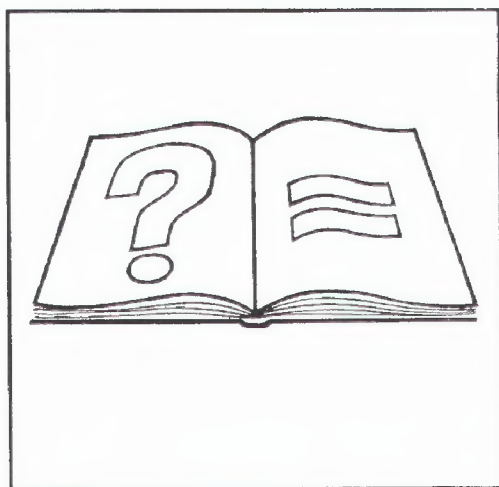


STATISTISK SENTRALBYRÅS HÅNDBØKER



HÅNDBOK FOR BRUK AV DATSY

STATISTISK SENTRALBYRÅ
OSLO

HÅNDBOK FOR BRUK AV DATSY

Statistisk Sentralbyrå

Oslo 1974

ISBN 82-537-0329-5

FORORD

I de senere årene har det blitt en stadig viktigere oppgave for Statistisk Sentralbyrå å bearbeide videre den innsamlede og arkiverte datamasse fra de ulike statistikkområder. Bearbeiding av statistiske data for analytiske formål stiller andre krav til programutstyr for databearbeiding enn den rene statistikkproduksjon. Byråets arbeid med utvikling av økonomiske modeller for analyse, prognose og planlegging har stilt særlig store krav til slikt programutstyr. I tilknytning til arbeidet med modellen MODIS IV tok Byrået initiativet til å få utviklet et generelt system for programmering av MODIS IV og lignende prosjekter. Et slikt system - kalt DATSY - ble utviklet av Norsk Regnesentral under ledelse av professor Sverre Spurkland på kontrakt for Finansdepartementet. Tilpassing av systemet for Byråets behov er utført i Byrået.

Denne håndboka gir en innføring og mer detaljert veiledning i bruken av DATSY. Systemet er i produktiv bruk, men har foreløpig et svært begrenset anvendelsesområde. Beskrivelsen av systemets muligheter vil være noe preget av dette. Videre utvikling av systemet vil skje i forbindelse med nye anvendelsesområder og bli dokumentert som Arbeidsnotater fra Statistisk Sentralbyrå.

Håndboka har blitt redigert av forsker Olav Bjerkholt. Sekretær Lars E. Aukrust skrev de opprinnelige utkast til del A og del B på basis av dokumentasjon av DATSY fra Norsk Regnesentral. Konsulent David Walker har tilrettelagt det meste av del C med bidrag fra forsker Dan Wøien, Norsk Regnesentral og konsulent Pål Lynum.

Statistisk Sentralbyrå, Oslo, 2. januar 1974

Petter Jakob Bjerve

INNHOLD

	Side
A. INNFØRING I DATSY FOR NYE BRUKERE	7
A-1. Generell innledning	7
A-1.1 Hva er DATSY?	7
A-1.2 Hva kan DATSY brukes til?	7
A-1.3 Hvilke fordeler og ulemper har DATSY?	9
A-2. Eksempler på bruk av DATSY	10
A-2.1 Eksempel 1. Direktiver, deklarasjoner, matriseaddisjon, arkivtape	10
A-2.2 Eksempel 2. Parametre, subtraksjon, innlesing fra hullkort	12
A-2.3 Eksempel 3. Innlesing av data fra hullkort med format	13
A-2.4 Eksempel 4. Lister, deling av lister, overføring av data til arkivtape	14
A-2.5 Eksempel 5. Rekordsett, innlesing av rekordsett, sortering	16
A-2.6 Eksempel 6. Redigering av rekordsett	18
B. DETALJERT VEILEDNING I BRUK AV DATSY	21
B-1. Grunnelementer i en DATSY-tekst	21
B-1.1 Tegn	21
B-1.2 Ord og ordtyper	21
B-1.3 Linje	22
B-1.4 Setning	22
B-1.5 Kommentar	22
B-2. Objekter, objektetegnavn, objektklasser	22
B-2.1 Dataobjekter	22
B-2.2 Hjelpobjekter	22
B-2.3 Innlesingsobjekter	23
B-2.4 Struktureringsobjekter	23
B-2.5 Lagringsobjekter	24
B-3. Setningstyper	25
B-3.1 Direktivsetninger	25
B-3.2 Deklarasjonssetninger	25
B-3.3 Arkivtapesetning	26
B-3.4 Deskripsjonssetninger	26
B-3.5 Hjelpesetninger, substituering	27
B-3.6 Hjelpesetninger for magnetbånd	27
B-3.7 Sluttsetninger	29
B-3.8 Tomme setninger	29
B-4. Innføring av data fra hullkort	30
B-4.1 Formatkort	30
B-4.2 Objektkort	31
B-4.3 Innlesing av parameter	31
B-4.4 Innlesing av navnord	31
B-4.5 Innlesing av liste	32

	Side
B-4.6 Innlesing av matrise	32
B-4.7 Innlesing av vektor	32
B-4.8 Innlesing av rekord	33
B-4.9 Innlesing av rekordsett	33
B-4.10 Generelt om formater	33
B-4.11 Innlesing av fremmedtape	35
B-5. Strukturen av et DATSY-program	36
B-5.1 DATSY-teksten	36
B-5.2 Dataavdelingen	38
B-6. Utskrifter fra DATSY	38
B-6.1 Ordinære utskrifter	38
B-6.2 Feilmeldinger	39
B-6.3 Utskrift av dataobjekter	40
B-6.4 Opsjoner på UNIVAC 1108	40
B-7. Dataobjekter med identiske navn	41
C. DIREKTIVBESKRIVELSER	42
C-1. Innledning	42
C-2. Generelle direktiver	42
C-2.1 Opplysnings- og utskriftdirektiver	42
C-2.2 Kontroll- og etableringsdirektiver	52
C-2.3 Konverteringsdirektiver	58
C-2.4 Matematiske direktiver	62
C-2.5 Liste- og sublistedirektiver	73
C-2.6 Rekordsettdirektiver	80
C-2.7 Andre manipuleringsdirektiver	86
C-2.8 Direktiver for regresjonsanalyse	90
C-3. Spesielle direktiver	106
C-3.1 Direktiver for MODIS IV	106
V e d l e g g	
Utkommet i serien Statistisk Sentralbyrås Håndbøker (SSH)	115

A. INNFORING I DATSY FOR NYE BRUKERE

A-1. Generell innledning

I del A vil det bli gitt en førstehandsorientering om DATSY som ikke forutsetter kjennskap til programmering og datamaskiner.

I denne delen er det forsøkt å besvare følgende spørsmål:

1. Hva er DATSY?
2. Hva kan DATSY brukes til?
3. Hvilke fordeler og ulemper har DATSY?

A-1.1 Hva er DATSY?

DATSY er utviklet for å tjene som et verktøy, for dataorganisering, modellformulering og matri-seberegning. Formålet har vært å lage et bruker- og problemorientert programmeringsystem.

DATSY-systemet inneholder en katalog og et sett direktiver. Katalogen inneholder en fortegnelse over og beskriver innholdet av alle data som er kjent for systemet. Direktivene kan utføre arbeidsoperasjoner. Kjennskap til vanlige programmeringsspråk er ikke nødvendig for enkel bruk av DATSY. Brukeren kan benytte direktivene ved å skrive en sekvens av DATSY-setninger. En slik sekvens kalles en DATSY-tekst. Eksempler på arbeidsoperasjoner direktivene kan utføre, kan være matrisemultiplikasjon, ekstrahering og sortering av rekordsett og regresjonsberegninger.

Den sentrale operative del av DATSY-systemet er et kontroll- og styreprogram som organiserer bruken av katalogen og settet av direktiver. De data som er gjort kjent for systemet oppbevares på magnetbånd. Disse kalles arkivtaper og utgjør til sammen et arkiv. Navnene på arkivtapene og opplysninger om hva de inneholder er innført i katalogen. For praktiske formål har arkivet en begrenset kapasitet.

DATSY er utviklet av Norsk Regnesentral etter oppdrag fra Finansdepartementet. Det er først og fremst kontroll- og styreprogrammet og katalogsystemet som er utviklet av Norsk Regnesentral. De fleste direktivene er utarbeidd i Statistisk Sentralbyrå.

A-1.2 Hva kan DATSY brukes til?

DATSY ble opprinnelig laget med henblikk på framtidig arbeid med økonomiske analyser og store planleggingsmodeller. I dag er DATSY et hjelpemiddel ved implementering og bruk av modeller. Systemet er utformet med sikte på at det skal være lett å foreta modifikasjoner av og eksperimenter med modellene.

I utformingen av systemet ble det lagt særlig vekt på enkle brukeregenskaper, standardisert og automatisert input/output, fleksible matrise- og rekordsettoperasjoner, utvidbart system, selv-dokumenterende program, sikkerhet for data og oversikt over kjente data. Kontroll- og styreprogrammet er en fast del av systemet, mens katalogen ajourføres automatisk og settet av direktiver kan utvides til å omfatte nye arbeidsoperasjoner. Utvidelsesmulighetene er viktige egenskaper ved systemet når det skal nyttes til formulering og igangsetting av nye modeller. Det er tatt sikte på å få et nært samsvar mellom en modell i matematiske likninger og de direktiver som skal til for å gjennomføre en løsning av modellen. Systemet har i første omgang blitt brukt til å implementere fjerde versjon av Statistisk Sentralbyrås prognosemodell MODIS. De direktiver som hittil er utviklet er tilpasset dette formål. Men DATSY vil kunne tilpasses andre brukere og problemområder.

Figuren på neste side gir en skjematisk framstilling av bruken av systemet. En bruker av DATSY har et problem som forutsetter visse operasjoner utført på et sett av data. Hans data kan være kjent for DATSY, dvs. innført i katalogen og oppbevart i arkivet, eller ikke. "Fremmede" data forutsettes å være lagret på et magnetbånd eller på hullkort.

Brukeren ønsker altså å få utført visse operasjoner på sine data for å løse sitt problem. Ved hjelp av brukermanualen skriver han et DATSY-program som angir hvilke direktiver han ønsker utført. Ved bruk av "fremmede" data må DATSY-programmet også inneholde de opplysninger systemet trenger for å kunne tolke magnetbåndet eller hullkortene som inneholder data.

En rekke mer generelle operasjoner kan utføres ved hjelp av direktiver. Direktiver for behandling av matriser og for regresjonsberegninger er velutviklet. Hvis det imidlertid skulle vise seg nødvendig med operasjoner som ikke kan utføres ved å sette sammen direktiver som foreligger, er dette mulig ved å innføre nye direktiver som i dag ikke foreligger. Slike nye direktiver tenkes utarbeidd av andre enn brukeren selv, da dette vil kreve programmeringsinnsats og mer inngående kjennskap til DATSY-systemet. Det er imidlertid tilgjengelig programutstyr som letter skriving og oppbygging av nye direktiver.

(Selve eksekveringen av direktivprogrammet på maskinen vil foruten DATSY-programmet kreve visse styrekort. Disse vil avhenge av hvilken maskin man vil benytte.)

For hvert brukerområde for DATSY vil det forutsettes at det er en dataadministrator som holder oversikt over arkivet. Han vil imidlertid være hjulpet av den automatiske ajourføring av katalogen i DATSY.

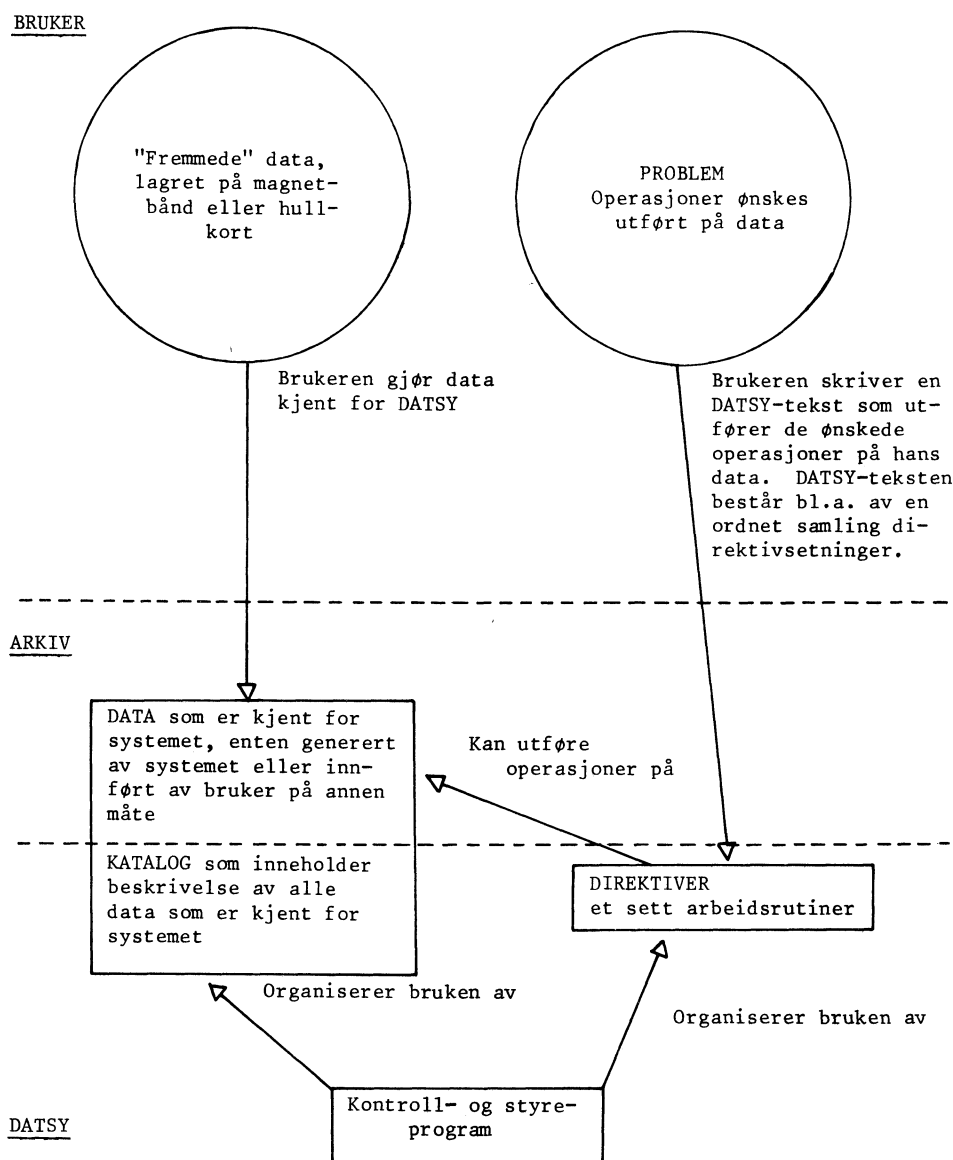


Fig. 1

A-1.3 Hvilke fordeler og ulemper har DATSY?

Den største fordelen en har tatt sikte på å oppnå ved utviklingen av DATSY, er at det skal være mulig for en bruker uten kjennskap til programmering å få utført databehandlings- og beregningsoppdrag. Som nevnt ovenfor vil brukeren for å få utført et oppdrag kunne nøye seg med å gi systemet opplysning om hvilke direktiver som ønskes utført og eventuelt om "fremmede" data. Disse opplysninger kan gis en lett tilgjengelig brukervennlig form. DATSY sparer brukeren for å angi en rekke opplysninger som er nødvendig for å få utført oppdraget på maskinen. Dette gjelder særlig opplysninger om input-output og den interne plassutnytting i maskinens hurtigminne og mellomlager (trommer, plater).

Direktivprogrammene kan dessuten langt på vei gjøres selvdokumenterende. Et program skrevet med DATSY-direktiver kan med litt innsats utformes som vanlige setninger. Følgende direktivsetning som kan være hentet fra et DATSY-program illustrerer dette:

ADDER MATRISEN MAT1 TIL MATRISEN MAT2 OG SETT SVARET INN I MATRISEN MAT3.

(Nærmere tolking av direktivet med forklaring av understrekningene vil bli gitt i A-2.1.)

Som et supplement til den brukervennlige spesifisering av oppdrag er det lagt stor vekt på at feilmeldinger og diagnostikk også er gitt en utforming som skal være forståelig for en bruker uten spesielt kjennskap til programmering. Spesielt er det tatt sikte på at visse typer feil i brukerens spesifisering av problemet skal oppdages av DATSY's kontroll- og styreprogram før selve eksekveringen av operasjonene.

DATSY er som nevnt ovenfor, utviklet for et bestemt problemområde, nemlig konstruksjon og drift av store økonomiske modeller. På dette område var det tidligere innvunnet mye erfaring særlig om ulemper ved mer direkte programmeringsmetoder. Dette forsøkte en å nyttiggjøre seg ved konstruksjonen av DATSY.

Hittil har DATSY blitt forholdsvis lite brukt for produktive formål. En må regne med at ikke alle ulemper ved bruk av DATSY har kommet klart fram. Et generelt trekk ved systemet er at det er lite fleksibelt på kort sikt. DATSY inneholder ikke et programmeringsspråk som kan brukes til å bygge opp direktiver som en del av systemet selv. Antall direktiver er gitt til enhver tid og nye direktiver må konstrueres utenfor systemet i et ordinært programmeringsspråk og "plugges inn" i DATSY. Unntak fra dette er muligheter som gis i DATSY for å konstruere "sammensatte" direktiver ved hjelp av direktiver som allerede finnes i systemet. Heller ikke dette kan gjøres uten en del innsats og kostnader.

Det har vist seg at systemet stiller store krav til maskinanlegget når det gjelder tilgjengelighet av kapasitet i hurtigminne og mellomlager og av båndstasjoner. Systemet er såpass plasskrevende på maskinen at det har vært nødvendig å pålegge en rekke skranker som er bindende for brukeren når det gjelder f.eks. antall kort med DATSY-tekst i en jobb, antall objekter, antall arkivtaper osv. Input og output av data til DATSY må foreligge på magnetbånd eller hullkort, altså ikke magnetiske platestasjoner eller trommer.

Alt i alt innebærer disse restriksjonene at systemet ikke blir så brukervennlig som opprinnelig forutsatt.

I DATSY-systemet er datamengden alltid forutsatt å tilhøre én av et lite antall klasser, som f.eks. matrise, rekordsett og vektor. Dette har gjort det mulig å standardisere en rekke operasjoner slik at de foregår uten spesifikasjoner fra brukeren. Til gjengjeld kan det hende en her mister noe fleksibilitet og eventuelt maskineffektivitet jamført med et system som ville krevd større brukerinn-sats. Til oppdrag som krever meget maskineffektive løsninger, f.eks. fordi datamengdene som inngår er meget store, kan derfor DATSY vise seg mindre godt egnet.

A-2. Eksempler på bruk av DATSY

Dette kapitlet er bygd opp omkring en serie eksempler. Eksempelene er forsøkt valgt slik at de til sammen belyser ulike problemer som en bruker kan møte. Rekkefølgen av eksemplene er valgt slik at de viktigste og enkleste definisjoner og framgangsmåter belyses først. Etter hvert øker vanskelighetsgraden og mer spesielle problemer belyses. Hvor langt en leser ønsker å lese i dette kapitlet, avhenger av hvor grundig en ønsker å sette seg inn i skriving av DATSY-programmer. Gjennomlesing av eksempel 1 vil være nok til å gi et visst generelt inntrykk av hvordan et DATSY-program bygges opp. I eksemplene gis det henvisninger til relevante kapitler i del B.

A-2.1 Eksempel 1. Direktiver, deklarasjoner, matriseaddisjon, arkivtape

Brukeren har sine data og et problem. Anta at data består av to matriser. Matrisene har egennavnene MAT1 og MAT2. Matrisene er oppbevart i DATSY-arkivet på en arkivtape som har egennavnet A33. Egennavnene MAT1, MAT2 og A33 er vilkårlig valgt. I eksemplet antas at matrisene er kvadratiske og ser slik ut:

$$\begin{array}{cc} \text{MAT1} & \text{MAT2} \\ \left[\begin{array}{cc} 1 & 2 \\ 1 & 1 \end{array} \right] & \left[\begin{array}{cc} 2 & 1 \\ 0 & 1 \end{array} \right] \end{array}$$

Brukeren ønsker å få maskinen til å addere de to matrisene sammen, og skrive svaret ut på papir. Det vil være nødvendig å starte med å gjøre brukeren kjent med visse sentrale betegnelser og begreper innen DATSY.

Objekt, objektklasse, objekttegnavn

Brukerens data betegnes som objekter. Hver matrise er et objekt. Objektene deles inn i objektklasser. Eksempler på slike klasser er matriser, vektorer, parametre, lister og rekordsett. Hvert objekt må videre ha et objekttegnavn. I vårt eksempel består brukerens data av to matriser. I DATSY terminologi vil det si at brukerens data består av to objekter som begge faller inn under (objekt-) klassen matriser, og deres (objekt-)egnavn er MAT1 og MAT2. Objektbegrepet står svært sentralt i DATSY. Kapittel B-2 gir en mer inngående innføring i dette.

Setninger

Et DATSY-program bygges opp av setninger. Det finnes flere typer setninger. Den viktigste er direktivsetningen, og vi starter med den. De forskjellige setningstyper er mer utførlig beskrevet i kapittel B-3. (Se B-1.3.)

Direktivsetninger

En bruker av DATSY har vanligvis til hensikt å få utført visse arbeidsoperasjoner på sine data-objekter. Eksempler på slike arbeidsoperasjoner er addisjon, og multiplikasjon av matriser og redigering av rekordsett. Slike arbeidsoperasjoner er i DATSY-terminologi betegnet som direktiver. Brukeren benytter direktivene ved å skrive direktivsetninger. Direktivsetninger er således brukerens instruks til maskinen om de arbeidsoperasjoner som ønskes utført.

I A-1.3 inngikk et eksempel på en direktivsetning:

ADDER MATRISEN MAT1 TIL MATRISEN MAT2 OG SETT SVARET INN I MATRISEN MAT3.

Alle direktivsetninger består av et direktivnavn og ett eller flere objekttegnavn. Direktivnavnet står alltid først i en direktivsetning. Objekttegnavnene følger etter hverandre i en bestemt rekkefølge. I dette eksemplet er verbet ADDER og objekttegnavnene MAT1, MAT2 og MAT3.

Direktivnavn er navn på direktiver, altså på operasjoner som kan utføres på objekter, og de er forsøkt valgt slik at de indikerer hvilken operasjon direktivet utfører. Det skulle være lett å forstå hva direktivet ADDER gjør.

I det eksempel som her er gitt på en direktivsetning, ses at visse ord er understreket. Alle direktivnavn er gitt dobbelt understrekning. Alle ord som har enkel understrekning er egennavn på objekter. Alle ord som ikke er understreket er "tomme" ord. Disse har ingen betydning og blir ignorert av maskinen. (Se B-1.2.) Følgende direktivsetning ville gitt samme resultat som den som er tatt som eksempel ovenfor:

ADDER MAT1 MAT2 MAT3.

Når brukeren får igjen utskrift fra maskinen inneholder denne utskriften bl.a. det program han selv skrev (se B-6.1). Utskriften av DATSY-programmet er automatisk understreket på den måten eksemplet viser. Understrekningen er gjort for å lette lesing av DATSY-programmet etterpå.

Viktig: En setning må alltid avsluttes av punktum. Dette er absolutt nødvendig fordi punktum er maskinens måte å skille mellom en setning og den neste. Det er en vanlig nybegynnerfeil å glemme punktum.

Arkivtapesetningen

Som nevnt er direktivsetninger den klart viktigste type setning. En annen setning som blir mye benyttet er arkivtapesetningen. Denne setningen står i en viktig særstilling. Arkivtapesetningen blir benyttet hver gang brukerens data på forhånd ligger på en arkivtape i DATSY-arkivet, og bare da. I vårt eksempel skal vi anta at de to matrisene MAT1 og MAT2 ligger på arkivtappen A33. Dette magnetbåndet må monteres på en båndstasjon på maskinen, og innholdet må leses inn i maskinen. (Med et DATSY-program skal det alltid følge visse styrekort og disse styrekortene sørger bl.a. for at den aktuelle arkivtape blir montert.) Arkivtapesetningen i brukerens DATSY-program er nødvendig for å få magnetbåndet innlest av maskinen. I vårt eksempel vil arkivtapesetningen ha følgende utseende:

ARKIVTAPE A33.

Deklarasjonssetninger

En annen type setning er deklarasjonssetningen. En deklarasjonssetning sørger for at egennavnet til et objekt gjøres kjent for systemet. Deklarasjonssetningen plasserer samtidig objektet i en bestemt klasse. I eksemplet skal to matriser adderes. Det er klart at under eksekvering av dette genereres en ny matrise. Denne matrise er ny for systemet og den har derfor ikke noe egennavn. Det får den ved hjelp av følgende deklarasjonssetning:

MATRISE MAT3.

Virkningen av dette er at maskinen nå vet at MAT3 er egennavnet til et objekt som tilhører klassen matrise. I vårt eksempel blir objektets innhold generert under regneoperasjonen. Innholdet kunne imidlertid også blitt innført fra f.eks. data på hullkort (se B-4.). I motsetning til direktivsetninger kan deklarasjonssetninger ikke inneholde tomme ord. Deklarasjonssetninger kan betraktes som definisjoner som setter navn på nye objekter.

Eksempler på andre deklarasjonssetninger som ikke er relevante for eksempel 1 er:

PARAMETER PI.

LISTE NAVNLISTE.

Flere objekter kan deklarerer i en og samme deklarasjonssetning. "Mellomrom" mellom objektene på kortet forteller maskinen at det er flere objekter, f.eks.

MATRISE MAT1 MAT2 MAT3.

Utskriftsetninger

I eksemplet antas at brukeren ønsker å få innholdet av den nye matrisen MAT3 skrevet ut på papir. Det finnes direktiver for dette formål. Hvordan man kan gå fram for å få objekter skrevet ut på papir framgår av avsnitt B-6.1. Brukeren må da skrive en utskriftsetning. Direktivet PRINTMAT skriver matriser ut på papir. Brukeren skriver følgende utskriftsetning:

PRINTMAT MAT3.

Sluttsetning

En DATSY-tekst må alltid avsluttes med en sluttsetning. Denne sluttsetning er vanligvis

SLUTTPROD.

Oppsummering av Eksempel 1

Brukerens framgangsmåte kan kort oppsummeres slik:

Han kontakter dataadministratoren og får vite at de to matriser han ønsker å benytte, MAT1 og MAT2, ligger på arkivtapen A33. Han slår opp i direktivbeskrivelsene (se del C) og finner de direktiver han trenger, ADDER og PRINTMAT. Han velger å kalle den nye matrisen som oppstår for MAT3. Han skriver så følgende DATSY-program:

ARKIVTAPE A33.

MATRISE MAT3.

ADDER MATRISEN MAT1 TIL MATRISEN MAT2 OG SETT SVARET I MATRISEN MAT3.

PRINTMAT MAT3.

SLUTTPROD.

Disse setninger punches så på hullkort (eller skrives på en teletype). Hvis en setning ikke får plass på ett hullkort (linje), kan setningen fortsettes på et nytt hullkort. Det er punktumet etter hver setning, ikke overgangen til et nytt hullkort, som forteller maskinen at setningen er avsluttet (se B-1.4).

Programmet med styrekort leveres så til regnemaskinen med DATSY-systemet. Etter kjøring får brukeren tilbake sin hullkortbunke og en papirutskrift fra maskinen. Denne utskriften inneholder bl.a. hans program med understrekinger slik som tidligere forklart. Dessuten får han skrevet ut matrisen MAT3. Utskrift med eventuelle feilmeldinger er beskrevet i avsnitt B-6.1 og B-6.2. Den siste del av utskriften vil se omtrent slik ut:

PRINTMAT MAT3.

	1	2
1	3,000	3,000
2	1,000	2,000

A-2.2 Eksempel 2. Parametre, subtraksjon, innlesing fra hullkort

Det eneste prinsipielt nye som innføres i dette eksemplet (og eksempele 3) er at brukers data ikke er kjent for systemet på forhånd. Dataene må derfor føres inn i maskinen fra hullkort. I dette tilfellet vil DATSY-programmet inkludere en dataavdeling etter DATSY-teksten (setninger).

Brukerens problem i eksempele 2 består i å subtrahere et tall fra et annet.

1 464 - 708 = ?

Brukerens data består følgelig av to objekter og det skal genereres et nytt objekt. Alle tre objekter er av klassen parameter. Brukeren velger å kalle sine to tall for henholdsvis P1 og P2 og svaret for SVARPARAMETER.

Ingen av de tre objekter er kjent for maskinen og brukeren skriver derfor følgende deklarasjonssetning:

PARAMETER P1 P2 SVARPARAMETER.

Brukeren leter i direktivbeskrivelsene og finner direktivet SUBTRAHER (se C-2.4):

SUBTRAHER parameter1/vektor1/matrise1 + parameter2/vektor2/matrise2 = parameter3/vektor3/matrise3.

Skråstrekene angir at objektene kan være av forskjellige klasser.

Dette direktiv utfører den ønskede operasjon. Brukeren formulerer følgende direktivsetning:

SUBTRAHER TAR PARAMETEREN P1 MINUS PARAMETEREN P2 OG SETTER SVARET INN I PARAMETEREN SVARPARAMETER.

Brukeren ønsker svaret skrevet ut på papir. Dette gjøres ved følgende utskriftsetning (se B-6.3):

PRINTVERDI SVARPARAMETER.

De to parameterverdiene er imidlertid ikke kjent for maskinen. De må skrives på hullkort og legges etter selve DATSY-programmet. For at maskinen skal oppfatte kortene som datakort må de ha en stjerne som første tegn (se B-4).

De datakort som er nødvendige til vårt eksempel er:

*P1 = 1 464

*P2 = 708

Det må her bemerkes at når data skal leses inn fra hullkort så er parametre (og navnord) et særtilfelle. Vanligvis vil det være nødvendig å fortelle maskinen hvordan dataene er skrevet på hullkortene. Dette gjøres med en egen type kort som kalles formatkort. For brukere med kjennskap til FORTRAN vil dette være enkelt fordi formatkortene i DATSY likner meget på FORTRAN-formater. For andre brukere kan det være noe komplisert og tidkrevende å sette seg fullstendig inn i bruken av formatkort. Dette vil imidlertid vanligvis ikke være nødvendig. Kapittel B-4 beskriver innlesing av data fra hullkort.

Det endelige program med DATSY-tekst og dataavdeling som løser brukerens problem i eksempel 2 kan da se ut som følger:

PARAMETER P1 P2 SVARPARAMETER.

SUBTRAHER TAR PARAMETEREN P1 MINUS PARAMETEREN P2 OG SETTER SVARET I PARAMETEREN SVARPARAMETER.

PRINTVERDI SVARPARAMETER.

SLUTTPROD.

*P1 = 1 464

*P2 = 708

A-2.3 Eksempel 3. Innlesing av data fra hullkort med format

Dette eksemplet vil ytterligere illustrere hvordan data leses inn fra hullkort. I eksempel 2 kunne dataene leses inn i såkalt "fritt format". Det er imidlertid bare parametre og navnord som kan leses inn i "fritt format". Eksempel 3 er en variant av eksempel 1. De to matrisene som skal adderes, antas nå ikke kjent for maskinen, men forutsettes lest inn fra hullkort. Hensikten med eksemplet er å gi brukeren en innledning til kapittel B-4 "Innlesing av data fra hullkort".

La oss anta at brukeren velger å starte med de nødvendige deklarasjonssetninger. Verken de to matrisene som skal leses inn, matrisen som genereres, eller formatet som skal benyttes antas å være kjent for maskinen på forhånd. Brukeren skriver da følgende deklarasjonssetninger:

MATRISE MAT1 MAT2 MAT3.

FORMAT F66.

Resten av setningene i selve DATSY-programmet kan stå uforandret fra eksempel 1, med unntak av at arkivtapesetningen utgår.

ADDER MATRISEN MAT1 TIL MATRISEN MAT2 OG SETT SVARET I MATRISEN MAT3.

PRINTMAT MAT3.

SLUTTPROD.

Etter DATSY-teksten må det følge en dataavdeling som inneholder de to sett med data, dvs. MAT1 og MAT2, med hvert sitt tilhørende objektkort og et formatkort. (Det forutsettes at samme format kan brukes til å lese inn begge matrisene.)

La oss anta at brukeren velger å lese matrisene inn med formatet (F14.8). Dette angir at matrisens elementer er punchet i felter med lengde 14 og det skal være 8 plasser bak komma. Han skriver da følgende formatkort:

*F66 = (F14.8)

De to objektkort kan videre se ut som følger:

*MAT1, R=2, K=2, F66

*MAT2, R=2, K=2, F66

Disse angir at begge matrisene har to rader, to kolonner og er punchet med format F66.

Det siste som da gjenstår er å punche dataene på den måte som formatet forutsetter. Rekkefølgen av kortene i dataavdelingen er ikke tilfeldig. Elementene i hver matrise må samles for seg og umiddelbart foran hvert sett med datakort må det tilsvarende objektkort plasseres. Når maskinen leser inn en rekke matriseelementer, plasserer den elementene i matrisen fra venstre mot høyre radvis ovenfra og nedover.

Det endelige program vil da se slik ut:

MATRISE MAT1 MAT2 MAT3.

FORMAT F66.

ADDER MATRISEN MAT1 TIL MATRISEN MAT2 OG SETT SVARET I MATRISEN MAT3.

PRINTMAT MAT3.

SLUTTPROD.

*F66 = (F14.8)

*MAT1, R=2, K=2, F66

bbbb1bbbbbbb

bbbb2bbbbbbb

bbbb1bbbbbbb

bbbb1bbbbbbb

*MAT, R=2, K=2, F66

bbbb2bbbbbbb

bbbb1bbbbbbb

bbbb0bbbbbbb

bbbb1bbbbbbb

b betyr blank dvs. ikke punchet.

A-2.4 Eksempel 4. Lister, deling av lister, overføring av data til arkivtape.

Dette eksemplet illustrerer hvordan data kan leses ut på en arkivtape.

Vi forutsetter at en liste med egennavnet ALISTE befinner seg på en arkivtape. Arkivtapen har egennavnet LISTETAPE. Listen består av alfanumeriske elementer (navnord) der hvert element har eksempelvis 6 tegn. Vi ønsker å få listen delt "vertikalt" ved at det av listen lages to nye lister med like mange elementer som ALISTE. Hvert element i den første av disse utgjøres av de fire første tegn i det tilsvarende element i ALISTE, mens hvert element i den andre listen utgjøres av de to siste tegn av det tilsvarende element i ALISTE. De to listene som genereres vil vi ha skrevet ut på papir og dessuten lagret på en arkivtape.

Brukeren må først avgjøre på hvilken arkivtape han skal skrive ut de to nye listene. Han legger fram problemet for dataadministratoren. Brukeren får for eksempel vite at han kan benytte en tom tape som har egennavnet A760. Denne tape er registrert i DATSY-arkivet og er således kjent for maskinen som en arkivtape.

I direktivbeskrivelsene finner brukeren beskrivelsen av direktivet DELLISTE (se C-2.7):

DELLISTE listel parameter liste2 liste3.

Dette direktivet arbeider med fire objekter. Det første objekt er listen som skal deles. Listen ALISTE ligger på arkivtapen LISTETAPE. Det er da ikke nødvendig med noen deklarasjonssetning for det første objektet, fordi det er kjent for maskinen på forhånd. En arkivtapesetning er imidlertid nødvendig:

ARKIVTAPE LISTETAPE.

Det neste objektet er en parameter. Det er denne parameteren som angir etter hvilket - regnet fra venstre - elementene i ALISTE skal deles. I dette eksemplet skal altså listen deles etter fjerde tegn. Brukeren velger å kalle denne parameteren for SAKS. Dette må gjøres kjent for maskinen med følgende deklarasjonssetning.

PARAMETER SAKS.

Verdien av denne parameteren skal leses inn fra hullkort (terminal). Som i eksempel 2 gjøres dette med et kort etter selve DATSY-programmet.

*SAKS = 4

De to siste objektene som inngår i direktivet er de to nye listene som genereres. Disse må gjøres kjent for systemet i en deklarasjonssetning. Brukeren velger å kalle dem UTLISTEA og UTLISTEB.

LISTE UTLISTEA UTLISTEB.

Selve direktivsetningen vil da få følgende form:

DELLISTE ALISTE SAKS UTLISTEA UTLISTEB.

For å få et mer oversiktlig og selvdokumenterende program kan direktivsetningen - som i de andre eksemplene - utvides med tomme ord, for eksempel:

DELLISTE DELER LISTEN ALISTE VERTIKALT I TO PÅ ET STED GITT VED PARAMETEREN SAKS SOM HAR VERDIEN FIRE OG KALLER DE TO NYE LISTENE UTLISTEA OG UTLISTEB.

De to nye listene ønskes skrevet ut på papir og oppbevart på magnetbånd. Utskriften på papir besørges som i tidligere eksempler av utskriftsetninger:

PRINTVERDI UTLISTEA.

PRINTVERDI UTLISTEB.

De nye listene skal så skrives på arkivtapen A760. Denne tape må først inngå i en arkivtapesetning, enten alene eller sammen med tapen LISTETAPE ARKIVTAPE LISTETAPE A760.

Det finnes en egen setning, skrivsetning, for å få objekter overført til arkivtaper.

SKRIV arkivtape Objekt1 Objekt2 Objektn.

Regler for bruk av denne setningen er som følger: Antall objekter kan variere. Arkivtapen må være tom. En konsekvens av at arkivtapen må være tom er at alt som skal skrives på tapen må skrives i en og samme skrivsetning. I dette eksemplet kan skrivsetningen se ut som følger:

SKRIV PÅ ARKIVTAPEN A760 OBJEKTENE UTLISTEA OG UTLISTEB.

Et DATSY-program som løser problemet i eksempel 4 kan da være:

ARKIVTAPE LISTETAPE.

ARKIVTAPE A760.

LISTE UTLISTEA UTLISTEB.

PARAMETER SAKS.

DELLISTE DELER LISTEN ALISTE PÅ ET STED GITT VED PARAMETEREN SAKS SOM HAR VERDIEN FIRE OG KALLER DE TO NYE LISTENE UTLISTEA OG UTLISTEB.

PRINTVERDI UTLISTEA.

PRINTVERDI UTLISTEB.

SKRIV PÅ ARKIVTAPEN A760 OBJEKTENE UTLISTEA OG UTLISTEB.

SLUTTPROD.

*SAKS = 4

A-2.5 Eksempel 5. Rekordsett, innlesing av rekordsett, sortering.

Eksempel 5 og 6 illustrerer bruken av rekordsett. Rekordsettoperasjoner kan virke noe mer komplisert for en ny bruker enn vektorer og matriser. Det er lagt stor vekt på å gjøre DATSY slagkraftig når det gjelder behandling av rekordsett. De viktigste datakilder vil typisk foreligge som rekordsett. I eksempel 5 skal et rekordsett leses inn fra hullkort og sorteres. I eksempel 6 skal et rekordsett som befinner seg på en arkivtape redigeres. Vanskeligheten for en ny bruker vil antakelig ligge i å bli kjent med en del nye begreper. For fullstendighetens skyld vil først rekord og rekordsett bli definert.

En rekord er betegnelsen på en ordnet sekvens av numeriske og alfanumeriske data. En rekord består således av ett eller flere elementærfelt. Hvert elementærfelt er enten alfanumerisk eller numerisk.

Et rekordsett er bygd opp av flere rekorder som har samme inndeling i elementærfelt.

