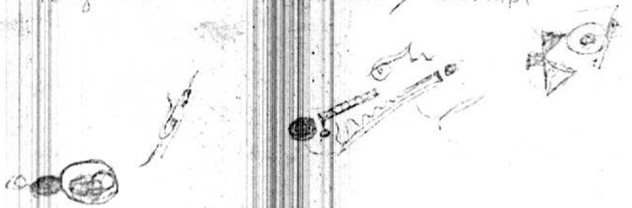
Blums 40 - 372

$\underbrace{40 - 372}_{73TM}$

Feb 28, 73 TM C

M. Blum: Prediction of deterministic seqs. Use of computers as a ~~[illegible]~~ Grammar; length of desc. relatively unimpl

Tech sq: | Al Myer's office on 8th floor
Talk (5th floor) 2P-3P
lunch ~ 12-1230



Eh) Given ~~[illegible]~~ 1 Dist of finite length, N. how could $[\xi_{m}]_{T}$ → 1 correct prob'ty as $T \to \infty$? Say it effective nature of t. rule changed at t. k^{th} symbol. $N \leq$ [illegible]

Its just that $\lim_{T\to\infty} \sum_i (P_i - P_i')^2 < K$ Decent: [illegible] K [illegible] down [illegible]

Staff
Gold (Grammars
J. Feldman
E. Ananisozi

L & M. Blum: (Deterministic sequence extrapoln

List pairs for seq. extrapoln.

$\varphi_i(x) = y$ ||| $i = M(y)$.

φ_i are Partial recursive functs.

$f = \varphi_i$ (i is t. index for f)

M identifys partial function f when i

$\mathcal{C}_M$ = set of functs that are identifyable.

$\mathcal{C}$ is identifyable if $\exists M \ni \mathcal{C} \subseteq \mathcal{C}_M$.

Some involvement w. complexity.

Have with machine guesses in Algorithm after [illegible] sample.

"Identify in t. limit".

I think t. idea is something like this: Say one is gn. t. fact that sequence S is in a certain set [S]. One knows [S]. If one has solved t. problem for set [S], then give any member of [S], S_x, say one knows t. first N symbols of S_x, one then guesses G(N), which of [S] this is. G(N) will change with N, but will remain constant after some bounded value of N. G(N) for these eventual large N will be t. correct identifn

I'm not quite sure as to just how relevant it is to CMI. B find t. desc. lengths irrelevant. (This, to some extent computation time tends to ↑ with input desc. (length)) Tho superficially, it would seem to be a special case of CMI, (i.e. deterministic seqs), the way they have set up t. problem — I'm not so sure how relevant it is.

On t. other hand, if it is relevant, it would be nice to be able use more kinds of seqs that have shorter computation times.

One big difnce betw. CMI & Blum's: CMI is concerned w. t. best prbly values obtainable for a finite seq. extrapoln. B is concerned w. obtaining an identific. algorithm that eventually will descr. t. sequence for large enuf seqs.

In both cases, t. problem is solvable only, if t. set of seqs to be extrapolated are limited in some way.

.25 As a side note: Al Meyer & G. Chaitin & others are working on things like thrms that state that using a gn. set of postulates, it will take a proof of at least N steps to prove that Grammar A is equivl. to Grammar B. (say A & B are given FS Gramms). "The length of a proof" corresponds to t. length of input to a UMC $N \equiv N_A + N_B$, where $N_{(A/B)}$ is t. length of Grammar (A/B)

SN) Sort of! In CMI, I constrain ~~...~~ t. ~~...~~ set of poss. TMs by ① Finite desc. ② ~~Computation~~ CB for all symbols < a known f(n). Perhaps if difnt constraints were usd, I would end up w. a signifly difnt soln. — perhaps more like Blum's. ~~Bt~~ Tho its hard for me to think of other kinds of constraints, Blum may be using other kinds based on recursive funct. theory

Mar 20, 73

<u>Blums</u>

TM

.01: $\boxed{44.90:}$ O.K.: Perhaps Blums idea: ~~that~~ Given a set of seqs, [S] → t. Time needed to compute t. N th ~~element~~ symbol (say one is given an integer, ≥ — we have a method to compute successive ~~elements~~ symbols in t. seq) is < F(N) ~~...~~ (F(N) is a monotone function)

.05 F(N) defines t. set of seqs. considered.

Say we are given / an s_i element of [S] — How many ~~elements~~ symbols of t. beginning of s_i do we have to look at before we know what element it is?

In t. present case, say we have looked at ~~...~~ t. first n ~~elements~~ symbols. We can then list all elements of [S] that have compu. time < F(N), ≥ then list t. ones that accord w. info thus far / say there are g_n such elements of [S]. As n ↑ we can discard members of this list many (but never add members). T. Q then is, for what forms of F(N) will we eventually end up w. only 1 member in t. set? (It is clear that t. no. in t. set must → a limit as n → ∞ since t. no. in t. set is a monotone not ↑, bounded below by 1.)

I think they may also consider other means (i.e. not .01–.05) of defining [S] — perhaps ending up w. a difnt. result. From t. abstract of t. talk, I think they

→ Hvr., if $\lim_{n\to\infty} g(n) > 1$, ~~...~~ — say it is 3, then this means that / those 3 members of [S] are identical. We may stipulate that this is impossl. by definition, — but if we have 3 difnt defns of ~~...~~ elements of [S], we can't always know that they are t. same forever. — i.e. they might differ for large enough n, but there is possibly no (effectively computable way to tell.

→ Considered all possl. ways of defining [S] — then they found for which [S]'s there did exist a soln. But say way it is clear from t. proof that for t. defn. of [S] of .01–.05 for any monoton ↑ F(N), a soln. exists — i.e. eventually the remaining possl. members of [S] will be t. same on t. next symbol, & this will continue to ∞, since there can only be a finite no. of disagreements.

Name Youre... time but ... would ...

.01 Blum

As applied to CMI: Consider the set of FDSS's that satisfy the constraints it takes $\leq F(N)$ steps ($\equiv T$) to compute the proby distrib of the N^{th} symbol to, say 10 digits accuracy (i.e. 1 part in 10^{10} accuracy).

