Dart + γ517 γ533 – γ558 + Merge ~~(summary)~~

Preparatory to t. SOS paper, c May 22 (±1d), 1962

(TNG seqs. for Mechanized Induction)

ZTB:

(There is a review of Dart ~ γ 433 – 467)

T. first part of t. review is from γ 534 – 542

γ546 – γ558 are calcs and genl. discn. preparatory to writing t. SOS paper

~~γ533 seems to be missing, two referred to by and by γ534 w Mar 7, 62 is on word; is on ZTB 14 (c-line)~~

W. Mar 7, 62

Dart
483.40 ← perhaps

~~[illegible]~~? (517)

.01 : On Th. Coding of <u>ZTB 141</u>, and Th. Coding of <u>WPSG's</u>:

Th. prob. of ZTB 141 was to code by devising defs. of ngms. Along with this, was Th. constraint that if α was defined as α ≡ ab then b could <u>never</u> follow a in ~~th~~ Th. intermediate code — i.e. ab would, if it existed, always be coded as α, or else (a or b) would have been part of some other defined symbol.

Th. effect was to reduce Th. ~~cost~~ rcost of symbols following a, because it is known that b cannot be one of them. Without this constraint, it ~~[illegible]~~ was felt necy. that $f_a \cdot f_b > e f_{ab}$ befor it was expedient to define $\alpha \equiv ab$. <u>With</u> this constraint $f_a \cdot f_b \neq f_{ab}$ was adequate, if Th. sample was large. (Note ~~[illegible]~~ that this made it useful to define $\alpha \equiv ab$ even if $f_{ab}=0$. (or even <u>in particular</u> when $f_{ab}=0$).

~~My now it~~ Th. trouble with such constraints, is that if one has a bunch of ngms of various lengths, and one is coding a corpus, it would be necy to define a large no. of "precidence rules" e.g. if $\alpha \equiv ab$ and $\beta \equiv bc$ then abc could be coded as either αc or $a\beta$. Th. "precidence rules" would have to make such decisions. I think that I did do a lot of work (even after β1300) on precidence rules, and I may have figured out a good way to deal with them.

Anyway: Th. ~~[illegible]~~ idea now, is to not have <u>any</u> precidence rules, but to consider <u>all</u> <u>new</u> <u>codings</u> that are made possl. by a defn. of some ngm. — so if we define $\alpha \equiv ab$, then abc can be now coded <u>both</u> as abc <u>and</u> αc. Then we will have to see what Th. conditions are, for it to be <u>much</u> worth while to define $\alpha \equiv ab$, since <u>any</u> defn. will ↓ rcost of coding to <u>some</u> extent. Computing Th. total change in cost, will be diflt, since there are <u>many</u> possl. new codings (of Th. corpus) assoc. with each new defn, and each such coding usually has a difrnt ~~[illegible]~~ cost.

Th. <u>main</u> <u>trouble</u> with Th. idea of Th. previous ¶: That if $f_{ab}=0$, it would <u>seem</u> that this new method would <u>always</u> give a low ↑ in pcost for this "of coding", for this new defn. — i.e. that it would always be "worth <u>very</u> little". Since defining $\alpha \equiv ab$ when $f_{ab}=0$ <u>does</u> seem <u>very</u> worth while (i.e. to <u>know</u> that b can't follow a), this is an impt. criticism → <u>533.30 spec</u>

But

533.01

W Apr 11, 62

Dart

533

.01:517.40 : A rather old idea — a review of it: A Mt. Carl. is a certain kind of MTM:

We start out with a trn. seq., and a fixed method to d... th. (Yes–No) [∴"MTM"] correctness and aprips. of th. Mt. Car. chosen trial solns. to th problems. Actually, we have 2 ~~MT's~~ TM's that are growing: (1) consists of, at a gn. time, a soln. string for all th. probs. in th. trn. seq. thus far. Also included are methods of coding a/o th. resultant methods of devising aprips for th. new Mt.C. string trial. These methods could ~~be included~~ included in th. soln. strings, but may be more convieniently stored separately.

(2) Th. T.M. that does th. searching. It tries various techniques to reduce search time, and its pgm string at any particular time, dcrbs. its methods thus far. New trials are made in attempts to speed up searches. Note that in a certain sense, the data in 2's string is data on regularities in ~~th.~~ 1's input corpus. When viewed in this more unified way, there may be a better way to unify 1 and 2, and reenforce them in a more unified way, rather than use separate reenftmnt, (th. sept. reenft. "factors" th. "reenft." and allocates responsibility — ∴ it can be more effective at first —

.29 latter, th. non-el. method should be used. → 550.01 spac

.30 : 532.40: We can, in our code for WPSG's / probly get th. same affects as ⟨th. constraints of defining $\alpha \equiv ab$ in a simple 2 gm defn. gramm.⟩ by making / similar constraints assoc. with each defn. in th. WPSG. e.g., if we define $\alpha \rightarrow ab, cd$; then whenever ab or cd appear they must be coded as α.

This seems like a very impt idea, and it may make PSG discy by hill-climbing much easier, since th. p costs are now probly more "correct".

Th. idea must, hvr, be expanded and exactified. Note that th. simple binary subsn. gramm w.o. loops or branches is a special case of WPSG, and so any modifications made in WPSG must be ∋ they give th. proper results in this degenerate case. → 540.25 spac 539.01 spac. 534.01

Sat Ap 14, 62

TMG

Dart (534)

~ Sept 196

.01 : 533.40 : What I want to do now, is review Th. rather recent Dart wor (449 — 467, and stuff in Dart just preceding 449), preparatory to writing a paper for Th. Chicago SOS conf.

