MTM ,MTM w. a few arrows allowed in tap seq.

||| ||| ||| H.C.TM.

TM_0, $TM_{\frac{1}{2}}$, TM_1 are dcrbd in <u>Plan 336-337</u>.20

This section develops $\overset{MTM}{TM_0}$:

Lots of general good ideas on TM's.

$$\boxed{\text{Part}\,\gamma\ 338-393}$$

Aug 7, 61 TMY.

Dart.

(339?)

.01 : : The following will be a development of TM₀ agenda as dcrbd. in Plan 336.01 - 337.20

TM₀ : Th. operation will be something like this: TM₀ first looks at th. input, makes some observations; On th. basis of them, he decides what operators to employ on th. input to produce th. output.

When a "new type" of input problem occures, sometimes th. ~~w~~ previously adequate operation method leads to either ① meaningless results (operations that are impossl. - they cause th. machine to stop or to get in an inf. loop) or ② wrong answer. Then, TM₀ tries to find a "signif" difference betw. th new input and past inputs. This may require a few more "new type" inputs. ~~th~~ Once successfully recognized, TM₀ trys to find new operators that will properly deal with them. This is done by ~~using~~ try old motor ops, or combinations of old sub-ops.

Under certain conditions of recurrent frustration, TM₀ will try "Back tracking" (Plans 333.02 & Plan 334.20).

O.K. now suppose we use a long seq. of arith, alg, etc. - and this TM₀ does learn it O.K. - Then we make a TM₂ ← Def: Plan 336.05 that works O.K. - Now just what sort of TM would fit into this, and what sort of hills could he climb? - could we climb all kinds of hills?

Aug 8, 61 TM8

Vart.

.01: 339.47 So, we "pas a curve" thru (or near x y pts of known hy G. We try to get for which th. mean G on that curve (average past and future x inputs) will be max.

.04 N.B., this model is "noiseless" s.g. it has th. ~~a~~ same G → given to a I,O pair whenever repeated. objection to plan 334.31 [do, by making th

.05 This model, also an impt. ~~[illegible]~~! that if we make a trial curve by fitting thru th. hy G pts. only, we may get th. rite form for th. curve ~~[illegible]~~ around an x region that is rather insignif — in th. sens that few input x's occurs in that region. Such an trial would soon be discarded empirically, since many ~~x pts.~~ inputs resulting in y's of low G would soon come in. But it is conceivable that one would continue to create trials of this poor sort.

Hwr: after a few such bad "trial curves", we would begin to have lots of ~~[illegible]~~ x, y pairs of low G, in th. most common x regions. ~~[illegible]~~ It will then be more dif'lt. for a gn. curve to "fit" th. few x, y pts. of known hy G and avoid th. more numerous x, y pts of low G. — A fortiori if a "closeness criterion" in th x, y plane is used when

.35 a curve doesn't exactly pass thru a known x, pt. → 342.05

I suspect that while this TM. with a tape seq. and an ~~[illegible]~~ external G, can be viewed as an hcTM, it is perhaps Def. best viewed as a special kind of TM i.e. a Reinforced TM — i.e. rTM. or RTM.

One way in which th. gen. h.c.TM ~~#~~ differs from RTM,

≈ 1.31

Aug 8, 61
Dart

TMX

.01:338.40: TM_4: It would seem that as a [illegible] of hill ima, th. method of <trying to find out what of th. known "high points" have in common, and a new trial in this ~~[illegible]~~ common mode> not very good — since all one can expect is another trial with about th. same average — not a hyper pt.

~~Hwvr~~ Hwvr — I'm not so sure that this is what I'm really contemplating for TM_1 ~~[illegible]~~ →(i.e. plan 334.31–33~~5~~.03)

This → idea; is essentially a special method of h.c. Which is useable in situations in which a typ. seq. exists.

I.E. say we have a bunch of I,O pairs of know G's. We ~~then~~ pass an operator thru ~~[illegible]~~ a set of pairs of hy~~[illegible]~~ G. This op. then has a hypermean G than th. entire set of I,O pairs (providing th. samp. sz. is large enuf and th. op. dern is ~~of~~ — low enuf rcost.) If th. smp. sz. is fairly smallish, th. variance of th. ~~[illegible]~~ prediction of "true mean G" will be largish — which is desirable — inca than we have a good chance of very large true mean G.

.30 A model for visualizing th. system:

A 3 dim. hill: x, y, and G are coords

x is input, y output, G is th. G of th. pair...

There is some unknown surface relating x, y and G.

Also th. statistics of th. inputs x, exist but we shall disregard them for th. present.

An "operator", is a curve relating x and y. in th. x y plane 3 in,

Aug 8, 6
Part ,

0/1340.40:1 [illegible] the [illegible]M asks questions and RTM [illegible]
RTM can be looked up as a noisy hill in whic[h] proposed off[spring] is given a G value for each trial, this G value being a (very) noisy estimate of the [illegible] future mean G of that-sop. — but the "noisy hill" view of RTM, is a very ruff, very incomplete view. A More comp[lete] view is discussed in Cod 273.1.01 – 311.40, A very good model for this view, is 339.30 ff. The optimum model here would devise a problistic density function for the x's of ?, devise a problistic form of the function G(x, y), and from this info, obtain an x,y curve of optimum expected mean G, or optimum with resp. to any other Gore.

So, it mite well be, that I can use the methods of Plan 334.31–33.0 for RTM, and [illegible] [illegible] would be a reasonable continuation of the [illegible] was contemplated for MTM (= M_0. There are [illegible] [illegible] [illegible] [illegible] given the power of the.

Very general [illegible] that [illegible] [illegible] (i.e. Plan 336.30) Yes?

(2) Can this RTM be used for ∃S? ← Yes. and if so, how? — and if not, [illegible] the present research path be modified so that it will?

.35 Well: RTM ~~[illegible]~~ can ultimately with suitable trg seqs., be made to ~~do~~ ~~operate~~ do a things a human can do. We can teach him Q,A, we can teach him to understand and give replys in English — we can, when RTM has been suffly created, ask him to improve a certain RTM design (i.e. his own) with resp. to a certain CIRC, and we would be able to do this

Aug 8, 6.   T' 8 ,

Dart.

.01: 341.40 Th. Great Advantage to working on such a RTM, ra

UcTM, is that rTM is close to human, ~~and~~ it i

~~to [illegible]~~ for me (a human) to devise heurs.

.04 by starting and perhaps modifying my own heurs.

.05 Th objection of 340.05, to Plan 3.34.31, will be partially (wholly) met if we do use a "closeness criterion" and take as many of Th. known (I,O) pts into account as possl. In estimat ~~in~~ expected mean G of an op., we will give low wt. to Th. G's of (I,O) pts "far ~~[illegible]~~ off Th. curve" (see 339.30 for Th. model referred to). As we get more pts. in the region where Th. curve is "bad", we will get more x,y pts. "near" Th. curve and these pts. will condemn Th. "apparently good" curves discussed in 340.05-.35.

Also, a more complete discn. of ~~[illegible]~~ RTM such as 339.30 analogizes ~~will~~ (and is treated in Code 273.1.01 – code 311) will elim. this difty. Such treatment takes both Th. x statistics and Th. expected form of G(x,y) into account.

Now, a course of action: ~~[illegible]~~ draw up a RTM close to intuitive, and close to present plans for $TM_{\frac{1}{2}}$, and criticize it on th. basis of Th. more complete model analogized by 339.30. My impression is that, at first approxn, such a T.M. would be very much like a TM_0 "passing an op ~~[illegible]~~ thru" Th. best known (I,O) pairs — Also, perhaps avoiding certain bad (I,O) pairs. ~~This mite be done~~ To evaluate Th. expected G of such ops (befor using them ~~to~~ to create responses),

343.01

Aug 8, 61

TMd.

.01; 342.40: it may be expedient to evaluate them over a ra
of (I,O)'s using "closeness criteria".

Perhaps another ~~is~~ step betw TM₀ and TM₁ :
As Imp seq., TM is gn. certain rite I,O pairs but also some ~~I~~ I,O pairs that it must avoid. We must then find ~~a~~ a low rcost op. that passes thru th. "rite" pairs, but avoids th. "wrong" pairs. I'm not so sure that this is much extra for ~~I~~ TM₀ to do — since in a MTM, there is only 1 rite answer, and all others are wrong. So in normal TM₀, if we gave th. rite answ. for certain (I,O) pairs, this would be equiv. to saying all others are wrong.

– But maybe we can find problems in which there are > 1 rite answ., and so, giving some "bad" answers would contain info that knowl. of a single rite answ. would not contain.

I think that perhaps th. concept of "closeness" is impt. here, in what I want — so we want to get as far as possl. from th. "bad" (I,O)'s. Hvr., th. fact remains, that th. info that TM has may be only that certain (I,O)'s are good, others bad, and he may not know what th. good responses are to certain I's, tho he may know certain responses to them are bad. Th. problem of using this info in an opt. way, to get a low rcost op. to pass thru some and avoid others (I,O)'s is certainly a fairly well defined problem

Aug 8, 6

Part

.01:345.40: A possl. defn. of th. soln. We wa ∋ in th. future, th. ratio of th. mean no. of good (I,O)'s i pass thru to th. no. of bad ones it will pass thru, w max. We may further stipulate that ~~all~~ every / ∂ to a gn. input is either "good" or "bad".

Another possl. "study TM" Betw. TM₀ and TM₁: Several (I,O)'s are gn., and TM must make his responses as "close" to them as possl.

.15 — — **Impt. Q:** Presumably, I will be eventually — i.e. not rite away in th. tng. seq — reenforcing TM₁ for speed of response. Then th. abss that TM₁ will use will, presumably, be selected partly with this goal in mind. Th. Q is — how much "choice" does TM₁ have? — i.e. can I use a fairly simple search procedure, and still have TM₁ end up with abss. that work well with even that simple search procedure? Can TM₁ compensate for a poor search routine and make th. all over performance (after much tng), as good as that of a machine with a better search procedure? ~~I.e. can, we use slower (at first) learning to compe~~ Can TM₁ "learn" to get around a particularly poor search method?

If so, it ~~was~~ mite be easier for me to start out with a poor (tho "adequately above threshold") search method, and have TM₁ "learn" to use an essentially better search method.

This is what Dick Friedberg mite have actually done — but he didn't know what he was doing so he didn't know just what to optimize, or whether his search time was adequate, or what Tng. seq. to use and with what speed.

Aug 8, 61 TM_1

Part

.01: 344.40: I must do this in more detail, to see just how (if necy) divide TM_1 into a "TM_1 part" and [illegible] externally controled by me "search part."

