Plans :

867 - 8493

10.8.60

Plans

867

.01: 1160.40: successful objects (with a freq. of their use freq times and combining them. [Concat. need be th. only comb. r
of instructions is "adequate"],

Now th. big trouble (I think) with this, was that if I a bunch of saq. of problems, and I got some new problem type, my trial mutations of th. old, successful ppm, I would h th. whole tup. saq. thru!

This can be avoided in 2 ways (this "devry" is sourc (at least temporary) joy) ① At first, use a "non-stationa i.e. : suppose just after th. rth problem, our lat operator begins to fail. Then we devise a new operator that comes into play after t = r only, that is to be used problems only. Th. costs of coding can, however, be costs are obtained by averaging over th. past code c hypcost previously successful op). → ② After one or more codes fo "new op" have been made, one tries to incorporate one of them into th of the op., w.o. using th. "absolute time" marker. is successfully done, of course, only when th. total of th. entire op. has been reduced. ③ V.G.

Select a set of "typical problems" from th. past corpus use these to test new trial ops, instead of running th th. entire tup. saq. These "Typical problems" can in which order they are to be tried ordered, so that th. "expected real time to failu for new trial ops. is minimized. (Note: these "typical probs" will not We can use these Set of probs problems to either devise a modification successful of th. old op. more directly, or to use this for tri on th. "incorporation phase" i.e. ②

I think both of these ideas are very much used in humans.

→ A standard way of doing ②: Devise a method for recognizing when problems of

10.39

10-5-60

Plans 868

.01 ! 67.40 : "th. new type" occurs. i.e. in what whose way [illegible] most previous probs. Th. "recognizer", hwr, must reject less [illegible] "time marker", or else one gains nothing. Hwr., a "time marker" may [illegible]

Cost of th. "time marker": This marker can just [illegible] symbol that indicates that th. T.S. after that point will have [illegible] treated difntly. If one uses r such markers in [illegible] code length K, they will each pcost $\frac{r}{K}$. This corresps. [illegible] coding of a truncation of a T.S. — and I suspect that th. [illegible] might often be better to use th. pcost $(\pm+\pi)/n\pi$ [illegible] truncation cost.

↳ results that are 100% accurate anyway — say only 80% [illegible] th. new problems are amenable to th. "new method"; [illegible] and 20% are amenable to th. old method. In [illegible] a case, a time marker, dividing th. T.S. into 2 sub-T.S. [illegible] of difnt. statistical properties [illegible] can [illegible] good. Later, if one could devr. a method to recognize [illegible] th. 20% with ~~number~~ hy error degree of accuracy, one could [illegible] dispense with th. time marker — tho even with th. time marker, such a recognizer would be v. g. In fact, if I had a recog. scheme it would have to be very good percentage-wise, for me to be able to trust it on th. very large previous corpus, because it would make a numerically large no. of errors (tho' percentage-wise, small). Th. time marker would certainly aid considerably — unless th. recog. scheme was very close to 100% accurate — (as it might be a MTM). [One can view th. time marker as one type of "recognizer"! Th. Problem of how to combine partitions of a set, to obtain optimum discrimination, is discussed at great length in IR ≈ 8180 ff]

Have TM able to work probs. many difnt ways, and store all of th. hy pcost ways. Th. exact mechanics of th. ≤ pcost evalu. is [illegible] immed. clear to me, but certainly we would want to store alternate methods, so we could use them, or [illegible] combs. of them

6901

10.8.60

Plans

γ69

.01 '.6840: for new trial methods.

Th. "Char. Methods" ideas of ~~[illegible]~~ EPM 11.10.01 ~~[illegible]~~ be a rather reasonable formalism to put this machine in

.05 Another idea is to try to use direct machine ~~[illegible]~~ because of their speed

.06 For these problems, and the sorts. — it may be possi

Newell and Si... data — learning

GENERAL PLANS: (10.8.60):

1) Do the simple Pgm. the simple nb PSG of Dart 46

2) Do some IR pgmg (around Dart 64). If pra... any results at all are obtained, this will be a reason... "pot boiler". Th. "Medical diag." prob. may be easi... to get data for; since it has been done on a com... but w.o. "Learning". Try to get names of people who did it, so I can get their data.

A/O Try to get SM data from U of Chi peop... to use for project! Kill 2 birds with 1 stone!

3) Work on "Human aided" TM of Plans B1160.20 – γ69.05. Th. will be getting us into the home stretch!

4) Try to work this out on a very "hi level" before getting to details.

(4.26.61): Th. pgm dcrbd in ZTB 141 (and Dart46) isn't ver... Its main values are outlined in Code 262.30 ff. My present idea is that th. IBM 709 pgm is almost ready for debuging, and I have had it almost key-punched, I shouldn't finish it now, but go on to (3) or some preparation for (3). Th. main possl. justifications for finishing th. present 709 pgm (a) Maka AF feel I'm doing something material (b) see

4.26.61

Plans 8 264

.01: 69.40 : if ~~the~~ the proofs of the ngm defs are about rite in ppmp. (d) I have gotten much of it key-punched ≈ $50 (e) It could be of some use - e.g. feeding a set of the ppm of Dart β1300, also some of the applics of Dart β mite be useful.

Rebutal: (a) ZTB 138 may make a big splash and keep happy. Also, I've written ZTB'S ~~138~~139, 140, 141. If they're not happy I can't waste too much time just trying to satisfy them, and needed time on serious researth. Also, I don't really need money. (b) Apparently (code 262.30) it would not do this much. (c) S true, — but I suspect that I'll not use fortran anyway, ~~but~~ my ~~fr~~ serious work, but something close (if not identical) (d) Do not throw good money after bad. Occasionally, hvr, it is for public relations (e) Hard to say — I may actually want use it to find sets of ngms to be used to make better "trys" in a Hill climbing T.M. - but untill I have the need — don't ppm it!

So, if I don't finish the Dart 46 ppm: what next? It would a bad idea to write some reviews, to familiarize myself what I've done, to prevent needless repetition. Also, a review written is always good to have.

4:27:61

TM8

Probly 8265

Plans

.01 : : Some impt. things that I want to consider ahead into a big final project.

1) Th. ideas of Plans . . . . β 1160.20— 869.06

2) Th. Observer, operator machine of EPM β1110.01 — 11 This machine looks like I could get it to do things Th. way human seem to, by getting a reasonable set of ops and obs, combination rules.

3) BSTM : I'm not sure as to Th. relationship of Th. Co Meth. of ind. inf. to this machine.

4) Th. problem of getting optimization of a T.M. w resp. to a goal, in a rather direct way. This -This co up in Th. SM & setting rule "problem."

5) I would like a T.M. like TM704, that uses basic machine in a very direct way.

4.28.61

TM's.

Phaus

01:265.40:

SN] On the BSTM and "integrated TM's". [illegible]
(a some discn. in the BSTM section on the poss. of not
(looped)
a hierarchy of TM's, but having a single TM with
recuft schedule, that gave F.B. (even as a funct. of responses
a long seq. of
the exact nature of the F.B. funct. was
In such a T.M., the division into Hierarchys is not nec
" " " " " also,
we can, presumably put the T
hill manually".

SN: Perhaps before working on this BSTM, work
an ordinary recuft T.M. See if a qual. solu. to
problem is, almost automatically a solu. to BSTM
providing one uses the rite "sequacial" recuft
schedule. There may be much in BSTM o

SN] For an ordinary recuft T.M. ① Perhaps dr
a model of a recuft net T.M. (I may have a
done this) ② Then see if I can modify the g
principals, so as to make a T.M. built around
properties of a stored prog. computer (e.g. T
I think working on ① and ② simultly wou
very helpful)

See various discns. of Hill-climbing TM's

4·29·61

Plan

.01: 266.40 Some of BSTM (J5 in particular), was [illegible]
idea that if one had a $TM_1 \neq TM_2$ situation, say,
TM_1 wouldn't, perhaps, have to be very "brite" to "get
A kind of TM_2 that might be an adequate st
TM_1, say, would be represented by a large, binary n
(This could be true if TM_1 is a neural net, or a pgm.)
Reinforcement is either +1 or -1. (Good or Bad). Various
are tried (= Binary nos.) with various inputs. Say
is th. Binary no. that represents th. ith trial TM_1.
Say I_j is the jth [illegible] input.