Eksempel 5 er bygd opp omkring følgende rekordsett:

```

bbblbGOGSTADbbbbb1931bOSLObbbbbOSLObbb
  2 GUNDERSON   1916 DRAMMEN  OSLO
  3 ESPELAND    1942 DRØBAK   DRØBAK
  4 SAUER       1937 ASKER   OSLO
  5 SKILBREI    1924 ÅS       DRØBAK
  6 LUND        1945 ASKIM    ASKIM
  7 NALMO       1924 OSLO     ASKER
  8 NANSEN      1907 OSLO     OSLO

```

Elementærfeltene i rekorder og rekordsett er definert som objekter. De må derfor ha egennavn. Disse objektene kan videre være av to klasser, avhengig av om de inneholder numerisk eller alfa-numerisk informasjon. De to objektklassene, som bare forekommer i forbindelse med rekorder og rekordsett, er NAVN og TALL (se avsnitt B-2.4). Rekordsettet over består av fem elementærfelt. Objekt én og tre fra venstre er numeriske og er av klassen TALL. Objekt to, fire og fem er tilsvarende alfa-numeriske, i dette tilfelle rent alfabetiske, og av klassen NAVN.

NAVN er en klasse av objekter. I rekorder og rekordsett kalles elementærfelt med alfanumerisk informasjon for objekter av klassen NAVN. Hvis objektene er nye for maskinen må de få egennavn som gjøres kjent for maskinen med en deklarasjonssetning:

NAVN PERSONNAVN BOSTED FØDESTED.

TALL er en annen klasse av objekter. I rekorder og rekordsett kalles elementærfelt med numerisk informasjon for objekter av klassen TALL. Hvis objektene er nye for maskinen må de få egennavn som gjøres kjent for maskinen med en deklarasjonssetning:

TALL NUMMER FØDSELSÅR.

Brukerens problem i eksempel 5 er at han vil sortere rekordene i det gitte rekordsettet etter bosted. Bak i denne manualen finner han direktivet SORTER.

SORTER rekordsett1 navn/tall/felt rekordsett2.

I dette direktivet inngår tre objekter. Det andre objektet angir hva rekordsettet skal sorteres etter. Brukeren skriver her BOSTED.

Det første og tredje objektet er av klassen rekordsett. Det første leses inn fra hullkort, det siste genereres av SORTER-direktivet. Begge disse objekter er nye for maskinen og brukeren må gjøre deres egennavn kjent for maskinen i en deklarasjonssetning, f.eks.:

REKORDSETT PERSONREGSTR KOMMUNEREGSTR.

Rekordsett (og rekorder) står i en særstilling blant objekter ved at det er ikke nok med en deklarasjonssetning for å innføre et objekt. Det må også skrives en deskripsjonssetning for hvert rekordsett som deklarerer.

Deskripsjonssetningen forklarer hvilke elementærfelt rekorden eller rekordsettet består av. Hver gang en rekord eller et rekordsett deklarerer må en deskripsjonssetning også skrives (se avsnitt B-3.4).

I eksemplet består begge rekordsett av fem elementærfelt. Brukeren skriver en deskripsjonssetning for hvert rekordsett:

PERSONREGSTR NUMMER PERSONNAVN FØDSELSÅR FØDSELSSTED BOSTED.

KOMMUNEREGSTR NUMMER PERSONNAVN FØDSELSÅR FØDSELSSTED BOSTED.

(Legg merke til at i utskriften fra maskinen på papir vil første ordet i en deskripsjonssetning bare bli enkelt understreket. Dette skyldes at det første ordet er selv et egennavn til et objekt, ikke et direktivnavn.)

Brukeren kan nå skrive selve direktivsetningen som utfører sorteringen:

SORTER REKORDSETTET PERSONREGSTR ALFABETISK ETTER BOSTED OG SETT SVARET I REKORDSETTET KOMMUNEREGSTR.

Det som nå står igjen for brukeren er å få dataene i PERSONREGSTR innlest i maskinen fra hullkort, og å få det sorterte rekordsettet KOMMUNEREGSTR skrevet ut på papir.

Innlesing av rekordsett fra hullkort.

Dette foregår analogt med det som ble beskrevet ovenfor i eksempel 3. Forskjellen ligger i at formatet blir noe mer komplisert. Hvis det er mulig, er det vanlig å bruke bare ett hullkort for hver rekord. Hvis rekordene er store kan de gå over flere kort. En skråstrek inne i et format forteller maskinen at rekorden fortsetter på neste kort. Dette er mer utførlig beskrevet i kapittel B-4 "Innlesing av data fra hullkort".

Nedenfor følger et eksempel på hvordan de aktuelle rekorder kan leses inn.

I selve DATSY-programmet må det være en deklarasjonssetning som introduserer et formategennavn:

FORMAT PERSONFORM.

I programmets dataavdeling legges et formatkort:

*PERSONFORM = (I4, 1X, A10, 2X, I4, 1X, A8, 1X, A8)

Umiddelbart foran kortene som inneholder de enkelte rekorder legges et objektkort:

*PERSONREGSTR, R=8, PERSONFORM

Like etter objektkortet følger datakortene.

Det eneste som nå mangler er en utskrifttsetning som får det sorterte rekordsettet KOMMUNEREGSTR skrevet ut på papir:

PRINTREKSETT KOMMUNEREGSTR.

Det endelige program vil da se ut som følger:

FORMAT PERSONFORM.

REKORDSETT PERSONREGSTR KOMMUNEREGSTR.

NAVN PERSONNAVN BOSTED FØDSELSSTED.

TALL NUMMER FØDSELSÅR.

PERSONREGSTR NUMMER PERSONNAVN FØDSELSÅR FØDSELSSTED BOSTED.

KOMMUNEREGSTR NUMMER PERSONNAVN FØDSELSÅR FØDSELSSTED BOSTED.

SORTER REKORDSETTET PERSONREGSTR ALFABETISK ETTER BOSTED OG SETT SVARET I REKORDSETTET KOMMUNEREGSTR.

PRINTREKSETT KOMMUNEREGSTR.

SLUTTPROD.

*PERSONFORM = (I4, 1X, A10, I6, 1X, A8, 1X, A8)

*PERSONREGSTR, R=8, PERSONFORM

bbb1bGOGSTADbbbb1931bOSLObbbbOSLObbbb

2	GUNDERSON	1916	DRAMMEN	OSLO
3	ESPELAND	1942	DRØBAK	DRØBAK
4	SAUER	1937	ASKER	OSLO
5	SKILBREI	1924	ÅS	DRØBAK
6	LUND	1945	ASKIM	ASKIM
7	NALMO	1924	OSLO	ASKER
8	NANSEN	1907	OSLO	OSLO

Maskinen vil etter dette skrive ut på papir KOMMUNEREGSTR:

REKORDSETT KOMMUNEREGSTR

PERSON- FØDSELS- FØDSELS-
NUMMER NAVN ÅR STED BOSTED

bbb7bNALMObbbbbb1924bOSLObbbbASKERbbb

6	LUND	1925	ASKIM	ASKIM
3	ESPELAND	1924	DRØBAK	DRØBAK
5	SKILBREI	1924	ÅS	DRØBAK
1	GOGSTAD	1931	OSLO	OSLO
2	GUNDERSON	1916	DRAMMEN	OSLO
4	SAUER	1937	ASKER	OSLO
8	NANSEN	1907	OSLO	OSLO

A-2.6 Eksempel 6. Redigering av rekordsett

I dette eksemplet skal et rekordsett som befinner seg på en arkivtape redigeres, dvs. det skal bygges opp et nytt rekordsett med deler fra det opprinnelige. Dette nye rekordsettet skal skrives ut på papir. Brukeren vil underveis bli gjort kjent med begrepet FELT innen DATSY. Rekordsettet fra eksempel 5 vil bli benyttet som data.

Rekordsettet med egennavnet PERSONREGSTR ligger i DATSY-arkivet på en arkivtape med egennavnet PERSONTAPE. Rekordsettet består av fem elementærfelt som alle er kjent for maskinen som objekter av klassene NAVN og TALL. Objekttegnavnene er de samme som i eksempel 5. Nedenfor følger en oppbygging av et DATSY-program som utfører redigeringen.

I og med at PERSONREGSTR ligger på en arkivtape er det nødvendig med en arkivtapesetning.

ARKIVTAPE PERSONTAPE.

Det skal genereres et nytt rekordsett. Anta at brukeren velger å kalle det NAVNREGISTER. Dette gjøres kjent i en deklarasjonssetning:

REKORDSETT NAVNREGISTER.

I eksempel 5 ble det presisert at når et rekordsett deklarerer må det også beskrives. Anta at brukeren ønsker at NAVNREGISTER skal inneholde elementærfeltene PERSONNAVN, FØDSELSSTED og FØDSELSÅR. Han behøver ikke deklare disse elementærfeltene da de er kjent for maskinen på forhånd. NAVNREGISTER må imidlertid beskrives. Dette kan gjøres med følgende beskrivelsessetning:

NAVNREGISTER PERSONNAVN FØDSELSSTED FØDSELSÅR.

Deskripsjonen kan imidlertid bli utført på en alternativ måte som vist nedenfor.

Bak i denne manualen finner brukeren et redigeringsdirektiv:

REDREKORD rekord1/rekordsett1 navn/tall/felt rekord2/rekordsett2.

I vårt eksempel er første objekt til dette direktivet rekordsettet PERSONREGSTR. Det siste objektet er det nye rekordsettet som brukeren valgte å kalle NAVNREGISTER. Objekt nr. 2 angir hvilke elementærfelt det nye rekordsett skal inneholde. I dette tilfellet skal NAVNREGISTER inneholde tre elementærfelt. Objekt nr. 2 kan imidlertid bare inneholde ett egennavn.

Dette problem kan brukeren løse ved uformelt sagt å lage en "pakke" av de tre elementærfelt slik at pakken bare har ett egennavn. Han innfører først et objekt av klassen FELT med en deklarasjonssetning. Han skriver så en beskrivelsessetning som fører til at det nye objektet "består av" de tre opprinnelige objekter. Et objekt av klassen FELT består således av det eller de elementærfelt som beskrivelsessetningen tilsier.

Anta at brukeren i dette eksemplet velger å innføre egennavnet HJELPEPAKKE på sitt objekt av klassen FELT. Det gjøres med følgende deklarasjonssetning.

FELT HJELPEPAKKE.

Brukeren skriver videre følgende beskrivelsessetning:

HJELPEPAKKE PERSONNAVN FØDSELSÅR FØDSELSSTED.

Brukeren er nå klar til å utforme direktivsetningen.

REDREKORD REDIGERER REKORDSETTET PERSONREGSTR VED Å TA UT ELEMENTÆRFELTENE GITT VED HJELPEPAKKE OG SETTE I REKORDSETTET NAVNREGISTER.

I dette eksemplet var bruken av felt absolutt nødvendig fordi antall objekter i direktivsetningen var fastlagt. Objekter av klassen FELT kan også brukes som hjelpemiddel ved rekord- eller rekordsettdeskripsjoner. Tidligere i eksempel 6 ble det gitt en beskrivelsessetning for rekordsettet NAVNREGISTER. I og med at hjelpepakke er gjort kjent for maskinen vil en mer kortfattet måte å oppnå det samme være å skrive følgende beskrivelsessetning:

NAVNREGISTER HJELPEPAKKE.

Vi har nå kommet så langt at vi kan få maskinen til å skrive ut NAVNREGISTER på papir med følgende utskriftsetning:

PRINTREKSETT NAVNREGISTER.

Som vanlig avsluttes DATSY-programmet med en sluttsetning.

Det endelige DATSY-program vil da se ut som følger:

ARKIVTAPE PERSONTAPE.

REKORDSETT NAVNREGISTER.

FELT HJELPEPAKKE.

HJELPEPAKKE PERSONNAVN FØDSELSSTED FØDSELSÅR.

NAVNREGISTER HJELPEPAKKE.

REDREKORD REDIGERER REKORDSETTET PERSONREGSTR VED Å TA UT ELEMENTÆRFELTENE GITT VED HJELPEPAKKE OG SETTE DEM I REKORDSETTET NAVNREGISTER.

PRINTREKSETT NAVNREGISTER.

SLUTTPROD.

Maskinen vil etter dette gi følgende utskrift på papir:

REKORDSETT PERSONNAVN.

PERSON- FØDSELS-FØDSELS-
NAVN STED ÅR

GOGSTADbbbOSLObbb1931

GUNDERSON DRAMMEN 1916

ESPELAND DRØBAK 1942

SAUER ASKER 1937

SKILBREI ÅS 1924

LUND ASKIM 1945

NALMO OSLO 1924

NANSEN OSLO 1907

B. DETALJERT VEILEDNING I BRUK AV DATSY

B-1. Grunnelementer i en DATSY-tekst

I dette kapitlet behandles de formelle elementer som en DATSY-tekst er bygd opp av. De etterfølgende kapitler vil gå nærmere inn på innholdet i disse.

B-1.1 Tegn

For å skrive en DATSY-tekst brukes tegn som inndeles i tre typer:

- (i) bokstaver og siffer,
- (ii) punktum
- (iii) alle andre tegn.

Blant de siste hører tegnet "mellomrom".

B-1.2 Ord og ordtyper

Et ord består av bokstaver, siffer og bindestrek (-). Et ord avsluttes av ethvert tegn som ikke er bokstav eller siffer, og av ny linje. I et ord blir de inntil 12 første siffer eller bokstaver registrert. Hvis det er flere tegn i et ord blir disse oversett.

Alle ord kan deles i to grupper, systemord og brukerord. Systemordene kan videre deles/være av tre typer:

- (i) fellesnavn
- (ii) direktivnavn
- (iii) hjelpeord

Brukerordene er enten objektegennavn eller tomme ord.

Ordtyper	
systemord	brukerord
fellesnavn	objektegennavn
direktivnavn	tomme ord
hjelpeord	

Fig. 2.

Systemordene er gitt spesiell permanent betydning for maskinprogrammet. Bruken av systemord er derfor underlagt spesifikke regler. Brukerord er ord som brukeren selv introduserer. I alle eksempler på DATSY-setninger er alle systemord skrevet med dobbel understrekning og ikke-tomme brukerord med enkel understrekning.

Et fellesnavn er navnet på en klasse av objekter. Eksempler på systemord av typen fellesnavn er: MATRISE, VEKTOR, PARAMETER, NAVNORD, REKORD, REKORDSETT, NAVN, TALL, FELT, FORMAT. Det er altså et fellesnavn for hver objektklasse.

Direktivnavn er navn på operasjoner som kan utføres på objektene. Ved utgangen av 1973 finnes det ca. 120 slike direktivnavn. I del C i denne håndboken er det gitt en oversikt og beskrivelse av alle eksisterende direktiver ordnet etter deres respektive funksjon og direktivnavn. Eksempler på direktivnavn er: ADDER, MATMULT, SORTER, BYTTKOL.

Hjelpeord er navn på spesielle operasjoner som ikke refererer til brukeren objekter, men til DATSY-språkteksten selv, til magnetbånd, eller til katalogen. Eksempler på systemord av typen hjelpeord er: SUBSTITUER, INNFØR, VELG, TØM, SKRIV, ARKIVTAPE, OPPLYSOBJEKT.

Objektnavn er fritt valgte navn som brukeren setter på de objekter han arbeider med.

Tomme ord er alle ord som ikke faller inn under de typer som er nevnt ovenfor. Tomme ord blir ignorert av maskinen. De kan benyttes av brukeren for å gjøre DATSY-programmene mer selvdokumenterende.

B-1.3 Linje

En linje inneholder 80 tegn. En linje kan være "tom" for såvidt som alle 80 tegn er "mellomrom".

B-1.4 Setning

En setning består av ord. En setning kan bare begynne med ny linje. En setning avsluttes av, og kan bare bli avsluttet av, punktum. Det følger at en setning kan gå over flere linjer, men at det bare kan være en setning på en linje. Det er en vanlig nybegynnerfeil å glemme punktum.

B-1.5 Kommentar

Alle tegn på en linje etter et punktum behandles som en kommentar og blir ikke tolket.

B-2. Objekter, objekttegnavn, objektklasser

Objektbegrepet er et meget sentralt begrep innen DATSY. God forståelse av dette er nødvendig for å skrive DATSY. Brukeren må skille klart mellom selve objektet, det egennavn objektet har og den klasse objektet tilhører. Det vil også være nyttig å skille objektene i to grupper; dataobjekter og hjelpeobjekter.

Alle objekter må ha et egennavn. DATSY identifiserer objektet ved dets egennavn. Hvis brukeren vil innføre et nytt objekt, må han gi det et egennavn. DATSY gir informasjon om at et nytt objekt med dette egennavnet blir innført ved at brukeren skriver en deklarasjonssetning i DATSY-programmet (se B-3.2). I denne deklarasjonssetningen innføres det fritt valgte egennavn som egennavn for et objekt av en bestemt klasse.

B-2.1 Dataobjekter

For brukeren vil dette være den viktigste gruppe objekter. Brukerens data er objekter. Hvis brukeren f.eks. skal arbeide med en matrise vil innholdet av matrisen være et dataobjekt. Alle objekter må tilhøre en objektklasse. Aktuelle klasser av dataobjekter er PARAMETER, VEKTOR, MATRISE, NAVNORD, LISTE, SUBLISTE, REKORD og REKORDETT.

Et bestemt tall vil være et objekt av klassen PARAMETER.

En vanlig tallvektor vil være et objekt av klassen VEKTOR.

En vanlig tallmatrise vil være et objekt av klassen MATRISE.

Et bestemt navn vil være et objekt av klassen NAVNORD.

En vanlig liste med navn vil være et objekt av klassen LISTE.

Et objekt av klassen SUBLISTE angir for hver av to lister hvilke elementer som er felles for begge listene.

En ordnet sekvens av numeriske og alfanumeriske data vil være et objekt av klassen REKORD. Hver rekord vil ha en bestemt struktur, dvs. hvert objekt av klassen REKORD må deskriberes som en sekvens av hjelpeobjekter av klassene NAVN og TALL. Dette er eksemplifisert i avsnittene A-2.3 og A-2.6.

En mengde rekorder som alle har samme struktur, dvs. samme inndeling i TALL og NAVN, vil være et objekt av klassen REKORDETT.

B-2.2 Hjelpeobjekter

Hjelpeobjekter er objekter som ikke direkte er brukerens data. For eksempel betraktes de arkivtaper som er kjent for DATSY som hjelpeobjekter. Analogt med dataobjektene må også alle hjelpeobjekter ha et egennavn og tilhøre en klasse. Navn settes på hjelpeobjekter som for dataobjekter ved hjelp av deklarasjonssetninger (se B-3.2). Bruken av magnetbånd representerer imidlertid et unntak fra dette som blir forklart i avsnitt B-2.5 "Lagringsobjekter".

Det er i dag ni klasser av hjelpeobjekter. Disse hører naturlig inn i tre grupper. Følgende fig. gir en oversikt over disse.

Hjelpeobjekter

Grupper av hjelpeobjektklasser	Objektklasser
Innlesingsobjekter	FORMAT
Struktureringsobjekter	NAVN TALL FELT
Lagringsobjekter	ARKIVTAPE NYTAPE EKSTERNTAPE FORTRANTAPE COBOLTAPE

Fig. 3.

B-2.3 Innlesingsobjekter

Hjelpeobjekter av klassen FORMAT er bare aktuelle i forbindelse med innlesing av data fra hullkort. Objekter av klassen FORMAT er "leseinstrukser" som stort sett likner FORTRAN-formater. Kapittel B-4, "Innføring av data fra hullkort", forklarer dette nærmere.

Objekter i denne klassen må gis egennavn og deklarerer på vanlig måte.

B-2.4 Struktureringsobjekter

Dette er en gruppe hjelpeobjekter som bare er aktuelle i forbindelse med bruk av rekorder og rekordsett. Som alle andre objekter må også objekter av klassene NAVN, TALL og FELT deklarerer.

a. NAVN

I rekorder og rekordsett kalles elementærfelt med alfanumerisk informasjon for objekter av klassen NAVN. Legg merke til at det er selve elementærfeltet og ikke dets innhold som er objektet i dette tilfellet.

b. TALL

I rekorder og rekordsett kalles elementærfelt med numerisk informasjon for objekter av klassen TALL. Legg merke til at det er selve elementærfeltet og ikke dets innhold som er objekter i dette tilfellet.

c. FELT

I rekorder og rekordsett kalles en kombinasjon av elementærfelt for et objekt av klassen FELT. Analogt med objekter av klassene NAVN og TALL er det selve elementærfeltkombinasjonen som er objektet, ikke elementærfeltenes innhold. Som alle andre objekter må også objekter av klassene FELT deklarerer. FELTbegrepet er kanskje noe vanskelig å forstå umiddelbart. I avsnitt A-2.6 er begrepet nærmere forklart ved et eksempel.

Sammenhengen mellom dataobjektene og struktureringsobjektene kan illustreres i en fig.

Struktureringsobjekter

Struktureringsobjekter	<u>tall</u>	<u>navn</u>	<u>felt</u>
dataobjekter	<u>parameter</u> <u>vektor</u> <u>matrise</u>	<u>navnord</u> <u>liste</u> -	<u>rekord</u> <u>rekordsett</u> -

Fig. 4.

B-2.5 Lagringsobjekter

a. ARKIVTAPE

Alle dataobjekter som er kjent for DATSY lagres på magnetbånd. Disse magnetbåndene betraktes som objekter av klassen ARKIVTAPE. Opplysninger om de enkelte dataobjektene er ført inn i katalogen. Likeledes er opplysninger om de forskjellige arkivtapene ført inn i katalogen. Til hver arkivtape hører to labelrekorder. Disse labelrekordene er forskjellige for alle arkivtaper, og funksjonerer som kjennetegn på arkivtapene. Labelrekordene er skrevet både på arkivtapen og i katalogen. Når en arkivtape skal brukes i et DATSY-program, må brukeren skrive en arkivtapesetning. Det blir da kontrollert at labelrekordene i katalogen og på det magnetbåndet som er montert på tapestasjonen, stemmer overens. Arkivtapesetningen er nærmere beskrevet i avsnitt B-3.6.

b. NYTAPE

Magnetbånd som er tomme og som ennå ikke er innført i katalogen oppfattes som objekter av klassen NYTAPE. Som alle andre objekter må også objekter av klassen NYTAPE deklarereres første gang de inngår i en DATSY-tekst. Selve innføringen skjer ved hjelp av en INNFØR-setning som er beskrevet i avsnitt B-3.6. Magnetbåndet går da over til å være et objekt av klassen ARKIVTAPE og katalogen blir ajourført.

Det er mulig å skrive på et tomt magnetbånd i det samme DATSY-program som innførte båndet.

c. EKSTERNTAPE

Det kan tenkes mer enn en DATSY-sentral. Magnetbånd som inneholder dataobjekter som er skrevet ved en annen DATSY-sentral, betraktes som objekter av klassen EKSTERNTAPE. Ved brukerens DATSY-sentral er slike eksterntaper ikke ført inn i katalogen og må først deklarereres på vanlig måte, dvs. som EKSTERNTAPE. Deretter følger en INNFØR-setning som er beskrevet i avsnitt B-3.6. Når INNFØR-setningen er eksekvert går magnetbåndet over til å være et objekt av klassen ARKIVTAPE mens dets egennavn forblir uforandret.

d. FREMMETAPE (FORTRANTAPE OG COBOLTAPE)

Med fremmedtape menes magnetbånd som er generert av andre program enn DATSY. Innholdet av en fremmedtape vil vanligvis være skrevet i et annet format enn innholdet av DATSY-genererte magnetbånd.

Fremmedtaper som er skrevet i standard FORTRAN-format kan betraktes som objekter av klassen FORTRANTAPE. Fremmedtaper som er skrevet i standard COBOL-format kan betraktes som objekter av klassen COBOLTAPE.

Innholdet av fremmedtaper av klassene FORTRANTAPE og COBOLTAPE kan innføres i DATSY hvis følgende forutsetninger er til stede:

- (i) Dataene på magnetbåndet må kunne identifiseres som et eller flere DATSY-objekter.
- (ii) Dataobjektene må være lagret fysisk etter hverandre på båndet.
- (iii) Hvert dataobjekt begynner med en ny blokk. (Med blokk menes her den fysiske lagringsenheten som ble brukt da båndet ble skrevet.)
- (iv) Dataobjektene på fremmedtapen kan ikke være av klassene PARAMETER, NAVNORD eller FORMAT.

Formaterte FORTRAN-taper vil vanligvis ikke tilfredsstille krav (iii). En COBOL-file kan inneholde rekorder av forskjellig type. Hvis hver rekord er kompatibel med DATSY kan vi oppfatte innholdet av en slik file som flere DATSY-objekter (rekordsett) blandet sammen. I så fall er imidlertid ikke krav (ii) tilfredsstilt.

I Statistisk Sentralbyrå vil det være vanlig å innføre data fra fremmedtaper. Ofte vil fremmedtapene være FORTRAN-taper med bare ett rekordsett. Forutsetningene er da til stede for at dataobjektene på fremmedtape kan innføres i DATSY.

Som alle andre objekter må også fremmedtaper gis klasse (FORTRANTAPE eller COBOLTAPE) og egen navn i en vanlig deklarasjonssetning. Et eksempel på en slik deklarasjonssetning er:

FORTRANTAPE 8807.

Innholdet av en fremmedtape kan så betraktes som en del av den dataavdeling som kommer etter DATSY-teksten (se B-4). Alle dataobjekter på fremmedtapen som enten skal brukes i det aktuelle DATSY-program eller som skal overføres til arkivtape, må defor ha et objektkort. (Se B-4.4.) Dataobjektene kan da refereres til på nøyaktig samme måte som objekter på hullkort. Dette betyr blant annet at objektene kan skrives ut på arkivtaper og på den måten bli registrert i katalogen.

De dataobjektene på en fremmedtape som er blitt referert til ved objektkort kan imidlertid også bringes inn i arkivet samlet ved at fremmedtapen selv innføres i arkivet som arkivtape. Dette gjøres ved hjelp av en INNFØR-setning som er beskrevet i avsnitt B-3.6. Samtidig blir katalogen automatisk ajourført.

B-3. Setningstyper

En DATSY-tekst er bygd opp av setninger. I dette kapitlet blir de forskjellige typer setninger beskrevet og forklart.

B-3.1 Direktivsetninger

En bruker av DATSY har vanligvis til hensikt å få utført visse arbeidsoperasjoner på sine dataobjekter. Eksempler på slike arbeidsoperasjoner er addisjon og multiplikasjon av matriser og redigering av rekordsett. Slike arbeidsoperasjoner er i DATSY-terminologi betegnet som direktiver. Brukeren benytter direktivene ved å skrive direktivsetninger. Direktivsetninger er således brukerens instruks til maskinen om de arbeidsoperasjoner som ønskes utført.

(Betegnelsen direktiv brukes også om den samling av FORTRAN-instruksjoner som utfører en arbeidsoperasjon. Et direktiv blir i den tolking et delprogram av DATSY. Direktivsetningen får maskinen til å benytte det aktuelle delprogram til å utføre en arbeidsoperasjon.)

De enkelte direktiver er gitt egennavn som er systemord. Disse egennavn er valgt slik at de til en viss grad er selvforklarende. F.eks. er den arbeidsoperasjon som utfører en addisjon av to matriser betegnet ADDER.

En direktivsetning begynner alltid med et direktivegennavn og avsluttes med punktum. Direktivegennavnet etterfølges av et bestemt antall brukerord, dvs. objektetegnavn. Innimellom disse kan brukeren fritt fylle inn tomme ord for å lette lesingen av direktivsetningen. Dette er vist i et eksempel i avsnitt A-2.1.

Del C i denne håndboken inneholder en liste over de ca. 120 direktiver som er innført i DATSY. Samtidig gis en beskrivelse om hvordan de respektive direktivsetninger skal konstrueres, dvs. hvor mange objektetegnavn direktivsetningen skal inneholde, hvilke klasser disse må tilhøre og hvilken rekkefølge de må forekomme i.

B-3.2 Deklarasjonssetninger

Brukeren vil ofte ha behov for å innføre nye brukerord. Deklarasjonssetninger benyttes til å gjøre egennavnet til et nytt objekt kjent for systemet som brukerord. Samtidig plasseres objektet i en bestemt klasse.

En deklarasjonssetning begynner alltid med et fellesnavn og avsluttes med punktum. Etter fellesnavnet følger ett eller flere ord som brukeren velger fritt, bortsett fra at systemord ikke kan benyttes og heller ikke brukerord som er deklartert annet sted i samme program. Virkningen av dette er at disse heretter vil bli oppfattet som egennavn for objekter av den klasse som deklarasjonssetningens fellesnavn tilsier. Deklarasjonssetninger kan ikke utfylles med tomme ord for å gjøre setningene mer lettlest.

Bare de tolv første tegn i et ord vil bli registrert som et nytt brukerord. Ytterligere tegn ignoreres. Hvis flere nye brukerord skal deklarerer i en og samme deklarasjonssetning, benyttes mellomrom eller komma for å skille dem.

Anta at et objekt, som ikke er et lagringsobjekt, med egennavn og klasse skal inngå i en DATSY-tekst for første gang. Objektet er deklarerert på vanlig måte. Hvis brukeren ønsker, kan han få dette objektet arkivert og katalogisert til senere kjøring. Dette gjøres med en SKRIV-setning (se B-3.6). Ved senere kjøring er det da ikke nødvendig å gjenta deklarasjonen av dette objektet. Objektet med egennavn og klasse kan betraktes og behandles som deklarerert i senere kjøring.

Objekter av klassene NYTAPE, EKSTERNTAPE og ARKIVTAPE blir automatisk bevart til senere kjøring. Objekter av klassene NYTAPE og EKSTERNTAPE deklarerer bare første gang de innføres. De betraktes deretter som objekter av klassen ARKIVTAPE. Ved alle senere kjøring hvor disse magnetbånd benyttes, må DATSY-teksten inneholde en arkivtapesetning (se B-3.3).

Eksempler på deklarasjonssetninger kan være:

MATRISE MAT1 MAT2 MAT3

REKORDSETT PERSONREGSTR.

NYTAPE 8764.

Forøvrig er deklarasjonssetninger og bruken av dem illustrert ved eksempler i avsnitt A-2.3.

B-3.3 Arkivtapesetning

Arkivtapesetningen benyttes hver gang brukerens data ligger på en arkivtape i DATSY-arkivet, og bare da. Arkivtapesetningen har mye til felles med deklarasjonssetningene, men avviker fra disse på det punkt at arkivtapesetningen må gjentas i hver eneste DATSY-tekst som anvender dataobjekter som er skrevet på det aktuelle magnetbånd. Magnetbåndet påkalles ved hjelp av arkivtapesetningen. Samtidig kontrolleres det at de to labelrekordene som er skrevet på det monterte magnetbånd og i katalogen stemmer overens (jfr. B-2.5).

B-3.4 Deskripsjonssetninger

Objekter av klassene REKORD, REKORDSETT og FELT har en struktur som må angis utover selve klassetilhørigheten. I tillegg til at de må deklarerer må de også deskriberes. De er sammensatt av ett eller flere elementærfelt. Hvert elementærfelt oppfattes som et eget objekt som kan være av klassene NAVN eller TALL. En deskripsjonssetning sier noe om den kombinasjon av elementærfelt som utgjør selve rekorden, rekordsettet eller feltet. Deskripsjonssetningen inneholder først egennavnet til den aktuelle rekord, rekordsett eller felt. Deretter følger egennavnene til de aktuelle elementærfelt i den rekkefølge de forekommer i objektet som deskriberes. De forskjellige egennavnene skilles med mellomrom eller komma, og deskripsjonssetningen avsluttes som vanlig med punktum. Et eksempel på en deskripsjonssetning kan være:

NAVREGISTER PERSONNAVN FØDSELSSTED FØDSELSÅR.

NAVREGISTER er her ment å være egennavnet til et rekordsett. De øvrige ord er egennavn til elementærfeltene i hver rekord i rekordsettet, dvs. egennavn til objekter av klassene NAVN og FELT.

Bruk av deskripsjonssetninger er mer eksemplifisert i to eksempler i avsnittene A-2.5 og A-2.6.

Objekter av klassene FELT, REKORD og REKORDSETT må som nevnt ovenfor alltid deskriberes. Det kan imidlertid også forekomme deskripsjon av arkivtape.

Grunnen til det er at det kan være ønskelig å forandre midlertidig navnet på objekter som befinner seg på en arkivtape. Hvis alle eller mange av objekttegennavnene skal forandres, gjøres det ved å deskribere arkivtapen. Dette kan også gjøres ved å skrive en VELG-setning for hvert objekt (se B-3.6).

Anta at brukeren har en arkivtape som inneholder m objekter, (ikke medregnet de to labelrekordene). Brukeren ønsker imidlertid å forandre egennavnene på disse objektene. Dette gjøres da ved at arkivtapen beskrives. En slik arkivtapedeskrripsjon består av arkivtapens egennavn etterfulgt av m ord som er de midlertidige navnene brukeren ønsker å benytte. Arkivtapedeskrripsjoner kan se ut som følger:

ARKIVTAPENAVN ORD1 ORD2 ORDM.

De m objektene på arkivtapen blir da imidlertid omdøpt til brukerens midlertidige navn i overensstemmelse med den rekkefølge objektene befinner seg i på magnetbåndet. De midlertidige objekt-egennavnene kan, men behøver ikke, deklarerer. Hvis de deklarerer, må objektklassene stemme overens med beskrivelsen av objektene i katalogen. Hvis det er ønskelig at noen av objektene beholder sine egentlige egennavn må disse tas med på riktig plass i arkivtapedeskrripsjonen.

Etter at arkivtapen er beskribert er det i det samme DATSY-program bare mulig å referere til objektene ved deres nye egennavn. En arkivtapedeskrripsjon er midlertidig i den forstand at den gjelder bare i det DATSY-program som inneholdt beskrripsjonen. Arkivtapedeskrripsjoner medfører altså ingen endring i katalogen.

B-3.5 Hjelpesetninger, substituering

Substitueretningen gir brukeren mulighet for å endre en foreliggende DATSY-tekst ved å bytte ut forekommende ord med andre. En substitueretning består av tre ord:

SUBSTITUER ORD1 ORD2

SUBSTITUER som alltid står først i setningen er et systemord. De øvrige to ord kan være systemord, brukerord eller tomme ord. Virkningen av en substitueretning er at alle forekomster av ordet ORD1 i setningene som følger byttes ut med ordet ORD2.

Mer enn en substitueretning kan forekomme i samme tekst og det må derfor klargjøres hvordan flere slike virker sammen. De føldende regler gjelder:

- (i) Substitusjonene angitt ved én setning utføres før noen av substitusjonene angitt ved neste substitueretning.
- (ii) En substitueretning virker ikke på andre substitueretninger.
- (iii) En substitueretning oppheves hvis det senere i teksten kommer en annen substitueretning der ORD1 er uforandret, mens ORD2 er erstattet av et annet.

Virkningen av setningen

SUBSTITUER ORD1 ORD2.

vil således bli opphevet for resten av DATSY-teksten etter setningen:

SUBSTITUER ORD1 ORD3.

B-3.6 Hjelpesetninger for magnetbånd

a. INNFØR-setningen

En INNFØR-setning brukes til å registrere nye magnetbånd i katalogen, dvs. innføre dem i katalogen slik at de blir objekter av klassen ARKIVTAPE.

En INNFØR-setning består alltid av ett systemord og tre brukerord og avsluttes av punktum. (I tillegg kan brukeren fylle inn tomme ord etter ønske.) Det første ordet er alltid systemordet INNFØR. Det første av brukerordene er egennavnet til det magnetbåndet som skal innføres. Magnetbåndet må som alle objekter være deklarerert med egennavn og klasse. Magnetbåndet kan være av klassene NYTAPE, EKSTERNTAPE, FORTRANTAPE eller COBOLTAPE. De siste to brukerordene i INNFØR-setningen er egennavn til to rekorder, som funksjonerer som labelrekorder. Disse to rekordene som må være deklarerert og beskribert føres inn

i katalogen sammen med magnetbåndets egennavn og informasjon om hva båndet inneholder. Dessuten skrives de ut på selve magnetbåndet. Ved senere kjøring betraktes magnetbåndet som et objekt av klassen ARKIVTAPE. Ved slik senere bruk av "arkivtapen" blir det kontrollert at labelrekordene på selve magnetbåndet er identisk med labelrekordene i katalogen. Legg merke til at det er mulig å skrive på arkivtapen i samme DATSY-program som den ble innført.

Objektklassene NYTAPE, EKSTERNTAPE, FORTRANTAPE og COBOLTAPE er beskrevet i avsnitt B-2.5.

Da eksterntapen er generert av et annet DATSY-system, inneholder de på samme måte som alle arkivtaper en fullstendig beskrivelse av sitt innhold i en egen file på magnetbåndet etter filen med objektene. Når INNFØR-setningen eksekveres bringes denne beskrivelsen av eksterntapens dataobjekter inn i katalogen.

Hvis det er fremmedtaper som blir innført, dvs. FORTRAN-taper eller COBOL-taper, vil eksekveringen av INNFØR-setningen føre til at de to første blokkene på magnetbåndet blir ført inn i katalogen som label-rekorder. Dette gjøres uavhengig av om magnetbåndet har label-blokker eller label-files. Label-rekordene blir å betrakte som objekter og gis de to egennavnene som er angitt sist i INNFØR-setningen.

En INNFØR-setning vil se ut som følger:

INNFØR TAPE REK1 REK2.

b. VELG-setningen

VELG-setningen gir brukeren mulighet til midlertidig å forandre navn på et enkelt objekt på en arkivtape. En VELG-setning består av ett systemord, to brukerord og minst ett tomt ord. Det første ordet er alltid systemordet VELG. Så følger egennavnet til den aktuelle arkivtape og så egennavnet til det aktuelle objekt, som må være et objekt på arkivtapen. (Mellom og foran disse to brukerordene kan det stå et vilkårlig antall tomme ord.) Etter objektets egennavn må det stå ett ord som ikke behøver å være deklart. Dette blir det nye, midlertidige navn på objektet. Setningen avsluttes av punktum. VELG-setningen kan da se ut som følger:

VELG PÅ TAPE OBJEKTET MED EGENNAVNEN OBJEKTNAVN NYTTNAVN.

Virkningen av denne setningen er at objektet midlertidig har fått et nytt egennavn, nemlig NYTTNAVN. Det presiseres at denne forandringen bare gjelder i det DATSY-program hvor VELG-setningen inngikk. Katalogen forandres ikke.

En DATSY-tekst kan inneholde flere VELG-setninger. En arkivtapedeskrripsjon har samme virkning som en sekvens av VELG-setninger, én VELG-setning for hvert av arkivtapens objekter. Arkivtapedeskrripsjoner "utføres" før VELG-setninger. Deskrripsjoner av arkivtape er beskrevet i avsnitt B-3.4. Se for øvrig også kapittel B-7 som gir en samlet oversikt over framgangsmåter når forskjellige objekter har samme egennavn.

c. TØM-setningen

TØM-setningen er motivert ut fra skriving på arkivtape. Det vil framgå av avsnittet nedenfor om SKRIV-setningen at det er en betingelse for å skrive på en arkivtape at den er tom for dataobjekter. Med det menes at katalogen må ha registrert arkivtapen som tom. (Om dataobjekter fysisk befinner seg på magnetbåndet er uvesentlig.)

Alle dataobjekter som skal skrives på en arkivtape, må skrives i en og samme SKRIV-setning. Brukeren vil derfor stadig ha behov for å "tømme" arkivtaper for så å kunne skrive på dem igjen.

Av hensyn til datasikkerheten er det ikke mulig å utslette dataobjektene på en arkivtape i ett DATSY-program. Den aktuelle framgangsmåte er imidlertid som følger. I et DATSY-program skriver brukeren en TØM-setning. Denne TØM-setning fører til at beskrivelsen i katalogen av dataobjektene i den aktuelle arkivtape blir utslettet. Men dataobjektene på arkivtapen blir ikke ødelagt. Arkivtapen blir faktisk overhodet ikke berørt av TØM-setningen og behøver derfor heller ikke deklarerer. DATSY oppfatter nå arkivtapen som en tom arkivtape. Det er av sikkerhetsmessige grunner imidlertid fortsatt ikke mulig å skrive på denne nye "tomme" arkivtapen i det samme DATSY-program som tømte magnetbåndet. I et annet DATSY-program er det mulig å skrive på denne tomme arkivtapen. Og det er først på dette tidspunkt at

de opprinnelige dataobjektene fysisk blir utslettet. Det følger av dette at hvis TØM-setningen har tømt et magnetbånd ved en feiltakelse, fjernet katalogbeskrivelsen av magnetbåndet, er det mulig å føre dataobjektene inn i systemet igjen ved å betrakte magnetbåndet som en eksterntape. Denne framgangsmåte kan virke noe tungvint, men sikkerheten økes.

