TM3 Spec

$\frac{101.7}{3} = 33.9$ fluid ounces. 33.8 oz = 1 liter! $\frac{1000}{33.8} = 29.585798$ ≠ 28.35!

20: 451.40: Now, if a certain cond; seems to be getting a lot of wt., would it not want to give it a hyer fraction of trials (?) — so its influence would be more accurate (larger ssz).

02 If so, We'd have to modify f. wts --- But drop this line for a moment:

03 Suppose we don't do .00, but just run the system for a long time. In Theory, we approach U.D. —

04 Since it is all discrete model can give us times all continuous values on them. As we slowly ↑ cc, TM gets better and better. In lines .03-.04 we considered cB=00 & it looked fine, but for because all models did cB=00 & no "ssz error (ssz is no of Mkt-Cap tiers [illegible] not corpus size)

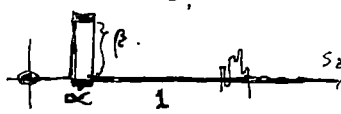
So the Q is how much error we have for ssz c00 --- particularly, small ssz!

— and could considerations of .00 — help it?

As each cond. accumulates data, it might be possible to emphasize certain of the conds so as to maximize the amt. of reduction in the result, per "unit trial" — This is somehow mindful of work on SM in deciding how much Money to invest in each stock so as to minz $\frac{\sigma}{\mu}$ (or $\frac{\sigma^2}{\mu}$?).

13 → So Q is: How well does it do in cB=cJS? (C.IT)

Also Note: for a corpus of such size (say SM) pc of corpus will be very small, so cJS would be enormous: I must find way to deal w. this!

7 (.13) Try simple y(x) function: 

say pc of discrete part of cond is p_1.

∴ T. pc of "soln" is, say $\alpha\beta \cdot p_1$. If it take time T for final $cJS = \frac{T}{\alpha\beta p_1}$.

0 Now let's do Mt Carlo search. Say it take time Δ to test a cond pc of a trial (to "test" it).

2-1 If we spend time T in search we will spend time $p_1 T$ on trials for cond.; ∴ we will do $\frac{p_1 T}{\Delta}$ trials. ∝ $\frac{p_1 T}{\Delta}$ is f. no. of trials that will be "successful" & will obtain pc = β.

Clearly, this is indep of β & ∴ seems wrong. T. Mc search seems divided into 2 parts:

24 i) Choosing a discrete part & ii) working on the continuous parts: Actually, After "the" "α" region is found, we can use standard continuous optim methods. To make α larger, use a small ssz (of corpus). Then when α is found, increase ssz to find its peak,

29 then ↑ ssz & look for new peak, using old peak as starting points → for a very nice treatment on this → 453.2C

So: It looks like main problem is finding the "α" region, & the cc of this would seem to not be much related to the size of αβ — eg. not ∝ $\frac{1}{\alpha\cdot\beta}$

On the other hand: Once one has picked a cond., a single random trial in "x" space, 451.16 will give rise to an ∃y which is the pc of that cond. — The problem is — that pc is (for larger corpus → larger ssz) very small. The probty of "hitting" α is also very small (usually) To αβ is certainly not the same as α. So while the Expected value is fine from βayes; the Variance and so estimate of y & hence in one's prediction, is enormous! So this is why we do .24-.29.

If we don't do ≈ (.24-.29) then we have to do a real Mt-Carlo trials to get the variance of the random ppn. to ≈ the var. of the actual distribution in X space. — Which is usually quite expensive. ——— 453.00

35m

On R functions: a Genzng Prcss. There is one way on: 430.15 – .33; 431.20, 451.30

Note: 430.10 I did w/out a nice MC code yet — Maybe later ~ 400.00!

430.10 is mainly about zPsip based on pcs

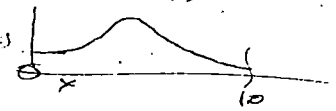
430.14 To do a search over a space of funcs w. discrete & continuous params:

This is to get funcs of hy pc that "fit" well — for nbrhoods (e.g. ≤ M),

Say one does it MonteCarlish: We randomly chose cand$_i$ w. pc = p_i; p_i is pc of discrete part of cand$_i$'s dcrn. Say its a continuous d.f. (T. discussn. probly doesn't care how many cont'n params there are). T. param has values apriori from 0 to 1, $(0,1)$.

It takes $c =$ zbtn to generate n trials, to see how good it fit is (Ξ zPsip) — b is incremental time for a trial

Say $Y_{\text{cand}}(\Xi \text{zPsip}) = Y(x)$ (x is cont'n param on $(0,1)$.)



We are really interested in $\exists y$: i.e. $\int_0^1 Y(x)dx \Big/ \int_0^1 1\,dy$ normzn consb = 1 in this case,

Say $\exists y_i$ is expected value of y for cand$_i$.

This is very wrong! See .33

To make r predictions. Say Race and test r different discrete cands, &

Z_i = prediction using present data, using discrete params of cand$_i$ & continuous param $\exists y_i$.

Then we have r distinct preds. The preds $\exists y_i$ is given wt p_i (.12). These preds are the d.f. of the output of our predn system. (see .33 for error.)

Say we have many cands ($r \gg 1$) & we only have n trials per cand.

It would seem like a good idea to do more trials for cands in which the single y_i was large.

In the usual MtCarlo Lsrch, we spend a fixed amount of time Δ on each MtCarlo trial.

So from this we'd do $\approx \frac{\Delta}{b_i}$ trials for each time cand$_i$ is chosen (I say "$\approx$" because they doesn't consider "startup costs" for a set of trials — in general, the no. of trials for cand$_i$ will be slightly a/more complicated fcn than $\frac{\Delta}{b_i}$)

If n or r is small enough & we have a no. of trials $\gg r$, then we will have several Δ periods per cand — so the no. of trials for cand$_i$ is $\propto p_i$, the no. of times it is (randomly) selected.

The Discn at .20–.23 on pg 1 is wrong. Say b is exactly (.15) $b_i = b$ is the same for all i. (Same inc. time for each trial). When we chose x_i for cand$_i$ & its graph is Y_i, we then make a prediction, using discrete params of cand$_i$ & continuous param, x_i, & we get this pred wt. Y_i. We do many trials, so the no. of trials cand$_i$ does is $\propto p_i$, & the no. of micro trials it does is $\propto \frac{p_i}{b}$. Some just wt. all preds using x_i by its corresp pred wt Y_i. The wt. $\propto$ to p_i comes automatically since the no. of trials for cand$_i$ is $\propto p_i$.

spec 452.00

20: 447.40: SM cont.: But, say I don't bet "every day" but only on certain days, Even so I have these empirical sequences of fractions of money made (i.e. $\frac{\text{amt won}}{\text{previous Bank}}$) using Kelly, I convert Bernoulli probs — so I have a pc seq. à la pc of coding the strategy, given mean gain — which enables me to compute "true expected yield" of the strategy.

Say I have r distinct strategies: Strat r has gain g_r (due to code cost) & unit. yield γ_r for the time interval (of trial.

The expected yield from this set of strats is $\sum_r p_r \gamma_r$.

If the set of strats has only 1 param, α w. uniform dist from 0 to 1 then expected yield is $\int_0^1 \alpha \, d\alpha = \frac{1}{2}$.

Review Past stuff on Forecasting of good Continuous Poems for ALP

A perhaps V.G. way to do SM Part would alcohol & write TM's for TM:

Use a large bunch of strategies: Then "factor" them à write a grammar for them; then use it. Grammar to generate new strategies. Freeze strategy in the original set, give a wt. of its Gore

Some strategies &/or predn schemes for or parts of them:

1) linear predn. 2) N.L. predn some standard ways: ANN, RANN.

2) If system is in state S_i in Morning, then bet/use system S_i in afternoon.

For example: If system found 4 points in 8 minutes on opening; Then bet/use for fixed time. T_0. or until end of day.

3) Cross Predn schemes:

4) Ways to save time in searching.

5) Perhaps look at old copys of "Stocks & Commodities" — or get them CD Rom.

6) Try using kinds of Prediction (for Compression) methods — from eg. "Pattern Classifn" & other sources

$25.03

TM3

SM .21

WWW.02six2sixcasino.com
1-800-922-4515

Give back odds for HR Betting.
In Curacao (Caribbean Island, Dutch)
Bets $.25 to $2.00 per bet. — HR
$200 is not so big ! (May be ok for me on !)
$1 to $5000 for Sports betting (Non-HR)

446.90 Another similar problem: Should I spend $ on trying to cure cancer or on testing General Relativity?

! On defining ngrams in BZZ: in 446.00:14 I considered ngrams ending in the corpus that ends in "β". The set of all ngrams in corpus that ends in β is unique. — i.e. $\tilde{\beta}$. By looking at $\tilde{\beta}$ I may be able to devise definitions" that are ① cheap ② cover much of $\tilde{\beta}$ ③ do not include many ngrams that don't end in β.

Another view: In BZZ I am treating the corpus as an "encyclopedia" and the problem is to obtain a cheap code (w. that corpus as input) for the next symbol. Cheap in bits cost (not nearly c cost). Stated this way, it seems I am thinking (to find) the pc of the next symbol of the corpus! (??)

BZZ Peny: Discuss w. Steve Wilson: To code a/ (complete corpus, we get all n "shifts" of the corpus in lexical order (probably I'd want to do corpus backwards, so I could predict forward!)
To do this incrementally, we just add a symbol to the corpus, then insert the augmented corpus in the alphabetical (lexical) list of shifted corpi.
To get pc's it might be best to put each shifted corpus in lexical order w.r.t. the all but last symbol (the last symbol being the one to be predicted). Or, if one does a circular shift as gets them in backwards lex order, this would also be o.k.

Another way to look at BZZ codes in terms of ngrams: Say the terminal string was "wxy" occurs, and we have a k symbol alphabet (Russian), & the corpus is of length n. Consider "xyz"). The ngram "xy" is a ngram of length n, with k^{n-1} members — all ending in "x" : "yz" is similarly a member of the k^{n-2} member ngram set of all n symbol ngrams that end in "yz"

SM: Take daily Δ prices for stock : Do dft of any/all of them, then divide into 2, 3, 4, or 5 levels. This is marked a radix sequence. Use BZZ to see if I can get useful predn.
Use long sequences, so that startup costs are minimal. Also look at actual probes obtained, so I'm sure no bits are being wasted.

Try it with variance smoothing: Obtain σ^2 in one of several ways & quantize (into k levels) in view of expected range of σ^2. Some ways to get σ^2 ① from Option prices ② x window var R window σ^2 ③ Median filter: so $e^{-\frac{|x|}{\sigma}}$ is used as d.f. — This should work better than Gaussian.
Q: Is there a way to do x window median filtering?
Try .26 w. various window widths & various stocks. Contrast w. 1, 2, 3, 4 coin (new) prob.
Using the quantization method of .26-.31, do prediction based on (previous) 1 or 2 or 3 lengths ngrams (simple Markov model). Compare resultant Entropy w. that obtained by BZZ.
Try .21 ff on cross reading using BZZ; I may be able to modify lexical analysis of (.10 ff)

SN on SM: ~~Say I am betting on a strategy using (really betting (really))~~ Say I have a strategy which is of form that says buy at any time at Bank. Then I can (assume really cost) associate w. that strategy, a probability estimate for each bet. This enables me to use Kelly & coding costs for the strategy & use AIP to decide how much $ to give a strategy. After I've been running a while. → 449.00 spec

: On dealing of ngrams in Bzz: Consider a linear corpus that's a linear seq. of symbols (notes, e.g.)
The last token is "β". Consider the ngram that is the set of all suffixes (to date) that have terminated in β.
Call this ngram $\tilde{\beta}$. (Since all the ngrams in $\tilde{\beta}$ are of length of seq. to date, none of them have ever occurred
before, so they can't be "productive"). Consider models of $\tilde{\beta}$ that have short clauses (b) are predictive (in the sense
of having occurred at least once before). Some ways to construct such ngrams:
1) The elements of $\tilde{\beta}$ are all of length 1. Augment $\tilde{\beta}$ with sequences that are "don't care" from the 1st to jth
element, not otherwise identical to some seq. in $\tilde{\beta}$.

.02-.03 is wrong: ngrams in $\tilde{\beta}$ are of various lengths, but no 2 elements of β are the same length.
Derv. $\tilde{\beta}$ elements "backward" - so the first token is always β. $\tilde{\beta}_r$ is of length r. $(r \le n)$.
Using this "backward" form: A way to combine 2 ngrams (say $\tilde{\beta}$) with another ngram (say γ, so for all possible strings $\equiv U$
we combine $\tilde{\beta}$ & U by giving intervals i, j $\le (i \le j) \le$ in (i,j) we AND $\tilde{\beta}$ & U
or for other regions we OR $\tilde{\beta}$ & U. Say $i=3$, $j=7$; then use of use AND
from 1 to 2 and 8 to ∞ and OR from 3 to 7, we get $\tilde{\beta}$ set but from 3 to 7 inclusive)
any symbols are legal parts of the ngrams.

The problem is an early skirmish w.r.t. problem: In the solstice (kind of meeting) I took ngrams & ngrams,
But I didn't appreciate that ngrams were "languages" that have "dems", & that wild ngrams were a complete
model of induction. What I probably should do: In the induction problems in my TSQ: write up the "solns"
to problems that I want to use; then see how I can derive ways to construct (combine) ngrams
to get the needed solns.

In .00-.14 I think I was trying to expand the scope of effectiveness of Bzz, but here, also,
I should perhaps first decide which of the new kinds of ngrams I want to be able to discover — then
find ways to use ngrams, ngram etc to express & generate these new kind of regs.

The Bzz model is an open Markov model. Each symbol puts us in a known state. We can't do define
sequences that put us in a known state, or "assign a distant" tokens to seqns that put us in a certain state.
Hidden Markov Models (HMM's) must be more general than Open Markmods since OMM is a more special
case of HMM. This is peculiar! HMM would seem to have essentially more powerful set of models than OMM.
[Models of this kind HMM in that the "states" is never directly observable; it's inferred probabilistically; it's never
known for sure]
Yet OMM can be generalized to wild ngrams which is universal!! This of OMM doesn't seem to "pass thru" Humm,
Yet it must include HMM!

In explaining how Induction etc. deals w. learning; & SciMethod. "SciMethod" is how the SciCommunity
"Learns" (to some extent). But the SciCommunity is More like Org. evolution in that it really has no "Top
Goals" — it makes up sub-goals as it develops. These sub-goals can lead no where or to place very
peculiar places. I suspect that the goal of "understanding the Universe as a whole" may be meaningless.
If the SciCommunity is given a problem, I think I know how it solves it or should solve it.
The Q of what it decides to solve is usually inductive, but very randomish. "Induction" here means that
its new subgoals are patterned after what seemed useful in the past? Deciding on what
new data to (acquire/pursue) is decided in a similar way.

(Animals), Humans
SciCommunity & Org evolution
All 3 "learn" in different ways. The SciCommunity learns via the "SciMethod"

447.00

D28.02

TM3

REV:

T. Bumpus p-soo D23037PM — no. of bumps & w. numbers: 24 Bumps May — before p433 the Bumps

.00 – .40 Is a v. bad quiz question of ~ 400.00 – 440.40

Revs: 437.00ff – 438, 439, 440, 441
429.31 – .40 Brent
421.12–.18, .25–.40
422.00–.40
403.24

20: 441.40: One way to organize y. ideas ab 441.00–.40: Consider what I feel is y. most direct approach to Strong A.I.: Tell how to do it: Then show how (most of) y. ideas of 441.00–.40 fit in.

The method is: Phase 1, Then Phase 2! (Tell in detail how Phase 1, 2 work). First: [illegible] find subset of Σ^{*} in NP/CA/(aj) is map. To approximate

24 In "Phase 1" we have an a priori [illegible] models of y. data PD1: PD1 is d-mod. D.t is incomputable. It is incomputable

5 To find good models of PD1, we have PD2 which is a PD to guide search for good approx (types) to PD1 ← Not y. best way to search, but ok for Phase 1, Best

SN In .05 we may run into "Encyc" problem! Guard by d. sets E (Like Encyc or Internet) to get best ans. "Encyc"

to Question w. $cc \leq CB_0$. If one has many Q's w. answers in E, one way decide "Index" for E. Explain: Encyc problem occurs when we don't want to have to code entire corpus to answer a Q. Perhaps E "just required E" is part of Q

The TSQ for PD2 is fr. seqn. of Q's, A's & Hopping functions O^{*}, that have been successful in y. past. Phase 1's strategy is to try techniques w to those that were successful in y. past. Phase 2 tries for "Best soln in available CB"

0 The TSQ for PD1 might be (rng & doms of Maples, MaDemoiselle: eg. equal, equal literally, equal numerically. Solve / [illegible] environment forms. distributive, Integral. These are all d-preds Q's to start.

T. problem of involving TSQ is simple — change of TSQ! One of big problems: What TSQ to use? How to represent S-funct's so that previous d-funct solns & other components are useful for next TSQ? How to use d-funct solns. to help create S-funct solns.

5 ..... In both PD1 & PD2, context is important part of discovery of short codes ("search"). w.o. context, each token has a unconditional probability. By using definitions of nyms, we can implement use of nyms for given context → pseudo. However, defining only nyms will only give nym context. For more general contexts, we need to be able to define nyms. For most general kind of context, we need to define wtd nyms. However, definitions has advantage of speed, but disadvantage of possibly wrong "parse". PL's of Tolerus documentary small as esse 4 (Scaling problem) if context is ignored. However, if nyms defined over larger area of "context" (uniform pattern) is to be strictly considered.

FN Re: .05 ff: If we did not be able to define wtd nyms nyms as contexts, Then it would seem that definitions of those forms could be used for pseudo "w.o. context"! T. simplest case is nym defns. In general, if can define nyms then we don't need wtd nyms to define nyms correctly. Similar ideas hold for defining nyms & wtd nyms. (See 441.23 for way to define nyms & wtd nyms (Key test is any pdl on strings :: definable (as by) a stochastic, or by defining a U.T.O machine or function.

One apparently good idea: Post Definitions were not necessary for inductions but they did a certain amt. of "pre parsing" of y. corpus to [illegible] save time in later parsing. The disadvantage is that y. defns did not always parse y. corpus properly — so they should be used only if they have hy PC: or if nyms & small set of defns that have almost all of y. PC of decision" at a branch.

28 — E.g. BZ2 doesn't use defns — Rather it is very fast. This is perhaps because of y. only kinds of Recognizers via BZ2 4.3 Concepts it uses is nyms (no nyms or wtd nyms)

30 One of Big Ideas of last month: Use of BZ2 for PD1 & PD2; Hvr. note .28! Also note need to deal w. "degeneracy" in codes for functions & "symmetries" & other equivs. Foundational Forms: This is not a problem peculiar to BZ2, hvr, it is important in all of TM: all of prob by exam, apriori exam, scratch.

33 Items in 441.00 –.12 not covered (mentioned) in 442.00–.33!

Alternative Approaches to "STRONG AI"; 415.20 – .33 lists ~ 14 "induction methods/AI methods. Each has deficiencies w.r.t implementing "STRONG A.I. How to "Mix" them to get SAI? Step

One way: For each approach, Like good things it does: its useful skills. Also tell how it misses SAI: what it needs to get SAI.

One way is to see how each of y. others can be brot to bear on y. deficiencies of each method.

20 : 432 3/4 ! It may be possible to prove 432 3/4 · 19, 20 by Hall induction !

Write out expression $\sum_{k=0}^{n} R(k)R(n-k)$; $R(i) = \prod_{j=0}^{i} \frac{4j+2}{j+2}$ ← Explicit formula.

each term has same no. of factors,

perhaps write expressions in terms of factorials.

