Tues July 17, 1956 T.M.

How to solve "moving problems", and "substitutions" with neural nets (in particular the "Contagion T.M.").

Special purpose scanning ckts. for positional ~~invariance~~ and size invariance (translations, rotations, dilitations — almost any projective xfmn). Also use ~~[illegible]~~ special ckts. for substituting one expression in another.

amt. of U it gives rise to — i.e. just how effective predictions based on it, are. In th. case of strs and ntpsts, it is not so easy to pin down th. U of an object, since it is imp. for such objects to be used alone in prediction. (This may not be too relevant either, since th. same thing is true for ΔU. ← No — th. ΔU of any object is clearly defined as th. ΔU tha betw. prediction runs with and without its use)

This appears to be a serious problem: i.e. How can one recognize objects that are good, in th. presence of other
.12 objects ~~That are~~ as good or better? see α79.34

.13 why th. $\square$ introduction (or elim. method) of α69.10 is ~~...~~ no good! a) Th. digms $\overset{S}{1}$, $\overset{S}{S}$, etc. appear very often. So what? — They are of no U. There is no simple way of getting rid of this annoyance. While th. method of α69.10 does help get rid of putting digms like $\overset{S}{S}$, ss etc. in th memory, it doesn't help much otherwise.

Th. best thing to do, at present, is to go back to th. old $\square$ method — using, perhaps, ops like $\overset{1}{\boxed{1}}$, rather than $\overset{1}{\square}$, since $\overset{1}{\boxed{1}}$ is more general.

In addition, th. use of >1 $\square$ per example, is expedient.

.31 → On th. other hand, while digms like $\overset{S}{1}$ etc. appear often, so do 11, 01 etc. In fact, $\square$ does not help much with $\overset{S}{1}$, since $\overset{S}{\boxed{1}}$ is of hy freq. ~~...~~ Actually 11 and $\overset{S}{1}$ are similar, in that they are both of hy freq., but low U. There appears to be no reason why th. idea of α69.10 cannot be used / yes there is ∃ See α81.01 for correct analysis

There is some Q., now about whether $\overset{1}{\square}$ or $\overset{1}{\boxed{1}}$ should be used. In studying a stochastic sequence, if one knows all digm. frequs, one can α75

T. M.

abs. method becomes reasonably effective in predicting, say new trial method is unlikely to have a hy ΔU. T.M. becomes very <u>conservative</u>. ~~This~~ This means that in order to make progress, <u>each step</u> of th. way must have hy ΔU. If there are 2 possl. methods of prediction, th. first one discovered will cause th. other to get small ΔU.

However, since T.M. has complete recent memory, if he gets badly stuck, he can backtrack some distance, suppress certain abs. methods, and allow others to develope.

I think I should continue with more detailed analysis of this Math T.M. — say thru ~, ⊕, ⊗ and +. See how long tetragms et. al. can continue — i.e. ~~ ~~ find out just where T.M. is <u>forced</u> to drop them, — ~~and keep~~ also keep tab of th. possible parallel developement of strs, ntpsts, etc.

Perhaps it is possl. to allow parallel developement of methods that are "allmost as good as th. 'best'"s. This seems reasonable, since th. "best" is only temporarily so, anyway. This would encourage th. str, ntpst methods ~~along~~ at th. <u>same time as</u> tetragms. ~~I Doubt~~ This would make B.G. more difficult, but one can always use some simple arbitrary method (like taking ~~choosing~~ th. choice of th. "best" ngm). It is not, however, immediately apparent as to how to do this. ~~X~~ If one has a fairly good prediction method already, than another one, which is almost as good, will give, if used along with th. first, a ΔU of ≈0. — This will be about th. same ΔU as an irrelevant method.

However: suppose one has 2 good methods. Then th. ΔU of <u>either</u> with resp. to th. other is ≈0 — i.e. th. difference betw. th. U. of using 1 and not th. other; and using both.

