

Arbeidsnotater

T A T I S T I S K S E N T R A L B Y R Å

IB 65/3

Oslo, 26. mars 1965

Preprocessor for desisjonstabeller ^{x)}

av

Thor Aastorp

I n n h o l d

1. Generell beskrivelse
2. Regler og begrensninger
3. Kontrollkortene
 - 3.1 Problemkort
 - 3.2 Tabellkort
 - 3.3 Spesifikasjonskort
 - 3.3.1 Egne Fortran-statements og -rutiner
 - 3.3.2 Testrelasjoner og desisjonsbetingelser
 - 3.3.3 Aksjoner og aksjonsselektorer
 - 3.4 Variable format-kort
4. Logical unit numbers

x) Eksempler på bruk av desisjonstabeller for spesifisering av revisjonskriterier er gitt i ANO IB 65/2: The Efficiency of Automatic Detection and Correction of Errors in Individual Observations as Compared with Other Means for Improving the Quality of Statistics, by Svein Nordbotten.

1. Generell beskrivelse

Preprocessoren er skrevet i Fortran for CDC 3 600. På grunnlag av kontrollkort som spesifiserer desisjonstabellen, vil preprocessoren generere følgende program og rutiner (alle i Fortran):

- a) Et hoved- eller rammeprogram som ved kjøring først sørger for å lese inn kontrollkort inneholdende formatspesifikasjon av input-data. På grunnlag av dette format blir de enkelte input-records lest inn. Etter at en input-record er lest inn, vil hovedprogrammet påkalle subrutine for de enkelte desisjonstabellene. Subrutinen for desisjonstabell nr. 1 blir påkalt og utført først, deretter blir de andre subrutinene påkalt i tabellnummer rekkefølge. Denne rekkefølgen er imidlertid ikke absolutt, men kan endres ved spesifisering.
- b) En fast subrutine som påkalles av subrutinene for de enkelte desisjonstabellene etter at de spesifiserte relasjoner er testet. Resultatet av disse testene blir i subrutinen sammenliknet med de spesifiserte desisjonsregler for å finne den første regel som blir tilfredsstilt.
- c) Der genereres en subrutine for hver desisjonstabell. En slik subrutine vil bestå av eventuelle egne Fortran-statements og -rutiner, testing av de spesifiserte relasjoner og rutiner med de oppgitte aksjoner.

2. Regler og begrensninger

Brukeren av denne preprocessor må være oppmersom på de begrensninger som gjelder og de regler man må følge ved spesifiseringen av sine desisjonstabeller. Nedenfor er listet de viktigste punktene; konvensjonene for utfylling av kontrollkortene vil bli gjennomgått i detalj i neste kapitel.

- a) Man kan spesifisere inntil 10 tabeller.
- b) Innen hver desisjonstabell kan man ha inntil 94 testrelasjoner, 32 desisjonsregler og 99 aksjoner.
- c) Man kan bruke inntil 999 spesifikasjonskort for hver tabell.
- d) Det er full anledning til (i visse tilfelle endog påkrevd) å spesifisere egne Fortran-statements og -rutiner.
- e) Antall variable format-kort (som angir formatet på input-data) kan være inntil 5 kort.
- f) Man kan lese inn maksimalt 99 variable (felter) for hver input-record. Variable blir lest inn i en real array Y med dimensjonen 100. Dette betyr at alle variable må leses inn etter F-format og at all referanse blir Y (1), Y (2), osv.

3. Kontrollkortene

3.1 Problemkort

Som første kontrollkort må man ha et såkalt problemkort med disse opplysningene:

- Kol. 1 - 2: PR (for problemkort)
- " 3 - 10: Problemnavn (samme regel som for Fortran program navn)
- " 11 - 12: Antall desisjonstabeller, t ($t \leq 10$)
- " 13 - 14: Antall variable innen hver record, Y ($Y < 100$)
- " 15 - 16: Antall variable format-kort, k ($k \leq 5$).

