4 · 12 · 79 General Plan:

1

1) Work on TM for next month or so.

2) Then Go to Cleveland w. Grace: Perhaps stay 2 wks or 1 month. Working on T.M. at that time: Terminate Cleve visit w. e.g. review.

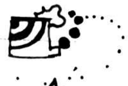
3) On returning to Cleveland; Try Getting job or spend Month on Minic
Perhaps spend ~2 wks on SM: Options: After Minic is working w. TTY, say : Basic or equiv; or Buy Forth. : Get Forth book from Decus
— Perhaps thru David → If SM looks good enuf, don't get job.

Perhaps, if larger investments in SM are Necy, Ask David & other friends to invest. [diffy w. IRS, perhaps]. Perhaps interest G. Brown. → Perhaps Mike can get me tempo Job at Intermetrics or Amer Sci.

4) Do take off a few days within next wk. to Make skull clock, fix Phone Butler & Make phone dialer.

---

5) So Best sequence: a) Soon after going to Camb., do
b) work on TM for ~1 mo.: Try to make some review at that time: say spend 3 days on review.
c) In Cleve: spend as much time on T.M. as Possl. : Perhaps spend total of 1 mo. in Cleve. working on T.M.
d) Return to Camb. Work on Minic for ~ 1 mo. Try to get it at pt. where it is useful for SM work: I.e. has Basic a/o Forth (Buy).
e) Work on Options for a couple of wks. If it looks good enuf, don't get job: Perhaps try to get others to invest to reduce Bkrp..



— Trouble: We don't get vacation: Also I do want to spend at least 1 mo (Sept. a/o Oct) working on House: "Siding" also fix ports, also fix Piers & ~~broad edges of boos~~ 2 places that were rotted.

.35

Mid Ap	TM
Mid May	TM in Cleve
Mid Jun	Minic
Mid ~~[illegible]~~ July	SM.
Aug 1	

Perhaps in Aug: Mini vacation: Say 2 wks in Montreal? or Nova Scotia w. Warren.
Then Sept in NH
Oct: Get Job $ [redacted] in South: Say Calif, Mex., S. Amer.

4·12·79 Genl. Plan:

There is some possy that in a month or so, I will have suffly brokenup (a'zd) TM so I can work on parts of it w.o. so much "warmup period".

— e.g. I could probly start work on Minic within day or 2's warm up.

— Hvr. TM work is harder when t. prob. is very general, & its expedient to have lots of info in rapid access memy.

So perhaps t. Elzn of TM should be an immediate sub-goal: T. clear definition of various sub. problems.

Then I can do 1.35 −.40 (left): Then in Aug. Take 2 wks off for brief vacation (Montreal or Novascotia (or Iceland?))

2 wk in Aug & and Sept & Oct on parts of TM.

Start looking for Job Nov 1: Try to get job in warm foreign country or California (San Fran) or ~ Ming for 4 to 6 mo.

---

1) So: **Now** Back to Cambg. work on ① Ph. Butler. ② Skull clock ③ Phone dialer.

2) work on TM to Mid May. Try to elementalize it.

3) Go to Cleve: work on TM to mid June:

4) Back to Camb.: work on Minic to mid July try to get Basic a/o Forth running for SM work.

5) to Aug 1 work on SM 2wks: Options: BRANCH.

If SM works: Play MKT to ~ Sept 1 | ~~works TM~~ House, TM, Sept, Oct

If SM doesn't work: work on TM Aug, Sep, Oct. work on House Sept, Oct.

If SM works Take vacation in Foreign Place — depending on how well it works.

If SM doesn't work ~~Get job~~ try to get Job starting Nov 1 in Vacation-like Place. (Mex., S.A., Calif. . Possibly Florida).

4.7.79 : TM: ~~Roman~~ Rom (Remenisce): This is a list of impt. things worked on in t. past that I've almost forgotten, that seem very relevant now:

1) 78 RS ~~5~~: 13.01 : A justification of optimality of Random search: Now, this is more interesting.

2) 78 RS 51.16 ff : Outline (& some detail) of TM approach (≈ present approach but less details... refs to some earlier work, I think).

3) 78 RS 49.10 : T. idea of TM searching for experiences that are educational.

4) 78 RS 45.01 — 46.10 on RTM : some approaches.

5) 78 RS 57.12 : on My Pgming Bugs & debugging as a source of heurs & trg. seq material

6) " 36.02 — ~37.28 : on t. sequential optzn. prob. I think I now understand a formal soln.

