



**ARTIKLER**

**13**



**DESIJONSTABELLER  
OG GENERERING AV MASKIN-  
PROGRAMMER FOR GRANSKNING AV  
STATISTISK PRIMÆRMATERIALE**

Av Svein Nordbotten

**DECISION TABLES AND GENERATION  
OF COMPUTER PROGRAMS  
FOR EDITING OF STATISTICAL DATA**

OSLO 1965

**STATISTISK SENTRALBYRÅ**

# Desisjonstabeller og generering av maskin- programmer for granskning av statistisk primærmateriale<sup>1</sup>

*av underdirektør Svein Nordbotten*

<sup>1</sup> Artikkelen ble delvis utarbeidd under opphold i USA 1963/64 med stipend fra Rockefeller Foundation.

## 1. Granskning av statistisk primærmateriale

Verdien av statistikernes arbeid er i høy grad avhengig av statistikkproduktenes nøyaktighet, dvs. at statistikkproduktene gir et godt bilde av de situasjoner de skal belyse. I motsetning til tidligere aksepterer statistikerne nå at statistikkproduktene alltid vil kunne være påvirket av feil og at målet ikke er å eliminere alle feil, men å søke å begrense dem slik at virkningen ikke gjør statistikkproduktene verdiløse. I videste forstand har nøyaktighetsbegrepet to sider. Den første er relasjonen mellom de begrep statistikkbrukeren mener han har behov for og det begrep statistikeren nytter når han planlegger sin undersøkelse. Forskjellen mellom disse begrep skyldes oftest mangel på forståelse eller kommunikasjon mellom statistikkbrukeren og statistikeren. Den annen side, som er relevant her, er forholdet mellom det faktiske måleresultat som foreligger og det resultat statistikeren ville fått om han kunne gjennomført målingen under ideelle forhold og i samsvar med definisjonen av de begrep han vil måle.

Statistikeren som vil planlegge gjennomføring av en statistisk undersøkelse

må ta standpunkt til hvor ofte undersøkelsen skal foretas, hvor detaljerte resultater skal gis, hvor nøyaktige resultatene bør være og hvor fort de skal foreligge. Med begrensede ressurser er disse avgjørelser ikke uavhengige og vil alle i større eller mindre grad reflekteres i resultatenes totale kvalitet. Med dette i tankene skal vi her begrense oss til nøyaktigheten.

Om det på en eller annen måte lot seg gjøre å få resultater av tilfredsstillende nøyaktighet, ville vi som regel være uinteressert i primærmaterialets eller de enkelte observasjoners nøyaktighet. Dessverre har vi imidlertid sjelden mulighet for å sikre oss at resultatene er tilfredsstillende uten å gå veien om granskning av primærmaterialet. I enkelte situasjoner kan forholdene ligge til rette for granskning etter statistiske kvalitetskontrollprinsipper og av utvalg fra primærmaterialet. I de fleste situasjoner vil imidlertid primærmaterialet bli så intensivt bearbeidet at bare fullstendig granskning vil gi tilfredsstillende garanti for brukbar nøyaktighet.

Etter hvert som både tallet på nye statistiske undersøkelser og massenes

størrelse øker blir det stadig vanskeligere å opprettholde den samme individuelle granskning av grunnmaterialet og det har vist seg at elektronisk databehandling skulle bli det hjelpemiddel som har gjort det mulig å opprettholde og tildels øke nøyaktigheten i statistiske produkter.

La oss se litt nærmere på hva det er som særpreger EDB som verktøy for granskning av primærstatistikk. Først og fremst er det EDB-systemets evne til å foreta logiske tester på et materiale etter en oppsatt instruks. Granskning av statistisk grunnmateriale består nettopp i å teste forskjellige relasjoner mellom de enkelte kjennemerker i hver observasjon. Den menneskelige kontrollør sammenligner de forskjellige opplysninger for å se om de svarer til det generelle bildet statistikeren gjennom erfaring har dannet seg, og dette søker vi også å gjøre ved hjelp av EDB-maskinene ved å utnytte deres evne til å foreta tester.

