Computation of universal objects for distributions over co-trees

Henrik Densing Petersen University of Copenhagen
Department of Mathematical Sciences
Universitetsparken 5,
2100 Copenhagen, Denmark
Email: m03hdp@math.ku.dk
Flemming Topsøe University of Copenhagen
Department of Mathematical Sciences
Universitetsparken 5,
2100 Copenhagen, Denmark
Email: topsoe@math.ku.dk

***Abstract*—For an arbitrary ordered set, one may consider the model of all distributions P for which an element which precedes another element is considered the more significant one in the sense that the implication $a \leq b \Rightarrow P(a) \geq P(b)$ holds. It will be shown that if the ordered set is a finite co-tree, then the universal predictor for the indicated model or, equivalently, the corresponding universal code, can be determined exactly via an algorithm of low complexity.**

Index terms – **Universal code, universal predictor, transitivity identity.**

Contents

I. Introduction, background and motivation

Throughout the paper we study finite co-trees, i.e. finite ordered sets Λ for which every non-maximal element a has a unique immediate successor, denoted a^+ (thus a^+ is the only element $b > a$ for which no element c satisfies $a < c < b$). To prevent misunderstanding, the order structure means that the conditions of transitivity, reflexivity and anti-symmetry ($x \leq y \wedge y \leq x \Rightarrow x = y$) hold.

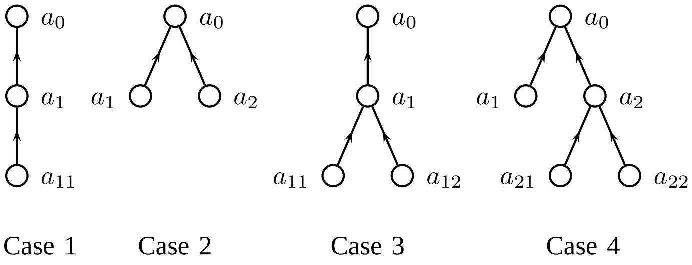


Figure 1. Some simple suspended co-trees

We often think of a co-tree as an *oriented graph* and refer to the elements as *nodes*. If an arrow points from a to b, then $a < b$. Among special nodes we find the *minimal* and the *maximal* nodes. If there is only one maximal node, we say that the co-tree is *suspended* with the maximal node as *top node*. Some examples of suspended co-trees are depicted in Figure 1. Case 1 is a *linear order* (characterized by the fact that also the reverse order is a co-tree), Case 2 is the simplest non-linear co-tree and cases 3 and 4 are only slightly more complicated. We shall return to the examples later in this section. Note that we have named the nodes in a systematic manner corresponding to their *levels*. The top node is in level 0 and is denoted a_0 (really thought of as $a_\emptyset$). Nodes in level k are indexed $a_{\varepsilon_1,\cdots,\varepsilon_k}$ with a string $\varepsilon_1,\cdots,\varepsilon_k$ of natural numbers of length k as index. They are defined recursively such that $a_{\varepsilon_1,\cdots,\varepsilon_{k-1}}$ is the immediate successor of all nodes of the form $a_{\varepsilon_1,\cdots,\varepsilon_{k-1},\varepsilon_k}$. Since we count levels from the top, the minimal nodes may well be in different levels (as in Case 4).

The indicated indexing of the nodes in a finite suspended co-tree points to a unique representation up to isomorphism. Indeed, Λ can be identified with a finite subset of the set $\mathcal{N}$ of finite strings (including the empty string) of natural numbers such that $\emptyset \in \Lambda$ and such that Λ is closed under *restriction* ($a_{\varepsilon_1,\cdots,\varepsilon_k} \in \Lambda$, $\nu \leq k \Rightarrow a_{\varepsilon_1,\cdots,\varepsilon_\nu} \in \Lambda$) and under "last element diminishing" ($a_{\varepsilon_1,\cdots,\varepsilon_k,n} \in \Lambda \Rightarrow a_{\varepsilon_1,\cdots,\varepsilon_k,m} \in \Lambda$ for

F. Topsøe has been supported by the Danish Natural Science Research Council and by INTAS, project 00-738

$m \leq n$). The *height* of Λ can be defined as the length of the longest string in the set $\mathcal{N}$ which identifies Λ. We shall later refer a few times to the representation described as the *standard representation* of Λ.

Cases 1-3 from Figure 1 are co-trees with *uniform branching*. In general, these co-trees are defined by a sequence $(k_1, \cdots, k_n)$ of natural numbers. Then $\Lambda[k_1, \cdots, k_n]$ denotes the suspended co-tree with k_ν immediate predecessors of each node in level $\nu - 1$ for $1 \leq \nu \leq n$. The sequence $(k_1, \cdots, k_n)$ is the *branching pattern* of the co-tree. Thus the co-trees in Cases 1-3 from Figure 1 are $\Lambda[1,1]$, $\Lambda[2]$ and $\Lambda[1,2]$, whereas the last co-tree is not of this type. Co-trees with uniform branching are identified with a product set of finite strings under the standard representation.

Among the subsets of a co-tree Λ, the *left sections* play a special role. Notation and definition is given by

$$a^{\downarrow} = \{b \in \Lambda | b \leq a\}\,.$$

We use the general notation $|\cdot|$ for "the number of elements in $\cdot$", and put $N(a) = |a^{\downarrow}|$. We also need the non-negative real numbers $\overline{N}(a)$ defined for $a \in \Lambda$ by

$$\overline{N}(a) = N(a) \ln N(a)\,. \tag{1}$$

A subset $B \subseteq \Lambda$ is *hereditary* if the implication $a \in B \Rightarrow a^{\downarrow} \subseteq B$ holds or, equivalently, if B is a union of left-sections.

By $M^1_+(\Lambda)$ we denote the set of all distributions (always understood to be probability distributions) over Λ. The *order model* $\mathcal{P} = \mathcal{P}(\Lambda)$, which is the object we will study, is defined as the model of all distributions P for which $P(a) \geq P(b)$ whenever $a \leq b$ [1]. We shall assign "representative" objects to this model, either in the form of the *universal predictor* or in the form of the *universal code*, objects which will be defined carefully below. Before we do so, we comment on the basic structure of $\mathcal{P}(\Lambda)$.

The uniform distributions over the left sections are of special significance. Indeed, denoting by U_a the uniform distribution over $a^{\downarrow}$, we find that all these distribution and then also all mixtures of them are members of the model $\mathcal{P}(\Lambda)$. Conversely, any distribution $P \in \mathcal{P}(\Lambda)$ can be written in a unique way as a convex mixture of uniform distributions over left sections:

$$P = \sum_{a \in \Lambda} w_a U_a\,. \tag{2}$$

In other words, the following structural result holds:

Proposition 1.1: The order model $\mathcal{P}(\Lambda)$ is a simplex with the distributions $(U_a)_{a \in \Lambda}$ as extremal distributions.

We take this result as background information. The interested reader will not find it hard to provide a proof. The decomposition (2) is the *barycentric decomposition of* P and, with reference to this decomposition, the *spectrum* of P is defined by

$$\sigma(P) = \{a | w_a > 0\}\,. \tag{3}$$

One of the objects we shall search for is related to *coding*. In this paper, a *code* over the *alphabet* Λ is identified with a *code length function* $\kappa : \Lambda \to [0, \infty]$, required to satisfy *Kraft's equality*

$$\sum_{a \in \Lambda} e^{-\kappa(a)} = 1\,.$$

Note that we have chosen to work with theoretical (natural) units, hence use exponentiation with respect to the natural base. The set of all codes over Λ is denoted $\mathrm{K}(\Lambda)$.

For $P \in M^1_+(\Lambda)$ and $\kappa \in \mathrm{K}(\Lambda)$ we denote by $\langle \kappa, P \rangle$ the *average code length*,

$$\langle \kappa, P \rangle = \sum_{a \in \Lambda} \kappa(a) P(a)\,.$$

The overall goal is to choose a code so as to minimize this quantity. If P is fixed, the minimum is attained for the code *adapted to* P, given by

$$\kappa(a) = \ln \frac{1}{P(a)} \text{ for } a \in \Lambda\,,$$

and the minimum value is the *entropy* of P,

$$\mathrm{H}(P) = \sum_{a \in \Lambda} P(a) \ln \frac{1}{P(a)}\,.$$

When κ is adapted to P, we also express this by saying that P is the distribution which *matches* κ.

Let $P \in M^1_+(\Lambda)$ and $\kappa^* \in \mathrm{K}(\Lambda)$. The *redundancy* associated with P and κ^*, or, in more suggestive terms, the *redundancy of* κ^* *with* P *as the "true" distribution* is denoted $\mathrm{D}(P\|\kappa^*)$ and defined as the difference between the actual average code length and the minimal achievable value, i.e.

$$\mathrm{D}(P\|\kappa^*) = \langle \kappa^*, P \rangle - \mathrm{H}(P)\,. \tag{4}$$

This quantity is nothing but the well known *Kullback-Leibler divergence* between P and the distribution P^* matching κ^*, in standard notation:

$$\mathrm{D}(P\|P^*) = \sum_{a \in \Lambda} P(a) \ln \frac{P(a)}{P^*(a)}\,.$$

The basic identity (4) is mostly written in the form

$$\langle \kappa^*, P \rangle = \mathrm{D}(P\|\kappa^*) + \mathrm{H}(P)\,, \tag{5}$$

and referred to as the *linking identity*. We need another basic identity, the *compensation identity* of [9], which makes more precise the fact that $(P,Q) \curvearrowright \mathrm{D}(P\|Q)$ is convex in P:

$$\sum_{\nu=1}^{k} \alpha_\nu \,\mathrm{D}(P_\nu\|Q) = \sum_{\nu=1}^{k} \alpha_\nu \,\mathrm{D}(P_\nu\|P^*) + \mathrm{D}(P^*\|Q)\,, \tag{6}$$

valid for any convex combination $P^* = \sum_{\nu=1}^{k} \alpha_\nu P_\nu$ and any distribution Q. In terms of codes, the identity reads

$$\sum_{\nu=1}^{k} \alpha_\nu \,\mathrm{D}(P_\nu\|\kappa) = \sum_{\nu=1}^{k} \alpha_\nu \,\mathrm{D}(P_\nu\|\kappa^*) + \mathrm{D}(P^*\|\kappa)\,, \tag{7}$$

[1] This model – and not the alternative choice of all order-preserving distributions – is considered to be the natural one, a main reason being that if a precedes b ($a \leq b$) this is taken as a sign that a is more "significant" than b, hence, for sensible distributions, one should have $P(a) \geq P(b)$ rather than the other way round. In terms of coding (see below) our choice appears even more natural as it reflects the good sense of associating the shorter code words to the more significant events.

with κ just any code and with κ^* the code adapted to $P^* = \sum_{\nu=1}^{k} \alpha_\nu P_\nu$. The compensation identity follows from the linking identity, cf. [9].

Returning to the order model $\mathcal{P} = \mathcal{P}(\Lambda)$, we define two quantities of *redundancy*, firstly the *guaranteed redundancy* of any code $\kappa^* \in \mathrm{K}(\Lambda)$ which is defined by

$$\mathrm{R}(\kappa^*) = \sup_{P \in \mathcal{P}} \mathrm{D}(P \| \kappa^*)$$

and then the *minimax redundancy* defined by

$$\mathrm{R}_{\min} = \inf_{\kappa^* \in \mathrm{K}(\Lambda)} \mathrm{R}(\kappa^*) \,.$$

Considering the constant code, we realize that $\mathrm{R}_{\min}$ is finite.

It may be seen directly, and also follows from results later on, that there exists a unique code κ^*, *the universal code*, such that $\mathrm{R}(\kappa^*) = \mathrm{R}_{\min}$. The distribution which matches the universal code is the *universal predictor*. It is considered the most unbiased representation of the model $\mathcal{P}$. The two universal objects identified, are those we shall aim at characterizing by an algorithm of low complexity. In order to achieve this goal we shall use a special instance of a result from general optimization theory, which is much used in information theory and there often ascribed to Kuhn and Tucker, cf. [1]. We formulate the result in a way adapted to our needs:

Proposition 1.2 (Kuhn-Tucker criterion): Consider the order model $\mathcal{P} = \mathcal{P}(\Lambda)$ associated with a finite co-tree Λ. Let $P^* \in \mathcal{P}$ and let κ^* be the code adapted to P^*. Assume that, for some constant R, the following two conditions hold:

$$\mathrm{D}(U_a \| \kappa^*) = R \text{ for } a \in \sigma(P^*) \,, \tag{8}$$

$$\mathrm{D}(U_a \| \kappa^*) \le R \text{ for all } a \in \Lambda \,. \tag{9}$$

Then P^* is the universal predictor, κ^* the universal code and $\mathrm{R}_{\min} = R$.