En TØM-setning begynner alltid med systemordet TØM. Deretter følger tre brukerord som kan være omgitt av et vilkårlig antall tomme ord. Setningen avsluttes med punktum. Det første brukerordet er egennavnet til den arkivtape som skal tømmes. De andre to brukerordene er egennavnene til arkivtapens label-rekorder. TØM-setningen kan se ut som følger:

TØM TAPE SOM HAR LABELENE REK1 OG REK2.

d. Skrivsetningen

Skrivsetningen gjør det mulig å få dataobjekter overført til arkivtaper for lagring. Arkivtapen må være tom og derfor må alle dataobjekter som skal skrives på arkivtapen, overføres i en og samme skrivsetning. Skrivsetningen medfører også at en fullstendig beskrivelse av alle objektene (lagringsobjektet og dataobjektene) i setningen skrives ut på den aktuelle arkivtape og i katalogen.

Alle objekter som inngår i beskrripsjonen av et av de n objektene i skrivsetningen kommer også med i beskrivelsen av arkivtapen. Formater som rekord og rekordsett kan ha knyttet til seg som atributt blir også tatt med. Arkivtapen vil derfor vanligvis "inneholde" flere objekter enn de n som er angitt i skrivsetningen.

Antall objekter som blir fysisk skrevet ut på arkivtapen kan imidlertid være mindre. Objekter fra klassene NAVNORD, PARAMETER, REKORD og FORMAT blir ikke skrevet ut på arkivtapen. For disse objektene blir verdiene lagt inn i katalogen. Objekter fra klassene TALL, NAVN og FELT har ingen verdi. Slike objekter blir imidlertid tatt med i beskrivelsen av arkivtapen hvis de forekommer i skrivsetningen.

Skrivsetningen begynner alltid med systemordet SKRIV. I tillegg til et vilkårlig antall tomme ord inneholder den minst to brukerord. Det første av brukerordene er egennavnet til den aktuelle arkivtape. De øvrige brukerord er egennavnene til dataobjektene som skal skrives ut. Setningen avsluttes som vanlig av punktum.

En skrivsetning kan da se ut som følger:

SKRIV PÅ ARKIVTAPEEGENNAVN DATAOBJEKTENE OBJ1EGENNAVN OG OBJ2EGENNAVN OG ... OBJNEGENNAVN.

e. Arkivtapesetningen

Arkivtapesetningen er utførlig beskrevet i avsnitt B-3.3. Den er bare nevnt her fordi den også naturlig hører inn under dette avsnittet om hjelpesetninger for magnetbånd.

B-3.7 Sluttsetninger

En DATSY-tekst må alltid avsluttes av en sluttsetning. Sluttsetningen består av bare ett ord og punktum. Ordet er enten SLUTTKONT eller SLUTTPROD.

(i) SLUTTKONT.

Denne setning angir at DATSY-teksten er avsluttet og hensikten med kjøringen er å kontrollere denne teksten.

(ii) SLUTTPROD.

Denne setningen angir at DATSY-teksten er avsluttet og at hensikten med kjøringen er å få utført de operasjoner som er angitt i teksten.

B-3.8 Tomme setninger

En tom setning er en setning som begynner med et tomt ord. Hensikten med å bruke tomme setninger er å lette senere lesing av teksten.

B-4. Innføring av data fra hullkort

Anta at nye data skal bringes inn i systemet. Den naturlige måten å gjøre dette på er å punche data på hullkort (evt. skriving på terminal). I tillegg må DATSY få vite hva slags objekter dataene utgjør, og det må bli gitt en instruks om hvordan datakortene skal leses og tolkes. Samtidig må data være punchet slik at de blir lest riktig når systemet følger sin leseinstruks.

Sentralt i innføringen av data fra hullkort står begrepet format. Et format kan sies å være den "måte" maskinen leser datakortene på. Maskinen leser datakortene i overensstemmelse med et gitt format - en gitt leseinstruks. I DATSY har FORMAT blitt innført som en egen klasse av hjelpeobjekter (se B-2.2). Alle objekter må ha egennavn. Derfor må alle formater som en bruker vil innføre gis et formategennavn. Det gjøres som vanlig ved deklarasjonssetninger (se B-3.2). Det er således viktig for forståelsen å skille mellom formatverdi, som er selve leseinstruksen, og formategennavn, som er det vilkårlige egennavn som settes på leseinstruksen.

Ved innlesing av et dataobjekt fra hullkort må brukeren foruten selve datakortene i alminnelighet punche to kort som legges sammen med datakortene etter sluttsetningen i DATSY-teksten. Det ene kortet angår selve dataobjektet, mens det andre kortet angår hjelpeobjektet - formatet. Det er derfor naturlig å gi dem ulike betegnelser. De blir kalt objektkort og formatkort. En fellesbetegnelse for de to er identifikatorkort. Begge typer identifikatorkort må ha en stjerne som spesialtegn punchet i første kolonne. Formålet med identifikatorkort er å identifisere for systemet de ulike dataobjekter og formater i kortstrømmen i DATSY-programmets dataavdeling.

NB. Innholdet av identifikatorkort er ikke setninger i DATSY-forstand og skal derfor heller ikke avsluttes av punktum.

B-4.1 Formatkort

Formålet med formatkort er å tilordne et formategennavn til et format. Formatkortet kan legges hvor som helst i dataavdelingen i DATSY-programmet, bare ikke innimellom de kort som utgjør et objekt. Formatkort skal punches med '*' i første kolonne. Et eksempel på et formatkort er

```
*FORM = (5X, 2(F7.2))
```

Her er '(5X, 2(F7.2))' selve formatet, mens 'FORM' er dette formatets egennavn. Nærmere forklaring av formatkortets innhold for de forskjellige klasser av dataobjekter kommer i de respektive avsnitt i dette kapitlet.

Det vil være typisk at samme datastruktur skal leses inn ved flere anledninger. F.eks. vil det være aktuelt å lese inn data for nasjonalregnskapet fra år til år. Det kan derfor være aktuelt å oppbevare visse formater i arkivet på liknende måte som andre objekter. Ved bruk av arkiverte formater er det ikke nødvendig verken med noe formatkort eller noen ny deklarasjon av det arkiverte formatet. En utskrift av katalogen vil vise hvilke arkiverte formater som er tilgjengelig til enhver tid og til hvilke datastrukturer de er tenkt benyttet.

For brukere med kjennskap til FORTRAN kan nevnes noen tilleggsopplysninger. De formater som aksepteres av DATSY er nær identiske med FORTRAN's (om nødvendig se en FORTRAN-manual). DATSY aksepterer følgende formatbokstaver

H, A, D, F, I, X, E, O.

De mest aktuelle format-bokstaver ved input av data er A, F, I, X. De viktigste avvikene fra FORTRAN-formater kan samles i fire punkter:

- (i) DATSY-formatene kan ha inntil 3 parentesnivåer.
- (ii) Foran en parentes inne i et format må det være angitt en repetisjonsfaktor. F1 = (5X, (F7.2)) blir altså ikke akseptert, mens F2 = (5X, 1(F7.2)) blir akseptert.
- (iii) DATSY aksepterer for eks. (A16) og oversetter det til (2A6, A4).

B-4.2 Objektkort

Det skal skrives et objektkort for hvert objekt som skal leses inn. Et objektkort kan betraktes som et slags "følgebrev" til et dataobjekt. Objektkortet må alltid ligge umiddelbart foran de datakort som utgjør dataobjektet. Rekkefølgen av dataobjekter i et DATSY-program's dataavdeling er vilkårlig, men hvert objekt må foreligge på et separat kortsett som umiddelbart etterfølger objektkort. Et datakort kan således ikke inneholde data som tilhører to forskjellige objekter.

Innholdet av objektkort har et eller flere ledd. Disse ledd vil vanligvis angi dataobjektets egennavn, dataobjektets dimensjoner og det aktuelle formats egennavn. De enkelte ledd skal være atskilt med komma. Et ledd kan bestå av to subledd i en bestemt rekkefølge, atskilt med likhetstegn. Første ledd eller subledd på objektkortet til et dataobjekt er dataobjektetegnavnet. De andre leddene kan komme i vilkårlig rekkefølge. Noen av leddene er obligatoriske. Analogt med hva som gjelder for formatkort skal alle objektkort ha 'x' i første kolonne. Et ledd som består av en vilkårlig rekke tegn innesluttet av apostrofer blir oppfattet som kommentar og kan fritt skrives hvor som helst på kortet.

Hvis det er nødvendig kan objektkortet gå over flere kort. Semikolon brukes som fortsettelsestegn og betyr at neste kort er en fortsettelse av det kort hvor semikolonet stod. Hvis semikolon står mellom to apostrofer blir det ignorert. Alt som står til høyre for et gyldig semikolon blir ignorert idet lesingen fortsetter på neste kort. Hvis semikolon brukes etter et ledd slik at neste ledd begynner på neste kort, er det ikke nødvendig med komma mellom de to leddene. Mellomrom ignoreres overalt unntatt mellom to apostrofer. Umiddelbart etter objektkortet følger hullkort med de dataene som utgjør objektet.

B-4.3 Innlesing av parameter

Parametre leses inn i såkalt "fritt format". Det vil si at det ikke er nødvendig med noe format. Formatkortet kan derfor sløyfes. Selve parameterverdien angis på objektkortet slik at det ved innlesing av parameter heller ikke vil forekomme datakort utover objektkortet.

Anta eksempelvis at det skal leses inn en parameter som er kalt 'P14'. I DATSY-teksten forekommer derfor

```
PARAMETER P14.
```

Hvis verdien skal være lik 28, blir altså objektkortet

```
xP14 = 28
```

Parameterverdiene kan være av to typer, heltall og flytende tall, dvs. tall med komma. Disse behandles på forskjellig måte i maskinen. For visse direktiver er det derfor i direktivdokumentasjonen i del C spesifisert at de aktuelle parametre må være av en spesiell type.

Hvis en parameter skal leses inn som et flytende tall, må det settes komma i parameterverdien på objektkortet. Da engelske konvensjoner følges, punches komma som punktum. I eksemplet ovenfor vil parameteren P14 leses som et heltall slik som verdien står i objektkortet, mens den ville ha vært lest som et flytende tall dersom objektkortet hadde vært

```
xP14 = 28.0 eller bare
```

```
xP14 = 28.
```

Et annet eksempel på et objektkort for en flytendetallsparameter er

```
xPI=3.14159      , 'PI ER;  
xFORUTSATT DEKLARERT SOM PARAMETER.'
```

Her er også illustrert bruken av apostrof som kommentartegn og semikolon som fortsettelsestegn.

B-4.4 Innlesing av navnord

Som parameter leses også navnord inn i "fritt format". Det er ikke nødvendig med noe format og derfor heller ikke med noe formatkort. Som for parameter angis verdien som skal leses inn i objektkortet slik at det vil være datakort for navnord. I objektkortet for navnord settes navnordets egen-

navn lik en navnordverdi. Navnordverdien kan bestå av inntil 77 vilkårlige tegn bortsett fra apostrof. Navnordverdien omsluttes av apostrofer. (Dette er et unntak i bruken av apostrofer i identifikort.)

B-4.5 Innlesing av liste

Lister må leses inn med format. Anta eksempelvis at en liste 'BEDRIFTSLISTE' skal leses inn med et format 'F1'. I DATSY-teksten vil altså følgende setninger forekomme:

LISTE BEDRIFTSLISTE

FORMAT F1.

I dataavdelingen skal det nå finnes et formatkort og et objektkort. Det siste vil umiddelbart være etterfulgt av datakort som inneholder listeelementene punchet i overensstemmelse med det format som er angitt på formatkortet.

Objektkortet for en liste skal som for parameter og navnord inneholde egnavnet for listen. Det må også inneholde formatnavnet. Dessuten kan det inneholde et ledd som angir antall elementer i lista, dette skrives:

R = heltall.

Et eksempel på et listeforamt kan være (A24). Det betyr at det på hvert datakort er et liste-element som er de 24 første tegn på kortet. Anta at brukeren ønsker å lese inn en liste med navnene på sju bedrifter med dette formatet. I DATSY-programmets dataavdeling legges følgende formatkort, objektkort og datakort:

*F1=(A24)

*BEDRIFTSLISTE, R=7, F1

SLIDRE DAMPYSTERI

HILMAR MELET FORLAG

CHRISTIANIA MARK OG SOPP

BERGMAN FISK OG VILT

KOLBOTN BYGG

SANDBERG MASKINER A/S

SANDI NORGE A/S

B-4.6 Innlesing av matrise

I avsnitt A-2.3 gis et eksempel på innlesing av matriser fra hullkort. Objektkortet for en matrise har generelt følgende ledd:

matrisenavn, R=heltall, K=heltall, RS, KS, formategenavn.

Objektkortet kan inneholde følgende ledd i tillegg til eventuelle kommentarled:

Ledd, matrisenavn, er obligatorisk. Det angir egnavnet til matrisen og må stå først etter '*' i første kolonne på kortet.

Ledd, R=heltall, angir antall rader i matrisen. Dette leddet kan sløyfes hvis leddet, KS, sløyfes.

Ledd, K=heltall, angir antall kolonner i matrisen. Dette leddet er obligatorisk.

Ledd, RS, angir at radsummen er gitt som ekstra kolonne. Dette leddet kan sløyfes.

Ledd, KS, angir at kolonnesummen er gitt som ekstra rad. Dette leddet kan sløyfes.

Ledd, formategenavn, er obligatorisk. Formater er nærmere beskrevet i avsnitt B-4.10.

Umiddelbart bak objektkortet må det følge datakort med matrisens elementer. Det er viktig å merke seg at elementene i matrisen leses inn radvis fra venstre mot høyre, ovenfra og nedover.

B-4.7 Innlesing av vektor

Objektkortet for en vektor har følgende ledd:

*vektornavn, R=heltall, S, formategenavn

Objektkortet har to obligatoriske ledd. Det første ledd må være vektorens egennavn, det andre obligatoriske leddet angir formatets egennavn. I tillegg til eventuelle kommentarledd kan ytterligere to ledd tilføres. Det ene er R=heltall som angir hvor mange elementer vektoren består av. Dette leddet kan sløyfes hvis hvert datakort inneholder like mange elementer. Det andre mulige tilleggsleddet, S, angir at summen av elementene er gitt som et ekstra element. Formater er nærmere beskrevet i avsnitt B-4.10.

B-4.8 Innlesing av rekord

Som eksempel vises til innlesing av rekordsett i avsnitt A-2.5

Objektkortet for rekordsett har følgende ledd:

*rekordnavn, formategennavn

Objektkortet består av to obligatoriske ledd foruten eventuelle kommentarledd. I første kolonne på hullkort må stå en stjerne. Så må rekordegennavnet følge. Det andre obligatoriske leddet er formategennavnet. Om format, se avsnitt B-4.10.

B-4.9 Innlesing av rekordsett

Eksempel på innlesing av rekordsett er gitt i avsnitt A-3.5.

Objektkortet for rekordsett har følgende ledd:

*rekordsettnavn, R=heltall, formategennavn

Objektkortet består av to obligatoriske ledd i tillegg til eventuelle kommentarledd. I første kolonne på hullkortet må stå en stjerne. Så må rekordsettegensnavnet følge. Det andre obligatoriske ledd er formategennavnet. Dessuten kan det angis R=heltall der heltallet angir antall rekorder i rekordsettet.

B-4.10 Generelt om formater

Maskinen leser datakort sekvensielt fra venstre. På ett hullkort er det plass til 80 tegn, et tegn i hver kolonne i kortet. Formatene brukes til å angi hvordan de 80 kolonner er brukt til numeriske, alfanumeriske og blanke felter. Formatene angir også om tall er heltall eller flytende tall. Formatet skrives alltid i en parentes.

*formategennavn = ()

Inni parentesen står ett eller flere ledd atskilt med komma som refererer til ett eller flere tenkte felt på datakortene.

Eksempel:

*F1 = (5X, A24, 10X, A16)

I overensstemmelse med denne formatsetningen leses datakortene i fire felt fra venstre. De første 5 tegn tolkes som et felt av typen X-format. De neste 24 tegn tolkes som et felt av typen A-format osv.

Hvis det står flere like felt etter hverandre på datakortene er det mulig å skrive en repetisjonsfaktor:

*F2 = (4A16)

Maskinen tolker dette som fire felt bestående av 16 tegn. Tilsvarende kan gjøres med grupper av felter:

*F3 = (5X, 2(3X, 2(A4)))

Det er mulig med inntil 3 parentesnivåer: Maskinen oppfatter dette som identisk med (5X, 3X, A4, A4, 3X, A4, A4). Interne parenteser må ha repetisjonsfaktor foran seg, selv om innholdet av parentesen bare skal benyttes en gang slik at repetisjonsfaktor er lik 1.

Når maskinen har lest igjennom et hullkort så langt som formatet tilsier, skifter den til neste kort, selv om formatet dekker mindre enn 80 kolonner, og begynner å lese fra venstre på den måten formatet tilsier. Slik blir alle kortene lest likt. Bare for rekorder og rekordsett kan et format beskrive mer enn et datakort. Da skrives en skråstrek ('/') for å angi skift til nytt kort.

For en bruker som er vant med å skrive sine data på kort kan det generelt anbefales at han baserer seg på å skrive i enkle formater, f.eks. bare et element på hvert kort. Da blir formatene enklere å skrive, og feil i ett enkelt element kan lettere rettes opp ved å punche bare dette elementet om igjen.

a. X-format

Et X-format kan generelt uttrykkes som (mX). Dette kan oppfattes som en mulighet til å få maskinen til å la være å lese et felt på datakortene, idet m angir hvor mange kolonner som ikke skal leses. Anta for eksempel at brukeren har en liste med navnord som foreligger punchet på hullkort. Navnordene skal leses inn som objekter av klassen liste. På hvert hullkort er det skrevet to navnord fra og med kolonne 12 til og med 20 og fra og med kolonne 24 til og med 32. Brukeren ønsker å skrive et format som gir maskinen beskjed om bare å lese de feltene hvor navnordene står. Et format som passer kan da være (11X, A9, 3X, A9). Dette illustrerer hvordan maskinen starter å lese sekvensielt fra venstre i overensstemmelse med formatet. Når den kommer til et X-format hopper den over så mange kolonner som m tilsier.

b. A-format

A-format benyttes til innlesing av data som består av alfabetiske, numeriske, eller spesielle tegn, eller kombinasjoner av disse. For en bruker av DATSY kan A-format være aktuelt ved innlesing av lister, rekorder og rekordsett.

A-format har generelt strukturen (An), hvor n er et heltall som angir hvor mange kolonner på datakortene som tolkes som ett element.

Et eksempel på et A-format kan være (A24). Når maskinen skal lese datakort i overensstemmelse med formatet (A24) leser den de første 24 tegn på hvert datakort og tolker dem som ett element.

c. F-format

F-format brukes til innlesing av numeriske data som skal være flytende tall. F-formatet har generelt strukturen (Fs.t) hvor s angir hvor mange av kolonnene som oppfattes som et flytende talls felt, mens t angir hvor mange desimaler det skal være bak desimalkomma. Brukeren kan enten angi komma direkte i datakortet ved å punche punktum der komma skal være plassert i tallet, eller punche tallet uten kommaangivelse og da vil formatet sørge for at komma blir satt inn i tallet når det leses. Et punchet komma, dvs. punktum, har altså prioritet over formatet.

Et anvendelig standard F-format kan være (F14.8). Differansen s-t blir da 14-8=6.

Anta som eksemplerat tallene 804.2; 62.476; 700 skal punches på hvert sitt hullkort slik at de skal leses av maskinen i overensstemmelse med formatet (F14.8).

De må da punches som følger:

Kolonnenr.	1	2	3	4	5	6	7	8	14
fra venstre															
i feltet				8	0	4	2								
					6	2	4	7	6						
					7	0	0								

d. I-format

I-format brukes til å lese inn heltall fra kort. Det er bare i forbindelse med visse direktiver at innlesing av heltall er aktuelt. Det er da presisert i direktivdokumentasjonen (se del C) at dataene må leses inn som heltall med I-format. I-formatet har generelt strukturen (Ib) hvor b er et heltall som angir hvor mange kolonner på datakortene som tolkes som et heltallsfelt. (I8) vil f.eks. være et format som leser de 8 første kolonnene på datakortene og tolker dem som ett heltall. Det må ikke forekomme annet enn numeriske tegn i et heltallsfelt på datakort.

e. Fortsettelsestegn

I forbindelse med rekorder og rekordsett kan ett og samme format "dekke" flere datakort. Dette gjøres ved å bruke et fortsettelsestegn skråstrek ('/') i formatet. Maskinen leser datakortene fra venstre i overensstemmelse med formatet. Hvis den kommer til en skråstrek i formatet hopper den til neste datakort. Den leser da sekvensielt fra venstre på det nye datakortet i overensstemmelse med den del av formatet som står til høyre for skråstreken. På denne måten er det mulig å få maskinen til å tolke det som skrives på flere datakort som en og samme rekord. Dette er særlig aktuelt hvis rekordene blir så store at de ikke får plass på ett datakort.

Eksempel:

```
*FORM1 = (A10, A10/,10X, A25)
```

Dette formatet leser en rekord med tre alfanumeriske felter (NAVN), de to første av lengde 10 og det siste av lengde 25. Feltene er punchet i to og to kort slik at de to første felt er punchet i kolonnene 1-10 og 11-20 i første rekord og det tredje feltet i kolonne 11-35 i annet kort.

B-4.11. Innlesing av fremmedtape

Av avsnitt B-2.5 framgår det at innholdet av en fremmedtape kan betraktes som en del av de datakort som ligger i DATSY-programmets dataavdeling. I overensstemmelse med dette må brukeren skrive et objektkort for hvert objekt på fremmedtapen som skal leses inn av DATSY. Objektkortene må følge etter hverandre i samme rekkefølge som objektene ligger på magnetbåndet.

Objektkort for objekter på fremmedtape inneholder de samme ledd som objektkort for objekter på hullkort. Disse er beskrevet under de forskjellige objektklassene avsnitt B-4.3 til og med B-4.9. I tillegg må objektkort for fremmedtaper inneholde inntil fire andre ledd som er spesielle for objekter på fremmedtape. Disse fire tilleggsleddene er forklart i det følgende.

Det er nødvendig å angi lengden av objektet. Dette kan gjøres på tre alternative måter som er beskrevet i punkt (iv) under:

(i) T-ledd

T = fremmedtapeegennavn

Dette leddet angir at det aktuelle objektet befinner seg på den fremmedtapen som har det egennavn som står i leddet, f.eks.:

T = FORTRAN68

(ii) D-ledd

Hvert objekt i arkivet er identifisert med dag-måned-år-sekund for når det ble generert. De objektene som føres inn i arkivet via fremmedtapen kan få dag-måned-år-sekund status ved det valgfrie leddet D = dd-mm-åå-s hvor:

dd = dag nummer

mm = måned nummer

åå = to siste sifre i året

s = positivt heltall som tolkes som sekunder fra midnatt

Hvis D-leddet utelates brukes dags dato og klokkeslett.

(iii) F- og B-ledd

F = heltall, B = heltall

Disse leddene angir hvor på magnetbåndet objektet begynner. Det forutsettes at brukeren kjenner til hva fremmedtapen inneholder og hvordan innholdet er skrevet i filer og blokker. Om nødvendig må brukeren få vite dette ved å undersøke hvordan fremmedtapen er laget. En fremmedtape kan inneholde flere filer som kan være atskilt av EOF-tegn. Den kan også ha en eller flere label-blokker i første eller i hver file. Eventuelt kan den ha en egen labelfile.

F- og B-leddet angir henholdsvis filenummer og blokknummer for objektet som skal leses. F- og B-leddet er ikke obligatoriske. Hvis ett eller begge disse leddene utelates, gis automatisk F og B verdi ut fra den posisjon magnetbåndet har i øyeblikket.

(iv) Lengden av objektet

Når objekter føres inn i systemet fra hullkort er det i alminnelighet ikke nødvendig å angi lengden av det fordi systemet avslutter ett objekt når det kommer til objektkortet for neste objekt. Når et objekt skal inn i systemet fra fremmedtape er det imidlertid ofte nødvendig å opplyse systemet om lengden av objektet. Dette kan gjøres på en av tre måter:

For det første kan lengden angis eksplisitt ved at leddet R = heltall inngår på objektkortet. Dette angir eksplisitt lengden av objektet.

For det andre kan lengden bli bestemt ved at det neste objektkortet (for det neste objektet som skal leveres) inneholder et B-ledd som angir hvor det neste objektet begynner og altså samtidig hvor det aktuelle objekt slutter. For det tredje kan lengden bli bestemt som resten av filen, dvs. fram til neste EOF-merke.

B-5. Strukturen av et DATSY-program

Hensikten med dette kapitlet er å beskrive reglene som gjelder for rekkefølge av den informasjon som gis til maskinen ved bruk av DATSY, dvs. rekkefølgen av setninger og eventuelt dataobjekter. En viss strukturering er fastlåst, men innenfor dette står brukeren fritt. Innenfor de regler som blir gitt i dette avsnittet kan brukeren med fordel forsyne DATSY-programmet med tomme ord og hensiktsmessige valg av egennavn slik at det blir mest mulig lettlest og selvdokumenterende. Generelt kan det sies at systemet er lagt til rette nettopp for dette.

På neste side er det satt opp en figur som gir en skjematisk framstilling av rekkefølgen av informasjon som er klar til å leveres inn til maskinen i form av en hullkortbunke eller skrevet direkte fra en terminal. Det framgår av figuren at først og sist i kortbunken må det ligge visse styrekort. Disse formidler forbindelsen mellom DATSY-systemet og selve det fysiske maskinutstyret og vil derfor være avhengig av hvilken maskin som benyttes.

Alle kort som ligger mellom styrekortene utgjør DATSY-programmet. DATSY-programmet deles videre inn i først en DATSY-tekst og så en dataavdeling. DATSY-teksten avsluttes alltid av en sluttsetning. Dataavdelingen er bare aktuell når nye data skal leses inn fra hullkort. DATSY-programmet kan således bestå bare av en DATSY-tekst som avsluttes av en sluttsetning.

Maskinprogrammet for DATSY-systemet vil først lese og tolke DATSY-teksten før det begynner behandlingen av dataavdelingen, dvs. leser inn dataobjekter og formater. Tolkningen av DATSY-teksten foregår imidlertid ikke i den rekkefølge setningene kommer. De forskjellige setninger som utgjør DATSY-teksten deles inn i hovedgrupper. Hovedgruppene behandles i en bestemt innbyrdes rekkefølge. Innen hver hovedgruppe behandles setningene i den rekkefølge kortene kommer i bunken.

Setningene i DATSY-teksten deles inn i følgende grupper og behandles i denne rekkefølge.

1. Substitueretninger
2. Deklarasjonssetninger og arkivtapasetninger
3. Deskripsjonssetninger
4. Innførsetninger
5. Velgsetninger
6. Direktivsetninger
7. Skrivsetninger
8. Tømsetninger
9. Sluttsetninger.

B-5.1 DATSY-teksten

I dette avsnittet samles punktvis regler for rekkefølgen av de setninger som skal inngå i en DATSY-tekst.

(i) Substitueretninger må plasseres på riktig sted i DATSY-teksten fordi de har virkning på alle setninger som står bak dem i teksten. Substitueretninger er alltid de første som blir tolket og utført.

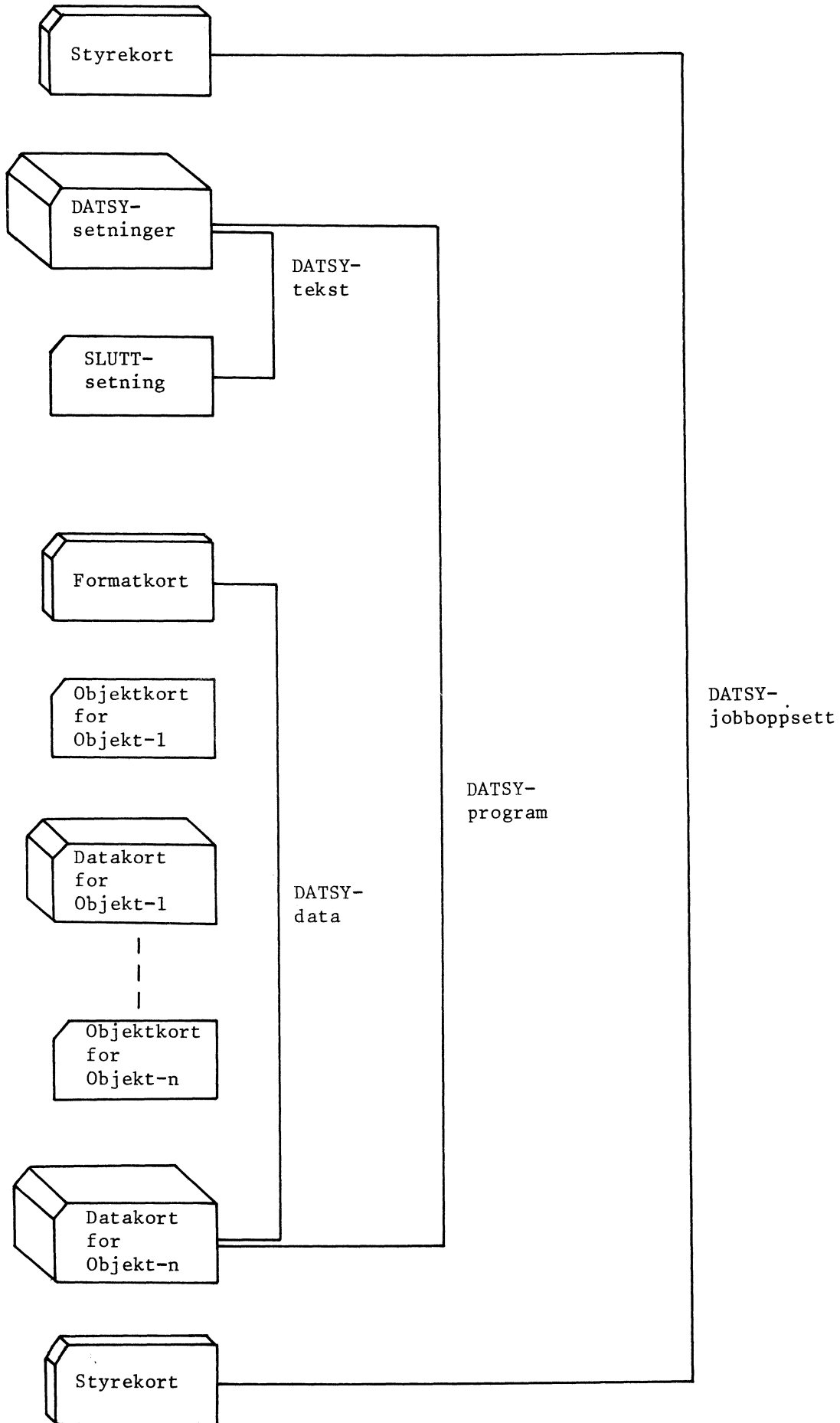


Fig. 5

(ii) Deklarasjonssetninger, arkivtapesetninger, beskrivelsessetninger, innførsetninger, velgsetninger, skrivsetninger og tømsetninger kan plasseres på vilkårlig sted i DATSY-teksten. Med det menes at deres innbyrdes rekkefølge er likegyldig og de kan plasseres innimellom andre setninger.

(iii) Direktivsetninger kan ikke plasseres vilkårlig i DATSY-teksten. Med det menes at deres innbyrdes rekkefølge må være overensstemmende med rekkefølgen av de operasjoner som ønskes utført. Det har f.eks. ingen mening å be maskinen skrive ut på papir et nytt objekt som ennå ikke er generert.

(iv) Den siste setning i en DATSY-tekst må være en sluttsetning.

B-5.2 Dataavdelingen

Hvis brukeren vil innføre nye dataobjekter kan dette gjøres ved å punche dem på kort. Hvordan dette gjøres framgår av kapittel B-4. Det vil generelt være tre typer kort som skal inn i DATSY-programmets dataavdeling: formatkort, objektkort og datakort. For rekkefølgen av disse kortene gjelder følgende regler:

(i) Hvis et dataobjekt går over flere datakort, må disse datakort komme etter hverandre samlet.

(ii) Objektkortet til et dataobjekt må plasseres umiddelbart foran de datakort som inneholder dataobjektet. Det kan være gunstig å tenke seg et dataobjekt som en "pakke" av ett objektkort og ett eller flere datakort.

(iii) Den innbyrdes rekkefølge av dataobjekter i en dataavdeling er vilkårlig.

(iv) Formatkortene kan plasseres vilkårlig bare ikke regel (i) og (ii) brytes. Med det menes at formatkortene kan plasseres vilkårlig hver for seg eller samlet før, mellom eller etter "pakkene" av objektkort og datakort.

B-6. Utskrifter fra DATSY

Utskriften som kommer fra maskinen gjør systemet og bruken av det til en viss grad selvdokumenterende. Dette kapitlet vil redegjøre for de utskrifter som gis. Utgangspunktet er en beskrivelse av en ordinær utskrift ved kjøring av et DATSY-program (B-6.1). De forskjellige typer feilmeldinger, og hvor de kommer i utskriften vil bli forklart (B-6.2). Et eget avsnitt viser så hvordan brukeren kan få gamle og nygenererte dataobjekter skrevet ut og hvor de eventuelt kommer i utskriften (B-6.3). Til slutt i kapitlet er et avsnitt som forklarer hvordan brukeren kan få maskinen til å skrive ut ytterligere opplysninger om kjøringen ved bruk av DATSY på UNIVAC 1108 (B-6.4).

B-6.1 Ordinære utskrifter

Utgangspunktet for dette avsnittet er at brukeren har levert inn til maskinen et DATSY-program som beskrevet i kapittel B-5. Med ordinære utskrifter menes utskrifter som maskinen gir uten at brukeren har benyttet verken opsjoner (se B-6.4) eller utskriftsetninger (se B-6.3), og at det heller ikke er blitt begått noen feil som fører til feilmeldinger (se B-6.2).

Etter maskinens standardutskrifter og utlister av eventuelle styrekort vil det skrives ut en gjengivelse av DATSY-teksten fram til og med sluttsetningen. Hvert enkelt kort i DATSY-teksten blir listet for seg og nummerert fra én og oppover. Dessuten vil alle systemord i DATSY-teksten bli gitt en dobbeltunderstrekning, mens alle ikke-tomme brukerord blir gitt enkel understrekning. Eventuelle tomme ord vil også bli gjengitt, men uten understrekning.

Hvis DATSY-programmet også inneholder en dataavdeling vil denne være det neste som skrives ut. Under overskriften "UTSKRIFT AV DATAKORTENE" blir innholdet av datakortene gjengitt samtidig som de nummereres fra én og oppover.

Til slutt vil utskriften inneholde en del tekniske opplysninger som har framkommet ved tolking av DATSY-programmet. Det gis bl.a. opplysninger om hvor mange nye objekter som er blitt generert.

Det blir videre gitt to tidsangivelser. "CPU-TID MEDGÅTT TIL DATSY-EKSEKVERING" angir den beregningstid det har tatt å behandle DATSY-programmet. (Den tid det har tatt å tolke styrekortene er ikke med her.) "TOTAL RUN-CPU-TID MEDGÅTT TIL NÅ" angir den totale beregningstid kjøring har tatt inkludert behandlingen av styrekort. Hvis samme kjøring inneholder flere DATSY-program angis "TOTAL RUN-CPU-TID MEDGÅTT TIL NÅ" etter behandlingen av hvert DATSY-program. Disse tidsangivelsene er da akkumulert.

Beregningstid er ikke det samme som gjennomløpt tid for en kjøring. Denne er angitt i utskriften ved at start-tidspunkt og slutt-tidspunkt skrives ut. Differensen mellom gjennomløpstid og beregningstid kalles liggetid. Liggetiden er bl.a. avhengig av den aktuelle belastningen på maskinen.

"DATSY LEVEL" forteller hvilken versjon av kontroll- og styreprogrammet som har vært benyttet. Datoen for siste oppdatering av programmet er også angitt.

B-6.2 Feilmeldinger

Ved konstruksjon av DATSY-systemet er det lagt stor vekt på at de feil en bruker gjør, skal bli oppdaget så tidlig som mulig. Hvis slike feil blir oppdaget, skal utskriften inneholde feilmeldinger som er ment å være selvforklarende og peke mest mulig direkte på den feil som er gjort. Hvis feilen er fatal, avbrytes behandlingen av programmet. Da det ikke er mulig å forutsi alle feil en bruker kan komme til å gjøre, vil de feilmeldinger som gis i enkelte tilfelle kunne være misvisende. Det kan også tenkes at maskinen oppdager flere tilsynelatende feil som følge av den ene reelle feil. Dette resulterer ofte i at utskriften inneholder flere feilmeldinger som følger av at brukeren har gjort én feil.

Anta f.eks. at brukeren har glemt å deklare et objekt. Alle steder hvor objektet inngår i teksten vil det da bli oppfattet som et tomt ord i stedet for et objekt. Dette kan føre til en rekke feilmeldinger om at antall objekter som inngår i direktivene, ikke er korrekt.

Brukeren bør også være oppmerksom på at systemet stadig forbedres og utvikles slik at det aldri vil bli absolutt helt utprøvet. Det kan tenkes at brukerens DATSY-program med eller uten feil vil kunne bringe fram svakheter og feil i selve systemet. Dette vil alltid være tilfelle hvis feilmeldingen inneholder sekvensen "INTERN FEIL". I tilfelle der brukeren får feilmeldinger han ikke forstår, bør han ta kontakt med systemgruppen som har ansvaret for DATSY.

I tillegg til alle feil i brukerens DATSY-program som DATSY kan analysere, vil det gjenstå feil som er en følge av feil i brukerens logikk i problemløsningen. Anta at brukeren tror han kan løse et problem på en bestemt måte. Hvis dette er galt, men brukeren anvender DATSY korrekt, får han galt svar, men selvsagt ingen feilmeldinger.

De feilmeldinger som DATSY gir kan naturlig deles i fire grupper. Dette gjøres i overensstemmelse med at først leses DATSY-teksten, så dataavdelingen, så sammenholdes disse to og til slutt utføres operasjonene i DATSY-teksten.

a. Feilmeldinger som refererer til DATSY-teksten

Disse feilmeldinger skrives ut samlet etter utskriften av DATSY-teksten. Et eksempel på en feil av denne type er at et objektetnavn er punchet feil i en direktivsetning. Et annet eksempel som refererer til DATSY-teksten kan være at et direktiv som krever fem dataobjekter bare har fire dataobjekter i sin setning. En vanlig nybegynnerfeil vil være å glemme punktum. Neste setning vil da oppfattes som en fortsettelse av forrige og antallet systemord og objekter i "setningen" blir galt.

b. Feilmeldinger som refererer til datakortene

Disse feilmeldinger skrives ut samlet etter utskriften av datakortene. Et eksempel på en feil av denne type kan være at et objektkort mangler stjerne i første kolonne. En annen typisk feil kan være at datakortene ikke er punchet i overensstemmelse med det format som skulle brukes.

c. Feilmeldinger som refererer til uoverensstemmelser mellom DATSY-teksten og dataavdelingen

Disse feilmeldinger skrives også samlet etter utskriften av dataavdelingen. Feilmeldinger av denne type refererer særlig til uoverensstemmelser mellom beskripsjoner i DATSY-teksten og formatene i dataavdelingen. Slike feilmeldinger kan være vanskelige å tolke. Hvis det forekommer feilmeldinger av type b og c i samme utskrift vil type b bli skrevet ut først.

d. Melding om feil som blir oppdaget under eksekveringen av DATSY-programmet

Vanligvis vil eksekveringen av DATSY-programmet bli avbrutt hvis en feil av denne type blir oppdaget. Feilmeldingene kommer da etter utskriften av dataavdelingen og eventuell utskrift fra direktiver som er utført før feilen inntraff. Hvis DATSY-teksten inneholder direktiver om at dataobjekter skal skrives ut på papir og brukerens feil oppdages etter at slik utskrift allerede er foretatt, kommer feilmeldingene etter disse dataobjekter. Disse feilmeldingene blir gjerne etterfulgt av en side med tekniske data som gir en viss systemteknisk informasjon om hva som har gått galt.

