Dart 468 – 483

Attempts to find gramm of PSL's & PSL's gen. sample set of S's.

Reference to "factoring", hill-climbing mutations. If soln. to FSL's is found probably soln. to PSL's can be found, since ZTB 124 shows they have very similar structures.

At 468, a notation for PSG's is devised so that cmii can be used rigourously for PSG gramm discy anal'n.

→ 468.20 PSGd and 543.01-545.40 extend this notation to non-dimensional PSG's.

in MF 13 Drink

000000

Wed 8, 61

TMS

Dart

(468)

.01 .467.40 : We have finally told which α is meant, we can start subscripts that tell which β is meant.

Now note that we can have loops with no trouble at all!

E.g. say $G \equiv M \alpha, \beta, A$

$\alpha \equiv B, G$

Then G 1 1 would mean M B.

G 2 " " A

G 1 2 2 " " M A

G 1 2 1 1 " " M(M B) = i.e. M M B.

This is done in a section of Info and Control paper — write ~ March 62.

This notation tells pretty much how I expected my old 'stock PSG' to work. This notation doesn't yet nearly tell how to put the probys in yet. The old method is, however, one acceptable way.

.19 This notation really seems to suggest a way to do PSGd! - e.g. $a^n b^n$

.20 SN : On How to use this notation for non-dim. PSG's.

Note: This part (.20 - .34) is probly done more generally in PSGd 543.01 - 545.40

Say $G \equiv A, B ; C, D$ then G 1 2 means B

i.e. 1 means the first pair, 2 means the 2nd component of the first pair.

To write grammar rules using such ntpsts:

$G \equiv A, B ; C, D$ $\mathfrak{B} \equiv E, F ; G, H$

$S \rightarrow G 1 1 \mathfrak{B} 1 1 G 1 2 \mathfrak{B} 1 2$ means:

This means the left member of G : i.e. A or C ; this means the left member — i.e. B or D, resp. Here this means the first use of G in this role.

The 1/1 rule

$S \rightarrow G 1 1 G 2 2 G 1 2 G 2 1$

in red and the arrows on the tie lines tell which is which. The tie lines are indicated

These 1's tell where the tie lines go, as do these 2's

This 1, 2 pair tell which ntpsts components of G to pair as do this (1, 2) pair

.34

.35 To "do" (i.e. devr. the gramm. of) $a^n b^n$:

Say we are given the corpus $S \equiv ab , aabb , aaabbb ;$

We can define $\alpha \equiv ab$, then $\beta \equiv \alpha b$; $\gamma \equiv a\alpha$

468.01

W. Oct 4, 61 TMY

Dart

.01: 468. 40: $\delta \equiv a\gamma$, $\epsilon \equiv \beta b$

At this point, since $\beta \equiv \alpha b$ and $\epsilon \equiv \beta b$ and $\alpha \equiv a, \alpha, \beta$; with

we can factor, obtaining $\alpha \equiv a\alpha, a$

So — we start out with the corpus ab, aabb, aaabbb.

Next we write gramm: $\alpha \equiv ab$, $\beta \equiv \alpha b$, $\epsilon \equiv \beta b$

$\gamma \equiv a\alpha$, $\delta \equiv a\gamma$.

Next we factor to get ≡ the gramm.

$A \equiv a, \alpha, \beta$ or $C \equiv a, \gamma, \delta$

$A \equiv ab, a$ $D \equiv aC$

$S \to aA$, or $S \to Db, ab$.

~~Actually, we should be able to factor to get~~

No!! — This is in the wrong direction. It would get us ab^n or a^nb.

The gramm we want is $S \to Ab$

or $S \to a\beta$ $A \to aS, a$

$\beta \to Sb, b$.

.24

.25 Well; say we factor $S \equiv ab, aabb, aaabbb$ into

$S \equiv a\beta b, ab$; $\beta \equiv ab, aabb$ This is a good factorizn.

because we can write $\beta = S$. — So $S \equiv aSb, ab$.