Mens den menneskelige gransker kan arbeide på grunnlag av en enkelt skriftlig eller til og med muntlig instruks fordi han er istand til å resonnerer og nytte sin personlige lærdom, erfaring og dømmekraft, må maskinene ha sin instruks nedskrevet i detalj og fullstendig entydig. Etter som det mulige antall situasjoner som kan tenkes oppstå under kontrollen er uhyre stort vil det være ugjørlig, eller i hvert fall upraktisk, å lage maskinprogrammer som tok korrekt vare på hver enkelt situasjon. En må søke å la maskinen granske de observasjonstyper som synes å kunne ha størst virkning på resultatene og håpe at de andre ikke er av vesentlig betydning. Her er åpenbart den manuelle granskning av primær materialet foreløpig overlegen sam-

menlignet med EDB. På den annen side har manuell kontroll en tendens til å bli subjektiv, og en kan observere systematiske forskjeller i resultatene fra forskjellige granskere. EDB bidrar derfor til konsekvent granskning og til oppfyllelse av kravet om at det må være en kjent og spesifisert sammenheng mellom statistiske observasjoner og resultater. EDB-systemene har hittil bare unntaksvis vært i stand til å kunne arbeide med de originale oppgaver. I de fleste tilfeller skjer det en overføring til f. eks. hullkort. Dette har to konsekvenser. EDB-granskning har for det første ikke det samme omfang av informasjon å arbeide med som manuell granskning. I manuell granskning vil oppgavegiverens håndskrift og overstrykninger etc. representere informasjon som ikke er tilgjengelig for EDB-granskning. For det annet vil overføringsprosessen selv også kunne forvrengte opplysninger og derfor kunne innføre feil som ikke oppstår i manuell granskning.

EDB-systemene kjennetegnes også ved sin store lagerkapasitet som tillater at granskningene kan gjøres simultane, dvs. langt flere faktorer kan samtidig sees i sammenheng. Mulighetene for å kunne se flere sider av et observasjonssett under ett synes å være et viktig steg i retning av realistisk granskning.

Det er imidlertid EDB-granskningens store hastighet som gjør det mulig å opprettholde og til dels øke nøyaktighetskravene samtidig som observasjonsmengden øker, og uten at det betyr redusert aktualitet eller detaljrikdom. Operasjonshastigheten har hatt den aller største betydning for granskningen med primærstatistikken. Uten denne hastighet ville vi etter hvert ha blitt tvunget

til å gi avkall både på aktualitet og detaljrikdom.

## 2. Desisjonstabeller for spesifikasjon og dokumentasjon

Det var anvendelsen av EDB som for alvor tvang statistikerne til å presisere sine granskningsproblemer. I den første tiden gikk dette ofte for seg på den måten at de som arbeidde for anvendelse av EDB studerte hva som ble gjort manuelt og satte opp en forsøksvis spesifikasjon. Når denne så ble forevist fagspesialisten, var ofte kommentarene disse: „Dette gjør vi nok av og til men det er også mange andre ting som må tas i betraktning“. Senere ble fagspesialistene mer aktive og skrev lange utredninger om alle forhold som skulle kontrolleres, og disse var ofte vanskelig for maskinspesialistene å tilegne seg. Skulle en virkelig ha klar sammenheng mellom observasjon og resultat var problemet å finne en mer kompakt og oversiktlig måte å spesifisere kontrollen på. En løsning som tilfredsstillte kravene til dokumentasjon og som nå prøves i amerikansk statistikkproduksjon er bruk av såkalte logiske desisjonstabeller for å beskrive kontroll- og korreksjonsmetoder.