Perhaps th. situation isn't ~~ ~~ really so bad! If I have an ngmst., I can tell, to some extent, the in a similar way, we can factor $\sim\begin{smallmatrix}1\\0\end{smallmatrix}$, $\sim\begin{smallmatrix}0\\1\end{smallmatrix}$ into

$\boxed{1|2} \times \left(\sim \overset{c}{X}\left(\begin{smallmatrix}1\\0\end{smallmatrix}, \begin{smallmatrix}0\\1\end{smallmatrix}\right)\right)$.

We then try neigbors of $\boxed{1|2}$ e.g. $\boxed{1|\;|2}$

and find that they are U for both ntpsts

$\left(\sim \overset{c}{X}\left(\begin{smallmatrix}1\\0\end{smallmatrix}, \begin{smallmatrix}0\\1\end{smallmatrix}\right)\right)$ and $\left(= \overset{c}{X}\left(\begin{smallmatrix}1\\1\end{smallmatrix}, \begin{smallmatrix}0\\0\end{smallmatrix}\right)\right)$

We mite, if we like, form th. $\beta\leq$ of these 2 ntpsts.

— in fact we can, since several different strs. times this $\beta\leq$ yeald ngmsts of hy U.

There is always th. Q. of whether tetragrams would form sooner. In general, it is clear, that in th. absence of hyer order abstracting devices, for a sufficiently "long" problem — like

$$= \begin{matrix}1&0&0&1&0&1&1&1&0\\1&0&0&\cancel{1}&0&1&1&1&\square\end{matrix}$$

, it will take longer for th. trigms to "propogate" to the end, than for suitable tetragms to be formed.

There may be another "out" however. In using tetragms in problems with many $\square$'s in them, it is nec. to "propogate down" to get an answer at th. rite-hand end. If one uses trigms, one saves time by doing this propagation "once and for all time" — tho in general, th. propogation is slower and less routine than th. tetragm propogation.

$\boxed{\text{STV}}$ This business with one abstraction or abs. method "swamping out" another may automatically solve th. B.G. problem! What happends, is that once an

2) $=, \sim, \not\equiv$:

$\sim \begin{smallmatrix}1\\0\end{smallmatrix}$; $\sim \begin{smallmatrix}0\\1\end{smallmatrix}$; $\sim \begin{smallmatrix}1&0\\0&1\end{smallmatrix}$; $\sim \begin{smallmatrix}0&0&1\\1&1&0\end{smallmatrix}$

$\square$'s in all places other than $=$ and $\sim$.

We have problems like

$= \begin{smallmatrix}1\\\square\end{smallmatrix}$, $\sim \begin{smallmatrix}\square&1\\1&\square\end{smallmatrix}$, $= \begin{smallmatrix}0&1&0&1\\\square&\square&0&1\end{smallmatrix}$, etc.

Th. trigms $= \begin{smallmatrix}1\\1\end{smallmatrix}$, $= \begin{smallmatrix}0\\0\end{smallmatrix}$, $\sim \begin{smallmatrix}1\\0\end{smallmatrix}$, $\sim \begin{smallmatrix}0\\1\end{smallmatrix}$ are U.

Th. tetragms $\cancel{\equiv}$ $\begin{smallmatrix}1&0\\1&0\end{smallmatrix}$, $\begin{smallmatrix}1&1\\1&1\end{smallmatrix}$ etc. are usually o.k. — e.g. in $= \begin{smallmatrix}0&1&0&1\\\underset{\gamma}{\square}&\underset{\beta}{\square}&\underset{\alpha}{\square}&1\end{smallmatrix}$, one can fill in α, then β, then γ.

~~[illegible]~~ However tetragrams <u>wouldn't</u> help solve

$= \begin{smallmatrix}1&0&\square\\\square&\square&1\end{smallmatrix}$ or $\sim \begin{smallmatrix}1&\square&1&\square\\\square&0&\square&1\end{smallmatrix}$ — unless one uses trigrams with them — e.g. in

$= \begin{smallmatrix}1&0&\overset{\gamma_2}{\square}\\\underset{\alpha}{\square}&\underset{\beta}{\square}&1\end{smallmatrix}$ one can use ~~t~~trigms to fill α, then tetragms for β, then γ.