It is possl. to discuss th. above either on a fairly theo level, or get a fairly concrete model of a RTM, and discuss th. Q: with resp. to ~~that~~ model. At th. present time, my preference is to make th. model first, then do th. theo. discn.

.10 <u>Some other things that one mite "reward" RTM for:</u>

Good! 1) Redundancy — i.e. resistance to random mutation in th. computer — th. mutation frequs being ∝ true component error rates

2) ~~lightness~~ Use of less total memory or less total arith. unit. This would make possl. liter, cheaper TM's of a gn. "capacity" — or a greater "capacity" per dollar a/o per pound.

3) Total speed — (of course!)

1.111 So what must be done now ① Some rather detailed dern of TM_0 with emph. on th. search method to be used. ~~Translati~~ 1.② Dern of TM_1, ~~then~~ giving details of how TM_0 is to be modified to get TM_1. — with emph. on search methods.

— Stick close to intuitive derns (i.e. derns of my intuitive processes) — <u>Then</u> make more rigorous: <u>Then</u> see ~~when~~ what can be done about 344.15ff. —

346.01

Aug 10, 69 TM8

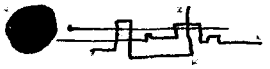
Dart

.01: 345.40: For a search method, at first just try a simple with choice $\propto$/ apri probty: of object. This is slow, but TM_0 on this basis, and then TM_1. Even this poor kind of search will probly cost much lite on 344.15 ~~#~~ by the time I get to TM_1.
Hvr, in all cases, make estimates of search time."

Aug 12, 61

LMγ

Dart



.01:345.40: A lot of <u>very</u> <u>good</u> work on MT

Epm ≐ Thao of MTM ≡ MTM Thao β768.01 → β

In particular, ≈[β1084.01 ff to β1087.40] discusses a Dart MTM using ordinary IBM704 orders..

This dvlpmt was discontinued around, because I didn't have a clear theoretical understanding of th. problem. I think that I may have one now — certainly enuf for MTM.

Using th. ~~[illegible]~~ notational ideas of EPM β1084.01ff, I think that I mite get a TM to learn $A\left(\begin{smallmatrix}\div\\ \times\\ +\end{smallmatrix}\right)B$. Hvr., th. genl. form (i.e. form of these probs. will depend very much on what probs. I intend to give in th. future of th. trg. seq.

Some types of probs: $A \dot{+} B$; $A \overset{\div}{\times} B$, $(A+B)C$; $(A+B)\div C$

$(E+(((A+B)-C)\times D))\div F$. $A+Z=B$ (solve for Z). $A+B+Z=(C+D)\div E$.

$A+B=C\times Z$ ⌐ $A+Z=B+2Z$
Equs. in 2 unknowns, then 3 unkn, then 4, etc.

Differentiation. of alg. expressns. (> 1 answ. possl.)

Trigonom. identities. (" " " ").

Look into th. Q of what sort of backgnd TM needs to Learn to ~~do~~ solve linear eqs in 4 unknowns (say), after having learned about solving eqs. in 1, 2, 3 unknowns.

Probly (since TM₀ will start off with my <u>human</u> ~~[illegible]~~ observations of "regularities") I should somehow ~~give~~ teach TM₀ a concept of what it means to solve an eq. — Perhaps somewhat in th. spirit of th. book "<u>Lincos</u>".

One way to write such a trg. seq: Assume (as in "Lincos") that th. pupil is very sophisticated mathically — that he already has th. ~~[illegible]~~ needed

Aug 13, 61

TMγ

Dart

.01: 34~~46~~40 concepts (≅ defns.) and only needs to figure in W is th. notation of th. input message

Perhaps start TM$_{\gamma}$ with positive integers only — then introduc neg. integers and zero, rational fractions, irrational nos., complex nos.

.07 SN An old, but very impt. observation / e.g. EPM β947.30: That heurs. and "regularitie can be looked upon as th. same sorts of things. A "heur" is a prefer reordering of trials. A "regularity" is a modification of rel. proby. of trials due to an observed possl. recoding. That th. "heur" in Q. worked in th. past, and is expected to work in th. future means that it corresps to an acceptable means of recoding, and is a useful "regularity".

A kind of impt. heur. would be an ordered list of "regularity types" to look for in a corpus or in th. code of a corpus.

In th. case of MTM, say $O_1, O_2, O_3 \cdots O_n$ are successive operators that have been devised to deal will successively larger parts of th. tng. seq.. Then th. corpus for th. "heurs", is $O_1 \frown O_2 \frown O_3 \frown O_4 \cdots \frown O_n$ — i.e. we must find regularities in this corpus, so we may quickly get good trials for

.35 O_{n+1} after we have seen all of $O_1 \frown \cdots \frown O_n$.

A very Optimistic approach to Dart TM: Plans γ1160.20 to ≈ γ272.40 (~10 pp).

There is some Q. as to just what sort of Tng. seq. I really want. Eventually, I'd like to have TM learn English. For a simple start on "English", th. "Baseball" language of th. Lincoln Labs group mite be good.

3.49.01

Aug 14, 61

Dart

TMγ

.01: 347.40: Note, hvr., that there must be some sort of fac
for putting "pure data" into TM — This is so we can
TM obtain an "education" (≡ trg. seq.), by reading conventional books.

One way to feed in "pure data" — just feed in a lit
data as declarative sentences, then ask Q's about th. senten
Th. (correct responses to pure data is null. Hvr., all meaningful
Q's must be answered. At first, make Q's follow closely,
th. declarative statements that contain their answers. Later, make
th. Q's follow more distantly. This has ~~th.~~ effect of making
th. data sentence that is relevant to a Q. be less
certainly determined by temporal closeness alone.

Later, ask Q's whose answs. depend on several
data s's, and upon logical manipulation of these s's.

But first, I want to get some sort of TM_0 outlined,
So I can then work on TM_1.

For ~~arith. reasoning~~ — use Polish notation, with parenths.

:30 What I've been looking for, is some sort of uniform format
in which to present Q's to TM_0.

I think there is some confusion about just what is meant
by reenfg. TM_1 for "speed." There are 2 speeds involved:
① TM_1 is, at any time an operator ② One "speed" is th.
mean time for that op. to solve all probs. in th. trg. seq., We
may want to wt. some probs. more than others. Th. idea
of "solving" a prob. is a TM_0 idea, hvr., that may or
may not be used in TM_1. A loss el. concept 350.01

Aug 15, 61 TMJ

Dart →

01: 349.40! Involves some function of th. / soln. times (for th and th. G's awarded by th. external reenforcer.

(b) We want th. search for an op. of hy G to be f — in th. case of TM$_0$, this problem is clearly defined if we use a Mc search in TM$_0$, we take th. first" consi op. that comes along. Thus our search time / is automatically minimized. → Th. Q. is whether we would, in TM$_0$, automatically end up with abs. types that minimized this search time.

In th. case of TM$_1$, th. goal in problem (b) has to be defined for each situation. What we want is that th. G's (including computation times) awarded to successive trial ops., ↑ as rapidly as poss/. Actually, in TM$_1$ th. / effective G for an individual response, is gn. on th. basis of ① [illegible] th. external G gn. for that response ② the response time — this is th. sum of th. time necy to find a new op (if a new one is desired) and th. time of operation of that op. in creating th. response.

Not exactly!! We may avoid certain TM$_0$ op. trials because they take long to test, rather than because they have low aprip. .35

Th. probs. of 349.30 – 350.35 will become a lot clearer, once I outline a TM$_0$ in some detail. e.g. assume any format, then see what differences changes of format make in learning speed.

Aug 8, 61 TMJ

Dart

.01 : 350.40 ! O.K. ! Consider Th. folg. format:

We will, in this tng. seq., want TM_0 to learn, eventually, equs. of various kinds. Each input to TM_0 will consist of 1 or more equs., some of them containing the letter, Z. After any presentation with a Z in it, TM_0's response, must be to solve for **Z**. for each input seq., there will be a special terminal symbol, indicating th. termination of that problem.

So: ~~a~~ **Inputs** mite be: ~~E~~ $A+B=Z$, $Z = C+D$.

(A, B, C, ect., are random nos. ~~E~~, Z. is a special symbol, as are +, =, −, ×, ÷, etc.)

We have 100 ordered registers for th. input data; (Most ideas ~ to EPM 1085.01 ff)

At first, assume that +, −, ×, etc., ~~take~~ all take ~~[illegible]~~ about th. same time. Use flt. pt, −, +, ~~and~~ × and ÷. This prelim. analysis will not be for any particular existing computer, but will be simply a study to concretize ideas.

Say our computer orders are: (EPM 1084.35)

1) Ad, n   +
2) Sb, n   −
3) ~~Mp~~ Ml, n   × ≡ (Ml, n)
4) Dv, n   ÷ ($Acc \rightarrow \frac{Ac}{n}$)
5) St, n   (Store Ac in register n. ≡ R_n
6) Tr_ϕ, n   If Ac = 0, get normal next insts.
      " " ≠ 0, " " " from R_n
7) Tr_+, n   Same as 6) but inst from R_n if Ac>0.
8) Stop.

U_0 contains 0.

S_1, S_2 ... S_{100} contain th. input.

U_1, U_2 .... contain nos. ~~and~~ identifying special symbols. (=, −, +, (,), etc, Z,)

V_1, V_2 ..... vacant "scratch paper"

Y_1, Y_2 ... output registers.

Aug 13, 61.

TMg,

Dart

.01 : 351.40 ! O.K. so : first set of probs : $\overset{s_1\ s_2\ s_3\ s_4\ s_5}{A + B = Z}$

"C_{U_1}" means" th. content of th. reg. whose name is U_1.

$C_{U_1} = V_i$ means that Reg. C_{U_1} contains th. name, V_1

$C_{U_1} = Z$ means " " " " " " Z

↑ this will be some number

~~make order 6) Be : Co (≡ compare) :~~

~~Co, n means if $C_n = C_{Ac}$, go to next order.~~

— No! — what I would need have would be a 2 address order.

~~address to compare~~ with Acc., another for possl. next instruction.

Use random nos. for each of th. "special symbols".

Constraints on orders. ~~(St, $\overset{s}{U_i}$)~~ ~~only~~ for St,

only (St, Y_i) and (St, V_i) are legal. (we may later want TM_0 to be able to store special symbols of its own devising e.g. (St, U_i) — for i > a certain no. — but this may be unnecy).

