TEKST NR 197 1990

Kontrafaktiske konditionaler i HOL

af:
Jesper Voetmann
Hans Oxvang Mortensen
Aleksander Høst-Madsen

vejleder: Stig Andur Pedersen

IMFUFA, Roskilde Universitetscenter
Postbox 260,4000 Roskilde

26. juni 1990

TEKSTER fra

IMFUFA ROSKILDE UNIVERSITETSCENTER
INSTITUT FOR STUDIET AF MATEMATIK OG FYSIK SAMT DERES FUNKTIONER I UNDERVISNING, FORSKNING OG ANVENDELSER

IMFUFA, Roskilde Universitetscenter, Postbox 260, 4000 Roskilde

KONTRAFAKTISKE KONDITIONALER I HOL

af: Jesper Voetmann, Hans Oxvang Mortensen, Alexander Høst-Madsen

Vejleder: Stig Andur Pedersen

IMFUFA tekst nr. 197/90 96 sider ISSN 0106-6242

Abstract

I projektet gennemgås teorien for kontrafaktiske konditionaler. Der indledes med afsnit om brug af forskellige ræsonneringsformer. Bl.a. deduktion, induktion og inexact reasoning. I de følgende afsnit opstilles en Gentzen-kalkyle i udsagnslogikken, på hvilken vi bygger en evaluator af kontrafaktiske konditionaler. Denne kritiseres, og der vises nogle simple egenskaber. Endelig implimenteres teorien i HOL88, et stort logikprogram. De grundliggende ideer i HOL gennemgås. Afslutningsvis er der en kritik af HOL, som logisk værktøj.

Indhold

Forord

Dette er et matematikprojekt fra Roskilde UniversitetsCenter's overbygning for matematik. Projektet skal dække kravene til modul 1 og modul 3.

Projektrapporten er i to dele. Først en teoretisk del, hvor vi redegør for ideen og matematikken bag de kontrafaktiske konditionaler. Dernæst en model-del, hvor vi redegør for implementeringen af denne teori. Til dette har vi brugt et logikprogram, HOL88. HOL arbejdet har været en stor del af projektet. Vi havde alle fra starten et ønske om at arbejde i HOL, men havde ingen anelser om hvor svært tilgængeligt det er. Resultatet blev en forholdsvis naiv evaluator af kontrafaktiske udsagn, baseret på udsagnslogikken. Projektet indeholder desuden en diskussion af HOL, som logisk værktøj.

Tak til Peter Øhrstrøm og Per Hasle for diskussioner og gode råd, tak til Jeff Paris for et inspirerende kursus i Inexact reasoning og tak til Stig Andur Pedersen for engageret vejledning.

Kapitel 1

Indledning

Datamaternes udbredelse har i dag gjort ekspertsystemer til et centralt forskningsfelt, og i tilknytning hertil er der sket en opblomstring i forskningen inden for bl.a. filosofi og psykologi. Der stræbes efter at udvikle en viden, der gør det muligt at indlægge slutningsregler i datamaskiner af en kvalitet, der gør dem i stand til at udføre brugbare ræsonnementer. Målestokken for denne kvalitet vil oftest være de ræsonnementer, som vi mennesker er i stand til at udføre. Den klassiske logik er langt fra tilstrækkelig til at matche vores dagligdags tænkning. Derfor foregår der et intenst arbejde med udviklingen af logik. Vi vil eksemplificere dette ved at vise den materielle implikations utilstrækkelighed og arbejdet for at formalisere kontrafaktiske konditionaler.

1.1 Materiel implikation

Fra den klassiske logik kendes den materielle implikation symboliseret ved $\Longrightarrow$. I dagligdags sprog gengives den ofte som 'hvis .. så..', hvilket kan være praktisk, men også upræcist, da '$\Longrightarrow$' og 'hvis .. så ..' ikke har den samme udtrykskraft.

Den materielle implikation har en entydigt defineret mening. Lad os se på den:

Hvis P og Q er 2 udsagn, der hver for sig enten kan være sande eller falske ($P, Q \in \{s, f\}$), så gælder der følgende sandhedstabel for implikationen $P \Longrightarrow Q$

Materiel implikation		
P	Q	$P \Longrightarrow Q$
s	s	s
s	f	f
f	s	s
f	f	s

Bemærk:

- den materielle implikation er altid sand med mindre **antecedenten**, P, er sand og **succedenten**, Q, er falsk.
- Den materielle implikations sandhedsværdi kan afklares alene ud fra kendskabet til P og Q.
- Hvis P er falsk vil implikationen altid være sand.

Den materielle implikation, $\Longrightarrow$, hører til mængden af logiske konnektiver, som blandt andet også rummer konjunktion ($\wedge$), disjunktion ($\vee$) og negation ($\neg$). De logiske konnektiver sammenkobler logiske udsagn (d.v.s. udsagn, der kan tildeles en sandhedsværdi) til sammensatte logiske udsagn.

1.2 Ræsonnering

At ræsonnere er at **drage konklusioner** og underbygge dem. Konklusioner drages på baggrund af **givne præmisser** og **gyldighed** er en relation mellem præmisserne og konklusionen. Fra et sæt af præmisser (logiske udsagn) kommer man til en konklusion ved hjælp af tilladte (gyldige) følgeslutninger. Hvis vi udfra præmisserne $P_1, P_2, \cdots, P_n$ kan slutte udsagnet Q vil vi skrive dette:

$$P_1, P_2, \cdots, P_n \vdash Q$$

Hvilke følgeslutninger, der er tilladte afhænger af hvilken ræsonneringsform, man anvender. Af den valgte ræsonneringsform følger, hvilken vægt man kan tillægge de dragne konklusioner. Et typisk eksempel på dette er forskellen mellem induktive og deduktive ræsonnementer. Vi vil ridse forskellene op og derefter kort fortælle hvad, der menes med inexact reasoning, der beskæftiger sig med gyldighed i situationer med mangelfulde præmiser.

1.2.1 Induktion

Induktion er ikke en sandhedsbevarende ræsonneringsform. Et velkendt eksempel: Ved almindelig færden i Danmark er alle de svaner, man møder, hvide. Vi kan derfor ved hjælp af induktion drage konklusionen: "Alle svaner er hvide". For at underbygge denne konklusion går vi herefter systematisk frem og registrerer samtlige svaner, vi ser. Jo flere svaner, der tilfredsstiller vores påstand, jo mere overbeviste bliver vi om sandheden af vores konklusion. Men helt sikre kan vi aldrig blive førend, vi har testet "ALLE" svaner - udtømt konklusionens genstandsområde. Selvom alle præmisserne til konklusionen er sande, er det ikke sikkert, at selve konklusionen er sand. Når man begrunder en induktiv konklusion, har man alene kvantiteten af observationer og fordelingen af disse at bygge på. Ved gyldighed af et induktivt ræsonnement, forstår vi at præmisserne kan give støtte til konklusionen, men ikke sikre den. Blot en enkelt observation der ikke opfylder vores påstand kan således falsificere konklusionen[1].

Der er forskel på empirisk og matematisk induktion. I matematikken benyttes induktion ofte til at bevise sætninger. Forskellen er, at man i matematikken er sikker på at induktionen gennemløber hele det genstandsområde, der er tale om, f.eks. de naturlige tal.

1.2.2 Deduktion

Deduktion er en sandhedsbevarende ræsonneringsform og et deduktivt ræsonnement er gyldigt i den forstand, at hvis præmisserne er sande, så er også konklusionen sand. Ved deduktion er det derfor nødvendigt at have en sand viden at udlede sine konklusioner fra, hvis man vil være sikker på konklusionernes sandhed. Præmisserne behøver ikke være sande. I matematiske beviser er det et yndet trik at tilføre ikke sande præmisser for at se hvilke kontradiktioner, der kan opnås. Den materielle implikation kan forbindes med tre gyldige slutninger ved deduktiv ræsonnering. Modus ponens, modus tolens og kædeslutningsreglen.

Modus ponens har følgende slutningsskema:

$$\frac{A, A \Longrightarrow B}{B}$$

[1]Der er faktisk observeret sorte svaner i Australien.

Denne skrivemåde angiver, at hvis præmisserne (som står over stregen): "A er sand" og "A implicerer B", så er det gyldigt at slutte konklusionen (som står under stregen): "B er sand".

Modus tolens har følgende slutningsskema:

$$\frac{\neg B, A \Longrightarrow B}{\neg A}$$

Hvis præmisserne er at "A implicerer B" og "B er falsk", så er den gyldige konklusion, at "A er falsk".

Kædeslutningsreglen har følgende slutningsskema:

$$\frac{A \Longrightarrow B, B \Longrightarrow C}{A \Longrightarrow C}$$

Hvis "A implicerer B" og "B implicerer C", så gælder det også at "A implicerer C". Denne regel kan udvides til en vilkårlig lang kæde af implikationer.

1.2.3 Ræsonnering med ufuldstændige eller modstridende data

Til tider mangler man information til at underbygge sine ræsonnementer. De fleste af vores dagligdags ræsonnementer er i virkeligheden ret dårligt underbyggede, og i nogle tilfælde bygger vores ræsonnementer på modstridende oplysninger. Betragt følgende eksempel:

1. Det er overskyet idag, og skyerne antyder, at det måske vil regne.
2. Hvis barometeret er højt betyder det som regel fint vejr.
3. Barometeret er rimeligt højt her til morgen.
4. Det bliver sikkert godt vejr idag.

Det ses, at der er mangel på viden. Dette antydes af udtryk som "rimeligt højt" og "sikkert godt". Endvidere er der modstridende oplysninger: skyer og højtryk. Disse forhold afskrækker os ikke i det daglige fra at lave følgeslutninger, selv om de ikke harmonerer helt med klassisk logik. Ræsonnering med modstridende data har man forsøgt at formalisere på forskellig vis. Eksempelvis kan nævnes Jeff Paris' studier indenfor *Inexact reasoning*. Jeff Paris arbejder med en vægtning af udsagnene, der ræsonneres på. De vægtede udsagn kan opdateres på forskellig vis f.eks. efter maximum entropi princippet.

Indenfor inexact reasoning beskæftiger man sig altså med måder at ræsonnere på, hvor konklusionerne ikke altid følger direkte af præmisserne. Dette kalder logikerne for usunde bevissystemer, mere om det senere.

Når der ræsonneres med ufuldstændige data (*Reasoning with incomplete information*), sker det tit på følgende vis: Hvis det plejer at være A og der ikke er særlig grund til at tro $\neg A$ så slut A (*default logic*). En anden tilgang er en *Closed world assumption*, hvor alt, hvad der ikke vides noget om, regnes for falsk.

Fælles for mange af logikkerne indenfor disse genrer er, at de ikke er **monotone**. Klassisk logik er monoton, hvilket vil sige, at hvis det er muligt udfra en mængde M af præmisser at udlede udsagnet A, så er det også muligt udfra en vilkårlig udvidelse af M. Vores dagligdags ræsonnering er ikke altid monoton. Når vi får nye oplysninger, der er i modstrid med hvad vi tidligere troede, vil vi som regel foretage en revurdering af vores viden. Udsagn, der før var sande, kan blive falske. Dette er ikke-monoton logik.

1.3 Kontra-faktiske konditionaler

Der er altså stor forskel mellem det deduktive ræsonnement og den måde mennesker i det daglige tænker og argumenterer på. Hvordan skal man f.eks. forstå en sætning som: "Hvis Pølle ikke havde giftet sig med Tut, så ville den borgerlige regering være faldet". Hvis vi formaliserer denne ytring ved hjælp af den materielle implikation, så er den trivielt sand, da Poul Schlüter giftede sig med Tut. Alligevel vil nogen mene at sætningen er interessant og sikkert kunne diskutere dens gyldighed.

Vores dagligdags sprog vrimler med denne type ræsonnering. Eksempel A: "Hvis ikke du kom for sent til middag, så ville maden have været varm". Eksempel B: "Hvis ikke Berlin muren var faldet, så ville elefanter have 2 snabler".

Man kan vel dårligt forestille sig en diskussion, der ikke rummer argumentation af denne type (A, selvfølgelig). Hvad er det der gør at eksempel A synes fornuftigt, mens eksempel B lyder som det rene nonsens? Det kunne se ud som om forskellen er, at der eksisterer et kausalt (årsagsmæssigt) forhold i eksempel A, hvad der ikke gør i B. Udsagn af typen som i eksemplerne kaldes kontra faktiske konditionaler. Et kontrafaktisk konditionale er et udsagn som "hvis P så Q", hvor antecedenten P enten vides eller formodes at være falsk ('i modsætning til fakta').

Det er ofte kausaliteten, der gør kontrafaktiske konditionaler interessante. Et godt eksempel har været behandlingen af katastrofen på Scandinavian Star. I forsøgene på at placere ansvaret og finde ud af hvem der skal betale, har de fleste af argumenterne været af formen "Hvis der havde været bedre skiltning og adgang til røgdykkerudstyr så ville mange mennesker være blevet reddet". Dette udsagn er kontrafaktisk i sin natur. Skiltningen var dårlig og røgdykkerudstyret var ikke hvor det skulle være, altså falske udsagn. Det interessante i denne sammenhæng er dog ikke sandhedsværdien af de enkelte udsagn, men opstillingen af en verden, der er magen til den faktiske, bortset fra skiltning og røgdykkerudstyr, og analysen af konsekvenserne i denne verden. En anden interessant ting ved de kontrafaktiske konditionaler er at de åbenbart har det med at slå ned på de unormale ting. Når man i dagligsproget benytter kontrafaktiske konditionaler er det næsten altid de væsentlige afvigelser fra det "normale", der lægges til grund for konditionalerne. Tag for eksempel sætningen: "Hvis purseren var blevet klippet dagen forinden, så ville flere passagerer være blevet reddet". Denne sætning vil man næppe høre, netop fordi afvigelsen (purserens hår) ikke er relevant i forhold til ulykken. Hvis man formaliserer den med den materielle implikation vil den få den samme sandhedsværdi som den forrige sætning. Forskellen er at ændringen i scenario ikke er interessant i det andet tilfælde. For at kunne vurdere blandt andet dette skal man skelne mellem de forskellige scenarier eller verdener, der dannes når man benytter kontrafaktiske konditionaler.

1.4 Mulige verdener

Mens vi angiver den materielle implikation med symbolet "$\Longrightarrow$", så vil vi angive et kontrafaktisk konditionale med "$\succ$ ". Ved den materielle implikation "$P_1 \wedge \cdots \wedge P_n \Longrightarrow Q$" kan vi afgøre sandhedsværdien af udsagnet alene ud fra kendskabet til de i implikationen indgående udsagns sandhedsværdi. Det kan vi ikke ved de kontrafaktiske konditionaler. Her er vi nødsaget til at inddrage alle de betingelser, der kan tænkes at påvirke vurderingen af gyldigheden af den kontrafaktiske slutning. Ved et kontrafaktisk konditionale skal vi altså forestille os udsagnet givet i en sammenhæng af flere betingelser. En sådan sammenhæng vil vi i det følgende kalde for en **verden**. For at være mere præcis, definerer vi en verden, som en mængde $\mathbf{S} = \{P_1, \cdots, P_n\}$, hvor P_i er et udsagn, der beskriver en egenskab ved verdenen.

> Debatten om gyldigheden af udsagnet "Hvis Pølle ikke havde giftet sig med Tut, så ville den borgerlige regering være faldet" afhænger således af hvilket verdensbillede den enkelte diskussionsdeltager har. Nogen vil mene at den megen presseomtale Schlüter fik med brylluppet var gunstigt for regeringen, - andre måske at det eneste betydningsfulde har været de politiske forlig, der er indgået på Christiansborg[2] o.s.v.

Lad der være givet en verden $\mathbf{S} = \{P_1, \cdots, P_n\}$. Vi ønsker nu at undersøge gyldigheden af det kontrafaktiske konditionale

$$(*)\ P_k \succ Q.$$

For at være kontrafaktisk må der gælde at P_k er (eller formodes at være) i modstrid med et eller flere af udsagnene i $\mathbf{S}$. (F.eks. kan $P_k = \neg P_1$.). Foreningsmængden af $\mathbf{S}$ og P_k, $(\mathbf{S} \cup \{P_k\})$ vil da være **inkonsistent**. En **konsistent** mængde er en mængde, hvoraf der ikke kan udledes kontradiktoriske udsagn (f.eks. $B \wedge \neg B$). Man kan sige, at det er en mængde af udsagn, der ikke er i modstrid med hinanden.

[2]Eksemplet kan ikke fuldstændigt dækkes af kontrafaktiske konditionaler. Der indgår tydeligvis også elementer af vurdering, der bedre ville kunne dækkes ved hjælp af teorier for 'inexact reasoning'.

Netop fordi P_k er kontrafaktisk forudsætter (*) den tænkelige eksistens af en verden, hvori P_k ikke er i modsætning til fakta, d.v.s. de øvrige præmisser. Undersøgelsen af det kontrafaktiske udsagns gyldighed, må derfor bygge på undersøgelsen af sådanne **mulige verdener**. Eller med andre ord: Lad **T** være en delmængde af **S**. For hvilke **T**, gælder det at foreningsmængden $\mathbf{T} \cup \{P_k\}$ er konsistent? Eller med helt andre ord: hvilke andre mulige verdener eller scenarier kan vi forestille os, hvori P_k er sand?

Hvis verdenen **S** består af n udsagn, vil der være 2^n delmængder af **S**. Selvom kun en del af disse vil føre til mulige verdener, må man normalt regne med, at der i denne mængde vil være mange, som er restriktioner af andre mulige verdener. For en nærmere undersøgelse er det alene de mulige verdener, **T**, der er **maksimalt lig** den faktiske verden **S**, der er interessante.

> Hvis vi tager eksemplet fra før med Scandinavian Star og skiltningen, så er den maksimalt lig verden, den faktiske blot med den ændring, at skiltning og røgdykkerudstyr var i orden. Den ændrede skiltning kan have givet andre problemer, som vi må må tage højde for, i opbygningen af den maksimalt lig verden. For eksempel kan skiltningen have givet anledning til kødannelser og nedtrampede personer, eller skiltningen kan have været vildledende. Dette skal med i den mulige verden.

'Maksimal lighed' kan vurderes efter kvantitative eller kvalitative kriterier. Når vi skal fjerne udsagn fra vores verden, er det så bedst at fjerne så få som muligt, eller er der nogle udsagn der bedre kan fjernes end andre ?

1.5 Gyldigheden af det kontrafaktiske konditionale

Når de mulige verdener er genereret står tilbage at vurdere selve konditionalet.

Givet en mulig verden **T**, hvor foreningsmængden $\mathbf{T} \cup \{P_k\}$ er konsistent, er undersøgelsen af det kontrafaktiske konditionale "$P_k \succ Q$"

reduceret til en undersøgelse af om Q følger af præmisserne i foreningsmængden.

> Tilbage til eksemplet: Når vi har genereret den nye verden hvor skiltningen og røgdykkerudstyret var i orden skal vi tage stilling til om der ville blive reddet flere passagerer i den verden. Hvis der blev reddet flere personer i alle de verdener, hvor skiltningen og røgdykkerudstyret var i orden, vil vi give det kontrafaktiske konditionale værdien sand.

Det er åbenlyst, at der i eksemplet er verdener, det er mere relevant at undersøge end andre. Oplysninger om de ansattes hårfarve, vil man ikke tillægge særlig stor betydning, mens naturlove og brandmyndighedernes faglige ekspertise ikke kan negligeres.

Om det kontrafaktiske konditionale $P_k \succ Q$ som helhed har gyldighed, kan vurderes ud fra om $P_k \Longrightarrow Q$ er sand i alle mulige verdener $\mathbf{T}$, som ligner tilstrækkeligt. Det er altså ikke nok, at der findes een mulig verden hvori $P_k \Longrightarrow Q$.

1.6 Eksempel

For at afrunde indledningen vil vi her til sidst give et eksempel på at anvendelse af kontrafaktisk ræsonnering i modsætning til deduktiv ræsonnering fører til forskellige resultater.

Lad os betragte følgende 2 beskrivelser af verden:

$$\mathbf{T1} = \{A, A \Longrightarrow B\} \text{ og } \mathbf{T2} = \{A, B\}$$

I klassisk logik vil de konklusioner, der kan udledes af **T1** og **T2** være identiske, da modus ponens anvendt på **T1** viser at B også er sand i **T1**. Men de kontrafaktiske konsekvenser af de 2 verdensbeskrivelser er forskellige. Hvis vi f.eks. betragter de mulige verdener, der kan udledes ved at tilføje det kontrafaktiske udsagn - $\neg B$ fås af

T2 den mulige verden $\{A\}$. Heraf kan udledes at $\neg B \succ A$.

T1 de mulige verdener $\{A\}$ og $\{A \Longrightarrow B\}$. Vi kan ikke som ved $T2$ udlede $\neg B \succ A$, da dette kontrafaktiske udsagn må forkastes ved undersøgelsen af den anden mulige verden $\{A \Longrightarrow B\}$ hvorfra vi ved brug af modus tolens kan slutte $\neg A$. $\neg B$ kan her kun være sand hvis $\neg A$ er sand (jævnfør sandhedstabellen for den materielle implikation). Når vi således kan udlede at $\neg B \succ A$ er et falsk kontrafaktisk konditionale med verdenen $T1$, kan vi slutte at $\neg(\neg B \succ A)$ er sand. Altså det modsatte af hvad, der kunne udledes fra $T2$. Det er således klart at de 2 verdener, $T1$ og $T2$, ikke er identiske ved en kontrafaktisk analyse.

Kapitel 2

Udsagnslogikken

Udsagnslogikken handler om forbindelser mellem udsagn. Udsagnene kan være de enkelte atomiske udsagn, eller de kan være sammensat af flere udsagn. Reglerne for hvorledes man kan sætte udsagn sammen kaldes syntaksregler og udsagnenes betydning bestemmes af semantikken. Hvilke udsagn der følger af andre udsagn er defineret i bevissystemet. Vi vil gennemgå to typer bevissystemer, Gentzen og Hilbert, med hovedvægten på Gentzenkalkylen.

2.1 Syntaks

De mindste elementer i sproget er givet ved sprogets alfabet. Disse elementer sættes sammen ved hjælp af syntaksregler og danner det fulde sprog.

Alfabet

1. En tællelig mængde U af udsagnslogiske symboler: $P_1, P_2, \cdots$
2. De logiske konnektiver: $\wedge$ (OG), $\vee$ (ELLER), $\Longrightarrow$ (IMPLIKATION), $\neg$ (NEGATION) og konstanten $\perp$ (FALSK).
3. Hjælpesymboler: "(" og ")" (paranteser).

Sproget

Sproget er den mindste mængde **L** af strenge over alfabetet sådan at:

1. Alle de udsagnslogiske symboler P_i og $\perp$ er i L, $i \in N$.
2. Hvis A er i **L**, så er $\neg A$ også i **L**.
3. Hvis A og B er i **L**, så er $(A \wedge B)$, $(A \vee B)$ og $(A \Longrightarrow B)$ også i **L**.
4. En streng er i **L** hvis og kun hvis den er dannet ved reglerne 1, 2 og 3.

Vi kalder elementerne i **L** for formler. Alle paranteserne behøver ikke blive sat. Vi indfører almindelige præcedens regler, hvor for eksempel $P_1 \wedge P_2 \Longrightarrow P3$ er kort for $((P_1 \wedge P_2) \Longrightarrow P3)$. Vi har hermed fuldstændig fastlagt vores sprog og går nu over til at give det en mening eller semantik.

2.2 Semantik

Hver formel i **L** gives en sandhedsværdi udfra følgende tabel.

Sandhedsværdi for formler i **L**					
P_1	P_2	$\neg P_1$	$P_1 \wedge P_2$	$P_1 \vee P_2$	$P_1 \Longrightarrow P_2$
s	s	f	s	s	s
s	f	f	f	s	f
f	s	s	f	s	s
f	f	s	f	f	s

Vi vil bruge bogstaverne $A, B, C, \cdots$ for vilkårlige formler fra sproget **L**.