Th. immediate folg. will be a set of isolated notes — they are ideas sugg. my reviewing of This work.

.06 from 460.04 : This ~~m~~ method of recognizing Th. "new situation" (i.e. building a special op. for that) than dealing with Th. new situation in a special ad hoch (at first) way, seems similar to ~~one~~ a common method in organic evolution — in which, in order to solve a new kind of prob., a new structure evolves. Th. old structure is, hvr., kept, and special devices are evolved to deal with it — by suppressing it, ~~by~~ for certain problem types, by xfmg. inputs befor it gets them a/o xfmg its outputs, with special xfmns. In general, in org. evol., it is always "easier" to modify an old operator, rather than create a new one — even tho Th. all-over complexity (i.e. rcost) of Th modified op. may be far > that of an entirely new one that may do Th. problem better. This is a natural consequence of Th. fact that hill-climbing (on a hill of slowly changing shape) is used to solve Th. ~~e~~ problem in org. evol. [or perhaps it is more correct to say that hill climbing and "peak ~~tracking~~ ~~~~" tracking on a hill of varying shape, are used. "Peak ~~hugging~~ tracking" means staying at a local maximum as Th. hill changes shape around one. A more picturesque ~~way~~ feature — that Th. ~~k~~ Peak being ~~hugged~~ tracked is "above water", and as Th. hill ~~mov~~ changes shape Th. positions (if any) of Th "above H2O" parts, move around. Org. evol. must then, "track" These peaks in order to "stay alive"].

Def

If org. evol. is viewed as hill climbing, one visualizes Th. density of trials being reenforced in Th. upward direction. In peak tracking, hvr., Th. trials that go into Th. H2O are distroyed.

Sat Ap 14, 62

Dart

535

.01!534.40!

A kind of "Language" that may be "universal", is th. corr[ect] coding method, but using symbols that symbolize operators. Ano[ther] way to get "universality", would be to allow more complex ways to combin[e] old successful "properties" to make new trial "properties".

---

462 summarizes with a set of subs. rules (and their notation) that seem adequate for arith. evaln.

A more exact formulation of 462.03 – .08 — using th. coding methods I've devised for WPSG's in th. "IC paper"

~~AαBγ→MαABφ~~

$\beta A \alpha B \gamma \to BM \alpha AB\gamma\phi$

$A\delta B \to M\delta AB\phi$

$(A) \to A\phi$

$\alpha \equiv +, -$

$\beta \equiv \alpha, ($

$\gamma \equiv \alpha,)$

$\delta \equiv /, \times$

these ≡ signs are unnecy. Th. usual way to treat this part, is to list these 4 defns first, as $+-, \alpha(, \alpha), /\times$ — Then follow with this part, using suitable subscripts. The "→" sign is treated as a regular symbol, like (, or A, etc.

I'm not sure about just how to deal with th. linkage lines. There was much discussn on this Q — of how best to indicate CDS (ont. ext. & substitn.)

Apart from th. CDS part, th. feature about th. above coding notation that makes it difrnt. from th. previous PSG/notation, is that all legal subsitns are used, and order is irrelevant. This means that we do not follow this "grammar" part with a seq. of nos. indicating "which choices were made". — Otherwise, e.g. it mite be best to condense β and γ into $\beta \equiv \alpha, (,)$, and use th. choice notations to tell when (or) was correct.

Also, perhaps th. A's and B's should be viewed as "linked" symbols.

i.e. $\beta A \alpha A \gamma \to \beta M \alpha A A \gamma$ Th. machine, in a sense, interprets

536.01

Wed Ap 18, 62

Dart

TM8

536

.01 535.40: A and B to be "any numbers" — and this could be viewed as a rules form $A \equiv 1, 2, 3 \ldots$

---

1 linear eq. in 1 unk. is treated a bit on 433.30 ff.

I think a common method of humans' working probs. is: ① Take soln. string for probs up to present one, and try it on new prob. If it works, fine; if not, try to modify it by "adding on" features so that it both continues to solve th. old probs and solves th. new. If this doesn't work, take th. soln. string that worked up to th. prob. befor th. last, and try to add onto, and/or modify it so that all probs in th. trng. seq. are solved. Continue this way, discarding only th. most recent parts of th. soln. string as time goes on.

.15 If th. soln. string is abacbeaac, then we can just have th. string grow on th. rt. as new probs appear — but it would drop off more and more of th. rt. end befor making new trial additions, if trouble was experienced.

Note that th. presence of a symbol seq at th. rt. end does not mean that this is closely related to th. machine doing that thing "last". Th. symbol seq. .15 is usually a set of instructions to write a pgm. for th. machine to actually do. Or it may be a pgm to write a pgm. to write pgm... etc.

This idea is meant to be something like org. evoln. — but I think that org. evol. usually doesn't discard things — it just "adds on", and may add on devices to inhibit some of th. old devices.

.30 Perhaps I could devise a routine that would just "add on" to th. soln. string, w.o. discarding ... just as org. evol. does. This could be done by having a certain symbols that gives th. effect of erasing th. last symbol of th. previous soln. string. This symbol would require a certain frequency just like any other symbol. In decoding th. soln. string, this symbol type would be enacted befor all other types. This would reduce th. string length.

