

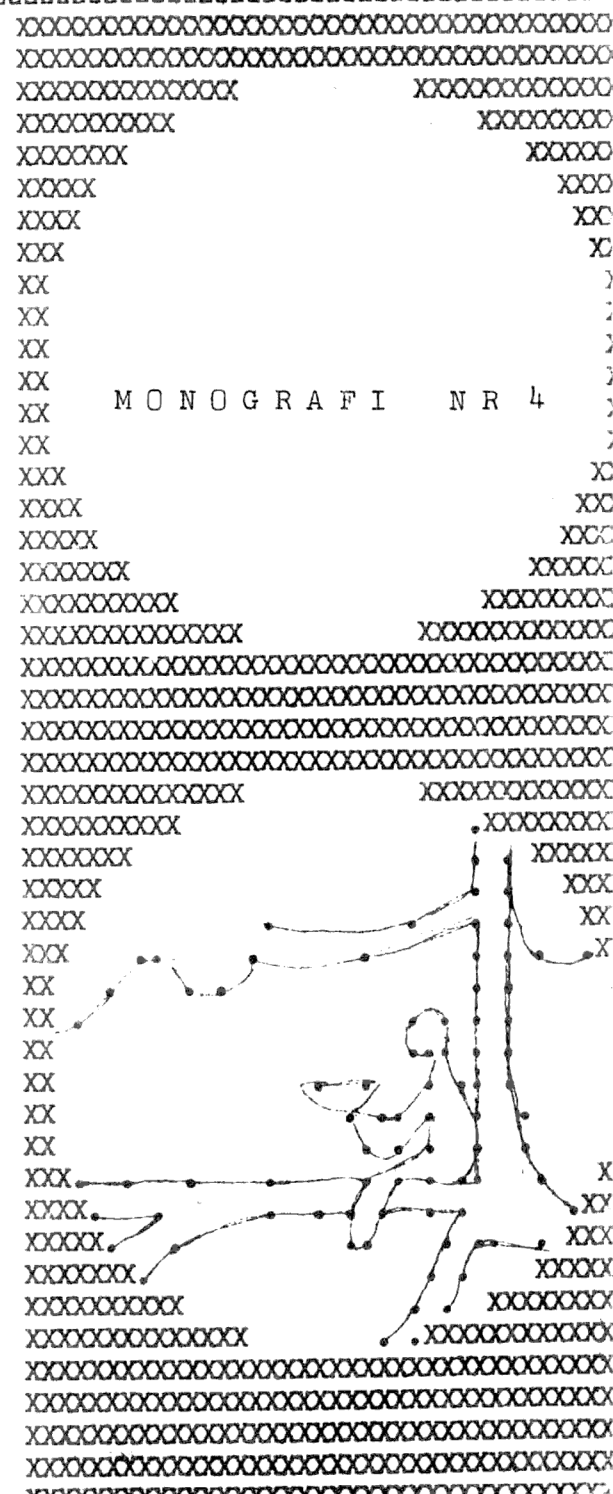
7.4

PSYKOLOGISK  
PÆDAGOGISK  
PÅ  
DLH  
JULI 1968

DATATRIMNING  
NOGLE OVERVEJELSER OM  
DATA REPRESENTATION AF  
PSYKOLOGISKE EKSPERI-  
MENTER

JAN RATTLEFF

Referat af et indlæg  
i lektor Svejgaards  
kollokvier i februar  
1968



Jeg vil redegøre for nogle metoder, vi har udviklet til behandling af psykologiske observationer. Det er kun en lille del af psykologien, som vi er nået frem til at kunne behandle automatisk, og den lille del vil jeg prøve at formalisere en smule ud fra en test-situation.

Det vil være almindeligt, at man har et prøveark, og at man sætter en forsøgsperson ved siden af prøvearket. På den anden side af bordet sidder en forsøgsleder, som iagttager forsøgspersonen. Forsøgslederen vil have noget at vide om nogle processer, der kan tænkes at foregå inde i hovedet på forsøgspersonen. Som sit første udgangspunkt har han en registrering af forsøgspersonens reaktion. Forsøgslederen sætter forsøgspersonen i en bestemt situation, og på den lille engelskprøve, der er vist på fig. 1, er de forskellige situationer nogle sætninger, som kan udfyldes med et ud af tre ord. Der står en parentes i hver situation med tre ord i, og det er meningen, at forsøgspersonen skal reagere med at sætte en streg under det af de tre ord, som han mener, er rigtigt. Nogle gange kan de situationer bestå i, at man viser forsøgspersonen et billede, og da vil hans reaktion ikke være nogle streger, men nogle lyde eller luftsvingninger, han sætter igang, og disse registrerer forsøgslederen på den måde, at han perciperer dem, således at de bliver til nogle ord, som han kender; men man kan også tænke sig, at de bliver registreret på en båndoptager, således, at man har fastholdt lydsvingningerne.

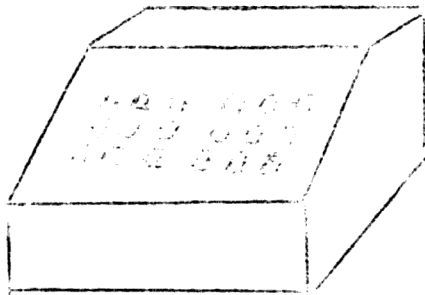
Vi har så prøvet at lave en abstraktion af de forskellige typer af psykologiske eksperimenter, som vi gerne vil behandle automatisk. Man laver også empiriske psykologiske undersøgelser på andre måder end ved hjælp af eksperimenter. Nogle gange spørger man forsøgspersoner, hvad de har været ude for, men det er noget, vi ikke vil gøre rede for her.