~~Ex~~ Th. "special symbols" can only be used for comparison — e.g. (Sb, U_3), followed by (Trg, n)

The U_0 ($C_{U_0} = 0$) is a bit diffrnt. I originally put it in, to be used for clearing, but perhaps (CA, n) (clear Ac and Ad n) would be better to add onto th. set of orders.

But, I'm getting much more detailed than I want to be yet! At th. present time, I want something like:

Problem : $\overset{s_1\ s_2\ s_3\ s_4\ s_5}{A + B = Z}$

Soln : Add C_{s_1} to C_{s_3} and present as answer.

Remember : I don't expect a computer to do a MC search to find solns. to th. early probs. At first, I will solve th. probs., then

353.01

.Aug 15,61 TMJ.

Part

.01:352.40. from these solns.; I will devise special order types, and to work th. set I've solved. With these orders and [illegible] it should be possl. to solve new, more diffl/x. probs, by a MC search by a real computer. I think th. moral to have Me look at th. entire (or much of th.) tng seq., and then try to get a reasonable op. soln. — Then take statistics on that soln. Try very hard to minimize th. ~~[illegible]~~ coding by defining operations, etc., that are not ad-hock, ~~in this sense~~ since they will be ~~[illegible]~~ very useful in th. future!

Aug 16,61: It has been ~~taking~~ me several days to ~~get~~ get into th. spirit ~~of th. idea~~ plans β1160.20 – γ69.05. I'm not quite sure I'm there yet!

For input; I can feed in 2 kinds of "data" ① "facts" ② Problems. || For TM₀ th. "facts" must all be correct. for TM₁ or TM½, there may be errors in th. "Facts". Th. "Problems" are sort of short run facts, and must be used to devise an output. Th. "Facts" do not in themselves demand an output. For a gn. problem, th. output will be a funct. of all previous facts, and all data ~~[illegible]~~ gn. ~~for~~ that problem. In this style of TM, th. machine will not have to figure out what practically everything means in its input. Instead, TM will be given much info about th. meaning of th. input format. This will not make th. induction problem easier; but will make th. induction problems concentrated on th. kinds of regularities in which I'm interested.

Another feature that I'd like this TM to have, 354:00.

Aug 9, 61 T.M.J.

Part 2

.01: 353.40! would be that I could tell him how to work

Th. way I could do this in th. present mode is to [illegible] a soln. to th. problem as simply as possl., using TM's [illegible] abss. that are in existance at that time.

In th. early MTM tng. seq. perhaps no "facts" will be — any info needed for agn. prob. will be obtainable from that problem input, and any info needed from other probs. will be carryed by th. operator that TM. has found adequate up to that time! It may be that we will want all of TM's "facts" to be somehow incorperated into ~~[illegible]~~ this operator. Th. trouble is, th. op. to be assoc. with a gn. fact is not always clear — e.g. for th. fact 1+3=4, shall we mean

1) problem, input 1+3=Z yeilds output 4 }
2) " " 1+Z=4 " " 3 } ?
(3) " " 1 Z 3=4 " " +
(4) " " 1+3 Z 4 " " =

Perhaps there is some way to make all of these possl. In a TM, those of these — responses deemed by TM, to be most useful, would be incorporated into th. operator.

We could put all "facts" in th. form of problems with their solns. Th. disadvantage of this is that then we couldn't give TM a book to read for "facts".

O.k. — well, write a tng. seq. for MTM. Write solns. in fairly intuitive English.

Aug 5, 6,

Part)

.01: 354.40:

Problem:

$R_1\ R_2\ R_3 R_4 R_5$ ~~[illegible]~~ $R_6\ R_7$ ← input Register nos

1) → $A + B = Z \ , \ Z$ This is to mean A

1 or more examples are gn., with ± random nos ~~for~~ for A

see EPM~~[illegible]~~ 1084.01 ff. for some notation conventions.

Soln: Add R_1 to R_3. ~~[illegible]~~ (place soln. in Ac ~~Actor~~ automatically).

1 2 3 4 5 6 7

2) $A - B = Z , Z$; Soln: ~~[illegible]~~

Look at R_2. If $C_{R_2} = +$, add R_1 to R_3

" " = −, subtr. R_3 from R_1

It may be possl. to simplify this pgm. much, if we have a specific set of computer orders. E.g. for 88.IBM 704:

Add R_1 ~~[illegible]~~ → if $^C R_2$ is + Add R_3
→ " " " − sub R_3

or Add R_1 | if R_2 is +, Add R_3
| otherwise sub R_3.

1 2 3 4 5 6 7 ← Register nos.

3) $Z = A + B , Z$

Soln (this includes soln. to $Z = {}_{-}A - B , Z$) <

Look at R_2 (also soln. starting with Look at R_4 is possl.).

If ~~[illegible]~~ $^C R_2$ is =, $r = 2$ | also soln. with "If C_{R_2} is +"

" ~~[illegible]~~ is not =, $r = 0$.

Ad. R_{1+r}; look at R_{2+r}; If $C_{R_{2+r}} = +$, Ad R_{2+r}

" " ≠ +, sub R_{2+r}

1 2 3 4 5 6 7

4) $Z = A - B , Z$ ________

Aug 19 01 My.

$\leftarrow$ Dart

.01. 355.40: 5) We may want to ~~try~~ give

$$\overset{1\ 2\ 3}{\underbrace{5\ 5\ 5}_{\text{arb. no. of spaces}}}\ \overset{4\ 5}{Z =}\ \overset{6\ 7\ 8}{A - B}\ \overset{9\ 10}{, Z}\ \overset{11}{.}\ \ldots$$

so that TM have to devise pgms ~~which~~ which are "position invariant". Such a concept mite be useful later.

6) $A \overset{\times}{\underset{\div}{+}} B = Z\ , Z$ $Z = A \overset{\times}{\underset{\div}{+}} B\ ; Z$ These can be also worked in the manner of 3), with little modificatio

TM —

.5 One of my goals in this prelim. trng seq. is to get enuf good abss (or heurs) in TM so that he can perform some fairly interesting induction with a random search that is not beyond th. capacity of existing machines.

Also, to check on my coding theo. of ind. inf to see .20 that I've not missed some impt. pts.

.21 A list of a few types of probs. that I mite want on th. ~~trnsa~~ trng. seq.:

1) $A(B+C) = Z$ 2) $(A \div (B+C)) + D = Z$ 3) Various random combs. of $+, -, \div, \times$ with parenths, $= Z$; Z on either side of $=$.

4) $Z + A = B$ 5) ~~Z+~~ $2Z = A$ 6) $Z + 2Z = A$

7) ~~Z+ZA~~ $Z = V + A$, $V = 3$. 8) $Z = V + A$; $V = 2Z + B$

(Solve 8 by both substitu. and subtraction).

We want to continue th. trng. seq. and solve by hand, untill we get to a pt. where a signific dif/t. prob. can be solved by random search on an existing computer.

.39 I think that it mite be well to try to solve each new problem by recognition operators, that tell how th.

357.01

Aug 2

4M8

. Dart 73

.01: 356.40: new prob. differs from th. old ones and th. new operation on th. new type of prob. is in th. spirit of Plans 67.01–68.40 —— In particular of Plans 67.40

.03 2:38 AM Aug 21

.04 Th. problem as of now: To get a tng seq. and a soln. th. tng seq. ∋ one can solve interesting problems, using M.C methods on existing machines using th. statistics of this "soln."

Th. approach I have been using thus far, is to just try writing a tng. seq. of arith. That would seem~~s~~ to lead to "understanding" of some fairly complex ideas — i.e. th. creation of some very useful abss.

20 Perhaps a better approach: Find some rather dif'lt. probs. which, for a human, would require some cleverness. Make up a series of abss. ∋ a human who has found these abss. useful in th. past, would tend to use them — successfully — in solving th dif'lt probs. of interest. Then work back to find ~~upon~~ a tng. seq that could develope these abss. in a TM.

Another point of impt: Th. idea of 356.39 – 357.03: i.e. th. construction of "obs" ops. to recognize a particular type of prob., and then devise a method of soln.

⟶ Newell-Simon have a set of abss. and heurs that can solve many kinds of problems What I could do, would be to study their heurs a bit, then devise a tng. seq. that will lead to (at least) those heurs.

My impression of how a simple man-like MTM might work! For th. inputs there would be a set of "obs" ops to characterize th. input problem. After characterizing is decided upon, th. input is xfmd. into th. output by 358.01

Aug 22, 61
Dart

TMy

Tyler 6

01:357.40. th suitable op.. When a new input occures for total method up to that time doesn't work, TM tr
① to find an obs to characterize that input type,
② To find an op. for that input type that gives th. rite

Some of th. work of TM consists of simplifying th. derivs. of th. obs and ops Certain obs may be merged by $\beta\Sigma$, if one can find a ~~comm~~ way to ~~give~~ assign them a common op

New trial ops and obs are formed by combining old obs and ops in standard ways, New obs and ops are tried which, in view of th. previous ops and obs. found useful, are of lowest cost — so they can be constructed Mt. Carlo — wise.

.20 Suppose we have obtained this set of obs and ops with correspondances betw. them, and we can view th. entire system as a sample of that has been created by some stoch lang. or more desirably — that th. ~~each~~ (obs, op.) pairs have been created by a single stoch. lang We then use this same stoch. lang. to create new trial pairs — (tho we can create th. obs's and ops

.35 "separately"). Now — ~~want~~ <as new ~~important~~ parts of th lang. seq. are learned, ~~is Receiving~~ and, in some sense, th "type" of stoch lang. is kept constant ~~[illegible]~~ — i.e. th. method of going from th. corpus (i.e. th. system of obss and ops) to th. stoch. lang and its params. remains const.>, is there any tendency for th. search time to ↓ thru "suitable" obs

.38 and ops being selected? — Probly "Yes" — see 365.35 We could try to improve th. search method and leave th. basic obs and ops. "invariant" (tho

359.40

Aug 22, 61 T.M.

"Dart"

01:358.40: they would effectively change anyway, or we could perhaps
effects of faster search, by leaving Th search
invariant and using obs's and ops that were bet
in this respect. It is this latter that we are most inte
in, and we want to know whether and to what extent
is possi.

It would be well, e.g., if we took a ~~stoch~~ gen. system
of obs and ops, along with Th search method and stoch. lang
and devised an improvement in Th ~~stoch~~ search method. Then
we showed ~~that~~ how Th effects of this improved search
method could have been obtained by devising new obs and
ops as changing Th statistics of Th old ones.

That this sort of thing is possi. is sugg. by organic evolution,
with its apparently fairly simple method, of ~~making~~ modifying Th old
chromo. strings to make new trial organisms.