Desisjonstabeller har vært brukt på forskjellige logiske problemer i mange år og har også vært antydnet som et mulig verktøy i statistikkproduksjonen tidligere, men det er først nå at de for alvor synes å være tatt i bruk. En desisjonstabell består som alle tabeller av et tabellhode, en forspalte og en tabellmatrise (se figur 1). Fordi en desisjonstabell i virkeligheten er to tabeller med et felles tabellhode satt sammen til en tabell, vil det alltid være en horisontal linje som

Fig. 1. Oppbygning av en desisjonstabell

Utsagn	Desisjonsregler
Testrelasjoner	Desisjonsbetingelser
Aksjoner	Aksjonsselektorer

deler tabellen i en desisjonsdel og en aksjonsdel. I desisjonsdelens forspalte skrives de forskjellige relasjoner som skal kontrolleres, og i forspalten for den etterfølgende aksjonsdel listes de forskjellige aksjoner som kan komme på tale. De enkelte kolonner i tabellmatrisen betegnes som desisjonsregler. Den del av desisjonsregelen som hører til tabellens øvre del kalles betingelsen mens den annen del utgjør aksjonsselektoren. Det mulige resultat av hver enkelt relasjon er når den prøves på en statistisk observasjon: „godta“ eller „forkast“, eller kortere: „ja“ eller „nei“. Desisjonsbetingelsen representerer den kombinasjonen av  $J$ - og  $N$ -er som kreves for regelen mens aksjonsselektoren inneholder  $X$ -er for markering av de aksjoner som ønskes når desisjonsbetingelsen er oppfylt.

I sin enkleste form er de eneste tillatte tegn i tabellmatrisens øvre del derfor  $J$  eller  $N$ , og tilsvarende i den nedre del tillates bare symbolene  $X$  eller blankt. Med  $M$  testrelasjoner vil vi ha  $2^M$  kolonner eller regler maksimalt. Dette krever både unødig brede tabeller og stor innsats å fylle ut fordi flere betingelser vanligvis fører til samme aksjonsseleksjon. Dette er løst ved at det også tillates blanke ruter i den øvre del av tabellen. Blank betyr her at det er uten betydning om testresultatene er  $J$  eller  $N$  av vedkommende test såfremt de andre test-

Fig. 2. Eksempel på desisjonstabell

	D 1	D 2	D 3	D 4	D 5	D 6	D 7
T 1 $P > 0$ .....	J	J	J	J	J	N	N
T 2 $A > 0$ .....	J	J	J	J	N	J	N
T 3 $P/A \geq G_1$ .....	J	J	N	N			
T 4 $P/A < G_2$ .....	J	N	J	N			
A 1 $P + \Sigma P \rightarrow \Sigma P$ .....	X						
A 2 $A + \Sigma A \rightarrow \Sigma A$ .....	X						
A 3 $(\Sigma P/\Sigma A) \cdot A \rightarrow P$ .....		X				X	
A 4 $(\Sigma A/\Sigma P) \cdot P \rightarrow A$ .....			X		X		
A 5 Manuell behandling .....							X
A 6 Stopp og revider grensene .....				X			
A 7 Fortsett med neste obs. ....	X	X	X		X	X	X

resultatene er i samsvar med desisjonsbetingelsen. En annen videreutbygging er å utstyre tabellen med en ekstra llers-kolonne med blank betingelsesd eom angir hvilke aksjoner som skal tas dersom test-resultatene av et observasjonssett ikke kan parres med noen av se foregående kolonner.

La oss ta for oss et forenklet produksjonsstatistikk eksempel hvor vi vil granske produksjon,  $P$ , og arbeidsinnsats,  $A$ . Figur 2 viser hvordan dette problemet kanskje vil bli spesifisert. På grunn av tidligere erfaring utformes relasjonene  $T1$ — $T4$  og grensene  $G_1$  og  $G_2$  fastsettes. Med fire relasjoner vil vi kunne få opp til 16 uavhengige betingelser eller betingelseskolonner men vi ser at disse lar seg redusere til 7 fordi utfallet  $T3$  og  $T4$  er uten betydning dersom  $T1$  og  $T2$  eller begge gir  $N$ .