<u>**SOOO**</u> !! we are apparently stuck with th. <u>same problem</u> — i.e. Tetragms rather than th. uses of strs and ~~[illegible]~~ ntpsts. There is, however, th. possibility that ntpsts, etc. will be easier to ~~[illegible]~~. form.

~~E.G.~~ $\cancel{\begin{smallmatrix}1\\1\end{smallmatrix}, \begin{smallmatrix}0\\0\end{smallmatrix}}$ ~~form a useful ngmst.~~

~~It becomes reasonable to try~~

$\cancel{\boxed{1|2} \times \left(\doteq \overset{c}{\times} \left(\begin{smallmatrix}1\\1\end{smallmatrix}, \begin{smallmatrix}0\\0\end{smallmatrix} \right) \right)}$.

E.G. We can factor ~~[illegible]~~ th. ngmst $=\begin{smallmatrix}1\\1\end{smallmatrix}$, $=\begin{smallmatrix}0\\0\end{smallmatrix}$ into $\boxed{1|2} \times \left(= \overset{c}{\times} \left(\begin{smallmatrix}1\\1\end{smallmatrix}, \begin{smallmatrix}0\\0\end{smallmatrix} \right) \right)$ α71

T.M.

Let us start with = :

1) = : some examples:

$= \substack{|\\|}$, $= \substack{o\\o}$, $= \substack{|\,o\\|\,o}$, $= \substack{o\,o\,\blacksquare\\o\,o\,\blacksquare}$

Discn: a) $\substack{|\\|}$ and $\substack{o\\o}$ appear to be th. imp. digms.

= will not become imp, nor will this new type of example, become imp., untill ~ is introduced.

.10 b) How to introduce □, ■ If I don't introduce it, I seem to get an enormous amt. of memory space wasted. Th. way to get around this: Don't store anything untill a question comes up. Then store all ngms relative to that Q. This sounds very reasonable. There is some question as to how far into th. future and past this should extend. If one hasn't had a Q. of a certain type for a very long time, perhaps one shouldn't continue to collect data that would help answer it.

~~Great~~! Not so great see α73.13 see also α73.31 - which tends to say this is o.k.

c) discn. of b): monogram freqs. aren't of much use — in b), th. most freq. mongrm is "S" (≡ space) This doesn't make mongm freqs. useless, however, since the relative freqs. of 0, 1, =, ~, etc. may still be useful — particularly, in combination with other ngmsts, that may make it possl. to easily eliminate "S" as a possl. answer (B.G. problem)

O.K.: so $\substack{|\\|}$ and $\substack{o\\o}$ do become th. imp. digms; mongms are unimp.; as are trigms.

of hy U in ntpst. construction. c) seems more natural.

2) In Questions, put >1 $\square$ in, per example.

e.g. $= \begin{matrix}1101\\ 1\boxed{1}\boxed{0}1\end{matrix}$

This is certainly ultimately desirable, and it helps T.M. form ngmsts of very large n, and find them useful.

3) Note on 1): That one will use $\boxed{\alpha}$ ostensibly, but one will pretend that one really didn't, and that $= \begin{matrix}1101\\ 1\boxed{1}01\end{matrix}$ means that $= \begin{matrix}1101\\ 1\square 01\end{matrix}$ "was presented to T.M. as a problem," Then $= \begin{matrix}1101\\ 1101\end{matrix}$ was presented as an "answer." Also, that perhaps $= \begin{matrix}1101\\ 1101\end{matrix}$ was presented as a "normal example".

4) Th. trng. seq. should have examples like

$= \begin{matrix}\boxed{1}\\ 1\end{matrix}$, $= \begin{matrix}\boxed{0}\\ 0\end{matrix}$, $= \begin{matrix}10\\ 1\boxed{0}\end{matrix}$, $\sim \begin{matrix}\bullet\\ \boxed{1}\end{matrix}$, $\sim \begin{matrix}\boxed{0}\boxed{0}\\ 1\ 1\end{matrix}$ init.