Note that this freq. would be $\propto$ past effectiveness of th. "lopping off" xform.

After th. "lopping off" symbols, one mite still want to preserve th. symbol frequencies that include th. ones erased. This makes it more likely that in th. new trials, one will add on something like th. erased symbols, so that th. probability of finding a soln. to both th. last prob and th. present problem is somewhat larger than th. product of their individual probabilities — i.e. one has some "hints" on how to solve th. old one.

537.01

Th Ap 19. 62

Dart → (537)

.01: 536.40: On 433.14 - 435.30 are 4 possl. forms of soln. to this TM. Following 433.14 and up to at least 438.20 are attempts to unify these since each seems to have desirable properties. My impression that some genl. success in this unification was obtained.

440: Th. general approach that I want to stick to. - i.e. explication of my own problem solving methods.

There is some uncertainty in my mind as to just what th. M_i and L_i are. I think M_i is th. soln. string for TM_i. L_i is perhaps some routine to discover M_i. See 454.30 for some examples of M_i, G_3, L_3.
" 406.35 ff for some discn. of M_i, G_3 and L_3

455.25 is an optimistic note! — that "within 3 days" I should have th. outline of a fairly good "little TM"! Also discusses ~~impt. ...~~ genl. approach and gives refs. to impt. work.

→ Well, from 454.30 it is clear that: M_i are th. machines or derns. of machines that are trial solns. of th. tng. seq. probs. For each point in th. tng. seq. we have a set of M_i's that were tried at that pt. — some of which were successful. As machines, th. inputs to th. M_i are various problems <like $(13.1 + 4.3)/91.2 - 8.01$> and th. outputs are attempted solns.

G_3 is a "creative Mt. Carlo program". It creates s's in th. stoch. lang., L_3, and these s's are th. trial machine derns, M_i. There is, hvr., some confusion here on 454.30, since there G_3 also is a machine that analyses previous M_i trials, the modes of operation that they used, and their computation times — and from this obtains a trial M_i.
see 463.10 to 467.09 } for more about G_3 —

446.20 has a list of impt. heurs for use with th. discovery of various sets of subsn. rules

.30 My impression is that th. impt. pts. of discn. in this seq. of Dart were: (1) 4 possl. forms of soln. to Dart TM (433.14-435.30) (2) Devising a good notation for CDS a/o non-CDC (≡ th. first 2 types of soln. of (1).) (3) Some discn. of heurs and th. modes of operation of th. higher order TM's (see 454.30 for mach. heurs; see 446.20 for heurs)
466.12 462.01-05 simple heurs. (" 449.25 for methodology ← impt.

From 449.01 to ≈455.36 is a lot of good discn. on just what direction to go in, and what work has been done, and what work needs to be done. 452.10ff is particularly good - also 453.02.

spec. 542.01

538.01

F Ap 20, 62 TMY

Dart

537.40 spec

.01: 541.40

542.01

2 (General review, cont.)

461.30 – 462.01 has a definite statement of how Th. Thing is to work. — This may be adequate for SOS paper. What I had in mind for Th. paper was something like a fixed G_3 (like 466.08 – .11 using symbol freqs, ngm defs and perhaps factoring into ngmsts — and few if any, other heurs. Th. Heurs of 463.01 – .05 would be interesting to generalize, but I don't know whether I really need them for Th. SOS paper. These heurs really should be looked into more, because Th. concepts involved seem like impt. ones. (E.g. Th. "if α has ever occured, β cannot follow unless γ comes first" type of thing — this is a "finite state" type of condition — i.e. we can create this type of sequence by a sequencial finite state machine, that is put into a certain state S_1 when α occures, and comes out of that state when γ occures Th. machine can't produce β when its in state S_1.

From 467.10 to 483.40 Th. notation for PSG coding is developed and much work on PSG discy. is done. My impression is that it starts out with something close to non-dim PSG, and then narrows down to ordinary PSG's.

From 467.13, one gets Th. idea that being able to define ~~[illegible]~~ ngmsts like $\gamma \equiv$ (anything that is <u>not</u> in ngmst α), is very useful. In general, ~~when~~ if Th. ngmst α is useful, it is not clear whether it is more useful to define α, or Th. complement of α, or both! — Th. complement of α has to be defined with care, since it isn't clear as to just what Th. whole universe of symbols is, that one is complementing with resp. to! — Tho perhaps in 467.13, it is clear.

---

411 to 414 have some perhaps impt. ideas on general methodology Also they refer to Plans 394 – 401 (and Plans B1160!).

Sun Ap 22, 62 T M 8.

Dart (546)

.01: 542.40 : Present plans for SOS paper: Just do arith. evalu ...

Use "soln." of 462.31.–.36 (with possible "shortening" of 463.06).

First compute pcost of entire soln. (very approxtly).

Then use simple trp. seq. and see how small Δ pcosts can be.

Then look into various local maxima that are dead ends — see [how] hard it is to get out of them.

.06 Note th. form of "soln". It is in th. form of a bunch of unordered CDS's. Hvr., a special kind of PSG notation is used to code this set of CDS's. Th. fact that they are unordered is impt., since it makes it possl. for use to code them using only a set of "pure gramm. rules" w.o. a list of "choices" to be made. As such, this is essentially a new grammar type that we have devised for certain unordered sets of objects.

