EWD157.html

Over de sequentiele interpretatie van een ponsband.

HEP

Als fundamentele bouwsteen beschouw ik de integer procedure, die ik HEP noem. Bij het lezen van n-gats band levert hij in eerste instantie een van 2↑n mogelijke waarden af, nl van en met 0 (voor blank tape) tot en met 2↑n-1 (voor all holes) en wel steeds de waarde die overeenkomt met de volgende n-ade.

Hoe we aan de band komen is een onderdeel van de initialisering van HEP, waarover ik me liever niet wilde uitlaten. Ik neem dus aan, dat we weten, over welke band we het hebben. Die kunnen we dan n-ade voor n-ade aftasten: die band kan dan wel opraken!

De vraag rijst direct, hoe HEP moet reageren, als "einde band" gedetecteerd wordt.

Mijn vóorstel is om het arsenaal van mogelijke, door HEP af te leveren waarden uit te breiden met een speciale waarde (ik denk aan 511 of -1), dáe "EINDE BAND" zal betekenen. Het is dan aan het programma om vast te stellen, of het nu genoeg gelezen heeft; zo nee, dat zal HEP opnieuw geinitialiseerd moeten worden (band aanvragen).

Onmiddellijk rijst de vraag, hoe HEP reageert, als nadat EINDE BAND" als waarde is afgeleverd, HEP weer wordt aangeroepen, voordat de volgende initialisatie heeft plaatsgehad.

Nu kunnen we drie kanten uit.

1) we kunnen dit interpreteren als letaal deraillement van het programma
2) we kunnen dit interpreteren als de indicatie, dat er dus blijkbaar meer informatie gevraagd wordt en als onderdeel van HEP implicíet de initieringsriten plegen (om díeuwe band vragen etc.)
3) we kunnen weer "EINDE BAND" afleveren.

Omdat ik HEP beschouw als mijn meest fundamentele bouwsteen prefereer ik het laatste; het maakt het gerbuik~~XXX~~ van HEP mogelijk zonder na elke aanroep te inspecteren, of "EINDE BAND" soms voorkwam "X:= HEP * HEP" kan nu. Een band wordt nu een halfrechte, oneindig lang in de toekomst, vanaf een zeker moment komen er alleen maar ~~XXXXXXXXXXXXXXXXXX~~ "EINDE BAND"'s op voor.

De routíne HEP is van toepassing, als we als band willen verwerken, bv. de output van een registrerend meetapparaat. De rest van deze notitie beschouwt de inhoud van de ponsband als derivaat van een paginabeeld: hier ligt een code aan ten grondslag, bv. telexcode of FLEXOWRITER code.

CHAR

In het volgende zal ik een en ander toelichten aan de hand van Flexowriter code; voor telexcode is een analoge beschouwing te houden, zolang er geen losse carriage returns zonder line feed gegeven worden.

En ik moet me zelfs beperken tot een vrij speciale Flexowriter: een Flexowriter met vaste spatiering, zonder backspace waarbij de tabulatorstoppen op vaste posities staan. (En dit apparaat zullen wij zelfs ~~X~~ nog iets moeten idealiseren: over de minimum afstand, waarin de tabulatie "het nog haalt" , bv. 1 niet, twee wel, moeten we een vaste afspraak maken.)

CHAR nu is een routine, die zich interesseert voor het paginabeeld: het is de functie van CHAR om bij het aftasten van een band een verslaglegging te doen, die een-eenduidig is toegevoegd aan het met de band corresponderende paginabeeld.

Ik stel me bij de definitie van het paginabeeld op het standpunt, dat
1) meervoudige underlining niet van enkelvoudige underlining is te onderscheiden
2) ~~XXX~~ meervoudige "stroking" niet van enkelvoudige "stroking" is te onderscheiden
3) dat de op de wagen ingestelde afstand van de regelopvoer niet tot het paginabeeld behoort, een extra regel blank wel
4) dat op het papier de plaats van de linker kantlijn bekend is (wie elke regel begint met een extra spatie fokt een ander paginabeeld)

Ik stel me voor, dat CHAR bij herhaald aanroepen dit verslag zal afleggen, daarbij ~~X~~ het paginabeeld regel voor regel verslaand (in volgorde van ~~XXXXX~~ boven naar beneden) en elke regel op zichzelf van links naar rechts.

Over de wijze van verslaglegging door CHAR zullen we enige conventies moeten treffen.

Ik stel me voor, dat CHAR niet elke regel met spaties tot een vaste maximum lengte hoeft aan te vullen, maar in plaats daarvan NLCR wel tot de output van CHAR te laten behoren; CHAR heeft dan wel de plicht om eventuele TABS en SPATIES aan het einde van de regel te onderdrukken.

Voor de verwerking van de TAB's stel ik voor ze te vertalen in een aequivalent aantal spaties: op deze wijze geeft een reeks CHAR-waarden zonder NLCR een beeld, dat invariant is voor zijn plaats op de regel. Wel zal CHAR voor het bepalen van deze aequivalentie de wagenpositie moeten bijhouden.