~~Let~~ Let us at first confine our interest to a TM's
~~with~~ ~~mu~~ — i.e a classical MTM.

.30 What we want, now, is an example: i.e. a typ. seq.,
a set of obs, ops, a stoch. lang and a "search method" —
Then an improvement in Th search method, then a possi.
set of modifications of Th obs and ops, that would be
about equiv. to this improvement of Th search method

Note: It will prob'y become clear that certain types of
improvements of Th search method will be inaccessable thru changes of
Th obs and ops. e.g. various sampling techniques for discarding
~~many~~ bad trials quickly. Th part of Th search process that

360.01

Aug 22, 0

3rt

TM

011 359.40 will be accessable, will be th. improvement of

th. trials — i.e. to get th. first successful [illegible]

.03 nearer to th. beginning of th. search. My [illegible]

all this means is that th. proby values as ga. by th.

.051 aug. become better and better as time goes on

Th. ideal proby eval. meth. would assign proby close to 1 to one possy and close to 0 to other possys. In th. case of Mt. Carl trys, this would mean usually a correct answ. on first try. → 361.02 →

.10

.11 Say Ob_1, Ob_2 . . , Ob_n are th. set of obs used to characterize th. input to decide what op. shall be used on it.

Let Ob_0 ~~be th. ob. that was used to chose~~ (≡ th. identity ob. that always says "yes") be th first ob used in th. trng. seq.

Let Ob_1 be th. first ob used when Ob_0 alone become inadequate .. let ob_n be th. ob. used when $Ob_0, Ob_1 \cdot \cdot Ob_{n-1}$ become inadequate in th. trng. seq. ⟹ Then th. way to use these obs. is this .. when an input comes in, we first apply Ob_n. If it is acceptable, then O.K. use Op_n: If not, try Ob_{n-1} next, if acceptable use Op_{n-1}, if not, try Ob_{n-2}, etc. — finally, all inputs are acceptable to Ob_0. When we get a new input, for which our system gives th. wrong answ., we try to create a new Ob_{n+1} of min. pcost, that will characterize this new type of untractable input.. I think that Ob_{n+1} must be difrnt from Ob_0, Ob . . . Ob_n, or else it could not possibly be adequate.

I dont think this input Ob_i system is optimum — but leaves

361.01

Dort →

TMJ.

.01 : 360.40 : it for th. white

.02 : 360.10 → What I'm vaguely thinking of, is some sort canonical form of th. universal "creative" stoch grammar

.04 which th. prob's all approach "correct" values as th. sample size → ∞ → 362.07

A sort of heuristic, to obtain closer to minimal codings:

After a trng. seq. (for MTM). has been "solved", ~~[illegible]~~ go thru th. trng. seq again, but instead of starting with aprioris for th. early problems, and slowly changing th. search statistics ~~[illegible]~~ in going thru th. trng. seq., use th. same constant search statistics for th. entire trng. seq, that were obtained after doing th. trng. seq. th. first time. After doing th. trng. seq. th. 2nd time, th. resultant statistics can be used again to do th. trng. seq. search. I think that th. code rcosts should ↓ as one codes and recodes again and again in this way. I feel that if th. trng. seq. is long, that th. statistics at th. end of th. first "learning" of it, will be far better than th. apri statistics --

There may be some reason to believe that in a search of this type, that this re-coding method should give some compression. Certainly we expect th. 2nd coding to be better at th. beginning of th. trng. seq. Th Q is, whether th all over statistics will change signify in th. 2nd or 3rd coding. ~~[illegible]~~. If we have a TMo with a search method and trng. seq., then

362.01

Aug 30. U.

Dart

.01: 361.40: This idea should be easy to test empir

Tng seq. is inordinately long, and mainly "hard dat

(tho ~~[illegible]~~ it may work O.k. in This latter case, early part of th. tng. seq. should go rather v

by machine using th. known statistics).

.07: 361.04 → : A sort of ~~[illegible]~~ "example" of a very simple type of prediction method that ~~[illegible]~~ is poor for small samples, but may become very good for large samples, is th. ngm method. — i.e. th. probty of various digits following a gn. ngm. As samp. sz. ↑ we can ↑ n. I suspect that this type of probby does not "include all possl. proby eval. methods", but it may be fairly good for large sample sz.

.20 My impression is that th. most general form of a creative stoch. grammar would be something like: We have a univ. Turmac. ~~[illegible]~~ As input, we have AR. A is a fixed string that dcrbs th grammar, R is an arb. random no. (of fixed length?!) — ~~[illegible]~~ R. ~~is now~~ has uniform density from 0 to 1, except that somehow R is encouraged to be a terminating decimal!?! Then th output of th. Turmac will be a suitably problistic set of strings for th. stoch. gram. in question.

A ~~[illegible]~~ defn. of a non creative stoch gramm. is that it assigns a proby to every conceivable string.

Actually, I can get a better feel for the problem in th. info. way: consider, e.g. th. Obi's of .360.11ff., This particular approach really breaks th. system down into parts. It may not be th. best way to make such a system but its close to th.

363.01

Aug 22

Part 7

.01: 362.40 · way man does it → perhaps not — see .20) Anyway, after along enut tng S

be possl. to get any ~~poss~~ conceivable creative stoch.

for producing Th. Ob$_i$'s. Ob$_i$ is a ~~[illegible]~~ function

inputs that maps them into "yes" (≡ 1) and "no" (≡ 0

.07 ~~Exampl~~ Say Th. inputs is a n digit binary no: Then there

are $2^{(2^n)}$ possl functs. on this input. Th. most general possl.

stoch. lang. for creating these functs is a proby density funct. on $2^{(2^n)}$ points

It is clear that with a large enuf sampl~~ing~~ we can get the 2^{2^n} params

of this stoch lang. by simple freq. counting.

Since 2^{2^n} is usually so large, this is th. riduco ad absurdum

~~of~~ large sample needed — and corresps to Th. simplest type

.19 of stoch. lang. conceivable. → 360.38

.20 **SN** I sort of doubt that th. ob. system of 360.01 ff is

at all close to what man does. Man usually makes a standard

Ob. ← On Th. basis of Th. result, he will either make a new

ob ~~[illegible]~~ or do an op — both of which would ~~result~~

depend on Th. result of th. first ob → ~~[illegible]~~.

On ~~[illegible]~~ Th next ob or ops ~~are~~ work on, and are

contingent on Th. results of these 2nd ary obs or ops and

so on

.37 To fix ideas — consider some of Th. ops and ops used in Th. arith. ~~tng. seq. of~~ 355.01 ff. → 366.35

Th. probs. as of now: ① Th. general MTM system of 360.11 ff. with

Th. Ob$_i$'s seems a bit distant from Th. human ~~system~~ method. There

is some Q. as to whether it could be made to work at all,

and, of course, th. more it deviates from human, Th. more 364.01

.01: 363.40: diflt it is to devise heors for it ← (see 366.35 for what may

.02 ② Th. very impt prob. of 359.30 ÷ 360.10 ff is finding see if one gets th. effects of heors being automatically improved ~~[illegible]~~ in a Mt. Carl. MTM. Th. idea

.04 359.30 — 360.10 is that th. Mt. Carl search process devises a creative sto gramm. for th. things searched for, and, with some stoch gramm. types as so th. predictiods stoch gramm. becomes very good."

Th. problem, then, was sort of switched to th. ~~[illegible]~~ set of Obj. These are a set of ~~ob~~ functs that map th. input into (0, 1) (≡ Yes, No). Tho these Obj's may not be a good ~~idea~~ for MTM, they are a good study problem for this → Q. 363.07 works a bit with this idea.

Anyway, say we get a very large set of strings that are th. output of some (fairly general) creative stoch Gramm. How well can we make a

.20 model of this stoch Gramm? 363.07 — .19 gives at least one way, if th. strings are all of bounded length. — i.e. we just count th. freq. of occurance of all possl ~~strings~~ strings

Now, it would seem that ① this method could be applied to stoch. gramms. in which arby long (tho finite) strings could be produced — tho good statistics on some of th. longer ~~str~~ (and presumably low probty) strings, would, of neccessity

.25 ~~take~~ ouger to obtain. ② Any sophistication in grammar approximation should give a better approximate stoch gramm. sooner (i.e. smaller sampl.). There are probably restriction on "Any" — so, I will have to find out just what gramm approxn. types are possl.

.35 An example of an approxt. stoch gramm. that works "well" for small samps. (in this sense of better than this simple method of counting freqs. of s's. 363.20 — .25) and poorly for large! Each symbol is assumed to have a basic ~~freq.~~ probty and ~~th probty of th~~ (th. sentence termination being such a symbol) and th. probty of ~~any~~ s. is approxtd. by th. product of th. basic probys of its component symbols. Called th. naive stoch Grammar ≡ NSG.

.38 A somewhat better approxt. stoch gramm: One in which any string or sub-string can be given a defin., and to have a probty. We may or may not want to include constraints on what symbols can legally follow which.) Anyway, I suspect that such a gramm. starts out (i.e. small samp.) better than

365.01

Aug-3,61
↑ 3rd

01: 364.40. : ~~[illegible]~~ NSG (see 364.35 for defn. of NSG), and 1

is at least as good as NSG, since it can

santences and count them if they occure often

I suspect that most stoch gramms that can at least define

are ~~abe~~ at least as "good" as NSG in Th. sense I'm inte

.10 SN] In addition to being able to define substrings, I think I'd want to be able to define objects like (a, b) which could combine with objects like c or (c, d) in various ways, with various probys — e.g.

$(a, b) \overset{\leftarrow \text{1 method of comb.}}{X_1}\, c \equiv acb$; $(a, b)\, \overset{\leftarrow \text{another method of comb}}{X_2}\, c \equiv (ca, b)$

.17 $(a, b)\ \overset{\text{say}}{X_5}\ (c, d) \equiv ac, bd$; $X_8(a, b) \equiv ab$. etc.

In 364.38 – 365.17 we have Th. descs of some poss. stoch gramms, but what is essential in this discn: is ~~Th~~ both ① Th. form of Th. gramm and (2) Th. method by which Th. sample is processed to obtain Th. params of Th. approxn. gramm.

We can compare any stoch gramm. with NSG by comparing Th. relative coding compressions achieved by each, ~~[illegible]~~ for corpi of various sizes. NSG can be used to "code" by ~~defining~~ using a special symbol for every s that occures > 1 times. Th. ~~s~~ symbols of s's that have only occured once, are coded as themselves and result in a ~~simple~~ modified Laplace rule for them.