Though essentially known, we provide a simple intrinsic proof:

Proof: By convexity of redundancy in the first variable – a consequence of (7) – and as the U_a's are the extremal distributions of $\mathcal{P}$, it follows from (9) that $\mathrm{R}(\kappa^*) \le R$. On the other hand, for every $\kappa \in \mathrm{K}(\Lambda)$, we find, using (8) and applying (7) with the w_a's the barycentric coordinates of P^*, that

$$\begin{aligned} \mathrm{R}(\kappa) &= \sum_{a \in \sigma(P^*)} w_a R(\kappa) \ge \sum_{a \in \sigma(P^*)} w_a \, \mathrm{D}(U_a \| \kappa) \\ &= \sum_{a \in \sigma(P^*)} w_a \, \mathrm{D}(U_a \| \kappa^*) + \mathrm{D}(P^* \| \kappa) \\ &= R + \mathrm{D}(P^* \| \kappa) \,. \end{aligned}$$

Thus, for every κ, the in itself interesting inequality $\mathrm{R}(\kappa) \ge R + \mathrm{D}(P^* \| \kappa)$ holds. As $\mathrm{D}(P^* \| \kappa) \ge 0$ with equality if and only if $\kappa = \kappa^*$, the stated result follows. ■

Through intrinsic reasoning, it is also possible to show that one can indeed find P^* and κ^* with properties as in Proposition 1.2, cf. [8]. It is comforting to know this, however, we do not need that result as we shall find P^* and κ^* directly which satisfy the conditions of the proposition. We do remark though that by exploiting the sufficiency as well as the necessity of the conditions of Proposition 1.2 it is easy to reduce the search for the universal objects associated with a general finite co-tree to the similar search for suspended co-trees [2].

	Case 1	Case 2	Case 3	Case 4
$\sigma(\Lambda)$	Λ	Λ	$\Lambda \setminus \{a_1\}$	Λ
$P^*(a_0)$	$\frac{4}{27} \cdot \frac{1}{Z}$	$\frac{1}{27} \cdot \frac{1}{Z}$	$\frac{1}{16} \cdot \frac{1}{Z}$	$\frac{27}{3125} \cdot \frac{1}{Z}$
$P^*(a.)$	$\frac{1}{4} \cdot \frac{1}{Z}$	$2 \times \frac{1}{Z}$	$\frac{1}{16} \cdot \frac{1}{Z}$	$\frac{1}{Z}\,,\ \frac{1}{27} \cdot \frac{1}{Z}$
$P^*(a..)$	$\frac{1}{Z}$		$2 \times \frac{1}{Z}$	$2 \times \frac{1}{Z}$
$\mathrm{R}_{\min} = \ln Z$	$\ln \frac{151}{108}$	$\ln \frac{55}{27}$	$\ln \frac{17}{8}$	$\ln \frac{256979}{84375}$

Table I
UNIVERSAL PREDICTORS FOR THE CO-TREES IN FIGURE 1

Objects P^* and κ^* will from now on denote the universal predictor and the universal code of a co-tree Λ under discussion. The *spectrum* of Λ is the spectrum of P^*: $\sigma(\Lambda) = \sigma(P^*)$. Nodes in $\sigma(\Lambda)$ are referred to as *active nodes* of Λ. The co-tree Λ has *full spectrum* if all nodes are active. Otherwise, the spectrum is *deficient*. An *anchor* is a distribution U_a with $a \in \sigma(\Lambda)$.

It turns out that the difficulty in determining P^* and κ^* lies in determining which nodes are active, i.e. in determining the spectrum. Offhand, there is little we can say:

Proposition 1.3: Every maximal node of a co-tree is active.

Proof: If a maximal node a is not active, $P^*(a) = 0$ must hold by Proposition 1.1. Let b be a node different from a (we may assume that Λ is not a singleton). Then $\mathrm{D}(U_b \| P^*) = \infty$, hence $\mathrm{R}(\kappa^*) = \infty$ which is absurd. ■

Regarding the structure of $\sigma(\Lambda)$, a first guess may be that Λ has full spectrum, but even for very simple co-trees this is not so. One example is provided by Case 3 of Figure 1. This fact is revealed by Table I which shows the nature of the universal predictor and the associated spectrum for all co-trees depicted in Figure 1. The correctness of the table is easily checked by appeal to Proposition 1.2. In all cases, a normalizing factor, Z, appears and the minimax redundancy is $\ln Z$, a fact that follows from Proposition 1.2 since all weights of minimal nodes when discarding the normalizing factor is 1 and since all minimal nodes are in the spectrum of the co-tree. That this behaviour holds generally will be demonstrated later in Proposition 3.2.

The natural interpretations related to codes as well as the significance of the problem outlined as one of *general univer-*

[2] Indeed, if Λ is the direct sum of (suspended) co-trees Λ_ν; $\nu = 1, \cdots, m$ with associated minimax redundancies R_ν and universal predictors P^*_ν then

$$P^* = \sum_{\nu=1}^{m} \frac{e^{\mathrm{R}_\nu}}{e^{\mathrm{R}_1} + \cdots + e^{\mathrm{R}_m}} P^*_\nu$$

is the universal predictor for Λ and $\ln \sum_\nu e^{\mathrm{R}_\nu}$ the associated minimax redundancy.

sal prediction and coding (general, because many other models than models related to order structure may be considered) is recognized in the information theoretical literature since long. Early works in this area include Fitingof [3] and Davisson [2]. The reader may also consult the survey article [4] by Feder and Merhav. Regarding the interesting connection which exists between minimax redundancy and maximal transmission rate, *capacity*, i.e. the *redundancy-capacity theorem* of Gallager and Ryabko, see [6]. In our situation, the result involves the discrete memoryless channel with Λ as input- as well as output alphabet and with the distributions $(U_a)_{a\in\Lambda}$ as the conditional output distributions, given an input letter (here a node in Λ). By the redundancy-capacity theorem, the optimal distribution on the input side is given by the barycentric coordinates of the universal predictor P^* and the optimal distribution on the output side is P^* itself. We shall not exploit this connection in the sequel. Rather, the situation is that the results which we shall develop can be used to show how to determine the optimal distributions and the capacity of the special discrete memoryless channels that can arise in the way described.

The motivation behind our very special study related only to order models in co-trees is many sided. Firstly, to the best of our knowledge, this class is the most comprehensive class for which an exact determination of universal objects can be provided either directly or via a reasonable algorithm. For the subclass of order models based on linearly ordered sets, a complete result already exists. It is due to Ryabko who developed a closed formula for the universal predictor, cf. [5]. For the larger subclass of co-trees with uniform branching, an algorithm was announced in Topsøe [10] but the details were never published.

Secondly, our main results, Theorems 5.1, 6.1 and 7.1, may also be considered as useful reservoirs of examples which may serve as test cases for future research. However, we remark that it appears very difficult, even theoretically impossible, to develop exact results expressed in terms of standard functions for other desirable models than those here considered, either based on order structures other than co-trees (e.g. trees) or on other constructs (such as Bernoulli models). Thus, the idea to look into models based on sequences rather than individual observations from an order model $\mathcal{P}$, is bound to fail. Severe obstacles to such a program exists as will be revealed by a reference to Galois theory (details will be provided in research by Harremoës and Topsøe, in preparation).

As a final motivation we note, as pointed out to us by Boris Ryabko, cf. also [7], that for certain applications to biology, information about biological species is sometimes available only in inconclusive form resulting – not in the direct determination of their relative numbers – but only in an ordering among the species, from the more frequent to the less frequent ones. Modelling as done here based on a co-tree is one possibility, though modelling based on trees rather than co-trees appear just as interesting, or perhaps even more interesting. However, models with trees in place of co-trees are without reach if you insist on expressing the universal objects in closed form.

II. Co-trees with full spectrum

As indicated in the introduction, the difficulty in determining the universal objects lies in determining the spectrum. In this section we study co-trees with full spectrum and demonstrate how the universal objects may be determined for these trees. It is convenient first to introduce some concepts: For a node $a \in \Lambda$, we denote by a^- the set of *immediate predecessors* of a, i.e. the set of all $b < a$ for which no node c satisfies $b < c < a$. Thus $a^- = \emptyset$ if a is a minimal node. Further, to each node we associate a certain *weight* defined by

$$W(a) = \frac{\prod_{b\in a^-} N(b)^{N(b)}}{N(a)^{N(a)}}\,, \tag{10}$$

and by Z we denote the *normalizing factor*

$$Z = \sum_{a\in\Lambda} W(a)\,. \tag{11}$$

Theorem 2.1: A co-tree Λ has full spectrum if and only if, for every pair of nodes (b, a) with $b \in a^-$, the inequality $W(b) \geq W(a)$ holds. And when this condition is satisfied, the universal predictor is given by normalization of W, i.e. $P^*(a) = W(a)/Z$ for any $a \in \Lambda$. Furthermore, $\mathrm{R}_{\min} = \ln Z$.

Proof: Assume that Λ has full spectrum. A straightforward analysis, which we shall leave to the reader, shows that the Kuhn-Tucker conditions of Proposition 1.2 can only be fulfilled with P^* given via W as described. As $P^* \in \mathcal{P}(\Lambda)$, the stated inequalities must hold. That, conversely, P^* is as stated when the inequalities hold amounts to simple checking based on Proposition 1.2. The formula $\mathrm{R}_{\min} = \ln Z$ follows e.g. by noting that $\mathrm{D}(U_a\|P^*) = \ln Z$ when a is a minimal node. ■

The probability distribution, call it W^*, obtained by normalization of W may be considered for any co-tree. One will see that the sought universal predictor, P^*, is the *information projection* of W^* on $\mathcal{P}(\Lambda)$. This fact – or the connection to a problem related to capacity indicated towards the end of Section I – may be exploited to calculate P^* via standard algorithms, cf. Chapter 13 of [1]. However, we stress that this will only lead to approximate determinations of P^*. As we are here concerned with precise determinations of P^*, either via a direct formula (possible only in special cases) or via an algorithm which stops with the exact result after finitely many steps, we shall not pursue this possibility. Another matter is that standard algorithms may be useful anyhow in order to guess what the spectrum is, and then exact formulas are easy to derive, cf. the discussion related to (19) in Section IV.

If we conceive Theorem 2.1 as an algorithm to check whether the co-tree in question has full spectrum or not, we note that the algorithm is very efficient as the number of inequalities which need to be checked is at most the number of nodes in the co-tree.

Let us have a closer look at Theorem 2.1 in the case of a co-tree $\Lambda[k_1, \cdots, k_n]$ with uniform branching. For such a co-tree, we denote by N_ν the common number of $N(a)$ for nodes in level ν ($\nu = 0, \cdots, n$). The N_ν's may be calculated recursively as follows:

$$N_n = 1, \quad N_\nu = 1 + k_{\nu+1} N_{\nu+1} \text{ for } \nu = n-1, \cdots, 0. \tag{12}$$

From Theorem 2.1 we derive the following corollary:

Corollary 2.1: The co-tree $\Lambda = \Lambda[k_1, \cdots, k_n]$ has full spectrum if and only if, for every $\nu = 0, 1, \cdots, n-2$,

$$\left(\frac{1}{N_\nu}\right)^{\frac{N_\nu}{\rho_\nu}} \left(\frac{1}{N_{\nu+2}}\right)^{1-\frac{N_\nu}{\rho_\nu}} \leq \frac{1}{N_{\nu+1}}, \tag{13}$$

where the numbers $\rho_0, \cdots, \rho_{n-1}$ are given by

$$\rho_\nu = (1 + k_{\nu+1}) N_{\nu+1} = N_\nu + N_{\nu+1} - 1\,.$$

The simple derivation is left to the reader.

Specializing further we obtain the following corollary which extends Ryabko's theorem [5] in a natural way (Ryabko's theorem corresponds to the case $k_1 = \cdots = k_n = 1$ which gives $N_\nu = n - \nu + 1\,;\ \nu = 0, 1, \cdots, n$):

Corollary 2.2: Every co-tree $\Lambda = \Lambda[k_1, \cdots, k_n]$ with $k_1 \geq k_2 \geq \cdots \geq k_n$ has full spectrum and the universal predictor P^* is given by

$$\begin{aligned} P^*(a) &= N_\nu^{-N_\nu} (N_{\nu+1})^{k_{\nu+1} N_{\nu+1}} / Z \\ &= N_\nu^{-N_\nu} (N_{\nu+1})^{N_\nu - 1} / Z \end{aligned}$$

for all points a in level ν ($\nu = 0, 1, \cdots, n$) with Z a normalization constant.

Proof: Once we have proved that (13) holds when $k_1 \geq \cdots \geq k_n$, the formula for P^* follows from Theorem 2.1 and (10). Note that the left hand side of (13), call it G, is a geometric mean and that the corresponding arithmetic mean is

$$A = \frac{1}{\rho_\nu} + \frac{\rho_\nu - N_\nu}{\rho_\nu N_{\nu+2}}$$

which can be written as

$$\frac{1 + k_{\nu+2}}{N_{\nu+1}(1 + k_{\nu+1})},$$

thus, when $k_1 \geq \cdots \geq k_n$, $A \leq \frac{1}{N_{\nu+1}}$, hence also $G \leq \frac{1}{N_{\nu+1}}$ holds, i.e. (13) does indeed hold. ■

Note that the cases 1,2, and 3 from Section I can be discussed based on the results of this section.

III. Relativization

Experience tells us that for typical optimization problems of the nature we are studying, "normalization" (via a "partition function" or constant) is natural. This is for instance reflected by the appearance of Z in Table 1. A natural idea then is to facilitate the search for universal objects by a prior normalization. We find it advantageous to work with codes rather than with distributions. Then, rather than normalizing via a division, we should normalize by a suitable subtraction. This then leeds to objects measured relative to optimal performance in some sense, and we speak about a process of *relativization*.

