-1

Troubles:

~~1. " ∥ developements: No criterion for deciding when ∥ line of devt. is any good~~ ~~try to refer to equivalent soln. in real languages, to solve this problem.~~

~~2. Th. □ problem: Using methods other than the present one lead to digms like $\frac{1}{5}$, $\frac{5}{1}$, etc. See α 73.13 for soln.~~

3. $\overset{1}{\square}$ v.s. $\overset{1}{\boxed{1}}$ digm → Xtn. in 1 dim. in 2 or more?

SN's

list types of words.

a) Red.

b) Automobile

c) Rain, cloudy weather.

d) Pressure, temperature (?)

e) electron (?)

f) Linearity (of an equ. – to sugg. ways to solve it)

g) 6, 7; 6 is "close" to 5 and 7; it is closer to 5 and 7, than to 4 and 8.

h) periodicity.

---

5 Words ↔ sets. Sets may have mixed orders of sets for elements. This causes trouble in deductive logic, but not nec. in ind. logic.

6 That my main approach to T.M. is the formalization of a set of rules for ind. logic.

Troubles

~~(4) A really good understanding of (3) implies understanding just what my ngms corresp. to in real lang.~~

(2) **Back again!** $\frac{1}{s}$ is a digm of hy freq.

∴ th. digm $\frac{1}{\boxed{1}}$ is not very useful, since ~~most~~ half of th. time $1 \to \frac{1}{s}$, not $1 \to \frac{1}{1}$. This may not be so bad. It can force T.M. to use trigrams at an early age.

What about this! T.M. can get ngm freqs. from anywhere on th. array — but th. U of an ngm. is determined by its efficiency in predicting th. □ positions only.

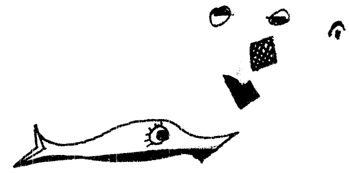
≈ Wed July 18, 1996   Notes   1

1) Def. of ugmst.

2) Just how □ is to be used.

3) Just what freq. counts will be made.

4) A tentative def. of U.



Use of both α 81.16 and many □'s in some Q's will make trigms. of hy U at an early age of T.M.

(simple Trigrams) and/ compact tetragms looks like enuf to solve any = or ~ problem

<bad>

fix up. 82.06 with a clearer def. of 81.16.

Should one retain ugmsts that don't have 100% relyability in prediction, for use as trial utδpst. componeats?

Perhaps: One doesn't really need more than a few ~~abs~~ methods of creating new ugms from old, if one starts out with 1) these basic methods and 2) Methods of forming hyer order sets. In this case, if one feeds T.M. a proper tug. sequence, it will, eventually, have to make the rite abss.. It may take a lot of time, but after they are made, T.M. can derive methods of getting new abss.

There is some Q. about just how much one is "pre-programing" such a T.M. by a "suitable" tug. seq. — and just how well he would eventually do on less ~~progr~~ carefully programed learning.

Th. def. of a "set" as I need it, is simply a set of instructions to tell 1) Whether a gn. example is a case or not 2) How many "times" (i.e. with what wt.) that example is "a case". This is for sets of sets of sets ..... in particular.

Another reason why it is good to average over all sqs. (not only □), is that one mite be gn. a series of correct examples, then a □, with question without the correct answer.

(Notes)

Ngm. of 1st kind e.g. $= \frac{1}{\boxed{1}}$ [used directly in prediction]

" " 2nd " e.g. $\frac{1}{1}$, $= \frac{1}{1}$, $\frac{1}{\boxed{1}}$

used as parts. of ntpsts.

One can ~~arbitra~~ ~~from any ... by~~

add $\boxed{}$ arround any digit of a ngm of ~~2nd~~ kind e.g. :

e.g. $\frac{1}{1} \rightarrow \frac{1}{\boxed{1}}$.

One can elim. the $\square$ from an ngm. of the 1st kind

e.g. $= \frac{1}{\boxed{1}} \longrightarrow = \frac{1}{1}$.

