TMY (702.5)

Feb 7, 64
Plans:

Genl notes: At Dart 538 / Ap 20, 62 I didn't have a notation for PSG coding much less th. concept of ~~un~~ unipoly parsed UPPSG's. I wanted such a notation for th. "non-dimens." PSG's — because they ~~[illegible]~~ appeared to be quite strong.

woops: 538.35 mentions a notation also 538.25 mentions PSG Notation used in Inf&Control Paper!

Perhaps it would be well to look into this! i.e.: non-dimensional / UPPSG's ~~[illegible]~~ mite be good enuf for Bootstrapping ~~with~~ after relatively little "hyly planned" trng. seq.

=> Th. genl. idea here. is ≈ TMP 1160.20 (Plans).

My present Idea is that I'd like to have a fairly good idea of TM's trng. seq. — i.e. th. lang. used as well as th. problems — such that they could lead to pgm optzn. ~~[illegible]~~ (i.e. ~~[illegible]~~ looking for ways to improve pgms of various sorts. — make up examples of pgms with redundant or slow sections in them. ~~Ex~~ Make examples of pgms by poor pgmrs v.s. expert pgmrs. (for same problem).

To sort of play down th. BS idea — ~~[illegible]~~ first get an open loop TM that works fairly well — that I can intelligently watch in action. ~~[illegible]~~ A BS TM would probably ~~[illegible]~~ use ~~[illegible]~~ means too confusing for me to understand. This idea of starting a "minimal" BSTM with a long Trng. seq. and expecting it to learn to be very brite "eventually", is rather unreasonable, since I probably wouldn't be able to design good trng. seqs. for it — I could just give it ordinary math ~~[illegible]~~ books to read, but I don't have any idea as to how long it
.30 would take for it to "get off th. gnd." cf. ETM 692.1 - 692.3

There was also some fairly recent stuff I wrote on th. prob. of trying to get a very brite TM not to devr. th. outside world. way to try prevent this was to use a rather simple Go th. TM — so it would ~~[illegible]~~ get reenf. for 1 input , and ∴ not be tempted to do " world."

Feb 7, 64
Plans

01 :702.5.40 What I really want, now, is an index, ~~telling~~ listing various impt. ideas, ~~[illegible]~~ telling where they ~~are~~ were discovered and where they are treated.

Th. devry pt. is impt., because certain work is usually unimpt. (and more or less skipable) if it ~~a~~ is about a problem that had later been solved in a much better way.

One impt. idea: Th. PSG ~~—~~ ~~written~~ coding notation used in
.15 Th. F and C paper: Dart 467.10 – 468.19
Th " " " " was writ in March 62 — It is mentioned in th. past at PSGd 543.02 i.e. Ap 21, 62. Dart 538.01,
Dart 468.20 starts this notation for ~~a~~ non-dim. PSG's — says | PSGd 543.01–545.4
is better — it is — ~~Th~~ A very general non-dim. PSG is defined, ~~very~~ — Also see TMβ 218.01ff (PSGd) for a very general PSL.
in which practically anything is possl. — Examples are gn. of how these genl. ideas
.20 are to be used for 'MT gramms and other complex gramms'

So — perhaps what I had in mind: ① To devise a fairly good trg. seq. leading up to problems in pgm. optzn ② to try to get TM's "work" to be in th. form of finding th. grams for non-dim. PSG's. ③ Try to get a ~~[illegible]~~ PSG discy pgm — and generalize it to non-dim PSG's.

A basic problem here is — can I express my intuitive ideas on how ~~to~~ I solve th. trg. seq. in th. form of non-dim. ~~a~~ PSG discy?

Plans. 399.30ff — dcrn. of part of a trg. seq. — (up to simult. line. equ

Th. idea of ZTB.143 (§2): ① that I would first make an MTM (MTM: only 1 correct answ. to Q's), then perhaps ② a TM that gave probty. distribns for possl. answs. Then ③ An RTM (recnft T.M.).

Th. Trg. seq. would be > ③ would work on pgm. optzn — with resp. to an arby. criterin.

A basic idea was that ~~Th~~ ① was not very far from ③ — i.e. they were ~~[illegible]~~ very similar.

Feb8,64

Plans

1. Importance of devising good lngs. ~~to~~ communicate to TM with.
2. " of getting TM to work probs close to str. of ~~E~~ English.
3. Possy. of using Newell-Simon GPS or EPAM or some other partly worked out system, as a "springboard" — since ~~[illegible]~~ genl. input output lngs will be a big problem.
4. Could LISP be used for genl. communication (input, output) as well as internally to TM?
5.

There are many possl. forms of TM: (1) One most powerful form is RTM (reenft. TM). Here, th. "correct" soln. of th. prob. is for TM not to try to get a low pcost op. that gives good G over past known/ in-out pairs. Instead, th. opt. approach is el. — i.e., to determine th. possl. forms of th. G function, and to optz. th. op. with resp. to that — as well as expected future inputs. One will probly, hvr., use a more direct, ~~[illegible]~~ ~~[illegible]~~ superficially less el. approach.

(2) Pure induction machines: a) ~~[illegible]~~ ~~[illegible]~~ Acceptable, in-out pairs are gn. to TM. — For a gn. new input, he must make a probty distrib. over possl. outputs and print out th. best 3 or 4 possys (say).

b) It may be in th. form of th. completion of sub-corpus — as in ZTB 128.

SN: It is my impression that much of th. above discn. was gone into in much detail in old "Plans", BS, JS, and perh

Plans.

TM18

703.5.40 To construct a Tng. seq. — work backwords:
Start out with a dcrn. of a TM That would be an acceptable sub-goal: Some Things it would do:

1). Understand and ans. in English. Actoally, English is fairly ambig. for in-out put. Thus I will probly. use more exact langs. Th. main pt. of understanding Eng., is so TM can read ordinary books.

2) ~~[illegible]~~ TM will have several modes of operation. For each prob. gn. to TM; Th. mode will be specified. Some mode specifications: a) Which corpus shall be used? — i.e. specify a bunch of sub-corpi. Various sets of These sub-corpi will be gn. various names. b) ~~[illegible]~~ ZTB128 mode ≡ MTM: i.e. find "best" (≡ most likely) element(s) for Th. specified ~~[illegible]~~ pts. of Th. input pattern. c) RTM mode: After TM gives reply, give th. σ value accorded th. assoc. in-out pair. This σ value may be gn. much later, or not at all. d) "Read Eng" mode: for input of books, papers, ect. — purely for data input. This data can become part of a corpus to be specified in a later problem (see a)). e) "Answ. in Eng." mode: A corpus is specified, and a Q. in Eng. is gn. for input. TM tries to answ., and may or may not be gn. a σ for Th. answ —

Actually, I think that I could get a fairly good BSTM w.o. having it do English. i.e. get it able to work on its own improvement — Then give it problems in PSG's w.o.Q ~~[illegible]~~ and non-dim. PSG's.