$\frac{a_{n+1}}{a_n} = \frac{4n+2}{n+2} = \frac{2(2n+1)}{n+2}$ $a_0 = 1$ $a_1 = \frac{2}{2} = 1$ $a_2 = \frac{4+2}{1+2} = 2$

$= \frac{2}{2} \cdot \frac{2 \cdot 3}{3} \cdot \frac{2 \cdot 5}{4} \quad \frac{2 \cdot 7}{5} \quad \frac{2 \cdot 9}{6} \quad \frac{2 \cdot 11}{7}$

2 3 4

$\frac{2 \cdot 7}{5}$ $R(n) \cdot \left(1 + \frac{n+1}{4n+2} + \frac{n}{4n-4} \cdot 2 \right)$

$R(n)R(0) + \frac{R(n)}{R(n)} \cdot \frac{R(n-1)}{R(n)} \frac{R(0)}{R(0)} \frac{R(1)}{R(0)} + = R(n)(R(0)) \cdot \left(\frac{R(n-1)}{R(n)} \cdot \frac{R(1)}{R(0)} + \frac{R(n-2)}{R(n-1)} \frac{R(2)}{R(1)} + \frac{R(n-3)}{R(n-2)} \cdot \frac{R(3)}{R(2)} \cdots \right.$

$R(n-1)$ (1) $R(1)$

$= R(n)R(0) \cdot \left(\frac{n+1}{4n-2} \cdot \frac{2}{2} + \frac{n}{4n-6} \frac{6}{3} + \frac{n-1}{4n-10} \frac{10}{4} \right)$ (2)

Show $R(n) = \frac{4n+2}{n+2}$

42

$\frac{n+i}{4n-j}$	$\frac{j}{3-i}$	$\frac{n+i}{4n-4i+6}$	$\frac{4i-6}{3-i}$	$j = 4i-6$

$\frac{n-i+2}{4n-4i+2}$ $\frac{4i-2}{i+1} = \frac{n-i+2}{i+1} \cdot \frac{4i-2}{4n-4i-2} = n$

$\frac{n-i-1+3}{i+1} \left(\quad \right) = \left(\frac{n+3}{i+1} - 1 \right) \frac{1}{\frac{4n}{4i-2} - 1}$

$= \sum_{i=0}^{n} \frac{1 - \frac{n+3}{i+1}}{1 - \frac{4n}{4i+2}}$?

26 Note: Something like $n!\,R(n)$ is actually redundancy — since can put objects in order as well as arrange partially ordered sets:

	1	2	6				
n	0	1	2	3	4	5	6
$R(n)$	1	1	2	5	14	42	132
$n!$	1	1	2	6	24	120	720
$F(n) = n!\,R(n)$	1	1	4	30	336	5040	95040
$R(n)/R(n-1)$	—	1	1				
	1	2	3	4	5	← $n+1$ (= no. of "levels")	

3 $G(n) = (n+1)! \cdot R(n)$ | 1 | 2 | 12 | 120 | 1680 |

$G(n)/G(n-1) = 2(2n-1)$ — No. Great surprise, since way have simple formula for $\frac{R(n)}{R(n+1)}$

	1	3	5	7

actually $(n-1)!\,R(n)$ is the effective redundancy

n	1	2	3	4
$(n-1)!\,R(n)$	2	12	120	1680
	2	12	120	1680
n	1	2	3	

$= \frac{(2n)!}{n!} = (n+1)(\cdots)(2n)$

I'm not so sure!

Sec 458.25 R

$= 4TM(6.26R)$

$= 4TM\ 6.20$

$\rightarrow 458.20$

(CN) $\frac{1}{x-1} = \sum_{n=0}^{\infty} \frac{n!}{x(x+1)\cdots(x+n)}$

Pol: "Operational Calculus" p 252 Corollary 1

Pol uses Gen Funcs: pp 251, 253, 272

$\frac{1}{1-x} = -\frac{1}{1-x} = 1 + x + x^2 + \cdots$

Typical exp: $\frac{d}{dx}\left(\frac{1}{x-1}\right) = \frac{-1}{(x-1)^2}$

double integrate $\frac{d^2}{dx}\left(\frac{1}{x-1}\right) = \frac{2}{(x-1)^2}$

90×180 9×9×100×2 16200 = 16.2K concepts.

≈ 360×180×2 / π

spec

0 :445.40 that I wanted the probability of [redacted] the next symbol being β. If I were using the context [redacted] · γ, (the ngram preceding β) to obtain [redacted] probability, I could obtain the same result by defining the [redacted] ngram γ·β (concat), and not use context. [--I'm not completely sure of this effect, however; I have not worked out the detailed math to see just how (or if) it works.] ← end of F.N.

There are at least two kinds of Context. The first "External Context", tells where the problem came from a/o how it arose. Some examples: (Problem in) chemistry; Physics, Math, differential equations. Or "the problem arose as a subproblem in converting a INV problem to a OZ problem," or "the problem arose after I tried to integrate using "Integration by parts".

The second "External Context" relates the previous corpus (or part of it) to the symbol to follow. Most generally, it is the problem of sequencial prediction. Less generally, the last n known symbols is a context for predicting the symbol to follow. More generally the previous (ngram / ngram) may be a member of an ngram set (ngmst) that can be used for predicting the probabilities of the next symbol. The most general kind of Context is the Weighted wighted ngmst; this can correspond to using the Universal Distribution for prediction, [redacted] if we use the entire previous corpus as "context".

16 A common kind of context ngmst is that used in raster scan graphics. If a line is k pixles long, then any pixel will have as context, (1) pixels immediately border it (2) epixel [redacted] that is k pixels border it. This latter corresponds to the set of all ngrams of length k+1, that start with a particular pixel. If each pixel can have n values, there will be n^k ngms in that ngmst.

D Ngmsts correspond to deterministic "formal languages". They can be described by generative grammars or by recognition functions.

Weighted ngmsts correspond to probabilistic languages. [redacted] They are probability density functions on the set of all strings. They can be described by

25 stochastic generative Grammars or by functions that map strings into probabilities.

To obtain the probability that the next symbol in the corpus will be X_i, we have an associated ngmst, N_i. For d induction N_i will be a d-ngmst. It could be defined by a function $F_i(\cdot)$ — say the range of F_i — the set of outputs, for all legal

2y inputs. Associated with F_i will be its apriori probability, a_i — the pc of its shortest description.

D For S-induction on that same corpus, N_i will be a S-ngmst — a weighted set of ngms — with each weight dependant on the description length of that ngm. If the N_i ngmst for d-induction was obtained as the Range of a function F_i (the set of outputs of F_i with legal inputs) — then we can associate with each legal output of F_i, the total pc of all inputs to F_i that obtain that output — creating a p-ngmst.

→ 4TM 12.00

cont from 16200

↓

450.95

10:444.40 There are many ways to try to solve the optimization problem of 444.24. In "Phase 1" we will begin by using Lsearch, with a guiding PD2 that is the same as PD1. After we have solved several problems, we will use a guiding PD2. That is based on the sequence of O''s that have been successful solutions. O''_k will be a function that successfully optimized 444.24 for $Q_1A_1 \cdots Q_kA_k$. One way to do this is to use BZ2 (or an augmentation of it) to extrapolate the sequence $O''_1, O''_2 \cdots O''_k$. BZ2 will give probabilities to all possible continuations of this sequence. This gives PD2, the prob distribn on O''_{k+1} — which are the candidates we test, using 444.24. We try the cands. in ± PD2 order and spend a maximum cc on each that is proportional to its PD2. [N.B. I do mean PD2: It is the Pd that assigns cc's (limits to candidates that guides the srch, not the order in which the trials are made)] A bit unclear! ↑ no ampl

---

The initial TSQ (≡ TRaining Sequence) will be the learning of definitions from Maple, MAthematica, etc. Learning: eval., eval. numerically, differentiate, Integrate, Solve literally, Solve numerically... can be an adequately difficult task — some of these functions can be arbitrarily difficult.

The initial training sequence will involve d-induction (determinism induction), in which there is only one correct answer for each question. Mathematical and (general problems can be of this kind. Though d-induction certainly does not cover the range of problems that we want TM to be able to solve, nevertheless, if we are able to teach TM to solve really difficult problems of this sort, it will be an encouragement for the general methodology we have been using.

23 Next on agenda is p-induction (probabilistic induction), in which each question may have several different answers — each with an associated probability.

p-induction ≡ s-induction

We want to devise TSQ's for p-induction that are able to use the concepts/definitions used in d-induction as useful concepts and sub-concepts for s-induction. Some suggestions on how this might be done are in the section on Context.

Actually, this is a BIG topic

30 **Context:** The need for contextual considerations arose in the "Alg. Notation Lrng. Program" in Sept '61. The boost in solutions of successive problems increased very rapidly — so it was clear that difficult problems would become impractical to solve. By taking "local context" (i.e. recent previous item) into account, it was felt that the pc of the next trial bit would be large enough to make the solutions of difficult problems possible. (The "Scaling Problem" → to deal with the "Scaling problem")

[An aside (≈ Footnote): More recently I realized that if I was using item definitions for induction, that simple item contexts would not increase my prediction accuracy: Suppose

SPEC 448.00

TM3 Rev:

: Summary of Notes at tops of pp.

1) Rev: see top of p 443 for list.

2) Org. Evoln. 414, 426, 427, 429, 439

3) TSQ, 403, 419, 425,

4) Other 1 page writeups: 413: How to Construct Recogn. Functs using BZZ

418, 420ff the Redundancy factor : Nuclear Reactor : External cross refs.

425 Context (Merch on Prez ~~Mission~~ after 425)

439 SM : Importance of Gambler's ruin in strategy management.

405 OSL : This may not be such a new idea : BZZ does it a "predictive MDL does it

Tag indices:

dot OSL

0 403 TSQ writeup

403 Rev

359 Fifo of 589

439 SM

359 OSL ev

437 rev

429 Org Evol. outside

425 Rev

427 org ev

426 org eval

0 425 TSQ

425 Context

simpl reform.

421 rev

420 "Metaphysics"

0 419 TSQ

419 "redundancy factor"

414 org ev

413 how to write a "recognizer" for nets.

O.K. So: a more detailed review of 443.04-.40 : I will then add ~~≡~~ "relevant" cross refs to Sections if ~ 400.00ff a/o refs to new ideas.

First: The problem is, given a sequence of Q A's: $Q_1, A_1, \ldots Q_n A_n$: To find a set of probabilistic functions, O^i, such that

$\alpha \equiv \prod_{i=1}^{n} a_i \prod_{j=1}^{n} O^i(A_j|Q_j)$ is as large as possible .... in the available time (or CB = computation cost limit ≡ "Computation Bound").

a_i is the a priori probability of the Probabilistic Function, $O^i(\cdot|\cdot)$.

To solve this problem, we start with a "known" PD (Probability Distribution), which is the a priori a_i, on the set of probabilistic functions, $O^i(\cdot|\cdot)$. PD can be realized in several ways: ① The AZ language of the IDSIA report ② The FORTRAN-like OOPS (any. that Jürgen uses ③ Either of these augmented by ≡ BZZ, which exploits redundancy by observing repetitions of patterns. ④ Probably many other ways.

The a priori PD should be selected by the "Trainer" so it seems useful for the training sequence being used.

D16.03

TM3

Rev .00

$\frac{(2n)!}{n!}$ (number)

$\frac{(2n)!}{2^n n!} = 1 \cdot 3 \cdot 5 \cdots (2n-1)$

must happen integral power of 2 factor!

n	0	1	2	3	4	5	6 ← wrong!
R(n)	1	1	2	5	14	42	
n! R(n)		2	12	120	1680		$= \frac{(2n)!}{n!} = n(n+1)(n+2)\cdots(2n)$
		6	10	14			
		3	5	7	← NR!!		← This means should R(n) is integer!

20: So: Big ideas: 437.00ff!

① 2 Pd's (PD1 ; PD2) approp Guidance

② FN2 of S89 : Related to (PD2, PD1): Phase2 of TM; Adaptive Lsrch. [This is an impl. set of ideas!]

③ Context : kinds of; Genesis of . : related (ETA) General Induction: Use to date (w. "Scaling" problem)

④ TSQ's for Phase1 : from Maple data, & MTM ect.

⑤ Use of ≈ BZZ "for enhancement of" for Phase1 ; for PD2 (or PD1).

⑥ Impln. of "Symmetries" in [redacted] apply of Funcs that are solns to probs. (Implt. in use of BZZ)

⑦ Org. Evoln: Various ideas: Use of BZZ for GA!

⑧ MTM solns → NMTM solns: / "How to express S-Funcs" related to & Also note General Context problem (.03)!
The 3 vector soln to the h(T) problem in Phase2 . (see .23) Z

⑨ STRONG AI: Try to solve by unifying, mixing, & augmenting various recent, & "Mature" approaches to induction 415, 20 -.33 lists some approaches. → (20Z)

That idea of best comp. of continuous forms for predn.

20 .12 One Approach: List various Methods: Tell just why each is deficient as approach to Strong A.I. Also, tell what each is Good for:

Then see which of [remaining] deal w. deficiencies of any of rest, [redacted] W

23 .11 Ngrams: How to (describe / measure): ① list-of-ngrams (2) any distance function from canonical ngram, Threshold. (Hamming, Euclid, Manhattan ...) ③ combine ngrams (Boolean; Concatenation) ④ Grammars: CFG; CSG; Trans Grammar (Generators for Decision Criterion "Recognizers") Boolean Any Universal machine defines a pd on all strings ∈ Σ* ; ∴ all ngrams

Grammar or Language" is a [redacted] General form — can be universal:

Q: Are ngrams a universal soln. for d-induction? Say we have a custom UTD machine; The set of all poss outputs defines an ngram. This Seems like a very general way to define ngrams. The process of its set is the process of the machine (or function) that defines its set. E.g. use ∀ ∃ Z formalism to assign pd's to functions: Then a given function defines the Range of that function. [redacted]

Or — conversely & the range of a function defines the ngram. (This "range" is for all possible inputs.

A definition of this kind is used to define (generative defn) d-languages. (CFG; CSG)

We consider "any" (∈ ngram) the set of all poss outputs of the "generator function"

→ Re "New Idea" of this last Month → Possible use of BZZ for Phase 1 : Use for PD1 ≈ PD2.

As is: BZZ has no Definitions"

TM3

50 : SN At various stages of TSQ, we may want TM to have distinct "wts" for it importance of different parts of the TSQ. This can be approximated by giving wt. factors that TM will use as exponents of assoc PC of subcorpi (or in other wts. for b costs).

SN - for recent "Review": For each item reviewed: Give final state of that problem: just what needs to be (solvd/done)? Keep this list in Mind! ABCde

36: 438.04 :→(MTM→NMTM tsq prolgm) :OK certainly seems clear that concepts for MTM should be useful for NMTM

:) : (2) We can Teach TM probs, strategy directly as "Academic Problems." Many (?) problems in this area can be formulated as MTM problems! — e.g. many grade (curve best) problems in books tests on probs/statistics can be formalized so they only have one correct soln. Th. MTM (d-concs) used

6 to solve such problems should be useful to solve other problems of S-p readn.

A NICE thing about this is that TM could learn many different ways to construct S-functs — Review in to convert before from

Another NICE thing about MTM lang is that much of Math can be stated in this form — in particular, we can give MTM problems in logical reasoning — Rich gives problems in which "logical reasoning" is useful to help solve them — the connection betw "Textbook logic" & "logic to be used for solving real problems" is "analogical" & ∴ learnable by TM.

Note that ~~NMTM~~ NMTM's solns need not be in SIU form: Any form is usable.

E.g. The 3 parameter forms of h (M) functions (See 438.07-08 for Bibl. refs)

DISCUSS N.B. — While (Parnas) ADT modification is d-induction (in final form), I strongly suspect that s-induction is

20 needed in many stages for d-funct's solns. Some of this problem/task can be done via PDZ in "Phaser" — we can augment this PDZ by (perhaps) by making improvements to BZZ — like upcasts & (finally) wtd upcasts.

directory folder

www.rhemcast86.com

NB BZZ.zip : written in "Euphoria": It's a set of files to implement → BZZ.

So see what it is: There exists a Euphoria to C xltr. (

Euphoria is normally an interpreted lang. — But speeds up w. conversion to C.

In c:/BZZ dir I have a dezipped BZZ.z.ip dir Not easy to understand!

[Think Main file is BZZ.E. only one ".E" file in this directory,

O Done by a .DLL file)

SNS: 1) try BZZ on BZZ for various beads, genes, etc. See if it can't compress it.

2) Association in t,-1 ; X ÷ [its impl. stores as strings dependency

3) Can I use Euphoria functions in EorR, to speed up Euph?

4) BZZ doesn't actually do "production", it only knows data. In this sense, it is ideal for "updating", which does some things.

D.12.03 Rev .00 - .07

THS SM .14 ← (T. Importance of Gambler's Ruin in strategy devising)

Org ev. 00

0 :438.02 : Org. Evolu: In Nature, before 2 viable individuals, combination is almost always viable: T. Variation is w.r.t. "alelas" which are "similar" sets of "genes" : How can we characterize this aspect of org evolu? Another trick is "Quick & Dirty", in which in many/most cases, a potentially unviable cand. is not born in 1st place — saving much cc in "trial" cost.

:07 ~ -417.00 -.26 on possl. improvements in "accepts/rejects" of Eng. text - i.e. what is a "true" d.f. ∃ (rank v.s. Probs).

08 :437.16 SM Some ideas on symmetric functions: I can suspect 2 funcs are identical if they agree for several distinct (or one BIG) random input. ∃ Can B use this idea in Search?

SM To Augment BZZ, I consider Regs of this type, Detected (used) by Wolff & by Ming Li in his Generics pgm)

Also, try to Think of other Pred pruning pgms, techniques, that could be included to improve BZZ

Remainder, Ingms B is a Big improvement of BZZ & wld Ngmsts is 1st most general type of pred. pass!?

14 SN SM In general, When ones "Bust", $B \le B_0$, multy yield ≤ 1. Say λ = mult yield.

then $\lambda = \lambda(B)$ $\lambda(B_0) = 0$; $\lambda(B=\infty) = \lambda_\infty$ (which is some finite value). ← Pair is the fully conf=1.

so $\lim_{B\to\infty} \lambda = \lambda_\infty$. As B approaches B_0, (fully conf) k à become 0 at $B = B_0$.

$k \to 1$ as $B \to \infty$. We want to make k a funct of B so our probs of Gambler's ruin (i.e. $B \ge B_0$) becomes small.

As $B \to \infty$, hvr, probs of GR (Gamb ruin) becomes small, à probs we should use $k \to 1$. The reason is : $k > 1$ if we approach max expected gain (rather than max at Bet Criterion $\ln(\text{gain})$). Hvr. for max expected gain, $k \to \infty$ (i.e. we bet as much as we can Borrow).

So for proper Bet about GR, we may want to use $k > 1$: Nothing Bet-theoretic $\leftarrow E \ln(\text{gain}) \downarrow$ $E(\text{gain}) \uparrow$ (for $k > 1$) — we may want $E(\text{gain})$ to be max, but w. some limits on probs of GR.

So Considerations of GR ordinarily are as important considerations of Gain or $\ln(\text{Gain})$ or whatever funct of Gain one is interested in.

The use of Gain maxim. is "Best" because of "Dutch Book" argt. . . . .

"Dutch Book" may not be relevant in real Mkt. & iteration, hvr. — including probs of GR.

0 — Perhaps $\max E(\ln(\text{Gain}))$ is ≈ $\max_{E(\text{gain})}$ if one considers one will be making Many bets. From a "Dynamic Pgms" pt. of view, $\max \ln(\text{Gain})$ for each bet

.32 may be best!

One way to be best : use k as large as it can be (>1 for large enuf B) à your probs of GR ab and of gn is ≥ a certain ←. (But no help args. at (.30 - .32)!) — for infinite yrs.

TM3 Rsrv Org evoln, 10

419.20-.26; 425.00-.09;

00 : (Spec 437.19) : TSQ of lrng defns of concs like solve integrate, differentiate, simplify : from Maple or MPearlstein.

02 Org Evoln. compared to (Somac or Bachtrack) 414.10 426.00; 427.00, 429.28, 439.00

02 Say TM has solid d-TSQ : How to use these macros as a beginning of A.2 S-TSQ : 437.20 is recent Note: Many earlier nodes.

04 Also Scan 420.02, 421.20 : 429.09 : 440.06, 437.20 : 414.10-415.40 : Possib Modifns in terms of various Meteor industry systems" to create "STRONG AI". 415.20-33 lists some industry systems. ART Life is Good Card.

07 Gathering h(T) functions for PD2 → Maybe related to 431.2? : 429.09-.19; 430.15-.33; (431.20ff, .23ff, .30ff)

08 435.00-.30

• 433.31-434.28 : A kind of understanding" of Courtois' Rewarding Carefully: I don't yet see "integrated view" in it.

10 437.19 Great Idea! : Try BZZ w degeneracy reduction on GA type problems: See 414.27-.30 for earlier version! GA has been used on. Try to find problems in which complete extract functions solved the problem. Perhaps Koza's books would help. Also look at the Rawlins Book on GP (by Goonan writers) — also other GA rawlins' books I know.

Rsrv 421.12-.18; 422.00-.29; 431.00; 438.00

I Dont Like to Code (implementation problems) in Perl 5.30 pp than summarize how Par'd'ise, Gottlieb w. Rochat/Tseng.

If would perhaps work fine for problems in which the eval'n. function took lots of time, so one could afford expensive ways to chose (candidate trials).

Another possy would be to try it on (Sparc) ANL. (Alg Notation/Lrng).

57 Our Procedural level: for SUMAC : Retain ~100 fairly disparate codes of corpus (so they are

19 at distance > D wrt BZZ) Use BZZ to searching "in ear" extensions of "codes Must Live".