$G(I_j, T_i)$ is th. reinf. (+1 or -1) of th. output
I_j is fed in to th TM_1 assoc. with T_i.

TM_2 tries to extrapolate to get a $T_K \rightarrow / \sum_j G$ [something]
will be max.

A simple way to do this: TM_2 uses "correlations / codi
(as in IR section) to obtain impt. ngms within th.
so that TM_2 can predict $G(I_j, T_K)$ [illegible] as well a
possl. without knowing I_j. TM_2 then tries T_K's for which
this prediction is G = +1 & with probty as close as possl. to 1.
Th. method of discovery of good ngms, that is discussed in IR
would be used.

If th. T_ℓ are pgms, then we have much apri info on
what ngms are "good" — e.g. an ngm will tend to remain
if is translated along th. vector T_ℓ. Also, ideas about what ngms
constitute subroutines, etc., are impt. This kind of info should be built
into TM_2's extrap. methods. There is a lot of discn. of
this sort of thing in early "Dart" section (& Dart β 1130)

268.0

2·29·61

Plan

.01: 267.40: So, using these T_q's as TM_1's, I could de[sign] TM_2. This $TM_1 - TM_2$ comb., will then be a hill [-climbing] TM_1 will have all possl. orders available and will ∴ be "univ[ersal]" ∴ I think, with a suitable T_{hq}, seq. and reinf. metho[d] be able to get TM_1 to be a $TM_1 \equiv TM_2$ combination. Th. r[ein]f. method would give reinf. for working long sets of probs. correctly and fast.

At the present time, I have some idea as to what TM_1 would like and also what TM_2 (correlational coding) would look like. I like to consider the pair, as a unit, operating together, as a TM_α and have TM_α be universal, with some sort of simple hyper-TM (perhaps just myself), to improve TM_α. I don't know how to make TM_α universal (— unless the universality of TM_1 automatically makes TM_α universal). This may have been discussed in (J5 ≈ Jan 5, 1959). I.e. TM_1 would be automatically modified so as to take optimum advantage of the fixed form of TM_2. Perhaps has to have a minimum amt. of Goodness for this to take place.

Then O.K. — say I have this fixed TM_2, and a univer[sal] TM_1 that is optimized with resp. to a mean-speed-of-correct-respons[e] (last-n-trials) reinf. criterion. If I do some "manual" placing on the hill, I will also have to put TM_1 [on] a hill that includes some hill-climbing techniques! 269.15

.35
.36 SNJ: **On Network TM's:** The big conceptual problem in [the] TM's was how to make the analog of defining a useful sub-rou[tine] in a sequencially pgmd. computer. One way, is to have each fre[e] net be described by a list in Polish notation, so that defns of sub-nets

269.01

2.29.61

phang

p.268.40: can be made and used.

However, another way to do it, that may be ~~a~~ to way its

If a particular "sub-net" or "sub-operator" is very useful,

reenforced by having its functions (randomly) duplicated in

So a highly reenft. ~~of~~ sub-op., will be duplicated in

(These ll duplications may not be exact duplicates, but will be fu

usually th. same). These duplicate ~~of~~ sub-nets will then

to be used in other sub-nets which will be activated w

net-as-a-whole, gets other input types (other problem

However, my ideas about th. last ¶ aren't very clear as yet.

.15 : 268.35: By going into a bit more detail on this 2 stage TM,

be able to see if its at all reasonable to expect it to get

clever.

In a slitely more general form than 267.01 ff (More in th.

of TM704): We have a universal machine, with a many-compa

determines its operational function. [$\vec{V}$ may be th. "instructions" of a seq. co

or th. x'ln. instructions for a Turmac; or the ~~xxxx~~ ~~wiring~~ connection st

of a ~~xxxx~~ "neural" net, ~~xxx~~ or...etc.] ~~xxxxxxxxxx~~ This Universal

is called TM_1. ~~xxxx~~ TM_1 is fed various inputs a

various outputs, and is given a G value, for each (I, O) pa

TM_2 looked at th. set of known $(\vec{V}, G(I,O))$ quads., and

this, using a rather long-run, reenft. schedule, makes th. next

TM_2 usually doesn't use very clever ~~or~~ inductive methods. ~~E~~ Th

method of correlational coding, or some even simpler method

may be used. G may often have only th. values +1

~~—~~ or TM_2 may default th. G info to that ~~x~~ range.

It may be possl. to show that a simple random net TM, with

connections, that is reenforced with a proper temporal ~~xxx~~ weighting, is a

~~s~~pecial case of th. above!

A refinement that may be very useful: ~~xxxx~~ In certain

2·29·61

Plans

.01 : 269.40 : cases, there is only 1 rite answer — this fl
all other replys get G = -1. Or there may be some other simp
by TM$_2$, to figure out G(I, O). In such a c
we may want TM$_2$ to be able to "exercise" TM$_1$ by pa
th. same Inputs to TM$_1$ again and again, and suitably
his replys. th point here, is that all this "exercise" does no
sample size — and we want to get optimum reply performa
a gen. (small) sample size.

SN : In many (perhaps most) extrapolation
we are given th. parsed s's of a lang, and
do make inductions on new s's. Some examples
are computer pgms, LISP, Mathl expressions in
(operators)
Polish notation, Spoken English as a typ. seq. — Usua
th. typ. seq. is arranged for the child so that he learns phrases
that are meaningful, and E then he can assume these p
are usually pos's whenever they occure.

Th. parsing may not be optimum, hvr, and if we had
$(\alpha(\beta(\gamma(\delta))))$ [Greek letter are ops], we mite do bet
with $(\alpha\beta)(\gamma\delta)$ — i.e. if $(\alpha\beta)$ and $(\gamma\delta)$ were ops
by freq.

Some impt. ideas relevant to a good 704 TM:

1) Th. possy. of a BS TM, using only a universal TM$_1$, w
a suitable reenft. routine, based on history of (I, O, G) sets

2) Use of th. IR npm discovery routine for hill climbing, to w
optimum pgms for this TM$_1$.

3) In addition to npm discy, use all other regularities in seqns. that
are to be used for computer pgms. Express pgm as a set of ops
in Polish notation. → see 7)

4) A "good Typ. seq" is one that encourages th. "rite" learn pmbrs
with rather short runs of guesses.

2.21.01

5.1.61

Plan

01 : 270.40 : 5) Th. idea of making G = +1 or −1 only, ... a correlational code (like IR code) for predicting whether ... will have G = +1 or −1. → I think it may be then p... to construct pgms/seques that have to, a pri a hy expe... in any particular case. (see 10)

6) There seem to be some good suggs. in TM704 β 730.01 ff
Also see th. long review of BSTM. ≈ Phys β 529.01

7) (From 3)): If one has a pre-parsed corpus, it is easy ... a suitable stochastic WPSG — i.e. th. xtn. probys are easy to get. to get a WPSG that fits much better, ... ngms should ... be defined. This will not change th. ... set of s's ... by th. gramm., but it will make th. ... gn. corpus have h... — which is very impt. If one doesn't define these I think th. resulting "fit" of th. gramm. is not very go... — e.g. w.o. these "defs", one wouldn't recognize ... figures of speech, etc. [this idea is relevant to

8) Actually, I should first do an analysis of th. applicn. of th. (correlatl.) lang. to Hill climbing. Then try to extend th. m... to PSG's with pre-parsed corpus.