This would encourage T.M. to form th. relation with = as impt. to th. problem. Also th. ngm $= \begin{matrix}1\\ 1\end{matrix}$ factors into

$\underset{str}{\boxed{1|2}} \times \left(=, \begin{matrix}1\\ 1\end{matrix}\right)$.

So $\begin{matrix}1\\ 1\end{matrix}$ becomes of hy U.

Note also $\boxed{1|2|2}$ is "close" to $\boxed{1|2}$ in some sense, and that

$\boxed{1|2|2} \times \left(=, \begin{matrix}1\\ 1\end{matrix}\right) \rightarrow\ = \begin{matrix}1\ 1\\ 1\ 1\end{matrix}$

approximately — along with other objects.

T.M.

=, ~, ⊕

4). For ∧ ⊗ $\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ \square & 0 \end{array}$, $\begin{array}{cc} 1 & 0 \\ \square & 0 \\ 1 & 0 \end{array}$, etc.

(There is ambiguity in $\begin{array}{cc} 1 & 0 \\ \square & 0 \\ 1 & 0 \end{array}$ also in $\begin{array}{cc} 1 & 0 \\ 0 & 0 \\ \square & 0 \end{array}$)

— At this point, clearly we cannot get 100% prediction with these hexagrams alone.

This difficulty would not be serious, if it had turned out that even befor this, one could not get 100% accuracy — say we fed T.M. problems like

⊕ $\begin{array}{c} 1 \\ \square \\ 1 \end{array}$ befor ⊗ was introduced.

If this had been introduced befor, T.M. wouldn't try too hard to get 100% prediction accuracy and would be ~~—~~ relatively satisfied with these hexagrams for ⊗ [This may not be true].

There is somewhat of an impasse here. It seems like T.M. is learning most everything with straight ngm. freq. counts, and is not making worth-while abss.

Some possl. remedies:

1) Do not use $\boxed{\alpha}$ around examples. Just use it on questions. Let T.M. make freq. counts on digits — but also let him make freq. counts on correct answers to questions. Say we ask and T.M. would then answer a Question each time. make both normal freq. counts, and freq. counts for $\boxed{\alpha}$ type squares. This would a) ↑ samplesize b) permit useful ngms to be derived which would be

very poor. How can one recognize ~~these new~~ This "new type" of example? — One characterizes all old examples by certain ~~param~~ abss. Th. new examples do not conform to These abss. E.G. If Th. symbol ⊕ appears for Th. <u>first</u> <u>time</u>, one is not too surprized that all of Th. old ngm freqs. do not hold. In such a situation, T.M. will use all of Th. old ~~ngm~~ ntpsts, strs and ngmsts — but mainly to form <u>new</u> ngmsts. Th. old ngmsts will not be expected to be of <u>very</u> hy U in prediction.

An imp. pt. here is that ~~ngmsts~~ ngmsts <u>not</u> containing □ will be of hy U in deciding whether an example is "unusual" or not. E.g. If Th. symbol ⊕ is "new", Th. symbol □ is not contained in this fact.

This idea is 1) <u>good</u> 2) Essential for T.M. eventually. It may even be O.K. to insert it as an initial Operator → but, rite now, try to review just how far one can get with Th. simple rules that I have. —

i.e. 1) simple freq. counts of ngms.
2) see just how much it helps to put ngms together to make new ngms. (this uses strs, and ntpsts that are Th. $\overset{c}{\times}$ of ngms)

At Th. present time, it is clear that T.M. could solve all probs up to <+ with carry line>, using only ngm. freqs.

1) for = : $\begin{matrix}1\\ \square\end{matrix}$ etc.

2) for = and ~ : $\begin{matrix}1 & 0\\ \square & 1\end{matrix}$; $\begin{matrix}1 & \square\\ 1 & 0\end{matrix}$, etc.

3) for $\overset{=,\sim,}{\wedge}$⊕ (≡ βΣ) $\begin{matrix}1\\ 1\\ \square\end{matrix}$, , $\begin{matrix} & s\\ 1 & 0\\ \square & 1\\ & s\end{matrix}$, etc.