Der er altså noget, der hedder det psykologiske eksperiment, og det er karakteriseret ved, at vi har nogle personer og nogle påvirkningskræfter i påvirkningssituationer og nogle reaktionsmåder. Personerne kalder vi N (det er noget, man traditionelt gør, det står for >>number<<); situationerne kalder vi K (det er påvirkningskræfter, der virke på forsøgspersonen og reaktionsmåderne kalder

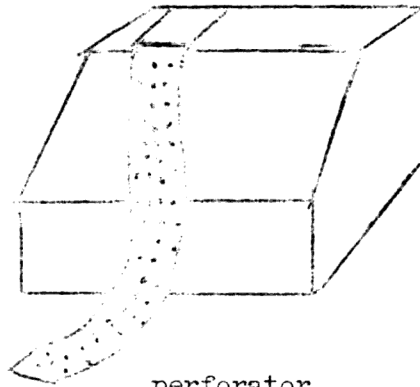
FIG I

In order to cross the ocean people have to use ( houses, ships, kingdoms ) osv. osv.

FIG II

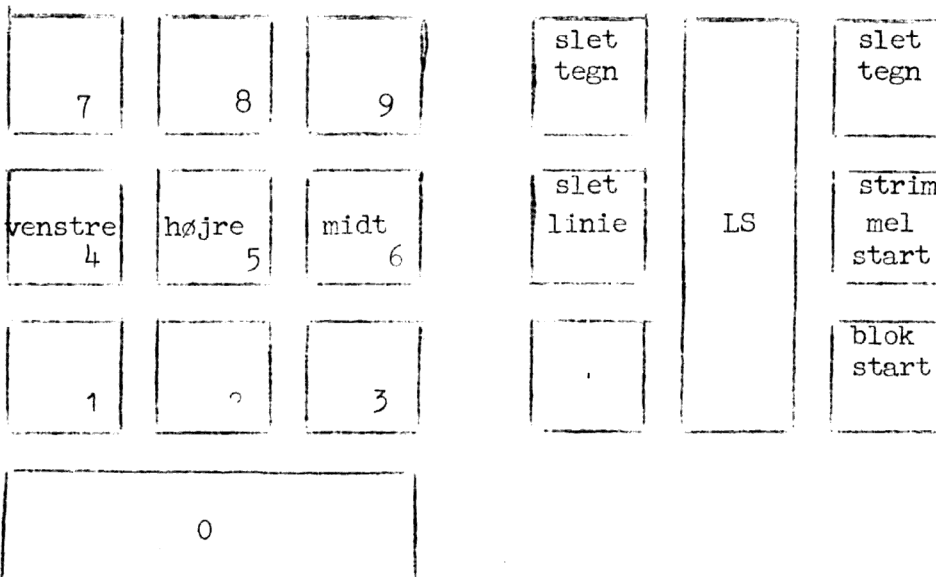


tastatur



perforator

FIG III



vi M. Så har vi noget, vi kalder det oprindelige system og noget, vi kalder for det beskrevne system. Ved det registrerede system vil vi sige, at vi har N personer med indeks OP.

Det står for det oprindelige antal personer, og så har vi en mængde af personer i den gruppe, der går fra 1 til NOP. KOP står for det oprindelige antal påvirkningskræfter, der går fra 1 til KOP. Og MOP står for det oprindelige antal reaktionsmåder, der går fra 1 til MOP. Det vil vi transformere til det beskrevne system, idet vi vil beskrive det som gående fra 1 til N og fra 1 til K og fra 1 til M, og hvis både N er lig med NOP og K er lig med KOP og M er lig med MOP, så vil der ikke være meget overblik skabt ved at gå fra det registrerede system til det beskrevne system; men som regel vil man enten sammenfatte de personer, der minder om hinanden i grupper, eller man vil slå flere spørgsmål sammen eller man vil sammenfatte flere måder at reagere på. Det oprindelige antal af personer kan evt. stå for antal af persongrupper, idet vi kunne tænke os, at det var en gruppes reaktion, vi vil registrere samtidig. I almindelighed vil det være en enkelt person.

Denne terminologi er udviklet i overensstemmelse med G. Rasch (personlig kommunikation) og er tidligere beskrevet (Karpatoff og Rattleff 1967).

Der er en asymmetri m.h.t. kendtheden og ukendtheden af disse dimensioner. Vi ved ikke på forhånd, hvor mange personer, vi vil registrere. Vi lægger en forsøgsplan, hvor nogle af børnene er syge, og nogle af de voksne kan nægte at komme. D.v.s. at NOP er ukendt for os. Antallet af påvirkningskræfter kender vi derimod på forhånd. Det ligger i vores forsøgsplan, på den måde vi laver vores prøveark på. Antallet af reaktionsmåder er den tredje dimension. Det kan enten være kendt eller ukendt. I dette tilfælde, hvor vi har tre muligheder i parentes - houses, ships, kingdoms - kan hver af de tre dele besvares ved en streg eller ikke streg. Dvs., at vi ved simpel kombinatorik kan udregne det mulige antal af reaktionsmåder, der er for forsøgspersonen i en sådan situation, hvor han bliver udsat for tre små delspørgsmål, der hver kan besvares på to måder. Det er et eksempel på, at antallet af reaktionsmåder er kendt, og det er selvfølgelig kun de registrerede reaktionsmåder, vi omtaler. Han kan også klø sig i håret, men det registrerer vi ikke. Hvis vi derimod viser en forsøgsperson et billede og spørger, hvad det ligner, så ved vi ikke på forhånd - vil vi ikke på forhånd kun-

ne forestille os, hvor mange forskellige måder, han kan svare på. Det er en principielt anden type af psykologiske eksperimenter, mener jeg.

