

Title:

D U E T

 **REGNECENTRALEN**

RC SYSTEM LIBRARY: FALKONERALLE 1 DK-2000 COPENHAGEN F

RCSL No: 21-V020

Edition: September 1977

Author: Edith Rosenberg

Keywords: RC4000, RC8000, SYSTEM80, database, SODA, DBMS, PRIMULA, SYSDOK, programming language, interpreter, structured programming.

Abstract: This manual describes the DUET language. DUET is a programming language primarily intended for administration applications especially transaction processing. It covers DBMS operations as well as the reading of input and the printing of output.

Danish edition: 202 sider

Indholdsfortegnelse

1. Indledning
2. Systemoversigt
3. Duetsproget
 - 3.1 Duetprogramstruktur
 - 3.1.1 Duethoved
 - 3.1.2 Duetblokke
 - 3.1.3 Duetinstruktioner
 - 3.1.4 Duetoperander
 - 3.2 Duetoperationer
 - 3.2.1 Execute
 - 3.2.2 Modify
 - 3.2.3 Compute
 - 3.2.4 Assign - of
 - 3.2.5 If - then
 - 3.2.6 Case - of
 - 3.2.7 Action - of
 - 3.2.8 For - do
 - 3.2.9 While - do
 - 3.2.10 Getline
 - 3.2.11 Read
 - 3.2.12 Print
 - 3.2.13 DB-operationer
 - 3.2.14 Select
 - 3.2.15 Exit
 - 3.2.16 Algol

- 4. Duetoversætteren
 - 4.1 Programtekst og listning
 - 4.2 Aktivering af oversætteren
 - 4.3 Ressourcekrav
 - 4.4 Logudskrift
 - 4.5 Fejludskrifter

- 5. Duetsystemet i et styreprogram
 - 5.1 Duettekster og algol blokstruktur
 - 5.2 Initialisering og afslutning
 - 5.2.1 Init_duet1
 - 5.2.2 Init_duet2
 - 5.2.3 Init_duetmaskine
 - 5.2.4 Close_duet
 - 5.3 Fejlprocedurer
 - 5.3.1 Duetdatafejl
 - 5.3.2 Duetprogramfejl
 - 5.3.3 Duetsystemfejl
 - 5.4 Algolspecialaktioner
 - 5.5 Reserverede algolnavne
 - 5.5.1 Brugerkendte algolnavne
 - 5.5.2 Frie arbejdsvariable
 - 5.5.3 Utilgængelige algolnavne

Appendix A. Syntaksbeskrivelse af duetsproget

Index.



DUET
MANUAL

Emne : Indhold

Afsnit:

Side: 3

Dato: 1.9.1977

Referencer

1. RCSL 21-V002: Database80
2. RCSL 21-V019: SODA
3. RCSL 21-V018: DES80-SODA-LD
4. RCSL 21-D005: Connected Files System
5. RCSL 21-V024: DES80-Konsulentvejledning
6. RCSL 21-T004: Teledata
7. RCSL 28-D017: Informal

1. Indledning

duetsystem

Duetsystemet er et generelt datamatisk værktøj for implementering af programmer på RC4000/8000, f.eks. til SYSTEM80.

Duetsystemet består af

duetabler

- en oversætter DUETABLER, der oversætter programmer skrevet i sproget DUET, og

duetmaskine

- en fortolker DUETMASKINE, der - indkopieret i brugerens styreprogram - udfører duetprogrammet.

duetsproget:
indlæsning

Duetsproget indeholder specielle faciliteter til indlæsning og kontrol af data, hvilket gør det betydelig lettere at skrive indlæseprogrammer i duet end i algol.

udskrivning

Også udskrivning af resultater implementeres særlig nemt i duetsproget, hvor trykkesystemet er baseret på primulaprocedurerne (som også anvendes i GENIUS).

soda-dbms

Acces til databaseindivider sker med brug af soda-dbms'en (ref.2), og duetsproget er derfor nært knyttet til soda systemets ld-beskrivelse, sodald.

sodald

Duetsproget er beskrevet i detaljer i afsnit 3 og oversætteren i afsnit 4.

duetmaskine
styreprogram

Duetmaskinen er den fortolker, der udfører et duetprogram. Den kan ikke arbejde alene, men må indkopieres i brugerens styreprogram, som skal tage sig af opstart og afslutning af kørslen.

SYSTEM80

I SYSTEM80 findes i øjeblikket to sådanne færdige styreprogrammer:

- dataentrysystemets (DES80) 'telescop' og
- telefaktsystemets 'teleop'.

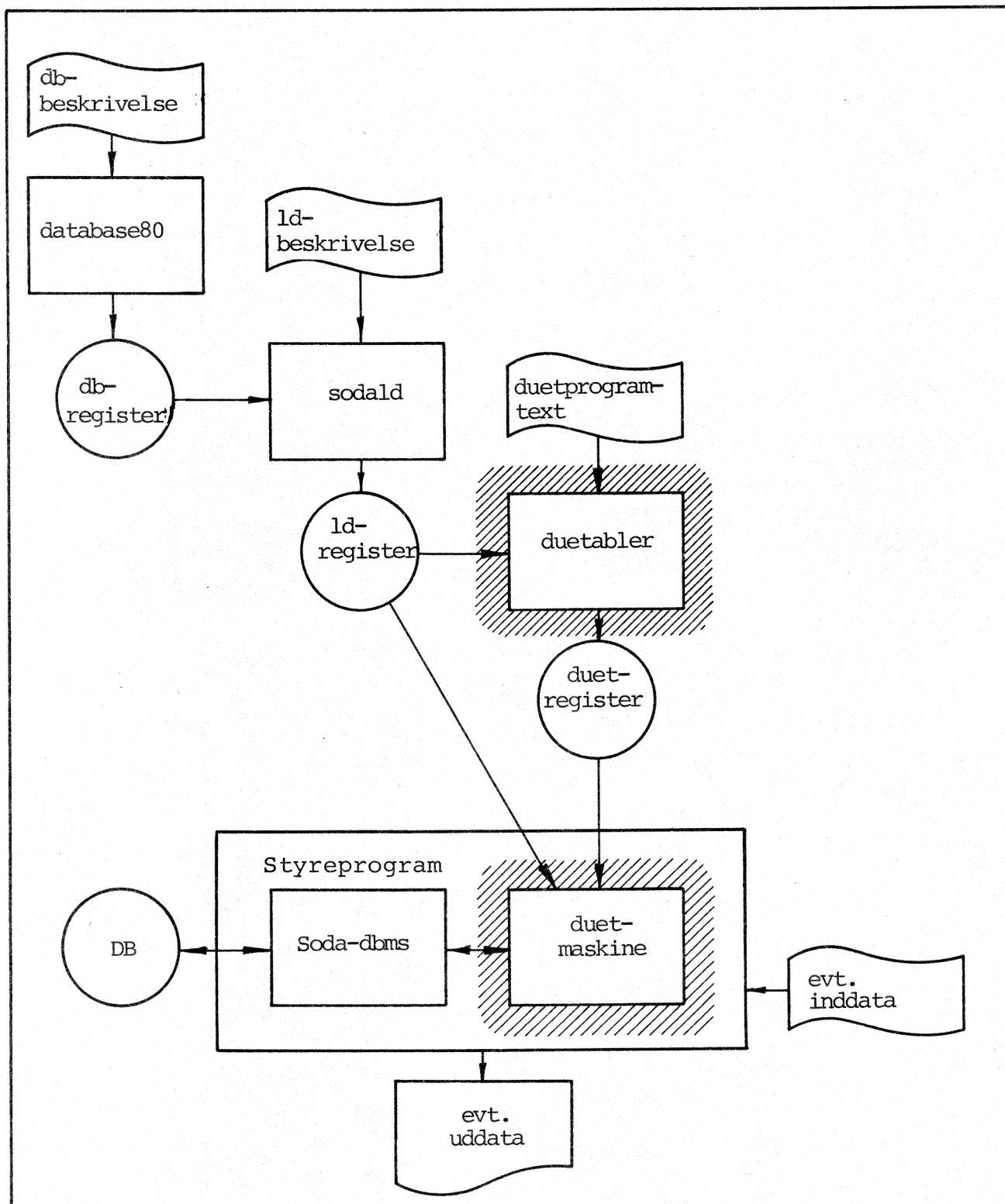
Til DES80 (ref. 3 og 5) er der kun behov for at kode rene duetprogrammer, og til telefakt (ref. 6) kan der laves duetkodede brugertrimninger, som skal indlægges i et eksisterende ramme-duetprogram. Her må duetprogrammerne følge de spilleregler, der er fastlagt af styreprogrammer og rammeprogram.

selvstændige
systemer

Man kan imidlertid sagtens lave helt nye applikationssystemer ved hjælp af duetsystemet med et selvstændigt duetprogram og styreprogram. Afsnit 5 indeholder anvisninger på etableringen af et styreprogram.

2. Systemoversigt

Dette afsnit giver en oversigt over duetsystemet og dets tilknytning til andre værktøjssystemer.



figur 2.1: Oversigt over duetsystemets relationer til omverdenen.

Figur 2.1 viser duetsystemet med de nødvendige omgivelser. Et færdigt applikationssystem består af et styreprogram sammen med det tilhørende ld-register og duet-register.

database80:
db-beskrivelse

Den første forudsætning for etableringen af et sådant applikationssystem er en databasebeskrivelse for den aktuelle database, baseret på 'database80' (ref. 1).

sodald:
ld-beskrivelse
ld-register

Dernæst skal den relevante delmængde af databasen beskrives i en lokal databasebeskrivelse, som oversættes af 'sodald' til et ld-register (ref. 2).

duetabler:
duetprogram

Duetoversætteren 'duetabler' læser et duetprogram sammenholdt med det tilhørende ld-register. Duetprogrammet bliver kontrolleret syntaktisk og semantisk, og hvis der ingen fejl findes, genereres et binært duetprogram, duet-register.

duetmaskine
soda-dbms

Dette oversatte duetprogram kan udføres af et styreprogram indeholdende fortolkeren 'duetmaskine' og soda-dbms'en, idet der kræves adgang til såvel ld-registret som duet-registret under kørslen.

duetblokke
duetarray

Duetprogrammet består af mindre enheder, duetblokke. Disse indlæses efter behov af duetmaskinen til et virtuelt lager, duetarray, hvis størrelse bestemmes af styreprogrammet. Ved reference fra en duetblok til en anden angives blokkens nummer samt et entrypunkt, der bestemmer, hvor i blokken indhoppet sker.

Der er flere fordele ved denne opdeling af duetprogrammet i blokke:

- Det bliver meget let at udarbejde brugertrimninger. Duetrammeprogrammets trimmepunkt er blot en reference til et entrypunkt i en særlig trimmeblok, hvor brugertrimningen kodes.
- Styreprogrammet kan køres i en mindre proces, idet det ikke er nødvendigt at have plads i core til hele duetprogrammet på én gang.
- Duetabler-oversætteren kan oversætte duetblokke enkeltvis og flette disse blokke ind i et eksisterende duet-register. Herved kan spares køretid ved ændringer i duetprogrammet og ved indlægning af nye brugertrimninger.

variable

d-array

Duetprogrammet kender alle de variable, der er erklæret i ld-beskrivelsens variabelafsnit, og de kan refereres enten ved deres navn eller nummer. Variablene findes i et fælles array, d-arrayet, hvori deres adresser er færdigberegnet af sodald-oversætteren. Da det oversatte duetprogram benytter disse adresser ved alle variabelreferencer, er det nødvendigt at genoversætte hele duetprogrammet efter ændringer i ld-beskrivelsens variabelafsnit.

duetinstruktion

Hver duetblok i et duetprogram består af et antal navngivne duetinstruktioner. Princippet i udførelsen af disse instruktioner ved hjælp

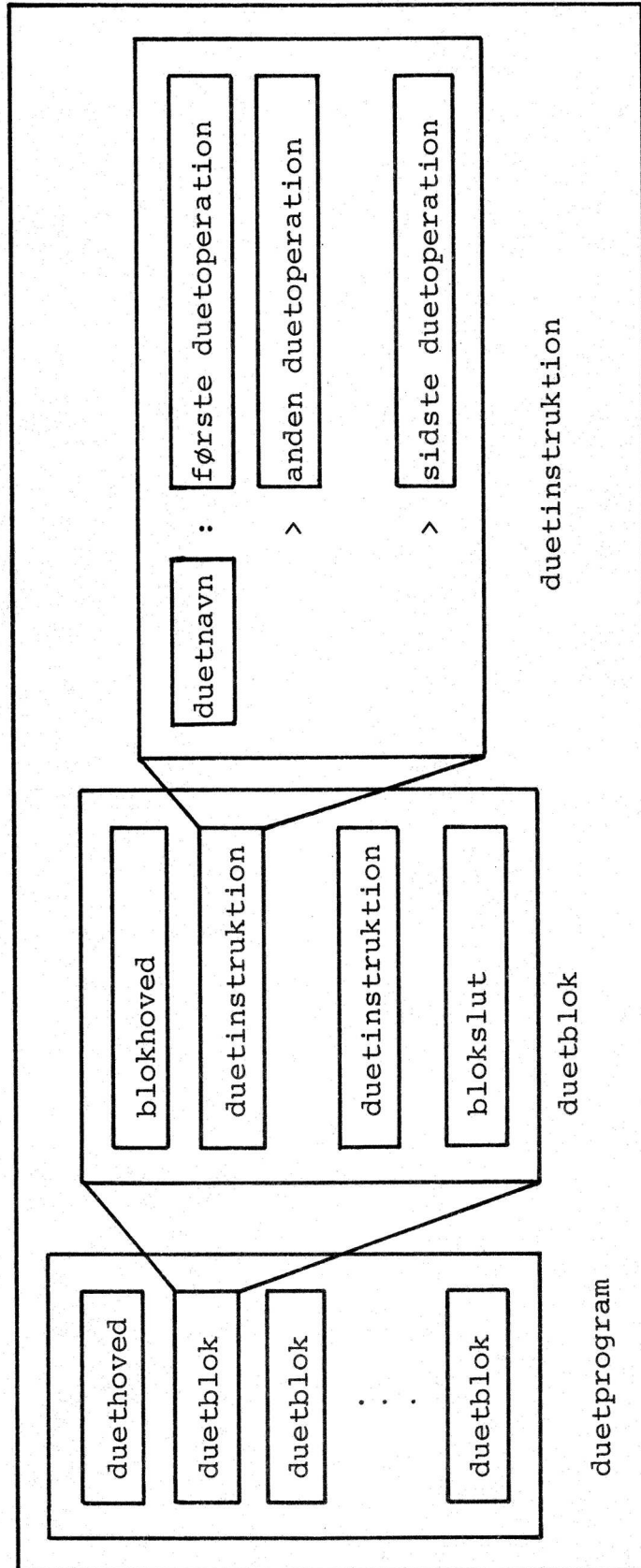
executeliste

af såkaldte executelister er udførligt forklaret i afsnit 3.1.

3. Duetsproget

I dette afsnit beskrives duetsprogets syntaks og semantik. Beskrivelsen er bygget på eksempler mens en formel syntaksbeskrivelse findes i appendix A.

3.1 Duetprogramstruktur



figur 3.1: Skematisk oversigt over duetprogrammets opbygning

Figur 3.1 viser skematisk opbygningen af et duetprogram som bestående af et duethoved og en eller flere duetblokke.

duetinstruk-
tion

Hver duetblok består af et blokhoved, et antal duetinstruktioner og et blokslut.

duetoperation
duetnavn

Endelig er en duetinstruktion opbygget af et duetnavn og en eller flere duetoperationer. Duetnavnet, der består af bogstavet 'd' efterfulgt af et heltal, identificerer instruktionen inden for den aktuelle blok. Navnet benyttes, når duetinstruktionen skal udføres, analogt til den måde en procedureidentifikation benyttes ved procedurekald i et algolprogram.

Rækkefølgen de enkelte duetinstruktioner skrives i indenfor en duetblok er ligegyldig, helt analogt til rækkefølgen af procedureerklæringer i en algolblok. Når en duetinstruktion aktiveres, udføres de enkelte operationer derimod altid i den anførte rækkefølge, ligesom de enkelte sætninger i en algolprocedurekrop.

executeliste

Udførelsesrækkefølgen for de enkelte duetinstruktioner i et program bestemmes dynamisk, primært styret af udførelsen af en særlig operation, der kaldes en executeliste. Denne indeholder en liste med duetnavne, som er lokale til den aktuelle blok, og de tilhørende instruktioner vil da blive udført i samme rækkefølge som navnene optræder i listen.

```

d315: execute  d12                ; executeliste
              d240
              d316
              s

      > modify  v_navn := 'skruer' ; variabelassign
              antal  := + 1
              s

      > compute v33(index) := v45 * (v12 + v16)
              s

      > print   <l 1 t6>   : 'saldo' ; trykning
              < 10 n5.2> : saldo
              s

      > if      saldo >1000.00 then d318 ; betinget operation

```

figur 3.2: eksempel på duetinstruktion

Figur 3.2 viser et eksempel på en duetinstruktion bestående af fem operationer. Enhver duetoperation indledes med en duetoperator der entydigt bestemmer det følgende format. Herefter følger en eller flere duetoperander adskilt med passende skilletegn, nøgleord eller numeriske operatorer.

For visse duetoperatorer vil formatet altid være fast og kræve et bestemt sæt duetoperander, mens andre duetoperatorer har et variabelt antal operander tilknyttet. I sidstnævnte tilfælde organiseres operanderne i en liste, hvis elementer normalt adskilles med lineskift. En sådan liste afsluttes altid med en separat linie med terminatoren 's' - det såkaldte duetslut.

duetoperator

duetoperand

duetslut

kommentar

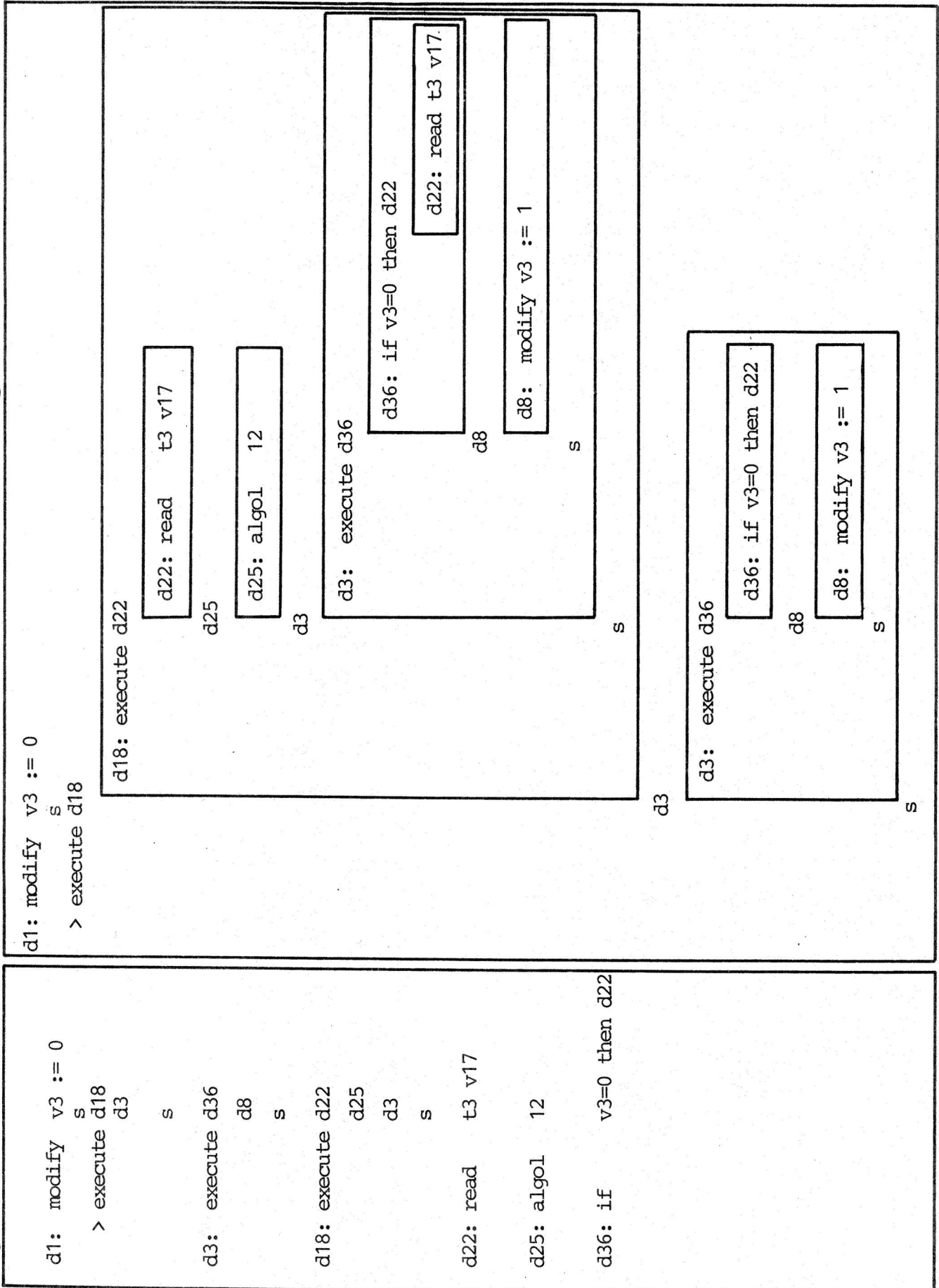
Alle linier i et duetprogram kan afsluttes med et semikolon og en kommentartekst, der ender ved linieskift. I udskriften fra duetableroversætterens vil sådanne kommentarer blive redigeret ind under hinanden (se afsnit 4.3). Det er tilladt at indføre tomme linier (evt. med en kommentar) overalt i programmet.

udførelse af
duetprogram

Udførelsen af instruktionerne såvel inden for en blok som inden for hele duetprogrammet sker i et antal dynamisk fastlagte niveauer. Hver gang man fra en executeliste beordrer udførelsen af en duetinstruktion, overføres kontrollen til et nyt niveau - i praksis ved, at der stakkes en pegepind på det aktuelle sted i executelisten, således at man senere, når den refererede instruktion er udført, kan gå videre i listen.

Når sidste instruktion i en executeliste er udført, returneres omvendt til foregående niveau, og når man når grundniveauet, dvs. det fiktive niveau, man befinder sig på før start, er udførelsen af programmet, subsidiært blokken, slut. Executelisten er således det helt fundamentale element i duetsproget og forståelsen af executelistens funktion bliver derfor helt afgørende for fornemmelsen af mulighederne i duet.

Figur 3.3 illustrerer princippet i den niveaumæssige afvikling af en duetblok. Vi antager, at instruktionen d15 netop er blevet aktiveret fra et sted benævnt 'p'. Til venstre ses det duetprogram, der udføres, og skemaet til højre viser, hvordan duetinstruktionerne genneløbes. Tallene længst til venstre benyttes som pegepinde i skemaet over programforløbet.



figur 3.3: Udførelse af en stump duetprogram med start i d1

3.1.1 Duethoved

Et duetprogram består, som vist i figur 3.1 af et duethoved og en eller flere duetblokke.

Duethovedet indeholder information til identifikation af duetprogrammet, såvel ved oversættelse som ved kørsel med det færdige duetprogram.

```
duetprogram 25/'teledata online'  
localdata 23
```

figur 3.4: duethoved

duetprogramnr.
og -navn

Figur 3.4 viser duethovedet for et duetprogram med duetprogramnr. = 25 og duetprogramnavn = 'teledata online', og dette duetprogram kører sammen med lokalbeskrivelse nr. 23. Duetprogramnavnet må højst være på 17 tegn.

3.1.2 Duetblokke

Duetblokkene er indført af to grunde. Dels for at muliggøre lokal navngivning af duetinstruktioner og dels som grundlag for en bruger-/programstyret segmentering, der kan blive nødvendig, hvis programmet er større end de 500 - 1000 duetoperationer (2048 ord oversat program), der er plads til i et duetprogram bestående af en enkelt blok.

bloknr

Alle duetblokke i et program er forsynet med et bloknr., der identificerer blokken entydigt. Dette bloknr. benyttes ved referencer fra en executeliste til ikke lokale duetinstruktioner, altså instruktioner i en anden blok.

entrypunkt

Da hver duetblok skal kunne oversættes separat, og da duetnavne er lokale inden for en blok, sker sådanne referencer til ikke lokale instruktioner ved hjælp af symbolske entrypunkter, der defineres i hovedet af hver blok.

Hver duetblok indeholder et blokhoved, at antal duetinstruktioner og et blokend (jf. figur 3.1).


```
begin 4: user 35 ; blok nr. 4 tilhører bruger 35
->      1 = d3
->      3 = d50
->     10 = d6
```

duetinstruktioner

```
end 4 ; samme nr. som i blokhoved
```

figur 3.5: blokhoved og blokend

I figur 3.5 ses blokhoved og blokend for en duetblok med 3 entrypunkter. Hvert entrypunkt definerer et entrynr svarende til et bestemt duetnavn, som skal eksistere inden for denne blok.

For bloknr, brugernr og entrynr gælder følgende værdiintervaller:

```
1 <= bloknr    <= 255
0 <= brugernr  <= 127
1 <= entrynr   <= 63
```

Definitionen af brugernr (user <brugernr>) kan udelades, hvis brugernr = 0.