So — as a first step; Th. TM would be able to:
Give "best" possl. completion to a gn. corpus. (≈MTM).
Give probty for any proposed completion of a gn. corpus, (re other possl. completion). (NMTM)

completion of ~~[illegible]~~ a corpus of hyest expecte
~~[illegible]~~ (for gn. input), output of "

Sun Feb 9, 64
Plans

TMy.

: 704.40: If I want a in-out (operator) type TM for Th. TM, Perhaps for 1: and 2:

1) Give "best" or "most likely" output assoc. with a gn. input or give "top N best" outp
2) " proby of any possl. output " " " "
3) Give output of hyest expected G, for gn. input or give "top N best outputs.
4) " proby of any particular G, for gn. output and gn. input.

SN Th. 3 seems to simplify a previous idea I had on BSTM — i.e. That ~~we~~ he <u>had</u> to have some idea of th. future inputs, in order to make a good estimate of an optimum guess for th. proby of anyG as a funct. of th. (in-out) pair; No such estimate of future inputs is necy., if one has lots of time ~~[illegible]~~ after each problem input is gn. If one wants a ready-made functional form ~~[illegible]~~ (operator) to be used <u>quickly</u> when an input appears, ~~[illegible]~~ - <u>Then</u> it would be advisable ~~for~~ TM to have some idea as to what future inputs were to be.

Note that ~~4) does not~~ if 4) is solved, 3) may <u>not</u> be — since th. no. of possl. outputs may be too large for an exhaustive search to be made.

Similarly, if 2) is solved, 1) may <u>not</u> be.

— So 1, 2, 3 ~~[illegible]~~ and 4 are all <u>related</u>, ~~the~~ tho <u>non-nesting</u> problems — i.e. no one completely solves any other one.

Th. prob. of just <u>what</u> <u>we want</u> in a H.C.TM is discussed at length in J5 a/o BS. I <u>think</u> th. idea is — gn. a set of input objects, with their assoc. G values, HCTM has to either (1) propose a new object of hy expected G or (2) propose a test object, so that knowing its G would get a very hy G object as soon as possl."

3 enables TM to "do experiments" — and there is how this mite become dangerous.

Mon Feb. 10, 64 ~~TM~~ TM &

Plans

: 706.40 : Seems to me, that I have several slitely difrnt. TM types in mind, here.

1) R T.M : trys to give response of hyest "expected" G (or some other funct. of G may be used).

2) HCTM : Given a set of objects with their G's ⓐ to find a new object of max. expected G or ⓑ to propose a new test object, so that knowing its G would help in ⓐ.

Various situations with regard to cost of testing new proposed objects. If cost is very hy, few or no "expts" are done;

" " " " low, many expts. may be done — or not depending on natures of Th. problem.

Also, a somewhat difrnt situation, in which Th. G mechanism is "open" to TM — (so an exhaustive search would be theoretically possl. — Tho/ usually impractical).

Also, Note: I probably don't need a very sophisticated HCTM, to start off. Probly a good enuf goal at first: to have its output be an object that has as hy as possl., a probty of a G > that of any previous known object. So it would try to find object of both hy expected G and large σ_G (with Th. obvious trade-off betw. Th. 2).

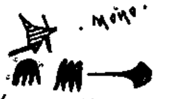
MTM or NMTM (These are predn. TM's — MTM is gn. probs. wi only 1 rite answ. (which simplifies evaln. of proposed codes a bit), and NMTM (non-Math TM) is for ordinary probty probs) — can be used for **MT** or **IR**, with fair effectiveness.

Also, MTM and NMTM can be simulated by ~~any~~ TM that is in operator form — i.e. examples are gn. in which Th. Th. config. with ~~[illegible]~~ Th. "to be predicted symbol" by "□", say — and Th. output, is r "desired" symbol at That pt

Mon Feb 10, 64

TMy

Plans



1. 7.07.40 : Another way to look at This all: Consider all outputs of "hy G", and, gn. an input, ask TM to find an output That is ~~most likely~~ to have "hy G".

A somewhat more refined way to view This: G has, say, ~~only~~ 2 or 3 levels, and we want an ~~top~~ output / most likely to be assoc. with Th. hyest level.

It is clear That even in a very exact sense, we can use a "predicting TM", for RTM work — e.g. say we quantize G into 4 or 5 levels — Than we ask for several outputs that will (most likely) give Th. hyest 2 or 3 levels of G. Then we get Th. G prob. distrib(a) for each of these outputs, and from those, we can select good output(s) trial(s).

Th. impt. Q. is not so much Th. formal ~~β~~ equiv. of RTM and pred. TM, — but Th. Q. is — will Th. formalized, quantified / intuitive ideas That I have for either be directly applicable to Th. other? — I Think So!

E.G. in → RTM, one heur. is to consider all (i,o)'s of hy G, and try to find a lang. for them — that avoids (i,o)'s of low G. In → NMTM, if we want an "output" corresp. to an "input", → Th. pair has hy G, I Think we could use Th. same Gramm. construction technique.

Infact — with a NMTM, — if we only had 2 levels of G ("Good" and "Bad"), we could still make a fairly good RTM, and our heurs. would be very close (when not identical) to those used for a "regular" RTM.

We have, apri Th. "fact" That if a gn. ~~in~~out put is hy G, it is simultly low G. So (I_1, O_2, G_{hy}) and (I_2, O_2, G_{lo}) ar ...mpatible — Tho if we stipulate sharp incompatibility ... may be too disturbed if we make a mistake in ... so let TM "discover" That Th. incompatibil...

Mon Feb 10, 64 TM

Plans

: 708.40 : A trouble with th. 2 levels of G. Say TM has gn us a O_i of fairly hy G, but we want something "better". All we can do then, is ~~mea~~ tell TM he was correct, but that we want another, diffrnt Output of hy G.

One way is to use 3 or 4 levels of G, and never use th. top levell — but th. top level automatically means something better than previous outputs for that same input. This particular situation ("betterment"), will, eventually, have a large no. of previous cases (triplets or quads or more — e.g. 1 input and 1 or more outputs; followed by an output that is "better than" any of th. previous outputs). As soon as

.20 several outputs for a gn. input have been obtained, th. client should try to order th. outputs and give this info to TM. from that info, he can make several examples of "betterment."