En variabeltilskrivning, v, defineres nu som en funktion fra mængden af udsagnssymboler U ind i **bool**. **bool** er mængden $\{s, f\}$. Efter at udsagnsvariablene er tildelt en værdi er det muligt at give alle formlerne i **L** en værdi ved funktionen v' fra **L** ind i **bool**. Et eksempel: $v'(P_1 \wedge P_2)$ får værdien s hvis og kun hvis v har tildelt både P_1 og P_2 værdien s. Dette vil vi fremover skrive som: $v \models (A)$, for en given formel A. Vi siger at v tilfredsstiller A. Hvis det ikke er tilfældet falsificerer v A. Hvis en formel A er sand under alle variabeltilskrivninger siger vi, at A er **valid**, og skriver

$$\models A$$

Givet en mængde $\mathbf{\Gamma}$ af formler siger vi at A er en semantisk konsekvens af $\mathbf{\Gamma}$ hvis alle variabeltilskrivninger, der gør elementerne i $\mathbf{\Gamma}$ sande, også gør A sand. Dette skriver vi:

$$\mathbf{\Gamma} \models A$$

2.3 Bevissystemer

Bevissystemets rolle er at formalisere, hvad vi forstår ved, at en formel logisk følger af en eller flere andre formler. Hvis det kan vises, at en formel følger af den tomme mængde, siges den at være et **teorem**. Der er mindst 2 forskellige typer bevissystemer, hver med deres fordele og ulemper. Det første, vi vil kigge på, er et såkaldt Hilbert-type bevissystem. Systemet bygger på en mængde af aksiomer og en eller flere slutningsregler. Ideen i dette system er, at man tager udgangspunkt i aksiomerne og ved hjælp af slutningsreglerne deducerer sig frem til det eller de teoremer man ønsker at vise. Dette kaldes også "forward proof". Den anden type kaldes Gentzen typen. Her er ideen, at tage udgangspunkt i den formel man ønsker at vise. Man har en række slutningsregler til rådighed, som alle mindsker kompleksiteten i de formler, som de benyttes på. Disse regler benyttes i mere eller mindre tilfældig rækkefølge, indtil aksiomerne nås, hvis formlen er et teorem. Dette kaldes "backward proof". Gentzenreglerne kan også benyttes til forward proof, men det er sjældent hensigtsmæssigt. Inden vi går over til at beskrive bevissystemer, skal vi lige nævne et par grundlæggende resultater vedrørende udsagnslogikken.

1) **Deduktionsteoremet:** Hvis $\mathbf{\Gamma}$ er en mængde af formler, A og B er vilkårlige formler og $\mathbf{\Gamma}, A \vdash B$, så gælder $\mathbf{\Gamma} \vdash A \Longrightarrow B$. Specielt, hvis $A \vdash B$ så $\vdash A \Longrightarrow B$.

2) **Gödels fuldstændighedssætning:** I udsagnslogikken er mængden af teoremer præcis lig med mængden af logisk valide formler (tautologier). Dette kan vi også skrive:

$$\vdash A \Longleftrightarrow \models A$$

3) **Afgørlighed af udsagnslogikken:** Givet en vilkårlig udsagnslogisk formel, kan det afgøres om formlen er valid eller ej.

De 2 første sætninger gælder også for førsteordens prædikatlogik. Den tredje gælder dog ikke, den afløses af

4) **Church sætning:** En førsteordens prædikatlogisk formel er semiafgørlig, forstået på den måde, at hvis den er valid findes der en procedure, der kan afgøre det i et endeligt antal skridt, men hvis formlen er falsificerbar vil proceduren måske fortsætte evigt.

2.3.1 Hilbert-typen

Bevissystemet vi vil vise her, virker på et sprog med færre konnektiver end det vi allerede har beskrevet. Det gør dog ikke så meget, da to konnektiver er rigeligt, hvis de vælges rigtigt. Ud fra $\wedge$ og $\neg$, $\vee$ og $\neg$ eller $\Longrightarrow$ og $\neg$ kan alle konnektiver dannes. Vi vil kun benytte konnektiverne $\Longrightarrow$ og $\neg$ og de to forkortelsesregler :

1. $(A \wedge B)$ svarer til $\neg(A \Longrightarrow \neg B)$
2. $(A \vee B)$ svarer til $(\neg A) \Longrightarrow B$

Det sprog, vi nu betragter, er det samme som det vi startede med at definere. Forstået på den måde, at vi ved hjælp af de to forkortelses regler, kan "oversætte" alle formler fra det gamle til det nye. Her følger så bevissystemet for vores "nye" sprog.

Hvis A, B og C er vilkårlige formler i vores sprog, så er de følgende sætninger aksiomer:

A1 $(A \Longrightarrow (B \Longrightarrow A))$

A2 $((A \Longrightarrow (B \Longrightarrow C)) \Longrightarrow ((A \Longrightarrow B) \Longrightarrow (B \Longrightarrow C)))$

A3 $((\neg B \Longrightarrow \neg C) \Longrightarrow ((\neg B \Longrightarrow C) \Longrightarrow B))$

Den eneste slutningsregel i dette system er modus ponens, altså at man fra A og $A \Longrightarrow B$ kan slutte B.

Dette bevissystem er fuldstændigt, hvilket vil sige at alle tautologier er teoremer. Eller sagt på en anden måde, alle sande formler kan bevises. Desude er det konsistent; der eksisterer ikke en formel A sådan at både A og $\neg A$ er teoremer.

2.3.2 Gentzen-typen

Vi vender nu tilbage til det oprindelige sprog med alle konnektiverne. Denne gang vil vi også lave en smule om på formlerne, inden vi forsøger at bevise dem. Vi betragter såkaldte sekventer.

DEFINITION 1
En **sekvent** er et par (Δ, Θ) hvor $\Delta =< A_1, A_2, \cdots, A_m >$ er en endelig sekvens af vilkårlige formler fra **L** og $\Theta =< B_1 \vee B_2 \vee \cdots \vee B_n >$ er en disjunktion af formler fra **L**. •○•

Hvis man ønsker at vise at en given formel B er et teorem dannes sekventen

$$<>\rightarrow< B >$$

og hvis man ønsker at undersøge om B følger logisk af formlerne $A_1, A_2, \cdots, A_m$ dannes sekventen

$$< A_1, A_2, \cdots, A_m >\rightarrow B.$$

Pilen, der indgår i sekventerne, er ikke med i sproget **L**. Pilen skal intuitivt forstås som implikation, dens betydning er fastlagt nedenfor. Når sekventen er dannet begynder man at bruge slutningsreglerne. I Gentzenkalkylen er der otte slutningsregler. Slutningsreglerne falder naturligt i to kategorier, nemlig dem der virker på venstresiden, og dem der virker på højresiden. De virker på den måde at de hele tiden mindsker kompleksiteten i sekventen. Efterhånden som kompleksiteten mindskes vil det vise sig om sekventen kan falsificeres.

Det bevissystem vi præsenterer her er en tunet udgave af det originale Gentzen-system. Grunden til, at vi har ændret det er, at vi i HOL kun kan arbejde med en enkelt formel på højresiden af sekventer. Dette har vi afhjulpet, ved at benytte disjunktioner i stedet for kommaer. Den eller de sekventer, der står over linjen i de respektive regler kalder vi præmisser og alt det under linjerne kalder vi konklusioner.

DEFINITION **2**
S1'erne angiver vilkårlige endelige mængder af formler og **S2**'erne angiver vilkårlige endelige disjunktioner af formler. Det der står over linjerne kalder vi præmisser, det der står under linjerne kalder vi konklusioner.

GENTZEN REGLERNE

- VENSTRE DISJUNKTION

$$\frac{A, \mathbf{S1} \rightarrow \mathbf{S2} \quad B, \mathbf{S1} \rightarrow \mathbf{S2}}{A \vee B, \mathbf{S1} \rightarrow \mathbf{S2}}$$

- VENSTRE IMPLIKATION

$$\frac{\mathbf{S1} \rightarrow A \vee \mathbf{S2} \quad B, \mathbf{S1} \rightarrow \mathbf{S2}}{\mathbf{S1}, A \Longrightarrow B \rightarrow \mathbf{S2}}$$

- VENSTRE KONJUNKTION

$$\frac{\mathbf{S1}, A, B \rightarrow \mathbf{S2}}{\mathbf{S1}, A \wedge B \rightarrow \mathbf{S2}}$$

- VENSTRE NEGATION

$$\frac{\mathbf{S1} \rightarrow A \vee \mathbf{S2}}{\mathbf{S1}, \neg A \rightarrow \mathbf{S2}}$$

- HØJRE KONJUNKTION

$$\frac{\mathbf{S1} \rightarrow A \vee \mathbf{S2} \quad \mathbf{S1} \rightarrow B \vee \mathbf{S2}}{\mathbf{S1} \rightarrow (A \wedge B) \vee \mathbf{S2}}$$

- HØJRE IMPLIKATION

$$\frac{\mathbf{S1}, A \rightarrow B \vee \mathbf{S2}}{\mathbf{S1} \rightarrow (A \Longrightarrow B) \vee \mathbf{S2}}$$

- HØJRE NEGATION

$$\frac{\mathbf{S1}, A \rightarrow \mathbf{S2}}{\mathbf{S1} \rightarrow \neg A \vee \mathbf{S2}}$$

En variabeltilskrivning v gør en sekvent

$$< A_1, A_2, \cdots, A_m > \rightarrow < B_1 \vee B_2 \vee \cdots \vee B_n >$$

sand hvis og kun hvis

$$v \models (A_1 \wedge A_2 \wedge \cdots \wedge A_m) \Longrightarrow (B_1 \vee B_2 \vee \cdots \vee B_n)$$

Sekventen er valid, hvis det gælder for alle variabeltilskrivninger. Sekventen kan falsificeres hvis og kun hvis, der eksisterer en variabeltilskrivning så

$$v \models (A_1 \wedge A_2 \wedge \cdots \wedge A_m) \wedge (\neg B_1 \wedge \neg B_2 \wedge \cdots \wedge \neg B_n)$$

•○•

LEMMA 0
For hver af reglerne i Definition 2 gælder der, at en variabeltilskrivning v falsificerer sekventen, der optræder som konklusion i reglen hvis og kun hvis v falsificerer mindst en af sekventerne, der optræder som præmisser. Tilsvarende gør v konklusionen i en regel sand hvis og kun hvis v gør alle reglens præmisser sande. •○•

Bevis: Beviset består i at tjekke sandhedstabellerne for de logiske konnektiver. Vi beviser her reglen for implikation på venstresiden, og overlader resten som opgave til læseren.

$\Longrightarrow$ venstre reglen:

$$\frac{\mathbf{S1} \rightarrow A \vee \mathbf{S2} \quad B, \mathbf{S1} \rightarrow \mathbf{S2}}{\mathbf{S1}, A \Longrightarrow B \rightarrow \mathbf{S2}}$$

For en vilkårlig variabeltilskrivning v gælder der, at v falsificerer konklusionen hvis og kun hvis v gør alle formlerne i **S1** sande, gør $(A \Longrightarrow B)$ sand og falsificerer formlerne i **S2**. For at gøre $(A \Longrightarrow B)$ sand skal v enten falsificere A eller gøre B sand. v skal altså, for at falsificere konklusionen, enten

1) falsificere både A og **S2** og gøre formlerne i **S1** sande, eller

2) gøre B, **S1** sande og falsificere **S2**.

Kravene 1) og 2) er netop kravene til at falsificere reglens præmisser. Konklusion og præmisser bliver altså falsificerede af de samme variabeltilskrivninger, hvilket skulle vises. •○•

Aksiomerne i Gentzenkalkylen er alle de sekventer, der indeholder den samme formel på højre- og venstresiden. Vi har også tunet mængden af aksiomer en smule, for at kunne arbejde med HOL. I vores kalkyle er aksiomerne vilkårlige sekventer $\Delta \rightarrow \Theta$ sådan at Δ indeholder mindst en formel, der er magen til en del af den eventuelle disjunktion på højresiden. Eller med andre ord, alle aksiomerne i vores kalkyle er sekventer, der kan skrives på formen:

$$< A_1, A_2, \cdots, X, \cdots, A_m > \rightarrow < B_1 \vee B_2 \vee \cdots \vee X \vee \cdots \vee B_n >$$

LEMMA 1
Ingen aksiomer kan falsificeres •○•

Bevis: Hvis man skal falsificere et aksiom skulle man gøre en formel på venstresiden sand og den samme formel falsk på højresiden, hvilket er umuligt. •○•

For at vise sundheden og fuldstændighed af vores Gentzenkalkyle vil vi nu formelt definere hvad et bevis og et bevistræ er. Vi benytter den samme "trænotation" som Gallier.

Mængden af bevistræer er den mindste mængde af træer indeholdende alle enkelt-knude bevistræer, (der blot består af en enkelt formel) hvor formlen er et aksiom. Desuden er mængden lukket under reglerne fra definition 2 på følgende måde:

(1) For et vilkårligt bevistræ T1 hvis rod er en sekvent $\Delta \rightarrow \Theta$ og for en vilkårlig instans af en een-præmis regel med præmis $\Delta \rightarrow \Theta$ og konklusion $\Delta' \rightarrow \Theta'$, er træet T hvis rod er $\Delta' \rightarrow \Theta'$ og hvis subtræ T/1 er magen til T1, et bevistræ.

(2) For to vilkårlige bevistræer T1 og T2 hvis rødder er sekventer $\Delta \rightarrow \Theta$ og $\Delta' \rightarrow \Theta'$ og for en vilkårlig instans af en to-præmis regel med præmisser $\Delta \rightarrow \Theta$ og $\Delta' \rightarrow \Theta'$ og konklusion $\Delta'' \rightarrow \Theta''$, er træet T hvis rod er sekventen $\Delta'' \rightarrow \Theta''$ og hvis subtræer T/1 og T/2 er magen til T1 og T2, et bevistræ.

Mængden af deduktionstræer defineres induktivt som den mindste mængde af træer, der indeholder alle enkelt-knude træer (ikke nødvendigvis aksiomer) og er lukkede under (1) og (2) som ovenfor. En sekvent $\Delta \rightarrow \Theta$ er beviselig, hvis der eksisterer et bevistræ for den. Det skriver vi

$$\vdash \Delta \rightarrow \Theta.$$

Vi kan nu vise at vores kalkyle er **sund**, altså at hvis en sekvent kan bevises, er den valid.

SÆTNING 1 Vores kalkyle er sund •○•

Bevis: Dette vises ved induktion på bevistræer. Ved lemma 1 har vi at alle enkelt-knude bevistræer (aksiomer) er valide. Herefter er der to tilfælde, at lave induktion over.

1.tilfælde: Roden i bevistræet T har en enkelt knude lige over sig. I dette tilfælde kommer T fra et træ T1 og en instans af en regel af formen:

$$\frac{\mathbf{S1}}{\mathbf{S2}}$$

Ved induktionshypotesen er **S1** valid. Her ved vi fra lemma 0 at **S1** er valid hvis og kun hvis **S2** er valid. Sætning 1 holder.

2.tilfælde: Roden i bevistræet har to knuder lige over sig. I dette tilfælde stammer T fra to bevistræer T1 og T2, og en instans af en regel:

$$\frac{\mathbf{S1} \quad \mathbf{S2}}{\mathbf{S3}}$$

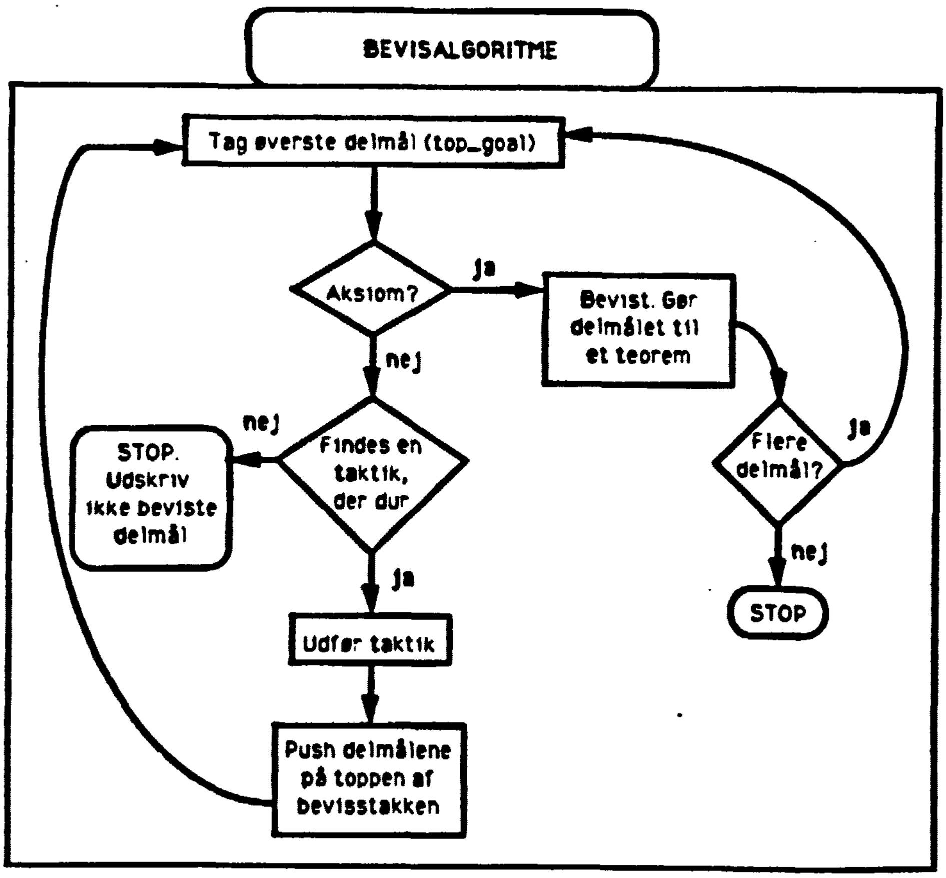
Ved induktionshypotesen ved vi at både **S1** og **S2** er valide. Fra lemma 0 ved vi desuden **S3** er valid hvis og kun hvis **S1** og **S2** er valide. Sætning 1 holder, hvilket skulle vises. •○•

Vi vil nu vise, at vores kalkyle også er komplet, altså at vi er i stand til at vise alle valide sætninger i den. Vi starter med at definere **kompleksitet** i en given sekvent, som antallet af logiske konnektiver der optræder i sekventen, undtagen eventuelle disjunktioner på højresiden.

LEMMA 3
Alle gentzenreglerne mindsker kompleksiteten. •○•

Bevis: Ses let •○•

Beviset for fuldstændighed bygger på den algoritme, vi har implementeret i HOL. Her ses en skitse af vores algoritme (inputtet er en sekvent):



Algoritmen virker dybde først og vil altid standse. Hvis inputsekventen er et aksiom, stopper algoritmen med det samme. Hvis inputsekventen ikke er et aksiom, benytter algoritmen en gentzenregel, der nedsætter kompleksiteten. Herefter opnås et eller to delmål, der tjekkes. hvis delmålene er aksiomer er inputtet bevist og algoritmen stopper. hvis det ikke er tilfældet benyttes gentzenreglerne påny. Dette fortsætter, indtil kompleksiteten er reduceret fuldstændig, og algoritmen stopper, da sekventerne kun indeholder et endeligt antal formler.

Kapitel 3

KFK matematisk afsnit

Vi vil i dette afsnit kort gennemgå nogle matematiske egenskaber som en teori for kontrafaktiske konditionaler bør besidde.

Vi vil i det følgende beskæftige os med teorien som Matthew L. Ginsberg fremlagde den i 1986. Desuden henvises til Stalnaker 1968.

Kapitlet handler primært om udsagnslogikken, idet visse komplikationer opstår, når man går op til førsteordenslogikken, jvnf. f.eks. Church sætning. Dette skitseres kort i slutningen af kapitlet.

Problemet er, som bekendt, at evaluere en sætning som f.eks. "Hvis muren ikke var faldet, ville USA beholde sine styrker i Europa". I Klassisk logik kan dette udtrykkes som

$$P \Longrightarrow Q$$

hvor P står for "muren er ikke faldet", og Q står for "USA beholder sine styrker i Europa". Men eftersom muren faktisk ER faldet, dvs. P er falsk, vil sætningen være trivielt sand, fuldstændigt uafhængigt af hvad Q beskriver. Man kunne med lige ret hævde det modsatte,

$$P \Longrightarrow \neg Q$$

eftersom også denne er trivielt sand. Som det ses, er den materielle implikation ($\Longrightarrow$) ikke velegnet til at evaluere sætninger som disse, hvor betingelsen (P) er falsk. Vi vil derfor indføre en anden slags implikation, hvor vi nærmere overvejer sandheden af disse sætninger. Denne implikation (Den kontrafaktiske implikation ($\succ$)) bør ikke være trivielt sand, blot fordi betingelsen er falsk. Sætningen fra før, kan nu skrives som

$$P \succ Q$$

og er nu ikke triviel. Den kan enten være sand eller falsk, og vi bliver nød til at tænke over svaret. Som sandhedstavle skrives:

Materiel vs kontrafaktisk			
P	Q	$P \Longrightarrow Q$	$P \succ Q$
s	s	s	s
s	f	f	f
f	s	s	bør overvejes
f	f	s	bør overvejes

3.1 Ønsker til konditionalet

Der er selvfølgelig mange måder at overveje konditionaler hvor antecedenten er falsk. Vi vil i det følgende formalisere vores ønsker til det kontrafaktiske konditionale i 5 punkter.

1. $(P \succ Q) \Longrightarrow (P \Longrightarrow Q)$. Denne relation følger direkte af ovenstående sandhedstavle. Det ses, at den kontrafaktiske implikation er mindst lige så stærk som materielle.

   Der er flere måder hvorpå man kan ønske sig, at relationen $\succ$ adskiller sig fra $\Longrightarrow$. Primært begrundelsen for af indføre den:

2. $\neg((P \succ Q) \land (P \succ \neg Q))$

3. $\succ$ er ikke monoton, dvs. man kan ikke udfra $P \succ Q$ slutte at $(P \land R) \succ Q$. F.eks. kan man ikke slutte

```
"Hvis sætningen:

        "Hvis muren ikke var faldet, ville USA
```

```
        beholde sine styrker i Europa."

    er sand, så er sætningen:

        "Hvis muren ikke var faldet,
        og Gorbatjov også var præsident i USA,
        ville USA beholde sine styrker i Europa."

    også sand."
```

Der er naturligvis også eksempler hvor $\succ$ opfører sig monotont, men vi kan ikke satse på det. Dette stemmer fint overens med punkt (1).

4. $\succ$ er ikke transitiv, dvs. man kan ikke udfra $P \succ Q$ og $Q \succ R$ slutte at $P \succ R$. I vores eksempel kan man altså ikke slutte

```
    "Hvis sætningen:

        "Hvis muren ikke var faldet, ville USA
        beholde sine styrker i Europa."

    er sand, og sætningen:

        "Hvis USA beholder sine styrker i Europa,
        så skal de hjælpe med at rive muren ned."

    er sand, er sætningen:

        "Hvis muren ikke var faldet,
        så skal USA's styrker hjælpe med at rive
        muren ned."

    også sand."
```

5. Kontraposition bør ikke være en gyldig slutningsregel for $\succ$, dvs. at man ikke udfra $P \succ Q$ kan slutte $\neg Q \succ \neg P$. Dette er også klart, hvis man undersøger eksemplet.

3.2 Algoritme til evaluering af KFK

Vi vil nu opstille en algoritme for at evaluere kontrafaktiske udsagn $P \succ Q$. Dette gøres ved at forestille sig en konsistent mulig verden, hvor udsagnet P er sandt, og derefter evaluere udsagnet Q i denne verden.

Lad **S** betegne den oprindelige verden, som det kontrafaktiske udsagn skal evalueres i. Denne verden betragtes som en mængde af informationer (udsagn) som vides sande. **S** kunne f.eks. være en række data i et databasesystem. Vi ønsker at evaluere udsagnet

$$(*) \quad \mathbf{S} \models P \succ Q$$

ved at finde en mulig verden **T**, hvor $\mathbf{T} \not\models \neg P$, for derefter at undersøge udsagnet

$$\mathbf{T} \cup \{P\} \models Q$$

Hvis dette er sandt, er det kontrafaktiske udsagn også sandt, ellers er det falsk.

Naturligvis bør man stille visse krav til dette **T**. Det skal ligne den oprindelige verden **S** så meget som muligt. Stalnaker udtrykker det på denne måde:

> "The informal truth conditions that were suggested above required that the world selected *differ minimally* from the actual world. This implies, first, that there are no differences between the actual world and the selected world except those that are required, implicitly or explicitly, by the antecedent".

Problemet er nu at finde dette **T**, hvilket desværre er nemmere sagt end gjort. Det er dog to særtilfælde hvor svaret er indlysende.

a. Hvis $P \in \mathbf{S}$ eller $\mathbf{S} \models P$, kan vi betragte $\succ$ som en almindelig implikation, og udtrykket kan reduceres til

$$\mathbf{S} \models Q$$

som vi nemt kan evaluere. I dette tilfælde kan vi altså sætte $\mathbf{T} = \mathbf{S}$.

b. Hvis $\mathbf{S} \not\models \neg P$, kan vi sætte $\mathbf{T} = \mathbf{S}$, eftersom denne ikke bliver inkonsistent når $\{P\}$ føjes til. Denne verden må nødvendigvis være den som ligner $\mathbf{S}$ mest, og hvor P ikke kan vises falsk.