Vi har to forskellige måder at registrere resultaterne på ved disse to typer af forsøg. Man plejer traditionelt at bruge en kodebåndsmaskine til at registrere svaret på, ligegyldigt om man kender antallet af udfald eller ej. Her har vi udviklet en anden meget sjov teknik, som jeg vil fortælle lidt om. Det viser sig nemlig, at når antallet af reaktionsmuligheder er kendt på forhånd, så er det også meget lille, fordi det betyder, at man har dannet sig et godt overblik over problemstillingen, når man kender måderne, sådan et forsøg kan blive udført på, og d.v.s. at man på forhånd har valgt et ret begrænset antal af måder. Hvis man tænker sig, at man vil registrere hver måde, f. p. kan svare på, ved hjælp af cifre, så vil det være nærliggende at benytte at apparatur som det på fig. 2 viste, hvor der ikke er ret mange knapper. Her vil det være rart, hvis vi har et tastatur, der ikke har så mange flere knapper, end der er udfald i forsøget, idet hver knap så kan stå for et udfald. Tastaturet kan kobles til en kodebåndsp perforator, og det fører frem til det næste, jeg vil tale om, nemlig hvordan man kan beskrive overførslen af information fra den prøve, der er vist på fig. 1, til kodebåndet. Der ses man i den begrænsede litteratur, jeg kender til, ordene: kodning, renskrivning, transkription og hulning brugt. Jeg vil mene, at de to ord, der vil være mest rimelig til at beskrive det, vil være enten renskrivning eller transkription, idet kodning involverer, at der skal ske en eller anden process, og hulning involverer, at hulledamen skal sidde og tænke på, hvilke huller, hun skal slå i strimlen.

Når vi i praksis skal overføre resultatet af sådan en prøve til hulfåndet, har vi her på tastaturet anbragt nogle betegnelser, idet vi nemlig ikke siger til hulledamen, - hvad vi kunne være fristet til at sige, - at hver gang personen svarer på den første valgmulighed i parentes, så skal hun trykke på et, og hver gang han svarer på de andre, så skal hun trykke på to og på tre. Vi har istedet givet hende et kort over dette tastatur. Til hver af vores opgaver har vi en sådan tegning af tastaturet, (som vist på fig. 3) og så siger vi, at dette er ikke et ettal eller to- eller tretal, men dette er et venstre og det er et midt og det er et højre, således at

vi giver ikke den instruktion til den, der skal hulle det ind, at man skal trykke på et ettal, når der står venstre, men vi siger, at denne knap hedder venstreknappen, denne midtknappen og denne højreknappen. På den måde skal den, der sidder og transkriberer ikke sidde og huske, hvad hvad betyder, men skal simpelthen lære, at denne knap hedder venstre, denne midt og denne højre. På den måde kan vi komme ud over, at vi først skal skrive det rent, der skal hules ind, og derpå foretage en indhulning, men vi kan lave en kombineret renskrivning og transkription, idet vi skifter datarepræsentation over til hulbånd. Så vil vi kunne repræsentere sådan et prøveark, som det, der er vist på fig. 1, med nogle tal, uden at vi i og for sig formulerer nogle kodningsl., som man skal gå og huske på

Dette var lidt om de psykologiske data, og jeg kan til slut nævne, at hvis vi registrerer svaret ved hjælp af ord, som vi noterer ned på et stykke papir, så er der ikke anden mulighed end at skrive på kodebåndsmaskine, hvad forsøgspersonen har svaret. Grunden til at vi bruger et lille tastatur er selvfølgelig, at vi regner med, at transkriptionen bliver hurtigere og sikrere, når der er færre muligheder for at slå forkert, men selvfølgelig er der mulighed for at slå >>houses<< i stedet for >>ships<< stadigvæk.

Så vil jeg fortælle lidt om en formalisering af de datastrukturer, der kan være tale om for os at arbejde med ud fra, hvordan vi mener, at et psykologisk eksperiment kan forløbe.

Som regel har vi altså en persons data ad gangen stående på hulbånd, sådan at der efter hinanden kommer forskellige personer. Der er tale om to forskellige slags data i forbindelse med vores forsøgsplan, dels nogle, der karakteriserer ham som forsøgsperson inden for denne mængde fra 1 til NOP, der fortæller, hvilken forsøgsperson, han er, således at vi bagefter kan kende ham igen fra de andre, dels nogle der fortæller, hvilke påvirkningskræfter, han er blevet påvirket af. Dvs. at det vil være et eller andet tal mellem 1 og KOP, og endelig vil der være tale om den måde, han har besvaret på fra 1 til MOP. Det vil kun være de data, som skiller forsøgspersonen, som vi tager med, når vi skifter datarepræsentationen til hulbånd. I forsøget her f.eks., om et givet svar er rigtigt eller forkert, er noget vi senere kan finde ud af ved at sammenholde beskrivelsen af prøvearkene med hans svar. Det behøver vi ikke at repræsentere i data. Det viser sig, at personen som regel er karakteriseret ved,

at man har lavet nogle andre forsøg med ham end det lille forsøg her. Man har nemlig gjort det forsøg, at man har spurgt ham, hvornår han er født feks.. Det er et psykologisk forsøg, hvor man har udsat ham for nogle andre påvirkninger og har fået nogle andre responser frem. Man vil enten spørge ham, hans lærer eller hans søn om, hvilket erhverv, han har. D.v.s at man faktisk udfører en række andre psykologiske eksperimenter på personen. Det vil være ret generelt for sådanne forsøg, at der er en hel del mere at registrere end det, der angår selve forsøget. Det fører frem til, at hver person må betegnes med et >>HOVED<<på den datablok, vi repræsenterer ham med. Vi må desuden have en KROP i vores datablok. Det er den, der angiver, på hvilken måde, han har reageret på vore påvirkningskræfter. Vi har så en samling af disse blokke. Det kalder jeg en STRIMMEL. Den består af noget, vi kalder et STRIMMELHOVED, en række blokke og et afslutningstegn >>END<<. Dette vil i populær forstand være det, man har på en rulle, når man har fået omsat det til hulbånd. Vi behøver altså ikke at meddele i starten af hver strimmel - i STRIMMELHOVEDET - hvor mange blokke, der kommer. Der kommer simpelthen lige så mange blokke, som der kan være hen til >>END<< - koden til sidst.

Vi vil så skelne mellem forskellige synspunkter, man kan anlægge, når man skal vurdere, om en given datarepræsentation er hensigtsmæssig eller ej.