If much "hill climbing" is to be done, hvr., .20 suggests that one mite as well let G be a continuous magnitude, since, if we want to ~~as~~ give TM "G order info" about a ~~set~~ of outputs this is equiv to assigning a continuum of G values to them. If we do have continuous G, TM can still use th. same "simple" heurs that were used to get hy G outputs. G had only 2 values.

In a more general sense, I should be able to take any problem type, and explicate/ and quantify my methods of solving it — in a way that could use practically any corpus/ type to help

d costs of various symbols, operators, ect.

is takes us back to 704.01 — i.e. what sort of TM do we want to

-goal, befor ① we ask it to improve itself and befor ②

ord. English.

Tu. Feb 11, 64 TMγ ←

Plans

1 : 7.09.40 : Say self-improvement comes befor understanding English.

If we want this sub-Goal TM to be able to "improve Pgms" — what sort of pgms should it try to "improve" — and for what kind of G ?

.05 — Perhaps a simple MT pgm. or a Q answering pgm. (like Bobrow's) or a PSG d word pgm., or more complex Grammar types or MT learning pgms with fairly complex grammars possl.

A "simple" Q answering TM could do anything! — i.e.

.09 We could ask it to do proofs, computations of integrals, xlations of ss from simplified Eng. to Symb. logical lang, solve "word probs." in math., select Chess moves, etc. Th. requests for each of these types of probs. can shift TM into mainly a single new corpus — but he can use Th. other corpi — or parts of them, that he finds

.18 useful (or coding ~~what they~~ techniques from them).

"Improving Th. pgm" means (a) faster results (b) less mem. used (c) More "accurate" results (less total cost of coding Th. corpus).

We can give TM certain "free" facts. — These amount to "eternal truths" — e.g. like certain symbols can follow certain others — and that every definite integral must have an upper and lower limit, and various definitions of various objects.

We can either give TM these facts in a special format (which, he learns, is always "correct"), or just somehow build these facts into TM's coding system, by giving him "free" symbols at certain pts. — i.e. as if Th. sample had been infinite for these particular facts".

SN) Th. above idea of "free" facts inserted into Th. corp can be made Th. basis of answering various problems involving Th. basis of Math.

In Th. category of "free facts", is Th. choice of Th. original Uni "free facts" can be taken under this category.

Th Feb 13, 64

Plans

.1 : 710.40: Th. idea of 710.05 – .18 seems an o.k. subgoal: A pure predn. T.M., operating in an input-output manner, with many "modes". With each problem a mode index no. ~~symbol~~ would be gn., that would tell what sort of thing was desired – e.g. whether MT, answ. in Eng., evaln. of arith., soln. of diff. eq., proof of thrm., or a "G" maxzn. etc. Sub-corpi with th. same index ~~no~~ no., would be categorized alike and would use ~ coding methods – tho TM could and would use some cross-coupling info betw. sub-corpi.

.10 Actually, using this "mode no.", we could have a special ~~[illegible]~~ ~~[illegible]~~ mode no. when a optzn. of a certain param. was desired.

E.g. N_0 ~~[illegible]~~ Δ Eval ~~[illegible]~~ would be this type of input.

N_0 is th. mode no. of this type of ~~[illegible]~~ input

Δ " a punctuation symbol (it may be a space).

Eval is a string.

Th. nature of this problem is: Given a certain Machine M_1, to find a string (≡ pgm.) X, ∋ $M_1(\text{Eval}^\frown X)$ = a no. which is as large as possl.

Here Eval, th. evaln. mechanism is "open" to TM.

What I'd like more, is simply th. input symbol, N_0. ← index no.

Previously, th. index N_0 was assoc. with ~~#~~ th. strings, X_i, and th. numerical G value G_i — i.e. $\boxed{N_0 \Delta X_i \Delta G_i}$. TM tries to find ~~a#~~ probable X_j Δ G_j ∋ G_j > th. largest previous G_i.

I could simply set G_j to some value > th. largest previous G_i, then give TM th. pure induction problem $N_0 \Delta \square \Delta G_j$ (TM must find ~~[illegible]~~ set of probable contents for "□

→ there is also th. case of partial openness of th. G knows that speed is impt.; ~~[illegible]~~ but th. rest of th. [illegible] explicit, but must be induced. — similarly, Th Feb 13, 64 — TMY

Plans

1: 711.40: is desirable

→ Note: since all "heurs" ~~[illegible]~~ that are re ordering of trials, amoun to a reevaln. of probys. of various trials — then it should be possl. to introduce each such heur. by changing th. interpretations and th. probys. of th. symbols in th. trials, and use ~~a~~ Bern. sub. seqs. for th. trials (or for th. codes of th. trials)

Make more detailed d'crns. of various "Modes" (like those of 710.09–.18 and 711.10–.37 in particular).

Also, try dcrbg. exactly what I do, when I try to optz. a Pgm. with resp. to an "open" or "partly open" or "closed" Gore — then try to express this in a way that is either a ordinary predn. TM prob., or ∋ it is an entirely new mode of operation, but that it uses th. prosts of coding ideas that th. ordin predn. TM used in its ordinary behavior mode..

Now, suppose I had a predn. TM, with an input–output type format. I could do a kind of RTM, since th. hill-climbing, Pgm optimization idea of 711.10–.37 is identical to what an RTM does. — Th. main difference is that in RTM, usually th. G funct. is not "open" to TM, while in th. pgm. optzn., it may be complete open (i.e. G as a funct of speed, mem. used, and "accuracy" (= prost)) Th. idea of 711.35–.37 is probly "good enuf" for much work. When I work out some actual attempted solns., I will also get some good ideas on how to improve this mode of operatio

One of th. main problems will be explication of various of my heur ideas, and putting them in a form that is useable to TM. For this th. simple arith. math TM will be a v.g. "study prob.".

I think that a pure predn. TM would be fairly adequate fo any sort of H.C. TM or RTM prob. that mite arise — sinc TM can be el'zd with predn. being th. crit

At very worst, any prob. can be solved by asking ld I do next"?

F. Feb 14, 64 TMS

Plang

01: 712.40 : So: First I want to make a rough outline of th. various stages of TM developmnt leading to a fairly usefule device. Then fill ~~[illegible]~~ in more and more details.