9) ... A GREAT Genlzn of Th. IR lang: Inst... using ngms of phrases as th. prediction element with Booli... combs. of them, — use "properties" of all kinds, ... Boolian combs. of them. By suitably chosing th. propert... it will be automatically easy to construct objects of hy expe...

All sorts of properties can be considered — e.g. th. presen... certain ngms, or ops, or ops used in various ways, or th. presen... phrases ... of certain kinds, etc.

At a "hyer level" we will want to have a "property generator" ... This device will combine properties and other abss., to produce

5.3.61 TMJ.

Play

.01 271 40: new properties of hy expected "U". We have, a rather natural way to get th. apsi U of a prop. Th. actual U of a property will also be assoc. with which it can help create strings (≡ pgms) with th. Perhaps certain properties will be assoc. with certain betw. probys involved in th. MC generation of th. trial.

10) / from (5) Th. idea is to first link th. "properties" to th. generation pr. used in th. MC gen. of th. pgm. strings. This involves th. of dividing th. "G" into 2 levels, then getting th. probty a gn. string having a G of +1, for arby. input. We may be able to make an approxt. gaulzn. for an arb. G spec. by giving non-equal wts. to all G's. (Ordinarily, for G = only, all G=+1's are gn. = wt.

11) A refinement of th. 2 level G idea! Have n. A good property is one that gives / th. set of known pgms a very skew distribn. on n levels and ∴ ↓ coding reqt.

12) Since many successful ($\bar{G} \approx +1$) pgms will have many long s. in common, we should use a search method for them, that is sig. difrnt from th. methods dcrbd in "IR". Hrr: note that we start using th. "correlational coding" used in IR, use th. method of Part 46 to find sig. ngms. We may catch th. long ngms at that stage, or we might a special pgm that looks for long ngms that are repeated > 1 time in th. corpus. Th. putting of th. corpus in numerical notation (say radix 27) and comparing "size" o. subsequences starting at various pts, might be useful. Th. trouble with Part 46, is that it may mal-parse a corpus, so that repeat. of some fairly long ngms are not noticed.

5.4.61

Plan

.01: 272.40: Note that I can arrange to have th. pgms if I write them suitably. Then th. problem of and property dcvry (for props = Boolean combs of ngms) will simplified. For this pre-parsed form of pgms th. o operator TM form of TM of EPM B1110 would

---

July 25, 61: Recent work (273.1.01 to 318.40) has been on impt. probs. involved in what I had considered to be a "final" — considerations of a TM_1, TM_2 comb. — / conds. under which (open loop) would dcvr that something like $TM_1 = TM_2$ would be good idea. Much work had been devoted to deciding what Gorc for TM_2 to use on TM_1, so that $\approx TM_1 = TM_2$ becom

Also, much work done on devising some new types of for use with operators that are to be improved for various kinds of hill-climbing probs. in h.r. and s. m these codes being devised to prevent ad-hoc mess.

Next, (BS318.25), it seems clear that any essenti "deterministic" TM would probably not last long because of comp failure — No!! see 323.22ff. — and so th. random-net type would have to b worked on more. (unless there is a way to get my "Z to do work — which is, in a certain spirit, a "deterministic" computer in th. sense that I can use its principals to construct what essentially a very large "derm." computer, that has ar large reliability.

One of th. basic motivations for th. careful analysis of Gorcs for TM_1 and TM_2 was that I wanted TM_2 to be able to supervise TM_1 with soft. competence, so I could leave th. system alone with a set of problems (typ. seq), and not

319.01

July 25, 61

Plans.

.01 : 273.40 : have to supervise th. system myself. Also, I co
for hill-climbing
myself, w.o. having to worry about whether my trials were a.h

— And also. I wanted to be sure a $TM_1 = TM_2$ com
find some solu. that "satisfied itself" — but did not satisfy

It appears that th. final TM will be
net TM. If I made a TM_1, TM_2 like 313
I could make Rj th. dcrn. of th. connections (or th. "co
input lines") of a "random net", and have th. "deta
$TM_1 = TM_2$ comb. optimize th. net (as an operator).

.15 However, even so, I will, eventually, have to understa
nets, and above all, how they can be made to corresp. to T
for th. uses I am putting Turmacs to — i.e. how to
"definitions" in a Post system. and how to implem
.22 any other coding methods I can think of for Turmacs
.23 So, a seq. of problems that should lead to a final, wo
random net TM:

(1) Study and summarize my work on codes of various ty
Try to ganlz. th. work as much as possl.

(2) Write Gorcs for TM_1, TM_2 and for various kind
Ranfd. op. TM's. In th. cases of th. latter, devise some
seqns. and see if th. ranf. criteria are reasonable.

(3) Perhaps make a TM_0 to solve certain types of pr
(— e.g. arith. learning) and let me be TM_1. Make su
that TM_1 uses th. rite Gorc on TM_0.

(4) Make a TM_0 to solve probs. with a continuous exter
as 313.04 -.10.

320-01

July 25, 61

Plans

.01: 3.19.40 : (5) study / Boolian network machines and random B nets preparatory so that I can make a correspondence and Turmacs (as 3.19.15-.22)

(6) Some work on HR and SM — and possibly the coding of continuous quantities.

(7) Design final "random net" TM, using Th. EPM section has some pretty good ideas on random net TM. Also, & ta in (most recent) Dart (I think) is a long section on Th. electronic Realization of a recent for random nets using H drum analog-wise. Also see recent work in section on "Rand" for good ideas.

SN What I want to do is make a tentative ordered list of things to work on, culminating in Th. final TM. 3.19.22 is an attempt at such a list.

An impt. Q is to what extent work on any of intermediate projects should be completed — i.e. into review articles, formal ZMTB, computer programming, etc.

Th. point of completing some of Th. intermediate projects past Th. point of apparently max utility, that a large project (e.g. final TM) may often more rapidly completed on Th. whole, by actually completing some progressively more Th. projects first. Ordinarily, an important sub-goal of these completions, would be to get continuance of funds a/o extra interest in Th.

3.21.01

July 30, 61.

TM

Plans

321.40 work. At th. present time, such goals a
small, positive (if not negative) weight.

A sequence of projects that are rather spac
culminate in a reasonable, malfunction-indic. TM:
Th. folg. are all operator TM's, with recent. g
for each (I, O) pair.

① Operators are ~~desc~~ dcrbd by strings of sym
and each such op. is gn. a probty by some ro
based on my a priori ideas about which ops.
"reasonable". Various ops are tried for th. strings of
and they are each given a G value by th. Gorc. This G
somehow includes an emphasis on "more probable" op
Also, somehow, th. G's of past ops tend to inf
probty values in new trial ops. We try, of c
to get an op. of max Gorc. This Gorc is,
think, something like th. expected G that th. gn. op.
will have for th. next (as yet unknown) inp
ov th. exp. G. averaged over a seq. of input
in th. far future.

② Same as ①, except that instead of "th. mos
op" we simply select ops that are dcrbd. by s
of symbols of minimal length. Th. set of symbols
fixed and finite.

July 30, 61

Plans

322.40 : ③ Same as ②, but instead of mi[…] for the ops., we make all code lengths the sa[…] long, hvr.) and try for ops that have the max no. o[…] ~~rapr~~ derns. ~~that~~ of that length.

④ Same as ③, but we somehow want to[…] ops. ~~[illegible]~~ ∋ all pts within a distance R about[…] pt. in code space, are equiv. derns of that[…] We want **R to be max.** (i.e. max redundan[…]

⑤ Same as ①, ②, ③, ④ but the corresp. […] in ~~[illegible]~~ neural nets.

