

Sequentiele interpretatie van ponsband.

Dit rapport handelt over de structuur en onderlinge samenhang van vier procedures (of proceduuretypen):

integer procedure HEP	voor het aftasten van ponsband
integerproceduuretype CHAR	voor het aftasten van een paginabeeld
integer procedure SYM	voor het aftasten van een rij basic symbols
real procedure NUM	voor het aftasten van een rij numerieke waarden.

Van het type CHAR beschouwen we er twee, zeg CHARF en CHART voor het aftasten van een paginabeeld van flexowriter resp. teleprinter; de uitbreiding naar meer dan twee verschillende CHAR's ligt voor de hand. We gaan voorbij aan het feit, dat het paginabeeldend vermogen van de teleprinter toevallig bijna geheel vervat is in dat van de flexowriter en beschouwen dus hun respectievelijke paginabeeldende vermogens als volslagen verschillend.

Als een routine van het type CHAR een band van het betrokken apparaat te verwerken krijgt, zal deze bij herhaald aanroepen een waardenreeks afleveren, die uniek verslag aflegt over het bij deze band behorende paginabeeld en wel regel voor regel, elke regel van links naar rechts. Het zijn de routines van het type CHAR, waarin de eigenschappen van de diverse hardware ponsers/bladschrijvers hun neerslag vinden. Over deze eigenschappen worden twee veronderstellingen gemaakt

a) We veronderstellen dat het hardware apparaat beschikt over enige geheugenelementen, waarvan de beginwaarde (het begin van) het paginabeeld mede kunnen bepalen, dat aan een band ontleend wordt. Wij noemen heersende case, wagenpositie etc. Hieraan beantwoordt, dat elke routine van het type CHAR, voordat hij over een paginabeeld een verslag kan gaan afleggen geactualiseerd (geinitialiseerd) moet worden, waarbij deze mechanische geheugenelementen een bij de actualisering opgegeven beginwaarde toegedacht krijgen. Na actualisering verslaat CHAR het paginabeeld, dat ontstaat wanneer

- 1) het apparaat in de toestand gebracht zou worden zoals bij de actualisering van CHAR opgegeven
- 2) de band ingelegd wordt met de eerste heptade die door CHAR via HEP gelezen gaat worden in het leesstation van het apparaat in kwestie.

b) We veronderstellen de door de hardware van de ponsers/bladschrijver gegeven code zo ingewikkeld, dat in het algemeen geen natuurlijke synchronisatie tussen paginabeeld en bandbeeld bestaat, maw. dat als het verslag van CHAR tot een bepaald punt op de pagina gevorderd is, het plotseling doorlezen van de band via HEP op een in het algemeen ongedefinieerd moment inhaakt.

Is het gebruik van CHARF en CHART -door de complexiteit van de apparatuur en door niets anders!- aan enige beperkende regels onderworpen, verder beogen we de grootst mogelijke algemeenheid. (Als een implementator voor sommige hieronder te noemen mogelijkheden geen amplooi ziet en ze weglaat, dan is dat zijn zaak; ik betrek ze in het beeld, om dan de weg tot latere toevoeging open te houden.)

De routine SYM kan op drie manieren aan een band een rij basic symbols ontleen

- 1) een rij basic symbols gedefinieerd op een flexowriterpaginabeeld; SYM zal dan via CHARF de band aftasten, dit interpretatieproces noemen we "symcharf".
- 2) een rij basic symbols gedefinieerd op een teleprinterpaginabeeld; SYM zal dan via CHART de band aftasten, dit interpretatieproces noemen we "symchart".
- 3) een rij basic symbols, zonder tussenkomst van een paginabeeld direct op het bandbeeld gedefinieerd; SYM zal dan via HEP de band aftasten, dit interpretatieproces noemen we "symhep".