Anyway, at present my impression is that a thoroly adequate subgoal would be th. creation of a pure predn. TM., having both discrete and continuous "input" and "output". Untill I find a better way to use it as a RTM, ~~use~~ (≡ hill climber), I can use th. technique of 711.10–.37 : Hvr. we can use $N_1 \Delta I \Delta$ ~~□~~ ΔG as "input to th. TM" ~~[illegible]~~ (N_1 is th. mode No.; I is an input string that asks a Q; G is a G value higher than any ever gn. befor for any $N_1 \Delta I \Delta$ ~~[illegible]~~ $\phi \Delta$ ~~[illegible]~~ [redacted] : ϕ is an output string ~~[illegible]~~ that attempts to answer I; Δ is punctuation

DEF Prediction TM
PTM : This is about th. same as Dart TM., hvr., PTM will try to have ① outputs of hy proby ② perhaps be able to give approxt. values of some hy P (or even low P) outputs ③ Can deal with both continuous and discrete data.

A perhaps impt. pt. : One method of "finding" codes for a gn. corpus, is to simply try codes at random, as input to Turmac, untill one hits correct ones. More sophisticated, is to count symbols in th. corpus and use th.) simple Barn. code. This, clearly, is far faster, and is ∴ a "heuristic." Now — can I characterize all ~~[illegible]~~ (or almost all types of) ~~[illegible]~~ heurs. as consisting of a Barn code on certain symbols — (but with th. signif. of th. symbols considerably modified)? ....

Th. construction of Barn. codes seems to differ profoundly fr[om] Monte Carlo construction of codes to solve arith. evaln., outlined in ZTB 143

Perhaps th. commonost kind of heur. is one that ()manteses on th. corpus and reorders th. trials

F Feb 14, 64 TMJ

Plans

Y. Bar-All
~~J.Y. Bar-Pierce~~
J. R. Bubblepiercer

01:713.40: with ~~a~~ Rat info. This is in th. spirit of th. operator-observer-algebra machine discussed some time ago. Ops. and obs are combined to make new obs. and ops., and th. results of th. obs control what is ~~to~~ to be done next — (i.e. ob, or op). Some sort of score was kept, as a sort of correln. matrix.

Present state of problem: I have several techniques of induction that, presumably, I can dcrb. in th. ob-op formalism. They are

① Bern seq.

② ZTB 141

③ ~~a~~ Constrn. of PSG's for induction.

④ Use of subsn. rule for induction (w. or w.o. ~~rules~~ — or ~~a~~ w. or w.o. temporal ordering of rules.

⑤

On th. other hand, it isn't clear as to what relationship they have to my normal, "intuitive" methods of induction..

SN) methods 1, 2, 3, 4 could perhaps be implemented by "2gm recognizers" and "2gm recognizrs" of hyer orders — so they all mite be fairly ~.

"Another" idea ~~now~~: To write a tng. seq.; then write th. solutions to th. probs.; then try to express th. solns. in a lng. using a small set of "words". Then see ~~whether~~ whether these "solns." are "reasonable hypoths." at each pt. in th. tng. seq. If not, ~~insert~~ interpolators into th. tng. seq. to reduce th. "jump size".

Hvr: For any tng. seq. and any set of solns, if a "complete" lng. to try to find solns., it will b

estimator of th. "jump size" be

M Feb 17, 64 TMg

Plans

1: 714.40 probs. in th. trng. seq.

One could just start with <a set of diff. probs. that one would want a "fairly brite" TM to work> with their solns. Then write a trng. seq. leading up to them, with not-too-large jumps betwn. successive probs.

SN] One common "search" method: One has a bunch of possl. regularities that one is testing, to see if ~~they~~ any apply to th. corpus, with suff. freq. to warrant their costs of defn.. Instead of testing them all, one observes, e.g., that th. seq. "ab" just occured. One then sets up a "ab" detector, and if "ab" occures again within a certain time, one takes it more seriously. If it doesn't occure after another certain time, one drops th. "ab" detector and uses th. same hardware to construct a detector for some other ~~z~~ zgm that has just occured.

→ ||| In general, I think I can express all T.M. probs. as "search" probs. For pure induction probs., this amounts to finding regularities that have a dcrn. cost << th. dcrn cost of a direct coding, which uses, say, th. Bern. code method.

For H.C. probs., we first want a soln. to th. prob. of finding G as a funct. of (I, O.) — this part is considered to be solved when one has found such a funct. that fits previous data and has a total dcrn. cost < that of some "direct coding" method. After a G funct. is found, one can then try to find a O ∋ G(I_i, O) is max. (I_i being th. present input).

SN There is some Q. abot. th. rôle of hours, in th. of 714.30 ff: Do hours simply have th. effect of reordering tria t they can be equiv. to modifying th. apripd? ol, hvr., hours involve more than reordering on th. new apripd. They also reorder on

Plan

1: 715.40: "ease" of making a trial — ("ease" usually means "speed — tho it may sometimes involve amt. of memory used).

Th. idea of ~~th~~ any heur, is usually to accumulate as much as possl. aprip / — in terms of Th. total aprip of Th. ~~[illegible]~~ objects tried — in a gn. amt. of time. (with passing respect to total mem. used).

This may not be entirely correct. — remember, one is looking not only for a code that fits, but one wants also, a code that fits that is of as hy. pcost. as possl.

In searching, retain Th. "code nos." or some sort of abreviated terms of Th. 10, say, best codes thus far — along with Th. pcost of each. After each trial, see if Th. pcost of that trial is > Th. pcost of Th. worst trial retained thus far. Using Th. 10 best codes for predn., will give a better predn. than just using Th. best code. — Also, ~~[illegible]~~ knowledge of these alternate codes will be useful when "backtracking"

What I can do: Make a big list of math. probs. that I could get TM to work: e.g. Arith eval.: Solving ~~=~~ N equs. in N unks. — first linear, then quadratic — then cubic, quartic .. proofs of theorems; soln. of Trig. identities; formal integrat (like Slagel).

In any of these probs., if Th. method for working any one of these probs. is suffly "close" to ~~th~~ Th. methods for previous probs, ~~then~~ (i.e. say < 10^6 trials necy) then we can say that TM has made a "signif. induction" ~~[illegible]~~

Hvr, ~~[illegible]~~, if ~~th~~ only ≤ 10^6 trials are necy, perhaps any hu Th. induction "trivial"! — Or, maybe it will o a human if < 100 trials are neede

Tu Feb 18, 64

Plan

716.40 is certainly of very much interest! It would seem, that if I do incorperate all of Th. human hours into TM at a pu. pt., that Th. "° of supprize" at TM's induction should be an ↑ function of Th. rcost of Th. induction only! Then Th. Q is — how big must Th. rcost be for us to feel a fair amt. of "creativity" has taken place?