20 .17-.18 by (also if doesn't do much : I want codes "close to" successful codes" that are not simply extensions of the old codes. One way to get this : say old codes are N bit strings. Use BZZ to make extensions of the codes that are of length ~N & try Rsrv out. Complete corpus.

Or, for Bachtrachy, [redacted] for all code length of N bits : Try extensions of the first N-k bits of old codes, using extensions of the N-bit corpus. (1)

I Guess T. problem of SUMAC update is (other than Bachtracking) : To make mathematical "close" to known, past successful corpus (2) have a type of Being accepted codes for augmented corpus. The R system (§ 8 of "Report") is one way to deal w. this.

30 Re: Augmentation of BZZ, using d-neuists! A neuist is used for predicting just as a neuron is used. However, Sander's Rsrv. have to agree on how far to step for suitable neuists. The further the steps, the longer it takes for coverage decoding — but we get more compression.

of next sub-sample of symbols

D.6.03

TMS

Rev .ooft

10: : [redacted] T. last MtgR u (399 to 436) has been summarizing some impt. ideas! would be Good to Review those pp. First do ruff page contents.

- 399.08 The idea of 2 PD's: +400.00 -.18 : 436.00 -.22 (latest on this.)
- 399.18 -.[redacted] on "FNZ of Sol 89".
- 400.37 Start of Discussn. of "Context" I think I finally came up w. a good understanding of it — Very General, very clear. — 435.05 -.27 was latest take" 435.19 wtd Nguns emerge from Context. possl. ← Great Breakthru.
  - So 2 impl. ideas: ① PD1, PD2 idea ② Context! Generl understanding of various levels of "context"
  - 433.31 - 434.28 [This to be an "integrated view" at this time.... revised, reorganized
  - 402.12 -.22 first impt listing of context types: 435.06 -.27 * Latest, listing of context types. Note
  - We can do this in 2 steps: d-Nguns (≡ d-language); Then Sngums (S-language ≡ Most general PD)
- TSQ's ; 403.00 -.16 early Ideas on TSQs: — later, Use to do sim in Maple; Inv interpolate, solve, differentiate, Graph... 438.00
- 404.50 - 405.11 ; 406.35 - 410.40 on 428.27 on Importance of symmetry of Pendulum (≡ "Degeneracy") ← T. general problem (using assoc. commonly distrib. laws) has probly been much work done in "Symb. Math"
  - 420.20 - .40 Amusg problem of rate of Growth of Simple symmetry.
  - 420.30 Gives recursion formula: 432.00 ÷ 40 work soon. 432½; 432¾.
  - 427.20 - .40
  - 432¾ .19, .20 Gives very simple recursion formulas. So R(n) is expressible as ratio of factorials ≈ 2^n
  - interesting math result on 432¾.32! → 439.08 → 442.00 The R(n) result has to be mult by some Prog like n! Note
  - If BZZ is improved it will on however comparison of sequential tests as well. (earlier) basic: possl. ordering of 6 things. (depending on how many duplicate terms)
- 406.31 (à probly better): T. idea of Using BZZ (a/o enhanced versions of it) for PDZ (or even PD1?) Versions of it 435.10 spec → 438.00

Also impt: Idea of BZZ

Symmetry: 2 recursion Formulas. 2. straight Rec. Mtn

Improving BZZ by Nguns (≡ d-language) & wtd nguns (≡ S-langs)

TS Q's usually Dealt in Maple, Mathematica etc

D.10.03 I was Concerned re how to use consensus in (Solns of MTM probs) for (Solns of MTM probs). (≡ d-funcls) (S-funcls.)

One way (perhaps): MTM may used d-nguns for preding MTM can generate Rsets to wtd nguns for S-prediction. Its Not clear as to just how a particular d-nguns is scaled to a S-nguns! just make it a

We could just make it a S-nguns w. = wts to all members = zero wt. for all others. Would Rule of much help in can have

S-preds?? Perhaps look at the degrees of the d-nguns. An nguns is a d-language: it a grammar

or [redacted] ("test" grammar) (Membership function among Rsets by Recog Grammars). A d-nguns / generative grammar can usually

be made stochastic by usually probing into n-choice parts. eg. $(N \to A;B) \Rightarrow N \xrightarrow[p=.7]{} B$ (p=.3 ⇒ A)

"tests" for strings can often be got in form of a real funct on strings w. a threshold of acceptance.

This can usually be converted to a probby distribn — p a threshold funct. or use $h(p)$, "Entropy measure"

Note that INV problems are often solved by converting them to OZ problems: T. yes/no criterion of the Problem definition is replaced by a "softer" soln condition xlating it to a OZ problem.

In similar ways, it should be possl. to replace a criterion for (d membership) by a nguns, by a "soft" criterion — that can be ≈ a S-nguns. So, this is a way to convert PST's for INV problems to S-nguns or S-Grammars or PD's.

Another Approach: Look at various Uses of Context in which nguns are useful: Then look at details of those nguns & see how a/ they can be generalized to S-functions,

425.10 was an attempt to get "S" Rfuncts for recogn. The Recogn("R") funcls of § 1 of "Report" are 426.24ff. 240 = accept. are

d-nguns (assoc. w. sets of Q's that Resp "accept", 413.03 Discusses Use of BZZ to get "R" funcls for Recogn. It gets

S-Rfuncts! — [redacted] Gets probby of various Continua.

D5.02

TM3

436

New Mexico
Botswana } 2000 ?
Bancoke } or 1999 ?
Budapest (Sept 1995)

$-40 \times \frac{9}{5} = -72 + 32 = -40$

: The "2 kinds of PD's" V.S. Periodic of the sample developing, changing during the career of the scientist. Does PD1 (comp) change? does PD2 change? Do both change?

One approach: As the experience of the scientist develops, his corpus (≡ TSQ) grows. The "prep" is Nant's Universal D.P. applied to the known corpus & hence pd for continuation. This assumes CBDA.

For finite CB's, the scientist has a "Summary Machine" (≡ Sumac) of each pt. in his history. The Sumac looks a bit like an app, but it is able to do "Back tracking" — So it is not exactly a Continuing "Sumac".

PD2 Guides "Lsrch" over the current prep & the current corpus. Note that the PD1-PD2 idea is a ruff model of human Problem Solving! — (It's for "Phase 1" only)

D7.03 399.17 (" ") PD1 is a Recursive Univ. D.P. — it is the P.D. on future continuations of the "known corpus" & is incomputable. PD2 is a method of approximating PD1 as a computable function. PD2 grows w.t. corpus, just as PD1 does. One such "method" is a PD guiding an Lsrch to find short codes for the Corpus, C_1

T' Corpus for PD2 can be regarded as part of the over-all-corpus of TM.

This PD2 corpus is closely related to a "current problem" that TM works on.

C_1 is the normal corpus, C_2 is the PD2 corpus. Perhaps the Corpus $C_1 \cup C_2$ should be used for all problems! [In general, the Complete trace of TM's activities should become part of C_2]

D10.03 C_2.] So: TM has been Solving ≥ TSQS! C_1 & C_2. We can coalesce them into a single TSQ; ① But we furnish "External" Context info That C_1 & C_2 are for ② But we furnish "External" Contexts info, That C_1 & C_2 are for: Somewhat Different Universes. This info to be in the form of INDEX's furnished by Trainer.

∴ At the start, TM keeps the 2 corpi separate, but later may decide to Merge Reinforcing degrees. Note that PD2 has a "special" corpus, in that it includes part of traces of TM, so it really has to be treated in a special way to prevent TM's "Illegal Self-Reenforcement"

D1:03

TM3

39744 colnos = 96 = 6×24 = 6 days: #24/day
414 colnos = #1.

4:36" 3:10P 4:36 2:34P

:00: 434.40 : 433.31 - 434.28 Gives a recount take on t. context ideas of 402.12-.22.

With t. "self context" idea of 434.33, this looks like a kind of (preliminary) understanding of context.

I do want to list various kinds of context & how to use/discover them.

[SN] There is a usual connection of Bayes Seq" & "Laps rule" in most kinds of implementable "context".

— So it's a very simple probability Model ... except that's contexts can be terribly complicated. — i.e. "Universal".

.06 So: starting w 402:12-.22;
a) raw token frequencies (no context, uncondl. probs) — But these can be derived. (New)
b) npms preceding t. token to be predicted as conditioners of pcpl next token

.10 2) npmsets (See 434.03-.14 for Discussn) : for "soft" npmsets see 434.33 -.40 — (The exact number & type of soft npmsets is to be determ'd.)

a) The R functions of S1 of Report (IDSR) are npmsets: (They are t. set of Q's assoc w. that R type) 425.10 ff is attempts to define probabilistic R's: 434.33-.40 on soft npmsets is relevant & maybe Much Bayesian 425.10 (425.10 is attempt on probabilistic R fncts.)

b) See (434.03-.14) for discn of defining, expanding, reducing npmsets (Analysis/Synthesis)

.15 3) Obvs Genzns: a) The "context" is usually adjacent to t. token to be predicted. Am impt

.16 Counter case is N DMM. "Raster" data. (Use of FFT to detect that kind of "closeness") .21

b) The "state" of a system producing all or part of t. corpus. — Say a finite state (or Finite state) Machine: Hidden Markov Models.

D2.03

.19 → 4) N.B. : T. wtd Npmset is t. most General kind of predictor (i.e. ≡ ALP) (or HMMs / HMM's)

It is t. most General kind of internal context. (But NOTE 446.22-.30)

.20 We need Good methods of deriving such npmsets (& their wts).

.12 .14 Contexts (like for Raster scan corpus) that have Gaps between t. relevant part of t. corpus & t. token to be predicted, (re 5:(3a)) can be represented by an npmset in which all npms are adjacent to t. token to be predicted.

Note: While the adequacy of a wtd npmset is important as a measure of how Universal means of representing Context, it is not t. only way. Any good wall definer of prodn. methods (e.g. Boolean w. definers of tokens, or BZP) can also be used.

.27

.30

f21.12 start of review of how Phrasal works.

N.B.: Avtd npmsets are most General predictor ∴ t. most General Context

TM3

Sage.context: Continued.

.00: 433.40: Ways to elz. context à gen. Phase elzns: Usual contexts (items adjacent to predicted Token).
One Generalz" = n-dim rasterscan, where "pts "near" to predicted token have to be found in
2 special way (FFT makes it easier to detect contexts of this sort).

.03 D1-03 Ngram contexts: expand to ngram set = (ngmst). This makes sse (in past) larger! — perhaps key impr.

ANALYSIS & SYNTHESIS: This matter is most important desideratum of a ngmst. — otherwise, if sse → ∞, it would be possl. to use individual ngm components of an ngmst. as contexts.
In general, we will be forming ngmsts & breaking them up into smaller ngmsts/or ngms.
This is the analysis/synthesis aspect of the progress of science.
Analysis = breaking up of ngmsts.; Synthesis is creation of NGmsts. This process
.10 continues w. the "Breaking Up" & "Synthesis" occurring different ways each time they are invoked.

.12 — — — We need techniques (functions) that do Synthesis & Analysis.
We try these functions on the data in (in a pipe) order, & see
.14 Which of them give compression (≡ prodn).

NB My original application of "Context" was to Modify the pc's of "Macros" (rather than Tokens). "Macros"
This is, of course a more general than Tokens, but is about the same if one has already defined
2 token to be a "Macro". Mainly, the original necessity of Context considerations for
.20 "Tokens" (≡ Macros) was the "Scaling" problem w. that w.o. context considerations
(i.e. uncondl. pc for all tokens/macros) — that pc's of goings to problems ↓ very
rapidly as one progressed in the TSQ.

The recent discn. of "Context" seems v.g. in sense of suggesting several impt techniques
to use in the phase I — maybe "more or less" adequate. I think the contexts discussed were
all d-facets, d-ngmsts. They are somewhat "s", since they have varied a prips &
.28 they obtain different pc's for prediction, depending on their past successes.
percentages of
→ I do want to review this in some detail — I don't want to forget any of it. Is 433.31-434.28 all of it?
.30 While context can predict any future confign., it is usually easiest to use contexts to predict
tokens (i.e. primitive or defined symbols).
402.12-.22 is an early discn. of 6 kinds of Contexts

.33 On "soft" ngmsts: A possl. (use/dark): We have an ngmst in which each possl ngm (of all possl ngms)
is given a pc. (which may or may not be summable to 1). These pc's given wts to be used in predicting
normalizable
a certain token (or set of tokens). To obtain such a set of pc's (wts.) do it backwards:
Use the corpus backwards & see (for each token) what the pd on "preceding" configns is. From a Bayesian
pt. of view, these pc's can be inverted to get the pc of the token in view of the past.
A "soft" ≡ wtd "ngmst is most gen. kind of Contexts, concept, PD → 435.19

IM3.....

.00 (spec 431.40)! TL. source(s) of f. data (Bayes) are stipulated to be mut. excl. pts for Y, Each w. its own a, M, σ^2.

.02 S. eq(9) of report looks reasonable. If there are no failures, we can get a = 1 à f. la Gaussian

.03 Gamma d.f. H. f. data. If there are no successes, we can/do get a = 0 à M, σ^2 are irrelevant.

■ .02 à .03 assume some kind of default apply for a, M, σ^2. With diffuse apriori, no successes in data would not give a=0. Also 100% success would not give a = 1. We might get f. value of a by STEINish Analysis — pooling lots of past data from other K, L pairs (430.21).

→ Perhaps we can just get our "a" values from Stein à get M à σ^2 by other, more conventional means.

STEIN is one way of dealing w. f. "zero frequency" problem.

.10 The Bayes. (w. STEIN) seems like a good way to deal w. the situation in which we have a bunch of K, L pairs (430.21) and we want to assign a good <a, M, σ^2> to each pair.

.12 Hvr, If we try different a different set of $[R^K_{pbm}]_s$ $[R^L_{pst}]_s$ functs, we get a different <a, M, σ^2> for each K, L pairs but (Pace will usually be a diffrnt no. of K, L pts) — NO!! — The f. no. of data pts is f. same. — How do we compare Bayesian fit, for Pace 2 situations? We end up w. a different no. of successes & failures for f. 2 diffrnt R functions sets. → No, f. no. of successes & failures remains invariant for Both sets of "K" functs. — This info is assoc. w. f. same set of empirical trials on

This means that we can multiply f. probys of success (to get a poly directly) by f. probys of failure, (which are actual probys) — T. resultant product is comparable to any other product of some no. of successes & failures. — This is ≈ f. Basic Update problem in Phase 2: (as earlier — by eqn of f.

So: .12 ff is very impt. when we are trying to fit a bunch of possible $\{R_{pbm}, R_{pst}\}$ functions to a batch of data on successes & failures of various trials.

- T. developement of .00 ff & its preceding section 430.10 — 40 Is a way to do some of Phase 2

.25 How to Generate $R^{z'}_{pbm,pst}$ cones & how to do f. search efficiently, remain to be discovered!

But (.25 R) can use parts of R functs obtained in solving f. d-problems of QATM-phase 1. — which may be a simple way to use MTM developements to help solve NMTM problems.

.30

.31 → Still very impt problems. How BZ2 discovers Contexts; What are limitations of contexts discoverable by BZ2 & how can we expand them? (ideally to Turing/universal status.)

.33 T. Turing/universal contexts: How does f. past (probabilistically) determine f. future? (or "context")

In BZ2 f. contexts are always Nems à count to f. token to be predicted. A simple context is an nems. — eg. T. previous token was an (even) integer. Another context/work be à state of a system (eg a Hidden Markov Model — That f. "state data" reader need not be in a finite state Machine — states"

.37 of a UMC are o/k. — in a TRMc, T. read head has "states". Also "context" can be more distant (as in raster scan

.38 → From .33: Any method of reading (future as f-function of past) defines a kind of "context".

SN Definitions reduce cc, but are only done when parsing is quite clear à BZ2 is fairly large. BZ2 will do OSL, but Defns are never f. final predictor in OSL. (since they obtain a specific parsing of corpus — that may be wrong.) → 434.00

≥ D.6.03
TM3

432 3

```
-C:\PB35\TM432.BAS
dim a##(6500)
a##(0)=1 : n=0
10 x##=0
for j=0 to n
x##=x##+a##(j)*a##(n-j)
next j
a##(n+1)=x##
print n+1,x##,6/(4-x##/a##(n))/(n+1)-1-1/(n+1)
if n=100 then end
n=n+1
goto 10
```

$$n+1 \qquad \frac{6}{\left(4 - a(n+1)/a(n)\right)(n+1)} - \left(1+\frac{1}{n+1}\right)$$

$$n+1 \to n \qquad \frac{6}{\left(4 - a(n)/a(n-1)\right)n} - \left(1+\frac{1}{n}\right) = 0$$

$$\frac{6}{(\quad)n} = 1+\frac{1}{n}$$

$$\frac{6}{4 - a(n)/a(n-1)} = n+1 \qquad 4 - \frac{2n}{2n-1} = \frac{6}{n+1}$$

$$\frac{2n}{2n-1} = 4 - \frac{6}{n+1} = \frac{4n+4-6}{n+1} = \frac{4n-2}{n+1} = 2\left(\frac{2n-1}{n+1}\right)$$

$$\frac{2n+1}{2n} = \frac{4(n+1)-2}{n+1+1} = \frac{4n+2}{n+2} = 2\left(\frac{2n+1}{n+2}\right)$$

4328.

$$\frac{2n+1}{2n} - \frac{4n+2}{n+2} \neq 0$$

T. pgm got zero for $n+1 =$ (1+4+37) then 42, 48

[ts likely to be exactly true for all n.

This gives a much faster way to compute the sequence.

$\frac{4n+2}{n+2}$ $\quad 2\frac{(2n+1)}{n+2}$ $\quad 2^m \frac{\prod_n^m (2n+1)}{\pi(n+2)}$ (product of odds.) So like 2^n

So we could write $a(n)$ as $\simeq 2^n \div (n+2)! \cdot 2 \times$ product of first n odd nos.

Product of first n odd nos $= (2n)! / 2^n \cdot n!$

so $a(n) \simeq \cdot \frac{2^n (2n+1)!}{n! \cdot (n+2)!}$

So $\prod_{n=1}^{m} \frac{4n-2}{n+1}$ is always an integer ($m \geq 1$). as $= \prod \frac{2n-2}{n+1}$; $\prod \frac{2n+1}{n+2}$ } second way unlikely!

does this really imply TM13 ? It looks like not

$\frac{2}{2} \cdot \frac{6}{3} \cdot \frac{10}{4} \quad \frac{12}{5}$

1 2 3 4

$\prod_{n=0}^{n} \frac{4n+2}{n+2}$ is an integer $n \geq 0$

$\frac{(4n-2)(4n+2)}{(n+1)(n+2)}$

$2 \cdot 2 \cdot \frac{(2n-1)(2n+1)}{(n+1)(n+2)}$

count no of even nos in num & in denom in $\prod \frac{2(2n+1)}{n+2} = \prod_n \left(4 \frac{2n+1}{2n+4}\right)$

$\frac{\pi\, 2(2n+1)}{\pi\, n+2}$ ... ?

→ 44200

D.6.05
TM3

67	2.203373E+37	3.911764
68	8.621893E+37	3.913043
69	3.374855E+38	3.914285
1	1	1
2	2	2
3	5	2.5
4	14	2.8
5	42	3
6	132	3.142857
7	429	3.25
8	1430	3.333333
9	4862	3.4
10	16796	3.454545
11	58786	3.5
12	208012	3.538462
13	742900	3.571429
14	2674440	3.6
15	9694845	3.625
16	3.535767E+7	3.647059
17	1.296448E+8	3.666667
18	4.776387E+8	3.684211
19	1.767263E+9	3.7
20	6.564121E+9	3.714285
21	2.446627E+10	3.727273

N 30.03

TM3 4PM yakuza : Ginza : N.3203 ... Sunday

427.40 — notagood!

196 230 280
14² 5.96 2.140
1.453 / 1.1710

00! 420.90! There are 2 series: one < R(n)! one > R(n).

The "<" seq. $A(2n+1) = (A(n))^2$

Ra ">" — no! — $B(2n+1) = (B(n))^2 \cdot (2n+1)$ ← No! It appears that B( ) is a lower bound to R as well!

$A_0 \equiv$ or say { 1, 3, 7, 15, 31 } = { successive values of $2^{n+1} - 1$ }

$\alpha \quad \alpha^2 \quad \alpha^4 \quad \alpha^8 \quad \ldots \quad \alpha^{2^{k-1}} = (\alpha^{2^k})^{\frac{1}{2}}$

i.e. the system forms, when it exists, is the smallest of $2n+1$ forms, for the cases of no n=4

k 0 1 2 3 4, 5

$n = 2^k - 1$ 0 1 3 7 15, 31

$A_1(k) \quad \alpha \cdot \alpha^2 \alpha^4 \alpha^8, \alpha^{16}, \alpha^{32} = \alpha^{(2^k-1)+1} = \alpha^{n+1}$

~~$B(k) = A(k) \cdot A$~~ no!