Flere entrynumre kan pege på samme duetinstruktion. Entrynr 0 er altid underforstået som første duetinstruktion i blokken, og kan ikke redefineres.

aktivering af
duetblok

Når en duetblok aktiveres, sker det som nævnt ved en reference fra en executeliste i en anden blok. Referencen indeholder to oplysninger, nemlig et bloknr og et entrynr. Duetmaskinen undersøger nu, om den angivne duetblok allerede findes tilgængelig i lageret.

Hvis blokken findes i lageret, aktiveres straks den duetinstruktion, der hører til det angivne entrynummer. I dette tilfælde tager aktiveringen af en duetblok ikke meget længere tid end aktiveringen af en lokal duetinstruktion i den oprindelige blok.

duetarray

Hvis blokken derimod ikke findes i lageret, hentes den automatisk ind fra duetregistret til et duetarray, hvis størrelse er defineret af brugerprogrammet.

Er der nu en eller flere tilstrækkeligt store frie pladser i dette array, anbringes blokken forrest i den mindste af disse pladser, hvorefter alt går, som havde blokken været inde hele tiden. Er der derimod ikke plads, bestemmes den placering i arrayet, som efter visse kriterier er den optimale, idet en eller flere allerede udlagte blokke overskrives af den nye. Som grundlag for optimaliseringen optælles alle referencer til en blok i en særlig tæller, hvis værdi bliver et mål for, hvor ønskværdigt det er at undgå overskrivning. Ved udlægningen søges endvidere antallet af ledige småpladser holdt nede, så en egentlig "garbage collection" kan undgås.

kontrol af
bloknr

Hvis den ønskede blok slet ikke findes i duetregistret, reagerer systemet med en duetprogramfejl (se afsnit 5.3.2).

kontrol af
brugernr

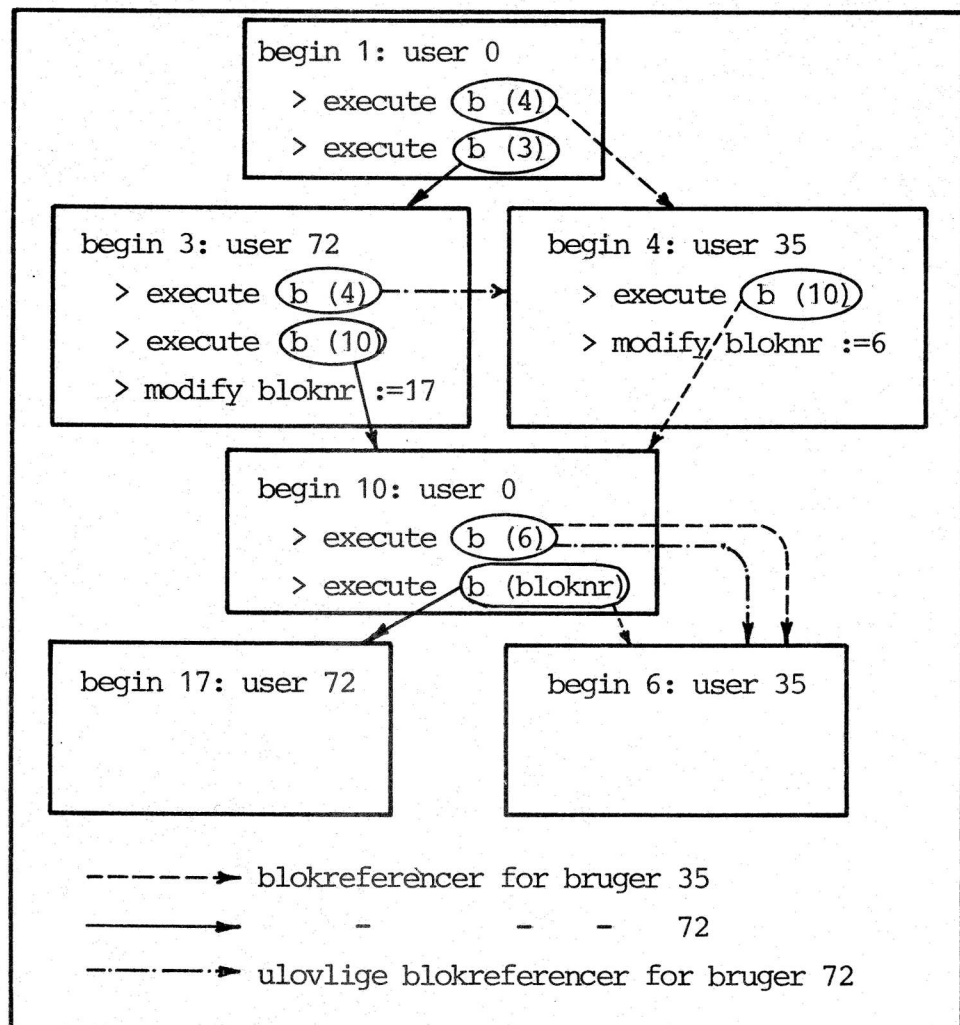
Inden udførelsen af en duetblok påbegyndes, kontrollerer duetsystemet at den nye bloks brugernummer er lovligt i forhold til den gamle blok:

Fra en blok med brugernr 0 kan programmet aktivere en hvilken som helst blok, men når der én gang har været aktiveret en blok med brugernr $\langle \rangle$ 0, registre-

res dette som kontrolbrugernr. Herefter tillades kun referencer til blokke med samme brugernr eller med brugernr 0. Kontrolbrugernummeret kan neutraliseres (nulstilles) ved at styreprogrammet kalder initialiseringsproceduren 'init_duetmaskine' (se afsnit 5.2), hvorefter det bliver muligt at behandle en anden brugers blokke.

Formålet med denne kontrol er at hindre at en brugertrimning kommer til at udføre en anden brugers program.

Figur 3.6 illustrerer denne kontrol.



figur 3.6: Kontrol af brugernr ved blokreferencer

Blok 1 er i eksemplet en programblok fælles for alle brugere. Blok 10 er en fælles trimmeblok mens de øvrige blokke er individuelle brugertrimninger. Fra blok 3 er det forbudt at aktivere blok 4. Fra blok 10 er det lovligt at hoppe til blok 6, hvis blok 10 selv var kaldt fra blok 3.

returnering

Når den udpegede duetinstruktion i en blokreference er færdigudført, returnerer programmet til den foregående blok, hvor det fortsætter behandlingen af executelisten.

Når et duetprogram startes op, sker det ubetinget med start i første duetinstruktion på blokken med laveste nummer.

3.1.3 Duetinstruktioner

Formatet af en duetinstruktion er vist længst til højre i figur 3.1 (side 2).

duetnavn

Duetinstruktionen identificeres af et duetnavn, der kan refereres fra duetoperationer i andre instruktioner. Duetnavnet erklæres foran første duetoperation i en instruktion ved bogstavet 'd' efterfulgt af et heltal og et kolon (se figur 3.7).

```
d32: execute d33
           d17
           s
>get      s2
>if       sodaresult >0 then d99
```

figur 3.7: eksempel på duetinstruktion

Et duetnavn skal erklæres i intervallet

```
d1 <= duetnavn <= d1023,
```

og oversætteren kontrollerer at duetnavnene er entydige inden for hver blok.

duetoperation

Efter duetnavnet følger den eller de duetoperationer, der indgår i instruktionen. Hver enkelt duetoperation skal afsluttes med linieskift, og starten af den næste operation skal markeres med tegnet '>'. Der kan indgå vilkårlig mange duetoperationer i en instruktion, principielt er det muligt at lade hele blokken bestå af en eneste lang duetinstruktion.

Den enkelte duetoperation består af en duetoperator og en eller flere duetoperander alt efter typen af operation. Duetoperationerne gennemgås systematisk i afsnit 3.2, mens de mulige operandtyper beskrives i næste afsnit.

3.1.4 Duetoperander

I duetoperationerne kan indgå forskellige typer operander, nemlig variabelreferencer, konstanter og duetnavne, som alle beskrives i dette afsnit. Derudover anvendes i forbindelse med db-operationer referencer til individsæt og individtyper som beskrevet i afsnit 3.2.13.

variabelre-
ferencer

Duetprogrammet kender og kan referere til alle variable erklæret i den tilhørende lokaldatabeskrivelse (se ref. 2 og 3) uden at der behøver at være genereret fieldvariabelerklæringer til dem.

variabel-
navn

Variable kan i flæng refereres ved navn eller ved nummer. Til en reference ved navn benyttes den fulde variabelidentifikator fra variabelerklæringen (bemærk specielt at også understreget space _ er betydende, modsat hvad der gælder for algolprogrammer).

variabelnr

Til en variabelreference ved nummer skrives blot bogstavet 'v' efterfulgt af variabelens nummer. Denne referencemetode kan dog ikke bruges i duetprogrammer baseret på en dataentry-lokaldatabeskrivelse, hvor variabelnumre er anonyme.

indiceret
variabel

Hvis en variabel er erklæret som array, kræves normalt en indicering i variabelreferencen som vist i figur 3.8.

variabelreferencer ved navn:

prisindex

varepris(prisindex)

variabelreferencer ved variabelnummer:

v1 ; simpel variabel

v25(2) ; indiceret variabel m. konst. index

v25(v1) ; - - - var. index

v28(v25(v1)) ; indicering i flere niveauer

figur 3.8: variabelreferencer

Indexet kan være konstant eller en variabel, som igen kan være indiceret i vilkårlig mange niveauer. Hvis index er konstant, udføres indexcheck under oversættelsen. Når index er en variabel, kan indexchecket først udføres på kørselstidspunktet, men oversætteren kontrollerer at alle indexvariable er erklæret uden decimaler.

simpel
variabel

En variabel uden arrayspecifikation kaldes en simpel variabel. Visse duetoperationer kræver at den refererede variabel er simpel.

variabeltype

Mange duetoperationer stiller krav til variabeltypen ved en variabelreference. I denne manual benyttes de betegnelser for variable med typerestriktioner, der er vist i figur 3.9.

variabelbetegnelse	tilladte variabeltyper
textvar	text
numvar	word, long, real, date, result
wordvar	word, date, result
heltalsvar	wordvar uden decimalangivelse
bitvar	bits
recnovar	recno, result.recno

figur 3.9: variabelbetegnelser

konstanter

Den anden type duetoperander er konstanter, hvor der i duetsproget skelnes mellem numeriske konstanter, tegnkonstanter og tekstkonstanter.

numeriske
konstanter

Numeriske konstanter er almindelige heltal eller decimaltal. Decimaltal kan af oversætteren lagres enten som et flydende tal (real) eller som et heltal med underforståede decimaler, alt afhængig den sam-

menhæng hvori konstanten forekommer.

automatisk
normalisering

Det kørende system vil altid sørge for automatisk normalisering, dvs. tage højde for forskelle i decimalantal mellem operander på venstre og højre side af en operator, jf. figur 3.10.

```
>if    v17 = 100 then ....
>if    v17 = 100.00 then ....
```

figur 3.10: automatisk normalisering

De to betingelser i figuren er ens, uanset om v17 er defineret med 0,1 eller 2 decimaler.

spill

Derimod vil duetsystemet principielt ikke kontrollere om der opstår spill ved assign til variable. Brugere der ønsker en sådan kontrol må derfor oversætte duetsystemet med spill.yes (jf. afsnit 5).

Tegnkonstanter skrives som et, to eller tre tegn (bogstaver, cifre og specialtegn undtagen punktum) omgivet af punktum, som vist i figur 3.11. En tegnkonstant lagres i programmet som et heltal (word = 24 bit) pakket af tegnenes isoværdier højrestillet, evt. med foranstillede nuller.

.a.	=	0	0	97
.ab.	=	0	97	98
.abc.	=	97	98	99
.?.	=	0	0	63
.1x:.	=	49	120	58
		<div style="text-align: center;"> </div> 8 bits 8 bits 8 bits		

figur 3.11: tegnkonstanter

tegnkonstanter En tegnkonstant er pr. definition numerisk, og den kan benyttes overalt hvor der tillades numeriske konstanter. Når der i manualen omtales numeriske konstanter menes der dermed såvel tal som tegnkonstanter.

tekstkonstanter Tekstkonstanter skrives som en tekst omgivet af apostroffer som vist i figur 3.12. En tekststreng kan være tom, og den øvre grænse for antallet af tegn er kun bestemt af, hvor meget der kan stå i en linie.

Den lagres på samme måde som i algol, dvs. venstre-stillet med mindst et afsluttende nultegn.

```
'lang tekst m. specialtegn ?'  
'kort'  
'           ; tom tekst
```

figur 3.12: tekstkonstanter

Alle konstanter der refereres i en blok, lagres sammen med de oversatte duetinstruktioner, således at konstanterne kun tager plads op i lageret sålænge den aktuelle blok er til stede i duetarrayet (jf. afsnit 3.1.2).

duetnavne Duetnavne udgør den tredje type duetoperander. Erklæring af duetnavne er beskrevet i afsnit 3.1.3, her drejer det sig kun om referencer til duetnavne.

En duetoperation kan henvise til en anden duetinstruktion i samme blok ved at anføre dennes duetnavn.

```

>execute d17
           d32
           s
>if      v35 = 1 then d0 else d33

```

figur 3.13: duetreference

Duetableroversætterens kontrollerer, at alle refererede duetnavne findes erklæret inden for blokken.

Foruden de erklærede duetnavne findes en standard duetinstruktion med navnet d0. Denne er en tom instruktion: der udføres intet når der refereres til d0.

værdielementer

Endelig findes i duetsproget begrebet værdielement, som dækker en kombination af de nævnte typer operander.

numerisk
værdielement

Et numerisk værdielement er således enten en talkonstant eller en tegnkonstant eller en numerisk variabel (simpel eller arrayelement).

tekst
værdielement

Et tekstværdielement er enten en tekstkonstant eller en tekstvariabel (simpel eller arrayelement).

3.2 Duetoperationer

I det følgende beskrives de enkelte mulige duetoperationer systematisk. Nedenfor findes en oversigt over samtlige duetoperationer klassificeret efter deres funktion:

Udførende operation	execute
Variabelassign	modify compute assign
Betingede operationer	if - then - else case - of action - of
Repeterende operationer	for - do while - do
Læseoperationer	getline read
Trykkeoperationer	print
Database operationer	get next lookup create put delete newset
Øvrige operationer	select exit algol

3.2.1 Execute

Den fundamentale udførende duetoperation er execute-listen, hvis funktion er beskrevet i afsnit 3.1.

```

d7: execute    d123          ; lokal duetreference
                d8, d36, d15
                b (16,5)      ; blokreference
                b(trimmeblok, 2) ; - , var.bloknr
                b(v15, v16)   ; - , var.bloknr
                                ;          og entrynr
                b(18), d4     ; - , uden entrynr
                s
    
```

figur 3.14: Executeliste

I figur 3.14 vises formatet af en executeliste. Den Den indledes med operatoren 'execute' og afsluttes med duetslut, tegnet 's'. Herimellem kan anbringes et vilkårligt antal aktionsreferencer, enten flere i samme linie med komma imellem, eller en enkelt i hver linie uden komma.

En aktionsreference kan enten være en lokal duetreference eller en blokreference.

lokal
duetreference

En lokal duetreference er blot et duetnavn ('d' efterfulgt af et heltal), dvs. reference til en anden duetinstruktion på samme blok (se afsnit 3.1.3)

Ved udførelsen af executelisten vil en lokal duetreference bevirke, at kontrollen midlertidigt overføres til den refererede duetinstruktion, men stedet i den aktuelle executeliste registreres og når instruktionen er udført, returneres kontrollen, så næste reference i listen kan udføres.

blokreference

En blokreference benyttes når man vil udføre duetinstruktioner på en anden blok. Ved en blokreference angives 'b' efterfulgt af bloknr og entrynr i parentes. Såvel bloknr som entrynr kan angives som en konstant (heltal) eller en variabelreference. I sidstnævnte tilfælde er det indholdet af den angivne variabel på kørselstidspunktet der bestemmer hvilken blok/entry der aktiveres.

Entrynummeret kan udelades i blokreferencen, i så fald underforstås den første duetinstruktion på blokken.

Bemærk at blokreferencer kun kan forekomme i en execute-liste. Alle andre duetoperationer kan kun referere til lokale duetnavne.

Oversætteren kan ikke kontrollere, at en blokreference er legal. Dels kan bloknummeret være variabelt, dels oversættes de enkelte blokke uafhængigt af hinanden, således at ingen kontroller udføres på tværs af blokgrænser. Derfor må blokreferencen kontrolleres af duetsystemet på kørselstidspunktet som beskrevet af afsnit 3.1.2.

Når duetslut nås i udførelsen af en executeliste, er denne udtømt. Det undersøges da, om der findes flere operationer i den aktuelle instruktion, og i så fald fortsættes med udførelsen af disse. Ellers returneres til foregående niveau i det dynamiske udførelseshierarki (jf. afsnit 3.1).

3.2.1.2 Programpunkter

En egentlig goto-operation kan ikke udtrykkes i duet. Den er i modstrid med den hierakiske udførelse af duetprogrammets executelister, hvor princippet hele tiden er det, at når en instruktion er udført i sin fulde konsekvens, returneres til foregående niveau. Udførelsen af en instruktion kan medføre overførelse af kontrollen til nye niveauer, men hvis programmet ikke havner i en uendelig løkke, vil det før eller senere vende tilbage til den oprindelige instruktion.

Men i visse situationer kan man have brug for at kortslutte udførelsen og vende direkte tilbage til et af de foregående niveauer, f.eks. ved datafejl eller til udhop af en programløkke før dennes stopværdi er nået.

Dette kan gøres ved hjælp af operationen exit i forbindelse med et programpunkt i executelisten.

```

>execute      d3
              d8
              d13
p2:          d13, d22
              d36
              s
    
```

figur 3.15: Executelite med programpunkt

programpunkt-
erklæring

Et programpunkt erklæres foran en linie i en execute-liste med bogstavet 'p' efterfulgt af et heltal i intervallet 1 - 9 og et kolon.

aktivt pro-
grampunkt

Når en executeliste udføres og en linie med en programpunkterklæring skal behandles, stakkes udover pegepinden for det aktuelle sted i listen også en markering af det specificerede programpunkt. Herved bliver det til et aktivt programpunkt.

På et vilkårligt tidspunkt kan der således være mange aktive programpunkter og for den sags skyld kan der være flere samtidig med samme nummer. Til hvert aktivt programpunkt er der imidlertid knyttet et niveau svarende til det dynamiske niveau i programmet, den tilhørende executeliste er aktiveret på (og svarende til aktuel position i stakken).

passivt pro-
grampunkt

Når der returneres fra den sidste instruktion i den linie, hvortil der er knyttet et programpunkt, gøres dette passivt igen i forbindelse med den naturlige afstakning. I eksemplet vil p2 på det aktuelle niveau altså kun være aktivt så længe d13 eller d22 eller derfra aktiverede instruktioner er under udførelse.

Hvis d13 aktiveres andet steds fra, vil p2 ikke blive aktiv (medmindre p2 også dér er erklæret som programpunkt).

exit

Med exit-operationen (beskrevet i afsnit 3.2.15) kan man referere til et programpunkt med den effekt, at når exit-operationen udføres, foretages en ubetinget returnering til det niveaumæssigt nærmeste aktive programpunkt med samme nummer som referencen. Herfra fortsættes så med næste linie af executelisten, uanset om den foregående linie eller de på mellemliggende niveauer aktiverede executelister er færdigbehandlede eller ej.

Hvis der ikke findes noget aktivt programpunkt med det givne nummer, returneres helt tilbage til grun-niveauet, dvs. duetprogrammet afsluttes.

select

Med operatoren select (se afsnit 3.2.14) kan man defi-nere automatisk exit til et programpunkt ved konsta-tering af datafejl, programfejl og/eller systemfejl. Der kan vælges forskellige programpunkter til forskel-lige fejl.

3.2.1.3 Betinget oversættelse

Executelisten kan udbygges med undertrykkelsesspeci-fikationer, der bevirker betinget oversættelse.

```
>execute    d35
            t1 p7: d123, d8, b(18)
                d32
                d13
            t2    b(bloknr, entrynr)
                s
```

figur 3.16: Executeliste med under-trykkelsesspecifikationer

undertryk-
kelsesspeci-
fikation

En undertrykkelsesspecifikation angives med bogsta-vet 't' (for test) efterfulgt af et heltal i interval-let 1 - 9. Den bevirker at hele linien bliver under-trykt ved en normal oversættelse, og dermed ikke kommer til at indgå i det færdige program. Linien oversættes kun hvis der ved kaldet af duetabler er angivet en include-parameter, hvis værdi er større end eller lig med den t-værdi, der er specificeret for linien.

Som eksemplet viser kan man også undertrykke en linie, hvis der er erklæret et programpunkt. Dette vil i så fald aldrig blive aktivt.

Hverken undertrykkelsespecifikation eller programpunkterklæring kan skrives i samme linie som execute-operationen. Hvis disse skal gælde den første reference i executelisten, må operationen skrives på en linie for sig, som vist i figur 3.17.

```
>execute
      p8:  d32
           d33
           s
```

figur 3.17: Programpunkt på 1. duetreference

3.2.2 Modify

Med modify-operationen kan variable af alle typer assignes ved simpel flytning af en værdi til en variabel. Desuden kan operationen bruges ved akkumulering i numeriske variable. Der kan derimod ikke udføres beregninger, og assignment sker uden brug af arbejdsvariable, hvorved modify bliver den hurtigste form for assign.

Formatet af modifyoperationen er en liste med variabel længde, afsluttet med duetslut. Figur 3.18 viser nogle af mulighederne i en modifyliste.

```
>modify  index := 3
          arrayvar(index) := simpel_word
          tæller :=+ 1
          saldo :=- indbetaling

          textvar := 'tekstkonstant'

          array2 := array1
          array := 0
          s
```

figur 3.18: Modifyliste

Hver linie består af en venstreside, en assignoperator og en højreside.

venstre side
er varref

Venstresiden skal normalt være en reference til en simpel variabel eller til et element af en arrayvariabel. I det sidste tilfælde skal index være en heltalskonstant eller en simpel heltalsvar (jf. afsnit 3.1.4). Der kan altså ikke indiceres i flere

niveauer på venstre side.

assignopera-
tor

Assignoperatoren kan være := for almindelig assign,
og :+ eller :- for akkumulering.

højre side er
værdielement

Højresiden skal være et værdielement, dvs. en
konstant eller en variabelreference. På højre side må
man gerne indicere i flere niveauer, og der er en
række specialoperationer til rådighed, som beskrives
i det følgende.

Højresiden skal typemæssigt svare til venstresiden.
Således kan et numerisk værdielement kun assignes
til en numerisk variabel, og et tekstelement kun til
en tekstvariabel.

Den følgende detaljerede beskrivelse af modifyopera-
toren opdeles i grupper efter arten af højresiden:

1. numerisk assign
2. tekstassign
3. anonymt assign
4. specialassign

3.2.2.1 Numerisk assign

Figur 3.19 viser de mulige former for numerisk assign.

```
>modify    index := 3
           tæller :=+ 1
           saldo :=- indbetaling
           array(index) := simpel_var
           kode := .xy.
           varegruppe := digits (6,4) of varenr
           s
```

figur 3.19: numeriske assign

Til numerisk assign kan benyttes alle tre assign-operatorer med følgende betydning.

```
:=      numvar      := højreside  
:+      numvar      := numvar + højreside  
:-      numvar      := numvar - højreside
```

Venstresiden skal være en numerisk variabelreference, dvs. reference til en simpel variabel eller et arrayelement af typen word, long, real, date eller result.

Højresiden kan være en numkonstant/tegnkonstant eller en numerisk variabelreference.

digits

På højresiden kan endvidere benyttes operatoren 'digits', hvis funktion illustreres af figur 3.20.

```
varenr = 7253124  
          |  
          | digits (6,4) af varenr
```

figur 3.20: digits

Digits-operatoren udtrækker en gruppe cifre fra en variabel. Variablens cifre tænkes nummereret fra højre således at mindst betydende ciffer har nummeret 1.

digits (a, b) of c

hvor $a \geq b$, udtrækker da et heltal bestående af cifrene fra og med cifferposition a til og med cifferposition b, og afleverer dette tal som resultat. Resultatet af operationen i figur 3.20 er altså 253.

Bemærk: Der tages ikke hensyn til underforståede decimaler i den variabel, digits opererer på. Resultatet af

digits (2,1) of saldo

hvor saldo har 2 decimaler, vil altså netop være decimalerne.

automatisk
normalisering

Ved numeriske assign udføres automatisk normalisering, dvs. der tages automatisk hensyn til forskelle i decimalantal mellem venstre side og den resulterende højre side, og til forskelle i type.

```

>modify w_2_dec := 123.4
          w_heltal := 123.4
          w_var   := l_var
          w_var   := r_var
          s
```

figur 3.21

I eksemplet figur 3.21 vil w_2_dec (erklæret med 2 decimaler) efter operationen have værdien 12340 (med betydningen 123.40) mens w_heltal indeholder 123.

Det er tilladt at assigne en longværdi eller en realværdi til en wordvariabel, men det kørende system kontrollerer ikke, om værdien kan rummes i denne variabel (jf. afsnit 3.1.4).