Lets go back abit: Consider formal itegration: its like Thrm. proving: i.e. One has a set of permissable xfmns, and one has to make a string of them ∋ Th. result satisfies a certain criterion (i.e. Th. permissable xfmns are alg. equivs. and Th. criterion is, that Th. resultant function is in a list of functs. ∋ one knows how to integrate these functs.

In Thrm. proving, Th. "list of xfms" involves Th. post set and thrm. set used. Th. "criterion" is that Th. result be Th. Thrm. one has to prove.

Also in Integ, one can make a guess and see if Th. derivative is Th. expressn. to be integrated, but we can skip this type of heuris. and see how far we can get.

Anyway, its not clear to me as to how these types of probs. can be expressed as running thru a list of trials — each trial being of known aprip:

~~Integn. seems more~~ Integn. and thrm. proving seem more like ⓐ Appling and ob. to some object ⓑ then on Th. basis of what Th. ob (servation) results in — deciding what op. to apply to what object(s).

Well... Yes, but: Say one of these trials

Apply Ob, if Th. result is "a", apply Op₂ if Th. criterion is satisfied.

To Feb 18, 64 TMg.

Plan

1: 717.40 : We can write this as $Op_1(X) \equiv Op_{(Ob_1(x))}(x)$ expressn. to be integrated

(i.e. $Ob_1(x)$ determines Th. subscript of Op).

(Note that if $Ob_i(x)$ can $=1$, we have a recursive, LISP - like defn., that ~~may or may~~ will not converge)

Anyway, we have, as our method of formal integration a bunch of operators like Op_1 and we try them in succession until one of Th. results satisfies Th. criterion".

(Th. problem for TM, is to invent operators like Op_1, using some ~~[illegible]~~ basic vocabulary. These ops can be invented as TM continues along Th. Trng. seq.

SN This Op_1 defn. can be defined in fortran w.o. any trouble : say Ob_1 always has an integer ≤ 5 as output.

(woops — if $Ob_1(x) = 1$, Th. defn. cannot converge! for $Ob_1(x)=1$, Th. defn. reduces to $Op_1(x) = Op_1(x)$.)

Anyway, I should look at integn. and Threm proving (say look at Slagel and Simon - Newell) too see if I can fit these methods into my scheme.

My, apri. impression is that both are pretty much like ~~my~~ one's intuitive methods of doing arith. evaln., equ. soln., etc. — i.e. one looks at Th. state of affairs with certain obs., ~~[illegible]~~ Th. results control one's ops. — one then uses new obs, ~~etc~~ to control new ops, etc. Th. routine, itself is fixed, w.o. any "trial", in Th. simple sense. Th. construction of Th. over-all operator, hvr., proceeds by trial and to a large extent. — ~~[illegible]~~

For integration (and probly many other types of probs) — say an op that solves all probs. in Th. trng seq. up we get a new prob. where Th. trng. seq — what to do? well, find out

Tu Feb 18, 64 T.M.Y.

Plan.

01:718.40 : prob. differs from th. old set. Get an ob. that ~~[illegible]~~ can pick out th. new type of prob. (along, possibly, with some of th. old), and devise a new method for this new type — probably th. new method's str. will be related to th. nature of th. new Ob — i.e. th. nature of th. difference betw. th. new prob. and th. old ones.

Note th. nature of th. entire prob.: At each pt. in th. tng seq. we want an op. that is of min. cost, that works "th. entire" " . To do this, we ~~usu~~ usually take an op. that works most of th. seq., and modify it slitely to work some new probs. Often this will work, but sometimes we will have to "backtrack."

At any rate, th. dern. of our "total op" at any pt., will not be very big — so it wouldn't have many "statistics" in it to help control ~~new~~ choices of new modifications. Th. choice of new modifications will have to be made on th. basis of ① my reasoning ② Statistics of th. ops. for th. soln. of other probs. from Arith. evaln., thrm. proving, eq. solving, etc.

38
200
26.3

Note that this "pooling" of statistics proceeds very naturally when we use th. "Mode No." idea of 711.01~~#~~f. We have a ob. that recognizes what mode is necy., and shifts th. machine to th. proper sub ~~[illegible]~~ operator. There is a tree-like str. to these operators, so th. PSG notation will probly be used for cost evaln. — I suspect th. universal method will have to be used; since th. ~~[illegible]~~ old method probly gives wrong results (i.e. ~~[illegible]~~ P_{AB} must be > ϵP_A before $\widehat{AB}$ can be defined) — (The perhaps it doesn't m much difference, since th. samples will usually be all.].

At any rate, this ob-op idea is very clos tively, so it should make explic

Tu Feb 18, ~~64~~
Plan

TMJ

01: 719.40: intuition easier.

~~No trouble with th. PSG notation: sub-trees do not necly~~

A possl. trouble with th. PSG notation: If 2 sub-trees are identical, they will have corresPg. sub-seqs in th. dern. — but if one sees 2 identical sub-seqs. in th. dern, they don't necly corresp. to identical sub-trees!

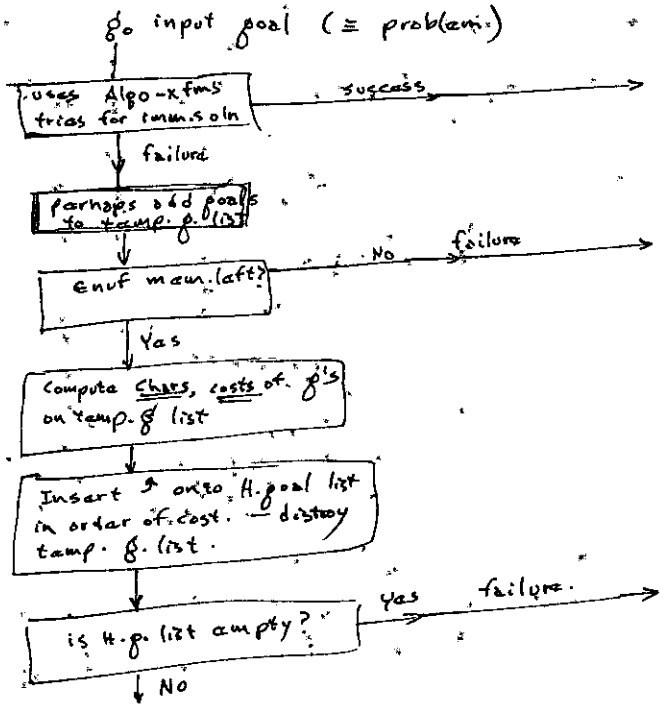
Actually, th. integers in th. dern. are just "choice" indications and coincidence of 2 such sub-seqs. in a dern can be a "heuristic indication" of identity of sub-trees, but little more — i.e. it would save time in [illegible] in finding identical sub-tree

W Feb 19, 64

TMS

Slagle's Symbolic Int. pgm. ("SAINT")
In Feigenbaum and Feldman, p. 191 ff.