.22 **IMP. NOTE.** → One of m[…] reasons for working on "random net T.M.'s" is that in a[…] TM big enuf to be interesting, I thot that (273.30) component failure would be […] to get anything interesting done. Hvr. — not so. If […] use 3 ordinary computers in ‖, the freq. of 2 failures in corresp components, is quite small. Say the failure rate in a 7090 is 1 in 100 hrs. They do 5 μsec/operation so this is $2 \times 10^5 \times 3.6 \times 10^3 \times 10^2 = 7.2 \times 10^{10} \approx (10^{11})$ operations per[…]

Consider a human neuron to be equiv. to a 7090 (but[…]

Say the human brain has 10^{10} neurons — each with .01 sec "mean response" "computation"[…]

In 10 hrs it does $10^{10} \times 10^2 \times 3.6 \times 10^3 \times 10 = 3.6 \times 10^{16}$ ~~[illegible]~~ comps.

3 7090's in ‖ would do $\approx (10^{11})^2 \approx 10^{22}$ "computations"/failure.

326.01

Aug 1, 61

Plan

.01 323.40 — So, if we build complex computers in p
get plenty of computation befor failure — equiv. to
of human computation — or ~ 300 yr. brains!

How many 7090's in // to simulate hum. brain power?
Brain: 10^{12} comps/sec. 5 - 7090's: 10^6 comps/sec —
So we would need ~ 5×10^6 7090's!
We can reduce the requirement by ① Saying 7090
complex as 100 neurons. ② speed up 7090 by
— So this gives $\frac{5 \times 10^6}{10^2 \times 10^3} = 50$. So Hum. brain equiv. to ~ 5
these rather fast computers. (We would have to triplicate, so say

But anyway, the moral is, that from these very ruff co
we can guess that we can get useful TM's w.o. using rand
Perhaps by the time I get a non-random net non-(random net) comput
going, I will have gotten some good ideas on how to do the tru
net TM. Or perhaps I can make a TM big enuf to work on the
problem.

.30 The type of TM that I am closest to being able to build
a MTM. We feed in math problems as inputs, and exp
answers as outputs. We have a routine of selecting the "M
"R. more probable" ops that give R. known rite answers for all k
past cases. → A ops ∑ of type machine might be
From such a TM, could we develop toward
a hill-climbing TM ∑? — i.e. an h.c. TM is, perhaps
in general, is capable of dealing with more realistic situation
and thus says. than ∑ an MTM.

327.01

Aug 2, 61

TMJ

Plan

01:326.40: ① A simple kind of h.c. TM: Have 2 levels of G (~~the~~ apriori th. median G) and use a correl. lang. to divide trial levels. Devise various "qualities" to use in th. correl. lang. could also be of th. sort that will help create new tria

An example of such a TM: TM_0's are pgms, R represent operators. Inputs are fn. and each R gives for each input. G is evaluated externally (usually partly on of speed of response). TM_1 tries to create R's maximum mean G (or some other G or c). ~~Also~~

Note, hwr, that this is an example of a noisey h.c. I think many (if not most) are of this type. If ma be always possl. to get rid of all of th. noise by comp (in some way), (th. "Expected" mean G in view of th. known in

② A more complex TM_2 that tries to find th. relation be th. empirical G and its trials, and thereby climb th. hill with fewer samples. Th. clevarness of th trick depends on th. "cost per sample" to TM_2 as oppo to "cost" of computing time.

③ An additional refinement to ②, would try to get th. probtys of all possl. <next input>'s.

④ A further refinement would hill-climb with a G that was based on an (only probabilistically known) input of the distant future, or upon a mean over a set of such inputs.

⑤ A slite refinement of ④ would try to define more exa just what I want in th. way of optimum TM behavior in th. long run. It might be something close to

3.28-07

W, Aug. 2, 61 TMJ

Plan

.01: 327.40 : a large mean G in Th. for future. Per
goal would be that of developing TM as rapidly a
with as short a typ. seq. as possl. into a dev
could solve real problems — i.e. problems that I actu
want to solve and can't w.o. a TM. A simp
of this type, would be a S.M. a/o H.R. T.M

.40 Much work on just what goals I want for a TM that is good

Now, for further "Plans" work: Write out more de
on Th. MTM of 326.30 and Th. h.c. TM's that follow it.
if they do, indeed, form a reasonable sequence, so that
model is worth while for itself — keeping in mind, tha
very often a very difft project is bes
accomplished by leading up to it by a seq. of simpler (total man hrs)
In particular, Th. coding methods of code 273.1.01 — 3
Th. codes are le
in Th. is sac
& should be examined, since they are very relevant.

.25 (This is Re MTM): Rather than Th MTM of Dart Rep
use an MTM utilizing whatever fast Th. functions are in R. with unit
Th. computer That is to be used. e.g. have it learn w
((18 + 341) − 8) ÷ 3 is 3, by numerical examples.)
The problem will be one of constructing operators "simil
to those that have been successful in the past — when
have to modify an op. (i.e. when Th. op. has come to a pro
that it doesn't respond to "properly".)
Th. problem of typ. seq. construction is paramount in
MTM. Any skill I gain here, will probly be direc
applicable to hc TM's.
→ Also, I want Th abstractions (≡ categorizns) used by Th. TM to

329.07

Aug 2, 61

Play

.01 328.40 be very close to those that I use. (
th. tng. say. — get a ruff tng say. —
abss. I use, then add or modify exa
that TM can learn these abss. properly

.07 ① Th. simplest G.C. TM (of 327.01 ff). Here, various "quali
be used to categorize objects to predict whether their G
be above or below th. median. MTM (of ②) could be
case of TM ①, if we ~~let~~ give successful R's → G's
and unsuccessful ones G's of 0.

Th. trouble is, this isn't quite what I had in mi
MTM — i.e. I expected to ~~have~~ have/ not more than one comp
successful R available at any time. — Even so, th.
may be formally quite close.

Hwr., I do feel that th. coding methods used in th. M
are very likely to be useful for the h.c.TM problem. Hwr. I
it would seem that this must be true, since MTM recog
certain "regularities" in the successful R — and these regular
must be useful for hcTM to distinguish betw. R's of hi
(i.e. G=1) and R's of low G (i.e. G=0, unsuccessful R

At any rate, I do feel MTM and hcTM to be
related, and it would seem worth while to work on both
th. same time — perhaps modifying both until they b
more similar. It would seem that th. problem of desig
a good tng. Say for MTM, should also be impt. in h

Another imp. Q about hcTM is whether it will be able to
climb noisey hills.

330.01

Apr 3, 61

Plan

.01: 329.40: ◎ On MTM: Say I were successful in [illegible] fairly good MTM — ② just what would such a machine d[illegible] how would this be a step toward hcTM?

② I would have a Trg seq. consisting of a seq. of inp[illegible] known proper responses to these inputs. It is th. job of [illegible] to devise an operator (≡ seq. of machine instructions) th[illegible]

Ⓘ Expresses th. relationship of input to output for all cases

Ⓘ This operator is of ~~machine~~ small rcost ⌈(not ready minimal) ∑ actually th[illegible] must be > a certain amt., depending on th. size of th[illegible] seq. and th. "info" contained in th. relation betw. ~~its~~ its and outputs. This ~~is~~ Q is discussed much in Code 23[illegible] at last

The operators, R, will incorperat in itself, all of th. heor[illegible] tricks and abstractions that I think are useful and not [illegible] for solving such problems as I'm interested in.

I may or may not want to ~~search~~ compute to see if th. [illegible] of R is by enuf with resp. to th. known trg. seq. See th[illegible] ideas on how to do this approximately — or get upper bound o[illegible] P cost threshold.

ⓑ To answ. this Q, I should ~~to~~ perhaps write a dcrn. of [illegible] what I'd want th. optimum h.c. TM to do, (there is much discn. this problem in) and then list some approx solns ~~approxns~~. that might be close to ~~that~~ th. above "both" idealized soln. to MTM. Th. "approxns" will be both in ①goal ② actual attem[illegible]

Some kinds of approxn. that might be considered:

① 2 or more levels of G and correl. codings, using "properties".