Vi vil se på to af de faser, der er i arbejdet med forsøgspersonerne, når vi skal omsætte dem fra det registrerede system til det beskrevne system. Der vil normalt være tre faser:

Den ene vil vi kalde for *xtrimning*. Det er alt, der har med at gøre med at sikre os, at udfaldene af de oprindelige engelskprøver i dette tilfælde bliver repræsenteret rigtigt på et hulbånd.

Dette medfører, at vi må tale om, hvad rigtigt eller forkert vil sige ved >>trimning<<.

Den næste fase vil vi kalde >>orientering<< Her orienterer vi os om, hvad der i grunden er sket under forsøget. Een vigtig ting ved orienteringsfasen er at finde ud af, hvor mange måder, der er svaret på. Her kan vi feks. ikke sige, at et verbalt svar kan have en vis længde og så undersøge alle de mulige kombinationer af bogstaver, der giver tekststrengene af denne længde. Det er ikke teknisk muligt.

I stedet bliver vi nødt til at opbygge et katalog over måder, en bestemt påvirkningskraft kan blive besvaret med, og så efterhånden finde ud af, hvor mange svar, der er.

Endelig har vi den tredje fase, som vi kunne kalde den *xstatistiske* fase, hvor vi opstiller forskellige hypoteser om samvirkende kræfter og reaktionsmåder osv.

To ønsker kan gøre sig gældende i den process, hvor vi skal fremstille et databånd, som vi kan bruge dels til at *xtrimme* d.v.s. bygge op, så det bliver mere og mere rigtigt, og dels til at vi kan arbejde videre, så at vi kan komme ind i orienteringsfasen. Det vil som regel være sådan, at vi kan have to forskellige datarepræsentationer: en som vi bruger, lige når vi transkriberer det over til hulbåndet; og en senere datarepræsentation. For øjeblikket har vi kunnet gøre det sådan, at de to datarepræsentationer har samme generelle struktur, sådan at vi i begge tilfælde har kunnet have et *xblokbegreb*. Det er noget meget centralt, idet det viser sig, at vi ved at udskille det blokbegreb har lavet nogle generelle programmer, der kan lave en fejlanalyse af, om en transkription kan være sket rigtigt eller forkert.

Når jeg har kaldt det *xtrimning* og ikke *xfejlsøgning*, som det almindeligvis kaldes, er det fordi, fejlsøgning betyder, at man finder nogle fejl; men man retter dem i og for sig ikke, medens trimming gerne skulle gå på en mere raffineret proces. Vi regner med, at databåndet er trimmet, til det har en passende grad af støjfrihed - det er ikke *xrigtigt*.

Man vil ofte m.h.t. fejl skelne mellem to typer af fejl: *strukturfejl* og *indholdsfejl*, eller man vil muligvis have *tredeelingen*: de grammatiske, syntaktiske og semantiske fejl, hvor grammatik går på de enkelte dele, syntaks på, hvordan de sættes sammen og semantik på, hvad de enkelte dele betyder. Nu har jeg blot skrevet, at vores datastruktur består af et hoved og en krop. Denne krop består selvfølgelig også af nogle andre dele, og da vil jeg benytte en almindelig måde at inddele sådan noget på : sætninger, ord og tegn. Det vil ikke være sådan, at tegn repræsenteres ved et enkelt



tegn eller et enkelt anslag på skrivemaskinen, men det går på tre niveauer i strukturanalysen: sætninger, ord og tegn.

I almindelighed, når man skal interpretere sådan noget på sit papir, vil der være en flydende overgang, hvornår er noget struktur og hvornår er noget indhold. Det er ihvertfald svært for mig at se grænserne. Men jeg vil alligevel prøve at sætte det i relation til den ide. Her kunne jeg tænke mig, at man kunne skære et sted imellem struktur og indhold og sige, at der er to måder, man kan afsløre fejl på, og det kan vi bruge til at klassificere fejlene med. Den ene fejltype kan vi undersøge ved at konfrontere med nogle regler, og den anden fejltype kan vi afsløre ved at konfrontere med  $\times$ virkeligheden $\times$ . Det vil da være spørgsmålet om, hvor stor en del af virkeligheden, man lægger ind i sine regler. Dette vil bestemme, hvor den grænse kan gå. Bostrup har benyttet to ord, som jeg synes, er meget gode. Han skelner mellem umulige og utrolige fejl, eller rettere sagt hændelser. Han mener, at man kan få oplysninger om begge dele af systemet. De umulige er umulige, de utrolige kan godt være mulige, men vil sjældent være det. (Bostrup, 1968).

Når man skal vælge, hvor meget man skal tage med under  $\times$ struktur $\times$ , vil det stort set være et valg, der er bestemt af, hvor let det vil være at lave programmer, som kan foretage en strukturkontrol. Jo mere man får med under struktur, jo mindre under indhold og omvendt.

Jeg kunne tænke mig at skrive nogle simple regler op, som man kunne tænke sig at undersøge, om datablokken tilfredsstillende. Sådanne simple regler vil være - hvis vi nu tænker os, at datakroppen består af nogle ord - at se, om det enkelte ord ligger inden for et område af ord. Hvis man kan nævne et tal, der er mindre end alle de ord, der forekommer på det sted, og hvis man kan nævne et, der er større, vil det være simpelt at teste reaktionsområdet her. Noget andet, man kunne undersøge, var sådan noget som dividerbarhed, om feks. et tal altid skal være deleligt med 5. Det vil også være forholdsvis simpelt at undersøge.

Noget man også kunne undersøge er, om en person, der går i 7. klasse, kan tænkes at være 30 år eller sådan noget. Det vil kræve, at man går ind i datablokken to forskellige steder og finder ud af

to forskellige punkter af blokken. Det er selvfølgelig mere kompliceret at lave et program, der kan undersøge en sådan afhængighed.