What I want is a good dcrn. of the method, so I can consider the prob. of "learning" it + using a suitable trg. seq.


g₀ input goal (≡ problem)
uses Algo-xfms
tries for imm. soln
success
failure
perhaps add goals to temp. g. list
No
failure
enuf mem. left?
Yes
Compute chars, costs of g's on temp. g list
Insert onto H. goal list in order of cost. — destroy temp. g. list.
Yes
failure.
is H. g. list empty?
No

Th. Feb 20, 64 TM

Slagle : "SAINT"

01:721.40 : Th. Slagle paper isn't clear on a few pts. — so I'll outline a possl. method ~~[illegible]~~ or methods, and see if some statements he makes can decide b[etween] them.

My impression is ~~that~~ ~~a~~ Jim's pgm. is pretty much th. folg:

Take th. input expression: Apply th. A.T.'s $\overset{\text{Algor. Xfms.}}{}$ as far as possl., then lookup in "Standard form list". If they are all on list, prob. is solved. If not, put "unlisted" ones on "Temp. Goal" list, and get their "characters", then put T.G.'s / with their chars on ~~Th. Goal~~ Heur. Goal list. ~~[illegible]~~

Look at goal tree, and th. expected costs of various goals, then decide which is best to work on. (This should, ideally, involve analysis of chars. of each ~~H~~ goal — since goals with certain chars. have appropriate Heur Xfms (H.T's) — that are more effective than other ~~H~~ H.T.'s.)

.20 This whole problem of what to do next — costs of determining various

.21 characters, etc., is a standard, tho diff/t. problem. — Assume one has statistics on th. effectivenesses of each of th. H.T.'s with resp. to each char. config.

.30 So th. Big Q's ~~are~~ are inventing:

1. Heur. Xfms.
2. Algorithmic xfms.
3. Characters.

.32

it may be that (2) are ~~[illegible]~~ finite ~~&~~ and small in no.. Jim used only 8 of them.

Various decisions / on what to do next will be made on th. basis of expected use of mem. and of time. We can (as Jim did) assume fixed total mem. avail/l. and then try to min. Σ time.

If done in th. above way, using Jim's 1, 2, 3, th. [resu]ltant pgm. should be more economical than Jim's, and [...] in some sense — i.e., fewer wrong leads, few[er]

[...]ch as th. prob. of .20—.21 will b[e]

Th ~~Feb 20, 64~~ TMγ

1:722.40: no: (if not all) TM. probs., I can try to work out a soln. by hand of my own.

Various configs. of chars. can be correlated with Th. H.T.'s that are best for them. Th. "clumping" of chars. is something I probly. solved in my "IR" work.

Slag's pgm. probably employs various "ruff and dirty," ~~soln~~ "cavalier" solns. to Th. probs. of 722.20-.21.

It would be well to look into Simon and Newell's GPS (Gen. prob. Solver), and see if Slagle's, Th. L.T., Gelernters, and other work, can all be put into Th. GPS form.

Hvr. even just Th. integrator has very basic, general probs. — e.g. How to ~~[illegible]~~ devr. Th. Xfms and Chars. of 722.30-.32?

I could start Th. TM off with some of them, along with a "complete" logo: — to ~~[illegible]~~ eval. Th. posts. of each symbol and use these posts

20 to invent new trials. An impt. Q here, is whether Th. xfms and chars actually used, are ~~[illegible]~~ of a form ∋ random combin could find them — or, is actual logical induction involving

4 "understanding" necy.?

So, perhaps stick with integration for a while, to look into .20 to .24 with a bit of quant. analy. Looking at Th. LISP expressns for some of these (722.30-.32) objects may help.

use of α, β Heur.

Th. work on Checkers (Samuels) and Kalah (Silver, McCarthy) May be of interest. See if I can get them into Th. same form as Slagle and GPS. Note that Samuels, by putting a li ~~[illegible]~~ ple learning into Th. rite place, was able to get a fairly cl

Th. Feb 20, 64   TMJ

01:723.40: It mite be expedient to adopt Slagle's or some other "cruder" soln: to th. probs. of 722.20-.21, and optimize th. xfmns of 722.30-.32 within this fixed, (non-optimum) framework. If th. xfmns have enuf "scope of applicn", they can undo any badness in th. "framework".

Note that I ~~am~~ now thinking of a RTM, in which th. G funct. is fairly "open" to TM — it is even partially elementalized. Th. RTM optzn. can proceed at a fairly rudementary level, by simply selecting out "entities" (= xfms, or chars) that are "reasonable", on th. basis of a simple ZTB 141 (Ngm) grammer or a uniparse PSG.

This addition of "learning" capability can ↑ th. power of th. machine very much — but only if th. "entities" that would improve operation are accessable in a reasonable no. of trials — i.e. th. inductive jump is not too large.

Symb. integn., Thrm. proving, alg. eq. solving, proving trig. identities, Geom. proving and dif. eq. solving can all be put in a form for GPS to solve, in terms of goals, sub-goals, chars. and heurs.

I mite be able to express th. PSG discy. prob. in these terms i.e. a set of xfmns, etc.

⊙→ From these probs., I can get an idea as to th. no. of trials needed to devr. various heurs., and what sort of Tng. seqns. are needed. Of much impt. here, is th. Q. of whether TM will actually have to "understand" th. reasons behind th. heurs before inventing them. If not, fine. If "understanding" is necy, I think this mite be equiv. to ~~th~~ a hyer order TM.

~~Th~~ Also of import.: Could such a TM improve its o-tion appreciably — also is such improvement ne- a TM str. can be devised so that se'

[illegible]

Th Feb 20, 64 — TM ~~γ~~

01: 724.90: of ~~=~~ TM" .

Ⓒ→ *Also of* much impt. — could such a TM understand and *read books in English* ?

In some ~~ea~~ earlier work on Ths (BS, or JS perhaps), I considered *me* as TM_2 / or TM_3. This involved looking at th. set of "successful" heurs, and trying to find general statistical rules for forming them.