3.2.2.2 Tekstassign

```

>modify adresse      := 'falkoner alle 90'
      txtvar(2)      := txtvar(1)
      varnavn        := name (v17)
      varnavn        := name (var (varnr))
      s
```

figur 3.22: Tekstassign

Til tekstassign kan kun benyttes assignoperatoren := og venstresiden skal være en reference til en variabel af typen tekst (simpel eller arrayelement).

Højresiden i tekstassign kan enten være en tekstkonstant eller reference til en tekstvariabel.

name

På højresiden kan endvidere benyttes operatoren 'name', der som resultat afleverer navnet af den specificerede variabel, i overensstemmelse med den sprogkode, der blev benyttet under ld-oversættelsen. Denne variabel kan specificeres direkte som i

```
name (v17)
```

der giver navnet på variabelen v17.

name(var)

Variablen kan også specificeres indirekte ved hjælp af operatoren 'var', f.eks.

```
name (var (v32))
```

Her indeholder v32 variabelnummeret på den variabel, hvis navn ønskes.

Oversætteren kontrollerer at venstresidevariablen er stor nok til at rumme den resulterende tekst på højre side.

3.2.2.3 Anonymt assign

```

>modify sort_krit := var (brugernøgle)
s
    
```

figur 3.23: Anonymt assign

var

Anonymt assign kan udføres ved hjælp af operatoren 'var' anbragt yderst på højre side. I eksemplet figur 3.23 skal variabelen brugernøgle indeholde et variabelnummer, og det er den derved udpegede variabel, der assignes til sort_krit.

Venstresiden skal være numerisk (og bør være en long). Da oversætteren ikke kan kontrollere typen af den resulterende højreside, gælder følgende konventioner for flytningen:

result.højreside		
word	} afleveres som numerisk værdi med automatisk hensyntagen til den modtagende variabels type og decimalantal	
long		
real		
date		
tekst	6 første tegn	} afleveres som bit-mønster uden decimalomregning
bits	4 første bytes	
recno	2 bytes	

Hvis den udpegede højresidevariabel er en arrayvariabel, benyttes dennes første element efter samme regler.

3.2.2.4 Specialassign

Ved specialassign forstås assign til noget, der ikke er en enkelt numvar eller textvar.

Det gælder

- flytning af et helt array
- nulstilling af et helt numerisk array
- assign til en bitvar og
- assign til en recno var

```

>modify   array1      := array2
          nøgle_aggr := individ_nøgle
          gemt_recno  := recno_result
          array2      := 0
          s
```

figur 3.24: Specialassign

Assignoperatoren ved specialassign kan kun være :=.

flytning af
array

Til flytning af et helt array skrives på begge sider af assignoperatoren en arrayidentifikator uden angivelse af index.

Ved denne type flytning kræves absolut typeoverensstemmelse mellem venstre og højre side. For numeriske arrays kræves endvidere at de to arrays er erklæret med samme decimalantal, mens der for arrays af type text/bits forlanges, at længden af de enkelte elementer er ens i begge arrays.

Hvis arrayene er af forskellig længde, flyttes kun så mange elementer som bestemt af det korteste array. Endelig tillades ikke flytning af mere end 2047 ord ialt.

nulstilling
af numarray

Nulstilling af et helt array kan gøres ved på venstre at skrive en arrayidentifikator uden index, og på højre side tallet 0. Denne mulighed gælder dog kun for numeriske arrays.

assign
bitvar

En bitvariabel kan overflyttes til en anden bitvariabel forudsat de to variable har samme længde.

assign
recnovar

Endelig kan en variabel af typen recno assignes, enten med værdien af en anden recnovariabel, eller med tallet nul.

3.2.3 Compute

Compute-operatoren kan bruges til at assigne numeriske variable med værdien af generelle numeriske udtryk. Der kan udføres beregninger med brug af de almindelige numeriske operatorer og paranteser, og den resulterende værdi kan assignes til flere variable ligesom i algol. Compute kan derimod ikke anvendes ved tekstassign.

Formatet af compute-operationen er en liste af variabel længde afsluttet med duetslut (tegnet 's'), se figur 3.25.

```
>compute v1:= v12(3):= 5
        saldo := saldo - indbetaling(lbnr)
        pris(priskode + 2) := 82 * (v13(v1) + v17)
        s
```

figur 3.25: Computeliste

Hver linie, der udgør et assign i sig selv, består af en venstreside, et assignsymbol (:=) og et numerisk udtryk på højre side.

venstreside
er generel
varref

Venstresiden er en reference til en eller flere simple numeriske variable eller elementer af numeriske arrays. Indexet på et arrayelement kan her - som det eneste sted i duetsproget - være et generelt numerisk udtryk efter samme regler som gælder for højresiden. Hvis der er flere variabelreferencer på venstresiden adskilles disse med assignsymbolet.

højreside er
numerisk ud-
tryk

Højresiden er et numerisk udtryk. Som operander kan optræde referencer til numeriske variable (simple eller arrayelementer ligesom på venstre side), samt numeriske

konstanter (heltal og decimaltal), og tegnkonstanter. Operanderne adskilles med de sædvanlige numeriske operatorer,

+ - * /

samt parenteser.

Højresiden kan udarte til en enkelt numerisk variabel eller konstant, men i disse tilfælde er det normalt billigere at bruge modifyoperationen.

real arbejds-
register

Alle beregningerne i en computeliste sker med brug af et arbejdsregister, der er en real variabel, med sædvanlig hensyntagen til eventuelle decimalangivelser for de refererende variable. Dette betyder imidlertid, at der mistes nøjagtighed, hvis der i beregningerne indgår longværdier med flere betydende cifre end der kan rummes i 36 bits.

3.2.4 Assign - of

selektivt
assign

Denne operation, der kaldes selektivt assign, benyttes til at assigne en simpel heltalsvariabel ud fra værdien af en testvariabel.

```

>assign   værdikode := test_var of
          .xy.   : 3
          973    : kode
          v32    : .abc.
          s
```

figur 3.26: Selektivt assign

assignliste

Operationen indeholder en assignliste af variabel længde over testværdier (til venstre for :) og de tilsvarende assignværdier, se figur 3.26. Både testværdier og assignværdier kan være simple heltalsvariable, heltalskonstanter eller tegnkonstanter, men ingen af dem må fylde mere end 24 bits.

assignvariabel
testvariabel

Den variabel der skal assignes skal være en simpel heltalsvariabel. Testvariablen kan derimod være en vilkårlig numerisk variabel, simpel eller arrayelement, blot må den ikke være af type real eller have decimalangivelse. Assignvariabel og testvariabel kan eventuelt være samme variabel.

Assignlisten kan enten afsluttes med duetslut (som i figur 3.26) eller med en alternativ værdi eller aktion (figur 3.27 og 3.28).

```

>assign   v2   := v1 of
          1    : 100
          2    : 0
          else  v35
```

figur 3.27: Selektivt assign med alternativ værdi

```

>assign type := kode of
    .ov. : 1
    .ok. : 2
    .ok2.: 2
    else  d19      ; fejlaktion
    
```

figur 3.28: Selektiv assign med alternativ
aktion

Operationen fungerer på den måde, at værdien af testvariablen sammenlignes successivt med testværdierne i listen. Når der mødes en testværdi der er lig med værdien af testvariablen, standses sammenligningerne, og den tilsvarende assignværdi tildeles assignvariablen.

Hvis ingen af testværdierne passer på testvariablen, afhænger reaktionen af, om assignlisten er afsluttet med duetslut, alternativ værdi eller alternativ aktion.

alternativ
værdi

En alternativ værdi angives med 'else' efterfulgt af en simpel heltalsvariabel, en heltalskonstant eller en tegnkonstant, og denne værdi tildeles ubetinget assignvariablen, hvis ingen af testværdierne passede.

alternativ
aktion

Hvis der er anført en alternativ aktion, dvs. 'else' efterfulgt af et duetnavn, er det denne duetinstruktion, der i stedet udføres, hvis der ikke blev fundet en testværdi.

duetslut

Endelig kan der være angivet duetslut, hvilket svarer til en tom alternativ aktion som 'else d0'. I dette tilfælde udføres der intet.

3.2.5 If - then

Betinget udførelse af duetinstruktioner kan ske ved hjælp af if-then-operationen, hvis format fremgår af eksemplerne i figur 3.29.

```
> if v32(v14(2)) <> 0 then d19 else d20

> if saldo < kreditgrænse then d930

> if kode = .t13. then d14 else d365
```

figur 3.29: if-then-operationer

numerisk
relation

Mellem if og then skal der være angivet en enkelt numerisk relation, hvor der som relationsoperatorer kan benyttes symbolerne

```
< : mindre end
> : større end
= : lig med
<>: forskellig fra
```

Der kan ikke benyttes sammensatte symboler (f.eks. <=) eller sammensatte relationer (f.eks. 'and' og 'or'). Venstresiden i en relation skal være en numerisk variabelreference (simpel eller array-element). Højresiden kan desuden være en numerisk konstant eller en tegnkonstant, altså et sædvanligt numerisk værdielement.

Hvis betingelsen er opfyldt (relationen er sand), udføres den duetinstruktion der er angivet efter then. Hvis betingelsen ikke er opfyldt udføres duetinstruktionen angivet efter else.

Hvis else er udeladt, udføres intet i else-situationen, svarende til 'else d0'.

3.2.6 Case - of

Case-afgreningen i duet ligner funktionsmæssigt case-aktioner i algol.

```
> case test_var of
  1: d32
    d318
    d0
  4: d32
    else d27
```

figur 3.30: case-operation

Formatet af en case-operation fremgår af figur 3.30. Tallene foran duetnavnene virker som kommentar og er redundante. De kan udelades, men hvis et tal er anført, kontrollerer oversætteren at det passer til placeringen i listen.

Testvariablen kan være en vilkårlig numerisk variabel, simpel eller arrayelement, blot må den ikke være af type real eller have decimalangivelse.

Ved udførelsen af en case-operation aktiveres den duetinstruktion, hvis placering i listen over duetnavne stemmer overens med værdien af testvariablen.

Hvis testvariablens værdi er mindre end 1 eller større end antallet af duetnavne i listen, udføres ubetinget instruktionen efter 'else'.

Case-listen kan eventuelt afsluttes med duetslut ('s') i stedet for 'else', hvilket svarer til 'else d0'.

3.2.7 Action - of

selektiv
afgrening

Denne operation, der kaldes selektiv afgrening, er i princippet meget lig selektivt assign (afsnit 3.2.4). Den bevirker udførelse af en duetinstruktion betinget af værdien af en testvariabel.

Hvor case-afgreningens værdiområde (jf. afsnit 3.2.6) er begrænset til en række konsekutive positive heltal fra en og opefter, er værdiområdet ved selektiv afgrening i praksis næsten ubegrænset. Til gengæld må der foretages en søgning i en liste over de enkelte relevante værdier. Den aktuelle situation må så afgøre hvilken af disse to muligheder, der er mest fordelagtig.

> action	text_var	of
	.xy.	: d19
	973	: d318
	v12	: d0
	else	d27

figur 3.31: selektiv afgrening

Operationen indeholder en aktionsliste af variabel længde over testværdier og tilsvarende duetaktioner. Listen afsluttes med duetslut ('s') eller en alternativ aktion (som i figur 3.31).

Testvariablen kan være en vilkårlig numerisk variabel, simpel eller arrayelement, blot må den ikke være af type real eller have decimalangivelse. Som testværdier kan benyttes simple heltalsvariable, heltalskonstanter og tegnkonstanter. Testværdierne må ikke fylde mere end 24 bits.

Ved udførelsen af en action-operation sammenlignes testvariablen successivt med testværdierne i listen. Hvis der findes en testværdi, der er lig med værdien af testvariablen standses sammenligningerne, og den tilsvarende duetinstruktion aktiveres.

Hvis ingen af testværdierne stemmer overens med testvariablen, er operationen uden virkning, med mindre der med 'else' er anført en alternativ aktion. I så fald udføres den der anførte instruktion.

3.2.8 For - do

Programløkker laves i duetsproget ved hjælp af for-do-operationen eller med while-operationen (afsnit 3.2.9).

```
> for kontrol_var := start, slut do d32  
> for v17 := 3,5 do d100
```

figur 3.32: for-do-operationer

Formatet af en for-løkke fremgår af eksemplerne i figur 3.32.

Kontrolvariablen skal være en simpel heltalsvariabel mens start- og slutværdi kan være heltallige numeriske værdielementer.

Operationen udføres på samme måde som algol-sætningen

```
for kontrol_var := start step 1 until slut do ...;
```

blot med den forskel, at i duet udregnes slutværdien en gang for alle før løkken påbegyndes. Man kan derfor ikke afbryde løkken indefra ved at ændre værdien af slut-elementet.

En afbrydelse af løkken indefra kan derimod fremkaldes ved at ændre værdien af kontrolvariablen, eller ved hjælp af exit-operationen (se afsnit 3.2.15).

3.2.9 While - do

En while-løkke i duet har format som vist i figur 3.33.

```
> while testvar <> 0 do d313
```

figur 3.33: whileoperation

Mellem while og do skal anføres en numerisk relation, der skal opfylde de samme syntaktiske krav som relationen i if-then (jf. afsnit 3.2.5).

Operationen virker således:

Hvis relationen er sand, udføres den angivne duetinstruktion, hvorefter relationen testes igen. Når relationen giver resultatet false, udføres duetinstruktionen ikke, og operationen er dermed afsluttet.

3.2.10 Getline

read_general

Getline-operationen bevirker et kald af læse-proceduren 'read_general'. Herved læses der en tekststreng fra zonen 'readz', som af styreprogrammet skal være åbnet til inddatatekstarealet.

> getline textvar

figur 3.34 getline

Parameteren til getline-operationen er en tekstvariabel, hvori hele den læste tekststreng afleveres. Derudover lagres teksten opdelt i felter til brug for læseoperationen 'read' (afsnit 3.2.11).

strengslut

Getline afslutter læsningen, når der i inddata mødes et tegn, der i systemets tegnklassetabel er defineret som strengslut-tegn (se afsnit 3.2.11). Det er normalt tegnene lineskift (ISO-værdi 10, nl) og end_medium (ISO-værdi 25, em). Hvis der ikke findes strengsluttegn, standser læsningen efter 150 tegn, svarende til den længste linie der kan læses af 'read_general'.

Hvis den læste linie er længere end den tekstvariabel, der er anført som parameter til getline, flyttes kun så meget af linien, som kan rummes i tekstvariablen. Hele linien er dog altid til rådighed for fortolkning med duetoperationen read.

I visse programtyper er det mest hensigtsmæssigt at brugerprogrammet selv aktiverer indlæsningen af en tekststreng ved et kald af 'read_general'. I så fald må det tilhørende duetprogram ikke benytte getlineoperationen.

3.2.11 Read

En tekststreng leveret af getline-operationen eller 'read_general' kan opfattes som en række felter, der successivt kan erkendes ved hjælp af duetoperationen 'read'.

Read-operationen kan

- læse et felt fra inddatastrengen (numerisk felt, tekstfelt eller tegnfelt) og
 - udføre værdigrænsekontrol på feltet
- eller
- assigne standardværdi for et felt.

3.2.11.1 Klassificering af felter

tegnklasse-
tabel

Fortolkningen af en linie styres af en tegnklassesetabel, der defineres af duetsystemets initialiseringsprocedure 'init_duet2', men kan redefineres af styreprogrammet. For hver indlæst ISO-værdi fastlægger tabellen en tegnklasse og en intern tegnværdi.

Den interne tegnværdi er i standardtabellen lig ISO-værdien, undtagen for store bogstaver (ISO-værdi 65-93), der ændres til små bogstaver (97-125).

Tegnklassen benyttes til at fastlægge afgrænsningen af en tekststreng, samt grænserne mellem og arten af felterne i den læste tekststreng. Figur 3.35 viser de eksisterende tegnklasser samt standardinitialiseringen af disse.

klasse	betydning	tegn
<9	ulovlig	
9	blind	
10	strengslut	nl em (ny lin/end medium)
11	text-delimiter	' (apostrof)
12	kommentar-delimiter	" (gåseøjne)
13	ciffer	0 1 2 3 4 5 6 7 8 9
14	alm. delimiter	sp, (space og komma)
15	fortegn/std.mærke	- (minus)
16	decimalpkt	. (punktum)
>16	øvrige tegn	bogstaver, specialtegn

figur 3.35: tegnklasser

kommentarfelt

Et kommentarfelt er et antal tegn angivet mellem to kommentar-delimiters (gåseøjne). Kommentarfelter overspringes altid af read.

tekstfelt

Et tekstfelt er et antal tegn angivet mellem to textdelimiters (apostroffer).

numerisk felt

Et numerisk felt består af cifre, eventuelt med et decimalpunktum og/eller et foranstillet minus-tegn. Efterstillet minustegn kan ikke læses.

tegnfelt

Et tegnfelt består af enkelttegn (bogstaver, specialtegn, og/eller cifre), der ved læsningen pakkes som en tegnkonstant.

strengslut

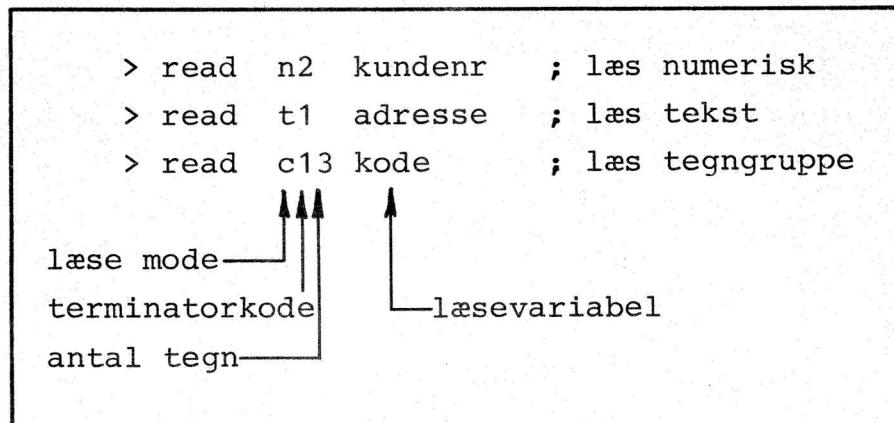
Strengsluttegn (nl/em) kan ikke indgå i noget felt. Det betyder, at hvis den afsluttende text-/kommentardelimiter mangler, afsluttes teksten/kommentaren ved strengslut.

alm. delimi-
ter

Almindelige delimiters (sp/,) kan indgå i tekstfelter og kommentarfelter, men afslutter numeriske felter og tegnfelter.

3.2.11.2 Læsespecifikation

Read-operationen styres af en læsespecifikation indeholdende en læsemode og en læsevariabel. Med hvert kald af read fortolkes ét felt af inputstrengen, og værdien af dette felt overføres til læsevariablen, hvis det er i overensstemmelse med den anførte læsemode.



figur 3.36: read-operation.

læsemode

Læse mode angiver arten af det felt, der skal læses, samt måden hvorpå det skal læses og re-præsenteres i den angivne læsevariabel.

n
numerisk

n angiver læsning af et numerisk felt, dvs. et heltal eller decimaltal med evt. fortegn. Læsevariablen skal være numerisk, og værdien afleveres i overensstemmelse med variabelens type og decimalangivelse.

t
 tekst

t står for læsning af et tekstfelt med om-
 givende tekstdelimiters (apostroffer). I
 læsevariablen, som skal være af type tekst,
 afleveres tegnene mellem de to tekstdelimi-
 ters på normal tekstform.

c
 chars

c angiver læsning af et tegnfelt (chars).
 Hertil skal specificeres hvor mange tegn,
 der højst må indgå i feltet (1, 2 eller 3).
 Læsevariablen skal være en heltalsvariabel
 (word), og de læste tegn repræsenteres heri
 på samme måde som tegnkonstanter lagres i
 duetprogrammet (se afsnit 3.1.4).

terminator-
 kode

Terminatorkoden definerer hvilken type delimiter
 der tillades som terminator for det aktuelle
 felt. Der gælder følgende koder:

- 0: terminatoren læses ikke med, og kontrol-
 leres ikke.
- 1: terminatoren skal være alm. delimiter
 (klasse 14)
- 2: terminatoren skal være strengslut-deli-
meter (klasse 10)
- 3: terminatoren kan være alm. eller streng-
slut-delimiter

resultatvar
 readterm

Efter læseoperationen fortæller resultatvariablen
 'result.readterm' hvilken terminator der blev
 læst:

result.readterm = 0: ingen terminator læst
 > 0: ISO-værdi for læst terminator

Hvis der læses terminator, bliver alle sammenhængende terminatorer læst, frem til næste felt eller evt. strengslut. Det er da altid værdien af den sidste terminator, der registreres og kontrolleres.

flere
læsespec

Der kan angives flere læsespecifikationer i en read-operation, hvormed menes, at der for det pågældende felt i input tillades flere forskellige arter, se figur 3.37.

```
> read  n1 tekstnr,  t1 tekst,  c13 kode
```

figur 3.37: read-operation med flere læsespecifikationer

Ved udførelsen af en read-operation kontrolleres det, at inputfeltet opfylder de syntaktiske krav svarende til læsemode i den første læsespecifikation. Er dette ikke tilfældet, forsøges læsning i overensstemmelse med en eventuel følgende læsespecifikation i den aktuelle operation. Således fortsættes, indtil feltet konstateres syntaktisk korrekt, eller der ikke er flere læsespecifikationer. I sidstnævnte tilfælde udløses en fejlaktion ved kald af proceduren 'duet-datafejl' (jf. afsnit 5.3.1).

```
> read  s,  c13 liniekode
```

figur 3.38: readspecifikation med læsning af foranstillet delimiter

s

Foran første læsespecifikation kan man angive 's'. Det betyder, at hvis der på det aktuelle sted i inputstrengen står en endnu ikke læst almindelig delimiter (sp/,), læses det efterfølgende felt ikke.

Denne markering har kun mening ved læsning af første felt i linien, eller hvis den sidst udførte readoperation læste med terminatorkode 0.

3.2.11.3 Værdigrænsekontrol

Når et felt er fundet syntaktisk korrekt, udføres en eventuel værdigrænsekontrol bestemt af læsevariablens værdispecifikation i ld-beskrivelsen. For en numerisk variabel kontrolleres at værdien ligger indenfor dennes tilladte værdispektrum. For en tekstvariabel kontrolleres at det læste antal tegn tilhører det tilladte interval.

For variable, der ikke har angivet værdigrænser kontrolleres blot, at værdien kan rummes i den aktuelle variabel.

Hvis den læste værdi ligger udenfor det tilladte værdispektrum, udløses en fejlaktion.

3.2.11.4 Standardværdi

eksplicit
standardmærke

Hvis der som felt i inputstrengen kun er givet et enkelt tegn af klassen standardmærke (minus-tegn), udføres ingen egentlig læsning. I stedet markeres standardværdi for den først specificerede læsevariabel, hvis denne er erklæret med standardværdi i ld-beskrivelsen. Hvis ikke, udløses en fejlaktion.

implicit
standardmærke

Samme mulighed for automatisk standardværdi-markering findes, hvis man med read-operationen forsøger at læse felter ud over sluttegnet i inputstrengen. Dette vil blive opfattet helt analogt til, at der som felt var anført et standardmærke. Faciliteten bevirker, at det i inddata er muligt uden særlig syntaktisk markering at afkorte linier, hvor alle resterende felter skal have standardværdi.

stdassign

Hvordan standardmærke behandles, afhænger af værdien af variabelen 'stdassign':

- stdassign <> 0: læsevariablen assignes med den standardværdi, den har fået tilknyttet i ld-beskrivelsen.
- stdassign = 0: læsevariablens værdi ændres ikke, idet systemet går ud fra at variabelen tidligere er blevet assignet med standardværdi, f.eks. ved en db-operation (jf. afsnit 3.2.13).

Variablen stdassign nulstilles ved kald af 'init_duetmaskine'. Den kan ændres af styreprogrammet efter dette kald, men den kan også ændres dynamisk af duetprogrammet ved hjælp af operationen 'select' (se afsnit 3.2.14).

resultatvar
readspec

Efter læseoperationen fortæller resultatvariablen 'result.readspec', hvilken læsespecifikation der er blevet udført:

result.readspec = 0: standardværdi eller foranstillet delimiter læst
> 0: nummeret på den effektuerede læsespecifikation.

(Hvis resultatvariablen ikke er defineret, forsvinder denne oplysning bare i den blå luft.)

3.2.12 Print

primula

Print-operationen benyttes til trykning af resultater. Trykningen sker med brug af primula-systemet, som benytter sig af en liniebuffer på 132 positioner, der kan assignes i vilkårlig rækkefølge.

Denne liniebuffer tømmes ved en særlig printkommando ud i et resultatareal via en zone i zonearrayet 'prinz'. Styreprogrammet definerer, hvor mange zoner dette zonearray indeholder, og åbner dem til hver sit resultatareal (jf. afsnit 5.2.1 og 5.5.1).

Med operationen 'select' (beskrevet i afsnit 3.2.14) vælges den aktuelle zone indenfor arrayet, og dette valg gælder indtil næste kald af 'select'. Kald af proceduren 'init_duetmaskine' vælger dog ubetinget prinz(1) som trykkekanal.