Det sidste, og interessante tilfælde er det hvor $\mathbf{S} \models \neg P$, det tilfælde hvor (*) bliver **kontrafaktisk**. I dette tilfælde kan man ikke, som i tilfælde (a) og (b) blot sætte $\mathbf{T} = \mathbf{S}$, eftersom man da ville få:

$$\mathbf{T} \cup \{P\} \models P \quad og \quad \mathbf{T} \cup \{P\} \models \neg P$$

og $\mathbf{T} \cup \{P\}$ vil være inkonsistent. Eftersom dette er utilfredsstillende, bliver man nød til at gøre lidt mere ved $\mathbf{T}$, så $\mathbf{T} \cup \{P\}$ bliver konsistent. Vi må rydde lidt op i $\mathbf{T}$, og pille de udsagn ud som gør dette inkonsistent. Men dette kan normalt gøres på op til flere måder. Hvis vi f.eks. ser på verdenen

$$\mathbf{S} = \{A, A \Longrightarrow B, B \Longrightarrow C, C \vee D\}$$

og vi ønsker at evaluere udsagnet

$$(**) \quad \mathbf{S} \models \neg C \succ D$$

kan

$$\mathbf{S} \cup \{\neg C\} = \{A, A \Longrightarrow B, B \Longrightarrow C, C \vee D, \neg C\}$$

gøres konsistent, således at man stadig kan udlede $\neg C$, på 14 forskellige måder. Men hvorledes afgøres, hvilken af de 14 verdener, der ligner den oprindelige mest? Vi kan hurtigt overbevise os om, at nogle af de 14 ligner $\mathbf{S}$ mere end andre. F.eks vil en verden som $\mathbf{T}' = \{A\}$ ligne $\mathbf{S}$ mindre end $\mathbf{T}'' = \{A, B \Longrightarrow C, C \vee D\}$, af den simple grund at $\mathbf{T}'$ er en delmængde af $\mathbf{T}''$. På denne måde kan vi, med den almindelige mængderelation forkaste en hel del af disse verdener som uinteressante, og kigge på de, med den normale mængderelation "$\subseteq$", maksimale mulige verdener. I vores eksempel er der tre af disse, nemlig:

- $\mathbf{T}_1 = \{A \Longrightarrow B, B \Longrightarrow C, C \vee D\}$

- $\mathbf{T}_2 = \{A, B \Longrightarrow C, C \vee D\}$ og
- $\mathbf{T}_3 = \{A, A \Longrightarrow B, C \vee D\}$

Hvilken af de tre der ligner **S** mest, er svært at afgøre, men fælles for dem er, at man udfra <u>hver</u> af dem, tilsat udsagnet om $\neg C$, kan udlede konklusionen D. Det er derfor rimeligt at hævde, at (**) er sandt. Generelt defineres :

DEFINITION

Givet udsagnet $P \succ Q$, lad da $\mathbf{W}(P, \mathbf{S})$ betegne mængden af maksimale mulige verdener

$$\mathbf{W}(P, \mathbf{S}) = \{\mathbf{T} \subseteq \mathbf{S} | \mathbf{T} \not\models \neg P \quad og \quad (\mathbf{T} \subset \mathbf{V} \subseteq \mathbf{S}) \Longrightarrow \mathbf{V} \models \neg P\}$$

Udsagnet (*) defineres nu som værende sandt, hvis og kun hvis

1. $\mathbf{W}(P, \mathbf{S}) = \emptyset$
   eller
2. $\forall \mathbf{T} \in \mathbf{W}(P, \mathbf{S}) : \quad \mathbf{T} \cup \{P\} \models Q$
   ellers er (*) falsk.

Begge krav kan diskuteres. Det første sikrer, at hvis betingelsen P er fuldstændigt absurd i **S**, dvs. man overhovedet ikke kan forestille sig en verden hvor P er sand, bliver det kontrafaktiske udsagn sandt. I dette tilfælde, vil $\succ$ derfor virke ligesom $\Longrightarrow$ (de vil begge være sande), hvilket retfærdiggør krav (1) fra forrige afsnit. desværre holder (2) så ikke i alle tilfælde. Dette vender vi tilbage til i næste afsnit.

Krav nummer to kan kritiseres for at være lige lovlig strengt. Mængden $\mathbf{W}(P, \mathbf{S})$ kan let gå hen og blive en meget stor mængde, især hvis **S** er stor. Jo mere vi ved om **S**, jo mindre sandsynlighed er der for, at et kontrafaktisk konditionale i **S** er sandt.

Et andet kritikpunkt er, at selv om alle $\mathbf{T} \in \mathbf{W}(P, \mathbf{S})$ er maksimale med den almindelige mængderelation, behøver de ikke alle at "ligne" (i Stalnakers forstand) **S** maksimalt.

Ginsberg giver et eksempel. Han ser på en verden

$$\mathbf{S} = \{A, B, C, A \wedge B \Longrightarrow \neg P, B \wedge C \Longrightarrow \neg P\}$$

Hvis vi undersøger $\mathbf{W}(P, \mathbf{S})$ finder vi bl.a. de to elementer:

$$\mathbf{T}_1 = \{A, C, A \wedge B \Longrightarrow \neg P, B \wedge C \Longrightarrow \neg P\}$$

$$\mathbf{T}_2 = \{A, B, C\}$$

Her mener Ginsberg at $\mathbf{T}_1$ ligner $\mathbf{S}$ mere end $\mathbf{T}_2$ gør. Han gør ikke nærmere rede for hvordan "ligner" ordningen ser ud, men skriver dog :

> "In general, however, it appears that worlds of interest correspond to changing specific situational information (A, B, and C above) rather than to removing rules of inference (the implications)".

Men hvis nogle elementer i $\mathbf{W}(P, \mathbf{S})$ er bedre end andre, hvorfor skal vi så beskæftige os med de dårlige?

Ginsberg kommer ud over problemet, ved at indføre et lidt guddommeligt "badworld" prædikat $B(\mathbf{T})$, som automatisk luger de dårlige verdener ud af mængden $\mathbf{W}(P, \mathbf{S})$. Hvis $B(\mathbf{T})$ er sandt når $\mathbf{T}$ er en dårlig verden, og falsk hvis $\mathbf{T}$ er OK, får vi:

$$\mathbf{W}'(P, \mathbf{S}) = \{\mathbf{T} \in \mathbf{W}(P, \mathbf{S}) | \neg B(\mathbf{T})\}$$

og krav nummer to fra før, omformuleres til

$$2. \quad \forall \mathbf{T} \in \mathbf{W}'(P, \mathbf{S}) : \quad \mathbf{T} \cup \{P\} \models Q$$

Hermed opnås et svagere, og formentlig bedre krav. Badworld prædikatet kan umiddelbart virke lidt mystisk, men det er bestemt ikke ubrugeligt. F. eks. vil det ofte være sådan, at visse udsagn har større informationsværdi end andre, og man derfor ikke ønsker at fjerne dem i de mulige verdener.

Hvis man kigger på verdenen

```
S = {"Der er kaffe i koppen",
     "Jorden eksisterer",...}
```

vil man formentlig være mere fristet til at glemme det første udsagn end det andet. Badworld prædikatet kunne her luge de mulige verdener ud, hvor vi ikke vidste om Jorden eksisterede. Bemærk venligst, at Badworldprædikatet kun tager et argument, nemlig en af delmængderne af verdenen **S**. Den er ikke afhængig af betingelsen P.

3.3 Ønskerne en gang til

1. $(P \succ Q) \Longrightarrow (P \Longrightarrow Q)$ Dette udsagn er altid sandt. Hvis succedenten er falsk, så er antecedenten det også.

2. $\neg((P \succ Q) \wedge (P \succ \neg Q))$. De mulige verdener, der bliver genereret, er konsistente. Derfor er det ikke muligt at udlede både Q og $\neg Q$. Der eksisterer imidlertid et specialtilfælde. Hvis mængden af mulige verdener er tom, vil konditionalet altid være sandt. Problemet er at vurdere et udsagn, hvor præmisserne er fuldstændig i modstrid med vores verden.

3. Vores logik er ikke monoton. Hvis vi kan vise et udsagn i en given verden, er det ikke sikkert, vi kan vise den i en udvidelse af verdenen. Dette kommer af, at udsagnet skal være sandt i alle maksimale verdener. Et eksempel:

$$\mathbf{T} = \{\neg P, P \wedge N \Longrightarrow Q, N, \neg(N \wedge R)\}$$

   I denne verden **T** kan man vise $P \succ Q$, men ikke $P \wedge R \succ Q$.

4. Vores logik er ikke transitiv, hvilket følgende er et eksempel på.

$$\mathbf{T} = \{P \Longrightarrow Q, A, A \Longrightarrow R, \neg P \Longrightarrow R, \neg(P \wedge R)\}$$

   I denne verden **T** kan man vise $P \succ Q$ og $Q \succ R$, men derimod kan man ikke vise $P \succ R$.

5. Og sidst et eksempel på uanvendeligheden af kontraposition.

$$\mathbf{T} = \{A, P, P \wedge A \Longrightarrow Q\}$$

I denne verden $\mathbf{T}$ kan det vises at $P \succ Q$, men ikke at $\neg Q \succ \neg P$.

3.4 Gärdenfors aksiomer

Vi vil nu se på, hvilke simple egenskaber $\succ$ har. Gärdenfors fremlagde i 1978 ni aksiomer, som han mente kontrafaktiske udsagn burde overholde. De otte af disse kan uden videre vises, i vores kalkyle:

(G1) $(P \succ Q) \wedge (P \succ R) \Longrightarrow (P \succ Q \wedge R)$.

(G2) $P \succ \top$, hvor $\top$ står for et tautologisk sandt udsagn, dvs. $\top$ er sandt i alle verdener.

(G3) Hvis $(Q \Longrightarrow R)$ er tautologisk sandt, vil det gælde at: $(P \succ Q) \Longrightarrow (P \succ R)$.

(G4) $P \succ P$.

(G5) Hvis $(P \Longleftrightarrow Q)$ er tautologisk sandt, vil det gælde at: $(P \succ R) \Longrightarrow (Q \succ R)$.

(G6) $(P \wedge Q) \Longrightarrow (P \succ Q)$.

(G7) $(P \succ Q) \Longrightarrow (P \Longrightarrow Q)$.

(G8) $(P \succ R) \wedge (Q \succ R) \Longrightarrow (P \vee Q) \succ r$.

Beviserne for disse sætninger er forholdsvis trivielle. Vi vil dog angive bevis for de sidste 4 af dem.

Bevis

(G5) Eftersom $(P \Longleftrightarrow Q)$ i alle verdener, vil $\mathbf{W}'(P, \mathbf{S}) = \mathbf{W}'(Q, \mathbf{S})$, hvilket betyder at $(P \succ R) \Longleftrightarrow (Q \succ R)$, som giver det ønskede. $\bullet \bigcirc \bullet$

(G6) Hvis $\mathbf{S} \models (P \wedge Q)$, fås $\mathbf{W}'(P, \mathbf{S}) = \{\mathbf{S}\}$, hvorfra Q naturligvis kan udledes. •○•

(G7) Det eneste der kan falsificere dette, er tilfældet hvor $(P \Longrightarrow Q)$ er falsk, og $(P \succ Q)$ er sand. Hvis $(P \Longrightarrow Q)$ skal være falsk, skal P være sand, og Q falsk. Men hvis P er sand, fås at $\mathbf{W}(P, \mathbf{S}) = \{\mathbf{S}\}$. I denne verden kan man ikke slutte Q, og det kontrafaktiske udsagn bliver derfor falsk. Heraf fås det ønskede. •○•

(G8) Lad $\mathbf{T} \in \mathbf{W}'(P \vee Q, \mathbf{S})$ og $\mathbf{T} \subset \mathbf{V} \subseteq \mathbf{S}$, hvis et sådant $\mathbf{V}$ eksisterer. Eftersom $\mathbf{T}$ er maksimal fås enten

a. $\mathbf{V}$ er "dårlig" ($B(\mathbf{V})$ er sand)
eller

b. $\mathbf{V} \models \neg P$ og $\mathbf{V} \models \neg Q$

Udfra definitionerne kan nu ses, at $\mathbf{T} \in \mathbf{W}'(P, \mathbf{S})$ og $\mathbf{T} \in \mathbf{W}'(Q, \mathbf{S})$. Ethvert element i $\mathbf{W}'(P \vee Q, \mathbf{S})$ er altså både i $\mathbf{W}'(P, \mathbf{S})$ og $\mathbf{W}'(Q, \mathbf{S})$. Heraf fås det ønskede. Hvis $\mathbf{V}$ ikke kan findes, har vi $\mathbf{S} \models p \vee Q$, dvs. $\mathbf{W}'(P \vee Q, \mathbf{S}) = \{\mathbf{S}\}$ og, enten $\mathbf{W}'(P, \mathbf{S}) = \{\mathbf{S}\}$ eller $\mathbf{W}'(Q, \mathbf{S}) = \{\mathbf{S}\}$, hvilket viser (G8). •○•

Gärdenfors' niende aksiom er et smertens barn.

(G9) $(P \succ Q) \wedge \neg(P \succ \neg R) \Longrightarrow (P \wedge R \succ Q)$.
Dette kan ikke umiddelbart vises, med vores nuværende teori. Beviset kræver en - ikke særlig intuitiv - partiel ordning af vores mulige verdener, som er noget stærkere end den almindelige mængderelation. Vi kan ikke retfærdiggøre en sådan ordning, men hvis vi kunne, ville beviset køre.

Bevis Det er tilstrækkeligt at vise, at

$$\mathbf{W}'(P \wedge R, \mathbf{S}) \subseteq \mathbf{W}'(P, \mathbf{S})$$

fordi, hvis det ikke er muligt at vise Q udfra $\mathbf{W}'(P \wedge R, \mathbf{S})$ er det heller ikke muligt at vise Q udfra $\mathbf{W}'(P, \mathbf{S})$ p.g.a. mængdeinklusionen, altså hvis succedenten i (G9) er falsk, så er antecedenten det også. Dette vil færdiggøre beviset.

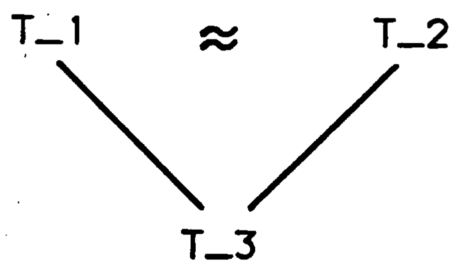
Vi vil nu indføre en partiel ordning, som skal beskrive en mulig verdens lighed med den oprindelige verden. Vi skriver $\mathbf{T}_1 \sqsubset \mathbf{T}_2$, hvis $\mathbf{T}_2$ ligner den oprindelige verden mere end $\mathbf{T}_1$ gør. Det er som sagt svært at sige hvordan en sådan ordning skal laves, udover at den skal respektere den almindelige mængderelation:

$$\mathbf{T}_1 \subset \mathbf{T}_2 \Longrightarrow \mathbf{T}_1 \sqsubset \mathbf{T}_2$$

Hvis $\mathbf{T}_1$ og $\mathbf{T}_2$ er usammenlignelige, dvs. hverken $\mathbf{T}_1 \sqsubset \mathbf{T}_2$ eller $\mathbf{T}_1 \sqsupset \mathbf{T}_2$ skrives at $\mathbf{T}_1 \approx \mathbf{T}_2$. Denne ordning har følgende egenskaber:

a. Den er transitiv.
b. Den er ikke symmetrisk. ($\mathbf{T}_1 \sqsubset \mathbf{T}_2$ og $\mathbf{T}_1 \sqsupset \mathbf{T}_2$ kan ikke forekomme samtidigt).
c. Den er ikke refleksiv. (Intet er mindre end sig selv).

Antag nu, at hvis $\mathbf{T}_1 \approx \mathbf{T}_2$ og $\mathbf{T}_3 \sqsubset \mathbf{T}_1$ så vil $\mathbf{T}_3 \sqsubset \mathbf{T}_2$. Herved opnås en modular ordning, $\sqsubset$.



Under denne antagelse, vil vi nu vise at

$$\mathbf{W}'(P \wedge R, \mathbf{S}) \subseteq \mathbf{W}'(P, \mathbf{S})$$

Lad

$\mathbf{T}$ være et givet medlem i $\mathbf{W}'(P \wedge R, \mathbf{S})$

R være et bestemt medlem i $\mathbf{W}'(P, \mathbf{S})$, således at $\mathbf{R} \not\models \neg R$. Et sådant **R** vil altid eksistere, hvis $(\neg(P \succ \neg R))$ er sandt. Hvis dette er falsk, er (G9) triviel. Eftersom $\mathbf{R} \not\models \neg R$ vil $\mathbf{R} \in \mathbf{W}'(P \wedge R, \mathbf{S})$.

Vi vil nu vise (kontradiktorisk), at **T** må være element i $\mathbf{W}'(P, \mathbf{S})$. Ellers kunne der findes et $\mathbf{V} \in \mathbf{W}'(P, \mathbf{S})$, således at $\mathbf{T} \sqsubset \mathbf{V}$ og $\mathbf{R} \approx \mathbf{V}$ (eftersom $\mathbf{R} \in \mathbf{W}'(P \wedge R, \mathbf{S})$). Men ud fra antagelsen har vi da $\mathbf{T} \sqsubset \mathbf{R}$, hvilket giver at $\mathbf{T} \in \mathbf{W}'(P \wedge R, \mathbf{S})$. Dette er en kontradiktion. •○•

3.5 Mulig udvidelse

Vi har i det foregående bygget vores kontrafaktiske kalkyle på udsagnslogikken. Vores kalkyle kan altså kun behandle udsagn og verdener beskrevet ved udsagnslogiske formler. Det ville give en væsentlig forøget udtrykskraft, hvis kalkylen var baseret på førsteordens prædikatlogik. Prædikater og kvantorer ville sætte os i stand til at opstille mere komplicerede scenarier. Der er dog også problemer forbundet ved at tage springet til førsteordens prædikatlogik.

1) Førsteordens prædikatlogik er ikke afgørlig. Det er ikke muligt at lave en kalkyle, der givet en førsteordens formel altid vil stoppe og fortælle om formlen er valid eller falsificerbar. Vores generering af mulige verdener indebærer en stor mængde beviser. Dette vil ikke være praktisk muligt i førsteordenslogikken.

2) Kontrafaktiske konditionaler er domæneafhængige. I førsteordenslogikken arbejder man med ét stort domæne, der ligger fast. Når man ønsker at formalisere kontrafaktiske konditionaler, skal man i mange tilfælde tage hensyn til mulige indskrænkninger eller udvidelser af domænet. Lad os endnu engang vende tilbage til eksemplet med Scandinavian Star. Vi betragter tre sætninger.

    a) Hvis skibets miserable tilstand var offentligt kendt før afgang, ville færre mennesker være omkommet.

b) Hvis skibet havde sejlet om dagen, ville færre mennesker være omkommet.

c) Hvis en passager undervejs telefonisk var blevet fortalt, hvilken dødssejler skibet var, ville færre mennesker være omkommet.

Den første sætning synes plausibel. Hvis det var kendt, at det var farlig at sejle med Scandinavian Star, ville færre mennesker formentlig have gjort det. Her er domænet, der interesserer os i princippet alle mennesker, der kunne finde på at sejle mellem Norge og Danmark. Den anden sætning synes også rimelig. Hvis passagererne ikke havde sovet, ville der ikke have været så mange omkomne. Her er domænet begrænset til personerne på skibet. I den tredje sætning sker der en sammenblanding af de to domæner, og det er den slags problemer man skal finde løsninger på, hvis man ændrer den underliggende logik. Når man danner mulige verdener vil nogle af udsagnenes domæner måske være forskellige.

3) Udvidelsen til førsteordens prædikatlogik ville være forbundet med store programmeringsmæssige problemer, som ikke har så meget med kontrafaktiske logik at gøre. Det mest interessante i forbindelse med projektet mener vi ikke er forskellen mellem udsagnslogik og førsteordens logik, men formaliseringen af kontrafaktiske konditionaler.

Kapitel 4

HOL-systemet

HOL (Higher Order Logic), er et system, udviklet til interaktiv bevisførelse i højere ordens logikken. Vi vil i dette afsnit give en introduktion til HOL systemet, i hvilket vi har implementeret en evaluator for kontrafaktiske konditionaler i udsagnslogikken.

4.1 En skitse af HOL's historie

HOL er bygget ovenpå programmeringssproget ML, som igen er bygget på sproget LCF. LCF (Logic for Computable Functions) blev oprindelig skrevet i Edinburgh i starten af 70'erne, af Gèrard Huet. LCF er en særlig stærk LISP, baseret på en typebundet lambdakalkyle. Det er specielt udviklet til, at arbejde med rekursive funktioner af stor kompleksitet. Et område hvor lambdakalkylen er velegnet. Huet's 'Edinburgh LCF' blev brugt i et fransk projekt ved navn 'Formel'. I starten af 80'erne blev LCF videreudviklet af Larry Paulson, og det er denne version - 'Cambridge LCF' - HOL er bygget på.

ML (Meta-Language) blev lavet som en overbygning af Edinburgh LCF. Det er idag er selvstændigt programmeringssprog, som er meget brugt. Det adskiller sig særligt fra LCF ved det behagelige type-check, som er en stor fordel for et funktionsprogrammeringssprog.

HOL blev udviklet af Mike Gordon, i starten af 80'erne, i Cambridge. Det primære formål med HOL var at verificere computer hardware. Den HOL version vi har arbejdet med, er 88 udgaven (HOL88).

4.2 Grund-ide i HOL88

I HOL bygger man **teorier**. En teori består af aksiomer, definitioner og teoremer. Grundtanken er, at man indtaster sine aksiomer, definitioner og typedefinitioner, og med disse begynder at bevise relevante teoremer. Beviste teoremer kan gemmes, og indgå i teorien - ligesom aksiomerne, og bruges i senere beviser. På denne måde opstår teorien. HOL's styrke er selve bevisprocessen. HOL er udviklet til at lette denne proces, samt sørge for, at beviset går rigtigt for sig. Når man har bevist en sætning i HOL, kan man være sikker på (mener Gordon), at beviset er rigtigt (inden for teorien). Dette sikrer naturligvis ikke, at teorien er konsistent aksiomatisk, men det skulle give en vis tryghed at vide, at man ikke kan lave fejl i beviser.

4.3 Mål og midler

Hvis man ønsker at vise en sætning i HOL, giver man den status som et **mål** (goal). Et mål bearbejdes med **taktikker**, svarende til 'backward proofing', som beskrevet i tidligere kapitel. En taktik omformulerer et mål til et, eller flere **delmål** (subgoals). Disse delmål kan igen behandles med taktikker, og så fremdeles. Herved fås er bevistræ, som HOL automatisk tager sig af. Til dette bruges den såkaldte goalstack, som holder rede på, hvilket delmål man er igang med at vise, samt hvilke delmål man mangler at vise. HOL accepterer et bevis, dvs. en sammensætning af taktikker, hvis samtlige grene i bevistræet er bevist (eller er aksiomer). Når et mål er bevist i HOL, kan det gemmes som teorem, og bruges senere.

Et HOL bevis er altså et 'backward proof', hvor man ved brug af taktikker opstiller et bevistræ, repræsenteret ved goalstack'en. Der er desuden mulighed for at benytte 'forward proof' i HOL, men HOL lægger op til, at 'backward proof' benyttes.