Relativization may be defined quite generally. However, we shall only have co-trees in mind for the present study. Therefore, let Λ denote a fixed co-tree and denote as usual by $\mathrm{R}_{\min}$ the minimax redundancy for the order model $\mathcal{P} = \mathcal{P}(\Lambda)$. Motivated by the considerations above, we introduce the *relativized universal code* as the function

$$\tilde{\kappa}^* = \kappa^* - \mathrm{R}_{\min}\,.$$

We first characterize this function among all *monotone* functions $\phi : \Lambda \to \mathbb{R}$. Here, monotonicity means that $\phi(b) \leq \phi(a)$ whenever $b \leq a$. A node $a \in \Lambda$ is *ϕ-active* if either a is a maximal node or else $\phi(a) < \phi(a^+)$ (recall that a^+ denotes the immediate successor of a). If a is not ϕ-active, a is *ϕ-inactive*. If $\phi = \tilde{\kappa}^*$ (or if $\phi = \kappa^*$), we regain the notion of active and inactive nodes introduced in Section I.

For the convenient formulation of the result below, we introduce the *accumulated function* ϕ^σ as the function defined on subsets of Λ by

$$\phi^\sigma(\Delta) = \sum_{b \in \Delta} \phi(b)\,. \tag{14}$$

Proposition 3.1: A real-valued function ϕ defined on Λ coincides with the relativized universal code $\tilde{\kappa}^*$ if and only if it is monotone and satisfies the two requirements:

$$\phi^\sigma(a^\downarrow) = \overline{N}(a) \text{ for every } \phi\text{-active node } a\,, \tag{15}$$

$$\phi^\sigma(a^\downarrow) \leq \overline{N}(a) \text{ for every node } a \in \Lambda\,. \tag{16}$$

Proof: We start with some preliminary observations related to any function ϕ on Λ. Put $R = \ln \sum_{a \in \Lambda} e^{-\phi(a)}$ and define κ as the code obtained from ϕ by "de-relativization", i.e. $\kappa = \phi + R$. Then κ is indeed a code: $\kappa \in \mathrm{K}(\Lambda)$. Let P denote the matching distribution. We claim that, for every node a, the equivalences

$$\mathrm{D}(U_a \| P) = R \Leftrightarrow \phi^\sigma(a^\downarrow) = \overline{N}(a)\,, \tag{17}$$

$$\mathrm{D}(U_a \| P) \leq R \Leftrightarrow \phi^\sigma(a^\downarrow) \leq \overline{N}(a) \tag{18}$$

hold. These equivalences are proved in a similar manner and we only give the details regarding (18). Add $\ln N(a)$ to the inequality on the left hand side and appeal to the linking identity (5), and you realize that this inequality is equivalent to the inequality $\kappa^\sigma(a^\downarrow) \leq \overline{N}(a) + N(a)R$, hence also, as claimed, to the inequality $\phi^\sigma(a^\downarrow) \leq \overline{N}(a)$.

Now assume that the conditions stated in the lemma hold for a function ϕ on Λ. By monotonicity of ϕ, $P \in \mathcal{P}$. Then, by the assumptions (15) and (16), we see from (17) and (18) that the conditions of the Kuhn-Tucker criterion, Proposition 1.2, are fulfilled. Therefore P is the universal predictor and hence $\phi = \tilde{\kappa}^*$.

Necessity of the conditions of the lemma follow in a similar way from necessity of the Kuhn-Tucker conditions which was stated after the proof of Proposition 1.2. In view of the focus on sufficiency, we do not provide the details. ■

We note that by applying the procedure in the first part of the proof, one obtains the universal code κ^* from the relativized universal code $\tilde{\kappa}^*$ by a simple process of de-relativization:

Corollary 3.1: The minimum redundancy and the universal code can be obtained from the relativized universal code by the formulas

$$\mathrm{R}_{\min} = \ln \sum_{a\in\Lambda} e^{-\tilde{\kappa}^*(a)}$$
$$\kappa^*(a) = \tilde{\kappa}^*(a) + \mathrm{R}_{\min} \text{ for } a \in \Lambda\,.$$

For each node $a \in \Lambda$, the left-section $a^\downarrow$ defines a co-tree in its own right. As an immediate corollary to Lemma 3.1 we find the following result:

Corollary 3.2: If $a \in \sigma(\Lambda)$, then the relativized universal code for the co-tree $a^\downarrow$ is obtained by restricting the relativized universal code for Λ to $a^\downarrow$. In particular, $\sigma(a^\downarrow) = \sigma(\Lambda) \cap a^\downarrow$.

We stress the importance of the assumption that a be active in this result. Without this assumption new active nodes in $a^\downarrow$ may appear, indeed a will become active, cf. Proposition 1.3 or Case 3 from Figure 1.

For the further study, consider, for any $a \in \Lambda$, the *control* of a, denoted $\overline{a}$, defined as the closest active node greater than or equal to a (i.e. $\overline{a} \in \sigma(\Lambda)$, $\overline{a} \geq a$ and no $c \in \sigma(\Lambda)$ satisfies $a \leq c < \overline{a}$). By Proposition 1.3, $\overline{a}$ is well defined for all $a \in \Lambda$. Clearly, $\overline{a} = a$ if and only if $a \in \sigma(\Lambda)$. The significance of the notion is summarized in the following simple facts:

Lemma 3.1: Let $a \in \Lambda$. Then, for any b with $a \leq b \leq \overline{a}$, $P^*(\overline{a}) = P^*(a)$ and, therefore, also $\tilde{\kappa}^*(\overline{a}) = \tilde{\kappa}^*(a)$ holds, whereas, if $b > \overline{a}$, then $P^*(b) < P^*(a)$ and hence $\tilde{\kappa}^*(b) > \tilde{\kappa}^*(a)$ holds.

Proof: We may assume that $a \in \Lambda \setminus \sigma(\Lambda)$. Let $P^* = \sum_{c\in\Lambda} w_c U_c$ be the barycentric decomposition of P^*. Then, for every c with $a \leq c < \overline{a}$, $w_c = 0$, hence $P^*(a) = P^*(\overline{a})$. If $b > \overline{a}$, $P^*(b) < P^*(\overline{a}) = P^*(a)$ as $w_{\overline{a}} > 0$. The stated properties follow. ■

Together with other facts, the lemma is used for the proof of the following useful result:

Proposition 3.2: Every minimal node of Λ is active. The relativized universal code is non-negative and vanishes on the minimal nodes – and nowhere else. The universal predictor assumes its maximal value on every minimal node and any other node has a strictly smaller probability.

Proof: Let a be a minimal node and put $b = \overline{a}$. Then $\tilde{\kappa}^*(a) = \tilde{\kappa}^*(b)$. By monotonicity of $\tilde{\kappa}^*$, $\tilde{\kappa}^*(b)$ is bounded below by the average $\frac{1}{N(b)}(\tilde{\kappa}^*)^\sigma(b^\downarrow)$ thus, by (15), $\tilde{\kappa}^*(b) \geq \ln N(b)$. Now,

$$0 = \overline{N}(a) \geq (\tilde{\kappa}^*)^\sigma(a^\downarrow) = \tilde{\kappa}^*(a) = \tilde{\kappa}^*(b) \geq \ln N(b)$$

and $N(b) = 1$, hence $b = a$ follows. We conclude that a is active. Thus minimal nodes are indeed active. We leave the proof of the remaining parts of the proposition to the reader, referring to the fact just established and to Proposition 3.1. ■

The proposition illuminates the definition of the relativized universal code $\tilde{\kappa}^*$. Indeed, we realize that $\tilde{\kappa}^*$ measures code length relative to the shortest codeword. This property is specific to co-trees and does not hold if relativization is considered more generally.

As a last application of the structure related to the notion of control, we establish the following result:

Proposition 3.3: For any node which is not active, the inequality of (16) is sharp, i.e. for such a node, $\tilde{\kappa}^{*\sigma}(a) < \overline{N}(a)$.

Proof: If $a \notin \sigma(\Lambda)$ then $\overline{a} > a$ and $\tilde{\kappa}^{*\sigma}(a) = \tilde{\kappa}^{*\sigma}(\overline{a}) = \overline{N}(\overline{a}) > \overline{N}(a)$ follows. ■

IV. Ideas on the way to an algorithm

Again, we consider the order model $\mathcal{P}$ for a co-tree Λ. We aim at developing an efficient algorithm for the determination of the universal objects. Instead of going directly into this we shall take time in this section first to explain the ideas behind.

The algorithm builds strongly on properties of certain special sets and certain associated numbers, called *brackets*. In order to motivate the introduction of these objects, we first observe that if, somehow, the spectrum $\sigma(\Lambda)$ is known, $\tilde{\kappa}^*$ is easy to calculate. As $\tilde{\kappa}^*(a) = \tilde{\kappa}^*(\overline{a})$ holds generally, we need only worry about the values of $\tilde{\kappa}^*$ for active nodes. So, let $a \in \sigma(\Lambda)$. If a is minimal, $\tilde{\kappa}^*(a) = 0$. If a is not minimal, denote by T the set of maximal nodes in $\big(a^\downarrow \cap \sigma(\Lambda)\big) \setminus \{a\}$. By Proposition 3.2, T is a "cross-section" of $a^\downarrow$ in the sense that every path from a to a minimal node meets T in exactly one point. We put $B = \bigcup_{t\in T} t^\downarrow$ and $S = a^\downarrow \setminus B$. Then $\tilde{\kappa}^*$ assumes the same value, $\tilde{\kappa}^*(a)$, on all nodes in S and considering the decomposition of $a^\downarrow$ in the two sets S and B, we find from Proposition 3.1 that

$$\begin{aligned}\overline{N}(a) = (\tilde{\kappa}^*)^\sigma(a^\downarrow) &= |S| \cdot \tilde{\kappa}^*(a) + (\tilde{\kappa}^*)^\sigma(B)\\ &= |S| \cdot \tilde{\kappa}^*(a) + \sum_{t\in T} (\tilde{\kappa}^*)^\sigma(t^\downarrow) = |S| \cdot \tilde{\kappa}^*(a) + \sum_{t\in T} \overline{N}(t)\\ &= |S| \cdot \tilde{\kappa}^*(a) + \overline{N}^\sigma(T)\,,\end{aligned}$$

and we conclude that

$$\tilde{\kappa}^*(a) = \frac{\overline{N}(a) - \overline{N}^\sigma(T)}{|S|} = \frac{\overline{N}(a) - \overline{N}^\sigma(T)}{N(a) - N^\sigma(T)}\,. \tag{19}$$

Note that this formula holds for all active nodes, including the minimal ones (for which $T = B = \emptyset$ and $S = \{a\}$).

The special sets we shall work with will mimic the roles of the sets T, B and S above. We find it convenient to take "B-type sets" as the basic type and derive "T"- and "S"-type sets from them. Thus, we call a set B a *blocking set for* $a \in \Lambda$ if B is a hereditary subset of $a^\downarrow \setminus \{a\}$ which contains all minimal nodes of $a^\downarrow \setminus \{a\}$. For such a set we define the *exterior of* B *in* a, $S_B(a)$, and the *ceiling of* B *in* a, $T_B(a)$, by

$$S_B(a) = a^\downarrow \setminus B\,, \tag{20}$$
$$T_B(a) = \text{ set of maximal nodes of } B\,. \tag{21}$$

The nodes in $T_B(a)$ are the first nodes in B we meet on paths from a to a minimal node. Note that the exterior $S_B(a)$ is always non-empty. The same is true for the sets B and $T_B(a)$ unless a is a minimal node of Λ, in which case only the empty set is blocking for a. Occasionally, we also refer to the *interior of* B *in* a which is the set $B \setminus T_B(a)$. Note that $a^\downarrow$ is the disjoint union of the exterior, the ceiling and the interior of B in a.

Let B be a blocking set for a. Guided by (19), we define the *bracket of a in B,* by

$$[a, B] = \frac{\overline{N}(a) - \overline{N}^{\sigma}\big(T_B(a)\big)}{|S_B(a)|}\,.$$

Note that the denominator above has a similar structure as the numerator. Indeed, $|S_B(a)| = N(a) - N^{\sigma}(T_B(a))$.

The algorithm we aim at depends crucially on the relations between blocking sets and their associated brackets. The properties we need are derived from certain combinatorial identities, the *transitivity identities* of Lemma 5.1 in the next section. Of special interest are blocking sets for a node a for which $[a, B]$ is maximal. By $[a]_{\max}$ we denote the largest value of $[a, B]$ for all blocking sets B for a. It turns out that among the blocking sets B for a with maximal bracket ($[a, B] = [a]_{\max}$), there exists a set-theoretically largest one (Proposition 5.3). This uniquely defined set is denoted $B^*(a)$ and the corresponding ceiling and exterior are denoted by $T^*(a)$ and $S^*(a)$.