.04

Then, for any N, there will be a finite no. of FDSS's that satisfy .01 – .04, say this is $\beta(N)$. As in 52.01 – .40 $\beta(N)$ is a monot funct of N, so it must $\rightarrow$ a constant as $N \rightarrow \infty$. So, for every function $F(N)$, there is an associated integer (probly not effectively computable) $\beta_\infty(F(N))$ (β_∞ is a map from functs, $F(N)$, onto integers) $\geq \beta_\infty = F(1)$ (! if $F(N)$ is monot)

As $N \rightarrow \infty$, we have a finite (tho probly large) no. of FDSS's to consider. β_∞ = Min. value of F

Say we have a sample sequence Σ, that is the Markov output of one of these finite set. Then for sufficiently large N, the correct FDSS will give least entropy to the sample (least of all FDSS's in the set), with proby $\rightarrow 1$ as $N \rightarrow \infty$.

.20

Note that there is no direct consideration of the length of descn" in the forgg. discn. There is some indirect consideration, because a descn of length L will take at least L steps to implement.

The way a.h. descns. are eliminated, is that for large N, an a.h. descn. of the sample seq. would not have "possibly done the win" for much smaller N. I think one can show that any a.h. descn. will eventually be elimd by this argt.

O.k.: 2 possl. difficulty w. .01 ff

1) Considering only fdss's for which $\leq T < F(N_0)$ for N_0 & for all $N < N_0$, constrains one to a finite set & $\beta(N) \rightarrow$ const as $N \rightarrow \infty$. Perhaps this part should be discarded. Allowing it means that "no back tracking" is possl. (I think)

constraint α

By eliminating this constraint, we allow $\beta(N)$ to $\rightarrow \infty$ as $N \rightarrow \infty$. Hvr, is it not true that ; a given $F(N)$ form & no constraint α is equivt. to constraint α but with a more rapidly rising $F(N)$?

— One can argue on either side!

Blum

2) Say we view 53.01 – .20 as an inductive method — More exactly, ~~we~~ for prediction of t. N+1 ($\underline{st}$) symbol:

a) we consider only those FSS's that satisfy t. constraint ($T \leq F(N)$)

b) of those, we use the single FSS having minimum ~~entrop~~ entropy for t. past ~~xx~~, to predict t. proby distribu for t. N+1 (st) symbol.

Using this, I have no idea on how large t. expected error in proby is — also, ~~xxx~~ while it certainly is ok for "large enuf" N, it is not clear that it is particularly good for small N. → See 56.01 – 56.37 for an estimate; 56.25 – .28 in particular

.15

I think both of these pts. may be impt. & should be gone into at some length.

.17 Well!! Perhaps t. last 2 pp. are all wrong! A dern. can be arbly long, yet take < T steps to ~~prnt~~ compute the 10th symbol of t. output. T. machine need not read t. entire dern. before it prints t. 10th output. This argument holds for t. problistic as well as deterministic seqs. → e.g. say t. dern. of a seq. contains t. sequence out to t. 10^{10} th symbol; t. machine need only read 10 symbols (of this) before it prints t. first 10 output symbols

In t. deterministic case, a completely A.H. dern. is possl. → See 133.01 – .18 for a genzn. of this objection. See 133.19 – 135.07 for a proof that its ok, anyway

Hvr, t. previous 2 pp did seem so nice & simple.

30 Is there any way I can resurrect some of it?? Yes! 55.10 →

Perhaps have another go at t. deterministic problem.

Blums may have t. additional constraint on t. set [S] — That t. members be recursively enumerable (= r.e.) This eliminates infinitely long purely A.H. seqs. — because they are not R.E.

If [S] is R.E., then a soln. always exists! Just pick t. first member. Match it w. successive symbols of t. sample until one mismatch occurs

Mar 20, 73 TM

Blum

Then go to t. next member of [S] & continue as befor. If t. current sample is t. i^{th} element of [S], then after finding $\leq$ i-1 miss matches, we will find t. ith seq. & have no more miss matches.

O.k., so just which non r.e. sets can we solve? — also, how does computation time enter?

.10 :54.30 A new way to define both t. deterministic & stochastic probs. so that t. discn. of ≈ 52.01 – 53.90 is O.k.! The deterministic seq. set is defined in t. folg. way: we have a finite string σ_i, punctuation symbol, Δ, & a binary no. N, followed by punctuation Δ again. When $\sigma_i \Delta N \Delta$ is fed in to t. machine, t. output of t. machine is t. N^{th} ~~member~~ symbol of sequence S_i. σ_i is defined to be t. "dcm" of S_i. (125)

In t. corresponding thing for stochastic/binary sequences when $\sigma_i \Delta N \Delta$ is inserted into t. machine, t. output is t. binary expansion of t. prob. ~~(pb.)~~ that t. N^{th} symbol will be 1 (to some stated accuracy). → (133.01)

25 **Note!** If we adopt t. convention $N \Delta \sigma_i \Delta$ as input Then we're in trouble again!

Perhaps Blum's method requires more total search effort than a search starting w. hy. post hypotheses first

Blum

:54.15 1) I like this Fdss prediction scheme that correspond to some extent to (Blum's for deterministic seqs).

T. Q. is: How bad can it be expected to be wrt.
1. CMI "optimum" method, for a gen. ~~seq~~ corpus length.?
A sort of order of magnitude estimate!

Say we are trying to extrapolate a ~~seq~~ corpus. We consider only ~~dn~~ dcns of fdds's ~~w~~ of C.B. (say no. of steps) > F(N), where N is corpus length. Say we use 55.10 ~.25! ~~Ou~~ This gives us t. limitation that no dcons of length > k (k ≡ G(F(N)) functional). So we need consider only 2^k possl. Fdss's.

[Enote; k = [redacted] F(1) if F(N) is a nond funct.]

Say t. true ~~de~~ dcn. of t. corpus is of length L. Then there will be k − L bits of ph'ness available in dcns of length k (k ≥ L — i.e. we assume that F is such that t. probys of t. corpus members are all computable within t. CB.)
The "correct" dcn. of t. corpus has L bits of corpus before it has to begin to start converging.

.25 I suspect that dcns of length K have an extra k−L bits of corpus, before they have to start converging. So that [redacted] "Blum's method" gives total sq. error ≤ k, while CMI gives ≤ L.

