



18.2.69

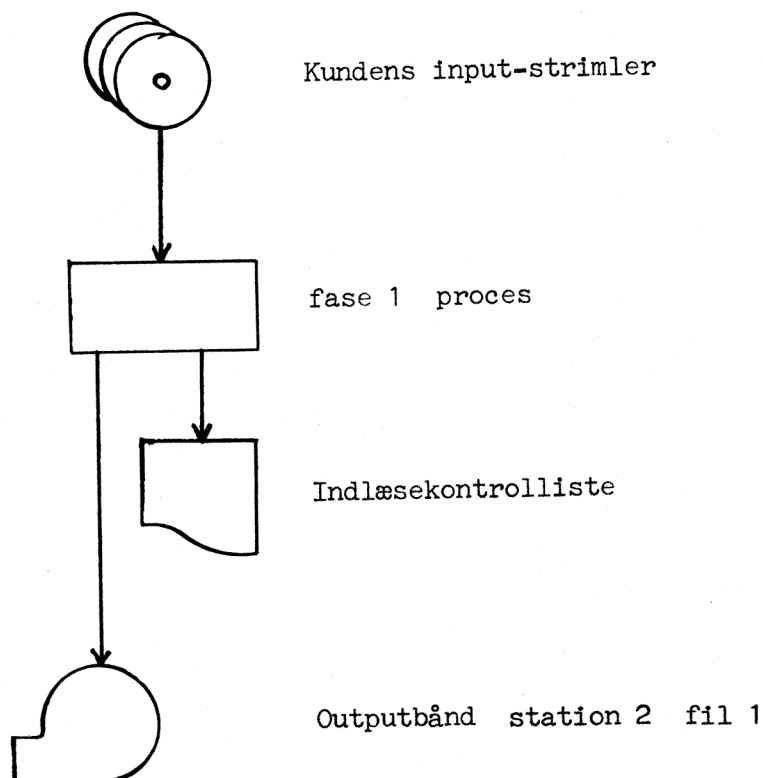
BØT

Fase 1 oversigt.

Fase 1 er RC-FAKTs indlæseprogram. På grundlag af kundens papirstrimler dannes et outputbånd med samtlige korrekte posteringer og kartoteksændringer. Samtidig skrives en indlæsekontrolliste på printeren dvs. en oversigt over alle posteringer, korrekte så vel som kasserede.

Inputsyntaxen ligger delvis fast i programmet og kan delvis ændres ved konsulenttrimningen. De enkelte inputarter er defineret i en række tabeller (der er trimbare), hvorimod den væsentligste del af syntakschecken sker i en fast aktionstabel.

Outputbåndet genereres specielt for hver inputtype i et stykke specialkode.



I N D H O L D S F O R T E G N E L S E

1)	<u>Definition af input</u>	1.0.
	Linietype-katalog.....	1.1.
	Liniespecifikationstabel.....	1.2.
	Grundelementtabel.....	1.3.
	Tegntabel.....	1.4.
	Tegnklasser.....	1.5.
	Hent linie specifikation.....	1.6.
2)	<u>Overordnet programlogik</u>	2.0
	Tilstande.....	2.1.
	Aktionstabellen.....	2.2.
	Aktions-indhops-tabel.....	2.3.
	Generel administration.....	2.4.
	Hent aktionsparameter.....	2.5.
3)	<u>Indlasesekvenser</u>	3.0.
	Læs tegn.....	3.1.
	Læs tal.....	3.2.
	Læs 0-tal.....	3.3.
	Læs text.....	3.4.
	Læs dato.....	3.5.
4)	<u>Aktioner</u>	4.0.
	Aktion 1.....	4.1.
	Aktion 2.....	4.2.
	Aktion 3.....	4.3.
	Aktion 4.....	4.4.
	Aktion 5.....	4.5.
	Aktion 6.....	4.6.
	Aktion 7.....	4.7.
	Aktion 8.....	4.8.
	Aktion 9.....	4.9.
	Aktion 10.....	4.10.

	Aktion 11.....	4.11.
	Aktion 12.....	4.12.
	Aktion 13.....	4.13.
	Aktion 14.....	4.14.
	Læs totallinie.....	4.15.
5)	<u>Specialkode.....</u>	5.0.
	Pak bogholderiposteringer.....	5.1.
	Pak købsnota.....	5.2.
	Pak følgeseddel.....	5.3.
	Pak ændringer og sletninger.....	5.4.
	Akkumuler beløb i købsnota.....	5.5.
	Test type i ændringer.....	5.6.
6)	<u>Output.....</u>	6.0.
	Indlæsekontrollisten.....	6.1.
	Fejltekst.....	6.2.
	Writecr.....	6.3.
	MB-register.....	6.4.
7)	<u>Diverse.....</u>	7.0.
	Nulstil.....	7.1.
	Start fase 1.....	7.2.
	Afslut fase 1.....	7.3.
8)	<u>Test-faciliteter.....</u>	8.0.
	Betingelser.....	8.1.
	Indhop og udhop.....	8.2.
	Udskrifter.....	8.3.
	Tabeller.....	8.4.

1. Definition af input

1.0. Opbygning

Inputmaterialet består af en række strimler (fysiske strimler), der er opdelt i 'logiske strimler'. En logisk strimmel består af en række bilag efterfulgt af et strimmel-slut-tegn.

Et bilag er opbygget af en bilagslinie efterfulgt af en række kropslinier. Betegnelsen bilag benyttes som fællesbetegnelse for de forskellige input-arter (dvs. også for inputmateriale, som normalt ikke benævnes på denne måde f.eks. kartoteksændringer).

Bilagslinien består af et bilagsstart-tegn efterfulgt af et tal, 'bilagstypen', der benyttes til at identificere bilaget med. Efter bilagstypen følger en række oplysninger adskilt med et specielt terminator-tegn og afsluttet med et linie-slut-tegn.

En kropslinie indledes med et specielt liniestart-tegn efterfulgt af et heltal, 'linietypen', der identificerer den enkelte kropslinie. Efter linietypen følger en række oplysninger på samme måde som i bilagslinien.

En oplysning er enten et heltal, en tekst eller en dato. Der skelnes mellem tal og datoer, fordi datoer lagres på en speciel måde.

Ovennævnte struktur kan ændres ved annulleringer, hvor der skelnes mellem bilagsannulleringer og linieannulleringer. En bilagsannullering vil annullere alt fra og med sidst læste bilagsstart til (men ikke med) næste bilagsstart. En bilagsannullering må forekomme overalt i et bilag.

En linieannullering må kun forekomme i en kropslinie, og vil annullere alt fra og med sidst læste liniestart til (men ikke med) næste liniestart eller bilagsstart (alt efter hvad der kommer først).

Den syntaktiske opbygning af inputmaterialet er defineret i BAC-DUN-notation i appendix A.

1.1. Linietypekataloget

Linietypekataloget er delt i 3 dele: I den første del står der oplysninger om bilagslinierne, hvilket igen er generelle oplysninger om hele bilaget. I den anden del står der oplysninger om de kropslinier, der ikke er kartoteksændringer. I den tredje del er ændringer behandlet.

1.1.1. Bilagslinietypekataloget

Kataloget har 13 indgange - 1 for hver bilagstype svarende til værdierne af bilagstyperne (1-13)..

For hver indgang findes følgende oplysninger:

- linespecifikationsreference, der giver index i linespecifikationskataloget (se afsn. 1.2.), hvor opbygningen af bilagslinien er defineret.
- hovedtype
- undertype, to størrelser, der gemmes i de af fase 1 dannede individer til identifikation i de senere faser.

- basisadresse i den resterende del af linietypekataloget (se nærmere i de følgende afsnit).
- intern bilagstype (= 0). Denne størrelse benyttes til undersøgelse af, om en linietype er korrekt (se afsn. 1.1.4.).
- ny intern bilagstype. Denne størrelse benyttes til sammenligning af den interne bilagstype (se afsn. 1.1.4.).

1.1.2. Kropslinier (- ændringer)

Kataloget har en indgang for hver type kropslinie og indeholder følgende oplysninger:

- liniespecifikationsreference, der sammen med basisadressen fra bilagslinien giver et index i linespecifikationstabellen, hvor opbygningen af kropslinien er defineret.
- hovedtype
- undertype, identifikation af linien i de senere faser.
- intern bilagstype (se nærmere i afsn. 1.1.4.).

1.1.3. Ændringer

Kataloget har en indgang for hvert oplysningsnummer i det tilsvarende kartotek. For hver indgang fås følgende oplysninger:

- liniespecifikationsreference, der sammen med basisadressen fra bilagslinien giver et index i linespecifikationstabellen, hvor opbygningen af ændringslinien er defineret.
- relativ kartotekscelle angiver den relative placering af den celle, der skal ændres. Oplysningsnummeret, der blev nævnt ovenfor, er en størrelse, der kun benyttes af kunden (eller under omlægningen til RC-FAKT).
- oplysningslængde, der er en definition af længden af det område, der skal ændres (hvis det er et heltal eller en dato, er tallet 1, hvorimod det for tekster er 5).
- bit-nr. Denne størrelse har kun betydning for debitorregistrets valgfri booleans, idet bit-nr. angiver placeringen af en enkelt boolean indenfor en celle.
- intern bilagstype (se nærmere i afsn. 1.1.4.)
- add/placer-kode, der definerer, hvilke ændringer, der er tilladt til den aktuelle oplysning i kataloget.
 - Hvis koden = 1, kan ændringen kun ske ved at placere noget nyt i cellen.
 - Hvis koden = 2, kan ændringen kun ske ved at addere (eller subtrahere) noget til (fra) cellen.
 - Hvis koden = 3, kan ændringen ske på begge måder.
 Hvilken ændringsform, der skal benyttes i det aktuelle tilfælde defineres i ændringslinien (=kropslinien).