The corpus itself can then be written, $\underset{ab}{2}$; $\underset{aabb}{1,2}$; $\underset{aaabbb}{1,1,2}$;

I think the point may be this: In factoring, I had previously found this to be of little value, since it usually would post much if at all. However, now, when one factors, one looks at the ~~multinomial factor~~, each upmost factor, and one sees if it overlaps much with any other known upmosts. [At each point in the work, have for each upmost, a list of its known members]. If there is much overlap, then it will pay to use the other (already defined)

470.01

W Oct 4 61 TMJ

Dart

.01 469.40: ngmst symbol, and perhaps add on a few symbols. Note overlap can be partial — neither ngmst of interest need

e.g., consider the example of 469.25 : also, say we have identity element I, → IA ≡ AI ≡ I for all A.

Then start with S = ab, aabb, aaabbb.

factor

S = a (I, ab, aabb) b

Now: compare S = ab, aabb, aaabbb and (I, ab, aabb)

There is overlap only on ab and aabb — so perhaps with this sample size (only 3 examples) we couldn't do this factorization and ↑ pcost, but doing the factorization, we get

S = a (I, ab, aabb) b = a (I, S) b .

or S = a (I, S) b

or S = ab, aSb .

since S has an ngms that doesn't have (ie. aaabbb) specifying ngms within S will cost a bit more than specifying ngms in , if was gn. a special deln. Also since has 1 ngm (I) that S doesn't have, we have to write S = a (I, S) b which also costs extra

Let's try it with only 1 sided factors, like 468.35 – 469.24.

Start: S = ab, aabb, aaabbb

S = (a, aab, aaabb) b ← I'm not sure that this factorization would

.25 take place — even if there were more examples of S gn.

— any way — then S = (a (I, ab, aabb)) b with the immediate

S = (a (I, S)) b = a (I, S) b = ab, a$b

would certainly be made (if the compsz. were large enuf) since as before (I, ab, aabb) and S have much overlap.

look into this — see 471.07

At the present time, there is some confusion in my mind about ≡ 2 apparently difnt approaches to derbg a gn set of S's:

(1) write out the pure grammar, then write the corpus as a sequence of choice integers that tell which choice to make at each branch pt.

(2) start out with <S = the corpus> (e.g. like S = ab, aabb, aaabbb) as an initial grammar and corpus deen. Then modify this grammar in ways so as to ↑ total pcost.

.01: 470.40: Well, actually, there is no difference betw. th. 2. Th... is a special case of th. first. E.g. along with th. gra[mm]

S = ab, aabb, aaabbb, one must also write, 1, 2, 3, since these nos. tell which choices were made to obtain th. corpus. — So we always have both th. "pure gram" and th. set of integers giving th. choices.

.07: 470.25: Re: th. poss. ↑ of cost assoc. with this unilateral factoring — also, th. Q of whether we would want to use th. notation

s = (a, aab, aaabb)b

.12 rather than s = αb

α = a, aab, aaabb. } Th. Q involves th. relative cost of a pair of parenths v.s. th. cost of introducing α and then defining α. Note that we have th. constraint that α must be defined after it is introduced, befor th. gramm terminates. This can, hvr., be automatic, since if α is th. first ugm. defn. introduced, then its "defn." must follow that of S. There are actually few constraints in this > gramm.

E.g. say we want to write th. gramm:

$S \rightarrow \alpha\beta, a, b$

$\alpha \rightarrow \gamma S, \beta\alpha, b$

$\beta \rightarrow a, b, \beta\beta\alpha$

$\gamma \rightarrow Sa$

We would write it: $\alpha\beta, a, b\phi\ \gamma S, \beta\alpha, b\phi\ a, b, \beta\beta\alpha\phi\ Sa\phi$

we don't need th. end of grammar stop, φ, since at this point, if there is an "end of defn", we cannot define any new ugmsts, since if we do, they will not have been used in th. rest of th. grammar, and such definitions could never be legally referred to in any conceivable coding.

