Konstruktiv matematik

Konstruktiv intuitionistisk fuldstændighed

D602F12, foråret 2013, Aalborg Universitet

27. juni 2013

AALBORG UNIVERSITET
STUDENTERRAPPORT

Institut for Datalogi
Selma Lagerlöfs Vej 300
9220 Aalborg Ø
Telefon (+45) 9940 9940
Fax (+45) 9940 9798
http://www.cs.aau.dk

Titel:
Konstruktiv matematik (Constructive mathematics)

Tema:
Konstruktiv intuitionistisk fuldstændighed (Constructive intuitionistic completeness)

Projektperiode:
Foråret 2013

Projektgruppe:
Gruppe d602f13

Gruppemedlemmer:
Daniel Hillerström
Mathias Ruggaard Pedersen

Vejleder:
Hans Hüttel

Oplagstal: 5

Rapport sideantal: 63

Appendiks sideantal: 8

Total sideantal: 73

Synopsis:

Denne rapport dokumenterer vores arbejde med konstruktiv matematik, og hvordan konstruktiv matematik med fordel kan udnyttes i datalogi. Målet med vores arbejde var at give et konstruktivt bevis for fuldstændighed af intuitionistisk udsagnslogik.
I rapportens første del beskriver vi den kontekst, konstruktivisme udsprang af, og nogle af uenighederne mellem konstruktivister og ikke-konstruktivister. Vi formulerer desuden de problemstillinger, vi vil udforske i resten af rapporten.
Herefter udarbejder vi den grundlæggende teori for intuitionistisk logik og viser nogle forskelle på klassisk og intuitionistisk logik. Baseret på dette giver vi et klassisk bevis for fuldstændighed.
Som kontrast til det klassiske bevis, giver vi derefter et konstruktivt bevis, og beskriver vores implementering af samme bevis. Afslutningsvis diskuterer vi, hvilke fordele en konstruktiv tilgang til matematik kan give, særligt fra en datalogisk synsvinkel.

Resume in English

In accordance with the study regulations we include here a resume of this project written in English.

Our initial interest spawned from the validity of the principle of the excluded middle (LEM). In classic mathematical logic LEM is a perfectly acceptable method for proving the existence of an object. We try to describe the nature of LEM and how it works in section 1.1. However, it caught our interest when we learned that there exists an other philosophical school of mathematics, namely the constructive mathematics, which rejects the universal validity of LEM. We discovered that the constructive mathematics is based upon the intuitionistic logic which happens to be based upon Brouwer's intuitionism. Therefore we briefly investigate both Brouwer's intuitionism and the intuitionistic logic in section 1.2 and 1.4.1. The intuitionistic logic is a formalised version of Brouwer's intuitionism and offers therefore a systematic interpretation of the logical connectives. This particular interpretation is known as the Brouwer-Heyting-Kolmogorov (BHK) interpretation, named after its contributors. The BHK-interpretation and the classical interpretation are seperated by their respective views on the existence of a mathematical object. For instance, the BHK-interpretation requires an existential proof to be a direct proof. The constructive mathematics inherits this arguably stricter proof requirement. Furthermore the constructive mathematics extends the notion of existence by requiring every existential proof to give an algorithm which, given the correct input, constructs the object in question. The notion of construction is therefore essential in constructive mathematics, hence the name.

During our research we discovered that some literature on intuitionistic logic gives classical proofs for some intuitionistic results. In particular, we noted that the Completeness theorem of intuitionistic logic in [vD08] is proved by contradiction. Given the strict proof requirements of the constructive mathematics and the fact that it is based upon the intuitionistic logic we found this very strange. This led us to our problem statement: "*How can we prove the Completeness theorem of intuitionistic logic using only constructive arguments?*"

Before we answered the problem statement we studied inductively defined sets in chapter 2. This is the theoretical foundation of much of the work conducted in the succeeding chapters. For example, the principle of induction, one of the main results of that chapter, is applied extensively when we prove results about propositional logic.

In chapter 3 we explore propositional logic for both classical logic and intuitionistic logic. First we present a grammar for the propositional language which serves as a framework for the rest of the work. Within this framework we define and describe derivation rules for both intuitionistic logic and classical logic. On the basis of this work we show some differences between the two systems, most notably that the classical equivalance between a proposition and its double negation is not valid in intuitionistic logic. However, we show that a proposition implies its double negation but not the other way around. Furthermore, we describe the semantics of intuitionistic logic using Kripke semantics. Finally we show the soundness and completeness of intuitionitic logic. The soundness is proved constructively using a proof by induction while the completeness result is proved classically in the same way as in [vD08].

In chapter 4 we set out to give a constructive proof of the completeness theorem for intuitionistic logic which is valid under the BHK-interpretation. Before attempting to prove the result we define the notion of a sequent, system, and tableau. Then the proof is divided into three parts

(i) Given a valid sequent we can construct a successful tableau.

(ii) Given a successful tableau we can give a proof for the sequent in the root of the tableau.

(iii) Given a tableau with an open leaf we can construct a counter Kripke-model.

The three results each give a precise algorithm which explains how to create the objects in question. We conclude that while the constructive proof is much longer than its classical counterpart, it is substantial in terms of computational information. The proof actually devises an algorithm ready for implementation on a computer. We describe a concrete implementation of some of the parts of the proof in chapter 5. We briefly discuss a possible implementation using the functional programming language Haskell. Doing so we discover how the definitions and proofs together form some kind of abstract program description and all that is left for the programmer is to translate these models and algorithms into a programming language of their choice.

Forord

I 2010 startede vi på datalogistudiet ved Aalborg Universitet. I løbet af det første studieår blev vi introduceret til en lang række grundlæggende koncepter i datalogien. Bl. a. blev vi introduceret for diskret matematik, hvis teori i særdeleshed finder sin anvendelse indenfor datalogien. Vi blev hurtigt begejstrede for emnet, hvilket affødte en stigende interesse for matematikken i datalogien. Derfor bestemte vi os for at supplere vores hovedfag med et bifag, nemlig matematik.

I foråret 2012 begyndte vi så at studere matematik sideløbende med datalogistudiet. I efteråret 2012, samt foråret 2013 har vi udelukkende beskæftiget os med matematik, herunder bl. a. kurser i abstrakt algebra og matematisk analyse i metriske rum. Sideløbende med matematikstudiet, har vi skullet arbejde på dette bachelorprojekt i datalogi og matematik. Da vi har en interesse for både matematik og datalogi, har vi længe ønsket at lave et projekt, der ligger i spændingsfeltet mellem matematik og datalogi. Denne rapport er et produkt af det projekt.

En særlig tak

Vi vil gerne give en særlig tak til vores vejleder Hans Hüttel. Først og fremmest for at takke ja til at vejlede os i dette projekt. Han har igennem hele dette projekt givet os hurtigt og grundigt hjælp og kritik – både i tide og utide. Desuden vil vi gerne takke ham for at efterkomme vores ønske om at lave et tværfagligt projekt.

Det online projekttitelblad

Denne rapport er en dokumentation af projektet, men indeholder ikke alt vi har lavet i dette projekt. Desuden ender projektet ikke med denne rapport. Derfor har vi lavet en hjemmeside, hvor alt væsentlig materiale i forbindelse med projektet kan findes. Hjemmesiden findes på `http://www.dhil.net/public/edu/aau/d602f13/`. På siden kan den komplette kildekode til vores programmel, samt en vejledning downloades.

---------------------------------- ----------------------------------
Daniel Hillerström Mathias Ruggaard Pedersen

Indhold

KAPITEL 1

Problemanalyse

"Reductio ad absurdum, which Euclid loved so much, is one of a mathematician's finest weapons."
— Godfrey H. Hardy (1877-1947) [Har40]

Bevis ved modstrid er en velkendt og ikke mindst velanvendt bevisteknik af matematikere. Eksempelvis er en stor del af resultaterne i [Wad10] (omhandler reel klassisk matematisk analyse) vist vha. modstridsargumenter. Vores undren tager udgangspunkt i netop gyldigheden af modstridsbeviser. Ideen om bevis ved modstrid er indbygget i den klassiske logik [Ros07], og er kendt som "det udelukkede tredje princip", hvilket implikerer at alle åbne matematiske problemer kan løses [Tro98].

I de følgende afsnit vil vi redegøre for det udelukkede tredje princip, samt undersøge alternativer til den klassiske matematik.

1.1 Det udelukkede tredje princip

Det udelukkede tredje princip (UTP) er et helt centralt princip i den klassiske logik. Bl. a. kan man se i [Wad10], at mange resultater i den klassiske reelle matematiske analyse benytter modstridsargumenter. Den klassiske logik indbyder også til dette. Vi kan sige, at den klassiske logik er en "binær logik". Med betegnelsen "binær logik" mener vi, at ethvert udsagn har to mulige konklusioner: sand eller falsk. UTP er netop denne antagelse:

For ethvert udsagn P, gælder enten P eller $\neg P$.

En direkte konsekvens af denne antagelse er, hvis vi ønsker at vise, at P er sand, så er det tilstrækkeligt at vise at det at antage $\neg P$ vil medføre en absurditet. Et sådan bevis kaldes et *modstridsbevis*. Strategien i et bevis ved modstrid er, at antage $\neg P$ og derefter vise, at det medfører en absurditet. Antag eksempelvis at vi har to udsagn P og Q, samt at vi ønsker at vise, at udsagnet P er sandt. Vi antager at $\neg P$ er sandt, og hvis vi ud fra den antagelse kan vise at $\neg P \to (Q \wedge \neg Q)$ er sandt, så får vi at $\neg P$ er falsk, hvilket klart er i modstrid med vores antagelse. Så må P være sand (bemærk at $(Q \wedge \neg Q)$ altid er falsk) [Ros07]. Betragt følgende eksempel fra talteori.

Proposition 1.1. *Antag at $a \in \mathbb{Z}$. Hvis a^2 er et lige tal, så er a et lige tal.*

Bevis. Vi viser dette ved modstrid. Antag at a^2 er lige (Q), og at a er ulige $(\neg P)$. Da a er et ulige tal findes et $k \in \mathbb{Z}$ således at $a = 2k + 1$. Dette medfører at

$$a^2 = (2k+1)^2 = 4k^2 + 4k + 1 = 2\underbrace{(2k^2 + 2k)}_{\text{element i } \mathbb{Z}} + 1,$$

altså a^2 er et ulige tal, men ifølge vores antagelse er a^2 et lige tal. Vi har at a^2 er et lige og ulige tal på samme tid $(Q \wedge \neg Q)$, en modstrid. $\square$

Bemærk, at vi "kun" har vist at $\neg P$ medfører en absurditet, men ifølge UTP er det tilstrækkeligt til at konkludere P er sand. I overstående eksempel, kan vi da også intuitivt overbevise os selv om, at hvis tallet ikke er ulige, så må det være lige. Vi har kun de to tilfælde.

Dette er dog ikke altid så intuitivt, men mange klassiske eksistensresultater er alligevel bevist ved modstrid [Pal97]. Lad os betragte et simpelt eksempel, som illustrerer dette.

Proposition 1.2 (Eksempel fra [Pal97])**.** *Enhver naturlig talfølge $\{a_n\}_{n \in \mathbb{N}}$ har et mindste element, dvs. der findes et $k \in \mathbb{N}$ således at $a_n > a_k$ for alle $n \neq k$.*

Bevis. Vi beviser dette ved modstrid. Antag at følgen $\{a_n\}$ ikke har noget mindste element, hvilket betyder at der for ethvert n findes $k > n$, således at $a_n > a_k$. Det medfører, at der findes en strengt aftagende delfølge $\{a_{n_k}\}_{k \in \mathbb{N}}$. Det må nødvendigvis gælde at $|a_{n_k} - a_{n_{k+1}}| \geq 1$, da delfølgen er strengt aftagende. Dette medfører, at vi for et tilpas stort $k \in \mathbb{N}$ får $a_{n_k} < 0$. En modstrid, fordi et sådan element *naturligvis* ikke kan være et naturligt tal. $\square$

1.2 Den matematiske krise i det 20. århundrede

Omkring begyndelse af det 20. århundrede oplevede matematikkens grundlag at blive godt og grundigt rystet. I denne tidsperiode fandt man flere paradokser, som satte spørgsmålstegn ved antagelsen om at matematikken havde et konsistent grundlag, der kunne formaliseres i matematikken selv. Fælles for paradokserne er at man opdagede dem indenfor mængdelæren, som er en af grundstenene i matematikken. Paradokserne viste at der i den daværende mængdelære eksisterede flere uoverensstemmelser. Et af matematikkens mest kendte paradokser blev opdaget af Bertrand Russell i 1902 [OR13].

Paradoks 1.3 (Russells paradoks [Ros07])**.** Betragt følgende mængde

$$S = \{X \mid X \notin X\}.$$

Er S så et element i S?

Der er to mulige svar på spørgsmålet, enten er S i S eller også er det ikke. Hvis $S \in S$, så skal S opfylde betingelsen $S \notin S$ ifølge definitionen af S, heraf ved vi at $S \in S$ ikke holder. Den eneste mulighed er så, at $S \notin S$ holder, men så opfylder S betingelsen for at være et element i S, så $S \notin S$ kan heller ikke være rigtigt. Uanset hvad, så kan vi ikke finde noget entydig svar på spørgsmålet, da begge tilfælde leder til en modstrid.

Kronecker havde flere år forinden kategorisk afvist lignende paradokser fundet i mængdelæren af Cantor [OR13]. Kronecker var selverklæret finitist, hvilket vil sige at han kun accepterede matematiske objekter der kan konstrueres ved brug af et endeligt antal trin. Han troede derfor ikke på eksistensen af en mængde á la den i Russells paradoks. En konsekvens af dette syn på eksistens, var at han heller ikke troede på eksistensen af de irrationelle tal [OR13].

Mange matematikere i Russells tid tog dog paradokset seriøst, og nogle mente endda at det underminerede matematikken [OR13]. Russell forsøgte selv, sammen med Whitehead, at gøre "skaden god igen" i deres fælles værk "*Principia Mathematica*" (1910), men det lykkedes ham ikke at lave en formel teori, som løste paradokset tilfredsstillende [OR13]. Dog fik de som resultat af deres arbejde grundlagt den moderne matematiske logik. Der opstod på baggrund af spørgsmålene om matematikkens grundlag flere skoler, baseret på forskellige matematikfilosofier, heriblandt formalismen og intuitionismen.

1.2.1 Hilberts formalisme

Hilbert interesserede sig for matematikkens grundlag og blev inspireret af Russells værk. Modsat Kronecker mente Hilbert at studiet af det uendelige er en væsentlig del af matematikken. Han mente dog, at de vanskeligheder, som man havde stødt på mht. matematikkens grundlag, muligvis skyldtes det uendelige [Jø11]. Hilberts pointe var, at man endnu ikke havde forstået det uendelige, samt hvilken rolle det spiller i matematikken.

Hilberts formalisme er et forsøg på at skabe et grundlag for matematik uden indbyggede modsigelser. Formalismen bygger på ideen om, at matematik i sig selv ikke har nogen sandhed, men i stedet har den sandhed, som vi tillægger den. Der er to centrale bestanddele i formalismen [Hor12]:

(i) Et formelt sprog,

(ii) og et aksiomatisk system.

Det naturlige sprog er åbent over for tvetydigheder, derfor skal vi ifølge Hilbert benytte et formelt sprog, som er givet ved en syntaks og en semantik [Hor12]. Syntaksen definerer opbygningsreglerne for de gyldige sproglige konstruktioner, mens semantikken bestemmer hvorledes konstruktionerne skal fortolkes. Et aksiomatisk system definerer hvordan vi ud fra sætninger, vi antager som gyldige (aksiomer), kan udlede nye sætninger. Tilsammen danner et formelt sprog og et aksiomatisk system et formelt system, hvori sandheden af et udsagn afhænger af den valgte semantik.

Hilbert ønskede med formalismen at give et aksiomatisk system for matematikken og eftervise, at dette var konsistent. Han troede, at når først vi havde valgt et formelt sprog, så vil vi kunne bevise at det aksiomatiske system er konsistent udelukkende ved brug af aksiomerne i systemet og det formelle sprog. Dog beviste Gödel, til Hilberts ærgelse, at et aksiomatisk system aldrig selv kan bevise at det er konsistent [Gö36]. Men Hilberts arbejde lagde grundlaget for det, vi i dag kalder en formaliseret logik.

1.2.2 Brouwers intuitionisme

L. E. J. Brouwer gav med sin intuitionisme et andet syn på matematikken, der modsat Hilberts syn ligger tættere opad Kroneckers finititiske syn. Brouwer forsøger ikke at "redde" matematikken, som Hilbert gjorde det. Han forsøger derimod at opstille et nyt grundlag for matematikken.

Modsat Hilbert, kunne Brouwer ikke se brugbarheden i en formaliseret logik. Han reducerede Hilberts formalisme til et "ligegyldigt spil med symboler" [Bro12]. Brouwer anså matematik som en idé der er skabt af mennesker og derfor eksisterer matematikken udelukkende i mennesket. For Brouwer er matematiske objekter mentale konstruktioner, der eksisterer i menneskets sind. Disse mentale konstruktioner udgør hjertet i matematikken [Tro98]. Eksistensbegrebet er helt centralt for Brouwer. Ifølge ham skal man for at bevise eksistensen af et matematisk objekt give en metode til at konstruere dette objekt. En sådan metode kaldes en *konstruktion*. Dette krav er klart strengere end det, man kender fra den klassiske logik.

Brouwers afhandling "*On the Foundations of Mathematics*" (1907) [Bro07] var hans første rene intuitionistiske værk, som udkom efter man havde fundet paradokser i mængdelæren, men Brouwer havde på daværende tidspunkt endnu ikke indset konsekvenserne af sit syn på den klassiske logik [Tro98]. Året efter sin afhandling udgav Brouwer artiklen "*On the unreliability of the logical principles*" (1908) [Bro08], hvori han havde indset konsekvenserne. Han viste, at man fra et intuitionistisk synspunkt ikke kan antage om et matematisk udsagn er sandt eller falsk. Vi kan kun hævde "P eller $\neg P$", når vi enten har et bevis for P, eller et bevis for at alle forsøg på at konstruere et bevis for P må føre til en absurditet. Dette betyder at UTP er problematisk set fra en intuitionistisk synsvinkel.

1.3 Konstruktiv matematik

Konstruktiv matematik er en særlig gren af matematikken, der bygger på Brouwers intutionisme. Den konstruktive matematik og klassiske matematik adskilles af UTP, da konstruktivister ikke accepterer UTP som værende universelt gyldigt. I bund og grund bygger konstruktiv matematik på ideen om, at matematik er konstruktivt, dvs. vi skal kunne konstruere de objekter, vi arbejder med i matematikken, hvilket også falder i god tråd med Brouwers opfattelse af matematikken. Desuden tog Brouwer, med sin intuitionisme, de første spadestik til den konstruktive matematik.

1.3.1 Det filosofiske fundament for konstruktivismen

En konsekvens af kravet om konstruktive eksistensbeviser er, at UTP ikke er universelt gyldigt i en konstruktivistisk forstand. For at se dette, lad os da se på de to fundamentale betingelser i konstruktivismen [MJ12]:

(i) Sandhed er ikke et primitiv.

(ii) Eksistens forstås konstruktivt.

Den første betingelse betyder, at "matematisk sandhed" sidestilles med at have lavet en konstruktion, der viser resultatet. Den anden betingelse siger, hvor stærkt eksistensbegrebet forstås indenfor konstruktivismen: når det er bevist, at et objekt eksisterer, så giver beviset en algoritme til at finde objektet. En konstruktivist kan derfor ikke acceptere et resultat, der er bevist ved brug af UTP. Fordi UTP medfører at vi kan bevise en eksistenspåstand P ved et modstridsbevis, dvs. give et ikke-konstruktivt bevis, er UTP problematisk for en konstruktivist [MJ12]. Vi ser altså, at UTP ikke er vellidt i den konstruktive matematik ligesom den heller ikke er det i den intuitionistiske logik, hvilket desuden er meget naturligt eftersom den konstruktive matematik baserer sig på den intuitionistiske logik. Vi beskrev tidligere, at Brouwer allerede indså dette i sit tidlige arbejde med intuitionismen. Lad os nu give et eksempel på hvad dette synspunkt medfører. Betragt følgende velkendte eksempel fra talteorien:

Formodning 1.4 (Goldbachs formodning [Ros07])**.** Ethvert lige heltal større end 2 kan udtrykkes som summen af to primtal.

Goldbachs formodning er, som navnet indikerer, blot en formodning og derfor endnu ikke bevist. Vi kan sige, at Goldbachs formodning er heuristisk velbegrundet; man har bl. a. konstruktivt bevist at formodningen holder for tallet $2 \cdot 10^{17}$ [OeS12, Ros07]. Der er dog en stor forskel på at have et heuristisk grundlag og så have et bevis. Hverken tilhængere af konstruktivisme eller klassisk logik accepterer et heuristisk argument som et bevis. Konstruktivismen og den klassiske logik adskiller sig netop i spørgsmålet om eksistensen af et bevis. I den klassiske logik følger det af UTP at der må eksistere et bevis eller et modbevis. Den konstruktive matematik tillader en anden mulighed. For ifølge konstruktivismens første betingelse er det fejlagtigt, at hævde at der findes et bevis for et resultat uden at have lavet et bevis, ligesom det er fejlagtigt at hævde, at der findes et modeksempel uden at have lavet et. Det giver en klar konflikt med UTP. Det åbner for den mulighed, at nogle matematiske problemer kan være uafgørbare.

Brouwer satte spørgsmålstegn ved UTPs universelle gyldighed. Han mente at antagelsen af UTP er ækvivalent med at påstå, at alle matematiske problemer har en løsning [Tro98, MJ12]. Vi har altså et filosofisk grundlag for at afvise UTP.

1.3.2 Den konstruktive renæssance

Konstruktiv matematik har oplevet en renæssance. Grundlaget og æren for denne renæssance kan i særdeleshed tilskrives Errett Bishops værk "*Foundations of constructive analysis*" (1967) [Bis67, MRR88]. Med ordene

> "It is no exaggeration to say that a straightforward realistic approach to mathematics has yet to be tried. It is time to make the attempt." [Bis67, MRR88]

satte Bishop sig for at forandre konstruktiv matematik fra at være en lille gren af matematikken, som kun logikere og filosoffer fandt interessant, til at være matematikerens fortrukne metode. Bishop har vist, at det er muligt, at konstruktivt udvikle en stor del af den matematik, som vi kender fra den klassiske analyse. Det er bemærkelsesværdigt, at Bishop gjorde en dyd ud af at tilføre den eksisterende matematiske notation og terminologi mere mening, i stedet for at udarbejde en ny speciel notation [MRR88]. Særligt gælder det at alle Bishops konstruktive resultater også er gyldige i den klassiske matematik [Bri13].

Konstruktiv matematik, med sin algoritmiske tilgang til resultater, er særligt interessant i teoretisk datalogi. Udviklingen og væksten af kraftfulde computere er med til at højne interessen for konstruktiv matematik. Computere har skabt større bevidsthed om effektive algoritmer og beregning generelt, og har desuden været med til at sætte fokus på faktisk implementering af konstruktiv matematik [MRR88]. Det første "større" resultat, der er blevet implementeret er firefarvesætningen [OR13].

Historien om firefarvesætningen

Firefarvesætningen er kendt som det første resultat, der er vist med computerstøtte.