3.2 Tabellkort

Foran spesifikasjonskortene for en desisjonstabell må man ha et tabellkort med disse opplysningene:

- Kol. 1 - 3: TAB (for tabellkort)
- " 4 - 5: Tabellnummer, n ($n \leq 10$)
- " 6 - 8: Antall spesifikasjonskort for tabellen, m ($m < 1\ 000$).
Tabellkortet er ikke medregnet i dette antallet.
- " 9 - 10: Antall testrelasjoner, r (≤ 94)
- " 11 - 12: Antall aksjoner, a ($a \leq 99$).

3.3 Spesifikasjonskort

Spesifikasjonskortene innen en desisjonstabell kan deles inn i tre grupper:

- i) Egne Fortran-statement og -rutiner
- ii) Testrelasjoner og desisjonsbetingelser
- iii) Aksjoner og aksjonsselektorer.

Spesifikasjonskortene har følgende format:

- Kol. 1 - 5: Statement-nr.
- " 6 : Continuation code
- " 7 - 48: Statement
- " 49 - 80: Desisjonsbetingelser eller aksjonsselektorer.

Under spesifikasjonen må man være oppmerksom på og ta hensyn til følgende:

- i) Feltene fra input-records blir lest inn i en real array Y. av dimensjonen 100. Referanse til en variabel innen input-recorden blir da Y (i) hvor i angir feltets plass innen recorden; Eksempel: Y (2), Y (6), osv. Arrayen Y inngår i en common-blokk som er felles for hovedprogrammet og subrutinene.

- ii) I spesifikasjonene bør man unngå å benevne common-blokker med navn som begynner med LLL. Videre bør man unngå å spesifisere egne variable med navn som TAB, Y, N1, N2, N3, osv. og navn som begynner med LLL eller LIST.
- iii) Man kan ikke operere med statement-numrene 99 900 - 99 999.

3.3.1 Egne Fortran-statements og -rutiner

De eneste tilfellene hvor det kreves spesifikasjoner i denne gruppen, er når man i en eller flere relasjoner har referert til lister (kodelister o.l.). Disse listene som skal ha et navn begynnende med LIST, må defineres og spesifiseres ved hjelp av Fortran-statement. Listen må være lagret som en én-dimensjonal array, dvs. at man må oppgi størrelsen på listen i et dimensjon-statement, f.eks. "Dimension LISTA (140)". Videre må man definere en integer variable Ni, hvor i er et nummer som angir i hvilken rekkefølge de enkelte listene blir referert til i relasjonene. Hvis LISTA er den andre listen som det blir referert til i relasjonene, må man definere N2 ved "Data (N2=140)" eller "N2=140". Det må derfor være like mange Ni som det er relasjoner med LIST i. På tilsvarende måte må man definere de enkelte elementene i listen, enten ved data-statements eller ved aritmetiske replacement-statements. Hvis den samme listen blir referert til i flere relasjoner i samme tabell, må man definere like mange Ni's, Eks.: LISTA blir referert til både som den andre og fjerde listen, en må da sette "N2=140" og "N4=140" (eventuelt "N4=N2=140").

I denne gruppen av spesifikasjonskort kan man spesifisere beregninger av ulike slag, innlesing av kontrolldata osv. Hvis man innfører variable som skal være felles for flere desisjonstabeller, må disse spesifiseres i common-blokker. Man må da være oppmerksom på at det bare er elementer i såkalte "labelled common" som kan defineres i data-statements.

Under skriving av statements i denne gruppen kan kol. 49-80 ikke brukes. Det betyr at man arbeider med et forkortet Fortran-kortformat. Continuation kan brukes på vanlig måte.

3.3.2 Testrelasjoner og desisjonsbetingelser

Under spesifiseringen i denne gruppen nyttes følgende format:

```
Kol. 1 - 5: Statement nr.
    " 6   : Continuation
    " 7 - 48: Relasjoner
    " 49 - 80: Desisjonsbetingelser.
```

De relasjonene som man av en eller annen grunn ønsker å referere til, enten i egne Fortran-statements eller i aksjonsdelen, må ha statement nr. Continuation-kort kan man bruke på vanlig måte.