.13

Let us make th. convention (protem) that th. M_i's will be th. actual unordered set of CDS's. G_3, wich is th. device that creates trial M_i's, has built into it th. idea of th. grammar type of .06–.13.

— Also, G_3 may have a few heurs for devising good trial<factor ngmsts>'s.

But, at first, I want G_3 to be fixed, and see how far th. TM can go.

Later, I may want to allow G_3 to grow, by having a hyar order TM make trial modifications of it — but I'm not sure that I want to go into this particular TM model that far.

Let T_0 be th. machine that interprets G_3's outputs [G_3 is a machine, ... in th. folg. discn.]. Th. input to T_0 consists of 2 parts ① a set of CDS's ② A problem, (like $3+(7+8/2)$). T_0's output consists of a randomly ordered applicn. of th. CDS's to th. input string, until no more of th. CDS's will work. Then T_0 stops and presents th. result as "Output".

Th. legal outputs of G_3, are strings with th. folg. alphabet: end of gramm. stop is impt. stop betw. CDS's.

.37 (,), +, –, ×, /, M, φ, →, Integers, A. this differs from 462.335, in which T_0 had to interprete a dcrn. using a gramm. of th. type .06–.13). The integers are constrained somewhat, but th. rest of th. symbols can be written in any order w.o. constraints, as "trial solns". [actually, the constraint on integers (they are to be used only as 547.01

* Sun Ap 22, 62 Tang

Darty (547)

.01 :546.40 : "subscripts" for th. symbols, A) is not absolutely
I could just give all integers, I, th. aprip $\frac{\Sigma}{I}$, where Σ
normalizing factor $\propto (\ln I_{max})^{-1}$, where I_{max} is th. largest integer built into G_3
thus far. Hvr., th. constraints on integers will be a havr./, — if I want
to use it).

Well: O.K.: Lets start: **First problem:** A + B

This is presented to T_0 in a form in which A and B are random nos.
• Hvr., T_0 knows that A and B are nos., and treats them in th. special
ways that nos. are to be treated when applying th. CDS's.
When T_0 is gn. a problem, it is also gn. <u>a</u> soln. or <u>the</u> soln.
If there is > 1 possl. soln., this will cause trouble. Perhaps we would
have to include <u>all</u> possl. solns., so G_3 could tell when any
proposed M_i was acceptable.
Th. idea of having ~~th.~~ T_0's inputs problems in th. form
of random nos. rather than literal expressions, is that if we
use random nos., we can readily tell (with almost certainty) rather quickly,
whether a proposed M_i is correct.

Some possl. solns. to A+B : ① A+B → ~~AAAAA~~ M+AB (polish notation)
Actually, I think M+AB or M+BA <u>has</u> to be on th. rt. side of →.
.25 Th. notation is A1+A2→M+A1A2 or A1+A2→M+A2A1
or $A\binom{1}{2} + A\binom{2}{1} \to M + A\binom{1}{2} * A\binom{2}{1}$. Th. actual integers used is fairly
.27 arby. we can, hvr, make th. convention that th. first integer used
<u>must</u> be 1. The next <u>must</u> be 1 or 2. Th. next can be either
either an integer used befor, or be 1 greater than th. largest
used thus far.
So th. only solns. possl. with these constraints are

There are 10 symbol types (not including integers) — so th.
only integers are legal here

Pcosts of .25 are

	A	1	+	A	2	→	M	+	A	1	A	2
							7	8	9	10	11	12
10 symb + integers	$\frac{1}{10}$	$\frac{1}{11}$	$\frac{1}{11}$	$\frac{2}{12}$	$\frac{1}{3}$	$\frac{1}{13}$	$\frac{1}{14}$	$\frac{2}{15}$	$\frac{3}{16}$	$\frac{2}{4}$	$\frac{4}{17}$	$\frac{2}{5}$
	1	2	3	4	5	6						
non-integer symbols	1		2	3		4	5	6	7		8	

$\frac{1}{10} \cdot \frac{1}{11} \cdot \frac{1}{6} \cdot \frac{1}{3} \cdot \frac{1}{13} \cdot \frac{1}{14} \cdot \frac{2}{15} \cdot \frac{3}{16} \cdot \frac{1}{2} \cdot \frac{4}{17} \cdot \frac{2}{5} \approx \frac{1}{10} \cdot \frac{1}{10} \cdot \frac{1}{6} \cdot \frac{1}{3} \cdot \frac{1}{10} \frac{1}{10} (1-\frac{1}{10})(1-\frac{3}{10}) \cdot \frac{2}{3} \cdot \frac{1}{5} \cdot \frac{2}{3} \cdot \frac{1}{5} \frac{1}{2} \cancel{\frac{1}{4}} \cdot \frac{2}{5}$

Sun Ap 22, 62 TMY

Dart

.01: 547.40: $\frac{1}{10}$ $\frac{1}{11}$ $\frac{1}{6}$ $\frac{1}{3}$ $\frac{1}{13}$ $\frac{1}{14}$ $\frac{2}{15}$ $\frac{3}{16}$ $\frac{1}{2}$ $\frac{4}{17}$ $\frac{2}{5}$ — mult ... 2 poss...