Nevertheless, it is still possl. to write gramms within these formalism, that can derive nothing a/o have redundant parts: e.g.

$S \rightarrow \alpha\beta$

$\alpha \rightarrow Sa, \beta\alpha$

$\beta \rightarrow a, b$

It is clear that some of th. ugms on th. rt. must be purely "basic alphabet substrings". Hvr, this is not suff't, as in this example, which produces no finite strings. — tho it does perhaps produce some infinite ones.

W. Oct 4, 61 TMY

Dart

.01: 471.40! Getting back to the factoring Q of 471.07:

.02 the pcost of $ab, aabb, aaabbb\phi$ 1, 2, 3
$\frac{1}{3} \cdot \frac{1}{4} \cdot \frac{1}{5}$ ← pcosts

.03 v.s. that of (see 471.12) $\alpha b\phi$ a, aab, aaabb ϕ 1, 2, 3
$\frac{1}{3} \cdot \frac{1}{4} \cdot \frac{1}{5}$ ← pcosts

Since the numerical parts on the rt. have the same pcosts, we need to compare the rest. Same no. of symbols in each, tho .02 would have more, if the samp. were larger, so let's consider a sample of 4:

$a^n b^n$ $(n = 1/4)$.

so

ab, aabb, aaabbb, aaaabbbbϕ 23 symbols

v.s.

$\alpha b\phi$ a, aab, aaabb, aaaabbbϕ 22 symbols.

basic alphabet: α a b s, ϕ — so pcost of α is $\frac{1}{4}$, since ; and ϕ are not legal there. (the α is in red because as soon as it is [illegible] used in the corpus, the "basic alphabet becomes $\beta \alpha$ a b s, ϕ).

This looks like a lot of trouble if one wants to get the exact pcosts.

Here, say we have a corpus $a^i b^i$ $(i = 1/N)$

Then, consider large N's. The marginal pcost of a b will be $\approx \frac{1}{2}$ for both codes. Here the non-factored code has N− more b's in it. The factored code has the additional symbols α and ϕ.

I think it very likely that for large n, the factored form will have higher pcost.

Tho it is by no means sure that we can do $a^i b^i$ with unilateral (or even bilateral) factoring, we will look at the (≡ mirror) palindrome loop next (e.g. aa, baab, babbabbab, etc).

In general, unilateral factoring should do no good, since if there are about n = no. of s's that end in a and b, then we would have to factor in to 2 sets, (i.e. ones ending in a, and those ending in b) — and choice betw. the 2 sets would use as much pcost as we'd gained.

Bilateral factoring will help, and should be able to work the prob. if the samp. is fairly large and the choice betw. a and b is fairly random — i.e. assuming a simple stoch PSL generating the sample.

473.01

Dart →

.01 : 472.40 : I hate to have to depend on bilateral fa[illegible]
actually, I don't think it need involve inordinate [illegible]

It seems clear that th. ~~prob~~ "mirror lang" couldn't concei[illegible]
be solved by unilateral factoring, because viewed from one [illegible]
there are no regularities!

Granted **bilateral factoring**, it would seem that both mirror langs a[nd]
$a^n b^n$ are "under control." Hvr. I'm not so sure of th. details of mirror lang.
and, **there** are some branch trails on $a^n b^n$ that may cause trouble —
e.g. one could factor S into ab, $aa(bb, abbb)$
particularly if ab were missing. Hvr. bilateral factoring
would tend to have a higher Δp cost and ∴ make this less likely.

Another possy is to define $\alpha \equiv aa$, or perhaps $\alpha \equiv aaa$

Also, consider $\alpha \equiv ab$; $\beta = a\alpha b$; $\gamma \equiv a\beta b$, $S \equiv \alpha, \beta, \gamma$.
Here, hvr, th. set α, β, γ can be easily factored into
$\approx a(\alpha, \beta, \gamma)b$.