.26 ~~We~~ I suspect that in most cases L >> k → In fact, one doesn't know how big L is, so one makes k >> L — perhaps k ≥ 10L

Note! Say "Blum's method" is defined to be; using the single Fdss w. C.B. < F(N) ∋ t. entropy of t. seq. is minimal (Maximum probby). The nature of this Fdss will change as N↑, but will settle down eventually.

.37

T. consequences of using k > L is t. same as using a 20 parameter ~~curv~~ polynomial to ~~fit~~ the "curve fit" for a bunch of data pts in which a st. line is "correct". T. more params one uses, t. more H.H. t. curve becomes, t. better it fits pastdata, & t. less likely it is to fit future data. Hvr, if one fixes on 20 params, Then, for a large ~~enuf~~ enuf sample, everything gets O.K. again

Ap 19, 73

1:56.40 Blum. Disn. w. Minsky! He said ~~he~~ That he might have understood Blum:

M's idea: That one essentially gives a ~~—~~ formula (known to converge) that gives a descr. of the i th member of the <set of acceptable sequences>

So, one starts w. t. first member s_1 & ~~spends ... seconds on it~~ one ~~then~~ computes successive ~~digits~~ symbols of it & tries to matches them to t. corpus symbols. If a match ~~or mismatch~~ occures/ before To seconds one ~~...~~ starts to compute t. next symbol. If a mismatch occures, one goes onto t. next member of t. seq. S_2, & starts matching its symbols w. t. corpus. If one has spent > To on/ any one symbol computing of t. S_1, one goes onto & start work on S_2, but one still continues work on S_1.

So, as time goes on, one is computing on a larger & larger set of sequences.

As a prediction of t. next sequence, one always gives, at any particular time, the S_i of smallest i, that has converged at that time.

At any one time, one is computing on an enormous no. of sequences.

As soon as a/ successful next symbol is predicted this way, one stops (temporarily) computing on t. members, S_j, of $j > i$.

~~It~~ Here, the rules for deciding when to ~~...~~ go up in t. seq. & which to/ temporarily stop work on, are ~~...~~ random, so far, arbitrarily chosen. — They should be chosen more carefully.

~~...~~ After they are, one ends up w. a set of ~~...~~ [S]'s that are solvable in this way.

So
Ap 14.73
Blum
(55.25 spac)

01: 132.90 : I had been considering sets, [S] ∋ there was a C.B. F(N) on t. compn. of t. N^{th} symbol. This was done in such a way ∋ the set of ~~accepta~~ s_i's that were acceptable upto N symbols always included t. set of s_i's that were acceptable upto N+1 (say) symbols. ~~[illegible]~~ This was very impt. in my analysis of 53.01 – 54.15. In 54.17, I decided it could be wrong. I had in mind a more genl. way to define t. CB ~~[illegible]~~ or restrictions on [S].

10 One ~~way~~ generalish way to do this, is that it should take a total of < F(N) steps to compute t. first N symbols of t. sequence. This means that if a sequence is discarded ~~[illegible]~~ because it exceeds t. C.B. for N = 10, say, it still might be acceptable

18 for N = ~~[illegible]~~ 11

19 Hwr, if F(N) is given, one can still do about t. same as 53.01 – 54.15 in t. foll. way: To predict the N_0^{th} symbol of t. corpus, pick the s_i of smallest i that ~~[illegible]~~ satisfies F ~~[illegible]~~ for ~~[illegible]~~ All $N < N_0$ & also does correct predns for all $N < N_0$.

Say i_0 is t. index of t. correct sequence. Then, in one's choices, one's i will sometimes ↑, sometimes ↓, but it will never get > i_0 — so one is considering a finite no. of sequences. Say $[S]^\alpha(N_0, i_0)$ is the ~~number~~ set of sequences that (a) have CB < F(N) for $N < N_0$ (b) have their $i \le i_0$ (c) Agree w. t. corpus upto N_0. ~~Clearly $G(N_0, i_0) < i_0$ & is a ~~ ~~monot funct of N_0.~~ Let $[S]^\beta$ be t. set that satisfies only conds. (b) & (c). (so $[S]^\alpha \subset [S]^\beta$). ~~The~~ The number of elements in $[S]^\beta(N_0, i_0)$ is $\le i_0$ & is a monot funct of N_0 — so this number must → a limit as $N_0 \to \infty$. So, if one picked any ~~random~~ member of $[S]^\beta$ for predicting t. N^{th} symbol, one would eventually be always correct.

ref Because $\lim_{N_0 \to \infty} \#[S]^\beta(N_0, i_0) = C$ (# = no. of elements in), & C is an integer, then there must be some N' ∋ $[S]^\beta(N_0', i_0) = C$ (since we can get within $\frac{1}{4}$ (say) of C, by making N_0 large enough.

Sat. Ap 14, 73 TM 134
Blum

So, for $[S]^\beta$, for $N_0 \geq N'$, any member of $[S]^\beta$ will give perfect prediction.

Since $[S]^\alpha$ is a subset of $[S]^\beta$ containing at least 1 member, (S_{i_0}), Then any member of $[S]^\alpha$ a-fortiori also must give

.07 perfect predn. for $N_0 \geq N'$.

.08 The foreg (133.19 ff) constructive proof works if The set [S] is defind in t. folg. way:

(a) A formula is given for the i^{th} description of t. i^{th} member of [S]. This descn. need not converge for all i for all symbols of t. seq.

F(N) is gn., so that any sequence of [S] must be derived from formula (a) & satisfy F(N) (= an upper bound on t. ccost of t. first N symbols of computing that sequence.

I don't know if t. defn. of .08 ff is "recursively enumerable" — i.e. I can think of no formula for t. k^{th} acceptable sequence. — In fact, I can't think of a way to find any members of t. Sequence!

.25 Anyway: Is t. method of soln. of (133.19 – 134.07) applicable to probabilistic sequences? Say we are given F(N) (133.10 – .18). We can't select t. P.E.M. of lowest i that satisfies t. C.B., that "works" for t. corpus because usually all PEM's will "work" to some extent.

Furthermore, since I think there may be an ∞ of P.E.M's that satisfy t. C.B., we cannot usually ask for t. one that

.30 fits t. corpus "best."