4.4 Taktikker og bevisfunktioner

En taktik er en funktion af typen:

Taktik: Mål → (Liste af delmål, Bevisfunktion)

Hvis man anvender en taktik på et givet mål, vil man som resultat få et par af de (eller det) delmål det oprindelige mål kan reduceres til, og en funktion. Denne funktion kaldes for **bevisfunktionen** eller (i HOL notation) 'justification-function'. Den har typen:

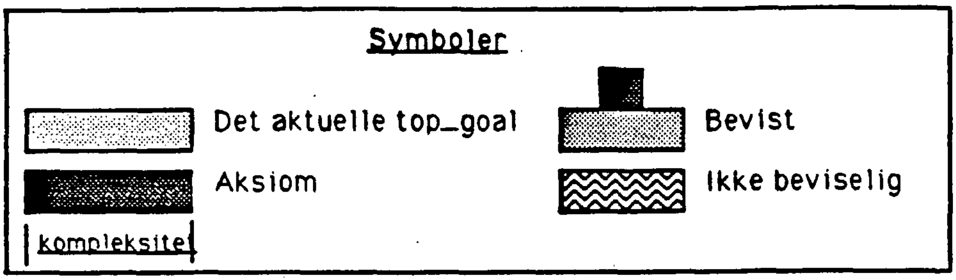
Bevisfunktion: Liste af teoremer → Oprindeligt mål som teorem

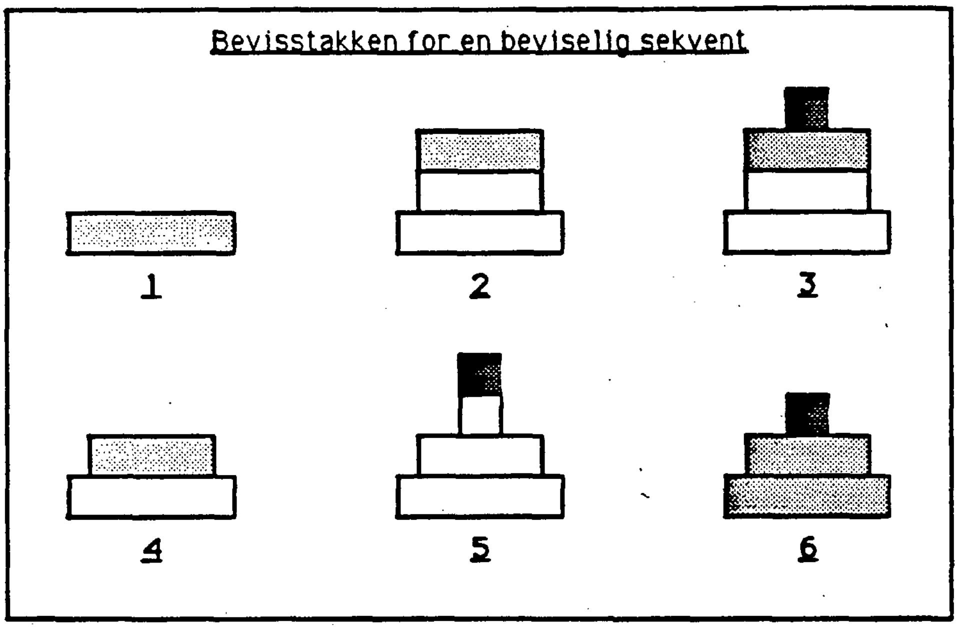
Bevisfunktionen kan siges at være "den inverse" af taktikken, i den forstand at hvis taktikken reducerer et mål til en liste af delmål, så skal bevisfunktionen gøre det modsatte. Lad os antage, at hele listen af delmål faktisk er blevet bevist. Så vil de alle have fået status som teoremer (dette ordner HOL automatisk). Hvis bevisfunktionen nu, udfra denne liste at teoremer (beviste delmål), ved logisk lovlige operationer kan opnå det oprindelige mål, regnes dette for bevist, og det får status som teorem.

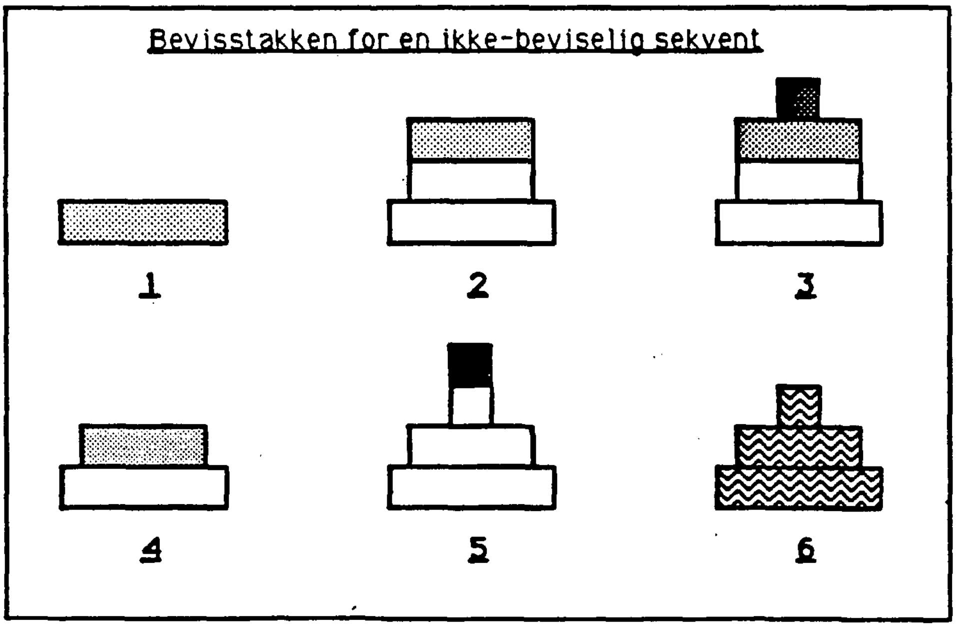
Hvis man i HOL ønsker at bevise en sætning, benytter man normalt 'backward proof'. Taktikkerne bruges til at dele et mål op i mindre delmål, og bevisfunktionen skal huske hvordan man kommer tilbage til det oprindelige mål. Man kan derfor sige, at bevisfunktionen er et 'forward proof' for sætningen. Følgende rekursive algoritme bruges som bevismetode i HOL. Vi ønsker at vise en sætning A.

```
0. Hvis A er et aksiom
1. så er det bevist
      og A får A status som teorem og returneres.
2. ellers
      hvis man ikke kan finde en brugbar taktik
        så kan A ikke vises i HOL
        ellers
          - brug en taktik på A
          - og få en liste af delmål (eks. B1 og B2)
          - og en bevisfunktion.
3.        - Hvis B1 kan vises
              så få teoremet B1 (TB1)
              ellers så stop.
4.        - Hvis B2 kan vises
              så få teoremet B2 (TB2)
              ellers så stop.
5.        - Brug bevisfunktionen på TB1 og TB2.
            Hvis resultatet af dette svarer til sætningen A,

              så er A bevist,
                 og teoremet A returneres.
```


Symboler
Det aktuelle top_goal
Bevist
Aksiom
Ikke beviselig
kompleksitet


Bevisstakken for en beviselig sekvent
1
2
3
4
5
6


Bevisstakken for en ikke-beviselig sekvent
1
2
3
4
5
6

Dette er en 'dybde først' algoritme. Som det ses, kan algoritmen ovenfor automatiseres fuldstændigt hvis man er sikker på, at alle de brugte taktikker **formindsker** kompleksiteten, samtidigt med at de **bevarer** informationen. Gentzen taktikkerne gør dette. Desværre er det slet ikke tilfældet for alle HOL-taktikker, hvilket måske også ville et være urimeligt krav. HOL er trods alt udviklet med henblik på højere ordens logikken, hvor fuldstændig automatisk bevisførelse ikke er muligt. Derfor er et HOLbevis normalt en **interaktiv** proces. Programmet holder check på, hvor man er i bevistræet, hvilke delmål som er vist, og hvilke man mangler at vise. Desuden husker HOL det formelle bevis (bevisfunktionen), for det man har lavet. Det interaktive består i, at man skal fortælle HOL, hvilke taktikker den skal bruge hvor. Desværre er der et mindre hav af forskellige taktikker, og man kommer nemt til, at bruge en masse tid på, at finde passende taktikker til forskellige situationer.

4.5 Hvad HOL bliver brugt til

HOL er ment som et værktøj, der skal lette udviklingen af nye teorier og tjekke disse teoriers gyldighed. Type-definitioner, tegndefinitioner og nye aksiomer kan laves med forholdsvis simple kommandoer. HOLs styrke ligger i at den indbyggede logik er konsistent og sund.

Der lægges ingen begrænsning på, hvor mange og hvilke aksiomer, der kan defineres. Det er derfor ikke svært at lave inkonsistente systemer i HOL. Eftersom man kan lave HOL logikken inkonsistent, ved blot at tilføje eet enkelt aksiom, er det en farlig leg. Man kan ikke garantere et bevis' gyldighed, hvis man har udvidet HOL's logik med ekstra aksiomer. HOL-forfatterne opfordrer derfor folk til kun at lave **konservative** udvidelser af HOL-logikken, dvs. ingen nye aksiomer, men gerne f.eks. nye typer og definitioner. Hvis man alene arbejder med konservative udvidelser af HOL logikken skulle det ikke være muligt at udlede falske teoremer. Konservative udvidelser regnes derfor som god HOLetik.

Alt i alt giver det at arbejde i HOL nogle fordele:

1. Hvis man kun laver konservative udvidelser af HOL-logikken, får man foræret en vis blåstempling af sin teori. HOL beviser har så småt fået status som værende 'gyldige', hvis man overholder HOL-etikken. Man ser derfor, at HOL beviser bliver brugt som argument for teoriers gyldighed ("Bevist i HOL", i en klar blå farve).
2. HOL overtager noget at det trivielle arbejde ved bevisførelsen for nye teoremer. Dels fører HOL beviserne i gennem i detaljer (alle underliggende teoremer bliver afprøvet hver gang) og dels er det i den interaktive afvikling af et bevis muligt at anvende egne tidligere udviklede teoremer.
3. At implementere en teori giver tit et større kendskab til teorien. Hvis man f.eks. har lavet nogle usmarte definitioner, vil man måske indse det, hvis man prøver at bruge dem i HOL. Her kan HOL fungere som forsøgskanin, som tester anvendeligheden af forskellige definitioner i teorien.

4.6 Kritik af HOL

Ved den første kontakt med HOL-systemet blev vores umiddelbare forventninger skuffet. Vi forventede at et program på en datamaskine, tilvejebringer en eller anden form for automatisering af arbejdsprocessen. Datamaskinen burde gøre arbejdet lettere end det ville være at udføre i hånden.

En del processer er naturligvis automatiseret i HOL-systemet. Således får man f.eks. oplyst om en given taktik er ubrugelig eller om den kan gennemføres. Hvis det sidste er tilfældet bliver bevistræet automatisk opdateret og den nye tilstand i bevisforløbet udskrevet.

Men hvilken taktik, der i en given sammenhæng skal anvendes får man ingen hjælp til. Kan man selv analysere sig frem til hvilken slutningsregel, der i en given situation vil være den rigtige at anvende, så er man selvfølgelig nået et stykke af vejen.

Men det møjsommelige arbejde med at finde ud af om nu også HOL-systemet rummer en taktik, der kan anvendes til netop dette formål tager længere tid end det vil gøre at udføre hele arbejdet i hånden. Dels er navngivningen af taktikkerne kryptisk og mange taktikker (som vi egentligt synes burde være der) findes ikke. Således f.eks. ikke de GENTZEN taktikker, som vi havde brug for.

Man får ingen hjælp til at tilegne sig et overblik over de tilgængelige taktikker. De til systemet medfølgende manualer er af en gennemført dårlig kvalitet for den, der er nybegynder med HOL-systemet. Måske er de gode nok til den, der i forvejen er en erfaren bruger[1].

4.7 Hvad vi har brugt HOL til

I dette projekt har vi brugt HOL lidt utraditionelt. Hvad vi primært havde brug for, var en **bevismaskine** som automatisk kunne generere beviser i udsagnslogikken. Denne opgave er HOL ikke bygget til, og vi har derfor været nød til at bygge det lidt om. For det første er Gentzen-kalkylen (som vi gerne ville arbejde med) ikke indbygget i standard HOL. For det andet er HOL beviser normalt ikke automatiske, men interaktive. Hvad vi havde brug for, var altså noget som HOL normalt ikke bliver brugt til. Vi skulle derfor tune systemet en hel del for at bruge det. Men selv om HOL ikke umiddelbart er bygget til vores formål, var der flere aspekter i det, vi kunne udnytte.

1. HOL er som sagt lavet i ML, som er et ganske fortrinligt sprog. Eftersom dette sprog fortrinsvis er lavet til at arbejde med logiske problemstillinger, har det - til vores formål - mange fordele frem for mange andre programmeringssprog.
2. Den struktur HOL benytter til beviser (bevistræet), har vi med fordel kunnet benytte. Vi har ganske vist været nød til, selv at programmere de nødvendige taktikker, men det at kunne bruge den i HOL indbyggede bevisstruktur, har sparet os for meget programmerings arbejde.
3. De type-definitioner som indgår i standard HOL (mål, teorem m.m.) kunne vi også drage fordel af.

[1] Det skal retfærdigvis nævnes, at de manualerne, vi har været præsenteret for, er 1. udgaver. Det ser ud som om, der er et arbejde i gang for at forbedre dem.

Lad det dog være sagt, at arbejdet med HOL ikke har været problemfrit. Ingen i gruppen kendte særligt til HOL (eller ML) ved projektets start. Vi har derfor brugt en del tid på, at sætte os ind i hvordan HOL virker. I særdeleshed bevistræet og taktikkerne. Endelig har vi haft problemer med bevisfunktionen. Vi ville gerne have, at vores udvidelse af HOL var konservativ, men har desværre haft nogle tekniske problemer med dette, navnlig på grund af bevisfunktionen. Vores udvidelse er en konservativ udvidelse af logikken. Men for at implementere udvidelsen, har vi måttet tage os visse friheder i forhold til 'HOL-etikken'.

Kapitel 5

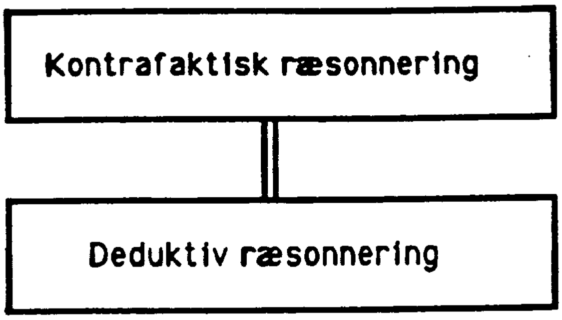
Modellen

Den model, som vi har fremstillet, har til formål at gøre det muligt at vurdere kontrafaktiske konditionalers gyldighed. Givet et kontrafaktisk konditionale: $P \succ Q$, består opgaven i, at generere nogle mulige, tænkte verdener. De mulige verdener skal opfylde en række krav:

- det i den 'virkelige' verden falske udsagn skal være sandt,
- de mulige verdener skal være **konsistente**
- og de skal være **maksimalt similære** med den faktiske verden.

I disse mulige verdener vurderes gyldigheden af udsagnet ved at undersøge om succedenten af udsagnet: Q, kan bevises. Man kan sige at $P \succ Q$ i den faktiske verden vurderes ved at vurdere $P \Longrightarrow Q$ i alle de mulige verdener. Beviset gennemføres ved hjælp af det underliggende bevissystem, som i vores model er Gentzen-kalkylen. Man kan således også udtrykke problemstillingen ved at sige, at den kontrafaktiske ræsonnering udføres ved at omforme problemet til et problem for den underliggende deduktive ræsonnering.

Det er frugtbart for udviklingen af et program på en datamaskine at formalisere denne overordnede struktur som det er skildret i følgende figur:



I den øverste kasse indgår de dele, der specifikt er knyttet til den kontrafaktiske ræsonnering: Vurdering af gyldighed, vurdering af maximal similaritet, generering af mulige verdener. Mens den nederste kasse rummer den underliggende logik, der dels sætter grænserne for hvilke kontrafaktiske udsagn, systemet kan kapere og som beviser de mulige verdeners konsistens. I vores model rummer denne kasse udsagnslogikken med Gentzen kalkylen som bevissystem.

Delingen tydeliggør, at det i princippet er muligt at udskifte f.eks. udsagnslogikken med 1. ordens logik uden at ændre den øvrige model.

5.1 Modellen i HOL

Det har været et konkret incitament for dette projekt at undersøge hvilke muligheder HOL giver for at implementere logiske ræsonnementer på en datamaskine. Som tidligere beskrevet har dette ikke været helt problemfrit og vi har fundet det nødvendigt at basere os primært på det underliggende programsprog, ML, i stedet for at anvende indbyggede HOL-taktikker. Vi har skrevet vores egne taktikker og anvender kun enkelte prædefinerede HOL-taktikker.

Det har været en væsentlig hjælp for udviklingen af programmet at vi har kunnet anvende de i HOL indbyggede faciliteter til at styre bevistræet. (HOLs subgoal package). For at konstruere programmet er det derfor alene nødvendigt at programmere taktikker og funktioner, der er specifikke for vores model.

Før vi går over til beskrivelsen af vores konkrete model gives her en redegørelse for tilknytningen til HOLs bevistræ og for den anvendte notation i den videre beskrivelse.

5.1.1 Tilknytningen til HOL

Tilknytningen til HOLs bevistræstruktur sker via anvendelsen af den indbyggede funktion: expand. Et funktionssprog som ML gør det muligt at skrive meget kompakt kode. F.eks. er den faktiske ML-kode for funktionen expand følgende:

```
let expand = expandf o VALID;;

hvor

let expandf tac =
      change_state(abs_goals
           (push_fsubgoals (rep_goals goals) tac));;
```

og hvor VALID, som navnet antyder undersøger om den givne taktik kan anvendes. o angiver funktions sammensætning.

For at gøre denne tekst mere læsevenlig har vi valgt at beskrive vores model i en pseudokode svarende til nedenstående eksempel. I pseudokode kan expand beskrives således:

```
expand(Taktik)
   a. tag den øverste Sekvent i BevisTræ
   b. anvend Taktik på denne Sekvent
   b1.  hvis det går godt
          så undersøg om BevisTræ er udtømt
          ellers udskriv en fejlmeddelelse
```

Det program, vi har udviklet vil blive beskrevet i en pseudokode svarende til dette eksempel. Kildeteksten, som den er skrevet i ML, er tilføjet i bilag. Kildeteksten er den egentlige detaljerede dokumentation for modellens implementering, men det er tilstræbt at pseudokoden nøjagtigt gengiver alle væsentlige algoritmiske aspekter af programmet.

Vi indfører endvidere følgende notation. Alle betydende variable vil blive skrevet med Stort begyndelses bogstav. Funktioner, vi selv har defineret skrives med STORE bogstaver. Indbyggede ML funktioner, som det skønnes nødvendigt at nævne vil blive understreget. Argumenter til funktioner vil blive (indrammet i paranteser). Den øvrige tekst i pseudokoden tjener til at tydeliggøre, hvad der foregår.

Fremstillingen af modellen vil iøvrigt foregå på så kaldt top_down maner, hvilket indebærer at underliggende funktioner vil blive anvendt i algoritmefremstillingen, førend de er beskrevet. Dette kan måske virke generende på nogen, men vi skønner, at det vil formidle et bedre overblik til læseren.

5.2 Kontrafaktiske funktioner

Vi har implementeret følgende funktioner, som varetager den kontrafaktiske ræsonnering.

- KFK : (Verden, Sekvent) $\longmapsto$ bool
- MULIGE_VERDENER : (Verden, Term) $\longmapsto$ [liste af verdener]
- KONSISTENT : Verden $\longmapsto$ bool

MULIGE_VERDENER funktionen er implementeret således at kun de mulige verdener, der ikke er restriktioner af andre mulige verdener bliver genereret. I en videre udvikling af modellen ville det være relevant at reducere antallet af mulige verdener med andre kriterier. Det ville i så fald være oplagt at tilføje en funktion, der givet en liste af mulige verdener, returnerer en ny (formodentlig reduceret) liste. Hvis denne funktion skal indføres vil dens type f.eks. kunne defineres således:

- MAX_SIMILÆR : [liste af verdener] $\longmapsto$ [liste af verdener]

5.2.1 KFK

KFK er en funktion, der vurderer gyldigheden af et kontrafaktisk konditionale (Sekvent) i en 'faktisk' verden (Verden) og som returnerer true eller false. Overordnet består den af 3 step, nemlig

1. dan de mulige verdener
2. vurdering af gyldigheden i disse verdener
3. returnering af resultatet.

En nærmere uddybning giver følgende fremstilling af funktionen:

```
KFK(Verden,Sekvent)
  1. dan de mulige verdener
    a. VerdensListe = MULIGE_VERDENER(Verden,
                               antecedenten af Sekvent)
  2. undersøg om udsagnet er sandt i alle disse verdener.
    a. Resultat := true
    b. for alle verdener, V' i VerdensListe
    b1. tilføj antecendenten af Sekvent til V'
    b2. dan sekventen S' := V' --> succedenten af
                                   Sekvent
    b3. Sandt := BEVIS(S')
    b4. hvis Sandt = false så Resultat := false
  3. returner Resultat
```

5.2.2 MULIGE_VERDENER

MULIGE_VERDENER er en funktion, der genererer en liste af mulige verdener givet en verden (Verden) og et kontrafaktisk udsagn (Term). For en given verden med n udsagn gives der 2^n forskellige delmængder. Det algoritmiske problem består i at begrænse undersøgelsen af disse delmængder, således at,

1. alle konsistente delmængder findes,
2. der ikke bliver undersøgt nogen delmængde, der er en restriktion af en i forvejen undersøgt konsistent delmængde,

3. ingen delmængde undersøges mere end een gang.

Den nuværende algoritme er desværre ikke helt tilfredsstillende i så henseende, idet krav nummer 2. ikke helt opfyldes. Problemet forårsages af at vi anvender en dybde-først algoritme. En bredde først algoritme ville forhindre dette unødvendige arbejde, men ville også så vidt vi kan se, gøre programmeringen mere besværlig, idet vi så ikke kunne udnytte HOLs bevistræ, som vi gør i den nuværende rekursive algoritme.

For at bøde på denne mangel i algoritmen, har vi indført en funktion, der rydder op i listen af verdener efter, den er genereret. Vi har opbygget funktionen MULIGE_VERDENER med en indlejret rekursiv funktion MV. MULIGE_VERDENER ser således ud:

```
MULIGE_VERDENER(Verden, Term)
  a. VerdensListe := MV(Verden, 0)
  b. VerdensListe := RYD_OP(VerdensListe)
  c. returner VerdensListe
```

MV

Funktionen MV er rekursiv. Den tager en Verden og et tal (Nummer) og returnerer en VerdensListe. Nummer anvendes som en pegepind for at holde styr på hvilket udsagn i Verden, vi aktuelt er nået til. Overordnet kan MV beskrives ved følgende step:

```
1. hvis Verden er tom
     så stop
2.   ellers hvis Verden er konsistent
3.           så tilføj Verden til VerdensListe
4.           ellers tilføj de konsistente delmængder
                af Verden til VerdensListe.
```

En mere præcis beskrivelse af funktionen gives ved:

```
MV(Verden, Nummer)
  a. P := Nummer
  b. hvis Verden er tom
  b1.  så stop
       ellers
  b2.   hvis KONSISTENT(Verden tilføjet Term)
  b2a.   så tilføj Verden til VerdensListe
         ellers
  b2b.     sålænge P < antallet af elementer i Verden
  b2b1       lad P := P + 1
  b2b2       lad V' := Verden fratrukket element nr P
  b2b3       tilføj MV(V',P-1) til VerdensListe
```

5.2.3 KONSISTENT

KONSISTENT er en funktion, der givet en Verden undersøger om denne verden rummer kontradiktioner. Hvis Verden er konsistent returneres <u>true</u> ellers returneres <u>false</u>. Undersøgelsen foretages ved at forsøge at bevise det falske ud fra Verden. Hvis det falske kan bevises, så er Verden inkonsistent. Derfor returneres negationen af resultatet af undersøgelsen. Givet en bevismaskine bliver koden meget simpel, idet vi blot skal danne Sekventen og derefter sætte bevismaskinen i gang.

```
KONSISTENT(Verden)
  Sekvent := Verden --> false
  not BEVIS(Sekvent)
```

5.3 Deduktive funktioner

Ønsker man at bevise en sekvent i Gentzen-kalkylen med et backward-proof sker det via den succesive anvendelse af de forskellige Gentzen-regler (i det følgende kaldet taktikker) på sekventen og de afledede delsekventer. I HOL terminologi tales om goal og subgoals. Taktikkerne anvendes enten på antecedenten (venstresiden af sekventpilen) eller på succedenten (højresiden af sekventpilen). Igennem denne proces reduceres kompleksiteten af sekventerne fra skridt til skridt. Processen stopper enten

1. når det samme udsagn optræder på både højre og venstresiden af sekventpilen
2. eller når både højre og venstresiden er reduceret til atomiske udsagn og der derfor ikke kan anvendes flere taktikker på dem.

Den enkelte delsekvent er bevist, når situation 1. indtræffer, mens sekventen ikke kan bevises, hvis situation 2. indtræffer.