$\frac{1}{10}$ $\frac{1}{10}$ $\frac{1}{6}$ $\frac{1}{3}$ $\frac{1}{10}$ $\frac{1}{10}$ $\frac{2}{10}$ $\frac{1}{5}$ $\frac{1}{2}$ $\frac{2}{5}$

$(1-\frac{1}{10})$ $\left(\frac{10}{14}\right)^3$

$\left(\frac{1}{10}\right)^6$ $\left(\frac{1}{9}\cdot\frac{9}{10}\right)\cdot\frac{1}{5}$ $= 2\times10^{-8}$ = pcost.

It would be easiest to compute pcosts by writing th. /expressions w.o. th. CDS in them — I.e. compute th. pcosts of th. integers separately.

This means that we can get ^exact^ pcost calcs. from simply noting how many times each symbol has occurred — as in a Bernoulli seq.

Next, give th. problem A − B

There is only 1 soln: $\phi A1 - A2 \to M - A1A2$

w.o. indices: $\phi A - A \to M - AA$.

~~total pcost with... for th.~~

First check on pcost of th. A + B → M + AB solns:

It should be $2\times\frac{4!\,2!\times 9!}{(10+8-1)!}\times\underbrace{\frac{1}{3}\times\frac{1}{2}\times\frac{2}{5}}_{\text{integers}}$ (2 × ← 2 solns)

$\frac{48\times9!}{17!}\times\frac{2}{15}\approx\frac{6}{17!}\times9! = \frac{6\times10^5\times.28}{.27\times10^{14}} = \boxed{6\times10^{-9}}$ pcost (off by factor of 3).

.20

for: 9 symbols Δpcost: $\frac{17!\;\;8!\;\;2\;\;2}{(17+9)!\;\;4!}$ (A, −, M)

A: 4 times
−: 2 times
M: 2 "

integers "+" 1 2 1 2 "−" 1 2 1 2

.12 $\frac{1}{3}$ $\frac{2}{4}$

woops! I really need ϕ, th. "end of gramm. stop".

It mults th. pcost of th. "+" soln. by $\frac{10}{18}\times\frac{1}{19}\approx\frac{1}{40}$ | so $\frac{6}{4}\times10^{-10}$

.30 it is $2\times\frac{4!\,2!\,10!}{(10+9-1)!}\times\underbrace{\frac{1}{3}\times\frac{1}{2}\times\frac{2}{5}}_{\frac{1}{15}}$ $= 1.5\times10^{-10}$ = pcost.

However, in obtaining th. original pcost of th. "−" soln., we have to lop off ϕ from th. old soln., and adjoin ϕ instead.

Well, I'm not so sure that its much worth while to go into th. exact pcosts of th. solns. of various probs. in th. trg. seq., — that is, not just using Barn. seq. coding. I have th. pcost of th. soln. A+B→M+AB, starting with "nothing" in th. way of freqs. It mite be worth

549.01

Mon Ap 23, 62 TMX

Dart

for this same so...

.01 : 548.40 : while to contrast this pcost / with that which ... after TM had learned ~~...~~ th. (exp... soln. of 462.31 – .36. I will write these w.o. th indices:

.05 $\begin{pmatrix}+\\-\\(\\)\end{pmatrix} A \begin{pmatrix}+\\-\\(\end{pmatrix} A \begin{pmatrix}+\\-\\(\\)\end{pmatrix} \rightarrow \begin{pmatrix}+\\-\\(\\)\end{pmatrix} M \begin{pmatrix}+\\-\end{pmatrix} AA \begin{pmatrix}+\\)\\(\end{pmatrix} \phi$ | $A \begin{pmatrix}/\\x\end{pmatrix} A \rightarrow M \begin{pmatrix}/\\x\end{pmatrix} AA\phi$

2 2 1 rule

$\{ 4 \times 2 \times 4$ ~~$4 \times 2 \times 4$~~

32 rules. (actually 50, because n should be considered) | 4 rules. I only see (3) — only (2) of which will occur

$A \times B \Rightarrow M \times AB$

$A \times B \rightarrow M \times BA$ and $A/B \rightarrow M/AB$.

No. of occurrences of various symbols:

.15 (	~~3~~ 32 + 1 = ~~33~~ 33	33	
)	~~3~~ 32 + 1 = ~~3~~ ~~17~~ 33	33	
M	~~3~~ 32 + 4 = 36	36	
A	$4 \times 32 + 4 \times 4 + 2 = 146$ (128, 16, 2, 146)	146	
.20 ϕ	32 + 4 + 1 = 37	37	
x	4 = 4	4	
/	4 = 4	4	
+	~~..~~ (32 + 32) = 64	64	
–	= 64	64	
$\rightarrow$	32 + 4 + 1 = 37	37	
		$\Sigma = 458$	

10 symbol types. (forgot ϕ ... and "n") 12

So, th. pcost of th. non-index part of $A + B \rightarrow M + AB\phi$ is:

$(P_A)^4 \; P_+^2 \; P_M \; P_\rightarrow \; P_\phi$

$\left(\frac{147}{468}\right)^4 \left(\frac{65}{468}\right)^2 \left(\frac{37}{468}\right)^3 \left(\frac{37}{468}\right)\left(\frac{37}{468}\right)$

$\frac{1}{(3.2)^4} \; \frac{1}{(7.2)^2} \; \frac{1}{(12.6)^3} = \frac{1}{105} \cdot \frac{1}{52} \cdot \frac{1}{2000}$

$= 10^{-7}$

for 548.20 we got a pcost of pcost of indices.

