00:137.40 Can be discrete.
If TM has lrnd to respond ≈ like "Maple", does it know enuf for us to try to get it to lrn to understand "English" about what it has lrnd? The Maple-machine can be given problems of A diffy in "solve", "integrate", "simplify", "plot", etc.

.50d (SN) Back to an earlier Q on t. "adequacy" of PPM for Lsrch: As an extreme case, Say we used full Unvl. df. for Lsrch over success problems in a TSQ, using solns of previous problems as corpus. This would recognize all possl regys in corpus & use them to speed up Lsrch. By using a less optimum P.D. to Guide Lsrch, we end up w. solns of Smaller pc & ∴ larger srch times.

T. main idea of $TM_1 = TM_2$ was that we could somehow make t. problem of improving t. Guiding P.D., ~~a problem for the~~ a part of t. normal TSQ. The Phase 1 → Phase 2 is one way to try to implement this.

.18 [SN] on t. need for 2-3 Tree & its large need for RAM. "T distribution sort" = Radix Sort or "Inner Loops" chapter 15, p201 — This uses some RAM, but not near as much as 2-3 Tree.

"If we put stuff in Lex order cheaply, as in BZ<P>, say, we could use the L & corpus for prediction of individual Tokens w.o. having to insert anything into L. We would have to know where to insert a particular context string, hvr. If t. suffixes were in actual Lex order (by addresses), we can do a "binary decision" search taking ~ $\log_2 N$ decisions.

Say we have 2 tables each N entries long. T_1 & T_2. Dm $T_1(N), T_2(N)$:

$T_1(i)$ gives the Lexical order of t. i^{th} shift of t. corpus.

$T_2(i)$ is inverse of $T_1()$: It gives t. shift no. of t. i^{th} lexical context.

To find where a context (Y) fits (its order no. will usually (maybe always?) be ≠ integer):

We compare Y w. shift no. $T_2(\frac{N}{2})$. If its before, t. next comparison is w. $T_2(\frac{N}{2} - \frac{N}{4})$
" " after " " " $T_2(\frac{N}{2}+\frac{N}{4})$

So we find Y's position after ≈ $\log_2 N$ comparisons.

Given Table $T_1()$ we can generate table $T_2()$ in N steps.

i.e. for i = 1 to N $T_2(T_1(i)) = i$: Next i

This would be good for periodic Updating of L. Like Eq of 124.33

The comparison process could be speeded Up somewhat by noting how many common symbols occurred in comparisons before there was a disparity. This would give idea of whether one should jump, say to $A+\frac{N}{16}$ or, if it is much closer, to $A+\frac{N}{100}$ or if it were more distant, to $A+\frac{N}{10}$, etc. → 162.04

4.29.04

4 TM

.00 : 130.40 : 130.37–.38 is impt. Learning to map probs to indices seems like a good training problem because it's easy (fast) to test solns. It's much easier than trying to map from factorized problems to solns... More time needed to check solns.

Th. problem to be solved is finding a best fitting f funct — amenable to L srch ! (would not random method of finding params work ?...)

.05 → Hvr. a major weakness of PPM seems to be its inability to try/ignore usable (or subfunctions) in different orders (?) Check to what extent this is true. (The universal langs can define sets & this may deal w. this problem (tho perhaps not in best way).

.09 In general, if α a_i should be followed by β and a_i can have many (random) values: PPM has to

.10 learn each a_i case separately !

In general, I certainly don't expect PPM to be able to recognize all kinds of regys, but I'm disturbed that it can't deal w. this particularly impt regy type. — Just how to characterize

.12 t. regy of interest? The t. regy is internal to a seq. yet its needed response is

.13 after t. sequence. PPM only recognizes regys that are found to t. ending of t. seq in a seq that (see 133.05 for conclusion of following discussn)

Could we get t. recognition type that we want by > 1 "layer" of prodn ?

SN: T. laws of physics all disallow "action at a distance" — so all causality propagates via a chain of adjacent "disturbances" — yet physics is able to deal w. very complicated causal sequences.

To what extent is (Koza's) GP (crossovers of function trees) able to deal w. $\approx$ .09 ff ? (.12–.13)

.20 e.g. We want to be able to categorize a set of seqs by how they begin. (Backwards : Context) or there is a certain characteristic of t. first 10 tokens obtained by a certain function (strings → reals)

. PPM couldn't discover much of this. Hvr. if we tried to find functions out first 10 tokens

PPM could help find appropriate ones. [10 yrs.] 10k to 10^6

.23 5.1.04 Hvr. anything that can't be done by PPM can be done "at a hyper level" by a univl. lang. — just how to implement this in a good general way, is unclear. Perhaps t. UMC looks at t. first level codes & tries to find regularities ! Or, UMC looks at $\vec{L}$ & tries to find regys — in addition to t. "normal" & normal regression regys. Perhaps have UMC look at t. errors PPM makes! After running a very long time looking for regys in PPM errors, on a very long corpus, t. UMC may get a "Universal correction" for PPM that signifty ↑ its accuracy.

On t. other hand, I expect that grammatical-like regys (nouns, etc) would ↑ PPM

.30 accuracy considerably, would be able to deal w. t. difty of $\approx$ .09 ff. (say .12–.13). A finite state Grammar might help (HMM). A possy: T. set of contexts that predicts either a particular token or a set of "backward contexts" (usually starting w. same token) — this set is a "state" candidate.

Essentially t. problem of .09, .12–.13 is a categorzn problem"

.35 → A perhaps V.G. way to deal w. ↑ is any other problem of incapacity of PPM". T. problem of "improving t. pd obtained by "PPM" becomes one of t. "low level problems" of t. system. By "improving t. pd" I mean to include cc as well as pc aspects of t. "pd" : i.e. it is a "genzd pd" in which both pc & cc are considered: t. pc is a funct of cc as well as of t. corpus!

A Q is : Do I want to (need to) do this in phase 1 ?