) but it also is of the form

$$s(x^n) = x_n \cdots x_{n-k+1} \quad (6)$$

where $k$ is the order of the process, and the reversed string $s(x^n)$ for $n \ge k$ is a state.[1]

[1]Here, we distinguish between a Markov source, namely a special case of FSM source satisfying (6), and a Markov chain. In fact, any FSM source is a Markov chain on $S$.

In view of the crucial requirement (4) it is clear that the right-hand side string in (6) should be as short as possible. Indeed, if for some $i$, $0 \leq i < k$, some string $x_{n-i+1} \cdots x_n$, and all strings $x_{n-k+1} \cdots x_{n-i}$

$$P(x|x_{n-i+1} \cdots x_n) = P(x|x_{n-k+1} \cdots x_n)$$

for all $x \in A$, then the set of states defined by the same suffix $x_{n-i+1} \cdots x_n$ would be equivalent and could be lumped together and represented by the suffix without any change in the process. The problem is that the resulting set of the shortest strings, which now need not be of the same length, may not admit an FSM description let alone a Markov one to generate the process. (Such an example is given in the last paragraph of this section.) On the other hand, if the states are not collapsed, the estimation of the distribution $P(x|x_{n-i+1} \cdots x_n)$ repeatedly in each of the equivalent states; i.e., from the symbol occurrences following the longer strings $x_{n-k+1} \cdots x_n$, would involve estimation of redundant parameters. These would then have to be estimated from fewer symbol occurrences than in the case where the same distribution is estimated from the symbol occurrences following the shorter string $x_{n-i+1} \cdots x_n$, resulting in suboptimal convergence.

These intuitive ideas can be formalized in terms of the *model complexity*, as the smallest number of bits needed to encode the parameters, which permits a measurement of the effect of the parameter redundancy. The model complexity restricts in a fundamental manner all the tasks the models are used for, often prediction, and it is important to keep it minimal, which can be achieved only when the parameters are nonredundant. To summarize, an efficient estimation of the conditional distributions (4) can only be done by counting the symbol occurrences following the shortest possible strings where (4) and (6) hold, which amounts to pooling the symbol occurrences at the redundant states. However, the set of the shortest strings satisfying (4) may admit no FSM implementation.

Fortunately, there is another type of "machine" that can do the implementation of a Markov source with an equally simple tree architecture, to be described next. Let (4) hold for a function $s(x^n)$ of the form

$$s(x^n) = x_n \cdots x_{0 \vee (n-k+1)} \tag{7}$$

for some $k \geq 0$, not necessarily the same for all strings, where $a \vee b$ denotes the larger of the two numbers; the case $k = 0$ is interpreted as defining the empty string $\lambda = x_0$. Hence, for $n \leq k$, $s(x^n) = x_n \cdots x_0$, so that the function is defined for strings of every length, even for $n = 0$. Any suffix of $x^n$, written in reverse order as $x_n x_{n-1} \cdots$, is called a *context* in which the "next" symbol $x_{n+1}$ occurs. The "terminal contexts" (7) (the shortest contexts satisfying (4)) act as states, but they do not necessarily satisfy (5). And, as we argued above, in fitting models it is the set of terminal contexts rather than the states of an FSM that are important. The difference is particularly dramatic in raster-scan images, where the value of a pixel is influenced mostly by the context defined by its few nearby past pixels. While the number of states of an FSM that would include these relevant pixels would be enormous, including all the pixels in between as well, a state space (7) can be implemented by a tree structure after a suitable reordering of the data scanned prior to $x_{n+1}$. With an FSM, it would be completely hopeless and useless to fit probability parameters to such a huge number of states.

Consider a $d$-ary tree, where the branches are labeled by the symbols in the alphabet. Each context defines a node in the tree reached by the path starting at the root with the branch $x_n$, followed by the branch $x_{n-1}$, and so on. The range $S$ defines a complete[2] subtree $T$ with the set of leaves denoted $S(T) = S$. Note that the use of the same notation for both the set of leaves of $T$ and the set of states of an FSM emphasizes their similarity, but it is not required that the tree model admit an FSM implementation, as given by (5). To specify a stationary source we need in addition a set of probability distributions $\{P(x|s) : s \in S\}$, $P(a|s) > 0$ for all $a \in A$ and $s \in S$. Much as a prefix code permits decoding of codewords without commas, the tree $T$ permits finding a distinguished context (7) for each symbol $x_t$ in the string $x^n$ and an implementation of the probability $P_T(x^n)$ as

$$P_T(x^n) = \prod_{t=0}^{n-1} P(x_{t+1}|s(x^t)) \tag{8}$$

for any string $x^n$. To compute $P_T(x^n)$ it remains to define the conditional probabilities in the right-hand side of (8) for the very first values of $t$, for which the contexts defined by (7) may not be leaves. One way to set these initial conditions is by use of a stationarity constraint. In this case, the conditional distributions at the leaves induce a unique conditional distribution $P(x|w)$ for each internal node $w$ as follows. First, we extend the tree $T$ to the perfectly balanced[3] complete supertree of $T$ having the same height as $T$, and assign the conditional distribution $P(x|s)$ to each leaf $su$ of the extended tree. Each such leaf defines a state of a Markov process. These conditional distributions, which define the state transition probabilities, are positive and, therefore, the corresponding chain is irreducible. Moreover, condition (6) clearly implies aperiodicity and hence ergodicity. Thus there exists a unique stationary distribution over the states. Next, this distribution defines, in turn, a unique probability of each node by the requirement (3) and by (2) a conditional distribution at each node of the tree $T$. For example, take $S = \{1, 01, 000, 001\}$, as the (prefix) set of states (reversed strings). To find the conditional distribution in the internal node 0, we have first $P(0|0) = P(00)/P(0)$ and then

$$P(0|0) = \frac{P(000) + P(001)}{P(000) + P(001) + P(010) + P(011)}.$$

Finally, the stationary probabilities of the triplets are obtained by extending $S$ to a third-order Markov source and finding its invariant distribution.

Although the probability $P_T(x^n)$ is well-defined, it will simplify later analysis if we let the machine define a slightly

[2] A $d$-ary tree is said to be complete if each node either is a leaf or has exactly $d$ offspring.