16

$B(0) = \beta$: $\beta \mid \beta^2 \mid \beta^4 \cdot \beta^3 \mid \beta^8 \cdot \beta^3 \cdot \beta \mid \beta^{16} \cdot \beta^4 \cdot \beta^2$

n = 0 1 | 3 | 7 | 15

It seems clear that the far larger and numbers of A seq. squaring is a much bigger source of growth than multiply $2n+1$.

I looked up, say 1 1 2 5 14 96 190 ... not. not (Google integer sequences) — Apparently it's not in it. First Prof Pray knows: (I fixed letting it write 1 "1" but it didn't help.

Is it related to the seq. giving no. of trees of n vars of depth k? (Which I factored into "closed form" in Szabo. (V(99)).

We know that R enjoys "Majorization" $X_n \le 1 X_{n-1} + 1 X_{n-2}$ | 26 $X_n = 1X_{n-1} + 1 \cdot X_{n-2} + 2 \cdot X_{n-3} + 5 \cdot X_{n-4} \ldots$

20 We can find the largest root of each assoc. char eq. before (over bound on R(n)).

$X^2 - X - 1$: $\frac{1 \pm \sqrt{1+4}}{2} = \frac{1}{2} \pm \frac{\sqrt{5}}{2} = 1.618$

$X^3 - X^2 - X - 2$ or do I have it backwards?

N = 68

D6.03: Dim A(100) B(100)

A(0) = 1 : N = 0

X=0: For j = 0 to N

X = X + A(j) * A(N-j)

Next j

A(N+1) = X : print N+1, X, X/A(N).

If N = 100 Then END

N = N+1 : GOTO X

overflow after N = 69 3.52 E38 / 3.9 14

Interactive on 4.20! 5+14 = 19 (not 21)!

Scrap 432½ for fruitless

! ± .37
± 307
± 4932

N = 200 5.122 E+5 3.97

N = 1000 9.17 E+597 3.994

2001 3.32 E+1199 3.997

4001 1.54 E+2405 3.9985

6501 1.056 E+3908 3.999077

30 ## loses 15 decimal precision
e.g. 10 emphasizes ## ± 4932

N	a(N)
6000	2.7 ... E+3606

$\frac{6}{(4 - a(n)/a(n-1))^n}$ For N > 6501 it blows up!

$\frac{6}{(4 - a(n)/a(n-1))^n}$ $-1 - \frac{1}{n+1} \approx 0$

$\pm \sim 10^{-16}$ $\sim 10^{-20}$ for $n < 20$

$\sim 10^{-19}$ if $n \ge 90$

$\sim 10^{-17}$ if $n > 100$

$1.00016 \ldots = 1 + \frac{1}{6n}$

Different objects $n = 1, 2, 4$

$\frac{6}{(\quad)} = 1 + \frac{1}{n+1}$ | $4 - \frac{a(n)}{a(n+1)} = \frac{6}{1 + \frac{1}{n+1}}$ $\quad \frac{a(n)}{a(n+1)} = 4 - \frac{6}{1 + \frac{1}{n+1}} = 4 - \frac{6n+6}{n+2}$

$= \frac{4n+8-6n-6}{n+2}$

If I did it at exact precision in Maple or Mathematica, it would probably be obvious $= \frac{2-2n}{n+2}$

all cases.

TM3

Electrat cost Ridgeway 4:30 pm. Back frc.
costs Rhona ~ 8.2 ¢/kwh.

430.40: So:: T. QB, is [redacted] t. MTM w. Ts q from dcdns in Maple → maybe hard MoPi problems, a good beginning (study problem) for MTM in general; & Phase 2 in particular?
Is it a good "study problem" for Engs. English?

SN How to change any d-funct into an S-funct! In each psm, there is a "next token" that is chosen. If there are 6 distinct tokens & d psm will generate 1 to 1 of the 6 choices each time. If we make a less sharp d.f. we get an S-funct. —— Not so easy! If we start w. a d-funct, Run [redacted] after the first choice (in the S-funct) of a token that is not that chosen by the d funct [redacted] Then for each subsequent choice of tokens, we can no longer use the choices in the d functions as a guide, since the all over sequence state of the Machine has changed. (Most certain kinds of function trees, on certain kinds of inputs, this is not so.)

Hwr, assoc w. each d function, we have a pc for each token, that was obtained when that d funct was discovered. One way is a "introdistance" function.

430.31 Prob'y the best way to deal w. this problem of S-funct construction/derivn:
Make [redacted] a tsg w. desired solns. Then Do Play the various kinds of S-TSG's:
Then gen. from this mix of solns. Then see how the tokens (& derivs, ect) of the [redacted] solns of d-problems can be used to Help construct solns to S-problems.
. It seems, like VG. idea. — V.G. Approach to Solns of S-Inductn problems

[redacted] . It is realized (a bit) in 430.15 ff ... (see also discussn leading to 430.15)
A Better way would be for me to write out solns ("h(T)" functions) & try to factor them
. a/o factor the reasoning Part (code to Run, so TM is able to (Find/search for) Run effectively (real) IF

23:430.29 Re: 430.15 – 29! Use the Rpsm funct to map the problems to a kpsdim space.
" " Rpst " " " " " " kps " "
Then use continuous funct [redacted] on the combined kpst+kps space
→ <Q, M, θ²> vector functions on Rest kpsdim space.

Also, "Would Multidim Scaling" (MDS) be useful? (I know little about MDS) — Maybe look in Google?

430.28 If αβ is reasonable! Then for each of the αβ cases one must have one of ssz!
I think for each of the αβ cases we must have at least one success & at least 1 failure.

Q: if the first success for each of the cases is at t_0, then can't Biz also date the failures for $t < t_0$. ??!! Go over eq.(9) p 16 of "Repeat" very carefully!
Remember that "h"-type distribns can have 2 different meanings: I think the meaning chosen for h(T) in eq(9) is density. Prob'y of first success at time T. Is (1-h) the write expression for the rvs ξ of eq(9)? Perhaps if I did write I wouldn't have to have both successes & failures cases!

Say we have a "case count" plot for a large no. of trials. | Both successes, failures. times
or cumulative no. of cases, as a function of times.

5800
433.00

M 28.03

T0M3

Electricity cost Rio Janeiro 4:30 PM. Back-fire.
Costs Roma ~ [redacted] 8.12 ¢/kwh.

.00 430.40: So:: T: Q3, is [redacted] t. MTM w. Tsq from "delas in Maple → many hard Math problems, a good beginning (study problem) for MTM in general; & Phase 2 in particular?
Is it a good "study problem" for Eng. English?

SN How to change only [redacted] d-function on S-funct: In each pson, there is a "next token" that is chosen. If there are 6 distinct tokens & d pson will give pc = 1/6 to 1 of the 6 choices each time. If we make a less sharp d.f. we get on S-funct. —— Not so easy! If we start w. a d-funct, then [redacted] after the first choice (for the S-funct) of a token that is not that chosen by the d funct [redacted] Then for each subsequent choice of tokens, we can no longer use the choices in the d functions as a guide, since the all over sequential state of the Machine has changed. (Not for certain kinds of function trees, or for certain kinds of algorithms. But is not so.)

.10 However, assoc w. each d function, we have a pc for each token, that was obtained when that d funct was discovered. One way is a "infodistance" function.
Infodistance

.14 430.31 Mayer [redacted] privily the best way to deal w. this problem of S-funct construction/derivation:
[redacted] a tsq w. desired solns. Then Do this for various kinds of S-TSP's:
Then learn from this mix of solns. Then see how the tokens (≡ decisions, each) of the [redacted] solns to d-problems can be used to help construct solns to S-problems.
.14 ff Seeing/Using VG. idea. — V.G. Approach to Solns of S-Industrial problems

.30 [redacted] . It is realized (a bit) in 430.15ff ... (see also discussn leading to 430.15)
A Better way would be for me to write out solns (≡ "h(T)" functions) & try to factor them & also factor the reasoning that led to them, so TM is able to (Find/Search for) them effectively (really)

.23: 43029 Re: 430.15 — 29! Use the Rpbm funct to map the problems to a (real) kpb dim space.
" " Rpst " " " " " " kps " "
Then use continuity of funct [redacted] on the combined kpb+kps space
→ ⟨a, μ, σ²⟩ vector from that space.

Also, "Wookie Multidim Scaling" be used? (I know little about "MDS") — Maybe look in Google?
MDS
Multi Dimensional Scaling

.30: 430.29 If αβ is assumed: Then for each of the αβ cases one must have one of 555!
I think however of the αβ cases we must have at least one success & at least one failure

→ Q: if the first success for each of the cases is at $t = T_0$, then can't Bz also date for failures for $t < T_0$? ?? !! Go over eq. (9) p 16 of "Repal" very carefully!
Remember that "h" — type distribns can have 2 different meanings: I think the meaning chosen for h(T) in Eq(9) is that of prob(t) density. of first success at time T. Is (1-h) the write expression for the rv. t of eq(9)? Perhaps if I did write it I wouldn't have to have both successes & failures cases!

Say we have a "case count" plot for a large no. of trials. | Both successes, failures: times & or cumulative no. of cases, as a function of times.

Spec 433.00

TMB

R(a)=1, R(1)= R(0)·R(0) =1 1420 on "Rmax problem".
R(2)= R0·R1 + R1·R0 = 2

Comes: Back at 1:30pm

431.10(+1) Way to Go Keep moving!

.00 : 429.90 : ON NMTM : S-Funcs: One way to implement Rsm w. "3IU" is to have 2 inputs define a f. discrete params of a [S-] function. — T "R" input (s) (Rsm can be >1 "R" input given. Continuous params of f. S. function. We use Mt Carlo method to get pc of corpus wrt - such a S-funct. If f. function has several continuous params (i.e. several "R" inputs) we get pc of corpus in same Mt Carlo way! It may be useful, sometimes to find r. actual peaks(s) à (their locs) w.i.2R(s). — say for hy precision pc's - like SM, acc .10 .06

HVr. note .10

So .00 could be one PST for "induction"; BZZ (different genre) can be another. So TM must learn to use f. write one in each problem.

.10 (06) A everpresent problem in (00) — fitting of Continuous params: The spread of those params — after Cancing other things) tells how many params to use. I guess one can usually get f. needed "spread" of continuous params by looking at f. past" If one can't, then perhaps one should'n't be doing Prodyn. in this Domain ☺. To do this for f. general problem of finding optimum curves w. continuous params: 431.10 .14

.15 : 429.19 : One (ruff) way to do this: we have a set of "R" funcs that look at problems, à la Rsm in a set of categories $[R^i_{PBM}]$ We have m R funcs for PST's. → $[R^j_{PST}]$; we can empirically get. $\langle a, \mu, \sigma^2 \rangle$ for each R^i_{PBM}, R^j_{PST} pairs by averaging over cases: At first f. two of R's in each of f. 2 categories will be quite small (~ 2 or 3 psts, - 3 or 4 pbm problem types) Rsm as (Tsp ss2) if we can have more.

.20 — Given a set of (P_{BM_i}, PST_j pairs, T) pts: To derive 2 sets of R funcs, $[R^k_{PBM}]$, $[R^q_{PST}]$ ∋ 2 $\langle a, \mu, \sigma^2 \rangle_{i,2}$ can be found that is a good predictor.

.21 Perhaps f. "h" eq worked (eq 9) of §2.2 of 30 Oct 2003 report. will be useful: we want

.24 to Maxz it. α = cardinality of $[R^k_{PBM}]$; β = cardinality of $[R^q_{PST}]$. s s' ss... s·s...

T. problem at .15 - .24 : Say we had

Then we have αβ pairs à αβ, 3 values $\langle a, \mu, \sigma^2 \rangle$. This is 3 αβ params — which is very large, so we'd have trouble optimizing (eq 9) w. that many params init. → 431.30 .28

.29 — If would be far better if find a way w. only, say 3(α+β) params, say! → 431.23

.30 A nice thing about .15 ff is that it is able to use f. R funcs à la sub-functions

.31 (that were obtained in f. QA induction of §2 of no part) to get useful S-funcs. → See 431.20

~ §N A (requires) esp, common way to create S-Rsm's: Take any S funct, or any 3IU (like BZZ or Az) à Threshold it. — This is ~ what is done in converting IRV problem to OZ problem. .33

34 : 429.35 : ON Generality of BZZ : Can it be used on 2 dim. raster pictures? Rissanen may have discussed this in his early paper on use of "context" for grammar. T. relevant unit of context could be discovered using f. autocorr of f. original signal. (or spectrum — which is changed by FFT) The auto corr for just a few values (for SM, say) is rather inexpensive. A 2D raster is equiv. to an RBUCS (see 448.16)

Bill

.00 (spec 428.99) cands. This effectively reduces # CJS. So the effect on search speed is a factor of >r but (usually probly) < r².

.02 425.02 : In Maple, etc, ! TM will learn what solve(x, expression), Integrate(x, expression).

N24.03 Mean: We can then go off into a seq. to teach how to solve/integrate progressively more difficult expressions. If this is done properly, TM should eventually be able to solve, integrate verbally difft. cases — perhaps beyond present state of symbolic Math.

.08 One Q is: How far can we go in Phase 1? Also: Can I put this MTM to solve more general kinds of Math problems or Arby difficulty?

.09 → A Big Q is: how to go from MTM problems of .02 Ff, to NMTM problems

.10 — In particular, the kinds of "s-funct" problems needed by Phase 2: — Involving parametrization of DEF [redacted] $h(\uparrow^{fsym})$ functions: $h(P_{arm_i}, PST_j, T)$. h can be verbl by 2 or 3 comm (pSym = problems = pSym)

.12 Continuous params: m, a (& often, but not nec, g_{rav}).

→ To do .10 it would be well for me to "try it myself" ("By hand"). It would seem that T: "R" funct of § (of "T. Report", (& Genens of Prom) (Szy #445.10)) would be useful have in exam telling what type of problem pSym is = "what kind of" PST, PST₂ is. Then, somehow, we have a funct that looks at the pair of categories, & finds if may to assign a_1, a_2 to them. This assignment could be done by purely statistical,

.19 Studies of pose category pairs. → 430.15

.20 Another approach to NMTM is to Make a TSQ for NMTM problems & try to find a TSQ of MTM problems that would be a good into to it. " "

.22 BZZ is an impt. kind of NMTM. Could we get TM to discover BZZ & genz/improve? The Genens of "Content" of 402.12-.22 are ways to Genz BZZ sd that it's perhaps as general as it could be ---- But I really need to do a lot of investigation before I can do this well. I have listed 4 common kinds of context there. There is a BIG JUMP to the General forms of S & G — Next has to be fill-in ☺. (Note 435.19 wid. Ngmsts are most general predictor. contrast.)

.28 : 427.10 426.12 SNOs Evoln: John Nash's work on "local equilibria" in Game Theory has been applied to Org evolution — in which pers determines to equilibrium pt. This

.30 analysis Might be relevant to my (understanding) analysis of SOMAC's (426.08)

.31 N25.03 GENERAL State of Project: If BZZ doesn't work well to implement phase 1, then, I will improve BZZ a/o use AZ or some other "compression" scheme. Similarly, T: redundancy reducer" may be easy or difft, but I can complete CJS, using any particular method for "redundancy reduction" (= "degeneracy evaln" = degen.evaln.) → Bigger type: modified CJS. : Major problems (not nec. in order of difficulty or "unsolvedness"):

.35 1) Does BZZ implement desires & other AZ tricks adequately? If not, fix it (Also see (.22)) → 430.34

2) Good ways to deal w. "degeneracy" → Also other Math problems of (.08)

→3) Good TSQ for MTM — Is long of "solve", "integrate" etc. (interesting?) (# to upgrade)

4) Link(3) to NMTM problems / TSQ's — in particular (.10)

tijeras = scissors

.00: 427.19 : If BZ2 did not do something like that, it would be a rather inefficient code.

.01 Say B = $b_1 b_2 b_3 b_4$; C = $c_1 c_2 c_3$. To code " B, C : sum of Pts is RPN, not Polish !)

$b_1 b_2 b_3 b_4 c_1 c_2 c_3$ + Sum : we code $b_1 b_2 b_3 b_4$, Pcon (endstring), Pcon $c_1 c_2 c_3$ (endstring) + Note .103 then Sum. ~~To code~~ we code $b_1 \cdots b_4 \Delta$ (endstring). To code c_1, use the contexts ($b_1 \cdots b_4 \Delta$; c_2 has ~~same~~ possible context $b_1 \cdots b_4 \Delta c_1$ acc to c_3 : "F" Δ Symbol has context $b_1 \cdots c_3$, & ~~then does what~~ "Sum" has context $b_1 \cdots b_4 \Delta c_1 \cdots c_3 \Delta$.

I think this implements OSL — e.g. it skips at the end of ~~the~~ Sum had accessed only one before, it would have its pc incremented over the current henry

.10 ~~—~~ accord before. .103 BN The pc of BAC should be same as pc of CAB — This would be True for CB=00, but for CB<00 it will often not be true.

How would it work if I coded it backward (Polish notation)?

? → If I did BZ2 on various Corpi — would doing Pcon Backward make any difference) ?

Try English, Try Lisp Expressions, "C" pgms, ect.

To code a seq. of characters, make a Basic file. each character followed by a Com. Pcode.

When I recursive subset : check that "Basic doesn't screw up" : look at Text codes.

On the problems 427.15 - .19 ; 428.00ff : I could just look at the Function trees that I want to code & devise a sequencial method of coding them, that is somewhat like BZ2

[If the "improved BZ2" has been programmed for speed/low memory use; I try to get copies & use its tricks to speed up my own pgms ;]

.20 The ideas of .00 - .10 on the Sequencial Coding of Trees, would seem to be fine here !

Since functions are represented by Trees, wouldn't a stack be a useful to store & represent, & code Pcon ? One normally uses a stack to do a Tree Stack. Here we have the tree that represents a function ; it also is a tree that represents our search over all pccs !

Function trees.

.27 SN On summary of function "Search" : By Not recognizing equivalent redundant ("degenerate") forms, we lose twice ① for redundance r, we waste time by testing something r times.

② If the redundance of the soln. is r, we ↑ its CTS by a factor of r by not considering redundance.

The 2 effects do not multiply exactly. I think the second effect is more important, since Search time ∝ CTS. ~~But the first effect also saves time~~ The first effect does not affect the stack, time, here. If the redundance of a terminal is r, & its row pc is p, & the take CC0 for a row test, Then we recognize redundance for 0, we spend time CC0 · r on that case. If we recognize the redundance, we only spend time CC0 on it — so we do save here. The fact that its pc has increased w. redundance recognition does not affect this result — we do test that cond. each time, but, because it took time pc.

→ Also, I think the normalization constraint is modified when we reject a bunch of r

Org. Evoln. (.00) ORG. EVOVUTION (.00) $P_M(x) = \sum_i 2^{-l(s_i,M)} = \sum 2^{-|s_i(x)|}$

.00 : spec 426.07 org. evolution : Org Evolution: Recently (on Internet ≈ ~~2 1/2~~ N20,03) : Evolution of microbes in 3 labs runs: Variations betw. 3 populations were "very small": Certain problems were solved in similarly different ways — I would expect more diversity ~~between~~ between 3 results

However, since the Environments did not change, the problems presented were minimal, (if at all). Some kinds of "probs" that might be tried: Slow/rapid changes of pH, temperature, nutrient composition, poisons, ...

The reason for lack of diversity between 3 populations: that the "problems" solved only had a few solns. By ~~not~~ trying a greater diversity of problems, more significant results

.10 might be obtained. However, since Envt. was constant, we had "neutral (mutations/variations)". → 429.28

1) **BZZ** : Redundancy : adequacy for representing subsets, sets

Say I'm looking for cases of "sum, mult, (·)". If "sum, mul" occurs. ("sum" is before "mul" (generally), I would miss it — So How do I deal w. this?

.15 2) General uncertainty of whether BZZ gets most regs obtained by "definitions".

e.g. (T) A, B, C — consider context of token "T": A, sum, C, B — 3 = context, but say "C" was a by pc definition ... i.e. it occurred frequently in the previous corpus.

BZZ would seem to not use this fact — which should ↑ compression.

.19 Well perhaps it does use this fact. B : C are coded as compactly as possible, Then sum B, C is coded → 428.00

.20 (420.AD) $R(n) \overset{\text{large } N}{=} 2\gamma R(n-1)\left(\frac{1}{k} + \frac{1}{k^2} + \frac{2}{k^3} + \frac{5}{k^4} + \frac{14}{k^5} \cdots\right)$.

$1 = 2 \times \frac{1}{k}\left(\cdots\right)$ ← this is true ~~always~~ $\propto n$ for large k !