With access to these sets, we define sets $T_0^*, T_1^*, \cdots$, the *ceiling hierarchy*, by a construction "from the top": We start with T_0^*, by definition the set of maximal nodes of Λ (the set of "ancestors"). Then, as T_1^*, we take the union of all sets of the form $T^*(t)$ with $t \in T_0^*$. A node in T_1^* is a "daughter" of the first *generation*. We continue "down the co-tree", thus in the next step consider "daughters" of the 2.nd. generation (i.e. nodes in $T^*(t)$ for some $t \in T_1^*$). Formally, for $i \geq 1$,

$$T_i^* = \bigcup_{t \in T_{i-1}^*} T^*(t)\,. \tag{22}$$

Clearly, the sets T_i^* are eventually empty. By $\overline{\sigma}(\Lambda)$ we denote the union

$$\overline{\sigma}(\Lambda) = \bigcup_{i \geq 0} T_i^*\,. \tag{23}$$

Two notions related to the ceiling hierarchy turns out to be useful. Firstly, for any node a, the *projection* of a on $\overline{\sigma}(\Lambda)$ is the unique node $t \in \overline{\sigma}(\Lambda)$ for which $a \in S^*(t)$. We do not know if this notion coincides with the previously defined notion of control (the $\overline{a}$'s from Section III). Anyhow, it is sufficiently close that we can argue with it in much the same way as in the discussion in the beginning of this section, thereby deriving a formula for $\tilde{\kappa}^*$. The second notion we need associates to any node $s \in \overline{\sigma}(\Lambda) \setminus T_0^*$, the unique node $t \in \overline{\sigma}(\Lambda)$ such that $s \in T^*(t)$. We call this node the *mother* of s. Using this notion we can characterize the spectrum as a certain subset of $\overline{\sigma}(\Lambda)$. The facts indicated constitute the content of our first main result, Theorem 5.1. We point out that perhaps $\overline{\sigma}(\Lambda) = \sigma(\Lambda)$ holds generally, but we do not know this. In spite of this unclear point, the characterizations of $\tilde{\kappa}^*$ and $\sigma(\Lambda)$ in Theorem 5.1 will be completely satisfactory, considering our aims. If you wish, $\overline{\sigma}(\Lambda)$ is the *extended spectrum* of Λ.

The construction behind Theorem 5.1 depends on the blocking sets $B^*(a)$. A naive search for these sets will require exponential time in the size of the problem (e.g. measured by the number of nodes in Λ). To develop an efficient algorithm, new ideas are needed. What we will do is to revert the construction and work "from the bottom" through the *minimality components* $M_0, M_1, \cdots, M_h$. Here, h is the height of Λ and the decomposition $\Lambda = M_0 \cup M_1 \cup \cdots \cup M_h$ is obtained by successive removals of minimal nodes, i.e. M_i is the set of minimal nodes of the co-tree

$$\Lambda \setminus \bigcup_{0 \leq j < i} M_j\,.$$

Let us explain in more detail why a construction from the bottom is to prefer. Fact is that when you work from the top, and consider candidates for the B^*-,S^*- and T^*-sets without knowing these sets for nodes further down the co-tree, you risk that after some time an inconsistency occurs and this will force you to discard previous work, and to start afresh. Quite differently, when you work from the bottom, the sets concerned remain unchanged once constructed as they are not influenced by the development further up in the co-tree. It should, however, be remarked that sets already constructed may later turn out to be superfluous as sets associated with nodes higher up in the co-tree, say nodes $b > a$, may "overshadow" sets already constructed in the sense that $S^*(b) \supseteq S^*(a)$ may happen. Anyhow, the main point is that during construction from the bottom, you proceed incrementally without discarding previous work. The insight needed to see that the algorithm from the bottom works will be developed in Section VI. As an indication of the good sense in working from the bottom we may also point to the fact that at least the start is unproblematic since then blocking sets are empty and brackets vanish.

The central part of the algorithm is a certain subroutine, referred to as the *central subroutine* which is called several times during the execution of the overall algorithm. The flow diagram for this subroutine is sketched in Figure 2. As input to the subroutine one takes a node $a \in \Lambda$, and as output the subroutine provides you with $B^*(a)$ and $[a]_{\max} = [a, B^*(a)]$, it being understood that the corresponding objects associated with nodes in $a^{\downarrow} \setminus \{a\}$ are already known. Together with $B^*(a)$, also $T^*(a)$ is recorded. Therefore, when the central subroutine has been called for all nodes in the co-tree as input, all ceilings $T^*(a)$ and all maximal brackets $[a]_{\max}$ will be known. Then, as the last step in the construction of $\tilde{\kappa}^*$, we again work "from the top" by appealing to Theorem 5.1. This provides you directly with the relativized universal code from which the universal code (hence also the universal predictor) may be constructed quite easily as explained in Section III.

V. Construction from the top

We start by developing some properties of blocking sets and brackets.

Proposition 5.1: Let B be a blocking set for a. Then the bracket $[a, B]$ vanishes if a is a minimal node and is positive otherwise.

Proof: If a is minimal, $B = \emptyset$ and the definition gives $[a, \emptyset] = 0$. If a is not minimal, put $M = \sum_{t \in T_B(a)} N(t)$ and note that $N(a) > M \geq 1$, hence the positivity of $[a, B]$

follows from the manipulations

$$\begin{aligned}&|S_B(a)|[a,B]\\&=N(a)\ln N(a)-\sum_{t\in T_B(a)}N(t)\ln\frac{N(t)}{M}-M\ln M\\&\geq N(a)\ln N(a)-M\ln M>0\,.\end{aligned}$$

■

We shall show that based on the brackets alone (thus not assuming any special knowledge about the spectrum $\sigma(\Lambda)$), the universal code can be constructed. Proposition 3.1 is an important step in this direction but there are obstacles to overcome in connection with the necessary checking of inequalities related both to (16) and to the requirement of monotonicity. It turns out that these problems can be handled efficiently, based on certain identities which allows one to compare brackets among each other. These comparisons involve two simple type of constructions, *filling* and *restriction*. Specifically, if B is a subset of Λ, and b any node, the *filling of B at b*, denoted $B\vee b$, is the set $B\cup b^{\downarrow}$. Typically, this construction is used if B is a blocking set for some node $a>b$. Then, if $b\notin B$, the process of filling leads to a new blocking set for a. We also consider the *restriction of B to $b^{\downarrow}$*, denoted by $B\wedge b$, and defined by $B\wedge b=B\cap b^{\downarrow}$. If B is a blocking set for some node $a>b$ and $b\notin B$, the restriction $B\wedge b$ will be a blocking set for b.

Lemma 5.1 (transitivity identities, basic case): Let $a>b$, let B be a blocking set for a and assume that $b\notin B$. Put $B^+=B\vee b$ and $B^-=B\wedge b$. Then the following identities hold:

$$|S_{B^+}(a)|\Big([b,B^-]-[a,B^+]\Big)=|S_B(a)|\Big([b,B^-]-[a,B]\Big)\,, \tag{24}$$

$$|S_B(a)|\Big([a,B]-[a,B^+]\Big)=|S_{B^-}(b)|\Big([b,B^-]-[a,B^+]\Big)\,, \tag{25}$$

$$|S_{B^-}(b)|\Big([b,B^-]-[a,B]\Big)=|S_{B^+}(a)|\Big([a,B]-[a,B^+]\Big)\,. \tag{26}$$

Proof: In view of the equality

$$|S_B(a)|=|S_{B^+}(a)|+|S_{B^-}(b)|\,,$$

each of the three identities can be derived from any of the other two. It therefore suffices to verify (24). For this, we exploit the equality above and the fact that $T_{B^+}(a)$ is the disjoint union of $\{b\}$ and the proper set-difference $T_B(a)\setminus T_{B^-}(b)$. Appealing also to the definition of brackets, we find that

$$\begin{aligned}&|S_B(a)|\Big([b,B^-]-[a,B]\Big)\\&=|S_B(a)|[b,B^-]-\overline{N}(a)+\overline{N}^{\sigma}(T_B(a))\\&=\Big(|S_{B^+}(a)|+|S_{B^-}(b)|\Big)[b,B^-]\\&\quad-\overline{N}(a)+\overline{N}^{\sigma}(T_{B^+}(a))-\overline{N}(b)+\overline{N}^{\sigma}(T_{B^-}(b))\\&=|S_{B^+}(a)|\Big([b,B^-]-[a,B^+]\Big)\,,\end{aligned}$$

thus (24) holds. ■

In order to ease the notation a bit, we agree that if a set of the form $B\wedge b$ is blocking for b, we may simply say that B is blocking for b and write $S_B(b)$ in place of $S_{B\wedge b}(b)$ and $[b,B]$ in place of $[b,B\wedge b]$. In the formulation of Lemma 5.1 we may thus write $S_B(b)$ rather than $S_{B^-}(b)$ and $[b,B]$ rather than $[b,B^-]$.

As all terms of the form $|S_\cdot(\cdot)|$ are positive, it is clear that we can use (24)-(26) for comparisons of brackets. We shall soon see instances of this. For now we note that the lemma implies that the numbers $[a,B]$, $[a,B^+]$ and $[b,B]$ are either identical or else $[a,B]$ lies strictly between $[a,B^+]$ and $[b,B]$, i.e. either $[a,B^+]<[a,B]<[b,B]$ or $[b,B]<[a,B]<[a,B^+]$ holds.

The transitive nature of the lemma is best revealed by generalizing the result. This we shall do in Section VIII.

We continue with some important observations based on Lemma 5.1 which involve special blocking sets. The blocking set B for a has *maximal bracket* if $[a,B]=[a]_{\max}$. Such a set is *set-theoretically maximal (minimal)* if it is not a proper subset (superset) of some other blocking set for a with maximal bracket.

Proposition 5.2: Let $a\in\Lambda$ and let B^* be a blocking set for a with maximal bracket.

(i) (monotonicity): If $b\in T_{B^*}(a)$, then $[a]_{\max}\geq[b]_{\max}$. If B^* is set-theoretically minimal, the sharp inequality $[a]_{\max}>[b]_{\max}$ holds;

(ii) (boundedness): If $b\in S_{B^*}(a)\setminus\{a\}$, then $[a]_{\max}\leq[b,B^*]$ and this inequality is sharp if B^* is set-theoretically maximal.

Proof: (i): If b is minimal, a cannot be minimal and the result follows from Proposition 5.1. Assume then that b is not minimal and denote by B any blocking set for b with maximal bracket. The set $B_0=(B^*\setminus\{b\})\cup B$ is a proper subset of B^* which is blocking for a. Then $[a]_{\max}\geq[a,B_0]$ with sharp inequality if B^* is set-theoretically minimal. By (25) applied to the set B_0 it then follows that $[a,B^*]\geq[b,B]$, i.e. $[a]_{\max}\geq[b]_{\max}$, with sharp inequality if B^* is set-theoretically minimal.

(ii): This follows by applying (26) with $B=B^*$. ■

Exploiting these results we obtain a useful uniqueness property:

Proposition 5.3: (uniqueness) For every node a, there exist two uniquely defined blocking sets for a with maximal bracket, $B^*(a)$ and $B_*(a)$, characterized as, respectively the set-theoretically largest such set and the set-theoretically smallest such set. In particular, for every blocking set B for a with maximal bracket, the inclusions $B_*(a)\subseteq B\subseteq B^*(a)$ hold.

Proof: Let B^* be a set-theoretically maximal blocking set for a with maximal bracket. Let $\tilde{B}^*$ be any blocking set for a with maximal bracket. We shall prove that $\tilde{B}^*\subseteq B^*$. Assume the contrary. Then there exists $b\in\tilde{B}^*\setminus B^*$, hence there also exists $b^{'}\in\tilde{T}^*\setminus B^*$ where $\tilde{T}^*$ denotes the ceiling of $\tilde{B}^*$ in a. By monotonicity, $[a,\tilde{B}^*]\geq[b^{'},B^*]$. And, by boundedness, $[b^{'},B^*]>[a,B^*]$. The two inequalities show that $[a,\tilde{B}^*]>[a,B^*]$ which is a contradiction as $[a,B^*]=[a,\tilde{B}^*]=[a]_{\max}$. We conclude, as desired, that $\tilde{B}^*\subseteq B^*$. The reverse inclusion is proved in a similar way when also $\tilde{B}^*$ is set-theoretically maximal. As any two set-theoretically maximal blocking sets for a with maximal brackets are equal, the largest such set, denoted $B^*(a)$, is well defined.

The facts needed to establish the results pertaining to minimal blocking sets are proved in a similar way. Details are left to the reader. ■

We do not know if there is a unique blocking set for a with maximal bracket, i.e. if $B_*(a) = B^*(a)$ holds generally.

For the constructions to follow, we have chosen to focus on the largest sets, the $B^*(a)$'s. We denote by $T^*(a)$ the ceiling in a associated with $B^*(a)$ and by $S^*(a)$ the exterior in a associated with $B^*(a)$. These are the sets we shall use for the construction of $\tilde{\kappa}^*$. Recall the introduction in Section IV of the ceiling hierarchy $(T_i^*)_{i\geq 0}$, cf. in particular (22) and (23). The largest index i with $T_i^* \neq \emptyset$ is the *ceiling index*. The ceiling index is at most the height of Λ, but is often smaller, e.g. for Case 3 of Figure 1, $\delta = 1$ whereas $h = 2$. In the extreme case when every maximal node is also a minimal node, T_0^* is the only non-empty set in the hierarchy and the ceiling index is 0.