In het bijzonder zullen we eisen, dat indien via SYM een reeks basic symbols van een band gelezen wordt, tussentijdse wisselingen tussen de drie interpretatieprocessen zijn toegestaan, zonder dat het SYM aanroepende programma hier enige zorg mee heeft. Dit impliceert

- 1) dat in een ALGOL tussen te voegen teksten van standaardprocedures een soort binaire banden zouden kunnen zijn, die SYM via symhiep zou kunnen lezen
- 2) dat een programma, dat in teleprintercode wordt aangeboden, van een omvattend blok kan worden voorzien door er een stukje flexowritertekst voor- en achteraan te plakken.

Evenzo laten wij voor NUM vier verschillende manieren open, waarin hij aan een band numerieke waarden ontleent

- 1) numerieke waarden, gedefinieerd op een rij basic symbols. NUM zal dan via SYM de band aftasten, wij noemen dit interpretatieproces "numsym". (Bij activiteit van numsym doet het niet ter zake of SYM via symcharf, symchart of symhiep werkt.)
- 2) numerieke waarden gedefinieerd op een flexowriterpaginabeeld; NUM zal dan via CHARF de band aftasten, wij noemen dit interpretatieproces "numcharf". (NB. Hierbij kunnen paginabeelden verwerkt worden, die niet als representatie van basic symbols beschouwd kunnen worden, bv. octale getallen met onderstreepte cijfers of zo.)
- 3) numerieke waarden gedefinieerd op een teleprinterpagina; NUM zal dan via CHART de band aftasten, wij noemen dit interpretatieproces "numchart".
- 4) numerieke waarden, zonder tussenkomst van basic symbols of paginabeeld, direct gedefinieerd op een bandbeeld; NUM zal dan via HEP de band aftasten, dit interpretatieproces noemen we "numhiep". (Op deze wijze zouden binaire voorstellingen direct van een band gehaald kunnen worden.)

Wederom moet een programma vele malen NUM kunnen aanroepen, ongeacht de wisselingen tussen deze vier wijzen van getalopbouw.

Behalve dat gemengde toegankelijkheid van een van de paginabeelden of het bandbeeld elkaar uitsluiten, proberen we zo liberaal mogelijk te zijn. In het bijzonder:

- a) Alternering tussen SYM en CHARF (CHART resp. HEP) is definieerbaar mits SYM werkt volgens symcharf (symchart resp. symhiep).
- b) Alternering tussen NUM en CHARF (CHART resp. HEP) is definieerbaar
 - 1) als NUM werkt volgens numcharf (numchart resp. numhiep)
 - 2) als NUM werkt volgens numsym en ~~SYM~~ SYM voldoet aan voorwaarde a).
- c) Alternering tussen NUM en SYM is definieerbaar
 - 1) als NUM werkt volgens numsym (aan de werkwijze van SYM is dan geen beperking opgelegd)
 - 2) als NUM werkt volgens numcharf (numchart resp. numhiep) en SYM werkt volgens symcharf (symchart resp. symhiep).

We observeren, dat er drie elkaar uitsluitende basisinterpretaties aan de gang kunnen zijn; welke van de drie vigeert leggen we vast in de waarde van een aan de informatiestroom toegevoegde integer statusvariabele "basint".

- basint = 1 de informatie is gedefinieerd op een bandbeeld, dat niet uitgelegd wordt als carrier van een paginabeeld (symhiep, numhiep en algemeen HEP toegestaan, CHARF en CHART niet van toepassing)
- basint = 2 CHARF actueel, dwz. het bandbeeld fungeert alleen als carrier van een flexowriterpagina, alleen CHARF van toepassing (en HEP alleen vanuit CHARF).
- basint = 3 CHART actueel. dwz. het bandbeeld fungeert alleen als carrier van een teleprinterpagina, alleen CHART van toepassing (en HEP alleen vanuit CHART).