Proposition 1.5 (Firefarvesætningen)**.** *Enhver ikke-orienteret planar graf kan farvelægges ved brug af højst fire farver.*

Firefarvesætningen blev først fremsat af Francis Guthrie, en af De Morgans tidligere studerende ved University College London. Guthrie formulerede firefarvesætningen, som et kortfarvelægningsproblem. Han spurgte De Morgan om det var rigtigt, at ethvert kort kan farvelægges ved brug af højst fire farver, således at der ikke findes to naboregioner med samme farve.

De Morgan blev interesseret i problemet, og han begyndte at spørge kollegaer om de kunne finde en løsningen på problemet. Han fik bl. a. spurgt Arthur Cayley, som senere udgav artiklen "*On the colouring of maps*" (1879) der handlede om problemerne i forbindelse med at bevise firefarvesætningen [OR13].

Den 17. juli 1879 mente en af Cayleys studerende, Alfred Bray Kempe, at have et bevis for firefarvesætningen. Kempe fik sit bevis udgivet i "*American Journal of Mathematics*" [OR13] efter netop Cayleys opfordning. Han modtog stor hæder for sit bevis. Men i 1890 viste Percy John Haewood, at Kempes bevis var forkert. Haewood skrev i sin artikel "*Map colouring theorem*" at hans formål var [OR13]:

> "[...] rather destructive than constructive, for it will be shown that there is a defect in the now apparently recognised proof."

Haewood var ikke selv i stand til at bevise firefarvesætningen, men han viste dog et svagere resultat, der gør brug af fem farver.

Der skulle gå godt 100 år, inden det endeligt lykkedes at lave et bevis for firefarvesætningen. Beviset blev lavet af Appel og Haken i 1976, og var revolutionerende pga. den metode, de havde anvendt til at lave det. De havde brudt resultatet ned til cirka 1800 enkelttilfælde, hvorefter de havde anvendt en computer til at undersøge tilfældene én efter én [Joh07]. De 1800 enkelttilfælde var så komplekse at det stort set var umuligt at verificere dem alle manuelt. Det tog også i omregnen af 1200 timer for computeren at gennemføre beviset [Joh07]. Resultatet var med til at skabe et paradigmeskifte indenfor matematik pga. bevismetoden. Der herskede stor skepsis om gyldigheden af resultatet, fordi man havde anvendt en computer. Dermed afhang tilliden til resultatet af tilliden til computere. Erfaringen med computere havde vist, at programmer som regel havde en tendens til at indeholde fejl.

Man fandt også fejl i nogle af de tilfælde, som computeren havde undersøgt. Appel og Haken forsøgte at udbedre de fundne fejl, men det gjorde ikke den skepsis der herskede mindre, tværtimod var den blot blevet forstærket af man havde fundet fejl.

I 2005 fik sagaen om firefarvesætningen en fortsættelse. Det lykkedes George Gonthier m. fl. at formalisere et bevis for firefarvesætningen gennem det bevisassisterende program Coq [Gon05]. Motivationen for at benytte Coq, skal netop ses i lyset af at mange stillede sig kritisk overfor anvendelse af computerstøtte i forbindelse med formel matematisk bevisførelse. Ideen med at benytte Coq var at fjerne afhængigheden til de mange forskellige computerplatforme, og dermed flytte afhængigheden over på en fælles platform. Resultatet heraf er, at beviset så kun afhænger af korrektheden af Coq. Gonthier m. fl. skrev et formelt program, som blev kørt igennem Coq, der til gengæld verificerede bevisets korrekthed [Gon05]. Hvis ikke man havde haft computerstøtte til at udføre beviset, så havde vi måske end ikke haft resultatet i vores tid.

1.4 Intuitionistisk logik

Brouwer formaliserede ikke selv sin intuitionisme. Den intuitionistiske logik vi kender i dag skyldes i høj grad Heyting, som oprindelig var en af Brouwers studerende [Hey56].

1.4.1 Brouwer-Heyting-Kolmogorov fortolkningen

I dette afsnit introducerer vi en særlig fortolkning af logiske konnektiver, der er kendt som Brouwer-Heyting-Kolmogorov (BHK)-fortolkningen, da den var implicit i Brouwers arbejde, og blev gjort eksplicit af Heyting, mens Kolmogorov foreslog den uafhængigt af de to førstnævnte [Tro98]. Fortolkningen er accepteret af konstruktivister [MJ12, Pal97].

Definition 1.6 (Brouwer-Heyting-Kolmogorov-fortolkningen)**.** Lad A, B være udsagn.

- Et bevis for $A \wedge B$ består af et bevis for A *og* et bevis for B.
- Et bevis for $A \vee B$ består af et bevis for A *eller* et bevis for B.
- Et bevis for $A \to B$ er en algoritme, der transformerer ethvert bevis for A til et bevis for B.
- Et bevis for $\neg A$ er en algoritme, der transformerer ethvert bevis for A til en absurditet.
- Der er intet bevis for $\perp$.

Vi siger at A er gyldig under BHK-fortolkningen, hvis vi kan lave en konstruktion p, således at p er et bevis for A.

Den overstående version af BHK-fortolkningen er mere eller mindre den version Heyting gav. Faktisk anså Heyting og Kolmogorov deres respektive versioner for at være forskellige. I Kolmogorovs fortolkning anses konstruktivisme som en kalkyle til problemløsning, dvs. enhver proposition repræsenterer et problem [Tro98]. Læst på denne måde siger BHK-fortolkningen at

- Problem $A \wedge B$ løses ved at løse både A og B.

- Problem $A \vee B$ løses ved at løse enten A eller B.
- $A \to B$ løses, hvis man kan reducere løsningen af problem B til en løsning af problem A.
- $\neg A$ løses, hvis man kan vise, at der ikke er en løsning til problem A.

I kontrast til Heytings fortolkning, anså Kolmogorov med denne fortolkning konstruktivt matematik som værende en udvidelse af klassisk matematik, blot med et rigere begreb om problemløsning. Heyting observerede dog, at "et problem A" indenfor Kolmogorovs fortolkning også blev udtrykt vha. et matematisk udsagn. Han oversatte så det "at løse problem A" til det at "lave et bevis for A". Derfor drog Heyting konklusionen, at de to fortolkninger var ens [Tro98]. Heraf har vi, hvorfor fortolkningen er navngivet BHK.

1.5 Problemformulering

Givet det stærke bånd mellem intuitionistisk logik og konstruktivisme, undrer det os, at vi finder anvendelse af ikke-konstruktive argumenter i resultater om intuitionistisk logik. I [vD08] bliver fuldstændighed af intuitionistisk logik, som er et helt centralt resultat, vist netop ved modstrid. Vi beskriver fuldstændighedssætningen og beviser den ud fra strategien i [vD08] i afsnit 3.5.2. Vi er ikke kun stødt på fænomenet i [vD08], men eksempelvis også i [SU98].

Konstruktivismens strenge krav til bevisførelse i mente, samt at konstruktivismen i vid udstrækning benytter sig af intuitionistisk logik, virker det besynderligt, at man ikke vælger at benytte sig af udelukkende konstruktive argumenter. Vi har set, at det udelukkede tredje princip er problematisk og ikke universelt gyldigt set fra et konstruktivistisk synspunkt. Det leder os derfor til følgende problemformulering, som tager udgangspunkt direkte i vores initierende problem:

Hvordan kan man bevise fuldstændighedssætningen for intuitionistisk logik udelukkende ved brug af konstruktive argumenter?

Problemet giver anledning til en række delspørgsmål

i) Hvilke forskelle er der mellem klassisk logik og intuitionistisk logik?

ii) Hvilke resultater er sætningen baseret på? Er de alle bevist ved brug af konstruktive argumenter?

iii) Hvordan kan vi implementere resultatet?

Det første delspørgsmål indbyder til en nærmere undersøgelse af både klassisk og intuitionistisk logik. For at kunne besvare selve problemformuleringen, skal vi selvfølgelig også forstå og være i stand til at formulere fuldstændighedssætningen, som den findes i [vD08]. Det tredje delspørgsmål giver os en datalogisk vinkling på problemet. Vi ønsker at undersøge, hvordan vi kan skabe en sammenhæng mellem vores bevis og et stykke programmel.

KAPITEL 2

Induktivt definerede mængder

"Beyond the natural numbers, addition, multiplication, and mathematical induction are intuitively clear."

— L. E. J. Brouwer (1881-1966) [OR13]

I dette kapitel giver vi en stringent definition af induktive mængder, fordi de er helt essentielle for det videre arbejde. Definitionerne og resultaterne i dette kapitel er derfor helt generelle for induktive mængder. Kapitlet indeholder en definition, tre hovedresultater og en diskussion af egenskaberne ved induktive mængder. Inspirationen til materialet i dette kapitel kommer fra [Hü10, Gal86, vD08], og vi har sammenflettet flere konkrete resultater og formuleret dem som mere abstrakte resultater.

2.1 Induktive mængder

Vi giver først en definition af induktive mængder, dernæst diskuterer vi et par tilhørende egenskaber.

Definition 2.1 (Induktiv mængde)**.** Lad M være en samling af objekter. Givet konstruktorer $f_1, \ldots, f_n$ defineret på M og aksiomer $A \subseteq M, A \neq \emptyset$ har vi en endofunktion $\mathcal{F}$, hvis billedmængde

$$\mathcal{F}(M) = \{y \mid y = f_i(x_1, \ldots, x_k),\, x_1, \ldots, x_k \in M,\, k < \infty\}$$

siges at være *induktiv* på A hvis og kun hvis

(i) $A \subseteq \mathcal{F}(M)$,

(ii) hvis $x_1, \ldots, x_k \in \mathcal{F}(M)$, så $f_i(x_1, \ldots, x_k) \in \mathcal{F}(M)$.

Desuden har $\mathcal{F}$ følgende egenskaber:

- $\mathcal{F}(\emptyset) = A$.

- $\mathcal{F}(M_1) \cup \mathcal{F}(M_2) \subseteq \mathcal{F}(M_1 \cup M_2)$.
- $M_1 \subseteq M_2 \Rightarrow \mathcal{F}(M_1) \subseteq \mathcal{F}(M_2)$ – $\mathcal{F}$ er monoton mht. $\subseteq$.

Den pedantiske matematiker ville sige at en mængde $\mathcal{F}(M)$ er induktiv med aksiomer A og konstruktorer $f_1, \ldots, f_n$. Hvis det er klart fra konteksten, så vil vi blot sige, at $\mathcal{F}$ er induktiv og lade aksiomerne og konstruktorer være underforstået. Vi siger, at en mængde S er induktivt defineret af $\mathcal{F}$ eller at $\mathcal{F}$ konstruerer S induktivt. Bemærk at konstruktorerne i definition 2.1 alle antages at være endofunktioner med endelig aritet, hvilket betyder at $f_i : M^n \to M$ for $n < \infty$. Det følger heraf, at mængden M i definition 2.1 er induktiv på A.

Punkt (ii) siger, at enhver induktiv mængde er lukket under $\mathcal{F}$. Som vi viser i sætning 2.5 så kan vi tænke på en induktivt defineret mængde som det mindste fikspunkt til konstruktørfunktionen $\mathcal{F}$. Inden vi formulerer og viser sætningen, vil give to karakteriseringer af dette fikspunkt. Betragt nu fællesmængden af alle induktive mængder under $\mathcal{F}$. Vi noterer den som

$$\mathcal{F}^* = \bigcap \{M \mid \mathcal{F}(M) \subseteq M\}. \tag{2.1}$$

Det følger direkte af definitionen, at $\mathcal{F}^*$ opfylder (i) og (ii) i definition 2.1, da den er fællesmængden af alle de mængder, der netop opfylder de to punkter. Altså er $\mathcal{F}^*$ selv en induktiv mængde. Bemærk at $\mathcal{F}^*$ aldrig kan være tom, da den er fællesmængden af alle induktive mængder på $\mathcal{F}(\emptyset)$, og vi har altid mindst en induktiv mængde på $\mathcal{F}(\emptyset)$, nemlig $M \supseteq \mathcal{F}(\emptyset)$.

Definition 2.2 (Induktiv aflukning)**.** Vi siger, at $\mathcal{F}^*$ er den *induktive aflukning* af $\mathcal{F}(\emptyset)$ under $\mathcal{F}$.

Lemma 2.3. *$\mathcal{F}^*$ er den mindste induktive mængde på $\mathcal{F}(\emptyset)$.*

Bevis. Lad X og Y betegne to vilkårlige induktive mængder under $\mathcal{F}$. Fra definitionen af $\mathcal{F}^*$ har vi at hvis $x \in \mathcal{F}^*$ så gælder $x \in X$ og $x \in Y$. Men hvis $y \in Y$ og $y \notin X$ så har vi $y \notin \mathcal{F}^*$ ifølge definitionen af $\mathcal{F}^*$. Altså må vi have at $\mathcal{F}^* \subseteq Y$ og $\mathcal{F}^* \subseteq X$, hvilket viser at $\mathcal{F}^*$ må være den mindste induktive mængde. □

Lad os nu definere dualen til $\mathcal{F}^*$. Antag $\mathcal{F}$ er givet, så kan vi iterativt anvende $\mathcal{F}$ på den tomme mængde $\emptyset$ et vilkårligt antal gange, eksempelvis $\mathcal{F}^2(\emptyset) = S$, hvor S er mængden konstrueret af $\mathcal{F}$ fra aksiomer $\mathcal{F}(\emptyset)$. Ved at anvende $\mathcal{F}$ på S konstruerer vi en ny mængde S' og så fremdeles. Observer nu, at monotoniciteten for $\mathcal{F}$ giver at

$$\mathcal{F}(\emptyset) \subseteq \mathcal{F}^2(\emptyset) \subseteq \mathcal{F}^3(\emptyset) \subseteq \ldots,$$

hvilket definerer en følge af mængder. Vi kan induktivt definere følgen $\{\mathcal{F}^i(\emptyset)\}_{i\geq 1}$ ved

$$\begin{aligned}\mathcal{F}(\emptyset) &= A,\\ \mathcal{F}^{i+1}(\emptyset) &= \mathcal{F}^i(\emptyset) \cup \{f_i(x_1,\ldots,x_k) \mid x_1,\ldots,x_k \in \mathcal{F}^i(A)\}.\end{aligned}$$

Det følger direkte af definitionen at $\mathcal{F}^i(\emptyset) \subseteq \mathcal{F}^{i+1}(\emptyset)$. Dermed er $\{\mathcal{F}^i(\emptyset)\}_{i\geq 1}$ en voksende følge mht. $\subseteq$. Men hvad er grænsen for denne følge? Intuitivt kan vi forestille os, at en grænse for $\{\mathcal{F}^i(\emptyset)\}_{i\geq 1}$ må være en supermængde for ethvert element i følgen. Lad os definere $\mathcal{F}_*$ som foreningsmængden af alle $\mathcal{F}^i(\emptyset)$, dvs.

$$\mathcal{F}_* = \bigcup_{i\geq 1} \mathcal{F}^i(\emptyset). \tag{2.2}$$

Hvis denne mængde er grænsen for $\{\mathcal{F}^i(\emptyset)\}_{i\geq 1}$, så skal den også være lukket under $\mathcal{F}$. Det følgende bemærkelsesværdige resultat for $\mathcal{F}^*$ og $\mathcal{F}_*$, viser at vores antagelse vedrørende $\mathcal{F}_*$ er korrekt.

Lemma 2.4 ([Gal86])**.** $\mathcal{F}^* = \mathcal{F}_*$.

Bevis. Vi viser først at $\mathcal{F}_*$ er en induktiv mængde. $\mathcal{F}^1(\emptyset)$ er per definition aksiomsmængden, og vi har derfor $\mathcal{F}(\emptyset) \subseteq \mathcal{F}_*$, hvilket viser at $\mathcal{F}_*$ opfylder (i) i definition 2.1. Vi skal nu vise at $\mathcal{F}_*$ er lukket under $\mathcal{F}$. For $x_1,\ldots,x_n \in \mathcal{F}_*$ er der en konstruktør f således at for et $i \geq 1$ har vi, at $x_1,\ldots,x_n \in \mathcal{F}^i(\emptyset)$, hvilket medfører $f(x_1,\ldots,x_n) \in \mathcal{F}^{i+1}(\emptyset)$. Heraf følger det at $f(x_1,\ldots,x_n) \in \mathcal{F}_*$. Dette betyder at $\mathcal{F}_*$ er induktiv på $\mathcal{F}(\emptyset)$.

Vi viser nu, at $\mathcal{F}_*$ og $\mathcal{F}^*$ er ækvivalente. Da $\mathcal{F}^*$ er den mindste induktive mængde på $\mathcal{F}(\emptyset)$, må vi have at $\mathcal{F}^* \subseteq \mathcal{F}_*$.

For at vise, at $\mathcal{F}_*$ er en delmængde af $\mathcal{F}^*$, kan vi vha. induktion vise at $\mathcal{F}^i(\emptyset) \subseteq \mathcal{F}^*$ for alle $i \geq 1$. Men dette er klart, fordi $\mathcal{F}^*$ er lukket under $\mathcal{F}$. Heraf har vi at $\mathcal{F}^* = \mathcal{F}_*$. $\square$

Resultatet medfører en del fleksibilitet i forhold til at konstruere den induktive aflukning af en induktiv mængde. Vi kender nu to ækvivalente konstruktionsmetoder, og kan derfor benytte den der gør arbejdet lettest i en given situation.

Vi er nu klar til at vise, at vi kan forstå en induktivt defineret mængde som et fikspunkt for konstruktørfunktionen $\mathcal{F}$.

Sætning 2.5 (Fikspunktsætningen for $\mathcal{F}$)**.** *Lad konstruktørfunktionen $\mathcal{F}$ være givet. Så er $M \stackrel{\text{def}}{=} \mathcal{F}_*$ den mindste mængde lukket under $\mathcal{F}$, der opfylder at $M = \mathcal{F}(M)$, dvs. M er et fikspunkt for $\mathcal{F}$.*

Bevis. For at bestemme fikspunktet, skal vi bestemme en løsning til ligningen $M = \mathcal{F}(M)$. Vi har allerede vist i (2.2), hvordan vi kan konstruere M. Lad os derfor vise, at $\mathcal{F}_*$ er et fikspunkt for $\mathcal{F}$.

$M \supseteq \mathcal{F}(M)$: Antag $y \in \mathcal{F}(M)$. Vi må for et $i \in \mathbb{N}$ have en konstruktør $f_i(x_1, \ldots, x_k) = y$, hvor $x_1, \ldots, x_k \in M$. Fordi k er endelig og ethvert x_j må findes i $M = \bigcup_{n \geq 0} \mathcal{F}^n(\emptyset)$ på et endeligt tidspunkt, så må der findes et endeligt n således at $x_1, \ldots, x_k \in \mathcal{F}^n(\emptyset)$. Det gælder derfor at $y \in \mathcal{F}(\mathcal{F}^n(\emptyset)) = \mathcal{F}^{n+1}(\emptyset)$, og definitionen for M giver, at det må være således at $y \in M$.

$M \subseteq \mathcal{F}(M)$: Antag at $y \in M = \bigcup_{j \geq 0} \mathcal{F}^j(\emptyset)$. Vi har så, at for et j gælder der at $y \in \mathcal{F}^j(\emptyset) = \mathcal{F}(\mathcal{F}^{j-1}(\emptyset))$. Det følger af definitionen af M og monotoniciteten for $\mathcal{F}$, at $\mathcal{F}(\mathcal{F}^{j-1}(\emptyset)) \subseteq \mathcal{F}(M)$. Derfor $y \in \mathcal{F}(M)$.

Heraf kan vi slutte at $M = \mathcal{F}(M)$. Lemma 2.4 giver at M er den mindste mængde lukket under $\mathcal{F}$. □

Korollar 2.6. *$\mathcal{F}_*$ er det mindste fikspunkt for $\mathcal{F}$. Vi noterer det* $\operatorname{fix} \mathcal{F}$.

Bevis. Lad $M \stackrel{\text{def}}{=} \mathcal{F}_*$, vi har fra sætning 2.5 at $M = \mathcal{F}(M)$. Antag nu at $S \neq M$ og $S = \mathcal{F}(S)$. Da M er den mindste mængde lukket under $\mathcal{F}$, må vi have at $M \subset S$, hvilket viser at $\operatorname{fix} \mathcal{F} = M$. □

Korollar 2.7. *Antag X er induktivt defineret af $\mathcal{F}$. Hvis $X \subseteq \mathcal{F}(X)$, så $X = \mathcal{F}_* = \mathcal{F}^* = \operatorname{fix} \mathcal{F}$.*

Bevis. Ligheden $\mathcal{F}_* = \mathcal{F}^* = \operatorname{fix} \mathcal{F}$ følger af lemma 2.4 og korollar 2.6. Det følger af (2.1) og (2.2) at $\mathcal{F}^* \subseteq X$ og $X \subseteq \mathcal{F}_*$, og heraf har vi $X = \mathcal{F}_* = \mathcal{F}^* = \operatorname{fix} \mathcal{F}$. □

De to karakteriseringer $\mathcal{F}^*$ og $\mathcal{F}_*$ af $\operatorname{fix} \mathcal{F}$ kaldes henholdsvis top-down og bottom-up definitioner. Top-down karakteriseringen skyldes Tarski [Tar55], mens bottom-up karakteriseringen skyldes både Tarski og Knaster [KT28].

2.2 Induktionsprincippet

En konsekvens af korollar 2.7 er *induktionsprincippet*, som uformelt siger, at hvis vi vil vise en påstand Φ gælder for alle $x \in \operatorname{fix} \mathcal{F}$, så skal vi betragte

$$X = \{x \mid \Phi(x) \text{ er sand}\}$$

og vise at $\operatorname{fix} \mathcal{F} \subseteq X$. Dette kan vi gøre ved at vise at $\mathcal{F}(X) \subseteq X$, hvilket svarer til at vise at hvis Φ er sand for et x, så er Φ også sand for $\mathcal{F}(x)$, hvilket kaldes strukturel induktion. Strukturel induktion er beskrevet nærmere i appendiks B.2. Vi formulerer nu dette bevisprincip formelt og giver et bevis.

Sætning 2.8 (Induktionsprincippet [vD08])**.** *Lad $\mathcal{F}$ være givet med konstruktorer $f_1, \ldots, f_n$. Antag Φ er en påstånd, så gælder Φ for alle $x \in \operatorname{fix} \mathcal{F}$ hvis* *(i)* $\Phi(x)$ *gælder for alle* $x \in \mathcal{F}(\emptyset)$*,*

(ii) $\Phi(x_1), \ldots, \Phi(x_k) \Rightarrow \Phi(x)$*, hvor* $x = f_i(x_1, \ldots, x_k)$*.*

Bevis. Lad X være en samling af objekter og Φ være defineret for elementerne i X. Antag at $\mathcal{F}(X) = \{x \in X \mid \Phi(x) \text{ gælder}\}$ er induktiv. Vi har så at $\mathcal{F}^* \subseteq \mathcal{F}(X)$, hvilket medfører at $\Phi(x)$ holder for alle $x \in \operatorname{fix} \mathcal{F}$. □

Induktionsprincippet og induktivt definerede mængder udgør tilsammen næsten hele fundamentet for vores videre arbejde, og vil blive benyttet ofte.

2.3 Rekursive definitioner

Vi har set induktionsprincippet, men der er endnu et vigtigt resultat vi mangler, et resultat som er særdeles anvendeligt sammen med induktionsprincippet. Antag at vi ønsker at definere en funktion F på en induktiv mængde M. Det åbenlyse spørgsmål er: Hvordan kan vi sikre at vores definition af F på M er konsistent? Det følgende resultat giver svaret på dette spørgsmål.