B-6.3 Utskrift av dataobjekter

Det framgår av avsnitt B-6.1 hva en ordinær utskrift inneholder. Utskriftsetningene er en av brukerens muligheter til å få skrevet ut på papir mer enn hva som er standard. Fem direktiver er utviklet for dette formål. Et eget avsnitt er her viet en oversikt over disse direktiver. De er også tatt med i den generelle oversikt over alle direktiver i del C.

(i) PRINTVERDI dataobjektegennavn

En slik utskriftsetning fører til at verdien av dataobjektet blir skrevet ut på papir etter DATSY-programmets dataavdeling. PRINTVERDI-setninger kan brukes på alle klasser av dataobjekter.

(ii) PRINTMATLIST matriseegennavn listeegennavn listeegennavn listeegennavn

Dette direktivet fører til at verdien av en matrise blir skrevet ut. Samtidig skrives også innholdet av tre lister: overskrift, radnavn og kolonnenavn.

(iii) PRINTMAT matriseegennavn.

Dette direktivet er en videre utvikling av PRINTMATLIST. Ved hjelp av direktivet IDENTIFISER er det mulig å tilordne lister til en matrise. Hvis det er gjort kan PRINTMAT direktivet få listene skrevet ut sammen med matrisen analogt med PRINTMATLIST, men uten at listeegennavnene spesifiseres i selve direktivet. Hvis en slik tilordning av lister ikke er foretatt, skrives bare verdien av matrisen med angitt rad- og/eller kolonnennummer.

(iv) PRINTREKSETT objektetegennavn.

Dette direktivet brukes for å få skrevet ut verdien av dataobjekter av klassene REKORD og REKORSETT. Samtidig skrives dataobjektets egennavn og beskrivelse.

(v) OPPLYSOBJEKT objektetegennavn.

Dette direktivet kan benyttes på objekter av alle klasser, inkludert hjelpeobjekter. Hensikten med dette direktivet er at brukeren kan få alle opplysninger om objektene selv når de er uten innhold eller ikke beskribert.

B-6.4 Opsjoner på UNIVAC 1108

Brukeren av opsjoner er svært maskinavhengig. Dette avsnitt forutsetter at brukeren kjører på en UNIVAC 1108 maskin.

Et av styrekortene er et EXEC-8 eksekveringskort. På dette kortet er det mulig å skrive en eller flere bokstaver som er gitt spesiell betydning kalt opsjoner. Dette avsnittet inneholder en liste over de opsjoner som er mest aktuelle for en bruker.

Alle de opsjoner som er beskrevet her har til felles at de representerer en mulighet til å få skrevet ut på papir supplerende opplysninger om kjøringen. M-opisjon og Z-opisjon er svært vanlig å bruke, og hvis det i løpet av en kjøring skal skrives på arkivtape er det vanlig å bruke følgende opsjonkombinasjon MZBC.

- (i) B-opsjon gir en utskrift av katalogen uten detaljer om de enkelte objektene.
- (ii) C-opsjon gir en mer utfyllende utskrift av katalogen med opplysninger om de enkelte objektene, som f.eks. datoen de ble innført. C-opsjon brukes sammen med B-opsjon eller ved bruk av skrivsetninger. Både B- og C-opsjoner er mest aktuelle for dataadministratoren, men en bruker kan med fordel anvende dem i forbindelse med skriving på magnetbånd.
- (iii) I-opsjon gir en forkortet utskrift av datakortene. Denne tenkes benyttet ved innlesing av store dataobjekter hvor det er uhensiktsmessig å få skrevet ut innholdet av samtlige datakort.
- (iv) K-opsjon gir opplysninger om hvilke operasjoner som foretas inne i direktivene og i noen tilfelle mer utfyllende varsler om mulige feil (uhensiktsmessigheter).
- (v) M-opsjon fører til at eksekveringstiden for direktivene blir skrevet ut. M-opsjon viser også om direktivene er utført. I visse tilfelle får man tekniske opplysninger om virkemåten av direktivene under eksekvering.
- (vi) R-opsjon fører til at det skrives ut et resymé over den teksten som blir lest inn.
- (vii) Z-opsjon gir utskrift bare hvis det blir begått feil av type d (se B-6.2). Dessuten gis systemteknisk informasjon om feil som er begått. Bruk av Z-opsjon gjør det raskere å rette systemfeil, da Z-opsjon vil skaffe til veie de opplysninger som er nødvendig for å rette systemfeil.

B-7. Dataobjekter med identiske navn

Innen en arkivtape og innen en dataavdeling i et DATSY-program må alle objekter ha forskjellig egnavn. Det vil imidlertid ofte kunne forekomme at to eller flere arkivtaper inneholder forskjellige objekter med samme egnavn. Det kan også tenkes at objekter på datakortene kan ha samme navn som objekter på en arkivtape. I DATSY-program som vil benytte arkivtaper og innleste dataobjekter på en slik måte at det inngår forskjellige objekter med samme navn må man ta hensyn til dette slik at direktivsetningene blir entydige. Brukeren har tre måter å gå fram på for å oppnå dette.

(i) Prioriteringsregler

Hvis et objektetegnavn i en DATSY-tekst ikke entydig peker på et bestemt objekt, men på to eller flere objekter, velges et av disse objektene i overensstemmelse med følgende prioriteringsregel. Hvis det finnes et objekt med det gitte egnavnet i DATSY-programmets dataavdeling velges dette objektet. Hvis ikke prioriteres de forskjellige arkivtaper i den rekkefølge de ble deklarerert. Denne prioriteringsregelen kommer til anvendelse først etter at eventuelle substitusjonssetninger (B-3.5), arkivdeskripsjoner (B-3.4) og velgsetninger (B-3.6) er utført.

(ii) VELG-setninger

Ved hjelp av VELG-setninger kan brukeren sette nye midlertidige, på dataobjekter på arkivtaper. De nye navnene har bare gyldighet i det DATSY-programmet der VELG-setningene forekommer. VELG-setninger er nærmere beskrevet i avsnitt B-3.6.

(iii) Arkivtapedeskripsjoner

Ved hjelp av arkivtapedeskripsjoner kan brukeren sette midlertidige navn på objektene på en arkivtape som nærmere beskrevet i avsnitt B-3.4.

C. DIREKTIVBESKRIVELSER

C-1. Innledning

I kapitlene C-2 og C-3 nedenfor er alle direktiver som er implementert eller under implementering beskrevet. Foran i hvert avsnitt er det gitt en oversikt over hvilke direktiver avsnittet inneholder. En del direktiver er ført opp i oversikter i flere avsnitt. Det gis da henvisning til det avsnitt der direktivet er beskrevet.

Innen hvert avsnitt er direktivene ordnet alfabetisk. For å lette oversikten begynner beskrivelsen av hvert direktiv på ny side, og direktivnavnet er angitt i øverste høyre hjørne på hver side direktivbeskrivelsen strekker seg over.

I beskrivelsen av hvert direktiv er det først angitt strukturen på direktivsetningen. Denne er angitt på formen:

Form: DIREKTIVNAVN fellesnavn1 fellesnavn2 ... fellesnavn3.

De understrekede argumentene angir objektklasse for de objekter - i gitt rekkefølge - som inngår i direktivet. Dersom et argument kan tilhøre alternative objektklasser, er alle disse angitt atskilt av skråstrek, f.eks. vektor/matrise. Det vil oftest også være innskutt tomme ord for å øke lesbarheten og tolkingen av direktivet.

For øvrig vil beskrivelsen gi så vidt mulig fullstendig informasjon som er nødvendig for å kunne bruke direktivet i et DATSY-program. Det logiske innholdet i operasjonen(e) som direktivet utfører er beskrevet og eventuelt formatet på output fra direktivet. Forutsatte egenskaper ved objekter som inngår som argumenter i direktivet, er beskrevet i den grad det er nødvendig for brukeren å kjenne til disse utover de som følger av objektklassen. De viktigste feilmeldinger som direktivet vil gi er også beskrevet eller angitt for en del direktiver.

I en del direktiver er det tillatt å benytte objekter fra en dummy arkivtape kalt STANDARD. Dersom et av disse skal benyttes må DATSY-teksten der direktivet inngår inneholde arkivtapesetningen.

ARKIVTAPE STANDARD.

Objekter på denne dummy arkivtappen blir referert til som STANDARD-objekter.

C-2. Generelle direktiverC-2.1 Opplysnings- og utskriftdirektiver

<u>Direktiv</u>	<u>Status pr. 1/1-74</u>
I-O	virker
LET	ikke implementert
NY-SIDE (se C-2.8)	-
OPPLYSOBJEKT	virker
PRINT	virker
PRINTMAT	ikke skrevet
PRINTMATLIST	virker
PRINTNRADER	virker
PRINTREKSET (se C-2,6)	-
PRINTVERDI	virker
SENDPRINT	ikke testet

Direktiv I-O

(Input-Output)

Form: I-O FRI/ta111/objekt1 arkivtape2/nytape2/eksterntape2/
navn2/parameter2.

Direktivet kan utføre to forskjellige operasjoner:

(i) Assigning av arkivtape

Når det første objektet er av klassen TALL med navn på objektet et 4-sifret nummer (inntil 6 sifre er tillatt) og det andre objektet er arkivtape2, vil arkivtapen bli montert ferdig til å skrive på. Formålet med direktivet er altså å unngå å beleggemagnetbåndstasjoner før det er nødvendig.

Direktivet skrives sist av direktivene men før skrivsetninger som skriver på arkivtapen. Magnetbåndet med arkivtapen må ikke være assignet (med ASG-kort) før XQT- kortet. Det bør imidlertid nevnes på kjøreseddelen til jobben.

Eks.:

ARKIVTAPE B.

TALL 8446.

I-O 8446 MONTERES TIL SKRIVING SOM ARKIVTAPE B.

SKRIV PÅ B OBJEKTER C OG D.

.
.

.

Det er vanligvis bare adgang til å få montert fra én til tre taper på denne måten, siden et større antall båndstasjoner ikke vil være tilgjengelig.

(Direktivet fungerer ikke på denne måten ennå, grunnet en feil i rutinen CONSOL fra NR*LIBRARY.)

Direktiv I-O kan brukes for å frigjøre eller assigne magnetbånd som ikke inneholder arkivtaper. Disse kan være nytaper, eksterntaper eller fremmedtaper, f.eks. Fortran-formaterte bånd lest eller skrevet av direktivene LESBCDTAPE og SKRIVBCDTAPE. Hvis magnetbåndet er en fremmedtape vil det andre objektet da være parameter2 istedenfor arkivtape2.

Eks.:

PARAMETER 2.

NAVN FRI.

TALL S14125.

I-O S14125 MONTERES (FERDIG TIL LESING OG SKRIVING) MED SPESELLE OPTIONS (-TIME) SOM TAPE NR 2.

LESBCDTAPE TAPEN PÅ UNIT NR 2 LESES MED FORMAT F OG GIR REKORDSETT BBB.

I-O FRI 2. (TA VEKK TAPE NR 2 FRA STASJONEN)

Magnetbånd må ikke deklarereres som ARKIVTAPE, hvis de ikke er innført som ARKIVTAPE. Videre pekes det på at hvis tapenummeret (f.eks. S14125) begynner med bokstaven 'S' (som eksempletgjør), vil tapen bli innlest med options TIEH på det ASG-kortet som direktiv I-O vil generere (ikke programmert riktig ved skrivende dato).

(ii) Fra DATSY-objekt til SDF-file (på UNIVAC) eller SSF-file (på H6060):

Hvis det første objektet ikke er FRI og heller ikke av klassen TALL, blir funksjonen av direktiv I-O som følger:

I-O skriv objekt1 ut på Fastrand SDF-file navn som blir opprettet av direktivet selv.

Objekt1 kan i prinsipp være et hvilket som helst objekt. Utskriften vil bli identisk med den tilsvarende utskriften fra PRINTVERDI, bortsett fra to punkter:

- a) Den vanlige hodeskriften ("PRINTVERDI objekt") mangler og objektets verdier begynner på første linje i SDF-filen navn2.
- b) Det kommer alltid ett ekstra mellomrom i den første printposisjonen, slik at alle de andre tegnene er således flyttet en posisjon til høyre.

Navnet på SDF- eller SSF-filen må altså deklarereres som NAVN. Filenavnene FILE1 og FILE2 er innført som STANDARD-objekter.

Formålet med denne bruken av I-O er at ved å skrive:

"ADD navn2. (punktum er viktig)

etter et identifikatorkort i data-avdelingen til en senere DATSY-eksekvering, kan man innlese objektet på nytt, kanskje etter at annen systemsoftware ("ED-prosessoren, f.eks.) er blitt brukt for å forandre det. (Innlesingen må skje med format, og dette resonnerer man seg fram til ut fra bruksanvisningen for direktiv PRINTVERDI, pluss det ekstra mellomrommet i den første printposisjonen.)

Med COPY,I-kort på UNIVACK kan objekt1 forandres videre fra en SDF-file til ett element i en programfile eller i en elementfile. På H6060 er dette ikke nødvendig.

Dette direktivet kan benyttes som et ledd i forandring av siffer i en LISTE til hele tall i en VEKTOR, eller omvendt, ved å skrive objektet ut på FILE1 og lese det inn igjen på nytt med et nytt format.

Følgende feilmelding skjer bare i forbindelse med I-O direktivet:

```
EXEC 8 HAR GITT FEIL-INTERRUPT VED CORE-ADRESSE ...
(FEIL-TYPE: 01, FEIL-KODE: 021, (SE EXEC 8 APP. B))
I/O CALL ERROR. ULOVLIG PARAMETER TIL EN I/O-OPERASJON.
PAKKEN LIGGER VED ...
```

Dette skjer når man har frigjort en arkivtape med I-O. DATSY forsøker nå å spole tilbake arkivtapen men kan ikke, fordi den ikke lenger er på båndstasjonen.

Direktiv LET

(LET etter verdi)

Form: LET i objekt1 i nærheten av horisontnr2 etter nærmeste forekomst av verdien spesifisert i parameter3/navnord3/rekord3 og angi posisjonen i horisontnr4 og vertikaltnr5.

Direktiv LET skal finne den nøyaktige posisjon i et generelt objekt objekt1 av en alfanumerisk eller numerisk verdi oppgitt av brukeren enten i parameter3, navnord3, eller i rekord3 (som kan brukes til spesifisering av flere elementer samtidig i et rekordsett). Letingen foretas først i et område som brukeren selv spesifiserer ved heltallsparameteren horisontnr2, som har tilsvarende tolking som horisontnr4, se fig. 6 nedenfor. Hele objekt1 blir imidlertid undersøkt.

Det er anledning til å bruke STANDARD-objektene DUMMY, TOPP, MIDT, eller BUNN som horisontnr2. Dette innebærer at letingen foretas først fra begynnelsen (DUMMY eller TOPP), fra midt (MIDT) eller fra slutten (BUNN) av objekt1.

Når verdien angitt i parameter3 (alternativt navnord3 eller rekord3) er funnet blir posisjonen satt inn i resultatobjektene horisontnr4 og vertikaltnr5, hvilke er heltallsparametere. (Med K-opsjon (se B-7.4) blir i tillegg horisontnr4 og vertikaltnr5 skrevet ut på linjeskriveren med tolking.)

Tolkingen av horisontnr4 og vertikaltnr5 er avhengig av klassen til det generelle objekt, objekt1, som kan være enten matrise, vektor, liste eller rekordsett.

Tabellen viser tolkingen:

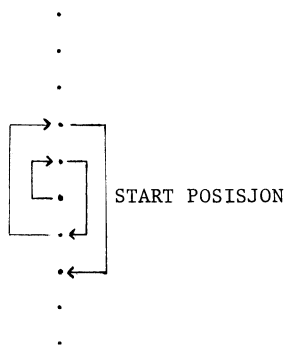
LET

<u>objekt1</u>	<u>horisontnr4</u>	<u>vertikaltnr5</u>
matrise	radnummer	kolonnennummer
vektor	elementnummer	fastsatt = 1
liste	elementnummer	fastsatt = 1
rekordsett	rekordnummer	elementærfeltnummer

Fig. 6

Når verdien angitt ved parameter3 (eller navnord3, eller rekord3) forekommer flere ganger i objekt1 må brukeren selv ta hensyn til leterekkefølgen, fordi posisjonen som angis i horisontnr4 og vertikaltnr5 er posisjonen til den første forekomst av verdien som blir funnet. Med standard-ordene DUMMY eller TOPP foregår letingen sekvensielt fra begynnelsen av objektet med BUNN sekvensielt fra slutten.

Med MIDT eller heltallsverdier angitt av brukeren som horisontnr2 søkes det først ved den horisontale posisjon som er angitt, fra venstre mot høyre. (F.eks. for rekordsett søkes først i den midterste rekord fra venstre til høyre, eller den rekord med rekkefølgenummeret horisontnr2. Se også i tabellen ovenfor.) Deretter søkes det like før, og så like etter, denne posisjonen etter mønsteret:



Etter et visst antall skritt vil det samme letemønsteret fortsette med den forskjell at hele grupper med linjer blir undersøkt før direktivet hopper til ny leteposisjon. Dette gjelder bare for store objekter. Av effektivitetshensyn bør brukeren treffe et fornuftig valg med hensyn til horisontnr2 når objekt1 er veldig stor.

Alfanumeriske tegn (liste, rekordsett):

Hvis objekt1 er en liste må det tredje objekt være navnord3 (eller alternativt rekord3 med bare ett alfanumerisk elementærfelt). Navnord3 kan ha færre tegn enn listeelementene i objekt1. Tegn sammenliknes fra venstre til høyre inntil det ikke finnes flere tegn å sammenlikne med i navnord3 (alternativt rekord3). Man kan således lete etter et listeelement ved å spesifisere bare de første få sammenhengende tegn i dette.

Med rekordsett derimot er dette noe annerledes. Her kan man styre letingen ved å bruke beskrivelsen til rekord3. Elementærfelt i rekord3 sjekkes mot elementærfelt i rekordsettet bare hvis beskrivelsene stemmer. Da er det også nødvendig at alle tegn i de sammenliknede elementærfelt stemmer overens. Altså, når man bruker rekord for å spesifisere leting i rekordsett må alle alfanumeriske tegn i et elementærfelt oppgis. (Rekorden kan imidlertid ha et mindre antall elementærfelt enn rekordsettet.)

Det går imidlertid også an å bruke navnord3 for å spesifisere leting i rekordsett. I slike tilfelle blir det søkt i alle elementærfelt i rekordsettet med tegnantall større eller lik tegnantallet i navnord3.

Numeriske verdier (matrise, vektor, rekordsett):

Numeriske verdier kan være heltall - eller flytende tall. Hele tall må stemme eksakt, mens flytende tall anses som like hvis differansen er meget liten (1×10^{-20}). (Dette er nødvendig pga. avrundingsfeil.) For leting etter en rekord i et rekordsett må som ovenfor, beskrivningene stemme overens. Elementærfelt som skal sammenliknes må være enten begge heltall eller begge flytende tall.

Direktiv OPPLYSOBJEKT

(OPPLYS om OBJEKT)

Form: OPPLYSOBJEKT objekt.

Ved å bruke dette direktivet får man følgende opplysninger om objektet skrevet ut på linjeskriver (terminal):

- (i) Hva slags objekt det er deklarerert som.
- (ii) Om det er lagret på arkivtape, deklarerert i samme DATSY-program, eller ikke.
- (iii) I tilfelle det er lagret på arkivtape får man dato og klokkeslett da det ble lagret. I tilfelle det ikke er lagret på arkivtape (og følgelig er deklarerert i kjøringen hvor direktivet er brukt), får man opplysning om objektet på dette trinn i programmet har en bestemt struktur og verdi, eller bare en bestemt struktur, eller bare er deklarerert.
- (iv) Om objektet har en bestemt struktur, får man vite hva denne er.

F.eks.:

For matrise, får man dimensjonene.

For rekord, felt og rekordsett får man navnene som ble brukt til å beskrivere objektet (dette punktet er ikke implementert ennå).

Direktiv PRINT

Form: PRINT objekt printes (normalt på linjeskriver eller terminal) med format oppgitt av brukeren.

Innobjekter: objekt, format.

Utobjekt: ingen.

objekt kan i prinsipp være et hvilket som helst objekt som har innhold (ikke FELT, NAVN og TALL). format må enten ligge ferdig i arkivet eller bli innlest fra datakort av brukeren. objekt og format må stemme overens, ellers stopper direktivet med feilmelding. (En begrensing her er at ledd i format utover dethundrede blir ikke sjekket, men format med flere enn 100 ledd kan likevel brukes.) Som vanlig kan et rekordsett/rekord ikke ha flere enn 24 elementærfelt.

Istedenfor format får brukeren lov til å spesifisere en heltallsparameter (som oppbevart på ARKIVTAPE standard) for å angi et antall linjer som skal hoppes over. Da må objekt også spesifiseres, men siden det ikke blir skrevet ut, gjør det ikke noe hvilket objekt (bortsett fra FELT, NAVN, TALL og uimplementerte objekter) man skriver her. Flere enn 99 linjer kan man ikke hoppe over ad gangen.

Likedan kan det spesifiseres et standardnavnord, NYSIDE, for å hoppe til toppen av neste side. Dette ligger også på standard. Hvis man ikke anvender standard, må man huske at parametrene og navnordet må ha identiske navn og innhold, som angitt i eksempel 2 nedenfor.

Navnet til objekt kan også skrives ut, ved å spesifisere standardnavnordet VEDNAVN. objekt må fremdeles skrives ut ved å bruke direktivet en gang til hvis man ønsker innholdene i tillegg. Ved bruk av direktiv SENDPRINT kan utskriften sendes til en file (Fastrand, tromme, disk) utenfor DATSY. Denne veien kan det også ordnes punching av kort med utskriften på. Eksempel 1 viser et enkelt oppsett, og resultatet, menseksempel 2 viser oppbygging og utskrift av en tabell.

Eksempel 1

```

PRINT REKSET1 MED NYSIDE.
PRINT REKSET1 MED 5 LINJER SKIPPET.

PRINT REKSET1 VEDNAVN.
PRINT REKSET1 3.
PRINT REKSET1 MED FORMAT1.
PRINT REKSET1 MED 10 LINJER SKIPPET.

FORMAT FORMAT1.
ARKIVTAPE MEDREKSET1, STANDARD.

SLUTTPROD.

*^ FORMAT1 = (4X, 'REKSET:' 1X,A10,1X,A5,1X,F8.3)

```

Resultat (på ny side):

```

---
---
--- (5 linjer skippet)
---
---
REKSET1
---
--- (3 linjer skippet)
---
^^^ REKSET1:~ABCDEFGHIJ~ XYZWX~-100.400
^^^ REKSET1:~BCDEFGHIJK~ YZWXY~+~41.321
---
---
---
--- (10 linjer skippet)
---
---
---

```


Eksempel 2

(Tabell, uten å bruke standard)

ARKIVTAPE MED-INNHOLDENE-TIL-TABELL.

PRINT INNHold (SOM ER EN MATRISE SOM LIGGER PÅ ARKIVTAPEN OVENFOR) MED NYSIDE.

PRINT INNHold MED 5 LINJER HOPPET OVER.

SETTVEDSIDE RADLISTE OG INNHold, RESULTATET BLIR TABELL.

REKORDSET TABELL.

NAVN RADENE.

TALL INNHold.

TABELL DESKRIBERES MED RADENE OG INNHold.

PRINT LISTE1 MED F3 (SE NEDENFOR).

FORMAT F1,F2,F3,F4.

PRINT TABELL MED 2 LINJER HOPPET OVER.

PRINT TABELL MED F4.

PRINT TABELL MED 1 NY LINJE.

PRINT LISTE2 MED FORMAT F3.

PARAMETER 1 2 5.

NAVNORD NYSIDE.

SLUTTPROD.

*F3= (25X,A80)

*F4= (25X,A24,2X,9F6.0)

*2 = 2

*1 = 1

*5 = 5

* NYSIDE = 'NYSIDE'

*F2= (A24)

*F1= (A80)

* LISTE1, F1

```
-----
----- tabell-hode -----
-----
```

* LISTE2, F1

```
----- tabell-bunn -----
```

* RADLISTE, F2

```
-----
-----
-----
-----
-----
```

Resultat:

```

---
---
--- (5 linjer blanke)
---
---
                                                    PRINT/PRINTMAT

-----
----- tabell-hode -----
-----

--- (2 linjer blanke)
---
----- (innholdene) -----
-----
-----

--- (1 linje blank)
----- tabell-bunn -----

```

Perspektiver for videreutvikling:

- (i) For å imøtekomme krav om spesielle tegn i innholdene (og ikke bare tall), samt komma istedenfor punktum som desimaltegn, trengs et direktiv som vil konvertere tallene til karakterer. Innholdene kan også forandres interaktivt fra terminal med en EDITOR, utenfor DATSY.
- (ii) Et direktiv som vil generere tabell-hoder fra kolonnelister kunne tenkes. Alternativt kunne dette gjøres interaktivt utenfor DATSY.
- (iii) Hvis DATSY noen gang blir interaktiv, har man bruk for et EDITOR-direktiv som vil kunne forandre tabelldele direkte fra terminal. Men det finnes ikke bruk for et slikt direktiv til batchkjøringer, fordi tabelldelene kan rettes, dupliseres og forandres mye bedre med en punchemaskin eller med en eksisterende EDITOR utenfor DATSY.
- (iv) H6060-systemet RUNOFF er relevant her når større mengder av tekst kommer inn.
- (v) DATSY egner seg ikke for tabellutprintinger hvis ikke tallene ligger i et DATSYarkiv fra før av. Interaktive og batchprogrammer skrevet i COBOL og brukt fra terminal er den eneste praktiske metoden (andre metoder tar for lang tid). Det henvises til systemene KOBLES og NATBLES.

Direktiv PRINTMAT

(PRINT MATrise)

Form: PRINTMAT matrise med eventuell tilhørende radliste, kolonnaliste og dokumentliste skrives ut på linjeskriveren.

Direktivet er lik PRINTMATLIST (se nedenfor) bortsett fra tre punkter:

- (i) De tilhørende listene nevnes ikke av brukeren, men tilknyttes matrisen på forhånd hvis nødvendig, med direktivene IDENTIFISER og DOKUMENTER.
- (ii) En eller flere av de tre mulige lister som kan være tilknyttet matrise kan mangle. I så fall blir rad- og/eller kolonnennumre brukt for å erstatte evt. manglende rad- og/eller kolonnenlister.

- (iii) De tre første listeelementer i dokumentasjonslisten blir til en tittel, mens de øvrige blir skrevet ut etter matrisen på en ny side med overskriften: DOKUMENTASJON FOR MATRISE matrise.

Direktiv PRINTMATLIST

(PRINT MATrise med LISTer)

Form: PRINTMATLIST matrisel/vektorl/parameterl med overskriften liste2 radnavnene liste3 og kolonnenavnene liste4.

Dette direktivet vil printe en matrise med overskrift, radnavn og kolonnenavn. Direktivet skriver 8 fulle kolonner ad gangen, hvert tall med 4 desimaler og uten eksponent. Overskriften tenkes tatt fra kort lest med format (A80). Hvert kort blir én linje i overskriften. Hvert enkelt element i radnavnlisten tenkes å ha et fast format mellom (A1) og (A24), dvs. at ingen av disse navnene må overstige 24 tegn. Hvis radnavnene er lenger, vil direktivet bare skrive ut de første 24 tegn. Et varsel blir printet ut på en ny side etterpå. Kolonnenavnene kan inneholde inntil 80 tegn hver.

Radnavnene skrives ut på én linje hver, mens kolonnenavnene deles i grupper av 12 tegn som printes under hverandre.

Matrisen kan være av en hvilken som helst størrelsesorden, og direktivet vil om nødvendig skrive den ut over flere sider. Disse sidene kan eventuelt limes sammen etterpå.

Istedenfor liste3 og liste4 kan man anvende vektorer, matriser med én kolonne, rekordsett med ett elementærfelt, og enhver kombinasjon av disse. (Istedenfor liste2 kan man angi et hvilket som helst objekt med innhold, dvs. ikke NAVN, TALL, og FELT.)

Direktiv PRINTNRADER

(PRINT N RADER)

Form: PRINTNRADER fra matrise skal de parameter første radene printes.

Innobjekter: matrise, parameter

Utobjekt: -

Dette direktivet vil gi utskrift av et oppgitt antall rader av en matrise. Antallet rader er gitt ved parameter. Hvis parameter er større enn eller lik antall rader som finnes i matrise vil hele matrisen bli skrevet ut. Hvis matrise har fått en liste av kolonnenavn knyttet til seg (f.eks. ved NAVNGIKOL) vil også disse navnene bli skrevet ut. Matriseradene vil bli skrevet ut en full rad ad gangen, over så mange printlinjer som er nødvendig.

Direktivet er spesielt beregnet til bruk ved manuell kontroll av datamatiser til regresjonsanalyser. Det kan benyttes før eller etter at matrisen har brukt med NAVNGIKOL, men det kan ikke benyttes til utskrift av de matriser med spesialstatus (ramatiser). Etter at en ramatrise har vært brukt med RASLUTTDATA vil den imidlertid bli en vanlig matrise og kan da brukes som argument til PRINTNRADER.

Direktiv PRINTVERDI

(PRINT er VERDIen av et objekt)

Form: PRINTVERDI objekt.

Dette direktivet printer ut verdier av et vilkårlig objekt. Navnet på objektet angis, og verdien printes i standardformat. Direktivet er fortrinnsvis beregnet på små objekter. Utskrifter begynner ikke på ny side.

Med visse objekter printes ut linjenumre.

Direktiv SENDPRINT

(SEND PRINT til utskriftfile)

Form: SENDPRINT til file1/HER.

(BARE TILGJENGELIG PÅ UNIVAC 1108-VERSJON)

File1 er et vilkårlig NAVN eller NAVNORD valgt og deklarerert av brukeren, mens HER er et STANDARD-objekt.

Eksempel på bruk:

ARKIVTAPE STANDARD TAPEABC.NAVN BRUKERFILE.PRINTVERDI A.SENDPRINT TIL BRUKERFILE.PRINTVERDI B.SENDPRINT TIL HER.PRINTVERDI C.SLUTTPROD.

Forutsatt at objektene A, B, og C ligger på arkivtape TAPEABC, vil denne kjøringen resultere i at objektene A og C vil bli skrevet ut over linjeskriveren. Objekt B (pluss hvilken som helst utskrift som kommer mellom de to SENDPRINT-setninger) blir skrevet med overskrift på filen som heter BRUKERFILE. SENDPRINT vil sende bort alle utskrifter til én og samme file, inntil SENDPRINT HER er påtruffet.

Mens direktiv I-0 (se ovenfor) vil opprette automatisk en ny file for hvert objekt, vil SENDPRINT bare sende print til en file som brukeren allerede har opprettet før DATSY-teksten eksekveres, slik:

"ASG,T BRUKERFILE, F2/1/TRK/1000

"XQT DATSY

I det samme runet etter DATSY-eksekveringen kan BRUKERFILE printes ut i redigert form (på UNIVAC 1108 ved hjelp av LISTUT-prosessoren). Dette er hovedvitsen med direktivet.

For å bruke LISTUT-prosessoren på UNIVAC 1108 har man følgende instruks:

Skriv først styrelinjen:

"LISTUT,G BRUKERFILE.

Skriv i de etterfølgende linjene ordrer av format:

P n = Skriv n antall linjer

N n = Hopp over de n følgende linjer
 F string = Finn karakterkombinasjonen "string".
 PL n = Kun de n første karakterer av hver linje skal skrives ut.

Eksempel:

```
"LISTUT BRUKERFILE
N 100
F PRINTVERDI APLUSS
P 40
F PRINTVERDI WSP
P 10
PL 45
P 20
```

I eksemplet skal LISTUT-prosessoren printe ut på linjeskriveren de første 40 linjer som begynner med PRINTVERDI APLUSS, som samtidig må begynne etter de første 100 linjer av print i BRUKERFILE (det kan ha forekommet en annen utskrift av APLUSS med forskjellige verdier innen de første 100 linjene i BRUKERFILE). Så går man fram og finner teksten PRINTVERDI WSP. Herfra printes det ut bare 10 linjer for å få et inntrykk av resultatene. Man setter på printlinjebreddene lik 45 tegn bare, slik at senere tegn ikke blir printet ut i det hele tatt, og fortsetter da i 20 linjer.

Prosessorene ED og EDIT på UNIVAC 1108 kan benyttes som alternativ, for å liste ut filen.

Filen (f.eks. BRUKERFILE) kan være katalogisert på forhånd (eller bruk styresetningen:

```
"ASG,CP BRUKERFILE, F2/1//200
```

i eksemplet ovenfor). Da kan den skrives ut i et senere run, f.eks. fra terminal.

C-2.2 Kontroll- og etableringsdirektiver

<u>Direktiv</u>	<u>Status pr. 1/1-74</u>
ANTDOBLIST	virker
DOKUMENTER	ikke testet
GENOBJEKT	virker
GENMAT (se C-2.3)	-
GENREK (se C-2.3)	-
IDENTIFISER	virker
KONVMATREK (se C-2.3)	-
KONVREKMAT (se C-2.3)	-
LET (se C-2.1)	-
MATRISEDIM	virker
OPPLYSOBJEKT (se C-2.1)	-
PAKKINN	virker
PAKKUT	virker
SAMMENLIGN	virker
SUBLISTGEN	virker
TAVEKKDOBB	virker
TRANSFORMER (se C-2.4)	-

Direktiv ANTDOBLIST(ANTall DOBbeltforekomster i
LISTE eller vektor bestemmes)Form: ANTDOBLIST liste/vektor.

Antall dobbeltforekomster av elementer i liste eller vektor bestemmes og blir skrevet ut.
Dette antallet blir et attributt til liste/vektor.

De attributter som et objekt får, blir en del av beskrivelsen av dette, og settes inn i data-arkivet når objektet blir skrevet ut på magnetbånd.

Direktiv DOKUMENTER

(DOKUMENTERer et objekt)

Form: DOKUMENTER objekt med liste/navnord/rekordsett.

Objekt er et vilkårlig objekt. Liste inneholder dokumentasjon om objekt og behøver ikke å ha noen bestemt form, sett fra en rent teknisk synsvinkel. Når objekt senere skrives i DATSY-arkivet (på en arkivtape) følger da liste med uten at den er blitt nevnt i en SKRIV-setning. Objektets dokumentasjon følger da objektet på magnetisk medium. Hvilken liste det er som dokumenterer et bestemt objekt, er angitt i utskriften fra et DATSY-arkiv. Denne opplysning blir også gitt av direktivet OPPLYSOBJEKT.

Utformingen av liste kan bestemmes konvensjonelt av brukere.

Se også direktiv PRINTMAT, avsnitt C-2.1.

Teknisk merknad: Navnet til liste blir fastsatt av direktivet som attributt 26 til objekt.

Direktiv GENOBJEKT

(GENERer OBJEKT)

Form: GENOBJEKT navnord bestemmer type, parameter1 og parameter2, dimensjonene, og parameter3 er vanligvis en verdi som inngår i objekt på en eller annen måte.

Direktivet lager mange forskjellige typer objekter, bestemt av navnord. Objekt er altså ut-objekt for dette direktivet og er matrise/vektor/parameter. Som navnord benyttes STANDARD-objekter, tillatte navnord er KMATRISSE/KUMULER/TESTDATA/IDENTSNUDD.

Klasse til objekt blir bestemt ved deklarasjon av brukeren, og den må stemme overens med det alternativ spesifisert ved navnord.

(i) navnord = KMATRISSE (Konstant MATRISSE, vektor eller parameter)

Det genereres en parameter1 x parameter2 matrise, med hvert element lik parameter3 i verdi. Standardverdier for parametrene er oppbevart i arkivet. Matrisen blir en flytendetalismatrise.

Det kan være greit å bruke GENOBJEKT KMATRISSE sammen med direktiv MATRISSEDIM (se dette), slik at verdier for parameter1 og parameter2 kan fås fra et annet objekt. Resultatmatrisen (objekt) fra GENOBJEKT KMATRISSE kan brukes f.eks. til å summere rader eller kolonner eller til å addere et tall til hvert element i en matrise. Dette diskuteres videre under de neste alternativer.

(ii) navnord = KUMULER (KUMULER matrise eller vektor)

GENOBJEKT KUMULER lager en kumuleringsmatrise med følgende struktur:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

dvs. med 1-tall på og under hoveddiagonalen. Når en matrise premultipliseres med en kumuleringsmatrise blir resultatet kumulering over kolonnene i matrisen som med direktiv ADDERKUMULA (se C-2.4):

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix}$$

kumulerings- matrise kumulert
matrise matrise

Ved å postmultiplisere med denne spesielle matrisen får man en lignende virkning på radene, sådan:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix}$$

I dette siste tilfelle har man også brukt en annen spesiellaget matrise for å vende kolonne- rekkefølgen om. (Denne matrisen beskrives nedenfor.)

Med navnord = KUMULER blir dimensjonene til kumuleringsmatrisen bestemt som før av parameter1 og parameter2 (som normalt fås fra direktiv MATRISEDIM). Parameter1 og parameter2 må iallfall være like. Parameter3 skal man normalt sette lik 1.

(iii) navnord = TESTDATA (TESTDATA blir laget)

En matrise, vektor, eller parameter blir laget med følgende verdier:

```

1.001  1.002  1.003  1.004  .....  1.555  .....  1.999
2.001  2.002  .....
3.001  3.002  .....
.      .
.      .
.      .
.      .
12.001 12.002
.      .
.      .
.      .

```

Disse testdataene kan benyttes ved testing av direktiver for å kunne gjenkjenne data etter manipulering i direktiver som uttestes. Poenget er altså at data er valgt slik at rad og kolonne framgår av verdien av matriseelementer som genereres.

Varsel: en kvadratisk matrise laget på denne måten er singular.

Parameter3 blir ignorert men noe må man spesifisere (f.eks. STANDARD-objektet 1).

(iv) navnord = IDENTSKUDD (IDENTitetsmatrise blir laget SKUDD)

Det lages en kvadratisk matrise som er en speilvendt enhetsmatrise (se eksempel nedenfor). Den brukes for å invertere rekkefølge av rader eller kolonner, ved henholdsvis pre- og postmultiplisering. Strukturen er altså følgende:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Eksempel:

$$\begin{bmatrix} 1.1 & 1.2 & 1.3 & 1.4 \\ 2.1 & 2.2 & 2.3 & 2.4 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1.4 & 1.3 & 1.2 & 1.1 \\ 2.4 & 2.3 & 2.2 & 2.1 \end{bmatrix}$$

og

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1.1 & 1.2 \\ 2.1 & 2.2 \\ 3.1 & 3.2 \\ 4.1 & 4.2 \\ 5.1 & 5.2 \end{bmatrix} = \begin{bmatrix} 5.1 & 5.2 \\ 4.1 & 4.2 \\ 3.1 & 3.2 \\ 2.1 & 2.3 \\ 1.1 & 1.2 \end{bmatrix}$$

(v) navnord = STIGENDE (matrise/vektor lages med stigende tall) (ikke implementert)

Parameter3 spesifiserer begynnelsetallet, f.eks. 1. Tallene blir konvertert automatisk til flytende som før.

Eksempel 1:

ARKIVTAPE STANDARD.