② I think there are several approxn. ideas in [illegible]

③ Th. pure h~~.~~c. prob. (w.o. trying to predict inputs) and th. noisey ~~hill~~ hill, and th. noiseless hill.

331.01

Aug 3, 61

Plan

.01:330.40: Perhaps what would be useful, would be to several ways in which MTM is a special case o[f]
Also, it would be useful to note in just what ways are used by hcTM — since this would make clearer its to MTM.

Note also, that there is another kind of h.c.TM — th[at I'm] probly not interested in just now. This hc TM is very relat[ed to] SM and H.R. In this h.c.TM, th. TM proposes input objects x_i (vectors) and is gn. G values ex[...] Th. goal is to find an x_i with max G as soon [as possible] (This goal has to be more clearly defined, but that's th[e idea] anyway.

Hvr., I'm not so sure this is difrnt from what I want now, in th. R's corresp. to th x_i's — and I want an R that w[ill] have hyl expected G for th. "next input."

There are various kinds of hcTMs — one's which consi[der] th. structure of th. "inputs" to varying degrees.

.30 (1/2) Woops! Another possl. intermediate step betw. MTM and hcTM! A MTM that can work even if their are occasional mistakes in th. Trg [seq.] Such a TM will have to have a much larger trg seq. hvr. The [goal] is to minimize th. total r cost of Ⓟ th. operator ② a lis[t of] deviations from th. predictions of th. operator. This is ess[entially] a digital version of th. generalized continuous curve-fitti[ng problem.]
This problem is not at all far from th. "simple" MTM, but it has th. advantage [that] all R's can be gn. a numerical "Goodness of fit" with an "almost fit" be[ing] useful. In th. pure MTM, we throw away info about "near misses."

Aug 4, 61

Plan

.01 331.40 : In general, what I can do now, for any o
is to write out a partial solution or type of
from entirely intuitive considerations — then th. cod
of ind. inf. will suggest a quantitative formulation.

.05 ⓪ : While 330.01 expresses th. goals of MTM, (and to some
of "noisy" MTM (≡ ⑤), th. methods by which this goal
achieved are not made clear. These are th. search me
"heuristics". A method for MTM that I had in m

.20 Another big problem that I'm not too sure of is th. Q.
how good are non-optimum sets of codes in making pred.
Can th. use of any such code be made equiv. to an optim
code, using a corpus with somewhat difrnt regul
in th. "distant past"?
I did much work on this problem recently → i.e. c
when I was asking just how good predictions I
expect from my 7090 pgms for coding with binar
definitions. I think this pgm. was debrbd. in ZT
See Cod 311.15 for newer discn.

.30 → as follows: Say one has an R_i that satisfies th. typ
up to T_1. At T_1+1 we get a counter case. We then
to add to the operator R_i, ① methods for recognizing
th. new probs. differ from th. old. (≡ an obs. operator) ② Ne
sub-ops. for dealing with th. new kinds of probs.. Both ①
are constructed using costs based on th. statistics of
the R_i to which these new ops. are to be adjoined. Remember

333.01

Aug 4, 61

Plan

.01: 332.40 : to include th. costs of punctuation!

.02 If th. new cases cannot be taken care of at r
cost, then ~~there~~ a way is TM will try to recod
of th. more recent parts of R_i (or use some o
lower pcost codes that were found) in attempts t
th. statistics and ∴ costs for th. new set of r
This is called "back tracking." ~~Th~~ Back track
facilitated if ~~some~~ codes are retained that are
too close to optimum, — as well as retaining op
.18 codes. → see 334.20-30

The above seems to be close to th. way humans f
th. requisite search for codes.

① hc TM : What I had in mind, intuitively, is
like this:

.22 ① A hc TM that doesn't seem too far from ~~I~~ my intoitive con
of th. way humans do it, and seems close to ~~the~~ M
(or, more exactly, to ½): We take th. ~~largest~~ N, (I
pairs of known hyest G, and try to fit a curve "
This "curve" is an operator, and ~~a~~ ∴ (I_i, O_i) of th. o
be "close" to an empirical hy G (I_j, O_j) by hav
I_i close or identical to I_j ~~and~~ ~~have~~ have O_i close or identical
we make N "as large as poss!." — ~~then~~ so as to
.36 size, yet if N is too large, mean G will ↓ → also scor
.37 Another, more general way to do this: Use all known
(I,O) pairs, and chose a op. ∴ → its (I,O)'s and close
to th. empirically hy G pairs, and distant from th. low

Aug. 4, 61

Plan

.01 : 333.40. G pairs. This latter seems fairly good, and [illegible] considerably more available info than Th. p[illegible]

.03 ① A typ. seq. for # c TM that would facilitate learning (... be fine for early typ., if not all typ), would be to ... th. same input to TM repeatedly, until he gets a[illegible] of hy G [illegible] for that input. Other, more sophisticated ty... be more like th. parent-child relation e.g. Giving reward. for Th. learning of phrases in ...

Another advantage of Th. h c TM of 333.37, is that th. ida... in h.c. problems, "closeness" is used by humans, and it would be well to do... in a TM.

I have done some work on Th. h c TM of 333.37 previously ... Phase notes, but I don't know just where. I think [illegible] I ... to use negative wts. for statistics on low G (I, O) pai...

.20 : 333.18 Note that a "part" of an op. (i.e. its response to a ce... sub-set of inputs) can be slightly modified by either ① ... a recogn. op. ② changing an "action" op. ③ changing th. (nature ... that is called into play by a recogn. op. ④ 2 or more of th. ... Also, we can leave each part of an op invariant by doi... 2 or more of th. above — so we may want to do on... to fix up part of an op., then do another to cance...

.30 Th. evil wrought by this change.

.31 ① A simple approxn. to 333.30 is 333.22-.36. If we don't use th. "closeness criterion" and have our op. actually pass thru as many of th. best N (I, O) pairs as possl., we will simplify things much, and get a lot closer to TM (½) ← Another useful modification

see Part 340.05 for objection - also Part 342.03 for 335.0

Aug 4, 61

Plan.

.01 334.40 is to use "closeness", but only th. simplest, m
(e.g. physical resolution, e). This will ↑ our samp.
.03 cost in time and memory.

Note that this he TM — even th. simplest type, will
lot more than MTM. This is because th. natur
"the N (I, O)'s of hyest G" changes with time. I
one discards old members of th. set as new ones
This means that if th. gen. method of 332.30–.40 is
be used, that a lot more revision (and perhaps "backtracking" (333.02
to be done than in MTM. (like 334.20–.30 ap.) will

= Epist MTM Th

A big ref. on an operator, observer T.M. is EPM 1107
that 1110.01 ff in particular. It seems to end on Epm 1143.40.

Epm Main
1110,
1143.

.25 But while this idea is fine, and I may, ultimately
I will at first start out by doing how I, imbeciline
do solve certain learning problems. I will first write
how I solve do these inductions — for th. entire type seq,
Then I will devise a "lang" to express these ideas form
and use th. coding method to evaluate things and see how it
works quantitatively.

So now, write a review of just what I expect to
in th. way of MTM and he TM, giving as much detail
both, as I can.

336.01

Aug 7, 61

Plans

.01: 335.40: O.K. — So first:

TM_0: (Refs: 328.25; 330.01 (Tells what TM_0 does.); 332.05 (Tells proposed method for TM_0 op...)

Anyway, TM_0 is a MTM; Inp seq. of inputs, and assoc. co...

No errors allowed in inp-seq.

.05 **$TM_{\frac{1}{2}}$**: (Ref: 331.30) Same as TM_0, but it will a few occasional errors ... Seq. are permissable. Actually $TM_{\frac{1}{2}}$ involves a more exact ... of the efficacy of the Govn. of this operator as a kind ... "short code" for an <input-output-set> corpus. Work o... should be done in such a way that $TM_{\frac{1}{2}}$ will be ... and easy. I think that this will be possl.