Carry line elim

In what sense is T.M. able to express

$$\begin{array}{r} 110 \\ +011 \\ \hline 1110 \\ 1010 \end{array}$$

as

$$\begin{array}{|c|} \hline 1 \\ \hline 2 \\ \hline 3 \\ \hline 4 \\ \hline \end{array} \times \underset{a \quad\; b \quad\;\; c \quad\;\; d}{(+110, 011, 1110, 1010)} \; ?$$

The point, is that the relation (ntpst), (a, b, c, d) is defined in an entirely different way from ever befor — i.e. by a rule that enables one to get c and d from any a and b..

Or even $\begin{array}{l} =10010 \\ \;\;10010 \end{array}$ as

$$\begin{array}{|c|} \hline 1 \\ \hline 2 \\ \hline \end{array} \times (=10010, 10010) \; ?$$

1) Th. ngms are symbols only. They are a code ∋ one can take any 7×10 array with digits in it (make up word for this); and

2) Tell if it is in th. ngm's set b) tell how many times th. ngm. occurs in it (≡ assignment of a no. to th. array, ∋ th. no. is betw. 0 and 70, inclusive.

There are rules for manipulating th. symbols. These symbol manip. rules (bear a corresp. to) (are reflected in) set. manip. rules.

Find out which correspondence is "on to part of" Perhaps homomorphism? Th. example I am trying to name is ngm symbols → sub sets of th. 7×10 array configs.

or "A completely maps onto part of B."

2) For carry line elim:

a) $= \begin{matrix}100110\\100110\end{matrix}$ is $\begin{array}{|c|c|}\hline 1 & 2 \\ \hline & 2 \\ \hline\end{array} \times (=, 100110)$

or $= \begin{matrix}\alpha\\\alpha\end{matrix}$ is a special case of $\begin{array}{|c|c|}\hline 1 & 2 \\ \hline & 2 \\ \hline\end{array} \times (=, a_i)$, where $(=, a_i')$ denotes th. set ∋ a_i can be any horizontal "linear" config.

b) Th. more general way to get this type of ngst. is by use of th. "function" idea.

One expression of this idea, is that in

$$\sim \begin{matrix}1001\\0110\end{matrix} = \underset{\text{str}}{\begin{array}{|c|c|}\hline 1 & 2 \\ \hline & 3 \\ \hline\end{array}} \times (\sim, 1001, f(\sim, 1001))$$

here $f(\sim, 1001)$ means that th. elements of $f(\sim, 1001)$ are derived from th. symbols $(\sim, 1001)$ by a certain set of operations. This "set of operations" is somehow represented by th. union of th. ngms

$\sim \begin{matrix}1\\0\end{matrix}$, $\sim \begin{matrix}0\\1\end{matrix}$, $\sim \begin{matrix}a & 1\\a & 0\end{matrix}$, $\sim \begin{matrix}a & 0\\a & 1\end{matrix}$, $\sim \begin{matrix}a & a & 1\\ & & 0\end{matrix}$, $\sim \begin{matrix}a & a & 0\\ & & 1\end{matrix}$

etc.

~~Odors : a filter may be looked upon as a linear xfmn. on the. odor. (≡ matrix). We have many accurately calibrated attenuation matrices~~

When we have a set of ngms like:

$\sim \begin{smallmatrix}1\\0\end{smallmatrix}$, $\sim \exists \begin{smallmatrix}0\\1\end{smallmatrix}$ etc., that are, in a sense, "complete", we should want to combine them — I think this should ↑ ~~[illegible]~~ sample size, and ↓ no. of members in the element set.

7×10 array ≠ element

Morphisms: ~~≢~~ a one.to.one mapping into maps ~~≢~~ a set A to subset of B

A one to one mapping onto, maps all of A to all of B.

A homomorphism is a mapping that preserves certain relations on a set of elements — like addition or multiplication.

Write Abstract at end. Also, write introduction — on nature and meaning of problem. — 6

Also, in writing in T.M. notebook give reasons for decisions or references to them

Explain various modifications and choices that mite have been made, E.G. Use of math T.M., with a correct answ. existing for every Q, make def. of a "consistant p.ngm" a useful def. — That this must eventually be dropped.

e.g. choice of 7×10 array not critical.

explain that def. of str. is very unsatisfactory.

There should be a more deterministic way to go from $= \begin{smallmatrix}1\\1\end{smallmatrix}$ to $= \exists \begin{smallmatrix}1\\1\end{smallmatrix}$ than via str., since str., as I have defined it is ambiguous when one part of an ntuple is a digm or trigram.