1.1.4. Test af bilags/linietyper

For let at kunne undersøge, om en linie må forekomme i det aktuelle bilag, er der i linietypekataloget for hver linietype en størrelse ('intern bilagstype'), der giver oplysning om, hvilket bilag den pågældende linie hører til.

Dette betyder, at der en gang for hvert bilag fastlægges en størrelse, der skal sammenlignes med den interne bilagstype for kropslinierne. Da denne størrelse er fælles for hele bilaget, findes den i bilagslinietypekataloget ('ny intern bilagstype').

Hver gang, der slås op i linietypekataloget (hvilket sker i aktion 2 og 14), bliver den interne bilagstype undersøgt. Hvis den ikke har samme værdi som referencestørrelsen, bliver linien kasseret (med fejludskriften 'linietype'). Bilagslinietyperne kan testes på denne måde, idet referencestørrelsen nulstilles ved afslutningen af hvert bilag (aktion 5).

1.2. Linienspecifikationskataloget

Fra linietypekataloget er der for hver linie givet en reference til linienspecifikationskataloget. Ved opslag med denne reference som index, fås en samling referencer til grundelementkataloget. Hver reference svarer til en oplysning, der skal indlæses, og rækkefølgen af referencerne definerer indlæse-rækkefølgen. Referencer til linienspecifikationskataloget sker altid via sekvensen 'hent linienspec', der er omtalt i afsn. 1.6.

1.3. Grundelementkataloget

Grundelementkataloget definerer dels, hvad der skal læses (det kan være et heltal, en tekst eller en dato), og dels hvor det indlæste skal lagres.

Afhængig af, hvad der læses, findes der desuden følgende oplysninger:

1.3.1. Tallæsning

- angivelse af, om det læste tal skal indgå i det indlæste bilags total (se nærmere om totaler i afsn. 4.15).
- taltilføjelse af hensyn til senere sorteringer (se nærmere under aktion 3, afsn. 4.3.1.).
- checkcifferkontrol (se nærmere under aktion 3, afsn. 4.3.5.).
- intervalgrænser. Grundelementkataloget indeholder oplysning om, der skal tages intervaltest (se aktion 3, afsn. 4.3.1.), og hvis dette er tilfældet, findes dels mindste laveste værdi, og dels antallet af tilladte værdier angivet.

1.3.2. Tekstlæsning

- angivelse af, hvor mange ord (å 6 tegn) teksten må fylde. I øjeblikket (april 69) er dette altid 5 ord (= 30 tegn).

Tekster kan ikke indgå i totaler.

1.3.3. Datolæsning

-angivelse af, om den læste dato skal indgå i totalen. Datoer pakkes på en særlig måde ($384 \times \text{år} + 32 \times (\text{md}-1) + \text{dag}$), men indgår som almindelige heltal i en eventuel total.

1.4. Tegntabellerne

En tegntabel har to funktioner. Dels opdeler den tegnene i klasser efter den funktion, tegnene har (tegnklasserne er beskrevet i afsn. 1.5.). Desuden konverterer den de eksterne værdier til interne værdier, hvorved man i programmet bliver uafhængig af tegnenes talværdier på strimlen.

I fase 1 er der to tegntabeller, hvilket vil sige, at man under indlæsningen kan køre med to forskellige strimmetyper.

Hver tegntabel har 66 indgange. Den første (index -1) definerer, hvordan læsningen skal foregå. Dels angives det, om paritetsfejl er tilladt, og i så fald adderes der 32 til tegnværdien (dette er nødvendigt af hensyn til 5-kanals strimler) og dels angives det, om tal er hullet 'normalt' eller efter Olivetti-konventioner (dvs. bagfra).

Tegntabellen skelner ikke mellem UPPER CASE-tegnet og LOWER CASE-tegnet, hvilket vil sige, at f.eks. tallet 326 kan hules som \sqrt{x} uden at der kommer syntax-fejl. En langt mere betydningsfuld konsekvens af dette er dog, at speciel-tegnene (som f.eks. bilagsstart og liniestart) ikke kan være 'almindelige' tegn i UPPER CASE.

I den resterende del af tabellen benyttes den eksterne tegnværdi som index (dvs. værdier fra 0 til 64), og fra tabellen fås tegnklassen og den interne værdi.

Den interne værdi af et tegn har kun betydning, når tegnet indgår i tal, tekster eller datoer (dvs. tegnklasserne 1, 2 og 3), og i disse tilfælde skal den interne værdi svare til flexowriterkoden, så f.eks. en tekst ikke skal konverteres når den skrives ud.

Første ord i tabellen benyttes kun i aktion 7 (se afsn. 4.7.) til at modificere visse programdele med. Den egentlige tegntabel behandles udelukkende af sekvensen 'læs tegn' (se afsn. 3.1.).

1.5. Tegnklasserne

I det følgende er de enkelte tegns funktioner beskrevet klasse for klasse. Eksemplerne er taget fra standard-tegntabellen for flexowriterstrimler.

Klasse	Indhold
0	Totalt blinde tegn. (F.eks. TAPE FEED) disse tegn bliver altid oversprunget uanset i hvilken forbindelse de kommer. ALL HOLES, der ligger udenfor tegntabellens område (værdi 639), bliver behandlet som et tegn fra klasse 0.
1	Talblinde teksttegn (f.eks. SPACE) er tegn, der er blinde ved tallæsning, men som indgår normalt i tekster.
2	Cifre. Ud over tegnene i klasse 0 og 1, er disse de eneste tegn, der kan indgå i tal (og datoer).

Klasse	Indhold
3	Teksttegn (f.eks. bogstaver) er tegn, der ikke findes i forvejen i klasse 1 eller 2, og som skal kunne indgå i tekster.
4	Terminatorer. Tegn af denne klasse er de eneste, der kan benyttes som skilletegn mellem to oplysninger. (f.eks. TABULATOR).
5	Bilagsstart. Dette tegn skal indlede ethvert bilag og er dermed samtidig starttegn for bilagslinien. (f.eks. <).
6	Liniestart. Tegnet angiver starten på en kropslinie. (f.eks. _ understregning).
7	Linieslut. Tegnet skal afslutte enhver linie (både bilagslinie og kropslinie) og er samtidig terminator for sidste oplysning i hver linie (f.eks. vognretur).
8	Strimmelslut. Tegnet (f.eks. STOP CODE) bør afslutte enhver strimmel, men virkningen af dette tegn kan provokeres manuelt af operatøren (ved at trykke MICROTEMPI STOP efterfulgt af NORMAL START når maskinen venter på strimmellæseren).
9	Linieannullering. Tegnet (f.eks. PUNCH ON) må forekomme overalt i en kropslinie efter liniestart-tegnet, og bevirker, at linien springes over (indtil næste linie- eller bilags-start) uden udskrift.
10	Illegale tegn. Hertil hører alle tegn, der ikke må forekomme på strimlerne (f.eks. tegn med paritetsfejl, END CODE osv.).
11	Bilagsannullering. Tegnet (f.eks. PUNCH OFF) må forekomme overalt i et bilag efter bilagsstart-tegnet, og bevirker, at bilaget bliver kasseret med udskriften 'bilags annull'. Læsningen fortsætter med næste bilagsstart-tegn.

1.6. Hent linie-spec

Da oplysningerne i liniespecifikationskataloget er pakket (med 4 pr. celle), og da der samtidig refereres til kataloget fra flere aktioner (se afsn. 4.3.0., 4.4. og 4.6.), er al behandling centraliseret i sekvensen 'hent linie-spec'.

Hent linie-spec benytter en hjælpevariabel (linie-spec-ordet), der skal initialiseres, før 'hent linie-spec' må kaldes (se afsn. 4.2. og 4.14.).

Så længe det specificerede antal oplysninger ikke er læst, giver 'hent linie-spec' en værdi, der direkte kan bruges som index i grundelementkataloget. Når sidste oplysning er læst, giver sekvensen værdien 0, hvilket benyttes til test af, at det korrekte antal oplysninger er læst (se afsn. 4.3.0, 4.4. og 4.6.).

2.0 Overordnet programlogik

Den væsentlige del af programlogikken i fase 1 er lagt i tabeller. Ligesom definitionen af inputmaterialeets struktur ligger i linieta-bellerne, er alle bestemmelser om, hvad programmet skal gøre når et givet tegn er læst, lagt i en tabel, aktionstabellen.

Aktionstabellen kræver en eller anden form for angivelse af, hvad der er sket tidligere i programmet. Denne information, der kaldes tilstanden, er defineret i det følgende.

2.1. Tilstande

De forskellige tilstande er angivet nedenfor med forklaring af, hvad der har bevirket, at programmet befinder sig i den pågældende tilstand. Desuden er de væsentlige reaktioner i tilstanden beskrevet.