My 15, 73: General Conclusion: (Perhaps write This to Blum): B's prob. is "what kind of constraint can one put on a set of seqs, ∋ ∃ a soln. to t. BW predn problem (of t. type he defines). ?". For certain kinds of constraints t. soln. is trivial. For others, more diflt. If I put analogous constraints on my (Gray) induction problem, ~~some~~ of them result in a ~~trivial soln. (i.e. t. shortest code that tracks t. known corpus)~~ reasonable soln. (e.g. say constraint is that ccost ≤ seq. length). For weaker constraints, B's prob may have a soln., but for analogous constraints, my prob. may have no soln. .25 – .30

∞
C

P. Kugel
(221.01)

July 7, 73 Blum.

Bldg. (35-410) TM cr. Mass & Vassar. 2

1.139.90 Phone talk w. Peter Kugel: work: MIT: 253-2165 (to 5 pm) 969-2339 (to 10 pm)

Advisors: H Putnam (Ma... Fame) M Gold (west coast language)

Peter wrote (PhD Thesis at Harvard) on Induction. The problem he defined was like M Blum's i.e. no probys involved: He says there are diffnt degrees of incomputability — So his solns. may differ from that of Blum's, who was interested only in effective computability I think.

P would probly like me to read his Thesis. He feels it is probably related to my work on induction.

At the present time I don't want to read it. ~~[illegible]~~ I may, however, want to browse thru it to try to get t. genl. ideas — perhaps some might be useful.

① Get Blums' records from Kugel

Discn w. ~~Al Meyer~~: O(f(x)).

perhaps defn on P28 of Blum's 13 Mar 73 report

This is very approxly what he said. → An operator (functs to functs) is "O-honest" if there exists a machine that can compute f(x) in

.21 a time that is somehow "about" O(f(x)). → → 217.06

Say one has several ~~appx~~ functions that agree w. t. corpus up to now. Then one will select t. one whose O-honest operator value is best, & continue w. that one. One switches from one to another funct. (as long as they are consi. w. corpus) on t. basis of how long their compns. are taking.

Anyway, the focus of t. method is on finding a funct seq. that takes a minimal amt. of time to compute & yet is consi w. t. corpus.

T. relevance to CMI: CMI can be regarded as taking all BW predictors ~~fr a~~ ~~ker~~ of t. next symbol of t. corpus & combinig them linearly w. wts ∝ $2^{-\text{Dcr length of that "predictor"}}$. So, if we can find all t. "predictors" that are consi — that take a min amt of

July 6, 73 TM 227

Blum

01: 221.40 time to compute, we can just wt. them w. $2^{-\text{Desc length}}$ & get our predn.

Hvr., I'm not at all certain of what's going on here. At best, I suspect we'll have t. difty of 212.18 that we had w. using SQPM's in a direct way for predn.

221.21: It mite be something like this: That we are interested in seqs. in which t. time to compute t. nth symbol is bounded by some funct. of n that doesn't grow too fast.

This would be close to my using CMI w. t. aux condition on t. fdss that t. CB be $< f(n)$ where n is t. ~~order of t~~ corpus length & $f(n)$ is some ↑ funct. (like ~~an~~ or 2^n).

22 As n ↑, Blum makes t. criterion for switching to a difrnt function (≡ hypothesis seq.) more
24 difficult to satisfy (i.e. ~~Hyarshyar~~ hysteresis) (i.e. unlikely). B's criterion for soln, is that eventually, one does not change one's choice of function. An alternate criterion, that he doesn't use is that one eventually doesn't change one's choice of "next symbol".

Hvr., using t. criterion that he uses and .22–.24, one gets a larger ~~crackled~~ class of funcs. over which solns. exist.

Some apparent differences betw. ~~me~~ Blum's & my work:
I am concernd w. practical induction. So I'm concernd w.
(a) How much computation is necy to do t. induction (b) How good an induction one can do w. a gn. SSZ. (i.e. optimum induction w. min. seq. length).
(c) Gray v.s. B.W. induction is again a much more practical problem.

1227.40

.01 Out. problem of "How can one limit t. set of possl. pams & still have a soln to t. induction prob?"

First, CMI only ~~considers~~ proves anything about finitely decidable pams.

pat For pams that can be approximated in a certain way by fd pams, it also works. All pams of science thus far have been of this latter sort.

If we limit our pams to those whose CB is bounded by some funct of S ($S \equiv$ t. corpus length) — Then we do have a soln. — in t. sense of ms error $\to 0$ — ~~or~~ even Σ ms error $< \infty$

If we are limited to fd pams and we are gn. a machine, & if we insert any fd pam into it, t. machine will eventually tell us if that pam is in t. possl. set or not — Then we have a soln. also: ~~To go~~ ~~To so~~

.20 T. Soln: for corpus of length S, examine all fd pams of length S or less via this machine. Use t. "all pams method" on all acceptable pams of desc length $\leq S$. As $S \to \infty$, we will eventually include t. correct pam.

Or, use $A + F(s)$ to chose t. best pam ~~of~~ desc length $\leq S$, & use this pam for t. proby distribn ~~extrapoln~~ of t. next symbol. I've haven't proved this will work, but it probly will — its a weak version t. SVHM. Any stronger versions could also be used.

.30 If one is gn. ① a machine that enumerates all acceptable pams. ② " " " " " un " " , → Then one can, from those 2 get a machine that tells us if any particular pam is or is not acceptable — so → .20 for a soln.

To do this, Have machine 1 present t. i^{th} pam. then " " 2 " " " nonpam → i+1, etc. Continue until either machine gives a decision on t. pam in Question.

I do not know if a soln. to .01 exists if one is gn. Machine ① (.30) only. i.e. (if t. set of acceptable

Oct 14, 73 Blum TM 355 2

pams is only known to be r.e. r.e. — (solved! → .25)
It is not nearly solvable

Hvr., T. problem of .354.01 is only important to t. extent that it may shed lite on t. other & impt. probs of ① T. probs in phil. of sci & sci. method ② ~~Practical induction~~.