Der ligger i reglerne et eller andet med, at vi har en vis ufrihed - det vi kalder en datastruktur. Dvs., at nogle data ikke kan varieres vilkårligt, men der er nogle regler. I virkeligheden - xindholdetx - har vi en stor frihed, og det er klart, at der altid må være en vis variation tilladt på vores databånd. Hvis vi fuldkomment kunne beskrive resultaterne af det psykologiske forsøg i form af nogle regler, så var der ingen grund til at lave forsøget, og slet ikke til at bruge tid på at databehandle det bagefter. Ofte benytter man den metode, at man (ved hjælp af et menneske) tilføjer redundans dvs. man udregner feks. nogle kontrolsummer, inden man transkribere det over til hulbåndet. Feks. hvor mange svar har han i venstre parentes, og så lader man systemet undersøge, om der er overensstemmelse mellem de to oplysninger. Ofte vil det vise sig, at de fejl, man finder, skyldes, at man har regnet forkert, når man har lavet kontrolsummerne.

Så har man den anden hyppigt brugte metode, der hedder korrekturlæsning, hvor en læser højt, og en anden lader som om, han følger med i et manuskript, og skal sige nej, hvis det ikke passer, hvad der står.

En tredje måde er kontrolhulning, som er meget lidt brugt, når man arbejder med hulbånd, men meget hyppigt brugt, når man arbejder med hulkort. Jeg ved ikke, om det afspejler nogle personlighedsfaktorer hos de to grupper af brugere. Kontrolhulning består i, at man har en maskine, som man kan lade indlæse den første transkription, man har lavet, og så sidder der en ved siden af, som skal lave en ny transkription. Kun hvis vedkommende trykker på de samme knapper, som der er trykket på første gang, vil det gå glat igennem, ellers stopper maskinen. Ved en fejlmeddelelse kan operatøren af maskinen så se, hvad det mon skulle være. Man placerer ansvaret helt asymmetrisk hos det sidste led i en sådan kontrolhulning.

Endelig er der den sidste metode, som ikke så mange bruger. Den hedder dobbelttranskription. Her lader man foretage to fuldkomment ens transkriptioner af to forskellige personer, og så får man to

halstrimler ud, og lader det automatiske system sammenligne, om de er identiske, og så meddele afvigelserne. Så siger man, at når to mennesker har lavet de samme fejl, vil den kun blive lavet et sted. Det er noget, man kan diskutere (Bostrup, 1968).

Dette er i og for sig de fejl, der kan opstå. Der er desuden sjovt nok en anden type af fejl, som det er meget vanskeligt at tænke sig på forhånd, hvis man laver en analyse ud fra dette. Det vil være en nummerfejl. Den er hverken struktur eller indhold. Den vil være tab eller gevinst. Hulledeamen snider f.eks. sine blanketter væk, eller de bliver siddende et andet sted i systemet, eller der bliver pludselig flere forsøgspersoner under arbejdet. (Dette er velkendt fra rotteforsøg, men af en anden grund). Jeg ved ikke rigtig, hvad vi skal kalde det, det er i hvert fald en type af fejl, som man kan begå.

Jeg vil sige lidt mere om den datastruktur, som vi har opstillet. Vi har hovedbegreberne xblokx og xstrimmelx, desuden har vi et tredje hovedbegreb, som jeg ikke har nævnt endnu, som hedder en xtekstlistex. En tekstliste er i og for sig en beskrivelse af den implicitte information om prøvearket eller påvirkningssituationen. Det er noget, vi tit har brug for at tilføje systemet på forskellige trin. Det kan f.eks. bruges til at undersøge, om spørgsmålene er besvaret rigtig eller forkert. Vores strimmel har lov at starte med et strimmelhoved, det betyder sådan set bare, at der er mulighed for at lave en kommentar i begyndelsen af en strimmel. Det ideelle - når systemet bliver fuldt udbygget - er at dette strimmelhoved automatisk bliver overført fra process til process ved trimningen, således at man hele tiden ved, hvad det er for en strimmel, man arbejder med. Slutningstegnet medfører, at vi ikke behøver meddele noget om, hvor lang strimlen er. Vi kan have lige så lange eller korte strimler, som det er praktisk. Vi kan have vores forsøg samlet på en eller flere strimler, ligesom det er bekendt. Vi kan selvfølgelig også rent fysisk lime to strimler sammen med klisterpapir, så vil de stadigvæk rent formelt blive opfattet som to strimler.

Det er desværre ikke formelt lykkedes os formelt at definere

struktur helt tilfredsstillende endnu. Jeg tror næsten, det nemmeste vil være, at jeg skriver et eksempel op på en datablok, f.eks. en, der kunne bruges til at karakterisere udfaldet af engelsk-forsøget, nævnt side 1. Allerførst skal vi definere en række værdier: START, LS, PUNKT, KOMMA og END, og det er altså nogle enkelttegn, som vi vælger frit, og som bare er nogle forskellige tegn. Kroppen består af en række helheder, og hver af helhederne består af en række dele. Hvis vi skulle transkribere prøven fra fig. 1 til sådan en datablok, så ville vi først have et start-tegn, f.eks. et q, så ville vi have et blokhoved, som består af et løbenummer, f.eks. kan det her være person nr. 17, og så består det af de andre test-uafhængige oplysninger, alder og andre forskellige ting f.eks. kan personen være 14 år, og så kan der være en anden oplysning om hans klasse. Der skal altid være et nummer, på det hoved, og så kan der være nogle tal. Bloknummeret skal være forskelligt for alle de blokke, vi arbejder med. I kroppen vil en helhed repræsentere et enkelt delforsøg, det vil altså her sige, at vi har 7 helheder, og hver helhed består her kun af en enkelt del, idet der nemlig ikke er nogen nærmere beskrivelse af påvirkningssituationen heri, den vil være implicit givet, så det vil sige, at hvis vi repræsenterer med nogle cifre, om man har svaret til højre eller til venstre eller i midten, så skulle vi bruge punktum som skilletegn, som adskiller de forskellige dele i kroppen, og så vil vi lade det sidste være et lineskift-tegn. Så kommer næste blok.