1. starttilstanden. Programmet er kun i denne tilstand, når læsningen af en ny strimmel skal påbegyndes. Programmet bliver i tilstand 1, indtil der læses et bilagsstart-tegn.
2. efter bilagstype. Programmet har læst bilagsstart-tegnet, samt bilagstypen. Hvis bilagstypen er afsluttet med en terminator, fortsættes i tilstand 3 - alle andre tegn bevirker at programmet kommer i en af fejltilstandene.
3. efter bilagsoplysning. Programmet er ved at læse oplysningerne i bilagslinien. Hvis oplysningen afsluttes med en terminator, beholdes tilstanden. Mødes der lineslut ændres tilstanden til 4 - i de øvrige tilfælde kommer programmet i en af fejltilstandene.
4. efter bilagslinie-slut. Hele bilagslinien er læst, og der læses til start af nyt bilag (hvilket ændrer tilstanden til 2) eller til start af en kropslinie (med ændring af tilstanden til 5).
5. efter linietype. Linietyper er indlæst. Hvis den er afsluttet med en terminator, sættes tilstanden til 6 - i alle andre tilfælde vælges en fejltilstand.
6. efter linieoplysning. Programmet er ved at læse oplysningerne i en kropslinie. Hvis oplysningen afsluttes med en terminator, bibeholdes tilstanden. Mødes lineslut, ændres tilstanden til 7 - i alle andre tilfælde vælges en af fejltilstandene.
7. efter lineslut. Der læses kommentarer indtil bilagsstart (med ændring af tilstanden til 2) eller liniestart (med ændring af tilstanden til 5) - andre tegn medfører valg af en fejltilstand.
8. efter linieannullering. Tilstanden ændres af bilagsstart (til 2), af liniestart (til 5), af strimmelslut (til 1) og af bilagsannullering (til 9) - alle andre tegn bibeholder tilstanden.
9. efter bilagsannullering. Tilstanden ændres af bilagsstart (til 2) og af strimmelslut (til 1) - alle andre tegn bibeholder tilstanden.
10. i bilagsfejl. Programmet går i denne tilstand, når en fejl, der medfører kassering af hele bilaget, er fundet. Tilstanden ændres ved bilagsstart (til 2) og ved strimmelslut (til 1) eller ved at det fejlbehæftede bilag annulleres (til 9).
11. i liniefejl 1. Programmet går i denne tilstand, når en fejl er opdaget i en købsnota eller en følgeseddel. Hvis fejlen ikke annulleres med linieannullering (tilstand sættes til 8) eller bilagsannullering (tilstand sættes til 9), kasseres hele bilaget når den aktuelle linie er helt læst (med overgang til tilstand 10). Læses der bilagsstart eller strimmelslut, sættes tilstanden til hhv. 2 og 1.

12. i liniefejl 2. Programmet går i den tilstand, når der er fundet fejl i en kropslinie (der ikke er fra en købsnota eller en følgeseddel). Hvis der læses liniestart, fortsættes med tilstand 2. Læses der bilagsstart, bilagsannullering eller strimmelslut, fortsættes der med tilstand 2, 9 eller 1. Alle øvrige tegn bibeholder tilstand 12.

2.2. Aktionstabellen

Aktionstabellen er en todimensional tabel med indgange for tilstand og tegnklasse. Indholdet af et enkelt felt i tabellen (aktionsordet) indeholder en række tal, der dels kan være numre på aktioner og dels være parametre til aktionerne. Aktionerne (der er nærmere beskrevet i afsn. 4) er enkeltprocesser, der foretager indlæsningen; hvilke aktioner, der skal udføres, og i hvilken rækkefølge, ses af aktionsordet. Parametrene er enten definition af en ny programtilstand eller nummeret på en fejltekst.

Eksempel, hvis der i tilstand 12 indlæses en bilagsstart (klasse 5) findes aktionsordet i tabellen som 12, 5, 2, 2 (ligesom i tabellen er numrene på aktioner understreget). Dette betyder, at først udføres aktion 12 (der ikke kræver nogen parameter) - udskrift af den kasserede linie. Derefter udføres aktion 5 (der heller ikke kræver nogen parameter) - afsluttende behandling af hele bilaget - og til sidst udføres aktion 2 (læsning af bilagstype), der bruger parameteren (2) til at definere den nye tilstand.

2.3. Aktionsindhopstabellen

Tabellen er endimensional med en indgang for hvert aktionsnummer. Tabelværdien er enten adressen på indhopsstedet i aktionen eller adressen på et stykke kode, der henter og hopper til den pågældende aktion.

2.4. Den generelle administration

Den generelle administration (generel adm.) er et stykke kode, der ud fra den aktuelle programtilstand og sidst læste tegns klasse slår op i aktionstabellen og henter aktionsordet.

Den generelle adm. slutter med at hoppe til første aktion i aktionsordet.

2.5. Hent aktionsparameter

Hent aktionsparameter (hent ak-param) administrerer udpakningen af de enkelte tal fra aktionsordet, og benyttes både til at hente nummeret på en aktion og til at hente en parameter til en aktion (de to talstørrelser er lagret på samme måde i aktionsordet).

Ved et særligt indhop i hent ak-param hentes dels nummeret på næste aktion, og dels hoppes der (via aktionsindhopstabellen) til den pågældende aktion.

3.0. Indlæsesequenser

Indlæsesequenserne bruges til al indlæsning undtagen operatørsvarerne i aktion 7 (se afsn. 4.7.). Den egentlige læsning sker med sekvensen 'læs tegn' (afsn. 3.1.), der læser ved hjælp af tegntabellen. De øvrige læsesekvenser benytter 'læs tegn' som undersekvens.

Tallæsning sker enten 'normalt' med sekvensen 'læs tal' eller efter Olivetti-konventioner med sekvensen 'læs 0-tal'. Hvilken af de to læseformer, der skal benyttes, bestemmes i aktion 7 (se afsn. 4.7.), afhængig af tegntabellen.

3.1. Læs tegn

Læs tegn kan indlæse et tegn på to forskellige måder. Enten læses der direkte fra strimmellæseren, eller også gentages behandlingen af sidst læste tegn. Hvilken af de to metoder, der skal benyttes, bestemmes fra aktionerne.

Når et tegn er læst, undersøges det først, om det er ALL HOLES (talværdi 639). Hvis dette er tilfældet, læses et nyt tegn.

Hvis tegnet er læst med paritetsfejl, adderes paritetsfejl-korrektionen til tegnet. Korrektionen kan enten være 0 (når paritetsfejl ikke er tilladt) eller 544 (når paritetsfejl skal accepteres), hvilket dels ophæver paritetsfejlangivelsen, og dels forøger værdien med 32.

Derefter sættes tegnlæsningen til direkte læsning fra strimmellæseren, så samme tegn ikke læses uendelig mange gange.

Nu undersøges værdien af tegnet. Hvis denne er større end 64, sættes tegnklassen direkte til 10 (illegale tegn), og der hoppes ud.

Hvis tegnet lå indenfor tegntabellens grænser, slås der op i denne, hvorved tegnklassen og den interne værdi findes. Hvis tegnklassen er 0 (blinde tegn), begynder der forfra på tegnlæsningen. I alle andre tilfælde hoppes der ud.

Hvis en strimmel ikke er forsynet med strimmel-slut-tegn, vil 'læs tegn' på et eller andet tidspunkt prøve at læse fra en 'tømt' strimmellæser. Operatøren kan i det tilfælde fra kontrolbordet fremkalde virkningen af strimmel-slut-tegnet.

3.2. Læs tal

Læs tal indleder med at nulstille den celle, hvor tallet skal lagres. Derefter læses et tegn (med 'læs tegn'), og tegnklassen undersøges. Hvis tegnklassen er 2 (cifre), multipliceres det hidtil læste tal med 10 og det nye ciffer adderes dertil. Hvis der kommer overløb ved multiplicationen, returneres der med terminatorklasse = 0, ellers gentages proceduren med læsning af et nyt tegn osv.

Læses der et tegn med klasse forskellig fra 2, undersøges det, om tegnet er blindt (klasse 1), hvorved læsningen gentages. Hvis dette ikke er tilfældet, returneres der med sidst læste tegns klasse. Hvorvidt afslutningstegnet var en korrekt terminator, må afgøres på det sted, hvor 'læs tal' blev kaldt, eller i aktionstabellen (det sidste er det normale).

3.3. Læs 0-tal

Læs 0-tal læser tal, hvor cifrene læses bagfra (altså med det mindst betydende ciffer først). Sekvensen indleder med at nulstille 'tal-cellen' og anbringe et 1-tal i en multiplikationsfaktor.

Der indlæses og testes et tegn på samme måde som i 'læs tal' (afsn. 3.2.). Læses der et ciffer, multipliceres det med multiplikationsfaktoren og adderes til tal-cellen. Derefter multipliceres multiplikationsfaktoren med 10 og næste tegn læses.

Overløb, talblinde tegn og afslutningstegn behandles som i 'læs tal'.

3.4. Læs tekst

Sekvensen kaldes med 2 parametre. Den første angiver, hvortil teksten skal læses. Den anden angiver, hvor mange celler teksten maksimalt må fylde.

Der læses ved hjælp af 'læs tegn' indtil et tegn med klasse større end 3. Når dette sker, hoppes der ud med terminator-klassen, som så skal testes af aktionstabellen (hvilket er det normale) eller af det stykke kode, der kaldte 'læs tekst'.

Tekster pakkes efter Help-3/Algol-4 konventioner med den undtagelse, at en celle ikke bliver fyldt op med sluttegn (værdi 10), hvis sidste celle ikke er helt opbrugt. Desuden ændres vognretur (64) ikke til 63.

Hver 6. gang et tegn er læst, anbringes tegnet 15 forrest i cellen for at angive (ved den senere trykning), at der skal skiftes til en ny celle. Når tekstens sluttegn læses, pakkes tegnet med værdi 10 ind i teksten på normal måde, og der hoppes ud. På denne måde kan man risikere, at en celle kun indeholder tekstslut-tegnet.

Hvis en tekst har den maksimale længde (6 x antallet af celler), bliver celle-slut-markeringen (15) rettet til tekst-slut (10).

3.5. Læs dato

Læs dato starter med at indlæse et heltal ved hjælp af enten 'læs tal' eller 'læs 0-tal' (se afsn. 3.2. og 3.3.). Tallet gemmes på heltalsform til en evt. akkumulering (se aktion 3, afsn. 4.3.2.), og splittes derefter op i dag, måned og år. Måned og dag gemmes på heltalsform til evt. udskrift på indlæsekontrol-listen.