Formatet af printoperationen er en liste af variabel længde afsluttet med duetslut (tegnet 's') som vist i eksemplet figur 3.39.

```

> print <p 1 t23>      : kundenavn; sideskift
      < 25 n8>         : kundenr
      d335             :                ; beregn saldo
      <21 25 n6.2>: saldo
      < 1 25 c9>      : .=.
      < 1             >                ;
      s

```

figur 3.39: print-liste

Hver linie er enten et duetnavn eller en printlinie bestående af et layout og en værdispecifikation.

duetnavn

Hvis linien er et duetnavn, skal det være navnet på en duetinstruktion i samme blok, og denne duetaktion vil blive udført på samme måde som i en execute-liste. Der kan ikke skrives blokreferencer, og der kan ikke skrives flere duetnavne i samme linie.

printlinie

En printlinie består af en layoutspecifikation, som beskriver hvordan der skal trykkes, og en værdispecifikation, som fortæller hvad der skal trykkes.

3.2.12.1 Layoutspecifikation

Layoutet skrives omsluttet af klammer <>. Mellem disse anføres enten

- en horisontalspecifikation alene

eller

- en vertikalspecifikation

og/eller

- en positionsangivelse,
- en layouttype og
- en layoutparameter

horisontal-
specifikation:
basisposition

En horisontalspecifikation angives med tegnet 'h' efterfulgt af en positionsangivelse, som kan være en numerisk konstant eller en simpel heltalsvariabel indesluttet i parentes. Hermed specificeres en basisposition, der under kørslen ubetinget adderes til alle positionsværdier, indtil en ny horisontalspecifikation mødes.

Basispositionen nulstilles af proceduren 'init_duetmaskine'.

vertikal-
specifikation

En vertikalspecifikation angiver at liniebufferen skal tømmes ud på resultatarealet, hvorefter der trykkes enten sideskift, vertikaltabulering eller linieskift (ISO-værdi hhv. 12, 11 eller 10). Dette angives i layoutet med henholdsvis tegnet p, w eller l - det sidste eventuelt indledt med en parameter, der specificerer antal linieskift. Parameteren kan være et heltal eller en simpel heltalsvariabel. I sidste fald bestemmes det aktuelle antal linieskift først under kørslen. Antallet af linieskift må ikke overstige 70, svarende til en hel side A4-høj.

position

Positionsangivelsen bestemmer nummeret i liniebufferen på første tegn af det aktuelle felt. Angivelsen kan - ligesom horisontalspecifikationen - enten være et heltal eller en simpel heltalsvariabel indekluttet i parentes. Positionen skal ligge i intervallet i 1-127.

layouttype og
layoutparameter

Layouttypen og layoutparameteren bestemmer formatet og størrelsen af det felt i liniebufferen, der skal trykkes, og stiller samtidig krav til typen af den værdi, der angives med værdispecifikationen.

t: text

Layouttypen t angiver teksttrykning, og værdien skal være af typen tekst. Layoutparameteren, der kan være et heltal eller en simpel heltalsvariabel i parentes, angiver hvor mange positioner der er afsat i liniebufferen. Hvis teksten fylder mere kortes den af.

c: char

Layouttypen c angiver tegntrykning (charprint). Værdien skal være numerisk, og den vil blive fortolket på samme måde som en tegnkonstant (jf. afsnit 3.1.4). Layoutparameteren skal også her være et heltal eller en simpel heltalsvariabel i parentes, der angiver hvor mange gange tegnkonstanten skal trykkes. (Tegntrykning er således velegnet f.eks. til at trykke understregninger o.l.).

n: num

Layouttypen n betegner numerisk trykning, og kræver at værdien er numerisk. Layoutparameteren hertil er opbygget af følgende elementer i den viste rækkefølge:

- en fastfortegnsmarkering
- en nulrepræsentationsangivelse
- en nulværdiangivelse
- antal principaler
- antal decimaler

fastfortegn

-

Fastfortegnsmarkeringen, der anføres med et minustegn, betyder, at hvis værdien er negativ, skal minustegn ubetinget trykkes i første feltposition. Udelades markeringen, trykkes negative værdier med flydende fortegn, dvs. umiddelbart før første betydende principale ciffer.

nulrepr.

z *

Nulrepræsentationsangivelsen fortæller, hvorledes ikke betydende principale cifre skal trykkes. Anføres tegnet z, trykkes foranstillede nuller, og anføres *, trykkes dette tegn i stedet for foranstillede nuller. Udelades angivelsen, trykkes foranstillede nuller som blanke.

nulværdi

b

Nulværdiangivelsen fortæller med tegnet b, at værdien nul ønskes trykt som helt blankt felt uanset eventuel nulrepræsentationsangivelse.

principaler

tal

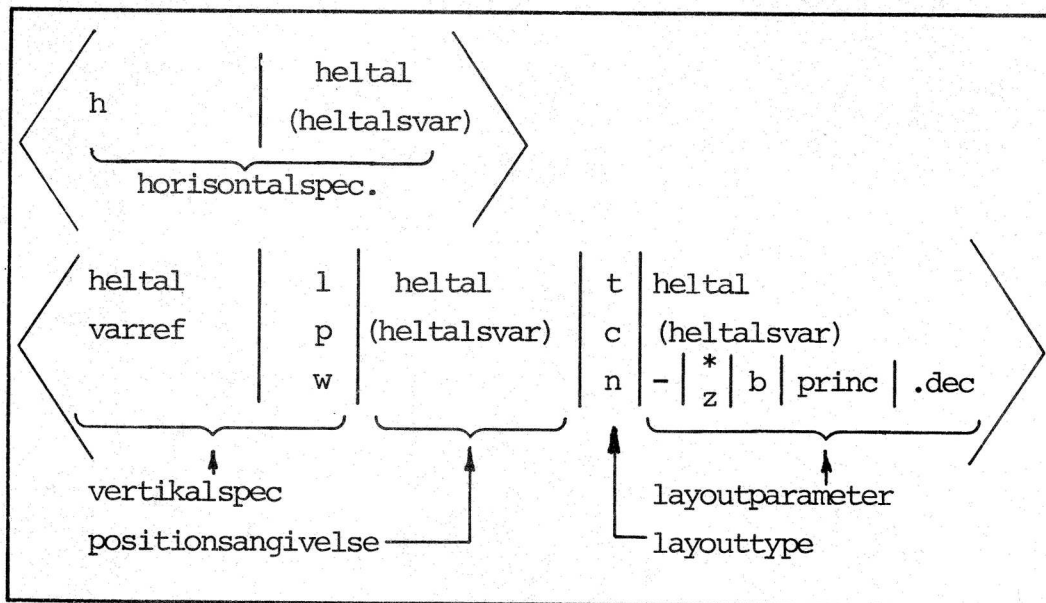
Antal principaler er et heltal, der angiver, hvor mange tegnpositioner der maksimalt skal optages af principaldelen af tallet. Ved angivelse af flydende fortegn vil minustegnet optage en af disse positioner. Principalspecifikationen er det eneste parameterelement, der ikke kan udelades.

decimaler

.tal

Antal decimaler specificeres med et punktum fulgt af et heltal, der angiver antallet af decimalcifre, der ønskes trykt. Disse vil altid blive trykt fuldt ud.

Figur 3.40 giver en uformel oversigt over layoutmulighederne:



figur 3.40: layout

Se også eksemplerne i figur 3.39 og 3.42.

3.2.12.2 Værdispecifikation

Hvis layoutet indeholder en layouttype (t, c el. n), skal der være angivet en værdispecifikation, hvis type skal svare til layouttypen.

tekstværdi

Til et tekstlayout (t) skal angives en tekstværdi, dvs.

- 1) en tekstkonstant (teksttegn indesluttet i apostroffer),
- 2) en tekstvariabel
- 3) et variabelnavn, f.eks. name(v32)

num.værdi

Til et numerisk layout (n) eller et tegnlayout (c) skal angives en numerisk værdi, der kan være

- 4) en numerisk konstant,
- 5) en tegnkonstant (teksttegn indesluttet mellem punktummer),
- 6) en numerisk variabel,
- 7) et udtryk af numeriske variable og/eller konstanter,
- 8) en ciffergruppe udtrykt af en numerisk variabel med operatoren 'digits'.

Punkterne 1-2 og 4-6 hører under begrebet 'værdielement' (jf. afsnit 3.2.4). Udtrykket i punkt 7 svarer til hvad der kan skrives efter assignsymbolet i en computeliste (jf. afsnit 3.2.3). Digits-operatoren og name-operatoren er beskrevet under modify (afsnit 3.2.2).

automatisk
normalisering

Numerisk trykning vil altid ske med automatisk hensyntagen til eventuelle decimaler i den værdi, der skal trykkes, således at layoutets decimalpunktum altid kommer lige efter værdiens enerposition, som vist i figur 3.41.

```
> print <l 1 n8>      : var_2_dec
      < 11 n5.2>     : var_2_dec
      < 21 n3.4>     : var_2_dec
      s
```

Når var_2_dec indeholder værdien 123.45, vil denne print-operation trykke linien:

```
123      123.45      123.4500
```

figur 3.41: behandling af decimaler i print

Mellem layoutet og værdispecifikationen kan skrives et kolon (af hensyn til den der er vant til at skrive geniusprogrammer), men kolon'et kræves ikke (af hensyn til dem der var vant til at bruge det tidligere duetsprog).

Figur 3.42 viser et større eksempel på en print-operation. Duetoversætteren sørger for at værdispecifikationerne og kommentarerne trykkes under hinanden, men ændrer ikke opstillingen af layouterne. Det kan anbefales, at man selv sørger for en vis redigering af disse.

```

d10: print  d11
          <21 4 t30 > kundenavn          ; evt. overskrift
          < (v2)n5.2 > saldo - betalt     ; tekstvariabel
          < 1 4 t30 > name(v32)           ; ny saldo
          < (v2)nb5.2> betalt             ; variabelnavn
          < h(v1) >                       ; basispos for dato
          d20                             ; tryk dato
          < h 0 >                          ; basispos nul
          < 1 13 t10 > 'kunderabat'       ; fast tekst
          < 23 c1 > ...                   ; kolon
          < 25 n3.1 > (v13(2) - v3)*100/v3 ; rabat
          < 30 c1 > 37                    ; pct-tegn
          < 1 >                            ; tøm sidste linie
          s

d11: modify linietaeller :+ 4
          s
          > if linietaeller > max_linie then d12

d12: modify sidenr :+ 1
          linietaeller := 0
          s
          > print <p 5 t30> 'kunde-/betalingsliste'
          < 50 t40> 'side'
          < 55 n3 > sidenr
          s

d20: print < 1 n2 > digits (2,1) of dato ; dag
          < 3 n-z2> digits (4,3) of dato ; md
          < 6 n-z2> digits (6,5) of dato ; år
          < 3 c1 > 46 ; pkt
          < 6 c1 > 46 ; pkt
          s
    
```

figur 3.42: print-eksempel

3.2.12.3 Trykning med standardlayout

Til erstatning for operationen 'list', der i det gamle duetsprog kunne bruges til standardudskrift af variable, er der indført to former for trykning med standardlayout. De er begge vist i figur 3.43.

```
print <l 1 t40> 'standardtrykning m. variabelnavne'  
    <a 5,20> numvar  
    <a 5,20> txtvar  
    <l 1 t40> 'standardtrykning m. faste tekster'  
    <s 5,25> 'kundenavn', ku_navn  
    <s 5,30> 'tilgodehavende', saldo  
s
```

figur 3.43: standardlayout

Standardlayoutet indeholder en layouttype, 'a' eller 's', og to positionsangivelser, heltal med komma imellem. Som værdispecifikation kan kun angives en simpel variabel af vilkårlig type.

Den første position angiver startpositionen for trykning af en variabelbetegnelse. Ved layouttype 'a' udskrives variabelens navn, mens der ved layouttype 's' trykkes den anførte tekstkonstant.

Den specificerede variabel trykkes med et standardlayout, der tager hensyn til variabelens type og eventuelle decimalantal.

3.2.13 DB-operationer

Database-operationerne i duet (db-operationer) bygger på sodasystemets DBMS-procedurer (se ref. 2), idet hver db-operation i duet kalder sodaproceduren af samme navn. Formatet af db-operationerne fremgår af eksemplerne i figur 3.44.

```
> get      s1
> next     s (sætnr)
> lookup   s7
> put      s (v2)
> delete   s2
> newset   s7

> create   s3, i7
> create   s (v15), i (v16)
```

figur 3.44: db-operationer

Parametrene angiver det sætnummer, der skal behandles, og for create desuden den ønskede individtype. Såvel sætnr som individtype kan angives med heltal (f.eks. s3 og i7), eller med reference til en simpel heltalsvariabel (f.eks. s (sætnr) og i (v16)). I det sidste tilfælde bestemmes sætnr/itype først på kørselstidspunktet.

Den følgende gennemgang af de enkelte db-operationer er meget overfladisk. For nærmere uddybelse henvises til sodamanageren (ref. 2).

- get Operationen get henter ved direkte acces et individ fra databasen, og gør dette individ til 'current record' i sættet. Værdier overføres fra individfelter til variable i overensstemmelse med de flytteassociationer i ld-beskrivelsen, der indeholder '<'.
- next Med next-operationen hentes det næste individ i sættet efter det der sidst blev læst med next, subsidiært efter newset til første individ. Det læste individ bliver 'current record' i sættet. Værdien overføres fra individfelter til variable som ved get.
- lookup Lookup-operationen undersøger om individet med de givne nøgler tilhører sættet. Selvom individet findes, bliver det ikke 'current record', og der bliver ikke udført feltflytninger. Lookup kan kun anvendes på sæt tilhørende cf-masterfiler (sættype M).
- create Operationen create opretter et nyt individ og gør dette til 'current record' i sættet. Variable knyttet til den aktuelle individtype initialiseres med standardværdi i overensstemmelse med de flytteassociationer i ld-beskrivelsen, der indeholder '*'.
- put Put-operationen afleverer sættets 'current record' i databasen, afhængigt af om denne er fremkommet ved en læsning (get/next) eller en oprettelse (create). Efter put findes ingen 'current record'.

Ved put efter get/next tilbageskrives et eksisterende individ i databasen, hvorunder værdier flyttes fra variable til individfelter i overensstemmelse med de flytteassociationer, der indeholder '>' eller '=>'.

Ved put efter create indsættes et nyt individ i databasen, og der flyttes fra variabel til individfelter i overensstemmelse med de flytteassociationer, der indeholder '->' eller '=>'. Placeringen af det nye individ i filen afhænger af sættypen som vist i skemaet, figur 3.45.

sættype	filtype	tidligere aktivitet	individ indsættes
M	cf-master	~	efter nøgler
B	bs-fil	~	altid sidst i filen (ved eof)
L	cf-list, singulært	~	indsættelse tillades ikke
	cf-list indiceret	newset next kædeslut	forrest i kæden foran tidl. 'current record' sidst i kæden

figur 3.45: placering af nyindsatte individer

delete

Med delete-operationen fjernes sættets 'current record' fra databasen, og der findes derefter ingen 'current record' i sættet. Der udføres ingen feltflytninger. Hvis sættet tilhører en bs-fil, slettes samtidig alle eventuelle efterfølgende individer i filen.

newset

Endelig er operationen newset beregnet til at initialisere et sekventielt gennemløb af filen. Der findes ingen 'current record' i sættet efter newset, og der udføres ingen feltflytninger. Udgangspositionen for den senere sekventielle læsning sættes afhængig af sættypen som vist i figur 3.46.

sættype	filtype	position sættes
M	cf-master	efter nøgler
B	bs-fil	identspec findes: efter nøgler ingen identspec: forfra
L	cf-list, singul.	newset tillades ikke
	cf-list, indic.	forfra i kæden

figur 3.46: position efter newset.

result.soda

Resultatet af db-operationen er tilgængelig i resultatvariablen defineret som result.soda med følgende værdier:

```

result.soda = 0: db-operation ok
             = 1: individ findes ikke
             >= 2: fejl ved db-operation
    
```

Resultatet 1 accepteres som resultat af operationerne lookup (det ønskede individ findes ikke) og next (kædeslut/filslut), mens dette resultat efter get vil bevirke kald af fejlproceduren 'duetdatafejl'.

Resultatværdier fra 2 og opefter bevirker altid kald af en fejlprocedure, enten 'duetdatafejl', 'duetprogramfejl' eller 'duetsystemfejl' afhængigt af fejllens art (jf. afsnit 5.3).

result.recno

Efter indsættelse af et nyt individ i en cf-listefil eller en bs-fil indeholder resultatvariablen result.recno positionen for det nye individ. Denne position kan benyttes som nøgle til senere direkte acces i filen/sættet.

3.2.14 Select

Operationen select er en skraldespandsaktion, hvormed man dynamisk under udførelsen af duet-programmet kan ændre forskellige kørselsparametre. Man kan vælge:

- trykkekanal for resultat-, fejl- eller testudskrifter,
- reaktion på fejlsituationer,
- reaktion på læsning af standardmærke samt
- sætte/slette testvariable.

Formatet af selectoperationen er en liste af variabel længde afsluttet med duetslut (tegnet 's'). Hver linie i listen udfører en af ovennævnte aktioner og de forskellige aktioner kan blandes vilkårligt mellem hinanden som i figur 3.47.

```
> select print on 3
      test a :+ 0,2,23
      print text on 4
      return on dataerror
      exit p9 on systemerror 11,12
      no stdassign on read
      s
```

figur 3.47: eksempel på select-liste

I det følgende gennemgås de forskellige muligheder i hver sit underafsnit.

3.2.14.1 Select print

Hermed kan vælges trykkekanal (dvs. index i zonearrayet prinz) for resultatudskrifter, fejludskrifter og testudskrifter. Formatet af select print linier fremgår af figur 3.48.

```
> select print on v32
      print data error on 6
      print program error on 9
      print system error on v16
      print test on 0
      s
```

figur 3.48: select print

kanaln

Kanalnummeret kan angives med et heltal i intervallet 0-9, eller med en simpel heltalsvariabel. Det kørende system kontrollerer at kanalnummeret ikke overstiger antallet af zoner i prinz.

out

Kanalnummer 0 betyder trykning på zonen out. Denne mulighed findes dog kun for fejludskrifter og testudskrifter. Resultatudskrifter kan kun trykkes på en af prinz-zonerne.

Standardværdier for kanalnumre benyttes når select print ikke har været kaldt. Standardkanal for resultatudskrifter er 1 (prin(1)) og for fejludskrifter og testudskrifter 0 (out). Disse standardkanaler vælges hver gang 'init_duetmaskine' kaldes, testkanal dog kun en gang ved kaldet af 'init_duet2'.

3.2.14.2 Select exit/return

Med select exit og select return vælges reaktionen efter fejludskrifter. Standardreaktionen på enhver fejl er udhop fra duetmaskinen (jf. afsnit 5.1). Dette kan ændres selektivt for hver enkelt fejltype til, at duetprogrammet enten hopper til et programpunkt (jf. afsnit 3.2.1.3) eller returnerer for at fortsætte programmet.

Mulighederne fremgår af figur 3.49.

```

> select exit p3   on data error 2,7,5 ; hop til p3
      return on data error 6,8       ; fortsæt uændret
      exit p0     on data error 11    ; udhop

      exit p0     on system error     ; udhop for alle
                                          ; systemfejl
      return on program error         ; fortsæt ved alle
                                          ; duetprogramfejl

s

```

figur 3.49: select fejlreaktioner

p0 kan ikke defineres som programpunkt. Det benyttes her til at angive udhop fra duetmaskinen.

Hvis man efter fejltypen (datafejl, programfejl, systemfejl) angiver et eller flere heltal, gælder reaktionen kun de således specificerede fejlnumre. Hvis fejlnummer udelades, gælder reaktionen for alle fejl af den pågældende type.

3.2.14.3 Select stdassign

Hermed vælges om der ved læsning af standardmærke skal assignes standardværdi til læsevariablen eller ikke. Der er kun to muligheder som vist i figur 3.50.

```

> select stdassign on read      ; standardværdi
      S                          ; assignes

> select no stdassign on read ; standardværdi
      S                          assignes ikke

```

figur 3.50: select stdassign

3.2.14.4 Select test

Man kan sætte og slette enkelte bits i testvariablene testa, testb, -----testg, testh ved hjælp af select test som vist i figur 3.51.

```

> select test a := on
      test b := off
      test c := 0, 1, 3, 7
      test d :=+ 3, 4
      test e :=- 21, 22, 23
      test f := duet_testvar
      test g :=+ v17
      test h :=- v16
      S

```

figur 3.51: select test

Bemærk: der kræves mellemrum mellem 'test' og testvariablens navn ('a', 'b', osv.).

assign-
operator

Der kan benyttes assignoperatorerne
:= for almindeligt assign
:+ for tilføjelse af yderligere bits og
:- for sletning af enkelte bits

testbits

Efter assignoperatoren følger en udpegning af de testbits, der skal sættes, tilføjes eller slettes i testvariablen. Testbits nummereres fra højre med numrene 0-23.

Der kan anføres 'on' eller 'off', der hhv. etstiller eller nulstiller alle testbits. Denne mulighed kan kun benyttes sammen med assignoperatoren :=.

Testbits kan også skrives som et eller flere heltal med komma imellem, hvormed anføres bitnumrene på de testbits, der skal sættes/tilføjes/slettes.

Endelig kan man anføre en simpel heltalsvariabel. Her er det bitmønster indeholdt i denne variabel, der sættes/tilføjes/slettes i testvariablen.

Testvariablene testa, testb, testc og testd er reserveret duetsystemet og sodaprocedurerne, mens variablene teste, testf, testg og testh er til rådighed for styreprogrammet.

3.2.16 Algol

algol
special-
aktion

Operationen algol er brugerens skraldespand. Med den kan man aktivere algolkodede specialaktioner i styreprogrammet, hvor man kan udføre programdele, der ikke egner sig for kodning i duet.

```
> algol 3 ('tekstkonstant', 12345,  
          name (v27),  
          adr (v27), adr (var (v16)),  
          kundenr)  
> algol 15
```

figur 3.53: algol

Til algol-operationen skal anføres et heltal, der angiver nummeret på den ønskede specialaktion.

Der kan desuden anføres en parentes indeholdende et vilkårligt antal parametre med komma og evt. lineskift imellem. Disse parametre skal fortolkes af specialaktionen, som beskrevet i afsnit 5.4.

De tilladte parametertyper er

- 1) tekstkonstant
- 2) tekstvariabel
- 3) variabelnavn specificeret v.hj. af name-operatoren
- 4) numerisk konstant
- 5) tegnkonstant
- 6) numerisk variabel
- 7) variabeladresse specificeret v.hj. af adr-operatoren

Name-operatoren er beskrevet under modify (afsnit 3.2.2.2).

adr ()

Med adr-operatoren angives at man ønsker udleveret adressen på den refererede variabel. Hvis variabelen er et array, eller hvis den skal benyttes som returparameter (assignes af specialaktionen), er det nødvendigt at bruge adr-operatoren. Den kan bruges som vist i figur 3.54.

```
> algol 2 (adr (simpel_var),
             adr (aray_var),
             adr (var (simpel_var)),
             adr (var (array_var (index))))
```

figur 3.54: algolaktion med adr-specificerede parametre

Efter 'adr(' kan direkte anføres et variabelnavn (eller -nummer). Til en arrayvariabel må ikke gives index. Den adresse specialaktionen vil få leveret bliver da fieldadressen for en specificeret simpel variabel, subsidiært fieldadressen for første element af en arrayvariabel.

adr (var())

Hvis der i stedet skrives 'adr (var(', skal den specificerede variabel indeholde variabelnummeret på den ønskede variabel. Det er da fieldadressen for denne indirekte anførte variabel, der regnes.

4. Duetoversætteren

duetabler

Duetoversætteren 'duetabler' er et algolprogram, der læser og kontrollerer duetprogrammet og oversætter det til programtabeller, der senere kan fortolkes af 'duetmaskinen'. Samtidig med oversættelsen kan der udskrives en redigeret listning af duetprogrammet.

duetregister

Hele duetprogrammet kan oversættes samlet, eller man kan nøjes med at oversætte enkelte duetblokke separat. Endvidere kan man for hver enkelt blok angive, om den oversat skal afleveres i et duetregister, eller om den blot skal kontrolleres. Kun korrekte duetblokke vil dog blive afleveret i duetregistret.

kontrol af
duethoved

Når et duetprogram oversættes samlet til et nyt duetregister, gemmes oplysningerne fra duethovedet (duetprogramnr. og -navn samt ldnr). Ved en senere deloversættelse og fletning med et gammelt duetregister kontrolleres, at duethoved i teksten og i duetregistret er ens.

4.1 Programtekst og listning

tekstfil

Duetprogramteksten kan læses fra enten en almindelig tekstfil på disc eller et sysdokregister.

sysdokregister

Hvis programteksten lagres i et sysdokregister, vil det være praktisk at anbringe de enkelte duetblokke i hver sit sideordnede underafsnit. Ved deloversættelse af en enkelt blok kan man da specificere læsning af det aktuelle underafsnit, hvorved man sparer den tid, oversætteren skal bruge til at skippe hen over de øvrige duetblokke. I dette tilfælde kan duethovedet dog ikke kontrolleres.

redigering

Ved indtastning af programteksten er det ikke nødvendigt at redigere linierne således, som de ser ud i eksemplerne i afsnit 3. Oversætteren vil selv sørge for redigering, når programmet listes.

sideskift

Listningen vil også blive forsynet med sideskift svarende til den papirtype, der listes på. Derudover kan man selv fremtvinge sideskift foran en duetoperation ved at skrive nysidesymbolet '-+-' og/eller en stopcode (ISO-værdi = 12). Foran blokstart tillades kun stopcode. Afsnit-skift i et sysdokregister vil derimod ikke fremkalde sideskift.