.35 Th. above discn. sugg. that Th. Q of 358.35–.38 has an answ. of "Yes" under certain conds. i.e. if Th. problem of constructing obs. and ops is solved ~~[illegible]~~ in a certain way and if at all Th.is answ. is "Yes" only to Th. extent that ~~[illegible]~~ Th. "impression" of 360.03–.05 is correct.

This discn may, conceivably, not apply to Th. ob. system of 360.11ff, since in that system (very probly) it is necy to devise an entirely difrnt Ob; ~~ex~~ each time. So Th. previous arguement about Th. NSG type gramm: ~~is~~ ~~not~~ being "adequate" for

366.01

[illegible]

.01 ; 365.40 : large enuf sampl. is not valid.

However, Th. ~~situation~~ situation ~~to~~ which Th. discn. is a possly impt. one, and I think Th. ~~study~~ reso is ~~of to be~~ of much import.

SN) In Th. ~~[illegible]~~ search methods being considered, we look for an op. that will operate on ~~Th.~~ a gn. input to produce a gn output. In Th. simple methods considered, Th. search does not in anyway make use of Th. knowledge of what Th. input and output are. Ops are just tried on Th. basis of ~~a~~ statistical study of what ops have been successful in Th. past. A Mtce Search under such conds. will tend to give a good soln., but will be slow. "Heurs" that do look at Th. input and required output, ~~will~~ can ~~be faster~~ give much faster success, but will tend to give ~~not such goodly~~ extrapolating solns — i.e. They will tend to be a bit ad-hoch. Th. degree ~~of~~ of ad-hochness will depend much upon — just what This heur. is.

I would like to be able to do without ~~a~~ heurs of This type. Whether or not This will be possl. may be ~~an~~ purely empirical Q, and vary considerably with Th. typ. seq.

I am working on a MTM $\equiv$ TMo — so heurs. of This type are possl. For a TM, hvr, heurs of "this type" are not ~~possl~~ possl. in any obvious way (tho clearly, a human ($\approx$ TM) can be shown completed examples and can learn from Them.).

.35 : 363.37 : While humans don't use 360.11ff for their "action op", They often do use it, at first, to solve new problems. After 360.11ff has obtained a soln., then Th. new obs and ops are further processed so as to integrate them with Th previous ~~[illegible]~~ system so as to reduce Th. total rcost further. This 367.01

ug , ing ,

ort -

.01: 366.40 "process of integration" may include th. ~~[illegible]~~ fusion of th. same op, etc

---

Well: To get back to th. prob ② of 369.02: Say we have a TM₀ running, and $[O_1, O_2, O_3 \ldots O_n]$ ~~are~~ is th. tempor seq. of <action ops>'s of TM₀. (O_{i+1} will usually include mo of O_i, and will have a few additions a/o changes). We then want to get a stoch gramma ~~lang.~~ to produce O_{n+1}, probabilistically. Now/ in this case, — what sort of stoch ~~langs~~ grams and "gramm fitting" methods will have th. remarks of 369.04 366.04 apply?

First of all, $O_1, O_2 \ldots$ are not a set of indep. /s's in th. ordinary sense — They are ordered and th. ordering is signif. [This problem was discussed at some length in Dart β 11 15 ff ~~[illegible]~~, but I don't think I knew enuf about coding at / that time]

For Simplicity, assume that th. O_i's are of th. obj, op type ~~[illegible]~~ outlined in 360.11 ff. I can try to get some idea of what ~~these~~ obs. and ops would look like ~~they~~ try to get an example of a heuristic (of th. kind that would give better proby values for th. O_{n+1} distribn), and see if we could expect to get th. effects of that heur using fairly simple obs, ops and "stoch gramm. fitter".

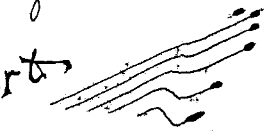
Consider $Ob_1, Ob_2, Ob_3 \ldots Ob_n$. We have th. constraint that Ob_i cannot be identical to Ob_j if $j \neq i$. — otherwise, just consider Ob_{n+1} to be th. next object to be predicted in th. seq. $Ob_1, Ob_2 \ldots\ldots Ob_n$. Th. non-repetition feature means that we can never treat th. Obi's as objects of which we can make freq. counts. ~~[illegible]~~.

A simple/ stoch lang. ~~for~~ approxn. would be one in which

368.01

f 24,61 TM γ,

Dart



·01 : 367.40 : Th. Obi 's were strings of computor ~~ubs~~ orders define ugms both compact and articulated ug[illegible]

Now — consider any expressable "regularity" in Th seq. O[illegible] will it (with large enuf samp.) be always expressable ~~as~~ by defining [illegible] more ugms, ~~[illegible]~~ and Their intersymbol constraints, and using symb[illegible] freq. counting? I think that all that Th. "adequate smp" means is that Th. ugm (i.e. Th. defn. of Th. "regularity") may be quite long (i.e. by r cost).

It may be that ~~s~~ regularities having to do with order in which Th. Obi's occure, cannot be expressed by ~~freq~~ statements about ugm freqs. and inter symbol constraints. — So at first consider regularities in an unordered set of (~~sets~~ strings of computer orders)'s.

~~*~~ Some strange regularities : Say Th. odd numbered (or prime numbered) digits ~~of~~ Th. strings tend to be A. How can we express This fact as an ugm? Well, for th "odd" case: For th Th. ~~[illegible]~~ articulated ugm, i = ntp (A, A, A, A, A). i.e. 5 A's, say. Then combine This with Th.

ntpst. $\left(\left(\begin{smallmatrix}A\\B\\C\end{smallmatrix}\right), \left(\begin{smallmatrix}A\\B\\C\end{smallmatrix}\right), \left(\begin{smallmatrix}A\\B\\C\end{smallmatrix}\right), \left(\begin{smallmatrix}A\\B\\C\end{smallmatrix}\right)\right)$ to form Th. ugmst $\left(\begin{smallmatrix}A\\B\\C\end{smallmatrix}\right) A \left(\begin{smallmatrix}A\\B\\C\end{smallmatrix}\right) A \left(\begin{smallmatrix}A\\B\\C\end{smallmatrix}\right) A \left(\begin{smallmatrix}A\\B\\C\end{smallmatrix}\right) A$.

Here, Th. alphabet of basic symbols is A, B, C.

↑— Here Th. entire method isn't entirely clear, but, anyway . . .

Another way is to allow freq. counts for each "digit position of Th. Obi's".

A fairly complex regularity .. If we let A be 1, B be 2 and C 3 Then if we take the product of Th. digits and theirs sum, these 2 nos will have a difference ~~[illegible]~~ ~~[illegible]~~ of 3 modulo 7.

~~*~~ i.e. say D_j is Th. integer assoc with Th. $j^{\underline{th}}$ digit ; Than $\sum D_j - \prod D_j = 3$ (mod 7).

.36 : 363.19 A fairly general kind of "regularity" : Say S_i is a "randomly chosen" string in an unordered Universe of strings:

369.01

ort-

.01: 368.40 ~~Thus~~ If **D** is a fixed string that dcrbs ...

$\vec{\ge}$ M $\mathbf{D}^\frown S_i$ = with probty **P** — where M is so

This method of dcrbg a "regularity" is more like a "...

This statement an only be gn. meaning if we make some sort on th. ~~[illegible]~~ (usually infinite) set of S_i's.

A more creative method: say ~~R~~ **R** is a "random no" ho... with a suitable density distribn. (e.g. th probty that it has just n digits is $\propto e^{-kn}$, where k is some const.). Then say D is th "Dcrn" of a stoch gramm (see 362.20 for th idea). Then M(D⌢R) is th. output of th. ~~[illegible]~~ creative stoch Gramm. (M is a Univ. Turing funct). ~~[illegible]~~ What we do, is ~~select~~ find some D ∋ M(D⌢R) creates th. known members of th. set [S_i] as often as possl. We mite be able to do this by taking various trial D's and "reenforcing" them when a random R gets one of th. S_i's. — Or, we can make a statistical study of D's and use a stoch gram with ugm defs to ~~[illegible]~~ characterize th. ones with successful trials

So, we create D's and using random R's, we form M(D⌢R) in ~~[illegible]~~ attempts to dcrb. th. set [S_i]. What we want, is a D that can be best used to dcrb. th. set [S_i]. We are using a Mt. Carlo method to find out (approxtly), how well each D dcrbs [S_i] — we must also include in "th. goodness of dcrn", the pcost of D.

Now, suppose that instead of using ~~a~~ approxn. stoch langs containing ugm. defs for th. D's, we just used ~~[illegible]~~ th. for 370.0, Ag ,6 TMJ,

Dart

01:369.40: Simpler (tho "adequate") stoch lang. that makes freq. successful D trials. (i.e. th. NS G of 364.35) How work for very large samples?

Well, just try all possl. D's in order of complexity, start with D = 0, 1, 01, 10, 11, 00, 000, etc. Test each D Mt. Carl.-wise by inserting "Random" R's into M(D⌢R). Actually, we need more than th. freq. with which M(D⌢R) produces members of [S_i] we need to know ~~[illegible]~~ approxly how freqly D produces each member of [S_i], and also know th. pcost of ~~th~~ each of th. S_i's when produced directly w.o th help of th.s D. Using these concepts, we could evaluate th. D's and eventually we would hit on ~~th~~ some very "good" D's — ≡ since we go thru them exhaustively.

We would like to be able to immediately say that use of a ~~st~~ stoch gramm for D's, involving defs of gpms, would certainly improve th. search rate. However, in th. above ¶ note that we did not assign probys to th. D's by means of th analysis — so th. D's did not form a stoch ~~gr~~ lang. in any simple sense. — So if we allowed defns. in our stoch D gramm, we wouldn't know how to assign "probys" to them.

But, even in this ¶, we do associate a proby with each D. This proby is th. pcost of derbg. th. set [S_i], using that D. ~~So suppose we were to try to construct~~ So ~~t~~ we assoc. with each D, th. pcost P(D).

Now — we ~~would like to~~ can try to devise a creative stoch. gramm. ~~that~~ will create D, with proby P(D). We can use ~~t~~ defs. in this gramm. hvr.

Thru a Mt. Carl method, we obtain for each D that

Dart.

·01 370.40 we've tried, a P(D) and its variance —. ...
to convert this variance into an ~~[illegible]~~ equivt. samp.
for th. associated D, and then, we pretend we have
large corpus ~~[illegible]~~ of D's ~~[illegible]~~ with ~~repetitious~~ copies
of each D ∝ in no. to th. assoc. samp. size. Using
this corpus, we simply devise a good set of npm defs. and
obtain their intersymbol constraints and freqs. — so we have an approxt. stoch
gram with defs. for th. corpus

We then use this approxt. gramm. to create new trial D's.
This gives us a greater sample density ~~[illegible]~~ near th. hy pts.
of th. "hill" we're climbing.

