created a new prgm.

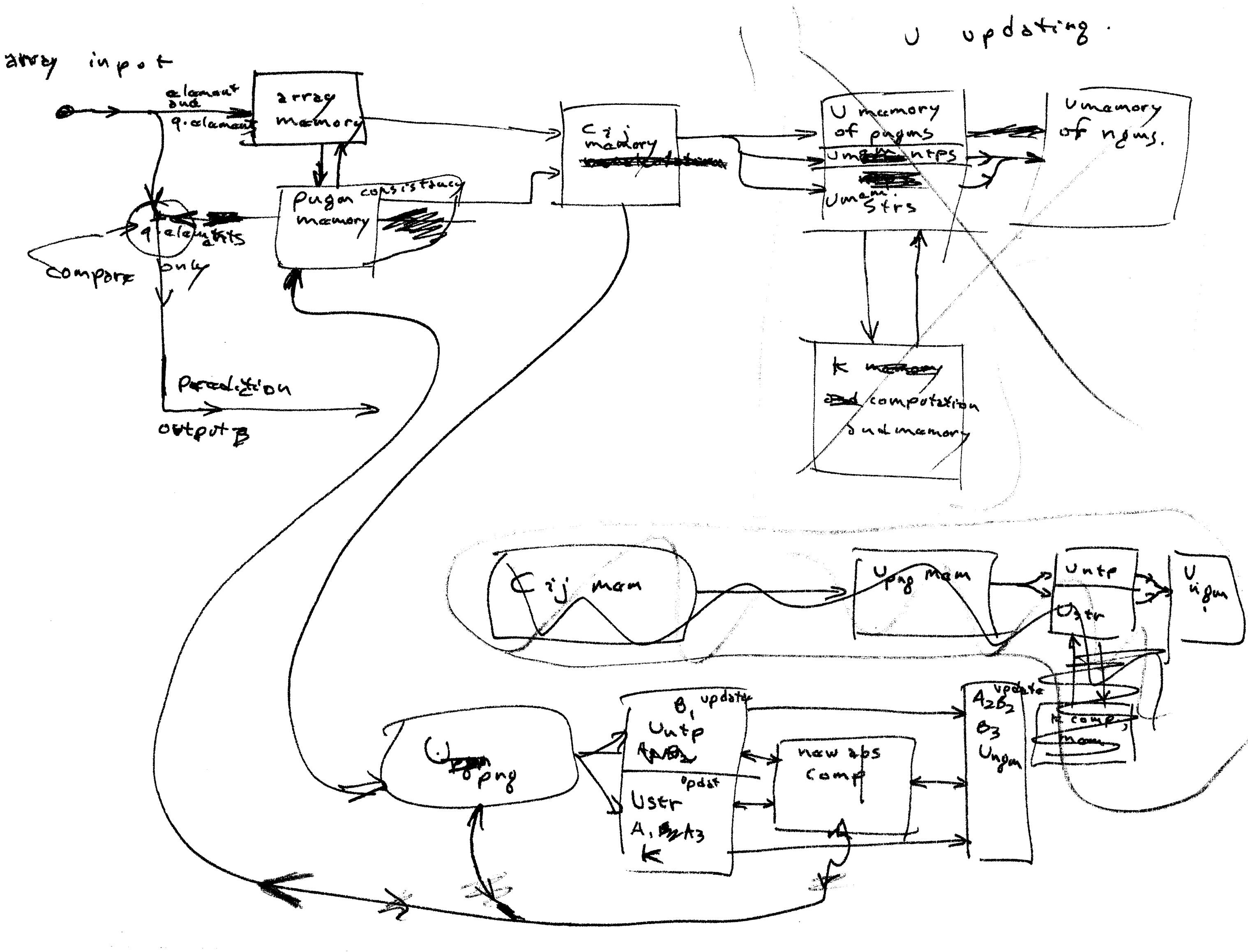
→ these are now pp. 14 - 38. Dart. Report