En given sekvent er bevist når alle de afledede delsekventer er bevist, mens den ikke kan bevises, hvis blot een af delsekventerne ender i situation 2. Systemet af sekvent og afledede delsekventer danner tilsammen bevistræet og det er dette bevistræ, som HOL systemet holder rede på i vores program.

Opbygningen af en bevismaskine for udsagnslogikken med Gentzen-kalkylen som bevissystem består herefter blot i at

a. programmere de enkelte Gentzen taktikker for højre og venstresiden af sekventpile

b. programmere en funktion, der vælger hvilken taktik, der skal anvendes over for en given sekvent.

c. programmere en funktion, der undersøger om det samme udsagn optræder på både højre- og venstresiden

d. og endelig at programmere en styre funktion, der holder maskinen kørende indtil der ikke kan udføres flere taktikker. Det vil sige til top sekventen (i HOL terminologi: top_goal) er bevist eller vist at den ikke kan bevises.

Konkret består vores bevismaskine af styrefunktionen BEVIS() og af ėn række funktioner, der opfylder HOLs krav til en taktik. Samlet er funktionernes typer defineret ved:

- BEVIS : Sekvent $\longmapsto$ ([liste af sekventer], funktion)[1]
- BEVIS_TAC : Sekvent $\longmapsto$ ([liste af sekventer], funktion)
- CHECK_TAC : Sekvent $\longmapsto$ ([liste af sekventer], funktion)
- V_IMP_TAC : Sekvent $\longmapsto$ ([liste af sekventer], funktion)
- V_KONJ_TAC : Sekvent $\longmapsto$ ([liste af sekventer], funktion)
- V_NEG_TAC : Sekvent $\longmapsto$ ([liste af sekventer], funktion)
- V_DISJ_TAC : Sekvent $\longmapsto$ ([liste af sekventer], funktion)
- H_IMP_TAC : Sekvent $\longmapsto$ ([liste af sekventer], funktion)
- H_KONJ_TAC : Sekvent $\longmapsto$ ([liste af sekventer], funktion)
- H_NEG_TAC : Sekvent $\longmapsto$ ([liste af sekventer], funktion)

5.3.1 BEVIS()

BEVIS er en funktion, der modtager en Sekvent og returnerer true, hvis sekventen kan bevises ellers returneres false. Da vi

[1]Funktionen BEVIS's type er i det faktiske program lidt anderledes, nemlig BEVIS : void $\longmapsto$ bool. Man skal altså ikke give BEVIS en Sekvent, som parameter. BEVIS henter selv Sekvent fra toppen af bevistræet. Dette gøres ved hjælp af funktionen expand, der altid arbejder på det øverste mål i bevistræet. Vi har valgt at se bort fra denne tekniske detalje i redegørelsen for algoritmerne

anvender HOLs bevistræ er konsekvenserne af BEVIS større end det her er beskrevet.

Hvis Sekvent kan bevises, så bliver beviset faktisk gennemført og det er muligt at få det udskrevet. Vi har indført et flag (af typen bool), der hedder Udskriv_Bevis. Hvis Udskriv_Bevis sættes til true, vil beviset blive udskrevet, mens det genereres.

Hvis Sekvent ikke kan bevises udskrives de delsekventer, hvor bevismaskinen er gået i stå . Ud fra disse delsekventer er det umiddelbart let at se, hvilken variabeltilskrivning, der falsificerer Sekvent. Pseudokoden for BEVIS ser således ud:

```
BEVIS(Sekvent)
   gentag BEVIS_TAC(Sekvent) til den fejler
   hvis Sekvent er bevist
     så returner true
     ellers returner false
```

5.3.2 Taktikker generelt

Som det tidligere er beskrevet er en taktik en funktion, der modtager et goal og returnerer en **liste af subgoals** og en **bevisfunktion**, der ud fra listen af subgoals kan genskabe det oprindelige goal. I pseudokode kan en taktiks opbygning beskrives på følgende måde:

```
Taktiknavn Sekvent
  dekomponer Sekvent
  sammensæt et eller flere delmål
  returner (liste af delmål, bevisfunktionen)
```

Taktikken fejler, hvis de manipulationer på Sekvent, der er fastlagt i taktikken ikke kan udføres.

5.3.3 BEVIS_TAC

BEVIS_TAC er en funktion, der opfylder de formelle krav, HOL stiller til en taktik. Den modtager en Sekvent og returnerer en liste af sekventer og en bevisfunktion. Men den gør det indirekte via en af de underliggende taktikker. HOL systemet rummer en funktion FIRST, der givet en liste af taktikker udfører den første, der ikke fejler. BEVIS_TAC kan derfor konstrueres ved følgende pseudokode:

```
BEVIS_TAC(Sekvent)
  udfør den første taktik, der dur fra listen
    [CHECK_TAC; V_KONJ_TAC; V_NEG_TAC; H_NEG_TAC;
     H_IMP_TAC; V_IMP_TAC; V_DISJ_TAC; H_KONJ_TAC]
  på Sekvent
```

Den her valgte metode er, som det fremgår, begrebsligt meget enkel, men den er også tidskrævende. (Det er en såkaldt brute force metode). Hvis Sekventen f.eks. er af typen $A, B \rightarrow A \wedge B$, hvor den rigtige taktik at anvende er H_KONJ_TAC, så bliver først de 7 andre taktikker undersøgt, og derefter findes den rigtige.

En optimering af koden kan foretages ved først at analysere Sekventen og derefter foretage et kvalificeret valg af den rigtige taktik. Da vores program er en prototype, har vi ikke satset på at optimere denne algoritme. Vi har alene ordnet listen af taktikker således at de taktikker, der får bevistræet til at vokse mindst muligt vælges først.

5.3.4 CHECK_TAC

CHECK_TAC er en funktion, der undersøger om den aktuelle sekvent er et aksiom og det derfor ikke er nødvendigt at dekomponere den yderligere. Denne situation fremkommer, hvis den samme term findes på både højre og venstreside af sekventpilen. I HOL opbevares antecedenten som en liste af termer, mens HOL kun tillader, at der optræder een term i succedenten. Denne højre side af sekventpilen har derfor form som en disjunktion af termer, der først må laves om til en liste af termer.

Algoritmen i CHECK_TAC kan beskrives som

```
CHECK_TAC(Sekvent)
  Lav succedenten af Sekvent om
          til en liste af termer.
  Find fællesmængden af antecedentlisten
                    og succedentlisten.
  Hvis den ikke er tom,
    så er Sekvent et aksiom
    ellers fejl
```

Algoritmen for CHECK_TAC har den fordel, at det ikke er nødvendigt at dekomponere sekventen ned til atomiske formler. Hvis den samme term findes på højre og venstresiden vil sekventen blive karakteriseret som et aksiom, uanset hvor kompliceret den givne term måtte være.

5.3.5 En gentzen taktik

Vi har programmeret 7 forskellige Gentzen taktikker. De er alle fremstille over det samme hovedskema som gengivet i afsnittet 'Taktikker generelt'. Vi vil derfor her alene se nærmere på een af taktikkerne, nemlig V_KONJ_TAC.

V_KONJ_TAC

Gentzen reglen for denne taktik ser således ud:

$$\frac{\mathbf{S1}, A, B \rightarrow \mathbf{S2}}{\mathbf{S1}, A \wedge B \rightarrow \mathbf{S2}}$$

Taktikken må først undersøge om den aktuelle sekvent overhovedet rummer en konjunktion på venstresiden. Denne undersøgelse er blind i den forstand, at der godt kan være flere konjunktioner, men der udføres ingen vurdering mellem dem. Det er den første den bedste, der vælges.

Hvis der findes en konjunktion, så splittes denne op i sin højre og venstreside og disse 2 termer sættes nu i en liste sammen med den resterende del af antecendenten. I vores algoritme har det ingen betydning hvor i listen den oprindelige konjunktions elementer placeres. Denne liste udgør antecedenten til det nye delmål, som har den samme succedent, som den aktuelle sekvent. Endelig returneres det nye delmål og en bevisfunktion. Problemerne heri gennemgå i det følgende afsnit.

```
V_KONJ_TAC(Sekvent)
  hvis der er en konjunktion i antecedenten
    så Konj := denne konjunktion
    ellers fejl
  lad A := venstresiden af Konj
  lad B := højresiden af Konj
  lad NyListe := listen af A, B
            og (antecedenten af Sekvent - Konj)
  lad Delmål := (Nyliste, succedenten af Sekvent)
  returner (Delmål, Bevisfunktion)
```

5.3.6 Problemet med bevisfunktionen

En taktik tager som sagt et mål som argument, og giver en liste af delmål samt bevisfunktionen for taktikken. Bevisfunktionen skal kunne tage en liste af teoremer som delmål, og skal ved logiske operationen kunne opnå det oprindelige mål. Vores problem med bevisfunktionen er, at vi ikke alene ud fra denne liste at teoremer og HOL slutningsreglerne, kan opnå et teorem som ligner det oprindelige mål tilstrækkeligt. Det kan sagtens være logisk ækvivalent med det oprindelige mål, men det er ikke sikkert, at HOL kan genkende det. Der opstår syntaktiske problemer.

Vi ser på eksemplet H_NEG_TAC, en taktik som skal udføre Gentzen reglen:

$$\frac{\Gamma, P_2 \vdash P_1 \vee P_3 \ (1)}{\Gamma \vdash P_1 \vee \neg P_2 \vee P_3 \ (2)}$$

H_NEG_TAC skal tage et mål som (2), og give et delmål som (1) samt en bevisfunktion. Denne skal som argument tage et teorem som (1), og give et teorem som (2). Hvis vi undersøger et mål som

$$A, B \vdash C \vee \neg D \quad (3)$$

skal taktikken omskrive det til delmålet:

$$D, A, B \vdash C \quad (4)$$

Og desuden angive et bevis for, at man udfra teoremet (4) og gyldige slutningsregler i HOL kan opnå (3). Det første er simpelt nok. Man lader computeren lede højresiden igennem indtil den finder et negeret udtryk. Den tager udtrykket ud af højresiden, sletter negationen, og sætter det ind på venstresiden. Ved denne procedure vil den udfra (3) opnå (4). Men hvorledes laves et generelt bevis som går den anden vej? Man skal alene udfra et teorem (4) samt indbyggede logiske regler i HOL opnå et teorem som (3). Men desværre skal dette være formuleret præcis som (3). Der er flere problemer forbundet med dette.

Lad os sige, at vi udfra teoremet (4) skal finde ud af, hvad det oprindelige mål har været. Vi ved, at vi skal tage et udtryk fra venstresiden, negere det, og flytte det over på højresiden et sted. Alt dette kan gøres ved logisk lovlige operationer (se appendix B). Men præcis hvilket udsagn, A, B eller D er det rigtige at tage ? Det er vigtigt at vælge det rigtige, ellers kan man jo ikke komme tilbage til det oprindelige mål. Dette problem kan dog løses ad teknisk vej. Man skal bare sørge for, at taktikken placerer det aktuelle udsagn først på venstresiden i delmålet. Man kan derefter konstruere bevisfunktionen således, at den arbejder på det første udsagn på venstresiden, hvilket vil være det rigtige. Men selv om man har løst dette første problem, støder man på et andet, mere alvorligt problem. Man ved hvilket udsagn man vil arbejde på (i eksemplet er det D'et), og man kan med lovlige operationer nemt negere det, fjerne det fra venstresiden, og komme det ind et sted på højresiden. Men hvor skal man komme det ind på højre siden? I eksemplet kan vi enten lave (3) eller

$$A, B \vdash \neg D \vee C \quad (5)$$

og vi ved ikke hvilket der er det oprindelige. Umiddelbart vil man som menneske mene, at (5) er det samme som (3). Men for HOL er der desværre forskel. Hvis man som oprindeligt mål har haft (3), men bevisfunktionen finder frem til et teorem som (5), vil HOL forkaste tanken om at bruge den taktik, fordi HOL ikke kan genkende (3) i (5).

Problemet i en nøddeskal er, at vi ikke udfra et teorem (1) med gyldige slutningsregler kan finde et unikt oprindeligt mål som (2), **med mindre vi kender det på forhånd.** Men eftersom man <u>skal</u> angive en bevisfunktion for enhver taktik, og vi <u>skal</u> bruge en taktik som H_NEG_TAC, har vi været nød til at bryde med enhver form for HOL-etik, og snyde.

5.3.7 Hvordan man omgår HOLs bevisfunktion

I HOL er der en direkte snydefunktion. HOL kender forskel på beviste og ubeviste påstande ved hjælp af typer. Teoremerne (type : thm) er de beviste sætninger. Målene er dem, man ønsker at vise (type : goal). Når man har bevist en sætning, har man lavet et mål om til et teorem.

Der skal faktisk ikke mere til. HOL's snydefunktion (mk_thm) kan lave ethvert mål om til det tilsvarende teorem, ved blot at ændre typen på målet. Med denne funktion bliver bevisfunktionen en smal sag.

- Bevisfunktion = λ (liste af teoremer). mk_thm (oprindeligt mål)

Man skal udfra en liste af teoremer (i eksemplet en liste bestående af et element, nemlig (4)), bevise det oprindelige mål (i eksemplet (3)). Dette gøres ved at tage listen af teoremer, glemme alt om dem, for til slut at fortælle, at vi har bevist det vi skulle.

Fordele og ulemper.

Bevisfunktionen som vi har lavet den, er temmeligt utraditionel. Fremgangsmåden er naturligvis fusket, og har da også nogle lidt uheldige konsekvenser. Af ulemper kan nævnes:

1. Man mister det dobbelt-check som er normalt i HOL. Bevisfunktionen burde give en ekstra sikkerhed i beviserne, idet brugeren normalt laver 'backward proof', burde bevisfunktionen være det formelle bevis. Hvis man snyder sig igennem denne, mister man naturligvis sikkerhed.
2. Bevisfunktionen er lavet til at bruges, ikke misbruges. Det er normalt en funktion, som man kan bruge, efter man har bevist en sætning ved hjælp af bevistræet. Bevisfunktionen skal på denne måde kunne genbruges. En snydebevisfunktion som den vi har lavet, vil i bedste fald være ubrugelig.
3. HOL-etikken fra tidligere går fuldstændigt fløjten. mk_thm er en "farlig" operation. Det er klart, at man med denne funktion kan lave stort set hvad som helst, specielt ugyldige teoremer. Man skal derfor bruge den varsomt, hvis man vil undgå fejl.

Der er dog også fordele og formildende omstændigheder:

1. Det virker hver gang. Det kan simpelt hen ikke gå galt.
2. Det går hurtigt, hvilket er en stor fordel når man laver automatisk bevisførelse. Det går hurtigere, fordi dobbeltchecket bliver reduceret et stort set ingenting. Dette er en ulempe når man ønsker sikkerhed i systemet, men en fordel hvis man sidder og venter på et bevis.
3. Man kan lave de ting (taktikker) man har brug for. Dette er, som skitseret ovenfor svært hvis man ikke snyder.
4. Det er nemt, når man først har fået ideen.
5. Vi har ikke brug for en bevisfunktion i vores bevismaskine. Vores implementering er konsistent, hvilket vi har vist hele tre gange. Først har vi vist sundheden af vores gentzenregler (afsnit 2), dernæst har vi vist reglerne i HOL's egen logik (appendiks B) og til slut har vi vist, at vores udvidelse er konservativ ved at bevise reglerne på computeren ved lovlige slutningsregler (appendiks C). Alt dette gør, at vi føler os tilstrækkeligt sikre uden bevisfunktioner.

Kapitel 6

Konklusion

Vores konklusion falder i tre afsnit. Først vil vi give en kritik af den teori, som er beskrevet i projektet. Dernæst gennemgår vi fordele og ulemper ved HOL. Til slut samler vi op i en diskussion af vores arbejde's status.

Kritik af teori Vi har ikke nogen bemærkninger til selve det udsagnslogiske grundlag, udover dets begrænsede udsagnskraft, hvilket er diskuteret i kapitel 3. Teorien, som vi har valgt at arbejde med og implementeret, er Ginsbergs version af de kontrafaktiske konditionaler. En styrke og en svaghed ved teorien er det såkaldte badworldprædikat. Prædikatet ville, hvis det blev implementeret, gøre teorien væsentligt hurtigere at arbejde med. Problemet er som nævnt, at det er forholdsvis svært håndgribeligt. Vi mener, at prædikatet nødvendigvis må være kontekstafhængigt. Det er afhængigt af hvilken situation man ønsker at implementere. I vores eksempel med Scandinavian Star, skulle et eventuelt badworldprædikat fjerne verdener hvor ilden ikke kunne brede sig, eller hvor røgen ikke udgjorde nogen fare. Disse ting er knyttet til det specifikke scenarium, og ville ikke nødvendigvis kunne overføres til andre. I det enkelte tilfælde er det også meget svært at måle similaritet mellem verdener. Vores bud på en ordning af verdenerne med hensyn til similaritet, er at inddrage nogle af teknikkerne fra Inexact Reasoning. Vi forestiller os, at man kunne vægte de enkelte udsagn, der indgår i en given sammenhæng, og på den måde afgøre similaritet. Problemerne, der er forbundet med en sådan udvidelse, synes store og ligger iøvrigt udenfor dette projekts rammer.

Kritik af HOL Vores kritik af HOL går først og fremmest på den usædvanligt dårlige dokumentation. Manualerne giver ikke nogen særlig stor hjælp og er iøvrigt mangelfulde. Målgruppen er selvfølgelig heller ikke den samme som for almindelige kommercielle programmer. Det er et begrænset antal mennesker, der har brug for at programmere i højere ordens logik og lambda kalkylen. HOL er udviklet til Hardware Verification, altså testning af chips til datamater. Hvorledes HOL tackler den slags problemstillinger, ved vi ikke, men det er ihvertfald ikke ligetil, at få HOL til at udføre udsagnslogiske beviser. En af grundene er udbudet af taktikker. Udvælgelsen og navngivningen af taktikker må være sket på baggrund af problemstillinger indenfor Hardware Verification. Vi har således været nødsaget til at programmere vores egne taktikker. Taktikkerne, som vi har programmeret, er logisk korrekte, hvilket vi har vist dels i afsnittet om gentzenlogik dels i appendix, hvor vi har bevist taktikkerne i HOL-logik. Det er os dog stadig en gåde hvorfor så simple taktikker ikke er til rådighed i standard-HOL.

Status Inden for Artificial Intelligence foregår der for tiden et stort arbejde med at formalisere logiske følgeslutninger. Man er i den forbindelse interesserede i at formalisere ikke-monotone logikker. Det er i lyset af dette, at vores arbejde har sin berettigelse. Det, der er kommet ud af vores arbejde, er en implementering af en ikke monoton logik. Et af problemerne med vores program er dets langsommelighed. Det tager lang tid at undersøge konsistens af en mængde af udsagn, og meget lang tid at undersøge alle delmængder af en mængde for konsistens. Vores arbejde er en prototype, og skal ikke betragtes som nogen form for ekspertsystem. Vi har således ikke forsøgt at optimere programmet i nogen nævneværdig grad.