--

eksternt
og internt
linier

I listningen bliver alle linier trykt med to linienumre. Først trykkes det eksterne linienr, dvs. bosslinienr eller sysdoklinienr. Derefter følger et internt linienr, der nummererer linierne inden for en blok fra 1 og opefter. Ved fejludskrifter er det altid det interne linienr, der refereres.

Ved redigeringen af de enkelte linier opdeles linien i felter, som vist i figur 4.1.

d132:	assign	kode := testvar of .a. : 3 11 : 2 else d15	; kommentar
>	execute	d17	
	p2:	d32, d18	; programpunkt
	t3	d998	; testudskrift
		s	
>	print	<1 17 n 5.2>: beløb < 26 t2> : 'kr'	
		s	; tom linie
instr. felt	operator- felt	operandfelt	kommentarfelt

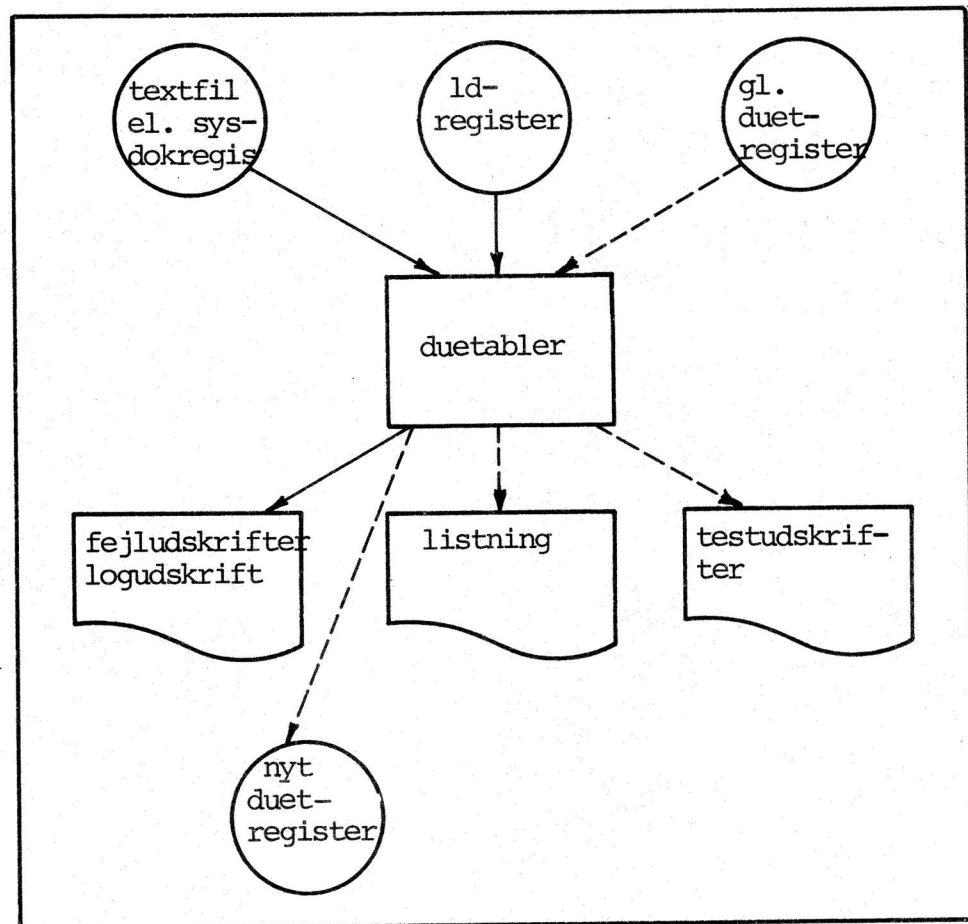
figur 4.1: liniefelter for redigering

Redigeringen sker således, at liniefelter af samme type altid trykkes under hinanden. Inden for operand- og kommentarfeltet må brugeren derimod foretage en eventuel redigering (f.eks. indsætte mellemslag for at få : under hinanden i en assignliste).

I en printlinie vil oversætteren dog redigere delvis, idet trykkeværdierne her redigeres under hinanden, men en evt. redigering inden for layoutet må brugeren styre.

4.2 Aktivering af oversætteren

I en kørsel med duetabler indgår 2 eller 3 inddatafiler og maksimalt 4 uddatafiler, som vist i figur 4.2.



figur 4.2: oversigt over inddata til og uddata fra duetabler

inndata

Inndatafilerne er

- en tekstfil eller et sysdokregister indeholdende duetprogramteksten,
- et ldregister (i form af et beskrivelsesregister) indeholdende en oversat ld-beskrivelse,
- eventuelt et gammelt duetregister fra en tidligere oversættelse af duetprogrammet.

uddata

Uddatafilerne, hvoraf kun den første altid dannes, er

- current output, hvori skrives eventuelle fejludskrifter samt en logudskrift,
- et diskareal, hvori det nye duetregister dannes,
- et diskareal til eventuel listning og
- et diskareal til eventuelle testudskrifter.

Brugen af disse filer trimmes med fp-parametre ved kaldet af oversætteren, som beskrevet i det følgende. Fp-parametrene anføres gruppevis, hvor hver gruppe indledes med et nøgleord efterfulgt af en eller flere parametre med punktum imellem. Figur 4.3 indeholder en oversigt over samtlige parameter-nøgleord.

parametre vedr. kildetekst:

- sysdok
- afsnit
- version
- duetttext
- bruger
- user
- init
- include

parametre vedr. listning:

- list
- listud
- papir

parametre vedr. ldregister:

- beskriv
- ldreg
- ldafsnit

parametre vedr. duetregister:

- oldduet
- newduet

parametre vedr. oversættelsesspecifikation:

- insert
- change
- delete
- translate
- size

parametre vedr. testudskrifter

- test
- testud

figur 4.3: oversigt over parameternøgleord

De fleste parametergrupper kan udelades, i så fald benyttes standardværdier, som beskrevet for hver enkelt parametergruppe. Hvis en parameterstype angives flere gange, vil den sidste være gældende, undtagen for parameter typerne insert, change, delete, translate og test, hvor eventuelle gentagelser supplerer tidligere forekomster.

Eksempler på kald af oversætteren findes i afsnit 4.3.

I det følgende beskrives de enkelte parametergrupper.

Parametre vedrørende kildetext:

Kildeteksten til duetprogrammet kan læses enten fra et sysdokregister eller fra en almindelig tekstfil. Ved læsning fra sysdokregister skal angives følgende parametergrupper:

sysdok-
register

```
sysdok.<sysdokreg_navn>  
afsnit.<afsnitsnummer>  
vers.<versionsnummer>
```

sysdok

Sysdokparameteren angiver, fra hvilket sysdokregister teksten skal læses.

Standard: sysdok. sysdokreg

afsnitnr

Afsnitparameteren angiver, fra hvilket afsnit i sysdokregistret teksten skal læses. Afsnitsnummeret angives på sædvanlig sysdokmåde med hovedafsnit og evt. underafsnitsnumre adskilt med punktum. Parameteren kan ikke udelades, hvis der læses fra sysdokregister.

version

Vers angiver, hvilken version af sysdokafsnittet der ønskes oversat.

Standard: sidste version

tekstfil

Ved læsning fra en almindelig tekstfil angives i stedet med

duettext

```
duettext.<duettext_navn>
```

navnet på det tekstareal, hvorfra duetprogrammet skal læses.

Standard: Hvis denne parameter udelades, læses der fra sysdokregister. Hvis parameteren angives, må der ikke angives nogen sysdokparametre.

De følgende parametre kan angives, uanset om der læses fra sysdok eller tekstfil:

brugernr

```
{ bruger }  
{ user } .<brugernr>
```

angiver brugernummeret for oversættelsen. Der kan kun oversættes duetblokke tilhørende én bruger i en enkelt kørsel, og den benyttede ldbeskrivelse skal tilhøre samme bruger eller bruger 0.

Standard: bruger.0

initialer

```
init.<initialer>
```

angiver initialerne på den, der har aktiveret duetoversættelsen, til brug for duetloggen. Standardværdi (= tom tekststreng) bør ikke anvendes.

include

```
include.<undertryk_grænse>
```

undertryk_grænse er et heltal i intervallet 0-9. Denne parameter angiver undertrykkelse af de testlinier i en execute-liste, der er forsynet med et testnr større end undertrykkelsesgrænsen. Standard: include.0, dvs. ingen testlinier medtages.

Parametre vedrørende listning:

```
list. {
      ja
      yes
      nej
      no
}
listud.<listud_navn>
papier. {
      <bosspapirformat>
      <linier_pr_side>.<tegn_pr_linie>
}
```

list

Listparameteren angiver, om der dannes en listning eller ej.

Fejludskrifter kommer i begge tilfælde også på current out.

Standard: list.nej

listud

Listudparameteren angiver navnet på det diskareal, hvor en redigeret listning skal skrives. Hvis området ikke findes i forvejen, vil det blive oprettet som en temporær fil, der auto-

matisk konverteres på den lokale printer.
Hvis området findes i forvejen, må brugeren selv sørge for udskrivning. Forekomst af denne parameter vil automatisk sætte parameteren list.yes.

Standard: listud.listud

papir

Papirparameteren angiver, hvordan det redigerede duetprogram skal formatteres. Der kan anføres et enkelt heltal, som angiver boss-papirformat, eller 2 heltal der bestemmer antal linier pr. side og antal tegn pr. linie.

I den redigerede listning trykkes sideskift:

- 1) når <linier_pr_side> overskrides
- 2) før hver blokstart
- 3) ved sideskifttegn (-+- el. stopcode) i input

<bosspapirformat>:

0: monitorpapir (<linier_pr_side> = 62)

2: A4-tværs (<linier_pr_side> = 40)

(A4-høj = 1 tillades ikke)

<linier_pr_side>.<tegn_pr_linie>:

Disse tal kan angives eksplicit i stedet for boss-papirformat. <tegn_pr_line> benyttes dog fore-

løbig ikke. Hvis <linier_pr_side> er mindre end 15, sætte sideskiftregel 1) ud af kraft, hvorved opnås en mere kompakt listning.

Standard: ved læsning fra sysdokregistret: som angivet i sysdokregistrets ejerinformation ved læsning fra textfil: papir.0

Parametre vedrørende ld-register:

ldregister

$\left. \begin{array}{l} \text{beskriv} \\ \text{ldreg} \end{array} \right\} .\langle \text{ld_registernavn} \rangle$ <p>ldafsnit.<ld_afsnitnummer></p>
--

angiver navn og afsnit for det beskrivelsesregister, hvor den oversatte ldbeskrivelse er lagret. Ordene beskriv og ldreg kan anvendes efter behag. Parameteren ldafsnit kan ikke udelades.

Standard for ld_registernavn: beskriv.beskrivreg

Parametre vedrørende duetregister:

gammelt og nyt duetregister

$\left. \begin{array}{l} \text{oldduet}.\langle \text{gl_duetreg_navn} \rangle \\ \text{newduet}.\langle \text{ny_duetreg_navn} \rangle \end{array} \right\} \left. \begin{array}{l} \langle \text{duetversion} \rangle \\ \end{array} \right\} \begin{array}{l} 1 \\ 0 \end{array}$
--

angiver navnene på hhv. gammelt og nyt duetregister. Hvis duetversion er anført, kontrolleres, at det stemmer med versionsnummeret i gammelt duetregister.

Hvis newduet-parameteren udelades, må oldduet heller ikke angives, og der dannes intet duet-register. Oversættelseskørslen bliver altså kun en kontrol af duetblokke.

Hvis newduet angives, men ikke oldduet, dannes et nyt duetregister med versionsnr 1, bestående af fejlfrít oversatte duetblokke.

Hvis både newduet og oldduet angives, flettes det gamle duetregisters blokke med fejlfrít oversatte blokke.

Parametre vedrørende oversættelsesspecifikation:

insert	{.<bloknr>} [*] ₁	}	1
change	{.<bloknr>} [*] ₁		
delete	{.<bloknr>} [*] ₁		
translate	{.<bloknr>} [*] ₁		

eller

{ insert change translate }	.all	1
		1

Disse parametre, som skal angives efter duet-registerparametrene, specificerer hvilke duetblokke der skal læses af oversætteren. Hvis



insert.all, change.all eller translate.all er angivet, oversættes alle de duetblokke, der findes i kildeteksten.

insert De angivne blokke indsættes i duetregistret. Tillades kun, hvis newduet er angivet. Hvis oldduet også er angivet, må de specificerede blokke ikke findes i gammelt duetregister.

change De angivne blokke erstattes af ny oversatte duetblokke. Tillades kun, hvis både oldduet og newduet er angivet.

delete De angivne blokke fjernes fra duetregistret. Tillades kun, hvis både oldduet og newduet er angivet.

translate De angivne blokke kontrolleres af oversætteren, men indsættes ikke i duetregistret. Tillades altid.

bloknr

Et bloknr må kun forekomme én gang. Blokke, der findes i et evt. gammelt duetregister, men ikke er nævnt i listen, overføres uændret til det nye duetregister. Det samme gælder for blokke, der er specificeret under translate, samt for fejlbehæftede blokke.

size

size.<udvidelsespct>

Denne parameter angiver en procentvis forøgelse af alle oversætterens interne tabeller. Den må kun anvendes, hvis oversætteren ellers går i indexfejl, og værktøjsgruppen skal underrettes, når det bliver nødvendigt at anvende size-parameteren. Size.100 betyder fordobling af alle tabeller.

Standard: size.0

Parametre vedrørende testudskrifter

Disse parametre må kun anvendes i samråd med værktøjsvedligeholdelsen i forbindelse med fejl i duetoversætteren.

testa	.ja
testb	.yes
testc	.nej
testd	.no
testg	{.<tal>} ₁ ²⁴
testh	.not {.<tal>} ₁ ²⁴

angiver, hvilke testbits der skal sættes i hvilke testvariable. Testbits nummereres fra 0 til 23.

Standard: ingen testbits sat.

$\text{testud.<testud_navn> \left\{ \begin{array}{l} \text{.udvid} \end{array} \right\} \begin{array}{l} 1 \\ 0 \end{array}}$
--

angiver navnet på det areal, hvor testudskrifterne dannes. Hvis '.udvid' udelades, dannes testudskrifterne cyklisk i arealet, ellers udvides arealet efter behov. Testudarealet bliver hverken oprettet eller udskrevet automatisk.
Standard: testud.testud

Fejlreaktioner for parameterfejl

Hvis der konstateres syntaksfejl i en parametergruppe, udskrives en fejlmarkering i form af:

- <*> hvis en ny parametergruppe optræder som ulovlig afslutning af den forrige parametergruppe, eller
- <*> hvis der er fejl i et nøgleord eller en parameter i den aktuelle parametergruppe.

Efter fejl i fp-parametrene, det være sig syntaksfejl eller fejl i konsistenskontroller, afbrydes kørslen.

4.3 Ressourcekrav

Ressourceforbruget til oversættelsesjobbet varierer med duetprogrammets størrelse, og de følgende tal må kun tages som en rettesnor. Kun bossparametre ud over standard medtages.

size 80000	I 100000 bytes forekommer ingen ekstra segmenttransporter. Minimum er 60000.
area 8	Kan nedsættes, hvis ikke alle uddatafiler dannes.
time	I 80000 bytes tager det ca. 1 minut at oversætte en duetblok af maksimal størrelse, og ca. 20 sek. at skippe en tilsvarende blok. I 60000 bytes er de tilsvarende tider 7 og 3 minutter.
perm	en duetblok af maksimal størrelse fylder 5-8 segmenter i duetregistret. Hvis dette findes i forvejen, skal man medregne plads til evt. udvidelse,

Eksempler på jobfiler.

```

job eah 28xxxx size 80000 area 8 time 3 0,
  perm dk28xxxx 36 1
mode list.yes

duetreg = set 1 dk28xxxx           ; sæt nyt duetregister
scope user duetreg

duetabler,
  duettext. eahduet,               ; kildetekst
  init.eah,
  bruger.0,
  ldreg.eahbeskriv ldafsnit.30,    ; ld-register
  list.ja papir.2,                 ; listning
  newduet.duetreg,                 ; nyt duetregister
  insert.all                        ; hele duetprogrammet

finis

```

figur 4.4: dan nyt duetregister

Eksemplet i figur 4.4 viser etablering af et nyt duetregister i 'duetreg' ud fra en programtekst i 'eahduet'. Listning på A4-tværs dannes i listud, som konverteres automatisk på den lokale printer.

```

job pl 28xxxx size 80000 area 8 time 3 0
mode list.yes

glduetreg = move duetreg
listud = copy 0 ; sæt listudareal

duetabler,
  sysdok.eahsysreg afsnit.22, ; kildetekst
  init.pl,
  bruger.0,
  ldreg.eahbeskriv ldafsnit.30, ; ld-register
  list.ja, ; listning
  oldduet.glduetreg, ; gl. duetregister
  newduet.duetreg, ; nyt duetregister
  change.1.7,
  delete.2

convert listud std
finis

```

figur 4.5: ret duetregister

I figur 4.5 vises rettelse af et duetregister. Duetblokkene 1 og 7 genoversættes, mens blok 2 fjernes. De øvrige blokke forbliver uændrede. For altid at kunne have nyeste version af duetregistret i samme areal, flyttes 'duetreg' til 'glduetreg' før oversættelsen.

Duetprogramteksten læses fra afsnit 22 af sysdok-registret 'eahsysreg'.

Listningen skal trykkes på den centrale printer, derfor sættes listudarealet før oversættelsen.

4.4 Logudskrift

Som afslutning på en duetoversættelse, hvori der dannes et nyt duetregister skrives en duetlog, der viser resultatet af kørslen samt indholdet af gammelt og nyt duetregister. Et eksempel herpå findes i figur 4.6.

Størrelsen 'bloksize' indeholder to tal. Det første viser hvor mange ord den oversatte blok fylder. Dette tal kan ikke overstige 2047.

Det andet tal angiver længden af blokken inklusive lagrede numeriske konstanter og tekstkonstanter. Dette tal har betydning for dimensioneringen af duetarrayet i det kørende system (jf. afsnit 3.1.2 og 5.2.1).

De øvrige oplysninger er selvforklarende.

duetabler log duetbloktest EAH. 7 7.03.1977 - 16.03

binary files: area version date ident

new duetfile: nyduetreg 7 EAH 070377.1603 31 duetbloktest

old duetfile: eahduetreg 6 PL 030377.1619

localdata : eahbeskriv 0 EAH 070377.1320 30. variabelld

textfile : afsnit

sysdok : eahsysreg 24 22. duetbloktest

duetabler :

block no	command	old version	!	block no	user	duet- vers.	ld- vers.	date	blocksize duet/total
1	insert		!	1+	0	7 EAH	0	070377.1603	16/23
2	copy	6 PL	!	2	0	6 PL	0	030377.1619	22/39
3	change	6 PL	!	3+	0	7 EAH	0	070377.1603	9/16
4	transl	4 ISC	!	4	0	4 ISC	0	030377.1400	6/14
5	insert		!	5+	0	7 EAH	0	070377.1603	6/14
6	-insert		!						
7	-change		!						
8	-delete		!						
9	-transl		!						
10	insert		!	10+	0	7 EAH	0	070377.1603	144/219
11	insert		!	11+	0	7 EAH	0	070377.1603	80/98
12	-insert		!						
13	-change		!						
14	-transl		!						
15	-delete		!						

total used 8 segments of 36

- : error: no change/insert/delete

+ : block inserted or changed

figur 4.6: duetlog

4.5 Fejludskrifter

Alle fejludskrifter fra duetoversætteren trykkes både på current output og på en eventuel listning.

I listningen vil fejludskrifterne være af formen:

<pre>***** <fejlnr> : <fejltext> { <fejlparameter> } 3 0</pre>
--

og den fejlbehæftede linie trykkes derefter uredigeret, indledt med **.

I current output vil fejlen blive udskrevet således:

<pre><linienr><tegnnr><text> <fejlnr> : <fejltext> { <fejlparameter> } 3 0</pre>
--

hvorefter den fejlbehæftede linie udskrives. <text> er den syntaktiske enhed, der gav anledning til fejlen, og <tegnnr> angiver, hvor i linien denne enhed starter. <linienr> er det interne linienr.

Inden de enkelte fejl bliver gennemgået i fejlnummerorden, kommer her nogle generelle bemærkninger.

Efter en fejl vil oversætteren prøve at "falde i hak" så hurtigt som muligt. Ved mange fejl er det dog nødvendigt at skippe resten af linien. Dette er anført ved forklaringen af hver enkelt fejl.

I visse fejlsituationer er det imidlertid nødvendigt at skippe helt frem til starten af næste duetoperation. I så fald vil de efterfølgende linier blive kasseret med fejltipe 1: SYNTAX, selv om de i virkeligheden er gode nok.

Hvis der er konstateret fejl under oversættelsen af en duetblok, afleveres denne ikke i et eventuelt duetregister, og på duetloggen vil denne blok blive særligt markeret.

Fejl i en blok vil derimod ikke påvirke oversættelsen af de øvrige blokke. Man kan derfor spare maskintid ved efter fejlrettelse kun at genoversætte de ændrede blokke.

1: SYNTAX: <text>

<text> den sidst læste syntaktiske enhed.
Denne syntaktiske konstruktion er ikke tilladt, resten af linie skippes.

2: ILLEGAL VARIABLENO <v_no>

Et variabelnr må ikke overstige det største benyttede variabelnummer i ld-beskrivelsen. Af hensyn til fortsat behandling simuleres reference til en simpel wordvar uden decimaler.

3: UNDECLARED VARIABLE <text>

<text> den sidst læste syntaktiske enhed er en identifikator eller et variabelnr, som ikke svarer til en variabel erklæret i ld-beskrivelsen. Af hensyn til fortsat behandling simuleres reference til en simpel wordvar uden decimaler.

4: ILLEGAL ERROR NUMBER <error_no>

Systemfejl. kontakt værktøjsvedligeholdelsen.

5: ILLEGAL OPERAND VALUE <value>

Benyttes hvor: undertrykspecifikation > 9
duetprogrampunkt > 9
select printkanal > 9
select ... on error: ulovligt fejlnr
select test: testbit > 23
select test: testbit dobbeltspecificeret
tal <> 0 i 'modify array := 0'
tal <> 0 i 'modify recno := 0'
algolaktionsnr < 1

Resten af linien skippes.

6: NOT SIMPLE VARIABLE <varname> <varno>

Hverken arrayvariabel eller arrayelement tillades her.

7: ILLEGAL VARIABLE TYPE <var_name> <var_no> <var_type>

Denne variabeltype må ikke benyttes i den givne sammenhæng.

8: ILLEGAL NUMBER <no>

<no> den ulovlige værdi

Der er angivet en talkonstant som er for stor til at lagres i 24 bit.

Fejlen forekommer ved testværdier/assignværdier i en assign- eller actionliste.

9: UNDEFINED DUETNAME <d_name>

<d_name>

det refererede duetnavn er ikke erklæret ved nogen duetinstruktion indenfor blokken.

10: ILLEGAL NUMBER OF DECIMALS <var_name> <varno>

Variablen er erklæret med decimaler, og kan derfor ikke bruges

 som index
 til indlæsning af tegngruppe
 som testvariabel i assign, action,
 case og for

Resten af linien skippes.

11: ILLEGAL BLOCKNO <block_no>

Bloknr må ikke overstige 255.
Hvis fejlen optræder i et blokhoved, skippes resten af programteksten.
Ved fejl i en blokreference fortsættes behandlingen.

12: ILLEGAL USERNO <userno>

Brugernr må ikke overstige 127.

13: USERNO INCOMPATIBLE WITH LD DESCRIPTION <userno> <ld_user>

<userno> brugernr i blokhoved
<ld_user> brugernr i ld-beskrivelse
Brugernr i blokhoved og i ld-beskrivelse skal være det samme, med mindre ét af numrene er 0.

14: ILLEGAL ENTRYNO <entryno>

Et entrynr må ikke defineres udenfor intervallet 1-63, og ikke refereres udenfor intervallet 0-63.

15: ENTRYPOINT PREVIOUSLY DEFINED IN LINE <line_no>

<line_no> det linienr, hvor entrypunktet er defineret første gang.

16: ILLEGAL END NUMBER <block_no> <end_no>

<block_no> fra blokhovedet
<end_no> fra end-linien
Disse to numre skal være ens.

17: OBS! BLOCKSIZE APPROACHING LIMIT <size> <max_size>

<size> længden i ord af den oversatte duetblok
<max_size> den maksimale bloklængde i ord.
Pladsen til eventuelle blokudvidelser er begrænset.
Dette er kun en anmærkning, som ikke påvirker generering af duetregister.

18: DUETNAME PREVIOUSLY DEFINED IN LINE <line_no>

<line_no> det linienr hvor duetnavnet er defineret første gang.

19: ubenyttet

20: ILLEGAL NUMBER OF CHARACTERS <no>

<no> det læste antal

Med en læsespecifikation af typen c kan der højst læses 3 tegn.

