Definition of t. T.S. of 53.25

.01: (132.40 spec) : The t. T.S. is → [redacted] There exists a seq. of Gops, Gop_i, so that 1) Gop_i can work from $(Q_0, A_0) \cdots$ thru (Q_i, A_i).

2) $\forall$ [redacted] i, Gop_{i+1} can be obtaind from Gop_i by using additional "information" pc_{i+1}. Also the cc (computing cost: usually in dollars) needed for Gop_{i+1} to xfm Q_{i+1} into A_{i+1} is cc_{i+1}.

.09 3) $\frac{cc_i}{pc_i} < A$; where A is a reasonable cc achievable with an available computer system. Hvr., more exactly, →

(A is t. amt of Money one is willing to spend on a soln.)

Usually, $Gop_i = Gop_{i+1}$: i.e. The same Gop is used to solve a sequence of problems.

→ 4) say N is t. average no. of probs solvd by a gu. Gop_i befor it fails & a new Gop must be found. ~~Then I think we want~~ [redacted] A (.09)

$\forall_i \; \frac{cc_i}{pc_i} < NA$ (so on t. average using our soln. algm, it takes $\sim cc = \frac{A}{N}$ to solve each problem (if searching for a soln. takes much more cc than solving probs w. t. correct Algm.)

5) It may well be that there are other seqns. of Gops that can solve t. T.S., but if t. T.S. is to be "acceptable" there must be at least one seq. of Gop_i's that satisfy conditions 1), 2), 3), 4). (also 6(a) is vital).

6) Suppose that $[Gop_j]$ is a seq. of Gops that can solve t. T.S. in t. manner of 1) ··· 4). Let [redacted] $[N_j]$ be a seq. of integers that tells how many probs are solved by each new Gop befor it fails. (so $N = \overline{N_j}$) [redacted]

(t. range of j is usually << that of i)

We want each N_j to be large enuf, so that when t. corresponding

.32 (a) Gop has solved N_j problems, there are NO [redacted] other Gops

.33 w. $\frac{cc_i}{pc} < NA$ that have solved $> N_j$ problems,

.34 (b) & there are few other Gops like this that can solve [redacted] N_j problems.

(This results in fewer Gop's having to be stord in Memory: modified for trials on new problems)

This is important if (more soln. algorithms) we are to retain only the Gops that [redacted] have t. largest N [redacted] value.

6)(a) Is ~~one~~ essential feature of t. Algm.

6)(b) is a softer constraint. It gives a quantitative tie to cc of soln. to t. extent that it's violated (perhaps easy to compute)

If cond: (if 6) is violated we have a T.S. of type 53.35

5.16.82 TS **The Problem**: To find a Gop w. max pc., that can work all of t. probs in t. T.S. (grand operator)

.01 : An Algm. to work t. T.S. of 53.25 (≡ 134.01—.40). (i.e. to find this Gop).

.02 We start out with a Gop$_0$ [redacted]

[redacted] à we try various modifns of it [redacted]. These modifns have various pcosts of dcrn. — say pc$_k$ for t. kth possl. modifn.

.09 A set of (trial) Gops; Gop$_{0,k}$ is produced, à these are tried on Q$_0$.

.10 We allow a cc of up to A · pc$_k$ for the creation of Gop$_{0,k}$ and its soln. of Q$_0$ → A$_0$. If it takes more cc, or if it gets t. wrong A$_0$, then that Gop is a failure, à we go on to test t. next Gop.

Since $\sum_k pc_k < 1$; t. total cc spent on all Reser trials is $\sum_a pc_k \cdot A < A$.

We then take all t. Gops that solvd Q$_0$ à see how many subsequent problems each can solve — allowing ≤ pc$_k$ A for each problem. [pc$_k$A is certainly ≥ t. cc needed to solve t. first probln t. subsequence: Any good Reasons for allowing ≤ pc$_k$A?]

.23 Say N$_j$ is t. most problems any one of t. Gops can solve before it fails. We then retain all Gops in memy that can solve N$_j$ probs. — say there are R such Gops.