For at datoen kan accepteres, skal bestanddelene ligge i følgende intervaller:

$$0 < \text{dag} < 32$$

$$0 < \text{måned} < 13$$

$$64 < \text{år} < 100$$

Hvis disse betingelser ikke er opfyldt, eller hvis der kom taloverløb ved tal-læsningen, sættes terminator-klassen til 0, og der hoppes ud.

Hvis datoen blev accepteret, bliver den derefter pakket på formen:

$$384 \times \text{år} + (\text{måned}-1) \times 32 + \text{dag}$$

og der returneres til kaldestedet (aktion 3).

Afsnit 4 Aktioner

Aktionerne er selvstændige programstykker, der hver for sig har en fast funktion. Aktionerne kaldes fra den centrale administration eller fra andre aktioner. Rækkefølgen af kaldene af aktionerne er bestemt i aktionstabellen (se afsn. 2.1).

Visse af aktionerne har en parameter, hvis værdi står i aktionstabellen. Denne parameter kan enten være nummeret på en ny programtilstand eller nummeret på en fejltekst (fejlteksttabellen er beskrevet i afsnit 6.1).

4.1. Aktion 1

Aktion 1 læser et tegn (med sekvensen 'læs tegn', afsn. 3.1) og gemmer tegnklassen. Den interne værdi af tegnet gemmes ikke. Dernæst kaldes 'hent ak-param' og aktionsparameteren gemmes som ny tilstand. Der returneres til 'den centrale adm.', hvor klassen og tilstanden benyttes til nyt opslag i aktionstabellen.

4.2. Aktion 2

Aktion 2's vigtigste funktion er at læse bilagstypen og initialisere indlæsningen af bilagslinien.

De enkelte funktioner er følgende:

Der ændres en pegepind, så al indlæsning sker til bilagslinie-området. Via sekvensen 'hent ak-param' hentes en ny værdi til 'tilstand'. Bilagstypen indlæses, terminatorklassen gemmes, og det undersøges, at værdien af bilagstypen ligger indenfor linietypekatalogets rammer. Der slås op i linietypekataloget, og linietype-kat-ordet hentes. Fra linietype-kat-ordet undersøges, at bilagstypen er legal (dette sker ved at undersøge, om den interne bilagstype er 0).

Der overføres en ny 'Intern bilagstype' til test af de efterfølgende linier. Endelig initialiseres brugen af linietypekataloget af hensyn til kropslinierne, og linespecifikationsordet initialiseres til indlæsning af bilagslinien. Der afsluttes med et hop til 'den generelle adm'.

Hvis der blev fundet fejl i bilagstypen, sættes tilstanden til 10 (fejl i bilag), fejltekst 11 vælges ('linietype') og der fortsættes i 'den generelle adm' uden initialisering som ovenfor.

4.3. Aktion 3

Aktion 3 er den egentlige indlæse-del af fase 1. I denne aktion indlæses de elementer de enkelte linier består af. Et element kan enten være et tal, en dato eller en tekst.

4.3.0. Fælles start

Hvilken element-type, der i et givet tilfælde skal læses, findes ved opslag i grundelementkataloget. Index til dette opslag er indholdet fra linespecifikationskataloget, hvorfra værdien fås ved et kald af 'hent linie-spec.' Hvis linespecifikationen = 0, er der struktur-fejl, og der hoppes ud med klasse = 12. Når grundelementet er fundet, spaltes aktion 3 ud i 3 dele:

4.3.1. Tallæsning

Tallæsningen starter med et kald af 'læs tal' (der enten læser et normalt tal eller et Olivetti-tal); derefter findes lagringsadressen fra grundelementkataloget, og tallet gemmes deri. Hvis det er specificeret i grundelementkataloget (GEkat), akkumuleres tallet til den aktuelle 'linietotal' (se nærmere under 'læs totallinie' 4.15).

Af hensyn til senere sorteringer, adderes der til konto/vare-numre et tal i den forreste del af cellen. Dette tal findes i GEkat, og adderes til det lagrede tal. Denne funktion udføres altid, så det ikke er nødvendigt at undersøge om et tal er et konto/vare-nummer, hvilket medfører, at modifikationstallet normalt er 0.

Hvis der i GEkat er specificeret intervalbegrænsning på tallet, udføres intervaltesten herefter.

Endelig udføres evt. en check-ciffer kontrol, hvilket kan være test af 7-rest, 10-rest eller 11-rest. Metoderne er beskrevet i afsn. 4.3.5.

Hvis intervalresten eller checkcifferkontrollen finder fejl i tallet, sættes terminatorklassen = 0.

4.3.2. Datolæsning

Datoen indlæses med sekvensen 'læs dato', hvorefter datoen evt. (afhængig af specifikationerne i GEkat) akkumuleres i linie-totalen på normal heltalsform.

4.3.3. Tekstlæsning

Fra GEkat hentes oplysninger om hvortil teksten skal læses, samt hvor mange ord teksten maksimalt må fylde. Derefter indsættes disse to parametre til kaldet af 'læs tekst', hvorefter teksten læses. Tekster kan ikke akkumuleres.

4.3.4. Fælles afslutning

Når et af de tre elementtyper er indlæst, afsluttes aktion 3 ved at hente en parameter fra aktionstabellen, der definerer den næste programtilstand. Desuden gemmes terminatorklassen (til næste opslag i aktionstabellen). Hvis klassen = 0 eller 12 skiftes der dog til repetering af sidst læste tegn i 'læs tegn'.

Der afsluttes med et hop til den generelle administration.

4.3.5. Checkcifferkontrol

Der kan i forbindelse med aktion 3 benyttes 1 af 3 forskellige slags checkcifferkontrol (restkontrol).

4.3.5.1. 7-rest kontrol

Tallet kontrolleres ved at fjerne sidste ciffer, og dividere de forreste cifre med 7. Hvis resten fra denne division = 0, sættes resten til 7. Tallet er korrekt, hvis resten = det sidste ciffer.

4.3.5.2. 10-rest kontrol

Når 10-resten skal dannes, tages de enkelte cifre bagfra, og multipliceres med skiftevis 2 og 1 (startende med 2). Hvis nogle af de herved fremkomne produkter er større end 9, erstattes de med deres tværsum (= tallet - 9). Summen af de korrigerede produkter trækkes fra det nærmeste, større tal, der ender på 0. Denne differens er checkcifret.

Når et tal med checkciffer skal undersøges, tages cifrene (incl. checkcifret) bagfra og multipliceres med skiftevis 1 og 2 (startende med 1). Igen erstattes to-cifrede tal med tværsummen, og igen summeres de korrigerede produkter. Summen divideres med 10, og skal give resten 0, for at det oprindelige tal er korrekt.

4.3.5.3. 11-rest kontrol

Når 11-resten skal dannes, tages de enkelte cifre bagfra, og multipliceres med henholdsvis 2, 3, 4 osv. De fremkomne produkter adderes, og summen divideres med 11. Resten trukket fra 11, er checkcifret. Da checkcifret kan blive større end 9, kan ikke alle tal forsynes med denne form for checkciffer.

Når et tal med 11-rest-checkciffer skal undersøges, tages cifrene bagfra (incl. checkcifret) og multipliceres med henholdsvis 1, 2, 3 osv. Produkterne adderes og divideres med 11. Tallet er korrekt, hvis denne division giver resten 0.

4.4. Aktion 4

Aktion 4 afslutter læsningen af bilagslinien.

Først undersøges det, at det korrekte antal oplysninger er indlæst. Dette gøres ved at hente næste linie-specifikation (med sekvensen 'hent linie-spec'), som skal være 0.

Hvis dette ikke er tilfældet, sættes terminatorklassen til 12, og der skiftes til repetering af sidst læste tegn, hvorefter der hoppes til den generelle administration.

Hvis linien indeholdt det korrekte antal oplysninger, fortsættes der med at overføre bilagsdatoen (= sidst læste dato) til en særlig celle, hvorfra den tages til indlæsekontrollisten. Derefter tælles indlæsenummeret op (med 10) og det nydannede indlæsenummer samt linie-type-katalog-ordet overføres til individområdet (linie-type-kat-ordet benyttes når bilagslinien senere redigeres om).

Endelig initialiseres læsningen af kropslinier, ved at danne basisparameterord til stakning af kropslinier samt indsætte basisadressen til kropslinie-området.

Til sidst nulstilles de variable, der benyttes under læsningen af en linie, og der returneres til den generelle administration.

4.5. Aktion 5

Aktion 5 færdigbehandler et helt indlæst bilag, samt udskriver det på indlæsekontrollisten.

Først undersøges, hvor mange kropslinier, bilaget består af. Hvis antallet er 0, kan kun sletninger accepteres; er der tale om andre bilagstyper, bliver de forkastet med udskriften 'antal linier'.

Hvis bilaget er en købsnota, undersøges om beløbet ('beløb 1') i bilagslinien er lig med summen af beløbene ('beløb 1'erne) i kropslinierne; hvis dette ikke er tilfældet kasseres bilaget med udskriften 'afstemning'.

Efter de indledende tests hentes et stykke specialkode, der derefter udføres. Specialkoden (se afsn. 5 ff) komprimerer de indlæste linier og skriver dem derefter på bånd (station 2).

Derefter udskrives bilaget på indlæsekontrollisten (se afsn. 6.1.); dvs. der udskrives bilagstype, bilagsnr., konto/vare-nummer, dato, evt. beløb (kun købsnoter og bogholderiposter) og en evt. beregnet og indlæst total.

Til sidst nulstilles individ-områderne og der fortsættes i næste aktion (hvilket sker med et specielt hop til 'hent aktionsparameter').