$1 = 2 \times \frac{1}{k}\left(\frac{n}{2}\right)$

$\mathbf{1} = \frac{n}{n}$ so $k = 2$ — for large n — seems right for very large n,

but for small n, k seems to be ≈ 3. (!!). No!! $k \neq 3$ for large n; ~~But~~ $k = n$ was obtained by assuming k was constant for large n:

say $R(n)/R(n-1) \to n$ for large n, $R(n) = 2 \times \frac{R(n)}{n}\left(1 + \frac{1}{n} + \frac{1}{n(n-1)} + \frac{1}{n(n-2)} \cdots\right)$ ($\gamma = R(\text{mul})$)

Say $R(n) = k \cdot R(n-1)$ and k is a function of n. $R(n) = k\,R(n-1) = R(n-1)$

.30 $R(n) \approx \left(R\left(\frac{n}{2}\right)\right)^2 \cdot (n^2)$ $n = 2^k$ $R(k+1) = (R(k))^2 \cdot (2^k)$

try $R(n) \approx R\left(\frac{n}{2}\right)^2 \cdot n$ $n = 2^k$ $R(k+1) = R(n)^2 \cdot 2^k = \left(R(n) \cdot 2^{\frac{n}{2}}\right)^2$

$R(n+2) = \left(R(n)^2 \cdot 2^n\right)^2 \cdot 2^{2n+1} = R(n)^4 \cdot 2^{3n+1}$

exponents $R(k) = 2, 4, 8, \cdots$

" of 2 k, $3k+1$, $7k+3$, $15k+7$, $31k+15$

n, $(2^2-1)k + 2^{2-1}$, $(2^3-1)k + 2^{3-1}$, $(2^4-1)k + 2^{3-1}$,

Taken to root: ~~Recurrence~~ is

$r = R(0)$: r, r^2, r^4, r^8, r^{16} | $R(k+1) = R(k)^2 \cdot 2^k$ | $k=0$ $R(1) = R(0)^2 \cdot 2^0 = \frac{r^2 \cdot 2^0}{R(1)}$

$\to = \frac{r^4 \cdot 2^0 \cdot 2^1}{k=1}$ | $r^8 2^6 \cdot 2^2 = r^8 2^3$ | $k=2$ | $r^{16} 2^6 \cdot 2^3 = r^{16} 2^9$ | $k=3$

432.00

TM3

Org Evoln (.00) Adequacy of BZZ .20 Function of Backtracking .20 Adequacy Adequacy

.00: (Spec 414.31 : Fig.26; 425.09) I had this idea of Org. Evoln. w. a sudden gross Change of envt., leading to Mass Extinctions" — corresponding to a SUMAC Given a diff't. problem, so few of its old ~~cells~~ codes survive. | 3 org.ev. Expts.

In Org. evoln., t. remaining lifeforms have envt complexity in their evolutionary search, so it is essentially, ~~a~~ "Universal" If C. envt. changes slowly enuf, it

.07 → will regain its former diversity. → org.evoln 427.01

.08 In a good Sumac (≡ suitably designed Sumac), t. same sort of thing should

.09 occure: There should be envt backtracking (or Equivalent) after a hard problem

.10 to restore "diversity" (≡ no. of somewhat distinct solns to present total Corpus).

[A Sumac can also be designed so it only has to solve all the past problems — .21

.12 or, [SOMEHOW use an ~~a~~ window rather than a Rect. Window] → 429.28 → consumacs' org. evoln.

As a form of (EN): Global warming is a slow change in envt. — most ecological systems will adapt; Man will also adapt, but in his usual way of wars between coalitions. Due to the present & near future technology of war, this can be very devastating — perhaps Much more than Global warming.

.20: .09 "The Equiv. of Backtracking" in any technique has ~~the~~ Becomes diversity to an adequate level. BZZ may be able to do this! Essentially, t. Function of Backtracking to an important extent: t. Q is: "would it be "adequate"?" Compare Backtracking Methods & BZZ in effectiveness: Can we get cross fertilization from t. 2 technique types?

.24: (425.40 Spec N22.03) The function that looks at ~~the~~ (Q1, QA1 ... Qn Qn An) & proposes a pd on Ans+1 is t. most General/Internal/Context, & I have considered it — ... But it needs to be elab. ☺ — i.e. it's "too complex" — i.e. say we are looking for a ≤ n-ary that maps (Q1 QiA1 ... Qk Qk Aik, Qik+1) to Qik+1 — we put these arguments in as one of t. "initial arguments" for BZZ (say) & look for functions that map it to Ans+1 (probabilistically). This function

.30 must work for i = 1 to i = n. As stated, t. CJS would be far too large for n of practically any size ... but it may work for n = 1 or 2 or 3. Using Phase Functions as initial approaches are ways to get functions for n=4, then n=5, ect.

Any Elemental-za could furnish Sub-functions that could make t. job of .24-.32 more feasible. The Method I'm currently considering for doing Phases, involving token freqs, deltas (pairings) & BZZ is ~~a~~ a generalization of t. problem of .24ff

N20.03

TM3

TSQI idea: V.G.! .00 → But note .06

Genl, CONTEXT: RESOLN .37 (Internal)

Ceno "suppm."

37.37 / 48.37$ v.> 54.54 B

42.5 / 11 p 420 23.6 conj 5.3 prot 5. fiber / 4 d.g

.00 **:4926 :** I [illegible] of TM learning to define of various expressns in Maple, Macsyma, Mathematica, "Irng" ... as a good beginning of a TSQ. On a more Practical level, doing those defns, would

.020 **Train Me** a lot in TSQ construction! — So: **V.G. Idea!** Much More! → (429.02)

The Qs are MTM problems. They can be a basis for NMTM problems as well (see 420.02 - .19)

.06 → One Q is known to find NMTM probs but are reasonable combins of MTM problems — i.e. [redacted] 420.18-.19!

.07 I May Interject Mix: Find impt NMTM problems), then look for MTM problems that use same.

Maybe "R" functions of §1 of "Turoport" would be a good example: Tho I haven't yet defined

.09 probabilistic R's, it would seem like a V.G. Idea to do so.

.10 A Quirk Attempt! A R (probabilistic ∃ s. funct.) is a pd on all Q's, w. peak values on certain set of

.11 Q's. — An immediate problem! Do we want a p.d. (normalizable) or probs values that don't have to be normalizable? — Could non-normalizable probs of this kind be applied to selections like (uniform from $K=-\infty$ to $K=+\infty$) all possl.?

Anyway: Also assoc w. each R funct, is a pd on P(A|Q) distributions. [redacted]

.15 Again, I'm not sure I want (or need) this pd to be unnormalizable.

.16 **T. Why** these R functs are used: A Q_{Any} comes in: All R functs have wts. (probs) for that Q_{Any} & also have wts for all $P(A|Q_{Any})$. From this info, we get a bunch of wtd. P_i functs & so we can get a probs def. over all A by using a suitably wtd

.19 mix over all the $P_i(A|Q_{Any})$.

.20 From a practical pt of view, the computations of .16-.19 may take too long so we have to use Approxns.

Re: the normzn/non normzn of .11-.15: The pd's might be normzd but the R's, (I'm not sure this is right.) so $\sum R_i(Q) = 1$. for each possl Q. This makes the R categories Mut. exclusive ← (??)

Or there may be some other way in which $R_i(Q)$ [redacted] can be normzd.

I wonder about Fuzzy (Zadeh) sets. Are they "normalizable"? — Maybe not: If so, they are different from probs in an essential way.

.30 (424.37 spec) On the 4 d-functions, R), as examples of Contexts that are sets of Ngms, w. simple defns.

I was thinking of each R as being defined by the set of Q_i that it "accepted" — This is a peculiar kind of "context"! — But it is a legit "context" since it "Conditions" a set of $P_i(A|Q)$-functs.

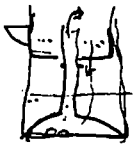
[N21.03] I may want to allow the 4 d-R functs to overlap — so any "input" is a legit context [redacted]

.35 [N22.03] On the General Q of How to implement various kinds of context implied by 402.19 (≡ the full Univl. set of Contexts), ← These are (so far) only "internal" contexts.

.37 That 402.12-.16 Don't nearly cover the set of cases implied by 402.19, means that there is a large Universe of Contexts that is never yet to discover.... So in teaching TM (How to Trn. TSQ is), I will be able to employ new Context types. → 426.24 spec

(435.19 more on our less solved this problem.)

426.24 spec



.00 (Spec 423.40) ! "In 2 kinds of PD": PD1 [redacted] is more apt to succeed to Strings & sets of codes that decrib. scenes to all of. problems Rust to. In summary, we try to find codes of hyperp Rust decrib. corpus. "2 kinds of PD" is one way to do this — by using a gradually modified PD_2 ($\neq PD_1$ usually) to guide Modified "Lstrch" operation. The "closest to previous trials" is obtained by Bzz w. $O_1 \cdots O_n$ corpus. This selection of trials, while still "TUniversal" is also heavily biased toward to past. However, this bias is "right" enough so that (personally) Specific "Backtracking" is not needed.

.10 N2003 In summary w. a finite world of codes & reference, there is a sharp division (cutoff) between "stored" & "not stored" — which makes explicit backtracking necessary ... since the code needed for the new examples (hrfc) may not be in the stored set of codes. However, when, using Bzz, the set of codes stored is "soft", all possible codes are "included" to some extent. An open Q is, "Is the "Closeness Criterion" used by Bzz using the corpus $O_1 \cdots O_n$ a good "Closeness Criterion" "? It can never exclude the desirable codes for $(QA)_{hrfc}$, but we can guess that they very low pc's ..... Somehow, we want this latter tragedy to be very unlikely. — In general, (the "Gardner's Rule" we can never be sure to avoid it!

.19 In General, the problem of generating the best possible PD2 (which is the "stored" set of codes) is a very important problem. Bzz w. corpus $O_1 \cdots O_n$ is

.20 one Guess.

SN Re: (.19 - .20) one way to "adjust" the "sharpness" of PD2's "peaking" about the $O_1 \cdots O_n$ corpus is to sharpen or flatten the individual Bernoulli D.F.'s (within Bzz / that constitute Bzz). (This method is not 100% perfect, but does move [redacted] in the right direction ☺ — one easy way to do it is to ↑ sizes of denominators of pc values. It amounts to stronger a priori equality of all tokens — a larger Sz for that "equality" ☺ — Another possible way is to take the nth roots of all final pc values (a renormalize, unless renormalization is automatic — which it often is) ← (the nth root may be very expensive look for cheap approximations where accuracy is not needed),

.30 Bzz to Weaknesses / Bottlenecks: 423.25, .27 seem most clever of possibilities. Sever

This is about contexts that are the OR of a set of Nems. 405.31, .39ff I think 405.31-.40 is about the only useful writing I've done on this: [& perhaps not very useful!]. I introduced the idea that ordinary, time numerical time series, Linear (& N.L.) regression was example. — But I think this/last was more like a PST. The R's of § of "Report" is an example. The R's relate a class of anums (= Q_i's) to a value PD on the strings that map $Q \to A$.

Clearly The R classes have sharp edges. Some function R's would be more general, .37 pressly more realistic — maybe easier to update (?). → 425.30

.38 ↳ See 425.10 for a quick attempt at "probilistic R functs.

TM3

.00 (spec 422.40): So the Q is: Can we implement 422.38-.40 by a suitable TSG ... or is it best to

"Tell" TM [redacted] (roughly) how to solve the problem by putting the O's

.02 into the [redacted] "R" form. [redacted]

Perhaps think about .00-.02 a bit: There may be other and better ways.

One way: Start out using TR system & get a bunch of O's:

Then just use BZZ w. corpus of those O's to get PD2.

Similar technique to [redacted] 420.02-07. A possible trouble: If there are

.10 only O's in the corpus produced R system, it bears to use the R system

subsequently, will be very hy (No never 1008): If we like, we can (illegally)

give the O's corpus w. R's loss Wt. Permit should have, to encourage "originality":

—There must be a legal way to do this! ....

.15 :422.11 19.03 Present bottlenecks: (Agm!) (1) 422.10 I'd like a clearer understanding of relationship

of "pure" context (BZZ) & "definitions". Are they really almost equiv.? Or could it

be that defns are actually more "correct" (better compression) looks likely? Defns. require

Specify (often completely) the parsing — which may be a big restriction. Maybe completely the parsing is ambiguous use all present w/ll.

.20 It looks like context includes delimiters plus other as does in ((So perhaps the main

(or only) advantage of defns, is lower CC (which can ultimately translate to

hyper pc, by effectively ↑ in CB [redacted] ← Evaluating this trade off & how

.24 & when to do it, sounds difficult (Maybe unsolvable in b. any exact sense)

.25 (422.20; 422.14): This is one aspect of "context" that may not cover the full general use of context of (402.12-22, 402.19 in particular.

.28 SN While BZZ may be a "power" equivalent to "defns" of simple & multiple (.25)

Sort, the pc's of "definition" may be different. To what extent do these differing pc's

.30 of definition" depend on initial choice of a pipe?

.31 SN Re: "2 kinds of PD's" V.S. "SUMAC" model of probabilistic knowledge Aquisition

In Sumac, w. each new problem/ som the approach is modified to include info contributed by that Problem Soln.

If CB=>, the system inherits prior best pass( ) results. For finite CB, there are many possible

ways to proceed. One way is to start by retaining the best 100 codes for the couple situations would find in

limited time. When a new problem is given, these 100 codes may be reduced to 10, 1 or none.

In any case, we have to search the codes "close" to the ones that formerly Worked also, or if

any known codes solve the problem, look for codes "close" to them, that also solve the problem

keeping the "diversity level" at 100 (at least).

N17·03

TM3

.00 : 421.40 : Introducing Definitions commits one to a certain parsing of t. corpus.
≅ BZZ uses all possl. parsings (Exhaust) — so it is certainly better in that respect.
On t. other hand Definitions markedly reduce t. no. of terms in t. corpus, so Searching is much faster. To compromise: we can use definitions, but only after they have their SSZ, & it is clear that they give very large pc ↑ int. [Even so, it may well be that we would want to retain alternative parses as well codes (or maybe alternate O²'s?)

Is this "summary detection, utilization" a new kind of Meer? (418.39)? Is this kind of "Quick Abort"? (Preconception abort ☺) Or, this is a way of Modifying t. PD2 int. Sense of changing order of trials.

.10 WEAKNESS 1 At present it is not altogether clear to me that BZZ "covers" definitions completely (other than t-CC aspect) → 423.15 This is v.g.

.12 ② Another (seems stretcher) weakness of BZZ is it doesn't properly include

.13 contexts that are like "OR" of a set of AGMs. An example of that kind of context is

.14 linear (& most non-linear) time series prediction (Also, the "R" function of §1 of "Report"... 405.31 -.40 for another dozen. 405.34ff → 423.2:

.15 ③ → A third (More Marco Gavrilov) weakness of BZZ in this application (421.19R)

PD2 doesn't look at $\bullet[Q_j A_j]_1^{n+1}$ in devising a PD for O_{n+1}. I think

(non) linear regression does look at, but these techniques are more like PSTs than PD's!

Perhaps "weakness 3" is one part of t. Character of Phase 1

.20 Disc. of weaknesses: ② (#2) can perhaps be fixed (if it is true & if it is serious). ③ Looks VERY SERIOUS. ← Actually not serious. How far one can go w.o. directly looking

.22 at t. problem is unclear. BUT we put in O^2, the responsibility of looking at t. problem

.23 & deciding what to do. — So essentially what we are doing is looking for a class of functions that can do this. Our Inductive Corpus, is a set of O_j that have been successful in this kind of problem in my past for smaller parts of t. Corpus.

.29

.30 One move in direction of .22 is to: "R" system for QATM of ~ §1.2 of "Report".

The R's look at t. problem & decide what class of P's to use.

→ This is a particular form for O^2's. The "R" system & Generalizations of it are of Much import in Phase 1

It's probly necy to have a good understanding of .22-.29 when designing TSQ's.

→ It is possl. that my view of .22-.29 ③ (.15ff) is not really a legit weakness of BZZ !!

I think I Got involved w. this argument some time ago & finally realized what was going on! — It took quite a while to figure it out at that time.

.38 N18·03 One [redacted] (non-CC) way to Deal w. t. problem of .22-.29 is to simply look for O^2's that do what is needed, and hope t. system can find suitable solns. — which will

.40 automatically include solns. for .22-.29 fairly.

TM3

REV .12–.18, .25–.40; 422.00–.40 ← USEFUL Review of IMPT Recent Ideas & present status of "Phase 1".

.08 : 420.40 ! I want to go over t. ideas of Context again: See how I'm using it in a general a way as poss'l. — see how Prof BZ3 is doing it properly: (Not 420.07 rely on Context)
325.12–.15 is an (apparently) much different sense of "context"! 322.31–323.18 | Note 321.00–.09
323.11–.18 removes some interesting ideas on entered t. general Phase 1 problem

→ Review this 322.31 ff. stuff carefully: It was interested in putting the O^i cands in Gen order for [redacted] (presumably) a Backtrack srch. I need to re-familiarize myself w. these ideas. In view of my use of (≈) BZZ, is [an effective equivalent of "Backtrack" automatic & perhaps rather good?

.10 Balance t. 322.31 ff. right against my more recent ideas on how Phase 1 should work.

.12 A Quick Summary of T. most Recent ideas on How I want Phase 1 (of QATM) to work! (Not so Quick!)

.13 QATMS Goal is to Maxz $\sum_i z_i \prod_{j=1}^{m} O^i(A_j/Q_j)$: ... To find a set of O^i's ⇒ [redacted] is Max. as t. app of the O^i are obtained by AZ or by BZZ or ≈ BZZ. [redacted] T. α is a function t. $[O^i]$ such and constitutes PD1 ... which we want to Maxz.

.18 PD2 is t. Probly of an O^i in view of t. previous sets of O^i's that have been "acceptable" (.25)

.19

.20 : 420.10 : This idea of using initial MTMTSQ as springboard for NMTM!
This may also work in a way for an RTM → QATM or optimizing TM. (More "controllable" than RTM at advanced level). This presumes Prof t. RTM has structures N to or reducible to cones, contexts, conditional pc's of or "Analogous" ← Too General? (concepts tokens NGMs)

.18R SN That PD2 doesn't look as t. problem that O_{n+1} has to solve, suggests that it is a not up. Method. to srch for O_{n+1}

.25 : (.18) → [redacted] PD2 may also include any "external" or "internal" context that t. trainer wants to include. Concievably, the solved problems set associ w. each of the past O^i's is part of t. context, but for Phase 1, we will not include it.

PD2 is used to search for O^i to maxz α of (.13R).

.30 ≈ BZZ is used to obtain PD2 [redacted] (which is a D.f. on O_{n+1}, a new problem)

.31 T. Corpus of ≈ BZZ is $\bigcup_{j=1}^{n \text{ AND } d} \Delta O_j$: Δ is a punctuation betw O_j's.

The corpus of .31 is a linear string. The structures of t. O_j's have been ("decanonicalized" "desymmetrized") by the methods of 418.26 – 419.18 ... Note 419.37–.38 on "How to do it"). The O_j's [redacted] that are found also have Prem pc's ↑ by 419.00–.18.

≈ BZZ doesn't use definitions as such. It uses pure compression + a "Huffman code" [redacted]'s probably sounds ((coding that could ↑ pc .... We also use symmetry (418.226–419.18) to give effective some effects of ((coding.

⇒ 422.00

N 16.03

MATH PROBLEM! (.20)

TM3.

420

Get AL Roll on A1 Dish, plate
~~to Row Thumb tokens.~~
~~Major Glue 2~~

.00: 419.40: w. t. "shortcut" of 419.34 since probly will not know if a given TSQ has enuf ESS for t. needed contexts, to be "an adequate TSQ".

.02 SN ~ 350.00 − 15 = .30 if argues that a MTM TSQ is not ordinarily a Directly useful introduction to

.03 a NMTM TSQ — T. first uses 2 IU; T. second uses 3 IU. However, TT value

of t. MTM soln's TSQ is that its PD2 EMESUS concs & contexts (over pc's such

.06 can be used directly for NMTM's. However, can we use concs, contexts designed for 2IU & use them in 3IU sets? .09

.07 → Some impt ideas on "Context" in PD2: 333.05,.09 Refto 327.21 :(322.31-328.18)

.09 (.06) Details of .06 R: Any subtrees in functions of always args. can be used in

.10 — funcls of 3 args (subtrees are "contexts") — ~~No. in part of t. input function Rule~~ → 421.20

~~does not refer to t. 3rd input ("R") is OK.~~

~~Whoops! I forgot! Recheck! Any subtree of a 2IU can be used in a 3IU~~

While subtrees of 3IU's are not usable for 2 IU constructions if ~~t~~ t. Process → subtrees contain refs to t. 3rd input ("R"), R is fact is not referred to (MTM → NMTM, 2IU → 3IU).