.25 Say Q$_s$ is t. first (same for all Gops) Q for which all of these Gops has faild.

For each of the R Gops, we try modifns like .02 f.f. ....

(essentially, loop to .02 [redacted]. but before that, Q$_s$ ← Q$_0$, A$_s$ ← A$_0$)

.24 {Total cc needed is ≤ RA

· If we run up against a Q,A that we can't solve even w. new Gop trials, then **A ← 2A** and loop to .09 }

Diffys of t. .01 ff algm:

If we assume that t. T.S. is of type 53.25 (134.01—.40)

Then there are 2 major diffys:

.32 1) The number of (redundant) Gops retained in memy tends to expand exponentially w. t. no. of new concepts needed. (107.38, 121.25 and 121.30) 2 distinct (bad) effects.

.35 2) When a bunch of Gops are found (as in .23—.25) that can all work t. next N$_j$ probs, we must also test these Gops on all past problems. The cc involved is ∝ c^2 (where c is t. no. of new concepts (≡ new Gops) needed for t. T.S.) (107.07, 106.01—.34). Anything much more rapid than cc being linear in c is probly unacceptable

Another ~~problem~~ kind of activity that I haven't written much about recently ~~is~~ that fits well into the task net formalism is "planning". Its a special way of breaking a problem into sub-goals. It xfms a problem into a (sub) task net.

On the gen'l concept of "planning" I don't have many good ideas. Newell may have something to say; Miller Galanter & Pribram ← (Ibid (MGP) pp 177-194 maybe relevant)

.10 A rather general method of making plans is by Analogy. Problem A has been solved. Problem X hasn't yet been solved. Prob A is analogous to X, in the sense that there is a xfmn that transforms X to A. There are elements in the soln. of Problem A that have corresp'g. elements in the domain of X. Those elements in X's space are used as tentative sub-goals, to see if solving them would, indeed, solve X. If they would, then this set of goals in X's

.20 space would constitute a "PLAN".

The foregoing dcrn. of the use of Analogy to devise plans is very general. — by itself, it is too general & it needs the guidance obtained by experience to tell which kinds of analogy are good in this area, & which ones can be used to plan what kinds of problems.

→ I suspect that most plans can be put in the form dcrb'd in .10 - .20 (I'm not so sure, hm.!)

Newell had the idea of a "Plan space" as being a space in which the problems to be solved were somehow simplified (say by leaving out certain parts of their dcrns) — then by solving the (thus simplified) problem in the plan space, one could guide the corresp'g path of work in the true problem space. This view is a sub-class of the more general Analogy method of .10 - .20.

→ More generally, a "Plan" is any constraint on how one attempts to work a problem. ∴ it is part of the dcrn. of the method of working the prob. However, to be a "plan" this dcrn. "morsel" must be shared by other problems. In this sense, it's a "morsel" would be "Named" if it were warranted — i.e. if it were used by enough probs. in the past. As such, it would be assigned a pc & a wt. for that pc. (N.B. so we'd know how to modify its p.c. when in the future, we get more data on its % of success).

A "PLAN" is anything 2 diffrnt problems solns (usually of different problems) have in common. A plan is somewhat "reasonable" if [redacted] naming it shortens t. total dcrns of t. solns. of t. 2 problems

"Naming" a plan, gives it a pc. — but ordinarily, we also have some (conditional) probty of t. effectiveness of a plan w.r.t. a particular problem. So, we "look at" t. problem using certain obs.. From th. results of these obs, we obtain a (pc of that plan wrt that problem) .... a conditional probability of a name (of a plan).

Th. name of t. plan is part of t. dcrn. of t. soln. attempt. After chosing a plan, we fill in various other parts of t. dcrn. of t. soln. attempt ... as specified by th. plan. Each of these parts of t. dcrn. has its pc., & so at each point in our carrying out t. plan, we have t. total pc to date ... as well as t. total cc to date. When, of course $\frac{cc}{pc}$ becomes $> T_0$, we drop that particular trial soln.