De statusinteger basint kent nog een vierde waarde
basint = 0 geen van de drie genoemde interpretaties vigeert; in het bijzonder,
als in deze toestand informatie van de band moet komen, die alternatief
op teleprinterpagina, flexowriterpagina, dan wel direct op bandbeeld
gedefinieerd kan zijn, dan zal de aanhef van het dan te lezen bandstuk duidelijk
moeten maken, of basint de waarde 1, 2 of 3 moet krijgen.

De routine HEP.

Nadat deze geïnitieerd is -dwz. de band is ingelegd- geeft de integer
procedure HEP per aanroep de volgende n-ade van de band. Op "EINDE BAND" reageert hij
door totaal de volgende initialisering de waarde "-1" af te leveren. De routine HEP
berekent bij herhaald aanroepen een functie gedefinieerd op het mathematisch model,
dat ik voor een eindige ponsband heb gekozen; een (in de toekomst) oneindige
halfrij "heptaden", die voor een zeker moment niet en na dat moment louter de
karakteristieke waarde "-1" bevat en aflevert. Voor dit model pleite

- 1) dat de oneindigheid de mathematische hanteerbaarheid van HEP alleen maar
ten goede kan komen
- 2) dat een mogelijk relevant aspect van de werkelijkheid (nl. einde band)
hierin zijn representatie vindt.

(Ik ben me bewust, dat andere mogelijk relevante aspecten van de werkelijkheid
onder tafel zijn geraakt. De breedte van het papier zou men kunnen willen gebruiken
ter bepaling van de keuze tussen alternatieve codes; de kleur van het papier of
het fabrieksnummer van de flexowriter, waarop een band geponst is, zou men bij het
optreden van lees- of ponsfouten voor diagnostische doeleinden misschien willen
weten!)

Vooralsnog zie ik geen noodzaak om HEP "een voorgift" te kunnen geven. Mocht
deze noodzaak optreden, dan moet een verstandige keuze gemaakt worden tussen de
bijna equivalente mogelijkheden, die naar aanleiding van CHAR nog ter sprake zullen
komen.

De routines van het type CHAR.

Deze sectie behandelt alleen CHARF, omdat ik met de teleprinter onvoldoende
op de hoogte ben en ik mij, vanwege onbekendheid met standaardpraktijken in het
telexverkeer, nauwelijks bevoegd acht om hier aanbevelingen te doen.

De routine CHARF die ik op het oog heb heeft betrekking op een MC-flexowriter
zonder variabele spatiering, zonder backspace, met vaste instelling van de regel-
opvoer en vaste spatiering van de tabulatorstoppen.

Initialisering van CHARF geschiedt onder opgave van twee parameters, de boolean
"lower case" en de integer "carriage position", die bij het begin van de regel op
0 gesteld wordt. Daarna levert CHARF per aanroep het verslag van het paginabeeld en
wel per positie op de pagina. (Elke "escaping" aanslag wordt verslagen in viervoud
als hij caseonafhankelijk is (dwz. de spatie) en in achttvoud als hij caseafhankelijk
is, nl. al of niet onderstreept en al of niet doorgestreept). Bij het waardebereik
van CHARF is tevens NLCR -dwz. overgang op een nieuwe regel- begrepen, TAB's
worden vertaald in een equivalent aantal spaties, spaties en TAB's aan het einde
van de regel worden vanwege hun onzichtbaarheid niet doorgegeven.

We kunnen (zie boven) bv. basic symbols hetzij op een flexowriterpagina,
hetzij op een teleprinterpagina definiëren en we hebben geeist van bv. een routine
als SYM, dat hij achter de schermen van de ene code op de andere kan overgaan.

Deze overgangen moeten door de band gestuurd worden. Van elke routine van het type CHARF stel ik me voor, dat er een karakteristieke (zelfafsluitende) ponsing (of ponscombinatie) is, die behelst, dat hiermee de heersende interpretatie tot nader order is afgelopen. Voor de procedure CHARF stelde ik me voor hiervoor de stopcode in de betekenis "EINDE CODE" te gebruiken.