$6 \times 10^{-9} \times \frac{15}{2} = 9 \times 10^{-8} \approx 5 \times 10^{-8}$ ≥ for 2 solns.

th. pcost of th. non-index part!

$= \frac{1}{2} \times 10^{-7}$.

no! — There is an factor of $\frac{1}{40}$ due to $\oplus$ (rather than ϕ) so its really $\frac{1}{8} \times 10^{-8} = 1.2 \times 10^{-9}$

recalc. of old pcost.

$A \quad + \quad M \rightarrow \phi$

.30 $\frac{4!}{14^4} \times \frac{2}{14^2} \times \frac{1}{(14)^3} = \frac{48}{14^9} = \frac{\overset{3}{\cancel{48}}}{\cancel{16} \times 1.4 \times 10^9} = 2.1 \times 10^{-9}$

$14^9 = 1.4^8 \times 1.4 \times 10^9 = 1.4 \times 2^4 \times 10^9$
1.4×16 "

$(1.45)^9$ | $\frac{1.4}{1.45} \times$ | ~~2.4~~ $\frac{5}{145} = \frac{1}{30}$

$(1\frac{1}{30})^9 = e^{\frac{9}{30}} = e^{.3} = 1.35$

So if I used a mean position of 14.5 rather than 14, I would ↓ total pcost by factor of only $e^{.3} = 1.35$.

This is wrong anyway: Th. end stop $\oplus$ is needed, instead of ϕ.

Th. pcost of $\oplus$ is $\frac{1}{468}$ — i.e. $\frac{1}{37}$ × pcost of ϕ.

So instead of 10^{-7} its 2.5×10^{-9}

550.01

spec 552.01

W. Ap 25, 62 TMY

Dart

.01 : 551.40 549.40 spec → : I better go thru this calcu. all over:

first: th. exact pcost for $A + B \to M + AB\phi$ w.o. th. pcost of th. ind

so — to coda → $A + A \to M + AA\phi$
1 2 3 4 5 6 7 8 9

There are 11 legal symbols (see 546.37), and 9 to be coded.

.05 $\overset{4}{A} \overset{2}{+} \overset{1}{M} \overset{1}{\to} \overset{1}{\phi}$ → total = 9

$$\frac{(11-1)!}{(11+9-1)!} \cdot 4! \cdot 2! = \frac{10! \quad 4! \; 2!}{19!} = \frac{3.6 \times 10^6}{1.2 \times 10^{17}} \times 48 = \boxed{1.5 \times 10^{-9}} \approx \beta$$

omit fact that there are 2 possl. codes —
$A + B \to M + AB$ and
$A + B \to M + BA$

---

using th. data of 549.15, but adding in th. symbol, ϕ:

$\Sigma = 458$ symbols (not 459); + 11 = 469

pcost of after all 37 rules of 549.05 have been learned.

$\overset{4}{A} \overset{2}{+} \overset{1}{M} \overset{1}{\to} \overset{1}{\phi}$

$$\left(\frac{14\cancel{0}}{4\cancel{10}}\right)^4 \left(\frac{65}{4\cancel{70}}\right)^2 \left(\frac{37}{4\cancel{10}}\right)\left(\frac{37}{4\cancel{10}}\right) \cancel{\frac{36}{410}}\; \frac{1}{470}$$

.20
$$\left(\frac{147}{470}\right)^4 \left(\frac{65}{470}\right)^2 \left(\frac{37}{470}\right)\left(\frac{37}{470}\right) \frac{1}{470} \approx \frac{10^{-7}}{37} \approx 2.5 \times 10^{-9} \quad \alpha$$
via 549.20

— which is practically th. same pcost as befor!

lets compare pcosts of individual symbols! first let us approximate th. result of .05 by a simpler trick like 549.30:

see 553.01 for reasons why this pcost mite be 37 times larger.

$\frac{4}{A} \quad \frac{2}{+} \quad \overset{1}{M} \overset{1}{\to} \overset{1}{\phi}$

.25
$$\frac{4!}{(15.5)^4} \; \frac{2!}{(15.5)^2} \; \frac{1}{(15.5)^3} = \frac{48}{(15.5)^9}$$

$$= \frac{48}{50 \times 10^9} \approx 10^{-9} \quad \beta$$

This isn't too far from th. 1.5×10^{-9} of .05.

$(15.5)^3 = 3.7 \times 10^3$
$(3.7)^3 = 50$

$(15.5)^2 = 280$
$(\text{"})^4 = 7.8 \times 10^4$
$\frac{4!}{(15.5)^4} = \frac{24}{7.8 \times 10^4} = 3.1 \times 10^{-4}$

$\frac{2}{280} = \frac{1}{140} = 7 \times 10^{-3}$
$\frac{37}{470} = 7.9 \times 10^{-2}$
$\frac{1}{15.5} =$

pcosts	4 A	2 +	M	→	ϕ	
via .20	10^{-2}	2×10^{-2}	7.9×10^{-2}	7.9×10^{-2}	2.1×10^{-3}	α
via .25	3.1×10^{-4}	7×10^{-3}	6.5×10^{-2}	6.5×10^{-2}	6.5×10^{-2}	β
	α 30 times as big	α 3 times as big	~ same	~ same	β 30 times as big	

So we end up with α about 3 times as big as β.

Th. thing is — they are about =. Th. ↑ in pcost of th. A's (via β) is offset by th. ↓ in pcost of th. ϕ in β.