TM_1: (Refs: 327.01 (This also derbs future possl. refinements.), to 328.19; 329.07; 333.22††; 334.03; 1337.31) This is a simp... of h.c. TM. [This] ref. gives [illegible] the direction of further develpmt ... more powerful h.c. TM's. Code 273.1.01 — 311.40 also has much on... the Q. of just what is the Govc. of the TM if we want it ... BS. Also BS section has much on this Q. Essentia... it tells just what ultimate goals (of excellency) an h.c ... will have.

→ **A very basic Idea**: That an h.c. TM — in addition ... being able to BS. itself, can be trained to do anything ... one would like. I think there is a section in TM704 ... that does outline some inp. seqs. for an h.c TM, to get ... it to do various diff't, interesting things.

→ **Another very Impt. idea** re Q. that needs more wo...

How much prediction accuracy can be expected from ... a gu. non-optimum coding method? 332.20 discusses this a bit.

→ Perhaps a main goal of the presently contemplated seq. of TM's, is that I can make a h.c TM, that can B.S itself. I will want to ... 337.01

Aug 7, 61

Plan

.01: 336.40 develops TM_0 and TM_1, in II, so as TM_1 will be developable toward a cRTM of Th. T like "pgm. improvem can work problems ~~of the type that~~ :. th. problem of improving itself.

→ Methodologically, th. most impt. idea, is to first th. ~~operation~~ lang. segus and operations of these on a very intuitive level, in ord. long ~~the~~ (≈ Engl Ran slowly put in more quantitative form. (ie. 335.

Work on TM_0, and TM_1, will be continued in Th. "L

Section ie. Dart 338.01.

.20 → On Dart 341.35 [to 342.04] it is decided to work on RTM (Fraenfd because this TM is capable of anything a human can do :. can be ~~a~~ BS TM ≤ — also th. hours a close to human hours, since humans are close to, if not exactly, a ~~a~~ RTM. Also RTM can be ~~be~~ taught Q's in English (≡ QATM), which is, I believe, a good form very powerful.

Actually, th. TM, that I've been contemplating ≡ a R — but I wasn't sure that I knew how to make a BSTM o it. However, since we can ask a RTM (→ QATM) to "improve himself", RTM can be a BSTM.

→ Dart 344.15 discusses a very impt. idea in a RTM, i what extent such a RTM can ~~be~~ learn eventually learn very fast even tho its routine hours for search are not v.p. Dart 345.10 extends this concept to "reliability" (≡ performance), cost, and wt.

394.01

Th. Sep 7, 61

Plan (396)

.01 : 337.40 A lot of recent work in Dart, has been in attempting th. possy. of Dart 344.15 — ie. a simple / TM operator w automatic simple search procedure, that somehow is able to develop equiv. to a very clever search procedure.

At th. present time, I'm not too sure as to just how work, tho. around befor Dart 386 I was fairly optimistic it was possl., and had th. germ of an idea as to how it might be

.15 The discussn. sort of culminated in "Th. system" Dart 381.03 – ~~387.35~~. Th. desirable features of th. system listed on Dart 387.06 ff — followed by a few of th. impt. probs, t

At th. present time, I'm not really sure as to how make an L_3. L_3 must be able to (eventually) devr. all possl. seq in th. set $[M_i]$. However, even if this part of th. sys doesn't work, I can still be L_3 for a while and still have all advantages of 387.06 ff, except #3. (sort of) ie. speed in search, as this seq. continues.

But th. remaining advantages are considerable

Also after I write up a few L_3's, I will have ideas on how to improve them automatically, and perhaps get th. TM, to do this improvement of L_3. — or just make a hyer order search TM to improve L_3. With such a hierarchy (even w.o. $TM_i = TM_{i+1}$), th. power of th. may be enuf to work signifit. probs. — such as externally asked and self improvement.

Dart 387.35 That th. system doesn't try to anticipate th. trend of probs seems to have the most serious criticism of th. end-goal of this system". This deficiency will certainly slow th. system down in th. typescript; Plan396.35 may have a way out of this.

395.01

F Sep. 8, 61

Plan

.01 : 394.40 : Th. folg. will be a plan of as much detail as po[...] ("TM.") "System" (called **TS**). Much of this follows Dart 381.0[...]

Def ← see 396.15 for meaning of "TS".

1) First step: Write a trg. seq. of Arith. problems[...] this trg. seq. should end up with problems of some con[...]

For th. bgn. of th. trg. seq., see Dart 356.21 ff / Dart 357.[...]

see part 2 354.01 ff for more specific solns. of simple probs.

[...]3?? Dart has some con[...] goals of trg[...]

Differentiation of alg. expressn. (≥ 1 soln. possl.)

Trig. identity simplifn. (" " " ").

"Proofs" of theorems or proofs of Trig. identities

Soln. of linear eq. in 4 unknowns after learning to solve them [...] and 3 unknowns.

Th. concept of "solving for x" by finding a no. that sa[...] eq. when substituted for x.

In order to do this for many problems, th. concept of [...] must prob'ly be learned, since most numerical solns. are not "exact".

Th. soln. of quadratic equs. by ⓐ factoring ⓑ a [...] method (equiv. to th. quadratic formula) — e.g. "completing th. square", a[...]

Th. soln. of alg. and transcendental equs. by various successi[...] approxn. methods. E.g. linear Newton, quadratic Newton, o[...]

Simpler methods — (e.g. successive subdivision of x axis to narrow down[...]

Problems involving a "search" so that idea of "improvement" is useful —[...]

from th. above set of problems, I should be able to design [...] seq. of some interest. Th. interesting Q is always how [...] many trials (and how much time/trial) must TM expend [...] to take various of th. "conceptual jumps"?

My present conception of th. operation of this par[...]

MTM: We start with a set of " " complete for universa[...] computer orders, [These need not corresp. to any existing or projected computer] [...] or a set of operations with rules for combining them (like th. ob[...] algebra).

F. Sep 8.61

Plan

.01:395.40: Th. "Probs" are fed to TM as inputs, to find simple-as-possl. operator that will work all probs. corre... Trial ops. (called M_i) are at first constructed by r... of Th. basic set of computer orders (or operations, or... This results in a seq. of ops. — each of which has be... up to a certain pt. in Th. trip. seq.

After a few such ops have been obtained, "L_3" ((second)? order TM) looks at, say, Th. last 10 successful ops. tries to find regularities in them that would be useful ... reducing Th. no. of trials necy. for Th. next problem. At ... these regularities will be npms defs and perhaps ntps de... As these operators get bigger (more complex) we ... begin to look for more complex regularities (& in co... .22 & larger "corpus"). Ideally, any regularity that I ... think of intuitively should be "accessable" to L_3. Th... exact meaning of this isn't entirely clear to me, hwr...

Th. regularities L_3 observes should be expressable in Th. of parameters (or other features) of Th. creative s... pram. that is to create Th. M_i trials. One big prob... will be to design a universe of Grammars G_3, s.t. Th... is possl. Also, we want to be able to construct ... regularities in G_3, to see if they hold in Th. set... Essentially, we want Th. new trial regularities in G_3 to b... formed by combining Th. old successful regulari...

.35 It is possl., at this point, for L_3 to notice Th. ... sequencial regularities of Th. sequence $M_1, M_2, M_3 \ldots$ → (s... for c... This would get around the objection #5) of Dor 387.35. Hwr., at first, we would n't bother with that.

397.01

F Sept 8, 61

Plan

.01: 396.40:

(M_i is the op. that has been successful on all I,O pairs up to t_i) → (M_i is ... has been successful ... pairs up to t_i)

TM_1 (≡ HCTM = RTM) : If we get properly (as the foregoing pp.), then we can modify a RTM (≡ reconfd TM) directly.