The useful ngmst of which $\sim \begin{smallmatrix}1001\\0110\end{smallmatrix}$ is a member

can be expressed as $\underbrace{\boxed{1\,|\,2\,|\,3\,|\,4}}_{\text{str.}} \times \underbrace{(\alpha \hat{x} \alpha \hat{X} \alpha \hat{X} \alpha \hat{X})}_{\text{ntpst.}}$

where $\alpha \equiv (\begin{smallmatrix}1\\0\end{smallmatrix}, \begin{smallmatrix}0\\1\end{smallmatrix})$

← pair ~~[illegible]~~ of digms (ntpst).

(Notes)

What I have been doing is running thru simple algebraic learning, trying to get T.M. to learn certain basic abss. which I think essential. I could just ~~than~~ build these into T.M., ad-hock, but I don't think that this would be effective. Reasons: 1) That th./ad-hock rules might ~~not~~ be too restrictive, ~~not~~ might do only part of what I really need — that non-ad hock rules are apt to extrapolate better 2) That th. ad-hock rules may actually do ~~harm~~ inadvertant harm — i.e. they may do th. ~~~~ wrong things in cases I had not thot of 3) That ad-hock rules will become too numerous and are not as easy to store and use, as th. more "natural", non-ad hock rules. 4) That, using ad-hock rules, it is quite possl. that one could completely miss some very imp. rules. This is far less likely with "natural" rules

---

~~on~~ On α91.08 there is th. q. about whether to include q. elements in th. ngm. counts.

Possibly define ~~~~ "deletion" of a pgram as its xfmn into an ngm by ommitting $\square$. Also creation of ~~~~ a pngm by th. converse process on an ngm.

Perhaps make pngm a special case of ngm?

Try eating twice a day: Noon and Midnite.

~~Sleep from~~ eat Noon - 1 PM, Mid - 1 AM. Sleep 1 AM - 9 AM

Only trouble is eating with others occasionally, but I think I could manage this. ~~Another~~ Other possibilities, but main idea is 2 meals/day and sleep rite ~~after~~ 1 of them.

Another poss. eat Mid - 1AM, sleep 1AM - 9AM eat 9AM - 10AM ← normally, I will be very hungry in th. morn. if I eat at Mid - 1AM.

Th. big disadvantage here, may be not taking advantage of

.02 early ~~morning~~ waking acuity for problems worked on just before sleep.

This is fixed in .23).07

Th. Q. of U of a pngm. Should U be $\propto$ no. of (correct) predictions? Should it get extra U, for predictions that few other pngms could do? (or "few" in terms of expected reliability).

Should th. 2nd and 3rd registers contain strsts and utpsts or strs and utps ?

.12 Suppose we have a □ in a q.element ∋ there are 2 pngms that fit it: one has a pngm count of 10, with 2 cases, th. other a pngm count of 6 with 4 cases. Each pngm predicts differently. How can one assign wt. to th. pngm.counts and cases?

Suppose we have a pngm set ∋ it fits a □ in 2 ways, each with a dift. prediction! eg. 1 $\boxed{1}$ and $\boxed{0}$. Then we have the q element with 1 □ in it. What does one do then?

Both of these cases can't happend too often, since as soon as they do happend, the q.element that caused them is followed by the "correct" completed element, and one of th. pngms causing trouble, is eliminated.

— It may, indeed, turn out that q.element φ's are never followed by corresponding correctly completed elements — but even so, these φ's shouldn't occure too often.

~~See bott of α85~~ transfered to bottom α85

Possible defs:

very probably not. 1) "Case" used on α86, also α94 — a defined on α86. Also "case count" (or use no. of cases instead) — defined on α95.01.

2) "counter case" mite be useful in defining "consistancy" — a consistent pngm is one with no counter cases.

— some ambiguity i.e. counter cases over C_1 or C_2 ? "↑ a11" "↑ □"

see 10).04

(Notes)

Th. reason I want to tie U to $\Box$, is that 1) Otherwise sss and other pgms for $s\boxed{5}s$, predicting s, get by U. 2) It would seem that we should give by **U** to pgms that predict things of value.