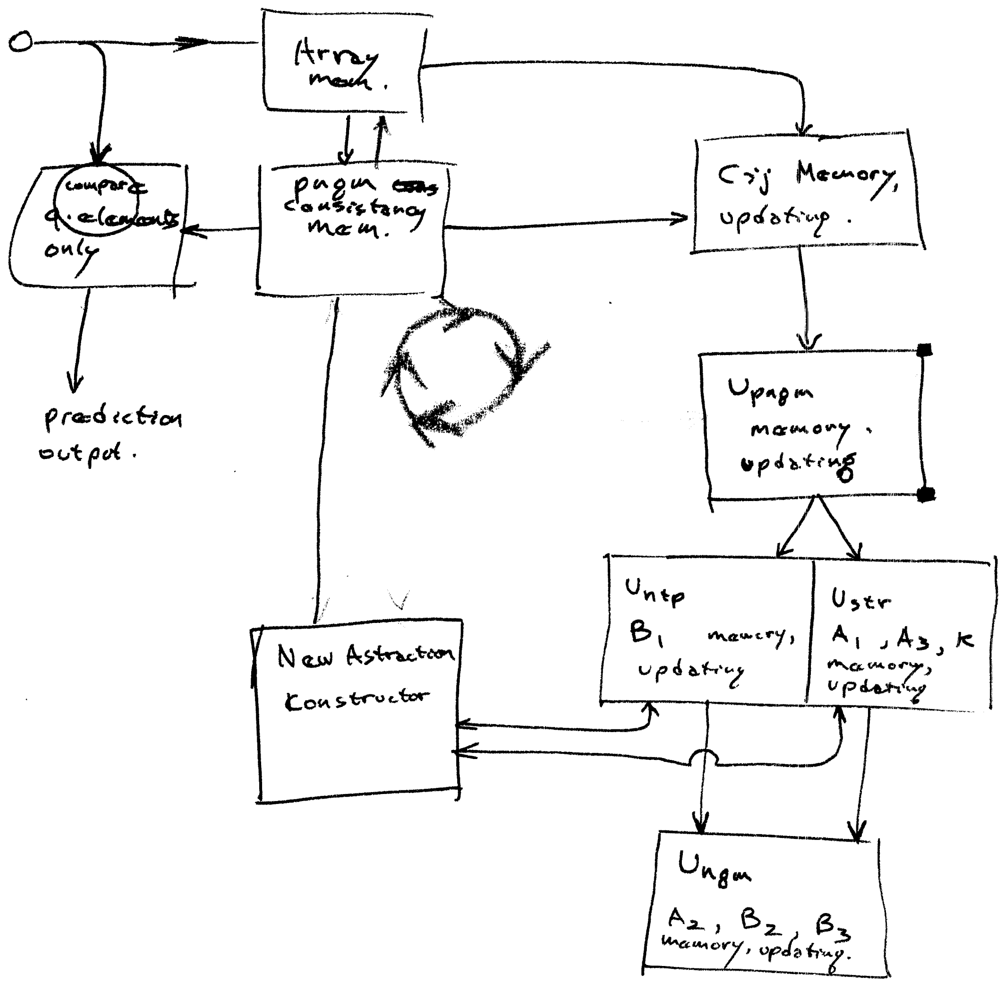
.02 in α122.06): If all of th. new prgms are inconsistant, do any of th. abss. need to be updated? — yes Well — just what happends when new prgms are created? — [redacted]

1) str x ntp
2) . . .
3) . . .
4) . . .

← (see α114) Many new abss. are created. Even if the old c_{ij}'s ~~[illegible]~~ are th. same, there are new i's and j's — ~~[illegible] all of th. new c_{ij}'s are zero~~. There are new c_{ij}'s — they are all zero. They do, however, effect all of th. U's, since some new U_{ij}'s = 0 — these U_{ij}'s did not, however, exist before.

Block diagram of α121, α118, α122)





A possl. block diagram. Flow for ~~=~~ search for applicable ~~***~~ pngm / with updating is red loop.

This flow diagram isn't too clear.

Wed Aug 8, 1956

Th. use of "C_{ij}" in th. exposition, for th. counts of all pugms ~~[illegible]~~ is bad. Some pugms are not express-able as th. product of a str. and ntp.

Re hash of T.M. ~~loop~~ operation rules:

1) T.M. gets element:

a) Store in array mem.
b) Scan thru pugm mem. to see if any C_{ij}'s have been changed due to an old pugm becoming inconsistent
~~do α118.05 to .11~~ ~~change these a bit.~~ ~~don't mention poss. of skipping.~~

c) If no C_{ij}'s change stop and wait for new input.
If ~~see~~ one or more C_{ij}'s change, do α118.05 to .11. and wait for new input.

make into 2 steps

2) T.M. get q. element.

a) store in array mem.

b) scan thru pugm memory to see if any C_{ij}'s have been changed due to an old pugm becoming inconsistent.

SN This is unnec., but it will ↓ th. confusion in th. loop that has been written. Unnec. because th. updating of C_{ij} when a new element comes in is unnec. This can be done when and only when a new q. element comes.