At t. present time, I think that t. solns. that I do have, ~~for~~ for .01, are good enuf for good usable suggestions for ① & ②

Incidently, I think t. set of non fd pams that CMI will work for are those (I think Willis discusses this) for which one can obtain an arbly good Fd pam. — i.e. say we have Pam, (non fd pam) — then Pam, is of ~~a~~ a type where CM I will work, if, for every ∈ (small) ∃ a Fd pam, Pam₂(∈) ∋ ~~B~~ Pam, & Pam₂(∈) agree on all proby estimates within ∈ (or their log probys sum within ∈). — (This "all" means for all possl corpi — (or conceivably, on a set of corpi of measure 1).
finitely derivable?

.25 If t. pams are known to be r.e. then this is not a sufft. cond. for a soln. to exist — because t. fd pams are r.e. — i.e. partially order them r wt C.B. — then order them further by lexigraphical order of derns. (shortest dern first) — & t. fd pams certainly are not a solvable set.

Since a r.e. set does work for Blums' problem one wonders why it wouldn't work for t. "proy" proby problem. Hvr., I did go over this (see disc. of "Blum" Blum Bin index) & ~~Blum's~~ a simplified form of Blum's method will not work for t. Gray proby problem.
That works for an r.e. set of ~~pams~~ deterministic strings

Blum

.01 : Re: This method of of enumerating all finitely describable (≡ F.D. pems) PEMS: I don't see that it would work that way — if at all. Say one starts w. a C.B. of 1 & considers all pems within (≡ <) that C.B. One can't do it so easily, because t. C.B. vary w. L, t. length of corpus.

10 One could get t. finite no. of pems w. C.B. < C, w. length = L — one could then have C, L scan thru

12 t. L, C plane — but still, for each pem, one doesn't

13 know if it will ~~[illegible]~~ ever converge for all lengths > L.

So, this counter example may be invalid, & it may well be, that if i had a ~~[illegible]~~ r.e. set of pems, I could ~~[illegible]~~ do arbly. good predn. for long enuf L.

.21 Consider t. folg. method: Say Pm_i is t. ith pem in this R.E. set. All members of t. set eventually converge for all L. To ~~produce~~ get t. proby distribn. for t. l's symbol of t. corpus, use t. all $i$pem's method on t. first l Pm_i's.

.25 ($i = 1 | l$).

If .21 – .25 is correct, it ~~[illegible]~~ is ~~[illegible]~~ gives t. same result as Blum in this case (i.e. a R.E. set can be predicted). Furthermore, t. proof of predictability is ≈ to a proof one could use on deterministic seqs.

On second thot, .21 — .25 may not ~~[illegible]~~ work all t. time. When one averages t. correct pem w. many others, I'm really not certain that one will nearly get very small error (or expected error) for large L.

— It may well be true, but I haven't proved it. Its a thrm vaguely related to t. SVH thrm(s)

Oct 20, 73
Blum

→ if 359.21 – .25 is true, then it can't be r.e.

01: 359.10 : On the Q. of whether the set of all fd pams is r.e. : ① If it is not r.e., it's not because there are "too many". – since ~~that~~ it is a proper subset of a ~~.~~ r.e. set — i.e. the set of all finite strings is the r.e. set. The set of fd pam dems is a proper subset of this.

② It's partly because one can't always tell ~~if~~ whether 2 pams are identical or not — but I think this is irrelevant — one doesn't care if 2 pams w. different dens are identical or not in CMI

③ One doesn't know, in general, if a gu. pam den. will converge for all ~~inputs~~ corpi of arby length.

Perhaps this, too, is irrelevant. One could go thru the routine of 359.10 – .13 — But I think I showed that in general, there is no such way of scanning that will work for all pams — i.e. the CB of a pam could always ↑ faster than any "scanning" scheme of C.B. V.S.L

⟨The set of all pams whose CB is bnd by T^2⟩ is this set R.E.? CMI certainly works for it.

Yes, it is r.e. — see ~~§~~ 361.01 – .20. T

Oct 22, 73 Blum TM 361

.01: 360.40

↳ T. set of Finite decrbl. Pcms may be r.e. ! ← only if t. CB. is < a known F(L)

O→ A umc; a c.B., K; a a non-printing input string to t. umc; defines a CPM.

Hvr., t. c.B. must be an ↑ funct. of L, t. corpus Length, or else we will give all our wt. to "U" says for large L.

But, t. trouble is, K(L) can never ↑ fast enough. For any fixed K(L), ∃ a FD Pcm that needs a CB that ↑ faster!

So, t. set of FD Pcms that have a CB bounded by a known F(L), are r.e. — but t. genl. set

.20 of FD Pcms is probly not r.e.

.21 O→ Thrm & proof: if a set of Pcms is r.e., CMI will converge in t. willis "sense. : Proof:

say Pm_i is t. i^th pcm. Let K(i, L) be t. largest CB needed to compute t. L'th symbols of Pm_j (j = 1 | i). Then, use t. normal willis method for the [redacted] L^th symbol, using CB = K(i, L).

Eventually, for large enuf L, ~~it~~ t. ~~proby~~ using K(i,L) as C.B., true Pcm will be "within t. CB" for all L > that one.

O→ Perhaps I can xfm any solu. of Blum's problem into a solu. of this more general problem! So far, I've done this for R.e. sets of Pcms. One Q is — How does Blum express other kinds of sets? (i.e. non-r.e?).

One way is: t. complement of a R.E. set. (which need not be r.e.).

Can I show that: If a soln. to t. "Grey"

Get passport

Stamps

Silicone Spray for Shoes

Thermometer(s)

Oct 21 small headache ~~...~~ ~~...~~ near r.E. eye

Oct 22; same

Oct 23 no (or uncertainty detectable) headache in morning

Snake bite kit.

Perhaps Scorpion bite fixer

Running board not dash board

Lookup: M. Goodall in "Bio proto t. a Synth. Systs. Vol I Barnard, Morley Klar Planum, 1962

Oct 23, 73 Blum

01:361.40: induction problem exists, then t. set of allowd pams must be R.E. ? (By soln., I mean something like, t. expected value of t. abs. value (or t. sq.) of t. error → 0 in ~~some~~ sense.

If no soln. exists t. set must be non r.e.
Woops! This is implied by R.e. set ⊃ soln. exists.

Or show: if set of pams is not r.e., then there is no soln.

∃ soln ⟷ set is R.E.

If A ⟷ B then does ~A ⟷ ~B ?