Hvis der har været fejl i bilaget (under en af de to tests, der er beskrevet ovenfor), afsluttes der med et hop til aktion 10 (der udskriver det kasserede bilag).

4.6. Aktion 6

Aktion 6 afslutter behandlingen af en kropslinie. Først overføres linietype-kat-ordet til individområdet, hvorefter det undersøges, om det aktuelle bilag kræver udførelse af specialkode for hver linie. Er dette tilfældet, hentes specialkoden ind i lageret og udføres.

Herefter undersøges det, om det korrekte antal oplysninger er indlæst, hvilket gøres ved at undersøge, om næste linespecifikation = 0. Er der fejl, sættes klassen = 12, fejlteksten 'antal oplysninger' sættes, og der fortsættes i den generelle administration.

Hvis antallet af oplysninger var korrekt, fortsættes der med at tælle indlæsenumeret op (med 10) og gemme det i individområdet.

Dernæst undersøges det, om der er plads til individområdet i individstakken. Er der ikke plads, sættes tilstanden til 10, fejlteksten 'antal linier' sættes, og der fortsættes i den generelle administration.

Hvis der er plads i individstakken, overføres individområdet til stakken, og linietotalen adderes til bilagstotalen, hvorefter individområde samt visse variable nulstilles, og der fortsættes i næste aktion.

4.7. Aktion 7

Aktion 7 behandler strimmelsslut. Der indledes med en nulstilling af variable samt individområder. Derefter udskrives 'strimmelsslut' på printeren (ved indhop i fase 1 er printeren ikke sluttet til, hvorved udskriften undgås). Desuden sættes linietælleren til 100 for at fremkalde formularskift med overskrift ved først følgende vognretur.

Derefter sættes tilstanden til 1 (starttilstanden) og operatør-udskriften 's, 1, 2' skrives på skrivemaskinen.

Svarer operatøren med 's', laves der formularskift på printeren, og sekvensen 'afslut fase 1' hentes og udføres.

Svares der derimod med '1' eller '2', initialiseres læsesekvenserne, der skrives en vognretur (hvilket fremkalder før nævnte formularskift), og der fortsættes i aktion 1.

Initialiseringen af læsesekvenserne sker ved, at behandlingen af tegntabellen, samt paritetsfejl initialiseres i 'læs tegn'. Desuden bestemmes det, om tallæsning skal ske 'normalt' eller efter Olivettikonventioner.

Hvis operatøren ikke har svaret korrekt (dvs. 's' eller '1' eller '2'), udskrives operatørteksten en gang til (med rødt), og et nyt svar kan gives.

4.8. Aktion 8

Aktion 8 henter en aktionsparameter og indsætter værdien som nummeret på fejlteksten (som ikke skrives på dette tidspunkt).

Der fortsættes i næste aktion (med specielt hop til 'hent aktionsparameter').

4.9. Aktion 9

Aktion 9 skriver en vognretur (med sekvensen 'writecr') og henter derefter næste aktionsparameter, der opfattes som nummeret på en fejltekst. Teksten udskrives, og der fortsættes i næste aktion (med specielt hop til 'hent aktionsparameter').

4.10. Aktion 10

Aktion 10 udskriver oplysninger fra et kasseret bilag; dvs. bilagstype, bilagsnr., konto/vare-nummer, dato og liniernr. for den linie, der sidst er indlæst. Denne linie vil normalt være den, hvor fejlen har optrådt, men i særlige tilfælde (f.eks. ved annulleringer og i de tilfælde, hvor fejlen først er opdaget, når hele bilaget er indlæst) kan linienummeret være forkert.

Efter linienummeret udskrives fejlteksten (som er sat af aktion 8).

Til sidst nulstilles individområder og variable, og der fortsættes i næste aktion.

4.11. Aktion 11

Aktion 11 træder kun i kraft ved programfejl (f.eks. optræder den på de steder i aktionstabellen, hvor der er logisk umuligt at komme).

På skrivemaskinen og printeren udskrives teksten 'programfejl', og hele lageret (samt registre) udskrives på printeren.

Der afsluttes med udhop til Help-3.

4.12. Aktion 12

Aktion 12 udskriver en kasseret linie, dvs. samme oplysninger som et kasseret bilag. Der afsluttes med at nulstille kropslinie-området samt variable, hvorefter der fortsættes i næste aktion.

4.13. Aktion 13

Normalt bevirker en fejl i en kropslinie, at kun den fejlbehæftede linie kasseres, men for købsnotaer (bilagstype 4) og følgesedler (bilagstype 5) gælder det, at en fejl i en kropslinie bevirker, at hele bilaget kasseres (med mindre linien bliver annulleret).

Aktion 13 vælger derfor ud fra bilagstypen en af de to tilstande 'i liniefejl': Tilstand 11 eller 12 (se nærmere i afsn. 2.1).

Når den nye tilstand er valgt, læses der et tegn, klassen gemmes til senere opslag i aktionstabellen, og der fortsættes i den generelle administration.

4.14. Aktion 14

Aktion 14 indlæser kropslinie-typen samt initialiserer indlæsningen af kropslinien.

Først tælles linienummeret (inden for det aktuelle bilag) op med 1, og en ny programtilstand defineres som næste aktionsparameter.

Derefter indlæses linietyper. Kommer der taloverløb ved læsningen, sættes terminator-klassen til 0, og der fortsættes i den generelle administration.

Hvis linietyper er 1023, aktiviseres læsningen af totallinien (se afsn. 4.15.2), ellers gemmes linietyper i individområdet, og værdien af linietyper undersøges ved opslag i linietypekataloget. Ved fejl sættes fejltæst 11 ('linietyper') der markeres, at næste tegnlæsning skal repetere sidste tegn, hvorefter der fortsættes i aktion 13.

Hvis linietyper blev accepteret, initialiseres sekvensen 'hent linie-spec', og der fortsættes i den generelle administration.

4.15.0. Totaler

Totalerne indgår i indlæsningen for at fange hullefejl etc. Indlæseprogrammet læser hhv. genererer totaler og skriver dem ud, men uden at give fejlreaktion hvis totalerne ikke stemmer.

Totalerne deles i to grupper:

4.15.1. Beregnede totaler

I grundelementkataloget kan det specificeres, at det pågældende grundelement (som skal være enten et tal eller en dato) skal adderes til en totalcelle under indlæsningen.

På grund af annulleringer er det imidlertid ikke nok med en enkelt celle, og der er derfor indført en 'linie-total' og en 'bilagstotal'. De grundelementer, der skal indgå i totalen akkumuleres i 'linie-total'en', og først når hele linien er indlæst og accepteret adderes 'linie-total'en' til 'bilagstotal'en' (aktion 6); herved undgås det, at annullerede og fejlbehæftede linier indgår i totalen.

4.15.2. Totallinier

En totallinie består altid kun af linietyper (1023) og den egentlige total. Totallinien indlæses af en undersekvens til aktion 14 (læs linietyper), og bevirker eventuelle tidligere totallinier glemmes, og at den nye total gemmes til udskrift på indlæsekontrollisten (aktion 5).

Totallinien indlæses udenom aktionstabellen (fordi totallinier ikke er behandlet i linieta-bellerne). Dette sker ved først at sætte tilstand til 8, hvilket er nødvendigt for senere (dvs. efter indlæsningen af totallinien) at kunne komme ind i aktionstabellen igen.

Derefter checkes linietype-terminatoren. Hvis terminator-klassen er forskellig fra 4, sættes fejltekst 4 ('terminator'), tilstanden ændres til 12, og der fortsættes i den generelle administration.

Hvis terminatoren var korrekt indlæses total-tallet, og dets terminator-klasse checkes som beskrevet ovenfor. Hvis terminatoren var korrekt (på dette sted klasse 7) gemmes total-tallet, der læses et tegn, hvorefter der fortsættes i den generelle administration.

5.0. Specialkode

Specialkodens funktion er at pakke de indlæste oplysninger sammen til et format, der passer bedst til de øvrige faser. Da denne pakning er individuel for de forskellige bilagstyper, er koden ikke generaliseret.

Specialkoden består af to dele. Den første del er en tabel over den egentlige kode (= anden del). Index i tabellen er bilagstypen, og tabelværdien giver direkte kodens placering i bufferen (et parameterord).

Specialkoden kan udføres fra to forskellige steder. Fra 'end bilag'-aktionen (nr. 5, afsn. 4.5.) hentes der altid et stykke specialkode, fordi denne bl.a. sørger for at skrive de dannede individer på båndet. For købsnotaer og kartoteksændringer er det desuden nødvendigt at udføre specialkode fra 'end linie'-aktionen (nr. 6, afsn. 4.6.) af test-mæssige grunde.

Da specialkoden kun indeholder få processer, der har indflydelse på logikken i fase 1, vil de enkelte dele kun blive meget kort omtalt i det følgende.

5.1. Pak bogholderiposteringer (type 1-3)

Ud over simple sammenpakninger og båndtransporter skal denne special-sekvens doublere visse oplysninger, for di bogholderiposteringerne benyttes til ajourføring to forskellige steder (dobbelt bogholderi). Dels akkumuleres samtlige beløb fra kropslinierne og gemmes i bilagslinien (og til udskrift på indlæsekontrollisten). Desuden tages debet/kredit-koden ('fortegnsangivelsen') fra bilagslinien, vendes om og anbringes i samtlige kropslinier.

5.2. Pak købsnota (type 4)

Sekvensen indeholder kun helt banale sammenpakninger.