453
+ 140
+ 2×46
+ 5×14

.18 → At present, I can't think of a ~~NMTM~~ problem that would be a good continuation

.19 of a MTM TSQ. → See 425.06-.07 paranoidas

14+5+4+5+14

1125 14

.20 SN Q: What is t. redundancy of t. Sum of n variables? In polish, we need n-1 "sum" symbols, and n variable symbols. The rest symbols must be vars. The vars can be any n or of

n-1 "sums" & n-2 vars. No. t. first symbol must be "sum"

1325
1510

S VV
n-2 s's; n-2 V's

At no point can we have more W's than S's. (cumulated)

Another approach: A trace of n-1 s's. Each S can have come out of it either 0, 1 or 2 s's.

Missing pp! 303, 304

Another approach: recursive soln.

.30 $R(k) = \sum_{i=0}^{k} R(k-i)\cdot R(i)$?

K = n-1

For practical use, we just compute R(k) & make a table for TCM to use.

NB R(n) has to be Mult by n! or (n-1)! to include ordering of n variables.

N.23.03 If asymptotic ratio exists for n→∞! Set ratio = k:

$$1 + \frac{1}{k} + \frac{2}{k^2} + \frac{5}{k^3} + \frac{14}{k^4} + \frac{42}{k^5} + \frac{132}{k^6} + \frac{429}{k^7} + \frac{1430}{k^8} + \frac{4862}{k^9} + \frac{16796}{k^{10}} = k$$

$$\frac{1}{k} + \frac{1}{k^2} + \frac{2}{k^3} + \frac{5}{k^4} + \frac{14}{k^5} + \frac{42}{k^6} + \frac{132}{k^7} + \frac{429}{k^8} + \frac{1430}{k^9} = 1$$

R(0) = 1 R(1) = 1 R(2) = (2)

R(3) = R(2)·R(0) + R(1)·R(1) + R(0)·R(2)

R(3) = 5 | 2 + 1 + 2 = 5

R(4) = R3·R0 + R2·R1 + R1·R2 + R0·R3

= 5·1 + 2·1 + 2·1 + 5·1

R(4) = 14

R(5) = 14 + 5 + 4 + 5 + 14 = 42

R(6) = (42 + 14 + 10)×2 = 132

R(7) = (132 + 46 + 28)×2 + 25 = 429

The seq. is defined by R(0) = 1, $R(N+1) = \sum_{i=0}^{N} R(N-i)\,R(i)$

If R(1) = α, R(2) is mult by $\alpha^2 = \alpha^{2!}$

R(3) is mult by $\alpha^6 = \alpha^{3!}$, R(4) mult by $\alpha^{24} = \alpha^{4!}$

R(n) is mult by $\alpha^{n!}$

R(n) = (2n previous entries) + 1 → 3 → → → 1

S S S S S

R(6) = (453 + 140 + 2×46 + 5×14)×2 = 1510

I think Ratio of R(n)/R(n-1) → 4 (now proved R(∞) = 0)

→ 432.00

TSQ writing: a new Task: (20)

403.00 is also on TSQ

.00 : 418.40 : Re: the repetition of symmetric functions identities of 418.26ff: They have 2 effects: (1) The contextual repetition of trials (2) The pc's of those trials are not/primarily summed into PD_1. If a function has k equiv. codes — we should test only one code & give that code a wt. of k. So the equivalence detection techniques of 418.20–.34 should be modified to mult. the wts. of each equivalence form by the multiplicity ("redundancy") of that form.

So the effects of 418.20–.34 have 2 major effects: (1) reduces search time

.10 (2) Gives much better pc's (≡ values for PD_i).

If a redundancy occurs in PD_2, it will multiply by k, the amount of time we'll spend generating & testing that case.

→ Of the 2 redundancy effects, in both PD_1 & PD_2, the most important of the effects may be in multiplying pc by the redundancy, k.

It is notable that both effects of symmetry are important even if the expression of interest

.18 occurs only one time in the corpus. If it occurs n times, its effect is ~~...~~ → k^n

.20 SN TSQ writing: A nice set of cases to have, would be the measures of various expression types in LISP, MAPLE, MACSYMA, MATHEMATICA, etc.

eg.: Eval( ); Solve(X,); Simplify( ) ?■: differentiation, integration, etc.

dot product, cross product, Matrix×vector, Matrix×Matrix . . .

.26 Solve (linear, non-linear) → (linear) Run non-linear.

→ I will have to analyse these 3 cases to see what derive an adequate → 425.00

Conc net (& associated helping Contexts) — To an important extent. BZ2 does consider contexts — But I may need "external contexts" or other kinds of context that BZ2 is unable to furnish. Remember that "Generalized Context" is any aux info that causes the pc of the next token to be a conditional rather than unconditional probability

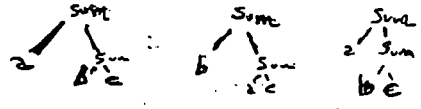
405.31 is one kind of generalization of "context": 402.12–.22 gives various Context types, 401.15 is a useful discussion of context

.34 It is tempting to try a "reasonable TSQ", using a fixed PD_1 & PD_2 both evaluated by BZ2, & hope that there is some "context" index to facilitate "learnability", but, it is more reasonable to try to pose specific contexts that I know of, that would be adequate for the TSQ. → Not so easy to do!

One of the important things in knowing whether a context is adequate, is knowing what its uses are.

(spec 4.11.26)

.00: 417.40 : On the commutativity problem; Consider t. sum problem a+b+c ! sum a sum bc ; b sum ac ; c sum ab
We all [agree] — giving a 3 way redundancy! So the Lexical order trick will solving that problem!



(A difrnt # track; we are relating t. functions in ≈ rec. order.
not so difrnt!
say sum a sum bc occurs early in t. function, then sum a sub bc is more likely to occur later.
" sum b sum ac " " " " " " sub b sum ac " " "
.10 — but sum a sum bc will not terminate falsely;
Ti reply will be that we can't [write P.D. quotes all functions] but use # recreate identical
functions many times. ← (True, but not now")

So: if sum a sub bc occurs, how to prevent sub b some c & some c, sum a b occurring. ?
sum a sub bc perhaps convert immediately to $\widetilde{\text{sum}}$ a bc ; $\widetilde{\text{sum}}$ is a over any finite # of objects
sum a sum , → $\widetilde{\text{sum}}$ a (·)(·) It occurs right start w. sum', we convert
1 2
.20 — More generally if any sort of "sum" starts w. "sum" we do a conversion to $\widetilde{\text{sum}}$; in which it
falls how many subty start w. sum. Well! — perhaps .20 is a way to decide w. some
kinds of symmetry. As soon as certain sequs occur (in t. generation of functions)
— Ti seq. is immediately rewritten.

In generating a function, each token goes on t. stack, along w. certain "state" variable
Values. When we encounter a variable, we pop. t. stack a few times...

.26 SN There are other equivalences that we may want to realize in our function generation;
t. associativity of addition sim a sub bc = sum sum a bc. (Pair is replaced by $\widetilde{\text{sum}}$ a bc)
distrib of mult. mul a sum a b = sum mul ab mul ac
.30 The trick of .20 - mine always rewrite a string into distributed form. This can lead to long
expressions, but it does make equivalences clear.

There are many symmetries — associativity distributivity in Logical functions
(like AND, OR, NOT, IF = (IF a b c) implies (choice function)). Must may be able to be
.34 dealt w. by ≈ .20 ff. Also Note TRIG Identities (Identity) 3 kinds of common Identities

(N15'05) Two impt ideas! (1) Ti idea that provable equality should be no. of terms.
In present case, this is done by writing all expressions in "standard forms".
.37 → This problem probably has been solved in "Symbolic Math. Technology" etc.
.38 → Macsyma, Reduce, Mathematica.
.39 (2) This looks like a new type of flow: (1) Big one.
(2) Quick-A look.
(3) Changing PDs

(3 kinds of Commutative Equality Axioms same thing (3) New type of flow.)

N14.03

3TM

.00:416.40! $c(i,T)$ is flat for small i, but drops rapidly for ~~[illegible]~~ $i > \alpha T$. (α is some constant)

.01 For "study" c can $= 0$ for $i > \alpha T$, but more generally c decays more slowly on T.

If c decays like $\frac{1}{i}$ [for $i > \alpha T$] then for most p_i, $\sum_i p_i\, c(i,T)$ will converge.

c includes a normalization constant so → not only converges, but sums to 1.

$c(i,T) =$ ~~[illegible]~~ $\frac{N(T)}{\alpha T + i}$ | $N(T)$ is a normalization const.

This $c(i,T)$ will work for most p_i's. We ~~may~~ may want $c(i,T)$ to ↓ more or less slowly w. T ... depending on the particular p_i d.f.

If $p_i = \frac{1}{i}$, say ($i = 1, 2 \ldots \infty$) then $c(i,T)$ can ~~[illegible]~~ decrease

.10 much more slowly than $\frac{1}{i}$ for large i: $\frac{1}{(\ln(i))^2}$ would be o.k. & other

decays like $\frac{1}{\ln(i)\cdot(\ln\ln i)^2}$ or whatever could be used.

In general, if we know the divergence rate of $\sum p_i$ we can design a $c(i,T)$ that causes $\sum_i p_i\, c(i,T)$ to converge w.r.t. i as slowly (or as rapidly) as we like.

The simplest $c(i,T)$ function is the sharp cutoff of [illegible] $(.01L)$

In Linguistics: in a language, new words are invented as [illegible] — but the situation w. corpi is different. We have a corpus of N words selected at

.20 "random" from a "very large" corpus. The no. of different words in the corpus will be a function of N.

For each value of N, we have a sample. By noting how the distribution varies w. N, perhaps we can guess at the D.f. for $N = \infty$.

There may be enough info in those Books I have on ~~freqs~~ of words in English text,

.26 to figure this out. Type-Token Mathematics: Herdan; Lang. as Choices & Chance | Computational analysis of present day Amer. English

---

.27 Perhaps use ~~BZZ~~ w. GA populations corpus to create new trials.

The mechanics of using BZZ on an "unordered corpus" [→ unordered set of prod/soln "], would have to be worked out.

One way: Each cand (i.e. each "case" of ~~[illegible]~~ actual trial) starts w. symbol Δ.

.30 we look at corpus to get ~~[illegible]~~ d.f. of tokens for next symbol — which is "a" say. Then

Obtained Monte Carlo-wise. Using "Δa" ~~[illegible]~~ as context, we obtain next symbol "Monte Carlo" & etc.

Formally, we ~~[illegible]~~ can probably do BZZ or close to it: But from a practical pt. of

.33 view, it may not be so easy! One possibly easy ~~way~~ BZZ: Use as "sequential corpus," all cases, bounded by Δ's. → we could use ~~[illegible]~~ to ~~[illegible]~~ Monte Carlo Generate

Cands. We may be able to give wts on various cases (depending on their ~~[illegible]~~ Goes (≡ fitness fcn))

→ (.33 R) ff sound similar (if not identical) to my plans to use BZZ in Phase 1. → as 406.31 ff

(re: 406.31 ff ... sumpower & ~~[illegible]~~ analysis of "Lexical order" to ≈ 411.05 ff. → {438.10 for More on BZZ in GA

N 13·03

3TM

4-16

.00 (speed 4:15:37) Re: Problem 4.15.37 : A passy: Look at present problem. from past problems select (neighbor one (or more)) closest to it. From population at soln. of Pn+1 mark neighbor, move back to earlier populations: 4. And "mark back". Iterate for neighbors that are further away.

.05 Use same Mut, cross as last neighbor.

Cases where this will not work much? Typically we have a linear filter design problem. Say the only previous "cases" are filter designs but for much

.10 different optim criteria — so we expect solns to be much different.

Another approach: Say we know as say of $[Q_i; A_i]_{1...n}$ pairs obtained by GA. Use some kind of math induction to get a very broad Anti d.f. from Q_{n+1}:

This broad D.f. is the initial population for soln.

Here Q_i decribs the ith problem in terms of its "fitness func".

.17 → Another quite different approach to QA inductively via GA!

We are looking for 1 (or more) $O_g \rightarrow \ni : \prod_{j=1}^{n} O_j(A_j | Q_j) = \max$ —

An induction problem. We get solns for small n, via GA, then retain populations

.20 of "n" for soln of "n+1". This "soln" looks very much like a SUMAC soln.

The Backtrack capability is obtained by retaining in the population, several solns à approx solns for several earlier problems. (How far back Reasonproblems are depends on the "Backtrack depth" one wants to consider).

N 14·03 Discussion of GA: A prominent diffy is apparently extreme inefficiency. First, we must assume all trials have bounded cc. We can get this by outlawing "recursion" or "loops" in pgms — this cuts down the "Universality" a lot (!!).

Using Monte Carlo mode — W. p_i as probty of a trial i is random but $\propto p_i$

.30 is probty of success of that trial, — in which case is execution time = Rest of rate such.

On the other hand, p_i the guiding such freq. need not be the p_i of the a cond's, being a soln. It could be $\sqrt{p_i}$ or ... — which is sort of optimum, but is correct choice to converge than p_i

Hey! If p_i is probty of soln. is not use of $\sqrt{p_i}$ for search, how can we do this if $\leq \sqrt{p_i}$ converges? (I don't know how to do a Monte Carlo or p_i where probty simulates Σp_i convergence) One (not so good) way is to have the probty distribn. vary w. each trial: so, say $p_i = f(t_i)(t=1,2,\ldots\infty)$. Given $f(t,t') \rightarrow k(t)$ then $\tilde{p}_i$ ($\sum_i \tilde{p}_i = 1$)

Say $f(t,t) = \tilde{p}_i \cdot C(i,T)$ ($C(i,T)$ is a "convergence" function) so $\sum_i \tilde{p}_i C(i,T)$ converges

"Convergent"

3 TM

MATURE INDUCTION SYSTEMS : "STRONG AI" (17)

Hat
→ BICMRB ←
Foods of world ?
Dead (OH).

.00: 414.90 of having several Children.

Forsch If one has to solve a problem: One wins money if one solves it: One loses if one doesn't. At "feeding time" one is given money: = Δ + one's fortune to use for Sold Group for having Children. (One's fortune can be <0).

Why have Children, & Sold Group? (See Interaction & Life & reasons).

Anyway, there would be 2 TSP's (a) T. subroutines, fights for like U.S. companies
(b) T. Government problem-solving TSP.
T. interaction of these 2 TSP's would be of much interest.

TM System is Minded of W. Hillis's compartmentalized GA's to solve a problem
.10 in Computer design.

Some reservations on Forgy: One could have parasitic Framework for problem solvers to feed off, from each Other!

(2) How much interaction would there be between facilities in an organism for trying out predictors/parasites & facilities for working problems?

(3) We would want something corresponding to "publishing" that would encourage individuals to (share/distribute) knowledge.

.17 More generally, we have various "Mature production systems": Now can we use One or more of them to construct a "Strong AI"?

Some Mature Systems:

.20
1) GA, Sim. Annealing.
2) Art. Life
3) ≈ "BZZ"
4) X (fm from 1 loop to another (on LARGE corpus).
5) Hidden Markov Models?
6) ANN (& RNN?)
7) ~ SVM (support vector machines) for categorizer (Vapnik) — Going to hyperdimensional for N.L. separation.
8) Tree (branch) selection for categorizer (i.e. ID3)
9) Linear/nonLinear Regression.
10) Theorem Provers ?
.30 11) Nonlinear Optim. for Reals. — continuous
12) L sych ?
13) Univ. D.F's
.33 14) ~~Clustering~~

Mature systems:
1) TSP ideas
2) MT (mach xltn)
3) HMM — because initial selection of states is not Mechanized & (?)
4) RNN (crazy slow).
5) GA's & variations
6) ANN

We might start by asking how each & lacks full Strong AI capability, then see if one or more of these technologies might fit.

Well (1) GA: Given problem: If it's a INV problem, a good fitness function has to be designed. For an Opt problem (can be inverse & / bdd: accuracy). Note often need to modify modify fitness funct.

.37 & Selection of initial population, Darwin mut, & Crossover rules. → (Spec 41600)

Org Evoln: (.18)

.00: 413.40 Not included in t. "definition" routine: Now, it appears that definns of PC's of tokens are all varieties of "context"; that BZZ should be able to deal w. Precondtns fairly well — along w. definitions?

.10 SN I had this idea of "SUMACS w. Backtrack". ~~With~~ with $C_B = 0$ we don't need backtrack. W. $C_B < \infty$, we have a certain no of solns for each pt. in TSQ. Anyway w.o. Backtrack, we start out w. K solns, & w. each new problem, t. no of solns ↓ (dim. K ↓) until it → ϕ ... no Soln. (or 1000)

However we deal w. this is Best if at any pt. we have $\leq$ say 100/solns, we

.17 backtrack until t. no. of solns is ≥ 100 (or 1000)

.18 In Organic Evolution we have a similar situation. We have a certain diversity/kinds population (an ecology in a forest). A "problem" is ~~an~~ external

.20 modification of t. ecology. If modifn is very large, we have "Mass Extinction" of species. If no often big changes occurs, t. no. of species will slowly ↑ to perhaps its pre-catastrophic level.

Normally, the external ecology changes slowly enough to Best adaptation and cross-over can maintain diversity level.

If diversity level is low & a "Big External Mod of ecology occurs" —

.35 it could destroy all life — for perhaps a long time.

.10 - .17 is about Sumacs; .18 - .38 is about Org evoln. It is interesting

.30 to compare how they deal w. new problems. A "Mass extinction" corresponds

.31 to a "very difficult problem" in Sumac.....

→ (426.00)

There are several systems that can be usefully compared to t. above 2.

Art Life, GA, in various forms,

Could we modify ALife so it would be designed to solve problems of increasing difficulty: i.e. a TSQ? One way might be to have a set of organisms! Each organism can recognize & solve one or more types of problems. It's a kind of "PANDAMONIUM". For solving a problem, an organism gets to participate

3 TM How to make R recognizers via BZZ (SVMs / ID3) (.03-.15)

60 x 250

60 k/q = 15k

52 x 5 = 260 wkdays/yr.

.00! 412.40 : models to see what problems arrise, & how they can be dealt w.

(Remember this is Phase1, & Prediction doesn't have to be VERY good — only good

.02 enuf to Get to Phase 2.) → .22

.03 SN In Devising R recognizers; TM is best Given indicates that it tries to associate w. & particular R's. Later to do "R" categorza, we have this corpus of

cases in which we know R(ite R's. We can try standard Categorza

pgms like ID3 (or SVM's (Support Vector Machines)) (If I can make hardware run this case) or use BZZ to categorize

in a soft way — like Li & Vitanyi did, by framing on known corpus &

.10 assing & additional coding cost of various possl. affixing to that Corpus.

Thereby say each R_i has a corpus $[\Phi_i]_{n_i}$. We train on each corpus $[\Phi_i^n]_n$ & see how much extra code len (⇒ PC) is added when we add this new Φ to each of the $[\Phi_i^n]$ corpi. Thus gives a relative PC of belonging

.15 → to each of the R_i's

The Q is: Are the veggys recognized by BZZ good enuf to give Good R's?

→ Also note: in ~ § 1.2 of this report (on updating R's) Various Methods of updating are discussed & possibly BZZ could be adapted to Bayesian-Merge of Prem

.20 — That section also has updating of P touchems assocs w. R's — & BZZ ought to be tried there as well.

.22: (.02) — The idea of 412.36 - 413.02 is not bad. It may be that several of the unpleasant attacks of 412.02 to 412.29 really don't have

much attack, & that ones of much attack can be fixed — perhaps w. an A.M. "soln". I could just examine some of the attacks, in detail, w/ BZZ or pc's of definitions.

NI2.05 — BZZ does have mask (?) of the veggys of AZ in it; so if there are Bugs in that applcn,

.30 Pete would suggest Bugs in my planned implementation w. AZ.

I suspect that the "corpus" should only have changed O_i's because the

Situation where we need a new O_3 is one of change! — Also,

the small changes of analog params is audio mode, we don't need

$O_2 \ldots O_n$ corpus Guidance for it. — So don't put Pete in $O_1 \ldots O_n$ corpus.

So superficially, it looks like ~ BZZ may be able to work Phase1 a bit!

I assume here that it is able to find a PC ($\equiv O_i$) for a single R (message) funct.

I previously had the idea that there were special (local) contexts, that were

JV 11·03

3TM

412

.00 : 411.90 : SN A notable difrnce betw. PD_1 & PD_2 ? PD_2 will usually have many O_i's in its corpus: Usually they will differ only a little from one another. PD_1 will have a much smaller, more group, corpus. | No! PD_1 has n, $Q_i A_i$ pairs into corpus. PD_2 has n $Q_i A_i O_i$ triplets (sort of) into corpus.

.02 → Q: Should we consider in PD_2's corpus every O_i that occurs — even when it does not change? ~~Usually~~ We could have a kind of (PD_1) corpus!