Based on the ceiling hierarchy we define a decomposition $(S_i^*)_{0\leq i\leq \delta}$ of Λ, with δ the ceiling index:

$$S_i^* = \bigcup_{a\in T_i^*} S^*(a) = \{a \in \Lambda | pr(a) \in T_i^*\} . \qquad (27)$$

The second characterization refers to the notion of projection introduced in Section IV. For the mother t of a node $s \in \overline{\sigma}(\Lambda) \setminus T_0^*$ we use the notation $t = \mu(s)$.

We can now state the main result of this section.

Theorem 5.1: With reference to the ceiling hierarchy and associated notions, the relativized universal code is given by

$$\tilde{\kappa}^*(a) = [pr(a)]_{\max} \text{ for all } a \in \Lambda , \qquad (28)$$

and the spectrum of Λ is the following subset of $\overline{\sigma}(\Lambda)$:

$$\sigma(\Lambda) = T_0^* \cup \{t \in \overline{\sigma}(\Lambda) \setminus T_0^* | \, [\mu(t)]_{\max} > [t]_{\max}\} . \qquad (29)$$

Proof: Denote by ϕ the function on Λ defined by $\phi(a) = [pr(a)]_{\max}$.

We shall verify the conditions of Proposition 3.1.

First, to prove monotonicity of ϕ, consider any path from a maximal node to a minimal node. Let $t_0 > t_1 > \cdots > t_k$ be the nodes in $\overline{\sigma}(\Lambda)$ on the path (thus t_k is a minimal node of Λ). Then, by monotonicity, cf. Proposition 5.2, $\phi(t_0) \geq \phi(t_1) \geq \cdots \geq \phi(t_k)$ and, by the definition of ϕ, $\phi(a) = \phi(t_i)$ for nodes on the path with $t_{i-1} < a \leq t_i$ (here, $0 \leq i < k$). This proves monotonicity along any path connecting a maximal node with a minimal node. Clearly then, ϕ is monotone on all of Λ. The argument above also shows that all nodes b with $t_i > b > t_{i+1}$ are inactive, thus

$$\sigma(\Lambda) \subseteq \overline{\sigma}(\Lambda) . \qquad (30)$$

Next, we consider a node $a \in \overline{\sigma}(\Lambda)$, say $a \in T_i^*$, and show that (15) holds. Put $U_j = T_j^* \cap a^{\downarrow}$ and $V_j = S_j^* \cap a^{\downarrow}$ for $j \geq i$. Let k be the largest integer such that $U_j \neq \emptyset$. Then (15) for the node a follows from the string of equalities:

$$\begin{aligned}\phi^{\sigma}(a^{\downarrow}) &= \sum_{j=i}^{k} \sum_{b\in V_j} \phi(b) = \sum_{j=i}^{k} \sum_{t\in U_j} |S^*(t)|\phi(t)\\ &= \sum_{j=i}^{k} \sum_{t\in U_j} \Big(\overline{N}(t) - \overline{N}^{\sigma}(T^*(t))\Big)\\ &= \sum_{j=i}^{k} \Big(\overline{N}^{\sigma}(U_j) - \overline{N}^{\sigma}(U_{j+1})\Big) = \overline{N}(a) .\end{aligned}$$

By (30), the validity of (15) for all $a \in \sigma(\Lambda)$ follows.

Finally, consider a node $b \in \Lambda \setminus \overline{\sigma}(\Lambda)$. To finish the proof, we need only establish the inequality (16) for b. In fact, we shall show that the sharp inequality $\phi^{\sigma}(b^{\downarrow}) < \overline{N}(b)$ holds. To this end, put $a = pr(b)$ and $B = B^*(a) \cap b^{\downarrow}$ and use results already established and the boundedness property of Proposition 5.2, to find that

$$\begin{aligned}\phi^{\sigma}(b^{\downarrow}) &= |S_B(b)|\phi(b) + \sum_{t\in T_B(b)} \phi^{\sigma}(t^{\downarrow})\\ &= |S_B(b)|[a]_{\max} + \sum_{t\in T_B(b)} \overline{N}(t)\\ &< |S_B(b)|[b, B] + \overline{N}^{\sigma}(T_B(b)) = \overline{N}(b) .\end{aligned}$$

We have now seen that $\phi = \tilde{\kappa}^*$, hence (28) holds. As the spectrum consists of the points of increase of $\tilde{\kappa}^*$, (29) follows. ■

We do not know if the inclusion (30) can be sharpened to an identity. Note that by the last part of the monotonicity statement of Proposition 5.2 this will be the case if $B^*(a) = B_*(a)$ holds generally.

VI. The central subroutine

We continue the study of universal objects associated with the model $\mathcal{P}(\Lambda)$ over a co-tree Λ.

The construction in Theorem 5.1 builds on the sets $B^*(a)$. As noted in Section IV, the theorem cannot be used directly to obtain an algorithm of low complexity. Instead, we speed up the construction by working "from the bottom" based on the decomposition $\Lambda = M_0 \cup M_1 \cup \cdots \cup M_h$ in minimality components.

We shall determine the B^*-sets for all nodes. For nodes in M_0 this is trivial, and we start by considering nodes in M_1, continue with nodes in M_2, and so on until we get at the nodes in M_h. We will assume that the decomposition in minimality components is given off-hand and not be concerned with the time it takes to determine this decomposition. The approach depends on the fact that given the co-tree, e.g. in terms of the standard representation as explained in Section I, the decomposition can be determined by an efficient algorithm.

The two propositions to follow are important technical tools needed to develop an efficient algorithm.

Proposition 6.1: (Γ-structure) Let $a \in \Lambda$. Then, for every $b \in S^*(a)$, the inclusion $S^*(b) \subseteq S^*(a)$ or, equivalently, $B^*(a) \wedge b \subseteq B^*(b)$ holds.

The name attached to the result lies in the shape of the letter “Γ” and will appear natural when we specialize to co-trees with uniform branching in the next section.

Proof: We shall actually prove a formally stronger result, viz. that, for $b \in S^*(a) \setminus \{a\}$,

$$B^*(a) \wedge b \subseteq B_*(b)\,. \tag{31}$$

Assume, for the purpose of an indirect proof, that this is not the case. Then, for some $b \in S^*(a) \setminus \{a\}$, there exists $t \in \big(T^*(a) \wedge b\big) \setminus B_*(b)$. We find that

$$\begin{aligned}[a]_{\max} \geq [t]_{\max} \geq [t, B_*(b)] &\geq [b, B_*(b)] \\ = [b]_{\max} &\geq [b, B^*(a)] > [a]_{\max}\,.\end{aligned}$$

Indeed, the first inequality follows by monotonicity as $t \in T^*(a)$, the second follows as $B_*(b)$ is blocking for t, the third follows by boundedness as $t \in b^{\downarrow} \backslash B_*(b)$, the equality is trivial, the next inequality follows as $B^*(a)$ is blocking for b and the last one follows by boundedness as $b \in S^*(a) \backslash \{a\}$. From the resulting contradiction we conclude that (31) holds. ■

The stronger result actually proved above supports the view that “normally” $B_*(a) = B^*(a)$.

For our second auxiliary result, let us agree to say that a blocking set B for a node a has the *monotonicity property* if, for any $t \in T_B(a)$, $[a, B] \geq [t]_{\max}$.

Proposition 6.2: (Characterization): For any $a \in \Lambda$, $B^*(a)$ can be characterized as the largest blocking set for a with the monotonicity property.

When applying this result we have a construction “from the bottom” in mind. Then the characterization makes good sense since, when searching for the set $B^*(a)$, all sets $B^*(b)$ with $b \in a^{\downarrow} \backslash \{a\}$ will be known and thus the monotonicity property can be checked for any candidate set B we may suggest for $B^*(a)$.

Proof: Let B be the largest blocking set for a with the monotonicity property. As $B^*(a)$ is a blocking set for a and has the monotonicity property, $B^*(a) \subseteq B$. To prove the reverse inclusion, assume, for the purpose of an indirect proof, that this is not the case. Then there exists $t \in T_B(a) \cap S^*(a)$ and we find that

$$[a, B] \geq [t]_{\max} \geq [t, B^*(a)] > [a]_{\max} \geq [a, B]\,.$$

This is a contradiction and we conclude that $B \subseteq B^*(a)$, hence $B = B^*(a)$ as claimed. ■

Before we turn to a development of the algorithm we aim at, we emphasize that in estimating the complexity of the algorithm, we shall neglect any contribution from efforts to make basic information about a co-tree studied accessible to us in a convenient form. We shall thus talk about *essential complexity* of the algorithms.

It turns out that the basic information we shall need about any specific co-tree can be listed as follows:

- the decomposition in minimality components, $\Lambda = M_0 \cup \cdots \cup M_h$,
- the map $a \curvearrowright a^-$ which makes the immediate predecessors of any node accessible to us,
- the map $a \curvearrowright a^{\downarrow}$ which gives access to the left sections,
- the map $a \curvearrowright N(a)$ and, finally,
- the map $a \curvearrowright \overline{N}(a)$.

Of course, there is some redundancy in this list (especially, $\overline{N}$ is given in terms of N). However, the list is chosen for convenience in view of the algorithm to follow. We shall not worry much about how the basic information can be provided, only remark that it is clear that if we specify a co-tree by the standard representation, the desired information can be provided via efficient algorithms operating on the underlying set of finite sequences.

The algorithm we shall now describe is based on Theorem 5.1 which shows that if we know, for every node a, the ceiling $T^*(a)$ as well as the maximal bracket $[a]_{\max}$, then it is easy to determine the relativized universal code, and hence the universal code and the universal predictor. Our algorithm calls several times the *central subroutine*, see Figure 2, which, for a given input a, calculates the *key objects* associated with a, taken to be the sets $B^*(a)$ and $T^*(a)$ and the number $[a]_{\max}$. Note that we find it convenient to work with both $B^*(a)$ and $T^*(a)$, though the one may of course be determined from the other.

For the minimal nodes $a \in M_0$, we already know what the key objects are and there is no reason to call any subroutine for these nodes. To determine the key objects associated with any node, we first call the central subroutine for nodes in the minimality component M_1, then for nodes in M_2 and so on until we get to the nodes in M_h (with h the height of Λ).

Let us have a closer look at the central subroutine. Consider a particular input $a \in \Lambda \setminus M_0$. When the subroutine is called it is assumed, though not shown explicitly in the flow diagram, that key objects about preceeding nodes have already been determined. Actually, this will be the case by the procedure chosen as nodes in $(M_0), M_1, \cdots, M_h$ are called in succession.

We use B, T and α as place-holders for the sought key objects associated with a. The largest blocking set for a altogether is $a^{\downarrow} \setminus \{a\}$. This is the first set we will test and our initial assignment box puts $B := a^{\downarrow} \setminus \{a\}$. We also right away assign the appropriate set to T and the appropriate value to α.

After the introductory assignments, we arrive at the central box, the (b, β)*-box*. It is important that when we come to this box, which may occur many times during the execution of the subroutine, B, T, and α are known to have certain properties: B must be a blocking set for a, $T = T_B(a)$ and $\alpha = [a, B]$ must hold, and then we stress that $B^*(a) \subseteq B$ must be known to hold. In order to carry out the calculations in the (b, β)-box, it is understood that there is a natural way to list the nodes in T, say as $t_1, \cdots, t_k$ (the standard representation of Λ may be used for this purpose). For the calculation, we go through all brackets $[t]_{\max}$ with $t \in T$, note the largest value and then consider the first node among $t_1, \cdots, t_k$ for which the corresponding bracket attains this value. By definition, this is the $\operatorname{Arg}\max$-node. As place-holders for this node and for the corresponding maximal bracket we use b, respectively β and

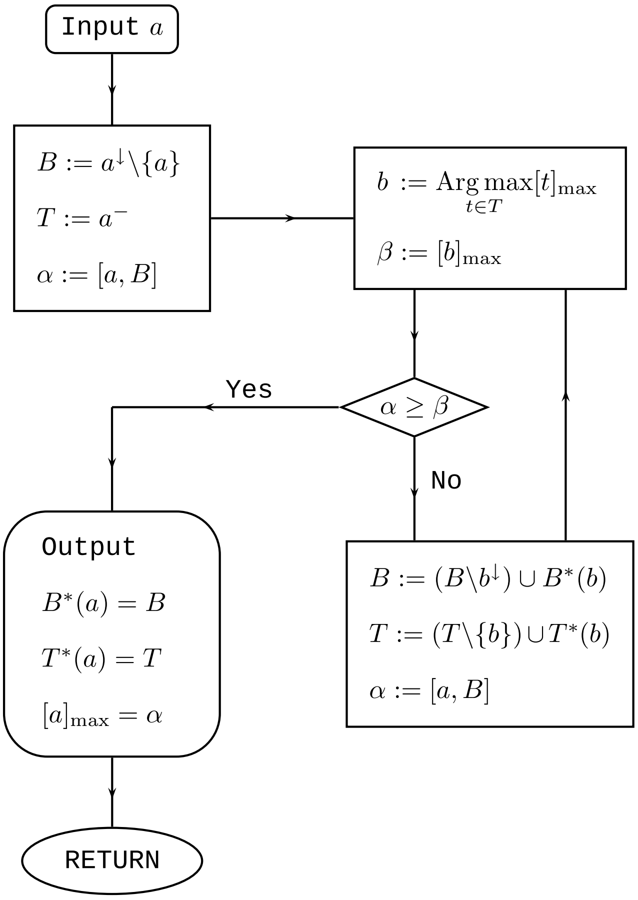

Input a
$B := a^{\downarrow}\backslash\{a\}$
$T := a^{-}$
$\alpha := [a, B]$
$b := \operatorname{Arg}\max_{t\in T}[t]_{\max}$
$\beta := [b]_{\max}$
Yes
$\alpha \geq \beta$
No
Output
$B^*(a) = B$
$T^*(a) = T$
$[a]_{\max} = \alpha$
$B := (B\backslash b^{\downarrow}) \cup B^*(b)$
$T := (T\backslash\{b\}) \cup T^*(b)$
$\alpha := [a, B]$
RETURN

Figure 2. Flow diagram for the central subroutine

thus carry out the assignments

$$b := \operatorname{Arg}\max_{t\in T}[t]_{\max}\,;\; \beta := [b]_{\max}\,.$$

Concerning the calculation of brackets in the central box and elsewhere in the subroutine, this is based on basic information about the co-tree (N's and $\overline{N}$'s) and on output (T^*'s) from previous calls of the subroutine according to the formula

$$[t]_{\max} = \frac{\overline{N}(t) - \sum_{s\in T^*(t)} \overline{N}(s)}{N(t) - \sum_{s\in T^*(t)} N(s)}\,. \tag{32}$$

After the central box comes the test-box "$\alpha \geq \beta$?". We realize that what is tested is really if B has the monotonicity property. If it does, $B = B^*(a)$ by Proposition 6.2 and we go to the output box and then return to the algorithm.