[3] A complete tree is said to be perfectly balanced if all the paths from the root to the leaves have the same length.

different probability function by changing the initial conditions. Let $s_0$ denote the all-zero leaf in the tree $T$, which plays the same role as an initial state in finite-state machines, and let $\tau$ be the maximum depth of the tree $T$. We then put

$$P_T(x^n|s_0) = \prod_{t=0}^{n-1} P(x_{t+1}|s(0^\tau x^t)) \tag{9}$$

where $0^\tau x^t$ denotes the string $x^t$ padded with $\tau$ initial zeros. This guarantees that the state is well-defined for every $t > 0$, which might not be the case otherwise. Let $\bar{s}$ denote the string $s$ written in reverse. Then, (9) can further be written in terms of sufficient statistics as

$$\log P_T(x^n|s_0) = \sum_{s\in S}\sum_{a\in A} n_{x^n}(a|s)\log P(a|s) \tag{10}$$

where $n_x(a|s)$ denotes the number of times the string $\bar{s}a$ occurs as a substring in $0^{|s|}x$, less the number of times it occurs as a substring in the string of zeros $0^{|s|}$ given by the length $|s|$ of $s$. This definition of the counts insures that for all $a \in A$ and all strings $x$

$$n_x(a|s) = \sum_v n_x(a|sv) \tag{11}$$

where $sv$ runs through all the leaves of any complete subtree rooted at $s$. Because of the initial edge effect this would not hold exactly if $n_x(a|s)$ were defined to be the number of times $\bar{s}a$ occurs in $x$.

The tree $T$ is called *minimal*, if for every node $w$ in $T$ with all its successors $w\nu$ as leaves, there exist $a, b$, and $c$ in $A$ satisfying $P(a|wb) \neq P(a|wc)$. Clearly, if for some such node $w$ the distributions $P(\cdot|wb)$ are equal for all $b$, we could lump the successors into $w$ and have a smaller complete tree representing the same process. If a minimal tree $T$ admits an FSM implementation, (5), the source is also called an FSMX (generated) source. Not every minimal tree admits such an implementation. Indeed, if $s(x^n) = x_n \cdots x_{n-k}$, then (5) can hold only if the next state $s(x^{n+1})$ is uniquely determined by the string $x_{n-k}\cdots x_n, x_{n+1}$ and hence its length cannot exceed that of $s(x^n)$ by more than one symbol, while no such restriction need hold for the contexts or the lengths of the leaves of a complete tree. The following simple example was given in [6]: Take $S = \{1, 00, 010, 011\}$, where the strings read in reverse are supposed to be the states. What would be the state following an emission of "0" at state 1? Clearly, there is no such state satisfying (5), because the only state of length two is 00 , which cannot be reached from state 1. By contrast, with a tree model, the tree $T$ with $S$ as the set of leaves will parse from the string $\cdots 10$ as long a context as needed to reach a leaf, either 011 if the string is $\cdots 110$, or 010 if the string is $\cdots 010$. The exceptions, of course, are strings shorter than the path to the leaves, for which the tree will assign the probability by the internal nodes as indicated prior to (8). In conclusion, we mention that a minimal tree guarantees that no node $w$, all of whose successors are leaves, can be equivalent to all these successors, which, consequently, cannot be replaced by the father node $w$. Notice that even in a minimal tree there may well be other sets of equivalent leaves, not necessarily siblings, having the same associated conditional probabilities. These equivalent nodes could, in principle, be lumped together thus reducing the number of parameters of the process. However, such a reduced parameterization may no longer admit a simple tree implementation nor a practical construction of a universal source, and we do not discuss such more general parametric representations of Markov sources in this paper.

## III. A Universal Finite-Memory Source

The algorithm Context, introduced in [1] and improved in [2] and [5], provides a practical means to estimate Markov sources in the tree form. The algorithm has two stages, the first for growing a large tree and the second for selecting from that tree a distinguished context to define the function $s(x^t)$ and hence the complete trees $T_t$. The algorithm grows the contexts and updates the occurrence counts by the following rules:

1) Start with the root with its symbol counts all zero. Recursively, having constructed the tree $\mathcal{T}_t$ (which may be incomplete) from $x^t$, read the symbol $x_{t+1}$. Climb the tree according to the path defined by $x_t x_{t-1}\cdots$, and increment the count of symbol $x_{t+1}$ by one for every node visited until the deepest node, say $x_t \cdots x_{t-j+1}$, is reached.
2) If the last updated count becomes at least 2, create a new node $x_t \cdots x_{t-j}$, and initialize its symbol counts to zero, except for the symbol $x_{t+1}$, whose count is set to 1. This completes the construction of $\mathcal{T}_{t+1}$.

The goal of this stage is to accumulate all the relevant contexts and the associated symbol statistics in a practical way as the length of the string grows. The tree will grow only in directions where repeated symbol occurrences take place, and the counts of all the symbols in all the contexts that have occurred are gathered, except a few early occurrences prior to the creation of the corresponding node. We could even gather these missed ones by backtracking and updating the counts, including those resulting from the padded initial zeros, to obtain $n_x(a|s)$ as defined after (10). Mainly for the simplicity in notation this is actually the case we analyze. The main results, however, are valid even for the easier to implement algorithm where the tree is grown by the given rules 1) and 2).

While letting the tree grow the algorithm also selects a certain distinguished context for each symbol $x_{t+1}$ from the growing source string $x^t = x_1 \cdots x_t$. These contexts together with the earlier ones are used to define the set of leaves $S_t$ of a complete tree $T_t$. There are several variants of the rule for the "optimal" context selection. The rules in [1] and [2] are based upon the application of the *MDL* principle or the calculation of the stochastic complexity. They compare, in effect, the difference between the empirical entropy of a father node and the sum of the entropies of all its sons, against their model cost difference, and makes the decision in favor of the winner. As illustrated in [6] with an example, this rule does not permit a consistent estimation of the data-generating source in all the cases of interest. As a remedy, a quite different context selection rule was proposed, one based upon the ideas of hypothesis testing. In this an upper bound $m$ on the model

order is assumed to be known. Then the empirical entropy calculated with each candidate model is compared with that obtained when its contexts are extended to the length $m$, and the shortest context for which the difference is below a threshold $\epsilon$ is chosen. A fixed threshold would not give consistent context estimates, which is why the threshold is required to shrink to zero at the same rate as the per-symbol model complexity in the theory of stochastic complexity. In the more general case where no maximum value for the model order is assumed, the numbers $m$ are increased with the length of the processed sequence. Although the resulting algorithm becomes prohibitively complex, since the estimation requires sums of an exponentially growing number of terms, the rule so modified was shown to provide a consistent estimation of any FSMX-generated source as well as to achieve the asymptotically optimal mean code length.