If these 2 PSL's can be solved in th. above ways,
Then one can also solve th. $\beta\Sigma$ of these 2 langs, and
th. a concat. of them:

E.g. $\beta\Sigma$: $S = \alpha, \beta$

$\alpha = ab, a\alpha b$

$\beta = a\beta a, b\beta b, aa, bb$.

.30 Gram. for concat. : $S = \alpha\beta$

$\alpha = ab, a\alpha b$

$\beta = a\beta a, b\beta b, aa, bb$.

$\beta\Sigma$ is ~~easy to~~ easy to solve — just solve th. α and β
subsets of S separately.

Concat. may be diflt: **One** way to do it: say S_1 is th.
set of s's in th. mirror lang., and ~~[illegible]~~ A is a fixed string.
Then $S_1 A$ is solvable by **bilateral** factorization, th. factors

W · Oct 4, 61 TMS

Dirt

.01: 473.40! being a( )aA and b( )bA. If Pk large enuf (!) A can be any s from $a^n b^n$, and we could solve for Pk. mirror part! Mirr. Pk. samp. would have to be quite large for than a large enuf sample of S, A to occure for any A at

.05 Another possy: Pk. mirror lang. may be solvable from Pk. inside — i.e. define $\alpha \equiv aa$, $\beta \equiv bb$, $\gamma \equiv b\alpha b$, $\delta \equiv a\alpha b$, etc. We mite be able to do this with a non-dim. gramm: eg.

$\alpha = a, a; b, b$ } α is a 2tpst.

then $\beta = \alpha 11 \alpha 12$ (1st component of α; 2nd comp. of α; first instance of α in this rule)

so $\beta = aa; bb$.

I think that working out from Pk. inside is better, since if is always possl. (if ever!).

In mirr. lang, aa is of notiably "unusual freq" only if sample is large (assuming a simple stoch creative PSG for Pk. creation of Pk. samp.) If Pk. mean s length is N, then aa occures $\approx (1 + \frac{2}{N})$ times as frequ as $f_a \cdot f_b = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$. This $(1 + \frac{2}{N})$ factor is true for either aa, bb or both, and is not subject to much statistical fluct. — So its not as bad as it sounded at first.

Anyway, after defining this α, we can define β, then $\gamma = \alpha 11 \beta \alpha 12$, etc. — each of α, β, γ etc. is useful to define and ↑ pcost. Then we just factor out α from $S = \alpha 11 (\beta, \gamma, \Delta) \alpha 12$, β and we get $S = \alpha 11\, S \alpha 12, \beta$.

I think we can do a similar sort of thing with $a^n b^n$ — but a bit simpler.

.35

.26 Actually what we can do (and this is not far from .05 -.35) is, in Pk. mirror lang, define a large set of Pk. central ngrams (since they usually occure many times in a large corpus), then find Pk. gramm of this set of ngrms by any good method — like bilateral factorizn.

475.01

W Oct 4, 61 TM

Dart

.01 : 474.40 : Th. / thing is, hvr, that it can be done with th. value of this
.02 of 473.30 as well as th. pure mirr. lang.

.03 some other langs to try:

③ $a^n S_1 b^n$; ④ $A\ a^n b^n\ A'$ (A' is th. mirror image of A)

Σ A is arby.

So $S_1 \equiv AA'$.

Note th. difference between th. CDS "rules" of 462, and th. WPSG. Th. recent work gives a set of defns, that are the "same" as $\alpha = ab$, then th. corpus is dcrbd. in th. usual way, by listing th. ugms and basic symbols used. Th. formalism used is identical to that used in ZTB 141 - except it is expanded "somewhat".