!! If I ever use th. above Mt. Carlo scheme in anything, it
would be well to do th. analysis in much detail, to give
quantitative estimates to th. various params. involved Then
devise a reenforcement scheme for npm probtys to approxt.
th. ideal method.

A brief review of th. idea: I have a bunch of unordered objects,
[s_i], and I want to make another one that is likely to be
accepted as one of them. I do this by devising ~~[illegible]~~ derns,
D, of stoch langs. ~~[illegible]~~ ∋ M(D⌒R) yeilds s's in th.
stoch lang. ~~[illegible]~~ Each stoch lang. is a partial dern. of th.
set [s_i], and ~~so~~ each D can be assigned a P(D)
∝ th. pcost of th. entire dern. of [s_i] using that D.

random no.

We then devise a (creative stoch gramm using defs & fugms)
to create trial D's. We use our best D's to
devise our next trials for members of [s_i].

~~Possl. direct proof & apptn of this~~ :

Th. above idea sounds very good, because with large amt of
sampl. it, i.e. th extrapolation, will become very good. Also any new heuristic

~j-40,

Dort .

.01: 371.40: devices that I want to add to improve Th npm. defs.> will make it even better — so I gain "adding-in" any improvement I can think of.

There are a few "loose ends" ① Th. business of co from something close to "G" assignment for D's, to proby as is it on a good theoretical footing ② my methods for constructing stoch grams with npm (and ritp) defs are not too goo ③ My methods of getting predictions or mt carlo output from such gramms are even poorer. ④ Th. idea of Inserting a random no. and D into a Turmac will have to be gone over and perhaps put into more practical form. ⑤ I don't yet have any ideas of what some good DS mite look like

Synthesis : What we've been doing, is using a stoch gramm. (of a npm defn type) to create Mt Carlowise stoch gramms of most genl. type. These secondary gramms were then used to create Mt carlowise th final now trial S. elements. Success or failure was used to reenforce the primary npm defn. gramm.

Now this cascade of 2 stages of stoch gramm. can be viewed as a particular method of using a reenforced single stoch gram. I.e. this cascading of 2 stoch gramms always yealds a stoch gramm. So, we have a single stoch gramm. that we can reenf. directly. This gram is in some sense, "universal" and has th capability of becoming very good with large sampl. of trng. seq.

Sat

6 6o

Dark

.01: 372.40: As I ~~was~~ have been using it th. was designed to extrapolate [s_i], an set of ~~bldg~~ strings. Perhaps I can extend system so that it can produce objects that w be accepted as members of th. ~~set~~ set [S_i] but not of th. set [V_j] (samples of [S_i] and [V_j] are both available). Th. idea here, is to use th. system for hill climbing, by ~~put~~ assigning to [S_i] known 'trial' pts. with G's ≠ above threshold and to [V_j] if below threshold. Th. threshold can be (th. median G obtained up to now) or th G of th. pt. of 10th (say highest ~~G~~ G thus far) (so [S_i] always has ≈ th. top 1/10 known members)

For a fairly smpl. H.C. prob. (i.e. TM₁), ~~[illegible]~~ have th. stoch gram with ugm defs. create (strings of symbols)s that are ~~[illegible]~~ trial ops. For these trial ops, one has external input, — then th. operator produces an output and gets its G awarded externally for that (I, O) pair. For each operator tried, we get 1 or more (I, O) pairs. From these and th. pcost of th. op., we obtain a mean and variance of th. G of that operator. Th. mean G (or some funct. of th G and its variance — with rewards for hy variance since this gives a possy of a truly (i.e. with long run) hy mean G.). of each op tried, is computed, and this ~~me~~ $\bar{G}$ is used to assign that op. to

374.01

SST Aug 20 61

Dart—

.01:373.40: th. [S_i] ~~class~~ set or th [V_i] set

There can be 2 stoch ngm defn. grammars. 1 gives th. probby an op. generated by that gram. will be in [S_i]; th. other

.04 for [V_j].

Some genl. simplifications and reconceptualizing of th. foregoing wo

1) Consider th. initial probby distribn. on th. set of random strings, R, as a stoch lang (This is certainly true)

Any dcrn, D, ~~[illegible]~~ defines a function, mapping strings into strings, and is mapping th. strings R into new string, M(D⌐R), and so th. ~~stoch~~ probby distribn. on th R's maps into a new probby distribn on ⊃, and th. stoch lang. of R, becomes a new stoch. lang.

This is a general conclusion that any function (e.g. D) on as set (e.g. R), maps ~~any~~ density function on that set (e.g. th. stoch lang on R) into a new density funct. (i.e. th. new stoch lang).

.30 2) A stoch lang. can be used as a dcrn. of ~~any~~ certain sets of strings (finite or infinite), if each member of that ~~set~~ is assigned ≠ probby by th. stoch. lang. Th. extra p.cost of dcrbg. th. set [S_i] by ~~means~~ of th. stoch lang, $P(s_j)$ ≡ probby of string s_j is $\sum_{s_i \text{ are all membs. of } [S_i]} P(s_i)$. somethings

This was analysed in a th. "recent" ~~[illegible]~~ (post (recent WJCC)) section of "Code" ≡ [Cod 288.01–292.22] ← gives idea as to what eqs are

So th total p.cost of ~~the~~ dcrn of [S_i] that employs th. D ≡ th. dcrn of th. stoch lang M(D⌐R) ~~[illegible]~~ is:

(p.cost of D)(~~[illegible]~~ total probby of [s_i] gn by stoch lang, D)(punctuation)

Don't ?

·01:374.40. 3) As th result of #2), it is [illegible]
is not, directly, a dern. of [S_i], it is clear tha[illegible] fo[illegible]
D's, one can, uniquely, derive a ~~dcrn~~ dern. of [S_i] a[illegible]
th. pcost of that dern. Furthermore, many of t
processes involved can be approximated by M.C [illegible]
Because ~~most~~ many D's do give derns of [S_i], on can
D's at random and use th. resultant pcosts of [S_i] dern
as "G" values to hill climb on, to find t "best" D.

Also, th. D's found (in this way), are very good for extrapolation of [S_i].

Contrast this random search for D's (where most D's fit at least a little bit) with th. choice of random derns of [S_i], in which hardly any derns fit at all ("at all" means = perfectly) The search time is lowered considerably, but, if one uses not random but ordered search (in strict order of pcost) for th. ~~exact~~ direct dern., th direct ~~exact~~ derns would usually be better when found.

4) Th. method used above for search, is for extraps ~~unorder~~ extrapolation of unordered sets, [S_i]. How can this be generalized to ~~ath~~ extraps. of other types ~~of~~ things?

This method of coding a set of objects by first defining a stoch lang (e.g 374.30) is a genlzn. of a coding idea of ca 288.61 – 292.22. In that case we had an a priori pd on th. set of derns of all objects $P(O_i)$. We want to derb. th. subset [Q_i] of objects In a simple way, its pcost would be $\left[\sum_{O_i \in [Q_i]} P(O_i)\right]$. Hvr, suppose we have a way of dcrbg. a subset D of objects that includes [Q_i]. Then we can view this subset as defining a new stoch lang ($\equiv P'(O_i)$) (≡ probty distribn)

Part

.01. 375.40 over th. set of a objects, This. new distrbn

is, D. h l rn. of , ow-r ?, po

sin

$$\left[\sum_{O_i\in[Q_i]} f'(O_i)\right] = \left[\sum_{O_i\in[Q_i]} f(O_i)\right]$$ since f(

a r normalizn constant with resp. to $P(O_i)$.

SN) If we have stoch langs. $P_1(O_i)$ and $f_2(O_i$

that be being used in attempts to derb. ~~the~~ a certain

set, then th. (renormalized) ~~E~~ stoch lago $f_1(O_i)\,f_2(O_i)$

is also a good try - as would be renorm) $P_1(O_i)+P_2(O_i)$.

d rns. of these stoch langs. are of ourse ≈ twice that of either

$f_1(O_i)$ or $f_2(O_i)$

Th. means proposed for extrapolating th. known sub at [illegible]

seems v. I would like to think of an example to

try to apply a "study problem".

Well, consider th. more compl "H.C." prob. of 373.01-374.04,

with th. "Good" set $[S_i]$ and th. "Bad" set $[V_j]$.

Say th. S_i's and V_j's are computer pgms for "searching" for solns. to "well defined ~~problem~~ ~~tasks~~ problems," and th. G is mean search

speed. Or they might be for "searching well defined hills" in which case the G is a combination of search time and hi'l ht.

~~[illegible]~~ obtained.

O.K. - perhaps - for a very signif. diff prob. but now lets go back to th. problems

of a Tng.seq. MTM 357.04 and .20; and also th. prob. of example to see if good heurs will automatically develope 359.30ff (= 358.20ff). We

want to consider these and this all together, and see if we can

come up with a good idea for a computer pgm with ~~E~~ th. desired features -

i.e. most of interesting info is fed to TM via Tng. seq., and

M improve rel methods automatically with ~~a~~ large tng. seq.

art →

.01: 379.40: ~~[illegible]~~ Very essentially, what we have coding method, is a h.c. method ~~that~~ to [illegible] anything — also, perhaps, a coding method that ↑ in [illegible] the no. of probs. worked ↑.

In the original prob. of 357.04 and .20, I wanted the trng. seq. that would lead to the ability to work a dif't prob. involving "creativity". ~~The~~ TM and Trng. seq. involved could be of type TM_0 (≡ MTM), $TM_{\frac{1}{2}}$ (≡ NMTM) or TM_1 (H.C. TM).

More exactly, for TM_1, ~~[illegible]~~ $[I_i, \overset{\text{output}}{M_i(I_i)}, G_i]_{i=1/n}$ the i th operator tried; external awarded G — usually includes resp time; is a set of known Input, Output, G triplets. From this, we want to create a M_{n+1} (operator) that will tend to xfm I_{n+1} into a ~~output~~ of hy G.

The idea ~~we nown~~ of 373.01–379.04 relevant to this is something like: say D_j is a d'n of a stoch. lang that can produce M_i's from "random nos." by $M(D_j, R_i) = M_{ij}$. Then we treat the M_i's of this data as pts on a "noisey hill" of ht. G_i at pt. M_i). We then want to find D_j's that produce M_i's of hy G. We divide the M_i set into 2 subsets — one of hy G, the other of low G. We then try to find a D_j that ~~[illegible]~~ "fits" best the hy G set (or perhaps just the top pts of that set).