What all of the forgg. looks like: say a, b, c, d are successive elements in t. trg. seq.. a is simple enuf, so that t. machine can code it w. direct L search.

Then to code b, we want t. code for b, given a "good" code for a (not necessarily t. shortest).

Then to code c we " " " " " c " "good" codes for a & b.

.25 → A Plan is just another regularity type that has occurd in t. past. It is a particular peculiar regularity type: it involves, usually, t. structure of t. solns. to problems. As such, Plans are heuristics just like any .28 others. They can be dcurd by TM_2 just like any other heuristic.

The nice thing about this: that it removes t. "PLAN" idea, from a A.H. device, to a well-integrated part of t. TM system. This means that we can genz. plans, have t. same Gorc for them as for other heuristics,

.29

.30 Perhaps Newish ideas: That Heurs are perhaps only ragys, in which one has considerd cc — i.e. for each body of data & To level, there is a set of ragys. We can regard a heur as ↓ t. cc of a ragy that would normally be inadmissable because of excessively hy cc. — This is perhaps closer to t. older version of what a heur is.

.39 If .30 is correct, this does give us a more integrated view of TM's behavior.

→ At present, My impression of t. reason for t. TM_1, TM_2 dicotomy:

Moving TS file

8.11.81 ~~DM~~ : Index-like: List of Impt. ideas refered to in text. ~ 140

(TS)

1) Backtracking 94.01 — discn.

refd: 127.23, .27; 130.35

2) Sub conscious Mind; — discuss: 81.32-.34; 89.22 - .40; 91.01 - .24 (main discn.)
(on Neg. Reeft. for Animals, humans 112.01 - .40)

refered to: 140.26

3) Updating & Concept Learning — 1980: 80TS 36.01

4) Heuristics: What are they?: 80TS {77.25 — .[redacted]; [redacted] — .39} : 81TS 86.10 - .19; 53.01

.20 5) PLANS: a) What are they → 80TS 76.01 — 77.39 ← All 80TS. Also see 73.20 (81TS).
81TS 142.17; 150.01, 154.03

[redacted] b) what are some impt plans: (1) 68.10 ff; 45.07; 59.01ff: Is it prob. in a set I can solve? If not, can I xfm it into a prob. I can solve. [redacted] (2) GPS: [redacted] 73.01

(3) Slagle's Saint & Multiple. ~ to GPS but may be better. (SAINT: 80TS 74)

(4) Subdivision of corpus into "parts" at same level [redacted] a/o Heirarchically:

130.25 — 144.90

(5) ANALOGY.

(6) PEM & CPM's as Plans: 148.37 - .38; 150.14;

6) [redacted] Maxmethod: 149.16R; also 122.01

7) Probilistic Algms: Probablistic Proofs of thms in Geometry 76.01
Laplace's hypoth in Physics: Modfyd form: 100.01.

8) A kind of CB for [redacted] UMC's that may make Lsrn easier! 76.08

Maxmethod 122

Physics, Randomness 100

Prob types 99

Saint: 80TS 74

Index-like file of various topics.

See [redacted] 82TS: 2.20.82 "Pagoed" for a longer

Mainly TS file Mainly **81TS**

2.20.82 ~~DM~~ see (81TS 8.11.81) ~ p140 for another index-like file

(TS)

Some Noted topics of interest in TS 4.3.81 to 8.30.81 : 13.01 to 209.40 ← 164.40 looks like {164 was the last page written in 1981 (9.6.81)}

- Back tracking : 94.01, 96.38 ; 127.23 ← for back tracking as related to the stop rule v.s. no stop rule soln of the Eval problem. ; 92.30

Neg. Reenf't (why bad) : 112.

Subconcious Mind 91

Def'ns of Θ operator : $D(\Theta)$ 106.09

Genl. Admin of TM work 113. (is on what kind of paper to write.)