Wat betreft de underlining en de stroking kunnen we verschillende wegen bewandelen.
1) elke "zichtbare, transporterende ponsing" bestaat in vier versies, al of niet met underlining, al of niet met stroking
2) een aanroep van CHAR kan wel "underlining" en "stroking" separaat doormelden. In dit geval moet, om der wille van de eenduidigheid een vaste afspraak gemaakt worden over de volgorde, waarin deze ~~afsp~~ door CHAR worden doorgemeld (bv. eerst de underlining, dan de stroke en dan het transporterende karakter).

Omdat CHAR dan toch verder moet kijken dan zijn neus lang is, prefereer ik de eerste mogelijkheid. Deze heeft de voordelen, dat
1) we van de willekeurige afspraak over de volgorde verlost zijn
2) nu met elke aanroep van CHAR een positieverplaatsing ~~XXXXX~~ van de wagen correspondeert.

CHAR heeft de eigenschap, dat mist de code bekend is, elke produceerbare band gelezen kan worden en over elke produceerbare pagina een uniek verslag wordt afgelegd.

Een willekeurige Flexowriterpagina is niet als willekeurige Telexpagina te interpreteren, noch omgekeerd. Ik stel daarom voor, om de codeonafhankelijke CHAR ~~(XXXXXXXXXX)~~ af te schaffen en zoveel CHAR's in te voeren, als we codes (i.e. apparaten)

hebben. Dus CHARF voor de verslaglegging over een Flexowriterpagina en CHART over de verslaglegging over een Telexpagina. (En wie naast de beschreven Flexowriter ook een Flexowriter met variabele spatiering etc. wil kunnen bespelen, zal daarvoor een andere CHARF, zeg CHARFV, moeten maken.)

De toevoeging van de integerreeks, die CHARF aan een Flexowriterpagina toekent heeft niets te maken met de integerreeks, die CHART aan een telexpagina toekent. Ik beschouw op dit niveau Flexowriterpagina en Telexpagina nl. als onvergelijkbare voorwerpen.

Kijk, CHARF mag twee Flexowriters,die alleen daarin verschillen, dat de ene een recht lettertype en de ander een cursief lettertype heeft, als aequivalent beschouwen (mits natuurlijk beide lettertypes het zelfde onderscheidende vermogen -zoals verschil tussen de letter O en het cijfer 0 hebben).

CHART mag twee teleprinters met de ALCOR symbolen als aequivalent beschouwen, ook al heeft de ene alphabet van kleine letters en de andere een alphabet van hoofdletters.

CHARF en CHART leggen verslag af over het paginabeeld en moeten elke pagina, die op het betrokken apparaat vervaardigbaar is, uniek verslaan; dit heeft nog niets te maken met ALGOL basic symbols! Er is een aparte beslissing nodig om de Flexowriter "A" resp."a" op papier te gebruiken als afbeelding van hetbasic symbol "A" resp. "a". Evenzo zal apart beslist moeten worden, dat het telexcharacter, dat verdacht veel lijkt op een van de gebruikelijke afbeeldingen van de eerste letter van het alphabet, geidentificeerd kan worden met hetzij het ALGOL basic symbol "A", dan wel het ALGOL basic symbol "a". Hoe deze beslissing uitvalt is een kwestie, die helemaal los staat van het feit, of teleprinters meestal met kleine, dan wel meestal met hoofdletters zijn uitgerust!

Ik zou mij kunnen voorstellen, dat het practisch zal blijken om het bereik van CHARF en dat van CHART een lege doorsnede te geven!

HEP is het aangewezen inspectiemiddel, als de programmeur in het bandbeeld geinteresseerd is, CHARF en CHART zijn de aangewezen inspectiemiddelen, als de programmeur in het paginabeeld geinteresseerd is. Een programmeur, die, bij wijze van spreken in manuscript een ALGOL-programma aanbiedt, is niet in het bandbeeld geinteresseerd, noch in de verwerking van het paginabeeld: zijn manuscript geeft in zijn prive-hardware representatie (nl. zijn handschrift en notatieconventies) een reeks basic symbols van ALGOL 60 en hij verwacht van de ponskamer (ik zou hier nog liver willen zeggen "typekamer"), dat deze reeks basic symbols op een van de voorhanden zijnde paginabeeldmakers (Flexowriters of teleprinters) wordt afgebeeld in overeenstemming met de specifieke regels, die voor elk van deze apparaten met in acht name van hun beeldend vermogen gekozen zijn. Wie aldus een text heeft aangeboden, mag <u>niet</u> protesteren als waar hij "<u>Boolean</u>" heeft geschreven, hij "<u>boolean</u>" op zijn papier vindt, tenminste niet, wanneer het laatste een van de volgens afspraak toelaatbare representaties is van het basic symbol, dat vermoedelijk bedoeld is. Was het niet zijn bedoeling om een ALGOL programma aan te bieden, maar een op een Flexowriter produceerbare text, dan is een dergelijke vrijheid natuurlijk niet veroorloofd (en waarschijnlijk doet hij er dan beter aan om zelf zijn tekst te Flexowriteren).