Resten af linien skippes.

21: ILLEGAL TERMINATOR CODE <code_value>

I en læsespecifikation tillades kun terminator-koderne 0, 1, 2 og 3.

Resten af linien skippes.

22: TOO MANY READSPEC

Der må højst forekomme 3 læsespecifikationer i en readoperation.

Resten af linien skippes.

23: ILLEGAL CASENO <case_no> <no>

<case_no> det læste nummer

<no> det forventede nummer

Uoverensstemmelse mellem antallet af duetaktioner i caseliste og et specificeret casenummer.

Resten af linien skippes.

24: TOO MANY DUETREF IN ONE LINE

I en executeliste tillades højst 7 duetnavne i en linie hvortil der er knyttet et duetpunkt. Hvis et duetpunkt skal omfatte mere end 7 duetaktioner, må disse samles i en ny executeliste. Resten af linien skippes.

25: ILLEGAL USE OF D0

d0 tillades ikke i entrypunktdefinition, executeliste, if-operation efter else
Resten af linien skippes.

26: OBS! DUETBLOCK TOO BIG <size> <max_size>

<size> længden i ord af den oversatte duetblok
<max_size> den maximale bloklængde i ord.
Blokken må deles op i to mindre blokke.

27: ILLEGAL DIGITS SPEC <no_a> <no_b>

I 'digits (a, b) of' kræves $a \geq b$

28: ILLEGAL INDEX VALUE <index_value>

En konstant indexværdi er angivet udenfor grænserne af den pågældende arrayvariabel.
Resten af linien skippes.

29: ILLEGAL SETNO <setno>

Duetoversætteren kontrollerer ikke om et sæt er defineret i ld-beskrivelsen, kun om sætnummeret ligger indenfor intervallet 1 - max_def_sætnr. Resten af linien skippes.

30: ILLEGAL RECORDTYPE <no>

Individtypen tillades ikke. Resten af linien skippes.

31: ILLEGAL CHARACTER <text>

<text> den sidst læste syntaktiske enhed indeholder et ulovligt tegn. Resten af linien skippes.

32: ILLEGAL BYTE <text>

<text> den sidst læste syntaktiske enhed kan ikke genkendes. Resten af linien skippes.

33: ILLEGAL DUETNAME <d_name>

<d_name> duetnavnet må ikke overstige d1023. Resten af linien skippes.

34: TOO MANY DIGITS IN NUMBER <no>

<no> antal cifre i et tal må ikke overstige 15

Resten af linien skippes.

35: ILLEGAL IDENT <text>

<text> den sidst læste syntaktiske enhed er registreret som variabelidentifikator, men variabelen kendes ikke.

Resten af linien skippes.

36: ledig

37: LISTAREA CANNOT BE CREATED <area_name> <no>

<area_name> listarealets navn

<no> fejlværdi fra oprettelsesforsøget

Der mangler ressourcer (segmenter, katalogindgange eller arealprocesser) til oprettelse af areal til listning. Duetprogrammet oversættes uden listning.

38: BLOCKEND MISSING <block_no>

<block_no> sidste behandlede duetblok

Duetprogramteksten afsluttet (end_medium/afsnit_slut uden blokend.

39: DOUBLE USED VARIABLE NAME <varname> <varno>

To variable i ldbeskrivelsen erklæret med samme navn. Systemfejl. Denne fejl burde være fanget af ld-oversætteren. Kontakt værktøjsvedligeholdelsen.

40: ASSIGN SYMBOL MISSING

:= mangler i en linie i en computeliste.

41: ILLEGAL DELIMITER <text>

<text> den ulovlige syntaktiske enhed. Syntaksfejl i et numerisk udtryk, resten af linien skippes.

42: ILLEGAL OPERATOR <text>

<text> den ulovlige syntaktiske enhed. Syntaksfejl i et numerisk udtryk, resten af linien skippes.

43: ILLEGAL OPERAND <text>

<text> den ulovlige syntaktiske enhed. Syntaksfejl i et numerisk udtryk, resten af linien skippes.

44: OPDSTAK OVERFLOW <no>

For mange parentesniveauer i et numerisk udtryk. Resten af linien skippes.

Udtrykket bør opdeles med explicit lagring af mellemresultater.

45: OPTSTAK OVERFLOW <no>

Som 44.

46: WORKVAR OVERFLOW

Et numerisk udtryk er for kompliceret til at kunne udregnes med de 4 eksisterende arbejdsregistre.

Resten af linien skippes.

47: ILLEGAL OPERATOR COMBINATION

Syntaksfejl i numerisk udtryk, resten af linien skippes.

48: ILLEGAL LENGTH OF TEXT/AGGR <length1> <length2>

Ved 'modify array1 := array2' kræves at længden af et enkelt element i de to arrays er den samme.

<length1>	}	antallet af teksttegn/aggregatbytes
<length2>		

Ved modify af simple text-/bits variable kræves at længden af venstreside-variablen \geq længden af højreside-variablen.

Resten af linien skippes.

49: SYNTAX: <text>

<text> den sidst læste syntaktiske enhed
Denne sproglige konstruktion er ikke tilladt.
Oversætteren fortsætter med at læse den næste syntaktiske enhed i linien.

50: ledig

51: NOT DUETREG IN OLDDUETREG

Det areal, der er specificeret som oldduet indeholder ikke noget duetregister. Der dannes ikke nyt duetregister.

52: ILLEGAL VERSION OF OLDDUET <version> <old_version>

<version> specificeret som fp-parameter
<old_version> version af gl. duetregister
Hvis duetversion er specificeret som fp-parameter, skal de to versionsnumre være ens.

53: ledig

54: DUETWORK CANNOT BE CREATED

Der mangler ressourcer (segmenter, katalogindgange eller arealprocesser) til at oprette et arbejdsareal for oversættelse af duetblokke. Der dannes ikke nyt duetregister.

55: INDEX MISSING <varname> <varno>

Den specificerede variabel er en arrayvariabel, index skal anføres. Resten af linien skippes.

56: ILLEGAL VERTICAL OPERAND OR POSITION

I et printlayout tillades højst 70 linieskift ad gangen (svarende til én side). Trykkepositionen skal angives i intervallet 1-127.

57: ILLEGAL NUMBER OF PRINCIPALS OR DECIMALS

I et printlayout tillades højst 15 principaler og højst 7 decimaler. Hvis decimaler er angivet, må hverken principaler eller decimaler være 0.

58: POSITION + FIELD LENGTH > 132

Feltet kan ikke trykkes indenfor den givne liniebuffer på 132 tegn.

59: BLOCK LOCATION ERROR

Systemfejl ved opdatering af newduet. Kontakt værktøjsvedligeholdelsen.

60: BLOCK SEQUENCENO ERROR

Bloknumre i teksten skal komme i stigende rækkefølge. Blokken skippes.

61: SYNTAX OR ILLEGAL VARTYPE <text>

<text> er et læst variabelnummer eller -navn.

Enten er der angivet en variabelreference på et sted hvor en sådan ikke tillades, eller også er den pågældende variabeltype ulovlig.

Resten af linien skippes.

62: SYSTEM ERROR: VARIABLE NAME ADDRESS CONFLICT <no> <adr>

Systemfejl under opbygning af tabeller til genkendelse af variabelnavne. Kontakt værktøjsvedligeholdelsen.

63: SYNTAX ERROR BEFORE BLOCKSTART <text>

<text> er den sidst læste syntaktiske enhed. Før blokstart tillades kun tomme linier og kommentarlinier. Resten af teksten frem til førstkommande blokstart (begin <bloknr>:) skippes uden yderligere udskrift.

64: UNKNOWN DUETBLOCK SPECIFIED FOR TRANSLATION <blockno>

Den specificerede duetblok findes ikke i duetprogramteksten.

65: SYSTEM ERROR: MISSING GOTO IN BYTEAKTION <no>

Programfejl i duetabler-oversætteren, kontakt værktøjsvedligeholdelsen.

66: INSERT DUETBLOCK: EXISTING BLOCKNO <blockno>

Den specificerede duetblok oversættes (syntaks kontrolleres), men indsættes ikke i newduet.

67: ILLEGAL USERNO FOR CHANGE/DELETE DUETBLOCK <blockno>

Den specificerede duetblok tilhører en anden bruger. I tilfælde af change oversættes blokken (syntaks kontrolleres) men afleveres ikke i newduet.

68: UNKNOWN DUETBLOCK SPECIFIED FOR DELETE <blockno>

Den specificerede duetblok findes ikke i oldduet.

69: TOO MANY MOVEWORDS IN MODIFY ARRAY <no>

<no> det samlede antal ord der flyttes
i én modifylinie må ikke overstige
2047.

Arrayet må enten flyttes med et element ad gangen,
eller det må deles op i kortere arrays i ld-beskriv-
elsen.

Resten af linien skippes.

70: DUET PROGRAM HEAD MISSING

Ved oversættelse med henblik på skrivning af helt
nyt duetregister skal duetprogramhovedet læses
med.

71: PROGRAM NO IN TEXT DO NOT MATCH WITH OLDDUET <programno> <old_programno>

<programno> programnr læst i duetprogramteksten
<old_programno> programnr registreret i oldduet
De to numre skal være ens.

72: DUET PROGRAM NAME TOO LONG

Duetprogramnavnet må ikke indeholde mere end 17 tegn. Navnet skæres af til 17 tegn.

73: SPECIFIED LD-NUMBER DO NOT MATCH LD-FILE <ldno> <old_ldno>

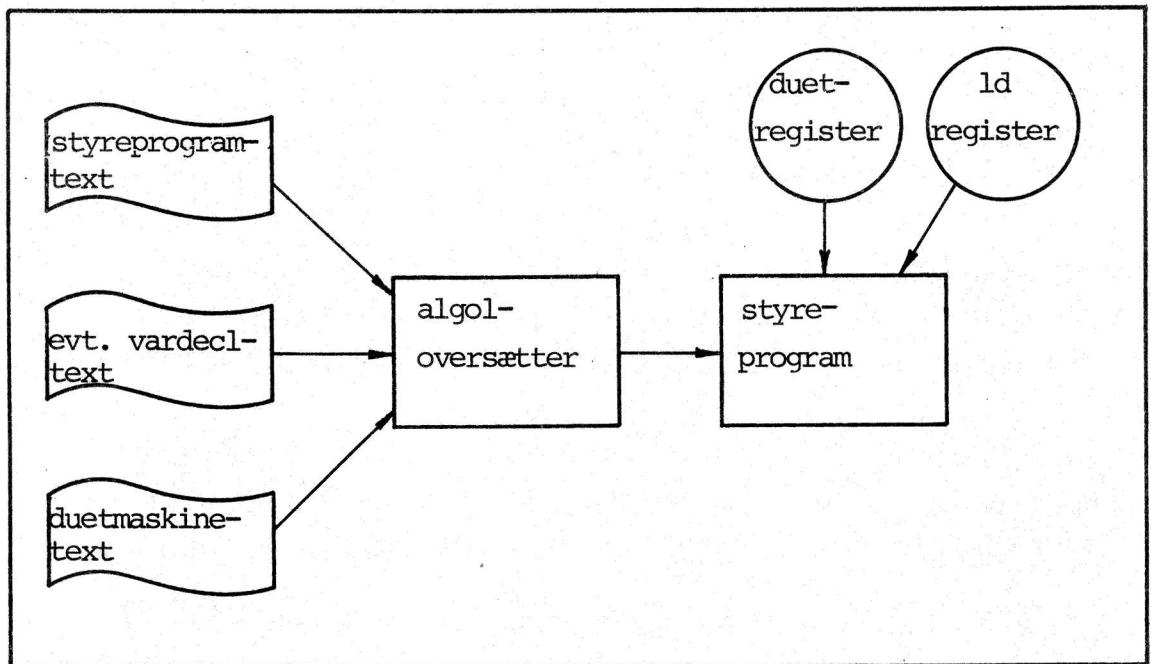
<ldno> ldnummer læst i duetprogramteksten
<old_ldno> ldnummer registreret i oldduet
De to numre skal være ens.

74: SYNTAX ERROR IN DUET PROGRAM HEAD

Kørslen afbrydes.

5. Duetsystemet i et styreprogram

Et oversat duetprogram lagret i et duetregister kan udføres af et styreprogram indeholdende fortolkeren 'duetmaskinen'.



figur 5.1: Det udførende duetsystem

I dette afsnit beskrives, hvordan et styreprogram etableres.

Telefaktsystemets 'teleop' og dataentersystemets 'telescop' er eksempler på sådanne styreprogrammer. Læsere, der skal kode duetprogrammer til disse systemer, kan derfor nøjes med at læse dette kapitel overfladisk.

5.1 Duettekster og algol blokstruktur

Duetmaskinen består af tre tekster, der skal indkopieres på passende steder i styreprogrammet med 'algol copy' som vist i figur 5.2. Desuden skal en eventuel erklæringsfil genereret af sodald-oversætteren indkopieres.

```

begin    <*ydre blok af styreprogram*>
.
.       erklæringer i ydre blok
.
algol   copy.duetttext1;    <*dueterklæringer*>
.
.       initialisering i ydre blok
.
init_duet1 ;                <*duetinitialisering*>

begin   <*indre blok af styreprogram*>
.
.       erklæringer i indre blok
.
procedure duetalgol(nr);    <*skal erklæres af brugeren*>
algol   copy.duetttext2;    <*dueterklæringer*>
algol   copy.vardecl;       <*variabelerklæringer fra sodald*>
.
.       initialisering i indre blok
.
init_duet2;                <*duetinitialisering*>
.
.       styreprogram før duetmaskine
.
algol   copy.duetttext3;    <*udførende duetmaskine*>
.
.       brugerafslutning
.
close_duet;                <*luk databasens filer*>
end     indre blok;
end     ydre blok;

```

figur 5.2: struktur af styreprogram

De tre tekster hedder

```
duetttext1
duetttext2
duetttext3
```

Disse indkopierer selv sodasystemets dbms-procedurer i form af 5 tekster

```
sodatext1
sodatext2
sodatext3
tflytproc
ldfields
```

og duetmaskinen kalder den eksterne procedure

```
read_general
```

samt primula-procedurerne, der findes på tekstform i

```
prim
```

Disse filer må alle være til stede på disk, når styreprogrammet oversættes. Primula-procedurerne oversættes ved kaldet 'i prim' før kaldet af algoloversætter.

duetttext1

duetttext1 indeholder dels erklæringer af variable, der benyttes som øvre grænse for arrayerklæringer i duetttext2, dels initialiseringsproceduren 'init_duet1' og fejludskriftsproceduren 'print_datafejl'.

'print_datafejl' er erklæret i blok1 for at give brugeren mulighed for at ændre formuleringen af datafejl-udskrifter ved at erklære en tilsvarende procedure i en indre blok.

Endelig indkopierer duetttext1 teksterne sodatext1 og ldfields fra sodasystemet.

duetttext2

duetttext2 indeholder erklæring af alle duetsystemets øvrige variable og procedurer. Den indkopierer desuden sodatext2, som selv indkopierer sodatext3 og tflytproc fra sodasystemet.

duetttext3

duetttext3 indeholder selve fortolkeren, dvs. den udførende del af duetmaskinen.

5.2 Initialisering og afslutning

Duetmaskinen initialiseres ved kald af initialiseringsprocedurerne 'init_duet1 og 'init_duet2. Hvis styreprogrammet er opbygget som en løkke, der aktiverer duetmaskinen flere gange, skal desuden proceduren 'init_duetmaskine' kaldes før hver ny aktivering. ('init_duetmaskine' kaldes første gang fra 'init_duet2').

Efter sidste aktivering af duetmaskinen skal styreprogrammet afslutte DBMS'en og prinzzonerne ved kald af proceduren 'close_duet'.

5.2.1 Integer procedure init_duet1(z);

Proceduren init_duet1 indlæser styreindividerne fra ld- og duetregistrene, og initialiserer dermed en række arraygrænser.

duetlog

Desuden skrives en duetlog på den zone der er specificeret som parameter til kaldet. Det kan enten være zonen 'out' eller 'fejclud'. Sidstnævnte skal i givet fald åbnes af styreprogrammet.

Før kaldet af init_duet1 skal styreprogrammet have initialiseret følgende variable:

duetareal	det antal ord der reserveres i duet-arrayet til udlægning af aktive blokke.
duetregnavn	navn på duetregistret
duetversion	hvis disse er forskellige fra 0,
ldversion	skal de være lig med de aktuelle versioner af duet- og ld-register.
ldreg_navn	navn og afsnitnummer for det beskriv-
ld_afsnit_	elsesregister, der indeholder ld-
nummer	registret.
max_kanal	antallet af zoner i zonearrayet prinz.
sd_extend_buf	størrelsen af den bufferudvidelsesreserve der ønskes pr. cf-master-fil (til brug for evt. extend_cf).

Procedurens returværdi er et bitmønster, der angiver resultatet af kaldet. Hvis alle bits er nul, er resultatet ok. De enkelte bits betyder:

- 1 : ikke ldregister i 'ldreg_navn'
- 1 shift 1: ldregistret er ikke dannet af sodald
- 1 shift 2: ldbeskrivelsen er ikke korrekt oversat
- 1 shift 3: forkert version af ldregister
- 1 shift 4: ld-styreindividet kan ikke findes

- 1 shift 12: checksumfejl i duet-styreindivid
- 1 shift 13: ikke duetregister i 'duetreg_navn'
- 1 shift 14: ubenyttet
- 1 shift 15: forkert version af duetregister
- 1 shift 16: duetareal for lille til største duetblok

5.2.2 Integer procedure init_duet2;

Proceduren init_duet2 indlæser resten af ld- og duetregistrene, nulstiller alle variable erklæret i ld-beskrivelsen og initialiserer tegnklassetabellen til brug for read-operationen. Desuden initialiseres prinz-zonerne til eventuel primulatrykning, og init_duetmaskine kaldes (jf. afsnit 5.2.3).

Før kald af init_duet2 skal brugerprogrammet have udført følgende initialiseringer:

prinz alle zoner i zonearrayet skal være åbnet til diskarealer eller båndfiler, men zonerne må ikke bruges til skrivning før efter init_duet2.

readz skal åbnes til inddataarealet, hvis der skal læses inddata.

Efter kaldet af init_duet2 kan følgende variable efter behov redefineres af styreprogrammet:

alfa tegnklassetabellen kan ændres, hvorefter intable(alfa) skal kaldes igen (jf. read, afsnit 3.2.11 og afsnit 5.5.1)

maxprintpos (=132) længde af liniebufferen til print

maxref (=20) se afsnit 5.5.1

testkanal (=0) testudskrifter trykkes som standard på zonen out, men kan med testkanal dirigeres til en af zonerne i prinz.

Desuden kan de variable der assignes af init_duet-maskine redefineres, se næste afsnit.

Procedurens returværdi er et bitmønster, der angiver resultatet af kaldet. Hvis alle bits er nul, er resultatet ok. Hvis der optræder fejl i init_duet2 men ikke i init_duet1, må det altid være en systemfejl, og man bør da kontakte værktøjsvedligeholdelsen.

5.2.3 Procedure init_duetmaskine;

Proceduren init_duetmaskine sætter duetmaskinen i neutral tilstand. Den kaldes fra procedure init_duet2 før første indhop i duetmaskinen, og kan kaldes fra styreprogrammet før eventuelle følgende indhop.

Den initialiserer følgende variable, som efter behov kan redefineres af styreprogrammet bagefter:

kanal := 1; prinz-zonenr for resultatudskrifter

```

datafejl_kanal     := 0;
programfejl_kanal := 0;
systemfejl_kanal  := 0;
    
```

fejludskrifter trykkes som standard på zonen fejlud, men kan ved ændring af kanalnummeret dirigeres til en af zonerne i prinz (jf. select, afsnit 3.2.14).

```

datafejlakt (1: max_datafejl)
programfejlakt (1: max_programfejl)
systemfejlakt (1: max_systemfejl)
    
```

nulstilles alle med betydningen: udhop af duetmaskinen ved alle fejl. Kan ændres til et positivt tal med betydning: hop til programpunkt, eller ændres til et negativt tal med betydning: fortsæt efter fejlen. (jf. select, afsnit 3.2.14).

```
max_instruktion := 8 388 607;
```

øvre grænse for antallet af duet-instruktioner der kan udføres inden for én aktivering af duetmaskinen. Hvis der aktiveres flere duetnavne end dette antal, antager duetsystemet, at der er en uendelig løkke i duetprogrammet. Denne kontrol får dog kun betydning, hvis styreprogrammet redefinerer max_instruktion til et lavere tal.

```
std_assign := 0;
```

når std_assign = 0 assignes der ikke standardværdi ved læsning af standardmærke. Kan ændres til 1 af styreprogrammet eller ved duetoperationen select hvis standardværdiassign ønskes (jf. read, afsnit 3.2.11).

```
instruktionstal := 0;
```

instruktionstælleren nulstilles. Efter udhop af duetmaskinen kan man af denne variabel aflæse hvor mange duetinstruktioner der er udført.

5.2.4 Procedure close_duet;

Proceduren close_duet skal kaldes af styreprogrammet før afslutning af kørslen, dvs. efter sidste udhop fra duetmaskinen.

Den lukker alle filer i databasen, og for bs-filer gemmes de nødvendige oplysninger i kataloghalen.

Desuden afsluttes skrivningen på hver af prinzzonerne således:

- liniebufferen tømmes ud på zonen (hvorved der trykkes en ekstra - muligvis tom - linie),
- der skrives end-medium på arealet og
- zonen lukkes

Derimod må styreprogrammet selv lukke zonen fejlud, da det er tænkeligt at man ønsker adgang til denne zone efter 'close_duet'.

Hvis styreprogrammet ønsker selv at sørge for afslutningen af prinzzonerne, kan man i stedet kalde sodasystemets procedure 'close_soda'.

close_soda

5.3 Fejlprocedurer

I alle fejlsituationer aktiverer duetmaskinen en af tre fejlprocedurer, nemlig 'duetdatafejl', 'duetprogramfejl' eller 'duetsystemfejl', afhængigt af om situationen skyldes fejl i inddata, i duetprogrammet eller i det omgivende system.

Fejlproceduren udskriver en meddelelse på den valgte fejludskriftszone, hvorefter den enten returnerer eller hopper ud af duetmaskinen eller hopper til et programpunkt i duetprogrammet (se afsnit 3.2.14, select og afsnit 5.2.3, init_duet-maskine).

5.3.1 Duetdatafejl

Procedure duetdatafejl kaldes ved fejl i inddata. Den kalder selv udskriftsproceduren 'print_datafejl', som er erklæret i yderste blok, men kan redefineres af styreprogrammet på et andet blokniveau (jf. afsnit 5.1).

Standardformatet for datafejl-udskrifter er

<pre>***** <fejltxt> { <fejlparameter> } 2 1</pre>

og alle datafejl-udskrifter er danske. I de fejl-udskrifter, der stammer fra read-operationen, er den første fejlparameter altid navnet på den variabel der er indlæst til, lagret i 'læsvar_navn'.

I det følgende gennemgås datafejludskrifterne i alfabetisk rækkefølge

DECIMALER MANGLER VED <var_navn>

Der er læst decimalpunktum efter et numerisk felt, men ingen decimaler

FORKERT DELIM EFTER <var_navn>

Det sidst læste felt er ikke afsluttet med alm. delimiter som krævet i læsespecifikationen

FORKERT DELIM ELLER LINIE FOR LANG EFTER <var_navn>

Det sidst læste felt er hverken afsluttet med alm. delimiter eller strengslut-delimiter.

FOR MANGE DECIMALER VED <var_navn> <max_decim>

Feltet i inddata har flere decimaler end den aktuelle læsevariabel tillader.
<max_decim> angiver det højeste tilladte decimalantal.

INDIVID MED ANGIVEN NØGLE FINDES ALLEREDE <sæt nr>

Individet afvises ved put efter create.

INDIVID MED ANGIVEN NØGLE FINDES IKKE I SÆT <sæt nr>

Ukendt individ forsøgt læst med get.

LINIE FOR LANG EFTER <var_navn>

Det sidst læste felt er ikke afsluttet med strengslut-delimiter (lineskift) som krævet i læsespecifikationen.

STANDARDMÆRKE ULOVLIGT FOR <var_navn>

Der er læst standardmærke til en variabel, der ikke tillader standardværdi.

SYSTEMFEJL: OPERATION PT. ULOVLIG

Denne duetdatafejl aktiveres altid efter en duet-programfejl eller en duetsystemfejl.

TEXTAFSLUTNING MANGLER VED <var_navn>

Et tekstfelt er afbrudt af streng-slut-delimiter (lineskift) før den afsluttende tekstdelimiter (apostrof)

TEXT FOR LANG VED <var_navn> <max_lgd>

Det læste tekstfelt er længere end den specificerede læsevariabel.
<max_lgd> angiver den højeste tilladte tekstlængde.

ULOVLIGT FELT EFTER <var_navn>

Indholdet af datafeltet efter det sidste korrekte felt svarer til typen af den/de anførte læsespecifikationer.

ULOVLIG VÆRDI FOR <var_navn> <værdi>

Værdien af det læste felt ligger udenfor det tilladte værdispektrum for den aktuelle læsevariabel.

<værdi> angiver den læste værdi for et numerisk felt, eller antallet af læste tegn for et tekstfelt.