7) " 31.28 ff : Defn of "F"; this corresp to my more recent Al_{gk}, but now I know more about t. reqts. of Al_k & just how Al_k is limited in acting on those reqts.

8) Somewhere between 78 RS 15.01 & 66.40 are some discns. of Biased search & cross refs to it & "U coding". Some relevant things mentioned: Use of previous trials info in coding to guide present search. Try to find this. 23.04 has some: 22.18 ; 16.01 ; 17.01 ← (hurr, these are not what I originally saw) ; 78 NKGS 8.26 ;

9) 78 RS ~~58~~.01 Ref. to "Software factory" as a TM realzn approach.

10) 78 NKGS: 2.10 : T. role of finite CB's in helping learn heuristics.

8-25-79 ≈ **TM. Genl:**

On t. idea of a Human, Machine interaction in order to solve problems "Best".

(1) The Human is told at all times what t. pcosts & ccost of ~~[illegible]~~ all available concepts are.

(2) In solving problems, t. machine is allowd to use t. human as a sbrtn. This is done by telling t. human what t. sub problem is & asking for a soln.

(3) In a similar way, t. human is allowd to use t. machine as a sbrtn. in his own. problem solving.

(4) The machine - human combn. is really a very large set of ‖ processors. Th. human is certainly one such set by himself, but t. machine also trys to solve problems by ‖ approaches — switching from one approach to another, depending on how things look: The human, usd as sbrtn, ~~a jus~~ to t. machine, just another ‖ approach — & similarly for t. human considering t. Machine.

(5) The machine considers t. human as a tool. As such, a certain amt. of ccost is necy for maintenance. In this case one impt. kind of "maintenance" is training the human to use the best (lowest ccost, hygest pcost) concepts, so that t. human's solns. are better. By observing t. humans rate of learning, t. machine can decide how much time to spend on tng. t. human in various areas. This is ~ to t. problem of t. Machine's teaching typing by giving more practice on those keys for which expected improvement per unit ~~of~~ practice time is largest.

So, t. machine will be often giving t. human "tng. problems"

I'm not sure if t. human ~~[illegible]~~ should give t. machine "tng. probs" — tho it may to get machine to derv. & suitably code suitable concepts that t. human thinks t. Machine ought to know about. [Yes! This is a useful device! See TS 66.28 — 77.40 (2.16.80) for discn.

Among other things, t. machine would teach human to type better if that seemed like it would ↑ thruput of t. system.

N2479 TM Genl:

→ A good final Approach for T.M.: typ. Seqs!

After T.M. works each problem, try to get as much (as poss) out of that particular soln. & soln. method & "soln trace" (= path of operations involving t. prob.).

.04 Have me do this at first, but as T.M. gets briter, it should eventually

.05 be able to do it fairly well itself.

.04 – .05 is Mindful of TM_2: & indeed, this operation is an impt part of TM_2's work, & is perhaps one kind of elzn. of it.

Tho this seems like a simple, trivial idea — it also seems to be very impt Methodologically!

One aspect of t. work: Try to "index" t. soln., so that when similar probs. arrise in t. future, similar (or suitably xfmd) soln. methods will tried. The problem & soln.

.12 must be genlzd as much as possl. For t. actual soln. method used, one wants ① T. set of all possl. probs w. that soln. (or a large subset of that set of probs ② A genzn. of that soln. & for each possl. genzn.,

.17 t. assoc. set of probs that it will solve.

T. Genzns of .12 – .17 should be done "linguistically" — i.e. a genzn. of something should be a "linguistic" xfn of it. Ideally, t. genzn. should be obtaind by deleting various bits from t. descn. of t. thing generalzd. One should devise definitions in t. lang. so that t. desird Genzn. can be put in this form. (Good Methodol trick!)

7·24·78: Rand Srch [13 – 15] ← See 4.01 ff. ; PW (cross coupling) ~~[struck out]~~ [24–25] ← See 1.03 ff

These seem to be 2 very impt. ideas:

.03 [PW (cross coupling) 24–25]: This is t. continuation of ~~[struck out]~~ an impt. idea starting at PW 21.20.

The idea is this: Say we have coded a corpus, A, and we are about to code t. next section of t. corpus, B. We make some "measurements" on B. A "measurement" is the obtaining of info. that restricts t. possl. says that B can be.

Examples of "measurements":

.13 1) B is 200 bits long (this is a peculiar kind of measurement – I'm not so sure how useful it is by itself.) and 73 of t. bits are 1.