Sætning 2.9 (Rekursive definitioner [vD08])**.** *Lad* $\mathcal{F}$ *med konstruktorer* $f_1, \ldots, f_m$ *være givet. Desuden lad følgende entydige afbildninger være givet*

$$\begin{aligned} H_a &: \mathcal{F}(\emptyset) \to A, \\ H_{f_i} &: A^n \to A, \end{aligned}$$

hvor $n < \infty$ *og* A *er en vilkårlig mængde. Så findes der en entydigt givet afbildning* $F : \mathcal{F}^* \to A$ *således at*

$$F(x) = \begin{cases} H_a(x) & \text{hvis } x \in \mathcal{F}(\emptyset), \\ H_{f_i}(F(x_1), \ldots, F(x_k)) & \text{hvis } x = f_i(x_1, \ldots, x_k). \end{cases}$$

Bevis. Antag funktionerne H_a og H_{f_i} er givet for $i = 1, \ldots, m$. Vi viser først entydighed. Antag at der findes to afbildninger $F_1, F_2 : \mathcal{F}^* \to A$, som opfylder betingelsen ovenfor. For at vise $F_1 = F_2$ skal vi vise, at deres billedmængder er ens for alle $x \in \mathcal{F}^*$. Betragt nu mængden

$$X \stackrel{\text{def}}{=} \{x \in \mathcal{F}^* \mid F_1(x) = F_2(x)\},$$

og bemærk, at hvis vi kan vise $X = \mathcal{F}^*$, så har vi vist $F_1 = F_2$.

Basisskridt Da H_a er entydigt bestemt, er det klart at $F_1(x) = F_2(x) = H_a(x)$ for alle $x \in \mathcal{F}(\emptyset)$, hvilket medfører at $\mathcal{F}(\emptyset) \subseteq X$.

Induktionsskridt Lad $y \in \mathcal{F}^*$. Antag $F_1(y) = H_{f_i}(F_1(x_1), \ldots, F_1(x_k))$ og $F_2(y) = H_{f_i}(F_2(x_1), \ldots, F_2(x_k))$ for $x_1, \ldots, x_k \in X$. Hvis $x \in \{x_1, \ldots, x_k\}$ så gælder $F_1(x) = F_2(x)$ således, at vi per induktionsantagelse får

$$F_1(y) = H_{f_i}(F_1(x_1), \ldots, F_1(x_k)) = H_{f_i}(F_2(x_1), \ldots, F_2(x_k)) = F_2(y).$$

Det viser at $F_1(y) = F_2(y)$, hvilket medfører at $y \in X$. Da x er valgt vilkårligt følger det af matematisk induktion, at $X = \mathcal{F}^*$.

Deraf følger det, at $F_1 = F_2$.

Vi viser nu eksistens. Lad F opfylde betingelserne i sætningen. For at vise F er en funktion, skal vi vise at der for ethvert $x \in \mathcal{F}^*$ findes præcist ét y således at $y = F(x)$. Vi viser dette vha. strukturel induktion.

Basisskridt Vi har $x \in \mathcal{F}(\emptyset)$. Da H_a er en funktion, følger det at vi får $F(x) = H_a(x) = y$ for et $y \in A$.

Rekursionsskridt Antag x er et sammensat element, således at $x = f_i(x_1, \ldots, x_k)$. Betragt følgende mængde

$$\{y \mid y = H_{f_i}(F(x_1), \ldots, F(x_k))\}.$$

Vi har per induktionsantagelse at for alle $j \in [1, k]$ findes præcist ét y_j således at $y_j = F(x_j)$, og da H_{f_i} er en funktion, følger det at mængden kun indeholder ét element, som er

$$y = H_{f_i}(y_1, \ldots, y_k) = F(x).$$

$\square$

KAPITEL 3

Udsagnslogik

"Nothing new had been done in Logic since Aristotle!"
— Kurt Gödel (1906-1978) [Li13]

Matematik benytter sig mere eller mindre eksplicit af et system af formel logik til at bevise sætninger. Vi vil i dette kapitel beskrive udsagnslogik formelt i en intuitionistisk udgave og delvis i en klassisk udgave. Vi har valgt at fokusere på udsagnslogik, da mange andre logiske systemer indeholder en version af udsagnslogik [HP02], og udsagnslogik kan derfor ses som værende grundlæggende i en vis forstand for disse andre logiske systemer.

3.1 Syntaks

I dette afsnit vil vi give en syntaks for udsagnslogik som vil danne grundlag for en beskrivelse af både klassisk og intuitionistisk udsagnslogik. Vores beskrivelse tager udgangspunkt i [vD08] og [SU98].

Vores intuitionistiske udsagnslogik har følgende alfabet:

- Atomare sætninger: $p_1, p_2, \dots$
- Konnektiver: $\vee, \wedge, \rightarrow, \bot$
- Andet: (,)

I udsagnslogik består hver atomar sætning af et udsagn. Vi vil benytte symbolerne p og q som metavariable for atomare sætninger.

Ud fra dette alfabet kan vi definere alle mulige udsagn ud fra den følgende grammatik:

$$\varphi ::= \bot \,|\, p \,|\, (\varphi \rightarrow \varphi) \,|\, (\varphi \vee \varphi) \,|\, (\varphi \wedge \varphi). \tag{3.1}$$

Bemærk at de ofte brugte konnektiver $\neg$ og $\leftrightarrow$ kan beskrives ud fra denne grammatik [SU98]. $\neg p$ i den intuitionistiske udsagnslogik betyder, at der findes en konstruktion som forvandler enhver konstruktion af p til et ikke-eksisterende objekt, mens $\neg p$ og $(p \rightarrow \bot)$ er ækvivalente i klassisk udsagnslogik. $\neg p$ kan derfor skrives som $(p \rightarrow \bot)$. Ligeledes kan $(p \leftrightarrow q)$

skrives som $((p \to q) \wedge (q \to p))$. Grammatikken (3.1) angiver præcist, hvilke opbygningsregler vi kan anvende, når vi konstruerer et udsagn. Faktisk giver grammatikken en induktiv definition af et udsagn. Konnektiverne er konstruktorer, og de atomare udsagn er aksiomer. Til senere reference, lad $\mathcal{P}$ være navnet på konstruktørfunktionen, således vi har den induktive aflukning $\mathcal{P}^*$. Den induktive aflukning er allerede veldefineret, så følgende er givet direkte fra afsnit 2.1.

Lemma 3.1. $\mathcal{P}^* = \operatorname{fix} \mathcal{P}$*, dvs. følgende egenskaber er opfyldt*

(i) $p_i \in \mathcal{P}^*$*, for* $i \in \mathbb{N}$ *og* $\perp \in \mathcal{P}^*$*,*

(ii) $\varphi_1, \varphi_2 \in \mathcal{P}^* \Rightarrow (\varphi_1 \wedge \varphi_2), (\varphi_1 \vee \varphi_2), (\varphi_1 \to \varphi_2) \in \mathcal{P}^*$*,*

(iii) $\varphi \in \mathcal{P}^* \Rightarrow (\neg\varphi) \in \mathcal{P}^*$*.*

Bevis. Resultatet gælder ifølge vores diskussion af induktive mængder i afsnit 2.1 og grammatikken (3.1). $\square$

I punkt (iii) skriver vi $(\neg\varphi)$ istedet for $(\varphi \to \perp)$, vi benytter denne mere kompakte notation fremover. Desuden når vi behandler (ii) generelt, vil vi benytte notationen $(\varphi_1 \square \varphi_2)$, hvor $\square$ er et af de binære konnektiver $(\to, \vee, \wedge)$. Fremover vil vi undlade at skrive parenteserne, medmindre de er nødvendige for at fremhæve konnektivernes præcedens.

Da vi har defineret et udsagn φ som værende en mængde, så giver det mening at snakke om delelementer af φ. Vi kalder et delelement af φ for et deludsagn, og vi inddeler deludsagn i to kategorier: *umiddelbare deludsagn* og *deludsagn*. I definition 3.2 giver vi en definition af begrebet.

Definition 3.2 (Deludsagn)**.** Et *deludsagn* er defineret rekursivt på konstruktionen af φ:

- Ethvert udsagn φ er et deludsagn af sig selv.
- Udsagnene φ_1 og φ_2 er umiddelbare deludsagn af $\varphi = \varphi_1 \square \varphi_2$.
- Udsagnet φ' er et umiddelbart deludsagn af $\varphi = \neg\varphi'$.
- Hvis φ_1 er et umiddelbart deludsagn af φ_2 og φ_2 er et umiddelbart deludsagn af φ_3, så er φ_1 et deludsagn af φ_3.

Vi siger at φ' er et *ægte deludsagn* af φ, hvis φ' er et deludsagn af φ og $\varphi' \neq \varphi$.

3.2 Bevisregler

Vi er interesserede i sammenhængen mellem sandhedsværdi og beviselighed, og vi vil derfor i dette afsnit give bevisreglerne for den udsagnslogik, vi har beskrevet. Syntaktisk er der, som beskrevet i afsnit 3.1, ingen forskel på den klassiske og den intuitionistiske udsagnslogik, men bevisreglerne for de to logikker er forskellige, og vi har derfor brug for to forskellige regelsæt.

Der findes flere forskellige systemer til at beskrive bevisregler, blandt andet Gentzens sekventkalkyle og Hilbert-stil bevisteori. Vi benytter os dog her af naturlig deduktion, som også skyldes Gentzen. [HP02] Bevisreglerne for inutitionistisk udsagnslogik er baseret på BHK-fortolkningen beskrevet i afsnit 1.4.1. Et bevis for $\varphi_1 \wedge \varphi_2$ kræver derfor at både φ_1 og φ_2 er bevist, mens et bevis for $\varphi_1 \vee \varphi_2$ kræver enten et bevis for φ_1 eller φ_2, og $\varphi_1 \to \varphi_2$ kræver at φ_1 kan laves om til et bevis for φ_2. Ud fra disse betragtninger får vi bevisreglerne for intuitionistisk logik, som er givet ved reglerne i tabel 3.1.

$$(\wedge\mathrm{I}) \quad \frac{\varphi_1 \quad \varphi_2}{\varphi_1 \wedge \varphi_2}\wedge\mathrm{I} \qquad\qquad (\wedge\mathrm{E}) \quad \frac{\varphi_1 \wedge \varphi_2}{\varphi_1}\wedge\mathrm{E} \text{ og } \frac{\varphi_1 \wedge \varphi_2}{\varphi_2}\wedge\mathrm{E}$$

$$(\vee\mathrm{I}) \quad \frac{\varphi_1}{\varphi_1 \vee \varphi_2}\vee\mathrm{I} \text{ og } \frac{\varphi_2}{\varphi_1 \vee \varphi_2}\vee\mathrm{I} \qquad (\vee\mathrm{E}) \quad \frac{\varphi_1 \vee \varphi_2 \quad \begin{matrix}[\varphi_1]\\ \vdots \\ \varphi_3\end{matrix} \quad \begin{matrix}[\varphi_2]\\ \vdots \\ \varphi_3\end{matrix}}{\varphi_3}\vee\mathrm{E}$$

$$(\to\mathrm{I}) \quad \frac{\begin{matrix}[\varphi_1]\\ \vdots \\ \varphi_2\end{matrix}}{\varphi_1 \to \varphi_2}\to\mathrm{I} \qquad\qquad (\to\mathrm{E}) \quad \frac{\varphi_1 \quad \varphi_1 \to \varphi_2}{\varphi_2}\to\mathrm{E}$$

$$(\bot) \quad \frac{\bot}{\varphi}\bot$$

Tabel 3.1: Bevisreglerne i naturlig deduktion for intuitionistisk udsagnslogik [vD08].

Bevisreglerne for klassisk udsagnslogik er givet ved de samme regler som i tabel 3.1, dog med tilføjelse af reglen (RAA), som kan ses i tabel 3.2. (RAA) er netop den regel, som tillader modstridsbeviser, da den tillader os at konkludere φ hvis vi kan vise, at $\neg\varphi$ fører til en modstrid. Efter vores betragtninger om UTP i kapitel 1 er det derfor ikke overraskende, at (RAA)-reglen er gyldig for klassisk, men ikke intuitionistisk, udsagnslogik.

$$\begin{array}{ll} \text{(RAA)} & \dfrac{\begin{array}{c}[\neg\varphi] \\ \vdots \\ \bot\end{array}}{\varphi}\,\text{RAA} \end{array}$$

Tabel 3.2: RAA-reglen [vD08].

Notationen betyder, at givet udsagnene over stregen, kan udsagnet under stregen udledes. Udsagnene over stregen kaldes *præmisser*, mens udsagnet under stregen kaldes *konklusionen* [vD08]. Notationen for ($\to$I) og (RAA) er en smule anderledes. Her benyttes en *hypotese* (φ_1 for ($\to$I) og $\neg\varphi$ for (RAA)), som angiver, at hvis man ud fra hypotesen kan udlede præmissen, så følger konklusionen. I ($\to$I) betyder det altså, at hvis vi ud fra φ_1 kan udlede φ_2, så kan vi konkludere at $\varphi_1 \to \varphi_2$. Læg dog også mærke til, at hypotesen står i firkantede parenteser. Dette angiver, at hypotesen er blevet fjernet. Bemærk desuden, at ($\to$I)-reglen også kan bruges uden hypoteser [vD08]. Reglen har da følgende form:

$$\frac{\varphi_1}{\varphi_2 \to \varphi_1} \to I.$$

Hvis det er muligt at lave en udledning med et givet udsagn φ som konklusion, siger vi at φ kan *bevises* [HP02]. Dette indfanges af følgende definition.

Definition 3.3 ([vD08])**.** Lad Γ være en mængde af hypoteser. Så er relationen $\Gamma \vdash_k \varphi$ defineret ved, at der findes en udledning ved brug af bevisreglerne for klassisk udsagnslogik med konklusion φ og en mængde ikke-fjernede hypoteser Γ. Hvis $\Gamma = \emptyset$, så skriver vi blot $\vdash_k \varphi$, og så kaldes φ et *teorem* i klassisk logik. Ligeledes er $\Gamma \vdash_i \varphi$ defineret ved, at der findes en en udledning ved brug af bevisreglerne for intuitionistisk udsagnslogik, og $\vdash_i \varphi$ angiver, at φ er et teorem i intuitionistisk logik.

En af forskellene på intuitionistisk udsagnslogik og klassisk udsagnslogik er at klassisk udsagnslogik ikke behøver reglerne ($\vee$I) og ($\vee$E). $\vee$-konnektivet kan nemlig beskrives ud fra konnektiverne $\neg$ og $\wedge$ i klassisk udsagnslogik. Vi har i tabel 3.3 og tabel 3.4 vist, at

$$\vdash_k \varphi_1 \vee \varphi_2 \leftrightarrow \neg(\neg\varphi_1 \wedge \neg\varphi_2), \tag{3.2}$$

hvilket betyder at vi i klassisk logik kan skrive $\neg(\neg\varphi_1 \wedge \neg\varphi_2)$ i stedet for $\varphi_1 \vee \varphi_2$. Bemærk dog, at i tabel 3.4, benyttes (RAA)-reglen, og teorem (3.2) er derfor ikke gyldig i intuitionistisk udsagnslogik, hvorfor ($\vee$I) og ($\vee$E) er nødvendige i intuitionistisk udsagnslogik.

$$
\begin{prooftree}
\AxiomC{$[\varphi_1 \vee \varphi_2]^4$}
\AxiomC{$[\varphi_1]^3$}
\AxiomC{$[\neg\varphi_1 \wedge \neg\varphi_2]^1$}
\RightLabel{$\wedge$E}
\UnaryInfC{$\neg\varphi_1$}
\RightLabel{$\rightarrow$E}
\BinaryInfC{$\bot$}
\RightLabel{$\rightarrow$I$_1$}
\UnaryInfC{$\neg(\neg\varphi_1 \wedge \neg\varphi_2)$}
\AxiomC{$[\varphi_2]^3$}
\AxiomC{$[\neg\varphi_1 \wedge \neg\varphi_2]^2$}
\RightLabel{$\wedge$E}
\UnaryInfC{$\neg\varphi_2$}
\RightLabel{$\rightarrow$E}
\BinaryInfC{$\bot$}
\RightLabel{$\rightarrow$I$_2$}
\UnaryInfC{$\neg(\neg\varphi_1 \wedge \neg\varphi_2)$}
\RightLabel{$\vee$E$_3$}
\TrinaryInfC{$\neg(\neg\varphi_1 \wedge \neg\varphi_2)$}
\RightLabel{$\rightarrow$I$_4$}
\UnaryInfC{$\varphi_1 \vee \varphi_2 \rightarrow \neg(\neg\varphi_1 \wedge \neg\varphi_2)$}
\end{prooftree}
$$

Tabel 3.3: Derivation for $\vdash_k \varphi_1 \vee \varphi_2 \rightarrow \neg(\neg\varphi_1 \wedge \neg\varphi_2)$.

$$
\begin{prooftree}
\AxiomC{$[\neg(\varphi_1 \vee \varphi_2)]^3$}
\AxiomC{$[\varphi_1]^1$}
\RightLabel{$\vee$I}
\UnaryInfC{$\varphi_1 \vee \varphi_2$}
\RightLabel{$\rightarrow$E}
\BinaryInfC{$\bot$}
\RightLabel{$\rightarrow$I$_1$}
\UnaryInfC{$\neg\varphi_1$}
\AxiomC{$[\varphi_2]^2$}
\RightLabel{$\vee$I}
\UnaryInfC{$\varphi_1 \vee \varphi_2$}
\AxiomC{$[\neg(\varphi_1 \vee \varphi_2)]^3$}
\RightLabel{$\rightarrow$E}
\BinaryInfC{$\bot$}
\RightLabel{$\rightarrow$I$_2$}
\UnaryInfC{$\neg\varphi_2$}
\RightLabel{$\wedge$I}
\BinaryInfC{$\neg\varphi_1 \wedge \neg\varphi_2$}
\AxiomC{$[\neg(\neg\varphi_1 \wedge \neg\varphi_2)]^4$}
\RightLabel{$\rightarrow$E}
\BinaryInfC{$\bot$}
\RightLabel{RAA$_3$}
\UnaryInfC{$\varphi_1 \vee \varphi_2$}
\RightLabel{$\rightarrow$I$_4$}
\UnaryInfC{$\neg(\neg\varphi_1 \wedge \neg\varphi_2) \rightarrow \varphi_1 \vee \varphi_2$}
\end{prooftree}
$$

Tabel 3.4: Derivation for $\vdash_k \neg(\neg\varphi_1 \wedge \neg\varphi_2) \rightarrow \varphi_1 \vee \varphi_2$.

$$\dfrac{\dfrac{\dfrac{[\varphi]^2 \quad [\neg\varphi]^1}{\bot}\,{\to}\mathrm{E}}{\neg\neg\varphi}\,{\to}\mathrm{I}_1}{\varphi \to \neg\neg\varphi}\,{\to}\mathrm{I}_2$$

Tabel 3.5: Derivation for $\vdash_k \varphi \to \neg\neg\varphi$. [vD08]

$$\dfrac{\dfrac{\dfrac{[\neg\varphi]^1 \quad [\neg\neg\varphi]^2}{\bot}\,{\to}\mathrm{E}}{\varphi}\,\mathrm{RAA}_1}{\neg\neg\varphi \to \varphi}\,{\to}\mathrm{I}_2$$

Tabel 3.6: Derivation for $\vdash_k \neg\neg\varphi \to \varphi$. [vD08]

En anden væsentlig forskel på klassisk og intuitionistisk udsagnslogik er hvordan dobbelt negation opfattes. I den klassiske logik opfattes dobbelt negation som ækvivalent med ingen negation. I tabel 3.5 og tabel 3.6 har vi udledt at

$$\vdash_k \varphi \leftrightarrow \neg\neg\varphi. \tag{3.3}$$

$\neg\neg\varphi$ kan altså erstattes med φ i klassisk udsagnslogik. Som i det foregående bemærker vi her, at tabel 3.6 gør brug af (RAA)-reglen, og teorem (3.3) er derfor ugyldig for intuitionistisk udsagnslogik.

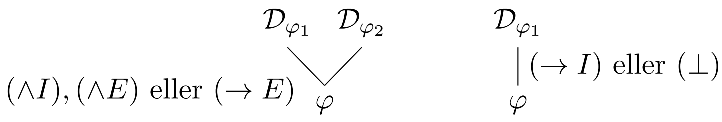
Dette kan også ses ved at bemærke, at (RAA) faktisk blot er en anvendelse af ($\to$I)-reglen, hvor teorem (3.3) benyttes. Vi får da, at

$$\dfrac{\begin{matrix}[\neg\varphi]^1\\ \vdots \\ \bot\end{matrix}}{\neg\neg\varphi}\,{\to}\mathrm{I}_1$$

og hvis vi erstatter $\neg\neg\varphi$ med φ, får vi (RAA)-reglen.

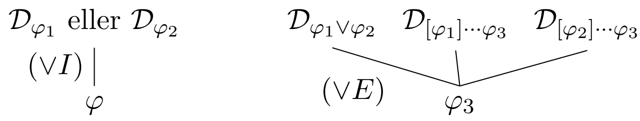
3.3 Derivationstræer

I dette afsnit introducerer vi en særlig vigtig matematisk træstruktur kaldet et derivationstræ. Derivationstræer bliver vigtige for os i forbindelse med bevisførelse. Vi giver en induktiv definition af et derivationstræ.



(a) (b)

Figur 3.1: Visualisering af konstruktorerne $(\wedge I), (\wedge E), (\rightarrow E), (\rightarrow I)$ og $(\bot)$.



(a) (b)

Figur 3.2: Visualisering af konstruktorerne $(\vee I)$ og $(\vee E)$.

Definition 3.4 (Derivationstræ)**.** Et derivationstræ $\mathcal{D}$ med konklusion φ er et træ, hvis knuder er elementer i $\mathcal{P}^*$, og kanterne er instanser af bevisreglerne fra tabel 3.1. Dvs. at alle udsagn er i sig selv et derivationstræ med et enkelt element. Vi kan forstå et derivationstræ som en induktiv mængde med aksiomer $\mathcal{P}^*$, hvor bevisreglerne agerer konstruktorer. Følgende viser hvorledes vi konstruerer et derivationstræ for et udsagn φ.

$(\wedge I), (\wedge E), (\rightarrow E)$. Givet derivationstræer $\mathcal{D}_{\varphi_1}$ og $\mathcal{D}_{\varphi_2}$ med henholdsvis φ_1 og φ_2 som rødder, konstruerer vi $\mathcal{D}_{\varphi}$ ved at forbinde roden af $\mathcal{D}_{\varphi_1}$ med en kant til knuden φ, ligeledes forbinder vi roden af $\mathcal{D}_{\varphi_2}$ med knuden φ. Se figur 3.1a for en visualisering af dette.

$(\rightarrow I), (\bot)$. Givet et derivationstræ $\mathcal{D}_{\varphi_1}$ med φ_1 som rod, konstruerer vi $\mathcal{D}_{\varphi}$ ved at forbinde roden af $\mathcal{D}_{\varphi_1}$ med knuden φ. Se figur 3.1b.

$(\vee I)$. Givet derivationstræ $\mathcal{D}_{\varphi_i}$, hvor $i = 1$ eller $i = 2$. Så konstruerer vi $\mathcal{D}_{\varphi}$ ved at forbinde roden af $\mathcal{D}_{\varphi_i}$ med knuden $\varphi = \varphi_1 \vee \varphi_2$. Se figur 3.2a.