So we have TM_1. He simply executes a pgm. to solve equs. prove thrms, etc. ~~[illegible]~~ Some of his params at any time are his set of heurs. These are, ~~t~~ in th. simple case, heur. xfms, algo. xfms and Chars. TM_2 looks at th. set of heurs., and their effectiveness, and proposes new heurs ~~[illegible]~~ a/o discards old ones, in attempts to ~~t~~ improve TM_1. ~~[illegible]~~

TM_2's particular mode of looking for regularities in th. set of ~~e~~ successful heurs is, at first, fixed — probly a Bern seq., upm seq. of PSG type gramm. — But TM_3 (≡ *me* at first) looks for other regularities in th. good heurs and tries any other means of improving *TM_2*.

To get TM_1 to do TM_3 work: First I have to be TM_3 myself for a while, to find out just what sorts of abss. are used. *Then*, in TM_3 (and perhaps in TM_2), one can't afford to be very profligate with one's "Trials", since they cost a lot — so one will spend a *lot* of time looking for heurs that have, for various "logical reasons" *very* hy probty of being "accepts or which have a very hy probty. of giving an ↑ in ~~[illegible]~~ TM_2.

Note, hvr, that in *many*, if not *most* cases, one doesn't *ntire run* to decide on th. probable effect

Th. new heur will be *only* occ

Th Feb 20, 64 TMS

Plans

01: 725.90 : applicable, and so one will only have to run those cases in ~~of~~ th. run. Also, one mite. e.g. use only those cases (to start off) in which th. old ~~[illegible]~~ heurs were either failures, or took too much time or mem.. In genl., thr. will be various tricks to [redacted] reduce time spent on confirming efficacy of various trial heurs.

At th. present time, I am torn betw. 2 possl. interm. goals

① Getting a fairly good T.M. to ~~be able~~ to "understand" Eng. (i.e. reply in Eng. to Eng. Q's – using Eng. data)

② Getting a TM that can signifly improve itself.

What I may need at present, is a more complete dscn. of a final (or "more final") T.M.

One thing that I'd [redacted] like to be able to do with TM: Give him a graded seq. of math texts, starting with simple algebra, and work up to very complex math. Have him work th. problems in th. texts – some of which can be very "creative". One trouble is that ~~both~~ elementary math texts draw much material from th. "real world", and this mite give TM some troubl. It would be interesting to see just which probs he has trouble with.

What I want, now, is an over-all view of th. expected futur course of this TM work. I think that I know enuf to be able to draw up such an outline.

Also, I would sort of like, as soon, as possl., some examp working what look like "genuinely creative" problems.

There is some Q. as to whether I can really expect T ad understand Eng. text on a vast variety of proper backgnd." E.g. when I read

Plans

.01 : 726.40 : on art. Int. e.g., I usually have to translate it, into my own set of concepts, befor I can "understand" it. Anything less than that is partly "rote memorization" and is of little ~~and~~ value in induction — So, befor understanding a text on a subject, TM must, in some sense, have a "conceptual vocab." for that subject. If he hasn't, he'll have to develope such a vocab., and this would be a very large conceptual jump, if he is to get a "useful" Vocab. of this sort.

What I want, is a general "format", or pgm. form, ∋ I can easily change problem types, insert new heurs, etc.

① 2 possl. formats: ⓐ Prodn.; ~~⇒~~ (I, O) pairs.

ⓑ GPS - try to find seq. of xfmns ∋ result satisfies criterion. — or, more generally, to find a string that satisfys certain criterion (i.e. th. "search" prob.).

.25 ② How M.T. and Q. answrg. and "Understanding" reading can be put into th. form of ⓐ:

ⓐ "Understanding" reading: This is MT from th. input into symb. logic lang. — i.e. into a set of strings that states all th. "info." in th. input. ~~2 a~~ The. sugs. for learning this is by example.

ⓑ Q. answering: MT of a Q into a criterion for deciding if a gn. statement is an answ. and a criterion for an answ. within th. symb. logic lang. of. Th. prob. of how to use info in th. symb. lang. for answ. th. Q is ~~of~~ a type ⓑ (GPS) problem.

ⓒ MT: Involves xltn from Russ. (say) into sym

xltn. from symb. lang. into Eng.. This metho

close to one used by humans.

"discourse analysis"

Tu Feb 25, 64 TMJ

9.25

Plan

1: 727.40: >1 sentence}.

I am fairly sure that induction must be used in any good soln. to GPS or Th. "search" prob — also, I think that any "search" can be expressed in a useful way as an induction problem. E.g. in 727.35 we have to make a search for solns. within Th. symb. logic. To do This inductively, we can present Th. fact that (only strings satisfying Th. criterion) are acceptable as a "free fact" — so TM. has zero cost for it. Then other heurs should be expressable as coding methods for TM₂ or TM₃.

Note that search probs are, in general, h.c. probs, and are not "simple" first order induction probs — but TM₂ or TM₃ type probs. —— Tho I could make all search probs into first order ind. probs. by including Th. search time and mem. used along with Th. thing found, as part of Th. total soln. Then I ask TM for a string that satisfies Th. criterion, and has small soln. time. (Tho there is some recursivity here that may give some Q. as to whether This is meaningful! — i.e. "soln." time includes time spent thinking about "soln. time.")

~~Whether~~ If I want to use Th. "prediction form" for TM I'll have to look at a search prob. see if I can get Th. ordinary heurs that Sim. and New. use, into Th. proper form as coding methods. Also, I must see if its possl. to use C.M.ii to generalz. These heurs (or any oth in an easy "natural" way.

One Genlzd. form of TM That would seem to me int. in: Th. "Mode no." TM of 711.01f:

Th. "H.c." mode; I just use any H.c. techneqs I can th con. to clear up any quant. difty

Tu Feb 25, 64 TMy

Plan

.01: 728.40: In Th. case of search probs (— GPS — symb. integn., Thrm proving, again, just use a proper mode no., and ~~setu~~ write out an explication of my own soln. methods.

(I could even have a chess a/o checkers mode !)

In all but Th. H.C. mode, Th. goal is fairly clear — one groups together all (probs and solns.) of that mode — and uses some cross mode coupling when expedient.

In H.C. mode, I usually use an approxt. soln. (like 711.10-.40), but if trouble arises, I use a more exact soln. ~~than~~ (e.g. one that tries to maximize Th. probty of Th. next trial having a G > that of any previous trial [redacted] — or some such goal(..)

DEF : Th. big advantage of this PolyModal TM (PMTM), is that I can work any kind of TM problem any way I like, and make this operation part of PMTM. E.G., I can use all of Slagle's and Sim. New's heurs in rather direct form..