.15 2) [The 4 gram 0110 occurs 25 times / B has 200 bits] [$\frac{200}{16} = 12\frac{1}{2}$ – on t. average it w'd be occurred 12 times]

3) B consists of a seq. of 103 real nos., each to 16 binary places

.17 4) Linear regressn w. 5 coifs has given a compressn factor of .23 in t. corpus.

.18 5) Maxm (optimum linear regn) has given a compressn. factor of .2 in t. corpus.

These measurements can be obtained directly on B, yielding t. measurements only (e.g (1) (.13)) or they can be obtained when one tries to code B – either they are derivs of t. result of t. coding (as .17 & .18 (4) (5)) or they may be obtained as a by product of a successful or unsuccessful attempt to code B – e.g. .15 (2);

Another concept ~~[struck out]~~ we need here is t. idea of a Pem. (≡ Probability evaln. Method). This is any effectively computable algorithm that enables us to take a sub-corpus & assign a probty to it. Examples of PEMS:

1) Maxm, giving all possl. nos. of coifs

2) Linear regressn. w. 5 coifs. (it is necy to say how t. first 5 "predns" were obtnd — an easy way would be to use direct coding ~~[struck out]~~ for them), using Maxm.

3) Bern seq. w. 0 & 1 only.

4) Bern seq. on seqns of 2 bits (≡ radix 4; it is necy to state where to start — i.e. phase must be gn. Or use both phases & "average".

5) Bern seq. w. radix 2^n: (This is a 1 param (n) set of Pems)

6) Use of simple definitions for coding (this was dscrbd in part II of t. I & C paper).

7) N.L. regressn using 5 terms & all 2nd order effects 5 linear, 5 squares & $\frac{5\times4}{2} = 10$ cross products = 20 coifs. Using Maxm for ~~these~~ that agreed upon set of coifs.

.01 After coding t. (relatively large) corpus, A, we observe, empirically that whenever measurement $\vec{\alpha}$ was made on a sub-corpus of A, & value $\vec{\alpha}_0$ was obtained for it, then Pam, P_3 obtained a bcost compression on t. average of .21.

.06 [bcost compression is ≡ $\frac{\text{bcost of sub corpus w.r.t. } P_3}{\text{no of bits in t. raw sub-corpus}}$]. An observation like

.08 .01 could be obtained by repeatedly obtaining $\vec{\alpha}_0$ over a very large set of sub-corpi.

.10 More generally, using general induction methods, we may conclude from t. data in A, that if measurement $\vec{\alpha}_0$ is obtained, that $P(c)$ is t. proby distribn. of compression c for Pam P_3. Or if P_n is a 1 param set of Pams, $\vec{\alpha}_0$ implies $P(n,c)$ ~~as t. compress~~ as t. proby. distribn. of compressn., c if P_n is used

.18 on a sub corpus.

Statement .10 – .18 will usually be obtained w.o. necly. observing

.20 t. measurement $\vec{\alpha}_0$ in any sub-corpus of A. Th. data on which [redacted] .10 – .18 is based can be <u>any</u> set of measurements on sub corpi of A & [redacted] compressions obtained w. <u>any</u> set of Pams operating on those sub-corpi, using any good inductive inference method. The conclusion is most <u>easily</u> obtained if t. data is like .06 – .08, but that situation is unusual — its like t. freq. concept of

.28 proby: "O.K. if t situation warrants using that induction method".

So, we make various measurements on B (including, perhaps, coding attempts of any ° of success). From these measurements & induction on B, we obtain expected values of compressions to be obtained thru several difnt Pams a/o families of Pams. Presumably we also have distribns of ccosts for these Pams w.r.t. B.

[The ~~ccost dis~~ proby distribns of these ccosts are obtained from induction on t. data in B — just as $P(c)$ was obtained in .20 – .28.] What we usually want, hvr is a <u>joint</u> proby

.32 distribn. for compression and ccost. (≡ $P(c, cc)$ of each Pam.)

We then order t. Pams in expected value of

.37 $\frac{\text{pcost wrt that Pam}}{\text{ccost " " "}}$.

.38 If t. raw corpus has N bits (.37) ≡ $\frac{2^{\text{compression wrt that pam}} \times N}{\text{ccost wrt that pam}}$

Rev

· we then try out t. Pams on B in order of [Expected value of (2.37).]