Aksjonene forutsetter granskning med automatisk korreksjon når enten en av kjennemerkeverdiene mangler eller når forholdet mellom dem er utenfor toleranseområdet. Desisjonsregelen  $D2$  gjør bruk av forutsetningen om at når for-

holdet ligger på oversiden av toleranseområdet er det sannsynligvis  $P$  som er uriktig mens regelen  $D3$  sier at om det ligger på undersiden må  $A$  korrigeres. Aksjonene  $A1$ — $A7$  forteller hva som skøl foretas i de enkelte tilfeller, f. eks. at manuell behandling ansees nødvendig når begge kjennemerkeverdier mangler.

Det finnes også mer avanserte typer av logiske desisjonstabeller. Den type som er beskrevet her er en tabell med såkalt begrenset inngang. Tabeller som tillater flere enn 2 resultater ved testing av en enkelt relasjon betegnes gjerne som tabeller med utvidet inngang. En tredje type tabeller tillater relasjoner som referer seg til andre typer av akseptanseområder enn de rent lineære, f. eks. en kodeliste.

Desisjonstabeller har i amerikansk statistikkproduksjon funnet anvendelse på flere felter.<sup>1</sup> Som eksempel kan nevnes at kontroll og korreksjon av den måned-

<sup>1</sup> James L. O'Brien: Some Promising Approaches to Computerizing Administrative Operations, Presentation to the 1964 Convention of the American Society Quality Control, Buffalo, N. Y., May 4—6, 1964, Mimeographed.

lige amerikanske handelsstatistikken er beskrevet ved hjelp av desisjonstabeller. Et annet eksempel er Jordbrukstillingen 1964. I forbindelse med denne telling er det utarbeidd nærmere 1000 desisjonstabeller som inneholder kontroll- og korreksjonsspesifikasjoner for de cirka 400 forskjellige kjennemerker det skal innhentes opplysninger om. Det hevdes her at det allerede under oppstilling av tabellene ble oppnådd meget i retning av å unngå inkonsistente kriterier. Erfaringene tyder på at tabellene vanligvis ikke bør være større enn 6—8 betingelser, og ikke inneholde mer enn 20—25 regler, dersom de skal være oversiktlige og noenlunde lett forståelige. Det synes ikke i praktiske anvendelser å by på særlige problemer å dele opp en større tabell i flere mindre. Tabellene stilles gjerne opp i ordnet form slik at alle  $J$ -er i første linje kommer først og deretter alle  $N$ -er. I annen linje under  $J$ -ene kommer igjen alle  $J$ -er først og deretter alle  $N$ -er osv., like ens blir det under  $N$ -ene i første linje. Tabellen i figur 2 er et eksempel på en ordnet tabell.

I ordnet form er det lett å kontrollere at tabellen ikke inneholder flere like desisjonsregler eller at noen muligheter er uteglemt. Med hensyn til rekkefølgen av relasjonene settes de vanligvis opp i den orden de ønskes gjennomført men den er i prinsippet likegyldig. Aksjonene må imidlertid settes opp i den rekkefølge de skal utføres så lenge som de velges ut i samme desisjonskolonne.

### 3. Automatisk generering av kontrollprogrammer

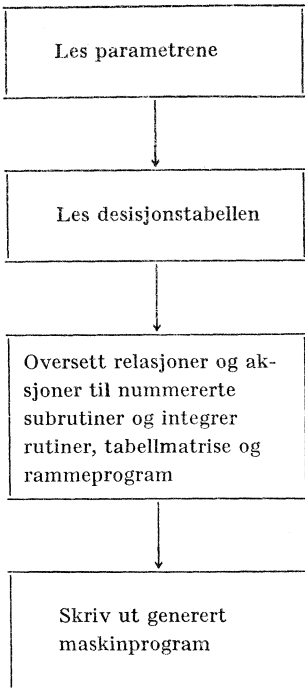
Mens desisjonstabellene synes å være et utmerket middel til spesifikasjon av

granskningsmetoder og for senere dokumentasjon gjenstår fortsatt problemet å få omsatt tabellenes innhold til EDB-programmer. De EDB-orienterte statistikere har lenge drøftet mulighetene for å lage såkalte generelle programmer for kontroll og korreksjon på samme måten som vi har generelle sorterings- og tabelleringsprogrammer.