Assume now that the test is negative, i.e. $[a, B] < [b]_{\max}$. It is a key point of the algorithm that then $b \in S^*(a)$ must hold. Assume the contrary. Then, as $B^*(a) \subseteq B$, $b \in T^*(a)$ and by monotonicity we then have $[a]_{\max} \geq [b]_{\max}$. Consider any $t \in T$ and note that

$$[t]_{\max} \leq [b]_{\max} \leq [a]_{\max}\,.$$

By boundedness, we must conclude from this that $t \in B^*(a)$ since, if $t \in S^*(a)$, $[a]_{\max} < [t, B^*(a)] \leq [t]_{\max}$ would hold, contradicting the inequalities above. Thus $T \subseteq B^*(a)$. Since T is the ceiling of B in a and since $B^*(a) \subseteq B$ we conclude that in fact $B = B^*(a)$ must hold. Then B does after all have the monotonicity property of Proposition 6.2. This contradicts the result of the test. All in all we conclude that indeed $b \in S^*(a)$.

Knowing this, we can apply the gamma structure, Proposition 6.1, and find that $B^*(a)$ is a subset of the set $\big(B \setminus \{b\}\big) \cup B^*(b)$. This set is a blocking set for a as b cannot be a minimal node (then $\beta = 0$ would hold and the test would have been positive). We take this set as our new set to be tested and make the proper assignments of B, T and α in the next box of the flow diagram. These possible key objects are then fed into the (b, β)-box and we continue until, eventually, the test for the monotonicity property is positive.

Remarks. Naturally, if the test is negative and there are several nodes in T with $[b, B^*(b)]$ maximal, we may economize and restrict the candidate set further. In more detail, put $m = \max_{t\in T}[t]_{\max}$ and assume that there are several nodes in T, say $b_1, \cdots, b_k$ with maximal bracket m. Then we may as our new assigned key objects take

$$B := \Big(B \setminus \bigcup_{\nu=1}^{k} b_\nu^{\downarrow}\Big) \cup \bigcup_{\nu=1}^{k} B^*(b_\nu)\,, \tag{33}$$

$$T := \big(T \setminus \bigcup_{\nu=1}^{k} \{b_\nu\}\big) \cup \bigcup_{\nu=1}^{k} T^*(b_\nu)\,, \tag{34}$$

$$\alpha := [a, B]\,. \tag{35}$$

It follows from our analysis above that the new set B still contains $B^*(a)$. Further, the α's increase through the subroutine. One way to see this when multiple reductions are performed as in (33) -(35) is to make the reductions step by step. First, put $B_0 = B$ (the old set B) and then define successive reductions by putting

$$B_\nu = \big(B_{\nu-1} \setminus b_\nu^{\downarrow}\big) \cup B^*(b_\nu)$$

for $\nu = 1, \cdots, k$. Then the set B_k is equal to the set defined in (33). This relies on successive applications of (25) and on monotonicity. Details are left to the reader. This remark will be important for the special co-trees to be discussed in the next section.

Other modifications may speed up the execution of the subroutine, e.g. one may note that nodes in M_1 can also, just as minimal nodes, be dealt with outside the subroutine and that some of the information about calculated brackets $[t]_{\max}$ at one stage may be reused for the next stage. We shall not be concerned here with such fine-tunings for general co-trees.

The full algorithm for the calculation of $\tilde{\kappa}^*$, and hence the sought universal objects, consists of the following steps:

- initialization providing basic information about the co-tree,
- trivial assignment of key objects to nodes in M_0,
- call of the central subroutine for all nodes in M_1,
- $\cdots$
- call of the central subroutine for all nodes in M_h,
- top-down construction of the ceiling hierarchy and simultaneous listing of the values of $\tilde{\kappa}^*$, cf. Theorem 5.1.

By the foregoing discussion, it is clear that the algorithm does indeed calculate the desired objects. It is also pretty clear that this is achieved in polynomial time in the size of the problem. Let us discuss this in more detail but only aim at a rough estimate of the efficiency of the algorithm. Firstly, as remarked before, we shall neglect the time consumed during initialization. Also, we shall not be concerned with the memory requirements of the algorithm or with the extra cost incurred by administrative operations involved in the memory management. Further, we shall not discriminate between various basic operations such as additions, subtractions, multiplications, divisions and comparisons of numbers as well as $0,1$-tests (based on known entities). The *essential complexity* of the algorithm, denoted $C(\Lambda)$, is then taken to be the number of basic operations needed from start to end of the algorithm with the reservations as indicated above.

We shall estimate $C(\Lambda)$ in terms of the number n of nodes in Λ. Clearly, $C(\Lambda) \leq n \cdot \max_{a\in\Lambda} C(a)$ where $C(a)$ denotes the essential complexity of the central subroutine when it is called with the node a as input.

For a fixed, we can estimate $C(a)$. Regarding the initial assignments, only the calculation of $[a, B]$ needs to be taken into account. As $[a, B] = \overline{N}(a) - \sum_{t\in a^-} \overline{N}(t)$, at most $|a^-|$ basic operations are needed, hence at most n such operations.

For the cycle "(b, β)-box to test-box to new assignments", this will be visited at most $|a^{\downarrow}|$ many times, hence at most n times. And for one run through the cycle we need at most $|T| \leq n$ basic operations for the determination of (b, β) (as the numbers $[t]_{\max}$ with $t \in T$ are already known). We permit ourselves to ignore the minimal requirement needed to carry out the $\alpha \geq \beta$ test. But we have to consider the requirement related to the new assignments of B, T and α. Regarding B, we need to know, for each node, whether the node is in the set or not. This can be decided by checking membership for each of the three sets B, $b^{\downarrow}$ and $B^*(b)$. As the sets $b^{\downarrow}$ and $B^*(b)$ are known, we only need to test membership for B, and this requires at most n tests. Similarly for T. And regarding α, we realize from (19) that at most $2\cdot|T| \leq 2n$ basic operations are needed. The new assignments thus require at most $4n$ basic operations.

The rough estimates above show that $C(a) \leq n + n(n + 4n) \leq 6n^2$.

We have now completed all elements in the proof of our second main theorem:

Theorem 6.1: The algorithm described above calculates the ceiling hierarchy and thereby the universal objects associated with a co-tree Λ in polynomial time. The essential complexity as defined above is at most $6 \cdot n^3$ where n is the number of nodes in Λ.

Remarks. By studying "worst possible scenarios" regarding the possibilities for the geometric locations of the ceilings calculated by the central subroutine it should be possible to bring down the estimate $6n^3$ quite significantly. We shall look into this in Section VII, but only for co-trees with uniform branching.

As another remark, recall that a main point of our endeavours have been to demonstrate that the universal objects can be determined in closed form. Thus, though the algorithm developed functions well as a numeric algorithm, the end result can easily be expressed in closed form since the basic numbers, the values of $\tilde{\kappa}^*$, are expressed by the brackets $\alpha = [a, B]$ resulting from the central subroutine (perhaps after neglecting certain values which are "overshadowed" during the very last part of the algorithm), and these brackets are given explicitly through (19).

We shall now demonstrate how the algorithm works in practice by investigating a particular, not too complicated and still reasonably "general" example, the co-tree Λ of Figure 3. The work carried out when following the algorithm is conveniently summarized in Figure 4. We have marked in black the top-node (known to be active) as well as all nodes which occur as nodes in a ceiling constructed during the algorithm. In particular, all minimal nodes are marked in black as $\bigcup_{a\in M_1} T^*(a) = M_0$. Furthermore, we have listed the exact as well as the approximate values of all brackets which are constructed during the algorithm and marked with a "dagger" the one and only value (calculated for the node a_{23}) for which the $(\alpha \geq \beta)$-test of the central subroutine is not passed (in the actual case because $\overline{5} - \overline{4} \geq \overline{4}$ does *not* hold).

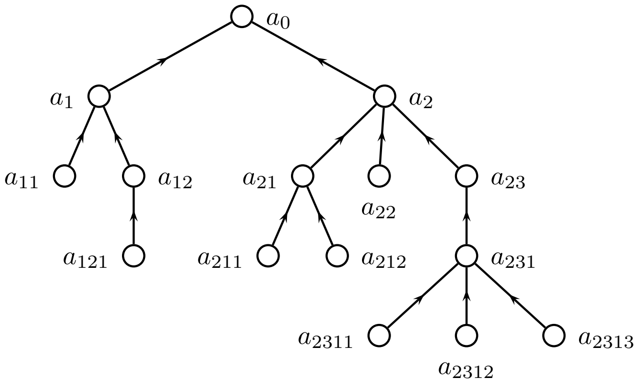


Figure 3. A "general" co-tree Λ.

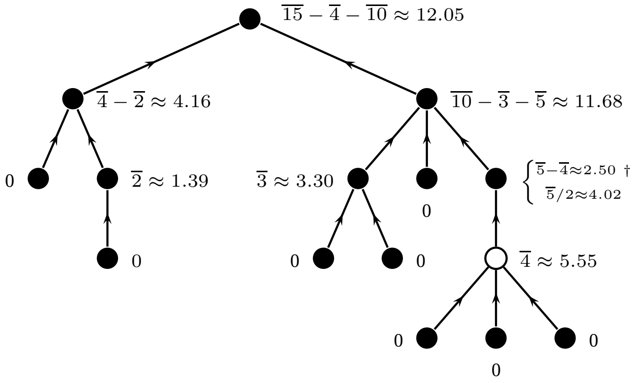
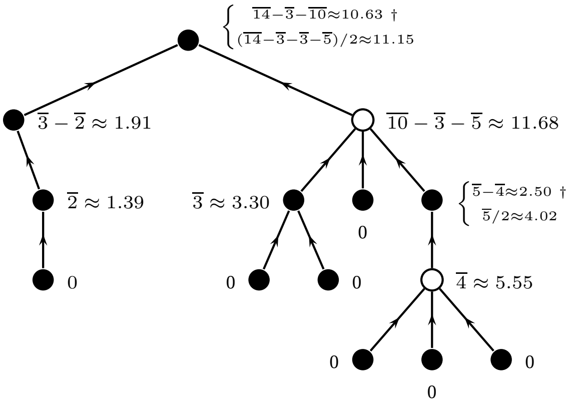

$\overline{15} - \overline{4} - \overline{10} \approx 12.05$
$\overline{4} - \overline{2} \approx 4.16$
$\overline{10} - \overline{3} - \overline{5} \approx 11.68$
0
$\overline{2} \approx 1.39$
$\overline{3} \approx 3.30$
0
$\overline{5} - \overline{4} \approx 2.50$ †
$\overline{5}/2 \approx 4.02$
0
0
0
$\overline{4} \approx 5.55$
0
0
0

Figure 4. The algorithm for Λ.

The result is that all nodes except a_{231} are active. For this you have to consider the very last part of the algorithm (the "top-down" part). This can easily be done based on information listed. For example, $T^*(a_0) = \{a_1, a_2\}$ (since the α-value $[a_0, B] = \overline{15} - \overline{4} - \overline{10}$ when testing



$\overline{14}-\overline{3}-\overline{10}\approx 10.63$ †
$(\overline{14}-\overline{3}-\overline{3}-\overline{5})/2\approx 11.15$
$\overline{3}-\overline{2}\approx 1.91$
$\overline{10}-\overline{3}-\overline{5}\approx 11.68$
$\overline{2}\approx 1.39$
$\overline{3}\approx 3.30$
0
$\overline{5}-\overline{4}\approx 2.50$ †
$\overline{5}/2\approx 4.02$
0
0
0
$\overline{4}\approx 5.55$
0
0
0

Figure 5. The algorithm for $\Lambda\setminus\{a_{11}\}$.

the set $B = (a_0)^{\downarrow} \setminus \{a_0\}$ is indeed greater than the β-value $\max_{t\in T_B(a_0)}[t]_{\max} = \overline{10} - \overline{3} - \overline{5}$). The final part of the algorithm, the "top-down" part is in fact necessary as examples show that not all "black" nodes need be active – some of these nodes may have been "overshadowed", as we shall illustrate at the end of the section.