En enkel relasjon skriver man slik: P op Q, hvor P og Q er aritmetiske uttrykk og "op" er en av disse relasjonskodene:

.EQ.	som betyr	Lik (=)
.NE.	" "	Forskjellig (\neq)
.GT.	" "	Større enn (>)
.GE.	" "	Større eller lik (\geq)
.LT.	" "	Mindre enn (<)
.LE.	" "	Mindre eller lik (\leq)

Resultatet av en test av en relasjon kan bli enten "yes" eller "no" avhengig av hvorvidt forholdet mellom de to aritmetiske uttrykkene tilfredsstiller den betingelse som relasjonskoden angir. Eksempel: Testresultatet av relasjonen "Y(2).GT.100." blir "yes" når verdien av Y(2) er større enn 100. Er verdien av Y(2) lik eller mindre enn 100, blir resultatet "no".

En testrelasjon kan bestå av flere enkle relasjoner. Relasjonene bindes da sammen med de logiske operasjonskodene:

.AND.
.OR.

Har man de to enkle relasjonene R1 og R2, kan man ha:

- i) R1.AND.R2; testresultatet av denne sammensatte relasjonen blir "yes" bare når resultatene av både R1 og R2 er "yes".
- ii) R1.OR.R2; her blir testresultatet "yes" når resultatet av R1 og/eller R2 er "yes".

Man kan også spesifisere testrelasjoner som består av flere enkle relasjoner, uten å binde dem sammen med .AND. eller .OR. Hver enkelt relasjon må da begynne på ny linje og etterfølge hverandre. Preprocessor vil behandle disse relasjonene på .OR.-basis.

I en relasjon er det anledning til å referere til hele lister (kode-lister o.l.). Disse listene må da defineres i egne Fortran-statements (punkt 3.3.1). En slik liste må ha et navn som begynner på LIST og i relasjonen må dette navnet med subscript I stå først; Eks.: LIST Nr 1 (I).EQ.Y(1) (Det som er understreket er obligatorisk i en slik relasjon). Man kan ikke referere til flere lister i en og samme testrelasjon.

I den første linjen innen hver testrelasjon spesifiserer man desisjonsbetingelsene (kol. 49-80). Man bruker her bare Y (for yes), N (for No.) eller blank.

For hver desisjonstabell kan man maksimalt ha 94 testrelasjoner og 32 desisjonsregler.

3.3.3 Aksjoner og aksjonsselektorer

Man kan ha inntil 99 aksjoner innen hver desisjonstabell. Hver aksjon spesifiseres ved hjelp av Fortran-statements. I det første kortet i en aksjon angir man dessuten aksjonsselektorene i kol. 49-80. Første statement i en aksjon kan ikke være format-statement. Videre kan ikke en aksjon avsluttes med IF-statement (med unntak av logiske IF-statements med én utgang).

Som nevnt tidligere vil subrutinene for de enkelte desisjonstabellene normalt bli påkalt og utført i nummerrekkefølge. Man har imidlertid anledning til å endre denne rekkefølgen ved spesifisering. Hvis man i aksjonsdelen i en desisjonstabell har et statement av typen "TAB=M", vil hovedprogrammet etter at aksjonen med dette statement er utført, påkalle subrutinen for desisjonstabell Nr. M. Dette er under forutsetning av at desisjonstabell Nr. M forekommer, hvis ikke blir neste input-record innlest og subrutinen for tabell nr. 1 påkalt. Eksempel: I desisjonstabell nr. 3 inneholder en aksjon dette statement: "TAB=6". Når denne aksjonen er fullført, blir desisjonstabell nr. 6 påkalt istedenfor tabell nr. 4. En kan imidlertid ikke nytte TAB=m i tabell m.

3.4 Variable format-kort

Dette eller disse kontrollkortene leses av rammeprogrammet mens de kontrollkortene som er nevnt foran, leses av preprocessoren.

Man må ha med så mange format-kort som angitt på problemkortet. Alle 80 kolonnene i kortet kan brukes. Vær oppmerksom på at alle variable må leses inn ved F-format.

4. Logical unit numbers

- a) Logical unit 1: Herfra leses kontrollkortene
- b) " " 2: Feilutskrifter
- c) " " 3: Liste over desisjonstabellene
- d) " " 4: På denne LU som må være magnetbånd genereres rammeprogram med subrutiner
- e) " " 5: Mellomlagringsbånd, kan ikke være scratch
- f) " " 6: Input-data.