Omdat ik voor routines, die in termen van CHARF werken zeer zeker wil toestaan, dat ze niet zelfafsluitende eenheden van de pagina lezen, zal CHARF wel een voorgift van een enkele waarde kunnen hebben.

Voor de routine CHARF doemt de volgende rol op.

De toestand "CHARF actueel" -dwz. basint = 2- kan slechts bewerkstelligd worden door de aanroep van een initialiseringsprocedure, waarbij de statusvariabelen van CHARF (te weten "heersende case" en "Wagenpositie") worden opgegeven. Door de initialiseringsprocedure wordt de voorgift ingesteld op "leeg".

Het geven van een voorgift zou op twee manieren kunnen

- 1) door het parameterloze commando "herhaal bij eerstvolgende aanroep de laatst afgeleverde waarde eerst nog een keer"
- 2) door het parameterhebbende commando "geef bij eerstvolgende aanroep eerst de hierbij opgegeven waarde".

Omdat de tweede methode wat flexibeler is -en de eerste bovendien slecht gedefinieerd is onmiddellijk na initialisatie- prefereer ik de tweede versie.

Indien aan CHARF een voorgift gegeven wordt op een moment, dat CHARF niet actueel is, heeft deze handeling geen effect.

Indien aan CHARF een voorgift gegeven wordt, terwijl een nog niet teruggegeven voorgift hing, dan overschrijft de laatste voorgift de eerste.

Een neveneffect van CHARF zal moeten zijn, om naar aanleiding van het ontmoeten van de stopcode de toestand "basint = 2" (dwz. CHARF's actualiteit) te beëindigen. Precieser: evenzo als CHARF onzichtbare spaties aan het einde van de regel niet doorgeeft, zal CHARF ook onzichtbare NLOR's niet doormelden, wanneer uiteindelijk een stopcode op de band gevonden wordt. Het optreden van de stopcode wordt aan het aanroepend programma doorgegeven, doordat CHARF met de ~~XXXXXXXX~~ waarde "-1" terugkeert, nu in de betekenis "EINDE CODE". Indien CHARF wordt aangeroepen tijdens "basint = 2" en hij keert terug met de waarde "-1", dan heeft dit "basint:= 0" als neveneffect. (Dit geldt ook, als deze -1 uit de voorgift komt; hiermee is dus een manier gegeven om ~~XXX~~ CHARF, als hij actueel mocht zijn, zijn, zijn actualiteit te laten beëindigen.) Als CHARF wordt aangeroepen tijdens "basint ≠ 2", dan keert de besturing eveneens terug met CHARF = -1, doch ditmaal zonder neveneffect.

Bij non-actualiteit van CHARF heeft het geven van een voorgift geen (neven)-effect, het aanroepen van CHARF evenmin.

Met deze definitie van het afvallen van de actualiteit heb ik het volgende bereikt: als SYM van de flexowriterpagina een niet afsluitend symbool leest, gevolgd door een basic symbol, dat 1 positie beslaat, gevolgd door een stopcode, dan heeft SYM bij het afleveren van het eerste symbool de volgende positie al gezien. CHARF heeft misschien de stopcode al gezien, maar nog niet doorgegeven, zodat hij zijn actualiteit nog niet heeft beëindigd. Dus kan de laatste positiebeschrijving nog in CHARF worden teruggeduwd.

Als CHARF wegens het doorgeven van de stopcode voor basint de overgang van 2 naar 0 heeft bewerkstelligd, dan is (als de stopcode via de band is gekomen) de synchronisatie met HEP gedefinieerd: hierna levert een aanroep van HEP de eerste

heptade, die volgt op de stopcode. (En dit is eigenlijk het enige moment, waarop ik synchronisatie tussen CHARF en daarna HEP expliciet zou willen definiëren. Uitgebreidere synchronisatieregels krijgen gauw iets gekunstelds en bovendien: de hemel weet, wat voor dol apparaat we morgen in het systeem moeten opnemen!)