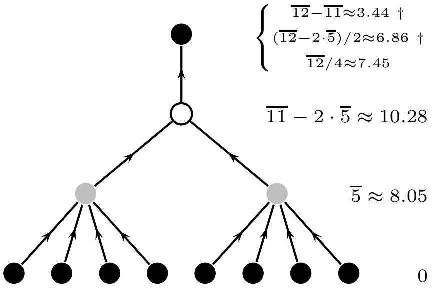
The final result is that all values listed in Figure 4, except the one which has been "daggered away" by a node higher up in the co-tree are the correct values of the relativized universal code. For the exceptional node we find that $\overline{\kappa}^*(a_{231}) = \overline{\kappa}^*(a_{23}) = \overline{5}/2$. The universal objects sought may then be obtained from Corollary 3.1. For this we need to calculate $\mathrm{R}_{\min}$. One finds that

$$\begin{aligned}\mathrm{R}_{\min} = \ln\Big(&8 + 2^{-2} + 3^{-3} + 2\cdot 5^{-5/2} + 4^{-3} + \\ &+ 3^3 2^{-10} 5^{-5} + 2^{18} 3^{-15} 5^{-5}\Big)\\ \approx{}& 2.12\,,\end{aligned}$$

measured in natural units, corresponding to 3.06 bits. This may be compared with the 3 bits necessary to encode the 8 minimal nodes which are equally probable under the universal predictor. In the expression above we have listed the contributions to $\mathrm{R}_{\min}$ (or rather to $e^{\mathrm{R}_{\min}}$) in descending order: First the contribution from the minimal nodes, then the node a_{12}, then the node a_{21}, then the two nodes a_{23}, a_{231}, then the node a_1, then the node a_2 and finally, the contribution from the top-node a_0.

In order to illustrate the sensitivity of the algorithm, consider also the co-tree $\Lambda^- = \Lambda\setminus\{a_{11}\}$. For this co-tree the algorithm gives a result which can conveniently be summarized in Figure 5. Again, no "overshadowing" takes place, but we note that a new inactive node pops up, the node a_2. Thus one cannot decide "locally" if a node is active or not. For this co-tree one finds $\mathrm{R}_{\min} \approx 2.01$ natural units ≈ 2.90 bits – compared to the approximately 2.81 bits needed to encode the 7 minimal nodes which have equal probabilities under the universal predictor.

Finally, concerning the phenomenon of overshadowing, Figure 6 shows the simplest example we can think of to illustrate this. For that co-tree, only the minimal and the maximal nodes are active and the 2 nodes in level 2 are "overshadowed".



$\overline{12}-\overline{11}\approx 3.44$ †
$(\overline{12}-2\cdot\overline{5})/2\approx 6.86$ †
$\overline{12}/4\approx 7.45$
$\overline{11}-2\cdot\overline{5}\approx 10.28$
$\overline{5}\approx 8.05$
0

Figure 6. A co-tree with "overshadowing".

VII. The case of co-trees with uniform branching

Consider a co-tree Λ of height n with uniform branching. Let $(k_1,\cdots,k_n)$ be the branching pattern. Denote by Λ_ν the set of all nodes in level ν. Put $K_\nu = |\Lambda_\nu|$ and, for a node $a\in\Lambda_\nu$, put $N_\nu = N(a)$ and $\overline{N}_\nu = \overline{N}(a)$. Clearly, $K_\nu = k_1\cdots k_\nu$, thus, recursively,

$$K_0 = 1,\quad K_\nu = k_\nu K_{\nu-1} \text{ for } \nu = 1,\cdots,n. \tag{36}$$

Regarding the convenient calculation of the N_ν's, see (12).

For the determination of $\tilde{\kappa}^*$, we shall specialize the algorithm of the previous section to the present situation of a co-tree with uniform branching. For reasons of symmetry – see also the discussion related to (33)-(35) – we need only work with certain special blocking sets. By $[\nu,\mu]$ we denote the bracket $[a,B]$ for a node $a\in\Lambda_\nu$ with the blocking set $B = a^{\downarrow}\cap\bigcup_{i\geq\mu}\Lambda_i$ for which then $T_B(a) = a^{\downarrow}\cap\Lambda_\mu$. These brackets are well-defined for points (ν,μ) with $0\leq\nu\leq n-1$ and $\nu+1\leq\mu\leq n$. We extend the definition by adding the point $(n,n+1)$. This point represents a minimal node and the empty blocking set. Therefore, we put $[n,n+1]=0$. For all other brackets we find that

$$\begin{aligned}[\nu,\mu] &= \frac{\overline{N}_\nu - k_{\nu+1}\cdots k_\mu\overline{N}_\mu}{N_\nu - k_{\nu+1}\cdots k_\mu N_\mu}\,. &(37)\\ &= \frac{K_\nu\overline{N}_\nu - K_\mu\overline{N}_\mu}{K_\nu N_\nu - K_\mu N_\mu}\,. &(38)\end{aligned}$$

The *bracket diagram* consists of all brackets. A numerical example is shown in Table 2.

Given ν, define $[\nu]_{\max}$ and τ_ν by

$$[\nu]_{\max} = \max_{\mu>\nu}[\nu,\mu]\,, \tag{39}$$

$$\tau_\nu = \operatorname*{Arg\,max}_{\mu>\nu}[\nu,\mu]\,. \tag{40}$$

Then, for a node $a\in\Lambda_\nu$, $T^*(a) = a^{\downarrow}\cap\Lambda_{\tau_\mu}$[3]. The numbers $[\nu]_{\max}$ are the *maximal brackets* and the τ_ν's are the *ceiling numbers*.

[3] to be sure, the *Argmax* in (40) has to be understood as the first index for which the maximum is reached, since we have not been able to exclude the possibility that the maximum is reached for several values of μ.

									0.00
8	12.62	12.69	12.76	12.84	9.55	5.76	6.59	5.55	8
7	18.51	18.77	19.04	19.32	13.19	5.97	8.68	7	
6	25.53	26.24	27.00	27.83	16.94	3.25	6		
5	81.21	91.91	106.17	126.14	85.39	5			
4	77.03	100.59	147.73	289.12	4				
3	6.33	6.33	6.33	3					
2	6.33	6.33	2						
1	6.34	1							
$^{\mu}/_{\nu}$	0								

Table 2. Bracket diagram for $\Lambda[1,1,1,4,5,1,2,3]$

The ceiling numbers can be determined directly from the bracket diagram. For instance, for $\Lambda[1,1,1,4,5,1,2,3]$, we find from the column in Table 2 with $\nu = 2$ that $\tau_2 = 4$ and that $[2]_{\max} \approx 147.73$. Then, by Theorem 5.1, the nodes in levels 0, 5, 7 and 8 are the active nodes. Further, the values of $\tilde{\kappa}^*$ for nodes in levels 0,1,2,3 and 4 is 81.21 and the values of $\tilde{\kappa}^*$ for nodes in levels 5,6,7 and 8 are, respectively 5.97, 5.97, 5.55 and 0.

Using the strategy as exemplified above for the calculation of $\tilde{\kappa}^*$, the full bracket diagram must be calculated and this amounts to about $n^2/2$ basic computations. This can be improved considerably by appeal to the algorithm developed in Section VI. For $\Lambda(1,1,1,4,5,1,2,3)$ one may for instance reduce the number of calculations of brackets from 36 (corresponding to Table 2) to 13 (will follow from results below). The basic facts we need are Propositions 6.1 and 6.2. The algorithm dictates that the bracket diagram is calculated for descending values of ν and ascending values of μ. To initialize, one sets $[n]_{\max} = 0$ and $\tau_n = n+1$. Then one calculates in succession $[n-1]_{\max}$ and τ_{n-1}, then $[n-2]_{\max}$ and τ_{n-2} and so on until $[0]_{\max}$ and τ_0 are calculated. On the way, the only tests that are performed are of the type "$[\nu,\mu] \geq [\mu,\tau_\mu]$?" and, in fact, not all these tests have to be performed as the result is bound to be negative (and hence $\tau_\nu > \mu$) in case, for a value of ξ with $\nu < \xi < \mu$, one has already found that $\tau_\xi > \mu$. This follows by Proposition 6.1.

In order to study more closely which tests can be neglected and which not, we introduce the abstract notion of a *Γ-diagram*. These diagrams are first discussed in their own right. After having developed a main property, Lemma 7.1 below, we return to the actual problem concerning co-trees.

Given are natural numbers $t_0, \cdots, t_n$ with $n \geq 1$ such that:

$$t_n = n+1\,, \tag{41}$$

$$\nu + 1 \leq t_\nu \leq n \text{ for all } 0 \leq \nu \leq n-1\,, \tag{42}$$

$$\text{if } \nu \leq \mu < t_\nu\,,\ \text{ then } t_\mu \leq t_\nu\,. \tag{43}$$

Then the *Γ-diagram* $G = G(t_0, \cdots, t_n)$ consists of all points (ν,μ) with $0 \leq \nu \leq n$ for which $\nu + 1 \leq \mu \leq t_\nu$. More precisely, G is a *Γ_n-diagrams* and n is the *height* of G. As a singular case we allow that $n = 0$. There is only one Γ_0-diagram, the *trivial diagram* consisting only of the point $(0,1)$.

By (43), if you consider the column from $(\nu,\nu+1)$ to (ν,t_ν) and place a horizontal bar on top of and to the right of (ν,t_ν) then you meet no points in G until you reach the diagonal element $(t_\nu, t_\nu + 1)$. Having the shape of the letter "Γ" in mind, this property accounts for the terminology "Γ-diagram". For a possibly more illuminating way of expressing the key property, see below.

A site $(\nu_0,\mu_0) \in G = G(t_0, \cdots, t_n)$ is a *test site* for G, if $\nu < n$ and $G(s_0, \cdots, s_n)$ is also a Γ_n-diagram where all the s_i are equal to t_i except s_ν which is set to μ. For example, all sites (ν,τ_ν) and $(\nu,\nu+1)$ with $\nu < n$ are test sites. For the Γ_8-diagram displayed in Figure 9, we have 17 test sites, corresponding to the marked positions. For a general Γ-diagram G, we denote by $\langle G\rangle$ the number of test sites.

Two operations on Γ-diagrams are worth pointing out: The *restriction* of $G(t_0, \cdots, t_n)$ to $\{\nu, \cdots, n\}$ is the $\Gamma_{n-\nu}$-diagram $G(t_\nu - \nu, \cdots, t_n - \nu)$ and the *direct sum* of the two Γ-diagrams $G(t_0, \cdots, t_n)$ and $G(s_0, \cdots, s_m)$ is the Γ_{n+m}-diagram $G(t_0, \cdots, t_{n-1}, s_0 + n, \cdots, s_m + n)$. Figure 8 provides an example of a direct sum.

For a Γ_n-structure $G = G(t_0, \cdots, t_n)$ we define the *spectral levels* $\sigma_0, \cdots, \sigma_\gamma$ by $\sigma_0 = 0$, $\sigma_i = t_{\sigma_{i-1}}$ for all values of $i \geq 1$ until you reach the index γ with $\sigma_\gamma = n$. We call $\gamma = \gamma(G)$ the *spectral index* of G. The spectral index of the trivial Γ-structure is 0, all other Γ-structures have positive spectral indices.

Note that the spectral levels $\sigma_0, \cdots, \sigma_\gamma$ can be constructed geometrically as indicated in Figure 10 by "letting the sun shine from the left" and noting the column numbers of the sunlit columns. The spectral index $\gamma(G)$ is the number of sunlit columns minus 1. Using the "sunshine terminology" we can also express the essential Γ-structure, formally given by the requirement (43), by saying that when the sun illuminates part of a column, it illuminates the entire column. And this property must also hold for restrictions of the Γ-diagram.

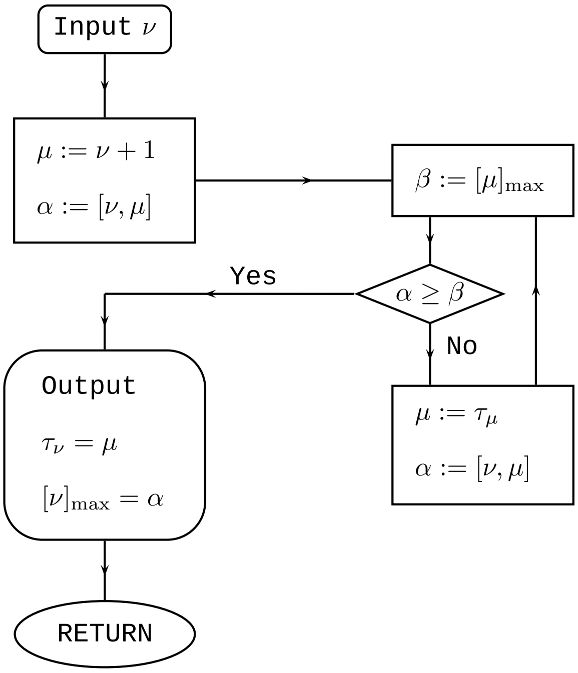


Figure 7. The central subroutine for co-trees with uniform branching

The combinatorial result we need is the following:

Lemma 7.1 (Γ-structure): For any Γ_n-structure G, $\langle G\rangle = 2n - \gamma(G)$, in particular, $\langle G\rangle \leq 2n$.