An early mention of the TS. Diffy that I worked on mainly ff : 116.25 : (to ~ 160.40)
(Not yet solved). This was the loop v.s. β^{12} form of the "Eval" operator
by 1983 I had solved it. Got 1 soln., then another soln. was implied by the "pluperfect TS." concept of 82TS 166.32 — ~ 174.40

Max Method 122 ; 149.16 R

Calcn. of rel. pc. of (≡ loop / stop rule) (No stop rule / no loop ($\equiv \beta^{12}$)) method of evaln. of alg. expressns : 117.19 / 118.17 — ff : Note for the more accurate work on this Q.
121.10, .23 ; 126.30 ; 127.17

~~8: My versn~~

- $H_x(y)$ & $H(X/y)$ my version! Variation of Chaitin! 13.01 ff

- Operator induction or extrapolating on I, O device 123.10.

Dividing up the corpus for Lsrch 139.01, 202.04 (≡ 162.04) (161.01 – 164.40)

- On "Plans" : (Analogy pp 76, 77.) ← This is in (from) 80 TS! (See other index on Plans)

PW (Parm Weighting) prob : 150.33, 151.00

Art on Evoln. of Intelligence : SA. June 65 pp 92–101

Associativity in **RPN** : 42.00 RPN infinal 40.27 → 43.21

Assigning type's to new defns : 48.28 (also refs to 80TS.

Substitution : the learning of : ~ 51.01
60.24 Bibliogy

Problem of : "How to parse an expression so it's in a certain form : 65.27

Optimum order of probs for a TS : 66.27

The "Generalization" heuristic. 67.01

GPS Heur : Xfm into solved prob. Heurs : 73.01

UNARY Functs, Hyp obs ; Boolean, Numerical constants : A List : 78.01

Ont. Appr. pof Numbers 122.25

~~Operator induction~~

- 81TS 159.30 : A kind of summary of the β^{12} v.s. Loop soln. problem.
Also note → 81TS 208.01 – 209.40

Reviews : ~~[illegible]~~

- 60.17 : work on T.S. up to that pt. Bibliog on various concepts worked on.

71.01 ff An outline of an Alg. TS. above

74.01 : "On browsing thru 80TS : some impt & amusing refs.

- 79.01 list of probs (in TS) solved or worked on : (few refs, hrr!)

93.01 Review of the state of the prob. at that pt.

99.01 A list of differnt types of problems. ← try's to show they are all equivalent to a certain type.

~140 (8.11.81) : A list of impt ideas in TS file (Mainly 81TS)

110.01 } A descn. of the TS being worked on (in English)
111.01 }

5.I.82 TS

53.35, .35; 39.35–40.08 ff. 94.16–95.04

O.K.: So t. Main diffys of t. methods of α [redacted] w. doubling (of 104.33) (& also γ in 107.17)

(1) 105.10 : Doubling/May not just double cc of solving t. T.S. [redacted]

.03 (2) 115.18 / 105.18 / 121.01 ← in particular / 166.23 : 167.03 : T. cc of solving a T.S. is ∝ t. cost of t. most difft concept it has : This seems wasteful. One would like A to vary over t. corpus. (125.25 may be helpful!)

.07 (3) 106.34 The c^2 [redacted] effect: If we have to checkout all trial GOP's on t. entire post of t. T.S., t. cc for solving t. probs alone (w.o. new t. c.c. of search) will be ∝ c^2, so it will be, for large enuf c (c = no. of concepts in T.S.) T. soln. to this may be related to that of (2): i.e. "localizn" of prob. [redacted] areas

Most Serious.