Fungeert het paginabeeld selchts als "carrier" voor een reeks ALGOL basic symbols. dan is het aangewezen inspectiemiddel een integer procedure "SYM". Het waardebereik van SYM heeft niets te maken met dat van CHARF en CHART, het onderscheidt waarschijnlijk alle basic symbols van ALGOL 60 (met uitzondering van het overbodige wordtteken?) en waarschijnlijk nog wel wat meer, NLCR bv.

SYM is een integer procedure, die heel nadrukkelijk slechts in enkelvoud bestaat; dus geen SYMF of SYMT.

Het afzondelijke bestaan van CHARF en CHART is gerechtvaardigd omdat wij hiermee paginabeelden kunnen aftasten,vervaardigd op apparaten met heel verschillende (Onvergelijkbare) beeldende vermogens. CHARF en CHART fungeren als optische aftasters; dat de facto de machine via een bandbeeld toegang tot het paginabeeld krijgt, is een omstandigheid, waarvan bij het gebruik van CHARF en CHART geheel geabstraheerd is, evenals van het feit, dat het feitelijke bandbeeld, dat slechts als "carrier" voor het paginabeeld fungeert, door het laatste niet eenduidig bepaald is. Aan deze abstractie ontlenen CHARF en CHART juist hun bruikbaarheid. De toevoeging van paginabeeld aan bandbeeld is bepaald door de eigenschappen van de respectievelijke apparaten.

Bij het gebruik van SYM fungeert, zoals gezegd, het paginabeeld op zijn beurt als carrier voor een reeks ALGOL basic symbols; de toevoeging van deze reeks aan een paginabeeld is nu bepaald door de conventies van de typekamer. Zoals CHARF en CHART abstraheren van het niet eenduidig bepaalde bandbeeld, zo abstraheert SYM van het niet eenduidig bepaalde paginabeeld. Het is deze abstractie, waaraan SYM zijn enorme bruikbaarheid ontleent; het is deze abstractie, die de programmeur ontheft van de plicht het paginabeeld aanslag voor aanslag,(spatie voor spatie!) te preciseren, het is tevens deze abstractie, die de ponskamer de vrijheid geeft in de keuze van het ponsapparaat en de layout.

Alle genoemde routines hebben hun stilzwijgend veronderstelde "own variables". Voor HEP is dit, waar welke band wordt afgetast, voor CHARF en CHART zijn dit de heersende case shift, de wagenpositie etc, voor SYM is dit onder meer de heersende code, dwz. of hij via CHARF, dan wel CHART moet werken. Omdat de afbeeldingsregels voor basic symbols op een pagina apparaatafhankelijk zijn, is deze heersende code de statusvariabele van SYM; dit is dan ook de reden, waarom ik CHAR heb afgeschaft.

Boven gegeven uitweidingen zijn de vrucht van een mislukking. Toen ik HEP, CHARF en CHART geconcipieerd had, ben ik gaan denken over wat men doen moet, als een programma, na via, zeg, CHARF een stuk band gelezen te hebben, via HEP een stukje doorleest, om daarna desnoods via CHARF weer door te kunnen lezen. Een vraag, die opkomt, zolang men CHARF beschouwt als een speciale manier van bandlezen, een vraag, die zinloos wordt, zodra men CHARF beschouwt als een mechanisme tot paginaaftasting. Zolang men CHARF beschouwt als een speciale manier van bandlezen en dus probeert om aan de alternering van CHARF en HEP een betekenis toe te kennen, duiken er allerlei vieze problemen op, waarvoor ik geen overtuigende oplossing heb kunnen vinden. Ik heb nog geprobeerd om naast HEP een "HEPF" en een "HEPT" in te voeren, die de plicht zouden hebben, voor CHARF, resp CHART de passerende heptade aan te kijken en de relevante updating van de statusinformatie van CHARF resp. CHART te verrichten. Nou, dit is mislukt, speciaal omdat er geen duidelijke manier is om de beide sequentiele processen -het aftasten van een band en het aftasten van een uit deze band afgeleid paginabeeld- , hoewel ze macroscopisch gezien wel synchroon lopen (het begin van de band komt overeen met het begin van het paginabeeld), microscopisch door een zinvolle (dwz. hanteerbare) conventie te synchroniseren. Ik kom dan ook tot de conclusie, dat het gemengde gebruik van HEP en CHAR een hinken op twee gedachten is, waarin je met de ene hand probeert terug te nemen, wat je met de andere gegeven hebt (nl. dat het precieze bandbeeld er niet toe doet). Ten aanzien van dit gemengde gebruik zou ik dan ook het standpunt willen innemen: of verbieden (en de implementatie hierop laten testen) of er bij zeggen, dat het undefined is, en dat de gebruiker wilde reacties mag verwachten, onder het motto "What else do you expect?".