.35 We would then try to find a ~~B~~ stoch ~~lang.~~ gram L_1 using ngm defs to produce D_j's that in turn tend to produce M_i's of hy G. Also, we would make a stoch ~~lang~~ gramm L_2 using ngm defs to produce D_j's that produce M_i's of low G. ~~again~~ We then will produce trial

381.01

Dart

.01: 380.40 M_i's that use th. hy freq. ngms of L_1 and avera

ngms. of L_2

.03 An idea sugg. by th. above: We have th.s

of (I_i, O_i, G_i)'s. With D_j we produce a M_k. We test M_k by giving it randomly chosen members of th. known input set $[I_i]$. We do this untill a few outputs of known G are produced. From th. length of th. M_k's dcrn and th. sample (I_i, O_i)'s of known G, an expected ~~[illegible]~~ $\bar{G}$ and σ_G^2 are obtained. Several M_k's are produced by th. D_j and evaluated this way. We also try other D_j's {produced by a ugm dcf. Gram} and use them to produce M_k's, which we evaluate. At any time, if we get a new ~~[illegible]~~ input and need a response, we use th. M_j of hyest exp $\bar{G}$, and use th. resultant external G obtained, to modify that $\bar{G}$ and σ_G^2.

Th. G fn. to a D_j is hyer, if ~~[illegible]~~ th. M_i's produced ~~[illegible]~~ result in a fast search for known (I,O) pairs of hy G. What we do, is try first th I_i ~~of hyest~~ with a known O_i of ~~hyest~~ hyest G. Then we try th lowest O_i, then th. ~~2nd~~ hyest, etc. Th. idea is to terminate th. search as rapidly as possl. by getting a few very hy or very low known I,O pairs for th. operator.

It may be necy to use a more randomized input search. Th. ~~over~~ ~~[illegible]~~ steering of th. above R may result in some serious errors/ i.e. sound fairly bad (I,O) pairs — but check on this.

.01: 381.40: → "There is an obvious refinement — & very impt. one — i.e. th devising of "rosen. th M_i don't have to produce ~~exact~~ responses exact as known responses. A more detailed analysis of th. re (i.e. its quantification) will suggest other refinements"

I drew up a 2 dim. analog of th.s TM, problem that is very good for "fixing ideas" 339.30 ~~which may be~~ in ~~[illegible]~~ Use this model for ~~further~~ discn. of TM, .

So, th. way this thing works, is that at any particular time, it has an M_i available which, to its knowledge up to that time, is "best". Meanwhile, between a/o during th. soln. of externally asked problems, it ~~works~~ searches around for M_i's of higher $\bar{G}$ a/o ~~better~~ better σ_G^2. It also searches for a better L_i (380.35). During its searches, it presents an output of th. newly $\bar{G}$ and σ_G^2 whenever it finds a "signifly better" M_i that th. previous "best". This output tells th. human operator (≡ client) how good to expect th. next answ to be. Th. client can tell TM to "meditate" until it gets a M_i of at least a certain goodness .. then work th. next problem.

.30 In additn. to this → nec'y. refinement, it would also be well to have TM able to ~~[illegible]~~ discover th. relationship betw. (I, O) and G, and ∴ be able to conceive and suggest new M_i's of greater expected G ~~than~~ ever befor. These → 2 refinements, if realized adequately, ~~we~~ could make th. resultant-TM very powerful,

.37

Altho th. idea of 381.03 — 382.37 (~~in~~cluding th. 2 "refinements") mite be O.k., it doesn't seem too close to what I feel is my own intuitive method of working probs. This diffy

383.01

Dart -

.01: 382.40' may be regarded as a penalty of want (
automatically better themselves as more and more
worked. Actually, the TM of 381.03 - 382.37, can
be so bad — even w.o. the 2 "refinements" — Essentia
TM directs itself to a ~~[illegible]~~ problem that is not really the who
problem — so the 2 "refinements" have to be tacked on
ad-hock, rather than appear naturally by themselves.

I think I could work out a TM working on the same genl. principals that was directed at a problem closer to what I really want, but the present system would be probly V.G. ~~[illegible]~~ for a "study problem"

Note that this system, while ostensibly designed as a TM_1, could be used as a TM_0 or $TM_{\frac{1}{2}}$ in the follg. way: Say the G for a trial M_i^k was the fraction of problems in the entire trg. seq. up to that pt., that ~~were~~ it worked correctly. We could reduce the search time by a great deal by inserting special devices that took advantage of the fact that we have an MTM trg. seq. This means that the "closeness criterion" for correct answers is ~~very~~ trivially 2 valued. Also, the maximum conceivable G value is 1 (≡ 100% correctness). We can use the sequencialness of the trg. seq. in a useful way.

Actually, this system, with the 2 ad-hock refinements, really isn't very distant from the ideal, non-ol. soln.. (Use the ~~[illegible]~~ 2 dim. model of 339.30 for thinking about this). We simply ~~devise a data~~ use this system, then, separately, we try to devise ① a closeness criterion and ② a functional relation betw (I, O), and G: Actually, a complete soln. of ② 384.01

[illegible]

.01: 383.40: makes ① superfluous, but a partial get a partial or complete soln. of ②

② means that for any trial M_{2i}, we can get th. G of I,O pair we try. Th. "closeness" criterion is a way to approximate th. function $G(I,O)$ — i.e. we [illegible] a bunch of $G(I,O)$ pts that are known, and any new (I_j, O_j) pt. ~~that~~ is not identical to th. known ones has its G evaluated by (a) how close th. new pt. is to th. old known pts. (b) th. mean and variance of th. G of new pts

Th. moral is. th. system of 38?.03–382.37 is useful both as ① a study problem since it can be used for TM_1 and TM_2 ② as an ~~approach to~~ approximate TM_1, (3) " an approximate TM_1 of less er. character when th. 2 refinements of 382.01 and .30 are considered. Which is a very serious approach to a very non-al. TM_1.

There are several impt. problems:

1) With L_1 and L_2 (see 380.35 for defs) one has th. problem of creating s's that have a max ratio of proby to being in L_1 rather than L_2. This could define L_3. i.e. if $P_1(s)$ is th. proby assigned to s by L_1, and sly for $P_2(s)$ and P_3 s, then $P_3(s) = K \frac{P_1(s)}{P_2(s)}$, where K is a normlzin const. (if $\sum P_3(s_i)$ converges). Th. Q, then, is — how to make a constructive gramm for an approxn to L_3.

Ideally, it would be well if we could reenforce L_3's gramm positively or negatively directly whenever it produced a hi or low $\bar{G}$ M_i, resp.

We had some objections to this system — i.e. we wanted 385.01

Start ⇒

.01 : 384.40 : Something like [illegible] L_3 to look at th [illegible] M_i's and G's — and from all this info, [illegible] next trial M_i ~~[illegible]~~ It was somehow t[illegible] present system doesn't do this. But, it does. Th[illegible] th. past data and trials is summarized to some extent, present form of L_3. Th. better L_3 is, th. better is th. summary. Actually, L_3 doesn't exactly summarize this data. This data is, hvr., available directly to L_3 (since it is stored). L_3 is an attempt at using this data to obtain good "next trial's"'s for M_i.

Lets try to get more specific about the system:

Say we start with an TM$_0$, and our trg. seq. has 100 examples — each essentially difrnt, so that a new M_i ~~was~~ almost always has to be created to cope with th. "next problem". Take th. last 10 successful M_i's (These 100 M_i's have been constructed "by hand" by me.). Pass a stoch Gramm F_3 thru them. F_3 is th. gramm. of L_3 [illegible]

It is rather impt. that L_3 be able to be th. most genl. kind of stoch gramm possl. About th. most complex stoch. gramm that I'm familiar with at pt, is th. stoch. context dep. PSG. Th stoch non-dtm PSG mite be better

Probly th. best thing to do would be to look at th. set of 10 "good" M_i's, — then find some kinds of stoch grams that could have created them — then try to genlz. these types of gramms to a more "universal" form.

Of course, one form of univ. stoch gram. is th. one that maps th. set of "all random nos" (≡ a density distribn. on th. set of all strings)

= 86.01

Part

.01: 385.40 into th. set of all strings, by use of maps R into th. new strings."

Considering L$_3$ and th. created M_i's as constituting stoch operator for responding to th. I_i's: We would like to able to consider such a simple stoch op. and be able to reenf directly. Hvr. this does not appear to be th. most efficient way to work. When we get an input I_i, we would like to select th. "best M_j thus far" to ~~realize~~ impliment th. response. If we just use this simple reenfd. op., then L$_3$ would create a new M_i which was untested and this new M_i would be used to obtain th. response. L$_3$ then could legitly be reenfd. on th. basis of this response. While L$_3$ would improve as th. trng. seq. continued, th. mean G of th. responses that it gave rise to, would not be v.g.. It would, e.g. not need a "closeness criterion" or an internal model of G as a function of (I, O). Essentially what this model does, is assume that th. (I, O) trials are very inexpensive, so it would rather make trials than figure out approxtly how good each M_i^I is be for using it to create a response.

Actually, I should be able to take th. rôle of a L$_3$!; By looking at th. last 10 successful M_i's, I should be able to see just what kind of gramm could have created them, and without much extra trouble (if any trouble at all), could create successful trials for th. next problem in th. trng. seq.

After constructing 1 (or 2) L$_3$'s, I should be able to devise a Gramm. to construct a whole stoch. lang of them.

So what apparently has to be done now: Construct a trng. seq. and a set of M_i's to solve it. Then devise some L$_3$'s to

+ er

386.40. Create th. M_is.

Before doing this, think a bit more about it.

clearly what I want.

Th. system certainly <u>does</u> appear to have many des

features. e.g.

.06 1) One can use it for TM_0, $TM_{\frac{1}{2}}$ or TM_1.

2) One can add th. 2 refinements (of "closeness" a/o G as a fu

(I,0)) and get a fairly non-rel. TM. (See 339.30 for a 2 dim. po of th. non-rel system.

3) If L_3 is a "universal" lang. then search speed <u>will</u> ↑ as th. trg seq. continues.

4) Th. system <u>can</u> give quick responses by using "th. best M_i to date".

5) Since it can be a TM, (recent. TM ≡ RTM) it can be taught things as an animal is taught, and can be made to work <u>any</u> kinds of probs — including understandin human speech and ~~a~~ being asked to improve itself.