$(\vee E)$. Givet derivationstræer $\mathcal{D}_{\varphi_1 \vee \varphi_2}, \mathcal{D}_{[\varphi_1] \cdots \varphi_3}$ og $\mathcal{D}_{[\varphi_2] \cdots \varphi_3}$. Vi konstruerer $\mathcal{D}_{\varphi_3}$ ved at forbinde rødderne af $\mathcal{D}_{\varphi_1 \vee \varphi_2}, \mathcal{D}_{[\varphi_1] \cdots \varphi_3}, \mathcal{D}_{[\varphi_2] \cdots \varphi_3}$ med knuden φ_3. Se figur 3.2b.

Roden i $\mathcal{D}$ er konklusionen φ.

Notationsmæssigt benytter vi notationen $\mathcal{D}_{\varphi}$ til at betegne derivationstræet, hvor φ er roden (konklusionen). Eftersom vi har defineret et derivationstræ som værende et en matematisk træstruktur, så giver det mening at benytte begrebet "højden" af et derivationstræ. Vi benytter sætning 2.9 og giver en rekursiv funktion for højden af et derivationstræ.

Definition 3.5 (Højdefunktionen)**.** Vi definerer højden af $\mathcal{D}_\varphi$, som den rekursive funktion $h : \mathcal{D}^* \to \mathbb{N}$.

$$h(\mathcal{D}_\varphi) = \begin{cases} 0 & \text{hvis } \mathcal{D}_\varphi \text{ er et blad.} \\ 1 + \max\limits_{j \leq n}(h(\mathcal{D}_{\varphi_j})) & \text{hvis } \mathcal{D}_\varphi \text{ har } n \geq 1 \text{ forgreninger.} \end{cases}$$

hvor φ_j en præmis til et vilkårligt udsagn φ.

Definitionen er ifølge sætning 2.9 konsistent, da h har et tilfælde for hver mulig konstruktion af $\mathcal{D}$. Begrebet om højde af et derivationstræ er vigtigt i sammenhæng med bevisførelse hvori der indgår derivationstræer.

3.4 Semantik

For at beskrive semantikken for intuitionistisk udsagnslogik vil vi benytte os af en Kripke-semantik. Først skal vi bruge følgende definition.

Definition 3.6 ([Hü10])**.** En *partielt ordnet mængde* er et par $(D, \sqsubseteq)$, hvor D er en mængde og $\sqsubseteq$ er en refleksiv, antisymmetrisk og transitiv binær relation over D.

En Kripke-semantik benytter sig af en partielt ordnet mængde, hvor der til hvert element i mængden er tilknyttet en mængde af sætninger i et givet sprog, som er blevet anerkendt som værende sande for netop det element. Hvert element i den partielt ordnede mængde kan altså ses som et erkendelsesniveau for en tænkt matematiker, hvor hvert erkendelsesniveau repræsenterer den viden, matematikeren har på dette niveau.

Definition 3.7 (Kripke-model [vD08])**.** En *Kripke-model* for et sprog $\mathcal{L}$ er et par $\mathcal{K} = (K, \Sigma)$, hvor K er en partielt ordnet mængde, og Σ er en funktion på K som opfylder at

(a) $\Sigma(k) \subseteq \mathcal{L}(\emptyset)$ for alle $k \in K$, hvor $\mathcal{L}(\emptyset)$ er de atomare sætninger i $\mathcal{L}$.

(b) $\perp \notin \Sigma(k)$ for alle $k \in K$.

(c) $k \sqsubseteq l \Rightarrow \Sigma(k) \subseteq \Sigma(l)$, for alle $l, k \in K$.

Kripke-modellen sikrer, at den tænkte matematikers viden ikke kan blive mindre, og at matematikeren aldrig anerkender absurditeter som værende sande. Definitionen definerer kun modellen for de atomare sætninger i $\mathcal{L}$. For enhver Kripke-model kan vi definere en relation rekursivt, som beskriver modellen for sammensatte sætninger. Denne relation er defineret i definition 3.8.

Definition 3.8. *Forcing-relationen* $\Vdash_{\mathcal{K}}$ for en givet Kripke-model $\mathcal{K}$ er defineret ved

- $k \Vdash_{\mathcal{K}} p$ hvis og kun hvis $p \in \Sigma(k)$.
- $k \Vdash_{\mathcal{K}} \varphi_1 \wedge \varphi_2$ hvis og kun hvis $k \Vdash_{\mathcal{K}} \varphi_1$ og $k \Vdash_{\mathcal{K}} \varphi_2$.
- $k \Vdash_{\mathcal{K}} \varphi_1 \vee \varphi_2$ hvis og kun hvis $k \Vdash_{\mathcal{K}} \varphi_1$ eller $k \Vdash_{\mathcal{K}} \varphi_2$.
- $k \Vdash_{\mathcal{K}} \varphi_1 \rightarrow \varphi_2$ hvis og kun hvis for alle $l \sqsupseteq k$ der gælder at $l \Vdash_{\mathcal{K}} \varphi_1 \Rightarrow l \Vdash_{\mathcal{K}} \varphi_2$.

Hvis det er klart ud fra konteksten, hvilken Kripke-model der er tale om, vil vi undlade $_{\mathcal{K}}$'et og blot skrive $\Vdash$.

Givet en mængde Γ, skriver vi $\Gamma \Vdash \varphi$ hvis for alle $\mathcal{K}$ der gælder at for alle $k \in K$ fås at hvis $k \Vdash \varphi'$ for alle $\varphi' \in \Gamma$, så gælder $k \Vdash \varphi$. Hvis der for alle Kripke-modeller $\mathcal{K}$ gælder at $k \Vdash \varphi$ for alle $k \in K$, så skriver vi $\Vdash \varphi$ og φ kaldes da en *tautologi* for den intuitionistiske udsagnslogik.

Hvis Γ er en mængde af udsagn, så skriver vi $k \Vdash \Gamma$ hvis $k \Vdash \varphi$ for alle $\varphi \in \Gamma$.

3.5 Sundhed og fuldstændighed af intuitionistisk udsagnslogik

I dette afsnit vil vi vise både sundhed og fuldstændighed af intuitionistisk udsagnslogik. Sundhed er den egenskab, at hvis et udsagn φ kan bevises ud fra en mængde præmisser Γ, så er det også sandt. Da sandhed i intuitionistisk logik ikke er et primitv, men i stedet defineres ud fra forcing-relationen, er sundhed i intuitionistisk logik den egenskab at

$$\Gamma \vdash_i \varphi \Rightarrow \Gamma \Vdash \varphi$$

er opfyldt. Omvendt er fuldstændighed den egenskab, at

$$\Gamma \Vdash \varphi \Rightarrow \Gamma \vdash_i \varphi,$$

og hvis et logisk system både er sundt og fuldstændigt, gælder der derfor at der er overensstemmelse mellem sandhed og beviselighed, i den forstand at alle sande udsagn kan bevises, og alle beviselige udsagn er sande.

3.5.1 Sundhed

I dette afsnit vil vi diskutere og bevise sundhedssætningen for intuitionistisk udsagnslogik. Vi introducerer først monotonicitetssætningen for forcing-relationen. Monotonicitetssætningen er en udvidelse af betingelse (c) i definition 3.7 for en Kripke-model til også at gælde for sammensatte udsagn. Sætningen fortæller os, at det vi ved på et givet tidspunkt, ved vi også på et senere tidspunkt derefter. Populært sagt, så kan vi ikke blive “dummere”.

Lemma 3.9 (Monotonicitet for $\Vdash$ [vD08])**.** *Hvis $k \sqsubseteq l$, så gælder at $k \Vdash \varphi \Rightarrow l \Vdash \varphi$.*

Bevis. Strukturel induktion på φ.

Basisskridt For p (atomare φ) følger det direkte af definition 3.7.

Rekursionsskridt For sammensatte udsagn. Vi antager i hvert af tilfældene, at resultatet holder for de umiddelbare deludsagn.

$\varphi = \varphi_1 \wedge \varphi_2$. Antag $k \Vdash \varphi_1 \wedge \varphi_2$ og $k \sqsubseteq l$. Da $k \Vdash \varphi_1 \wedge \varphi_2$, følger det af definition 3.8 at $k \Vdash \varphi_1$ og $k \Vdash \varphi_2$. Induktionsantagelsen giver os at ($l \Vdash \varphi_1$ og $l \Vdash \varphi_2$), hvilket gælder hvis og kun hvis $l \Vdash \varphi_1 \wedge \varphi_2$.

$\varphi = \varphi_1 \vee \varphi_2$. Antag $k \Vdash \varphi_1 \vee \varphi_2$ og $k \sqsubseteq l$. Så har vi $k \Vdash \varphi_1 \vee \varphi_2$, som ifølge definition 3.8 er ensbetydende med $k \Vdash \varphi_1$ eller $k \Vdash \varphi_2$. Induktionsantagelsen giver os at ($l \Vdash \varphi_1$ eller $l \Vdash \varphi_2$), hvilket gælder hvis og kun hvis $l \Vdash \varphi_1 \vee \varphi_2$.

$\varphi = \varphi_1 \rightarrow \varphi_2$. Antag $k \Vdash \varphi_1 \rightarrow \varphi_2$, $k \sqsubseteq l$. Antag $l \sqsubseteq t$ og $t \Vdash \varphi_1$, heraf følger det, at $k \sqsubseteq t$, så $t \Vdash \varphi_2$. Vi får altså $l \Vdash \varphi_1 \rightarrow \varphi_2$.

□

Vi er nu i stand til at vise det følgende resultat.

Sætning 3.10 (Sundhedssætningen [vD08])**.** *$\Gamma \vdash_i \varphi \Rightarrow \Gamma \Vdash \varphi$.*

Bevis. Induktion på højden af $\mathcal{D}_\varphi$ med hypoteser Γ. Kripke-modellen $\mathcal{K} = (K, \Sigma)$ er valgt fast i dette bevis.

Basisskridt $h(\mathcal{D}_\varphi) = 0$, vi har at derivationstræet kun består af en enkel knude φ. Hvis $\Gamma \vdash_i \varphi$, så må vi have at $\varphi \in \Gamma$ og så det er klart at hvis $k \Vdash \Gamma$, så $k \Vdash \varphi$ for alle $k \in K$.

Induktionsskridt $h(\mathcal{D}_\varphi) > 0$, vi har anvendelse af en bevisregel. Vi skal i hvert tilfælde vise, at k forcer konklusionen i bevisreglen.

$(\wedge I)$. Antag $k \Vdash \Gamma \Rightarrow k \Vdash \varphi_1 \wedge k \Vdash \varphi_2$ for alle $k \in K$. Vælg et k således at $k \Vdash \Gamma$, så får vi ifølge induktionsantagelsen at $k \Vdash \varphi_1$ og $k \Vdash \varphi_2$, så $k \Vdash \varphi_1 \wedge \varphi_2$.

$(\wedge E)$. Antag $k \Vdash \Gamma \Rightarrow k \Vdash \varphi_1 \wedge \varphi_2$ for alle $k \in K$. Vælg k således at $k \Vdash \Gamma$, så følger det at $k \Vdash \varphi_1 \wedge \varphi_2$, hvilket gælder hvis og kun hvis $k \Vdash \varphi_1$ og $k \Vdash \varphi_2$.

$(\vee I)$ Antag $k \Vdash \Gamma \Rightarrow k \Vdash \varphi_1 \vee k \Vdash \varphi_2$ for alle $k \in K$. Vælg et k således at $k \Vdash \Gamma$, så får vi ifølge induktionsantagelsen at $k \Vdash \varphi_1$ eller $k \Vdash \varphi_2$, således $k \Vdash \varphi_1 \vee \varphi_2$.

($\vee E$). Antag $k \Vdash \Gamma \Rightarrow k \Vdash \varphi_1 \vee \varphi_2$ for alle $k \in K$, samt $k \Vdash \Gamma, \varphi_1 \Rightarrow k \Vdash \varphi_3$ for alle $k \in K$ og $k \Vdash \Gamma, \varphi_2 \Rightarrow k \Vdash \varphi_3$ for alle $k \in K$. Lad $k \Vdash \Gamma$, så giver induktionsantagelsen os, at $k \Vdash \varphi_1 \vee \varphi_2$ således at $k \Vdash \varphi_1$ eller $k \Vdash \varphi_2$. I det første tilfælde har vi $k \Vdash \Gamma, \varphi_1$ således at $k \Vdash \varphi_3$. I det andet tilfælde har vi $k \Vdash \Gamma, \varphi_2$ således at $k \Vdash \varphi_3$. I begge tilfælde har vi $k \Vdash \varphi_3$.

($\rightarrow I$). Antag $k \Vdash \Gamma, \varphi_1 \Rightarrow k \Vdash \varphi_2$ for alle $k \in K$. Lad $k \Vdash \Gamma$, $k \sqsubseteq l$ og $l \Vdash \varphi_1$. Vi ønsker at vise, at $k \Vdash \varphi_1 \rightarrow \varphi_2$. Det følger af monotoniciteten for $\Vdash$, at $l \Vdash \Gamma$. Induktionsantagelsen giver os, at $l \Vdash \varphi_2$. Derfor har vi for alle $l \sqsupseteq k$ at $l \Vdash \varphi_1 \Rightarrow l \Vdash \varphi_2$, således $k \Vdash \varphi_1 \rightarrow \varphi_2$.

($\rightarrow E$). Antag $k \Vdash \Gamma \Rightarrow k \Vdash \varphi_1$ og $k \Vdash \Gamma \Rightarrow k \Vdash \varphi_1 \rightarrow \varphi_2$ for alle $k \in K$. Vælg k således at $k \Vdash \Gamma$. Det følger af induktionsantagelsen, at for alle $l \sqsupseteq k$ gælder $l \Vdash \varphi_1 \Rightarrow l \Vdash \varphi_2$ og $l \Vdash \varphi_1$. Da $k \sqsubseteq k$ så har vi $k \Vdash \varphi_1 \Rightarrow k \Vdash \varphi_2$, heraf $k \Vdash \varphi_2$.

($\perp$) Antag $k \Vdash \Gamma \Rightarrow k \Vdash \perp$ for alle $k \in K$. Semantisk kan det ikke lade sig gøre, at k forcer Γ, da vi så vil have at $\perp \in \Sigma(k)$, hvilket strider mod definitionen af vores Kripke-model $\mathcal{K}$. Derfor må $k \Vdash \Gamma \Rightarrow k \Vdash \varphi$ for alle $k \in K$ være korrekt.

□

3.5.2 Klassisk bevis for fuldstændighed

Vi vil i dette afsnit give et klassisk, ikke-konstruktivt bevis for fuldstændighed af intuitionistisk udsagnslogik. I kapitel 4 giver vi et konstruktivt bevis for fuldstændighed af intuitionistisk udsagnslogik. Et væsentligt led i at vise fuldstændighed for den intuitionistiske udsagnslogik er at konstruere en mængde af udsagn, som er lukket under deduktion og som opfylder kravet om, at der til enhver disjunktion i mængden skal være mindst ét af elementerne i disjunktionen, som også er indeholdt i mængden. En sådan mængde kaldes en primisk teori.

Definition 3.11 (Primisk teori [vD08]). En mængde af sætninger Γ kaldes en *primisk teori* hvis følgende er opfyldt:

(a) Γ er lukket under $\vdash_i$, dvs. at hvis $\Gamma \vdash_i \varphi$, så $\varphi \in \Gamma$.

(b) Hvis $\varphi_1 \vee \varphi_2 \in \Gamma$, så er $\varphi_1 \in \Gamma$ eller $\varphi_2 \in \Gamma$.

Et lignende begreb findes for klassisk logik, hvor det kaldes en konsistent teori. En konsistent teori i klassisk logik er en teori, hvor det ikke er muligt at bevise en absurditet $\perp$. [HP02]

For at bevise fuldstændighed af intuitionistisk udsagnslogik vil vi først konstruere en maksimal primisk teori, der opfylder at $\Gamma \nvdash_i \varphi$. Ud fra denne vil vi konstruere en *modmodel* for $\Gamma \vdash_i \varphi$, dvs. en Kripke-model $\mathcal{K}$, som opfylder at $k \Vdash \Gamma$ og $k \nVdash \varphi$ for et $k \in K$. Denne modmodel benyttes som modstrid i beviset for fuldstændighed.

Det er værd at bemærke, at begge de følgende lemmaer stort set kun benytter sig af konstruktive argumenter. Der er nogle enkelte modstridsargumenter i lemmaernes beviser, men det er alligevel iøjnefaldende at lemmaerne, der leder op til fuldstændighedssætningen i høj grad er konstruktive, men at selve fuldstændighedssætningen alligevel bevises ved modstrid.

Lemma 3.12 ([vD08])**.** *Hvis $\Gamma \nvdash_i \varphi$, så er det muligt at udvide Γ til en primisk teori Γ' som opfylder at $\Gamma' \nvdash_i \varphi$.*

Bevis. Vi konstruerer en primisk teori Γ' trinvist ved at konstruere en følge af mængder $\Gamma_0 \subseteq \Gamma_1 \subseteq \Gamma_2 \subseteq \cdots \subseteq \Gamma'$, hvor $\Gamma_0 \stackrel{\text{def}}{=} \Gamma$. Betragt Γ_k hvor $k = 0, 1, 2, \dots$ og $\Gamma_k \nvdash_i \varphi$. Lad $\varphi_1 \vee \varphi_2$ være den første disjunktive sammensatte sætning, som vi ikke har gennemgået endnu, og hvorom det gælder at $\Gamma_k \vdash_i \varphi_1 \vee \varphi_2$. Da $\Gamma_k \nvdash_i \varphi$, kan vi ikke både have $\Gamma_k \cup \{\varphi_1\} \vdash_i \varphi$ og $\Gamma_k \cup \{\varphi_2\} \vdash_i \varphi$, for ellers gælder der ifølge ($\wedge$E)-reglen at $\Gamma_k \vdash_i \varphi$, hvilket er i strid med lemmaets antagelse. Derfor må der enten gælde at $\Gamma_k \cup \{\varphi_1\} \nvdash_i \varphi$ eller $\Gamma_k \cup \{\varphi_2\} \nvdash_i \varphi$. Lad nu

$$\Gamma_{k+1} \stackrel{\text{def}}{=} \begin{cases} \Gamma_k \cup \{\varphi_1\} & \text{hvis } \Gamma_k \cup \{\varphi_1\} \nvdash_i \varphi \\ \Gamma_k \cup \{\varphi_2\} & \text{ellers} \end{cases}$$

og $\Gamma' \stackrel{\text{def}}{=} \bigcup_{k \geq 0} \Gamma_k$, hvilket er vores konstruktion af den primiske teori Γ'.

Vi vil nu bevise, at Γ' opfylder kriterierne for en primisk teori.

(b) Betragt $\varphi_1 \vee \varphi_2 \in \Gamma'$ og lad k være det mindste tal, der opfylder at $\Gamma_k \vdash_i \varphi_1 \vee \varphi_2$. $\varphi_1 \vee \varphi_2$ kan ikke være gennemgået af vores konstruktionsalgoritme endnu, da k er det mindste tal, som opfylder $\Gamma_k \vdash_i \varphi_1 \vee \varphi_2$. Desuden ved vi at $\Gamma_h \vdash_i \varphi_1 \vee \varphi_2$ for alle $h \geq k$. Vores konstruktionsalgoritme vil for et eller andet $h \geq k$ gennemgå $\varphi_1 \vee \varphi_2$, og vi får derfor enten at $\varphi_1 \in \Gamma_{h+1}$ eller $\varphi_2 \in \Gamma_{h+1}$, og da $\Gamma_{h+1} \subseteq \Gamma'$, følger det at $\varphi_1 \in \Gamma'$ eller $\varphi_2 \in \Gamma'$.

(a) Lad $\Gamma' \vdash_i \varphi$. Så gælder der at $\Gamma' \vdash_i \varphi \vee \varphi$, og så ved vi fra (b), at $\varphi \in \Gamma'$.

Nu mangler vi kun at vise at $\Gamma' \nvdash_i \varphi$. Beviset er ved induktion efter k.

Basisskridt For $k = 0$ får vi fra lemmaets antagelse og fra definitionen af Γ_0, at $\Gamma_0 \nvdash_i \varphi$.

Induktionsskridt For $k > 0$. Antag at $\Gamma_k \nvdash_i \varphi$. Da vores konstruktionsalgoritme altid vælger et φ_j, hvor $j \in \{1, 2\}$ således at $\Gamma_k \cup \varphi_j \nvdash_i \varphi$, følger det at $\Gamma_{k+1} \nvdash_i \varphi$.

Antag nu at $\Gamma' \vdash_i \varphi$. Så vil der gælde at $\Gamma_k \vdash_i \varphi$ for et eller andet k, men dette er en modstrid med at vi lige har vist at $\Gamma_k \nvdash_i \varphi$ for alle k, så $\Gamma' \nvdash_i \varphi$. $\square$

Lemma 3.13 (Modeleksistenslemma [vD08])**.** *Hvis $\Gamma \nvdash_i \varphi$, så eksisterer der en Kripke-model $\mathcal{K} = (K, \Sigma)$ så $k \Vdash_{\mathcal{K}} \Gamma$ og $k \nVdash_{\mathcal{K}} \varphi$ for et $k \in K$.*

Bevis. Brug først lemma 3.12 til at udvide Γ til en primisk teori Γ' så $\Gamma' \nvdash_i \varphi$, og lad derefter $\mathcal{P}(\emptyset) = \{p_0, p_1, \ldots\}$, så $\mathcal{P}(\emptyset)_I = \{p_i \mid i \in I\}$, hvor $I \subseteq \mathbb{N}$. Lad

$$\Gamma_I \stackrel{\text{def}}{=} \{\varphi \mid \varphi \in \Gamma' \text{ og } \varphi \text{ er kun konstrueret af atomare formler fra } \mathcal{P}(\emptyset)_I\}$$

for alle $I \subseteq \mathbb{N}$, og $\Gamma_{I_1} \sqsubseteq \Gamma_{I_2}$ hvis $I_1 \subseteq I_2$. Så er $K = (\Gamma_I, \sqsubseteq)$ en partielt ordnet mængde. Γ_I er desuden en primisk teori. Betingelse (a) for en primisk teori er opfyldt, da Γ' er en primisk teori, så vi ved at hvis $\Gamma' \vdash_i \varphi$, så er $\varphi \in \Gamma'$. Så hvis $\Gamma_I \vdash_i \varphi$, så gælder der at $\varphi \in \Gamma'$ og φ må kun være konstrueret af atomare formler fra $\mathcal{P}(\emptyset)_I$, for ellers kan den ikke udledes fra Γ_I, så $\varphi \in \Gamma_I$. Betingelse (b) for en primisk teori er opfyldt, da den er opfyldt for Γ', og hvis $\varphi_1 \vee \varphi_2$ kun er konstrueret ud fra atomare formler fra $\mathcal{P}(\emptyset)_I$, så må φ_1 og φ_2 også kun være konstrueret ud fra atomare formler fra $\mathcal{P}(\emptyset)_I$. Så hvis $\varphi_1 \vee \varphi_2 \in \Gamma_I$, så er $\varphi_1 \in \Gamma_I$ og $\varphi_2 \in \Gamma_I$.

Lad $\Sigma(\Gamma_I) \stackrel{\text{def}}{=} \{p \mid p \in \Gamma_I\}$. Så gælder der at

(a) $\Sigma(\Gamma_I) \subseteq \mathcal{P}(\emptyset)_I$ for alle $\Gamma_I \in K$.

(b) $\perp \notin \Sigma(\Gamma_I)$ for alle $\Gamma_I \in K$.