We start out our corpus with an imaginary string of th. "basic symbols" (Including punctuation). Next, we list th. definitions. This is followed by th. Φ (= grammar stop) symbol, which is followed by th. corpus descripta integers, which use th. defns. Th. defns. use those elaborate sometimes looped "subscript" conventions. I think this notation is well in hand.

Th. CDS "rules" (sets of) were not th. same as th. sets of "defns" in th. WPSG. They were, e.g., unordered. To th. extent that there is a search involved in finding th. rite xfunn, we can assign a cost to each choice of xfunn, and ∴ get a total cost for th. entire set of problems worked. Maximizing this cost might mean reducing th. no. of incorrect trials — which is desirable. Focus

To what extent can th. CDS Gramm. be patterned after the WPSG as a dcrn of a corpus? To what extent can th. CDS rules as an Op. be patterned after th. WPSG as a dcrn of a corpus?

as a dcrn.

It would seem that th. CDS gram as dcrn, could not directly be patterned after th. WPSG dcrn. This is because in CDS, one can't tell how many legal passes whether a gu. defined symbol has, until one knows what follows it in th. corpus. / One sided CDS uniformly (say left side only) can, hvr, be done w.o. any trouble. I'm not certain as to whether a one sided CDS is really "more powerful" than a WPSG — i.e. that it can express some langs. beyond a WPSG's power.

Also, there is th. Q: If one has a CDSG in which th. verdac results are indep. of any ordering of th. rules, then is it poss'l. that this lang must be expressable by a WPSG?

476.01

Th Acts, 68 TMY

Part

.01: 475. 40: This sideroad onto PSG isn't nuts so bad. I want TM to be able to work with ethnic langs, and PSL's an approxn. to them, and have many impt. characteristics of

---

We can use a format for CDS dcrn. that is similar to that of WPSL, but it will apparently not have the same hour value for gram dcrry. I.e. First write the gramm. as subs. rules — like:

1) S → a b M c , c M d ,
2) M → α β
3) b M c → b A B c , b X c
4) α → c d β
⋮ ⋮ ⋮

Then, write 1 or 2 depending on which of the 2 subs's will be used. Say 1 is used, so we have now a b M c

There are now 2 possl. subs: ~~α β or b A B c~~ so we write rule 2 or rule 3. Say it is rule 3. so we write "2" (i.e. 2nd possy.). Next we tell whether it is b A B c or b X c we want: so 1 or 2.

Note that the foregoing notation treats the gram rules as substit. rules, and not as datus, as in WPSG. For this reason, I think the WPSG hill-climbing technq will not work for CDS gram discy — tho since WPSG is a special case of CDS, I can try to see if the WPSG methods suggest CDS discy methods.

A worth while goal would be to either develope the dcrry of non-dim WPSG's a/o CDS. grams. These are both very powerful, and can be used by any level of TM for discovering regularities in a corpus a/o a gramm. Also, I want TM to be able to deal with ethnic langs.

---

1) Get the formal dcrns using non-dim. PSL's more exact. the 1 dim. WPSL notation seems well in hand. 2) Get the list of gram. discy methods for WPSL's more formalized a/o just dcrb these rules in more detail.

477.01

Th. Oct 5, 61 TMY

Dartd

.01 : 476.40 : One of th. rules for PSG disc that I've [illegible] is factoring a set of defined terms. e.g., say

$\alpha \equiv ab$
$\beta \equiv a\alpha b$
$\gamma \equiv a\beta b$

We then factor them into $a(I, \alpha, \beta$

But suppose we had $\alpha = ab$
$\beta = a\alpha b, cd.$
$\gamma = a\beta b$

(1) How would we factor it then? This looks like we factor th. gramm rather than th. corpus!

(2) Another Q is — suppose we have found th. gramm in a way that leads to a low local max — how can we go back? I think I discussed this business of "going back to th. last fork", and th. general backward exploration technique. → 481.05

.15 (3) Is th. non-dim notation easy to solve PSL's "from th. inside out" as in 474.05 - .35 ? — or is th. method of 474.36 - 475.02 adequate?