GENOBJEKT STIGENDE 4 3 3 MAT.

MAT vil inneholde:

$$\begin{bmatrix} 3 & 4 & 5 \\ 6 & 7 & 8 \\ 9 & 10 & 11 \\ 12 & 13 & 14 \end{bmatrix}$$

Eksempel 2:

ARKIVTAPE STANDARD.

GENOBJEKT STIGENDE 3 1 10 VEK.

VEK vil inneholde:

$$\begin{bmatrix} 10 \\ 11 \\ 12 \end{bmatrix}$$

Direktiv IDENTIFISER

(IDENTIFISERer radene eller kolonnene i en matrise)

Form: IDENTIFISER RADENE/KOLONNENE i matrise med liste.Innobjekter: RADENE/KOLONNENE/ELEMENTENE, matrise/liste/parameter/rekordsett, liste/navnord, parameter/vektor/rekordsett.

Utobjekt: -

Normalt brukes dette direktivet for å knytte en alfanumerisk liste til en matrise, der hvert element i liste identifiserer én rad (eller én kolonne) i matrise. RADENE og KOLONNENE er STANDARD-objekter (se B-1), deklarerert som NAVNORD. I tilfelle RADENE er oppgitt vil direktivet sjekke om antall rader i matrise stemmer med antall elementer i liste, og tilsvarende når brukeren oppgir KOLONNE. Hvis man vil tilknytte én radliste pluss én kolonneliste, må direktivet brukes to ganger.

Når matrise blir skrevet på en arkivtape vil sådanne tilknyttede lister følge med, uten at de blir eksplisitt nevnt i skrivsetningen. Hvis matrise blir brukt i et senere DATSY-program hvor arkivtapen er blitt deklarerert vil de tilknyttede listene også være tilgjengelige (i likhet med alle andre objekter på arkivtapen som eksplisitt er skrevet).

Istedenfor matrise kan andre objekter benyttes som angitt ovenfor.

Formen: IDENTIFISER ELEMENTENE i vektor med liste.

kan også anvendes, der ELEMENTENE også er STANDARD-objekt.

RADER kan alltid skrives istedenfor RADENE, likeledes KOLONNER istedenfor KOLONNENE.

Direktivet kan også brukes for å identifisere radene eller kolonnene i matrise eller andre objekter med annet enn listeelementer, f.eks. med vektor av flytendetail som kan tolkes som nedre intervallgrenser. Se under innobjekter ovenfor.

Det er tillatt å benytte vektorer med flytende tall enten med (n) eller (n+1) elementer, hvor (n) er antall rader/kolonner i matrise. Det (n+1)-te elementet kan da tolkes som en øvre grense for det siste intervallet. Vektorer kan også tilknyttes vektorer med tilsvarende tolking.

Radene og kolonnene i matriser, vektorer og parametere kan identifiseres, og linjene i et rekordsett kan også identifiseres ved å bruke RADENE.

Til identifisering er det tillatt: lister, navnord, parametere, vektorer, og rekordsett med ett elementærfelt.

Teknisk merknad: Navnet på liste blir tilknyttet matrise som attributt nr. 24 (kolonner), 25 (rader), og 26 (dokumentasjonsliste).

Direktiv MATRISEDIM

(få ut MATRISEDIMensjoner)

Form: MATRISEDIM kopiér dimensjonene til matrise over til parameter1 og parameter2.Innobjekt: matrise/vektor/liste/rekordsett.Utobjekter: parameter1, parameter2.

Antall rader blir satt i parameter1, og antall kolonner i parameter2, som da blir heltallsparametere. (De kan bl.a. brukes i direktivene DELHORISONT, DELKOLVIS, GENOBJEKT, og LET.)

Dette direktivet er svært viktig. Det gir mulighet for å lage dimensjonsuavhengige DATSY-tekster. F.eks. kan en tenke seg en makroøkonomisk modell programmert der antall aktiviteter, sektorer, og varer blir bestemt av data og ikke av program, for å unngå feil ved forandringer.

Med vektor, liste, og rekordsett får man i parameter1 antall elementer, listeelementer, og rekorder. I parameter2 får man da 1 for vektor, og antall ord i bredden med liste og rekordsett. Denne siste er mest av teknisk interesse.

(PAKK matrise INN)

Direktiv PAKKINN

Form: PAKKINN pakker matrise som rekordsett.

Dette direktivet "pakker" en matrise ved å generere et rekordsett med en rekord for hvert ikke-null element i matrise. Ved dette oppnås en mer effektiv lagring og overføring dersom matrise har mange nullelementer (f.eks. 80-90 pst. nullelementer).

rekordsett kan beskrives som

rekordsett PAKKMAT.

der PAKKMAT er et STANDARD-objekt (se C-1). PAKKMAT inneholder tre TALL for linjenummer, kolonnennummer og verdifelt. rekordsett kan også beskrives uten bruk av PAKKMAT og beskrivelsen må da bestå av tre TALL.

Ved bruk av PAKKMAT må det undersøkes at de direktiver der matrisen skal inngå, godtar en pakket matrise som innobjekt. (Se også direktivet PAKKUT).

Direktiv PAKKUT

(PAKK matrise UT igjen)

Form: PAKKUT pakker rekordsett ut til matrise igjen.

Se beskrivelsen av direktiv PAKKINN. PAKKUT brukes når den innpakkede representasjon ikke lenger ønskes. Nullelementer blir representert igjen på vanlig måte.

(rekordsett er forutsatt beskribert som forklart i beskrivelsen av PAKKINN.)

Direktiv SAMMENLIGN

(SAMMENLIGN objektverdier)

Form: SAMMENLIGN objekt1 med objekt2 m.h.t.verdi, og STOPP/FORTSETT hvis det er noe som ikke stemmer.

objekt1 og objekt2 kan være av forskjellig klasse og ha forskjellige beskrivninger m.v. (og selvsagt kan de også ha forskjellige navn). Inntil 15 forskjeller skrives ut vanligvis, mens 250 forskjeller kan skrives ut når K-opsjon er fastsatt (se B-6.4). Flytende tall må stemme med hverandre inntil en konvensjonell verdi på 1×10^{-20} .

STOPP og FORTSETT er STANDARD-objekter (se C-1), deklarerert som navnord.

Direktiv SUBLISTGEN

(SUBLISTGENerering)

Form: SUBLISTGEN genererer subliste som snittet av liste1 og liste2. (Istedenfor liste1 og/eller liste2 vil direktivet godta et rekordsett, dersom dette bare har ett elementærfelt og dette er alfa-numerisk.)

Dette er tenkt som den naturlige måten å lage sublister til bruk i sublistedirektiver (se C-2.5). Det er verdt å merke seg at sublister ikke kan leses inn fra hullkort. Ved å spesifisere i liste1 navn over hva man allerede har (f.eks. i radene av en matrise) og i liste2 navn over hvilke av disse man vil ha tak i, kan man generere en subliste som er egnet til bruk f.eks. i direktivet EKSRAD (se C-2.5).

Det blir kontrollert at ingen dobbeltforekomster finnes i listene. Hvis dobbeltforekomster oppdages, stopper kjøringen og feilmelding gis.

Direktiv TAVEKKDOBB

(TA VEKK DOBBeltforekomster)

Form: TAVEKKDOBBeltforekomster i listel/vektor1 og sett resultatet i liste2/vektor2.
 (Istedenfor listel/vektor1 og/eller liste2/vektor2 vil direktivet godta et rekordsett, dersom dette bare har ett elementærfelt, sml. SUBLISTGEN ovenfor.)

Direktivet vil fjerne multiple forekomster av verdier i en liste (vektor) og generere en ny liste (vektor) uten dobbeltforekomster.

Første forekomst av en verdi vil bli bevart.

Klassen til utobjektet er styrt av brukerens deklarasjon. vektor1 og vektor2 må være heltallsvektorer i tilfelle.

Hvis innobjektet er alfanumerisk må utobjektet også være det, og hvis innobjektet er et heltallsobjekt må utobjektet også være et heltallsobjekt. (Direktivet stopper med feilmelding hvis f.eks. vektor2 er utobjekt og innobjektet er listel.)

Direktivet kan brukes f.eks. etter at man har ekstrahert en liste fra et rekordsett der liste-elementene brukes som rad- eller kolonnenavn til en matrise. Det passer naturlig sammen med direktivet SORTER (se C-2.6). Sammenlign også med direktivet ANTDOBLIST (se dette).

(Når en nypunchet liste har uforutsette dobbeltforekomster kan det være punchefeil -- i så fall bør man rette på feilene og ikke bruke TAVEKKDOBB. Ellers kan det hende at feilpunchede liste-elementer forsvinner helt.)

C-2.3 Konverteringsdirektiver

<u>Direktiv</u>	<u>Status pr. 1/1-74</u>
CNVADD*	virker
GENMAT	ikke implementert
GENREK	ikke implementert
I-0 (se C-2.1)	-
KONVMATREK	virker
KONVREKMAT	virker
LESBCDTAPE	virker
PAKKINN (se C-2.2)	-
PAKKUT (se C-2.2)	-
SKRIVBCDTAPE	virker

Program CNVADD

Form: "XQT,opsjoner CNVADD

Programmet 'CNVADD' konverterer IBM 360 BCD Fortran-tape til UNIVAC 1108 element tape i 'COPIN'-format.

Input-tapen assignes slik: "ASG,TIEH SSB,C,XXXX

Output-tapen assignes slik: "ASG,T ADDTAPE,C,YYYY

XXXX og YYYY er nr. på tapene.

* CNVADD er ikke et vanlig DATSY-direktiv, men et eget program som kan brukes for konvertering på UNIVAC 1108.

Det forutsettes at input-tapen har fast blokkstørrelse. Hvis en blokk avviker fra første blokk, kommer det beksjed om dette. For hver gruppe på 80 karakterer på input-tapen, lages det ett kortbilde på output-tapen.

Datakort: Fra kolonne 1 punches maks. 12 alfanumeriske karakterer som blir navnet på elementet. Hvis datakort mangler, vil navnet 'OBJEKT' bli brukt. I kolonne 13 og utover punches eventuelt antall kortbilder som skal printes, se (i) nedenfor.

Tillatte opsjoner i XQT-kortet er 'P', 'S' og 'B':

- (i) P-opsjon vil resultere i printing av hvert kortbilde på output-tapen, med nummerering. Hvis man ønsker å printe bare de n første kortbildene, må man sammen med P-opsjon spesifisere tallet n på datakortet. Det punches f.o.m. kolonne 13.
- (ii) S-opsjon vil resultere i skipping av 1 file på input-tapen før konverteringen begynner (f.eks. label file).
- (iii) B-opsjon vil resultere i at den siste blokken på input-tapen ikke vil komme med på output-tapen. Programmet kan eksekveres flere ganger. For hver eksekvering blir en file på input-tapen konvertert til en file - inneholdende ett element - på output-tapen.

Med opsjoner TIEH på "ASG-kortet menes bl.a. at input-tapen skal være BCD kode, 'even parity', og 800 bits per inch.

Direktiv GENMAT

(GENererer MATrise fra rekordsett)

Form: GENMAT rekordsett brukes for å generere matrise med listel/vektor1 og liste2/vektor2.

Innobjekt: rekordsett

Utoobjekter: matrise, listel/vektor1, liste2/vektor2

Dette direktivet ligner på direktiv KONVREKMAT, men er mye enklere. Med KONVREKMAT oppgis rad- og kolonneliste (eller vektorer), mens GENMAT vil produsere disse selv som utobjekt. rekordsett må ha den struktur som er nærmere beskrevet under KONVREKMAT.

GENMAT vil sette klassen til utobjektene slik at de stemmer med feltene i rekordsett, selv om brukeren har deklarerert annerledes. (Disse objektene kan således anta 4 mulige klassekombinasjoner avhengig av objektklasse for de to første elementærfeltene i rekordsett.) Hvis et av de to første elementærfelt i rekordsett er alfanumerisk (deskribert med NAVN), blir det tilsvarende utobjekt en liste, hvis TALL (heltall), en heltallsvektor. Hvis et av de to første elementærfelt i rekordsett er flytendetail gis feilmelding om dette og direktivet stopper.

Etter GENMAT blir listel/vektor1 og liste2/vektor2 tilknyttet matrise som attributt. Det er således ikke nødvendig å bruke direktivet IDENTIFISER etterpå.

Direktiv GENREK

(GENererer REKordsett fra matrise)

Form: GENREK matrise konverteres til et rekordsett.

Innobjekt: matrise

Utoobjekt: rekordsett

For en beskrivelse av strukturen til rekordsett, se direktivet KONVMATREK. Forskjellen mellom direktiv GENREK og KONVMATREK er bare det at med KONVMATREK nevnes en radliste og en kolonnaliste eksplisitt, mens med GENREK nevnes ikke disse i direktivsetningen.

Sekvensen:

IDENTIFISER RADENE i matrisel med liste1.

IDENTIFISER KOLONNENE i matrisel med liste2.

GENREK fra matrisel generer rekordsett1.

vil ha samme virkning som setningen:

KONVMATREK matrisel med liste1 og liste2 konverteres til rekordsett1.

(Men i det første eksemplet vil liste1 og liste2 fortsatt være tilknyttet matrisel - se direktiv IDENTIFISER.)

Når matrise mangler enten radliste eller kolonneliste, eller begge, vil GENREK stoppe med feilmelding.

Direktiv KONVMATREK

(KONVerterer MATrise til REKordsett)

Form: KONVMATREK tar matrisel med liste2 og liste3 og produserer fra disse et rekordsett4 som representerer matrisen.

Innobjekter: matrisel/vektor1/parameter1, liste2/vektor2/rekordsett2, liste3/vektor3/rekordsett3.

Utobjekt: rekordsett4.

Direktivet er laget som det motsatte av KONVREKMAT, som kan brukes for å kunne konvertere rekordsett4 tilbake til matrisel igjen.

rekordsett4 lages med tre elementærfelt, hvorav de to første spesifiserer posisjonen til et element i matrisen med henholdsvis radnavn og kolonnenavn fra liste2 og liste3. Det tredje elementærfeltet inneholder verdien til elementet (forutsatt flytende). En rekord lages for hvert element i matrisen bortsett fra nullelementer. Elementene konverteres radvis fra venstre til høyre, slik at rekordsett4 vil være sortert etter nøkkelen radnavn x kolonnenavn.

Navnet til matrisel blir oppbevart som 'historikk-attributt' til rekordsett4. Direktiv OPPLYSOBJEKT kan brukes da for å få vite hvilken matrise et rekordsett er evt. konvertert fra.

vektor2 og vektor3 må være heltallsvektorer i tilfelle. rekordsett2 og rekordsett3 må ha bare ett elementærfelt hvis de brukes. Det er tillatt å bruke kombinasjoner, f.eks. liste2 sammen med vektor3.

Direktiv KONVREKMAT

(KONVerterer REKordsett til MATrise)

Form: KONVREKMAT rekordsett1 med liste1/vektor1 og liste2/vektor2 blir konvertert til matrisel og tellingsmatrisen matrise2.

Innobjekter: rekordsett1, liste1/vektor1, liste2/vektor2

Utobjekter: matrisel, matrise2

rekordsett1 må ha en spesiell struktur med tre elementærfelt, hvorav de to første angir rad- og kolonneposisjoner i matrisel, mens det tredje angir en verdi som skal settes i matrisel i denne posisjon.

Denne verdien må være de flytendetail. Rad- og kolonneposisjonene angis ved hjelp av alfanumeriske opplysninger fra de to listene, listel og liste2, i de to første elementærfeltene. Radrekkefølgen i matrisel bestemmes av listel/vektor1.

Dimensjonene til matrisel blir bestemt av dimensjonene til radlisten listel og kolonnenlisten liste2. Det samme gjelder for matrise2. Dersom rekordsettl inneholder flere rekorder som refererer seg til samme matriseposisjon blir disse aggregert, dvs. verdiene addert. Direktivet vil gi melding hvis dette skjer, for at man skal kunne sjekke at det ikke skyldes feil. Direktivet vil bruke tellingsmatrisen til dette formålet. Brukeren kan også bruke den til andre formål. matrise2 er en flytendetailsmatrise som viser det totale antall elementer lest inn i hver posisjon i matrisel.

Dersom det er oppgitt vektor1 (vektor2) istedenfor listel (liste2) tolkes den som en vektor av nedre intervallgrenser, hvor siste element er den øverste intervallgrense. Dimensjonene til matrisel og matrise2 blir da én mindre enn dimensjonen til en slik vektor. Verdiene i vektor1 (vektor2) må være stigende. Hver nedregrenseverdi inkluderes i intervallet som den definerer. vektor1 (vektor2) må være flytendetailvektorer. (Denne bruken av direktivet er ikke testet.)

Direktivet fungerer raskest når listel/vektor1 og radnavnene/radverdiene i rekordsettl er sortert tilsvarende. I tilfelle dette ikke gjelder virker direktivet likevel, men ikke så raskt. U-sortert og med stor matrisel vil direktivet være håpløst tungvint i drift. (I verste fall vil direktivet stoppe, og gi feilmelding. Likevel kan man da forsere kjøringen ved å bruke "XQT,S-kort, på UNIVAC 1108. Andre eksekveringsopsjoner er innført for å utnytte en tom ('dedicated') UNIVAC 1108.)

Direktiv LESBCDTAPE

(LESer BCD-TAPE)

Form: LESBCDTAPE les magnetbånd eller Fastrand file (tapestasjon) nummer parameter med formatet format og sett resultatet i objekt.

(BARE TILGJENGELIG PÅ UNIVAC 1108-VERSJON)

Innobjekter: parameter, format

Utobjekt: objekt

Dette direktivet er innført for å kunne lese inn i DATSY objekter som er skrevet i FORTRAN på IBM 360. Et magnetbånd produsert i FORTRAN på IBM 360 må først konverteres på UNIVAC 1108 med programmet MKONFT (se bruksanvisning annetsteds). Det blir da konvertert til Fielddata, dvs. til en Fastrand-file eller magnetbånd som kunne vært skrevet i UNIVAC FORTRAN V.

parameter er et heltall som magnetbåndet eller Fastrandfilen får ved styrekort før DATSY-programmets begynnelse. Tillatte verdier av parameter er 2, 3, 4, 15-29.

format blir oversatt til det FORTRAN-formatet som tapen er skrevet i. (Hvis brukeren ønsker at et elementærfelt skal ha flere enn 6 karakterer, f.eks. 15, da skal han skrive A15 for dette leddet i formatet. Hvis brukeren istedenfor skriver 2A6,A3 blir dette da tolket som tre forskjellige elementærfelt.)

objekt får den klassen brukeren deklarerer. Hvis objekt er et rekordsett, må det beskrives, og beskripsjonen må stemme med format m.h.t. elementærfelt. For å lese inn flytendetailsmatrise, vektor, eller parameter må objekt være fra før av brukt som et slikt objekt, f.eks. ved setningene:

ARKIVTAPE standard.

GENOBJEKT KMATRISE 1 1 1 objekt.

Direktiv SKRIVBCDTAPE

(SKRIV BCD-TAPE)

Form: SKRIVBCDTAPE på magnetbånd assignet til båndstasjon parameter skrives med format objektet objekt.

Innobjekter: parameter, format, objekt

Utobjekt: -

Dette direktivet skriver objekt ut på et magnetbånd i Fieldata eller BCD format ved hjelp av FORTRAN WRITE, avhengig av opsjoner på et "ASG-kort, slik at magnetbåndet med objekt deretter kan leses i FORTRAN på UNIVAC 1108 eller eventuelt IBM 360. (Det siste forutsetter en ytterligere konvertering til BCD-format som kan utføres på IBM 360.) Objektnavnet skrives automatisk i posisjonene 1-12 i hver rekord på magnetbåndet, unntatt i den siste. Direktivet likner svært på LESBCDTAPE, som utfører den motsatte funksjon.

format kommer under DATSY's regler for formater, og må således stemme med beskripsjonen til objekt i tilfelle objekt er et rekordsett. Tillatte verdier på parameter, som er forutsatt heltall, er 2, 3, 4, eller 11-29.

(I motsetning til FORTRAN vil DATSY ikke rewinde magnetbåndet etter eksekvering av et program. Derfor kan "MARK-kort brukes for å skrive hardware end-of-file (EOF) på magnetbåndet. Likevel avsluttes skrivingen ved hver bruk av direktivet med et konvensjonelt merke, nemlig ved at de første 6 posisjoner i rekorden inneholder '999999' (alfanumerisk).

Eksempel på bruk:

"ASG, TIEH 2,C,8881 . (assigning av magnetbåndet)

"XQT DATSY

ARKIVTAPE STANDARD TAPEA.

SKRIVBCDTAPE på enhetnummer 2 med FORMA skriv objekt A.

SLUTTPROD.

"MARK 2.

("MARK-kortet er ikke nødvendig men kan være fordelaktig.)

Objekt A og formatet FORMA er forutsatt å befinne seg på arkivtape TAPEA. De angitte opsjoner på "ASG-kortet er nødvendig for å skrive en BCD tape med 800 b.p.i. M-opsjon for "ASG-kortet virker ikke i level 31 av EXEX-8.

BCD-tapen må ikke forveksles med en arkivtape, eksterntape eller en fremmedtape (se B-2.5). BCD-tapen kan monteres og frigjøres midt i et DATSY-program ved bruk av direktivet I-O (se C-2.1).

C-2.4 Matematiske direktiver

<u>Direktiv</u>	<u>Status pr. 1/1-74</u>
ABS	virker
ADDER	virker (✕)
ADDERKUMULA	virker
BIGRADSKALER	ikke testet
DIAGONALISER	virker
DIVIDER	virker
ELEMULT	virker
ENDRING	virker
GENOBJEKT (se C-2.2)	-
IMINUSA	virker

✕) Fortsatt beheftet med visse svakheter.

Direktiv (forts.)Status pr. 1/1-74 (forts.)

INVERS	virker
INVERTER	virker (*)
KOLADDER	virker
KOLONNE-LAG	virker
KOLSKALER	virker
MULTIPLISER	virker
PAKKINN (se C-2.2)	-
PAKKPAKKMULT	virker
PAKKUT (se C-2.2)	-
RADADDER	virker
RADSKALER	virker
SOLVE	virker (*)
SUBTILBAKE	virker
SUBTRAHER	virker (*)
SUMKOL	virker
SUMRAD	virker
TRANSFORMER	virker
TRANSPONER	virker
TRIANGULER	virker
TRITILINVERS	virker
VANPAKKMULT	ikke testet

ABS

Direktiv ABS

(ABSoluttverdi)

Form: ABS absoluttverdien av matrise1/vektor1/parameter1 plasseres i matrise2/vektor2/parameter2.

Innobjekt: matrise1/vektor1/parameter1

Utobjekt: matrise2/vektor2/parameter2

Absoluttverdi beregnes på vanlig måte. Negative elementer skifter fortegn. Positive elementer kopieres direkte. Null forblir null.

Utobjektet må ha samme klasse som innobjektet.

ADDER

Direktiv ADDER

(ADDERer numeriske objekter)

Form: ADDER parameter1/vektor1/matrise1 + parameter2/vektor2/matrise2 = parameter3/vektor3/matrise3.

Innobjekter: parameter1/vektor1/matrise1, parameter2/vektor2/matrise2

Utobjekt: parameter3/vektor3/matrise3.

Direktivet gjelder for både hel- og flytendetallsobjekter. Utobjektet får klasse identisk med klassen til første innobjekt etter at dette direktivet er brukt.

*) Fortsatt beheftet med visse svakheter.

Direktiv ADDERKUMULA

(ADDERing og KUMULering av rader)

Form: ADDERKUMULA adder og kumuler radene i matrisel/vektor1 og sett resultatet i matrise2/vektor2.

Innobjekt: matrisel/vektor1

Utobjekt: matrise2/vektor2

Direktivet adderer rader kumulativt nedover i matrisen.

F.eks.:

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 6 & 7 & 8 \\ 1 & 10 & 11 & 12 \\ 1 & 14 & 15 & 16 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 8 & 10 & 12 \\ 3 & 18 & 21 & 24 \\ 4 & 32 & 36 & 40 \end{bmatrix}$$

vektor1 blir addert og kumulert som om den var en matrise med én kolonne. Brukeren deklarerer klassen på utobjektet. (Direktivet stopper med feil melding hvis den var urimelig.)
Se også direktiv GENOBJEKT.

Direktiv BIGRADSKALER

Form: BIGRADSKALER matrisel deles i kolonner etter vektor1 og de enkelte kolonner brukes til å radskalere matrise2 delt etter vektor2 (heltall) og resultatet settes i matrise3.

Innobjekter: matrisel, vektor1, matrise2, vektor2

Utobjekt: matrise3

Direktivet utfører radskaleringsoverasjon (se direktiv RADSKALER) på deler av matrise2 delt etter kolonner.

vektor1 er en heltallsvektor med like mange elementer som antall kolonner i matrisel og hvert element er lik 1. (vektor1 kan genereres med direktiv GENOBJEKT der kolonnedimensjonen av matrisel finnes med direktiv MATRISEDIM.)

vektor2 er en heltallsvektor av samme dimensjon som vektor1. Elementene i vektor2 angir antall kolonner i de respektive deler av matrise2 som skal radskaleres av den tilsvarende kolonne i matrisel. Summen av elementene i vektor2 må derfor være lik antall kolonner i matrise2.

Se forøvrig direktiv RADSKALER i dette avsnitt og direktiv STORADSKALER i avsnitt C-3.1.

Direktiv DIAGONALISER

(DIAGONALISERer en vektor)

Form: DIAGONALISER vektor til en matrise som har vektoren som hoveddiagonal, og null ellers.

Innobjekt: vektor

Utobjekt: matrise

I mange tilfelle kan direktivene RADSKALER og KOLSKALER erstatte bruken av DIAGONALISER, nemlig der DIAGONALISER blir etterfulgt av en pre- eller postmultiplisering av matrise. RADSKALER og KOLSKALER vil være vesentlig mer effektive i slike tilfelle og bør derfor foretrekkes.

Direktiv DIVIDER

(DIVIDERer elementvis)

Form: DIVIDER alle elementene i matrisel/vektor1/parameter1 divideres med de tilsvarende elementene i matrise2/vektor2/parameter2, og resultatet settes i matrise3/vektor3/parameter3.

Innobjekter: matrisel/vektor1/parameter1, matrise2/vektor2/parameter2

Utoobjekt: matrise3/vektor3/parameter3

Objekter av forskjellige klasser kan forekomme, både som innobjekter og utobjekt, men de to innobjektene må være identisk m.h.t. dimensjon. Det går f.eks. an å dividere elementene i vektor1 med tilsvarende elementer i matrise2 dersom matrise2 har bare én kolonne, og samme antall rader som vektor 1 har elementer.

Klassen til utobjektet bestemmes av brukeren selv ved deklarasjon. Hvis den er urimelig, vil direktivet stoppe og skrive ut feilmelding om dette.

Direktiv ELEMULT

(ELEMENTvis MULTiplisering)

Form: ELEMULT. Tilsvarende elementer i matrisel/vektor1/parameter1 og matrise2/vektor2/parameter2 multipliseres sammen og resultatet settes i matrise3/vektor3/parameter3.

Innobjekter: matrisel/vektor1/parameter1, matrise2/vektor2/parameter2

Utoobjekt: matrise3/vektor3/parameter3

Det blir testet at innobjektene har samme dimensjon. Direktivet fastsetter dimensjonene til utobjektet lik dimensjonene til innobjektene.

Direktiv ENDRING

(ENDRINGer mellom elementer)

Form: ENDRING mellom elementene i vektor1/matrisel blir satt inn i vektor2/matrise2, mens prosentendringer blir satt inn i vektor3/matrise3.

Innobjekt: vektor1/matrisel

Utoobjekter: vektor2/matrise2, vektor3/matrise3

$$\begin{aligned} \text{Hvis } \underline{\text{vektor1}} &= (x_1, x_2, \dots, x_n) && \text{blir} \\ \underline{\text{vektor2}} &= (x_2 - x_1, x_3 - x_2, \dots, x_n - x_{n-1}) && \text{og} \\ \underline{\text{vektor3}} &= \left(-\frac{x_2 - x_1}{x_1} \cdot 100, \dots, \frac{x_n - x_{n-1}}{x_{n-1}} \cdot 100 \right). \end{aligned}$$

vektor2 og vektor3 har begge (n-1) elementer hvis vektor1 har n elementer.

Hvis innobjektet er matrisel blir de to utobjektene matrise2 og matrise3. De ovenfornevnte beregninger utføres for hver kolonne av matrisel. matrise2 og matrise3 har derfor én rad mindre enn matrisel.

Høyere differanser kan beregnes ved å bruke direktivet flere ganger på rad. Merk at prosentendringene ikke er basert på det første elementet. (Hvis man derimot skal ha det første element som prosentbasis kan man beregne seg fram fra vektor2 med direktivene DELHORISONT, DIVIDER eller INVERS, og MULTIPLISER.)

Direktiv ENDRING er det motsatte av direktiv ADDERKUMULA. Hvis parameter nedenfor inneholder første element i vektor1, kan man regne tilbake fra vektor2 til vektor1 ved hjelp av følgende direktiver:

SETTPÅTOPP parameter vektor2 vektor4.
 ADDERKUMULA vektor4 vektor1.

Direktiv IMINUSA

(Identitetsmatrise MINUS matrisen A)

Form: IMINUSA matrise1 subtraheres fra identitetsmatrise og resultatet settes i matrise2.

Innobjekt: matrise1

Utobjekt: matrise2

Direktivet vil typisk bli brukt som et ledd i beregningen av en Leontief-invers innenfor kryssløpsanalyse. Betegnelsen 'A' på matrise1 er brukt fordi det er vanlig i litteraturen, men direktiv IMINUSA legger ikke noen begrensninger på innholdet av innobjektet, bortsett fra det at det må være kvadratisk.

(Se f.eks. i D. Gale, The Theory of Linear Economic Models, 1960, eller i R. Dorfman, P. Samuelson og R. Solow, Linear Programming and Economic Analysis, 1958.)

Direktiv INVERS

(INVERSE elementer)

Form: INVERS elementene i matrise1/vektor1/parameter1 inverteres hver for seg og resultatet settes i matrise2/vektor2/parameter2.

Innobjekt: matrise1/vektor1/parameter1

Utobjekt: matrise2/vektor2/parameter2

Eksempel på bruk av direktivet:

INVERS A B.

ABS B B.

ELEMULT A B A.

Etter denne sekvens blir A endret slik at positive elementer settes lik +1, negative elementer settes lik -1, mens nullelementer inverteres ikke, og forblir null. (Null er definert som et tall hvis absolutte verdi er mindre enn 1×10^{-20} .)

Direktiv INVERTER

(INVERTERer matrise)

Form: INVERTER matrise1 og sett resultat i matrise2.Innobjekt: matrise1Utobjekt: matrise2

Inverteringen utføres ved en eliminasjonsmetode med kolonnevis flytting i core og pivotering. Rutinen utnytter ikke eventuell spesiell struktur i matrise1, men til gjengjeld konvergerer den alltid, etter et endelig antall skritt. (I tilfelle man har en matrise1 med en spesiell struktur bør man overveie å velge en spesiell metode som utnytter denne strukturen.)

Det sjekkes at matrise1 har like så mange kolonner som rader. Etter sekvensen:

MATRISER A B C.INVERTER A B.MULTIPLISER A B C.

blir C en identitetsmatrise, hvis ikke A er singular.

Direktiv KOLADDER

(KOLonner får ADDERt et element)

Form: KOLADDER adder til hvert element i hver kolonne i matrise1/vektor1 det tilsvarende element i vektor2/matrise2/parameter2 og sett resultatet i matrise3/vektor3.Innobjekter: matrise1/vektor1, vektor2/matrise2/parameter2Utobjekt: matrise3/vektor3

Det blir kontrollert at antall kolonner i matrise1 er lik antall elementer i vektor2. matrise3 får samme dimensjon som matrise1.

1. element i vektor2 adderes til hvert element i 1. kolonne i matrise1, osv.

Direktiv KOLONNE-LAG

(KOLONNER blir LAGget)

Form: KOLONNE-LAG sett parameter1 ganger vektor2 på venstre side av matrise3, fjern et tilsvarende antall kolonner på høyre side, og sett det endelige resultatet i matrise4.Innobjekter: parameter1, vektor2, matrise3Utobjekt: matrise4

Kolonne-lag er et sammensatt direktiv.

Matrise4 vil få samme dimensjoner som matrise3 har. Parameter1 må være en heltallsparameter som går opp i antall kolonner i matrise3 et helt antall ganger. Verdien til parameter1 bør skaffes ved direktiv MATRISEDIM fra et annet objekt, gjerne en liste som styrer dimensjon, i en modell.

Direktivet er beregnet til bruk i økonomiske modeller, der kolonnene representerer år og alternativer med like mange alternativer for hvert år.

Direktiv KOLSKALER

(KOLonner SKALERes)

Form: KOLSKALER multipliser hver kolonne i matrisel/vektor1/parameter1 med det tilsvarende element i vektor2 og sett resultatet i matrise3.

Innobjekter: matrisel/vektor1/parameter1, vektor2

Utobjekt: matrise3

Det blir kontrollert at antall elementer i vektor2 stemmer overens med antall kolonner i matrisel. matrise3 får samme dimensjoner som matrisel.

1. element i vektor2 multipliseres med hvert element i 1. kolonne i matrisel osv.

Direktiv MULTIPLISER

(MULTIPLISERer numeriske objekter)

Form: MULTIPLISER parameter1/vektor1/matrisel med parameter2/vektor2/matrise2 og sett resultatet i parameter3/vektor3/matrise3.

Innobjekter: parameter1/vektor1/matrisel, parameter2/vektor2/matrise2

Utobjekt: parameter3/vektor3/matrise3

Dette er et generelt direktiv som tolker hver teoretisk mulig kombinasjon av innobjektene etter reglene listet nedenfor.

(i) MULTIPLISER matrisel matrise2 matrise3.

Her brukes vanlig matrisemultiplikasjon, om dimensjonene til matrisel og matrise2 stemmer overens, dvs. om kolonnedimensjon av matrisel er lik raddimensjon av matrise2.

(ii) MULTIPLISER vektor1 matrise2 vektor3.

En vektor vil ordinært bli tolket som kolonnevektor. For denne multipliseringen betraktes vektor1 som en radvektor og da er multipliseringen definert og tillatt hvis dimensjonen av vektor1 er lik raddimensjon av matrise2.

(iii) MULTIPLISER matrisel vektor2 vektor3.

Vanlig postmultiplisering av en matrise med en kolonnevektor - tillatt hvis dimensjonene stemmer.

(iv) MULTIPLISER vektor1 vektor2 parameter3.

Her beregner direktivet det indre produktet av vektor1 og vektor2, dvs. direktivet multipliserer hvert element i vektor1 med det tilsvarende elementet i vektor2 og adderer produktene. Antall elementer må stemme overens.

(v) MULTIPLISER parameter1 matrise2 matrise3.

MULTIPLISER matrisel parameter2 matrise3.

MULTIPLISER parameter1 vektor2 vektor3.

MULTIPLISER vektor1 parameter2 vektor3.

MULTIPLISER parameter1 parameter2 parameter3.

Direktivet multipliserer hvert element i objektet med den angitte parameteren.

For å multiplisere en (nx1) vektor/matrise med en (1xm) tenkt linjevektor eller matrise, må det andre objektet være matrise.

Postmultiplisering av en matrise med en (kolonne-)vektor er mange ganger mer effektivt enn pre-multiplisering av den transponerte matrise med den tilsvarende radvektor. Dette er p.g.a. den radvise lagring av matriser.

Direktiv PAKKPAKKMULT

(PAKKet matrise og PAKKet matrise MULTIpliseres)

Form: PAKKPAKKMULT matrisen representert i pakket form av rekordsett1 (med tre elementærfelt) multipliseres med matrisen representert av rekordsett2, og resultatet settes i rekordsett3.

Innobjekter: rekordsett1, rekordsett2

Utobjekt: rekordsett3

Direktivet svarer til direktiv MULTIPLISER, men det godtar bare pakkede matriser som mulige innobjekter. (PAKKPAKKMULT vil seinere bli erstattet av MATMULT, som skal kunne multiplisere alle fire kombinasjoner av pakkede og upakkede matriser, samt de andre operasjonene som nå utføres av MULTIPLISER.) Se også direktivene MULTIPLISER, PAKKINN, PAKKUT, og VANPAKKMULT.

Direktiv RADADDER

(RADer får ADDERt et element)

Form: RADADDER adder til hvert element i radene i matrise1 det tilsvarende element i vektor/
matrise2/parameter og sett resultatet i matrise3.

Innobjekter: matrise1, vektor/matrise2/parameter

Utobjekt: matrise3

Det blir kontrollert at antall rader i matrise1 er lik antall elementer i vektor. matrise3 får samme dimensjoner som matrise1.

1. element i vektor adderes til hvert element i 1. rad i matrise1, osv.

RADADDER kan benyttes for å duplisere (lage mange eksemplarer av) f.eks. en VEKTOR som mangfoldige kolonner i en MATRISE.

F.eks.:

ARKIVTAPE STANDARD.

GENOBJEKT KMATRISE 50 50 0 A.

GENOBJEKT STIGENDE 50 1 1 LITEN-A.

MATRISE A.

VEKTOR LITEN-A.

RADADDER A med LITEN-A blir til A.

Matrisen A vil da inneholde vektoren LITEN-A som hver av sine kolonner. LITEN-A ble generert i eksemplet med stigende tall nedover kolonnen, 1,2,3,4,5, Matrisen A ble opprinnelig generert av GENOBJEKT med bare nuller i.

Direktiv RADSKALER

(RADer SKALERes)

Form: RADSKALER multipliser hver rad i matrisel med det tilsvarende element i vektor og sett resultat i matrise2.

Innobjekter: matrisel, vektor

Utobjekt: matrise2

Det blir kontrollert at antall elementer i vektor stemmer overens med antall rader i matrisel. matrise2 får samme dimensjoner som matrisel.

1. element i vektor multipliseres med alle element i 1. rad i matrisel, osv.

Direktiv SOLVE

Form: SOLVE løs det lineære ligningssystem med koeffisienter matrisel, ukjente vektor2/matrise2 og høyreside vektor3/matrise3.

Innobjekter: matrisel, vektor3/matrise3

Utobjekt: vektor2/matrise2

matrisel må være kvadratisk. vektor2/matrise2 får dimensjoner og klasse kopiert fra vektor3/matrise3.

Metoden er Gauss-Jordan eliminasjon med pivotering. Kolonnene blir redusert i fast rekkefølge.

Hvis alle matrisene får plass i core (dvs. har mindre enn ca. 8 000 elementer totalt) bør direktivene TRIANGULER og SUBTILBAKE heller brukes.

Direktiv SUBTILBAKE

(SUBstituer TILBAKE)

Form: SUBTILBAKE med triangulert matrisel, vektor (pivoteringsrekkefølge) og matrise2 av høyresider som input; løsningene vil overskrive høyresidene som output.

Innobjekter: matrisel, vektor, matrise2

Utobjekt: matrise2

Dette direktivet er beregnet til å brukes etter direktivet TRIANGULER (se dette) for tilsammen å løse lineære ligninger med én eller flere høyresider. Argumentene matrisel og vektor skal normalt være outputargumenter fra TRIANGULER, mens matrise2 inneholder høyresiden(e) i ligningssystemet som kolonne(r). Etter utførelsen av direktivet vil høyresidene være erstattet av de tilsvarende løsninger.

Direktivet holder alle tre argumentene i helhet i core. Metoden er tilbakesubstitusjon i den triangulerte matrisen med høyresiden(e), idet det tas hensyn til pivoteringsrekkefølgen som er gitt i vektor.

Direktiv SUBTRAHER

(SUBTRAHER numeriske objekter)

Form: SUBTRAHER parameter1/vektor1/matrise1 + parameter2/vektor2/matrise2 = parameter3/vektor3/matrise3.

Innobjekter: parameter1/vektor1/matrise1, parameter2/vektor2/matrise2

Utobjekt: parameter3/vektor3/matrise3

Direktivet gjelder for både hel- og flytendetallsobjekter.