W Ap 25, 62 YMγ

Sort

.01: 552.40: Actually, I suspect that th. pcost of ϕ for α is not so [≐ 552.20]
because if ϕ does not occur at th. end — and, instead, ϕ occurs, and is followed by various gibberish, this "soln." will be acceptable. Even hvr., th. symbol ϕ must occur, to terminate th. mess.

So — my impression, then, is that because of this, α should be about 37 times as probable as indicated — so it is perhaps 50 times as probable as β — since for both situations, th. pcost of th. "last" symbol will be about ~~[illegible]~~ th. same.

For β, its th. pcost of ϕ (which is 6.5×10^{-2}) — and for α, its th. pcost of ϕ (which is $\approx \frac{37}{470} \approx 7.9\times10^{-2}$). [These estimates are all rather off — I ommited ϕ, ϕ and a in th. rules of 549.05]

Well! — its clear that th. pcosts of th. soln. , $A+B \to M+AB$ is quite low ($<10^{-9}$ after one includes th. pcosts of indices for A and B) — but not yet out of th. range of computers to be expected in th. next 5 yrs. — And probably ~~well~~ within th. range of ‖ computers presently constructable. — (But this is fairly irrelevant.)

Th. pcosts of rules with real context dependence, would be signifly ~~[illegible]~~ smaller. (say $\times10^{-4}$).

If we give th. machine A − B to learn, and he gets it, he will be able to do $A+B-C+D+E-F-H$, etc.

Next, we could teach parenthesis: e.g. $A-(B+C)$.

He will first rewrite this as $A-(D)$; $(D \equiv M+BC)$

Hvr. — say we teach him (A) first.

Th. soln. is $(A)\to A$. Its soln. is short and ∴ of relatively hy pcost. There is a "dead-end" here: e.g. $A)\to A\,\Phi$ ($A\to A\phi$ would also solve (A), but wouldn't extrap. properly. It is, hvr, of much lower pcost than $(A)\to A\phi$ 10 symbols v.s. 6. A pcost factor of $\sim 10^4$.

So this could then do $A+B-((C-D)+(E+F-G))-(H+I)$

Next, $A\times B \to M\times AB$; to do problems like $A\times B$.

— But it couldn't do other probs like $A+B\times C$. It could do $A\times(C+D)$

F Ap 27, 62 TMδ

Dart

.01'553.40 : Well: Consider that $A \times B$ and A/B were le[arned]
— Then $A\binom{+}{-}B$. — rather than th. ~~pr~~ present order. C[onsider]
th. diffty in learning, in th. 2 cases.

If $A(\mp)B$ is learned first, then after $A \times B$ is learned
we have to unlearn and revise th. $A \mp B \rightarrow M \mp AB$ rules.

~~If th~~ I don't know just how this revision can be managed!

As was noted before, this revision can be done either by CDS
or by ordering th. rules.

Say ~~he~~ he learns $A \times B$, and A/B null symbol.

Then, give $A+B$. Th. soln. will be $\binom{+}{n}A+B\binom{+}{n} \rightarrow \binom{+}{n}M+AB\binom{+}{n}$

— There are 4 rules ~~of the type~~ included here

Then, if we give (A), with soln $(A) \rightarrow A$, we will have
to add more rules to get $\begin{pmatrix}(\\)\\+\\n\end{pmatrix}A+B\begin{pmatrix}(\\)\\+\\n\end{pmatrix} \rightarrow \begin{pmatrix}(\\)\\+\\n\end{pmatrix}M+AB\begin{pmatrix}(\\)\\+\\n\end{pmatrix}$

.20 — which are 16 rules —
i.e. 12 more.

SN Note th. inclusion of th. new symbol, n —
This is a null symbol ~~AAAAAA~~ — it denotes th. end of a ~~pro~~
~~senten~~ string at th. present time — we may want to change
its meaning later.

Actually, I think th. most interestin Q, will be on just how to implement th. revisions. It may be useful to view each rule as a separate object, and ∴ equally subject to revision — rather than ~~to~~ have th. last few rules only, subject to revision via th. erasure symbols. I think that I can work on this prob. w.o. first ~~to~~ going into th. ~~t~~ method of discover of th. "factorization" of th. formulae of .20. Tho, it is conceivable that th. factorizn problem mite cast some lite on th. "revision" problem.

F Ap 27, 62 TMg

Part 3

or: 554.40: Th. Revision prob: ~~AAAA~~ Consider what oc

One has these rules that work for A×B, A/B, an
(A) and a few examples with A+B. Hvr., they
don't work for more complex probs. that have both × and
them.

Now — for me, it would be clear that if a set of unordered rules
got me down to a number (say 1.3053) in working a
prob — and that no was th. wrong answ., then I would
know that one or more of th. rules were wrong, and that addition
of new rules could not alone fix things. i.e. with th. expression
1.3053, we could not do anything further because there was no more "info."
... what I can do, is see how I would solve th. problem — try
to find >1 way — Then decide which of my heurs I should include
in th. machine.

One impt. idea is th. allocation of "responsibility" for error. I think
we can let th. TM realize that th. "rules" are unordered,
and hence, in one sense, "equiv." Also, that most recently
learned rules have hyer aprip of being th. "bad" ones.

Consider a gn. problem: say rules ①, ③ and ⑥ were
used in "solving" it, and th. wrong answ. was obtained .. then
one or more of 1, 3 and 6 is wrong.