In MTM we have those ops, M_i. Also, G_3 has (a stochastic grammar) for making new trial M_i's. These can be carried over directly from MTM.

In MTM, L_3 can test each M_i created, since single correct responses are known for all inputs. On basis, a particular trial G_3 can be evaluated — i.e. how a particular G_3 is very good if it takes only a few trial M_i's to find one that fits completely all past inputs.

In RTM, an externally evaluated G is given for each response to an input. At first, TM can estimate the $\bar{G}$ and σ_G of a trial M_i, by feeding it inputs of "known output", until some outputs of known G are obtained. In more advanced machines, TM will try to get an approxt. functional relation between G and any (I, O) pair, by seeing how "close" (I, O) in question is, to other (I, O)'s of known G. (Later, more general forms for fitting "G = a funct. of (I, O)" will be used). Using this approx. form of $G(I, O)$, TM can then evaluate M_i's approxtly, very quickly, by using a few previous inputs as tr... and evaluating the expected G's of the resultant (I, O)'s. (Vary by/ signif.),

In RTM, we can then get G_3 to try to create

398.01

F. Sep 8, 61

Plan

.01 397.40 : M_i's of hy $\bar{G}$. As in MTM, L_3 will ... in Th. M_i's of known hy $\bar{G}$.

Th. M_i of highest $\bar{G}$ known up to that time will usually be a ... make a response to any input.

.05 There is some confusion in Th. above discus. of L_3 — for MTM and RTM. At first ① I spoke of finding "reg" in Th. "good" M_i's, then somehow incorporating the regularities into Th. creative mach. Gramm, G_3.

Later ② I spoke of trial modifications of G_3 — as a hill-climbing problem — with hill ht. determined by with which G_3 could find a "consi" M_i (for MTM) a better than before "hyar $\bar{G}$ than before" M_i (for RTM)

The this is probly a less el. soln., I think that probly this will be used at first. A possy. is to use ① find good trial modifications of G_3 (ie. by looking for "reg" in Th. "good" M_i's), then using ② as a less el. ... of Th. efficacys of Rase modifications of G_3.

Th. plan for MTM, is first to construct a typ. seq. leading to some interesting capability and some interesting kind of induction. Next, I will write a seq. of M_i's leading to Th. requirements of (or several such seqs) that "solve" Th. typ. seq., ... Th. basic computer orders or ops. in a "minimal" fashion. I ... try to make Th. solns. use abss. that were useful in ... earlier probs in Th. typ. seq. at each pt. in Th. typ. seq. I will try to devise defns. so that Th. solns. I have in mind are compact and perhaps close to "Minimal".

I will then try to devise an L_3 that can search for

For Sep 8, 61

[redacted] Plan

.01 398.40: "regularities" in th. "latest" of th. M_i's, and
to ① create defns. to ~~make~~ & rcost of coding th.
~~defins. wh to~~ this is always possl., follows by defn. from th. fact that
found is a "regularity".) ② Modify th. Gram,
that it produces th. desired ~~by~~ "Good" M_i set
~~now~~ hy pcost as possl.

All of this manipulation of G_3 and th. search for re
in th. M_i, will be done at first "by hand" —
I find a good way to mechanize it.

Th. finding of regularities in th. M_i: This wi
Def done by starting with a search simple regularities (≡ regs), then us
th. successful regs. to combine with eachother to
produce new trial regs., ~~which~~ which are then trie
and then etc . . . . .

Th. thing that I'm vagust about is G_3 - its g
form and mode of modification.

In th. trg. seq. — try to keep things on a pretty in
level befor deciding what kind of order a/o ops to use.

Do th. same thing in discussing "regs" found by
i.e. keep it intuitive (i.e. in English) as long as po

.30 ~~As~~ An example of what "intuitive" form of th. trg.
write mean: " First MTM will learn th. meaning of ~~+~~ +, then —
× and ÷. Next, that — is th. inverse of + and vise
" " × " " " ÷ " "
Next, combined operations — evaln. of expressions containing
+, —, ×, ÷ and parenths.

Or use polish notation throuout. Unfortly, this make
th. associativity and commutativity of certain operations
have low aprip.

400.01

F Sep 8, 61
Plan

.01: 399.40: Next, ~~example~~ solve simple linear equs. — 1 un
" " " " " 2 "
" " " " " 3

Th. above trg. seq. may actually be suf. to give inte
induction probs. e.g. How big a jump would it be fro
.10 3 unknowns to, say 5 unknowns? → 401.18

If th. above trg. seq. is inadequate try 395 Plans
probs. that mite be made into an adequat trg se
An Alternate trg. seq. would go from linear equs in 1 unknown, to quadra

Things to discuss further:

1) It is possl. to order th. trial M_i's rather simply, in looking for o
Def is "consi" (i.e. consistent with all previous (I, O)'s) in MTM. ~~Ma~~
Defining ugms and any heurs that search spaced ~~s~~ sort of mo
apriori of M_i's. Discuss interaction betw. heurs, apriori,
"info in th. <corpus up to now>".

2) Th. details of th. action of L_3 or G_3 are not clear. I s
to have 2 difrnt ideas in mind. (a) "Randomish" Mutation of an "almost a
G_3 to produce new trial G_3's (b) Noticing regularities in th
set of "Good" M_i's and incorporating these regularities int
I should be able to merge these 2 approaches. (398.0 with
Also, I should define G_3, L_3, ~~a~~ and "Th. hyar
TM" more exactly.

3) 396.22 : "That all regularities that are intuitively conceivable (by m
in th. set of good ~~[illegible]~~ M_i, should be accessable to G_3."
Th. meaning of this must be clarified. ~~[illegible]~~ One way to
this would be to put examples of $[M_i]$'s and express
th. regularities in them intuitively. ~~It~~ Also these regularities
must be creatively expressably by G_3.

For more general ideas on this consider difrnt forms in
which th. M_i mite be expressed — e.g. in Lisp, in 400.01

Sun Sep 10, 61.

Plan

.01: 400.40 conventional 1 or 2 or 3 or 4 address comp. orders, system, etc.

.03 SN An example of a "sequencial regularity" that should be discovered by G_3 — or should be "wired into" G_3. Say the M_i are in Th. form of recognition ops (≡ obs) & xfun ops (≡ ops) as Dart 360.11 ff. Then G_3 should be able to realize that the form of M_{i+1} is related to that of M_i by M_{i+1}'s having either (a) one new Ob. or (b) a simplification of the set of previous ob_j's. (a) and a new Op_{i+1}. Again it should be built into G_3, that (a) is more likely.

.18: 400.10 : I suspect that the seq. of 399.30 - 400.10 would be better constructed, if I worked backwards — i.e. Take the final induction problem I want TM to solve and list the "concepts" that would be adequate to solve it in a reasonable no. of trials. (Note that it might be expedient to make several lists of concepts — each of which is adequate.)

Again, I would like to do this by starting out on as intuitive a level as possl. — e.g. the sort of "notation" used in the discn of Dart 390.35 - 391.35 — Note also how there, we work backwards

Note that the writing of the trng. seq. in "intuitive" notation makes it possl. to xlt. it directly into any computer lang. type. So — perhaps for studying L_3, I should keep the seq. $\{M_i\}$ in intuitive form and write the trng. seqs. of $\{M_i\}$ in intuitive form first — then, consider various formalisms in which the M_i and in which G_3 may be expressed. Continue in Dart: Dart 402.01

484.01

W Dec 6, 61

Plans

.01 : 401.40 : Th. A.F. seems to be breathing down my neck. Big Wheel at AFOSR. Milsaps, feels that (publishing in reputable journals) is the thing. Th. Q is, how much can I expected to [illegible] year? Say 2 serious papers for Info and Control - 1 for I and C, and another in some conference pr[illegible]. This latter is good, if it is an "invited paper" - because can spend less time preparing it. This latter seems .2 papers a year would not be so trivial as to be Journal [illegible]. Th. 2 papers can be, for myself, good "review artic[illegible]" actual papers presented will be a cut-down versions of my [illegible] i.e. no refs. to TM notes, little discn. of reasons for various [illegible] (work pts. of discn. . . .