Desisjonstabellene representerer en standard måte å uttrykke granskningsmetoder på som sterkt inviterer til arbeid med generelle programmer. Det eksisterer nå generatorprogrammer som automatisk omsetter punchede desisjonstabeller til komplette produksjonsprogrammer. De eksisterende generatorprogrammer er ikke alle slik at de høver for de utsagn som vil forekomme i en desisjonstabell for statistisk granskning og det arbeides derfor med generatorprogrammer spesialkonstruert for statistikeren. Det nyttes forskjellige algoritmer som grunnlag for arbeidet alt etter som de ferdige programmer ønskes plassbesparende eller raske. Vi skal her beskrive to metoder:

*Metode A:* Som alle generatorsystemer er det 2 prosesser i maskinen en må holde adskilt. Den første er oppbygning eller generering av det egentlige granskningsprogram mens den annen prosess er bruk av dette program på det statistiske materiale. Oppbygningen utføres av den egentlige genereringsrutine som gjør bruk av en permanent programramme og data i form av punchede desisjonstabeller. (Se figur 3.) Dessuten krever genereringen visse parametre som f. eks. tallet på testrelasjoner:  $M$ , aksjoner:  $A$  og regler:  $R$ . Relasjonene og aksjonene må være skrevet i en form som generatortrutinen kan forstå, f. eks.

Fig. 3. Genereringsrutine



i ALGOL, COBOL eller FORTRAN. Det som i grove trekk skjer når en tabell blir lest inn er at forspalte og tabellmatrise blir skilt. Hver enkelt linje i forspalten blir omsatt til maskinspråk og oppbevart i form av nummererte subrutiner. Programrammen, subrutiner og desisjonskolonnene blir deretter integrert til et komplett program for kontroll og korreksjon som kan brukes senere på det statistiske grunnmateriale.

Dette program kan virke som diagrammet i figur 4 viser. Det leser først inn sine parametre, f. eks. tallet på observasjonssett,  $N$ , verdiene for grensene  $G_1$  og  $G_2$  samt initialverdier  $P_0$  og  $A_0$  for  $P$  og  $A$ . Et observasjonssett leses inn om gangen. Vi forutsetter en binærmaskin med logiske operasjoner i sitt ordrepertoir og antar for enkelthets

skyld at tallet på binærsiffer pr. ord er større enn tallet på testrelasjoner.

La oss først se på lagring av desisjonstabellen. Hver kolonnes betingelsesdel får to binærord  $B^1$  og  $B^2$ . I første binærord  $B^1$  representeres  $J = 0$ ,  $B = 0$  og  $N = 1$  slik at det til hvert betingelseselement svarer en bit. For å kunne skille mellom  $J$  og  $B$  nyttes annet binærord  $B^2$  hvor en  $B = 0$  mens  $J$  og  $N$  begge er 1. Aksjonsselektoren vil kreve bare et ord  $B^3$  pr. desisjonsregel dersom tallet på aksjoner ikke overstiger tallet på bits i ordet. Figur 5 viser hvordan tabellen i figur 2 vil bli lagret.