(c) $\Gamma_{I_1} \sqsubseteq \Gamma_{I_2} \Rightarrow \Sigma(\Gamma_{I_1}) \subseteq \Sigma(\Gamma_{I_2})$ for alle $\Gamma_{I_1}, \Gamma_{I_2} \in K$.

$\mathcal{K} \stackrel{\text{def}}{=} (K, \Sigma)$ er derfor en Kripke-model.

Vi vil nu vise at $\Gamma_I \Vdash \varphi_0 \Leftrightarrow \Gamma_I \vdash_i \varphi_0$ ved strukturel induktion på φ_0.

Basisskridt Atomare φ_0.

- **(⇒)** Hvis $\Gamma_I \Vdash \varphi_0$, så betyder det at $\varphi_0 \in \Sigma(\Gamma_I) = \{p \mid p \in \Gamma_I\}$, så $\varphi_0 \in \Gamma_I$. Det gælder derfor, at $\Gamma_I \vdash_i \varphi_0$.
- **(⇐)** Dette følger af sundhed (sætning 3.10).

Rekursionsskridt Sammensatte φ_0.

$\varphi_0 = \varphi_1 \wedge \varphi_2$. Antag $\Gamma_I \Vdash \varphi_1 \Leftrightarrow \Gamma_I \vdash_i \varphi_1$ og $\Gamma_I \Vdash \varphi_2 \Leftrightarrow \Gamma_I \vdash_i \varphi_2$.

- **(⇒)** $\Gamma_I \Vdash \varphi_1 \wedge \varphi_2 \Leftrightarrow \Gamma_I \Vdash \varphi_1$ og $\Gamma_I \Vdash \varphi_2$ per definition 3.8. Det følger derfor af induktionsantagelsen, at $\Gamma_I \vdash_i \varphi_1$ og $\Gamma_I \vdash_i \varphi_2$. Dette betyder, at $\Gamma_I \vdash_i \varphi_1 \wedge \varphi_2$ ifølge ($\wedge$I)-reglen.

($\Leftarrow$) $\Gamma_I \vdash_i \varphi_1 \wedge \varphi_2 \Rightarrow \Gamma_I \vdash_i \varphi_1$ og $\Gamma_I \vdash_i \varphi_2$. Induktionsantagelsen giver derfor at $\Gamma_I \Vdash \varphi_1$ og $\Gamma_I \Vdash \varphi_2$ hvilket gælder hvis og kun hvis $\Gamma_I \Vdash \varphi_1 \wedge \varphi_2$.

$\varphi_0 = \varphi_1 \vee \varphi_2$. Antag $\Gamma_I \Vdash \varphi_1 \Leftrightarrow \Gamma_I \vdash_i \varphi_1$ eller $\Gamma_I \Vdash \varphi_2 \Leftrightarrow \Gamma_I \vdash_i \varphi_2$.

($\Rightarrow$) $\Gamma_I \Vdash \varphi_1 \vee \varphi_2 \Leftrightarrow \Gamma_I \Vdash \varphi_1$ eller $\Gamma_I \Vdash \varphi_2$. Induktionsantagelsen giver da, at $\Gamma_I \vdash_i \varphi_1$ eller $\Gamma_I \vdash_i \varphi_2$, hvilket medfører, at $\Gamma_I \vdash_i \varphi_1 \vee \varphi_2$ ifølge ($\vee$I)-reglen.

($\Leftarrow$) Da Γ_I er en primisk teori, får vi fra egenskab (b) af en primisk teori, at $\Gamma_I \vdash_i \varphi_1 \vee \varphi_2 \Rightarrow \Gamma_I \vdash_i \varphi_1$ eller $\Gamma_I \vdash_i \varphi_2$. Induktionsantagelsen giver da, at $\Gamma_I \Vdash \varphi_1$ eller $\Vdash \varphi_2 \Leftrightarrow \Gamma_I \Vdash \varphi_1 \vee \varphi_2$ ifølge definition 3.8.

$\varphi_0 = \varphi_1 \to \varphi_2$. Antag $\Gamma_I \Vdash \varphi_1 \Leftrightarrow \Gamma_I \vdash_i \varphi_1$ og $\Gamma_{I'} \Vdash \varphi_2 \Leftrightarrow \Gamma_{I'} \vdash_i \varphi_2$, hvor $\Gamma_I \subseteq \Gamma_{I'}$.

($\Rightarrow$) Antag, at $\Gamma_I \nvdash_i \varphi_1 \to \varphi_2$. Så gælder der, at $\Gamma_I \cup \{\varphi_1\} \nvdash_i \varphi_2$. Ud fra den måde, vi har konstrueret mængden Γ_I på, kan vi udvide den til en mængde Γ_I'', som opfylder at $\Gamma_I \cup \{\varphi_1\} \subseteq \Gamma_I''$ og $\Gamma_I'' \nvdash_i \varphi_2$. Der gælder, at $\Gamma_I'' \vdash_i \varphi_1$, så induktionsantagelsen giver, at $\Gamma_I'' \Vdash \varphi_1$. Da $\Gamma_I \Vdash \varphi_1 \to \varphi_2$ og $\Gamma_I \subseteq \Gamma_I''$, får vi at $\Gamma_I'' \Vdash \varphi_2$. Induktionsantagelsen giver derfor, at $\Gamma_I'' \vdash_i \varphi_2$. Vi har altså både at $\Gamma_I'' \nvdash_i \varphi_2$ og $\Gamma_I'' \vdash_i \varphi_2$, hvilket er en modstrid, og der må derfor gælde at $\Gamma_I \vdash_i \varphi_1 \to \varphi_2$.

($\Leftarrow$) Antag, at $\Gamma_I \vdash_i \varphi_1 \to \varphi_2$. Hvis $\Gamma_I \subseteq \Gamma_{I'}$ og $\Gamma_{I'} \Vdash \varphi_1$, så får vi per induktionsantagelsen at $\Gamma_{I'} \vdash_i \varphi_1$. Da $\Gamma_I \subseteq \Gamma_{I'}$, får vi at $\Gamma_{I'} \vdash_i \varphi_1 \to \varphi_2$ og ($\to$E)-reglen giver derfor at $\Gamma_{I'} \vdash_i \varphi_2$. Induktionsantagelsen giver da at $\Gamma_{I'} \Vdash \varphi_2$. Vi kan derfor konkludere at $\Gamma_I \Vdash \varphi_1 \to \varphi_2$.

Vi har altså vist, at $\Gamma_I \Vdash \varphi_0 \Leftrightarrow \Gamma_I \vdash_i \varphi_0$.

Lad $k \overset{\text{def}}{=} \Gamma'$. Γ' er konstrueret så $\Gamma' \nvdash_i \varphi$, og vi får derfor at $k \nVdash \varphi$. Da $\Gamma \subseteq \Gamma'$, gælder der at $\Gamma' \vdash_i \Gamma$, så $k \Vdash \Gamma$, og da $k \in K$ er beviset færdigt. $\square$

Vi er nu klar til at formulere og vise fuldstændighedssætningen for intuitionistisk logik.

Sætning 3.14 (Fuldstændighed [vD08]). $\Gamma \Vdash \varphi \Rightarrow \Gamma \vdash_i \varphi$.

Bevis. Beviset er ved modstrid. Lad $\Gamma \Vdash \varphi$ og antag $\Gamma \nvdash_i \varphi$. Da eksisterer der ifølge lemma 3.13 en Kripke-model $\mathcal{K}$ hvor $k \nVdash \varphi$ for et $k \in K$. Dette betyder, at $\Gamma \nVdash \varphi$, hvilket er i modstrid med forudsætningen at $\Gamma \Vdash \varphi$. Altså gælder der at $\Gamma \vdash_i \varphi$. $\square$

Hermed er fuldstændighed for intuitionistisk udsagnslogik vist, men da vi i sætning 3.14 har brugt et modstridsargument, er der altså tale om et ikke-konstruktivt bevis. Vi vil derfor som det næste give os i kast med at give et konstruktivt bevis for fuldstændighedssætningen.

KAPITEL 4

Konstruktivt bevis for intuitionistisk fuldstændighed

"God made the natural numbers, all the rest is the work of man."
— Leopold Kronecker (1823-1891) [Li13]

Vi vil i dette kapitel give et konstruktivt bevis for fuldstændighed af intuitionistisk udsagnslogik. Vores bevis tager udgangspunkt i [Und90]. Før vi giver selve beviset, vil vi indføre nogle begreber som bliver brugt i beviset.

4.1 Sekventer og semantiske tableauer

Vi har brug for de følgende to definitioner.

Definition 4.1 (Sekvent [Und90])**.** En *sekvent* er et par $\langle \Gamma; \Delta \rangle$, hvor Γ og Δ er mængder af udsagn.

Definition 4.2 (System [Und90])**.** Et *system* $\mathcal{S}$ er en mængde af sekventer.

Et semantisk tableau er en træstruktur, der er opbygget ud fra nogle bestemte tableauregler. Tableausystemer er målorienterede, dvs. hver gang en tableauregel anvendes, kommer vi nærmere målet for tableauet. For vores tableauregler er målet at være lukket under deludsagn, og vores tableauregler tilføjer derfor nye deludsagn af de allerede eksisterende udsagn, der bringer os tættere på målet om at være lukket under deludsagn. En enkelt af vores tableauregler tilføjer også en ny sekvent, men denne sekvent kan aldrig være en delmængde af en allerede eksisterende sekvent, og den nye sekvent vil bestå af deludsagn af allerede eksisterende udsagn. Vi vil altså efter endeligt mange trin have opnået målet om lukning under deludsagn ved brug af vores tableauregler.

$$(T\wedge) \qquad \frac{\langle \Gamma; \Delta\rangle \cup \mathcal{S}}{\langle \Gamma \cup \{\varphi_1, \varphi_2\}; \Delta\rangle \cup \mathcal{S}}$$

hvis $\varphi_1 \wedge \varphi_2 \in \Gamma$ og ($\varphi_1 \notin \Gamma$ eller $\varphi_2 \notin \Gamma$)

$$(F\wedge) \qquad \frac{\langle \Gamma; \Delta\rangle \cup \mathcal{S}}{\langle \Gamma; \Delta \cup \{\varphi_1\}\rangle \cup \mathcal{S} \qquad \langle \Gamma; \Delta \cup \{\varphi_2\}\rangle \cup \mathcal{S}}$$

hvis $\varphi_1 \wedge \varphi_2 \in \Delta$ og ($\varphi_1 \notin \Delta$ og $\varphi_2 \notin \Delta$)

$$(T\vee) \qquad \frac{\langle \Gamma; \Delta\rangle \cup \mathcal{S}}{\langle \Gamma \cup \{\varphi_1\}; \Delta\rangle \cup \mathcal{S} \qquad \langle \Gamma \cup \{\varphi_2\}; \Delta\rangle \cup \mathcal{S}}$$

hvis $\varphi_1 \vee \varphi_2 \in \Gamma$ og ($\varphi_1 \notin \Gamma$ og $\varphi_2 \notin \Gamma$)

$$(F\vee) \qquad \frac{\langle \Gamma; \Delta\rangle \cup \mathcal{S}}{\langle \Gamma; \Delta \cup \{\varphi_1, \varphi_2\}\rangle \cup \mathcal{S}}$$

hvis $\varphi_1 \vee \varphi_2 \in \Delta$ og ($\varphi_1 \notin \Delta$ eller $\varphi_2 \notin \Delta$)

$$(T\rightarrow) \qquad \frac{\langle \Gamma; \Delta\rangle \cup \mathcal{S}}{\langle \Gamma; \Delta \cup \{\varphi_1\}\rangle \cup \mathcal{S} \qquad \langle \Gamma \cup \{\varphi_2\}; \Delta\rangle \cup \mathcal{S}}$$

hvis $\varphi_1 \rightarrow \varphi_2 \in \Gamma$ og ($\varphi_1 \notin \Delta$ eller $\varphi_2 \notin \Gamma$)

$$(F\rightarrow) \qquad \frac{\langle \Gamma; \Delta\rangle \cup \mathcal{S}}{\langle \Gamma; \Delta\rangle \cup \mathcal{S} \cup \langle \Gamma \cup \{\varphi_1\}; \{\varphi_2\}\rangle}$$

hvis $\varphi_1 \rightarrow \varphi_2 \in \Delta$ og for alle sekventer $\langle \Gamma'; \Delta'\rangle \in \mathcal{S}$ der gælder at ($\Gamma \cup \{\varphi_1\} \not\subseteq \Gamma'$ og $\varphi_2 \notin \Delta'$)

Tabel 4.1: Tableauregler for intuitionistisk udsagnslogik [Fit69].

Tableaureglerne vi har brug for til beviset er givet i tabel 4.1, og er inspireret af [Fit69]. Der kan opstilles flere tableauregler end disse, men da vi ikke har brug for andre i beviset, undlader vi at beskrive flere.

Dette giver anledning til følgende definition.

Definition 4.3 (Semantisk tableau)**.** Et *semantisk tableau* er en træstruktur, der er opbygget via tableaureglerne i tabel 4.1.

En knude $\mathcal{S}$, for hvilken ingen af reglerne i tabel 4.1 kan anvendes, kaldes et blad.

Definition 4.4 (Blad)**.** Et *blad* i et semantisk tableau er en knude $\mathcal{S}$, hvor alle sekventerne i knuden er lukket under deludsagn, dvs. at der for alle sekventer $\langle \Gamma; \Delta \rangle \in \mathcal{S}$ gælder at

- hvis $\varphi_1 \wedge \varphi_2 \in \Gamma$ så gælder at $\varphi_1 \in \Gamma$ og $\varphi_2 \in \Gamma$,
- hvis $\varphi_1 \vee \varphi_2 \in \Gamma$ så gælder at $\varphi_1 \in \Gamma$ eller $\varphi_2 \in \Gamma$,
- hvis $\varphi_1 \to \varphi_2 \in \Gamma$ så gælder at $\varphi_1 \in \Delta$ eller $\varphi_2 \in \Gamma$,
- hvis $\varphi_1 \wedge \varphi_2 \in \Delta$ så gælder at $\varphi_1 \in \Delta$ eller $\varphi_2 \in \Delta$,
- hvis $\varphi_1 \vee \varphi_2 \in \Delta$ så gælder at $\varphi_1 \in \Delta$ og $\varphi_2 \in \Delta$,
- hvis $\varphi_1 \to \varphi_2 \in \Delta$ så gælder at der findes en sekvent $\langle \Gamma'; \Delta' \rangle \in \mathcal{S}$ således at $\Gamma \subseteq \Gamma'$ og $\varphi_1 \in \Gamma'$ og $\varphi_2 \in \Delta'$.

Hvis et blad opfylder at $\Gamma \cap \Delta \neq \emptyset$ for en sekvent $\langle \Gamma; \Delta \rangle$ i bladet, så siger vi at bladet er *lukket*, og hvis bladet ikke opfylder dette, siger vi at bladet er *åbent* [Und90]. Hvis alle blade i et semantisk tableau er lukkede, så siger vi at tableauet er *vellykket*.

Definition 4.5 (Gyldighed)**.** Vi siger at en sekvent $\langle \Gamma; \Delta \rangle$ er *gyldig* hvis der for alle Kripke-modeller $\mathcal{K}$ gælder at hvis $k \Vdash \Gamma$ for alle $k \in K$, så medfører det at der eksisterer et $\varphi \in \Delta$ så $k \Vdash \varphi$ for alle $k \in K$. Hvis $\langle \Gamma; \Delta \rangle$ er gyldig, skriver vi $\Gamma \Vdash \Delta$.

4.2 Det tredelte konstruktive bevise

Vi vil i dette afsnit give et konstruktivt bevis for fuldstændighed af intuitionstisk udsagnslogik. Beviset er delt op i tre delbeviser, som tilsammen beviser fuldstændigheden.

Ideen er at vise at man ud fra en gyldig sekvent kan konstruere et vellykket tableau, samt at man ud fra ethvert vellykket tableau kan lave et bevis for en sekvent i roden af tableauet. Hvis algoritmen i beviset køres på et enkelt udsagn φ med præmisser Γ, vil roden i tableauet kun bestå af sekventen $\langle \Gamma; \varphi \rangle$, så hvis der er tale om et vellykket tableau, vil vi kunne konkludere, at $\Gamma \vdash_i \varphi$, som ønsket.

Sætning 4.6.

(i) Hvis $\Gamma \Vdash \Delta$, da findes der et vellykket tableau med rod $\langle \Gamma; \Delta \rangle$.

(ii) For ethvert åbent blad $\mathcal{S}$ i et tableau gælder det for enhver sekvent $\langle \Gamma; \Delta \rangle \in \mathcal{S}$ at man ud fra $\mathcal{S}$ kan konstruere en modmodel for $\langle \Gamma; \Delta \rangle$.

Bevis. (i) Dette følger af sætning 4.7.

(ii) Vi viser først hvordan man ud fra et vilkårligt blad konstruerer en Kripke-model, hvor punkterne i Kripke-modellen er sekventerne i bladet, og konstruerer derefter ud fra denne en modmodel for et vilkårligt åbent blad.

Lad $\mathcal{S}$ være et vilkårligt blad i et tableau, og lad relationen $\sqsubseteq$ være givet ved

$$\langle \Gamma; \Delta \rangle \sqsubseteq \langle \Gamma'; \Delta' \rangle \Leftrightarrow \Gamma \subseteq \Gamma'.$$

Så er $\sqsubseteq$ en partiel ordning, da $\subseteq$ er en partiel ordning, og vi får derfor at $K \stackrel{\text{def}}{=} (\mathcal{S}, \sqsubseteq)$ er en partielt ordnet mængde.

Lad nu $\Sigma(\langle \Gamma; \Delta \rangle) \stackrel{\text{def}}{=} \{p \mid p \in \Gamma \text{ og } p \in \mathcal{P}(\emptyset)\}$. Da gælder, at

(a) $\Sigma(k) \subseteq \mathcal{P}(\emptyset)$ for alle $k \in K$,

(b) $\perp \notin \Sigma(k)$ for alle $k \in K$ og

(c) hvis $k \sqsubseteq l$, så gælder at $\Sigma(k) \subseteq \Sigma(l)$ for alle $l, k \in K$.

$\mathcal{K} \stackrel{\text{def}}{=} (K, \Sigma)$ er altså en Kripke-model. Betragt nu et vilkårligt åbent blad $\mathcal{S}$. Vi vil konstruere en modmodel for en vilkårlig sekvent $\langle \Gamma; \Delta \rangle \in \mathcal{S}$.

$\mathcal{S}$ er et blad, og vi ved derfor at alle sekventer i $\mathcal{S}$ er lukket under deludsagn, så enhver sekvent har egenskaberne givet i definition 4.4.

Vi skal vise, at $k \Vdash \Gamma$ og $k \nVdash \Delta$ for et $k \in K$. Vi viser først, at $k \Vdash \varphi$ for alle $\varphi \in \Gamma$ for et $k \in K$ ved strukturel induktion på et vilkårligt udsagn φ.

Basisskridt φ er atomar, dvs. $\varphi \in \mathcal{P}(\emptyset)$. Hvis $\varphi \in \Gamma$, får vi at $\varphi \in \Sigma(\langle \Gamma; \Delta \rangle)$, så $\langle \Gamma; \Delta \rangle \Vdash \varphi$.

Rekursionsskridt φ er et sammensat udsagn.

$\varphi = \varphi_1 \wedge \varphi_2$. Antag at hvis $\varphi_1 \in \Gamma$ og $\varphi_2 \in \Gamma$, så gælder at $\langle \Gamma; \Delta \rangle \Vdash \varphi_1$ og $\langle \Gamma; \Delta \rangle \Vdash \varphi_2$. Vi ved at $\varphi_1 \in \Gamma$ og $\varphi_2 \in \Gamma$, så induktionsantagelsen giver at $\langle \Gamma; \Delta \rangle \Vdash \varphi_1 \wedge \varphi_2$.

$\varphi = \varphi_1 \vee \varphi_2$. Antag at hvis $\varphi_1 \in \Gamma$ eller $\varphi_2 \in \Gamma$, så gælder at $\langle \Gamma; \Delta \rangle \Vdash \varphi_1$ eller $\langle \Gamma; \Delta \rangle \Vdash \varphi_2$. Vi ved at $\varphi_1 \in \Gamma$ eller $\varphi_2 \in \Gamma$, så induktionsantagelsen giver at $\langle \Gamma; \Delta \rangle \Vdash \varphi_1 \vee \varphi_2$.

$\varphi = \varphi_1 \to \varphi_2$. Antag at hvis $\varphi_1 \in \Delta'$ eller $\varphi_2 \in \Gamma'$, hvor $\langle \Gamma; \Delta \rangle \sqsubseteq \langle \Gamma'; \Delta' \rangle$, så gælder at $\langle \Gamma'; \Delta' \rangle \nVdash \varphi_1$ eller $\langle \Gamma'; \Delta' \rangle \Vdash \varphi_2$. Vi skal vise at for alle sekventer $\langle \Gamma'; \Delta' \rangle$, hvor $\langle \Gamma; \Delta \rangle \sqsubseteq \langle \Gamma'; \Delta' \rangle$, gælder der at $\langle \Gamma'; \Delta' \rangle \nVdash \varphi_1$ eller $\langle \Gamma'; \Delta' \rangle \Vdash \varphi_2$. Vi ved at $\langle \Gamma; \Delta \rangle \sqsubseteq \langle \Gamma'; \Delta' \rangle$ hvis og kun hvis $\Gamma \subseteq \Gamma'$, så $\varphi_1 \to \varphi_2 \in \Gamma'$. Vi ved desuden at $\varphi_1 \in \Delta'$ eller $\varphi_2 \in \Gamma'$. Induktionsantagelsen giver da, at $\langle \Gamma'; \Delta' \rangle \nVdash \varphi_1$ eller $\langle \Gamma'; \Delta' \rangle \Vdash \varphi_2$, så $\langle \Gamma'; \Delta' \rangle \Vdash$

$\varphi_1 \to \varphi_2$. Refleksiviteten af $\sqsubseteq$ giver at $\langle \Gamma; \Delta \rangle \sqsubseteq \langle \Gamma; \Delta \rangle$, så $\langle \Gamma; \Delta \rangle \Vdash \varphi_1 \to \varphi_2$.

Vi viser nu, at $k \nVdash \varphi$ for alle $\varphi \in \Delta$ for et $k \in K$ ved strukturel induktion på et vilkårligt udsagn φ.

Basisskridt φ er atomar. Bemærk, at da $\Gamma \cap \Delta = \emptyset$, medfører $\varphi \in \Delta$ at $\varphi \notin \Gamma$, så $\varphi \notin \Sigma(\langle \Gamma; \Delta \rangle)$, og vi får derfor at $\langle \Gamma; \Delta \rangle \nVdash \varphi$.

Induktionsskridt φ er et sammensat udsagn.

$\varphi = \varphi_1 \wedge \varphi_2$. Antag at hvis $\varphi_1 \in \Delta$ eller $\varphi_2 \in \Delta$, så gælder at $\langle \Gamma; \Delta \rangle \nVdash \varphi_1$ eller $\langle \Gamma; \Delta \rangle \nVdash \varphi_2$. Vi ved at $\varphi_1 \in \Delta$ eller $\varphi_2 \in \Delta$, så induktionsantagelsen giver at $\langle \Gamma; \Delta \rangle \nVdash \varphi_1 \wedge \varphi_2$.