[Note that for large N, 2.38 (& ∴ 2.37) is relatively indep of ccost & is very strongly dependent on compression ratio. This factor was an impt subject of discussion in "Random Search" 6.28 – ~9.40 & I think t. continuation, of ≃10.01 – 12.20 was v.g.]

T. forgg. deals ~~……~~ w t. TM2 problem: i.e. what is a most efficient search strategy.

Hvr., another affect of deciding to try a particular Pam on B, because a certain set of measurements on B had been obtaind, is that t. pcost of that particular Pam may be increased sometimes. This is through a renormalization process, which is desrbd in PW 22.20 – .40 ; 25.01 – .10 and 25.20 – .34. Occasionally, deciding ~~to~~ use a certain Pam$_i$ because ~~…~~ measurement $\vec{\alpha}_0$ was obtained, will ↑ t. pcost of B w.r.t. Pam$_i$ over t. pcost of using Pam$_i$ directly, on B.

---

w.r.t. T. "efficient search strategy":

1) We may find it convenient to obtain t. proby distribn of $\left(\frac{pcost}{ccost}\right)$ of Pam$_i$ wrt B, directly, as a function of a fair no. of params (≡ measurements) of B, (instead of obtaining a joint distribn of compression & ccost as in 2.32)

2) Using t. forgg. ~~……~~ (or practically any other) efficient search strategy necessarily Biases t. search, so t. proby values obtained, need not be correct on t. average. (See PW 19.01 – 21.19 for discn. of ~~……~~ how & why such techniques tend to bias a search)

Rsm 7·27·78
REV

.01 For [RANDOMSEARCH] 13-15 (7·24·78)

13.01 : This section is a Theoretical justifn. of RANDOM SEARCH being optimum. The idea is that with a given [redacted] Umc, M, and no previous corpus,

.04 t. ~~search~~ optimum search must be ROTE or ~~≈~~ RANDOM. → .28

F.N. : By "rote search," I mean all codes of max pcost are tried first — so t. codes are tried in pcost order, using some sort of computation Bound (will is) or [redacted] membership in some complexity class, to decide that certain codes are "meaningless" (or to simply cut off compn. on a code that has not yet converged). [redacted] One Possl. C. B. could be that the ccost since t. last output symbol is > some threshold, To.

"Random search" is some modifn of "rote search" in which t. probty of a hy pcost [redacted] code trial will be an ↑ funct of its pcost. We may or may not want to do these random trials w. replacement. Ordinarily rote search gives us more pcost/cost than random search. Hvr., random search may sometimes be of especially low ccost for certain kinds of Hardware; Also random search gives a "Mixed strategy" which is sometimes useful in a universe in which TM has an "Opponent".

For Most [redacted] present descn., Rote & Random search are about equivalent.

.28 :04 : Here by "optimum" I mean most pcost derived per ccost expended. The Great Idea of / RSearch 13.05, is that if we have coded part of t. corpus, we can modify M so that it completely includes all info obtained in coding that part of t. corpus. W.r.t. this new M', t. optimum [redacted] search is again random.

Rs 13.10 — 14.02 analyses t. foregoing idea in more detail. Even when M' is based on only [redacted] an incomplete [redacted] set of codes for t. previous corpus (which is true in all practical situations), t. argt. holds.

Actually, t. [redacted] argt. of 13.01 — 14.02 is not completely rigorous — (two parts of it are).

8.24.78
Rev

Study for Review of R.S. 50.28 – 62.08

50.28 – 53.11: Listing of 5 major areas of TM work: This would be good for next yr's. proposal!

53.12 .... A new idea on "Block coding". Sequencial coding of Blocks. — One Block after t. other, but each Block need not be coded sequencially. I want to get 53.12 – 62.08 clearly stated. As it is, it is a not alltogether clear idea.

53.25 – 31 This may be a very impt. idea. The Obs. & Pems & other operators are all treated on an = footing, & they are selected on t. basis of which one will ↑ t. Gorc most. T. Gorc is (t. expected pc/cc of t. corpus or t. expected pc of t. corpus w.r.t. t. max cc. available) w.r.t. t. state of t. system.

(or we select so as to ↑ E (expected value) of t. Gorc — & t. Gorc then leaves out &

A less el. approach would consider this as a sequencial maxzn. problem — in which sequences of operators are considered. I think this may avoid certain "local maxima" that may occure — tho rigorously, local maxima usually can't occure: We... just chose t. best operator at each point, & it is unusual that all operators should produce no expected affect on t. Gorc. — "At our wits end" we can always construct new Pems.