5.3.2 Duetprogramfejl

Proceduren 'duetprogramfejl' kaldes, når duetmaskinen møder ulovlige konstruktioner i duetprogrammet.

Standardformatet for programfejludskrifter er

```

*****PROGRAM ERROR fnr: CALL FROM:  BLOCK INSTR  OPT   (ADR  PIL  ART)
                                xxxxxxx  xxx  xxx xxxxxxx  xx  xx  xx
<fejltxt> { <fejlparameter> } 3
                                                0
    
```

fnr er det aktuelle fejlnr

CALL FROM er enten DUET eller navnet på en db-operation

BLOCK }
INSTR }
OPT } angiver bloknummer, duetnavn og duetoperation ved det ulovlige sted i duetprogrammet

ADR }
PIL }
ART } interne værdier i duetmaskinen, som kan hjælpe værktøjsvedligeholdelsen i tilfælde af fejl i duetsystemet.

I det følgende gennemgås duetprogramfejlene i fejlnummerorden.

1. ILLEGAL BLOCKNO OR BLOCK MISSING <block_no>

Forsøg på aktivering af en ikke eksisterende duetblok (jf. execute, afsnit 3.2.1)

2. ILLEGAL BLOCKREF, USER CONFLICT <block_no> <user_no>

Forsøg på aktivering af duetblok tilhørende en anden bruger.

<user_no> er brugernummeret på den ulovlige blok (jf. execute, afsnit 3.2.1)

3. ILLEGAL ENTRYPOINT <entry_no>

Forsøg på aktivering af en duetblok med indhop i et ikke eksisterende entrypunkt (jf. execute, afsnit 3.2.1)

4. Ledig**5. ILLEGAL VARIABLE NUMBER <var_no>**

Ved en indirekte variabelreference refereres til en ikke eksisterende variabel.

6. INDEX <index_value>

Ulovlig indexværdi ved reference til arrayvariabel.

7. ILLEGAL CHANNEL NUMBER <channel_no>

Med select print er der valgt en ikke eksisterende zone i zonearrayet prinz, dvs. channel_no > max_kanal, eller channel_no = 0 for resultatudskrifter (jf. select, afsnit 3.2.14)

8. INSTR. COUNT EXCEEDED <inst_count> <max_inst>

instruktionstælleren overstiger 'max_instruktion', måske uendelig løkke i duetprogrammet.

9. ILLEGAL SET NUMBER <set_no>

ulovligt sætnummer i db-operation

10. USAGE CONFLICT <set_no>

db-operation i uoverensstemmelse med sæterklaringens usage-specifikation i ld-beskrivelsen

11. RECORD STATE ILLEGAL FOR THIS OPERATION <set_no> <record_state>

ulovlig db-operation i forhold til sættets
individtilstand.

record state:

- 0 efter open
- 1 efter get
- 2 efter put (direkte acces) eller efter fejl
- 3 efter delete (direkte acces)
- 4 efter create
- 5 efter newset
- 6 efter next
- 7 efter put (sekventielt)
- 8 efter delete (sekventielt)
- 9 efter kædeslut ved delete
- 10 efter kædeslut ved next
- 11 efter kædeslut i tom kæde
- 12 efter end-of-set

12. MOVE ERROR <set_no> <text> <addr> <value>

Fejl ved flytning mellem variable og felter i
forbindelse med en db-operation.

<addr> og <value> er altid adressen på og værdien af den aktuelle variabel.

<text> er en yderligere forklaring af fejlårsagen:

SPILL DURING TRANSFER FROM VAR

Værdien i en variabel kan ikke rummes i det aktuelle felt.

INDEX

værdien i en variabel benyttet som index ligger uden for et arrays grænser, eller overstiger det aktuelle antal repetitioner for en repeterende feltgruppe.

ILLEGAL NUMBER OF REPET. IN RPG

repetitionsantal ved oprettelse af repeterende feltgruppe uden for det tilladte interval.

13. SET CLOSED FOR SEQ. ACCESS BY NEWSET ON ANOTHER SET <set_no>

14. DAUGHTER RECORDS ASSOCIATED WITH CURRENT RECORD

Nedlæggelse af masterfil-individ ikke tilladt, når individet har datterindivider tilknyttet.

15. ledig

16. CURRENT RECORD REMOVED FROM DB <set_no>

Et individ, der er læst med get eller next findes ikke i filen ved put eller delete.

17. MOTHER RECORD MISSING IN SIDE CHAIN <set_no>

Ved indsættelse af nyt individ i en listefil mangler det sidemoderindivid, der blev registreret ved create.

18. MOTHER RECORD REMOVED FROM DB <set_no>

Moderindividet for et sæt (sættype L) er fjernet efter newset.

19. ILLEGAL STATE FOR MOTHER SET <set_no> <record state>

Der findes ingen current record hentet fra databasen i modersættet ved newset (sættype L).

20. ILLEGAL CURRENT RECORD TYPE IN MOTHER SET <set_no>

Ved newset (sættype L) er current record i moder-sættet en individtype, der ikke kan være moder for aktuelt sæt.

21. RECORD TYPE ILLEGAL IN SET <set_no> <record_type>

Den specificerede individtype ved create tilhører ikke sættet.

22. PRINTVALUE EXCEEDS FIELD RANGE <channel> <value>

Layoutsprængning ved print. I resultatarealet trykkes værdien som '????'
<channel> angiver den aktuelle resultatzone
<value> er den for store værdi.

23. ILLEGAL PRINTPOSITION <start_pos>

Startpositionen ved print er specificeret uden for liniebufferen.

24. PRINT FIELD EXCEEDS LINE BUFFER <start_pos> <antal>

Beregnet slutposition ved print ligger uden for liniebufferen.

25. ILLEGAL BS.OPERATION, POSITION AFTER EOF <set_no>

Ved get eller newset (sættype B) ligger den specificerede position efter end-of-file.

26. ILLEGAL DELETE POSITION <set_no>

Current record i sættet (sættype 2) kan ikke fjernes, fordi et andet sæt i samme fil har en current record efter denne position.

27. SET NOT OPEN FOR SEQUENTIAL ACCESS <set_no>

Newset har ikke været kaldt, eller end-of-set er nået.

5.3.3 Duetsystemfejl

Proceduren 'duetsystemfejl' kaldes, hvis duet-maskinen finder inkonsistenser i duetprogrammet, ld-beskrivelsen eller databasen.

I tilfælde af systemfejl bør man normalt kontakte værktøjsvedligeholdelsen.

Standardformatet for systemfejludskrifter er

*****SYSTEM ERROR fnr:	CALL FROM:	BLOCK INSTR	OPT	(ADR	PIL	ART)	
	xxxxxxx	xxx	xxx	xxxxxxxx	xx	xx	xx
<fejltext>	{ <fejlparameter> }			3			
				0			

Oplysningerne i systemfejllinien er de samme som i en programfejllinie (se afsnit 5.3.2). I det følgende gennemgås systemfejlene i fejlnummerorden.

Oplysningerne i <> er navnene på de variable, der udskrives som yderligere indikering af systemfejlen.

1. DUETREL <duetrel> <duetstop>

Inkonsistens i det binære duetprogram i forbindelse med aktivering af en duetinstruktion.

2. BIT23 = 0

Inkonsistens i det binære duetprogram i forbindelse med udførelsen af en duetinstruktion.

3. 'DUETNAVN'

Inkonsistens i det binære duetprogram i forbindelse med udførelsen af en duetinstruktion.

4. VALUE OF 'UDTRYK' <val> <duetstop>

Inkonsistens i det binære duetprogram eller fejl i duetmaskinens procedure 'udtryk'.

5. BIT23 = 1 <duetstop>

Inkonsistens i det binære duetprogram eller fejl i duetmaskinens procedure 'udtryk'.

6. LD-TABLES <sætnr>

Inkonsistens i den binære ld-beskrivelse.

7. CHECKSUM <sætnr>

Checksumfejl i databaseindivid, inkonsistens i databasen.

8. RECORDLENGTH <sæt nr>

Inkonsistens i databasen: længden af et individ læst med get, next eller lookup svarer ikke til db-beskrivelsen.

9. CONNECT MISSING IN SIDE CHAIN <sæt nr>

Inkonsistens i databasen: det aktuelle listefil-individ er ikke connected til alle tilhørende sidemødre.

10. FILE EXPANSION IMPOSSIBLE <resultcf>

Ved put er filudvidelse med extendcf mislykket.

<resultcf> angiver årsagen:

2 filen er udvidet, men zonebufferen for lille (variablen sd_extend_buf må initialiseres med en større værdi, hvis der skal indsættes mange individer i én kørsel).

>10000 ressourcemangel.

11. RECORD CREATION TOO EXPENSIVE <resultcf>

Indsættelse af nyt individ (put efter create) mislykket.

<resultcf> angiver årsagen (se ref. 4).

12. DELETION OF LAST RECORD IN FILE

Der skal altid være mindst ét individ i en cf-masterfil.

13. SKIPWORD

Inkonsistens i det binære duetprogram i forbindelse med exit til et programpunkt.

14. OVER/UNDERFLOW <staktrin>

Fejl i duetmaskinen ved stakning eller afstakning, evt. for mange niveauer af execute-operationer.

15. ERROR IN BLOCK TRANSFER <bloknr> <fejltext>

Fejl ved læsning af en duetblok.
<fejltext> angiver yderligere årsag:

BLOCKLENGTH ERROR**BLOCKNO ERROR****CHECKSUM ERROR****RECORDNO ERROR**

angiver alle inkonsistens i det binære duetregister.

BLOCK/LD VERSION INCONSISTENCY

Duetblokken er oversat med en anden version af ld-beskrivelsen end den det kørende system har indlæst.

16. MOVE ERROR <text>

Inkonsistens opstået ved flytning mellem variable og felter i forbindelse med en db-operation. <text> er en yderligere forklaring af fejlårsagen:

ILLEGAL ENTRYADDR IN 'FLYTTETAB'**ILLEGAL MOVEINSTRUCTION: <type>, <addr>**

angiver begge inkonsistens i den binære ld-beskrivelse.

17. ILLEGAL TYPE OF VALUE ELEMENT <type>

Inkonsistens i det binære duetprogram eller fejl i duetmaskinens procedure 'take value'.

5.4 Algolspecialaktioner

duet_algol

Algolspecialaktioner i duetprogrammet udføres ved at duetmaskinen kalder proceduren 'duet_algol(nr)' med algolaktionsnummeret som parameter.

Denne procedure skal erklæres af styreprogrammet på det blokniveau, hvor duettext2 indkopieres. Proceduren kan f.eks. se ud som vist i figur 5.3.

```
procedure duet_algol(nr);
value nr; integer nr;
case nr of
begin
  begin <*aktion 1*>
    .
    .
    .
  end;
  .
  .
  .
  begin <*aktion x*>
    .
    .
    .
  end
end duet_algol;
```

figur 5.3: procedure duetalgol

Hver enkelt algolaktion må selv fortolke sine parametre ved successive kald af duetmaskinens integer procedure 'takevalue'.

duetparam

Variablen 'duetparam' fortæller hvor mange parametre der er angivet, og dermed hvor mange gange 'takevalue' skal kaldes.

takevalue

Returværdien af 'takevalue' angiver hvordan parametren er læst og hvor dens værdi er lagret, som vist i skemaet figur 5.4.

takevalue	værdi type	værdi lagret som heltal	værdi lagret som flydende tal
0	real	~	reg
1		værdi	reg
2	word	d.w, værdi	reg
3	long	d.l, værdi	reg
4	text	d.t	~
5	adr(procedure 'varadr' er kaldt	

figur 5.4: Resultat af 'takevalue'

varadr

Proceduren 'varadr' kaldes fra 'takevalue', når parameteren er specificeret som adr(...). I dette tilfælde angiver 'type' hvor den resulterende variabel findes, som vist i figur 5.5. Endvidere angiver 'lgd', hvor mange ord variabelen fylder.

'type'	variabel- type	variabel findes i	'lgd'
0	text	d.t	textlgd i ord
1	~	~	~
2	word	d.w	1
3	long	d.l	2
4	real	d.r	2
5	recno	d.w	1
6	date	d.w	1
7	bits	d.t	aggregatlgd i ord

figur 5.5: Resultat af 'varadr'

Bemærk: hvis variabelen er en arrayvariabel, er det altid adressen på 1. element der udregnes.

goto

En algolspecialaktion må normalt ikke indeholde et 'goto' som leder ud af proceduren. Hvis behandlingen af duetprogrammet skal fortsættes, er det nødvendigt at 'duet_algol' returnerer til det sted, hvorfra den blev kaldt. Dog er det tilladt at hoppe til en label uden for duetmaskinen, og dermed afbryde duetprogrammet.

5.5 Reserverede algolnavne

Nedenfor findes en fortegnelse over samtlige algolnavne, der erklæres globalt af duetteksterne. Desuden er alle navne, der erklæres af soda-dbms'en, reserveret (jf. ref. 2).

Fortegnelsen er opdelt i tre lister omfattende algolnavne, som

- 1) brugeren skal/må benytte,
- 2) må benyttes som arbejdsvariable,
- 3) ikke må røres.

Variablene i den første liste er ordnet efter brug, mens de to andre lister er opstillet alfabetisk.

5.5.1 Brugerkendte algolnavne

Navne vedrørende algol-specialaktioner

(jf. afsnit 5.4 og 3.2.16).

takevalue

procedure takevalue;

kaldes fra algol-specialaktioner for at hente aktionens parametre.

l

long field l;

r

real field r;

w

integer field w;

t

integer array field t;

reg

real reg;

værdi

long værdi;

lgd

integer lgd;

type

integer type;

indeholder returværdier fra kald af 'takevalue', som beskrevet i afsnit 5.4. Kan derudover frit anvendes som arbejdsvariable.

d

integer array d(-3:maxd);

det basisarray, hvori alle variable erklæret i sodald samt alle konstanter er udlagt. Disse kan refereres fra algol-specialaktioner ved fieldning, enten med fieldvariable genereret af sodald-oversætteren, eller med en af standardfieldvariablene w, l, r eller t assignet af procedure takevalue.

duetalgol

procedure duetalgol(nr);

skal erklæres af styreprogrammet til at udføre algol-specialaktioner.

duetparam integer duetparam;
 angiver antallet af parametre til en algol-special-
 aktion.

Navne vedrørende læsning (jf. afsnit 3.2.10 og
 3.2.11)

read_general external procedure read_general (readz, linie,
 felter, felttype, linieindex);
 læser en inddatalinie, som beskrevet ved duet-
 operationen getline, til brug for operationen
 read. Normalt kodes indlæsningen i duetprogram-
 met, men i visse programtyper er det mest hen-
 sigtsmæssigt, at styreprogrammet selv indlæser
 datalinierne. I disse tilfælde skal styrepro-
 grammet kalde read_general med de ovennævnte pa-
 rametre, og duetprogrammet må i så fald ikke be-
 nytte operationen getline. Dette gælder f.eks.
 både 'teleop' og 'telescop'.

readz zone readz (128, 1, stderr);
skal åbnes af styreprogrammet inden kald af
 init_duet2 (jf. afsnit 5.2.2). Bruges som pa-
 rameter til evt. kald af read_general.

linie real array linie (1:25);
 felter long array felter (1:80);
 felttype integer array felttype (1:80);
 linieindex integer linieindex;
 bruges som parametre til read_general. I arrayet
 linie afleveres den læste inddatalinie som text.

alfa integer array alfa (0:127);
 initialiseres af init_duet2 med standard tegn-
 klassetabellen for indlæsning (jf. afsnit 3.2.11,
 specielt figur 3.35). Denne standardtabel kan
 ændres af styreprogrammet med efterfølgende kald
 af 'intable (alfa)'. Hvert element af alfa skal
 have formatet

tegnklasse shift 12 + tegnværdi

std_assign integer std_assign;
 se afsnit 5.2.3.

Navne vedrørende print (primulategykning)
 (jf. afsnit 3.2.12 og 5.2.1-4).

prinz zone array prinz (max_kanal, 128+38, 1, prinzproc);
 Hver zone i prinz skal åbnes til et outputareal
 før kald af init_duet2 (jf. afsnit 5.2.2).

max_kanal integer max_kanal;
 bestemmer antallet af zoner i prinz. Skal initia-
liseres af styreprogrammet før kald af init_duet1.

max_printpos integer max_printpos;
 bestemmer længden af primulasystemets liniebuffer
 i hver af prinz-zonerne. Initialiseres af init_
 duet1 til 132, men kan redefineres af styrepro-
 grammets ydre blok.

setprimula

```
procedure setprimula (zonenr);
integer zonenr;
hjelpeprocedure, der muliggør skrivning med write
på prinz-zonerne. Inden kald af write skal styre-
programmet kalde
    endprint (prinz (zonenr));
og efter write afsluttes med
    setprimula (zonenr);
```

kanal

```
integer kanal;
bestemmer hvilken prinz-zone der benyttes til ud-
skrifter med print. Sættes til 1 af init_duet-
maskine. Kan ændres af styreprogrammet eller med
operationen select, men må ikke nulstilles (jf.
afsnit 3.2.14 og 5.2.3).
```

Navne vedrørende udførelse af duetprogram

duetareal

```
integer duetareal;
skal initialiseres af styreprogrammet inden kald
af init_duet1. Bestemmer størrelsen (målt i ord)
af det duetarray hvortil duetblokke indlæses (jf.
afsnit 3.1.2 og 5.2.1).
```

max_ref

```
integer max_ref;
bestemmer 'halveringstiden' for blokbrug, hvori
duetsystemet registrerer hvor hyppigt hver en-
kelt duetblok bliver refereret. Med denne re-
gistrering kan man opnå at de oftest benyttede
blokke så vidt muligt bliver liggende i lageret
(duetarrayet) (jf. afsnit 3.1.2).
```


For at undgå at en blok, som i en periode har været benyttet meget, men derefter ikke refereres i lang tid, bliver ved med at ligge fast i lageret, halveres tællerne i blokbrug hver gang maxref blokreferencer er udført.

maxref initialiseres af init_duet2 til 20.

instruktionstal integer instruktionstal;
max_instruktion integer max_instruktion;
instruktionstælleren og dennes øvre grænse (jf. afsnit 5.2.3).

Navne vedrørende initialisering (jf. afsnit 5.2)

duetregnavn real array duetregnavn (1:2);
duetversion integer duetversion;
skal initialiseres af styreprogrammet inden kald af init_duet1 med arealnavn og versionsnr for det oversatte duetprogram.

ldregnavn real array ldregnavn (1:2);
ld_afsnit_nummer integer ld_afsnit_nummer;
ldversion integer ldversion;
skal initialiseres af styreprogrammet inden kald af init_duet1 med arealnavn, afsnitnummer og version for den oversatte ldbeskrivelse.

fejlud zone fejlud (128, 1, stderr);
skal åbnes og lukkes af styreprogrammet til brug for skrivning af duetlog/fejludskrifter, dersom disse ikke omdirigeres til out/prinz.

init_duet1 integer procedure init_duet1 (z);
 zone z;
 initialiseringsproceduren i duetttext1. Skal kaldes
af styreprogrammet med out eller fejlud som para-
meter, prinz-zonerne må ikke benyttes. Se afsnit
5.2.1.

init_duet2 integer procedure init_duet2;
init_duetmaskine procedure init_duetmaskine;
 initialiseringsprocedurer i duetttext2. Se afsnit
5.2.2 og 5.2.3.

close_duet procedure close_duet;
 afslutningsprocedure for lukning af zoner. Se
afsnit 5.2.4.

Bemærk: variable der skal initialiseres findes
også under print og udførelse af duetprogram.

Navne vedrørende fejludskrifter (jf. afsnit 5.3)

datafejl_akt integer array datafejl_akt (1: max_datafejl);
datafejl_kanal integer datafejl_kanal;
programfejl_akt integer array programfejl_akt (1: max_programfejl);
programfejl_kanal integer programfejl_kanal;
systemfejl_akt integer array systemfejl_akt (1: max_systemfejl);
systemfejl_kanal integer systemfejl_kanal;
 initialiseres af init_duetmaskine, kan ændres med
select-operationen eller af styreprogrammet. Se
afsnit 5.2.3.

max_datafejl integer max_datafejl;
max_programfejl integer max_programfejl;
max_systemfejl integer max_systemfejl;
 Øvre grænser for fejlakt-arrayene. Må ikke assignes
 af styreprogrammet.

print_datafejl procedure print_datafejl (z, fejlnr);
 zone z; integer fejlnr;
 se afsnit 5.3.1.

læsvarnavn long array læsvarnavn (1:4);
fejlværdi long fejlværdi;
 indeholder navnet på den sodavariabel, der sidst
 er benyttet til indlæsning, samt den ulovlige
 værdi ved kald af print_datafejl.

Navne vedrørende test

testa integer testa;
testb integer testb;
testc integer testc;
testd integer testd;
teste integer teste;
testf integer testf;
testg integer testg;
testh integer testh;
 testvariable, der kan assignes efter behov af
 styreprogrammet og/eller ved duetoperationen
 select (jf. afsnit 3.2.14).

testkanal integer testkanal;
 bestemmer hvor testudskrifter trykkes (jf. af-
 snit 5.2.2).

Øvrige navne

duetdato	integer duetdato;
duettid	integer duettid; assignes af init_duet1 med ISO-dato og klokkeslet (time-minut-sekund) for kørselstart, og udskrives på duetloggen.
duetsystem_version	integer duetsystem_version;
duetsystem_dato	integer duetsystem_dato;
duetsystem_init	real array duetsystem_init (1:2); indeholder versionsnr og udgivelsesdato samt initialerne for den person, der har ansvaret for den aktuelle udgave af duetsystemteksterne. Trykkes af init_duet1 på duetloggen.
systemtext	long array systemtext (1:4); indeholder duetprogrammets navn (fra duethovedet), som udskrives på duetloggen.
nl	boolean nl;
sp	boolean sp; assignes af init_duet1 med hhv. 'false add 10' og 'false add 32'.
tipotens	integer array tipotens (0:6); assignes af init_duet2 med titalspotenserne 1, 10, 100, ..., 1000000.
trykbits	procedure (z, antal, word); zone z; integer antal, word; udskriver indholdet af parameteren 'word' som bitmønster. Parameteren 'antal' angiver hvor mange bits af word, regnet fra højre, der trykkes.

takename

```
real procedure takename (ra);  
real array ra;  
simulerer increase. Er identisk med proceduren  
'hentnavn' i 'begin80', men har fået sit eget  
navn for ikke at genere de brugere der anvender  
begin80 i ydre blok af styreprogrammet.
```

5.5.2 Frie arbejdsvariable

integer	adr
integer	adrtype
integer	aktion
integer	antal
boolean	b
integer	fejlbits
integer	fejlnr
integer	formatinf
integer	fra
integer	ftype
long array field	haf
integer	i
integer array	ia (1:20)
integer array field	inf
integer	j
integer	k
integer	klasse
integer	kode
long array field	laf
long	lword
integer	minus
integer	mode
integer	norm
integer	nr
integer	opd
integer	opdtype
integer	operator
integer	opord
integer	opt
integer	pil
real array filed	raf
real	reg1

integer	spec
integer	stop
integer	sætnr
long	talværdi
integer	term
integer	termkrav
integer array field	text
integer	til
integer	vadr
integer	val
integer	var
long	varinf
integer	varnr
long	varord
integer	word

5.5.3 Utilgængelige algoritmanavne

procedure	afstak
label	aktion_fundet
integer array	basispos
label	bestem_element
integer array	blokbasis
integer array	blokbrug
integer	blokbruger
integer	bloklgd
integer array	bloklængde
integer	blokno
integer	bloknr
integer array	blokplac
integer	blokslut
label	checksoda
procedure	checkværdi
integer	dadr
integer	duetadr
integer	duetart
integer	duetbasis
procedure	duetdatafejl
integer	duetnavn
integer	duetoperator
integer	duetord
integer	duetpil
procedure	duetprogramfejl
integer	duetpunkt
integer	duetrel
long array	duetstak

integer	duetstop
procedure	duetsystemfejl
procedure	duettest
zone	duetz
label	element_fundet
integer	entrynr
label	exit
integer	exitnr
long	fejlværdi
integer	feltindex
procedure	flytblok
label	fortsæt
label	fortsæt_execute
label	fortsæt_layout
label	fortsæt_modify
label	genoptag_instruktion
integer	gl_duetnavn
integer	index
boolean	in_if
integer	instruktionstal
integer array	klumpbasis
integer array	klumpref
integer	lokalbase
integer	læsvar
integer	maxblok
integer	maxd
integer	maxklump
integer	maxstak
integer	maxvar
integer	maxvartext
procedure	nyblok
label	perform
integer	position

integer	primula_state
procedure	print_duetttest
procedure	print_flyttefejl
procedure	print_programfejl
procedure	print_systemfejl
blockprocedure	prinz_proc
integer	punktblok
integer	punktpil
integer array	punktstak
integer	refantal
integer	reftal
integer field	resultat1
integer field	resultat2
integer field	resultat3
integer field	resultat4
integer field	resultat5
label	skipterm
label	slut_læs
label	slut_short
procedure	stak
integer	stakpunkt
integer	staktrin
integer	stopadr
procedure	takenum
label	test_relation
procedure	tryk_identid
label	tryk_vertikal
integer array	typelængde
label	udfør
procedure	udtryk
procedure	varadr
integer array field	vartextbase
procedure	varværdi
procedure	værdigrænsekontrol

Appendix A. Syntaksbeskrivelse af duetsproget

Dette appendix indeholder en formel syntaksbeskrivelse af duetsproget. Syntaksen er beskrevet ved hjælp af INFORMAL-sproget (ref. 7), dog med den modifikation at konkateneringssymbolet -* er blevet erstattet af space.