Det prikkede felt  $A$  i fig. 4 viser desisjonsrutinen. Her blir først observasjonen testet mot alle relasjoner. Utfallet av test  $i$  blir satt lik 0 eller 1 og satt inn i posisjon  $i$  i et resultatord  $T$ . Når alle  $M$  tester av observasjonen er utført, utføres en såkalt ikke-ekvivalens operasjon ( $\neq$ ) på  $T$  og første binærord  $B^1$  i en desisjonskolonne. Resultatet  $T^1$  av denne operasjon er et binærord med 1 i alle posisjoner hvor  $T$  og  $B^1$  har ulike bits. Med andre ord, dersom  $T^1 = 0$  må vi ha funnet den søkte desisjonskolonne. På den annen side, når  $T^1 \neq 0$  er det to alternativer. De posisjoner som er ulike kan svare til  $N$ -elementer i  $T$  og blanke i betingelsen og da har vi også funnet den søkte desisjonskolonne. Det andre alternativ er at vi ikke arbeider med den kolonne vi søker og må starte på den neste. For å undersøke hvilket av de to alternativ som er det riktige foretas en logisk multiplikasjon ( $\&$ ) av  $T^1$  og det annet binærord  $B^2$  som gir resultat  $T^2$ . I  $T^2$  vil vi bare få 1 dersom begge de to operander har 1 i tilsvarende binærposisjon. Dersom  $T^2 = 0$  må vi derfor ha funnet den søkte regel.