Mr Coriolis, spoon fed alley cat.

Cat Coriolis, spoon fed alley cat cat! Cat cat cat cat cat cat cat cat cat

A Big possl. trouble in solving a^nb^n! That th. defn. ab may not give as great a pcost as th. defn. aa. This is, hvr., not certain. The ab occurs in fragly, its defn. is relevant everytime a occurs, and perhaps every time b occurs (it should be symm., whether one codes rite to left or v. versa).

This ↑ is of some import., because defining ab, is one way to solve PSG's "from th. inside, out."

One mite tautotrivially make many ngm defns, wich one discards if they do not soon lead to factorizations. Even so, th. sample from a^nb^n could then lead to defining $\beta \equiv bb$, $\alpha \equiv aa$, $\beta\beta = \gamma$, etc., and eventually to th. gramm a^nb^m. For large sample, a^nb^m clearly is of lower pcost than a^nb^n, because a^nb^m derbs each S in th. corpus by 2 params, m and n, while a^nb^n requires only 1 param.

This seems reasonable: e.g. in th. case of a^nb^n, defns ab, aa and bb should all be tried, and one should see where they lead — if anywhere. In this case, ab leads. 478.01

TJ Oct 10, 61 TMY

Dart

.01: 477.40 to $a^n b^n$, with ~~bb~~ aa and bb leading to $a^n b^m$. In these cases, it is not easy to follow these leads very far before the superiority of $a^n b^n$ becomes clear. Since $a^n b^n$ is a 1 param. there is little that can probably be done to ↑ pcost - other than reduce ~~s~~ ~~and~~ n by a certain constant, if the first few members of the $a^n b^n$ set are missing.

.10 As for mir. langs., there are no ~~compact~~ defs. of "compact ugms", that will work, other than aa, baab, etc. — ie. [redacted] as's. These defns will, even in the lang ③ of 475.03 lead to a soln.

A better kind of defn. that will perhaps lead to the soln. of all PSG types (both 1 dim. or non-dim.) is the 2 tp..

— e.g. $\alpha \equiv (a, a)\ \phi$
$\beta \equiv (b, b)$

so, abbbaabbba would be written:

$\alpha_1 \beta_1 \beta_1 \beta_1 \alpha_1 \alpha_2 \beta_2 \beta_2 \beta_2 \alpha_2$.

If we want to solve only 1 dim. WPSL's, we have the constraint that we cannot have langs like $a^n b^m c^n d^m$ i.e. interlocking nests. This translates into a simple constraint for ϕ, which signify ~~the~~ ↑ its pcost. The more exact code is

$\alpha_{11}\ \beta_{11}\ \beta_{21}\ \beta_{31}\ \alpha_{21}\ \alpha_{22}\ \beta_{32}\ \beta_{22}\ \beta_{12}\ \alpha_{12}$
1 2 3 4 5 6 7 8 9 10

sticking to 1 dim WPSL's

— the first 5 symbols are without much constraint. for the 6th symbol, we can start with α or β, say. If α is chosen, 2 must follow, since α_{1i} would make non-1 dimpsl's possl. Then the 1st subscript 2 must follow. Next, (since the $\alpha_{21}\ \alpha_{22}$ pair have been "elimed" (sortof)), β has a pcost of $\frac{3}{3+2}$ (α would have a pcost of $\frac{2}{3+2}$).

479.01

Tu Oct 10, 61 TMG

Dart.

.01: 478.40! There are, in addition (perhaps) to Th. "fac~~t~~ discussed a while ago, 2 other PSG discy me~~t~~ that I am ~~now~~ much better able to formalize for "discy W.O. instrator". These are (1) ZTB 125 - Th. "factor I'm not sure that I ever made it very plausible that it would even with an instructor. (2) ZTB 124 (Paris) The "basic loop" method. Since this makes a PSL correspond to a FSL, I should try to "do" FSL's W.O. instructor, ~~to~~ then gen'lz. Th. method to PSL's.