Hvr., t. advanced consideration of sequences of operators is a form of "planning" & is, ordinarily, a good way to do things.

Hvr. in 54.01 → .25 we have this function, F that looks at t. state of t. system & decides on what ob or Pem to invoke, & t. idea is to optimize t. F, obset, Pem set combination.

There seems to be a big contrast in approach betw. 53.25 – .30 & 54.01 – .25

55.30 Devrs that most codes normally usd are m t. form of Block codes (e.g. lin. regn., coding w. defns., coding t. Pure Bern Seq.)

55.35 suggests that I investigate how these coding methods were devrd., so I can express them in some hy pc notation. I.e. factor these coding methods into a set of hy pc abss.

This would be a useful approach to understanding Block coding. — Also, it could be usefully studied for other problem types (e.g. t. entire list of 50.28 – 52.40. T. problem of understanding "Block coding" does seem to be impt., & this 55.35 – 46 does seem to be t. rite approach.

Impt.

56.01 — .30 Expands t. forg (55.30-.40) idea.
56.38-57.09 - This is on that idea of "pc w.r.t. a situation" or "w.r.t. a CB".
I think this is an impt. idea: I did write a lot on it - say within last 2 months in R.S. or P.W.

57.10 - 59.22 T. "software factory" idea as one possl. set of probs. & source of good ideas.

59.23 - 60.08 A list of what needs to be workd on for a useful (for me of, say 1979) review of Block coding.
60.16 — .29: How Bern coding & "coding w. defns" are simple cases of "Block Coding".
60.30 — 62.08 An attempt at a new formalism for induction: We end up w. something that seems relevant to Horse Race Predn. Initially, t. idea is that a fund. operation is counting (operating on t. s.c.) objects — & each such operation is defind by an algorithm that tells how to recognize t. object being counted. These algorithms will have pc's based on their definitions, in t. usual way — also on how freqly they were usd in t. past.

61.14 gives a tentative guess of how this kind of formalism mite be usd for predn. I don't have any clear idea as to how reasonable t. formula is!

.20 If A_i & A_j are ~~diferent~~ really t. same identifyer w. 2 difrnt dcrns, t. formula is O.K., since $P(A_i)\ A_i(c\urcorner_i)$ say $+\ P(A_j)\ A_i(c\urcorner_i)$ [which $= A_j(c\urcorner_i)$] $= (P(A_i)+P(A_j))(A_i(c\urcorner_i))$; $[P(A_i)+P(A_j)]$ is t. truer pc of t. object dcrbd. Perhaps similar reasoning will apply if there are necessary rigorous/mathical constraints betw $A_i(c)$ & $A_j(c)$

.26 [indepy of c]), so t. formula would be O.K. anyway.

Hvr., see RS 62.20 for a serious counter argt!

62.25 Concludes that eq. 61.14 is wrong, but mite be in a good direction: Suggests expanding, exploring Bern. seq. coding in that genre. This may be a very productive idea!

8 25 78 REV of : RS 50.28 – 62.28

1) See Rev 5.01 – 6.40 for prelimy good running dcrn. of what is contnd.

2) T. (most impt.) main point of this section is 55.30 ; i.e. An impt. idea is t. concept of "Block Coding": This means we obtain a new chunk of corpus, & try to code it using any tricks we know — & usually these methods are not "sequencial". Hvr., "sequenciality" does enter, since we do not change t. pc's of abss usd in coding, during t. coding of t. Block. These pc's are updated befor (& after) each Block is coded.

Examples of such non-sequencial coding are: a) ordinary Bern seqs coding in which we first count t. symbol freqs, then devise suitable defns, or devise suitable **FOR's** to express t. pcm that we've **found**. b) Coding w. defns c) linear regn . . .

$(1.05)^{75} = e^{\frac{75}{20}} = e^{3.75} = 2^{\frac{3.75}{\ln 2}} = 3.96$ ~ 4 bits.

In order to understand & usefully genz. such

.20 "Sequencial Block coding", I want to first do a fair no. of examples of it. **T.** way this is done, is to write out t. coding or prodn. method, then try to find out how it could have been dcvrd, & t. reasonable pc / and reasonable cc, from simpler **concepts**........

This is done w. a fair no. of cases, like — are listed on 50.38 – 52.40 + "on software factory" 57.10ff

What I end up w. is a large set of relatively simple concepts from which t. prodn. methods (or prob solving methods) usd in t. probs of 50.38 – 52.40 (+ sw factory) could have been dcvrd.