---

c) search thru pugm mem. for appropriat. consistant pugm.

d) If proper pugm is found, change proper C_{ij}.

e) If ~~any C_{ij}'s have changed that~~ 2.b) or 2.c), have resulted in any change in C_{ij}, do α118.05 to .11, then go to 2f)
If not, go directly to 2 f)

f) If ~~2~~ 2.d) yeilded suitable ~~proper~~ pugm, stop and wait for new input. If 2 d) did not yeild ~~proper~~ suitable pugm, go to step 2 ~~8~~)

~~2~~ 8) Construct / several new pugms of hy apriori U that / are not in th. pugm. mem, that apply to th. interrogation sq. in q. Do this by constructing new strs, ngms, utps and combining them, using methods of (α 114.32 ff.)

$\boxed{\text{SN}}$ Whenever New abs. are created — whether or not C_{ij}'s are changed, we must update all abss. — However, whenever we create new abss., we also / always create new pugms — ~~whenever~~ even if their C_{ij}'s ~~are~~ may be zero, this does affect ~~all of th. pugms~~ many of th abss.

Th. moral is — that th. ~~loop~~ program of α121 α118 α122 is probably O.K., tho not too easy to understand.

SN on "search" problems: Just run thru some problems to see just what kinds of "characters" one uses in directing search. These characters may be of th. kind that either chose / new sub-goals, or tell a reasonable step to try, toward a gn. sub-goal. — or, when a gn. sub-goal is gn, tell whether a step should be taken, or a new sub-goal placed. Also time estimates on various paths are imp. "Sequential analysis" may be imp. — just as a method of directing one's attention to th. rite points.

I don't think this problem is particularly diff. — apart from th. routine of getting new tricks from old.

My approach to th. / sub-proble " ~~...~~ ~~thendan~~ construct a pugm that fits th. q.element " has been in th. spirit of Ashby: i.e. describe ~~...~~ a large no. of possibilities, then use exhaustive search, since th. problem is then "well defined".

Thurs Aug 9, 1956

In th. problems in which I applied this method, th. search wasn't long enuf to make any difference, but eventually, I will have to get around to th. "search" problem. What I may need is only a quick way of searching, rather than a general method of solving all search problems — tho probably th. latter is required.

Exposition: Mention that most of th. equations are nec. only "in spirit," ~~that extremely rough~~ to give an idea as to just what ~~variables the~~ measurables the various U's depend upon. In a real machine, only the roughest approx. solns. would be used. Probably powers of 2 are adequate in representing their values.

Serious trouble! After doing " = " T.M. finds a certain set of abss. useful. However, as soon as ~ is introduced, all of th. old pngms become of zero U, since they are inconsistent, and so do all of th. other abss. that are based on th. U's of th. pngms. This difficulty can be elim. in several ways — a good one is using NMTM, in which a pngm is not discarded untill many counter examples are found.

— However, probably th. best thing to do is leave it alone, and start from scratch with $\sim \begin{smallmatrix}1 & 0 & 0 \\ 0 & 1 & 1\end{smallmatrix} \ldots$,

---

Also, explain technique of forming relevant pngms. — by a) starting with a possl. pngm. and trying to construct it by basic operations (the inversion problem)

or b) Iterating all possl. pugms with U < a certain amt., and chosing out th. relevant ones. (Exhaustive search, ruffly in orderof apri U.
We have a simple ~~clue in a)~~, however, since we have the apri U's ~~whic~~ of th. components. — also if a "useless" pugm is constructed, it may very well be useful later. I.e. this is analogous to inverting sin x by constructing a table of sin x. It is practical if $\sin^{-1} x$ is used very often. In th. present case this ~~[illegible]~~ may or may not be so.

This is imp. and interesting: That th. finding of a relevant pugm, of th. hyest possl. aprip U, is a "well defined problem of th. second kind" → more exactly, th. "$1\frac{1}{2}$th kind"! and it is an inversion problem — like ~~solving~~ finding a sequence of string xfmns ~~operations~~ that proves a Theorem. It is a problem in which sub-goals are imp., and in which one has, to some extent, an integrated servoing signal, that tells one how hy th. Uaprip is, of th. ~~state~~ present state of xfmn.

.28 Also write Introduction, saying that this is not a progress report in th. ord. sense, but merely serves to indicate ~~[illegible]~~ in what direction I am working — th. general methodology and approach used. While some details are given, the details are only to make the methods clear. ~~[illegible]~~ The details themselves are not necessarily those given in the text, in addition, these details have been ~~will be modified con~~ and will continue to be considerably modified, as the work progresses.