Nedenfor gives en fortegnelse over INFORMAL-sprogets symboler og deres betydning. For en mere uddybende forklaring henvises til ref. 7, samt til SODA-manualens appendix A (ref. 2).

.=	definitionssymbol for syntaktisk enhed (SE),
;	afslutning af udsagn,
!	alternativer,
(.)	SE kan udelades eller angives netop én gang,
(*)	SE kan udelades eller angives vilkårlig mange gange,
(+)	SE skal forekomme mindst én gang,
' '	SE er en konstant tekststreng,
< >	indhold af et tegnsæt,
(:)	restriktionssymbol for datamængde.

I den efterfølgende syntaksbeskrivelse er alle grundbegreber defineret til sidst, mens alle andre syntaktiske enheder defineres snarest muligt efter det sted, de nævnes første gang.

```
duetprogram      . = duet_head  duet_block(*);

duet_head         . = duetprg_ident  nl  ld_ident  nl;

duetprg_ident     . = 'duetprogram'  duet_no  '/'  textconst;

duet_no          . = NUMBER;

ld_ident         . = ('localdata'  ld_no)      !
                  ('dataentry'  'ld'  ld_no);

ld_no            . = NUMBER;

duet_block      . = block_head  duet_instr(*)  block_end;

block_head       . = 'begin'  block_no  ':'  nl
                  entrypoint(*);

block_no         . = NUMBER (1 TO 255);

entrypoint       . = '->'  entry_no  '='  d_name  nl;

entry_no         . = NUMBER(1 TO 63);

block_end        . = 'end'  block_no  nl;

duet_instr      . = nl(*)  np(.)  d_name  ':'  duet_operation
                  (nl(.)  '>'  duet_operation) (*);
```



```
duet_operation      .= (execute_opt  !
                          modify_opt  !
                          compute_opt !
                          assign_opt  !
                          if_opt      !
                          action_opt  !
                          case_opt    !
                          for_opt      !
                          while_opt   !
                          getline_opt !
                          read_opt     !
                          print_opt    !
                          db_opt       !
                          select_opt  !
                          exit_opt     !
                          algol_opt    ) nl;

execute_opt        .= 'execute' execute_line(+) 's';

execute_line        .= (tno(.) pno(.) duetref (',' duetref)(*))(.) nl;

tno                 .= 't' NUMBER(1 TO 9);

pno                 .= 'p' NUMBER(1 TO 9);

duetref             .= d_name | block_ref;

block_ref           .= 'b(' block_spec ',' entry_spec ')';

block_spec          .= block_no | simpel_numvar;

entry_spec          .= entry_no | simpel_numvar;
```

```

modify_opt          . = 'modify' modify_line 's';

modify_line          . = (leftside modify_symbol rightside) (.) nl;

leftside            . = varref | array_var;

modify_symbol       . = := | :+ | :- ;

rightsided          . = IF leftside = array_var THEN
                        (array_var | 0 )
                        ELSE
                        (varref      |
                         num_const   |
                         text_const  |
                         char_const  |
                         digits_spec |
                         name_spec   |
                         var_spec    |
                          );

digits_spec         . = 'digits' '(' digitno ',' digitno ')'
                        'of' varref;

digitno             . = NUMBER(1 TO 15);

name_spec           . = 'name' '(' (varref | var_spec) ')';

var_spec            . = 'var' '(' (numvar | var_spec) ')';

compute_opt        . = 'compute' compute_line(*) 's';

compute_line        . = (numvar ':=') (+) num_expr;

num_expr            . = '-' (.) num_operand (num_opt num_operand) (*);

```


`action_line` `.= test_value ':' d_name nl;`

`test_value` `.= num_const | char_const | simp_wordvar;`

`else_action` `.= 'else' d_name;`

`case_opt` `.= 'case' simp_wordvar 'of'`
 `case_line(*) (else_action | 's');`

`case_line` `.= (caseno ':')(.) d_name nl;`

`else_action` see `action_opt`.

`for_opt` `.= 'for' simp_wordvar ':=' from_value ',' to_value`
 `'do' d_name nl;`

`from_value` `.= num_const | char_const | numvar;`

`to_value` `.= num_const | char_const | numvar;`

`while_opt` `.= 'while' relation 'do' dname;`

`relation` see `if_opt`

`getline_opt` `.= 'getline' simpel_textvar;`

`read_opt` `.= 'read' read_spec (',' read_spec)(*);`

`read_spec` `.= read_mode read_var;`

```

read_mode      . = 'n' term_code |
                't' term_code |
                'c' term_code   no_of_chars;

term_code      . = NUMBER(0 TO 3);

no_of_chars    . = NUMBER(1 TO 3);

read_var       . = IF read_mode = 't' THEN textvar
                ELSE numvar;

print_opt      . = 'print'
                (print_action | print_line | std_line) (*) 's';

print_action   . = d_name nl;

print_line     . = layout value_spec(.) nl;

layout         . = basispos_spec | print_spec;

basispos_spec  . = '<' 'h' position '>';

printspec      . = '<' verticalspec(.)
                (position layouttype layoutparam)(.) '>';

verticalspec   . = 'p' |
                'w' |
                (no_of_lines | simp_wordvar) 'l';

no_of_lines    . = NUMBER(1 TO 70);

```

```
position      . = fixed_pos | '(' simp_wordvar ');
fixed_pos     . = NUMBER(1 TO 127);
layouttype   . = 'n' | 't' | 'c';
layoutparam  . = TABLE layouttype = (
               't' § textlength,
               'c' § no_of_rep,
               'n' § fixed_sign(.) zero_repr(.)
                  zero_value(.) principals
                  ('.' decimals)(.) );
textlength   . = no | '(' simp_wordvar ');
no_of_rep     . = no | '(' simp_wordvar ');
no           . = NUMBER(1 TO 127);
fixed_sign    . = '-';
zero_repr     . = '*' | 'z';
zero_value    . = 'b';
principals    . = NUMBER(1 TO 15);
decimals     . = NUMBER(1 TO 7);
```



```

value_spec      . = IF layouttype = 't' THEN
                  (text_const |
                   textvar    |
                   name_spec)
                  ELSE
                  (num_const  |
                   char_const |
                   numvar     |
                   num_expr   |
                   digits_spec);

name_spec }
digits_spec }      see modify_opt

num_expr      see compute_opt

std_line      . = stdlayout vartext simpel_var;

std_layout    . = '<' std_layouttype pos1 ',' pos2 '>';

std_layouttype . = 's' | 'a';

pos1          . = fixed_pos;

pos2          . = fixed_pos;

vartext       . = IF std_layouttype = 's' THEN textkonst;

db_opt        . = create_opt |
                  lookup_opt |
                  get_opt    |
                  put_opt     |
                  delete_opt |
                  newset_opt ;

```

```

create_opt      . = 'create' set_spec itype_spec;

lookup_opt     . = 'lookup' set_spec;

get_opt        . = 'get' set_spec;

put_opt        . = 'put' set_spec;

delete_opt     . = 'delete' set_spec;

newset_opt     . = 'newset' set_spec;

set_spec       . = 's' set_no |
                's(' simpel_wordvar ')';

set_no         . = NUMBER(1 TO max_set);

itype_spec     . = 'i' itype_no |
                'i(' simpel_wordvar ')';
                <*record type*>

itype_no       . = NUMBER

select_opt     . = 'select' select_line(*) 's';

select_line    . = nl |
                (print_result_line |
                 print_error_line |
                 exit_line |
                 return_line |
                 test_line |
                 std_assign_line ) nl;

print_result_line . = 'print' 'on' result_channel;

result_channel . = NUMBER(1 TO 9);

```

```

print_error_line      .= 'print' (error_type | 'test') 'on' channel;

error_type            .= 'data' 'error' |
                        'program' 'error' |
                        'system' 'error';

channel              .= NUMBER(0 TO 9);

exit_line            .= 'exit' pno 'on' error_type
                      (error_no (',' error_no) (*)) (.);

pno                  see execute_opt

error_no             .= NUMBER(1 TO max_error);

return_line         .= 'return' 'on' error_type;

test_line           .= 'test' testvar testassign_symbol test_value;

testvar             .= 'a' | 'b' | 'c' | 'd' |
                      'e' | 'f' | 'g' | 'h';

testassign_symbol   .= ':= ' | ':+ ' | ':- ' ;

test_value          .= numvar | 'on' | 'of' |
                      testbit (',' testbit) (*);

testbit             .= NUMBER(0 TO 23);

std_assign_line     .= 'no' (.) 'stdassign' 'on' 'read';

exit_opt          .= 'exit' pno;

pno                  see execute_opt

```

```
algol_opt          . = 'algol' parameter_list(.);

parameter_list     . = '(' parameter (',' parameter) (*) ')';

parameter          . = num_const      |
                   .   text_const     |
                   .   char_const     |
                   .   varref         |
                   .   name_spec      |
                   .   adr_spec;

name_spec          . = see modify_opt.

adr_spec           . = 'adr' '(' (variable | var_spec) ')';

var_spec           . = see modify_opt.
```

Grundbegreber:

nl	.= <10> ';' textchar(*) <10> (+);
np	.= ('++' <12>) nl(*);
d_name	.= 'd' NUMBER(1 TO 999);
varref	.= simple_var array_var '(' index ')';
simple_var	.= variable (: no arrayspec);
array_var	.= variable (: arrayspec);
variable	.= 'v' NUMBER(1 TO max_var) var_ident;
index	.= numvar NUMBER(1 TO max_index);
numvar	.= varref (: vartype = word, long, real, date);
textvar	.= varref (: vartype = text);
simple_numvar	.= simple_var (: vartype = word, long, real, date);
simple_textvar	.= simple_var (: vartype = text);
simple_wordvar	.= simple_var (: vartype = word);
num_const	.= NUMBER ('.' NUMBER ());
text_const	.= ap textchar(*) ap;
ap	.= <39>; <* apostrof: ' *>
char_const	.= '.' shortchar shortchar(.) shortchar(.) '.';
shortchar	.= <letter, digit, ',-+;:;!/?/_=<>*()%&'>
textchar	.= <shortchar, '.'>

Index

action	3-42
action_line	A-6
action_opt	A-5
adr	3-76, 3-77
adr_spec	A-12
afsnitnr	4-9
aktionsliste	3-42
aktivt programpunkt	3-23, 3-75
a-layout	3-64
alfa	5-9, 5-38
algol	3-76ff
algolnavne	5-35ff
algol_opt	A-12
algolspecialaktion	3-76, <u>5-32ff</u> , 5-36
alternativ aktion	3-37, 3-42
alternativ værdi	3-37
ap	A-13
apostrof	3-17, 3-49, 3-51
arbejdsregister	3-35
array_var	3-14, A-13
assign_line	A-5
assignliste	3-36
assign_opt	A-5
assignoperator	3-27, 3-28, 3-30, 3-32, 3-34, 3-74
assign_value	A-5
assign variabel	3-36
automatisk normalisering	3-16, 3-29, 3-62

Index (fortsat)

basispos_spec	A-7
basisposition	3-37
beskrivreg	4-13
betinget oversættelse	3-24
bitvariabel	3-15
block_end	A-2, 3-9
block_head	A-2
block_no	A-2
block_ref	A-3
block_spec	A-3
blokend	3-9
blokhoved	3-9
bloknr	3-8, 4-14ff
blokreference	3-21
bosslinienr	4-3
brugernr	3-9, 3-10, 4-10
brugertrimning	2-3
bs-fil	3-67, 5-13
case	3-40
case_line	A-6
case_opt	A-6
cf-listefil	3-67
cf-masterfil	3-66, 3-67
change	4-14, <u>4-15</u>
channel	A-10
char_const	A-13
char_print	3-59
chars	3-51

Index (fortsat)

ciffergruppe	3-28, 3-61
c-layout	3-59
close_duet	5-3, 5-6, <u>5-13</u> , 5-41
compute	<u>3-34ff</u> , 3-61
compute_line	A-4
compute_opt	A-4
create	3-66
create_opt	A-10
current record	3-66ff
d-array	2-3, 3-10, 5-36
database	3-65
datafejl	3-52, 3-53, 5-15ff
datafejlakt	5-11, 5-41
datafejlkanal	5-11, 5-41
db-beskrivelse	2-2
db-operation	3-65ff
db_opt	A-9
decimaler	3-59
decimals	A-8
delete (db-operation)	3-67
delete (duetblok)	4-14, <u>4-15</u>
delete_opt	A-10
delimiter	3-50, <u>3-51</u>
digitno	A-4
digits	<u>3-28</u> , 3-61
digits_spec	A-4
d_name	A-13
duetabler	1-1, 2-2, 4-1, 5-1
duetaktion	3-57
duetalgol	5-3, 5-4, <u>5-32</u> , 5-34, 5-36

Index (fortsat)

duetareal	5-7, 5-39
duetarray	2-2, 5-7, 5-39
duet_block	A-2
duetblok	2-2, <u>3-8ff</u> , 4-1, 4-14, 5-39
duetdatafejl	5-14, <u>5-15ff</u>
duetdato	5-43
duet_head	A-2
duethoved	<u>3-7</u> , 4-1
duet_instr	A-2
duetinstruktion	2-4, <u>3-3</u>
duetlog	4-21, 5-7, 5-43
duetmaskine	1-1, 1-2, 2-2, 4-1, 5-4
duetnavn	3-3, <u>3-13</u> , 3-17, 3-57
duet_no	A-2
duetoperand	3-4, <u>3-14ff</u>
duetoperation	3-3, <u>3-13</u> , A-3
duetoperator	3-4
duetoversætter	1-1, 2-2, 4-1
duetparam	5-33
duetprg_ident	A-2
duetprogram	2-2, 4-6, A-2
duetprogramfejl	5-14, <u>5-19ff</u>
duetprogramnavn	3-7
duetprogramnr	3-7
duetprogramtekst	4-2
duet_ref	A-3
duetreference	3-17, 3-20
duetregister	2-2, 4-1, 4-6, 4-13ff, 5-7
duetregnavn	5-7, 5-40
duetslut	<u>3-4</u> , 3-20, 3-26, 3-34, 3-36, 3-41, 3-42, 3-56, 3-70

Index (fortsat)

duetsystem	1-1
duetsystemfejl	5-14, <u>5-27ff</u>
duettekster	5-2
duetttext1	5-3, 5-4
duetttext2	5-3, 5-4, 5-5, 5-32
duetttext3	5-3, 5-4, 5-5
duetttextnavn	4-10
duettid	5-43
duetversion	4-13, 5-7, 5-40
eksplicit standardmærke	3-53
eksternt liniernr	4-3
else_action	A-6
else_line	3-37, 3-42, A-5
entry_no	A-2
entrynr	3-9
entrypoint	A-2
entrypunkt	2-2, 3-8
entry_spec	A-3
erklæringsfil	5-2
error_no	A-11
error_type	A-10
execute_line	A-3
executeliste	2-4, 3-3, 3-5, <u>3-20ff</u>
execute_opt	A-3
exit	3-23, 3-44
exit_line	A-11
exit_opt	A-11

Index (fortsat)

fastfortegn	3-59
fejlud	5-7, 5-13, 5-40
fejludskrifter (fra duetmaskinen)	3-71
fejludskrifter (fra oversætteren)	4-6, <u>4-23ff</u>
fejlværdi	5-42
feltflytning	3-66ff
fixed_pos	A-8
fixed_sign	A-8
flytteassociation	3-66ff
for	3-44
foranstillet delimiter (i inddata)	3-52
for_opt	A-6
fp-parameter	4-6ff
fp-parameter nøgleord	4-7
from_value	A-6
get	3-66
getline	3-46
getline_opt	A-6
get_opt	A-10
goto	3-22, 5-34
gåseøjne	3-49
heltalsvariabel	3-15
horisontalspecifikation	3-57

Index (fortsat)

if	3-38
if_opt	A-5
implicit standardmærke	3-54
include	3-24, 4-11
index	A-13
individfelt	3-66
individtype	3-65
informal	A-1
init_duet1	5-3, 5-4, 5-6, <u>5-7</u> , 5-41
init_duet2	3-48, 3-71, 5-3, 5-6, <u>5-9</u> , 5-41
init_duetmaskine	3-54, 3-56, 3-58, 3-71, 5-3, 5-6, <u>5-11</u> , 5-41
initialer	4-10
insert	4-14, <u>4-15</u>
instruktionstal	5-12, 5-40
internt linienr	4-3, 4-23
itype_no	A-10
itype_spec	A-10
jobfil for oversættelse	4-19, 4-20
kanal	3-71, 5-39
kildetekst	4-9, 4-15
kommentar-delimiter	3-49
kommentarfelt	
(i inddata)	3-49
kommentar	
(i duetprogram)	3-5
konstant	3-15

Index (fortsat)

l	5-36
layout	A-7
layout_param	A-8
layoutparameter	3-57, <u>3-58ff</u>
layoutspekifikation	3-57ff
layoutsprængning	5-25
layouttype	3-57, 3-58ff, 3-61, A-8
ldafsnitnr	4-13, 5-7, 5-40
ld-beskrivelse	2-2, 3-66ff
ldfields	5-4
ld_ident	A-2
ld_no	A-2
ldreg	4-13
ld-register	2-2, 4-6, 4-13, 5-7
ldregnavn	5-7, 5-40
ldversion	5-7, 5-40
leftside	A-4
lgd	5-33, 5-34
liniebuffer	3-56
linienr	4-3
linieskift	
(i duetprogram)	3-5
linieskift (i print)	3-58
list	4-11
listning	4-1, 4-2, 4-6
listud	4-11
logudskrift	4-21, 5-7, 5-43
lookup	3-65
lookup_opt	A-10
læsemode	3-50
læsespekifikation	3-50, 3-52, 3-53,
læsevariabel	3-50, 5-15
læsvar_navn	5-15, 5-42

Index (fortsat)

maxdatafejl	5-41, 5-42
maxinstruktion	5-12, 5-21, 5-40
maxkanal	5-7, 5-38
maxprintpos	5-9, 5-38
maxprogramfejl	5-41, 5-42
maxref	5-9, 5-39
maxsystemfejl	5-41, 5-42
modify	<u>3-26ff</u> , 3-61
modify_line	A-4
modify_opt	A-4
modify_symbol	A-4
name	<u>3-30</u> , 3-61, 3-76
namespec	A-4
next	3-66
newduet	4-13
newset	3-66, <u>3-68</u>
newset_opt	A-10
nl	A-13
n-layout	3-59
no	A-8
no_of_chars	A-7
no_of_lines	A-7
no_of_rep	A-8
normalisering	3-16
np	A-13
nulrepræsentation	3-59
nulstilling af num.array	3-32
nulværdiangivelse	3-60
num_const	A-13
numerisk felt	
(i inddata)	3-49
numerisk konstant	3-15, 3-61, 3-76
numerisk trykning	3-59

Index (fortsat)

numerisk udtryk	3-34, 3-61
numerisk variabel	<u>3-15</u> , 3-61, 3-76
numerisk værdi	3-61
num_expr	A-4
num_operand	A-5
num_opt	A-5
numvar	3-15, 3-61, 3-76, A-13
oldduet	4-13
oversættelsesjob	4-19, 4-20
papir	4-12
parameter	A-12
parameter_list	A-12
passivt programpunkt	3-23
pno	A-3
pos 1	A-9
pos 2	A-9
position	3-57, <u>3-58</u> , A-8
prim	5-4
primula	3-56, 5-4
principaler	3-60
principals	A-8
print	<u>3-56ff</u>
print_action	A-7
print_datafejl	5-4, 5-15, 5-42
print_error_line	A-10
printkanal	3-56, 3-70, 3-71
print_line	A7
printlinie	3-57, 4-4
print_opt	A-7

Index (fortsat)

print_result_line	A-10
print_spec	A-7
prinz	3-56, 3-71, 5-9, 5-13, 5-38
programfejlakt	5-11, 5-41
programfejlkanal	5-11, 5-41
programpunkt	3-22, 3-72, 5-14
programtekst	4-2
put	3-6
put_opt	A-10
r	5-36
read	3-46, <u>3-48ff</u>
read_general	3-46, 3-48, 5-4, 5-37
read_mode	A-7
read_opt	A-6
read_spec	A-6
readspecifikation	3-50, 3-52, 3-53
readvar	A-7
readz	3-46, 5-9, 5-37
real arbejdsregister	
(reg)	3-35
recnovar	3-15
redigering af program-	
tekst	4-2ff
reg	5-36
relation	3-38, 3-45, A-5
relopt	A-5
ressourcekrav (for	
oversætteren)	4-18
resultatareal	3-56
resultatudskrift	3-71
resultatvariabel	3-51, 3-55, 3-69
result_channel	A-10

Index (fortsat)

return_line	A-11
right side	A-4
sd_extend_buf	5-7
select	3-24, 3-54, 3-56, <u>3-70ff</u> , 5-42
select exit	3-72
select_line	A-10
select_opt	A-10
select print	3-71
select return	3-72
select stdassign	3-73
select test	3-73
selektiv afgrening	3-42
selektivt assign	3-36
set_no	A-10
set_primula	5-39
set_spec	A-10
shortchar	A-13
sideskift (i duetpro- gramlistning)	4-2
sideskift (i print)	3-58
simpel variabel	3-15
simple_numvar	A-13
simple_textvar	A-13
simple_var	A-13
simple_wordvar	A-13
size	4-16
s-layout	3-64
soda_dbms	1-1, 2-2, 3-65, 5-35
sodald	1-1, 2-2, 5-2
sodatext1, -2, -3	5-4
spill	3-16
standardlayout	3-64

Index (fortsat)

standardmærke	3-53, 3-70, 3-73
standardværdi	3-48, 3-53, 3-66, 3-73
stdassign	3-54, 5-12, 5-38
std_assign_line	A-11
std_layout	A-9
std_layoutype	A-9
std_line	A-9
strengslut	3-46, 3-49
strengslut-delimiter	3-51
styreprogram	1-1, 1-2, 2-2, 3-11, 3-46, 3-48, 3-54, 3-56, 3-74, <u>5-1ff</u> , 5-32
syntaksbeskrivelse	A-1
sysdoklinienr	4-3
sysdokregister	4-2, 4-6, 4-9, 4-13
systemfejlakt	5-11, 5-41
systemfejlkanal	5-11, 5-41
sæt	3-65ff
sættype	3-67, 3-68
t	5-36
take value	5-33, 5-36
tegnfelt (i inddata)	3-49, 3-51
tegnklassetabel	3-48, <u>3-49</u> , 5-9, 5-38
tegnkonstant	3-17, 3-61, 3-76
tegntrykning	3-59
tekstdelimiter	3-49, 3-51
tekstfelt (i inddata)	3-49
tekstfil (programtekst)	4-2, 4-6, 4-10
tekstkonstant	3-17, 3-61, 3-76
teksttrykning	3-58
tekstvariabel (textvar)	3-15, 3-51, 3-61, 3-76

Index (fortsat)

teleop	1-2, 5-1
telescop	1-2, 5-1
term_code	A-7
terminatorkode	3-50, <u>3-51</u>
testassign_symbol	A-11
testbit	3-74, A-11
testkanal	3-71, 5-10, 5-42
test_line	A-11
testudskrift (fra duet- maskinen)	3-70, 3-71, 5-42
testudskrift (fra over- sætteren)	4-16
test_value	A-5, A-6, A-11
testvar	A-11
testvariabel (assign/ action)	3-36, 3-40, 3-42
testvariabel (testud- skrifter)	3-70, 3-73, 5-42
testværdi (assign/ action)	3-37, 3-43, 3-61
textchar	A-13
textconst	A-13
textlength	A-8
textvar	3-15, A-13
tflytproc	5-4
t-layout	3-58
tno	A-3
to_value	A-6
translate	4-14, <u>4-15</u>
trimmepunkt	2-3
trykkekanal	3-56, 3-70, 3-71
type	5-33, 5-34, 5-36

Index (fortsat)

udførelse af duet-	
program	3-5
udvidelsesprocent	4-16
undertrykkelses-	
specifikation	3-24, 4-11
user	3-9, 4-10
value	A-5
value_spec	A-9
var	3-30, 3-31
vardecl	5-3
variabel	2-3, 3-14, 3-66, A-13
variabelnavn	3-14, 3-30, 3-61, 3-76
variabelnr	3-14
variabelreference	3-14
variabeltype	3-15
variable	A-13
varref	A-13
var_spec	A-4
vartext	A-9
version	4-9
vertical_spec	A-7
vertikalspecifikation	3-57, <u>3-58</u>
vertikaltabulering	3-58
værdi	5-36
værdielement	<u>3-18</u> , 3-27, 3-61
værdigrænsekontrol	3-48, <u>3-53</u>
værdispecifikation	3-57, <u>3-61</u>
værdispektrum	3-53

Index (fortsat)

w	5-36
while	3-45
while_opt	A-6
wordvar	3-15
write (mellem print)	5-39
zero_repr	A-8
zero_value	A-8