A way to "do" FSL's (and perhaps PSL's): Make various sets of npms ~~in any subset of this npmst~~ that are meant to be Th. same pos. If any subset of this set has sig. difnt ("connection" "statistics") than it will be definable as a new npmst, and it will be a difnt pos.

For FSL's an equiv. set of substrings is one that ~~determines from Th.~~ has Th. same state xfun matrix, e.g. in a gn. set, they will all be able to transfer from state 1 to 3 or 8; from 2 to 2, 8. or 9; etc.

.30 A trouble here is that for stoch. FSL's, we have difnt probys for transfering, and so we get probly an ∞ of "equiv. classes". .33 ⟹ (see 483.01 for a reasonable idea)

A rigorous treatment of stoch PSL's: Each string has a matrix giving its state transfer probys. It is an N × N matrix if its an N state lang. Each symbol has such a matrix, and Th. matrix of any string is Th. ordered product of Th. matrices of Th. elements of that string.

For a non-stoch FSL there are only 0's and 1's in the corresp. matrices, and there are only a finite no. of such matrices possible.

A formal soln. of Th. stoch. FSG problem: Assume that Th. no. of states is N. If there are m basic symbols, then we have mN^2 unknowns in Th. m matrices. We take Th. set of s's in Th. corpus, and assume that each one, transfers from S_0 (initial) to S_∞ (terminal). [S_i = state]. Then Th. probty of each gn. s.,

480.01

Tu Oct 10, 61 TMS

Dart

.01:479.40 will be a Th. 0,∞ element of Th. matrix that is Th. "element" of Th. / matrices of that s. We want the product of elements for all of Th. s's in Th. corpus to be max. — I.e. so Th. corpus has a max. probty of having been created.

To solve this — we use Lagrange's multipliers to keep Th. rows of Th. matrices Σ to 1, and take partial dirives. Th. eqns. themselves are very complex, but it is conceivable that these partial dirives wouldn't be.

Th. notation for "Basic strings and loops" of 2TB 124 could (probly) without much difty, be rigorized to a quantitative derv. method — similar to Th. ideas of 478.10–.40. Th. idea in 2TB 124, was that Th. insertion of certain "parts of speech" into other 2tps in certain ways/ would leave Th. "pos" of these 2tps invariant. Again, as with FSL's there is Th. same difty that one gets in going from determ. to stoch. langs. i.e. 479.30–.30.

— It would seem that if one could solve stoch FSL's using Th. "basic loop" method, that one could also solve PSL's (even non-dim ones) using trivial modification → 483.01 of these methods.

.25

On "factoring" (monomial factoring): say we have Th. 3 xfms $\alpha \to ab, ac, ad$. We factor them into $\alpha \to a(b,c,d)$.

Also, say we have $\beta \to e(b,c,d)$. Now — it will not be worth while to give these 2 b,c,d sets a "common name", unless they have "about" Th. same freq. distribs in Th. 2 cases — otherwise we will be destroying info and not ↑ pcost.

In general, it does not pay (in terms of ↑ pcost) to merge 2 sets unless their statistics are close enuf.

Note that one can occasional merge parts of 2 sets with profit. e.g. say $\alpha \equiv a, b, c, d$ and $\beta \equiv a\, b\, c\, d$ (rel. freqs: .1 .2 .2 .5 ← → .5 .2 .2 .1) then we can partially merge these by $\gamma \equiv bc$

$\alpha \equiv a, \gamma, d$; $\beta \equiv a \gamma d$.