NUMBER "v.s." sets of sets . . . etc.

It is somewhat likely, that humans store th. concept "3", not as th. set of all triples, but as the result of a "counting" operation ~~or~~ ∃ ~~...~~ — Matching objects to th. integers.

— However, this does, correspond to th. equivalence class of all triples — i.e. 2 things equivalent to th. integer 3, must be equivalent to each other.

At any rate, I think that there ought to be a more natural, directly intuitive approach to ~~...~~ all ~~of~~ th. concepts that I want T.M. to learn.

IMP. IDEA: Instead of trying to get T.M. to "~~discover~~ invent" th. concepts of "negative numbers," "imaginary nos.", etc, give it problems, ∋ th. soln. does not directly involve these objects, but could most easily be obtained using them. Then see if T.M. could ~~...~~ form the internalized abss. that would yeald the correct answers. I can "look inside" T.M. and see just what trg. seq. is nec. so that it would tend to form, internally, abss methods equivalent to neg. nos. and im. nos.

I don't think that T.M. is really proceeding properly. What I have done, is taken ~~these~~ some combination and xform methods, that I think are reasonable, and tried to make them do what ~~is~~ required.

What should be done, is to try to trace out my intuitive processes in solving these problems then find a set of epist. rules that form an adequate "basis" for them.

General Discussion: From about ~~α1~~ $\alpha 120$ to $\alpha 130$ there has been an enormous waste of time and effort that may have to be rewritten.

T.M. should be gn. q. elements like

$=\begin{matrix}1\\1\end{matrix}$, $\sim\begin{matrix}1\\0\end{matrix}$, $=\begin{matrix}5&0\\5&0\end{matrix}$, etc., which make it

imp. for it to develope tetragrams, etc., which turn out to be useless, in th. long run.

We want to develope strs like $\boxed{1|2}$, $\boxed{1|\ |\ |2}$, $\boxed{1|\ |\ |\ |2}$, etc., which are useful in th. long run.

A disturbing feature, is that about a month ago I realized this difficulty and its "soln", and wrote some about it.

There is th. possy. of just continuing in th. present direction.

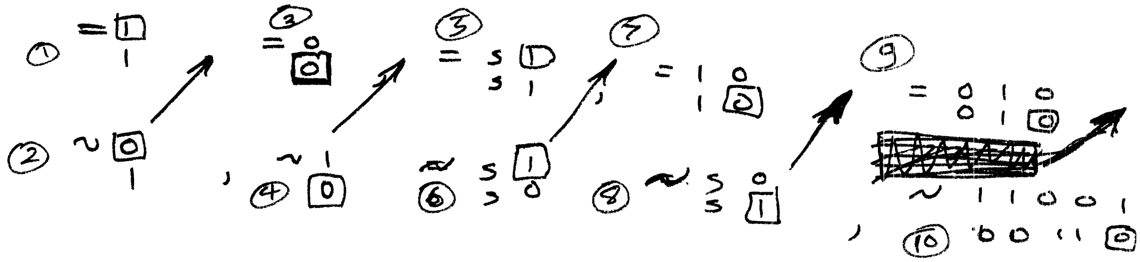
SN I certainly don't like this discontinuous behavior, in which a whole bunch of abss suddenly change there U values to near zero when a few pngms become inconsistent.

Any way! A possl. plan for th. chapter on "examples".

1) List the comb. rules and equations that are to be used. If extra equs. are used, indicate that ∃ a, b, c are to be added to.

2) Make outline of th. tag. seq. Then determine just which pugms, ngms ntps and strs become imp.

.10 So first th. tag. sequ:



Suppose T.M. get $= \begin{matrix} 001011 \\ 00101\boxed{1} \end{matrix}$ after $= \begin{matrix} 110 \\ 11\boxed{0} \end{matrix}$, so it has to make a rather large "jump".

.25 i.e. from th. ~~#~~ str. [1|1|2] to [1|1|1|1|2] . to 80).01

On Inversion (cart. and other inversions)

If $str \times N_t = P$ with N_t and P of hy U, then $U_{str} = \frac{1}{n}\sum_{i=1}^{n} U_{P_i}$ − const. — with no special rule for this inversion. So inversions of this sort, are simply a rapid method to find objects of hy U, not a "really" new method.

Inversions are "Economy Measures".

Fri Aug 10, 1956

Possibly Great Idea: Mixing ~~methodology and~~ methods of forming new words, with the words themselves.

Suppose we store the word "3" as a set of 3 objects, plus a set of <u>instructions</u> that tell how to match it with other sets of objects.