5.3. Pak følgeseddel (type 5)

Udover simple sammenpakninger, skal specialkoden indsætte rabatter fra rabatlinier i varelinier. Dette sker ved at gennemløbe kropslinierne bagfra (modsat indlæserækkefølgen), og indsætte en rabatstørrelse i varelinierne. Denne rabatstørrelse er 0 fra starten men forhøjes, hvor gang der mødes en rabatlinie. Rabatsatserne fra rabatlinierne (der kan være 1 eller 2 satser i hver linie) sammensættes på en sådan måde, at mængden af rabatter erstattes med en enkelt rabatsats.

Når den samlede rabat skal reguleres med et nyt element fra kæderabatten, sker det efter følgende algoritme:

$$\text{rabat} := 1000 - (1000 - \text{rabat}) \times (1000 - \text{rabatpromille}) / 1000;$$

5.4. Pak ændringer og sletninger (type 6-13)

Først sammenpakkes oplysningerne fra bilagslinien, hvorefter kropslinierne (ændringslinierne) behandles. Hvis der er tale om sletninger, returneres der på dette tidspunkt.

En ændring består af to dele. Den første del indeholder oplysninger fra linie-typekataloget (se afsn. 1.1.3.) og definerer hvordan ændringen skal foretages, samt hvor lang den er (dvs. hvor mange celler, den fylder).

Den anden del af ændringen er selve den størrelse, der skal adderes, subtraheres eller placeres i kartoteket. Denne del fylder 1 celle for numeriske størrelser og 5 celler for tekster.

Ændringer pakkes sammen i individet - umiddelbart efter oplysningerne fra bilagslinien - så mange der nu er plads til (dvs. maksimalt 7 pr. individ). Når et individ er fyldt op, skrives det på båndet, bilags-linie-oplysningerne kopieres over til næste individ, der derefter fyldes op med ændringer.

Denne sammenpakning af linier indenfor et enkelt individ kræver speciel manipulation med indlæsenummeret, så dette kun stiger med 10 for hvert individ på båndet.

5.5. Akkumulering af beløb i købsnotaer (type 4)

Dette stykke specialkode danner summen af beløbene i kropslinierne indenfor en købsnota, dog med den undtagelse at fradrags- og rabat-linie-beløb indgår med modsat fortegn.

Totalbeløbet benyttes senere (aktion 5, afsn. 4.5.) til sammenligning med beløbet i bilagslinien.

5.6. Test ændringstyper (type 6-9)

Specialkoden sammenligner ændringstypen (placer, adder eller subtraher, talværdier 1-3) i ændringslinien med den tilsvarende beskrivelse af tilladte ændringer i linietypekataloget.

Testen sker ved at ændre en eventuel subtraktion til en addition (3 til 2) idet de to ændringer altid vil være tilladt samtidig. Derefter multipliceres det korrigerede ændringsønske logisk med ændringsbeskrivelsen i linietypekartoteket. Denne beskrivelse er nemlig 1 for placering tilladt, 2 for addition (eller subtraktion) tilladt og 3 for begge dele tilladt. Hvis resultatet af den logiske multiplikation er 0, er der ønsket en ulovlig ændring, og fejltekst nr. 14 ('ændring') sættes, og der fortsættes i aktion 12 (udskrivning af fejllinie, se afsn. 4.12.) i stedet for i aktion 6, der kaldte specialkoden. Efter at aktion 12 er udført, fortsættes der med den aktion, der er nævnt efter aktion 6 i aktionstabellen.

6.0 Fase 1 output

Fase 1 leverer to former for output: Dels afleveres alle accepterede linier som individer på magnetbånd (register 2.1.) til behandling i de følgende faser; og dels udskrives en liste (indlæsekontrollisten) over alle accepterede bilag, samt de bilag eller linier, der er kasseret på grund af syntaxfejl eller lignende.

6.1. Indlæsekontrollisten

Listens overskrift indeholder firmanavn, dato for kørslen, sidenummer og søjleoverskrift for anmærkninger etc.

Venstre del af listen indeholder de bilag, der er blevet accepteret (helt eller delvis). Fra hvert bilag udskrives følgende oplysninger:

- bilagstype (eller bilagsart), nemlig det tal, der står umiddelbart efter bilagsstart-koden.
- bilagsnummer, hvilket er en af oplysningerne fra bilagslinien.
- konto/vare-nummer, hvilket er en anden oplysning fra bilagslinien.
- dato, som tages fra bilagslinien (kørselsdatoen står jo i overskriften). Der udskrives kun dag og år.
- beløb (kommer kun ved bogholderiposteringer og købsnotaer, type 1-4). Beløbet er blevet beregnet under indlæsningen som summen af de beløb, der er angivet i kropslinierne.
- RC-total, summen af de enkeltoplysninger i bilaget, som er specificeret til at indgå i kontrolsummen.
- bruger-total, indholdet af sidst læste totallinie (se afsn. 4.15.).

Højre side af indlæsekontrollisten indeholder oplysninger om de bilag eller enkeltlinier, der er blevet kasseret på grund af syntax-fejl eller lignende. Udskriften består af følgende oplysninger:

- bilagstype
 - bilagsnummer
 - konto/vare-nummer
 - dato
- Oplysningerne er de samme som beskrevet ovenfor. Datoen bliver dog kun skrevet, hvis hele bilagslinien er læst uden fejl.

For lettere at kunne finde fejlen, identificeres denne desuden med følgende oplysninger:

- linienummer. Hvis syntax-fejlen har ligget i bilagslinien, udskrives linienummeret ikke. Normalt vil linienummeret angive nummeret på den linie, der indeholder fejlen, men i særlige tilfælde (f.eks. når et bilag annulleres efter der er forekommet en syntax-fejl) kan linienummeret være forkert.
- fejltekst. Teksten beskriver meget kort årsagen til fejlen. De enkelte tekster er beskrevet i afsn. 6.2.

Enkelte syntax-fejl kan ikke henføres til et bestemt bilag (det drejer sig om fejl i kommentarfeltet før første bilag på en strimmel); i disse tilfælde skrives fejlteksten yderst til venstre på listen uden yderligere identifikation af fejlen.

Hver gang, der mødes strimmeslut, udskrives teksten 'strimmeslut' helt til venstre på listen, og der skiftes til ny side.

6.2. Fejltekster

Fejlteksterne findes i en særlig tabel (fejltexttabellen), hvortil der blandt andet refereres fra aktionstabellen.

Nr.	Tekst	Forklaring
1	bilagsannull.	Teksten optræder når et bilag annulleres med bilagsannulleringstegnet. Linienummeret er kun korrekt, hvis der ikke er mødt syntaxfejl før annulleringstegnet.
2	kommentar	Udskriften kommer, hvis der forekommer ulovlige tegn i kommentarfeltet, hvor kun tegn, der er tilladte i tekster, må forekommer.
3	ill. tegn	Udskriften betyder, at et tegn af klasse 10 er læst uden at blive annulleret.
4	terminator	En oplysning afsluttes med et forkert skilletegn.
5	illegal ann.	Teksten betyder, at man prøver at annullere en bilagslinie med lineannullering, eller at et annulleringstegn står i kommentarfeltet før første bilag på en strimmel.
6	tal-fejl	Udskriften betyder, at det er forsøgt at indlæse et tal på mere end 11 cifre, eller at et intervalbegrænset tal har ligget udenfor grænserne, eller at et tal har haft et forkert check-ciffer.
7	tekst-fejl	En tekst har været for lang. Udskriften forekommer dog også, når et tal (eller dato) er afsluttet med et tekst-tegn i stedet for med en terminator eller lineslut.
8	str. slut	Strimmel-slut-tegnet er læst midt i bilaget.
9	bilags-start	Et bilags-start-tegn er læst midt i et bilag.
10	antal opl.	En linie indeholder et forkert antal oplysninger.
11	linie-type	En linietype har forkert værdi. Linietypen har enten været for stor til at kunne bruges til tabelopslaget i linietypetabellen, eller har ikke været tilladt indenfor det aktuelle bilag.
12	antal linier	Et bilag indeholder for mange linier. Grænsen ligger ved 107.
13	afstemning	Total-beløbet i en købsnotas bilagslinie er ikke summen af de tilsvarende beløb i kropslinierne.
14	ændring	Ulovlig kartoteksændring. Forsøg på at ændre ved addition (eller subtraktion) hvor der kun må ændres med placering - eller det stik modsatte forhold.

6.3. Writecr

Writecr er en sekvens, der udfører alle linieskift og formularskift på indlæsekontrollisten. Sekvensen indleder med at skifte linie, og tælle antallet af udskrevne linier op med 1 (af hensyn til prisberegning for kørslen).

Dernæst tælles antallet af linier pr. side op med 1, og det undersøges, om siden er fyldt. Hvis dette ikke er tilfældet, returneres til kaldstedet.

Hvis siden er fyldt op, skal der trykkes formularskift og overskrift på den nye side. Før dette gøres, stoppes der dog, hvis KB-lampen på manøverbordet er tændt, af hensyn til papirskift.

Writecr kaldes altid umiddelbart før, der skrives noget. Dette betyder, at udskriften ikke kan komme til at slutte med en side, der kun indeholder overskrift. Til gengæld bliver trykningen på en enkelt-bufret printer lidt langsommere end normalt.

6.4. MB-register 2.1.

Registret indeholder alle accepterede linier fra den aktuelle og evt. også foregående indlæsninger. Normalt vil et individ indeholde information fra en enkelt linie, men for kartoteksændrings vedkommende, kan et enkelt individ indeholde oplysninger fra op til 7 ændrings-(krops-)linier.

Individerne har rent logisk forskellig længde, men bliver af hensyn til magnetbåndsystemet skrevet med længden 20. I det følgende er visse fælles dele af individerne beskrevet.