Beïnvloeding van basint.

Rechtstreeks HEPpen is niet verenigbaar met de actualiteit van CHARF ~~XXXXXXXX~~ of CHART. De vraag is, hoe het systeem reageert wanneer men deze (en dergelijke) regels probeert te schenden. In het bijzonder:

- 1) actualisering van CHARF terwijl basint $\neq 0$ is
- 2) actualisering van CHART terwijl basint $\neq 0$ is
- 3) rechtstreeks HEPpen terwijl basint = 2 of = 3 is.

Als basint $\neq 0$ is, betekent dit heel expliciet, dat we ons tot nader order tot een bepaalde bandinterpretatie verplicht hebben. Je kunt zeggen: dan nog een keertje CHARF of CHART initialiseren is kennelijk misplaatst (en daar de consequentie van een letale programmafout aan verbinden); je kunt ook zeggen: het herdefiniëren van de bandinterpretatie is dan kennelijk die "nader order" en het zal wel goed zijn. Uit opportunistische overwegingen neig ik tot het laatste; voor argumenten, dat het de moeite waard is om dit als letale programmafout te behandelen, sta ik open.

Rechtstreeks HEPpen terwijl basint = 2 of = 3 is, is vreemder, omdat we immers ons op het standpunt hebben gesteld, dat bij actualiteit van een CHARF de synchronisatie met HEP in het algemeen ongedefinieerd is. Om dit als letale programmafout te brandmerken, is me echter te gortig. Daarvoor is het primitivum HEP me te fundamenteel. Wel betekent rechtstreeks HEPpen, dat een eventuele CHARF-interpretatie niet meer valide is. Ik stel daarom voor, om HEP als neveneffect te laten hebben

~~XXXXXXXXXXXX~~ "if basint $\neq 1$ then basint := 0" .

(Wordt HEP met basint = 2 vanuit CHARF aangeroepen, dan kan CHARF zelf, voor verletting met "basint := 2" dit neveneffect weer ongedaan maken.)

De initialisering van CHARF en CHART zijn de enige manier om basint = 2 of = 3 te krijgen; verder zal de programmeur basint = 0 of basint = 1 mogen zetten.

De keuze tussen symhiep, symcharf en symchart.

De keuze tussen symhiep, symcharf en symchart hangt af van de heersende waarde van basint. Maw. wie weet, dat zijn SYM via symcharf zal moeten werken, kan zulks bereiken, door voor de eerste aanroep van SYM de CHARF te initialiseren; evenzo kan hij a priori interpretatie volgens symchart of symhiep via basint dicteren. Zodoende is het mogelijk om via SYM flexowriterbanden te lezen, zonder dat deze zich in hun aanhef als zodanig identificeren. Willen we echter de keuze niet a priori opleggen, maar van de band laten afhangen, dan kunnen we dit doen door

- 1) voor de eerste aanroep van SYM de variabele basint op 0 in te stellen.
- 2) ponsconventies te eisen, zodat aan het begin van de band te zien is, met welke waarde van basint tot nader order doorgelezen moet worden. SYM kan dan zelf zó nodig CHARF of CHART actualiseren. (Ik zou voor de flexowriteraanhef willen voorstellen na skippen van blanks erases (en stopcodes?) eerst een NLCR of een casedefinitie, voor teleprinterbanden na skippen van blanks eerst een CR, een NL of een casedefinitie; de door SYM ingelaste CHARF- of CHART-initialisatie zal voor de onbekende parameter een standaardkeuze doen, bv. begin van de regel, lower case dan wel letter-shift.)

Om tot normalisering van deze aanheffen te komen doe ik een beroep op alle belanghebbenden. Ik verwacht dan, dat iedereen zijn ponsreij zo zal instrueren, dat ~~XXX~~ de aldaar geponste banden -speciaal ALGOL-programma's en gegevensbanden- conform die conventie aanheffen, ook al heeft hij dit voor eigen bedrijf niet nodig. Tevens verwacht ik van elke beperkte implementatie van eigen bedrijf (zeg "alleen flexowriter banden"), dat die flexowriterbanden met genormaliseerde aanhef accepteert.