The relation "1 greater than" is a set of instructions telling how to get the set of instructions for $n+1$ from the set of instructions for n.

In general, this is a set of instructions for getting a new object from an old. Expressed extensively, we would simply write all ordered pairs of the old and new objects. In practice — (intensive definition), we would give the rule. An example is the str., S. We can express it as the relation $(S, N_{ti}, S \times N_{ti})(i=-\infty \to +\infty)$ or as the method by which one goes from N_{ti} to $S \times N_{ti}$.

A set of instructions for getting a new word from an old, is a <u>relation</u>, But it is also a <u>device</u> used to create new abss. As a <u>relation</u> it is a <u>word</u> — as a <u>device</u>, it is part of the <u>methodology</u>.

This is related, perhaps, to the str. scanning scheme of 67) and 68)

.01 from 78).25 :

① = ■ over 1 — □, ⓪, etc. tried — N.G.

= □, □ over 1 tried — (6 possl. cases — all indecisive.)

1.5 = 1 1 ■ = [1] and [1] over 1 now only consistant digms.

How to Solve Problems: ① state problem. ② solve problem or break into sub-problems ③ Solve sub problems or write down next problem ④ Solve any unsolved problems, break them into sub-problems or write down new problems. ⑤ same as ④, ⑥ same as ⑤

---

② ~ [0] over 1 — prediction is 1

3.5 ~ 0 over 1 — ① over 1 is inconsistant.

③ = 0 over ■ — = [1] inconsistant. 0 over [1] and = [1] are possl. digms: ~~ambiguity, so no prediction.~~ prediction is 1

3.5 = 0 0

0 over [1] and = [1] are inconsistant. There are no consistant digms.

④ ~ 1 over ■ — ~ [1] and 1 over [1] are consistant. Th. prediction is 1

(4.5) ~ 1 over 0   ~ [1] and 1 over [1] are incons.

(5)

The Tng. seq. on 28.10 is too fast, and has ~~too~~ little info, per element. Perhaps I should make the Tng-seq. fairly ad.hoc, to illustrate the various methods of combination.

(1) = 0 ■ / 0 ■

(2) = [1] 0 / 1 0 : [1] is incons.
[1] over 1 is consistent.
∴ prediction is 1

(3) = ■ [0] / ■ 0   [0] over 0 is consistent.
prediction is 0.

(4) ~ 0 0 / 1 1   [1] over 1 is incons.

(5) ~ ■ 0 / ■ 1   1 over [1] is cons.   prediction is 1

(6)   1 over [1] is incons.   ~~and~~ ~ [1] ~~are~~ cons.

~ [■] 1 0 / ● 0 1   [1] over 0 is cons

prediction is 1

(7) = [1] ■ 0 1 0 / 1 0 0 1 0   no cons. digms.
= [ ] 0 , = [1] over 1 , [ ] 0 over 1
cons   cons   incons.
prediction is 1

(82)

$\boxed{1}$ over 1 and ∴ 1 over $\boxed{2/1}$ are imp. ngms. Also th.

str$_5$ $\boxed{1/2}$, $\boxed{2/1}$ and ~~to~~ some extent, $\boxed{\begin{matrix}1\\1\end{matrix}}$.

SN It mite be well to make th U of an ngm. a mean of th. pngms that it could by xfand into. This simplifies updating ~~and~~ ~~[illegible]~~ a great deal, since there then exists only about 1 imp. updating equ. for U's.

---

(8) ~ 1 0 1 over □ 1 0   no cons. digms.

Use more elements, — otherwise there is no way for t.M. to learn about th. environs of ~ and =.

What about many = examples, with □ randomly placed — then ~~[illegible]~~ a few ~ elements. Then ~ 1 0 over □ 1 should be O.K. followed by = □ 0 over 1 0 . . .

O.K. :

(1) = 1 0 1 over 1 0 1 , (2) = 0 0 $\boxed{1}$ over 0 0 1   ~~[illegible]~~ $\boxed{1}$ over 1 is cons.

as is 0 $\boxed{1}$ and 0 $\boxed{1}$

prediction is 1

(3) = 0 0 1 1 over 0 $\boxed{0}$ 1 1   ~~[illegible]~~ 0 over $\boxed{0}$ is cons. 0 over $\boxed{1}$ is cons, ∴ no pred.