In [5] the original context selection rules in [1] and [2] were modified in a particularly efficient and easy to implement way. The algorithm was applied to a number of data compression problems with impressive results. For example, in randomly selected text files the per-symbol code length was typically 15–20% below the length obtained with a version of the Ziv–Lempel algorithm. The analysis of the algorithm with the given rule, however, turns out to be difficult, and in the following we study an easier-to-analyze modified version which still admits a reasonably simple implementation. To state the rule define

$$\hat{P}_x(a|s) = \frac{n_x(a|s)}{\sum_{b\in A} n_x(b|s)} \tag{12}$$

unless

$$\sum_{b\in A} n_x(b|s) = 0$$

in which case $\hat{P}_x(a|s) = 0$. We remind the reader that $n_x(a|s)$ denotes the number of times the string $\bar{s}a$ occurs as a substring in $0^{|s|}x$ less the number of times it occurs in $0^{|s|}$. For each node $sb$, $b \in A$, in a tree define

$$\Delta_x(sb) = \sum_{a\in A} n_x(a|sb) \log \frac{\hat{P}_x(a|sb)}{\hat{P}_x(a|s)}. \tag{13}$$

This is extended to the root node by $\Delta_x(\lambda) = \infty$. In words, $\Delta_x(sb)$ denotes the (ideal) code length difference when the symbol occurrences in context $sb$ have been encoded with the statistics gathered at the father node $s$ and with its own statistics. It is clearly nonnegative. This differs from the cross entropy used in [1], [2], and [6], written here not-per symbol occurrence but as follows:

$$\Delta\hat{H}_x(s) = n_x(s)\hat{H}_x(s) - \sum_{b\in A} n_x(sb)\hat{H}_x(sb)$$

where

$$\hat{H}_x(s) = -\sum_{a\in A} \hat{P}_x(a|s) \log \hat{P}_x(a|s).$$

The cross entropy is also nonnegative. By permitting the code length comparison between a node and each of its sons separately, as in (13), it is possible to make a finer differentiation between the nodes' performance, which results in a context selection rule that tends to yield a shorter code length.

We now complete the selection of the optimal context and of $T_t$. Let

$$S'_t = \{\text{the deepest node } w \text{ in } \mathcal{T}_t | \Delta_{x^t}(w) \geq C \log (t+1), \; |w| \leq g(t)\} \tag{14}$$

where $C$ is a constant, $g(t)$ is a strictly increasing function of $t$ to be specified later, and $|w|$ denotes the length of $w$. Define $T_t$ as the smallest complete supertree of $S'_t$. Note that while $S'_t$ is a subtree of $\mathcal{T}_t$, completion may cause some leaves of $T_t$ not to be in $\mathcal{T}_t$. The context selection rule is then defined by

$$s(x^t) = s_t \tag{15}$$

where $s_t$ denotes the longest path $x_t \cdots x_{t-k}$ in the intersection of $\mathcal{T}_t$ and $T_t$. The effect of the function $g(t)$ is two-fold. On the one hand, it restricts the search for the node where (14) holds, thus decreasing the probability of overestimation (selecting a node deeper than the optimal one), but on the other hand, it may increase the underestimation probability by not letting the number of nodes searched grow as fast as the data would dictate. The choice $g(t) = c \log t$, where $c = 1/\log d$, turns out to be a good tradeoff. Finally, the source assigns the probability

$$P(x^n) = \prod_{t=0}^{n-1} P_{x^t}(x_{t+1}|s_t) \tag{16}$$

to every string $x^n$, where the conditional probabilities are given by the rule [8]

$$P_{x^t}(a|s_t) = \frac{n_{x^t}(a|s_t) + 1/2}{\sum_{a\in A} n_{x^t}(a|s_t) + d/2}. \tag{17}$$

This rule is seen to be a slight modification of Laplace's rule of succession. Note that the probability (17) differs from the maximum likelihood estimate (12). The reason for using two different measures is that the code length difference (13) is computed in an easier-to-analyze nonpredictive way. If a predictive approach (using (17)) were used, the code length for context selection would include that portion of the model cost pertaining to the nodes in question, and we would not need the penalty term used in (14). This predictive rule is used in [5]. Despite the fact that the rule the algorithm employs for context selection compares code lengths calculated from the past strings in a two-pass nonpredictive way, the algorithm itself assigns probabilities to strings in a predictive way.

One may wonder why the set $S'_t$ itself does not qualify as a set of leaves, which would simplify the context selection rule to just finding the deepest node $w_t^*$ where

$$\Delta_{x^t}(w_t^*) \geq C \log (t+1)$$

and $|w_t^*| \leq g(t)$ hold, rather than our worrying about the additional nodes required to complete the tree. The reason is that some deepest node might be equivalent to an internal node of $T_t$, and if we cut the tree at such an internal node we might

lose other of its descendant leaves which are not equivalent and which must be retained for optimality. In fact, the simpler rule is used in practice, but even with the rule (14) and (15) it is not necessary to rebuild the whole tree $T_t$ for every $t$, which would turn the whole scheme impractical. Instead, for each $t$ we need to maintain the trees $T_t$ dynamically by marking its nodes within $\mathcal{T}_t$ with binary flags turned to 1. It suffices to examine only the nodes along the path to the leaf of $\mathcal{T}_t$ defined by $x_t x_{t-1} \cdots$ in order to find the deepest node $w_t^*$ according to (14), together with the nodes along the other paths starting at $w_t^*$ in $T_{t-1}$ in case $w_t^*$ falls within this tree. The way in which this is actually done does not affect the main results of this paper, and its detailed description is relegated to Appendix III.