Direktiv SUMKOL

(SUMmer KOLonner)

Form: SUMKOL Summerer over kolonner i matrise/vektor1 og sett resultatet i vektor2/parameter.

Innobjekt: matrise/vektor1

Utobjekt: vektor2/parameter

Utobjektet får den klassen det hadde før direktivet, men hvis denne var urimelig vil direktivet stoppe og gi feilmelding.
Se alternativt direktiv GENOBJEKT.

Direktiv SUMRAD

(SUMmerer RADer)

Form: SUMRAD Summerer over rader i matrise/vektor1 og setter resultatet i vektor2/parameter.

Innobjekt: matrise/vektor1

Utobjekt: vektor2/parameter

Det blir summert over radene slik at matrise går over til en vektor2 (kolonnevektor), og vektor1 blir uendret. (Med SUMKOL, derimot, blir en vektor summert til en parameter.)

Hvis matrise bare har en rad, kan resultatet være parameter. Brukeren bestemmer selv klassen på utobjektet ved sin deklarasjon av objektet. Det blir testet at man ikke summerer en matrise med flere rader til en parameter.

Se alternativt direktiv GENOBJEKT.

Direktiv TRANSFORMER

(TRANSFORMERer numerisk objekt)

Form: TRANSFORMER ifølge parameter en matrise1/vektor1/parameter1 sett resultatet i matrise2/vektor2/parameter2.

Innobjekter: parameter, matrise1/vektor1/parameter1

Utobjekt: matrise2/vektor2/parameter2

For parameter benyttes STANDARD-objekter:

parameter = KVADRATISK/LOGARITMISK/LOGARITME10/KVADRATROT/EKSPONENSIELT /ABS/SINUS/COSINUS/NULL-1/-1NULL1
(Disse STANDARD-objektene har verdiene fra 1, inntil 10.) Navnet angir hva slags transformasjon som skal utføres. Transformasjoner utføres element for element i matrise1/vektor1/parameter1.

Eks.:

ARKIVTAPE STANDARD.

TRANSFORMER KVADRATISK MATRISE99 og sett resultatet i NY-MATRISE

TRANSFORMER LOGARITMISK MATRISE99 og sett resultatet i MATRISE99 igjen.

NULL-1 og -1NULL1 utfører henholdsvis:

- i) Avmerking av tall forskjellig fra null-avmerkingstallene er 0 og 1
- ii) Avmerking av positive, negative tall og tall som er null. H.h.v. 1, - 1, 0.

Direktiv TRANSPONER

(TRANSPONER matrise)

Form: TRANSPONER matrisel og sett resultat i matrise2.

Innobjekt: matrisel

Utobjekt: matrise2

Direktivet utfører en ordinær transponering av en vilkårlig matrise.

Direktiv TRIANGULER

(TRIANGULERer en matrise)

Form: TRIANGULER matrisel til triangulær matrise2 og gi pivoteringsrekkefølgen i vektor.

Innobjekt: matrisel

Utobjekter: matrise2, vektor

Direktivet utfører en Gaussisk triangulering av en forutsatt ikke-singulær matrisel. Direktivet benytter radpivotering, og rekkefølgen som radene blir pivotert i blir angitt i vektor. Den triangulerte matrise2 og vektor er beregnet som input til direktiver for ligningsløsning og invertering.

Direktivet har den ulempe at det krever plass i core for input-matrisen som helhet. Hvis matrisen er liten gjør dette ikke noe. Outputmatrisen allokeres derimot blokkvis. Metoden som brukes er Gaussisk eliminering med radskalering og radpivotering. Det øvre triangel av output-matrisen inneholder det Gaussiske triangel, det nedre triangel inneholder multiplikatorene som blir brukt under elimineringen. Da pivoteringen utføres uten fysisk ombytte av radene, vil de to triangler bli ekte triangler først etter en ombytting av radene i pivoteringsrekkefølgen. I vektor inneholder 1. element radnummeret til første rad som ble pivotert, 2. element radnummeret til annen rad som ble pivotert osv.

Direktiv TRITILINVERS(fra TRIangulert matrise
TIL INVERS matrise)Form: TRITILINVERS fra triangulert matrise1 og vektor til invers matrise2.Innobjekter: matrise1, vektorUtobjekt: matrise2

Dette direktivet er beregnet for å brukes etter direktivet TRIANGULER for tilsammen å danne den inverse av inputmatrisen til TRIANGULER. matrise1 og vektor vil normalt være utobjekter fra TRIANGULER.

Direktivet krever alle tre argumentene i helhet i core. Direktivet er et annenordens direktiv som først setter opp en identitetsmatrise av samme orden som originalmatrisen, og dernest kaller inn SUBTILBAKE med identitetsmatrisen som høyresider.

Direktiv VANPAKKMULT(VANlig matrise og PAKKet
matrise MULTIpliseres)

Form: VANPAKKMULT matrise1 (av vanlig type) multipliseres med matrisen representert i pakket form av rekordsett2 (med tre elementerfelt), og resultatet settes i matrise3, som er av vanlig type.

Innobjekter: matrise1, rekordsett2Utobjekt: matrise2

Direktivet svarer til direktiv MULTIPLISER, men det godtar bare pakket matrise som andre objekt. VANPAKKMULT vil seinere bli erstattet av MATMULT, som skal kunne multiplisere alle fire kombinasjoner av pakkede og upakkede matriser, samt de andre operasjonene som nå utføres av MULTIPLISER. Se også under direktivene MULTIPLISER, PAKKINN, PAKKUT, og PAKKPAKKMULT.

Ved transponering av alle matrisene kan dette direktivet anvendes også for å multiplisere en pakket matrise med en vanlig matrise. (Dette er mer effektivt enn et eventuelt PAKKVANMULT direktiv.) Pakkede matriser kan transponeres ved hjelp av REDREKORD og SORTER slik:

ARKIVTAPE STANDARD.

omvendt KOLPAKKNR RADPAKKNR PAKKETVERDI.REDREKORD pakkmat1 omvendt = pakkmat1.SORTER pakkmat1 på omvendt = pakkmat1.

pakkmat1 deskribert ved PAKKMAT.

C-2.5 Liste- og sublistedirektiverDirektivStatus pr. 1/1-74

ADDTILDELKOL	ikke testet
ADDTILDELMAT	virker
ADDTILDELRAD	ikke testet
BYTTKOL	virker
BYTTMAT	virker
BYTTRAD	ikke testet
EKSKOL	virker
EKSMAT	virker
EKSRAD	virker

<u>Direktiv (forts.)</u>	<u>Status pr. 1/1-74 (forts.)</u>
KOMBKOL	virker
KOMBRAD	virker
MULTDELKOL	ikke testet
MULTDELMAT	virker ikke
MULTDELRAD	ikke testet
SETTKOL	virker
SUBEKSMULT	virker
SUBLISTGEN (se C-2.2)	-

Direktiv ADDTILDELKOL(ADDer TIL en DEL av
KOLonnene i matrise)

Form: ADDTILDELKOL til hvert element i kolonnene i matrisel/vektor1 gitt ved subliste adder parameter (flytende tall) og sett resultatet i matrise2/vektor2.

Innobjekter: matrisel/vektor1, subliste, parameter

Utobjekt: matrise2/vektor2

Klassen til utobjektet blir bestemt av brukeren. (Direktivet stopper med feilmelding hvis den er urimelig.)

Med vektor1 som innobjekt blir vektoren tolket som kolonnevektor, og parameteren blir addert til hvert element i vektoren som en følge av dette.

Se også direktivene ADDTILDELMAT og SUBLISTGEN.

Direktiv ADDTILDELMAT

(ADDer TIL en DEL av en MATrise)

Form: ADDTILDELMAT til hvert element i del av matrisel gitt ved sublistel subliste2 adder parameter (flytende tall) og sett resultatet i matrise2.

Innobjekter: matrisel, sublistel, subliste2, parameter

Utobjekt: matrise2

Direktivet er identisk med MULTDELMAT, bortsett fra at det her gjelder addering.

Se også under direktivene EKSMAT, SUBLISTGEN, ADDTILDELRAD og ADDTILDELKOL.

Direktiv ADDTILDELRAD(ADDer TIL en DEL av
RADene i matrise)

Form: ADDTILDELRAD til hvert element i radene i matrisel/vektor1 gitt ved subliste adder parameter (flytendetail) og sett resultatet i matrise2/vektor2.

Innobjekter: matrisel/vektor1, subliste, parameter

Utobjekt: matrise2/vektor2

Når første innobjekt er vektor1 blir dette tolket (som vanlig) som kolonnevektor. Klassen til utobjektet blir bestemt av brukeren. (Direktivet stopper med feilmelding hvis den er urimelig.)

Se under direktivene ADDTILDELMAT, ADDTILDELKOL og SUBLISTGEN.

Direktiv BYTTKOL

(BYTT ut KOLONner i matrise)

Form: BYTTKOL kolonnene i matrise1 gitt ved subliste byttes ut med matrise2/vektor2 og resultatet settes i matrise3.

Innobjekter: matrise1, matrise2/vektor2, subliste

Utobjekt: matrise3

Direktivet er lik direktivet BYTTMAT, med den forskjell at det gjelder utbytting (overskriving) av hele kolonner som er referert ved subliste.

2. objektet må ha samme antall kolonner som subliste angir skal byttes ut.

Direktiv BYTTMAT

(BYTT ut delMATrise)

Form: BYTTMAT del av matrise1 gitt ved sublistel subliste2 byttes ut med matrise2 og resultatet settes i matrise3.

Innobjekter: matrise1/vektor1, matrise2/vektor2, sublistel, subliste2

Utobjekt: matrise3/vektor3

Tilsvarende som for direktiv EKSMAT vil første sublistel referere til rader og subliste2 til kolonner i matrise1. Man kan tenke seg denne delmatrisen først ekstrahert etter samme regler som for EKSMAT matrise2 satt inn i matrise1 i de rader og kolonner det ble ekstrahert fra. Den må passe nøyaktig m.h.t. antall rader og kolonner. Ikke hele en rad eller kolonne blir nødvendigvis berørt, eventuelt bare en del av en rad eller en del av en kolonne.

De ekstraherte (overskrevne) delrader og delkolonner behøver ikke å være gruppert på noen bestemt måte. De nye delradene og delkolonnene blir sortert inn i den opprinnelige matrisen i den rekkefølgen de gamle delradene og delkolonnene ble ekstrahert og kastet. Se under direktiv EKSMAT m.h.t. omstokking. Se også under direktiv SUBLISTGEN.

Bare flytendetails matriser blir tillatt som objekter i BYTTMAT.

Direktiv BYTTRAD

(BYTT ut RADER i en matrise)

Form: BYTTRAD radene i matrise1 gitt ved subliste byttes ut med matrise2 og resultatet settes i matrise3.

Innobjekter: matrise1/vektor1/rekordsett1/liste1, matrise2/vektor2/rekordsett2/liste2, subliste

Utobjekt: matrise3/vektor3/rekordsett2/liste2

Se under direktivene BYTTMAT og BYTTKOL. Ved vektor1 blir radene elementer i vektoren.

Klasse til utobjektet kan fastsettes av brukeren. (Direktivet stopper med feilmelding hvis den er urimelig.)

Vektor kan brukes hvorsomhelst en matrise med én kolonne kunne brukes i direktivet.

matrise2/vektor2/rekordsett2/liste2 må tilsvare i antall rader/elementer/rekorder det delobjekt man vil bytte ut.

Direktiv EKSKOL

(EKStraher KOLonner fra matrise)

Form: EKSKOL ekstraher de kolonnene av matrisel som er angitt ved subliste og sett resultatet i matrise2.

Innobjekter: matrisel, subliste

Utobjekt: matrise2

EKSKOL ligner EKSMAT, med den forskjell at her gjelder det bare hele kolonner som skal ekstraheres, i en rekkefølge spesifisert av subliste.

Se også direktiv SUBLISTGEN.

Direktiv EKSMAT

(EKStraher MATrise fra matrise)

Form: EKSMAT ekstraher del av matrisel gitt ved sublistel subliste2 og sett resultatet i matrise2.

Innobjekter: matrisel, sublistel, subliste2

Utobjekt: matrise2

sublistel skal være en subliste som refererer til rader, mens subliste2 refererer til kolonner. (Se direktiv SUBLISTGEN.)

Eks.:

SUBLISTGEN SUBA består av RADERIA og RADUTVALG.

SUBLISTGEN SUBB består av KOLONNERIA og KOLONNEUTVALG.

EKSMAT ekstraher ut av matrisen A den delmatrisen angitt ved SUBA og SUBB og sett resultatet i B.

LISTE RADERIA KOLONNERIA

RADUTVALG KOLONNEUTVALG.

MATRISE A B.

SUBLISTE SUBA SUBB.

Listene må være entydige (ingen dobbeltforekomster). Se under direktiv ANTDOBLIST i denne sammenhengen. SUBLISTGEN vil gi melding om dobbeltforekomster.

De listeelementene i listene RADUTVALG og KOLONNEUTVALG behøver ikke å forekomme i den samme orden som i RADERIA og KOLONNERIA (som betegner radene og kolonnene i matrise A). Ved å forandre rekkefølge av listeelementene kan brukeren foreta simultan omstokking av rader og kolonner i A, samtidig med ekstrahering.

Den refererte omstokkingen er noe forskjellig fra en fullstendig omstokking av vilkårlige elementer i matrisen.

De radene og kolonnene som brukeren kan ekstrahere ved bruk av dette direktivet behøver heller ikke være noen samlet blokk eller delmatrise i matrisel.

Matriseelementene må være flytendetaill.

Direktiv EKSRAD

(EKStraher RADer)

Form: EKSRAD ekstraher rader av matrisel/vektor1/rekordsett1/listel som er gitt ved subliste og sett resultatet i matrise2/vektor2/rekordsett2/liste2.

Innobjekter: matrisel/vektor1/rekordsett1/listel, subliste

Utobjekt: matrise2/vektor2/rekordsett2/liste2

Se direktivene EKSMAT og EKSKOL. EKSRAD er definert analogt med disse.

vektor1 blir tolket som kolonnevektor. Hvis rekordsett1 eller listel er innobjekt, må utobjektet ha samme klasse. Utobjektet kan være vektor2 bare hvis strukturen tillater og brukeren har deklarerert utobjektet vektor. Ellers blir det automatisk matrise.

Direktiv KOMBKOL

(KOMBinér KOLONner fra to matriser)

Form: KOMBKOL STOKK/STOKKADDER/ADDER kolonner ved at matrisen subliste2 matrise2/vektor2 adderes til matrisen subliste3 matrise3/vektor3 og resultatet settes i matrise4/vektor4 med (kolonne-) liste5.

Innobjekter: STOKK/STOKKADDER/ADDER, subliste2, matrise2/vektor2, subliste3, matrise3/vektor3, liste5

Utobjekt: matrise4/vektor4

STOKK, STOKKADDER, og ADDER er STANDARD-objekter, deklarerert som NAVNORD.

Kolonner fra to (ikke nødvendigvis forskjellige) matriser adderes og/eller stokkes om i ny rekkefølge

(i) STOKK

Hvis man forsøker å plassere flere enn én kolonne i samme kolonneposisjon i resultatmatrisen, eller spesifiserer (implisitt eller utilsiktet) at en kolonne i resultatmatrisen skal stå tom, stoppes direktivet med feilmeldingen: 'STRØKET OFTEDALS TEST'.

(ii) ADDER

Direktivet vil addere kolonner fra matrise2 til kolonner i matrise3. Utobjektet matrise4 får like så mange kolonner som det fins i matrise3, og innobjekt liste5 må stemme med dette i dimensjon. Kolonner i matrise3 er det ikke lov å stokke om, ellers stopper direktivet med feilmelding. Kolonner fra matrise2 kan derimot stokkes om.

ADDER-operasjon styres av en spesiell liste som brukeren lager, som må beskrive alle kolonnene i matrise2 på en bestemt måte. Listen tar som utgangspunkt kolonnenlisten over matrise3, som forutsettes å eksistere fra før av. De kolonnene fra matrise2 som skal adderes til kolonner i matrise3 må få identiske navn med disse. Kolonner i matrise2 som skal ignoreres gis navn som ikke fins blant navnene i listen over kolonnene til matrise3. Den spesielle listen (som vi døper L2) som lages slik brukes da med direktiv SUBLISTGEN for å generere subliste2 for bruk i KOMBOL. I eksemplet er L3 den forutsatt eksisterende listen over kolonnene i matrise3.

SUBLISTGEN subliste2 L3 L2.

Med ADDER blir subliste3 betydningsløs og overflødig, og den skal senere fjernes i en sammensatt versjon av direktivet KOMBKOL. I mellomtiden må brukeren selv generere subliste3 med direktivet SUBLISTGEN, slik:

SUBLISTGEN subliste3 L3 L3

Direktiv SUBLISTGEN vil ikke prosessere lister med dobbeltforekomster, derfor er det dessverre umulig å addere flere kolonner samtidig fra én og samme matrise ved KOMBKOL. Men man kan skrive én og samme matrisenavn som både matrise2 og matrise3. Da får man addert i hvert fall to kolonner fra én og samme matrisen. Direktiv KONVMATREK kan brukes med lister med multiple forekomster for å addere mange kolonner og/eller rader i en og samme matrise, fulgt av KONVREKMAT.

(iii) STOKKADDER

STOKKADDER-operasjonen er en blanding av STOKK og ADDER. Alle mulige omstokkinger og adderinger tillates fra begge innmatriser med unntak av dobbeltforekomster i de kolonnelistene som blir brukt (se ovenfor). I tillegg får man en ny mulighet, det å lage tomme kolonner med nuller. Dette gjør man implisitt ved å nevne i liste5 kolonner som ikke fins i L2 og L3. Som under ADDER vil kolonner som L2 og L3 har felles bli addert til hverandre. Dessuten er det slik her at hverken L2 eller L3 har noen spesiell stilling - begge står like, og subliste2 og subliste3 genereres med direktiv SUBLISTGEN på en analog måte:

SUBLISTGEN subliste2 liste5 L2.

SUBLISTGEN subliste3 liste5 L3.

vektor1 blir tolket som kolonnevektor. Det er naturlig å bruke et NAVNORD som kolonneliste til en vektor.

Klasse til utobjektet kan bestemmes av brukeren. (Hvis den er urimelig stopper direktivet med feilmelding.)

NB. Hvis brukeren vil kombinere elementer fra to vektorer kan han først bruke direktiv TRANSPONER for å snu vektorene slik at de forandres til matriser med én rad. Resultatet transponeres tilbake med samme direktiv.

Direktiv KOMBRAD

(KOMBinér RADER fra to matriser)

Form: KOMBRAD STOKK/STOKKADDER/ADDER rader ved at delmatrisen subliste2 matrise2/vektor2 adderes til delmatrisen subliste3 matrise3/vektor3 og resultatet settes i matrise4/vektor4 med (rad-) liste5.

Innobjekter: subliste2, matrise2/vektor2, subliste3, matrise3/vektor3, liste5.

Utobjekt: matrise4/vektor4

Se under direktiv KOMBKOL. KOMBRAD er helt analog med direktiv KOMBKOL med de tilsvarende operasjoner utført på rader i matriser.

MULTDELKOL/MULTDELMAT/MULTDELRAD

Direktiv MULTDELKOL(MULTipliser en DEL av
KOLonnene i matrise)

Form: MULTDELKOL multipliser hvert element i kolonnene av matrisel/vektor1 gitt ved subliste med flytende tallet parameter og sett resultatet i matrise2/vektor2.

Innobjekter: matrisel/vektor1, subliste, parameter

Utobjekt: matrise2/vektor2

Klassen til utobjektet blir bestemt av brukeren hvis strukturen tillater.

Se også under direktivene MULTDELMAT, MULTDELRAD og SUBLISTGEN.

Direktiv MULTDELMAT(MULTipliser elementene i en
DEL av MATrise)

Form: MULTDELMAT multipliser del av matrisel gitt ved sublistel og subliste2 med parameter og sett resultatet i matrise2.

Innobjekter: matrisel, sublistel, subliste2, parameter

Utobjekt: matrise2

Direktivet skalerer en del av matrisel med parameter.

Delmatrisen gitt ved sublistel og subliste2 behøver ikke å være en sammenhengende blokk i matrisel. De samme regler gjelder som for direktivene EKSMAT, BYTTMAT o.fl. Dette betyr ikke at et hvilket som helst utvalg av matriseelementer kan skaleres på denne måten ved bare én eksekvering av direktivet, men dette kan gjøres ved flere gangers bruk av direktivet.

Se også under direktivene EKSMAT, MULTDELRAD, MULTDELKOL, MULTIPLISER og SUBLISTGEN.

parameter må være flytende tall.

Direktiv MULTDELRAD(MULTipliser en DEL av
RADene i matrise)

Form: MULTDELRAD multipliser hvert element i radene av matrisel/vektor1 gitt ved subliste med parameter (flytende) og sett resultatet i matrise2/vektor2.

Innobjekter: matrisel/vektor1, subliste, parameter

Utobjekt: matrise2/vektor2

Direktivet skalerer kolonner i matrisel/vektor1 med parameter1.

vektor1 blir tolket som kolonnevektor. Klassen til utobjektet blir bestemt av brukeren hvis strukturen tillater.

Se også under direktivene MULTDELMAT og SUBLISTGEN.

Direktiv SETTKOL

(SETT KOLONner i matrise)

Form: SETTKOL fra matrisel med kolonneliste listel trekkes ut kolonner gitt ved liste2 og settes inn i matrise2.

Innobjekter: matrisel, listel, liste2

Utoobjekt: matrise2

Direktivet brukes for å plukke ut kolonner fra matrisel og plassere dem i matrise2.

Matrisene må være flytendetailsmatriser. Listene må ha samme lengdeenhet.

Antall kolonner i matrisel må stemme overens med antall elementer i listel. Antall kolonner i matrise2 vil bli lik antall elementer i liste2. matrise2 vil få samme antall rader som matrisel.

Hvis et navn i liste2 også forekommer i listel blir den tilsvarende kolonne overført til matrise2. Hvis listel har dobbeltforekomster blir bare kolonnen som tilsvarer den første forekomst av navnet, benyttet. Hvis liste2 har dobbeltforekomster blir samme kolonne fra matrisel overført til flere kolonner i matrise2. Hvis et navn i liste2 ikke forekommer i listel blir det generert en nullkolonne i matrise2.

Direktivet kan benyttes både til ekstrahering og utvidelse av matriser. (Da letingen etter kolonner i matrisel foregår fra første kolonne, kan direktivet være noe mindre effektivt for matriser med mange kolonner). Jfr. EKSKOL.

Direktiv SUBEKSMULT

(Sammensatt av SUBlistgen, EKSRad, og MULTIpliser)

Form: SUBEKSMULT matrisel, navnord, liste, matrise2, matrise3.

Innobjekter: matrisel, navnord, liste, matrise2

Utoobjekt: matrise3

(Dette direktivet er et sammensatt direktiv. Direktivene SUBLISTGEN, EKSRAD, og MULTIPLISER benyttes internt.)

Direktivet ekstraherer en bestemt rad fra matrisel og denne raden premultipliseres med matrise2. liste navngir radene i matrisel og navnord som er et element i liste angir hvilken rad i matrisel som skal ekstraheres.

C-2.6 RekordsettdirektiverDirektivStatus pr. 1/1-74

BYTTFELT

virker

BYTTREK

virker

DELHORIZONT (se C-2.7)

-

EKSAGGREDREK

virker

EKSREKORD

virker*

GENMAT (se C-2.3)

-

*) arbeider ineffektivt ved leting etter flere rekordtyper av gangen.

<u>Direktiv (forts.)</u>	<u>Status pr. 1/1-74 (forts.)</u>
GENREK (se C-2.3)	-
KONVMATREK (se C-2.3)	-
KONVREKMAT (se C-2.3)	-
LESBCDTAPE (se. C-2.3)	-
LINK	ikke implementert
PAKKINN (se C-2.2)	-
PAKKUT (se C-2.2)	-
PRINTREKSET	virker
REDREKORD	virker
SETTPÅTOPP (se C-2.7)	-
SETTVEDSIDE (se C-2.7)	-
SKRIVBCDTAPE (se C-2.3)	-
SORTER	virker

Direktiv BYTTFELT (BYTT ut FELTer i rekordsett)

Form: BYTTFELT i rekordsett1 bytt ut alle elementærfelt angitt ved rekordsett2 med de tilsvarende elementærfeltene i rekord3 og sett resultatet i rekordsett4.

Innobjekter: rekordsett1, rekordsett2, rekord3

Utobjekt: rekordsett4

Se under direktivet BYTTREK. Direktiv BYTTFELT er identisk med BYTTREK bortsett fra at det her ikke er nødvendig at en hel rekord fra rekordsett2 stemmer med de relevante elementærfelt i rekordsett1. Det er nok at et eller flere elementærfelt stemmer, for at disse blir byttet ut i de aktuelle stedene i rekordsett1.

Flere enn fem elementærfelt i rekordsett2 og rekord3 er ikke tillatt. Dette innebærer ikke noen réell begrensning, fordi flere elementærfelter kan håndteres ved å benytte direktivet flere ganger.

Direktiv BYTTREK (BYTT ut REKorder i rekordsett)

Form: BYTTREK i rekordsett1 bytt ut alle rekordene angitt ved rekordsett2 med rekord3 og sett resultatet i rekordsett4.

Innobjekter: rekordsett1, rekordsett2, rekord3

Utobjekt: rekordsett4

Hver rekord i rekordsett2 (som er forutsatt å være mye mindre enn rekordsett1) bli sammenlignet med de relevante elementærfeltene i rekordsett1. Når en rekord fra rekordsett2 stemmer overens med en rekord fra rekordsett1 blir de aktuelle elementærfeltene i rekorden i rekordsett1 byttet ut med innholdet av rekord3.

rekordsett2 og rekord3 må av brukeren være identisk beskribert. Det er ikke nødvendig at rekordsett2 blir beskribert med alle de elementærfelt som rekordsett1 har, men alle elementærfelt i rekordsett2 skal også finnes i rekordsett1. D.v.s. at rekordsett2 må inneholde et utvalg av elementærfelt i rekordsett1.

rekordsett4 må bli beskribert med en struktur av elementærfelt som er lik rekordsett1, men de enkelte NAVN og TALL som blir brukt kan være andre.

Se også under direktivet BYTTFELT.

Direktiv EKSAGGREDREK

(EKStraher, AGGreger og
REDiger REKordsett)

Form: EKSAGGREDREK fra rekordsett1 foretar en simultan ekstrahering og aggregering etter spesifikasjoner i rekord1, rediger resultatrekordene etter spesifikasjoner i rekord2 og sett resultatet i rekordsett2.

Innobjekter: rekordsett1, rekord1, rekord2

Utobjekt: rekordsett2

Dette direktivet er svært nyttig og slagkraftig i behandlig av rekordsett. Beskrivelsen av direktivet kan bl.a. av denne grunn virke noe innviklet. Beskrivelsen av direktivets virkemåte er nedenfor angitt med utgangspunkt i innholdet av rekord1, rekord2 og beskrivelsen til rekordsett2.

(i) rekord1

rekord1 kan bare inneholde alfanumeriske elementærfelt (navn). Alle felt i første del av rekord1 må også inngå i rekordsett1, men rekkefølgen er likegyldig. Normalt vil rekord1 bare inneholde noen av de felter som inngår i rekordsett1. Felter som ikke er berørt av de operasjoner som skal utføres bør av effektivitetsgrunner ikke tas med i rekord1. Elementærfeltene i rekord1 må leses inn med samme format som de tilsvarende felter i rekordsett1.

For å gi spesifikasjoner om ekstrahering og aggregering benyttes to spesialtegn, '/' og '⌘'. Alle andre tegn kalles normaltegn. Det kan i spesielle tilfelle være ønskelig å benytte andre spesialtegn og disse kan i tilfelle angis i riktig rekkefølge i de to siste elementærfelt i rekord1 (som altså ikke motsvares av felter i rekordsett1). De to spesialtegn kalles for henholdsvis sammenligningstegn (/) og utfyllingstegn (⌘).

Ekstraheringen spesifiseres ved at det ekstraheres fra rekordsett1 bare rekorder som stemmer overens med rekord1 i alle posisjoner med normaltegn. Alle andre posisjoner i rekord1 må altså være utfylt med spesialtegn.

Aggregering spesifiseres ved å bruke sammenligningstegn i rekord1. En sekvens av rekorder som passerer ekstraheringstesten vil bli aggregert sammen dersom de stemmer overens i alle posisjoner der sammenligningstegnet er brukt. Aggregeringen utføres sekvensielt slik at rekkefølgen av rekordene i rekordsett1 vil ha betydning for resultatet. Aggregeringen innebærer at alle numeriske felt akkumuleres. Det øvrige av den aggregerte rekord overføres fra den første rekord i sekvensen som aggregeres.

Der hvor det i rekord1 ikke er benyttet normaltegn eller sammenligningstegn forutsettes det å være innsatt utfyllingstegn. (Blank oppfattes her som et normaltegn).

For å høyne effektiviteten av ekstrahering/aggregering er det innført tre spesialopsjoner for hyppig forekommende anvendelser. Disse opsjoner påkalles ved å angi 'EKSTRAHER', 'AGGREGER' eller 'TA VEKK' i rekord1, i første felt etter det siste som er felles for rekord1 og rekordsett1, istedenfor et eventuelt sammenligningstegn angitt av brukeren.

EKSTRAHER: Ingen aggregering finner sted, bare ekstrahering.

AGGREGER: Alle rekorder vil bli aggregert til én rekord. Dette gir samme resultat som hvis sammenligningstegnet ikke ble brukt, men er mer effektivt. Samtidig ekstrahering kan utføres som vanlig, altså slik at en ekstrahert del av rekordsett1 aggregeres til én rekord.

TA VEKK: Rekorder blir tatt vekk istedenfor ekstrahert, slik at rekordsett2 blir det som ellers ville vært restmengden. Denne opsjon kan ikke kombineres med aggregering.

Brukeren kan fremdeles angi sitt eget utfyllingstegn i et ekstra, siste felt hvis det er nødvendig.

(ii) rekord2

Første del av rekord2 inneholder felter fra rekordsett1 i vilkårlig rekkefølge. Elementærfeltene i rekord2 må leses med samme format som de tilsvarende felter i rekordsett1. rekord2 benyttes til å endre innholdet av felter. Et tegn forskjellig fra blank i et alfanumerisk felt i rekord2 vil medføre at tegn blir satt inn i denne posisjon i alle rekorder i rekordsett2. Hvis det er behov for det, kan brukeren innføre sitt eget tegn istedenfor blank ved å angi det i et ekstra siste felt i rekord2. Blank (eller brukerens spesialtegn) settes i rekord2 i alle posisjoner som ikke skal redigeres. Numeriske felter kan også endres, men bare på den måten at en verdi angitt i rekord2 vil bli addert til det tilsvarende felt i alle rekorder.

(iii) Rekordsett2

Deskripsjonen til rekordsett2 kan benyttes til å stokke om på feltene og til å fjerne felter fra rekordene i rekordsett1. rekordsett2 kan beskrives av brukeren med et utvalg av (muligens omstokket) felter fra rekordsett1, som bestemmer strukturen i rekordsett2.

Av tekniske grunner må deskripsjonen til rekordsett2 omfatte minst alle feltene i rekord1, med unntak av eventuelle felt i rekord1 som er brukt til å innføre spesialtegn eller gi noen av opsjonene 'EKSTRAHER', 'AGGREGER', o.s.v. Hvis ikke dette er gjort, vil man få feilmeldingen:

'IKKE POSITIV ELM. FELTLENGDE' (fra rutine FEDIST).

Direktiv EKSREKORD

(EKStraher REKORDer)

Form: EKSREKORD ekstraher alle rekorder i rekordsett1/rekord1 som har samme verdi som en rekord i rekordsett2/rekord2/liste felt/navn/tall og sett resultatet i rekordsett3.

Innobjekter: rekordsett1/rekord1, rekordsett2/rekord2/liste, felt/navn/tall

Utobjekt: rekordsett3

Direktivet finner fram til den gruppe elementærfelt som rekordsett1, rekordsett2, og felt har felles, uansett rekkefølge. Det leses framover i rekordsett1, og de rekorder i rekordsett1 som har samme verdier i alle disse elementærfelt som minst én rekord i rekordsett2 blir ekstrahert.

For alternative objekter fungerer direktivet tilsvarende. Dersom liste benyttes istedenfor rekordsett2, refererer felt bare til rekordsett1.

Med eksekverings-opsjon K (se B-6.4) skrives det ut en melding om hvor mange elementærfelt rekordsett1, rekordsett2, og felt har felles.

Direktiv LINK

(LINK elementærfelt)

Form: LINK rekordsett1 felt1 sammenlignes med rekordsett2 felt2 og når innholdet stemmer overens i alle elementærfelt føyes felt3 fra det andre objektet til rekordene i det første objektet og resultatet settes i rekordsett3.

Eksempel:

Formålet er her å få føyd til i rekordsett1 som bl.a. inneholder gatenavn de tilsvarende kart-referanser fra et nøkkelrekordsett, nemlig rekordsett2.

rekordsett1:

<u>PNAVN</u>	<u>GATENAVN</u>
JENSEN	DRONNINGENSGT.
BAKER	STORTINGSGT.
CHARLIE	STORTINGSGT.
ANDERSSON	STORTINGSGT.

rekordsett2:

<u>GATE</u>	<u>NORD</u>	<u>VEST</u>
DRONNINGENSGT.	A	1
STORTINGSGT.	B	2

felt1: GATENAVNfelt2: GATEfelt3: NORD VEST

PNAVN, GATENAVN, GATE, NORD, VEST er DATSY-objekter av klasse NAVN.

Resultatet rekordsett3 blir da slik, hvis det er blitt beskribert med en slik setning i DATSY-programmet rekordsett3 beskribert ved PNAVN, GATENAVN, NORD og VEST.

rekordsett3

<u>PNAVN</u>	<u>GATENAVN</u>	<u>NORD</u>	<u>VEST</u>
JENSEN	DRONNINGENSGT.	A	1
BAKER	STORTINGSGT.	B	2
CHARLIE	STORTINGSGT.	B	2
ANDERSSON	STORTINGSGT.	B	2

Resultatet er et rekordsett som angir kartreferanser til personer som før var bare blitt identifisert ved gatenavn.

Flere opplysninger er nødvendige for å kunne bruke direktivet skikkelig i forskjellige situasjoner:

(i) Det kan være et vilkårlig antall elementærfelt i felt1 og felt2, både av klasse, NAVN og TALL (heltall eller flytende). Men felt1 og felt2 må stemme angående de tilsvarende elementærfelttypene. I eksemplet ovenfor kunne ikke felt2 blitt beskribert som TALL, siden felt1 var beskribert som NAVN.

(ii) rekordsett1 må være sortert på forhånd med hensyn til felt1, og rekordsett2 med felt2 - se under direktiv SORTER. Feilmelding gis og direktivet stopper hvis ikke dette er riktig.

(iii) Direktiv REDREKORD kan brukes etterpå for eventuelt å redusere eller bytte om elementærfelt i rekordsett3, eller for å døpe om elementærfelt.

(iv) rekordsett3 må beskrives med NAVN og TALL i samme rekkefølge som rekordsett1, fulgt av NAVN og TALL i samme rekkefølge som i felt3.

(v) Det er tillatt med elementærfelt i rekordsett2 foruten de som er spesifisert ved felt2 og felt3.

(vi) Direktivet arbeider effektivt.

Direktiv PRINTREKSET

(PRINT REKordSETt)

Form: PRINTREKSET Skriv på linjeskriveren rekordsett/rekord med hodespalte av de elementærfeltnavn som ble brukt i beskripsjonen.

Dette direktivet skriver ut rekordsett eller rekord i standardformat. Hvis hver rekord blir for bred for papiret, blir det varslet om dette. I så fall er det mulig å bruke REDREKORD for å splitte opp rekordsettet for utprinting. De enkelte deler kan da bli skrevet på forskjellige sider og satt sammen med saks og lim senere, hvis det er ønsket.

Direktivet er vesentlig billigere i bruk enn PRINTVERDI for store rekordsett, men krever mer arbeid ved konvertering til annen maskin.

Det å liste ut en stor rekordsett på UNIVAC 1108 ved Fjerndata A.S koster lett 500 - 1000 kr. i 1973 - priser, med PRINTVERDI. Derfor kan det være store gevinster ved å bruke PRINTREKSET.

P.g.a. en avrundings feil i UNIVAC-rutinen EDIT\$ vil enkelte tall bli printet ut litt mindre enn de virkelig er. Dette kan aksepteres i de fleste anvendelser.

Eksempler: 1) 0,003 blir til 2.9999-03 (altså 2.9999×10^{-3} eller 0,0029999)
og 2) 2854 blir til 2853.9

Direktiv REDREKORD

(REDuser REKORDsett)

Form: REDREKORD For alle rekorder i rekordsett1/rekord1 ta med felt/navn/tall og sett de reduserte rekordene i rekordsett2/rekord2.

Innobjekter: rekordsett1/rekord1, felt/navn/tall

Utobjekt: rekordsett2/rekord2

Utobjektet rekordsett2 må være beskribert tidligere slik at beskripsjonen stemmer med resultatet av reduseringen. Det er bare nødvendig at strukturen stemmer, ikke de enkelte navn og tall som er brukt. En kan f.eks. bruke felt til denne beskripsjon slik:

rekordsett2 felt.

REDREKORD rekordsett1 felt rekordsett2.

Dette siste er en grunn til at direktivet angir hvilken del av rekordene som skal beholdes framfor hva som skal tas bort.

Direktiv SORTER

(SORTER objekt)

Form: SORTER rekordsett1/liste1/vektor1 på felt/navn/tall/navnord/SORTLISTE/SORTVEKTOR og sett resultatet i rekordsett2/liste2/vektor2.

Innobjekter: rekordsett1/liste1/vektor1, felt/navn/tall/navnord

Utobjekt: rekordsett2/liste2/vektor2

Dette direktivet kan benyttes til å sortere i stigende orden et rekordsett etter vilkårlig sorteringsnøkkel (satt sammen av elementærfelt i beskrivelse av rekordsettet), en liste eller en vektor.

Ved sortering av rekordsettl angis sorteringsnøkkelen som felt. Ved sortering på et elementærfelt kan også navn eller tall benyttes som sorteringsnøkkel.

Ved sortering av listel eller vektorl oppgis istedenfor sorteringsnøkkel henholdsvis SORTLISTE og SORTVEKTOR som er STANDARD-objekter deklarerert som navnord.

Normalt vil innobjektet som sorteres ha samme klasse som utobjektet, men et rekordsett med bare ett elementærfelt kan sorteres til et utobjekt som er liste eller vektor og omvendt.

C-2.7 Andre manipuleringsdirektiver

<u>Direktiv</u>	<u>Status pr. 1/1-74</u>
DELHORIZONT	virker
DELKOLVIS	virker
DELLISTE	virker
GRUPPELISTE	virker
KOMBDELLISTE	virker
KOPI	virker
SETTPÅTOPP	virker
SETTVEDSIDE	virker
TAVEKKDOBB (se C-2.2)	-

Direktiv DELHORIZONT

(DEL HORIZONTalt)

Form: DELHORIZONT del objekt1 etter linje nummer parameter og sett delene i objekt2 og objekt3.

Innobjekter: objekt1 (MATRISE/VEKTOR/LISTE/REKORDSETT), parameter

Utobjekter: objekt2 (samme), objekt3 (samme).

Direktivet er tenkt brukt sammen med direktiv LET, som kan brukes for å lage en verdi for parameter. Derved mister man ikke den egenskap som man bør ha i økonometriske modeller og lignende systemer, at alle dimensjoner kan lett varieres. Til engangsoperasjoner kan brukeren selv fastsette en verdi for parameter.

Det er ikke noe i veien for å deklare f.eks. VEKTOR for objekt2 og MATRISE for objekt3 bare kombinasjonen er rimelig. I dette eksemplet måtte objekt1 da være enten VEKTOR, eller MATRISE med bare én kolonne.