More specific present plans: first get 2 TB 141 out. This already written and has only to be proofred.

.20 Next write I C paper ... This will, for me, be a rather discn. of Bl. coding method and various variations, also much discn weak pts. in Bl. theory, and work that must be done. Also [illegible] specific applicns. to specific problems, that have been worked to some extent. i.e. ① Bern. Seq. ② Coding with binary data. ③ Coding with ngm data (I to work this out quickly ④ PSG coding ⑤ Non-Dim. PSG. Coding. ⑥ for IR ⓐ Presence or absence of strings and combs. of strings ⓑ some [illegible] with freq. of phase within Th. doc. ⑦ Th. soln. to Th. B.G. prob. ⑧ Coding of continuous data ⓐ Curvefitting ⓑ [illegible] Pattern [illegible] for printed or written letters. ⓒ How to use randomly chosen params. ⑨ Th. genl. finite history Markov chain.

.35 Next, write Chicago SOS conf. paper ("Typ seqs. for mechanized [illegible]. For my own use, I should write a very complete review of my [illegible] methodology, and Bl. reasons for Th. choices involved. In particular, [illegible] a fairly detailed outline of much of Th. expected future work.

- For Th. SOS conf. paper, just give part of a typ. seq., - some solns., and some [illegible] solns. to th. prob. of generating solns. There

485.01

W Dec 6, 61

plans

.01 : 484.40 : is no particular virtue in writing th. paper ...
— with th. "latest developments" — since, as it is, th. ...
quite long and adequately meaty.

Dec 27, 61 On th. most recent work in "Dart" : (e.g. say Dart 402.01 — ≈480.01)
idea was to follow very closely my intuitive methods of working prob...
quantifying th. probys involved by using cmoziz (≡ coding Meth...
Unfortly, I fear that in this recent work I have strayed f...
intuitive notions, see Dart 440 for explanation of basic ideas of th. "intuitive App..." and have gotten into an excessively rig...
notation. Th. problem of th. 2nd order TM ... That wa...
— that of PSGd WOQ — is fairly distant from intuitive p...

Th. formalism used at th. first level was that of an operator co...
a bunch of (unordered, I think) substitn. rules. Rules of this s...
close to th. intuitive ones, for probs. like arith. eval; ... p...
solving n linear eqs. in n unks. ; solving quadratic eqs. I'm not ...
that there are many other prob. types that can be worked this w...
(i.e. in which this way is close to th. intitive method). Instead of atta...
th. higher order problem directly, try to find other kinds of ...
problems, in which other formalisms are more close to th. intuitive ...
Also, consider th. 2nd order problems as being 1st order for another T...
try to obtain intuitive-like formalisms for them. Th. idea is to ...
a unified set of formalisms that appear to be adequate for all probs. ...
think of. — Note that for some probs, it may be necy. to ha...
long trng. seq before th. solns. within th. formalisms used have reason...
Also note that th. 2nd order problems are somewhat difft. fr...
1st order probs. Th. 1st ord. probs. are ops. — th. 2nd ...
probs. may not be ops.

Also, I'm not too sure about how this idea of nth order TM is rel...
to my ordinary concepts of intuitive regularities.

Feb 1, 62

Plans

.01: 485.40: A rather old intuitive concept of a TM: (This for recent work section in "Dart"). At any time, "total method of coding" - that includes parsing and coding in terms of a set of abss. that have been found to be g new data comes in, TM. can ~~work~~ code this data in these abss. in a routine way.

In addition, TM examines small samples of recent ~~or~~ and recent partially and totally coded input fo regularities. Finding any of them amounts to a new but must be confirmed on th. basis of a larger sample. ~~man~~ man and animals, this "larger sample" is usually obt by waiting for more data input — but it can, in pr (and probly in practice) be obtained from th. recor near past, as well.

A Q. is: Does there exist a set of code symbols (wit and as set of <Things to look for>'s ⇒ using th. technique of finding/defining new abss., we can by iteration of method, discover "all" regularities in a corpus? If cannot do it for "all" can we do it for a fairly larg set of regularity types?

Essentially th. method is that of ZTB141, exce that instead of looking for new zgms (or even ngms), w look for various other regularities in th. corpus corpus. Also we try various parsings of th. corpus, i looking for hyer order abss.

It may be possible to pattern this TM very close upon my own apparent mental processes, since this is appar th. way humans work.

As for th. coding itself: each abs. is gn. a probability which

492.01 492.01

Plans

.01 491.40 . is ≡ its prost. Th. hy prost abss.
which scan/ and code Th. corpus as it come in ||
human CNS, certain ops. have such a hy prost, to
data runs thru them like transducers and actually a
(that to be irrelavant) to be lost). Other
abss. are tried in turn (sequencially) in order of P
Also certain abss. of very low prost are tried at ra
small / recent input samples.

Actually, this allocation of trials / test of an absa is not entirely on
of prost, but partly on Th. basis of how much time
equipment is needed to implement Th. trial. —
a sort of "op. res." prob.

Th. Monte Carlo aspect of Th. coding and is only evidea
Th. low prost ~~abss~~ coding methods. for Th. hy prost abss., Th.
~~and~~ parsing is / certain or almost certain.

An apparently good source of problems for such a machine,
text. Here, Th. old abss. are ~~the~~ whatever parsing rules are k
along with their ~~probable~~ prosts. Say we use aWPSG to start.
could more gramm. rules of this sort be devrd? Other rule t
are Chomsky's "x fmns" - that are / (perhaps) common ways in which many s
~~are~~ parsed. Actually, I don't know if, at present, I have enough
of things to recognize" to be able to devr. many PSG rul
— There is hvr., some possy. that I do — i.e. see recent w
in "Dart", in which I was ~~&~~ using Th. palindrome lang. and
$a^n b^n$ as "test Langs" to test Th. power of Th. methods prop
Th. trouble is — using this approach, I get into th. PSG dis
prob. — which is a diff't. prob. I would like to take a typ. s
in which I ~~=~~ more or less could solve all of Th. probs
w.o. much diffy. Possibly if I used English as input, I could find
other, non WPSG-type rules that would be adequate.

¶93.01 49

+ lang
(8493)
.01: 492.40!

In particular — larger context dependance

Things to look for in trying to find more param
substitu. probys upon in a WPSL — this mite lo
Using previous history of substitus. at a substitu. pt. mite
some of Choms' "x fmn. rules".

Another early idea, was to have TM learn one la
learn another L_2 ("learn" in th. sense of being able to predi
well as possl.), then learn x/tn from L_1 to L_2.
MT learning, is not too distant from input → o
or stim → response learning.

Hvr., at first, perhaps, try to apply this idea to r
regularities other than human lang sequs. (which are prob
rather by state of dvlpmnt) — try noticing regularities in
series — like S.M., or H.R. If human lang. is
stick to more directly intuitive regularity types — then
lots have been found, see if they are strong enuf to do
2 test PSG's ($a^n b^n$ and th. palindrom lang, or "mirror lang.") — for hum
Such a seq. of tests mite be: 2 gm defs, ... ngm defs, optimizn of
ngmsts (i.e. define an ngm as one that usually comes befor a certain word
or by pcost), then define new ngmsts by combin
ngmsts and noticing a certain set of ngms usually precede
previously define ngmst. — Notice structures — i.e. certain general forms
as's or phrases — find ways in which various gra
rules could have been devrd (but use th. grammar rule types of cl
grammarians, not Chomsky. — this is because th. classical gramm ty
probly easier to devr for humans using "conventional (for h
techniques.)