(a) Eg for Algebra: Many problems w. short random nos as args or few probs w. long random nos as args. In (a) case, many O_i's, usually the same. In (b), every problem (usually) gives a new O_i. ~~Example~~ In (a) we will usually have several "incorrect" O_i's until the correct O_i gets almost all wt.

Very local changes

.10 In the case of analog prediction, should we count small trivial changes in continuous params of O_i as a new addition to the corpus.

Should we just consider changed ~~wts~~ O_i's to be in PD_2's corpus?

.15 What about cases in which we have many solns for O_i, each w. its own wt?

Run 5/8 mi

.15 may cast needed light! If we consider all O_i's that were found, each w. its own wt ($\in PD_1$), we could consider all cases that arise (?) It contains lower ~~wt~~ wt. solns that could be used for Back Tracking!

10' ≈ 5 m
km
~.62 mi
.619

.20 — Just how to use the .15 info is unclear! Perhaps like a GA Population!

Well, The wts of the various O_i are (almost) directly usable as sample wts; But Big Problem is that each O_i gets lower & lower wt as e_i, q_i!!

3 1/3 ft = 1 meter
3.333 ft / 5250 ft

Maybe normalize ~~abbr~~ the wts for each i? But what wt. to associate w. each O_i set? Keep the Eye on the Goal: The prodn of O_{n+1} of max wt.

.26 One way: Have exponentially ↓ wts as one goes into the past. Exponentially "adaptive". Many diffs w.26! Will repeat things of a problem class

.29 "increase" in wt. w.r.t. wt. into the past?

.30 T. General Problem Here is: Given the O_i seq. (with or w.o. the $Q_i A_i$'s) To get good ~~estimates~~ (guesses/estimates) of O_{n+1}. There are Some that look at Q_{n+1} & A_{n+1} and derive O_{n+1} using Plotinfo.

Hotel Aurola Holiday Inn

Fill out paper

11:50
78.50 ?
Paris

Say we did know about sequences of identical O_i's — perhaps including info on different ~~Q's for them~~ Q_i, A_i's for them.

.36 One way to deal w. this: Try different modes of dealing w. the problem e.g. Use only O_i's that ~~are~~ are different from previous (O_{i-1})

Use all O_i's; Use O_i's in which the difference betw O_{i-1} & O_i is > a certain amount. ——— Then try BZZ (or equiv) on each of these amount.

font

.00 : I do want to look at the "slow" statistical predictors for compression, to see just how much better they are ... just what additional redys they can detect.

NB A common commutative extension on many inputs, is additional multiplicn. args. We can just put R. args in Lex order $A \leq B \leq C \leq D \ldots$

.05 Q: Does BZ2 Do any thing better than beyond AZ? I was thinking of BZ2 as doing t. searching magic for good coding — But it only works for sequential codes — what I've been able to do is usefully apply it to Polish formulas.

.10 How can it be applied to QATM (Phase 1)?

[Perhaps it is usefully applicable to $\{PD_{23}\}$: which is (I think) sequential induction. The thing to be coded is retrieved codes $O_1, O_2 \ldots O_n$. This is the reduction of a sequence of Polish formulae. So viewed in this way, BZ2 may be a "BIG DEAL" in getting TM off the Ground!

In working w. Polish formulas, it would be a good idea to use a "STACK" — Not only the evaln, but for applys & for constraints on symmetry.

The sequence $O_1, O_2 \ldots O_n$ does not take t. assoc. problems into account. This is not a catastrophic criticism, but it would be nice if we could somehow include in t. abstr. t. problems assoc. w. t. O_i.

.20 t. sequence $\langle O_i, Q_i, A_i \rangle_{i=1-n}$ has t. info, but would BZ2 be able to use t. info? Would it be "good enuf" for Phase 1, not to include QA info? — I think so: theoretical discussion this Q.

The R functions in § 1.2 of t. 2nd IDSIA report do implement a context of t. problem itself (Q_i). — Could BZ2 be applied to recogn. of such contexts? My impression is that t. R's are

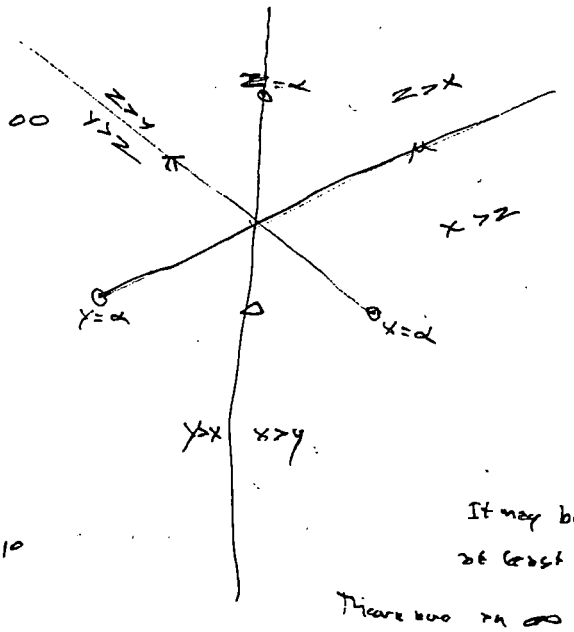
.26 a distinct kind of context from that which BZ2 works w. R. → 9:18.00

.30 The recent foray into symmetric funcls & t. use of BZ2 in Phase 1 of QATM started around 405:3) in a discussn of whether definitions were nasty or good or convenient because of speed or .... Could one do very well W.O. them.

Question (1) T. kind of context at 405.31 is t. kind used in BZ2. I was comparing that kind of Context w. that of 405.34: A different kind of class of Token Configns.

Question (2) T. Q was: what was really tricky of these 2 kinds of Contexts?

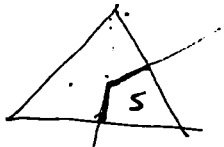
In the "R" funcls of Phase 1 QATM are one example of classes of configns of Tokens. — In this case: classes of Q's.



These 3 lines (actually planes) Λ, Π, Δ each divide space in to 2 parts.

region R is defined by

$X>Z$ $Y>Z$: in R, flipping X, Y leaves it invariant.



region S is defined by

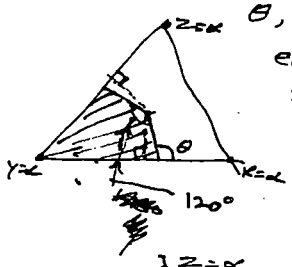
$X>Z$

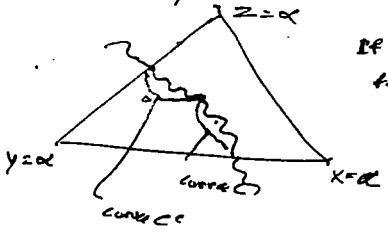
$X>Y$

in S flipping Z, Y leaves it invariant.

It may be that if we have rotation symmetry, then we have at least one flip symmetry exists (which one can choose).

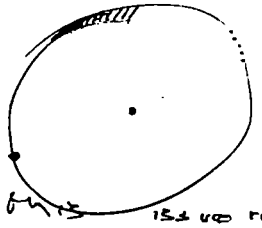
There are ∞ many of rotationally symmetric shapes: Eg. for any value of θ, the radial arcs in movements and will give rotational symmetry, but for each θ, certain flips will give invariances, others will not. In the figure (15°) some Z, X flips leave it invariant, others don't.





If curves < or > are rotationally but 120° displaced, then no rotational symmetry for the respective intersection point.

A circle is perhaps a better space for/study of rots, flips:

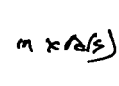


Well, I don't want to spend more time on this issue right now.

There are very probably mathematicians who know about symmetry groups (say in knots) that could answer all Q's directly!

So far, to get rot. symmetry we needed for 3 arcs, one could do $A \subset B$, $A \subset C$ then — this also gives that A, B flip & A, C flip symmetries, but not B, C flip symmetry.

Whether there exist functions w. rot symmetry and not certain partial flip symmetries — is our question. $A \subset B \subset C$ gives like only 8 of these!



Anyway; for functs w. 2 args (which is all we will have time for) $A \subset B$ is quite fine & will enable us to show all functions have a unique representation in Polish! So we can use ZB2 or whatever — or better "skeletal" functions" to detect regularities.

N·9·03

409

3 TM

.00 : A better way! We want to region that is

Above line γ $x+y < \alpha$

below line π $x+2y > \alpha$

below line μ. $y+2x > \alpha$

$y+2z > \alpha$

Since $\alpha \geq x+y+z$

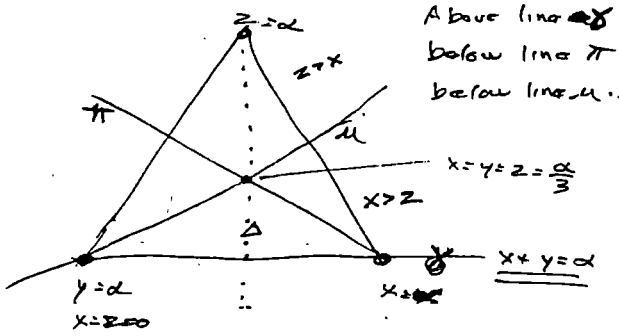
$x+y < x+y+z$ so $0 < z$

$x+2y > x+y+z$

$y > z$

$x+2z > x+y+z$

$\therefore z > y$!?



.10 equation of line μ : $x+y+z=\alpha$; $x=z$

$y+2x=\alpha$

or $y+2z=\alpha$

line π : $x+y+z=\alpha$

$y=z$

$x+2y=\alpha$

or $x+2z=\alpha$

$x+2y=x+y+z$

$y=z$

$x+2z=x+y+z$

$z=y$

$x+2z > x+y+z$

$z > y$

$z=\alpha$, $x=y=0$ is above π & μ

$x+2y=0$ at that pt. so $x+2y<\alpha$

---

line π is $y=z$ we want the side where $y=\alpha$; $x,z \geq 0$.

i.e. $\boxed{y>z}$

line μ → $\boxed{x>z}$

.20 line γ $z>0$ ✓ (looks ok. on (408.26L))

The condition $z>0$ is probably slightly wrong (from graph of (408.26L)).

$x>z$, $y>z$ look like $x>z$, $y>z$ literally; since we maps Cox arbitrarily into integer ordering.

This amounts to $B>A$, $C>A$ literally (probably $B \geq A$, $C \geq A$ is correct)

Since we want "on or below"

$B \geq A$, $C \geq A$ seems to give wrong results; i.e. 408.cc

.30 The constraint $B \geq A$ was to ensure a flip symmetry; i.e. $ABC = BAC$.

Two flip symmetries gave a rotation, so that gave all 6 symmetries.

Maybe the $B \geq A$ & $C \geq A$ constraints don't combine that way!

Well, say $B \geq A$ & $C \geq A$. $A\,B\,C$ If we rotate, to! $B\,C\,A$, it violates constraint

$C\,A\,B$ constraint.

If we flip AaB : BAC violates constraint — so we don't have soln. to problem

Quick check: the Geometric Soln. Seems to be Correct!

So Paradox !

N 7:03

3TM

.00 : By [redacted] constraint $A \le B \le C$ (lex really) we make all permutations equivt.

However to [redacted] make $ABC \leftrightarrow BAC$ is illegal; just do $A \le B$ lexically.

So we can get any single flip by lexical constraint.

To make all rotations equivt, but not any flips. How to do this?

.05 What about [redacted] lex constraints. $A \le B$; $A \le C$. ? Note: 2 flips one a rotation

Actually there are more than 6 permutations! Add

AAB; ABB, ABA, AAA; We assume A, B, C are distinct.

So a total of 10 possys; w. 3 inputs.

.10 There are [redacted] 5 possl. symmetries 3 flips; rotations;

4 possys: Lexical $A \le B \le C$ gives all 6 permutations equivalent.

$A \le B$; $B \le C$, $A \le C$ give 3 flip-equivalencies.

6 permutations equivt. all 2 equivt. Like "A+B+C"

How to get rot. symmetries?

$A \le B$ $C \le A$ so [redacted] $C \le A \le B$

A B C

B C A

C A B

A possl way to deal w. this: [redacted] want to prevent rotary symmetries from occurring in our [redacted] trials for A B C. Whenever a trial occurs, we write out the permutations of that trial that we don't want to try (assume).

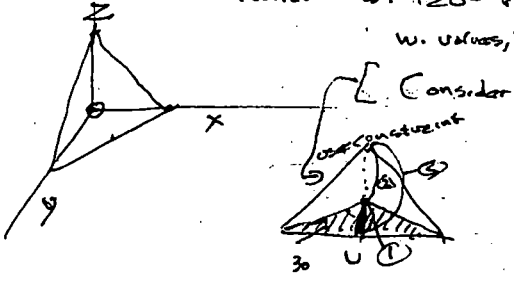
.20

Could we do it with HASH coding?

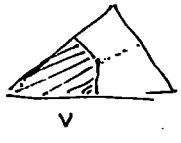
Think of 3 axis in 3 space. Functions of x, y, z

Functions invariant w. 120° rotation about line $x = y = z$. We only have to "fill in" w. values, one third of the space. How do we characterize that space?

.26 [Consider the slices, $x + y + z =$ [redacted] or. ⇒ △



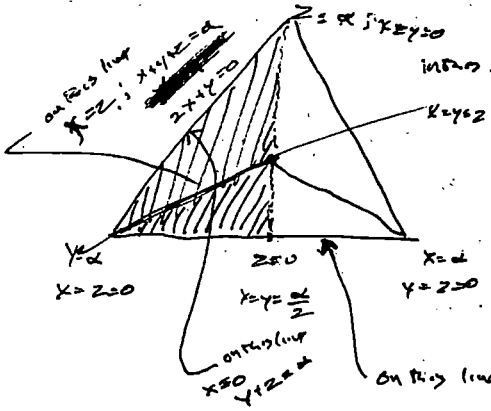
or



I think there are an ∞ of ways to draw the boundaries.

.30

If A, B, C can be represented by integers, this analysis should work.



Inside Δ, $x+y+z=\alpha$

$x=y=z=\frac{\alpha}{3}$

∴ so we want opt. below $x+y=\alpha$ & $2x+2y=\alpha$

In red Δ:

$x < \frac{\alpha}{2}$

There are 6 △ & △ altogether.

etc.

How to say opt. is in one of these!

N.9.03
3TM

.00: 406.90 : Must be in Lexical Order.

Listing functions of Pairs sort in pc order: sounds difficult, but perhaps isn't.

After "Sum 3" has been written, the only legal symbols that can follow, are lexically 3 or later.

Can I use this trick for other symmetries in functions?

For order of args a,b,c being irrelevant, this with inverse trick can deal w. it:

.07 What about $f(a,b,c,d) = f(c,d,a,b)$? I.e. certain permutations etc. ABC

.08 args is leave f. funct invariant. ← How must be lexically ≥ ab ! BAC

What about general permutations? The idea is that if several permutations are equiv., then only one of those permutations is legal. (.07-.08 is an example — but can I generalize it to all possible permutations?).

.10

Remember, AP-permutations form a group. Any subset of permutations form a subgroup.

.13 STUDY of Permutations: Say ABC, CAB are equiv; then all powers of this permutation

.14 must be equiv; i.e. BCA. To define this A < B < C lexically.

I seem able to deal w. "special cases": But I'm not sure that a gen'l. rule (yet). Also e.g. in .13 ff, if I'm not sure that the gen'l. I've proposed (but too restrictive — so it may outlaw cases I want to have.

I.e. There are 3 equiv. permutations, b/c a total of 6 possl. permutations:

.20 e.g. BAC, ACB, BCA are not in equivalence sets of (ABC, BCA, CAB) for reasons

A possl Approach. Show that all symmetric funcs can be expressed as functions of symmetric 2 input functions.

In .20, there are 2 symmetry elements: each forms a subgroup of 3 elements — Because 2 subgroups are indep. If we put these 2 symmetry elements one group, they generate all 6 elements: No! A rotation element: applied 3 times, generates 3 permutations. Then, if we flip AB (in ABC) we get BAC; & operation then applied again → ABC; so all 3 reflection operations are self inverses.

.30 So we have 4 subgroups: ① & rotation group ①②③ & ④ & interchange or reflection subgroups.

2 flip operators BAC & ACB. Each operator gen. CAB which is a rotation, — See Philosophy applying do — BCA → CBA. 2/so BCA " " rotation.

Which is the other flip operator: so any pair of flips generates entire set.

Any flip & any rot generators is a valid set.

Summary each flip generates 2 members & is a self inverse

each rot " 3 members.

Any flip & any of two operators (flip or rot) generates all 6

Any combin of rotors generators only 3 distinct values.

N.8.03

3TM

.00: 405.40 to Answer 405.36-40, I have to detail the ways in which these contexts influence pc's of Tokens (or configs of Tokens (≡ngrams))

.04 405. SN Also, I'm not so sure about 405.12-.30 — (Recall it's possible to have useful Contexts, w.o. using "definitions") — i.e. To use Contexts for adequate predn, w.o. ever using "Defns" — yet getting as good & usually better pc's. → see 423.15-.24: Avg. Standard analysis →

.06 N9.03 First I have to review use of defns in predn. My old idea of how to do defns may be a bit incorrect: i.e. I probably didn't analyze the pcost of defining a token, properly! It has to do w. "punctuation cost". I was using a standard simple model for Punctn cost, but I think it's more complex. About the simplest kind of punctuation is the "comma"... which tells where word boundaries are. There is the $2^{-\log^* n}$ pc of the integer, n — so we could put integers between each word to tell how long they are.

Another way is the "End" Symbol, Δ. Its pc is (mean word length)$^{-1}$. If ↓ the pc per token by the factor $(1-\frac{1}{n})$, n being mean word length, pc of Δ is $\frac{1}{n}$.

More generally, if we have a stock of tokens w. pc's = [pc's], we can insert the comma symbol Δ to separate tokens. We can assign a pc to Δ that maximizes the pc of the corpus. More generally given the [pc's] stock, how can we design a prefix code w. max mean pc of messages? $bc_i \equiv -\log_2 pc_i$. So $pc_i = 2^{-bc_i}$.

.30 Here's a better way to code the messages into codes (or maybe not put in legal), using uniform words of (key) length L, so token i has just $pc_i \cdot 2^L$ words assigned to it.

T. result is how punctuation seems to cost so little!!

SN Does this punctuation cost occur in some way in "HMC problem" (Hierarchy Code?)? Well, the Poppel discussion is for a known [pc's], for "like" steady state situation. T. Q of interest is not that, but cost of defining new symbols. Is this prop by EC (many at ell) "on symbols w. zero freq. Cluster" relevant? Maybe our conclusion may be: Next within one can study freqs of known symbols. If a new symbol had a freq. of ε, then its probby of not having occurred until now, is $\approx \frac{1}{\varepsilon}$ of ε has a certain value. We can make rules for computing ε that maximize pc of corpus. (possibly params of rules, can be selected to make pc of many different past corpora.

.31 SN BZ2 (or whatever it's name — i.e. v.fr Compression code for Text (strings) or they mention statistical methods that are somewhat better, but much slower). These are for Text strings. Could I arrange the corpus codes so that refrs would be always reflected in linear string compress. codes?

.35 One thing I'd have to decide w. is the commutativity of certain functions: e.g. 405.33-405.11 A B sum ≡ B A sum are equivalent strings. Perhaps for commutative functions, args are always written in Lexical Order! Using Polish (or Reverse Polish RPN) might make it easier:

As soon as we write the name of a (commutative) function, we know that its args

3TM

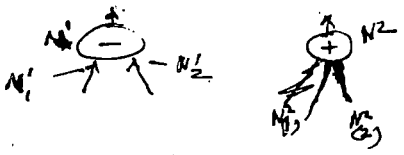
OSL(.21,22)

$\ln 1.5 = 1 + .5 - \frac{.5^2}{2} = 1.375 + \frac{.5^3}{3\cdot 8} - \frac{1}{2^4 \cdot 4} - \frac{1}{4^6} = \frac{1}{64}$

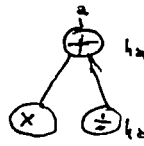
$\ln 30 = 3.4$

$\frac{3.4}{15} \quad \frac{6.8}{30} = .227 \quad \frac{1}{72} \quad \frac{1}{64} = .014$

.00 ; i.e. order is irrelevant. For non-commutative functions, each input is distinct



Say



has accord; & we want to search this content of z has accord marked.

We look at all ⊕ nodes: We want ⊕ nodes w. input from ⊗ & ÷. order is irrelevant. This may double the work we need in verifying "similarity", but this effect does not "multiply by 2" every time we iterate in a search. So it seems to be a not bad effect.

.10

.18 There may be other shortcuts in verifying duplication of a context here in the past.