Til at identificere individtypen (f.eks. varelinie eller kreditorpostering) findes hoved- og undertype (taget fra linietypekataloget). Desuden indeholder alle individer, der er dannet på grundlag af kropslinier, også linietyper. Ved definitionen af individtypen er det (eller de) kartotek(er), individet skal operere på, defineret.

Til definition af hvor individet skal operere indenfor et givet kartotek, indeholder alle individer et konto- eller vare-nummer. At dette nummer er modificeret (se afsn. 4.3.1.) skyldes udelukkende, at man vil gøre sorteringskriterierne så simple som muligt.

For at identificere individerne udadtil (ved senere udskrifter) indeholder hvert individ et bilagsnummer og en dato. Begge oplysninger skal forekomme i alle bilagstyper.

Endelig gemmes der også oplysning om indlæserækkefølgen, idet hvert individ får et nummer, indlæsenummeret, der tælles op med 10 for hvert individ, der dannes.

7. Diverse sekvenser

De følgende sekvenser falder ikke naturligt ind under anvendte gruppering, og er derfor beskrevet samlet her.

7.1. Nulstil

Sekvensen benyttes til nulstilling af flere sammenhængende variable. I programmet er de variable netop samlet i grupper, så det bliver så let og overskueligt som muligt at initialisere (=nulstille) dem.

Sekvensen kaldes med 2 parametre, nemlig første og sidste celle, der skal nulstilles.

7.2. Start fase 1

Indhop i fase 1 (og de efterfølgende faser) sker ved, at det omkringliggende system henter den første tromlekanal i koden ned i ferritlageret, og begynder at udføre den.

Start fase 1 indleder med at hente linieta-beller, fejlteksttabellen samt specialkoden fra tromlen og lagre dem i bufferen, hvorfra de benyttes under kørslen.

Dernæst skal båndet åbnes (station nr. 2, fil nr. 1) og indlæsesnummeret initialiseres. Båndåbningen (trimningen) kan ske på to forskellige måder: Normal åbning eller åbning for fortsat skrivning. Hvilken af de to åbninger, der skal anvendes, bestemmes af det omkringliggende system. Hvis der åbnes normalt, skal indlæsesnummeret nulstilles, hvorimod indlæsesnummerets initialiseringsværdi står på båndet, hvis der åbnes for fortsat skrivning.

Hvis der er specificeret under indlæsningen af fase 1, overføres tegntabellen til ferritlageret, hvor den er lidt hurtigere tilgængelig end i bufferen.

Til sidst hentes den resterende del af fase 1 til ferritlageret (fra tromlen), og der fortsættes i aktion 7 (se afsn. 4.7.).

7.3. Afslut fase 1

Afslut fase 1 kaldes fra aktion 7 (se afsn. 4.7.), når operatøren specificerer at indlæsningen er slut.

Sekvensen lukker båndet og anbringer indlæsesnummeret i næste fil-etikette (hvilket sker ved at åbne næste fil for skrivning).

Der afsluttes med et hop til det omkringliggende system, der så sørger for at aktivere næste fase.

8.0. Test faciliteter

Af hensyn til indkøring og afprøvning af fase 1, er der mange steder i koden indsat test-udskrifter (debugudskrifter). For at udnytte lageret så godt som muligt, er udskrifterne fjernet i rutine-programmet, hvilket gøres ved at betinge indlæsningen af udskrifterne.

Da udskrifterne og de øvrige test-faciliteter ligger på et meget detaljeret niveau, er de kun overfladisk beskrevet her.

8.1. Betingelser

Under indlæsningen af fase 1, kan man bestemme hvor mange testudskrifter (debugudskrifter) man vil have. Mulighederne er:

Udskrift af hvert enkelt tegn, der læses

Udskrift af hver oplysning

Udskrift fra hver aktion samt hver tilstand/klasse

Udskrifter fra underskevenser

Desuden kan man under indlæsningen reservere ekstra ferritlagerplads (på bekostning af kørselshastighed), og endelig kan man definere, at indhop og udhop til/fra fase 1 skal ske direkte i stedet for gennem det omkringliggende system (running system). Herved opnår man, at fase 1 kan køres uafhængigt af de øvrige faser, men til gengæld må man selv sørge for, at linie- og tegn-tabeller bliver indlæst.

8.2. Direkte indhop og udhop

Ved direkte indhop skal programmet sørge for, at running system bliver hentet ind i ferritlageret. Desuden skal båndet åbnes (båndstart, tapetest), og det skal angives, om man vil køre ny eller fortsat indlæsning (se afsn. 7.2.).

Disse funktioner varetages af et stykke kode, der kun indlæses, når man har valgt at køre med direkte ind- og ud-hop.

Ved direkte udhop rewindes båndet, og der hoppes til Help-3.

8.3. Debugudskrifter

På det mest detaljerede niveau vil man for hvert tegn, der læses, få udskrevet den externe værdi, den interne værdi samt tegnklassen.

På det noget højere niveau får man udskrevet hvert tal, tekst og dato, der indlæses. Datoudskriften indeholder (decimal) udskrift af datoen både på upakket (decimal) og på pakket form.

Fra hver aktion har man mulighed for at få udskrevet teksten 'ak' efterfulgt af aktionens nummer. Desuden vil den generelle administration udskrive den værdi af tilstand og klasse, der er benyttet til opslag i aktionstabellen. Sidstnævnte udskrift er nok den vigtigste af alle debugudskrifterne.

Endelig kan man få udskrifter fra diverse undersekvenser. Det drejer sig om udskrift af hver aktionsparameter (fra 'hent ak-param', afsn. 2.5.), af hver líniespecifikation (fra 'hent linie-spec', afsn. 1.6.) og udskrift af grænserne af det interval, der bliver mulstillet (se afsn. 7.1.). Endelig vil sekvensen writecr (se afsn. 6.2.) udskrive en række '-'-tegn.

8.4. Debug-tabeller

Af hensyn til indkørslen og afprøvningen af fase 1, er der lavet et specielt sæt linietafeller og en særlig tegntabel.

Linietafellerne indeholder samtlige linietyper, og for hver linie det maksimale antal oplysninger. På passende steder i grundelementtabellen er der desuden indsat oplysning om total-dannelse og check-ciffer-kontrol (se afsn. 1.3.).

Begge tegntabeller benytter flexowriterkode, hvor en delspecialtegn er erstattet af 'sjældne' bogstaver (x, y, z, w, q). Tegntabel 1 læser almindelige tal og tillader ikke paritetsfejl. Denne tabel benyttes til langt den største del af testmaterialet. Tegntabel 2 tillader paritetsfejl (se afsn. 3.1.) og læser tal efter Olivettikonventioner.