(4) = 0 0 1 1 0 over 0 0 $\boxed{1}$ 1 0   0 over $\boxed{1}$ is incons. 1 over $\boxed{1}$ is cons.

prediction is 1

5 $= \begin{matrix} & 0 & 0 & 1 \\ \boxed{1} & 0 & 0 & 1 \end{matrix}$ $\begin{matrix} 1 \\ \boxed{1} \end{matrix}$ is cons.

pred. is 1.

6 $\sim \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$, 7 $\sim \begin{matrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{matrix}$, 8 $\sim \begin{matrix} 0 & & 0 & 1 \\ 1 & 0 & 1 & 0 \end{matrix}$ 9 $\sim \begin{matrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$

10 $\sim \begin{matrix} \boxed{0} & 1 & \\ 1 & 0 & 0 \end{matrix}$. no cons. digms. at all

$\sim \begin{matrix} \boxed{0} \\ 1 \end{matrix}$ is cons, ~~$\boxed{0}$ 1 is incons~~

~~$\sim \boxed{0}$ 1 is cons.~~

~~$\sim \boxed{0}$ is cons.~~

Prediction is 0.

$\begin{matrix} \boxed{0} \\ 1 \end{matrix}$, $\begin{matrix} 0 \\ 1 \end{matrix}$ hy U digms

11

$= \begin{matrix} & 1 & 0 & 1 & 1 \\ & \boxed{1} & 0 & 1 & 1 \end{matrix}$

$\begin{matrix} \boxed{2} \\ \boxed{1} \end{matrix}$, $\begin{matrix} \boxed{1} \\ \boxed{2} \end{matrix}$, $\boxed{1\,2}$, $\boxed{2\,1}$ hy U strs

(0, 1), (1, 0), imp pairs

$(\ , \begin{matrix} 0 \\ 1 \end{matrix})$, $(\begin{matrix} 0 \\ 1 \end{matrix}, \ )$ imp. strsn. pairs.

no digms are cons

$= \begin{matrix} 1 \\ \boxed{1} \end{matrix}$ is cons. so $\begin{matrix} 1 \\ 1 \end{matrix}$ gets hy U.

$\begin{matrix} 1 & 0 \\ \boxed{1} & \end{matrix}$ incons. $\begin{matrix} 1 & \\ \Box & 0 \end{matrix}$ incons.

$= \Box\ 0$ incons.

so $\begin{matrix} 1 \\ \boxed{1} \end{matrix}$, $\begin{matrix} 1 \\ 1 \end{matrix}$, ~~become~~ get hy U

$(=, \begin{matrix} 1 \\ 1 \end{matrix})$, $(\begin{matrix} 1 \\ 1 \end{matrix}, =)$ is hy U.

(12) $\sim \begin{matrix} 1 & \boxed{} & 0 & 1 \\ 0 & 1 & 1 & 0 \end{matrix}$ try $\boxed{1|1|\,|2}$, $\boxed{2|1|1|\,}$ times $\left(\sim, \begin{matrix}0\\1\end{matrix}\right)$.

∴ hyU.

(13) $= \begin{matrix} 1 & 0 & 0 & 1 \\ 1 & \boxed{} & 0 & 1 \end{matrix}$

$\begin{matrix}0\\0\end{matrix}$ tried since $00 = \boxed{1|1} \times 0$.

$= \begin{matrix}\cancel{8} & 0 \\ \cancel{8} & 0\end{matrix} = \boxed{1|1|1|2} \times \left(=, \begin{matrix}0\\0\end{matrix}\right)$

$= \boxed{1|2} \times (=, \boxed{} \times (0))$

(14) $\sim \begin{matrix} \boxed{1} & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{matrix}$

$\sim \begin{matrix}\boxed{1}\\0\end{matrix}$ is consistant. $\left(\sim, \begin{matrix}1\\0\end{matrix}\right), \left(\begin{matrix}1\\0\end{matrix}, \sim\right)$ are hyU.

(15) $\sim \begin{matrix} 0 & 0 & 1 & 1 \\ 1 & 1 & \boxed{0} & 0 \end{matrix}$ $\boxed{1|1|1|2}$, $\boxed{2|1|1|1}$

e 23

Also $\boxed{1|2} \times \left(\begin{matrix}0\\1\end{matrix}, \begin{matrix}1\\0\end{matrix}\right) = \begin{matrix}0&1\\1&0\end{matrix} \rightarrow \begin{matrix}0&1\\1&\boxed{0}\end{matrix}$ appears and is consistant. hyU.

also $\begin{matrix}1&1\\\boxed{0}&0\end{matrix}$

to 86(6).25 (directly)

Th. Q. is, "Is $\boxed{1|1|1|2}$ closer to $\boxed{1|2|\,}$ than $\begin{matrix}0&1\\1&0\end{matrix}$ is to 0, 1, $\boxed{1}$ etc.?"
str.