Try this "soln": pgm $1\boxed{1}$ has pgm count of 10, and 2 cases, $\overset{1}{\boxed{0}}$ pgm count 5; 4 cases. $\overset{1}{\boxed{0}}$ is clearly more <u>relevant</u>: ~~case of $\boxed{0}$~~ it is th. relative size of 2 and 4 that are imp. However, since 2 and 4 are small, one can be > th. other thru chance — so $\overset{1}{\boxed{0}}$ isn't <u>much</u> better than $1\boxed{1}$. If $\overset{1}{\boxed{0}}$ had 40 cases and $1\boxed{1}$ 20, th. situation would be more decisive.

There is then th. q. of how this applies to th. situation in which there are no answers following th. q. elements.

.22 Consider: If pgm α has a count of n and c cases, what is th. proby. of a counterexample on th. next element or q.element? T.M. could simply <u>determine</u> this <u>empirically</u>, as a function of n and c. — with suitable "smoothing" after theoretical analysis gives an apri idea of th. functional form to be expected.

$\boxed{SN}$ An imp. problem is th. optimum non-linear filter problem. Give a few reasonable examples of how to smooth, (Th. <u>continuous</u> case, of course!).

It is <u>not</u> nec. to follow q.elements by th. correctness. A pgm can have many <u>useful</u> cases without knowledge of their correctness.

Say an pgm has count 10 and 3 cases. What is th. probable fraction of th. 7 (= 10 − 3) that

(Notes)

"could havebeen" cases? Can't get much meaning into this.

Plots of count and cases v.s. expected success of next try, may suffer sharp discontinuities when new concepts are introduced.

Getting a T.M. of th. present type to learn **Tic Tac Toe** would be interesting. It would have to learn to count to decide whose move it was, and that only one square need be filled in. Perhaps it could learn to match o's and x's to determine what to do, if this would be a simpler operation.

There is some Q. as to whether it should try to learn T.T.T. from th. beginning — on a clean slate, or learn some arithmetic first, with its associated strs. and atpsts.

Th. first problem of (6.12 is not a serious prediction problem. It occures rarely, and at such times some sort of simple weighting (linear or mean square) for "count" and "cases" can be used. Th. big Q. is to what extent U should be a function of no. of "cases" — v.s. "count".

From ordinary language, a "word" is "good" if 1) it is often useful in prediction 2) It is useable when no other "word" will help much 3) It is economical — i.e. it can be used for very many phenomena, so that with just a few such words, one can predict practically everything that is asked. (In th. present Math T.M., 3) is perhaps th. same as 1)).

To summarize 1) 2) and 3): A word is "good" if it promises to be one of a small set of words that successfully cover all contingencies. (→ Occam's razor)

From a more systematic standpoint a way to get a set of words of hy U!

Take all <u>cases</u> up to time t. Above each write all consistant pngms that were relevant to it. Try to get th. smallest set of pngms that covers all <u>cases</u>.

Clearly, one can do this ad-hock, by making a new pngmst that is th. $\beta\Sigma$ of all pngms that <u>do</u> cover all cases, but this <u>is</u> ad-hock and ∴ illegal (≡ of low $aprip\Sigma$ of usefulness). — actually this wouldn't be too bad — except for some incompatibilities — Also th. <u>whole</u> pngmst would be thrown out as soon as one counter example was found. This wouldn't be true if one kept th. pngmsts separate.

<u>General Conclusion</u>: U should be obtained mostly, if not entirely, from th. no. of <u>cases</u> that a pngmst has applied to.

For T.M.'s in general: Ideally, <u>many</u> □'s should be asked, and T.M. should be gn. th. rite answers to them. If T.M. is gn. many □'s, but no <u>specific</u> answers to them, T.M. can <u>still</u> operate efficiently. If, however, very few □'s, or none at all, are gn., then T.M. must assume something like "all digit predictions are of = importance" — or one may ad-hock exclude as of very low U, predictions of th. space, S. These are clearly bad operating conditions — in which T.M. doesn't know just what in its envt. is important.

"Count" can be looked upon as a method of comparing U's of 2 entities, if their no. of <u>cases</u> are =. However, th. wt. of "count" is much less than no. of <u>cases</u>.

An analogy is a ">" relation for complex nos. $A > B$ if $Re\,A > Re\,B$. if $Re\,A = Re\,B$, then $A > B$ if $Im\,A > Im\,B$.