Fig. 4. Rammeprogram

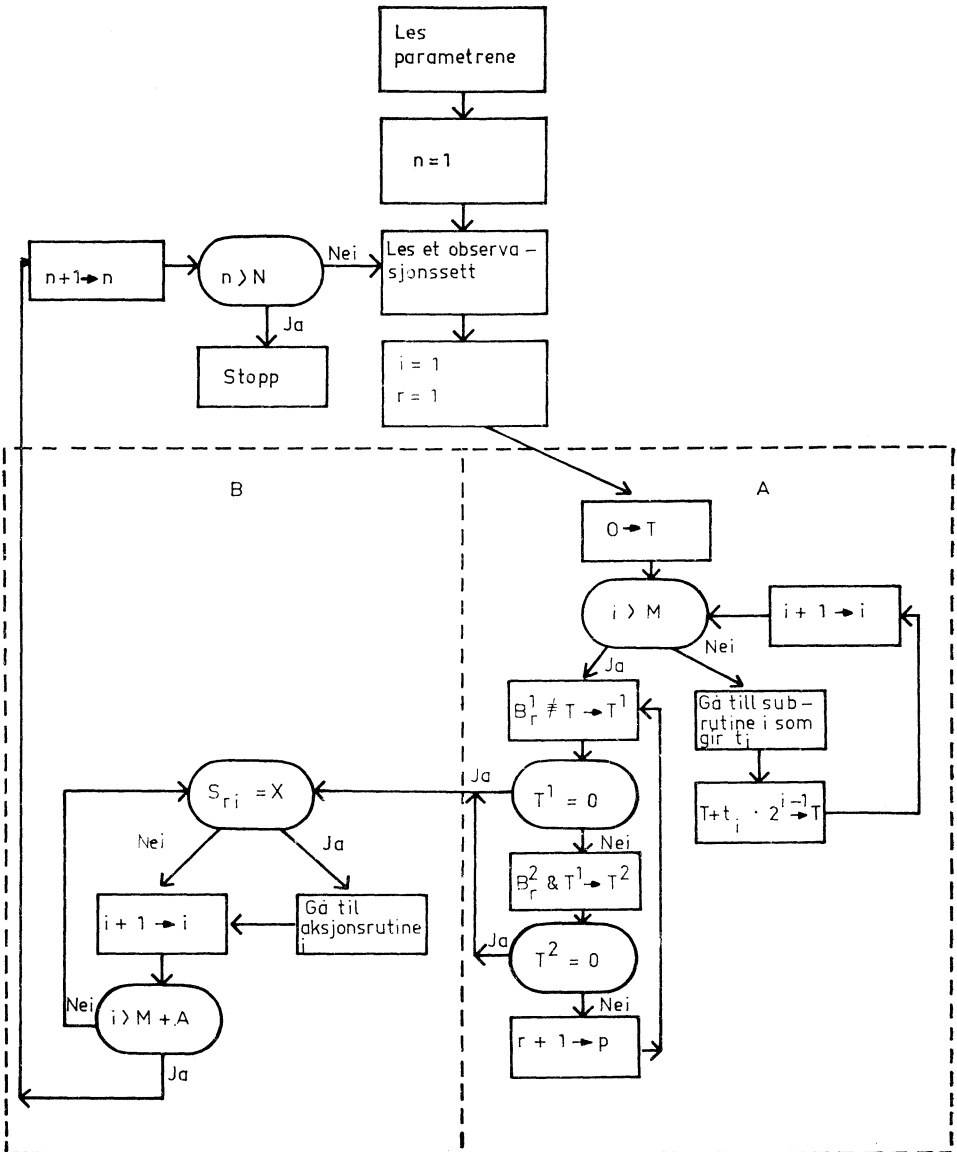


Fig. 5. Desisjonstabell generert etter metode A

	D 1		D 2		D 3		D 4		D 5		D 6		D 7	
	$B_1^1$	$B_1^2$	$B_2^1$	$B_2^2$	$B_3^1$	$B_3^2$	$B_4^1$	$B_4^2$	$B_5^1$	$B_5^2$	$B_6^1$	$B_6^2$	$B_7^1$	$B_7^2$
T 1	0	1	0	1	0	1	0	1	0	1	1	1	1	1
T 2	0	1	0	1	0	1	0	1	1	1	0	1	1	1
T 3	0	1	0	1	1	1	1	1	0	0	0	0	0	0
T 4	0	1	1	1	0	1	1	1	0	0	0	0	0	0
	$B_1^3$		$B_2^3$		$B_3^3$		$B_4^3$		$B_5^3$		$B_6^3$		$B_7^3$	
A 1	1		0		0		0		0		0		0	
A 2	1				0		0		0		0		0	
A 3	0		1		0		0		0		1		0	
A 4	0		0		1		0		1		0		0	
A 5	0		0		0		0		0		0		1	
A 6	0		0		0		1		0		0		0	
A 7	1		1		1		0		1		1		1	

Dersom desisjonsreglene har lik sannsynlighet vil metode A kreve i gjennomsnitt  $R$  tester av resultatene fra de logiske operasjoner for å finne den søkte desisjonsregel, mens lagring av tabellmatrisen vil kreve  $R \cdot (2 + 1)$  ord.

I det prikkede felt  $B$  er selektorrutinen som velger ut aksjonsrutiner på grunnlag av  $X$ -posisjonene i  $B^3$ -ordet i den desisjonskolonne som ble funnet i desisjonsrutinen.

*Metode B:* En metode som krever relativt stor lagerplass men få tester pr. observasjon er følgende. På samme måte som metode A forutsettes en binærmaskin og for enkelhets skyld opprettholdes forutsetningen om at tallet på relasjoner og aksjoner hver for seg er mindre enn tallet på bits pr. ord i maskinen.

Hver desisjonskolonne representeres under genereringen ved et ordpar  $B^1$  og  $B^2$ . Betingelsesdelen omsettes i det første ordet  $B^1$  ved at  $J = 0$  og  $N = 1$ , mens

aksjonsselektoren omsettes i andre med  $X = 1$  og Blank = 0. Når en nå møter en kolonne med Blank i betingelsesdelen kopieres ordparet i to ordpar med henholdsvis 0 og 1 i den posisjon som svarer til Blank i desisjonskolonnen. Er det flere blanke elementer, f. eks.  $b$  blanke elementer, vil en således få i alt  $2^b$  ordpar, alle med lik  $B^2$ , istedet for den opprinnelige desisjonskolonne.