.17 (4) 105.02 If we add t. refinement of (γ) to α, so that t. probs of ff. T.S. need not be in exactly t. rite order (See 98.03–.20), then we must characterize t. params of this improvement before we can decide [redacted] that a particular value of A has failed & we must A ← 2·A [redacted] → 109.01 → See 116.38 for Bibliography on γ

also = 94.16–95.04

.20 5·2·82 | 2:18A

Also, See 114.23 for unifn w. (8)

.22 (5) 104.26 T. General Q of simply retaining only t. GOPs that solve t. most problems. This is an impt Q. that I may have sort of solvd, but I'm not sure as to t. reasons! — See 8) (107.35)

.24 (6) 102.03 : 101.37–102.03 → continues to 103.40 This is t. Q of just how TM can solve successive approxn. probs or Search probs. That take [redacted] squirrelly difft cc for each problem. 102.04 to n. 29 is a parlimy discn. but I don't yet understand this or some impt. assoc. Q's. — like what various pc's are t. probabilities of. → 120.31 seems to be relevant, however

SN (β) (64.35ff) is an Elzn. of t. GOP concept. It has assoc. w. it some (apparently) modeling of t. α & γ search techniques. I'll have to look into this after getting α, γ straightened out better. Also see if I can find other methods of elzn.

(7) 97.36 : Re γ : Can A.H. (spurious) solns. screw up t. use of γ to deal w. slightly misordered T.S.'s.? See 113.38

.35 (8) 110.10 T. needed concept may not always be the one that solves t. most problems immediately. See discn. following. Very impt.! → See 114.23 for unifn w. (4)

.37 (9) 115.35 A certain concept needed is not directly included in t. T.S.

.38 (10) 121.25 : Many Very redundant codes for t. same GOP. Gives spuriously low cc thresholds. — Wastes enormous amt. of cc. T. no. of redundant codes is exponential in t. no. of concepts in t. corpus! } Also see 141.09

5.20.80!

Impt. Problems, projects:

1) Work on TM < Perhaps organize past writings better.
Make list of headings: Use Files under a) headings
b) Time of creation

2) Work on good Computer System } keys
hardware. .....

3) Get [redacted] I.B. in good shape! Maintain it:
Get logging boys prob. straightened out.

4) Vacation travel.

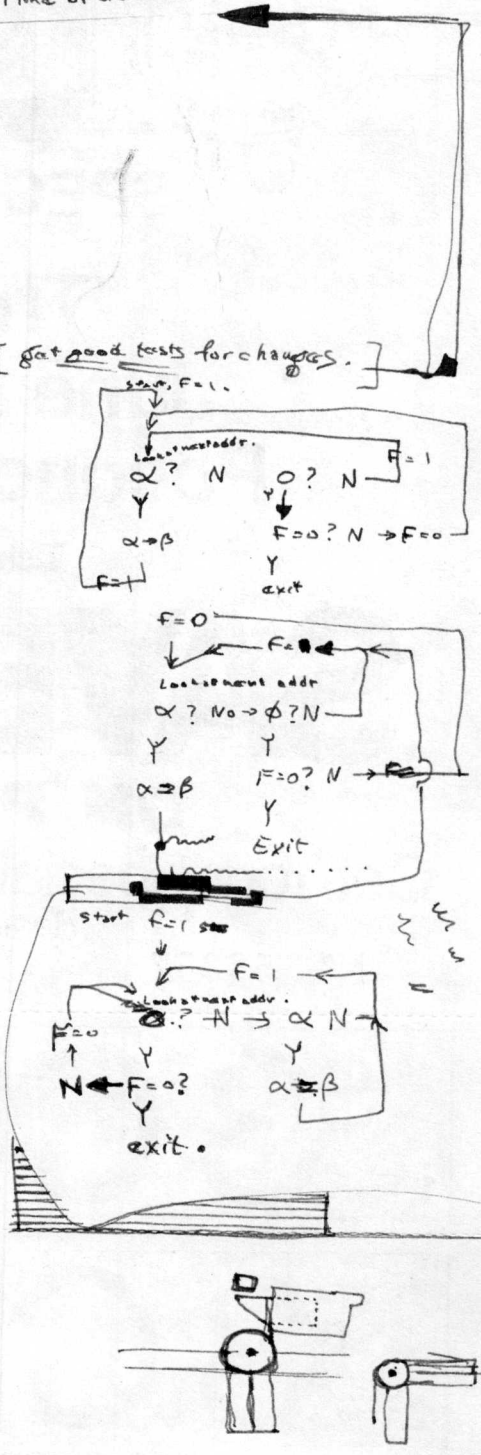
5) [redacted] Brain Drugs $, Hyperbaric $ O_2, etc. [Get good tests for changes.]