I tilfelle objekt1 har tilknyttet en radliste er det bedre å utnytte det og bruke direktivet EKSRAD istedenfor DELHORIZONT.

Direktiv DELKOLVIS

(DEL KOLonneVIS)

Form: DELKOLVIS del matrisel etter kolonnennummer parameter og sett delene i matrise2 og matrise3.

Innobjekter: matrisel, parameter

Utobjekter: matrise2, matrise3

Direktivet er programmert primært for bruk i sammensatte direktiver. Det kan også tenkes brukt som et mer generelt datamanipuleringsdirektiv. I tilfelle matrisen har tilknyttet en kolonne-liste, er det bedre å referere til kolonner via denne liste og bruke direktiv EKSKOL (se C-2.5).

Se også direktiv LET (for å lage en verdi til parameter).

Direktiv DELLISTE

(DEL LISTE)

Form: DELLISTE del listel vertikalt etter karakteren gitt ved parameter og sett de delene i liste2 og liste3.

Innobjekter: listel, parameter

Utoobjekter: liste2, liste3

Direktivet godtar istedenfor til liste et rekordsett med bare ett elementærfelt. Det blir testet at dette feltet er alfanumerisk og ikke flytendetail eller heltall. Forskjellige objekter kan ha forskjellige klasser. Klasser til resultatobjektene blir bestemt av brukeren ved deklarasjon. I tilfelle disse objektene skal være rekordsett så må de beskrives med ett alfanumerisk elementærfelt, f.eks.:

PARAMETER 2.

REKORDSET A B C.

DELLISTE A 2 B C.

NAVN VARENR5 VARENR2 VARENR3.

A VARENR5.

B VARENR2.

C VARENR3.

En fordel ved å benytte rekordsett med ett elementærfelt framfor lister er at objektene da vil være direkte tilgjengelige for direktiv REDREKORD.

Direktiv GRUPPELISTE

(få ut GRUPPER fra LISTE)

Form: GRUPPELISTE TELL/TAMED de gruppene i listel spesifiserte ved liste2 og sett resultatet i (heltalls-)vektor1/liste3.

Innobjekter: navnord, listel, liste2

Utoobjekt: vektor1/liste3

Direktivet utfører enten en ekstrahering fra en liste eller en gruppevis opptelling av elementer i en liste.

TELL og TAMED er STANDARD-objekter.

Avhengig av om TELL eller TAMED er første innobjekt, får man vektor1 eller liste3 som utobjekt. (Direktivet fastsetter objektklassen her, uansett brukerens deklarasjoner.)

(i) TELL

Med TELL vil direktivet lage en vektor (heltall), som utobjekt, som angir antallet på elementer i listel som fins i hver gruppe. Antall elementer i vektor1 blir lik antall grupper. Med TELL vil

liste2, som angir gruppeinndelingen, bli spesifisert av brukeren på følgende måte: Hvert element i liste2 skal være det første element til hver gruppe som blir således definert i listel. Hvis første element i listel og liste2 ikke stemmer, kommer det feilmelding.

(ii) TAMED

Med TAMED vil liste2 bli tolket annerledes, og resultatet vil bli liste3 som vil inneholde spesifiserte elementer fra listel. Nå behøver ikke det første elementet i liste2 stemme overens lenger med det første elementet i listel, fordi det vil nå spesifisere begynnelsen av den første gruppe elementer i listel som skal overføres til liste3. Andre elementet i liste2 spesifiserer da første elementet i den neste gruppen. Denne gruppen skal IKKE overføres fra listel til liste3. Så går det videre, med regelen at ujevne (1, 3, 5, ...) nummererte elementer i liste2 viser hvor overføringen skal slåes på, mens elementer i liste2 med jevne nummer (2, 4, 6, ...) viser hvor i listel overføringen skal slåes av.

Eksempel 1:

ARKIVTAPE STANDARD.

GRUPPELISTE TELL elementer i listel i grupper spesifisert ved liste2 og sett resultatet i vektor1.

FORMAT 80A1.

VEKTOR vektor1.

LISTE listel liste2.

PRINTVERDI vektor1.

SLUTTPROD.

*80A1 = (80A1)

* listel, 80A1, R=15

ABCDEFGHIJKLMNO

* liste2, 80A1, R=5

ACFIN

Her vil vektor1 bli printet ut slik:

PRINTVERDI VEKTOR1

2

3

3

5

2

Eksempel 2:

Med samme dataavdeling som før, men med følgende DATSY-tekst:

ARKIVTAPE STANDARD.

GRUPPELISTE TAMED elementer i listel spesifiserte ved liste2 og sett dem i liste3.

FORMAT 80A1.

LISTE listel liste2 liste3.

PRINTVERDI liste3.

SLUTTPROD.

Får man:

PRINTVERDI LISTE3

A

B

F

G

H

N

O

Direktiv KOMBDELLISTE

(KOMBiner DELer av LISTER)

Form: KOMBDELLISTE elementene i liste1 fra og med tegn nummer parameter1 til parameter2 med parameter3 blanke (på høyre side) kombineres med elementene i liste2 fra tegn nummer parameter4 til tegn parameter5 og resultatet settes i liste3.

Innobjekter: liste1, parameter1, parameter2, parameter3, liste2, parameter4, parameter5

Utobjekt: liste3

Istedenfor lister kan rekordsett med ett, alfanumerisk, felt benyttes.

Blanding av lister og rekordsett er ikke tillatt.

Direktiv KOPI

(KOPIer)

Form: KOPI objekt1 kopieres til objekt2.

Innobjekt: objekt1

Utobjekt: objekt2

Direktivet kan bl.a. brukes i sammensatte direktiv hvor f.eks. dimensjonsforandringer er nødvendige, men ikke tillatt p.g.a. andre hensyn. (For øvrig kan det tenkes brukt til omdøping av objekter, men dette kan gjøres enklere ved å beskrivere en arkivtape.)

For rekordsett må beskrripsjonene til objektene stemme overens.

Direktivet kan også brukes for å forandre klasse til et objekt hvor dette er nødvendig. Det er sjeldent.

Direktiv SETTPÅTOPP

(SETT PÅ TOPP)

Form: SETTPÅTOPP objekt1 settes på toppen av objekt2 og resultatet kalles for objekt3.

Alle objekter kan settes på toppen av hverandre i en hvilken som helst fornuftig kombinasjon. Se også direktiv SETTVEDSIDE.

Eksempel:

SETTPÅTOPP listel rekordsett2 liste3.

(I eksemplet må rekordsett2 bestå av ett elementærfelt som er alfanumerisk, og med samme bredde som i listel. Bare grupper med 6 karakterer ad gangen teller ved breddesammenligning, slik at en LISTE som er 6 karakterer bred kan settes på toppen av en som ble innlest med bare 1 karakter i bredden.)

Direktiv SETTVEDSIDE

(SETT VED SIDE)

Form: SETTVEDSIDE objekt1 settes ved siden av objekt2 og resultatet kalles for objekt3.

objekt2 legges til høyre for objekt1. Alle objekter kan settes ved siden av hverandre i en hvilken som helst fornuftig kombinasjon.

Se også direktiv SETTPÅTOPP.

Eksempel:

SETTVEDSIDE vektor1 liste2 rekordsett3.

(I eksemplet må vektor1 og liste2 inneholde samme antall elementer/linjer.)

C-2.8 Direktiver for regresjonsanalyse

<u>Direktiv</u>	<u>Status</u>
DELVEKTOR	virker
NAVNGIKOL	"
NY-SIDE	"
PRINTNRADER (se C-2.1)	-
SUBLISTKON	virker
RAABSKOL	"
RAADDKOL	"
RABININF	"
RABINUTF	"
RADIVKOL	"
REKSPKOL	"
RAELOBSINF	"
RAELOBSUTF	"
RAFORSKYV	"
RAMENTFELT	"
RAMENTKOL	"
RAKBRKOL	"
RAKLARGJØR	"
RAKOPIKOL	"
RAKORRMAT	"
RAKUBKOL	"
RAKVADKOL	"
RAKVRTKOL	"

<u>Direktiv (forts.)</u>	<u>Status (forts.)</u>
RALNKOL	virker
RAMINVERD	"
RAMULTKOL	"
RAPRIKORRMAT	"
RAPRIMSTAV	"
RAREGRESJON	"
RASLUTTDATA	"
RASUBKOL	"

Alle disse direktivene virker, men de er ikke testet særlig mye foreløpig.

Kommentar til regresjonsdirektiver

Regresjonsanalyse kan utføres i DATSY ved hjelp av direktiver som alle har prefikset RA i direktivnavnet, f.eks. RAHENTKOL. Ved å anvende et utvalg av disse direktivene i et direktivprogram sammen med eventuelle andre direktiver, kan man få utført regresjonsanalyse som er godt tilpasset behovet i hver enkelt oppgave, både hva angår redigering av input data og utvelging av de typer resultater man ønsker fra analysen. Nye direktiver for redigering eller beregninger kan utvikles hvis behovet melder seg.

Direktivene for regresjonsanalyse (RA-direktiver) kan deles i to grupper:

- I) Redigering av input data.
- II) Utføring av regresjonsberegninger.

Som startdata for direktivene i gruppe II aksepteres en vanlig flytende matrise der hver kolonne representerer en variabel. Denne matrisen kan være bygget opp ved hjelp av spesialdirektivene i gruppe I, men kan også være skaffet til veie på annen måte. Maksimalt antall variable, uavhengige og avhengige til sammen, som direktivene kan behandle i en analyse, er gitt ved en systemparameter (PARAMETER MAXVAR i PROC-elementet RAMAX), og kan forandres av en DATSY systemprogrammerer. Den er inntil videre satt lik 50.

Direktivene i gruppe I (RA-direktiver for redigering av data) utgjør en "lukket" gruppe direktiver. Det første direktiv som eksekveres fra denne gruppen må alltid være RAKLARGJØR. Dette direktivet vil sette av plass til en særskilt stor matrise til å samle data i og hvor dataene eventuelt kan transformeres og omordnes. Denne matrisen vil samtidig bli gitt spesialstatus, noe som resulterer i at matrisen bare vil være akseptabel som argument for regresjonsanalysedirektivene i gruppe I. Direktivene i gruppe I vil på sin side bare akseptere en matrise av denne spesielle type som hovedargument. Denne spesialstatus er nødvendig, da matrisen foruten data også inneholder annen informasjon som benyttes og oppdateres av direktivene i gruppe I.

Etter at all ønsket bruk er gjort av direktivene i gruppe I, eksekveres direktivet RASLUTTDATA, som vil foreta en sluttredigering av matrisen og omdanne den til en vanlig flytende matrise uten spesialstatus. Etter dette direktivet vil matrisen være akseptabel som argument til ethvert direktiv som kan operere på en vanlig flytende matrise, mens den ikke lenger vil være akseptabel som hovedargument til direktiver i gruppe I. Det er altså nå mulig å operere videre på matrisen med vilkårlige andre direktiver før regresjonsberegningene eller i stedet for regresjonsberegninger.

Som startdata for direktivene i gruppe II for beregning aksepteres en matrise hvor hver kolonne representerer én variabel. Denne matrisen vil ofte være bygget opp ved hjelp av spesialdirektivene i gruppe I for dataredigering, men kan også være skaffet til veie på annen måte.

Gangen i direktivkallene for utførelse av regresjonsanalyse, gitt datamatriksen, vil vanligvis være:

- 1) Direktivet NAVNGIKOL eksekveres. Direktivet har som argument datamatriksen, samt en liste inneholdende navnene man ønsker å referere til variablene med. Direktivet vil knytte listen til matrisen som et attributt.

- 2) Direktivet RAKORRMAT eksekveres. Input argument er datamatriksen, output argument er en matrise som inneholder korrelasjonskoeffisienter, middeltall og standardavvik for variablene. Navnelisten for datamatriksen vil automatisk bli knyttet også til outputmatriksen.
- 3) Om ønskes kan direktivet RAPRIMSTAV eksekveres. Input argumenter er matriksen som ble produsert av RAKORRMAT, samt en liste inneholdende et ønsket utvalg av variabelnavnene. Direktivet vil gi en utskrift av middeltall og standardavvik for de angitte variablene.
- 4) Om ønskes kan direktivet RAPRIKORRMAT eksekveres. Input argumenter er matriksen som ble produsert av RAKORRMAT, samt en liste inneholdende et ønsket utvalg av variabelnavnene. Direktivet vil gi en utskrift av alle korrelasjonskoeffisienter mellom de angitte variablene.
- 5) Direktivet RAREGRESJON eksekveres. Input argumenter er matriksen som ble produsert av RAKORRMAT, samt en liste inneholdende et ønsket utvalg av variabelnavnene. Direktivet vil utføre regresjonsanalyse hvor den første variabelen i listen er avhengig variabel, mens de øvrige angitte er frivariabel (forklaringsvariabel). Direktivet føyer selv til konstantleddet i regresjonsligningen. Direktivet vil gi en utskrift av alle de vanlige resultater og samtidig legge resultatene i en output-matrise for eventuell videre behandling med andre direktiver.
- 6) Om ønskes kan direktivet RAREGRESJON eksekveres flere ganger med forskjellige utvalgslisters.

Direktivet RAREGRESJON vil ikke utføre residualanalyser. Til dette formål kan det implementeres ett eller flere separate direktiver. Input argumenter til disse vil vanligvis være datamatriksen, samt matriksen som ble produsert av RAREGRESJON.

Foruten de nevnte direktiver er det også implementert en del hjelpedirektiver for analysen:

- Direktivet PRINTNRADER, som kan benyttes til å gi utskrift av et oppgitt antall rader av datamatriksen. Det kan benyttes når man ønsker en kontroll av dataene.
- Direktivet NY-SIDE (se også direktiv PRINT), som vil gi sideskift i utskriften. Ved gjentatte eksekveringer av RAREGRESJON kan dette benyttes til å dele utskriftene av analysene opp i grupper.

Hvis man prøver å bruke en matrise som ikke har spesialstatus som hovedargument til et direktiv i gruppe I (unntatt RAKLARGJØR) vil man få eksplisitt feilmelding om dette, fulgt av terminering av jobben.

Hvis man prøver å bruke en matrise som har spesialstatus som argument til et direktiv utenfor gruppe I (eller som argument til direktiv innen gruppe I på en plass hvor vanlig matrise ventes, se enkeltbeskrivelsene) vil man få feilmeldingen:

DALLOK HAR FUNNET EN FEIL.
 UDEFINERT LENHET I OBJEKTTABELL.

Det gjøres oppmersom på at dette er en system-feil-melding som er "lånt" til dette formålet. Denne feilmeldingen har altså to mulige årsaker:

- a) Den som framgår direkte av meldingens tekst, nemlig at et argument til et direktiv ikke har hatt enhetslengden definert når den skulle ha vært det. Dette skyldes vanligvis feil i implementeringen av direktivet.
- b) At en matrise med spesialstatus har vært brukt på feil sted (som beskrevet ovenfor). Dette framgår ikke av meldingens tekst, og denne mulige årsak må huskes av brukerne av regresjonsanalyse-direktivene.

I direktivbeskrivelsene nedenfor vil betegnelsen ramatrise bli benyttet for en matrise med spesialstatus.

Overalt i de følgende beskrivelser hvor alternativene vektor/subliste er gitt menes med vektor en heltall vektor som inneholder de kolonne- (eller variabel-) numrene man ønsker å referere til.

Det er ikke nødvendig at kolonne- (variabel-)numrene forekommer i stigende rekkefølge i vektor, men det anbefales for oversiktens skyld. Elementene i vektor vil bli benyttet i den rekkefølge de står. I vektor er også flere gangers forekomst av samme kolonne- (variabel-) nummer tillatt. I så fall vil samme kolonne (variabel) bli benyttet tilsvarende antall ganger. Dette siste kan ikke oppnås ved bruk av subliste da flere gangers forekomst her ikke tillates av systemet. Se også direktivene DELVEKTOR og SUBLISTKON.

Direktiv DELVEKTOR

(DELVEKTOR som snitt av to vektorer)

Form: DELVEKTOR vektor1 finnes som snittet av vektor2/subliste2 og vektor3/subliste3.

Innobjekter: vektor2/subliste2, vektor3/subliste3

Utobjekt: vektor1

Dette direktivet vil generere en vektor av numeriske pekere som er et utvalg av pekerne i en annen vektor (eller subliste): vektor1 (heltall) vil inneholde de pekere som forekommer både i vektor2/subliste2 og i vektor3/subliste3. Rekkefølgen i vektor1 av de pekere som er felles vil være den samme som i vektor2/subliste2.

Resultatet ved flerdobbelte forekomster av pekere i vektor2 og vektor3 vil være: Hvis en peker forekommer én eller flere ganger i vektor2 og dessuten (én eller flere ganger) i vektor3/subliste3, vil vektor1 inneholde pekeren like mange ganger som l. vektor (på tilsvarende steder i sekvensen).

Eksempel 1:

1. vektor = 3,1,4,3

2. vektor = 1,3

vil gi

vektor1 = 3,1,3

Eksempel 2:

1. vektor = 1,3,4,5

2. vektor = 3,3,5

vil gi

vektor1 = 3,5

Direktiv NAVNGIKOL

(NAVNGI KOLonner)

Form: NAVNGIKOL knytt til matrisel en listel med navn på matrisekolonnene.

Innobjekter: matrisel, listel

Utobjekt: matrisel

Dette direktivet vil knytte listel til matrisel som et attributt.

Dette setter senere direktiver som har matrisel som argument i stand til å finne fram til listel uten at denne er argument. Direktivet NAVNGIKOL kontrollerer at antall navnord i listel er lik antall kolonner i matrisel og gir feilmelding hvis antallene ikke er like. Maksimalt antall navnord tillatt i listel er det samme som maksimalt antall variable RA-direktivene kan behandle. Maksimalt antall karakterer i hvert navnord i listel er av utskriftsmessige grunner satt til 24.

I forbindelse med RA-direktivene representerer kolonne-navnene navnene på variablene i analysen, og direktivet RAKORRMAT (se nedenfor) krever at inputmatrisen tidligere har vært argument til NAVNGIKOL med en tilsvarende navneliste som annet argument.

Direktiv NY-SIDE

(NY SIDE)

Form: NY-SIDE objekt.

Innobjekt: objekt

Utobjekt: -

Dette direktivet vil forårsake at det neste som blir skrevet ut på printer begynner på ny side. Av systemmessige grunner krever direktivet et argument. Dette argumentet kan være et vilkårlig objekt som er innlest eller definert tidligere i programmet.

Direktivet er spesielt beregnet på bruk sammen med RA-direktivene. Disse direktivene gir bare noen blanke linjer før eventuell utskrift, ingen av dem gir sideskift før utskrift. Dette er gjort for å muliggjøre kompakt utskrift av mange analyser. Direktivet NY-SIDE kan så brukes til å dele utskriftene opp i grupper.

Direktiv SUBLISTKON(SUBLISTeKONvertering
fra vektor)Form: SUBLISTKON til subliste fra vektor.Innobjekt: vektorUtobjekt: subliste

Dette direktivet vil generere en subliste som svarer til en vektor (heltall) av numeriske pekere. Det er beregnet til hjelp i de tilfelle hvor man har valgt å benytte en vektor av pekere i noen direktiver som har alternativene vektor/subliste, og så ønsker en tilsvarende subliste for direktiver som bare kan benytte dette.

Den genererte subliste vil ha følgende innhold: Hoveddelen (første halvpart) vil inneholde de samme pekere som vektor, men sortert i stigende rekkefølge. Dessuten vil eventuelle flere gangers forekomster av pekere bli redusert til en gangs forekomster (da "dubletter" ikke er tillatt i en subliste). Annen halvpart av subliste (som sjelden benyttes av noe direktiv) vil inneholde de "naturlige" tall (1, 2, 3) i stigende rekkefølge.

Eksempel:

vektor = 6,4,1,6,2,7

vil gi

subliste =

1	1
2	2
4	3
6	4
7	5

Direktiv RAABSKOL(RA-direktiv for ABSoluttverdi
av KOLonner)

Form: RAABSKOL i ramatrise ta absoluttverdi av kolonnene vektor1/subliste1 og plasser i vektor2/subliste2.

Innobjekter: ramatrise, vektor1/subliste1, vektor2/subliste2

Utobjekt: ramatrise

Dette direktivet vil beregne absoluttverdier av de angitte kolonner og sette de framkomne verdier i nye kolonner på følgende måte: Verdiene i kolonnen som angis av 1. element i vektor1/subliste1 transformeres til absoluttverdi og de framkomne verdier settes inn i kolonnen som angis av 1. element i vektor2/subliste2. Derneft utføres det samme for kolonnene som angis av 2. element osv. vektor1/subliste1 og vektor2/subliste2 må ha samme lengde.

Resultantkolonnene som angis av vektor2/subliste2 kan være kolonner som inntil nå ikke har vært benyttet, eller noen eller alle av resultatkolonnene kan settes inn i kolonner som inneholder data fra før. (Det er også tillatt å sette resultatkolonner inn i operandkolonnene.) I dette tilfelle ville alle tidligere data i disse kolonnene bli fjernet før resultatkolonnene blir satt inn.

Resultantkolonnene vil ha definerte verdier i alle rader hvor operandkolonnen har definerte verdier, mens de vil være merket internt som udefinerte i rader hvor operandkolonnen er udefinert.

Direktiv RAADDKOL(RA-direktiv for ADDering
av KOLonner)

Form RAADDKOL i ramatrise adder kolonnene vektor1/subliste1 og kolonnene vektor2/subliste2 og sett resultatet i kolonnene vektor3/subliste3.

Innobjekter: ramatrise, vektor1/subliste1, vektor2/subliste2, vektor3/subliste3

Utobjekt: ramatrise

Dette direktivet vil addere kolonner parvis og sette resultatene i nye kolonner på følgende måte: Kolonnen som angis av 1. element i vektor1/subliste1 adderes til kolonnen som angis av 1. element i vektor2/subliste2, og det framkomne resultat settes inn i kolonnen som angis av 1. element i vektor3/subliste3. Derneft utføres det samme for kolonnene som angis av 2. element osv. vektor1/subliste1 og vektor2/subliste2 må ha samme lengde.

Resultantkolonnene som angis av vektor3/subliste3, kan være kolonner som inntil nå ikke har vært benyttet, eller noen eller alle av resultatkolonnene kan settes inn i kolonner som inneholder data fra før. (Det er også tillatt å sette resultatkolonner inn i operandkolonnene.) I dette tilfelle vil alle tidligere data i disse kolonnene bli fjernet før resultatkolonnene blir satt inn.

Resultantkolonnene vil ha definerte verdier i alle rader hvor begge operandkolonnene har definerte verdier, mens de vil være merket internt som udefinerte i rader hvor en eller begge operandkolonner er udefinerte.

Direktiv RABININF(RA-direktiv for BINærtransformasjon,
INnenFor intervall)

Form: RABININF i ramatrise transformer kolonnene vektor1/subliste1 til binærvariable med verdi 1 dersom verdien ligger mellom vektor2 og vektor3 og null ellers og sett resultatet i vektor4/subliste4.

Innobjekter: ramatrise, vektor1/sublistel, vektor2, vektor3, vektor4/subliste4
 Utobjekt: ramatrise

Dette direktivet vil foreta en binærtransformasjon av kolonner på følgende måte: Først undersøkes for hver verdi i kolonnen som angis av 1. element i 1. vektor1/sublistel om verdien ligger i intervallet fra (men ikke med) 1. element i vektor2 til og med 1. element i vektor3, hvis det er tilfelle blir resultatet = 1, ellers blir resultatet = 0. Den resulterende verdi settes inn på tilsvarende plass i kolonnen som angis av 1. element i vektor4/subliste4. Derneft utføres det samme for kolonnen som angis av 2. element i vektor1/sublistel osv.

vektor1 og vektor2 må være flytende. Antall kolonner som angis av vektor1/sublistel og vektor4/subliste4 må være det samme og være lik lengdene av vektor2 og vektor3.

Det gjøres oppmerksom på at hvis korresponderende elementer i vektor2 og vektor3 har samme verdi, vil resultatet alltid bli = 0, dvs. intervallet inneholder ingen verdier.

Resultantkolonnene som angis av vektor4/subliste4 kan være kolonner som inntil nå ikke har vært benyttet, eller noen eller alle av resultantkolonnene kan settes inn i kolonner som inneholder data fra før. (Det er også tillatt å sette resultantkolonner inn i operandkolonnene.) I dette tilfelle vil alle tidligere data i disse kolonnene bli fjernet før resultantkolonnene blir satt inn. Resultantkolonnene vil ha definerte verdier i alle rader hvor operandkolonnen har definerte verdier, mens de vil være merket internt som udefinerte i rader hvor operandkolonnen er udefinert.

Direktiv RABINUTF

(RA-direktiv for BINærtransformasjon, UTenFor intervall)

Form: RABINUTF i ramatrise transformer kolonnene vektor1/submatrisel til binærvariable med verdi null dersom verdien ligger mellom vektor2 og vektor3 og 1 ellers og sett resultatet i vektor4/submatrise4.

Innobjekter: ramatrise, vektor1/submatrisel, vektor2, vektor3, vektor4/submatrise4
 Utobjekt: ramatrise

Dette direktivet vil foreta en binærtransformasjon av kolonner på liknende måte som direktivet RABININF (se dette), bare blir her resultatet = 1 hvis den undersøkte verdien ligger utenfor det oppgitte intervall og = 0 hvis den ligger innenfor.

Direktiv RADIVKOL

(RA-direktiv for DIVisjon av KOLonner)

Form: RADIVKOL i ramatrise divider kolonne vektor1/sublistel med kolonnene vektor2/subliste2 og sett resultatet i kolonnene vektor3/subliste3.

Innobjekter: ramatrise, vektor1/sublistel, vektor2/subliste2, vektor3/subliste3
 Utobjekt: ramatrise

Dette direktivet er helt analogt med direktiv RAADDKOL (se dette), bare er regneoperasjonen her divisjon istedenfor addisjon.

Direktiv RAEKSPKOL

(RA-direktiv for EKSPonensiering
av KOLonner)Form: RAEKSPKOL i ramatrise, e^{kolonnene} vektor1/subliste1 = kolonnene vektor2/subliste2.Innobjekter: ramatrise, vektor1/subliste1, vektor2/subliste2Utobjekt: ramatrise

Dette direktivet er helt analogt med RAKVADKOL, bare er regneoperasjonen her eksponential-funksjonen (e^x) i stedet for kvadrering.

RAEKSPKOL/RAELOBSINF/RAELOBSUTF

Direktiv RAELOBSINF(RA-direktiv for ELiminering av
OBServasjoner INNenFor intervall)Form: RAELOBSINF i ramatrise eliminer observasjoner (rader) hvis verdi i kolonnene vektor/subliste ligger mellom vektor1 og vektor2.Innobjekter: ramatrise, vektor/subliste, vektor1, vektor2Utobjekt: ramatrise

Dette direktivet kan eliminere de rader i ramatrise som oppfyller et visst kriterium. For hver rad i ramatrise utfører direktivet følgende: For kolonnen som angis av 1. element i vektor/subliste undersøkes om verdien er større enn (men ikke lik) 1. element i vektor1 og samtidig mindre enn eller lik 1. element i vektor2. Dernest undersøkes for matriseelementet som angis av 2. element i vektor/subliste og om verdien er større enn 2. element i vektor1 og mindre enn eller lik 2. element i vektor2, osv. Hvis ett eller flere av de angitte elementer i en rad ligger innenfor de gitte intervaller (gitt ved vektor1 og vektor2), elimineres hele raden fra matrisen, ellers beholdes den.

vektor1 og vektor2 må være av type flytende. Lengdene av vektor1 og vektor2 må være like og lik antall kolonner som angis av vektor/subliste.

Det gjøres oppmerksom på at hvis korresponderende elementer i vektor1 og vektor2 har samme verdi, vil resultatet for enhver verdi av tilsvarende matriseelement bli at det ligger utenfor intervallet, dvs. intervallet inneholder ingen verdier.

I tilfelle av ufullstendige rader i ramatrise (som kan forekomme øverst og nederst i matrisen) vil fremdeles de elementer som måtte være definert blant dem som angis av vektor/subliste bli underkastet de beskrevne tester.

Direktiv RAELOBSUTF(RA-direktiv for ELiminering av
OBServasjoner UTenFor intervall)Form: RAELOBSUTF i ramatrise eliminer observasjoner (rader) hvis verdi i kolonnene vektor/subliste ligger utenfor intervallene gitt ved vektor1 og vektor2.Innobjekter: ramatrise, vektor/subliste, vektor1, vektor2Utobjekt: ramatrise

Dette direktivet kan eliminere rader i ramatrise som oppfyller et visst kriterium, på lignende måte som direktiv RAELOBSINF (se dette). Men i RAELOBSUTF elimineres i stedet en rad hvis ett eller flere av de angitte matriseelementer i raden ligger utenfor tilsvarende gitte intervall(er), ellers beholdes den.

Direktiv RAFORSKYV(RA-direktiv for FORSKYVning
av kolonner)

Form: RAFORSKYV i ramatrise forskyv hver kolonne så mange posisjoner som angitt i vektor.

Innobjekter: ramatrise, vektor

Utobjekt: ramatrise

RAFORSKYV/RAHENTFELT

Dette direktivet kan forskyve kolonnene i ramatrise nedover eller oppover med forskyvning som angitt for hver enkelt kolonne ved hjelp av vektor.

vektor må være av type heltall, og lengden av vektor må være lik høyeste benyttede kolonnenumer i ramatrise. En positiv verdi av et heltall i vektor angir at tilsvarende kolonne i ramatrise skal forskyves nedover angitt antall posisjoner, en negativ verdi angir at tilsvarende kolonne skal forskyves oppover angitt antall posisjoner, mens en verdi = 0 angir at kolonnen ikke skal forskyves. Det er ingen begrensninger på størrelsene av forskyvningene.

De to viktigste formål ved anvendelsen av dette direktivet vil være:

- a) "Synkronisering" av data hentet inn fra forskjellige kilder.
- b) Introduisering av lead eller lag mellom variable, eventuelt etter at man f.eks. har anvendt direktivet RAKOPIKOL til å lage dubletter av visse kolonner.

Direktivet kan om ønskes utføres flere ganger. Det kan f.eks. være aktuelt først å anvende direktivet for formål a), dernest utføre variabeltransformasjoner og så anvende direktivet for formål b).

Eventuelle ufullstendige matriserader som måtte oppstå øverst og nederst i ramatrise på grunn av forskyvningene, vil senere bli fjernet av direktivet RASLUTTDATA.

Direktiv RAHENTFELT(RA-direktiv for å HENTE
FELT fra rekordsett)

Form: RAHENTFELT hent fra rekordsett elementærfelt som angitt ved navn/tall/felt og føy dem til ramatrise.

Innobjekter: rekordsett, navn/tall/felt, ramatrise

Utobjekt: ramatrise

Dette direktivet vil hente spesifiserte elementærfelter fra rekordsett og føye dem til ramatrise som kolonner. Elementærfeltene blir hentet ut i den rekkefølge de forekommer i rekordsett og satt inn som kolonner suksessivt i ramatrise til høyre for siste foregående innføyde kolonne. De innføyde kolonnene vil alltid bli paassert slik at kolonnenes øverste element kommer i rad nr. 1 av ramatrise. En eventuell forskyvning av hentede kolonner i forhold til hverandre kan gjøres med direktivet RAFORSKYV etter at alle data er hentet inn til ramatrise. Dette direktivet vil man normalt anvende en gang for hvert rekordsett man ønsker å hente data fra.

Elementærfeltene kan være av type flytende, heltall eller dobbel presisjon. Hvis et elementærfelt er av type heltall eller dobbel presisjon, vil det bli konvertert til en flytende kolonne ved innføyelsen i ramatrise.

Rutinene i gruppe I er organisert slik at all henting av data inn til ramatrise må være utført før noen omordninger av dataene eller transformasjoner av dataene i ramatrise gjøres. Dette medfører at så snart noe direktiv for omordning eller transformasjoner er utført på ramatrise, vil flere kall på RAHENTFELT med denne ramatrise som argument ikke bli akseptert av systemet.

Direktiv RAHENTKOL(RA-direktiv for å HENTE
KOLonner fra matrise)Form: RAHENTKOL hent fra matrise kolonnene vektor/subliste og føy dem til ramatrise.Innobjekter: matrise, vektor/subliste, ramatriseUtobjekt: ramatrise

Dette direktivet vil hente spesifiserte kolonner fra matrise og føye dem til ramatrise. Kolonnene blir hentet ut i den rekkefølge de forekommer i vektor/subliste og satt inn suksessivt i ramatrise til høyre for siste foregående innføyde kolonne. De innføyde kolonnene vil alltid bli plassert slik at kolonnenes øverste element kommer i rad nr. 1 av ramatrise. En eventuell forskyvning av hentede kolonner i forhold til hverandre kan gjøres med direktivet RAFORSKYV etter at alle data er hentet inn til ramatrise.

Dette direktivet vil man normalt anvende en gang for hver matrise man ønsker å hente data fra. matrise kan være av type flytende, heltall eller dobbel presisjon. Hvis typen er heltall eller dobbel presisjon, vil de overførte kolonnene bli konvertert til type flytende ved innføyelsen i ramatrise.

Rutinene i gruppe I er organisert slik at all henting av data inn til ramatrise må være utført før noen omordninger av dataene eller transformasjoner av dataene i ramatrise gjøres. Dette medfører at så snart noe direktiv for omordning eller transformasjoner er utført på ramatrise, vil flere kall på RAHENTKOL med denne ramatrise som argument ikke bli akseptert av systemet.

Direktiv RAKBRTKOL(RA-direktiv for KuBikk-
RoT av KOLonner)Form: RAKBRTKOL i ramatrise ta kubikkrot av kolonnene vektor1/subliste1 og sett resultatet i vektor2/subliste2.Innobjekter: ramatrise, vektor1/subliste1, vektor2/subliste2Utobjekt: ramatrise

Dette direktivet er helt analogt med direktiv RAABSKOL (se dette), bare er regneoperasjonen her kubikkrot istedenfor absoluttverdi.

Direktiv RAKLARGJØR(RA-direktiv for
KLARGJØRing)Form: RAKLARGJØR matrise.Utobjekt: (ra-)matrise

Dette direktivet vil sette av plass til en matrise hvor data for regresjonsanalyse kan samles fra andre DATSY matriser og rekordsett, og hvor de innsamlede data kan transformeres og omordnes. Etter dette direktivkallet vil matrisen få spesialstatus og vil derfor videre bli betegnet med ramatrise.

Tilgjengelig plass i den opprettede matrisen bestemmes av to systemparametre (PARAMETER MAXOBS og PARAMETER MAXVAR i PROC-elementet RAMAX). Plassen er inntil videre satt til 10 000 observasjoner av 50 variable.

Kallet på RAKLARGJØR vil ødelegge eventuell informasjon som måtte befinne seg i argumentmatrisen fra før. Den må derfor på dette tidspunkt ikke inneholde data som skal brukes videre.

Direktiv RAKOPIKOL(RA-direktiv for kopiering
av KOLonner)Form: RAKOPIKOL i ramatrise kopier kolonnene vektor1/sublistel over i kolonnene vektor2/subliste2.Innobjekt: ramatrise, vektor1/sublistel, vektor2/subliste2.Utobjekt: ramatrise

Dette direktivet er helt analogt med direktiv RAABSKOL (se dette), bare utføres her ingen regneoperasjon men operandkolonnene overføres uendret til resultatkolonnene.

Direktiv RAKORRMAT(RA-direktiv for KORRelasjons-
MATrise)Form: RAKORRMAT ut fra matrisel av data beregn matrise2 av korrelasjonskoeffisienter.Innobjekt: matriselUtobjekt: matrise2

Dette direktivet har som input matrisel, hvor hver kolonne inneholder observasjoner av en variabel. For denne datamatriksen vil direktivet beregne korrelasjonskoeffisientene for alle mulige par av variable og plassere dem i matrise2. Direktivet beregner også middeltall og standardavvik for variablene, og plasserer også dem i matrise2. Sammensetningen av matrise2 er beskrevet nedenfor. Direktivet forutsetter at en navneliste er blitt knyttet til kolonnene i matrisel før kallet og vil automatisk knytte denne listen også til kolonnene i matrise2.

Direktivet RAKORRMAT gir ingen utskrift av resultatene. For utskrift av middeltall og standardavvik finnes direktivet RAPRIMSTAV, og for utskrift av korrelasjonskoeffisientene RAPRIKORRMAT (se nedenfor).

matrise2, slik som den produseres av dette direktivet, er obligatorisk input argument til direktivet RAREGRESJON (se nedenfor).

Korrelasjonsmatriksen som produseres av RAKORRMAT er av type dobbel presisjon og har M+3 rader og M kolonner, hvor M er antall variable i input-matriksen til RAKORRMAT.

	1	2	-----	M
1	-----			
2				
.				
.				
.				
M	-----			
M+1	-----			
M+2	-----			
M+3	-----			

De M første radene utgjør en kvadratisk og symmetrisk matrise hvor elementet i i'te rad og j'te kolonne inneholder den beregnede korrelasjonskoeffisienten mellom i'te og j'te variabel i input-matrisen.

I M+1'te rad inneholder j'te element middeltallet for j'te variabel i input-matrisen.

I M+2'te rad inneholder j'te element standardavviket for j'te variabel i input-matrisen.

I M+3'te rad inneholder alle elementer antall observasjoner (= antall rader i input-matrisen).

Direktiv RAKUBKOL

(RA-direktiv for KUBering
av KOLonner)

Form: RAKUBKOL i ramatrise kuber kolonnene vektor1/subliste1 og sett resultatet i kolonnene vektor2/subliste2.

Innobjekter: ramatrise, vektor1/subliste1, vektor2/subliste2

Utobjekt: ramatrise

Dette direktivet er helt analogt med direktiv RAABSKOL (se dette), bare er regneoperasjoner her kubering istedenfor absoluttverdi.

Direktiv RAKVADKOL

(RA-direktiv for KVADrering
av KOLonner)

Form: RAKVADKOL i ramatrise kvadrer kolonnene vektor1/subliste1 og sett resultatet i kolonnene vektor2/subliste2.

Innobjekter: ramatrise, vektor1/subliste1, vektor2/subliste2

Utobjekt: ramatrise

Dette direktivet er helt analogt med direktiv RAABSKOL (se dette), bare er regneoperasjonen her kvadrering istedenfor absoluttverdi.

Direktiv RAKVRTKOL

(RA-direktiv for KVadratRoT
av KOLonner)

Form: RAKVRTKOL i ramatrise ta kvadratroten av kolonnene vektor1/subliste1 og sett resultatet i kolonnene vektor2/subliste2.

Innobjekter: ramatrise, vektor1/subliste1, vektor2/subliste2

Utobjekt: ramatrise

Dette direktivet er helt analogt med direktiv RAABSKOL (se dette), bare er regneoperasjonen her kvadratroten istedenfor absoluttverdi.

Direktiv RALNKOL(RA-direktiv for LN
av KOLonner)

Form: RALNKOL i ramatrise ta naturlig logaritme av kolonnene vektor1/subliste1 og sett resultatet i kolonnene vektor2/subliste2.

Innobjekter: ramatrise, vektor1/subliste1, vektor2/subliste2

Utobjekt: ramatrise

Dette direktivet er helt analogt med direktiv RAABSKOL (se dette), bare er regneoperasjonen her naturlig logaritme istedenfor absoluttverdi.

Direktiv RAMINVERD(RA-direktiv for
MINimal VERDi)

Form: RAMINVERD i ramatrise i kolonnene vektor/subliste skift ut elementer som er mindre enn vektor1 med grenseverdien.