B ⊃ A
~A ⊃ ~B
∃ soln ⊃ set R.e.
~set r.e ⊃ no soln.

Sounds not so reasonable! i.e. perhaps there are some small non-r.e. sets that are easy to predict!

Some Very Good Looking Ideas!

1) Blums don't deal w.t. problem of deterministic seq. ~~prediction,~~ extrapoln but w.t. problem of function extrapoln: i.e. up. a seq. of $(x, f(x))$ pairs, to guess what $f(x)$ is, eventally.

2) T. Blums' problem of ~~their~~ determst. seq. extrapolation is – what kinds of sets of seqs. are extrapolable "in t. limit"?

A soln. is: I Think it is possl. iif t. set is r.e.

.34 Proof (a) If t. set is r.e. then it is certainly possl. to extrapolate it "in t. limit".

.36 (b) to show that t. possy of extrapoln. "in t. limit" implies t. set is r.e.;
Consider t. set of binary seqs assoc. w. t. nos. 0, 1, 2, . . . ∞.
If each of these seqs. is gu. as a sample set., to t. extrapoln. machine, its choice will be one of t. members of t. allowd set. Since t. machine can detect any

Blum.

01: 362.40 member of t. set, ~~[...]~~ for each member of t. set
∃ at least one initial sub seq. of it ∋ t.
machine would chose that member at end of
that sub seq. So, this means that ~~for every~~
if we go thru t. nat. nos. as samples, t. machine
will eventual ~~chose any~~ chose every member of
t. set at least once. We can discard repeats
since we have an arby large many. — So this
means assoc. w. every nat. no., is a member of
.18 t. set & visa versa. → Hvr. see 365.01–.04 for why t. forgg. may be a waste of time ~

---

3) ~~[...]~~ We can use t. argts. of 2) a) b) (362.34 – 363.18) to show that its also true of Blums' problem for functions! Take Blums' finite sample seqs. of (x_i, $f(x_i)$) pairs. Assign a Gödel no. to each sample seq. ~~Each~~ Blums' technique maps each such Gödel no. into one member of t. set of all ~~[...]~~ legal functions. (This corresps to 362.36ff (≡ b)). The a) section (362.34ff) in t. present case is also trivial. ~~→~~ see 365.10 for why this is wrong!

Blum also uses the null "*" to indicate nil in t. domain. They say this is really to make possl. t. funct w. empty domain — but is there any other reason?

Anyway, we can include * in t. Gödel numbering ~~by~~ using t. folg. ~~[...]~~ trick: Say we have a no. seq. like x_1, $f(x_1)$, *, x_2, $f(x_2)$ . . . A seq. of 5 nos. for any x_i or $f(x_i)$ we assign t. no. x_i+1 or $f(x_i)+1$ for *, we assign 0, Then use Göd nos. for this new seq. of nos.

Blums' method doesn't make a guess for every ~~[...]~~ finite sample seq. — but this simply

Blum.

01:363.40 means that every integer (≡ Göd no.) doesn't map into a function. Hvr., every function certainly has ∞ Göd nos. assoc. w. it.

4) For my "Gray" induction problem: I've shown (361.21) that if a set of Pams is r.e., then CMI will "will is converge".

10 Now if a Method of induction converges in "any clearly defined sense", then the set of "acceptable Pams" has a set of r.e. pams, ∋ every member of the "acceptable set" has a member of t. r.e. set "arbly close to it". "Arbly close" means that either of t. pair would be said to "converge to the other" in whatever this "clearly defined sense" is.

To show .10 ff, consider t. sample seqs. of 362.36 ff. They map t. induction method into members of this set of Pams that "converges in t. "clearly defind sense"" to t. ~~t.~~ any member of t. "acceptable set".

Re: .10 ff: The things that converge to one another are t. Pams not t. seqs. or t. seqs of probys. Viewed ~~fr~~ in that way, it is clear that there is one pam that is "arbly close" to every member of t. "set of acceptable Pams" for all members of t. set — i.e. t. single pam that is t. Solution !

.35 (Note that here "closeness" is not transitive or is it even apprxly an equiv. relation).

A probably very trim on t. "Thrm" of .10 (if it is true at all) is that t. set of Pams for which it is not true be of measure (i.e. probty) zero.

Anyway, I think I want t. "decided on "Pam" & t. "actual Pam" to differ by a small amount in t. sense that ~~Bar Diff~~ t. expected value (wrt t. true pam) of say, t. → 366.01 spec

0.21.73

Blum

.01! 361.90! Hvr., ro: ~~a~~ Threm 362.36 : Since, (on P3 of t. Blum Report) i, is t. index of $\phi_i(x)$ — which is to be "an ~~a~~ extension of $f(x)$ — it seems clear that ~~a~~ t. set ϕ_i must be r.e., via t. index $\underline{i}$!! which is t. presented "soln. thus far";

Hvr, even if t. set of legal "f's" is not given by integers like t. i in ϕ_i — but is gn. by a finite dern to a UMC., these finite derns are orderable (lexicographically) — so they must be r.e.!

So perhaps I'm missing t. pt. of all of this!

.10 No! — Not at all! An argt. would say that all finite dervable fcns are R.e.! The problem is that for certain values of argt. ~~a~~ some of t. $\phi_i(x)$'s don't converge. Also, t. C.B. for computing $\phi_i(x)$ can be arbly large — i.e. for all C, $\exists i, x. \ni \phi_i(x)$ takes a CB > C.

.23 O.K. Take ~~St.~~ Blums' problem:

.24 Say t. tng. seq. is always of t. form: $(0, f(0), 1, f(1), 2, f(2) \ldots \ldots)$

.25 Lemma) For every ind inf. machine (in B's sense) $\exists$ a function $G(n)$ that increases so fast, that if a function $h(n)$ takes $\geq G(n)$ steps to compute, then this ind inf machine can't identify $h(n)$.