To aide with this "working backward", I also work on t. "working forward" problem of elementary tng. seqs. of various kinds. The work of .20 helps point t. elementary tng. seqs toward t. needed concepts. Also, t. work on elementary **T.S.** should help directly w. understanding how to do t. probs of .20.

→ RS 59.23 – 60.08 gives more specific things needed in this Review

RS 62.29 – .36 : Has cross Refs to "Search Strategy" 1.01 – 13.01 : These are very much like t. forgg stuff on "Block coding", but a slightly difnt. view (perhaps): But t. genl. conclusion is t. same: i.e. RS 50.30 – 56.30 ; & Search Strat 12.01 – .30.

82678 **Rev of RS [50.23 – 62.40] & S S (Earth Strategy) [1.01 – 13.01]**

T. idea-as-a-whole, is an integrated study leading to T.M.

RS 50.28 – 53.11 lists 5 areas of research: They are:

1) General CBI research; Theory, some aspects of applicn.

2) Elementary trg. seq. construction — in a variety of fields if possl.

.10 3) Starting w. some task that we can get a machine to do (say typical A.I. "production" problems). (was thinking of "Blocks World"? 46.583) We work backward to find a set of more basic abss. from which these production pgms can be derbd. Such a set of abss. can be obtaind by asking how these production prob. solns. could have been dcvrd. ... This is "working Backward". Using these abss. & assoc pcs., we try to "work forward" (as in (2)) to dcvr/ ~~better~~ solns to more advancd problems, or better solns. to t. old ones.

4) T. prob. of TM_2: How to improve TM_1's mode of operation.
— This is a special case of 3), but is gn. special attention because of its importance.

5) Examine impl. cases of induction as in a) child learning b) History of sci. c) Perhaps Psychol. literature. d) Books on "How to solve probs." a/o "be creative" e) Work on induction by others: Such as [redacted] in A.I., pattern recogn., Learning of Languages, etc.

RS 50.28 – 53.11 has some expansion of t. forgg. ideas: also places to get info. & ideas.

6) Learning to "understand" English: QATM is one aspect of this problem.
S This is a special case of 3), but is gn. special attention because of its importe.
General language learning. Relation of a 2 or 3 dim scene to its English dcrn. ← (This is a special form of Mach Translation), Winograd's "Blocks World", etc.

.28

T. work of 3) is in t. spirit of **PMTM**. One theoretical problem.... is to decide just how t. info from t. various modes is to be combined.

A long time ago, I did some very (Manic) exciting work on PMTM. This was befor t. fire & this work is refrenced in t. reviews I did after t. fire. At that time, I felt that this was a v.g. approach & I had lots of ideas on just how to do it.

.37 My present state of mind is pretty much that. I have been at this point several times in t. past. It is most certainly at least a fairly good approach to T.M., and I keep returning

82878 **REV.**

yellower
turn p11.

.01 to it after various campaigns on "short cuts"

I want this note to be a "Book Mark", in the sense that after some work on the Proposal (& perhaps Minic) I will come back to this point & continue. Hvr., I want to be sure I can do this, by leaving enough notes & refs., so that the point of this approach is quite clear.

I'll list the 6 points of Rev. 8.01 — .28 again, going into more detail & giving more refs., so the Me of 1979, say, can pick this up again.

1) General CBC (≡ CMC) research: Refs: RS. 50.29; 57.15! Some problems: Details of Mxum. for linear, N.L. regn., Powers spectra. Mixing operator, unorderd & orderd data for PMTM; V.H. problem of whether shortest code is within bounded distance of best; To what extent is P_M' not unique soln.? ; Perhaps work on optimum search strategies in terms of dollar cost. The general problem of optimum search (Much work that I've done on this, usually on "random" or "rote" search e.g. KGS and NKGS.).

RS. 56.38 on ... Unnecessity of "CC" — But also work in SS note on ... reasonable CB. from "OOL" work. (S52.30)

.22 2) Elementary trng. seq. construction: The 1962 Chicago paper is one good descn. of this: (see TS 1:01 ff 32778 for recent discn. of that paper). All the T.S. (fill) on this. ("file"?) Also a fair amount is in PRO. See RS. (52.28 & 52.32) for a brief list of some trng.-seq. types.

In (3) one goal is formalzn of strategy so IPC ideas evaluated for min hardware cost. Also see RS 53.01

My present view: While this approach to TM, would eventually work, it would require a very large trng. seq. to get to any interesting points. In (3), I hope to get to interesting TM w. less elementary Trng. — But I don't have as clear ideas on how to do (3) as on how to do (2), so work on (2) is meant (partly) as a "**STUDY PROBLEM**" for (3).