Give stars in front of ~~sections~~ paragraphs to be ommited ~~at 1st reading~~

✳ Follow exact discription of Math T.M. with an explaination of just what pngms, ugms, strs, and ntpsts correspond to in ordinary lang.

Also <u>Contents</u> — with much explaination.

Also write <u>abstract</u> of paper

Also insert abreviations and put them <u>with</u> definitions (<u>prediction</u> ngram ≡ pngm)

Explain that all underlined words are specially defined in the text, if there is any doubt about their being special words.

(Notes)

Desbutol

On Consistency: I have defined consistency as no counter examples over all squares. It is possible to define "weak consistency" to be correct predictions in all cases.

.04 Def: Shall "weak consistency" be defined?

~~Usefulness : Abbreviation, U~~.

Defs: pngmst, ngmst, utpst, strst?

Define U of all but pngms in terms of expect U of pngms to result from their xfmn into pngms.

Explain in simple case how to get U's of utpsts and strs: Say s_i are strs, and n_j are utpsts. Let U_{ij} be the observed U of the pngm resulting from $s_i \times n_j$. Let U_{s_i} and U_{n_j} be the U's of s_i and n_j resp. Then we want to adjust the U_{s_i}'s and U_{n_j}'s $\ni$ U_{ij} is best approximated by $U_{s_i} + U_{n_j}$.

~~Explain how U varies with aprip and cases.~~

Also possibility that U may not be mainly a function of cases no. of a pngm, but may be assigned with considerations of what other pngms are available — so as to cover all cases with economy of no. of pngms used.

Shall I describe xfmns and combinations; and then creation of ngms and pngms separately; or use a unified approach?

There must be a ~~[illegible]~~ continual verbal housecleaning — discarding words that are completely covered in function by "uniformly better" words. In this process new, better, words may be devised — in fact, perhaps some essential word building devices are introduced at this stage.

This housecleaning process is nec. 1) epistemologically 2) to keep th. search processes for th. correct word within bounds.

Description of th. Tng. sequ.

1) Use many D's per q. element.

2) All q. elements are followed by th. correctly completed elements. (Not done)

3) For =, ~, etc., use problems of various length, e.g.

~~[illegible]~~ = 5, ~ 10110 01000 etc.

4) This care in ~~selection~~ construction of tng seq. should be nec. only at th. very beginning. Later T.M. should be able to deal with most any thing. An imp. prob., is just ~~[illegible]~~ which abss. T.M. must have befor th. tng. seq. become less imp. — An imp. characteristic of ~~[illegible]~~ such abss., is that they are statistical and are not ~~rise or fall~~ destroyed by one counter example. For this reason, Math T.M. is bad, since a different process must be used in dealing with abs. methods than with direct prediction of interrog. squs.

5) cases and counts of prgms are both stored., but case no. is used as a final index for U.

if we want $U_{ij} \approx x_i + y_j$ $(i = 1|m)$
how good is the approximation $(j = 1|n)$

$$x_i = \frac{1}{n}\sum_{j=1}^{n} U_{ij}$$
$$y_j = \frac{1}{m}\sum_{i=1}^{m} U_{ij}$$
? Then $U_{ij} = \frac{1}{n}\sum_j U_{ij} + \frac{1}{m}\sum_i U_{ij}$

if $U_{ij} = x_i + y_j$ then $\frac{1}{n}\sum_j U_{ij} = x_i + \bar{y}$

$\vec{X}' = \vec{x} + \bar{y}\cdot\vec{I}$ $\qquad \sum_i\sum_j U_{ij} = \bar{x} + \bar{y}$.

$\vec{y}' = \vec{y} + \bar{x}\cdot\vec{I}$.

so $$x_i \approx \left(\frac{1}{n}\sum_{j=1}^{n} U_{ij}\right) - \frac{1}{2}\frac{1}{mn}\sum\sum U_{ij}$$
$$y_j \approx \left(\frac{1}{m}\sum_{i=1}^{m} U_{ij}\right) - \frac{1}{2}\frac{1}{mn}\sum\sum U_{ij}$$ isn't a bad approximation to try. Another, better fit can be gotten by writing

$$x_i = \left(\frac{1}{n}\sum_{j=1}^{n} U_{ij}\right) - \frac{1}{2}\frac{1}{mn}\sum\sum U_{ij} + \delta$$
$$y_j = \left(\frac{1}{m}\sum_{i=1}^{m} U_{ij}\right) - \frac{1}{2}\frac{1}{mn}\sum\sum U_{ij} - \delta$$

absolutely unnec.

and finding the optimum δ. — A simple closed expression can probably be found for δ.