6) Clean up, straighten' out Camb. Apt.

7) Get new place to Live.

8) Get General System to decides on what to work on next.

9) (Part of ~~[redacted]~~ 1,2,9$) Devise filing system for objects, documents.



6.5.80 Genl. Admin., work Records, etc.

At all times, have in rapid access storage. the current most impt. unsolvd probs. I keep these in mind at all times. This makes it possl. for me to work on them all t. time, & spend lots of pure "mental doodling time" on them. Also, it insures my spending more time on t. truely impt problems.

Periodically (say every week), I can write a prog. report on t. 5 or 10 top problems, detailing possl. sub. probs., etc.

"Paging Problems": (1) In list of most impt. projects, How to decide on ordering (2) for "most impt problems" what to decide to put in Rapid access storage.

(3) In paper (or other files): I will have a pile of 100 or 200 pp. of readily accessable pages. Which pages shall I put there? Recency of last use is a factor. Also Genl. importance.

N2482

Gen Admin. (Plan).

I really need some means to get more work done on TM.
The main spurious Energy Sinks for me:

1) Loud
2) [redacted] computers.

Time spent on uc. could be useful toward TM, but so far, it hasn't been.

Bottlenecks in [redacted] uc: (1) reliability (a) Disc. (b) cpu.

(2) Getting good, easily used, operating system: Say ForPR.
Also fast ~~easy to use~~ hardware System

When I get O.S working O.K., I write work on RTM in Arcade game form. Discuss w. James Wheaton whom is more advanced. form.
Or, work on HR or SM (say, optics [redacted], or any very new kind of but offer)

I may need a more Global Plan.

Possy. of Getting Grant from [redacted] AFOSR. (or NSf at, say Berkely or UCLA).

Perhaps to obtain optimum Soln. to several of these

Problems: Go to Hyer Level Goal.

Interim Soln: 4 hrs/d. on TM (again!): Plan at least 4 hrs./day.

→ Try to devise more comfortable work place at 26B.
Get More comfortable Chair.
Perhaps tie & foam Cushions to chairy wood Chair..

$T = 1\,\mu s$
$E = 1\,\mu J$

$E_0 = \alpha \int_0^T E(t)\,dt$
$= \alpha \frac{E}{2} T^2$
$10 = \frac{\alpha}{2} \cdot k \cdot 10^{-12}$
$\alpha = \frac{20}{k \cdot 10^{-12}}$
$= 2 \times 10^{10}$

$\frac{2\times10^{10}}{\rho \cdot 3 \cdot 4} \cdot k \cdot 10^{-24}$
$\frac{10^{13} \times 10^{-24}}{\rho}$
$= 10^{-12}\,\rho$ (!)

7V, .35A
~ 2.5 W.

$\frac{2\times10^{10}}{10^9} E^4$
$t = 10^{-3}, E = 100$
→ 1 volt out!

3/19/83 TM:

.01 A genzn of the r-radix alphabet for CBI — so that ② T.V.H. method ~~[illegible]~~ gives ~~[illegible]~~ more like a continuous distribution of probys.

Use a large alphabet w. r symbols. Assoc with i^{th} symbol, is the proby p_i ; $\Sigma p_i = 1$.

One apparently good way to do this: use $p_i = \epsilon$ for $i \neq n$, ~~to~~ w. small ϵ. so that $p_n = 1-(n-1)\epsilon$: $\log_2 p_n$ is very close to $\emptyset$ i.e. $\approx -(n-1)\epsilon$. ; this $(n-1)\epsilon$ is the "quantization error" of the alphabet.

I guess r=2 is workable!

Now: How to map the original corpus into the new alphabet! or into strings in the new alphabet.

On 8/14: Maybe indication of TM ~~[illegible]~~ is not readable!

SN

42588 — Haven't been able to figure out meaning of ~~[illegible]~~ !