$\varphi = \varphi_1 \vee \varphi_2$. Antag at hvis $\varphi_1 \in \Delta$ og $\varphi_2 \in \Delta$, så gælder at $\langle \Gamma; \Delta \rangle \nVdash \varphi_1$ og $\langle \Gamma; \Delta \rangle \nVdash \varphi_2$. Vi ved at $\varphi_1 \in \Delta$ og $\varphi_2 \in \Delta$, så induktionsantagelsen giver at $\langle \Gamma; \Delta \rangle \nVdash \varphi_1 \vee \varphi_2$.

$\varphi = \varphi_1 \to \varphi_2$. Antag at hvis $\varphi_1 \in \Gamma'$ og $\varphi_2 \in \Delta'$, hvor $\langle \Gamma; \Delta \rangle \sqsubseteq \langle \Gamma'; \Delta' \rangle$, så gælder der at $\langle \Gamma'; \Delta' \rangle \Vdash \varphi_1$ og $\langle \Gamma'; \Delta' \rangle \nVdash \varphi_2$. Vi skal vise, at der findes en sekvent $\langle \Gamma'; \Delta' \rangle$, hvor $\langle \Gamma; \Delta \rangle \sqsubseteq \langle \Gamma'; \Delta' \rangle$, hvorom det gælder at $\langle \Gamma'; \Delta' \rangle \Vdash \varphi_1$ og $\langle \Gamma'; \Delta' \rangle \nVdash \varphi_2$. Vi ved at der eksisterer en sekvent $\langle \Gamma'; \Delta' \rangle \in \mathcal{S}$ så $\Gamma \cup \{\varphi_1\} \subseteq \Gamma'$ og $\varphi_2 \in \Delta'$. Da $\Gamma \subseteq \Gamma \cup \{\varphi_1\} \subseteq \Gamma'$, får vi at $\langle \Gamma; \Delta \rangle \sqsubseteq \langle \Gamma'; \Delta' \rangle$. Vi kan altså konkludere, at $\langle \Gamma'; \Delta' \rangle \nVdash \varphi_1 \to \varphi_2$. Refleksiviteten af $\sqsubseteq$ giver at $\langle \Gamma; \Delta \rangle \sqsubseteq \langle \Gamma; \Delta \rangle$, så $\langle \Gamma; \Delta \rangle \nVdash \varphi_1 \to \varphi_2$.

□

Sætning 4.7. *Lad $\mathcal{S}$ være et system. Hvis det om en sekvent $\langle \Gamma; \Delta \rangle \in \mathcal{S}$ gælder at $\Gamma \Vdash \Delta$, så er der et vellykket tableau med $\mathcal{S}$ som rod.*

Strategien i beviset er at lave induktion efter antallet af manglende deludsagn i $\mathcal{S}$. De manglende deludsagn er mængden af de udsagn, der skal tilføjes for at alle sekventer i $\mathcal{S}$ er lukket under deludsagn. Algoritmen virker således at ved hver eneste regelanvendelse får vi mindst én ny knude i vores tableau, hvor det samlede antal manglende deludsagn er gået ned med mindst én. Når antallet af manglende deludsagn er nul, så er vi nået et blad og algoritmen terminerer.

Bevis for sætning 4.7. Beviset er ved induktion efter antallet af manglende deludsagn i $\mathcal{S}$. Lad $\Psi : \mathcal{S} \to \mathbb{N}$ være funktion, der returnerer antallet af manglende deludsagn i $\mathcal{S}$.

Basisskridt $\Psi(\mathcal{S}) = 0$, vi har at $\mathcal{S}$ er et blad. Vi har per antagelse at der findes en gyldig sekvent $\langle \Gamma'; \Delta' \rangle \in \mathcal{S}$. Så ved vi at der ikke eksisterer en modmodel for $\langle \Gamma'; \Delta' \rangle$, og derfor er bladet, ifølge sætning 4.6ii), ikke åbent. Vi må derfor have at bladet er lukket, og dermed $\Gamma \cap \Delta \neq \emptyset$.

Induktionsskridt $\Psi(\mathcal{S}) \geq 1$. Der er i alt seks mulige tilfælde. I hvert tilfælde er vores induktionsantagelse, at hvis der findes en gyldig sekvent i $\mathcal{S}'$, hvor $\mathcal{S}'$ er et system med færre antal manglende deludsagn, så findes der et vellykket tableau med $\mathcal{S}'$ som rod. Vi konstruerer dermed i hvert tilfælde et eller flere systemer, som hver især er rod i et vellykket tableau relativt til $\mathcal{S}$. For en vilkårlig sekvent, $\langle \Gamma; \Delta \rangle \in \mathcal{S}$, antag $\Gamma \Vdash \Delta$ og vælg det passende tilfælde:

Hvis $\varphi_1 \wedge \varphi_2 \in \Gamma$ og ($\varphi_1 \notin \Gamma$ eller $\varphi_2 \notin \Gamma$). Det følger af monotoniciteten for $\Vdash$, at $\Gamma \cup \{\varphi_1, \varphi_2\} \Vdash \Delta$. Vi konstruerer derfor $\mathcal{S}'$:

$$\mathcal{S}' = (\mathcal{S} \setminus \langle \Gamma; \Delta \rangle) \cup \langle \Gamma \cup \{\varphi_1, \varphi_2\}; \Delta \rangle.$$

Det følger af induktionsantagelsen at $\mathcal{S}'$ er rod i et vellykket tableau.

Hvis $\varphi_1 \wedge \varphi_2 \in \Delta$ og ($\varphi_1 \notin \Delta$ og $\varphi_2 \notin \Delta$). Jvf. semantikken for $\Vdash$, så må vi have at $\Gamma \Vdash \Delta \cup \{\varphi_1\}$ og $\Gamma \Vdash \Delta \cup \{\varphi_2\}$. Vi konstruerer de to systemer $\mathcal{S}_1$ og $\mathcal{S}_2$:

$$\mathcal{S}_1 = (\mathcal{S} \setminus \langle \Gamma; \Delta \rangle) \cup \langle \Gamma; \Delta \cup \{\varphi_1\} \rangle$$
$$\mathcal{S}_2 = (\mathcal{S} \setminus \langle \Gamma; \Delta \rangle) \cup \langle \Gamma; \Delta \cup \{\varphi_2\} \rangle.$$

Det følger af induktionsantagelsen, at $\mathcal{S}_1$ er rod i et vellykket tableau og $\mathcal{S}_2$ er rod i et vellykket tableau.

Hvis $\varphi_1 \vee \varphi_2 \in \Gamma$ og ($\varphi_1 \notin \Gamma$ og $\varphi_2 \notin \Gamma$). Det følger af monotoniciteten for $\Vdash$, at $\Gamma \cup \{\varphi_1\} \Vdash \Delta$ og $\Gamma \cup \{\varphi_2\} \Vdash \Delta$. Vi konstruerer derfor de to systemer $\mathcal{S}_1$ og $\mathcal{S}_2$:

$$\mathcal{S}_1 = (\mathcal{S} \setminus \langle \Gamma; \Delta \rangle) \cup \langle \Gamma \cup \{\varphi_1\}; \Delta \rangle$$
$$\mathcal{S}_2 = (\mathcal{S} \setminus \langle \Gamma; \Delta \rangle) \cup \langle \Gamma \cup \{\varphi_2\}; \Delta \rangle.$$

Induktionsantagelsen giver, at $\mathcal{S}_1$ er rod i et vellykket tableau og $\mathcal{S}_2$ er rod i et vellykket tableau.

Hvis $\varphi_1 \vee \varphi_2 \in \Delta$ og ($\varphi_1 \notin \Delta$ eller $\varphi_2 \notin \Delta$). Det følger af semantikken for $\Vdash$, at vi må have $\Gamma \Vdash \Delta \cup \{\varphi_1, \varphi_2\}$. Vi konstruerer systemet $\mathcal{S}'$:

$$\mathcal{S}' = (\mathcal{S} \setminus \langle \Gamma; \Delta \rangle) \cup \langle \Gamma; \Delta \cup \{\varphi_1, \varphi_2\} \rangle.$$

Det følger af induktionsantagelsen at $\mathcal{S}'$ er rod i et vellykket tableau.

Hvis $\varphi_1 \to \varphi_2 \in \Gamma$ og ($\varphi_1 \notin \Delta$ og $\varphi_2 \notin \Gamma$). Vi har at $\Gamma \cup \{\varphi_1 \to \varphi_2\} \Vdash \Delta$, så følger det af forcing-semantikken for implikation at $\Gamma \cup \{\varphi_2\} \Vdash \Delta \cup \{\varphi_1\}$. Derfor konstruerer vi de to systemer $\mathcal{S}_1$ og $\mathcal{S}_2$:

$$\mathcal{S}_1 = (\mathcal{S} \setminus \langle \Gamma; \Delta \rangle) \cup \langle \Gamma; \Delta \cup \{\varphi_1\} \rangle$$
$$\mathcal{S}_2 = (\mathcal{S} \setminus \langle \Gamma; \Delta \rangle) \cup \langle \Gamma \cup \{\varphi_2\}; \Delta \rangle.$$

Induktionsantagelsen giver, at $\mathcal{S}_1$ er rod i et vellykket tableau og $\mathcal{S}_2$ er rod i et vellykket tableau.

Hvis $\varphi_1 \to \varphi_2 \in \Delta$ og for alle sekventer $\langle \Gamma'; \Delta' \rangle \in \mathcal{S}$, der gælder at $(\Gamma \cup \{\varphi_1\} \not\subseteq \Gamma'$ eller $\varphi_2 \notin \Delta')$, og ingen af de foregående tilfælde gælder. I dette tilfælde mangler der ikke nogen deludsagn i en enkelt sekvent, men derimod mangler der en sekvent i systemet. Vi har at $\Gamma \Vdash \{\varphi_1 \to \varphi_2\}$, det gælder så at $\Gamma \cup \{\varphi_1\} \Vdash \varphi_2$. Herfra kan vi konstruere $\mathcal{S}'$:

$$\mathcal{S}' = \mathcal{S} \cup \{\langle \Gamma \cup \{\varphi_1\}; \{\varphi_2\} \rangle\}.$$

Det følger af induktionsantagelsen at $\mathcal{S}'$ er rod i et vellykket tableau.

□

Sætning 4.8. *Hvis $\mathcal{S}$ er rod i et vellykket tableau, da findes en sekvent $\langle \Gamma; \Delta \rangle \in \mathcal{S}$ så $\Gamma \vdash_i \Delta$.*

Bevis. Beviset er ved induktion i tableauets dybde.

Ideen er at vise, at, givet et vellykket tableau, vil det altid være muligt at finde en sekvent $\langle \Gamma; \Delta \rangle$ i ethvert blad så $\Gamma \vdash_i \Delta$. Dette er basisskridtet. Herefter vil vi arbejde os op af træet, og vise at for en vilkårlig knude $\mathcal{S}$ vil det gælde at der findes en sekvent $\langle \Gamma; \Delta \rangle \in \mathcal{S}$ så $\Gamma \vdash_i \Delta$ hvis det samme gør sig gældende får knuderne umiddelbart under $\mathcal{S}$. Dette er induktionsskridtet.

Basisskridt Ingen af sidebetingelserne for tableaureglerne er opfyldt, så vi er nået et blad $\mathcal{S}$. Vi ved, at $\Gamma \vdash_i \Delta$ hvis $\Gamma \cap \Delta \neq \emptyset$, og da tableauet er vellykket, har vi netop at $\Gamma \cap \Delta \neq \emptyset$ for en sekvent $\langle \Gamma; \Delta \rangle \in \mathcal{S}$, så mindst ét af elementerne i Δ kan bevises ud fra Γ, og vi får derfor at $\Gamma \vdash_i \Delta$.

Induktionsskridt Betragt en vilkårlig knude $\mathcal{S} \cup \langle \Gamma; \Delta \rangle$ i tableauet, som opfylder betingelserne for mindst én af tableaureglerne. Da må $\mathcal{S}$'s børn være konstrueret ud fra en af tableaureglerne.

(T$\wedge$) Hvis $\mathcal{S} \cup \langle \Gamma; \Delta \rangle$ har ét barn $\mathcal{S}'$, som opfylder at $\mathcal{S}' = \langle \Gamma \cup \{\varphi_1, \varphi_2\}; \Delta \rangle \cup \mathcal{S}$, så må $\mathcal{S}'$ være konstrueret ved hjælp af (T$\wedge$)-reglen. Vores induktionsantagelse siger, at der må findes en sekvent $\langle \Gamma'; \Delta' \rangle \in \mathcal{S}'$ så $\Gamma' \vdash_i \Delta'$. Hvis $\langle \Gamma'; \Delta' \rangle \in \mathcal{S}$, så har vi en beviselig sekvent i $\mathcal{S} \cup \langle \Gamma; \Delta \rangle$. Hvis $\langle \Gamma'; \Delta' \rangle \notin \mathcal{S}$, så må vi have $\langle \Gamma'; \Delta' \rangle = \langle \Gamma \cup \{\varphi_1, \varphi_2\}; \Delta \rangle$. Vi får altså at

$$\cfrac{\begin{matrix}\Gamma & \varphi_1 & \varphi_2 \\ & \vdots & \\ & \ldots & \end{matrix}}{\Delta}$$

og da vi ved ud fra tableaureglerne, at $\varphi_1 \wedge \varphi_2 \in \Gamma$, får vi, ved brug af ($\wedge$E)-reglen, at

$$\dfrac{\Gamma \quad \dfrac{\varphi_1 \wedge \varphi_2}{\varphi_1}\wedge\mathrm{E} \quad \dfrac{\varphi_1 \wedge \varphi_2}{\varphi_2}\wedge\mathrm{E}}{\begin{array}{c}\vdots\\ \ldots\\ \hline \Delta\end{array}}$$

så $\Gamma \vdash_i \Delta$.

(F$\wedge$) Hvis $\mathcal{S} \cup \langle \Gamma; \Delta \rangle$ har to børn $\mathcal{S}_1$ og $\mathcal{S}_2$, som opfylder at $\mathcal{S}_1 = \langle \Gamma; \Delta \cup \{\varphi_1\} \rangle \cup \mathcal{S}$ og $\mathcal{S}_2 = \langle \Gamma; \Delta \cup \{\varphi_2\} \rangle \cup \mathcal{S}$, så må $\mathcal{S}_1$ og $\mathcal{S}_2$ være konstrueret ved (F$\wedge$)-reglen. Vores induktionsantagelse siger, at der må findes to sekventer $\langle \Gamma_1; \Delta_1 \rangle \in \mathcal{S}_1$ og $\langle \Gamma_2; \Delta_2 \rangle \in \mathcal{S}_2$ så $\Gamma_1 \vdash_i \Delta_1$ og $\Gamma_2 \vdash_i \Delta_2$. Hvis $\langle \Gamma_1; \Delta_1 \rangle \in \mathcal{S}$ eller $\langle \Gamma_2; \Delta_2 \rangle \in \mathcal{S}$, så har vi en beviselig sekvent i $\mathcal{S} \cup \langle \Gamma; \Delta \rangle$. Hvis ikke, så må der gælde at

$$\begin{array}{c}\Gamma\\ \vdots\\ \ldots\\ \hline \varphi_1 \quad \Delta\end{array} \qquad \begin{array}{c}\Gamma\\ \vdots\\ \ldots\\ \hline \varphi_2 \quad \Delta\end{array}$$

og da vi ved ud fra tableaureglerne $\varphi_1 \wedge \varphi_2 \in \Delta$, bruger vi ($\wedge$I)-reglen til at få

$$\dfrac{\begin{array}{c}\Gamma\\ \vdots\\ \ldots\\ \hline \varphi_1 \quad \varphi_2 \quad \Delta\end{array}}{\Delta}\wedge\mathrm{I}$$

så $\Gamma \vdash_i \Delta$.

(T$\vee$) Hvis $\mathcal{S} \cup \langle \Gamma; \Delta \rangle$ har to børn $\mathcal{S}_1$ og $\mathcal{S}_2$, som opfylder at $\mathcal{S}_1 = \langle \Gamma \cup \{\varphi_1\}; \Delta \rangle \cup \mathcal{S}$ og $\mathcal{S}_2 = \langle \Gamma \cup \{\varphi_2\}; \Delta \rangle \cup \mathcal{S}$, så må $\mathcal{S}_1$ og $\mathcal{S}_2$ være konstrueret ud fra (T$\vee$)-reglen. Ifølge induktionsantagelsen findes der to sekventer $\langle \Gamma_1; \Delta_1 \rangle \in \mathcal{S}_1$ og $\langle \Gamma_2; \Delta_2 \rangle \in \mathcal{S}_2$ så $\Gamma_1 \vdash_i \Delta_1$ og $\Gamma_2 \vdash_i \Delta_2$. Hvis $\langle \Gamma_1; \Delta_1 \rangle \in \mathcal{S}$ eller $\langle \Gamma_2; \Delta_2 \rangle \in \mathcal{S}$, så har vi en beviselig sekvent i $\mathcal{S} \cup \langle \Gamma; \Delta \rangle$. Hvis ikke, så har vi at

$$\begin{array}{c}\Gamma \quad \varphi_1\\ \vdots\\ \ldots\\ \hline \Delta\end{array} \qquad \begin{array}{c}\Gamma \quad \varphi_2\\ \vdots\\ \ldots\\ \hline \Delta\end{array}$$

og da vi ved fra tableaureglerne, at $\varphi_1 \vee \varphi_2 \in \Gamma$, får vi, ved brug af ($\vee$E)-reglen, at

$$\dfrac{\begin{array}{c}\Gamma \quad [\varphi_1] \quad [\varphi_2]\\ \vdots\\ \hline \Delta\end{array} \qquad \varphi_1 \vee \varphi_2}{\Delta}\;\vee\mathrm{E}$$

så $\Gamma \vdash_i \Delta$.

(F$\vee$) Hvis $\mathcal{S} \cup \langle \Gamma; \Delta \rangle$ har ét barn $\mathcal{S}'$, som opfylder at $\mathcal{S}' = \langle \Gamma; \Delta \cup \{\varphi_1, \varphi_2\} \rangle \cup \mathcal{S}$, så må $\mathcal{S}'$ være konstrueret ved hjælp af (F$\vee$)-reglen. Vores induktionsantagelse siger, at der må findes en sekvent $\langle \Gamma'; \Delta' \rangle \in \mathcal{S}'$ så $\Gamma' \vdash_i \Delta'$. Hvis $\langle \Gamma'; \Delta' \rangle \in \mathcal{S}$, så har vi en beviselig sekvent i $\mathcal{S} \cup \langle \Gamma; \Delta \rangle$. Hvis ikke, så har vi at

$$\begin{array}{c}\Gamma\\ \vdots\\ \hline \Delta \quad \varphi_1 \quad \varphi_2\end{array}$$

og kan så anvende ($\vee$I)-reglen til at konkludere at

$$\begin{array}{c}\Gamma\\ \vdots\\ \hline \Delta \qquad \dfrac{\varphi_1}{\varphi_1 \vee \varphi_2}\,\vee\mathrm{I} \qquad \dfrac{\varphi_2}{\varphi_1 \vee \varphi_2}\,\vee\mathrm{I}\end{array}$$

og da vi ved fra tableaureglerne, at $\varphi_1 \vee \varphi_2 \in \Delta$, får vi altså at $\Gamma \vdash_i \Delta$.

(T$\to$) Hvis $\mathcal{S} \cup \langle \Gamma; \Delta \rangle$ har to børn $\mathcal{S}_1$ og $\mathcal{S}_2$, som opfylder at $\mathcal{S}_1 = \langle \Gamma; \Delta \cup \{\varphi_1\} \rangle \cup \mathcal{S}$ og $\mathcal{S}_2 = \langle \Gamma \cup \{\varphi_2\}; \Delta \rangle \cup \mathcal{S}$, så må $\mathcal{S}_1$ og $\mathcal{S}_2$ være konstrueret ud fra tableaureglen (T$\to$). Vores induktionsantagelse siger, at der må eksistere to sekventer $\langle \Gamma_1; \Delta_1 \rangle \in \mathcal{S}_1$ og $\langle \Gamma_2; \Delta_2 \rangle \in \mathcal{S}_2$ så $\Gamma_1 \vdash_i \Delta_1$ og $\Gamma_2 \vdash_i \Delta_2$. Hvis $\langle \Gamma_1; \Delta_1 \rangle \in \mathcal{S}$ eller $\langle \Gamma_2; \Delta_2 \rangle \in \mathcal{S}$, så har vi en beviselig sekvent i $\mathcal{S}$. Hvis ikke, så har vi at

$$\begin{array}{c}\Gamma\\ \vdots\\ \hline \Delta \quad \varphi_1\end{array} \qquad\qquad \begin{array}{c}\Gamma \quad \varphi_2\\ \vdots\\ \hline \Delta\end{array}$$

desuden ved vi fra tableaureglerne, at $\varphi_1 \to \varphi_2 \in \Gamma$. Så ved brug af ($\rightarrow$E)-reglen, får vi at

$$\cfrac{\cfrac{\cfrac{\Gamma \atop \vdots}{\Delta \quad \varphi_1} \quad \varphi_1 \rightarrow \varphi_2}{\varphi_2}\rightarrow\mathrm{E} \quad \Gamma}{\cfrac{\vdots}{\Delta}}$$

så $\Gamma \vdash_i \Delta$.

(F$\rightarrow$) Hvis $\mathcal{S} \cup \langle \Gamma; \Delta \rangle$ har ét barn $\mathcal{S}'$, som opfylder at $\mathcal{S}' = \langle \Gamma; \Delta \rangle \cup \mathcal{S} \cup \langle \Gamma \cup \{\varphi_1\}; \{\varphi_2\} \rangle$. så må $\mathcal{S}'$ være konstrueret ud fra tableaureglen (F$\rightarrow$). Vores induktionsantagelse giver, at der findes en sekvent $\langle \Gamma'; \Delta' \rangle \in \mathcal{S}'$, hvorom der gælder at $\Gamma' \vdash_i \Delta'$. Hvis $\langle \Gamma'; \Delta' \rangle \in \mathcal{S} \cup \langle \Gamma; \Delta \rangle$, så har vi en beviselig sekvent i $\mathcal{S} \cup \langle \Gamma; \Delta \rangle$. Hvis ikke, så har vi

$$\cfrac{\Gamma \quad \varphi_1 \atop \vdots}{\varphi_2}$$

og ved anvendelse af ($\rightarrow$I)-reglen får vi da at

$$\cfrac{\cfrac{\Gamma \quad [\varphi_1] \atop \vdots}{\varphi_2}}{\varphi_1 \rightarrow \varphi_2}\rightarrow I$$

så $\Gamma \vdash_i \varphi_1 \rightarrow \varphi_2$ og da $\varphi_1 \rightarrow \varphi_2 \in \Delta$ ifølge tableaureglerne, kan vi altså konkludere at $\Gamma \vdash_i \Delta$.

$\square$

Vi har altså vist, at hvis vi har et system $\mathcal{S}$ bestående af kun én sekvent $\langle \Gamma; \varphi \rangle$, hvorom der gælder at $\Gamma \Vdash \varphi$, så findes der et vellykket tableau med $\mathcal{S}$ som rod, samt at hvis $\mathcal{S}$ er rod i et vellykket tableau, så findes der en sekvent i $\mathcal{S}$ som er beviselig. Da $\mathcal{S}$ kun består af én sekvent $\langle \Gamma; \varphi \rangle$, får vi altså at $\Gamma \vdash_i \varphi$, og vi konkluderer at $\Gamma \Vdash \varphi \Rightarrow \Gamma \vdash_i \varphi$, altså at den intuitionistiske udsagnslogik er fuldstændig.