~~Thus we~~ $G(n)$ can be constructed as soon

.34 as we are gn. t. ind. inf mach. → to find $G(n)$ → 367.07

.35 If we have a set of functions ϕ_i ~~[illegible]~~ $\ni$

.36 they all take $\leq H(n)$ steps to compute $\phi_i(n)$, ~~[illegible]~~

& $H(n)$ is known, then ~~the set ϕ_i is r.e.~~ ~~[illegible]~~ t. set of functions can be identified. ~~[illegible]~~

.40 Use upto $H(n)$ to test $\phi_i(n)$. If it doesn't converge

square of their difference, is small & → 0. Those 2 pems differ by very little, even in t. immediate future, as opposed to t. single pem that's ~~&~~ close to "all" pems (364.35) – but this closeness exists only in t. far future.
—— So I may be getting at something real!

~~Then~~ On CMI again: About t. ~~E~~ r.e. idea ~~of E~~ 364.10ff, I doubt if r.e. is just what I want. That t. E A c.B., G(n) exist for all pems for corpus length n, is certainly a suffy. cond. → 370.06

Anyway, I'm beginning to see CMI & Blum's problem are very similar. Blums want to identify a funct that maps integers into integers. A big problem for them is Getting CB's. CMI looks for good PEMS — which are ~~maps~~ functions from integers (partial corpi) to integers (finite accuracy probability values) (the zeros at t. front of these probty values may give trouble. ~~E~~ One way to avoid this: The binary probty ".0001011" is ~~written~~ treated as t. integer: "1101000".

One can do a search over all possl. pems (≡ functions) in t. acceptable set —— Hvr., ~~a~~ t. output will (if t. ~~E~~ tested pem converges) not ~~neces~~ be a yes–no (as Blums' functs), but a Gorc value (say $(A_i + F_i(s)/s)$) One somehow settles

Bestof pem — proof of partial corpus (up to S) wrt that pem.

on a pem w. minimum Gorc & goes on to ~~E~~ a longer corpus.
—Or, one uses t. "All ipems method" after a large enuf no. of pems have been tested (say all pems with $A \leq S$).

Blcan

.01 : 365.40 or obtain the vile output, ~~say ϕ_{i+1} tried~~ then $i \doteq i+1$ ($i \leftarrow i+1$) & go to 365.40.

Thrm
→ So a set of functs is identifyable (under conds 365.23–.24) iif 365.35–.36 is true. (i.e. $\exists$ $H(n)$ exists for that set of functs)

.07 : 365.34 Given an induction machine, M, to find a $G(n)$ assoc. with it. The idea of "identifn" is that for suff'ly large n, M finally settels on a ϕ_i, and the C.B.'s for/successive computation of ~~guess~~ $\phi_i(n)$'s are all within the C.B. that M decides on. One way to get an idea of what $G(n)$ mite look like: Feed M a pair $(0, f(0))$. ~~Eg~~ ~~tell it that~~ ask it how long it wants to work on its first trial ϕ_i / with no final $\phi_i(0)$ output before going to another trial. ~~say this is~~ Call this $G(0)$. ~~how long it can work~~ Then let it get $\phi_j(0)$ correct. Then see how long it must wait w.o. output, befor ϕ_j is rejected for $\phi_j(1)$. — call this length of time $G(1)$ — Similarly, calculate $G(2) \ldots\ldots G(n)$. — Hvr. this isn't very good — we don't know just what M will do.

Eg Lets try to force M to oscillate betw. 2 ϕ's: ϕ_0 & ϕ_1.

→ Or Can we find a C.B. for $\phi_i(n)$ $\ni$ no matter how M got to / try to compute $\phi_i(n)$, its C.B. will be too large?

→ The contrary of 365.25–.34 is quite unreasonable! It is "$\exists$ an ind inf machine, M, $\ni$ no matter how fast $h(n) \uparrow$ with n ($h(n)$ is the max over the set of acceptable functs), M can identify f. → Spec 368.20

0.25.73

Blum:

01: ≈ 367.40 The other point is: If $G(n)$ bounds ~~~~ $H(n)$ over t. set S, of acceptable functs, then are t. functs of S ~~~~ r.e.? Certainly, for any n, there are only a finite no. of functs with $H(n) \leq G(n)$.

~~~~ Hvr, I don't see how one could find even one of t. functs in S, if one knew only $\vec{G}$ ~~well~~. One could list certain sets of functs that have known behavior at ∞ — but I can't think of any other way. AH! One can perhaps enumerate all functions for which a proof exists that its $H_i(n) \leq G(n)$ for all n. One simply lists all possl. proofs & checks them to see if they prove this. Eventually, if a proof exists, one will find it this way, but it's likely that $\exists$ many functs that satisfy this & no proof exists for it.

20: 367.40: T. contrary of 365.25 – .34 is not so strict. Perhaps we need: for any $G(n)$ $\exists$ at least one f_i ~~~~ in t. set $\ni$ $H_i(n) > G(n)$.

[illegible]
~~~~

0.26.73 Brookline 19 B. st. Brookline 876·9074 Latino TM369

Bloms'

01: 366.40: Another, probly better, way to do CMI is t. regular way — have we consider a fixd function (i.e. t. UMC) - its trial inputs are codes — t. expected "f(x)" is t. partial corpus.

Re: continuing in this direction — Re: Blums

1) I dont understand ~~Expos~~ example 2 (P.4)

2) P.6 Gold's Thrm! that no ind inf mach can identify all

1. rec. functs. — This seems contrary to 3.28-.29 (perhaps t. set of rec. functs is not r.e.!)

3) P 31.38-.40 suggs that their methods mite be, in some sense economical & useful for practical induction. In CMI, an analogous set of problems! Given various constraints on how CMI is to be implemented, whats t. best way to get max $\frac{\text{prost}}{\text{ccost}}$? → 372.07

18

4) P31.33-.37 may give a good idea as to what their method is.

Σ

Some impt. remarks on Blums' method

1) The ordering of t. functions tried is ϕ_i ($i=1/\infty$) since ϕ_i is effectively computable, this means that i is a code for ϕ_i — this gives an upper bnd on t. amt. of "info" or "pcost" of ϕ_i (i.e. $\log_2 i$). So in some sense, Blums' method (& perhaps any other conceivable method), ends up picking t. least complex (i.e. lowest i) ϕ_i that will a) fit t. data fug Sq b) fit under t. C.B's imposd by t. method.

2) A set of functions, ϕ_i & a seq. of argts in t. fug. seq. can be characterized by a seq. of vectors (one vector for each i) that gives t. ccost to compute t. output for t. N^{th} argument. Vector components can be ∞ $+\infty$.