Appendiks A

Kildetekst

```
set_flag(`timing`,true);;
letref Udskriv_Bevis = false;;
top_print print_all_thm;;

% ************************************************%
% ************ Diverse hjaelpefunktioner **********%
% ************************************************%

letref Indryk = 0;;

let Tabs n =
    letref nn = n in
     while nn > 0 do (print_string `\5`;
                 nn := nn - 1);;

let Print_Hyps asl =
    let print as =
       (Tabs Indryk;
     print_string `   [ `;
     print_term (as);
     print_string ` ]`;
     print_newline())
    in
     do (map print (rev asl));;
```

```
let Print_Goal (asl,w) =
     Tabs Indryk;
     print_term w;  print_newline();
     Print_Hyps asl;  print_newline();;

let Mk_List (Term : term) =
     letref w,(x : term#term), TermList =
          Term, ("none:bool","none:bool"),[]
     in
     (while is_disj w do
          (x := dest_disj w;
               TermList := TermList@[fst x];
          w := snd x));
          TermList := TermList@[w];;

% ***************************************************%
% ************ Definition af Gentzen taktikker *******%
% ***************************************************%


% ***************************************************%
% #######  VENSTRE DISJUNKTION - GENTZEN LOGIK #######%
%
                A,S1 |- S2  B,S1 |- S2
                ----------------------
                    A \/ B,S1 |- S2
%

let V_DISJ_TAC Sekvent =
   letref Asl1,(w: term),Asl2,Asl3
                    = fst Sekvent, snd Sekvent,[],[]
   in
     (while ((not (is_disj (hd Asl1))) & (not Asl1 = nil)) do
                         (Asl2 := Asl2@[hd Asl1];
                          Asl1 := tl Asl1);
        let Disj = hd Asl1
        in
          Asl2 :=  [fst (dest_disj Disj)]@Asl2@(tl Asl1);
          Asl3 :=  [snd (dest_disj Disj)]@(tl Asl2);
          if Udskriv_Bevis
```

```
                then (print_string `\L`;
                     Indryk := Indryk + 1;
                     Print_Goal Sekvent;
                     Tabs Indryk;
                     print_string ` V_DISJ_TAC \L`
                     );
      ([(Asl2,w);(Asl3,w)],\[Th1;Th2].(mk_thm(Sekvent))));;


% ****************************************************%
% #######  VENSTRE IMPLIKATION - GENTZEN LOGIK #######%
%
          S1 |- A \/ S2  B,S1 |- S2
          ---------------------
             S1,A ==> B |- S2
%

let V_IMP_TAC Sekvent =
   letref Asl1,(w: term),Asl2,Asl3
                         = fst Sekvent, snd Sekvent,[],[]
   in
     (while (if (Asl1=nil) or (is_imp (hd Asl1)
                              &(not is_neg (hd Asl1)))
            then false else true)
     do
        (Asl2 := Asl2@[hd Asl1];
         Asl1 := tl Asl1);
        let Imp = if not (Asl1 = nil)
                     then hd Asl1 else NO_TAC Sekvent
        in
          Asl2 :=  Asl2@(tl Asl1);
          Asl3 :=  [snd (dest_imp Imp)]@Asl2;
          if Udskriv_Bevis
              then (print_string `\L`;
                   Indryk := Indryk + 1;
                   Print_Goal Sekvent;
                   Tabs Indryk;
                   print_string ` V_IMP_TAC \L`
                   );
```

```
    ([(Asl2,mk_disj (fst (dest_imp Imp),w));(Asl3,w)],
     \[Th1;Th2].(mk_thm(Sekvent)));;


% ******************************************************%
% #######  VENSTRE KONJUNKTION - GENTZEN LOGIK #######%
%
          S1,A,B |- S2
          -------------
          S1,A/\B |- S2
%

let V_KONJ_TAC Sekvent =
   letref Asl1,(w: term),Asl2 = fst Sekvent, snd Sekvent,[]
   in
     (while ((not Asl1 = nil) & ((not (is_conj (hd Asl1)))) ) do
               (Asl2 := Asl2@[hd Asl1];
                Asl1 := tl Asl1);
     let Konj = if not (Asl1 = nil)
                    then hd Asl1 else NO_TAC Sekvent
     in
          Asl2 :=  [fst (dest_conj Konj)]
                    @[snd (dest_conj Konj)]@Asl2@(tl Asl1));
          if Udskriv_Bevis
               then (print_string `\L`;
                    Indryk := Indryk + 0;
                    Print_Goal Sekvent;
                    Tabs Indryk;
                    print_string ` V_KONJ_TAC \L`
                    );
    ([(Asl2,w)],\[Th].(mk_thm(Sekvent)));;


% ******************************************************%
% ########  VENSTRE NEGATION - GENTZEN LOGIK ########%
%
          S1 |- A,S2
          -----------
          S1,~A |- S2
%
```

```
let V_NEG_TAC Sekvent =
   letref Asl1,(w: term),Asl2 = fst Sekvent, snd Sekvent,[]
   in
     (while ((not Asl1 = nil)&(not (is_neg (hd Asl1)))) do
               (Asl2 := Asl2@[hd Asl1];
                Asl1 := tl Asl1);
     let Neg = if not (Asl1 = nil)
                    then hd Asl1 else NO_TAC Sekvent
     in
          Asl2 :=  Asl2@(tl Asl1);
          if Udskriv_Bevis
               then (print_string '\L';
                     Indryk := Indryk + 0;
                     Print_Goal Sekvent;
                     Tabs Indryk;
                     print_string ' V_NEG_TAC \L'
                     );
     ([(Asl2,mk_disj (dest_neg Neg,w))],
                         \[Th].(mk_thm(Sekvent))));;


% ***************************************************%
% ####### HOEJRE KONJUNKTION - GENTZEN LOGIK #########%
%
          S1 |- A , S2    S1 |- B, S2
          ---------------------------
               S1 |- A /\ B, S2
%

let H_KONJ_TAC Sekvent =
   letref (Asl1: term list),(R_Term_Liste: term list),
          (S2: term list),w1,w2
          = fst Sekvent, Mk_List (snd Sekvent),
            [],"none:bool","none:bool"
   in
     (while ((not R_Term_Liste = nil)
               &(not (is_conj (hd R_Term_Liste))))
     do
          (S2 := S2@[(hd R_Term_Liste)];
```

```
          R_Term_Liste := tl R_Term_Liste);
      let Konj = if not (R_Term_Liste = nil)
                    then hd R_Term_Liste else NO_TAC Sekvent
      in
        w1,w2 := list_mk_disj
                   (S2@[fst (dest_conj Konj)]@(tl R_Term_Liste)),
                 list_mk_disj (S2@[snd (dest_conj Konj)]
                                  @(tl R_Term_Liste))
                    );
      if Udskriv_Bevis
           then (print_string `\L`;
                 Indryk := Indryk + 1;
                 Print_Goal Sekvent;
                 Tabs Indryk;
                 print_string ` H_KONJ_TAC \L`);
      ([(Asl1,w1);(Asl1,w2)],\[Th1;Th2].(mk_thm(Sekvent)));;


% ***************************************************%
% ####### HOEJRE IMPLIKATION - GENTZEN LOGIK #########%
%
                S1,A |- B , S2
          ---------------------------
               S1 |- A ==> B, S2
%

let H_IMP_TAC Sekvent =
   letref (Asl1: term list),(R_Term_Liste: term list),
          (S2: term list),w1,Asl2
          =
          fst Sekvent, Mk_List (snd Sekvent),
          [],"none:bool",[]
   in
     (while ((not (is_imp (hd R_Term_Liste)))
              & (not R_Term_Liste = nil))
      do
          (S2 := S2@[(hd R_Term_Liste)];
          R_Term_Liste := tl R_Term_Liste);
          let Imp = hd R_Term_Liste
          in
```

```
          w1,Asl2 := list_mk_disj
                          (S2@[snd (dest_imp Imp)]
                           @(tl R_Term_Liste)),
                          [(fst (dest_imp Imp))]@Asl1
                          );
          if Udskriv_Bevis
              then (print_string `\L`;
                    Indryk := Indryk + 0;
                    Print_Goal Sekvent;
                    Tabs Indryk;
                    print_string ` H_IMP_TAC \L`
                    );
      ([(Asl2,w1)],\[(Th1 :thm)].(mk_thm(Sekvent)));;


% **************************************************%
% ######### HOEJRE NEGATION - GENTZEN LOGIK #########%
%
              S1,A |- S2
          ----------------------
             S1 |- ~A, S2
%

let H_NEG_TAC Sekvent =
     if is_neg (snd Sekvent)
          then (DISCH_TAC Sekvent)
          else
            letref  (Asl1: term list),
                    (R_Term_Liste: term list),
                    (S2: term list),w1,Asl2
                    =
                    fst Sekvent, Mk_List (snd Sekvent),
                    [],"none:bool",[]
            in
               (while ((not (is_neg (hd R_Term_Liste)))
                       & (not R_Term_Liste = nil))
               do
                    (S2 := S2@[(hd R_Term_Liste)];
                    R_Term_Liste := tl R_Term_Liste);
                    let Neg = hd R_Term_Liste
```

```
                   in
                   w1,Asl2 := list_mk_disj
                                 (S2@(tl R_Term_Liste)),
                                 [(dest_neg Neg)]@Asl1
              );
         if Udskriv_Bevis
              then (print_string '\L';
                   Indryk := Indryk + 0;
                   Print_Goal Sekvent;
                   Tabs Indryk;
                   print_string ' H_NEG_TAC \L'
                   );
  ([(Asl2,w1)],\[(Th1 :thm)].(mk_thm(Sekvent)));;


% *****************************************************%
% ######### HOEJRE DISJUNKTION - GENTZEN LOGIK ##########%
% ############ CHECK_TAC - ER MAALET NAAET? #############%
%
Her defineres funktionen CHECK_TAC, der undersoeger om der i et
givet Sekvent optraeder samme udsagn paa baade hoejre og venstre
side af foelgeslutningen.
Af tekniske grunde repraesenteres lister af udsagn i succedenten
(hoejre siden) som disjunktioner.
I undersoegelsen splittes disse udsagn op i enkeltsstaaende
atomiske formler.
%

letrec Disj_List (T:term) =
  if (is_disj T)
   then fst(dest_disj T).Disj_List (snd(dest_disj T))
   else [T];;

let CHECK_TAC GOAL =
  if not (null (intersect (fst(GOAL))
               (Disj_List (snd GOAL))))
   then (([]:goal list), \ ([]:thm list).mk_thm(GOAL))
   else failwith 'CHECK_TAC';;
```

```
%***************************************************%
%####### AUTOMATISERING AF GENTZEN BEVISER #########%
%***************************************************%


let BEVIS_TAC Sekvent =
  let Taktik =  [CHECK_TAC;V_KONJ_TAC;V_NEG_TAC ;
                 H_NEG_TAC;H_IMP_TAC;
                 V_IMP_TAC;V_DISJ_TAC;
                 H_KONJ_TAC]
     in (FIRST Taktik Sekvent);;

let BEVIS() =
     letref Er_Bevist,x = false, (top_goal())
     in
     (Indryk := (0);
     e(REPEAT BEVIS_TAC);
     (x = top_goal()) ? (Er_Bevist := true);
     Er_Bevist);;

%***************************************************%
%****************** KONSISTENT *******************%
%***************************************************%
%* KONSISTENT undersoeger om en given maengde af udsagn *%
%* er konsistent. Givet en maengde W undersoeges om *****%
%* sekventen  W |- F  kan bevises.                  *****%
%* Hvis det kan bevises, saa er W inkonsistent.      ****%
%***************************************************%

let KONSISTENT W =
     set_goal(W,"F");
     not BEVIS();;

%***************************************************%
%*************** MULIGE_VERDENER *****************%
%***************************************************%
%*** MULIGE_VERDENER genererer alle de restriktioner  ***%
%*** af verdenen 'W', hvori det gaelder at det kontra- **%
%*** faktiske udsagn 'm'  kan tilfoejes uden at disse  **%
```

```
%*** verdener hver i saer bliver  inkonsistente ****%
%**************************************************%

let Ryd_Op W =
     letref V,t1,t2 = W,length W,0
     in
     while t1>1 do
          (t2:= t1-1;
         while t2 > 0 do
          V := if ((el t2 V) =
                    (intersect (el t1 V) (el t2 V)))
               then (t1 := t1-1;
                    (subtract V [(el t2 V)]))
               else V;
               t2 := t2 - 1);
          t1 := t1 - 1
          );
     V;;

let MULIGE_VERDENER (W,m: term) =
     letref (VerdensListe : term list list) = []
 in
  letrec MV (W,n) =
     letref p = n in
     if not (W = nil) then do (
          VerdensListe:=
          if KONSISTENT (W@[m]) then VerdensListe@[W]
          else
               (while p<length W do
                    (p:=p+1;
                     MV((subtract W [el p W]),p-1)
                    ); VerdensListe))
  in
     Udskriv_Bevis := false;
     MV (W,0);
     Ryd_Op VerdensListe;;


%**********************************************%
```

```
let KFK W p q =
     letref VerdensListe,Resultat,count
               = MULIGE_VERDENER (W,p), true, 1
     in
     while (Resultat & ((count -1)< (length VerdensListe)))
     do   (set_goal ((el count VerdensListe)@[p],q);
          Resultat := BEVIS();
          count := count + 1
          );
          Resultat;;
```

Appendiks B

Beviser for HOL-regler

Vi begynder med et bevis for, at disjunktionen på højresider kommuterer (DISJ_KOM). Dette betyder, at flere beviser kan simplificeres. For eksempel vil reglen H_KONJ, som vi har implementeret den, se således ud:

$$\frac{S1 \vdash S2 \vee A \vee S3 \quad S1 \vdash S2 \vee B \vee S3}{S1 \vdash S2 \vee (A \wedge B) \vee S3}$$

Men på grund af DISJ_KOM er det nok at vise reglen:

$$\frac{S1 \vdash A \vee S4 \quad S1 \vdash B \vee S4}{S1 \vdash (A \wedge B) \vee S4}$$

eftersom vi kan flytte rundt på højresiden som vi har lyst, og kan sætte $S4$ til at være $S2 \vee S3$. Negationsreglerne er vist begge veje, da de kan bruges i senere beviser.

DISJ_KOM

$$\vdash A \vee B \Longleftrightarrow B \vee A$$

$$\vdash A \vee B \quad \text{ASS}$$

$$\vdash \lambda t1t2.\ \forall x.\ (t1 \Longrightarrow x) \Longrightarrow ((t2 \Longrightarrow x) \Longrightarrow x)\ AB \quad \text{OR\_DEF}$$

$$\vdash \forall x.\,(A \Longrightarrow x) \Longrightarrow ((B \Longrightarrow x) \Longrightarrow x) \quad \text{BETA\_CONV}$$

$$\vdash (A \Longrightarrow x) \Longrightarrow ((B \Longrightarrow x) \Longrightarrow x) \quad \text{SPEC x}$$

$$A \Longrightarrow x \vdash (B \Longrightarrow x) \Longrightarrow x \quad \text{UNDISCH}$$

$$A \Longrightarrow x,\ B \Longrightarrow x \vdash x \quad \text{UNDISCH}$$

$$B \Longrightarrow x \vdash (A \Longrightarrow x) \Longrightarrow x \quad \text{DISCH}$$

$$\vdash (B \Longrightarrow x) \Longrightarrow ((A \Longrightarrow x) \Longrightarrow x) \quad \text{DISCH}$$

$$\vdash \forall x.\,(B \Longrightarrow x) \Longrightarrow ((A \Longrightarrow x) \Longrightarrow x) \quad \text{GEN x}$$

$$\vdash \lambda t1 t2.\,\forall x.\,(t1 \Longrightarrow x) \Longrightarrow ((t2 \Longrightarrow x) \Longrightarrow x)BA \quad \text{BETA\_CONV}$$

$$\vdash B \vee A \quad \text{OR\_DEF}$$

V_NEG

$$\frac{S1 \vdash A \vee S2}{S1,\ \neg A \vdash S2}$$

$$S1 \vdash A \vee S2 \quad \text{ASS}$$

$$S1 \vdash \neg A \Longrightarrow S2 \quad \text{DISJ\_IMP}$$

$$S1,\ \neg A \vdash S2 \quad \text{UNDISCH}$$

V_NEG_B

$$\frac{S1,\ \neg A \vdash S2}{S1 \vdash A \vee S2}$$

$$S1,\ \neg A \vdash S2 \quad \text{ASS}$$

$$S1 \vdash \neg A \Longrightarrow S2 \quad \text{DISCH}$$

$$S1 \vdash \neg\neg A \vee S2 \quad \text{IMP\_DISJ\_THM}$$

$$S1 \vdash A \vee S2 \quad \text{NOT\_CLAUSES}$$

H_NEG

$$\frac{S1,\ A \vdash S2}{S1 \vdash \neg A \vee S2}$$

$$S1,\ A \vdash S2 \quad \text{ASS}$$

$$S1 \vdash A \Longrightarrow S2 \quad \text{DISCH}$$

$$S1 \vdash \neg A \vee S2 \quad \text{IMP\_DISJ\_THM}$$

H_NEG_B

$$\frac{S1 \vdash \neg A \vee S2}{S1,\ A \vdash S2}$$

$$S1 \vdash \neg A \vee S2 \quad \text{ASS}$$

$$S1 \vdash A \Longrightarrow S2 \quad \text{IMP\_DISJ\_THM}$$

$$S1,\ A \vdash S2 \quad \text{UNDISCH}$$

V_KONJ

$$\frac{S1,\ A,\ B \vdash S2}{S1,\ A \wedge B \vdash S2}$$

$$S1,\ A,\ B \vdash S2 \quad \text{ASS}$$

$$S1,\ A \vdash B \Longrightarrow S2 \quad \text{DISCH}$$

$$S1,\ A \vdash \neg B \vee S2 \quad \text{IMP\_DISJ\_THM}$$

$$S1 \vdash A \Longrightarrow (\neg B \vee S2) \quad \text{DISCH}$$

$$S1 \vdash \neg A \vee \neg B \vee S2 \quad \text{IMP\_DISJ\_THM}$$

$$S1 \vdash \neg(A \wedge B) \vee S2 \quad \text{DE\_MORGAN\_THM}$$

$$S1,\ A \wedge B \vdash S2 \quad \text{H\_NEG\_B}$$

H_KONJ

$$\frac{S1 \vdash A \vee S2 \quad S1 \vdash B \vee S2}{S1 \vdash (A \wedge B) \vee S2}$$

$$S1 \vdash A \vee S2 \quad \text{ASS}$$

$$S1 \vdash \neg S2 \Longrightarrow A \quad \text{DISJ\_IMP}$$

$$S1,\ \neg S2 \vdash A \quad \text{(i) UNDISCH}$$

$$S1 \vdash B \vee S2 \quad \text{ASS}$$

$$S1 \vdash \neg S2 \Longrightarrow B \quad \text{DISJ\_IMP}$$

$$S1,\ \neg S2 \vdash B \quad \text{(ii) UNDISCH}$$

$$S1,\ \neg S2 \vdash A \wedge B \quad \text{CONJ (i) (ii)}$$

$$S1 \vdash (A \wedge B) \vee S2 \quad \text{V\_NEG\_B}$$

V_DISJ

$$\frac{S1,\ A \vdash S2 \quad S1,\ B \vdash S2}{S1,\ A \vee B \vdash S2}$$

$$S1,\ A \vdash S2 \quad \text{ASS}$$

$$S1 \vdash A \Longrightarrow S2 \quad \text{(i) DISCH}$$

$$S1,\ B \vdash S2 \quad \text{ASS}$$

$$S1 \vdash B \Longrightarrow S2 \quad \text{(ii) DISCH}$$

$$S1 \vdash (A \vee B) \Longrightarrow S2 \quad \text{OR\_IMP (i) (ii)}$$

$$S1,\ A \vee B \vdash S2 \quad \text{UNDISCH}$$

V_IMP

$$\frac{S1 \vdash A \vee S2 \quad S1,\ B \vdash S2}{S1,\ A \Longrightarrow B \vdash S2}$$

$$S1 \vdash A \vee S2 \quad \text{ASS}$$

$$S1 \vdash \neg A \Longrightarrow S2 \quad \text{DISJ\_IMP}$$

$$S1,\ \neg A \vdash S2 \quad \text{(i) UNDISCH}$$

$$S1,\ B \vdash S2 \quad \text{(ii) ASS}$$

$$S1,\ \neg A \vee B \vdash S2 \quad \text{V\_DISJ (i) (ii)}$$

$$S1,\ A \Longrightarrow B \vdash S2 \quad \text{IMP\_DISJ\_THM}$$

H_IMP

$$\frac{S1,\ A \vdash B \vee S2}{S1 \vdash A \Longrightarrow B \vee S2}$$

$S1,\ A \vdash B \vee S2$ ASS

$S1,\ A \vdash \neg S2 \Longrightarrow B$ DISJ_IMP

$S1,\ \neg S2,\ A \vdash B$ UNDISCH

$S1,\ \neg S2 \vdash A \Longrightarrow B$ DISCH

$S1 \vdash (A \Longrightarrow B) \vee S2$ V_NEG_B

Appendiks C

Beviserne lavet i HOL

```
*********** DISJ_KOM *******************

#let th1 = mk_thm
([],"a\/b");;
#th1 = |- a \/ b
Intermediate theorems generated: 1

#let th2 = REWRITE_RULE [OR_DEF] th1;;
th2 = |- (\t1 t2. !t. (t1 ==> t) ==> (t2 ==> t) ==> t)a b
Intermediate theorems generated: 111

#let th3 = BETA_RULE th2;;
th3 = |- !t. (a ==> t) ==> (b ==> t) ==> t
Intermediate theorems generated: 86

#let th4 = SPEC "t :bool" th3;;
th4 = |- (a ==> t) ==> (b ==> t) ==> t
Intermediate theorems generated: 1

#let th5 = UNDISCH th4;;
th5 = a ==> t |- (b ==> t) ==> t
Intermediate theorems generated: 2

#let th6 = UNDISCH th5;;
th6 = a ==> t, b ==> t |- t
Intermediate theorems generated: 2
```

```
#let th7 = DISCH "a==>t" th6;;
th7 = b ==> t |- (a ==> t) ==> t
Intermediate theorems generated: 1

#let th8 = DISCH "b==>t" th7;;
th8 = |- (b ==> t) ==> (a ==> t) ==> t
Intermediate theorems generated: 1


#let spe = "(\t1 t2. !t. (t1 ==> t) ==> (t2 ==> t) ==> t)b a";;
spe = "(\t1 t2. !t. (t1 ==> t) ==> (t2 ==> t) ==> t)b a" : term

#let spe1 = DEPTH_CONV BETA_CONV spe;;
spe1 = |- (\t1 t2. !t. (t1 ==> t) ==> (t2 ==> t) ==> t)b a
= (!t. (b ==> t) ==> (a ==> t) ==> t)
Intermediate theorems generated: 85

#let spe2 = SYM spe1;;
spe2 = |- (!t. (b ==> t) ==> (a ==> t) ==> t)
= (\t1 t2. !t. (t1 ==> t) ==> (t2 ==> t) ==> t)b a
Intermediate theorems generated: 1

#let th9 = GEN "t :bool" th8;;
th9 = |- !t. (b ==> t) ==> (a ==> t) ==> t
Intermediate theorems generated: 1

#let th10 = SUBS [spe2] th9;;
th10 = |- (\t1 t2. !t. (t1 ==> t) ==> (t2 ==> t) ==> t)b a
Intermediate theorems generated: 1

#let bellera = SYM OR_DEF;;
bellera = |- (\t1 t2. !t. (t1 ==> t) ==> (t2 ==> t) ==> t) = $\/
Intermediate theorems generated: 1

#let th11 = REWRITE_RULE [bellera] th10;;
th11 = |- b \/ a
Intermediate theorems generated: 24
```

```
*********** V_NEG ***********************

#let th1 = mk_thm (["s1 :bool"],"a\/s2");;
th1 = s1 |- a \/ s2
Intermediate theorems generated: 1

#let th2 = DISJ_IMP th1;;
th2 = s1 |- ~a ==> s2
Intermediate theorems generated: 7

#let th3 = UNDISCH th2;;
th3 = s1, ~a |- s2
Intermediate theorems generated: 2


*********** V_NEG_B *********************

#let th1 = mk_thm (["s1 :bool";"~a"],"s2 :bool");;
th1 = s1, ~a |- s2
Intermediate theorems generated: 1

#let th2 = DISCH "~a" th1;;
th2 = s1 |- ~a ==> s2
Intermediate theorems generated: 1

#let th3 = REWRITE_RULE [IMP_DISJ_THM] th2;;
Theorem IMP_DISJ_THM autoloaded from theory `arithmetic`.
IMP_DISJ_THM = |- !t1 t2. t1 ==> t2 = ~t1 \/ t2

th3 = s1 |- a \/ s2
Intermediate theorems generated: 35


*********** H_NEG *****************

#let th1 = mk_thm (["s1 :bool";"a :bool"],"s2 :bool");;
th1 = s1, a |- s2
Intermediate theorems generated: 1
```

```
#let th2 = DISCH "a :bool" th1;;
th2 = s1 |- a ==> s2
Intermediate theorems generated: 1

#let th3 = REWRITE_RULE [IMP_DISJ_THM] th2;;
th3 = s1 |- ~a \/ s2
Intermediate theorems generated: 39


*********** H_NEG_B *****************

#let th1 = mk_thm (["s1 :bool"],"~a \/ s2");;
th1 = s1 |- ~a \/ s2
Intermediate theorems generated: 1

#let th2 = DISJ_IMP th1;;
th2 = s1 |- ~~a ==> s2
Intermediate theorems generated: 7

#let th3 = REWRITE_RULE [NOT_CLAUSES] th2;;
th3 = s1 |- a ==> s2
Intermediate theorems generated: 33

#let th4 = UNDISCH th3;;
th4 = s1, a |- s2
Intermediate theorems generated: 2


************ V_KONJ *****************

let th1 = mk_thm ((["s1 :bool";"a :bool";"b :bool"],"s2
:bool"));;
th1 = s1, a, b |- s2
Intermediate theorems generated: 1

#let th2 = DISCH "b :bool" th1;;
th2 = s1, a |- b ==> s2
Intermediate theorems generated: 1
```

```
#let th3 = REWRITE_RULE [IMP_DISJ_THM] th2;;
th3 = s1, a |- ~b \/ s2
Intermediate theorems generated: 39

#let th4 = DISCH "a :bool" th3;;
th4 = s1 |- a ==> ~b \/ s2
Intermediate theorems generated: 1

#let th5 = REWRITE_RULE [IMP_DISJ_THM] th4;;
th5 = s1 |- ~a \/ ~b \/ s2
Intermediate theorems generated: 63

#let th6 = mk_thm ([],"a\/(b\/c) = (a\/b)\/c");;
th6 = |- a \/ b \/ c = (a \/ b) \/ c
Intermediate theorems generated: 1

#let th7 = REWRITE_RULE [th6] th5;;
th7 = s1 |- (~a \/ ~b) \/ s2
Intermediate theorems generated: 64

#let spe = SPEC "a : bool" DE_MORGAN_THM;;
spe = |- !t2. (~(a /\ t2) = ~a \/ ~t2) /\ (~(a \/ t2) = ~a /\
~t2)
Intermediate theorems generated: 1

#let spe2 = SPEC "b : bool" spe;;
spe2 = |- (~(a /\ b) = ~a \/ ~b) /\ (~(a \/ b) = ~a /\ ~b)
Intermediate theorems generated: 1

#let spe3 = CONJUNCTS spe2;;
spe3 = [|- ~(a /\ b) = ~a \/ ~b; |- ~(a \/ b) = ~a /\ ~b] : thm
list
Intermediate theorems generated: 2

#let spe4 = hd spe3;;
spe4 = |- ~(a /\ b) = ~a \/ ~b

#let spe5 = SYM spe4;;
spe5 = |- ~a \/ ~b = ~(a /\ b)
Intermediate theorems generated: 1
```

```
#let th8 = SUBS [spe5] th7;;
th8 = s1 |- ~(a /\ b) \/ s2
Intermediate theorems generated: 1

#let th9 = DISJ_IMP th8;;
th9 = s1 |- ~~(a /\ b) ==> s2
Intermediate theorems generated: 7

#let th10 = REWRITE_RULE [NOT_CLAUSES] th9;;
th10 = s1 |- a /\ b ==> s2
Intermediate theorems generated: 49

#let th11 = UNDISCH th10;;
th11 = s1, a /\ b |- s2
Intermediate theorems generated: 2


************ H_KONJ *****************

#let th1 = mk_thm(["s1 :bool"],"s2\/a");;
th1 = s1 |- s2 \/ a
Intermediate theorems generated: 1

#let th2 = DISJ_IMP th1;;
th2 = s1 |- ~s2 ==> a
Intermediate theorems generated: 7

#let th3 = UNDISCH th2;;
th3 = s1, ~s2 |- a
Intermediate theorems generated: 2

#let th4 = mk_thm(["s1 :bool"],"s2\/b");;
th4 = s1 |- s2 \/ b
Intermediate theorems generated: 1

#let th5 = DISJ_IMP th4;;
th5 = s1 |- ~s2 ==> b
Intermediate theorems generated: 7
```

```
#let th6 = UNDISCH th5;;
th6 = s1, ~s2 |- b
Intermediate theorems generated: 2

#let th7 = CONJ th3 th6;;
th7 = s1, ~s2 |- a /\ b
Intermediate theorems generated: 1

#let th8 = DISCH "~s2" th7;;
th8 = s1 |- ~s2 ==> a /\ b
Intermediate theorems generated: 1

#let th9 = REWRITE_RULE [IMP_DISJ_THM] th8;;
th9 = s1 |- s2 \/ a /\ b
Intermediate theorems generated: 51


************* V_DISJ ******************

#let th1 = mk_thm (["s1 :bool";"a :bool"],"s2 :bool");;
th1 = s1, a |- s2
Intermediate theorems generated: 1

#let th2 = DISCH "a :bool" th1;;
th2 = s1 |- a ==> s2
Intermediate theorems generated: 1


#let th3 = mk_thm (["s1 :bool";"b :bool"],"s2 :bool");;
th3 = s1, b |- s2
Intermediate theorems generated: 1

#let th4 = DISCH "b :bool" th3;;
th4 = s1 |- b ==> s2
Intermediate theorems generated: 1

#let spe = SPEC "a :bool" OR_IMP;;
spe = |- !b c. a \/ b ==> c = (a ==> c) /\ (b ==> c)
Intermediate theorems generated: 1
```

```
#let spe1 = SPEC "b :bool" spe;;
spe1 = |- !c. a \/ b ==> c = (a ==> c) /\ (b ==> c)
Intermediate theorems generated: 1

#let spe2 = SPEC "s2 :bool" spe1;;
spe2 = |- a \/ b ==> s2 = (a ==> s2) /\ (b ==> s2)
Intermediate theorems generated: 1

#let spe3 = SYM spe2;;
spe3 = |- (a ==> s2) /\ (b ==> s2) = a \/ b ==> s2
Intermediate theorems generated: 1

#let th5 = CONJ th2 th4;;
th5 = s1 |- (a ==> s2) /\ (b ==> s2)
Intermediate theorems generated: 1

#let th6 = REWRITE_RULE [spe3] th5;;
th6 = s1 |- a \/ b ==> s2
Intermediate theorems generated: 48

#let th7 = UNDISCH th6;;
th7 = s1, a \/ b |- s2
Intermediate theorems generated: 2


********* V_IMP ********************

#let th1 = mk_thm (["s1 :bool"],"a\/s2");;
th1 = s1 |- a \/ s2
Intermediate theorems generated: 1

#let th2 = DISJ_IMP th1;;
th2 = s1 |- ~a ==> s2
Intermediate theorems generated: 7


#let th3 = mk_thm (["s1 :bool";"b :bool"],"s2 :bool");;
th3 = s1, b |- s2
Intermediate theorems generated: 1
```

```
#let th4 = DISCH "b :bool" th3;;
th4 = s1 |- b ==> s2
Intermediate theorems generated: 1

#let th5 = CONJ th2 th4;;
th5 = s1 |- (~a ==> s2) /\ (b ==> s2)
Intermediate theorems generated: 1

#spe3;;
|- (a ==> s2) /\ (b ==> s2) = a \/ b ==> s2
Run time: 0.0s
```