## IV. Main Results

Algorithm Context with the context selection rule in Section III and (17) defines an information source by (16). This simply means that given any string as an input to the algorithm, it will deliver as the output the negative logarithm of the probability of the string in such a manner that the axioms for a random process (2) are satisfied. Hence, in particular, unless so desired, the algorithm does not explicitly give the estimates of the parameters of the optimally fitting tree machine. In fact, the universal process defined may be applied without our ever actually specifying the estimated tree machine. Our main results consist of a theorem stating that the so-constructed information source is universal in the class of finite memory sources, and that it is asymptotically optimal in the strong sense that it reaches the asymptotic stochastic complexity both in the mean sense and almost surely.

*Theorem 1:* Let $T$ be any minimal complete tree with $K$ leaves defining a finite-memory source $P_T(X^n)$ with the probability assignment (10), where $P(a|s) > 0$ for all $a \in A$ and $s \in S$. Then both

$$\frac{1}{n} \log \frac{P_T(X^n)}{P(X^n)} \leq \frac{K(d-1)}{2n} \log n + O(1/n) \tag{18}$$

with $P_T$-probability 1, and

$$\frac{1}{n} E_T \log \frac{P_T(X^n)}{P(X^n)} \leq \frac{K(d-1)}{2n} \log n + O(1/n) \tag{19}$$

where $P(X^n)$ is given by the universal source (16) for $C > 2(d+1)$ and the expectation $E_T$ is taken with respect to the distribution $P_T(X^n)$.

The proof of the theorem is given in Appendix II. It is based on a key lemma which states that the probability of the "error" event $E^t = \{x^t | T_t \neq T\}$ associated with the context selection rule not only tends to zero as $t \to \infty$ but it tends to zero fast enough. Specifically, we prove in Appendix I the following.

*Lemma 1:* Let $C > 2(d+1)$. Then

$$\sum_{t=1}^{\infty} P_T(E^t) \log t < \infty. \tag{20}$$

Actually, for (18) to hold it is sufficient that the sum is finite without the factors $\log t$, but the proof of such a result is no simpler than that of the stronger form. By the Borel–Cantelli Lemma, this implies strong consistency.

## V. An Application: Sequential Decision Problems

An immediate application of the universal source is a universal data compression system, for which we only need to add an arithmetic encoding/decoding unit to process each symbol $x_{t+1}$ in its context with the predictive distribution (17). Theorem 1 together with [2, Theorem 1] ensures that the resulting compression is asymptotically optimal to any desired accuracy if we choose sufficiently large registers to carry out the required computations.

The universal source can also be used for sequential prediction, and, more generally, to make universal sequential decisions on the future outcomes of the observed sequence. An appropriate and, in fact, asymptotically optimal decision rule results from Bayes' rule as applied to the universal probability. For example, in the binary prediction case the rule may be stated as

$$\hat{x}_{t+1} = \begin{cases} 0, & if P(x^t 0) > P(x^t 1) \\ 1, & if P(x^t 0) \leq P(x^t 1) \end{cases} \tag{21}$$

where $P(x^t)$ is the probability assigned to $x^t$ by the universal source. Such a scheme can be generalized to any decision problem with a specified loss function and a decision rule that results by minimization of the universal risk; i.e., the expected loss, where the expectation is taken with respect to the universal probability.

We give a brief analysis of the binary prediction problem to illuminate the special role played by the code length as a risk function in Theorem 1 above. Consider first the Bernoulli source, where the tree $T$ consists of the root with symbol probabilities $P_T(a|\lambda) = P_T(a)$. If $u$ denotes the symbol which has the larger of the two probabilities and we always predict the next symbol as $u$, the fractional number of errors per symbol is

$$\Pi_T(x^n) = \frac{1}{n} \sum_{t=0}^{n-1} \delta(x_{t+1}, u) \tag{22}$$

where $\delta(x, \hat{x}) = 0$ if $\hat{x} = x$, and 1, otherwise. The mean per-symbol prediction error is then $\Pi_T = 1 - P_T(u)$. If the symbol that has the larger probability is not known, we may use the predictor (21), written as $\hat{x}_{t+1} = 0$, if $n_{x^t}(0|\lambda) > t/2$ and 1, otherwise. Then, if $\hat{\Pi}(x^n)$ denotes the associated fractional number of errors made in the string $x^n$, obtained when $u$ in (22) is replaced by this predictor, the mean error is given by

$$E_T \hat{\Pi}(X^n) = \Pi_T + (1 - 2\Pi_T) P_T(D_n(u)) \tag{23}$$

where

$$D_n(u) = \begin{cases} \{x^n | \sum_{t=1}^{n} x_t > n/2\}, & \text{if } u = 0 \\ \{x^n | \sum_{t=1}^{n} x_t \leq n/2\}, & \text{if } u = 1. \end{cases} \tag{24}$$

The tail probabilities are summable, converging in a monotone increasing manner to the limit $1/(2(1-2\Pi_T)^2)$, evaluated in [9], which gives

$$E_T \hat{\Pi}(X^n) - \Pi_T \leq \frac{1}{2n(1-2\Pi_T)}. \tag{25}$$

This inequality without the expectation holds almost surely with respect to $P_T$.