~~Mmmm~~ A ind inf mach in Blums' sense, is a method to ~~test~~ select a next i value & decide what CB to use on it, based on experience thus far w. other ~~_____~~ (or t. same) i's a/o N values.

3) Perhaps any ~~a~~ Method to solve B's prob can be usd to do CMI — since this, too, involves ccosts of computations of ~~[illegible]~~ values of a fixd function, for various argts.

Blum

1: 369.40 1) A major Q is whether Blums' method ° perhaps all other methods can be each characterized by a function G(n) (368.07 ff)

A way to get something like G(n): Construct ϕ_0 in the following way, & let ϕ_0 be t. soln. to B's problem: Say t. tng seq is 1, f(1) \$; 2, f(2); 3 f(3) . . . .

T. C.B. for $\phi_0(1)$ is ∋ t. machine will, after trying it for a while, leave it to try $\phi_1(1)$ — which quickly does not fit. After perhaps trying other ϕ's, t. machine returns to $\phi_0(1)$ — this time with a greater C.B. This time it is successful & goes on to $\phi_0(2)$. Hvr., it takes too much time, so it leaves ϕ_0 & goes elsewhere. When it returns to $\phi_0(2)$ t. C.B. usd is large enuf & it succeeds. — This continues for all N values in $\phi_0(N)$ — so t. machine never converges to ϕ_0.

An easier way to do it: The ccost of $\phi_0(1)$ is > that t. machine allows for its first encounter. Whatever this ccost is, t. machine eventually will return & it maybe several returns are nec'y, befor t. CB is large enuf to do $\phi_0(1)$. When it finally does it, $\phi_0(2)$ has CB. C_2, which is than too large for t. machine of that time. When it returns to $\phi_0(2)$ after 1 or more returns, its n CB maybe eventually $> C_2$, so it does $\phi_0(3)$, which has a CB that's too large — so t. machine leaves again, & eventually returns, etc. (If it never returns, then fine, do frustrate t. machine!)

The point is, that given any ind inf machine, we can construct C_{i+1} from knowledge of t. previous C_i's & t. behavior of t. machine on them.

If C^M_N is t. C.B. usd by t. machine at any time, we can calculate a seq of C_i's ∋ t. machine would never converge to ϕ_0, but return to it again and again.

We could always arrange so that t. first time t. machine tries $\phi_0(N)$ $\rightarrow C_N = C^M+1$ — so t. next time t. machine tries $\phi_0(N)$, $C_N > C^M+1$ so it will succeed. This recursion relation is enuf to get non-convergence on ϕ_0.

0.28.73

Blom

370.40 In t. forzg, wo ondup with C_N as a funct of $N \ni \phi_0$ will frustrate t. machine. Hvr, it is ~~certain~~ likely that for most machines, $\vec{C}$ will depend on t. CB distribu of ~~Each~~ ϕ_i's other than ϕ_0.

06: 366.10 → **Note**: The set of Pams usd by science is a subset of all t. provably convergent Pams. This set of Pams is [redacted] r.e.

08 To show this, simply list t. set of all proofs in order, then test them
09 in order to see if each proves a gn. pam ~~[illegible]~~ always converges. Actually t. "science pams" are not finitely dscrbable (they usually have params w. ∞ accuracy), but they can always be approxd. to arby accuracy by a finitely dscrbable pam.

17 Trouble is, a Pam that frustrates CMI is also provably convergent if it is designed to be orth to all "provably convergent" pams! Is this

20 discovery equivt to Gödel's theorem?

→ **Well**: ~~Fo~~ for each (string, Univl. Tmc, CB) combnn., a converging ~~ARM~~ Pam is defined. ~~[illegible]~~ Say our Tmc is fixd. ∴ T. (CB & string) are a r.e. set. [say t. C.B. is a π limited Umc] Say, instead of a fxd. CB, we use $C(N)$ — a C.B. that depends on Corpus length. ~~[illegible]~~ Even if we use $C(K,N)$ so ~~[illegible]~~ $C(N)$ is a funct. of parameter K, there will be Pams that for any fixd K & large enuf N, are outside t. C.B. —

$\lambda(\dot{x}(\sin(x)))(3)$

~~[illegible]~~ $(\sin(x))$

T. idea of "provably convergent" pams is that a proof shows they converge for all N.

Anyway re: .17 – .20 a string plus a Umc, consitutes a pam — but this pam need not converge for all continuations of t. string, as input to t. Umc. Hvr, a certain subset of strings do so converge, & a certain subset of those have proofs that they converge for all **N**.

We can then effectively enumerate all strings of provable convergence w. a 2 dim scan like ~~that~~ needed in .08 – .09,

So, if one does CMI over all provably convl. pams, one ends up with a provably convl. pam. Hvr, as mentd in .17, a deterministic seq. that is provably convl., can frustrate this CMI pam. T. place that Gödel enters is ~~the~~ perhaps that any finite set of axioms for proving things, will always be "incomplete" in t. sense that there will be ~~[illegible]~~ thms that are neither provable or disprovable by that finite set of axioms.

Perhaps this construction of a counter case is a pam that

0.31.73

Even

1:372.40 is not "provably convergt." within t. set of axioms usd..

Another possl. area of error in t- argt., is the (ipam, pam) difty, but I don't think its relevant. I can use any string to define a pam — it need not be an ipam derin., because I'm not using exactly t. "All pams" method (?). on PCA! → Gen 383.01

.06

.07 : 369.18 Anyway, CMI is, directly a very ~~[illegible]~~ hy ccost way to do induction, as I remember — it's very uneconomical — since it retains even very long codes for t. corpus in t. hope that ~~[illegible]~~ t. corpus will eventually make a very small fraction of them turn out to be useful.

SVH certainly seems far better, but I don't know if it gets as good predns as normal CMI..

Well, say one has this set of Pams, $\{Pm_i\}$, & one wants to use them for predn.

Blums perhaps chose "≈Pams" to be regarded as "best" on t. basis of (a) consistency w. corpus (b) low i ($\approx$ bcost) (c) low ccost ($\equiv$ c.B.) (d) large no. of corpus symbols explaind.

8:
815
900
930
945

(Gen 374.01)