I visse tilfælde vil vi have det sådan, at vi kun registrerer nogle af delforsøgene. Et meget hyppigt psykologisk forsøg vil bestå i, at MOP er lig med 2, d.v.s. enten kan han svare rigtigt eller forkert. Så vil det ikke være nødvendigt at registrere de tilfælde, hvor han svarer rigtigt, f.eks. På den måde skal vi til gengæld angive numrene på dem, han har besvaret forkert, så vil det altså sige, at vores helheder i kroppen består ikke i reaktionsmåder, men af et nummer, hvorpå en bestemt reaktion er sket. Og det behøver ikke at indvolvere, at det oprindelige system indeholder 2 muligheder, det kan godt være, at det indeholder 7 muligheder, men der er 6 forkerte og kun en rigtig, og vi er bare interesseret i den, og det ved vi på forhånd, og så er der ingen grund til at meddele, hvad han har reageret med, bare hvad numre, han har reageret på en bestemt måde på, så angiver vi her numrene på dem, f.eks. nummer 3

har den pågældende reaktion. Den næste person - det kan være person nr. 18, som også går i 7. klasse, - har måske kun reageret på den måde på to af spørgsmålene, spørgsmål 1 og spørgsmål 17. I det tilfælde vil vi sige, at vores blokstruktur er helt fast. Fast eller ikke fast vil være en hovedinddeling af blokkene. Når strukturen ikke er fast, er konventionen den, at der sidst i hovedet skal stå, hvor mange helheder, der er i den pågældende krop. I og for sig har vi to forskellige systemer, for det er to meget forskellige tilfælde at behandle. I det her tilfælde har vi altså et blokstarttegn og en række helheder, og vi kan se, at vi ikke rigtig har nogen dele, det er altså ikke altid, vi har alle niveauer repræsenteret i vores blokke.

Så vil jeg sige en lille smule om, hvordan jeg har tænkt mig, at det system skal udarbejdes, som skal udføre kontrollen med disse ting. Desværre er det system, som vi bruger i øjeblikket, så utilfredsstillende efter det overblik, jeg har fået efter at have arbejdet med det, at jeg ikke vil beskrive det her, (men se Rattleff, 1967, og Rattleff og de Waal, 1968). Jeg kan beskrive nogle detaljer i det, som kan være interessante. M.h.t. kontrollen af de strukturelle regler have jeg forestillet mig, at det skulle foregå ved hjælp af et program med den struktur, som Ole Brun Madsen omtalte (lektor Svejgaards datalogikollokvier, efteråret 1967), som består af en tilstandstabel. Der har vi en række symbolkasser, og vi har en række tilstande, som vi kan tænke os at befinde os i. Jeg tror, det kan betale sig at udføre to typer af strukturel analyse på sådan en blok, nemlig dele den ind i hoved og krop og tage de to hver for sig.

Jeg skulle måske starte med de mest omfattende klasser og diskutere, om det er muligt at kontrollere om strukturen:  $\times$  en strimmel  $\times$  er tilstede. Og det vil jeg ikke mene, det er, for hvis der mangler en end-kode på et databånd, vil Gier gå i en undefineret tilstand, fordi man mangler en måde til at se, om data er sluppet op, og så vil det ikke være muligt at få gennemført sådan en analyse, og det samme vil være tilfældet med en kommentar. En strimmelkommentar vil man ikke kunne analysere, for den vil jo være defineret som det, der kommer inden den første blok. Hvis vi ikke får noget output, så ved vi, at der ikke har været en strimmel.

Jeg kunne tænke mig at analysere blokhovedet ved hjælp af indlæsefunktionen `readinteger` i Gier algol 4. Det vil være praktisk. Der kunne man godt f.eks. tillade, at der var nogle overflødige delimiters.

Hvis vi er enige om, at alle programmer læses med en række kald af `readinteger`, skulle det i og for sig ikke gøre noget, at der forekommer en række mærkelige ting heroppe - så er det defineret i og med definitionen af `readinteger`, (Naur, 1967).

Jeg vil derfor i det følgende kun betragte kroppen, og der kan vi også sige, at forskellen på hoved og krop består i, at hovedet repræsenterer data på mere konventionel måde, det er nogle bioplysninger, vi godt vil have med. Vi har en række symbolkasser: blinde tegn, ulovlige tegn, starttegn (som altid kun indeholder et tegn) og andre tegn, det er alle de lovlige tegn, og `xLSx` og punkt (som er terminator for en helhed) og endelig `xendx`.

Så er der en række tilstande, som vi kan være i, vi kan f.eks. være midt inde i en helhed, eller vi kan være efter en helhed eller lige før start. Så kan vi skrive, hvorfor en tilstand programmet skal gå over i, når det i en bestemt tilstand møder et tegn af en eller anden klasse. På den måde vil man kunne skrive det hele som et casestatement. Det viser sig, at den tabel, som jeg arbejder med, bliver ganske overordentlig kedelig, idet der kommer til at stå det samme under alle tegnene. Det er selvfølgelig meget rart, at det er sådan, men det bliver lidt mindre spændende at fremstille, idet jeg så ikke behøver at have en fin transitionsmatrix.

Det var i og for sig strukturkontrollen; men så vil jeg yderligere på det sted i trimmefasen indkorporere en undersøgelse af, om en bestemt helhed ligger indenfor et bestemt variationsområde. Der vil jeg have to arrays: `min` og `max`. Jeg vil desuden tale om typer af en helhed. En helhed kan have to forskellige typer, den kan enten være af typen `string` eller af typen `integer`. Størrelsesområdet for en helhed, hvis den er af typen `integer`, vil være et tal, og det vil kunne beskrives som liggende i et interval mellem to heltal. Hvis den er af typen `string`, vil dens størrelse kunne defineres efter hvor mange tegn, der indgår i helheden, d.v.s., at jeg kan arbejde med en værdi, der hedder størrelsen af en helhed, og det er

et antal tegn. Og der kan man så bekvemt i programmet indsætte den funktion med tre arrays, en, der meddeler mindsteværdien af helhederne, en, der meddeler størsteværdierne, og en, der meddeler om den pågældende helhed er af den ene eller den anden type.