No, δ is not really necessary. Since we form our apri U's by $U_{ij} \approx x_i + y_j$ adding δ to x_i and subtracting it from x_j is irrelevant. We can, indeed, let

$$x_i = \frac{1}{n}\sum_{j=1}^{n} U_{ij} \quad \text{and} \quad y_j = \frac{1}{m}\sum_{i=1}^{m} U_{ij}$$ if we

remember to subtract out the constant, $\frac{1}{mn}\sum\sum U_{ij}$ every time we let $U_{ij} \approx x_i + y_j$ i.e. $U_{ij} \approx x_i + y_j - \frac{1}{mn}\sum\sum U_{ij}$

to 12).12

Th. apri U is not enuf. The weight or variance of this aprip U must also be given. It can be obtained by computing
(U ultimate, observed $-$ U_{apri}) for many pngms. This will have to be done separately for every "method" of combining ngms, ntps and strs.

Because of this, ngms, ntps and strs may have several U's, — one for each combination method.

A trouble: one would like to get (U ultimate obs.) for all pngms, but usually it is not available. $U_{obs.}$ is always availabe, but if it has had few cases, th. U_{aprip} in it is too important. For small samples, it is imposs. to elim. th. effect of U_{apri}, since it and its wt. are a nec. part of $U_{obs.}$

Perhaps what we really want, is not such good prediction of $U_{ult.obs.}$, but good predictions of when U is likely to be zero. Here again, Math T.M. is peculiar, since there are many U's that will soon be known to be zero. This mite easily be used to advantage. the U_{apri} could be a measure of th. probability that th. pngm created will not immediately be gn. a U of zero, because of a counterexample.

Actually, most U's may be "ultimately" zero, because in a reasonably variable envt., they will have counterexamples eventually. At any rate, a useful concept mite be, how many "cases" and "counts" a pngm had befor it "died".

[SN] A special device may have to be introduced into T.M. so that when >1 $\square$ occures in a pngm, T.M. will not get info that it shouldn't. e.g. in $1\;1\;\frac{0\,\boxed{\gamma}}{\alpha\,\boxed{0}\,\boxed{\beta}}$, T.M. should not look upon this as a case of $\boxed{0}\,\beta$.

This mite be avoided by giving th. 9 · elements and th. correct corresponding element, separately in time.

(Notes)

~~We have~~

<u>MATH - T.M.</u> : We have run into several objectionable features of Math T.M. - They are based on the fact that absolutely rite or wrong answers do not correspond ~~ing~~ to what is met in the ordinary language. In particular, this make U evaluation unnatural, and it makes the manipulation of methods, (which are of necessity probabilistic in effectiveness) different from the operation of prediction pugms.

Obtaining Variance of U: I suggested a separate U and variance for each, say, str, and each kind of application of a str. Now clearly, at first one will have very small samples, so one will have to group all applications of a str. together to get U and th. variance of U.

.07 There is some q. about whether one needs a greater sample to get U or variance of U. — i.e. one may have to group in order to get variance of U, but not in order to get U. to (16.16

One may have to Group all ngms together to get U and variance U, at first. (Also all strs together, and all ntps together). This grouping will have to stop at some point — (say when all strs have been gn. a common U), and one will have to use "apri faith" th. rest of th. way.

Also, as one gets bigger and bigger samples, one can continue subdividing th. various words into smaller functional groups. E.g. one can have several prediction parameters per word, to be combined with those of any other word.

Sounds good, + how does one do it? Perhaps it would be best to break up a word into several sub words, each with only 2 (say) params, rather than give more params. per word. This would ~~make many~~ give a more unified programming approach, rather than sporatically hunting down words that have a big enuf sample to warrant their getting an extra parameter.

Actually, there is a more general, exact approach to this problem. I think th. problem is identical to that of "How many coiffs in a regression curve can I use for a gn. sample size" — Th. answ. is, that one can use infinitely many coiffs, if, indeed one puts in all one's apri info. This bussiness where one only