July 26, 1956 Abraham Robinson

$\forall x\ Px \supset \sim Mx$

For economy in time, memory, do not begin to collect data on a pgm untill at least 1 ~~(or perhaps 2)~~ case occures. This probably a trick that is often (if not always) done in human thot. In fact, probably the pgm is observed first, economy

Gimlet

In thinking about infinite sets, one can think about finite sets of size ~ say 12, and have special rules of manipulation of this set of 12, that have correspondances in operations on infinite sets.

Th. manipolation of ideas about infinite sets by mathians is an example of a large rift between one's model and th. things talked about.

→ then one tries to find ways by which it could have been "constructed" according to a certain fixed set of rules from other useful pgms. In fact, one probably has many "rules of thumb" for th. search procedure, by which one tries to find th. "permissably constructed" pgm of that fits an observed □.

This sounds very much like what is actually done in real human thot.

It wite be well to list these ad-hoc "economy measures" that make for a complex T.M., but do not alter th. essential epist. structure of T.M.

If one defines a set by a machine that one can apply to any object that will tell one whether or not it is in th. set, one can derive any property of th. gp. from 1) knowledge of th. mechanics of th. machine 2) knowledge of just what kinds of objects it is permissable to plug into it.

In th. case of th. idea of integer", list a big bunch of q-elements in which this idea is imp. in making pgms that are useful. Then define th. word in a

way ⇒ it is easy to apply it in all cases of interest.
It may be that one can most conveniently do this by defining
>1 concept. In general, I think that if one has any concept
that one wants T.M. to absorb, one can simply list a bunch of cases
in which one wants it to be used, and then devise an entity
that will do the job. T.M. will be forced to learn it,
if it is the only way to solve the hyp. seq.. The only thing
is to devise the abs. in such a form that T.M.'s discovery
of its use is of by probability
E.g. take the set of all pairs; the idea of "integers" etc.
A kind of application of these ideas is the extension of
$= 11$, $= 121$, $= 12321$ ... to $= 1 \underbrace{2 \ldots 2}_{n} 1$.

It would be well for most people to carry with them, a "medical history" in which all "qualified" M.D.'s who worked on them, would write what they observed and did. (Also x ray dosages). This record would be very useful in autopsies, as well as for any medical work. It would include dental work as well. As it is, one ought to keep reproductions of various x rays at least, to minimize repetition of certain x rays. Trouble is, perhaps x rays negatives would lose much info on reproduction — but I don't think this is ess. tra

---

Hilbert's 17th prob.: on the representability of pos. def. rat. functs as sums of sqrs. of rat. functs. — Has been proved

---

Relevance: Landau has proved a theorem in number theory 1) assuming the hypoth. Riemann hypoth. 2) " " " " is false
However, perhaps if the Riemann hyp. is (false / true) but false, then assuming it is (true / false) implies everything. From this we may conclude that either R's hyp. is true and relevant or R's hyp. is untrue false, and this falsity is relev.

I am assuming that a thg. seq. based on elementary arithmetic does have a reasonable sequence of imp. Abstinits.

In working on these hyer problems, be sure to get several problems to work on simultaneously, to prevent ad-hocicity.

Another imp. Q. would be whether with. a seq. of probs. that I use, th. relevant abss. will follow from each other rather smoothly.

In "carry line removal", 1) list desired abstractions that would be desirable in implimenting th. idea 2) If their is a seq. of abs. forming operations, find some way to abstractly categorize it, so that one can group it with other methods, to find seqns. of operations that have a hy aprip of being useful.

As a preliminary T.M. problem: Make up sequences of instructions for, say IBM 704, → the desired problems are solved. Th. reason why this is a good problem, is that th. 704 contains instructions of th. self-programming type, so that a T.M. can very quickly learn hyer order abstractions.

A possl. tng. sequ.: show T.M. a problem statement - like "add 8 to 13", and then show it th. instruction sequence that solves it. Th. kinds of abss. that T.M. must make, are rather complex, but the problems are presented in a linear order - which may make analysis easier — i.e. th. descriptions of strs is very simple.

IMP: It should be kept in mind, and explained at the outset, that the Uaprips are aprips and nothing more. That almost always they will be useless, but harmless since they will never occure. The normal "economy measure", of not computing ngms* untill they occure, is very signif., and applies to most ngms in a material way. The ambiguity of mult. of strs. by ntps, often very by order becomes unimp., if viewed in this lite

---

Note that consistency of Uaprips. of the same ngm. obtained by different seqs. of operations, is not essential. Uaprips are not and cannot be expected to be that accurate. If a conflict exists a weighted mean may be taken, or a special freq. run may be made for such coincidences.