481.01

W · Oct 11, 61

TM 8

Part 2

.01: 480.40 : Another trick would be: let $\gamma \equiv a^{-1}$

Then $\alpha = \gamma^{.5}, d^{.5}$; $\beta = a^{.4}, \gamma^{.5}, d^{.1}$

Th. conditions under which these kinds of defs would be useful must be worked out in some detail.

.05 : 477.15 I would like some sort of "self-correction" feature of th. gramm discy method — s. that if I get into a local max, then, with a larger sample, I would automatically get out of it. Tho this would be nice, and is perhaps possl. for certain types of local maxima — it seems not to be possl. for th. folp., if I use anything like th. kinds of "steps" on th. hill, that I've outlined up to now; e.g. say one is gu. a samp. of $a^n b^n$, and one has obtained th. gramm of th. local max — $a^n b^m$. It would seem that there is now no way to realize that th. langs a^n and b^m are related in a simple way. A hyer order TM could view these a^n and b^m — type derns. and soon discover that $m = n$, but a rather complex hyer order TM would be needed unless m and n were written in a special notation — i.e. n represented by $1^{(n)}1$. But even so, I'm not so sure it would be easy to get th. desired affect.

My present conception of my previous work on PSG doc:
That I had it in th. form of a h.c. prob., and I had 4 types of steps: (1) ngms (2) monomial factoring (3) Multinomial factoring (4) perhaps "branching" (whatever that was) (5) loop closing. (one of these step types was included in another)

My method was to go as hy as possl. with steps of type (1) — then take a few step (2) types, then go back to (1), then back and forth betw 1 and 2 till a local max is reached — then go to (3), etc. Perhaps th. folp. idea!

482.01

w. oct 11, 61 TMY

Dart

.01'.488.40 ! Take steps of type 1 untill local max. Then step of type 2. etc.

Or. more compactly: ① take best step of type i. If possl. (and i ≠ 1 and i ≠ 4) set i → whichever is greater of i-1 or i+1 and go to ①

If such is not possl. set i → i+1 or i → 4 (and i=1) and go to ①

Or: if you can: make a good step of agn. type / make th. best one possl. of agn. type and try at a i lower type.

If you could make a good step of agn. type try at th. next hyer level.

Th. limits are th. levels i=1 and i=4; i=0 means "i=1", and i=5 means "stop"

For a while, it was not that making up the optimum "non-branched" PSG. ("Non-branched meant" only defs like $\alpha \equiv abc$. not $\alpha \equiv abc, cbb, rar$. → was an adequate step to be followed by "branching" and "looping" to make a complete PSG. Unfortly, a difty was experienced in devising th. nbPSG. i.e. th. fallacious $p_{ab} > p_a p_b e$ subs. criterion, finally this was solved, and th. 7090 pgm and ZTB (4) were writ.

I think that th. nbPSG was not hvr., adequate in th. sense desired

Now, with th. present notation, defining agns, branching looping and factoring, are more like a single step type than befor. Also, I suspect that I have a much better notation than befor, that gives a better idea as to when and why a gn. step type really "does any good."

483.01

F oct 20, 61 TMJ

Dart → 479.33, 2650
480.25 spac

(483)

.01: 482.40

"Factoring" in stoch FSL's:

Say A, B and C are sub-strings that occasionally have bag...

Then, if A, B and C terminate in R. state distribution, ...

(NO – this wouldn't work).

A possl. idea is to make ngrams of substrings that have appear... have v statistics — e.g. their freqs are close; they are followed &/o preceded by v ngrams, etc. As more data comes in, we can afford to be define new ngrams and thereby make finer distinctions.

A sort of way of looking at this: say an ngm has a d x d matrix. — Then with a gv. samp. size, we can only distinguish k levels of each of R. d^2 matrix elements. As samp. sz ↑, k ↑.

Another trick: try to compute R. relationship betw. symbol freq. a/o 2gm freq. &/o ngm freq., and R. matrix elements.

spac. pged 543.01 ← this may be unbrant

517.01 perhaps