Some probs. that must be solved:

1) Get a simple trg seq. and set of M_i's to solve it.

2) Devise a <u>form</u> for L_3 — i.e. a good type or set of types of univ. stoch. prgm.

3) Figure out a way to get L_3 to create M_i's that have a max. prob'ty of ~~being~~ being in L_1 rather than L_2

4) What is a good way to modify L_3 as a direct result of the success or failure of ~~th~~ outcome of its M_i?

.35 5) Sysm does <u>not</u> try a guess what th. next prob m will be and pattern its trials on his guess. This amounts to observing a sequential (temporal) pattern in th. successful trials. See Plan 396.35 for possl. soln

I think 1) and 2) would be good to work on first. Write some typical M_i's, and some possl. L_3's that could have created them. Then, express in intuitive form, various regularities in this (or any hypothetical) set of M_i's ~~so~~ try to devise a form of L_3 ∋ these regularities are "easy to express."

.01: ~~387.40~~? ~~Say~~ M_i are 1 dim. strips of ...

One possl. form of ~~[illegible]~~ G_3 is to be sub-strips of orders, ntps of orders and asso... various methods of combining them.

For a goal to work foward — consider th. GPS. of ... Here we have a kind of problemsolving routine that works f... variety of problems. Can we make a trg. seq. for which a hum... would discover their heurs. of "closeness", "improvement", etc. ~~[illegible]~~ Then a G_3 must be designed to make such learning not unlikely.

It may be that th. Sim-New. heurs are softly complex, so that a TM that could learn them with a reasonable trg. seq. has a hy likelyhood of being able to learn anything thru a suitable trg. seq. Their heurs have th. advantage of being close ~~to~~ human, so they tend to be easier to work with than most.

.20 Q. that looks like I'm ~~[illegible]~~ discarding much impt. recent work:

If I use M_i's of this → form, and make L_3 consist of a simple s, that creates M_i's according to th. simply observed ugm (and ntp) statistics — (these statistics will change with time, and new ugms and ntps will be defined) — could such a TM automatically increase its search speed as was desired and which desire resulted in th. somewhat more complex system.

Well, there is this possy, that we can use simple statistics on th. successful M_i's ~~th~~ (e.g. like ZTβ 141) to make th. new trial M_i's.

Another thing is to consider "Lisp" as an M_i form ~~[illegible]~~ a/o a G_3 form.

This → is pretty much what I had in mind quite some time ago (i.e. Flang β1160.20ff) > 1yr. ago!

Part - 1

.01:388.40: A basic related Q: Suppose we ha

hy G set: Suppose ~~[illegible]~~ G_3 has defined

ntps to code this set of M_i's. Now, suppose we

"regularity" in these M_i's Does there exist a set of

defs. ∋ this "regularity" corresps to a certain ygm c/o ntp

If so, is there any reasonable way such a regularity could have been derrd? — i.e. ~~[illegible]~~ would what would be normally regarded as a reasonable trng seq for that regularity, for a human, be ~~an~~ adequate trng seq for th. simple ygm, ntp learning mechanism?

~~A~~ possl. flexability: We can take this set of hy G M_i's and try "parsing" it various ways There will be difrnt sets of defs of ygm's/ntps that give fairly good coding, — yet give essentially difrnt. extrapolations. We may want to use this flexability to ~~be~~ ~~try~~ be able to try new methods of recognizing regularitis in th. corpus. Also, there may be difrnt strings of orders that give th. identically same M_i's — but certain ~~[illegible]~~ representations make certain regularities more apparent to th. ygm/ntp defn. stock. ~~[illegible]~~ Gramm. — This is a sort of more generalized form of "reparsing" or "~~[illegible]~~ alternate sets of defs"

What I would like would be some way in which all types of regularities could eventually be recognized in th. M_i set — when th. M_i set become large enouf.

Also, I would like a "creative Grammar" to create th. trial M_i's and characterize th. hy G set of M_i's.

.35 Say we have a bunch of ~~[illegible]~~ hy G M_i's. We scan them for regularities — out of a certain set of regularities ~~#~~ Th regularities that occure get defined with hy post and are combined to make new trial regularities to look for, etc.

Note, hvr, we would like these "regularities" to be

390.01

D218

.01: 389.40: in a form to contribute to th. str

It would seem that ~~creative~~ regularities in [illegible] should be also subject to th. same comb. rules for h, as, ~~as a~~ regularities in a directly descriptive gramm.

Note, hvr, that often, ~~the only way~~ an entirely adequate combination rule for abss, is concat. This can be true in create descriptive gramms. In such a case, I think that defining [illegible] need be th. only type of regularity to be recognized!

I think this last ¶ should be ~~explicated or~~ clarified with some examples from either descriptive or creative "properties".

To go back a bit (389.35), we want to be able to ~~derive~~ discover all possi. regularities in this set of hy or M_i's. Essentially, we have a ~~TM~~ hyper order TM to do this, and we want that hyper order TM to be "universal" and ~~a~~ creative.

Perhaps this idea of combining old regularities by concatn. to form new ones can be easily studied in (1) ordinary computer-order sequs. (2) LISP operators.

Some simple "descriptive" regularities. e.g in a sequencial corpus: that th. symbol "a" occurs with "unusual freq". can be symbolized by writing "a". That ~~a~~ "bc" occures with unusual freq. can be symbolized by writing "bc". Note that we will look for th. regularity "bc" if b and c are of hy pcost.

.35 A more complex type of regularity ~~as~~ ~~should~~ for, say, a linear ~~set~~ corpus, should be dcrbd. intuitively — Then have it expressed as a combination of -several simpler regularities a/o combination ~~of~~ symbols.

i.e. say th. odd / ~~noted~~ symbols in th. corpus tend to be "a" with signify difrnt statistics than th. even noted symbols.

Dart

.01 . 390.40 : This idea can be "factored" into th. rel. freq of th. symbol a.

① Th. idea of th. odd symbols as a subset of [illegible] concept. ② Also th. idea of th. ~~[illegible]~~ rel. freq. of a in any subset of symbols is impt. So we can do our original idea by ~~[illegible]~~F((odd symbols), ($a_{freq.}$)).

(odd symbols) is an argument of F. ~~An~~ An alternate first argt. of F would be any subset of symbols.

~~[illegible]~~ We may want to use Polish notation, or something like it, here.

Odd symbols: We have th. notion here of numbering th. symbols to ~~[illegible]~~ designate them. ~~[illegible]~~
The concept of a subset of these nos designating a subset of symbols. — th. concept of th. odd nos. as a subset of th. concept of sub-set of integers nos. — Th. concept of "odd nos." — th. concept of integers.

We have listed here, a set of concepts that would have to get hy pcost (and have a suitable trng seq. ~~[illegible]~~ to make their being defined likely) befor th. idea of ~~[illegible]~~ counting th. freq. of th. symbol "a" over th. odd symbols would be
.35 a reasonable thing to try.

Note that there are other seqns of abss. that could be defined to make this concept have hyh enuf a prip.

Note that th. goal is to find all possl. regularities in a ~~sample~~ set of M_i's — possibly w.o. being too economical about sample size.
One idea was to rewrite th. corpus, making use of one regularity to compress coding — then scan this compressed code for regularities.
If we have arranged notation properly, it may well be that
392.01

a 18

.01 : 39/40: the only regularities we ~~need~~ look for like ~~[illegible]~~ ngms &, or ntps. I am sure of ZTB 141, which is a descriptive gram., but [illegible] analysed to obtain th. gramms of th. associated gr [illegible]

According to this ~~[illegible]~~ expected routine, any gramm type ho[illegible] e.g. FSG strong or weak) or xfml Grim! (using, hvr, excess large samp. sz.). Even so, this would be great for MT lear[n]ing which we do have a very large sample, readily available. Th. goal, hv of eventually obtaining all regularities (tho lagging behind with samp sz.) seems a bit too ~~[illegible]~~ ambitious, and perhaps imposs/.

IN 2?1 B 4? I think that rewriting th. corpus in terms of one ngm, may conceal th. ~~existence~~ unusual freq of another ngm.

SN Incidently, it seems very likely, now that using a nbPSL like ZTB 141 on a large corpus, would result in a bunch of ngm defs, from which one could "easily" obtain ngmsts (by factoring th. set of ~~defns~~ ~~[illegible]~~ like "Th. mech. of ling. learn"), and then loop rules.

Th. reasoning is this: say we have a stoch PSG. then it will produce (substrings that ~~a~~ is a pos >'s with "unusual" freq. ~~[illegible]~~ and this "unusual freq" will be recognized if th. samp.sz. is large enuf.

E.G. consider th. lang. $a^n b^n$. Ngms: aa, bb, and ab all have unusual freqs. In th. case of this lang hvr, I don't know just now I would proceed with th. discovery of th. gramm, after ~~[illegible]~~ ngms ~~were~~ & vrd.

There is some point in not having "too good" heurs for L3! Th. idea is this: Suppose one has a straight MTC. search for an object and one takes th. first object that "fits". An "heur" is a reordering of th. trials by modification of th. a priori. As such, 01:392.40 : it may lead more quickly to th choosing of a obj... choice will then not have as much likelyhood of aprip. (i.e. it will not usually be as "good" as for ...") Note that th. heur. modifies th. Mt. C. search as tho th. aprips were diffrnt.. This apparent a... alters th. search ordering. Th. actual aprips remain th. same. This discn, hvr, must be worked out in a b... detail to see just ~~when~~ it applies. Some (if not all) heur actually do alter th. aprips, since these heurs are based on info from previous searches related but not identical to th. one in question.

8

18 In th. vein of 388:20 : I think that if th. TM actually developes from a trng. seq. using Mt. C. methods, that th. ngms that would appear would actually tend to be very useful in producing new trials by simple concatn. of hi prob ngms.

Now suppose that it turned out that th. odd symbols of th M_i's had signify difrnt statistics than th. even symbols. There is no way I can think of to "notice" this regularity if one is only allowed ngm defs. Perhaps th inter-symbol cnstrts would help a bit. In fact, intersymbol constraints could do it completely! — e.g. say we have 3 basic symbols (a, b, c). Then th freqs in th. odd positions are O_a, O_b O_c; in th even positions E_a, E_b, E_c. Then we devise a set of 6 symbols: $\alpha = aa$; $\beta = ab$, etc, and we express th. entire corpus string with these new 6 symbols. This completely takes care of th difference in frequs in even and odd positions.

Well — this still isn't what I want. Suppose th. prime ordered symbols had difrnt statistics than th others — e.g. suppose th. prime symbols were always "a".