KAPITEL 5

Implementering af fuldstændighedsbeviset

"An algorithm must be seen to be believed."
— Donald E. Knuth [Knu97]

I dette kapitel giver vi en beskrivelse af vores implementering af fuldstændighedsbeviset fra kapitel 4. Vi giver kun en implementering af sætning 4.7, hvilket egentlig betyder vi kun har en delvis implementering af fuldstændighedsbeviset. Vores fokus er derfor på konstruktion af et tableau givet et udsagn. Desuden giver vi en implementering af en metode til at tjekke om et givet tableau er vellykket. Dette giver faktisk tilsammen en automatisk bevistjekker, eller nærmere en bevisgenerator, da algoritmen i sætning 4.7 egentlig konstruerer et bevis.

Vi har valgt det funktionelle programmeringssprog Haskell som implementeringssprog. Egentlig kunne vi have valgt et hvilket som helst sprog, men da vi ikke har nogen tidligere erfaring med funktionel programmering, så har det været et ønske at prøve dette.

5.1 Syntaktisk sukker

Dette kapitel forsøger på ingen måde at give en introduktion til Haskell og funktionel programmering, men vi vil give en uformel beskrivelse af et par essentielle begreber i Haskell, særligt fordi disse er nye for os.

Vores kildekodeeksempler indeholder pseudo-Haskell-kode, dvs. det er ikke Haskell-kode der kan evalueres uden modifikation. Vi gør dette for at gøre kildekodeeksemplerne mere læsevenlige, og at eksemplerne notationsmæssigt så vidt muligt ligner vores forudgående arbejde. Eksempelvis benytter vi de græske symboler $\mathit{\Gamma}, \Delta, \varphi, \ldots$ istedet for at skrive "Gamma, Delta, phi". Se tabel 5.1 for en oversættelse af symbolerne til Haskell-syntaks. For den fulde og faktiske kildekode henviser vi til det online projekttitelblad (link findes i forordene, på side vii), hvor den kan downloades.

Symboler	Oversættelse
$\mathit{\Gamma}, \Delta, \varphi, \mathcal{S}, (T\wedge)$	`gamma, delta, phi, system, tAnd`
$\lambda, \rightarrow$	`\, ->`
$\cap, \cup$	`` `Set.intersection`, `Set.union` ``
$\{\varphi_1, \varphi_2\}$	`Set.insert phi1 (Set.singleton phi2)`
$\notin, \neq$	`` `Set.notMember`, /= ``

Tabel 5.1: Oversættelse af symboler til Haskell-kode.

Haskell er baseret på λ-kalkylen, der skyldes Alonzo Church [Hü10]. Ideen om λ-udtryk er helt essentielt i λ-kalkylen. For os er det tilstrækkeligt at forstå et λ-udtryk som værende en anonym funktion. I Haskell udtrykkes et λ-udtryk ved følgende syntaks:

$$\lambda x \rightarrow e,$$

hvor e er et vilkårligt Haskell-udtryk, muligvis selv et λ-udtryk. Vi siger, at x er den bundne variabel, og at e er λ-udtrykkets *krop* [Hü10]. Enhver forekomst af x i e erstattes med den aktuelle værdi af x på kaldstidspunktet.

Haskell giver en syntaktisk genvej til at bygge lister ud fra en række kriterier. Det svarer næsten til mængdebyggernotationen vi kender fra matematikken, forskellen er blot at resultatet er en liste og ikke en mængde. Lad os illustrere dette ved et eksempel:

```
[ x | x <- [1..10], x `mod` 2 == 0 ],
```

resultatet er en liste med alle de lige tal mellem 1 og 10 (inklusiv). Denne teknik kaldes *list comprehension* [Hut07]. Til sammenligning kan vi i mængdebyggernotation udtrykke overstående som

$$\{x \mid x \in \{1, \ldots, 10\},\ \text{hvor } x \equiv 0 \pmod 2\}.$$

Sædvanligvis betyder kommaet i denne kontekst "og". List comprehension giver en koncis metode til at generere nye lister fra allerede eksisterende lister, hvorom givne betingelser er opfyldt.

Da Haskell er et rent funktionelt sprog, så beskrives alt vha. funktioner. I princippet kan et Haskell-program opfattes som en funktion, der kalder andre funktioner for at opnå programmets formål. En funktion er i princippet et navn, der er tilknyttet et λ-udtryk. Funktionstyper beskrives ved en syntaks, der minder en meget om den vi kender fra matematikken. Tag for eksempel

```
doSomething :: a → b → c,
```

som er en funktion, der tager to argumenter: et argument af typen `a` og et af `b`. Til gengæld returnerer den en værdi af typen `c`. Operatoren $\rightarrow$ er højre-associativ [Hut07], dvs. følgende er ækvivalent med overstående

$$\texttt{doSomething :: a} \rightarrow \texttt{(b} \rightarrow \texttt{c).}$$

Vi gør os her en særlig observation. Faktisk tager `doSomething` et argument af typen `a` og returnerer til gengæld en funktion, der tager et argument af typen `b` og returnerer en værdi af typen `c`. Vi kan tænke på det, som at `doSomething` tager et argument og returnerer et λ-udtryk som tager det næste argument osv. Denne teknik, hvor funktioner tager deres argumenter ét ad gangen, kaldes *currying* [Hut07].

5.2 Modellering af strukturer

I dette afsnit vil vi gennemgå, hvordan vi modellerer tableauer, systemer, sekventer og udsagn.

5.2.1 Udsagn

Vi er selvfølgelig nødt til at have en repræsentation af et udsagn, for at der overhovedet er noget at arbejde med. I kapitel 3 gav vi en induktiv definition af et udsagn vha. en grammatik. Faktisk svarer vores implementering nøjagtigt til grammatikken (3.1). Kildekodeeksempel 5.1 viser vores implementering i pseudo-Haskell-kode.

Kildekode 5.1: Implementering af grammatikken

```
#1  type AtomProp = String

#2  data Formula  = Atom AtomProp
                  | Bot
                  | Formula `And` Formula
                  | Formula `Or` Formula
                  | Formula `Implies` Formula
```

Selve repræsentationen af grammatikken i vores program stemmer meget godt overens med (3.1). `#1` og `#2` er medvirkende til at implementeringen adskiller sig fra (3.1). Nøgleordene `type` og `data` er reserverede ord i Haskell. Nøgleordet `type` anvendes til at lave et typealias, eksempelvis fortæller vi at typen `AtomProp` er et alias for typen `String`. Dette er medvirkende til at øge læsbarheden af koden, og kan fungere som en abstraktion over en mere kompliceret type. Nøgleordet `data` anvendes også til at definere en type, men modsat `type` kan vi her angive, hvordan den abtrakte datatype konstrueres. På venstreside af lighedstegnet står typens navn, som vi har kaldt `Formula`. På den højre side står hvordan typen kan konstrueres. Vi kan altså konstruere `Formula` ved brug af konstruktorer `Atom`, `Bot` ($\perp$), `And` ($\wedge$), `Or` ($\vee$), `Implies` ($\rightarrow$), fuldstændig ligesom vi beskrev i afsnit 3.1.

5.2.2 Tableauer, systemer og sekventer

Vores repræsentation af tableauer, systemer og sekventer er i nogen grad inspireret af [Und90]. Kildekodeeksempel 5.2 viser, hvorledes vi beskriver dem i vores implementering.

Kildekode 5.2: Implementering af tableauer, systemer og sekventer

```
#1  type Sequent = (Set Formula, Set Formula)
#2  type System  = [Sequent]
#3  type Tableau = Tree System
```

Vi har valgt at repræsentere en sekvent `#1` som et par, hvor hver komponent er en mængde af udsagn. Dette stemmer også overens med den definition vi gav af en sekvent i afsnit 4.1. Til at repræsentere en mængde benytter vi den stærke abstrakte datatype `Set`, som er en del af standardbiblioteket i Haskell. Vi siger, at `Set` er en *stærk* abstrakt datatype, fordi den skjuler sine implementeringsdetaljer. Derimod siger vi at liste-typen `[a]` er en *svag* abstrakt datatype, hvilket betyder at vi kender dens implementeringsdetaljer og derfor kan vi udnytte dem. Et system `#2` har vi netop valgt at repræsentere som en liste af sekventer. Den umiddelbare grund til vi har valgt at benytte `Set` og ikke listetypen til at repræsentere en mængde af udsagn er, at der for `Set` allerede findes implementeringer af operationerne $\cup$ og $\cap$. Særligt $\cap$ får vi brug for i forbindelse med at tjekke om et blad er lukket.

Endeligt har vi valgt at repræsentere et tableau `#3` som et træ, hvor hver knude er et system. Det er selvfølgelig et meget naturligt valg set i lyset af den definition vi gav af et tableau. Desuden er `Tree` også en svag abstrakt datatype.

5.3 Tableaukonstruktions algoritmen

I dette afsnit beskriver vi, hvordan vi overordnet har implementeret algoritmen i beviset for sætning 4.7. Vi giver ikke alle implementeringsdetaljerne her, dvs. vi beskriver ikke hver enkelt funktion og tableaukonstruktør.

5.3.1 Tableaukonstruktorerne

I afsnit 4.1 beskrev vi tableaukonstruktionsreglerne, og vi giver nu en beskrivelse af vores implementering for en regel. Grunden til vi ikke viser dem alle er, at reglerne i store træk er ens, og varierer blot i deres sidebetingelser og deres output (et eller to systemer). Kildekodeeksempel 5.3 viser typen af en konstruktør.

Kildekode 5.3: Typen for tableaukonstruktionsreglerne

```
type Constructor    = System → Maybe [System]
```

```
#2  caseApplies           :: Constructor → System → Bool
    caseApplies c 𝒮     = case c 𝒮 of
                            Just _   → True
                            Nothing  → False
```

En konstruktør `#1` er defineret som en funktion, der tager et system som input, og *måske* returnerer en liste af systemer. Vi har valgt, at implementere hver tableaukonstruktionsregel, således at den returnerer enten et eller to systemer, hvis den kan anvendes på et givet system, ellers returnerer den "ingenting". Typen `Maybe` er en abstrakt datatype, hvis implementeringsdetaljer er:

`data Maybe a = Just a | Nothing.`

Vi siger at `Maybe` er en *parametriseret type*; typeparameteren `a` kan erstattes med en vilkårlig type.

Funktionen `caseApplies #2` efterprøver om en given tableaukonstruktionsregel `c` kan anvendes på et givet system $\mathcal{S}$. I funktionen udnytter at vi kender til implementeringsdetaljerne af `Maybe`, og vi spørger derfor i "`case c` $\mathcal{S}$ `of`" om hvordan returværdien er konstrueret. Hvis den er konstrueret vha. `Just` returneres sand, ellers returneres falsk. Denne teknik kaldes *pattern matching*, og er en teknik man kan anvende på alle svage abstrakte datatyper.

I kildekodeeksempel 5.4 viser vi, hvordan vi har implementeret tableaukonstruktionsreglen $(T\wedge)$. Da eksemplet fylder lidt mere end de foregående eksempler, har vi inkluderet små kommentarer inde i eksemplet.

Kildekode 5.4: Implementering af $(T\wedge)$

```
    (T∧) :: Constructor
    (T∧) 𝒮
#1   | meets_cond = Just [𝒮']
     | otherwise  = Nothing
       where
#2      meets_cond = length validsequents > 0
        -- Sidebetingelsen
#3      cond Γ̃ = (λ(φ1 `And` φ2)→ (φ1 ∉ Γ̃)) || (φ2 ∉ Γ̃))
        -- Find de sekventer, der opfylder betingelserne
#4      validsequents= [((Γ',Δ'),u) | (Γ',Δ') ← 𝒮, let u =
      findConjunction (cond Γ') Γ', u ≠ Nothing]
#5      (sequent,u)  = head validsequents
        (Γ,Δ)        = sequent
        -- Udtraek de direkte deludsagn fra konjunktionen
#6      (Just (φ1 `And` φ2)) = u
        -- Konstruktion af det nye system
#7      sequent' = ({φ1,φ2} ∪ Γ, Δ)
#8      𝒮'       = sequent' : (filter (λx→x ≠ sequent) 𝒮)
```

Ved #1 tjekker vi om det givne system $\mathcal{S}$ opfylder betingelserne for anvendelse af tableaureglen $(T\wedge)$. I #4 benytter vi *list comprehension* til at finde alle de sekventer, hvor udsagnsmængden Γ' indeholder en konjunktion samt opfylder sidebetingelsen #3. Den resulterende liste er af typen `[(Sequent,Formula)]`, hvor udsagnet i andenkomponenten er den konjunktion, der opfylder sidebetingelsen. Funktionen `findConjunction` finder, som navnet antyder, en konjunktion i en given mængde af udsagn. Funktionen har typen

```
findConjunction :: (Formula → Bool) → Set Formula → Maybe Formula.
```

Den tager en boolesk funktion som input. Denne funktion angiver om et enkelt konjunktionsudsagn opfylder betingelserne. Denne funktion bliver anvendt på hvert eneste udsagn, der er en konjunktion, i udsagnsmængden Γ'. Sidebetingelsen #3 er præcis denne funktion, som bliver givet som aktuel parameter til `findConjunction`. I #2 tjekker vi, om der findes en eller flere sekventer, som opfylder betingelserne. Hvis der gør det, så vælger vi blot den første i listen #5.

I #6 udtrækker vi de umiddelbare deludsagn fra konjunktionen `u`. Til dette benytter vi igen pattern matching. Derefter konstrueres den nye sekvent #7. Denne indsættes bagefter i det nye system $\mathcal{S}'$ #8, hvor den originale sekvent er fjernet gennem en filtrering af listen $(\mathcal{S})$.

5.3.2 Konstruktion af et tableau

Vi har nu set et eksempel på implementering af en tableaukonstruktionsregel. I kildekodeeksempel 5.5 ser vi, hvordan funktionerne er med til at lave en instans af et tableau.

Kildekode 5.5: Konstruktion af et tableau

```
     tableau   :: Formula → Tableau
#1   tableau φ  = construct [(∅, {φ})]

     construct :: System → Tableau
     construct 𝒮
#2       | notLeaf   = Tree.Node 𝒮 (map construct systems)
#3       | otherwise = Tree.Node 𝒮 []
           where
            result         = applyCases 𝒮
            notLeaf        = result ≠ Nothing
            (Just systems) = result

#4   applyCases :: Constructor
     applyCases 𝒮
         | caseApplies (T∧) 𝒮  = (T∧) 𝒮
         | caseApplies (F∧) 𝒮  = (F∧) 𝒮
         | caseApplies (T∨) 𝒮  = (T∨) 𝒮
```

```
  | caseApplies (F∨) 𝒮   = (F∨) 𝒮
  | caseApplies (T →) 𝒮  = (T →) 𝒮
  | caseApplies (F →) 𝒮  = (F →) 𝒮
  | otherwise            = Nothing
```

Funktionen `tableau` `#1` konstruerer et tableau fra et givet udsagn. Den kalder funktionen `construct` med systemet $\mathcal{S} = \{\langle\emptyset; \{p\}\rangle\}$ som input. Selve konstruktionen af tableauet sker rekursivt, i hvert kald af `construct` konstrueres en knude i tableauet. Hvis den nye knude i tableauet ikke er et blad `#2`, så benyttes konstruktøren `Tree.Node` til at konstruere et træ, hvor $\mathcal{S}$ er rod. Børnene til $\mathcal{S}$ bliver konstrueret ved at kalde `construct` rekursivt på de nye systemer, som tableaukonstruktionsreglerne har returneret efter at have været anvendt på $\mathcal{S}$. Funktionen terminerer, når $\mathcal{S}$ er et blad `#3` og den konstruerede knude får den tomme liste `[]` som barn.

Tableaukonstruktionsreglerne bliver prøvet én efter én af funktionen `applyCases`. Den er af typen `Constructor`, fordi den opfører sig som en tableaukonstruktionsregel. Hvis ingen af tableaukonstruktionsreglerne kan anvendes på den aktuelle parameter $\mathcal{S}$, så returneres `Nothing`, og så ved funktionen `construct` at den skal konstruere et blad.

5.3.3 Et vellykket tableau

Til at tjekke om et givet tableau er vellykket, benytter vi to funktioner: `isClosed` og `isSuccessful`. Funktionen `isClosed` tjekker om betingelsen $\Gamma \cap \Delta \neq \emptyset$ er opfyldt for en given sekvent $\langle\Gamma; \Delta\rangle$. Derimod arbejder `isSuccessful` sig rekursivt igennem et givet tableau. Implementeringsdetaljerne findes i kildekodeeksempel 5.6

Kildekode 5.6: Tjek om et tableau er vellykket

```
     isClosed :: Sequent → Bool
#1   isClosed sq = (Γ ∩ Δ) ≠ ∅
                   where (Γ,Δ) = sq

     isSuccessful :: Tableau → Bool
#2   isSuccessful (Tree.Node 𝒮 subforest)
                  | subforest == [] = any isClosed 𝒮
                  | otherwise       = all isSuccessful
                    subforest
```

`isClosed` `#1` er en simpel boolesk funktion, der returnerer om fællesmængden mellem Γ og Δ er ikke-tom. Funktionen `isSuccessful` `#2` benytter pattern matching på et tableau. Hvis `subforest` er den tomme liste `[]`, så har vi et blad. Den booleske højereordensfunktion `any` tager `isClosed` og anvender den på enhver sekvent $\langle\Gamma; \Delta\rangle \in \mathcal{S}$, den returnerer sand hvis mindst én sekvent opfylder betingelsen. Hvis ikke vi har et blad, så anvendes `isSuccessful` rekursivt på alle børnene af $\mathcal{S}$.

5.4 Fra bevis til software

Vi har vist de essentielle dele af vores implementering. Faktisk er overgangen fra beviset for sætning 4.7 til et stykke software ikke så stor. Selvfølgelig er der en del syntaktiske konstruktioner i implementeringen, som ikke findes i det matematiske bevis. Det er dog bemærkelsesværdigt, at vi stort set bare har oversat induktionsbeviset direkte til et stykke Haskell-kode. Dette er eksempel på, hvor informationsrigt et konstruktiv bevis kan være kontra et indirekte bevis. Det lader til at definitionerne og resultaterne i den konstruktive matematik tilsammen udgører en form for abstrakt programbeskrivelse. Fra beskrivelsen er det op til programmøren at oversætte det til et programmeringssprog. Vi kan i hvertfald for vores fuldstændighedsbevis i kapitel 4, konkludere at det er direkte anvendeligt modsat det klassiske fuldstændighedsbevis vi gav for sætning 3.14.

KAPITEL 6

Konklusion

"But Nature flies from the infinite, for the infinite is unending or imperfect, and Nature ever seeks an end."

— Aristoteles (384-322 f.v.t.) [Li13]

Vores undren tog udgangspunkt i det udeladte tredje princip (afsnit 1.1). Som en reaktion på den matematiske krise i det 20. århundrede lagde Brouwer, med sin intuitionisme, grundlaget for den intuitionistiske skole indenfor matematikken. Modsat klassisk logik, så er UTP ikke universelt gyldigt i den intuitionistiske logik. Dette ledte os til den konstruktive matematikfilosofi, som baserer sig på den intuitionistiske logik og accepterer derfor ikke UTP. I den konstruktive matematik er begrebet *konstruktion* helt essentielt, hvilket bl. a. ses i det meget stærke eksistensbegreb, som den konstruktive matematik opererer med. For at påvise eksistensen af et matematik objekt, skal man ifølge den konstruktive matematik give en algoritme, som konstruerer dette objekt. Det er en helt anden tankegang end den, der findes i den klassiske matematik. Denne tankegang fangede vores nysgerrighed, og vi undersøgte derfor den konstruktive matematik og i særdeles den intuitionistiske logik nærmere med udgangspunkt i [vD08]. Til vores store overraskelse fandt vi at beviset for den intuitionistiske fuldstændighed er vist ved modstrid i [vD08]. Givet den konstruktive matematiks strenge beviskrav, samt det at den baserer sig på den intuitionistiske logik, var dette selvfølgelig meget iøjnefaldende. Endvidere fandt vi ikke blot dette fænomen i [vD08], men også i [SU98]. Ud fra dette formulerede vi vores problemformulering: *Hvordan kan man bevise fuldstændighedssætningen for intuitionistisk logik udelukkende ved brug af konstruktive argumenter?*

For at besvare dette måtte vi først forstå hvad fuldstændighedssætningen egentlig siger. I kapitel 3 beskæftigede vi os derfor med udsagnslogik, og beskrev både en syntaks for udsagnslogik der både omfatter klassisk og intuitionistisk udsagnslogik. Herefter beskrev vi bevisreglerne for udsagnslogik, igen både i en klassisk og en intuitionistisk udgave, og viste nogle af forskellene på de to logikker.

Vi beskrev semantikken for intuitionistisk udsagnslogik og gav på baggrund af alt dette et bevis for sundhed af intuitionistisk udsagnslogik samt et klassisk bevis for fuldstændighed af intuitionistisk udsagnslogik.

I kapitel 4 gav vi det konstruktive bevis for intuitionistisk fuldstændighed, som vi havde undret os over ikke var at finde i [vD08]. Dette bevis gjorde brug af sekventer og semantiske tableauer, og gav en algoritme for, hvordan man for ethvert udsagn enten kan konstruere et semantisk tableau, der beviser udsagnet, eller konstruere en modmodel til udsagnet.

6.1 Resultater

I afsnit 1.5 fremsatte vi vores problemformulering samt en række delspørgsmål, som vi ønskede at finde svar på. Vi vil her opsummere de svar, vi er fundet frem til.

6.1.1 Konstruktivt bevis for fuldstændighedssætningen

Hovedpunktet i vores problemformulering var spørgsmålet *"Hvordan kan man bevise fuldstændighedssætningen for intuitionistisk logik udelukkende ved brug af konstruktive argumenter?"* Som svar på dette, viste vi i kapitel 4 hvordan man kan give et bevis for fuldstændigheden af intuitionistisk udsagnslogik kun ved brug af konstruktive argumenter. Vores bevis gør brug af tableauregler, der ud fra et givet udsagn opbygger en træstruktur kaldet et semantisk tableau.

Tableaureglerne kræver nogle sidebetingelser for at kunne blive brugt, og hver tableauregel introducerer nye deludsagn af de allerede eksisterende udsagn. Da der kun kan være endeligt mange deludsagn af det oprindelige udsagn, vil tableauet derfor blive konstrueret efter endeligt mange anvendelser af tableaureglerne. Vi har vist, at for et færdigkonstrueret tableau gælder én af to egenskaber:

(i) hvis udsagnet er gyldigt, så har man et bevis for udsagnet,

(ii) ellers kan man konstruere en modmodel for udsagnet ud fra et af de åbne blade i tableauet.

Hvis vi har et bevis, kan dette konstrueres ved at betragte tableauet på hovedet, dvs. fra bladene og opefter. Så vil hver tableauregel svare til en af de bevisregler, vi beskrev i afsnit 3.2, og på denne måde kan et bevis opbygges ud fra tableauet. Hvis vi derimod har en modmodel, så kan denne konstrueres ved at opbygge en Kripke-model ud fra et bestemt blad i tableauet. På denne måde kommer det konstruerede tableau altså til at indeholde information, som kan bruges til enten at lave et bevis for det oprindelige udsagn eller at lave en modmodel for det oprindelige udsagn.

En klar forskel mellem det konstruktive bevis og det ikke-konstruktive bevis er at det konstruktive bevis er substantielt længere, hvilket kan være en af begrundelserne for at benytte et ikke-konstruktivt bevis i en lærebog som for eksempel [vD08].