spe3 fra V_DISJ kan bruges igen.

```
#let th6 = REWRITE_RULE [spe3] th5;;
th6 = s1 |- ~a \/ b ==> s2
Intermediate theorems generated: 56

#let spe = SPEC "a :bool" IMP_DISJ_THM;;
spe = |- !t2. a ==> t2 = ~a \/ t2
Intermediate theorems generated: 1

#let spe1 = SPEC "b :bool" spe;;
spe1 = |- a ==> b = ~a \/ b
Intermediate theorems generated: 1

#let spe2 = SYM spe1;;
spe2 = |- ~a \/ b = a ==> b
Intermediate theorems generated: 1

#let th7 = REWRITE_RULE [spe2] th6;;
th7 = s1 |- (a ==> b) ==> s2
Intermediate theorems generated: 47

#let th8 = UNDISCH th7;;
th8 = s1, a ==> b |- s2
Intermediate theorems generated: 2
```

```
*********** H_IMP *****************

#let th1 = mk_thm (["s1 :bool";"a :bool"],"s2\/b");;
th1 = s1, a |- s2 \/ b
Intermediate theorems generated: 1

#let th2 = DISJ_IMP th1;;
th2 = s1, a |- ~s2 ==> b
Intermediate theorems generated: 7

#let th3 = UNDISCH th2;;
th3 = s1, a, ~s2 |- b
Intermediate theorems generated: 2

#let th4 = DISCH "a :bool" th3;;
th4 = s1, ~s2 |- a ==> b
Intermediate theorems generated: 1

#let th5 = DISCH "~s2" th4;;
th5 = s1 |- ~s2 ==> a ==> b
Intermediate theorems generated: 1

#let th6 = ONCE_REWRITE_RULE [IMP_DISJ_THM] th5;;
th6 = s1 |- ~~s2 \/ (a ==> b)
Intermediate theorems generated: 10

#let th7 = REWRITE_RULE [NOT_CLAUSES] th6;;
th7 = s1 |- s2 \/ (a ==> b)
Intermediate theorems generated: 49
```

Appendiks D

Litteraturliste

D.1 Bøger

DAV 83 Davis, Martin D. & Weyuker, Elaine J.
Computability, Complexity, and Languages
Academic Press, 1983.

GAL 86 Gallier, Jean H.
Logic for computer science
Harper & Row, New York 1986.

ETH 88 Etherington, David W.
Reasoning with incomplete information
Pitman, London 1988.

HIN 86 Hindley, J. Roger & Seldin, Jonathan P.
Introduction to λ-Calculus
Cambridge University Press, Cambridge 1986.

LEW 73 Lewis, David
Counterfactuals
Blackwell 1973.

MEN 79 Mendelson, Elliott
Introduction to mathematical Logic
D. Van Nostrand, New York 1979.

ROS 89 Rossen, Lars
HOL Formalisation of RUBY
Instituttet for datateknik DTH 1989.

D.2 Tidsskrifter

GIN 86 Ginsberg, M. L.
Counterfactuals
Artificial Intelligence 30, 1986.

GOR 79 Gordon Michael J. et.al.
Lecture Notes in Computer Science 78: Edinburgh LCF
Springer-Verlag 1979

STA 75 Stalnaker
Causation and Conditionals
Oxford University Press, London 1975.

Brugervejledning.

Programmet er lavet til at evaluere kontrafaktiske konditionaler. For beskrivelse af kontrafaktiske konditionaler samt evalueringsalgoritme henvises til teksten i projektet. Her vil vi kort beskrive de funktioner, som vi har programmeret. Det grundlæggende bevissystem som vi har mekaniseret er gentzen-kalkylen for et udsagnslogisk sprog med de logiske konnektiver /\ (OG), \/ (ELLER), ==> (IMPLIKATION) og ~ (NEGATION). Bevisalgoritmen kan bevise alle valide formler i dette sprog, hvis formlen er falsificerbar udskrives den del af bevistræet hvor den falsificerende variabeltilskrivning direkte kan aflæses. For beskrivelse af bevissystemet se projektet.

Det første der skal gøres når man ønsker at bevise en given sekvent er at definere sekventen for maskinen ved hjælp af funktionen **set_goal**. Funktionen tager sekventer som input. Sekventerne kan have vilkårligt mange formler på venstresiden men kun en formel på højresiden. Et eksempel:

```
set_goal(["A/\B";"C:bool"],"A==>C");;
```

set_goal tager altså som input et par hvor det første element skal være en liste (med kantede paranteser omkring og de enkelte elementer adskilt med semikolon ;) det andet element skal blot være en term. Termerne er de udsagnslogiske formler omgivet af citationstegn. Som det ses i eksemplet skal man i nogle tilfælde angive typen af det udtryk man arbejder med. I tilfælde hvor HOL ikke selv direkte kan aflæse typen (C i eksemplet), skal den angives.

Når goal'et er defineret kan man begynde at benytte gentzenreglerne. I forlængelse af det foregående eksempel kunne man skrive:

```
e(V_KONJ_TAC);;
```

Dette vil bevirke at reglen udføres på det aktuelle goal, hvorefter bevistræet opdateres i henhold til gentzen-reglerne. e'et ovenfor er HOL's expand-funktion, der skal benyttes når man ønsker at anvende en regel på goal'et. Der sker ikke noget, hvis man forsøger at benytte en regel på goal'et, der ikke duer. HOL svarer med en fejlmelding, og man kan forsøge med en ny regel. Reglerne der er til rådighed i vores programpakke er følgende:

1) V_IMP_TAC
2) V_KONJ_TAC
3) V_NEG_TAC
4) V_DISJ_TAC
5) H_IMP_TAC
6) H_KONJ_TAC
7) H_NEG_TAC

Man kan også i stedet for at benytte reglerne som de står ovenfor bruge funktionen **BEVIS_TAC**. Denne funktion applicerer blot den første regel, der virker. Reglerne tjekkes efter en liste, hvor de er ordnet i rækkefølgen 2,3,7,5,1,4,6.

BEVIS_TAC kan anvendes på følgende vis:

e(BEVIS_TAC);;

Man kan også vælge at få beviset kørt automatisk. Dette gøres ved at skrive:

BEVIS();;

Hvis dette skrives vil det aktuelle goal blive forsøgt bevist.

Vi har nu beskrevet de grundlæggende funktioner i vores automatiserede bevissystem. Vi har ovenpå bevismaskinen lavet nogle funktioner, der kan bruges til at tjekke konsistens af mængder, danne konsistente delmængder og evaluere kontrafaktiske konditionaler. **KONSISTENT** er en funktion, der givet en liste af termer, undersøger om de udsagnslogiske formler indeholdt i termerne udgør en konsistent mængde. Eks:

KONSISTENT ["A:bool";"A==>B";"~B"];;

Funktionen giver som output enten true eller false, alt efter om mængden er konsistent eller inkonsistent. I eksemplet ovenfor vil maskinen svare false.

MULIGE_VERDENER er en funktion, der som input tager en liste af termer plus en term (et goal). Funktionen danner udfra listen og termen alle de konsistente delmængder af listen, der ikke er i modstrid med termen. Den udskriver dog ikke de delmængder, der er restriktioner af andre konsistente delmængder. Eks:

MULIGE_VERDENER(["A:bool";"A==>B"],"~B");;

Som output på eksemplet vil man få [["A==>B"];["A"]], altså en liste af lister. Den sidste og afgørende funktion, som vi har programmeret hedder **KFK**. Denne funktion benytter alle de foregående funktioner. Som input tager funktionen en liste (verden) og to termer. Eks:

KFK ["~P";"P/\N==>Q";"N:bool";"~(N/\R)"] "P:bool" "Q:bool";;

KFK vil givet udtrykket i eksemplet evaluere det kontrafaktiske udsagn P ▸ Q i den verden, der er beskrevet i listen.

1/78 "TANKER OM EN PRAKSIS" - et matematikprojekt.
Projektrapport af: Anne Jensen, Lena Lindenskov, Marianne Kesselhahn og Nicolai Lomholt.
Vejleder: Anders Madsen

2/78 "OPTIMERING" - Menneskets forøgede beherskelsesmuligheder af natur og samfund.
Projektrapport af: Tom J. Andersen, Tommy R. Andersen, Gert Krenøe og Peter H. Lassen
Vejleder: Bernhelm Boss.

3/78 "OPGAVESAMLING", breddekursus i fysik.
Af: Lasse Rasmussen, Aage Bonde Kræmmer og Jens Højgaard Jensen.

4/78 "TRE ESSAYS" - om matematikundervisning, matematiklæreruddannelsen og videnskabsrindalismen.
Af: Mogens Niss
Nr. 4 er p.t. udgået.

5/78 "BIBLIOGRAFISK VEJLEDNING til studiet af DEN MODERNE FYSIKS HISTORIE".
Af: Helge Kragh.
Nr. 5 er p.t. udgået.

6/78 "NOGLE ARTIKLER OG DEBATINDLÆG OM - læreruddannelse og undervisning i fysik, og - de naturvidenskabelige fags situation efter studenteroprøret".
Af: Karin Beyer, Jens Højgaard Jensen og Bent C. Jørgensen.

7/78 "MATEMATIKKENS FORHOLD TIL SAMFUNDSØKONOMIEN".
Af: B.V. Gnedenko.
Nr. 7 er udgået.

8/78 "DYNAMIK OG DIAGRAMMER". Introduktion til energy-bond-graph formalismen.
Af: Peder Voetmann Christiansen.

9/78 "OM PRAKSIS' INDFLYDELSE PÅ MATEMATIKKENS UDVIKLING". - Motiver til Kepler's: "Nova Stereometria Doliorum Vinariom".
Projektrapport af: Lasse Rasmussen.
Vejleder: Anders Madsen.

10/79 "TERMODYNAMIK I GYMNASIET".
Projektrapport af: Jan Christensen og Jeanne Mortensen.
Vejledere: Karin Beyer og Peder Voetmann Christiansen.

11/79 "STATISTISKE MATERIALER".
Af: Jørgen Larsen.

12/79 "LINEÆRE DIFFERENTIALLIGNINGER OG DIFFERENTIALLIGNINGSSYSTEMER".
Af: Mogens Brun Heefelt.
Nr. 12 er udgået.

13/79 "CAVENDISH'S FORSØG I GYMNASIET".
Projektrapport af: Gert Kreinøe.
Vejleder: Albert Chr. Paulsen.

14/79 "BOOKS ABOUT MATHEMATICS: History, Philosophy, Education, Models, System Theory, and Works of".
Af: Else Høyrup.
Nr. 14 er p.t. udgået.

15/79 "STRUKTUREL STABILITET OG KATASTROFER i systemer i og udenfor termodynamisk ligevægt".
Specialeopgave af: Leif S. Striegler.
Vejleder: Peder Voetmann Ckristiansen.

16/79 "STATISTIK I KRÆFTFORSKNINGEN".
Projektrapport af: Michael Olsen og Jørn Jensen.
Vejleder: Jørgen Larsen.

17/79 "AT SPØRGE OG AT SVARE i fysikundervisningen".
Af: Albert Christian Paulsen.

18/79 "MATHEMATICS AND THE REAL WORLD", Proceedings af an International Workshop, Roskilde University Centre, Denmark, 1978. Preprint.
Af: Bernhelm Booss og Mogens Niss (eds.)

19/79 "GEOMETRI, SKOLE OG VIRKELIGHED".
Projektrapport af: Tom J. Andersen, Tommy R. Andersen og Per H.H. Larsen.
Vejleder: Mogens Niss.

20/79 "STATISTISKE MODELLER TIL BESTEMMELSE AF SIKRE DOSER FOR CARCINOGENE STOFFER".
Projektrapport af: Michael Olsen og Jørn Jensen.
Vejleder: Jørgen Larsen

21/79 "KONTROL I GYMNASIET-FORMÅL OG KONSEKVENSER".
Projektrapport af: Crilles Bacher, Per S.Jensen, Preben Jensen og Torben Nysteen.

22/79 "SEMIOTIK OG SYSTEMEGENSKABER (1)".
1-port lineært response og støj i fysikken.
Af: Peder Voetmann Christiansen.

23/79 "ON THE HISTORY AF EARLY WAVE MECHANICS - with special emphasis on the role af realitivity".
Af: Helge Kragh.

24/80 a+b "MATEMATIKOPFATTELSER HOS 2.G'ERE".
1. En analyse. 2. Interviewmateriale.
Projektrapport af: Jan Christensen og Knud Lindhardt Rasmussen.
Vejleder: Mogens Niss.

25/80 "EKSAMENSOPGAVER", Dybdemodulet/fysik 1974-79.

26/80 "OM MATEMATISKE MODELLER".
En projektrapport og to artikler.
Af: Jens Højgaard Jensen m.fl.

27/80 "METHODOLOGY AND PHILOSOPHY AF SCIENCE IN PAUL DIRAC's PHYSICS".
Af: Helge Kragh.

28/80 "DIELEKTRISK RELAXATION - et forslag til en ny model bygget på væskernes viscoelastiske egenskaber".
Projektrapport af: Gert Kreinøe.
Vejleder: Niels Boye Olsen.

29/80 "ODIN - undervisningsmateriale til et kursus i differentialligningsmodeller".
Projektrapport af: Tommy R. Andersen, Per H.H. Larsen og Peter H. Lassen.
Vejleder: Mogens Brun Heefelt.

30/80 "FUSIONSENERGIEN - - - ATOMSAMFUNDETS ENDESTATION".
Af: Oluf Danielsen.
Nr. 30 er udgået.

31/80 "VIDENSKABSTEORETISKE PROBLEMER VED UNDERVISNINGSSYSTEMER BASERET PÅ MÆNGDELÆRE".
Projektrapport af: Troels Lange og Jørgen Karrebæk.
Vejleder: Stig Andur Pedersen.
Nr. 31 er p.t. udgået.

32/80 "POLYMERE STOFFERS VISCOELASTISKE EGENSKABER - BELYST VED HJÆLP AF MEKANISKE IMPEDANSMÅLINGER MÖSSBAUEREFFEKTMÅLINGER".
Projektrapport af: Crilles Bacher og Preben Jensen.
Vejledere: Niels Boye Olsen og Peder Voetmann Christiansen.

33/80 "KONSTITUERING AF FAG INDEN FOR TEKNISK - NATURVIDENSKABELIGE UDDANNELSER. I-II".
Af: Arne Jakobsen.

34/80 "ENVIRONMENTAL IMPACT AF WIND ENERGY UTILIZATION".
ENERGY SERIES NO. I.
Af: Bent Sørensen
Nr. 34 er udgået.

35/80 "HISTORISKE STUDIER I DEN NYERE ATOMFYSIKS UDVIKLING".
Af: Helge Kragh.

36/80 "HVAD ER MENINGEN MED MATEMATIKUNDERVISNINGEN?".
Fire artikler.
Af: Mogens Niss.

37/80 "RENEWABLE ENERGY AND ENERGY STORAGE".
ENERGY SERIES NO. 2.
Af: Bent Sørensen.

38/81 "TIL EN HISTORIETEORI OM NATURERKENDELSE, TEKNOLOGI OG SAMFUND".
Projektrapport af: Erik Gade, Hans Hedal, Henrik Lau og Finn Physant.
Vejledere: Stig Andur Pedersen, Helge Kragh og Ib Thiersen.
Nr. 38 er p.t. udgået.

39/81 "TIL KRITIKKEN AF VÆKSTØKONOMIEN".
Af: Jens Højgaard Jensen.

40/81 "TELEKOMMUNIKATION I DANMARK - oplæg til en teknologivurdering".
Projektrapport af: Arne Jørgensen, Bruno Petersen og Jan Vedde.
Vejleder: Per Nørgaard.

41/81 "PLANNING AND POLICY CONSIDERATIONS RELATED TO THE INTRODUCTION OF RENEWABLE ENERGY SOURCES INTO ENERGY SUPPLY SYSTEMS".
ENERGY SERIES NO. 3.
Af: Bent Sørensen.

42/81 "VIDENSKAB TEORI SAMFUND - En introduktion til materialistiske videnskabsopfattelser".
Af: Helge Kragh og Stig Andur Pedersen.

43/81 1."COMPARATIVE RISK ASSESSMENT OF TOTAL ENERGY SYSTEMS".
2."ADVANTAGES AND DISADVANTAGES OF DECENTRALIZATION".
ENERGY SERIES NO. 4.
Af: Bent Sørensen.

44/81 "HISTORISKE UNDERSØGELSER AF DE EKSPERIMENTELLE FORUDSÆTNINGER FOR RUTHERFORDS ATOMMODEL".
Projektrapport af: Niels Thor Nielsen.
Vejleder: Bent C. Jørgensen.

45/82 Er aldrig udkommet.

46/82 "EKSEMPLARISK UNDERVISNING OG FYSISK ERKENDESE-
1+11 ILLUSTRERET VED TO EKSEMPLER".
Projektrapport af: Torben O.Olsen, Lasse Rasmussen og Niels Dreyer Sørensen.
Vejleder: Bent C. Jørgensen.

47/82 "BARSEBÄCK OG DET VÆRST OFFICIELT-TÆNKELIGE UHELD".
ENERGY SERIES NO. 5.
Af: Bent Sørensen.

48/82 "EN UNDERSØGELSE AF MATEMATIKUNDERVISNINGEN PÅ ADGANGSKURSUS TIL KØBENHAVNS TEKNIKUM".
Projektrapport af: Lis Eilertzen, Jørgen Karrebæk, Troels Lange, Preben Nørregaard, Lissi Pedesen, Laust Rishøj, Lill Røn og Isac Showiki.
Vejleder: Mogens Niss.

49/82 "ANALYSE AF MULTISPEKTRALE SATELLITBILLEDER".
Projektrapport af: Preben Nørregaard.
Vejledere: Jørgen Larsen og Rasmus Ole Rasmussen.

50/82 "HERSLEV - MULIGHEDER FOR VEDVARENDE ENERGI I EN LANDSBY".
ENERGY SERIES NO. 6.
Rapport af: Bent Christensen, Bent Hove Jensen, Dennis B. Møller, Bjarne Laursen, Bjarne Lillethorup og Jacob Mørch Pedersen.
Vejleder: Bent Sørensen.

51/82 "HVAD KAN DER GØRES FOR AT AFHJÆLPE PIGERS BLOKERING OVERFOR MATEMATIK ?"
Projektrapport af: Lis Eilertzen, Lissi Pedersen, Lill Røn og Susanne Stender.

52/82 "DESUSPENSION OF SPLITTING ELLIPTIC SYMBOLS".
Af: Bernhelm Booss og Krzysztof Wojciechowski.

53/82 "THE CONSTITUTION OF SUBJECTS IN ENGINEERING EDUCATION".
Af: Arne Jacobsen og Stig Andur Pedersen.

54/82 "FUTURES RESEARCH" - A Philosophical Analysis of Its Subject-Matter and Methods.
Af: Stig Andur Pedersen og Johannes Witt-Hansen.

55/82 "MATEMATISKE MODELLER" - Litteratur på Roskilde Universitetsbibliotek.
En biografi.
Af: Else Høyrup.

Vedr. tekst nr. 55/82 se også tekst nr. 62/83.

56/82 "EN - TO - MANGE" -
En undersøgelse af matematisk økologi.
Projektrapport af: Troels Lange.
Vejleder: Anders Madsen.

57/83 "ASPECT EKSPERIMENTET"-
Skjulte variable i kvantemekanikken?
Projektrapport af: Tom Juul Andersen.
Vejleder: Peder Voetmann Christiansen.
Nr. 57 er udgået.

58/83 "MATEMATISKE VANDRINGER" - Modelbetragtninger over spredning af dyr mellem småbiotoper i agerlandet.
Projektrapport af: Per Hammershøj Jensen og Lene Vagn Rasmussen.
Vejleder: Jørgen Larsen.

59/83 "THE METHODOLOGY OF ENERGY PLANNING".
ENERGY SERIES NO. 7.
Af: Bent Sørensen.

60/83 "MATEMATISK MODEKSPERTISE"- et eksempel.
Projektrapport af: Erik O. Gade, Jørgen Karrebæk og Preben Nørregaard.
Vejleder: Anders Madsen.

61/83 "FYSIKS IDEOLOGISKE FUNKTION, SOM ET EKSEMPEL PÅ EN NATURVIDENSKAB - HISTORISK SET".
Projektrapport af: Annette Post Nielsen.
Vejledere: Jens Høyrup, Jens Højgaard Jensen og Jørgen Vogelius.

62/83 "MATEMATISKE MODELLER" - Litteratur på Roskilde Universitetsbibliotek.
En biografi 2. rev. udgave.
Af: Else Høyrup.

63/83 "GREATING ENERGY FUTURES:A SHORT GUIDE TO ENERGY PLANNING".
ENERGY SERIES No. 8.
Af: David Crossley og Bent Sørensen.

64/83 "VON MATEMATIK UND KRIEG".
Af: Berhelm Booss og Jens Høyrup.

65/83 "ANVENDT MATEMATIK - TEORI ELLER PRAKSIS".
Projektrapport af: Per Hedegård Andersen, Kirsten Habekost, Carsten Holst-Jensen, Annelise von Moos, Else Marie Pedersen og Erling Møller Pedersen.
Vejledere: Bernhelm Booss og Klaus Grünbaum.

66/83 "MATEMATISKE MODELLER FOR PERIODISK SELEKTION I ESCHERICHIA COLI".
Projektrapport af: Hanne Lisbet Andersen, Ole Richard Jensen og Klavs Frisdahl.
Vejledere: Jørgen Larsen og Anders Hede Madsen.

67/83 "ELEPSOIDE METODEN - EN NY METODE TIL LINEÆR PROGRAMMERING?"
Projektrapport af: Lone Biilmann og Lars Boye.
Vejleder: Mogens Brun Heefelt.

68/83 "STOKASTISKE MODELLER I POPULATIONSGENETIK"
- til kritikken af teoriladede modeller.
Projektrapport af: Lise Odgård Gade, Susanne Hansen, Michael Hviid og Frank Mølgård Olsen.
Vejleder: Jørgen Larsen.

69/83 "ELEVFORUDSÆTNINGER I FYSIK"
- en test i 1.g med kommentarer.
Af: Albert C. Paulsen.