Det er stort set den kontrol, som jeg mener, det kan betale sig at tage med på det strukturelle niveau.

Jeg nævnte før nogle *xumuligx* og *xutroligx* oplysninger. Det begreb vil jeg ikke bruge, jeg vil bruge *xområdex* her. Og så tror jeg ikke, at det kan betale sig at bruge dividerbarheden. Det bliver nok for specielt at udnytte. Desuden vil jeg benytte, om der forekommer enkelte ulovlige tegn. Det er de kontroller, der bliver lavet. Altså: er der det bestemte antal enheder i kroppen, som der skal være, og er der nogle ulovlige tegn i kroppen, og ligger enhederne inden for det variationsområde, de må mht. størrelse. Desuden vil jeg i hovedet se, om der er det antal tal, der skal være.

Så kommer vi til de andre kontroller, jeg har omtalt, og som jeg har prøvet. De bliver nævnt i den rækkefølge, jeg har prøvet at benytte dem i. Jeg startede med at benytte menneskelig tilførsel af redundans, som jeg mener, ikke kan betale sig, fordi mit princip er at få brugt så meget maskintid som muligt. Dvs. at få så meget hjælp af datamaskinen som muligt. Så mange som muligt af de processer, som skal udføres her, skal udføres automatisk. Når man så har fundet ud af, hvordan processerne kan udføres automatisk, må man finde ud af, hvordan de hurtigt kan udføres automatisk, men det er ikke blevet noget problem endnu. Og i øvrigt vil jeg mene, at de processer er så enkle, at de kan udføres med enkle og effektive programmer. Hovedsynspunktet har for mig været, at det skal være et menneskevenligt system. Det skal altså hjælpe med at finde og rette fejl. Det skal ikke bare først kræve, at visse regler bliver overholdt, og så skælde ud, når de ikke bliver det. Derfor går redundanstilførsel ikke, og korrekturlæsninger går heller ikke, for det er simpelthen det værste arbejde, man kan beskæftige sig med, foruden at det giver en lav sikkerhed.

Så er der kontrolhulning. Her har jeg prøvet to systemer. Kontrolhulning vil sige, at man har en maskine, der læser det hulbånd ind, som man har lavet, og som man skal arbejde sammen med, og som så siger, om der er uoverensstemmelser, og så skal man meddele, om det er en selv, der har ret, eller den foregående transkribent.

Her har jeg prøvet kontrolhulning ved hjælp af to forskellige maskiner. Den ene er Fridens »Veripunch«, som er en modificeret flexowriter, som kan lave det arbejde. Den giver ingen udskrift af strimlen. Den læser en strimmel ind, så har man et tastatur, man skal skrive på. Nogle lamper lyser, hvis der er uoverensstemmelse.

Så har Bodenhoff en, der hedder Mohawk, som kan læse ind på magnetbånd og undersøge, om det er i orden. Jeg har med meget velvilje fra Bohrinstituttet fået lov til at konventere nogle data til magnetbånd, og med velvilje fra Bodenhoff fået lov at sidde derinde og lege med den maskine. Det var heller ikke noget, jeg brød mig om, idet en analyse af de arbejdsprocesser, som et menneske laver, der kontrolhuller, viser, at der stort set er tale om to typer af processer, der bestemmer, om noget skal opfattes som fejl eller ej. Det ene er, om man mener, at den, der har lavet den foregående strimmel er sød, og den anden består i, om man mener, at vedkommende er idiot. Det vil medføre en tendens til at synes, at man selv har ret eller den anden har ret, som ødelægger det efter min opfattelse, foruden at det er et kedeligt arbejde.

Så er der dobbelthulning tilbage at bruge. Den har den fordel, at hvis man beskriver det, der foregår i arbejdssituationen, behøver de to transkribenter ikke at vide, at det er dobbelthulning. Det vil virke som enhver anden transkription. Det har den anden fordel, at der bliver taget uafhængig stilling to gange til de oprindelige registreringer. Det har endelig den fordel, at vi får noget at vide om vores databånd. Det er ikke noget, der bliver undertrykt på en tidlig fase. Vi får noget at vide, og det kan muligvis medføre, at vi finder ud af, at det var en dårlig tilrettelæggelse som gør, at en bestemt fejl hyppigt opstår. Der kan ved hjælp af nogle statistiske metoder undersøges noget om værdien.



Ud fra hvor mange uoverensstemmelser, der findes, kan man sige noget om, hvor mange, der så er tilbage. Der vil nemlig være en del fejl her, der vil være identiske. Man kan begå en fejl det ene sted og det andet sted, som er den samme, og det vil systemet ikke opdage. Men i øjeblikket vil vi mene, at metoden er sikker og god.

Ved hjælp af vores bloknumre, som vi kræver skal være forskellige allesammen og desuden fortløbende inden for den mængde, vi arbejder med, kan vi lave et program, som kan undersøge, om der er tab eller gevinst. Desuden kræver det, at vi ved, hvor mange prøveblanketter, der oprindeligt var. Det må vi have talt op, uafhængigt af transkriptionen.

Jeg vil lige sige noget om en teknisk finesse, som vi har udviklet, og som er et resultat af det lille tastatur, og som gør sig gældende når man arbejder med sådan et apparat. Det gør man jo ved at bruge venstre hånd, som hviler på den lille knop, og så kan man arbejde i blinde her, medens man har fuld opmærksomhed til blanketterne. Man kan sidde ved et skrivebord, som kan være stort og dejligt, og det støjer ikke, fordi apparatet står i et andet rum. Man sidder så her og har fuld opmærksomhed til det krævende arbejde, det er at holde sig orienteret i manuskriptet, og kan tyde, hvad der står, uden at man behøver at se på knapperne.