For tree sources the total number of errors made in the string $x^n$ is the sum of the errors made in each context. Suppose first that we know the data-generating tree machine with all the conditional probabilities $P(a|s)$ at its leaves. Then by predicting each symbol as the one with the higher conditional probability in its context, we get with an extension of (22) the mean per symbol error as

$$\Pi_T = \sum_{s \in S} P_T(s) \Pi_{T,s} \tag{26}$$

where $\Pi_{T,s}$ denotes the smaller of the two conditional symbol probabilities in a context $s$ of stationary probability $P_T(s)$. Suppose next that we do not know the conditional probabilities in the contexts and instead use the majority rule predictor. An extension of (25) to this case can be complicated. However, to illustrate our methods it suffices to express the upper bound in the nonexplicit form of [9, Theorem 2]. If $\hat{\Pi}_T(x^n)$ denotes the fractional number of prediction errors made per symbol in the string $x^n$, we get the inequality

$$\begin{aligned} &E_T \hat{\Pi}_T(X^n) - \Pi_T \\ &\quad \le \sum_{s \in S} \frac{1}{2n} \sum_{t=0}^{\infty} P_T\{s_t = s, n_{x^t}(1|s) = n_{x^t}(0|s)\} \\ &\quad \equiv \sum_{s \in S} \frac{C_{T,s}}{n} \end{aligned} \tag{27}$$

where each $C_{T,s}$ is a positive constant that depends on the tree source and on $s$. One can further show [9], that in a suitable sense the right-hand side of (27) is also a lower bound on the expected fraction of extra errors over $\Pi_T$ made by any predictor. Unlike the lower bound for the code length [2], one cannot expect any lower bound for the prediction error to hold simultaneously for all predictors and essentially all sources. For example, in the class of independent binary sources the trivial predictor $\hat{x}_{t+1} = 0$ has the mean per symbol error which equals the ideal $\Pi_T$ for half of the sources, namely, whenever $P(0) \ge 1/2$. For the other half, however, the error will exceed the right-hand side of (25). See [9] for more details regarding the lower bound on prediction.

Our main result in this section is to show that even with an unknown model structure the predictions can be done by the rule (21) with $P(x^t)$ given by the probability of the universal source. The resulting fraction $\hat{\Pi}(x^n)$ of prediction errors satisfies

$$E_T \hat{\Pi}(X^n) - \Pi_T \le \sum_{s \in S} (C_{T,s} + K_{T,s})/n \tag{28}$$

where $K_{T,s}$ is a positive constant. This is because even if the maximum loss of unity is added whenever there is an error in selecting the correct context of the data-generating tree $T$ with the universal source, that is, whenever $x^t \in E^t$ in the notation of Lemma 1, by this lemma the mean additional loss is uniformly bounded. Hence, a term $K_{T,s}/n$ gets added to the per-symbol loss. Again by the Borel–Cantelli Lemma the error made cannot take place infinitely often with $P_T$-probability one, and hence (28), without the expectation, holds almost surely.

## Appendix I
## Proof of Lemma 1

The error event $E^t$ is composed of two events. One way a string $x^t$ can lead to a tree $T_t$, different from the data-generating tree $T$, is by overestimation; i.e., there is a leaf $s$ in the set of leaves $S$ of $T$ and a deeper node $su$ with length not exceeding $g(t)$ in which $\Delta_{x^t}(su) \ge C \log(t+1)$. Clearly, if

$$O^t_{su} = \{x^t | \Delta_{x^t}(su) \ge C \log(t+1)\}$$

then the set of the overestimation strings may be written as

$$O^t = \bigcup_{s \in S, |s| < |su| \le g(t)} O^t_{su}. \tag{A1}$$

The other way is by underestimation, which we discuss after having dealt in Lemma 2 with the overestimation case.

*Lemma 2:* Let $C > 2(d+1)$. Then

$$\sum_{t=1}^{\infty} P_T(O^t) \log t < \infty. \tag{A2}$$

*Proof:* Let $x^t$ belong to $O^t_{su}$ for some node $su$ such that $s \in S$ and $u \equiv wc$, where $c \in A$. Clearly, we can write $u$ that way for some string $\bar{w}$, possibly empty, because the length of $u$ is at least one. We then have by (10)

$$\log P_T(x^t|s_0) = R_s(x^t|S) + \sum_{a \in A} n_{x^t}(a|s) \log P(a|s) \tag{A3}$$

where we define

$$R_s(x^t|S) = \sum_{z \in S - \{s\}} \sum_{a \in A} n_{x^t}(a|z) \log P(a|z). \tag{A4}$$

Our plan is to replace the node $s$ by the set of leaves $\{sv\}$ of a smallest complete subtree of $T_t$ rooted at $s$ such that $sw$ is one of the leaves. Denote by $S_s$ the larger set of the leaves consisting of $S - \{s\} \cup \{sv\}$. This amounts to expressing $n_{x^t}(a|s)$ in terms of the sum

$$n_{x^t}(a|sw) + \sum_{v \ne w} n_{x^t}(a|sv)$$

by (11), and we have

$$\log P_T(x^t|s_0) = R_{sw}(x^t|S_s) + \sum_{a \in A} n_{x^t}(a|sw) \log P(a|sw) \tag{A5}$$

where, as in (A4),

$$R_{sw}(x^t|S_s) = \sum_{z \in S_s - \{w\}} \sum_{a \in A} n_{x^t}(a|z) \log P(a|z). \tag{A6}$$

If we replace in (A5) $P(a|sw)$ by $\hat{P}_{x^t}(a|sw)$ we get an upper bound for $\log P_T(x^t|s_0)$. And by further replacing $n_{x^t}(a|sw)$ by the sum

$$\sum_{b \in A} n_{x^t}(a|swb)$$

we get the inequality

$$\log P_T(x^t|s_0) \le R_{su}(x^t|S_s) + \sum_{a\in A}\sum_{b\in A} n_{x^t}(a|swb)\log \hat{P}_{x^t}(a|sw). \tag{A7}$$

Define next another process by the tree with the set of leaves $S_{su}$, obtained from $S_s$ by replacing the node $sw$ by its sons, the possibly new all-zero initial context $s_0'$, and the leaf distributions given by $Q_{su}(y^t|s_0'.x^t)$ defined as

$$\log Q_{su}(y^t|s_0'.x^t) = R_{sw}(y^t|S_s) + \sum_{a\in A}\sum_{b\neq c} n_{y^t}(a|swb)\log \hat{P}_{x^t}(a|sw) + \sum_{a\in A} n_{y^t}(a|su)\log \hat{P}_{x^t}(a|su).$$