Th. dist. betw. th. strs. is 2, but th. apriori U of $\begin{matrix}0&1\\1&0\end{matrix}$ mite be rather low, since it may have to be built from scratch. —— Tho not so; we have, at a certain level, th. idms $\begin{matrix}0\\1\end{matrix}, \begin{matrix}1\\0\end{matrix}, \begin{matrix}0\\0\end{matrix}, \begin{matrix}1\\1\end{matrix}$. So we can get pairs by th. cart. prod. of this set with itself — then mult by strs $\boxed{1|2}$ or $\boxed{2|1}$ to get $\begin{matrix}0&1\\1&0\end{matrix}$, etc.

So at this stage of th. game, we cannot afford th. jump —— in fact, whether we like it or not, Th. tetragram ~~[illegible]~~ $\begin{smallmatrix}0&1\\1&0\end{smallmatrix}$ is of such hy apri U, that it would get formed along with, say $\boxed{1\,|\,1\,|\,2}$, provided $\begin{smallmatrix}0\\1\end{smallmatrix}$ and $\begin{smallmatrix}1\\0\end{smallmatrix}$ were around. In such a case, the str would not → say $\boxed{1\,|\,1\,|\,1\,|\,2}$, since it would be unnec, since th. ptetragrams would all be available.

However th. str. $\boxed{1\,|\,1\,|\,1\,|\,2}$ would have very hy U, since it has/had many cases in th. past — from both = and ∼. Th. tetragm $\begin{smallmatrix}0&1\\1&0\end{smallmatrix}$ has had only cases from ∼, and these aren't too frequent, because of its individual str. — i.e. there are 4 ∼ t.gms: $\begin{smallmatrix}1&1\\0&0\end{smallmatrix}$, $\begin{smallmatrix}0&0\\1&1\end{smallmatrix}$, $\begin{smallmatrix}1&0\\0&1\end{smallmatrix}$, $\begin{smallmatrix}0&1\\1&0\end{smallmatrix}$.

We get about 2 ~~of the~~ cases of these per q.element, so that this is only ∼ ~~[illegible]~~ ½ as frequent as some ~~of~~ th. strs like $\boxed{1\,|\,1\,|\,2}$. Tho this particular str. is not used every time — so th. tetgms. may have hyer U after all!!

Well — if they want to have hyer U, there isn't much one can do about it!

Nor that is nec to do about it! With ⊕, all old ptetragms will become uncons.. Th. ngms $\begin{smallmatrix}1\\1\end{smallmatrix}$, $\begin{smallmatrix}0\\0\\1\end{smallmatrix}$, $\begin{smallmatrix}1\\0\end{smallmatrix}$, $\begin{smallmatrix}0\\1\end{smallmatrix}$, will still be hy U, as will $\boxed{1\,|\,1\,|\,1\,|\,2}$, ~~[illegible]~~ $\boxed{1\,|\,1\,|\,2}$, etc.

These dgms will suggest ~~[illegible]~~, ~~etc.~~ $\begin{smallmatrix}1&1\\1&\end{smallmatrix}$, $\begin{smallmatrix}1\\1&1\end{smallmatrix}$, $\begin{smallmatrix}1\\1\\1\end{smallmatrix}$, $\begin{smallmatrix}&1\\1&1\end{smallmatrix}$

etc. . .

~~[illegible]~~ When we get ngms like $\begin{smallmatrix}1\\0\\1\end{smallmatrix}$ and $\begin{smallmatrix}1\\1\\1\end{smallmatrix}$, it will be quite reasonable to try $\begin{smallmatrix}1&1\\0&1\\1&1\end{smallmatrix}$

~~This will be~~ For ~~[illegible]~~ cases like ⊗ and even +, this will continue to be true. T.M. will alway start using $\boxed{1\,|\,1\,|\,1\,|\,2}$, etc, which will have enormous U, in retrospect, but eventually th. ~~[illegible]~~ trigms or tetragms will form, and then hexagrams or octogm will form from them by simple juxtaposition .

A way to avoid this and make th. str. method nec. is to ~~o~~ occassionally give examples like :

= 0 5 0 1 5 1 5
0 5 0 1 5 $\boxed{1}$ 5

16 wps

Anyway, we get back to th. old saw, that if a technique is nec., it will be found — if not, it is less likely.

3 wps

Italico Cat

So: continue with th. examples: Go to ⊕, ⊗ and +, trying to throw in ---(-) occassionally, and noting (to reader) that Hexagony will be used, etc.