Innobjekter: ramatrise, vektor/subliste, vektor1

Utobjekt: ramatrise

Dette direktivet kan modifisere angitte kolonner i ramatrise slik at ingen absoluttverdier i kolonnene er mindre enn gitte grenser. Først undersøkes alle matriseelementer i kolonnen som angis av 1. element i vektor/subliste. De matriseelementer i denne kolonnen som har absoluttverdi mindre enn eller lik verdien av 1. element i vektor1 skiftes ut med denne grenseverdi, mens fortegnet på matriseelementet blir beholdt. Derneft sammenlignes matriseelementene i kolonnen som angis av 2. element i vektor/subliste med grenseverdien gitt ved 2. element i vektor1, og eventuelle utskiftninger foretas, osv.

vektor1 må være av type flytende. Bare absoluttverdiene i vektor1 blir benyttet, eventuelle negative fortegn vil bli ignorert. Lengden av vektor1 må være lik antall kolonner som angis av vektor/subliste.

Direktiv RAMULTKOL(RA-direktiv for MULTiplisering
av KOLonner)

Form: RAMULTKOL i ramatrise multipliser kolonnene vektor1/subliste1 med kolonnene vektor2/subliste2 og sett resultatet i kolonnene vektor3/subliste3.

Innobjekter: ramatrise, vektor1/subliste1, vektor2/subliste2, vektor3/subliste3

Utobjekt: ramatrise

Dette direktivet er helt analogt med direktiv RAADDKOL (se dette), bare er regneoperasjoner her multiplikasjon istedenfor addisjon.

Direktiv RAPRIKORRMAT(RA-direktiv for printing av
KORRelasjonsMATrise)

Form: RAPRIKORRMAT ut fra matrise av korrelasjoner, middeltall og standardavvik korrelasjonene mellom alle de variable angitt i liste.

Innobjekter: matrise, liste

Utobjekt: -

Dette direktivet tar som input matrise som er produsert av direktiv RAKORRMAT (se dette). Fra innholdet av denne vil det skrives ut på printer en tabell over korrelasjonskoeffisientene mellom alle mulige par av de variable hvis navn er angitt i liste. Rekkefølgen av koeffisientene vil være bestemt av rekkefølgen av variablene i liste. Koeffisientene vil bli skrevet ut på form som et nedre triangel av en kvadratisk matrise, delt opp i blokker på en passende måte. Umiddelbart foran denne utskriften vil det bli skrevet ut en liste som gir korrespondansen mellom variabelnavnene og deres rekkefølgenummer i datamatriksen; disse numrene blir så benyttet til identifikasjon av de enkelte korrelasjonskoeffisienter.

Variabelnavnene i liste må være et utvalg av (eller alle) variabelnavnene i listen som ble knyttet til den tilsvarende datamatriksen med NAVNGIKOL, men rekkefølgen behøver ikke være den samme. Det er viktig at navnene på de utvalgte variablene skrives nøyaktig som i listen som var argument til NAVNGIKOL, da de ellers ikke vil bli gjenkjent av systemet.

Direktiv RAPRIMSTAV(RA-direktiv for PRInting av
Middeltall og STandardAVvik)

Form: RAPRIMSTAV ut fra matrise av korrelasjoner, middeltall og standardavvik print resultater for variable angitt i liste.

Innobjekt: matrise, liste

Utobjekt: -

Dette direktivet tar som input matrise som er produsert av direktiv RAKORRMAT (se dette). Fra innholdet av denne vil det skrives ut på printer en tabell over middeltall og standardavvik for de variable hvis navn er angitt i liste. Resultatene vil bli skrevet ut for variablene i den rekkefølgen de er gitt i liste. Variablene vil bli identifisert i utskriften ved sine navn og variabelnummeret i den opprinnelige datamatrikse. Variabelnavnene i liste må være et utvalg av (eller alle) variabelnavnene i listen som ble knyttet til den tilsvarende datamatriksen med NAVNGIKOL, men rekkefølgen behøver ikke være den samme. Det er viktig at navnene på de utvalgte variablene skrives nøyaktig som i listen som var argument til NAVNGIKOL, da de ellers ikke vil bli godkjent av systemet.

Direktiv RAREGRESJON

(RA-direktiv for REGRESJON)

Form: RAREGRESJON ut fra matrisel av korrelasjoner, middeltall og standardavvik og liste av avhengig og uavhengige variable beregn matrise2 av regresjonskoeffisienter og deres kovarianser.

Innobjekter: matrise1, liste

Utobjekt: matrise2

Dette direktivet tar som input matrise1 som er produsert av direktiv RAKORRMAT (se dette), samt en liste som inneholder et utvalg av (eller alle) variabelnavnene i listen som med NAVNGIKOL ble knyttet til datamatriksen som var input-argument til RAKORRMAT. Den første variabelen i liste vil bli brukt som avhengig variabel i regresjonsanalysen, de øvrige i liste vil bli brukt som uavhengige variable (forklaringsvariable). Direktivet føyer selv til konstantleddet i regresjonsligningen. Rekkefølgen av variabelnavn i liste behøver ikke være den samme som i listen som var argument til NAVNGIKOL.

Direktivet vil gi følgende utskrifter fra analysen:

Antall uavhengige variable.

Antall observasjoner.

Rest-standardavvik. Dette er kvadratroten av rest-variansen for regresjonsligningen.

Multipel korrelasjonskoeffisient for regresjonsligningen (estimat).

F.verdi. Denne er F-observatoren for testing av hypotesen at regresjonskoeffisientene til alle forklaringsvariablene er lik 0. Det er også F-observatoren for testing av hypotesen at den multiple korrelasjonen er lik 0.

Frihetsgrader. Disse er frihetsgradene for F-observatoren.

Signifikansnivå. Dette er det nivået F er signifikant på.

F.eks. betyr et signifikansnivå = 0.078 at ovennevnte hypoteser ville bli forkastet ved testing på 10 prosent nivå, men ikke på 5 prosent nivå.

Regresjonskoeffisientene (estimer).

Regresjonskoeffisientenes standardavvik (estimer).

t-verdier. Disse er absoluttverdier av Student's t-observatorer for testing av de enkelte hypoteser at den tilsvarende regresjonskoeffisient er lik 0.

Signifikansnivåer. Disse er signifikanser for t-verdiene ved testing av de nevnte hypoteser. Nivået for den enkelte test er statistisk korrekt bare når man ignorerer de øvrige testene.

Frihetsgrader. Disse er frihetsgrader for t-verdiene.

I tilfelle lineær avhengighet mellom frivariablene innbyrdes (dvs. i de tilfelle hvor normalmatrisen ville bli singulær) vil programmet utelate de overflødige fri-variable og utføre analysen med de gjenværende. Etter analysen vil programmet skrive en melding om hvilke variable som er utelatt, og hvilke de ble funnet å være avhengige av. I angivelsene av hvilke variable de er avhengige av forekommer utskriften "SAMT MULIGVIS KONSTANTLEDDET". Dette betyr at den lineære sammenhengen muligvis kan inneholde et konstantledd (som kan oppfattes som en faktor ganger konstantleddet i regresjonsligningen). Hvis det angis at en utelatt variabel bare er avhengig av konstantleddet, betyr dette altså at den utelatte variabelen er konstant.

Output-argumentet matrise2 inneholder regresjonskoeffisientene, restvariansen og den fullstendige kovariansmatrisen for regresjonskoeffisientene. Dette muliggjør at etterfølgende direktiver kan utføre residualanalyser, beregne konfidensintervaller, o.l. Kovariansmatrisen som produseres av RAREGRESJON er av type dobbel presisjon og har N+3 rader og N kolonner, hvor N er det totale antall variable som er angitt i inputlisten til RAREGRESJON (den avhengige variable medregnet).

1 2 -----	N
1	
2	
.	
.	
.	
N	-----
N+1	-----
N+2	-----
N+3	-----

I det følgende betegner 1. regresjonskoeffisient konstantleddet, 2. regresjonskoeffisient koeffisienten for 1. uavhengige variabel, etc.

De N første radene utgjør en kvadratisk og symmetrisk matrise hvor elementet i i'te rad og j'te kolonne inneholder den beregnede kovarians mellom i'te og j'te regresjonskoeffisient. I N/1'te rad inneholder j'te element den j'te beregnede regresjonskoeffisient.

I N+2'te rad inneholder alle elementene restvariansen for regresjonsligningen.

I N+3'te rad inneholder alle elementene antall frihetsgrader for restvariansen.

Direktiv RASLUTTDATA

(RA-direktiv for SLUTT-
redigering av DATA)

Form: RASLUTTDATA ramatrise.

Innobjekt: ramatrise

Utobjekt: (ra-)matrise

Dette direktivet vil foreta en sluttredigering av dataene i ramatrise. De operasjoner som utføres er:

- a) Fjerning av ufullstendige datarader i topp og bunn av matrisen som måtte være oppstått av ulike kolonnelengder og/eller bruk av direktivet RAFORSKYV.
- b) Fjerning av den spesielle ekstrainformasjon som ble opprettet av RAKLARGJØR og benyttet av de øvrige dataredigeringsdirektivene i RA-gruppen.
- c) Reduksjon av matrisens dimensjoner fra de maksimalt tillatte til de faktisk anvendte.
- d) Forandring av matrisens status fra spesialmatrise til vanlig flytende matrise. Dette medfører at matrisen etter utførelsen betegnes med matrise (og ikke med ramatrise). Det vil altså si at matrisen fra nå av blir uakseptabel som hovedargument for dataredigeringsdirektivene i gruppe I av RA-direktiver, mens den blir akseptabel som argument for alle direktiver som kan operere på en vanlig flytende matrise.

Hvis direktivet RASLUTTDATA finner tomme kolonner mellom de benyttede i ramatrise opp til høyeste benyttede kolonne, vil programmet terminere med en feilmelding. Brukeren bør derfor passe på ved bruk av RA-direktivene for transformasjoner at det etter at alle transformasjoner er utført ikke gjenstår tomme kolonner mellom de benyttede.

Direktiv RASUBKOL(RA-direktiv for SUBtraksjon
av KOLonner)

Form: RASUBKOL i ramatrise subtraher fra kolonnene vektor1/subliste1 kolonnene vektor2/subliste2 og sett resultatet i kolonnene vektor3/subliste3.

Innobjekter: ramatrise, vektor1/subliste1, vektor2/subliste2, vektor3/subliste3

Utobjekt: ramatrise

Dette direktivet er helt analogt med direktiv RAADDKOL (se dette), bare er regneoperasjonen her subtraksjon istedenfor addisjon.

C-3. Spesielle direktiver

Dette kapitlet inneholder beskrivelser av direktiver som er utviklet for helt spesielle anvendelser og som derfor i alminnelighet ikke har allmenn interesse for alle brukere og potensielle brukere av DATSY.

C-3.1 Direktiver for MODIS IV

<u>Direktiv</u>	<u>Status</u>
ADDRADADDER	virker
BEREGN-KUMULASKATT	virker
BEREGN-MARPINNTEKT	virker
BEREGN-PAALOEPEPDE-SKATT	virker
GENGAMMA	virker
GENGAMMADATA	ikke implementert
KOLONNE-LAG (se C-2.4)	-
SKATT-ETTER-INNTEKTSTRINN	virker
STORADSKALER	virker
TRAPESMETODE	virker
XFORDEL	virker

Direktiv ADDRADADDER

Form: ADDRADADDER parameter1, parameter2, matrisel, matrise2, matrise3.

Innobjekter: parameter1, parameter2, matrisel, matrise2

Utobjekt: matrise3

Dette direktivet er et sammensatt direktiv. Direktivene ADDER, DELKOLVIS, GENOBJEKT, KOPI, RADADDER, SETTVEDSIDE, NAVNO og PARA benyttes. (De to sistnevnte direktivene benyttes bare i sammensatte direktiver.)

Parameter1 og parameter2 er heltallsparametre som angir antall kolonner i henholdsvis matrisel og matrise2. Parameterverdiene må i tillegg være slik at parameter1 dividert med parameter2 er lik en heltallsparameter P. Direktivet forutsetter at P er minst lik 2.

ADDRADDER/BEREGN-KOMULASKATT/BEREGN-MARPINNTEKT

Direktivet adderer hver av de P første kolonnene i matrise1 til første kolonne i matrise2. Resultatet danner de P første kolonnene i matrise3. De P + 1 til 2P kolonnene i matrise3 dannes ved først å addere kolonnene 1 + (P + 1), 2 + (P + 2),, P + 2P i matrise1. Hver av kolonnene vi da får, adderes så til andre kolonne i matrise2. Resultatet danner de P + 1 til 2P kolonnene i matrise3. De resterende kolonner i matrise3 dannes ved å fortsette den ovenfor skisserte systematikken. Utobjekt matrise3, vil da ha samme dimensjon som innobjekt matrise1.

I den nåværende utforming av MODIS IV (pr. 1/1-74), benyttes dette direktivet i kapitalslit-modellen. Matrise1 er der kapitalslit for nyinvesteringer i prognoseårene fordelt på investeringsaktiviteter og alternativer, mens matrise2 er kapitalslit for basisårets kapital fordelt på investeringsaktiviteter og prognoseår. Parameteren P står for antall forløp.

Direktiv BEREGN-KUMULASKATT

Form: BEREGN-KUMULASKATT, bruk matrise1 og matrise2 som input og beregn matrise3.

Direktivet er beregnet på å brukes i skatteprogrammet i MODIS.

Objekter som inngår i direktivet:

Input:

Matrise1 som viser marginalsattesatsene i hvert intervall gitt ved matrisen i punkt 4 i direktivet BEREGN-MARPINNTEKT, progresjonsgrensene.

Matrisen har de samme dimensjoner som den sist nevnte matrise. Det er en kolonne for hvert alternativ.

Matrise2 beregnet av direktivet ENDRING. Matrisen har samme antall kolonner som matrise1.

Linjetallet er en mindre enn linjetallet i matrise1. Elementene viser lengden av progresjonsintervallene.

Output:

Matrise3 kumulaskatt, som har samme dimensjoner som matrise2. Den er output fra direktivet ADDERKUMULA når input er produktet av de tilsvarende elementer i matrise2 og matrise1 unntatt siste linjen i denne matrise1.

Direktiv BEREGN-MARPINNTEKT

Form: BEREGN-MARPINNTEKT, bruk matrise1, matrise2, vektor, matrise3, matrise4 og matrise5 som input og beregn matrise6, matrise7, og matrise8.

Direktivet er beregnet på å brukes i skatteprogrammet i MODIS. Det har som input:

Matrise1 med en linje som viser antall skattytere i hvert intervall.

Matrise2 med en linje som viser den samlede inntekt i hvert intervall. Samme antall elementer som matrise1.

Vektor som viser inntektsintervallenes nedre grense. Samme antall elementer som matrise1.

Matrise3 som viser nedre grense for de nye intervallene (progresjonsgrensene) som antall og inntekt skal beregnes for. En kolonne for hvert alternativ. Antall linjer i matrise3 er lik antall intervaller.

Det må være samme antall intervaller i hvert alternativ.

Matrise4 med en linje. Elementene er sysselsettingsindekser som matrisel justeres med. Ett element for hvert alternativ. Samme antall kolonner som matrise3.

Matrise5 med en linje. Elementene er inntektsindekser som intervallgrensene i vektor justeres med. Som for matrise4 er det et element for hvert alternativ. Samme antall kolonner som matrise3.

Den samlede inntekt, matrise2, justeres med produktet av de tilsvarende elementer i matrise4 og matrise5.

Output er:

Matrise6 som viser antall skattytere i hvert av de nye intervallene. (Jfr. matrise3).

Det er en kolonne for hvert alternativ. Elementene i matrise6 svarer til de tilsvarende elementer i matrise3 slik at et bestemt element viser antall skattytere i det intervallet som har det tilsvarende element i matrise3 som nedre grense og elementet i samme kolonne på linjen nedenfor som øvre grense. Nedre grense hører til intervallet, øvre grense til neste intervall. matrise6 har samme dimensjoner som matrise3.

Matrise7 som viser den samlede inntekt i hvert av de nye intervallene. Tilsvarende matrise6. Dimensjonene er de samme.

Matrise8, marpinntekt, tilsvarende matrise6 og matrise7 og har samme dimensjoner som dem. Marpinntekt er et mellomresultat i skatteberegningen og brukes som input til direktivet BEREGN-PAALOEPEDE-SKATT. Elementene er lik elementene i matrise7 minus produktet av de tilsvarende elementer i matrise3 og matrise6.

Direktiv BEREGN-PAALOEPEDE-SKATT

Form: BEREGN-PAALOEPEDE-SKATT, bruk matrisel, matrise2, matrise3 og matrise4 som input og beregn matrise5.

Direktivet er beregnet på å brukes i skatteprogrammet i MODIS.

Følgende objekter inngår i direktivet:

Input:

Matrisel som viser antall skattytere i de inntektsintervallene som beregningen utføres for. matrisel er beregnet av direktivet BEREGN-MARPINNTEKT. Den er objekt nr. 7 i beskrivelsen av dette direktivet. (Se beskrivelse av direktivet BEREGN-MARPINNTEKT.)

Matrise2, kumulaskatt, beregnet av direktivet BEREGN-KUMULASKATT. (Se beskrivelse av dette direktivet.)

Matrise2 har en linje mindre enn matrisel, men samme antall kolonner. Antall kolonner viser antall alternativer beregningen utføres for.

Matrise3 som viser marginalsattesatsene i hvert inntektsintervall som beregningen utføres for. Det er den samme matrisen som objekt nr. 1 til direktivet BEREGN-KUMULASKATT. (Se beskrivelse av dette direktivet.) matrise3 har de samme dimensjonene som matrisel.

Matrise4, marpinntekt, beregnet av direktivet BEREGN-MARPINNTEKT, objekt nr. 9 til dette direktivet. (Se beskrivelse av direktivet.)

Matrise4 har de samme dimensjonene som matrisel.

Output:

Matrise5, påløpende skatt, viser beregnet skatt. Den er et resultat av en addisjon av to matriser. Den første av disse er framkommet ved at matrise2, kumulaskatt, er multiplisert element for element med matrisel unntatt første linje i denne. Den andre er framkommet ved at matrise4, marpinntekt, er multiplisert element for element med matrise3. Første linje i denne matrisen er fjernet før addisjonen.

Dimensjonene i matrise5 blir de samme som i matrise2.

Direktiv GENGAMMA

(GENerér GAMMA)

Form: GENGAMMA spesialrekordsettet GAMMADATA og listene vareliste og sektorliste blir brukt til generering av den transponerte spesialmatrisen GAMMA^T, og heltallsvektoren GAMMAVEKTOR, matrisen SIGMAP^T, og aktivitetsliste.

Forklaring av objektene:

- (1) GAMMADATA må være spesielt laget til generering av GAMMA^T. Den må inneholde fire elementærfelt, som inneholder respektivt:
 - (i) Sektornavn
 - (ii) Varenavn
 - (iii) Flytende tall som settes inn i kolonnene i GAMMA^T
 - (iv) Hele tall som blir rekkefølgenumre som angir hvilke kort angår hvilken aktivitet i hver sektor.
- Sektornavnene er forutsatt sortert på samme måten som i sektorlisten.
- (2) og (3) Vareliste og sektorliste er lister i A5 format som beskriver varene og sektorene som legges til grunn for matrisen GAMMA, og som brukes i GAMMADATA.
- (4) Av tekniske grunner blir GAMMA^T produsert og ikke selve GAMMA. Etter at GAMMA^T er transponert til GAMMA, vil kolonnene i GAMMA vise strukturen til aktivitetene og sektorene i MODIS IV modellen. Hver aktivitet svarer til en kolonne i GAMMA, og sektorene svarer til fortløpende grupperinger av disse kolonnene.
- (5) GAMMAVEKTOR er en heltallsvektor laget av direktiv GENGAMMA som viser hvor mange aktiviteter det finnes i hver sektor i GAMMA. GAMMAVEKTOR vil beskrive også sektorstrukturen til den endelige aktivitetsmatrisen, Λ .
- (6) SIGMAP^T er en aggregeringsmatrise som inneholder bare 1'er og 0'er. Den er laget av direktivet transponert fra dens opprinnelige formulering, fra GAMMAVEKTOR, og har dimensjon (aktivitet, sektor).

- (7) Aktivitetsliste er en liste med format (A10), med genererte sifferkoder som representerer genererte aktiviteter. Den består av de fem første tegn fra de to første elementærfelt i GAMMADATA, satt tett sammen.

Metode.

Antallet av aktiviteter som skal finnes i hver enkelt sektor oppgis i GAMMADATA, som inneholder fire elementærfelt (se(1) ovenfor).

Første feltet inneholder sektornavn. GAMMADATA leses sekvensielt av direktivet, og når det kommer til et nytt sektornavn i det første elementærfeltet begynner direktivet på ny sektor i matrisen GAMMA^T .

Direktivet leser nedover i GAMMADATA slik at det finner fram til antall aktiviteter i den nye sektoren. Det er forutsatt at brukeren har angitt i fjerde feltet til GAMMADATA lokale rekkefølgenumre (1,2,3 ... osv.) som svarer til de aktivitetene han vil sette i sektoren. Hvis sektoren inneholder bare én aktivitet kan rekkefølgenummeret stå som blank. Flere kort i én sektor kan få samme rekkefølgenummer (de må være fortløpende), når de til sammen beskriver én aktivitet.*

Når direktivet genererer en aktivitet i GAMMA vil den opptre forskjellig, avhengig av om det gjelder den første aktivitet i sektoren, eller en senere aktivitet. Med den første aktivitet blir kolonnen først initialisert med tallet 1.0, og så blir senere nuller lagt inn på akkurat de stedene hvor det ikke står null i minst én annen aktivitet i sektoren. Dette gjøres først etter at de andre aktivitetskolonnene i samme sektoren først er blitt nullstilt og senere forsynt med ett eller flere* flytende tall hver fra det 3. elementærfelt i GAMMADATA**, i kolonneposisjoner angitt ved varenavn spesifisert i det 2. elementærfelt i GAMMADATA.

Vareliste blir sammenlignet med varenavn i det 2. elementærfeltet i GAMMADATA for å finne fram til radposisjoner i GAMMA^T .

Notater:

Sektornavnene må forekomme i samme rekkefølgen som tilsvarende sektornavn i det 1. elementærfeltet i GAMMADATA. Sektorliste og vareliste kan bare være A6 format eller mindre. Sektorliste og vareliste kan også være rekordsett med bare ett elementærfelt hver.

Som en logisk følge av dette gjør det ikke noe hvilket flytende tall som står i 3. elementærfelt til en rekord som skal fastsette strukturen til den 1. aktivitet i sektoren. Typisk får en sektor med bare 1 aktivitet 1'er i hele kolonnen.

Eksempel

Ett kort i GAMMADATA hvor det står 12345 i de 5 første posisjoner og ingenting annet, vil normalt danne én ny sektor 12345 med én aktivitetskode, 12345^..... Hvis én sektor 12345 nettopp er blitt behandlet umiddelbart før dette kortet treffes, vil en feilmelding bli skrevet ut, fordi direktivet vil da forsøke å lage en tilsvarende ekstra aktivitet til den forrige sektoren 12345, og vil ikke finne noen rekkefølgenumre i det 4. elementærfelt.

Blank er altså fullt tillatt som et varenavn eller sektornavn, selv om direktivet vil klage når det første sektornavn er blank.

Foreløpig er det en grense på 100 aktiviteter i én sektor. Dette kan utvides ved å forandre i Fortranteksten og direktivattributtfilen (file 3) til direktivet, ved 3 steder totalt.

Når varenavnet 10000 er funnet i 2. elementærfelt i GAMMADATA, blir det produsert en hel kolonne med nuller i GAMMA, til den sektoren som da finnes i 1. elementærfelt i GAMMADATA.

*) Ikke implementert pr. 1/12-73. **) Det blir i dag bare satt inn tallet 1.0 fast, men dette er lett rettet på (linje 106 i subrutinen GAMMAG).

Direktivet kan tenkes skrevet om for å få vekk en del uønskeligetrekk, hvorav det verste er at feil i sektornavnene forårsaker stopp uten at senere feil blir meldt fra. Punchedfeil kan da føre til at direktivet må kjøres om flere ganger, for å rette i GENGAMMA.

Se også direktiv GENGAMMADATA.

Direktiv GENGAMMADATA

(GENERÉR GAMMADATA)

Form: GENGAMMADATA fra spesialmatrisen GAMMA/AKTMAT med spesiell struktur beskrevet av bl.a. en vareliste og en aktivitetsliste generér rekordsett GAMMADATA/AKTMATDATA.

Et direktiv PRINTGAMDATA er planlagt. Direktivene GENGAMMADATA og PRINTGAMDATA er ledd i et system for å skrive ut på papir spesialmatrisene i MODIS IV, på brukerorientert måte. Først bruker man direktiv GENGAMMADATA for å produsere enten rekordsett GAMMADATA eller AKTMATDATA, da bruker man direktiv PRINTGAMDATA for å skrive rekordsettet ut på linjeskriveren på en spesiell måte som kan inkludere tekst på aktivitetene og sektorene (men ikke varene).

Direktivet fungerer på forutsetningen at det gjelder GAMMADATA når et rekordsett med det navnet opptrer som første objekt. Alle andre mulige navn vil forårsake AKTMAT-tolkningen.

Vareliste er en liste som beskriver radene i GAMMA/AKTMAT.

Aktivitetsliste er en liste hvorav hvert element betegner og tilsvarer én kolonne i spesialmatrisen. Den genereres typisk av direktiv GENGAMMA, som produserer matrisen $GAMMA^T$ fra GAMMADATA, en vareliste, og en sektorliste. Den skal være mindre eller lik 12 tegn i bredde og skal normalt ikke inneholde tekst, men bare alfanumeriske sifferkoder.

For en beskrivelse av strukturen til spesialmatrisen GAMMA og rekordsett GAMMADATA, se under direktiv GENGAMMA. AKTMAT er en hvilken som helst matrise med samme rad- og kolonnestruktur som spesialmatrisen GAMMA. AKTMATDATA er en analog representasjon av matrisen AKTMAT, som tilsvarer den representasjon som GAMMADATA gir av matrisen GAMMA. Siden matrisen AKTMATDATA normalt ikke vil bli noe annet enn innobjekt til direktiv PRINTGAMDATA er dens struktur uinteressant.

Grunnen til at utskriftsoperasjonen er delt i to direktiver (GENGAMMADATA og PRINTGAMDATA) er at i tilfelle GAMMA kan GENGAMMADATA brukes til noe annet. Dette blir nå beskrevet.

Modellen MODIS IV blir typisk strukturert slik på produksjonssiden: Først lages GAMMADATA manuelt for å fastsette inndelingen av sektorene i aktiviteter, og så genereres spesialmatrisen GAMMA (ved hjelp av direktiv GENGAMMA), som senere multipliseres med andre data for å få en aktivitetsmatrise, AKTMAT. Det kan skje forandringer i GAMMA og AKTMAT ved å forandre i GAMMADATA og repetere prosessen. Det kan også forandres i spesialmatrisen GAMMA og/eller AKTMAT.

Ved direkte forandringer i GAMMA eller AKTMAT (og hvis nødvendig også i aktivitetsliste, vareliste, og sektorliste) vinner man mye tid. Men da er GAMMADATA foreldet, og forsøk på å oppdatere den vil nesten sikkert resultere i inkonsekvens og feil. Direktiv GENGAMMADATA kan derimot brukes på en naturlig måte (som et ledd i utskrift av den nye spesialmatrisen GAMMA) for å gjennomføre helt konsekvente forandringer i GAMMADATA. Senere forandringer i modellen kan så skje enten via GAMMADATA eller direkte i GAMMA og/eller AKTMAT. Dette blir da valgfritt.

Direktiv SKATT-ETTER-INNTEKTSTRINN

Form: SKATT-ETTER-INNTEKTSTRINN, bruk matrise1, matrise2, vektor1, vektor2, parameter1, parameter2, parameter3, subliste, matrise5 og matrise6 som input og beregn matrise7, matrise8 og matrise9.

Direktivet er laget til bruk ved skatteberegningen i MODIS. Det tar bare ett alternativ om gangen og de forskjellige skattegrupper kan bare beregnes hver for seg.

Direktivet har som input:

1. Matrise1 er en matrise med en linje. Den viser antall skattytere i hvert inntektsintervall.
2. Matrise2 er også en matrise med en linje. Den viser den samlede inntekt i hvert intervall. Den har samme antall elementer som matrise1.
3. Vektor1, en vektor som viser inntektsintervallenes nedre grense. Den har samme antall elementer som de to matrisene som er nevnt ovenfor. Vektor er i denne sammenheng en matrise med 1 kolonne.
4. Vektor2, en vektor som viser nedre grense for de nye intervallene (progresjonsgrensene) som antall og inntekt skal beregnes for.
5. Parameter1, en parameter som i denne sammenheng er en matrise med ett element. Den er en sysselsettingsindeks som matrise1 justeres med.
6. Parameter2, en parameter som viser inntektsindeksen som intervallgrensene i vektor1 justeres med.
7. Parameter3 er en parameter som brukes til å justere påløpende skatt med f.eks. fra kr. til 100 000 kr.
8. Subliste lages av direktivet SUBLISTGEN og brukes i dette direktivet til å ekstrahere de elementene fra den beregnede antallsmatrisen som det skal beregnes påløpende skatt for. Det er de elementer som tilsvarer elementene i marginalsatsmatrisen. Antall linjer må være lik antall linjer i marginalsatsmatrisen.
9. Matrise5, en matrise som viser kumulaskatten beregnet ved direktivet ADDERKUMULA. Dimensjonene er en linje mindre enn i vektor1, en kolonne.
10. Matrise6, en matrise med samme dimensjoner som matrise5. Den viser marginalsattesatsene i de forskjellige intervallene som beregningen foretas for.

Output er:

11. Matrise7 er en matrise som viser antall skattytere i hvert av de nye intervallene. (Jfr. punkt 4 ovenfor.) Dimensjon: 1 linje, like mange elementer som vektor2.
12. Matrise8, en matrise som viser den samlede inntekt i hvert av de nye intervallene. Dimensjonene er de samme som for matrise7.
13. Matrise9, påløpende skatt, viser beregnet skatt. Dimensjonene for matrisen er: Antall linjer, en mindre enn antall elementer i matrise7 og matrise8, 1 kolonne. Se for øvrig output fra direktivet BEREGN-PAALOEPEDE-SKATT.

Direktiv STORADSKALER

Form: STORADSKALER matrisel deles i enkelte kolonner som brukes til å radskalere matrise2 delt etter vektor3 (heltall) og resultatet settes i matrise4.

Innobjekter: matrisel, matrise2, vektor3

Utobjekt: matrise4

Direktivet utfører radskaleringsoperasjon (se direktiv RADSKALER) på deler av matrise2 delt etter kolonner.

vektor3 er en heltallsvektor. Elementene i vektor3 angir antall kolonner i de respektive deler av matrise2 som skal radskaleres av den tilsvarende kolonne i matrisel. Summen av elementene i vektor3 må derfor være lik antall kolonner i matrise2.

I praksis hittil har vektor3 alltid vært GAMMAVEKTOR (se direktiv GENGAMMA), men direktivet har muligens et større anvendelsesområde.

Direktiv TRAPESMETODE

Form: TRAPESMETODE bruk matrisel, matrise2, vektor3/matrise3, vektor4/matrise4, vektor5/matrise5/parameter5, vektor6/matrise6/parameter6 som input og beregn matrise7 og matrise8.

Direktivet er laget for å brukes til skatteberegningen i MODIS. Det beregner antall skattytere og den samlede inntekt i intervallene mellom nye oppgitte progresjonsgrenser når antall skattytere og samlet inntekt i hvert intervall innenfor en annen kjent progresjonsskala er gitt.

Input er:

Matrisel, antall skattytere i hvert intervall innenfor en kjent progresjonsskala.

Matrise2, samlet inntekt innenfor hvert av intervallene. Matrisen har samme dimensjon som matrisel.

Vektor3/matrise3 som må ha dimensjonen 1 den ene veien. Den andre veien må dimensjonen være lik antall kolonner i matrisel. Vektor3/matrise3 viser nedre grense for hvert intervall i den opprinnelige progresjonsskala.

Vektor4/matrise4 som viser nedre grense for intervallene i den progresjonsskala som beregningene skal utføres for. Dimensjonen den ene veien må være 1.

Vektor5/matrise5/parameter5. Objektet har like mange elementer som det er antall linjer i matrisel. Det inneholder sysselsettingsindekser som elementene i matrisel justeres med (multipliseres).

Vektor6/matrise6/parameter6. Objektet har samme dimensjoner som vektor5/matrise5/parameter5 og inneholder prisindekser som de opprinnelige intervallgrensene, vektor3/matrise3, justeres med. Den samlede inntekt, matrise2, justeres med produktet av sysselsettingsindeksen og prisindeksen.

Output er:

Matrise7, antall skattytere i hvert intervall i den nye progresjonsskala, vektor4/matrise4. Den har samme antall kolonner som vektor4/matrise4 og samme antall linjer som matrisel.

Matrise8, samlet inntekt i hvert intervall i den nye progresjonsskala. Matrisen har samme dimensjoner som matrise7.

Direktiv XFORDEL

Form: XFORDEL eksogene anslag gitt i rekordsett1 (DXMAT) med styreskjema rekordsett2 (DSMAT) etter aggregeringsliste rekordsett3 (AGGMAT) og fordelingstall vektor1/INDEKS og forløpsliste liste1 (FLISTE) og årsliste liste2 (ALISTE) og nivåtall vektor2/INDEKS med liste3 og sett resultatet i rekordsett4 (MXMAT).

Dette direktivet er utviklet for å prosessere eksogen input til MODIS IV. Beskrivelsen nedenfor er derfor knyttet til denne anvendelsen.

DXMAT, DSMAT, AGGMAT og MXMAT er rekordsett med følgende feltinndelinger:

DXMAT matrisenavn, variant, variabell, variabel2, år, alternativ, dimensjon, verdi.

DSMAT matrisenavn, forløp, år, alternativ.

AGGMAT variant-0, variant-1, variant-2, ..., variant-n.

MXMAT variabell, variabel2, år, forløp, verdi.

Brukerens eksogene forutsetninger er punchet på kort gruppevis etter en inndeling av de variable. Hver gruppe kalles en DX-matrise og utgjør et rekordsett. Direktivet behandler bare én DX-matrise av gangen. I rekordsettet som utgjør DX-matrisen, er det punchet bare én eksogen verdi i hvert kort. I beskrivelsen av DXMAT er matrisenavn navnet på DX-matrisen, f.eks. 'DX14'. Skjemaet for DX-matrisen har variabelnavn i forspalten og det fylles ut flere kolonner for ulike år eller alternativer. For de forskjellige kolonner innen en og samme matrise kan det benyttes ulike spesifikasjoner av forspalten. Disse kalles for varianter. Det vil alltid finnes en basisvariant, nullvarianten, som de øvrige er aggregeringer av. Dersom feltet variant i DXMAT er deklarerert som TALL angir feltverdien variantnr., dersom feltet er deklarerert som NAVN angir feltverdien det navn som er brukt i beskrivelsen av AGGMAT.

I hver rekord i DX-matrisen er det altså angitt navnet på matrisen og i hvilken variant av skjemaet den variable er spesifisert. Variabelkoden er gitt i to deler, variabell og variabel2. verdi vil oftest være oppgitt som endring fra et år til det neste. Sistnevnte år vil i så fall være oppgitt i rekorden. For dette året kan det være spesifisert flere alternativer av DX-matrisen. alternativ er derfor også oppgitt. Verdier som refererer seg til forskjellige forløp kan ha samme alternativ. Dimensjon angir om verdi er oppgitt som (i) nivåtall (N), (ii) absolutt endring fra foregående år (A) eller (iii) prosentvis endring fra foregående år (P).

DSMAT er styreskjemaet for DX-matrisen punchet som et rekordsett. Hver rekord inneholder navn på DX-matrisen. DSMAT skal inneholde én og bare en rekord for hver krysskombinasjon av forløp og år og for denne kombinasjon er angitt hvilket alternativ som kommer til anvendelse som endringstall eller nivåtall.

AGGMAT er en aggregeringsliste, dvs. et rekordsett som angir sammenhengen mellom ulike varianter av en variabelinndeling. Første felt vil alltid inneholde nullvarianten. De øvrige felter inneholder de øvrige varianter slik at en rekord i AGGMAT angir for en gitt variabel i nullvarianten i hvilken variabel i hver av de øvrige varianter den er inkludert.

FLISTE er en liste som inneholder de forløp som inngår i modellkjøringen. ALISTE er en liste som inneholder de beregningsår som inngår (i ordnet rekkefølge). vektor2 er en vektor med nivåtall for basisåret for de variable i DX-matrisen etter nullvarianten. LISTE er en liste med variabelnavn etter nullvarianten, altså til feltet variant-0 i AGGMAT. vektor1 er en vektor med tall som skal brukes som fordelingsnøkler ved bruk av andre varianter enn nullvarianten. I alminnelighet vil vektor1 være lik vektor2. INDEKS er et STANDARD-objekt som i dette direktivet angir at de variable i DX-matrisen er indekser som er lik én i basisåret. Behovet for fordelingstall og nivåtall for basisåret er da ikke til stede.

Direktivet skal bygge opp MXMAT fra de tall som er gitt i DXMAT. MXMAT må bygges opp etter stigende år fordi tallene i DXMAT oftest vil være gitt som endringstall fra år til år. I de tilfelle tallene i DXMAT er gitt etter en annen variant enn nullvarianten må tallene fordeles etter nullvarianten ved hjelp av AGGMAT dersom det dreier seg om relative anslag og også ved hjelp av vektor1 dersom det gjelder absolutte anslag. DXMAT kan inneholde år og matrisealternativer som ikke er med i DSMAT, og disse skal i så fall ignoreres.

Utkommet i serien Statistisk Sentralbyrås Håndbøker (SSH)

- Nr. 1 Regler for publikasjonenes utstyr m.v. i serien Statistisk Sentralbyrås Håndbøker
- " 2 Veiledning for nye assistenter
- " 3 Regler for maskinskrivning i Statistisk Sentralbyrå
- " 4 Innføring i maskinregning. Hefte 1. Addisjonsmaskiner
- " 5 Innføring i maskinregning. Hefte 2. Kalkulasjonsmaskiner
- " 6 Regler for utstyr m.v. for publikasjoner i seriene Norges offisielle statistikk (NOS) og Samfunnsøkonomiske studier (SØS) og publikasjonen Statistiske meldinger
- " 7 Prinsipper for konstruksjon av statistiske blanketter
- " 8 Framlegg til nordisk statistisk terminologi
- " 9 Standard for næringsgruppering
- " 10 Hjemmel for innkreving av oppgaver
- " 11 Kurs i hullkortmaskiner
- " 12 Adresseliste for folkeregistrene
- " 13 Standard for handelsområder
- " 14 Innføring i DEUCE
- " 15 Programmering for DEUCE. Første hefte
- " 16 Alfasetemet. Et lettkodingssystem for DEUCE
- " 17 Håndbok for DEUCE-operatører
- " 18 Programmering for DEUCE. Annet hefte
- " 19 Varenomenklatur for industristatistikken
- " 20 Regler for publiseringsarbeidet m.v. i Statistisk Sentralbyrå
- " 21 Håndbok for 1401-programmerere og -operatører
- " 22 Statistisk testing av hypoteser ved regresjonsberegninger
- " 23 Utsnitt om prinsipper og definisjoner i offisiell statistikk
- " 24 Standard for gruppering av sykdommer - skader - dødsårsaker i offentlig norsk statistikk
- " 25 Veiledning for brukere av den økonomiske modell MODIS III
- " 26 Statistisk varefortegnelse for utenrikshandelen
- " 27 Utsnitt om prinsipper og definisjoner i offisiell statistikk. Fra Forbruksundersøkelsen 1958
- " 28 Standard for utdanningsgruppering i offentlig norsk statistikk
- " 29 Norsk-Engelsk ordliste
- " 30 Lov, forskrifter og overenskomst om folkeregistrering
- " 31 Håndbok for bruk av NATBLES
- " 32 Konsumprisindeksens representantvarer. Gruppering etter leveringssektor og art
- " 33 Håndbok for bruk av DATSY

Pris kr. 9,00

ISBN 82-537-0329-5