Genereringen vil nå gi en komplett desisjonstabell med i alt  $2^M$  ordpar eller desisjonsregler. Første binærord,  $B^1$ , kan oppfattes som et tall som identifiserer aksjonsselektoren  $B^2$  fordi alle betingelsesdeler nå er forskjellige. Sorterer vi derfor ordparrene med  $B^1$  som sorteringsnøkkel vil de  $2^M$  ordparrene bli entydig ordnet etter nummer fra 0 (=  $J J J \dots J$ ) til  $2^M - 1$  (=  $N N N \dots N$ ). Etter sortering har vi derfor ikke lenger noe behov for å beholde  $B^1$  fordi  $B^2$ 's plass i rekken gir samme opplysning.

Lagerplassen det vil være behov for til  $B_1^2, \dots, B_{(2^M)}^2$  er derfor  $2^M$  ord.

Desisjonsrutinen i programmet blir nå meget enkel. Først testes observasjonene mot relasjonene og testresultatet bygges opp i  $T$  på samme måte som i metode A. Desisjonen består nå i å hente fram aksjonsselektor nr.  $(T + 1)$ .

Bortsett fra de  $M$  tester av observasjonen krever således desisjonen ingen tester. Som en ser er denne desisjonsrutinen nært beslektet med den „binære strengkontroll“ som ofte nyttes i granskningsprogrammer.

Sammenligner vi metode A og B anvendt på eksemplet i figur 2 finner vi at metode A vil kreve  $R \cdot (2 + 1) = 21$  ord for lagring av desisjonsmatrisen, mens metode B i dette tilfelle bare krever  $2^M = 16$  ord. Dessuten vil metode A i dette eksempel omfatte i gjennomsnitt  $R = 7$  flere tester enn metode B.

Ser vi derimot på en tabell med 10 testrelasjoner og 20 desisjonsregler gir sammenligningen et annet resultat. Nå vil metode A og B kreve henholdsvis  $20(2 + 1) = 60$  ord og  $2^{20} = 1024$  ord for

lagring av desisjonsmatrisen. Metode A krever i gjennomsnitt 20 flere tester enn metode B mens den siste vanligvis vil kreve lengre genereringstid.

#### 4. Sluttmerknader

Et generatorprogram for EDB-maskinen CDC 3600 basert på metode A er utviklet og under utprøving i Statistisk Sentralbyrå i Oslo. Systemet tillater flere tabeller, set-relasjoner for kodekontroll, gjentakelse av tabellrutinene etc.

Desisjonstabellene og genereringsprogrammene er her framstilt som høvelige hjelpemiddel for å effektivisere gransking av statistiske individualoppgaver. For fullstendighets skyld bør det nevnes at de også kan nyttes på andre områder, f. eks. for å generere tabelleringsprogrammer. Om desisjonstabellene med tilhørende genereringsprogrammer vil være mer hensiktsmessige enn de spesielle genereringsprogrammer som utvikles med sikte på å løse tabelleringsproblemer er en annen sak.

## Decision tables and generation of computer programs for editing of statistical data


by *S. Nordbotten*

Statistical data are frequently so intensively processed that the quality of the results depends on a control of the individual observations. Previously this editing was performed by clerks, but to-day much of the work has been handed over to electronic computers. Both manual and automatic editing has their advantages as well as disadvantages.

The automatic editing required a much more precise specification of the control criteria, and the logical decision table has been proposed by American statisticians as a convenient tool for specification and documentation. Decision tables and their application in this field are discussed and a simple example is presented.

The decision tables are also a natural basis for developing standard editing programs, generators etc. The paper describes two approaches for construction of general generator programs which produce object programs for special editing application. The first program which has a relative small storage requirement uses logical operations on the decision matrix, while the second which is fast, but requires more space, implies direct computation of the action package number from the joint test result of the decision relations.

The construction of a generator based on the first mentioned method is reported.



Publikasjonen utgis i kommisjon hos  
H. Ascheoug & Co., Oslo, og er til salgs hos alle bokhandlere

Pris kr. 4,00