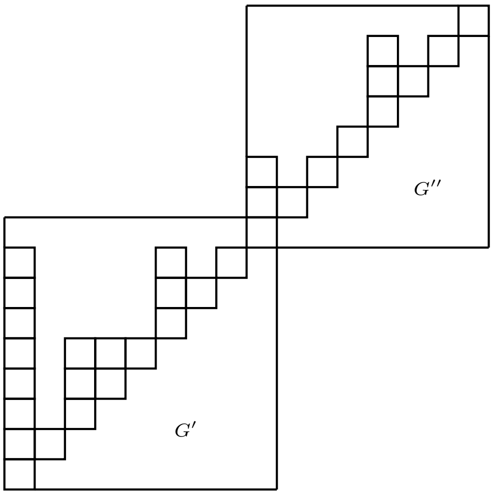


Figure 8. Illustration of last part of the proof of Lemma 7.1

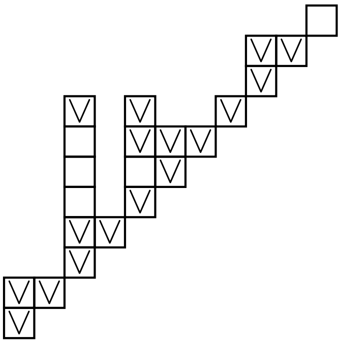


Figure 9. A Γ_{10}-diagram with test sites

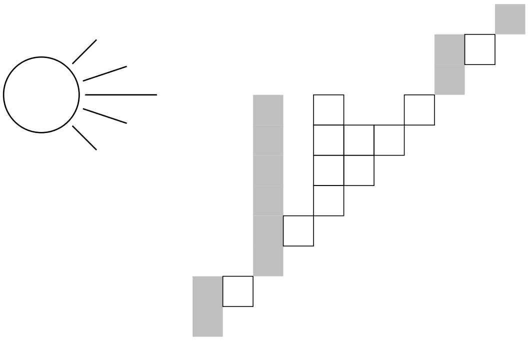


Figure 10. Sunlit culumns for diagram in Figure 9

Proof: The proof is by induction on the spectral index. To start the induction we have to prove the implication $G \in \Gamma_n$, $\gamma(G) = 1 \Rightarrow \langle G\rangle = 2n-1$. This is proved by induction on n. The induction start is easy. Then assume that the implication holds for indices smaller than n. Let $G \in \Gamma_n$ satisfy $\gamma(G) = 1$, i.e. $(0, n) \in G$. Consider

$$G^* = G \setminus \{(n, n+1)\} \setminus \{(\nu, n) \mid 0 \le \nu \le n-2\}.$$

Then $G^* \in \Gamma_{n-1}$ and $\gamma(G^*) = 1$. Thus $\langle G^*\rangle = 2n-3$ by the induction hypothesis.

In order to compute $\langle G\rangle$ and $\langle G^*\rangle$, first remark that for a point (ν, μ) with $\mu \le n-2$, the equivalence $(\nu, \mu) \in G \Leftrightarrow (\nu, \mu) \in G^*$ holds and the point is a test site for G if and only if it is a test site for G^*. It remains to consider points (ν, μ) with $\mu = n$ or $\mu = n-1$. Let $\{\nu < n-1 | t_\nu = n\} = \{r_1, \cdots, r_k\}$ with $r_1 < \cdots < r_k$. Then $k \ge 1$ and $r_1 = 0$. Likewise, let $\{\nu < n-2 | t_\nu = n-1\} = \{s_1, \cdots, s_l\}$ with $s_1 < \cdots < s_l$. Here, $l = 0$ may happen corresponding to the case with no sites of the form requested. Note that by the Γ-structure of G, $s_1 > r_k$. All $k+1$ sites $(\nu, \mu) \in G$ with $\mu = n$ are test sites for G, whereas G^* only has one site with $\mu = n$ and this site $((n-1, n))$ is not a test site. Among the $k+l+1$ sites (ν, μ) with $\mu = n-1$ in G as well as in G^*, there are $1+l+1$ test sites in G (the sites $(r_k, n-1), (s_1, n-1), \cdots, (s_l, n-1)$ and $(n-2, n-1)$), whereas all these sites are test sites for G^*. It follows that there are $\big((k+1)+(l+2)\big) - \big(0+(k+l+1)\big) = 2$ more test sites in G than in G^*, hence $\langle G\rangle = 2n-1$ as desired.

We now go back to the main induction proof and assume that the claimed result holds for all Γ-structures with a spectral index less than some fixed number $\gamma \ge 2$. Consider a Γ-diagram $G = G(t_0, \cdots, t_n)$ with $\gamma(G) = \gamma$. Note that G is the direct sum of $G' = G(t_0, \cdots, t_{t_0}, t_0+1)$ and the restriction G'' of G to $\{t_0, \cdots, n\}$ as indicated in Figure 8. As $\gamma(G') = 1$, $\langle G'\rangle = 2t_0 - 1$ by the first part of the proof and by the induction hypothesis, $\langle G''\rangle = 2(n-t_0) - (\gamma - 1)$. Clearly, $\langle G\rangle = \langle G'\rangle + \langle G''\rangle$. Therefore, we find that $\langle G\rangle = 2n - \gamma(G)$. This is the desired result and the induction is complete. ∎

After our excursion into combinatorics we return to the study of a given co-tree $\Lambda = \Lambda(k_1, \cdots, k_n)$ with ceiling numbers $\tau_o, \cdots, \tau_n$. The Γ*-diagram associated with* Λ is the diagram $G = G(\tau_0, \cdots, \tau_n)$. That this is indeed a Γ-diagram follows from Proposition 6.1.[4]

The algorithm we shall discuss consists of three parts:

- initialization,
- construction of the Γ-diagram,
- determination of the spectral levels, final output.

The initialization consists of the calculation of the numbers N_ν, K_ν, $K_\nu N_\nu$ and $K_\nu \overline{N}_\nu$ for $\nu = 0, \cdots, n$. For this, the formulas (12), (36) and (38) are used. In total, $4n$ basic operations are needed for the calculations. You may also consider as part of the initialization the assigment of start values $\tau_n = n+1$ and $[n]_{\max} = 0$ for the next step in the algorithm.

The key part of the algorithm is the calculation of the Γ-diagram, i.e. the numbers τ_ν, as well as the calculation of the associated maximal brackets, the $[\nu]_{\max}$'s. This is achieved by successive calls of the *central subroutine*. Though basically the same as for general co-trees, there are essential simplifications as also indicated earlier. This is partly achieved by symmetry considerations, partly by our focus only on those sites in the Γ-diagram where we really have to make a test. The flow diagram is sketched in Figure 7. The subroutine is called for all ν, starting with the highest value, $n-1$, and ending with the value $\nu = 0$. When all these calls have been made you realize that you only have to calculate a bracket (following

[4] In passing, we conjecture that every Γ-diagram can arise in this way. To illustrate the conjecture, observe that there are 5 Γ_3-diagrams and these may be realized as Γ-diagrams associated with the co-trees with branching patterns, respectively $(1,1,1)$, $(1,1,2)$, $(1,2,3)$, $(2,1,2)$ and $(1,2,4)$ (regarding the last pattern, see also Figure 6).

(38)) and to perform a test corresponding to test sites of the Γ-diagram. Therefore, referring to Lemma 7.1, no more than $4n$ basic operations are involved in the calls of the subroutine (for this, a test "$\alpha \geq \beta$?" as well as a calculation $\alpha := [\nu, \mu]$ is counted as a basic operation).

The final part of the algorithm is a "top-down" determination of the output of the algorithm, understood to be the spectral levels $\sigma_0 = 0, \sigma_1 = \tau_{\sigma_0}, \cdots, \sigma_\gamma = n$ and the associated maximal brackets. We suggest that these data are listed in the form $(\sigma_0, [\sigma_0]_{\max})$ $(= (0, [0]_{\max})$, $(\sigma_1, [\sigma_1]_{\max}), \cdots, (\sigma_\gamma, [\sigma_\gamma]_{\max})$ $(= (n, 0))$. Each such pair is considered to involve only one basic operation, thus adding at most n such operations. If you also want to calculate $\mathrm{R}_{\min}$ as part of the final output, another n basic operations are needed.

Considering the above discussion, we have proved our last main result which may be summarized as follows:

Theorem 7.1: Consider a co-tree $\Lambda = \Lambda(k_1, \cdots, k_n)$. Apply the modification of the algorithm from Section VI as described above. Then the number of tests performed during execution of the algorithm is at most $2n$ and the essential complexity of the entire algorithm, understood as the number of basic operations needed to carry out initialization, determination of the Γ-diagram and the listing of all pairs of spectral levels and associated maximal brackets is at most $9n$.

VIII. Conclusions and final comments

The paper offers a reasonably complete study of algorithms for the precise determination of universal objects associated with the model of all distributions over a co-tree for which the implication $a < b \Rightarrow P(a) \geq P(b)$ holds. A relatively simple situation corresponds to the case when the universal predictor satisfies the implication above with strict inequality. The key result here is Theorem 2.1 with Corollary 2.2 as a natural extension of Ryabko's result from 1979.

However, the main results concern general co-trees. It appears convenient to introduce a notion of *relativization* applied to codes. This concept is believed to be of interest also outside the scope of the present paper.

Theorem 5.1 provides basic insight into the structure of the universal objects, but quite some extra work is involved before a reasonable algorithm, presented in Theorem 6.1 is in house. Only a crude estimate of the complexity of that algorithm is discussed. For the special case of co-trees with uniform branching this is much refined. The main result is Theorem 7.1. One may comment that the resulting algorithm is easy to implement, also on simple programmable pocket calculators.

It will be observed that the key to the results are purely combinatorial facts, isolated in the transitivity identities in Lemma 5.1 for the general algorithm and supplied with a count of test sites in Lemma 7.1 for the special case of co-trees with uniform branching.

Regarding Lemma 5.1, we state a natural generalization:

Lemma 8.1 (transitivity identity, general case): Let $k \geq 2$ and consider nodes $a_1, \cdots, a_k$ with $a_1 > \cdots > a_k$. Assume that B is a blocking set for a_1 and that $a_k \notin B$. Put $B_i = B \vee a_i$ for $i = 2, \cdots, k$ and $B_{k+1} = B$. Then

$$\sum_{i=2}^{k} |S_{B_i}(a_1)| \Big([a_{i-1}, B_i] - [a_i, B_{i+1}] \Big)$$
$$= |S_B(a_1)| \Big([a_1, B] - [a_k, B] \Big) .$$

This result displays the transitive structure more clearly than the identities (24)-(26). As to the proof, it can be accomplished by a natural induction argument. We leave the details to the reader.

As to Lemma 7.1 there may well be simpler, more direct proofs based on links to other combinatorial structures. As an indication of this we note that the number of Γ_n-structures is the *Catalan number* $\frac{1}{n+1}\binom{2n}{n}$, which appears in many other contexts of combinatorial analysis[5].

It is a curious feature of the technical analysis that the logarithmic function only appears rather sporadicly. Accordingly, other functions may be considered. Without going into details, this may result in computations of universal objects tied to other notions of entropy and divergence than the standard notions of pure Shannon theory.

Acknowledgments

Boris Ryabko introduced the second named author to the general problems involved and several useful discussions are acknowledged. The complex of problems have also been discussed with Peter Harremoës and a number of related results, but for other models, have been developed, though not yet published. Finally, we thank Jan Caesar for technical assistance.

References

[1] T. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[2] L. D. Davisson. Universal noiceless coding. *IEEE Trans. Inform. Theory*, 19:783–795, 1973.

[3] B. Fitingof. Coding in the case of unknown and changing message statistics. *Probl. Inform. Transmission*, 2(2):3–11, 1966. in Russian.

[4] N. Merhav and M. Feder. Universal prediction. *IEEE Trans. Inform. Theory*, 44(6):2124–2147, Oct. 1998.

[5] B. Ya. Ryabko. Encoding of a source with unknown but ordered probabilities. *Problems of Information Transmission*, 15:71–77, 1979.

[6] B. Ya. Ryabko. Comments on "a source matching approach to finding minimax codes". *IEEE Trans. Inform. Theory*, 27:780–781, 1981. Including also the ensuing Editor's Note.

[7] B.Ya. Ryabko and F. Topsøe. Universal coding for sources with partially ordered probabilities. In *Proceedings IEEE International Symposium on Information Theory*, Washington, June 2001. IEEE.

[8] F. Topsøe. An information theoretical identity and a problem involving capacity. *Studia Scientiarum Mathematicarum Hungarica*, 2:291–292, 1967.

[9] F. Topsøe. Basic concepts, identities and inequalities – the toolkit of information theory. *Entropy*, 3:162–190, 2001. http://www.unibas.ch/mdpi/entropy/ [ONLINE].

[10] F. Topsøe. Exact Prediction and Universal Coding for Trees. In *Proceedings ISIT 2001*, Washington, June 2001. IEEE.

[5] the formula may be proved by noting that the sought numbers satisfy the recursion relation $\alpha_n = \sum_{\nu=1}^{n} \alpha_{\nu-1}\alpha_{n-\nu}$.