Initialisering van SYM.

Wat betreft de keuze van interpretatieproces is de status van SYM dus geregeld door basint. We zitten er nog mee, dat ook SYM (juist SYM!) een voorgift van een enkele waarde moet kunnen hebben. Definieren we het geven van een voorgift door overschrijving van een eventueel nog hangende voorgift (als bij CHAR) dan kunnen we SYM in dit opzicht initialiseren door een voorgift te geven en 1 maal SYM aan te roepen.

Initialisering van NUM.

De interpretatiewijze van SYM is volledig door basint bepaald. Dit geldt niet voor NUM: immers als basint bv. CHARF als actueel aanwijst, dan ~~XXXXX~~ hebben we nog de keuze tussen ~~NUMXXXXXXXNUMXXXXXXXNUM~~ numcharf en numsym via symcharf. NUM heeft dus nog een boolean statusvariabele, die we ~~XXX~~ "numint" noemen en die aangeeft of numsym actueel is. Is numint true, dan bepaalt basint slechts de keuze tussen symcharf, symchart en symhep; is numint false, dan bepaalt basint tevens de keuze tussen numcharf, numchart en numhep. De boolean "numint" zij in beide richtingen zetbaar.

Ik stel voor om de derde denkbare toestand (numint ongedefinieerd) te laten vervallen en in alle twijfelgevallen numint true te zetten. (Aan het begin van een nieuwe band.) Dit is een keuze, die aan de interpretatiewijze numsym een zekere voorkeur geeft.

Immers: laat -onder voortdurende actualiteit van CHARF- NUM een tijdlang via numsym gewerkt hebben en laten we nu op de Flexowriterpagina willen aangeven, dat van nu af aan tot nader order NUM via numcharf moet werken. Het feit, dat numint false moet worden, moet worden gemeld door de band, met wie NUM slechts via SYM contact heeft. Dit kan op verschillende manieren; ik noem

- 1) door af te spreken, dat het optreden van bepaalde getalscheiders door NUM geïnterpreteerd zal worden als "ga over op numint = false"
- 2) door een extra basic symbol in te voeren met deze betekenis
- 3) door alles, wat niet als basic symbol interpreteerbaar is als "nonsense" door te melden en diezelfde betekenis te geven.

(Ik ga steeds meer voor de eerste mogelijkheid ~~XXX~~ voelen.)

In alle gevallen kunnen we bereiken, dat een band niet via numsym gelezen gaat worden, door in de aanhef dit aan te geven. Het resultaat is, dat NUM banden leest via numsym, tenzij anders vermeld. Dit lijkt in overeenstemming met wat we wensen.

De verwerking van "EINDE BAND".

Tot nog toe hebben we alleen erover gesproken, hoe HEP op einde band reageert. Daar hebben we afgesproken, dat het optreden van EINDE BAND mogelijk wel tot de semantische informatie behoort en HEP is niet uitgerust met automatische aanvraag van volghand. Nu moeten we dit voor de volgende niveaus (CHAR, SYM en NUM) beslissen.

Ook op het niveau van CHAR zou ik nog geen automatische volgbandaanvraag willen inbouwen. Sterker: ik vond het hachelijk om de door CHAR veronderstelde codecontinuïteit zich over fysisch gescheiden stukken band te laten uitstrekken en stel daarom voor CHARF -wanneer hij via HEF "EINDE BAND" detecteert- op deze detectie te laten reageren als op de stopcode en "EINDE CODE" te laten doormelden (en dan \bar{X} basint := 0 uit te voeren). Het CHAR aanroepend programma kan dan via HEF inspecteren of er nog meer band is, zo nee, beslissen of het meer wil zien etc.