//,

After th. examples, note to reader that th. generalize structural extrapolation from $\boxed{1\ 1\ 2}$ to $\boxed{1 \ldots n \ldots 2}$ will not take place until th. concept of number occures, in th. sense that is useful or ~~nec~~ nec., elsewhere as well, ... That "Th. concept of no." is being "worked upon".

.25 ~~sss~~ from 84).23

(16)

~ 0 0 1 1 1
1 1 0 $\boxed{0}$ 0

1 1
$\boxed{0}$ 0 is consistant and is used.
prediction is 0 .

$\boxed{N}$ I have been assuming that I.M. stores in its pugm memory only pugms that have at least 1 case — ~~whether or not they~~ both consistant and inconsistant pugms. { This fact ~~was~~ has been ~~th~~ made part of th. rules — see α122.10

~~(17)~~ ~~~0011~~ ~110$\boxed{1}$01 .
001010

$\boxed{1' \ | \ | \ | \ | \ 2}$ is used (? ?)
why ?

18 (+)

1 1 0
0 1 0
$\boxed{1}$ ■ 0

$\boxed{\frac{1}{2}} \times (1, 1) =$ 1 1 1 → 1 1 $\boxed{1}$.

~~since no other trigm or tetrgm~~ fits.

19 (+)

0 ■ 0 0 1
1 0 0 ■
1 $\boxed{1}$ ■ 1
0

0
0
0

There are no cons. trigms of th. form 1 0 $\boxed{1}$, because 1 0 s has occured. 0 1 $\boxed{0}$ 1 is, however, cons., as yet, and has reasonaby hy apri U. — so th. prediction is 0.

19 (+)

■ 0 1 0
0 ■ 1 0
1 1 $\boxed{1}$ 0

0 1
1 $\boxed{0}$ → incons.

0
$\boxed{0}$ 1 1 → incons

1 1
1 $\boxed{1}$ still is cons. yet.

also 1 0
$\boxed{1}$ 0 is cons.

See 93.30 for 18 and 19

SN If only pugms with at least 1 case apeput in th. pugm mem, we will not have T. M. recognize that a certain new str (say) would have been good in th. past, as well as now. T.M. only uses new strs to make new pugms.

Not so: In getting th. U of a str., it is mult. by every ntp. that it can, an th. U of th. resultant pugms is taken. This U should be calculated somehow on th. basis of just how likely a str, ntp. product will yeald a useful pugm. — Th. putting on of ☐ may come later — i.e. suppose we make a rule in simple T.M.'s that only ~~ntps are~~ ngms are, produced by str × ntp., and that to get pugms, one must add ☐ later.

This point should be made clear in th. exposition.

SN perhaps we want $U^{apri}_{str_i \times Ntp_j} = U_{str_i} \times U_{Ntp_j}$

This may conform to a better to what one wants. For the case in which U's are added, the math is simple, but if a str, say, has hy U, then any ntp. mult. by it will give a pgm of hy U apri — in fact, there will be rather poor discrimination. This will have to be looked into to a greater extent.

I think that now I can continue with a few examples of ⊕, then finally, +.

Note in exposition, that in (18) and (19) T.M. hasn't yet learned what ⊕ is. I should make a ruff study to find out just how many examples must be gn. before **T.M.** learns something. Apparently, the first thing that must be done is un learning. This can only be done by

1) Making all the old bad pgms incons.
2) Making sure that any new ultimately wrong pgms that mite be constructed, are rendered incons.

In the case of ⊕, we want to kill $\begin{matrix}1\\1\end{matrix}\begin{matrix}1\\1\end{matrix}$, $\begin{matrix}1\\0\end{matrix}\begin{matrix}0\\1\end{matrix}$ et al — all 8 of them. We will want to keep

$\begin{matrix}1\\1\end{matrix}$, $\begin{matrix}0\\0\end{matrix}$, $\begin{matrix}1\\0\end{matrix}$, $\begin{matrix}0\\1\end{matrix}$, $\boxed{1\,2}$? $\boxed{2|1}$, $\boxed{1|\,|2}$ $\boxed{2|\,|1}$ et al.

all ngms like $\begin{matrix}1\\0\end{matrix}\begin{matrix}\\1\end{matrix}$ should have been incons. long ago. Any ngm that is 3 digits hy, that doesn't contain ⊕, is probably incons. — unless it is like $\begin{matrix}1&0\\1&1\\1&1\end{matrix}$, which is of low aprip.

.40