På den anden side har det konstruktive bevis den fordel, at selve beviset udgør en algoritme for at konstruere et bevis, hvis et sådant findes, eller konstruere en modmodel hvis der ikke findes et bevis. På denne måde bliver beviset mere end blot et argument for, hvorfor sætningen er sand. Dermed kan beviset eksempelvis bruges til at implementere et stykke software, som automatisk kan generere beviser eller modmodeller. Sådanne automatiske bevisgeneratorer findes allerede i flere forskellige implementeringer og udgør et aktivt forskningsområde, se for eksempel [Ott12].

6.1.2 Forskelle mellem klassisk og intuitionistisk logik

Et andet af vores spørgsmål i problemformuleringen var *"Hvilke forskelle er der mellem klassisk logik og intuitionistisk logik?"* Vi diskuterede de filosofiske forskelle mellem klassisk logik og intuitionistisk logik i kapitel 1. I kapitel 3 formaliserede vi både den klassiske og den intuitionistiske udsagnslogik og kunne ud fra dette konkludere nogle mere konkrete forskelle.

Vi viste, at den klassisk gyldige ækvivalens mellem et udsagn og dets dobbelte negation, altså $p \leftrightarrow \neg\neg p$ ikke er gyldigt i den intuitionistiske udsagnslogik, hvilket hænger tæt sammen med intuitionismens afvisning af modstridsargumenter. Dog viste vi, at den ene vej, nemlig $p \to \neg\neg p$ stadig er gyldig i intuitionistisk udsagnslogik.

Desuden viste vi, at konnektivet $\vee$ ikke er strengt nødvendigt i klassisk udsagnslogik, da $\vee$ kan udtrykkes ud fra konnektiverne $\neg$ og $\wedge$. Det samme gør sig ikke gældende for intuitionistisk udsagnslogik, hvor $\vee$-konnektivet er nødvendigt. Også her er det dog muligt at vise den ene vej i intuitionistisk udsagnslogik.

Der kan selvfølgelig udledes andre forskelle mellem klassisk og intuitionistisk udsagnslogik, men især den første forskel er central, da den vedrører intuitionismens afvisning af modstridsargumenter.

6.1.3 Nødvendige resultater for fuldstændighedssætningen

Vores tredje spørgsmål i problemformuleringen lød *"Hvilke resultater er sætningen baseret på? Er de alle bevist ved brug af konstruktive argumenter?"* I afsnit 3.5.2 gav vi et klassisk bevis for fuldstændighed af intuitionistisk udsagnslogik. Til dette brugte vi to lemmaer: et om udvidelse af primisk teori til en anden primisk teori, og et om eksistensen af en modmodel. Interessant nok er begge disse lemmaer i høj grad konstruktive. I begge tilfælde bliver det objekt, det ønskes at vise eksistensen af, konstrueret. I begge tilfælde bliver der dog også brugt enkelte modstridsargumenter undervejs, og vi kan derfor ikke sige at lemmaerne er fuldstændig konstruktivistiske. Det kan derfor virke underligt, at selvom lemmaerne udviser en konstruktiv tilgang, så gøres de ikke fuldstændigt konstruktive.

6.1.4 Implementering af beviset

Det sidste spørgsmål i vores problemformulering var *"Hvordan kan vi implementere resultatet?"* Vi gav i kapitel 5 en delvis implementering af fuldstændighedsbeviset, dvs. vi implementerede kun sætning 4.7. Vi så i praksis hvor informationsrigt et konstruktiv bevis kan være. Beviset gav os algoritmen til at konstruere et tableau, og derefter skal man blot oversætte dette til et konkret programmeringssprog. På en måde kan vi sige at den konstruktive matematik giver os en abstrakt programbeskrivelse. Definitionerne specificere de matematiske objekter vi arbejder med, mens beviserne giver os algoritmen til at konstruere disse objekter.

Algoritmen implementerede vi i det funktionelle programmeringssprog Haskell.

6.2 Perspektivering

I dette projekt har vi udelukkende beskæftiget os med udsagnslogik. En umiddelbar udvidelse af projektet ville være at inkludre prædikatslogik. Præditkatslogikken bygger oven på udsagnslogikken, men hvor udsagnslogikken kun beskæftiger sig med lukkede udsagn, så beskæftiger prædikatslogikken sig med åbne udsagn. Dvs. prædikatslogikken arbejder med udsagn hvori der forekommer frie variabler, mens udsagnslogikken ikke tillader disse, heraf terminologien åbent og lukket.

Desuden har vi i vores arbejde med projektet flere steder set Curry-Howard-isomorfien nævnt, bl. a. i [Und90] og [SU98]. Curry-Howard-isomorfien er et resultat, der siger noget om sammenhængen mellem typer og beviser, og danner derfor en endnu stærkere sammenhæng mellem konstruktive beviser og deres implementering. Vi ville derfor gerne se på denne isomorfi som det næste i vores projekt, og forstå nogle af dens implikationer, samt undersøge, hvordan netop dette resultat kan bruges i en implementering af beviset. I samme forbindelse ville vi også gerne undersøge typeteori, som vi har set benyttet i [Und90].

Slutteligt er der mulighed for mange udvidelser af vores implementering. Det er selvfølgelig oplagt at implementere de resterende dele af fuldstændighedsbeviset, således at bevistjekkeren kan konstruere en modmodel ud fra et ikke-vellykket tableau. Derudover kunne vi gøre bevistjekkeren til en rigtig interaktiv bevistjekker. Desuden kunne det være interessant at implementere en komplet bevistjekker for både intuitionistisk logik og klassisk logik, således at det er muligt at tjekke et udsagn både intuitionistisk og klassisk.

BILAG A

Relationer og funktioner

I dette appendiks beskriver vi sammenhængen mellem relationer og funktioner. Indholdet anvendes kun direkte i beviset for sætning 2.9.

A.1 Relationer

Definition A.1 (Relation [Hü10])**.** Lad $A_1, \ldots, A_k$ være en familie af mængder. En k-nær relation mellem $A_1, \ldots, A_k$ er enhver delmængde af $A_1 \times \cdots \times A_k$. Hvis $k = 2$, siger vi at det er en binær relation.

For binære relationer benytter vi ofte infiksnotation, dvs. hvis R er en relation, så skriver vi aRb istedet for $(a, b) \in R$.

A.2 Funktioner

Vi kan forstå en funktion F som en binær relation mellem to mængder X og Y. Funktionen F tildeler entydigt ethvert element i $x \in X$ et element $y \in Y$. Med entydigt skal forstås, at der ikke findes et par i $X \times Y$, hvor førstekomponenterne er ens, dvs.

$$\text{hvis } (x, y), (x', y') \in X \times Y \text{ og } y \neq y', \text{ så } x \neq x'. \tag{A.1}$$

Definition A.2 (Total funktion [Hü10])**.** Lad $F \subseteq X \times Y$ være en binær relation mellem to vilkårlige mængder X og Y. F er en total funktion hvis alle elementer $x \in X$ entydigt tildeles et element $y \in Y$. Vi noterer dette som $F : X \to Y$, hvor X kaldes definitionsmængden ($\mathbf{Dm}\, F$) og Y værdimængden ($\mathbf{Vm}\, F$) af F.

Ofte lader man betegnelsen "total" være implicit, og anvender blot betegnelsen "funktion" istedet. For en funktionanvendelse $F(x) \in Y$ på et argument $x \in X$, siger vi at $F(x)$ er billedet af x i mængden Y.

Definition A.3. Lad $F : X \to Y$ være en funktion. Så siger vi at mængden

$$F(X) \stackrel{\text{def}}{=} \{x \in X \mid f(x) \in Y\} \subseteq X$$

er billedmængden af F, og vi kalder mængden

$$F^{-1}(Y) \stackrel{\text{def}}{=} \{y \in Y \mid y = f(x), x \in X\} \subseteq Y$$

for urbilledmængden af F.

Lemma A.4. *To funktioner er ens hvis og kun hvis deres definitionsmængder og værdimængder er ens.*

Bevis. Lad $f, g : X \to Y$ være to funktioner. Lad os betragte f og g som to binære relationer.

(⇒) Det følger af definition A.2 at hvis to funktioner er ens, så er deres definitionsmængder og værdimængder ens.

(⇐) Vi vil vise at $f \subseteq g$. Antag $(x, y) \in f$, så har vi $x \in \mathbf{Dm}\, f$, da $\mathbf{Dm}\, f = \mathbf{Dm}\, g$ må vi have at $x \in \mathbf{Dm}\, g$. Fordi $x \in \mathbf{Dm}\, g$ findes et $z \in \mathbf{Vm}\, g$ således at $(x, z) \in g$. Det følger af (A.1) at

$$(x, y) = (x, z) \in g,$$

heraf følger $y = z$, hvilket viser at $f \subseteq g$. Et analogt argument kan gives for inklusionen den anden vej $f \supseteq g$. Heraf $f = g$.

□

BILAG B

Bevisteknikker

I dette appendiks redegør vi for nogle udvalgte bevisteknikker, som vi har anvendt. Vi har allerede i afsnit 1.1 diskuteret modstridsbeviser, og vi behandler derfor ikke denne teknik i dette appendiks. Samme gælder for konstruktive beviser. Derimod vil vi diskutere induktionsbeviser og strukturel induktion.

B.1 Matematisk induktion

Bevis ved induktion er en særlig teknik til at vise alle elementer i en uendelig mængde besidder en særlig egenskab [Sip06]. Et induktionsbevis består af to dele: et basisskridt og et induktionsskridt. Lad os illustrere skabelonen for et induktionsbevis. Lad Φ være en vilkårlig egenskab for elementer i en mængde M, således $\Phi(k)$ betyder, at Φ er sand for det k'te element i M.

Basisskridt Vis at $\Phi(1)$ er sand.

Induktionsskridt For alle $k \geq 1$, antag at $\Phi(k)$ er sand og anvend denne antagelse til at vise $\Phi(k+1)$ er sand.

Antagelsen i induktionsskridtet kaldes *induktionsantagelsen* [Sip06]. Når vi har vist begge skridt, så har vi at $\Phi(k)$ er sand for alle $k \in \mathbb{N}$. Bevismetoden er gyldig, da basisskridtet viser at $\Phi(1)$ er sand, herefter vi ved at $\Phi(2)$ er sand, fordi induktionsskridtet viser at hvis $\Phi(1)$ er sand, så er $\Phi(2)$ sand, men vi ved allerede at $\Phi(1)$ er sand [Sip06]. Denne proces gentager sig for $\Phi(3)$, som er sand hvis $\Phi(2)$ er sand osv.

B.2 Strukturel induktion

Strukturel induktion er en særlige type af matematisk induktion, som anvendes til at give beviser for resultater på rekursivt definerede strukturer. Vi benytter eksempelvis strukturel induktion til at vise resultater for udsagn. Et strukturelt induktionsbevis består ligesom et induktionsbevis af to skridt. Vi giver følgende generelle skitse for et strukturelt induktionsbevis [Ros07]:

Basisskridt Vi viser, at resultatet holder for alle $x \in \mathcal{F}(\emptyset)$.

Rekursionsskridt Vi viser, at hvis resultatet holder for konstruktionerne $x_1, \ldots, x_n$, så holder resultatet også for den nye konstruktion $f_i(x_1, \ldots, x_n)$.

Gyldigheden af strukturel induktion følger af matematisk induktion [Ros07]. Betragt følgende resultat for eksemplets skyld, som vi viser ved brug af strukturel induktion.

Proposition B.1. *Ethvert udsagn φ har et lige antal parenteser.*

Bevis. Lad φ være et vilkårligt udsagn. Vi definerer funktionen

$$N(\varphi) = \begin{cases} 0 & \text{hvis } \varphi \text{ er atomar eller } \varphi = \bot. \\ 2 + N(\varphi_1) + N(\varphi_2) & \text{hvis } \varphi = (\varphi_1 \Box \varphi_2). \\ 2 + N(\varphi_1) & \text{hvis } \varphi = (\neg \varphi_1). \end{cases}$$

der returnerer antallet af parenteser i et udsagn φ.

Basisskridt Vi har at φ er et atomar udsagn eller $\varphi = \bot$, derfor $N(\varphi) = 0$, og 0 er et lige tal.

Rekursionsskridt Vi har at φ er et sammensat udsagn. Der er to tilfælde:

$\varphi = (\varphi_1 \Box \varphi_2)$ Antag $2m = N(\varphi_1)$ og $2n = N(\varphi_2)$ for $m, n \in \mathbb{N}$. Vi har per induktionsantagelse, at $N(\varphi) = 2 + 2m + 2n = 2(m + n + 1)$ for $m, n \in \mathbb{N}$, hvilket viser at φ har et lige antal parenteser.

$\varphi = (\neg \varphi_1)$ Antag $2n = N(\varphi_1)$ for $n \in \mathbb{N}$. Vi har per induktionsantagelse, at $N(\varphi) = 2 + 2n = 2(n + 1)$ for $n \in \mathbb{N}$, hvilket viser at φ har et lige antal parenteser.

□

BILAG C

Biografi

"The manifestations of the self within the restrictions and in the forms of this life are irruptions of truth."

— L. E. J. Brouwer (1881-1966) [Bro96]

L. E. J. Brouwer kan betragtes som den konstruktive matematiks fader, og derfor som en central person i den konstruktive matematiks historie. Vi giver derfor i dette appendiks en kort biografi af Brouwer, samt en kort redegørelse for Brouwer-Hilbert kontroversen for at afslutte historien påbegyndt i afsnit 1.2, 1.2.1 og 1.2.2.

C.1 L. E. J. Brouwer

Brouwers fulde navn er *Luitzen Egbertus Jan Brouwer*, traditionelt set citeres han som L. E. J. Brouwer eller blot Brouwer [OR13]. I figur C.1 ses et billede af Brouwer. Brouwer blev født den 27. februar 1881 og døde den

Figur C.1: Portræt af L. E. J. Brouwer [S913]

2. december 1966 i et trafikuheld i en alder af 85 [OR13]. Han var en hol-

landsk matematiker og matematikfilosof, der dog ikke kun filosoferede over matematik, men også over livet. I sin bog "*Life, Art, and Mysticism*" (1905) [Bro96] giver han et filosofisk syn på mennesket, religion, kærlighed, kunst og samfund.

I 1907 blev hans ph.d.-afhandling "*On the foundations of mathematics*" (1907) publiceret, hvilken gav et stort bidrag til den daværende debat om matematikkens grundlag [OR13]. Bidraget blev dog ikke vel mødt af alle. Der var nogle som anså Brouwers filosofiske ideer om matematikkens grundlag for at være kontroversielle. Brouwer blev faktisk hurtigt opmærksom på, at ikke alle var lige begjestret for hans tanker [OR13]. Korteweg, som var Brouwers ph.d.-vejleder, var ikke glad for de filosofiske ideer i afhandlingen; han havde endda krævet at flere af disse blev fjernet fra arbejdsbladene. Korteweg bad desuden Brouwer om at koncentrere sig om mere "respektabel matematik", således han kunne forbedre sit matematiske ry og sikre sig en akademisk karriere [OR13]. Brouwer fortsatte dog med at udvikle sine ideer, hvilke skulle ende med at danne rammen for intuitionisme og senere den intuitionistiske logik.

Brouwer beskæftigede sig i sin karriere også med topologi og viste flere store resultater indenfor for emnet. Det resultat han måske er mest kendt for, er sit fikspunktresultat indenfor topologien. Han ligger endda selv navn til resultatet, der er kendt som Brouwers fikspunktsætning.

Sætning C.1 (Brouwers fikspunktsætning [Cor13])**.** *Lad $K \subset \mathbb{R}^d$ være en kompakt, konveks mængde, og $f : K \to K$ være en kontinuert funktion. Så har f et (ikke nødvendigvis entydigt) fikspunkt, dvs. der findes et punkt $x \in K$ således at $f(x) = x$.*

Bevis. For et bevis se [Cor13]. □

Selve sætningen er i vores kontekst ikke interessant i sig selv, men det bemærkelsesværdige for os er, at beviset for sætningen indeholder modstridsargumenter. Brouwer brugte en stor del af sin karriere på at arbejde med konstruktiv matematik, så det må være skæbnens ironi, at netop det resultat, han er mest kendt for, er vist vha. modstrid.

David Hilbert beundrede Brouwer og var medvirkende til at Brouwer blev forfremmet til professor ved Amsterdam Universitet i 1912 [OR13]. Det var faktisk først herefter, at Brouwer for alvor begyndte at udvikle intuitionismen.

C.2 Brouwer-Hilbert kontroversen

I 1914 havde Brouwer fået en plads i redaktionen på "*Mathematische Annalen*", et tysk matematisk tidsskift. Godt 20 år senere blev Brouwer involveret i en offentlig kontrovers med Hilbert, som havde hjulpet Brouwer med at blive professor ved Amsterdam Universitet. Kontroversen handlede om den redaktionelle politik på tidsskriftet. Hilbert mente at Brouwer med sin intuitionisme var en trussel mod matematikken, og at han var begyndt at få for meget magt [OR13]. Derfor forsøgte Hilbert i 1928 at få fjernet Brouwer fra redaktionen, hvilket efter nogen stridigheder lykkedes. Kampen med Hilbert var nedværdigende for Brouwer, og endte med at tage hårdt på ham, og var en af flere hårde episoder, der skulle ramme ham i de senere år [OR13].

Litteratur

[Bis67] Errett Bishop. *Foundations of constructive analysis.* McGraw-Hill, 1967. Digitaliseret d. 22. maj 2008. 6

[Bri13] Douglas Bridges. Constructive mathematics. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Spring 2013 edition, 2013. 6

[Bro07] L. E. J. Brouwer. *Over de Grondslagen der Wiskunde*. PhD thesis, Universiteit van Amsterdam, 1907. Kilden er på hollandsk. 4

[Bro08] L. E. J. Brouwer. De onbetrouwbaarheid der logische principes. *Tijdschrift voor Wijsbegeerte*, pages 152–158, 1908. Kilden er på hollandsk. 4

[Bro12] L. E. J. Brouwer. Intuitionism and formalism. *American Mathematical Society*, 37:55–64, 1912. 4

[Bro96] L. E. J. Brouwer. Life, art, and mysticism. *Notre Dame Journal of Formal Logic*, 37:389–429, 1996. Oversat af Walter P. van Stigt. 57, 58

[Cor13] Horia Cornean. Notes for analyse 1 and analyse 2. `http://people.math.aau.dk/~cornean/analyse2_F13/noter-analyse1og2-2013.pdf`, April 2013. 58

[Fit69] Melvin Chris Fitting. *Intuitionistic Logic Model Theory and Forcing*. North-Holland Publishing Company Amsterdam, 1969. 32

[Gal86] Jean Gallier. *Logic for Computer Science: Foundations of Automatic Theorem Proving*. Wiley, 1986. Tilgængelig via `http://www.cis.upenn.edu/~jean/gbooks/logic.html`. 11, 13

[Gon05] Georges Gonthier. A computer-checked proof of the four colour theorem. `https://research.microsoft.com/en-us/um/people/gonthier/4colproof.pdf`, 2005. 7, 8

[Gö36] Kurt Gödel. *Über formal unentscheidbare Sätze der Principia Mathematica und verwandte Systeme I.* 1936. *On formally undecidable propositions of Principia Mathematica and related systems I* in Solomon Feferman, ed., 1986. *Kurt Gödel Collected works, Vol. I.* 4

[Har40] Godfrey H. Hardy. *A Mathematician's Apology.* 1940. http://todayinsci.com/H/Hardy_Godfrey/HardyGodfrey-Quotations.htm. 1

[Hey56] Arend Heyting. *Intuitionism: An Introduction.* North-Holland Publishing Company Amsterdam, 1956. 8

[Hor12] Leon Horsten. Philosophy of mathematics. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy.* Summer 2012 edition, 2012. 3, 4

[HP02] Vincent F. Hendricks and Stig Andur Pedersen. *Moderne elementær logik.* Høst & Søns Forlag, 2002. 17, 19, 20, 27

[Hut07] Graham Hutton. *Programming in Haskell.* Cambridge University Press, 2007. 42, 43

[Hü10] Hans Hüttel. *Transitions and Trees.* Cambridge University Press, 2010. 11, 24, 42, 53

[Joh07] Mikkel Willum Johansen. Computere i matematikken. http://www.dr.dk/DR2/VidenOm/Blogs/Huseksperternes+blog/Klummer+og+blogs/20070402130145.htm, April 2007. 7

[Jø11] Klaus Frovin Jørgensen. Om ideale elementer i matematikken. http://matematikfilosofi.ruc.dk/ideale.pdf, september 2011. 3

[Knu97] Donald Ervin Knuth. *The Art of Computer Programming*, volume 1: Fundamental Algorithms. Addison Wesley, third edition, 1997. 41

[KT28] Bronislaw Knaster and Alfred Tarski. Un théorème sur les fonctions d'ensembles. *Ann. Soc. Polon. Math.*, pages 285–309, 1928. 14

[Li13] Weiping Li. Famous math quotes. http://www.math.okstate.edu/~wli/teach/fmq.html, 2013. En samling af citater fra berømte matematikere. 17, 31, 49

[MJ12] Maarten McKubre-Jordens. Constructive mathematics. In Bradley Dowden and James Fieser, editors, *Encyclopedia of Philosophy: Online.* December 2012. ISSN 2161-0002. 5, 6, 8

[MRR88] Ray Mines, Fred Richman, and Wim Ruitenburg. A course in constructive algebra. http://math.fau.edu/richman/HTML/CONSTRUC.HTM, 1988. Forord, samt noter af F. Richman. 6

[OeS12] Tomás Oliveira e Silva. Goldbach conjecture verification. http://www.ieeta.pt/~tos/goldbach.html, April 2012. 5

[OR13] John J. O'Connor and Edmund F. Robertson. The mactutor history of mathematics archive. http://www-history.mcs.st-andrews.ac.uk, 2013. Et online matematik historisk arkiv. 2, 3, 6, 7, 11, 57, 58, 59

[Ott12] Jens Otten. Theorem provers. http://www.cs.uni-potsdam.de/~jeotten/provers.html, August 2012. 51

[Pal97] Erik Palmgren. Constructive mathematics. www.math.uu.se/~palmgren/cm-notes.ps, August 1997. 2, 8

[Ros07] Kenneth H. Rosen. *Discrete Mathematics and its Applications.* McGraw-Hill, international edition, 2007. 1, 2, 3, 5, 55, 56

[S913] S9. Bibliographical dictionary. http://www.s9.com/images/portraits/3948_Brouwer-Luitzen-Egbertus-Jan.jpg, May 2013. Fotografi af L. E. J. Brouwer. 57

[Sip06] Michael Sipser. *Introduction to the theory of computation.* Course Technology, international second edition, 2006. 55

[SU98] Morten Heine B. Sørensen and Pawel Urzyczyn. Lectures on the curry-howard isomorphism. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.17.7385, 1998. 9, 17, 49, 52

[Tar55] Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific J. Math.*, pages 285–309, 1955. 14

[Tro98] A.S. Troelstra. From constructivism to computer science. *Theoretical computer science*, pages 233–252, 1998. 1, 4, 6, 8, 9

[Und90] Judith Underwood. A constructive completeness proof for intuitionistic propositional calculus. 1990. 31, 33, 44, 52

[vD08] Dirk van Dalen. *Logic and Structure.* Springer, fourth edition, 2008. v, vi, 9, 11, 14, 15, 17, 19, 20, 22, 24, 26, 27, 28, 29, 30, 49, 50, 51

[Wad10] William R. Wade. *An Introduction to Analysis.* Pearson, fourth edition, 2010. 1