Define the equivalence relation $y^t \equiv x^t$ if for every $a$ in $A$, $n_{y^t}(a|sw) = n_{x^t}(a|sw)$ and $n_{y^t}(a|su) = n_{x^t}(a|su)$. This partitions the set $O_{su}^t$ into, say, $\mu_{su}$ equivalence classes. Let $\sigma_{x^t}$ be the equivalence class containing $x^t$. Then for $y^t \in \sigma_{x^t}$, $\Delta_{x^t}(su) = \Delta_{y^t}(su)$, and with (13)

$$\log Q_{su}(y^t|s_0', x^t) = R_{sw}(y^t|S_s) + \sum_{a\in A} n_{y^t}(a|sw)\log \hat{P}_{x^t}(a|sw) + \Delta_{y^t}(su). \tag{A8}$$

With (A7) and (A8) this further implies

$$\log P_T(y^t|s_0) \le \log Q_{su}(y^t|s_0', x^t) - \Delta_{x^t}(su). \tag{A9}$$

Since $Q_{su}(\sigma_{x^t}|s_0'.x^t) \le 1$, we get the inequality

$$P_T(\sigma_{x^t}|s_0) \le 2^{-\Delta_{x^t}(su)} \le (t+1)^{-C}$$

and

$$P_T(O_{su}^t|s_0) \le \mu_{su}(t+1)^{-C}.$$

Since $0 \le n_{x^t}(a|sw) \le t$ and $0 \le n_{x^t}(a|su) \le t$ for $a$ ranging over $A$, there can be no more than $(t+1)^{2d}$ equivalence classes. Further, there are no more than $d^{g(t)}$ distinct sequences $su$, which for

$$g(t) = \frac{\log t}{\log d}$$

and $C > 2(d+1)$ implies

$$P_T(O^t|s_0) < K(t+1)^{-(1+\epsilon)}$$

for the positive number $\epsilon = C - 2(d+1)$. This completes the proof of Lemma 2.

We now turn to the underestimation case. Underestimation can take place in two ways. First, it may happen that the string $x^t$ is such that the set of searched nodes does not even include the entire tree $T$. This will happen if either the length of the deepest leaf in $T$ exceeds $g(t)$ or if the string $x^t$ is such that the set of the created nodes $\mathcal{T}_t$ does not include $T$. Further, whenever a string $\bar{w} = \cdots cba$ occurs in the string $x^t$ so does, of course, each of its shorter suffix strings, such as $cba$, $ba$, and $a$. The rule for growing the tree $\mathcal{T}_t$ is such that if

$$n_{x^t}(w) = \sum_a n_{x^t}(a|w)$$

is not smaller than $d|w|$, then surely all of the nodes $a$, $ab$, $abc$, including $w = abc\cdots$, will have been created as nodes of $\mathcal{T}_t$. Therefore, since $g(t)$ tends to infinity as $t$ grows, for sufficiently large $t$ the strings which cause this first type of underestimation belong to the set

$$U_1^t = \{x^t | n_{x^t}(s) < d|s|.\ \text{some } s \in T\}. \tag{A10}$$

The second type of underestimation is by (14) seen to occur for strings in the set

$$U_2^t = \{x^t | \Delta_{x^t}(zw) < C\log(t+1).\ \text{for some internal node } z \text{ of } T.\ \text{all } zw \in \mathcal{T}_t\}. \tag{A11}$$

The set of all strings that cause underestimation are then included in the union $U^t = U_1^t \cup U_2^t$. By Lemma 2, to complete the proof of Lemma 1 it suffices to show the following.

*Lemma 3:* Let $C > 2(d+1)$. Then

$$\sum_{t=1}^{\infty} P_T(U^t)\log t < \infty. \tag{A12}$$

*Proof:* Assuming that the state probabilities are bounded away from zero, a well-known result in the theory of large deviations (see, for example, [2, eq. (A4)]) states that for sufficiently large $t$, any $s \in T$, and any constant $c$

$$P_T\{n_{x^t}(s) < c\} < G\gamma^{-t}$$

for some constants $G > 0$ and $\gamma > 1$. It follows that

$$\sum_{t=1}^{\infty} P_T(U_1^t)\log t < \infty.$$

We consider next the event $U_2^t$. Let $T'$ denote the set of nodes $w$ in $T$ for which $wb$ is a leaf of $T$ for every $b \in A$. Clearly, any internal node of $T$ either belongs to $T'$ or has a descendant in $T'$. Therefore, (A11) takes the form

$$U_2^t = \{x^t | \Delta_{x^t}(wub) < C\log(t+1).\ \text{for some } w \in T'.\ \text{and all } wub \in \mathcal{T}_t\}.$$

Note that $\bar{u}$ may be the null string $\lambda$. Thus with

$$\Psi_{x^t}(wu) \equiv \sum_{b\in A} \Delta_{x^t}(wub)$$

we obtain

$$U_2^t \subseteq \{x^t | \Psi_{x^t}(wu) < dC\log(t+1)\ \text{for some } w \in T' \text{ and every } u\}.$$

To upper-bound the probability of $U_2^t$ we use techniques from the theory of large deviations as applied to Markov chains; in particular, a well-known lemma due to Csiszár, Cover, and Choi [10, Lemma 2(a)]. Since not every minimal complete tree $T$ admits an FSM implementation we must first define an FSMX source equivalent to the data-generating source. Let

$R$ denote the smallest complete supertree of $T$ that defines an FSMX source, and let $L$ denote the set of its leaves. For every leaf $z \in T$, assign the conditional probability $P(\cdot|z)$ to each extended node $zu$, where $zu \in L$. Taking the all-zero initial state $s'_0 \in L$, we obtain an FSMX source $P_R(X^n|s'_0)$ which is equivalent with $P_T(X^n|s_0)$. Consider the subtree of $R$ rooted at $w$ and defined by the descendants of $w \in T'$. Denote the set of leaves of this subtree by $L(w)$ and the set of its internal nodes, including the root $w$, by $R(w)$. Clearly, we can assume that $R(w)$ is a subtree of $\mathcal{T}_t$, for otherwise we could proceed as with $U_1^t$. With these definitions we have

$$U_2^t \subseteq \{x^t | \sum_{z \in R(w)} \Psi_{x^t}(z) < |R(w)| dC \log(t+1), \text{ for some } w \in T'\}.$$