— Also (©), (3) can be regarded as a "study prob" for (2). (3) can give (a direction / directions)(a goal / goals), for the trng. seqs of (2).

8.37–9.01 This can be expanded: Why the "shortcuts" were abandoned. Also list some of them: SFTM, Ideas in IPC, KGS, NKGS, PREsent RS. Search Strat.

3) This is the Big Point of t. present approach.
Refs: Rev. 8.10; RS (50.38 – 51.03, 51.25 – .40; 55.30 – 56.30 [very relevant]; 59.30 – 60.08)
Search Strategy (1.01 – 13.01 [12.01 is very good] but more narrowly 3.30 – 9.40 & even more narrowly 3.30 – 5.01)
Also, the very old work on PMTM is very relevant (see "FIRE" notes).
[I may also have done relevant work under title: GIM ("Genl. Induction Machine")]

RS ($\tilde{\tilde{2}}$6.01 – 49.40) is relatively easy to browse thru fast! Has much discn. of t. modes of PMTM. (Apparently only a few are needed in order to do just about all human-type tasks.) Also, has much discn. leading to t. presently contemplated approach to PMTM. [crossed out]

What (3) does is to take various ≥ intelligent tasks that a machine can do (we will later give some examples of such tasks) and factor t. pgms into more basic concepts. These more basic concepts are to be further factord. What we want is a max pcost representation of this set ~~of [crossed out] perform~~ of task performing pgms.

T. factorization will be of t. entire set of pgms — (hence PMTM):

One major method to find such factorizations is to try to find ways in which t. task pgms could have been devd. — trg. seqs: from previously less able pgms. This part of t. process is calld "working backward".

When we have a fairly good set of factors & assoc. pcosts (pcs), we can then use these (in random (or pcost orderd) combination) to produce trial solns. to new, more advanced problems — a/o "better" solns. to t. old ones. This soln. of advanced probs. is calld "working forward". It amounts to starting t. elementary trg. seqs of (2) at an advanced point. In order to be able to do this, it would be well to have some experience w. (2).

Advantages:
1) goal is form for shortcuts
2) direction for trg. seqs.
3) Faster achievement of useful TM.
4) Use of work of others: More interest by others.
5) Less chance of serious errors in basic ideas than in "trg. seqs" approach.
(6) More ΙΙ TM work is good: prevent serious deadends."
7) Easier to publish "readable partial success" papers.

3) (continud): Some impt. probs. of t. desird type:

a) Doing induction via Bern freqs, binary defns, linear regressn, n-l. regn., clustering .... factor these ind. methods: prest-order combns. of these factors gives new trial induction methods. Factors found partly by asking how these ind. methods could have been derrd.

b) simple Inductive optzn: Gn. a set of object, G pairs [redacted] $[O_i, G_i]$ to devise a object O_j ∋ t. expected value of O_j's G is max (or some other funct. of t. G distribn. of O_j is max).

.11 c) "Blocks world", induction of relationship betw. a [redacted] English (or simplified Eng.) dcrn. of a "scene" (in [redacted] 2 or 3 dims) & t. scene itself.
Induction of Mach xltn. rules betw. 2 computer langs or other langs.
QA [redacted] (Questn. Answrng) machines.

→ d) Winston's induction prob: Genz of my soln. to more general concepts that & simple Boolian operators.

e) Grammatical induction w. various kinds of Grammars — perhaps see (or Buy!) K.S. Fu's book.

.20 f) [redacted] Playing Chess, checkers & learning to play these games. This prob. is perhaps a useful "study problem" for t. very impt. problem of sequencial optzn.

[redacted]

g) Software factory RS(57.10 – 59.22) is recent ref. I think there's a much longer, much earlier dcrn. of this device. Here we try to factor & assign pcs to concepts used in t. soln. of real programming probs by humans. T. better t. assignment of pcs to [redacted] proposd solns, th. easier it is for t. programmer. When we start, t. human pgmr. does most of t. work — As t. system matures, more & more of t. work is done by t. machine.

This problem is of much commercial interest & could be t. basis of a actual "SW. factory".

.33 h) Sequencial optimizn: This is t. problem of planning experiments, w. a cost funct. for expts., & a goal for [redacted] a sequence of operations using t. info obtnd by t. expts. See RS 36.02 ff for a dcrn. & applicns. The TM_2 problem (I.E. prob. of improving TM_1) is of this sort. (f) (.20) seems to be related to this problem.