70/83 "INDLÆRINGS - OG FORMIDLINGSPROBLEMER I MATEMATIK PÅ VOKSENUNDERVISNINGSNIVEAU".
Projektrapport af: Hanne Lisbet Andersen, Torben J. Andreasen, Svend Åge Houmann, Helle Glerup Jensen, Keld Fl. Nielsen, Lene Vagn Rasmussen.
Vejleder: Klaus Grünbaum og Anders Hede Madsen.

71/83 "PIGER OG FYSIK"
- et problem og en udfordring for skolen?
Af: Karin Beyer, Sussanne Blegaa, Birthe Olsen, Jette Reich og Mette Vedelsby.

72/83 "VERDEN IFVLGE PEIRCE" - to metafysiske essays, om og af C.S Peirce.
Af: Peder Voetmann Christiansen.

73/83 ""EN ENERGIANALYSE AF LANDBRUG"
- økologisk contra traditionelt.
ENERGY SERIES NO. 9
Specialeopgave i fysik af: Bent Hove Jensen.
Vejleder: Bent Sørensen.

74/84 "MINIATURISERING AF MIKROELEKTRONIK" - om videnskabeliggjort teknologi og nytten af at lære fysik.
Projektrapport af: Bodil Harder og Linda Szkotak Jensen.
Vejledere: Jens Højgaard Jensen og Bent C. Jørgensen.

75/84 "MATEMATIKUNDERVISNINGEN I FREMTIDENS GYMNASIUM"
- Case: Lineær programmering.
Projektrapport af: Morten Blomhøj, Klavs Frisdahl og Frank Mølgaard Olsen.
Vejledere: Mogens Brun Heefelt og Jens Bjørneboe.

76/84 "KERNEKRAFT I DANMARK?" - Et høringssvar indkaldt af miljøministeriet, med kritik af miljøstyrelsens rapporter af 15. marts 1984.
ENERGY SERIES No. 10
Af: Niels Boye Olsen og Bent Sørensen.

77/84 "POLITISKE INDEKS - FUP ELLER FAKTA?"
Opinionsundersøgelser belyst ved statistiske modeller.
Projektrapport af: Svend Åge Houmann, Keld Nielsen og Susanne Stender.
Vejledere: Jørgen Larsen og Jens Bjørneboe.

78/84 "JÆVNSTRØMSLEDNINGSEVNE OG GITTERSTRUKTUT I AMORFT GERMANIUM".
Specialrapport af: Hans Hedal, Frank C. Ludvigsen og Finn C. Physant.
Vejleder: Niels Boye Olsen.

79/84 "MATEMATIK OG ALMENDANNELSE".
Projektrapport af: Henrik Coster, Mikael Wennerberg Johansen, Povl Kattler, Birgitte Lydholm og Morten Overgaard Nielsen.
Vejleder: Bernhelm Booss.

80/84 "KURSUSMATERIALE TIL MATEMATIK B".
Af: Mogens Brun Heefelt.

81/84 "FREKVENSAFHÆNGIG LEDNINGSEVNE I AMORFT GERMANIUM".
Specialerapport af: Jørgen Wind Petersen og Jan Christensen.
Vejleder: Niels Boye Olsen.

82/84 "MATEMATIK - OG FYSIKUNDERVISNINGEN I DET AUTOMATISEREDE SAMFUND".
Rapport fra et seminar afholdt i Hvidovre 25-27 april 1983.
Red.: Jens Højgaard Jensen, Bent C. Jørgensen og Mogens Niss.

83/84 "ON THE QUANTIFICATION OF SECURITY":
PEACE RESEARCH SERIES NO. I
Af: Bent Sørensen
nr. 83 er p.t. udgået

84/84 "NOGLE ARTIKLER OM MATEMATIK, FYSIK OG ALMENDANNELSE".
Af: Jens Højgaard Jensen, Mogens Niss m. fl.

85/84 "CENTRIFUGALREGULATORER OG MATEMATIK".
Specialerapport af: Per Hedegård Andersen, Carsten Holst-Jensen, Else Marie Pedersen og Erling Møller Pedersen.
Vejleder: Stig Andur Pedersen.

86/84 "SECURITY IMPLICATIONS OF ALTERNATIVE DEFENSE OPTIONS FOR WESTERN EUROPE".
PEACE RESEARCH SERIES NO. 2
Af: Bent Sørensen.

87/84 "A SIMPLE MODEL OF AC HOPPING CONDUCTIVITY IN DISORDERED SOLIDS".
Af: Jeppe C. Dyre.

88/84 "RISE, FALL AND RESURRECTION OF INFINITESIMALS".
Af: Detlef Laugwitz.

89/84 "FJERNVARMEOPTIMERING".
Af: Bjarne Lillethorup og Jacob Mørch Pedersen.

90/84 "ENERGI I 1.G - EN TEORI FOR TILRETTELÆGGELSE".
Af: Albert Chr. Paulsen.

91/85 "KVANTETEORI FOR GYMNASIET".
1. Lærervejledning
Projektrapport af: Biger Lundgren, Henning Sten Hansen og John Johansson.
Vejleder: Torsten Meyer.

92/85 "KVANTETEORI FOR GYMNASIET".
2. Materiale
Projektrapport af: Biger Lundgren, Henning Sten Hansen og John Johansson.
Vejleder: Torsten Meyer.

93/85 "THE SEMIOTICS OF QUANTUM - NON - LOCALITY".
Af: Peder Voetmann Christiansen.

94/85 "TREENIGHEDEN BOURBAKI - generalen, matematikeren og ånden".
Projektrapport af: Morten Blomhøj, Klavs Frisdahl og Frank M. Olsen.
Vejleder: Mogens Niss.

95/85 "AN ALTERNATIV DEFENSE PLAN FOR WESTERN EUROPE".
PEACE RESEARCH SERIES NO. 3
Af: Bent Sørensen

96/85 "ASPEKTER VED KRAFTVARMEFORSYNING".
Af: Bjarne Lilletorup.
Vejleder: Bent Sørensen.

97/85 "ON THE PHYSICS OF A.C. HOPPING CONDUCTIVITY".
Af: Jeppe C. Dyre.

98/85 "VALGMULIGHEDER I INFORMATIONSALDEREN".
Af: Bent Sørensen.

99/85 "Der er langt fra Q til R".
Projektrapport af: Niels Jørgensen og Mikael Klintorp.
Vejleder: Stig Andur Pedersen.

100/85 "TALSYSTEMETS OPBYGNING".
Af: Mogens Niss.

101/85 "EXTENDED MOMENTUM THEORY FOR WINDMILLS IN PERTURBATIVE FORM".
Af: Ganesh Sengupta.

102/85 OPSTILLING OG ANALYSE AF MATEMATISKE MODELLER, BELYST VED MODELLER OVER KØERS FODEROPTAGELSE OG - OMSÆTNING".
Projektrapport af: Lis Eilertzen, Kirsten Habekost, Lill Røn og Susanne Stender.
Vejleder: Klaus Grünbaum.

103/85 "ØDSLE KOLDKRIGERE OG VIDENSKABENS LYSE IDEER".
Projektrapport af: Niels Ole Dam og Kurt Jensen.
Vejleder: Bent Sørensen.

104/85 "ANALOGREGNEMASKINEN OG LORENZLIGNINGER".
Af: Jens Jæger.

105/85 "THE FREQUENCY DEPENDENCE OF THE SPECIFIC HEAT AF THE GLASS REANSITION".
Af: Tage Christensen.

"A SIMPLE MODEL AF AC HOPPING CONDUCTIVITY".
Af: Jeppe C. Dyre.
Contributions to the Third International Conference on the Structure of Non - Crystalline Materials held in Grenoble July 1985.

106/85 "QUANTUM THEORY OF EXTENDED PARTICLES".
Af: Bent Sørensen.

107/85 "EN MYG GØR INGEN EPIDEMI".
- flodblindhed som eksempel på matematisk modellering af et epidemiologisk problem.
Projektrapport af: Per Hedegård Andersen, Lars Boye, CarstenHolst Jensen, Else Marie Pedersen og Erling Møller Pedersen.
Vejleder: Jesper Larsen.

108/85 "APPLICATIONS AND MODELLING IN THE MATEMATICS CURRICULUM" - state and trends -
Af: Mogens Niss.

109/85 "COX I STUDIETIDEN" - Cox's regressionsmodel anvendt på studenteroplysninger fra RUC.
Projektrapport af: Mikael Wennerberg Johansen, Poul Katler og Torben J. Andreasen.
Vejleder: Jørgen Larsen.

110/85 "PLANNING FOR SECURITY".
Af: Bent Sørensen

111/85 JORDEN RUNDT PÅ FLADE KORT".
Projektrapport af: Birgit Andresen, Beatriz Quinones og Jimmy Staal.
Vejleder: Mogens Niss.

112/85 "VIDENSKABELIGGØRELSE AF DANSK TEKNOLOGISK INNOVATION FREM TIL 1950 - BELYST VED EKSEMPLER".
Projektrapport af: Erik Odgaard Gade, Hans Hedal, Frank C. Ludvigsen, Annette Post Nielsen og Finn Physant.
Vejleder: Claus Bryld og Bent C. Jørgensen.

113/85 "DESUSPENSION OF SPLITTING ELLIPTIC SYMBOLS 11".
Af: Bernhelm Booss og Krzysztof Wojciechowski.

114/85 "ANVENDELSE AF GRAFISKE METODER TIL ANALYSE AF KONTIGENSTABELLER".
Projektrapport af: Lone Biilmann, Ole R. Jensen og Anne-Lise von Moos.
Vejleder: Jørgen Larsen.

115/85 "MATEMATIKKENS UDVIKLING OP TIL RENESSANCEN".
Af: Mogens Niss.

116/85 "A PHENOMENOLOGICAL MODEL FOR THE MEYER-NELDEL RULE".
Af: Jeppe C. Dyre.

117/85 "KRAFT & FJERNVARMEOPTIMERING"
Af: Jacob Mørch Pedersen.
Vejleder: Bent Sørensen

118/85 "TILFÆLDIGHEDEN OG NØDVENDIGHEDEN IFØLGE PEIRCE OG FYSIKKEN".
Af: Peder Voetmann Christiansen

119/86 "DET ER GANSKE VIST - - EUKLIDS FEMTE POSTULAT KUNNE NOK SKABE RØRE I ANDEDAMMEN".
Af: Iben Maj Christiansen
Vejleder: Mogens Niss.

120/86 "ET ANTAL STATISTISKE STANDARDMODELLER".
Af: Jørgen Larsen

121/86 "SIMULATION I KONTINUERT TID".
Af: Peder Voetmann Christiansen.

122/86 "ON THE MECHANISM OF GLASS IONIC CONDUCTIVITY".
Af: Jeppe C. Dyre.

123/86 "GYMNASIEFYSIKKEN OG DEN STORE VERDEN".
Fysiklærerforeningen, IMFUFA, RUC.

124/86 "OPGAVESAMLING I MATEMATIK".
Samtlige opgaver stillet i tiden 1974-jan. 1986.

125/86 "UVBY,β - systemet - en effektiv fotometrisk spektral-klassifikation af B-,A- og F-stjerner".
Projektrapport af: Birger Lundgren.

126/86 "OM UDVIKLINGEN AF DEN SPECIELLE RELATIVITETSTEORI".
Projektrapport af: Lise Odgaard & Linda Szkotak Jensen
Vejledere: Karin Beyer & Stig Andur Pedersen.

127/86 "GALOIS' BIDRAG TIL UDVIKLINGEN AF DEN ABSTRAKTE ALGEBRA".
Projektrapport af: Pernille Sand, Heine Larsen & Lars Frandsen.
Vejleder: Mogens Niss.

128/86 "SMÅKRYB" - om ikke-standard analyse.
Projektrapport af: Niels Jørgensen & Mikael Klintorp.
Vejleder: Jeppe Dyre.

129/86 "PHYSICS IN SOCIETY"
Lecture Notes 1983 (1986)
Af: Bent Sørensen

130/86 "Studies in Wind Power"
Af: Bent Sørensen

131/86 "FYSIK OG SAMFUND" - Et integreret fysik/historie-projekt om naturanskuelsens historiske udvikling og dens samfundsmæssige betingethed.
Projektrapport af: Jakob Heckscher, Søren Brønd, Andy Wierød.
Vejledere: Jens Høyrup, Jørgen Vogelius, Jens Højgaard Jensen.

132/86 "FYSIK OG DANNELSE"
Projektrapport af: Søren Brønd, Andy Wierød.
Vejledere: Karin Beyer, Jørgen Vogelius.

133/86 "CHERNOBYL ACCIDENT: ASSESSING THE DATA.
ENERGY SERIES NO. 15.
AF: Bent Sørensen.

134/87 "THE D.C. AND THE A.C. ELECTRICAL TRANSPORT IN AsSeTe SYSTEM"
Authors: M.B.El-Den, N.B.Olsen, Ib Høst Pedersen, Petr Visčor

135/87 "INTUITIONISTISK MATEMATIKS METODER OG ERKENDELSES-TEORETISKE FORUDSÆTNINGER"
MASTEMATIKSPECIALE: Claus Larsen
Vejledere: Anton Jensen og Stig Andur Pedersen

136/87 "Mystisk og naturlig filosofi: En skitse af kristendommens første og andet møde med græsk filosofi"
Projektrapport af Frank Colding Ludvigsen
Vejledere: Historie: Ib Thiersen
Fysik: Jens Højgaard Jensen

137/87 "HOPMODELLER FOR ELEKTRISK LEDNING I UORDNEDE FASTE STOFFER" - Resume af licentiatafhandling
Af: Jeppe Dyre
Vejledere: Niels Boye Olsen og Peder Voetmann Christiansen.

138/87 "JOSEPHSON EFFECT AND CIRCLE MAP."
Paper presented at The International Workshop on Teaching Nonlinear Phenomena at Universities and Schools, "Chaos in Education". Balaton, Hungary, 26 April-2 May 1987.

By: Peder Voetmann Christiansen

139/87 "Machbarkeit nichtbeherrschbarer Technik durch Fortschritte in der Erkennbarkeit der Natur"

Af: Bernhelm Booss-Bavnbek
Martin Bohle-Carbonell

140/87 "ON THE TOPOLOGY OF SPACES OF HOLOMORPHIC MAPS"

By: Jens Gravesen

141/87 "RADIOMETERS UDVIKLING AF BLODGASAPPARATUR - ET TEKNOLOGIHISTORISK PROJEKT"

Projektrapport af Finn C. Physant

Vejleder: Ib Thiersen

142/87 "The Calderón Projektor for Operators With Splitting Elliptic Symbols"

by: Bernhelm Booss-Bavnbek og
Krzysztof P. Wojciechowski

143/87 "Kursusmateriale til Matematik på NAT-BAS"

af: Mogens Brun Heefelt

144/87 "Context and Non-Locality - A Peircan Approach
Paper presented at the Symposium on the Foundations of Modern Physics The Copenhagen Interpretation 60 Years after the Como Lecture. Joensuu, Finland, 6 - 8 august 1987.

By: Peder Voetmann Christiansen

145/87 "AIMS AND SCOPE OF APPLICATIONS AND MODELLING IN MATHEMATICS CURRICULA"
Manuscript of a plenary lecture delivered at ICMTA 3, Kassel, FRG 8.-11.9.1987

By: Mogens Niss

146/87 "BESTEMMELSE AF BULKRESISTIVITETEN I SILICIUM"
- en ny frekvensbaseret målemetode.

Fysikspeciale af Jan Vedde

Vejledere: Niels Boye Olsen & Petr Viščor

147/87 "Rapport om BIS på NAT-BAS"

redigeret af: Mogens Brun Heefelt

148/87 "Naturvidenskabsundervisning med Samfundsperspektiv"

af: Peter Colding-Jørgensen DLH
Albert Chr. Paulsen

149/87 "In-Situ Measurements of the density of amorphous germanium prepared in ultra high vacuum"

by: Petr Viščor

150/87 "Structure and the Existence of the first sharp diffraction peak in amorphous germanium prepared in UHV and measured in-situ"

by: Petr Viščor

151/87 "DYNAMISK PROGRAMMERING"

Matematikprojekt af:
Birgit Andresen, Keld Nielsen og Jimmy Staal

Vejleder: Mogens Niss

152/87 "PSEUDO-DIFFERENTIAL PROJECTIONS AND THE TOPOLOGY OF CERTAIN SPACES OF ELLIPTIC BOUNDARY VALUE PROBLEMS"

by: Bernhelm Booss-Bavnbek
Krzysztof P. Wojciechowski

153/88 "HALVLEDERTEKNOLOGIENS UDVIKLING MELLEM MILITÆRE OG CIVILE KRÆFTER"

Et eksempel på humanistisk teknologihistorie
Historiespeciale

Af: Hans Hedal

Vejleder: Ib Thiersen

154/88 "MASTER EQUATION APPROACH TO VISCOUS LIQUIDS AND THE GLASS TRANSITION"

By: Jeppe Dyre

155/88 "A NOTE ON THE ACTION OF THE POISSON SOLUTION OPERATOR TO THE DIRICHLET PROBLEM FOR A FORMALLY SELFADJOINT DIFFERENTIAL OPERATOR"

by: Michael Pedersen

156/88 "THE RANDOM FREE ENERGY BARRIER MODEL FOR AC CONDUCTION IN DISORDERED SOLIDS"

by: Jeppe C. Dyre

157/88 " STABILIZATION OF PARTIAL DIFFERENTIAL EQUATIONS BY FINITE DIMENSIONAL BOUNDARY FEEDBACK CONTROL: A pseudo-differential approach."

by: Michael Pedersen

158/88 "UNIFIED FORMALISM FOR EXCESS CURRENT NOISE IN RANDOM WALK MODELS"

by: Jeppe Dyre

159/88 "STUDIES IN SOLAR ENERGY"

by: Bent Sørensen

160/88 "LOOP GROUPS AND INSTANTONS IN DIMENSION TWO"

by: Jens Gravesen

161/88 "PSEUDO-DIFFERENTIAL PERTURBATIONS AND STABILIZATION OF DISTRIBUTED PARAMETER SYSTEMS:
Dirichlet feedback control problems"

by: Michael Pedersen

162/88 "PIGER & FYSIK - OG MEGET MERE"

AF: Karin Beyer, Sussanne Blegaa, Birthe Olsen, Jette Reich , Mette Vedelsby

163/88 "EN MATEMATISK MODEL TIL BESTEMMELSE AF PERMEABILITETEN FOR BLOD-NETHINDE-BARRIEREN"

Af: Finn Langberg, Michael Jarden, Lars Frellesen

Vejleder: Jesper Larsen

164/88 "Vurdering af matematisk teknologi
Technology Assessment
Technikfolgenabschätzung"

Af: Bernhelm Booss-Bavnbek, Glen Pate med
Martin Bohle-Carbonell og Jens Højgaard Jensen

165/88 "COMPLEX STRUCTURES IN THE NASH-MOSER CATEGORY"

by: Jens Gravesen

166/88 "Grundbegreber i Sandsynligheds-regningen"

Af: Jørgen Larsen

167a/88 "BASISSTATISTIK 1. Diskrete modeller"

Af: Jørgen Larsen

167b/88 "BASISSTATISTIK 2. Kontinuerte modeller"

Af: Jørgen Larsen

168/88 "OVERFLADEN AF PLANETEN MARS"
Laboratorie-simulering og MARS-analoger undersøgt ved Mössbauerspektroskopi.

Fysikspeciale af:

Birger Lundgren

Vejleder: Jens Martin Knudsen
Fys.Lab./HCØ

169/88 "CHARLES S. PEIRCE: MURSTEN OG MØRTEL TIL EN METAFYSIK."
Fem artikler fra tidsskriftet "The Monist" 1891-93.

Introduktion og oversættelse:

Peder Voetmann Christeansen

170/88 "OPGAVESAMLING I MATEMATIK"
Samtlige opgaver stillet i tiden 1974 - juni 1988

171/88 "The Dirac Equation with Light-Cone Data"
af: Johnny Tom Ottesen

172/88 "FYSIK OG VIRKELIGHED"
Kvantemekanikkens grundlagsproblem i gymnasiet.

Fysikprojekt af:

Erik Lund og Kurt Jensen

Vejledere: Albert Chr. Paulsen og Peder Voetmann Christiansen

---

173/89 "NUMERISKE ALGORITMER"
af: Mogens Brun Heefelt

174/89 " GRAFISK FREMSTILLING AF FRAKTALER OG KAOS"

af: Peder Voetmann Christiansen

175/89 " AN ELEMENTARY ANALYSIS OF THE TIME DEPENDENT SPECTRUM OF THE NON-STATONARY SOLUTION TO THE OPERATOR RICCATI EQUATION

af: Michael Pedersen

176/89 " A MAXIUM ENTROPY ANSATZ FOR NONLINEAR RESPONSE THEORY"

af : Jeppe Dyre

177/89 "HVAD SKAL ADAM STÅ MODEL TIL"
af: Morten Andersen, Ulla Engström, Thomas Gravesen, Nanna Lund, Pia Madsen, Dina Rawat, Peter Torstensen

Vejleder: Mogens Brun Heefelt

178/89 "BIOSYNTESEN AF PENICILLIN - en matematisk model"
af: Ulla Eghave Rasmussen, Hans Oxvang Mortensen, Michael Jarden

vejleder i matematik: Jesper Larsen
biologi: Erling Lauridsen

179a/89 "LÆRERVEJLEDNING M.M. til et eksperimentelt forløb om kaos"

af: Andy Wierød, Søren Brønd og Jimmy Staal

Vejledere: Peder Voetmann Christiansen
Karin Beyer

179b/89 "ELEVHÆFTE: Noter til et eksperimentelt kursus om kaos"

af: Andy Wierød, Søren Brønd og Jimmy Staal

Vejledere: Peder Voetmann Christiansen
Karin Beyer

180/89 "KAOS I FYSISKE SYSTEMER eksemplificeret ved torsions- og dobbeltpendul".
af: Andy Wierød, Søren Brønd og Jimmy Staal

181/89 "A ZERO-PARAMETER CONSTITUTIVE RELATION FOR PURE SHEAR VISCOELASTICITY"
by: Jeppe Dyre

183/89 "MATEMATICAL PROBLEM SOLVING, MODELLING. APPLICATIONS AND LINKS TO OTHER SUBJECTS - State. trends and issues in mathematics instruction

by: WERNER BLUM, Kassel (FRG) og
MOGENS NISS, Roskilde (Denmark)

184/89 "En metode til bestemmelse af den frekvensafhængige varmefylde af en underafkølet væske ved glasovergangen"

af: Tage Emil Christensen

185/90 "EN NÆSTEN PERIODISK HISTORIE"
Et matematisk projekt

af: Steen Grode og Thomas Jessen

Vejleder: Jacob Jacobsen

186/90 "RITUAL OG RATIONALITET i videnskabers udvikling"
redigeret af Arne Jakobsen og Stig Andur Pedersen

187/90 "RSA - et kryptisk system"
af: Annemette Sofie Olufsen, Lars Frellesen og Ole Møller Nielsen

Vejledere: Michael Pedersen og Finn Munk

188/90 "FERMICONDENSATION - AN ALMOST IDEAL GLASS TRANSITION"
by: Jeppe Dyre

189/90 "DATAMATER I MATEMATIKUNDERVISNINGEN PÅ GYMNASIET OG HØJERE LÆREANSTALTER

af: Finn Langberg

190/90 "FIVE REQUIREMENTS FOR AN APPROXIMATE NONLINEAR RESPONSE THEORY"

by: Jeppe Dyre

191/90 "MOORE COHOMOLOGY, PRINCIPAL BUNDLES AND ACTIONS OF GROUPS ON C*-ALGEBRAS"

by: Iain Raeburn and Dana P. Williams

192/90 "Age-dependent host mortality in the dynamics of endemic infectious diseases and
SIR-models of the epidemiology and natural selection of co-circulating influenza virus with partial cross-immunity"

by: Viggo Andreasen

193/90 "Causal and Diagnostic Reasoning"

by: Stig Andur Pedersen

195/90 "STADIER PÅ PARADIGMETS VEJ"
Et projekt om den videnskabelige udvikling der førte til dannelsen af kvantemekanikken.

Projektrapport for 1. modul på fysikuddannelsen, skrevet af:

Anja Boisen, Thomas Hougård, Anders Gorm Larsen, Nicolai Ryge.

Vejleder: Peder Voetmann Christiansen

196/90 "ER KAOS NØDVENDIGT?"
- en projektrapport om kaos' paradigmatiske status i fysikken.

af: Johannes K. Nielsen, Jimmy Staal og Peter Bøggild

Vejleder: Peder Voetmann Christiansen