Now, define

$$\Omega_{x^t}(w) \equiv \sum_{z \in R(w)} \Psi_{x^t}(z).$$

It can be readily verified that

$$\Omega_{x^t}(w) = \sum_{z \in L(w)} \sum_{a \in A} n_{x^t}(a|z) \log \frac{\hat{P}_{x^t}(a|z)}{\hat{P}_{x^t}(a|w)}. \tag{A13}$$

Clearly, it suffices to show that for every $w \in T'$ and some $\epsilon(w) > 0$, the set

$$U_{w,\epsilon}^t \equiv \{x^t \in A^t | t^{-1} \Omega_{x^t}(w) \le \epsilon(w)\}$$

satisfies

$$\sum_{t=1}^{\infty} P_R(U_{w,\epsilon}^t | s'_0) \log t < \infty. \tag{A14}$$

We prove (A14) using Csiszár, Cover, and Choi's lemma, which requires that the error event be given in terms of a set of probability distributions such that it includes a certain empirical distribution derived from $x^t \in U_{w,\epsilon}^t$. For each $s \in L$ let $s(a)$ denote the leaf in $L$ defined by the longest path $as$ in $R$. In other words, since $R$ admits an FSMX implementation we may view $s$ as a state of an FSM and $s(a)$ is the next state into which the symbol $a$ takes $s$ according to (5). Consider the two-dimensional empirical distribution defined over $L \times L$

$$\hat{P}_{x^t}(s,z) \equiv \begin{cases} t^{-1} n_{x^t}(a|s), & \text{if } z = s(a), \text{ some } a \in A \\ 0, & \text{otherwise.} \end{cases}$$

Note that

$$\sum_{a,z} n_{x^t}(a|z) = t.$$

In general, the marginals of this distribution are not equal. For a distribution $Q(\cdot,\cdot)$ over $L \times L$, let $\bar{Q}(\cdot)$ denote its left marginal. Define

$$Q(s|z) \equiv \frac{Q(z,s)}{\bar{Q}(z)}, \quad s, z \in L, \ \bar{Q}(z) \neq 0.$$

Further let

$$Q(s|w) \equiv \frac{\sum_{z \in L(w)} Q(z,s)}{\sum_{z \in L(w)} \bar{Q}(z)}, \quad s \in L, \ w \in T', \ \sum_{z \in L(w)} \bar{Q}(z) \neq 0.$$

Finally, let $\Gamma$ denote the set of distributions over $L \times L$ defined by $Q(\cdot,\cdot) \in \Gamma$ iff

$$\epsilon(Q) \equiv \sum_{z \in L(w)} \sum_{s \in L} Q(z,s) \log \frac{Q(s|z)}{Q(s|w)} \le \epsilon(w).$$

Note that $\epsilon(Q) \ge 0$ for every distribution $Q$. By (A13) and the definition of $U_{w,\epsilon}^t$ it follows that $x^t \in U_{w,\epsilon}^t$ if and only if $\hat{P}_{x^t}(\cdot,\cdot) \in \Gamma$ or, equivalently

$$P_R(U_{w,\epsilon}^t | s'_0) = P_R\{\hat{P}_{x^t}(\cdot,\cdot) \in \Gamma | s'_0\}.$$

Denote by $\Gamma_0$ the set of distributions belonging to the closure of $\Gamma$ (relative to the set of all distributions over $L \times L$) and for which the two marginals are identical. By the above mentioned large deviations lemma we then obtain

$$\limsup_{t \to \infty} \frac{1}{t} \log P_R\{\hat{P}_{x^t}(\cdot,\cdot) \in \Gamma | s'_0\} \le -D$$

where

$$D \equiv \min_{Q \in \Gamma_0} D(Q \| P_R)$$

and

$$D(Q\|P_R) \equiv \sum_{s,z \in L} Q(s,z) \log \frac{Q(s,z)}{\bar{Q}(s) P(z|s)}.$$

(Here $0 \log 0 = 0 \log \frac{0}{0} = 0$ and $\log \frac{h}{0} = \infty$, if $h > 0$). Clearly, $P(z|s)$ is unambiguously determined by $P(a|s)$ for $a \in A$. Note that, by the definition of $\Gamma$, $D$ is independent of $t$, and in order to prove (A14) it suffices to show that there exists $\delta > 0$ such that if $\epsilon(w) < \delta$ then $D \neq 0$. Now, by the irreducibility of the Markov chain defining the source $P_R(X^n)$, it can be readily seen that the unique distribution $Q^0(\cdot,\cdot)$ over $L \times L$ with two identical marginals for which $D(Q^0\|P_R) = 0$ is

$$Q^0(s,z) = P_R^0(s) \times P(z|s), \quad s, z \in L$$

where $P_R^0(\cdot)$ denotes the (unique) stationary distribution defined by $P(\cdot|\cdot)$. Now, define $\delta \equiv \epsilon(Q^0)$. Clearly, if $\delta = 0$ then the distribution $Q^0(\cdot|z) = P(\cdot|z)$ must be the same for every $z \in L(w)$. Hence $\delta > 0$, for otherwise the set of leaves $wb$ of $T$, $b \in A$, could be replaced by $w$ to obtain an equivalent source, thus contradicting the minimality of $T$. Therefore, with $\epsilon(w) < \delta$, it follows that $Q^0(\cdot,\cdot)$ is not in $\Gamma_0$. Consequently, $D \neq 0$, and the proof is complete.