There is no mention of strsts, ntpsts, pngmsts or ngmsts. in the exposition. There is a mention of cart product of say, ntps and ngms, to obtain ntps, — which is pointless if the ntp. or ngm isn't an ntpst. or ngmst, resp.

It mite be expedient to say that with the introductions of concepts like ngmsts, strsts, etc., that a few important ideas like cart products, $\beta\Sigma\neq$ and $\beta\Pi$, and "functions" are introduced. The "count" of an ntpst, eg. is simply the Σ of the "counts" of the "ntps" that are its components

Three imp. comb. methods. That will not be mentioned, because of no mention of strsts, ngmsts ntpsts, pngmsts.

a) Cart products of sets to form ntpsts

b) functions nptst (α_i, β_j) $(\gamma_k, \delta_\ell) \longrightarrow (\alpha_i, \beta_j, \delta_\ell)$ for $\beta_j = \gamma_k$

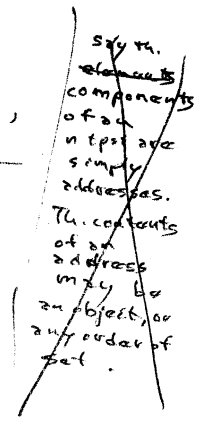
c) Boolean Sums, products of sets to form new sets. (perhaps no $\overline{\in}$:)

d) Cartesian and other division types that have to do with sets.

A way to implement Th. sets of sets of sets....

idea: Make every ngmst also a permissable ngm. (or, more generally every <(pngmst or ngmst)> also a <pngm or ngm>) This makes it possl. to have ntples that are relations betw. numbers. E.g. Consider Th. ntpst $\gamma \equiv \alpha_i \overset{c}{\times} \beta_j$, where α_i [i = 1 / ∞] is Th. set of all objects. — So γ_2 is Th. set of all pairs. Let γ_i be Th. set of all i tuples. — Then.

Th. ntpst $\delta \equiv (\gamma_i, \gamma_{i+1})$ corresponds to Th. relation "One greater than".



To impliment this: make every component of any ntp., be an address. Th. content of an address may be any* object — ie. an ngm, ntpst, str. ... etc. Each address will have a set of instructions that tell whether an "object" is a fit member of the set that this address defines.

Still, Th. process of finding even one pngmst to fit

a □, may be quite difficult. One can't very well go thru ~~the entire set~~ every pugmst that one has.

There will, however, be certain clues available, to implement Th. search.

It may be nec. to make a hyer order T.M. to search, but I think Th. problem is far smaller than Th. original one. Also it is more well-defined —(tho it may be a "well defined problem of Th. second kind"). It may be possl. to use some rather simple, fixed, searching schemes for this 2nd order T.M. ~~#~~ A T.M. with even ~~some~~ a simple,

These general ideas on How T.M. work is to continue, should be explained at end of introductory opus. Also explain ~~utps~~ ngms ts, etc.

→ fixed, 2nd order T.M., may be clever enuf to be able to improve, or be its own, 2nd order T.M. There is little danger of "oscillation" since this ~~#~~ 2nd order T.M. has a fairly well defined problem.

Th. introduction of hyer order sets may make many of Th. specific ~~iteration of~~ "combination methods" that were introduced ad-hock, unec. — i.e. these "comb methods" may be automatically introduced via Th. proper typ. seq.

It would be very expedient - almost necessary that T.M. be able to improve its own methods of creating new abss..

It would be very encouraging if I were able to show how Th. introduction of ~~Th. set of sets~~ hyer order sets can in even some cases, give

rise to what seem to be new methods of creating new abss. from old, and/or for evaluating their prediction parameters.

An essentially new method of creating pugmsts from utpst is by card. division: pugmst = utpst ngm, utpst, or pugmist

The top utpsts may be produced by "functions", or BT or BE, producing it by ∈ or permutation wouldn't give any new pugmsts that we couldn't get without this operation.

Bottom utpst can have been gotten in any way

Not too sure about these remarks.

Since one can't (I don't think) mult a str. by in utps if all of th. components of th. utp are not ngms or pugms, one can only use utps that have sets as components in this way, I

What one would do, is state various kinds of inductive conclusions or indicate just what sets are desired to be close to what, in terms of this method of dealing with sets of sets. See just what "methods of constructing new pugmsts from old" they imply.

So : th. present program: In main text of detailed exposition, do not go up to utpsts, etc.

However in item 9) of page 31)

mention utpsts, etc.

" several new combination and xfmn methods of 36)

" how sets of sets is imp. and how it can be implimented.

— by sets of definition instructions in each box.

on th. search problem — how it may need a hyperorder

T.M.

37 Consider th. foll. kind of induction: we note that =1, then =S1, then =SS1 are all "good" — to induce that =SSS1 is probably "good".