Perhaps TM could ask for more problems — even befor he
has found acceptable soln. to all old ones. — So
he can try difrnt. sets of rules on them, and hence
allocate responsibility better.

.35

So: Th. situation! I have various possl. heurs. "W.O." th. heurs,
Th. method of soln. would be to simply try random seqns. of symbols as trial
solns., and use th. Bernoulli coding or some such simple regularity
finder (like l'ngm xtn probys) for coding. Th. time to find solns would
be maximal. With each possl. set of heurs, th. speed would be ↑,
and it would be possl. to "solve" various probs. in th. tng. seq.
quicker. Also, it would be possl. to use less carefully
constructed tng. seqns.

Ap 27, 62 TMS
Dart

.01: 555.40: Some heurs that I've discussed a bit:
(1) Deciding to make new trials by changing terminal and using Bernoulli pcost random trials.
(2) Using special conventions about indexing A with integers
(3) Use of npm. coding (as ZTB 141)
(4) Use of th. special coding of 462.
(5) Use of th. non-dimens PSG coding of PSG-d 543.01 - 545.40.
(6) Th. error responsibility allocation ideas of 555.01 -.35

So — th. idea is to see how far and how fast with how poor a tng. seq. I can go with all of these or a subset of them.
Then, as part of this, see if I can devise a good notation for th. heurs — or some way of expressing many of them compactly, so that these heurs have a reasonable pcost, and so that new trials for heurs will be suggested

Another impt. approach is via ordered rules. These probly have easier heurs — since one can discover th. rules first, and their precidence rules later.

This is beginning to look like around 433.14 - 435.30 and th. folg discns folg. 435/ (up to at least 438.20) in which I tried to unify these 4 approaches, with perhaps some success.

It would seem that with fairly simple heurs, and th. CDS set of rules, one couldn't do much. E.g. one would be restricted to a tng. seq. in which there was little or no "backtracking". Th. "erosure symbol" concept, in itself, wouldn't be good enuf to deal with th. tng seq. in which + and - and ( ) were learned before x and /.

But this, in itself, is of interest — i.e. knowing just how far and fast one can go, using certain limited heurs and with certain tng. seqs.

As one expands th. set of heurs, one can do better. Th. idea, is to devise a formalism for expressing th. heurs, so that one can devise a tng seq. for th. heurs.

I think that I'm finally getting back into th. proper spirit of

≡ Sat Ap 28, 62 TMγ

Dart

.01: 556.40: around 440 etc. E.G. I ~~really~~ really understand what are (454.30≡ and 431.37≡ — also explains G_4)

I think th. main idea was to formalize G_3 as much as so that th. problems of getting better G_3's ~~that the~~ was similar to G_3's problem of getting better M_i's. Then we give TM problems so that th. M_i have to solve problems similar to th. probs. that G_3 has to solve.

It may not be as easy as this (i.e. $TM_1 = TM_2$) — but at any rate, this work of formalizing G_3 (i.e. formalizing th. heurs of getting th. M_i's) will be stage steps in th. right direction.

I can now read all this early (~ 440) stuff and understand it much better. Start around 425 ≡ and read it all over again. 425 ff (th. next few pps) deal with L_3 and G_3 — so I want to review that.

So: Back to Main Review:

.20: 427.10: Seems to me, that if I knew $G(x,y)$, then I could, at low cost, define $F(x)$ to be that value of y for which $G(x,y)$ is max. This is a well-defined function if $G(x,y)$ is known. Also, we don't have to predict what th. next x_i will be.

Th. particular soln. discussed around 426-7 is a particular el-zn. of th. general soln. An impt., rather general Q is — what accuracy can one expect from particular, aprox. el. methods?

A rather non-el. formulation of th. problem: Given several x_i, y_i, G_i triplets. Gn. a new x_j, to find a y_j for it ∋ th. expected value (or some other funct. of th. probty distribn. for G of that x_j, y_j) ~~in optimum~~ of th. assoc. G value, is max.

Th. past F_k's that have been tried can be viewed as part of th. code for th. past $[(x_i, y_i, G_i)]$ set, augmented by a new x_j, y_j, G_j

.35 on 462.33: Note that if we define th. 4 ngms M+, M−, MX, M/ then we can define th. 4 ngms M+A1A2, M−A1A2, etc. — This will ↑ th. aprip of 462.33 tremendously!

Sun Ap 30, 62

Dart

558

TMY

.01: 557.40: I think that I should write th. SOS paper now, and I should write G. Jacobi and tell him that th. abstract [illegible] be revised — that th. paper will be less specific.

What to include in paper:

Th. problem and CDS soln.

Discussn. of speed of soln. (a) if x, \, (,) are taught before + and —. Discuss mildness of ↑ of speed as more rules are lea[rned]

Under condition (a) show how soln. speed increases when th. special grammar of 462 is used

Perhaps discuss p, $\sqrt{p}$, v.s. "flat" rule. Contrast with non-stoch. rule that takes trials in order of proby. This is uniformly better — i.e. fewer trials, and better extrapolation. This mite be an appendix — if th. paper isn't long enuf w.o. it.

As I'm thinking of it now, th. abstract, as submitted, would be literally correct — but it would be a bit of a ~~misinter~~ misdirection, since it would emphasize less of what has been done, and give more conjectural things.