## Appendix II
## Proof of Theorem 1

We prove the theorem for the probability assignment (10) with the initial all-zero context $s_0$. By Lemma 1 and the Borel–Cantelli Lemma, the set $C$ of infinite $d$-ary sequences for which there exists an integer $N(x)$ such that for every $t > N(x)$, $T_t = T$, has $P_T$-probability 1. Let $\hat{H}_{x^t}(T)$ denote the empirical conditional entropy of a sequence $x^t$ with respect to $T$, namely

$$\hat{H}_{x^t}(T) \equiv t^{-1} \sum_{s \in T} n_{x^t}(s) \hat{H}_{x^t}(s).$$

Had the universal probability (16) and (17) been computed using the true (unknown) tree $T$ instead of $T_t$, we would have obtained for *every* $x^t$, [8]

$$-t^{-1}\log P(x^t) \le \hat{H}_{x^t}(T) + \frac{K(d-1)}{2t}\log t + O(t^{-1}).$$

Therefore, for every $x \in C$ and every $t > N(x)$ we have

$$-t^{-1}\log P(x^t) \le \frac{N(x)}{t}\log(2N(x)+d) + \hat{H}_{x^t}(T) + \frac{K(d-1)}{2t}\log t + O(t^{-1}). \quad \text{(A15)}$$

Now, define

$$B \equiv \{x \in A^\infty \mid \limsup_{t\to\infty}[-\log P(x^t) - t\hat{H}_{x^t}(T) - \frac{K(d-1)}{2}\log t] < \infty\}. \quad \text{(A16)}$$

By (A15) and (A16), $C$ is a subset of $B$ and, consequently, $P_T(B) = 1$. Furthermore, by the asymptotic equipartition property

$$t^{-1}\log P_T(x^t) + \hat{H}_{x^t}(T) \to 0$$

with $P_T$-probability 1, which completes the proof of (18).

The inequality (19) follows from Lemma 1 by the arguments in [6, Theorem 4(a)].

## APPENDIX III

In this appendix we describe an efficient updating of the trees $T_t$. Actually, by examining the nodes stated at the end of Section III only, we do not maintain the trees $T_t$, but rather certain slightly bigger trees, say $\bar{T}_t$. The nodes of $\bar{T}_t$ and $T_t$ along the current path $x_t x_{t-1}\cdots$ are identical, and thus the difference between them does not affect the context selection. At the nodes $w$ along the other paths no updating of $\mathcal{T}_t$ takes place so $\Delta_{x^t}(w) = \Delta_{x^{t-1}}(w)$, but it may happen that a deepest node $w^*$ where $\Delta_{x^{t-1}}(w^*) \ge C\log t$ was true ceases to satisfy this relation when $t-1$ is replaced by $t$. Hence, such a node may no longer belong to $T_t$, and if we leave it in $\bar{T}_t$ (to avoid visiting too many nodes) we get $\bar{T}_t \supseteq T_t$. On the other hand, at the nodes outside of $\bar{T}_{t-1}$, $\Delta_{x^{t-1}}(w) < C\log t$ is true, and it remains to hold when $t-1$ is replaced by $t$ unless $\Delta_{x^{t-1}}(w)$ gets updated, which can take place only at the nodes along the current path and their offsprings.

In order to update $\bar{T}_{t-1}$ we proceed as follows. Having found $w_t^*$, two cases can occur: Either $w_t^*$ is in the tree $\bar{T}_{t-1}$, or it is deeper than the corresponding leaf, say $z_{t-1}$. In the latter case, $s_t = w_t^*$, and the algorithm extends $\bar{T}_{t-1}$ to $\bar{T}_t$ by flipping the flag to 1 at each node along the path between the leaf $z_{t-1}$ of $\bar{T}_{t-1}$ and $w_t^*$ as well as their son nodes, except those of $w_t^*$, which is a leaf. The former case, where $w_t^*$ falls within $\bar{T}_{t-1}$ is a bit more involved. If $w_t^*$ is a leaf, clearly $s_t = w_t^*$. In the remaining case, where $w_t^*$ is an internal node, the question is whether or not the tree $\bar{T}_{t-1}$ can be pruned at this node and hence making it a leaf. To find out we must check if any path in $\bar{T}_{t-1}$, starting at $w_t^*$, has a node where

$$\Delta_{x^t}(w) = \Delta_{x^{t-1}}(w) \ge C\log t$$

holds and whose length does not exceed $g(t)$, of course. If such paths exist, we must retain them in $\bar{T}_t$, and we set $s_t$ as the son node of $w_t^*$, defined by the current path. In addition, we prune the tree $\bar{T}_{t-1}$ at all the son nodes of $w_t^*$ which were not retained by the path check just made.

## ACKNOWLEDGMENT

The authors gratefully acknowledge P. Algoet for his suggestions on the prediction results.

## REFERENCES

[1] J. Rissanen, "A universal data compression system," *IEEE Trans. Inform. Theory*, vol. IT-29, no. 5, pp. 656–664, 1983.
[2] ——, "Complexity of strings in the class of Markov sources," *IEEE Trans. Inform. Theory*, vol. IT-32, no. 4, pp. 526–532, 1986.
[3] ——, "Universal coding, information, prediction, and estimation," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 4, pp. 629-636, 1984.
[4] ——, "Noise separation and MDL modeling of chaotic processes," in *Proc. Workshop "From Statistical Physics to Statistical Inference and Back"* (Cargese, Corsica, Aug. 31–Sept. 12, 1992).
[5] G. Furlan, "Contribution a l'étude et au développement d'algorithmes de traitement du signal en compression de données et d'images," Ph.D. dissertation, l'Université de Nice, Sophia Antipolis, France (in French), 1990.
[6] M.J. Weinberger, A. Lempel, and J. Ziv, "A sequential algorithm for the universal coding of finite-memory sources," *IEEE Trans. Inform. Theory*, vol. IT-38, no. 3, pp. 1002–1014, 1992.
[7] M.J. Weinberger, N. Merhav, and M. Feder, "Optimal sequential probability assignment for individual sequences," *IEEE Trans. Inform. Theory*, vol. IT-40, no. 2, pp. 384–396, 1994.
[8] R.E. Krichevskii and V.K. Trofimov, "The performance of universal encoding," *IEEE Trans. Inform. Theory*, vol. IT-27, no. 2, pp. 199–207, 1981.
[9] N. Merhav, M. Feder, and M. Gutman, "Some properties of sequential predictors for binary Markov sources," *IEEE Trans. Inform. Theory*, vol. 39, no. 3, pp. 887–892, 1993.
[10] I. Csiszár, T.M. Cover, and B. Choi, "Conditional limit theorems under Markov conditioning," *IEEE Trans. Inform. Theory*, vol. IT-33, no. 6, pp. 788–801, 1987.