Herved opstår der forskellige vanskeligheder. For det første at man ikke kan se, hvad det er for et tegn, man har lavet. Man kan ikke se, hvor man er, hvis der pludselig kommer en forstyrrelse. Vi kan tænke os, at vores prøveark stort set er organiseret fortløbende, således at oplysningerne skal transkriberes i den rækkefølge, de kommer i. Vi kan tage det mest banale først. Det vil være det, at man siger, hov, jeg kom til at trykke på venstre. Det skulle være højre. Så kan man gøre det, de anbefaler i brochuren: gå ud i baglokalet og skrue et hak tilbage. Det er meget vanskeligt at gøre. Det bliver altid gjort forkert. Så kan man lave et specielt tegn til alle huller, således at det bliver *xoverhulletx* og senere skal opfattes som blindt. Her ville det rareste være, hvis der var en fejlknop, som selv gjorde det, hvis der her på siden

var en knap, der betød ~~xannullerx~~ det sidste tegn. Det har vi derfor også lavet. Så trykker man lige så mange gange på den knap, som man trykkede forkerte tegn. Dvs. at hvis der skal stå AABACD, og vi kommer til at skrive AABCD, så siger vi, hov, der skulle have været et A imellem B og C. Dvs. jeg ville gerne slette de to sidste tegn. Vi gør så det, at vi skriver to slette tegn, det er en stakpointer, kan man måske kalde det. Vi skriver så videre ACD.

Det gør arbejdet vældig rart, for det viser sig, at man bruger slette tegn overordentlig meget, idet man nemlig ikke alene bruger det i de tilfælde, hvor man slår fejl, men også i de tilfælde, hvor man bliver i tvivl, om man har slået rigtigt eller forkert. Man vil på en skrivemaskine gøre det, at man kigger op på arket og ser, hvad man har lavet. Det gør man ikke her. Man siger, slog jeg nu rigtigt, slet-tegn og så tilbage og skrive det engang til. Det kan gøre det lidt besværligt, når man senere skal læse det, men den, der sidder og transkriberer spekulerer ikke på, at det bliver den streng, der bliver produceret. Vedkommende, der transkriberer, tænker selvfølgelig på den streng, der er rigtig, og spekulerer ikke over, at der står nogle tegn imellem dem. Så kan der ske det, at man glemmer, hvor man er, og at man så siger, hov, hvor er jeg nu henne. Da ville det være rart, hvis man kunne sige, jeg vil gerne være heroppe, inden det foregående linieskift. D.v.s. at man kan trykke på en, der hedder slet-linie, som sidder nedenunder, og slet-linie fører stakpointeren tilbage til og med det sidste linieskift. Og der kan man altså trykke så mange gange, man vil. Så vil man få en fuldstændig datarepræsentation, og så kommer det an på, om man kan lave et program, der kan pille det fra hinanden igen.

Og så endelig er der noget, man af og til har brug for, og det er slet-blok. F.eks. hvis man siger: ~~xnej, det var noget møg. Jeg vil gerne slette til og med sidste blokstartx~~, kan man føre stakpointeren tilbage dertil ved hjælp af et slet-blok. Det ville være vældig rart, hvis man kunne bruge dette helt frit, og vi tænkte først på, at man også måtte kunne slette et slet-tegn, men da ville det være nemmere at skrive det sidste tegn end at slette slet-tegnet, ellers kunne man ikke komme over det, så ville det være en uendelig rekurs. Det kan man altså ikke, men i det øjeblik, man siger, nu vil jeg gerne slette de syv linier der, - nåhr nej, siger man her, det er nok noget møg alligevel, lad mig slette de to første

linier - så skal vi have et tegn, der betyder slet-linie - så siger man, nåhr nej, jeg vil kun slette den ene linie, så trykker man på et slet-tegn til. Og så er man videre igen. Det samme gælder blokke. Og reglen er altså, at slette tegn kan slette de andre, og slet-linie kan også slette et slet-blok. Det virker vældig naturligt, når man sidder i transkriptionssituationen. Når man skal lave et program, virker det nogen gange lidt underligt. Og der er visse ting, man skal tage stilling til f.eks. en symbolstreng af denne type: ABC slet-tegn slet-tegn CC slet-tegn slet-tegn slet-tegn. Den kan godt opfattes, når man sidder og transkribere. Selvfølgelig kan man udfra et program definere, hvad betydningen er. Men det er min erfaring, at en sådan symbolstreng ikke vil forekomme, fordi man da vil bruge slet-linie af psykologiske grunde.

Programmet er lavet således, at det arbejder sig baglæns igennem databåndet, og jeg mener, at man principielt kun kan analysere sådanne symbolstreng baglæns, men det er ikke rigtig bevist endnu. Ellers skal man altså gemme det - hvis man gerne vil have den forlæns indlæsning i en buffer som er uendelig stor, for ellers kan det være, man taber noget, og fejlen ved at gemme det i en buffer vil være den, at hvis der mangler et afsluttende symbol, så kommer systemet i en undefineret situation. Det kommer vores ikke, selv om der ikke er noget afsluttende symbol, når vi behandler det baglæns fortløbende tegn for tegn. Og så har det yderligere en fordel, at når man laver en transkription, så laver man en rulle, og næste dag vil man gerne kunne fortsætte den, så man ikke skal spole tilbage igen, derfor vil det være vældig naturligt at rulle strimler op baglæns, det forekommer ligetil for hulle damerne - ligesom når man optager på båndoptager - det er simpelthen den måde, som et bånd afleveres fra hullestuen på, hvis man ikke giver nogen instruktion om retningen.

Vanskelighederne ved xtranskription i blindex bliver fuldkommen overvundet ved at bruge denne teknik.

## Litteratur:

Bostrup, L. Produktionen af datastrimler, NYTs monografi nr. 6,  
duplikeret, København, Danmarks Pædagogiske Institut, 1968

Naur, P. A Manual of Gier algol 4  
København, Regnecentralen, 1967

Rattleff, J. og de Waal, B. Bloktrimning II  
duplikeret, København, Danmarks Lærerhøjskole, 1968

Rattleff, J. og de Waal, B. Sortering  
duplikeret, København, Danmarks Lærerhøjskole, 1967