Whitter than B. P.9

Sequencial Maxzn. prob: T. problem reduces to a simple inductive optzn. prob: i.e. to find a function relating past trials & their successes, to t. new trial to be made. Recorded RS 40.20

Mention Game playing as a PMTM Mode that seems close to t. seql. optzn problem. α-β heur? Recorded

4) Language learning: T. goal is to get a machine that can read any book in English, & understand it, in t. sense of storing its info content in a useful, accessable form. T. criterion of having done this is a) Ability to correctly answer Q's about t. books b) Ability to use t. info in t. books in solving problems. Some intermediate goals & study probs are in (3)(c, d (to some extent), e) [Rev 11.11]

[Note Pratt's remark! "It is easy to learn a second lang. after one knows a first lang"]

This area of work is included in 3) but is mentioned here because of its special importance.

5) TM_2 : T. problem of improving t. operation of TM_1:

This ~~work will~~ part of TM will be initially done by me. It is not an immediate need — i.e. we can get a functioning TM w. only a rudimentary TM_2.

TM_2 is a special case of h) (11.13) [Rev] but is mentioned here because of its special importance in a very hy level TM.

Game playing (e.g. Chess) [see Rev. 11.20] seems related to this prob.

↙10.01 – 11.40 ↙9.22

Why [illegible] active work on 3) is a good addition to 2)

(a) "Shortcuts" for TM: Over t. years, I have tried to devise various (usually formal) systems in which TM could be pgmd in a very simple way, so that only trg. seqs need be inputed, & t. machine itself mite be realized w. very fast, very cheap hardware.

Some examples of such schemes: SFTM; Ideas in recent ICP; KGS, NKGS, RS(1.01 – 62.40)…. (on Random Search [illegible])

Trouble was, I didn't have any clear idea as to what kinds of operations on ongoing, useful TM would be using, so I really didn't know in what directions to push those studies.

The projects of 3) would give such needed info, & make possl. work on th. hardware & software shortcuts that I'd contemplated.

& This would also be done by 2), but not in a way that is as relevant to a "final TM". Use of 3) gives more possl. clues as to t. form of a SFTM or any other "short-cut" form.

— (b) T. trg. seqs. of 2) are directed toward "hy level TM behavior. The tasks of 3) make this goal much clearer & are good sub-goals to aim at, thus simplifying t. task of 2). [Also work on 2) clarifies work on 3)]

— (c) I think that getting a hy level TM via 2) would take an enormous amount of trg. — that working on 3) would reduce that work considerably — particularly if I can, in 3) use t. work of others in A.I. & patt. [illegible] recogn. In patt. recogn. this mite get other people working on CBI — but not in A.I.s, where the paradigm is too well fixed — or more specifically decided against probablistic models.

— (d) [illegible] · By [redacted] pursuing 3) as well as 2) we have a larger no. of ∥ approaches to T.M. If any one component of 2) or 3) fails or dead-ends, t. effect is far less critical wrt t. over-all task.

(e) For a sponsored project, it is easier to publish understandable "partial progress" papers on 3) than on 2), because [illegible] t. projects in 3) actually do useful (or almost useful) things. In 2) a trg. seq. run that I would consider to be very successful, would be hard to sell as a "break thru" to most editors.

82978 **REV:**

When I want to return to t. main line of TM work (i.e. not PRO), Read Rev 8.01 – 13.40 carefully, & go over t. refs. refered to. T. material reviewed is what I want to work on & it is t. desired approach.

On returning, go over each of t. 6 points of 8.01 – .28 (as expanded on 9.01 – 12.40). Write a list of t. main problems in each area & try to tell how far I've Gotten on each. Try to find appropriate sections of TM notes.

T. main thrust of work will be Points 3) (10.01 – 11.40) & 2) (9.22). In t. case of 2) get a list of good trg. seqns to work on; Try that v.g. trick for (Try to find out!) muchly speeding up arith learning. Try to find other good elementary trg. seqns.

for 3) Try to find a few probs to get started on. S.W. factory is nice, but may be too complex: but investigate it. | 1.20.80: using Forth, it may not be so Diff(?) "FORTH mc" may be an approach to such a S.W. Factory!

SN (Many Many yrs. ago) I made a long list of Many probs. to which A.I. approach was useful. Perhaps try to find that list.