De consequentie hiervan is de eis, dat als we met SYM in diverse codes willen kunnen lezen, dan in elk geval op het begin van elke fysische band de code-identificerende aanhef staat. Is aan deze voorwaarde voldaan, dan kunnen we een lange band uit een aantal deelstukken assembleren door plakken; wanneer alle deelstukken eindigen met de stopcode, dan kan dit plakproces zelfs zonder er aandacht aan te besteden of bij die plakjes codewisselingen optreden.

Een en ander voel ik als streke aanwijzing om SYM wel met automatische aanvraag van volgband te versieren. (Onder het motto: het hoeft tot het SYMmend programma niet door te dringen, dat die logisch nu zonder meer toegestane plakjes misschien niet de facto zijn uitgevoerd). En ik stel me zelfs voor om op dit niveau aan het fysisch einde band geen semantische betekenis meer toe te kennen (en dan ook consequent, dus niet bv. nog een keer doormelden). Dat een programma, dat basic symbols van een band pulkt dan gegarandeerd ongevoelig is voor de gebruikte code en voor het fysisch aantal banden lijkt me een groter goed dan het verlies van de mogelijkheid om met SYM een programma te schrijven, dat het aantal basic symbols op een band telt.

Het is aan SYM om nieuwe band aan te vragen, en wel, wanneer tijdens inspectie (basint = 0) blijkt, dat HEF de waarde -1 aflevert. Het aanvragen van een nieuwe band hebben als neveneffect, dat numint = true gezet wordt.

Nog enkele opmerkingen over de synchronisatie.

We nemen aan, dat NUM via numsym de getallen als niet zelfafsluitende eenheden uit de basic symbol stroom pulkt. (B.v. het teken van het volgende getal kan als separator gebruikt worden.) Als alleen NUM gebruikt zou worden kan dit op twee manieren gerealiseerd worden.

- 1) het teveel gelezen basic symbol wordt in SYM teruggeduwd
- 2) het teveel gelezen basic symbol wordt in NUM opgeslagen.

Zodra we alternerend NUMmen en SYMmen maakt dit wel verschil. Ik geloof, dat iedereen methode 1) verkiest.

Als echter SYM in termen van CHARG gedefinieerd is, dan zij nu ook een alternering tussen NUM en CHARG gedefinieerd. Men lette er op, dat na NUM het CHARFen het paginabeeld aftast te beginnen achter het teveel gelezen basic symbol!

Wie de band "zonder overslaan" sequentieel wil aftasten zal na NUM eerst een keer SYM moeten doen en dan pas CHARFen. Een open vraag is, of SYM zijn voorpift moet behouden bij rechtstreeks CHARFen, CHARTen of HEFpen. Ik ben geneigd deze vraag ontkennend te moeten beantwoorden; anders heeft nl na CHARF herhaald SYMmen niets meer te maken met in successie aftasten en dat kon wel eens \bar{X} "a trouble shooter's nightmare" worden.

Het laatste getal.

Lijkt ons de opgave "tel de basic symbols op deze band" wat gezocht, programma's die een onbekend aantal getallen moeten verwerken zijn heel bekend. De truc "zet voorop het aantal" kennen we allemaal en we weten allemaal, hoe hinderlijk deze kan zijn.

Ik zou me kunnen voorstellen, dat we naast NUM een boolean procedure NUM*MARK introduceren en dit tweetal samen definiëren op een willekeurige rij van numerieke waarden en zg, "marks" (waarbij een mark als getalscheider fungeert).

De boolean procedure NUM*MARK geeft antwoord op de vraag of er bij doorlezing eerst een "mark" komt, zo ja, dan is daarmee deze mark gepasseerd. In de diverse codes moet dit realiseerbaar zijn, zonder NUM meer statusvariabelen te geven. (Als we HEP niet met de terugduwfaciliteit uitrusten kost dit ons een heptade per getal, maar daarover zou ik niet willen vallen).