Note: 727.25-728.01 have a very good idea toward Th. etzn. of "learning to read".

At first, I will want TM to be able to answer English-text math probs. in Eng.

Th. next step — for Physics probs (or chem. probs.) will be much more diff't..

.5 SN) Note that all predn. probs can be expressd as search probs. — in Th. form of finding a code a/oE predn. methods that give min b cost to Th. corpus. Th. advantage of exp. ..n. probs. in this form is that I can then use Sim. an. pfm. for most of Th. work.

TMg

W Feb 26, 64

Plan

.01: 729.40: Some "kinds" of probs. That could corresp. to diffrnt mod of PMTM: [Also see: "Kinds of Lngs." near Early "Plans" for many examples.]

1) chess

2) checkers

3) Numerical T.S. predn. — each method of predn. could use a difnt. mode.

4) Eng → symb. logic MT

5) symb. log. → Eng MT

6) Letter, no. recogn, ~~[illegible]~~ patt. discy. — for optical pick-up of Eng text.

7) ZTB 128 predn. (Arith TM ≡ MTM).

8) Symb. integration

9) Thrm. proving { logic
geometry ← (note new heur. of use of diagram).

10) ~~E~~ Trig. Identities

11) Simult. eq. (or just 1 equ.) solving.

12) I.R. (B. "clumping" prob).

13) digital T.S. predn. ⓐ Bern Seq.
ⓑ ZTB 1.41
ⓒ PSG's for predn.

14) Devrg. new heurs ≡ th. TM₂ prob.

15) R.W. probs in physical manipulation. Taking things appart and putting them together. washing dishes. Soldering ckts. Setting up and performing physics/ and chem. expts. These probs. involve sensory f.B., so I must be careful not to let things get "out of hand" — i.e., be su That TM doesn't find any ways to get more direct f.B., wo. doing what ~~I~~ I want him to do.

16) Optical pickup of pictures, and R.W. info. — Graphs for Physics texts.

~~[illegible]~~ Organic Chemical Synthesis. TM's ou → spec 734.01 output can be merely ~~[illegible]~~ a seq. of instructions on how to [illegible] a ~~technician~~

[illegible]

W Feb 26, 64 TMf

Plan

.01:730.40:

DEF QI: (Question of th. Instant)

So, I'd like to make a PMTM, because I could then work on any type of prob., and use any approach I liked (practical and yet all that work would be ~~to~~ useable for any important modality of th. TM that I wish to use it for.

So: ① I'd like to make a PMTM:

But: Th. search prob. is genzable to include all predn. probs. (728.35), and th. genlzd search prob. has already been formalized by Sim. + New's GPS — so I could use their work and build on it.

So: ② It would seem that there are advantages to emphasizing th. "search" aspects of a TM (which PMT would not do).

.25 AH! A possy.! Let all search probs. that occur in $PMTM_1$, be handled by TM_2! e.g., Say $PMTM_1$ is gn. th. prob. of proving ~~with~~ Thm., Thm_1 by making a seq. of transfm from a gn. set of posts. PMTM "knows" what th. prob. is (having been gn. this info "free" (i.e. $PMTM_1$ was gn. th. critn for soln.). An exhaustive search could, in theory solve so TM_1 ~~~~ has (usually — since th. search may not converge) a ~~~~ brief code for th. output in terms of th. input, for all proof generation probs.

Th. only diffy, is speeding up th. search, and speed and mem. economy are probs. for TM_2, relegate almost all of this prob. to TM_2.

O.k.; so, then: What is there for TM_1 to

are expressable as "searches"

Wfeb 26, 64 TMY

Plan

.01 : 731.40 : well, it is a "heur. trick" to use certain readymade search methods for TM, — i.e. carbon modes in which a reasonabl soln. to th. search prob. has already been obtained — tho this soln., of course, is probly not optimum by any means and TM can certainly still improve it.

SN / Research method : At most times in my TM work, I should try to have at least 1 or 2 QI's (Q's of th. instant) that haven't been solved yet, and whose soln(s). appears to be a bottleneck to progress.

.18 QI : ① Is PMTM with th. trick of 731.25 adequate ≷, so I can work on a " w.o. difty ?

.20 ② If PMTM is O.k., what modes and sets of probs. should I start on ? Do I want to emphasize "learning to work in Eng. a/o self-improvement capability ?

.23 ③ Just what do I want ~~to~~ my final TM to be able to do — (what will I use it for at first?) and how specific must I be at th. present time?

Note; th. above 3 QI's are heirarchical — 1 is contingent on 2 is contingent on 3. (not exactly - but O.k

To solve ① (.18) : Look at various probs. of 730. and see if th. proposed methods are adequate. Some prob. types to consider ⓐ Eval. of Ar.th. expressns. in which (+, A, B) is used to designate (A+B), and enuf parens. are used thruout. [(X, A, B) ≡ (A×B) etc.]. (It's clear to me as to just what th. prob. is here")

ⓑ Some "pure induction" probs : IR (th. "clum

we are gn. th. data (corpus) of hand-categorized do

gn. a set of abss., and means for combining thos

Sat Feb 28, 64 TMJ

Plan

01:732:40! (c) ZTB 141 : similar to IR, in th. sense that new trial abss. are formed in a very simple way from old successful abss.

(d) Say I got a successful h.c. method for unpars PSG. This would be a simple pure induction method.

SN) Is it true that all searches ~~[illegible]~~ for ~~[illegible]~~ a good inductive code can be put in th. form of simply trying simple binary combs. of old successful abss? — Then ternary combs, etc.

I.e. Say we have pcost P_i for $(i = 1/m)$ ..; m abss. Then we try combining these symbols in order of aprioP, using first that n tuple of symbols of hyest pcost; then that n tuple of next hyest, etc. — Tarrytown

Note : In (b), (c), (d) above, at all times, a code for th. corpus is available — Th. pgm. merely consists of a pgm to search for ~~more~~ better codes!

This is in contrast to th. "search" prob., in which one hasn't yet found what one is looking for. An ~~ex~~ "exhaustive" search is usually theoretically possl. and would usually be O.K. but takes too long. So TM_2 (as TM_1 does in b, c, d) must try to improve on th. search technique — usually starting off with something much better than "exhaustive" search.

Wrong! In ZTB 141, there is only 1 code

→ SN) In ZTB 141, one could do better, by retaining, at all times, th. 10 best codes, and using each of these as a pt. for th. next "opt. step" — then, of th. 20 new codes, ~~sta~~ retain only th. "top 10" again. (A "code" corresps to an set of 2.pm. defs.