.19 Note! In implementing vs. [redacted] context in the foregoing kind of way, it seems to be unnecessary to make "Definitions"; in fact it is more "free" than a implementation involving "defns". The advantage of Defns, is that it shortens & current code & makes it faster & easier to find new vargys in it. The Disadvantage is that it commits one to a particular parsing of the "known" code .... that will be "incorrect" in the sense that there exist(s) parsing(s) that will eventually result in a shorter over-all code.

.20

.21 Would the foregoing implementation of Context facilitate OSL (shelling)?

.22 It would seem so!

A kind of Compromise w.r.t. Making Defns (Re .12–.20): Make a set of definitions, but only after it has become clear that the defns are very good! i.e. Resp ↑ the corpus p.c. much more than any alternative set of Defns. This decreases the probability that one will be able to get better codes by using a parsing of the corpus in terms w. those defns. → 406.04

.30

.31 The Context of 404.33ff is one particular kind of Context: A single context of Tokens adjacent to the token to be predicted. A slightly generalization would allow the context to be not adjacent to the predicted token

.34 The generalization I'm thinking about is a class of contexts of tokens that are defined in a "simple" way,

.35 that conditionl. pc of a (usually (adjacent, close-by)) token. The previous kind of context could realize a class of contexts, but rc would be much larger:

.36 WOOPS: .36 may be quite wrong! Say we have a 10 contexts, each of same length L, that predict 1 w. pc. = p0. Its # same as one ..... ?? ... → 406.06

.40

3TM

or "BZIP2"

PZ2 — it corrects Pgen

"EZ" — Try at Google. EZTEN

since $\tilde{P}_{c_i}$ is always < true P_{c_i}

.00 : 403.3# SN on LSMA — "Modified Lgrsh".

For random Lgrsh: we do trial w. $P_C = P_{C_i}$ for duration Δ. For cc_i cc_2 it takes $\frac{cc_i}{\Delta}$ trials. Each trial takes time Δ, & we need $< \frac{1}{P_{C_i}}$ trials to hit P_{C_i} once.

so $\frac{cc_i}{\Delta P_{C_i}}$ trials to hit it cc_i times. Each trial takes time Δ so take $< \frac{cc_i}{\Delta P_{C_i}} \cdot \Delta = \boxed{\frac{cc_i}{P_{C_i}}}$ for soln. If we use $\tilde{P}_{C_i}$ ($\equiv PD_2$) to guide search & find solns.

It will take $< \boxed{\frac{\tilde{cc}_i}{\tilde{P}_{C_i}}}$ to find a particular soln. That takes $\tilde{cc}_i$ to generate & test, & is assigned $\tilde{P}_{C_i}$ by guiding PD_2.

.10 : 403.40 : Using PD_2, t. "soln." can be of very low $\tilde{P}_{C_i}$ because Progs 2. models & algms that always output a code.

$\tilde{P}_{C_i} \approx 1$

or "Ad Hoc" model of R.A. corpus: (Promiscuous A.H. models also described in 501647, for Inductive Grammars). Recently I showed (395.18-.22) that QATM also has these 2 Grammar types — Sequence/ produ. also has Rare 2 types.

Prog 2 "solns" while of very low CJS ($\frac{\tilde{cc}_i}{1} = \tilde{cc}_i$) usually gives terrible Gencs via pd_1. These are t. extremas of PD_2 (extremely bad). Problem w extremas is

.20 when $PD_2 \equiv PD_1$, so t. soln. is t normal Lcost = $\frac{cc_i}{P_{C_i}}$, which may be fairly good P_{C_i}, but t. Lcost = CJS could be much too large. Often (Usually?) in t. CJS for PD_2 guidance $\frac{\tilde{cc}_i}{\tilde{P}_{C_i}}$ = CJS will be accessible, but P_{C_i} will not be so good (never as good as that obtained by straight Lgrsh?).

Perhaps Many or Most PD_2 true algms w. $\tilde{P}_{C_i} = 1$. T. strong compression Algms like LZ or PZ2 give $\tilde{P}_{C_i} = 1$, & their P_{C_i}'s are almost always better than those of t. AH or promisc models.

.30 SN t.s.t. corpus size ↑, t. no of cases (& t. t. UNCONDITIONAL P_{C_i}'s of t. cases) must ↓ ↑ ?

.31 w. ↑ corpus size, we also have more contexts & ↑ in ssz'lt for each context

Can effect .30 help end w.s.d'ty of .30 to keep t. induction problem "feasible"?

.33 In counting how many times a Tree-corpus has occurred in t. MSP, It's easy to take commutativity of functions (+, ×, ...) into account. This seems to cause much of search time. Can I devise a better representation;... to help get rid of this d'fty?

Consider: for multifunction : 2 inputs but they are equivalent

TSQ write: :ooff : [redacted] 419.20 also on [redacted] TSQ

Loop

:00 : SN On writing TSQ's: I'd like to be able to write a TSQ, that I felt had adequate info, so that it should be tunable w. acceptable CJS! Then devise inductive mechanisms that are able to transfer "info" obtained by trng to solve one problem, to be used to facilitate soln. of t. next (& future) problems. T. idea of Genld content (402.12-.22) should be helpful in devising these inductive mechanisms

One way to write a TSQ: First make a seq. of problems that I feel ought to be adequate for trng / to solve some more diffl. problem.

Next write down various ways in which t. trng of .0-.03 could take place: Then try to get TM able to use all of these mechanisms. In general, I should be able to get TM to learn any Conc. that I "understand" by devising a Suitable t.S.P. (That TSQ may have to have a very large CSZ, huge [redacted] Start w- ANL (to recursive evaln). If I can't do it, Assume TM understands ANC [redacted] — This means I'm using RPN if it is easier for TM to do decline problems: So RPN is built into TM. — It's t. way we communicate w. TM.

:20 : Perhaps best use of Time in Costa Rica! Write very clear review of impt. ideas of last ~100 pp. Write clear review of "T. 2 PD's" à "Content" first. This should be clear to me of {t same or +1 yr.}. Working backward to earlier ideas in these 100 pp., t. later ideas may significantly clarify/solve, earlier problems.

:24 N·8·03 Review: P 403 is back about 100 pp. 6/8/04: unclear as to now what PD_2 is.

The "2 PD's" Idea: This is (ab initio anyway) Relevant to IND (Induction Probs) only. PD_1 is the a priori. This is a reader & assoc. w.e univl. lang. [AZ (≈ LISP), or FORTH ≈ OOPS [redacted], or whatever.] Because of t. way t. code is generated (see discn of AZ in IDSIA 2003 report: Appendix A for demo of AZ), at any point in time code sequences, we know its PD_1 (≡ a priori). The goal is to find one or more codes that distb. a particular corpus, that have as hy ≤ apriori as possl.

To do this, one could do an Lsrch w. PD_1 as t. "Guiding PD." This is "Normal Lsrch" An alternative is "PD_2 guided Lsrch": (see 404.00-08 for Details): In any Lsrch, if t. i'th cand. is a soln., its CJS will be $\frac{\tilde{cc}_i}{\tilde{pc}_i}$. $\tilde{cc}_i$ is cc needed to generate i'th cand.: $\tilde{pc}_i$ is t. pc used in t. "Guiding PD" — so $\tilde{pc}_i$ may be PD_1 or PD_2 — depending on what was used for t. srch.

3TM

402

11 Sept 03
10:59 P
10:59 P

00 : 40 1.40 : Hm, since Phase 1 is Only "Phase1", External Context will be quite limited. The framework will specify it extremely narrowly at first: Later, many contexts will be given to a problem & only a few (if any) will be relevant. TM has to determine which are relevant. This narrowing down of relevant contexts is trivial for a new problem, since we have all Other contexts that have been already tried, & TM can know which are relevant. The problem w. many given contexts arises in "updating", re determining which of all poss'l. contexts are useful for helping specify the PST of a new problem:

10 Other (approaches to problems) in selecting/using "Contexts" occurs in the "5 update situations" in
§ 1.2 of t. 2003 "IDSIA Report. (IDSIA-16-03) (Here I'm regarding "R's" as "Contexts".)

12 So a list of Context Types:

1) Token. Req. : This is unavoidable (at first)

2) a) Data of my Item i. : These are at first unconditional's

b) Other definitions ... (?)

16 3) The "R" type of content. § 1.2 of IDSIA-16-03

4) External context "This is a (Math problem/ Code/ PDE...)"

19 5) T. (Yuing Universal Genl. PD) : T. poss'l next Token is a cond. pr; w. "condition" being all previous Data & Solns & Soln methods plus External Contexts.

20 6) Any function like (5) but not universal, computable & not nearly having all those sets. (presumably contexts 1,2,3,4 are all covered by this).

22 In 1,2,3,4,5, I want to see just how each context type is implemented, discovered; so I can organize them properly. (e.g. parsing "R" from data structures.)

N.7.03 In this study, Look at the B_{zz3} (or zB_3?) Diff from B_{zz}? that v.g. compression programs, using "Context". Then I can apply any aspects of these techniques to more generalized "contexts": T. "contexts" may use any all previous of the above — i.e. may "nest". My impression: these are very slicky, hun, but nevertheless, 2 even permissible contexts of larger, t. compression ←---- This seems unrecognizable perhaps gaming to a Busy Beaver system. T. system itself is probably improvable because it was oriented toward practical compression & so it tends to look for "Best" code, rather than use all possible codes.

N.B. Various kinds of Context could be used in both PD_1 & PD_2: But remember, PD_1 & PD_2 have different Coupl.

00: : So O.K. token prog. is a context.
What about "Definitions"? I guess they can be regarded as contexts, same as token progs.
A slite difference: That token "contexts" are for tokens pc's or tokens only!
03 Definitions (e.g. "Nprocs") involve pc's that extend further into the future.

So 400.39 — 401.03 says Context can include token progs & token definitions!
So: so far, Context includes all the kinds of progs that I'm including in PD
models. Yes

10 400.39 ff is an impt New IDEA: I want to understand it better! — The idea that all progs I've been considering as PD_2 are contexts. Actually, this is a much diferent (& at all different) sense in the use of the term "context" in text compression.

15 Superficially, it would seem that any PD would be expressable as context sensitive p.e. — i.e. the pc of a token (or new) is expressed as a function of "what has come before." — i.e. conditional pd, with the "condition" being the known past corpus.

So it seems clear that Context can probly cover all models of the PD_2 — Restricting ourselves to context discovery would be same as limiting ourselves to using a PD_2 to guide Lsrch — which is probly not bad but is certainly not the most General way to solve problems (or do models) — e.g. Quick Abort & Long during Lsrch are techniques not covered by Models of PD_2: See 389.18 (about FN of Sol 89) for discn. of "Variations of Lsrch" & possible ways to include them in PD_2 .... This would be a possible (possibly messy) extension of Normal use of PD_2 for Lsrch guidance. It is an augmentation of the simpler Phrase model. We have, perhaps, alternatives of augmenting the simpler Phrase/Index model or of using a better/longer TSQ with simpler Phrase/Index (i.e. no long Query Lsrch)

30 Might be a good idea to list a large no. of "Contexts": Try to group them, to arrange the list so that it is clear that there are only a few (or one) way(s) to generate Contexts.
Well, the most General context is the most General PD_2, if I know how to generate all possible PD_2's. The only hitch here is that some contexts are "external": We might not think of them as possible arguments for the conditional aspect of PD_2.

About "External" contexts: If External Context can be anything in RW, then agree this is a "LARGE space"! → 402.00

See 435.19 for Good Genius of Context: Wtd upmost

N5.03

3TM

00: (Spec 399.08) 399.08 is a good explanation of what PD2 is about. It is not the most General Soln to a heuristic Search problem, but it is meant to be a fairly good soln. — (perhaps very efficient) — Good enough to Bootstrap to Phase 2.

The idea of a PD2 attempt to implement the ~~problem~~ search problem of 399.08 (i.e. doing this w. a Prob distribution) is certainly not the ~~most powerful way to~~ best way to solve the problem: Minimal differentiators! (1) No long durn search (2) No doing of "Experiments" (see 399.27 -.34).

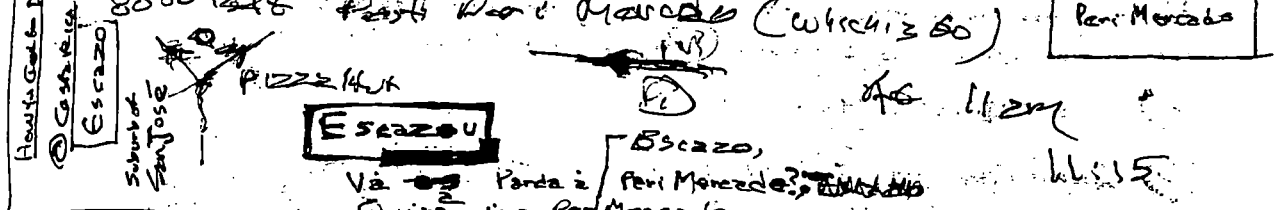
To some extent 399.08 can learn better trials [within the manner of the FN of 89: [399.18 -.33] — by having the p.d. being updated after each trial, — (since PD2 ~~detailed~~ can depend on info in the successes & failures of any & all previous trials — The Biz seems getting very close to the methods of Phase 2 ... (I may, in fact, introduce Phase 2 in ≈ this way.)

Anyway Phase 1 takes to implement 399.08 in as simple a way as possible, & get good enough to get to Phase 2 — (28)

Pasco Glorie 2 or 3 blocks west Panda Bus Stop

Escazu Cisco Quiznos to go to Peri Mercado via Escazu

Go to left fork Peri Mercado (which is Go)



(14) N6.03 So: The PD2 idea is a P.D. that avoids a modified Lisch. It is not the most General Form of Heur, ("Won" is closer; using PSTs), but if we stick to that form we are restricting the form of our heurs; but we (presumably) will not do badly — & good enough (w. suitable TSQ) to get to Phase 2 (.00-.16 discuss) loss/augmentations of the PD2 approach to Phase 1 — Depending on how good our TSQ is & on how good PD2 is, we may need to augment PD2 in those ways to get to Phase 2.

So. Going back to ~398.52 ff: Attempt to integrate "Content" & "Rotandolus" & Seven Tropes:

Token frequency is a context. Formation of, (the no of times, or the accumulation to-corpus up to now) is a context, & when Coloring w. other token tropes — which would be Contexts

397.40: Well, Context can be defined in a very general way, involving Conditional PC's — which can, in turn, be defined by Codes.

A "definition" is a far more general idea — defined by a code.

Frequency seems like a very elf. idea; but it seems to be very impt. in pc realm.

So, apparently 2 general types of PD: PD_1: reggys in ~~[illegible]~~ corpus:

PD_2 Reggys ~~[illegible]~~ successful & unsuccessful attempts to find codes for corpi. → for Phase I → 400.00

So, one approach is to simply look at human methods of search for various problem types & see if I can find any good general characterizings.

T. lists of 398.00—.06; 399.05—.40 two attempts in Phy direction.

Is full "Phase 2" ≈ t. most General such method?

For some time I had t (Mistaken) idea that all hours could be expressed as modifications of PD_2 (tho I didn't ~~[illegible]~~ understand t PD_1, PD_2 dichotomy at that time). One misunderstanding was that Phy didn't take "ring during search" into account.

The Footnote of sol 89 did take ring during search into account, ~~[illegible]~~ — the search algm was quite close to a general PST, but I don't think I knew, at that time, just how to find & implement PST's of that kind.

Essentially, I had t PC of t. candidates be a ~~[illegible]~~ (somehow optimized) function of all of TM's history (including "traces"). This is, indeed, close to t. most General hour-like possl., but I don't think I had figured out any ways to ~~[illegible]~~ find such "hours" (= PST's).

Hyp. Int. sense of being able to implement "ring during search" T. FN of 89 was Very Good .25 ff does something a bit inaccuracy - but it would still work very well.

Q: Just how was t. "FN of 89" to ~~[illegible]~~ implement "experiments" i.e. trials (or computations) made to gain info. not nec'ly directly to try to solve a problem.? — T "experiments" need not be "trials". In t. sense of direct attempts at solns of t. problem being worked on. (Tho I was certainly aware of t. impt. of "experiments" for RTM .... perhaps I had a way of ~~[illegible]~~ the implementing them as a kind outgrowth of previous experiences of all kinds.

Anyway, t. idea of PST is very General & covers t. "hours" of t. "FN of 89" & doing experiments (in front of a PST).

00 397.40: Well, Contexts can be defined in a very general way, involving Conditional PC's — which can, in turn, be determined by Codes.

" "

A definition is a very general idea — defined by a code.

Frequency seems like a very efficient idea; but it seems to be very important in pc realm.

So, there's 2 general types of PD: PD_1: regexps in [redacted] corpus:

.08 PD_2 Regexps [redacted] successful & unsuccessful attempts to find codes for corpi. → for Phase I → 400.00

So, one approach is to simply look at human methods of search for various problem

10 types & see if I can find any good general characterizing.

T. lists of 398.00-.06; 391.05-.40 are attempts in this direction.

Is still "Phase 2" a/t most General such method?

For some time I had/t (Mistaken) idea that all hours could be/t expressed as modifiers of PD_2 (tho I didn't [redacted] understand/t PD_1, PD_2 dichotomy at that time). ≠ One more difficulty was that this didn't take "ring during Search" into account. Maybe not so wrong! See 436.11-.22

.11

.18 The Footnote of Sol 89 did take (ring during Search into account, [redacted] — the [redacted] search algm was quite close to a general PST, but I don't think I knew, at/t that time, just how to find & implement PST's of that kind.

Essentially, I had/t PC of/t Cands be a [redacted] (somehow optimized) function of all of TM's history (including "Pieces"). This is, indeed, close to/t most General halting tester, but I don't think I had/t even

.26 a/t any ways to [redacted] find such "hours" (≡ PST's).

.27 **Q:** Just how was/t "FN of 89" to [redacted] implement "experiments" i.e. trials (or computations) made/t gain info, not really aimed to directly try to solve a problem.? — The "experiments" were/t need not be "Cands", in/t sense of direct

30 [redacted] attempts at solns of/t problem being worked on. (Tho I was certainly aware of/t impt/c of "experiments" for RTM..... Perhaps I had a way of [redacted] the implementing them as a kind out growth of previous experience of all kinds.

.34

Anyway, the idea of PST is very General & covers/t "hours" of/t "FN of 89" & doing experiments (as part of a PST).

Aw. In/t sense of being able to implement "ring during search" T. FN of 89 was Very Good .25 for a dichotomy - but it would still work very well.

3 PM

12:3 4 5 N 6 7
S S: M To W.

1905 = Eltono.

30 337.40 Anything it would be useful

In 396.04-.11 : The points; 1) Generalized "Context": The general definition of context (see 391.21 & 392.21, 25, 31-.36 390.35 ~ 391.04

contexts for tokens or for things larger than Tokens.

2) Recogn. formats,

3) Defns à pc's of Tokens.

4) T. SUMAC Concept.

36 5)

Since recogn. formats serve a part of contexts, (1); 2)) à (3) - also easily 2 kinds of "PD2". Sumac is related, but

38 39 — how to characterize the relation? .32

391.04 SN Re: 390.35 – 391.04 (on contexts) In Physics, Chem ("Mature" Sciences") the no. of concs does not seem to grow much — The t. no. of concs is quite large; but also is not relatively small no. of concs from which the others are derived, in a very hypothetic manner. (So little bit of pc in

N403 no. of concs. fc? → Re: Plus last : IN ANL (Saab), the defns were quite sharp, yet we had a seriously "Scaling" problem. So in the "Mature Sciences" it is not at all clear that we don't have the Usual "Scaling Effects"

SN On optzn problem v.s. Recgnt. Irng. As Best Model for Strong A.I.

T. OZ problem seems simpler à if solved, could solve the recgnt. Irng. problem. If the Recgnt. Irng problem is solved (for a good trainer) it could solve the general optzn problem. — So both problems are solve the other"

- T. problems also seems to be well defined.
- T. division of the system into sub-problems may have resolved in sub-goals that were not well defined (e.g. in T. Recognition "functional" approach to CATCH.

Eg. One Q is: Is OZ really simpler (as it seems to be)

Another is T. general danger of RTM — uncontrollability

The Plus occurs in any hy level problem solver as well, It seems to be more insidious in RTOM's (i.e. the trainer will not be aware of what he himself really wants — the goals given To RTOM may have subgoals of unexpected side effects. — i.e. the subgoals may be very bad for people in general.

32 .08 So, I could do more on integrating the (Recognition) System à the general context idea. This is closed up to large extent by 435.06-27 435.19 in particular

Hwr., T. Defns. à pc's of Tokens seems to be a unrelated way to (A) Get apriori (PD1) à (B) Get parts of PD2 (from the T. PD that Guides the Strm.

Apparently, Coding is the most General way to describe regularities. Contexts is a broad class of regs (or parts of regs) : A defns of tokens à pc's of tokens is another impt class of regs : But neither is as general as coding.