RC - FAKT fase 1

AKTIONS-TABEL

20.1.1969.
BØT/IH.

KLASSE TIL- STAND	0 talfejl	1 tal-blinde	2 cifre	3 tekst-tegn	4 terminator	5 bilagstart	6 liniestart	7 linieslut	8 str. slut	9 linie - ann.	10 illegal	11 bilags-ann.	12 opl. fejl
1 start	<u>1</u> , 1	<u>1</u> , 1	<u>1</u> , 1	<u>1</u> , 1	<u>9</u> , 2, <u>1</u> , 1	<u>2</u> , 2	<u>9</u> , 2, <u>1</u> , 1	<u>9</u> , 2, <u>1</u> , 1	<u>7</u>	<u>9</u> , 5, <u>1</u> , 1	<u>9</u> , 3, <u>1</u> , 1	<u>9</u> , 5, <u>1</u> , 1	<u>11</u>
2 e.b. type	<u>8</u> , 6, <u>1</u> , 10	<u>11</u>	<u>11</u>	<u>8</u> , 7, <u>1</u> , 10	<u>3</u> , 3	<u>8</u> , 9, <u>10</u> , 2, 2	<u>8</u> , 4, <u>1</u> , 10	<u>8</u> , 4, <u>1</u> , 10	<u>8</u> , 8, <u>10</u> , 7	<u>8</u> , 5, <u>1</u> , 10	<u>8</u> , 3, <u>1</u> , 10	<u>8</u> , 1, <u>1</u> , 9	<u>11</u>
3 e.bl. opl.	<u>8</u> , 6, <u>1</u> , 10	<u>8</u> , 7, <u>1</u> , 10	<u>8</u> , 7, <u>1</u> , 10	<u>8</u> , 7, <u>1</u> , 10	<u>3</u> , 3	<u>8</u> , 9, <u>10</u> , 2, 2	<u>8</u> , 4, <u>1</u> , 10	<u>1</u> , 4	<u>8</u> , 8, <u>10</u> , 7	<u>8</u> , 5, <u>1</u> , 10	<u>8</u> , 3, <u>1</u> , 10	<u>8</u> , 1, <u>1</u> , 9	<u>8</u> , 10, <u>1</u> , 10
4 e bl. slut	<u>11</u>	<u>1</u> , 4	<u>1</u> , 4	<u>1</u> , 4	<u>8</u> , 2, <u>1</u> , 10	<u>4</u> , 5, 2, 2	<u>4</u> , 14, 5	<u>8</u> , 2, <u>1</u> , 10	<u>4</u> , 5, 7	<u>8</u> , 5, <u>1</u> , 10	<u>8</u> , 3, <u>1</u> , 10	<u>8</u> , 1, <u>1</u> , 9	<u>8</u> , 10, <u>1</u> , 10
5 e.l. type	<u>8</u> , 6, <u>13</u>	<u>11</u>	<u>11</u>	<u>8</u> , 4, <u>13</u>	<u>3</u> , 6	<u>8</u> , 9, <u>10</u> , 2, 2	<u>8</u> , 4, <u>13</u>	<u>8</u> , 4, <u>13</u>	<u>8</u> , 8, <u>10</u> , 7	<u>1</u> , 8	<u>8</u> , 3, <u>13</u>	<u>8</u> , 1, <u>1</u> , 9	<u>11</u>
6 e.l. opl.	<u>8</u> , 6, <u>13</u>	<u>8</u> , 7, <u>13</u>	<u>8</u> , 7, <u>13</u>	<u>8</u> , 7, <u>13</u>	<u>3</u> , 6	<u>8</u> , 9, <u>10</u> , 2, 2	<u>8</u> , 4, <u>13</u>	<u>1</u> , 7	<u>8</u> , 8, <u>10</u> , 7	<u>1</u> , 8	<u>8</u> , 3, <u>13</u>	<u>8</u> , 1, <u>1</u> , 9	<u>8</u> , 10, <u>13</u>
7 e.l. slut	<u>11</u>	<u>1</u> , 7	<u>1</u> , 7	<u>1</u> , 7	<u>8</u> , 2, <u>13</u>	<u>6</u> , 5, 2, 2	<u>6</u> , 14, 5	<u>8</u> , 2, <u>13</u>	<u>6</u> , 5, 7	<u>1</u> , 8	<u>8</u> , 3, <u>13</u>	<u>8</u> , 1, <u>1</u> , 9	<u>8</u> , 10, <u>13</u>
8 e.l. ann.	<u>11</u>	<u>1</u> , 8	<u>1</u> , 8	<u>1</u> , 8	<u>1</u> , 8	<u>5</u> , 2, 2	<u>14</u> , 5	<u>1</u> , 8	<u>5</u> , 7	<u>1</u> , 8	<u>1</u> , 8	<u>1</u> , 9	<u>11</u>
9 e.b. ann.	<u>11</u>	<u>1</u> , 9	<u>1</u> , 9	<u>1</u> , 9	<u>1</u> , 9	<u>10</u> , 2, 2	<u>1</u> , 9	<u>1</u> , 9	<u>10</u> , 7	<u>1</u> , 9	<u>1</u> , 9	<u>8</u> , 1, <u>1</u> , 9	<u>11</u>
10 i b. fejl	<u>11</u>	<u>1</u> , 10	<u>1</u> , 10	<u>1</u> , 10	<u>1</u> , 10	<u>10</u> , 2, 2	<u>1</u> , 10	<u>1</u> , 10	<u>10</u> , 7	<u>1</u> , 10	<u>1</u> , 10	<u>8</u> , 1, <u>1</u> , 9	<u>11</u>
11 i l. fejl I	<u>11</u>	<u>1</u> , 11	<u>1</u> , 11	<u>1</u> , 11	<u>1</u> , 11	<u>10</u> , 2, 2	<u>1</u> , 10	<u>1</u> , 11	<u>10</u> , 7	<u>1</u> , 8	<u>1</u> , 11	<u>8</u> , 1, <u>1</u> , 9	<u>11</u>
12 i l. fejl II	<u>8</u> , 6, <u>1</u> , 12	<u>1</u> , 12	<u>1</u> , 12	<u>1</u> , 12	<u>1</u> , 12	<u>12</u> , 5, 2, 2	<u>12</u> , 14, 5	<u>1</u> , 12	<u>12</u> , 5, 7	<u>1</u> , 8	<u>1</u> , 12	<u>8</u> , 1, <u>1</u> , 9	<u>11</u>

FORKORTELSER:

- e. efter
- b. bilag
- l. linie
- bl. bilagslinie

Understregede tal betegner aktioner.

De øvrige tal er parametre til foregående aktion.

'Syntaktisk opbygning af korrekt inputmateriale'

```

<RC-FAKT input> ::= <fysisk strimmel>1⊥
<fysisk strimmel> ::= <logisk strimmel>1⊥
<logisk strimmel> ::= <kommentar>0⊥<bilag>0⊥<strimmelsslut>
<bilag> ::= <normal bilag> | <annulbilag>
<normal bilag> ::= <bilagslinie><linie>0⊥
<annulbilag> ::= <bilagsstart><bilagstegn>0⊥
    <bilagsannullering><bilagstegn>0⊥
<bilagslinie> ::= <bilagsstart><bilagstype><liniehale>
<linie> ::= <kropslinie> | <annullinie> | <totallinie>
<kropslinie> ::= <liniestart><linietype><liniehale>
<annullinie> ::= <liniestart><annulrest>
<totallinie> ::= <liniestart><totallinietype><totallinierest>
<totallinierest> ::= <terminator><total-tal><linieafslutning> | <annulrest>
<annulrest> ::= <linietegn>0⊥<linieannullering><linietegn>0⊥
<liniehale> ::= <linieelement>1⊥<linieafslutning>
<linieelement> ::= <terminator><oplysning>
<oplysning> ::= <heltal> | <tekst> | <dato>
<bilagstype> ::= <heltal i intervallet 1-13>
<linietype> ::= <heltal i intervallet 1-136>
<total-linietype> ::= 1023
<total-tal> ::= <heltal>
<tekst> ::= <tekstdel>030<blinde>0⊥
<dato> ::= <dag><md><år>
<dag> ::= <heltal i intervallet 1-31>
<md> ::= <heltal i intervallet 1-12>
<år> ::= <heltal i intervallet 65-99>
<heltal> ::= <taltegn>011<talblinde>0⊥

```

<linieafslutning>	::= <linieslut><kommentar> ₀ [‡]
<kommentar>	::= <blinde> <talblinde teksttegn> <ciffer> <teksttegn>
<bilagstegn>	::= <tegn af klasse 0,1,2,3,4,6,7,9,10 eller 11>
<linietegn>	::= <tegn af klasse 0,1,2,3,4,7,9 eller 10>
<taltegn>	::= <talblinde> ₀ [‡] <ciffer>
<tekstdel>	::= <blinde> ₀ [‡] <teksttegn>
<talblinde>	::= <blinde> <talblinde teksttegn>
<blinde>	::= <tegn af klasse 0>
<talblinde teksttegn>	::= <tegn af klasse 1>
<ciffer>	::= <tegn af klasse 2>
<teksttegn>	::= <tegn af klasse 3>
<terminator>	::= <tegn af klasse 4>
<bilagsstart>	::= <tegn af klasse 5>
<liniestart>	::= <tegn af klasse 6>
<linieslut>	::= <tegn af klasse 7>
<strimmelslut>	::= <tegn af klasse 8>
<linieannullering>	::= <tegn af klasse 9>
<bilagsannullering>	::= <tegn af klasse 11>

Individ- og linie klassifikation

Bilagstype (bilagslinietype)	Intern bilagstype	Hoved- type	Under- type	Mulige linietyper
1	3	12	1	1-4
2	3	12	2	1-4
3	3	12	3	1-4
4	2	6	2	Gælder kun bilagslinien
4	2	6	4	5-6
4	2	6	5	7-10
5	1	9	1	Gælder kun bilagslinien
5	1	9	4	11-28
5	1	9	5	32-33
5	1	9	6	29-31
5	1	9	7	34
6	9	0	4	2-24,28,35,42,46-47
7	6	0	1	2-46,99,106-108
8	7	0	2	2-12
9	8	0	3	2-11
10	13	15	4	Ingen
11	10	15	1	Ingen
12	11	15	2	Ingen
13	12	15	3	Ingen



REGISTERBESKRIVELSE

Registeridentifikation:

RC-FAKT 1/54OE

2.1

Udarbejdet af: BØT	Betegnelse: 2.1 Indlæst materiale (usorteret)	Side 1 af: 1	
Dato: 21.02.69		Arkiv ref:	
Output fra: fase 1	I/O-system: Bent Larsen GSL 263 ff	Checksum i celle: nej	
	Etikettesystem: systemets faste etiketter		
Input til: fase 2 og fase 3	Max. bloklængde: 320 ord	Paritet: XIX / ulige	Individlængde i celle: nej
	Max. individlængde: 20	Variabel: ja / XIX	

Yderligere information: 1) sorteringskriterier 2) antal individformater 3) format bestemmes af:

Materialet skrives altid på station nr 2 fil nr 1

Individerne identificeres efter ord 1 .9 hovedtypen
5 .9 undertypen
og evt. 5 .19 linietypen

I alt findes 24 (mere eller mindre) forskellige individ-typer

Den fælles opbygning af individerne er beskrevet for sig.

Der skelnes mellem 2 typer individer :

bilags-individer, der indeholder oplysninger fælles for et helt bilag
(altså fra bilagslinien).

kropsindivider , der indeholder oplysninger fra de enkelte kropslinier
(evt. suppleret med oplysninger fra bilagsindividet).

Kartoteksændringer og sletninger er pr definition bilagsindivider

Ændringer (reference, ændret af, dato, gl. værdi):



INDIVIDBESKRIVELSE

Registeridentifikation:

RC-FAKT 1 / 540 E

2.1

Udarbejdet af:
BØT

Individtype:

Side:

0

Dato:
21.02.69

Fælles individ-format

Arkiv ref:

Individlængde:
20

Genereret af:
fase 1

Beskrivelse under register:

Ord nr.	Fra pos.	Til pos.	Betegnelser	Repræsentation / ordformat / værdigrænser / henvisning
0	0	2	definition af konto/vare	nummer : 0 varenummer
				1 debitornummer
				2 kreditornummer
				3 kovedbogskontonummer
0	3	39	konto/vare-nummer	
1	0	9	hovedtype	0, 6, 9, 12 eller 15
2	0	39	dato	specielt format 384xår +32x(md-1) + dag
3	0	39	bilagsnummer	
4	0	39	indlæsenummer	definerer indlæserækkefølgen, skifter 10 for hvert individ
5	0	9	undertype	
5	10	19	linietype	= 0 for bilags-individer = extern linietype for kropsindivider

Ændringer (reference, ændret af, dato, gl. værdi):

